ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και Ανάπτυξη ασφαλούς συστήματος επικοινωνίας με χρήση της τεχνολογίας Blockchain.

ΠΑΠΑΔΟΠΟΥΛΟΥ ΘΕΩΝΗ ΑΝΝΑ
Α.Μ. 19390301

Εισηγητής: Ι. ΒΟΓΙΑΤΖΗΣ, Καθηγητής

Αθήνα, Μάρτιος 2023

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και Ανάπτυξη ασφαλούς συστήματος επικοινωνίας με χρήση της τεχνολογίας Blockchain.

ΠΑΠΑΔΟΠΟΥΛΟΥ ΘΕΩΝΗ ΑΝΝΑ
Α.Μ. 19390301

Αθήνα, Μάρτιος 2023

Εισηγητής:

Ιωάννης Βογιατζής, Καθηγητής

Εξεταστική Επιτροπή:

Ιωάννης Βογιατζής, Καθηγητής

Ioannis Voyiatzis Digitally signed by Ioannis Voyiatzis
Date: 2023.03.21 17:48:46 +02'00'

Χρήστος Τρούσσας, Επίκουρος Καθηγητής

Christos Troussas Digitally signed by Christos Troussas
Date: 2023.03.20 17:25:18 +02'00'

Γεώργιος Μελετίου ΕΔΙΠ

Georgios Meletiou Digitally signed by Georgios Meletiou
Date: 2023.03.20 13:30:02 +02'00'

Ημερομηνία εξέτασης : 17/3/2023

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματος μου».

Η Δηλούσα

Παπαδοπούλου Θεώνη Άννα

ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας τη διπλωματική μου εργασία θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Βογιατζή για την εμπιστοσύνη που μου έδειξε και τη δυνατότητα να εκπονήσω αυτή τη διπλωματική εργασία αλλά και για τα υπέροχα μαθήματα που παρακολούθησα από αυτόν κατά τη διάρκεια των σπουδών μου.

Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τον κ. Μελετίου για το χρόνο που αφιέρωσε και την καθοδήγηση που παρείχε.

Ευχαριστώ το Ίδρυμα συνολικά και ιδιαίτερα το τμήμα Μηχανικών και Πληροφορικής, που διαχειρίζεται εξαιρετικά καταστάσεις που προκύπτουν, όπως αυτή της πανδημίας, με εξαιρετική ανταπόκριση.

Ευχαριστώ και τους καθηγητές που είχα τη τιμή να γνωρίσω, αλλά δεν αναφέρω ονομαστικά, καθ' όλη τη διάρκεια των σπουδών μου.

Θα ήθελα να ευχαριστήσω και έναν αφανή συμπαραστάτη τον γιατρό του νοσοκομείου Άγιος Ανδρέας, Παναγιώτη Καρανάσιο που μου έδωσε τη δυνατότητα να μπορώ να έχω μια φυσιολογική καθημερινότητα χωρίς πολλές εκπλήξεις.

Τέλος θα ήθελα να ευχαριστήσω τον εαυτό μου που παρά τα εμπόδια, συνέχισα να προσπαθώ σε όλη τη διάρκεια των σπουδών μου αλλά και για τη συγκεκριμένη διπλωματική, όχι μόνο να ολοκληρώσω την έρευνα αλλά να βρω κίνητρα και περισσότερες διαστάσεις που θα αποτελούν εργαλεία και εφόδια για το μέλλον.

Περίληψη

Η ανταλλαγή μηνυμάτων και ο έλεγχος αυθεντικότητας τους είναι μια ιδιότητα ζωτικής σημασίας στην επικοινωνία. Σήμερα, τα ηλεκτρονικά μηνύματα είναι η πιο χρησιμοποιούμενη εφαρμογή δικτύου. Παρά το ότι υπάρχει πληθώρα ψηφιακών τεχνικών στις επικοινωνίες, ταυτόχρονα συνδυάζονται με πολλά τρωτά σημεία. Δημόσιες και ιδιωτικές δομές είναι αναγκαίο να επικοινωνούν με ασφάλεια μεταξύ τους χρησιμοποιώντας ψηφιακά μέσα. Τα γεγονότα της εποχής κατέδειξαν ελλείψεις αρκετών μηχανισμών που μέχρι πρόσφατα στηρίζονταν στην φυσική παρουσία. Η αυξανόμενη ζήτηση για ασφαλή και απαραβίαστα δεδομένα έχει οδηγήσει στην ανάπτυξη νέων τεχνολογιών, όπως το blockchain, που καθιστά δυνατή την παροχή ασφαλούς επικοινωνίας σε συνδυασμό με αμετάβλητο τρόπο καταγραφής δεδομένων μηνυμάτων. Για αυτό το σκοπό, παρουσιάζεται και προτείνεται η χρήση ενός πλαισίου ανταλλαγής μηνυμάτων που βασίζεται σε blockchain για τη δημιουργία μιας αποκεντρωμένης εφαρμογής. Η χρήση έξυπνων συμβολαίων που θα παρουσιαστεί, εισάγεται ως το μέσο διασφάλισης ότι όλα τα μέρη που εμπλέκονται σε μια επικοινωνία συμμορφώνονται με προκαθορισμένους κανόνες, τα μηνύματα που ανταλλάσσονται δεν μπορούν να αλλοιωθούν ή να διαγραφούν και τυχόν διαφορές που προκύπτουν κατά τη διαδικασία είναι δυνατό να επιβλέπονται από καθορισμένες αρχές. Στόχος είναι να δημιουργηθεί ένα ασφαλές σύστημα επικοινωνίας που δεν μπορεί να αλλάξει, καθιστώντας το ιδανική λύση για ασφαλή, αξιόπιστη επικοινωνία σε έναν όλο και πιο ψηφιοποιημένο κόσμο.

Λέξεις κλειδιά: χρονοσήμανση δεδομένων, έξυπνα σύμβαση, απαραβίαστα δεδομένα, αυθεντικότητα μηνυμάτων, αποκεντρωμένες εφαρμογές μηνυμάτων

Abstract

The exchange of messages and their authentication is a vital property in communication. Nowadays, electronic messaging is the most used network application. Although there is an abundance of digital techniques in communications, at the same time they are combined with many vulnerabilities. Public and private structures need to communicate securely with each other using digital tools. The events of the time demonstrated the shortcomings of several mechanisms that until recently relied on physical presence. The increasing demand for secure and tamper-proof communication systems has led to the development of new technologies, such as blockchain, that can be used to create secure communication systems combined with an immutable way of recording message data. To this end, the use of a blockchain-based messaging framework to build a decentralized application is presented and proposed. The use of smart contracts to be presented is introduced as a means of ensuring that all parties involved in a communication comply with predefined rules, messages exchanged cannot be tampered with or deleted, and any differences that arise during the process can be monitored by specified authorities. The goal is to create a secure communication system that cannot be changed, making it the ideal solution for secure, reliable communication in an increasingly digitized world.

