ΧΑΡΑΛΑΜΠΟΣ Ζ. ΠΑΤΡΙΚΑΚΗΣ
ΕΛΕΝΗ-ΑΙΚΑΤΕΡΙΝΗ ΛΕΛΙΓΚΟΥ
ΔΗΜΗΤΡΙΟΣ Γ. ΚΟΓΙΑΣ

ΑΛΥΣΙΔΕΣ ΣΥΣΤΟΙΧΙΩΝ (BLOCKCHAIN)

Εισαγωγή στην τεχνολογία και παραδείγματα

ΚΑΛΛΙΠΟΣ
ανοικτές ακαδημαϊκές εκδόσεις

Εθνικό Πρόγραμμα Ανάπτυξης 2021-2025

ΧΑΡΑΛΑΜΠΟΣ Ζ. ΠΑΤΡΙΚΑΚΗΣ
Καθηγητής

ΕΛΕΝΗ-ΑΙΚΑΤΕΡΙΝΗ ΛΕΛΙΓΚΟΥ
Αναπληρώτρια Καθηγήτρια

ΔΗΜΗΤΡΙΟΣ Γ. ΚΟΓΙΑΣ
Εντεταλμένος Διδάσκων, Ερευνητής

ΑΛΥΣΙΔΕΣ ΣΥΣΤΟΙΧΙΩΝ (BLOCKCHAIN)

Εισαγωγή στην τεχνολογία και παραδείγματα

ΚΑΛΛΙΠΟΣ
ανοικτές ακαδημαϊκές εκδόσεις

Αλυσίδες συστοιχιών (Blockchain)
Εισαγωγή στην τεχνολογία και παραδείγματα

Συγγραφή

Χαράλαμπος Ζ. Πατρικάκης (Κύριος Συγγραφέας)

Ελένη-Αικατερίνη Λελίγκου (Συν-συγγραφέας)

Δημήτριος Γ. Κόγιας (Συν-συγγραφέας)

Κριτικός αναγνώστης

Σούλλα Λουκά

Συντελεστές έκδοσης

Γλωσσική Επιμέλεια: Γιάννης Γαλανόπουλος

Γραφιστική Επιμέλεια: Κατερίνα Γαλάτη

Κεντρική Ομάδα Υποστήριξης

Γλωσσικός Έλεγχος: Δημήτρης Κονάχος, Γεωργία Τριανταφυλλίδου

Γραφιστικός Έλεγχος: Χρήστος Κεντρωτής

Βιβλιοθηκονομική Επεξεργασία: Ευδοξία Καρλή

ΚΑΛΛΙΠΟΣ
Εθνικό Μετσόβιο Πολυτεχνείο
Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN: 978-618-5726-49-2

Βιβλιογραφική Αναφορά: Πατρικάκης, Ζ. Χ., Λελίγκου, Αικ.-Ε., & Κόγιας, Γ. Δ. (2023). *Αλυσίδες συστοιχιών (Blockchain) – Εισαγωγή στις τεχνολογίες και παραδείγματα* [Μεταπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. http://dx.doi.org/10.57713/kallipos-171

Πίνακας Περιεχομένων

Πίνακας συντομεύσεων-ακρωνυμίων

AES	Advanced Encryption Standard
BIP	Bitcoin Improvement Proposal
BTC	Bitcoin Token
GDPR	General Data Protection Regulation
DAG	Directed Acycle Graphs
DApp	Decentralized Application
DeFi	Decentralized Finance
DLT	Distributed Ledger Technology
DNS	Dynamic Name Service
DoS	Denial of Service
DPoS	Delegated Proof of Stake
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EIP	Ethereum Improvement Proposal
ERC	Ethereum Request for Comments
ETH	Ethereum Token (Ether)
EVM	Ethereum Virtual Machine
FIFO	First In First Out
IPFS	InterPlanetary File System
MCMC	Markov Chain Monte Carlo
NFT	Non-Fungible Tokens
P2P	Peer – to -Peer
PoA	Proof of Activity / Authority
PoB	Proof of Burn
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
PoH	Proof of History
PoI	Proof of Importance
PoS	Proof of Stake
PoW	Proof of Work
RLP	Recursive Length Prefix
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RPoW	Reusable Proof of Work
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SPV	Simplified Payment Verification
TCP	Transmission Control Protocol
Tps	Transactions per second
UTXO	Unspent Transaction Output

Πρόλογος

Το βιβλίο *Αλυσίδες Συστοιχιών (Blockchain)*, με συγγραφείς τους Χαράλαμπο Πατρικάκη, Αικατερίνη-Ελένη Λελίγκου και Δημήτριο Κόγια, παρέχει μία σύνοψη της τεχνολογίας blockchain, μαζί με τις δυνατότητές της για καινοτομία όταν συνδυάζεται με άλλες τεχνολογίες, όπως IoT, AI, VR/AR, προς όφελος της ανθρωπότητας.

Το πρώτο Κεφάλαιο παρέχει μια βασική επισκόπηση της τεχνολογίας και των χαρακτηριστικών της, καθώς και τους διάφορους τύπους blockchains που υπάρχουν, μαζί με μια σύντομη ιστορική διαδρομή. Το δεύτερο Κεφάλαιο περιγράφει τις δύο δημοφιλέστερες αλυσίδες blockchain, το Bitcoin και το Ethereum, μαζί με τα έξυπνα συμβόλαια, ενώ το τρίτο Κεφάλαιο περιγράφει κύρια τεχνολογικά στοιχεία, όπως τα ψηφιακά κλειδιά και την κρυπτογραφία. Περιγράφεται η χρήση τους στη δημιουργία διευθύνσεων και στη συνέχεια με τη σειρά τους στη λειτουργία των πορτοφολιών. Στη συνέχεια, οι συγγραφείς περιγράφουν πώς πραγματοποιούνται οι συναλλαγές με παραδείγματα σε Bitcoin και Ethereum, καθοδηγώντας τον αναγνώστη/στρια βήμα προς βήμα. Αυτό βοηθά στην καλύτερη κατανόηση του υλικού. Το Κεφάλαιο 5 ασχολείται με τους αλγόριθμους συναίνεσης, ενώ το Κεφάλαιο 6 με τα έξυπνα συμβόλαια. Στα επόμενα κεφάλαια παρουσιάζονται τα Tokens, με αναφορά στο πρότυπο ERC του Ethereum, καθώς και στα αντίστοιχα πρότυπα άλλων δικτύων. Ακολούθως συζητούνται οι αποκεντρωμένες εφαρμογές (DApp) μαζί με διάφορα εργαλεία για την ανάπτυξη ενός DApp, παρέχοντας σε πιθανούς προγραμματιστές σημεία εκκίνησης για τον σχεδιασμό και την ανάπτυξη αποκεντρωμένου λογισμικού. Ένα από τα αγαπημένα μου σημεία είναι το Κεφάλαιο 9, όπου οι συγγραφείς παρουσιάζουν πώς μπορούμε να χρησιμοποιήσουμε και να εφαρμόσουμε την τεχνολογία blockchain. Σε αυτή την ενότητα δίνονται παραδείγματα τομέων και κλάδων της βιομηχανίας που μπορούν να εφαρμόσουν την τεχνολογία, καθώς και οι τρόποι με τους οποίους μπορούν να επωφεληθούν από αυτήν. Αυτά τα παραδείγματα έχουν εφαρμογή σε ευρύ φάσμα της οικονομίας, όπως, για παράδειγμα, στην εφοδιαστική αλυσίδα, στον ιατρικό τομέα, στον δημόσιο/κυβερνητικό τομέα και στο εμπόριο. Τα τελευταία κεφάλαια παρουσιάζουν τα χαρακτηριστικά των εφαρμογών που ανήκουν στην κατηγορία των Τεχνολογιών Κατανεμημένου Καθολικού και συζητούν την πρακτική παρουσίαση θεμάτων που σχετίζονται με χαρακτηριστικά της τεχνολογίας μέσω του δωρεάν εκπαιδευτικού εργαλείου ETH.build.

Ολόκληρο το βιβλίο είναι καλογραμμένο, με απλές εξηγήσεις για κάποιον που δεν έχει οποιοδήποτε υπόβαθρο στις νέες αυτές τεχνολογίες. Τα γραφήματα που υπάρχουν σε όλο το βιβλίο βοηθούν στην απλή εξήγηση των διάφορων εννοιών, οι οποίες για μερικούς ανθρώπους μπορεί να είναι δυσνόητες. Επιπλέον, τα παραδείγματα παρουσιάζονται λεπτομερώς, μαζί με στιγμιότυπα οθόνης, δίνοντας τη δυνατότητα στους αναγνώστες να τα εφαρμόσουν μόνοι τους, ώστε να εξασκηθούν για καλύτερη κατανόηση της τεχνολογίας.

Το εκπαιδευτικό περιβάλλον του blockchain εξελίσσεται συνεχώς. Πολλά πανεπιστήμια σε όλο τον κόσμο αναζητούν την εισαγωγή σχετικών μαθημάτων. Όσοι από εμάς σχεδιάζουμε και διδάσκουμε τέτοια μαθήματα αντιμετωπίζουμε το πρόβλημα του εντοπισμού του κατάλληλου αναγνωστικού υλικού για τους φοιτητές. Ως εκ τούτου, συνιστώ ανεπιφύλακτα το βιβλίο να μεταφραστεί και στα αγγλικά, καθώς πολλοί εκπαιδευτικοί, φοιτητές και ερευνητές θα το βρουν πολύ χρήσιμο όχι μόνο για τη χρήση του στην τάξη αλλά και για την ενίσχυση των γνώσεών τους.

Λουκά Σούλλα (Κριτικός αναγνώστης)

Εισαγωγή

Το βιβλίο αυτό έχει ως αντικείμενο του την τεχνολογία του blockchain, η οποία αποτελεί μία από τις πιο ενδιαφέρουσες τεχνολογίες που αναδείχθηκαν τα τελευταία χρόνια. Το ενδιαφέρον δεν εστιάζεται αποκλειστικά στην τεχνολογία αυτή από μόνη της, αλλά και από τις δυνατότητες που προσφέρει για να συνδυαστεί με άλλες υπάρχουσες (π.χ. Internet of Things) ή αναδυόμενες τεχνολογίες (π.χ. Artificial Intelligence, Augmented Reality), έτσι ώστε να προσφέρουν σύγχρονες λύσεις που μπορούν να βοηθήσουν τον άνθρωπο.

Ο σκοπός του βιβλίου είναι τριπλός:

- Να αποτελέσει ένα εγχειρίδιο αναφοράς το οποίο θα δώσει τα βασικά εφόδια στον αναγνώστη να κατανοήσει και να νιώσει άνετος να χρησιμοποιήσει βασικές γνώσεις πάνω στην τεχνολογία του blockchain.
- Να ενεργοποιήσει τον αναγνώστη να αναζητήσει περαιτέρω πληροφορίες για τα θέματα που τον ενδιαφέρουν περισσότερο στην τεχνολογία, έχοντας τις βάσεις για να προχωρήσει πιο βαθιά στα ενδιαφέροντά του.
- Να βοηθήσει να συμβούν όλα τα παραπάνω χρησιμοποιώντας την ελληνική γλώσσα, αφαιρώντας έτσι τυχόν εμπόδια που μπορεί να δημιουργούνται για κάποιους στην ενασχόληση με την τεχνολογία αυτή.

Σημαντικό είναι ότι για την ανάγνωση του βιβλίου ο/η αναγνώστης/τρια δεν χρειάζεται να έχει κάποια προηγούμενη γνώση καθώς οι προ-απαιτούμενες γνώσεις χτίζονται προοδευτικά μέσα σε αυτό. Με τον τρόπο αυτό δεν είναι αναγκαστικό να χρησιμοποιηθεί αποκλειστικά από όσους έχουν προηγούμενη εμπειρία στην τεχνολογία αυτή ή γενικότερα στον κλάδο της Πληροφορικής, αλλά και από όσους ενδιαφέρονται γενικότερα να καταλάβουν τι είναι το blockchain και πώς λειτουργεί.

Το βιβλίο αυτό χρησιμοποιεί τις δύο πιο γνωστές ανοικτές υλοποιήσεις της τεχνολογίας του blockchain, αυτές του Bitcoin και του Ethereum, για να εξηγήσει τα ιδιαίτερα χαρακτηριστικά της τεχνολογίας, τον τρόπο που αυτές οι πλατφόρμες λειτουργούν, αλλά και τις διαφορές τους. Ιδιαίτερα με τη βοήθεια του Ethereum, και ορισμένων επιλεγμένων εργαλείων που βασίζονται σε αυτό, σας δίνεται η δυνατότητα να πειραματιστείτε με πιο πρακτικά θέματα, στον βαθμό που ο/η καθένας/μία νιώθει άνετα.

Επιπλέον, στο βιβλίο αναφέρονται αρκετές υλοποιήσεις και εργαλεία, άλλες πιο αναλυτικά, άλλες πιο επιφανειακά. Σκοπός είναι να γίνουν γνωστές σε εσάς, έτσι ώστε να γνωρίζετε πού θα αναζητήσετε περισσότερες πληροφορίες όταν αντιμετωπίσετε ένα θέμα που καλύπτεται από αυτές.

Οι αναφορές στο τέλος του κάθε κεφαλαίου περιέχουν πολλούς συνδέσμους στο Διαδίκτυο, καθώς πολλή γνώση προς το παρόν βρίσκεται διασκορπισμένη σε αυτό. Παράλληλα, θα βρείτε και ορισμένα συγγράμματα και δημοσιεύσεις, ορισμένα από τα οποία (θα το αντιληφθείτε από τη συχνότητα επανάληψής τους) αποτελούν αναφορά για όλους όσοι ασχολούνται με την τεχνολογία αυτή.

Κλείνοντας, η συγγραφή του βιβλίου στα ελληνικά είχε τις δικές της προκλήσεις. Η πιο σημαντική ήταν η μετάφραση των εννοιών, καθώς, πολλές φορές, δεν υπήρχε κάποιος όρος που να έχει προκριθεί στα κείμενα στη γλώσσα μας. Στην περίπτωση αυτή εμφανίζεται η επικρατέστερη μετάφραση (σε μία περίπτωση γίνεται αναφορά και σε περισσότερες μεταφράσεις που βρέθηκαν) για αναφορά, αν και οι αγγλικοί όροι παραμένουν, για τη δυνατότητα προσαρμογής στη διεθνή βιβλιογραφία.

Ελπίζουμε με το σύγγραμμα αυτό να πετύχουμε τους σκοπούς μας όπως αναφέρθηκαν, αλλά και να συμβάλουμε στην ανάπτυξη νέων εκπαιδευτικών μαθημάτων με βάση την τεχνολογία του blockchain.

Η συγγραφική ομάδα

ΚΕΦΑΛΑΙΟ 1

Χαρακτηριστικά της Τεχνολογίας της Αλυσίδας Συστοιχιών (Blockchain)

Σύνοψη

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση των βασικών χαρακτηριστικών της τεχνολογίας blockchain, καθώς και των επιμέρους μεθόδων και τεχνικών που συνδυάστηκαν μοναδικά για να δημιουργήσουν την τεχνολογία αυτή. Ειδικότερα, αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα που παρουσιάζει μια λύση βασισμένη σε blockchain, περιγράφονται οι βασικές κατηγορίες δικτύων blockchain που συναντώνται σήμερα και απαντιέται μία σειρά ερωτημάτων που βοηθούν να αποφασιστεί ή όχι η εφαρμογή της τεχνολογίας αυτής (σε συγκεκριμένη κάθε φορά περίπτωση).

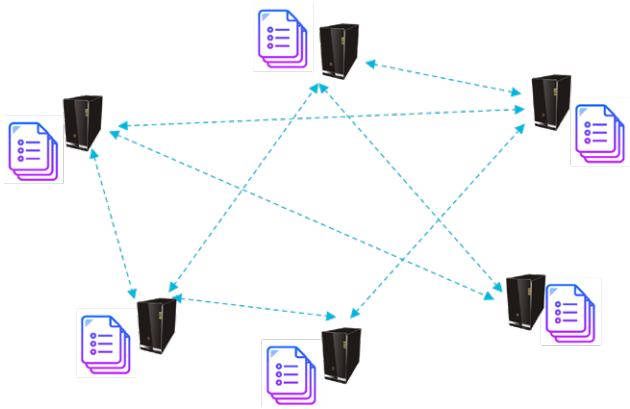
Προαπαιτούμενη γνώση

Δεν απαιτείται προαπαιτούμενη γνώση.

1.1 Περιγραφή και βασικά χαρακτηριστικά

Η τεχνολογία της *Αλυσίδας Συστοιχιών (blockchain)* είναι μια τεχνολογία που παρουσιάστηκε για πρώτη φορά το 2008 στη γνωστή δημοσίευση του Satoshi Nakamoto (2008). Αξίζει να σημειωθεί ότι η πραγματική ταυτότητα του συγγραφέα παραμένει άγνωστη μέχρι σήμερα.

Στη δημοσίευση αυτή περιγράφεται η αρχική και πιο γνωστή εφαρμογή της τεχνολογίας αυτής: ένα σύστημα ηλεκτρονικών συναλλαγών που χρησιμοποιεί ένα καινούργιο ψηφιακό νόμισμα, το bitcoin. Αυτός είναι και ο λόγος που η έννοια του blockchain συνδέθηκε αμέσως στο ευρύ κοινό με το συγκεκριμένο κρυπτονόμισμα, ειδικά τα πρώτα χρόνια του, παρόλο που ως όρος (η λέξη blockchain) δεν εμφανίζεται καθόλου στην εν λόγω δημοσίευση. Αντιθέτως, οι δύο λέξεις: *συστοιχία* (*block*) και *αλυσίδα* (*chain*) εμφανίζονται ξεχωριστά, με την πάροδο του χρόνου όμως ενώθηκαν σε μία (blockchain). Έτσι σχηματίζεται το όνομα της τεχνολογίας που υποστηρίζει την εφαρμογή του Bitcoin.



Εικόνα 1.1 *Μια απεικόνιση ενός κατανεμημένου συστήματος κοινού εδαφίου για ηλεκτρονικές πληρωμές.*

Πέρα όμως από την παρουσίαση του ψηφιακού νομίσματος, η εν λόγω δημοσίευση παρουσίαζε και τον βασικό σκοπό της πρώτης εμφάνισης της τεχνολογίας του blockchain, που αφορά τον σχεδιασμό και τη λειτουργία ενός ολοκληρωμένου ψηφιακού συστήματος ηλεκτρονικών πληρωμών (**Εικόνα 1.1**). Ένα τέτοιο σύστημα αποτελείται από:

- Ένα *κατανεμημένο δίκτυο ομότιμων κόμβων* (*Peer-to-Peer, P2P*) στο οποίο όλοι οι συμμετέχοντες αλληλεπιδρούν απευθείας ο ένας με τον άλλο, χωρίς τη διαμεσολάβηση κάποιας ενδιάμεσης έμπιστης

(ή και μη) οντότητας. Αυτό, σε αντίθεση, για παράδειγμα, με τα κλασικά συστήματα οικονομικών συναλλαγών, όπως αυτά των τραπεζικών, όπου η παρουσία αυτής της έμπιστης οντότητας είναι κυρίαρχη. Σκοπός είναι να επιτευχθούν μικρότεροι χρόνοι στην ολοκλήρωση των απευθείας συναλλαγών.

- Ένα *εδάφιο* (*ledger*) στο οποίο και καταγράφονται όλες οι συναλλαγές μεταξύ των χρηστών. Το εδάφιο αυτό είναι ουσιαστικά κοινό, καθώς αντιγράφεται σε όλους τους χρήστες και ενημερώνεται συνεχώς για να εξασφαλιστεί η δυσκολία παράκαμψης ή αλλοίωσής του. Έχει τη μορφή μιας αλυσίδας από συστοιχίες (blocks) που περιέχουν τις συναλλαγές των χρηστών.
- Από *χρήστες* που συναλλάσσονται απευθείας μεταξύ τους με τη βοήθεια ειδικών διεπαφών για την αποθήκευση των κλειδιών τους, γνωστών ως πορτοφολιών.

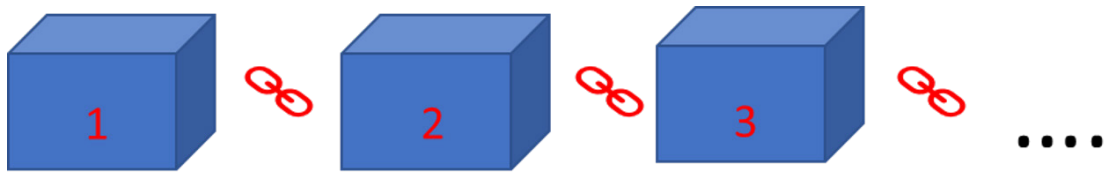
Επιπλέον, το σύστημα χαρακτηρίζεται από:

- *Χρήση μοναδικών διευθύνσεων* για αποστολή και λήψη συναλλαγών. Για να γίνει δυνατή η αναγνώριση και η επικοινωνία μέσα στο δίκτυο, κάθε χρήστης αποκτά ένα *μοναδικό ιδιωτικό κλειδί* (μεγέθους 256 bits). Το κλειδί αυτό καθορίζει μονοσήμαντα μια διεύθυνση στο δίκτυο που χρησιμοποιεί ο χρήστης κατά την επικοινωνία του μέσα σε αυτό. Δεν είναι δυνατόν να έχει ένας άλλος χρήστης το ίδιο ιδιωτικό κλειδί και την ίδια διεύθυνση, όπως δεν είναι δυνατόν να προκύψουν διαφορετικές διευθύνσεις από ένα συγκεκριμένο ιδιωτικό κλειδί.
- *Μονιμότητα και αμεταβλητότητα των δεδομένων*, καθώς υπάρχει η δυνατότητα μόνο προσθήκης νέων δεδομένων (συναλλαγών) στο κοινό ledger, χωρίς τη δυνατότητα αφαίρεσής τους από αυτό. Δηλαδή, όταν μια συναλλαγή εγκριθεί, προστεθεί σε ένα block και γραφτεί στο κοινό ledger, δεν δύναται πλέον να αφαιρεθεί. Αυτό εξασφαλίζεται με χρήση τεχνικών κρυπτογράφησης. Οι τεχνικές αυτές εφαρμόζονται τόσο στον τρόπο δημιουργίας του κάθε block συναλλαγών όσο και στη σύνδεση μεταξύ των blocks για τη δημιουργία της αλυσίδας που αποτελεί και το κοινό εδάφιο.
- *Χρονολογική διάκριση των συναλλαγών*, οι οποίες για να γραφούν στο εδάφιο μαζεύονται σε συστοιχίες (blocks) και ενώνονται κρυπτογραφικά σε μια αλυσίδα. Κάθε καινούργια ομάδα δεδομένων θα εισάγεται σε μια νέα συστοιχία, χωρίς δυνατότητα μελλοντικής αφαίρεσής τους ούτε από τη συστοιχία αυτή ούτε, κατ' επέκταση, και από την αλυσίδα. Η σειριακή αυτή εισαγωγή των δεδομένων επιτρέπει τη χρονική τους διάκριση, καθώς κάθε νέα συστοιχία προστίθεται/συνδέεται στις υπάρχουσες (δηλαδή στην εκάστοτε νεότερη).
- *Χρήση ψηφιακών υπογραφών για την απόδειξη της αυθεντικότητας των συναλλαγών*. Οι ψηφιακές υπογραφές εξασφαλίζουν ότι οι συναλλαγές έχουν δημιουργηθεί από τον πραγματικό ιδιοκτήτη του μεταφερόμενου ποσού. Με τον τρόπο αυτόν οι ψηφιακές υπογραφές συνεισφέρουν στη δημιουργία μιας σχέσης εμπιστοσύνης προς το σύστημα. Για την υλοποίησή τους χρησιμοποιείται το ιδιωτικό κλειδί που αποκτά ο χρήστης για να δημιουργήσει τη διεύθυνσή του στο δίκτυο.
- *Σύνθετες τεχνικές συναίνεσης* (*consensus*) που επιτρέπουν σε κάθε κόμβο στο κατανεμημένο σύστημα να λαμβάνει αποφάσεις που ακολουθούν τους κανόνες για την ορθή λειτουργία του συστήματος. Ανάλογα με το είδος της υλοποίησης μιας λύσης blockchain, είναι πιθανό να δοθούν από το σύστημα τα κατάλληλα κίνητρα για την εξασφάλιση της σωστής υλοποίησης της συναίνεσης στο κατανεμημένο δίκτυο. Δημοφιλείς τεχνικές αποτελούν η *Απόδειξη Εργασίας* (*Proof of Work, PoW*) καθώς και η *Απόδειξη Συμμετοχής*[1] (*Proof of Stake, PoS*). Λεπτομέρειες για αυτά θα βρείτε στο Κεφάλαιο 5.

Συνοπτικά, μπορεί να ειπωθεί ότι η τεχνολογία του blockchain διαχειρίζεται δεδομένα που δημιουργούνται από απευθείας συναλλαγές μεταξύ των χρηστών του δικτύου. Τα δεδομένα συγκεντρώνονται σε ομάδες και οργανώνονται σε συστοιχίες (blocks), οι οποίες συνδέονται με μοναδικό τρόπο μέσω ισχυρής κρυπτογράφησης. Η μοναδική αυτή σύνδεση των blocks μεταξύ τους δημιουργεί μία αλυσίδα (chain) από blocks, όπως φαίνεται και στην **Εικόνα 1.2**. Τα περιεχόμενα της αλυσίδας αποθηκεύονται στο (κοινό) ledger και διαμοιράζονται σε όλους τους συμμετέχοντες στο δίκτυο. Επιπλέον, η συγκεκριμένη δομή και χρήση της αλυσίδας σε ένα δίκτυο blockchain επιτρέπει σε αυτό να δώσει έμφαση στην ασφάλεια, στην αποθήκευση και στη διαχείριση των

[1] Συχνά συναντάται στα ελληνικά και ως Απόδειξη Πονταρίσματος. Στο βιβλίο θα συναντήσουμε τον αγγλικό όρο.

δεδομένων στις συστοιχίες αυτές, εφαρμόζοντας λύσεις που κάνουν χρήση κρυπτογραφίας και ψηφιακών υπογραφών.



Εικόνα 1.2 *Μια Αλυσίδα Συστοιχιών (blockchain) της οποίας η σύνδεση μεταξύ των blocks με τις συναλλαγές επιτυγχάνεται με ισχυρή κρυπτογραφία.*

Στη συνέχεια αναλύονται τα ιδιαίτερα χαρακτηριστικά της τεχνολογίας blockchain και γίνεται αναφορά στα πλεονεκτήματα και τα μειονεκτήματα της (Ενότητα 1.2). Ακολουθεί (Ενότητα 1.3) μια ανασκόπηση της σειράς των γεγονότων που οδήγησαν στην εμφάνιση της τεχνολογίας blockchain καθώς και των βημάτων που ακολούθησαν την αρχική της παρουσίαση. Επιπρόσθετα, γίνεται μια ανάλυση των γνωστών τύπων δικτύων blockchain (Ενότητα 1.4) που έχουν δημιουργηθεί. Αυτό θα επιτρέψει στον κάθε ενδιαφερόμενο να αποφασίσει για τα ιδιαίτερα χαρακτηριστικά μιας δικής του λύσης που θα βασίζεται στην τεχνολογία blockchain. Στα επόμενα κεφάλαια αναδεικνύονται ευρύτερες περιπτώσεις χρήσης της τεχνολογίας ώστε να γίνει κατανοητός ο τρόπος που μπορεί αυτή να ενσωματωθεί και να επιδράσει σε σύγχρονες εφαρμογές, όπως είναι, για παράδειγμα, η διαχείριση της *Εφοδιαστικής Αλυσίδας (ΕΑ)* ή η *αποκεντρωμένη ψηφιακή ταυτοποίηση* (*Decentralized Identities, DIDs*).

Ταυτόχρονα, θα γίνει δυνατή η αναγνώριση εκείνων των περιπτώσεων χρήσης στις οποίες η εφαρμογή μιας λύσης blockchain θα προσφέρει σημαντική βελτίωση της απόδοσης του συστήματος και, επομένως, θα αξίζει να εξεταστεί πραγματικά στο σύνολό της (π.χ. μελετώντας και οικονομικά κριτήρια πέρα από τεχνικά). Για τον λόγο αυτό στην Ενότητα 1.5 παρουσιάζονται συγκεκριμένα κριτήρια, τα οποία και θα πρέπει να ληφθούν υπόψιν κατά τη διαδικασία λήψης μιας απόφασης ενσωμάτωσης μιας λύσης βασισμένης στο blockchain.

Τα κριτήρια αυτά ελέγχου για την καταλληλότητα ενσωμάτωσης μιας λύσης βασισμένης στο blockchain αποτελούν τη βάση αναφοράς στα επόμενα κεφάλαια του βιβλίου, όταν συζητούνται προτάσεις για τον σχεδιασμό λύσεων.

1.2 Πλεονεκτήματα και μειονεκτήματα του blockchain

Η εμφάνιση του κατανεμημένου συστήματος ηλεκτρονικών πληρωμών με τη χρήση του κρυπτονομίσματος bitcoin ανοίγει τον δρόμο στη διερεύνηση των πλεονεκτημάτων και μειονεκτημάτων της τεχνολογίας blockchain.

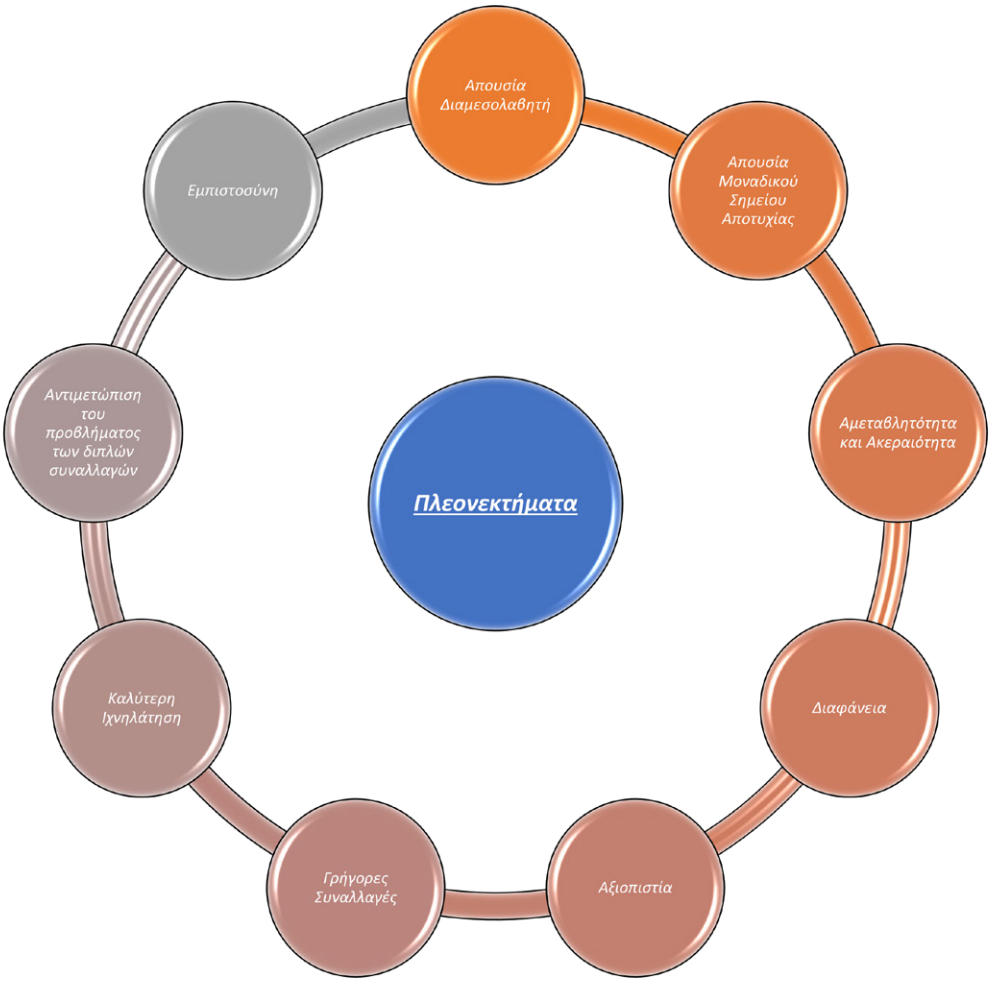
Η γνωριμία των πλεονεκτημάτων οδηγεί στην αξιολόγηση, στην αποδοχή και στην επιλογή εφαρμογής της τεχνολογίας. Από την άλλη πλευρά, η γνωριμία των μειονεκτημάτων οδηγεί σε προσπάθειες για την αντιμετώπισή τους, στην αποδοχή λύσεων που ήδη έχουν προταθεί αλλά και στην πρόταση νέων.

Πλεονεκτήματα:

- *Απουσία έμπιστης οντότητας* (ή *ενδιάμεσου*): Ένα δίκτυο blockchain είναι εκ φύσεως κατανεμημένο, με τους χρήστες να συνδιαλέγονται απευθείας μεταξύ τους, χωρίς την παρουσία μιας τρίτης (έμπιστης) οντότητας που θα παρακολουθεί και θα εγγυάται την εύρυθμη λειτουργία. Ταυτόχρονα, επίσης, για τις υπηρεσίες αυτές που προσφέρει, ο ενδιάμεσος θα κρατήσει κάποια αμοιβή αυξάνοντας το κόστος της συναλλαγής για τον χρήστη. Με την απουσία του ενδιάμεσου σε ένα δίκτυο blockchain εξαφανίζεται η προμήθεια αυτή, ενώ ο έλεγχος για την ομαλή λειτουργία του δικτύου μεταφέρεται στους χρήστες, οι οποίοι αναλαμβάνουν να τον αντικαταστήσουν. Επιπλέον, οι χρήστες λειτουργούν προς όφελος του δικτύου είτε για να λάβουν ένα (οικονομικό) αντάλλαγμα για τη συμμετοχή τους είτε γιατί έχουν επιλέξει ότι αυτό είναι προς το συμφέρον τους (ομοίως, θα έχει γίνει επιλογή και των συμμετεχόντων στην περίπτωση αυτή, όπως αναφέρεται αναλυτικά και στην Ενότητα 1.4).
- *Απουσία μοναδικού σημείου αποτυχίας*: Λόγω της κατανεμημένης φύσεώς τους, τα δίκτυα blockchain επιτυγχάνουν να συνεχίσουν την εύρυθμη λειτουργία τους ακόμα και όταν δημιουργηθούν προβλήματα (π.χ. τεχνικά θέματα ή επιθέσεις) σε έναν αριθμό κόμβων του δικτύου. Επιπρόσθετα, λόγω των

εφαρμοζόμενων τεχνικών συναίνεσης, είναι δυνατή η εύρυθμη λειτουργία του δικτύου ακόμα και όταν ένας αριθμός χρηστών (μικρότερος του 50% των συμμετεχόντων) λειτουργεί κακόβουλα.

- *Αμεταβλητότητα και ακεραιότητα*: Η χρήση κρυπτογραφίας στην αποθήκευση των συναλλαγών σε ένα block, καθώς και στη σύνδεση μεταξύ δύο διαδοχικών blocks στην αλυσίδα ενός blockchain, ενισχύει τη δυσκολία στη διαγραφή πληροφορίας από την αλυσίδα και, ως αποτέλεσμα αυτού, από το κοινό ledger που διαμοιράζεται στο δίκτυο. Με αυτόν τον τρόπο εξασφαλίζεται η αμεταβλητότητα του ledger και η ακεραιότητα των δεδομένων που φυλάσσονται σε αυτό και αποθαρρύνονται οι προσπάθειες εξαπάτησης των συμμετεχόντων.
- *Διαφάνεια*: Οι συναλλαγές σε ένα δίκτυο blockchain είναι εμφανείς σε όλους τους συμμετέχοντες σε αυτό (τουλάχιστον στα δημόσια δίκτυα, βλ. περισσότερα στην Ενότητα 1.4). Αυτό συμβαίνει για να είναι εφικτή η επιβεβαίωση της εγκυρότητάς τους κατά τη διαδικασία της συναίνεσης. Έτσι, ενώ μπορεί ο οποιοσδήποτε συμμετέχει στο δίκτυο να παρακολουθήσει τις συναλλαγές που αντιστοιχούν σε μια διεύθυνση του δικτύου (ή αλλιώς σε κάποιο πορτοφόλι αυτού), δεν είναι δυνατή η αναγνώριση του φυσικού προσώπου στον οποίο αντιστοιχεί η διεύθυνση αυτή, καθώς δεν παρέχεται τέτοιου είδους πληροφορία στο δίκτυο. Με τον τρόπο αυτόν υπάρχει πλήρης διαφάνεια των συναλλαγών και ένα είδος (ψευδο-)ανωνυμίας με τη χρήση διευθύνσεων που δεν μπορούν να αντιστοιχηθούν σε φυσικά πρόσωπα.
- *Αξιοπιστία*: Οι κανόνες που διέπουν μια λύση blockchain δεν δύνανται να αλλαχθούν από μια οντότητα σε οποιοδήποτε χρονικό διάστημα. Αντιθέτως, οι όποιες αλλαγές ξεκινούν από προτάσεις που ψηφίζονται από το σύνολο των συμμετεχόντων και οι αποφάσεις λαμβάνονται βάσει πλειοψηφίας. Η διαδικασία αυτή προσδίδει στο σύστημα μια αξιοπιστία, που αυξάνεται όσο ο αριθμός των χρηστών μεγαλώνει. Όσο περισσότεροι χρήστες συμμετέχουν στη διαδικασία των αποφάσεων, τόσο πιο δύσκολο θα είναι για κάποιον να χειραγωγήσει τα αποτελέσματα.
- *Γρήγορες συναλλαγές*: Καθώς οι συναλλαγές ολοκληρώνονται με την απευθείας συμμετοχή των εμπλεκόμενων χρηστών, η διάρκειά τους είναι σύντομη. Η όποια καθυστέρηση εισάγεται από το δίκτυο και είναι κομμάτι της συνεννόησης των κόμβων (διαδικασία γνωστή και ως *συναίνεση*) για τη συμφωνία της ορθής επιλογής του επόμενου block της αλυσίδας στο ledger αλλά και της ενημέρωσης όλων των συμμετεχόντων στην επιλογή αυτή.
- *Καλύτερη ιχνηλάτηση*: Η διαφάνεια στις συναλλαγές, που διακρίνει το σύστημα, σε συνδυασμό με την ακεραιότητα και αμεταβλητότητα, επιτρέπει την πλήρη ιχνηλάτηση μιας συναλλαγής μέσα στο ledger.
- *Αντιμετώπιση του προβλήματος των διπλών συναλλαγών*: Το πρόβλημα των διπλών συναλλαγών είναι πολύ σημαντικό για ένα σύστημα ηλεκτρονικών πληρωμών. Αφορά τις συναλλαγές εκείνες στις οποίες οι χρήστες αξιοποιούν την ψηφιακή μορφή του συστήματος και προσπαθούν να χρησιμοποιήσουν το ίδιο ποσό από χρήματα σε δύο διαφορετικές αγοραπωλησίες/συναλλαγές. Ενώ με τη χρήση του φυσικού χρήματος μπορούμε να είμαστε σίγουροι ότι ένα νόμισμα θα χρησιμοποιηθεί σε μία και μόνη συναλλαγή μας, στον ψηφιακό κόσμο θα πρέπει να αποκλειστεί η περίπτωση που κάποιος θα προσπαθήσει να χρησιμοποιήσει τα ίδια χρήματα σε δύο διαφορετικές συναλλαγές.
  Στο blockchain κάτι τέτοιο αντιμετωπίζεται με την επιβεβαίωση όλων των συναλλαγών που ανήκουν σε ένα block, προτού αυτό προστεθεί στην αλυσίδα. Έτσι, το υπόλοιπο ποσό κάθε διεύθυνσης ανανεώνεται αμέσως, και όταν έρθει η νέα συναλλαγή θα επιδράσει στο ανανεωμένο ποσό. Αν και οι δύο συναλλαγές προστεθούν στο ίδιο block, τότε, εφόσον το υπόλοιπο δεν επαρκεί για να εξυπηρετηθούν και οι δύο, μόνο η μία θα περάσει και η άλλη θα ακυρωθεί.
- *Εμπιστοσύνη*: Καθώς ένα δίκτυο blockchain επιδεικνύει αμεταβλητότητα, ακεραιότητα, διαφάνεια και ιχνηλάτηση, τότε καταφέρνει να κερδίσει την εμπιστοσύνη των χρηστών που συμμετέχουν σε αυτό. Αυτοί γνωρίζουν ότι, αν κάποιος δεν δράσει νόμιμα, τότε (αργά ή γρήγορα) το δίκτυο θα τον απομονώσει ή θα τον αναγκάσει να συμμορφωθεί με τους κανόνες του.



Εικόνα 1.3 *Πλεονεκτήματα της τεχνολογίας blockchain.*

Μειονεκτήματα:

- *Επεκτασιμότητα*: Αποτελεί ένα από τα πιο σημαντικά μειονεκτήματα της τεχνολογίας, καθώς όσο το δίκτυο αυξάνει σε μέγεθος, τόσο πιο χρονοβόρο είναι για έναν καινούργιο κόμβο να κατεβάσει ολόκληρο το ledger για να μπορέσει να συμμετέχει ενεργά στη διαδικασία της συναίνεσης. Υπολογίζεται ότι για το δίκτυο του Bitcoin το μέγεθος του ledger είναι πάνω από 250 GB, αυξάνοντας έτσι τις απαιτήσεις σε υλικό (hardware) που πρέπει να έχει ένας νέος κόμβος για να μπει στο δίκτυο. Για την αντιμετώπιση του προβλήματος έχουν προταθεί διάφοροι τύποι blockchain. Περισσότερες λεπτομέρειες στην Ενότητα 1.4.
- *Ιδιωτικά κλειδιά*: Όπως αναφέρθηκε προηγουμένως, η εφαρμογή λύσεων ασύμμετρης κρυπτογραφίας βασίζεται στη χρήση ιδιωτικών/δημόσιων κλειδιών για τον κάθε χρήστη του δικτύου, καθώς μέσω μιας συγκεκριμένης διαδικασίας (βλ. Κεφάλαιο 3) οδηγούν στην παραγωγή της διεύθυνσής του στο δίκτυο. Έτσι, αν και η διεύθυνση κάποιου μπορεί να είναι γνωστή σε όλους, το ιδιωτικό του κλειδί πρέπει να παραμείνει κρυφό, καθώς χρησιμοποιείται και για την υπογραφή των συναλλαγών ως απόδειξη ιδιοκτησίας των μεταφερόμενων ποσών. Αυτό έχει ως αποτέλεσμα, αν χαθούν (ή κλαπούν) τα ιδιωτικά κλειδιά ενός χρήστη, αυτός πλέον *να μην μπορεί* να έχει πρόσβαση στο περιεχόμενο του λογαριασμού του (πορτοφόλι). Χρειάζεται, λοιπόν, να γίνει συνήθεια η διατήρηση και φύλαξη των κλειδιών και η προστασία τους από σύγχρονους ψηφιακούς κινδύνους.
- *Ταχύτητα*: Αν και η ολοκλήρωση των συναλλαγών είναι πολύ πιο γρήγορη με την απουσία ενδιάμεσων, η επεξεργασία τους από το σύστημα ενδέχεται να εμφανίζει κάποια καθυστέρηση. Συγκεκριμένα, ο αριθμός των *συναλλαγών ανά δευτερόλεπτο* (*transactions per second, tps*) που μπορεί να διαχειριστεί ένα από τα βασικά δίκτυα blockchain είναι περίπου 7 tps (Bitcoin) ή 30 tps (Ethereum). Αν αυτό

συγκριθεί με τα σύγχρονα συστήματα των πιστωτικών καρτών, παρατηρείται ότι υστερεί σημαντικά (με τον αριθμό αυτό να κυμαίνεται σε μερικές χιλιάδες το δευτερόλεπτο στην περίπτωσή τους). Αν βέβαια συγκριθεί με τον χρόνο μεταφοράς χρημάτων σε παγκόσμιο επίπεδο μεταξύ τραπεζικών συστημάτων διαφορετικών χωρών, τότε δεν είναι τόσο προβληματικός. Τελικά, όμως, αν ένα ηλεκτρονικό σύστημα πληρωμών θέλει να είναι ανταγωνιστικό, θα πρέπει να βελτιώσει την απόδοσή του στο θέμα της ταχύτητας. Προς αυτή την κατεύθυνση, τελευταία έχουν προταθεί λύσεις που βασίζονται σε άλλες εφαρμογές της *Τεχνολογίας Κατανεμημένων Εδαφίων* (*Distributed Ledger Technologies, DLTs*) οι οποίες επιτυγχάνουν την επιθυμητή βελτίωση (βλ. Κεφάλαιο 10).

- *Αμφιβολίες ιδιωτικότητας*: Δεδομένης της διαφάνειας που χαρακτηρίζει τις υλοποιήσεις δικτύων blockchain και παρά την (ψευδο-)ανωνυμία των συμμετεχόντων, για πολλούς χρήστες η ποσότητα της πληροφορίας που είναι προσβάσιμη σε τρίτους είναι περισσότερη από όση θα επιθυμούσαν. Αυτό έχει ως συνέπεια να δημιουργούνται αμφιβολίες ως προς το επίπεδο της ιδιωτικότητας που περιλαμβάνεται στις υλοποιήσεις αυτές. Αν σε αυτό περιληφθεί και η αδυναμία διαγραφής πληροφοριών, τότε γίνεται αντιληπτό ότι οι λύσεις blockchain αντιμετωπίζουν ολοένα και περισσότερα θέματα σε εφαρμογές όπου διαχειρίζονται ευαίσθητα δεδομένα.
  Ιδιαίτερα, από την έναρξη της εφαρμογής του Γενικού Κανονισμού για την Προστασία των Δεδομένων (GDPR) στην Ευρωπαϊκή Ένωση τα θέματα αυτά έχουν οδηγήσει σε υλοποιήσεις όπου πλέον δεν αποθηκεύονται τα ίδια τα δεδομένα στο δίκτυο blockchain αλλά η τοποθεσία τους, οπότε με τη χρήση λύσεων κρυπτογραφίας μπορεί να επιβεβαιωθεί αν κάτι έχει αλλάξει σε αυτά ή όχι (Ault, 2018).
- *Μεγάλο κόστος*: Γνωρίζοντας τη σημασία του μεγάλου αριθμού από κόμβους σε ένα δίκτυο blockchain, η δημιουργία αυτής της υποδομής μπορεί να αποτελέσει ένα ακριβό εγχείρημα, ανάλογα με την περίπτωση. Βεβαίως, υπάρχουν λύσεις που προσφέρονται ακόμα και στη μορφή Blockchain as a Service[2] από εταιρείες νεφοϋπολογιστικής, αλλά και πάλι το κόστος μπορεί να είναι σημαντικό. Επιπλέον, η χρήση του διαδεδομένου αλγόριθμου συναίνεσης PoW έχει συντελέσει στην κατανάλωση ποσών ενέργειας από τους εμπλεκόμενους στη διαδικασία, κάτι που αυξάνει ακόμα περισσότερο το κόστος διατήρησης και συμμετοχής στο δίκτυο.
  Για την αντιμετώπιση του θέματος αυτού υπήρξε μελέτη που έχει οδηγήσει σε εφαρμογή νέων τεχνικών συναίνεσης (π.χ. PoS), οι οποίες αντιμετωπίζουν ακριβώς το χαρακτηριστικό αυτό χωρίς να κάνουν εκπτώσεις στην ευρεία αποτελεσματικότητα του PoW.
- *Αποδοχή*: Όπως συμβαίνει και με κάθε καινούργιο χαρακτηριστικό, έτσι και με την τεχνολογία του blockchain η αποδοχή του από το ευρύ κοινό έρχεται πολύ αργά. Ιδιαίτερα και λόγω του πρωτοποριακού τρόπου λειτουργίας του, που έρχεται σε αντίθεση με το σύνολο των συνηθειών μας (π.χ. κεντρικοποιημένες λύσεις με χρήση διακομιστών) σε σημαντικούς τομείς της ζωής μας (π.χ. οικονομικός), απαιτείται χρόνος για να βρεθεί ο τρόπος με τον οποίο θα μπορέσει να παίξει μεγαλύτερο ρόλο στην καθημερινότητα όλων μας. Επιπλέον, η εξοικείωση των ανθρώπων με τα βασικά χαρακτηριστικά της ψηφιακής ασφάλειας θα βοηθήσει σημαντικά στην αποδοχή και στην ευρύτερη χρήση της τεχνολογίας του blockchain.

[2] Online Σύνδεσμος: https://www.investopedia.com/terms/b/blockchainasaservice-baas.asp

Εικόνα 1.4 *Μειονεκτήματα της τεχνολογίας blockchain.*

1.3 Συστατικά της τεχνολογίας του blockchain – Ιστορία

Η τεχνολογία blockchain αποτέλεσε μια ενδιαφέρουσα πρόταση σε θέματα που έχουν επίπτωση στις οικονομικές συναλλαγές των ανθρώπων. Με την εμφάνιση και τη δημοτικότητα του πρώτου ευρείας χρήσης ψηφιακού κρυπτονομίσματος (bitcoin) έγινε μια ανάλυση του συνόλου των μεθόδων και τεχνικών που συνδυάζονται με μοναδικό τρόπο στην εφαρμογή του Bitcoin. Βέβαια, επιμέρους χαρακτηριστικά της εφαρμογής, όπως η προσπάθεια δημιουργίας ενός συστήματος ηλεκτρονικών πληρωμών, ήταν ήδη γνωστά στην επιστημονική κοινότητα.

Στη συνέχεια αναλύονται τεχνικές που έγιναν γνωστές είτε ως συστατικά του blockchain είτε ως βάση του μετά από κατάλληλες προσαρμογές. Η αναφορά θα ακολουθήσει τη σειρά χρονολογικής εμφάνισής τους και θα τονίζει τα επιτεύγματα που έγιναν δυνατά με τη χρήση τους. Πληρέστερη ανάλυση θα είναι αντικείμενο επόμενων κεφαλαίων.

1.3.1 Περίοδος 1979-2007: Η προεργασία

Όπως αναφέρθηκε, αρκετά από τα στοιχεία που χαρακτηρίζουν ένα δίκτυο blockchain εμφανίστηκαν πολύ νωρίτερα, ωρίμασαν και τελικά συνδυάστηκαν και χρησιμοποιήθηκαν στην παρουσίαση του κατανεμημένου συστήματος ηλεκτρονικών πληρωμών το 2008. Αυτά προέρχονταν κυρίως από το πεδίο της κρυπτογραφίας (π.χ. αλυσίδα συστοιχιών, συναλλαγές, κλειδιά και ψηφιακές υπογραφές κ.ά.) και από το πεδίο της κατανεμημένης υπολογιστικής (π.χ. δίκτυα ομότιμων κόμβων, τεχνικές συγχρονισμού κατανεμημένου δικτύου, τεχνικές συναίνεσης κ.ά.).

Ένα από αυτά τα συστατικά είναι τα *δένδρα Merkle* (*Merkle Trees*). Αναφέρθηκαν για πρώτη φορά το 1979 στη διδακτορική διατριβή του Ralph Merkle (στο Πανεπιστήμιο Stanford) και έγιναν ευρύτερα γνωστά με τη σχετική δημοσίευση (Merkle, 1989). Αποτέλεσαν ένα εργαλείο διαμοιρασμού δημόσιων κλειδιών και

παραγωγής ψηφιακών υπογραφών. Ο Merkle, μάλιστα, προστάτευσε με πατέντα τη σχετική πρότασή του. Τα δένδρα Merkle περιγράφουν, επιπλέον, και μια δομή δεδομένων η οποία επιτρέπει την επιβεβαίωση εγγραφών, επιτρέποντας με τον τρόπο αυτόν τη συλλογή τους σε μια συστοιχία. Στο blockchain χρησιμοποιήθηκαν κυρίως για να προσθέσουν τη δυνατότητα συμμετοχής περισσότερων της μιας συναλλαγών σε ένα block.

Κατόπιν, το 1982 ο David Chaum περιγράφει στη δική του διδακτορική διατριβή στο Πανεπιστήμιο της Καλιφόρνια (Berkeley) ένα σύστημα που περιέχει πάρα πολλά παρόμοια χαρακτηριστικά με ένα σύστημα blockchain. Πιο συγκεκριμένα, περιγράφεται η λειτουργία ενός κατανεμημένου υπολογιστικού συστήματος το οποίο λειτουργεί και συντηρείται έχοντας κερδίσει την εμπιστοσύνη ενός συνόλου από αμοιβαία δύσπιστους χρήστες. Από την περιγραφή του αυτή απουσιάζει η χρήση ενός αλγόριθμου συναίνεσης, όπως συμβαίνει στη δημοσίευση του Bitcoin, το οποίο περιγράφει έναν τέτοιο μηχανισμό, το γνωστό πλέον PoW. Σήμερα θεωρείται ως η πρώτη απόπειρα για την περιγραφή ενός πρωτοκόλλου blockchain, όπως φαίνεται και στο Sherman et al. (2019), εφόσον κέρδισε την προσοχή και, τελικά, την αναγνώριση της ερευνητικής κοινότητας. Ο Chaum, βεβαίως, παρέμεινε ενεργός στον χώρο της κρυπτογραφίας, ιδρύοντας την εταιρεία DigiCash, πρωτοπόρος στον προσδιορισμό και χρήση ηλεκτρονικού χρήματος, προσφέροντας ανωνυμία με υλοποίηση ειδικών πρωτοκόλλων κρυπτογραφίας που είχε δημιουργήσει ο ίδιος (η εταιρεία, τελικά, κήρυξε πτώχευση το 1998).

Το 1991 είναι χρονιά ορόσημο. Ο Stuart Haber μαζί με τον W. Scott Stornetta συνέλαβαν και περιέγραψαν το πρώτο σύστημα κρυπτογραφικά συνδεδεμένης αλυσίδας συστοιχιών, πρόδρομο του σημερινού συστήματος blockchain (Habert & Stornetta, 1991). Το πέτυχαν στην προσπάθειά τους να προτείνουν μια μέθοδο στην οποία κάθε έγγραφο που θα εισέρχεται στο σύστημα θα συνοδεύεται από τη δική του *χρονοσφραγίδα* (*timestamp*), η οποία και δεν θα μπορούσε να αλλοιωθεί από έναν κακόβουλο χρήστη. Επεκτείνοντας τη δουλειά τους, το 1992 με τη συνδρομή του Dave Bayer προσθέτουν στο σύστημά τους και τη χρήση των δένδρων Merkle, επιτρέποντας με τον τρόπο αυτόν τη συλλογή περισσότερων εγγραφών μέσα σε ένα block, βελτιώνοντας την αποτελεσματικότητα του συστήματος (Bayer et al., 1992).

Το 1997 ο Adam Back παρουσίασε έναν αλγόριθμο με το όνομα *hashcash*, που βασίζεται σε *κρυπτογραφικό κατακερματισμό* (*hash*[3]) για να περιορίσει την αποστολή ανεπιθύμητων μηνυμάτων ηλεκτρονικής αλληλογραφίας (spam), καθώς και ως μέτρο άμυνας απέναντι σε Επιθέσεις Άρνησης Υπηρεσίας (Denial of Service) (Back, 1997). Η λογική του hashcash βασιζόταν στην απαίτηση για προσφορά μεγάλης υπολογιστικής ισχύος για την παραγωγή ενός αποτελέσματος το οποίο όμως, με τη σειρά του, θα μπορούσε εύκολα να επιβεβαιωθεί. Στην περίπτωση των ηλεκτρονικών μηνυμάτων, αυτή η υπολογιστική ισχύς απαιτείται για την παραγωγή μιας σφραγίδας που έμπαινε στην κεφαλίδα του μηνύματος ως απόδειξης ότι ολοκληρώθηκε η διαδικασία αυτή. Με τον τρόπο αυτόν η καθυστέρηση που εισήγαγε η τεχνική αυτή λειτουργούσε αποτρεπτικά σε πιθανούς spammers που επιδιώκουν την αποστολή μεγάλου αριθμού μηνυμάτων σε πολύ μικρό χρόνο. Η ιδέα αυτή είχε προταθεί πρώτα το 1992 (Dwork & Naor, 1992) και ο Back βασίστηκε σε αυτή για την πρόταση της τεχνικής του. Στο blockchain ο αλγόριθμος αυτός αποτέλεσε κομμάτι της τεχνικής συναίνεσης στο δίκτυο και έγινε γνωστός με το όνομα Proof of Work (PoW).

Ταυτόχρονα, το 1999 σημειώνεται η μεγάλη επιτυχία του Napster, η οποία φέρνει στην πρώτη γραμμή της επικαιρότητας και της προσοχής τα *Δίκτυα Ομότιμων Κόμβων* (*Peer-to-Peer, P2P*), που θα αποτελέσουν σημαντικό κομμάτι της τεχνολογίας blockchain.

Τέλος, το 2004 ο Hal Finney, ένας από τους πρωτοπόρους του Bitcoin, παρουσίασε ένα πρωτότυπο σύστημα για ψηφιακό χρήμα, που λεγόταν *Επαναχρησιμοποιούμενη Βεβαίωση Εργασίας* (*Reusable Proof of Work, RPoW*). Το σύστημα αυτό δημιουργεί *μια μάρκα* (*token*) με τη μορφή μιας *συμβολοσειράς* (*string*) που έχει υπογραφεί από το ιδιωτικό κλειδί του κατόχου και το οποίο καταχωρίζεται σε έναν (κεντρικό) διακομιστή μαζί με όλα τα υπόλοιπα διακριτικά των άλλων χρηστών. Στη συνέχεια, το token αυτό μπορεί να μεταφερθεί σε άλλον χρήστη (βάσει του δημόσιου κλειδιού αυτού) σε μια συναλλαγή που θα φέρει την υπογραφή του αποστολέα.

Το σύστημα αυτό κατόρθωνε να αντιμετωπίσει το σημαντικότερο πρόβλημα στα ηλεκτρονικά συστήματα, δηλαδή αυτό του εντοπισμού και αποτροπής των διπλών συναλλαγών (γνωστό και ως double spending), δίνοντας τη δυνατότητα στους χρήστες για πρόσβαση σε πραγματικό χρόνο στον διακομιστή με τις εγγραφές των tokens, με σκοπό την επιβεβαίωση ή/και διόρθωση της ιδιοκτησίας αυτών.

Ο Finney, όντας το άτομο που έλαβε bitcoins στην πρώτη συναλλαγή που καταγράφηκε στο δημοφιλές

[3] *Συναρτήσεις κατακερματισμού* (*hash functions*) ονομάζονται οι μαθηματικές εκείνες συναρτήσεις οι οποίες λαμβάνουν μια είσοδο οποιουδήποτε μεγέθους και παράγουν μια κρυπτογραφημένη έξοδο σταθερού μήκους. Το αποτέλεσμα στην έξοδο ονομάζεται και «hash». Ο όρος αυτός θα χρησιμοποιηθεί με αυτή την έννοια στη συνέχεια του βιβλίου.

αυτό δίκτυο (2009), είναι ένα πρόσωπο που θεωρήθηκε ότι βρίσκεται πίσω από το ψευδώνυμο του Satoshi Nakamoto, αν και ο ίδιος έχει απορρίψει τους ισχυρισμούς αυτούς, που συνέχισαν να υπάρχουν και μετά τον θάνατό του, το 2014.

1.3.2 2008-2013: Η πρώτη περίοδος του blockchain (Blockchain 1.0)

Η περίοδος αυτή έχει ως σημείο εκκίνησης τη δημοσίευση του Nakamoto, που περιέχει την περιγραφή του κατανεμημένου συστήματος ηλεκτρονικών πληρωμών που παρουσιάστηκε προηγουμένως. Για τον λόγο αυτόν η περίοδος αυτή είναι επικεντρωμένη στις οικονομικές υπηρεσίες που εισήγαγε η τεχνολογία του blockchain.

Έτσι, το 2009 καταγράφεται η πρώτη συναλλαγή στο δίκτυο του Bitcoin από τον δημιουργό του, προχωρώντας και στην υλοποίηση της ιδέας του με χρήση λογισμικού ανοικτού κώδικα. Το Bitcoin, με αυτόν τον τρόπο, επέτρεψε σε όλο τον κόσμο να έχει πρόσβαση στον κώδικα του και, βέβαια, να είναι ελεύθερος να αποτελέσει μέλος της πρώτης υλοποίησης ανοικτού δικτύου blockchain παγκοσμίως.

Δύο σημαντικές ημερομηνίες για το δίκτυο του Bitcoin και για την περίοδο αυτή είναι:

- η 12η Ιανουαρίου 2009, όταν και καταγράφεται στο block 170 η πρώτη συναλλαγή κρυπτονομισμάτων, με τη μεταφορά 10 bitcoins από τον λογαριασμό (ή αλλιώς το πορτοφόλι) του Satoshi Nakamoto προς τον λογαριασμό του Hal Finney.
- η 22α Μαΐου του 2010, όπου καταγράφεται η πρώτη συναλλαγή για αγορά φυσικού προϊόντος και η πληρωμή του σε bitcoins. Έτσι, ο Laszlo Hanyecz αγόρασε 2 κουτιά πίτσα αξίας 25$ πληρώνοντας για αυτά με 10.000 bitcoins και, αυτομάτως, μετατρέποντάς τη στην πιο ακριβή πίτσα στην ιστορία, δεδομένου ότι η αξία του bitcoin για το 2021 ήταν σταθερά πάνω από τα 35.000$ το ένα.

Τα ίχνη του δημιουργού του Bitcoin χάνονται προς τα τέλη του 2010, ύστερα από μια αναβάθμιση στο λογισμικό του ψηφιακού νομίσματος για την αντιμετώπιση ενός σφάλματος που εντοπίστηκε στο block 74.638, το οποίο επέτρεπε τη δημιουργία ενός απροσδόκητα μεγάλου αριθμού από bitcoins από το πουθενά.

Παρ' όλα αυτά, το σύστημά του και η τεχνολογία blockchain αρχίζουν και εδραιώνονται και κερδίζουν σε συμμετοχή αλλά και σε απόδοση, με την τιμή του κρυπτονομίσματος να ανεβαίνει αργά αλλά με σταθερά βήματα. Νέα κρυπτονομίσματα (π.χ. Namecoin, NameID) προσπαθούν να μιμηθούν την πορεία αυτή και εμφανίζονται στο προσκήνιο, προσθέτοντας νέα χαρακτηριστικά και βελτιώσεις σε αυτά που παρείχε το Bitcoin.

Η περίοδος αυτή όμως δεν ανήκει μόνο στα κρυπτονομίσματα. Οι αλγόριθμοι συναίνεσης σε κατανεμημένα περιβάλλοντα, όπως είναι το *Proof of Work* (*PoW*) ή, εναλλακτικά, το *Proof of Stake* (*PoS*) κερδίζουν την προσοχή, όπως και η δυνατότητα χρονοσήμανσης των συναλλαγών σε αυτά. Μάλιστα, αρχίζει και η επισήμανση των πρώτων αδυναμιών της τεχνολογίας, με την απαιτούμενη υπολογιστική ισχύ και τον μεγάλο χρόνο απόκρισης να αναφέρονται συχνά. Το ενδιαφέρον του κόσμου ολοένα και αυξάνεται, και έτσι αρχίζουν και αναφέρονται τα κενά που υπάρχουν στους τρόπους διαλειτουργικότητας των εφαρμογών της τεχνολογίας blockchain. Χρησιμοποιούνται, δε, από τους πολέμιους αυτής ως βασικά επιχειρήματα για τη μείωση της ευρύτερης αποδοχής και χρήσης της τεχνολογίας.

1.3.3 2013-2016: Η δεύτερη περίοδος του blockchain (Blockchain 2.0)

Η περίοδος αυτή χαρακτηρίζεται από την εμφάνιση του δικτύου του Ethereum, το οποίο ο δημιουργός του Vitalik Buterin περιέγραψε το 2013-14 (Buterin, 2013· Buterin, 2014) και υλοποίησε το 2015. Η περιγραφή (και η υλοποίηση) του δικτύου του Ethereum είναι πολύ σημαντική για την τεχνολογία του blockchain, καθώς αποτέλεσε και την πρώτη προσπάθεια να αναδειχθούν οι σημαντικές δυνατότητες της τεχνολογίας, πέρα από τη χρήση της σε ένα σύστημα ηλεκτρονικών πληρωμών.

Με το Ethereum χρησιμοποιείται ένα ζωντανό δίκτυο blockchain για να αποθηκευτούν προγράμματα χρηστών, γνωστά και ως *έξυπνες συμβάσεις* (*smart contracts*), τα οποία βασίζονται στην αμεταβλητότητα των δεδομένων που προσφέρει η τεχνολογία για να εξασφαλίσουν ότι δεν έχει γίνει καμία αλλαγή στον κώδικά τους. Επιπλέον, τα προγράμματα αυτά χρησιμοποιούνται για να παρακινήσουν συναλλαγές όταν ικανοποιούνται επιβεβαιωμένα (μέσα από το blockchain πάλι ή από έμπιστες εξωτερικές πηγές, γνωστές ως oracles[4,5]) οι

[4] Online Σύνδεσμος: https://ethereum.org/en/developers/docs/oracles/

[5] Online Σύνδεσμος: https://chain.link/education/blockchain-oracles

συνθήκες που περιγράφονται στις συμβάσεις αυτές.

Δεδομένης της δυνατότητας ανάπτυξης smart contracts σε υλοποιήσεις blockchain, είναι πλέον σημαντικό να υπάρχει τρόπος επικοινωνίας και κλήσης αυτών από τους χρήστες, ανάλογα με την περίπτωση χρήσης. Για να γίνει εφικτό αυτό, άρχισαν να παρουσιάζονται οι πρώτες *Αποκεντρωμένες Εφαρμογές* (*Decentralized Applications, DApps*). Αυτές έδιναν τη δυνατότητα σε χρήστες να αξιοποιήσουν τις νέες δυνατότητες του blockchain, αλλά και άνοιγαν το πεδίο για χρήση της τεχνολογίας σε νέους τομείς. Τα DApps δεν χρησιμοποιούσαν στη βάση τους κεντρικούς servers, αλλά ένα δίκτυο blockchain. Αυτό τους επέτρεπε να χρησιμοποιούν κατανεμημένες λύσεις για αποθήκευση και επικοινωνία, ενώ μια εφαρμογή χρήστη (συνήθως μια web εφαρμογή την περίοδο εκείνη) επέτρεπε να γίνονται οι απαραίτητες κλήσεις και συναλλαγές στο δίκτυο.

Με τη δυνατότητα δημιουργίας DApps και την αξιοποίηση των smart contracts ξεκίνησε μια συνεχής αναζήτηση για την εύρεση λύσεων που βασίζονται στην τεχνολογία του blockchain σε πολλούς τομείς της καθημερινής μας ζωής, όπως: στην απόδοση ψηφιακών εκπαιδευτικών τίτλων, στη διαχείριση ιατρικών δεδομένων, στην παρακολούθηση και διαχείριση δεδομένων της εφοδιαστικής αλυσίδας και πολλά άλλα. Αξίζει να αναφερθεί ότι, πέρα από την προσπάθεια ενσωμάτωσης της τεχνολογίας του blockchain σε υπάρχοντες τομείς χρήσης, έγινε και μια προσπάθεια να δημιουργηθούν εφαρμογές οι οποίες αξιοποιούν τις καινούργιες δυνατότητες που εισάγει η τεχνολογία του blockchain, όπως, για παράδειγμα, η δυνατότητα δημιουργίας ψηφιακών συλλεκτικών αντικειμένων (γνωστά ως crypto collectibles) τα οποία μπορούν να διαφέρουν μεταξύ τους και, επιπλέον, μπορούν να ανταλλαχθούν (ή να πωληθούν) από τους συμμετέχοντες στο δίκτυο.

Ένα τέτοιο παράδειγμα είναι και τα CryptoKitties[6]. Αυτά αποτέλεσαν ένα πρωτοποριακό DApp που λειτούργησε ως παράδειγμα για τη μελλοντική ανάπτυξη των Non-Fungible Tokens (NFT, βλ. Κεφάλαιο 7). Μάλιστα, το DApp των CryptoKitties αποτέλεσε ένα παιχνίδι το οποίο συγκέντρωσε έναν πολύ μεγάλο αριθμό από χρήστες, με αποτέλεσμα τον Δεκέμβριο του 2017 να δημιουργηθεί συμφόρηση στο δίκτυο του Ethereum, καθώς οι συναλλαγές του παιχνιδιού αποτέλεσαν πάνω από το 70% των συναλλαγών όλου του δικτύου.

1.3.4 2017-σήμερα: Η τρίτη περίοδος του blockchain (Blockchain 3.0)

Με την αυξανόμενη επιτυχία των smart contracts και την ανάπτυξη και χρήση σημαντικού αριθμού από DApps διαμορφώνεται η τάση για διεύρυνση του τρόπου χρήσης λύσεων που βασίζονται στο blockchain. Από την άλλη πλευρά, η ολοένα και μεγαλύτερη απασχόληση του κοινού με την τεχνολογία είχε ως αποτέλεσμα να επισημανθούν τα μειονεκτήματά της και, ως επακόλουθο, να υπάρξουν συζητήσεις και προτάσεις επίλυσης αυτών.

Στο επίκεντρο των συζητήσεων για τα μειονεκτήματα του blockchain έχουν μπει θέματα σχετικά με:

- την *επεκτασιμότητα* των δικτύων: δεδομένων των συνεχώς αυξανόμενων απαιτήσεων σε επεξεργαστική ισχύ και σε αποθηκευτικό χώρο, αρχίζουν και δημιουργούνται σημαντικά προβλήματα για τον αριθμό των νέων κόμβων που θα μπορούν να καλύψουν τις απαιτήσεις·
- τη *διαλειτουργικότητα* μεταξύ δύο διαφορετικών λύσεων blockchain: η αδυναμία επικοινωνίας μεταξύ των δύο βασικών δικτύων blockchain (Bitcoin, Ethereum) καθώς και η δημιουργία νέων πλατφορμών έχουν κάνει αναγκαία τη δυνατότητα επικοινωνίας και ανταλλαγής δεδομένων μεταξύ αυτών, χωρίς όμως να υπάρχει κάποιο πρότυπο για να βασιστούν αυτές·
- την *αντοχή* των λύσεων blockchain: με τις περισσότερες λύσεις να βασίζονται σε χρήματα που προέρχονται από ενδιαφερόμενους που θέλουν να χρηματοδοτήσουν μια προσπάθεια, η συνέχεια και η μακροζωία αυτών θα εξαρτηθεί από τον τρόπο που θα φροντίσουν να εξασφαλίσουν την κάλυψη των απαραίτητων εξόδων τους. Ιδανικά, θα αναζητηθούν αποκεντρικοποιημένες λύσεις που θα λειτουργούν αυτόνομα με γνώμονα το καλό του δικτύου.

Την περίοδο αυτή έχουν αρχίσει να κάνουν την εμφάνισή τους και νέες υλοποιήσεις δικτύων (πέρα από το Ethereum και το Bitcoin) οι οποίες δανείζονται από αυτά τα θετικά τους και προβάλλουν τις δικές τους προτάσεις για την αντιμετώπιση των μειονεκτημάτων, εισάγοντας τις δικές τους ιδέες στον χώρο του blockchain. Για παράδειγμα, η χρήση λύσεων *Κατανεμημένων Εδαφίων* (*Distributed Ledgers*) με τη βοήθεια *Άκυκλων Κατευθυντικών Γράφων* (*Directed Acycle Graphs, DAGs*) για την παρακολούθηση των συναλλαγών (σε αντίθεση με την αλυσίδα κατάστιχων) μελετάται ως εναλλακτική για τη χρήση σε περιπτώσεις όπου τα

[6] Online Σύνδεσμος: https://www.cryptokitties.co/

δεδομένα είναι μεγάλα σε αριθμό και συχνότητα, όπως συμβαίνει στην περίπτωση της δημιουργίας εφαρμογών για το *Διαδίκτυο των Πραγμάτων* (*ΔτΠ*).

Οι λύσεις που βασίζονται στους DAGs προσφέρουν μια καλύτερη επεκτασιμότητα και επιτυγχάνουν καλύτερες τιμές στον αριθμό των συναλλαγών ανά δευτερόλεπτο (transactions per second, tps). Η παρουσίαση μίας από τις πιο γνωστές λύσεις (IOTA) θα γίνει στο Κεφάλαιο 10, το οποίο και εστιάζει στην παρουσίαση ευρύτερων λύσεων που ανήκουν στην κατηγορία αυτή.

Με βάση την εμπειρία από την ως τώρα εφαρμογή και στην προσπάθεια αναγνώρισης και διόρθωσης των μειονεκτημάτων, το Ethereum ανακοινώνει τα σχέδιά του για αλλαγή στον τρόπο συναίνεσης στο δίκτυο. Το Ethereum 2.0, όπως ονομάζεται η καινούργια έκδοσή του, αναμένεται να έχει αντιμετωπίσει τα μειονεκτήματα που αναγνωρίστηκαν ενσωματώνοντας σύγχρονες λύσεις (π.χ. συναίνεση μέσω PoS, χρήση τεχνικών καταμερισμού του δικτύου με σκοπό τον καλύτερο έλεγχό του). Η νέα αυτή έκδοση προβλέπεται να ολοκληρωθεί το 2023.

1.4 Οι διαφορετικοί τύποι του blockchain

Για να δημιουργηθεί ένα δίκτυο blockchain, απαιτείται να ληφθούν κάποιες αποφάσεις που θα καθορίζουν τον τρόπο λειτουργίας του και τα χαρακτηριστικά του. Ανεξάρτητα όμως με τις αποφάσεις αυτές, όλοι οι τύποι του blockchain που θα παρουσιαστούν εδώ εμπεριέχουν τα βασικά χαρακτηριστικά της τεχνολογίας:

- χρησιμοποιούν τεχνικές συναίνεσης για την επαλήθευση των συναλλαγών και την προσθήκη των νέων blocks και
- αποθηκεύουν στους κόμβους το κοινό εδάφιο.

Έτσι, από τις πιο σημαντικές αποφάσεις είναι αυτή που αφορά τον τύπο του δικτύου blockchain, δηλαδή μια απόφαση που σχετίζεται με το ποιος έχει πρόσβαση σε αυτό. Αρχικά, τα πρώτα δημοφιλή δίκτυα blockchain (Bitcoin, Ethereum) ήταν δημόσια, επιτρέποντας στον καθένα να αποκτήσει μια διεύθυνση και να συνδεθεί στο δίκτυο παίζοντας ενεργό ρόλο σε αυτό. Όμως, με την αύξηση των εφαρμογών, προέκυψαν σενάρια στα οποία η συνεργασία των εμπλεκομένων αφορούσε λίγα μέρη, συχνά εταιρείες, με αποτέλεσμα να δημιουργηθούν λύσεις ιδιωτικές, με έλεγχο όσων έχουν πρόσβαση στο δίκτυο καθώς και των ενεργειών τους μέσα σε αυτό. Στη συνέχεια προέκυψαν και άλλοι τύποι δικτύων που επέλεξαν έναν συνδυασμό χαρακτηριστικών από τους δύο κύριους τύπους.

Επιπλέον, σημαντική επιλογή είναι και αυτή που αφορά το τι επιτρέπεται να κάνει ο κάθε χρήστης μέσα σε κάθε δίκτυο blockchain. Για παράδειγμα, αν θα μπορεί να συμμετέχει ενεργά στη διαδικασία της συναίνεσης και στη δημιουργία των νέων blocks ή θα μπορεί απλώς να εκτελεί συναλλαγές και να έχει πρόσβαση μόνο στις δικές του. Έτσι, δημιουργούνται πάλι δύο βασικές (υπο)κατηγορίες δικτύων: αυτά της ελεύθερης συμμετοχής στις λειτουργίες του δικτύου (γνωστά και ως *χωρίς άδεια δίκτυα* ή *permissionless*) και αυτά τα οποία χρειάζονται την *απόδοση άδειας* για την ενεργή συμμετοχή ενός χρήστη σε αυτά (γνωστά και ως *permissioned*).

Στη συνέχεια αναλύονται οι γνωστοί τύποι λειτουργίας ενός δικτύου blockchain με τα χαρακτηριστικά τους. Επιπλέον, γίνεται μια σύγκριση των πλεονεκτημάτων και των μειονεκτημάτων αυτών μαζί με αναφορές στις πιο γνωστές υλοποιήσεις της κάθε κατηγορίας.

1.4.1 Δημόσια δίκτυα blockchain

Τα *ανοικτά* ή *δημόσια δίκτυα blockchain* είναι τα δίκτυα εκείνα τα οποία είναι ελεύθερα σε όλους. Ο κάθε ενδιαφερόμενος μπορεί να συνδεθεί και –στις περισσότερες περιπτώσεις– να αποκτήσει αντίγραφο του κοινού εδάφιου (ledger), να κάνει συναλλαγές, να συμμετέχει στη διαδικασία επιβεβαίωσης, αλλά και στη δημιουργία blocks. Επομένως, πρόκειται για δίκτυα ελεύθερης συμμετοχής που δεν απαιτούν εν γένει κάποια άδεια για την εγγραφή συναλλαγών στο εδάφιο με την εύρεση του επόμενου block. Παρ' όλα αυτά, όσον αφορά την ελευθερία ενεργειών, υπάρχουν και περιπτώσεις όπου –παρά την ελεύθερη πρόσβαση– σε κάποιους κόμβους αποδίδονται περισσότερα δικαιώματα σχετικά με τη συμμετοχή στις επιβεβαιώσεις και στην αποθήκευση ολόκληρου του ledger.

Πρόκειται για πλήρως κατανεμημένα δίκτυα στα οποία οι συμμετέχοντες κόμβοι αναλαμβάνουν τη δύσκολη εργασία της συλλογής των συναλλαγών και της δημιουργίας των blocks. Ταυτόχρονα, ο αριθμός των κόμβων του συστήματος είναι πολύ σημαντικός, καθώς όσο πιο πολλοί είναι αυτοί τόσο πιο ισχυρό είναι το σύστημα

και τόσο πιο δύσκολο είναι για κάποιον εξωτερικό χρήστη να καταφέρει να το ελέγξει. Παράλληλα, όμως, ο μεγάλος αριθμός κόμβων εισάγει και περισσότερη καθυστέρηση στη λήψη αποφάσεων (π.χ. μικρό tps).

Για τη συναίνεση στα δίκτυα αυτά χρησιμοποιούνται τεχνικές όπως η *Απόδειξη Εργασίας* (*PoW*) ή η *Απόδειξη Συμμετοχής* (*PoS*). Η πρώτη αποτελεί δοκιμασμένη και επιτυχημένη λύση, αλλά απαιτεί σημαντική υπολογιστική ισχύ για την εύρεση του block. Η δε δεύτερη έρχεται ως ένα μέτρο βελτίωσης της πρώτης στον παράγοντα της ενεργειακής κατανάλωσης, η οποία γενικά θεωρείται σημαντικό μειονέκτημα.

Τα δημόσια blockchains είναι ιδιαίτερα δημοφιλή, καθώς σε αυτά συμπεριλαμβάνονται οι πρώτες και πιο γνωστές και διαδεδομένες υλοποιήσεις, όπως είναι τα δίκτυα *Bitcoin*[7] και *Ethereum*[8]. Ταυτόχρονα, όμως, υπάρχουν και δημόσια δίκτυα blockchain, τα οποία, ενώ επιτρέπουν την πρόσβαση σε όλους, δημιουργούν οντότητες (κόμβους / χρήστες) οι οποίες έχουν άλλα δικαιώματα σε σχέση με τους άλλους. Τέτοια δημόσια δίκτυα είναι τα *EOS*[9] και το *Ripple*[10].

Στα *πλεονεκτήματα* των δημόσιων δικτύων blockchain περιλαμβάνονται:

- Η ελεύθερη πρόσβαση στο δίκτυο, η συμμετοχή στη διαδικασία συναίνεσης και η δημιουργία των νέων blocks.
- Η ανάπτυξη του αισθήματος της εμπιστοσύνης στο δίκτυο και στη λειτουργία αυτού από τους χρήστες.
- Αυτή η αύξηση της εμπιστοσύνης στο δίκτυο έρχεται με την απόδοση των κατάλληλων κινήτρων, που θα συντελέσουν έτσι ώστε το συμφέρον του κάθε χρήστη να συμβαδίζει με το καλό του δικτύου. Τα κίνητρα αυτά παίρνουν συνήθως τη μορφή της απόδοσης κάποιας (οικονομικής) ανταμοιβής για τη συμμετοχή του χρήστη στο δίκτυο. Έτσι, οι χρήστες αναμένεται να λειτουργούν για το καλό του δικτύου παρά αποκλειστικά για το προσωπικό τους συμφέρον.
- Η απουσία του οποιουδήποτε ενδιάμεσου παράγοντα στη λειτουργία του δικτύου, προσφέροντας μια πραγματικά κατανεμημένη μορφή συστήματος.
- Η αυξημένη ασφάλεια που παρέχεται από τον (συνήθως) μεγάλο αριθμό χρηστών, λόγω και της ελεύθερης πρόσβασης σε αυτό. Ο μεγάλος αριθμός συμμετοχών δυσκολεύει την αποτελεσματική επίθεση στο δίκτυο, καθώς συνήθως οι αποφάσεις λαμβάνονται βάσει πλειοψηφίας των κόμβων. Είναι γνωστό ότι οι τεχνικές συναίνεσης που εφαρμόζονται στις λύσεις δημόσιων blockchain είναι ικανές να εγγυηθούν τη σωστή απόφαση για την εγκυρότητα ενός block (βάσει του αριθμού των κόμβων που λένε την αλήθεια), λύνοντας το γνωστό Πρόβλημα των Στρατηγών του Βυζαντίου (βλ. Κεφάλαιο 5).
- Στην πράξη αυτό σημαίνει ότι ο αριθμός των κόμβων που χρειάζεται να ελέγχει κάποιος για να επιτεθεί αποτελεσματικά σε ένα δίκτυο Blockchain μπορεί να είναι μικρότερος από το 50% (Lamport, 1982). Η πιθανότητα αυτή μειώνεται όσο η συμμετοχή στα δίκτυα αυτά είναι ευρεία. Για αυτόν τον λόγο, ο αριθμός των κόμβων του δικτύου αποτελεί μια κρίσιμη παράμετρο.
- Η διαφάνεια στις συναλλαγές, καθώς όλοι οι συμμετέχοντες μπορούν να έχουν πρόσβαση σε αυτές μέσω του αντίγραφου του ledger που έχουν στη διάθεσή τους. Οι συναλλαγές στο δίκτυο αυτό δεν είναι κρυπτογραφημένες, όμως οι εμπλεκόμενοι δεν εμφανίζονται με τα πραγματικά ονόματά τους αλλά μόνο μέσω των διευθύνσεών τους στο δίκτυο (ψευδο-ανωνυμία). Με τον τρόπο αυτόν προσφέρεται πλήρης διαφάνεια στο δίκτυο, καθώς είναι δυνατόν να εντοπιστούν όλες οι συναλλαγές μιας διεύθυνσης, ακόμα και αν δεν είναι γνωστός ο πραγματικός κάτοχος αυτής.

Στα *μειονεκτήματα* των δημόσιων δικτύων blockchain περιλαμβάνονται:

- Ο μικρός αριθμός συναλλαγών ανά δευτερόλεπτο (tps) που παρατηρούνται. Για παράδειγμα, έχει μετρηθεί ότι στο δίκτυο του Bitcoin έχουμε τη δημιουργία ενός καινούργιου block κάθε 10 λεπτά, πράγμα που μεταφράζεται σε έναν αριθμό επεξεργασίας μόλις 7 συναλλαγών το δευτερόλεπτο (tps). Αυτός είναι ένας αριθμός ο οποίος δεν μπορεί να συγκριθεί με τα συμβατικά συστήματα ηλεκτρονικών καρτών τα οποία είναι σε θέση να επεξεργάζονται πολλές χιλιάδες συναλλαγές το δευτερόλεπτο.

---

[7] Online Σύνδεσμος: https://bitcoin.org/en/

[8] Online Σύνδεσμος: https://ethereum.org/en/

[9] Online Σύνδεσμος: https://eos.io/

[10] Online Σύνδεσμος: https://ripple.com/

- Η καταγραφή αυτού του πλήθους των συναλλαγών οφείλεται και στις τεχνικές συναίνεσης που χρησιμοποιούνται (ενδεικτικά αναφέρονται τα PoW, PoS). Το PoW έχει επικριθεί και για το σημαντικό ποσό ενέργειας που καταναλώνεται από τους κόμβους στην προσπάθειά τους να δημιουργήσουν το επόμενο block στην αλυσίδα του blockchain.
- Η αδυναμία εύκολης επέκτασης των δικτύων αυτών. Η προσθήκη νέων χρηστών γίνεται ολοένα και πιο δύσκολη καθώς το δίκτυο και η αλυσίδα μεγαλώνουν. Για παράδειγμα, ένας νέος χρήστης στο δίκτυο του Bitcoin θα πρέπει, προτού καταφέρει να συμμετάσχει ενεργά σε αυτό, να κατεβάσει όλο το εδάφιο, το οποίο έχει μέγεθος εκατοντάδες GBs.

Η γενική αίσθηση είναι ότι τα δημόσια δίκτυα blockchain εμπεριέχουν όλα εκείνα τα χαρακτηριστικά που κάνουν την τεχνολογία αυτή να ξεχωρίζει (π.χ. κατανεμημένη δομή, ελεύθερη πρόσβαση, απουσία τρίτων ενδιάμεσων μερών). Περιπτώσεις όπου επιλέγεται η χρήση ενός τέτοιου τύπου blockchain είναι: οικονομικές εφαρμογές, έρανοι συλλογής χρημάτων, καθώς και εφαρμογές για τη διαχείριση και την εκτέλεση εκλογών ή ψηφοφοριών, όπου η διαφάνεια και η εμπιστοσύνη των συμμετεχόντων είναι κρίσιμες.

1.4.2 Ιδιωτικά δίκτυα blockchain

Με την επιλογή της τεχνολογίας blockchain σε όλο και περισσότερες περιπτώσεις εφαρμογών έγινε αντιληπτή η ανάγκη να υπάρχει έλεγχος στην πρόσβαση των δεδομένων, ανάλογα και με τον ρόλο του κάθε χρήστη. Την ανάγκη αυτή εξυπηρετούν τα ιδιωτικά δίκτυα blockchain.

Τα δίκτυα αυτά έχουν ως κύριο χαρακτηριστικό ότι δημιουργούνται και ελέγχονται από μία οντότητα. Η λειτουργία τους μπορεί να λαμβάνει χώρα σε ένα πιο περιορισμένο περιβάλλον και είναι αναγκαία η άδεια από την οντότητα αυτή για τη συμμετοχή στο δίκτυο ενός χρήστη. Δεν μπορεί ο οποιοσδήποτε να αποτελέσει μέρος αυτού και να αποκτήσει πρόσβαση στο κοινό ledger.

Στο πλαίσιο αυτό υπάρχει, συχνά, και ένα στάδιο αυθεντικοποίησης των χρηστών προτού εισέλθουν στο δίκτυο. Έτσι, η εμπιστοσύνη μεταφέρεται από το δίκτυο (δημόσια blockchain) στον χρήστη για την ορθή του δράση μέσα σε αυτό.

Το χαρακτηριστικό αυτό κάνει τα ιδιωτικά δίκτυα blockchain να είναι, συνήθως, πιο μικρά σε μέγεθος από τα δημόσια και, λόγω του ελέγχου από μία οντότητα, να εμφανίζουν μεγάλο ποσοστό κεντρικοποίησης. Πέρα από αυτό, όμως, η λειτουργία του δικτύου είναι παρόμοια και περιέχει τα γνωστά χαρακτηριστικά της τεχνολογίας blockchain, όπως: διαφάνεια, ασφάλεια συναλλαγών και εμπιστοσύνη στο δίκτυο.

Η βέλτιστη χρήση του τύπου αυτού δικτύου είναι εσωτερικά σε ένα εταιρικό περιβάλλον. Γνωστές υλοποιήσεις που βασίζονται στη δημιουργία και χρήση ιδιωτικών δικτύων αποτελούν το *Hyperledger Fabric*[11] (που υποστηρίζεται και από το Linux Foundation) και το *Corda*[12].

Στα *πλεονεκτήματα* των ιδιωτικών δικτύων περιλαμβάνονται:

- *Η μεγαλύτερη ταχύτητά τους*: Καθώς ο αριθμός των κόμβων είναι πιο μικρός, είναι πιο εύκολο για το δίκτυο να έρθει σε συναίνεση και να δημιουργήσει το επόμενο block.
- *Η μεγαλύτερη δυνατότητα επέκτασης*: Λόγω του μικρού μεγέθους που έχουν αυτού του είδους τα δίκτυα, διευκολύνεται ένας καινούργιος κόμβος να συγχρονιστεί γρήγορα με τους υπόλοιπους κατεβάζοντας το κοινό ledger και, κατόπιν, να ξεκινήσει να παίζει ενεργό ρόλο στις λειτουργίες του δικτύου.
- *Η μικρότερη ενεργειακή κατανάλωση*: Άλλοι αλγόριθμοι συναίνεσης που χρησιμοποιούνται είναι μεν πιο αποδοτικοί σε δίκτυα τέτοιου μεγέθους, αλλά είναι πιο επιρρεπείς σε λάθη σε μια επίθεση από κακόβουλους χρήστες. Ο έλεγχος για την είσοδο στο δίκτυο επιτρέπει στα δίκτυα αυτά να μετατοπίσουν την εμπιστοσύνη στον χρήστη και να εφαρμόσουν αλγόριθμους συναίνεσης που είναι πιο γρήγοροι και αποδοτικοί (π.χ. practical Byzantine Fault Tolerance, pBFT).

Στα *μειονεκτήματα* των ιδιωτικών δικτύων αναφέρονται:

- *Η έλλειψη γνήσιας κατανεμημένης υλοποίησης*: Το δίκτυο χαρακτηρίζεται από ισχυρή κεντρικοποίηση λόγω της μίας οντότητας που το διαχειρίζεται.

[11] Online Σύνδεσμος: https://www.hyperledger.org/use/fabric
[12] Online Σύνδεσμος: https://www.corda.net/

- *Η έλλειψη εμπιστοσύνης στο δίκτυο*: Προέρχεται από την ύπαρξη του ελέγχου από την κεντρική οντότητα.
- *Κίνδυνοι για την ασφάλεια που προέρχονται από τον μικρό αριθμό συμμετεχόντων*: Κάνει ευάλωτο το δίκτυο σε επιθέσεις από τρίτους, εφόσον αυτοί καταφέρουν να αποκτήσουν πρόσβαση σε αυτό.

Περιπτώσεις χρήσης στις οποίες έχει μελετηθεί ο σχεδιασμός και η εφαρμογή λύσεων που βασίζονται σε ιδιωτικές λύσεις δικτύων blockchain περιλαμβάνουν: τη διαχείριση της εφοδιαστικής αλυσίδας (ένας τομέας ο οποίος επηρεάστηκε έντονα από τις επιπτώσεις της πανδημίας της Covid-19, εκθέτοντας πολλές από τις αδυναμίες του), τους τίτλοι ιδιοκτησίας περιουσιακών στοιχείων, στους οποίους το blockchain προσφέρει διαφάνεια και ευκολία επιβεβαίωσής τους.

Πέρα, όμως, από τους δύο παραπάνω τύπους, οι οποίοι και είναι οι βασικοί τύποι blockchain, δημιουργήθηκαν στη συνέχεια και άλλοι, οι οποίοι είναι κυρίως υβριδικοί, καθώς δανείζονται στοιχεία από τους προαναφερθέντες και δημιουργούν ιδιαίτερους συνδυασμούς που εφαρμόζονται σε διαφορετικές περιπτώσεις κάθε φορά. Αυτοί οι (υβριδικοί) τύποι blockchain θα παρουσιαστούν στη συνέχεια.

1.4.3 Δίκτυα κοινοπραξίας blockchain

Τα δίκτυα κοινοπραξίας περιέχουν χαρακτηριστικά και των δύο βασικών τύπων δικτύων blockchain. Δηλαδή, κάποια από τα δεδομένα στις συναλλαγές είναι δημόσια, ενώ κάποια άλλα είναι ιδιωτικά. Πρόκειται για δίκτυα τα οποία απαιτούν άδεια για να συνδεθεί κάποιος χρήστης σε αυτά και να έχει πρόσβαση στις λειτουργίες τους. Παρ' όλα αυτά, η ύπαρξη κοινοπραξίας οργανισμών που αναλαμβάνει να διαχειριστεί το δίκτυο, σε αντίθεση με τον έλεγχο από έναν οργανισμό όπως συμβαίνει στα ιδιωτικά δίκτυα, αυξάνει αρκετά την κατανεμημένη φύση του δικτύου αυτού, ενισχύοντας την αποκέντρωσή του.

Στα δίκτυα αυτά ο ρόλος που θα έχει ο κάθε συμμετέχων καθορίζεται από την κοινοπραξία οργανισμών που έχει αναλάβει τη διαχείριση του δικτύου, καθώς είναι συγκεκριμένοι οι κόμβοι που λαμβάνουν μέρος στη διαδικασία της συναίνεσης. Έτσι, υπάρχουν κόμβοι οι οποίοι έχουν εκτελεστικό ρόλο, καθώς μπορούν να εκκινούν ή να λαμβάνουν συναλλαγές αλλά και να συμμετέχουν στη διαδικασία της συναίνεσης για το επόμενο block στην αλυσίδα. Ταυτόχρονα υπάρχουν και απλοί κόμβοι-μέλη οι οποίοι επιτρέπεται μόνο να δημιουργούν και να δέχονται συναλλαγές, χωρίς να έχουν πρόσβαση στο κοινό ledger.

Γνωστό παράδειγμα υλοποίησης ενός δικτύου κοινοπραξίας αποτελεί το Energy Web Foundation[13], που επικεντρώνεται στη χρήση της τεχνολογίας του blockchain για την ενίσχυση και παρακολούθηση των ενεργειών που στοχεύουν στην απαλλαγή από τον άνθρακα στην παγκόσμια οικονομία.

Τα *πλεονεκτήματα* της χρήσης ενός δικτύου κοινοπραξίας είναι:

- Η δυνατότητα να τροποποιηθεί κατά βούληση ο έλεγχος των πόρων.
- Η αυξημένη ασφάλεια σε σχέση με τα ιδιωτικά δίκτυα blockchain, λόγω των περισσότερων κόμβων που συμμετέχουν.
- Η μεγαλύτερη αποτελεσματικότητα σε σχέση με τα δημόσια blockchain, καθώς ο αριθμός των κόμβων που συμμετέχουν είναι αρκετά μικρότερος (αλλά μεγαλύτερος από αυτόν των ιδιωτικών).
- Η ταχύτητα των συναλλαγών είναι αρκετά ικανοποιητική.
- Η ενεργειακή κατανάλωση είναι μικρότερη από αυτή των δημόσιων blockchains.
- Η χρήση σαφών και αποτελεσματικών τεχνικών διακυβέρνησης.
- Η δυνατότητα ελέγχου στην πρόσβαση ανάλογα με τον χρήστη.

Τα *μειονεκτήματα* αυτού του είδους των δικτύων περιλαμβάνουν:

- Τον κίνδυνο για την ασφάλεια του δικτύου με την απώλεια του ελέγχου ενός κόμβου του δικτύου από κακόβουλους χρήστες.
- Λιγότερη διαφάνεια, που εξαρτάται από τον ρόλο του χρήστη μέσα στο δίκτυο.
- Την επιβολή κανόνων και περιορισμών που έχουν αρνητικά αποτελέσματα στην απόδοση του δικτύου.
- Λιγότερη ανωνυμία (λόγω και της αυξημένης ανάγκης για αυθεντικοποίηση) όσο σε άλλα δίκτυα blockchain.

[13] Online Σύνδεσμος: https://www.energyweb.org/

Περιπτώσεις χρήσης στις οποίες είναι δυνατόν να χρησιμοποιηθούν δίκτυα κοινοπραξίας συμπεριλαμβάνουν την παρακολούθηση τροφίμων, καθώς και δίκτυα για ανακοίνωση ερευνητικών αποτελεσμάτων και δεδομένων.

1.4.4 Υβριδικά δίκτυα blockchain

Αυτού του είδους τα δίκτυα έχουν αρχίσει και εμφανίζονται πρόσφατα. Μοιάζουν πολύ με τα δίκτυα κοινοπραξίας, καθώς και αυτά συνδυάζουν στοιχεία από τους δύο βασικούς τύπους (δημόσια και ιδιωτικά). Ενδείκνυνται σε περιπτώσεις στις οποίες δεν θέλει κάποιος να κάνει χρήση ενός αμιγώς δημόσιου δικτύου αλλά ούτε και ιδιωτικού, οπότε δημιουργείται ένας συνδυασμός που κρατά τα καλύτερα χαρακτηριστικά και των δύο, επιτρέποντας τη δυνατότητα επιλογής ποια δεδομένα/συναλλαγές θα γίνουν ιδιωτικά και ποια δημόσια.

Σημαντική διαφορά, όμως, στην περίπτωση των υβριδικών δικτύων (σε σχέση με αυτά της κοινοπραξίας) είναι ότι χρειάζεται άδεια για την πρόσβαση στο δίκτυο. Εφόσον αυτή αποκτηθεί, τότε όλοι έχουν πρόσβαση τόσο στα ιδιωτικά χαρακτηριστικά του δικτύου όσο και στα δημόσια, χωρίς να υπάρχουν περαιτέρω ρόλοι μέσα σε αυτό. Επιπλέον, συνήθως ελέγχεται από μια οντότητα με την προσθήκη των ελεύθερων (δημόσιων) διεργασιών.

Μια από τις πιο γνωστές υλοποιήσεις στην κατηγορία αυτή αποτελεί η πλατφόρμα Dragonchain[14], η οποία προσφέρει επιχειρηματικές λύσεις και εφαρμογές που μπορούν εύκολα να ενσωματωθούν στις ανάγκες μιας επιχείρησης. Επίσης, υπάρχει και το IBM Food Trust[15], το οποίο αποτελεί μια λύση που αφορά τη διαχείριση της εφοδιαστικής αλυσίδας.

Στα *πλεονεκτήματα* των δικτύων blockchain του τύπου αυτού περιλαμβάνονται:

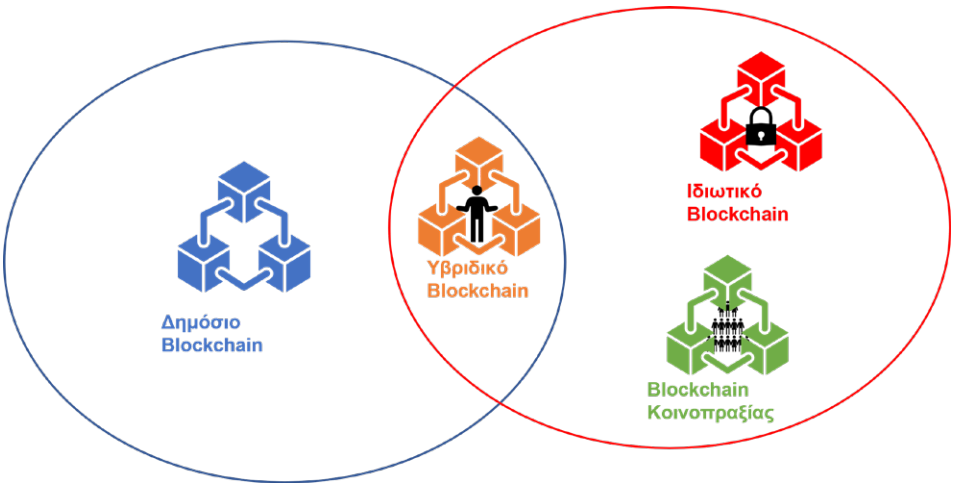
- Η δυνατότητα επιλογής ώστε να μην είναι δημόσια όλα τα δεδομένα και οι συναλλαγές στο δίκτυο.
- Η ευκολία αλλαγής των κανόνων βάσει των αναγκών των μελών του δικτύου.
- Η καλύτερη επεκτασιμότητα σε σχέση με ένα δημόσιο δίκτυο.
- Η υποστήριξη ενός ιδιωτικού επιπέδου παρά τη συμμετοχή των χρηστών σε ένα δημόσιο δίκτυο.

Στα *μειονεκτήματα* αυτού του είδους των δικτύων παρατηρούνται:

- Η έλλειψη πλήρους διαφάνειας των συναλλαγών.
- Η απουσία κινήτρων και επιβραβεύσεων για τη συμμετοχή των χρηστών.
- Η δυσκολία αναβάθμισης του (upgrade) όποτε αυτό κριθεί απαραίτητο.

Δίκτυα υβριδικού τύπου blockchain είναι δυνατόν να προκριθούν σε εφαρμογές για πώληση ακινήτων, όπου τα εταιρικά δεδομένα μπορούν να βρίσκονται στο ιδιωτικό δίκτυο ενώ τα δεδομένα που ενδιαφέρουν τους αγοραστές στο δημόσιο. Χρήση επίσης τέτοιων δικτύων μπορεί να γίνει και σε εφαρμογές λιανικής πώλησης.

Στην **Εικόνα 1.5** αποδίδονται οι διάφοροι τύποι υλοποιήσεων blockchain που έχουν μελετηθεί μέχρι τώρα.



Εικόνα 1.5 *Μια γραφική απεικόνιση των τύπων blockchain.*

[14] Online Σύνδεσμος: https://dragonchain.com/
[15] Online Σύνδεσμος: https://www.ibm.com/blockchain/solutions/food-trust

Έχοντας ολοκληρώσει την παρουσίαση των γνωστών τύπων δικτύων Blockchain, τα βασικά χαρακτηριστικά κάθε τύπου δικτύου συνοψίζονται στον **Πίνακα 1.1.**

Τύπος Δικτύου	Ελεύθερη Πρόσβαση	Πρόσβαση στο ledger	Ταχύτητα Συναλλαγών	Αποκεντροποίηση	Επεκτασιμό-τητα	Διαφάνεια	Κατανάλωση Ενεργ.
Δημόσιο	Πλήρης	Πλήρης	Μικρή	Πλήρης	Μικρή	Πλήρης	Μεγάλη
Ιδιωτικό	Με άδεια	Πιθανή	Μεγάλη	Μικρή	Μεγάλη	Μικρή	Μικρή
Κοινοπραξία	Με άδεια	Πιθανή	Πολύ Μεγάλη	Μερική	Μεγάλη	Μικρή	Μικρή
Υβριδικό	Με άδεια	Δυνατή	Μεγάλη	Μεγάλη	Πολύ Μεγάλη	Μικρή	Μικρή

Πίνακας 1.1 *Συγκεντρωτικά χαρακτηριστικά των 4 τύπων δικτύων blockchain.*

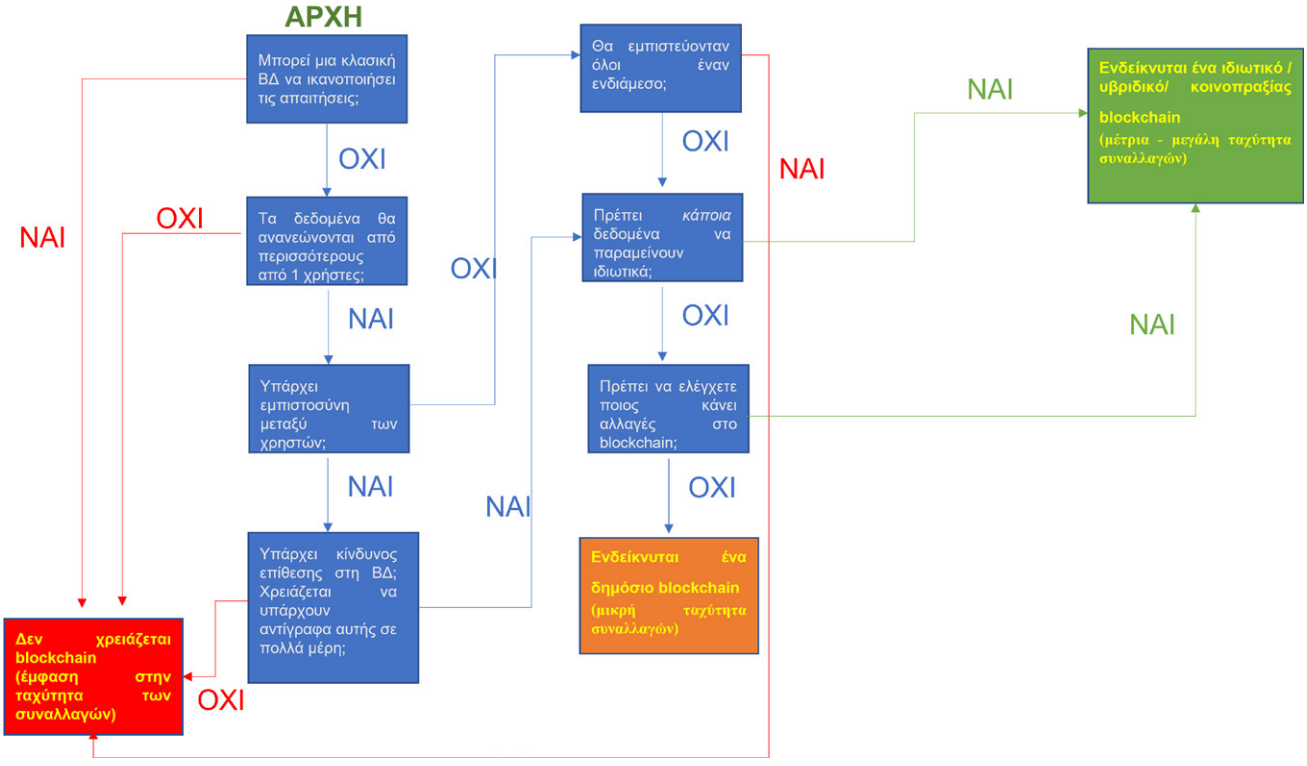
1.5 Έλεγχος για χρήση μιας λύσης που βασίζεται στην τεχνολογία blockchain

Το ενδιαφέρον που κέρδιζε συνεχώς η τεχνολογία του blockchain είχε ως αποτέλεσμα την αναζήτηση και άλλων περιπτώσεων εφαρμογής της. Προς την κατεύθυνση αυτή βοήθησε η παρουσία του δικτύου Ethereum με τις δυνατότητες που αυτό παρέχει, οι οποίες είναι πέρα από την εφαρμογή ψηφιακών πληρωμών (βλ. Κεφάλαιο 2).

Όπως έχει αναφερθεί, η τεχνολογία του blockchain έχει ορισμένα σημαντικά πλεονεκτήματα (π.χ. διαφάνεια, αξιοπιστία, αποκεντροποίηση, ιχνηλάτηση) αλλά και αρκετά μειονεκτήματα (π.χ. κατανάλωση ενέργειας, επεκτασιμότητα, ταχύτητα συναλλαγών). Τα τελευταία αυτά πρέπει να αντιμετωπιστούν, αφού μπορεί να επιδράσουν διαφορετικά στην απόδοση του συστήματος, ανάλογα και με την υπό μελέτη περίπτωση. Επομένως, είναι σημαντικό να είναι γνωστός ακριβώς ο τρόπος που θα χρησιμοποιηθεί η τεχνολογία αυτή ώστε να εξασφαλιστεί ότι θα σημειωθεί βελτίωση στην απόδοση που θα προέλθει από τη χρήση αυτής.

Στην **Εικόνα 1.6** έχουν συγκεντρωθεί μια σειρά από ερωτήσεις που θα πρέπει να απαντήσει όποιος ενδιαφέρεται να δημιουργήσει μια εφαρμογή που θα βασίζεται στην τεχνολογία blockchain. Οι ερωτήσεις πρέπει να απαντηθούν έχοντας ο καθένας στο μυαλό του μια συγκεκριμένη περίπτωση χρήσης και όχι γενικά. Η σειρά των ερωτήσεων εμφανίστηκε αρχικά στο επιστημονικό περιοδικό *Spectrum* του IEEE (Peck, 2017) και εδώ έχει προσαρμοστεί κατάλληλα, επεκτείνοντας τις επιλογές για την κάλυψη όλων των τύπων δικτύου που παρουσιάστηκαν στην Ενότητα 1.4.

Με τις ερωτήσεις αυτές ο χρήστης οδηγείται βήμα βήμα τόσο στην απόφαση για τη χρήση της τεχνολογίας, αλλά και στην επιλογή του κατάλληλου τύπου δικτύου για την κάλυψη των αναγκών του. Η επιλογή γίνεται αρχικά ανάμεσα στους δύο βασικούς τύπους δικτύων (δημόσια ή ιδιωτικά) blockchain και στη συνέχεια ακολουθεί μεγαλύτερη εξειδίκευση για την επιλογή, βάσει συνθηκών, μεταξύ υβριδικών δικτύων ή δικτύων κοινοπραξίας.



Εικόνα 1.6 *Επιλογές σχεδίασης για την εφαρμογή μιας λύσης blockchain οι οποίες μπορούν να απαντηθούν έχοντας κατά νου μια συγκεκριμένη περίπτωση χρήσης της τεχνολογίας.*

Όπως γίνεται αντιληπτό από την Εικόνα 1.6:

- Δεν προτείνεται η χρήση μιας λύσης Blockchain εφόσον:
  - είναι δυνατόν να εξυπηρετηθούμε αποτελεσματικά από μια παραδοσιακή Βάση Δεδομένων (σχεσιακή ή μη σχεσιακή),
  - μόνο ένα μέλος χρειάζεται να γράφει στη Βάση Δεδομένων,
  - όλοι οι συμμετέχοντες θα ήταν πρόθυμοι να εμπιστευτούν κάποιο κοινό τρίτο μέρος,
  - η Βάση Δεδομένων που χρησιμοποιούμε δεν είναι πιθανό να δεχθεί επίθεση (π.χ. δεν έχει σύνδεση με το Διαδίκτυο) και δεν επιθυμούμε να υπάρχουν αντίγραφα αυτής σε άλλα μέρη.
- Προτείνεται η χρήση ενός δημόσιου δικτύου blockchain εφόσον:
  - μια παραδοσιακή Βάση Δεδομένων δεν ικανοποιεί τις ανάγκες,
  - πολλοί χρήστες θα πρέπει να μπορούν να γράφουν στη Βάση Δεδομένων,
  - δεν υπάρχει η ανάγκη για χρήση μιας κοινής έμπιστης οντότητας και δεν υπάρχει και εμπιστοσύνη μεταξύ των συμμετεχόντων,
  - χρειάζονται πολλαπλά αντίγραφα της Βάσης Δεδομένων σε διαφορετικά σημεία για την αντιμετώπιση επιθέσεων,
  - δεν υπάρχουν ιδιωτικά δεδομένα και
  - δεν είναι επιθυμητός ο έλεγχος από κάποια οντότητα για αλλαγές στο δίκτυο (σε επίπεδο λογισμικού).
- Προτείνεται η χρήση ενός ιδιωτικού δικτύου blockchain (ή ενός υβριδικού ή κοινοπραξίας) εφόσον:
  - μια παραδοσιακή Βάση Δεδομένων δεν ικανοποιεί τις ανάγκες,
  - πολλοί χρήστες θα πρέπει να μπορούν να γράφουν στη Βάση Δεδομένων,
  - δεν υπάρχει η ανάγκη για χρήση μιας κοινής έμπιστης οντότητας και δεν υπάρχει και εμπιστοσύνη μεταξύ των συμμετεχόντων,
  - χρειάζονται πολλαπλά αντίγραφα της Βάσης Δεδομένων σε διαφορετικά σημεία για την αντιμετώπιση επιθέσεων,
  - υπάρχουν ιδιωτικά δεδομένα (όχι μόνον) και
  - υπάρχει έλεγχος από κάποια οντότητα για αλλαγές στο δίκτυο (σε επίπεδο λογισμικού).

Με την ολοκλήρωση των απαντήσεων στις ερωτήσεις της Εικόνας 1.6 θα υπάρχει η βάση για την εύρεση της κατάλληλης λύσης στην κάθε περίπτωση. Είναι σημαντικό να ολοκληρωθεί η διαδικασία αυτή, καθώς είναι πιθανό να βρεθεί ότι δεν είναι προτεινόμενη η χρήση της τεχνολογίας blockchain στην υπό μελέτη περίπτωση και, έτσι, να αποφευχθούν αναγκαία έξοδα για την ολοκλήρωσή της.

Επιπλέον, εφόσον βρεθεί ότι η εφαρμογή μιας λύσης που είναι βασισμένη στην τεχνολογία blockchain είναι κατάλληλη, τότε είναι δυνατή η περαιτέρω μελέτη και αναζήτηση των απαραίτητων χαρακτηριστικών υλοποίησης από τις υπάρχουσες (βλ. Ενότητα 1.4).

Βεβαίως, ιδιαίτερα αν έχει προκύψει η επιλογή ιδιωτικού δικτύου ή αν δεν είναι εύκολο να απαντηθεί η ερώτηση για χρήση δημόσιων ή ιδιωτικών δεδομένων, θα πρέπει να αναζητηθούν και επιλογές που ανήκουν στους δύο πιο πρόσφατους τύπους δικτύων (κοινοπραξίας και υβριδικών).

Βιβλιογραφία

Ault, M. (2018). *Why new off-chain storage is required for blockchains*. Document version 1.0. 2018. Online πηγή: https://www.researchgate.net/publication/328513107_Why_new_off-chain_storage_is_required_for_blockchains_Document_version_10

Back, A. (1997). *HashCash Popular proof-of-work system*. March 1997. Online πηγή: http://hashcash.org/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Bayer, D., Haber, S., & Stornetta, W. S. (1992). Improving the Efficiency and Reliability of Digital Time-Stamping. *Sequences*, 2, pp. 329-334.

Buterin, V. (2013). *Ethereum. White paper.* Online πηγή: https://ethereum.org/en/whitepaper/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *White paper*, *3*(37). Online πηγή: https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/Etherium.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Dwork, C., & Naor, M. (1992). Pricing via Processing or Combatting Junk Mail. *Advances in Cryptology – Crypto '92. Lecture Notes in Computer Science*. Springer, 740, 18 May 2001, pp. 139-147.

Haber, St., & Stornetta, S. W. (1991). How to time-stamp a digital document. *Journal of Cryptology*, *3*, pp. 99-111.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* (TOPLAS), 4(3), pp. 382-401.

Merkle, R. (1989). A Certified Digital Signature. *CRYPTO 20*. August 1989. Online πηγή: https://link.springer.com/content/pdf/10.1007/0-387-34805-0_21.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Nakamoto, S. (2008). Bitcoin: *A Peer-to-Peer Electronic Cash System*. 2008. Πηγή: https://bitcoin.org/bitcoin.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Peck, M. E. (2017). Blockchain world – Do you need a blockchain? This chart will tell you if the technology can solve your problem. *IEEE Spectrum, 54*(10), pp. 38-60. Online πηγή https://ieeexplore.ieee.org/abstract/document/8048838

Sherman, A. T., Javani, F., Zhang, H., & Golaszewski, E. (2019). On the Origins and Variations of Blockchain Technologies. *IEEE Security & Privacy*, 17(1), pp. 72-77.

ΚΕΦΑΛΑΙΟ 2

Γνωστές Υλοποιήσεις/Πλατφόρμες: Τα Παραδείγματα του Bitcoin και του Ethereum

Σύνοψη

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση των αρχικών και πιο δημοφιλών υλοποιήσεων της τεχνολογίας blockchain, του Bitcoin και του Ethereum. Παρουσιάζονται τα ιδιαίτερα χαρακτηριστικά τους και ο τρόπος λειτουργίας τους, με έμφαση στην ελεύθερη πρόσβαση σε αυτά, τις τεχνικές συναίνεσης που χρησιμοποιούν, αλλά και ανάλυση των βασικών διαφορών τους (π.χ. δημιουργία αποκεντρωμένων εφαρμογών και έξυπνων συμβάσεων).

Σκοπός είναι να γίνει μια γνωριμία με τις δύο βασικές υλοποιήσεις, που χρησιμοποιούνται συχνά ως αναφορές, και να ενισχυθούν οι γνώσεις σας σε αυτές σε επιλεγμένους τομείς στα κεφάλαια που ακολουθούν.

Προαπαιτούμενη γνώση

Η μελέτη του Κεφαλαίου 1.

2.1 Περιγραφή και βασικά χαρακτηριστικά

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, τα Bitcoin και Ethereum αποτελούν τις δύο υλοποιήσεις blockchain που απέκτησαν σημαντική δημοφιλία και χαρακτηρίζουν δύο σημαντικές χρονικές περιόδους (βλ. Ενότητα 1.3) στην ιστορία της τεχνολογίας. Οι δύο αυτές λύσεις έχουν αρκετές ομοιότητες αλλά και σημαντικές διαφορές.

Στο κεφάλαιο αυτό αναλύονται τα βασικά χαρακτηριστικά τους, οι ομοιότητες και οι διαφορές τους.

2.2 Bitcoin

Το Bitcoin δημιουργήθηκε το 2009 ως ένα ελεύθερης πρόσβασης, χωρίς ανάγκη για απόκτηση άδειας για τη συμμετοχή, *δημόσιο blockchain δίκτυο*. Σκοπός του ήταν να αποτελέσει ένα εναλλακτικό ηλεκτρονικό σύστημα πληρωμής, ελεύθερο από ενδιάμεσους. Έτσι, θεωρείται ότι αποτελεί, κυρίως, ένα μέσο ανταλλαγής (ή συναλλαγής) κρυπτονομισμάτων, που ονομάζονται *bitcoins* (ή BTC σε συντομογραφία).

Ο κώδικας του Bitcoin είναι ελεύθερης πρόσβασης και μπορεί ο οποιοσδήποτε να τον κατεβάσει, να τον εγκαταστήσει, καθώς και να τον διαβάσει. Με την εγκατάσταση του κώδικα, ο χρήστης γίνεται μέλος ενός κατανεμημένου δικτύου από ομότιμους κόμβους (δίκτυο P2P) που «τρέχει» το πρωτόκολλο του Bitcoin και μπορεί να λειτουργεί ως κόμβος αυτού.

Κάθε κόμβος στο δίκτυο του Bitcoin μπορεί να:

- Αποκτήσει πρόσβαση στο κοινό ledger που έχει την αλυσίδα των blocks με τις συναλλαγές των χρηστών του δικτύου και να κατεβάσει το δικό του αντίγραφο. Αυτό θα ανανεώνεται κάθε φορά που προστίθεται ένα καινούργιο block.
- Ακολουθεί τους κανόνες συναίνεσης του πρωτοκόλλου, που αφορούν τους κανόνες που πρέπει να ισχύουν για να επιβεβαιωθεί η καθεμία συναλλαγή που τίθεται προς εκτέλεση στο δίκτυο του Bitcoin για την ενσωμάτωσή της σε ένα μελλοντικό block. Επιπρόσθετα, οι κανόνες αυτοί αφορούν την ολοκλήρωση όλων των απαραίτητων ενεργειών για την εξασφάλιση της εγκυρότητας του προτεινόμενου block.
- Επιβεβαιώνει την παραγωγή (ή εξόρυξη) νέων (κρυπτο-)νομισμάτων στο δίκτυο.
- Χρησιμοποιεί τον αλγόριθμο PoW ως τεχνική συναίνεσης συνολικά στο δίκτυο σχετικά με την απόφαση για την επιλογή του block που θα είναι το επόμενο που θα προστεθεί στην αλυσίδα.
- Στη συνέχεια θα παρουσιαστούν αναλυτικά τα ιδιαίτερα χαρακτηριστικά του Bitcoin. Ορισμένα από αυτά θα αναπτυχθούν σε μεγαλύτερη έκταση σε ξεχωριστά κεφάλαια, αλλά θα γίνει μια σύντομη παρουσίασή τους στην ενότητα αυτή.

Με τον τρόπο αυτό θα είναι δυνατή η σύγκριση των δικτύων αυτών στην Ενότητα 2.4.

2.2.1 Κόμβοι και λειτουργίες

Για να αποτελέσει μέλος του δικτύου του Bitcoin, ένας κόμβος θα πρέπει να αποκτήσει μια διεύθυνση δικτύου και να συνδεθεί με υπάρχοντες ενεργούς κόμβους αυτού.

Για να το πετύχει αυτό, αρχικά χρησιμοποιεί την πληροφορία που υπάρχει στο πρωτόκολλο του Bitcoin και αφορά την ενημέρωση για τις διευθύνσεις συγκεκριμένων κόμβων στο δίκτυο. Αυτοί, με τη σειρά τους, θα δώσουν πληροφορίες για επιπλέον διευθύνσεις ενεργών κόμβων του δικτύου Bitcoin. Στην ουσία πρόκειται για κόμβους που παίζουν ρόλο αντίστοιχο με αυτόν που έχουν για τη λειτουργία του Διαδικτύου οι εξυπηρετητές DNS[16], βοηθώντας δηλαδή τη σύνδεση νέων κόμβων με το δίκτυο του Bitcoin ενημερώνοντας τους νέους κόμβους για τις διευθύνσεις IP των ενεργών κόμβων.

Μόλις ο νέος κόμβος ενημερωθεί, θα προσπαθήσει να δημιουργήσει σύνδεση επικοινωνώντας με τους κόμβους που ενημερώθηκε, για να εισαχθεί στο δίκτυο του Bitcoin. Με την επίτευξη της σύνδεσης ακολουθεί ένας μηχανισμός *χειραψίας* (*handshaking*), ανάλογος με αυτόν που συμβαίνει και στα δίκτυα υπολογιστών κατά τη δημιουργία συνδέσεων TCP[17]. Η πληροφορία που ανταλλάσσεται στη διαδικασία της χειραψίας αφορά την απόκτηση λεπτομερειών για τον κόμβο και το δίκτυο του Bitcoin. Έτσι, λαμβάνονται πληροφορίες, όπως η έκδοση του πρωτοκόλλου του Bitcoin που χρησιμοποιείται από τον κόμβο, οι διευθύνσεις IP των δύο συμμετεχόντων και το ύψος του εδαφίου (ledger), δηλαδή ο αριθμός του τελευταίου block που γνωρίζει ο κόμβος.

Από το σημείο αυτό και μετά η σύνδεση με το δίκτυο έχει επιτευχθεί. Ο τρόπος που θα λειτουργήσει ο καινούργιος κόμβος εξαρτάται από τον *τύπο* και τον *ρόλο* που έχει επιλέξει να παίξει. Μπορεί ως μέλη ενός δικτύου ομότιμων κόμβων όλοι οι κόμβοι να είναι ίσοι, όμως στο δίκτυο του Bitcoin ένας κόμβος μπορεί να έχει έναν διαφορετικό ρόλο, ανάλογα με τις ενέργειες που μπορεί να εκτελέσει.

Οι ενέργειες που μπορεί να εκτελέσει ένας κόμβος στο δίκτυο του Bitcoin χωρίζονται σε *τέσσερις βασικές κατηγορίες* (Antonopoulos, 2017): α) δικτυακές ενέργειες, β) ενέργειες για την αποθήκευση της Βάσης Δεδομένων (ledger) του blockchain, γ) ενέργειες για την εξόρυξη bitcoins και δ) ενέργειες για την προσφορά υπηρεσιών πορτοφολιού.

Η **Εικόνα 2.1** μας δείχνει έναν κόμβο, που ονομάζεται *κόμβος Αναφοράς* (*Reference node*) και ο οποίος μπορεί να εκτελέσει και τις τέσσερις αυτές ενέργειες.

Εικόνα 2.1 *Οι τέσσερις ενέργειες που μπορεί να εκτελέσει ένας κόμβος Αναφοράς (Reference node) στο δίκτυο του Bitcoin.*

Ειδικότερα, ένας κόμβος στο δίκτυο του Bitcoin μπορεί να εκτελέσει έναν συνδυασμό από τις εξής ενέργειες:

- *Δικτυακές ενέργειες*: Εκτελούνται από *όλους τους κόμβους* του δικτύου, καθώς περιλαμβάνουν τις λειτουργίες που είναι απαραίτητες για την είσοδο και τη συμμετοχή κόμβων στο δίκτυο. Περιλαμβάνουν, ακόμα, τις λειτουργίες δρομολόγησης για κάθε κόμβο που επιτρέπει την εύρεση και

[16] DNS (Dynamic Name Service) είναι το πρωτόκολλο που αντιστοιχεί τη διεύθυνση μιας ιστοσελίδας στην IP του web server που τη φιλοξενεί.

[17] TCP (Transmission Control Protocol) είναι πρωτόκολλο του Επιπέδου Μεταφοράς που εγγυάται την επιτυχημένη μεταφορά ενός πακέτου δεδομένων. Απαιτεί την εγκαθίδρυση επικοινωνίας μεταξύ των δύο πλευρών (υλοποιώντας τη χειραψία).

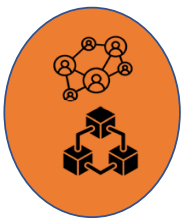
την επικοινωνία με άλλους κόμβους στο δίκτυο. Επίσης, τις *διαδικασίες επικύρωσης και προώθησης* των συναλλαγών και, κατ' επέκταση, των blocks, που πραγματοποιούνται από όλους τους κόμβους του δικτύου.

- *Αποθήκευση Εδαφίου* (*Ledger*): Ορισμένοι κόμβοι, επιπλέον, αναλαμβάνουν να αποθηκεύσουν όλο το εδάφιο (ledger) και, βέβαια, να το ανανεώνουν κάθε φορά που ένα καινούργιο block συνδέεται στην αλυσίδα. Με τον τρόπο αυτόν οι κόμβοι αυτοί μπορούν να επιβεβαιώνουν συναλλαγές σε όλη την αλυσίδα χωρίς κάποια επιπλέον πληροφορία από άλλον κόμβο.
  Εναλλακτικά, υπάρχει και η δυνατότητα να διατηρείται μόνο ένα μέρος του εδαφίου (ledger) από έναν κόμβο, προς χάριν οικονομίας σε αποθηκευτικό χώρο. Οι κόμβοι αυτοί, γνωστοί και ως *απλοί κόμβοι* (*lightweight*), έχουν επικοινωνία με κάποιον άλλο κόμβο που έχει όλο το εδάφιο (ledger) για να μπορέσουν να επιβεβαιώσουν μια παλαιότερη συναλλαγή, χρησιμοποιώντας μια ειδική μέθοδο με το όνομα *Simplified Payment Verification* (*SPV*) για την επιβεβαίωση (Nakamoto, 2008). Έτσι, οι lightweight κόμβοι δεν θεωρούνται ότι μπορούν να εκτελέσουν την ενέργεια *αποθήκευσης του εδαφίου* (ledger).
- *Εξόρυξη*: Αφορά το σύνολο των ενεργειών που εκτελούνται από έναν κόμβο στην προσπάθεια που κάνει για την επίλυση των μαθηματικών προβλημάτων που συμπεριλαμβάνονται στον αλγόριθμο του PoW. Η επιτυχής επίλυση του προβλήματος θα αποτελέσει και το έναυσμα για την προσθήκη του block στην αλυσίδα του blockchain και στο κοινό εδάφιο (ledger).
- *Προσφορά Υπηρεσιών Πορτοφολιού* (*Wallet*): Αφορά το σύνολο των ενεργειών που χρειάζονται για την αποστολή και λήψη συναλλαγών. Περιλαμβάνει πρόσβαση και αποθήκευση των κλειδιών του χρήστη και της διεύθυνσής του. Για τη σωστή λειτουργία τους οι κόμβοι που θα εφαρμόζουν την ενέργεια της προσφοράς υπηρεσιών πορτοφολιού θα πρέπει να έχουν πρόσβαση και στο εδάφιο (ledger), για να μπορέσουν να επιβεβαιώσουν την ορθότητα των συναλλαγών που αφορούν τη διεύθυνση του χρήστη.

Ο συνδυασμός των ενεργειών που χαρακτηρίζουν τη λειτουργία ενός κόμβου στο δίκτυο του Bitcoin καθορίζουν τον τύπο και τις δυνατότητες του κόμβου αυτού.

Στη συνέχεια θα αναλυθούν τα χαρακτηριστικά των κάθε τύπου κόμβων που υπάρχουν βάσει των διαφορετικών συνδυασμών ενεργειών που συναντώνται στο δίκτυο του Bitcoin.

- Κόμβοι Αναφοράς (*Reference Nodes*): Όπως αναφέρθηκε και προηγουμένως, κόμβοι αναφοράς καλούνται οι κόμβοι εκείνοι οι οποίοι μπορούν να προσφέρουν και τις τέσσερις δυνατές ενέργειες. Δηλαδή να έχουν δυνατότητα δρομολόγησης στο δίκτυο, να κρατούν αντίγραφο ολόκληρου του εδαφίου (ledger), να εργάζονται για τη δημιουργία του επόμενου block ως *miners* και να μπορούν να συναλλάσσονται με χρήση των υπηρεσιών ενός πορτοφολιού στο δίκτυο.



- *Πλήρεις Κόμβοι* (*Full Nodes*): Αυτοί οι κόμβοι αποτελούν τη ραχοκοκαλιά του δικτύου, καθώς είναι κόμβοι οι οποίοι αποθηκεύουν και ανανεώνουν ολόκληρο το εδάφιο (ledger) και, επίσης, έχουν τη δυνατότητα δρομολόγησης για τη μεταφορά του σε άλλους που θα ζητήσουν να ενημερωθούν από αυτούς. Οι κόμβοι αυτοί (περίπου 10.000 σε αριθμό στο δίκτυο Bitcoin) παίζουν σημαντικό ρόλο γιατί κρατούν και ενημερώνουν την ιστορία του δικτύου και συνεισφέρουν στην επικύρωση και επιβεβαίωση των συναλλαγών αυτού.
  Μιας και το δίκτυο του Bitcoin είναι δημόσιο, είναι δυνατόν σε οποιονδήποτε να διατηρήσει έναν τέτοιον κόμβο, αρκεί να μπορέσει να καλύψει τις απαιτήσεις σε hardware που χρειάζεται για αυτό (ιδιαίτερα σε θέματα μνήμης και αποθηκευτικού χώρου). Ένας τέτοιος κόμβος συνδυάζει τις *δικτυακές ενέργειες* και τις *ενέργειες αποθήκευσης του εδαφίου* (*ledger*).

Οι πλήρεις κόμβους διαχωρίζονται μεταξύ τους:

α) σε *κόμβους αρχείου* (*archival nodes*), οι οποίοι και έχουν όλο το εδάφιο (ledger) από το πρώτο block (ονομάζεται block γέννησης) μέχρι το πιο πρόσφατο, και σε

β) *επιμέρους κόμβους* (*pruned nodes*), οι οποίοι αρχίζουν να αποθηκεύουν το εδάφιο (ledger) μέχρις ότου φθάσουν να έχουν αποθηκεύσει έναν συγκεκριμένο όγκο δεδομένων. Μόλις φθάσουν το όριο αυτό, τότε αντικαθιστούν τα blocks με τα νεότερα. Διατηρούν μόνο την επικεφαλίδα και τον αριθμό των blocks που αντικαθιστούν.

Να σημειωθεί εδώ ότι οι επιμέρους πλήρεις κόμβοι μπορούν να συμμετέχουν κανονικά στη διαδικασία της συναίνεσης.

- *Απλοί Κόμβοι* (*Lightweight Nodes*): Πρόκειται για κόμβους, όπως αναφέρθηκε και παραπάνω, οι οποίοι δεν έχουν το πλήρες αντίγραφο του εδαφίου (ledger), με αποτέλεσμα να μην είναι δυνατόν να επιβεβαιώσουν μια συναλλαγή χωρίς τη βοήθεια ενός πλήρους κόμβου. Επομένως, η σύνδεση των κόμβων αυτών με κάποιον πλήρη κόμβο είναι συνήθης. Αντιθέτως, οι κόμβοι αυτοί μπορούν να δρομολογήσουν μέσα στο δίκτυο, αλλά είναι ιδιαίτερα γνωστοί για την ικανότητά τους να προσφέρουν υπηρεσίες πορτοφολιού.
  Έτσι, επιτρέπουν τη μετάδοση συναλλαγών και τη λήψη αυτών. Οι περισσότεροι χρήστες του δικτύου Bitcoin οι οποίοι ενδιαφέρονται να συνδιαλλαγούν με αυτό δημιουργούν έναν λογαριασμό σε ένα πορτοφόλι και ουσιαστικά εισέρχονται στο δίκτυο του Bitcoin ως απλοί κόμβοι.

  Ένας τέτοιος κόμβος συνδυάζει τις δικτυακές ενέργειες και αυτές που αφορούν τις *υπηρεσίες πορτοφολιού*.
- *Κόμβοι Εξόρυξης* (*Miners*): Πρόκειται για κόμβους οι οποίοι και συμμετέχουν ενεργά στη δημιουργία της αλυσίδας, καθώς προσπαθούν να λύσουν το μαθηματικό πρόβλημα που θα τους δώσει το εισιτήριο για να βρουν το επόμενο block.

Επομένως θα πρέπει να συμπεριλαμβάνουν την ενέργεια της *εξόρυξης*.
Επιπλέον αυτής της ενέργειας, υπάρχουν δύο πιθανοί συνδυασμοί:

α) να λειτουργεί ο κόμβος ανεξάρτητα από άλλους, οπότε θα πρέπει να έχει ακόμα και τη δυνατότητα να εκτελεί *δικτυακές ενέργειες* αλλά και να *αποθηκεύει αντίγραφο του εδαφίου* (*ledger*) για να προχωρήσει στην εξόρυξη του νέου block, ή

β) να αποτελεί μέρος μιας δεξαμενής κόμβων εξόρυξης, οπότε, πέρα από την αντίστοιχη ενέργεια, χρειάζεται να έχει επικοινωνία με τον βασικό κόμβο που θα διαχειρίζεται όλους τους κόμβους που ανήκουν στη δεξαμενή αυτή.

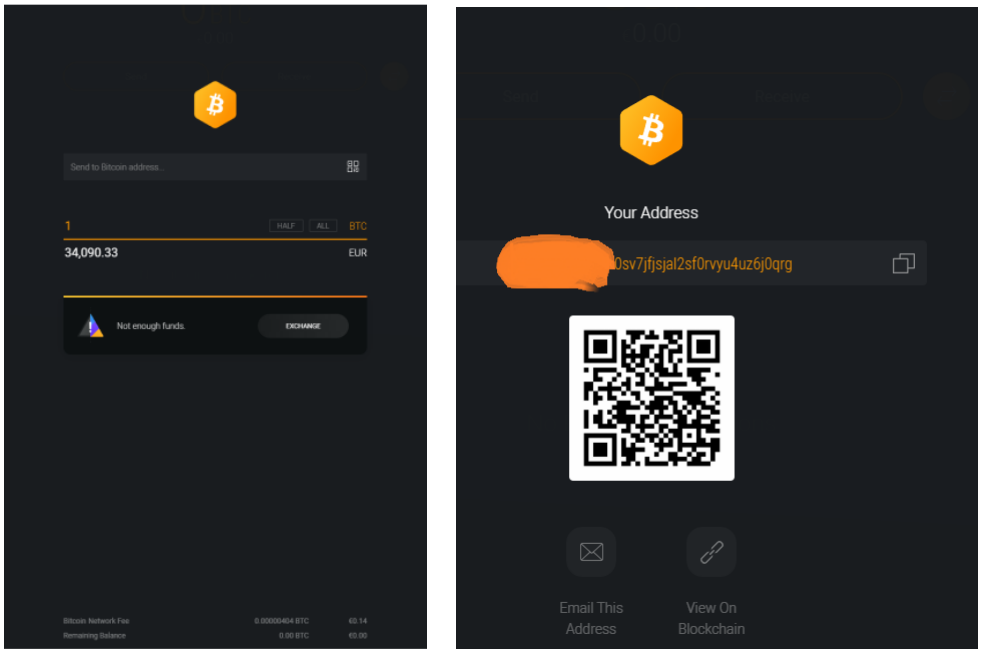
2.2.2 Συναλλαγές – Πορτοφόλια

Το δίκτυο του Bitcoin έχει ως κύρια λειτουργία του να παίξει τον ρόλο ενός μέσου ανταλλαγής κρυπτονομισμάτων (bitcoins) με τη δημιουργία και μετάδοση συναλλαγών για την αγορά προϊόντων και υπηρεσιών.

Ένας χρήστης μόλις αποκτήσει πρόσβαση στο δίκτυο του Bitcoin, δημιουργώντας λογαριασμό σε ένα πορτοφόλι (wallet) και αποτελώντας έναν lightweight κόμβο στο δίκτυο, χρειάζεται να έχει τη διεύθυνση ενός άλλου χρήστη για να μπορέσει να την προσθέσει στο πορτοφόλι του (βλ. Κεφάλαια 3 και 4). Κατόπιν, αφού συμπληρώσει το ποσό που θα μεταφέρει, μπορεί να στείλει τη συναλλαγή στο δίκτυο προς επικύρωση και ενσωμάτωση σε ένα από τα επόμενα blocks της αλυσίδας.

Δηλαδή, για τη δημιουργία μιας συναλλαγής απαιτούνται τρία στοιχεία:

- η διεύθυνση του αποστολέα,
- η διεύθυνση του παραλήπτη και
- το ποσό της μεταφοράς 1 bitcoin (BTC) από μια εφαρμογή πορτοφολιού, μαζί με την τρέχουσα αντιστοιχία της σε ευρώ. Για να ολοκληρωθεί η συναλλαγή, υπολείπεται (πέρα από την κατοχή του ποσού) και η συμπλήρωση της διεύθυνσης του παραλήπτη. Η διεύθυνση θα πρέπει να είναι μια πραγματική διεύθυνση στο δίκτυο του Bitcoin (βλ. Κεφάλαιο 3).



