

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

**Εφαρμογή Blockchain με Εφαρμογή σε Ιατρικά
Δεδομένα**

Εμμανουήλ Ανεζίνης Π2015097
Επιβλέπουσα: Άννα Σωτηροπούλου

30 Σεπτεμβρίου 2022

Επιβλέπουσα

Άννα Σωτηροπούλου. ΕΔΙΠ Τμήματος Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Άννα Σωτηροπούλου. ΕΔΙΠ Τμήματος Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Θεμιστοκλής Έξαρχος. Επίκουρος Καθηγητής, Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Εμμανουήλ Μάγκος. Καθηγητής, Τμήμα Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Αυτή η προσπάθεια αφιερώνεται στον παππού μου,
χωρίς τον οποίο δεν θα έφτανα ποτέ εδώ.

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι να προτείνει ένα σύστημα ανταλλαγής εικόνων με χρήση τεχνολογίας Blockchain, όπου η ακεραιότητα των εικόνων μπορεί να ελεγχθεί χωρίς περιθώρια λαθών και αμφιβολιών.

Στην αρχή υπάρχει επεξήγηση βασικών όρων και αρχών των Blockchain, ενώ στη συνέχεια παρουσιάζεται το πρόβλημα το οποίο αποτέλεσε έναυσμα για την δημιουργία της εφαρμογής. Προχωρώντας, λαμβάνει χώρα αναλυτική επεξήγηση των διακριτών λειτουργιών που προσφέρει η εφαρμογή, των ρόλων των χρηστών της, ενώ στη συνέχεια γίνεται παρουσίαση των κρυπτογραφικών τεχνικών που αποτελούν βάσεις για την ασφαλή λειτουργία της εφαρμογής.

Ακολουθεί η ποιοτική ανάλυση της εφαρμογής με γνωστές δημόσιες υλοποιήσεις Blockchain, με σκοπό την εξαγωγή γόνιμων συμπερασμάτων. Εν κατακλείδι, γίνεται περιγραφή της λειτουργικότητας της εφαρμογής μέσω εικόνων της Επιφάνειας Διεπαφής Χρήστη, ταυτόχρονα με την περιγραφή των σημαντικότερων τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση.

Πρόλογος και Ευχαριστίες

Η παρούσα πτυχιακή εργασία με τίτλο “Εφαρμογή Blockchain με Εφαρμογή σε Ιατρικά Δεδομένα”, αποτελεί αντανάκλαση των γνώσεων που έλαβα κατά τη διάρκεια των σπουδών μου στο Ιόνιο Πανεπιστήμιο, και της ικανότητας εφαρμογής αυτών σε πρακτικό επίπεδο. Θα ήθελα να ευχαριστήσω ιδιαίτερα την επιβλέπουσα καθηγήτρια αυτής της προσπάθειας, κυρία Σωτηροπούλου, για την πολύτιμη υποστήριξη και καθοδήγηση που μου παρείχε κατά τη διάρκεια της φοίτησής μου. Ταυτόχρονα, θα ήθελα να ευχαριστήσω το σύνολο του διδακτικού προσωπικού του Ιονίου Πανεπιστημίου για όλα τα ακαδημαϊκά εφόδια που μου προσέφερε. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, για όλη την υποστήριξη που μου παρείχε καθ’ όλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Περίληψη	4
Πρόλογος και Ευχαριστίες	5
Περιεχόμενα	6
Α' Εισαγωγή	9
Α'.1 Έννοια του Blockchain	9
Α'.1.1 Public Blockchains	10
Α'.1.2 Private Blockchains	10
Α'.2 Ιστορική αναδρομή	10
Α'.3 Block	11
Α'.4 Δοσοληψίες	11
Α'.5 Αλγόριθμοι ομοφωνίας	12
Α'.5.1 Αρχή Απόδειξης Μόχθου (Proof of Work)	13
Α'.5.2 Επίθεση Πλειοψηφίας 51%	14
Α'.6 Κρυπτονομίσματα	14
Β' Προτεινόμενο Σύστημα	14
Β'.1 Υπάρχον πρόβλημα	14
Β'.2 Σκοπός του συστήματος	15
Β'.3 Health-Coin	15
Β'.4 Σενάρια Χρήσης και τύποι χρηστών	16
Β'.4.1 Χρήστες	16
Β'.4.2 Αποστολή εικόνων	17
Β'.5 Απαιτήσεις Συστήματος	17
Β'.6 Ιδιαιτερότητες του προτεινόμενου Blockchain	17
Β'.6.1 Τρόπος Αποθήκευσης	18
Β'.6.2 Client Side: PouchDB	18
Β'.6.3 Peer-2-Peer Networking	19
Β'.6.4 Timestamp server	19
Β'.6.5 Δέντρα Merkle στα Block	19
Β'.6.6 In-browser Mining	20
Β'.7 Απόδοση κλειδιών	20
Β'.8 Ταυτότητα και προστασία των χρηστών	21
Β'.9 Απευθυνόμενο κοινό	21
Β'.10 Ποιοτική και Ποσοτική σύγκριση συστήματος	22
Β'.10.1 Αποθήκευση	22
Β'.10.2 Ασφάλεια	22
Β'.10.3 Διανομή και χρήσεις	23
Β'.10.4 Κόστος	23
Β'.11 Περιορισμοί του συστήματος	23
Γ' Οθόνες εφαρμογής	24
Δ' Τεχνολογίες Υλοποίησης	27
Δ'.1 Javascript	27

Δ'.2 Typescript	27
Δ'.3 Angular	27
Δ'.4 Bootstrap	28
Δ'.5 Elliptic Curve Cryptography	28
Δ'.6 Secure Hash Algorithm 256	30
Ε' Σημαντικές βιβλιοθήκες και εργαλεία	31
Ε'.1 Node Package Manager	31
Ε'.2 Crypto-js	31
Ε'.3 Elliptic	31
ΣΤ' Μελλοντικές προεκτάσεις και συμπεράσματα	32
ΣΤ'.1.1 Ασφαλή Πρωτόκολλα επικοινωνίας	32
ΣΤ'.1.2 Docker	32
ΣΤ'.1.3 Docker Compose	32
ΣΤ'.1.4 Server Side: CouchDB	33
ΣΤ'.1.5 Portainer	33
ΣΤ'.2 Health-coin και συνήθειες πρακτικές	34
ΣΤ'.3 Έξυπνα Συμβόλαια	35
ΣΤ'.4 Miners	36
ΣΤ'.5 Υποστηριζόμενοι τύποι αρχείων	36
ΣΤ'.6 Συμπεράσματα	36
Βιβλιογραφικές Αναφορές	37
Appendix	39

Α' Εισαγωγή

Ο παράγοντας της πληροφορίας, από τις απαρχές του πολιτισμού, καθορίζει και κατευθύνει σε μεγάλο βαθμό την πορεία της ανθρωπότητας. Αυτό συμβαίνει καθώς ο άνθρωπος, ως σκεπτόμενο ον, βασίζει τις αποφάσεις του πάνω στο σύνολο των πληροφοριών που έχει στη διάθεση του, όπου η πληροφορία αποτελεί προϊόν επεξεργασίας ακατέργαστων δεδομένων, τα οποία ο άνθρωπος λαμβάνει από το περιβάλλον του. Η εξάπλωση του Παγκόσμιου Ιστού επέφερε αδιανόητη αύξηση στον όγκο των πληροφοριών που 'δημιουργούνται' και διακινούνται, γεγονός που αναδεικνύει περαιτέρω την ανάγκη για δυνατότητα ελέγχου της εγκυρότητάς τους. Η ανάγκη αυτή είναι παρούσα σε πληθώρα τομέων της κοινωνίας (πχ. χρηματοπιστωτικές συναλλαγές, έλεγχος ιατρικών δεδομένων), όπου για κάθε περίπτωση απαιτείται ξεχωριστή προσέγγιση, ανάλογα με τον σκοπό προς επίτευξη, ενώ ταυτόχρονα οφείλουμε να σεβόμαστε τα δικαιώματα των χρηστών στα δεδομένα τους, καθώς παρατηρούνται περιπτώσεις όπου δεν τους αποδίδονται. Θα επικεντρωθούμε λοιπόν στις περιπτώσεις αποθήκευσης ιατρικών δεδομένων από ιατρικούς οργανισμούς, προτείνοντας εφαρμογή blockchain που καθιστά τους χρήστες μοναδικούς ιδιοκτήτες των δεδομένων τους, με πλήρη διαφάνεια στην ροή της πληροφορίας χωρίς να εμποδίζονται δυνατότητες διαμοιρασμού και εκχώρησης δικαιωμάτων. Ταυτόχρονα οι χρήστες καθίστανται υπεύθυνοι για τη διατήρηση της ασφάλειας των δεδομένων τους, μέσω της διαφύλαξης των ιδιωτικών κλειδών που τους παρέχονται.

Επιλέξαμε την υλοποίηση του συγκεκριμένου συστήματος με τεχνολογία blockchain, καθώς τα χαρακτηριστικά που τη διέπουν ουσιαστικά αποτελούν και τα κύρια χαρακτηριστικά που επιθυμούμε για το προτεινόμενο σύστημα. Πιο συγκεκριμένα αναφερόμαστε στα χαρακτηριστικά της διαφάνειας των δοσοληψιών που αποτελούνται από μεταφορτώσεις ή μετακινήσεις αρχείων, της αποκλειστικής πρόσβασης στα δεδομένα που κρυπτογραφούνται από τον ιδιοκτήτη τους με χρήση ιδιωτικού κλειδιού, στην παροχή δικαιωμάτων διαμοιρασμού όπου το επιθυμεί ο χρήστης, με τα δεδομένα που μεταφορτώνονται να παραμένουν μόνιμα αποθηκευμένα και ασφαλή, χωρίς ύπαρξη πιθανότητας διαγραφής τους.

Α'.1 Έννοια του Blockchain

Η τεχνολογία αλυσίδας συστοιχιών (BlockChain-BC) ανήκει στην ευρύτερη οικογένεια των τεχνολογιών καταμεμημένου καθολικού (distributed ledger table) [1]. Παρουσιάζει πολλά όμοια χαρακτηριστικά με τη δομή αποθήκευσης δεδομένων συνδεδεμένης λίστας, όπου τα δεδομένα συνδέονται μέσω Κρυπτογραφικών τιμών, και πιο συγκεκριμένα, Τιμών Κατακερματισμού (hash values) [2]. Η δομή αυτή διέπεται από αυστηρούς κανόνες (consensus algorithms), τους οποίους οφείλουν να τηρούν όλοι οι συμμετέχοντες, έτσι ώστε να διασφαλίζεται η ομαλή λειτουργία του BC. Η διαχείριση των BC λαμβάνει χώρα συνήθως μέσω χρήσης του μοντέλου Δικτύων Ομότιμων Κόμβων (peer-to-peer network) [3], όπου όλοι οι συμμετέχοντες έχουν ίδια δικαιώματα, χωρίς να υπάρχει κεντρική εξουσία (central authority), ενώ ταυτόχρονα παρέχεται καταμεμημένη δικτύωση, παράγοντας που ενισχύει την ασφάλεια του συστήματος. Κάθε συσκευή που συμμετέχει στο δίκτυο

αποτελεί έναν κόμβο (node), ο οποίος συμμετέχει στην αποθήκευση των δεδομένων, τον διαμοιρασμό και την επικύρωση τους.

Κατά τη δημιουργία και παραμετροποίηση τους, ο δημιουργός προσθέτει ειδικά καθοριστικά χαρακτηριστικά, με αποτέλεσμα το BC μετά την παραμετροποίηση του να ακολουθεί, έστω άτυπα, κάποιο συγκεκριμένο Δόγμα (paradigm), ενώ παρατηρούνται μεγάλα επίπεδα διαφοροποίησης από κατηγορία σε κατηγορία. Τα παραπάνω μας οδηγούν στον γενικό διαχωρισμό των BC σε δύο κύριες κατηγορίες.

A'.1.1 Public Blockchains

Ένα δημόσιο (public) Blockchain [4] αποτελεί πλατφόρμα ανοιχτού λογισμικού (open-source) χωρίς Κεντρική εξουσία (decentralized), αλλά και χωρίς περιορισμούς ως προς το ποιος, ή τον αριθμό χρηστών, που μπορούν να συμμετέχουν στο δίκτυο. Οι χρήστες έχουν δικαίωμα ελεύθερης επικοινωνίας μεταξύ τους, ενώ τους παρέχονται δικαιώματα ανάγνωσης (read), εγγραφής (write) και ελέγχου (audit) του BC. Καθώς όλα τα μέλη είναι ομότιμα, οι αποφάσεις που αφορούν το BC λαμβάνονται με την υλοποίηση αλγορίθμων ομοφωνίας, ενώ περιλαμβάνεται και σύστημα επιβράβευσης, το οποίο προάγει την ειλικρίνεια μεταξύ των συμμετεχόντων και κατ' επέκταση την εύρυθμη λειτουργικότητα του BC. Βασικό χαρακτηριστικό αποτελεί το γεγονός ότι δεν απαιτείται από τους χρήστες ειδική άδεια για πραγματοποίηση δοσοληψιών ή ενεργειών (permissionless).

A'.1.2 Private Blockchains

Τα ιδιωτικά (Private) BC παρουσιάζουν σημαντικές διαφορές σε σχέση με τα δημόσια, καθώς δημιουργούνται έχοντας ως θεμέλιο λίθο την προσθήκη δυνατοτήτων διαχειριστή (administrative privileges) [4]. Συνήθως κατασκευάζονται από οργανισμούς οι οποίοι θέτουν τον εαυτό τους ως την Κεντρική Εξουσία (central authority) και συνεπώς η λήψη των αποφάσεων εξαρτάται αποκλειστικά από τον εκάστοτε οργανισμό. Σημαντική διαφορά αποτελεί επίσης το γεγονός ότι δεν υπάρχει ελευθερία συμμετοχής για όποιον το επιθυμεί, αντιθέτως ο οργανισμός καλείται να προεπιλέξει τους συμμετέχοντες, αποκλειστικά, και να τους αποδώσει δικαιώματα κατά τη δική του κρίση. Επίσης καθίσταται απαραίτητο, όταν κάποιος χρήστης επιθυμεί να πραγματοποιήσει μια δοσοληψία, ενδέχεται να αναγκαστεί να ζητήσει άδεια από την Κεντρική Εξουσία (permissioned). Η κεντροποίηση της εξουσίας (centralization) οδηγεί στο να είναι το BC πιο ευάλωτο, καθώς οποιαδήποτε διαφθορά της CA καταλογίζεται ως αστοχία ενός και μόνου σημείου (single point of failure).

A'.2 Ιστορική αναδρομή

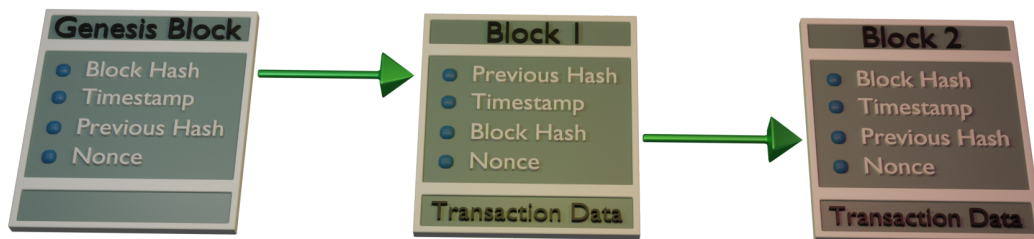
Η πρώτη εφαρμογή της τεχνολογίας BC, αποδίδεται στον Satoshi Nakamoto με τη δημιουργία του ψηφιακού νομίσματος Bitcoin (BTC) [5]. Ο Satoshi εφάρμοσε την υλοποίηση των αμετάβλητων χρονολογημένων πιστοποιήσεων εγγράφων με χρήση δέντρων Merkle [6], των Haber S - Stornetta WS [7], σε συνδυασμό με τον αλγόριθμο ομοφωνίας (consensus algorithm) που εφαρμόζει την Αρχή Απόδειξης Μόχθου (proof of work), η οποία είναι παρόμοια με την αντίστοιχη υλοποίηση του HashCash, με δημιουργό τον Adam Back [8].

A'.3 Block

Τα block αποτελούν το κύριο δομικό στοιχείο ενός BC, ενώ οι συνδέσεις των block μπορούν να χαρακτηριστούν ως μια αλυσίδα, γεγονός από το οποίο απορρέει και το όνομα Αλυσίδα Συστοιχιών. Τα εκάστοτε δεδομένα προς μεταφορά αποθηκεύονται εμφωλευμένα στα block. Τα block επίσης περιέχουν πληροφορίες όπως η χρονοσήμανση παραγωγής του block, η τιμή κατακερματισμού (hash) του και η τιμή του προηγούμενου. Οι συνδέσεις των τιμών κατακερματισμού μεταξύ των block προσδίδουν στο BC το γνώρισμα της αμεταβλητότητας (immutability), καθώς οποιαδήποτε μετατροπή στο hash ενός block επιφέρει αλυσιδωτές αλλαγές στα προηγούμενα, αλλά και τα επόμενα block που συνδέονται μαζί του, γεγονός το οποίο δρα ως αποτρεπτικός παράγοντας για όλες τις οντότητες που ενδέχεται να θέλουν να παραποιήσουν το καθολικό (ledger) χωρίς ύπαρξη ομοφωνίας.