Keywords: Digital Timestamp, blockchain, blockchain messages, smart contracts, tamper-proof messages

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Συντομογραφίες

ABI	Application Binary Interface
BFT	Byzantine Fault Tolerance
BIS	Bank for International Settlements
CA	Certificate Authority
DAG	Directed acyclic graph
DApp	Decentralized Application
DDoS	Distributed Denial-of-Service
DLT	Distributed Ledger Technology
EIPs	Ethereum Improvement Proposals
IoT	Internet of Things
IPFS	Inter Planetary File System
NGI	Next Generation Internet
NIST	National Institute of Standards and Technology
PKI	Public Key Infrastructure
SHA	Secure Hash Algorithm
SPOF	Single Point Of Failure
TPS	Transactions Per Second
EE	Ευρωπαϊκή Ένωση

ΚΕΦΑΛΑΙΟ 1

1. Εισαγωγή

Το πρώτο ολοκληρωμένο και λειτουργικό Blockchain εμφανίστηκε για πρώτη φορά στο έγγραφο της δημοσίευσης του Bitcoin από κάποιον με το ψευδώνυμο Nakamoto που περιγράφει ένα νέο αποκεντρωμένο κρυπτονόμισμα(Nakamoto, 2008). Το Blockchain Bitcoin προσελκύει την προσοχή και στη συνέχεια, αναπτύσσονται πολλά κρυπτονομίσματα και έργα που βασίζονται στο Blockchain ξεπερνώντας τον αρχικό σκοπό αυτής της πρότασης. Η πρόταση του Satoshi Nakamoto είναι ένα σύστημα ηλεκτρονικών συναλλαγών που δεν βασίζεται στην εμπιστοσύνη(Nakamoto, 2008) της εύρυθμης και ασφαλούς λειτουργίας του αλλά στην απόδειξη εργασίας εισάγοντας την έννοια συναίνεσης του δικτύου για τη καταχώρηση δεδομένων. Ξεκίνησε από ήδη γνωστά πλαίσια ψηφιακών νομισμάτων από υπάρχουσες μελέτες ερευνητών και προγραμματιστών που κατασκευάζονταν από ψηφιακές υπογραφές, διατηρούσαν την ιδέα του ελέγχου της ιδιοκτησίας, αλλά με ελλείψεις και χωρίς τρόπο αποτροπής ενός σημαντικού προβλήματος, αυτό της διπλής δαπάνης(Nakamoto, 2008).

Το Blockchain είναι μια τεχνολογία κατανεμημένης λογιστικής(Swan, 2015) που επιτρέπει ασφαλείς και διαφανείς συναλλαγές μεταξύ πολλών μερών χωρίς την ανάγκη μιας κεντρικής αρχής ή κεντρικού κόμβου. Η εισαγωγή της τεχνολογίας Blockchain στο τεχνολογικό χώρο συνδύασε την κρυπτογραφία, την τεχνολογία κατανεμημένων συστημάτων, την τεχνολογία ομότιμης δικτύωσης και άλλες ήδη γνωστές τεχνολογίες. Οι πρακτικές εφαρμογές που υλοποιηθηκαν στη συνέχεια, οι βιβλιογραφικές και δημοσιευμένες έρευνες, κατέδειξαν την τεχνολογία Blockchain άξια προσοχής για διερευνηση και αξιοποίηση της σε ποικίλους τομείς. Έχει μελετηθεί ευρέως για την αξιοποίηση της σε τομείς όπως τα χρηματοοικονομικά, η υγειονομική περίθαλψη, η

εφοδιαστική αλυσίδα και το Διαδίκτυο των Πραγμάτων (Swan, 2015· Crosby et al., 2016· Li et al., 2017· Zohar, 2015). Στον χρηματοπιστωτικό κλάδο, το blockchain έχει εφαρμοστεί για τη δημιουργία ασφαλών και διαφανών συστημάτων πληρωμών (Antonopoulos, 2014). Στην υγειονομική περίθαλψη, μπορεί να βοηθήσει στη διαχείριση των δεδομένων ασθενών με ασφάλεια και αποτελεσματικότητα (Mettler, 2016). Ο κλάδος της εφοδιαστικής αλυσίδας μπορεί να επωφεληθεί από την ικανότητα του Blockchain να παρακολουθεί και να ανιχνεύει προϊόντα από το σημείο προέλευσης έως το σημείο κατανάλωσης (Iansiti και Lakhani, 2017). Το IoT μπορεί να ενισχυθεί με την ικανότητα του Blockchain να δημιουργεί ασφαλή και αποκεντρωμένα δίκτυα Christidis and Devetsikiotis, 2016). Συνολικά, οι πιθανές εφαρμογές της τεχνολογίας blockchain είναι πολυπληθείς και μπορούν να φέρουν επανάσταση σε πολλούς κλάδους.

Μια ιδιαίτερα θετική εξέλιξη για την υιοθέτηση της τεχνολογίας Blockchain, ή αναφερόμενη ως τεχνολογίας κατανεμημένης λογιστικής DLT, είναι η αποδοχή της από θεσμικά όργανα όπως είναι αυτό της Ευρωπαϊκής Ένωσης. Το Δεκέμβριο του 2021 με δελτίο τύπου(European Council, 2021) από το Ευρωπαϊκό Συμβούλιο της Ευρωπαϊκής Ένωσης ανακοινώθηκε ότι τα κράτη μέλη έχουν εγκρίνει μια συμφωνία που επιτεύχθηκε στο Ευρωπαϊκό Κοινοβούλιο σχετικά με τη χρήση της DLT στην ΕΕ. Η συμφωνία αποσκοπεί στην παροχή νομικής ασφάλειας και ρυθμιστικής σαφήνειας για τη χρήση της DLT σε διάφορους τομείς, όπως τα χρηματοοικονομικά, η υγεία και η διαχείριση της εφοδιαστικής αλυσίδας. Στοχεύει επίσης στην προώθηση της καινοτομίας και της διαλειτουργικότητας στον χώρο DLT, διασφαλίζοντας παράλληλα τον σεβασμό των αξιών και των αρχών της ΕΕ. Στο δελτίο τύπου αναφέρεται επίσης ότι η συμφωνία επιτεύχθηκε μετά από αρκετούς γύρους διαπραγματεύσεων και ότι το επόμενο βήμα είναι η αποστολή της στο Ευρωπαϊκό Κοινοβούλιο για τελική έγκριση. Σε συνέχεια αυτού του δελτίου τύπου, στις 21 Δεκεμβρίου 2022 εγκρίθηκε επίσημα (European Parliament, 2022) και θα τεθεί σε ισχύ 18 μήνες μετά την αναφερόμενη έγκριση, η οποία