A B

Εικόνα 2.2 *Χρήση ενός πορτοφολιού, δηλαδή μιας εφαρμογής στο web, για την αποστολή και λήψη bitcoins: Α. Η διαδικασία αποστολής 1 BTC από μια εφαρμογή πορτοφολιού, αφού συμπληρωθεί η διεύθυνση του παραλήπτη. Β. Η διαδικασία λήψης μέσω διαμοιρασμού της δικής μας διεύθυνσης Bitcoin στον αποστολέα για να τη συμπληρώσει στην αποστολή (αντίστοιχη εικόνα με την Α).*

Προσοχή: *Οι διευθύνσεις είναι πραγματικές και έχουν καλυφθεί για την αποφυγή αποστολής οποιασδήποτε συναλλαγής προς αυτές, καθώς ο λογαριασμός αυτός χρησιμοποιείται μόνο κατά τη συγγραφή του βιβλίου και δεν θα συνεχιστεί η παρακολούθηση/υποστήριξή του μετά.*

Στην **Εικόνα 2.2Β** φαίνεται η διεύθυνση του λογαριασμού που δημιουργήθηκε στο πορτοφόλι που χρησιμοποιήθηκε για την επίδειξη του τρόπου παραλαβής bitcoins. Όπως φαίνεται, όταν θέλει κάποιος να παραλάβει bitcoins, θα πρέπει να ενημερώσει τον αποστολέα για τη διεύθυνση στην οποία θα πρέπει να σταλούν αυτά. Η εφαρμογή προσφέρει εναλλακτικούς τρόπους για τη διάδοση της πληροφορίας (π.χ. η αντιγραφή και αποστολή με email είναι ένας κοινός τρόπος).

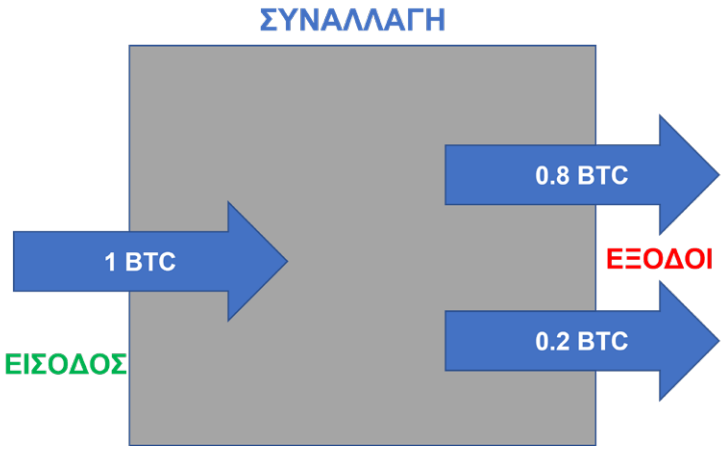
Μόλις συμπληρωθούν τα απαραίτητα στοιχεία που αναφέραμε προηγουμένως, τότε το πορτοφόλι οφείλει να βρει στην κατοχή μας το ποσό που θέλουμε να μεταφέρουμε. Αυτό γίνεται με τον εντοπισμό του στο ledger (εδάφιο) από τη χρήση του σε προηγούμενες συναλλαγές οι οποίες έφθασαν στη διεύθυνση του πορτοφολιού και (αθροιστικά) περιέχουν ένα ποσό ίσο (ή και μεγαλύτερο) από αυτό που είναι προς μεταφορά στην τρέχουσα (νέα) συναλλαγή. Αυτές οι συναλλαγές προς τη διεύθυνσή μας τοποθετούνται στην είσοδο της νέας μας συναλλαγής (βλ. Κεφάλαιο 4 για τη δομή μιας συναλλαγής στο δίκτυο του Bitcoin και Εικόνα 2.3) και στην έξοδο συμπληρώνεται η διεύθυνση του παραλήπτη.

Εφόσον στην είσοδο το ποσό που μπαίνει είναι αθροιστικά μεγαλύτερο από αυτό που δηλώνουμε προς μεταφορά, το υπόλοιπο πρέπει να γυρίσει πίσω, και έτσι προστίθεται η διεύθυνση του πορτοφολιού που κάνει τη συναλλαγή στην έξοδό της. Έτσι, στην περίπτωση αυτή θα υπάρχουν δύο διευθύνσεις στην έξοδο της συναλλαγής: η διεύθυνση του παραλήπτη και η διεύθυνση του πορτοφολιού.

Στην πράξη, η συναλλαγή προσομοιάζει τις χρηματικές συναλλαγές που έχουμε συνηθίσει στον φυσικό κόσμο με φυσικά χρήματα. Αφορά την απόδοση ενός ποσού, και εφόσον αυτό είναι μεγαλύτερο από την αξία του προϊόντος που αγοράζεται, τότε το υπόλοιπο επιστρέφεται ως ρέστα. Το ίδιο συμβαίνει και σε μια συναλλαγή στο Bitcoin.

Ακολούθως, η συναλλαγή υπογράφεται ψηφιακά από τον ιδιοκτήτη των μεταφερόμενων χρημάτων (αποστολέας της συναλλαγής) και προωθείται στο δίκτυο για επικύρωση, επιβεβαίωση και διάδοση. Η προσθήκη της υπογραφής αυτής παίζει τον ρόλο της *απόδειξης ιδιοκτησίας*. Και για τον λόγο αυτόν μπαίνει έτσι ώστε να εξασφαλιστεί η ταυτότητα του αποστολέα και, βέβαια, η δυνατότητα επιβεβαίωσης της ιδιοκτησίας του μεταφερόμενου ποσού. Αυτό επιτυγχάνεται με προσπέλαση στο ledger (εδάφιο) και επιβεβαίωση.

Η **Εικόνα 2.3** δείχνει ένα παράδειγμα για τη μεταφορά του ποσού του 1 BTC που αναφέρεται στην Εικόνα 2.2. Φαίνονται τόσο οι είσοδοι όσο και οι έξοδοι, συμπεριλαμβανομένης και της εξόδου επιστροφής του υπολοίπου στη δική μας διεύθυνση.



Εικόνα 2.3 *Παράδειγμα δημιουργίας μιας συναλλαγής από μια εφαρμογή πορτοφολιού στο δίκτυο του Bitcoin.*

2.2.3 Επικύρωση και επιβεβαίωση

Όπως αναφέρθηκε προηγουμένως, υπάρχει μια συγκεκριμένη ενέργεια η οποία εκτελείται από τους περισσότερους κόμβους, και αυτή είναι η *δικτυακή λειτουργία*. Η λειτουργία αυτή, πέρα από το κομμάτι της δρομολόγησης στο δίκτυο, συμπεριλαμβάνει και τη διαδικασία της επικύρωσης (της ορθότητας) των συναλλαγών καθώς και της περαιτέρω διάδοσής τους μέσα στο δίκτυο.

Η διαδικασία της επικύρωσης περιλαμβάνει τον έλεγχο της συναλλαγής από κάθε κόμβο ως προς τη δομή και τη σύνταξή της και εξασφαλίζει ότι μια συναλλαγή είναι ορθά δομημένη και, επομένως, μπορεί να προχωρήσει για επιβεβαίωση από τους κόμβους εκείνους που θα αναλάβουν την εξόρυξη. Πρόκειται για μια διαδικασία η οποία γίνεται από όλους τους κόμβους, έτσι ώστε να εντοπιστεί νωρίς αν υπάρχουν λάθη και να μην προχωρήσει η διάδοση της συναλλαγής περαιτέρω στην περίπτωση αυτή.

Καθώς η συναλλαγή διαδίδεται στο δίκτυο, θα φθάσει τελικά σε όλους τους συνδεδεμένους σε αυτό κόμβους. Οι κόμβοι που εκτελούν την ενέργεια της εξόρυξης θα συλλέξουν τις συναλλαγές, θα επιλέξουν ένα υποσύνολο αυτών, θα επιβεβαιώσουν την ορθότητά τους και κατόπιν θα τις βάλουν σε μια σειρά με την ένταξή τους σε ένα από τα μελλοντικά blocks προτού ξεκινήσουν τη διαδικασία της εξόρυξης αυτού.

Κατά τον έλεγχο της επιβεβαίωσης μιας συναλλαγής αναζητούνται παλαιότερες συναλλαγές οι οποίες και δικαιολογούν την ύπαρξη του μεταφερόμενου ποσού από τον ιδιοκτήτη του. Για να το πετύχουν αυτό, οι κόμβοι θα πρέπει να έχουν πρόσβαση στο εδάφιο (ledger), όντας πλήρεις κόμβοι του δικτύου του Bitcoin, ακόμα και αν δεν εκτελούν την ενέργεια της εξόρυξης. Για παράδειγμα, εάν πρέπει να στείλει κάποιος 1 BTC από τη διεύθυνσή του σε μια άλλη, θα αναζητηθεί η ορθότητα των συναλλαγών που έχουν προστεθεί στην είσοδο της νέας αυτής συναλλαγής. Αν τα ποσά επιβεβαιωθούν, τότε η τρέχουσα συναλλαγή θεωρείται έγκυρη και προχωρά. Σε αντίθετη περίπτωση, η συναλλαγή ακυρώνεται και δεν προχωρά η επεξεργασία της από το δίκτυο. Αυτό απεικονίζεται και στην Εικόνα 2.2Β, όπου η εφαρμογή του πορτοφολιού είναι συνδεδεμένη με έναν πλήρη κόμβο και μπορεί αμέσως να αναγνωρίσει την απουσία κατοχής των χρημάτων που πρέπει να αποσταλούν, όπως φαίνεται και από το μήνυμα που εμφανίζεται στο κάτω μέρος της εικόνας.

Όσες συναλλαγές περάσουν τα στάδια της επικύρωσης και της επιβεβαίωσης είναι έτοιμες για να συμπεριληφθούν από κάποιον κόμβο εξόρυξης στην προσπάθεια εύρεσης του επόμενου block στην αλυσίδα του Bitcoin. Μέχρι τότε παραμένουν σε μια δεξαμενή εκκρεμών συναλλαγών που υπάρχει σε κάθε κόμβο εξόρυξης.

2.2.4 Εξόρυξη – Συναίνεση

Η διαδικασία της εξόρυξης πραγματοποιείται από τους αντίστοιχους κόμβους (miners) που μπορούν να τη φέρουν σε πέρας. Ξεκινά με την επιλογή ενός συνόλου επιβεβαιωμένων συναλλαγών, τις οποίες έχει επικυρώσει

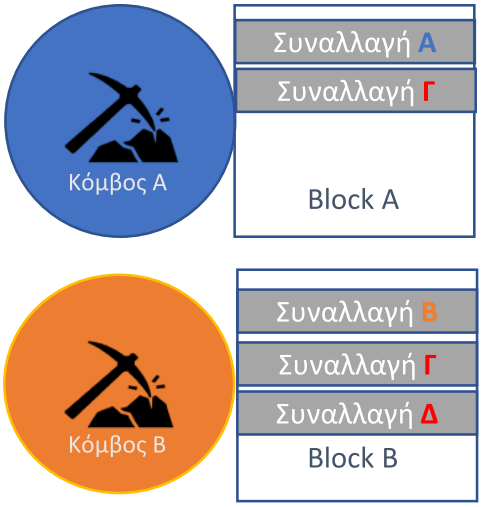
και επαληθεύσει ο κόμβος, και την προσθήκη τους στο υπό εξόρυξη block. Λόγω της κατανεμημένης φύσης του συστήματος, θα πρέπει να γίνει κατανοητό ότι τα υποψήφια νέα blocks που ετοιμάζονται για εξόρυξη από όλους τους κόμβους miners σε μια χρονική στιγμή μπορεί να μην περιέχουν τις ίδιες συναλλαγές ούτε τον ίδιο αριθμό συναλλαγών.

Έτσι, για παράδειγμα, έστω ότι υπάρχουν τέσσερις συναλλαγές με τα ονόματα *Συναλλαγή Α, Συναλλαγή Β, Συναλλαγή Γ* και *Συναλλαγή Δ*, καθώς και δύο υποψήφια blocks, το *Block Α* και το *Block Β*, των οποίων την εξόρυξη έχουν αναλάβει ο *Κόμβος Α* και ο *Κόμβος Β*.

Η σύνθεση των συναλλαγών στα δύο υποψήφια blocks μπορεί να είναι διαφορετική, όπως φαίνεται και στην **Εικόνα 2.4**. Εκεί φαίνεται ότι τα δύο νέα υποψήφια blocks (Block Α και Block Β) έχουν διαφορετική την πρώτη συναλλαγή και το Block Β έχει συμπεριλάβει και μία συναλλαγή επιπλέον (Συναλλαγή Δ).

Αυτό μπορεί να οφείλεται σε διάφορους λόγους, όπως:

- Αρχικά, στο δίκτυο του Bitcoin όλα τα νέα υποψήφια blocks θα διαφέρουν *τουλάχιστον ως προς την πρώτη συναλλαγή* που θα περιέχουν (Συναλλαγή Α για το Block Α και Συναλλαγή Β για το Block Β). Αυτό οφείλεται στο γεγονός ότι η πρώτη συναλλαγή σε κάθε κόμβο εξόρυξης αποτελεί την αμοιβή του δικτύου στον κόμβο σε περίπτωση επιτυχίας στην εύρεση του επόμενου block. Έτσι, θα είναι διαφορετική για κάθε κόμβο, μιας και θα υπάρχει η διεύθυνση του καθενός από αυτούς ως διεύθυνση προορισμού. Η συναλλαγή αυτή είναι η μόνη συναλλαγή στην οποία δημιουργούνται από το δίκτυο νέα bitcoins και ονομάζεται συναλλαγή COINBASE.
- Κατά δεύτερο λόγο, η πιο συνηθισμένη αιτία σχετίζεται με τον χρόνο διάδοσης των συναλλαγών στο δίκτυο. Έτσι, αν ο Κόμβος Β βρίσκεται πιο κοντά σε αυτόν που δημιούργησε τη Συναλλαγή Δ, θα γνωρίσει για αυτήν πριν από τον Κόμβο Α και μπορεί να τη συμπεριλάβει στο υποψήφιο νέο block πριν από αυτόν.



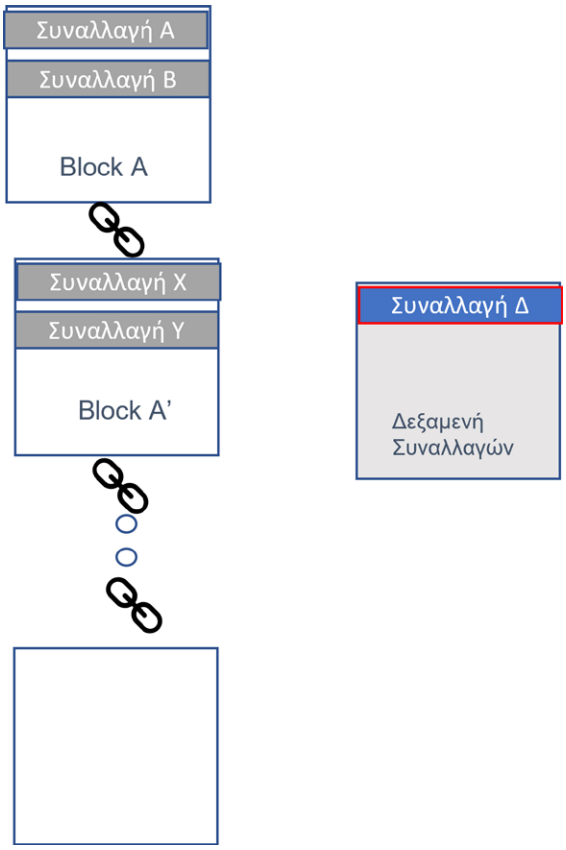
Εικόνα 2.4 *Δύο κόμβοι που ανταγωνίζονται για τη δημιουργία του επόμενου block. Οι συναλλαγές που περιέχονται στα δύο υποψήφια blocks διαφέρουν.*

Η κατάσταση που απεικονίζεται στην Εικόνα 2.4 είναι συνήθης και δεν δημιουργεί πρόβλημα στο δίκτυο του Bitcoin. Όποιο από τα δύο υποψήφια blocks κερδίσει τον ανταγωνισμό για να αποτελέσει το επόμενο block που θα προστεθεί στην αλυσίδα θα προσθέσει τις συναλλαγές που χρησιμοποίησε στην εξόρυξη στο ledger (εδάφιο).

Έτσι, οι συναλλαγές μπαίνουν και σε μια ορισμένη χρονική σειρά, που μπορεί να επιβεβαιωθεί με τον έλεγχο του εδαφίου (ledger).

Αν υπάρχει κάποια συναλλαγή που δεν είναι τύπου COINBASE και η οποία έμεινε εκτός λόγω της συμμετοχής της στο block εκείνο που δεν κατάφερε να κερδίσει στον «αγώνα» για την εύρεση του επόμενου block, θα επιστρέψει στη δεξαμενή διαθέσιμων συναλλαγών και θα είναι διαθέσιμη για επιλογή από τους κόμβους στα επόμενα υποψήφια blocks.

Στην **Εικόνα 2.5** βλέπουμε πώς είναι η αλυσίδα αν το Block Α μπει σε αυτήν και πώς η Συναλλαγή Δ περνά στη Δεξαμενή Συναλλαγών για να επιλεχθεί σε μεταγενέστερο block.



Εικόνα 2.5 *Η αλυσίδα με την προσθήκη του Block A και η επιστροφή της Συναλλαγής Δ στη δεξαμενή Συναλλαγών στους κόμβους εξόρυξης για επιλογή σε μελλοντικό υποψήφιο block.*

Κατόπιν, μόλις ολοκληρωθεί η επιλογή και η συλλογή των συναλλαγών στα υποψήφια blocks, τότε ολοκληρώνεται ο κορμός του υποψήφιου block και δημιουργείται η επικεφαλίδα του. Η επικεφαλίδα του block περιέχει (μεταξύ άλλων) πληροφορίες σχετικά με τον αριθμό των συναλλαγών που περιλαμβάνει, με τη θέση (συχνά αναφέρεται και ως ύψος) στην αλυσίδα όπου βρίσκεται το block, καθώς και το αποτέλεσμα της συνάρτησης του κατακερματισμού του προηγούμενου block στην αλυσίδα, για να δημιουργηθεί η σύνδεση μεταξύ τους (δείτε και Εικόνα 2.5). Επιπλέον, αναφέρεται και η τρέχουσα τιμή του επιπέδου δυσκολίας[18] για τη δημιουργία του block.

Αμέσως μετά ξεκινά η προσπάθεια εξόρυξης του block, με την επίλυση του μαθηματικού προβλήματος που θα παίξει ρόλο στη διαδικασία συναίνεσης, μέσω της τεχνικής Proof of Work (PoW). Η διαδικασία αυτή είναι ένας αγώνας ταχύτητας. Το κάθε block ξεκινά την προσπάθεια αυτή δοκιμάζοντας νέους συνδυασμούς από δεδομένα, τα οποία, αφού περάσουν από μια ειδική συνάρτηση κατακερματισμού[19] (στο Bitcoin η συνάρτηση αυτή είναι η SHA-256), βγάζουν ένα αποτέλεσμα. Το αποτέλεσμα αυτό συγκρίνεται με το τρέχον επίπεδο δυσκολίας για τη δημιουργία του block, πληροφορία που βρίσκεται στην επικεφαλίδα του block. Το επίπεδο δυσκολίας είναι ένας αριθμός που μεταβάλλεται από το δίκτυο για λόγους προστασίας του. Αν η τιμή που θα προκύψει από την κρυπτογράφηση είναι μέσα στο σύνολο των λύσεων που προέρχεται από το επίπεδο δυσκολίας, τότε η όλη διαδικασία είναι επιτυχημένη και ο κόμβος έχει ολοκληρώσει την εξόρυξη του (νέου) block.

Θα πρέπει να τονίσουμε ότι η διαδικασία της επίλυσης του PoW έχει μοναδικό αποτέλεσμα, το οποίο δεν μπορεί να προβλεφθεί. Επομένως, δεν υπάρχει τρόπος να συντομευτεί η διαδικασία. Θα πρέπει να εισαχθούν όλοι οι συνδυασμοί στη συνάρτηση κατακερματισμού και να παραχθεί το αποτέλεσμα που θα συγκριθεί με το σύνολο των αριθμών που ικανοποιούν το επίπεδο δυσκολίας.

---

[18] Το επίπεδο δυσκολίας ρυθμίζεται από το δίκτυο και αντανακλά πόσο μεγάλο είναι το σύνολο των αριθμών που επιλύουν το PoW. Όταν ο βαθμός δυσκολίας αυξάνεται, τότε το σύνολο των αριθμών μικραίνει, και αντίστροφα.

[19] Η συνάρτηση κατακερματισμού (hash function) είναι μια μαθηματική συνάρτηση η οποία δέχεται μια είσοδο οποιουδήποτε μεγέθους και παράγει μια έξοδο σταθερού μεγέθους που μπορεί να είναι από 32 έως 256 bits.

Ταυτόχρονα, όμως, η συνάρτηση κατακερματισμού παράγει πάντα το ίδιο αποτέλεσμα κάθε φορά που θα έχει την ίδια είσοδο. Επομένως, αν είναι γνωστή η είσοδος και το αποτέλεσμα, είναι εύκολο να επιβεβαιωθεί από τον οποιονδήποτε ότι όντως αυτή είναι η λύση του PoW.

Επόμενο βήμα, μετά την επιτυχή ολοκλήρωση της διαδικασίας του PoW, είναι *η διάδοση του νέου block σε όλο το δίκτυο*. Θα πρέπει να ενημερωθούν και οι υπόλοιποι κόμβοι ότι έχει βρεθεί το επόμενο block, να το επιβεβαιώσουν ελέγχοντας την επίλυση του PoW και, αμέσως μετά, να το βάλουν στην αλυσίδα και να ξεκινήσουν να δουλεύουν για την εύρεση του επόμενου. Για τον λόγο αυτό θα ξεκινήσουν πάλι να συλλέγουν συναλλαγές στο νέο block και να προχωρούν στην επίλυση του PoW για αυτό.

Κατά τη διάρκεια του σταδίου της ενημέρωσης είναι πιθανό το δίκτυο να βρεθεί σε μια κατάσταση που, με μικρή διαφορά χρόνου, δύο κόμβοι εξόρυξης έχουν επιλύσει σωστά τον αλγόριθμο του PoW και έχουν βρει το επόμενο block. Τα δύο αυτά blocks θα είναι και διαφορετικά μεταξύ τους, κατ' ελάχιστο στην πρώτη συναλλαγή. Στην περίπτωση αυτή, και για μικρό διάστημα, στο δίκτυό μας θα υπάρχουν κόμβοι που θα ενημερώνονται από τον έναν κόμβο και άλλοι που θα ενημερώνονται από τον άλλο κόμβο. Έτσι, η πληροφορία για την κατάσταση της αλυσίδας (και φυσικά του ledger) θα είναι διαφορετική, με αποτέλεσμα να δημιουργείται μια *διακλάδωση στην αλυσίδα του blockchain* (*blockchain fork*).

Η διαφορά που διαπιστώνεται πρέπει να επιλυθεί με κάποιον τρόπο από το ίδιο το δίκτυο, χωρίς την παρέμβαση κάποιου εξωτερικού παράγοντα. Στην αρχική του δημοσίευση ο Nakamoto (2008) αναφέρει τον τρόπο που θα επιλύονται οι διαφορές αυτές. Αναφέρει ότι το δίκτυο θα συγκλίνει στη μεγαλύτερη αλυσίδα. Δηλαδή θα επικρατήσει η εκδοχή εκείνη πάνω στην οποία θα χτιστεί πρώτα το επόμενο block, καθιστώντας αυτήν μεγαλύτερη από την άλλη. Επομένως, όποια από τις δύο εκδόσεις καταφέρει να αποτελέσει τη βάση πάνω στην οποία θα χτιστεί το νέο block αυτή, αυτομάτως, θα είναι και η επικρατέστερη.

Γίνεται αντιληπτό ότι η εξόρυξη μέσω της επίλυσης του νέου PoW δεν είναι το μοναδικό κριτήριο. Αντιθέτως, έχει σημασία και η ταχύτερη ενημέρωση του δικτύου για αυτό, για να αυξηθούν οι κόμβοι που προσπαθούν να προχωρήσουν στην επίλυση του επόμενου block πάνω από ένα από αυτά που έχουν δημιουργήσει τη διακλάδωση στην αλυσίδα.

Στατιστικά, αναφέρεται ότι διακλάδωση με 1 block συμβαίνει περίπου μία φορά κάθε ημέρα στο δίκτυο του Bitcoin. Αυτό αποδεικνύει ότι το δίκτυο επιτυγχάνει αποτελεσματικά τη σύγκλισή του.

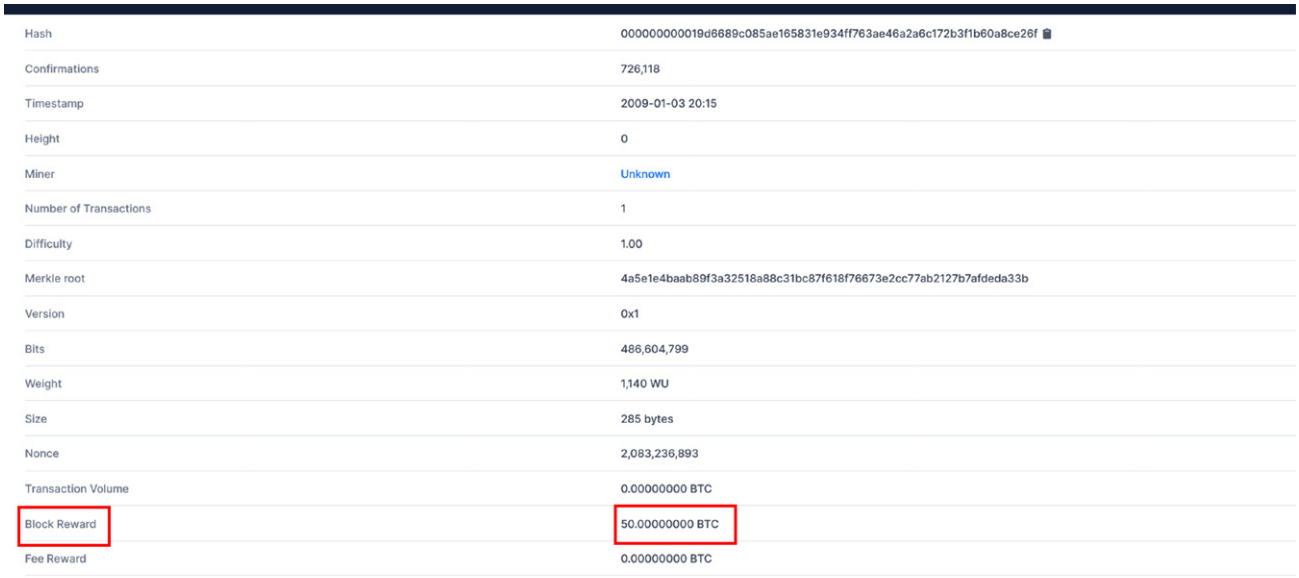
Στο δίκτυο του Bitcoin η εξόρυξη του επόμενου block πραγματοποιείται κατά μέσο όρο *κάθε 10 λεπτά*. Τόσος είναι ο μέσος χρόνος που απαιτείται για να επιλυθεί το δύσκολο μαθηματικό πρόβλημα που εισάγεται από τον αλγόριθμο συναίνεσης, τον PoW. Μάλιστα, ο βαθμός δυσκολίας για την εξόρυξη του επόμενου block μεταβάλλεται (ανά 2 εβδομάδες περίπου), έτσι ώστε ο μέσος χρόνος εξόρυξης να παραμένει στα 10 λεπτά. Αυτό προφανώς έχει επίπτωση και στον αριθμό των συναλλαγών ανά δευτερόλεπτο που μπορεί να επεξεργαστεί ένας κόμβος στο δίκτυο του Bitcoin. Η τιμή αυτή θεωρείται πως είναι πολύ χαμηλή (περίπου ίση με 7 tps για το Bitcoin).

Ο αριθμός αυτός θεωρείται μικρός για ένα σύστημα οικονομικών συναλλαγών, ιδιαίτερα αν συγκριθεί με τα υπάρχοντα συστήματα (π.χ. πιστωτικών καρτών) τα οποία είναι ικανά να διαχειριστούν χιλιάδες συναλλαγές το δευτερόλεπτο. Όμως, ο μηχανισμός του PoW για τη συναίνεση παίζει έναν πολύ σημαντικό ρόλο για το δίκτυο Bitcoin, καθώς συνεισφέρει στη δημιουργία ενός χρονικού περιθωρίου στο οποίο θα είναι δυνατή η σύγκλιση του δικτύου. Αυτό έχει ιδιαίτερη σημασία στην περίπτωση που υπάρχει διακλάδωση, όπως αναλύθηκε προηγουμένως.

Τέλος, όπως παρουσιάστηκε και πριν, σε κάθε υποψήφιο νέο block η πρώτη συναλλαγή είναι μοναδική σε κάθε block, καθώς αποτελεί και μια επιβράβευση του δικτύου στο miner για την εργασία του. Ο λόγος ύπαρξης αυτής της συναλλαγής ως πρώτης σε ένα υποψήφιο block είναι για την οικονομική επιβράβευση που προσφέρεται από το δίκτυο στους κόμβους (βλ. Ενότητα 1.4).

Στα δημόσια δίκτυα blockchain (όπως είναι και το Bitcoin) η μέθοδος αυτή έχει προτιμηθεί για να ενισχύσει τους κόμβους να συμπεριφέρονται σύμφωνα με τους κανόνες και να αποθαρρύνει κάθε προσπάθεια για επίθεση ή επιρροή στο δίκτυο. Έτσι, λοιπόν, ένας κόμβος επιβραβεύεται για την επιτυχή εξόρυξη του επόμενου block της αλυσίδας με την επιβράβευση νέων BTCs. Αρχικά, η επιβράβευση αυτή ήταν ίση με 50 BTC, αλλά έχει αποφασιστεί να μικραίνει στο μισό της κάθε 210.000 blocks που προστίθενται στην αλυσίδα. Αυτό υπολογίζεται ότι συμβαίνει περίπου κάθε τέσσερα χρόνια. Σήμερα (2022) η τιμή για την επιβράβευση έχει φθάσει να είναι ίση με 6,25 BTC.

Στην **Εικόνα 2.6** φαίνεται η επικεφαλίδα του πρώτου block που έγινε στο δίκτυο του Bitcoin, γνωστό και ως block *γέννησης* (*genesis*). Εκεί φαίνεται η ημερομηνία, καθώς και το ποσό της επιβράβευσης που, όπως αναφέρθηκε, αρχικά ήταν ίσο με 50 BTC.



Hash	000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Confirmations	726,118
Timestamp	2009-01-03 20:15
Height	0
Miner	Unknown
Number of Transactions	1
Difficulty	1.00
Merkle root	4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
Version	0x1
Bits	486,604,799
Weight	1,140 WU
Size	285 bytes
Nonce	2,083,236,893
Transaction Volume	0.00000000 BTC
Block Reward	50.00000000 BTC
Fee Reward	0.00000000 BTC

Εικόνα 2.6 *Η επικεφαλίδα του πρώτου block στην αλυσίδα του Bitcoin.*

Το block αυτό περιείχε μία και μόνη συναλλαγή, που αφορούσε την επιβράβευση του miner για τη δημιουργία του. Η **Εικόνα 2.7** δείχνει τη συναλλαγή αυτή.

Στο Κεφάλαιο 4 αναλύονται περισσότερα παραδείγματα συναλλαγών και blocks.

Εικόνα 2.7 *Η πρώτη συναλλαγή στο δίκτυο του Bitcoin.*

2.2.5 Ασφάλεια

Το Bitcoin παρουσιάζει σημαντική ασφάλεια, η οποία πηγάζει από την κατανεμημένη φύση του και από τη χρήση ισχυρής κρυπτογραφίας. Με τη βοήθεια της τεχνολογίας του blockchain το Bitcoin έχει βγάλει τον έλεγχο της ασφάλειας από έμπιστους ενδιάμεσους και τον έχει εναποθέσει στους χρήστες. Πλέον είναι οι χρήστες που με όπλο την κρυπτογραφία έχουν στην κατοχή τους τα κλειδιά των λογαριασμών τους και αναλαμβάνουν να δημιουργήσουν και να υπογράψουν συναλλαγές στο δίκτυο.

Επιπλέον, στο δίκτυο του Bitcoin οι συναλλαγές είναι ορατές σε όλους, καθώς δεν περιέχουν κάποια ευαίσθητη πληροφορία, όπως είναι, για παράδειγμα, ο αριθμός μιας πιστωτικής κάρτας, που μπορεί να υποκλαπεί. Επιπρόσθετα, μια συναλλαγή στο δίκτυο του Bitcoin δεν μπορεί να αλλαχθεί από τη στιγμή που μπήκε στο ledger, ενισχύοντας έτσι την αξιοπιστία του συστήματος. Ταυτόχρονα, έχει δοθεί στο δίκτυο του Bitcoin η δυνατότητα μια συναλλαγή να χρειάζεται έγκριση για να προχωρήσει από παραπάνω του ενός λογαριασμούς (ή υπογραφές), δημιουργώντας έτσι τις απαραίτητες συνθήκες ομοφωνίας πριν από την απόδοση στο δίκτυο.

Από την άλλη πλευρά, ο πολύ κρίσιμος ρόλος της κρυπτογραφίας και των κλειδιών σε ένα δίκτυο blockchain μπορεί να λειτουργήσει και αποτρεπτικά. Και αυτό γιατί εάν κάποιος χάσει τα κλειδιά του τότε αυτόματα γίνεται αδύνατη η πρόσβαση στο πορτοφόλι και, κατ' επέκταση, στα κρυπτονομίσματά του.

2.3 Ethereum

Η πλατφόρμα του Ethereum (2022) είναι η επόμενη ανοικτού κώδικα υλοποίηση της τεχνολογίας Blockchain, η οποία έτυχε ευρείας διάδοσης και αποδοχής. Στην ενότητα αυτή παρουσιάζονται τα βασικά χαρακτηριστικά της.

Σκοπός του Ethereum δεν ήταν να αποτελέσει ένα ηλεκτρονικό σύστημα πληρωμών, όπως το Bitcoin, αλλά να *ένα δημόσιο δίκτυο blockchain γενικού σκοπού*. Το δίκτυο αυτό θα μπορεί να προσομοιωθεί με έναν «παγκόσμιο κατανεμημένο υπολογιστή», τον οποίο θα μπορούσε να χρησιμοποιήσει κάποιος όχι μόνο για να ολοκληρώσει μια συναλλαγή αλλά και για ευρύτερη χρήση, όπως, για παράδειγμα, για την εκτέλεση προγραμμάτων (που ονομάζονται *έξυπνες συμβάσεις – smart contracts*), καθώς και αποκεντρωμένων εφαρμογών (γνωστών και ως *Decentralized Applications* ή *DApps*).

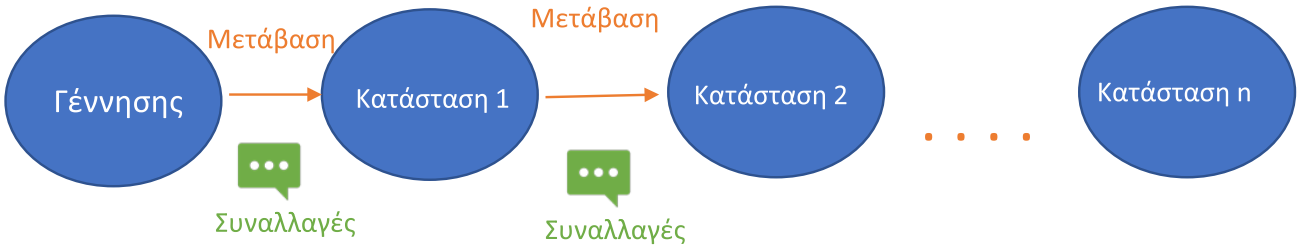
Για να μπορέσει, όμως, το Ethereum να λειτουργήσει ως ένας παγκόσμιος κατανεμημένος υπολογιστής, θα πρέπει να μπορεί να παρακολουθεί, εκτός από τις συναλλαγές που λαμβάνουν χώρα μεταξύ των χρηστών, και τη συνολική κατάσταση στην οποία βρίσκεται το σύστημα-μηχανή. Για την τρέχουσα κατάσταση θα πρέπει να συμφωνούν και να ανανεώνουν την πληροφορία τους όλοι οι κόμβοι-μέλη του κατανεμημένου δικτύου μετά την κάθε μετάβαση.

Για να μπορέσει να πετύχει τη συναίνεση, τον συγχρονισμό και την αποθήκευση των αλλαγών κατάστασης του συστήματος, το Ethereum χρησιμοποιεί ως μέσο την τεχνολογία του blockchain και ως εργαλείο το κρυπτονόμισμα ether. Οι αλλαγές των καταστάσεων μπαίνουν σε διαφορετικά, διαδοχικά blocks στην αλυσίδα, ενώ το κρυπτονόμισμα (ether ή gas) έχει σκοπό να χρηματοδοτεί τη δυνατότητα χρήσης των προγραμμάτων (smart contracts) αυτών και την ανταλλαγή του ανάμεσα στα μέλη του δικτύου.

Με τον όρο *κατάσταση* στο Ethereum εννοείται μια μεγάλη δομή δεδομένων που περιέχει τις συναλλαγές που λαμβάνουν χώρα και τις μεταβολές στους λογαριασμούς των χρηστών σύμφωνα με αυτές. Δεν αφορούν, όμως, όλες οι συναλλαγές τη μεταφορά ποσών ανάμεσα σε δύο χρήστες, αλλά και τη χρήση ether[20] για την εκτέλεση εντολών σε προγράμματα που έχουν αποθηκευτεί στο δίκτυο (smart contracts).

Όσον αφορά τα προγράμματα αυτά, επειδή δεν είναι πρακτική η συγγραφή νέου κώδικα κάθε φορά που απαιτείται μια ενέργεια, το Ethereum επιτρέπει την εγκατάσταση των προγραμμάτων στο δίκτυο και την κλήση των κατάλληλων συναρτήσεών τους κάθε φορά που απαιτείται. Επειδή όμως οι κλήσεις αυτές θα φέρουν αλλαγές στην κατάσταση μηχανής στον κόμβο που θα τις εκτελέσει, θα πρέπει τα προγράμματα αυτά να είναι εγκατεστημένα σε όλους τους κόμβους. Έτσι, θα είναι δυνατή η επαλήθευση των αλλαγών αυτών από όλους τους κόμβους-μέλη του δικτύου και η συμφωνία για τη νέα κατάσταση του συστήματος.

Με αυτή τη λογική, το Ethereum θεωρείται μια δημόσια, *κατανεμημένη μηχανή καταστάσεων* (*distributed state machine*). Σε αυτήν αποθηκεύονται οι μεταβολές στις καταστάσεις, όπως αναφέρονται στις συναλλαγές στα blocks, σε ένα ολοένα αυξανόμενο εδάφιο (ledger) που διαμοιράζεται μεταξύ των κόμβων του δικτύου, όπως φαίνεται στην **Εικόνα 2.8**.



Εικόνα 2.8 *Η εναλλαγή των καταστάσεων του δικτύου βάσει των συναλλαγών που τρέχουν σε κάθε βήμα.*

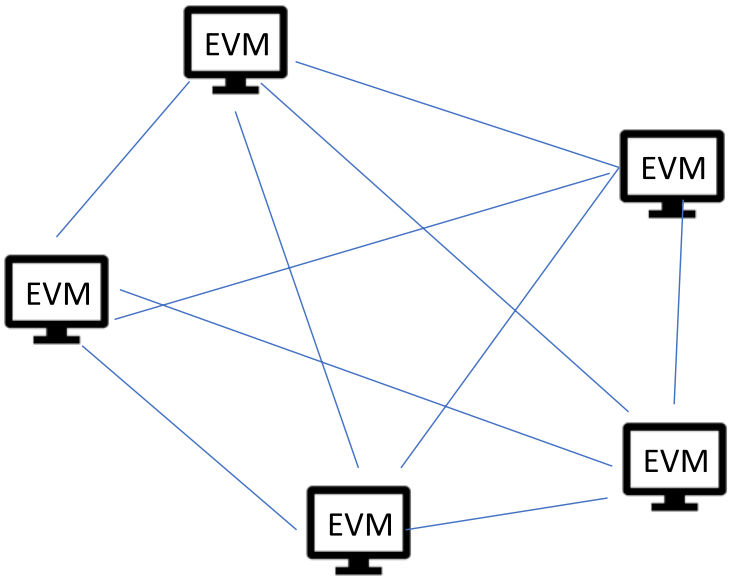
Στη συνέχεια αναφέρονται περισσότερες λεπτομέρειες σχετικά με τα βασικά χαρακτηριστικά του Ethereum, όπως είναι: η Εικονική Μηχανή του Ethereum (EVM), η χρήση του ether και του gas, τα πορτοφόλια, τα έξυπνα συμβόλαια, οι αποκεντρωμένες εφαρμογές κ.ά.

[20] Το όνομα του κρυπτονομίσματος στο Ethereum.

2.3.1 Η Εικονική Μηχανή του Ethereum (Ethereum Virtual Machine, EVM)

Ο τρόπος με τον οποίο είναι δυνατή η παρακολούθηση και επιβεβαίωση των αλλαγών στη κατάσταση του συστήματος στο δίκτυο του Ethereum είναι μέσω της δημιουργίας και της εκτέλεσης της *Εικονικής Μηχανής του Ethereum* (*Ethereum Virtual Machine* ή *EVM*).

Η EVM είναι μια μηχανή υπολογισμού που δημιουργεί ένα περιβάλλον που απεικονίζει ένα στιγμιότυπο της κατάστασης του δικτύου του Ethereum. Τρέχει σε χιλιάδες κόμβους ταυτόχρονα (**Εικόνα 2.9**), οι οποίοι τη χρησιμοποιούν για να μπορέσουν να επιβεβαιώσουν τις αλλαγές στην (κοινή) κατάσταση των μηχανών του συστήματος που προκαλούνται από τις συναλλαγές των χρηστών και από την εκτέλεση των smart contracts.

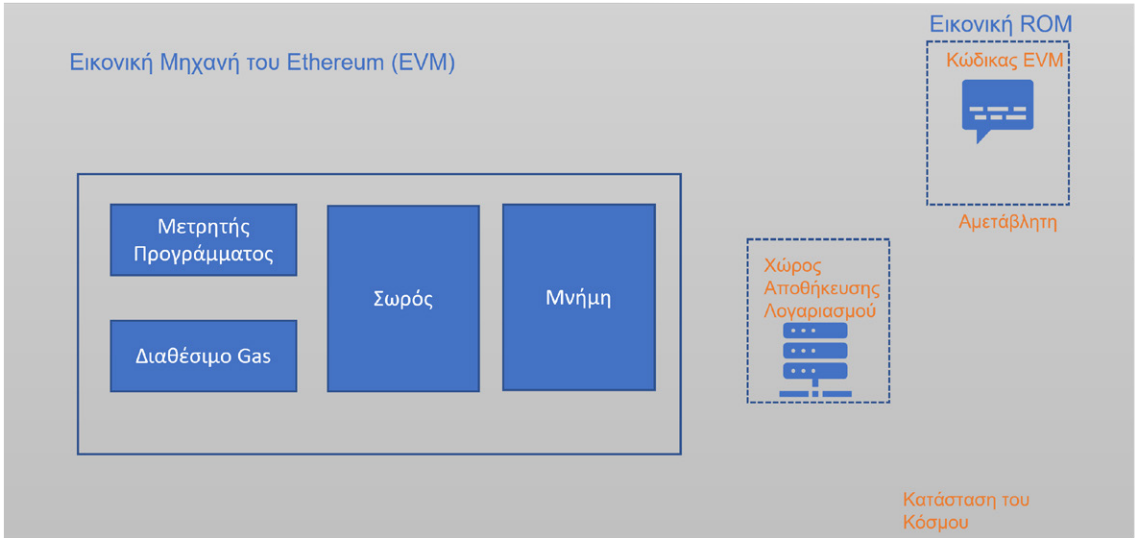


Εικόνα 2.9 *Το δίκτυο του Ethereum.*

Ειδικά για τα smart contracts, το περιβάλλον EVM είναι εκείνο που επιτρέπει την εκτέλεση κώδικα με τη μετατροπή όλων των εντολών σε μορφή bytecodes. Η μορφή αυτή επιτρέπει την εκτέλεση του κώδικα μέσα στην EVM από όλους τους κόμβους του δικτύου.

Πιο αναλυτικά, επιτρέπεται με τη μετάφραση σε bytecodes στα smart contacts να αξιοποιήσουν τη στοίβα (stack) που δημιουργείται μέσα στην EVM για την αποθήκευση των λέξεων μεγέθους 256 bits, διευκολύνοντας την εκτέλεση των προγραμμάτων που έχουν εγκατασταθεί.

Στην **Εικόνα 2.10** φαίνεται και μια απεικόνιση της αρχιτεκτονικής της EVM.



Εικόνα 2.10 *Η αρχιτεκτονική της EVM.*

Θα πρέπει να τονιστεί σχετικά με την EVM ότι ως σύστημα κανόνων διαχείρισης δεδομένων επιτυγχάνει *πληρότητα κατά Turing* (*Turing completeness*)[21]. Αυτό σημαίνει ότι η EVM μπορεί να προσομοιώσει μια μηχανή Turing, δηλαδή να προσομοιώνει τη λογική οποιουδήποτε αλγόριθμου. Πλήρεις κατά Turing θα πρέπει να είναι, κατ' επέκταση, και οι γλώσσες προγραμματισμού που χρησιμοποιούνται για τη σύνταξη των smart contracts, που είναι τα προγράμματα που τρέχουν μέσα στην EVM.

Οι περισσότερες σύγχρονες γλώσσες προγραμματισμού είναι πλήρεις κατά Turing και το ίδιο συμβαίνει και με τις γλώσσες που αναπτύχθηκαν για τη σύνταξη και εκτέλεση smart contracts σε υλοποιήσεις blockchain.

Όμως, αν και η πληρότητα κατά Turing εμπεριέχει τη δυνατότητα εκτέλεσης όλων των προγραμμάτων, δεν είναι ποτέ σίγουρο ότι τα προγράμματα θα ολοκληρωθούν κατά την εκτέλεσή τους, ακόμα και αν η προσομοίωσή τους είναι επιτυχής. Έτσι, υπάρχει ο κίνδυνος ένα πρόγραμμα σε έναν κόμβο να διακοπεί για κάποιον λόγο ή να μπει σε έναν ατέρμονα βρόχο επανάληψης. Αυτή η κατάσταση θα μπορούσε να αντιμετωπιστεί αν το πρόγραμμα εκτελούνταν σε έναν υπολογιστή, αλλά είναι πολύ δύσκολο να αντιμετωπιστεί αν βρεθεί σε αυτήν ένα δημόσιο δίκτυο blockchain.

Το δίκτυο του Ethereum χρησιμοποιεί το κρυπτονόμισμά του (ether) για να αποφύγει αυτές τις καταστάσεις, όπως θα φανεί και στη συνέχεια.

2.3.2 Έξυπνες συμβάσεις (smart contracts)

Με τον όρο *έξυπνες συμβάσεις* (*smart contracts*) στο δίκτυο του Ethereum εννοούνται τα προγράμματα εκείνα που αποθηκεύονται και εκτελούνται με ντετερμινιστικά αποτελέσματα μέσα στην EVM, ως μέλη του πρωτοκόλλου του δικτύου του Ethereum (Antonopoulos & Wood, 2019).

Αυτά τα προγράμματα/συμβάσεις γράφονται σε γλώσσα προγραμματισμού υψηλού επιπέδου, όπως είναι η Solidity (2022), η οποία και έχει δημιουργηθεί για τον λόγο αυτόν. Τα smart contracts για να εκτελεστούν θα πρέπει να μετατραπούν σε bytecodes, για να μπορούν, στη μορφή αυτή, να τρέξουν μέσα στην EVM. Άλλες γνωστές γλώσσες προγραμματισμού που χρησιμοποιούνται για τη συγγραφή smart contracts σε υλοποιήσεις blockchain είναι οι γλώσσες Rust (2022), JavaScript (2022), Vyper (2022) και Yul.

Στο Ethereum η χρήση της Solidity είναι πολύ διαδεδομένη, όπως φαίνεται από πρόσφατη έρευνα (Mix, 2019) και είναι και η γλώσσα που χρησιμοποιείται στα παραδείγματα του βιβλίου.

Τα smart contracts συντάσσονται με τη βοήθεια κάποιου ειδικού προγράμματος συγγραφής (π.χ. για την ανάπτυξη προγραμμάτων σε Solidity χρησιμοποιείται ευρέως το Remix[22]), προτού ακολουθήσει η εγκατάστασή τους στο δίκτυο του Ethereum (ή σε ένα δοκιμαστικό δίκτυο). Για το βήμα της εγκατάστασης ενός smart contract στο δίκτυο πραγματοποιείται μια ειδική συναλλαγή προς μια συγκεκριμένη διεύθυνση, την 0x0, η οποία και είναι η διεύθυνση δημιουργίας συμβάσεων στο δίκτυο του Ethereum.

Με την ένταξή τους στο δίκτυο, τα smart contracts αποκτούν πλέον τη δική τους διεύθυνση, η οποία και χρησιμοποιείται είτε ως αποδέκτης σε συναλλαγές για την αποστολή χρημάτων προς το smart contract (συνήθως για αποθήκευση σε αυτό) είτε κατά την κλήση συναρτήσεων του smart contract. Είναι σημαντικό να τονιστεί ότι το πρόγραμμα τρέχει *μόνο* εφόσον κληθεί κάποια συνάρτησή του μέσω της απαραίτητης συναλλαγής στη διεύθυνση του contract.

Το ιδιωτικό κλειδί της διεύθυνσης του smart contract δεν γίνεται γνωστό σε κανέναν, ούτε στον δημιουργό του προγράμματος. Οπότε, στην πραγματικότητα, το κλειδί αυτό ανήκει στο ίδιο το smart contract. Άμεσο αποτέλεσμα αυτού είναι και το ότι τα smart contracts δεν μπορούν να ξεκινήσουν μια συναλλαγή. Αντιθέτως, μπορούν να αντιδρούν σε συναλλαγές καλώντας με τη σειρά τους άλλα smart contracts, δημιουργώντας με τον τρόπο αυτό σύνθετες διαδρομές εκτέλεσης.

Ο κύκλος της ζωής ενός smart contract περιλαμβάνει τη συγγραφή του, την εγκατάστασή του στο δίκτυο (με την εκτέλεση της ειδικής συναλλαγής που αναφέρθηκε προηγουμένως προς τη διεύθυνση δημιουργίας συμβάσεων) και την εκτέλεσή του σε αυτό. Εάν πρέπει να υπάρξει η οποιαδήποτε διόρθωση ή αναβάθμισή του, τότε θα πρέπει να εγκατασταθεί από την αρχή νέα έκδοση και να ρυθμιστούν όλες οι αποκεντρωμένες εφαρμογές να δείχνουν στη νέα διεύθυνση του contract.

---

[21] Το όνομα έχει προέλθει από τον Alan Turing, τον επιστήμονα με ιδιαίτερη συνεισφορά στον τομέα της θεωρίας υπολογισμού και της τεχνητής νοημοσύνης, με ιδιαίτερη δράση στην αποκωδικοποίηση των κρυπτογραφημένων μηνυμάτων των Γερμανών κατά τη διάρκεια του Β΄ Παγκόσμιου Πολέμου.

[22] Online Σύνδεσμος: https://remix-project.org/

Αυτό συμβαίνει γιατί τα smart contracts εγγράφονται στο ledger του blockchain και, όπως έχει αναφερθεί και στο Κεφάλαιο 1, είναι αδύνατη η οποιαδήποτε αλλαγή στο περιεχόμενο αυτού. Επομένως χρειάζεται να επαναληφθεί όλη η διαδικασία για κάθε αλλαγή που πρέπει να ενσωματωθεί.

Σαν ένα επιπλέον χαρακτηριστικό, το δίκτυο του Ethereum επιτρέπει την εκτέλεση της εντολής *selfdestruct*, με την οποία ένα smart contract δεν σβήνεται, αλλά σταματά να δέχεται συναλλαγές στη διεύθυνσή του. Είναι δυνατόν να αναζητηθούν στο δίκτυο συναλλαγές και ενέργειες όπου συμμετείχε το contract στο παρελθόν, αλλά δεν επιτρέπεται να υπάρξουν νέες στο μέλλον.

Σημαντικό ρόλο στην εκτέλεση ενός smart contract παίζει και η χρήση του ether, του κρυπτονομίσματος του Ethereum. Για την εγκατάσταση ενός smart contract στο δίκτυο του Ethereum είναι υποχρεωτικό να πληρωθεί ένα ποσό σε gas, το οποίο είναι το αντίτιμο που πρέπει να αποδοθεί και το οποίο μπορεί κάποιος να το πληρώσει μόνο σε ether (ή gwei, όπως συχνά αναφέρεται). Πέρα από την εγκατάσταση όμως για την κλήση και εκτέλεση μιας συνάρτησης στο smart contract, πάλι θα πρέπει να αποδοθεί ένα ποσό από την εφαρμογή για την ολοκλήρωση της συναλλαγής.

Ο λόγος που χρησιμοποιείται το gas στο Ethereum είναι διπλός:

α) για να εμποδιστεί η χωρίς λόγο χρήση των πόρων του δικτύου, αλλά και

β) για να σταματήσει η εκτέλεση ενός προγράμματος που έχει πέσει σε έναν ατέρμονα βρόχο επανάληψης, ξοδεύοντας άσκοπα τους πόρους αυτούς.

Να σημειωθεί ότι στην κατάσταση (α) μπορεί να βρεθεί ένα contract σκοπίμως από τον δημιουργό του, ενώ στη (β) μπορεί να βρεθεί από λογικό λάθος στον κώδικα του contract.

Προηγουμένως αναφέρθηκε ότι τα smart contracts είναι προγράμματα που ακολουθούν την πληρότητα κατά Turing. Ωστόσο, επειδή υπάρχει αβεβαιότητα ως προς την τελική εκτέλεση ενός προγράμματος προτού αυτό ολοκληρωθεί, το δίκτυο του Ethereum αποφάσισε να θωρακιστεί από πιθανές περιπτώσεις που όντως ένα πρόγραμμα εισέρχεται σε έναν ατέρμονα βρόχο.

Για τον λόγο αυτόν έχει εισαχθεί στο Ethereum η χρήση του gas (βλ. Υποενότητα 2.3.4), έτσι ώστε κάποιος να πληρώνει για την εκτέλεση των εντολών και, αν τα χρήματα δεν φθάσουν (π.χ. γιατί μπήκε το πρόγραμμα σε έναν τέτοιο ατέρμονα βρόχο), τότε να μην ολοκληρώνεται η συναλλαγή και η κατάσταση του συστήματος να επανέρχεται σε αυτήν όπου βρισκόταν πριν από την κλήση αυτή. Είναι φανερό, επομένως, ότι η χρήση του gas έχει μπει στο δίκτυο του Ethereum για να μη χρειαστεί η επανεκκίνηση του δικτύου σε περίπτωση προβλήματος κατά την εκτέλεση ενός smart contract.

Τέλος, ως συμπέρασμα μπορεί να ειπωθεί ότι τα smart contracts αποτέλεσαν μια πολύ σημαντική προσθήκη, καθώς έδωσαν τη δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν τα χαρακτηριστικά του blockchain για να τρέξουν προγράμματα αξιόπιστα, επεκτείνοντας τη χρήση της τεχνολογίας και σε τομείς όπως: διαχείριση της εφοδιαστικής αλυσίδας, δημιουργία παιχνιδιών και ψηφιακής ταυτοποίησης (βλ. Κεφάλαιο 9).

2.3.3 Πορτοφόλια (Wallets)

Όπως και στο Bitcoin, όπου για τη σύνδεση με το δίκτυο είναι απαραίτητη η απόκτηση διεύθυνσης και ενός ζεύγους ιδιωτικού/δημόσιου κλειδιού, έτσι και στο Ethereum είναι απαραίτητο το στάδιο αυτό. Τα ζεύγη των κλειδιών φυλάσσονται σε πορτοφόλια και χρησιμοποιούνται για να δώσουν πρόσβαση σε κρυπτονομίσματα ή *κρυπτο-συλλεκτικά αντικείμενα* (*crypto-collectibles*) που έχει στην κατοχή του ο ιδιοκτήτης ως απόδειξη ιδιοκτησίας. Επιπλέον, χρησιμοποιούνται για την υπογραφή των συναλλαγών, όπως συμβαίνει και στο Bitcoin. Ένα πορτοφόλι που συνδέεται με το δίκτυο του Ethereum γενικά είναι σε θέση να εκτελεί συναλλαγές με άλλους λογαριασμούς, αποστέλλοντας ethers ή με smart contracts, ακόμα και με tokens που έχουν δημιουργηθεί και έχει στην κατοχή του (βλ. Κεφάλαιο 7). Βασική προϋπόθεση για την όποια συναλλαγή είναι η δυνατότητα αγοράς gas (μέσω των διαθέσιμων ethers) για να μπορέσει να πραγματοποιηθεί αυτή.

Ένα από τα πλέον γνωστά πορτοφόλια, το οποίο είναι άμεσα συνδεδεμένο με το δίκτυο του Ethereum, είναι και το Metamask[23]. Πρόκειται για ένα πρόγραμμα το οποίο μπορεί να εγκατασταθεί στον υπολογιστή ως επέκταση (add-on) σε έναν σύγχρονο browser (Chrome, Firefox, Edge) ή στο κινητό, όπου είναι διαθέσιμο για τα δύο βασικά λειτουργικά (iOS και Android). Οδηγίες για την εγκατάσταση του πορτοφολιού Metamask (σε περίπτωση που δεν το έχετε ήδη) θα βρείτε στο Παράρτημα Α στο τέλος του βιβλίου.

[23] Online Σύνδεσμος: https://metamask.io/

Ο λογαριασμός στο Metamask δίνει πρόσβαση στα κλειδιά του χρήστη, που του/της επιτρέπουν να αποκτήσει διεύθυνση και να πραγματοποιεί συναλλαγές στο δίκτυο του Ethereum και των δοκιμαστικών δικτύων αυτού. Για περισσότερες οδηγίες σχετικά με την εγκατάσταση του Metamask δείτε το Παράρτημα Α.

Μέσω του Metamask δίνεται η δυνατότητα σύνδεσης πρωτίστως με το δίκτυο του Ethereum αλλά και με ένα σύνολο από δοκιμαστικά δίκτυά του, όπως είναι τα Ropsten, Kovan, Rinkeby και Goerli. Τα δίκτυα αυτά μπορούν να χρησιμοποιηθούν για δοκιμαστικούς σκοπούς, όπως, για παράδειγμα, για την εγκατάσταση ενός smart contract και για έλεγχό του σε ένα πραγματικό δοκιμαστικό περιβάλλον ανάπτυξης. Το κάθε δίκτυο, μάλιστα, έχει το δικό του κρυπτονόμισμα, το οποίο όμως δεν έχει πραγματική αξία και υπάρχει μόνο για την εξυπηρέτηση των ενεργειών στο δίκτυο αυτό, σε αντιστοιχία με τη χρήση του gas στο Ethereum.

Επομένως, για να μπορέσει κάποιος να προχωρήσει σε μια εγκατάσταση ενός smart contract σε ένα δοκιμαστικό δίκτυο, θα πρέπει να αποκτήσει ένα ποσό από το κρυπτονόμισμα που υποστηρίζει. Αυτό είναι εύκολο, καθώς σε κάθε δοκιμαστικό δίκτυο λειτουργεί μια υπηρεσία (ονομάζεται faucet) που επιτρέπει σε κάποιον να του ζητήσει ένα σταθερό ποσό από το κρυπτονόμισμα, δίνοντας τη διεύθυνσή του στο Metamask για την ολοκλήρωση της συναλλαγής. Πρόσβαση στην υπηρεσία αυτή μπορεί να έχει κάποιος και μέσα από το Metamask πατώντας στην επιλογή Buy και επιλέγοντας Test faucet. Για να αποφευχθεί η αλόγιστη χρήση της υπηρεσίας αυτής, επιτρέπεται συνήθως η χρήση ανά διεύθυνση μόνο μία φορά μέσα στην ημέρα.

Με τον τρόπο αυτόν είναι δυνατόν να αποκτήσει κάποιος το απαραίτητο ποσό που χρειάζεται για την εγκατάσταση και δοκιμή ενός smart contract σε ένα από τα δοκιμαστικά δίκτυα του Ethereum.

Θα πρέπει να σημειωθεί ότι η χρήση παρόμοιας υπηρεσίας είναι συχνή, ιδιαίτερα σε δίκτυα blockchain που δεν χρησιμοποιούν πραγματική αξία χρημάτων αλλά είναι δοκιμαστικά ή ιδιωτικά. Ένα τέτοιο παράδειγμα αποτελεί και η κοινοπραξία Bloxberg[24], η οποία έχει δημιουργηθεί με σκοπό να προωθήσει την ερευνητική δράση χρησιμοποιώντας ένα δίκτυο blockchain για να αποδείξει την ιδιοκτησία ερευνητικών εργασιών. Μέσω της δημιουργίας συνεργασιών αναμένεται να προωθήσει την αποδοχή και χρήση της τεχνολογίας σε ερευνητικά και ακαδημαϊκά ιδρύματα σε όλο τον κόσμο.

Κλείνοντας, είναι δυνατή η σύνδεση του Metamask και με ιδιωτικά δίκτυα που έχουν δημιουργηθεί από έναν χρήστη, αρκεί αυτά να έχουν ως βάση τους το δίκτυο του Ethereum.

Για περισσότερες πληροφορίες για τα είδη και τη χρήση των πορτοφολιών σε ένα δίκτυο blockchain δείτε το Κεφάλαιο 3.

2.3.4 Ether – Gas

Ether είναι το όνομα του κρυπτονομίσματος στο δίκτυο του Ethereum. Συχνά συναντάται και με τη συντομογραφία «ETH».

Υπάρχουν πολλές υποδιαιρέσεις του 1 ETH με τη μικρότερη να ονομάζεται wei. Ισχύει ότι 1 wei = 10^{-18} ETH ή αλλιώς 1 ETH = 10^{18} wei = 1.000.000.000.000.000.000 wei. Το wei αποτελεί τη συνηθέστερη βάση για τις συναλλαγές στο δίκτυο του Ethereum.

Στον **Πίνακα 2.1** φαίνονται οι υποδιαιρέσεις του ETH και τα ονόματά τους.

Τιμή (σε Wei)	Εκθέτης	Ονομασία	Ονομασία σε SI
1	1	Wei	Wei
1.000	10^3	Babbage	Kilo Wei
1.000.000	10^6	Lovelace	Mega Wei
1.000.000.000	10^9	Shannon	Giga Wei
1.000.000.000.000	10^{12}	Szabo	Microether
1.000.000.000.000.000	10^{15}	Finney	Milliether
1.000.000.000.000.000.000	10^{18}	Ether	Ether
1.000.000.000.000.000.000.000	10^{21}	Grand	Kiloether
1.000.000.000.000.000.000.000.000	10^{24}	-	Megaether

Πίνακας 2.1 *Οι υποδιαιρέσεις του ether (ETH).*

[24] Online Σύνδεσμος: https://bloxberg.org/

Το κρυπτονόμισμα χρησιμοποιείται σε συναλλαγές μεταξύ χρηστών (αντίστοιχα με το Bitcoin) αλλά και για την εγκατάσταση ή/και εκτέλεση smart contracts. Τέλος, μπορεί να χρησιμοποιηθεί για την αγορά tokens, όπως είναι, για παράδειγμα, τα ERC-20[25] (βλ. Κεφάλαιο 7).

Επιπλέον, όμως, το ETH χρησιμοποιείται και για την αγορά gas. Το gas αποτελεί μια ξεχωριστή μονάδα μέτρησης, αποτελεί ένα κρυπτονόμισμα του Ethereum που αφορά την πληρωμή του δικτύου για την επεξεργασία και εκτέλεση των συναλλαγών, καθώς και για την αποθήκευση των δεδομένων στους πόρους του. Το όνομά του παραπέμπει στη βενζίνη που χρειάζεται ένα όχημα για να κινηθεί. Όσο υπάρχει, το όχημα κινείται. Έτσι και στο Ethereum, όσο υπάρχουν ethers και αγοράζει κάποιος gas, τόσο μπορεί να εκτελέσει ενέργειες στο δίκτυο. Για παράδειγμα, αν θέλει κάποιος να κάνει μια συναλλαγή, θα αναγκαστεί να χρησιμοποιήσει πόρους του δικτύου (επεξεργασία, αποθήκευση), οπότε θα πρέπει να πληρώσει ένα (μικρό) αντίτιμο για αυτό. Το αντίτιμο αυτό είναι το gas και το πληρώνει σε ETH. Για την ακρίβεια, η τιμή του gas δίνεται σε μια υποδιαίρεση του wei (Πίνακας 2.1), που είναι τα gwei. Επίσης, οι miners[26] είναι αυτοί που αποφασίζουν για την τιμή του gas βάσει της ζήτησης για επεξεργαστική ισχύ στο δίκτυο. Οπότε, η τιμή του gas είναι δυναμική.

Τέλος, ο χρήστης που πραγματοποιεί μια συναλλαγή μπορεί να δηλώσει μέχρι πόσο gas είναι διατεθειμένος να ξοδέψει για την ολοκλήρωσή της, συμπληρώνοντας το πεδίο *Gas Limit* (π.χ. στο πορτοφόλι του στο Metamask, από το οποίο δημιουργεί τη συναλλαγή). Αναλόγως με την τιμή που θα συμπληρωθεί στο πεδίο θα επηρεαστεί και το αποτέλεσμα της συναλλαγής.

Μια μεγάλη τιμή μπορεί να βοηθήσει τη συναλλαγή να μπει σε block πιο γρήγορα, καθώς έχει περισσότερες πιθανότητες να επιλεγεί από minters (λόγω της αμοιβής). Αν δοθούν πιο λίγα, τότε μπορεί να παραμείνει στην αναμονή περισσότερο. Επίσης, αν μια συναλλαγή χρειάζεται περισσότερο gas από αυτό που είναι διαθέσιμο από τον χρήστη, ακυρώνεται.

Έτσι, αν για ένα smart contract χρειάζεται περισσότερο gas για να ολοκληρωθεί η κλήση μιας συνάρτησής του, τότε το smart contract θα επιστρέψει στην κατάσταση που ήταν πριν από την κλήση και θα σταματήσει εκεί. Ταυτόχρονα, όμως, ο χρήστης θα χάσει το gas που έδωσε. Αυτό λειτουργεί ως αποτρεπτικό μέτρο στην απασχόληση των minters με ενέργειες που δεν επιβραβεύονται αρκετά. Αυτό είναι κάτι που θα μπορούσε, σε αντίθετη περίπτωση, να χρησιμοποιηθεί και ως ένα είδος επίθεσης στο δίκτυο. Με το μέτρο αυτό αυξάνεται σημαντικά με τον χρόνο το κόστος για την πραγματοποίηση μιας τέτοιας επίθεσης.

2.3.5 Μάρκα/Διακριτικό (Token)

Οι *μάρκες* (ή *διακριτικά* ή *tokens*) αποτελούν ένα σημαντικό χαρακτηριστικό του δικτύου του Ethereum, αλλά γενικά και άλλων δικτύων blockchain (π.χ. Solana[27]). Αρχικά, η έννοια του token είχε ταυτιστεί αποκλειστικά με την έννοια του κρυπτονομίσματος, καθώς τέτοια ήταν και η χρήση των tokens στην αρχή. Χαρακτηριστικό παράδειγμα αποτελεί το Bitcoin, το οποίο και αποτελεί ένα είδος token. Όμως με την εμφάνιση του δικτύου Ethereum και την ευρεία χρήση των tokens σε νέα παραδείγματα εφαρμογής που έφερε αυτό, μέσω της προτυποποίησης της δημιουργίας τους (βλ. Κεφάλαιο 7), έγινε δυνατή μια ευρύτερη χρήση και αξιοποίησή τους. Στη συνέχεια ακολουθεί μια ανάλυση των χαρακτηριστικών που μπορεί να έχει μια υλοποίηση ενός token.

Ξεκινώντας από τον πραγματικό κόσμο, τα tokens είναι περισσότερο γνωστά για την εφαρμογή τους σε συγκεκριμένες περιπτώσεις χρήσης, εξυπηρετώντας έναν συγκεκριμένο σκοπό. Η χρήση τους, κυρίως, γίνεται για να αποδειχθεί η άδεια πρόσβασης του κατόχου ενός token σε μια περιοχή ή ως απόδειξη σε οικονομικές συναλλαγές, για την επιβεβαίωση ότι έχει αποδοθεί το αντίτιμο για την αγορά του token. Στην τελευταία περίπτωση, συνηθισμένο παράδειγμα συναντάται σε συναυλιακούς χώρους, όπου η πληρωμή για την αγορά ενός προϊόντος (π.χ. αναψυκτικό, νερό κτλ.) αποδεικνύεται με την απόκτηση μιας μάρκας σε ειδικά ταμεία, η οποία μάρκα επιστρέφεται στην καντίνα με την παραλαβή του προϊόντος κατανάλωσης.

Αντίθετα, στον κόσμο του blockchain τα tokens αποτελούν μια πιο αφηρημένη έννοια. Ένα token σε ένα δίκτυο blockchain μπορεί να αντιπροσωπεύει πολλά διαφορετικά πράγματα, όπως: τα δικαιώματα ενός χρήστη (π.χ. για μια συνδρομή ή για τη συμμετοχή του σε μια εκλογική διαδικασία), την ταυτότητά του (π.χ. με τη μορφή

---

[25] ERC είναι τα αρχικά από τις λέξεις Ethereum Request for Comments και πρόκειται για τεχνικές οδηγίες προς τους δημιουργούς υλικού στο δίκτυο του Ethereum. Οι οδηγίες αυτές περιλαμβάνουν συμφωνίες για προδιαγραφές και πρότυπα για συμβάσεις.

[26] Minters στο Ethereum.

[27] Online Σύνδεσμος: https://solana.com/

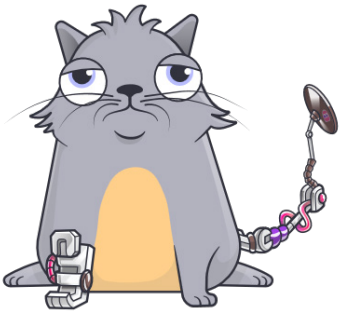
ενός avatar στον ψηφιακό κόσμο ή ως ψηφιακό «αντίγραφο» του διαβατηρίου του), ένα κρυπτονόμισμα που έχει ένα καθορισμένο από τους συμμετέχοντες οικονομικό αντίτιμο, ακόμα και την ιδιοκτησία ενός αντικειμένου.

Στην πράξη, όμως, τα tokens σε ένα δίκτυο blockchain είναι αποτελέσματα εκτέλεσης προγραμμάτων τα οποία τρέχουν με τη μορφή smart contracts και τα οποία δημιουργούν ψηφιακά αντικείμενα που μπορεί να έχουν οποιαδήποτε από τις προαναφερθείσες χρήσεις (και όχι μόνο).

Επιστρέφοντας στην εξαιρετικά κοινή και δημοφιλή περίπτωση χρήσης ενός token για τη δήλωση της ιδιοκτησίας ενός αντικειμένου, το αντικείμενο που συμβολίζεται με ένα token είναι δυνατό να υπάρχει μόνο στον ψηφιακό κόσμο του blockchain ή να υπάρχει στον φυσικό κόσμο και να περνά στον ψηφιακό για την καλύτερη διαχείρισή του.

Τα tokens που υπάρχουν μόνο στον ψηφιακό κόσμο και τα οποία ακολουθούν τους κανόνες του δικτύου blockchain (π.χ. συναίνεσης) μέσα στο οποίο γεννήθηκαν και συναλλάσσονται ονομάζονται *κρυπτο-συλλεκτικά αντικείμενα* ή *crypto-collectibles*.

Παραδείγματα αποτελούν τα crypto-Kitties (βλ. Υποενότητα 1.3.3 και **Εικόνα 2.11**), τα οποία είναι ψηφιακά αντικείμενα που δεν έχουν κάποια αντιπροσώπευση στον πραγματικό κόσμο, αλλά έχουν και ανήκουν στον ιδιοκτήτη τους στο δίκτυο του Ethereum.



Εικόνα 2.11 *Ένα crypto-Kitty (Πηγή εδώ).*

Τα tokens αυτά μπορούν να πωληθούν σε όποιον δώσει το μεγαλύτερο αντίτιμο σε ether. Επιπλέον, είναι δυνατόν να συνδυαστούν για τη δημιουργία νέων με πιο σπάνια χαρακτηριστικά, αυξάνοντας έτσι την αξία τους στον ψηφιακό κόσμο.

Εναλλακτικά, ένα token μπορεί να αντιπροσωπεύει ένα αντικείμενο του φυσικού κόσμου, που θα μπορούσε να είναι, για παράδειγμα, ένα άγαλμα τέχνης ή ένας τίτλος ιδιοκτησίας ενός οικοπέδου. Στην περίπτωση αυτή, μπορεί να γίνει χρήση ενός token για την ψηφιακή αναπαράσταση της ιδιοκτησίας, κάτι που θα επέτρεπε την αναζήτηση του ιστορικού αυτής στο δίκτυο blockchain. Ταυτόχρονα, θα ήταν δυνατή η ευκολότερη παρακολούθηση, διαχείριση και επιβεβαίωση των όποιων συναλλαγών σχετίζονται με αλλαγή της ιδιοκτησίας αυτής.

Οι περιπτώσεις αυτές, βέβαια, έχουν και επιπλέον κινδύνους, καθώς πέρα από τους κανόνες του δικτύου blockchain μέσα στο οποίο γίνεται η διαχείριση της ιδιοκτησίας των tokens και τους οποίους πρέπει να ακολουθούν οι χρήστες για την όποια συναλλαγή συχνά τα φυσικά αντικείμενα πρέπει να υπακούσουν και σε νόμους του πραγματικού κόσμου. Οπότε είναι αναγκαίο οι συναλλαγές που λαμβάνουν χώρα στον ψηφιακό κόσμο να είναι επιτρεπτές και νόμιμες και στον πραγματικό.

Επιπλέον, τα tokens στον κόσμο του blockchain μπορούν να έχουν κάποια ιδιαίτερα χαρακτηριστικά που τα συντροφεύουν και τα ξεχωρίζουν. Έτσι, τα tokens μπορούν να:

- είναι μοναδικά (ή και όχι),
- διαιρεθούν σε μικρότερης αξίας, π.χ. ένα κρυπτονόμισμα (ή όχι, π.χ. ένα έργο τέχνης).

Ένα token που συμβολίζει ένα κρυπτονόμισμα *δεν είναι μοναδικό*. Μπορεί εύκολα να ανταλλαχθεί με ένα άλλο αντίστοιχο, χωρίς να έχει κάποια διαφορά η ανταλλαγή αυτή ως προς την αξία που έχει στα χέρια του ο ιδιοκτήτης πριν και μετά από αυτήν. Για παράδειγμα, όπως ακριβώς ένα νόμισμα του 1€ είναι ισοδύναμο με οποιοδήποτε άλλο νόμισμα του 1€, έτσι και ένα κρυπτονόμισμα ίσο με 1Ether είναι ισοδύναμο με ένα οποιοδήποτε άλλο κρυπτονόμισμα του 1Ether. Μάλιστα, είναι δυνατό να διαιρεθούν και σε μικρότερης αξίας,

σύμφωνα και με τον Πίνακα 2.1. Τέτοιου είδους tokens λέγονται *εναλλάξιμα* (*fungible*) και είναι ιδιαίτερα συχνά στο δίκτυο του Ethereum.

Στο Κεφάλαιο 7 θα βρείτε περισσότερες λεπτομέρειες σχετικά με τα είδη των tokens και την υλοποίησή τους.

Από την άλλη πλευρά, υπάρχουν tokens που είναι μοναδικά και δεν μπορούν να ανταλλαχθούν με κάποιο άλλο ισοδύναμο. Αυτά τα tokens ονομάζονται *Non-Fungible Tokens* (*NFTs*)[28]. Το κάθε NFT έχει το δικό του μοναδικό χαρακτηριστικό (id), που χρησιμοποιείται για να ξεχωρίζει μέσα σε ένα δίκτυο blockchain και, επίσης, δεν μπορεί να διαιρεθεί σε μικρότερα κομμάτια.

Ταυτόχρονα, όμως, τα NFTs μπορούν να αποτελέσουν μέρος μιας συναλλαγής στην οποία παραχωρούνται τα δικαιώματα του token από τον τρέχοντα ιδιοκτήτη σε έναν άλλο, συνήθως με κάποιο οικονομικό αντίτιμο. Αυτό που δεν μπορεί να γίνει όμως είναι να δοθεί κάποιο άλλο «ισοδύναμο» token πίσω στον αρχικό ιδιοκτήτη, όπου με τη λέξη «ισοδύναμο» εννοείται η έννοια της αντιστοίχισης όλων των χαρακτηριστικών του και όχι αποκλειστικά από θέμα αξίας. Έτσι, το token που αντιπροσωπεύει την ιδιοκτησία ενός συγκεκριμένου αντικειμένου (είτε ψηφιακού είτε φυσικού) είναι μοναδικό.

Σε ένα δίκτυο blockchain τα tokens αποτελούν ένα σύνολο από δεδομένα τα οποία μπορούν να προγραμματιστούν έτσι ώστε να έχουν προκαθορισμένες ιδιότητες και να εκτελούν προκαθορισμένες ενέργειες. Η μεγάλη επιτυχία του δικτύου του Ethereum βρίσκεται στη δημιουργία προτύπων τα οποία μπορούσε να χρησιμοποιήσει ο δημιουργός ενός καινούργιου token για να βασιστεί στην κατασκευή τους, αναλόγως με το είδος του token που ήθελε να αναπτύξει (Fungible ή Non-Fungible). Το πρότυπο αυτό (π.χ. ERC20 για τα Fungible Tokens ή ERC721[29] για τα NFTs) επιτρέπει τον ορισμό ενός κοινού τρόπου με τον οποίο μπορούν να διαβαστούν, δημιουργηθούν και χρησιμοποιηθούν τα διάφορα tokens μέσα στο δίκτυο του Ethereum.

Τα πρότυπα αυτά άνοιξαν τον δρόμο για την ανάπτυξη εφαρμογών, τόσο D-apps (βλ. Υποενότητα 2.3.6) όσο και πορτοφολιών (βλέπετε Ενότητα 3.5), που μπορούσαν να διαχειριστούν tokens και μπορούσαν να εκτελέσουν συναλλαγές με αυτά.

Αυτό έχει ιδιαίτερη σημασία, καθώς η αποστολή του ether αποτελεί εσωτερική πράξη στο δίκτυο του Ethereum, όχι όμως και η συναλλαγή των tokens. Η τελευταία απαιτεί τη σύνταξη και εγκατάσταση στο δίκτυο ενός smart contract το οποίο θα περιγράφει πώς θα γίνουν οι ενέργειες αυτές και το οποίο θα εξηγεί τον υπολογισμό του αριθμού των tokens που έχει ένας χρήστης στην κατοχή του. Παραδείγματα των περιεχομένων των προτύπων αυτών υπάρχουν στο Κεφάλαιο 7.

---

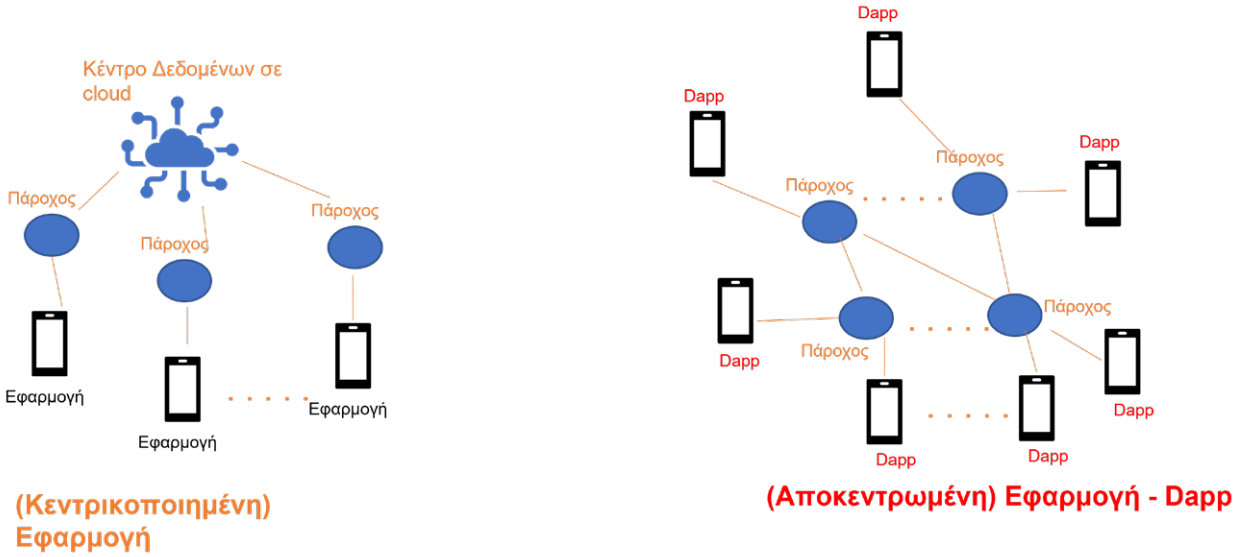
[28] Στα ελληνικά δεν έχει επικρατήσει μια συγκεκριμένη μετάφραση για τον όρο NFTs. Μεταφράζεται ως «μη ανταλλάξιμο διακριτικό», «μη εναλλάξιμα κρυπτοπαραστατικά» ή «μη εμπορεύσιμες μάρκες». Στο βιβλίο, για διευκόλυνση, θα αποδίδεται με τα αρχικά των αγγλικών λέξεων που ξεχωρίζουν στη διεθνή βιβλιογραφία.

[29] ERC είναι τα αρχικά από τις λέξεις Ethereum Request for Comment, δηλαδή πρόκειται για προτάσεις βελτίωσης του πρωτοκόλλου του Ethereum. Οι αριθμοί που συντροφεύουν τις προτάσεις αυτές αντιστοιχούν στους αριθμούς (προς επίλυση) που έλαβαν οι προτάσεις αυτές στο GitHub.

2.3.6 Αποκεντρωμένες Εφαρμογές (Decentralized Applications, DApps)

Οι *αποκεντρωμένες εφαρμογές* αποτέλεσαν ένα σημαντικό κομμάτι των εργαλείων του Web3, της προσπάθειας δηλαδή να επανασχεδιαστεί ο Παγκόσμιος Ιστός με τη χρήση λύσεων που βασίζονται στην τεχνολογία του blockchain, με έμφαση στην αποκέντρωση και στη χρήση των tokens σε οικονομικής φύσεως εφαρμογές (Ethereum-Web3, 2022).

Αποκεντρωμένη είναι μια εφαρμογή η οποία χρησιμοποιεί μια διεπαφή χρήστη, συνήθως στο web, και η οποία έχει εγκατασταθεί και τρέχει πάνω σε ένα δίκτυο blockchain, σε αντίθεση από το να βρίσκεται σε έναν (κεντρικοποιημένο) εξυπηρετητή (**Εικόνα 2.12**).



Εικόνα 2.12 *Σύγκριση λειτουργίας μιας τυπικής εφαρμογής στο Διαδίκτυο σε σχέση με μια Αποκεντρωμένη Εφαρμογή (DApp).*

Πιο αναλυτικά, ένα DApp χαρακτηρίζεται από:

- *Μια εφαρμογή για αλληλεπίδραση με χρήστη* (*frontend*): Για τη δημιουργία μιας τέτοιας εφαρμογής γίνεται χρήση ενός συνόλου από γλώσσες προγραμματισμού οι οποίες χρησιμοποιούνται ευρέως για τη δημιουργία μιας διαδικτυακής εφαρμογής (π.χ. HTML, CSS, JavaScript). Για το κομμάτι αυτό δεν απαιτείται κάποια γνώση για τον τρόπο με τον οποίο θα διαχειριστεί το DApp τα δεδομένα που θα εισέλθουν από την εφαρμογή.
- Συνήθως, όμως, αυτό που χρειάζεται είναι η χρήση των βιβλιοθηκών web3.js για τη σύνδεση με το δίκτυο του Ethereum και με το smart contract που θα διαχειριστεί τα δεδομένα που θα εισαχθούν.
- *Ένα smart contract* (*backend*): Στην περίπτωση των DApps, η διαχείριση των δεδομένων θα πρέπει να γίνει μέσα στο δίκτυο του blockchain με τη βοήθεια ενός προγράμματος το οποίο ξέρει πώς να τα διαχειριστεί. Αυτά τα προγράμματα είναι smart contracts τα οποία έχουν γραφτεί σε γλώσσα που καταλαβαίνει το δίκτυο (βλέπε Υποενότητα 2.3.2) και τα οποία αναλαμβάνουν να υλοποιήσουν την ουσία της εφαρμογής, είτε δημιουργώντας συναλλαγές στο δίκτυο του Ethereum είτε καλώντας άλλα smart contracts μέσα σε αυτό. Υπάρχουν δύο σημεία που χρήζουν ιδιαίτερης προσοχής στο σημείο αυτό:
  I. το κόστος σε gas που χρειάζεται για την εγκατάσταση και χρήση του smart contract και
  II. το γεγονός ότι ο κώδικας του smart contract δεν μπορεί να διορθωθεί, αλλά θα πρέπει να εγκατασταθεί από την αρχή η όποια νέα έκδοση (με συνέπειες στο [i]). Επομένως, ο σχεδιασμός και η υλοποίηση κάθε smart contract απαιτεί μεγάλη προσοχή και έλεγχο.
- *Αποθήκευση δεδομένων*: Ολοένα και πιο συχνά, τόσο λόγω κόστους αλλά και λόγω μεγάλου όγκου, τα δεδομένα μιας αποκεντρωμένης εφαρμογής δεν αποθηκεύονται μέσα στο δίκτυο του blockchain. Αντιθέτως, αυτά αποθηκεύονται σε εξωτερικούς κατανεμημένους χώρους αποθήκευσης (με χρήση πρωτοκόλλων όπως το IPFS[30] ή το Swarm[31], ιδιαίτερα δημοφιλείς λύσεις στο Ethereum) και χρησιμοποιείται μόνο το μοναδικό αποτύπωμά τους ως αναφορά στο δίκτυο του Ethereum.

[30] Online Σύνδεσμος: https://ipfs.io/
[31] Online Σύνδεσμος: https://api.gateway.ethswarm.org/bzz/swarm.eth/

Όπως κάθε λύση, το ίδιο και η χρήση ενός Dapp έχει τα πλεονεκτήματα και τα μειονεκτήματά της (Ethereum-DApp, 2022).

Συνοπτικά, αυτά είναι:

Πλεονεκτήματα:

- *Μηδενικός χρόνος μη λειτουργίας*: Η εγκατάσταση του smart contract σε ένα δίκτυο blockchain εξασφαλίζει ότι όλοι οι κόμβοι αυτού έχουν συνεχώς πρόσβαση στον κώδικα ο οποίος και τρέχει μέσα στην EVM. Επομένως, η λειτουργία της εφαρμογής μπορεί να συνεχίσει απρόσκοπτα.
- *Ιδιωτικότητα και έλλειψη λογοκρισίας*: Για τη δημιουργία συναλλαγών στο δίκτυο blockchain χρειάζεται κάποιος να αποκτήσει μια διεύθυνση του δικτύου και κατόπιν να συναλλάσσεται χρησιμοποιώντας αυτήν, χωρίς να είναι αναγκαίο να εμφανίσει την πραγματική του ταυτότητα. Επιπρόσθετα, δεν υπάρχει κάποια οντότητα στο δίκτυο η οποία απαγορεύει σε οποιονδήποτε τη συναλλαγή στο δίκτυο.
- *Ακεραιότητα δεδομένων*: Όπως έχει τονιστεί, τα δεδομένα που μπαίνουν στο ledger δεν μπορούν να μεταβληθούν, εξασφαλίζοντας με τον τρόπο αυτόν την ακεραιότητά τους.
- *Επιβεβαίωση συμπεριφοράς*: Ο τρόπος με τον οποίο το smart contract θα διαχειριστεί τα δεδομένα είναι συγκεκριμένος και προβλεπόμενος, επιτρέποντας την επιβεβαίωση των ενεργειών του ακόμα και μέσα σε ένα περιβάλλον όπου δεν υπάρχει μια έμπιστη οντότητα, όπως συμβαίνει σε ένα αποκεντρωμένο περιβάλλον.

Μειονεκτήματα:

- *Συντήρηση*: Ο κώδικας σε ένα smart contract δεν μπορεί να αλλάξει, δυσκολεύοντας έτσι τη συντήρηση ενός DApp.
- *Έξοδα απόδοσης*: Δεδομένης της συμμετοχής ολόκληρου του δικτύου του blockchain στη λειτουργία και επιβεβαίωση των συναλλαγών ενός DApp, το κόστος αυξάνεται εκθετικά.
- *Συμφόρηση δικτύου*: Δεδομένης της ικανότητας του δικτύου του Ethereum να επεξεργάζεται 10-15 συναλλαγές το δευτερόλεπτο (tps), τα όρια του δικτύου είναι καθορισμένα. Σε περίπτωση που DApps παράγουν παραπάνω συναλλαγές, τότε θα επέλθει συμφόρηση στο δίκτυο, δημιουργώντας μεγαλύτερες καθυστερήσεις.
- *Εμπειρία χρήστη*: Εξαρτάται από τη δυνατότητα να γίνει πιο απλή η διασύνδεση με το δίκτυο του blockchain, καθώς τα βήματα αυτή τη στιγμή μπορεί να θεωρηθούν αποτρεπτικά (ή δύσκολα) από κάποιους χρήστες.

Κλείνοντας την κουβέντα για τα DApps, η σελίδα *State of the DApps* (2022) συγκεντρώνει έναν μεγάλο αριθμό από περισσότερα από 3.000 DApps σε γνωστά δίκτυα blockchain. Αποτελεί μια πηγή για να μπορέσει ο κάθε ενδιαφερόμενος να αναζητήσει κάποια εφαρμογή ή να δει τι υπάρχει στον χώρο. Οι εφαρμογές εκεί δεν περιορίζονται μόνο στο Ethereum, αλλά καλύπτουν και άλλα γνωστά δίκτυα blockchain (π.χ. EOS, GoChain, xDai κ.ά.).

2.3.7 Τεχνικές συναίνεσης

Όπως συμβαίνει και με το Bitcoin, έτσι και το Ethereum εφαρμόζει έναν αλγόριθμο συναίνεσης για να μπορέσει να συμφωνήσει το δίκτυο στην επιλογή του επόμενου block που θα προστεθεί στην αλυσίδα. Μάλιστα, η τεχνική που χρησιμοποιείται είναι, προς το παρόν, η ίδια με αυτήν του Bitcoin, δηλαδή το PoW. Όμως η εφαρμογή της τεχνικής έχει διαφορές σε σχέση με το Bitcoin, και για αυτόν τον λόγο παρατηρείται διαφορά στο tps που επιτυγχάνουν τα δύο αυτά δίκτυα (βλέπε Ενότητα 1.2).

Η τεχνική του PoW που εφαρμόζεται στο Ethereum ονομάζεται *Ethash* και έχει προκύψει από την εξέλιξη του συνδυασμού δύο άλλων αλγόριθμων: του αλγόριθμου Dagger (που τον είχε εφεύρει ο V. Buterin, ιδρυτής του Ethereum) και τον αλγόριθμο Hashimoto (που τον είχε εφεύρει ο Thaddeus Dryja). Ο μεν αλγόριθμος Dagger χρησιμοποιούσε έναν *Κατευθυνόμενο Άκυκλο Γράφο* (*Direct Acyclic Graph, DAG*) για τη δημιουργία μιας πολύ μεγάλης δομής δεδομένων, η οποία ανανεωνόταν κάθε 30.000 blocks.

Το μέγεθος αυτού του γράφου ξεκίνησε να είναι περίπου 1GB και πλέον είναι περίπου 5GB. Στη δομή αυτή πραγματοποιούνται υπολογισμοί μνήμης με σκοπό να χρησιμοποιηθούν από τον αλγόριθμο του Hashimoto για

την ολοκλήρωση της διαδικασίας του mining. Ο τελευταίος είχε ως βασικό στόχο να εισαγάγει μια λειτουργία η οποία θα ήταν δύσκολο να την ολοκληρώσουν οι δημοφιλείς κάρτες ASIC (Application Specific Integrated Circuit). Οι κάρτες ASIC χρησιμοποιούνται κατά κόρον στην εξόρυξη του BTC[32] και βασίζονται στον επεξεργαστή τους για την παραγωγή νέων hashes[33] στην προσπάθεια ολοκλήρωσης του PoW.

Το Ethereum, από την αρχή του σχεδόν, σχεδίαζε να εγκαταλείψει, κάποια στιγμή, το PoW και να χρησιμοποιήσει τον αλγόριθμο Proof of Stake (PoS), θέλοντας να αντιμετωπίσει τις αδυναμίες του PoW. Τέτοιες θεωρούνται η μεγάλη επεξεργαστική ισχύς που απαιτείται για την επίλυση του μαθηματικού γρίφου και η οποία οδηγεί σε σημαντική κατανάλωση ενέργειας από τους συμμετέχοντες στη διαδικασία του mining. Η μετάβαση σε PoS αποτελεί τμήμα της καινούργιας έκδοσης του πρωτοκόλλου, γνωστού και ως Ethereum 2.0, και αναμένεται να γίνει μέσα στο 2022, αν και η ημερομηνία αυτή έχει ανανεωθεί αρκετές φορές.

Ο αλγόριθμος του PoS, γνωστός και ως *Casper* στο Ethereum, δεν βασίζεται στη δημιουργία και δοκιμή πολλών hashes ανά δευτερόλεπτο (δηλαδή στην εξόρυξη του PoW), αλλά λειτουργεί με μια διαφορετική προσέγγιση. Έτσι, στο PoS το δίκτυο ορίζει έναν αριθμό από αξιολογητές και όποιος θέλει μπορεί να πάρει μια ελεύθερη θέση. Για να το καταφέρει αυτό, θα πρέπει να προχωρήσει σε μια ειδική συναλλαγή. Σε αυτήν αποστέλλει σε μια ειδική διεύθυνση του δικτύου ένα ποσό από τα διαθέσιμα κρυπτονομίσματά του (ether για το Ethereum). Αυτά θα δεσμευτούν εκεί και θα επιστρέψουν στον ιδιοκτήτη τους εφόσον αυτός εκπληρώσει σωστά τον ρόλο του ως αξιολογητής, που είναι να προτείνει το επόμενο block στην αλυσίδα του blockchain. Όταν οι προτάσεις του κάθε αξιολογητή είναι σωστές και συναινούν (στην ψηφοφορία) και οι υπόλοιποι ομότιμοί του, τότε το block που προτείνει εισάγεται στην αλυσίδα και αυτός θα επιβραβευθεί παίρνοντας πίσω το ποσό του που έχει δεσμευτεί και ένα επιπλέον μερίδιο, που είναι ανάλογο με το ποσό που είχε δεσμεύσει.

Οι αξιολογητές προτείνουν σε σειρά, ο ένας μετά τον άλλον, μέχρι να τελειώσει η θητεία τους μετά από ένα ορισμένο αριθμό blocks που θα έχουν επιβεβαιώσει ή προτείνει, οπότε και επιστρέφονται τα χρήματα και δίνεται η αμοιβή.

Σε περίπτωση που ένας αξιολογητής προτείνει ένα block το οποίο απορρίπτεται από τους υπόλοιπους είτε γιατί περιέχει μια παράνομη συναλλαγή ή για οποιονδήποτε άλλο λόγο, τότε αυτός κινδυνεύει με την απώλεια των χρημάτων που έχει δεσμεύσει αναλαμβάνοντας τον ρόλο του αξιολογητή. Αυτό έχει σκοπό να λειτουργήσει και ως ένα μέτρο πίεσης προς τους αξιολογητές για να ενεργήσουν για το καλό του δικτύου, κάνοντας συχνά ακριβή την προσπάθεια εξαπάτησης αυτού. Περισσότερα για τα προτερήματα και τα μειονεκτήματα του PoS στο Κεφάλαιο 5.

2.3.8 Oracles

Τα *oracles* αποτελούν συστήματα τα οποία επικοινωνούν δεδομένα σε ένα δίκτυο blockchain. Τα δεδομένα αυτά συχνά προέρχονται από τον εξωτερικό κόσμο και μεταφέρονται μέσω συναλλαγών μέσα στο δίκτυο του blockchain σε ένα smart contract, που αναλαμβάνει να τα διαχειριστεί. Ιδανικά, θα πρέπει τα δεδομένα που εισάγονται από κάποιο oracle να προέρχονται από μια έμπιστη πηγή ή από συνδυασμό πολλών oracles που δημιουργούν ένα κατανεμημένο δίκτυο.

Με τον τρόπο αυτόν είναι δυνατόν να εξυπηρετηθούν πολλές περιπτώσεις χρήσης (βλέπε και Κεφάλαιο 9) στις οποίες τα smart contracts μπορεί να χρειάζονται δεδομένα που είτε βρίσκονται εκτός του blockchain είτε παράγονται τυχαία. Υπενθυμίζεται ότι μέσα στο blockchain και, συγκεκριμένα για το Ethereum, μέσα στην EVM όλες οι αποφάσεις είναι ντετερμινιστικές, έτσι ώστε όλοι οι κόμβοι να φθάνουν στο ίδιο αποτέλεσμα και να επέρχεται η συναίνεση. Η εισαγωγή της οποιασδήποτε τυχαιότητας δύναται να αποσυντονίσει το σύστημα, καθώς θα οδηγήσει στη λήψη διαφορετικών αποφάσεων για την ίδια μετάβαση στη μηχανή καταστάσεων. Κάτι τέτοιο δεν είναι αποδεκτό σε ένα σύστημα blockchain.

Έτσι, η οποιαδήποτε τυχαιότητα θα πρέπει να έρθει από εξωτερικούς παράγοντες και να εισέλθει μέσα σε μια συναλλαγή στο blockchain. Αυτό το κενό καλύπτουν τα oracles.

Η μεγάλη ποικιλία από πηγές δεδομένων εκτός αλυσίδας (off-chain) οδήγησε και στη δημιουργία διαφόρων ειδών oracles, τα οποία εν γένει χαρακτηρίζονται από ένα σύνολο από κοινές λειτουργίες: ανάκτηση δεδομένων, έλεγχός τους, επεξεργασία τους και προώθησή τους στο δίκτυο blockchain.

---

[32] Η ανάγκη για χρήση μιας κάρτας ASICS σε συνδυασμό με τις χρεώσεις στο ρεύμα είναι κάτι που έχει οδηγήσει, σύμφωνα με πολλούς, σε μια κεντρικοποίηση της λειτουργίας εξόρυξης στο Bitcoin (Beikverdi & Song, 2015).

[33] Το hash είναι μια σειρά από αριθμούς (συγκεκριμένου μεγέθους) που είναι αποτέλεσμα της εφαρμογής μαθηματικών συναρτήσεων, γνωστές ως *συναρτήσεις κατακερματισμού.*

Στο *Chainlink* (2022) παρουσιάζεται μια σειρά διαφορετικών τύπων από oracles, τα οποία μπορούν να χρησιμοποιηθούν ανάλογα με την περίπτωση.

2.4 Ομοιότητες και διαφορές

Το Ethereum έχει αρκετά κοινά χαρακτηριστικά με το Bitcoin: κάνει χρήση ενός κατανεμημένου δικτύου από ομότιμους κόμβους, εφαρμόζει το ίδιο πρωτόκολλο κατανεμημένης συναίνεσης στο δίκτυο (το PoW) και ακολουθεί συγκεκριμένους κανόνες συναίνεσης, εφαρμόζει λύσεις που βασίζονται σε σύγχρονες τεχνικές κρυπτογραφίας και ψηφιακών υπογραφών και διαχειρίζεται συναλλαγές με τη χρήση του δικού του κρυπτονομίσματος (ether).

Δεδομένης όμως της διαφορετικής προσέγγισης σχετικά με τον ρόλο τους, τα δύο δίκτυα εμφανίζουν και σημαντικές διαφορές.

Ο **Πίνακας 2.2** συνοψίζει περιληπτικά τις ομοιότητες και τις διαφορές των δύο πρώτων και πιο δημοφιλών δικτύων blockchain:

	BITCOIN	**ETHEREUM**
ΔΗΜΙΟΥΡΓΙΑ 1ΟΥ BLOCK	09/01/2019	30/07/2015
ΣΚΟΠΟΣ	Να αποτελέσει μια εναλλακτική πλατφόρμα για χρηματικές συναλλαγές	Να αποτελέσει τον παγκόσμιο υπολογιστή για αποκεντρωμένες εφαρμογές
ΑΛΓΟΡΙΘΜΟΣ ΣΥΝΑΙΝΕΣΗΣ	PoW	PoW (μετάβαση σε PoS)
ΧΡΟΝΟΣ ΔΗΜΙΟΥΡΓΙΑΣ BLOCK	10 λεπτά κατά μ.ό.	15 δευτερόλεπτα κατά μ.ό.
ΑΡΙΘΜΟΣ ΣΥΝΑΛΛΑΓΩΝ / ΔΕΥΤΕΡΟΛΕΠΤΟ	7	30
ΑΡΙΘΜΟΣ BLOCKS / ΩΡΑ (Μ.Ο.)	6	263
ΑΡΙΘΜΟΣ BLOCKS	732.726	14.623.151
ΑΡΙΘΜΟΣ ΕΝΕΡΓΩΝ ΔΙΕΥΘΥΝΣΕΩΝ (24 ΩΡΕΣ)	794.098	902.184
ΜΕΓΕΘΟΣ ΤΟΥ LEDGER	465,47 GB	345,17 GB
ΤΥΠΟΣ BLOCKCHAIN	Δημόσιο	Δημόσιο

Πίνακας 2.2 *Σύγκριση ομοιοτήτων και διαφορών ανάμεσα σε Bitcoin και Ethereum.*[34]

[34] Online Σύνδεσμος: https://bitinfocharts.com/

Βιβλιογραφία

Antonopoulos, A. M. (2017). *Mastering Bitcoin. Programming the Open Blockchain* (2nd ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., & Wood, G. (2019). *Mastering Ethereum* (1st ed.). O'Reilly Media, Inc.

Beikverdi, A., & Song, J. (2015). The trend of centralization in Bitcoin's distributed network. *IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing (SNPD)*. pp. 1-6. Online πηγή: https://ieeexplore.ieee.org/document/7176229

Chainlink (2022). *What is a Blockchain Oracle?*. Online πηγή: https://chain.link/education/blockchain-oracles [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Ethereum website (2022). Online πηγή: https://ethereum.org/en/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Ethereum-DApp (2022). *Ethereum Documentation*. Online πηγή: https://ethereum.org/en/developers/docs/dapps/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Ethereum-Web3 (2022). *Introduction to Web3*.0. Online πηγή: https://ethereum.org/en/web3/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

JavaScript (2022). Επίσημο Site. Online πηγή: https://www.javascript.com/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Mix (2019). *These are the top 10 programming languages in blockchain*. Hard Fork, TDW. March 2019. Online πηγή: https://thenextweb.com/news/javascript-programming-java-cryptocurrency [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* 2008. Online πηγή: https://bitcoin.org/bitcoin.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Rust (2022). *Rust v.1.59.0.* Online πηγή: https://www.rust-lang.org/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Solidity (2022). *Solidity v0.8.13, Documentation.* Online πηγή: https://docs.soliditylang.org/en/v0.8.13/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

StateofDapps (2022). *State of Dapps, Explore Decentralized Application.* Online πηγή: https://www.stateofthedapps.com/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Vyper (2022). Επίσημο Site. Online πηγή: https://vyper.readthedocs.io/en/stable/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 3

Χρήση Κλειδιών και Απόκτηση Διευθύνσεων

Σύνοψη

Στο Κεφάλαιο αυτό αναλύεται ο ρόλος που παίζει η κρυπτογραφία στην τεχνολογία του blockchain, ξεκινώντας με τη δημιουργία και τη χρήση ενός ζεύγους κλειδιών (ιδιωτικού και δημόσιου) σε ένα δίκτυο blockchain.

Ακολούθως παρουσιάζεται η σχέση που έχει το ζεύγος αυτό των κλειδιών με τη δημιουργία μιας διεύθυνσης δικτύου για τα δύο γνωστά δίκτυα blockchain (Bitcoin και Ethereum), καθώς και η χρήση μιας εφαρμογής πορτοφολιού για την αποθήκευσή τους.

Επιπλέον, εξηγείται η βασική διαφορά στον τρόπο δημιουργίας κλειδιών από ένα πορτοφόλι και αναλύονται τα είδη των πορτοφολιών που συναντώνται. Τέλος, επισημαίνεται πώς μπορεί να βρεθεί ο κατάλληλος τύπος του πορτοφολιού ανάλογα με τον αναμενόμενο τρόπο χρήσης του.

Προαπαιτούμενη γνώση

Ανάγνωση και κατανόηση των Κεφαλαίων 1 και 2.

3.1 Μια εισαγωγή στην κρυπτογραφία

Η κρυπτογραφία χρησιμοποιείται στη σύγχρονη εποχή στις ψηφιακές επικοινωνίες ως τρόπος που επιτρέπει την απόκρυψη του περιεχομένου ηλεκτρονικών μηνυμάτων και που μετατρέπει το περιεχόμενό τους σε μια μορφή που δεν είναι κατανοητή από τον οποιονδήποτε. Η αλλαγή αυτή απαιτεί προσπάθεια από αυτόν που θα υποκλέψει το μήνυμα έτσι ώστε να μπορέσει να το μεταφράσει, αποκωδικοποιώντας το.

Μάλιστα, έχουν αναπτυχθεί διάφορες μέθοδοι κρυπτογράφησης ως τώρα και καθεμία καλύπτει διαφορετικές ανάγκες ασφάλειας.

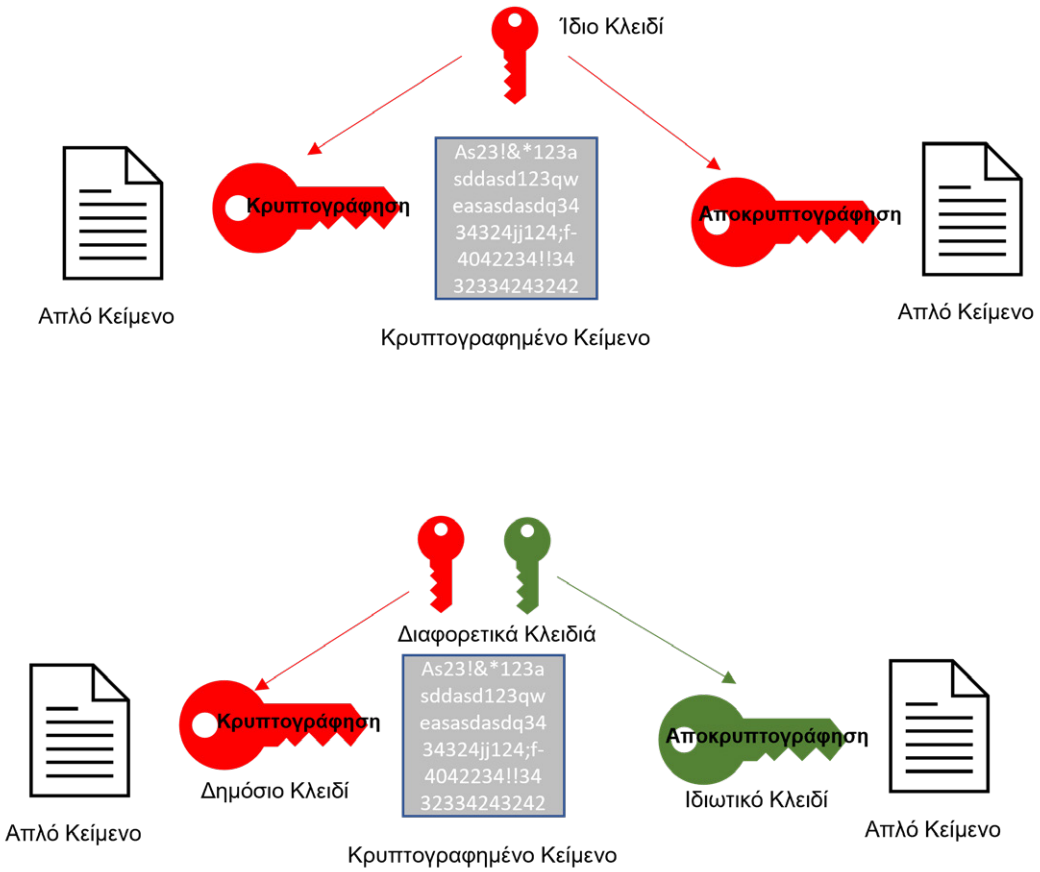
Οι μέθοδοι αυτές μπορούν να χωριστούν σε δύο μεγάλες βασικές κατηγορίες:

- *Ασύμμετρης Κρυπτογραφίας*: Η μέθοδος αυτή είναι γνωστή και με το όνομα Κρυπτογράφηση Δημόσιου Κλειδιού. Κάνει χρήση δύο διαφορετικών κλειδιών για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων. Το κλειδί που χρησιμοποιείται στην κρυπτογράφηση είναι το δημόσιο κλειδί, ενώ αυτό που χρησιμοποιείται για την αποκρυπτογράφηση είναι το ιδιωτικό. Γνωστά παραδείγματα αλγόριθμων που υλοποιούν αυτής της μορφής την κρυπτογραφία είναι: ο αλγόριθμος Diffie-Hellman, ο RSA[35] και οι τεχνικές Ελλειπτικών Καμπυλών (Hoffestein et al, 2021).
- *Συμμετρικής Κρυπτογραφίας*: Πρόκειται για παλαιότερη αλλά εξαιρετικά αποτελεσματική και γρήγορη μέθοδο στην οποία χρησιμοποιείται ένα κλειδί και για τις δύο διαδικασίες (κρυπτογράφηση και αποκρυπτογράφηση). Γνωστοί αλγόριθμοι που υλοποιούν αυτής της μορφής την κρυπτογραφία είναι: η AES (Advanced Encryption Standard), ο Twofish και ο 3-DES.

Στην **Εικόνα 3.1** φαίνεται ο τρόπος χρήσης των κλειδιών στις δύο αυτές μεθόδους κρυπτογράφησης. Στην περίπτωση της συμμετρικής είναι ίδιο το κλειδί στον αποστολέα και στον παραλήπτη, ενώ στην ασύμμετρη όχι.

Αξίζει να αναφερθεί ότι ένας από τους λόγους που οδήγησαν στην ασύμμετρη κρυπτογραφία (που είναι πιο νέα) σχετίζεται και με την ανάγκη διαμοιρασμού του κοινού κλειδιού (στην περίπτωση της συμμετρικής) μέσα σε ένα πιθανώς μη προστατευμένο κανάλι. Από την άλλη πλευρά, η ασύμμετρη δεν αντιμετωπίζει τέτοιο κίνδυνο, καθώς η κάθε πλευρά έχει το δικό της ζεύγος κλειδιών.

[35] Rivest, Shamir, Adleman (RSA).



Εικόνα 3.1 *Χρήση Συμμετρικής (επάνω) και χρήση Ασύμμετρης Κρυπτογραφίας (κάτω).*

Ταυτόχρονα, λαμβάνοντας υπόψη ότι μόνο ο κάτοχος του ιδιωτικού κλειδιού (από το ζεύγος των κλειδιών) μπορεί να αποκρυπτογραφήσει ένα μήνυμα που έχει κρυπτογραφηθεί με χρήση του δημόσιου κλειδιού, η επανασύσταση ενός μηνύματος μπορεί να αποτελέσει και απόδειξη ιδιοκτησίας του κλειδιού αυτού από εκείνον που το πραγματοποίησε.

Αυτό είναι ένα χαρακτηριστικό που είναι ζητούμενο σε ένα σύστημα blockchain.

Στον **Πίνακα 3.1** συγκεντρώνονται οι διαφορές μεταξύ των δύο επικρατέστερων μεθόδων κρυπτογραφίας.

Διαφορές	**Συμμετρική Κρυπτογραφία**	**Ασύμμετρη Κρυπτογραφία**
Μέγεθος Δεδομένων	Χρήση για αποστολή μεγάλων δεδομένων	Χρήση για αποστολή μικρών δεδομένων
Χρήση Πόρων	Χαμηλή	Υψηλή
Μέγεθος Κλειδιών	128-256 bits	Κλειδί RSA 2.048-bit ή μεγαλύτερο
Αριθμός Κλειδιών	Ένα κλειδί για κρυπτογράφηση και αποκρυπτογράφηση	Δύο κλειδιά: ένα για κρυπτογράφηση και ένα για αποκρυπτογράφηση
Ασφάλεια	Λιγότερο ασφαλές λόγω της χρήσης 1 κλειδιού	Μεγαλύτερη λόγω της χρήσης 2 κλειδιών
Ιστορία	Παλιά τεχνική	Νεότερη μέθοδος
Κίνδυνοι	Η χρήση ενός κλειδιού	Η απώλεια του ιδιωτικού κλειδιού
Ταχύτητα	Γρήγορη	Πιο αργή

Πίνακας 3.1 *Διαφορές ανάμεσα στις δύο κύριες μεθόδους κρυπτογράφησης (συμμετρική και ασύμμετρη).*

Σημαντικό για την εφαρμογή λύσεων που χρησιμοποιούν ασύμμετρη κρυπτογραφία είναι η δυνατότητα παραγωγής του ζεύγους των κλειδιών. Το ιδιωτικό κλειδί είναι δυνατόν να δημιουργηθεί με τυχαία επιλογή μιας σειράς από bits συγκεκριμένου μεγέθους, την οποία και γνωρίζει μόνο ο κάτοχος του κλειδιού αυτού.

Κατόπιν πρέπει να εφαρμοστεί μια μέθοδος για τη δημιουργία του δημόσιου κλειδιού. Η μέθοδος αυτή πρέπει να είναι μίας κατεύθυνσης, να μην είναι εύκολο δηλαδή να αντιστραφεί, ώστε να μην είναι δυνατή η εύρεση του ιδιωτικού κλειδιού από όποιον έχει γνώση του δημόσιου.

Μια τέτοια μέθοδος αποτελεί η *Κρυπτογραφία Ελλειπτικών Καμπυλών* (*Elliptic Curve Cryptography, ECC*), η οποία έχει γνωρίσει σημαντική αναγνώριση και χρήση τα τελευταία χρόνια, κυρίως λόγω της πολύ καλής της απόδοσης σε φορητές συσκευές (έξυπνα τηλέφωνα ή ταμπλέτες). Η δυνατότητα που προσφέρει για αυξημένη ασφάλεια ενώ, ταυτόχρονα, διατηρεί το μέγεθος του κλειδιού μικρό έχει βοηθήσει στην εφαρμογή της σε τέτοιες συσκευές.

Στον **Πίνακα 3.2** φαίνονται τα μεγέθη κλειδιών που απαιτούνται από την ECC και τη RSA για την επίτευξη ασφάλειας ίδιου επιπέδου. Η RSA επιλέχθηκε καθώς είναι και αυτή μέθοδος ασύμμετρης κρυπτογραφίας με μεγάλη διάδοση και αναγνώριση.

Μήκος Κλειδιού (bits)	RSA (bits)	ECC (bits)	Έγκυρο μέχρι το έτος…
80	1024	160-223	2010
112	2048	224-255	2030
128	3072	256-383	Μετά το 2031
192	7680	384-511	
256	15360	512+	

Πίνακας 3.2 *Μέγεθος κλειδιών για την επίτευξη ίδιου επιπέδου ασφάλειας ανάμεσα σε RSA και ECC.*

Επιπλέον, η ECC είναι μια μέθοδος η οποία βασίζεται στο πρόβλημα του διακριτού λογάριθμου. Χρησιμοποιεί μαθηματικές πράξεις (προσθέσεις και αφαιρέσεις) πάνω σε σημεία μιας δεδομένης ελλειπτικής καμπύλης για να μπορέσει να δημιουργήσει το ζεύγος του ιδιωτικού και δημόσιου κλειδιού που απαιτείται.

Ξεκινώντας από ένα σημείο στην ελλειπτική καμπύλη (γνωστό και ως γεννήτορας G), είναι δυνατόν πολλαπλασιάζοντας με έναν αριθμό να καταλήξει κάποιος σε ένα άλλο σημείο πάνω σε αυτήν (βλ. Ενότητα 3.3). Ο αριθμός με τον οποίο πολλαπλασιάζεται το σημείο γεννήτορας είναι το *ιδιωτικό κλειδί του χρήστη*, ενώ *το δημόσιο κλειδί* είναι το σημείο στο οποίο καταλήγει κανείς πάνω στην καμπύλη μετά την ολοκλήρωση του πολλαπλασιασμού.

Θα πρέπει να τονιστεί ότι κάθε φορά που ξεκινά κάποιος από το ίδιο σημείο –γεννήτορα G– ο πολλαπλασιασμός με τον ίδιο αριθμό θα καταλήξει στο ίδιο τελικό σημείο πάνω στην ελλειπτική καμπύλη. Από την άλλη πλευρά, γνωρίζοντας ένα σημείο στην καμπύλη, είναι αδύνατον να βρεθεί από ποιο σημείο ξεκίνησε κάποιος για να καταλήξει σε αυτό. Θα πρέπει να δοκιμαστούν όλα τα σημεία στην καμπύλη ως υποψήφια σημεία εκκίνησης για την επιβεβαίωση.

Όλα αυτά τα χαρακτηριστικά οδήγησαν στην ολοένα και ευρύτερη χρήση της μεθόδου ECC, καθώς και στο να λάβει αναγνώριση από την Εθνική Υπηρεσία Ασφαλείας (National Security Agency, NSA) των ΗΠΑ.

Μάλιστα, όπως παρουσιάζεται και στην Ενότητα 3.2, η μέθοδος αυτή χρησιμοποιείται πολύ σε ένα σύστημα blockchain, και η εφαρμογή της σε τέτοια συστήματα συνετέλεσε σημαντικά στην ευρύτερη αποδοχή και χρήση της.

Στην κρυπτογραφία χρησιμοποιούνται, επίσης, συναρτήσεις κατακερματισμού. Ως *συνάρτηση κατακερματισμού* (*hash function*) ορίζεται μια μαθηματική συνάρτηση η οποία μπορεί να λάβει στην είσοδό της δεδομένα οποιουδήποτε μεγέθους και να τα κρυπτογραφήσει, αποδίδοντας στην έξοδό της το αποτέλεσμα (που ονομάζεται hash) συμπιεσμένο σε ένα σταθερό μέγεθος. Αντίστοιχα, με την ECC τα αποτελέσματα της συνάρτησης κατακερματισμού δεν είναι ευκόλως αντιστρέψιμα, δηλαδή δεδομένης μιας εξόδου, η εύρεση της εισόδου που οδήγησε στην έξοδο αυτή δεν μπορεί να προβλεφθεί χωρίς να γίνουν δοκιμές όλων των συνδυασμών και ελέγχου των αποτελεσμάτων τους. Επιπρόσθετα, η έξοδος της συνάρτησης κατακερματισμού θα είναι πάντα η ίδια κάθε φορά που θα τίθεται η ίδια είσοδος σε αυτήν και δεν θα υπάρχει άλλη είσοδος που θα καταλήγει στην ίδια έξοδο.

Τέλος, σε μια προσπάθεια επέκτασης των αποτελεσμάτων των συναρτήσεων κατακερματισμού, υπάρχει η δυνατότητα να συνδυαστούν αυτά σε ένα δυαδικό δένδρο, κάθε φύλλο του οποίου μπορεί να χρησιμοποιηθεί ως είσοδος (με το δυαδικό του ταίρι) σε μια συνάρτηση κατακερματισμού. Διατρέχοντας το δένδρο και χρησιμοποιώντας τα (δυαδικά) ζευγάρια που δημιουργούνται σε κάθε επίπεδο ως είσοδοι σε μια συνάρτηση κατακερματισμού, είναι δυνατόν να φθάσουμε στη ρίζα του δένδρου. Αυτή θα είναι, επίσης, ένα hash το οποίο μπορούμε να πούμε ότι αντιπροσωπεύει όλα τα δεδομένα του δένδρου (δένδρο Merkle).

Στο Κεφάλαιο 4 γίνεται μια πιο λεπτομερής εξήγηση για τον τρόπο με τον οποίο χρησιμοποιούνται τα δένδρα Merkle στα σύγχρονα συστήματα blockchain. Στο Κεφάλαιο 1 έχει γίνει μια ιστορική αναδρομή για τα δένδρα Merkle και τον ρόλο τον οποίο αυτά παίζουν σε ένα σύστημα blockchain.

3.2 Κρυπτογραφία στο blockchain

Δεδομένης της φύσης της τεχνολογίας του blockchain, που αποτελεί μια τεχνολογία που επικεντρώνεται στη διαχείριση ψηφιακών δεδομένων και συναλλαγών, η χρήση της κρυπτογραφίας αποτέλεσε ένα πολύ σημαντικό εργαλείο για να μπορέσει να προσφέρει τα αναμενόμενα αποτελέσματα, όπως ακεραιότητα συναλλαγών, ανωνυμία, ασφάλεια.

Στην ενότητα αυτή παρουσιάζεται ο τρόπος που χρησιμοποιείται η κρυπτογραφία σε ένα δίκτυο blockchain χρησιμοποιώντας ως παραδείγματα τα δίκτυα του Bitcoin και του Ethereum.

Η κρυπτογραφία στον κόσμο του blockchain εφαρμόζεται σε περισσότερα από ένα σημεία. Ενδεικτικά, χρησιμοποιείται για:

- Τη δημιουργία των διευθύνσεων που απαιτούνται για τη συμμετοχή σε ένα δίκτυο blockchain.
- Τη δημιουργία των blocks και τη σύνδεση μεταξύ τους στο ledger.
- Την επίτευξη συναίνεσης στο δίκτυο, καθώς οι προς χρήση τεχνικές εφαρμόζουν κάποια μέθοδο κρυπτογραφίας.
- Τη δημιουργία ψηφιακών υπογραφών στις συναλλαγές, που χρησιμοποιούνται για απόδειξη της ιδιοκτησίας των συναλλασσόμενων πόρων.

Θα πρέπει να αναφερθεί ότι σε ένα δημόσιο δίκτυο blockchain (π.χ. Bitcoin, Ethereum), σε αντίθεση με ό,τι θα ανέμενε κάποιος που γνωρίζει τώρα την τεχνολογία, *το περιεχόμενο των συναλλαγών δεν είναι κρυπτογραφημένο*. Αυτό συμβαίνει γιατί πρέπει να είναι δυνατή η ανάγνωσή του από όλους τους κόμβους του δικτύου, οι οποίοι και θα αναλάβουν την επιβεβαίωσή του και θα δώσουν (ή όχι) τη συναίνεσή τους για την εγκυρότητα της συναλλαγής.

Για να είναι όμως αποτελεσματική η χρήση της κρυπτογραφίας σε ένα δίκτυο blockchain, θα πρέπει αυτή να παρουσιάζει κάποια ιδιαίτερα χαρακτηριστικά. Έτσι, θα πρέπει:

- *Να είναι μιας κατεύθυνσης*: Δεν θα πρέπει το αποτέλεσμα της χρήσης μεθόδων κρυπτογραφίας να μπορεί να αντιστραφεί εύκολα. Αντιθέτως, θα πρέπει να είναι αδύνατον να επιτευχθεί αυτή η αντιστροφή.
- *Να είναι ντετερμινιστική*: Κάθε φορά που θα επαναλαμβάνεται η ίδια διαδικασία με χρήση της ίδιας εισόδου θα πρέπει να λαμβάνεται η ίδια έξοδος.
- *Να έχει μοναδικό αποτέλεσμα*: Η έξοδος που θα λαμβάνεται μετά από χρήση μιας κρυπτογραφικής μεθόδου θα πρέπει να είναι μοναδική και να επαναλαμβάνεται κάθε φορά που λαμβάνεται η ίδια είσοδος. Δεν θα πρέπει να υπάρχει και δεύτερη είσοδος που θα καταλήγει στην ίδια έξοδο με κάποια άλλη.
- *Να μπορεί να επιβεβαιωθεί*: Η επανάληψη της διαδικασίας μπορεί να χρησιμοποιηθεί (σε γραμμικό χρόνο) για την επιβεβαίωση της απόδοσης.

Λαμβάνοντας υπόψη τα χαρακτηριστικά αυτά, υπογραμμίζεται ότι όλες οι μέθοδοι που παρουσιάστηκαν στην Ενότητα 3.1 έχουν βρει τη χρήση τους σε ένα σύστημα blockchain, καθώς καταφέρνουν και παρουσιάζουν τα επιθυμητά αυτά χαρακτηριστικά.

Η ασύμμετρη κρυπτογραφία χρησιμοποιείται για τη δημιουργία του ζεύγους ιδιωτικού και δημόσιου κλειδιού. Η χρήση της ασύμμετρης κρυπτογραφίας προκρίθηκε για να δοθεί η δυνατότητα στον οποιονδήποτε να μπορεί να αποκτήσει ένα ιδιωτικό κλειδί και (από εκεί) μια διεύθυνση για να μπει σε ένα δίκτυο blockchain, χωρίς να υπάρχει ανάγκη για επικοινωνία με κάποια οντότητα που θα τον προμηθεύσει με το κλειδί της κρυπτογραφίας.

Πιο συγκεκριμένα, γίνεται χρήση της Κρυπτογραφίας Ελλειπτικών Καμπυλών (ECC) για την εύρεση του δημόσιου κλειδιού, ύστερα από την επιλογή του ιδιωτικού κλειδιού. Μάλιστα, η καμπύλη που χρησιμοποιείται ως αναφορά για την εύρεση του δημόσιου κλειδιού προέρχεται τόσο στο Bitcoin όσο και στο Ethereum από το ίδιο πρότυπο που έχει αναγνωριστεί από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των ΗΠΑ (National Institute of Standards and Technology, NIST), με την ονομασία secp256k1[36].

Η λύση αυτή ενσωματώνει όλα τα χαρακτηριστικά που αναφέρονται παραπάνω, επομένως καλύπτει τις προϋποθέσεις για χρήση σε ένα σύστημα blockchain. Θα πρέπει να δοθεί στην είσοδο του αλγόριθμου ένα ιδιωτικό κλειδί ίσο με 256 bits και, ακολούθως, παράγεται στην έξοδο ένα δημόσιο κλειδί πάλι ίσο με 256 bits.

[36] Online Σύνδεσμος: https://en.bitcoin.it/wiki/Secp256k1

Επιπρόσθετα, έχουν επιλεγεί και εφαρμόζονται συναρτήσεις κατακερματισμού οι οποίες και καλύπτουν τις παραπάνω απαιτήσεις. Συγκεκριμένα, το δίκτυο του Bitcoin χρησιμοποιεί δύο συναρτήσεις κατακερματισμού, τόσο για την εύρεση των διευθύνσεων από το δημόσιο κλειδί όσο και για τον υπολογισμό των hashes σε ένα block συναλλαγών και για την εφαρμογή του PoW. Οι συναρτήσεις κατακερματισμού του Bitcoin είναι ο Secure Hash Algorithm (SHA) και ο RACE Integrity Primitives Evaluation Message Digest (RIPEMD), πιο συγκεκριμένα οι SHA256 και RIPEMD160 (Antonopoulos, 2017, 2019). Ο αριθμός που συνοδεύει τους αλγόριθμους υποδηλώνει τον αριθμό των bits της εξόδου του αλγόριθμου. Έτσι, ο μεν SHA έχει έξοδο ίση με 256 bits, ενώ ο RIPEMD ίση με 160 bits. Για την παραγωγή μιας διεύθυνσης οι δύο αυτοί αλγόριθμοι χρησιμοποιούνται σειριακά, με την έξοδο του SHA256 να γίνεται είσοδος του RIPEMD160 (βλ. Ενότητα 3.4).

Αντίστοιχα, στο Ethereum η συνάρτηση κατακερματισμού είναι ο Keccak-256. Η ιστορία γύρω από τη συνάρτηση αυτή έχει το δικό της ενδιαφέρον. Η συνάρτηση είχε λάβει μέρος ως υποψήφια στον διαγωνισμό Κρυπτογραφικών Συναρτήσεων Κατακερματισμού SHA-3. Μάλιστα, είχε κερδίσει τον διαγωνισμό, αλλά στην πορεία έγιναν ορισμένες βελτιώσεις. Μιας και η διαδικασία καθυστερούσε, όταν το Ethereum έπρεπε να επιλέξει συνάρτηση, επέλεξε τον Keccak-256, αντί να αναμένει και να επιλέξει το πρότυπο SHA-3, που περιείχε κάποιες τροποποιήσεις από το NIST ως βελτιώσεις. Στην πράξη, συχνά αναφέρεται το SHA-3 ως το πρότυπο της συνάρτησης κατακερματισμού στο Ethereum, αν και στην πραγματικότητα πρόκειται για τον Keccak-256.

Όπως συμβαίνει και στην περίπτωση του Bitcoin, ο Keccak-256 δημιουργεί εξόδους οι οποίες είναι μεγέθους 256 bits. Και στα δύο δημοφιλή δίκτυα οι συναρτήσεις κατακερματισμού χρησιμοποιούνται για τη εύρεση των διευθύνσεων του δικτύου με είσοδο το δημόσιο κλειδί του χρήστη.

Τέλος, η κρυπτογραφία χρησιμοποιείται και με τη μορφή των ψηφιακών υπογραφών και στα δύο γνωστά δίκτυα που μελετώνται. Πιο συγκεκριμένα, γίνεται χρήση του αλγόριθμου *Elliptic Curve Digital Signature Algorithm* (*ECDSA*), ο οποίος χρησιμοποιεί τη μέθοδο ECC (που παρουσιάστηκε παραπάνω) για τη δημιουργία του ζεύγους των κλειδιών (ιδιωτικό – δημόσιο). Κατόπιν χρησιμοποιεί το ιδιωτικό κλειδί του χρήστη για να υπογράψει τη συναλλαγή (βλ. Κεφάλαιο 4).

Η υπογραφή αυτή παίζει τον ρόλο της απόδειξης ιδιοκτησίας, καθώς ο καθένας μπορεί να χρησιμοποιήσει το δημόσιο κλειδί του χρήστη που υπέγραψε ώστε να επιβεβαιώσει ότι ήταν αυτός που δημιούργησε (και υπέγραψε) τη συναλλαγή. Με αυτόν τον τρόπο εμποδίζεται ο κάθε χρήστης να δημιουργήσει μια συναλλαγή ξοδεύοντας τα χρήματα κάποιου άλλου χρήστη, καθώς η ψηφιακή υπογραφή είναι απαραίτητη ως απόδειξη ιδιοκτησίας των χρημάτων που μεταφέρονται. Δεδομένου ότι, για να είναι έγκυρη μια ψηφιακή υπογραφή, θα πρέπει αυτή να έχει χρησιμοποιήσει το ιδιωτικό κλειδί του χρήστη, δεν είναι δυνατόν για τον οποιονδήποτε να ξεγελάσει το σύστημα δημιουργώντας συναλλαγή εκ μέρους τρίτου.

Το παράδειγμα αυτό υπογραμμίζει τη σημασία της αποθήκευσης και προστασίας του ιδιωτικού κλειδιού σε ένα σύστημα blockchain, όπως τονίζεται συχνά και στη συνέχεια.

Στον **Πίνακα 3.3** φαίνονται οι επιλογές σε μεθόδους κρυπτογραφίας που έχουν γίνει από το Bitcoin και το Ethereum.

Κρυπτογραφία	**Bitcoin**	**Ethereum**
Μέθοδος Δημιουργίας Κλειδιού	Ασύμμετρη	Ασύμμετρη
Μέθοδος Δημιουργίας Δημόσιου Κλειδιού	ECC (secp256k1)	ECC (secp256k1)
Συνάρτηση Κατακερματισμού	SHA-256, RIPEMD160	Keccak-256
Ψηφιακές Υπογραφές	ECDSA	ECDSA

Πίνακας 3.3 *Μέθοδοι κρυπτογραφίας στο Bitcoin και το Ethereum.*

3.3 Κλειδιά

3.3.1 Η σημασία των κλειδιών σε ένα δίκτυο blockchain

Όπως ήδη αναφέρθηκε, για τη δημιουργία του ζεύγους ιδιωτικών και δημόσιων κλειδιών σε ένα δίκτυο blockchain γίνεται χρήση ασύμμετρης κρυπτογραφίας.

Τα κλειδιά ενός χρήστη παίζουν πάρα πολύ κρίσιμο ρόλο και αποτελούν πολύ σημαντικό κομμάτι της συμμετοχής των χρηστών στο δίκτυο.

Ενδεικτικά τονίζεται ότι με το ζεύγος των κλειδιών ένας χρήστης μπορεί:

- να αποκτήσει διεύθυνση και να μπει στο δίκτυο,
- να συμμετάσχει σε συναλλαγές στο δίκτυο στέλνοντας ή λαμβάνοντας ποσά από τη / στη διεύθυνσή του,
- να υπογράψει τις συναλλαγές του στο δίκτυο, αποδεικνύοντας ότι αυτός είναι ο κάτοχος των χρημάτων που μεταφέρονται.

Επομένως, γίνεται αντιληπτό ότι χωρίς τα κλειδιά, και ιδιαίτερα χωρίς το ιδιωτικό κλειδί, που είναι και το πιο βασικό κλειδί γιατί αποτελεί τη βάση για τον υπολογισμό των υπολοίπων (βλέπε και **Εικόνα 3.2**), ένας χρήστης αποκλείεται από τη συμμετοχή και τη διενέργεια συναλλαγών στο δίκτυο του blockchain.

Εικόνα 3.2 *Βήματα δημιουργίας δημόσιου κλειδιού και διεύθυνσης δικτύου, ξεκινώντας από την επιλογή του ιδιωτικού κλειδιού σε ένα δίκτυο blockchain.*

Για αυτό το ιδιωτικό κλειδί του χρήστη θα πρέπει να φυλάσσεται με προσοχή, να μην κοινοποιείται και να μη μοιράζεται με κανέναν. Όπως, αντίστοιχα, γίνεται με το όνομα χρήστη και τον κωδικό εισόδου στον τραπεζικό λογαριασμό, οι οποίοι και δεν διαμοιράζονται, τον ίδιο ρόλο και την ίδια αντιμετώπιση πρέπει να λαμβάνει σε ένα σύστημα blockchain το ιδιωτικό κλειδί του χρήστη. Συνήθως, τα κλειδιά ενός χρήστη δημιουργούνται και αποθηκεύονται μέσα σε ειδικές εφαρμογές που ονομάζονται *πορτοφόλια* (*wallets*).

Στη συνέχεια της ενότητας αυτής θα παρουσιαστούν ορισμένες λεπτομέρειες για τα πορτοφόλια, αλλά για πιο αναλυτικές πληροφορίες για τα είδη των πορτοφολιών καθώς και για τον τρόπο λειτουργίας τους δείτε την Ενότητα 3.5. Επίσης, στο Παράρτημα Α δίνονται οδηγίες για την εγκατάσταση και τη δημιουργία λογαριασμού στο δημοφιλές πορτοφόλι για το Ethereum «Metamask».

3.3.2 Η επιλογή του ιδιωτικού κλειδιού

Όπως αναφέρθηκε προηγουμένως (βλ. και Εικόνα 3.2), η δημιουργία (ή επιλογή) του ιδιωτικού κλειδιού αποτελεί τη βάση για τη συμμετοχή ενός χρήστη σε ένα δίκτυο blockchain.

Το ιδιωτικό κλειδί έχει μέγεθος ίσο με 256 bits ή 32 bytes και υπάρχουν διάφοροι τρόποι να δημιουργηθεί. Ο πιο απλός είναι σαν μια συμβολοσειρά από 256 bits. Η συμβολοσειρά αυτή γενικά μπορεί να είναι εκφρασμένη, εκτός από το δυαδικό, σε δεκαεξαδική μορφή (αποτελούμενη από 64 δεκαεξαδικά στοιχεία) ή να είναι σε μορφή Base64[37]. Μάλιστα, υπάρχουν διάφοροι τρόποι με τους οποίους μπορεί να δημιουργηθεί η σειρά αυτή από έναν χρήστη (σε οποιαδήποτε από τις προαναφερθείσες μορφές).

Ένα παράδειγμα ενός ιδιωτικού κλειδιού εκφρασμένο και στις τρεις παραπάνω μορφές είναι το ακόλουθο:

Bits:*0001011011110011100110110001100011100101101011011011111111010001100101000111001110000110100110000100010011010110011011010110110001001111011111001101100001000010111011000000010010010110011100110011001010111100110011111001010001111101111000010100111101111*
Δεκαεξαδικό: *16f39b18e5adbfa328e70d3089acdad89ef3610bb0124ce65995e67947de14f7*

Base64: *FvObGOWtv6Mo5w0wiaza2J7zYQuwEkzmWZXmeUfeFPc=*

[37] Πρόκειται για ένα σχήμα κωδικοποίησης bits σε κείμενο, το οποίο αντικαθιστά σειρές από 24 bits με τέσσερα ψηφία του συστήματος Base64 (των 6-bits). Τα 64 ψηφία του συστήματος επιλέγονται με την προϋπόθεση να μπορούν να εκτυπωθούν. Υπάρχουν διάφορες υλοποιήσεις. Η Base64 υλοποίηση της MIME περιέχει τα γράμματα A-Z, a-z και τους αριθμούς 0-9. Αυτά είναι 62 ψηφία σύνολο (26 + 26 +10). Στη συνέχεια, τα επόμενα 2 ψηφία μπορούν να αλλάζουν ανά υλοποίηση. Συχνά χρησιμοποιούνται οι χαρακτήρες («+») και «/».