A'.4 Δοσοληψίες

Οι δοσοληψίες (transactions) που δύναται να πραγματοποιηθούν μέσα στα πλαίσια ενός BC, ενδέχεται να αφορούν διαφορετικούς τύπους δεδομένων. Η πλειοψηφία των υλοποιήσεων BC αφορά τη διατήρηση καθολικού (ledger) για καταγραφή χρηματοπιστωτικών συναλλαγών, αλλά ταυτόχρονα λειτουργούν και παραλλαγές, όπου αποθηκεύουν τιμές κατακερματισμού, οι οποίες ενδέχεται να χρησιμοποιηθούν μετέπειτα για πληθώρα λόγων όπως η εξακρίβωση ακεραιότητας δεδομένων, η πρόσβαση σε αρχεία με χρήση της τιμής κατακερματισμού τους αλλά και άλλες εφαρμογές.



Εικόνα 1. Ενδεικτική δομή Blockchain

Α'.5 Αλγόριθμοι ομοφωνίας

Οι αλγόριθμοι ομοφωνίας (consensus algorithm) αξιοποιούνται από καταναεμημένα συστήματα, έτσι ώστε να επέλθει συμφωνία όλων των συμμετεχόντων ως προς οποιαδήποτε τιμή/παράμετρο (data value) του συστήματος. Στη περίπτωση του BC, οι αλγόριθμοι ομοφωνίας αξιοποιούνται έτσι ώστε η πλειοψηφία των χρηστών (nodes), να είναι σε θέση να συμφωνήσει για την κατάσταση του Καθολικού (ledger), που διανέμεται και αποθηκεύεται καταναεμημένα από όλους τους χρήστες. Οι αλγόριθμοι αυτοί σχεδιάστηκαν με γνώμονα την επίτευξη αξιοπιστίας (reliability) σε περιβάλλοντα όπου, ενδεχομένως, σημαντικό ποσοστό των χρηστών να θεωρείται αναξιόπιστο [9].

A'.5.1 Αρχή Απόδειξης Μόχθου (Proof of Work)

Ο αλγόριθμος ομοφωνίας με τίτλο Αρχή Απόδειξης Μόχθου υιοθετήθηκε για πρώτη φορά από τον Satoshi Nakamoto με την υλοποίηση του Bitcoin [5]. Κατά τη πραγματοποίηση δοσοληψιών στο BC, όλες οι εκκρεμείς δοσοληψίες αποθηκεύονται με σκοπό, μετά την πιστοποίηση και τον έλεγχο τους, τα δεδομένα τους να μετατραπούν σε δομή Block, και να προστεθούν στο BC. Για να συμβεί αυτό όμως, είναι απαραίτητος ο υπολογισμός της τιμής κατακερματισμού του υποψήφιου Block προς πρόσθεση. Ο αλγόριθμος κατακερματισμού που χρησιμοποιεί το Bitcoin είναι ο SHA-256 (Secure Hash Algorithm), ο οποίος παράγει τιμή κατακερματισμού μήκους 64 χαρακτήρων. Ο πρώτος χρήστης (node), που θα υπολογίσει την τιμή αυτή, έχει δικαίωμα να προτείνει το επόμενο block προς πρόσθεση στην αλυσίδα. Οι υπόλοιποι χρήστες μπορούν να ελέγξουν αν η τιμή είναι σωστή, καθώς το αποτέλεσμα των συναρτήσεων κατακερματισμού επαληθεύεται πολύ εύκολα, αλλά υπολογίζεται πολύ δυσκολότερα. Κατά τη διαδικασία υπολογισμού, ο αλγόριθμος ομοφωνίας επιβάλλει έναν πολύ σημαντικό κανόνα: Η τιμή κατακερματισμού οφείλει να ξεκινάει από συγκεκριμένο αριθμό προκαθορισμένων χαρακτήρων, και στη περίπτωση του bitcoin με 8 μηδενικά ψηφία στην αρχή. Η προσπάθεια προσθήκης των ψηφίων, ώστε να ταιριάζουν στις απαιτήσεις του αλγορίθμου απαιτεί πολλαπλές επαναλήψεις, γεγονός το οποίο καθιστά τη διαδικασία χρονοβόρα και απαιτητική. Ταυτόχρονα, με τις επαναλήψεις διατηρείται ο αριθμός NONCE (Number used Once), ο οποίος προστίθεται στις πληροφορίες οι οποίες είναι απαραίτητες για τον υπολογισμό του hash. Ο τυχαίος αριθμός που προστίθεται αυξάνει περαιτέρω τον αριθμό των επαναλήψεων, προσθέτοντας έναν ακόμα τυχαίο παράγοντα στον υπολογισμό της τιμής κατακερματισμού. Το μέγεθος των απαιτήσεων καθορίζεται από τη δυσκολία (difficulty) με την οποία έχει παραμετροποιηθεί ο δημιουργός το Blockchain. Η δυσκολία του Bitcoin έχει επιλεγεί ώστε να είναι δυνατό να παράγεται ένα block ανά δέκα λεπτά, με απαιτήσεις ενέργειας κατά τον υπολογισμό 700 κιλοβατώραν ανά ώρα (kWh).

Ο μεγάλος αριθμός δυσκολίας καθιστά πολύ δύσκολο τον επανυπολογισμό της αλυσίδας των Block, με τη δυσκολία να αυξάνεται ανάλογα με τον αριθμό των μπλοκ, προσθέτοντας επιπλέον χαρακτηριστικά ασφαλείας [5]. Το γεγονός ότι μια οντότητα/χρήστης επένδυσε μεγάλα ποσά ενέργειας για τον υπολογισμό της σωστής τιμής, μεταφράζεται ως ότι οι άλλοι χρήστες μπορούν να του αναθέσουν τη μόνιμη πρόσθεση του Block ως δημόσια πληροφορία στο BC. Σε περίπτωση κατάθεσης λανθασμένης τιμής, οι χρήστες έχουν το δικαίωμα απόρριψης της προσθήκης του Block, ενώ σε άλλες εφαρμογές BC έχουν ακόμα και τη δυνατότητα επιβολής προστίμου στην οντότητα του δήλωσε λανθασμένη τιμή (slashing). Ταυτόχρονα με τη μόνιμη προσθήκη του Block, αποδίδεται στην οντότητα, ως επιβράβευση για τον επιτυχημένο υπολογισμό, συγκεκριμένη ποσότητα κρυπτονομισμάτων (cryptocurrency). Το έπαθλο επαναπροσδιορίζεται ταυτόχρονα με τη δυσκολία, ώστε να διατηρείται ο χρόνος υπολογισμού ανάλογα με την πρόοδο της τεχνολογίας, αλλά και να προσαρμόζεται ο ρυθμός απελευθέρωσης των νομισμάτων με σκοπό τον χαρακτηρισμό του νομίσματος ως αποπληθωριστικό, με δυνατότητα διατήρησης και περαιτέρω αύξησης της αξίας του. Η διαδικασία υπολογισμού στον συγκεκριμένο αλγόριθμο αναφέρεται ως Εξόρυξη (Mining). Ο αλγόριθμος ομοφωνίας Proof of Work, υπό τις κατάλληλες συνθήκες, αποτελεί ένα πολύ ασφαλές

μέσο ομοφωνίας, όμως υπό άλλες συγκεκριμένες συνθήκες ενδέχεται να υπάρξει μια κακόβουλη οντότητα, η οποία θα έχει τη δυνατότητα να υφαρπάξει τον έλεγχο του BC [5].

A'.5.2 Επίθεση Πλειοψηφίας 51%

Ο αλγόριθμος ομοφωνίας μπορεί να παραβιαστεί στην περίπτωση που κάποια οντότητα κατέχει πάνω από τη μισή συνολική υπολογιστική δύναμη, την οποία κατέχει το σύνολο των συμμετεχόντων του δικτύου. Σε αυτή τη περίπτωση η οντότητα αυτή μπορεί να εγκρίνει δοσοληψίες χωρίς τη σύμφωνη γνώμη των υπόλοιπων συμμετεχόντων, καθώς μπορεί να υπολογίσει τις τιμές κατακερματισμού μελλοντικών block αυτόνομα, αλλά και να επανακατασκευάσει τις τιμές προηγούμενων [10]. Στη περίπτωση του Bitcoin, η διασπορά των υπολογιστικών μέσων, δηλαδή των εργαλείων υπολογισμού (κάρτες γραφικών) και ο τεράστιος όγκος απαιτούμενης υπολογιστικής ισχύς δρουν ως κατασταλτικοί παράγοντες για αυτή την ευπάθεια.

A'.6 Κρυπτονομίσματα

Ο συγκεκριμένος παράγοντας αποτελεί αφενός μέσο πρόσδοσης χρηματικής αξίας στο BC, και αφετέρου αποτελεί ένα από τα βασικά διαθέσιμα κίνητρα ενασχόλησης, καθώς ενθαρρύνεται η έννομη και ομαλή συμμετοχή στον αλγόριθμο ομοφωνίας με σκοπό το κέρδος. Μέσα από αυτή τη διαδικασία οι χρήστες προτιμούν να διατηρήσουν τις ενέργειες τους σε επιτρεπτά πλαίσια, με αποτέλεσμα την προαγωγή εμπιστοσύνης και καλής θελήσεως για συνολική εργασία με κύριο γνώμονα χρηματική αποζημίωση [11].

B' Προτεινόμενο Σύστημα

B'.1 Υπάρχον πρόβλημα

Σύμφωνα με παρατηρήσεις της διεθνούς επιστημονικής κοινότητας, η καταγραφή και αποθήκευση ιατρικών δεδομένων επιδεικνύει σοβαρές παθογένειες. Σε πληθώρα περιπτώσεων έχει παρατηρηθεί καταγραφή ανακριβών στοιχείων, τα οποία σχετίζονται με το ιατρικό ιστορικό των ασθενών [12]. Παράλληλα δεν αποκλείονται κακόβουλες ενέργειες οι οποίες σχετίζονται με το ανθρώπινο δυναμικό που είναι υπεύθυνο για την αποθήκευση των δεδομένων. Το γεγονός αυτό ενδέχεται να οδηγήσει σε λανθασμένες διαγνώσεις, καθώς το ιατρικό προσωπικό λαμβάνει το ιστορικό ως μια σημαντική παράμετρο στη διαδικασία διάγνωσης. Η έλλειψη αξιόπιστων, και ασφαλών συστημάτων διαμοιρασμού εικόνων οδηγεί πολλές φορές τους ασθενείς να αποθηκεύουν οι ίδιοι τις εξετάσεις τους και να προβαίνουν σε πολλαπλές μεταφορές των εικόνων, παράγοντας ο

ο οποίος δρά κατασταλτικά ως προς τη διατήρηση της ακεραιότητας των δεδομένων. Σκεπτόμενοι επίσης το ιδεώδες της Εξατομικευμένης Ιατρικής, ιδεώδες του οποίου η λειτουργία βασίζεται και προϋποθέτει ανάλυση και ύπαρξη ακριβών (accurate) δεδομένων, θεωρείται κρίσιμη η ύπαρξη συστήματος διαμοιρασμού και αποθήκευσης εικόνων, το οποίο εξασφαλίζει τη διατήρηση της ακεραιότητας των πληροφοριών που διακινούνται στο σύστημα, παρέχοντας αξιόπιστες μεθόδους επαλήθευσης, χωρίς περιθώρια αποτυχίας.

B'.2 Σκοπός του συστήματος

Το παρακάτω προτεινόμενο σύστημα έχει ως απώτερο σκοπό την παροχή δυνατότητας Ανταλλαγής εικόνων, με κύρια λειτουργικότητα τον αδιαμφισβήτητο έλεγχο της ακεραιότητας (integrity) των εικόνων μετά τη μεταφορά και παραλαβή τους. Το σύστημα αποτελείται από δύο κύρια μέρη: Τη διαδικτυακή εφαρμογή (web application) που δρα ως διεπαφή χρήστη (user interface) σε συνδυασμό με δημόσιο BC, χωρίς προαπαιτούμενες ή ειδικές άδειες (public permissionless Blockchain)δρα ως εργαλείο εξακρίβωσης της ακεραιότητας των εικόνων μετά τη μεταφορά και παραλαβή τους.

B'.3 Health-Coin

Η εισαγωγή λειτουργικότητας Κρυπτονομίσματος αποτέλεσε απόφαση η οποία προέκυψε μετά από εκτεταμένη βολιδοσκόπηση και αποκρυστάλλωση δεδομένων που αφορούν συμπεριφορές ανθρώπων, όσον αφορά δεδομένα ευαίσθητης φύσεως. Σε ένα ιδανικό περιβάλλον, όπου η ανθρώπινη ζωή θεωρείται συνολικά το σημαντικότερο αγαθό, θα μπορούσε να διατίθεται υπολογιστική δύναμη για έλεγχο και εξακρίβωση ευαίσθητων ιατρικών δεδομένων, χωρίς την ύπαρξη χρηματικής αποζημίωσης. Τα δεδομένα που προδιαγράφεται να μετακινούνται ενδέχεται να αποτελούν βιοδείκτες ή εξετάσεις ασθενών, πάνω στους οποίους λαμβάνονται αποφάσεις που αφορούν τη ζωή, αυτή καθ' αυτή, των ασθενών. Η συμμετοχή στη διασφάλιση της ποιότητας τέτοιων κρίσιμων δεδομένων, θα 'έπρεπε' να είναι αρκετή ως ηθική ικανοποίηση, έτσι ώστε ο οποιοσδήποτε συνεισφέρει, να βελτιώνει την ιατρική περίθαλψη όχι μόνο εκείνου, αλλά και των συνανθρώπων του.

Δυστυχώς όμως, παρατηρούμε συχνά φαινόμενα τα οποία βάζουν σε κίνδυνο ανθρώπινες ζωές, στο βωμό του χρήματος. Φυσικά αναφερόμαστε σε γεγονότα, όπως η απόπειρα ηλεκτρονικής επίθεσης σε νοσοκομείο Παίδων στην Βοστώνη της Αμερικής [12], ή οι επαναλαμβανόμενες ηλεκτρονικές παραβιάσεις σε ιατρικούς οργανισμούς [13], μας οδηγούν στο συμπέρασμα ότι επουδενί δεν μπορούμε να βασιστούμε στην ανθρώπινη καλοήθεια, όταν αφορά ζωές ανθρώπων, ακόμα και όταν αφορά ζωές μικρών παιδιών.

Σημαντικό μερίδιο ευθύνης βέβαια πρέπει να αποδοθεί και στους οργανισμούς οι οποίοι είναι υπεύθυνοι για τη διαφύλαξη ιατρικών δεδομένων, γεγονός το οποίο επιβεβαιώνεται από σχετικές μελέτες [13]. Ταυτόχρονα παρατηρείται συλλογική προσπάθεια αποτροπής του φαινομένου, μέσα από δημοσιευμένες επιστημονικές μελέτες, όπου προτείνονται λύσεις σχετικά με την καταπολέμηση των παθογενειών που περιγράφηκαν [14].

Τα παραπάνω μας οδηγούν στο συμπέρασμα ότι θεωρείται εποικοδομητική προσθήκη η εισαγωγή επιπλέον μηχανισμών και κινήτρων, ώστε να προάγεται περαιτέρω η ομαλή συμμετοχή στη διαδικασία επίτευξης ομοφωνίας. Για αυτούς τους λόγους εισάγαμε λειτουργικότητα Κρυπτονομίσματος, με την αυτοσχέδια ονομασία “Health-Coin”. Η λειτουργικότητα αυτή συνεισφέρει σε δύο ύψιστης σημασίας τομείς: Οι συμμετέχοντες ενθαρρύνονται να συμμετέχουν χωρίς να είναι απαραίτητο να μεταφέρουν εικόνες, και η εισαγωγή ενός προγράμματος επιβράβευσης τους προτρέπει να δρουν σύμφωνα με τους κανόνες που διέπουν τη πλατφόρμα. Σε μελλοντική υλοποίηση με σημαντικές βελτιώσεις και παροχές του συστήματος, το νόμισμα θα μπορούσε να έχει ποικίλες χρήσεις, πόσο μάλλον σε περίπτωση υιοθέτησης της πλατφόρμας από μεγάλους ιατρικούς οργανισμούς, όπου σε τέτοια περίπτωση θα ήταν απαραίτητη περαιτέρω μελέτη των χρηματοοικονομικών χρήσεων του νομίσματος. Ένα BC χωρίς τον παράγοντα των κρυπτονομισμάτων προσφέρει περιορισμένες λειτουργίες, αλλά ταυτόχρονα εισάγει παράγοντες οι οποίοι ενδέχεται να δράσουν ως ευπάθειες με σκοπό την παραβίαση του.