αναμένεται να είναι στα μέσα του 2024. Όπως φαίνεται λοιπόν η τεχνολογία Blockchain δεν έχει τραβήξει μόνο την προσοχή, αλλά κινεί και σημαντικούς φορείς σε διαβουλεύσεις και αποφάσεις ώστε να ανοίξει ο δρόμος για την αξιοποίηση της. Αξίζει να σημειωθεί επίσης πως το προαναφερόμενο είναι μόνο ένα από τα παραδείγματα που μελετήθηκαν για τη παρούσα έρευνα όσον αφορά την καθολική αποδοχή της τεχνολογίας.

Τι σημαντικό έχει όμως η τεχνολογία blockchain που ακόμα και οντότητες όπως αυτός της ΕΕ συζητούν την εφαρμογή της σε διαφορετικούς τομείς; Η επόμενη επαναστατική υλοποίηση της τεχνολογίας Blockchain παρουσιάστηκε το 2014 από τον Vitalik Buterin με το όνομα Ethereum(Buterin, 2014). Η τεχνολογία blockchain στην οποία βασίζεται το Ethereum διατηρεί τα θετικά που παρέχει το Bitcoin, όπως απαραβίαστα δεδομένα που δεν μπορούν να ανακληθούν ή να αποποιηθούν από τη ταυτότητα τους , πρόσθεσε όμως σε αυτά την υποστήριξή για έξυπνα σύμβαση. Τα “έξυπνα συμβόλαια” είναι συμβάσεις αυτοεκτελούμενες με τους όρους της συμφωνίας να γράφονται απευθείας σε γραμμές κώδικα. Είναι προγράμματα υπολογιστών που εκτελούν αυτόματα τους όρους μιας σύμβασης όταν πληρούνται προκαθορισμένες προϋποθέσεις που έχουν οριστεί στον κώδικα. Τα έξυπνα συμβόλαια αποθηκεύονται σε μια αλυσίδα μπλοκ, καθιστώντας τα διαφανή, ασφαλή, απαραβίαστα και μη τροποποιήσιμα. Η δυνατότητα να μπορούν να δημιουργηθούν εφαρμογές που ορίζονται πάνω στο Blockchain δίνει τη δυνατότητα για την υλοποίηση πληθώρας συστημάτων αξιοπιστίας και απόδειξης, που όπως αναφέρει ο Buterin(Buterin, 2014, p 22), μπορεί να είναι πιο εξελιγμένα από αυτά που είναι διαθέσιμα σήμερα.

Ένα από τα κύρια προβλήματα με τα τρέχοντα κύρια συστήματα είναι το ζήτημα της εξουσιοδότησης. Οι δημόσιες αρχές χρειάζεται συχνά να έχουν πρόσβαση σε μηνύματα που ανταλλάσσονται μεταξύ ατόμων ή οργανισμών για την εκτέλεση των καθηκόντων τους, όπως στην περίπτωση νομικών ερευνών. Ωστόσο, αυτή η διαδικασία εξουσιοδότησης μπορεί να είναι

περίπλοκη και μπορεί να οδηγήσει σε διαφωνίες μεταξύ αρχών και ατόμων ή οργανισμών. Ένα άλλο πρόβλημα είναι η έλλειψη μηχανισμών χρονοσήμανσης και μη απόρριψης στα τρέχοντα συστήματα ανταλλαγής μηνυμάτων, γεγονός που καθιστά δύσκολη την επαλήθευση της γνησιότητας και της ακεραιότητας των μηνυμάτων που ανταλλάσσονται. Αυτό μπορεί να οδηγήσει σε ζητήματα παραποίησης δεδομένων, απώλειας δεδομένων και δυσπιστίας μεταξύ των μερών. Το blockchain μπορεί να χρησιμοποιηθεί για τη δημιουργία ασφαλών συστημάτων επικοινωνίας σε συνδυασμό με έναν αμετάβλητο τρόπο καταγραφής της ταυτότητας των μηνυμάτων. Για το σκοπό αυτό, παρουσιάζεται και προτείνεται ένα ασφαλές σύστημα επικοινωνίας, βασισμένο σε έξυπνες συμβάσεις, που δεν μπορεί να αλλάξει, καθιστώντας το την ιδανική λύση για αξιόπιστη, επαληθεύσιμη και ταυτόχρονα ασφαλή επικοινωνία μεταξύ δύο μερών.

1.1 Υπόβαθρο και κίνητρα

Η τεχνολογία Blockchain, με τη δημιουργία του Bitcoin, έχει αλλάξει εντελώς τον τρόπο που μπορεί να αποδοθεί η ταυτοποίηση, η αξιοπιστία και η μη απόρριψη των συναλλαγών. Στον πυρήνα του, το blockchain είναι μια κατανεμημένη βάση δεδομένων που επιτρέπει στους συμμετέχοντες να επαληθεύουν τις κρυπτογραφημένες συναλλαγές με ασφάλεια. Ένα από τα πιο σημαντικά πλεονεκτήματα της τεχνολογίας blockchain που μπορεί να αξιοποιηθεί είναι η ικανότητά της να παρέχει ασφαλή και επαληθεύσιμη επικοινωνία μεταξύ των συμμετεχόντων. Τα παραδοσιακά συστήματα ανταλλαγής μηνυμάτων και ψηφιακών υπογραφών υποφέρουν από περιορισμούς που μπορούν να ξεπεράσουν τα συστήματα που βασίζονται σε blockchain, όπως ο κίνδυνος αποτυχίας μοναδικού σημείου(Lynch, 2009) ή η έλλειψη διαφάνειας και εμπιστοσύνης στην ταυτότητα των συμμετεχόντων.