Ο πιο εύκολος τρόπος για την επιλογή του ιδιωτικού κλειδιού είναι να δημιουργηθεί ένας λογαριασμός σε μια εφαρμογή πορτοφολιού η οποία θα δημιουργήσει τα κλειδιά και θα τα αποθηκεύσει. Ο χρήστης μπορεί να μη μάθει ποτέ ποιο είναι το ιδιωτικό του κλειδί, εκτός αν θελήσει να μεταφέρει το πορτοφόλι του σε άλλη συσκευή, οπότε θα χρειαστεί να γνωρίζει τα κλειδιά του για να τα περάσει στη νέα αυτή συσκευή. Συχνά, για τη μεταφορά αυτή το ιδιωτικό κλειδί μετασχηματίζεται σε μια μνημονική φράση 24 λέξεων, που πρέπει να σημειώσει ο χρήστης για να επιτύχει τη μεταφορά αυτή (βλ. Ενότητα 3.5).

Εναλλακτικά, θα μπορούσε η επιλογή των 256 bits να γίνει τυχαία από τον χρήστη δημιουργώντας μία τυχαία σειρά από 256 ψηφία, επιλέγοντας μόνος του την τιμή του επόμενου bit (0 ή 1)[38]. Ούτε αυτός ο τρόπος είναι δύσκολος, αλλά χρειάζεται χρόνος για να ολοκληρωθεί και είναι αρκετά επιρρεπής σε λάθη κατά την αντιγραφή του κλειδιού μετά τη δημιουργία του. Βέβαια, πρόκειται για έναν τρόπο ο οποίος εξασφαλίζει ότι δεν είναι γνωστό από κάποιον άλλον το κλειδί που δημιουργείται (αρκεί να είστε μόνοι σας στο δωμάτιο όταν σημειώνεται τα bits)!

Τρίτη επιλογή είναι η χρήση μιας συνάρτησης παραγωγής τυχαίων αριθμών. Δυστυχώς, οι περισσότερες γλώσσες προγραμματισμού, ενώ έχουν ενσωματώσει συναρτήσεις παραγωγής τυχαίων αριθμών, στην ουσία δεν είναι πραγματικά τυχαίοι, αλλά έχουν μια προκαθορισμένη αλληλουχία βάσει της αρχικής τους παραμέτρου εισόδου (γνωστή και ως seed στον προγραμματισμό). Η χρήση μιας τέτοιας συνάρτησης δεν ενδείκνυται για την επιλογή του ιδιωτικού κλειδιού, γιατί είναι εύκολο να βρεθεί, καθώς αρκεί να χρησιμοποιηθεί ο πρώτος αριθμός και στη συνέχεια θα βρεθούν εύκολα τα επόμενα νούμερα.

Για τη δημιουργία του ιδιωτικού κλειδιού χρειάζεται να υπάρχει αυξημένη εντροπία, και για τον λόγο αυτόν έχουν δημιουργηθεί ειδικές κρυπτογραφικές συναρτήσεις σε κάποιες γλώσσες προγραμματισμού (π.χ. στην Python η βιβλιοθήκη *secrets*), οι οποίες παίρνουν μια είσοδο εντροπίας η οποία λαμβάνεται από το λειτουργικό σύστημα και η οποία μπορεί να προέρχεται από: τις διεργασίες που τρέχουν στον επεξεργαστή, από την κίνηση του ποντικιού, από την κάμερα ή το μικρόφωνο, ή ακόμα και από έναν συνδυασμό από όλα αυτά. Με τον τρόπο αυτόν εξασφαλίζεται ότι η αρχική κατάσταση δεν είναι ευκόλως προβλέψιμη ούτε αναπαραγώγιμη.

Εν κατακλείδι, στην περίπτωση που κάποιος χρησιμοποιήσει μια συνάρτηση προγραμματισμού για τη δημιουργία του ιδιωτικού κλειδιού θα πρέπει να προσέξει να μην κάνει χρήση μιας απλής συνάρτησης παραγωγής τυχαίων αριθμών, αλλά μιας συνάρτησης παραγωγής κρυπτογραφικά τυχαίων αριθμών. Η τελευταία κάνει χρήση και μιας στιγμιαίας πηγής εντροπίας για να προσδιορίσει το αρχικό seed και να παραγάγει τους τυχαίους αριθμούς.

Τέλος, υπάρχει και ένας ακόμα τρόπος. Μπορεί κάποιος να χρησιμοποιήσει μια λέξη-φράση ως είσοδο σε μια συνάρτηση κατακερματισμού (όπως ο SHA-256) και να λάβει την έξοδο με μέγεθος 256 bits, το οποίο και θα είναι το ιδιωτικό του κλειδί. Αυτή η μέθοδος χρησιμοποιήθηκε και για την παραγωγή του ιδιωτικού κλειδιού που παρουσιάστηκε νωρίτερα. Το αποτέλεσμα πέρασε και από εργαλεία[39] μετατροπής στα υπόλοιπα συστήματα για την εύρεση των αποτελεσμάτων.

Ανεξάρτητα του τρόπου παραγωγής του ιδιωτικού κλειδιού, θα πρέπει ο καθένας να είναι προσεκτικός και να μην το διαμοιράζεται, καθώς αποτελεί τον πιο σημαντικό παράγοντα για τη συμμετοχή του στον κόσμο του blockchain.

3.3.3 Η δημιουργία του δημόσιου κλειδιού

Όπως φαίνεται και στην Εικόνα 3.2, το δημόσιο κλειδί υπολογίζεται από το ιδιωτικό κλειδί του χρήστη με μοναδικό, μη αντιστρεπτό τρόπο. Τόσο το Bitcoin όσο και το Ethereum χρησιμοποιούν για να υπολογίσουν το δημόσιο κλειδί λύσεις που ανήκουν στις μεθόδους Κρυπτογραφίας Ελλειπτικών Καμπυλών, και ιδιαίτερα το πρότυπο secp256k1.

Το πρότυπο secp256k1 χρησιμοποιεί την ελλειπτική καμπύλη που φαίνεται στην **Εικόνα 3.3**, η οποία ορίζεται από τη συνάρτηση:

$$y^2 = (x^3 + 7)\ στο\ (\mathbb{F}_p) \qquad (3.1)$$

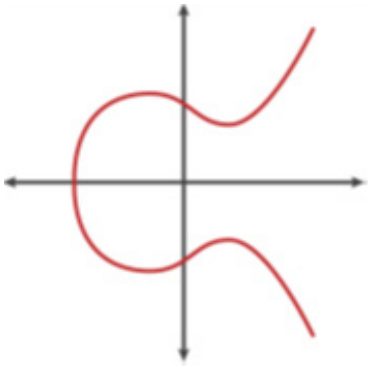
[38] Ο τελικός αριθμός που θα παραχθεί θα πρέπει να είναι μεγαλύτερος του 1 και μικρότερος από το 2^{256}. Για την ακρίβεια, πρέπει να είναι μικρότερος από το n-1, όπου n = 1,15-10^{77}, που είναι λίγο μικρότερο από 2^{256}.

[39] Online Σύνδεσμος: https://emn178.github.io/online-tools/base64_encode.html

ή

$$y^2 \ mod \ p = \ (x^3 + 7) \ mod \ p \qquad (3.2)$$

Η καμπύλη αυτή, μιας και χρησιμοποιείται στην κρυπτογραφία, πρέπει να ορίζεται μέσα σε ένα πεπερασμένο πεδίο. Έτσι και στον secp256k1 η χρήση του *mod p* δηλώνει ότι καμπύλη ορίζεται στο πεπερασμένο πεδίο των πρώτων αριθμών τάξης p (γραμμένο και ως $\mathbb{F}_p$ πιο πάνω), όπου το p είναι ίσο με: $p = 2^{256}-2^{32}-2^9-2^8-2^7-2^6-2^4-1$, που αποτελεί έναν πολύ μεγάλο πρώτο αριθμό, κοντά στο 2^{256}.



Εικόνα 3.3 *Η ελλειπτική καμπύλη του προτύπου secp256k1 (Πηγή: Antonopoulos, 2017).*

Το δημόσιο κλειδί δεν είναι τίποτα άλλο από ένα σημείο πάνω σε αυτή την καμπύλη. Για να βρεθεί, όμως, το σημείο αυτό, θα πρέπει να εκτελεστεί η πράξη του πολλαπλασιασμού στην ελλειπτική καμπύλη, πράξη η οποία έχει κάποιες ιδιαιτερότητες.

Ο πολλαπλασιασμός αυτός δηλώνεται από την εξίσωση P = p * G, όπου:

- P είναι το δημόσιο κλειδί,
- p είναι το ιδιωτικό κλειδί και G είναι ένα σημείο βάσης (ή γεννήτορας).

Θα πρέπει να τονιστεί ότι γνώση του G και του P δεν είναι δυνατόν να οδηγήσει στο p, καθώς η διαίρεση που πρέπει να γίνει για να υπολογιστεί σχετίζεται με το πρόβλημα της εύρεσης του διακριτού λογάριθμου. Στην ουσία, αυτό σημαίνει ότι απαιτείται η δοκιμή όλων των αριθμών για την εύρεση εκείνου που χρησιμοποιήθηκε ως συντελεστής στον πολλαπλασιασμό.

Επιπλέον, το σημείο βάσης G αποτελεί το σημείο εκκίνησης πάνω στην ελλειπτική καμπύλη και δηλώνεται στο πρότυπο του secp256k1 ίσο με:

G=040x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8 (σε μη συμπιεσμένη μορφή[40]).

Ξεκινώντας από το (κοινό) σημείο γεννήτορα G, αυτό θα πρέπει να πολλαπλασιαστεί τόσες φορές όσες είναι και το αρχικό ιδιωτικό κλειδί του χρήστη. Αυτό ισοδυναμεί με την πρόσθεση του κοινού σημείου βάσης (γεννήτορας) με τον εαυτό του p φορές:

$$P = p * \underbrace{G = G + G + G + \ldots. + G}_{p \text{ φορές}} \qquad (3.3)$$

Ενδεικτικά αναφέρεται πως για
p = 16f39b18e5adbfa328e70d3089acdad89ef3610bb0124ce65995e67947de14f7, όπως και παρουσιάστηκε στην Υποενότητα 3.3.2 (δεκαεξαδική μορφή απεικόνισης του ιδιωτικού κλειδιού), τότε ο πολλαπλασιασμός του αριθμού αυτού με το G (όπως δίνεται παραπάνω) έχει ως αποτέλεσμα:

P=040x6b84bd1f0180d83b415b3e237be5c6275e3808c2ede1a7b94f76bed333a7722e4c7361f3b5cb0badbef9c56f6a2b081f0d3fa8a0411a4f85761e25440c82c85b

[40] Η χρήση συμπιεσμένης μορφής αφορά αποκλειστικά το Bitcoin.

Για την επιβεβαίωση των πράξεων μπορεί να χρησιμοποιηθεί το online εργαλείο στον σύνδεσμο στην αναφορά[41]. Εκεί θα μπει το κλειδί p και επιλέγοντας secp256k1 ecdsa (μιας και είναι ο αλγόριθμος που χρησιμοποιείται στο Bitcoin και Ethereum και υποδηλώνει τη χρήση ψηφιακών υπογραφών σε κλειδιά που προήλθαν από πολλαπλασιασμό ελλειπτικής καμπύλης) θα βρεθεί η τιμή στην έξοδο 64b hex. Τα 64 bytes υποδηλώνουν μη συμπιεσμένη έξοδο. Τα 33 bytes υποδηλώνουν συμπιεσμένη[42], η οποία επίσης φαίνεται στον πίνακα και αφορά αποκλειστικά το δίκτυο του Bitcoin.

Ο **Πίνακας 3.4** έχει τις τιμές που χρησιμοποιήθηκαν στο παράδειγμα που μελετάται για αναφορά:

Παράμετρος	**Τιμές (Hex)**
p	16f39b18e5adbfa328e70d3089acdad89ef3610bb0124ce65995e67947de14f7
G (μη συμπ.)	0479BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798483ADA7726A3C4655DA4FBFC0E1 108A8FD17B448A68554199C47D08FFB10D4B8
G (συμπ.)	0279BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798
P (μη συμπ.)	046b84bd1f0180d83b415b3e237be5c6275e3808c2ede1a7b94f76bed333a7722e4c7361f3b5cb0badbef9c56f6a2b081f0d3fa8a0411 a4f85761e25440c82c85b
P (συμπ.)	036b84bd1f0180d83b415b3e237be5c6275e3808c2ede1a7b94f76bed333a7722e

Πίνακας 3.4 *Αποτελέσματα χρήσης του secp256k1 για την εύρεση ενός δημόσιου κλειδιού.*

Στον Πίνακα 3.4 έχουν τονιστεί με έντονο χρώμα τα προθέματα που χρησιμοποιούνται (1 byte ή 2 δεκαεξαδικοί χαρακτήρες) για να προσδιορίσουν τη μορφή στην οποία εκφράζεται το δημόσιο κλειδί και το σημείο βάσης.

Ο **Πίνακας 3.5** δείχνει την εξήγηση του byte αυτού[43]:

Πρόθεμα	**Εξήγηση**	**Μήκος σε bytes για το Δημόσιο Κλειδί**
04	Σημείο χωρίς συμπίεση	65
03	Συμπιεσμένο σημείο με άρτιο y	33
02	Συμπιεσμένο σημείο με περιττό y	33

Πίνακας 3.5 *Εξήγηση των προθεμάτων για τη μορφή του δημόσιου κλειδιού στο Bitcoin.*

Κατόπιν, η πράξη της πρόσθεσης δύο σημείων (A και B) σε μια ελλειπτική καμπύλη περιλαμβάνει τη χάραξη της ευθείας που ενώνει τα δύο σημεία και τον προσδιορισμό του τρίτου σημείου (K) στο οποίο η ευθεία αυτή τέμνει την καμπύλη. Έστω ότι οι συντεταγμένες του σημείου εκείνου είναι (x, y), τότε το αποτέλεσμα της πρόσθεσης των A και B είναι το (x, -y), όπου το -y είναι το συμμετρικό του y ως προς τον άξονα x.

Με τη λογική αυτή, και καθώς η πράξη του πολλαπλασιασμού ενός σημείου με έναν ακέραιο αριθμό μπορεί να αντικατασταθεί από πρόσθεση του ίδιου σημείου με τον εαυτό του, το πρόβλημα μεταφέρεται στην εύρεση της ευθείας εκείνης που περνά από ένα σημείο στην καμπύλη. Καθώς οι ευθείες αυτές είναι άπειρες, επιλέγεται να χρησιμοποιηθεί η εφαπτόμενη ευθεία στο σημείο.

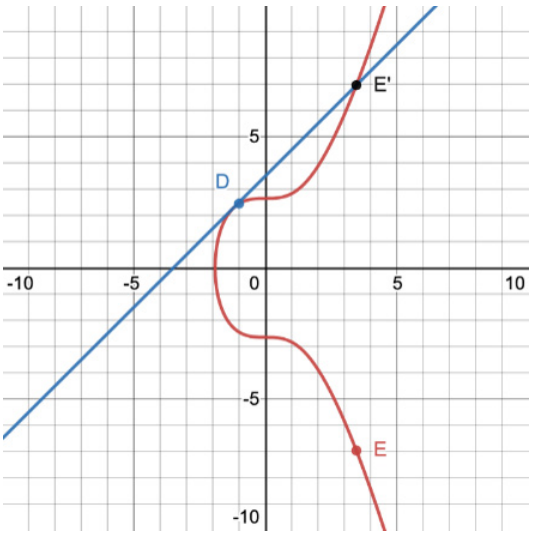
Στην **Εικόνα 3.4** φαίνεται το αποτέλεσμα του πολλαπλασιασμού 2*D = D + D, δηλαδή της πρόσθεσης του D με τον εαυτό του. Το σημείο E΄ είναι το ζητούμενο (συμμετρικό του E).

Στη συνέχεια, εφόσον το p > 2, μπορείτε να υπολογίσετε με τον ίδιο τρόπο το σημείο 4*D = 2*D + 2* D = E΄ + E΄ κ.ο.κ.

[41] Online Σύνδεσμος: https://paulmillr.com/noble/

[42] Η διαφορά μεταξύ συμπιεσμένης και μη συμπιεσμένης εξόδου είναι ότι η πρώτη περιέχει μόνο τη συντεταγμένη του άξονα x (που χρησιμοποιείται για τον υπολογισμό του άξονα y, όποτε χρειαστεί), ενώ η δεύτερη έχει και τις δύο συντεταγμένες του σημείου.

[43] Το Ethereum χρησιμοποιεί μόνο μη συμπιεσμένες τιμές στα δημόσια κλειδιά του, οπότε στο δίκτυο αυτό το πρόθεμα του δημόσιου κλειδιού θα είναι πάντα 04.



Εικόνα 3.4 *Το αποτέλεσμα της πράξης D +D = E σε μια ελλειπτική καμπύλη (Πηγή: Antonopoulos, 2017).*

Το πρότυπο secp256k1 λαμβάνει μια είσοδο ίση με το ιδιωτικό κλειδί μήκους 256 bits (32 bytes) και δημιουργεί μια έξοδο 520 bits (65 bytes), που αποτελείται από το σημείο στην καμπύλη (64 bytes, 32 bytes το μήκος της κάθε συντεταγμένης) και το πρόθεμα μήκους 1 byte (βλ. Πίνακα 3.5), στην περίπτωση που οι υπολογισμοί γίνονται χωρίς συμπίεση, όπως συμβαίνει στο δίκτυο του Ethereum.

3.4 Διευθύνσεις

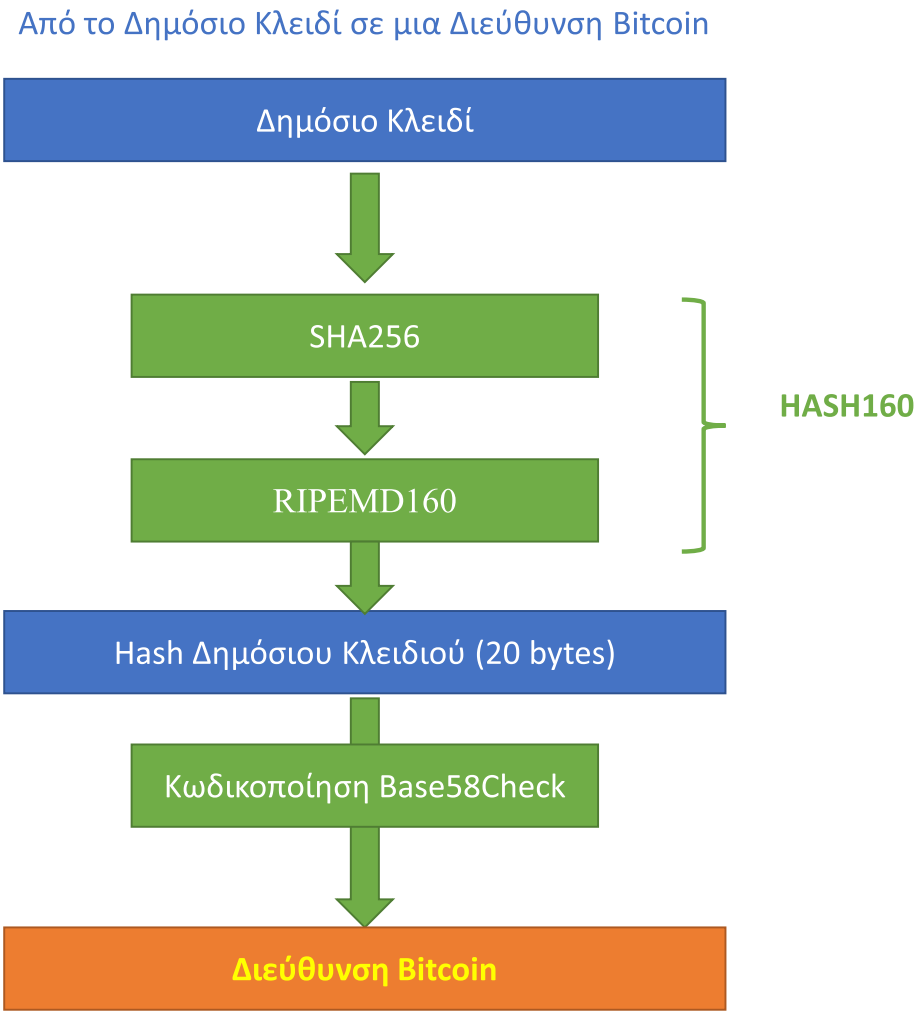
Η διεύθυνση σε ένα δίκτυο blockchain αποτελεί έναν ακόμα χαρακτηριστικό, μοναδικό αριθμό με τον οποίο μπορεί να αναγνωριστεί και να συμμετέχει κάποιος μέσα στο δίκτυο αυτό παίρνοντας μέρος σε συναλλαγές προς ή από τον αριθμό αυτόν.

Η διεύθυνση προέρχεται από το δημόσιο κλειδί του χρήστη με χρήση μονόδρομων συναρτήσεων κατακερματισμού τόσο στο δίκτυο του Ethereum όσο και σε αυτό του Bitcoin (Εικόνα 3.2). Τα δύο δίκτυα χρησιμοποιούν παρόμοια προσέγγιση με μικρές διαφορές στην υλοποίηση, όπως αναλύεται στη συνέχεια.

Bitcoin: Στο δίκτυο του Bitcoin για τον υπολογισμό της διεύθυνσης γίνεται χρήση ενός ζεύγους συναρτήσεων κατακερματισμού (SHA256 και RIPEMD160).

Αρχικά, το δημόσιο κλειδί (όπως υπολογίστηκε από την ελλειπτική καμπύλη με σημείο εκκίνησης τον γεννήτορα και το ιδιωτικό κλειδί να υποδεικνύει τον αριθμό των μετακινήσεων) μπαίνει στη συνάρτηση κατακερματισμού SHA256 και προκύπτει ένα hash ίσο με 256 bits (34 bytes). Κατόπιν, το hash αυτό μπαίνει ως είσοδος σε μια δεύτερη συνάρτηση κατακερματισμού, την RIPEMD160, και παράγεται ένα νέο hash ίσο με 160 bits (20 bytes). Το αποτέλεσμα είναι να δημιουργηθεί μια διεύθυνση του δικτύου Bitcoin.

Η διαδικασία αυτή φαίνεται στην **Εικόνα 3.5**.



Εικόνα 3.5 *Ο υπολογισμός μιας διεύθυνσης του δικτύου Bitcoin με αφετηρία το δημόσιο κλειδί ενός χρήστη.*

Όμως δεν είναι αυτή η τελική μορφή της διεύθυνσης που χρησιμοποιείται. Αντιθέτως, έπεται και ένα ακόμα βήμα, στο οποίο η διεύθυνση θα λάβει μια πιο φιλική προς τον χρήστη μορφή, ακολουθώντας την κωδικοποίηση *Base58Check*. Σκοπός είναι να υπάρχει μια επιπλέον δικλίδα ασφαλείας που να αποτρέπει τυχόν λάθη στη συγγραφή της διεύθυνσης.

Στην Υποενότητα 3.3.2 παρουσιάστηκε η κωδικοποίηση Base64 και αναφέρθηκε το σύνολο των χαρακτήρων από τους οποίους αποτελείται. Η κωδικοποίηση Base58 μοιάζει πολύ με αυτήν, για την ακρίβεια είναι ίδια, με την αφαίρεση ορισμένων αλφαριθμητικών: το μηδέν (0), το κεφαλαίο όμικρον (O), το λατινικό γράμμα l και το κεφαλαίο λατινικό γράμμα i (I) μαζί και με τα 2 σύμβολα που συμπεριλαμβάνονται στο αλφάβητο της Base64 (το «+» και το «/»).

Οι λόγοι που αφαιρούνται αυτά τα σύμβολα από τις διευθύνσεις του Bitcoin είναι ότι, συχνά, σε διαφορετικές γραμματοσειρές τα σύμβολα αυτά μπορεί να προκαλέσουν σύγχυση και έτσι να αυξήσουν την πιθανότητα να δοθεί μια λάθος διεύθυνση.

Εν ολίγοις, το αλφάβητο της Base58 είναι το:

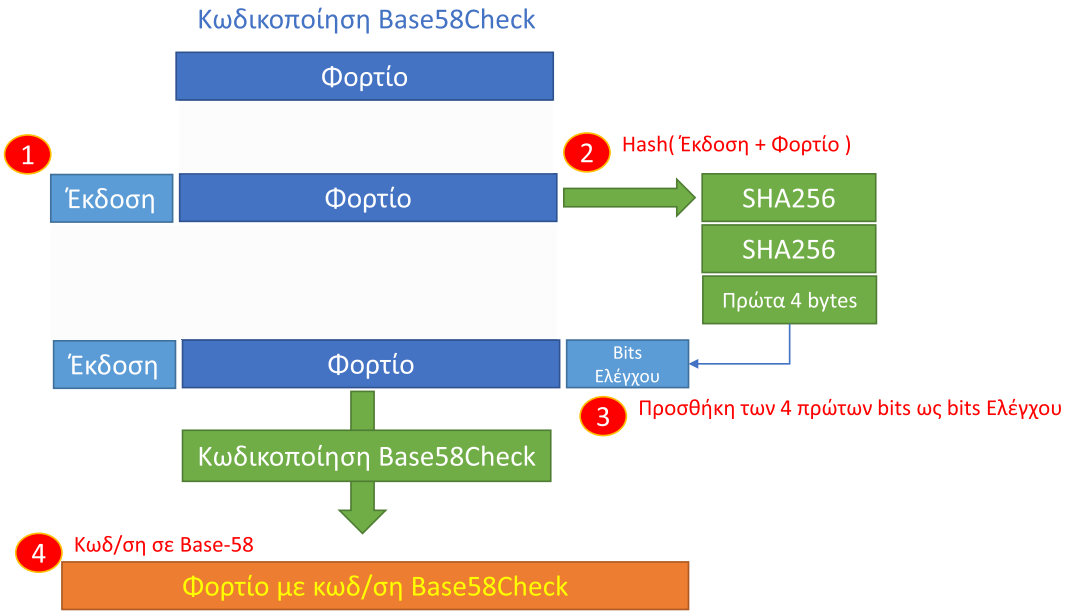
123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz

Η κωδικοποίηση Base58Check έχει και έναν ακόμα βαθμό ελέγχου για την αποφυγή λαθών. Η κωδικοποίηση Base58Check περιλαμβάνει:

- Hash του δημόσιου κλειδιού ως δεδομένα.
- Χρήση των 58 αλφαριθμητικών χαρακτήρων.

- Προσθήκη ενός byte μπροστά ως πρόθεμα, που δηλώνει την έκδοση. Οι αρχικές διευθύνσεις στο Bitcoin χρησιμοποιούσαν το 0x00, και αυτό οδηγούσε στον χαρακτηριστικό άσο (1), που είναι ο αριθμός από τον οποίο ξεκινούν οι διευθύνσεις στο Bitcoin.
- Χρήση 4 bytes για έλεγχο στο τέλος. Τα bytes αυτά θα χρησιμοποιούνται για την επιβεβαίωση της ορθότητας της διεύθυνσης προς αποφυγή συντακτικών λαθών.

Όπως φαίνεται και στην **Εικόνα 3.6**, για την επιλογή των 4 bytes που προστίθενται στο τέλος για έλεγχο ορθότητας λαμβάνονται υπόψη το πρόθεμα του ενός byte μαζί με το hash του δημόσιου κλειδιού. Το ζεύγος αυτό θα χρησιμοποιηθεί ως είσοδος στη συνάρτηση κατακερματισμού SHA256 που χρησιμοποιείται στο Bitcoin. Το αποτέλεσμα της εξόδου ανατροφοδοτεί τη συνάρτηση κατακερματισμού SHA256. Από το τελικό hash των 34 bytes κρατούνται τα 4 πρώτα bytes, και αυτά μπαίνουν ως bytes ελέγχου στο τέλος της διεύθυνσης. Στη συνέχεια, το συνολικό αποτέλεσμα κωδικοποιείται χρησιμοποιώντας το αλφάβητο της κωδικοποίησης Base58. Το τελικό αποτέλεσμα της διεύθυνσης θα πρέπει να ξεκινά από 1, καθώς αυτή είναι η μετάφραση των μηδενικών στο πρώτο byte (byte έκδοσης).



Εικόνα 3.6 *Ο μηχανισμός ελέγχου που εισάγεται με την κωδικοποίηση Base58Check.*

Κατά την επαλήθευση της διεύθυνσης, το λογισμικό αποκωδικοποίησης μπορεί να υπολογίσει από το δημόσιο κλειδί και το byte της έκδοσης το παραγόμενο hash, τροφοδοτώντας αντίστοιχα δύο φορές τη συνάρτηση κατακερματισμού SHA256 με το αποτέλεσμα και συγκρίνοντας τα 4 πρώτα bytes από τους υπολογισμούς με τα bytes ελέγχου της διεύθυνσης. Αυτός είναι ένας τρόπος που χρησιμοποιήθηκε στο Bitcoin για την αποφυγή των συντακτικών λαθών κατά την απόδοση των διευθύνσεων του δικτύου.

Στην πράξη αποδείχθηκε πολύ χρήσιμη προσθήκη, καθώς απέτρεψε αρκετά λάθη, τα οποία μπορούσαν να είχαν οδηγήσει στην απώλεια αρκετών bitcoins κατά τη μεταφορά τους σε άγνωστες διευθύνσεις.

Ethereum: Σε αντιστοιχία με το τι συμβαίνει στο Bitcoin, στο Ethereum οι διευθύνσεις έχουν τον ρόλο των μοναδικών χαρακτηριστικών που χρησιμοποιούνται κατά τις συναλλαγές στο δίκτυο. Επίσης, οι διευθύνσεις αυτές προέρχονται από το δημόσιο κλειδί με τη βοήθεια μιας μονόδρομης συνάρτησης κατακερματισμού. Η διαφορά σχετίζεται με την επιλογή της συνάρτησης αυτής, καθώς στο Ethereum έχει επιλεγεί η συνάρτηση Keccak-256 (βλ. Ενότητα 3.2). Έτσι, το δημόσιο κλειδί θα μπει ως είσοδος στο Keccak-256, και από την έξοδό του θα κρατηθούν τα τελευταία 20 bytes (160 bits), τα οποία και αποτελούν τη διεύθυνση στο δίκτυο του Ethereum.

Στον **Πίνακα 3.6**, που ακολουθεί, φαίνονται οι τιμές που λαμβάνουν τα ζεύγη των κλειδιών και η διεύθυνση στο δίκτυο του Ethereum που παράγεται για αυτά για την περίπτωση του ζεύγους κλειδιών που αναγράφεται και στον Πίνακα 3.4.

Παράμετρος	Τιμές (Hex)
p	16f39b18e5adbfa328e70d3089acdad89ef3610bb0124ce65995e67947de14f7
P	046b84bd1f0180d83b415b3e237be5c6275e3808c2ede1a7b94f76bed333a7722e4c7361f3b5cb0badbef9c56f6a2b081f0d3fa8a0411a4f85761e25440c82c85b
Διεύθυνση	0x9150d2416dcd4e0aa3b4d63a5aa0690d2de60853

Πίνακας 3.6 *Υπολογισμός της διεύθυνσης στο δίκτυο Ethereum για το ζεύγος κλειδιών του Πίνακα 3.5.*

Συνήθως, στο Ethereum μπροστά από τη διεύθυνση θα βρεθεί το πρόθεμα *0x*, για να δηλώσει ότι αυτή είναι εκφρασμένη στο δεκαεξαδικό σύστημα.

Επιπλέον, στο Ethereum δεν ακολουθήθηκε η λογική της πρόσθετης ασφάλειας που χρησιμοποιεί το Bitcoin με τη χρήση της κωδικοποίησης Base58Check και την προσθήκη μικρού αριθμού bytes ελέγχου στο τέλος. Αυτό συνέβη γιατί θεωρήθηκε ότι μια τέτοια λύση θα μπορούσε να ενσωματωθεί σε πιο υψηλά στρώματα στην αρχιτεκτονική του συστήματος. Και αυτό γιατί αρχικά αναμενόταν ότι θα γινόταν χρήση υπηρεσιών ονοματοδοσίας που θα αντιστοιχούσαν τους αριθμούς των διευθύνσεων με ονόματα, για διευκόλυνση.

Δυστυχώς, όμως, αυτό καθυστέρησε πολύ να ολοκληρωθεί, με αποτέλεσμα στο ενδιάμεσο να υπάρχουν αρκετά λάθη στους αριθμούς των διευθύνσεων (π.χ. λόγω κακής αντιγραφής).

3.5 Πορτοφόλια (Wallets)

Τα πορτοφόλια παίζουν πολύ σημαντικό ρόλο σε ένα δίκτυο blockchain. Στο Κεφάλαιο 2 παρουσιάστηκαν τα βήματα (συναλλαγές) για την αποστολή και λήψη bitcoins με τη βοήθεια μιας σύγχρονης εφαρμογής πορτοφολιού.

Τα πορτοφόλια αποτελούν το μέσο σύνδεσης με ένα δίκτυο blockchain και γενικά παρουσιάζουν τις εξής ιδιότητες:

- Ευκολία στη χρήση.
- Αυξημένη ασφάλεια.
- Δυνατότητα για γρήγορες συναλλαγές σε παγκόσμιο επίπεδο.
- Μικρά κόστη για την πραγματοποίηση των συναλλαγών.
- Δημιουργία συναλλαγών σε πολλά διαφορετικά δίκτυα blockchain, επιτρέποντας τη σύνδεση σε αυτά και εξυπηρετώντας συναλλαγές με διάφορα κρυπτονομίσματα.

Στη συνέχεια αναλύονται τα διαφορετικά είδη πορτοφολιών, ανάλογα με τον τρόπο λειτουργίας τους και τα ιδιαίτερα χαρακτηριστικά τους.

3.5.1 Ντετερμινιστικά ή μη ντετερμινιστικά πορτοφόλια

Σε αντίθεση με την αίσθηση της πλειοψηφίας των χρηστών, τα πορτοφόλια *δεν χρησιμοποιούνται για την αποθήκευση των κρυπτονομισμάτων*. Αντιθέτως, το πορτοφόλι του κάθε χρήστη έχει αποθηκευμένα τα κλειδιά του. Αυτά μπορούν να χρησιμοποιηθούν για να υπογράψουν συναλλαγές για τη μεταφορά κρυπτονομισμάτων ή να δημιουργήσουν διευθύνσεις για την παραλαβή τους. Επομένως, το πορτοφόλι δίνει πρόσβαση στα χρήματα μέσω της κατοχής των κλειδιών του χρήστη, αλλά δεν έχει τα χρήματα. Αυτά βρίσκονται στο δίκτυο του blockchain και όχι σε κάποιον κόμβο ή χρήστη/πρόγραμμα.

Επίσης, να αναφερθεί ότι τα σύγχρονα πορτοφόλια έχουν παραπάνω από ένα ζεύγος ιδιωτικού/δημοσίου κλειδιού (και κατ' επέκταση και διευθύνσεις). Αυτό συμβαίνει γιατί έχει κριθεί προτιμότερο, για λόγους ιδιωτικότητας, όταν χρησιμοποιηθεί μια διεύθυνση για αποστολή ή λήψη κρυπτονομισμάτων, να μη χρησιμοποιείται συχνά στη συνέχεια.

Αυτό, βέβαια, δεν σημαίνει ότι ο κάτοχος δεν έχει πρόσβαση στα κρυπτονομίσματα. Απλώς, για λόγους ιδιωτικότητας (μιας και, όπως έχει αναφερθεί, οι συναλλαγές είναι δημόσιες και μη κρυπτογραφημένες) το πορτοφόλι του χρήστη γεννά νέα κλειδιά (από ένα κυρίως κλειδί που χρησιμοποιείται ως αναφορά) για να εξυπηρετήσει αυτή την ανάγκη.

Ταυτόχρονα, το πορτοφόλι αναλαμβάνει να παρακολουθεί τις συναλλαγές του χρήστη σε όλα τα κλειδιά που παράγει, για να ενημερώνει το ποσό που υπάρχει σε κάθε διεύθυνση και, ανάλογα, να το υπολογίζει σε μελλοντικές συναλλαγές.

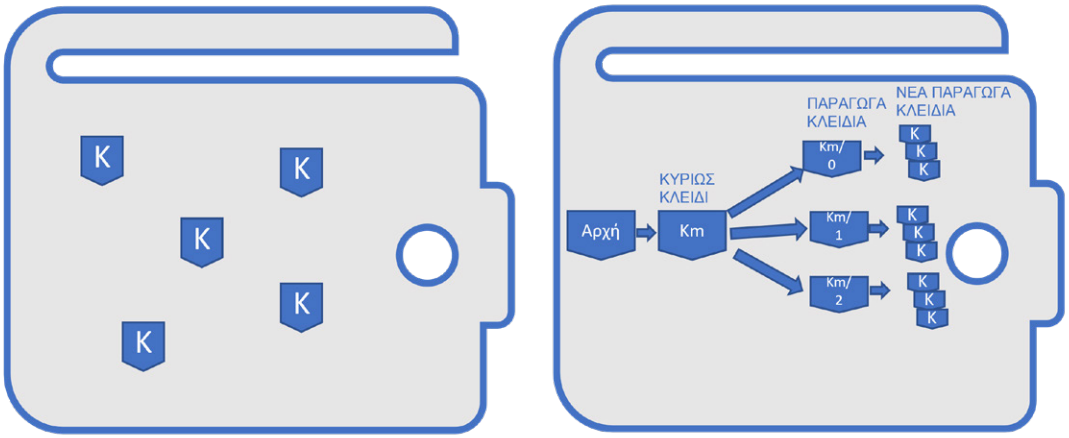
Δεδομένης της αποθήκευσης των κλειδιών στα πορτοφόλια, ο τρόπος με τον οποίο δημιουργούνται τα κλειδιά σε αυτά χαρακτηρίζει και τον τύπο του πορτοφολιού. Έτσι, δύο είναι οι βασικοί τύποι πορτοφολιών, με μια χαρακτηριστική διαφορά μεταξύ τους που αναφέρεται στη σχέση που υπάρχει ανάμεσα στα κλειδιά που δημιουργούν:

- τα μη ντετερμινιστικά πορτοφόλια και
- τα ντετερμινιστικά

Τα πρώτα (μη ντετερμινιστικά) έχουν το χαρακτηριστικό ότι το κάθε κλειδί δημιουργείται από έναν διαφορετικό τυχαίο αριθμό, που δεν σχετίζεται με κάποιον προηγούμενο και άρα δεν μπορεί να αναπαραχθεί[44] σε μια αλληλουχία. Αν και αποτελεί μια πολύ ασφαλή μέθοδο, η έλλειψη δυνατότητας αναπαραγωγής των κλειδιών μειώνει τη φορητότητα του πορτοφολιού. Δηλαδή δεν είναι εύκολο το να μεταφερθεί σε έναν άλλο υπολογιστή (ή κάποιο κινητό).

Σε αντίθεση με αυτά, τα ντετερμινιστικά παράγουν ένα κύριο κλειδί, συχνά με τη βοήθεια ενός seed, και από αυτό στη συνέχεια παράγονται τα ιδιωτικά και (αντίστοιχα) τα δημόσια κλειδιά και οι διευθύνσεις. Αυτή η λειτουργία διευκολύνει τη μεταφορά του πορτοφολιού σε μια νέα συσκευή.

Στην **Εικόνα 3.7** φαίνεται μια απεικόνιση της διαφοράς ανάμεσα σε αυτά τα δύο βασικά είδη πορτοφολιών.



Εικόνα 3.7 *Μια απεικόνιση ενός μη ντετερμινιστικού πορτοφολιού (αριστερά) και ενός ντετερμινιστικού με seed που οδηγεί στην αναπαραγωγή του κυρίως κλειδιού (δεξιά).*

Όσον αφορά την ευκολία μεταφοράς ενός ντετερμινιστικού πορτοφολιού από μια συσκευή σε μια άλλη, υπάρχουν και ορισμένες αρκετά φιλικές προς τον χρήστη τεχνικές που τη διευκολύνουν σημαντικά. Αυτές οι τεχνικές εστιάζουν στην απομνημόνευση και μεταφορά του αρχικού seed, από το οποίο μπορούν να ανακατασκευαστούν τα κλειδιά του πορτοφολιού.

Η τεχνική της απομνημόνευσης μιας σειράς από 12-24 λέξεις (γνωστή και ως BIP39[45]) παρουσιάστηκε ως *Πρόταση Βελτίωσης στο Bitcoin* (*Bitcoin Improvement Proposal, BIP*), αλλά γρήγορα υιοθετήθηκε και από άλλα δίκτυα blockchain. Σύμφωνα με την πρόταση αυτή, για την εύρεση του αρχικού seed, αντί να αντιγράφονται 256 bits που θα αντιστοιχούν στο αρχικό seed, καλείται ο χρήστης να αντιγράψει ένα σύνολο από λέξεις, τις οποίες πρέπει να αναπαράγει με την ίδια σειρά κατά τη διάρκεια της μεταφοράς του πορτοφολιού σε άλλη συσκευή (ή πρόγραμμα).

Ο τρόπος με τον οποίο δημιουργούνται οι λέξεις αυτές είναι με την επιλογή τους από ένα αλφάβητο που έχει δημιουργηθεί, με την εύρεση των κατάλληλων δεικτών για την κάθε λέξη. Για την επιλογή των δεικτών

[44] Στη βιβλιογραφία (Antonopoulos & Wood, 2019) η έλλειψη συσχέτισης στα κλειδιά έχει δημιουργήσει το όνομα JBOK (Just a Bunch Of Keys) για τα πορτοφόλια αυτά.

[45] Η περιγραφή της πρόταση βρίσκεται στον online σύνδεσμο https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki). Την πρόταση την έκανε η εταιρεία Ledger, γνωστή για την ανάπτυξη και δημιουργία πορτοφολιών hardware (cold πορτοφόλια).

αυτών, η εφαρμογή του πορτοφολιού μόλις της ζητηθεί να δημιουργήσει έναν νέο λογαριασμό-πορτοφόλι (άλλη επιλογή αποτελεί η μεταφορά ενός υπάρχοντος λογαριασμού-πορτοφολιού) παράγει έναν αριθμό από X bits (όπου X ανάμεσα σε 128-256 bits) από μια πηγή εντροπίας, π.χ. από το λειτουργικό σύστημα.

Κατόπιν, αφού περάσει την τιμή της πηγής ως είσοδο στη συνάρτηση κατακερματισμού SHA-256, κρατά Y bits (όπου Y ανάμεσα σε 4-24 bits) από το hash και τα προσθέτει ως bits ελέγχου στο τέλος της εισόδου. Έτσι, πλέον, ο αριθμός των bits είναι ίσος με X+Y.

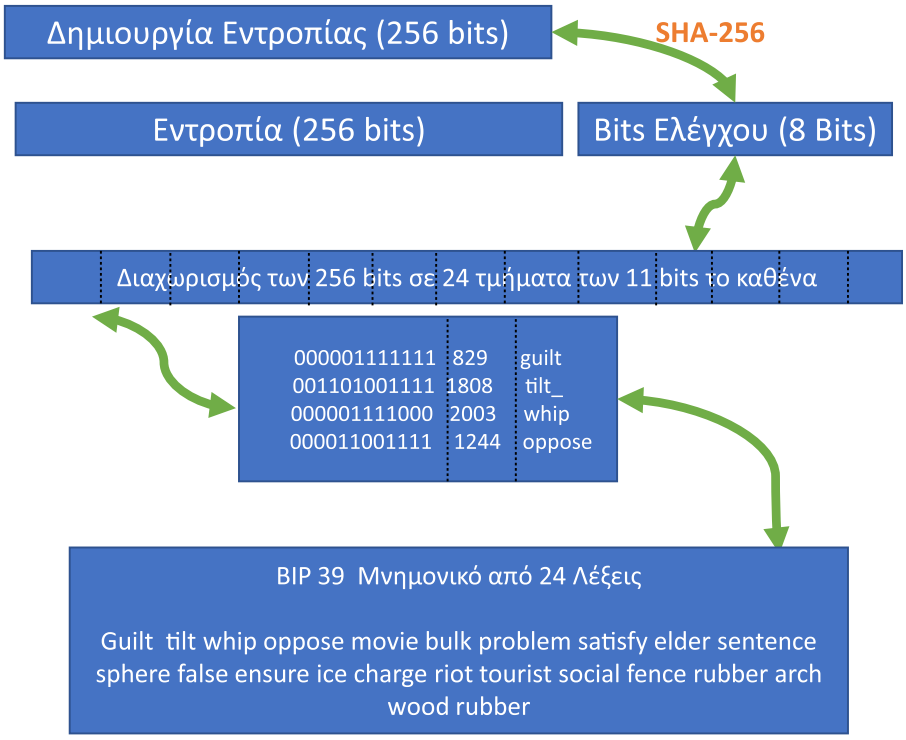
Κατόπιν, το άθροισμα αυτό χωρίζεται σε ομάδες των 11 bits και η καθεμία από αυτές είναι ο δείκτης σε ένα αλφάβητο από 2.048 λέξεις.

Στον **Πίνακα 3.7** φαίνονται οι συνδυασμοί που συναντώνται σχετικά με τον αριθμό των bits της εντροπίας και των bits ελέγχου, καθώς και με τον αριθμό των λέξεων απομνημόνευσης που παράγουν.

Εντροπία (Bits)	Bits Ελέγχου	Bits Ελέγχου + Εντροπίας	Αριθμός Λέξεων Απομνημόνευσης
128	4	132	12
160	5	165	15
192	6	198	18
224	7	231	21
256	8	264	24

Πίνακας 3.7 *Λέξεις απομνημόνευσης ανάλογα με τον αριθμό των bits ελέγχου και εντροπίας.*

Η διαδικασία της παραγωγής των λέξεων απομνημόνευσης απεικονίζεται και στην **Εικόνα 3.8** για καλύτερη κατανόηση.



Εικόνα 3.8 *Διαδικασία παραγωγής λέξεων απομνημόνευσης.*

Ο τρόπος αυτός είναι ιδιαίτερα διαδεδομένος σήμερα και χρησιμοποιείται σε πολλές από τις σύγχρονες εφαρμογές πορτοφολιών, τόσο στο δίκτυο του Bitcoin όσο και σε αυτό του Ethereum.

Στον **Πίνακα 3.8** φαίνεται ένα παράδειγμα με χρήση του online[46] εργαλείου παραγωγής λέξεων Mnemonic Code Converter, όπου φαίνονται οι τιμές της εντροπίας που δημιουργήθηκαν (στο δεκαεξαδικό), το αποτέλεσμα του αθροίσματος της εντροπίας με το hash τους (πάλι στο δεκαεξαδικό), με την προσθήκη των bits ελέγχου και, τέλος, οι 12 λέξεις απομνημόνευσης, που είναι το αποτέλεσμα της νέας (φιλτραρισμένης) εντροπίας.

[46] Online Σύνδεσμος: https://iancoleman.io/bip39/#english

Οι (12) αυτές λέξεις πρέπει να αποθηκευτούν σε ένα ασφαλές μέρος και να μην κοινοποιηθούν σε κανέναν.

Εντροπία (HEX)	Bits Ελέγχου (HEX)	Bits Ελέγχου + Εντροπίας	Αριθμός Λέξεων Απομνημόνευσης	Λέξεις
b272d09a9abfd3da90ffe80b1e53c2f0	0E	132	12	raven notable chase cube wood unhappy dumb wonder arch verb vague tide

Πίνακας 3.8 *Λέξεις απομνημόνευσης ανάλογα με τον αριθμό των bits ελέγχου και εντροπίας.*

Συνήθως οι συμβουλές περιέχουν οδηγίες για την αντιγραφή και την αποθήκευσή τους σε ένα ασφαλές μέρος (μέχρι και σε χρηματοκιβώτιο), ιδιαίτερα αν υπάρχουν αρκετά χρήματα (σε κρυπτονομίσματα) συνδεδεμένα με το πορτοφόλι αυτό, πάντα μέσω των κλειδιών του πορτοφολιού.

3.5.2 Κατηγορίες πορτοφολιών

Ανεξαρτήτως αν τα πορτοφόλια είναι ντετερμινιστικά ή όχι, υπάρχουν δύο βασικές κατηγορίες στις οποίες χωρίζονται: στα λεγόμενα *«ζεστά» πορτοφόλια* (*hot wallets*) και στα *«κρύα» πορτοφόλια* (*cold wallets*). Η καθεμία κατηγορία έχει τα δικά της χαρακτηριστικά και τα δικά της είδη.

Στη συνέχεια γίνεται μια παρουσίαση των πορτοφολιών που εντάσσονται σε καθεμία κατηγορία, συνοδευόμενη από μια ανάλυση των ομοιοτήτων και των διαφορών τους.

Ζεστά πορτοφόλια (*Hot wallets*): Το κύριο χαρακτηριστικό των πορτοφολιών που ανήκουν στην κατηγορία αυτή είναι ότι πρόκειται για πορτοφόλια που είναι συνεχώς συνδεδεμένα στο Διαδίκτυο. Αυτό σημαίνει ότι και τα κλειδιά τους είναι διαθέσιμα στο Διαδίκτυο, τόσο για την άμεση χρήση τους σε συναλλαγές όσο και για επιθέσεις από κακόβουλους χρήστες.

Τα πορτοφόλια αυτά ταιριάζουν πιο πολύ σε χρήστες οι οποίοι κάνουν συχνά συναλλαγές και επιζητούν ένα περιβάλλον διεπαφής που είναι εύκολο στη χρήση. Βεβαίως, ο κίνδυνος επιθέσεων είναι αρκετά αυξημένος λόγω της μόνιμης σύνδεσης στο Διαδίκτυο, μέσω υποκλοπής των κωδικών για τον λογαριασμό του πορτοφολιού. Για τον λόγο αυτό, συχνά, λαμβάνονται αυξημένα μέτρα ασφάλειας (π.χ. διπλή αυθεντικοποίηση).

Σημαντικό είναι ότι σε αρκετές από αυτές τις λύσεις τα πορτοφόλια που προσφέρονται είναι σε μορφή λογισμικού και δημιουργείται ο λογαριασμός χωρίς όμως να συνοδεύεται με την πρόσβαση στα κλειδιά. Κάτι που αποτελεί πολύ σημαντικό πρόβλημα ασφάλειας, καθώς η κατοχή των κλειδιών είναι το πιο σημαντικό στοιχείο για τη συμμετοχή σε ένα δίκτυο blockchain. Στην περίπτωση αυτή, η κατοχή των κλειδιών θυσιάζεται για τη δημιουργία ενός περιβάλλοντος που είναι φιλικό στον χρήστη και επιτρέπει άμεσες συναλλαγές.

Οι υποκατηγορίες που ανήκουν στα ζεστά πορτοφόλια είναι:

- *Τα πορτοφόλια για υπολογιστές* (*desktop wallets*): Αφορούν προγράμματα/εφαρμογές που κατεβαίνουν στον υπολογιστή ή στο laptop στα οποία δημιουργείται ένας λογαριασμός χρήστη. Κατόπιν, μέσω χρήσης λέξεων απομνημόνευσης, δημιουργείται το seed για το κύριο κλειδί, το οποίο προτείνεται στον χρήστη να αποθηκευτεί, για να μπορέσει να αποκτήσει πρόσβαση στο πορτοφόλι του και από άλλη συσκευή.
  Το σημαντικό στις λύσεις αυτές είναι ότι τα κλειδιά είναι στην κατοχή του χρήστη, ο οποίος έχει και την ευθύνη της διαφύλαξής τους και της χρήσης τους.

  Γνωστές λύσεις που ανήκουν στην κατηγορία αυτή είναι το Electrum και το Exodus.

- *Διαδικτυακά πορτοφόλια* (*Web wallets*): Πρόκειται για προγράμματα στα οποία αποκτά πρόσβαση ο χρήστης μέσω του Διαδικτύου, χωρίς να χρειάζεται να εγκαταστήσει κάποιο από αυτά. Απαιτείται μόνο η δημιουργία λογαριασμού για την πρόσβαση στο πορτοφόλι, οπότε υπάρχει ευκολία πρόσβασης από οποιαδήποτε συσκευή.
  Στα μειονεκτήματα είναι ότι συχνά ο χρήστης δεν είναι κάτοχος των κλειδιών του, οπότε στην ουσία δεν έχει τον πλήρη έλεγχο των κρυπτονομισμάτων του.

  Παραδείγματα τέτοιου είδους πορτοφολιών αποτελούν πορτοφόλια γνωστών ανταλλακτηρίων κρυπτονομισμάτων, όπως είναι της Coinbase και της Binance.

- *Πορτοφόλια για έξυπνα κινητά* (*Mobile wallets*): Λειτουργούν παρόμοια με τα desktop πορτοφόλια, καθώς απαιτείται η λήψη και η εγκατάσταση μιας εφαρμογής στο κινητό του χρήστη. Είναι αρκετά εύκολα στη χρήση και πολύ φιλικά στον χρήστη, είναι όμως και πολύ ευάλωτα σε επιθέσεις και απαιτείται ιδιαίτερη προσοχή στη χρήση τους.
  Γνωστά παραδείγματα αποτελούν το Mycelium και το Metamask[47].

Κρύα πορτοφόλια (*Cold wallets*): Στην κατηγορία αυτή ανήκουν τα πορτοφόλια εκείνα τα οποία δεν είναι συνεχώς διασυνδεδεμένα με το Διαδίκτυο, και έτσι θεωρούνται πιο ασφαλή. Τα πορτοφόλια αυτά περιέχουν τα κλειδιά του χρήστη και συχνά έχουν τη μορφή ενός USB με μια οθόνη ή ενός χαρτιού.

Προτιμώνται από χρήστες με σκοπό την ασφαλή αποθήκευση των κρυπτονομισμάτων τους και όχι τη συχνή δημιουργία συναλλαγών. Όταν χρειάζεται να πραγματοποιήσουν οι χρήστες συναλλαγή, τότε συνδέουν το cold πορτοφόλι τους με μια hot λύση για πορτοφόλι (η οποία είναι συμβατή στη συνεργασία με το cold πορτοφόλι που κατέχουν) για να πραγματοποιήσουν τη συναλλαγή τους.

Πιο αναλυτικά, η κατηγορία των cold πορτοφολιών περιλαμβάνει:

- *Πορτοφόλι σε χαρτί* (*Paper wallet*): Πρόκειται για ένα χαρτί το οποίο περιέχει το ζεύγος κλειδιών του χρήστη (υπάρχει συχνά και σε μορφή κώδικα QR). Αποτελούσαν διαδεδομένη μορφή για αρκετά χρόνια λόγω του ότι έπρεπε κάποιος να χάσει το χαρτί για να χάσει την πρόσβαση στα χρήματα, αλλά τελευταία, και με την ανάπτυξη των χαρακτηριστικών των σύγχρονων πορτοφολιών, δεν είναι πλέον τόσο δημοφιλή.
- *Πορτοφόλια υλικού* (*Hardware wallets*): Πρόκειται για την πιο διαδεδομένη μορφή κρύου πορτοφολιού. Μοιάζουν με USB και συνδέονται στην αντίστοιχη θύρα με τον υπολογιστή για να επικοινωνήσουν με το λογισμικό (hot wallet) για τη σύνδεση με το Διαδίκτυο και την εκτέλεση των συναλλαγών. Η χρήση τους είναι ιδιαίτερα εύκολη και τα επιπλέον χαρακτηριστικά ασφάλειας έχουν κάνει τέτοιου είδους λύσεις πολύ δημοφιλείς στους ιδιοκτήτες κρυπτονομισμάτων.
  Γνωστές εταιρείες που προμηθεύουν τέτοια πορτοφόλια είναι οι Ledger και Trezor.

Συμπερασματικά, να αναφερθεί ότι δεν υπάρχει μια λύση η οποία ταιριάζει σε κάθε περίπτωση. Αντιθέτως, η επιλογή του κατάλληλου τύπου πορτοφολιού βασίζεται στην κάθε χρήση ξεχωριστά. Συχνά, ένας συνδυασμός λύσεων είναι πιο εξυπηρετικός.

Συνοψίζοντας, στον **Πίνακα 3.9** παρουσιάζεται μια σύγκριση των χαρακτηριστικών των πορτοφολιών που ανήκουν στους δύο βασικούς τύπους που παρουσιάστηκαν σε αυτή την υποενότητα.

Κριτήρια	Ζεστά Πορτοφόλια	Κρύα Πορτοφόλια
Ορισμός	Εφαρμογές που συνδέονται στο Διαδίκτυο και αποθηκεύουν σε αυτό τα κλειδιά των χρηστών.	Πορτοφόλι που αποθηκεύει τα κλειδιά του χρήστη εκτός Διαδικτύου.
Τρόπος λειτουργίας	Δεδομένης της συνεχούς σύνδεσης στο Διαδίκτυο, οι συναλλαγές με κρυπτονομίσματα ολοκληρώνονται άμεσα και γρήγορα.	Συναλλαγές ξεκινούν στο Διαδίκτυο αλλά περνούν από το εκτός Διαδικτύου πορτοφόλι για να υπογραφούν, προτού ολοκληρωθούν διαδικτυακά.
Πιθανοί χρήστες	Επενδυτές κρυπτονομισμάτων και ιδιώτες που στοχεύουν σε γρήγορες συναλλαγές.	Χρήστες που θέλουν να αποθηκεύσουν με ασφάλεια τα κρυπτονομίσματά τους.
Τύποι	Υπάρχουν εφαρμογές για υπολογιστές ή κινητά καθώς και διαδικτυακές εφαρμογές (συχνά κρυπτο-ανταλλακτηρίων).	Πορτοφόλια σε χαρτί και πορτοφόλια υλικού είναι οι πιο γνωστοί τύποι κρύων πορτοφολιών.
Ασφάλεια	Υπάρχει κίνδυνος από τη συνεχή συνδεσιμότητα τους στο Διαδίκτυο.	Τα ιδιωτικά κλειδιά δεν φαίνονται ποτέ στο ζωντανό δίκτυο, προσφέροντας ένα σημαντικό επίπεδο ασφάλειας.

Πίνακας 3.9 *Χαρακτηριστικά ζεστών και κρύων πορτοφολιών.*

Κλείνοντας, μια από τις πιο γνωστές φράσεις στον κόσμο του blockchain λέει: «Not your keys, not your money». Η φράση αυτή υπογραμμίζει το επιστέγασμα όλων όσα πρέπει να θυμάται κανείς σχετικά με τα πορτοφόλια και τη χρήση τους. Ότι δηλαδή τα κλειδιά σας είναι η πόρτα σας για την είσοδο και την ταυτοποίησή σας στο δίκτυο αναφορικά με τις ενέργειες και την κατοχή σας σε αυτό.

[47] Γνωστό πορτοφόλι στο δίκτυο του Ethereum, το οποίο υπάρχει και σε μορφή επέκτασης στον browser του χρήστη.

Στην πραγματικότητα, η φράση αυτή καλεί τους χρήστες να μην επαναπαυτούν στη χρήση έτοιμων λύσεων που αντικαθιστούν την κλασική τραπεζική με μια άλλη εφαρμογή, αλλά να πάρουν τον έλεγχο των συναλλαγών και των κλειδιών τους, καθώς μόνο τότε μπορούν να νιώθουν ασφαλείς για την ιδιοκτησία τους στον κόσμο του blockchain.

Βιβλιογραφία

Antonopoulos, A. M. (2017). *Mastering Bitcoin. Programming the Open Blockchain* (2nd ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., & Wood, G. (2019). *Mastering Ethereum* (1st ed.). O'Reilly Media, Inc.

Hoffestein, J., Pipher, J., & Silverman, J. H. (2021). *Υπολογιστική Κρυπτογραφία: Μαθηματικές Δομές, Αλγόριθμοι, Ψηφιακές Υπογραφές, Πλέγματα και Κρυπτογραφία, Θεωρία Bitcoin* (2η έκδ.). Εκδόσεις ΦΟΥΝΤΑΣ, 2021, Κωδικός Ευδόξου: 94701365.

ΚΕΦΑΛΑΙΟ 4
Συναλλαγές

Σύνοψη

Το κεφάλαιο αυτό εστιάζει στις συναλλαγές που γίνονται σε ένα δίκτυο blockchain. Ιδιαίτερα, χρησιμοποιεί παραδείγματα από συναλλαγές στα δίκτυα Bitcoin και Ethereum για να εξηγήσει τη δομή και τη σύνταξή τους, εστιάζοντας στη βασική διαφορά που χαρακτηρίζει τον τρόπο που διαχειρίζονται τις συναλλαγές τα δύο αυτά μεγάλα δίκτυα. Επιπλέον, εξηγείται ο τρόπος με τον οποίο χρησιμοποιούνται (στις εφαρμογές πορτοφολιού) οι ψηφιακές υπογραφές του χρήστη σε κάθε συναλλαγή, για να επιβεβαιώσουν την ιδιοκτησία του ποσού που μεταφέρεται.

Τέλος, παρουσιάζεται ο τρόπος με τον οποίο τα δένδρα Merkle χρησιμοποιούνται για να διευκολύνουν την αναζήτηση μιας συναλλαγής σε ένα block, αλλά και πώς επιτρέπουν την ενσωμάτωση περισσότερων συναλλαγών σε ένα block.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 2 και 3.

4.1 Μορφές συναλλαγών σε ένα δίκτυο blockchain

Γενικά, οι συναλλαγές σε ένα δίκτυο blockchain είναι το πιο σημαντικό κομμάτι της λειτουργίας του. Μάλιστα, μπορεί να πει κανείς ότι όλα τα χαρακτηριστικά της τεχνολογίας έχουν διαμορφωθεί γύρω από την εκτέλεση, επεξεργασία, αποθήκευση, εύρεση και την ασφάλεια των συναλλαγών που λαμβάνουν χώρα.

Οι συναλλαγές σε ένα δίκτυο blockchain αποτελούνται από δεδομένα τα οποία περιέχουν τη μεταβίβαση ποσών από μία διεύθυνση σε μια άλλη ή, ακόμα, και κώδικα, στην περίπτωση που η διεύθυνση προορισμού αντιστοιχεί, για παράδειγμα, σε ένα smart contract (περίπτωση του Ethereum). Επιπλέον, εκτός από τα δεδομένα, οι συναλλαγές πρέπει να περιέχουν και την υπογραφή του δημιουργού της συναλλαγής η οποία, όπως αναφέρθηκε και στο Κεφάλαιο 3, παίζει τον ρόλο της απόδειξης ιδιοκτησίας των μεταφερόμενων ποσών ή δεδομένων.

Ανάλογα με τον τρόπο επεξεργασίας των δεδομένων των συναλλαγών, υπάρχουν δύο βασικές προσεγγίσεις από τα γνωστά δίκτυα blockchain. Έτσι, υπάρχουν τα δίκτυα που χρησιμοποιούν τις εξόδους των συναλλαγών που έχουν *ποσά που δεν έχουν ξοδευτεί* (*Unspent Transaction Output, UTXOs*) και υπάρχουν και άλλα τα οποία χρησιμοποιούν *λογαριασμούς* (*Account-based*) το υπόλοιπο των οποίων ενημερώνεται μετά από κάθε συναλλαγή που τους επηρεάζει. Γνωστό παράδειγμα των πρώτων δικτύων αποτελεί το Bitcoin, ενώ γνωστό παράδειγμα των δεύτερων αποτελεί το Ethereum.

Στη συνέχεια, για να γίνει κατανοητό πώς λειτουργούν και πώς συντάσσονται οι συναλλαγές, ακολουθεί αναλυτική περιγραφή των δύο αυτών μεθόδων δημιουργίας και διεκπεραίωσης των συναλλαγών στα δίκτυα blockchain.

4.1.1 Δίκτυα blockchain που βασίζονται σε UTXOs

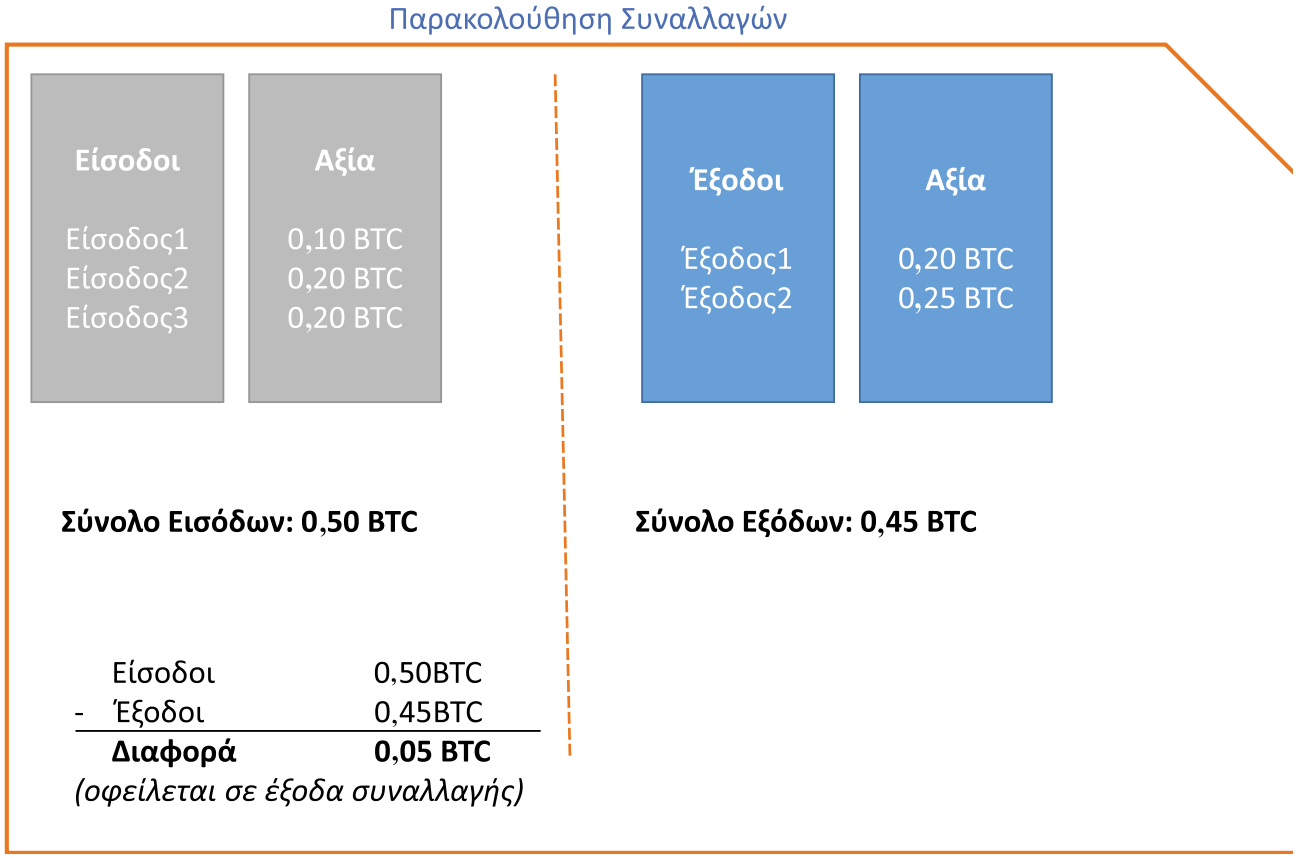
Σε ένα δίκτυο blockchain που λειτουργεί με UTXOs δεν υπάρχουν λογαριασμοί χρηστών και πορτοφόλια στο επίπεδο πρωτοκόλλου. Αντιθέτως, τα ποσά που μεταφέρονται αντιστοιχούν σε μια σειρά από εξόδους συναλλαγών, γνωστές με το όνομα UTXOs, οι οποίες παραμένουν στο πορτοφόλι του ιδιοκτήτη αναμένοντας να χρησιμοποιηθούν.

Στα δίκτυα αυτά κάθε συναλλαγή έχει έναν αριθμό από εισόδους και παράγει έναν αριθμό από εξόδους. Οι έξοδοι αυτές είναι ποσότητες *αδιαίρετες*, οι οποίες εφόσον δημιουργηθούν τότε πρέπει να χρησιμοποιηθούν όπως είναι. Ένα πορτοφόλι χρησιμοποιεί τις εξόδους αυτές με το να τις συμπεριλάβει ως εισόδους σε επόμενες, μελλοντικές, συναλλαγές.

Η **Εικόνα 4.1** δείχνει τη βασική μορφή μιας συναλλαγής σε ένα δίκτυο UTXO. Η συναλλαγή αυτή αποτελείται από ένα σύνολο εισόδων (μία ή περισσότερες) οι οποίες παράγουν ένα νέο σύνολο από εξόδους (επίσης μπορεί

να είναι μία ή περισσότερες). Τόσο οι είσοδοι όσο και οι έξοδοι της συναλλαγής αποτελούνται από UTXOs, τα οποία χρειάζονται το κλειδί του χρήστη για να αποδειχθεί η ιδιοκτησία τους και να συμπεριληφθούν στη συναλλαγή ως είσοδοι. Μάλιστα, εξαιτίας της μορφής που έχουν οι συναλλαγές στα δίκτυα blockchain που βασίζονται σε UTXOs, η χρήση αυτών επιτρέπει την αποθήκευση των χρημάτων που συναλλάσσονται στο δίκτυο ως μια λίστα από UTXOs.

Στην **Εικόνα 4.1** φαίνεται ένα παράδειγμα με 2 συναλλαγές ως είσοδοι και 2 ως έξοδοι. Επίσης, φαίνεται ότι η διαφορά του αθροίσματος των UTXOs στην είσοδο συγκρινόμενη με το άθροισμα των UTXOs στην έξοδο δεν είναι ίση με μηδέν, αλλά είναι μεγαλύτερη. Αυτό συμβαίνει γιατί κάθε συναλλαγή για να τύχει επεξεργασίας και να μπει σε ένα υποψήφιο block (αρχικά) θα πρέπει να περιλαμβάνει και ένα ποσό ως «αμοιβή» για τον κόμβο miner που θα αναλάβει να εκτελέσει τις ενέργειες αυτές.



Εικόνα 4.1 *Παράδειγμα εισόδων και εξόδων σε μια συναλλαγή σε ένα δίκτυο blockchain που υποστηρίζει UTXOs.*

Πρακτικά, η χρήση των UTXOs προσομοιάζεται πολύ πετυχημένα με τη χρήση των μετρητών στην πραγματική ζωή. Αν κάποιος θέλει να αγοράσει ένα προϊόν που κοστίζει 5,5 ευρώ, θα αναζητήσει έναν συνδυασμό από νομίσματα που αθροίζουν στην ίδια αξία για να πληρώσει. Εάν δεν βρει ακριβώς το συγκεκριμένο ποσό, τότε θα ψάξει να βρει έναν συνδυασμό από νομίσματα που θα έχουν χρηματικό ισοδύναμο μεγαλύτερο της ζητούμενης αξίας. Έστω ότι βρίσκει ένα χαρτονόμισμα αξίας 10 ευρώ. Θα το χρησιμοποιήσει στη συναλλαγή του και θα λάβει στο τέλος και ρέστα αξίας κάτι λιγότερο από 4,5 ευρώ (συνυπολογιζόμενα και τα έξοδα επεξεργασίας).

Όπως υπάρχουν νομίσματα με συγκεκριμένη αξία στην πραγματική ζωή, έτσι και στο blockchain υπάρχουν UTXOs με συγκεκριμένη τιμή (**Εικόνα 4.1**). Ο τρόπος με τον οποίο ένα πορτοφόλι που δημιουργεί μια συναλλαγή χρησιμοποιεί τα UTXOs είναι εντελώς αντίστοιχος με το προηγούμενο παράδειγμα. Έτσι, αν κάποιος θέλει, για παράδειγμα, να ξοδέψει 1,85 BTC για να αγοράσει ένα προϊόν από το Διαδίκτυο, τότε το πορτοφόλι του χρήστη θα αναζητήσει όλα τα UTXOs τα οποία γνωρίζει ότι χρειάζονται ένα από τα κλειδιά που έχει δημιουργήσει το πορτοφόλι αυτό για τον χρήστη, έτσι ώστε να χρησιμοποιηθούν στη συναλλαγή. Εάν βρει ένα UTXO το οποίο θα αντιστοιχεί σε 1,85 BTC, τότε το χρησιμοποιεί ως είσοδο στη συναλλαγή[48], προσθέτει

[48] Στην πραγματικότητα, όπως φαίνεται και στην Εικόνα 4.1, το ποσό πρέπει να είναι ελαφρώς μεγαλύτερο για να μπορέσει να καλύψει και τα έξοδα επεξεργασίας του δικτύου.

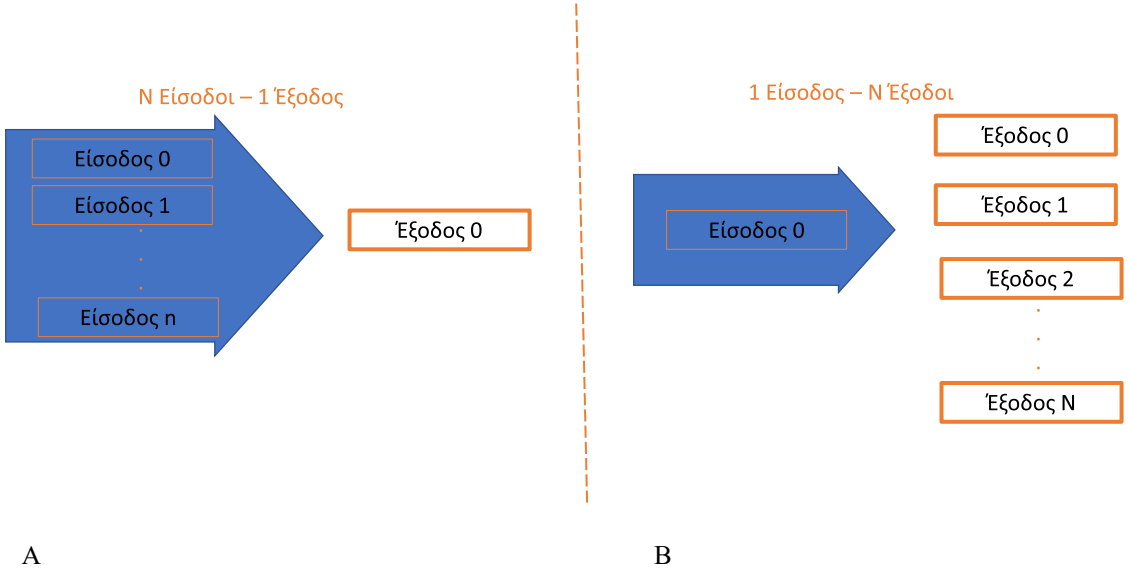
τη διεύθυνση του πωλητή και παράγεται μια έξοδος στην οποία απεικονίζεται η μεταφορά των χρημάτων στη διεύθυνση του πωλητή.

Εφόσον δεν βρεθεί τέτοιο UTXO, τότε το πορτοφόλι θα αναζητήσει τον συνδυασμό εκείνο που ικανοποιεί την απαίτηση για τη συναλλαγή. Δηλαδή ψάχνει να βρει ένα άθροισμα από UTXOs το οποίο θα είναι ελαφρώς μεγαλύτερο από το ζητούμενο ποσό των 1,85 BTC.[49] Χρησιμοποιώντας τον συνδυασμό αυτόν από UTXOs στην είσοδο, τότε ως έξοδο στη συναλλαγή αυτή (εκτός από την αποστολή των 1,85 BTC στη διεύθυνση του πωλητή) θα δημιουργείται και μια δεύτερη έξοδος, η οποία θα περιέχει τα ρέστα του χρήστη, μετά και την αφαίρεση του ποσού της αμοιβής των κόμβων του δικτύου.

Στη συνέχεια, αυτή είναι η έξοδος (UTXO) που θα αναγνωρίζεται από το πορτοφόλι του χρήστη και που μπορεί να χρησιμοποιηθεί μελλοντικά ως είσοδος σε μια άλλη συναλλαγή.

Συνολικά, για να μπορέσει ένα πορτοφόλι να χρησιμοποιήσει με τον πιο αποδοτικό τρόπο τα UTXOs, θα πρέπει να βρει τον βέλτιστο συνδυασμό εισόδων και εξόδων. Ο συνδυασμός αυτός, εκτός από την περίπτωση όπου έχουμε 1 είσοδο και 1 έξοδο, όπως αναφέρθηκε και στο παράδειγμα προηγουμένως, μπορεί να περιέχει και τις περιπτώσεις που φαίνονται στην **Εικόνα 4.2**. Δηλαδή να υπάρχουν πολλές είσοδοι οι οποίες πρέπει να συνδυαστούν για να προκύψει 1 έξοδος (**Εικόνα 4.2Α**). Μπορεί όμως να ισχύει και το αντίστροφο. Δηλαδή να υπάρχει 1 είσοδος η οποία δημιουργεί πολλές (περισσότερες από 1) εξόδους. Αυτό γίνεται αν ο χρήστης θέλει να στείλει σε διαφορετικούς παραλήπτες ένα ποσό.

Παράδειγμα μιας τέτοιας συναλλαγής είναι η περίπτωση της μισθοδοσίας, όπου ο υπεύθυνος μιας εταιρείας δημιουργεί μια συναλλαγή για να δώσει τους μισθούς των υπαλλήλων. Στην περίπτωση της **Εικόνας 4.2Β** θα πρέπει η λειτουργία αυτή να υποστηρίζεται και από το πορτοφόλι του χρήστη.



Εικόνα 4.2 *Παραδείγματα μορφών συναλλαγών με χρήση UTXOs: Α. Χρήση πολλών εισόδων και μία έξοδος, Β. Χρήση μίας εισόδου με πολλές εξόδους.*

Επιπλέον, πάλι εφόσον υποστηρίζεται από το πορτοφόλι του χρήστη, είναι δυνατόν μια συναλλαγή να έχει στην είσοδο ένα UTXO, το οποίο όμως να απαιτεί περισσότερες από μία υπογραφές για να μπορέσει να δώσει την άδεια μεταφοράς των χρημάτων. Η περίπτωση αυτή ονομάζεται *multisig* και είναι αρκετά συχνή, καθώς επιτρέπει τον διαμοιρασμό της ευθύνης για τη χρήση των χρημάτων μεταξύ περισσότερων ατόμων. Επίσης, αυξάνει και την προστασία του λογαριασμού, καθώς θα πρέπει ένας επιτιθέμενος να αποκτήσει περισσότερα κλειδιά για να ξεκλειδώσει τα χρήματα που διαχειρίζεται το πορτοφόλι.

Κλείνοντας την παρουσίαση των UTXOs, να αναφερθεί ότι, ενώ το παράδειγμα της χρήσης των πραγματικών χρημάτων προσομοιώνει με επιτυχία τη χρήση των UTXOs, υπάρχουν και διαφορές. Έτσι, κατά τη συναλλαγή με φυσικά χρήματα δεν υπάρχουν έξοδα επεξεργασίας, όπως κατά την εκτέλεση συναλλαγών με UTXOs. Επίσης, τα χρήματα είναι συγκεκριμένα (νομίσματα και χαρτονομίσματα συγκεκριμένης τιμής), ενώ τα UTXOs

[49] Υπενθυμίζεται ότι το ποσό πάλι πρέπει να είναι ελαφρώς μεγαλύτερο, λόγω των εξόδων επεξεργασίας από τους κόμβους στο δίκτυο.

που μπορούν να δημιουργηθούν εξαρτώνται από τις ανάγκες εξυπηρέτησης των συναλλαγών και μόνο και όχι από διακριτές, σταθερές τιμές.

4.1.2 Δίκτυα blockchain που βασίζονται σε λογαριασμούς (accounts)

Σε αντίθεση με τα δίκτυα blockchain που διαχειρίζονται UTXOs, υπάρχουν και άλλα δίκτυα τα οποία διαχειρίζονται τα νομίσματα όχι ως εξόδους συναλλαγών, αλλά ως ποσότητες που βρίσκονται μέσα σε έναν λογαριασμό. Στην ουσία πρόκειται για μια αντιμετώπιση η οποία θυμίζει τη χρήση των τραπεζικών λογαριασμών.

Έτσι, για παράδειγμα, αν ένας λογαριασμός έχει μέσα 20 ethers και θέλει να δημιουργήσει μια συναλλαγή στην οποία αποστέλλει 12,25 ethers, τότε θα το κάνει απευθείας και θα παραμείνουν 7,75 ethers στον λογαριασμό του αποστολέα, ενώ ο λογαριασμός του παραλήπτη θα έχει 12,25 ethers (θεωρούμε για ευκολία ότι δεν είχε άλλα χρήματα από πριν μέσα). Με τον τρόπο αυτό δεν χρειάζεται να γίνει μια συναλλαγή με είσοδο 20 ethers η οποία θα αφήσει σε μία έξοδο τα 12,25 ethers και θα επιστρέψει στον αρχικό ιδιοκτήτη, σε μια δεύτερη έξοδο, τα 7,75 ethers, όπως θα γινόταν στην περίπτωση των UTXOs.

Με τη χρήση ενημέρωσης υπολοίπων σε λογαριασμούς χρηστών μια συναλλαγή δεν δηλώνει (προκαθορίζει) την τελική κατάσταση και εξαρτάται από την αρχική της συνθήκη (δηλαδή από τα ποσά στους λογαριασμούς των συμμετεχόντων).

Για τον λόγο αυτόν ακριβώς χρειάζεται να ληφθεί ιδιαίτερη φροντίδα κατά την εκτέλεση πολλών συναλλαγών από έναν συγκεκριμένο λογαριασμό, έτσι ώστε να πραγματοποιηθούν με τη σειρά εκείνη που θα επιτρέψει την επιτυχή ολοκλήρωσή τους.

4.2 Δομή συναλλαγών

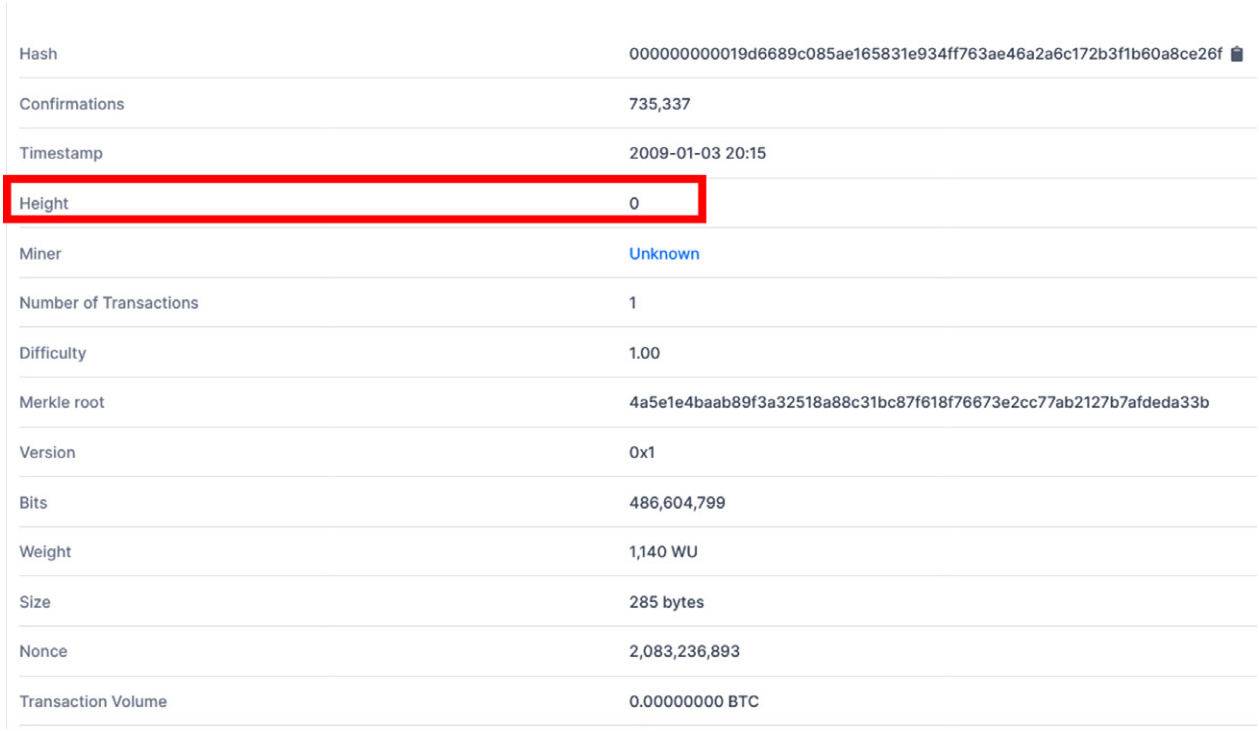
Έχοντας γνωρίσει τη βασική διαφορά που υπάρχει στη μορφή των συναλλαγών στα δύο μεγάλα δημόσια δίκτυα blockchain, στο Bitcoin και στο Ethereum, στη συνέχεια θα γίνει μια πιο λεπτομερής ανάλυση και παρουσίαση της δομής των συναλλαγών στα δύο δίκτυα.

4.2.1 Δομή συναλλαγής στο δίκτυο Bitcoin

Ο κύριος σκοπός μιας συναλλαγής στο δίκτυο του Bitcoin είναι η μεταφορά BTC μεταξύ χρηστών/διευθύνσεων. Οι συναλλαγές στο δίκτυο Bitcoin δεν είναι κρυπτογραφημένες, επομένως ο καθένας μπορεί να αναζητήσει και να δει όλες τις συναλλαγές και όλα τα blocks που έχουν δημιουργηθεί στο δίκτυο.[50]

Στην **Εικόνα 4.3** φαίνεται η επικεφαλίδα του πρώτου block στο δίκτυο του Bitcoin, με το νούμερο 0, που ονομάζεται και *block γέννησης* (*genesis block*), το οποίο δημιουργήθηκε στις 3 Ιανουαρίου του 2009.

[50] Τα προγράμματα αυτά στην ουσία συνδέονται σε έναν πλήρη κόμβο και οπτικοποιούν την πληροφορία που διαβάζουν από το ledger. Ενδεικτικά, ένας τέτοιος explorer βρίσκεται στον σύνδεσμο: https://www.blockchain.com/explorer?view=btc



Hash	000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Confirmations	735,337
Timestamp	2009-01-03 20:15
Height	0
Miner	Unknown
Number of Transactions	1
Difficulty	1.00
Merkle root	4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
Version	0x1
Bits	486,604,799
Weight	1,140 WU
Size	285 bytes
Nonce	2,083,236,893
Transaction Volume	0.00000000 BTC

Εικόνα 4.3 *Η επικεφαλίδα του block γέννησης στο δίκτυο του Bitcoin.*

Ο τρόπος που πραγματοποιείται η μεταφορά των BTC συμπεριλαμβάνει την εύρεση και συμμετοχή UTXOs ως εισόδους, καθώς και τον καθορισμό των UTXOs που θα παραχθούν στην έξοδο. Τόσο οι είσοδοι όσο και οι έξοδοι είναι δυνατόν να έχουν συνδυασμούς από ένα ή περισσότερα UTXOs.

Στην **Εικόνα 4.4** φαίνεται η συναλλαγή που περιέχεται στο block γέννησης του Bitcoin (Εικόνα 4.3). Σε αυτή φαίνεται η μεταφορά 50 BTC στον λογαριασμό με διεύθυνση *1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa*,[51] ο οποίος και ανήκει στον δημιουργό του δικτύου του Bitcoin, τον Satoshi Nakamoto.

Επιπλέον, φαίνεται και το hash της συναλλαγής (*4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b*), το οποίο και αποτελεί το χαρακτηριστικό με το οποίο θα αναφερθεί μια νέα, μελλοντική, συναλλαγή σε αυτό το UTXO, όταν θελήσει να τη χρησιμοποιήσει ως είσοδο.

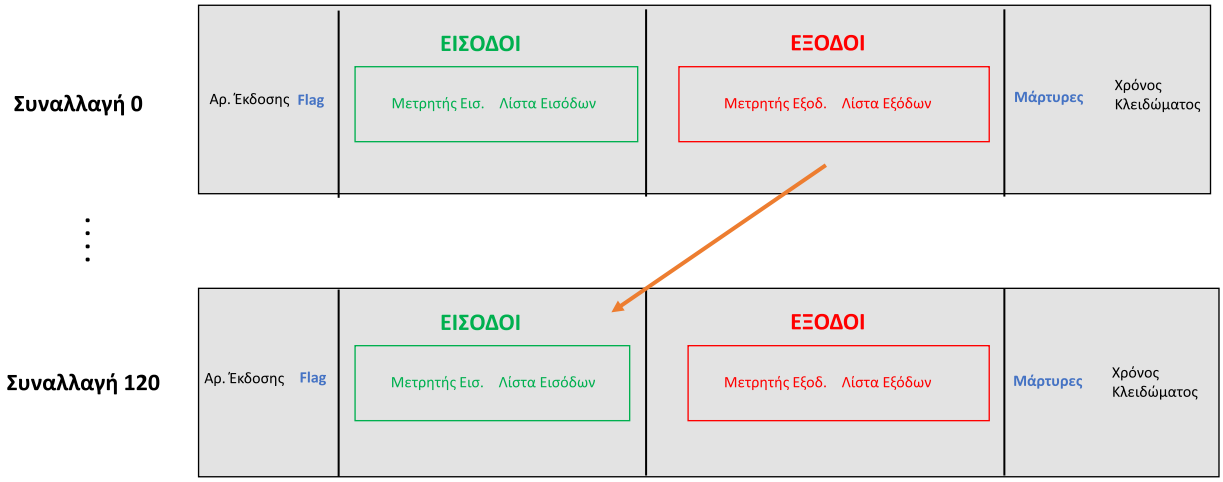
Η συγκεκριμένη διεύθυνση πάντως δεν έχει προβεί σε καμία μεταφορά bitcoins. Αντιθέτως, έχει λάβει 68,53544671 BTC, όπως μπορεί να φανεί με μια αναζήτησή της στην προαναφερθείσα εφαρμογή.

Εικόνα 4.4 *Η συναλλαγή που περιέχεται στο block γέννησης στο δίκτυο του Bitcoin.*

Στην **Εικόνα 4.5** φαίνονται τα μέρη μιας συναλλαγής και με λεπτομέρεια ο τρόπος που χρησιμοποιούνται τα UTXOs σε αυτήν. Όπως αναφέρθηκε, η κάθε συναλλαγή θα έχει τουλάχιστον 1 είσοδο και 1 έξοδο, η οποία και αναμένει στο πορτοφόλι του χρήστη να χρησιμοποιηθεί ως είσοδος σε μελλοντική συναλλαγή. Στην Εικόνα απεικονίζεται και αυτό το χαρακτηριστικό με τη χρήση μιας εξόδου UTXO της Συναλλαγής 0 ως είσοδο στη Συναλλαγή 120.

Για διευκόλυνση μπορεί να θεωρηθεί ότι τα κουτιά που συμβολίζουν τις εισόδους και τις εξόδους περιέχουν 1 είσοδο και 1 έξοδο και ότι οι συναλλαγές 0 και 120 βρίσκονται σε διαφορετικά blocks.

[51] Η συναλλαγή αυτή αφορά δημιουργία καινούργιων νομισμάτων από το δίκτυο (COINBASE) και για αυτό δεν έχει διεύθυνση αποστολέα.



Εικόνα 4.5 *Τα μέρη μιας συναλλαγής στο δίκτυο του Bitcoin. Θεωρείται ότι υπάρχει 1 είσοδος και 1 έξοδος σε κάθε συναλλαγή και πως οι δύο συναλλαγές που απεικονίζονται βρίσκονται σε διαφορετικά blocks.*

Στον **Πίνακα 4.1** αναλύεται η δομή του κάθε μέρους μιας συναλλαγής, όπως αυτά φαίνονται στην Εικόνα 4.5.

Πεδίο	Περιγραφή	Μέγεθος (bytes)	Αναστροφή στη σειρά των bytes
Αριθμός Έκδοσης (Version No)	Τιμή 1 ή 2 (πιο πρόσφατη)	4	Ναι
Σημαία (Flag)	Έχει πάντα τιμή 0001 (προαιρετικό) Δηλώνει την ύπαρξη του πεδίου witness	2 (προαιρετικό)	Όχι
Μετρητής Εισόδων (In-counter)	Θετικός Ακέραιος Αριθμός (Varint = VI) Δηλώνει τον αριθμό των εισόδων στη συναλλαγή	1-9	Όχι
Λίστα Εισόδων (list of Inputs)	Δομή που περιλαμβάνει τις εισόδους (βλ. Πίνακα 4.2)	Μεταβλητός αριθμός ανάλογα και με την τιμή του Μετρ. Εισόδου	–
Μετρητής Εξόδων (Out-counter)	Θετικός Ακέραιος Αριθμός (Varint = VI) Δηλώνει τον αριθμό των εξόδων στη συναλλαγή	1-9	Όχι
Λίστα Εξόδων (List of Outputs)	Δομή που περιλαμβάνει τις εξόδους (βλ. Πίνακα 4.3)	Μεταβλητός αριθμός ανάλογα και με την τιμή του Μετρ. Εξόδου	–
Μάρτυρες (Witnesses)	Μια λίστα από Μάρτυρες (Witnesses)	Μεταβλητό	–
Χρόνος Κλειδώματος (nLockTime)	Πρόκειται για τον μικρότερο χρόνο στον οποίο μπορεί να μπει μια συναλλαγή στο blockchain	4	Ναι

Πίνακας 4.1 *Παρουσίαση των μερών μιας συναλλαγής στο δίκτυο του Bitcoin.*

Ο *Αριθμός Έκδοσης* αποτελείται από 4 bytes και ενημερώνει τους κόμβους και τους miners για την έκδοση των κανόνων που πρέπει να χρησιμοποιηθούν για την επαλήθευση της συναλλαγής. Με τον τρόπο αυτόν επιτρέπεται η ενημέρωση των κανόνων, χωρίς αυτή να επηρεάζει παλαιότερες εκδόσεις, που συνεχίζουν και λειτουργούν κανονικά.

Το πεδίο που δηλώνει το *Flag*, όταν υπάρχει, υποδηλώνει ότι η συναλλαγή ακολουθεί τη δομή της σημαντικής αναβάθμισης του πρωτοκόλλου του Bitcoin, με το όνομα *Segregated Witness* (*SegWit*). Η αναβάθμιση αυτή έχει σκοπό να ενισχύσει την προστασία των συναλλαγών από κακόβουλη χρήση και να αυξήσει τη χωρητικότητα του block. Πρόκειται για την αναβάθμιση που έπαιξε σημαντικό ρόλο στην ενσωμάτωση του δικτύου Lightning (Antonopoulos et al., 2021) στο Bitcoin. Η τιμή του flag είναι 0001 και ενημερώνει για την ύπαρξη του πεδίου Witness στη δομή της συναλλαγής.

Το πεδίο *Witnesses* έχει επιπλέον δεδομένα, τα οποία μπορούν να χρησιμοποιηθούν για να επιβεβαιωθεί η εγκυρότητα της συναλλαγής, χωρίς όμως να είναι αναγκαία για την ολοκλήρωσή της. Ο αριθμός των witnesses αντιστοιχεί στον αριθμό των εισόδων. Άρα, για μία είσοδο θα υπάρχει μόνο μία τιμή στο πεδίο. Η τιμή του πεδίου Flag, όμως, είναι ανεξάρτητη από τον αριθμό των witnesses. Την αφορά μόνο η παρουσία (ή όχι) ακόμα και ενός witness.

Ο *Χρόνος Κλειδώματος* (*nLockTime*) συμβολίζει το μικρότερο ύψος στην αλυσίδα του ledger στο οποίο μπορεί να συμπεριληφθεί η συναλλαγή. Θα πρέπει να φθάσει το ύψος αυτό για να μπορέσει να συμμετέχει

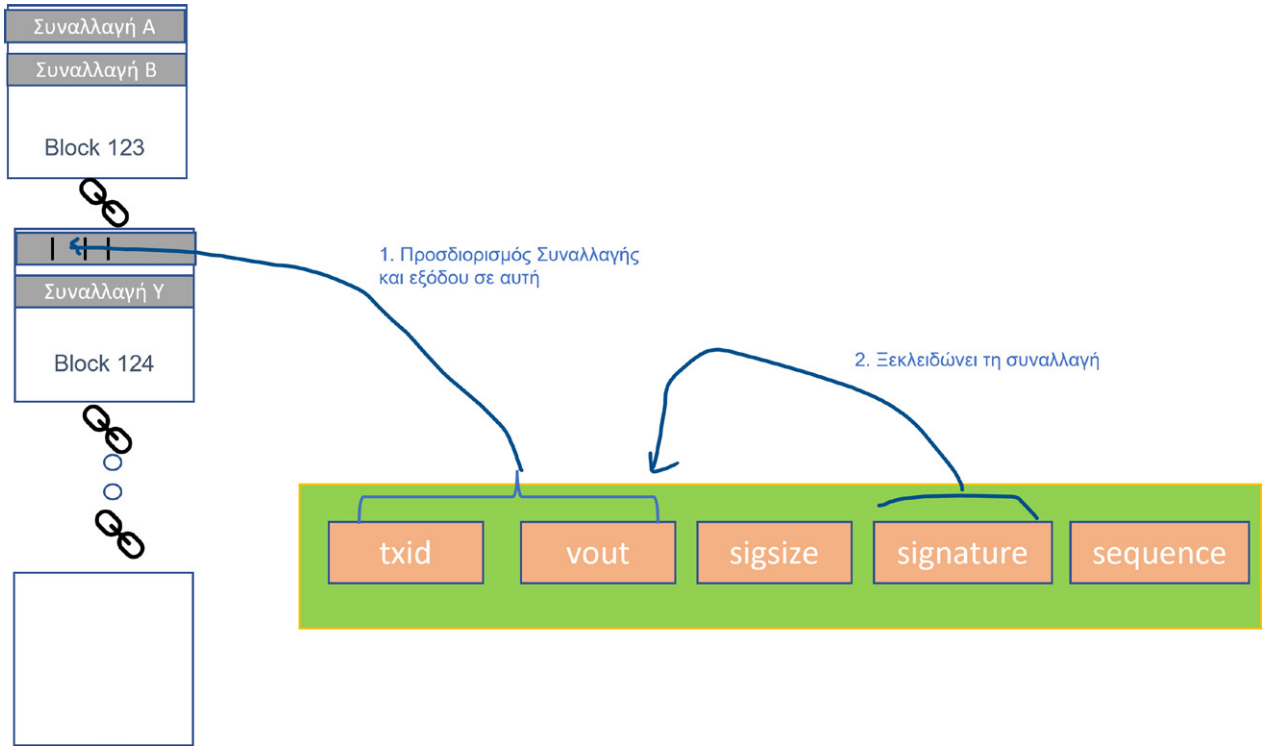
σε ένα υποψήφιο block. Πιο συγκεκριμένα, εάν έχει τιμή <500 εκατομμύρια ($5*10^8$), ερμηνεύεται ως ύψος στο ledger και οι miners πρέπει να αναμένουν να φθάσουν σε αυτό. Εάν έχει τιμή >500 εκατομμύρια ($5*10^8$), μετατρέπεται σε χρονική σφραγίδα unix (δηλαδή δείχνει τον αριθμό των δευτερολέπτων από την 1/1/1970).

Η τελευταία στήλη στον Πίνακα 4.1 δείχνει εάν η απόδοση της τιμής για το πεδίο εκφράζεται με αντιστροφή της σειράς των bytes κατά τη μετατροπή της συναλλαγής σε bytes για την αποστολή της στο δίκτυο. Για παράδειγμα, η έκδοση 1 με χρήση 4 bytes πρέπει να αποδοθεί (στο δεκαεξαδικό) ως 00000001 Η. Όμως, λόγω της αντιστροφής, αποδίδεται ως 01000000 Η. Το σημείο αυτό θα τονιστεί και στη συνέχεια, στο παράδειγμα με την αναλυτική εξήγηση της δομής μιας πραγματικής συναλλαγής του δικτύου Bitcoin.

Οι *Μετρητές Εισόδων* και *Εξόδων* (*In-Counter, Out-Counter*) δηλώνουν τον αριθμό των εισόδων και εξόδων που συμμετέχουν στη συναλλαγή. Αποτελεί πληροφορία που θα χρειαστούν και στη συνέχεια οι λίστες εισόδων και εξόδων.

Η *Λίστα Εισόδου* περιέχει τις απαραίτητες πληροφορίες (περιγραφή τους και ξεκλείδωμά τους) που αφορούν τα UTXOs που συμμετέχουν ως είσοδοι στη συναλλαγή.

Μια απεικόνιση των περιεχομένων της φαίνεται στην **Εικόνα 4.6**. Υπενθυμίζεται στο σημείο αυτό ότι κάθε UTXO συμμετέχει με το ακέραιο ποσό των χρημάτων που του έχουν αποδοθεί. Σε μια συναλλαγή μπορούν να συμμετέχουν από 1 έως 2^{32} είσοδοι. Το άθροισμα των ποσών των UTXOs στην είσοδο υπολογίζεται και αποδίδεται στις εξόδους, μειούμενο κατά ένα τμήμα λόγω των εξόδων επεξεργασίας.



Εικόνα 4.6 *Μια πιο αναλυτική παρουσίαση των περιεχομένων στη Λίστα Εισόδου.*

Μια πιο λεπτομερής ανάλυση των περιεχομένων της Λίστας Εισόδου φαίνεται στον **Πίνακα 4.2**.

Πεδίο	Περιγραφή	Μέγεθος (bytes)	Αναστροφή στη σειρά των bytes
txid (Hash UTXO)	Αναφορά σε προηγούμενη συναλλαγή	32	Ναι
Index (UTXO)	Δείκτης του αριθμού της εξόδου στο προηγούμενο UTXO (Θετικός Ακέραιος Αριθμός)	4	Ναι
Μήκος ScriptSig	Μεταβλητή που δηλώνει πόσα bytes είναι το μήκος του ScriptSig (Θετικός Ακέραιος Αριθμός Varint = VI)	1-9	Όχι
ScriptSig	Ένα Script που ξεκλειδώνει το UTXO	Μεταβλητό	Όχι
Αριθμός Ακολουθίας (Sequence)	Πλέον έχει τιμή 0xFFFFFFFF (απενεργοποιημένο)	4	Ναι

Πίνακας 4.2 *Ανάλυση των περιεχομένων της Λίστας Εισόδου.*

Το πεδίο *txid* προσδιορίζει το hash της συναλλαγής σε μια από τις εξόδους όπου βρίσκεται το UTXO που θα χρησιμοποιηθεί ως είσοδος. Σε συνδυασμό με το πεδίο *index*, το οποίο υποδηλώνει τον αριθμό της εξόδου στη συναλλαγή, προσδιορίζουν με μοναδικό τρόπο το UTXO το οποίο μπαίνει στη Λίστα Εισόδου. Το πεδίο index ξεκινά από την τιμή 0 για να υποδείξει την 1η έξοδο στην εν λόγω συναλλαγή. Στην Εικόνα 4.6 το βήμα 1 απεικονίζει αυτή την αντιστοίχιση που επιτυγχάνεται από τα δύο αυτά πεδία.

Επιπλέον, υπάρχει το πεδίο *Μήκος ScriptSig* που δηλώνεται το μήκος του ScriptSig που ακολουθεί.

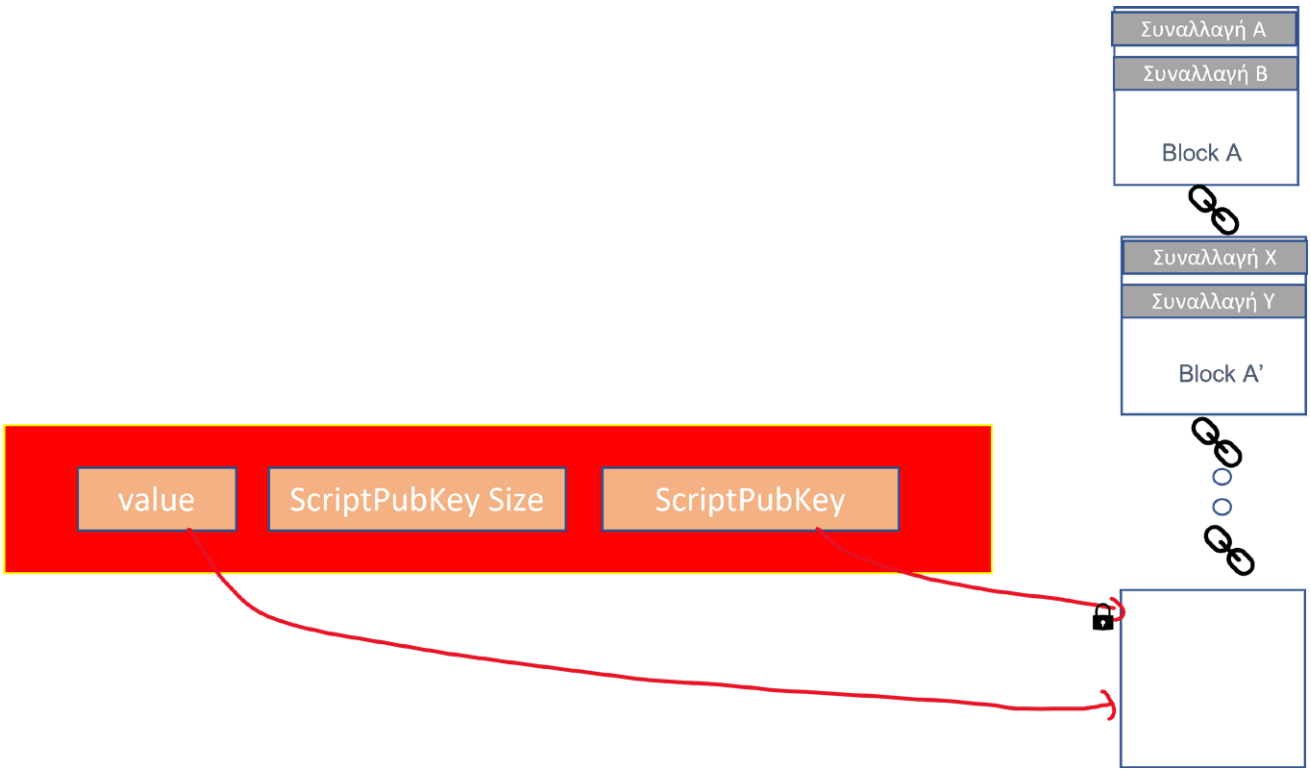
Κατόπιν, το πεδίο *ScriptSig* περιέχει ό,τι χρειάζεται για να ξεκλειδώσει το UTXO και να επιτρέψει την αποστολή των χρημάτων. Είναι γραμμένο στη γλώσσα Script και, συνήθως, περιέχει την υπογραφή και το δημόσιο κλειδί του ιδιοκτήτη το οποίο χρησιμοποιείται για την επιβεβαίωση της υπογραφής. Στην Εικόνα 4.6 φαίνεται ως το βήμα 2.

Ένα απλό παράδειγμα για το πώς λειτουργεί το ScriptSig είναι το εξής: Έστω ότι το script κλειδώματος είναι το «12» και έστω ότι το script ξεκλειδώματος είναι το «5+7». Επειδή το άθροισμα στο script ξεκλειδώματος είναι ίσο με τον αναμενόμενο αριθμό στο script κλειδώματος, αυτό καταφέρνει να το ξεκλειδώσει επιτυχώς.

Στο πεδίο του *αριθμού ακολουθίας* ένας αριθμός μικρότερος από το 0xFFFFFFFF υποδηλώνει ότι η συναλλαγή δεν είναι στην τελική της μορφή. Ωστόσο πρόκειται για ένα πεδίο που, λόγω της μη ικανοποιητικής υλοποίησής του, έχει πλέον απενεργοποιηθεί και όλες οι τιμές είναι 0xFFFFFFFF.

Η *Λίστα Εξόδου* αποτελείται από τις καινούργιες εξόδους (UTXOs) με τα ποσά τα οποία τους αντιστοιχούν, καθώς και το απαραίτητο script που θα κλειδώνει τα χρήματα και θα προσδιορίζει τι χρειάζεται για να απελευθερωθούν και να μεταφερθούν. Το script αυτό (ScriptPubKey) αποτελεί το ένα μισό από όσα χρειάζονται για το ξεκλείδωμα και πρέπει να συνδυαστεί με το ScriptSig όταν η συναλλαγή χρησιμοποιηθεί ως είσοδος για να ξεκλειδωθούν τα χρήματα.

Η **Εικόνα 4.7** απεικονίζει τα περιεχόμενα στη Λίστα Εξόδου και ο **Πίνακας 4.3** τα αναλύει με περισσότερη λεπτομέρεια.



Εικόνα 4.7 *Μια πιο αναλυτική παρουσίαση των περιεχομένων στη Λίστα Εξόδου.*

Πεδίο	Περιγραφή	Μέγεθος (bytes)	Αναστροφή στη σειρά των bytes
Τιμή (value)	Η Τιμή σε satoshi[52] των κρυπτονομισμάτων της εξόδου	8	Ναι
Μήκος ScriptPubKey	Δηλώνει το μήκος του κώδικα κλειδώματος που ακολουθεί	Μεταβλητό	Όχι
ScriptPubKey	Ένα script που κλειδώνει την έξοδο	–	Όχι

Πίνακας 4.3 *Ανάλυση των περιεχομένων της Λίστας Εξόδου.*

4.2.1.1 Ανάλυση μιας πραγματικής συναλλαγής στο δίκτυο του Bitcoin

Στη συνέχεια αναλύεται με ένα παράδειγμα η δομή μιας συναλλαγής UTXOs στο δίκτυο του Bitcoin. Θα χρησιμοποιηθεί ως αναφορά μια τυχαία επιλεγμένη συναλλαγή από το **block 500.000**[53]. Πρόκειται για τη 2η συναλλαγή στο block, οπότε ο δείκτης index θα πρέπει να έχει την τιμή 1.

Στην **Εικόνα 4.8** διακρίνονται οι πρώτες συναλλαγές που υπάρχουν στο block 500.000. Στο κόκκινο πλαίσιο βρίσκεται η συναλλαγή που θα αναλυθεί, με hash:

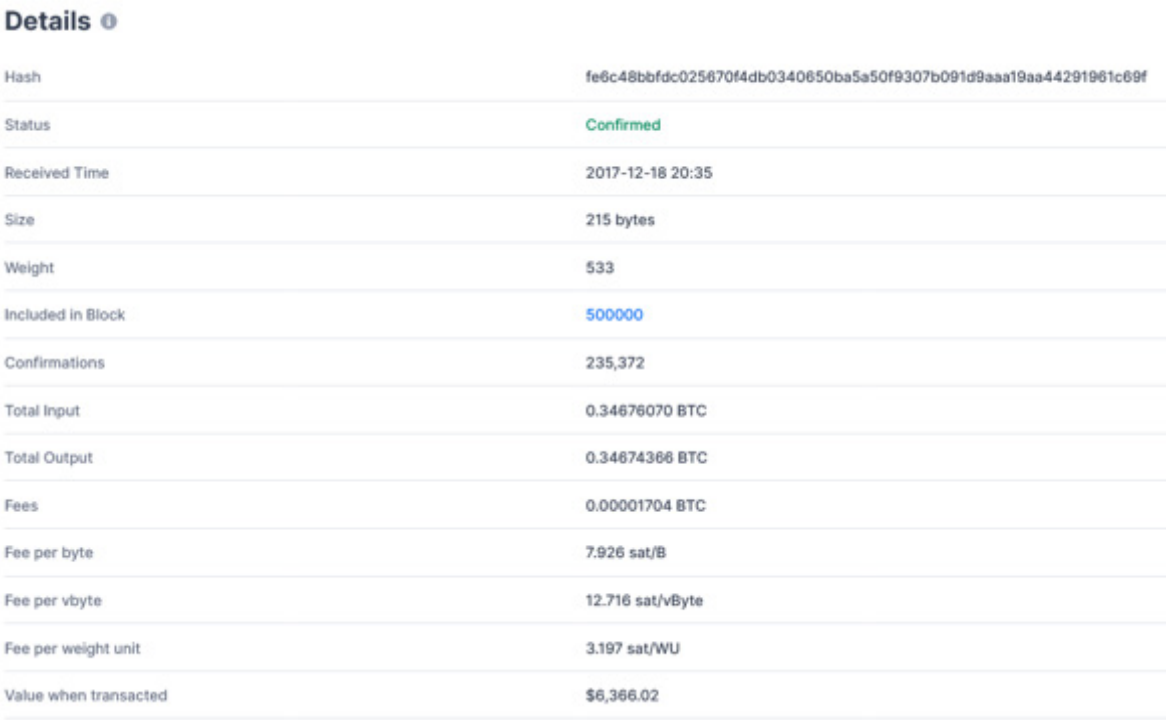
Hash: fe6c48bbfdc025670f4db0340650ba5a50f9307b091d9aaa19aa44291961c69f

Εικόνα 4.8 *Οι πρώτες συναλλαγές στο block 500.000. Μέσα στο πλαίσιο βρίσκεται αυτή του παραδείγματος.*

Λεπτομέρειες της 2ης συναλλαγής στο block 500.000 φαίνονται στην **Εικόνα 4.9**, μέσω ενός προγράμματος block explorer που έχει πρόσβαση στο ledger.

[52] Πρόκειται για υποπολλαπλάσιο του 1 BTC. Για την ακρίβεια 1 satoshi = 10^{-8} BTC.

[53] Μπορείτε να δείτε αναλυτικά τα περιεχόμενα του block 500.000 στον online σύνδεσμο: https://www.blockchain.com/btc/block/500000



Details

Hash	fe6c48bbfdc025670f4db0340650ba5a50f9307b091d9aaa19aa44291961c69f
Status	Confirmed
Received Time	2017-12-18 20:35
Size	215 bytes
Weight	533
Included in Block	500000
Confirmations	235,372
Total Input	0.34676070 BTC
Total Output	0.34674366 BTC
Fees	0.00001704 BTC
Fee per byte	7.926 sat/B
Fee per vbyte	12.716 sat/vByte
Fee per weight unit	3.197 sat/WU
Value when transacted	$6,366.02

Εικόνα 4.9 *Η συναλλαγή του παραδείγματος.*

Επιπλέον, για τη μετάδοσή τους στο δίκτυο, οι συναλλαγές μετατρέπονται σε bits, που αποδίδονται και ως μια μεγάλη σειρά από δεκαεξαδικούς χαρακτήρες (με την ονομασία hex raw), όπως φαίνεται και στη συνέχεια:

Hex raw:
01000000000101d553fbabaf1b26977b6e5d403af9f4b567b3e28484321a6fb02e2824984e3e500000000017160
0142b2296c588ec413cebd19c3cbc04ea830ead6e78ffffffff01be1611020000000017a91487e4e5a7ff7bf78b8a89
72a49381c8a673917f3e870247304402205f39ccbab38b644acea0776d18cb63ce3e37428cbac06dc23b59c6160
7aef69102206b8610827e9cb853ea0ba38983662034bd3575cc1ab118fb66d6a98066fa0bed01210304c01563d
46e38264283b99bb352b46e69bf132431f102d4bd9a9d8dab075e7f00000000

Η μορφή αυτή απεικονίζει ακριβώς όσα φαίνονται στους **Πίνακες 4.1**, **4.2** και **4.3**.

Έτσι, στην **Εικόνα 4.10** με χρώματα φαίνονται τα τμήματα εκείνα της συναλλαγής που αφορούν τα εισαγωγικά πεδία και τις εισόδους:

- της έκδοσης (version number – χρώμα μπλε) με τιμή 01000000,
- το πεδίο του flag (χρώμα γκρι) με τιμή 0001 (προκαθορισμένη τιμή),
- του μετρητή εισόδων (In counter – χρώμα κόκκινο) με τιμή 01 (μία είσοδος),
- της Λίστας Εισόδου. Πιο αναλυτικά:
  - το hash του UTXO (χρώμα πορτοκαλί) με τιμή d553fbabaf1b26977b6e5d403af9f4b567b3e28484321a6fb02e2824984e3e50,
  - τον δείκτη (index – χρώμα κόκκινο) που αναφέρει ποια έξοδος στη συναλλαγή θα χρησιμοποιηθεί, με τιμή 0 (00000000)
  - το πεδίο όπου δηλώνεται το μήκος του ScriptSig (χρώμα πράσινο), με τιμή 17^{H}, δηλαδή 23 bytes (ή 46 δεκαεξαδικοί χαρακτήρες[54])
  - το πεδίο του ScriptSig (χρώμα ανοικτό πράσινο), με τιμή 1600142b2296c588ec413cebd19c3cbc04ea830ead6e78
- το πεδίο της ακολουθίας (με χρώμα μαύρο) και τιμή ffffffff.

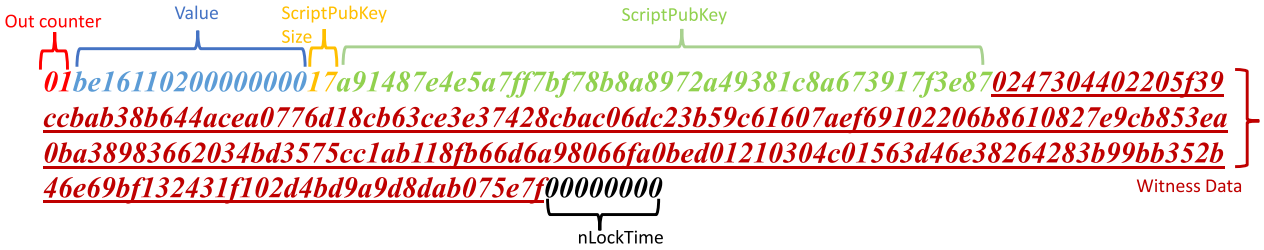
[54] Δύο δεκαεξαδικοί χαρακτήρες ισοδυναμούν με 1 byte.

Εικόνα 4.10 *Τα εισαγωγικά τμήματα της συναλλαγής και το τμήμα της Εισόδου.*

Στη συνέχεια αναλύεται το κομμάτι της συναλλαγής που αφορά την έξοδο (Πίνακας 4.3) και τα τελευταία μεγέθη του Πίνακα 4.1. Στην **Εικόνα 4.11** φαίνονται με χρώματα τα πεδία αυτά.

Πιο αναλυτικά, φαίνονται τα πεδία:

- του Μετρητή Εξόδου (Out Counter – χρώμα κόκκινο), με τιμή 01,
- το πεδίο της Λίστας Εξόδου, που αποτελείται από τα πεδία:
  - της τιμής σε Satoshi της εξόδου (value – χρώμα μπλε), τιμή ίση με 34674366 (ή 00000000021116be στο δεκαεξαδικό)
  - της δήλωσης μήκους του ScriptPubKey (ScriptPubKey size – χρώμα κίτρινο) και τιμή ίση με 17 Hex, δηλαδή 23 bytes
  - του ScriptPubKey (χρώμα πράσινο), τιμή ίση με *a91487e4e5a7ff7bf78b8a8972a49381c8a673917f3e87*
- το πεδίο με τα δεδομένα του Witness (κόκκινο χρώμα) με τιμή ίση με 02473044……075e7f
- το πεδίο nLockTime (μαύρο χρώμα), με τιμή ίση με 00000000.



Εικόνα 4.11 *Τα τελικά πεδία της συναλλαγής και το τμήμα της εξόδου.*

Ολοκληρώνοντας την ανάλυση του παραδείγματος, να αναφερθεί ότι στο Bitcoin δεν γίνεται πουθενά αναφορά για διευθύνσεις (ούτε αποστολέα ούτε παραλήπτη), ούτε επίσης αναφορά για το ποσό που μεταφέρεται.

Σύμφωνα με την ανάλυση που προηγήθηκε, παρατηρείται ότι δεν υπάρχει κάποιο πεδίο της συναλλαγής που δίνει κάποια πληροφορία σχετική με αυτές τις έννοιες. Αυτό όμως δεν εμποδίζει τη συναλλαγή να πραγματοποιηθεί χωρίς προβλήματα ή παρεξηγήσεις. Αυτό συμβαίνει γιατί αυτές οι έννοιες σχηματίζονται από την εφαρμογή του πορτοφολιού (ή τον explorer) που χρησιμοποιείται για διευκόλυνση του χρήστη και δεν είναι απαραίτητες σε επίπεδο πρωτοκόλλου.

Έτσι, για παράδειγμα, η διεύθυνση του αποστολέα μπορεί να μην αναφέρεται, αλλά είναι δυνατόν να επιβεβαιωθεί έμμεσα, καθώς θα είναι ο κάτοχος του κλειδιού που ξεκλειδώνει το UTXO που συμπεριλαμβάνεται στην είσοδο της συναλλαγής. Μόνο αυτός θα μπορεί να το πετύχει αυτό.

Με τον ίδιο έμμεσο τρόπο μπορεί να επιβεβαιωθεί η διεύθυνση του παραλήπτη, καθώς θα μπορεί να ξεκλειδώσει το ScriptPubKey που φαίνεται στη συναλλαγή (βλ. Υποενότητα 4.3.2). Επομένως, η έλλειψη αναφοράς σε διεύθυνση αντιμετωπίζεται επιτυχώς από το πρωτόκολλο του Bitcoin.

Αντίστοιχα, επειδή ένα UTXO δεν μπορεί να σπάσει αλλά μεταφέρεται ακέραιο, γνωρίζοντας το UTXO στην είσοδο θα πρέπει να ανατρέξει ένας κόμβος-αξιολογητής στη συναλλαγή εκείνη στην οποία δημιουργείται αυτό ως έξοδος και εκεί, στο πεδίο value, θα βρει το ποσό που του αντιστοιχεί.

Με τον τρόπο αυτό θα πρέπει να γίνει μια αναζήτηση και ανάγνωση όλων των συναλλαγών που επηρεάζουν την τρέχουσα, προτού αυτή αξιολογηθεί ως έγκυρη και συμπεριληφθεί σε ένα υποψήφιο νέο block.

4.2.2 Δομή συναλλαγής στο δίκτυο Ethereum

Η δομή της συναλλαγής στο δίκτυο του Ethereum έχει ορισμένα κοινά χαρακτηριστικά αλλά και σημαντικές διαφορές με μια συναλλαγή στο δίκτυο του Bitcoin. Το δίκτυο Bitcoin δημιουργεί συναλλαγές βάσει των UTXOs, ενώ το δίκτυο του Ethereum λειτουργεί βάσει ανανέωσης υπολοίπου στους λογαριασμούς.

Στο Ethereum μια συναλλαγή περιλαμβάνει μετάβαση στον χώρο καταστάσεων, όπως αυτός είναι αποθηκευμένος στην EVM στον κάθε κόμβο του δικτύου. Επιπλέον, στο Ethereum γίνεται χρήση του gas για την επεξεργασία και επιβεβαίωση των συναλλαγών (βλ. Κεφάλαιο 2).

Υπενθυμίζεται ότι το Ethereum έχει δύο είδη λογαριασμών: τους *λογαριασμούς χρηστών* (γνωστούς ως *Externally Owned Accounts, EOAs*) και τους *λογαριασμούς συμβάσεων* (*Contracts*). Οι μεν μπορούν να ξεκινούν συναλλαγές, ενώ οι δε μπορούν μόνο να συμπεριληφθούν ως παραλήπτες και δεν έχουν δικαίωμα να δημιουργούν συναλλαγές.

Με αυτά ως δεδομένα, αναλύεται στη συνέχεια η δομή μιας συναλλαγής στο δίκτυο του Ethereum.

Τα πεδία της δομής μιας συναλλαγής στο Ethereum φαίνονται στον **Πίνακα 4.4**, μαζί με μια σύντομη περιγραφή της λειτουργίας τους:

Πεδίο	**Περιγραφή**	**Μέγεθος (bytes)**
Nonce (T_n)	Αύξων αριθμός που μετρά τον αριθμό των επιβεβαιωμένων συναλλαγών από αυτή τη διεύθυνση (EOA)	Μέχρι 32
Gas Price (T_p)	Το ποσό που είναι διατεθειμένος να πληρώσει για το gas ο δημιουργός (τιμές σε wei)	Μέχρι 32
Gas Limit (T_g)	Η μέγιστη ποσότητα σε gas που είναι διατεθειμένος να πληρώσει ο δημιουργός για τη συναλλαγή	Μέχρι 32
To Address (T_t)	Η διεύθυνση του παραλήπτη	20
Value (T_v)	Το ποσό των ethers (ETH) που μεταφέρονται	Μέχρι 32
Data (T_i / T_d)	Προαιρετικό πεδίο για δεδομένα	0 – απεριόριστο
Signature Component v (T_w)	Τα στοιχεία αυτά αποτελούν τμήμα της υπογραφής με χρήση του αλγορίθμου ECDSA βάσει του δημιουργού της συναλλαγής	1 (συνήθως) Τιμές 27 ή 28 (πριν την EIP-155) Τιμές ChainID*2 + 35 ή 36 (μετά την EIP-155)
Signature Component r (T_r)		32
Signature Component s (T_s)		32

Πίνακας 4.4 *Ανάλυση των πεδίων συναλλαγής στο δίκτυο του Ethereum.*

Το πεδίο του *Nonce* είναι αυτό που δημιουργεί συχνά συγχύσεις ως προς τη λειτουργία του. Αυτό προέρχεται εν μέρει από τη συνωνυμία του με τη λειτουργία του Proof of Work στο Bitcoin, όπου και αποτελεί μέρος της εύρεσης του κατάλληλου hash που ικανοποιεί το επίπεδο δυσκολίας για την εύρεση του νέου block.

Στο Ethereum όμως η χρήση του είναι τελείως διαφορετική. Εδώ χρησιμοποιείται για να παρακολουθεί τον αριθμό των επιβεβαιωμένων συναλλαγών που έχουν δημιουργηθεί από μία διεύθυνση. Ο ρόλος του είναι να βοηθήσει στην αποφυγή της δημιουργίας διπλών συναλλαγών από μία διεύθυνση και να επιτρέψει τη χρονική τοποθέτηση των συναλλαγών, για την καλύτερη εξυπηρέτησή τους. Ειδικά, λαμβάνοντας υπόψη ότι στο Ethereum πραγματοποιείται ενημέρωση της κατάστασης της μηχανής μετά από κάθε συναλλαγή, η σειρά με την οποία τυχαίνει να επεξεργαστούν οι συναλλαγές μπορεί να επηρεάσει το αποτέλεσμα του τελικού ποσού στον λογαριασμό. Το nonce επιτρέπει με την αρίθμηση να τοποθετηθούν σε μια χρονική σειρά οι συναλλαγές ενός λογαριασμού.

Στη συνέχεια, τα πεδία *Gas Limit* και *Gas Price* δηλώνουν πόσα είναι διατεθειμένος ο δημιουργός της συναλλαγής να ξοδέψει για την αγορά gas. Το gas θα το χρειαστεί για την επεξεργασία της συναλλαγής από τον minter[55] στο δίκτυο. Επιπλέον, δηλώνεται το συνολικό ποσό που είναι διατεθειμένος να πληρώσει για την ολοκλήρωση της συναλλαγής (βλ. Υποενότητα 2.3.4). Να σημειωθεί μόνο ότι ένα πορτοφόλι μπορεί να ρυθμίσει τις τιμές στα πεδία ανάλογα με την προτεραιότητα που θέλουμε να δοθεί από το δίκτυο στην επεξεργασία της

[55] Minter είναι η ονομασία του miner στο Ethereum. Αντίστοιχα, η διαδικασία ονομάζεται minting και όχι mining.

συναλλαγής μας. Η αύξηση στην τιμή των πεδίων μπορεί να βοηθήσει στη γρηγορότερη επεξεργασία τους από έναν minter του δικτύου.

Το πεδίο *To address* αφορά τη δήλωση της διεύθυνσης του παραλήπτη για τη συναλλαγή.

Τα πεδία *Value* και *Data* είναι τα πεδία που μεταφέρουν το περιεχόμενο μιας συναλλαγής και αφορούν την εκχώρηση της ποσότητας των ETH που μεταφέρονται καθώς και τα δεδομένα που χρειάζεται να μεταφερθούν, αντίστοιχα. Θα πρέπει να τονιστεί ότι τα πεδία αυτά μπορεί να είναι συμπληρωμένα ή όχι. Αν το πεδίο Value έχει τιμή, τότε η συναλλαγή είναι χρηματική και αφορά τη μεταφορά ETH. Αν το πεδίο Data είναι συμπληρωμένο, αφορά τη μεταφορά δεδομένων σε έναν λογαριασμό smart contract (π.χ. αναφορά της συνάρτησης που θα κληθεί).

Μια συναλλαγή κρυπτονομισμάτων μπορεί να έχει παραλήπτη τη διεύθυνση ενός smart contract, αλλά πρέπει να έχει προγραμματιστεί πώς θα τα διαχειριστεί. Ειδάλλως τα χρήματα, πρακτικά, καίγονται, καθώς οι λογαριασμοί smart contract δεν δημιουργούν συναλλαγές. Από την άλλη, η αποστολή δεδομένων προς έναν λογαριασμό χρήστη δεν έχει κάποιο νόημα, από τη στιγμή μάλιστα που οι εφαρμογές για το πορτοφόλι των χρηστών δεν έχουν προγραμματιστεί για να τα διαχειριστούν. Παρ' όλα αυτά, όλες οι παραπάνω συναλλαγές είναι δυνατές. Στην ουσία, δηλαδή, επιτρέπονται και οι 4 δυνατοί συνδυασμοί για το συμπλήρωμα των 2 αυτών πεδίων. Ακόμα και η αποστολή 0 ETH, παρόλο που δεν έχει κάποιο ιδιαίτερο νόημα για το δίκτυο, το οποίο θα τη δεχθεί και θα την επεξεργαστεί κανονικά.

Τέλος, τα πεδία *v*, *r*, *s* εμπεριέχουν την ψηφιακή υπογραφή του δημιουργού. Χρησιμοποιούνται από τον ECDSA αλγόριθμο για την επαλήθευση της ταυτότητας του χρήστη που ξεκινά τη συναλλαγή. Ο τρόπος με τον οποίο συμπληρώνεται εξαρτάται από το αν η συναλλαγή έχει συμβεί πριν ή μετά την EIP155[56] (2014). Για περισσότερες λεπτομέρειες δείτε στην Υποενότητα 4.3.3.

Για τη μεταφορά των συναλλαγών στο δίκτυο του Ethereum θα πρέπει να γίνει μια μετατροπή της πληροφορίας του κάθε πεδίου σε μια σειρά από bits, ακολουθώντας μια κοινή δομή για αυτό. Ο μηχανισμός που εφαρμόζεται στο Ethereum για τη λειτουργία αυτή είναι η τεχνική του *Recursive Length Prefix* (*RLP*), που επιτρέπει την κωδικοποίηση και αποκωδικοποίηση των συναλλαγών.

Ο RLP παίρνει ένα αντικείμενο[57] και το απεικονίζει ως bytes ή ως πίνακα, που είναι και οι μόνοι τύποι δεδομένων που χρησιμοποιούνται από τον RLP. Επιπλέον, όντας ντετερμινιστικός αλγόριθμος, η τοποθέτηση των στοιχείων στους πίνακες είναι καθορισμένη, με αποτέλεσμα η κωδικοποίηση αυτή να επιτυγχάνεται κάθε φορά που δοκιμάζεται η ίδια είσοδος.

Στον αλγόριθμο RLP, επειδή δεν μεταφέρονται οι ονομασίες των πεδίων, υπάρχει πάντα ένας δείκτης που αναφέρει το μήκος του κάθε πεδίου, για να διευκολύνει την αποκωδικοποίηση των δεδομένων. Εξαιτίας αυτού, το τελικό μέγεθος της συναλλαγής θα είναι ελαφρώς μεγαλύτερο από το άθροισμα σε bytes των πεδίων του Πίνακα 4.4.

4.2.2.1 Ανάλυση μιας πραγματικής συναλλαγής στο δίκτυο του Ethereum

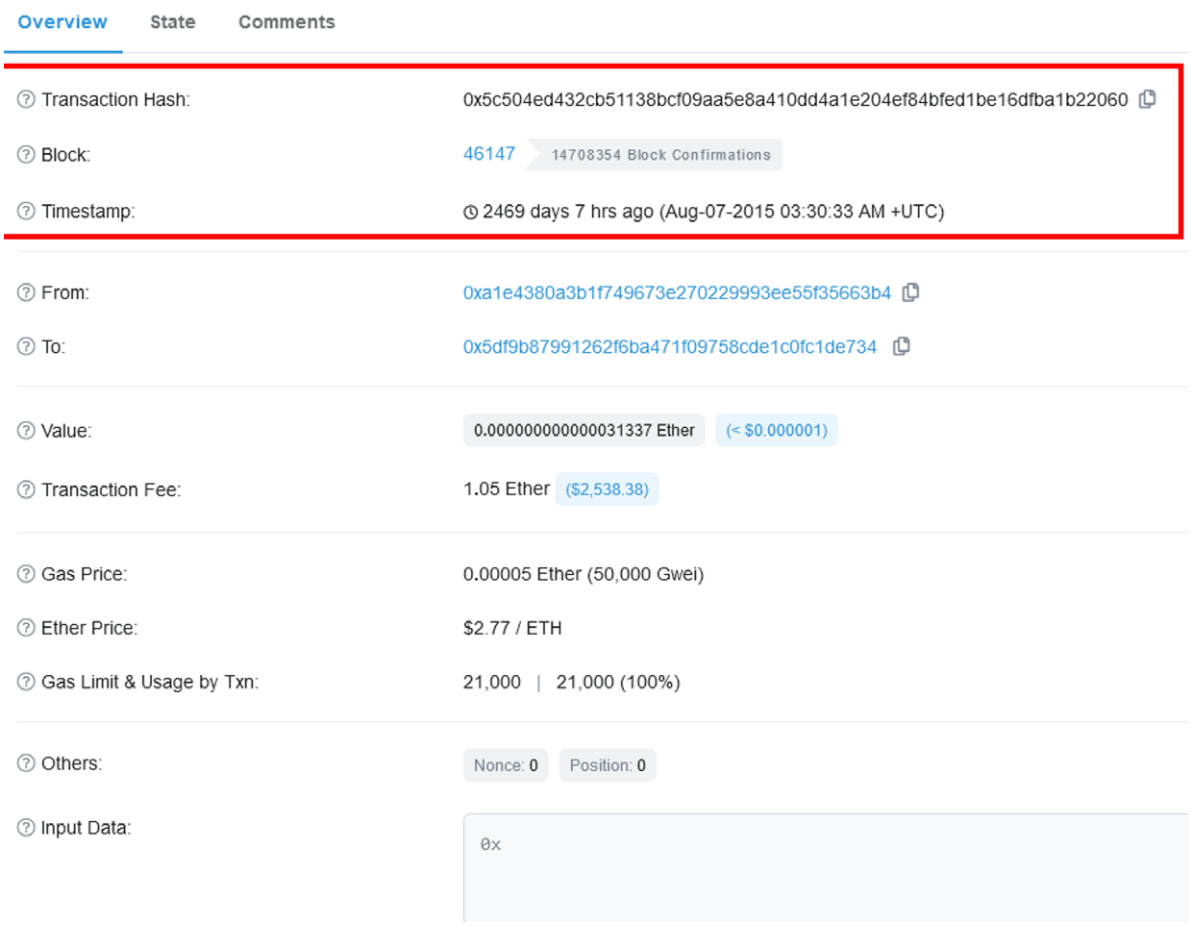
Στη συνέχεια θα αναλυθούν δύο παραδείγματα συναλλαγών στο δίκτυο του Ethereum. Ένα που αναλύει μια συναλλαγή πριν από την εφαρμογή της EIP-155 και ένα μετά. Σκοπός είναι να εντοπιστούν οι διαφορές και να εξηγηθούν. Να σημειωθεί ότι τα παραδείγματα που χρησιμοποιούνται αφορούν απλές χρηματικές συναλλαγές και στις δύο περιπτώσεις.

Πριν την EIP-155: Η συναλλαγή επιλέγεται τυχαία και είναι μια από τις πρώτες που καταγράφηκαν. Στην **Εικόνα 4.12** φαίνεται η ανάλυση της συναλλαγής αυτής στην εφαρμογή Etherscan, που είναι μια εφαρμογή που απεικονίζει τα δεδομένα από το ledger. Συνήθως είναι συνδεδεμένη με έναν πλήρη κόμβο για να έχει πρόσβαση σε όλη την πληροφορία που είναι αποθηκευμένη στο ledger. Η συγκεκριμένη εφαρμογή είναι ιδιαίτερα δημοφιλής στο Ethereum.

---

[56] Ethereum Improvement Proposal (EIP)-155: Δημιουργήθηκε το 2016 από τον V. Buterin και έχει σκοπό να αποτρέψει τη δημιουργία replay attacks ενσωματώνοντας πληροφορίες όπως το Chain-ID στη συναλλαγή. Με τον τρόπο αυτό, μια συναλλαγή θα μπορούσε να διεκπεραιωθεί μόνο σε μια αλυσίδα (chain) και όχι σε άλλες.

[57] Ως ένα αντικείμενο θεωρούνται ένα string (κενό ή μη) ή και μια λίστα από strings (ή άλλα αντικείμενα).



Εικόνα 4.12 *Τα πεδία της συναλλαγής του παραδείγματος όπως φαίνονται στο Etherscan.*

Στο κόκκινο πλαίσιο φαίνεται ότι η συναλλαγή βρίσκεται στο **block 46147**, το οποίο δημιουργήθηκε στις 7/8/2015 και έχει hash:

5c504ed432cb51138bcf09aa5e8a410dd4a1e204ef84bfed1be16dfba1b22060.