B'.4 Σενάρια Χρήσης και τύποι χρηστών

B'.4.1 Χρήστες

Για το σύστημα προδιαγράφεται μοναδική ομάδα ομότιμων χρηστών, χωρίς ύπαρξη διαχειριστών. Κατά την εγγραφή τους στο σύστημα, αποδίδεται στους χρήστες μοναδικό ζευγάρι δημοσίου και ιδιωτικού κλειδιού, τα οποία χρησιμοποιούνται σε αρκετές πτυχές του συστήματος, όπως η ηλεκτρονική υπογραφή των χρονοσημάνσεων, η κρυπτογράφηση των εικόνων, αλλά και φυσικά η αποκρυπτογράφηση τους. Για το σύστημα προδιαγράφονται δύο ξεχωριστές λειτουργικότητες που αφορούν τους χρήστες: την αποστολή και παραλαβή εικόνων.

B'.4.2 Αποστολή εικόνων

Ο χρήστης επιλέγει την εικόνα προς αποστολή, και ακόμα πριν τη μεταφόρτωση του (Upload), αξιοποιείται η συνάρτηση κατακερματισμού (hash function) για την παραγωγή Τιμής Κατακερματισμού (hash), η οποία αποδίδεται στην εικόνα και είναι μοναδική για αυτό. Οποιαδήποτε απειροελάχιστη τροποποίηση του περιεχομένου του block επιφέρει πλήρη αλλαγή της τιμής κατακερματισμού [2]. Στη συνέχεια η τιμή αυτή προστίθεται ως χαρακτηριστικό δοσοληψίας (transaction) στο BC, καθιστώντας την αποθήκευση της τιμής μόνιμη και διαθέσιμη για προβολή για οποιαδήποτε περίπτωση. Στη συνέχεια ο αποστολέας, με αξιοποίηση Ασύμμετρης Κρυπτογραφίας, κρυπτογραφεί την εικόνα με το δημόσιο κλειδί του παραλήπτη, έτσι ώστε μόνο ο παραλήπτης να έχει πρόσβαση σε αυτό, και εν συνεχεία η εικόνα αποστέλλεται στο λογαριασμό (wallet) που αντιπροσωπεύεται από το δημόσιο κλειδί του παραλήπτη. Κατά την παραλαβή του κρυπτογραφημένου αρχείου, ως πρώτο βήμα καταγράφεται η αποκρυπτογράφηση του, με τη χρήση του ιδιωτικού κλειδιού του παραλήπτη. Εν συνεχεία, ο παραλήπτης έχει δικαιώματα προβολής του Block και συνεπώς της εικόνας. Στο προτεινόμενο σύστημα ο έλεγχος των τιμών κατακερματισμού και η αποκρυπτογράφηση της γίνεται αυτόματα, καθώς η εκάστοτε εικόνα αποθηκεύεται μέσα στα block, συνεπώς περνά από τη διαδικασία εξακρίβωσης που ισχύει για όλα τα δεδομένα προέρχονται στο block.

B'.5 Απαιτήσεις Συστήματος

Οι σχετιζόμενες λειτουργικές απαιτήσεις με το σύστημα αφορούν έναν συγκεκριμένο παράγοντα:

1. Ο χρήστης οφείλει να διαθέτει τον κατάλληλο φυσικό εξοπλισμό (hardware) με δυνατότητα εκτέλεσης της εφαρμογής.

B'.6 Ιδιαιτερότητες του προτεινόμενου Blockchain

Για την εκπόνηση της παρούσας υλοποίησης, στο πλαίσιο της Πτυχιακής Εργασίας, επιλέχθηκαν λύσεις οι οποίες παρέχουν παρόμοιες λειτουργικότητες με τα υπάρχοντα BC, με την παρουσία όμως αξιοσημείωτων διαφορών. Ως σημείο αναφοράς και σύγκρισης επιλέγουμε το Bitcoin, όχι επειδή αποτελεί το BC με τη μεγαλύτερη αξία παγκοσμίως, αλλά επειδή διαθέτει σχεδόν πανομοιότυπο αλγόριθμο ομοφωνίας με το προτεινόμενο σύστημα. Η σύγκριση κρίνεται πολλά υποσχόμενη, καθώς μέσα από την ανάδειξη των θεμελιωδών διαφορών που υπάρχουν ενδέχεται να προκύψουν γόνιμα συμπεράσματα, αλλά και προτάσεις για μελλοντικές προεκτάσεις.

B'.6.1 Τρόπος Αποθήκευσης

Η πρώτη σημαντική διαφορά που παρατηρείται αφορά τον τρόπο με τον οποίο αποθηκεύονται δεδομένα στις δύο υλοποιήσεις. Στη περίπτωση του BTC, το BC διανέμεται κατανεμημένα σε όλους τους χρήστες, ενώ στην περίπτωση μας, επιλέχθηκε μια πιο κεντροποιημένη (centralised) λύση, η οποία προσομοιώνει τις προαναφερόμενες λειτουργίες του BTC, χωρίς την προσθήκη ενός Κατανεμημένου Δικτύου. Για να επιτευχθεί μια τέτοια υλοποίηση, επιλέχθηκε η βάση δεδομένων PouchDB [15], η οποία περιγράφεται παρακάτω.

B'.6.2 Client Side: PouchDB

Η PouchDB αποτελεί υλοποίηση Βάσης Δεδομένων βασισμένη στη γλώσσα προγραμματισμού Javascript, ενώ έμπνευση για τη δημιουργία της αποτέλεσε η CouchDB [16]. Η συγκεκριμένη Βάση έχει αρκετά ενδιαφέροντα χαρακτηριστικά, όπως το γεγονός ότι μπορεί να λειτουργήσει μέσα στα πλαίσια του Φυλλομετρητή, αλλά και να συγχρονιστεί με την CouchDB, η οποία περιγράφεται παρακάτω. Στη βάση αποθηκεύεται τοπικά για τον χρήστη το Blockchain, μέσα από το οποίο ο χρήστης έχει το δικαίωμα να προβεί σε έγκυρες δοσοληψίες. Σε αυτή την περίπτωση, δημιουργείται μια νέα έκδοση του BC, η οποία αποθηκεύεται στην τοπική βάση που έχουμε δημιουργήσει, η οποία είναι υπεύθυνη για τη διανομή της καινούργιας κατάστασης του BC σε όλους τους συμμετέχοντες, ενώ η κατάσταση της βάσης που αποστέλλεται προς αποθήκευση οφείλει να τηρεί όλους τους κανόνες διεξαγωγής δοσοληψιών, καθώς σε περίπτωση μη έγκυρων δεδομένων η δοσοληψία δεν θα προχωρήσει.

B'.6.3 Peer-2-Peer Networking

Η αρχιτεκτονική δικτύου Peer2Peer του BTC επιτρέπει σε όλους τους συμμετέχοντες να διατηρούν αντίγραφα δεδομένων που αφορούν την αλυσίδα, για την περίπτωση ανάγκης επαλήθευσης, γεγονός που αποτελεί βασικό χαρακτηριστικό της συγκεκριμένης δομής. Στο πλαίσιο της παρούσας εργασίας επιλέξαμε να μην δημιουργήσουμε πολλαπλούς αποκεντροποιημένους κόμβους (nodes), καθώς μια τέτοια μορφή δικτύωσης απαιτεί πολλαπλούς πόρους σε Υλικό (hardware), σε συνδυασμό με το πλήθος παραμετροποιήσεων και ρυθμίσεων που απαιτούνται σε επίπεδο Λογισμικού (software) και Υλικού. Με τη χρήση του συνδυασμού Pouch και Couch, δημιουργούμε δίκτυωση με παρόμοια χαρακτηριστικά με ένα κατακεντρωμένο δίκτυο, παρέχοντας ιδανικές συνθήκες για την πρώτη έκδοση της εφαρμογής (Alpha Stage). Βέβαια χωρίς την αρχιτεκτονική P2P, στην οποία όλοι οι ενεργοί κόμβοι είναι σε αναμονή και περιμένουν να συμμετάσχουν στη διαδικασία του mining, παρατηρούμε μια σημαντική διαφορά. Όταν κάποιος χρήστης του συστήματος καταθέτει μια δοσοληψία, ενώ εκείνη ελέγχεται, δεν ελέγχεται από άλλους χρήστες του συστήματος, καθώς ο χρήστης έχει τη δυνατότητα να προβεί στη διαδικασία του mining, κατέχοντας μόνο τη δική του υπολογιστική δύναμη. Ταυτόχρονα αφαιρείται από το σύστημα η έννοια του “αγώνα δρόμου” για το ποιος θα καταφέρει να καταθέσει τη σωστή τιμή κατακερματισμού που αντιστοιχεί στο μελλοντικό block προς πρόσθεση στην αλυσίδα

B'.6.4 Timestamp server

Αξιοσημείωτη διαφορά αποτελεί επίσης το γεγονός ότι το BTC διατηρεί και λειτουργεί αποκεντρωμένο εξυπηρετητή έκδοσης πιστοποιητικών Χρονοσήμανσης, ενώ το προτεινόμενο σύστημα χρησιμοποιεί την τοπική ώρα που λαμβάνει από την εκάστοτε συσκευή στην οποία λειτουργεί, μέσω του Φυλλομετρητή (browser). Η ύπαρξη αποκεντρωμένου και ανεξάρτητου Εξυπηρετητή αποτελεί σημαντική προσθήκη όσον αφορά στην εγκυρότητα και το επίπεδο ασφάλειας των πιστοποιητικών που εκδίδονται, με αποτέλεσμα η υλοποίηση του ως μελλοντική επέκταση στο προτεινόμενο σύστημα να θεωρείται μια απαραίτητη και χρήσιμη προσθήκη.

B'.6.5 Δέντρα Merkle στα Block

Το BTC αξιοποιεί κρυπτογραφικά δέντρα Merkle, τα οποία αποτελούν δομή όπου αποθηκεύονται τιμές κατακερματισμού δεδομένων. Όλες οι δοσοληψίες ενός block αποτελούν τα φύλλα του δέντρου, ενώ η ρίζα που προκύπτει μπορεί να χρησιμοποιηθεί ως μέσο επαλήθευσης όλων των δοσοληψιών. Το κύριο πλεονέκτημα χρήσης της συγκεκριμένης τεχνικής προκύπτει από το γεγονός ότι μειώνει δραστικά τον απαιτούμενο χρόνο επαλήθευσης των δοσοληψιών. Στη προτεινόμενη υλοποίηση, όπου δεν υπάρχει αφενός μεγάλος όγκος δοσοληψιών και αφετέρου μεγάλος αριθμός block, ο έλεγχος εγκυρότητας τους δεν παρουσιάζει μεγάλο κόστος σε υπολογιστική

ενέργεια. Συμπερασματικά, αποφασίσαμε να μην χρησιμοποιήσουμε τη συγκεκριμένη τεχνική, καθώς ενδείκνυνται για εφαρμογές BC με τεράστια δημόσια χρήση.

B'.6.6 In-browser Mining

Στη περίπτωση του BTC, όπου η υπολογιστική δυσκολία του φτάνει σε δυσθεώρητα ύψη, και με συνολική κατανάλωση ρεύματος σχεδόν εφάμιλλη με μία χώρα όπως η Ουκρανία [17], οι συμμετέχοντες στο δίκτυο χρησιμοποιούν Κάρτες Γραφικών (Graphics Processing Unit-GPU), οι οποίες δύνανται να υποστηρίξουν πολύπλοκους υπολογισμούς τέτοιου βεληγεκού [17]. Στην παρούσα όμως περίπτωση όπου η υπολογιστική δυσκολία είναι παραμετροποιήσιμη και ελεγχόμενη, αξιοποιούμε τον Φυλλομετρητή, ο οποίος με τη σειρά του αξιοποιεί τον Κεντρικό Επεξεργαστή (Central Processor Unit-CPU), για τη διενέργεια των απαιτούμενων πράξεων. Κρίνεται μείζονος σημασίας να σημειωθεί ότι σε περίπτωση υπέρμετρης αύξησης της μαθηματικής δυσκολίας ο Φυλλομετρητής δεν ανταποκρίνεται, καθώς το επίπεδο των μαθηματικών πράξεων υπερκερνά τις υπολογιστικές δυνατότητες του Κεντρικού Επεξεργαστή.

B'.7 Απόδοση κλειδιών

Όπως προαναφέρθηκε, η απόδοση δημοσίων και ιδιωτικών κλειδιών λαμβάνει χώρα κατά την είσοδο του χρήστη στην ιστοσελίδα. Μετά την ανάθεση, γίνεται αποθήκευση του ζευγαριού κλειδιών στην κρυφή μνήμη (cache) του Browser, έτσι ώστε σε περίπτωση επόμενης σύνδεσης ο χρήστης να είναι συνδεδεμένος με το ίδιο ζευγάρι κλειδιών. Σε περίπτωση εκκαθάρισης της μνήμης cache, θα υπάρξει επανάληψη της διαδικασίας απόδοσης ζευγαριού κλειδιών. Κατά την είσοδό του ο χρήστης έχει τη δυνατότητα να αποθηκεύσει το ιδιωτικό του κλειδί σε μορφή αρχείου κειμένου (txt file), το οποίο μπορεί να χρησιμοποιήσει μετέπειτα για μεταφόρτωση του ζευγαριού του στη σελίδα από οποιαδήποτε άλλη συσκευή ή διαφορετικό Φυλλομετρητή, έχοντας ουσιαστικά πρόσβαση στο περιεχόμενο που ενδέχεται να έχει αποσταλεί στη δημόσια διεύθυνση του. Αυτό συμβαίνει καθώς, η κατοχή του ιδιωτικού κλειδιού ενός ζευγαριού επιτρέπει τον επανυπολογισμό της δημόσιας διεύθυνσης, δηλαδή του δημοσίου κλειδιού του ζευγαριού.

Η αντίστοιχη διαδικασία υπολογισμού του ιδιωτικού κλειδιού γνωρίζοντας το δημόσιο κλειδί κρίνεται αδύνατη με τις σημερινές μαθηματικές γνώσεις που διαθέτει η ανθρωπότητα, καθώς υπάγεται στο ανοιχτό πρόβλημα που ονομάζεται DLP (Discrete Log Problem) και ουσιαστικά αποτελεί μια από τις βασικές αρχές της Κρυπτογραφίας Ελλειπτικών Καμπυλών, την οποία αξιοποιούμε για την δημιουργία ζευγαριών κλειδιών, ηλεκτρονικών υπογραφών και εν τέλει κρυπτογράφηση των μεταφερόμενων δεδομένων[5]. Παρόμοια τακτική απόδοσης ζευγαριών κλειδιών ακολουθείται από κολοσσούς του κλάδου όπως το Monero και το Arweave [18,19]. Με την παραπάνω τεχνική, τα ζευγάρια χρηστών και συνθηματικών δεν αποθηκεύονται σε κεντρική Βάση

Δεδομένων, γεγονός που ενδυναμώνει την αποκεντροποίηση του συστήματος. Ταυτόχρονα, με την απουσία κάποιας κεντρικής Βάσης, εξαλείφεται ο κίνδυνος εισαγωγής ενός single point of failure, καθώς σε περίπτωση παραβίασης της Βάσης, οι λογαριασμοί των χρηστών θα βρίσκονταν ανεπανόρθωτα εκτεθειμένοι.

B'.8 Ταυτότητα και προστασία των χρηστών

Οι χρήστες του συστήματος παραμένουν ανώνυμοι, καθώς δεν υπάρχει κανένας άμεσος συσχετισμός μεταξύ ζευγαριών-κλειδιών και φυσικών προσώπων. Ο χρήστης έχει δικαίωμα δημιουργίας και χρήσης πολλαπλών ζευγαριών κλειδιών, με αποτέλεσμα να δυσχεραίνεται περαιτέρω ο συσχετισμός δημόσιου κλειδιού (public key) με τον ίδιο.

B'.9 Απευθυνόμενο κοινό

Η παρούσα διαδικτυακή εφαρμογή, στην τρέχουσα μορφή της, απευθύνεται για χρήση σε μεμονωμένα άτομα, τα οποία επιθυμούν να μεταφέρουν τις ιατρικές τους εικόνες.

Η εφαρμογή δεν προορίζεται για χρήση από ιατρικούς οργανισμούς, καθώς σε μια τέτοια περίπτωση, κρίνεται απαραίτητη η εφαρμογή αυστηρότερων πρωτοκόλλων ασφαλείας, καθώς και πλήρης συμμόρφωση με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (General Data Protection Regulation). Η τεχνολογία Blockchain θα μπορούσε να χαρακτηριστεί αντιδιαμετρική με το GDPR, καθώς ο GDPR επιβάλλει την ύπαρξη δυνατότητας διαγραφής δεδομένων, εφόσον ο χρήστης το επιθυμεί, ενώ ένα από τα κυριότερα χαρακτηριστικά του BC είναι η αμεταβλητότητα των δεδομένων του (immutability).