Ο πρωταρχικός στόχος αυτής της διπλωματικής είναι να εισάγει ένα ασφαλές σύστημα ανταλλαγής μηνυμάτων βασισμένο στην τεχνολογία blockchain που επιτρέπει την επαληθεύσιμη ανταλλαγή μηνυμάτων. Το προτεινόμενο σύστημα περιλαμβάνει την υλοποίηση ενός έξυπνου συμβολαίου και μιας διεπαφής ιστού για συνομιλία, με έμφαση στην παροχή αξιόπιστης ταυτότητας και αδιαβλητων δεδομένων μηνυμάτων, όπως ο αποστολέας, ο παραλήπτης, η χρονική σήμανση και το περιεχόμενο. Το αυξανόμενο ενδιαφέρον για την τεχνολογία blockchain και τις πιθανές εφαρμογές της καθιστούν αυτή τη διατριβή μια σημαντική συμβολή στον τομέα της επικοινωνίας σε περιπτώσεις που η χρονοσήμανση των δεδομένων και η μη αποποίηση παίζουν σημαντικό ρόλο. Το προτεινόμενο πλαίσιο και το μοντέλο έξυπνων συμβολαίων θα χρησιμεύσουν ως βάση για την ανάπτυξη ασφαλών και αποκεντρωμένων συστημάτων που μπορούν να αντιμετωπίσουν ζητήματα του πραγματικού κόσμου, προωθώντας τη διαφάνεια και την εμπιστοσύνη στην ψηφιακή επικοινωνία.

1.2 Δήλωση προβλήματος

Η τεχνολογία Blockchain παρέχει μια εναλλακτική προσέγγιση στην ψηφιακή επικοινωνία, την πιστοποίηση και την ανταλλαγή μηνυμάτων, επιτρέποντας την πραγματοποίηση αυτών με μειωμένο κόστος και χωρίς καθυστερήσεις Η κατανεμημένη φύση του Blockchain διασφαλίζει ότι καμία μεμονωμένη οντότητα δεν μπορεί να ελέγξει ή να χειριστεί το σύστημα ανταλλαγής μηνυμάτων και οι κρυπτογραφικοί μηχανισμοί του παρέχουν ισχυρά χαρακτηριστικά ασφάλειας και ελέγχου ταυτότητας που αποτρέπουν τη μη εξουσιοδοτημένη πρόσβαση και την παραβίαση. Τα έξυπνα συμβόλαια μπορούν να χρησιμοποιηθούν για να ορίσουν τους κανόνες και την εξουσιοδότηση στη πρόσβαση των μηνυμάτων με εύκολο τρόπο.

Ωστόσο, ενώ υπάρχουν αρκετά πλαίσια ανταλλαγής μηνυμάτων που βασίζονται σε blockchain, έχουν περιορισμούς και δεν υλοποιούν το

παρουσιαζόμενο ζητούμενο. Ως εκ τούτου, στην ανάγκη για ένα αποτελεσματικό σύστημα ανταλλαγής μηνυμάτων που διατηρεί, την ασφάλεια αλλά προσφέρει επιπλέον την αδιάσειστη χρονοσημανση, και την μη αποποίησης της αποστολής τους, η αξιοποίηση της τεχνολογίας blockchain είναι μια ιδανική και χαμηλού κόστους λύση. Πιο συγκεκριμένα, με τη χρήση ψηφιακών συμβολαίων είναι εφικτό να δοθεί εξουσιοδότηση για συγκεκριμένες λειτουργίες όπως η προβολή μιας συνομιλίας μεταξύ δύο μερών χωρίς να είναι προσβάσιμα σε όλους και χωρίς να χάνεται η ασφάλεια του συστήματος. Αυτό είναι το πρόβλημα που στοχεύει να αντιμετωπίσει η παρούσα μελέτη.

1.3 Στόχοι και πεδίο έρευνας

Ο πρωταρχικός στόχος αυτής της έρευνας είναι η ανάπτυξη ενός ασφαλούς συστήματος ανταλλαγής μηνυμάτων για την διασφάλιση της μη αποποίησης και της χρονοσήμανσης των μηνυμάτων με τη τεχνολογία Blockchain και με δυνατότητα άμεσης επιβεβαίωσης μηνυμάτων σε εξουσιοδοτημένες αρχές με απλό τρόπο. Το προτεινόμενο σύστημα στοχεύει να παρέχει ένα σύστημα ανταλλαγής μηνυμάτων των συμμετεχόντων, όπου οι συμμετέχοντες μπορούν να αποδείξουν ότι μια συνομιλία πραγματοποιήθηκε σε συγκεκριμένο χρόνο, τα μηνύματα που ανταλλάχθηκαν δεν είναι δυνατόν να τροποποιηθούν άρα η επικοινωνία είναι με βεβαιότητα αυτή που έχει καταχωρηθεί. Οι αντίστοιχες αρχές μέσω του έξυπνου συμβολαίου μπορούν να επιβεβαιώσουν την πραγματοποίηση της μόνο από τη δυνατότητα να την παρακολουθήσουν.

Για την επίτευξη αυτού του στόχου, έχουν προσδιοριστεί οι ακόλουθοι ερευνητικοί στόχοι:

1. Διερεύνησης του ιστορικού της εξέλιξης της τεχνολογίας Blockchain βιβλιογραφικά
2. Διερεύνησης των βασικών δομών της τεχνολογίας blockchain και επιλογή προτεινόμενης τεχνολογίας για την υλοποίηση της εφαρμογής.

3. Διερεύνηση των περιορισμών και των κινδύνων των παραδοσιακών συστημάτων ανταλλαγής μηνυμάτων για την παροχή ασφαλούς και αδιάψευστης επικοινωνίας μεταξύ των μερών.
4. Αξιολόγηση υπαρχόντων εφαρμογών ανταλλαγής μηνυμάτων βασισμένα σε blockchain.
5. Ανάπτυξη ενός πλαισίου ανταλλαγής μηνυμάτων που βασίζεται σε έξυπνα συμβόλαια και περιγραφή των βασικών τους σημείων.
6. Σχεδιασμός και εφαρμογή ενός έξυπνου συμβολαίου ανταλλαγής μηνυμάτων που θα μπορούν να ελεγχθούν από κατάλληλες αρχές με τη γλώσσα Solidity
7. Επεκτασιμότητα και συμβατότητα για εφαρμογή της Έξυπνης σύμβασης με δίκτυα που υποστηρίζουν την εικονική μηχανή Ethereum.
8. Χρήση μιας ενδεικτικής εφαρμογή Ιστού με node.js και React που αλληλεπιδρά με πρωτόκολλο blockchain συμβατό με Ethereum.
9. Η εφαρμογή επιτρέπει στους χρήστες να στέλνουν και να λαμβάνουν μηνύματα χρησιμοποιώντας το υλοποιημένο συμβόλαιο ανταλλαγής μηνυμάτων σε δοκιμαστικό δίκτυο. Οι αρχές μπορούν να έχουν πρόσβαση στο περιεχόμενο της συνομιλίας από το περιβάλλον Remix στο δίκτυο Fuji-C-Chain που τρέχει η εφαρμογή.