Για τη μελέτη της συναλλαγής θα χρησιμοποιηθεί η δεκαεξαδική της μορφή:

```
0xf8678086 2d79883d2000 82 5208 94 5df9b87991262f6ba471f09758cde1c0fc1de734 82 7a69 80 1c a0
88ff6cf0fefd94db46111149ae4bfc179e9b94721fffd821d38d16464b3f71d0 a0 45e0aff800961c
fce805daef7016b9b675c137a6a41a548f7b60a3484c06a33a
```

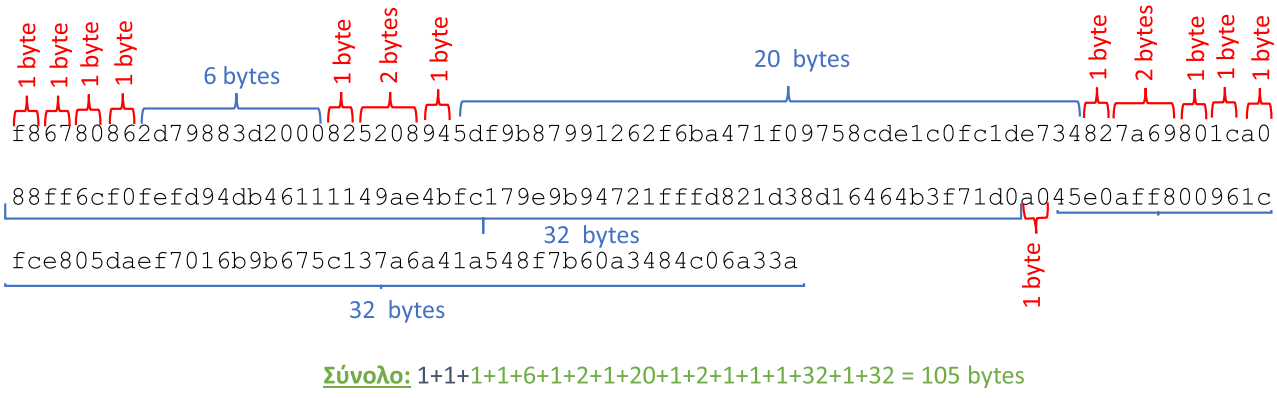
Με έντονα γράμματα φαίνονται τα bytes εκείνα που προστίθενται από την εφαρμογή του RLP για να δώσουν πληροφορία για το μήκος των πεδίων του Πίνακα 4.4.

Ακολουθεί η ανάλυση των τιμών:

- Η τιμή **0x** στην αρχή υποδηλώνει ότι ο αριθμός που ακολουθεί είναι στο δεκαεξαδικό. Δεν προσμετράται στο μέγεθος της συναλλαγής.
- Η τιμή **f8** υποδηλώνει ότι το συνολικό ωφέλιμο φορτίο είναι μεγαλύτερο από 55 bytes και βοηθά στον προσδιορισμό του μήκους αυτού αφαιρώντας την τιμή από την αρχική, που είναι f7 (f8 – f7 = 01 byte)
- Η τιμή **67** (103 στο δεκαδικό) δηλώνει το μήκος του πεδίου. Άρα 103 bytes είναι το μήκος της συναλλαγής. Η τιμή αυτή θα επιβεβαιωθεί στο τέλος της ανάλυσης των τιμών των πεδίων.
- Η τιμή **80** δείχνει την τιμή για το πεδίο Nonce. Για την ακρίβεια, στην RLP το 0 είναι το 80, επομένως το nonce είναι ίσο με 0.
- Η τιμή **86** δείχνει το μήκος του πεδίου που αφορά το Gas Price. Είναι 6 bytes (δηλαδή 12 δεκαεξαδικοί χαρακτήρες).

- Η τιμή **2d79883d2000** δείχνει το πεδίο Gas Price. Μετατρέποντάς το στο δεκαδικό στην τιμή $5*10^{-5}$ ETH.
- Η τιμή **82** δείχνει το μήκος του πεδίου που αφορά το Gas Limit. Είναι 2 bytes.
- Η τιμή **5208** δείχνει το Gas Price. Με μετατροπή στο δεκαδικό βρίσκεται η τιμή 21000.
- Η τιμή **94** δείχνει το μέγεθος του επόμενου πεδίου, που αφορά τη διεύθυνση του παραλήπτη. Σε αυτή πρέπει να αφαιρεθεί η τιμή 80^{H} (σύμφωνα με το RLP), και έτσι βρίσκεται ότι το μέγεθος είναι 16^{H}, δηλαδή 20 bytes.
- Η τιμή **5df9b87991262f6ba471f09758cde1c0fc1de734** είναι η διεύθυνση του παραλήπτη (επιβεβαιώνεται και στην Εικόνα 4.12).
- Η τιμή **82** δείχνει το μήκος του πεδίου που αφορά το ποσό που μεταφέρεται. Είναι στα επόμενα 2 bytes.
- Η τιμή **7a69** δείχνει το ποσό σε wei[58]. Μετατρέποντάς το στο δεκαδικό, υπολογίζεται ότι είναι 31337.
- Η τιμή **80** δείχνει το μήκος του πεδίου δεδομένων που είναι κενό (0 bytes)
- Η τιμή **1c** δείχνει το byte για την απεικόνιση του τμήματος v της υπογραφής. Στο δεκαδικό μεταφράζεται σε 28.
- Η τιμή **a0** δείχνει το μήκος του πεδίου r (συντεταγμένη x στην ελλειπτική καμπύλη). Για την εύρεση της τελικής τιμής πρέπει να γίνει η πράξη $a0 - 80^{H} = 20^{H}$ ή αλλιώς 32 bytes.
- Η τιμή **88ff6cf0fefd94db46111149ae4bfc179e9b94721fffd821d38d16464b3f71d0** υποδηλώνει τη συντεταγμένη x.
- Η τιμή **a0** δείχνει το μήκος του πεδίου s (συντεταγμένη y στην ελλειπτική καμπύλη). Για την εύρεση της τελικής τιμής πρέπει να γίνει η πράξη $a0^{H} - 80^{H} = 20^{H}$ ή αλλιώς 32 bytes.
- Η τιμή **45e0aff800961cfce805daef7016b9b675c137a6a41a548f7b60a3484c06a33a** υποδηλώνει τη συντεταγμένη y.

Στην **Εικόνα 4.13** φαίνεται το συνολικό μήκος της συναλλαγής που αναλύθηκε παραπάνω, το οποίο είναι ίσο με 105 bytes. Στην παραπάνω ανάλυση αναφέρθηκε η πληροφορία που δόθηκε στο 2ο byte, που υποδείκνυε ότι το ωφέλιμο φορτίο είναι ίσο με 103 bytes. Αυτό επαληθεύεται καθώς αν από τα 105 (συνολικά) αφαιρέσουμε τα 2 αρχικά θα μείνουν πράγματι 103 bytes, που ορθώς περάσαν ως πληροφορία.



Εικόνα 4.13 *Το μήκος σε bytes του κάθε πεδίου της συναλλαγής και το άθροισμα όλων των μηκών των πεδίων.*

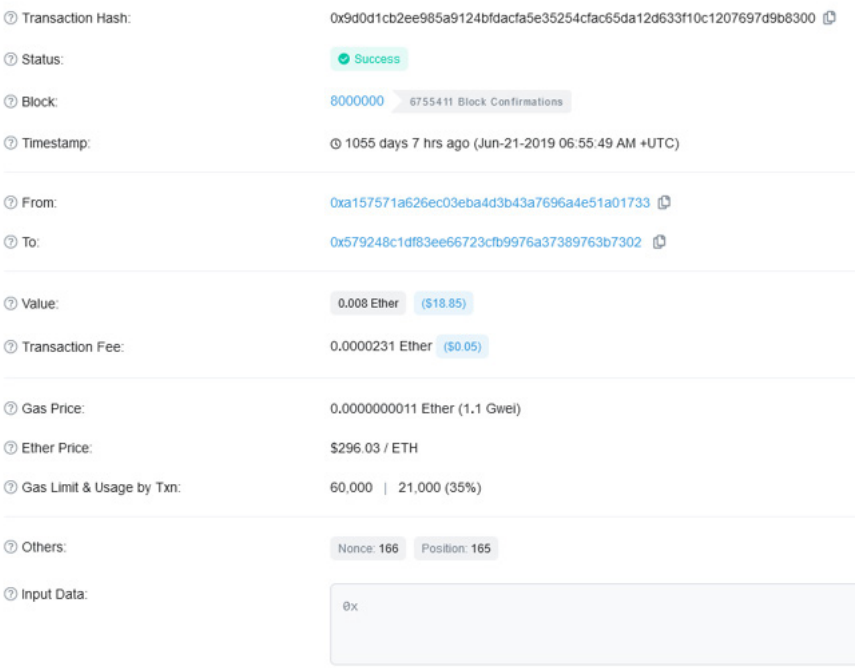
Μετά την EIP-155: Στη συνέχεια αναλύεται μια πιο πρόσφατη συναλλαγή η οποία ακολουθεί τις αλλαγές που ενσωματώθηκαν με την υιοθέτηση της πρότασης EIP-155. Στη δομή της συναλλαγής οι αλλαγές επικεντρώνονται κυρίως στο περιεχόμενο του τμήματος *v* της υπογραφής, που τώρα περιέχει και πληροφορία σχετική με την αλυσίδα (δηλαδή το δίκτυο) στην οποία και θα καταγραφεί η συναλλαγή.

[58] Υπενθυμίζεται ότι 1 wei = 10^{-18} ETH (βλ. Εικόνα 2.12).

Για την ανάλυση επιλέχθηκε μια συναλλαγή που βρίσκεται στο **block 8.000.000**, το οποίο και δημιουργήθηκε στις 21/7/2019 και περιέχει 168 συναλλαγές.

Η επιλογή της συναλλαγής μέσα στο block έγινε τυχαία.

Η ανάλυση της στο Etherscan φαίνεται στην **Εικόνα 4.14**.



Εικόνα 4.14 *Τα πεδία της συναλλαγής του παραδείγματος όπως φαίνονται στο Etherscan.*

Το hash της είναι το:

0x9d0d1cb2ee985a9124bfdacfa5e35254cfac65da12d633f10c1207697d9b8300

Σε δεκαεξαδική μορφή η συναλλαγή γράφεται:

```
0xf86b81a6844190ab0082ea6094579248c1df83ee66723cfb9976a37389763b730287 1c6bf526340
000802 5a0f8d6574f15155e2cdb65341b613f256ed1a32a0e75a5647e957b835496e8b0a2a059488f
13833a95074560ec4b56e9178e52416423766756a4bb40bb91e1b89171
```

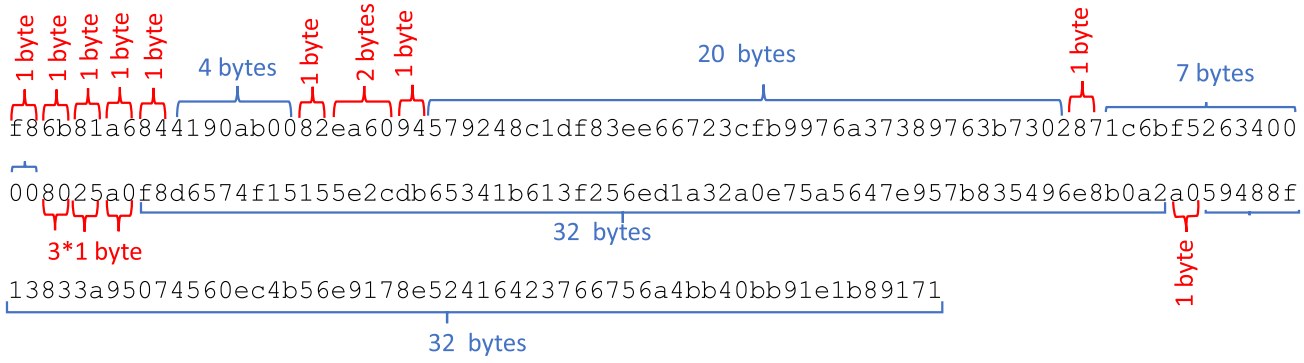
Και πάλι με έντονα γράμματα φαίνονται τα bytes που δείχνουν υπόδειξη για το μέγεθος του πεδίου.

Ακολουθεί η ανάλυση της συναλλαγής:

- Η τιμή **0x** στην αρχή υποδηλώνει ότι ο αριθμός που ακολουθεί είναι στο δεκαεξαδικό. Δεν προσμετράται στο μέγεθος της συναλλαγής.
- Η τιμή **f8** υποδηλώνει ότι το συνολικό ωφέλιμο φορτίο είναι μεγαλύτερο από 55 bytes και βοηθά στον προσδιορισμό του μήκους αυτού αφαιρώντας την τιμή από την αρχική, που είναι f7 (f8 – f7 = 01 byte)
- Η τιμή **6b** (107 στο δεκαδικό) δηλώνει το μήκος του πεδίου. Άρα 107 bytes είναι το μήκος της συναλλαγής. Η τιμή αυτή θα επιβεβαιωθεί στο τέλος της ανάλυσης των τιμών των πεδίων.
- Η τιμή **81** δείχνει ότι το επόμενο byte θα δώσει το περιεχόμενο του πεδίου Nonce.
- Η τιμή **a6** δείχνει το nonce του λογαριασμού. Στο δεκαδικό ο αριθμός αυτός είναι ίσος με 166 (επιβεβαιώστε και από την Εικόνα 4.14).
- Η τιμή **84** δείχνει το μήκος του πεδίου που αφορά το Gas Price. Είναι 4 bytes.
- Η τιμή **4190ab00** δείχνει το πεδίο Gas Price. Μετατρέποντάς το στο δεκαδικό, στην τιμή $11*10^{-8}$ ETH.

- Η τιμή **82** δείχνει το μήκος του πεδίου που αφορά το Gas Limit. Είναι 2 bytes.
- Η τιμή **ea60** δείχνει το Gas Price. Με μετατροπή στο δεκαδικό βρίσκεται η τιμή 60.000.
- Η τιμή **94** δείχνει το μέγεθος του επόμενου πεδίου, που αφορά τη διεύθυνση του παραλήπτη. Σε αυτή πρέπει να αφαιρεθεί η τιμή 80^{H} (σύμφωνα με το RLP) και έτσι βρίσκεται ότι το μέγεθος είναι 16^{H}, δηλαδή 20 bytes.
- Η τιμή **579248c1df83ee66723cfb9976a37389763b7302** είναι η διεύθυνση του παραλήπτη (επιβεβαιώνεται και στην Εικόνα 4.14).
- Η τιμή **87** δείχνει το μήκος του πεδίου που αφορά το ποσό που μεταφέρεται. Είναι στα επόμενα 7 bytes.
- Η τιμή **1c6bf526340000** δείχνει το ποσό σε wei[59]. Μετατρέποντάς το στο δεκαδικό, υπολογίζεται ότι είναι $8*10^{15}$. Δηλαδή $8*10^{-3}$ ETH.
- Η τιμή **80** δείχνει το μήκος του πεδίου δεδομένων που είναι κενό (0 bytes).
- Η τιμή **25** δείχνει το byte για την απεικόνιση του τμήματος v της υπογραφής. Στο δεκαδικό μεταφράζεται σε 37.
- Η τιμή **a0** δείχνει το μήκος του πεδίου r (συντεταγμένη x στην ελλειπτική καμπύλη). Για την εύρεση της τελικής τιμής πρέπει να γίνει η πράξη $a0 - 80^{H} = 20^{H}$ ή αλλιώς 32 bytes.
- Η τιμή **f8d6574f15155e2cdb65341b613f256ed1a32a0e75a5647e957b835496e8b0a2** υποδηλώνει τη συντεταγμένη x.
- Η τιμή **a0** δείχνει το μήκος του πεδίου s (συντεταγμένη y στην ελλειπτική καμπύλη). Για την εύρεση της τελικής τιμής πρέπει να γίνει η πράξη $a0^{H} - 80^{H} = 20^{H}$ ή αλλιώς 32 bytes.
- Η τιμή **59488f13833a95074560ec4b56e9178e52416423766756a4bb40bb91e1b89171** υποδηλώνει τη συντεταγμένη y.

Στην **Εικόνα 4.15** ακολουθεί η ανάλυση του μήκους της συναλλαγής σε bytes με βάση όσα αναφέρονται και από τα πεδία (σημειωμένα με bold). Έτσι ισχύει ότι, αν αφαιρεθούν τα 2 πρώτα bytes που έχουν περάσει πριν δοθεί η πληροφορία για το μήκος της συναλλαγής, τότε όντως αυτή έχει το αναμενόμενο μήκος των 107 bytes.



Σύνολο: 1+1+1+1+1+4+1+2+1+20+1+7+1+1+1+32+1+32 = 109 bytes

Εικόνα 4.15 *Το μήκος σε bytes του κάθε πεδίου της συναλλαγής και το άθροισμα όλων των μηκών των πεδίων.*

Περισσότερη ανάλυση για τη χρήση των πεδίων r, s, v ακολουθεί στην Ενότητα 4.3.

4.3 Χρήση ψηφιακών υπογραφών

Ο ρόλος των ψηφιακών υπογραφών σε ένα δίκτυο blockchain περιγράφεται από δύο βασικά αποτελέσματα που προκύπτουν από τη χρήση τους:

[59] Υπενθυμίζεται ότι 1 wei = 10^{-18} ETH. Περισσότερα στην Εικόνα 2.12 (Κεφάλαιο 2).

- Τη δυνατότητα του κατόχου κρυπτονομισμάτων να αποδείξει την ιδιοκτησία αυτών και να επιτρέψει τη μεταφορά τους σε μια συναλλαγή με τρόπο αδιάψευστο από τον οποιονδήποτε.
- Να αποτρέψει την οποιαδήποτε αλλαγή στη συναλλαγή μετά την υπογραφή της από τον δημιουργό της.

Με την επίτευξη των δύο αυτών βασικών χαρακτηριστικών η χρήση των ψηφιακών υπογραφών παίζει έναν πολύ σημαντικό ρόλο, καθώς επιτυγχάνει την αύξηση της εμπιστοσύνης στο κατανεμημένο δίκτυο blockchain.

Όπως έχει αναφερθεί, τόσο το Bitcoin όσο και το Ethereum χρησιμοποιούν ψηφιακές υπογραφές και μάλιστα εφαρμόζουν τον αλγόριθμο ECDSA (Elliptic Curve Digital Signature Algorithm) για να τις δημιουργήσουν.[60]

Στη συνέχεια, αναλύεται ο τρόπος με τον οποίο δημιουργούνται και επαληθεύονται οι ψηφιακές υπογραφές στα δύο αυτά βασικά δίκτυα Blockchain.

4.3.1 Ψηφιακές υπογραφές στις συναλλαγές

Μια ψηφιακή υπογραφή με χρήση του ECDSA αποτελείται από τον συνδυασμό δύο μερών:

ένα *τυχαίο* μέρος (r) + το μέρος της *υπογραφής* (s)

Ο υπολογισμός αυτών των δύο μερών γίνεται ως εξής:

- *Τυχαίο μέρος (r)*: Αρχικά, δημιουργείται ένας τυχαίος αριθμός (n). Αυτός ο τυχαίος αριθμός, για λόγους ασφαλείας, δεν θα πρέπει να δημιουργηθεί από μια συνάρτηση προγραμματισμού υπεύθυνη για τη δημιουργία τυχαίων αριθμών, ακόμα και αν αυτή δέχεται ως όρισμα έναν αριθμό που έχει προέλθει από μια πηγή εντροπίας.
  Έτσι, εφαρμόζεται το πρότυπο RFC 6979 για την επιλογή του αρχικού, τυχαίου αριθμού k (Antonopoulos, 2017), ο οποίος πρέπει να δημιουργείται για κάθε συναλλαγή. Δεδομένου του τυχαίου αυτού αριθμού, γίνεται χρήση της ελλειπτικής καμπύλης (βλ. Κεφάλαιο 3). Εκεί, ξεκινώντας από το σημείο γεννήτορα G, εκτελείται ο πολλαπλασιασμός k*G. Το αποτέλεσμα είναι ένα (άλλο) σημείο R στην καμπύλη, το οποίο και θα αποτελέσει το τυχαίο τμήμα της υπογραφής. Για την ακρίβεια, το τυχαίο τμήμα αποτελείται μόνο από τη συντεταγμένη του άξονα x του σημείου R (R_x), δηλαδή $r = R_x$.
- *Τμήμα υπογραφής (s)*: Για το δεύτερο κομμάτι της υπογραφής χρησιμοποιείται το ιδιωτικό κλειδί του χρήστη καθώς και το hash της συναλλαγής για να συνδυαστεί μοναδικά με το κλειδί.
  Πιο αναλυτικά, πολλαπλασιάζεται το ιδιωτικό κλειδί του χρήστη (p) με το σημείο R (συγκεκριμένα με το R_x). Το αποτέλεσμα του πολλαπλασιασμού συνδυάζεται με το hash των δεδομένων της συναλλαγής (Tr), έτσι ώστε να μην μπορεί να χρησιμοποιηθεί σε άλλη συναλλαγή. Τέλος, το τμήμα s υπολογίζεται με διαίρεση του αποτελέσματος με τον αρχικό τυχαίο αριθμό k, που υπολογίστηκε στην αρχή της διαδικασίας υπολογισμού του μέρους r.

Ο παρακάτω τύπος αποτυπώνει τη διαδικασία υπολογισμού του τμήματος s:

$$s = \frac{R_x * p + hash(Tr)}{k} \qquad (4.1)$$

Η ψηφιακή υπογραφή D είναι ο συνδυασμός των δύο αυτών αριθμών: D(r,s).

Η διαδικασία αυτή ακολουθείται τόσο στο δίκτυο του Ethereum όσο και σε αυτό του Bitcoin. Η βασική διαφορά τους είναι στη συνάρτηση κατακερματισμού που χρησιμοποιούν (βλ. Κεφάλαιο 3).

4.3.1.1 Επαλήθευση

Η διαδικασία της επαλήθευσης μιας ψηφιακής υπογραφής έχει ως σκοπό να αποδειχθεί ότι ο χρήστης που ξεκινά τη συναλλαγή είναι όντως αυτός που έχει τη δυνατότητα να το κάνει αυτό. Για να γίνει αυτό, είναι αρκετό να φανεί ότι το δημόσιο κλειδί του χρήστη είναι αυτό που δημιούργησε την υπογραφή. Μάλιστα, σε όλη τη διαδικασία της επαλήθευσης το ιδιωτικό κλειδί του χρήστη δεν χρειάζεται να εμφανιστεί πουθενά.

[60] Το Bitcoin από το block 709.632 (Taproot update) ενεργοποίησε και τη χρήση των ψηφιακών υπογραφών Schnorr παράλληλα με την ECDSA. Το Ethereum εξετάζει τη χρήση υπογραφών BLS (Boneh-Lynn-Schacham) στην έκδοση 2.0 του πρωτοκόλλου, χωρίς να επηρεάζεται η χρήση του ECDSA για την έκδοση 1.0.

Για την επαλήθευση θα πρέπει να γίνουν οι πράξεις που συμπεριλαμβάνουν δύο σημεία πάνω στην καμπύλη της ECDSA. Η πρόσθεση μεταξύ των δύο αυτών σημείων υποδεικνύει ένα τρίτο σημείο το οποίο πρέπει, για να ολοκληρωθεί ορθά η διαδικασία, να έχει την ίδια συντεταγμένη -x με το τυχαίο σημείο R (R_x).

Τα δύο σημεία στην καμπύλη ορίζονται ως εξής:

Σημείο 1: Για να βρεθεί, πρέπει να πολλαπλασιαστεί το σημείο G (γεννήτορας) με τον αριθμό d. Ο αριθμός d είναι ίσος με το αποτέλεσμα της διαίρεσης του hash της συναλλαγής [hash(Tr)] με τον αριθμό s (εξίσωση 1). Δηλαδή καταλήγουμε στο σημείο 1 με την ολοκλήρωση της πράξης:

$$G * hash(Tr) / s \qquad (4.2)$$

Σημείο 2: Για να βρεθεί, πολλαπλασιάζεται το δημόσιο κλειδί P με το R_x/s, όπως φαίνεται και στην εξίσωση (4.3):

$$P * R_x / s \qquad (4.3)$$

Η πρόσθεση των δύο αυτών σημείων γίνεται με τη χάραξη της ευθείας που περνά από τα δύο αυτά σημεία πάνω στην ελλειπτική καμπύλη. Η ευθεία αυτή τέμνει την ελλειπτική καμπύλη σε ένα τρίτο σημείο. Όπως αναφέρθηκε και νωρίτερα, πρέπει η συντεταγμένη x του τρίτου αυτού σημείου να είναι ίση με την R_x.

Στη συνέχεια αναλύεται ο τρόπος που εφαρμόζεται στην πράξη η υπογραφή και η επιβεβαίωσή της στα δίκτυα του Bitcoin και του Ethereum.

4.3.2 Ένα παράδειγμα επαλήθευσης της υπογραφής στο δίκτυο του Bitcoin

Στο Bitcoin η ψηφιακή υπογραφή μπαίνει σε κάθε είσοδο UTXO που συμμετέχει στη συναλλαγή για να ξεκλειδώσει τα συνδεδεμένα bitcoins, ενώ, ταυτόχρονα, η συνθήκη κλειδώματος τοποθετείται σε κάθε έξοδο.

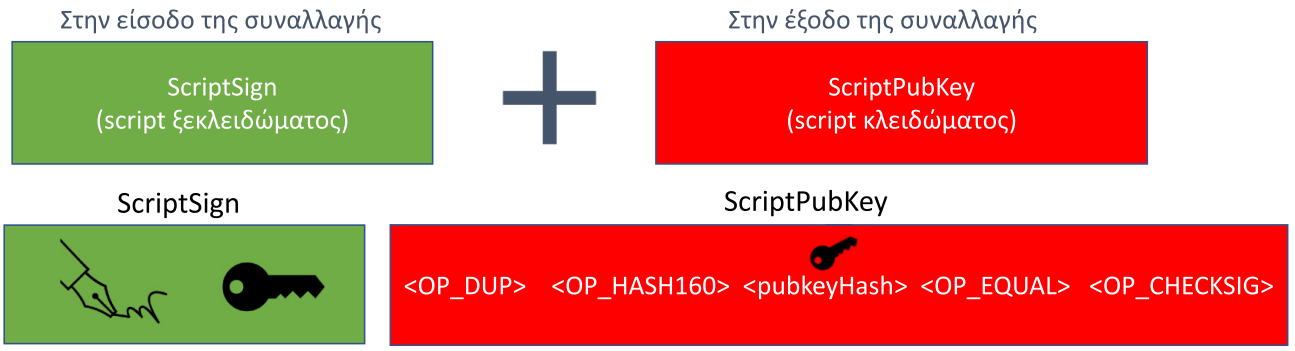
Για το κλείδωμα και το ξεκλείδωμα της πληροφορίας το Bitcoin χρησιμοποιεί τη γλώσσα *Script*, μια γλώσσα που χρησιμοποιεί ως δομή δεδομένων μια στοίβα/σωρό (stack). Η γλώσσα περιγράφει τις ενέργειες που πρέπει να γίνουν στα δεδομένα, λαμβάνοντας υπόψη ότι σε μια στοίβα το πρόγραμμα μπορεί να τοποθετήσει ή να αφαιρέσει δεδομένα σύμφωνα με τον αλγόριθμο *First In First Out* (*FIFO*).

Για να μπορέσει να γίνει η διαδικασία της επαλήθευσης, πρέπει να συνδυαστεί η πληροφορία που βρίσκεται:

- Στο *script κλειδώματος*, το οποίο και έχει δημιουργηθεί στη συναλλαγή που εμφανίζεται ως έξοδος το συγκεκριμένο UTXO. Σύμφωνα με την προηγούμενη ανάλυση, το script κλειδώματος περιέχεται στο πεδίο *ScriptPubKey* και περιλαμβάνει τη συνθήκη που πρέπει να ικανοποιείται για να ξεκλειδώσουν τα χρήματα.
- Στο *script ξεκλειδώματος*, το οποίο και προστίθεται στην είσοδο της νέας συναλλαγής για να ξεκλειδώσει τα χρήματα. Όπως αναφέρθηκε, το script ξεκλειδώματος περιέχεται στο πεδίο *ScriptSig*.

Στην **Εικόνα 4.16** φαίνεται ο τρόπος με τον οποίο η πληροφορία στα δύο αυτά πεδία πρέπει να συνδυαστεί για να έχει το επιθυμητό αποτέλεσμα. Η γλώσσα script εκτελεί τις εντολές με σειρά από αριστερά προς τα δεξιά και ενημερώνει τη στοίβα για το τελικό αποτέλεσμα. Έτσι, αντιγράφει το περιεχόμενο του ScriptSig και κατόπιν το περιεχόμενο του *ScriptPubKey*.

Για να είναι επιτυχής η διαδικασία, θα πρέπει στο τέλος αυτής η στοίβα να περιέχει την τιμή TRUE (στην πράξη τον αριθμό 01 ή κάποιον άλλο θετικό αριθμό).



Εικόνα 4.16 *Επαλήθευση των συνθηκών ολοκλήρωσης μιας συναλλαγής.*

Το περιεχόμενο του ScriptPubKey εξαρτάται από το είδος της συναλλαγής που πραγματοποιείται. Μια τέτοια συναλλαγή μπορεί να είναι τύπου:

- *Pay To Pubkey*: Απαιτεί από το script ξεκλειδώματος να περιέχει το δημόσιο κλειδί του χρήστη για το ξεκλείδωμα των χρημάτων.
- *Pay To PubkeyHash*: Απαιτεί από το script ξεκλειδώματος να περιέχει το hash από το δημόσιο κλειδί για το ξεκλείδωμα των χρημάτων.
- *Pay To Multisig*: Απαιτεί από το script ξεκλειδώματος να περιέχει τις υπογραφές ορισμένων ή όλων των δημόσιων κλειδιών που έχουν συμπεριληφθεί στο κλείδωμα των χρημάτων.
- *Pay To Script Hash*: Απαιτεί από το script ξεκλειδώματος να περιέχει το κλειδί για το script που έχει δημιουργήσει ο χρήστης.

Στη συνέχεια αναλύεται το παράδειγμα ενός script που αφορά μια συναλλαγή που χρησιμοποιεί *Pay To PubkeyHash* τύπο κλειδώματος. Πρόκειται για την πλέον συνηθισμένη περίπτωση συναλλαγής στο δίκτυο του Bitcoin.[61]

Στην περίπτωση αυτή, το script κλειδώματος έχει συνήθως το εξής περιεχόμενο:

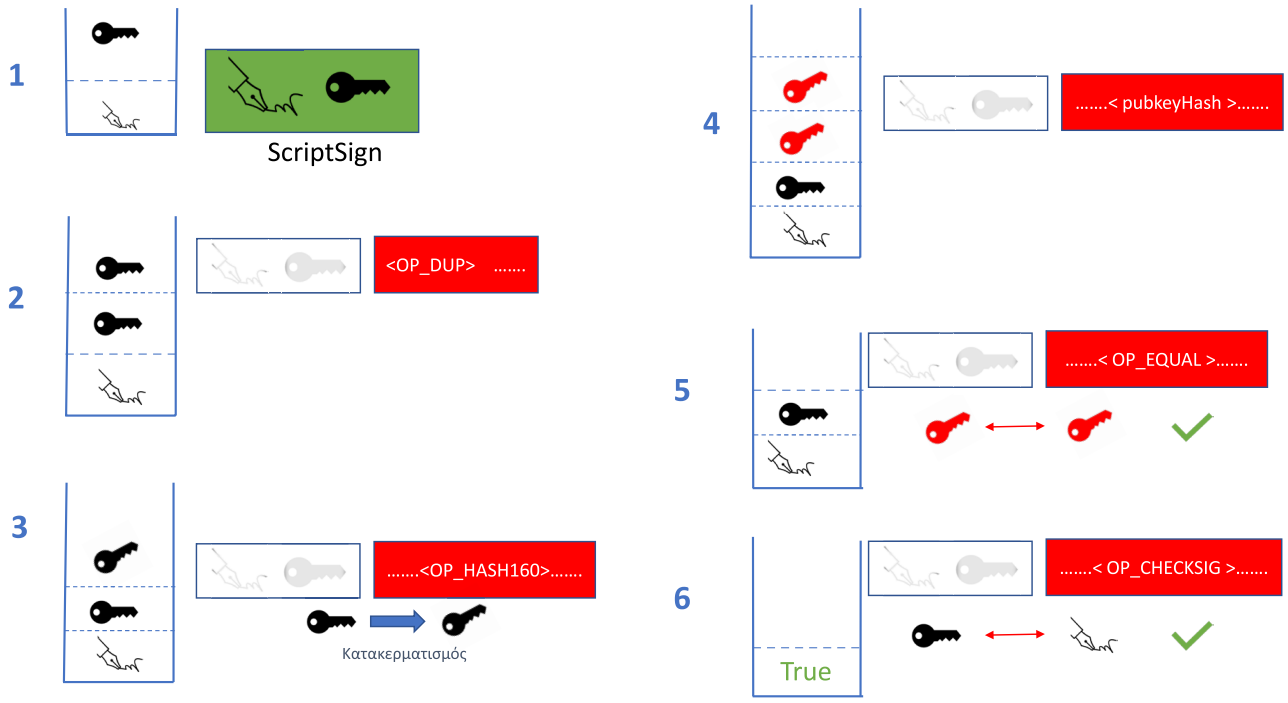
ScriptPubKey = OP_DUP OP_HASH160 <HASH ΔΗΜΟΣΙΟΥ ΚΛΕΙΔΙΟΥ> OP_EQUAL OP_CHECKSIG

Ενώ το ScriptSig = <Υπογραφή> <Δημόσιο Κλειδί>

Βάζοντάς τα μαζί στη μορφή που φαίνεται στην Εικόνα 4.16, δημιουργείται η συνθήκη που απεικονίζεται στο κάτω μέρος. Ξεκινώντας την επαλήθευση από αριστερά προς τα δεξιά, πραγματοποιούνται τα εξής βήματα, που απεικονίζονται και στην **Εικόνα 4.17**:

1. Τοποθετούνται στη στοίβα η υπογραφή και το δημόσιο κλειδί του χρήστη (ScriptSign).
2. Το δημόσιο κλειδί αντιγράφεται και μπαίνει στην κορυφή της στοίβας σύμφωνα με την εντολή *ScriptPubKey* (*OP_DUP*).
3. Από το αντίγραφο που είναι πάνω στη στοίβα υπολογίζεται το hash και μπαίνει στη θέση του στη στοίβα σύμφωνα με την εντολή *ScriptPubKey* (*OP_HASH160*).
4. Το hash του δημόσιου κλειδιού που υπάρχει στο *ScriptPubKey* (*pubkeyHash*) τοποθετείται και αυτό στην κορυφή της στοίβας.
5. Συγκρίνονται οι τιμές των δύο hashes: αυτού που υπολογίστηκε από το δημόσιο κλειδί και αυτού που υπάρχει στο script κλειδώματος. Αν είναι ίσα, βγαίνουν από τη στοίβα.
6. Υπολογίζεται από το δημόσιο κλειδί η υπογραφή του χρήστη και συγκρίνεται, σύμφωνα με το ScriptPubKey (OP_CHECKSIG) με την υπογραφή στη στοίβα. Αν είναι ίδια, βγαίνουν από τη στοίβα και αντικαθίστανται από την τιμή TRUE (δηλαδή το 01), επιβεβαιώνοντας έτσι ότι η συναλλαγή μπορεί να ολοκληρωθεί.

[61] Η διαφορά με τον τρόπο Pay To Pubkey είναι το μικρότερο μέγεθος λόγω της χρήσης του hash αντί για το κλειδί.



Εικόνα 4.17 *Η διαδικασία επαλήθευσης σε μια συναλλαγή τύπου Pay To PubkeyHash.*

Αν μετά την ολοκλήρωση της επαλήθευσης δεν απομείνει η τιμή TRUE, τότε η συναλλαγή δεν θα ξεκλειδώσει, καθώς δεν έχουν παρουσιαστεί οι κατάλληλες πληροφορίες για να ξεκλειδώσει το script. Τέλος, στην πράξη, ο υπολογισμός των 2 scripts γίνεται ξεχωριστά και η μεταφορά των αποτελεσμάτων του σωρού πραγματοποιείται ανάμεσα στις δύο εκτελέσεις.

Αν και στην αρχή ο υπολογισμός αυτός γινόταν μαζί στην ίδια στοίβα, στην πράξη αποδείχθηκε ότι μπορεί κάποιος να αξιοποιήσει τη διαδικασία αυτή και, έτσι, τελικά τρέχουν ξεχωριστά και τα αποτελέσματα μεταφέρονται μαζί για την επαλήθευση.

4.3.3 Ψηφιακές υπογραφές στις συναλλαγές στο δίκτυο του Ethereum

Το Ethereum χρησιμοποιεί και αυτό ψηφιακές υπογραφές στις συναλλαγές του (Υποενότητα 4.2.2). Τα πεδία v, r, s έχουν την πληροφορία που αφορά την υπογραφή καθώς και τις απαραίτητες υποδείξεις για την επαλήθευσή της. Ιδιαίτερα η τιμή της μεταβλητής *v* παίζει σημαντικό ρόλο και έχει ανανεωθεί με την EIP155. Αρχικά έπαιρνε τις τιμές 27 ή 28.

Η επιλογή του ενός από τους 2 αριθμούς χρησιμοποιείται για να βοηθήσει στη διαδικασία επιβεβαίωσης, καθώς, όπως παρουσιάστηκε, για την επαλήθευση η διαδικασία οδηγεί στην εύρεση ενός σημείου πάνω στην ελλειπτική καμπύλη και εστιάζει ιδιαίτερα στη συντεταγμένη x του σημείου αυτού. Όμως στην ελλειπτική καμπύλη (λόγω συμμετρίας ως προς τον άξονα x) είναι 2 σημεία που ικανοποιούν την εξίσωση (R και R΄), τα οποία έχουν αντίθετες τιμές στη συντεταγμένη x, δηλαδή R_x, $-R_x$ αντίστοιχα. Άρτια τιμή του v μεταφράζεται στην επιλογή του ενός σημείου. Αντίστοιχα, η περιττή τιμή προσδιορίζει το άλλο σημείο. Συγκεκριμένα, άρτια τιμή στο v υποδεικνύει το σημείο R, ενώ περιττή τιμή υποδεικνύει το σημείο R΄.

Μετά την εφαρμογή της EIP-155, η τιμή στη μεταβλητή v περιλαμβάνει μια τιμή η οποία ισούται με το διπλάσιο της τιμής προσδιορισμού της αλυσίδας (Chain ID) στην οποία θα μεταδοθεί η συναλλαγή, συν την τιμή 35 ή 36 για την υπόδειξη του σημείου στην καμπύλη, όπως αναλύθηκε πριν.

Ο **Πίνακας 4.5** παρουσιάζει τις τιμές που λαμβάνουν τα Chain IDs των διαφόρων δικτύων στο δίκτυο του Ethereum, και τα οποία χρησιμοποιούνται για να διαμορφώσουν την τιμή του v.

Δίκτυο	Αλυσίδα	Chain ID	ID Δικτύου	Τύπος
Ethereum Mainnet	ETH	1	1	Παραγωγή
Ropsten	ETH	3	3	Τεστ
Rinkeby	ETH	4	4	Τεστ
Goerli	ETH	5	5	Τεστ
Kovan	ETH	42	42	Τεστ
Dev	ETH	2018	2018	Ανάπτυξη
Classic	ETC	61	1	Παραγωγή
Morden	ETC	26	2	Τεστ
Kotti	ETC	6	6	Τεστ
Astor	ETC	212	212	Τεστ

Πίνακας 4.5 *Οι τιμές των Chain IDs για τα δίκτυα που υποστηρίζονται από το Ethereum καθώς και ο τύπος αυτών.*

Στην πράξη, για να υπογραφεί μια συναλλαγή στο Ethereum πρέπει να γίνουν τα εξής βήματα:

- Δημιουργείται η δομή των 9 πεδίων: nonce, gasPrice, gasLimit, to, value, data, chainID, 0,0 και παράγεται η κωδικοποιημένη μορφή της χρησιμοποιώντας τον αλγόριθμο RLP που χρησιμοποιείται στο Ethereum (Υποενότητα 4.2.2).
- Υπολογίζεται το hash της RLP μορφής της συναλλαγής (αποτέλεσμα του προηγούμενου βήματος). Ο αλγόριθμος κωδικοποίησης είναι ο Keccak-256.
- Παράγεται η υπογραφή με τη χρήση του ιδιωτικού κλειδιού του χρήστη στο αποτέλεσμα (hash) του προηγούμενου βήματος.
- Τα στοιχεία r, s, v της υπογραφής προστίθενται στη συναλλαγή στα 3 τελευταία πεδία της.

Κατά την επαλήθευση, σύμφωνα με τους Antonopoulos και Wood (2019), χρησιμοποιούνται οι τιμές των πεδίων r και s (όπως έχει αναλυθεί). Η κατάληξη είναι ο προσδιορισμός 2 σημείων πάνω στην ελλειπτική καμπύλη R και R΄. Ανάλογα με την επιλογή του σημείου, είναι δυνατόν να υπολογιστούν 2 πιθανά δημόσια κλειδιά από τους τύπους (4) και (5):

$$P_1 = (s*R_x - c*G) / r \qquad (4.4)$$

ή

$$P_2 = (s*R'_x - n*G) / r \qquad (4.5)$$

όπου:

- P_1 και P_2 είναι τα δύο υποψήφια δημόσια κλειδιά του χρήστη
- r είναι η τιμή του αντίστοιχου πεδίου της υπογραφής
- s αντίστοιχα η τιμή του αντίστοιχου πεδίου της υπογραφής
- R_x και R'_x είναι οι 2 υποψήφιες τιμές για το προσωρινό δημόσιο κλειδί
- c είναι τα n μικρότερα bits από το hash του μηνύματος
- n είναι η τάξη της ελλειπτικής καμπύλης
- G είναι το σημείο γεννήτορας στην ελλειπτική καμπύλη.

Ανάλογα με την υπόδειξη του σημείου R_x ή R'_x μόνο ένα κλειδί θα χρειαστεί να υπολογιστεί.

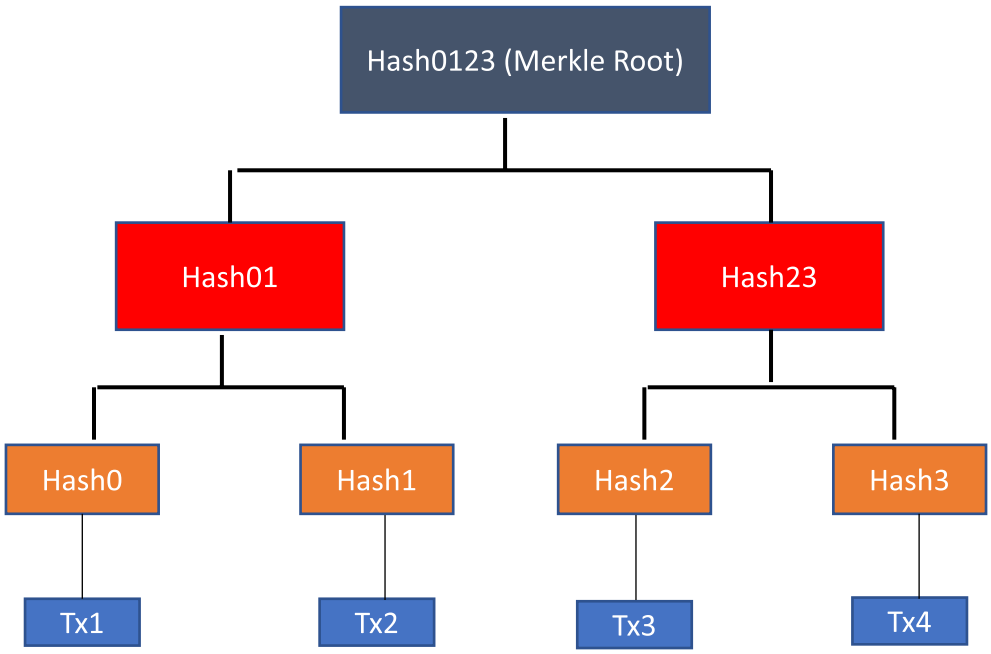
4.4 Χρήση των δένδρων Merkle στις συναλλαγές

Τα δένδρα Merkle είναι μια δομή δεδομένων που παρουσιάστηκε στη διδακτορική διατριβή του Ralph Merkle το 1979 (βλ. Κεφάλαιο 1). Με τη δομή αυτή είναι δυνατόν να εκφραστούν μέσω hash όλες οι συναλλαγές που υπάρχουν μέσα σε ένα block. Έτσι, αν προκύψει μια αλλαγή σε μια συναλλαγή (δηλαδή σε ένα block), τότε, καθώς το παραγόμενο hash του νέου block θα είναι τελείως διαφορετικό, θα είναι δυνατόν να ανιχνευθεί η αλλαγή αυτή.

Πιο αναλυτικά, τα δένδρα Merkle είναι ένα είδος δυαδικού δένδρου το οποίο αποτελείται από κόμβους φύλλα (leaf nodes), ενδιάμεσους κόμβους (non-leaf) και μια κορυφή (ή ρίζα, root). Σε αυτό κάθε (νέος) κόμβος leaf είναι αποτέλεσμα της κωδικοποίησης μεταξύ των 2 κόμβων-παιδιών του. Εξαίρεση αποτελούν οι αρχικοί κόμβοι leaves που δημιουργούνται ως αποτέλεσμα εφαρμογής μιας συνάρτησης κατακερματισμού για να δημιουργηθεί το hash μιας συναλλαγής που συμπεριλαμβάνεται στο block.

Διατρέχοντας ανά ζευγάρι leaves το δένδρο, ξεκινώντας από τους κόμβους leaf και διατρέχοντας το δένδρο από κάτω προς τα πάνω, θα φθάσει κανείς στην κορυφή. Αυτή εκφράζεται με ένα μοναδικό hash, το οποίο έχει προέλθει από τα hashes όλων των προηγούμενων ζευγών από κόμβους και, έτσι, θεωρείται ότι εκφράζει τις συναλλαγές αυτές.

Σαν παράδειγμα χρησιμοποιείται το δένδρο που φαίνεται στην **Εικόνα 4.18**, το οποίο αποτελείται από 4 αρχικές συναλλαγές. Τα hashes αυτών αποτελούν τους αρχικούς κόμβους leaves του δένδρου Merkle.



Εικόνα 4.18 *Η δομή ενός δένδρου Merkle.*

Στους αρχικούς κόμβους (leaf nodes) θα περάσουν τα hashes των συναλλαγών, όπως φαίνεται στο κάτω μέρος της Εικόνας 4.18. Δηλαδή, Hash0 = Hash(Tx1)[62], Hash1 = Hash(Tx2) κ.ο.κ. Στη συνέχεια, τα δύο ζευγάρια από κόμβους leaves, δηλαδή τα Hash0 – Hash1 και Hash2 – Hash3, θα συνδυαστούν για να φτιάξουν τους κόμβους του (ακριβώς) ανώτερου επιπέδου, δηλαδή τους κόμβους Hash01 και Hash23. Για την ακρίβεια, ισχύει ότι Hash01 = Hash(Hash0+Hash1) και αντίστοιχα για το Hash23 = Hash(Hash2+Hash3).

Τέλος, από τον συνδυασμό των 2 αυτών κόμβων και την είσοδό τους στη συνάρτηση κατακερματισμού, το hash, που θα είναι το αποτέλεσμα θα αποτελέσει το περιεχόμενο της ρίζας του δένδρου [Hash0123 = Hash(Hash01+ Hash23)]. Το αποτέλεσμα αυτό είναι γνωστό ως *Merkle Root* και αποτελεί πληροφορία η οποία διαμοιράζεται στην επικεφαλίδα των blocks του Bitcoin και του Ethereum.

Ιδιαίτερα στο Ethereum, η εφαρμογή του εργαλείου αυτού είναι πιο έντονη, καθώς στην επικεφαλίδα του κάθε block περιέχονται τα hashes από 3 διαφορετικά δένδρα. Ο λόγος που συμβαίνει αυτό σχετίζεται με τη διαφορά ανάμεσα σε αυτό και το Bitcoin, καθώς το Ethereum αποθηκεύει την κατάσταση του κόσμου πέρα από τις συναλλαγές που πραγματοποιούνται. Έτσι, λοιπόν, τα 3 αυτά δένδρα έχουν πληροφορίες σχετικά με: τις *συναλλαγές*, την *κατάσταση* του κόσμου και *αποδείξεις* που είναι, ουσιαστικά, δεδομένα που δείχνουν την επίδραση της κάθε συναλλαγής.

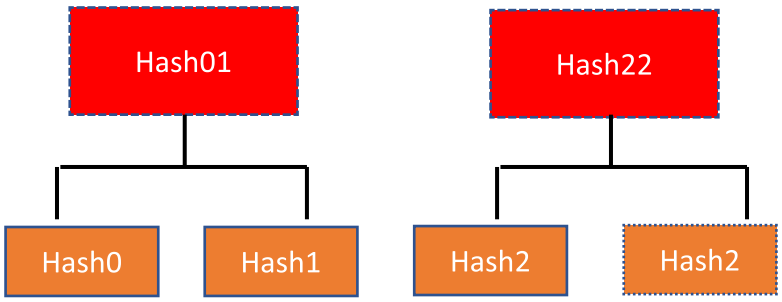
Επιπλέον, στο Ethereum χρησιμοποιείται μια πιο εκλεπτυσμένη μορφή των Merkle Trees, η οποία ονομάζεται *Merkle Patricia Tree*. Τα δένδρα αυτά ενδείκνυνται για να εκφράσουν πληροφορία καταστάσεων

[62] Στο Bitcoin ο υπολογισμός του hash βρίσκεται με την εφαρμογή δύο φορές του αλγορίθμου SHA256.

καλύτερα από τη δυαδική μορφή που παρουσιάστηκε πριν και η οποία χρησιμοποιείται στο Bitcoin. Ο λόγος είναι ότι τα Merkle Patricia Trees διαχειρίζονται καλύτερα τον χάρτη κλειδιών-τιμών που δημιουργείται για την παρακολούθηση της κατάστασης του κόσμου στο Ethereum, ενώ τα δυαδικά Merkle Trees διαχειρίζονται αποτελεσματικά τα hashes των δεδομένων των συναλλαγών (Buterin, 2015).

Στην Εικόνα 4.18 παρουσιάστηκε ένα παράδειγμα που αποτελείται από άρτιο αριθμό συναλλαγών. Στην πραγματικότητα μπορεί ο αριθμός των συναλλαγών που περιέχονται σε ένα block, εκτός από πολύ μεγαλύτερος, να είναι και περιττός. Στην περίπτωση αυτή, η τελευταία συναλλαγή, που μένει χωρίς ζευγάρι, δημιουργεί ένα αντίγραφό της για να συμπληρωθεί το δυαδικό δένδρο και δημιουργεί ζευγάρι μαζί του.

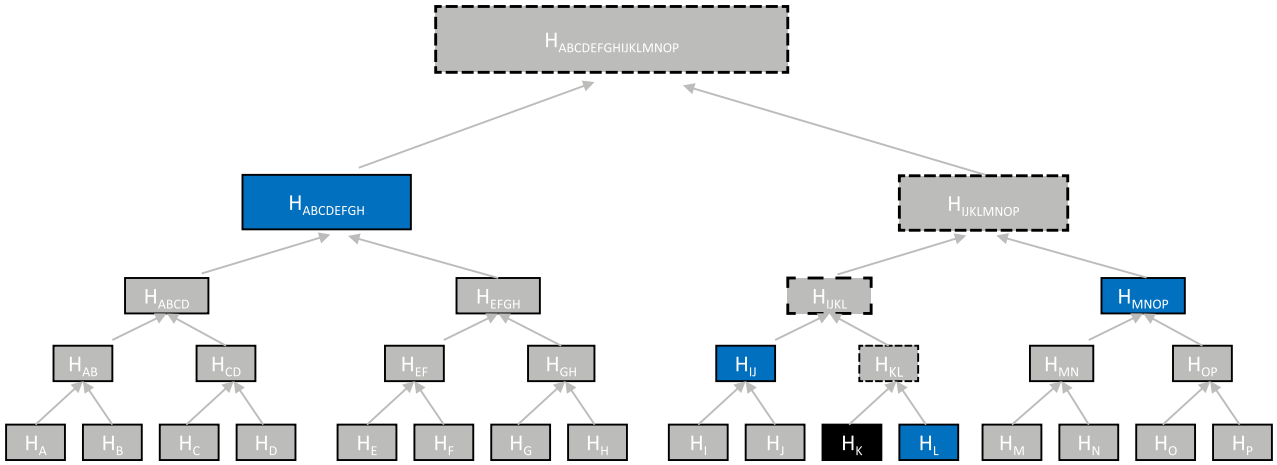
Αυτό φαίνεται στην **Εικόνα 4.19**, που δείχνει πώς δημιουργούνται τα ζευγάρια στους κόμβους leaf στην περίπτωση περιττού αριθμού συναλλαγών. Ακολούθως, ο τρόπος υπολογισμού του Root είναι ο ίδιος με αυτόν που παρουσιάστηκε στην Εικόνα 4.18.



Εικόνα 4.19 *Η δομή ενός δένδρου Merkle στην περίπτωση περιττού αριθμού συναλλαγών.*

Ένα από τα σημαντικά πλεονεκτήματα της χρήσης των Merkle Trees στις συναλλαγές είναι ότι απλουστεύει σημαντικά τη διαδικασία αναζήτησης και επιβεβαίωσης συμμετοχής μιας συναλλαγής μέσα σε ένα block. Για την ακρίβεια, όταν χρειαστεί να αναζητηθεί η παρουσία μιας συναλλαγής σε ένα block, δεν χρειάζεται να ανατρέξει κανείς σε όλες τις συναλλαγές αυτού. Αντιθέτως, αρκεί να γνωρίζει συγκεκριμένες συναλλαγές, οι οποίες και δημιουργούν ένα *μονοπάτι μέσα στο block*. Το μονοπάτι αυτό, που ονομάζεται και *Merkle path*, ξεκινά από τη συναλλαγή που πρέπει να επιβεβαιωθεί και καταλήγει στη ρίζα του block. Η τιμή της ρίζας, που υπολογίζεται, θα πρέπει να είναι ίση με την τιμή του Merkle Root στην επικεφαλίδα του block. Εφόσον αυτό ισχύει, τότε επιβεβαιώνεται η συμμετοχή της συναλλαγής στο block αυτό.

Στην **Εικόνα 4.20** παρουσιάζεται ένα παράδειγμα επιβεβαίωσης της συμμετοχής της συναλλαγής H_K σε ένα (συγκεκριμένο) block. Για την επιβεβαίωση θα πρέπει να χρησιμοποιηθεί το μονοπάτι που δίνεται από τις συναλλαγές που φαίνονται με μπλε χρώμα στην Εικόνα 4.20, δηλαδή τις: H_L, H_{IJ}, H_{MNOP}, $H_{ABCDEFGH}$. Η γνώση του μονοπατιού αυτού επιτρέπει τον υπολογισμό των hashes που φαίνονται με διακεκομμένες γραμμές (H_{KL}, H_{IJKL}, $H_{IJKLMNO}$) καθώς και, τελικά, τον υπολογισμό του Root για τη σύγκριση.



Εικόνα 4.20 *Χρήση ενός μονοπατιού για την επιβεβαίωση ύπαρξης μιας συναλλαγής μέσα σε ένα block.*

Η εύρεση του Merkle path το οποίο χρειάζεται για την επιβεβαίωση της συμμετοχής μιας συναλλαγής σε ένα block αναλαμβάνεται, συνήθως, από τις εφαρμογές για πορτοφόλι του χρήστη. Αυτές, είτε είναι δυνατόν να προχωρήσουν στην αποθήκευση του path (μαζί με τη συναλλαγή) είτε μπορούν να ρωτήσουν για αυτό και να μάθουν την πληροφορία από έναν κόμβο που έχει ολόκληρο το ledger.

Τέλος, έχει ενδιαφέρον να συζητηθεί γιατί επιλέχθηκε να ακολουθηθεί η δομή του δυαδικού δένδρου και όχι ο συνδυασμός όλων των hashes από τις συναλλαγές σε μια μεγάλη είσοδο, το hash της οποίας μπορεί να αποτελέσει το Merkle root, μιας και αυτό ενδιαφέρει πρακτικά. Ο λόγος σχετίζεται με τη διαδικασία επιβεβαίωσης που αναπτύχθηκε προηγουμένως. Η δυνατότητα παρακολούθησης μέσω ενός μονοπατιού (Merkle path) για την επιβεβαίωση της συμμετοχής μιας συναλλαγής σε ένα block είναι πολύ σημαντική και εύκολη στην υλοποίηση. Αναδεικνύει, μάλιστα, σημαντικά τη χρήση των Merkle Trees σε μια λύση blockchain. Από την άλλη πλευρά, αν γινόταν χρήση ενός μεγάλου, συνολικού hash θα έπρεπε κάθε φορά να χρησιμοποιηθεί το σύνολο όλων των συναλλαγών από τις οποίες αποτελείται το hash, για την επιβεβαίωση.

Ο **Πίνακας 4.6** δείχνει την αποτελεσματικότητα της εφαρμογής των Merkle Trees, όσον αφορά τον αριθμό των hashes που πρέπει να γνωρίζει κανείς για να επιβεβαιώσει την παρουσία μιας συναλλαγής σε ένα block. Έτσι, φαίνεται ότι για 65.535 συναλλαγές χρειάζονται μόλις 16 hashes για την επιβεβαίωση.

Αριθμός Συναλλαγών	**Μέγεθος block (εκτίμηση)**	**Μέγεθος path (βάσει αριθμού hashes)**	**Μέγεθος path (bytes)**
16 συναλλαγές	4 kB	4 hashes	128
512 συναλλαγές	128 kB	9	288
2.048 συναλλαγές	512 kB	11	352
65.535 συναλλαγές	16 MB	16	512

Πίνακας 4.6 *Η αποτελεσματικότητα της χρήσης δένδρων Merkle για τις συναλλαγές σε λύσεις blockchain.*

Συνοψίζοντας, με τα δένδρα Merkle επιτυγχάνεται να:

- Εξασφαλιστεί η ακεραιότητα των συναλλαγών σε ένα δίκτυο blockchain με τον υπολογισμό του Merkle Root.
- Μειωθεί ο χρόνος αναζήτησης καθώς και ο αριθμός από μονοπάτια που πρέπει να ακολουθήσει κάποιος για να επιβεβαιώσει τη συμμετοχή μιας συναλλαγής σε ένα block. Μάλιστα, το μέγεθος των μονοπατιών είναι σημαντικά μειωμένο, όπως και ο αριθμός πράξεων (υπολογισμούς hashes), όπως φαίνεται και στον Πίνακα 4.6.
- Να μεταδοθεί μικρός αριθμός δεδομένων ως πληροφορία για την εύρεση του μονοπατιού σε ένα block.

Βιβλιογραφία

Antonopoulos, A. M. (2017). *Mastering Bitcoin. Programming the Open Blockchain* (2nd ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., & Wood, G. (2019). *Mastering Ethereum* (1st ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., Osuntokun, O., & Pickhardt, R. (2021). *Mastering the Lightning Network. A second Layer Blockchain Protocol for Instant Bitcoin Payments* (1st ed.). O'Reilly Media, Inc.

Buterin, V. (2015). Merkling in Ethereum, *Ethereum Foundation Blog*. Online πηγή: https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Buterin, V. (2014). *Ethereum Improvement Proposals*. 2014. Online πηγή: https://eips.ethereum.org/EIPS/eip-155 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Pornin, T. (2013). (RFC 6979). Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA). *Request for Comments: 6979*, IETF, 2013. Online πηγή: https://datatracker.ietf.org/doc/html/rfc6979 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 5
Συναίνεση σε Κατανεμημένα Δίκτυα

Σύνοψη

Το Κεφάλαιο αυτό επικεντρώνεται στο θέμα της επίτευξης συναίνεσης σε ένα κατανεμημένο δίκτυο, στην ανάδειξη της σημασίας αυτής σε τέτοιου είδους περιβάλλοντα και επισημαίνονται οι διαφορές που παρουσιάζονται σε σχέση με κεντρικοποιημένες λύσεις. Επιπρόσθετα, εισάγεται το Πρόβλημα των Βυζαντινών Στρατηγών (Byzantine Generals Problem) και εξηγείται ο τρόπος με τον οποίο οι τεχνικές συναίνεσης στο περιβάλλον ενός δικτύου blockchain μπορούν να δώσουν μια λύση.

Τέλος, αναφέρονται γνωστές τεχνικές (π.χ. PoW, PoS) που έχουν χρησιμοποιηθεί για την επίτευξη της συναίνεσης στα πιο δημοφιλή δίκτυα blockchain.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1 και 2.

5.1 Κατανεμημένα δίκτυα και συναίνεση

Η συναίνεση παίζει πολύ σημαντικό ρόλο στα κατανεμημένα δίκτυα, καθώς απουσιάζει σε αυτά μια έμπιστη οντότητα η οποία θα μπορούσε να βοηθήσει στην επιβολή της συναίνεσης, όπως συμβαίνει σε κεντρικοποιημένα κυρίως περιβάλλοντα. Στις κατανεμημένες λύσεις, όπως είναι το blockchain, είναι οι ίδιοι οι κόμβοι που καλούνται να αποφασίσουν για την ορθότητα και την εγκυρότητα κάθε block, βασιζόμενοι στη δική τους κρίση και μόνον.

Δεδομένης της συχνής έλλειψης εμπιστοσύνης μεταξύ των μερών που συμμετέχουν σε ένα δίκτυο blockchain, η ανάγκη συνεννόησής τους για την ορθή λειτουργία του δικτύου επιβάλλει την αποδοχή και την εφαρμογή κοινών και προσυμφωνημένων κανόνων. Επιπλέον, για την καλύτερη απόδοση του δικτύου, χρειάζεται η συμμετοχή του μεγαλύτερου δυνατού αριθμού κόμβων στη διαδικασία της συναίνεσης. Συχνά, μάλιστα, σε ανοικτά (ή δημόσια) δίκτυα έχει προτιμηθεί και η απόδοση κινήτρων για την ενίσχυση της εποικοδομητικής συμμετοχής των κόμβων στη διαδικασία της συναίνεσης.

Η δήλωση αυτών των κοινών κανόνων, καθώς και η τεχνική με την οποία πραγματοποιείται η εφαρμογή τους σε ένα δίκτυο blockchain αποτελούν χαρακτηριστικά της σχεδίασης μιας λύσης. Πρόκειται για την επιλογή του *αλγόριθμου συναίνεσης* ο οποίος και θα εφαρμοστεί στο δίκτυο.

Γνωστές λύσεις αλγόριθμων συναίνεσης αποτελούν το *Proof of Work* (*PoW*) και το *Proof of Stake* (*PoS*). Το PoW αποτελεί την επιλογή στα δίκτυα του Bitcoin και του Ethereum, ενώ το PoS αποτελεί επιλογή στη νέα έκδοση του Ethereum (v 2.0), που αναμένεται να ξεκινήσει τον Σεπτέμβριο του 2022.

Παράλληλα με αυτές τις δύο τεχνικές, ο χώρος του σχεδιασμού ενός αλγόριθμου κατανεμημένης συναίνεσης αποτελεί έναν χώρο μεγάλου επιστημονικού ενδιαφέροντος, που και αναθερμάνθηκε με την επιτυχία και τη διάδοση της τεχνολογίας του blockchain. Σημείο αναφοράς για την επιστημονική κοινότητα είναι η εργασία των Lamport et al. (1982). Σε αυτήν αναδεικνύεται το πρόβλημα της συναίνεσης σε κατανεμημένα περιβάλλοντα και παρομοιάζεται αυτό με την ανάγκη ενός υπολογιστικού συστήματος να συνεχίσει την ορθή λειτουργία του παρά τις πιθανές βλάβες σε στοιχεία του. Μάλιστα, με τη βοήθεια ενός παραδείγματος, όπως αυτού της ανάγκης επικοινωνίας μεταξύ ορισμένων στρατηγών του Βυζαντίου για την κατάληψη μιας εχθρικής πόλης, οι συγγραφείς κατάφεραν να αναδείξουν το πρόβλημα ως ένα πρόβλημα της Θεωρίας Παιγνίων[63]. Παρουσίασαν, επίσης, και ορισμένες χρήσιμες λύσεις σε αυτό.

Το *Πρόβλημα των Βυζαντινών Στρατηγών* αναλύεται στην επόμενη ενότητα. Εκεί εξηγούνται και οι λόγοι που αυτό συνδέεται με την αντιμετώπιση προβλημάτων από ένα υπολογιστικό σύστημα. Επίσης, εξηγείται το πώς έδωσε λύση στο πρόβλημα αυτό η τεχνολογία του blockchain, ενώ θα γίνει αναφορά στις πιο διαδεδομένες τεχνικές συναίνεσης στον κόσμο του blockchain.

[63] Online Σύνδεσμος: εδώ

5.1.1 Το Πρόβλημα των Βυζαντινών Στρατηγών

Η συναίνεση σε ένα κατανεμημένο περιβάλλον είναι μια σύνθετη διαδικασία που πρέπει να είναι αποδοτική, οδηγώντας το δίκτυο σε συμφωνία. Στον στόχο αυτόν, η διαδικασία της συναίνεσης έχει να αντιμετωπίσει, εκτός από την έλλειψη μιας έμπιστης οντότητας, τις πιθανές βλάβες των κόμβων του δικτύου και τις πιθανές κακόβουλες[64] ενέργειες κόμβων αυτού.

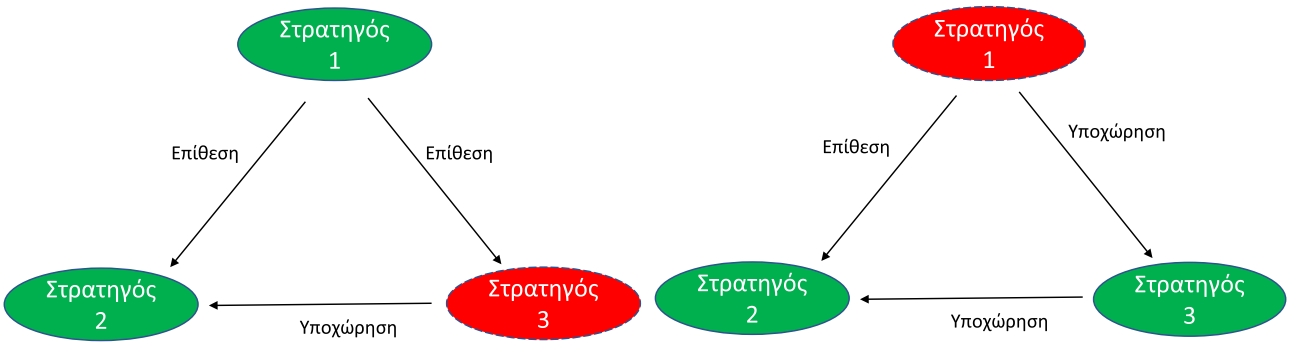
Σύμφωνα με τους Lamport et al. (1982), ένα υπολογιστικό σύστημα πρέπει να μπορεί να συνεχίσει την ομαλή του λειτουργία ακόμα και αν κάποιο από τα τμήματά του σταματήσει να λειτουργεί, είτε λόγω τεχνικού προβλήματος είτε λόγω αποστολής λανθασμένων πληροφοριών που οδηγεί σε μειωμένη απόδοση. Δημιουργείται, δηλαδή, μια αντιστοιχία ανάμεσα στις ανάγκες για κατανεμημένη συναίνεση με την εύρυθμη λειτουργία ενός υπολογιστικού συστήματος ικανού να αντιμετωπίσει προβλήματα που επηρεάζουν την απόδοσή του. Με τον τρόπο αυτόν η όποια πρόταση μπορεί να βοηθήσει στην επίλυση προβλημάτων της μιας μορφής θα μπορεί να είναι αποτελεσματική και στην αντιμετώπιση των θεμάτων της άλλης.

Μάλιστα, για την αντιστοιχία των δύο αυτών καταστάσεων παρουσιάστηκε ένα σενάριο για την καλύτερη απεικόνισή τους. Έτσι, στο σενάριο αυτό τμήματα του στρατού του Βυζαντίου, το καθένα με τον δικό του στρατηγό (από εκεί προέκυψε και το όνομα της δημοσίευσης που έχει τίτλο «Το Πρόβλημα των Βυζαντινών Στρατηγών»), έχουν περικυκλώσει μια πόλη με σκοπό να την καταλάβουν.

Οι στρατηγοί θα χρειαστεί να επικοινωνήσουν, μέσω μηνυμάτων (γραπτών ή προφορικών), για να συνεννοηθούν για το πλάνο της επίθεσης. Μάλιστα, η συμφωνία πρέπει να επέλθει ως αποτέλεσμα της επικοινωνίας αυτής, ακόμα και στην περίπτωση που τα μηνύματα χαθούν ή αλλοιωθούν είτε από εχθρούς είτε από στρατηγούς-προδότες που θέλουν να σαμποτάρουν την επιχείρηση. Για να γίνει αυτό δυνατό, στο σενάριο των Βυζαντινών Στρατηγών, όπου δεν υπάρχει κάποιο ασφαλές κανάλι επικοινωνίας, θα πρέπει να χρησιμοποιηθεί κάποιος αλγόριθμος.

Μάλιστα, αν και φαινομενικά το πρόβλημα δείχνει απλοϊκό, αποδείχθηκε ότι δεν μπορεί να επιτευχθεί συναίνεση εφόσον τα μηνύματα μεταφέρονται προφορικά (οπότε και είναι στην ευχέρεια του αγγελιοφόρου να μεταφερθεί σωστά το μήνυμα) στην περίπτωση των τριών στρατηγών (κόμβων), με τον έναν από αυτούς να έχει τον ρόλο του προδότη.

Στην **Εικόνα 5.1** φαίνονται δύο διαφορετικά σενάρια, που αναδεικνύουν το πρόβλημα στην περίπτωση των τριών στρατηγών με έναν κακόβουλο συμμετέχοντα. Τα σενάρια αλλάζουν τον πιθανό κακόβουλο χρήστη για να επιβεβαιώσουν τον τρόπο που αυτός επηρεάζει την απόφαση του βυζαντινού στρατού (δηλαδή του δικτύου).



Εικόνα 5.1 *Η περίπτωση των τριών στρατηγών με έναν κακόβουλο χρήστη, όπως φαίνεται στο Lamport et al., 1982. Ο Στρατηγός 1 δίνει την εντολή, με προφορικό μήνυμα, αν θα επιτεθούν ή θα υποχωρήσουν και οι άλλοι δύο στρατηγοί πρέπει να διαδώσουν και να ακολουθήσουν την εντολή. Με κόκκινο συμβολίζεται ο κακόβουλος στρατηγός.*

Γενικεύοντας, στην περίπτωση των προφορικών μηνυμάτων, αποδείχθηκε ότι για την αποτελεσματική αντιμετώπιση *m* κακόβουλων στρατηγών (κόμβων) θα πρέπει να υπάρχουν *τουλάχιστον 3*m+1* στρατηγοί στο δίκτυο (Lamport et al., 1982). Τότε, η απόφαση που θα πάρουν οι στρατηγοί θα είναι η ορθή, ανεξάρτητα από την προσπάθεια των κακόβουλων χρηστών να την αλλοιώσουν.

Λύση στο Πρόβλημα των 3 Στρατηγών (κόμβων), και όχι μόνο, μπορεί να υπάρξει και στην περίπτωση όπου χρησιμοποιούνται γραπτά μηνύματα, τα οποία και είναι υπογεγραμμένα από τον εκάστοτε στρατηγό.

[64] Ως κακόβουλες θεωρούνται οι ενέργειες που γίνονται προς όφελος κάποιων χρηστών και όχι του δικτύου. Για παράδειγμα, η επίτευξη double spending από έναν κόμβο θεωρείται κακόβουλη ενέργεια.

Μάλιστα, αναφέρεται ότι μια λύση στο πρόβλημα με τη συμμετοχή 3 στρατηγών μπορεί να βρεθεί εφόσον οι υπογραφές των στρατηγών που λένε αλήθεια δεν μπορούν να πλαστογραφηθούν (δεν γίνεται καμία αναφορά για τις υπογραφές των κακόβουλων στρατηγών) και εφόσον είναι δυνατή η ανίχνευση της όποιας τροποποίησης στο περιεχόμενο του μηνύματος. Στην ουσία, το πρόβλημα λύνεται αν μπορεί να αποδειχθεί ποιος είναι αυτός που το έστειλε και πώς μπορεί να αποδείξει ότι αυτό που λέει είναι σωστό (ή δεν έχει αλλοιωθεί).

Το Πρόβλημα των Βυζαντινών Στρατηγών αποτελεί ένα πρόβλημα της Θεωρίας των Παιγνίων, το οποίο αναδεικνύει τις δυσκολίες της συναίνεσης σε ένα κατανεμημένο δίκτυο. Με τις δυσκολίες αυτές να περιλαμβάνουν, σε σημαντικό βαθμό, την εμπλοκή κακόβουλων χρηστών (με τη μορφή των στρατηγών στο σενάριο), οι οποίοι προσπαθούν να εκμεταλλευτούν την κατάσταση προς όφελός τους.

Συχνά, οι δυσκολίες αυτές ονομάζονται *Βυζαντινά σφάλματα* (*Byzantine Faults*) και οι αλγόριθμοι συναίνεσης που θα εφαρμοστούν στα δίκτυα αυτά θα πρέπει να επιτυγχάνουν να λαμβάνουν τις σωστές αποφάσεις και να λειτουργούν αποδοτικά, παρά την πιθανότητα ύπαρξης τέτοιων σφαλμάτων. Πρέπει, δηλαδή, να αποκτήσουν *ανοχή στα Βυζαντινά σφάλματα* (*Byzantine fault tolerance*).

Το ίδιο όμως συμβαίνει και σε ένα υπολογιστικό σύστημα, το οποίο και πρέπει να μπορεί να αντεπεξέρχεται σε προβλήματα (λογισμικού ή υλικού) τα οποία μπορεί να παρουσιαστούν τυχαία και δεν είναι δυνατή η πρότερη πρόβλεψη για τον τρόπο που θα επηρεάσουν το σύστημα. Προφανώς, αυτού του είδους τα σφάλματα είναι τα πιο δύσκολα στη διαχείριση και στη διόρθωση.

Συμπληρωματικά, παραδείγματα συστημάτων τα οποία πρέπει να λάβουν αποφάσεις και να αντιμετωπίσουν επιδεικνύοντας ανοχή σε πιθανά Βυζαντινά σφάλματα αποτελούν: η διαχείριση των πυρηνικών εγκαταστάσεων και η λειτουργία των μηχανών αεροπλοΐας.

Τέλος, οι κεντρικοποιημένες λύσεις *δεν λύνουν το πρόβλημα των Βυζαντινών Στρατηγών*. Με τη χρήση της κεντρικής οντότητας επέρχεται η συναίνεση και δεν χρειάζεται ο κάθε κόμβος να λάβει από μόνος του αποφάσεις. Φυσικά, οι λύσεις αυτές έχουν τα δικά τους μειονεκτήματα, καθώς κινδυνεύουν από πιθανή διαφθορά της κεντρικής οντότητας, η οποία θα επηρεάσει την απόδοση όλου του συστήματος.

5.1.2 Blockchain: Λύνοντας το Πρόβλημα των Βυζαντινών Στρατηγών

Η λύση του Προβλήματος των Βυζαντινών Στρατηγών από τη φύση της δεν μπορεί να είναι ντετερμινιστική αλλά είναι περισσότερο πιθανοτική, λόγω της έλλειψης απευθείας επικοινωνίας μεταξύ των κόμβων ενός κατανεμημένου δικτύου και της αδυναμίας παροχής εγγυήσεων.[65]

Στο πλαίσιο αυτό, η λύση θα πρέπει να κάνει χρήση μηχανισμών οι οποίοι επιτυγχάνουν ανοχή στα Βυζαντινά σφάλματα και θα πρέπει να επιβάλει συγκεκριμένους κανόνες στην υλοποίησή της. Επιπλέον, καθώς ο κάθε κόμβος είναι μόνος του και επικοινωνεί μέσω ενός μη ασφαλούς καναλιού με τους ομότιμούς του, είναι ιδιαίτερα κρίσιμο να δημιουργηθεί ένα αξιόπιστο κανάλι επικοινωνίας, παρά την αβεβαιότητα που υπάρχει τριγύρω.

Το blockchain επιτυγχάνει τη δημιουργία ενός ασφαλούς καναλιού με τη δημιουργία του ledger αποτελούμενου από blocks από συναλλαγές. Ταυτόχρονα, η χρήση της κρυπτογραφίας:

α) με την απόδοση κλειδιών και διευθύνσεων στους χρήστες, τα οποία μπορούν να χρησιμοποιηθούν για να επιβεβαιώσουν τον αποστολέα μιας συναλλαγής, και

β) με τη χρήση της στα blocks των συναλλαγών για τη δημιουργία της αλυσίδας αλλά και της εξασφάλισης ότι κάποιο block δεν μπορεί να αλλοιωθεί (μέσω των Merkle Trees),

επιτρέπει την επίτευξη ενός αισθήματος εμπιστοσύνης και αποτελεί τη βάση για να συμφωνήσουν οι κόμβοι στην επιλογή του νέου block. Τέλος, η χρήση του κοινού, κατανεμημένου ledger που επιτρέπει σε όλους τους κόμβους να κρατούν την πληροφορία ενδυναμώνει το δίκτυο, δυσκολεύοντας τη χειραγώγησή του.

Με τον τρόπο αυτόν μπορεί το δίκτυο να δώσει λύσεις σε προβλήματα όπως η αλλαγή των δεδομένων ενός μηνύματος (συναλλαγής) ή η αντιμετώπιση των διπλών συναλλαγών (βλ. Κεφάλαιο 1 για ορισμό), προβλήματα

[65] Η πιθανοτική λύση αφορά τον κάθε κόμβο και όχι το σύστημα. Δηλαδή δεν είναι σίγουρο ότι το πρόβλημα θα λυθεί από κάθε κόμβο (π.χ. miner). Είναι όμως σίγουρο ότι ένας κόμβος θα το λύσει και, εφόσον το αποδείξει, τότε θα επέλθει συμφωνία και συναίνεση στο δίκτυο. Για παράδειγμα, στο PoW ένας miner θα βρει το νέο block και θα το προτείνει. Δεν θα προτείνει ο κάθε κόμβος ένα νέο block. Έτσι, ο κάθε miner έχει τις πιθανότητές του (για αυτό και πιθανοτική λύση), αλλά το δίκτυο συνολικά θα καταφέρει να έρθει σε συναίνεση, παρακινώντας τον κάθε κόμβο να συμμετάσχει στη διαδικασία του mining.

αρκετά σημαντικά, ιδιαίτερα για την εφαρμογή λύσεων που περιλαμβάνουν οικονομικές συναλλαγές.

Επιπλέον, τα χαρακτηριστικά (α) και (β) αντιμετωπίζουν τις δύο συνθήκες που παρουσιάστηκαν ως ικανές για την επίλυση του Προβλήματος των Βυζαντινών Στρατηγών στην περίπτωση των γραπτών μηνυμάτων (βλ. Υποενότητα 5.1.1).

Στο δίκτυο του blockchain οι κόμβοι είναι οι στρατηγοί και ο αλγόριθμος συναίνεσης είναι οι κανόνες που ακολουθούν για τη λύση του Προβλήματος των Βυζαντινών Στρατηγών.

Πιο αναλυτικά, σύμφωνα με τον Αντωνόπουλο (2017), στο δίκτυο του Bitcoin τέσσερις διεργασίες είναι εκείνες που συμμετέχουν ανεξάρτητα στον κάθε κόμβο και εργάζονται για την επίτευξη της συναίνεσης στο δίκτυο.

Οι τέσσερις αυτές διεργασίες είναι:

1. Η επαλήθευση για την ορθότητα της κάθε συναλλαγής, σύμφωνα με τους κανόνες που ισχύουν στο δίκτυο, από κάθε πλήρη κόμβο.
2. Η ανεξάρτητη επιλογή των συναλλαγών και η τοποθέτησή τους για τη δημιουργία του νέου block από κάθε κόμβο miner. Το block που θα επιλεγεί θα πρέπει να έχει και απόδειξη για την εργασία του (και για την επίτευξη του βαθμού δυσκολίας που έχει καθορίσει το δίκτυο) μέσω της εκτέλεσης του αλγόριθμου συναίνεσης, του PoW.
3. Η ανεξάρτητη επαλήθευση για την ορθότητα του κάθε νέου block από τους κόμβους miners, οι οποίοι και είναι υπεύθυνοι για να διατηρούν και να ανανεώνουν την αλυσίδα των blocks.
4. Η επιλογή από τον κάθε κόμβο εκείνης της αλυσίδας που έχει να δείξει τη μεγαλύτερη επεξεργαστική επιτυχία μέσω του αλγόριθμου του PoW.

Τα παραπάνω έχουν αναλυθεί στο Κεφάλαιο 2 και μπορούν εύκολα να γενικευτούν τόσο για το Ethereum όσο και για άλλα δίκτυα blockchain, αντικαθιστώντας τη συμμετοχή του PoW με τον όποιο αλγόριθμο συναίνεσης (π.χ. PoS ή Delegated PoS) έχει επιλεγεί στην κάθε περίπτωση.

Γίνεται αντιληπτό, επομένως, ότι οι αλγόριθμοι συναίνεσης είναι αυτοί που βοηθούν στην επίτευξη συμφωνίας για το επόμενο block και ενεργοποιούν τους μηχανισμούς ασφαλείας του δικτύου του blockchain για την προστασία του, με την είσοδο του block στην αλυσίδα και την ενημέρωση των υπόλοιπων κόμβων του δικτύου.

Ο αλγόριθμος αυτός, επιπλέον, εισάγει την πιθανότητα για την εύρεση της λύσης από τον κάθε κόμβο, επιτρέποντας έτσι σε όλους να δοκιμάσουν να βρουν τη λύση και να συνεισφέρουν στην επίτευξη της συναίνεσης στο δίκτυο.

Στη συνέχεια αναλύονται οι βασικές τεχνικές συναίνεσης, με έμφαση στη λογική που χρησιμοποιούν για να εξασφαλίσουν τη συναίνεση στο κατανεμημένο περιβάλλον ενός δικτύου blockchain.

5.2 Τεχνικές συναίνεσης σε ένα δίκτυο blockchain

Μέχρι τώρα τονίστηκε η σημασία της εφαρμογής μιας αποδοτικής λύσης για την επίτευξη της συναίνεσης σε ένα κατανεμημένο δίκτυο blockchain. Η συναίνεση σε ένα δίκτυο blockchain έχει στη βάση της τις εξής αρχές: το περιεχόμενο των blocks να είναι *αμετάβλητο* (δεν υπάρχει αλλαγή στο παρελθόν) και *οριστικό* (δεν θα υπάρξει αλλαγή στο μέλλον).

Με αυτά δεδομένα, στη συνέχεια θα μελετηθεί ο τρόπος που επιτυγχάνεται αυτή από τις δύο βασικές τεχνικές που χρησιμοποιούνται σήμερα: από το PoW και το PoS.

5.2.1 Proof of Work (PoW)

Σε συνέχεια όσων έχουν παρουσιαστεί στο Κεφάλαιο 2, το PoW αποτελεί έναν από τους πιο σημαντικούς και πιο διαδεδομένους αλγόριθμους συναίνεσης, με εφαρμογή σε δημοφιλείς υλοποιήσεις blockchain, όπως το Bitcoin και το Ethereum (v 1.0).

Το PoW χρησιμοποιεί τη διαδικασία της «εξόρυξης» (mining) όταν έχουν επιβεβαιωθεί οι συναλλαγές που έχουν επιλεγεί για να μπουν στο νέο block. Στη διαδικασία του mining συμμετέχουν όλοι οι κόμβοι που έχουν αποθηκευμένο όλο το ledger και μπορούν να εκτελέσουν ενέργειες mining (βλ. Κεφάλαιο 2).

Η διαδικασία του mining είναι στην ουσία ένας αγώνας ταχύτητας. Νικητής στον αγώνα είναι αυτός ο κόμβος (miner) ο οποίος θα καταφέρει να λύσει τον δύσκολο επεξεργαστικό γρίφο του PoW, δημιουργώντας ένα (υποψήφιο) νέο block.

Το mining περιλαμβάνει τα εξής τρία βασικά βήματα, τα οποία φαίνονται και συνοπτικά στην **Εικόνα 5.2**:

1. Συνεχής κατακερματισμός της κεφαλίδας του block για την εύρεση λύσης στο PoW.

2. Σε κάθε επανάληψη του κατακερματισμού (βήμα 1) το hash μεταβάλλεται (τουλάχιστον κατά 1 bit). Η αλλαγή γίνεται στο πεδίο Nonce του block, το οποίο και έχει προστεθεί ακριβώς για την εξυπηρέτηση του PoW.
3. Κάθε φορά, η τιμή που προκύπτει συγκρίνεται με τον βαθμό δυσκολίας του δικτύου για την εγκυρότητά της. Δηλαδή ελέγχεται πότε το παραγόμενο αποτέλεσμα είναι ένας αριθμός ο οποίος είναι μικρότερος από έναν στόχο.

Εικόνα 5.2 *Τα βήματα για την εύρεση λύση στο PoW.*

Εξηγώντας πιο αναλυτικά τα παραπάνω βήματα, η διαδικασία ξεκινά με την επιβεβαίωση των συναλλαγών και με την επιλογή τους για προσθήκη στο block. Κατόπιν υπολογίζεται το Merkle Root και προστίθεται στην επικεφαλίδα του block. Στη συνέχεια, η επικεφαλίδα περνά από τη συνάρτηση κατακερματισμού που χρησιμοποιεί η κάθε υλοποίηση (SHA-256 το Bitcoin, και Keccak ή SHA-3 το Ethereum) για να παραχθεί το hash της επικεφαλίδας του block (το μπλε τμήμα στην Εικόνα 5.2).

Ακολούθως, το τμήμα Nonce συμπληρώνεται με τυχαίους αριθμούς και το συνολικό κομμάτι περνά ξανά από τη συνάρτηση κατακερματισμού. Το αποτέλεσμα συγκρίνεται με τον βαθμό δυσκολίας, πληροφορία που βρίσκεται στην επικεφαλίδα του block. Εφόσον ο συνολικός αριθμός που δηλώνεται από το παραγόμενο hash είναι μικρότερος από τον αριθμό του επιπέδου δυσκολίας, το block έχει ολοκληρώσει επιτυχώς το mining. Θα πρέπει να τονιστεί ότι η διαδικασία εύρεσης του σωστού nonce για το κάθε block δεν μπορεί να συντομευθεί και πρέπει να δοκιμαστούν όλοι οι πιθανοί συνδυασμοί, καθώς η αλλαγή ακόμα και ενός bit από το περιεχόμενο του πεδίου αλλάζει τελείως την έξοδο από τη συνάρτηση κατακερματισμού. Επίσης, δεν είναι δυνατόν να είναι γνωστό (ή να προβλεφθεί) από πριν το αποτέλεσμα που θα έχει η χρήση κάποιων αριθμών (ως τιμές στο πεδίο Nonce) και, επομένως, είναι αναγκαίο να δοκιμαστούν όλοι οι συνδυασμοί για να επιβεβαιωθεί (ή όχι) η εύρεση του περιεχομένου του πεδίου Nonce σε κάθε block.

Στο παράδειγμα παρακάτω φαίνεται η διαφορά που μπορεί να έχει η αλλαγή ενός αριθμού στο hash. Έτσι, θεωρούμε ότι το κομμάτι που αντιπροσωπεύει το hash της επικεφαλίδας block είναι (για λόγους ευκολίας) η πρόταση: *This is an introduction to blockchain* και συμπληρώνονται οι αριθμοί από το 1 έως 5 για το πεδίο Nonce.

Στον **Πίνακα 5.1** φαίνονται τα αποτελέσματα τόσο με τη χρήση του SHA-256 (Bitcoin) όσο και με τη χρήση του Keccak-256 (Ethereum).

Η διαφορά στις εξόδους είναι εμφανείς τόσο μεταξύ των δύο διαφορετικών συναρτήσεων κατακερματισμού (τιμές στη 2η και 3η στήλη) όσο και στην ίδια συνάρτηση κατακερματισμού (στήλη) για την ελαφρά διαφορά στην είσοδο.

Hash (Header + Nonce)	Έξοδος SHA-256	Έξοδος Keccak-256
This is an introduction to blockchain	ee8e053fdab3cd5…	7985a5fbe583be1…
This is an introduction to blockchain1	3a56ea82a11cf9c…	2f95d31154e3f88…
This is an introduction to blockchain2	011bdc924cb3489…	e3e981659ecc723…
This is an introduction to blockchain3	bb8bccb754b5a00…	e5d030247f60b17…
This is an introduction to blockchain4	d880272b545e4a2…	1e226adcc7ebb68…
This is an introduction to blockchain5	bdaa8035ef51bbe…	8276d2b212e5d39…

Πίνακας 5.1 *Το αποτέλεσμα του PoW με προσθήκη τυχαίων τιμών στο πεδίο Nonce.*

Μετά την εύρεση του κατάλληλου nonce, σειρά έχει η ενημέρωση του δικτύου για την επιτυχία εύρεσης του νέου block. Ο κάθε κόμβος που ενημερώνεται για αυτό λαμβάνει το νέο block και επιβεβαιώνει ότι το hash της επικεφαλίδας μαζί με το αναγραφόμενο nonce είναι όντως τέτοιο που ικανοποιεί τον βαθμό δυσκολίας.

Στην πράξη, στο Bitcoin η επιβεβαίωση του PoW είναι απλή υπόθεση. Αυτό συμβαίνει γιατί για να ικανοποιείται ο ζητούμενος βαθμός δυσκολίας θα πρέπει η λύση να χαρακτηρίζεται από 17 μηδενικά ψηφία στην αρχή, για να εξασφαλιστεί ότι είναι αρκετά μικρός ο αριθμός. Και πάλι, όμως, επαλήθευση του αποτελέσματος της συνάρτησης κατακερματισμού είναι απαραίτητη για τη συναίνεση.

Όποιος κόμβος επιτύχει να ολοκληρώσει πρώτος το PoW τότε αναλαμβάνει να ενημερώσει τους υπόλοιπους για την επιτυχία του. Όσοι κόμβοι ενημερώνονται επιβεβαιώνουν το νέο block και ξεκινούν αμέσως την προσπάθεια εύρεσης του επόμενου.

Η διαδικασία αυτή συνεχίζεται συνεχώς στο δίκτυο. Σπανίως όμως παρατηρείται το φαινόμενο δύο κόμβοι να βρουν σχεδόν ταυτόχρονα το νέο block. Αυτό είναι δυνατό μιας και έχει ο καθένας να λύσει διαφορετικό

πρόβλημα, λόγω π.χ., διαφορών στις συναλλαγές που έχουν συμπεριληφθεί (ακόμα και απλώς να υπάρχει διαφορά στη σειρά τους, οδηγεί σε άλλο πρόβλημα λόγω των Merkle Trees και των συναρτήσεων κατακερματισμού). Καθώς, επίσης, η επίλυση του σημείου δυσκολίας δεν είναι ένας αριθμός αλλά ένα σύνολο από αριθμούς (από τον αριθμό 1 μέχρι μια τιμή που αποτελεί και τη δήλωση του σημείου δυσκολίας), είναι πιθανό 2 κόμβοι να ολοκληρώσουν σχεδόν ταυτόχρονα τη διαδικασία του PoW για το block που έχει ο καθένας τους.

Στην περίπτωση αυτή υπάρχουν, για ένα μικρό διάστημα, δύο διαφορετικές αλυσίδες μέσα στο δίκτυο, δημιουργώντας ένα fork[66] στο ledger του blockchain. Τη μία θα την έχουν όσοι ενημερώθηκαν από τον έναν κόμβο και την άλλη θα την αναγνωρίζουν όσοι ενημερώθηκαν από τον δεύτερο (π.χ. λόγω εγγύτητας στον κόμβο που βρήκε το block).

Είναι σημαντικό να γίνει αντιληπτό ότι τη στιγμή αυτή *δεν* έχει γίνει κάποιο λάθος, καθώς και οι δύο κόμβοι έχουν ολοκληρώσει ορθά τη διαδικασία του PoW. Από την άλλη, δεν έχει επέλθει συναίνεση στο δίκτυο, καθώς υπάρχουν δύο αλυσίδες. Για να επέλθει συναίνεση, θα πρέπει το δίκτυο να επιλέξει μία από αυτές και όλοι οι κόμβοι να συγκλίνουν στην αλυσίδα αυτή.

Αυτό γίνεται με τη δημιουργία του επόμενου block. Η αλυσίδα πάνω στην οποία θα τοποθετηθεί το νέο block θα αποτελέσει τη μακρύτερη αλυσίδα, η οποία, σύμφωνα και με τον Nakamoto (2008), θα είναι αυτή στην οποία και θα συγκλίνει το δίκτυο. Για τον λόγο αυτόν ο κανόνας της επιλογής της πιο μακριάς αλυσίδας για την επίτευξη της συναίνεσης έχει πάρει και το όνομα της *συναίνεσης Nakamoto* (*Nakamoto consensus*).

Με τη σύγκλιση οι κόμβοι όλοι ενημερώνονται για τη δομή και το περιεχόμενο των blocks και τα τοποθετούν στο ledger, διαγράφοντας όποια άλλα blocks είχαν πιστέψει ότι ήταν τα σωστά.

Λόγω της πιθανότητας δημιουργίας κάποιου fork[67] και, επομένως, αλλαγής του block που θα μπει στο ledger, ένας miner που βρίσκει το επόμενο block δεν πληρώνεται αμέσως από το δίκτυο. Αντιθέτως, αναμένει έναν αριθμό από επιβεβαιώσεις προτού γίνει σίγουρο ότι το δικό του block είναι αυτό που έχει μπει στο ledger. Ο αριθμός των επιβεβαιώσεων είναι περίπου 6 για το Bitcoin και 7 για το Ethereum, αν και στην πράξη μπορεί να χρειαστεί έλεγχος των τελευταίων 250 blocks στο δίκτυο του Ethereum.

Βεβαίως, εξακολουθεί και υπάρχει μικρή πιθανότητα η ταυτόχρονη ανακάλυψη νέων blocks να συνεχίσει τον συγχρονισμό της και στα 2 παράλληλα ledgers πριν από τη σύγκλιση (δηλαδή να έχουμε 2 ταυτόχρονα νέα blocks για δεύτερη φορά). Αυτή η κατάσταση αναμένεται να μην αποτελέσει πρόβλημα και να επέλθει, τελικά, η συναίνεση μόλις δημιουργηθεί νέο block μόνο σε ένα από τα δύο παρακλάδια του εν δυνάμει fork.

Αξίζει να αναφερθεί ότι στο Bitcoin τον Απρίλιο του 2013 παρουσιάστηκε fork των 26 ταυτόχρονων blocks! Αυτό το fork είχε δημιουργηθεί λόγω της αλλαγής στη Βάση Δεδομένων που χρησιμοποιούσαν οι κόμβοι (από την Berkeley DB έγινε μετάβαση στη LevelDB) και στην οποία δεν είχαν προλάβει να μεταβούν όλοι, με αποτέλεσμα ένα σφάλμα στους περιγραφείς αρχείων να δημιουργήσει πρόβλημα σε όσους δεν είχαν προχωρήσει στη μετάβαση.

Εν κατακλείδι, το PoW έχει ορισμένα σημαντικά πλεονεκτήματα, όπως την αποτελεσματική αντιμετώπιση του double spending και τη δικαιοσύνη μεταξύ όλων των κόμβων, καθώς όλα στηρίζονται στην κρυπτογραφία και όχι σε αποφάσεις τρίτων. Από την άλλη, έχει και ορισμένα σημαντικά μειονεκτήματα, όπως είναι η σημαντική κατανάλωση ενέργειας που απαιτείται για την επίλυση του μαθηματικού γρίφου και ο μικρός αριθμός συναλλαγών ανά δευτερόλεπτο που, συχνά, εμφανίζει.

5.2.2 Proof of Stake (PoS)

Το PoS αποτελεί μια εναλλακτική πρόταση για την επιβεβαίωση των blocks σε ένα δίκτυο blockchain. Χρησιμοποιείται για να επέλθει συναίνεση στο δίκτυο αλλά και για την ενημέρωση του κοινού ledger με το αποτέλεσμα της συναίνεσης αυτής. Οι λόγοι για τους οποίους αναπτύχθηκε, και πλέον χρησιμοποιείται, η νέα αυτή τεχνική συναίνεσης σχετίζονται με την αντιμετώπιση των μειονεκτημάτων της τεχνικής του PoW. Τα μειονεκτήματα αυτά έχουν εντοπιστεί μετά από μια σημαντική και εξαιρετικά επιτυχημένη περίοδο χρήσης του.

Το PoS έρχεται κυρίως να αντιμετωπίσει το σημαντικό ενεργειακό αποτύπωμα του PoW. Λόγω των έντονων επεξεργαστικών υπολογισμών για την εύρεση μιας λύσης εντός του επιπέδου δυσκολίας, χρειάζεται μια προσέγγιση η οποία να είναι πολύ λιγότερη απαιτητική ενεργειακά. Έτσι, στο PoS καταργείται το mining και πλέον όσοι κόμβοι επιθυμούν να λάβουν μέρος στη διαδικασία εύρεσης του νέου block στο ledger συναλλάσσονται με ένα ειδικό smart contract, στο οποίο και τοποθετούν τα κρυπτονομίσματα που επιθυμούν

[66] Η ορολογία έχει προέλθει από το γεγονός ότι τα δύο ταυτόχρονα blocks αποδίδονται συχνά σχηματικά σαν ένα πιρούνι μέχρις ότου να επέλθει σύγκλιση σε μία από τις δύο επιλογές.

[67] Στο Bitcoin fork του ενός block γίνεται κατά μέσο όρο μία κάθε μέρα.

να χρησιμοποιήσουν για να αποκτήσουν το δικαίωμα δημιουργίας του νέου block. Η τεχνική αυτή μοιάζει με ποντάρισμα για την απόκτηση του δικαιώματος και, για τον λόγο αυτόν, η διαδικασία τοποθέτησης των κρυπτονομισμάτων στο smart contract θεωρείται ένα είδος ποντάρίσματος (stake).

Με τη συλλογή όλων των stakes από τους κόμβους, το smart contract επιλέγει τυχαία ανάμεσα σε αυτούς έναν (ο οποίος πλέον ονομάζεται *εκτιμητής*) για να προτείνει το νέο block. Ο κόμβος-εκτιμητής που επιλέγεται δημιουργεί το νέο block, αφού επαληθεύσει τις συναλλαγές που περικλείει σε αυτό, και κατόπιν το παρουσιάζει στο δίκτυο για επαλήθευση από τους ομότιμούς του κόμβους-εκτιμητές.

Εφόσον όλα έχουν γίνει σωστά και οι εκτιμητές επιβεβαιώσουν την εγκυρότητα του νέου block, ο κόμβος που το πρότεινε επιβραβεύεται για τη δουλειά του, μαζεύοντας όλα τα fees των συναλλαγών που έχουν μπει στο block. Αν όμως το νέο block έχει συναλλαγές που είναι άκυρες, τότε αυτό θα ακυρωθεί από τους εκτιμητές και ο κόμβος που το πρότεινε κινδυνεύει να χάσει το stake που έχει βάλει στο smart contract. Τα κρυπτονομίσματα αυτά παίζουν, δηλαδή, και τον ρόλο της εγγύησης της ορθής λειτουργίας του κόμβου που συμπεριλαμβάνεται στη λίστα των εκτιμητών.

Η μέθοδος αυτή έχει αποκτήσει αρκετούς υποστηρικτές, ιδιαίτερα λόγω της αντιμετώπισης του προβλήματος της ενεργειακής επιβάρυνσης που φέρνει το PoW και χρησιμοποιείται σε αρκετές γνωστές υλοποιήσεις δικτύων blockchain, όπως είναι το Cardano και το Solana. Θα είναι δε και ο μηχανισμός συναίνεσης στη νέα έκδοση του Ethereum (Ethereum 2.0), αντικαθιστώντας το PoW. Αναμένεται, επίσης, στο Ethereum να προσφέρει και δυνατότητες επέκτασης με χρήση τμημάτων (shards), δηλαδή μικρότερες αλυσίδες που θα τρέχουν παράλληλα με την κύρια αλυσίδα και θα επιτρέπουν την καλύτερη διαχείριση των δεδομένων. Οι εκτιμητές θα χρειαστεί να έχουν πρόσβαση στα δεδομένα του τμήματος και όχι σε ολόκληρη την κύρια αλυσίδα, με αποτέλεσμα να μην είναι αναγκαίο κάποιος ειδικός εξοπλισμός για τη συμμετοχή των κόμβων.

Στα πλεονεκτήματα του PoS, πέρα από την ενεργειακή βελτίωση, θα πρέπει να αναφερθεί η απουσία ανάγκης για εξειδικευμένο εξοπλισμό για τη συμμετοχή στη διαδικασία. Αυτό είναι ένα χαρακτηριστικό το οποίο προσδίδει μεγαλύτερη αποκέντρωση στους χρήστες, μιας και διευκολύνεται ο καθένας να συμμετάσχει. Επιπλέον, προσδίδει αυξημένη ασφάλεια, καθώς καθιστά πολύ ακριβή τη χειραγώγηση του δικτύου από έναν κακόβουλο χρήστη. Επίσης, οι λιγότεροι υπολογισμοί οδηγούν σε περισσότερες συναλλαγές ανά δευτερόλεπτο.

Στα μειονεκτήματά του περιλαμβάνεται η απαίτηση για χρήση κρυπτονομισμάτων για το stake, η οποία μπορεί να είναι αποτρεπτική, καθώς θα πρέπει κάποιος να επενδύσει πρώτα σε αυτά έτσι ώστε μετά να έχει κέρδος από τη συμμετοχή του και να μπορεί να συνεχίσει να υποστηρίζει τον ρόλο του εκτιμητή. Το αποτέλεσμα είναι να κινδυνεύει να είναι πιο πολύ κεντρικοποιημένο παρά τα χαρακτηριστικά του, που αρχικά ενισχύουν την αποκέντρωση.

5.2.3 Σύγκριση Proof of Work (PoW) με Proof of Stake (PoS)

Έχοντας παρουσιάσει τα ιδιαίτερα χαρακτηριστικά της καθεμίας τεχνικής, ακολουθεί μια ομαδοποίησή τους. Έτσι, στον **Πίνακα 5.2** φαίνεται μια περίληψη των δυνατοτήτων των δύο σημαντικών αλγόριθμων συναίνεσης: του PoW και του PoS. Φαίνονται, επίσης, οι διαφορές τους, τα πλεονεκτήματα και τα μειονεκτήματά τους.

Χαρακτηριστικό	PoW	PoS
Δημιουργία και επιβεβαίωση νέου block	Η δημιουργία του block εξαρτάται από την επεξεργαστική εργασία του κάθε κόμβου.	Το ποσό των κρυπτονομισμάτων (stake) επηρεάζει την πιθανότητα δημιουργίας του νέου block.
Διαμοιρασμός κερδών	Ο miner που βρίσκει το block ανταμείβεται με καινούργια νομίσματα για αυτό.	Η ανταμοιβή του κόμβου που βρίσκει το block αφορά μόνο τη συλλογή των transaction fees από τις συναλλαγές που έχουν μπει σε αυτό.
Ανταγωνισμός	Οι miners ανταγωνίζονται για την επίλυση ενός δύσκολου επεξεργαστικού γρίφου.	Ένας αλγόριθμος αποφασίζει για τον κόμβο που θα προτείνει το νέο block βάσει του ποσού που κατατέθει αυτός ως stake.
Κεντρικοποίηση	Λόγω της ενισχυμένης επεξεργαστικής ισχύος και της μεγάλης κατανάλωσης σε ενέργεια, σημειώνεται σημαντική κεντρικοποίηση των κόμβων σε συγκεκριμένες γεωγραφικές περιοχές	Οι μειωμένες απαιτήσεις σε ενέργεια επιτρέπουν τη μεγαλύτερη αποκεντρικοποίηση. Η απαίτηση για ένα σημαντικό ποσό για το stake όμως επηρεάζει το αποτέλεσμα της πράξης αυτής.
Ειδικός εξοπλισμός	Γίνεται χρήση ASIC[68]s και GPUs[69] για το mining των νομισμάτων.	Μια οποιαδήποτε συσκευή που μπορεί να λειτουργήσει ως server είναι αρκετή.

[68] Application-Specific Integrated Circuits.
[69] Graphics Processing Unit.

Προσθήκη κακόβουλου block	Για την αποδοχή ενός κακόβουλου block οι επιτιθέμενοι πρέπει να έχουν το 51% της επεξεργαστικής δύναμης του δικτύου.	Για την αποδοχή ενός κακόβουλου block οι επιτιθέμενοι πρέπει να έχουν το 51% των κρυπτονομισμάτων του δικτύου.
Αποτελεσματικότητα και Αξιοπιστία	Τα συστήματα του PoW έχουν μικρή αποτελεσματικότητα στην κατανάλωση της ενέργειας και είναι ακριβά (λόγω της απαίτησης ειδικού εξοπλισμού και της τιμής του ηλεκτρικού ρεύματος). Έχουν σημαντική αξιοπιστία.	Τα συστήματα του PoS έχουν καλύτερη απόδοση ενεργειακά και μικρότερο κόστος. Είναι λιγότερο αξιόπιστα.
Ασφάλεια	Όσο πιο μεγάλο είναι ένα hash, τόσο περισσότερο ασφαλές είναι το δίκτυο.	Όσο μεγαλύτερο είναι το stake, τόσο πιο ασφαλές είναι το δίκτυο (η κακόβουλη χρήση κοστίζει σημαντικά).

Πίνακας 5.2 *Σύγκριση των χαρακτηριστικών του PoW με το PoS.*

5.2.4 Άλλες τεχνικές συναίνεσης

Όπως αναφέρθηκε προηγουμένως, υπάρχει μια σημαντική ανάπτυξη αλγόριθμων που αντιμετωπίζουν με διαφορετική προσέγγιση το θέμα της κατανεμημένης συναίνεσης. Στη συνέχεια γίνεται μια σύντομη παρουσίαση ορισμένων τέτοιων λύσεων.

- *Delegated Proof of Stake* (*DPoS*): Πρόκειται για μια τεχνική η οποία έχει πολλές ομοιότητες με το PoS (Howell, 2022). Για την ακρίβεια, και στις δύο αυτές τεχνικές (DPoS και PoS) οι συμμετέχοντες καλούνται να χρησιμοποιήσουν ως «εχέγγυο» ορθής λειτουργίας στη συναίνεση του δικτύου ένα ποσό από κρυπτονομίσματα (stake). Η διαφορά τους είναι στο γεγονός ότι στη δημιουργία και επιβεβαίωση του κάθε νέου block συμμετέχουν μόνο κόμβοι που επιλέγονται ως αντιπρόσωποι (delegates).
  Η επιλογή των αντιπροσώπων γίνεται από τους χρήστες του δικτύου και είναι παρόμοια με μια ψηφοφορία. Σε αυτήν όμως, αντί για την ψήφο, οι χρήστες αποστέλλουν ένα τμήμα από τα κρυπτονομίσματά τους για να ενισχύσουν το stake του αντιπροσώπου που έχουν επιλέξει. Έτσι, αυξάνονται οι πιθανότητες του αντιπροσώπου να επιλεγεί να συμμετάσχει στη δημιουργία του νέου block. Συνήθως, ανάλογα και με την υλοποίηση, ο αριθμός των αντιπροσώπων είναι μεταξύ 20 και 100.
  Όπως αναφέρθηκε, η διαδικασία της αρχικής επιλογής αντιπροσώπων αλλά, κατόπιν, και της επιλογής ανάμεσά τους αυτών που θα συμμετάσχουν στη δημιουργία του κάθε block είναι δυναμική. Επομένως, οι αντιπρόσωποι που επιλέχθηκαν στο παρελθόν μπορεί να μην επιλεγούν να συμμετάσχουν ξανά στο άμεσο μέλλον, επιτρέποντας έτσι την ανανέωση και τη συμμετοχή αρκετών κόμβων στη διαδικασία της συναίνεσης.
  Το κέρδος του απλού χρήστη μπορεί να είναι, επίσης, σημαντικό. Δεδομένου ότι ο κάθε αντιπρόσωπος που συμμετέχει στο νέο block λαμβάνει ως ανταμοιβή όλα τα έξοδα των συναλλαγών (transaction fees) που μπήκαν στο block, ενώ αναλαμβάνει να μοιράσει τα fees αυτά σε όσους χρήστες τού έδωσαν ποσά. Η πολιτική είναι να επιστραφεί σε κάθε χρήστη ποσό από τα κέρδη αντίστοιχο με το ποσοστό του στο σύνολο του stake. Δηλαδή, αν ένας χρήστης έχει δώσει ποσοστό ίσο με 5% των κρυπτονομισμάτων που ο αντιπρόσωπος έβαλε για stake, τότε θα λάβει πίσω το 5% των ανταμοιβών των συναλλαγών του block. Να σημειωθεί ότι η μεταφορά χρημάτων από τον χρήστη στον αντιπρόσωπο δεν γίνεται πραγματικά (δηλαδή δεν καταγράφεται συναλλαγή μεταξύ των πορτοφολιών τους), αλλά γίνεται χρήση μιας υπηρεσίας μεταφοράς ποσών για stake, που είναι υπεύθυνη για τη λειτουργία αυτή.
  Ο μηχανισμός αυτός θεωρείται ότι είναι πιο δημοκρατικός ως προς τη σύνθεση των κόμβων που συμμετέχουν στη δημιουργία του block, καθώς δίνει τη δυνατότητα σε ένα ποικιλόμορφο σύνολο συμμετεχόντων να μπορέσει να συμμετάσχει στη διαδικασία της συναίνεσης. Και αυτό γιατί η επιλογή των αντιπροσώπων από τους χρήστες δεν βασίζεται αποκλειστικά στο ποσό που βάζουν ως stake, αλλά στη φήμη τους στο δίκτυο. Έτσι, είναι δυνατόν να αποκτήσουν χρήματα από άλλους χρήστες, που θα ενισχύσουν το stake τους, και με τον τρόπο αυτόν τις πιθανότητές τους για συμμετοχή στη δημιουργία του νέου block.
  Τέλος, το DPoS φαίνεται πως αποτελεί μια λύση αρκετά επίκαιρη στον χώρο του blockchain, καθώς έχει υιοθετηθεί από ανερχόμενες προτάσεις στον χώρο, όπως είναι τα δίκτυα EOS και Cardano.
- *Proof of Activity* (*PoA*): Πρόκειται για μια υβριδική τεχνική που συνδυάζει τα χαρακτηριστικά των δύο τεχνικών συναίνεσης, του PoW και του PoS (Bentov et al., 2014).

Σε αυτήν το block αρχικά δημιουργείται με την ίδια επεξεργαστική διαδικασία που λαμβάνει χώρα στο PoW, η οποία και είναι ενεργειακά επιβαρυντική. Για την εύρεση του κάθε block η εργασία πρέπει να ικανοποιεί μια συγκεκριμένη τιμή του επιπέδου δυσκολίας. Το αρχικό block που δημιουργείται είναι κενό. Μόλις ολοκληρωθεί η εύρεση του νέου block, τότε σε αυτό θα πρέπει να μπουν συναλλαγές και στη συνέχεια να επικυρωθεί από ένα σύνολο επικυρωτών, οι οποίοι επιλέγονται βάσει του stake που έχουν δώσει. Δηλαδή, η συμπλήρωση των συναλλαγών και η επικύρωση γίνονται με χαρακτηριστικά του πρωτοκόλλου PoS. Μάλιστα, για την επικύρωση διατηρείται ένα χρονικό διάστημα μέσα στο οποίο οι κόμβοι επαλήθευσης πρέπει να υπογράψουν το νέο block, αλλιώς χάνει τη σειρά του και προχωρά η αναζήτηση στο επόμενο.

Το PoA δημιουργεί ένα νέο block κάθε 5 λεπτά και έχει ως πλεονεκτήματα ότι είναι ασφαλές, καθώς η απόφαση των επικυρωτών είναι αυτή που προσθέτει το κάθε νέο block και η επιλογή αυτών είναι δυναμική. Στα μειονεκτήματά του συγκαταλέγεται η χρήση της ενεργειακά επιβαρυντικής διαδικασίας εύρεσης του PoW.

Η πιο γνωστή λύση δικτύου blockchain η οποία χρησιμοποιεί το PoA είναι το Decred (DCR)[70]. Στο Decred, μετά την επιτυχημένη επικύρωση ενός block, το άθροισμα των transaction fees διαμοιράζεται στον miner του PoW (60% της αμοιβής) και στους επαληθευτές που επικυρώνουν το νέο block με το PoS (30%). Το υπόλοιπο 10% δίνεται για την υποστήριξη της ανάπτυξης του πρωτοκόλλου.

- *Proof of Authority* (*PoA*): Είναι μια ακόμα τεχνική η οποία χρησιμοποιεί ως βάση το PoS, αλλά αυτή τη φορά, εκτός από το stake, οι επαληθευτές ρισκάρουν και τη φήμη τους (Cointelegraph, 2022). Για αυτό οι επαληθευτές είναι λίγοι σε αριθμό και επιλέγονται από μια δύσκολη διαδικασία, η οποία όμως δίνει ίσες ευκαιρίες σε όλους τους συμμετέχοντες. Η δυσκολία αυτή στη διαδικασία επιλογής έχει σκοπό να ενθαρρύνει τη συμμετοχή σε αυτή μόνο όσων επιθυμούν τη μακρόχρονη ενασχόλησή τους με τον ρόλο αυτό, αποτρέποντας άλλους που έχουν εφήμερες βλέψεις.

  Κατά τη διάρκεια της διαδικασίας επιλογής αξιολογείται η φήμη του κάθε συμμετέχοντος και εφόσον επιλεγεί τα στοιχεία του γίνονται γνωστά στο δίκτυο. Έτσι, με τη συμμετοχή του στην επιλογή συναλλαγών για το νέο block και, επομένως, στη συναίνεση του δικτύου, οι πράξεις του αξιολογούνται και είτε ενισχύουν τη φήμη του είτε την υποβαθμίζουν.

  Τα χαρακτηριστικά της τεχνικής αυτής την κάνουν περισσότερο δημοφιλή σε ιδιωτικά δίκτυα blockchain ή σε δίκτυα κοινοπραξίας. Ο λόγος είναι ότι, στην πράξη, βασίζονται στην κρίση ορισμένων (μικρού αριθμού) συμμετεχόντων και, επομένως, στα μειονεκτήματά της συγκαταλέγεται η κεντρικοποίηση που φέρνει στη λειτουργία της η τεχνική αυτή. Χαρακτηριστικό βέβαια που ταιριάζει ιδιαίτερα στα προαναφερθέντα είδη δικτύων blockchain. Ένα ακόμα μειονέκτημα είναι και το γεγονός ότι η ταυτότητα των επαληθευτών είναι γνωστή, με αποτέλεσμα να μπορούν να δεχθούν πιέσεις στον φυσικό κόσμο.

  Από την άλλη πλευρά, στα πλεονεκτήματά της είναι, πάλι, αυτή η μικρή ομάδα επαληθευτών, η οποία μπορεί, αν είναι πιο ευέλικτη, και να καταφέρει μεγαλύτερους ρυθμούς συναλλαγών ανά δευτερόλεπτο (tps), καθώς και πιο συχνή δημιουργία νέου block. Τέλος, το γεγονός ότι η ταυτότητα των επαληθευτών είναι γνωστή μπορεί να επηρεάσει θετικά και τον τρόπο συμπεριφοράς τους μέσα στο δίκτυο, καθώς επιθυμούν να διατηρήσουν αυτή την καλή τους φήμη.

  Η τεχνική του Proof of Authority αποτελεί μια σημαντική επιλογή, ιδιαίτερα στα είδη δικτύων που προαναφέρθηκαν. Έχει βρει τον δρόμο της σε υλοποιήσεις, με πιο γνωστή αυτή της VeChain[71] και είναι ιδιαίτερα δημοφιλής στον τομέα της διαχείρισης της εφοδιαστικής αλυσίδας.

- *Proof of Burn* (*PoB*): Πρόκειται για μια νέα τεχνική η οποία και αποτελεί μια παραλλαγή του PoW, χωρίς όμως την ενεργειακή κατανάλωση που το χαρακτηρίζει (Frankenfield, 2021). Στο PoB ένας miner, για να κερδίσει το δικαίωμα να φτιάξει το επόμενο block, μπορεί να στείλει τα χρήματά του σε έναν ειδικό λογαριασμό. Ανάλογα με το ποσό που θα στείλει, θα του δοθεί και το δικαίωμα να δημιουργήσει νέα blocks. Είναι, δηλαδή, σαν να χρησιμοποιεί τα χρήματα για να «αγοράσει» εγκαταστάσεις και να τις χρησιμοποιήσει για να εξορύξει blocks.

  Η διεύθυνση προς την οποία στέλνονται τα χρήματα δεν ανήκει σε κανέναν. Κανείς δεν έχει τα ιδιωτικά της κλειδιά, αλλά όλοι μπορούν να παρακολουθήσουν τις συναλλαγές προς αυτήν. Ως

[70] Online Σύνδεσμος: https://decred.org/
[71] Online Σύνδεσμος: https://www.vechain.org/

επακόλουθο, ό,τι κρυπτονομίσματα στέλνονται στη διεύθυνση αυτή στην ουσία «καίγονται» (εξ ου και το όνομα της τεχνικής), καθώς δεν μπορούν πλέον να αξιοποιηθούν από κανέναν στο δίκτυο. Στην πραγματικότητα, η πράξη της αποστολής κρυπτονομισμάτων για να «καούν» χρησιμοποιείται ως μια ένδειξη εμπιστοσύνης στο δίκτυο, η οποία αναμένεται να επιβραβευθεί μελλοντικά, μέσω της κατάκτησης του δικαιώματος συμμετοχής στη διεργασία του mining.

Ακόμα, υπάρχει η δυνατότητα της αποστολής χρημάτων σε άλλα μέλη του δικτύου για να τα στείλουν εκείνα προς καύση, ενισχύοντας τη δυνατότητα αυτών για συμμετοχή στη δημιουργία blocks. Επίσης, ανάλογα με την υλοποίηση, επιτρέπεται είτε η καύση του κρυπτονομίσματος του δικτύου το οποίο εφαρμόζει το PoB είτε η καύση άλλων νομισμάτων (π.χ. bitcoins). Η ανταμοιβή όμως για τη δημιουργία του block είναι πάντα στη μορφή του κρυπτονομίσματος του δικτύου.

Τέλος, το δίκτυο, για να αποφύγει την απόκτηση πλεονεκτήματος από τους παλιότερους χρήστες, έχει φροντίσει για μια περιοδική καύση των κρυπτονομισμάτων του δικτύου, μικραίνοντας τη διαφορά μεταξύ παλιών και νέων χρηστών.

Παράδειγμα δικτύου που χρησιμοποιεί τη νέα αυτή τεχνική αποτελεί το Slimcoin[72], το οποίο επιτρέπει την καύση κρυπτονομισμάτων όχι μόνο ως δικαίωμα συμμετοχής στη διαδικασία του mining, αλλά και για την παράταση του χρονικού διαστήματος λήψης νέων blocks.

- *Proof of Capacity* (*PoC*): Πρόκειται για μια επίσης καινούργια τεχνική η οποία έχει στόχο να αντιμετωπίσει την ενεργειακή κατανάλωση που προκαλεί το PoW. Ουσιαστικά, το PoC χρησιμοποιεί χώρο στον σκληρό δίσκο για να αποδείξει ότι έχει το ενδιαφέρον για να συμμετάσχει στο δίκτυο (TheLuWizz, 2021).

  Έτσι, στο PoC ένας χρήστης χρησιμοποιεί τον ελεύθερο χώρο στον δίσκο του πριν από τη δημιουργία του block. Εκεί παράγει τις τιμές των υποψήφιων nonces με τη χρήση συναρτήσεων κατακερματισμού από τον λογαριασμό του στο δίκτυο (Account ID). Η Shabal 256 είναι μια τέτοια συνάρτηση κατακερματισμού που χρησιμοποιείται από το Burtscoin, την πιο δημοφιλή (προς το παρόν) υλοποίηση blockchain που χρησιμοποιεί τον αλγόριθμο του PoC. Η λίστα με τις υποψήφιες τιμές για το nonce από κάθε λογαριασμό αποτελεί έναν σχεδιασμό του σκληρού δίσκου του χρήστη, προαπαιτούμενο βήμα για τη συμμετοχή του στη διαδικασία. Ο συνδυασμός της λίστας με το Account ID του χρήστη επιτρέπει τον διαχωρισμό των όμοιων τιμών nonce μεταξύ των διαφορετικών χρηστών.

  Επιπλέον, η τιμή του κάθε nonce στη λίστα αποτελείται από 8.192 hashes (αριθμούμενα από το 0 έως το 8.191). Κάθε ζευγάρι από hashes σε μια τιμή του nonce σχετίζεται και αποτελεί ένα scoop. Επομένως, κάθε nonce περιέχει 4.096 scoops, τα οποία το καθένα αποτελείται από 64 bytes δεδομένων και από 2 hashes.

  Κατά τη διάρκεια του mining κάθε υποψήφιος miner υπολογίζει τον αριθμό στο scoop και κατόπιν χρησιμοποιεί αυτόν τον αριθμό για να παράγει δεδομένα. Τα δεδομένα αυτά αποτελούν έναν υπολογισμό για τη μικρότερη χρονική διάρκεια μετά τη δημιουργία του προηγούμενου block, πέρα από την οποία κάποιος miner μπορεί να βρει και να προσθέσει το νέο block. Ο miner υπολογίζει αυτή τη μικρότερη χρονική διάρκεια για όλα τα scoops και όλα τα nonces που έχει υπολογίσει και κρατά τη (συνολικά) μικρότερη τιμή που έχει βρει. Αν είναι και η μικρότερη σε όλο το δίκτυο, μπορεί μόλις ολοκληρωθεί να προχωρήσει στην εξόρυξη και προσθήκη του νέου block. Αλλιώς, αν τον προλάβει κάποιος άλλος miner, κρατά τις τιμές αυτές για το νέο block και επαναλαμβάνεται η διαδικασία.

  Με τον τρόπο αυτόν το PoC δημιουργεί 1 block κάθε 4 λεπτά, σε αντίθεση με το PoW που δημιουργεί κάθε 10. Συνολικά, στα πλεονεκτήματα της τεχνικής συμπεριλαμβάνονται η χρήση κοινών σκληρών δίσκων, η καλύτερη ενεργειακή απόδοση (30 φορές πιο αποδοτικό ενεργειακά σε σχέση με το PoW), ο μικρότερος χρόνος για την εύρεση του νέου block (πάντα συγκρινόμενο με το PoW, όπου έχει 50% βελτίωση) και, τέλος, το γεγονός ότι ο σκληρός δίσκος δεν είναι αναγκαίο να χρησιμοποιηθεί συνολικά αποκλειστικά για τη διαδικασία αυτή. Στα μειονεκτήματά του είναι ότι υστερεί σε υιοθέτηση από πολλές λύσεις, αν και μελετάται από αρκετές.

- *Proof of Elapsed Time* (*PoET*): Πρόκειται για μια τεχνική η οποία, και αυτή, προσπαθεί να αντιμετωπίσει την ενεργειακή κατανάλωση που εισήγαγε το PoW. Για να το πετύχει αυτό, στο PoET κάθε κόμβος τρέχει ένα ειδικό ρολόι, το οποίο επιλέγει τυχαία μια τιμή και μετρά αντίστροφα (Centieiro, 2021). Όταν μηδενιστεί, τότε αυτό σημαίνει ότι ο κόμβος μπορεί να δημιουργήσει το νέο block. Αν προλάβει

[72] Online Σύνδεσμος: https://slimcoin.info/

το ρολόι άλλου κόμβου να μηδενιστεί πρώτα, τότε επιλέγεται νέα τιμή και ξεκινά από την αρχή.
Δεδομένης της χρήσης ενός ρολογιού, θα πρέπει να εξασφαλιστεί ότι όλοι οι κόμβοι λειτουργούν με τους ίδιους όρους ως προς τον προσδιορισμό της τιμής για το ρολόι αυτό. Έτσι, λοιπόν, γίνεται χρήση συγκεκριμένου hardware της εταιρείας Intel, το οποίο είναι ένας ειδικός επεξεργαστής που ονομάζεται SGX (Software Guard Extensions). Ο επεξεργαστής αυτός επιτρέπει τον λογικό διαχωρισμό της μνήμης του, η οποία και δεν μπορεί να τροποποιηθεί ή να αποκτήσει κάποιος πρόσβαση σε αυτή. Σε αυτό το τμήμα τοποθετείται ο κώδικας υλοποίησης της τεχνικής του PoET, εξασφαλίζοντας την ομαλή λειτουργία του σε όλους τους κόμβους που πληρούν τις προϋποθέσεις. Ο λόγος χρήσης αυτής της λύσης από την Intel σχετίζεται με την αρχική υλοποίηση της τεχνικής του PoET στο δίκτυο του Hyperledger Sawtooth το 2016-17.
Τέλος, έχοντας την παραπάνω τεχνική απαίτηση, πρέπει να εξασφαλιστεί ότι όλοι οι κόμβοι του δικτύου την ικανοποιούν, προτού γίνουν μέλος αυτού. Επομένως, κάθε κόμβος παίρνει ειδική άδεια για τη συμμετοχή του (εφόσον αποδειχθεί ότι ικανοποιεί την απαίτηση) και, έτσι, επηρεάζεται και το είδος των δικτύων όπου μπορεί να εφαρμοστεί η τεχνική αυτή.
Στα πλεονεκτήματα της τεχνικής είναι η μεγαλύτερη ενεργειακή οικονομία που επιφέρει με την επιβολή των τυχαίων time-outs, ενώ στα μειονεκτήματα η ισχυρή εξάρτηση από τη συγκεκριμένη λύση μιας εταιρείας για την υλοποίηση.

- *Proof of History* (*PoH*): Το PoH προσπαθεί να λύσει το πρόβλημα του χρόνου σε ένα κατανεμημένο δίκτυο. Για να το καταφέρει αυτό, αντί να βασίζεται στην απόδοση και εύρεση του απόλυτου χρόνου όπου έγινε ένα συμβάν, επιχειρεί να το τοποθετήσει πριν ή μετά συγκεκριμένων χρονικών στιγμών αναφοράς στο δίκτυο (Yakovenko, 2018). Για αυτό χρησιμοποιεί μια σειρά από κατακερματισμούς, οι οποίοι λαμβάνουν χώρα σε έναν υπολογιστή, για να αποδείξει τον χρόνο που πέρασε στο δίκτυο.
Στην ουσία, το PoH βασίζεται στη λογική ότι δεν μπορεί να προβλεφθεί μια έξοδος 120 δευτερόλεπτα πιο μετά, αν κάποιος δεν έχει κάνει όλους τους ενδιάμεσους υπολογισμούς. Έτσι, λοιπόν, το PoH κάνει χρήση μιας από τις συναρτήσεις κατακερματισμού που αναφέρθηκαν στο Κεφάλαιο 3, ξεκινώντας από μια τυχαία είσοδο και παράγοντας σε κάθε βήμα την έξοδο της συνάρτησης, η οποία μετά θα ξαναχρησιμοποιηθεί ως είσοδος στην ίδια συνάρτηση. Με τον τρόπο αυτόν, εφόσον αποθηκεύονται σε κάθε βήμα η έξοδος και ο αριθμός των φορών που έχει ανατροφοδοτηθεί η κάθε έξοδος στη συνάρτηση κατακερματισμού, δημιουργείται μια αντικειμενική αναφορά στον χρόνο μέσα στο σύστημα.
Το θετικό είναι ότι, αν και για τη δημιουργία της σειράς που απαιτείται στο PoH πρέπει να χρησιμοποιηθεί ένας επεξεργαστής, η επαλήθευση είναι δυνατόν να γίνει παράλληλα σε κάθε κομμάτι δεδομένων της ιστορίας αυτής.
Η τεχνική αυτή έχει βρει εφαρμογή σε ένα από τα πλέον ανερχόμενα δίκτυα blockchain, στο Solana, και έχει κερδίσει από τη δημοτικότητα του δικτύου.

- *Proof of Importance* (*PoI*): Πρόκειται για μια τεχνική η οποία παρουσιάστηκε το 2015 από το NEM (New Economy Movement) blockchain, η οποία αποτελεί μια παραλλαγή του PoS.
Στο PoI για τη δημιουργία του νέου block δεν παίζει ρόλο μόνο το stake ενός κόμβου. Αντιθέτως, λαμβάνονται υπόψη και άλλοι παράγοντες, όπως: οι πρόσφατες συναλλαγές του κόμβου και τα ποσά που διακινήθηκαν εκεί, το ποσό που έχει δηλωθεί για χρήση στο stake, και αν υπάρχει συνεργασία μεταξύ κόμβων για τον συνδυασμό του stake τους για τη δημιουργία του νέου block.
Με τη βαρύτητα να πέφτει σε πρόσφατες κινήσεις, μαζί με τα ποσά που διακινήθηκαν σε αυτές, το PoI ενισχύει τη συμμετοχή στο δίκτυο και καταπολεμά την αγορά και αποθήκευση κρυπτονομισμάτων. Επιπλέον, ο τρόπος με τον οποίο μαζεύεται το ποσό για το stake είναι πολύ διαφορετικός. Έτσι, κάθε 24 ώρες μπορεί να γίνει διαθέσιμο προς χρήση ως stake το 10% του ποσού που έχει στην κατοχή του ο χρήστης και το οποίο δεν έχει δηλωθεί ότι θα χρησιμοποιηθεί ως stake. Δηλαδή, αν κάποιος έχει 100 κρυπτονομίσματα, την πρώτη μέρα θα μπορεί να διαθέσει προς stake τα 10 και τη δεύτερη τα 9 (100-10 = 90 και επιτρέπεται το 10% του 90). Με τον τρόπο αυτό δεν μπορεί ένας χρήστης να διαθέσει προς stake όλα του τα χρήματα για να κερδίσει τον ρόλο του εκτιμητή για ένα μικρό χρονικό διάστημα. Λαμβάνοντας υπόψη όλα όσα προαναφέρθηκαν, το PoI δημιουργεί ένα σκορ το οποίο και χρησιμοποιείται για να βρεθεί ποιος κόμβος θα βάλει το νέο block.
Το PoI επιτυγχάνει να αντιμετωπίσει με επιτυχία το θέμα της συγκέντρωσης πλούτου για χρήση του ως stake, καθώς η διαδικασία ευνοεί τις συναλλαγές.

Βιβλιογραφία

Antonopoulos, A. M. (2017). *Mastering Bitcoin. Programming the Open Blockchain* (2nd ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., & Wood, G. (2019). *Mastering Ethereum* (1st ed.). O'Reilly Media, Inc.

Antonopoulos, A. M., Osuntokun, O. & Pickhardt, R. (2021). *Mastering the Lightning Network. A second Layer Blockchain Protocol for Instant Bitcoin Payments* (1st ed.). O'Reilly Media, Inc.

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract]. *SIGMETRICS Perform.* Evaluation Rev., 42, pp. 34-37.

Centieiro, H. (2021). *What's Proof of Elapsed Time.* Medium. May 2021. Online πηγή: https://medium.com/nerd-for-tech/whats-proof-of-elapsed-time-4f67cf3f45b3 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Cointelegraph (2022). *Proof-of-authority vs. proof-of-stake: Key differences explained.* Online πηγή: https://cointelegraph.com/blockchain-for-beginners/proof-of-authority-vs-proof-of-stake-key-differences-explained [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Frankenfield, J. (2021) *Proof of Burn.* Investopedia. Online πηγή: https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Howell, J. (2022). *Delegated Proof of Stake (DPoS) – Explained.* 101 Blockchains. Online πηγή: https://101blockchains.com/delegated-proof-of-stake-dpos/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems, 4*(3), pp. 382-401. Online πηγή: https://dl.acm.org/doi/10.1145/357172.357176

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* Online πηγή: https://bitcoin.org/bitcoin.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

TheLuWizz (2021). *What is Proof of Capacity (PoC)*. Medium. Online πηγή: https://medium.com/the-capital/what-is-proof-of-capacity-poc-443ade0e01cc [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Yakovenko, A. (2018). Proof of History: A clock for Blockchain. *Medium*. AprilOnline πηγή: https://medium.com/solana-labs/proof-of-history-a-clock-for-blockchain-cf47a61a9274 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 6
Έξυπνες Συμβάσεις (Smart Contracts)

Σύνοψη

Στο Κεφάλαιο αυτό παρουσιάζονται τεχνικές λεπτομέρειες για τις έξυπνες συμβάσεις (smart contracts) που υλοποιούνται στο δίκτυο του Ethereum (βλ. Κεφάλαιο 2). Επιπλέον, θα γίνει συγγραφή ενός smart contract στο περιβάλλον του Remix με χρήση της γλώσσας Solidity.

Κατόπιν, με τη βοήθεια του πορτοφολιού Metamask, αυτό το smart contract θα εγκατασταθεί στο δοκιμαστικό δίκτυο Goerli του Ethereum.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2 και 4. Προτείνεται η ολοκλήρωση των βημάτων στο Παράρτημα Α.

6.1 Έξυπνες συμβάσεις: Περιγραφή και τεχνικές λεπτομέρειες

Οι *έξυπνες συμβάσεις* (*smart contracts*) αποτελούν προγράμματα τα οποία αποθηκεύονται και εκτελούνται μέσα σε ένα δίκτυο blockchain. Όλοι οι κόμβοι του δικτύου έχουν τα προγράμματα αυτά, παρακολουθούν τις αλληλεπιδράσεις με αυτά και επιβεβαιώνουν τις αλλαγές στη μηχανή κατάστασης του κόσμου του δικτύου που αυτές επιφέρουν.

Στο Κεφάλαιο 2 έγινε μια πρώτη παρουσίαση των smart contracts και των χαρακτηριστικών τους χρησιμοποιώντας το δίκτυο του Ethereum. Στο Κεφάλαιο αυτό δίνονται περισσότερες τεχνικές λεπτομέρειες για τα smart contracts με την παρουσίαση και ανάλυση ενός απλού παραδείγματος. Εξηγείται η σύνταξή του και παρουσιάζεται το περιβάλλον του Remix[73], στο οποίο θα γραφεί ο κώδικας του smart contract. Στη συνέχεια περιγράφονται τα βήματα για την εγκατάστασή του σε ένα δοκιμαστικό δίκτυο του Ethereum (βλ. Πίνακα 4.5 στο Κεφάλαιο 4).

Για την ολοκλήρωση της διαδικασίας εγκατάστασης του smart contract στο δοκιμαστικό δίκτυο χρειάζεται να δημιουργηθεί ένας λογαριασμός στο web πορτοφόλι του Metamask (βλ. Παράρτημα Α).

6.1.1 Ιστορία των smart contracts

Πρώτος ο Nick Szabo (1997) παρουσίασε τόσο την ονομασία όσο και την έννοια των smart contracts σε εργασία του το 1997. Παρά την ονομασία τους, στην ουσία τα smart contracts δεν χαρακτηρίζονται από κάποια ευφυΐα (smart). Είναι απλώς προγράμματα τα οποία «αντιδρούν» (εκτελώντας συναλλαγές) όταν συγκεκριμένες συνθήκες ικανοποιούνται. Ταυτόχρονα, δεν αποτελούν νομικές συμβάσεις (contracts), καθώς δεν έχουν δημιουργηθεί (ή εγκριθεί) από κάποια νομική αρχή. Επομένως, η ονομασία smart contracts είναι ελαφρώς παραπλανητική, αλλά έχει επικρατήσει πλέον γενικώς στον χώρο του blockchain και ειδικά στο δίκτυο του Ethereum.

Μάλιστα, μια πετυχημένη μεταφορά για το τι είναι και τι κάνει ένα smart contract βρίσκεται σε αυτή την (ίδια) εργασία του Szabo (1997), στην οποία ένα smart contract παρομοιάζεται με μια αυτόματη μηχανή πώλησης αγαθών. Σε αυτές τις μηχανές επιλέγει κάποιος το προϊόν που θέλει, βάζει το απαραίτητο χρηματικό ποσό που απαιτείται και λαμβάνει το προϊόν από τη μηχανή. Η διαδικασία μπορεί να επαναληφθεί πολλές φορές, με το ίδιο αποτέλεσμα. Το ίδιο συμβαίνει και με το smart contract.

Όπως η μηχανή, έτσι και το smart contract συναλλάσσεται με όποιον επιθυμεί να δώσει το απαραίτητο ποσό για να δεχθεί τις υπηρεσίες αυτού. Επίσης, όλοι οι κόμβοι μπορούν να τρέξουν τη συναλλαγή και να φθάσουν στο ίδιο αποτέλεσμα, επομένως μπορούν και να το επιβεβαιώσουν (ή όχι).

Και στις δύο περιπτώσεις η λογική των κινήσεων βρίσκεται στη μηχανή. Στην περίπτωση του smart contract, αυτό το ίδιο αποτελεί τη μηχανή και έχει ενσωματωμένη (με τη μορφή κώδικα) την προβλεπόμενη λογική. Με

[73] Online διαδικτυακή εφαρμογή που χρησιμοποιείται ευρέως για τη συγγραφή smart contracts σε γλώσσα Solidity.

τον τρόπο αυτόν επιτυγχάνεται η αντικατάσταση του οποιουδήποτε ενδιάμεσου στη συναλλαγή, καθώς το smart contract είναι ικανό να διαχειριστεί την όποια συναλλαγή με προκαθορισμένο (και γνωστό) τρόπο και να παραγάγει ένα αποτέλεσμα που μπορεί να επιβεβαιωθεί από όλους τους κόμβους του δικτύου. Όπως ακριβώς και μια αυτόματη μηχανή πώλησης αγαθών.

6.1.2 Τα smart contracts ως λογαριασμός στο Ethereum

Όπως παρουσιάστηκε και στο Κεφάλαιο 2, στο Ethereum υπάρχουν δύο είδη λογαριασμών:

- οι λογαριασμοί απλών χρηστών (γνωστοί και ως Externally Owned Accounts), στους οποίους έχει πρόσβαση ο καθένας μέσω των ιδιωτικών κλειδιών του, και
- οι λογαριασμοί συμβάσεων, οι οποίοι αποδίδονται σε smart contracts τα οποία έχουν εγκατασταθεί στο δίκτυο του Ethereum.

Αυτοί οι τύποι λογαριασμών έχουν ορισμένα κοινά χαρακτηριστικά αλλά και διαφορές. Έτσι και οι δύο τύποι λογαριασμών μπορούν να:

- λαμβάνουν και να αποθηκεύουν ethers και tokens και
- μπορούν να συναλλάσσονται με (άλλα) smart contracts που έχουν αναπτυχθεί στο δίκτυο.