Ταυτόχρονα, ο GDPR απαιτεί δυνατότητα διόρθωσης, δυνατότητα επιβολής περιορισμού επεξεργασίας, αλλά και άλλους κανόνες, οι οποίοι έρχονται σε πλήρη αντιπαράθεση με τις συνήθειες αρχές λειτουργίας που ακολουθούνται από τα Public Permissionless Blockchains. Συνεπώς, κρίνεται απαραίτητη η θεσμοθέτηση ενός νομοθετικού πλαισίου, το οποίο θα επιτρέπει ενδεχομένως την παραίτηση του χρήστη, μετά από συγκατάθεση, από κάποια από τα αναφερόμενα δικαιώματα του GDPR. Σε διαφορετική περίπτωση, οι οργανισμοί θα αναγκαστούν να στραφούν σε χρήση κεντροποιημένων υλοποιήσεων BC, με όποια μειονεκτήματα, ιδιαιτερότητες αλλά και πλεονεκτήματα αυτά περιλαμβάνουν.

Β'.10 Ποιοτική και Ποσοτική σύγκριση συστήματος

Μετά τη σύγκριση του προτεινόμενου συστήματος με γνωστές εφαρμογές blockchain, θεωρείται απαραίτητη μια επιπλέον σύγκριση με τις ευρύτερες πρακτικές που ακολουθούνται στον τομέα αποθήκευσης ιατρικών δεδομένων, δηλαδή την χρήση βάσεων δεδομένων. Η σύγκριση αφορά πολλαπλές πτυχές των υλοποιήσεων, δίνοντας ιδιαίτερη έμφαση στη μεθοδολογία με την οποία αποθηκεύονται, προστατεύονται, διανέμονται και χρησιμοποιούνται τα ιατρικά δεδομένα των χρηστών, σε κάθε μια από αυτές. Επιπλέον παρατίθενται συγκρίσεις σχετικά με το κόστος υλοποίησης και την χρηστικότητα των συγκεκριμένων τεχνολογιών.

Β'.10.1 Αποθήκευση

Στις περιπτώσεις των περισσότερων ιατρικών οργανισμών χρησιμοποιούνται δύο διαφορετικοί τρόποι αποθήκευσης των ιατρικών δεδομένων: είτε ενοικιάζεται εξοπλισμός αποθήκευσης από εταιρείες και η αποθήκευση γίνεται στον χώρο του οργανισμού, είτε ενοικιάζεται χώρος στο “νέφος” από αντίστοιχες εταιρείες, όπου τους παρέχεται ανεξάρτητος χώρος για αποθήκευση των δεδομένων. Πραγματικός κάτοχος των δεδομένων, αλλά και υπεύθυνος για τη διατήρησή τους, είναι ο εκάστοτε ιατρικός οργανισμός. Στη περίπτωση του προτεινόμενου συστήματος τα δεδομένα αποθηκεύονται κρυπτογραφημένα στο κατανεμημένο δίκτυο από όλους τους χρήστες του blockchain, ενώ μόνο ο ιδιοκτήτης των αρχείων έχει πρόσβαση σε αυτά, αλλά είναι και ο μοναδικός υπεύθυνος για την διατήρηση της ασφάλειάς τους, που συνεπάγεται με την προστασία του ιδιωτικού του κλειδιού. Το γεγονός αυτό τον καθιστά πραγματικό κάτοχο των δεδομένων του, με δυνατότητες προβολής και διαμοιρασμού τους σε οποιαδήποτε οντότητα εκείνος επιθυμεί.

Β'.10.2 Ασφάλεια

Προχωρώντας στον τομέα της ασφάλειας, παρατηρούμε ότι σε περιπτώσεις βάσεων δεδομένων που έχουν δημιουργηθεί με γνώμονα την ασφάλεια (στρατιωτικού, κρατικού επιπέδου), τα επίπεδα ασφάλειας που παρέχονται είναι εφάμιλλα, αν όχι καλύτερα, από τη τεχνολογία blockchain. Αξίζει όμως να εστιάσουμε σε συγκεκριμένες περιπτώσεις όπου η ασφάλεια των συγκεκριμένων υποδομών ενδέχεται να παραβιαστεί. Το κοινό χαρακτηριστικό όλων των σύγχρονων βάσεων δεδομένων είναι ότι απαιτούν, στις περισσότερες υλοποιήσεις τους, την ύπαρξη διαχειριστικού δυναμικού για τη συντήρηση και αναβάθμιση τους, ενώ ταυτόχρονα υπάρχουν πολλαπλές οντότητες με διάφορα επίπεδα πρόσβασης σε πολλαπλά αρχεία. Το γεγονός αυτό ενδέχεται να θέσει την ασφάλεια του συνόλου του συστήματος σε κίνδυνο, ανάλογα με τα επίπεδα διαπίστευσης μπορεί να έχει ο εκάστοτε χρήστης, από τον οποίο έχουν εκμαιευτεί τα στοιχεία πρόσβασης του από τρίτες οντότητες. Στη περίπτωση του Blockchain μόνο οι πραγματικοί κάτοχοι των δεδομένων τους έχουν πρόσβαση σε αυτά, καθώς μόνο εκείνοι έχουν το

ιδιωτικό κλειδί που παρέχει πρόσβαση στα αρχεία. Η τεχνολογία blockchain δεν παρουσιάζει ευπάθειες που μπορούν να δράσουν ως single point of failure, καθώς η παράκαμψη του αλγορίθμου Απόδειξης Μόχθου απαιτεί τη συλλογή της πλειοψηφίας της υπολογιστικής δύναμης που υπάρχει στο σύστημα, όπου σε περιπτώσεις με μεγάλο πλήθος συμμετεχόντων κρίνεται πολύ δύσκολο[5].

B'.10.3 Διανομή και χρήσεις

Στη περίπτωση των βάσεων δεδομένων, κάποιος χρήστης δεν γνωρίζει ποιές οντότητες έχουν πρόσβαση στα αρχεία τους, καθώς δεν ενημερώνονται για το γεγονός από τους ιατρικούς οργανισμούς, με αποτέλεσμα να μην είναι ξεκάθαρη η ροή της πληροφορίας, για τους άμεσα εμπλεκόμενους χρήστες. Αντιθέτως, η τεχνολογία blockchain επιτρέπει πλήρη απεικόνιση της ροής των πληροφοριών, καθώς όλες οι μεταφορές αρχείων καταγράφονται και διανέμονται ως δημόσια γνώση στο σύνολο των χρηστών μέσω της κοινοποίησης της υπάρχουσας αλυσίδας. Παράλληλα, με την παροχή ειδικών αδειών χρήσης των αρχείων, αποφεύγεται η παρεμπόδιση των κατάλληλων χρήσεων των δεδομένων από τους εκάστοτε οργανισμούς.

B'.10.4 Κόστος

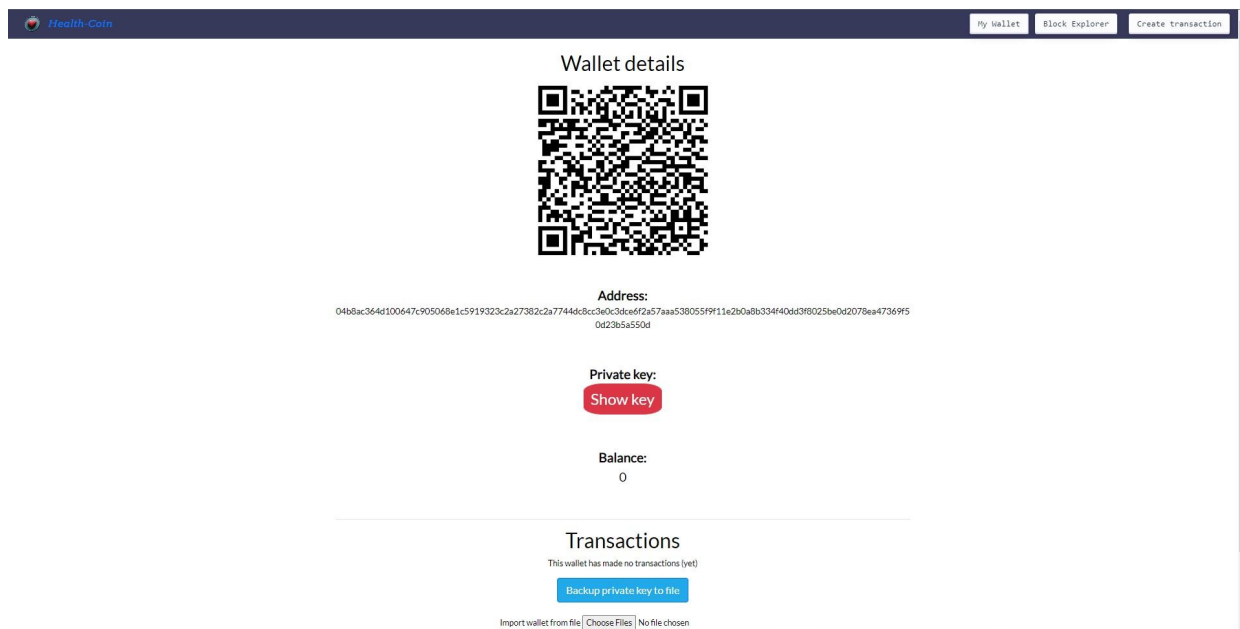
Οι παραδοσιακές βάσεις δεδομένων παρουσιάζουν μηδαμινές απαιτήσεις σε ενεργειακούς πόρους, σε σχέση με υλοποιήσεις Blockchain που αξιοποιούν τον αλγόριθμο Απόδειξης Μόχθου. Αυτό συμβαίνει καθώς σε δημόσιες υλοποιήσεις που έχουν σκοπό την επίτευξη ασφάλειας, οι αλγόριθμοι ομοφωνίας εκδίδουν μαθηματικές πράξεις υπέρμετρης δυσκολίας, των οποίων η λύση απαιτεί τεράστια ποσά ενέργειας για να βρεθεί [17], ενώ ταυτόχρονα απαιτεί μεγάλο χώρο αποθήκευσης από όλους τους χρήστες για τη διατήρηση του αντιγράφου της αλυσίδας.

B'.11 Περιορισμοί του συστήματος

Στη παρούσα μορφή του συστήματος, παρατηρούνται συγκεκριμένοι περιορισμοί που αφορούν το μέγεθος και τον τύπο των αρχείων που μεταφέρονται. Μετά από εκτεταμένες δοκιμές εικόνων με μέγεθος έως και 30 Megabyte, καταλήξαμε στο συμπέρασμα ότι ο εξυπηρετητής σε επίπεδο ανάπτυξης εφαρμογής που παρέχεται από την angular δεν διαθέτει την δυνατότητα να αποδώσει (render), τόσο μεγάλες εικόνες. Κατά την ανάπτυξη της εφαρμογής δόθηκε περισσότερο έμφαση στην ανταλλαγή εικόνων, ενώ θεωρείται αρκετά σημαντική η ανάγκη να υπάρχει δυνατότητα αποστολής και άλλων ειδών αρχείων (.pdf ή .zip).

Γ' Οθόνες εφαρμογής

Κατά την είσοδο στην εφαρμογή, ο χρήστης βλέπει τη δημόσια διεύθυνση που του αποδόθηκε, ενώ ταυτόχρονα μπορεί να προβεί σε προβολή ή λήψη του ιδιωτικού του κλειδιού, ακόμα και να μεταφορτώσει παλαιότερο ιδιωτικό κλειδί, από το οποίο η κρυπτογραφία ελλειπτικών καμπυλών μας επιτρέπει τον υπολογισμό της δημόσιας διεύθυνσης του.



Εικόνα 2. Αρχική οθόνη της εφαρμογής

Στη περίπτωση που ο χρήστης επιλέξει δημιουργία δοσοληψίας, μεταφέρεται στη παρακάτω σελίδα, όπου καλείται να επιλέξει διεύθυνση προς αποστολή, αλλά και να μεταφορτώσει την εικόνα που επιθυμεί να αποστείλει.

The screenshot shows the 'Create transaction' page of the Health-Coin wallet. The page has a dark blue header with the 'Health-Coin' logo and navigation links: 'My Wallet', 'Block Explorer', and 'Create transaction'. The main content area is titled 'Create transaction' with the subtitle 'Transfer images or make transactions!'. It contains several input fields: 'From address' (pre-filled with a long hexadecimal string), 'To address' (with a placeholder address), 'Amount' (set to '1'), and a 'File input field' with a 'Choose File' button. A blue button at the bottom is labeled 'sign & create transaction'. At the very bottom, it says 'Thesis of EMMANOUIL ANEZINIS'.

Εικόνα 3. Οθόνη Δημιουργίας Δοσοληψίας

Μετά την επικύρωση της δοσοληψίας, ο χρήστης καλείται να πατήσει το κουμπί της εικόνας με τίτλο “Start Mining”, έτσι ώστε η δοσοληψία του να προστεθεί μέσα στο επόμενο μπλοκ προς δημιουργία.

The screenshot shows the 'Pending transactions' page of the Health-Coin wallet. The header is the same as in the previous image, but the 'Create transaction' button is now disabled. A green notification bar at the top says 'Transaction created successfully!'. Below it, the title 'Pending transactions' is followed by the text 'These transactions are waiting to be included in the next block. Next block is created when you start the mining process.' A table lists the pending transaction:

#	From	To	Amount	Payload	Timestamp	Valid?
0	044386f5b22cc52e4... (That's yours)	0416a75e8fec9d73c...	1		1664454876143 Sep 29, 2022, 15:34	

Below the table is a blue button labeled 'Start mining'. At the bottom, it says 'Thesis of EMMANOUIL ANEZINIS'.

Εικόνα 4. Οθόνη εκκρεμών δοσοληψιών

Σε περίπτωση επιθυμίας προβολής όλων των υπαρχόντων μπλοκ, ο χρήστης μπορεί να επιλέξει το κουμπί “Block Explorer”, και θα μεταφερθεί στην οθόνη που απεικονίζεται παρακάτω. Από εκεί έχει τη δυνατότητα εξέτασης των δημόσιων διευθύνσεων που εμπλέκονται στη δοσοληψία, αλλά και το σύνολο όλων των συναλλαγών στο οποίο έχει εμπλακεί η επιλεγμένη δημόσια διεύθυνση.

Health-Coin

My WalletBlock ExplorerCreate transaction

Blocks on chain

Each card represents a block on the chain. Click on a block to see the transactions stored inside.

Block 1 (Genesis block)

Hash

62044430e173de377a5f1591870063bc9...

Hash of previous block

0

Nonce

0

Timestamp

1664450000000

Block 2

Hash

0853776cc193f6f282f86a35fa35e63b6d8...

Hash of previous block

62044430e173de377a5f1591870063bc9...

Nonce

26

Timestamp

1664454896036

Transactions inside block 2

#	From	To	Amount	Payload	Timestamp	Valid?
0	044386f5b22cc52e4... (That's yours!)	0416a75e8fec9d73c...	1		1664454876143 Sep 29, 2022, 13:34	
1	System	044386f5b22cc52e4... (That's yours!)	100 (Block reward)		1664454896036 Sep 29, 2022, 13:34	

Εικόνα 5. Οθόνη προβολής υπαρχόντων συστοιχιών

Δ' Τεχνολογίες Υλοποίησης

Δ'.1 Javascript

Η Javascript (JS) είναι ουσιαστικά η κύρια γλώσσα προγραμματισμού που αξιοποιήθηκε για την περάτωση της εργασίας [20]. Η JS είναι μία ελαφριά, διερμηνευόμενη (Interpreted) γλώσσα προγραμματισμού, με δυνατότητες αντικειμενοστρέφειας, η οποία υποστηρίζει ταυτόχρονα συναρτήσεις πρώτης τάξης (first class functions). Η JS αξιοποιείται ως scripting γλώσσα από τη συντριπτική πλειοψηφία των διαδικτυακών ιστοσελίδων. Η JS υποστηρίζει πολλαπλά προγραμματιστικά δόγματα (multi paradigm), ενώ η κληρονομικότητα στην γλώσσα εκπονείται μέσω προτύπων (prototype-based programming).