Συνολικά, αυτή η έρευνα στοχεύει να συμβάλει στον τομέα της πιστοποίησης ανταλλαγής μηνυμάτων αναπτύσσοντας ένα πυρήνα μιας εφαρμογής που βασίζεται σε blockchain. Με αυτόν τον τρόπο, αυτή η έρευνα θα παρέχει πληροφορίες για τον τρόπο, τα πιθανά οφέλη και τις προκλήσεις της χρήσης τεχνολογίας blockchain σε εφαρμογές πιστοποίησης μηνυμάτων. Στην παρούσα πρόταση θα αναφερθούν οι περιορισμοί που παρουσιάζονται σε αντίστοιχα συστήματα που βασίζονται σε παραδοσιακές τεχνολογίες ανταλλαγής μηνυμάτων ωστε να αναδειχθεί η ανάγκη για την ύπαρξη μιας τέτοιας υλοποίησης.

ΚΕΦΑΛΑΙΟ 2

2. Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο θα σκιαγραφηθεί η βασική δομή της τεχνολογίας Blockchain και τα συστατικά της μέρη. Το βιβλιογραφικό μέρος της μελέτης σχετικά με το ιστορικό και την εξέλιξη της τεχνολογίας αποτελεί ένα σημαντικό μέρος στην κατανόηση της αλλά και στην κατανόηση ότι η ύπαρξη διαφορετικών υλοποιήσεων απευθύνονται σε κάποιον σκοπό ή πραγματοποιήθηκαν για να συνεισφέρουν στη λύση κάποιου προβλήματος. Ένα μεγάλο μέρος της παρούσας μελέτης αφιερώθηκε στην αναζήτηση της κατάλληλης τεχνολογίας για την υλοποίηση ενός ασφαλούς συστήματος επικοινωνίας που θα καλύπτει τις απαιτήσεις και το σκοπό ενός τέτοιου συστήματος. Τα δομικά στοιχεία της τεχνολογίας Blockchain που μελετήθηκαν έχουν αρκετά κοινά στη βάση τους αλλά και διάφορες μεταξύ τους. Η βασική δομή ωστόσο και η λογική τους είναι αυτά που έχουν φέρει τον επαναστατικό χαρακτήρα τους.

2.1 Ιστορία και Εξέλιξη του Blockchain

Αν και το πρώτο γνωστό blockchain είναι το Bitcoin Οι ρίζες της τεχνολογίας blockchain μπορούν να εντοπιστούν από παλαιότερες δημοσιεύσεις στο πεδίο της κρυπτογραφίας και της επιστήμης των υπολογιστών, με σημαντικό ορόσημο να είναι η εργασία του 1991 των Stuart Haber και W. Scott Stornetta (Haber, S., & Stornetta, 1991) με τίτλο "How to Time-stamp a Digital Document". Η εργασία πρότεινε μια μέθοδο για τη δημιουργία μιας ασφαλούς και αδιάψευστης εγγραφής ψηφιακών δεδομένων, η οποία έθεσε τα θεμέλια για δομές που μοιάζουν με blockchain και θα μπορούσαν να χρησιμοποιηθούν για τη δημιουργία μόνιμου αρχείου ψηφιακών συναλλαγών. Η μέθοδος

χρονοσήμανσης Haber και Stornetta, ήταν μια από τις πρώτες πρακτικές εφαρμογές της χρονοσφράγισης ψηφιακών εγγράφων με χρήση κρυπτογραφικών τεχνικών. Η μέθοδος περιλαμβάνει τη χρήση μιας συνάρτησης κατακερματισμού για τη δημιουργία ενός ψηφιακού δακτυλικού αποτυπώματος του εγγράφου, το οποίο στη συνέχεια αποστέλλεται σε μια αρχή χρονοσήμανσης για καταγραφή του χρόνου στο έγγραφο. Η αρχή χρονοσήμανσης δημιουργεί ένα νέο κατακερματισμό συνδυάζοντας τον αρχικό κατακερματισμό με μια χρονική σήμανση και στη συνέχεια υπογράφει ψηφιακά το νέο κατακερματισμό χρησιμοποιώντας ένα σύστημα υποδομής δημοσίου κλειδιού (PKI). Το ψηφιακά υπογεγραμμένο και αποτέλεσμα κατακερματισμού έγγραφο, επιστρέφεται στον αποστολέα, ο οποίος μπορεί να το χρησιμοποιήσει για να επαληθεύσει τη χρονική σήμανση και τη γνησιότητα του εγγράφου.

Στα τέλη της δεκαετίας του 1990, ένας άλλος ερευνητής ονόματι Nick Szabo(1997) εισήγαγε την έννοια των "έξυπνων συμβάσεων", τα οποία είναι αυτοεκτελούμενα συμβόλαια με τους όρους της συμφωνίας μεταξύ αγοραστή και πωλητή να γράφονται απευθείας σε γραμμές κώδικα. Αν και η ιδέα του Szabo δεν εφαρμόστηκε πλήρως εκείνη την εποχή, έθεσε τα θεμέλια για μελλοντικά έξυπνα συμβόλαια σε blockchain.

Το 1998, ο Wei Dai πρότεινε ένα αποκεντρωμένο σύστημα ψηφιακών νομισμάτων που ονομάζεται "b-money", το οποίο χρησιμοποίησε την κρυπτογραφία για τον έλεγχο της δημιουργίας και της μεταφοράς χρημάτων χωρίς να βασίζεται σε μια κεντρική αρχή. Ενώ το b-money δεν εφαρμόστηκε ποτέ, άνοιξε το δρόμο για την ανάπτυξη μελλοντικών συστημάτων που βασίζονται σε blockchain.