Όσον αφορά τις διαφορές τους, οι απλοί λογαριασμοί χρηστών:

- δημιουργούνται χωρίς κόστος και
- μπορούν να ξεκινήσουν μια συναλλαγή μεταφέροντας είτε ethers είτε tokens.

Από την άλλη πλευρά, οι λογαριασμοί των smart contracts:

- χρειάζονται κάποιο ποσό για να δημιουργηθούν –το ποσό αυτό δίνεται ως έξοδος αποθήκευσης του κώδικα της σύμβασης στο δίκτυο–,
- δεν μπορούν να κάνουν συναλλαγές –εξαίρεση αποτελεί η λήψη μιας συναλλαγής που ως απάντηση ζητά από το smart contract να ξεκινήσει, με τη σειρά του, μια συναλλαγή–,
- δίνουν έναυσμα για την εκτέλεση κώδικα ο οποίος μπορεί να έχει ποικίλα αποτελέσματα και να προκαλέσει μια σειρά από ενέργειες (π.χ. μεταφορά tokens, κλήση ενός δεύτερου smart contract) – το έναυσμα αυτό δίνεται με τη λήψη συναλλαγών από λογαριασμούς χρηστών σε έναν λογαριασμό smart contract.

6.1.3 Γλώσσες συγγραφής smart contracts και χρήση gas

Η συγγραφή και η εγκατάσταση ενός smart contract στο δίκτυο του Ethereum μπορεί να γίνει από κάθε χρήστη που έχει λογαριασμό στο δίκτυο. Μάλιστα, ο λογαριασμός του contract θα δημιουργηθεί με την εγκατάσταση του προγράμματος στο δίκτυο από έναν λογαριασμό χρήστη. Επομένως, αυτό που μένει είναι η γνώση μιας από τις υπάρχουσες γλώσσες προγραμματισμού για τη σύνταξη ενός smart contract.

Στο δίκτυο του Ethereum οι γλώσσες προγραμματισμού στις οποίες μπορεί να γραφεί ένα smart contract είναι η *Solidity* και η *Vyper*. Πρόκειται για δύο γλώσσες που μπορούν να χρησιμοποιηθούν και σε άλλες πλατφόρμες και έχουν ως κοινό χαρακτηριστικό ότι μπορούν να τρέξουν, αφού μεταγλωττιστούν σε bytecodes, μέσα στο περιβάλλον της EVM (βλ. Κεφάλαιο 2), που είναι ένα πολύ κρίσιμο συστατικό του δικτύου του Ethereum. Μάλιστα, η γλώσσα Solidity είναι αυτή που θα χρησιμοποιηθεί στο βιβλίο αυτό για τη συγγραφή των smart contracts.

Κοινό χαρακτηριστικό των δύο αυτών γλωσσών προγραμματισμού είναι ότι υποστηρίζουν την ιδιότητα της *πληρότητας κατά Turing* (βλ. Κεφάλαιο 2). Αυτή η ιδιότητα σημαίνει ότι είναι δυνατόν να καταλήξει ένας κόμβος στο ίδιο αποτέλεσμα κάθε φορά που θα εκτελέσει τον κώδικα του smart contract. Αυτό το χαρακτηριστικό είναι σημαντικό, γιατί επιτρέπει στους κόμβους του δικτύου να επαληθεύσουν τις αλλαγές των καταστάσεων που δημιουργούνται από τις κλήσεις των συναρτήσεων του smart contract, και τις συναλλαγές που αυτές προκαλούν.

Με την EVM να προσφέρει έναν χώρο στον οποίο τρέχουν τα smart contracts, έμφαση δίνεται στην ποιότητα του κώδικα που γράφεται σε αυτά. Ο λόγος είναι διότι, συχνά, ο κώδικας ενός smart contract μπορεί να οδηγήσει σε (εξωτερικές) επιθέσεις οι οποίες καταλήγουν στην απώλεια κρυπτονομισμάτων από χρήστες που συναλλάσσονται με το contract. Επομένως, τα λάθη στον κώδικα του smart contract δημιουργούν κενά ασφαλείας, που μπορεί να γίνουν στόχος επιθέσεων από κακόβουλους χρήστες. Μάλιστα, έχει καταγραφεί ένας

αρκετά σημαντικός αριθμός επιτυχών επιθέσεων, οι οποίες ευνοήθηκαν από τέτοια προγραμματιστικά λάθη και πέτυχαν τη μεταφορά κρυπτονομισμάτων προς όφελος των επιτιθέμενων (Sayeed et al., 2020).

Ένα ακόμα χαρακτηριστικό, που ενισχύει την ανάγκη για την αποφυγή λαθών κατά τη συγγραφή κώδικα για ένα smart contract, είναι το γεγονός ότι ο κώδικας ενός smart contract, από τη στιγμή που αυτό εγκατασταθεί στο δίκτυο (και πάρει διεύθυνση), δεν μπορεί να τροποποιηθεί. Επομένως, το οποιοδήποτε σφάλμα υπάρχει θα παραμένει. Η επιδιόρθωση γίνεται με την εγκατάσταση μιας νέας έκδοσης του smart contract (σε άλλη διεύθυνση) και με αλλαγή όλων των κλήσεων του smart contract προς τη νέα αυτή διεύθυνση. Επιλεκτικά μπορεί να γίνει χρήση της συνάρτησης *selfdestruct()*, αν και ακόμα και η συνάρτηση αυτή έχει χρησιμοποιηθεί για να γίνει επίθεση και να υποκλαπούν κρυπτονομίσματα.

Ένα άλλο σημαντικό θέμα που σχετίζεται με την εκτέλεση των smart contracts στο Ethereum είναι και η ανάγκη κατανάλωσης gas. Η χρήση του gas (βλ. Κεφάλαιο 2) βασίζεται στο γεγονός ότι για να εκτελεστεί το smart contract θα πρέπει να δαπανηθούν πόροι του δικτύου, οπότε προβλέπεται η υπηρεσία αυτή να χρεώνεται, αρχικά με ένα μικρό ποσό σε κρυπτονομίσματα. Βέβαια, το gas είναι άμεσα συνδεδεμένο με την τιμή του κρυπτονομίσματος (του ether στο Ethereum), επομένως άνοδος στην τιμή αυτού μεταφράζεται και άνοδος στο gas, κάνοντας έτσι πιο ακριβή την εκτέλεση του smart contract.

Εκτός όμως από τη χρήση του gas ως ενός είδους ανταμοιβής προς το δίκτυο για την εργασία του, αυτό χρησιμοποιήθηκε και για να εξασφαλίσει ότι όλα τα smart contracts θα τερματίσουν τη λειτουργία τους. Το τελευταίο αποτελεί ζητούμενο στο Ethereum, καθώς, λόγω της πληρότητας κατά Turing, πρέπει όλοι οι κόμβοι να ολοκληρώσουν το πρόγραμμα που εκτελούν και να φθάσουν και σε κοινό αποτέλεσμα. Όμως, σε περίπτωση, για παράδειγμα, σφάλματος στον κώδικα, το οποίο και οδηγεί σε έναν ατέρμονα βρόχο, το πρόγραμμα δεν θα μπορούσε να καταλήξει. Σε κάθε άλλη περίπτωση, θα μπορούσε κάποιος να «σκοτώσει» τη διεργασία του προγράμματος, αλλά αυτό πλέον δεν βρίσκεται σε έναν υπολογιστή αλλά σε χιλιάδες σε ολόκληρο τον κόσμο. Επομένως, αυτή η επιλογή δεν υπάρχει πλέον και πρέπει να αντικατασταθεί από μια άλλη λύση.

Αυτή τη λύση προσφέρει η χρήση του gas. Όταν ένα smart contract, λόγω κακού κώδικα, εγκλωβιστεί σε έναν ατέρμονα βρόχο, καταναλώνει όλο το διαθέσιμο gas, χωρίς να μπορέσει να ξεφύγει από αυτόν. Μόλις καταναλωθεί όλο το gas, τότε ακυρώνονται όλες οι αλλαγές κατάστασης που έχει επιφέρει το smart contract και επανέρχονται στην προ εκτέλεσης κατάστασή τους.

Επομένως, ο ρόλος του gas είναι διττός: και ως ανταμοιβή του δικτύου για τη χρήση των πόρων του και ως εξασφάλιση ότι δεν θα εγκλωβιστεί το smart contract από κακής ποιότητας κώδικα.

Επιπρόσθετα, τα smart contracts στο Ethereum είναι δημόσια. Δηλαδή μπορεί ο οποιοσδήποτε γνωρίζει τη διεύθυνσή τους να ξεκινήσει την αλληλεπίδραση μαζί τους. Από μια άποψη, λοιπόν, τα smart contracts είναι ένα είδος δημόσιας (ή ανοικτής) διεπαφής προγραμματισμού (Application Programming Interface, API) του χρήστη με το δίκτυο του blockchain.

Όμως έχουν και αυτά τους περιορισμούς τους στο τι μπορούν να κάνουν. Έτσι, για παράδειγμα, δεν μπορούν να γνωρίζουν για θέματα που αφορούν τον «πραγματικό κόσμο» και τα οποία είναι εκτός του blockchain. Μάλιστα, αυτό συμβαίνει επίτηδες, καθώς η δυνατότητα κλήσης από τα smart contracts εξωτερικών (προς το blockchain) πηγών είναι δυνατό να έχει απροσδιόριστη επίπτωση στην επίτευξη της συναίνεσης στο κατανεμημένο δίκτυο. Για παροχή πληροφοριών από εξωτερικές πηγές, όποτε αυτό είναι αναγκαίο, χρησιμοποιούνται ειδικοί μηχανισμοί που ονομάζονται oracles (βλ. Κεφάλαιο 2).

Τέλος, ένας ακόμα περιορισμός των smart contracts στο Ethereum είναι το μέγιστο μέγεθος που επιτρέπεται να έχουν και το οποίο δεν πρέπει να ξεπερνά τα 24KB, για να μην κινδυνεύει να μείνει από gas κατά την εγκατάσταση και λειτουργία του.

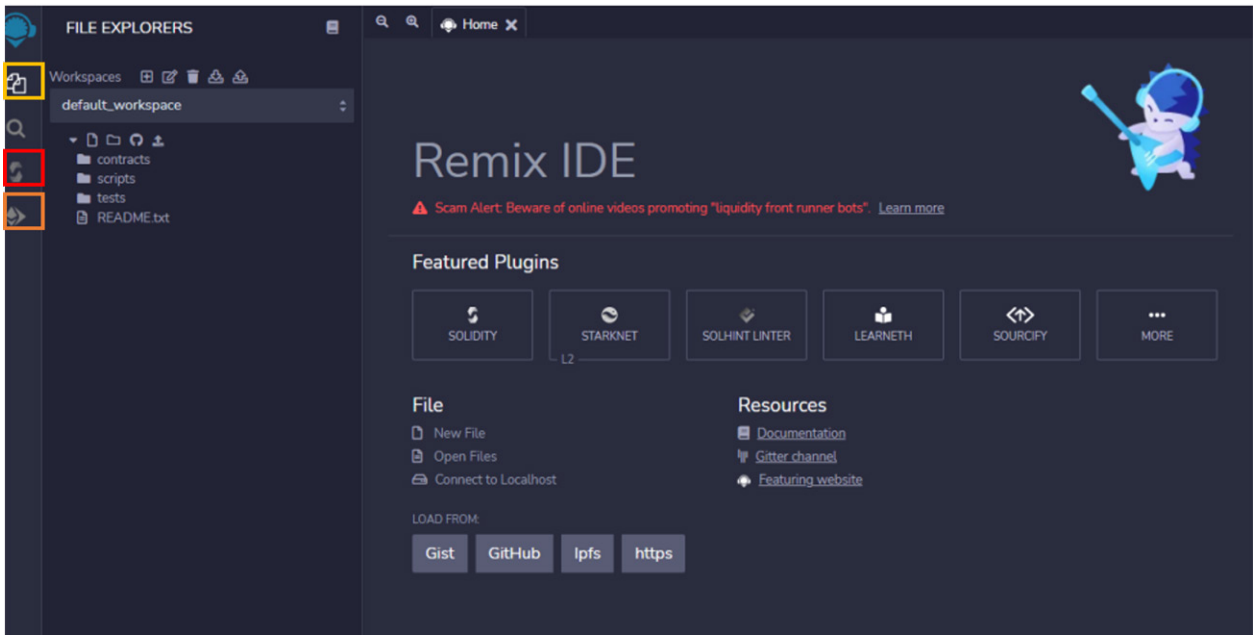
6.2 Το πρώτο smart contract

Έχοντας καλύψει το θεωρητικό μέρος που αφορά τα smart contracts, σειρά έχει να καλυφθεί και το πρακτικό.

6.2.1 Το περιβάλλον του Remix

Για το πρακτικό μέρος χρειάζεται η χρήση ενός περιβάλλοντος στο οποίο είναι δυνατή η συγγραφή και μεταγλώττιση των προγραμμάτων των smart contracts. Ένα τέτοιο περιβάλλον, ιδιαίτερα γνωστό στον κόσμο των προγραμματιστών στο Ethereum, αποτελεί το Remix.[74]

[74] Είσοδος μέσω του online συνδέσμου: https://remix.ethereum.org



Εικόνα 6.1 *Το περιβάλλον του Remix για τη σύνταξη, μεταγλώττιση και δοκιμή smart contracts στη γλώσσα Solidity.*

Στην **Εικόνα 6.1** φαίνεται η βασική σελίδα της εφαρμογής, η οποία αποτελείται από τα εξής στοιχεία, που φαίνονται στο μενού στα αριστερά και τα οποία χρησιμοποιούνται στο παράδειγμα σύνταξης ενός smart contract που ακολουθεί.

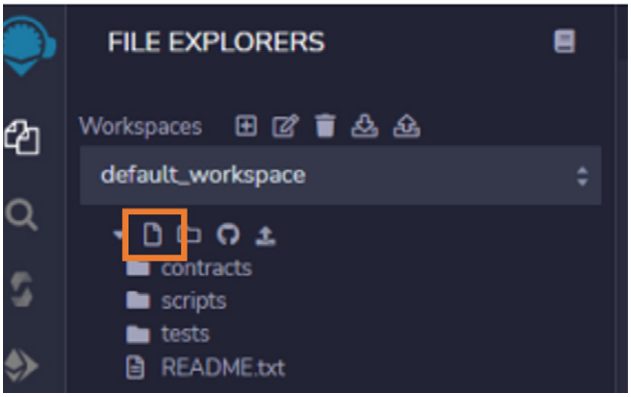
Αυτά τα στοιχεία είναι:

- Ο file explorer (μέσα στο κίτρινο πλαίσιο στην Εικόνα 6.1), στον οποίο γράφονται τα προγράμματα.
- Ο μεταγλωττιστής της γλώσσας Solidity (μέσα στο κόκκινο πλαίσιο). Χρησιμοποιείται μόλις ολοκληρωθεί η σύνταξη του συμβολαίου για τη μεταγλώττισή του.
- Το μενού για την Εγκατάσταση του contract και την Εκτέλεση συναλλαγών (πορτοκαλί πλαίσιο), το οποίο χρησιμοποιείται μετά τη συγγραφή και μεταγλώττιση για τη δοκιμή του contract.

6.2.2 Η συγγραφή και μεταγλώττιση του κώδικα

Το smart contract που θα αναλυθεί ως παράδειγμα (Guibert, 2022) προσομοιάζει τη λειτουργία μιας εφαρμογής ανταλλαγής μηνυμάτων.

Αποτελείται από δύο συναρτήσεις: η πρώτη είναι η *sendMessage()* και χρησιμοποιείται για την αποστολή ενός μηνύματος που αποθηκεύεται στο δίκτυο του blockchain και η δεύτερη, η *getMessages()*, βρίσκει τα μηνύματα και τα εμφανίζει.



Εικόνα 6.2 *Δημιουργώντας νέο contract.*

Για τη σύνταξη του προγράμματος επιλέγεται πάνω από τον φάκελο contracts το αριστερό εικονίδιο δίπλα στο βελάκι, όπως φαίνεται στην **Εικόνα 6.2**. Αυτό είναι το εικονίδιο δημιουργίας νέου smart contract.

Δίνεται κατόπιν το όνομα *ChatExample.sol* στο contract. Η κατάληξη *.sol* υποδηλώνει τη γλώσσα σύνταξης του συμβολαίου, η οποία και θα είναι η γλώσσα Solidity. Ο καινούργιος χώρος εργασίας που δημιουργείται χρησιμοποιείται για τη συγγραφή του κώδικα του smart contract. Αυτός φαίνεται στην **Εικόνα 6.3**.

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.12;
contract ChatExample {

 struct Message {
 address waver;
 string content;
 uint timestamp;
 }

 Message[] messages;

 function sendMessage(string calldata _content) public {
 messages.push(Message(msg.sender, _content, block.timestamp));
 }

 function getMessages() view public returns (Message[] memory) {
 return messages;
 }

}
```

Εικόνα 6.3 *Το smart contract του παραδείγματος (ChatExample.sol) γραμμένο σε γλώσσα Solidity.*

Στη συνέχεια αναλύεται ο κώδικας της Εικόνας 6.3, ξεκινώντας από τις τρεις πρώτες γραμμές:

- Η 1η γραμμή αποτελεί ένα σχόλιο στο οποίο περιέχεται μια δήλωση για την άδεια χρήσης του προγράμματος. Πρόκειται για μια προαιρετική εντολή, η οποία αν λείπει δημιουργεί μια προειδοποίηση από τον μεταγλωττιστή (χωρίς να επηρεάζει την εκτέλεση του contract)[75].

[75] Αναλυτικά η λίστα αδειών της SPSX βρίσκεται στον online σύνδεσμο https://spdx.org/licenses/. Εκεί εξηγούνται αναλυτικά όλες οι τιμές.

- Η 2η γραμμή περιέχει την κωδική λέξη *pragma*, η οποία χρησιμοποιείται για να δηλώσει την έκδοση του μεταγλωττιστή που απαιτείται για τη μεταγλώττιση του contract (προσοχή να έχει γίνει η σωστή επιλογή στο Remix). Το συγκεκριμένο contract χρειάζεται την έκδοση 0.8.12.
- Η 3η γραμμή περιέχει τη δήλωση του contract, με χρήση της κωδικής λέξης *contract* και την απόδοση του ονόματος αυτού.

Στη συνέχεια, το βάρος πέφτει στη λειτουργία του προγράμματος. Όπως έχει αναφερθεί, το πρόγραμμα αυτό αποθηκεύει μηνύματα στο blockchain και κατόπιν τα βρίσκει και τα εμφανίζει. Η πληροφορία που αποθηκεύεται όμως δεν είναι μόνο το σώμα του μηνύματος, αλλά και η διεύθυνση του αποστολέα μαζί με τη χρονοσφραγίδα που υποδεικνύει πότε έγινε η αποστολή του εν λόγω μηνύματος.

Όπως φαίνεται και στην **Εικόνα 6.4**, όλη αυτή η πληροφορία, για να αποδοθεί, θα χρειαστεί να χρησιμοποιηθεί ένα struct που έχει το όνομα Message και αποτελείται από:

- μια μεταβλητή τύπου address για την αποθήκευση της διεύθυνσης του αποστολέα,
- μια μεταβλητή τύπου string για την αποθήκευση του σώματος του μηνύματος,
- μια μεταβλητή τύπου uint (unsigned integer) για την αποθήκευση της χρονοσφραγίδας.

```
struct Message {
address waver;
string content;
uint timestamp;
}

Message[] messages;
```

Εικόνα 6.4 *Το τμήμα δήλωσης του struct στο contract του παραδείγματος.*

Η τελευταία γραμμή στον κώδικα της Εικόνας 6.4 δημιουργεί μια μεταβλητή κατάστασης (state variable[76]) τύπου Message[], δηλαδή είναι ένας πίνακας που περιέχει αντικείμενα του struct Message.

Τώρα ήρθε η ώρα να προστεθούν στον κώδικα και οι δύο συναρτήσεις που προαναφέρθηκαν. Στην Εικόνα 6.3 υπάρχει ο κώδικας των συναρτήσεων αυτών για αναφορά.

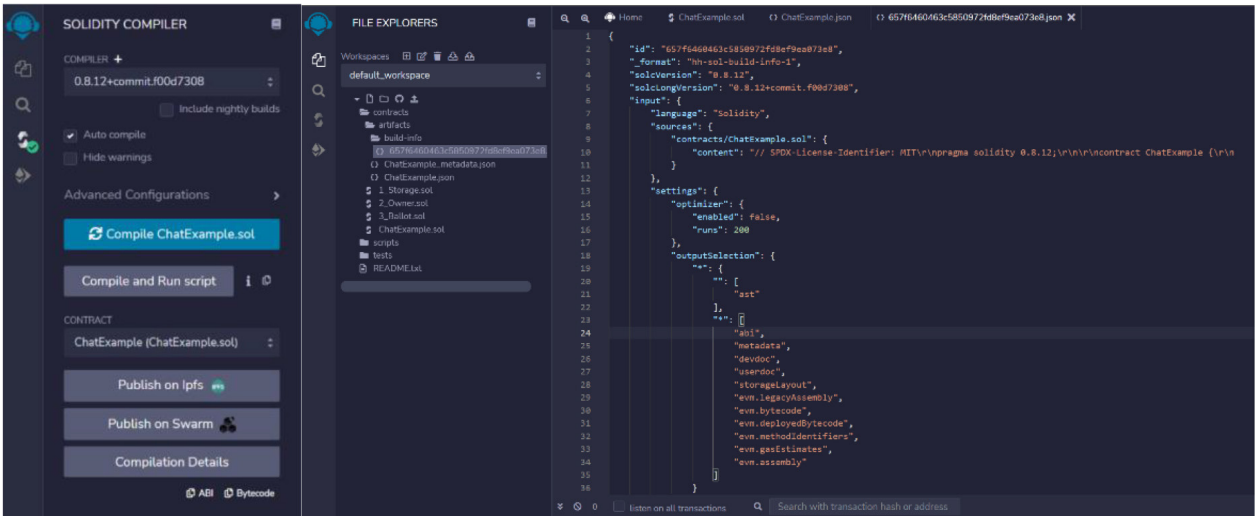
- Συνάρτηση *sendMessage()*: Η συνάρτηση αυτή δέχεται ως παράμετρο μια μεταβλητή τύπου string με όνομα *_content*. Η κωδική λέξη *calldata*, που ακολουθεί τη λέξη string, υποδεικνύει πού αποθηκεύεται το περιεχόμενο της μεταβλητής.[77]
  Στο σώμα της η συνάρτηση τοποθετεί τη διεύθυνση του αποστολέα (msg.sender), το σώμα του μηνύματος (_content) και την τιμή της χρονοσφραγίδας (block.timestamp) μέσα στη μεταβλητή κατάστασης που δημιουργήθηκε προηγουμένως (*Message[]*).
  Οι λέξεις *msg* και *block* είναι *παγκόσμιες μεταβλητές* (*global variables*) και η μεν πρώτη έχει πρόσβαση σε πληροφορίες όπως είναι η διεύθυνση του χρήστη που την καλεί, ενώ η δεύτερη γνωρίζει χαρακτηριστικά του block όπως είναι ο χρόνος προσθήκης του στο ledger.
- Συνάρτηση *getMessages()*: Αν και ως συνάρτηση δεν έχει την πολυπλοκότητα της πρώτης, εντούτοις παρουσιάζει ορισμένα χαρακτηριστικά που συναντώνται συχνά σε προγράμματα που γράφονται στη Solidity. Έτσι, η λέξη *view*, που ακολουθεί τη δήλωση του ονόματος της συνάρτησης, αποτελεί *τροποποιητή* (*moderator*) και δηλώνει ότι οι μεταβλητές κατάστασης δεν μπορούν να μεταβληθούν μέσα στη συνάρτηση αυτή (για αυτό και ορίζεται ως τύπου view). Επιπλέον, το γεγονός ότι επιτρέπει μόνο την ανάγνωση γλιτώνει τον χρήστη από την υποχρέωση να πληρώσει gas για να την εκτελέσει, καθώς η εκτέλεσή της δεν επιφέρει κάποια συναλλαγή στο δίκτυο.
  Η κωδική λέξη *public* υποδηλώνει ότι είναι μια συνάρτηση που μπορεί να κληθεί (και) εκτός του contract που παρουσιάζεται, όπως, για παράδειγμα, από κάποιο άλλο contract. Η εντολή *returns* επιστρέφει τα μηνύματα που έχουν αποθηκευτεί στη μεταβλητή messages.

[76] Αυτό το είδος μεταβλητής αποθηκεύει την τιμή της μόνιμα στο contract.

[77] Η *calldata* λειτουργεί σαν μνήμη στη Solidity και υποδεικνύει μια προσωρινή θέση για τα δεδομένα, τα οποία δεν μπορούν να αλλοιωθούν. Τα δεδομένα είναι προσβάσιμα μόνο μέσα από τη συνάρτηση στην οποία δηλώνονται.

Τέλος, η εντολή memory υποδηλώνει ότι η απάντηση θα επιστρέψει σε έναν προσωρινό χώρο μνήμης και δεν θα αποθηκευτεί κάπου μόνιμα (απαιτώντας στη δεύτερη περίπτωση και περισσότερο gas να πληρωθεί, ενώ τώρα μπορεί να παραμείνει μικρό ως έξοδο).

Το επόμενο βήμα για την ολοκλήρωση της σύνταξης του προγράμματος είναι η μεταγλώττισή του και ο εντοπισμός τυχόν σφαλμάτων στο Remix. Για τη μεταγλώττιση επιλέγετε το εικονίδιο μέσα στο κόκκινο πλαίσιο (Εικόνα 6.1). Εκεί μπορεί κάποιος να επιλέξει το κουτάκι που λέει Auto compile ζητώντας από το Remix να βρει και να φορτώσει τον κατάλληλο compiler (σύμφωνα με την έκδοσή του, που έχει αποδοθεί στη 2η γραμμή με την κωδική λέξη pragma).



Εικόνα 6.5 *Η μεταγλώττιση του smart contract.*

Η έκδοση του compiler επιβεβαιώνεται στο πάνω αριστερά μέρος της οθόνης, όπως φαίνεται και στην Εικόνα 6.5 (αριστερά). Επιλέγοντας το Auto compile, και στην περίπτωση που δεν έχουν γίνει λάθη κατά τη συγγραφή του contract, ένα πράσινο σήμα θα γνωστοποιήσει την ορθή ολοκλήρωση της διαδικασίας της μεταγλώττισης.

Εναλλακτικά, είναι δυνατή η επιλογή της έκδοσης του compiler που επιθυμεί ο καθένας και κατόπιν πατώντας στο *Compile ChatExample.sol* να προχωρήσει στη μεταγλώττιση του κώδικα.

Με την ολοκλήρωση μιας μεταγλώττισης χωρίς σφάλματα, φαίνεται πως στη διαδρομή artifacts/build-info του file explorer έχουν προστεθεί αρχεία (metadata, json) σχετικά με το smart contract που δημιουργήθηκε. Το αρχείο *ChatExample.json* περιέχει τη σύνδεση με τις βιβλιοθήκες και το bytecode. Το αρχείο *ChatExample_metadata.json* περιέχει τα metadata της μεταγλώττισης του contract.

6.2.3 Εγκατάσταση του smart contract στο blockchain δίκτυο Goerli

Για το επόμενο βήμα θα πρέπει να γίνουν δοκιμές στον κώδικα για να διορθωθούν τα όποια λάθη. Επειδή η εγκατάστασή του στο κεντρικό δίκτυο του Ethereum (Mainnet) απαιτεί να δοθούν χρήματα και καθώς τα όποια λάθη δεν θα μπορούν να διορθωθούν αργότερα, προτείνεται η εγκατάστασή του σε ένα από τα δοκιμαστικά δίκτυα του Ethereum.

Συνοδευτικά χρειάζεται και χρήση του Remix για την αλληλεπίδραση και δοκιμές με το smart contract.

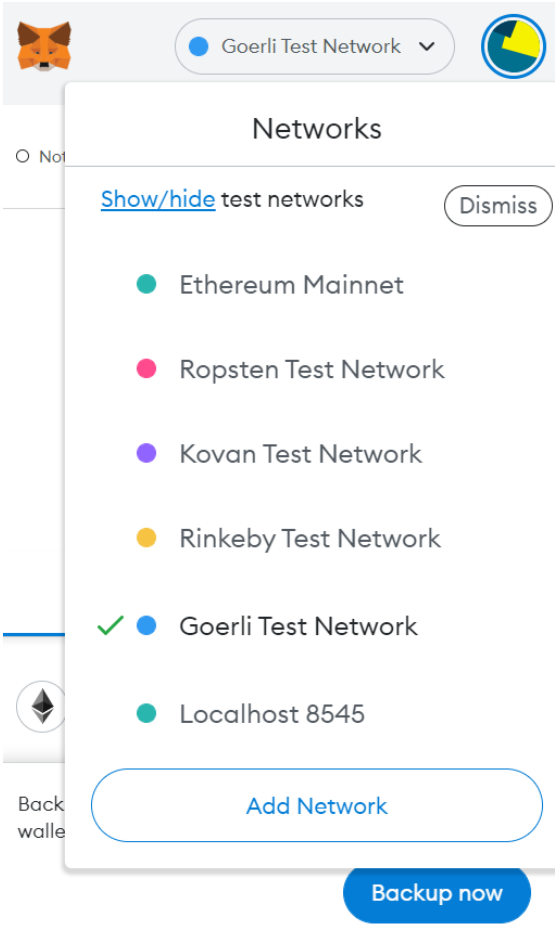
Για να μπορέσει όμως να γίνει η εγκατάσταση του contract είτε στο κεντρικό δίκτυο του Ethereum είτε σε κάποιο δοκιμαστικό, θα χρειαστεί να προηγηθεί η εγκατάσταση ενός πορτοφολιού. Στην περίπτωσή μας χρησιμοποιείται το Metamask.

Στο Παράρτημα Α υπάρχουν αναλυτικές οδηγίες για την εγκατάσταση του Metamask (σε browser της επιλογής σας) και για τη δημιουργία ενός λογαριασμού σε αυτό.

Στη συνέχεια θεωρείται ότι ο αναγνώστης έχει ακολουθήσει τα βήματα και έχει προχωρήσει στην εγκατάσταση του Metamask.

Επόμενο βήμα της εγκατάστασης του Metamask είναι η σύνδεσή του με το Remix, για να χρησιμοποιηθεί ως μέσο της εγκατάστασης του smart contract στο δίκτυο που επιλέγεται. Η πρόσβαση στο Remix θα πρέπει να γίνεται με χρήση του browser στον οποίο και έχει εγκατασταθεί το Metamask.

Τώρα, για να συνδεθούν αυτές οι δύο εφαρμογές, πρέπει πρώτα να επιβεβαιωθεί στο Remix ότι έχει γίνει compile το contract *ChatExample.sol* και κατόπιν στο Metamask ο λογαριασμός που έχει φτιάξει ο κάθε χρήστης να συνδεθεί στο δίκτυο Goerli. Το δίκτυο Goerli είναι το δοκιμαστικό δίκτυο του Ethereum που χρησιμοποιείται για την εγκατάσταση του smart contract στο παράδειγμα που παρουσιάζεται εδώ. Η **Εικόνα 6.6** δείχνει ότι αλλαγή του δικτύου στο πορτοφόλι του Metamask πραγματοποιείται πατώντας στο βελάκι στην κορυφή και από το μενού που αναπτύσσεται επιλέξετε το δίκτυο Goerli.



Εικόνα 6.6 *Επιλογή του Goerli δικτύου στο Metamask.*

Όμως, παρόλο που το δίκτυο είναι δοκιμαστικό, χρειάζονται και εδώ κρυπτονομίσματα για την εγκατάσταση του smart contract. Για την ακρίβεια, χρειάζεται να αποκτηθούν Goerli ETH! Η βασική διαφορά είναι ότι τα κρυπτονομίσματα αυτά δεν έχουν κάποια αξία στον πραγματικό κόσμο και μπορεί κάποιος να τα αποκτήσει δωρεάν. Υπάρχουν, μάλιστα, ειδικά προγράμματα (ονομάζονται faucets) τα οποία αναλαμβάνουν να στείλουν μικρά ποσά σε όποια διεύθυνση του δικτύου τούς ζητήσει.

Συνήθως, τα faucets στέλνουν ένα ποσό σταθερό μία φορά την ημέρα, δηλαδή θα αρνηθούν να στείλουν αν κάποιος επαναλάβει την αίτησή του για Goerli ETH σε λιγότερο από 24 ώρες από την τελευταία του. Η επιβεβαίωση μπορεί να γίνει πολύ απλά από το faucet, με αναζήτηση της τελευταίας συναλλαγής της διεύθυνσης στο δίκτυο και με τον έλεγχο της χρονοσφραγίδας αυτής.

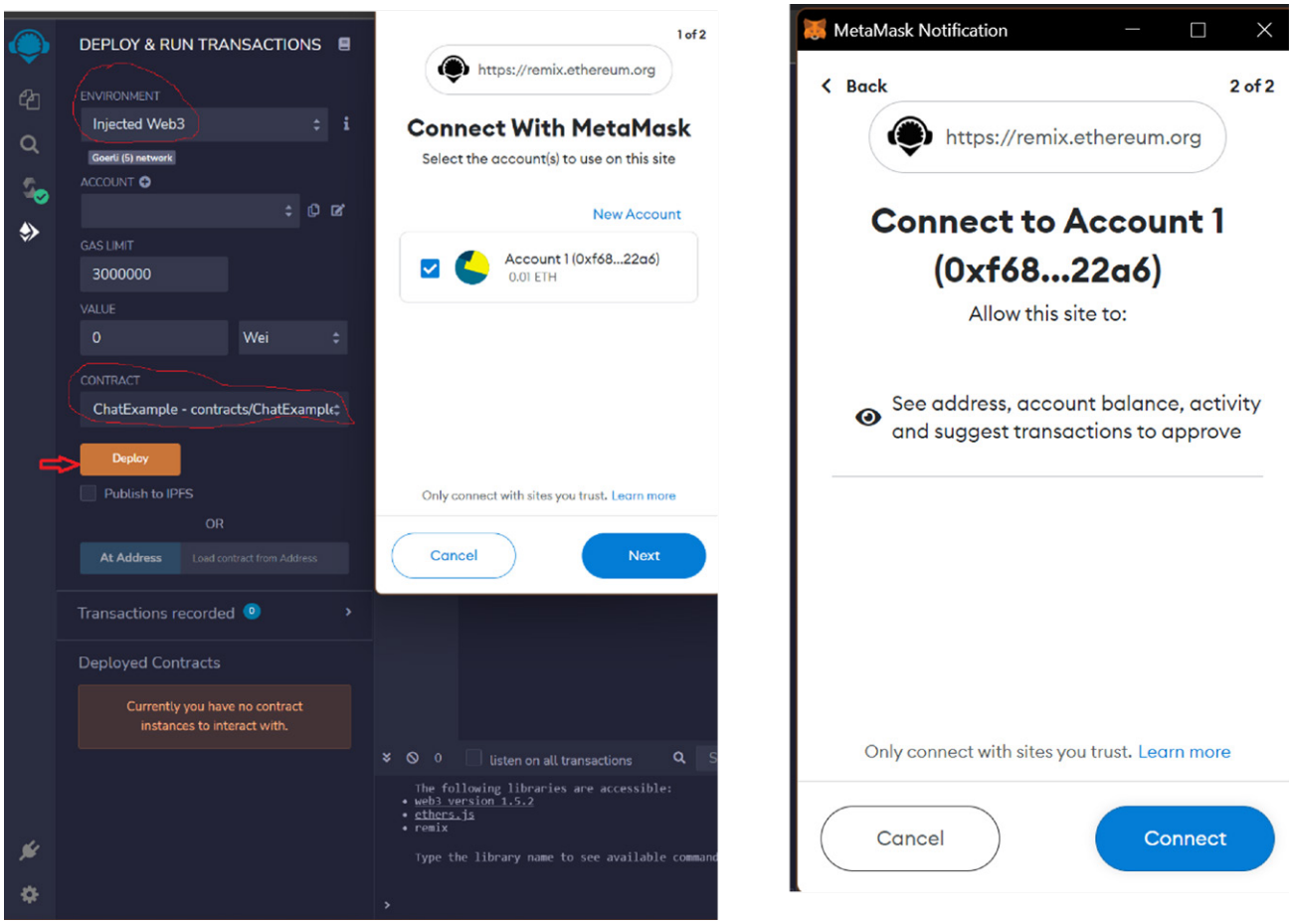
Στην περίπτωση του παραδείγματος μπορεί να χρησιμοποιηθεί το faucet για τη λήψη 0,01 Goerli ETH στη διεύθυνση: https://goerlifaucet.com/[78]. Εναλλακτικά, είναι δυνατό μέσω Metamask να επιλέξει κάποιος *Buy* και μετά (αν κατέβει πιο κάτω) την επιλογή *Test Faucet*, που θα τον πάει σε ένα άλλο faucet του ίδιου δικτύου για να ζητήσει τα Goerli ETH. Τα 0.01 Goerli ETH είναι αρκετά για την εγκατάσταση και εκτέλεση του smart contract του παραδείγματος.

Το επόμενο βήμα είναι η ολοκλήρωση της σύνδεσης του Metamask με το Remix. Το Remix δημιουργεί ένα Web 3.0 αντικείμενο στην ιστοσελίδα, το οποίο μπορούμε να ζητήσουμε να συνδεθεί με το Remix για να αλληλεπιδράσει μαζί του.

[78] Θα χρειαστεί να δημιουργηθεί ένας λογαριασμός στο Alchemy για να ολοκληρωθεί η αποστολή των Goerli ETH.

Για να γίνει η ρύθμιση αυτή στο Remix, πρέπει να επιλεχθεί από το αριστερό μενού η επιλογή *Deploy and Run Transactions* (πορτοκαλί πλαίσιο στην Εικόνα 6.1). Κατόπιν, αφού γίνει επιβεβαίωση ότι έχει ολοκληρωθεί επιτυχώς η μεταγλώττιση του smart contract του παραδείγματος, θα πρέπει στο μενού environment να αλλάξει η επιλογή από JavaScript VM (London) σε Injected Web3. Μόλις γίνει η επιλογή, δημιουργείται ένα παράθυρο στο οποίο ζητείται η επιβεβαίωση της σύνδεσης με το Metamask. Στο παράθυρο επιλέγεται ο ίδιος λογαριασμός στο Metamask που χρησιμοποιήθηκε για την επικοινωνία με το faucet. Δηλαδή, ο λογαριασμός που έχει δεχθεί τα Goerli ETH.

Στην **Εικόνα 6.7** φαίνεται το βήμα αυτό για τη σύνδεση των δύο εφαρμογών. Επίσης, φαίνεται το όνομα του smart contract το οποίο εγκαθίσταται στο μενού *Contract* καθώς και η επιλογή *Injected Web3* στο μενού *Environment*. Τέλος, φαίνεται και το κουμπί *Deploy*, που θα πρέπει να πατηθεί αμέσως μετά για την εγκατάσταση του smart contract.

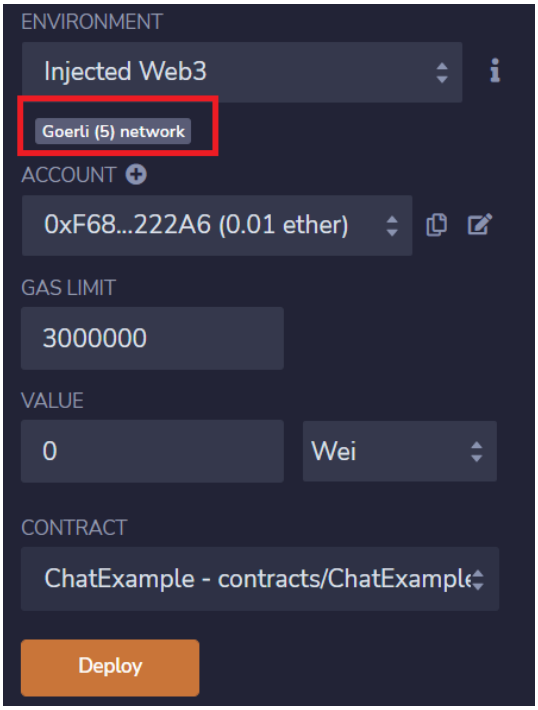


Εικόνα 6.7 *Η σύνδεση του Remix με τη διεύθυνση στο Goerli δίκτυο μέσω του Metamask.*

Πλέον, η σύνδεση του Remix με το δίκτυο Goerli μέσω του Metamask έχει ολοκληρωθεί. Για επιβεβαίωση μπορεί να γίνει έλεγχος στο μενού *Deploy and Run Transactions* στο Remix, αν ακριβώς κάτω από την επιλογή *Injected Web3* στο μενού *Environment* αναφέρεται το δίκτυο Goerli, όπως φαίνεται στην **Εικόνα 6.8** μέσα στο κόκκινο πλαίσιο.

Το τελευταίο βήμα, τώρα, είναι να ολοκληρωθεί ο έλεγχος για την εγκατάσταση. Οι τιμές στο μενού *Account* (που έχει ανανεωθεί από το Metamask) σχετικά με τα *Gas Limit* και *Value* παραμένουν στις προκαθορισμένες τιμές τους, όπως φαίνονται στην Εικόνα 6.8. Τώρα ήρθε η ώρα για να πατηθεί το κουμπί *Deploy*.

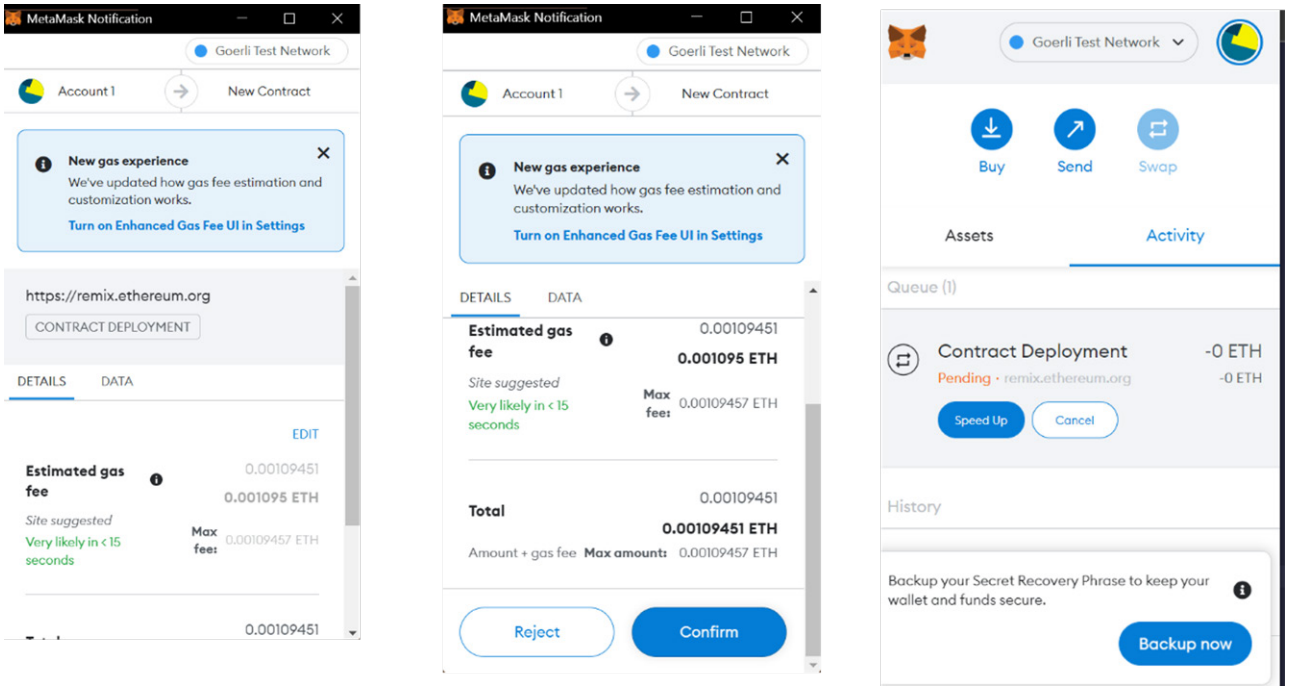
Μόλις πατηθεί, θα ανοίξει το Metamask και θα ζητήσει την επιβεβαίωση της συναλλαγής, η οποία θα κοστίσει 0,00109451 ETH. Κοιτώντας προσεκτικά, θα διακρίνει κάποιος ότι το ποσό που χρεώνεται η συναλλαγή αφορά αποκλειστικά και μόνο τα transaction fees. Δηλαδή, δεν υπάρχει κάποια επιπλέον χρέωση από το δίκτυο για την αποθήκευση του συμβολαίου, παρά μόνο τα χρήματα επιβράβευσης στους κόμβους που συμμετέχουν στη συναίνεση του δικτύου.



Εικόνα 6.8 *Η επιβεβαίωση της σύνδεσης με το δίκτυο Goerli στο Remix.*

Με την αποδοχή στα μηνύματα που εμφανίζονται στο Metamask, εκτελείται η συναλλαγή και θα πρέπει κάποιος να περιμένει λίγα δευτερόλεπτα[79] προτού του έρθει μήνυμα στην κονσόλα του Remix για την ολοκλήρωση της συναλλαγής.

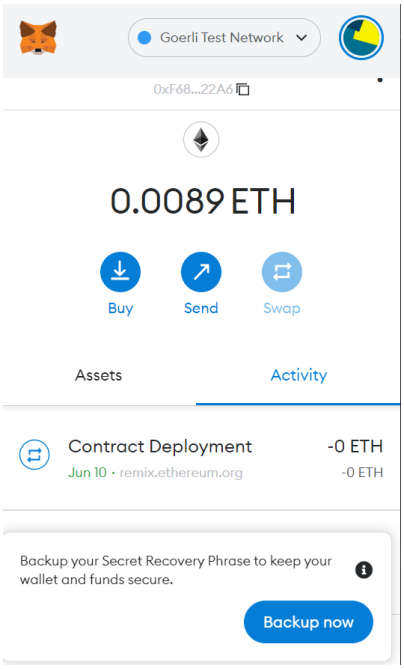
Στην **Εικόνα 6.9** απεικονίζεται μια συλλογή από μηνύματα που λαμβάνονται για την ολοκλήρωση της συναλλαγής στο Metamask.



Εικόνα 6.9 *Τα μηνύματα στο Metamask για την εγκατάσταση του smart contract μετά το πάτημα του κουμπιού Deploy στο Remix.*

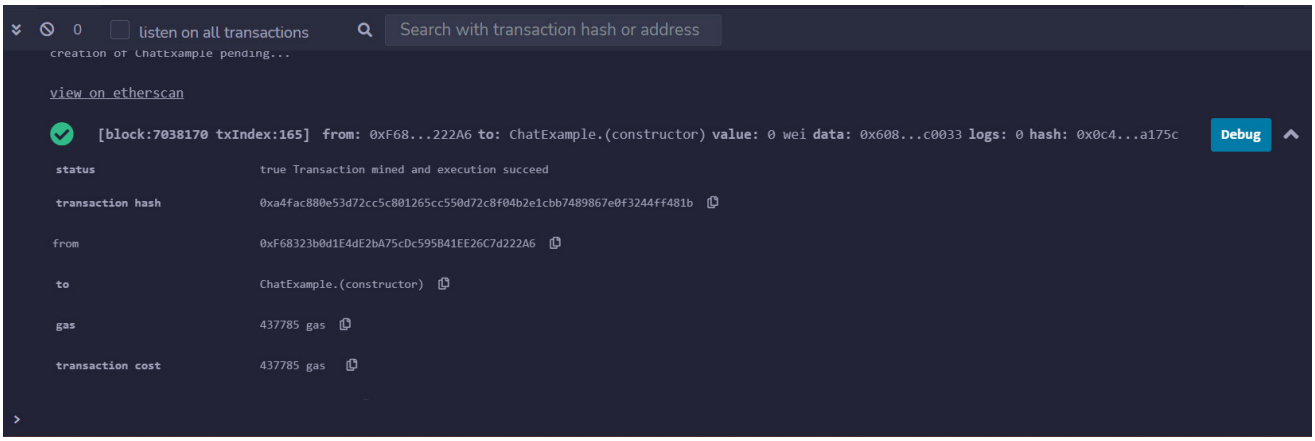
[79] Ο χρόνος διαφέρει ανάλογα με τις τρέχουσες συνθήκες του δικτύου κατά την εκτέλεση της συναλλαγής.

Μετά το πέρας της συναλλαγής, το ποσό στον λογαριασμό στο Metamask θα ανανεωθεί και από 0,01 ETH θα είναι πλέον ίσο με 0,0089, όπως φαίνεται και στην **Εικόνα 6.10**. Η επιβεβαίωση για την ολοκλήρωση της συναλλαγής (και την ένταξη της σε block) λαμβάνεται και στην κονσόλα του Remix, όπου εμφανίζονται οι λεπτομέρειες της συναλλαγής.



Εικόνα 6.10 *Η ανανέωση του υπολοίπου στο Metamask μετά την ολοκλήρωση της συναλλαγής.*

Στην Εικόνα **6.11** φαίνεται το μήνυμα αυτό.

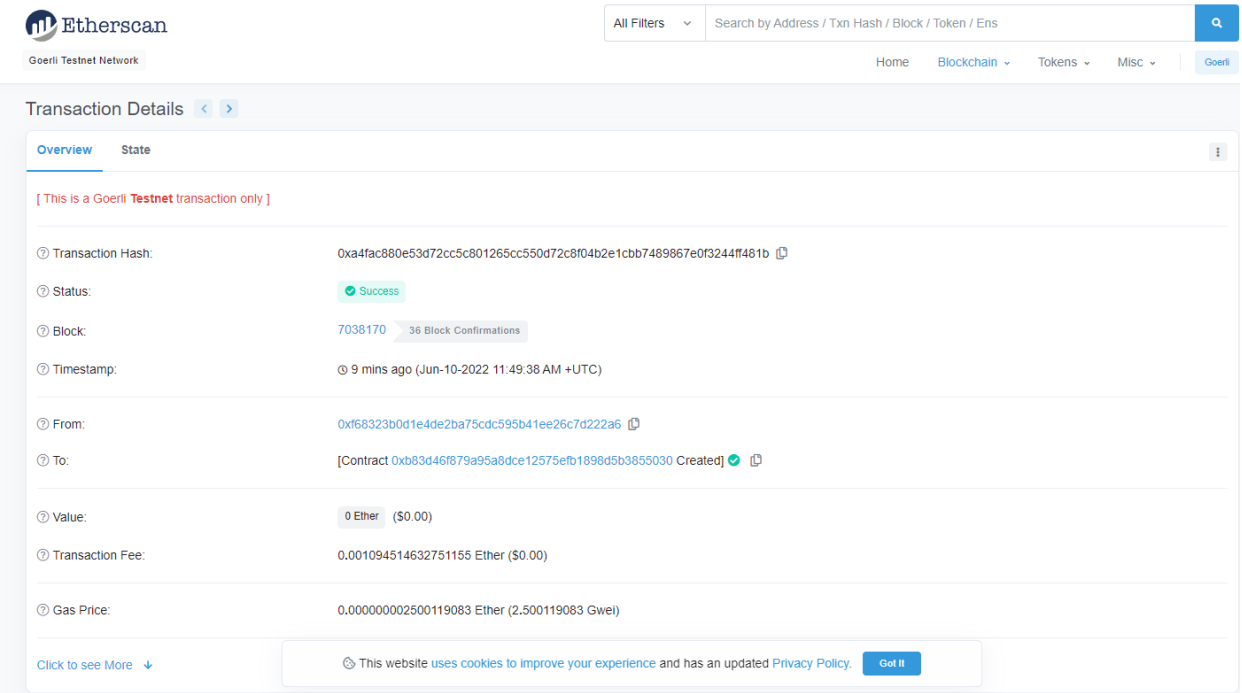


Εικόνα 6.11 *Η επιβεβαίωση της ολοκλήρωσης της εγκατάστασης του συμβολαίου στο Remix.*

Υπάρχει ένας ακόμα τρόπος να βρει κανείς περισσότερες λεπτομέρειες για τη συναλλαγή, και αυτός είναι να πατήσει στο link που φαίνεται στην αρχή της Εικόνας 6.11, που καλεί τον χρήστη να δει τις λεπτομέρειες στο block explorer (λέει *view on etherscan*). Έτσι, στην **Εικόνα 6.12** φαίνονται οι λεπτομέρειες της συναλλαγής στην εφαρμογή του etherscan.[80]

Μια σημαντική πληροφορία είναι η διεύθυνση του contract, η οποία, όπως φαίνεται, είναι η *0xb83d46f879a95a8dce12575efb1898d5b3855030*. Ακόμα, επιβεβαιώνεται και η διεύθυνση του δημιουργού ότι είναι η ίδια με αυτή στο Metamask.

[80] Ο Online Σύνδεσμος για την εύρεση των λεπτομερειών της συναλλαγής στο etherscan είναι στο: https://goerli.etherscan.io/tx/0xa4fac880e53d72cc5c801265cc550d72c8f04b2e1cbb7489867e0f3244ff481b

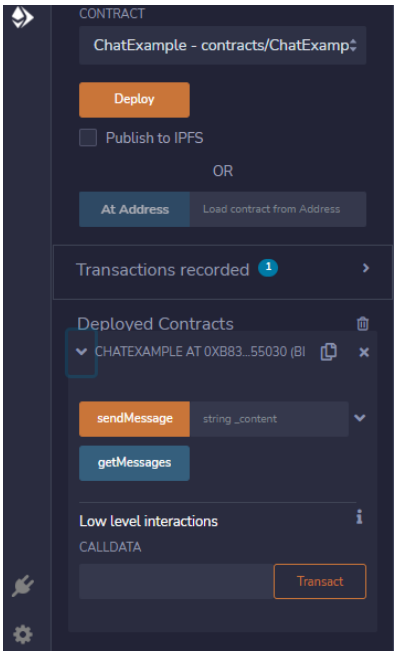


Εικόνα 6.12 *Οι λεπτομέρειες της συναλλαγής όπως φαίνονται στο etherscan.*

6.2.4 Αλληλεπίδραση με το smart contract

Με την επιτυχή εγκατάσταση του smart contract στο δίκτυο Goerli, είναι δυνατή η αλληλεπίδραση μαζί του μέσω του Remix. Αργότερα, στο Κεφάλαιο 8, θα παρουσιαστεί πώς μπορεί να δημιουργηθεί μια αποκεντρωμένη εφαρμογή (DApp) για να επιτύχει αυτή την αλληλεπίδραση. Προς το παρόν θα γίνει χρήση των δυνατοτήτων του Remix.

Επιστρέφοντας στο Remix, πάντα στην επιλογή *Deploy&Run Transactions*, στο μενού *Deployed Contracts* θα πρέπει να υπάρχει το *ChatExample.sol*. Πατώντας στο βελάκι, για να εμφανιστούν όλες οι επιλογές, εμφανίζονται δύο έγχρωμα κουτάκια με τα ονόματα των συναρτήσεων που έχει το smart contract, όπως φαίνεται και στην **Εικόνα 6.13**. Το ένα κουτάκι αφορά τη συνάρτηση *sendMessage* και αναμένει κείμενο (*string _content*), ενώ το άλλο αφορά τη συνάρτηση *getMessages* και επιστρέφει τα μηνύματα.

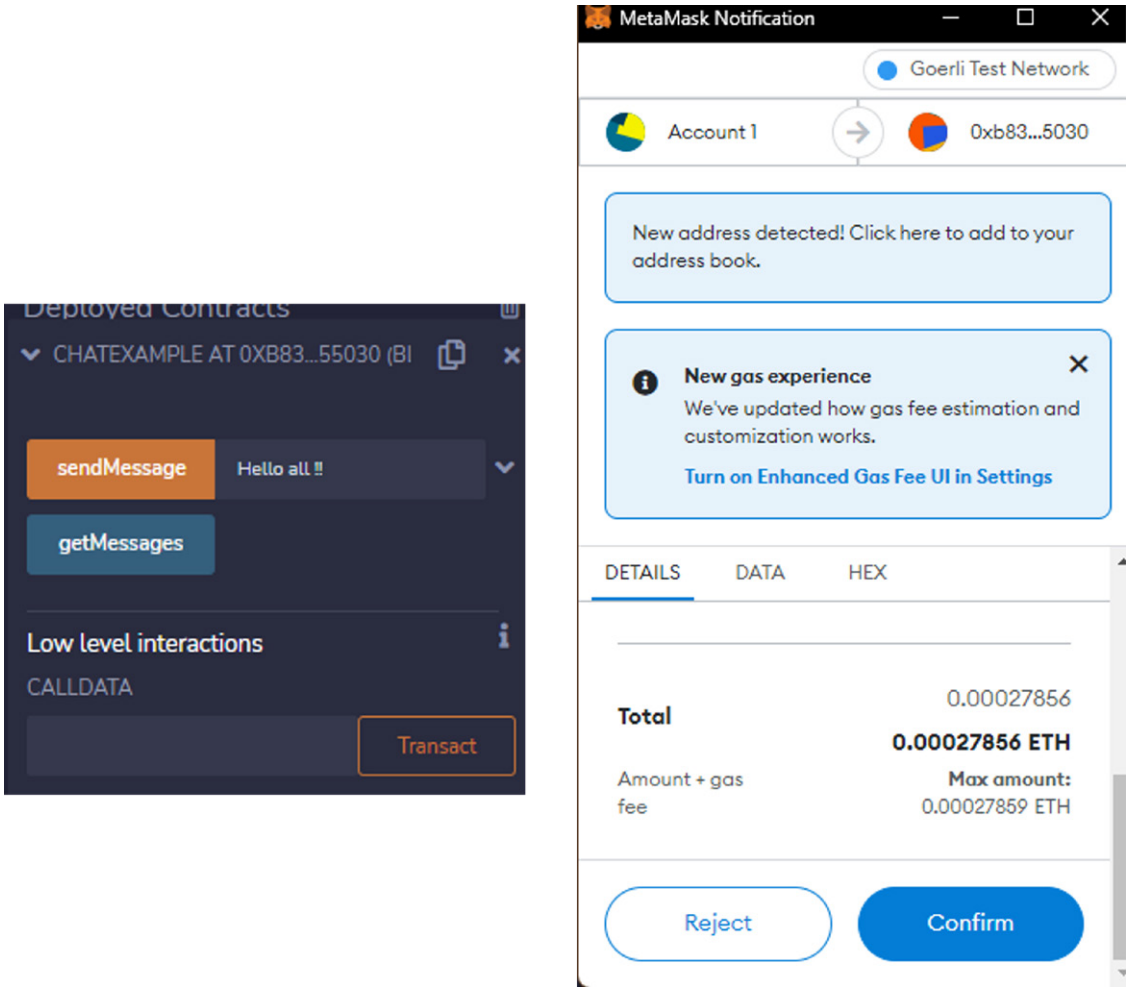


Εικόνα 6.13 *Η πρόσβαση στις συναρτήσεις του smart contract μέσω Remix.*

Αυτά τα κουτάκια θα είναι και το μέσο αλληλεπίδρασης με το smart contract. Πρώτα θα γραφτεί ένα μήνυμα στο κενό δίπλα στη συνάρτηση *sendMessage*. Στην **Εικόνα 6.14** φαίνεται ότι γράφεται η πρόταση *Hello all !!*

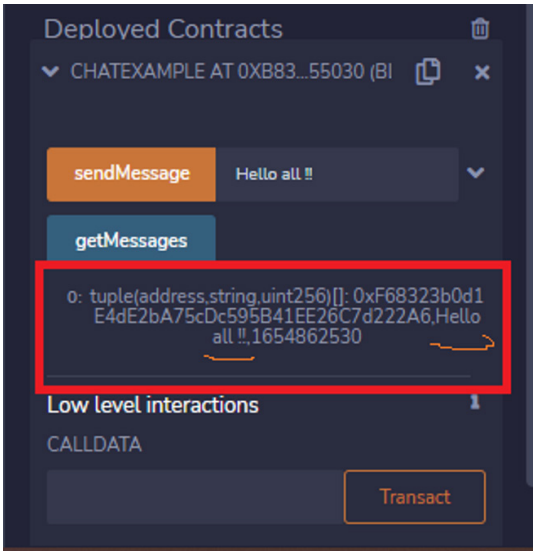
Μόλις περαστεί το περιεχόμενο του μηνύματος, τότε μπορεί να πατηθεί το πορτοκαλί κουμπί με το όνομα της συνάρτησης *sendMessage*. Αμέσως θα φορτώσει το Metamask και θα ζητηθεί να επιβεβαιωθεί η συναλλαγή που ξεκινά λόγω της αποστολής του μηνύματος στο δίκτυο, μαζί με πληροφορίες σχετικά με το κόστος αυτής.

Η Εικόνα 6.14 απεικονίζει το μήνυμα αυτό στο Metamask για τη συναλλαγή που παρουσιάστηκε (το μήνυμα που γράφηκε παραπάνω, δηλαδή).



Εικόνα 6.14 *Η αποστολή μηνύματος μέσω της συνάρτησης sendMessage και η αποδοχή της συναλλαγής στο Metamask.*

Τελευταίο βήμα είναι η χρήση της συνάρτησης *getMessages* για να διαβαστούν όλα τα μηνύματα. Στην περίπτωσή μας το μήνυμα: *Hello all !!* Για αυτό μπορεί κάποιος να πατήσει στο Remix το μπλε κουμπί με το όνομα της συνάρτησης. Τότε θα παρατηρηθεί μια διαφορά σε σχέση με πριν. Θα δει τα αποτελέσματα να εμφανίζονται κάτω από τη συνάρτηση, όπως φαίνεται και στην **Εικόνα 6.15**.



Εικόνα 6.15 *Η ανάγνωση των μηνυμάτων μετά την εκτέλεση της getMessages.*

Ταυτόχρονα, όμως, μπορεί να δει και τα αποτελέσματα στο παράθυρο του command line στο Remix. Η πληροφορία εκεί είναι πιο αναλυτική και βοηθά να γίνει η σύνδεση με τον κώδικα της συνάρτησης που μελετήθηκε νωρίτερα.

Η **Εικόνα 6.16** απεικονίζει την πληροφορία αυτή.

```
"0":"tuple(address,string,uint256)[]:
0xF68323b0d1E4dE2bA75cDc595B41EE26C7d222A6,Hello all !!,1654862530"
```

Εικόνα 6.16 *Η έξοδος της συνάρτησης getMessages στο command line του Remix.*

Μία πολύ σημαντική διαφορά στην εκτέλεση των δύο συναρτήσεων είναι ότι μετά την εκτέλεση της πρώτης (*sendMessage*) δημιουργείται μια συναλλαγή, για αυτό και γίνεται σύνδεση με το Metamask για έγκριση (εφόσον υπάρχουν αρκετά χρήματα). Για την ολοκλήρωση των ενεργειών της δεύτερης συνάρτησης δεν έγινε κάτι παρόμοιο. Ο λόγος είναι, όπως είχε εξηγηθεί και κατά την ανάλυση του κώδικα του smart contract, ότι δεν προκύπτει κάποια συναλλαγή με την κλήση αυτής. Η αναζήτηση γίνεται χωρίς να αλλάξει κάτι στο ledger, επομένως δεν απαιτείται σύνδεση με το Metamask για την επιβεβαίωση αυτής.

Πρόταση για τη συνέχεια: Ένας τρόπος να συνεχίσει κάποιος με τη μελέτη του παραδείγματος είναι να περάσει και άλλα μηνύματα μέσα στο blockchain μέσω της συνάρτησης *sendMessage*. Κατόπιν μπορεί να καλέσει την *getMessages* και να επιβεβαιώσει ότι βλέπει τα μηνύματα. Αν και το δεύτερο μέρος μπορεί να εκτελεστεί άφοβα, για την εισαγωγή των μηνυμάτων θα πρέπει όποιος ασχοληθεί να έχει αρκετά Goerli ETH για να μπορέσει να τα περάσει στο δίκτυο με τις απαραίτητες συναλλαγές.

Τέλος, στην περίπτωση που τα ETH που έχει πάρει κάποιος από το faucet της επιλογής του δεν φθάνουν, τότε μπορεί να περιμένει 24 ώρες για να ζητήσει νέα, και έτσι να συνεχίσει με την εκτέλεση της επέκτασης του παραδείγματος.

Βιβλιογραφία

Guibert, T. (2022). *Create Your First Ethereum Smart Contract with Remix IDE*. Online πηγή: https://betterprogramming.pub/create-your-first-ethereum-smart-contract-with-remix-ide-667e46e81901 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Sayeed, S., Marco-Gisbert, H., & Caira, T. (2020). Smart Contract: Attacks and Protections. *IEEE Access, 8*, pp. 24416-24427. Online πηγή: https://ieeexplore.ieee.org/document/8976179

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday, 2*(9), Online πηγή: https://journals.uic.edu/ojs/index.php/fm/article/view/548

ΚΕΦΑΛΑΙΟ 7
Χρήση και Δημιουργία Tokens

Σύνοψη

Στο Κεφάλαιο αυτό παρουσιάζονται λεπτομέρειες που αφορούν το πώς χρησιμοποιούνται τα smart contracts για τη δημιουργία tokens σε ένα δίκτυο blockchain. Ιδιαίτερη έμφαση δίνεται στον ρόλο που έπαιξε το Ethereum στην ανάπτυξη των tokens με την εισαγωγή των αντίστοιχων προτύπων για αυτά. Ακολούθως γίνεται μια παρουσίαση των πιο δημοφιλών προτύπων ERC Token στο δίκτυο του Ethereum.

Τέλος, παρουσιάζονται και άλλα παραδείγματα προτύπων για tokens στο Ethereum, αλλά και σε άλλα δίκτυα (π.χ. EOS, Tezos).

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2 και 6.

7.1 Τι είναι ένα token

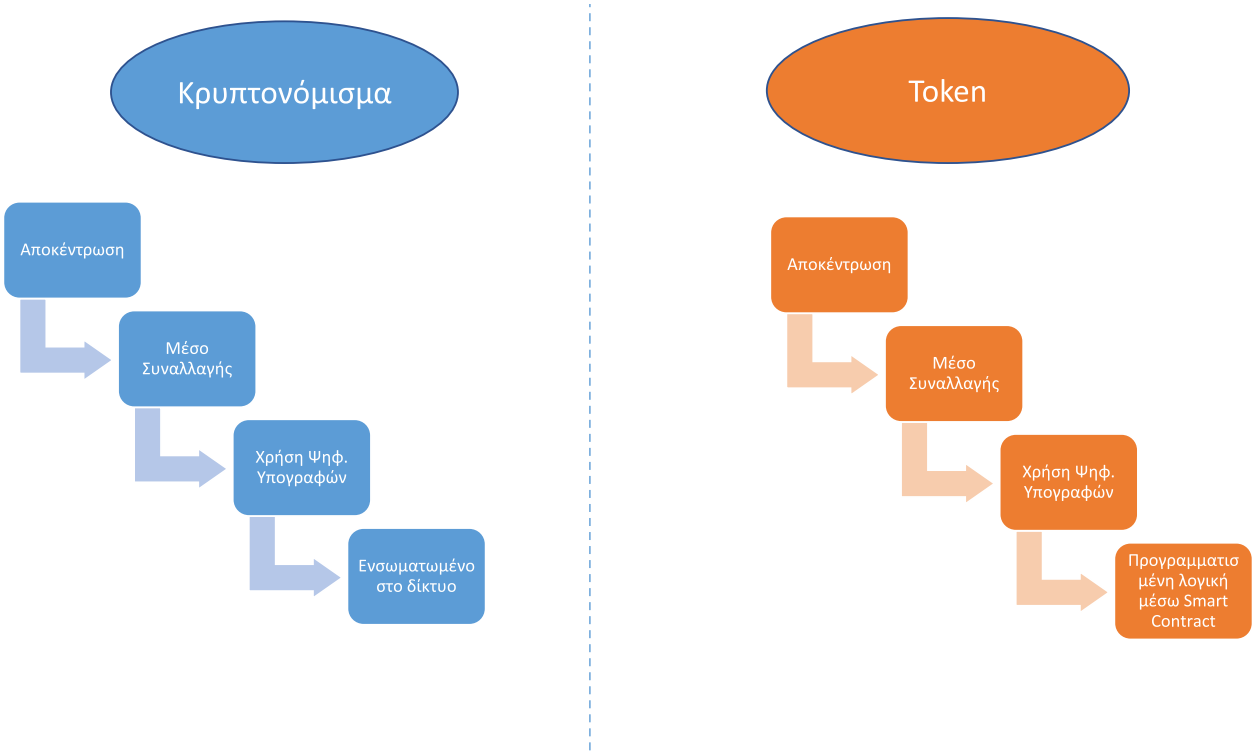
Τα tokens αποτελούν πολύ χρήσιμα εργαλεία στον κόσμο του blockchain, τα οποία όμως συχνά έχουν παρεξηγηθεί. Ιδιαίτερα στην τεχνολογία του blockchain με τη λέξη token εννοείται η οποιαδήποτε αναπαράσταση ενός αντικειμένου η οποία και ενδείκνυται να ακολουθεί συγκεκριμένες οδηγίες, που ονομάζονται *πρότυπα* (*standards*).

Στο Κεφάλαιο 2 αναπτύχθηκαν παραδείγματα στα οποία οι μάρκες/διακριτικά (ή απλώς tokens) έχουν αποδειχθεί πολύ χρήσιμες σε συγκεκριμένες εφαρμογές του φυσικού κόσμου, όπως είναι, για παράδειγμα, οι οικονομικές συναλλαγές ή η απόδειξη δικαιώματος πρόσβασης σε έναν χώρο. Ταυτόχρονα, όμως, με τη βοήθεια των χαρακτηριστικών της τεχνολογίας του blockchain (π.χ. χρήση κρυπτογραφίας για παρακολούθηση ιστορικού συναλλαγών) και ιδιαίτερα με τη χρήση smart contracts, τα tokens έχουν βρει νέους, αλλά και ανανεωμένους, ρόλους χρήσης και εφαρμογές στον ψηφιακό κόσμο.

Αυτό συμβαίνει επειδή στον ψηφιακό κόσμο επιτυγχάνεται να συνδυαστεί με τη μορφή ενός token τόσο η απεικόνιση αμιγώς ψηφιακών αντικειμένων (γνωστά και ως *digital assets*), όπως είναι, για παράδειγμα, τα CryptoKitties, όσο και αντικειμένων του φυσικού κόσμου, όπως είναι, για παράδειγμα, ο τίτλος ιδιοκτησίας ενός ακινήτου ή ενός έργου τέχνης. Μάλιστα, με τη χρήση των smart contracts για την υλοποίηση των tokens παρέχεται η δυνατότητα να αποκτήσει το καθένα από αυτά τα δικά του μοναδικά χαρακτηριστικά και ακόμα, ανάλογα με την υλοποίηση, τη δική του μοναδική αξία (βλ. Κεφάλαιο 2). Επίσης, δίνεται η δυνατότητα στα tokens να συναλλάσσονται επιτυχώς με ασφάλεια και διαφάνεια, με τη βοήθεια της κρυπτογραφίας και των ψηφιακών υπογραφών που βρίσκονται στην καρδιά μιας λύσης βασισμένης στην τεχνολογία του blockchain.

Για παράδειγμα, τα tokens αποτέλεσαν και μια λύση που ευδοκίμησε επιτυχώς σε έναν από τους πιο δημοφιλείς τομείς εφαρμογής στον κόσμο του blockchain, στον τομέα της *αποκεντρωμένης οικονομίας* (*Decentralized Finance, DeFi*). Έτσι, για αρκετό καιρό και για πολύ κόσμο η σημασία της λέξης token ήταν συνυφασμένη με αυτήν ενός κρυπτονομίσματος. Την αίσθηση αυτή ενίσχυε και το γεγονός ότι τα πιο γνωστά κρυπτονομίσματα (το bitcoin και το ether) είναι και αυτά ένα είδος token.

Επομένως, οι δύο αυτές έννοιες έχουν πολλά κοινά χαρακτηριστικά, αλλά και σημαντικές διαφορές, όπως φαίνεται και στην **Εικόνα 7.1**.



Εικόνα 7.1 *Οι διαφορές μεταξύ ενός token και ενός κρυπτονομίσματος.*

Σύμφωνα με την Εικόνα 7.1, ένα token αποτελεί ένα μέσο μεταφοράς αξιών σε ένα αποκεντρωμένο σύστημα, παρομοίως με ένα κρυπτονόμισμα. Επιπλέον, στις συναλλαγές που συμμετέχουν tokens ή κρυπτονομίσματα χρησιμοποιούνται ψηφιακές υπογραφές για την απόδειξη της ιδιοκτησίας της μεταφερόμενης αξίας. Η βασική διαφορά τους όμως έγκειται στο γεγονός ότι το κρυπτονόμισμα δημιουργείται από το πρωτόκολλο blockchain του δικτύου και είναι υπεύθυνο αυτό για τη διαχείρισή του. Σε αντίθεση με το οποιοδήποτε token, το οποίο δημιουργείται σε επίπεδο εφαρμογής με την υλοποίηση ενός smart contract (Κεφάλαιο 2).

Αυτό έχει ως αποτέλεσμα:

Α) Τα έξοδα που πληρώνει κάποιος για τη δημιουργία συναλλαγής στην οποία θα μεταφέρεται ένα token να είναι μεγαλύτερα από τα αντίστοιχα έξοδα μεταφοράς του κρυπτονομίσματος.

Β) Τη χρήση του κρυπτονομίσματος του δικτύου για την ολοκλήρωση της μεταφοράς του token.

Γ) Να μη χρειάζεται να δημιουργείται ξεχωριστό δίκτυο blockchain για να ενσωματώσει ένα token, αλλά αυτό μπορεί να συνυπάρξει, όπως περιγράφηκε, με το κρυπτονόμισμα του δικτύου στο οποίο θα αναπτυχθεί το smart contract.

Για παράδειγμα, στο δίκτυο του Ethereum έχουν δημιουργηθεί τουλάχιστον 350.000 smart contracts (Tang, 2021) που περιέχουν την υλοποίηση ενός συγκεκριμένου προτύπου token.[81] Οι συναλλαγές των tokens αυτών όμως χρειάζονται και πάλι την αγορά gas για την ολοκλήρωσή τους στο δίκτυο του Ethereum. Το gas όμως αγοράζεται μόνο με ETH, επομένως θα πρέπει να υπάρχει και αυτό στην κατοχή του αποστολέα των tokens.

Από την άλλη πλευρά, είναι σημαντικό να τονιστεί ότι έγινε δυνατή η ανάπτυξη ενός τόσο σημαντικού αριθμού από tokens μέσα σε ένα δίκτυο blockchain χωρίς να παρουσιαστεί η ανάγκη για δημιουργία ξεχωριστών δικτύων για την κυκλοφορία αυτών. Η διαδικασία αυτή ενισχύθηκε από τη δημιουργία των κατάλληλων προτύπων στο Ethereum, επιτρέποντας την αλληλεπίδραση μεταξύ των διάφορων tokens που δημιουργήθηκαν σε αυτό, όπως εξηγείται λεπτομερώς στη συνέχεια.

7.2 Είδη tokens

Η χρήση ενός token καλύπτει ένα ευρύ φάσμα περιπτώσεων, που περιέχουν ενδεικτικά:

- την απόδοση πόντων σε ένα σύστημα συλλογής τέτοιων,

[81] Η αναφορά σχετίζεται το πρότυπο ERC-20, που παρουσιάζεται αργότερα στο κεφάλαιο.

- τη χρήση τους ως αντικειμένων σε ένα ψηφιακό παιχνίδι,
- τη χρήση τους ως εισιτηρίων κάποιας λοταρίας,
- την εκδήλωση δικαιώματος ψήφου,
- την προσφορά κάποιας οικονομικής αξίας,
- την ιδιοκτησία ενός αντικειμένου.

Σε καθεμία από τις περιπτώσεις αυτές τα tokens παίζουν έναν ιδιαίτερο ρόλο, ο οποίος μπορεί να διαφέρει ανάμεσά τους. Έτσι, για παράδειγμα, τα tokens που χρησιμοποιούνται για να αποδώσουν το δικαίωμα σε μια ψηφοφορία έχουν την ίδια αξία μεταξύ τους, τουλάχιστον ανάμεσα σε όσους έχουν το δικαίωμα αυτό. Αυτό σημαίνει ότι το token μπορεί να ανταλλαχθεί μεταξύ δύο κατόχων χωρίς να υπάρχει κάποιο πρόβλημα στο σενάριο χρήσης. Κάτι αντίστοιχο συμβαίνει και με τη χρήση τους ως μέσου μεταφοράς χρηματικής αξίας, όπως είναι, για παράδειγμα, το ETH. Η ανταλλαγή ενός 1 ETH με ένα άλλο ETH δεν επηρεάζει την αξία που έχει κάποιος στην κατοχή του (βλ. και Κεφάλαιο 2). Τα tokens, λοιπόν, τα οποία είναι ισοδύναμα και τα οποία στην ανταλλαγή τους έχουν *ίση αξία* για τα δύο μέρη (κατά τη μεταφορά ίσης ποσότητάς τους) λέγονται *Fungible Tokens.*

Στην αντίθετη περίπτωση, η ιδιοκτησία ενός αμαξιού δεν μπορεί να είναι ισοδύναμη με την ιδιοκτησία μιας οικίας, οπότε τα tokens που δημιουργούνται στο σενάριο αυτό δεν μπορούν να ανταλλαχθούν με τον ίδιο τρόπο με τον οποίο γίνονται οι αλλαγές στην προηγούμενη περίπτωση. Αυτό σημαίνει ότι τα tokens αυτά έχουν διαφορετικά χαρακτηριστικά και ότι είναι μοναδικά. Αλλά δεν σημαίνει ότι *δεν μπορούν να ανταλλαχθούν.* Η ανταλλαγή, ως ενέργεια (ή συναλλαγή), επιτρέπεται να συμβεί, αλλά η αξία που λαμβάνει το κάθε μέρος είναι διαφορετική και όχι ίδια, όπως στην περίπτωση των Fungible Tokens. Έτσι, τα tokens τα οποία λόγω αξίας διαφέρουν, είναι δηλαδή κατά μια έννοια μοναδικά, ονομάζονται *Non-Fungible Tokens* ή, σε συντομογραφία, NFT, όπως έχουν γίνει πλέον πιο γνωστά.

7.3 Προτάσεις και πρότυπα για tokens στο Ethereum

Με την ενσωμάτωση των smart contracts και τη δυνατότητα που αυτά προσδίδουν για τη δημιουργία ψηφιακών αντικειμένων έγινε πολύ εύκολο στους προγραμματιστές να δημιουργήσουν τέτοια αντικείμενα, να τα ενσωματώσουν και να τα διαχειριστούν στο δίκτυο του Ethereum. Σύντομα, όμως, παρουσιάστηκε η ανάγκη για αλληλεπίδραση μέσω συναλλαγών μεταξύ των διαφορετικών υλοποιήσεων/αντικειμένων. Για να ικανοποιηθεί η ανάγκη αυτή λοιπόν, προτάθηκε η εφαρμογή κοινών προτύπων που χρησιμοποιούνται ως αναφορά για τη δημιουργία, για παράδειγμα, υλοποιήσεων σε επίπεδο εφαρμογής, όπως είναι τα tokens.

Η ύπαρξη αυτών των κοινών προτύπων (standards) επιτρέπει να αποκτήσουν όλοι οι συμμετέχοντες στο σύστημα έναν κοινό τρόπο να διαχειρίζονται τις συναλλαγές που δημιουργούνται από υλοποιήσεις που ακολουθούν τα πρότυπα αυτά. Για παράδειγμα, με τη βοήθεια των προτύπων που αφορούν τη δημιουργία και τις ενέργειες ενός token (π.χ. ERC-20, ERC-777) θα είναι δυνατό για ένα πορτοφόλι ή ανταλλακτήριο (ή γενικότερα για την κάθε υλοποίηση που αναλαμβάνει να διαχειριστεί tokens) το οποίο δραστηριοποιείται στο δίκτυο του Ethereum να γνωρίζει τον τρόπο με τον οποίο θα ολοκληρώσει μια συναλλαγή όπου συμμετέχουν τα tokens αυτά. Επιπλέον, τα πρότυπα αυτά θα έχουν διευκρινίσεις και για τον τρόπο με τον οποίο θα υπολογίζεται το υπόλοιπο μιας διεύθυνσης σε αριθμό από tokens καθώς και πώς μεταβάλλεται το υπόλοιπο αυτό ανάλογα με την πρόσθεση ή αφαίρεση tokens από αυτό.

Στο Ethereum η προσπάθεια για τη δημιουργία standards υλοποιήθηκε, αρχικά, με τρόπο αντίστοιχο με το Bitcoin. Δηλαδή με τη δυνατότητα υποβολής προτάσεων προς μελέτη και έγκριση από ολόκληρη την κοινότητα του Ethereum. Τέτοιες προτάσεις, οι οποίες πήραν το όνομα *Προτάσεις Βελτίωσης του Ethereum* (*Ethereum Improvement Proposals, EIPs*), μπορούσε να υποβάλει ο κάθε χρήστης του δικτύου, ο οποίος και είναι υπεύθυνος για την υποστήριξή τους παρέχοντας την αναγκαία επιχειρηματολογία επί αυτών. Μάλιστα στην EIP-1[82] υπάρχει αναλυτική παρουσίαση του τρόπου σύνταξης μιας EIP, καθώς και αναλυτικές πληροφορίες για το τι είναι και ποιοι τύποι EIP μπορούν να υπάρχουν.

Επιπλέον, ο χρήστης αναλαμβάνει και τον συντονισμό μεταξύ των συμμετεχόντων στο δίκτυο, με σκοπό την επίτευξη συναίνεσης ως προς το περιεχόμενο της πρότασής τους. Οι EIPs δεν ήταν αναγκαστικά μόνο σχετικές

[82] Μπορείτε να βρείτε την EIP-1 στον online σύνδεσμο εδώ: https://eips.ethereum.org/EIPS/eip-1

με πρότυπα εφαρμογών, αλλά σε αυτές συμπεριλαμβάνονται και προτάσεις βελτιώσεων της λειτουργίας του πρωτοκόλλου και του δικτύου.

Στην περίπτωση των tokens, οι προτάσεις αυτές αφορούσαν το επίπεδο εφαρμογής στο δίκτυο του blockchain και περιέχουν συμβουλευτικές οδηγίες για τον τρόπο με τον οποίο θα πρέπει να γράφονται τα smart contracts που δημιουργούν tokens. Όπως αναλύεται και στη συνέχεια, περιέχουν μια αναφορά στις συναρτήσεις που θα πρέπει να υλοποιούν τα smart contracts για τη δημιουργία των tokens, έτσι ώστε να υπάρχει μια ομοιογένεια στη δομή τους, που θα διευκολύνει και τη διαχείρισή τους από άλλες εφαρμογές του δικτύου. Βέβαια, αυτό δεν σημαίνει ότι τα tokens θα είναι ίδια, καθώς το περιεχόμενο των μεταβλητών που προσδίδουν τα ιδιαίτερα χαρακτηριστικά στο token θα διαφέρει. Πιο αναλυτικά αυτό φαίνεται στη συνέχεια για τα πρότυπα που θα μελετηθούν.

Τέλος, οι EIPs που λαμβάνουν την έγκριση για ενσωμάτωση στο δίκτυο στην τελική τους μορφή ονομάζονται *ERCs* (*Ethereum Requests for Comments*) και συνοδεύονται και από έναν αριθμό, ίδιο με αυτόν που πήραν κατά την αρχική υποβολή τους ως EIPs.[83]

Στη συνέχεια, το κεφάλαιο εστιάζει στα πιο δημοφιλή πρότυπα ERCs τα οποία αφορούν τη δημιουργία tokens. Ενδεικτικά αναφέρονται τα πρότυπα: ERC-20, ERC-721, ERC-777 και ERC-1155.

7.3.1 Το Πρότυπο ERC-20

Το πρότυπο ERC-20 αποτελεί το πιο γνωστό πρότυπο για τη δημιουργία tokens, με χιλιάδες smart contracts να έχουν βασιστεί σε αυτό ήδη, δημιουργώντας τις δικές τους υλοποιήσεις στο δίκτυο του Ethereum.

Ξεκίνησε ως Πρόταση για Βελτίωση στο δίκτυο του Ethereum (EIP-20) και υποβλήθηκε προς έλεγχο τον Νοέμβριο 2015 (EIP-20, 2015). Η πρόταση γράφτηκε από τους Fabian Vogelsteller και Vitalik Buterin και αφορά τη διαχείριση των Fungible Tokens, εκείνων των tokens δηλαδή που δεν έχουν διαφορά στα χαρακτηριστικά και στις ιδιότητές τους. Ο σκοπός της πρότασης, όπως περιγράφεται και στην EIP-20, είναι να δημιουργηθεί μια πρότυπη διεπαφή που θα επιτρέπει σε ένα token στο δίκτυο του Ethereum να μπορεί να επαναχρησιμοποιηθεί από άλλες εφαρμογές, όπως είναι υλοποιήσεις πορτοφολιών και ανταλλακτήρια.

Πιο αναλυτικά, στην πρόταση παρουσιάζεται η υλοποίηση μιας πρότυπης διεπαφής σε μορφή κώδικα που μπορεί να χρησιμοποιηθεί σε ένα smart contract, η οποία και περιγράφει τη βασική λειτουργικότητα ενός token. Ενδεικτικά, περιγράφεται ο τρόπος με τον οποίο μπορεί να γίνει η μεταφορά των tokens μέσα στο δίκτυο καθώς και πώς παρέχεται έγκριση για τη χρήση tokens από ένα τρίτο μέρος (ή εφαρμογή).

Για να επιτευχθεί ο σκοπός, η πρόταση αποτελείται από τη δήλωση συναρτήσεων (σε γλώσσα προγραμματισμού συγγραφής smart contract, όπως είναι η Solidity) που περιέχουν τη βασική λειτουργία ενός token στο δίκτυο του Ethereum. Επιπλέον, αναφέρονται και 2 *Events* τα οποία έχουν σκοπό να ενημερώνουν την εφαρμογή (D-app στην περίπτωση του token) ότι έχει γίνει μια αλλαγή στο blockchain και στο smart contract. Τα events αυτά επιστρέφουν δεδομένα, τα οποία και πρέπει να τα διαχειριστεί το D-app.

Προχωρώντας στην παρουσίαση των δηλώσεων των συναρτήσεων που συμπεριλαμβάνονται στην πρότυπη διεπαφή στην EIP-20, να ληφθεί υπόψη αρχικά η αναφορά στην έκδοση της γλώσσας Solidity, που πρέπει να χρησιμοποιηθεί για την ορθή σύνταξη του κώδικα για τη χρήση της διεπαφής. Κατόπιν παρουσιάζονται οι δηλώσεις των 6 προτεινόμενων *συναρτήσεων* (*functions*) που περιλαμβάνονται στο πρότυπο, μαζί με τις 3 προαιρετικές που μπορούν επίσης να χρησιμοποιηθούν για πλήρη συμμόρφωση με αυτό.

Η δήλωση των συναρτήσεων (βασικών και προαιρετικών) που ορίζονται στη διεπαφή στην EIP-20 φαίνεται στον **Πίνακα 7.1**.

[83] Ο αριθμός αυτός είναι στην πράξη ο αύξων αριθμός υποβολής πρότασης ως EIP.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event	Βασική/ Προαιρετική
function *totalSupply()* public view returns (uint256)	Επιστρέφει τον συνολικό αριθμό από tokens που έχουν δημιουργηθεί. Ο αριθμός αυτός επιστρέφεται ως ένας ακέραιος των 256 bits.	–	Βασική
function *balanceOf*(address owner) public view returns (uint256 balance)	Επιστρέφει τον αριθμό των tokens που έχει στην κατοχή της μια συγκεκριμένη διεύθυνση (owner). Ο αριθμός αυτός επιστρέφεται ως ένας ακέραιος των 256 bits.	–	Βασική
function *transfer*(address to, uint256 value) public returns (bool success)	Στέλνει *value* το πλήθος tokens από τον λογαριασμό του χρήστη στη διεύθυνση *to*. Επιστρέφει το αποτέλεσμα (Ναι ή Όχι) και ενεργοποιεί event. ΠΡΕΠΕΙ να αναφέρει αν δεν υπάρχουν αρκετά tokens στον λογαριασμό του χρήστη. Η μεταφορά 0 tokens πρέπει να εξυπηρετείται κανονικά.	ΝΑΙ (Transfer event)	Βασική
function *transferFrom*(address _from, address _to, uint256 _value) public returns (bool success)	Μεταφέρει value το πλήθος tokens από τη διεύθυνση *from* στη διεύθυνση *to*. Επιστρέφει το αποτέλεσμα (Ναι ή Όχι) και ενεργοποιεί Event. Για να γίνει η μεταφορά, θα πρέπει να έχει δοθεί έγκριση για το ποσό στη διεύθυνση *from* μέσω της *allowance*, οπότε και ανανεώνεται το ποσό αυτής. Με τη συνάρτηση αυτή επιτρέπεται στο contract να μεταφέρει tokens για εμάς, μετά την έγκρισή μας. Η μεταφορά 0 tokens πρέπει να εξυπηρετείται κανονικά.	ΝΑΙ (Transfer event)	Βασική
function *allowance*(address _owner, address _spender) public view returns (uint256 remaining)	Επιστρέφει το πλήθος από tokens του *owner* που έχει στη διάθεσή της η διεύθυνση του *spender* για να ξοδέψει ακόμα. Αρχικά, η τιμή αυτή είναι ίση με 0. Αλλάζει με χρήση των συναρτήσεων *transferFrom* και *approve*.	–	Βασική
function *approve*(address _spender, uint256 _value) public returns (bool success)	Επιστρέφει αν είναι επιτυχής η ανάθεση (Ναι ή Όχι) *value* tokens προς χρήση από τη διεύθυνση του *spender*. *Προσοχή*: Προτού ενημερωθεί το ποσό αυτό σε νέα τιμή με νέα κλήση της συνάρτησης, προτείνεται (για αποφυγή επιθέσεων) να πηγαίνει πρώτα στην τιμή 0 και μετά, με νέα κλήση, στη νέα τιμή που θέλει ο χρήστης. Με αυτόν τον τρόπο αποφεύγεται η δυνατότητα χρήσης και των δύο τιμών ταυτόχρονα από κακόβουλους χρήστες.	ΝΑΙ (Approval event)	Βασική
function *name()* public view returns (string)	Επιστρέφει το όνομα του token. Η συνάρτηση αυτή χρησιμοποιείται για διευκόλυνση στη χρήση, αλλά τα smart contracts και οι διεπαφές θα πρέπει να μπορούν να λειτουργούν και χωρίς να υπάρχουν δεδομένα για τη συνάρτηση αυτή.	–	Προαιρετική
function *symbol()* public view returns (string)	Επιστρέφει το σύμβολο του token.	–	Προαιρετική
function *decimals()* public view returns (uint8)	Επιστρέφει τον αριθμό των δεκαδικών ψηφίων που χρησιμοποιούνται από το token για τη λειτουργία του. Π.χ. μια τιμή ίση με 5 σημαίνει ότι το ποσό των tokens πρέπει να διαιρεθεί με το 10^5 (100.000) για να αποδοθεί σωστά.	–	Προαιρετική

Πίνακας 7.1 *Δήλωση συναρτήσεων στην EIP-20 (βασικές + προαιρετικές).*

Πέρα από τις συναρτήσεις αυτές, στο πρότυπο περιλαμβάνονται και δύο Events (Approval και Transfer), που περιγράφονται στον **Πίνακα 7.2**. Αυτά τα events ενημερώνουν την εφαρμογή του χρήστη (front-end) για τις αλλαγές στο blockchain που λαμβάνουν χώρα με την εκτέλεση συγκεκριμένων συναρτήσεων του προτύπου.

Η σύνδεση των events αυτών με τις συναρτήσεις φαίνεται επίσης και στον Πίνακα 7.1.

Δήλωση Event	Περιγραφή
event *Transfer*(address indexed _from, address indexed _to, uint256 _value)	Ενεργοποιείται κατά τη μεταφορά tokens (ακόμα και μηδενική τιμή το ενεργοποιεί). Ένα νέο contract που δημιουργεί νέα tokens θα πρέπει να έχει ως διεύθυνση from την τιμή 0x0.
event *Approval*(address indexed _owner, address indexed _spender, uint256 _value)	Ενεργοποιείται με την ανανέωση της έγκρισης του πλήθους tokens με χρήση της συνάρτησης approve().

Πίνακας 7.2 *Δήλωση Events στην EIP-20.*

Είναι σημαντικό να τονιστεί ότι το πρότυπο αφήνει ανοικτό το θέμα της ποσότητας των tokens που θα δημιουργηθούν. Δηλαδή αφήνει στον εκάστοτε δημιουργό ενός token να προχωρήσει στη δική του εφαρμογή σχετικά με τη διαθέσιμη ποσότητα των tokens του. Η συνάρτηση totalSupply() δεν δηλώνει τον αριθμό αυτό, αλλά επιστρέφει το αποτέλεσμα της απόφασης για το στάδιο αυτό.

Ο πιο απλός τρόπος για τη δημιουργία των tokens είναι να δημιουργηθεί εξαρχής η συνολική ποσότητά τους και να αποθηκευτεί αυτή στη διεύθυνση του contract. Υπάρχει, επίσης, η δυνατότητα να δημιουργούνται tokens στην πορεία, συνήθως ως επιβράβευση στους miners ή στους χρήστες. Όπως αναφέρθηκε και πριν, είναι στη διακριτική ευχέρεια του δημιουργού να αποφασίσει για τον τρόπο δημιουργίας των tokens στο δίκτυο, καθώς και να υλοποιήσει τον τρόπο επιλογής.

Από πλευράς υλοποίησης του προτύπου, υπάρχουν αρκετές υλοποιήσεις, με πιο δημοφιλείς τις εξής δύο:

- υλοποίηση από το OpenZeppelin,[84]
- υλοποίηση από ConsenSys.[85]

Τις υλοποιήσεις αυτές μπορεί να χρησιμοποιήσει ο καθένας για τη δημιουργία του δικού του token, εισάγοντας τις κλήσεις τους στην αρχή του κώδικά του.

Αυτό θα φανεί και στο παράδειγμα που ακολουθεί για διευκρίνιση και αναφορά.

7.3.1.1 Παράδειγμα δημιουργίας token τύπου ERC-20

Ένα παράδειγμα δημιουργίας ενός ERC-20 token με τη χρήση της υλοποίησης του προτύπου ERC-20 από την OpenZeppelin παρουσιάζεται στη συνέχεια. Για την ανάπτυξη του κώδικα για το token χρησιμοποιείται το εργαλείο Remix, το οποίο έχει χρησιμοποιηθεί και προηγουμένως στο βιβλίο (βλ. Κεφάλαιο 6).

Το πρότυπο της OpenZeppelin, στην έκδοση 4 που βρίσκεται, έχει ενσωματώσει, πέρα από τις συναρτήσεις (βασικές και προαιρετικές) που εμφανίζονται στην EIP-20 και ορισμένες νέες συναρτήσεις οι οποίες διευκολύνουν τους δημιουργούς, καθώς προσφέρουν επιπρόσθετες λειτουργίες που έχουν αποδειχθεί ότι χρησιμοποιούνται συχνά στις σύγχρονες εφαρμογές (D-apps).

Για τον λόγο αυτό στον **Πίνακα 7.3** παρουσιάζονται για αναφορά και οι δηλώσεις των *επιπρόσθετων συναρτήσεων* που έχουν ενσωματωθεί στην υλοποίηση της OpenZeppelin για το πρότυπο ERC-20.

[84] Online Σύνδεσμος στο github με τον κώδικα εδώ.
[85] Online Σύνδεσμος στο github με τον κώδικα εδώ.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
constructor(string memory name_, string memory symbol_)	Δίνει αρχικές τιμές στα name_ και *symbol_* (προαιρετικές συναρτήσεις στον Πίνακα 7.1). Οι τιμές που δίνονται είναι σταθερές και δεν μπορούν να αλλάξουν. Για το decimals η καθορισμένη αρχική τιμή είναι 18.	–
function *increaseAllowance*(address spender, uint256 addedValue) public virtual returns (bool)	Αυξάνει το ποσό που δόθηκε ως *allowance* από τη διεύθυνση που καλεί τη συνάρτηση προς τον spender. Εναλλακτική της *approve*() για την αποφυγή γνωστών προβλημάτων (βλ. Πίνακα 7.1). *Απαίτηση*: • Ο *spender* δεν μπορεί να είναι η διεύθυνση 0x0.	Ναι (Approval event)
function *decreaseAllowance*(address spender, uint256 subtractedValue) public virtual returns (bool)	Μειώνει το ποσό που δόθηκε ως *allowance* από τη διεύθυνση που καλεί τη συνάρτηση προς τον *spender*. Εναλλακτική της *approve()* για την αποφυγή γνωστών προβλημάτων (βλ. Πίνακα 7.1). *Απαιτήσεις*: • Ο *spender* δεν μπορεί να είναι η δ/νση 0x0. • Ο *spender* πρέπει να έχει *allowance* από αυτόν που καλεί τη συνάρτηση, με ποσό ίσο με το *subtractedValue*.	Ναι (Approval event)
function *_transfer* (address from, address to, uint256 amount) internal	Μεταφέρει *amount* από *tokens* από τη διεύθυνση του *from* σε αυτήν του *to*. Αποτελεί εναλλακτική της *transfer* και μπορεί να χρησιμοποιηθεί για να πραγματοποιήσει αυτόματα κόστη συναλλαγών. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Οι δ/νσεις *from* και *to* δεν πρέπει να είναι ίσες με 0x0. • Η δ/νση from πρέπει να έχει τουλάχιστον *amount* από *tokens*.	Ναι (Transfer event)
function *_mint*(address account, uint256 amount) internal	Δημιουργεί μια ποσότητα ίση με τη μεταβλητή *amount* από tokens. Αυξάνει τη συνολική παραγωγή των διαθέσιμων tokens. Ενεργοποιεί το *Transfer* event. *Απαίτηση*: • Ο *account* δεν μπορεί να είναι η δ/νση 0x0.	Ναι (Transfer event)
function *_burn*(address account, uint256 amount) internal	Καταστρέφει amount πλήθος από tokens, μειώνοντας έτσι το συνολικά διαθέσιμο ποσό (*totalSupply*). Ενεργοποιεί το *Transfer* event με τη διεύθυνση *to* ίση με 0x0. *Απαίτηση*: • Ο *account* δεν μπορεί να είναι η δ/νση 0x0 και πρέπει να έχει διαθέσιμο ποσό από tokens ίσα με *amount*.	Ναι (Transfer event)
function *_approve*(address owner, address spender, uint256 amount) internal	Θέτει το ποσό των tokens του *owner* που μπορεί να διαχειριστεί ο spender ίσο με *amount* (σαν *allowance*). Αποτελεί εναλλακτική της συνάρτησης approve(). Ενεργοποιεί το *Approval* event. *Απαίτηση*: • Οι δ/νσεις *owner* και *spender* δεν πρέπει να είναι οι 0x0.	Ναι (Approval event)
function *_spendAllowance*(address owner, address spender, uint256 amount) internal	Ανανεώνει το ποσό που επιτρέπει ο *owner* στον *spender* βάσει του ποσού (*amount*) που έχει ξοδευτεί. Δεν ανανεώνει το ποσό στην περίπτωση που έχει δοθεί πρόσβαση σε άπειρο ποσό. Επιστρέφει στην πρότερη κατάσταση αν δεν είναι αρκετό το ποσό της *allowance*. Μπορεί να ενεργοποιήσει το *Approval* event.	Ναι (Approval event)
function *_beforeTokenTransfer* (address from, address to, uint256 amount) internal	Συνάρτηση hook[86] η οποία καλείται πριν από την όποια μεταφορά tokens (συμπεριλαμβανομένων και των *mint* και *burn* ενεργειών). Απαιτήσεις: • Όταν καμία από τις δ/νσεις *from* και *to* δεν είναι η 0x0, τότε *amount* από *tokens* θα μεταφερθούν από την from στην *to*. • Αν η δ/νση from είναι η 0x0, τότε amount από tokens θα δημιουργηθούν (mint) για τον to. • Αν η δ/νση to είναι η 0x0, τότε *amount* από tokens ιδιοκτησίας του from θα καταστραφούν (*burn*). • Οι δ/νσεις *from* και *to* δεν πρέπει να είναι μαζί ίσες με 0x0.	–
function *_afterTokenTransfer* (address from, address to, uint256 amount) internal	Συνάρτηση hook η οποία καλείται πριν από την όποια μεταφορά tokens (συμπεριλαμβανομένων και των *mint* και *burn* ενεργειών). Απαιτήσεις: • Όταν καμία από τις δ/νσεις *from* και *to* δεν είναι η 0x0, τότε *amount* από tokens έχουν μεταφερθεί από την *from* στην *to*. • Αν η δ/νση *from* είναι η 0x0, τότε *amount* από tokens δημιουργήθηκαν (*mint*) για την *to*. • Αν η δ/νση *to* είναι η 0x0, τότε *amount* από tokens ιδιοκτησίας του *from* καταστράφηκαν (*burn*). • Οι δ/νσεις *from* και *to* δεν πρέπει να είναι *μαζί* ίσες με 0x0.	–

Πίνακας 7.3 *Οι επιπρόσθετες συναρτήσεις που έχουν ενσωματωθεί στην υλοποίηση του ERC-20 από την Open Zeppelin.*

[86] *Hook* είναι συναρτήσεις που δημιουργούν ένα κεντρικό σημείο αναφοράς για να κουμπώσουν (στα αγγλικά χρησιμοποιείται η λέξη hook) άλλες συναρτήσεις και να επεκτείνουν την αρχική συμπεριφορά τους. Καλούνται πριν ή μετά την πραγματοποίηση μιας ενέργειας για να δημιουργήσουν ένα σημείο αναφοράς.

Έχοντας πλέον μια πλήρη εικόνα για την υλοποίηση του προτύπου ERC-20 από την OpenZeppelin, στη συνέχεια παρουσιάζεται η δημιουργία του *4Blockchain* token (4Blck σε συντομογραφία).

```
// contracts/GLDToken.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract 4BlockchainToken is ERC20 {
    constructor(uint256 initialSupply)
ERC20("4Blockchain", "4Blck") {
        _mint(msg.sender, 1000*10**18);
    }
}
```

Εικόνα 7.2 *Ο κώδικας για τη δημιουργία ενός ERC-20 token βασισμένος στο πρότυπο από την OpenZeppelin.*

Ο παραπάνω κώδικας δείχνει πως είναι δυνατόν με τη βοήθεια του προτύπου της OpenZeppelin να δημιουργηθεί ένα ERC-20 token μέσα σε λίγες γραμμές κώδικα. Στην ουσία, το μόνο που χρειάστηκε ήταν να προσθέσουμε τιμές στον constructor σχετικά με το όνομα και τη συντομογραφία του token μας.

Επιπρόσθετα, δημιουργήθηκαν 1.000 4Blck Tokens με την κλήση στη συνάρτηση mint. Η παράμετρος 1000*10**18 ουσιαστικά αρχικοποίησε τα decimals και έδωσε τιμή στην ποσότητα του token, ικανοποιώντας το συνολικό απόθεμά του στο δίκτυο (supply). Να σημειωθεί εδώ ότι τα decimals χρησιμοποιούνται για να γίνει δυνατή η μεταφορά τμήματος ενός token από μια διεύθυνση σε μια άλλη, καθώς δεν επιτρέπεται η μεταφορά, για παράδειγμα, 1,5 4Blck από τη Solidity και την EVM. Αντίθετα, αυτό μπορεί να γίνει αν γραφτεί ως **1500000000000000000** *4Blck*, τιμή που αν διαιρεθεί με το 10^{18} (τιμή που δόθηκε στο decimal() από τον constructor) θα είναι ίση με 1,5 4Blck.

Κλείνοντας με το παράδειγμα, αυτή τη στιγμή δεν μπορεί να έχει κάποιος πρόσβαση στο token που δημιουργήθηκε, καθώς αυτό δεν έχει αναπτυχθεί σε ένα δίκτυο blockchain. Θα χρειαστεί είτε να δημιουργηθεί τοπικά ένα δίκτυο blockchain, στο οποίο και να εγκατασταθεί το token προς χρήση, είτε να ανέβει σε ένα πραγματικό/δοκιμαστικό δίκτυο. Για την περίπτωση της δημιουργίας ενός τοπικού δικτύου, περισσότερες πληροφορίες θα δείτε στο Κεφάλαιο 8.

7.3.2 Το Πρότυπο ERC-721

Το πρότυπο ERC-721 έρχεται να προχωρήσει ένα βήμα παραπέρα το πρότυπο ERC-20, που αφορά τη δημιουργία tokens. Έτσι, στην Πρόταση για Βελτίωση στο Ethereum με αριθμό 721 (EIP-721, 2018), που υποβλήθηκε τον Ιανουάριο του 2018 από τους Entriken, Shirley, Evans και Sachs, παρουσιάζεται ένα πρότυπο για τη διαχείριση, τον εντοπισμό και τη μεταφορά των Non-Fungible Tokens (NFTs). Πρόκειται δηλαδή για τα tokens εκείνα που έχουν διαφορές στην αξία, στη χρήση ή/και στα χαρακτηριστικά, και το πρότυπο αυτό χρησιμοποιείται για την παρακολούθηση της ιδιοκτησίας τους με τη βοήθεια της τεχνολογίας blockchain.

Το πρότυπο ERC-721 είναι πιο σύνθετο από το ERC-20 και αποτελείται επιπλέον από ορισμένες προαιρετικές προεκτάσεις, κάποιες από τις οποίες δηλώνονται σε ξεχωριστά smart contracts, όπως φαίνεται και στην ανάλυση που ακολουθεί.

Πιο αναλυτικά, η EIP-721 παρουσιάζει την περιγραφή μιας πρότυπης διεπαφής σε κώδικα που μπορεί να συμπεριληφθεί σε smart contracts για τη διαχείριση των NFTs. Η πρόταση, εμπνευσμένη από το ERC-20, έχει σκοπό να περιγράψει την πρότυπη διεπαφή που θα επιτρέπει σε εφαρμογές (πορτοφόλια, δημοπρασίες ή ανταλλακτήρια) να επεξεργάζονται tokens στο δίκτυο του Ethereum. Η ανάγκη για την υποβολή μιας νέας πρότασης για τη διαχείριση των tokens πηγάζει από το γεγονός ότι το (μέχρι τότε) υπάρχον πρότυπο ERC-20 δεν ήταν κατάλληλο για την παρακολούθηση ξεχωριστών (μοναδικών) tokens, παρά εξυπηρετεί τη μαζική παρακολούθηση tokens ίσης αξίας (Fungible Tokens).

Όσον αφορά την EIP-721, σε αυτή συμπεριλαμβάνονται οι δηλώσεις 9 συναρτήσεων, οι οποίες και φαίνονται στον **Πίνακα 7.4**.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
function *balanceOf*(address _owner) external view returns (uint256)	Επιστρέφει τον αριθμό από NFTs που έχει στην κατοχή της μια συγκεκριμένη διεύθυνση (*_owner*). Ο αριθμός αυτός επιστρέφεται ως ένας ακέραιος των 256 bits και η απάντηση μπορεί να είναι και το μηδέν.	–
function *ownerOf*(uint256 _tokenId) external view returns (address)	Επιστρέφει τη διεύθυνση του κατόχου ενός NFT που προσδιορίζεται από τον 256-bits αριθμό που δίνεται ως παράμετρος στη συνάρτηση. *Απαίτηση*: • Πρέπει να υπάρχει η *tokenId*.	–
function *safeTransferFrom*(address _from, address _to, uint256 _tokenId, bytes data) external payable	Μεταφέρει το *tokenId* από τη διεύθυνση *from* στη διεύθυνση *to*. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: Οι δ/σεις *from* και *to* δεν πρέπει να είναι ίσες με 0x0. • Το *tokenId* πρέπει να υπάρχει και η δ/ση *from* να είναι ο ιδιοκτήτης του. • Εάν καλείται η συνάρτηση από κάποιον που δεν έχει τη δ/ση *from*, πρέπει πρώτα να εγκριθεί η πράξη είτε από την *approve()* είτε από τη *setApprovalForAll*(). • Εάν η δ/ση το είναι δ/ση smart contract, τότε αυτό πρέπει να υλοποιεί το *ERC-165*, που καλείται για την ορθή μεταφορά.	NAI (Transfer event)
function *safeTransferFrom*(address _from, address _to, uint256 _tokenId) external payable	Παρόμοια με τη συνονόματη συνάρτηση, μεταφέρει το *tokenId* από τη διεύθυνση *from* στη διεύθυνση *to*. Απουσιάζει η μεταβλητή *data*, καθώς η τιμή της θεωρείται ίση με " " για την παρούσα συνάρτηση. Ενεργοποιεί το *Transfer* event. Ισχύουν οι ίδιες απαιτήσεις με τη συνονόματη συνάρτηση.	NAI (Transfer event)
function *transferFrom*(address _from, address _to, uint256 _tokenId) external payable;	Μεταφέρει την ιδιοκτησία ενός NFT από τη διεύθυνση *from* στην *to*. *ΠΡΟΣΟΧΗ* να ελεγχθεί πρώτα ότι η διεύθυνση το είναι ικανή να λάβει NFTs, αλλιώς θα χαθούν αυτά για πάντα. *ΠΡΟΤΕΙΝΕΤΑΙ* η χρήση της *safeTransferFrom().* Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: Οι δ/σεις *from* και το δεν πρέπει να είναι ίσες με 0x0. • Το *tokenId* πρέπει να υπάρχει και η δ/ση *from* να είναι ο ιδιοκτήτης του. • Εάν καλείται η συνάρτηση από κάποιον που δεν έχει τη δ/ση *from*, πρέπει πρώτα να εγκριθεί η πράξη είτε από την *approve()* είτε από την *setApprovalForAll().* • Εάν η δ/ση το είναι δ/ση smart contract, τότε αυτό πρέπει να υλοποιεί το *IERC721Receiver.onERC721Received*, που καλείται για την ορθή μεταφορά.	NAI (Transfer event)
function *approve*(address _approved, uint256 _tokenId) external payable;	Δίνει άδεια στη διεύθυνση *_approved* για να μπορεί να μεταφέρει το NFT. Η άδεια ολοκληρώνεται μετά τη μεταφορά. Η έγκριση της διεύθυνσης 0x0 ακυρώνει όλες τις προηγούμενες άδειες. Ενεργοποιεί το *Approval* event. *Απαιτήσεις*: • Ο καλών πρέπει να είναι ο ιδιοκτήτης του token ή να έχει έγκριση για αυτό. • Το *tokenId* πρέπει να είναι υπαρκτό.	NAI (Approval event)
function *setApprovalForAll*(address _operator, bool _approved) external;	Εγκρίνει ή ακυρώνει την έγκριση προς μια τρίτη διεύθυνση (*_operator*) για τη διαχείριση των tokens μέσω κλήσης των *transferFrom*() ή *safeTransferFrom*(). Ενεργοποιεί το *ApprovalForAll* event. *Απαίτηση*: • Δεν πρέπει να την καλέσει ο *_operator*.	NAI (ApprovalForAll event)
function *getApproved*(uint256 _tokenId) external view returns (address);	Επιστρέφει τη διεύθυνση του λογαριασμού που έχει πάρει έγκριση για την *tokenId*. Αν δεν έχει πάρει κανείς έγκριση, επιστρέφει τη διεύθυνση 0x0. *Απαίτηση*: • Πρέπει να υπάρχει η *tokenId*.	–
function *isApprovedForAll*(address _owner, address _operator) external view returns (bool);	Επιστρέφει την απάντηση (NAI/OXI) αν ο *_operator* έχει την άδεια να διαχειρίζεται τα NFTs του *_owner*.	–

Πίνακας 7.4 *Δήλωση συναρτήσεων στην EIP-721.*

Μαζί με τις συναρτήσεις, στην EIP-721 παρουσιάζονται και τα 3 events που επιστρέφονται κατά την κλήση των συναρτήσεων του Πίνακα 7.4.

Τα 3 αυτά events φαίνονται στον **Πίνακα 7.5**.

Δήλωση Event	Περιγραφή
event *Transfer*(address indexed _from, address indexed _to, uint256 _tokenId)	Ενεργοποιείται κατά τη μεταφορά tokens, δηλαδή με την αλλαγή στην ιδιοκτησία ενός token. Ενεργοποιείται και κατά τη γέννηση ή την καταστροφή των tokens.
event *Approval*(address indexed _owner, address indexed _approved, uint256 indexed _tokenId)	Ενεργοποιείται με την ανανέωση της έγκρισης του πλήθους tokens με χρήση της συνάρτησης *approve()*.
event *ApprovalForAll*(address indexed _owner, address indexed _operator, bool _approved);	Ενεργοποιείται όταν ανανεώνεται ή κόβεται η έγκριση από τον *_owner* στον *_operator* για τη διαχείριση των NFTs του πρώτου.

Πίνακας 7.5 *Δήλωση events στην EIP-721.*

Στην EIP-721, επιπλέον, δηλώνεται ότι κάθε smart contract το οποίο και θα ακολουθεί το πρότυπο αυτό θα πρέπει, πέρα από το ERC-721, να ακολουθεί και το πρότυπο ERC-165 ή αλλιώς το πρότυπο για πορτοφόλι (wallet) όπως αναφέρεται στην EIP-721.

Το πρότυπο αυτό υποστηρίζει τις ασφαλείς μεταφορές για το ERC-721 και η μοναδική του συνάρτηση φαίνεται στον **Πίνακα 7.6**.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
function *onERC721Received*(address _operator, address _from, uint256 _tokenId, bytes _data) external returns(bytes4);	Κατά τη μεταφορά ενός token (σύμφωνα με το πρότυπο ERC-721) με χρήση της *safeTransferFrom*() (όπως αυτή ορίζεται στην EIP-721) το contract τρέχει η συνάρτηση *onERC721Received()*. Η συνάρτηση αυτή μπορεί να ακυρώσει τη μεταφορά του token εάν δεν επιστρέψει την ενδεικτική τιμή στο contract.	–

Πίνακας 7.6 *Δήλωση συνάρτησης προτύπου ERC-165*

Επιπροσθέτως, η EIP-721 συμπεριλαμβάνει και δύο επεκτάσεις (*metadata* και *enumeration* extensions), οι οποίες μπορούν να υλοποιηθούν και βοηθούν στην απάντηση ερωτήσεων σχετικά με το όνομα του NFT καθώς και σχετικά με τα χαρακτηριστικά αυτού. Και οι δύο επεκτάσεις είναι προαιρετικές.

Ακολουθεί πιο αναλυτική περιγραφή τους:

- Η προαιρετική επέκταση με τίτλο *metadata* προσδίδει τη δυνατότητα απόδοσης ονόματος και συντομογραφίας αυτού (ως σύμβολο), μαζί με του tokenId.
- Η προαιρετική επέκταση με τίτλο *enumeration* προσδίδει τη δυνατότητα καταμέτρησης των tokens στην αλυσίδα. Συνήθως δεν συμπεριλαμβάνεται λόγω της σημαντικής ποσότητας από gas που πρέπει να καταναλωθεί για τη λειτουργία της.

Ο **Πίνακας 7.7** παρουσιάζει τις συναρτήσεις των δύο επεκτάσεων με την περιγραφή τους.

Δήλωση Συνάρτησης	Περιγραφή	Επέκταση που συναντάται
function *name()* external view returns (string _name);	Επιστρέφει το όνομα της συλλογής από tokens.	metadata
function *symbol()* external view returns (string _symbol);	Επιστρέφει τη συντομογραφία του ονόματος της συλλογής των tokens.	metadata
function *tokenURI*(uint256 _tokenId) external view returns (string);	Επιστρέφει ένα μοναδικό διακριτικό για ένα token (με την ονομασία Uniform Resource Identifier, URI). Τα διακριτικά αυτά μπορούν να δείχνουν σε ένα αρχείο JSON με συγκεκριμένη δομή η οποία ακολουθεί το JSON Σχήμα Μεταδεδομένων του ERC-721 (Εικόνα 7.3)	metadata
function *totalSupply()* external view returns (uint256);	Επιστρέφει τον συνολικό αριθμό από tokens που παρακολουθούνται από το contract. Το κάθε token έχει έναν διακριτό ιδιοκτήτη.	enumeration
function *tokenByIndex*(uint256 _index) external view returns (uint256);	Επιστρέφει την token ID για το token με διακριτικό αριθμό ίσο με την τιμή της μεταβλητής _index. Χρησιμοποιείται συνδυαστικά με την *totalSupply()* για τη σωστή τιμή στο _index.	enumeration
function *tokenOfOwnerByIndex*(address _owner, uint256 _index) external view returns (uint256);	Επιστρέφει την token ID για το token με συγκεκριμένο διακριτικό (*index*) ενός συγκεκριμένου ιδιοκτήτη (*_owner*). Χρησιμοποιείται συνδυαστικά με το *balanceOf()* για την απόδοση όλων των tokens ενός *_owner*.	enumeration

Πίνακας 7.7 *Συναρτήσεις των προαιρετικών επεκτάσεων του προτύπου ERC-721, όπως φαίνονται στην EIP-721.*

Στην **Εικόνα 7.3** φαίνεται η δομή του αρχείου JSON που αποτελεί αναφορά σύμφωνα με το Σχήμα Μεταδεδομένων JSON της ERC-721.

Εικόνα 7.3 *Η δομή του αρχείου JSON σύμφωνα με το Σχήμα Μεταδεδομένων JSON της EIP-721 (EIP-721, 2018).*

```
{
    "title": "Asset Metadata",
    "type": "object",
    "properties":
        { "name":
            { "type": "string",
              "description": "Identifies the asset
              to which this NFT represents" },
         "description":
            { "type": "string",
             "description": "Describes the asset to
            which this NFT represents" },
         "image":
            { "type": "string",
             "description": "A URI pointing to a
resource with mime type  image/the asset to
which this NFT represents. Consider making
any images at a width between 320 and 1080
pixels and aspect ratio between 1.91:1 and
4:5 inclusive."          }
            }
}
```

Η πρόταση κλείνει με την παράθεση ορισμένων επιφυλάξεων και προϋποθέσεων, αρκετές από τις οποίες τονίζεται ότι θα μπορούν να αντιμετωπιστούν σε μελλοντικές εκδόσεις της γλώσσας Solidity.

Στη συνέχεια ακολουθεί η παρουσίαση ενός smart contract που υλοποιεί το πρότυπο ERC-721 κάνοντας χρήση της υλοποίησης, ξανά, από την OpenZeppelin.

7.3.2.1 Παράδειγμα δημιουργίας token τύπου ERC-721

Στη συνέχεια, κάνοντας και πάλι χρήση της υλοποίησης από την OpenZeppelin της πρότυπης διεπαφής του ERC-721, θα γίνει η παρουσίαση της σύνταξης ενός smart contract για τη δημιουργία και διαχείριση ενός NFT.

Πριν γίνει η παρουσίαση του κώδικα, πρέπει να γίνει μια παρουσίαση των συναρτήσεων που περιέχονται στην υλοποίηση, οι οποίες, όπως και στην περίπτωση του ERC-20, προσδίδουν ορισμένες επιπρόσθετες λειτουργίες στη διεπαφή με σκοπό την αρτιότερη διαχείριση των NFTs.

Έτσι, λοιπόν, στον **Πίνακα 7.8** βλέπουμε τις επιπρόσθετες συναρτήσεις που περιέχονται στην υλοποίηση της OpenZeppelin, πέρα από αυτές που παρουσιάστηκαν στον Πίνακα 7.4. Να σημειωθεί ότι στην υλοποίηση αυτή έχει ενσωματωθεί η προαιρετική επέκταση metadata αλλά όχι η προαιρετική επέκταση enumeration. Το όνομα του συμβολαίου με την παρακάτω υλοποίηση είναι το *ERC721URIStorage.sol*.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
constructor(string name_, string memory symbol_)	Αρχικοποιεί το contract δίνοντας τιμές στα *name_* και *symbol_* (όπως αυτά ορίστηκαν στον Πίνακα 7.7).	–
function *supportsInterface*(bytes4 interfaceId) public view virtual override(ERC165, IERC165) returns (bool)	Η συνάρτηση επιτρέπει την ενίσχυση της υλοποίησης της διεπαφής του πορτοφολιού με την εισαγωγή ενός επιπλέον αριθμού/διακριτικού καταμέτρησης.	–
function *_baseURI*() internal view virtual returns (string memory)	Μέθοδος που χρησιμοποιείται για τον υπολογισμό του tokenURI. Αν έχει τιμή, τότε το URI για κάθε token υπολογίζεται ως η συνένωση του *tokenID* με το *baseURI*. Η αρχική τιμή είναι το κενό.	–
function *_safeTtransfer* (address from, address to, uint256 tokeId, bytes memory data) internal virtual	Μεταφέρει με ασφάλεια το token με το δεδομένο tokenId από τη διεύθυνση *from* στη διεύθυνση *to*. Πρώτα εξασφαλίζει ότι ο το γνωρίζει το πρωτόκολλο ERC-721. Η παράμετρος *data* αποτελείται από επιπλέον δεδομένα που στέλνονται προς τον *to*. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Οι δ/σεις from και to δεν πρέπει να είναι ίσες με 0x0. • Το tokenId πρέπει να υπάρχει και η δ/ση from να είναι ο ιδιοκτήτης του. • Εάν η δ/ση το είναι δ/ση smart contract, τότε αυτό πρέπει να υλοποιεί το *IERC721Receiver*. *onERC721Received*, που καλείται για την ορθή μεταφορά.	Ναι (Transfer event)
function *_exists*(uint256 tokenId) internal view virtual returns (bool)	Επιστρέφει αν υπάρχει το token ή όχι. Τα tokens τα διαχειρίζονται οι ιδιοκτήτες τους ή όσοι έχουν λάβει έγκριση μέσω των συναρτήσεων *approve()* ή *setApprovalForAll()*. Τα tokens αρχίζουν να υπάρχουν όταν γίνουν mint[87] και σταματούν να υπάρχουν όταν καούν (με χρήση της συνάρτησης *burn()*).	–
function *_isApprovedOrOwner*(address spender, uint256 tokenId) internal view virtual returns (bool)	Επιστρέφει αν έχει το δικαίωμα ο spender να διαχειριστεί το *tokenId*. *Απαίτηση*: • Το tokenId πρέπει να υπάρχει.	–
function *_safeMint*(address to, uint256 tokenId) internal virtual	Δημιουργεί με ασφάλεια ένα νέο token και το μεταφέρει στη διεύθυνση *to*. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Το *tokenId* δεν πρέπει να υπήρχε από πριν (νέο tokenId). • Εάν το *to* είναι smart contract, αυτό θα πρέπει να υλοποιεί το ERC-165.	Ναι (Transfer event)
function *_safeMint*(address to, uint256 tokenId, bytes memory data) internal virtual	Παρόμοιο με το συνονόματο, με την προσθήκη του πεδίου data, όπου μεταφέρονται στους παραλήπτες στο ERC-165 contract. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Το tokenId δεν πρέπει να υπήρχε από πριν (νέο *tokenId*). • Η δ/ση *to* δεν μπορεί να είναι η 0x0.	Ναι (Transfer event)

[87] Μόλις μπουν σε block.

function *_mint*(address to, uint256 tokenId) internal virtual	Δημιουργεί (*mint*) το νέο token αποδίδοντάς του μια *tokenId* και το μεταφέρει στη διεύθυνση του *to*. Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Το *tokenId* δεν πρέπει να υπήρχε από πριν (νέο tokenId). • Η δ/ση *to* δεν μπορεί να είναι η 0x0. *Σημείωση*: Η χρήση της συνάρτησης δεν ενθαρρύνεται, αντίθετα ζητείται να προτιμάται η χρήση της *_safeMint()*.	Ναι (Transfer event)
function *_burn*(uint256 tokenId) internal virtual	Καταστρέφει το *tokenId*. Μαζί διαγράφεται και κάθε έγκριση για τη χρήση του token. Ενεργοποιεί το Transfer event. *Απαίτηση*: • Το *tokenId* πρέπει να υπάρχει.	Ναι (Transfer event)
function *_transfer*(address from, address to, uint256 tokenId) internal virtual	Μεταφέρει την ιδιοκτησία ενός NFT από τη διεύθυνση *from* στην *to*. Δεν έχει περιορισμούς για τον *msg.sender* (σε αντίθεση με την *transferFrom()*). Ενεργοποιεί το *Transfer* event. *Απαιτήσεις*: • Η δ/ση *to* δεν πρέπει να είναι ίση με 0x0. • Το *tokenId* πρέπει να υπάρχει και η δ/ση *from* να είναι ο ιδιοκτήτης του.	Ναι (Transfer event)
function *_approve*(address to, uint256 tokenId) internal virtual	Δίνει δικαίωμα διαχείρισης στον *to* πάνω στο *tokenId*. Ενεργοποιεί το *Approval* event.	Ναι (Approval event)
function *_setApprovalForAll*(address owner, address operator, bool approved) internal virtual	Δίνει δικαίωμα στον *operator* να διαχειρίζεται όλα τα tokens στην κατοχή του *owner*. Ενεργοποιεί το *ApprovalForAll* event.	Ναι (ApprovalForAll event)
function *_requireMinted*(uint256 tokenId) internal view virtual	Επιστρέφει αν η *tokenId* έχει γίνει minted ή όχι.	–
function *_beforeTokenTransfer*(address from, address to, uint256 tokenId) internal virtual	Συνάρτηση hook η οποία καλείται πριν από την όποια μεταφορά tokens (συμπεριλαμβανομένων και των mint και burn ενεργειών). *Απαιτήσεις*: • Όταν καμία από τις δ/σεις from και to δεν είναι η 0x0, τότε το *tokenId* μεταφέρεται από τη *from* στην *to*. • Αν η δ/ση *from* είναι η 0x0, τότε το *tokenId* δημιουργείται (mint) για τον *to*. • Αν η δ/ση *to* είναι η 0x0, τότε το *tokenId* καταστρέφεται (burn). • Οι δ/σεις *from* και *to* δεν πρέπει να είναι *μαζί* ίσες με 0x0.	–
function *_afterTokenTransfer*(address from, address to, uint256 tokenId) internal virtual	Συνάρτηση hook η οποία καλείται πριν από την όποια μεταφορά tokens (συμπεριλαμβανομένων και των mint και burn ενεργειών). *Απαιτήσεις*: • Όταν καμία από τις δ/σεις *from* και *to* δεν είναι η 0x0. • Οι δ/σεις *from* και *to* δεν πρέπει να είναι *μαζί* ίσες με 0x0.	–

Πίνακας 7.8 *Οι επιπρόσθετες συναρτήσεις που έχουν ενσωματωθεί στην υλοποίηση του ERC-721 από την OpenZeppelin με όνομα ERC721URIStorage.sol.*

Επιπρόσθετα, στο *ERC721URIStorage.sol* ο κώδικας που θα χρησιμοποιηθεί κάνει χρήση και ενός ακόμα smart contract, με όνομα *Counters.sol*. Σε αυτό το smart contract η OpenZeppelin έχει δημιουργήσει μια υλοποίηση που δημιουργεί counters οι οποίοι, μέσω του smart contract, είναι δυνατόν να αυξηθούν, να μειωθούν ή να μηδενιστούν. Είναι προσαρμοσμένοι για να δράσουν επικουρικά στην καταγραφή του αριθμού των ERC721 token IDs που θα δημιουργηθούν και με αυτή τη λογική χρησιμοποιούνται στον κώδικα της υλοποίησης που φαίνεται στην Εικόνα 7.4.

Τέλος, θα γίνει χρήση και του contract *Ownable.sol*, το οποίο επιτρέπει να προσδιοριστούν ορισμένες συναρτήσεις έτσι ώστε να μπορεί μόνο ο ιδιοκτήτης του contract (*onlyOwner*) να προβεί σε συγκεκριμένες ενέργειες, όπως, για παράδειγμα, να δημιουργήσει (mint) νέα tokens. Αυτό μπορεί να τροποποιηθεί, ανάλογα με τις ανάγκες της υλοποίησης.

Η Εικόνα 7.4 έχει τον κώδικα για την υλοποίηση της OpenZeppelin για να δοθεί μια εικόνα για το ποια μορφή μπορεί να πάρει ένα smart contract που δημιουργεί αντικείμενα ως Non-Fungible Token. Στο παράδειγμα θα χρησιμοποιηθούν τα δύο contracts που αναφέρθηκαν προηγουμένως. Το όνομα του token που θα δημιουργηθεί είναι ConsertLab και σε συντομογραφία θα γράφεται CLab.

Θα προσέξετε ότι από την υλοποίηση απουσιάζει εντελώς η έννοια των δεκαδικών (decimals) που παρουσιάστηκε στα ERC-20 tokens. Ο λόγος της απουσίας αυτής είναι ότι ένα ERC-721 δεν μπορεί να διαιρεθεί σε μικρότερα κομμάτια, όπως είναι δυνατόν με ένα ERC-20 token.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract ConsertLab is ERC721, ERC721URIStorage, Ownable {
        using Counters for Counters.Counter;
        Counters.Counter private _tokenIdCounter;

        constructor() ERC721("ConsertLab", "CLab") {}
        function safeMint(address to, string memory uri) public onlyOwner {
                uint256 tokenId = _tokenIdCounter.current();
                _tokenIdCounter.increment();
                _safeMint(to, tokenId);
                _setTokenURI(tokenId, uri);
}
// The following functions are overrides required by Solidity.
function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
}

function tokenURI(uint256 tokenId) public view override(ERC721, ERC721URIStorage) returns (string
memory) {
                return super.tokenURI(tokenId);
        }
}
```

Εικόνα 7.4 *Παράδειγμα δημιουργίας ενός ERC-721 token με χρήση της υλοποίησης από την OpenZeppelin.*

7.3.3 Το Πρότυπο ERC-777

Το Πρότυπο ERC-777 βασίζεται πάνω στην EIP-777 (2017) και αποτελεί ένα πρότυπο που προτείνει έναν νέο τρόπο αλληλεπίδρασης με τα contracts των Fungible Tokens, κρατώντας τη συμβατότητα με τις προηγούμενες υλοποιήσεις.

Στην EIP-777 παρουσιάζονται νέα, προχωρημένα χαρακτηριστικά για την αλληλεπίδραση με tokens. Χαρακτηριστικά αναφέρονται φορείς διαχείρισης (operators) οι οποίοι αναλαμβάνουν να αποστείλουν tokens εκ μέρους διευθύνσεων τρίτων (με λογαριασμό σύμβασης ή χρήστη). Τμήμα της διαχείρισης αυτής είναι η αυτόματη αποστολή ειδοποιήσεων στους ιδιοκτήτες με τη χρήση *hooks* για την κάθε αποστολή ή λήψη. Με αυτό τον τρόπο διευρύνουν τις δυνατότητες των ιδιοκτητών πάνω στα tokens τους. Ακόμα, επιτυγχάνει να διορθώσει την αβεβαιότητα που υπήρχε γύρω από τη χρήση των δεκαδικών (decimals), καθώς και για το minting νέων tokens και την καταστροφή αυτών (burn) με τη δημιουργία των κατάλληλων events.

Επιπλέον, αναφέρεται ότι χρησιμοποιεί το πρότυπο ERC-1820, το οποίο αποτελεί ενημέρωση του ERC-820 και στο οποίο περιγράφεται η δημιουργία ενός smart contract που θα παίζει τον ρόλο της κοινής βάσης καταγραφής (universal registry). Σε αυτή την κοινή βάση θα μπορούν να εγγράφονται είτε *λογαριασμοί χρηστών* (*user accounts*) είτε *λογαριασμοί συμβάσεων* (*contract accounts*) και να δηλώνουν ποιες διεπαφές υποστηρίζουν και ποιο smart contract είναι υπεύθυνο για την εφαρμογή τους. Με αυτόν τον τρόπο ο οποιοσδήποτε μπορεί να αναζητήσει στη βάση αυτή αν μια διεύθυνση μπορεί να υλοποιήσει ένα συγκεκριμένο πρότυπο και με ποιο contract το κάνει αυτό.

Ο λόγος που το ERC-777 συνδέεται με το ERC-1820 είναι για να μπορέσει να διατηρήσει τη συμβατότητά του με προηγούμενα συμβόλαια (π.χ. ERC-20), αλλά και για να είναι σε θέση να επιβεβαιώσει ότι μια διεύθυνση είναι συμβατή με αυτό για να την ειδοποιήσει όταν λάβει ένα token. Επιπλέον, διευθύνσεις οι οποίες δεν έχουν κάποιο hook εγγεγραμμένο στη βάση, με ενημέρωση για συμβατότητα με το ERC-777, αυτομάτως θεωρούνται μη συμβατές με αυτό και η όποια μεταφορά token ακυρώνεται. Με τον τρόπο αυτόν εξασφαλίζεται ότι δεν θα μεταφερθούν ERC-777 tokens σε κάποιο smart contract που δεν είναι συμβατό με το πρότυπο.

Πιο αναλυτικά, στην EIP-777 (2017) αναφέρονται τα εξής χαρακτηριστικά που προσδίδονται σε αυτήν:

- Χρησιμοποιεί την ίδια λογική για τη μεταφορά των tokens που ακολουθείται και για τη μεταφορά των ethers, δηλαδή μια συνάρτηση της μορφής: *send(dest, value, data)*.
- Είναι δυνατόν ένας λογαριασμός χρήστη ή σύμβασης να ελέγχει και να ακυρώνει ποιο token δέχεται ή αποστέλλει μέσω της εγγραφής στην αντίστοιχη συνάρτηση hook. *tokensReceived* για την παραλαβή και *tokensToSend* για την αποστολή είναι τα αντίστοιχα ονόματα των συναρτήσεων hooks. Και οι δύο συναρτήσεις hook πρέπει να έχουν τη δυνατότητα επιστροφής στην προηγούμενη κατάσταση (revert) για να υλοποιήσουν τη δηλωμένη άρνηση για τη μεταφορά.
- Η συνάρτηση hook *tokensReceived* επιτρέπει την αποστολή tokens σε ένα contract καθώς και την ταυτόχρονη ειδοποίηση αυτού για τη μεταφορά σε μία συναλλαγή. Σε αντίθεση με το ERC-20, το οποίο χρειαζόταν να τρέξει δύο συναρτήσεις (approve και transferFrom) για την ίδια πράξη (δηλαδή ενεργοποίηση του Transfer event).
- Ο κάτοχος των tokens μπορεί να ορίσει νέους διαχειριστές των tokens του ή να πάψει τους ήδη ορισμένους. Συνήθως ως διαχειριστές ορίζονται επιβεβαιωμένα contracts που ανήκουν, για παράδειγμα, σε ανταλλακτήρια ή σε συστήματα αυτόματης χρέωσης.
- Κάθε συναλλαγή από tokens περιέχει πεδία με όνομα *data* καθώς και *operatorData*, που επιτρέπουν την ελεύθερη διακίνηση δεδομένων από τον ιδιοκτήτη και τον διαχειριστή αντίστοιχα.
- Είναι συμβατά με προηγούμενες εκδόσεις πορτοφολιών (wallets) τα οποία και δεν υλοποιούν τη συνάρτηση hook *tokensReceived*. Για να υποστηρίξει τη συμβατότητα αυτή, εγκαθιστά ένα ενδιάμεσο contract το οποίο και την υλοποιεί για να εξυπηρετήσει το wallet.