Δ'.2 Typescript

Η Typescript αποτελεί αντικειμενοστραφή γλώσσα προγραμματισμού με χαρακτηριστικά ανοικτής πρόσβασης (open source), η οποία διατηρείται, υποστηρίζεται και παρέχεται από τη Microsoft [21]. Ουσιαστικά αποτελεί επέκταση της JS, καθώς προσθέτει λειτουργικότητες όπως προσθήκη τύπου δεδομένων (data types), δυνατότητα ελέγχου τύπων μεταβλητών (type checking), αλλά και δυνατότητα να διατηρούμε συναρτήσεις (functions) και αντικείμενα (objects) ιδιωτικά (private) και ασφαλή από οποιαδήποτε παραποίηση. Αποτελεί ένα υπερσύνολο της JS, το οποίο αξιοποιούμε για καλύτερη συγγραφή, διατήρηση και διόρθωση του κώδικα.

Δ'.3 Angular

Η Angular είναι μία πλατφόρμα ανάπτυξης εφαρμογών (Development Platform) [22], η οποία βασίζεται στην Typescript. Αποτελεί Πλαίσιο Ανάπτυξης Εφαρμογών (development framework) το οποίο βασίζεται στη χρήση Συνιστωσών (component-based), όπου το κάθε συστατικό αποτελεί ένα κομμάτι της εφαρμογής. Η συγκεκριμένη προσέγγιση προάγει την ευκολία δοκιμών, αναβαθμίσεων και ελέγχου, ενώ καθιστά τον κώδικα περισσότερο οργανωμένο, και συνεπώς αναγνώσιμο. Η Angular παρέχει πληθώρα έτοιμων βιβλιοθηκών, οι οποίες μπορούν να χρησιμοποιηθούν για πολλαπλές δραστηριότητες, όπως η δικτύωση της εφαρμογής, η διαχείριση Φορμών (forms management) και η επικοινωνία μεταξύ Χρήστη (client) και διακομιστή (server). Η Angular συνιστάται για κατασκευή κλιμακώσιμων εφαρμογών (scalable), όπου η υλοποίηση τους μπορεί να επιτευχθεί από μικρότερες ομάδες προγραμματιστών.

Δ'.4 Bootstrap

Το Bootstrap αποτελεί framework της CSS (Cascading Style Sheets) [23], το οποίο αξιοποιείται για υλοποίηση του front-end. Διαθέτει κώδικα ανοιχτής πρόσβασης, ενώ παρέχει πρότυπα (templates) για υλοποιήσεις HTML, CSS αλλά και JS όπως κουμπιά (buttons), φόρμες (forms) και άλλα στοιχεία που αξιοποιούνται στην διεπαφή χρήστη (user interface). Η χρήση της συγκεκριμένης τεχνολογίας αποδείχθηκε άκρως αποδοτική, καθώς παρέχει εύχρηστες βιβλιοθήκες, οι οποίες μειώνουν δραματικά τον απαιτούμενο χρόνο για την ανάπτυξη της εφαρμογής. Επιπλέον, το Bootstrap επικεντρώνεται στο δόγμα Responsive Web Design, όπου οι εφαρμογές σχεδιάζονται με τρόπο ώστε να λειτουργούν απρόσκοπτα ανεξάρτητα από το αν η πρόσβαση γίνεται από PC ή ηλεκτρονική κινητή συσκευή.

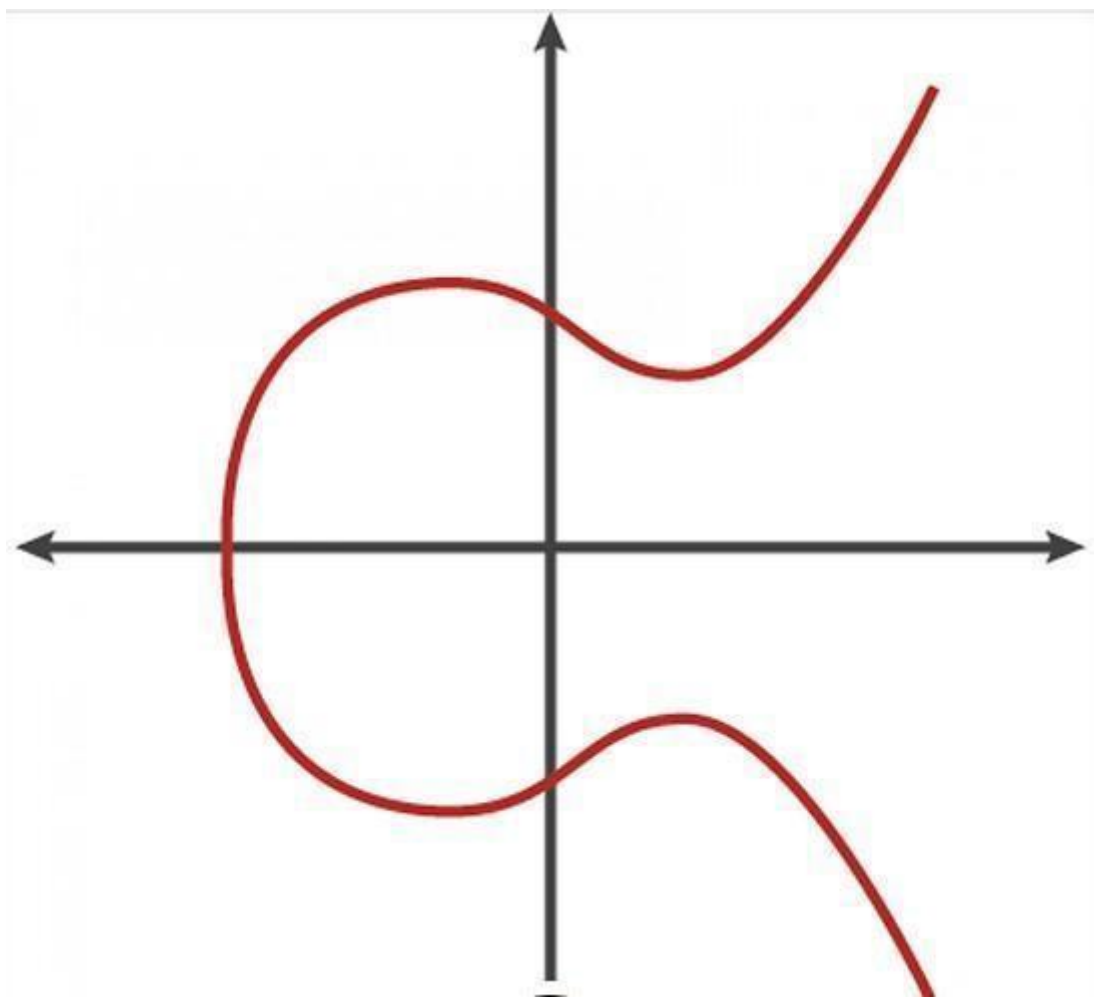
Δ'.5 Elliptic Curve Cryptography

Όπως προαναφέρθηκε, η Κρυπτογραφία με χρήση Ελλειπτικών Καμπυλών (Elliptic Curve Cryptography ή ECC) αποτελεί τον ακρογωνιαίο λίθο της εφαρμογής, όσον αφορά πολλαπλές πτυχές ασφαλείας του συστήματος [24]. Αξιοποιείται για παραγωγή (generate) ζευγαριών κλειδιών, υπογραφή ψηφιακών πιστοποιητικών, κρυπτογράφηση των δεδομένων προς μετάδοση και αρκετές άλλες λειτουργίες [25].

Η μορφή που χρησιμοποιήθηκε ανήκει στον τομέα της Ασύμμετρης Κρυπτογραφίας. Για την υλοποίησης της τεχνικής, επιλέχθηκε η ελλειπτική καμπύλη secp256k1, η οποία χρησιμοποιείται επίσης από το BTC για τις ίδιες λειτουργίες με το παρόν προτεινόμενο σύστημα. Όλες οι εξισώσεις που περιγράφουν ελλειπτικές καμπύλες ακολουθούν το πρότυπο $y^2 = x^3 + ax + b$, ενώ στην περίπτωση της secp256k1 όπου $a=0$ και $b=7$ καταλήγουμε στη μορφή $y^2 = x^3 + 7$.

Παρακάτω αποτυπώνεται η απεικόνιση της καμπύλης σε άξονες x,y . Το παραγόμενο ιδιωτικό κλειδί είναι μήκους 256 bit, ενώ επίσης στη συνέχεια παρουσιάζεται πίνακας με τα αντίστοιχα μεγέθη που απαιτούν άλλες μορφές κρυπτογράφησης, έτσι ώστε να επιτευχθεί το ίδιο επίπεδο ασφαλείας, σύμφωνα με τον αμερικανικό κρατικό οργανισμό NIST (National Institute of Standards and Technology).

Μάλιστα, ένας ακόμα αμερικανικός οργανισμός, η Επιτροπή Συστημάτων Εθνικής Ασφάλειας (Committee on National Security Systems), συνέστηνε, μέχρι και το 2015, την ECC με μήκος κλειδιού 256 bit για την προστασία κυβερνητικών εγγράφων με την ετικέτα “Απόρρητο”, ενώ για την προστασία δεδομένων με ετικέτα “Άκρως Απόρρητο” απαιτούνταν χρήση ECC με μήκος κλειδιού 384 bit, η οποία αξιοποιεί την εκδοχή ελλειπτικής καμπύλης P384.



Εικόνα 7. Αναπαράσταση Ελλειπτικής Καμπύλης secp256k1

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Εικόνα 8. Πίνακας Σύγκρισης Κρυπτογραφικών
Τεχνικών

Δ'.6 Secure Hash Algorithm 256

Ο αλγόριθμος κατακερματισμού που επιλέχθηκε για την παραγωγή των τιμών κατακερματισμού είναι ο SHA-256 (Secure Hash Algorithm). Αποτελεί μέρος της οικογένειας αλγορίθμων SHA-2, οι οποίοι σχεδιάστηκαν από την Εθνική Υπηρεσία Ασφαλείας της Αμερικής (National Security Agency), ενώ δημοσιεύθηκαν για πρώτη φορά το 2001.[26]

Ε' Σημαντικές βιβλιοθήκες και εργαλεία

Ε'.1 Node Package Manager

Το πρόγραμμα διαχείρισης πακέτων Node (Node Package Manager), αξιοποιήθηκε για την εγκατάσταση όλων των απαραίτητων βιβλιοθηκών. Παρακάτω παρουσιάζονται οι κύριες βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

Ε'.2 Crypto-js

Η βιβλιοθήκη crypto-js μας παρέχει έτοιμες συναρτήσεις και αλγορίθμους που μας επιτρέπουν την υλοποίηση του αλγορίθμου SHA-256 για τον υπολογισμό των απαραίτητων τιμών κατακερματισμού που χρειάζεται κάθε block πριν την προσθήκη του στο BC.

Ε'.3 Elliptic

Η βιβλιοθήκη elliptic μας παρέχει πρόσβαση σε συναρτήσεις που αξιοποιούν Κρυπτογραφία ελλειπτικών καμπυλών, τις οποίες και αξιοποιούμε για τη διασφάλιση της ασφάλειας του συστήματος. Αξιοσημείωτες δυνατότητες που παρέχονται από τη βιβλιοθήκη, αποτελούν η γεννήτρια παραγωγής ζευγαριών κλειδιών, η κρυπτογράφηση των δεδομένων με τα προαναφερθέντα κλειδιά, αλλά και η υπογραφή ηλεκτρονικών πιστοποιητικών με τον αλγόριθμο ECDSA (Elliptic Curve Digital Signature Algorithm).

ΣΤ' Μελλοντικές προεκτάσεις και συμπεράσματα

Η παραπάνω προτεινόμενη λύση υλοποιήθηκε σε τοπικό επίπεδο (localhost), ενώ ταυτόχρονα υπήρξε σχεδιασμός για προσαρμογή της εφαρμογής σε ιστοσελίδα στο Διαδίκτυο, έτσι ώστε να υπάρχει πραγματική χρήση, ακόμα και σε αυτό το στάδιο της ανάπτυξης. Παρακάτω παρουσιάζονται οι προδιαγραφές που θα απαιτούνταν για την προσαρμογή του συστήματος.

ΣΤ'.1.1 Ασφαλή Πρωτόκολλα επικοινωνίας

Καθώς η διαδικτυακή εφαρμογή βρίσκεται σε ιστοσελίδα προσβάσιμη από όλους, οφείλουμε να λάβουμε μέτρα για την διασφάλιση ασφαλούς μετάδοσης των δεδομένων από χρήστη σε χρήστη. Για την επίτευξη αυτού του στόχου, προτείνονται οι τεχνολογίες HTTPS (Hypertext Transfer Protocol Secure) και SSL (Secure Socket Layer). Ο συνδυασμός των προαναφερθέντων τεχνολογιών επιτρέπει την ασφαλή μετάδοση των δεδομένων σε κρυπτογραφημένη μορφή, αποτρέποντας την ανάγνωσή τους από κακοήθεις οντότητες, οι οποίες ενδέχεται να προσπαθήσουν να υποκλέψουν το περιεχόμενο προς μετάδοση.

ΣΤ'.1.2 Docker

Η τεχνολογία Docker [27], αποτελεί ανοιχτή πλατφόρμα και σύνολο εργαλείων που χρησιμοποιούνται για την ανάπτυξη, διάθεση και εκτέλεση (run) εφαρμογών. Το Docker μάς επιτρέπει να διατηρήσουμε την εφαρμογή μας σε ελαστικά απομονωμένο (isolated) περιβάλλον, το οποίο ονομάζεται Περιέκτης (container). Η τακτική αυτή επιτρέπει τον εύκολο διαμοιρασμό, διόρθωση και αναβάθμιση των container, ενώ εξασφαλίζει ότι όλα τα container δουλεύουν το ίδιο για όλους, ανεξαρτήτως τεχνικών χαρακτηριστικών.

ΣΤ'.1.3 Docker Compose

Η τεχνολογία Docker Compose [28] αξιοποιείται σε περιπτώσεις όπου η εφαρμογή αποτελείται από πολλά docker container. Με την χρήση της γλώσσας σειροποίησης YAML, μας επιτρέπεται η δημιουργία ενός αρχείου διαμόρφωσης (configuration file), το οποίο μπορεί να χρησιμοποιηθεί για την παράταξη, συνδυασμό και παραμετροποίηση πολλαπλών docker container.

ΣΤ'.1.4 Server Side: CouchDB

Η CouchDB αποτελεί Βάση Δεδομένων η οποία δημιουργήθηκε με γνώμονα την αλληλεπίδραση με το Διαδίκτυο, αλλά και τη δυνατότητα ασφαλούς και συνεχούς πρόσβασης στα αρχεία. Τα δεδομένα αποθηκεύονται σε μορφή JSON, ενώ πρόσβαση σε αυτά αποκτάται μέσω του Φυλλομετρητή μέσω αιτημάτων HTTP (HyperText Transfer Protocol) ή ερωτημάτων στη Βάση (queries). Ο συγχρονισμός που λαμβάνει χώρα μεταξύ των δύο βάσεων δεδομένων μάς επιτρέπει την προσομοίωση λειτουργικότητας ενός κατακευκμένου δικτύου. Η CouchDB εγκαθίσταται και παραμετροποιείται με τη βοήθεια περιέκτη, το οποίο διαχειριζόμαστε με το λογισμικό Portainer [29].

ΣΤ'.1.5 Portainer

Το Portainer [29] συνιστά εργαλείο διαχείρισης container και είναι συμβατό με περιέκτες Docker ή Kubernetes. Παρέχει πλήρως λειτουργική Διεπαφή Χρήστη για τη διαχείριση, παραμετροποίηση και ανάπτυξη (deploy) εφαρμογών που βασίζονται σε περιέκτες. Η τεχνολογία αυτή συνέβαλε σημαντικά στη μείωση του χρόνου ανάπτυξης της εφαρμογής, καθώς η εναλλακτική επιλογή, η οποία είναι η χρήση του Docker Compose μέσω γραμμής εντολών σε λειτουργικό σύστημα Linux, αποτελεί αρκετά πιο χρονοβόρα διαδικασία και ελλοχεύει κίνδυνος καθυστερήσεων κατά την ανάπτυξη του κώδικα.

ΣΤ'.2 Health-coin και συνήθειες πρακτικές

Εξετάζοντας τις κοινές πρακτικές που ακολουθούνται από δημόσια BC του κλάδου, σχετικά με τη χρήση των κρυπτονομισμάτων τους, παρατηρούμε ένα επικρατόν χαρακτηριστικό: την άμεση συσχέτιση των κρυπτονομισμάτων με χρηματική αξία. Κατά τη δημιουργία του νομίσματος, οι δημιουργοί καλούνται να ορίσουν συγκεκριμένους κανόνες που το διέπουν, οι οποίοι κανόνες στο σύνολο τους ορίζονται ως Token Economics (tokenomics). Ο όρος περιλαμβάνει το σύνολο των παραγόντων που καθορίζουν ένα νόμισμα, όπως την αρχική του αξία, τον αριθμό νομισμάτων προς παραγωγή, τον τρόπο διανομής των νομισμάτων, την προσφορά και ζήτηση, τους μηχανισμούς επιβράβευσης των χρηστών και τον ρυθμό απόσυρσης (Burning) των νομισμάτων. Παρόλο που προβήκαμε σε περιγραφή του όρου για λόγους κατανόησης, δεν θα προχωρήσουμε στην εφαρμογή του στο παρόν σύστημα. Η συγκεκριμένη επιλογή δικαιολογείται από το γεγονός ότι το παρόν σύστημα σχεδιάστηκε με γνώμονα την επαλήθευση των εξετάσεων, με σκοπό να αποτραπούν ιατρικά λάθη βασισμένα σε παραποιημένα δεδομένα, και όχι για την παραγωγή χρηματικού κέρδους. Στο παρόν σύστημα, τα νομίσματα αποδίδονται ως μια μορφή επίδειξης ευγνωμοσύνης, στους χρήστες του συστήματος και δεν αντανακλούν κάτι διαφορετικό.