Το 1997 και το 2002, ο Jianying Zhou πρότεινε λύσεις μη απόρριψης που είναι σχετικές και συνδέονται με την ιστορία του blockchain. Οι προτάσεις του Zhou στόχευαν να αντιμετωπίσουν το ζήτημα της μη άρνησης στο πλαίσιο των

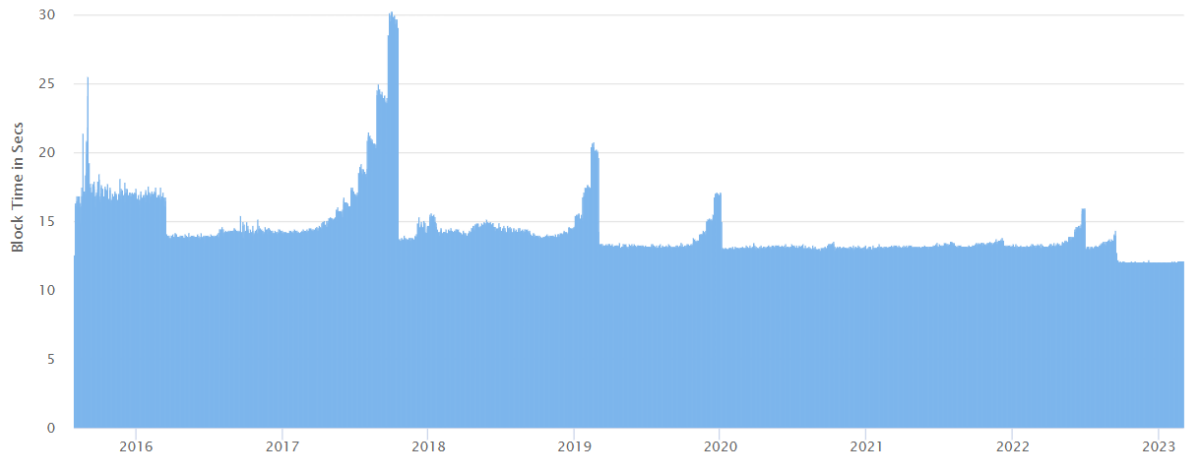
ψηφιακών συναλλαγών και των ηλεκτρονικών υπογραφών. Η ανάπτυξη της τεχνολογίας blockchain έδωσε μια λύση σε αυτό το πρόβλημα, προσφέροντας μια ασφαλή και αδιάψευστη καταγραφή των ψηφιακών συναλλαγών, επιτυγχάνοντας τη μη απόρριψη. Το έργο του Zhou συνέβαλε στην ανάπτυξη της τεχνολογίας που γνωρίζουμε σήμερα ως blockchain.

Το 2002, ο Adam Back εισήγαγε τον αλγόριθμο απόδειξης εργασίας "Hashcash", ο οποίος είχε στόχο να περιορίσει τα ανεπιθύμητα μηνύματα ηλεκτρονικού ταχυδρομείου και τις επιθέσεις άρνησης υπηρεσίας. Το Hashcash απαιτούσε από τους αποστολείς να εκτελέσουν μια υπολογιστική εργασία που θα απαιτούσε χρόνο και πόρους, περιορίζοντας έτσι τον αριθμό των μηνυμάτων που μπορούσαν να στείλουν. Ο αλγόριθμος εισήγαγε επίσης την έννοια του "nonce", μια τυχαία παραγόμενη τιμή που χρησιμοποιείται για την παραγωγή ενός κατακερματισμού με συγκεκριμένα χαρακτηριστικά (Narayanan, Arvind, et al, 2016).

Η σημαντική στιγμή για την τεχνολογία blockchain ήρθε το 2008, με τη δημοσίευση της εργασίας του Satoshi Nakamoto «Bitcoin: A Peer-to-Peer Electronic Cash System»(2008). Το έγγραφο εισήγαγε την έννοια ενός αποκεντρωμένου ψηφιακού νομίσματος στο πρώτο επιτυχημένο blockchain για να δημιουργήσει ένα ασφαλές και διαφανές αρχείο όλων των συναλλαγών. Το Bitcoin επιτρέπει στους ανθρώπους να στέλνουν και να λαμβάνουν πληρωμές χωρίς την ανάγκη κεντρικής αρχής ή μεσάζοντα. Το βιβλίο "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction"(Narayanan, Arvind, et al, 2016) καλύπτει ένα ευρύ φάσμα θεμάτων που σχετίζονται με τα κρυπτονομίσματα, συμπεριλαμβανομένης της ιστορίας και της εξέλιξης των κρυπτονομισμάτων, τις τεχνικές πτυχές της τεχνολογίας blockchain, τις νομικές και οικονομικές επιπτώσεις των κρυπτονομισμάτων. Ένα αρκετά ενδιαφέρον εισαγωγικό κεφάλαιο του βιβλίου, το “HINTS ABOUT SATOSHI”(p XXII-XXVI), παρουσιάζει μια προοπτική για τις πρώτες μέρες του Bitcoin και τον σημαντικό ρόλο που έπαιξε ο Nakamoto Satoshi σε συνδυασμό με την

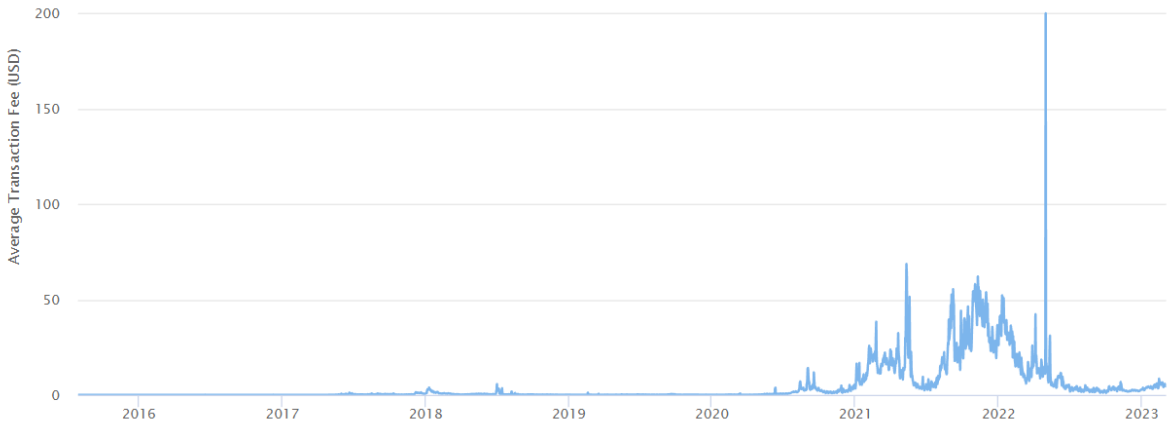
αποδοχή και τη συνεργασία της κοινότητας. Η τεχνολογία του Bitcoin έκανε την έννοια της συμμετοχής των μερών στις αποφάσεις και τη βελτιστοποίηση του συστήματος πράξη και όπως φαίνεται είναι κάτι που λάμβανε υπόψη του και ο ίδιος φερόμενος ως Satoshi.