Στον **Πίνακα 7.9** φαίνονται οι συναρτήσεις που πρέπει να εφαρμόζονται από κάθε token για να είναι συμβατό με το πρότυπο ERC-777.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
function *name()* external view returns (string memory)	Επιστρέφει το όνομα της συλλογής από tokens.	–
function *symbol()*external view returns (string memory)	Επιστρέφει το σύμβολο του token.	–
function *granularity()* external view returns (uint256);	Επιστρέφει τη μικρότερη ποσότητα από το token, η οποία δεν μπορεί να διαιρεθεί περισσότερο. Αυτό σημαίνει πως όλες οι λειτουργίες (δημιουργία, καταστροφή και μετακίνηση) θα πρέπει να γίνονται σε τιμές πολλαπλάσιες της τιμής αυτής. Η συνηθέστερη τιμή είναι ίση με 1.	–
function *totalSupply()* external view returns (uint256);	Επιστρέφει τον αριθμό των tokens που έχουν δημιουργηθεί.	–
function *balanceOf*(address owner) external view returns (uint256)	Επιστρέφει τον αριθμό των tokens που έχει στην κατοχή της μια συγκεκριμένη διεύθυνση (*owner*). Ο αριθμός αυτός επιστρέφεται ως ένας ακέραιος των 256 bits.	–
function *send*(address recipient, uint256 amount, bytes calldata data) external;	Μεταφέρει *amount* από tokens από τον λογαριασμό του ιδιοκτήτη σε αυτόν του *recipient*. Εφόσον οι συναρτήσεις hooks του ιδιοκτήτη και του *recipient* έχουν εγγραφεί για αποστολή και λήψη, τότε οι αντίστοιχες συναρτήσεις θα κληθούν με *data στο αντίστοιχο πεδίο* και κενό στο πεδίο operatorData. Ενεργοποιεί ένα *Sent* event. *Απαιτήσεις*: • Ο ιδιοκτήτης πρέπει να έχει τουλάχιστον *amount* από tokens. • Ο *recipient* δεν μπορεί να είναι η δ/ση 0x0. • Εάν ο *recipient* είναι contract, πρέπει να υποστηρίζει τη διεπαφή για τη λήψη ERC-777 tokens.	Ναι (Sent event)

function *burn*(uint256 amount, bytes calldata data) external;	Καταστρέφει *amount* από tokens από τον λογαριασμό του ιδιοκτήτη και μειώνει κατά το ποσό αυτό τον συνολικό αριθμό από διαθέσιμα tokens. Εφόσον η διεύθυνση του ιδιοκτήτη έχει καταχωρίσει τη συνάρτηση hook για την αποστολή, θα κληθεί η συνάρτηση με *data στο αντίστοιχο πεδίο* και κενό στο πεδίο operatorData. Ενεργοποιεί το Burned event. *Απαίτηση*: • Ο ιδιοκτήτης πρέπει να έχει τουλάχιστον *amount* από tokens.	Ναι (Burned event)
function *isOperatorFor*(address operator, address tokenHolder) external view returns (bool);	Επιστρέφει αν ο *operator* είναι (ή όχι) διαχειριστής του *tokenHolder*. Αρχικά, όλοι οι λογαριασμοί είναι διαχειριστές του εαυτού τους.	–
function *authorizeOperator*(address operator) external;	Μετατρέπει τον *operator* ως διαχειριστή του λογαριασμού που καλεί τη συνάρτηση. Ενεργοποιεί το *AuthorizeOperator* event. *Απαίτηση*: • Ο λογαριασμός κλήσης δεν πρέπει να είναι ο ίδιος με τον *operator*.	Ναι (AutorizeOperator event)
function *revokeOperator*(address operator) external;	Αφαιρεί τον ρόλο διαχειριστή από τον *operator* για τα tokens του λογαριασμού κλήσης. Ενεργοποιεί το *RevokeOperator* event. *Απαίτηση*: • Ο λογαριασμός κλήσης δεν πρέπει να είναι ο ίδιος με τον *operator*.	Ναι (RevokeOperator event)
function *defaultOperators*() external view returns (address[] memory);	Επιστρέφει τη λίστα με τους εξ ορισμού διαχειριστές. Αφορά όλα τα tokens, ακόμα και αν δεν έχει κληθεί η *authorizeOperator()*. Η λίστα δεν μπορεί να αλλάξει, αλλά μπορεί κάποιος να εκτελέσει τη *revokeOperator* σε κάποια διεύθυνση.	–
function *operatorSend*(address sender, address recipient, uint256 amount, bytes calldata data, bytes calldata operatorData) external;	Μεταφέρει *amount* από tokens από τον *sender* στον *recipient*. Ο καλών τη συνάρτηση πρέπει να έχει δικαιώματα διαχειριστή στον λογαριασμό του *sender*. Εφόσον οι συναρτήσεις hooks του *sender* και του *recipient* έχουν εγγραφεί για αποστολή και λήψη, τότε οι αντίστοιχες συναρτήσεις θα κληθούν με *data στο αντίστοιχο πεδίο* και *operatorData* αντίστοιχα. Ενεργοποιείται το *Sent* event. *Απαιτήσεις*: • Ο *sender* πρέπει να έχει τουλάχιστον *amount* από tokens. • Ο *sender* δεν μπορεί να είναι η δ/ση 0x0. • Ο καλών τη συνάρτηση πρέπει να είναι διαχειριστής του *sender*. • Ο *recipient* δεν μπορεί να είναι η δ/ση 0x0. • Εάν ο *recipient* είναι contract, πρέπει να υποστηρίζει τη διεπαφή για τη λήψη ERC-777 tokens.	Ναι (Sent event)
function *operatorBurn*(address account, uint256 amount, bytes calldata data, bytes calldata operatorData) external;	Καταστρέφει *amount* από tokens από τον λογαριασμό *account*. Ο καλών τη συνάρτηση πρέπει να έχει δικαιώματα διαχειριστή στον λογαριασμό του *account*. Εφόσον η διεύθυνση του *account* έχει καταχωρίσει τη συνάρτηση hook για την αποστολή, θα κληθεί η συνάρτηση με *data στο αντίστοιχο πεδίο* και *operatorData* αντίστοιχα. Ενεργοποιεί το *Burned* event. *Απαιτήσεις*: • Ο *account* πρέπει να έχει τουλάχιστον amount από tokens. • Ο *account* δεν μπορεί να είναι η δ/ση 0x0. • Ο καλών τη συνάρτηση πρέπει να είναι διαχειριστής του *account*.	Ναι (Burned event)

Πίνακας 7.9 *Δήλωση συναρτήσεων στην EIP-777.*

Εκτός όμως από τις συναρτήσεις, η EIP-777 ορίζει και 5 events, τα οποία και παρουσιάζονται στον **Πίνακα 7.10.**

Δήλωση Event	Περιγραφή
event *Minted*(address indexed operator, address indexed to, uint256 amount, bytes data, bytes operatorData);	Ενεργοποιείται με τη δημιουργία *amount* ποσού από tokens από τον *operator* που διατίθενται στον *to*. Στο event μπορούν να μπουν και επιπλέον *data* ή/και *operatorData*.
event *Burned*(address indexed operator, address indexed from, uint256 amount, bytes data, bytes operatorData);	Ενεργοποιείται όταν ο operator καταστρέψει *amount* από tokens από τη διεύθυνση του *account*. Στο event μπορούν να μπουν και επιπλέον *data* ή/και *operatorData*.
event *AuthorizedOperator*(address indexed operator, address indexed tokenHolder);	Ενεργοποιείται όταν ο *operator* γίνει διαχειριστής του *tokenHolder*.
event *RevokedOperator*(address indexed operator, address indexed tokenHolder);	Ενεργοποιείται όταν ο *operator* χάσει την άδεια διαχείρισης των tokens του *takenHolder*.
event *Sent*(address indexed operator, addressed indexed from, address indexed to, uint256 amount, bytes data, bytes operatorData);	Ενεργοποιείται κατά τη μεταφορά tokens από τον *operator* μεταξύ του *from* και *to*. Στο event μπορούν να μπουν και επιπλέον *data* ή/και *operatorData*.

Πίνακας 7.10 *Δήλωση events στην EIP-777.*

7.3.3.1 Παράδειγμα δημιουργίας token τύπου ERC-777

Στη συνέχεια, όπως επαναλήφθηκε και για τα δύο προηγούμενα πρότυπα για tokens που μελετήθηκαν, θα παρουσιαστεί ένα πρακτικό παράδειγμα υλοποίησης του προτύπου ERC-777, ακολουθώντας την υλοποίηση από την OpenZeppelin για αυτό.

Όπως τονίζεται σε αυτήν, η υλοποίηση δεν διαχειρίζεται τον τρόπο δημιουργίας των tokens[88] και, επομένως, η χρήση της συνάρτησης *_mint()* που εισάγεται για τη δημιουργία τους είναι επιβεβλημένη. Η συμβατότητα με το ERC-20 εξασφαλίζεται ακολουθώντας τις οδηγίες της EIP-777. Μάλιστα, τόσο το event Sent (ERC-777) όσο και το event Transfer (ERC-20) ενεργοποιούνται κανονικά κατά τη μεταφορά των tokens.

Επιπρόσθετα, η τιμή για το granularity (ERC-777) είναι εξαρχής ίση με το 1. Αυτό σημαίνει ότι δεν υπάρχουν περιορισμοί για τη μεταφορά, δημιουργία και καταστροφή των tokens, εξυπηρετώντας απόλυτα τη συμβατότητα με την ERC-20.

Στον **Πίνακα 7.11** φαίνονται οι συναρτήσεις που εισάγονται (επιπλέον αυτών της EIP-777) κατά την υλοποίηση της ERC-777 από την OpenZeppelin.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
constructor(string memory name_, string memory symbol_, address[] memory defaultoperators_)	Αρχικοποιεί το contract δίνοντας τιμές στα name_ και symbol_. Ο πίνακας των defaultOperators_ μπορεί να είναι κενός.	–
function *_mint*(address account, uint256 amount, bytes memory userData, bytes memory operatorData) internal virtual;	Δημιουργεί *amount* από tokens και τα αναθέτει στον account αυξάνοντας τον συνολικά διαθέσιμο αριθμό από tokens. Εφόσον η διεύθυνση του *account* έχει καταχωρίσει τη συνάρτηση hook για την αποστολή tokens, θα κληθεί η συνάρτηση με *data* και *operatorData* στο αντίστοιχο πεδίο. Ενεργοποιεί το Minted event (ERC-777) και το Transfer event (ERC-20). *Απαιτήσεις*: • Ο *account* δεν μπορεί να είναι η δ/ση 0x0. • Εάν ο *account* είναι contract, πρέπει να υποστηρίζει τη διεπαφή για τη λήψη ERC-777 tokens.	Ναι (Minted event) + (Transfer event)[ERC-20]

[88] Δεν υπάρχει αντίστοιχη συνάρτηση στην EIP-777 για τη διαδικασία αυτή, η οποία εξαρτάται από την υλοποίηση.

function *_mint*(address account, uint256 amount, bytes memory userData, bytes memory operatorData, bool require ReceptionAck) internal virtual;	Δημιουργεί *amount* από tokens και τα αναθέτει στον account, αυξάνοντας τον συνολικά διαθέσιμο αριθμό από tokens. Εφόσον η τιμή στο *ReceptionAck* είναι True και η διεύθυνση του *account* έχει καταχωρίσει τη συνάρτηση hook για την αποστολή tokens, θα κληθεί η συνάρτηση με *data* και *operatorData* στο αντίστοιχο πεδίο. Ενεργοποιεί το Minted event (ERC-777) και το Transfer event (ERC-20). *Απαιτήσεις*: • Ο *account* δεν μπορεί να είναι η δ/ση 0x0. • Εάν ο *account* είναι contract, πρέπει να υποστηρίζει τη διεπαφή για τη λήψη ERC-777 tokens.	Ναι (Minted event) + (Transfer event)[ERC-20]
function *_send*(address from, address to, uint256 amount, bytes memory userData, bytes memory operatorData, bool require ReceptionAck) internal virtual;	Χρησιμοποιείται για την αποστολή *amount* από tokens από τον *from* στον *to*. Εφόσον η διεύθυνση του *account* έχει καταχωρίσει τη συνάρτηση hook για την αποστολή tokens, θα κληθεί η συνάρτηση με *data* και *operatorData* στο αντίστοιχο πεδίο.	–
function *_burn*(address from, uint256 amount, bytes memory data, bytes memory operatorData) internal virtual;	Χρησιμοποιείται για την καταστροφή tokens.	–
function *_spendAllowance*(address owner, address spender, uint256 amount) internal virtual;	Ανανεώνει την έγκριση του *owner* προς τον *spender* βάσει του ποσού *amount* που ξοδεύτηκε. Δεν ενημερώνει την έγκριση αν αυτή έχει δοθεί για άπειρο ποσό. Ακυρώνεται η πράξη αν δεν υπάρχουν αρκετά διαθέσιμα tokens προς παραχώρηση έγκρισης. Μπορεί να ενεργοποιήσει Approval event.	{Approval event}

Πίνακας 7.11 *Οι επιπρόσθετες συναρτήσεις που έχουν ενσωματωθεί στην υλοποίηση του ERC-777 από την OpenZeppelin.*

Ο κώδικας της υλοποίησης του ERC-777 token με όνομα ConsertLabToken.sol φαίνεται στην **Εικόνα 7.5**. Στον κώδικα αυτό φαίνεται πως μπορεί να δημιουργηθεί ένα ERC-777 token με το δικό του όνομα και σύμβολο. Για την εγκατάσταση του contract στο δίκτυο θα πρέπει να δοθεί τιμή για το initialSupply. Ο πίνακας με τους defaultOperators μπορεί αρχικά να είναι κενός.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC777/ERC777.sol";

contract ConsertLabToken is ERC777 {
    constructor(uint256 initialSupply, address[] memory defaultOperators)
        ERC777("ConsertLab", "CLab", defaultOperators)
    {
        _mint(msg.sender, initialSupply, "", "");
    }
}
```

Εικόνα 7.5 *Παράδειγμα δημιουργίας ενός ERC-777 token με χρήση της υλοποίησης από την OpenZeppelin.*

7.3.4 Το Πρότυπο ERC-1155

Το πρότυπο ERC-1155 είναι γνωστό και ως το πρότυπο που μπορεί να υποστηρίξει τη δημιουργία πολλών ειδών από tokens, παίζοντας έναν συγκεντρωτικό και ενισχυτικό ρόλο στα πρότυπα που αναλύθηκαν παραπάνω. Υποβλήθηκε αρχικά ως Πρόταση προς Βελτίωση στο Ethereum (EIP-1155, 2018) τον Ιούνιο του 2018, παρουσιάζοντας μια πρότυπη διεπαφή που μπορεί να διαχειριστεί πολλών ειδών tokens, τόσο Fungible όσο και Non-Fungible Tokens ή ακόμα και άλλους συνδυασμούς. Με τον τρόπο αυτόν το πρότυπο ERC-1155 μπορεί να

πετύχει την ίδια συμπεριφορά με τα ERC-20 και ERC-721 tokens, ξεχωριστά αλλά και ταυτόχρονα. Μάλιστα, μπορεί να το κάνει βελτιώνοντας την απόδοση των προτύπων αυτών.

Οι λόγοι που ένα τέτοιο συνδυαστικό (και όχι μόνο) πρότυπο παρουσιάζεται σχετίζονται με το γεγονός ότι η χρήση ξεχωριστών προτύπων και συμβολαίων, ανάλογα με το είδος των tokens, μπορεί να είναι πετυχημένη, αλλά επιβαρύνει το Ethereum με bytecodes από δεδομένα, επηρεάζοντας και την απόδοσή του. Επιπλέον, οι αποκεντρωμένες εφαρμογές (DApps), όπως είναι, για παράδειγμα, τα παιχνίδια που βασίζονται στην τεχνολογία του blockchain, δύνανται να δημιουργήσουν έναν σημαντικό αριθμό από tokens που θα χρειάζονται ένα νέο πρότυπο για να τα εξυπηρετήσει.

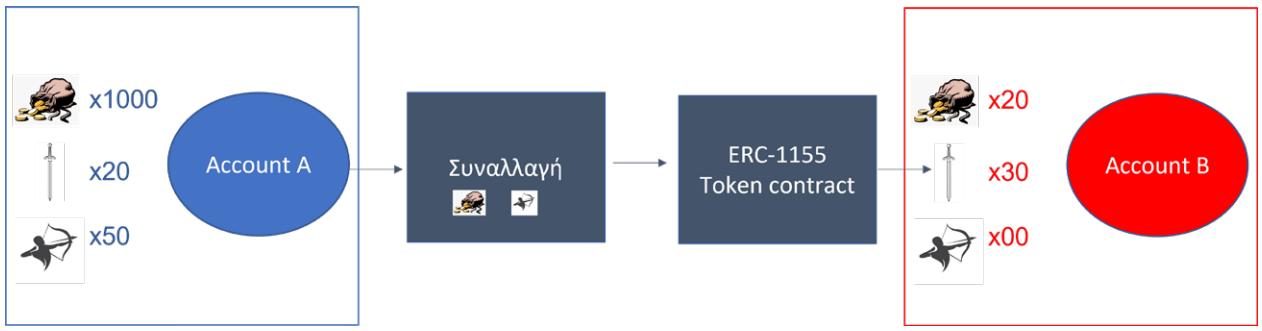
Έτσι, λοιπόν, στην ERC-1155, χωρίς αυτή να είναι προσανατολισμένη σε παιχνίδια και μόνο, παρουσιάστηκε η πρότυπη διεπαφή που έχει ως ιδιαίτερο χαρακτηριστικό της ότι μπορεί με ένα smart contract να εκπροσωπήσει πολλαπλά tokens. Για να το πετύχει αυτό, διαφέρει από τα ERC-20 και ERC-777 στο γεγονός ότι η συνάρτηση *balanceOf()* δεν λειτουργεί με τον ίδιο τρόπο όπως σε αυτά. Αντιθέτως, η συνάρτηση αυτή στο ERC-1155 περιλαμβάνει και μία επιπλέον παράμετρο, με την ονομασία *id*, για να παίξει τον ρόλο του διακριτικού που θα βοηθήσει στην υπόδειξη του token για το οποίο ζητείται να βρεθεί το υπόλοιπο (balance).

Η χρήση της παραμέτρου αυτής παρομοιάζεται καλύτερα με τη χρήση της αντίστοιχης μεταβλητής *id* που χρησιμοποιείται στο πρότυπο ERC-721. Εκεί το κάθε token που δημιουργείται παίρνει το δικό του id και η τιμή της μεταβλητής αυξάνεται με κάθε νέα δημιουργία. Θα πρέπει να θυμηθείτε ότι το πρότυπο ERC-721 δημιουργεί ξεχωριστά και μοναδικά tokens. Επομένως εκεί η συνάρτηση *balanceOf()* επιστρέφει τον συνολικό αριθμό από ξεχωριστά tokens (το καθένα με το δικό του id) που έχει στη διάθεσή του ο κάτοχος μιας συγκεκριμένης διεύθυνσης. Επιπλέον, σε κάθε id θα αντιστοιχεί ένα και μόνο token. Στην ERC-1155 ο ρόλος του id είναι αρκετά διαφορετικός, καθώς κάθε ξεχωριστό id μπορεί να αντιστοιχεί σε ένα σύνολο από tokens. Επομένως, το υπόλοιπο που θα επιστρέφει η συνάρτηση balanceOf() για ένα συγκεκριμένο id αφορά τον αριθμό των αντιγράφων (δηλαδή το πλήθος) από tokens που έχουν κοινό αριθμό στο id τους. Ο αριθμός αυτός μπορεί να είναι και μεγαλύτερος από 1.

Με την προσέγγιση αυτή, εφαρμογές οι οποίες απαιτούν τη δημιουργία διαφορετικού τύπου tokens δεν θα χρειάζεται να εγκαθιστούν διαφορετικά contracts για τη διαχείρισή τους. Αντίθετα, η εγκατάσταση ενός ERC-1155 contract θα είναι αρκετή για να πετύχουν τη δουλειά τους, εξασφαλίζοντας και μικρότερο κόστος όσον αφορά τα έξοδα για το gas.

Ένα ακόμα ιδιαίτερο χαρακτηριστικό του προτύπου είναι ότι το γεγονός πως χρειάζεται ένα μόνο contract για να λειτουργήσει του επιτρέπει να εκτελέσει ενέργειες σε περισσότερα από ένα tokens με κάθε συναλλαγή (η δυνατότητα αυτή ενεργοποιείται με τη λέξη Batch). Τα tokens μπορούν να επιλεχθούν με τη βοήθεια της απόδοσης του id τους και στο πρότυπο έχουν συμπεριληφθεί συναρτήσεις (π.χ. *safeBatchTransferFrom*) οι οποίες εκτελούν αυτή την απόδοση ενεργειών σε περισσότερα από 1 id σε μία συναλλαγή.

Στην **Εικόνα 7.6** φαίνεται ένα τέτοιο παράδειγμα. Σε αυτή λογαριασμοί χρηστών που συμμετέχουν σε ένα παιχνίδι έχουν στην ιδιοκτησία τους διαφορετικά αντικείμενα με διαφορετικά χαρακτηριστικά. Εφόσον τα tokens αυτά υποστηρίζουν το πρότυπο ERC-1155, τότε θα μπορούσε να γίνει η μεταφορά 2 αντικειμένων (tokens) από τον λογαριασμό Α στον λογαριασμό του Β.



Εικόνα 7.6 *Παράδειγμα χρήσης batch μεταφοράς ERC-1155 tokens σε μια αποκεντρωμένη εφαρμογή.*

Τέλος, στην EIP-1155 αναφέρεται ότι θα πρέπει να υποστηρίζεται και η πρότυπη διεπαφή ERC-165, και πιο συγκεκριμένα η συνάρτηση *supportsInterface()* η οποία θα επιστρέφει τη σταθερή τιμή True εφόσον η κωδική τιμή **`0xd9b67a26`** δίνεται στην παράμετρο *interfaceID*. Περισσότερες λεπτομέρειες στην EIP-165 (2018).

Στη συνέχεια, στον **Πίνακα 7.12**, παρουσιάζονται οι συναρτήσεις που περιέχονται στην EIP-1155.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
function *balanceOf*(address account, uint256 id) external view returns (uint256);	Επιστρέφει τον αριθμό των tokens που υπάρχουν για κάθε *id* και που ανήκουν στον *owner* (μπορεί να είναι διαφορετικού τύπου tokens) *Απαίτηση*: • Ο *owner* δεν μπορεί να είναι η δ/ση 0x0.	–
function *balanceOfBanch*(address[] calldata accounts, uint256[] calldata ids) external view returns (uint256[] memory);	Αποτελεί την batch έκδοση της προηγούμενης εντολής. Δηλαδή λειτουργεί όπως και η απλή έκδοση, αλλά τρέχει για έναν αριθμό από *owners > 1*. *Απαίτηση*: • Θα πρέπει ο αριθμός των εισόδων στον πίνακα *accounts* να είναι ίσος με αυτόν στον πίνακα *ids*.	–
function *setApprovalForAll*(address operator, bool approved) external;	Προσδίδει ή αφαιρεί την άδεια στον *operator* να μεταφέρει τα tokens του λογαριασμού που κάλεσε τη συνάρτηση, σύμφωνα και με την *approved*. Ενεργοποιεί το ApprovalForAll event. *Απαίτηση*: • Η συνάρτηση δεν μπορεί να έχει κληθεί από τον *operator*.	Ναι (ApprovalForAll event)
function *isApprovedForAll*(address account, address operator) external view returns (bool);	Επιστρέφει True αν ο *operator* έχει πάρει έγκριση να διαχειριστεί τα tokens του account.	–
function *safeTransferFroml*(address from, address to, uint256 id, uint256 amount, bytes calldata data) external;	Επιστρέφει *amount* από tokens τύπου *id* από τον *from* στον *to*. Ενεργοποιεί το TransferSingle event. *Απαιτήσεις*: • Η to δεν μπορεί να είναι η δ/ση 0x0. • Εάν αυτός που καλεί τη συνάρτηση δεν είναι ο *from*, τότε θα πρέπει να έχει πάρει έγκριση για τη μεταφορά μέσω της setApprovalForAll(). • Ο *from* θα πρέπει να έχει έναν αριθμό από tokens με το δεδομένο id, που θα πρέπει να είναι τουλάχιστον ίσος με *amount*. • Εάν το *to* είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferSingle event)
function *safeBatchTransferFroml*(address from, address to, uint256[] calldata id, uint256[] calldata amounts, bytes calldata data) external;	Αποτελεί την batch έκδοση της προηγούμενης εντολής. Δηλαδή εκτελεί μεταφορές για περισσότερα ids και amounts. Ενεργοποιεί το TransferBatch event. *Απαιτήσεις*: • Οι πίνακες *id* και *amount* πρέπει να έχουν το ίδιο μήκος. • Εάν το το είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferBatch event)

Πίνακας 7.12 *Δήλωση συναρτήσεων στην EIP-1155.*

Ταυτόχρονα, ορίζονται και τα 4 events που φαίνονται στον **Πίνακα 7.13**.

Δήλωση Event	Περιγραφή
event *TransferSingle*(address indexed operator, address indexed from, address indexed to, uint256 id, uint256 value);	Ενεργοποιείται όταν *amount* από tokens τύπου token *id* μεταφέρονται από τον *operator* από τον *from* στον *to*.
event *TransferBatch*(address indexed operator, address indexed from, address indexed to, uint256[] ids, uint256[] values);	Αντίστοιχο με το event TransferSingle, στο οποίο οι *owners*, *from* και *to*, είναι οι ίδιοι για όλες τις μεταφορές.
event *ApprovalForAll*(address indexed account, address indexed operator, bool approved);	Ενεργοποιείται όταν ο *account* ενεργοποιεί ή αφαιρεί την έγκριση διαχείρισης στον *operator*, σύμφωνα και με την approved.
event *URI*(string value, uint256 indexed id);	Ενεργοποιείται όταν το URI για τον τύπο token με το διακριτικό *id* αλλάζει στην τιμή του *value*. Εάν ενεργοποιηθεί το event για ένα token *id*, τότε το πρότυπο εγγυάται ότι η τιμή της *value* θα είναι ίση με αυτήν που επιστρέφεται από το MetadataURI.uri.

Πίνακας 7.13 *Δήλωση events στην EIP-1155.*

Ο ορισμός και η περιγραφή της συνάρτησης *uri* στην επέκταση MetadataURI, που αναφέρθηκε προηγουμένως στο event URI, φαίνεται στον **Πίνακα 7.14**.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
function *uri*(uint256 id) external view returns (string memory);	Επιστρέφει το URI για το τύπο token *id*.	–

Πίνακας 7.14 *Δήλωση συνάρτησης στην επέκταση MetadataURI στην EIP-1155.*

7.3.4.1 Παράδειγμα δημιουργίας token τύπου ERC-1155

Στη συνέχεια παρουσιάζεται ο κώδικας για τη δημιουργία ενός token που ακολουθεί το πρότυπο ERC-1155. Όπως και στις προηγούμενες υλοποιήσεις που παρουσιάστηκαν προηγουμένως, έτσι και εδώ έχει επιλεχθεί η υλοποίηση της OpenZeppelin για τον κώδικα του παραδείγματος.

Όπως φαίνεται και στην **Εικόνα 7.7**, χρησιμοποιούνται δύο contracts, που εισάγονται για την υλοποίηση. Το ένα (ERC1155.sol) είναι το contract υλοποίησης της EIP-1155 από την OpenZeppelin, ενώ το δεύτερο (Ownable.sol) χρησιμοποιήθηκε και στην υλοποίηση του ERC-721 token και αφορά τη δυνατότητα ελέγχου πρόσβασης σε ορισμένες συναρτήσεις έτσι ώστε να μπορεί μόνο ο ιδιοκτήτης του contract να έχει πρόσβαση σε αυτές.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract ConsertLab is ERC1155, Ownable {
    constructor() ERC1155("CLab") {}

    function setURI(string memory newuri) public onlyOwner {
        _setURI(newuri);
    }

    function mint(address account, uint256 id, uint256 amount, bytes memory data)
        public
        onlyOwner
    {
        _mint(account, id, amount, data);
    }

    function mintBatch(address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data)
        public
        onlyOwner
    {
        _mintBatch(to, ids, amounts, data);
    }
}
```

Εικόνα 7.7 *Παράδειγμα δημιουργίας ενός ERC-1155 token με χρήση της υλοποίησης από την OpenZeppelin.*

Πριν παρουσιαστεί ο κώδικας, θα γίνει μια παρουσίαση των συναρτήσεων που περιέχονται στην υλοποίηση από την OpenZeppelin του προτύπου της EIP-1155, οι οποίες επεκτείνουν τις ορισμένες στο πρότυπο συναρτήσεις. Αυτές οι επιπλέον συναρτήσεις φαίνονται στον **Πίνακα 7.15**.

Δήλωση Συνάρτησης	Περιγραφή	Ενεργοποιεί Event
constructor(string memory uri_);	Αρχικοποιεί το contract δίνοντας τιμές στο uri. Λεπτομέρειες πιο κάτω στη συνάρτηση _setURI.	–
function *_safeTransferFroml*(address from, address to, uint256 id, uint256 amount, bytes memory data) internal virtual;	Επιστρέφει *amount* από tokens τύπου *id* από τον *from* στον *to*. Ενεργοποιεί το TransferSingle event. *Απαιτήσεις*: • Η *to* δεν μπορεί να είναι η δ/ση 0x0. • Ο *from* θα πρέπει να έχει έναν αριθμό από tokens με το δεδομένο *id* που θα πρέπει να είναι τουλάχιστον ίσος με *amount*. • Εάν το *to* είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferSingle event)
function *_safeBacthTransferFroml*(address from, address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data) internal virtual;	Αποτελεί την batch έκδοση της προηγούμενης εντολής. Δηλαδή εκτελεί μεταφορές για περισσότερα ids και amounts. Ενεργοποιεί το TransferBatch event. *Απαίτηση*: • Εάν το *to* είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferBatch event)
function *_setURI*(string memory newuri) internal virtual	Αποδίδει ένα νέο URI σε κάθε τύπο token, βασίζοντας τον μηχανισμό αντικατάστασης στο id, όπως δηλώνεται και στην EIP. Με τον μηχανισμό αυτό, σε κάθε σημείο στο URI ή στο αρχείο JSON αναφέρεται ως string το id θα αντικαθίσταται από το ID του τύπου token. Επειδή τα URIs που παράγονται δεν έχουν ουσιαστική απεικόνιση, δεν ενεργοποιείται κάποιο event.	–
function *_mintl*(address to, uint256 id, uint256 amount, bytes memory data) internal virtual;	Δημιουργεί *amount* από tokens τύπου *id* και τα δίνει στη διεύθυνση *to*. Ενεργοποιεί το TransferSingle event. *Απαιτήσεις*: • Η *to* δεν μπορεί να είναι η δ/ση 0x0. • Εάν το *to* είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferSingle event)
function *_mintBatchl*(address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data) internal virtual;	Η batch έκδοση της προηγούμενης εντολής. Ενεργοποιεί το TransferBatch event. *Απαιτήσεις*: • Οι πίνακες *ids* και *amounts* πρέπει να έχουν το ίδιο μήκος. • Εάν το *to* είναι ένα smart contract, θα πρέπει αυτό να υποστηρίζει τη συνάρτηση hook για τη λήψη tokens της EIP-1155 και να επιστρέφει την αναμενόμενη τιμή αποδοχής.	Ναι (TransferBatch event)
function *_mintl*(address from, uint256 id, uint256 amount) internal virtual;	Καταστρέφει *amount* από tokens τύπου *id* που ανήκουν στον *from*. Ενεργοποιεί το TransferSingle event. *Απαιτήσεις*: • Η *from* δεν μπορεί να είναι η δ/ση 0x0. • Ο *from* πρέπει να έχει στην κατοχή του *amount* από tokens τύπου *id*.	Ναι (TransferSingle event)
function *_burnBatchl*(address from, uint256[] memory ids, uint256[] memory amounts) internal virtual;	Η batch έκδοση της προηγούμενης εντολής. Ενεργοποιεί το TransferBatch event. *Απαίτηση*: • Οι πίνακες *ids* και *amounts* πρέπει να έχουν το ίδιο μήκος.	Ναι (TransferBatch event)
function *_setApprovalForAll*(address owner, address operator, bool approved) internal virtual;	Προσδίδει την άδεια στον *operator* να διαχειριστεί όλα τα tokens του *owner*. Ενεργοποιεί το ApprovalForAll event.	Ναι (ApprovalForAll event)

function *_beforeTokenTransfer*(address operator, address from, address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data) internal virtual;	Συνάρτηση hook που καλείται πριν από την όποια μεταφορά tokens. Αυτό καλύπτει και τις διαδικασίες γέννησης και καταστροφής (mint, burn). Η ίδια συνάρτηση καλείται για τις απλές αλλά και τις μαζικές (batch) μεταφορές. Στις απλές μεταφορές το μήκος των πινάκων *ids* και *amounts* είναι ίσο με 1. *Απαιτήσεις (για κάθε ζεύγος ids και amount)*: • Όταν οι δ/σεις *from* και *to* δεν είναι ίσες με 0x0, τότε *amount* από tokens του *from* τύπου *id* θα μεταφερθούν στον *to*. • Αν η *from* είναι ίση με *0x0*, τότε *amount* από tokens τύπου id θα γεννηθούν για τον *to*. • Αν η *to* είναι ίση με *0x0*, τότε *amount* από tokens τύπου id που ανήκουν στον *from* θα καταστραφούν. • Οι δ/σεις *from* και *to* δεν είναι ποτέ ταυτόχρονα ίσες με *0x0*. • Οι πίνακες *ids* και *amounts* πρέπει να έχουν το ίδιο μη μηδενικό μήκος.	–
function *_afterTokenTransfer*(address operator, address from, address to, uint256[] memory ids, uint256[] memory amounts, bytes memory data) internal virtual;	Συνάρτηση hook που καλείται μετά την όποια μεταφορά tokens. Αυτό καλύπτει και τις διαδικασίες γέννησης και καταστροφής (mint, burn). Η ίδια συνάρτηση καλείται για τις απλές αλλά και τις μαζικές (batch) μεταφορές. Στις απλές μεταφορές το μήκος των πινάκων *ids* και *amounts* είναι ίσο με 1. *Απαιτήσεις (για κάθε ζεύγος ids και amount)*: • Όταν οι δ/σεις *from* και *to* δεν είναι ίσες με 0x0, τότε *amount* από tokens του *from* τύπου *id* θα μεταφερθούν στον *to*. • Αν η *from* είναι ίση με *0x0*, τότε *amount* από tokens τύπου id θα γεννηθούν για τον *to*. • Αν η *to* είναι ίση με *0x0*, τότε *amount* από tokens τύπου id που ανήκουν στον *from* θα καταστραφούν. • Οι δ/σεις *from* και *to* δεν είναι ποτέ ταυτόχρονα ίσες με *0x0*. • Οι πίνακες *ids* και *amounts* πρέπει να έχουν το ίδιο μη μηδενικό μήκος.	–

Πίνακας 7.15 *Οι επιπρόσθετες συναρτήσεις που έχουν ενσωματωθεί στην υλοποίηση του ERC-1155 από την OpenZeppelin.*

Στη συνέχεια, χρησιμοποιώντας την επιλογή Wizard της OpenZeppelin, θα δημιουργηθεί ένα contract που θα επιτρέπει τη δημιουργία ERC1155 tokens. Ο κώδικας φαίνεται στην Εικόνα 7.7.

Στο contract αυτό έχουν συμπεριληφθεί τα δύο contracts που αναφέρθηκαν προηγουμένως (ERC1155.sol, Ownable.sol) και επίσης έχουν επιλεχθεί:

- η δυνατότητα να είναι mintable, που δημιούργησε τις συναρτήσεις mint() και mintBatch(),
- η δυνατότητα updatable URI, που δημιουργεί τη συνάρτηση setURI().

Μπορείτε να αναπαραγάγετε τον κώδικα στο Remix και να κάνετε deploy το contract και να αλληλεπιδράσετε με τις συναρτήσεις. Επίσης, μπορείτε να δημιουργήσετε και να δώσετε τα δικά σας URI, ανάλογα με την εφαρμογή που θέλετε να δημιουργήσετε και να τα δώσετε ως είσοδο στη συνάρτηση setURI για να δείτε και τα αποτελέσματα.

7.3.5 Συνοπτικά

Συνοψίζοντας με τα χαρακτηριστικά των διάφορων ειδών tokens (ERC-20, ERC-721, ERC-1155) που μελετήθηκαν ως ώρα, ακολουθεί ένας συγκεντρωτικός πίνακας.

Τεχνικές Προδιαγραφές	ERC-20	ERC-721	ERC-1155
Ευκολία στη χρήση	Μια ενέργεια ανά συναλλαγή.	Μια ενέργεια ανά συναλλαγή. Νέο contract για κάθε τύπο token.	Πολλαπλές ενέργειες ανά συναλλαγή. Ένα συμβόλαιο για κάθε τύπο.
Λειτουργίες burn και mint	Δεν ενσωματώνει υλοποίηση. Επιλογή του χρήστη.	Διαθέτει.	Διαθέτει.
Τύπος token που υποστηρίζεται	Fungible	NFT	Fungible + NFTs
Smart Contracts	Χρήση 1 smart contract.	Χρήση νέου contract για κάθε τύπο token.	Χρήση 1 smart contract για όλους τους τύπους token.
Αποτελεσματικότητα	Χρειάζεται περισσότερο αποθηκευτικό χώρο.	Μπορεί να χρειαστεί περισσότερο αποθηκευτικό χώρο.	Χρειάζεται λιγότερο αποθηκευτικό χώρο.
Μεταφορά tokens	Μεταφορά 1 token σε κάθε συναλλαγή.	Μεταφορά 1 token σε κάθε συναλλαγή.	Μεταφορά πολλών tokens σε κάθε συναλλαγή.
Τύπους Μεταφοράς	Μεταφορά αξίας ανάμεσα σε χρήστες.	Μεταφορά δικαιωμάτων ιδιοκτησίας.	Μεταφορά αξίας ή δικαιωμάτων.
Παραδείγματα Χρήσης	Binance Coin, OmiseGo.	*Decentraland, Cryptokitties.*	Εξαγοράσιμα κουπόνια αγορών.

Πίνακας 7.16 *Συνοπτική περιγραφή των τεχνικών προδιαγραφών για τα tokens ERC-20, ERC-721, ERC-1155.*

7.4 Άλλα πρότυπα για tokens

Παρακάτω θα παρουσιαστούν ορισμένα διαφορετικά πρότυπα tokens και οι πλατφόρμες στις οποίες αυτά αναπτύσσονται.

- *EOS*[89]: Η πλατφόρμα EOSIO προσφέρει τα βασικά χαρακτηριστικά μιας λύσης που βασίζεται στην τεχνολογία του blockchain, όπου συμπεριλαμβάνονται η συναίνεση για τη δημιουργία ενός block, η δημιουργία λογαριασμών, η αποστολή και η λήψη tokens, το δικαίωμα ψήφου και η δυνατότητα πολλαπλών υπογραφών σε μια συναλλαγή.
- Πρόκειται για μια πλατφόρμα ανοικτού κώδικα που χρησιμοποιεί τα δικά της contracts για τη δημιουργία των tokens. Έτσι, το smart contract που δίνεται για τη δημιουργία tokens στην πλατφόρμα EOSIO προσδιορίζει τη δομή και τις ενέργειες που επιτρέπονται για τη δημιουργία και διαχείριση των tokens στην πλατφόρμα. Επιπλέον, το επίσημο token της πλατφόρμας, το EOS, έχει δημιουργηθεί βασιζόμενο στο contract αυτό.
- *NEO*[90]: Πρόκειται για μια ακόμα πλατφόρμα ικανή να υποστηρίξει τη δημιουργία και εκτέλεση contracts που επιτρέπουν τη δημιουργία και διαχείριση tokens στο δίκτυό της.
  Μάλιστα, έχουν οριστεί 3 πρότυπα για τη διαχείριση tokens στο δίκτυο, τα εξής:
  - NEP-5: Προσφέρει τις οδηγίες προς ένα σύστημα για τον μηχανισμό αλληλεπίδρασης με smart contracts που δημιουργούν tokens.
  - NEP-11: Παρουσιάζει το πρότυπο για τη συγγραφή contracts για NFTs με χρήση της γλώσσας C#.
  - NEP-17: Αποτελεί ενημέρωση της NEP-5 και παρέχει, όπως και αυτή, τις οδηγίες για τη δημιουργία tokens στο NEO δίκτυο blockchain, με γενικευμένες οδηγίες για την αλληλεπίδραση με τα smart contracts που τα δημιουργούν.

[89] Online Σύνδεσμος: https://eos.io/
[90] Online Σύνδεσμος: https://neo.org/

- *Tezos*[91]: Στο δίκτυο της Tezos προτάσεις για τεχνικές βελτιώσεις γίνονται με τη μορφή Tezos Interoperability Proposals (TZIP), σε αντιστοιχία με το Ethereum. Στις προτάσεις αυτές βρίσκονται οι τεχνικές λεπτομέρειες για την υλοποίηση των νέων χαρακτηριστικών, οδηγίες για το πώς μπορούν να ολοκληρωθούν και παρουσίαση των βελτιώσεων που προσφέρουν.
  Στο δίκτυο της Tezos υπάρχουν τα ακόλουθα πρότυπα για tokens:
  - TZIP-7: Εφαρμόζει τις ενέργειες για τη μεταφορά tokens καθώς και για την έγκριση για την αποστολή τους από άλλους λογαριασμούς. Είναι παρόμοιο, ουσιαστικά, με το πρότυπο ERC-20 στο Ethereum.
  - TZIP-12: Σε αυτό προτείνεται μια ομοιόμορφη διεπαφή για tokens που καλύπτει τόσο τα fungible όσο και τα NFTs. Είναι αντίστοιχο με το ERC-1155 στο Ethereum.
- *Binance Chain*[92]: Πρόκειται για το δίκτυο blockchain της γνωστής εταιρείας που τρέχει ένα από τα πιο δημοφιλή ανταλλακτήρια κρυπτονομισμάτων. Σε αυτό υπάρχει το πρότυπο BEP-2, το οποίο περιγράφει ένα σύνολο από κανόνες που πρέπει να ακολουθούν τα tokens έτσι ώστε να είναι αποτελεσματικά στο δίκτυο και να αλληλεπιδρούν με αυτό. Πρόκειται για πρότυπο το οποίο αντιστοιχεί στο ERC-20. Υπάρχει και το πρότυπο BEP-20, το οποίο συμβαδίζει ακόμα περισσότερο με το ERC-20 και είναι συμβατό και με contracts στο Ethereum. Το BEP-20 πρότυπο αφορά όμως tokens τα οποία έχουν δημιουργηθεί στο δίκτυο της Binance Smart Chain.
- *Chainlink*[93]: Η Chainlink έχει γίνει ιδιαίτερα γνωστή για την εξειδίκευσή της στη δημιουργία oracles μηχανισμών οι οποίοι συνδέονται σε δημοφιλείς πλατφόρμες blockchain (π.χ. Ethereum, Solana). Επιπλέον, χρησιμοποιεί και το δικό της token, το οποίο ονομάζεται LINK και χρησιμοποιείται για την επιβράβευση των κόμβων της Chainlink όταν τραβούν δεδομένα για τα smart contracts. Ακολουθεί το πρότυπο ERC-677 (2017), το οποίο κληρονομεί από το ERC-20 και, επιπλέον, επιτρέπει στις συναλλαγές των tokens να έχουν και δεδομένα που θα μεταφέρονται στο contract.
- *Cardano*[94]: Το Cardano είναι ένα ακόμα δίκτυο blockchain το οποίο επιτρέπει τη δημιουργία tokens. Μάλιστα σε αυτό οι χρήστες δεν είναι αναγκαίο να γράψουν το δικό τους smart contract και να ακολουθήσουν κάποιο πρότυπο, καθώς τα tokens είναι εγγενή στο σύστημα και μπορούν να δημιουργηθούν εύκολα και απλά. Με τον ίδιο τρόπο δημιουργείται το ADA, που είναι το token που χρησιμοποιεί το Cardano στο δίκτυό του (αντίστοιχο με το ether στο Ethereum).

Πριν κλείσει η παρουσίαση άλλων προτύπων tokens που υπάρχουν είτε στο Ethereum είτε σε άλλα δίκτυα blockchain, θα γίνει μια αναφορά σε έναν καινούργιο τύπο token, με την ονομασία *Soulbound tokens*.

Τα Soulbound tokens προτάθηκαν στην εργασία του Butterin et al. (2022) (ιδρυτής του Ethereum) μαζί με τους E. G. Weyl, P. Ohlhaver, οι οποίοι προτείνουν τη δημιουργία ενός νέου τύπου από NFTs, τα οποία όμως δεν μπορούν να μεταφερθούν από έναν λογαριασμό σε έναν άλλον. Τα tokens αυτά θα περιέχουν όλα τα χαρακτηριστικά του ιδιοκτήτη τους στον κόσμο του Web 3.0 και, για τον λόγο αυτό, δεν θα μπορούν να μεταφερθούν σε άλλο άτομο. Θα παίζουν, ουσιαστικά, τον ρόλο της ψηφιακής ταυτότητας του ατόμου στον κόσμο του Web 3.0.

Πρόκειται για μια πρωτοποριακή ιδέα, η οποία είναι πολύ νέα και ακόμη βρίσκεται σε θεωρητικό στάδιο. Αναμένεται με ξεχωριστό ενδιαφέρον ο τρόπος που θα υλοποιηθεί και θα διαμορφωθεί στην πορεία.

---

[91] Online Σύνδεσμος: https://tezos.com/

[92] Online Σύνδεσμος: https://www.binance.com/en/blog/all/bnb-chain-blockchain-for-exchanging-the-world-304219301536473088

[93] Online Σύνδεσμος: https://chain.link/

[94] Online Σύνδεσμος: https://cardano.org/

Βιβλιογραφία

Weyl, E. G., Ohlhaver, P., & Buterin, V. (2022). *Decentralized Society: Finding Web3's Soul*. Διαθέσιμο στο SSRN: https://ssrn.com/abstract=4105763

Vogelsteller, F., & Buterin, V. (2015). *EIP-20 Token Standard*. Online πηγή: https://eips.ethereum.org/EIPS/eip-20 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Reitwießner, C., Johnson, N., Vogelsteller, F., Baylina, J., Feldmeier, K., & Entriken, W. (2018). *EIP-165: Standard Interface Detection*. Online πηγή: https://eips.ethereum.org/EIPS/eip-165 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Entriken, W., Shirley, D., Evans, J., & Sachs, N. (2018). *EIP-721 Non-Fungible Token Standard*. Online πηγή: https://eips.ethereum.org/EIPS/eip-721 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Dafflon, J., Baylina, J., & Shababi, T. (2017). *EIP-777 Token Standard*. Online πηγή: https://eips.ethereum.org/EIPS/eip-777 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Radomski, W., Cooke, A., Castonguay, P., Therien, J., Binet, E., & Sandford, R. (2018). *EIP-1155 Multi Token Standard*. Online πηγή: https://eips.ethereum.org/EIPS/eip-1155 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ethereum.org (2022). *Ethereum ERC-20 Token Standard*. Online πηγή: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ERC-677 (2017). Ellis, S. *ERC:transferAndCall Token Standard*. Online πηγή: https://github.com/ethereum/EIPs/issues/677 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Tang, N. (2021). *Ethereum ERC-20 Tokens – All you need to know*. Online πηγή: https://phemex.com/academy/erc20-token-ethereum [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 8

Αποκεντρωμένες Εφαρμογές (DApps)

Σύνοψη

Το Κεφάλαιο αυτό εστιάζει σε ένα από τα βασικά συστατικά της αρχιτεκτονικής δομής του επονομαζόμενου Web 3.0, τις Αποκεντρωμένες Εφαρμογές (Decentralized Applications, DApps). Έτσι, αφού γίνει μια εισαγωγή στο τι είναι το Web 3.0, παρουσιάζονται αναλυτικά η δομή μιας Αποκεντρωμένης Εφαρμογής (DApp) και τα ιδιαίτερα χαρακτηριστικά της σε σύγκριση με τις υπάρχουσες (κεντρικοποιημένες) εφαρμογές.

Με έμφαση στην πλατφόρμα του Ethereum, γίνεται ανάλυση των τεχνολογιών που συνδυάζονται για τη δημιουργία ενός σύγχρονου DApp, τι προσφέρει η κάθε τεχνολογία, καθώς και παρουσίαση εναλλακτικών λύσεων. Τέλος, γίνεται μια αναφορά στις ιστοσελίδες που φιλοξενούν εφαρμογές DApps και παρουσιάζονται, κατ' επιλογή, ορισμένα DApps που έχουν αποκτήσει σημαντική δημοφιλία.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2 και 3.

8.1 Αποκεντρωμένες εφαρμογές (DApps)

Στο Κεφάλαιο 2 έγινε μια πρώτη παρουσίαση για το τι είναι και ποια είναι τα χαρακτηριστικά ενός DApp. Εδώ γίνεται μια ευρύτερη παρουσίαση των ιδιαίτερων χαρακτηριστικών, εμπλουτισμένη με στοιχεία που είναι επίκαιρα στα σύγχρονα DApps.

Στη συνέχεια γίνεται μια γνωριμία με το περιβάλλον των Web 2.0 και Web 3.0, με έμφαση στο δεύτερο, όπου τα DApps υλοποιούνται και λειτουργούν. Η αναφορά στο Web 2.0 προκειμένου να γίνει κατανοητή η διαφορά λειτουργίας μιας εφαρμογής σε καθένα από τα περιβάλλοντα αυτά.

8.1.1 Τι είναι το Web 2.0

Η τρέχουσα έκδοση του Internet είναι γνωστή με το όνομα Web 2.0. Περιέχει όλα εκείνα τα χαρακτηριστικά που έχουν συντελέσει στη δημιουργία ενός Παγκόσμιου Ιστού στον οποίο η αλληλεπίδραση των χρηστών με το περιεχόμενο που διαμοιράζεται είναι συνεχής, εύκολη και άμεση.

Σε αυτό έχει παίξει σημαντικό ρόλο η δημοφιλία των κοινωνικών δικτύων καθώς και των υπηρεσιών στο Web, οι οποίες συχνά υποστηρίζονται από μεγάλες εταιρείες, π.χ. Google, Microsoft, YouTube, Facebook κτλ. Μάλιστα, για τη δημιουργία εφαρμογών στον κόσμο αυτό απαιτείται γνώση γλωσσών προγραμματισμού, όπως HTML5, JavaScript, CSS, Python κ.ά. (Vermaak, 2021· Cointelegraph, 2022· Kasireddy, 2021).

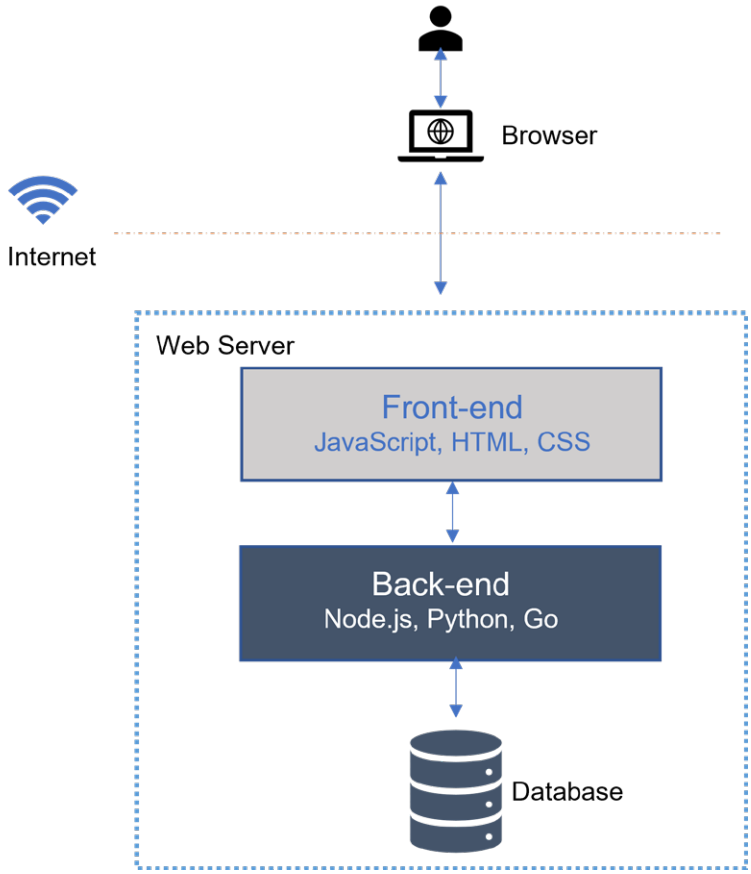
Ένα παράδειγμα μιας διαδεδομένης εφαρμογής στο Web 2.0 αποτελεί και το Reddit[95], το οποίο είναι μια κοινωνική πλατφόρμα περιεχομένου όπου ο κάθε χρήστης μπορεί να ανεβάσει ένα σχόλιο ή ένα αρχείο πολυμέσων και οι υπόλοιποι μπορούν να ψηφίσουν και να σχολιάσουν αν τους αρέσει ή όχι.

Για να λειτουργήσει το Reddit, ένα σύνολο από τεχνικά χαρακτηριστικά πρέπει να επιλυθεί. Το σύνολο αυτό των χαρακτηριστικών απεικονίζονται στην **Εικόνα 8.1**, που δείχνει και την αρχιτεκτονική μιας εφαρμογής στο Web 2.0.

Πιο συγκεκριμένα, χρειάζεται να βρεθεί λύση για:

- Την αποθήκευση των δεδομένων των χρηστών (ονόματα, κωδικοί, σχόλια, posts κτλ.)
- Την εκτέλεση του κώδικα ενεργειών της εφαρμογής. Πρόκειται για τον κώδικα στο backend ο οποίος και τρέχει όταν ανεβαίνει νέο περιεχόμενο από έναν χρήστη ή όταν κάποιος αλληλεπιδράσει με το περιεχόμενο αυτό. Συνήθως ο κώδικας αυτός έχει γραφτεί σε Node.js ή Java Python.
- Την εκτέλεση του κώδικα που εμφανίζει την εφαρμογή στον χρήστη για να αλληλεπιδράσει μαζί της (frontend). Για παράδειγμα, πώς θα φαίνεται το site και πώς θα απεικονίζεται η σελίδα του χρήστη ή η αλληλεπίδρασή του με ένα post. Αυτό το κομμάτι γράφεται συνήθως σε JavaScript, HTML, CSS.

[95] Online Σύνδεσμος: https://www.reddit.com/



Εικόνα 8.1 *Η αρχιτεκτονική μιας Web 2.0 εφαρμογής.*

Ο συνδυασμός των παραπάνω εκτελείται κάθε φορά που ένας χρήστης αλληλεπιδρά με την εφαρμογή στο κινητό ή στον υπολογιστή του (front-end). Στη συνέχεια, τα δεδομένα αυτά πηγαίνουν στο back-end. Εκεί συλλέγονται και αποθηκεύονται στη Βάση Δεδομένων στον server που φιλοξενεί και το back-end κομμάτι της εφαρμογής. Ταυτόχρονα, αν τα δεδομένα αυτά οδηγούν σε μια απάντηση, αυτή θα ξεκινήσει τη λειτουργία της, που θα καταλήξει στο front-end και στην οθόνη του χρήστη.

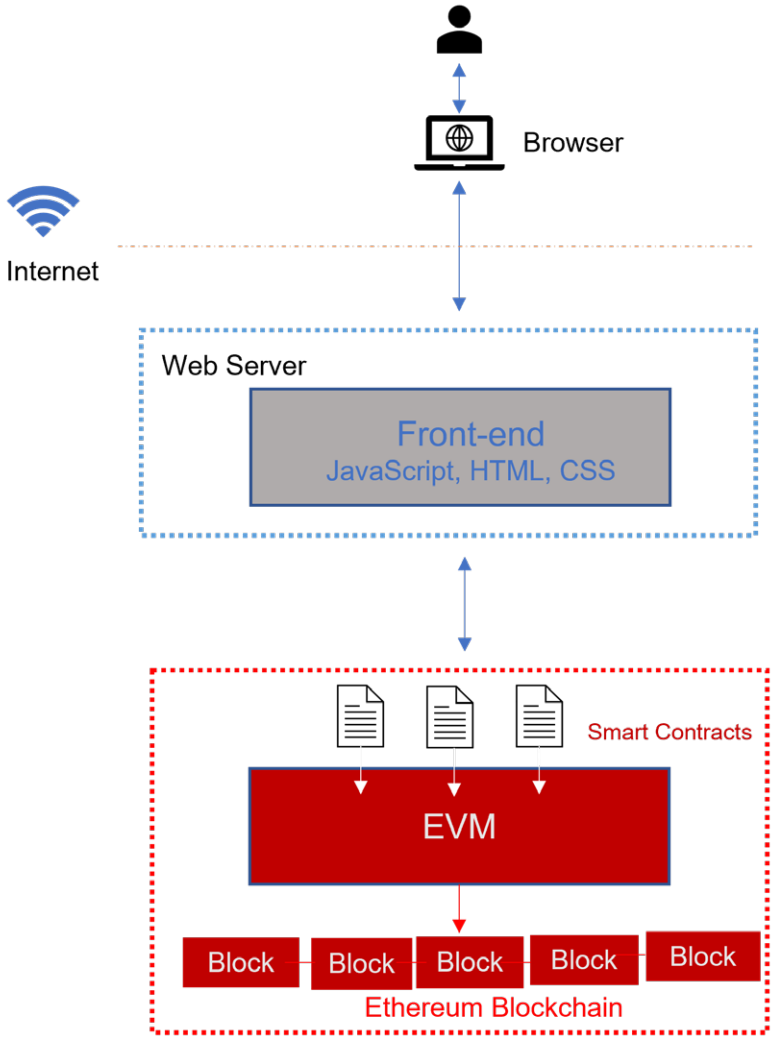
Επομένως, πρόκειται για μια κεντρικοποιημένη αρχιτεκτονική που βασίζεται σε γνωστά σημεία στο Διαδίκτυο, τους servers που διαχειρίζονται ή/και αποθηκεύουν τα δεδομένα του front-end. Όπως ακριβώς φαίνεται και στην Εικόνα 8.1.

Με τον ερχομό όμως του Web 3.0, αυτή η αρχιτεκτονική αλλάζει, όπως φαίνεται παρακάτω.

8.1.2 DApps στο Web 3.0

Το Web 3.0 έρχεται να αλλάξει σημαντικά την αρχιτεκτονική που φαίνεται στην Εικόνα 8.1 εισάγοντας μια νέα δομή και έναν καινούργιο τρόπο λειτουργίας. Ο πιο σημαντικός τρόπος με τον οποίο το κατορθώνει αυτό είναι με την αποκεντρωμένη λειτουργία. Αυτή εφαρμόζεται με τη βοήθεια κατανεμημένων servers που συνεργάζονται ή, συνήθως, με τη βοήθεια και υποστήριξη από ένα δίκτυο blockchain, που είναι από τη φύση του κατανεμημένο.

Η **Εικόνα 8.2** δείχνει την αρχιτεκτονική μιας εφαρμογής που βασίζεται στο Web 3.0 και υλοποιεί τα εγγενή του χαρακτηριστικά με τη βοήθεια του δικτύου του Ethereum, μιας και αυτό αποτελεί τη βάση για όσα συζητούνται στο κεφάλαιο αυτό.



Εικόνα 8.2 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής.*

Όπως φαίνεται και στην Εικόνα 8.2, η βασική διαφορά βρίσκεται στο back-end της αρχιτεκτονικής, καθώς πλέον αυτό δεν αποθηκεύεται σε έναν server, αλλά, αντιθέτως, βρίσκεται σε ένα σύνολο από κόμβους, οι οποίοι συνεισφέρουν εξίσου. Ουσιαστικά υπάρχει μια αφαίρεση του ενδιάμεσου παράγοντα και η επικοινωνία γίνεται απευθείας με το δίκτυο του blockchain, που βρίσκεται σε πολλούς χρήστες/κόμβους και το οποίο ενσωματώνει όλα τα απαραίτητα χαρακτηριστικά για την υποστήριξη της εφαρμογής Web 3.0.

Σε αυτή τη βάση σημαντικό ρόλο παίζει η δυνατότητα χρήσης της EVM του Ethereum από τους κόμβους, έτσι ώστε να είναι δυνατή η αποθήκευση των μεταβάσεων στις καταστάσεις οι οποίες προκαλούνται από την εκτέλεση του κώδικα στα smart contracts (βλ. Κεφάλαιο 2). Τα smart contracts εξάλλου αποτελούν το back-end τμήμα μιας εφαρμογής στο Web 3.0. Αποτελούν, δηλαδή, το μέσο που υλοποιεί τις ενέργειες που καλούνται από το front end κομμάτι της εφαρμογής. Ο κάθε κόμβος αποθηκεύει τα smart contracts που εγκαθίστανται, έτσι ώστε να είναι δυνατή η επιβεβαίωση από αυτόν της καθεμίας συναλλαγής που μπαίνει στο σύστημα ακολουθώντας τους αυστηρούς κανόνες συναίνεσης του κατανεμημένου δικτύου (π.χ. PoW ή PoS, ανάλογα με την έκδοση του πρωτοκόλλου στο Ethereum).

- Επικοινωνία με κόμβο δικτύου blockchain:

Δεδομένης της αλλαγής που φέρνει η κατανεμημένη φύση του συστήματος στην αρχιτεκτονική μιας Web 3.0 εφαρμογής, υπάρχει και μία ακόμα σημαντική διαφορά, που δεν είναι ευδιάκριτη στην Εικόνα 8.2. Αυτή σχετίζεται με την επικοινωνία μεταξύ του front-end και του δικτύου blockchain που βρίσκεται σε κάθε κόμβο. Όταν ένα smart contract καλείται (από μια ενέργεια του χρήστη στο front-end), αυτό σημαίνει ότι η εφαρμογή θα πρέπει να συνδεθεί σε έναν κόμβο του δικτύου blockchain και κατόπιν να περάσει σε αυτόν την

αλληλεπίδραση που θα έρθει από το front-end. Στον κόμβο θα κληθεί το smart contract και θα εκτελέσει την προβλεπόμενη ενέργεια. Αυτή η ενέργεια μπορεί είτε να γράψει κάτι (με τη μορφή συναλλαγής) στο ledger του δικτύου blockchain, αλλάζοντας την κατάσταση του κόσμου σε αυτό, να αναζητήσει κάτι στο ledger ή, ακόμα, και να καλέσει ένα νέο smart contract που είναι αποθηκευμένο στον (κάθε) κόμβο του δικτύου. Κατόπιν είναι σειρά του κόμβου αυτού να ενημερώσει τους υπόλοιπους κόμβους στο δίκτυο για την ενέργεια αυτή, ειδικά αν έχει επίδραση στην κατάσταση του δικτύου με τη δημιουργία μιας συναλλαγής. Έτσι, ολόκληρο το δίκτυο θα ενημερωθεί και θα εγκρίνει (ή όχι) τη συναλλαγή επικαιροποιώντας την κατάσταση του κόσμου αυτού. Αυτό, λοιπόν, που απουσιάζει από την Εικόνα 8.2 είναι ο τρόπος που εκτελείται αυτή η επικοινωνία του front-end με το back-end και το δίκτυο blockchain.

Στην πράξη, η επικοινωνία αυτή μπορεί να ολοκληρωθεί με δύο τρόπους (Kasireddy, 2021):

- Με τη δημιουργία και υποστήριξη ενός δικού σας κόμβου του blockchain του Ethereum (ή του δικτύου που θα επιλέξετε να εγκαταστήσετε το smart contract σας).
- Με τη χρήση κόμβων που παρέχονται ως υπηρεσία από τρίτους, όπως, για παράδειγμα, από την εταιρεία Infura[96], την Alchemy[97] και την Quicknode[98].

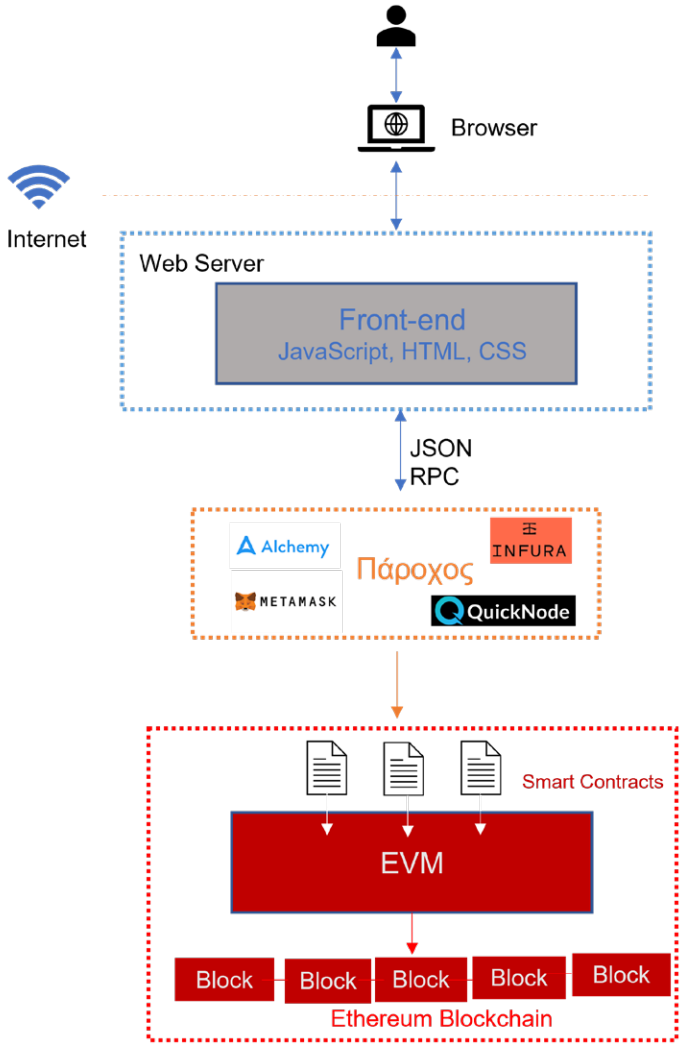
Η πρώτη επιλογή, της διατήρησης ενός κόμβου στο επιθυμητό δίκτυο blockchain, είναι η πιο δύσκολη. Αυτό συμβαίνει γιατί απαιτεί τόσο τεχνική γνώση για τη δημιουργία και τη διαχείριση ενός κόμβου αλλά και οικονομική δυνατότητα για τη συντήρηση και λειτουργία του κόμβου. Με την πάροδο του χρόνου και καθώς δεδομένα συνεχώς θα προστίθενται στο ledger, πιθανώς θα χρειαστεί και παρέμβαση στο hardware του υπολογιστή/κόμβου, έτσι ώστε να είναι αυτός ικανός να αντεπεξέλθει στις νέες απαιτήσεις. Επιπλέον, ο χρόνος που χρειάζεται για την αρχική ενημέρωση του ledger είναι επίσης σημαντικός. Πάντως, η επίσημη σελίδα του Ethereum περιέχει πλούσιο υλικό, που μπορεί να χρησιμοποιηθεί για καθοδήγηση στη δημιουργία νέου κόμβου[99] σε όσους θεωρούν ότι αυτή είναι μια προσέγγιση που τους ταιριάζει καλύτερα.

Στην περίπτωση που κάποιος επιθυμεί μια διαφορετική προσέγγιση, υπάρχει και αυτή η δυνατότητα. Με την επιλογή αυτή, ο δημιουργός της εφαρμογής DApp θα χρειαστεί να συνεργαστεί με έναν πάροχο, που θα τρέχει εκείνος έναν κόμβο και ο δημιουργός της εφαρμογής θα πρέπει απλώς να συνδεθεί σε αυτόν για να εγκαταστήσει την εφαρμογή του στο δίκτυο.

Για την επικοινωνία με τον πάροχο συνήθως χρησιμοποιούνται ευρέως γνωστά πρότυπα που διευκολύνουν ομοιόμορφα τις εφαρμογές στο στάδιο αυτό. Έτσι, για την επικοινωνία με έναν Ethereum client εφαρμόζεται η προδιαγραφή JSON-RPC (2013), η οποία χρησιμοποιεί ένα ελαφρύ πρωτόκολλο απομακρυσμένης κλήσης το οποίο δεν κρατά καταστάσεις και προσδιορίζει πολλές δομές δεδομένων μαζί με τους κανόνες επεξεργασίας αυτών. Η μορφή των δεδομένων είναι JSON και δεν επηρεάζεται από το πρωτόκολλο μεταφοράς, πράγμα που του επιτρέπει να χρησιμοποιείται είτε με συνδέσεις με χρήση sockets είτε με συνδέσεις HTTP είτε με κάποιο άλλο πρωτόκολλο.

Η **Εικόνα 8.3** δείχνει πού τοποθετείται η χρήση του παρόχου για τη σύνδεση με έναν κόμβο του δικτύου στην αρχιτεκτονική μιας εφαρμογής Web 3.0.

[96] Online Σύνδεσμος: https://infura.io/
[97] Online Σύνδεσμος: https://www.alchemy.com/
[98] Online Σύνδεσμος: https://www.quicknode.com/
[99] Online Σύνδεσμος: https://ethereum.org/el/run-a-node/



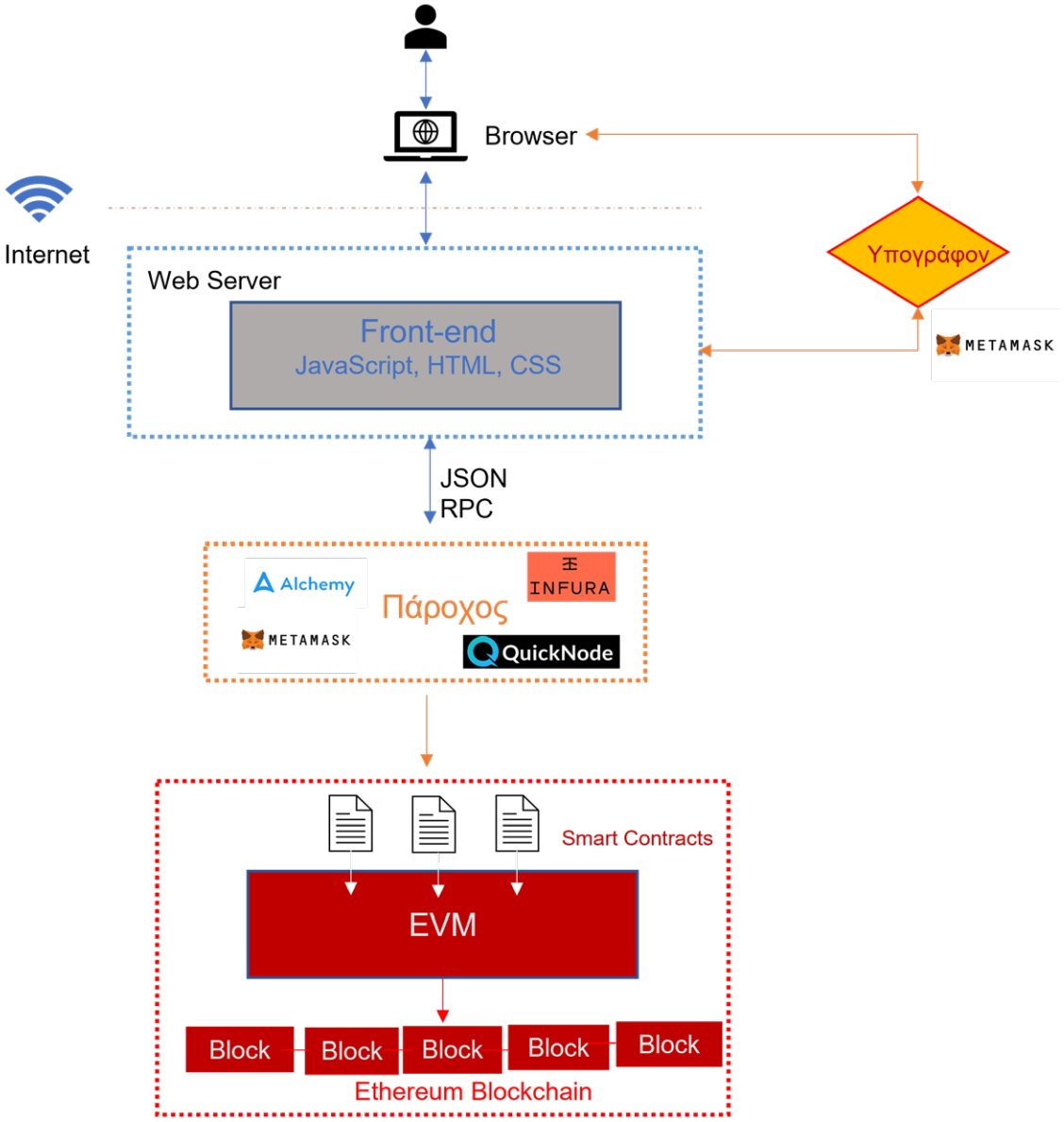
Εικόνα 8.3 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής με τη χρήση παρόχου για την προσφορά ενός κόμβου στο δίκτυο blockchain για την εγκατάσταση της εφαρμογής.*

- Εγγραφή στο ledger:

Με τη σύνδεση στον κόμβο με τη βοήθεια ενός παρόχου δίνεται η δυνατότητα στον χρήστη να διαβάσει τι υπάρχει στο ledger. Όμως ακόμα δεν έχει τη δυνατότητα να επέμβει και να γράψει κάτι σε αυτό. Για παράδειγμα, ο χρήστης μπορεί να χρησιμοποιήσει το DApp για να αναζητήσει τα posts ή τα σχόλια ενός άλλου χρήστη, καθώς για την απάντηση σε αυτό το ερώτημα απαιτείται μόνο η αναζήτηση στο ledger και όχι η δημιουργία κάποιας εγγραφής σε αυτό.

Για να του επιτραπεί κάτι τέτοιο, χρειάζεται να πιστοποιήσει την ταυτότητά του στο δίκτυο, να μπορέσει δηλαδή να υπογράψει τη συναλλαγή με την οποία θα προστεθούν τα δεδομένα στο ledger. Στα Κεφάλαια 2 και 3 έγινε αναφορά στα πορτοφόλια τα οποία χρησιμοποιούνται για τη διαδικασία υπογραφής μιας συναλλαγής προτού αυτή προωθηθεί στο δίκτυο και επιλεγεί για να μπει σε ένα νέο block. Το Metamask μπορεί να χρησιμοποιηθεί και στο παράδειγμα που παρουσιάζεται εδώ, λόγω και της συμβατότητάς του με το δίκτυο του Ethereum.

Στην **Εικόνα 8.4** φαίνεται και η προσθήκη του Metamask. Αυτό καλείται από το front-end, καθώς εκεί δημιουργείται η συναλλαγή. Το front-end επικοινωνεί με τον browser στον οποίο είναι εγκατεστημένο (ως πρόσθετο συνήθως) το Metamask, ζητώντας από τον χρήστη να συνδεθεί στον λογαριασμό του για να μπορέσει να χρησιμοποιήσει το ιδιωτικό του κλειδί για να υπογράψει τη συναλλαγή. Ταυτόχρονα, όμως, το Metamask μπορεί να θεωρηθεί και πάροχος, καθώς ο χρήστης σε αυτό έχει τη δυνατότητα να επιλέξει το δίκτυο με το οποίο θα συνδεθεί για να πραγματοποιήσει τη συναλλαγή. Για τον λόγο αυτόν εμφανίζεται τόσο στην Εικόνα 8.3 όσο και στην Εικόνα 8.4 στο κουτί με τους παρόχους.



Εικόνα 8.4 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής με την προσθήκη παρόχου καθώς και πορτοφολιού (Metamask) για την υπογραφή των συναλλαγών.*

- Αποθήκευση δεδομένων off-chain:

Η αρχιτεκτονική μέχρι στιγμής, όπως φαίνεται στην Εικόνα 8.4, καταφέρνει και καλύπτει τις βασικές ανάγκες μιας εφαρμογής Web 3.0. Υπάρχει όμως και το θέμα της αποθήκευσης των δεδομένων. Καθώς στο ledger μπορεί κανείς μόνο να προσθέσει δεδομένα, δημιουργείται πρόβλημα με το ολοένα αυξανόμενο μέγεθος του ledger. Ταυτόχρονα, η αύξηση αυτή έχει επίδραση και στις συναρτήσεις των smart contracts που πρέπει να αναζητήσουν ή και να προσθέσουν δεδομένα στο ledger. Και αυτό εισάγει μια αυξανόμενη καθυστέρηση στην απόδοση του δικτύου και, κατ' επέκταση, της εφαρμογής. Επιπρόσθετα, η συγγραφή δεδομένων στο ledger κοστίζει, μιας και αυξάνεται ο χώρος που καλύπτεται από τα δεδομένα του δικτύου. Η χρέωση για την προσθήκη δεδομένων μπορεί να είναι προβληματική από αρκετούς χρήστες.

Για όλους αυτούς τους λόγους έχει αναπτυχθεί τελευταία η τάση να μπαίνουν μόνο ορισμένα δεδομένα στο ledger και τα υπόλοιπα να διατηρούνται εκτός αυτού. Η τοποθεσία των δεδομένων όμως αποθηκεύεται στο ledger, έτσι ώστε να μπορεί να γίνει επιβεβαίωση για την εγκυρότητά τους μέσω της παραγωγής ενός hash, όποτε αυτό χρειαστεί.

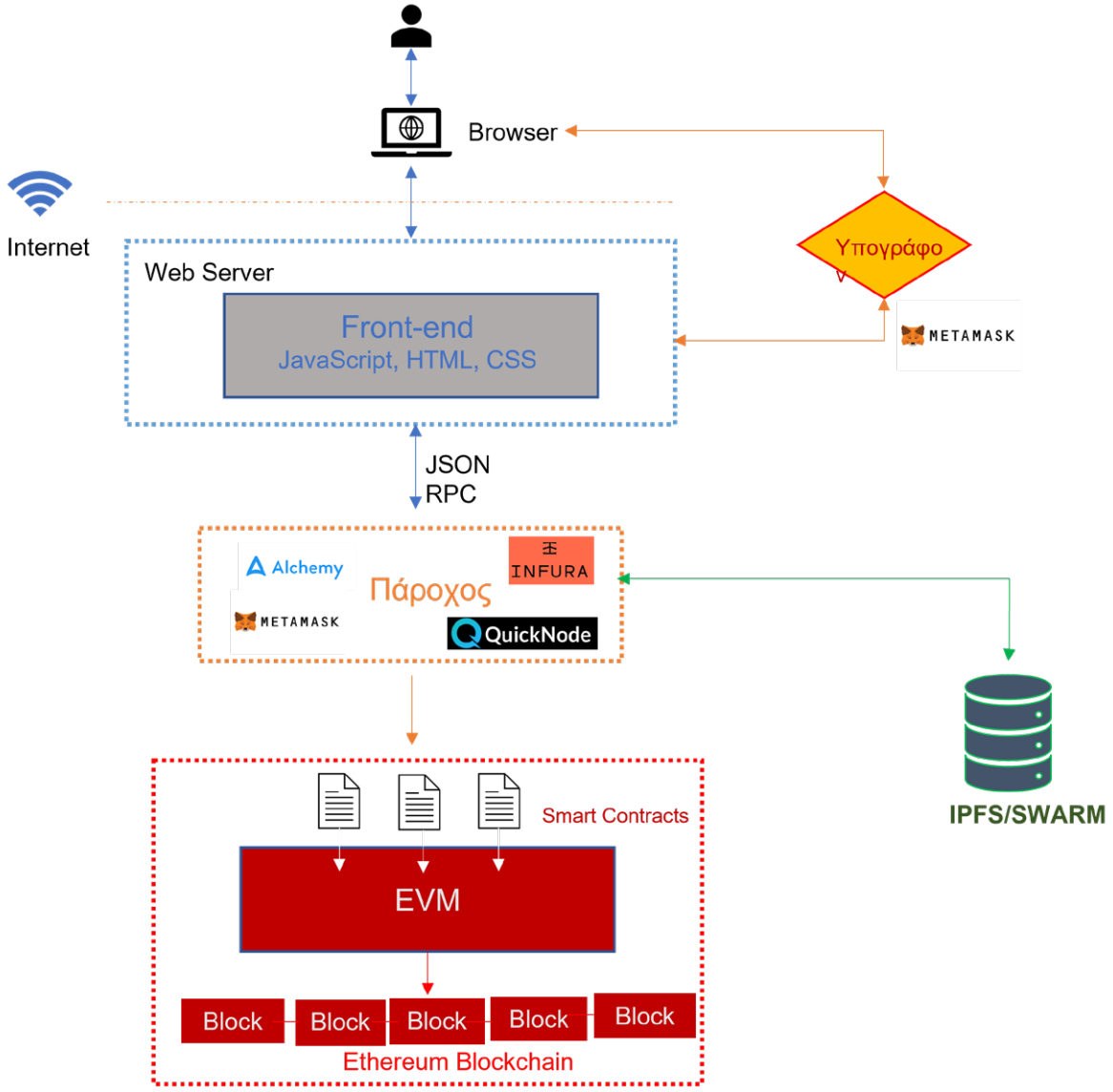
Για την αποθήκευση των δεδομένων εκτός δικτύου χρησιμοποιούνται δύο πολύ διαδεδομένες κατανεμημένες λύσεις: το IPFS[100] και το Swarm[101].

- Το *IPFS* (*InterPlanetary File System*) είναι ένα πρωτόκολλο και ένα δίκτυο κατανεμημένων κόμβων που συνεργάζονται για την αποθήκευση και τον διαμοιρασμό αρχείων μέσα σε ένα κατανεμημένο σύστημα αρχείων. Χρησιμοποιεί διευθυνσιοδότηση βάσει περιεχομένου (content addressing) (Sheldon, 2021) για τον προσδιορισμό με μοναδικό τρόπο ενός αρχείου μέσα στον κατανεμημένο χώρο των μελών/κόμβων του. Ο τρόπος λειτουργίας του επιτρέπει την αντικατάσταση του http(s) στη λειτουργία του Διαδικτύου, διευκολύνοντας την εύρεση αρχείων στο σύστημα αυτό.
  Επιπρόσθετα, υπάρχει ένα συμπληρωματικό πρωτόκολλο, το *Filecoin* (2022), το οποίο και προσφέρει κίνητρα με βάση την κρυπτο-οικονομία στους κόμβους IPFS για να διαμοιραστούν την αποθήκευση των αρχείων. Το Filecoin λειτουργεί συμπληρωματικά με το IPFS. Η χρήση του IPFS είναι αναγκαία από τον κόμβο για την εφαρμογή του Filecoin. Αντιθέτως, η χρήση του IPFS μπορεί να γίνει χωρίς την εφαρμογή του Filecoin.
  Για την υλοποίηση ενός IPFS κόμβου μπορούν να χρησιμοποιηθούν online υπηρεσίες που παρέχονται από εξειδικευμένους παρόχους, όπως η Infura (Εικόνα 8.3) και η Pinata[102].
- Το *Swarm*, σε αντιστοιχία με το IPFS, προσφέρει και αυτό την ανάπτυξη ενός αποκεντρωμένου δικτύου για την αποθήκευση αρχείων σε αυτό. Βασική διαφορά του με το IPFS είναι ότι ο μηχανισμός παροχής κρυπτο-οικονομικών κινήτρων είναι έμφυτος στο πρωτόκολλο και όχι συμπληρωματικός, όπως είναι στο IPFS. Μάλιστα, για να μπορέσει να λειτουργήσει ο μηχανισμός αυτός, γίνεται διαχείρισή του με τη χρήση smart contracts στο δίκτυο του Ethereum.

Η **Εικόνα 8.5** δείχνει την αρχιτεκτονική μιας Web 3.0 εφαρμογής με την προσθήκη της δυνατότητας αποθήκευσης δεδομένων εκτός αλυσίδας (off-chain) με τη βοήθεια των λύσεων κατανεμημένης αποθήκευσης που προαναφέρθηκαν.

Από την Εικόνα 8.5 φαίνεται ότι ο πάροχος του κόμβου του δικτύου blockchain είναι αυτός που συνδέεται με το κατανεμημένο δίκτυο κόμβων που χρησιμοποιείται για την αποθήκευση των δεδομένων off-chain. Ο λόγος σχετίζεται και με το γεγονός ότι, συνήθως, το hash που παράγεται από τα δεδομένα και την αποθήκευσή τους στο IPFS ή το Swarm μπαίνει στον blockchain και στο ledger. Με τον τρόπο αυτόν εξασφαλίζεται ότι, αν και off-chain, τα δεδομένα δεν έχουν αλλοιωθεί. Δηλαδή, η παραγωγή του hash σε αυτά θα πρέπει να ταυτίζεται με την τιμή του που έχει αποθηκευτεί στο ledger. Μόνο τότε είναι δυνατόν να εξασφαλιστεί ότι δεν υπάρχει αλλοίωση στα δεδομένα.

[100] Online Σύνδεσμος: https://ipfs.io/
[101] Online Σύνδεσμος: https://www.ethswarm.org/
[102] Online Σύνδεσμος: https://www.pinata.cloud/



Εικόνα 8.5 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής με την προσθήκη εφαρμογών για κατανεμημένη αποθήκευση των δεδομένων εκτός της αλυσίδας του blockchain (off-chain).*

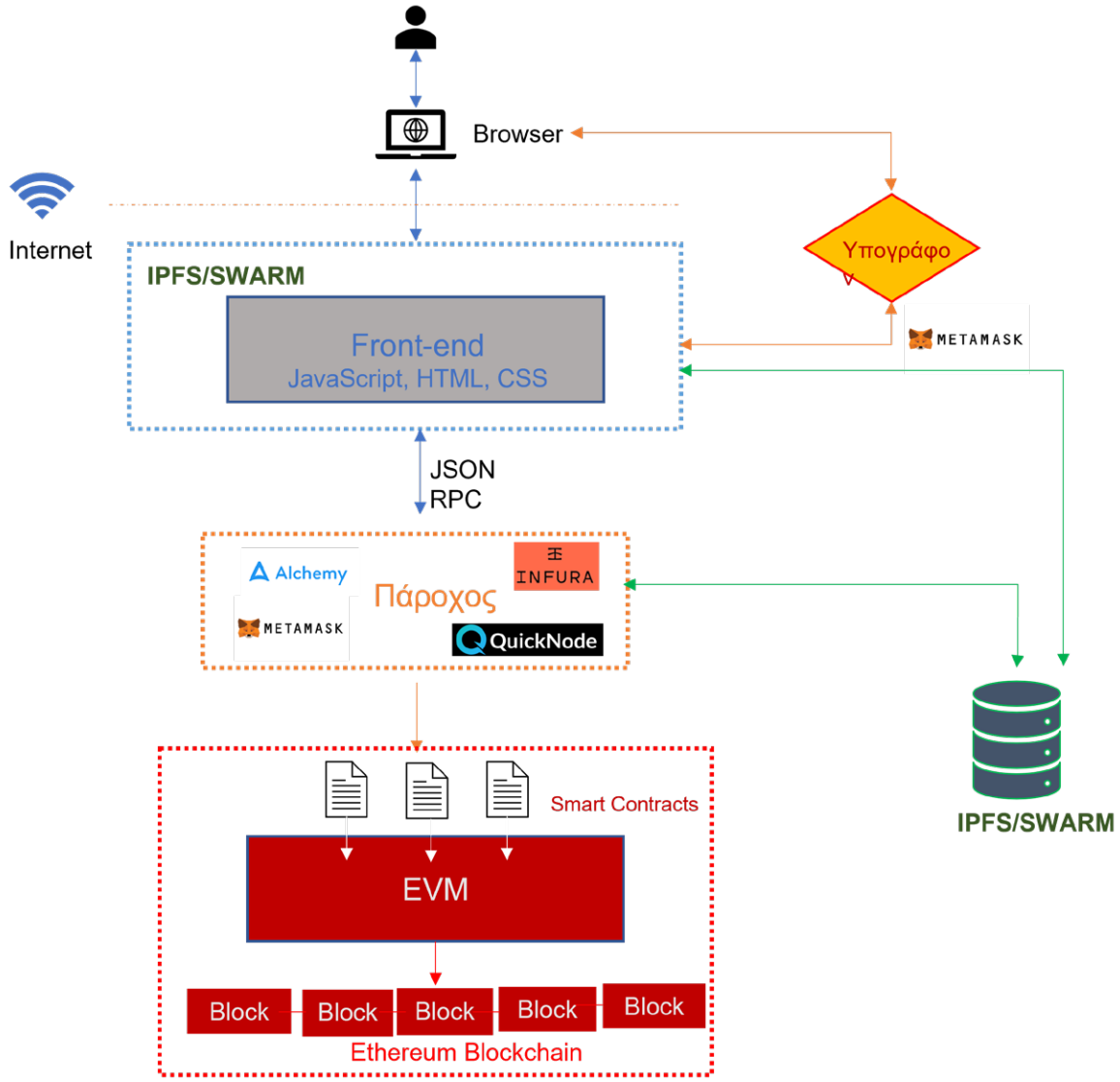
- Αμιγώς κατανεμημένη υλοποίηση:

Συνεχίζοντας τη μελέτη της δομής μιας εφαρμογής Web 3.0, όπως αυτή απεικονίζεται στην Εικόνα 8.5, παρατηρείται ότι το σύστημα δεν είναι πλήρως κατανεμημένο.

Παρ' όλες τις προσπάθειες και τις επιλογές που έχουν γίνει για να παρουσιαστεί μια πλήρως εναρμονισμένη με το Web 3.0 δομή, δεν έχει προβλεφθεί κάτι διαφορετικό για την αρχή της όλης συζήτησης. Δηλαδή δεν έχει προβλεφθεί πώς θα έρθει σε επαφή ο χρήστης με την εφαρμογή, πιο συγκεκριμένα πώς θα αποκτήσει πρόσβαση στο front-end κομμάτι αυτής. Ο γνωστός τρόπος από το Web 2.0 είναι να κατεβάσει και να εγκαταστήσει την εφαρμογή από κάποιον server, κάτι το οποίο και θα μπορούσε να αποτελέσει και μοναδικό σημείο αποτυχίας σε περίπτωση που αυτός ο server βγει εκτός λειτουργίας.

Εναλλακτικά, σε μια στρατηγική πιο πολύ εναρμονισμένη με τη λογική του Web 3.0, θα μπορούσε να προκύψει μια σχεδιαστική αλλαγή η οποία και θα προσφέρει μια καθαρά κατανεμημένη εμπειρία στον χρήστη. Για να είναι εφικτό αυτό, θα πρέπει η εφαρμογή να βρίσκεται αποθηκευμένη στο κατανεμημένο σύστημα διαμοιρασμού των αρχείων που χρησιμοποιείται και για την αποθήκευση των δεδομένων off-chain (π.χ. IPFS ή Swarm).

Στην **Εικόνα 8.6** φαίνεται η διττή αυτή χρήση των πρωτοκόλλων IPFS και Swarm για την αποθήκευση τόσο του κώδικα της εφαρμογής όσο και των δεδομένων off-chain που παράγονται από την εφαρμογή.



Εικόνα 8.6 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής με την προσθήκη ενός κατανεμημένου συστήματος διαμοιρασμού αρχείων για την αποθήκευση του κώδικα για το front-end τμήμα της εφαρμογής.*

- Ανάγνωση δεδομένων:

Η αρχιτεκτονική μιας σύγχρονης Web 3.0 εφαρμογής δεν ολοκληρώνεται όμως εδώ. Εκτός από την προσθήκη δεδομένων στο ledger, με τη βοήθεια του front-end για τη δημιουργία των συναλλαγών και του πορτοφολιού για την υπογραφή τους, είναι αναγκαία και η δυνατότητα για ανάγνωση δεδομένων από αυτό. Μάλιστα, η ανάγνωση είναι μια ενέργεια που δεν απαιτεί πληρωμή για την ολοκλήρωσή της, σε αντίθεση με την προσθήκη δεδομένων στο ledger.

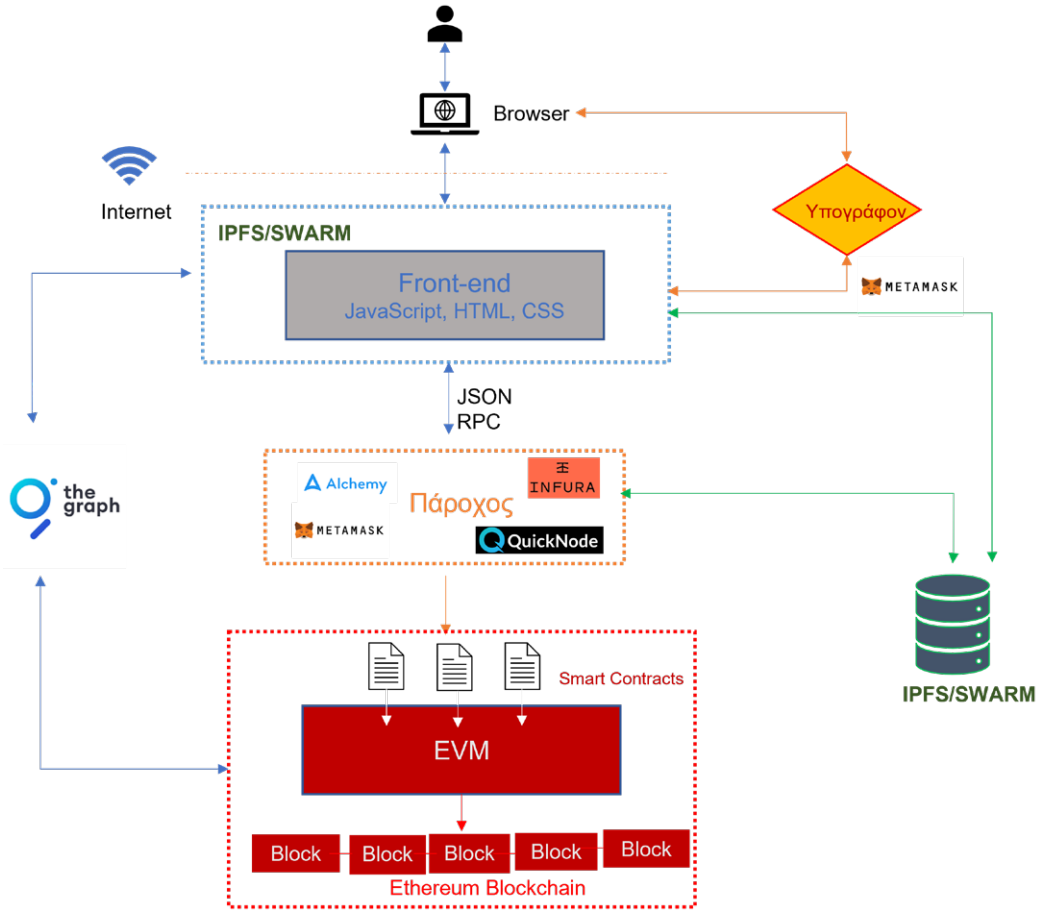
Υπάρχουν δύο δυνατοί τρόποι για την ανάγνωση δεδομένων από ένα smart contract ή από το ledger:

- *Η χρήση των events κατά τη σύνταξη των smart contracts*: Όπως παρουσιάστηκε και στο Κεφάλαιο 7, τα events χρησιμοποιούνται για να ειδοποιήσουν το front-end ότι έχει γίνει μια αλλαγή στο ledger. Για να μπορέσει να καταλάβει πότε υπάρχει event, το front-end πρέπει να φροντίσει να ακούει για συγκεκριμένα events και να έχει προγραμματιστεί ο τρόπος που θα αντιδράσει όταν αντιληφθεί ότι έχει προκύψει ένα.
  Αυτή είναι μια μέθοδος ιδιαίτερα διαδεδομένη στη δημιουργία DApps καθώς και των smart contracts. Παρ' όλα αυτά, δεν παύει να έχει τα δικά της μειονεκτήματα. Έτσι, υπάρχει το πρόβλημα της παράλειψης προσθήκης ενός event στο smart contract είτε λόγω κακού προγραμματισμού είτε γιατί μια ανάγκη αναπτύχθηκε στην πορεία κατά τη χρήση. Το αποτέλεσμα είναι να μην είναι σε θέση να

το αντιληφθεί το front-end και να το διαχειριστεί. Αυτό, αυτόματα, οδηγεί στην ανάγκη συγγραφής και εγκατάστασης νέου smart contract που θα περιλαμβάνει και το νέο event, με ό,τι οικονομικές επιβαρύνσεις μπορεί να φέρει κάτι τέτοιο. Επίσης, με την ανάγκη να παρακολουθεί το front-end πολλά events, για να γνωρίζει πότε και πώς θα αντιδράσει, η διαδικασία αρχίζει και περιπλέκεται σημαντικά. Αυτό, με τη σειρά του, μπορεί να οδηγήσει σε νέα λάθη.

- *Χρήση του πρωτοκόλλου The Graph*[103]: Πρόκειται για ένα πρωτόκολλο δημιουργίας ευρετηρίου το οποίο δύναται να βοηθήσει σημαντικά στην παρακολούθηση και διαχείριση των αλλαγών στο ledger, παρόλο που δρα εκτός αυτού (off-chain). Με το πρωτόκολλο αυτό ο χρήστης μπορεί να υποδείξει ποια smart contracts, ποιες συναρτήσεις και ποια events χρειάζεται να παρακολουθεί, προσθέτοντάς τα στο ευρετήριο του πρωτοκόλλου. Η προσθήκη στο ευρετήριο είναι παρόμοια με τη δημιουργία υπο-γράφων (subgraphs) που βοηθούν στην ευκολότερη αναζήτηση των δεδομένων. Επιπλέον, το The Graph δύναται να προγραμματιστεί για τον τρόπο με τον οποίο θα δημιουργήσει οντότητες, ως αποτέλεσμα του εντοπισμού συγκεκριμένων αλλαγών στο ledger, έτσι ώστε να είναι εύκολο για το front-end να το διαχειριστεί.
  Μεγάλο πλεονέκτημα του πρωτοκόλλου αποτελεί και η γλώσσα αναζήτησης (query language) που χρησιμοποιεί, με την ονομασία GraphQL. Η γλώσσα αυτή έχει αποκτήσει πολλούς υποστηρικτές, καθώς θεωρείται ότι η χρήση της και η δημιουργία ερωτημάτων με αυτήν είναι εύκολες. Οι αναζητήσεις και τα ερωτήματα που γράφονται στη γλώσσα αυτή επιτρέπουν στο πρωτόκολλο να αναζητήσει στο subgraph τα δεδομένα που ενδιαφέρουν την εφαρμογή με εύκολο και γρήγορο τρόπο ή ακόμα και on-chain.

Η νέα μορφή της αρχιτεκτονικής για τη Web 3.0 εφαρμογή με την προσθήκη λύσεων για αναζήτηση δεδομένων στο ledger φαίνεται στην Εικόνα 8.7



Εικόνα 8.7 *Η αρχιτεκτονική μιας Web 3.0 εφαρμογής με την προσθήκη υπηρεσίας (The Graph) για αναζήτηση δεδομένων στο ledger.*

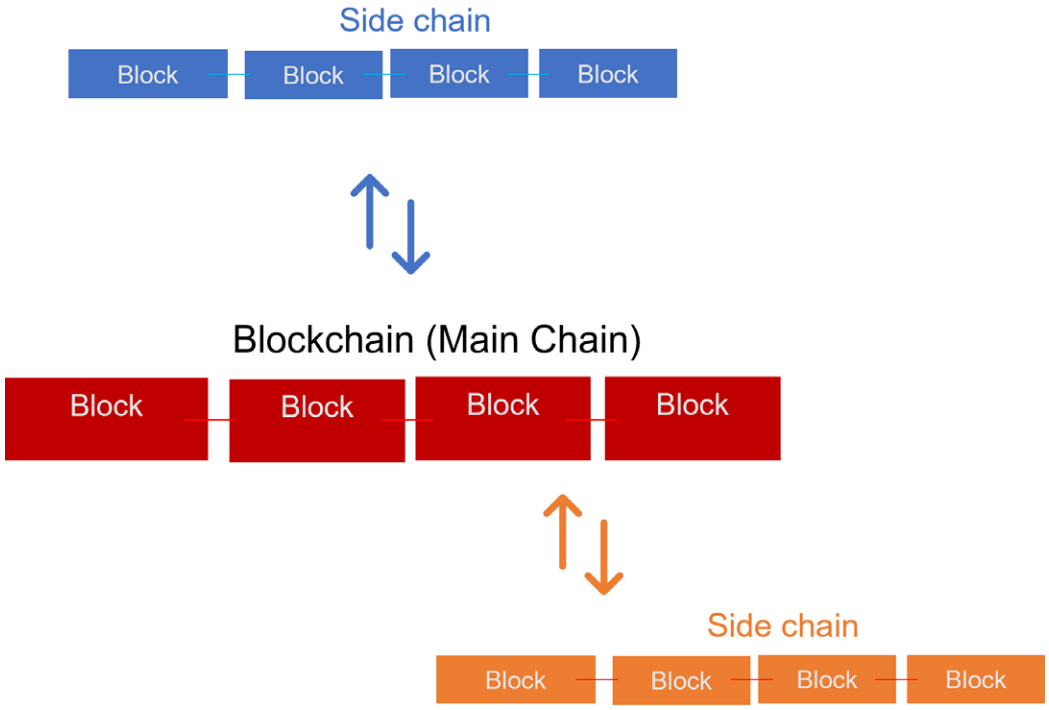
[103] Online Σύνδεσμος: https://thegraph.com/en/

- Επέκταση:

Το επόμενο σημαντικό θέμα σχετίζεται με την επέκταση της εφαρμογής έτσι ώστε να μπορέσει να εξυπηρετήσει περισσότερους χρήστες.

Δυστυχώς, μέχρι πρόσφατα το δίκτυο του Ethereum είχε σημαντικά προβλήματα σε αυτόν τον τομέα, κυρίως λόγω των ιδιαίτερα αυξημένων τιμών στο κόστος (με τη μορφή των fees) για την αγορά gas και την εγγραφή δεδομένων στο ledger. Έτσι, αποθαρρύνθηκαν πολλοί για να δημιουργήσουν νέες εφαρμογές σε αυτό. Αντιθέτως, για να το αντιμετωπίσουν, αρκετοί κατέφυγαν στη χρήση άλλων δικτύων που έχουν καλύτερη επεκτασιμότητα, όπως είναι το Polygon[104]. Το Polygon αποτελεί μια πολύ αποδοτική λύση Επιπέδου 2, η οποία έχει τις δικές της παράπλευρες αλυσίδες (side chains), οι οποίες αναλαμβάνουν να διαχειριστούν τις συναλλαγές του τμήματος του δικτύου που αντιστοιχεί σε αυτές. Ανά περιοδικά διαστήματα, τα αποτελέσματα όλων των συναλλαγών μαζεύονται και στέλνονται ομαδικά ως μια συναλλαγή στο κυρίως δίκτυο για καταγραφή και επιβεβαίωση. Έτσι, μειώνονται οι καθυστερήσεις που προκύπτουν στο κυρίως δίκτυο, καθώς πλέον η χρονοβόρα διαδικασία της δημιουργίας και επαλήθευσης της συναλλαγής έχει περάσει στα side chains και ολοκληρώνεται off-chain. Τα δεδομένα περνούν on-chain στην περιοδική σύνοψη που αναφέρθηκε.

Στην **Εικόνα 8.8** φαίνεται η διάκριση του κυρίως δικτύου από τις παράπλευρες αλυσίδες (side chains) που αναφέρθηκαν.



Εικόνα 8.8 *Η δομή του συστήματος με τη χρήση μιας κύριας αλυσίδας και των παράπλευρων αλυσίδων (side chains) που προσφέρονται για τη βελτίωση της επεκτασιμότητας του δικτύου και των DApps.*

Εκτός από το Polygon, άλλες τεχνικές Επιπέδου 2 που έχουν δοκιμαστεί ως μέσο αύξησης της επεκτασιμότητας του δικτύου είναι και τα λεγόμενα Rollups (Optimistic Rollups και ZKRollups) (Smith 2022). Η καθεμία έχει τα προτερήματα και τα μειονεκτήματά της, αλλά μαζί αποτελούν τις σύγχρονες λύσεις που μελετώνται για την αύξηση της επεκτασιμότητας στις λύσεις blockchain.

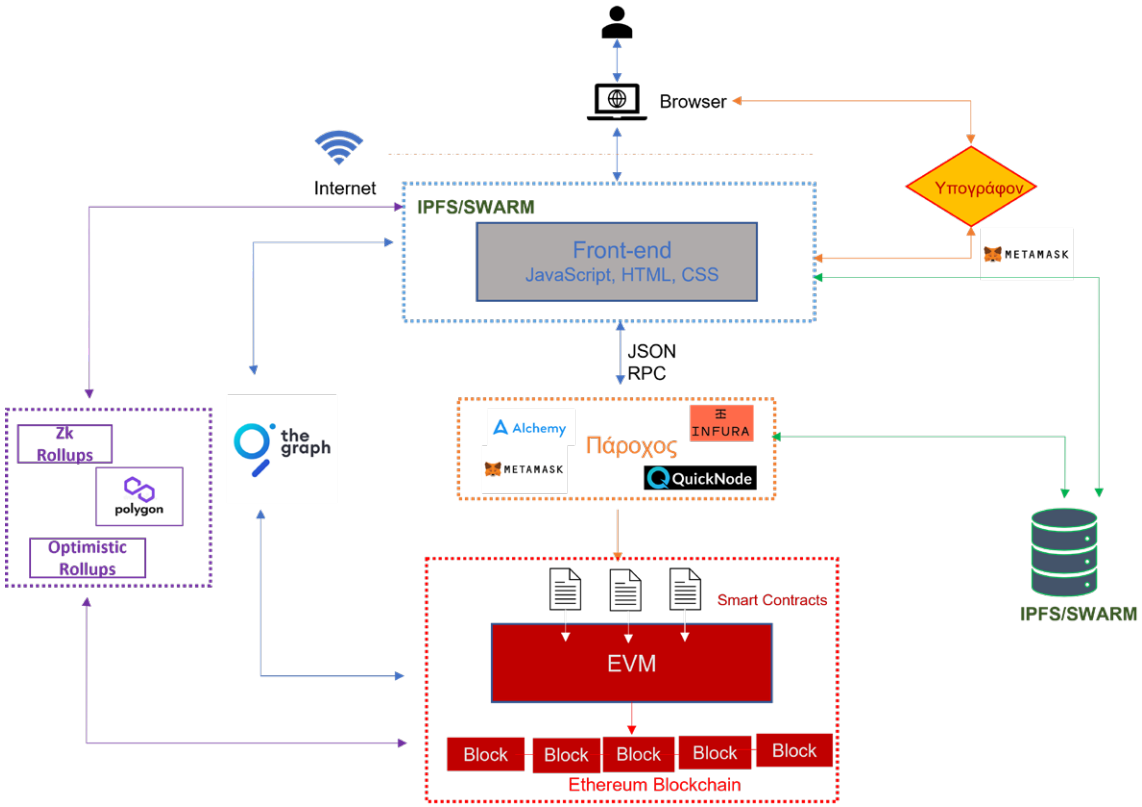
Αυτό άλλωστε φαίνεται και από την επιλογή τους για εφαρμογή στη νέα έκδοση του Ethereum 2.0, η οποία συνδυάζεται και με τη χρήση του PoS ως πρωτοκόλλου συναίνεσης.

Οι λύσεις επέκτασης που συζητούνται αφορούν την επικοινωνία του front-end με το δίκτυο του blockchain (Ethereum στο παράδειγμα εδώ). Έτσι, το σχήμα της δομής μιας Web 3.0 εφαρμογής ανανεώνεται για να συμπεριλάβει και την περίπτωση της επεκτασιμότητας. Η νέα αυτή έκδοση φαίνεται στην **Εικόνα 8.9**.

Ο συνδυασμός όλων αυτών των επιλογών συντελεί στην ανάπτυξη σύγχρονων DApps. Το πρόβλημα που δημιουργείται είναι η ύπαρξη πολλών λύσεων καθώς και ο τρόπος με τον οποίο μπορεί ένας προγραμματιστής να αλληλεπιδράσει μαζί τους.

[104] Online Σύνδεσμος: https://polygon.technology/

Στην επόμενη υποενότητα θα παρουσιαστούν κάποια γνωστά εργαλεία που χρησιμοποιούνται για την ανάπτυξη σύγχρονων Web 3.0 εφαρμογών.



Εικόνα 8.9 *Η αρχιτεκτονική μιας εφαρμογής Web 3.0 με τη χρήση λύσεων επεκτασιμότητας.*

8.1.3 Εργαλεία για την ανάπτυξη DApps

Όπως έχει φανεί με την ανάλυση στο 8.1.1 και στο 8.1.2, η ανάπτυξη ενός σύγχρονου Web 3.0 DApp απαιτεί τον συνδυασμό διάφορων τεχνολογιών. Συνήθως, ένας προγραμματιστής μπορεί να επιλέξει να χρησιμοποιήσει ειδικά προγράμματα τα οποία μπορούν να του προσφέρουν απαραίτητα εργαλεία ή να επιλέξει ειδικά πλαίσια προγραμματισμού (frameworks) τα οποία μπορούν να του προσφέρουν τις απαραίτητες λειτουργίες με πολύ απλό τρόπο.

Ενδεικτικά, τα frameworks αυτά προσφέρουν τη δυνατότητα στον προγραμματιστή για:

- Να δημιουργήσει το δικό του τοπικό blockchain για τις δοκιμές τους.
- Να μεταγλωττίσει και να τεστάρει τα προγράμματά τους (π.χ. smart contracts).
- Να συνδεθεί με το δίκτυο blockchain (Ethereum) για την εγκατάσταση smart contracts είτε πρόκειται για τοπικό δίκτυο είτε για το δημόσιο δίκτυο του Ethereum.
- Να χρησιμοποιήσει κατανεμημένη αποθήκευση αρχείων (π.χ. IPFS).

Αρκετές από τις λύσεις που θα παρουσιαστούν στη συνέχεια έχουν ήδη αναφερθεί κατά την ανάπτυξη της δομής μιας εφαρμογής Web 3.0. Παρ' όλα αυτά θα αναφερθούν τα ονόματά τους και εδώ για πληρότητα στην κάλυψη του θέματος.

Έτσι, λοιπόν, γνωστά frameworks που χρησιμοποιούνται για την ανάπτυξη DApps είναι τα *Pranesh 2022* (Anwar, 2021):

- *Truffle*[105]: Πρόκειται για ένα από τα πλέον διαδεδομένα περιβάλλοντα ανάπτυξης DApp για το Ethereum. Επιτρέπει τη δημιουργία projects, τη συγγραφή και τη μεταγλώττιση smart contracts, καθώς και την εκτέλεση δοκιμαστικών και την αλληλεπίδραση με το contract. Επιπλέον, επιτρέπει την ανάπτυξη ενός τοπικού δικτύου blockchain (που δεν επικοινωνεί με κάποιο πραγματικό) καθαρά για δοκιμές.

[105] Online Σύνδεσμος: https://trufflesuite.com/

Τέλος, είναι σημαντικό ότι μπορεί να τρέξει scripts για τις δοκιμές σε δύο γλώσσες (JavaScript και Solidity) και ότι επιτρέπει τη διαχείριση των smart contracts καθώς και την όποια αναβάθμισή του. Αποτελεί μια ολοκληρωμένη σουίτα από εργαλεία που διευκολύνουν τους προγραμματιστές.

- *Hardhat*[106]: Είναι ένα σύγχρονο περιβάλλον ανάπτυξης λύσεων για το Ethereum. Περιλαμβάνει στοιχεία που επιτρέπουν τη διαμόρφωση, τη σύνταξη, τη μεταγλώττιση και την εγκατάσταση smart contracts και DApps. Για να το πετύχει αυτό, η Hardhat προσφέρει τη δυνατότητα ανάπτυξης ενός τοπικού κόμβου του Ethereum (με το όνομα Hardhat Network) το οποίο μπορεί να χρησιμοποιηθεί από τον προγραμματιστή για να δοκιμάσει τις δικές του λύσεις σε ένα προστατευμένο περιβάλλον.
- *Metamask*: Το δημοφιλές πορτοφόλι που χρησιμοποιείται για τη σύνδεση των χρηστών με το δίκτυο του Ethereum (ή με τα δοκιμαστικά του). Εναλλακτικά, επιτρέπει και την εισαγωγή ρυθμίσεων για άλλες συνδέσεις σε άλλα δίκτυα.
- *Remix*: Εργαλείο για τη σύνταξη και την εγκατάσταση εφαρμογών όπου μπορούν να δοκιμαστούν εύκολα για την αποτελεσματικότητά τους. Στα θετικά του είναι και η ανάπτυξη ποικίλων πρόσθετων που μπορούν να εγκατασταθούν για να επιτρέψουν ακόμα περισσότερες λειτουργίες στους προγραμματιστές.
- *Ganache*: Προσφέρει ένα ασφαλές περιβάλλον για το Ethereum, αφού επιτρέπει στους προγραμματιστές να δοκιμάσουν την απόδοση των DApps προτού τα εγκαταστήσουν στο πραγματικό δίκτυο. Σε συνδυασμό με το Truffle προσφέρει ένα ολοκληρωμένο περιβάλλον για ανάπτυξη και δοκιμή των smart contracts.
- *OpenZeppelin SDK*: Πρόκειται για μια από τις μεγαλύτερες σουίτες με κώδικα για τη συγγραφή smart contracts η οποία χρησιμοποιείται ευρέως από πολλούς προγραμματιστές που ασχολούνται με το δίκτυο του Ethereum. Προσφέρει πολλές επιλογές στους προγραμματιστές και διευκολύνει πολύ την αλληλεπίδραση με τα smart contracts ανεξαρτήτως του επιπέδου γνώσης του χρήστη.
- *The Graph*: Όπως παρουσιάστηκε και προηγουμένως, αποτελεί μια ιδιαίτερα αποδοτική και σύγχρονη λύση για την αναζήτηση και εύρεση δεδομένων στο ledger. Συνοδεύεται με μια ισχυρή και εύκολη στη χρήση γλώσσα αναζήτησης και έχει διαδοθεί ανάμεσα στους προγραμματιστές για την πολύ καλή απόδοση που έχει σε μεγάλα δίκτυα blockchain που, αλλιώς, θα παρουσίαζαν σημαντικές καθυστερήσεις στην ολοκλήρωση των αναζητήσεών τους.
- *Scaffold.eth*[107]: Πρόκειται για μια σύγχρονη σουίτα εφαρμογών που περιέχει και κάποιες από τις λύσεις που αναφέρονται εδώ (π.χ. το Hardhat). Έχει σκοπό να προσφέρει ένα ολοκληρωμένο περιβάλλον το οποίο θα χρησιμοποιηθεί και για εκπαιδευτικούς σκοπούς, εισάγοντας τους νέους χρήστες σταδιακά στην εκμάθηση συγγραφής smart contracts και στη δοκιμή τους. Ταυτόχρονα, παρέχοντας κάποια παραδείγματα smart contract, τα συμπληρώνει και με εφαρμογές με χρήση της βιβλιοθήκης React για να μπορέσει να δημιουργήσει απλά Dapps και να επιτρέψει στον χρήστη να κάνει απλές αλλαγές και να τις δει να εκτελούνται. Για να ολοκληρωθεί όμως αυτό, χρειάζεται η δυνατότητα ανάπτυξης ενός (τοπικού) δικτύου, το οποίο και παρέχεται με τη βοήθεια της Hardhat.
- *Alchemy*[108]: Πρόκειται για μια πλήρη σουίτα ανάπτυξης Web 3.0 εφαρμογών με συνδέσεις σε πολλά δίκτυα blockchain, διευκολύνοντας τον προγραμματιστή. Παρέχει μια μεγάλη ποικιλία από APIs, τα οποία χωρίζονται σε κατηγορίες και μπορεί να τα χρησιμοποιήσει ο χρήστης στην εφαρμογή του.

8.1.4 Αναζήτηση υπαρχόντων DApps

Με τη δημοφιλία των Web Apps ολοένα και να αυξάνεται έχουν δημιουργηθεί σελίδες τις οποίες μπορεί να επισκεφθεί κανείς και να αναζητήσει κάποιο. Μάλιστα, οι ιστοσελίδες αυτές περιέχουν DApps τα οποία και είναι συμβατά όχι μόνο με το blockchain του Ethereum αλλά και με άλλες πλατφόρμες, όπως είναι οι EOS, Polygon, Solana κ.ά.

---

[106] Online Σύνδεσμος: https://hardhat.org/

[107] Online Σύνδεσμος: https://github.com/scaffold-eth/scaffold-eth

[108] Online Σύνδεσμος: https://www.alchemy.com/

Επιπλέον, συχνά στις σελίδες αυτές τα DApps χωρίζονται ανάλογα με το είδος τους και το περιεχόμενο. Ανάμεσα στις κατηγορίες αυτές βρίσκονται τα DApps με θέμα την Αποκεντρωμένη Οικονομία (DeFi), τα NFTs, καθώς και παιχνίδια που βασίζονται στο blockchain.

Ενδεικτικά, ορισμένες ιστοσελίδες που έχουν πληροφορίες για DApps είναι οι:

- State of DApps (https://www.stateofthedapps.com/): Από τις πιο γνωστές πλατφόρμες με πληροφορίες για DApps. Εκεί θα βρείτε πληροφορίες για πάνω από 3.000 εφαρμογές οι οποίες και τρέχουν σε περισσότερα από 7 πλατφόρμες. Παρέχει τη δυνατότητα να προσθέσετε και το δικό σας DApp.
- Dapp.com (https://www.dapp.com/): Μια επίσης δημοφιλής πλατφόρμα η οποία φιλοξενεί DApps και επιτρέπει και την υποβολή των δικών σας. Καλύπτει, και αυτή, έναν σημαντικό αριθμό από πλατφόρμες και μια μεγάλη ποικιλία κατηγοριών.
- DappRadar (https://dappradar.com/): Μια ακόμα ιστοσελίδα που φιλοξενεί DApps, με έμφαση σε αυτά που βασίζονται σε NFTs και σε DeFis.
- Στις περισσότερες από αυτές τις ιστοσελίδες ο χρήστης μπορεί να αναζητήσει και να βρει τα DApps που έχουν τη μεγαλύτερη βαθμολογία σύμφωνα με την κριτική των χρηστών. Ενδεικτικά αναφέρονται ορισμένα DApps, όπως είναι τα: Cryptokitties, Decentraland[109], Upland, τα οποία και έχουν απασχολήσει σημαντικό αριθμό ενδιαφερομένων.

[109] Online Σύνδεσμος: https://decentraland.org/?utm_source=StateOfTheDApps

Βιβλιογραφία

Anwar, H. (2021) *10 Best Ethereum Development Tools*. May 2021. Online πηγή: https://101blockchains.com/best-ethereum-development-tools/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Cointelegraph (2022). *What is Web 3.0: A beginner's guide to decentralized internet of the future*. Online πηγή: https://cointelegraph.com/blockchain-for-beginners/what-is-web-3-0-a-beginners-guide-to-the-decentralized-internet-of-the-future [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Filecoin (2022). Online πηγή: https://filecoin.io/ [Τελευταία Πρόσβαση: Δεκέμβριος 2022].

JSON-RPC (2013). *JSON-RPC 2.0 Specification*. JSON-RPC Working Group. Online πηγή: https://www.jsonrpc.org/specification [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Kasireddy, P. (2021). *The Architecture of a Web 3.0 Application*. Online πηγή: https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ethereum.org (2022). *Ethereum: Dapp Development Frameworks*. July 2022. Online πηγή: https://ethereum.org/en/developers/docs/frameworks/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Smith (2022). *Ethereum: Scaling*. July 2022. Online πηγή: https://ethereum.org/en/developers/docs/scaling/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Sheldon, R. (2021). *Content-Addressed storage (CAS)*. December 2021. Online πηγή: https://www.techtarget.com/searchstorage/definition/content-addressed-storage [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Vermaak, W. (2021). *What is Web 3.0?*. Online πηγή: https://coinmarketcap.com/alexandria/article/what-is-web-3-0 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 9
Τομείς Χρήσης και Σχεδιασμός

Σύνοψη

Το κεφάλαιο αυτό παρουσιάζει με παραδείγματα τους σημαντικότερους τομείς της ζωής του ανθρώπου οι οποίοι αναμένεται να επηρεαστούν σημαντικά (ή έχουν ήδη αρχίσει να επηρεάζονται) από την εφαρμογή λύσεων που βασίζονται στην τεχνολογία του blockchain. Επιπλέον, αναφέρονται λύσεις που εφαρμόζονται ή που σχεδιάζονται να εφαρμοστούν στους τομείς αυτούς και ακολουθεί μια τεχνική περιγραφή τους.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2 και 8.

9.1 Τομείς χρήσης της τεχνολογίας blockchain

Η τεχνολογία του blockchain έχει επιτύχει μέσα σε σύντομο διάστημα να αποδείξει τα πλεονεκτήματά της, κυρίως με το παράδειγμα των κρυπτονομισμάτων. Ταυτόχρονα, όμως, έχει αποδείξει με τη χρήση των smart contracts ότι ο τρόπος χρήσης της δεν είναι αναγκαίο να περιοριστεί στην ανάπτυξη και διαχείριση των κρυπτονομισμάτων, αλλά μπορεί με όχημα αυτά να γενικευτεί σε μια ευρύτερη γκάμα από σενάρια χρήσης. Έτσι, λοιπόν, στον Szabo (2016) παρουσιάζονται περιπτώσεις χρήσης που αξιοποιούν τα χαρακτηριστικά της τεχνολογίας blockchain για τη βελτίωση της απόδοσης των εφαρμογών τους.

Από τα πιο σημαντικά χαρακτηριστικά που παίζουν σημαντικό ρόλο στη βελτίωση της απόδοσης είναι η συγγραφή smart contracts (βλ. Κεφάλαιο 6). Με τα smart contracts έγινε δυνατή η εισαγωγή μιας λογικής η οποία και επιτρέπει την αυτόματη εκτέλεση ενεργειών (με τη μορφή συναλλαγών) εφόσον ικανοποιούνται ορισμένες προδιαγραφόμενες συνθήκες. Προφανώς, το περιεχόμενο των smart contracts μεταβάλλεται ανάλογα με την περίπτωση και προσαρμόζεται αντίστοιχα. Επιπλέον, η δυνατότητα συνδυασμού των smart contracts με την τεχνολογία του blockchain προσδίδει μεγαλύτερη αξιοπιστία στο νέο σύστημα λόγω των ιδιαίτερων χαρακτηριστικών που φέρνει αυτή (βλ. Κεφάλαιο 1).

Οι περιπτώσεις χρήσης που μελετώνται συμπεριλαμβάνουν, μεταξύ άλλων:

- τη δημιουργία ψηφιακών ταυτοτήτων,
- τη διαχείριση της εφοδιαστικής αλυσίδας,
- την οικονομία και το εμπόριο,
- την υγεία, και
- τον δημόσιο τομέα και υπηρεσίες αυτού.

Είναι σημαντικό να αναφερθεί ότι τα επιμέρους χαρακτηριστικά μιας λύσης μπορεί να διαφέρουν, καθώς επίσης και ο κώδικας των smart contracts. Ορισμένες λύσεις μπορεί να βασίζονται σε permissioned δίκτυα blockchain, ενώ άλλες σε permissionless. Επιπλέον, μπορεί μια λύση να χρειάζεται την υποστήριξη από ένα δημόσιο δίκτυο, ενώ μια άλλη από ιδιωτικό ή υβριδικό. Τέλος, η επιλογή του αλγόριθμου συναίνεσης μπορεί να διαφέρει σε κάθε εφαρμογή. Αυτό συμβαίνει γιατί δεν υπάρχει μία λύση που καλύπτει όλες τις περιπτώσεις. Αντιθέτως, η κάθε περίπτωση είναι μοναδική και έχει τα δικά της χαρακτηριστικά και τις δικές της απαιτήσεις.

Στη συνέχεια αναλύονται οι τρόποι με τους οποίους επηρεάζεται ο κάθε τομέας από λύσεις blockchain, δίνονται παραδείγματα τέτοιων λύσεων και ακολουθεί μια τεχνική ανάλυση των χαρακτηριστικών των λύσεων αυτών (Szabo, 2016· Bailey, 2022· Geroni, 2021· Daley, 2022).

9.2 Παραδείγματα λύσεων ανά τομέα χρήσης

Στην ενότητα αυτή γίνεται η παρουσίαση των τομέων χρήσης καθώς και των εφαρμογών που ξεχωρίζουν σε καθέναν από αυτούς, μαζί με τις προκλήσεις που καλούνται να αντιμετωπίσουν.

9.2.1 Η περίπτωση των ψηφιακών ταυτοτήτων

Η ύπαρξη και η χρήση μιας ψηφιακής ταυτότητας, η οποία θα παίζει τον ρόλο που έχει η φυσική ταυτότητα στον φυσικό κόσμο, είναι ένα συνεχές ζητούμενο. Παρά την πρόοδο της τεχνολογίας, δεν έχει γίνει δυνατή η πλήρης αντικατάσταση της φυσικής ταυτότητας στον ψηφιακό κόσμο. Αυτό έχει ως αποτέλεσμα να υπάρχει η ανάγκη δημιουργίας νέου λογαριασμού σε κάθε νέα ψηφιακή υπηρεσία που χρειάζεται ο χρήστης, είτε αυτή είναι η δημιουργία ενός λογαριασμού email είτε η είσοδος στο web banking της (κάθε) τράπεζάς του είτε ο λογαριασμός σε ένα μέσο κοινωνικής δικτύωσης.

Όλες αυτές οι υπηρεσίες απαιτούν από τον χρήστη τα στοιχεία του και τη δημιουργία ξεχωριστού λογαριασμού σε αυτές, με κίνδυνο να χάσει ή να ξεχάσει τον κωδικό εισόδου. Επιπλέον, κατά τη διαδικασία αυτή είναι πιθανόν να επαναληφθούν βήματα που έγιναν και σε άλλες υπηρεσίες, με αποτέλεσμα να υπάρχει μια αίσθηση επανάληψης στους χρήστες. Ακόμα όμως και με την ολοκλήρωση της διαδικασίας, κανείς δεν μπορεί να είναι σίγουρος ότι ο κάτοχος του λογαριασμού έχει χρησιμοποιήσει πραγματικά στοιχεία για τη δημιουργία του. Επιπλέον, τα στοιχεία του χρήστη ανήκουν πλέον στον πάροχο, ο οποίος είναι υπεύθυνος για τη φύλαξη και διαχείρισή τους.

Σε αντίθεση με τα παραπάνω, υπάρχει η ανάγκη να δημιουργηθεί μια ψηφιακή ταυτότητα για κάθε χρήστη, η οποία θα έχει τα εξής χαρακτηριστικά:

- *Θα είναι ασφαλής, με σεβασμό στην ιδιωτικότητα και στην εμπιστευτικότητα του χρήστη*: Τα δεδομένα του δεν θα είναι διαθέσιμα σε άλλους, εκτός αν επιλέξει ο ίδιος ο χρήστης να διαμοιραστεί κάποια από αυτά με ένα άλλο συναλλασσόμενο μέρος και για όσο διάστημα θα επιλέξει.
- *Θα είναι μινιμαλιστική*: Θα παρουσιάζει στο συναλλασσόμενο μέρος κάθε φορά μόνο όσα δεδομένα επιλέξει ο χρήστης και είναι απαραίτητα για την ολοκλήρωση της συναλλαγής. Έτσι, για παράδειγμα, για την απόδειξη της ηλικίας κάποιου (για την πρόσβαση σε ένα μέρος) θα μπορεί να δείχνει μόνο την ηλικία του (ή ακόμα και απλώς μια επιβεβαίωση ότι είναι ενήλικος) και όχι παρελκόμενες πληροφορίες που δεν είναι αναγκαίες για την ολοκλήρωση της τρέχουσας συναλλαγής.
- *Θα είναι ανθεκτική*: Δεν θα μπορεί κάποιος να τη λογοκρίνει ή να τη σβήσει χωρίς την έγκριση του χρήστη ή/και της αρμόδιας αρχής. Έτσι, δεν θα μπορεί κάποιος να τη χρησιμοποιήσει αντί για τον πραγματικό της κάτοχο.
- *Θα είναι σταθερή*: Δεν πρέπει να μπορεί να αφαιρεθεί από τον χρήστη χωρίς την έγκρισή του.
- *Έλεγχος στον ιδιοκτήτη*: Μόνο ο ιδιοκτήτης θα μπορεί να αποφασίζει για τον τρόπο και το μέρος που θα χρησιμοποιηθεί η ταυτότητα, μαζί με τις πληροφορίες που θα εμφανιστούν σε κάθε περίπτωση.
- *Θα είναι φορητή*: Η μεταφορά της θα είναι εύκολη και άμεση. Θα μπορούσε να χρησιμοποιηθεί το κινητό τηλέφωνο του χρήστη χωρίς να υπάρχει ανάγκη για το οποιοδήποτε έγγραφο.

Με τη βοήθεια ειδικών smart contracts και την υποστήριξη του blockchain είναι δυνατόν να αναπτυχθούν οι μηχανισμοί εκείνοι που θα επιτρέψουν τη δημιουργία κατανεμημένων ψηφιακών ταυτοτήτων που εξαρτώνται από τον χρήστη αποκλειστικά[110], μαζί με τον έλεγχο για το περιεχόμενο και τη χρήση αυτών ανά περίπτωση. Με τον τρόπο αυτόν και τα δύο μέρη μπορούν να είναι σίγουρα για τη διαφάνεια και ορθότητα των συναλλαγών που λαμβάνουν χώρα. Ο χρήστης γιατί θα έχει στην κατοχή του όλα τα στοιχεία του, που θα του έχουν επιβεβαιωθεί από εγκεκριμένες οντότητες του δικτύου. Και, επιπλέον, θα επιλέγει εκείνος τι και πότε θα φανερώνεται στο μέρος με το οποίο συνδιαλέγεται. Από την άλλη, οι συνδιαλεγόμενοι θα μπορούν να απευθύνονται στο δίκτυο για την επαλήθευση των στοιχείων του χρήστη.

Για να γίνει όμως αυτό πραγματικότητα, χρειάζεται να αναπτυχθούν και smart contract τα οποία θα μπορούν να χρησιμοποιηθούν για την έκδοση πιστοποιητικών αλλά και την επιβεβαίωσή τους. Θα χρειαστεί να οριστούν έμπιστες μονάδες στο δίκτυο που θα αναλάβουν να εκπροσωπήσουν τον μηχανισμό έκδοσης ενός ψηφιακού

[110] Ο όρος που συναντάται συχνά είναι Self-Sovereign Identities (SSIs).

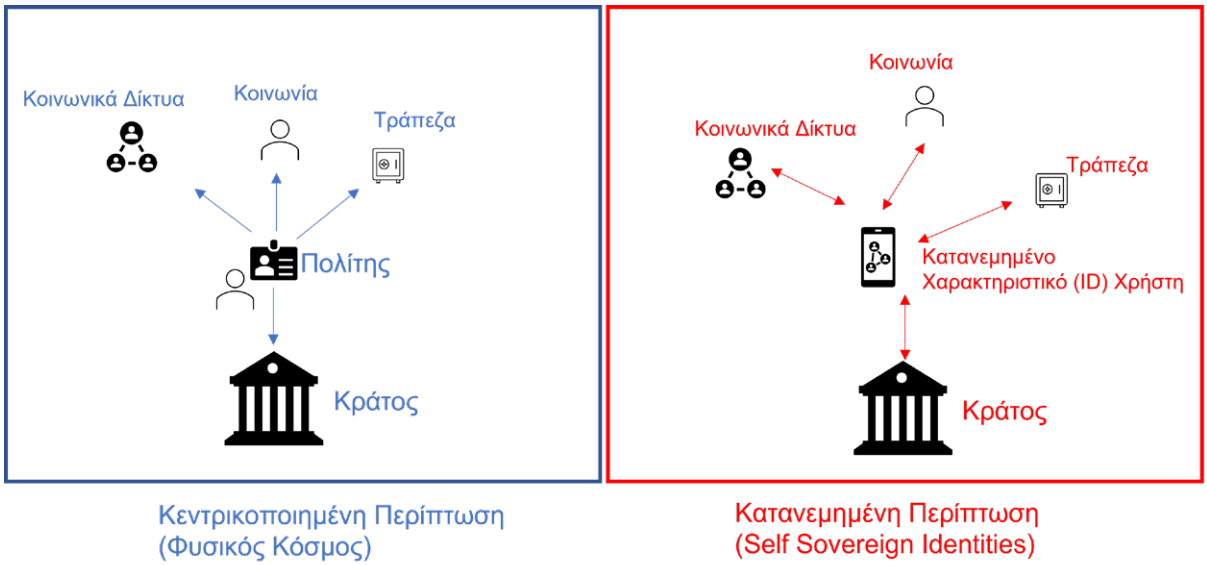
πιστοποιητικού, το οποίο και θα μπορεί, στη συνέχεια, να προστεθεί στην ταυτότητα του χρήστη και να είναι διαχειρίσιμο από αυτόν.

Από τα παραπάνω αναδεικνύονται τόσο τα μειονεκτήματα της τρέχουσας κεντρικοποιημένης περίπτωσης όσο και τα πλεονεκτήματα που μπορεί να υπάρξουν από τη χρήση μιας λύσης κατανεμημένων ταυτοτήτων.

Ο **Πίνακας 9.1** συνοψίζει τα αποτελέσματα αυτά για την περίπτωση της ψηφιακής ταυτότητας με τη βοήθεια μιας κατανεμημένης λύσης που βασίζεται στο blockchain, ενώ η **Εικόνα 9.1** απεικονίζει τους τρόπους λειτουργίας στο παρόν (κεντρικοποιημένος) και στο μέλλον με τη βοήθεια του blockchain (κατανεμημένος).

Προβλήματα στην κεντρικοποιημένη περίπτωση	Πλεονεκτήματα της κατανεμημένης περίπτωσης
Ακριβή και χρονοβόρα διαδικασία επιβεβαίωσης των στοιχείων των χρηστών.	Οι ιδιώτες ελέγχουν τα δεδομένα τους και επιτρέπουν την απόκρυψη όσων χρειάζονται και όπου χρειάζονται.
Περιορισμένος έλεγχος στις πιθανές διαρροές δεδομένων από τις έμπιστες οντότητες λόγω επιθέσεων ή δολιοφθοράς.	Οι υπηρεσίες δεν πρέπει να διατηρούν δεδομένα, αλλά θα μπορούν να τα επιβεβαιώνουν μειώνοντας την ευθύνη αυτή.
Μεγάλη ευθύνη στη διαφύλαξη των δεδομένων των χρηστών που δημιουργεί μοναδικά σημεία αποτυχίας, που αποτελούν στόχους των κακόβουλων χρηστών.	Αυξημένη συμβατότητα, αξιοπιστία και διαλειτουργικότητα.

Πίνακας 9.1 *Μειονεκτήματα και πλεονεκτήματα της κεντρικοποιημένης περίπτωσης σε σύγκριση με την κατανεμημένη.*



Εικόνα 9.1 *Παράδειγμα κεντρικοποιημένης χρήσης για την ταυτοποίηση (αριστερά) και κατανεμημένης χρήσης (δεξιά).*

Στην Εικόνα 9.1 φαίνεται πως στον φυσικό κόσμο ο πολίτης θα πρέπει να έχει αντίγραφο των εγγράφων και αυτά να προσκομίζει στις συναλλαγές του με τρίτους. Στην κατανεμημένη λειτουργία και στη χρήση των Self-Sovereign Identities ο χρήστης χρησιμοποιεί το κινητό του και το δίκτυο blockchain στις συναλλαγές του, και σε αυτό ανατρέχουν και οι συνδιαλεγόμενοι για την επαλήθευση των στοιχείων.

Για να μπορέσει να γίνει εφικτή μια μετάβαση στην κατανεμημένη λειτουργία και στη χρήση των Self-Sovereign Identities πρέπει να γίνουν ορισμένα βήματα για να προετοιμάσουν το έδαφος. Στα βήματα αυτά περιλαμβάνονται:

- Η ευρύτερη αποδοχή στην έκδοση και χρήση ψηφιακών βεβαιώσεων μαζί με την ανάπτυξη ενός κατάλληλου νομοθετικού πλαισίου για τη δημιουργία εμπιστοσύνης σε αυτά.
- Η αύξηση της ασφάλειας στην ανάπτυξη των smart contracts με σκοπό να τονιστεί η εμπιστοσύνη στο δίκτυο.
- Η τεχνική σύνδεση με τους παρόχους βεβαιώσεων.
- Ο κατάλληλος σχεδιασμός και η υλοποίηση πρωτοκόλλων και προτύπων που θα αυξήσει τη διαλειτουργικότητα μεταξύ των εμπλεκόμενων μερών αλλά και μεταξύ των δικτύων που θα δημιουργηθούν.

Στην πράξη έχουν αρχίσει και γίνονται προσπάθειες για την εύρεση των κατάλληλων προτύπων και πρωτοκόλλων για την παράδοση ενός ολοκληρωμένου συστήματος το οποίο μπορεί να χρησιμοποιηθεί για την απόδοση *Κατανεμημένων Αναγνωριστικών* (*Distributed Identities, DIDs*). Έτσι, το W3C Forum παρέδωσε τον Ιούλιο του 2022 την πρώτη ολοκληρωμένη έκδοση για το πρότυπο δημιουργίας κατανεμημένων αναγνωριστικών (DIDs) (W3C, 2022), εξέλιξη πολύ σημαντική για τη δημιουργία εφαρμογών που θα υιοθετήσουν το πρότυπο αυτό.

Ταυτόχρονα, το πρόγραμμα Hyperledger, με την υποστήριξη του Linux Foundation και πολλών γνωστών εταιρειών όπως η IBM, έχει δημιουργήσει μια υλοποίηση, με το όνομα Indy, η οποία είναι συμβατή με το πρότυπο της W3C. Το Hyperledger Indy προσφέρει ένα περιβάλλον ανάπτυξης εφαρμογών που βασίζονται στα DIDs με σκοπό να βοηθήσει στην ευρύτερη αποδοχή τους. Επιπλέον, το uPort[111] αποτελούσε ένα DApp που δούλευε στο Ethereum και που υλοποιούσε την έννοια των ψηφιακών ταυτοτήτων. Τελευταία έχει χωριστεί σε δύο νέα projects που ονομάζονται Serto και Veramo και έχουν στόχο να συνεχίσουν να προσφέρουν λύσεις σχετικές με τα DIDs. Τέλος, ακόμα και η Microsoft έχει δημιουργήσει την εφαρμογή Entra[112] για τη δημιουργία κατανεμημένων αναγνωριστικών με την υποστήριξη του Azur Cloud.

Μιας και πρόκειται για μια υπηρεσία που θα προσφέρεται σε όλους, αναμένεται η λύση που θα επικρατήσει να περιλαμβάνει ένα δημόσιο ανοικτό δίκτυο blockchain στο οποίο όλοι θα μπορούν να γράφουν. Οι επιβεβαιώσεις αυτών θα ξεδιαλύνουν τι είναι πραγματικότητα και τι όχι.

9.2.2 Η διαχείριση της εφοδιαστικής αλυσίδας

Ο τομέας της διαχείρισης των προϊόντων σε μια Εφοδιαστική Αλυσίδα (ΕΑ) αποτελεί, επίσης, έναν τομέα που ενισχύεται σημαντικά από την ανάπτυξη και χρήση λύσεων που βασίζονται στο blockchain. Ο τομέας αυτός παραδοσιακά περιλαμβάνει όλα εκείνα τα βήματα που κάνει ένα προϊόν από την κατασκευή του (με τη μορφή των υλικών-συστατικών του), τη διακομιδή και αποθήκευσή του μέχρι την πώλησή του. Πρόκειται, δηλαδή, για την εποπτική παρακολούθηση όλων των σταδίων της ζωής ενός προϊόντος με συμμετέχοντες (πιθανώς) διαφορετικούς σε περισσότερα από ένα από αυτά τα στάδια.

Η παρακολούθηση των ενεργειών στην πορεία που περιγράφηκε δεν είναι εύκολη ούτε απλή. Έχει ενισχυθεί από τη χρήση λύσεων του *Διαδικτύου των Πραγμάτων* (*Internet of Things, IoT*), αλλά έχει πολλά κενά στη διαχείριση, στη συλλογή και στην αποθήκευση των δεδομένων. Εκεί μπορεί να βοηθήσει η τεχνολογία του blockchain, καθώς γίνεται να προσφέρει ένα ενδιάμεσο στάδιο το οποίο θα ενώνει τις διάφορες ετερογενείς λύσεις που υπάρχουν χρησιμοποιώντας μια κοινή γλώσσα για την εύρυθμη λειτουργία του συστήματος. Επιπλέον, θα μπορεί να αποθηκεύει τα δεδομένα σε όλα τα στάδια, αποδίδοντας με κατάλληλα smart contracts την άδεια σε εγκεκριμένα μέρη να εποπτεύουν καθορισμένα σημεία της πορείας. Το πιο σημαντικό όμως είναι ότι μπορεί να δώσει πρόσβαση στον τελικό χρήστη σε κρίσιμη ως προς την επιλογή του προϊόντος πληροφορία, με την εγγύηση ότι αυτή δεν έχει αλλοιωθεί μέσα στην αλυσίδα.