Σε περίπτωση χρήσης της πλατφόρμας από ιατρικούς οργανισμούς, τα νομίσματα δύναται να χρησιμοποιηθούν ενδεχομένως ως μέσο ανταλλαγής για παροχή ιατρικών υπηρεσιών. Οι συμβαλλόμενοι με το σύστημα οργανισμοί θα μπορούσαν να παρέχουν τη διεξαγωγή απλών ιατρικών εξετάσεων, με αντάλλαγμα ένα συγκεκριμένο αριθμό νομισμάτων που θα μπορεί να ορίσει ο εκάστοτε οργανισμός. Ταυτόχρονα όμως ελλοχεύει ο κίνδυνος συσχέτισης των νομισμάτων, με το διακριτό χρηματικό κόστος των εξετάσεων, γεγονός που ενδέχεται να οδηγήσει σε αποτελέσματα τα οποία δεν είναι επιθυμητά, δηλαδή στη δημιουργία ισοτιμίας μεταξύ κρυπτονομίσματος και υλικών χρημάτων.

ΣΤ'.3 Έξυπνα Συμβόλαια

Για την πλήρη πραγμάτωση και ανάδειξη των δυνατοτήτων ενός BC, η προσθήκη λειτουργικότητας έξυπνων συμβολαίων θεωρείται απαραίτητη. Τα έξυπνα συμβόλαια (Smart Contracts) αποτελούν προγράμματα υπολογιστή ή πρωτόκολλα δοσοληψιών, τα οποία αποθηκεύονται στο BC, σε περίπτωση εκπλήρωσης προκαθορισμένων συνθηκών [30]. Οι ενέργειες που ενδέχεται να πραγματοποιηθούν από τα συμβόλαια περιλαμβάνουν εκτέλεση και τροποποίηση δοσοληψιών ταυτόχρονα με την καταγραφή προκαθορισμένων γεγονότων. Τα συμβόλαια αυτά αξιοποιούνται για επίτευξη συμφωνιών μεταξύ χρηστών, χωρίς την παρέμβαση ενδιάμεσης τρίτης οντότητας, παρέχοντας πλαίσιο στο οποίο αποφεύγονται κακόβουλες ενέργειες.

Στην περίπτωση του προτεινόμενου συστήματος, τα συμβόλαια αυτά θα χρησιμοποιούνταν για παροχή πρόσβασης σε εικόνες από τους ιδιοκτήτες τους σε τρίτες οντότητες. Στη παρούσα μορφή του συστήματος, μόνο ο παραλήπτης των εικόνων έχει δικαιώματα ανάγνωσης τους, ενώ με την προσθήκη των συμβολαίων, η εκχώρηση δικαιωμάτων ανάγνωσης μπορεί να συμβεί απρόσκοπτα. Ειδικά στην περίπτωση υιοθέτησης από ιατρικούς οργανισμούς, οι ασθενείς θα έχουν τη δυνατότητα παραχώρησης δικαιωμάτων σε πολλαπλές οντότητες, καθιστώντας των διαμοιρασμό των εικόνων τους μεταξύ οργανισμών εφικτό.

Αξίζει να σημειωθεί ότι, σε αντίθεση με τα παραδοσιακά γραπτά συμβόλαια, παρατηρείται η έλλειψη επαρκούς νομοθετικού πλαισίου, καθώς τα έξυπνα συμβόλαια δε θεωρούνται απαραίτητα δεσμευτικά από νομικής πλευράς [31]. Η Λευκορωσία αποτέλεσε το πρώτο κράτος που νομοθέτησε την ύπαρξη έξυπνων συμβολαίων το 2017 [32], ενώ έκτοτε ακολούθησαν και κάποιες πολιτείες της Αμερικής όπως η Αριζόνα [33].

ΣΤ'.4 Miners

Η εισαγωγή της λειτουργίας Κρυπτονομίσματος στο BC προδιαγράφει την ένταξη αυτής της 'ειδικής' ομάδας χρηστών, οι οποίοι διαθέτουν την υπολογιστική τους δύναμη στο σύστημα με αντάλλαγμα κάποιο μέρος από το συνολικό έπαθλο, που ορίζεται από τον Αλγόριθμο Ομοφωνίας. Δεν συμμετέχουν στη διαδικασία αποστολής εικόνων, ωστόσο ο ρόλος τους κρίνεται καθοριστικός για την ασφάλεια, την αποκεντροποίηση (decentralisation) και την επίτευξη κατανεμημένου σχεδιασμού. Με την ύπαρξη πολλαπλών ανεξάρτητων οντοτήτων έχουμε αυτόματα μεγαλύτερη διαθέσιμη υπολογιστική ισχύ, γεγονός που καθιστά την επίθεση πλειοψηφίας αρκετά δυσκολότερη. Επίσης, καθώς δεν υπάρχει περιορισμός ως προς τους συμμετέχοντες στον αλγόριθμο ομοφωνίας, μπορεί να προδιαγραφεί ακόμα και συμμετοχή ιατρικών φορέων, ιδρυμάτων και συλλόγων, με σκοπό την περαιτέρω αποκεντροποίηση του δικτύου, αλλά και την ύπαρξη πολλαπλών οντοτήτων των οποίων η ειλικρινής συμμετοχή συνεισφέρει στη θωράκιση της αξιοπιστίας του BC, αλλά και στην ασφαλή και απρόσκοπτη λειτουργία του.

ΣΤ'.5 Υποστηριζόμενοι τύποι αρχείων

Την παρούσα στιγμή υποστηρίζεται μόνο η μεταφόρτωση εικόνων, καθώς η εφαρμογή αναπτύχθηκε με αυτόν τον συγκεκριμένο προσανατολισμό, ενώ στο μέλλον προδιαγράφεται η υποστήριξη πολλαπλών τύπων αρχείων, για παράδειγμα αρχεία με κατάληξη “.pdf” ή “.zip”.

ΣΤ'.6 Συμπεράσματα

Μέσα από τη μελέτη που διεξήχθη στα πλαίσια της εργασίας, είμαστε σε θέση να παραθέσουμε συμπεράσματα που σχετίζονται την υλοποίηση και λειτουργικότητα της εφαρμογής. Το προτεινόμενο σύστημα προσανατολίζεται στην διατήρηση της ιδιωτικότητας και ιδιοκτησίας των αρχείων, ενώ παρέχει διαφάνεια και αμεταβλητότητα ως προς το περιεχόμενο που κατατίθεται. Εν κατακλείδι, οφείλουμε να σημειώσουμε ότι το κόστος λειτουργίας σε περιπτώσεις δημόσιας υιοθέτησης θα ήταν αρκετά υψηλό, ενώ οι απαιτούμενες ανάγκες για τη διατήρηση και λειτουργία του συστήματος θα αυξάνονταν γραμμικά με το πέρας του χρόνου.

Βιβλιογραφικές Αναφορές

1. Distributed Ledger Technology: beyond block chain (PDF) (Report). Government Office for Science (UK). January 2016. Retrieved 29 August 2016.
2. Knuth, D. The Art of Computer Programming, Sorting and Searching 1973;3:527. Addison-Wesley, Reading, MA., United States
3. Cope J. What's a Peer-to-Peer (P2P) Network? (<https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html>)
4. Haque AKM B, Rahman M. Blockchain Technology: Methodology, Application and Security Issue. Int J Comput Netw Secur 2020;20(2):21-30.
5. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. (<https://bitcoin.org/bitcoin.pdf>)
6. Merkle RC. Protocols for public key cryptosystems. In Symposium on Security and Privacy, IEEE Computer Society, 1980:p122-133.
7. Haber S, Stornetta WS. How to time-stamp a digital document. J Cryptol 1991;3(2):99-111.
8. Back A. Hashcash - A Denial of Service Counter-Measure (2002) (<http://www.hashcash.org/papers/hashcash.pdf>, 2002.) T.H.Π 26/08/22
9. Awati R. Consensus Algorithm. (<https://www.techtarget.com/whatis/definition/consensus-algorithm#:~:text=A%20consensus%20algorithm%20is%20a,involving%20multiple%20users%20or%20nodes.>) T.H.Π 24/08/22
10. Frankenfield J. 51% attack. Investopedia 2022. (<https://www.investopedia.com/terms/1/51-attack.asp>) T.H.Π 16/09/22
11. Milutinovic M. Cryptocurrency. Ekonomika 2018;64(1):105–122. doi:10.5937/ekonomika1801105M
12. <https://www.shrednations.com/white-papers/mishandling-medical-records-guide-to-preventing-disaster/?cn-reloaded=1> T.H.Π 23/08/22
13. Budke CA, Enko PJ. Physician Practice Cybersecurity Threats: Ransomware. Mo Med. 2020;117(2):102–104. PMID: 32308225
14. Kim L. Cybercrime, ransomware, and the role of the informatics nurse. Nursing 2020;50(3):63-65. doi: 10.1097/01.NURSE.0000654064.67531.c5.
15. <https://pouchdb.com/> T.H.Π 31/08/22
16. <https://couchdb.apache.org/> T.H.Π 29/07/22
17. <https://digiconomist.net/bitcoin-energy-consumption/> T.H.Π 29/07/22
18. <https://www.getmonero.org/> T.H.Π 28/07/22
19. <https://www.arweave.org/> T.H.Π 29/07/22

20. <https://www.javascript.com/> T.H.II 29/08/22
21. <https://www.typescriptlang.org/> T.H.II 29/08/22
22. <https://angular.io/> T.H.II 29/08/22
23. <https://getbootstrap.com/> T.H.II 29/08/22
24. Miller V. Use of elliptic curves in cryptography. *Advances in Cryptology — CRYPTO '85 Proceedings*. CRYPTO. Lecture Notes in Computer Science 1985;85:417–426. Penard W, van Werkhoven T. On the Secure Hash Algorithm family
25. Fact Sheet NSA Suite B Cryptography. *U.S. National Security Agency*. Archived from the original on 2009-02-07.)
26. <https://www.docker.com/> T.H.II 28/08/22
27. <https://docs.docker.com/compose/> T.H.II 27/08/22
28. <https://www.portainer.io/> T.H.II 31/08/22
29. Savelyev, Alexander, Contract Law 2.0: «Smart» Contracts As the Beginning of the End of Classic Contract Law (December 14, 2016). Higher School of Economics Research Paper No. WP BRP 71/LAW/2016, <http://dx.doi.org/10.2139/ssrn.2885241>
30. CleanApp (January 21, 2019). "Crypto's Founding Fallacy: How Mistakes in the 'Smart Contract' Genesis Block Weaken the Whole Chain" (<https://medium.com/cryptolawreview/cryptos-founding-fallacy-aaa151b795ff>) T.H.II 20/09/22
31. Makhovsky, Andrei (December 22, 2017). "Belarus adopts crypto-currency law to woo foreign investors" (<https://www.reuters.com/article/us-belarus-cryptocurrency-idUSKBN1EG0XQ>) T.H.II 20/09/22
32. "Arizona HB2417 - 2017 - Fifty-third Legislature 1st Regular" (<https://legiscan.com/AZ/text/HB2417/id/1588180>) T.H.II 20/09/22

Appendix

Κλάση Transaction

```
const SHA256 = require('crypto-js/sha256');
const EC = require('elliptic').ec;
const ec = new EC('secp256k1');

class Transaction {
  /**
   * @param {string} fromAddress
   * @param {string} toAddress
   * @param {number} amount
   * @param {string} payload
   */
  constructor(fromAddress, toAddress, amount, payload) {
    this.fromAddress = fromAddress;
    this.toAddress = toAddress;
    this.amount = amount;
    this.timestamp = Date.now();
    this.payload = payload;
  }

  /**
   * Dhmiourgia hash value
   *
   * @returns {string}
   */
  calculateHash() {
    return SHA256(this.fromAddress + this.toAddress + this.amount + this.timestamp)
      .toString();
  }

  /**
   * Ypografh sunallaghs me to kleidi apo to elliptic object. H ypografh apothikeuetai mesa sth sunallagh
   kai meta
   * kai meta sto blockchain.
   *
   * @param {string} signingKey
   */
  signTransaction(signingKey) {
    // Gia na petuxei h sunallagh to pair of keys prepei na einai valid.

    if (signingKey.getPublic('hex') !== this.fromAddress) {
      throw new Error('You cannot sign transactions for other wallets!');
    }

    // Ypologismos, ypografh kai apothikeusi hash
    const hashTx = this.calculateHash();
    const sig = signingKey.sign(hashTx, 'base64');

    this.signature = sig.toDER('hex');
  }
}
```

```

* Elegxos egkurothtas ypografhs
*
*
* @returns {boolean}
*/
isValid() {

  if (this.fromAddress === null) return true;

  if (!this.signature || this.signature.length === 0) {
    throw new Error('No signature in this transaction');
  }

  const publicKey = ec.keyFromPublic(this.fromAddress, 'hex');
  return publicKey.verify(this.calculateHash(), this.signature);
}
}

```

Καταγράφουμε τα χαρακτηριστικά που ορίζουν μια δοσοληψία, υπολογίζουμε το hash που αποτελείται από τα χαρακτηριστικά, υπογράφουμε τη συναλλαγή, και ελέγχουμε με διάφορες μεθόδους την εγκυρότητα των στοιχείων που εισάγονται όπως τα ζευγάρια κλειδιών.

Κλάση Block

```

class Block {
  /**
   * @param {number} timestamp
   * @param {Transaction[]} transactions
   * @param {string} previousHash
   */
  constructor(timestamp, transactions, previousHash = "") {
    this.previousHash = previousHash;
    this.timestamp = timestamp;
    this.transactions = transactions;
    this.nonce = 0;
    this.hash = this.calculateHash();
  }

  /**
   * epistrefei to hash enos block
   *
   * @returns {string}
   */
  calculateHash() {

```

```

    return SHA256(this.previousHash + this.timestamp + JSON.stringify(this.transactions) +
this.nonce).toString();
}

/**
 * Ekkinhsh diadikasias mining. Allazei to nonce wste to block na ksekinai me arketa mhdenika
 * oso kai h duskolia.
 *
 * @param {number} difficulty
 */
mineBlock(difficulty) {
  while (this.hash.substring(0, difficulty) !== Array(difficulty + 1).join('0')) {
    this.nonce++;
    this.hash = this.calculateHash();
  }

  console.log(`Block mined: ${this.hash}`);
}

/**
 * Elegxei hash kai upografei mesa sto block kai dinei true/false
 *
 *
 * @returns {boolean}
 */
hasValidTransactions() {
  for (const tx of this.transactions) {
    if (!tx.isValid()) {
      return false;
    }
  }
}

return true;
}
}

```

Αποθηκεύουμε τα χαρακτηριστικά που καθορίζουν ένα block, υπολογίζουμε το hash του, και μέσω της μεθόδου mineblock ξεκινάμε τη διαδικασία εύρεσης τιμής κατακερματισμού που απαιτεί ο αλγόριθμος, ενώ παρέχεται και μέθοδος για την επικύρωση όλων των συναλλαγών.