Ο Vitalik Buterin πρότεινε το Ethereum στα τέλη του 2013, ενώ ήταν ακόμα έφηβος. Είχε συμμετάσχει στην κοινότητα του Bitcoin και μάλιστα είχε ιδρύσει το περιοδικό Bitcoin(Finley, 2014). Ωστόσο, πίστευε ότι η τεχνολογία θα μπορούσε να χρησιμοποιηθεί για περισσότερα από ένα ψηφιακό νόμισμα. Το Ethereum ανακοινώθηκε το 2014 και ξεκίνησε το 2015(Tikhomirov, 2018) τη λειτουργία του με στόχο τη δημιουργία μιας καθολικής πλατφόρμας εφαρμογών που βασίζεται σε blockchain. Ενσωματώνει μια πλήρη γλώσσα Turing, καθιστώντας θεωρητικά δυνατή την έκφραση όλων των πρακτικών υπολογισμών σε γραμμές κώδικα με μια γλώσσα που ονομάζεται Solidity. Αυτά είναι τα λεγόμενα έξυπνα συμβόλαια ή αλλιώς Smart Contracts. Η αλυσίδα μπλοκ Ethereum είναι ικανή να ανταποκρίνεται στα αιτήματα των χρηστών. Αυτή η βελτιωμένη λειτουργικότητα εισήγαγε νέες προκλήσεις ασφάλειας που σχετίζονται με το σχεδιασμό γλώσσας και την ασφάλεια στον προγραμματισμό διαδικτυακών εφαρμογών πάνω στην τεχνολογία που εισήγαγε το Bitcoin. Το Ethereum αν και δεν διέψευσε τον Buterin για τη χρησιμότητα του αντιμετώπισε αρκετες προκλήσεις.



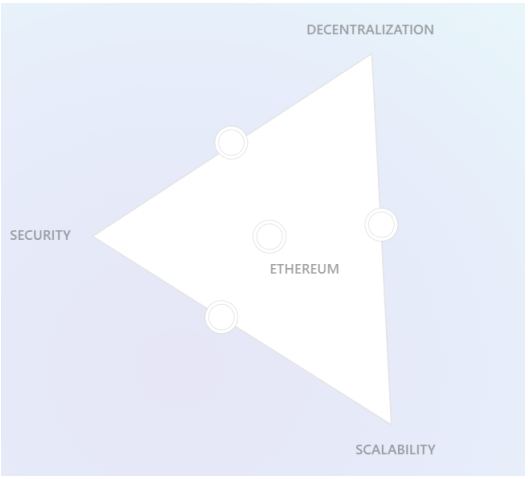
Εικόνα 2.1.1: Γράφημα Χρόνου Έγκρισης block στο Ethereum από etherscan.io

Μία από τις πιο σημαντικές προκλήσεις που αντιμετωπίζει το δίκτυο Ethereum είναι η κλιμάκωση. Η τρέχουσα έκδοση του blockchain Ethereum μπορεί να επεξεργαστεί μόνο περίπου 15-45TPS, συναλλαγές ανά δευτερόλεπτο , κάτι που είναι λιγότερο από αυτό που απαιτείται για να χειριστεί τη ζήτηση μιας παγκόσμιας αποκεντρωμένης πλατφόρμας εφαρμογών. Καθώς ο αριθμός των χρηστών και των εφαρμογών στο blockchain του Ethereum έχει αυξηθεί, το δίκτυο γίνεται όλο και πιο συμφορημένο, με πιο αργούς χρόνους συναλλαγών. Στην εικόνα 2.1 παρουσιάζεται ο χρόνος που απαιτείται για να συμπεριληφθεί ένα μπλοκ στην αλυσίδα μπλοκ Ethereum σε συνάρτηση με τα έτη. Προς τα τέλη του 2022(CNBC, 2022) υλοποιήθηκε μια μεγάλη αλλαγή(De Vries, 2023) στο μηχανισμό συναίνεσης που γίνονται αποδεκτά τα μπλοκς στο δίκτυο κάνοντας τον χρόνο αυτό σχεδόν σταθερό περίπου 12 δευτερόλεπτα.



Εικόνα 2.1.2: Γράφημα Εξόδων Συναλλαγής στο Ethereum από etherscan.io

Ακόμα μία πρόκληση στο Ethereum είναι οι υψηλές χρεώσεις συναλλαγών. Λόγω της συμφόρησης του δικτύου και της μεγάλης ζήτησης για επεξεργασία συναλλαγών, οι χρεώσεις είναι υψηλές, καθιστώντας ακριβή για τους χρήστες τη χρήση του δικτύου. Στην εικόνα 2.2 φαίνεται πως η χρέωση για μια συναλλαγή έχει αγγίξει μέχρι και την τιμή των 0.07ether, το νόμισμα που αντιστοιχεί στο blockchain ethereum και για εκείνη την περίοδο ήταν περίπου σε αντιστοιχία 200$.



Εικόνα 2.1.3: Τρίλημμα Κλιμάκωσης από etherscan.io

Η αναβάθμιση που προαναφέρθηκε αντιμετώπισε εν μέρη τις υψηλές τιμές στο τέλος συναλλαγής όμως παραμένουν υψηλές, για σχετικά μικρό όγκο συναλλαγών στο τρέχον δίκτυο. Η αναβάθμιση του δικτύου Ethereum είναι ακόμα σε εξέλιξη ενώ το δίκτυο είναι ήδη σε λειτουργία και προβλέπεται συνέχεια στη βελτιστοποίηση του να αντιμετωπίσει το λεγόμενο τρίλημμα κλιμάκωσης όπως φαίνεται στην εικόνα 2.3. Στόχος του είναι να μπορεί να εξυπηρετεί δεδομένα και εφαρμογές που στηρίζονται πάνω του διατηρώντας την ασφάλεια και τον αποκεντρωμένο χαρακτήρα του.

Από την λειτουργία του Ethereum και μετά κυκλοφόρησαν πολλά διαφορετικά πρωτόκολλα Blockchain που υποστηρίζουν τη δυνατότητα ανάπτυξης εφαρμογών καθένα με τα δικά του θετικά και αρνητικά χαρακτηριστικά. Σε αρκετά από αυτά υπάρχει συμβατότητα για εκτέλεση του συμβολαίου που έχει γραφτεί για το Blockchain Ethereum αρκεί να υποστηρίζουν την εικονική μηχανή Ethereum EVM. Η EVM είναι ένα εικονικό μηχάνημα που εκτελείται στο blockchain και επιτρέπει έξυπνες συμβάσεις γραμμένες σε γλώσσες προγραμματισμού υψηλού επιπέδου, όπως η Solidity, να εκτελούνται στο δίκτυο. Η EVM είναι υπεύθυνη για την ερμηνεία και την εκτέλεση του bytecode που δημιουργείται από τον μεταγλωττιστή για μια συγκεκριμένη έξυπνη σύμβαση. Πρόκειται για ένα κρίσιμο στοιχείο που έφερε το δίκτυο Ethereum, καθώς επιτρέπει την αποκέντρωση των εφαρμογών και την εκτέλεση των