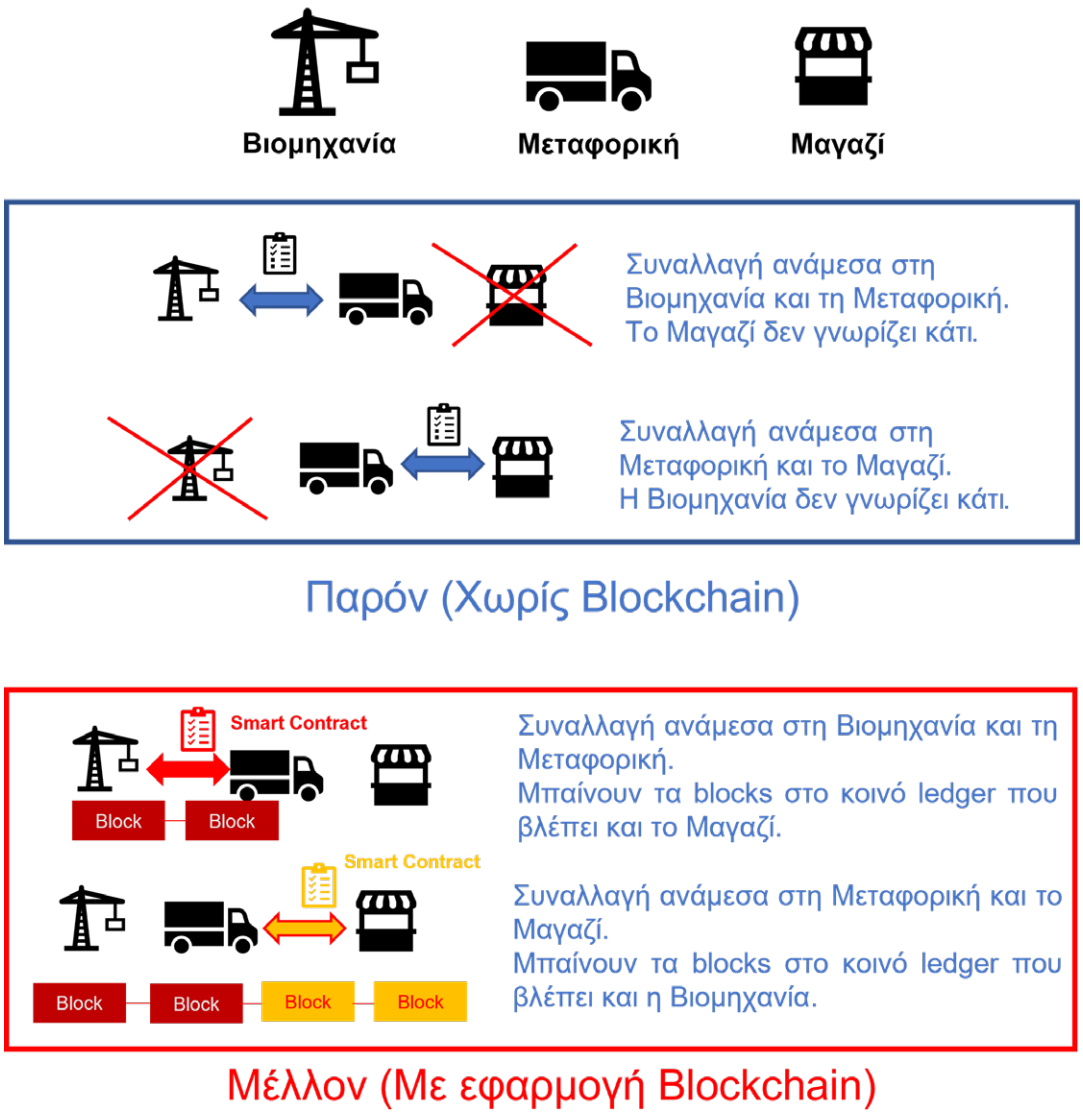
Γενικά, με τη βοήθεια των ιδιαίτερων χαρακτηριστικών της τεχνολογίας, έχουν σχεδιαστεί και υλοποιηθεί λύσεις που προσφέρουν (Gaur & Gaiha, 2020):

- *Διαφάνεια στον εντοπισμό προϊόντων και συναλλαγών*: Με την είσοδο στο ledger δεδομένων τόσο θέσης όσο και πιο εξειδικευμένων πληροφοριών (π.χ. θερμοκρασία μεταφοράς για ευπαθή προϊόντα) από όλα τα στάδια της ΕΑ επιτρέπεται στους εγκεκριμένους χρήστες αλλά και στον τελικό χρήστη να παρακολουθεί την εξέλιξη της πορείας ενός προϊόντος και να βεβαιωθεί ότι δεν υπάρχει κάποια αλλοίωση σε αυτήν. Επιπλέον, η αδυναμία διαγραφής δεδομένων από το ledger επιτρέπει την ενίσχυση της διαφάνειας στη διαδικασία καθώς και την αύξηση της εμπιστοσύνης στο σύστημα.
- *Αυξημένη αποτελεσματικότητα*: Η δυνατότητα ελέγχου μέσω ενός κατανεμημένου δικτύου της πληροφορίας για το κάθε προϊόν επιτρέπει να βεβαιωθούν οι υπεύθυνοι του κάθε σταδίου μέσα στην ΕΑ ότι η πορεία του προϊόντος μέσα στο στάδιο ολοκληρώθηκε σύμφωνα με τα προβλεπόμενα πρότυπα και τις οδηγίες. Εξασφαλίζεται έτσι ότι, αν κάποιο προϊόν έχει αλλοιωθεί και πρέπει να αποσυρθεί από την αγορά, θα εντοπιστεί εύκολα η παρτίδα του και θα ολοκληρωθεί η δράση αποτελεσματικά και γρήγορα.
- *Σταθερότητα και εμπιστοσύνη*: Με την αύξηση της διαφάνειας και της αποτελεσματικότητας αυξάνεται και η εμπιστοσύνη στο σύστημα. Ενδεικτικά αναφέρεται ότι ένα από τα προβλήματα που έφερε η

[111] Online Σύνδεσμος: https://www.uport.me/
[112] Online Σύνδεσμος: https://www.microsoft.com/en-us/security/business/microsoft-entra

πανδημία της Covid-19 ήταν και οι σοβαρές ελλείψεις που παρουσιάστηκαν σε πολλές ΕΑ. Ένας τρόπος αντιμετώπισης αυτού, που ενισχύει τη σταθερότητα του συστήματος, είναι η δυνατότητα να δοθεί πρόσβαση σε περισσότερους προμηθευτές έτσι ώστε να αντισταθμιστεί η έλλειψη που θα παρουσιάσουν οι βασικοί συνεργάτες/προμηθευτές μιας εταιρείας.

Για να προσφέρει το blockchain όλα αυτά τα χαρακτηριστικά, θα πρέπει να χρησιμοποιήσει smart contracts, όπως φαίνεται και στην **Εικόνα 9.2**. Η Εικόνα 9.2 απεικονίζει τη δυνατότητα παρακολούθησης των δεδομένων σε όλα τα στάδια, μιας και τα αποτελέσματα των συναλλαγών προστίθενται στο κοινό ledger και μπορεί να τα αναζητήσει και μέλος της ΕΑ που δεν συμμετέχει σε αυτές.[113]



Εικόνα 9.2 *Παράδειγμα διαχείρισης της εφοδιαστικής αλυσίδας χωρίς τη χρήση λύσης που βασίζεται στο blockchain (πάνω) και με λύση που βασίζεται στο blockchain και στα smart contracts (κάτω).*

Επίσης, ο **Πίνακας 9.2** περιέχει τα σημεία που χρειάζονται αντιμετώπιση στην προσέγγιση της διαχείρισης ΕΑ χωρίς blockchain, καθώς και τα πλεονεκτήματα που μπορεί να έχει η εφαρμογή μιας λύσης που βασίζεται στη δημοφιλή τεχνολογία.

[113] Συνήθως αναπτύσσονται ρόλοι και δικαιώματα χρήσης έτσι ώστε ευαίσθητες εμπορικές λεπτομέρειες να μην είναι προσβάσιμες από τρίτα μέρη. Αλλά η σημαντική πληροφορία μπορεί να είναι.

Προβλήματα στην απλή περίπτωση (χωρίς blockchain)	Πλεονεκτήματα της περίπτωσης εφαρμογής blockchain λύσης
Περιορισμένη ορατότητα και πρόσβαση στα δεδομένα λόγω των ποικίλων ανεξάρτητων υλοποιήσεων και της πρόθεσης διαμοιρασμού τους μόνο με συγκεκριμένους εταίρους.	Απλοποίηση των σύνθετων συστημάτων παράδοσης.
Ανάγκη για ομοιόμορφη καταχώριση των δεδομένων για την αποδοτικότερη επεξεργασία τους.	Επίτευξη παρακολούθησης αποθέματος με μεγάλη λεπτομέρεια καθώς και διασφάλιση των παραδόσεων. Μπορεί να οδηγήσει δυνητικά σε βελτίωση της χρηματοδότησης στην ΕΑ καθώς και στην επίτευξη ασφάλειας και μείωσης των κινδύνων.
Ασυμβατότητες στη μορφή των δεδομένων και κενά στην παρακολούθηση της πορείας των προϊόντων στην ΕΑ.	Βελτίωση στην ανίχνευση, στον εντοπισμό και στην επαλήθευση για την αντιμετώπιση των περιστατικών κλοπής ή απάτης.

Πίνακας 9.2 *Μειονεκτήματα και πλεονεκτήματα της περίπτωσης που δεν εφαρμόζεται μια λύση blockchain σε σύγκριση με αυτήν όπου εφαρμόζεται όσον αφορά τη διαχείριση της ΕΑ.*

Για την επίτευξη όμως της ολοκληρωμένης μετάβασης σε μια κατάσταση που εφαρμόζεται πλήρως ένα σύστημα blockchain είναι αναγκαίο να ολοκληρωθούν ακόμα ορισμένα βήματα. Σε αυτά περιλαμβάνονται:

- Η χρήση έμπιστων oracles (βλ. Ενότητα 2) για την επιβεβαίωση των καταχωρίσεων από μια οντότητα ως προς την ορθότητά τους (τα δεδομένα δεν θα αλλαχθούν, επομένως θα πρέπει να είναι ορθά για να είναι ωφέλιμα).
- Η παροχή, ο διαμοιρασμός και η ανανέωση ταυτοτήτων στους εμπλεκόμενους για να είναι διακριτοί οι ρόλοι τους μέσα σε μια ΕΑ.

Επιπλέον, θα πρέπει να τονιστεί ότι οι λύσεις που έχουν εφαρμοστεί, κατά κύριο λόγο, περιέχουν εφαρμογή ιδιωτικών ή υβριδικών δικτύων blockchain για να μπορέσουν να διαχειριστούν τόσο τις συναλλαγές που αφορούν εταιρικά δεδομένα όσο και αυτές που αφορούν δημόσια.

Ανάμεσα στις λύσεις που έχουν αρχίσει και χρησιμοποιούνται είναι το IBM Food Trust[114] (Xevgenis et al., 2020), που δημιουργήθηκε και υποστηρίζεται από την IBM. Η λύση αυτή επικεντρώνεται στην τροφική αλυσίδα και επιτυγχάνει να δείξει τη σημασία της ενισχυμένης ανίχνευσης και του εντοπισμού που φέρνει η τεχνολογία του blockchain. Με την ανάπτυξη ενός ιδιωτικού δικτύου blockchain που βασίζεται στην πλατφόρμα του Hyperledger Fabric, που υποστηρίζει η IBM, επιτυγχάνεται η υλοποίηση ελέγχου πρόσβασης στα δεδομένα ανάλογα με τον ρόλο του κάθε συμμετέχοντος στην τροφική αλυσίδα. Επίσης, δίνεται η δυνατότητα στους δημιουργούς των δεδομένων να ελέγξουν και εκείνοι την πρόσβαση τρίτων σε αυτά. Επιπλέον, χρησιμοποιείται ένας μοναδικός αριθμός (όπως είναι ο Universal Product Code ή το Global Trade Item Number) για την αναζήτηση ενός προϊόντος μέσα στην αλυσίδα.

Μια ακόμα γνωστή λύση στον χώρο είναι η CargoX[115], η οποία χρησιμοποιεί λύση που είναι προσαρμοσμένη στη ναυτιλία και στη δημιουργία, διαχείριση και διανομή των απαραίτητων εγγράφων για ένα τέτοιο ταξίδι. Βασισμένο στο Ethereum, δημιουργεί συναλλαγές, με τους συμμετέχοντες να γνωρίζουν τις απαραίτητες διευθύνσεις για την παρακολούθηση της πορείας της συναλλαγής στο δίκτυο, αποτρέποντας τρίτους, που δεν έχουν την πληροφορία αυτή, από το να την αποκτήσουν.

9.2.3 Η διαχείριση ιατρικών και κλινικών δεδομένων

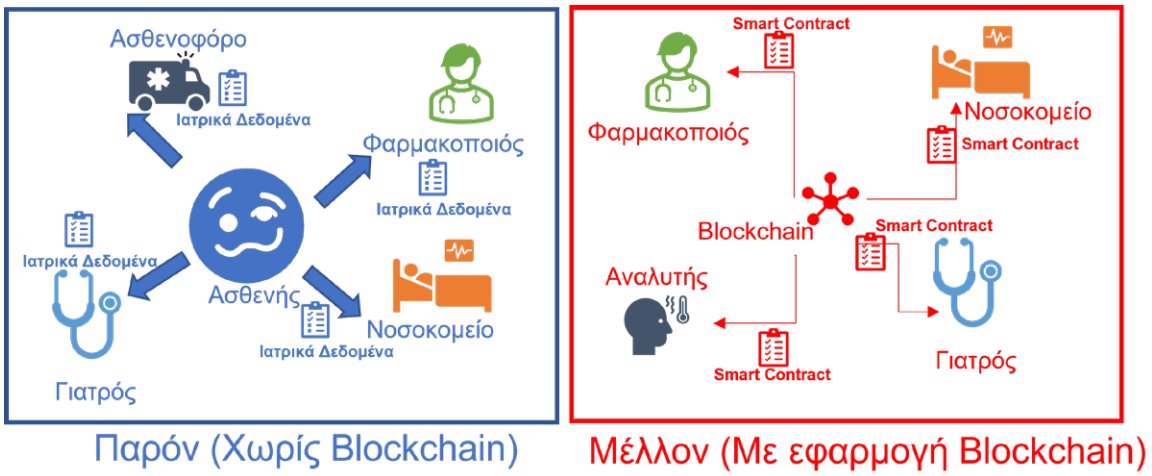
Η περίπτωση των ιατρικών και κλινικών δεδομένων είναι ιδιαίτερα κρίσιμη, μιας και περιέχει δεδομένα που είναι ευαίσθητα και, συχνά, προσωπικά. Στην περίπτωση αυτή τίθεται θέμα διαχείρισης των δεδομένων ασθενών, που διατηρούνται σε ξεχωριστά αρχεία. Αυτός ο διαχωρισμός των δεδομένων σε ξεχωριστές εγγραφές είναι σημαντικό να σταματήσει και να αντικατασταθεί από μια πηγή στην οποία θα έχουν πρόσβαση όλοι οι εμπλεκόμενοι στη διαδικασία (γιατροί, ασθενείς, νοσοκομεία, φαρμακοποιοί). Το αποτέλεσμα είναι να υπάρχει πιο γρήγορη διάγνωση που θα είναι και πιο αποτελεσματική, μιας και θα έχει περισσότερα δεδομένα για να βγει.

Σημαντικό όμως για την εύρεση μιας κατανεμημένης λύσης στο θέμα είναι η δυνατότητα επιλογής από τον κάθε ασθενή για το ποιος θα έχει πρόσβαση στα δεδομένα του. Αυτό λύνεται με τη βοήθεια του blockchain και τη χρήση της ασύμμετρης κρυπτογραφίας και της σύνταξης smart contracts που χρησιμοποιούνται σε αυτό.

[114] Online Σύνδεσμος: https://www.ibm.com/blockchain/solutions/food-trust
[115] Online Σύνδεσμος: https://cargox.io/

Έτσι, θα είναι δυνατός ο συντονισμός της πληροφορίας που χρειάζεται για την αντιμετώπιση ενός συμβάντος από όλα τα μέρη που πρέπει να συμμετάσχουν σε αυτό.

Αυτό φαίνεται στην **Εικόνα 9.3**, όπου αριστερά βλέπουμε τα διαφορετικά αρχεία που κρατά το κάθε εμπλεκόμενο μέρος, ενώ δεξιά φαίνεται πως είναι δυνατόν ο κάθε εμπλεκόμενος να πάρει πρόσβαση – μέσω smart contracts– στην πληροφορία που τον αφορά, χρησιμοποιώντας το blockchain ως το κοινό μέσο αποθήκευσης των δεδομένων.



Εικόνα 9.3 *Παράδειγμα διαχείρισης ιατρικών/κλινικών δεδομένων χωρίς τη χρήση λύσης που βασίζεται στο blockchain (αριστερά) και με λύση που βασίζεται στο blockchain και στα smart contracts (δεξιά).*

Όσον αφορά τα δεδομένα που προκύπτουν κατά τη διάρκεια κλινικών δοκιμών για την εύρεση νέων θεραπειών καταπολέμησης σοβαρών ασθενειών, εδώ η τεχνολογία του blockchain μπορεί να βοηθήσει να διευκολυνθεί η επικοινωνία μεταξύ ομάδων που εκτελούν πειράματα σε παρόμοιες ασθένειες. Με τον τρόπο αυτόν όλοι μπορούν να βοηθηθούν από όσα έχουν καταγράψει οι συνάδελφοί τους, έτσι ώστε να μπορέσουν να βελτιώσουν πιο γρήγορα τα αποτελέσματά τους και να προχωρήσουν την έρευνά τους.

Ο **Πίνακας 9.3** παρουσιάζει τα μειονεκτήματα της τρέχουσας χωρίς χρήση blockchain υλοποίησης και υπογραμμίζει τα πλεονεκτήματα της χρήσης μιας λύσης που εφαρμόζει smart contracts και υποστηρίζεται από ένα δίκτυο blockchain.

Προβλήματα στην απλή περίπτωση (χωρίς blockchain)	Πλεονεκτήματα της περίπτωσης εφαρμογής λύσης με blockchain
Καθυστερήσεις στην αντιμετώπιση πανδημιών λόγω εμποδίων στον διαμοιρασμό δεδομένων ανάμεσα στα ερευνητικά εργαστήρια.	Αυξημένη ορατότητα και μείωση στο κόστος με τον εξορθολογισμό των διαδικασιών εγκατάστασης για δοκιμές.
Περιορισμός στην κατανόηση των πλεονεκτημάτων/μειονεκτημάτων μιας θεραπείας λόγω ελλιπούς ενημέρωσης.	Αυξημένη πρόσβαση σε δεδομένα από διάφορα ινστιτούτα κατά τη διάρκεια επιδημιών, τα οποία προστατεύονται με τη διατήρηση της ιδιωτικότητας.
Περιορισμένη συμμετοχή του ασθενή λόγω της έλλειψης συνεπούς διαχείρισης της συναίνεσης.	Αυξημένη αυτοματοποίηση στην απόκτηση και στον εντοπισμό της συναίνεσης για την πρόσβαση στα διαμοιραζόμενα δεδομένα.
Προβλήματα στη διατήρηση του απορρήτου και στην ταυτοποίηση των ασθενών λόγω της μεταφοράς και του διαμοιρασμού των δεδομένων.	Αυξημένη εμπιστοσύνη στην ιδιωτικότητα των ασθενών.

Πίνακας 9.3 *Μειονεκτήματα και πλεονεκτήματα της περίπτωσης που δεν εφαρμόζεται μια λύση blockchain σε σύγκριση με αυτήν όπου εφαρμόζεται όσον αφορά τα δεδομένα υγείας ή/και των κλινικών δεδομένων.*

Για να είναι εφικτή όμως η δημιουργία μιας λύσης που αξιοποιεί τα πλεονεκτήματα της τεχνολογίας blockchain, θα πρέπει να αντιμετωπιστούν, κυρίως μέσω της σύνταξης των κατάλληλων smart contracts, οι παρακάτω προκλήσεις:

- Η προσαρμογή των διαδικασιών και των ομάδων στη διαχείριση ενός σημαντικού αριθμού δεδομένων από άλλες δοκιμές, καθώς και η δυνατότητα διαμοιρασμού μεθόδων και τεχνικών με σκοπό να επιτευχθεί το επιθυμητό αποτέλεσμα σε μικρότερο χρονικό διάστημα. Πιθανώς να χρειαστεί και η ανάπτυξη νέων τρόπων υπολογισμού αποτελεσμάτων και τρόπων (συν)εργασίας.

- Θα πρέπει να βρεθεί ένας ασφαλής και αποτελεσματικός τρόπος έτσι ώστε να εξασφαλιστεί η ταυτοποίηση, η αυθεντικοποίηση και η έγκριση πρόσβασης με τη χρήση smart contracts που τρέχουν σε ενεργά δίκτυα blockchain.
- Παροχή οικονομικών κινήτρων για τη δημιουργία νέων αγορών για τα δεδομένα (όπως είναι τα κλινικά δεδομένα) που μπορούν να οδηγήσουν σε μια νέα ανάπτυξη στον χώρο των κλινικών δοκιμών.

Ορισμένες λύσεις που έχουν παρουσιαστεί στην αγορά περιλαμβάνουν την Chronicled[116], η οποία εξειδικεύεται στην εξυπηρέτηση των συναλλαγών μεταξύ εταιρειών που δραστηριοποιούνται στον τομέα υγείας. Πιο συγκεκριμένα, η εταιρεία διατηρεί σε blockchain τα συμβόλαια μεταξύ των μερών για αναφορά και έλεγχο. Επιπλέον, υπάρχει η WholeCare[117], η οποία εξειδικεύεται στην εύρεση λύσεων παροχής βοήθειας σε όσους την έχουν ανάγκη, με την προσφορά μιας σειράς από λύσεις μέσω μιας πλατφόρμας που βασίζεται στην τεχνολογία του blockchain. Επιπλέον, υπάρχει το Patientory[118], ένα σύστημα καταγραφής ιατρικών δεδομένων για ασθενείς και ιατρούς που βασίζεται επίσης στην τεχνολογία του blockchain. Πέρα από τη δυνατότητα εγκεκριμένου διαμοιρασμού των ιατρικών δεδομένων, η πλατφόρμα εξυπηρετεί και την επικοινωνία μεταξύ του ιατρού και του ασθενή. Τέλος, υπάρχει και η Nebula Genomics[119], μια εταιρεία που ειδικεύεται στην κατανόηση του ανθρώπινου γονιδιώματος. Συλλέγει ατομικές πληροφορίες από ασθενείς και τις προσθέτει ανώνυμα μέσω χρήσης κρυπτογραφίας στο δίκτυο blockchain για να χρησιμοποιηθούν ως αναφορά, χωρίς να μπορεί είτε να αναγνωριστεί ο κάτοχος είτε να κλαπούν.

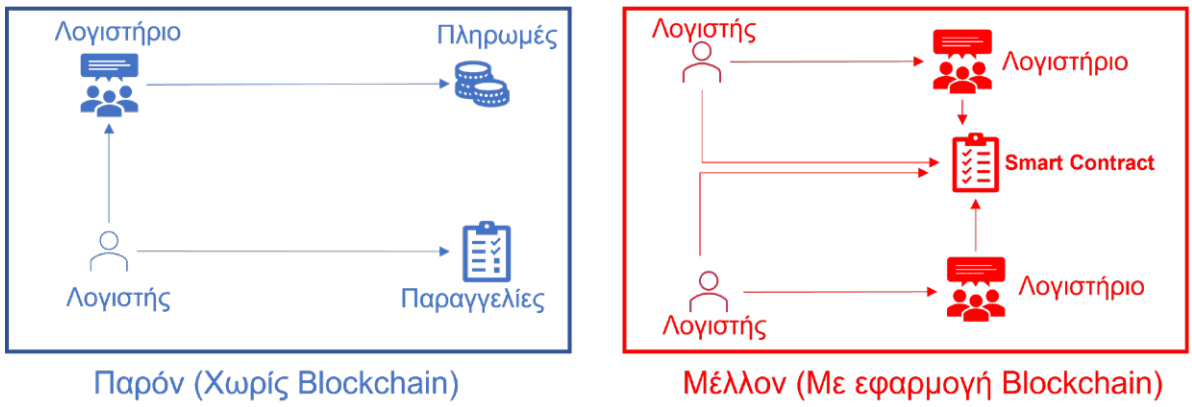
Στις περισσότερες από τις προαναφερθείσες περιπτώσεις τα δίκτυα blockchain που χρησιμοποιούνται είναι ιδιωτικά, για να εξασφαλίσουν την ελεγχόμενη πρόσβαση σε αυτά. Επιπλέον, η χρήση smart contracts μπορεί να βοηθήσει στην απόδοση πρόσβασης στα δεδομένα που αποθηκεύονται στο blockchain.

9.2.4 Η εφαρμογή στην οικονομία και στο εμπόριο

Η οικονομία αποτελεί το πιο χαρακτηριστικό παράδειγμα σεναρίου χρήσης της τεχνολογίας του blockchain. Είναι ο χώρος που αναδεικνύει τα πιο πρωτοποριακά χαρακτηριστικά της τεχνολογίας και στον οποίο έχουν δοκιμαστεί πολλές λύσεις.

Ενδεικτικά αναφέρεται η δυνατότητα διαχείρισης των οικονομικών εγγραφών μιας εταιρείας, η οποία θα επιτρέψει την ακριβή και διαφανή συλλογή των δεδομένων, τα οποία μέσω των smart contracts μπορούν να αποκτούν μια ομοιόμορφη δομή. Με τον τρόπο αυτό διευκολύνεται η δημιουργία αναφορών, η διαχείριση των δεδομένων και ο έλεγχός τους από εσωτερικούς ή εξωτερικούς φορείς.

Η **Εικόνα 9.4** δείχνει τη διαφορά ανάμεσα στις περιπτώσεις χρήσης blockchain, καθώς πλέον αυτό μπορεί να αποτελέσει σημείο αναφοράς τόσο εσωτερικά στην εταιρεία όσο και στις επαφές της με άλλους συνεργάτες εκτός αυτής. Η λύση αυτή προκρίνει τη χρήση ενός ιδιωτικού blockchain, καθώς απαιτείται έλεγχος για την πρόσβαση σε αυτό, καθώς και διαχωρισμός στο επίπεδο πρόσβασης των συμμετεχόντων.



Εικόνα 9.4 *Παράδειγμα διαχείρισης οικονομικών εγγραφών και δεδομένων χωρίς τη χρήση λύσης που βασίζεται στο blockchain (αριστερά) και με λύση που βασίζεται στο blockchain και στα smart contracts (δεξιά).*

[116] Online σύνδεσμος: https://www.chronicled.com/
[117] Online σύνδεσμος: https://wholecarehub.com/
[118] Online σύνδεσμος: https://builtin.com/company/patientory
[119] Online σύνδεσμος: https://builtin.com/company/nebula-genomics

Ο **Πίνακας 9.4** συνοψίζει τα ζητήματα που παραμένουν ανοικτά με την τωρινή προσέγγιση καθώς και τι μπορεί να φέρει ως πλεονέκτημα μια λύση που εφαρμόζει τα χαρακτηριστικά ενός blockchain.

Προβλήματα στην περίπτωση άνευ blockchain	Πλεονεκτήματα της περίπτωσης εφαρμογής λύσης με blockchain
Τα λογιστικά συστήματα είναι επιρρεπή στην απάτη και στα σφάλματα, καθώς ελέγχονται απευθείας από τις οικείες οντότητες.	Βελτιωμένη ακεραιότητα των δεδομένων στις συναλλαγές που οδηγεί σε αυξημένη σταθεροποίηση της αγοράς.
Απαιτούν σημαντικά έξοδα λειτουργίας, καθώς η κάθε εταιρεία διατηρεί τις δικές της υποδομές.	Μείωση των εξόδων για λογιστικά πληροφοριακά συστήματα μέσω του διαμοιρασμού του κόστους σε πολλούς οργανισμούς.
Απαιτείται σημαντικό μέρος του προσωπικού για την επεξεργασία των συναλλαγών με συστήματα που δεν αλληλεπιδρούν με αυτά της εταιρείας.	Βελτίωση του εταιρικού κεφαλαίου λόγω της μεγαλύτερης οικονομικής ευχέρειας.

Πίνακας 9.4 *Μειονεκτήματα και πλεονεκτήματα της περίπτωσης που δεν εφαρμόζεται μια λύση blockchain σε σύγκριση με αυτήν όπου εφαρμόζεται όσον αφορά την οικονομική διαχείριση.*

Για να γίνει δυνατή η πλήρης προσαρμογή των συστημάτων στην εφαρμογή λύσεων που βασίζονται στην τεχνολογία του blockchain, θα πρέπει να ξεπεραστούν και ορισμένες προκλήσεις, όπως:

- Η ανάπτυξη μιας πλατφόρμας που θα βοηθά στη δημιουργία smart contracts που χρησιμοποιούνται για τη σύνταξη οικονομικών συναλλαγών.
- Δημιουργία ενός συνόλου από πρότυπα που αφορούν τα ψηφιακά αντικείμενα που παίρνουν τη μορφή token.
- Διαλειτουργικότητα μεταξύ των δικτύων blockchain (ή και DLT) και των συμβατικών οικονομικών συστημάτων.
- Δημιουργία μιας ομάδας ειδικών για τον έλεγχο της ορθότητας των smart contracts που διέπουν τις οικονομικές συναλλαγές.

Προϊόντα που έχουν δημιουργηθεί στον τομέα της *Αποκεντρωμένης Οικονομίας* (*Decentralized Finance, DeFi*) περιλαμβάνουν λύσεις όπως αυτή της Algorand[120], η οποία προσφέρει μια σειρά από προϊόντα νέας γενιάς καθώς και πρωτόκολλα που επιτρέπουν να μειωθεί η διαφορά ανάμεσα στις συναλλαγές στον πραγματικό και στον ψηφιακό κόσμο. Κάνει χρήση ενός συνόλου από layer-1 blockchains για να προσφέρει επεκτασιμότητα, ασφάλεια και οριστικοποίηση συναλλαγών. Αυτές βρίσκουν εφαρμογή σε περιπτώσεις χρήσης που περιλαμβάνουν τη σύνταξη ασφαλιστικών συμβολαίων, τις εφοδιαστικές αλυσίδες και τη βιομηχανία των ψηφιακών παιχνιδιών. Ακόμα, η Chain.io[121] δημιουργεί υποδομές νεφοϋπολογιστικής βασισμένες στο blockchain, που χρησιμοποιούνται για την προσφορά οικονομικών υπηρεσιών, όπως είναι η μεταφορά κρυπτονομισμάτων.

Από τεχνικής πλευράς, οι λύσεις βασίζονται σε ιδιωτικά (ή υβριδικά) δίκτυα, καθώς, λόγω της φύσεώς τους, ενισχύεται ο έλεγχος των συμμετεχόντων σε αυτά.

9.2.5 Η εφαρμογή στις δημόσιες υπηρεσίες

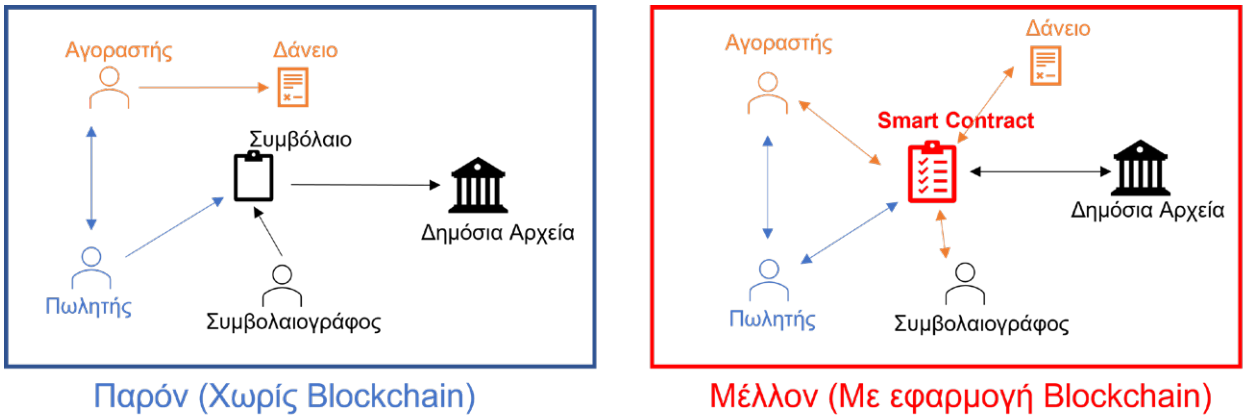
Μια από τις περιπτώσεις χρήσης της τεχνολογίας blockchain που έχει δείξει μεγάλη δυναμική, αν και ακόμα δεν έχουν υπάρξει οι αναμενόμενες εξελίξεις, αποτελεί και η δυνατότητα χρήσης της στον δημόσιο τομέα. Σκοπός είναι να ελαφρύνει τη γραφειοκρατία, να αυξήσει την ασφάλεια των εγγράφων, να επιτύχει την έγκυρη και ουσιαστική λογοδοσία για αμέλεια ή λάθη και να αυξήσει την αποτελεσματικότητα στην ολοκλήρωση των χρονοβόρων διεργασιών του δημόσιου τομέα. Παράδειγμα μιας υπηρεσίας που μπορεί να επηρεαστεί σημαντικά είναι η καταγραφή της ιδιοκτησίας κατοικιών και η αγορά/μεταβίβασή τους.

Με το blockchain στην περίπτωση αυτή, με τη βοήθεια από ειδικά smart contracts είναι δυνατόν να περιοριστεί η απάτη και να ενισχυθούν η ακεραιότητα των συναλλαγών, η αποτελεσματικότητα και η διαφάνειά τους.

Στην **Εικόνα 9.5** απεικονίζεται η κατάσταση λειτουργίας ενός παραδοσιακού συστήματος για την καταγραφή της ιδιοκτησίας και τη μεταφορά κατά την πώλησή της, που περιλαμβάνει, μεταξύ άλλων, συμβολαιογράφους, ιδιοκτήτες ή/και πωλητές, υποθήκη και καταγραφή στα δημόσια έγγραφα. Κατά κύριο λόγο, όλοι έρχονται σε

[120] Online σύνδεσμος: https://www.algorand.com/
[121] Online σύνδεσμος: https://chain.io/

επαφή με τον συμβολαιογράφο που φτιάχνει το συμβόλαιο (τίτλος ιδιοκτησίας) και το καταθέτει στο Δημόσιο. Επιπλέον, στην Εικόνα 9.5 απεικονίζεται πώς μπορεί να βελτιωθεί το υπάρχον σύστημα με την προσθήκη smart contracts που υποστηρίζονται από ένα δίκτυο blockchain. Στην περίπτωση αυτή, το smart contract βρίσκεται στην καρδιά των συναλλαγών και όλες οι προαναφερθείσες οντότητες αλληλεπιδρούν με αυτό, είτε εκκινώντας τη διαδικασία είτε δεχόμενες προτροπή από αυτό.



Εικόνα 9.5 *Παράδειγμα διαχείρισης εγγραφών ιδιοκτησίας χωρίς τη χρήση λύσης που βασίζεται στο blockchain (αριστερά) και με λύση που βασίζεται στο blockchain και στα smart contracts (δεξιά).*

Ο **Πίνακας 9.5** συνοψίζει τα μειονεκτήματα της τρέχουσας κατάστασης και τα θετικά που φέρνει η εφαρμογή μιας λύσης που βασίζεται στο blockchain.

Προβλήματα στην περίπτωση χωρίς blockchain	**Πλεονεκτήματα της περίπτωσης εφαρμογής λύσης με blockchain**
Απαιτήσεις για χρήση κεφαλαίων λόγω ασυμβατότητας των υποδομών.	Μεγαλύτερη εμπιστοσύνη στις ταυτότητες των εμπλεκομένων, εξορθολογισμός των επιμέρους διεργασιών και μείωση των εξόδων ελέγχου.
Μη αποτελεσματική επιβεβαίωση ταυτοτήτων και αργή διαδικασία συλλογής υπογραφών.	Αυτοματοποιημένες ειδοποιήσεις και ενσωμάτωση διαδικασιών ασφάλειας και προστασίας της ακεραιότητας των εγγράφων.
Οι χειροκίνητες διαδικασίες καθυστερούν την ολοκλήρωση των σταδίων και δημιουργούν προϋποθέσεις για πιθανή αλλοίωση ή απώλεια εγγράφων.	Μείωση του αριθμού από απάτες στη μεταβίβαση ιδιοκτησίας.
Πολλά άτομα μπορεί να εμφανίζονται (εσφαλμένα) ως ιδιοκτήτες ενός ακινήτου, χωρίς να είναι αυτό αληθές.	Ενισχυμένη ρευστότητα.

Πίνακας 9.5 *Μειονεκτήματα και πλεονεκτήματα της περίπτωσης που δεν εφαρμόζεται μια λύση blockchain σε σύγκριση με αυτήν όπου εφαρμόζεται όσον αφορά την ψηφιακή διαχείριση τίτλων ιδιοκτησίας.*

Οι δυσκολίες που πρέπει να αντιμετωπιστούν προτού μια λύση όπως αυτή που περιγράφηκε προηγουμένως χρησιμοποιηθεί περιλαμβάνουν:

- Τη χρήση μιας κοινής, τυποποιημένης μορφής για την εγγραφή των δεδομένων από όλους τους συμμετέχοντες στη σύνταξη ενός συμβολαίου. Η κοινή αυτή μορφή θα περιέχει έναν εμφανή χώρο για τη συλλογή των δεδομένων καθώς και τα πεδία για τις υπογραφές.
- Την ανάπτυξη κοινών πρωτοκόλλων για την επικοινωνία μεταξύ των συμμετεχόντων μερών καθώς και με το δημόσιο αρχείο.
- Χρήση και πιστοποίηση των ψηφιακών ταυτοτήτων.

Οι τρόποι με τους οποίους επηρεάζεται ο δημόσιος τομέας, όμως, από τη χρήση της τεχνολογίας του blockchain δεν περιορίζεται μόνο στη διαχείριση (και εναλλαγή) των ψηφιακών τίτλων ιδιοκτησίας αλλά επεκτείνεται και σε άλλους τομείς, όπως: η διενέργεια εκλογών, η έκδοση πιστοποιητικών ή διαχείριση vouchers για αγορές. Μάλιστα, έχουν αρχίσει και κάνουν την εμφάνισή τους εφαρμογές που επιδιώκουν να καλύψουν ένα ολοένα και πιο ευρύ φάσμα υλοποιήσεων.

Έτσι, υπάρχει η εφαρμογή Kaleido[122], η οποία και προσφέρει λύσεις με χρήση της τεχνολογίας του blockchain οι οποίες και καλύπτουν ένα ευρύ φάσμα περιπτώσεων χρήσης (7 τέτοιες περιπτώσεις), με τον δημόσιο τομέα να αποτελεί τη μία από αυτές. Οι υπηρεσίες της λύσης καλύπτουν τη μείωση του κινδύνου και τον έλεγχο καθώς και τον εξορθολογισμό των κρατικών υπηρεσιών στην υγεία, στην εκπαίδευση και στη μισθοδοσία. Επιπλέον, υπάρχει η εφαρμογή Voatz[123], η οποία και προσφέρει μια πλατφόρμα για ψηφοφορίες κάνοντας χρήση ενός δικού της κρυπτογραφημένου συστήματος βιομετρικών το οποίο επιτρέπει τη χρήση του κινητού τηλεφώνου για την απρόσκοπτη συμμετοχή στις εκλογές. Παρόμοια λειτουργία συναντούμε και από τη Follow my Vote[124], που προσφέρει μια εικονική ψηφιακή κάλπη για τη διαδικασία.

Οι λύσεις που υλοποιούνται μπορεί να περιλαμβάνουν την ανάπτυξη δημόσιου ή υβριδικού τύπου blockchain, καθώς έχει σημασία να είναι προσβάσιμο από όλους τους πολίτες αλλά και να είναι δυνατή η μοναδική αναγνώρισή τους μέσα στο δίκτυο (π.χ. για τη συμμετοχή στις εκλογές).

9.2.6 Άλλες περιπτώσεις χρήσης

Η τεχνολογία blockchain είναι πάρα πολύ σημαντική και ικανή να δώσει λύσεις σε πολύ περισσότερες εφαρμογές από όσες παρουσιάστηκαν, ενδεικτικά, στο κεφάλαιο αυτό.

Πιο συγκεκριμένα, η χρήση του blockchain αναμένεται να έχει επίδραση:

- *Στην ανάπτυξη του Internet of Things* (*IoT*), μιας και θα αποτελεί την ιδανική πλατφόρμα για τη διαχείριση δεδομένων (αρκετές φορές ευαίσθητων, όπως είναι οι καρδιακοί παλμοί ενός ασθενή).
- *Στη βιομηχανία*: Με την πρόοδο στην ανάπτυξη smart contracts θα γίνει εφικτή η αυτοματοποίηση των βιομηχανικών διαδικασιών για την παραγωγή νέων (ή και υπαρχόντων) προϊόντων.
- *Στον τομέα των παιχνιδιών*: Η προσθήκη μοναδικών ψηφιακών αντικειμένων που μπορούν να γίνουν αντικείμενο συναλλαγής μέσα στο παιχνίδι θα αποτελέσει τη βάση για τη δημιουργία νέων παιχνιδιών με κατανεμημένους ρόλους.
- *Στην κοινωνική ζωή*: Το blockchain μπορεί να αποτελέσει το μέσο για τις ψηφιακές συναλλαγές σε έναν εικονικό κόσμο, όπως είναι αυτός που δημιουργείται μέσα σε ένα Metaverse. Μάλιστα, αναμένεται να διευκολύνει την αποθήκευση και τη μεταφορά των ψηφιακών χαρακτηριστικών του χρήστη όσο και των ψηφιακών αντικειμένων που του ανήκουν, συνεισφέροντας σε μια μοναδική εμπειρία χρήσης.
- *Στη ναυτιλία*, με τη συμβολή της στη δημιουργία έξυπνων λιμένων όπου η γραφειοκρατία θα έχει αντιμετωπιστεί και η αυτοματοποίηση των εγγραφών και των λειτουργών θα ελέγχεται με τη βοήθεια της τεχνολογίας.

Η χρήση και αξιοποίηση της τεχνολογίας του blockchain σε πολλά ετερογενή περιβάλλοντα χρήσης οφείλεται στη δυνατότητά της να παρουσιάζει διαφορετικούς τρόπους ανάπτυξης κάθε φορά. Η λύση του δικτύου δεν είναι μοναδική, αντιθέτως προσαρμόζει τα χαρακτηριστικά της (τύπος δικτύου, αλγόριθμος συναίνεσης) ανάλογα με την περίπτωση. Κοινό χαρακτηριστικό όμως αποτελεί η ανάγκη για διαμοιρασμό δεδομένων σε περισσότερους από έναν χρήστες μέσα σε ένα περιβάλλον το οποίο θα εμπνέει εμπιστοσύνη, αν τα συμμετέχοντα μέρη δεν εμπιστεύονται το ένα το άλλο.

[122] Online σύνδεσμος: https://www.kaleido.io/
[123] Online σύνδεσμος: https://voatz.com/
[124] Online σύνδεσμος: https://followmyvote.com/

Βιβλιογραφία

Bailey, D. (2022). *19 Blockchain application use cases that will surprise you*. Medium Online πηγή: https://medium.com/supplain/19-blockchain-application-use-cases-that-will-surprise-you-8f6792256bc8 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Gaur, V., & Gaiha, A. (2020). Building a Transparent Supply Chain. *Harvard Business Review*. May-June 2020. Online πηγή: https://hbr.org/2020/05/building-a-transparent-supply-chain [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Geroni, D. (2021). *Top 12 Smart Contracts Use Cases*. 101 Blockchain. Online πηγή: https://101blockchains.com/smart-contract-use-cases/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Daley, S. (2022). *34 Blockchain Applications and Real-World Use Cases*. Ενημερώθηκε τον Αύγουστο του 2022. Built In. Online πηγή: https://builtin.com/blockchain/blockchain-applications [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Szabo, N. (2016). Smart Contracts Alliance σε συνεργασία με την Deloitte με πρωτοβουλία του Chamber of Digital Commerce. *Smart Contracts: 12 Use Cases for Business & Beyond: A Technology, Legal & Regulatory Introduction*. Πρόλογος από Nick Szabo. December 2016. Online πηγή: https://www.perkinscoie.com/images/content/1/6/v2/164979/Smart-Contracts-12-Use-Cases-for-Business-Beyond.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Xevgenis, M., Kogias, D., Leligou, H., Chatzigeorgiou, C., Feidakis, M., & Patrikakis, C. (2020). A survey on the available blockchain platforms and protocols for Supply Chain Management. *IOT4SAFE 2020 the 1st International Workshop on IoT infrastructures for safety in pervasive environments, June 2, 2020 (Virtual)*.

W3C (2022). *Decentralized Identifiers (DIDs) v1.0: Core architecture, data model and representations*. W3C Recommendation. July 2022. Online πηγή: https://www.w3.org/TR/did-core/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΚΕΦΑΛΑΙΟ 10

Τεχνολογίες Κατανεμημένου Καθολικού (ΤΚΚ)

Σύνοψη

Το κεφάλαιο αυτό παρουσιάζει τα χαρακτηριστικά των λύσεων που ανήκουν στην κατηγορία των Τεχνολογιών Κατανεμημένου Καθολικού (ΤΚΚ), αναλύοντας τις ιδιότητές τους. Έμφαση δίνεται στις ομοιότητες και διαφορές που υπάρχουν ανάμεσα στα χαρακτηριστικά όλων των λύσεων που ανήκουν στην κατηγορία αυτή με αυτά της τεχνολογίας blockchain, η οποία και είναι επίσης μέρος της κατηγορίας.

Επιπλέον, ως παράδειγμα, αναλύεται μια χαρακτηριστική λύση ΤΚΚ, το δίκτυο του IOTA, για να καλυφθεί και μια διαφορετική λύση που είναι πολλά υποσχόμενη και ανερχόμενη. Τέλος, παρουσιάζονται σενάρια χρήσης τα οποία μπορούν να εξυπηρετηθούν από λύσεις που χαρακτηρίζονται ως ΤΚΚ (ή DLTs).

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2 και 3.

10.1 Τεχνολογίες Κατανεμημένου Καθολικού (ΤΚΚ): Τι είναι και πότε δημιουργήθηκαν

Οι *Τεχνολογίες Κατανεμημένου Καθολικού* ή αλλιώς *Distributed Ledger Technologies* (*DLTs*) αποτελούν μια νέα τεχνολογική περιοχή η οποία κερδίζει σημαντικό έδαφος με μεγάλη ταχύτητα στις προτιμήσεις του κοινού και της βιομηχανίας. Τα DLTs έχουν τα δικά τους βασικά χαρακτηριστικά, τα οποία συνολικά επιτρέπουν σε κάποιον να τα θεωρήσει, σε απλοϊκή απόδοση, ως σύγχρονες κατανεμημένες Βάσεις Δεδομένων.

Όμως, στην πράξη, τα DLTs είναι κάτι περισσότερο από μια απλή Βάση Δεδομένων, όπως γίνεται αντιληπτό από την παρουσίαση των χαρακτηριστικών τους που ακολουθεί.

Αν έπρεπε να περιγραφούν τα DLTs σε κάποιον μη γνώστη της εν λόγω τεχνολογίας, θα μπορούσαν να παρουσιαστούν ως μια τεχνολογία ομότιμων κόμβων οι οποίοι και επικοινωνούν μέσω του Διαδικτύου, με σχετικά κοντινά παραδείγματα τον διαμοιρασμό αρχείων και τη διαδικτυακή τηλεφωνία.

Στα πιο ιδιαίτερα χαρακτηριστικά των DLTs συμπεριλαμβάνονται:

- Μια νέα προσέγγιση στον έλεγχο, στην αποθήκευση και στον διαμοιρασμό των δεδομένων και της πληροφορίας με χρήση χρονοσφραγίδων (timestamps) για τη διατήρηση της τάξης σε αυτά.
- Η παρουσία της συνολικής (και κοινής) πληροφορίας σε πολλά κέντρα συλλογής δεδομένων που παίζουν τον ρόλο των κόμβων στο δίκτυο, με δυνατότητες ενημέρωσης και αποθήκευσης.
- Η διαχείριση της αποστολής πληροφορίας μέσω ενός κατανεμημένου δικτύου υπολογιστών που καλούνται να υλοποιήσουν αυστηρά πρωτόκολλα για την επίτευξη συναίνεσης και την αποθήκευση της πληροφορίας με χρήση τεχνικών κρυπτογραφίας και ψηφιακών υπογραφών.

Έχοντας κατά νου τα χαρακτηριστικά αυτά, πρέπει να αναφερθεί ότι τα DLTs ουσιαστικά κινούνται πάνω σε δύο βασικούς πυλώνες:

1) Στην αποτελεσματική λειτουργία του συστήματος, που συμπεριλαμβάνει τη δημιουργία, τον έλεγχο, την αποθήκευση και τον διαμοιρασμό των δεδομένων, *χωρίς την εμπλοκή κάποιας* (*«έμπιστης»*) *κεντρικής οντότητας* στη διαδικασία αυτή.
2) Στην αποφυγή του *double spending* (βλ. Κεφάλαιο 2), δηλαδή στο να μην είναι δυνατή η χρήση του ίδιου ψηφιακού αντικειμένου (token ή κρυπτονομίσματος – για τη διαφορά δείτε το Κεφάλαιο 7) σε περισσότερες από μία συναλλαγές ταυτόχρονα.

Η γέννηση των DLTs, ουσιαστικά, έγινε με την παρουσίαση και ανάπτυξη του δικτύου του Bitcoin, το οποίο και υποστηρίζεται από την τεχνολογία του blockchain. Εξάλλου, όπως μπορεί να γίνει κατανοητό από τα παραπάνω, η τεχνολογία του blockchain έχει πολλές ομοιότητες (βλ. Κεφάλαια 1-2) με τα χαρακτηριστικά των

συστημάτων DLT. Ο λόγος είναι απλός και σχετίζεται με το γεγονός ότι και η ίδια η τεχνολογία του blockchain αποτελεί ένα είδος συστήματος DLT.

Με τα συστήματα του blockchain να χαρακτηρίζονται από τη λειτουργία τους σε κατανεμημένα δίκτυα, χωρίς να υπάρχει κάποια κεντρική οντότητα, στα οποία η πληροφορία (συναλλαγές) καταγράφεται σε ένα ολοένα και αυξανόμενο κατάστιχο, το οποίο και δεν μπορεί να διαιρεθεί ούτε να σβηστεί κάποια είσοδος, αυτά καλύπτουν τα βασικά χαρακτηριστικά των DLTs. Επιπλέον, τα δίκτυα blockchain παρουσιάζουν λύση στο πρόβλημα του double spending με τη χρήση ισχυρής κρυπτογραφίας και ψηφιακών υπογραφών στις συναλλαγές, με την εφαρμογή κανόνων συναίνεσης για την εγγραφή νέων δεδομένων στο ledger.

Ταυτόχρονα, όμως, τα δίκτυα blockchain παρουσιάζουν και μια σημαντική *διαφορά* σε σχέση με τα συστήματα DLTs. Η διαφορά αυτή είναι στον τρόπο με τον οποίο αποθηκεύουν την πληροφορία αυτή στο ledger, με τη δημιουργία blocks από συναλλαγές τα οποία συνδέονται με κρυπτογραφία μεταξύ τους, δημιουργώντας αλυσίδες που ολοένα και αυξάνουν σε μέγεθος καθώς είναι αδιαίρετες.

Σε αντίθεση με το blockchain, σε ένα σύστημα DLT η πληροφορία μπορεί να αποθηκεύεται στο κοινό ledger με διαφορετικούς τρόπους, όπως θα φανεί και στην επόμενη υποενότητα του κεφαλαίου.

Το σημαντικό όμως είναι ότι η πληροφορία αυτή, όπως και να είναι αποθηκευμένη, δεν μπορεί να αλλοιωθεί και διαμοιράζεται σε όλους τους κόμβους του δικτύου για επιβεβαίωση και αποθήκευση.

10.2 Οι τύποι των DLTs και η χρήση τους

Τα DLTs, παρόμοια με τα δίκτυα blockchain, χωρίζονται σε δύο βασικές κατηγορίες, αναλόγως με το πώς επιτυγχάνεται η πρόσβαση στο δίκτυο αλλά και αν είναι δυνατή η πρόσβαση στο ledger και η εγγραφή σε αυτό. Έτσι, λοιπόν, συναντούμε DLTs τα οποία και απαιτούν άδεια για την είσοδο στο δίκτυο και για να προτείνουν οι κόμβοι προσθήκες στο κοινό ledger (permissioned DLTs). Επίσης, υπάρχουν και DLTs τα οποία επιτρέπουν την ελεύθερη πρόσβαση τόσο στο δίκτυο όσο και στην εγγραφή δεδομένων στο ledger (permissionless DLTs). Τέλος, υπάρχουν και υβριδικές εφαρμογές που αποσκοπούν στη συγκέντρωση των πλεονεκτημάτων της κάθε περίπτωσης για την καλύτερη απόδοση του συστήματος. Στο Κεφάλαιο 1 έχει γίνει μια λεπτομερής παρουσίαση των κατηγοριών δικτύων blockchain, η οποία μπορεί να γενικευτεί και για τα DLTs.

Στη συνέχεια, έχοντας ήδη τονίσει το γεγονός ότι το blockchain αποτελεί έναν τύπο DLT, έχει ενδιαφέρον να γίνει αναφορά πάνω σε άλλες γνωστές λύσεις που αποτελούν υλοποιήσεις DLTs και να παρουσιαστεί ο τρόπος που λειτουργούν και οι διαφορές τους με το blockchain.

Έτσι, άλλοι τύποι γνωστών DLT συστημάτων πέρα από το blockchain, που έχει αναλυθεί διεξοδικά στο παρόν βιβλίο, αποτελούν:

- *Hashgraph*[125]: Η λύση της Hedera Hashgraph είναι μια σύγχρονη εναλλακτική προσέγγιση στην επίτευξη κατανεμημένης συναίνεσης με έναν τρόπο που διαφέρει από την προσέγγιση των λύσεων που βασίζονται στην τεχνολογία του blockchain. Αποτελεί έναν τύπο DLT ο οποίος χρησιμοποιεί ένα κατανεμημένο δίκτυο και αποθηκεύει δεδομένα με χρήση κρυπτογραφίας και χρονοσφραγίδων. Ο τρόπος όμως που το επιτυγχάνει αυτό διαφέρει.

  Έτσι, στη Hashgraph όλα τα δεδομένα αποθηκεύονται χωρίς να αναμένουν την ομαδοποίησή τους σε blocks και χωρίς τη δημιουργία κάποιας αλυσίδας. Κάθε κόμβος που συναλλάσσεται με κάποιον άλλον θα μεταφέρει σε αυτόν όλες τις πληροφορίες που γνωρίζει μέχρι εκείνη τη στιγμή. Έτσι, και αυτός σε μελλοντικές του συναλλαγές θα μεταφέρει και την πληροφορία που έμαθε σε προηγούμενες επικοινωνίες που είχε. Η διαδικασία αυτή μοιάζει με το κοινό «κουτσομπολιό» (gossip) και έτσι χαρακτηρίζεται για να διευκολύνει την κατανόησή της.

  Με τον τρόπο αυτόν όλοι οι χρήστες συμμετέχουν στη διαδικασία της συναίνεσης (και όχι μόνο οι miners ή minters) και τελικά όλοι οι κόμβοι γνωρίζουν τι έχει γίνει προηγουμένως στο δίκτυο. Έτσι επιτυγχάνεται η συναίνεση, χωρίς μάλιστα να υπάρχει λόγος για τη διεξαγωγή μιας ψηφοφορίας για αυτό. Θα πρέπει να τονιστεί ότι, λόγω της δυνατότητας συμμετοχής όλων των συναλλαγών στον γράφο καθώς και της έλλειψης των χρονικών περιορισμών που εισάγονται στο blockchain (π.χ. για την εφαρμογή του PoW), η λύση της Hashgraph ενσωματώνει πολλά μαθηματικά εργαλεία για να εξασφαλίσει την ολοένα και καλύτερη απόδοση του δικτύου τους.

[125] Online Σύνδεσμος: https://hedera.com/

- *Directed Acycle Graphs* (*DAGs*): Οι *Κατευθυνόμενοι Άκυκλοι Γράφοι* (DAGs) αποτελούν και αυτοί ένα άλλο είδος DLT, το οποίο επίσης καταφέρνει μια σημαντική ανάπτυξη πρόσφατα.
  Τα DLTs που βασίζονται σε DAGs δεν υποχρεώνουν τον κάθε κόμβο να αποθηκεύσει ολόκληρο το ledger, αλλά ζητούν για κάθε συναλλαγή ο κόμβος να επιβεβαιώσει δύο (τουλάχιστον) προηγούμενες συναλλαγές του δικτύου. Με τον τρόπο αυτό χτίζεται ένας κατευθυνόμενος γράφος στον οποίο στο ένα άκρο είναι οι καινούργιοι κόμβοι με τις συναλλαγές τους και στο άλλο άκρο οι παλαιότεροι. Κάθε νέος κόμβος θα εντάσσεται στο δίκτυο και το μονοπάτι που ξεκινά από αυτόν και διατρέχει τον γράφο προς έναν παλαιότερο κόμβο ολοένα και θα μεγαλώνει. Όσο πιο μεγάλο γίνει το μονοπάτι προς έναν κόμβο, τόσο πιο έγκυρη είναι η συναλλαγή που ολοκλήρωσε ο κόμβος αυτός.
  Αυτός ο τρόπος προσδίδει στο σύστημα περισσότερη επεκτασιμότητα, καθώς ο κάθε κόμβος δεν χρειάζεται να αποθηκεύει ολόκληρο τον γράφο παρά μόνο συγκεκριμένα μονοπάτια μέσα σε αυτόν, και έτσι μέλη του γράφου μπορούν να γίνουν και συσκευές περιορισμένης επεξεργαστικής ισχύος και χώρου αποθήκευσης.
  Η πιο γνωστή υλοποίηση DAG είναι αυτή του *IOTA*[126], που θα μελετηθεί με περισσότερη λεπτομέρεια στη συνέχεια, η οποία έχει κερδίσει σημαντικά οφέλη από την καταλληλότητά της για χρήση από συσκευές του *Διαδικτύου των Πραγμάτων* (*Internet of Things, IoT*).
- *Holochain*[127]: Πρόκειται για ένα framework ανάπτυξης κατανεμημένων εφαρμογών. Σε αυτό η κάθε εφαρμογή αποθηκεύει δεδομένα τοπικά στη μνήμη με συγκεκριμένο τρόπο, έτσι ώστε να υπάρχει μια σύνδεση μεταξύ τους σε μια αλυσίδα από hashes. Κατόπιν, με τη βοήθεια agents και χρησιμοποιώντας μια αρχιτεκτονική παρόμοια με τον τρόπο που λειτουργούν τα torrents, φροντίζει να υπάρχουν δεδομένα (τουλάχιστον τμήμα αυτών) σε όλους τους κόμβους, ενισχύοντας την ταχύτητα και την απόδοση των εφαρμογών.
  Σημαντικό ρόλο για την καλύτερη απόδοση έχει ότι η συγγραφή των δεδομένων στον «κοινό χώρο» πρέπει να ακολουθεί αυστηρά τις οδηγίες του πλαισίου, για να εξασφαλιστεί ότι δεν μεταφέρονται κακόβουλα αρχεία. Επιπλέον, η χρήση των hashes στη δημιουργία της αλυσίδας (μία για κάθε εφαρμογή) απαγορεύει τη διαγραφή δεδομένων από αυτήν επιτρέποντας μόνο τον χαρακτηρισμό τους ως διαγραμμένων, για να μην προβάλλονται από τις διεπαφές που τα απεικονίζουν.
  Ουσιαστικά, στο Holochain κάθε DApp δημιουργεί και υποστηρίζει το δικό του δίκτυο, κάνοντας τη διαχείρισή του πιο εύκολη για το σύστημα. Μάλιστα, σύμφωνα με μετρήσεις (Brock, 2021), η απόδοσή του απέναντι στο Ethereum είναι αρκετά ικανοποιητική. Δεν ισχύει αυτό όμως για όλων των ειδών τα αρχεία. Έτσι, λοιπόν, σύμφωνα με τον Bierling (2022), φαίνεται ότι η απόδοση του Holochain διατηρεί τα υψηλά επίπεδα που περιγράφηκαν όταν οι εφαρμογές ανταλλάσσουν μέτρια προς μικρά αρχεία. Στα μεγάλα αρχεία η απόδοση παύει να είναι το ίδιο καλή, οπότε συστήνεται η χρήση του σε λύσεις που αναδεικνύουν τις ικανότητές του.
- *Tempo* (*Radix*)[128]: Το Tempo (πρώην Radix) είναι ένα DLT το οποίο έχει στόχο να αυξήσει την επεκτασιμότητα και την απόδοση, ενισχύοντας τα χαρακτηριστικά της κατανεμημένης φύσης του.
  Έτσι, λοιπόν, το Tempo χρησιμοποιεί λογικά ρολόγια, με την απλοϊκή μορφή των καταμετρητών που δεν μειώνονται και οι οποίοι προσφέρουν ένα είδος χρονικής απόδειξης για τη διαχείριση μιας συναλλαγής από έναν κόμβο. Παράλληλα, ο κάθε κόμβος προσθέτει στις συναλλαγές που διαχειρίζεται ένα (εύκολα υπολογίσιμο) hash, που μπορεί να χρησιμοποιηθεί για την επιβεβαίωση της συμμετοχής του στην εξυπηρέτηση της συναλλαγής κατά τη διαλεύκανση αντιρρήσεων ως προς την ορθότητά της.
  Ταυτόχρονα, για καλύτερη απόδοση και αύξηση της επεκτασιμότητας, το ledger χωρίζεται από την αρχή σε έναν αριθμό από μικρότερα τμήματα (shards). Το κάθε shard ανατίθεται σε ένα πλήθος από κόμβους, ελαττώνοντας έτσι τον αριθμό των κόμβων μέσα σε κάθε shard που πρέπει να συναινέσουν για μια συναλλαγή.
  Επεκτείνοντας στο θέμα της συναίνεσης στο δίκτυο, το Radix επεξεργάστηκε έναν νέο αλγόριθμο, με το όνομα *Cerberus* (Casar, 2020), ο οποίος αναλαμβάνει να αντιμετωπίσει ορισμένα κενά του αρχικού αλγόριθμου συναίνεσης. Έτσι, λοιπόν, το Cerberus είναι ένας επεκτάσιμος pBFT αλγόριθμος συναίνεσης ο οποίος αντιμετωπίζει το πρόβλημα των πολλών μηνυμάτων για την απόκτηση συναίνεσης

[126] Online Σύνδεσμος: https://www.iota.org/
[127] Online Σύνδεσμος: https://www.holochain.org/
[128] Online Σύνδεσμος: https://www.radixdlt.com/

με μια παραλλαγή τεχνικής εύρεσης ενός αρχηγού. Στην παραλλαγή αυτή δεν ακολουθείται η συνήθης οδός της αντικατάστασης του αρχηγού όταν αυτός πάψει να είναι διαθέσιμος, αλλά προχωρά στην εύρεση καινούργιου για την έγκριση της κάθε συναλλαγής (σε κάθε shard). Με τον τρόπο αυτόν είναι δυνατόν να συνδεθούν μεταξύ τους οι αρχηγοί (και οι ψηφοφορίες τους), επιτρέποντας έτσι τον παραλληλισμό των συναλλαγών, αυξάνοντας ταυτόχρονα την απόδοση του δικτύου.
Σε δοκιμές για το σύστημα στις 12 Ιουνίου του 2019 χρησιμοποιήθηκε ως είσοδος στο σύστημα του Radix όλο το ledger του Bitcoin (10 χρόνια συναλλαγών). Το δίκτυο το αποτελούσαν 1.000 κόμβοι, οι οποίοι ανέλαβαν να επεξεργαστούν όλες τις συναλλαγές καθώς και να επιβεβαιώσουν τις ψηφιακές υπογραφές σε αυτές. Το αποτέλεσμα ήταν να μετρηθούν πάνω από 1 εκατομμύριο συναλλαγές το δευτερόλεπτο στο δίκτυο, δίνοντας μια σημαντική απόδειξη της παραλληλίας που προσφέρει η λύση αυτή.

Στη συνέχεια ακολουθεί μια βαθύτερη παρουσίαση των DAGs μέσω της εφαρμογής τους σε μια αρκετά διαδεδομένη εφαρμογή, το IOTA.

10.3 Λειτουργία των DLTs: Το παράδειγμα του IOTA

Μια από τις πιο σημαντικές υλοποιήσεις DLT είναι αυτή του IOTA. Το IOTA είναι ένα πρωτόκολλο μεταφοράς δεδομένων και αξίας, το οποίο είναι ανοικτό και δωρεάν. Ως σενάρια χρήσης του εμφανίζονται πολλά. Ανάμεσά τους αυτά που ξεχωρίζουν είναι: η εφαρμογή του στην ανάπτυξη ψηφιακής ταυτότητας, στη δημιουργία πορτοφολιών για οχήματα για την αποπληρωμή των εξόδων ηλεκτρικής φόρτισής τους, και για λήψη αποφάσεων σε αστικά κέντρα σχετικά με την εύρεση της συντομότερης διαδρομής.

Στη συνέχεια θα παρουσιαστούν αναλυτικά τα τεχνικά χαρακτηριστικά της λύσης και θα δοθεί έμφαση στον τρόπο που λειτουργεί το DAG, το οποίο ονομάζεται *Tangle* και βρίσκεται στον κορμό της λειτουργίας του IOTA.

10.3.1 IOTA: Ιστορία και χαρακτηριστικά του

Ένα κοινό λάθος στην αντίληψη για το IOTA αφορά τη σημασία και την προέλευση του ονόματός του. Συχνά, εσφαλμένα θεωρείται ότι αυτό σχετίζεται με το *Διαδίκτυο των Πραγμάτων* (*Internet of Things*, IoT), μιας και χρησιμοποιείται αρκετά σε λύσεις που ευδοκιμεί η εν λόγω τεχνολογία. Παρ' όλα αυτά, το IOTA δεν είναι τίποτα άλλο από το ελληνικό γράμμα γιώτα και σημαίνει κάτι πολύ μικρό.

Οι David Sønstebø, Sergey Ivancheglo, Dominik Schiener και Serguei Popov ίδρυσαν το IOTA το 2015, με την πρώτη ανοικτή προσφορά νομισμάτων (Initial Coin Offering, ICO) να γίνεται τον Νοέμβριο και τον Δεκέμβριο του 2015. Στη συνέχεια, η ομάδα του IOTA ιδρύει το ομώνυμο ίδρυμα (IOTA Foundation), μια μη κερδοσκοπική οργάνωση με έδρα το Βερολίνο. Σκοπός του ιδρύματος είναι η ανάπτυξη και προτυποποίηση καινούργιων λύσεων που βασίζονται στα DLTs.

Το κυρίως δίκτυο του IOTA (Main Net) τέθηκε σε λειτουργία τον Ιούλιο του 2016 και το 2018 δημοσιεύτηκε το whitepaper που περιγράφει το TANGLE, το DAG που βρίσκεται στην καρδιά του IOTA (Popov, 2018). Το Tangle αποθηκεύει το ledger και βρίσκεται σε κάθε κόμβο του δικτύου του IOTA.

Χαρακτηριστικό του δικτύου είναι ότι έχει το δικό του κρυπτονόμισμα (με την ονομασία IOTA) και ότι όλα τα κρυπτονομίσματα έχουν δημιουργηθεί αχρηστεύοντας την εφαρμογή της οποιαδήποτε διαδικασίας mining. Μάλιστα, σύμφωνα με υπολογισμούς, συνολικά θα υπάρχουν γύρω στα 32% περισσότερα IOTAs από ότι bitcoins το 2140.

Στα βασικά χαρακτηριστικά του συγκαταλέγονται (Mobelfish, 2018):

- *Η επεκτασιμότητα*: Σε αντίθεση με τις κλασικές λύσεις που βασίζονται στο blockchain, οι οποίες γίνονται ολοένα και πιο δύσχρηστες με την αύξηση του αριθμού των συναλλαγών (και κατ' επέκταση του αριθμού των blocks στην αλυσίδα), το DAG του IOTA γίνεται πιο ισχυρό και αποδοτικό όσο αυξάνεται ο αριθμός των συναλλαγών. Το επιχείρημα αυτό ενισχύθηκε ύστερα από δοκιμές (2017) που έδειξαν ότι σε ένα σχετικά μικρό δίκτυο από 250 κόμβους επιτεύχθηκε η επιβεβαίωση 112 συναλλαγών το δευτερόλεπτο. Σύμφωνα με το αποτέλεσμα αυτό, αναμένεται πως η απόδοση του δικτύου, μετρώμενη σε αριθμό επιβεβαιώσεων συναλλαγών, θα βελτιώνεται με την αύξηση του αριθμού των συναλλαγών.

- *Η αποκέντρωση*: Το IOTA, όπως αναφέρθηκε και προηγουμένως, δεν έχει miners. Κάθε δημιουργός μιας συναλλαγής (ή κόμβος) σε αυτό είναι ταυτόχρονα και χρήστης αλλά και επαληθευτής. Με τον τρόπο αυτόν όλοι συμμετέχουν στην επίτευξη συναίνεσης στο δίκτυο, επιτρέποντας στο δίκτυο την πλήρη αποκέντρωσή του.
- *Τα μηδενικά τέλη συναλλαγών*: Στο IOTA δεν υπάρχουν τέλη συναλλαγών, καθώς δεν υπάρχουν και miners για να τους αποδοθούν. Αυτόματα, αυτό σημαίνει ότι το IOTA μπορεί να χρησιμοποιηθεί για μικροσυναλλαγές[129], οι οποίες δεν έχουν νόημα να συμβούν στα άλλα δίκτυα blockchain (π.χ. Bitcoin, Ethereum), εφόσον τα τέλη για την πραγματοποίηση των συναλλαγών αυτών συχνά είναι μεγαλύτερα από το πραγματικό ποσό που μεταφέρεται.
- Στις συναλλαγές το μικρότερο ποσό που μεταφέρεται είναι ίσο με 1 IOTA (κατ' αντιστοιχία με το 1 wei στο Ethereum και το 1 satoshi στο Bitcoin).
- *Προστασία κβαντικών υπολογισμών* (*δεν υποστηρίζεται πλέον*): Αν και η χρήση των κβαντικών υπολογιστών δεν είναι ακόμη ευρεία, εκτιμάται πως με τον ερχομό τους θα είναι σε θέση να σπάσουν τους κώδικες που χρησιμοποιούνται στην κρυπτογραφία. Έτσι, το IOTA, προετοιμαζόμενο για να αντιμετωπίσει αυτόν τον κίνδυνο, εφαρμόζει το Σχήμα Ψηφιακών Υπογραφών Μιας Χρήσης του Winternitz, το οποίο και είναι ανθεκτικό στους κβαντικούς υπολογισμούς (Buchman et al., 2011).

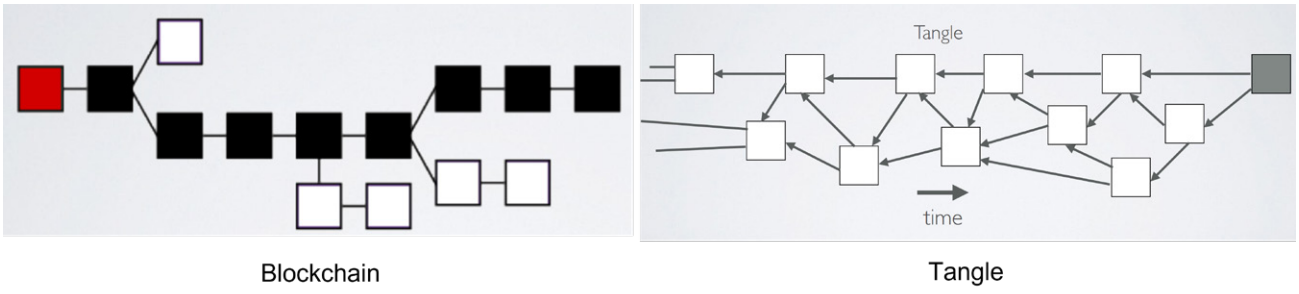
Η χρήση αυτών των υπογραφών όμως εγκαταλείφθηκε και στη θέση της χρησιμοποιείται άλλο σχήμα δημιουργίας υπογραφών (βασισμένο στην EdDSA, που θα παρουσιαστεί παρακάτω). Αυτό συνέβη λόγω αλλαγών στη φιλοσοφία του δικτύου καθώς και ανακατατάξεων στους συμμετέχοντες στο εγχείρημα. Με την αλλαγή αυτή αποφασίστηκε να δοθεί έμφαση στην καλύτερη ανάπτυξη για την υποστήριξη εφαρμογών σήμερα παρά επιμονή σε θέματα που μπορούν να αντιμετωπιστούν αργότερα.

10.3.2 IOTA Tangle: Χαρακτηριστικά

Με το IOTA να χαρακτηρίζεται ως 3ης γενιάς δημόσιο και permissionless DLT, ο τρόπος που καταγράφεται η πληροφορία στο ledger είναι πολύ διαφορετικός από την αλυσίδα από blocks που περιέχουν συναλλαγές που χρησιμοποιούνται σε ένα δίκτυο blockchain.

Στο IOTA οι κόμβοι που κάνουν συναλλαγές δημιουργούν μεταξύ τους συνδέσεις που παίζουν τον ρόλο των ακμών σε έναν *Άκυκλο Κατευθυνόμενο Γράφο* (*DAG*). Άκυκλος είναι ο γράφος γιατί οι νέοι κόμβοι που δημιουργούν συναλλαγή δεν μπορούν να επιβεβαιώσουν μια πολύ παλιά συναλλαγή, αλλά επιβεβαιώνουν μόνο νέες που αναμένουν επιβεβαίωση. Με αυτόν τον τρόπο ο γράφος αναπτύσσεται προς μία κατεύθυνση (για αυτό και Κατευθυνόμενος) και δεν δημιουργούνται κύκλοι (για αυτό και Άκυκλος).

Η δομή που βασίζεται σε DAGs ονομάζεται *Tangle* και φαίνεται στην Εικόνα 10.1 σε αντιστοιχία και με τη δομή του ledger στο blockchain, για λόγους σύγκρισης.



Εικόνα 10.1 *Η δομή του ledger στο blockchain με την αλυσίδα από blocks συναλλαγών (αριστερά) και η δομή του Tangle στο IOTA με χρήση DAGs (δεξιά) (Πηγή: Mobelfish, 2018).*

Στο Tangle κάθε τετράγωνο (κόμβος του γράφου) απεικονίζει μια συναλλαγή. Για να μπορέσει να εκτελεστεί η συναλλαγή αυτή, δηλαδή να προστεθεί ο κόμβος στην άκρη του Tangle, θα πρέπει να συνδεθεί με υπάρχοντες κόμβους. Έτσι, στο IOTA προβλέπεται ο κάθε νέος κόμβος να πρέπει να επιβεβαιώσει 2 προηγούμενες, μη

[129] Μικροσυναλλαγές είναι οι συναλλαγές που γίνονται για πολύ μικρά ποσά.

επιβεβαιωμένες συναλλαγές (κόμβους) για να εισαχθεί στο δίκτυο. Νεότερες εκδόσεις του δικτύου υποστηρίζουν έναν μεγαλύτερο αριθμό από μη επιβεβαιωμένες συναλλαγές, ανεβάζοντάς τον από 2 έως 8.

Στην Εικόνα 10.1 το γκρι τετράγωνο δεξιά απεικονίζει έναν νέο κόμβο, δηλαδή μια συναλλαγή που θέλει να ενταχθεί στον γράφο. Για να γίνει μέλος αυτού, θα πρέπει να συνδεθεί (δηλαδή επιβεβαιώσει) 2 (τουλάχιστον) από τις συναλλαγές (κόμβους) οι οποίες δεν έχουν κάποια εισερχόμενη ακμή. Δηλαδή δεν έχουν επιβεβαιωθεί ακόμη. Οι κόμβοι αυτοί ονομάζονται *κορυφές* (ή *tips*) στο Tangle. Παρ' όλα αυτά, οι κόμβοι κορυφές, ως μέλη του δικτύου, έχουν επιβεβαιώσει 2 (τουλάχιστον) προηγούμενες συναλλαγές αυτού. Οι συναλλαγές που επιβεβαιώνονται σταματούν να είναι κορυφές και πλέον μπαίνουν ένα βήμα πιο μέσα στον γράφο, καθώς νέες συναλλαγές προστίθενται (προς τα δεξιά στην Εικόνα 10.1) και αποκτούν τον ρόλο της κορυφής. Με τον τρόπο αυτό χτίζεται και η κατευθυντικότητα του γράφου στο Tangle.

Καθώς ο αριθμός των μη επιβεβαιωμένων συναλλαγών αναμένεται να είναι μεγαλύτερος από 2 (που απεικονίζεται στην Εικόνα 10.1), το ΙΟΤΑ τρέχει έναν αλγόριθμο για να επιλέξει μια κορυφή που αντιπροσωπεύει τις μη επιβεβαιωμένες συναλλαγές που θα επιβεβαιώσει για να ενταχθεί στον γράφο. Ο αλγόριθμος είναι ο *Markov Chain Monte Carlo* (*MCMC*) και η επιλογή του κόμβου που παράγεται ως αποτέλεσμα της εκτέλεσης του αλγόριθμου δεν εξαρτάται από προηγούμενες επιλογές αλλά ακολουθεί τον κανόνα που έχει συμφωνηθεί εξαρχής (Speagle, 2019).

Άλλα χαρακτηριστικά του IOTA Tangle σχετίζονται με ιδιότητες του γράφου που δημιουργείται. Έτσι, λοιπόν, υπάρχει το *ύψος*, που ορίζεται ως το μήκος (σε αριθμό από κόμβους) της μεγαλύτερης προσανατολισμένης διαδρομής από έναν κόμβο (αρχή) μέχρι τον πρώτο κόμβο (genesis) του γράφου (πέρας). Τέλος, υπάρχει και η έννοια του *βάθους*, το οποίο ορίζεται ως το μήκος της μεγαλύτερης αντίστροφης διαδρομής από έναν κόμβο προς κάποιον άλλο συγκεκριμένο.

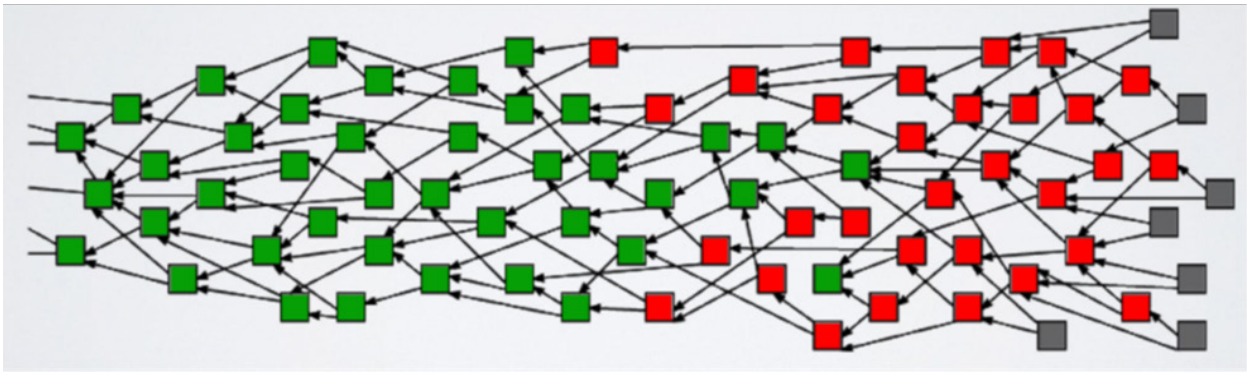
Θα πρέπει να τονιστεί ότι η δομή του Tangle επιτρέπει την παράλληλη ανάπτυξή του, καθώς δεν μαζεύονται όλες οι συναλλαγές (κόμβοι) στο ίδιο σημείο, όπως σε ένα σύστημα blockchain, όπου μπαίνουν σε σειρά στην αλυσίδα. Επομένως και αυτό το χαρακτηριστικό ενισχύει την αυξημένη επεκτασιμότητα που παρουσιάζει το ΙΟΤΑ σε σύγκριση με τις λύσεις που βασίζονται στο blockchain.

10.3.3 Δημιουργία, εκτέλεση και επιβεβαίωση συναλλαγών

Για τη δημιουργία και εκτέλεση μιας συναλλαγής ακολουθούνται τα εξής πέντε απλοϊκά βήματα (IOTA, 2022):

1) *Δημιουργία συναλλαγής*: Συμπληρώνεται η δομή της συναλλαγής με στοιχεία όπως: τη διεύθυνση προορισμού και αποστολέα, την τιμή που μεταφέρεται (μπορεί να είναι και μηδενική) και, προαιρετικά, το όποιο μήνυμα συμπεριλαμβάνεται σε αυτήν.
2) *Υπογραφή*: Με τη βοήθεια του ιδιωτικού κλειδιού η συναλλαγή υπογράφεται, για να εξασφαλιστεί ότι έχει δημιουργηθεί αποκλειστικά και μόνο από τον αποστολέα.
3) *Επιλογή κορυφής*: Γίνεται επιλογή από 2 (έως 8) μη επιβεβαιωμένων συναλλαγών για επιβεβαίωση.
4) *Proof of Work*: Οι κόμβοι στη συνέχεια θα εκτελέσουν τον αλγόριθμο του PoW αλλά όχι για να έρθουν σε συναίνεση, απλώς ως μέτρο αποφυγής επιθέσεων spam (δηλαδή αλόγιστης δημιουργίας μηνυμάτων από έναν κόμβο). Έτσι θα παραχθεί ένα nonce για την επιτυχή ολοκλήρωση της διαδικασίας. Επιπλέον, οι συναλλαγές που επιβεβαιώνονται θα ελεγχθούν ως προς την ορθότητά τους και ως προς το αν δημιουργείται κάποια σύγκρουση από την εκτέλεσή τους.
5) *Εκτέλεση*: Η ολοκληρωμένη συναλλαγή αποστέλλεται σε έναν κόμβο που αναλαμβάνει να την προωθήσει στους γειτονικούς του και να φθάσει σε ολόκληρο το δίκτυο. Η συναλλαγή πλέον είναι διαθέσιμη να συμπεριληφθεί στο βήμα 3 (Επιλογή Κορυφής) μιας νεότερης συναλλαγής, που θα έρθει να μπει και αυτή στο δίκτυο.

Για την επιβεβαίωση των συναλλαγών θα πρέπει να γίνει κατανοητή η κατάσταση στην οποία αυτές βρίσκονται ανά πάσα χρονική στιγμή μέσα στο Tangle. Έτσι, στην **Εικόνα 10.2** απεικονίζονται χρωματικά οι διαφορετικές καταστάσεις στις οποίες μπορεί να συναντήσει κάποιος μια συναλλαγή στο Tangle.



Εικόνα 10.2 *Οι καταστάσεις που βρίσκονται οι συναλλαγές στο Tangle του IOTA (Πηγή: Mobelfish, 2018).*

Συγκεκριμένα, στην Εικόνα 10.2 διακρίνονται τρεις καταστάσεις. Με:

- *Πράσινο χρώμα*: Διακρίνονται οι συναλλαγές που έχουν επιβεβαιωθεί και για τις οποίες έχει επιτευχθεί συναίνεση στο δίκτυο.
- *Κόκκινο χρώμα*: Πρόκειται για συναλλαγές που έχουν δεχθεί κάποιες επιβεβαιώσεις αλλά παραμένει μια αβεβαιότητα ως προς την εγκυρότητά τους και αναμένεται η πλήρης επιβεβαίωσή τους.
- *Γκρι χρώμα*: Πρόκειται για κορυφές (tips), δηλαδή για μη επιβεβαιωμένες συναλλαγές.

Σκοπός όλων των συναλλαγών είναι να αποκτήσουν πράσινο χρώμα και να είναι συναινετικά επιβεβαιωμένες. Για να επιτευχθεί αυτό, θα πρέπει να υπάρχει ένα μονοπάτι από κάθε γκρι συναλλαγή προς την επί μελέτη κόκκινη συναλλαγή. Η παρουσία μονοπατιού από την άκρη του δικτύου προς κάποιο ενδιάμεσο σημείο αυτού σηματοδοτεί την (έμμεση) επιβεβαίωση της εγκυρότητας της συναλλαγής αυτής, καθώς θα ανήκει σε ένα μονοπάτι επιβεβαιωμένων συναλλαγών. Επομένως, σε όλες οι συναλλαγές, καθώς προστίθενται νέες, θα αυξάνεται η πιθανότητα σύντομα να μπορούν να απεικονιστούν με το πράσινο χρώμα. Έτσι, λοιπόν, ο γράφος θα χρωματίζεται πράσινος με την πάροδο του χρόνου, ενώ θα διατηρεί κόκκινο και γκρι χρώμα στις άκρες του με την προσθήκη νέων συναλλαγών.

Μάλιστα, το επίπεδο επιβεβαίωσης μιας συναλλαγής μπορεί να μετρηθεί και χρησιμοποιείται ως ένα μέτρο που υποδηλώνει και το χρώμα αυτής ή τις μεταβάσεις του. Αυτό μετριέται αν εκτελεστεί N φορές, ξεκινώντας από την υπό μελέτη συναλλαγή, ο αλγόριθμος επιλογής κορυφής. Κάθε φορά θα μετριέται αν η κορυφή που επιλέγει ο αλγόριθμος έχει μονοπάτι προς τη συναλλαγή αυτή ή όχι. Στην ουσία υπολογίζεται αν υπάρχει το βάθος της συναλλαγής προς την εκάστοτε επιλεγμένη κορυφή. Αν ο τελικός αριθμός που επιτυγχάνεται αντίστροφο μονοπάτι είναι M, τότε το επίπεδο επιβεβαίωσης ισούται με M/N.

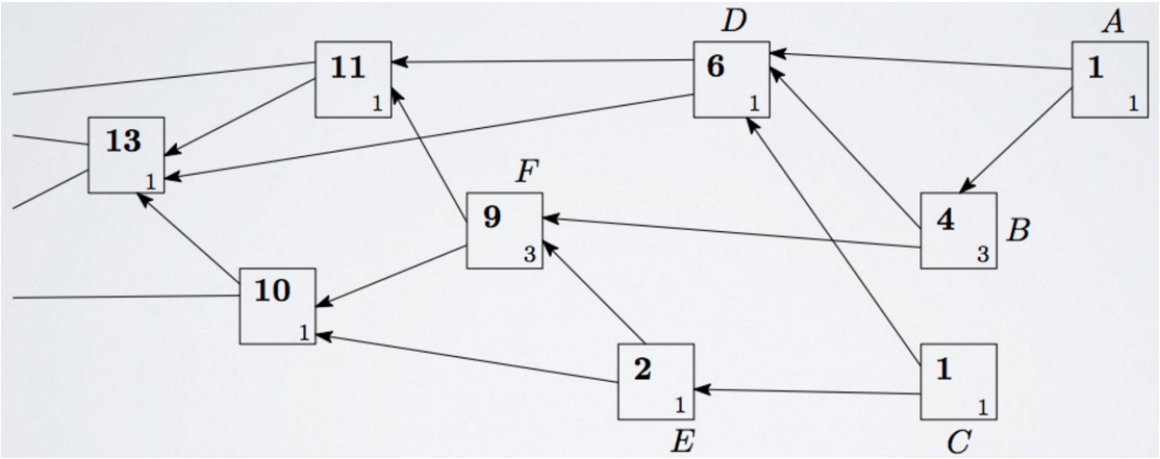
Επιπλέον, το αποτέλεσμα αυτό μπορεί να χρησιμοποιηθεί από χρήστη για να κρίνει αν θα ολοκληρώσει μια συναλλαγή ή θα περιμένει να αυξηθεί η απόδοση της συναλλαγής. Παρόμοια, δηλαδή, με την αναμονή ενός αριθμού επιβεβαιώσεων σε ένα δίκτυο blockchain προτού να θεωρηθεί ασφαλής η ολοκλήρωση της συναλλαγής. Βέβαια, όσο πιο μεγάλο το βάθος μιας συναλλαγής, τόσο μεγαλύτερος και ο χρόνος που χρειάζεται για την πλήρη επιβεβαίωσή της.

10.3.4 Ατομικό και συσσωρευμένο βάρος συναλλαγής στο Tangle

Στο IOTA η έννοια του βάρους χρησιμοποιείται αρκετά και συνδέεται με την κάθε συναλλαγή. Έτσι, υπάρχουν δύο είδη βαρών που θα παρουσιαστούν: τα ατομικά βάρη της κάθε συναλλαγής και τα συσσωρευμένα βάρη αυτών.

Ξεκινώντας από το ατομικό βάρος, αυτό αρχικοποιείται σε κάθε συναλλαγή ανάλογα με το πόσο δύσκολο είναι το PoW που πρέπει να υλοποιήσει ο κόμβος για τη συναλλαγή. Οι τιμές που μπορεί να πάρει το ατομικό βάρος είναι πολλαπλάσιες του αριθμού 3. Το συσσωρευμένο βάρος μιας συναλλαγής είναι το άθροισμα των ατομικών βαρών της υπό μελέτη συναλλαγής συν τα ατομικά βάρη όλων των συναλλαγών που επιβεβαιώνουν την αρχική είτε έμμεσα (μέσω μονοπατιού) είτε άμεσα (απευθείας).

Στην **Εικόνα 10.3** φαίνεται ένα στιγμιότυπο του Tangle το οποίο περιλαμβάνει και τις πληροφορίες σχετικά με το ατομικό και το συσσωρευμένο βάρος μιας συναλλαγής. Ο μικρός αριθμός κάτω δεξιά είναι το ατομικό βάρος της συναλλαγής, ενώ ο μεγαλύτερος αριθμός πάνω αριστερά είναι το συσσωρευμένο βάρος.



Εικόνα 10.3 *Παράδειγμα υπολογισμού συσσωρευμένου βάρους στο Tangle του IOTA (Πηγή: Mobelfish, 2018).*

Χρησιμοποιώντας την Εικόνα 10.3 ως βάση, θα εξηγηθεί η τιμή του συσσωρευμένου βάρους μιας συναλλαγής. Έτσι, η τιμή της F είναι ίση με 9. Για την επιβεβαίωσή της θα πρέπει να αναφερθούν όλες οι συναλλαγές που έχουν ένα μονοπάτι που περνά από την F. Κατόπιν θα αθροιστούν τα ατομικά βάρη των συναλλαγών αυτών. Τέτοιες συναλλαγές είναι οι: A, B, C και E. Επομένως, το συσσωρευμένο βάρος της F είναι, κατά αντιστοιχία, ίσο με 3 (της F) + 1 (της A) + 3 (της B) + 1 (της C) + 1 (της E) = 9.

Σε ένα ακόμα παράδειγμα, το συσσωρευμένο βάρος της E είναι ίσο μόνο με 2, γιατί περιλαμβάνει τα ατομικά βάρη των E και C μόνο (1 + 1).

Επιπλέον, το συσσωρευμένο βάρος αποτελεί και ένα σημαντικό μέτρο για κάθε συναλλαγή, καθώς μια μεγάλη τιμή υποδηλώνει μια «δημοφιλή» συναλλαγή. Ακόμα, με κάθε νέα συναλλαγή που εντάσσεται στο δίκτυο ανανεώνονται οι τιμές όλων των συσσωρευμένων βαρών που επηρεάζονται από την ένταξή της στο Tangle. Ο σκοπός χρήσης του είναι και για την αποφυγή κακόβουλων συναλλαγών, καθώς αναμένεται πως καμία τέτοια συναλλαγή δεν θα αποκτήσει μεγάλη τιμή στο συσσωρευμένο βάρος της.

10.3.5 Τύποι διευθύνσεων στο IOTA

Για τη δημιουργία διευθύνσεων στο IOTA χρησιμοποιείται παρόμοια τεχνική με αυτήν που εφαρμόζεται στα δίκτυα blockchain. Έτσι, γίνεται χρήση ασύμμετρης κρυπτογραφίας, και ιδιαίτερα ενός ζεύγους δημόσιου/ιδιωτικού κλειδιού που συνδυάζεται με την παραγωγή ενός seed από το οποίο ξεκινούν όλα. Επομένως, κάθε διεύθυνση αντιστοιχεί μοναδικά σε ένα ιδιωτικό κλειδί το οποίο χρησιμοποιείται ως απόδειξη ιδιοκτησίας, όπου αυτό χρειάζεται.

Ενώ παλαιότερα το IOTA χρησιμοποιούσε δύο είδη διευθύνσεων, από τον Αύγουστο του 2020, που ξεκίνησε η πρώτη φάση της νέας έκδοσης του δικτύου (με όνομα Chrysalis), χρησιμοποιεί μόνο έναν τύπο, με ονομασία *Ed25519*. Μάλιστα, η διεύθυνση υπολογίζεται ως το hash που παράγει ο αλγόριθμος *BLAKE2b-256* αν δοθεί ως είσοδος σε αυτόν το hash που παράγεται από τον αλγόριθμο Ed25519 όταν αυτός τροφοδοτηθεί με είσοδο το δημόσιο κλειδί του χρήστη.

Η Ed25519 είναι ένα σύγχρονο σχήμα κρυπτογράφησης και δημιουργίας ψηφιακών υπογραφών που βασίζεται στην EdDSA (Bernstein, 2012), παρόμοιο με την ECDSA, η οποία χρησιμοποιεί τον SHA-512 και τη συνάρτηση Curve25519 (Bernstein, 2006). Τα πλεονεκτήματα της EdDSA είναι ότι υποστηρίζει μικρότερο μέγεθος μηνυμάτων και τη δυνατότητα διαχείρισης ποσών (στη μορφή του token του IOTA στην περίπτωση εδώ) και την ασφαλή μεταφορά τους, εφόσον αυτά έχουν αποθηκευτεί σε ασφαλές χρηματοκιβώτιο.

Παρακάτω βλέπετε ένα παράδειγμα μιας διεύθυνσης στο IOTA που είναι τύπου Ed25519:

iota1q9f0mlq8yxpx2nck8a0slxnzr4ef2ek8f5gqxlzd0wasgp73utryj0w6qwt

Η διεύθυνση αυτή βασίζεται στην επέκταση Bech32, που ξεκίνησε από το Bitcoin και έχει αναδυθεί ως μια σταθερά στην απόδοση διευθύνσεων που υλοποιείται από πολλές εφαρμογές πορτοφολιών.

Ένα string που ακολουθεί την επέκταση Bech32 αποτελείται από ένα σύνολο το πολύ 90 χαρακτήρων και απαρτίζεται από τα εξής μέρη:

- Ένα μέρος που είναι εύκολο να διαβαστεί από ανθρώπους και αφορά την απεικόνιση του πρωτοκόλλου του IOTA και τον διαχωρισμό μεταξύ του κυρίως δικτύου (Main Net) και του δοκιμαστικού δικτύου. Οι διευθύνσεις που αφορούν το Main Net συμπεριλαμβάνουν τη λέξη *iot*, ενώ αυτές που αφορούν το δοκιμαστικό δίκτυο τη λέξη *tio* στην αρχή τους. Επομένως, η διεύθυνση που δόθηκε παραπάνω αφορά μια διεύθυνση στο κυρίως δίκτυο του IOTA, όπως φαίνεται με κόκκινο χρώμα.
- Υπάρχει ένας χαρακτήρας διαχωρισμού, που στην περίπτωση του IOTA είναι ο αριθμός 1 (φαίνεται με μπλε χρώμα στο παράδειγμα παραπάνω).
- Το υπόλοιπο είναι το τμήμα των δεδομένων.

10.3.6 Τύποι μηνυμάτων/συναλλαγών

Τα μηνύματα (ή συναλλαγές) που μεταφέρονται στο δίκτυο του IOTA αποτελούν τους κόμβους του Tangle και συχνά περιέχουν και δεδομένα. Τα δεδομένα αυτά τα επεξεργάζονται όλοι οι κόμβοι εφόσον μεταφέρουν δεδομένα του πρωτοκόλλου. Άλλα αφορούν εφαρμογές που τρέχουν στο δίκτυο και άλλα περιέχουν εμφωλιασμένα δεδομένα.

Επιπλέον, όσον αφορά τη μορφή των συναλλαγών, το IOTA σταμάτησε να υποστηρίζει τη χρήση λογαριασμών (σε αντιστοιχία με το Ethereum) και υιοθέτησε τη λογική του Bitcoin (βλ. Κεφάλαιο 4) με τα Unspent Transcaction Outputs (UTXOs). Ο λόγος είναι ότι κρίθηκε πως ο τρόπος αυτός είναι πιο γρήγορος από τη συλλογή όλων των συναλλαγών σε μια ομάδα και την αποστολή τους ως μία μονάδα, που είχε υιοθετηθεί προηγουμένως.

10.3.7 Ο Συντονιστής (Coordinator)

Πολλοί από τους επικριτές του IOTA υποστηρίζουν ότι αυτό δεν είναι πλήρως αποκεντρωμένο λόγω της ύπαρξης του *Συντονιστή* (*Coordinator*) σε αυτό. Ο Συντονιστής αποτελείται από ένα σύνολο από πλήρεις κόμβους οι οποίοι αναλαμβάνουν να στέλνουν περιοδικά ειδικές συναλλαγές στο δίκτυο για να βοηθήσουν στη σωστή λειτουργία του. Προς απάντηση των επικριτών, οι συναλλαγές του Συντονιστή ελέγχονται από τους κόμβους ως προς την ορθότητά τους.

Πιο αναλυτικά, κάθε δίκτυο IOTA έχει τον δικό του Συντονιστή, ο οποίος είναι υπεύθυνος για να στέλνει ειδικά μηνύματα που περιέχουν την υπογραφή του ανά τακτά χρονικά διαστήματα (περίπου ανά δευτερόλεπτο). Τα μηνύματα αυτά ονομάζονται *milestones* και περιλαμβάνουν τις συναλλαγές που εκκρεμούν προς επιβεβαίωση. Καμία συναλλαγή δεν μπορεί να δεχθεί επιβεβαίωση αν δεν έχει γίνει αναφορά σε αυτήν, έμμεσα ή άμεσα, από τον Συντονιστή.

Ο κώδικας δημιουργίας του Συντονιστή δεν είναι ελεύθερος αλλά βρίσκεται προεγκατεστημένος μέσα στον κώδικα του δικτύου. Επιπλέον, οι κόμβοι στο δίκτυο γνωρίζουν τη διεύθυνση του Συντονιστή, επομένως μπορούν εύκολα να επιβεβαιώσουν από την υπογραφή του ότι τα μηνύματα (οι συναλλαγές) που έχει δημιουργήσει αυτός προέρχονται πράγματι από αυτόν. Μιας και στα μηνύματα (συναλλαγές) του Συντονιστή περιέχονται και οι προς επιβεβαίωση κορυφές (tips), για να μπορεί να γίνει ένας συγχρονισμός του δικτύου μέσω αυτών, τα μηνύματα του Συντονιστή είναι αριθμημένα. Κάθε νέο μήνυμα του Συντονιστή ανανεώνει αυτή τη λίστα.

Όπως γίνεται αντιληπτό, ο ρόλος του Συντονιστή είναι πολύ επιδραστικός στο δίκτυο και για αυτό και υπήρχαν τα επιχειρήματα περί έλλειψης αποκέντρωσης εξαιτίας αυτού. Το ίδρυμα του IOTA, υπεύθυνο για την ανάπτυξη και εκτέλεση του πρωτοκόλλου, έχει τονίσει ότι η παρουσία του Συντονιστή ήταν ένα επιβεβλημένο βήμα για την περίοδο που το δίκτυο βρισκόταν στην πρώτη ανάπτυξή του. Έχει τονιστεί ότι, όταν φθάσει το δίκτυο σε θέση να επιτύχει έναν μεγάλο αριθμό συναλλαγών το δευτερόλεπτο, τότε θα μπορεί να αφαιρεθεί ο Συντονιστής και να αφεθεί το δίκτυο να αναπτυχθεί μόνο του.

Η κατάσταση του δικτύου χωρίς τον Συντονιστή είναι η έκδοση 2.0 του πρωτοκόλλου, και η εργασία για την κατάσταση έχει ξεκινήσει. Μάλιστα, το κωδικό όνομα που έχει δοθεί για την έκδοση 2.0 είναι *Coordicide* και είναι συνδυασμός των αγγλικών λέξεων *Coordinator* + *Suicide*. Πρόκειται, δηλαδή, για την έκδοση στην οποία «φεύγει» ο Συντονιστής.

Τέλος, η μετάβαση στη νέα αυτή εποχή χωρίς τον Συντονιστή έχει ξεκινήσει. Το μεταβατικό δίκτυο, με το κωδικό όνομα *Chrysallis*, είναι ενεργό και αποτελεί την έκδοση 1.5 του πρωτοκόλλου. Το πρώτο στάδιο του δικτύου ενεργοποιήθηκε τον Αύγουστο του 2020 και όλα τα χαρακτηριστικά του δικτύου Chrysallis προστέθηκαν τον Απρίλιο του 2021. Επιπρόσθετα, στο Chrysallis έχουν προστεθεί κάποια χαρακτηριστικά του τελικού δικτύου (π.χ. η αυτόματη εύρεση κόμβων για τη δημιουργία συνδέσεων στο δίκτυο).

Άλλα σημαντικά χαρακτηριστικά που προστίθενται στο IOTA είναι και η δυνατότητα συγγραφής και εγκατάστασης smart contracts, κάτι που προηγουμένως δεν ήταν δυνατόν. Αυτό γίνεται μέσω του IOTA Smart Contract Protocol, σύμφωνα με το οποίο είναι δυνατόν να εγκατασταθούν smart contracts πάνω από το Tangle, επιτρέποντας στον καθένα να δημιουργήσει ένα δίκτυο blockchain που υποστηρίζεται από ειδικούς κόμβους που τρέχουν το λογισμικό WASP. Οι κόμβοι αυτοί επιτρέπουν την εγκατάσταση smart contract και, επίσης, την επιβεβαίωση των συναλλαγών σε αυτό. Όταν επέλθει επιβεβαίωση, οι κόμβοι στέλνουν την αλλαγή κατάστασης στο Tangle για αποθήκευση. Για περισσότερες πληροφορίες πάνω στο θέμα μπορείτε να δείτε στο IOTA Wiki (2022).

10.3.8 Στιγμιότυπα (snapshots)

Αρχικά στο IOTA είχε αποφασιστεί να λαμβάνονται σε σταθερές χρονικές περιόδους στιγμιότυπα του δικτύου. Τα στιγμιότυπα αυτά θα επέτρεπαν την αποφυγή της δημιουργίας μιας μεγάλης Βάσης Δεδομένων για την αποθήκευση των συναλλαγών, καθώς αυτή θα έσβηνε με το πέρας του κάθε στιγμιότυπου. Μάλιστα, για τη μεταφορά της κατάστασης στους λογαριασμούς των χρηστών από το ένα στιγμιότυπο στο άλλο γινόταν μια περίληψη του υπολοίπου στους λογαριασμούς και όσοι δεν είχαν μηδενικό υπόλοιπο μεταφέρονταν ως αρχική κατάσταση στο νέο στιγμιότυπο.

Η κατάσταση αυτή είχε τα πλεονεκτήματα αλλά και τα μειονεκτήματά της. Σημαντικό ήταν και το κομμάτι της προετοιμασίας για το επόμενο snapshot έτσι ώστε να μεταφερθούν όλα τα αποτελέσματα των ενεργειών του προηγούμενου. Επίσης, με την εφαρμογή των snapshots θα πρέπει να υπάρχουν και κάποιοι κόμβοι οι οποίοι θα έχουν όλες τις συναλλαγές σε όλα τα snapshots, έτσι ώστε να μπορέσει όποιος επιθυμεί να ανατρέξει στο ιστορικό τους.

Τα στιγμιότυπα έπαψαν να λειτουργούν με τη μετάβαση στο Chrysallis, όπου το δίκτυο σταμάτησε να λειτουργεί με τη μορφή ανανέωσης του υπολοίπου σε λογαριασμούς, αλλά προέβη στη χρήση UTXOs για τις συναλλαγές.

10.4 Περιπτώσεις χρήσης των DLTs

Έχοντας μελετήσει τα ιδιαίτερα χαρακτηριστικά των DLTs, και σε συνέχεια του Κεφαλαίου 9, όπου παρουσιάστηκαν οι περιπτώσεις χρήσης που αφορούσαν την τεχνολογία του blockchain, η οποία, όπως αναφέρθηκε, αποτελεί ένα είδος από DLT, στην ενότητα αυτή θα γίνει μια ευρύτερη παρουσίαση των περιπτώσεων χρήσης για τα DLTs.

Έτσι, ενδεικτικά, ορισμένα παραδείγματα περιπτώσεων χρήσης στα οποία τα DLTs μπορεί να είναι αποδοτικά περιλαμβάνουν:

- *Την εφαρμογή των smart contracts σε βιομηχανικές διεργασίες*: Η δημιουργία και εκτέλεση smart contracts είναι άμεσα συνδεδεμένη με τα DLTs και η χρήση τους στη βιομηχανία αναμένεται να αυξηθεί ραγδαία τα επόμενα χρόνια, όσο και η τεχνολογία ωριμάζει περισσότερο. Παραδείγματα χρήσης στον τομέα αυτό περιλαμβάνουν: την παρακολούθηση των συνθηκών διασφάλισης ποιότητας σε κάθε στάδιο ανάπτυξης ενός υλικού και τη δημιουργία τιμολογίων και προϋποθέσεων για διακανονισμό.
- *Την αποφυγή κλοπής της ψηφιακής ταυτότητας*: Η δυνατότητα συνδυασμού των τεχνολογιών DLTs με την τεχνολογία του IoT αναμένεται να προσφέρει τα απαραίτητα εργαλεία για την εύρεση λύσεων που επιτυγχάνουν τη δυνατότητα υποστήριξης ψηφιακών ταυτοτήτων οι οποίες είναι μοναδικές και δεν μεταφέρονται. Με τον τρόπο αυτό θα είναι, επίσης, δύσκολο σε κάποιον να τις κλέψει ή να τις δανειστεί.
- *Τη διαχείριση της τροφικής αλυσίδας:* Τα DLTs μπορούν να αυτοματοποιήσουν πολλές διαδικασίες διαχείρισης και να αποτελέσουν μια κοινή γλώσσα για τις διάφορες λύσεις που εφαρμόζονται στα διαφορετικά στάδια κάθε τροφικής αλυσίδας.
- *Τις υπηρεσίες υγείας*: Δίνεται η ευκαιρία για τη δημιουργία μιας κοινής βάσης για τα δεδομένα των ασθενών, με τα οποία με την κατάλληλη άδεια θα μπορεί να παρέχεται πρόσβαση σε αυτά σε τρίτα μέρη που χρειάζεται να επέμβουν (π.χ. γιατρός, νοσοκομείο).
- *Τη διαχείριση της ενέργειας*: Με τη βοήθεια των DLTs θα είναι εύκολη και άμεση η καταμέτρηση και πληρωμή των καταναλώσεων των χρηστών. Επιπλέον, θα είναι δυνατή η έμπιστη διαχείριση της καταγραφής της ενέργειας που παράγεται από χρήστες και που προστίθεται στο κεντρικό δίκτυο, ενισχύοντας τη συμμετοχή τους στο έργο αυτό.

- *Τη διαχείριση εκλογικών διαδικασιών*: Η δυνατότητα διατήρησης της ανωνυμίας και η εγκυρότητα της ψήφου είναι βασικά χαρακτηριστικά που πρέπει να έχει ένα εκλογικό σύστημα. Αυτά όμως είναι και χαρακτηριστικά που μπορεί να συναντήσει κανείς σε ένα DLT σύστημα. Οι προσπάθειες έχουν ήδη ξεκινήσει για την υλοποίηση ενός κατανεμημένου συστήματος διαχείρισης εκλογικών διαδικασιών και αναμένονται ελπιδοφόρα αποτελέσματα στο άμεσο μέλλον.

Βιβλιογραφία

Bierling, M. (2022). *Introduction to Holochain, A Post-Blockchain Crypto Technology*, Unblock, Online Πηγή: https://unblock.net/introduction-holochain/ Τελευταία πρόσβαση: Δεκέμβριος 2022.

Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. In: M. Yung, Y. Dodis, A. Kiayias & T. Malkin (Eds.), *Public Key Cryptography – PKC 2006. PKC 2006. Lecture Notes in Computer Science*, vol. 3958, 2006. Springer, Berlin, Heidelberg. Online πηγή: https://link.springer.com/chapter/10.1007/11745853_14

Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., & Yang, B. Y. (2012). High-speed high-security signatures. *Journal of cryptographic engineering*, *2*(2), 2012, pp. 77-89.

Brock, A. (2021). *Limits to Blockchain scalability vs. Holochain Medium platform*, June 2021. Online πηγή: https://artbrock.medium.com/limits-to-blockchain-scalability-vs-holochain-19685dcb89f9 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Buchmann, J., Dahmen, E., Ereth, S., Hülsing, A., & Rückert, M. (2011). On the security of the Winternitz one-time signature scheme. In: *Proceedings of the 4th international conference on Progress in cryptology in Africa (AFRICACRYPT'11)*. Springer-Verlag, Berlin, Heidelberg, 2011, pp. 363-378.

Casar, F., Hughes, D., Primero, J., & Thornton, S. (2020). Cerberus, a parallelized BFT Consensus Protocol for Radix. *White Paper*. March 2020. Online πηγή: https://assets.website-files.com/6053f7fca5bf627283b582c2/608811e3f5d21f235392fee1_Cerberus-Whitepaper-v1.01.pdf [Τελευταία πρόσβαση: Δεκέμβριος 2022].

IOTA (2022). *IOTA Introduction*. Online πηγή: https://legacy.docs.iota.works/docs/getting-started/1.5/introduction/overview [Τελευταία πρόσβαση: Δεκέμβριος 2022].

IOTA Wiki (2022). *IOTA Smart Contracts*. Online πηγή: https://wiki.iota.org/shimmer/smart-contracts/overview [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Mobelfish (2018). *IOTA Tutorial*, Online πηγή: https://www.mobilefish.com/developer/iota/iota_quickguide_tutorial.html [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Popov, S. (2018). *The Tangle–Version 1.4.3. Whitepaper*. Online πηγή: https://www.semanticscholar.org/paper/The-Tangle-Popov/43586b34b054b48891d478407d4e7435702653e0 [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Radix Blog (2020). *Tempo – Consensus Lessons Learned.* Online πηγή: https://www.radixdlt.com/post/tempo-consensus-lessons-learned [Τελευταία πρόσβαση: Δεκέμβριος 2022].

Speagle, J. S. (2019). *A conceptual introduction to markov chain monte carlo methods, arXiv preprint arXiv:1909.12313*.

ΚΕΦΑΛΑΙΟ 11
Πρακτικά Παραδείγματα

Σύνοψη

Το κεφάλαιο αυτό εστιάζει στην πρακτική παρουσίαση θεμάτων που σχετίζονται με χαρακτηριστικά της τεχνολογίας του blockchain, τα οποία καλύφθηκαν σε προηγούμενα κεφάλαια του βιβλίου θεωρητικά. Τα θέματα που θα συζητηθούν έχουν ως βάση το δίκτυο του Ethereum και περιλαμβάνουν μεταξύ τους έννοιες όπως την ασύμμετρη κρυπτογραφία, τη δημιουργία διευθύνσεων σε ένα δίκτυο blockchain και την υπογραφή μηνυμάτων.

Για τον λόγο αυτό, θα χρησιμοποιηθεί ένα δωρεάν εκπαιδευτικό εργαλείο, το ETH.build, το οποίο προτείνεται από το Ethereum για την εξοικείωση του κόσμου με τις έννοιες του blockchain.

Προαπαιτούμενη γνώση

Ανάγνωση των Κεφαλαίων 1, 2, 3, 4 και 5.

11.1 Το ETH.Build ως εργαλείο εκμάθησης

Καθώς η τεχνολογία του blockchain έχει καταφέρει να τραβήξει το ενδιαφέρον ολοένα και περισσότερου κόσμου, η ανάγκη για να γίνει κατανοητή πλήρως από όλους όσοι τη γνωρίζουν για πρώτη φορά μεγαλώνει σημαντικά. Για τον λόγο αυτόν η ανάπτυξη εκπαιδευτικών εργαλείων που βοηθά στην εκμάθηση των χαρακτηριστικών της τεχνολογίας του blockchain έχει σημειώσει σημαντικά βήματα τα τελευταία χρόνια.

Ιδιαίτερα το Ethereum υποστηρίζει ένα αξιοπρόσεκτο σύνολο από εργαλεία εκμάθησης μέσα από την ιστοσελίδα του.[130] Τα εργαλεία αυτά απευθύνονται σε μια μεγάλη ποικιλία από ενδιαφερόμενους, ανεξαρτήτως της πρότερης εμπειρίας τους στον χώρο ή γενικότερα. Σκοπός είναι να χρησιμοποιηθούν από όσους επιθυμούν να έχουν τα κατάλληλα μέσα για να μάθουν.

Ένα από τα εργαλεία που υπάρχουν στην ιστοσελίδα του Ethereum είναι και το ETH.Build (2022). Πρόκειται για ένα εργαλείο το οποίο προσφέρει ένα περιβάλλον όπου ο χρήστης μπορεί να δει με απλά βήματα τον τρόπο με τον οποίο υλοποιούνται οι βασικές έννοιες που αποτελούν τη ραχοκοκαλιά του blockchain. Για να το πετύχει αυτό, χρησιμοποιούνται blocks, τα οποία σύρονται στην επιφάνεια εργασίας για να εξυπηρετήσουν διαφορετικά σενάρια που καλύπτουν θέματα της θεωρίας (π.χ. κρυπτογραφία, ψηφιακές υπογραφές, συναλλαγές). Στα blocks αυτά συμπληρώνονται από τον χρήστη μόνο ορισμένες ιδιότητές τους, κάτι που δεν απαιτεί ιδιαίτερη γνώση. Έτσι, είναι δυνατόν ο χρήστης να αφοσιωθεί στην κατανόηση του σεναρίου παρά να απασχοληθεί με το πώς θα τρέξει το σενάριο.

Στη συνέχεια θα γίνει μια παρουσίαση επιλεγμένων σεναρίων μέσα από την πλατφόρμα του ETH.Build και θα ακολουθήσει εξήγηση των κινήσεων πραγματοποίησής του. Επιπλέον, η παρουσίαση του κάθε σεναρίου θα συνοδεύεται και με αντιστοίχιση με τις θεωρητικές έννοιες που αναλύθηκαν στο βιβλίο.

Τα επιλεγμένα σενάρια περιλαμβάνουν:

1) Σενάριο 0: Γνωριμία με το περιβάλλον του ETH.Build.
2) Σενάριο 1: Χρήση συναρτήσεων κατακερματισμού.
3) Σενάριο 2: Ζεύγη κλειδιών και ψηφιακές υπογραφές.
4) Σενάριο 3: Κρυπτογραφία και αποστολή μηνυμάτων.
5) Σενάριο 4: Συναλλαγές.

Τέλος, ο κώδικας για όλα τα σενάρια που παρουσιάζονται στη συνέχεια μπορεί να βρεθεί και εδώ.

[130] Online σύνδεσμος: https://ethereum.org/en/developers/learning-tools/

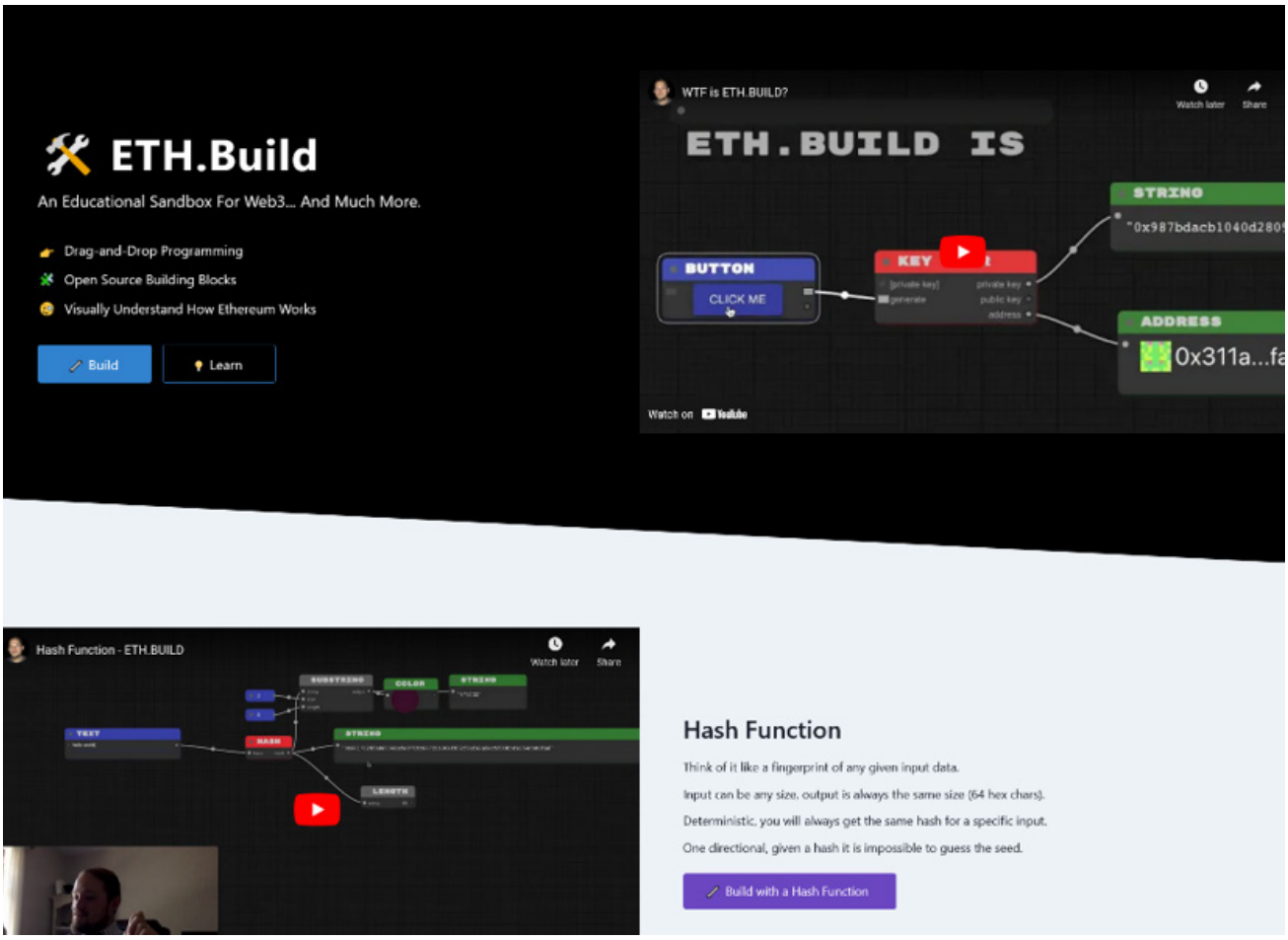
11.1.1 Σενάριο 0: Γνωριμία με το περιβάλλον του ETH.Build

Βασικά στοιχεία σεναρίου:

Σενάριο 0	
Περιγραφή	Δημιουργία της πρώτης ροής στο ETH.Build με έμφαση στην εξήγηση των βασικών χαρακτηριστικών λειτουργίας μιας συνάρτησης κατακερματισμού (hash), με είσοδο ποικίλου μήκους και έξοδος σταθερού και ίσου με 256 bits (64 δεκαεξαδικοί χαρακτήρες).
Blocks που χρησιμοποιούνται	• Crypto/Hash • Input/Text • Display/Watch (STRING)
Ενότητες του βιβλίου που καλύπτονται	Κεφάλαια 1, 2 και 3.

Πίνακας 11.1 *Περιγραφή του Σεναρίου 0*

Για μια πρώτη γνωριμία με το περιβάλλον ανάπτυξης ETH.Build μπορεί κάποιος να επισκεφθεί την αρχική σελίδα (ETH, 2022), όπου θα βρει ό,τι φαίνεται στην **Εικόνα 11.1**.



Εικόνα 11.1 *Η αρχική σελίδα του ETH.Build.*

Στο πάνω μέρος υπάρχουν δύο κουμπιά (*Build, Learn*) μαζί με ένα εισαγωγικό video που σας επιδεικνύει εν συντομία τις δυνατότητες του εργαλείου:

- Το κουμπί Build δίνει πρόσβαση στον χώρο εργασίας, στον οποίο έχουν προστεθεί κάποια blocks που μπορείτε να χρησιμοποιήσετε ως βάση για να φτιάξετε το δικό σας σενάριο. Εναλλακτικά, μπορείτε να σβήσετε ό,τι υπάρχει και να ξεκινήσετε από το μηδέν να δουλεύετε πάνω στο σενάριο που επιθυμείτε.[131]

[131] Για τη διαγραφή του block / των blocks από την επιφάνεια εργασίας είναι δυνατόν είτε η επιλογή του(ς) και μετά χρήση του πλήκτρου delete είτε (αν χρειάζεται να επιλεγούν όλα) χρήση του συνδυασμού Ctrl+A και μετά του πλήκτρου delete.

- Το κουμπί *Learn* σας μεταφέρει στο κανάλι του συγγραφέα (Austin Griffith) στο YouTube. Εκεί μπορείτε να δείτε συνολικά όλα τα videos για τα σενάρια που παρουσιάζονται στην αρχική σελίδα.

Πηγαίνοντας πιο κάτω στην αρχική σελίδα, μπορείτε να δείτε την περιγραφή των σεναρίων που καλύπτονται από το εργαλείο. Για κάθε σενάριο υπάρχουν δύο επιλογές:

- Να παρακολουθήσετε ένα video στην αγγλική γλώσσα με τα θέματα που καλύπτονται στο σενάριο.
- Να πατήσετε να σας ανοίξει μια νέα επιφάνεια εργασίας που περιέχει ορισμένα modules που χρειάζεστε για την ανάπτυξη του σεναρίου.

Στο κεφάλαιο αυτό καλύπτονται αρκετά από τα παραδείγματα, με έμφαση σε όσα έχουν άμεση συσχέτιση με τα πρώτα κεφάλαια του βιβλίου, στα οποία και γίνεται ανάπτυξη κυρίως θεωρητικών εννοιών. Για να είναι εύκολο να προχωρήσετε στην αντιγραφή των βημάτων στη συνέχεια, υποτίθεται ότι ξεκινάτε να εργάζεστε σε έναν κενό χώρο εργασίας.

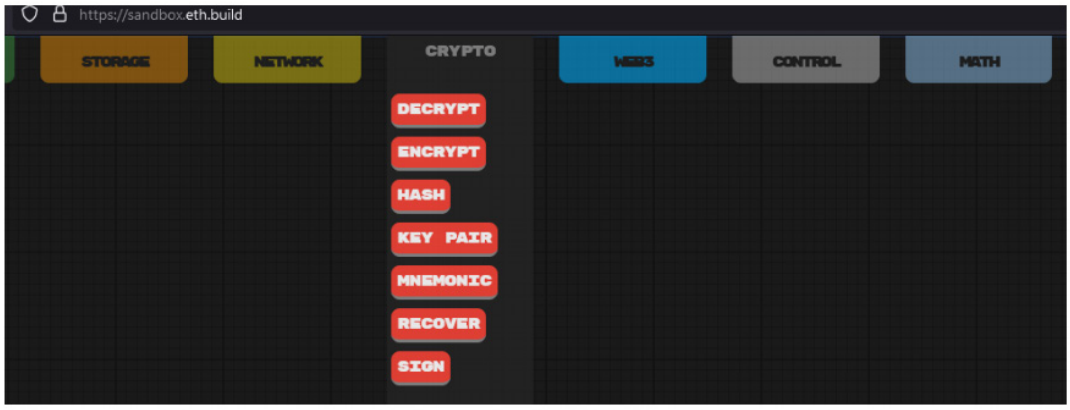
Υπενθυμίζεται ότι η δημιουργία μιας κενής επιφάνειας εργασίας μπορεί να γίνει με όποιον από τους δύο τρόπους που αναφέρθηκαν προηγουμένως:

- Πατώντας το κουμπί Build στην αρχική οθόνη και διαγράφοντας τα blocks που σας εμφανίζονται.
- Πατώντας πάνω στο κουμπί Build που συνοδεύει το κάθε σενάριο στη σελίδα και καθαρίζοντας από blocks την επιφάνεια εργασίας.

Η επιφάνεια εργασίας αποτελείται από menus στα τρία μέρη της οθόνης. Στο πάνω μέρος τα menus περιέχουν blocks και είναι χωρισμένα σε κατηγορίες/είδη. Τα blocks αυτά αποτελούν τα δομικά συστατικά των σεναρίων που θα φτιαχτούν και μπορεί κάποιος να τα σύρει (ή να τα καλέσει) στην επιφάνεια εργασίας για να δημιουργήσει ένα νέο σενάριο. Ακόμα, υπάρχει menu στη δεξιά πλευρά της οθόνης που περιέχει βασικές ενέργειες που μπορούν να πραγματοποιηθούν (π.χ. αντιγραφή, διαγραφή ενός block). Τέλος, υπάρχουν menus και στο κάτω μέρος. Κάποια από αυτά έχουν blocks που δεν ανήκουν στις κατηγορίες στο πάνω μέρος, ενώ το κεντρικό menu στο κάτω μέρος περιέχει πιο βασικές ρυθμίσεις, όπως η αποθήκευση και η φόρτωση αρχείων με σενάρια.

Στη συνέχεια, ξεκινώντας από μια κενή επιφάνεια εργασίας, δίνεται έμφαση στην παρουσίαση ορισμένων blocks. Πιο συγκεκριμένα, το menu Crypto έχει ορισμένα από τα blocks που χρησιμοποιούνται πολύ συχνά στη συνέχεια, όπως τα: *Decrypt, Encrypt, Hash* και *Key Pair* (**Εικόνα 11.2**). Πατώντας επάνω στο menu εμφανίζονται τα blocks που ανήκουν στην κατηγορία που εκφράζει αυτό.

Έτσι, στην Εικόνα 11.2 φαίνονται τα blocks που ανήκουν στην κατηγορία Crypto (που είναι συντομογραφία για το Cryptography – Κρυπτογραφία).



Εικόνα 11.2 *Τα blocks που βρίσκονται στο menu Crypto στην επιφάνεια εργασίας του ETH.Build.*

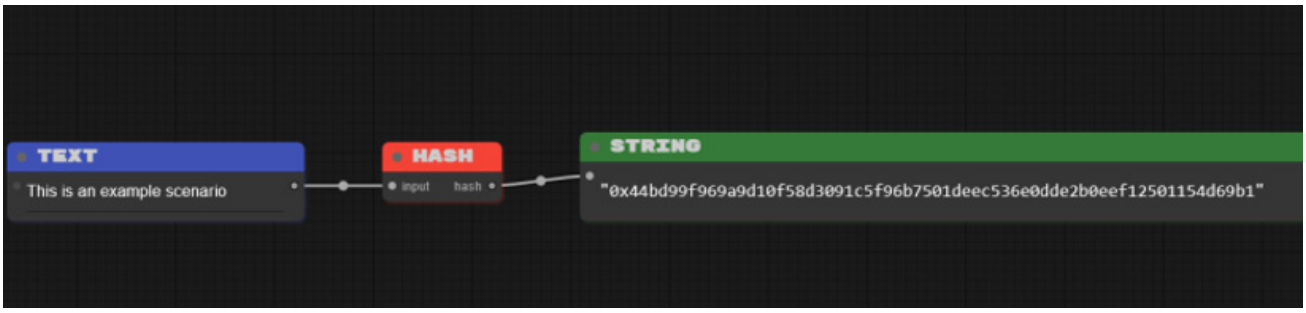
Για να δημιουργήσετε ένα παράδειγμα, μπορείτε να επιλέξετε το επιθυμητό block από το menu (όπως φαίνεται και στην Εικόνα 11.2). Εναλλακτικά, μπορείτε να πατήσετε τα πλήκτρα *SPACE* + / για να ανοίξει ένα μικρό menu, στο οποίο μπορείτε να πληκτρολογήσετε το όνομα του block.

Για τη δημιουργία του πρώτου σεναρίου θα χρησιμοποιηθεί ένα block το οποίο αναλαμβάνει να παίξει τον ρόλο μιας συνάρτησης κατακερματισμού. Το block αυτό, με την ονομασία HASH στο ETH.build, ανήκει στο

menu Crypto και φαίνεται στην Εικόνα 11.2. Για τη δημιουργία του block στην επιφάνεια εργασίας θα πρέπει να χρησιμοποιηθεί η ακόλουθη επιλογή: *SPACE* + / + *HASH* + *ENTER*.

Μόλις προστεθεί το block, φαίνεται ότι έχει 1 είσοδο (Input) και 1 έξοδο (hash). Στην είσοδο θα πρέπει να προστεθεί το κείμενο που θα μπει στη συνάρτηση κατακερματισμού και στην έξοδο (hash) θα παραχθεί το αποτέλεσμα. Για να είναι δυνατόν να εισαχθεί κείμενο στην είσοδο του block, μπορείτε να κάνετε διπλό κλικ (αριστερό) πάνω στο input. Τότε, αυτόματα θα εισαχθεί το block TEXT (ανήκει στο menu Input). Μέσα σε αυτό μπορείτε να εισαγάγετε το κείμενο που θα χρησιμοποιηθεί ως είσοδος στη συνάρτηση κατακερματισμού που υλοποιεί το block HASH. Προς το παρόν εισάγετε την πρόταση *This is an example scenario*. Για να δείτε το αποτέλεσμα του κατακερματισμού, θα πρέπει να κάνετε διπλό κλικ στην επιλογή hash (έξοδος) του block HASH. Τότε, ένα νέο block με όνομα STRING εμφανίζεται και μέσα σε αυτό απεικονίζεται η έξοδος της συνάρτησης κατακερματισμού. Η έξοδος, όπως φαίνεται και στην **Εικόνα 11.3**, είναι ένας δεκαεξαδικός αριθμός που αποτελείται από 64 HEX[132] χαρακτήρες:

0x44bd99f969a9d10f58d3091c5f96b7501deec536e0dde2b0eef12501154d69b1



Εικόνα 11.3 *Τα blocks που χρειάζονται για τη δοκιμή χρήσης μιας συνάρτησης κατακερματισμού (είσοδος + έξοδος).*

Αν αλλάξετε το κείμενο της εισόδου στο block TEXT, τότε θα δείτε ότι παράγεται μια τελείως διαφορετική έξοδος, η οποία όμως έχει το ίδιο μήκος σε αριθμό δεκαεξαδικών συμβόλων. Για παράδειγμα, η προσθήκη ενός θαυμαστικού (!) στο τέλος της πρότασης στην είσοδο (*This is an example scenario!*) οδηγεί σε μια τελείως διαφορετική έξοδο, την:

0x8d98cec96a9592d81e17dca1f02c38d43c2d6cdbdb7e8d4c91307d31288c95bf

Να σημειωθεί ότι η έξοδος αποτελείται και πάλι από 64 δεκαεξαδικά σύμβολα.

Από την άλλη πλευρά, όμως, όσες φορές χρησιμοποιηθεί η ίδια είσοδος και η έξοδος πάντα θα είναι η ίδια. Δηλαδή, αν χρησιμοποιήσετε τις εισόδους που αναφέρθηκαν ακριβώς με τον ίδιο τρόπο (χωρίς επιπλέον κενά), θα φθάσετε να βλέπετε τις εξόδους που φαίνονται παραπάνω.

Αυτό είναι βασικό χαρακτηριστικό της κρυπτογραφίας που χρησιμοποιείται στο blockchain, η οποία παρουσιάστηκε στο Κεφάλαιο 3. Στο κεφάλαιο αυτό αναδείχθηκε και στην πράξη με το απλό παράδειγμα (σενάριο) που φαίνεται στην Εικόνα 11.3.

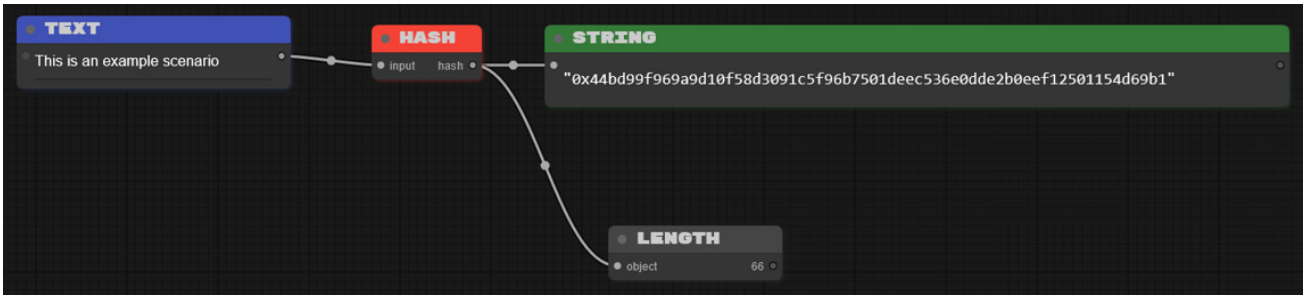
Πριν ολοκληρωθεί το πρώτο δοκιμαστικό σενάριο, θα γίνει και μια επαλήθευση του μήκους της εξόδου, δηλαδή του αριθμού των δεκαεξαδικών χαρακτήρων που αυτή έχει. Αναφέρθηκε προηγουμένως ότι ο αριθμός αυτός είναι ίσος με 64 χαρακτήρες. Πιθανώς στην επαλήθευσή σας μετρώντας θα καταλήξατε ότι οι χαρακτήρες είναι 66. Οι δύο πρώτοι χαρακτήρες όμως δηλώνουν το σύστημα στο οποίο αποδίδονται οι αριθμοί (δείτε και σημείωση στο κάτω μέρος). Έτσι το 0x δηλώνει ότι ο αριθμός που ακολουθεί είναι δεκαεξαδικός και το υπόλοιπο εκφράζει τον αριθμό αυτόν.

Γνωρίζοντας ότι ένας δεκαεξαδικός χαρακτήρας χρειάζεται 4 bits για να αποδοθεί, είναι εύκολο να υπολογιστεί ότι οι 64 δεκαεξαδικοί χαρακτήρες εκπροσωπούν μια (σταθερού μεγέθους) έξοδο ίση με 256 bits (4*64).

---

[132] Το δεκαεξαδικό σύστημα συμβολίζεται σε συντομογραφία ως HEX (ή H στο τέλος του αριθμού). Επίσης, μπορεί να χρησιμοποιηθεί και ο συμβολισμός στην αρχή 0x (όπως φαίνεται και στην Εικόνα 11.3).

Για να επιβεβαιωθεί ο αριθμός των χαρακτήρων στην έξοδο της συνάρτησης κατακερματισμού, θα χρησιμοποιηθεί ένα block που επιστρέφει το μήκος της εξόδου. Το block αυτό ονομάζεται LENGTH και για να προστεθεί πρέπει να χρησιμοποιήσετε τον συνδυασμό *SPACE* + / + *LENGTH* και να επιλέξετε το block OBJECT/LENGTH. Στη συνέχεια συνδέστε την άκρη της εξόδου του HASH με την είσοδο στο block LENGTH. Για να γίνει η σύνδεση, κάντε αριστερό κλικ στην έξοδο του HASH. Χωρίς να αφήσετε το κουμπί στο mouse, μεταφέρετε τον κέρσορα πάνω από την είσοδο του block LENGTH (γράφει object) και αφήστε τον πάνω της. Η διαδικασία είναι πετυχημένη όταν δείτε τη σύνδεση να αποδίδεται στην επιφάνεια εργασίας. Με την ολοκλήρωση της εργασίας, θα πρέπει να βλέπετε αυτό που φαίνεται στην **Εικόνα 11.4**. Προσέξτε ότι η έξοδος του block LENGTH αναφέρει ότι η έξοδος του HASH αποτελείται από 66 σύμβολα (64 + 2 αρχικά που δηλώνουν το σύστημα).



Εικόνα 11.4 *Τα blocks που χρειάζονται για το πρώτο δοκιμαστικό σενάριο με την προσθήκη του LENGTH για την επιβεβαίωση του μήκους της εξόδου από τη συνάρτηση κατακερματισμού.*

Μπορείτε να αλλάξετε την είσοδο με ό,τι κείμενο σκεφτείτε, ελέγχοντας ταυτόχρονα τόσο τη μεγάλη αλλαγή στην έξοδο του HASH (περιεχόμενο του block STRING) όσο και τη σταθερή τιμή στον αριθμό των συμβόλων που φαίνεται στο block LENGTH. Η **Εικόνα 11.4** απεικονίζει τη ροή που έχει δημιουργηθεί για την επιβεβαίωση του μήκους της εξόδου της συνάρτησης κατακερματισμού HASH.