Κλάση blockchain

```
class Blockchain {
  constructor() {
    this.chain = [this.createGenesisBlock()];
    this.difficulty = 2;
    this.pendingTransactions = [];
    this.miningReward = 100;
  }

  /**
   * @returns {Block}
   */
  createGenesisBlock() {
    return new Block(Date.parse('2022-09-30'), [], '0');
  }

  /**
   *
   * gurnaei to hash tou teleutaiou block
   *
   * @returns {Block[]}
   */
  getLatestBlock() {
    return this.chain[this.chain.length - 1];
  }

  /**
   * Lambanei oles tis sunallages, tis metatrepei se block kai ksekina th diadikasia tou mining.
   *
   * @param {string} miningRewardAddress
   */
  minePendingTransactions(miningRewardAddress) {
    const rewardTx = new Transaction(null, miningRewardAddress, this.miningReward, "");
    this.pendingTransactions.push(rewardTx);

    let block = new Block(Date.now(), this.pendingTransactions, this.getLatestBlock().hash);
    block.mineBlock(this.difficulty);

    console.log('Block successfully mined!');
    this.chain.push(block);

    this.pendingTransactions = [];
  }

  /**
   * Prothesi epomenhs sunallaghs ston pinaka pending sunallagwn kai
   * sigoureuei oti h sunallagh einai upogegrammeni
   *
   * @param {Transaction} transaction
   */
  addTransaction(transaction) {
    if (!transaction.fromAddress || !transaction.toAddress) {
      throw new Error('Transaction must include from and to address');
    }
  }
}
```

```

// Elegxos egkurothtas sunallaghs
if (!transaction.isValid()) {
    throw new Error('Cannot add invalid transaction to chain');
}

if (transaction.amount <= 0) {
    throw new Error('Transaction amount should be higher than 0');
}

this.pendingTransactions.push(transaction);
}

/**
 * epistrefei to balance tou xrhsth
 *
 * @param {string} address
 * @returns {number} The balance of the wallet
 */
getBalanceOfAddress(address) {
    let balance = 0;

    for (const block of this.chain) {
        for (const trans of block.transactions) {
            if (trans.fromAddress === address) {
                balance -= trans.amount;
            }

            if (trans.toAddress === address) {
                balance += trans.amount;
            }
        }
    }

    return balance;
}

/**
 *Epistrefei oles tis sunallages pou exei kanei ena public address.
 *
 * @param {string} address
 * @return {Transaction[]}
 */
getAllTransactionsForWallet(address) {
    const txs = [];

    for (const block of this.chain) {
        for (const tx of block.transactions) {
            if (tx.fromAddress === address || tx.toAddress === address) {
                txs.push(tx);
            }
        }
    }

    return txs;
}

/**

```

```

* elegxei thn egkyrothta twn block, mazi me ta sundeomena hash, enw
* tautoxrona elegxei kai oles tis upogegrammenes sunallages.
*
* @returns {boolean}
*/
isChainValid() {

  // Elegxos Genesis Block
  const realGenesis = JSON.stringify(this.createGenesisBlock());

  if (realGenesis !== JSON.stringify(this.chain[0])) {
    return false;
  }

  // Elegxos upoloipwn block
  for (let i = 1; i < this.chain.length; i++) {
    const currentBlock = this.chain[i];

    if (!currentBlock.hasValidTransactions()) {
      return false;
    }

    if (currentBlock.hash !== currentBlock.calculateHash()) {
      return false;
    }
  }

  return true;
}

module.exports.Blockchain = Blockchain;
module.exports.Block = Block;
module.exports.Transaction = Transaction;

```

Με την κλάση `blockchain` δημιουργούμε το πρώτο `block` της αλυσίδας, ενώ έχουμε μεθόδους για προβολή του τελευταίου `block`, τη δημιουργία νέων `block` μέσω της μεθόδου `minePendingTransactions`, τον έλεγχο ύπαρξης αξιόπιστης υπογραφής στη συναλλαγή, τη διατήρηση και ενημέρωση για τα περιεχόμενα κάποιας δημόσιας διεύθυνσης, μέθοδο για έλεγχο εγκυρότητας όλης της αλυσίδας και όλων των δοσοληψιών.

Κεντρικό αρχείο HTML

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Blockchain </title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>

  <link rel="stylesheet" type="text/css" href="assets/css/bootstrap.min.css">
  <script src="node_modules/pouchdb/dist/pouchdb.min.js"></script>
</body>
</html>
```

Package.json

```
{
  "name": "ng-frontend",
  "version": "0.0.1",
  "scripts": {
    "ng": "ng",
    "start": "ng serve --host 172.18.0.6 --disable-host-check",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e",
    "prod": "ng build --aot --progress false",
    "deploy": "ng deploy"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^14.0.2",
    "@angular/common": "^14.0.2",
    "@angular/compiler": "^14.0.2",
    "@angular/core": "^14.0.2",
    "@angular/forms": "^14.0.2",
    "@angular/platform-browser": "^14.0.2",
    "@angular/platform-browser-dynamic": "^14.0.2",
    "@angular/router": "^14.0.2",
    "angular-2-local-storage": "^3.0.2",
    "angular-cli-ghpages": "^1.0.0",
    "angular2-qrcode": "^2.0.3",
    "buffer-browserify": "^0.2.5",
    "crypto": "npm:crypto-browserify",
    "crypto-browserify": "^3.12.0",
    "crypto-js": "^4.1.1",
  }
}
```

```

"debug": "^4.3.4",
"eciesjs": "^0.3.14",
"elliptic": "^6.5.4",
"pouchdb": "^7.3.0",
"pouchdb-adapter-http": "^7.3.0",
"pouchdb-browser": "^7.3.0",
"qrcode": "^1.5.0",
"reflect-metadata": "^0.1.13",
"buffer": "npm:buffer-browserify",
"stream": "npm:stream-browserify",
"stream-browserify": "^3.0.0",
"tslib": "^2.0.0",
"zone.js": "~0.11.6"
},
"devDependencies": {
"@angular-devkit/build-angular": "^14.0.2",
"@angular/cli": "^14.0.2",
"@angular/compiler-cli": "^14.0.2",
"@angular/language-service": "^14.0.2",
"@types/jasmine": "^4.0.3",
"@types/jasminewd2": "~2.0.3",
"@types/node": "^18.0.0",
"@types/pouchdb-browser": "^6.1.3",
"jasmine-core": "^4.2.0",
"karma": "~6.4.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage-istanbul-reporter": "~3.0.2",
"karma-jasmine": "~4.0.0",
"karma-jasmine-html-reporter": "^1.5.0",
"typescript": "^4.7.4"
}
}

```

Στο παραπάνω αρχείο αναγράφονται όλες οι απαραίτητες βιβλιοθήκες και εργαλεία που είναι απαραίτητα για τη δημιουργία της εφαρμογής. Με τη χρήση της εντολής `npm install` μπορεί κάποιος να λάβει όλα τα απαραίτητα στοιχεία, αρκεί να βρίσκεται στον ίδιο φάκελο με αυτό το αρχείο.

Block View Component HTML code

```

<div class="card" [class.border-primary]="isSelectedBlock()">
  <div class="card-body">
    <h5 class="card-title">Block {{ getBlockNumber() }} <small class="text-muted"
*ngIf="block.previousHash == 0">(Genesis block)</small></h5>
  </div>

  <ul class="list-group list-group-flush">
    <li class="list-group-item">
      <span class="">Hash</span>
      <br>
      <div class="text-truncate" [style.color]="'#' + block.hash.substring(0,6)">
        <small>{{ block.hash }}</small>
      </div>
    </li>
  </ul>
</div>

```

```

<span class="">Hash of previous block</span>
<br>

<div class="text-truncate" [style.color]="'#' + block.previousHash.substring(0,6)">
  <small>{{ block.previousHash }}</small>
</div>
</li>

<li class="list-group-item">
  <span class="">Nonce</span><br>
  <div class="text-truncate text-muted">
    <small>{{ block.nonce }}</small>
  </div>
</li>

<li class="list-group-item">
  <span class="">Timestamp</span><br>
  <div class="text-truncate text-muted">
    <small>{{ block.timestamp }}</small>
  </div>
</li>
</ul>

<!-- <div class="card-body"> -->
  <!-- <span class="card-link" *ngIf="!blockHasTx()">Block has no transactions</span> -->
  <!-- <span class="card-link" *ngIf="blockHasTx()">Contains {{block.transactions.length}}
transactions</span> -->
  <!-- </div> -->
</div>

```

BlockView Component TypeScript code

```

import { Component, OnInit, Input } from '@angular/core';
import { BlockchainService } from '../services/blockchain.service';

@Component({
  selector: 'app-block-view',
  templateUrl: './block-view.component.html',
  styleUrls: ['./block-view.component.scss']
})
export class BlockViewComponent implements OnInit {
  @Input()
  public block;

  @Input()
  public selectedBlock;

  private blocksInChain;

  constructor(private blockchainService: BlockchainService) {
    this.blocksInChain = blockchainService.blockchainInstance.chain;
  }

  ngOnInit() {
  }
}

```

```

blockHasTx() {
  return this.block.transactions.length > 0;
}

isSelectedBlock() {
  return this.block === this.selectedBlock;
}

getBlockNumber() {
  return this.blocksInChain.indexOf(this.block) + 1;
}
}

```

Παρέχονται συνθήκες για το ποια μορφή του blockchain πρέπει να γίνει αναπαράσταση στο browser.

Transaction table HTML code

```

<p *ngIf="transactions.length === 0">This block has no transactions</p>

<table class="table table-hover table-striped" *ngIf="transactions.length > 0">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">From</th>
      <th scope="col">To</th>
      <th scope="col">Amount</th>
      <th scope="col">Payload</th>
      <th scope="col">Timestamp</th>
      <th scope="col">Valid?</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let tx of transactions; index as index">
      <td>{{index}}</td>
      <td class="text-truncate" style="max-width: 100px;">
        <a [routerLink]="['/wallet', tx.fromAddress]" *ngIf="tx.fromAddress !==
null">{{tx.fromAddress}}</a>

        <span *ngIf="tx.fromAddress === null">System</span>

        <span class="text-muted" *ngIf="blockchainService.addressIsFromCurrentUser(tx.fromAddress)">
          <br><small>(That's yours!)</small>
        </span>
      </td>
      <td class="text-truncate" style="max-width: 100px;">
        <a [routerLink]="['/wallet', tx.toAddress]">{{tx.toAddress}}</a>

        <span class="text-muted" *ngIf="blockchainService.addressIsFromCurrentUser(tx.toAddress)">
          <br><small>(That's yours!)</small>
        </span>
      </td>
    </tr>
  </tbody>
</table>

```

```

        {{tx.amount}}
        <span *ngIf="tx.fromAddress === null" class="text-muted"><br><small>(Block
reward)</small></span>
    </td>
    <td>
        
        <span *ngIf="tx.payload === null" class="text-muted"><br><small>(No payload)</small></span>
    </td>
    <td>
        {{tx.timestamp}}<br>
        <span class="text-muted"><small>{{tx.timestamp | date:'MMM d, y, HH:mm'}}</small></span>
    </td>
    <td style="max-width: 40px;">
        <!-- <span *ngIf="tx.isValid()">✓ </span>
        <span *ngIf="!tx.isValid()">✗ </span> -->
    </td>
</tr>
</tbody>
</table>

```

Μέσα από συνθήκες ngIf της angular αλλά και πληροφορίες για την υπάρχουσα μορφή blockchain απεικονίζονται στον browser οι συναλλαγές που αφορούν τον εκάστοτε χρήστη.

Transaction Table Typescript code

```

import { Component, OnInit, Input } from '@angular/core';
import { BlockchainService } from '../services/blockchain.service';
import { encrypt, decrypt, PrivateKey, PublicKey } from 'eciesjs'
import { Buffer } from 'buffer';

@Component({
  selector: 'app-transactions-table',
  templateUrl: './transactions-table.component.html',
  styleUrls: ['./transactions-table.component.scss']
})
export class TransactionsTableComponent implements OnInit {
  @Input()
  public transactions = [];
  public k1: PrivateKey;
  constructor(public blockchainService: BlockchainService) {
    this.k1 = PrivateKey.fromHex(this.blockchainService.walletKeys[0].privateKey);
  }

  decryptImg(img) {
    try {
      if (img) {
        return decrypt(this.k1.secret, Buffer.from(img, 'base64')).toString()
      }
    } catch (e) {
      return "";
    }
  }
}

```

```

ngOnInit() {
}
}

```

Κάθε φορά που κάποιος χρήστης επιλέγει προβολή δοσοληψίας, και το ιδιωτικό του κλειδί ταιριάζει με την δημόσια διεύθυνση που έχει γίνει η δοσοληψία, τότε γίνεται αποκρυπτογράφηση της εικόνας για προβολή.

Blockchain Viewer HTML code

```

<div class="container">
  <h1>Blocks on chain</h1>
  <p>Each square represents a block on the chain. Click on a block to see the transactions inside.</p>
</div>

<div class="container" style="overflow-x: scroll;white-space: nowrap;">
  <app-block-view
    *ngFor="let block of blocks"
    [block]="block"
    [selectedBlock]="selectedBlock"
    (click)="showTransactions(block)"
  ></app-block-view>
</div>

<br><br>

<div class="container">
  <h1>Transactions inside block {{ getBlockNumber(selectedBlock) }}</h1>

  <app-transactions-table
    [transactions]="selectedBlock.transactions"
  ></app-transactions-table>
</div>

```

BlockchainViewer Typescript code

```

import { Component, Input, OnInit } from '@angular/core';

import { BlockchainService } from '../services/blockchain.service';

@Component({
  selector: 'app-blockchain-viewer',
  templateUrl: './blockchain-viewer.component.html',
  styleUrls: ['./blockchain-viewer.component.scss']
})
export class BlockchainViewerComponent implements OnInit {

```

```

public blocks = [];
public selectedBlock = null;

constructor(private blockchainService: BlockchainService) {
  this.blocks = blockchainService.blockchainInstance.chain;
  this.selectedBlock = this.blocks[0];
  console.log(this.blocks);
}

ngOnInit() {
  //edw tha kalesw to endpoint mou
}

showTransactions(block) {
  console.log(block);
  this.selectedBlock = block;
  return false;
}

blockHasTx(block) {
  return block.transactions.length > 0;
}

selectedBlockHasTx() {
  return this.blockHasTx(this.selectedBlock);
}

isSelectedBlock(block) {
  return this.selectedBlock === block;
}

getBlockNumber(block) {
  return this.blocks.indexOf(block) + 1;
}
}

```

Σε αυτό το κομμάτι κώδικα παρέχονται συνθήκες που αφορούν ποιο block θα εμφανιστεί στην οθόνη, οι οποίες παρέχονται από τη συνάρτηση blockchain service μαζί με υποπεριπτώσεις προβολής που εξαρτώνται από τις επιλογές τους χρήστη.

Create Transaction Component Typescript code

```

import { Buffer } from 'buffer';
import { decrypt, encrypt, PrivateKey, PublicKey } from 'eciesjs';

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';

import { Transaction } from '../../blockchain.js';
import {

```

```

BlockchainService,
IWalletKey,
} from '../services/blockchain.service';

@Component({
  selector: 'app-create-transaction',
  templateUrl: './create-transaction.component.html',
  styleUrls: ['./create-transaction.component.scss']
})
export class CreateTransactionComponent implements OnInit {
  public newTx = new Transaction();
  public ownWalletKey: IWalletKey;

  constructor(private blockchainService: BlockchainService, private router: Router) {
    this.newTx = new Transaction();
    this.newTx.toAddress =
'0416a75e8fec9d73cb2e2cd2b62e00b123bc98addc21abb9b36ebf42514e83061ac18f18bb54204c6cb499
d8390fda16f7b75eebf2dfd2ef715cc5616d03c3a04';
    this.newTx.amount = 1;
    this.newTx.payload = false;
    this.ownWalletKey = blockchainService.walletKeys[0];
  }

  ngOnInit() {
    (<any>window).uploadedImage = false;
  }

  uploadFile(event: Event) {
    const element = event.currentTarget as HTMLInputElement;
    this.readFile(element.files[0]);
  }

  readFile(file)
  {
    return new Promise((resolve) =>
    {
      if (file.size)
      {
        const reader = new FileReader();
        reader.onload = (e) =>
        {
          (<any>window).uploadedImage = e.target.result;
          resolve({
            binary: file,
            b64: e.target.result,
          });
        };
        reader.readAsDataURL(file);
      }
    });
  }

  createTransaction() {
    const newTx = this.newTx;
    console.log(newTx);

    // default dhmosia dieuthunsh

```

```

newTx.fromAddress = this.ownWalletKey.publicKey;
newTx.signTransaction(this.ownWalletKey.keyObj);
// const k1 = new PrivateKey(Buffer.from(this.ownWalletKey.privateKey, 'hex'));
const k1 = PrivateKey.fromHex(this.ownWalletKey.privateKey);
const k2 =
PublicKey.fromHex('0416a75e8fec9d73cb2e2cd2b62e00b123bc98addc21abb9b36ebf42514e83061ac18
f18bb54204c6cb499d8390fda16f7b75eebfb2dfd2ef715cc5616d03c3a04');
console.log(k1);
const crypteData = encrypt(k2.uncompressed, Buffer.from((<any>window).uploadedImage));
const stringified = JSON.stringify(crypteData.toString('base64'));
// console.log();
// console.log(encrypt(k1.secret, encrypt(k1.publicKey.toHex(), data)).toString());
if ((<any>window).uploadedImage) {
    newTx.payload = stringified;
}
try {
    this.blockchainService.addTransaction(this.newTx);
} catch (e) {
    alert(e);
    return;
}
(<any>window).uploadedImage = false;
this.router.navigate(['/new/transaction/pending', { addedTx: true }]);
this.newTx = new Transaction();
}
}

```

Στο παρόν αρχείο παρέχονται συναρτήσεις έτσι ώστε όταν ο χρήστης πατήσει το κουμπί create transaction τότε καλείται να επιλέξει το αρχείο το οποίο θέλει να στείλει, το οποίο κρυπτογραφείται με τη δημόσια διεύθυνση που έχει οριστεί από τη συνάρτηση ενώ επίσης καλεί άλλες συναρτήσεις για προσθήκη της δοσοληψίας στον πίνακα εκκρεμών συναλλαγών έτσι ώστε ο χρήστης να προχωρήσει στο τελευταίο στάδιο του mining.

Pending transactions HTML code