Όταν ολοκληρωθεί ένα σενάριο, είναι δυνατόν να αποθηκευτούν τα blocks που χρησιμοποιήθηκαν, μαζί με τις ρυθμίσεις που έγιναν σε αυτά. Αυτό επιτυγχάνεται με την επιλογή όλων των blocks (χρήση του συνδυασμού πλήκτρων Ctrl + A) και μετά από το menu MODULES επιλέγετε το SAVE. Επιπλέον, όποιος επιθυμεί να δει ολοκληρωμένο το σενάριο μπορεί μέσα από το ETH.Build να επιλέξει από το MODULES την επιλογή LOAD για να καταφέρει να φορτώσει στον υπολογιστή τη δουλειά αυτή.

Όσοι δυσκολεύτηκαν να παρακολουθήσουν τα βήματα για τη δημιουργία της ροής του σεναρίου μπορούν να πάνε στο ETH.Build και σε μια κενή επιφάνεια εργασίας να πατήσουν LOAD και να επιλέξουν να φορτώσουν τον έτοιμο κώδικα του Σεναρίου 0 που υπάρχει στο link στην αρχή του κεφαλαίου.

Τέλος, στο επόμενο Σενάριο (Σενάριο 1) χρησιμοποιείται η βάση που δόθηκε στην Εικόνα 11.4 και επεκτείνεται στη δημιουργία κλειδιών και διευθύνσεων.

11.1.2 Σενάριο 1: Χρήση συναρτήσεων κατακερματισμού (Hash Functions) και Merkle Trees

Βασικά στοιχεία σεναρίου:

Σενάριο 1	
Περιγραφή	Δημιουργία ροής που προσθέτει πάνω στα αποτελέσματα του Σεναρίου 0 όσον αφορά τις εξόδους από μια συνάρτηση κατακερματισμού. Επίσης, επιδεικνύεται πώς ο συνδυασμός των εξόδων από τις συναρτήσεις κατακερματισμού μπορεί να οδηγήσει σε ένα μοναδικό αποτέλεσμα, όπως ακριβώς συμβαίνει και στα Merkle Trees που αναπτύσσονται μέσα σε ένα block και αφορούν τις συναλλαγές που υπάρχουν μέσα σε αυτό.
Blocks που χρησιμοποιούνται	• Crypto/Hash • Display/Watch (STRING) • Display/Color • Input/Number • Input/Text • Input/Length • String/Substring • Storage/File Drop • String/Combine
Ενότητες του βιβλίου που καλύπτονται	Κεφάλαια 1, 2 και 3.

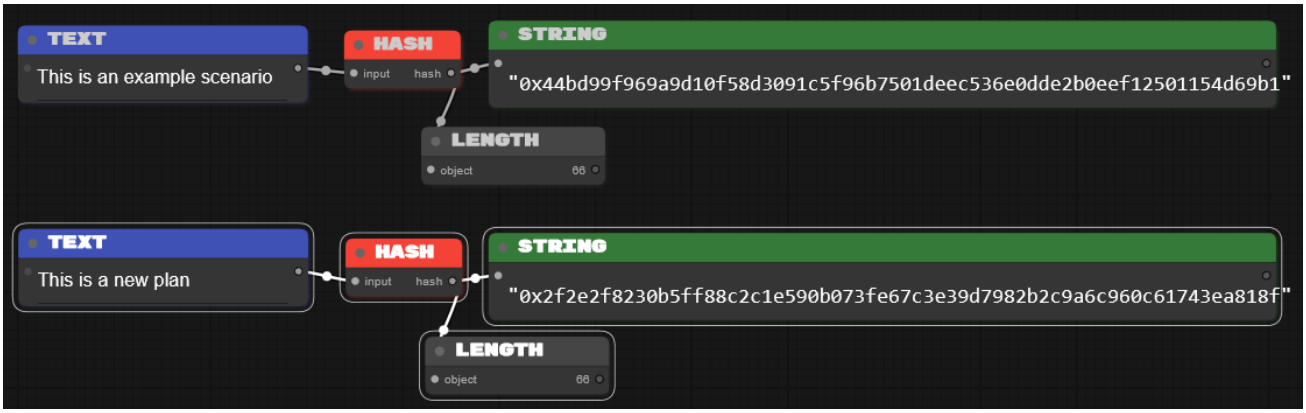
Πίνακας 11.2 *Περιγραφή του Σεναρίου 1.*

Το Σενάριο 1 συνεχίζει από το σημείο που έκλεισε το Σενάριο 0 και προχωρά να δώσει έναν οπτικό τρόπο παρουσίασης των εξής θεμάτων:

1. Οι συναρτήσεις κατακερματισμού μπορούν να δέχονται ποικίλου μεγέθους εισόδους και να παράγουν σταθερού μεγέθους εξόδους.
2. Κάθε φορά που μια συνάρτηση κατακερματισμού δέχεται την ίδια είσοδο θα παράγει *πάντοτε* την ίδια έξοδο.
3. Οι συναρτήσεις κατακερματισμού λειτουργούν σε μία κατεύθυνση. Δηλαδή, αν πάρει κάποιος την έξοδο μιας συνάρτησης κατακερματισμού, δεν θα μπορεί να βρει την είσοδο. Μόνο με δοκιμές (και τύχη!).

Το θέμα 1, όπως αναφέρεται παραπάνω, επιβεβαιώθηκε στο Σενάριο 0 με την αλλαγή στην είσοδο (δηλαδή το κείμενο που μπαίνει στο block TEXT) που ζητήθηκε, η οποία προκαλεί και αλλαγή στην έξοδο του block. Από το block LENGTH επιβεβαιώνεται ότι, αν και το περιεχόμενο της εξόδου είναι διαφορετικό, παρ' όλα αυτά το μήκος παραμένει σταθερό στους 66 χαρακτήρες (64 δεκαεξαδικά σύμβολα + 2 χαρακτήρες που συμβολίζουν το σύστημα με το οποίο αποδίδονται οι υπόλοιποι αριθμοί).

Στην **Εικόνα 11.5** φαίνεται αυτή η διαφορά στην έξοδο καθώς και το σταθερό μήκος της.



Εικόνα 11.5 *Η επιβεβαίωση της σταθερού μήκους εξόδου της συνάρτησης HASH, με διαφορετικό όμως αποτέλεσμα όταν αυτή τροφοδοτείται από διαφορετικές εισόδους.*

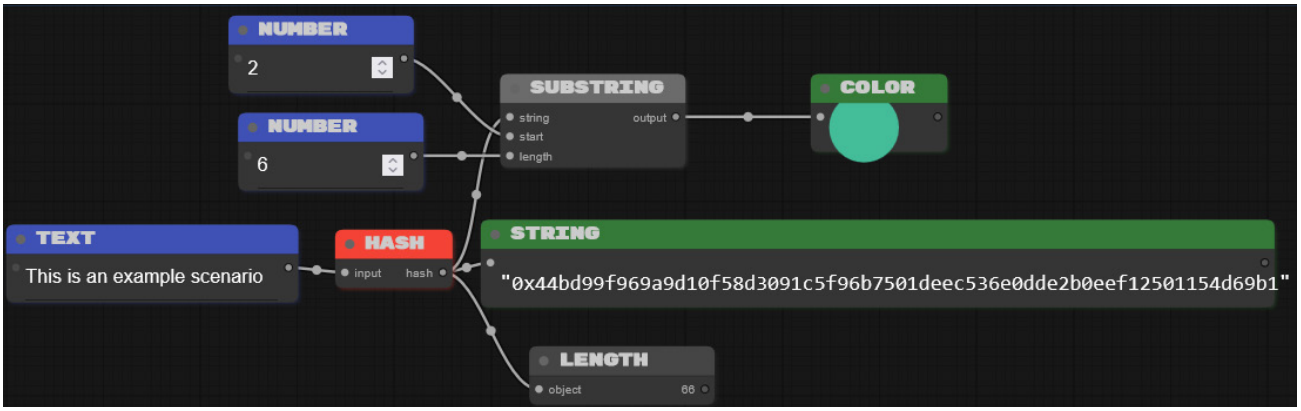
Δουλεύοντας παραπάνω τα θέματα 1 και 2, το εργαλείο του ETH.Build δίνει έναν ακόμα πιο ευχάριστο τρόπο να απεικονιστεί το πώς λειτουργεί η συνάρτηση κατακερματισμού. Αυτό γίνεται με την απομόνωση ορισμένων

bits της εξόδου και της μετατροπής τους σε ένα χρώμα. Έτσι, με την αλλαγή στην είσοδο θα βλέπετε και αλλαγή στα χρώματα της εξόδου.

Για να το πετύχετε αυτό, θα χρησιμοποιήσετε τη ροή στην Εικόνα 11.4 και θα προσθέσετε ένα block με το όνομα SUBSTRING. Αυτό γίνεται με τον συνδυασμό *SPACE* + / + *SUBSTRING* και επιλέγοντας με το mouse το μοναδικό block που έχει μείνει ως επιλογή. Στη συνέχεια συνδέετε (δείτε στο Σενάριο 0 πώς δημιουργείται μια σύνδεση μεταξύ δύο blocks) την έξοδο από το block HASH με την είσοδο STRING από το block SUBSTRING. Ο λόγος που χρησιμοποιείται το block αυτό είναι ότι για την απόδοση του χρώματος στην έξοδο θα γίνει χρήση του προτύπου RGB, το οποίο χρησιμοποιεί 24 bits (ή 6 δεκαεξαδικούς χαρακτήρες). Οι 6 αυτοί δεκαεξαδικοί χαρακτήρες που θα χρησιμοποιηθούν θα είναι οι 6 πρώτοι χαρακτήρες της εξόδου (χωρίς να συμπεριλαμβάνονται τα αρχικά σύμβολα δήλωσης μετρικού συστήματος). Επομένως, με το block αυτό θα πρέπει να απομονωθούν τα bits της εξόδου και να αποδοθούν σαν χρώματα. Αυτό γίνεται δίνοντας τις κατάλληλες τιμές στις άλλες δύο εισόδους του block SUBSTRING. Έτσι, με διπλό κλικ στην έξοδο start θα εμφανιστεί ένα block INPUT/NUMBER, στο οποίο δηλώνουμε από ποιον χαρακτήρα της εισόδου STRING θα αρχίσει να παρακολουθεί την τιμή το block SUBSTRING. Δεδομένου ότι οι 2 πρώτοι χαρακτήρες δηλώνουν το μετρικό σύστημα, η παρακολούθηση θα πρέπει να ξεκινήσει από τον τρίτο. Η τιμή όμως που θα δοθεί μέσα στο block NUMBER είναι η τιμή 2, καθώς, όπως συμβαίνει συχνά στον προγραμματισμό, η αρίθμηση ξεκινά από τη θέση 0.

Στη συνέχεια θα πρέπει να δηλωθεί ο αριθμός των χαρακτήρων που θα παρακολουθεί το block SUBSTRING. Με διπλό κλικ στην είσοδο length δημιουργείται ένα νέο block NUMBER, στο οποίο αυτή τη φορά θα περάσει η τιμή 6 (θυμηθείτε ότι 6*4 = 24 bits, όσα χρειάζεται και το RGB).

Τέλος, θα προστεθεί το block που θα δείχνει το χρώμα στο οποίο αντιστοιχούν τα 24 πρώτα bits της εξόδου της συνάρτησης HASH. Θα χρειαστεί να χρησιμοποιήσετε τον συνδυασμό *SPACE* + / + *COLOR* και να επιλέξετε το block. Κατόπιν θα συνδέσετε την είσοδο του νέου Color block με την έξοδο του SUBSTRING. Όταν ολοκληρωθούν τα βήματα αυτά, θα δείτε το χρώμα στο block COLOR. Η ροή σας πρέπει να είναι ίδια με αυτήν που απεικονίζεται στην **Εικόνα 11.6**.



Εικόνα 11.6 *Η ροή του Σεναρίου 1 με την προσθήκη του block COLOR για τη χρωματική απόδοση των πρώτων 24 bits της εξόδου της συνάρτησης HASH.*

Μπορείτε να δοκιμάσετε να αλλάξετε την είσοδο για να παρατηρήσετε τόσο την αλλαγή στην έξοδο (στα δεκαεξαδικά σύμβολα) όσο και στο χρώμα της εξόδου (block COLOR).

Επιπλέον, μπορείτε να χρησιμοποιήσετε στην είσοδο της συνάρτησης κατακερματισμού αντί για ένα κείμενο ένα οποιοδήποτε αρχείο και να παρατηρήσετε την έξοδο (και το χρώμα). Αυτό μπορεί να γίνει αν προσθέσετε το block Storage/File Drop πατώντας: *SPACE* + / + *File Drop* και επιλέγοντας το block που απομένει. Κατόπιν θα πρέπει να κόψετε τη σύνδεση μεταξύ TEXT και HASH (επιλογή του καλωδίου και delete) και να δημιουργήσετε νέα σύνδεση μεταξύ του File Drop και του Hash (input). Όταν ολοκληρωθεί αυτή, θα μπορείτε να μεταφέρετε (σύροντας και αφήνοντας) οποιοδήποτε αρχείο (π.χ. μια φωτογραφία) και θα δείτε την έξοδο πώς μεταβάλλεται αντίστοιχα.

Ο κώδικας της ροής μέχρι εδώ μπορεί να βρεθεί στο αποθετήριο των παραδειγμάτων στον Φάκελο Σενάριο 1, με τίτλο Scenario1_PartA.module.

Μέχρι το σημείο αυτό έχουν καλυφθεί τα δύο από τα τρία θέματα του παραδείγματος. Για να γίνει αντιληπτό το θέμα 3, θα χρησιμοποιηθεί μια έξοδος της συνάρτησης HASH:

0x85cc825a98ec217d960f113f5f80a95d7fd18e3725d37df428eb14f880bdfc12

Κοιτώντας προσεκτικά, θα εντοπίσετε ότι δεν είναι ίδια με καμία από τις προηγούμενες εισόδους που συζητήθηκαν εδώ. Επομένως δεν γνωρίζετε την είσοδο που χρησιμοποιήθηκε για την εύρεσή της. Βρίσκεστε, επομένως, ακριβώς στη θέση κάποιου που γνωρίζει την έξοδο και θέλει να βρει την είσοδο (π.χ. αφορά το κλειδί ή τη διεύθυνση ενός χρήστη). Ο μόνος τρόπος για να το πετύχετε είναι να δημιουργήσετε τη ροή της Εικόνας 11.3 και να δοκιμάσετε να προσθέσετε εισόδους και να συγκρίνετε την έξοδο με τη δοθείσα. Βεβαίως, δεν μπορείτε να ξέρετε αν αυτή η έξοδος προκύπτει από κείμενο (TEXT) ή ένα αρχείο (File Drop). Στη δεύτερη περίπτωση πρέπει να έχετε και αντίγραφο του αρχείου για την επιβεβαίωση. Αντιλαμβάνεστε, τώρα, τη δυσκολία στην επίλυση του προβλήματος από την αντίθετη φορά. Στο Κεφάλαιο 3 έχει εξηγηθεί και μαθηματικά γιατί θεωρούνται μονής κατεύθυνσης οι συναρτήσεις κατακερματισμού.

Όσο για τη λύση, σας καλούμε να δοκιμάσετε να τη βρείτε. Απαντήσεις θα δοθούν στο τέλος της περιγραφής του Σεναρίου 1.

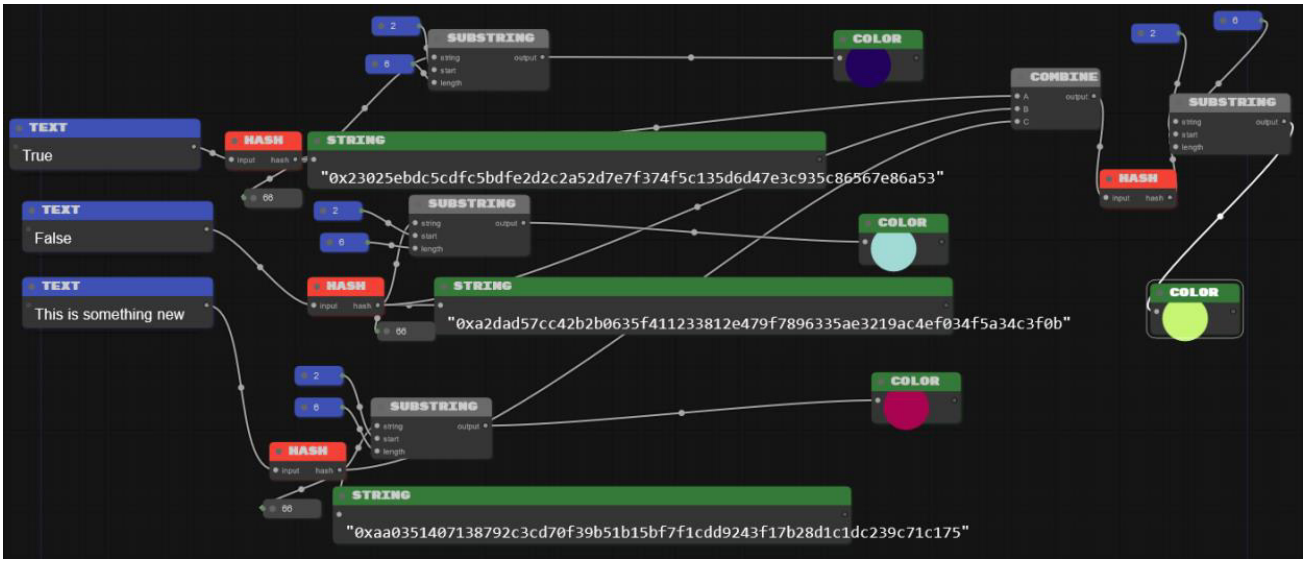
Merke Trees

Η λογική της χρήσης συναρτήσεων κατακερματισμού είναι παρούσα σε πολλά σημεία σε ένα δίκτυο blockchain (βλ. Κεφάλαια 2 και 3). Ένα από τα σημεία αυτά είναι και η δημιουργία των Merkle Trees σε ένα block για να αποδοθούν εν συντομία όλες οι συναλλαγές που περικλείονται σε αυτό.

Η δημιουργία των Merkle Trees σε μικρογραφία μπορεί να γίνει με τη βοήθεια της αναπαραγωγής της ροής στην Εικόνα 11.5 και με την προσθήκη ορισμένων blocks για τη διαχείριση της συνολικής εικόνας.

Αρχικά μπορείτε να αντιγράψετε όλα τα blocks της Εικόνας 11.5 και να τα αναπαραγάγετε άλλες δύο φορές από κάτω (και περισσότερες, αν μπορείτε να τα βολέψετε στην επιφάνεια εργασίας). Κατόπιν πρέπει να προστεθεί το block STRING/Combine (*SPACE* + / + *Combine*) και να συνδεθεί με τις εξόδους από τις τρεις συναρτήσεις HASH. Κατόπιν, στην έξοδο προστίθεται μια νέα συνάρτηση HASH που συνδέεται με ένα block SUBSTRING με παρόμοιες εισόδους με τα άλλα (start 2, length 6) και έξοδο σε ένα νέο COLOR block.

Η ροή σας θα πρέπει να είναι αυτή που απεικονίζεται στην **Εικόνα 11.7**.



Εικόνα 11.7 *Η ροή που προσομοιάζει τη λειτουργία των Merkle Trees.*

Το COLOR block με το ανοικτό πράσινο χρώμα δείχνει το συνολικό HASH του συνδυασμού των τριών εισόδων. Μπορείτε να αλλάξετε το κείμενο στην είσοδο και να παρατηρήσετε πώς μεταβάλλεται τόσο το χρώμα στο HASH που συνδέεται με αυτό, αλλά και το χρώμα του συνδυασμού.

Το παραπάνω παρουσιάζει τη λειτουργία των Merkle Trees (Κεφάλαιο 3) και αναδεικνύει με εμφανή τρόπο (με τη βοήθεια των χρωμάτων) πώς μια αλλαγή στο περιεχόμενο μιας συναλλαγής μπορεί να επηρεάσει εντελώς το παραγόμενο Merkle Root (το χρώμα στον συνδυασμό δηλαδή).

Απάντηση: Η λέξη που χρησιμοποιήθηκε ως είσοδος και που σας ζητήθηκε να αναζητήσετε ενώ γνωρίζετε μόνο την έξοδό της ήταν η λέξη *Test*. Μπορείτε να τη βάλετε ως είσοδο σε μια συνάρτηση HASH και να επιβεβαιώσετε την έξοδο με αυτήν που αναφέρεται παραπάνω.

11.1.3 Σενάριο 2: Ζεύγη κλειδιών και ψηφιακές υπογραφές

Βασικά στοιχεία σεναρίου:

Σενάριο 2	
Περιγραφή	Το σενάριο αυτό ασχολείται με την παραγωγή και χρήση του ζεύγους κλειδιών της ασύμμετρης κρυπτογραφίας (ιδιωτικό – δημόσιο) για την παραγωγή διευθύνσεων στο Ethereum και τη δημιουργία ψηφιακής υπογραφής. Τέλος, αποδεικνύεται ότι η χρήση της υπογραφής και του μηνύματος μπορεί να αποδείξει τη διεύθυνση του αποστολέα (άρα και τη γνησιότητα ή μη του μηνύματος).
Blocks που χρησιμοποιούνται	• Crypto/Key Pair • Crypto/Sign • Crypto/Recover • Display/Watch (STRING) • Display/Address • Input/Text • Input/Button • Object/Length
Ενότητες του βιβλίου που καλύπτονται	Κεφάλαια 1, 2 και 3.

Πίνακας 11.3 *Περιγραφή του Σεναρίου 2.*

Στο Σενάριο 2 δημιουργείται μια ροή η οποία χρησιμοποιείται για την επίδειξη των δυνατοτήτων της ασύμμετρης κρυπτογραφίας. Πιο αναλυτικά:

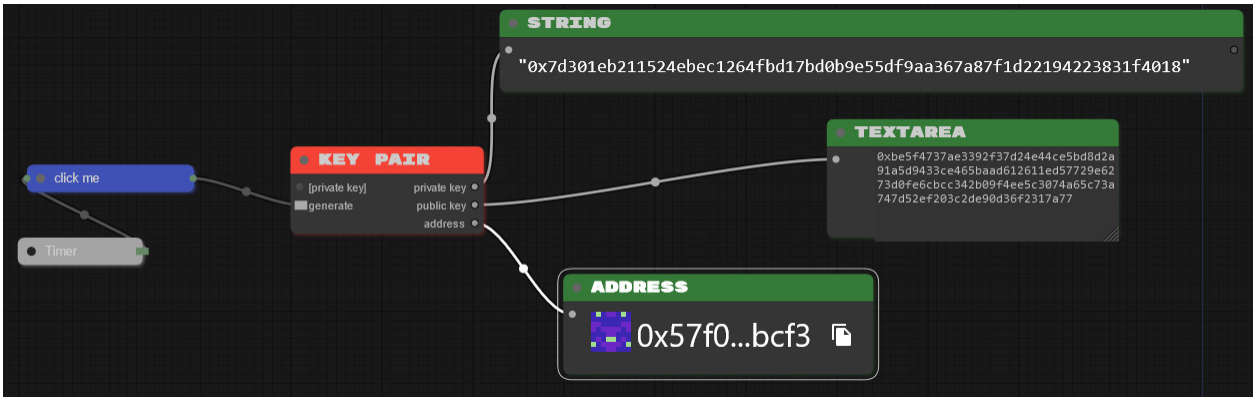
- Θα δημιουργηθούν τα ζεύγη κλειδιών (ιδιωτικό, δημόσιο) ξεκινώντας από την τιμή του πρώτου.
- Θα δημιουργηθούν οι διευθύνσεις στο Ethereum από το ζεύγος κλειδιών.
- Θα παραχθούν μηνύματα που έχουν υπογραφεί ψηφιακά από τον χρήστη.
- Θα γίνει επιβεβαίωση ότι ο χρήστης έχει υπογράψει ψηφιακά ένα μήνυμα.

Για την εξυπηρέτηση των σκοπών του σεναρίου κεντρικό ρόλο παίζει το block KEY PAIR (στο menu Crypto). Το block αυτό ουσιαστικά υλοποιεί τον αλγόριθμο Elliptic Curve Digital Signature Algorithm (ECDSA) (δείτε Κεφάλαιο 3) για την παραγωγή του ζεύγους κλειδιών και της διεύθυνσης.

Το πρώτο βήμα, λοιπόν, είναι η προσθήκη του block αυτού στην επιφάνεια εργασίας με τον γνωστό από προηγουμένως τρόπο.[133] Το block αυτό δέχεται 2 εισόδους (η 1η είσοδος είναι η προσθήκη από άλλο block του ιδιωτικού κλειδιού, ενώ η 2η είσοδος δημιουργεί τυχαία το κλειδί το οποίο μπορεί να επαναλαμβάνει ανάλογα με τον Timer που θα χρησιμοποιηθεί) και έχει 3 εξόδους.

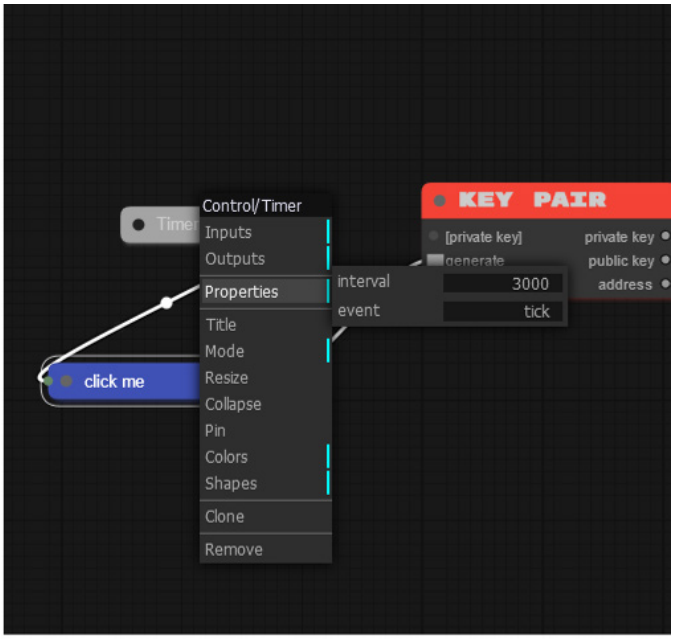
Η 1η έξοδος απεικονίζει (κάνοντας διπλό κλικ πάνω στην έξοδο) σε ένα block STRING την απόδοση του ιδιωτικού κλειδιού (private key), το οποίο είτε δημιουργήθηκε από τον χρήστη (μέσω της 1ης εισόδου) είτε από το KEY PAIR block (μέσω της 2ης εισόδου). Επιπλέον, έχοντας ως βάση το ιδιωτικό κλειδί, κάνοντας διπλό κλικ στη 2η έξοδο (public key) θα δείτε το δημόσιο κλειδί που συνδέεται με το ιδιωτικό στο block TEXT AREA που αναπτύσσεται για την προβολή. Τέλος, με διπλό κλικ στην 3η έξοδο (ADDRESS) φαίνεται η διεύθυνση στο Ethereum που συνδέεται με το ζεύγος κλειδιών που δημιουργείται.

[133] *SPACE* + / + (όνομα block) και επιλογή αυτού.



Εικόνα 11.8 *Η ροή στο Σενάριο 2 με βάση το block του KEY PAIR για τη δημιουργία του ζεύγους ιδιωτικού – δημόσιου κλειδιού και της διεύθυνσης στο Ethereum που τους αντιστοιχεί.*

Όπως φαίνεται και στην **Εικόνα 11.8**, έχει χρησιμοποιηθεί η 2η είσοδος (διπλό κλικ επάνω της) για τυχαία δημιουργία του ιδιωτικού κλειδιού. Μάλιστα, με το διπλό κλικ δημιουργούνται 2 blocks: το click me (για να δοθεί έναυσμα στο KEY PAIR να δημιουργήσει το ιδιωτικό κλειδί) αλλά και το Timer, το οποίο ρυθμίζει κάθε πότε θα παράγεται νέο ιδιωτικό κλειδί. Με δεξί κλικ πάνω στο Timer μπορείτε να αλλάξετε τις ρυθμίσεις για την περιοδική δημιουργία νέου ιδιωτικού κλειδιού, όπως φαίνεται και στην **Εικόνα 11.9**. Σε όλα τα blocks η χρήση του δεξιού κλικ επιτρέπει την πρόσβαση σε χαρακτηριστικά του block.



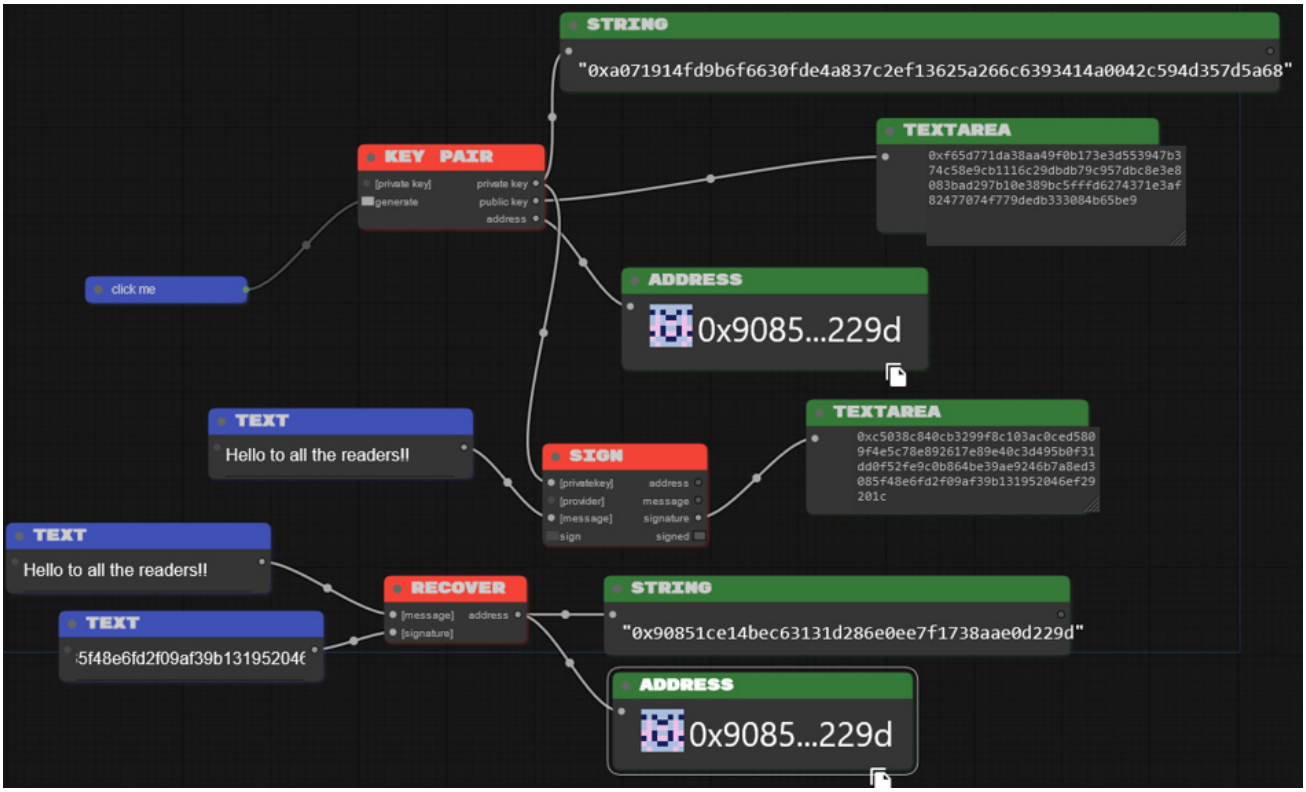
Εικόνα 11.9 *Οι επιλογές που υπάρχουν ως προς την επιλογή χρονικού διαστήματος για την αλλαγή του ιδιωτικού κλειδιού με χρήση δεξιού κλικ στο block Timer.*

Στη συνέχεια θα γίνει προσθήκη στην υπάρχουσα ροή (**Εικόνα 11.8**) του block που θα χρησιμοποιηθεί για την υπογραφή ενός μηνύματος. Το block είναι το SIGN (menu *Crypto*) και το οποίο έχει 4 εισόδους και 4 εξόδους. Τοποθετώντας το, θα συνδέσετε την έξοδο του KEY PAIR (private key) με την ομώνυμη είσοδο του block SIGN. Με τον τρόπο αυτόν το block έχει ό,τι χρειάζεται για να υπογράψει το μήνυμα. Στη συνέχεια, με διπλό κλικ στην είσοδο message δημιουργείται ένα TEXT block όπου θα εισαγάγετε το μήνυμα που θα υπογραφεί ψηφιακά από εσάς. Υπενθυμίζεται ότι η υπογραφή είναι το κρυπτογραφικό αποτέλεσμα του συνδυασμού του ιδιωτικού κλειδιού με το μήνυμα. Με διπλό κλικ στην έξοδο signature θα δείτε την υπογραφή που αφορά το υπάρχον μήνυμα (μην ξεχάσετε να προσθέσετε κάτι στο TEXT block). Προσέξτε πως, αν αλλάξετε το κείμενο του μηνύματος, θα αλλάξει και η υπογραφή σας.

Ακολούθως θα χρησιμοποιηθεί η υπογραφή και το μήνυμα και θα μελετηθεί πώς είναι δυνατόν να επιβεβαιωθεί η διεύθυνση του αποστολέα. Για να γίνει η επιβεβαίωση, χρειάζεται να χρησιμοποιηθεί το block RECOVER, το οποίο δέχεται 2 εισόδους: το μήνυμα και την υπογραφή, και ως έξοδο έχει τη διεύθυνση του αποστολέα. Με δεξί κλικ στις 2 εισόδους δημιουργούνται 2 TEXT blocks. Σε αυτά θα πρέπει να αντιγραφούν στην 1η είσοδο το μήνυμα (από το προηγούμενο block) και στην 2η είσοδο η υπογραφή (έξοδος του block SIGN). Με διπλό κλικ στην έξοδο παρουσιάζεται η διεύθυνση του αποστολέα του μηνύματος. Κοιτώντας στο block KEY PAIR είναι δυνατόν να επιβεβαιωθεί ότι όντως έχει αποσταλεί από τον χρήστη, του οποίου το ιδιωτικό κλειδί χρησιμοποιήθηκε για τη δημιουργία της υπογραφής.

Η **Εικόνα 11.10** δείχνει την τελική ροή του Σεναρίου 2, με την επιβεβαίωση του αποστολέα του μηνύματος.

Ακόμα, έχει συνδεθεί και ένα block Display/ADDRESS με την έξοδο της RECOVER για να είναι πιο εύκολο οπτικά να δείτε ότι οι δύο διευθύνσεις όντως είναι ίδιες.



Εικόνα 11.10 *Η τελική ροή στο Σενάριο 2 με χρήση των blocks δημιουργίας ψηφιακής υπογραφής και επιβεβαίωσης της διεύθυνσης του αποστολέα*

Στο blockchain με τον τρόπο αυτόν ένας χρήστης μπορεί να υπογράψει ένα μήνυμα και ένας άλλος χρήστης, αφού λάβει το μήνυμα και την υπογραφή του χρήστη, να επιβεβαιώσει μέσω της διεύθυνσης ότι όντως αυτός το έχει στείλει. Η επιβεβαίωση οφείλεται στο γεγονός ότι μόνο ένας χρήστης γνωρίζει το ιδιωτικό κλειδί που χρησιμοποιείται για την υπογραφή και η διεύθυνση εξαρτάται από αυτό το ιδιωτικό κλειδί.

Ο κώδικας της συνολικής ροής υπάρχει στο αποθετήριο, στον φάκελο Σενάριο 2.

Το επόμενο σενάριο προχωρά ένα βήμα παραπέρα, με την αποστολή κωδικοποιημένων μηνυμάτων με το δημόσιο κλειδί του χρήστη που απαιτούν γνώση του ιδιωτικού κλειδιού για την αποκωδικοποίηση και ανάγνωσή τους.

11.1.4 Σενάριο 3: Κρυπτογραφία και αποστολή μηνυμάτων

Βασικά στοιχεία σεναρίου:

Σενάριο 3	
Περιγραφή	Το σενάριο αυτό ασχολείται
Blocks που χρησιμοποιούνται	• Crypto/Sign • Crypto/Key Pair • Crypto/Hash • Crypto/Recover • Crypto/Encrypt • Crypto/Decrypt • Display/Watch (STRING) • Display/Address • Input/Text • Input/Button • Network/Publish • Network/Subscribe
Ενότητες του βιβλίου που καλύπτονται	Κεφάλαιο 1,2 και 3

Πίνακας 11.4 *Περιγραφή του Σεναρίου 3.*

Στο σενάριο αυτό θα μελετηθεί περαιτέρω η επικοινωνία και θα αναδειχθεί ο ρόλος της κωδικοποίησης και της ασύμμετρης κρυπτογραφίας.

Πιο αναλυτικά, στο Σενάριο 3 θα μελετηθεί μια ροή με την οποία:

- θα δημιουργηθεί ένας χρήστης με το δικό του ζεύγος κλειδιών και τη δική του διεύθυνση,
- θα χρησιμοποιηθεί ένας pub-sub server για την αποστολή μηνυμάτων,
- τα μηνύματα που θα αποσταλούν θα είναι και χωρίς κωδικοποίηση και με κωδικοποίηση.

Σκοπός είναι να φανεί η διαδικτυακή επικοινωνία και πώς αυτή επηρεάζεται από τη χρήση κωδικοποίησης ή όχι. Στην περίπτωση χρήσης, θα πρέπει η κωδικοποίηση να γίνει με τέτοιον τρόπο ώστε μόνο ο αποδέκτης να μπορεί να διαβάσει το μήνυμα αυτό. Κατόπιν θα συζητηθεί πώς μπορεί να επεκταθεί το σενάριο στην περίπτωση δύο χρηστών με χρήση του pub-sub server για την επικοινωνία.

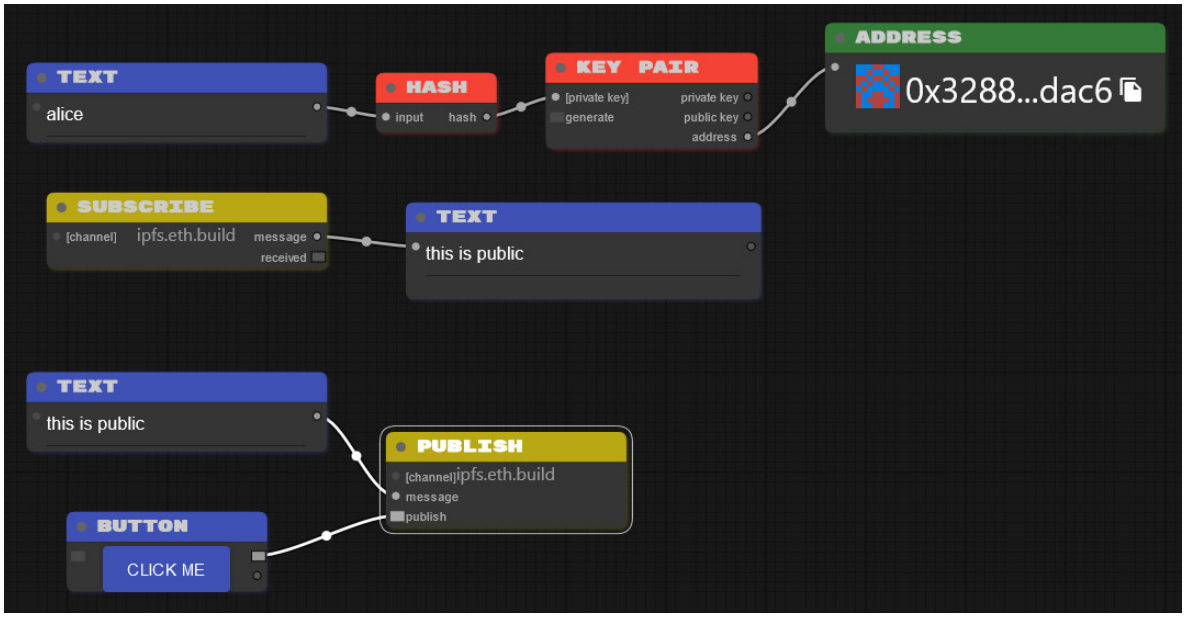
Αρχικά, για τη ροή θα χρησιμοποιηθούν τα ίδια blocks που απεικονίζονται στην Εικόνα 11.7. Δηλαδή, ό,τι απαιτείται για τη δημιουργία ενός ιδιωτικού κλειδιού και των παραγώγων αυτού (δημόσιο κλειδί, διεύθυνση), βασικά στοιχεία ενός λογαριασμού χρήστη. Ας ονομαστεί ο χρήστης αυτός *alice* και αυτό θα είναι και το κείμενο από το οποίο θα παραχθεί το ιδιωτικό κλειδί.

Κατόπιν, θα τοποθετηθούν τα blocks εκείνα που αφορούν τη δημιουργία και την αποστολή ενός μηνύματος. Το μήνυμα αυτό θα ταξιδέψει διαδικτυακά μέσω ενός pub-sub server που υποστηρίζεται από την εφαρμογή του ETH.Build. Για να γίνει αυτό, χρειάζονται δύο διαφορετικά blocks: ένα για την αποστολή του μηνύματος (PUBLISH) και ένα για τη λήψη αυτού (SUBSCRIBE). Η αποστολή και η λήψη γίνεται στο κανάλι που προσφέρεται από το block με το όνομα *ipfs.eth.build*.

Έτσι, λοιπόν, θα προστεθούν τα blocks Network/PUBLISH και Network/SUBSCRIBE για να προσφέρουν τις προαναφερθείσες λειτουργίες. Ξεκινώντας πρώτα από το block PUBLISH, με διπλό κλικ στην είσοδο message προστίθεται ένα TEXT block, το οποίο χρησιμοποιείται για τη σύνταξη του μηνύματος που θα ταξιδέψει στο δίκτυο. Με διπλό κλικ στην είσοδο PUBLISH προστίθεται ένα Input/BUTTON block, το οποίο και θα πατήσετε όταν θα είστε έτοιμοι να στείλετε το μήνυμα στο δίκτυο.

Για να διαβαστεί το μήνυμα αυτό, θα πρέπει να προστεθεί και το αντίστοιχο block. Αυτό είναι το block Network/SUBSCRIBE. Σε αυτό κάνουμε διπλό κλικ στην πρώτη έξοδο για να δημιουργηθεί ένα κουτί όπου θα απεικονιστεί το μήνυμα που στέλνεται μέσω του δικτύου.

Η ροή που πρέπει να έχετε μπροστά σας είναι αυτή που φαίνεται στην **Εικόνα 11.11**. Πρόκειται για μια ροή που αποστέλλει μη κρυπτογραφημένα μηνύματα μέσω ενός ipfs και στην οποία έχει δημιουργηθεί ο λογαριασμός του χρήστη *alice* για μελλοντική χρήση.

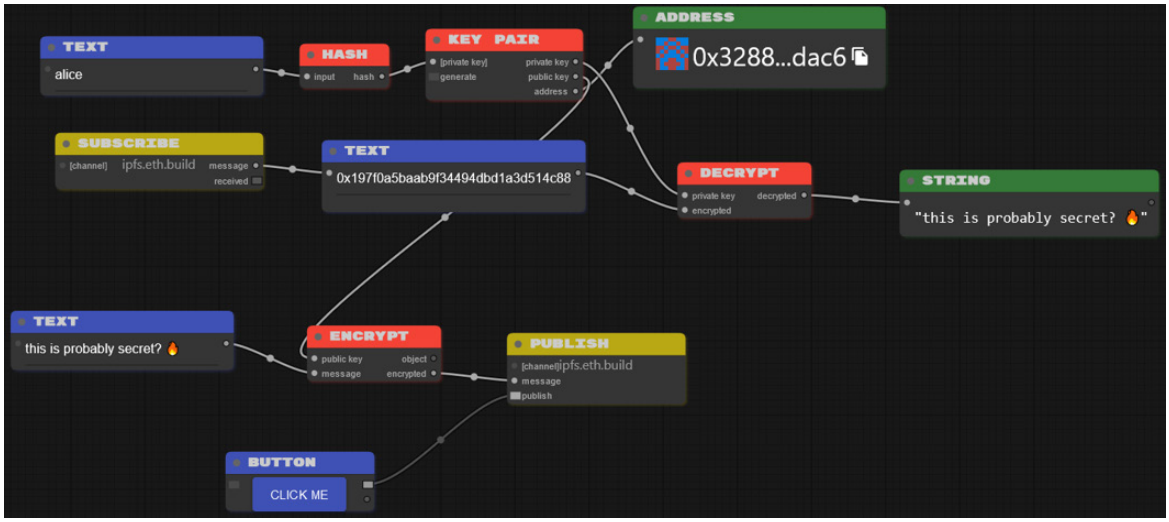


Εικόνα 11.11 *Η ροή στο Σενάριο 3 για την αποστολή ενός μη κρυπτογραφημένου μηνύματος μέσω του διαδικτύου (κανάλι ipfs).*

Στη ροή αυτή μπορείτε να συμπληρώσετε ένα μήνυμα στο block TEXT στην είσοδο του block PUBLISH και να πατήσετε το κουμπί CLICK ME για να ταξιδέψει το μήνυμα στο δίκτυο. Το block PUBLISH χρησιμοποιεί το default κανάλι που αναφέρθηκε. Είναι σημαντικό να γίνει αντιληπτό ότι η επικοινωνία αυτή είναι διαθέσιμη σε όλους όσοι ακούνε το κανάλι αυτό τη στιγμή εκείνη. Επομένως, όλοι όσοι έχουν φτιάξει την ομάδα blocks SUBSCRIBE και TEXT μπορούν να απεικονίσουν ό,τι περνά από το δίκτυο τη στιγμή εκείνη. Στην Εικόνα 11.10 το κείμενο που αποστέλλεται στο κανάλι ipfs είναι το *this is public*. Επίσης, φαίνεται και η λήψη του μηνύματος από το block SUBSCRIBE.

Το επόμενο βήμα είναι η αποστολή κρυπτογραφημένου μηνύματος στο ίδιο κανάλι. Από τη στιγμή που μπορεί ο οποιοσδήποτε να ακούσει το κανάλι αυτό, η αποστολή κρυπτογραφημένου μηνύματος μπορεί να βοηθήσει στο να μπορέσει να αποκωδικοποιηθεί μόνο από τον χρήστη προς τον οποίο και προορίζεται. Αυτό θα επιτευχθεί εφόσον η κωδικοποίηση του μηνύματος γίνει με το δημόσιο κλειδί του χρήστη. Τότε, θα απαιτείται γνώση του ιδιωτικού κλειδιού για την αποκρυπτογράφηση του μηνύματος στην πλευρά της λήψης (block SUBSCRIBE).

Η **Εικόνα 11.12** παρουσιάζει τις αλλαγές που πρέπει να γίνουν στη ροή που φαίνεται στην Εικόνα 11.10 για την αποστολή κρυπτογραφημένου μηνύματος από το κανάλι ipfs, που χρησιμοποιείται στο Σενάριο 3.



Εικόνα 11.12 *Η ροή στο Σενάριο 3 για την αποστολή ενός κρυπτογραφημένου μηνύματος μέσω του διαδικτύου (κανάλι ipfs).*

Όπως βλέπετε, στην Εικόνα 11.12 έχουν προστεθεί ορισμένα blocks για να αποδοθεί η κρυπτογράφηση του μηνύματος καθώς και η αποκρυπτογράφηση αυτού. Για την ακρίβεια:

- Έχει προστεθεί ένα block Crypto/ENCRYPT ανάμεσα στο μήνυμα (block TEXT) και στην αποστολή του (block PUBLISH).
- Στο block ENCRYPT η μία είσοδος είναι το μήνυμα και η άλλη το δημόσιο κλειδί του χρήστη στον οποίο απευθύνεται το μήνυμα. Στην περίπτωση της Εικόνας 11.11 η *alice* στέλνει μήνυμα στον εαυτό της, αλλά αυτό θα αλλάξει στην επέκταση.
- Παρατηρείτε ότι το μήνυμα που φαίνεται στο block TEXT μετά το block SUBSCRIBE δεν βγάζει νόημα, καθώς είναι κωδικοποιημένο. Το μήνυμα που λαμβάνεται απεικονίζεται ως: *0x197f0a5baab9f34494dbd1a3d514c88a03f3c39b075ea4eeace3b10bbf1d4ceb4b8425338486f44648f7b97c764870be18788e5ef85c3494d62f0d237cf5e353a19ae77438bf9e61652abc4748d977506830ac705d3080217ae80ecc9728ec583489663a592ca2b30959c76966ded4faa6*
- Για την αποκρυπτογράφηση του μηνύματος, αυτό πηγαίνει ως είσοδος σε ένα block DECRYPT που δέχεται και το ιδιωτικό κλειδί του χρήστη για την αποκρυπτογράφηση.
- Το αποκρυπτογραφημένο μήνυμα φαίνεται στο block STRING, που δίνεται ως έξοδος του block DECRYPT.

Ο κώδικας της ροής για την αποστολή κωδικοποιημένων αλλά και μη κωδικοποιημένων μηνυμάτων φαίνεται στον φάκελο Σενάριο 3.

Επέκταση με χρήση δύο χρηστών: Το σενάριο που παρουσιάστηκε μπορεί να επεκταθεί με την προσθήκη ενός δεύτερου χρήστη. Ο χρήστης αυτός μπορεί να δημιουργηθεί στην ίδια επιφάνεια εργασίας με πριν. Διαφορετικά, είναι δυνατόν να ανοίξετε το πρόγραμμα σε άλλο παράθυρο (ή tab) του browser σας και να δημιουργήσετε εκεί μια νέα ροή (ακολουθώντας τα βήματα παραπάνω) απλώς με νέο ιδιωτικό κλειδί για τον νέο χρήστη. Στη συνέχεια θα θεωρηθεί ότι τα ιδιωτικά κλειδιά των δύο χρηστών είναι *alice* και *bob*, για ευκολία.

Οι αλλαγές που πρέπει να γίνουν στην περίπτωση δύο χρηστών αφορούν αποκλειστικά την περίπτωση αποστολής κρυπτογραφημένων μηνυμάτων. Έτσι, στην περίπτωση αυτή θα πρέπει:

- Ο χρήστης που στέλνει το μήνυμα θα πρέπει να γνωρίζει το δημόσιο κλειδί του άλλου χρήστη και να το περάσει ως είσοδο (μαζί με το μήνυμα) στο block ENCRYPT.
- Ο χρήστης που λαμβάνει το μήνυμα θα χρησιμοποιήσει το ιδιωτικό του κλειδί (όπως και πριν δηλαδή) για την αποκρυπτογράφησή του.

Όπως γίνεται αντιληπτό, θα μπορούσαν οι δύο χρήστες να μην ήταν καν στον ίδιο χώρο. Βρείτε έναν φίλο σας και, ο καθένας από το σπίτι του, στείλτε μηνύματα μέσω του ETH.Build. Αφού ανταλλάξετε τα δημόσια κλειδιά σας!

11.1.5 Σενάριο 4: Συναλλαγές

Βασικά στοιχεία σεναρίου:

Σενάριο 4	
Περιγραφή	Το σενάριο αυτό ασχολείται
Blocks που χρησιμοποιούνται	• Crypto/Key Pair • Crypto/Hash • Display/Watch (STRING) • Display/Address • Display/Dollars • Input/Text • Input/Button • Web3/Balance • Web3/Transaction • Web3/Send Tx • Unit/From Wei • Unit/To Wei
Ενότητες του βιβλίου που καλύπτονται	Κεφάλαια 1, 2, 3 και 4.

Πίνακας 11.4 *Περιγραφή του Σεναρίου 4.*

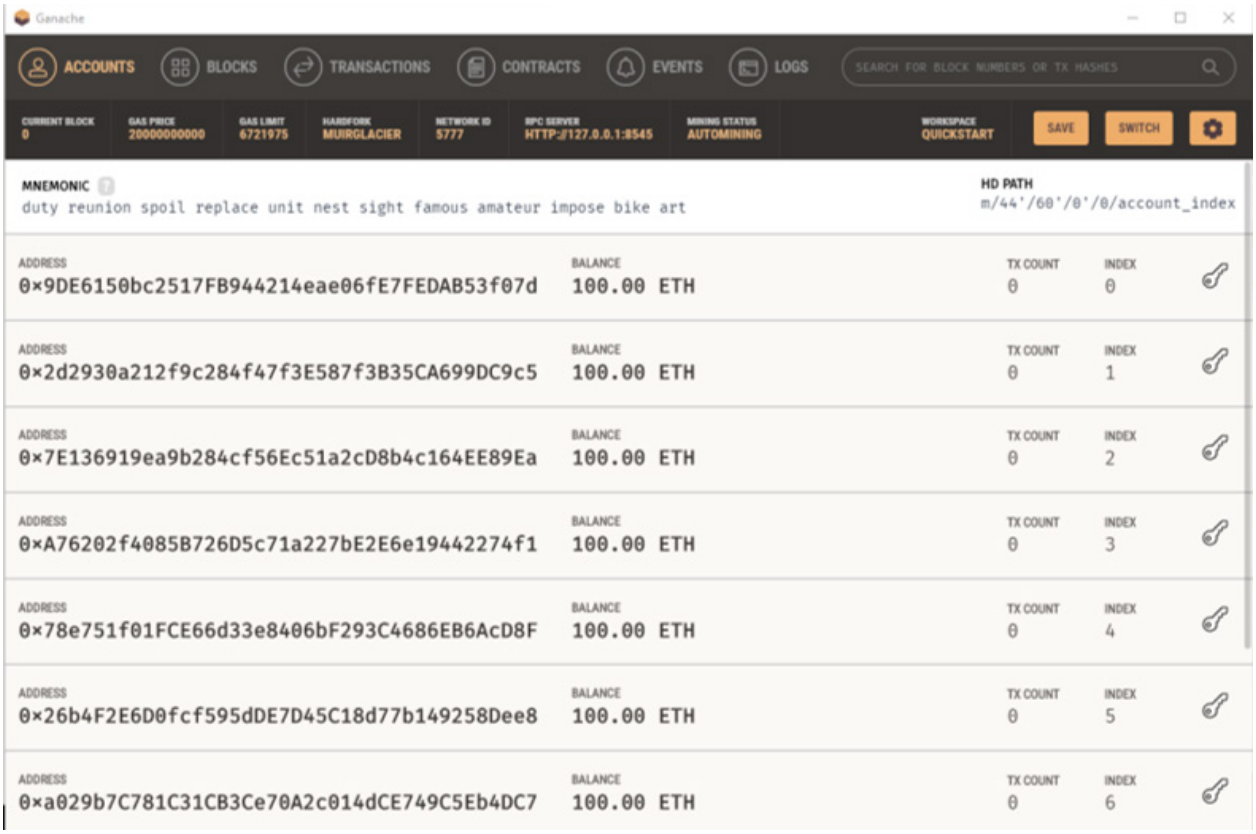
Το σενάριο αυτό ασχολείται με τη δημιουργία συναλλαγών σε ένα δίκτυο blockchain. Για να μπορέσει να τρέξει το σενάριο, θα πρέπει να συνδεθεί το ETH.Build με ένα δίκτυο blockchain, το οποίο θα δημιουργήσετε τοπικά στον υπολογιστή σας και το οποίο θα συνδεθεί με το ETH.Build.

Πιο αναλυτικά, το Σενάριο 4 περιλαμβάνει:

- τη δημιουργία ενός τοπικού δικτύου blockchain με τη βοήθεια του Ganache, μέλος της σουίτας εφαρμογών Truffle[134] (δείτε Κεφάλαιο 8),
- τη σύνδεση του τοπικού δικτύου με το ETH.Build,
- τη δημιουργία συναλλαγών στο τοπικό δίκτυο και έλεγχος των υπολοίπων στους εμπλεκόμενους λογαριασμούς για επιβεβαίωση της επιτυχούς αποστολής,
- για τη συναλλαγή θα χρειαστεί να γίνει μετατροπή του ποσού από wei (λήψη) και σε wei (αποστολή) (δείτε Κεφάλαια 2 και 4).

Ξεκινώντας το Σενάριο 4, πρώτα θα περιγραφούν τα βήματα για την εγκατάσταση του προγράμματος και τη δημιουργία του τοπικού δικτύου blockchain. Το πρόγραμμα είναι το *Ganache*, μέλος της σουίτας της *Truffle*. Για να το κατεβάσετε, θα επισκεφθείτε τη σελίδα χρησιμοποιώντας το link στην αναφορά στο κάτω μέρος. Κατόπιν επιλέγετε στη σελίδα το κουμπί LEARN MORE κάτω από το Ganache. Στη σελίδα που ανοίγει επιλέγετε να κατεβάσετε το πρόγραμμα (αν δεν θέλετε Windows, επιλέγετε Λειτουργικό Σύστημα). Στη συνέχεια προχωρήστε στην εγκατάσταση του προγράμματος.

Για το Ganache θα χρειαστεί να γνωρίζετε ότι, μόλις ανοίξετε την εφαρμογή, αυτή δημιουργεί έναν ιδιωτικό κόμβο Ethereum στο τοπικό μηχάνημα που τρέχει το Ganache. Επιπλέον, δημιουργούνται 10 λογαριασμοί χρηστών, που είναι ενσωματωμένοι στον κόμβο. Ο κάθε λογαριασμός περιέχει αρχικά 100 ethers, που δεν έχουν καμία πραγματική αξία. Ένα πλεονέκτημα του Ganache είναι ότι διευκολύνει τη διασύνδεση με το Metamask (δείτε Παράρτημα Α). Επίσης, στο Παράρτημα Α θα δείτε πώς μπορείτε να αλλάξετε την πόρτα του RPC server σε 8545 για το παράδειγμα. Ο λόγος έχει να κάνει ότι διευκολύνει την επικοινωνία με το Metamask (εφόσον αυτό χρειαστεί).



Εικόνα 11.13 *Η εισαγωγική εικόνα στο Ganache.*

[134] Online σύνδεσμος: https://trufflesuite.com/

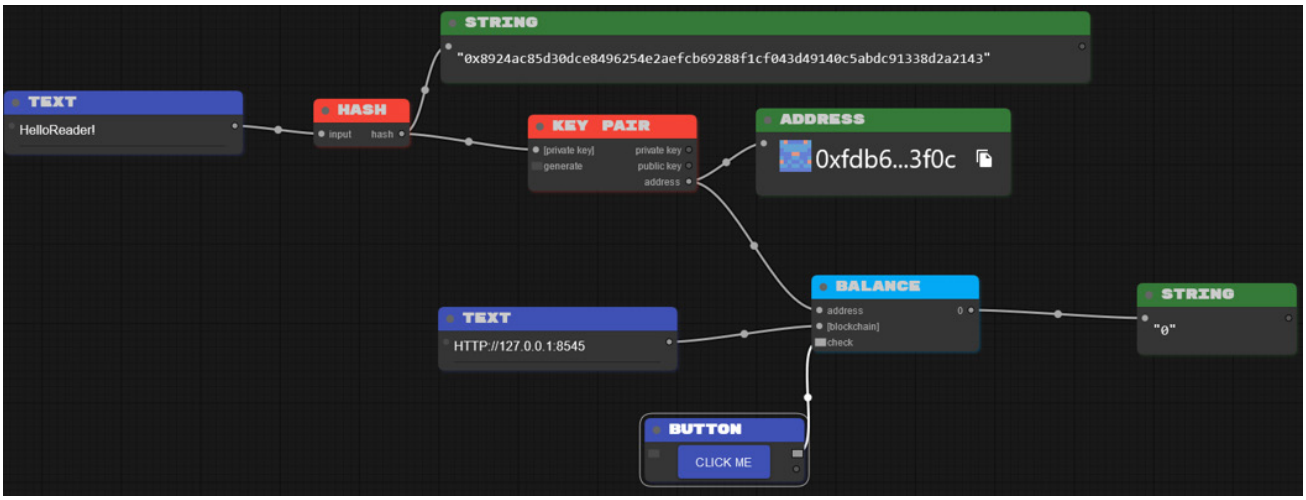
Αφού εγκαταστήσετε το Ganache, τρέχετε το πρόγραμμα και επιλέγετε Quickstart. Η αρχική σας εικόνα είναι αυτή που φαίνεται στην **Εικόνα 11.13**. Εκεί φαίνονται οι διευθύνσεις των λογαριασμών (Addresses), τα ποσά που έχει ο καθένας μέσα (100.00 ETH) και στο πάνω μέρος τα χαρακτηριστικά του δικτύου. Σε αυτά προσοχή χρειάζεται στην (τοπική) διεύθυνση του RPC Server, που θα τη χρειαστείτε στο ETH.Build (αλλά και για να συνδεθεί το Metamask με το Ganache). Μπορείτε, επίσης, να πατήσετε και στην εικόνα κλειδιού σε κάθε διεύθυνση για να δείτε το ιδιωτικό κλειδί που αντιστοιχεί στη διεύθυνση αυτή.

Έχοντας πλέον έτοιμο το τοπικό δίκτυο του blockchain, μπορείτε να επιστρέψετε στο ETH.Build. Εκεί, σε μια κενή επιφάνεια εργασίας δημιουργείτε έναν λογαριασμό χρήστη με τη χρήση των blocks KEY PAIR, HASH, TEXT, STRING και ADDRESS. Η ροή φαίνεται στην **Εικόνα 11.14**.

Εικόνα 11.14 *Η αρχική ροή δημιουργίας χρήστη με τη διεύθυνση του για σταθερή λέξη ως είσοδος στη συνάρτηση HASH.*

Στην Εικόνα 11.14 χρησιμοποιείται η λέξη HelloReader! για να παραγάγει το ιδιωτικό κλειδί και, τελικά, τη διεύθυνση του χρήστη, η οποία φαίνεται στο block ADDRESS.

Στη συνέχεια θα προσθέσουμε τα απαραίτητα blocks για να συνδεθεί ο χρήστης στο τοπικό δίκτυο που δημιουργεί το Ganache και να ελεγχθεί το ποσό των ETH που έχει στον λογαριασμό του. Για να γίνει αυτό, κεντρικό ρόλο παίζει το block Web3/BALANCE. Στην Εικόνα 11.15 βλέπετε τη συνέχεια της ροής με την εισαγωγή των blocks για τον έλεγχο του υπολοίπου στον λογαριασμό του χρήστη.



Εικόνα 11.15 *Η συνέχεια της ροής με σύνδεση με το Ganache για τον έλεγχο του υπολοίπου.*

Το block BALANCE έχει 3 εισόδους: η 1η είσοδος θα πάρει τη διεύθυνση του χρήστη (address) από το block KEY PAIR, στη 2η είσοδο με διπλό κλικ εμφανίζεται το block TEXT, στο οποίο και προσθέτετε την (τοπική) διεύθυνση του RPC server (HTTP://127.0.0.1:8545) και στην 3η είσοδο κάνετε διπλό κλικ για να εμφανιστεί το block BUTTON, το οποίο θα πατήσετε για την επικοινωνία με το δίκτυο που δίνεται στο TEXT. Τέλος, έχει μία έξοδο που δείχνει τι ποσό υπάρχει μέσα στη διεύθυνση αυτή.

Όπως βλέπετε στην Εικόνα 11.4, το υπόλοιπο στη διεύθυνση είναι 0 γιατί δεν υπάρχει η διεύθυνση αυτή μέσα στο Ganache. Αντιθέτως, αν κάνουμε κάποιες αλλαγές έτσι ώστε να μπει στον λογαριασμό η μία διεύθυνση από το Ganache, τότε θα πρέπει να βλέπετε στο υπόλοιπο την τιμή των 100 ETH, που φαίνονται και στο Ganache. Για να προχωρήσουν οι αλλαγές, θα πρέπει να επιλεγεί μια διεύθυνση από το Ganache και να αντιγραφεί το ιδιωτικό κλειδί αυτής.

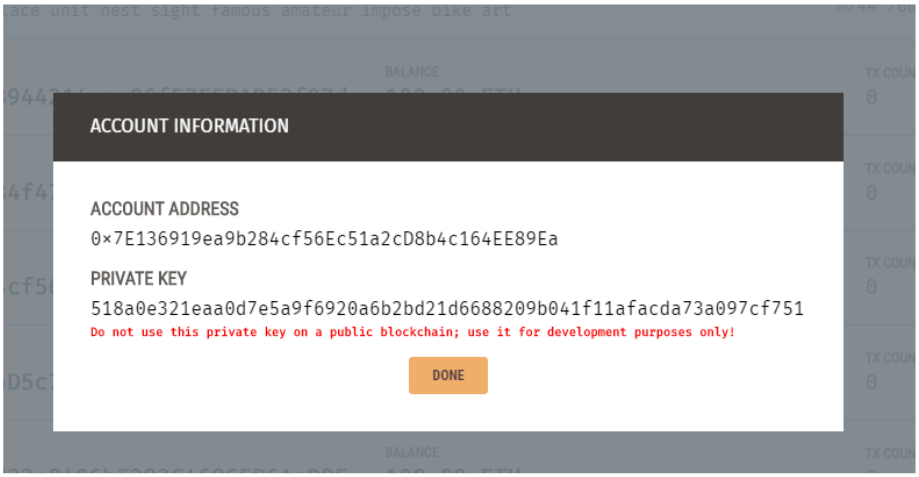
Η διεύθυνση που εισάγεται είναι αυτή με Index 2 (δείτε και Εικόνα 11.13):

```
0x7E136919ea9b284cf56Ec51a2cD8b4c164EE89Ea
```

Για την ακρίβεια, εισάγεται το ιδιωτικό κλειδί της πατώντας πάνω στο αντίστοιχο εικονίδιο:

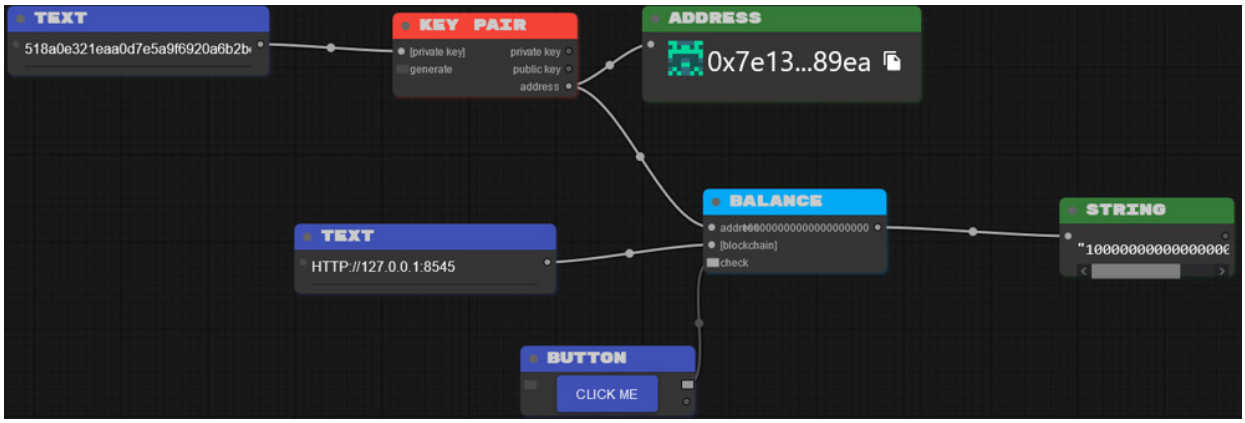
```
518a0e321eaa0d7e5a9f6920a6b2bd21d6688209b041f11afacda73a097cf751
```

Η **Εικόνα 11.16** δείχνει στο Ganache τις τιμές του ιδιωτικού κλειδιού για την επιλεγμένη διεύθυνση.



Εικόνα 11.16 *Το ιδιωτικό κλειδί για την επιλεγμένη διεύθυνση στο Ganache.*

Οι αλλαγές που πρέπει να γίνουν στη ροή στο ETH.Build για να εισαχθεί το παραπάνω ιδιωτικό κλειδί φαίνονται στην **Εικόνα 11.17**.

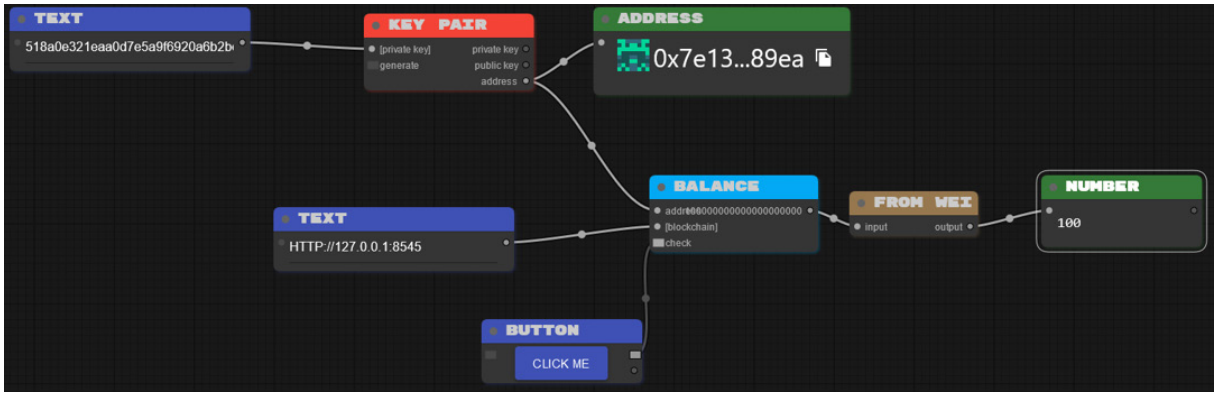


Εικόνα 11.17 *Η συνέχεια της ροή με σύνδεση με το Ganache για τον έλεγχο του υπολοίπου.*

Στην πράξη έχει αφαιρεθεί το block της συνάρτησης HASH και έχει συνδεθεί απευθείας το block KEY PAIR με το block TEXT, που έχει τώρα μέσα την τιμή του κλειδιού. Παρατηρήστε ότι οι αλλαγές αντιστοιχούν σε νέα διεύθυνση στο δίκτυο, η οποία είναι *ίδια* με τη διεύθυνση στο Ganache! Κατόπιν μπορείτε να δείτε ότι η τιμή που φαίνεται ως υπόλοιπο είναι ιδιαίτερα μεγάλη. Ο λόγος είναι ότι αυτή αποδίδεται σε wei (δείτε Κεφάλαια 2 και 4 σχετικά με ETH και wei).

Για τη μετατροπή των wei σε ETH θα χρειαστεί να προστεθεί 1 block ανάμεσα στα blocks BALANCE και STRING. Το block αυτό είναι το Web3/FROM WEI. Όπως βλέπετε, όταν προστεθεί, τότε το αποτέλεσμα είναι 100 ETH, όπως γνωρίζετε και από το Ganache.

Στην **Εικόνα 11.18** μπορείτε να δείτε τη ροή με την προσθήκη του block για τη μετατροπή.



Εικόνα 11.18 *Η ροή με τη μετατροπή του υπολοίπου από wei σε ETH.*

Στη συνέχεια θα δημιουργήσετε την πρώτη σας συναλλαγή. Για να το πετύχετε, θα χρειαστείτε το block Web3/ TRANSACTION. Πρόκειται για ένα block που δέχεται 9 εισόδους και βγάζει 4 εξόδους. Από τις εισόδους θα χρειαστούμε τα βασικά στοιχεία για μια συναλλαγή. Δηλαδή από ποια διεύθυνση (private key), σε ποια διεύθυνση (to) και ποιο ποσό (value).

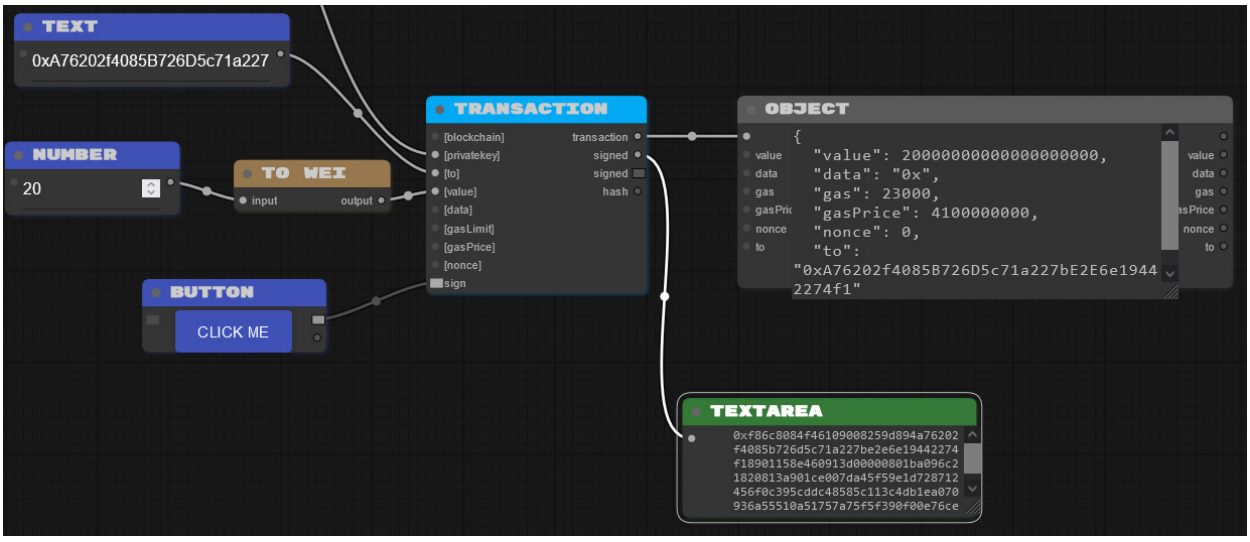
Η είσοδος Private key θα συνδεθεί με την αντίστοιχη έξοδο του KEY PAIR παραπάνω. Για την είσοδο το θα επιλεχθεί μια διεύθυνση από το Ganache. Έστω ότι επιλέγεται η 4η διεύθυνση (Index 3) με τιμή:

`0xA76202f4085B726D5c71a227bE2E6e19442274f1`

Για την τιμή (value) κάνουμε διπλό κλικ στην είσοδο και εμφανίζεται το block NUMBER. Επειδή ο αριθμός που θα μπει εκεί πρέπει να είναι σε wei, παρεμβάλετε ανάμεσα στο block NUMBER και στο block TRANSACTION το block Web3/TO WEI. Στη συνέχεια, στο block NUMBER συμπληρώνετε το ποσό (όπως βλέπετε στην **Εικόνα 11.19** – προτείνεται ενδεικτικά το 20) και κάνετε διπλό κλικ στην είσοδο sign για να προχωρήσει η συναλλαγή.

Συνοψίζοντας, η συναλλαγή γίνεται από τον χρήστη με διεύθυνση 0x7E136919ea9b284cf56Ec51a2cD8b4 c164EE89Ea προς τον χρήστη με διεύθυνση 0xA76202f4085B726D5c71a227bE2E6e19442274f1 και γίνεται μεταφορά 20 ETH.

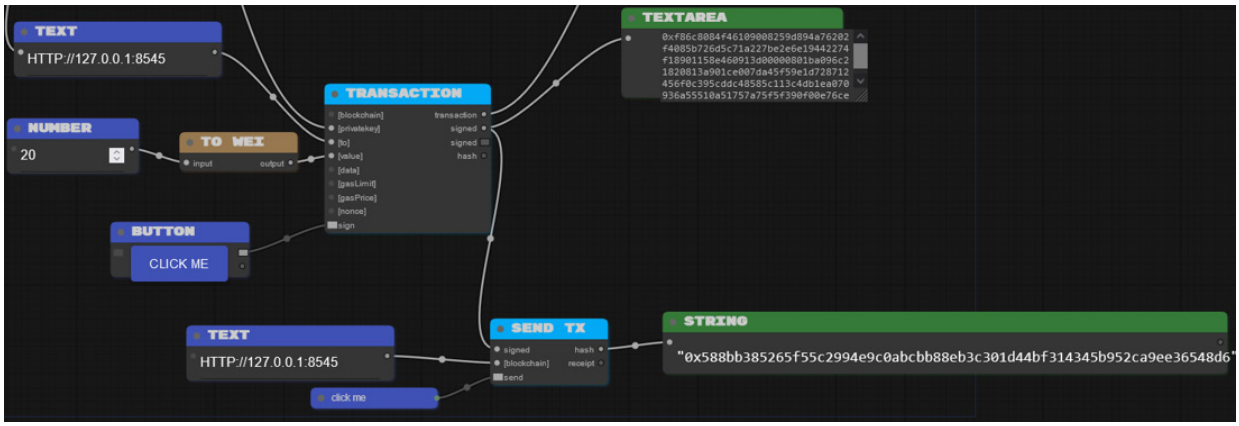
Κατόπιν, στην έξοδο πατάτε τη 2η έξοδο (signed) για να δείτε την κωδικοποιημένη μορφή της συναλλαγής. Επιπλέον, μπορείτε να πατήσετε και στην 1η έξοδο transaction για να δείτε τη δομή της συναλλαγής. Εκεί θα παρατηρήσετε ότι οι τιμές για τα gas, gasPrice, nonce έχουν αποδοθεί αυτόματα.



Εικόνα 11.19 *Οι προσθήκες στη ροή που χρειάζονται για τη δημιουργία μιας συναλλαγής.*

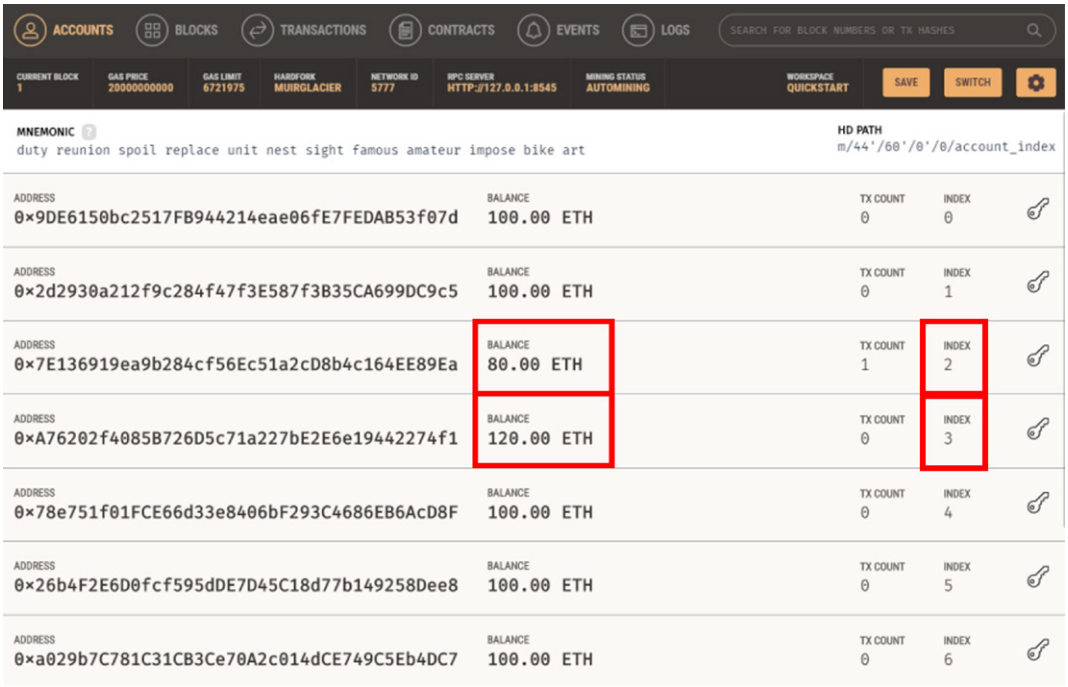
Η συναλλαγή αυτή δημιουργήθηκε στο ETH.Build, αλλά προς το παρόν δεν έχει αποσταλεί σε κάποιο δίκτυο blockchain για να την επεξεργαστεί και να ενημερώσει τα υπόλοιπα των εμπλεκόμενων διευθύνσεων. Αυτό το βήμα ολοκληρώνεται με τη βοήθεια ενός άλλου block, του Web3/SEND TX. Πρόκειται για ένα block το οποίο έχει 3 εισόδους και 2 εξόδους. Στην 1η είσοδο (signed) θα πρέπει να συνδεθεί η αντίστοιχη (2η) έξοδος του block TRANSACTION. Κατόπιν, στη 2η είσοδο θα μπει η διεύθυνση του RPC server που φαίνεται στο Ganache (όπως χρησιμοποιήθηκε και στο BALANCE). Η 3η είσοδος εισάγει τον timer για να μπει η συναλλαγή μέσα στο δίκτυο. Από την άλλη, μπορείτε να κάνετε διπλό κλικ στην 1η έξοδο (hash) για να δείτε το κρυπτογραφημένο αποτέλεσμά της.

Η Εικόνα **11.20** δείχνει την ολοκλήρωση της ροής στην πλατφόρμα του ETH.Build.



Εικόνα 11.20 *Οι προσθήκες στη ροή που χρειάζονται για τη δημιουργία μιας συναλλαγής.*

Αφού πατήσετε το click me (είσοδος SEND TX), μπορείτε να επιστρέψετε στο Ganache για να δείτε την αλλαγή στο υπόλοιπο των διευθύνσεων. Θα πρέπει η διεύθυνση του αποστολέα (index 2) να έχει 80 ETH (100-20) ενώ η διεύθυνση του παραλήπτη (index 3) να έχει 120 ETH (100+20). Η **Εικόνα 11.21** δείχνει αυτή ακριβώς την αλλαγή μέσα στο Ganache.

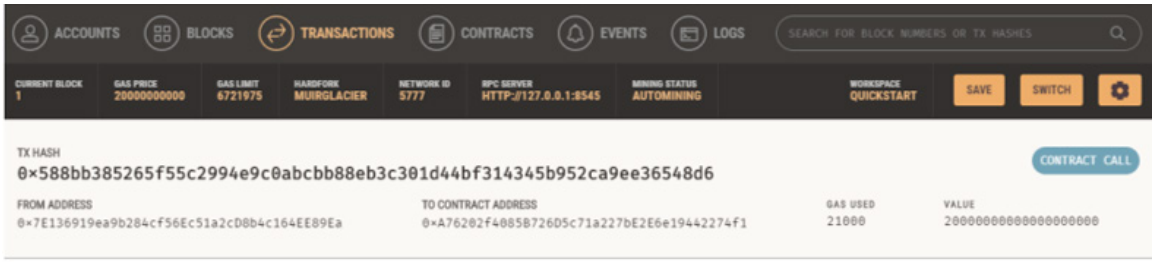


Εικόνα 11.21 *Η εκτέλεση της συναλλαγής στο Ganache.*

Επιπλέον, στο Ganache μπορείτε να επιλέξετε από το menu στο επάνω μέρος την επιλογή Transactions. Εκεί θα εμφανιστεί η συναλλαγή που μόλις πραγματοποιήσατε στο ETH.Build. Μπορείτε, μάλιστα, να επιβεβαιώσετε

ότι είναι αυτή συγκρίνοντας το hash που βλέπετε στο Ganache με το αντίστοιχο hash στο ETH.Build, στην έξοδο του block SEND TX.

Στην **Εικόνα 11.22** βλέπετε τη συναλλαγή όπως έχει καταγραφεί στο Ganache. Στην Εικόνα 11.20 φαίνεται και το hash της συναλλαγής για επιβεβαίωση. Επιβεβαιώνονται επίσης και οι διευθύνσεις.



Εικόνα 11.22 *Η απεικόνιση της συναλλαγής στο Ganache και η επιβεβαίωση του hash αυτής.*

Ο κώδικας όλης της ροής βρίσκεται στο αποθετήριο στον φάκελο Σενάριο 4.

Με το Σενάριο αυτό ολοκληρώνεται το πρακτικό κομμάτι του βιβλίου. Σε αυτό έγινε μια προσπάθεια να απεικονιστούν ορισμένα παραδείγματα που βοηθούν στην κατανόηση των πρώτων θεωρητικών κεφαλαίων (κυρίως των 1, 2, 3 και 4).

Οι αναγνώστες καλούνται να δοκιμάσουν να υλοποιήσουν τις ροές και να πειραματιστούν με το αποτέλεσμα της καθεμίας. Επιπλέον, μπορούν να επαναλάβουν τα αντίστοιχα θεωρητικά κεφάλαια για να είναι σίγουροι ότι έχουν κατανοήσει σε βάθος τις έννοιες που παρουσιάζονται.

Βιβλιογραφία

ETH (2022). A. Griffith. *ETH.Build: An Educational Sandbox For Web3… And Much More*. Online πηγή: https://eth.build/ [Τελευταία πρόσβαση: Δεκέμβριος 2022].

ΠΑΡΑΡΤΗΜΑ

Δημιουργία Λογαριασμού στο Πορτοφόλι Metamask

Σύνοψη

Στο Παράρτημα δίνονται οδηγίες για τη δημιουργία λογαριασμού στο Πορτοφόλι του Metamask για να μπορέσετε να το χρησιμοποιήσετε στα παραδείγματα του βιβλίου ή και για προσωπική σας χρήση.

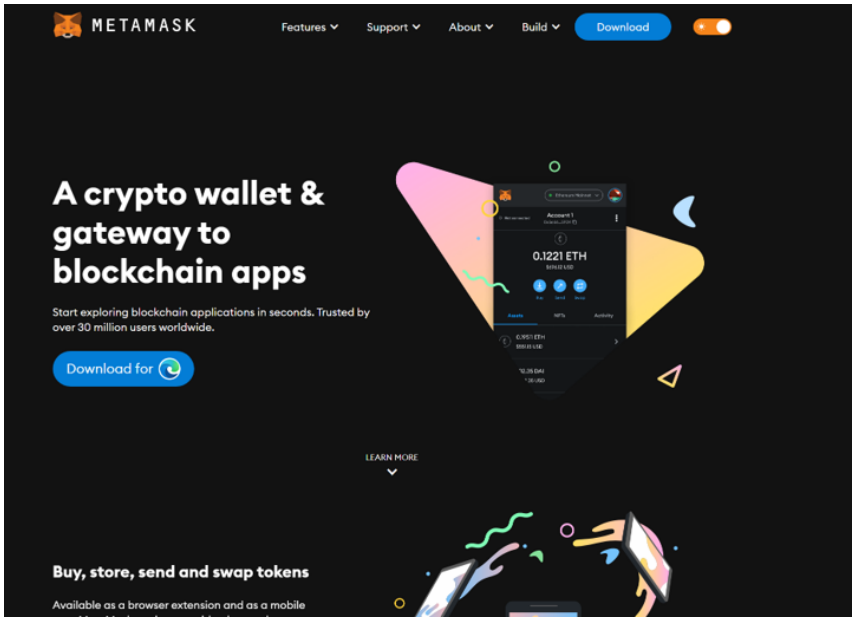
Προαπαιτούμενη γνώση

Καμία.

Π.1 Δημιουργώντας λογαριασμό στο Metamask

Όπως παρουσιάστηκε στο βιβλίο, το Metamask είναι ένα δημοφιλές πορτοφόλι το οποίο διευκολύνει τη σύνδεση με το δίκτυο του Ethereum. Η διευκόλυνση προκύπτει από το γεγονός ότι έχει προεγκατεστημένες τις απαραίτητες ρυθμίσεις για τη σύνδεση στο κυρίως δίκτυο αλλά και στα δοκιμαστικά δίκτυα του Ethereum. Αυτό όμως δεν σημαίνει ότι δεν μπορεί να χρησιμοποιηθεί για σύνδεση σε άλλα (custom) δίκτυα.

Το Metamask είναι ένα web wallet το οποίο μπορεί να εγκατασταθεί ως επέκταση σε όλους τους νέους browsers. Στη συνέχεια δίνονται οδηγίες για την εγκατάσταση του πορτοφολιού του Metamask τοπικά στον Edge browser. Τα βήματα είναι παρόμοια σε όλους τους browsers (Chrome, Firefox, Edge), επομένως ο καθένας μπορεί να επιλέξει ελεύθερα τον browser της αρεσκείας του και να παρακολουθήσει τα βήματα που παρουσιάζονται στο Παράρτημα.

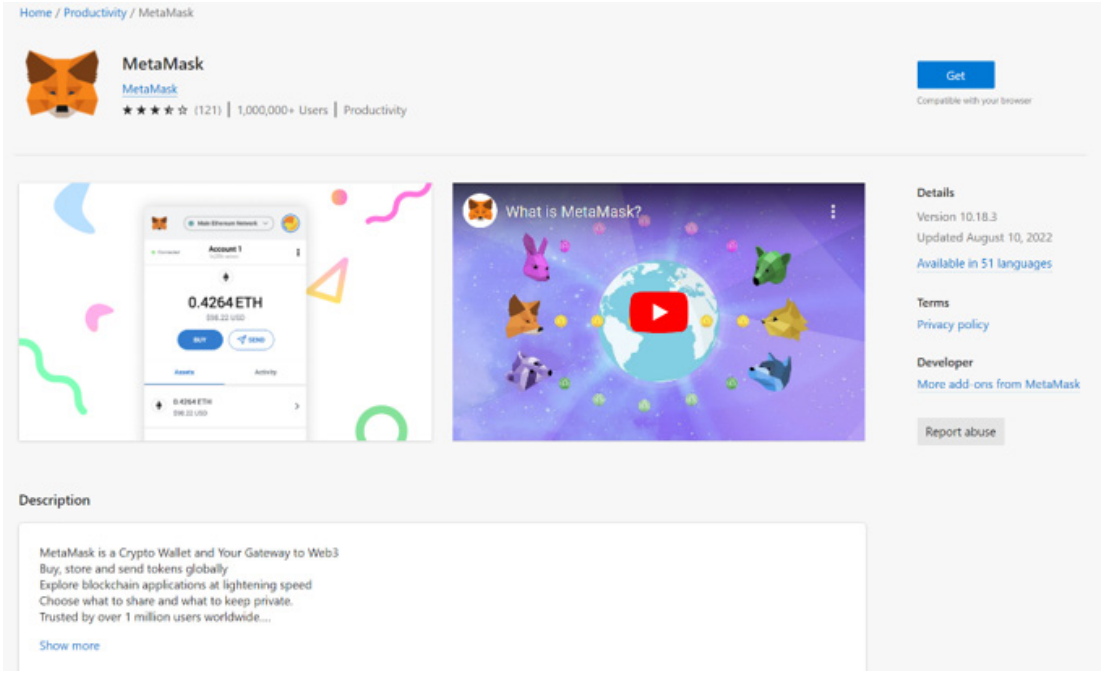


Εικόνα Π.1 *Η αρχική σελίδα για την εγκατάσταση του Metamask.*

Αρχικά θα πρέπει να επισκεφθείτε τη σελίδα του Metamask[135], μέσω του browser στον οποίο θα κάνετε την εγκατάσταση. Στην **Εικόνα Π.1** βλέπετε την αρχική σελίδα. Ανάλογα με τον browser που θα χρησιμοποιήσετε για την επίσκεψή σας, θα σας προτείνεται και η εγκατάσταση σε αυτόν. Προσέξτε να επιβεβαιώσετε την ιστοσελίδα ότι είναι σωστή, καθώς έχουν παρατηρηθεί διάφορες προσπάθειες απάτης.

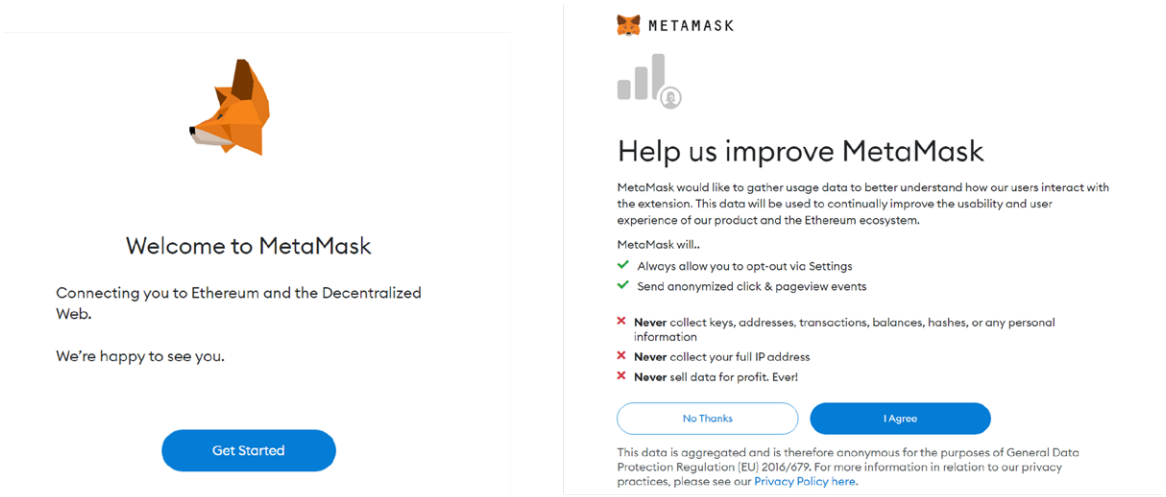
[135] Online σύνδεσμος: https://metamask.io

Κατόπιν πατάτε στο κουμπί για το κατέβασμα της εφαρμογής και στο νέο παράθυρο που σας ανοίγει επιλέγετε την εγκατάσταση.



Εικόνα Π.2 *Εγκαθιστώντας την επέκταση του Metamask (Βήμα 1).*

Μόλις ολοκληρωθεί η εγκατάσταση, θα σας ανοίξει ένα νέο παράθυρο για τη δημιουργία λογαριασμού στην εφαρμογή.

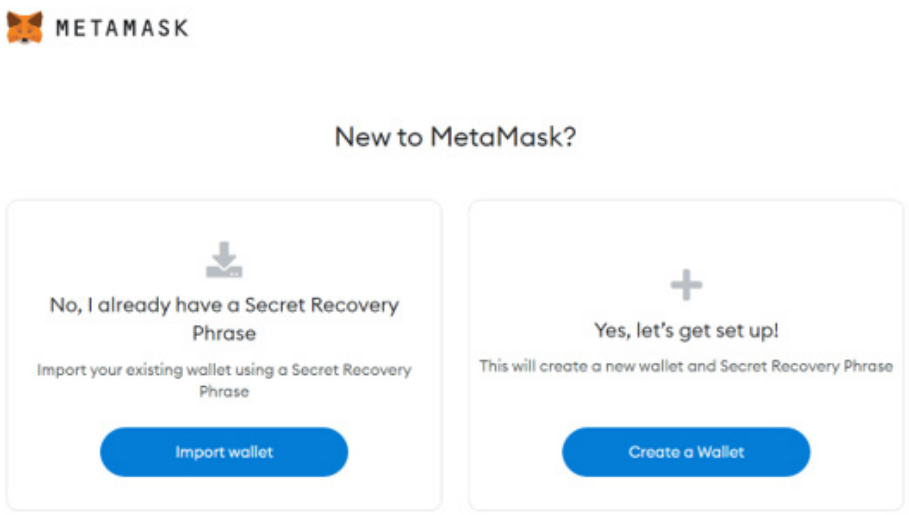


Εικόνα Π.3 *Εγκαθιστώντας την επέκταση του Metamask (Βήμα 2).*

Από εκεί επιλέγετε να ξεκινήσετε και δίνετε τη συγκατάθεσή σας (ή όχι) για τη συλλογή δεδομένων για τη βελτίωση της εφαρμογής.

Στην επόμενη σελίδα πρέπει να επιλέξετε αν θα δημιουργήσετε ένα νέο wallet ή θα προχωρήσετε με τη μεταφορά ενός υπάρχοντος που είναι εγκατεστημένο σε άλλον υπολογιστή. Εδώ θα καλυφθεί η πρώτη περίπτωση. Οδηγίες για τη δεύτερη επιλογή θα δοθούν κατά τη διάρκεια της περιγραφής της πρώτης.

Επιλέγετε, επομένως, *Create a Wallet*.

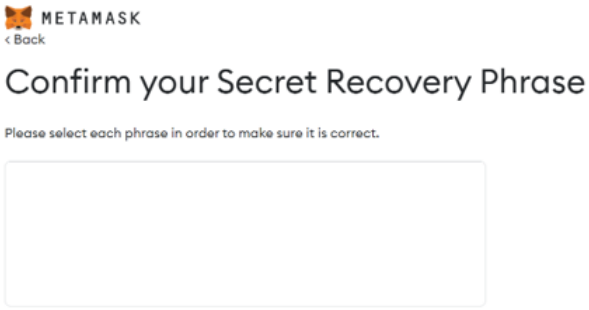


Εικόνα Π.4 *Εγκαθιστώντας την επέκταση του Metamask (Βήμα 3).*

Κατόπιν καλείστε να δώσετε και να επιβεβαιώσετε έναν κωδικό για την είσοδό σας στην εφαρμογή και να αποδεχθείτε τους όρους χρήσης. Ακολουθεί ένα σύντομο video σχετικά με τις 12 λέξεις που διασφαλίζουν την ακεραιότητα του πορτοφολιού σας και οι οποίες ορίζουν το ιδιωτικό κλειδί σας, όπως παρουσιάστηκε στο Κεφάλαιο 3.

Μετά την παρακολούθηση του video προχωράτε στην επόμενη σελίδα, στην οποία σας εμφανίζονται οι 12 λέξεις. Θυμηθείτε ότι αυτές πρέπει να καταγραφούν και να φυλαχθούν. Μπορούν να χρησιμοποιηθούν αν στο Βήμα 3 (**Εικόνα Π.4**) επιλέξετε την άλλη περίπτωση *Import Wallet*. Θα τις χρησιμοποιήσετε σίγουρα στο επόμενο βήμα, οπότε τις αντιγράφετε (ή τις κατεβάζετε) και προχωράτε.

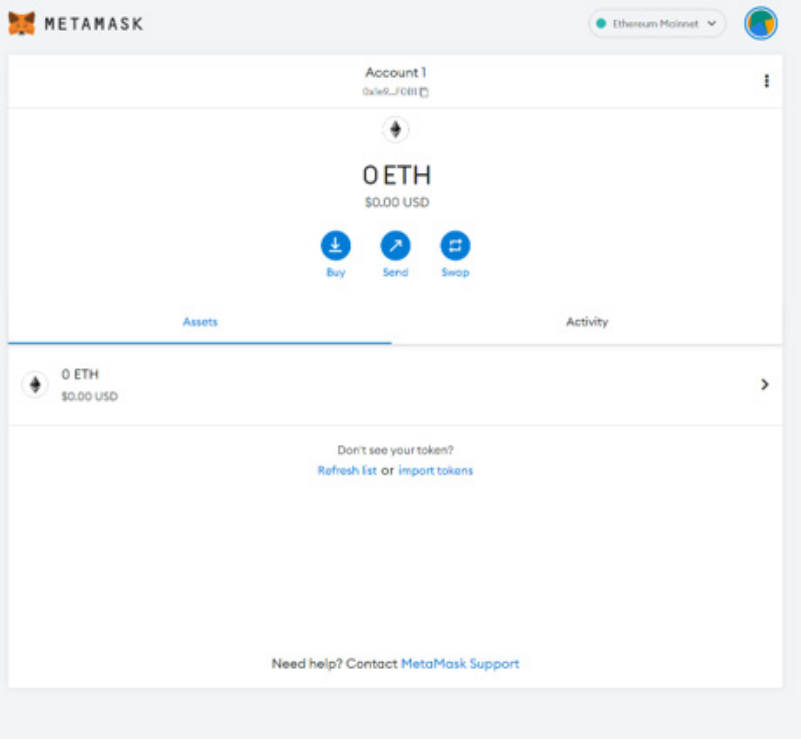
Στην επόμενη σελίδα θα σας ζητηθεί να περάσετε τις λέξεις αυτές με την ακριβή σειρά με την οποία σας έχουν δοθεί (**Εικόνα Π.5**).



Εικόνα Π.5 *Εγκαθιστώντας την επέκταση του Metamask (Βήμα 4).*

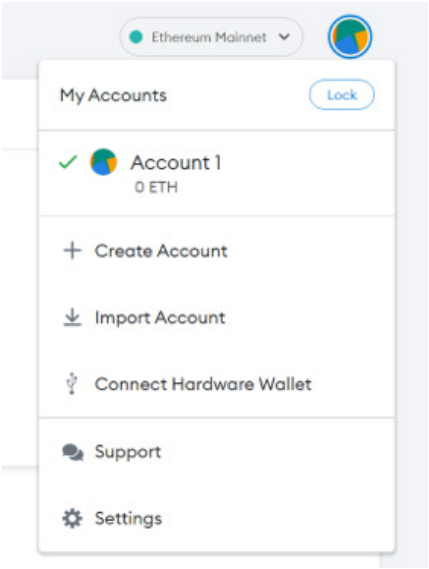
Με την επιτυχή απόδοση των λέξεων ολοκληρώνεται η διαδικασία δημιουργίας του πορτοφολιού.

Στη συνέχεια εισέρχεστε στην οθόνη όπου φαίνεται το πορτοφόλι σας. Θα διαπιστώσετε με την είσοδό σας σε αυτό ότι είναι ήδη συνδεδεμένο με το κυρίως δίκτυο του Ethereum, όπως φαίνεται και πάνω δεξιά στην **Εικόνα Π.6**.



Εικόνα Π.6 *Η αρχική σελίδα του Metamask με ενεργή σύνδεση με το Ethereum Mainnet.*

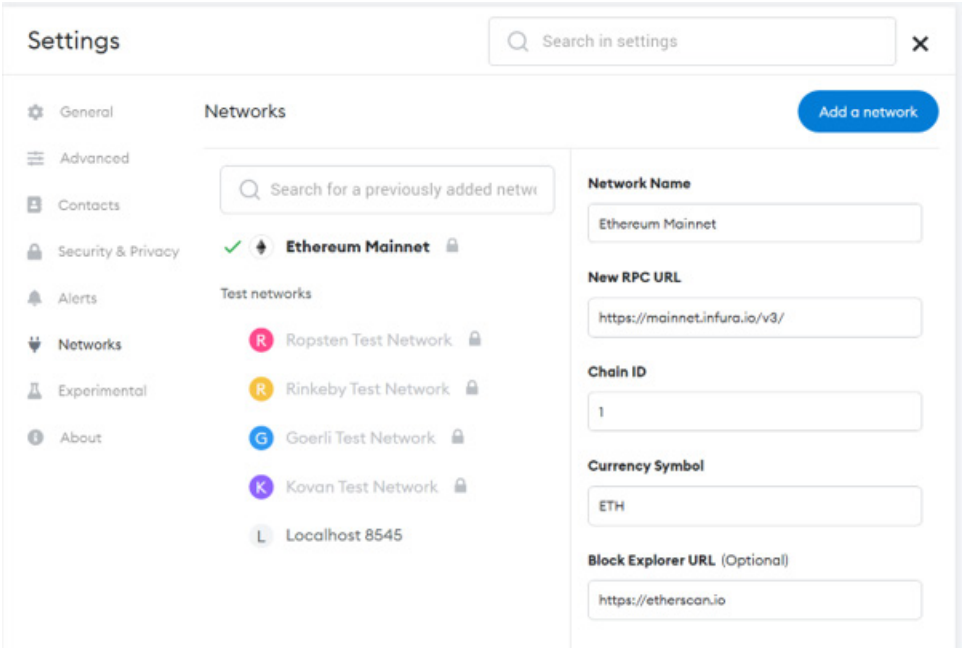
Πατώντας πάνω δεξιά στο εικονίδιο (**Εικόνα Π.7**), μπορείτε να δείτε τους λογαριασμούς σας, να δημιουργήσετε νέο λογαριασμό ή να εισαγάγετε έναν δικό σας. Επίσης, μπορείτε να συνδέσετε ένα hardware wallet. Στο κάτω μέρος είναι οι ρυθμίσεις.



Εικόνα Π.7 *Επιλογές στην αρχική σελίδα του Metamask.*

Πατώντας στις ρυθμίσεις (Settings), έχετε το σύνολο των ρυθμίσεων της εφαρμογής. Ιδιαίτερο ενδιαφέρον έχει η επιλογή Networks, στην οποία μπορείτε να επιλέξετε το δίκτυο με το οποίο θα συνδεθεί ο λογαριασμός σας και στο οποίο θα προωθηθεί η όποια συναλλαγή κάνετε με το πορτοφόλι.

Στην **Εικόνα Π.8** βλέπετε τις επιλογές. Προσέξτε ότι περιλαμβάνονται, πέρα από το κυρίως δίκτυο του Ethereum, που είναι προεπιλεγμένο, και τα δοκιμαστικά.



Εικόνα Π.8 *Επιλογές δικτύων στις ρυθμίσεις του Metamask.*

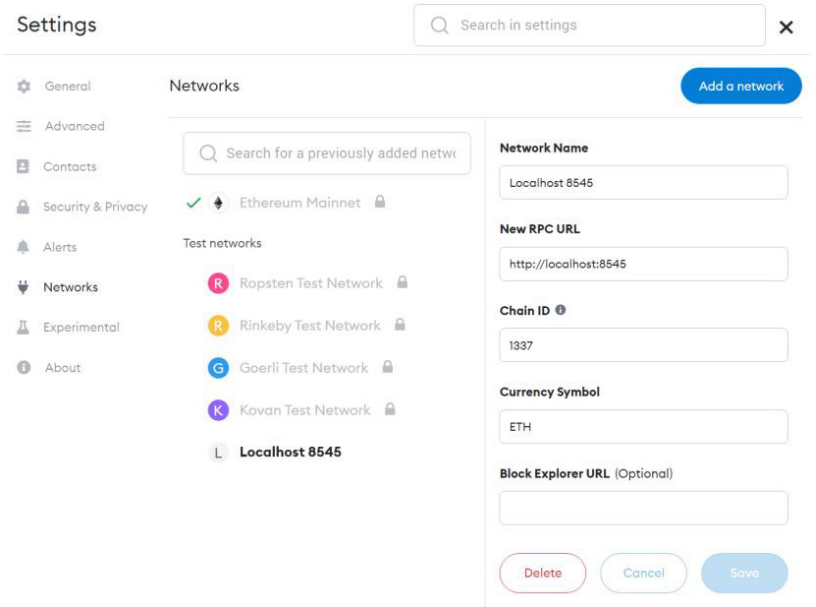
Δύο επιλογές στην Εικόνα Π.8 που είναι σημαντικές είναι: η δυνατότητα προσθήκης νέου δικτύου (*Add a network*) καθώς και η επιλογή *Localhost 8545*. Αν χρειάζεστε να συνδεθείτε σε ένα δίκτυο που έχει τα δικά του χαρακτηριστικά, μπορείτε να επιλέξετε το *Add a network*. Ενώ, αν χρειάζεστε να συνδεθείτε με κάποιο τοπικό δίκτυο blockchain, μπορείτε να χρησιμοποιήσετε την επιλογή Localhost.

Στη συνέχεια θα παρουσιαστεί πώς μπορεί να γίνει αυτή η σύνδεση με τοπικό δίκτυο.

Π.1.1 Συνδέοντας το Metamask με το Ganache

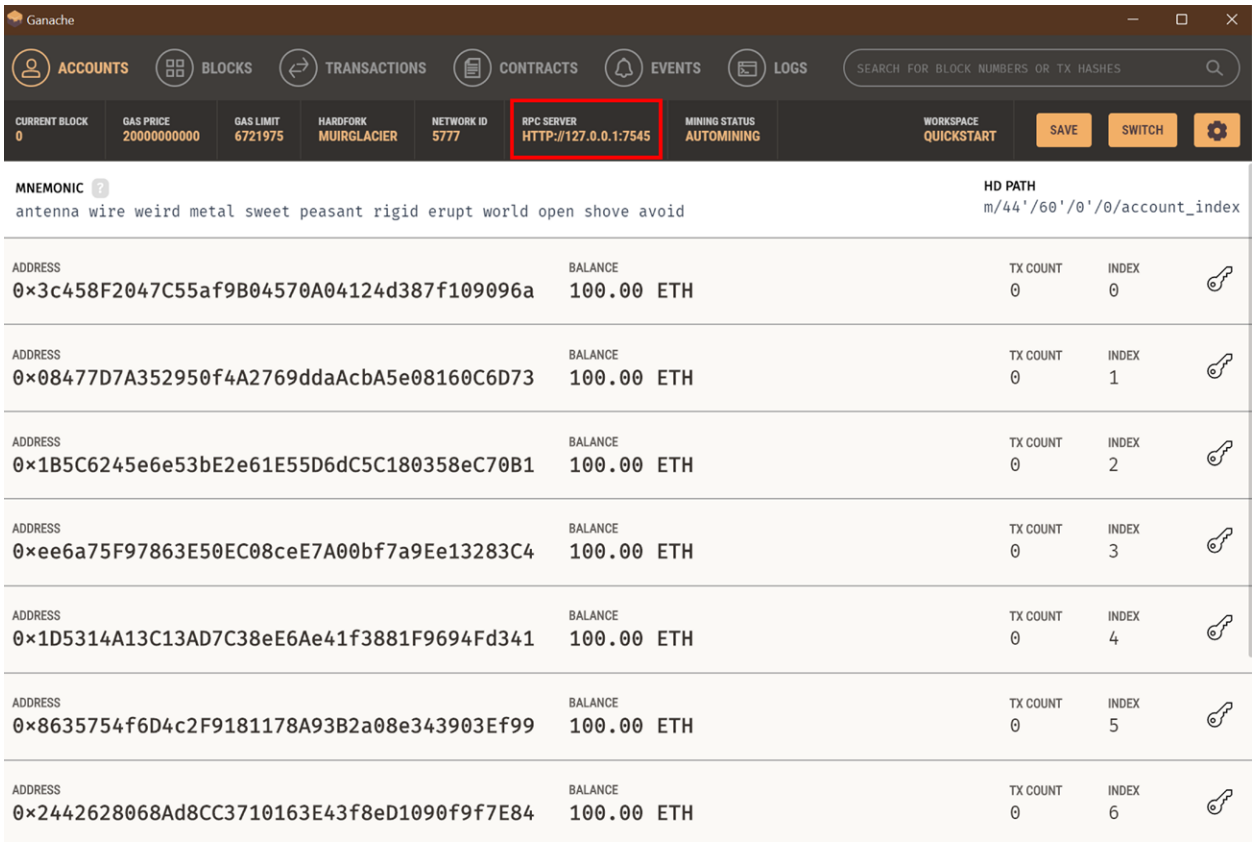
Θα παρουσιαστούν τα βήματα για τη σύνδεση του Metamask με ένα τοπικό δίκτυο που έχει δημιουργηθεί με τη βοήθεια του Ganache. Λεπτομέρειες για το Ganache δείτε στο Κεφάλαιο 11.

Στην **Εικόνα Π.9** βλέπετε στο Metamask τις επιλογές που υπάρχουν έτοιμες για τη σύνδεση με ένα τοπικό δίκτυο. Σημαντικό στην περίπτωση του Ganache είναι η πόρτα στην τοπική διεύθυνση για το RPC, που φαίνεται ότι είναι η http://localhost:8545.



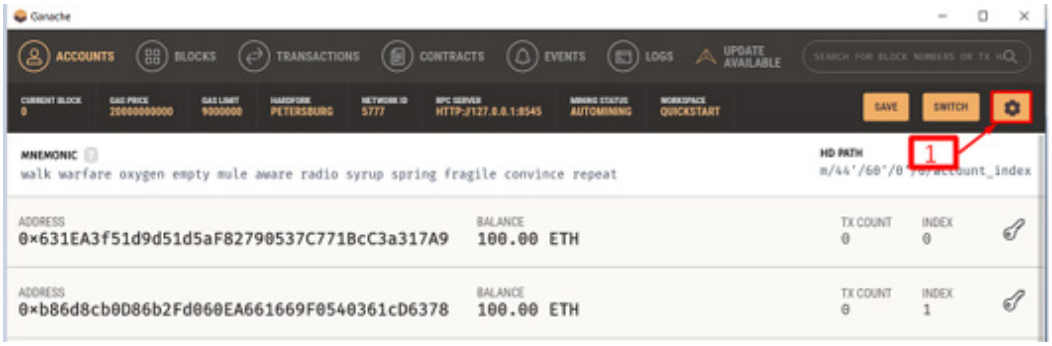
Εικόνα Π.9 *Επιλογές στις ρυθμίσεις του Metamask για τη σύνδεση με τοπικό δίκτυο.*

Αυτό είναι σημαντικό γιατί το Ganache αρχικά συνδέεται με την πόρτα 7545 (**Εικόνα Π.10**). Οπότε πρέπει να γίνει αλλαγή στην πόρτα στο Ganache.

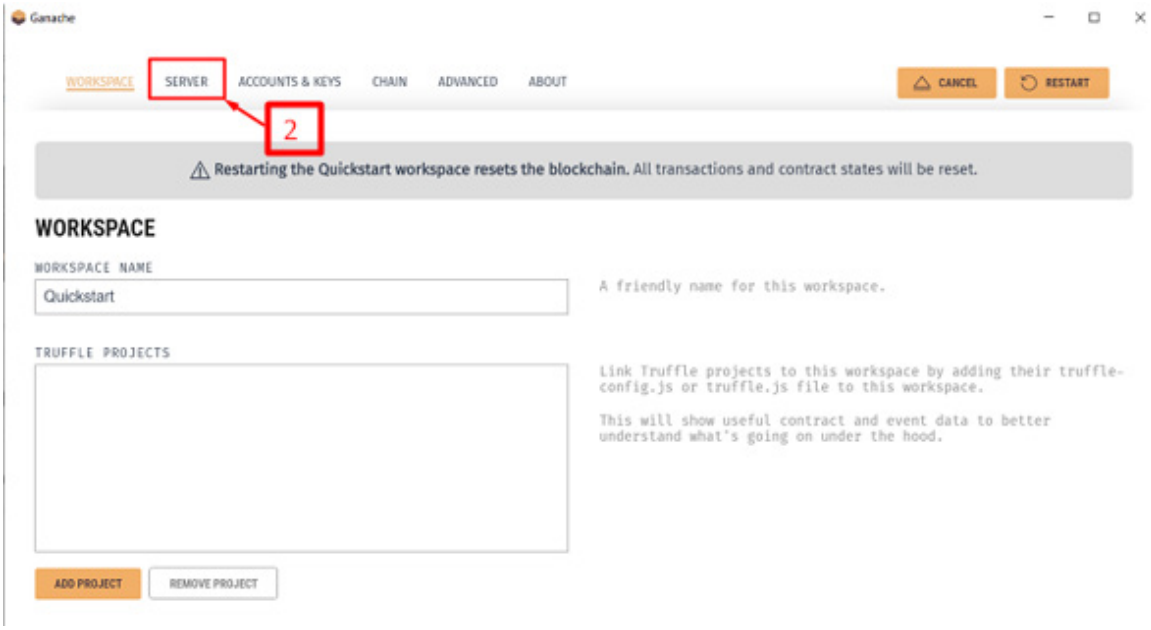


Εικόνα Π.10 *Η αρχική εικόνα στο Ganache.*

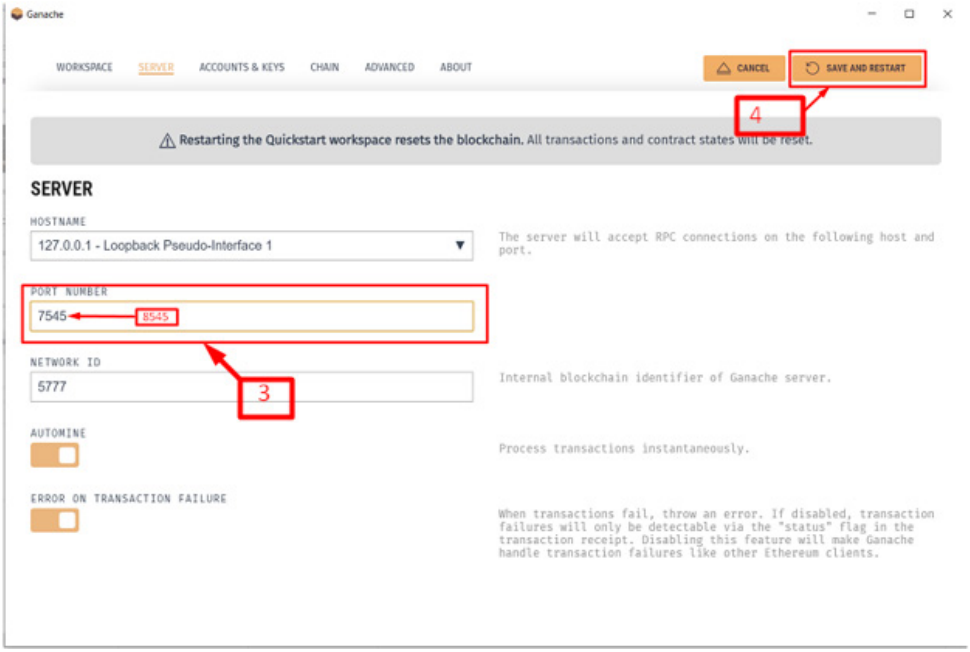
Για να γίνει η αλλαγή στο Ganache, ακολουθείτε τα βήματα που φαίνονται στις **Εικόνες Π.11-13**.



Εικόνα Π.11 *Αλλαγή πόρτας του RPC Server στο Ganache (Βήμα 1).*

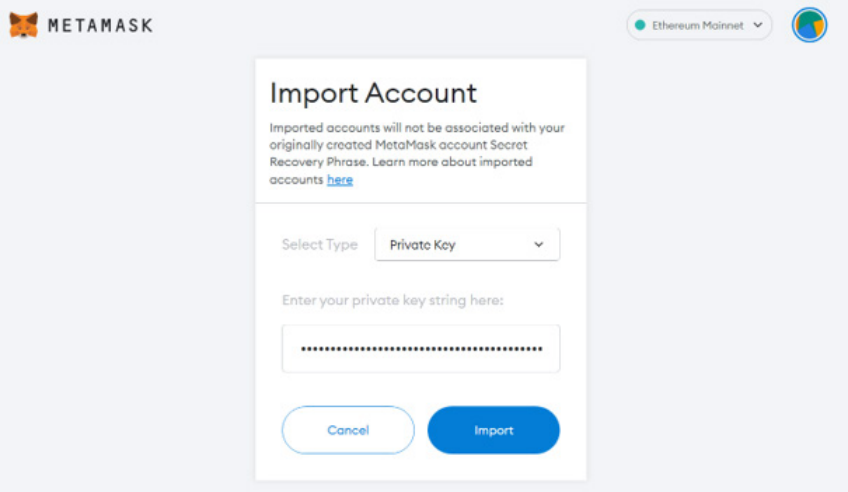


Εικόνα Π.12 *Αλλαγή πόρτας του RPC Server στο Ganache (Βήμα 2).*



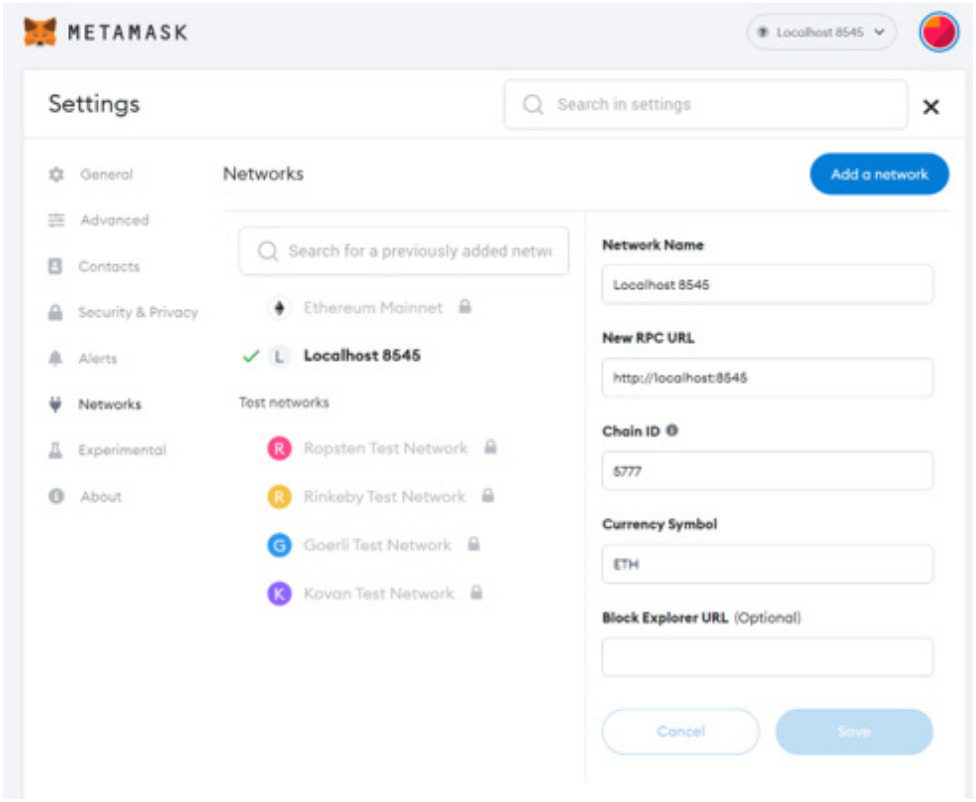
Εικόνα Π.13 *Αλλαγή πόρτας του RPC Server στο Ganache (Βήμα 3).*

Κατόπιν θα πρέπει να επιστρέψετε στο Metamask και να δημιουργήσετε έναν λογαριασμό χρησιμοποιώντας το ιδιωτικό κλειδί ενός (από τους 10) λογαριασμούς που έχουν δημιουργηθεί στο Ganache. Έστω ότι θα χρησιμοποιηθεί το κλειδί του λογαριασμού με δείκτη 1. Πατώντας στο κλειδί στο Ganache δίπλα από το Index 1, αντιγράφετε το κλειδί. Στο Metamask, ξεκινώντας από τις ρυθμίσεις (**Εικόνα Π.7**), επιλέγετε *Import Account* και συμπληρώνετε το κλειδί που αντιγράψατε προηγουμένως. Η **Εικόνα Π.14** δείχνει την επιλογή για φόρτωση του ιδιωτικού κλειδιού στο Metamask.



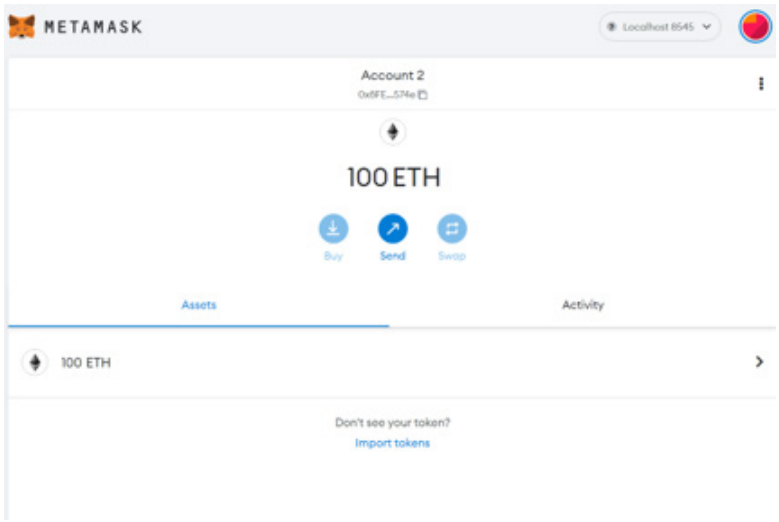
Εικόνα Π.14 *Προσθήκη του ιδιωτικού κλειδιού από το Ganache στο Metamask.*

Το επόμενο βήμα αφορά τη σύνδεση του Metamask με το τοπικό δίκτυο του Ganache. Αυτό το πετυχαίνετε από τις ρυθμίσεις (**Εικόνα Π.9**), επιλέγοντας το τοπικό δίκτυο. Προσοχή το Chain ID να είναι το ίδιο με το Network ID στο Ganache (**Εικόνα Π.13**). Η **Εικόνα Π.15** δείχνει τις ρυθμίσεις που πρέπει να ισχύουν στο Metamask για τη σύνδεση με το Ganache.



Εικόνα Π.15 *Οι ρυθμίσεις του τοπικού δικτύου στο Metamask.*

Επιστρέφοντας στην αρχική οθόνη του Metamask, η επιβεβαίωση για τη σύνδεση με το Ganache γίνεται κοιτώντας το υπόλοιπο του λογαριασμού. Τώρα θα πρέπει να δείτε ότι το υπόλοιπο έχει αλλάξει σε 100 ETH, όσο είναι και το ποσό στο Ganache. Αυτό φαίνεται στην **Εικόνα Π.16**.

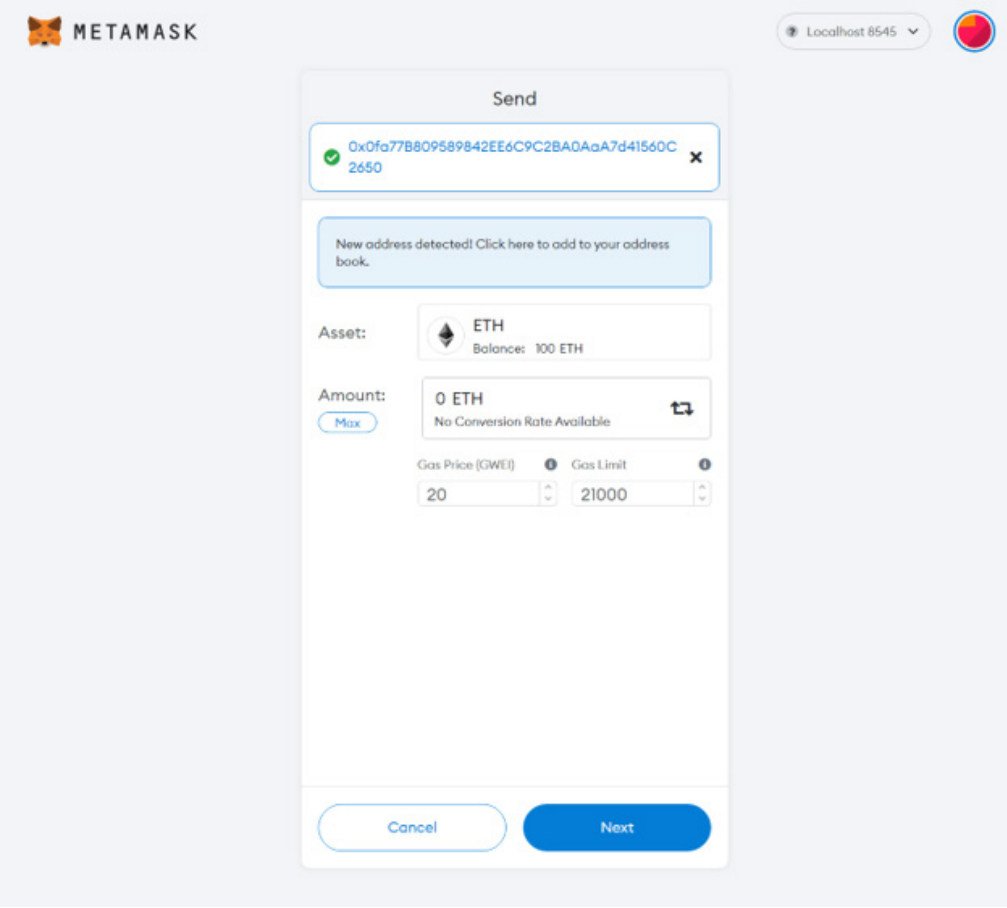


Εικόνα Π.16 *Η αρχική σελίδα στο Metamask που επιβεβαιώνει τη σύνδεση στο τοπικό δίκτυο.*

Π.1.2 Κάνοντας μια συναλλαγή στο Metamask

Κλείνοντας το Παράρτημα γνωριμίας με το Metamask, θα πραγματοποιηθεί και μια συναλλαγή σε αυτό.

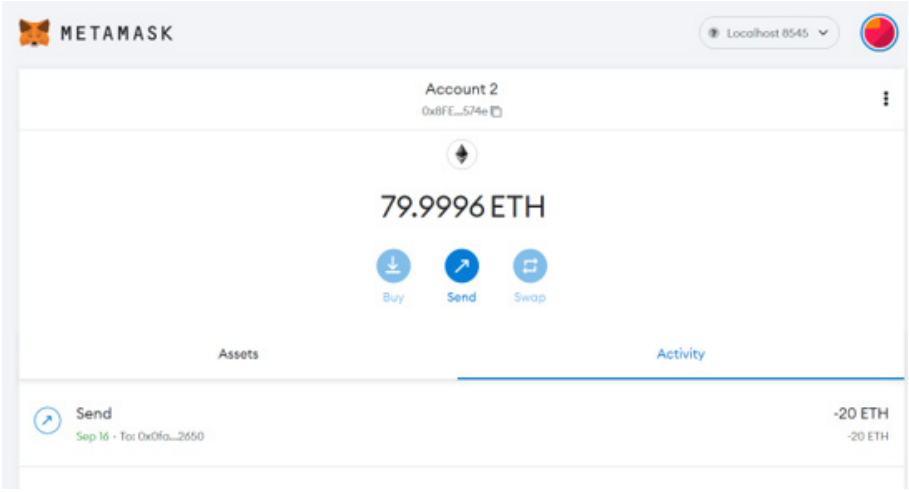
Στην αρχική σελίδα του πορτοφολιού θα πατήσετε την επιλογή *Send*. Στη νέα σελίδα που δημιουργείται αναμένεται από εσάς να προσθέσετε τη διεύθυνση του αποδέκτη των ETH που θα μεταφέρετε. Πηγαίνετε πίσω στο Ganache και επιλέγετε τη διεύθυνση που αντιστοιχεί στον δείκτη 2. Την αντιγράφετε και την περνάτε στο Metamask, όπως φαίνεται στην **Εικόνα Π.17**.



Εικόνα Π.17 *Η εισαγωγή της διεύθυνσης στο Metamask και η προετοιμασία του επόμενου βήματος για τη συναλλαγή.*

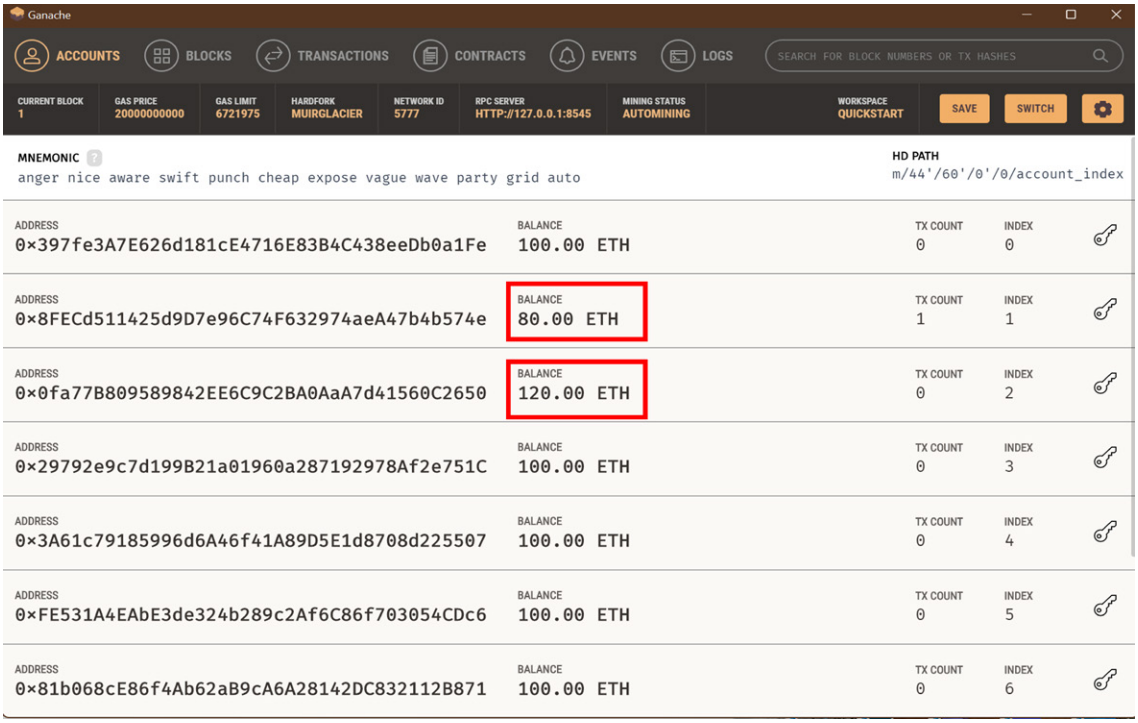
Αν η διεύθυνση είναι έγκυρη, τότε μας επιτρέπεται να συμπληρώσουμε το ποσό στη συναλλαγή (**Εικόνα Π.17** πεδίο *Amount*). Συμπληρώνετε ένα ποσό μικρότερο του 100 και πατάτε *Next*. Για δοκιμή μπορείτε να πειραματιστείτε και με ποσό μεγαλύτερο για να δείτε το μήνυμα που θα σας βγάλει το Metamask. Επίσης, μπορείτε να επιλέξετε και το κουμπί *Max*, για να συμπληρωθεί το μέγιστο επιτρεπόμενο ποσό. Αυτό, λόγω των gas fees, είναι μικρότερο από 100.

Πατώντας 20 στο ποσό και το κουμπί *Next*, ζητάτε να επιβεβαιωθεί η συναλλαγή. Πατώντας *Confirm*, η συναλλαγή στέλνεται στο δίκτυο. Με την επιστροφή στην κύρια σελίδα του πορτοφολιού, βλέπετε αρχικά τη συναλλαγή ως εκκρεμή μέχρι να επιβεβαιωθεί από το δίκτυο. Τότε, θα δείτε το υπόλοιπο του λογαριασμού σας να ανανεώνεται (**Εικόνα Π.18**).



Εικόνα Π.18 *Η εικόνα στο Metamask μετά την επεξεργασία της συναλλαγής στην οποία αποστέλλονται 20 ETH.*

Για να επιβεβαιώσετε την επεξεργασία της συναλλαγής, μπορείτε να επιστρέψετε και στο Ganache, όπου θα δείτε τα υπόλοιπα στις δύο εμπλεκόμενες σε αυτή διευθύνσεις (**Εικόνα Π.19**).



Εικόνα Π.19 *Η εικόνα στο Ganache μετά την επεξεργασία της συναλλαγής.*

Το σύγγραμμα εισάγει τον αναγνώστη στον κόσμο της τεχνολογίας του blockchain, χωρίς να χρειάζονται προαπαιτούμενες γνώσεις. Παρουσιάζονται τα βασικά χαρακτηριστικά της τεχνολογίας και αναλύονται οι πιο γνωστές εφαρμογές της (Bitcoin, Ethereum). Κατόπιν, αναλύεται η χρήση της ασύμμετρης κρυπτογραφίας στην τεχνολογία του blockchain, καθώς και ο τρόπος που εκτελούνται οι συναλλαγές σε ένα τέτοιο δίκτυο. Ιδιαίτερη έμφαση δίνεται στη χρήση ψηφιακών υπογραφών στις συναλλαγές. Στη συνέχεια, εξηγούνται ο ρόλος της συναίνεσης σε ένα κατανεμημένο δίκτυο και πώς επιτυγχάνεται αυτή στα δύο δίκτυα αναφοράς που αναφέρθηκαν προηγουμένως. Ακόμα, παρουσιάζεται ο ρόλος των έξυπνων συμβάσεων και δίνονται παραδείγματα συγγραφής και ανάπτυξής τους σε ένα δίκτυο blockchain, κατάλληλο για δοκιμές. Ο ρόλος και η χρήση των tokens σε ένα δίκτυο blockchain αναλύονται στη συνέχεια. Έμφαση δίνεται στην κατανόηση των διαφορετικών ειδών από tokens που μπορούν να δημιουργηθούν, εστιάζοντας στις διαφορές και στην περιγραφή των περιπτώσεων χρήσης που ταιριάζουν καλύτερα στο κάθε είδος. Το δίκτυο αναφοράς για την παρουσίαση των tokens είναι το Ethereum και γίνεται αναλυτική αναφορά στα πρότυπα που έχουν δημιουργηθεί σε αυτό για την υλοποίησή τους.

Το επόμενο βήμα εστιάζει στην παρουσίαση του τρόπου λειτουργίας και εφαρμογής των αποκεντρωμένων εφαρμογών (DApps), και ιδιαίτερα στις διαφορές του Web 2.0 με το Web 3.0. Κατόπιν, γίνεται μια παρουσίαση δημοφιλών περιπτώσεων χρήσης της τεχνολογίας του blockchain, συνοδευόμενη με την απαραίτητη επιχειρηματολογία που αναλύει τα κέρδη που σημειώνονται από την εφαρμογή μιας λύσης που βασίζεται στην τεχνολογία αυτή.

Για πληρέστερη παρουσίαση του χώρου, προς το τέλος του συγγράμματος ακολουθεί η παρουσίαση των Τεχνολογιών Κατανεμημένου Καθολικού, των οποίων το blockchain αποτελεί υποσύνολο. Τέλος, υπάρχει ένα κεφάλαιο πρακτικής εξάσκησης με την παρουσίαση ενός online ανοικτού εργαλείου, που επιτρέπει στον αναγνώστη να εξασκηθεί σε μεγάλο μέρος της θεματολογίας που αναπτύχθηκε στα προηγούμενα κεφάλαια.

Το παρόν σύγγραμμα δημιουργήθηκε στο πλαίσιο του Έργου ΚΑΛΛΙΠΟΣ+	
Χρηματοδότης	Υπουργείο Παιδείας και Θρησκευμάτων, Προγράμματα ΠΔΕ, ΕΠΑ 2020-2025
Φορέας υλοποίησης	ΕΛΚΕ ΕΜΠ
Φορέας λειτουργίας	ΣΕΑΒ/Παράρτημα ΕΜΠ/Μονάδα Εκδόσεων
Διάρκεια 2ης Φάσης	2020-2023
Σκοπός	Η δημιουργία ακαδημαϊκών ψηφιακών συγγραμμάτων ανοικτής πρόσβασης (περισσότερων από 700) • Προπτυχιακών και μεταπτυχιακών εγχειριδίων • Μονογραφιών • Μεταφράσεων ανοικτών textbooks • Βιβλιογραφικών Οδηγών
Επιστημονικά Υπεύθυνος	Νικόλαος Μήτρου, Καθηγητής ΣΗΜΜΥ ΕΜΠ

ISBN: 978-618-5726-49-2 **DOI:** http://dx.doi.org/10.57713/kallipos-171

Το παρόν σύγγραμμα χρηματοδοτήθηκε από το Πρόγραμμα Δημοσίων Επενδύσεων του Υπουργείου Παιδείας