```

<div class="container">

  <div class="alert alert-success" role="alert" *ngIf="justAddedTx">
    Transaction created successfully!
  </div>

  <h1>Pending transactions</h1>
  <p>These transactions are waiting to be included in the next block. Next block is created when you
start the mining process.</p>

  <app-transactions-table
    [transactions]="pendingTransactions">
  </app-transactions-table>

  <button class="btn btn-primary" *ngIf="pendingTransactions.length > 0 && miningInProgress ===
false" (click)="minePendingTransactions()">
    Start mining
  </button>

```

```

    <div *ngIf="miningInProgress === true">
      Mining block.. Hang on...
    </div>
  </div>

```

Pending transactions Typescript code

```

import { LocalStorageService } from 'angular-2-local-storage';
import PouchDB from 'pouchdb';

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';

import { BlockchainService } from '../services/blockchain.service';

@Component({
  selector: 'app-pending-transactions',
  templateUrl: './pending-transactions.component.html',
  styleUrls: ['./pending-transactions.component.scss']
})
export class PendingTransactionsComponent implements OnInit {
  public pendingTransactions = [];
  public miningInProgress = false;
  public justAddedTx = false;

  constructor(public blockchainService: BlockchainService, private router: Router, private route:
  ActivatedRoute, public _localStorageService: LocalStorageService) {
    this.pendingTransactions = blockchainService.getPendingTransactions();
  }

  ngOnInit() {
    if (this.route.snapshot.paramMap.get('addedTx')) {
      this.justAddedTx = true;

      setTimeout(() => {
        this.justAddedTx = false;
      }, 4000);
    }
  }

  minePendingTransactions() {

    this.miningInProgress = true;
    this.blockchainService.minePendingTransactions();
    this.miningInProgress = false;
    this._localStorageService.set('chain',
    JSON.stringify(this.blockchainService.blockchainInstance.chain));

    PouchDB.plugin(require('pouchdb-adaptor-http'));
    var db = new PouchDB('blockchain');
    db.get('blockchain').catch(function (err) {
      if (err.name === 'not_found') {
        console.log('I will create an empty chain')

```

```

const chainDoc = {
  "_id": "blockchain",
  "value": JSON.stringify((<any>window).blockchainInstance.chain),
};
return db.post(chainDoc);
} else { // hm, some other error
  throw err;
}
}).then(function (chainDoc) {
  chainDoc.value = JSON.stringify((<any>window).blockchainInstance.chain);
  return db.post(chainDoc);
}).catch(function (err) {
  // handle any errors
});

this.router.navigate(['/']);
}
}

```

Το παραπάνω κομμάτι κώδικα αποθηκεύει την καινούργια κατάσταση του blockchain στην rouchdb, σε περίπτωση προσθήκης νέων block απο τον χρήστη.

Settings HTML code

```

<div class="container">
  <h1>Settings</h1>
  <p>Control how the blockchain behaves when new transactions or blocks are created. Changes are automatically saved.</p>

  <br>

  <div class="form-group">
    <label for="difficulty">Difficulty</label>
    <input type="number" class="form-control" id="difficulty" aria-describedby="difficultyHelp"
    [(ngModel)]="blockchain.difficulty">
    <small id="difficultyHelp" class="form-text text-muted">
      Difficulty controls how long the mining process takes. Higher numbers will make mining a lot slower!
    </small>
    <br>Default: 2
  </div>

  <br>

  <div class="form-group">
    <label for="miningReward">Mining reward</label>
    <input type="number" class="form-control" id="miningReward"
    aria-describedby="miningRewardHelp" [(ngModel)]="blockchain.miningReward">
    <small id="miningRewardHelp" class="form-text text-muted">
      How much "coins" a miner receives for successfully creating a new block for the chain.
    </small>
    <br>Default: 100
  </div>

```

</div>

</div>

Wallet component HTML code

```
<div class="container" *ngIf="walletAddress">
  <h1>Wallet Details</h1>
  <div>
    <qr-code [value]="walletAddress" [size]="350"></qr-code>
  </div>

  <p class="addresshtml" style="word-wrap:break-word;">
    <strong class="actualaddress">Address:</strong>
    <br>
    {{walletAddress}}
  </p>

  <p class="privatekey" style="word-wrap:break-word;">
    <strong>Private key:</strong>
    <br>
    <button *ngIf="!showPrivateKey" type="button" class="btn btn-danger"
(click)="showPrivateKey=true">Show key</button>
    <qr-code *ngIf="showPrivateKey" [value]="walletPrivateKey" [size]="350"
(click)="showPrivateKey=false"></qr-code>
    <input *ngIf="showPrivateKey" class="form-control" type="text" [value]="walletPrivateKey"
readonly>

  </p>

  <p class="balance" style="word-wrap:break-word;">
    <strong>Balance:</strong>
    <br>
    {{ balance }}
  </p>

  <hr>

  <h1>Transactions</h1>
  <p *ngIf="transactions.length === 0">This wallet has made no transactions (yet)</p>
  <app-transactions-table
    [transactions]="transactions"
    *ngIf="transactions.length > 0"
  ></app-transactions-table>
</div>

<div class="container">
  <a [href]="fileUrl" class="btn btn-primary btn-lg" download="wallet.txt">Backup private key to
file</a>
</div>
<br/>
<div class="container">
  Import wallet from file <input type='file' accept='text/plain' multiple (change)='openFile($event)'>
</div>
```

Το παρόν κομμάτι του κώδικα, σε συνδυασμό με τον κώδικα που παρέχεται παρακάτω, αποτελεί τη κεντρική σελίδα της εφαρμογής όπου ο χρήστης μπορεί να δει αφενός το δημόσιο κλειδί του και αφετέρου το ιδιωτικό του κλειδί με το πάτημα του κουμπιού που παρέχεται. Ταυτόχρονα παρέχονται τα κουμπιά για λήψη του παρόντος ζευγαριού κλειδιού που αξιοποιείται και τη μεταφόρτωση ξεχωριστού ζευγαριού, περίπτωση στην οποία ο χρήστης μεταφορτώνει ένα txt αρχείο, και στη συνέχεια το καινούργιο ζευγάρι φαίνεται στην οθόνη του.

wallet component scss code

```
@import
url('https://fonts.googleapis.com/css2?family=Lato&family=Oswald:wght@300&display=swap');

.container {
  color: black;
  text-align: center;
  font-family: 'Lato', sans-serif;
}

.container > addresshtml {
  font-size: 50px;
}

.addresshtml {
  font-size: 17px;
  padding-top: 3rem;
}

.actualaddress {
  font-size: 25px;
  padding-top: 1rem;
}

.privatekey {
  font-size: 25px;
  padding-top: 3rem;
}

.balance {
  font-size: 25px;
  padding-top: 3rem;
  padding-bottom: 3rem;
}

.btn.btn-danger {
  font-size: 30px;
  border-radius: 25%;
}
```

```
.btn.btn-primary.btn-lg {
  background-color: rgb(33, 169, 233);
}
```

```
// .container > qr-code {
//   size: 5px;
// }
```

Wallet Component Typescript code

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { BlockchainService } from '../services/blockchain.service';
import { DomSanitizer } from '@angular/platform-browser';
import { LocalStorageService } from 'angular-2-local-storage';
```

```
interface IWalletKey {
  keyObj: any;
  publicKey: string;
  privateKey: string;
}
```

```
@Component({
  selector: 'app-wallet',
  templateUrl: './wallet.component.html',
  styleUrls: ['./wallet.component.scss']
})
```

```
export class WalletComponent implements OnInit {
```

```
  public walletAddress = "";
  public walletPrivateKey = "";
  public balance = 0;
  public transactions = [];
  private showPrivateKey = false;
  private blob;
  public walletKeys:IWalletKey;
```

```
  constructor(private route: ActivatedRoute, public blockchainService: BlockchainService, private
  _localStorageService: LocalStorageService, private sanitizer: DomSanitizer) {}
```

```
  fileUrl;
```

```
  ngOnInit() {
    this.route.params.subscribe( (params) => {
      const blockchain = this.blockchainService.blockchainInstance;
      this.walletAddress = this.blockchainService.walletKeys[0].publicKey;
      this.walletPrivateKey = this.blockchainService.walletKeys[0].privateKey;

      try {
```

```

    this.balance = blockchain.getBalanceOfAddress(this.walletAddress);
    this.transactions = blockchain.getAllTransactionsForWallet(this.walletAddress);

    //Paragwgh .txt arxeiou
    this.blob = new Blob([this.blockchainService.walletKeys[0].privateKey], { type:
'application/octet-stream' });
    this.fileUrl =
this.sanitizer.bypassSecurityTrustResourceUrl(window.URL.createObjectURL(this.blob));
    } catch(e) {
        console.log(e);
    }
    });
}

openFile(event) {
    let input = event.target;
    for (var index = 0; index < input.files.length; index++) {
        let reader = new FileReader();
        reader.onload = () => {
            try {
                this._localStorageService.set('privateKey', reader.result.toString());
                window.location.reload();
            } catch(e) {
                alert("File is not a valid wallet: " + e);
            }
        }
        reader.readAsText(input.files[index]);
    };
}
}
}

```

Το παρόν κομμάτι κώδικα υλοποιεί τις λειτουργικότητες που περιγράφηκαν παραπάνω, τοποθετώντας στα πεδία της html τις κατάλληλες τιμές, αλλά και είναι υπεύθυνο για τη λειτουργικότητα των κουμπιών λήψης και μεταφόρτωσης.

Wallet Balance HTML code

```

<div class="container">
  <h1>Wallet details</h1>

  <p style="word-wrap:break-word;">
    <strong>Address:</strong>
    <br>
    {{walletAddress}}
  </p>

  <p>
    <strong>Balance:</strong>
    <br>

```

```

    {{ balance }}
  </p>

  <hr>

  <h1>Transactions</h1>
  <p *ngIf="transactions.length === 0">This wallet has made no transactions (yet)</p>
  <app-transactions-table
    [transactions]="transactions"
    *ngIf="transactions.length > 0"
  ></app-transactions-table>
</div>

```

Wallet Balance Component Typescript code

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { BlockchainService } from '../services/blockchain.service';

@Component({
  selector: 'app-wallet-balance',
  templateUrl: './wallet-balance.component.html',
  styleUrls: ['./wallet-balance.component.scss']
})
export class WalletBalanceComponent implements OnInit {
  public walletAddress = "";
  public balance = 0;
  public transactions = [];

  constructor(private route: ActivatedRoute, private blockchainService: BlockchainService) {}

  ngOnInit() {
    this.route.params.subscribe( (params) => {
      this.walletAddress = params['address'];

      const blockchain = this.blockchainService.blockchainInstance;
      this.balance = blockchain.getBalanceOfAddress(this.walletAddress);
      this.transactions = blockchain.getAllTransactionsForWallet(this.walletAddress);
    });
  }
}

```

Με τη χρήση του παραπάνω κώδικα υπολογίζεται το υπόλοιπο σε κρυπτονομίσματα κάθε χρήστη, υπολογίζοντας το άθροισμα όλων των συναλλαγών που έχει πραγματοποιήσει ο χρήστης.

App routing module

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { BlockchainViewerComponent } from
'./pages/blockchain-viewer/blockchain-viewer.component';
import { SettingsComponent } from './pages/settings/settings.component';
import { CreateTransactionComponent } from './pages/create-transaction/create-transaction.component';
import { PendingTransactionsComponent } from
'./pages/pending-transactions/pending-transactions.component';
import { WalletBalanceComponent } from './pages/wallet-balance/wallet-balance.component';
import { WalletComponent } from './pages/wallet/wallet.component';

const routes: Routes = [
  {path: '', component: WalletComponent },
  {path: 'settings', component: SettingsComponent},
  {path: 'new/transaction', component: CreateTransactionComponent },
  {path: 'new/transaction/pending', component: PendingTransactionsComponent },
  {path: 'wallet/:address', component: WalletBalanceComponent },
  {path: 'explorer', component: BlockchainViewerComponent},
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Το παραπάνω αρχείο ευθύνεται για τη σωστή δρομολόγηση στις καρτέλες της εφαρμογής, ανάλογα με τις επιλογές του χρήστη κατά την πλοήγηση και επιλογή των αντίστοιχων κουμπιών.

App Html code

```
<!-- <nav class="navbar navbar-light bg-light"> -->
<nav class="navbar">

  <a class="navbar-brand" routerLink="/" >
    Health-Coin
  </a>

  <div>
    <button routerLink="/new/transaction/pending" class="btn btn-outline-dark"
    *ngIf="thereArePendingTransactions()">
      Pending transactions
      <span class="badge badge-dark">{{blockchain.pendingTransactions.length}} </span>
    </button>
    &nbsp;
    <button routerLink="/" class="btn btn-outline-dark">
      My Wallet
    </button>
    &nbsp;
    <button routerLink="/explorer" class="btn btn-outline-dark">
```

```

    Block Explorer
  </button>
  &nbsp;
  <!-- <button routerLink="/settings" class="btn btn-outline-dark">
    Settings
  </button> -->
  &nbsp;
  <button routerLink="/new/transaction" class="btn btn-outline-dark">
    Create transaction
  </button>
</div>
</nav>

<br>

<router-outlet></router-outlet>

<footer class="text-muted text-center" style="margin-top: 150px;">
  <small>Thesis of EMMANOUIL ANEZINIS<br>
  </small>
</footer>

```

Το αρχικό αρχείο html που περιέχει την μπάρα πάνω στην οποία βρίσκονται τα κουμπιά πλοήγησης της εφαρμογής, μαζί με ρυθμίσεις δρομολόγησης ανά επιλογή.

App component scss code

```

@import url('https://fonts.googleapis.com/css2?family=Arvo:ital@1&display=swap');

.navbar {
  background-color: #36395A;
}

.navbar-brand
{
  position: relative;
  background: url(../assets/img/SVG_LOGO.png);
  background-position-x: -0.3rem;
  text-align: center;
  padding-left: 2rem;
  padding-right: 2rem;
  width: 250px;
  background-size: contain;
  background-repeat: no-repeat;
  text-decoration-color: #FCFCFD;
  font-family: 'Arvo', serif;

}

/* CSS */

```

```

.btn {
  align-items: center;
  appearance: none;
  background-color: #FCFCFD;
  border-radius: 4px;
  border-width: 0;
  box-shadow: rgba(45, 35, 66, 0.4) 0 2px 4px, rgba(45, 35, 66, 0.3) 0 7px 13px -3px, #D6D6E7 0 -3px 0
  inset;
  box-sizing: border-box;
  color: #223044;
  cursor: pointer;
  display: inline-flex;
  font-family: "JetBrains Mono", monospace;
  height: 40px;
  justify-content: center;
  line-height: 1;
  list-style: none;
  overflow: hidden;
  padding-left: 16px;
  padding-right: 16px;
  position: relative;
  text-align: left;
  text-decoration: none;
  transition: box-shadow .15s, transform .15s;
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
  white-space: nowrap;
  will-change: box-shadow, transform;
  font-size: 16.5px;
}

.btn:focus {
  box-shadow: #D6D6E7 0 0 0 1.5px inset, rgba(45, 35, 66, 0.4) 0 2px 4px, rgba(45, 35, 66, 0.3) 0 7px
  13px -3px, #D6D6E7 0 -3px 0 inset;
}

.btn:hover {
  box-shadow: rgba(45, 35, 66, 0.4) 0 4px 8px, rgba(45, 35, 66, 0.3) 0 7px 13px -3px, #D6D6E7 0 -3px 0
  inset;
  transform: translateY(-2px);
}

.btn:active {
  box-shadow: #D6D6E7 0 3px 7px inset;
  transform: translateY(2px);
}

```

App component typescript code

```
import PouchDB from 'pouchdb';

import { Component, OnInit } from '@angular/core';

import { environment } from '../environments/environment';
import { BlockchainService } from '../services/blockchain.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  public blockchain;
  environment;
  constructor(private blockchainService: BlockchainService) {
    this.blockchain = blockchainService.blockchainInstance;

    var db = new PouchDB('blockchain');
    db.get('blockchain').then(function (doc) {
      (<any>window).blockchainInstance.chain = JSON.parse(doc.value);
    });
  }

  ngOnInit() {
  }

  thereArePendingTransactions() {
    return this.blockchain.pendingTransactions.length > 0;
  }
}
```

Τέλος, στον παραπάνω κώδικα παρέχονται ρυθμίσεις για την αποθήκευση της κατάστασης του blockchain στην pouchdb.