έξυπνων συμβάσεων με αξιοπιστία. Μερικά από αυτά που υποστηρίζουν τη γλώσσα Solidity συμπεριλαμβάνοντας και άλλη γλώσσα προγραμματισμού για συγγραφή έξυπνων συμβάσεων είναι:

- Ethereum - Solidity και η γλώσσα Vyper
- Avalanche - Solidity, Go, Java
- xDai - Solidity, Vyper, Lity
- Polygon (MATIC) - Solidity, Vyper
- Binance Smart Chain - Solidity, Vyper
- Polkadot - "ink!", Rust, C++, Solidity
- Harmony - Solidity, Rust
- Ontology - Solidity, Python, Java
- Elrond - Solidity, Rust, C++
- Nervos - Solidity, C

Έξυπνες συμβάσεις με άλλη υποστήριξη γλωσσών:

- Cardano (Haskell)
- Solana (Rust)
- EOSIO (C++)
- Algorand (TEAL)
- Hyperledger Fabric (Go/JavaScript)
- Corda (Kotlin/Java)

Η εξέλιξη στο πεδίο της τεχνολογίας blockchain είναι συνεχώς αναπτυσσόμενη με γρήγορους ρυθμούς και πολλές εξελίξεις καθημερινά. Όχι μόνο αναπτύσσονται νέα blockchains αλλά αναβαθμίζονται και τα ήδη υπαρκτά. Πολλά από αυτά έχουν διαφορετικούς μηχανισμούς συναίνεσης, φθηνότερο κόστος συναλλαγής και καλύτερη συνολική κλιμάκωση στο δίκτυο. Σε κάθε περίπτωση τα δίκτυα που έχουν δοκιμαστεί σε κρίσιμες θέσεις, όπως συμφόρηση δικτύου, κυβερνοεπιθέσεις, ζωντανή αναβάθμιση και όλα τα πιθανά προβλήματα που μπορεί να προκύψουν, η αρχιτεκτονική τους και η δυναμική των ομάδων που συμμετέχουν στον προγραμματισμό τους, παίζουν

καθοριστικό ρόλο. Δεν είναι τυχαία η αναφορά στην αναβάθμιση του Ethereum καθώς αυτή ήταν μια στιγμή όπου διαφορετικές ομάδες προγραμματιστών ήταν σε αναμονή να υποστηρίξουν τα ζητήματα που θα προέκυπταν(CNBC, 2022). Μην ξεχνάμε άλλωστε πως στα περισσότερα δημόσια blockchains, τυχόν αλλαγές ή αναβαθμίσεις στο δίκτυο, συμπεριλαμβανομένων των έξυπνων ενημερώσεων συμβολαίων, απαιτούν έγκριση από τον μηχανισμό συναίνεσης, ο οποίος συνήθως επιτυγχάνεται μέσω συμφωνίας σε ολόκληρη την κοινότητα(Kiayias & Lazos (2022). Αυτό οφείλεται στο γεγονός ότι τα δημόσια blockchains έχουν σχεδιαστεί για να είναι αποκεντρωμένα καθολικά. Τυχόν αλλαγές που έγιναν στο δίκτυο μπορούν να επηρεάσουν την ασφάλεια, τη σταθερότητα και την εμπιστοσύνη στο σύστημα. Ως αποτέλεσμα, η διαδικασία λήψης αποφάσεων για αναβαθμίσεις ή αλλαγές γίνεται συχνά μέσω του μηχανισμού συναίνεσης. Αυτός είναι ένας πολύ δυνατός μηχανισμός που θα πρέπει να συμπεριλαμβάνεται στις αποφάσεις επιλογής για ανάπτυξη εφαρμογών σε ένα δίκτυο Blockchain.

2.2 Βασικά Χαρακτηριστικά της τεχνολογίας Blockchain

Για να κατανοήσουμε πώς λειτουργεί το blockchain, είναι απαραίτητο να κατανοήσουμε τα θεμελιώδη στοιχεία όπως τον κατακερματισμό, τους μηχανισμούς συναίνεσης, τις συναλλαγές, το τι είναι ένα block και τα δεδομένα τους, ώστε να σχηματιστεί ένα blockchain. Ουσιαστικά πρόκειται για μια αλυσίδα μπλοκ που περιέχει ένα αρχείο συναλλαγών που διασφαλίζονται με χρήση κρυπτογραφίας. Η τεχνολογία Blockchain είναι ένα κατανεμημένο σύστημα καθολικού που επιτρέπει την ασφαλή, διαφανή και αδιάψευστη τήρηση αρχείων των συναλλαγών. Μερικά από τα βασικά της τεχνολογίας blockchain:

Αποκέντρωση: Η τεχνολογία Blockchain είναι ένα κατανεμημένο σύστημα που επιτρέπει την ασφαλή, διαφανή και αδιάψευστη τήρηση αρχείων των συναλλαγών. Αντί για κεντρική αρχή ή μεσάζοντα, οι συναλλαγές επαληθεύονται και καταγράφονται από ένα δίκτυο κόμβων ή υπολογιστών.

Ασφάλεια: Η τεχνολογία Blockchain χρησιμοποιεί κρυπτογραφία για την ασφάλεια των δεδομένων που είναι αποθηκευμένα στο blockchain. Κάθε μπλοκ στην αλυσίδα μπλοκ συνδέεται με το προηγούμενο μπλοκ χρησιμοποιώντας μια κρυπτογραφική συνάρτηση κατακερματισμού, η οποία δημιουργεί ένα μοναδικό ψηφιακό αποτύπωμα των δεδομένων του μπλοκ. Η συνάρτηση κατακερματισμού καθιστά πολύ δύσκολη την αλλαγή οποιωνδήποτε δεδομένων στο blockchain χωρίς να ανιχνεύονται, επειδή ακόμη και η μικρότερη αλλαγή στα δεδομένα θα έχει ως αποτέλεσμα διαφορετικό κατακερματισμό.

Διαφάνεια: Η τεχνολογία Blockchain είναι διαφανής, πράγμα που σημαίνει ότι ο καθένας μπορεί να δει τις συναλλαγές στο blockchain. Ωστόσο, η ταυτότητα των μερών που εμπλέκονται στις συναλλαγές μπορεί να παραμείνει ανώνυμη.

Συναίνεση: Οι συναλλαγές στο blockchain επαληθεύονται και επικυρώνονται μέσω ενός μηχανισμού συναίνεσης, πράγμα που σημαίνει ότι πολλοί κόμβοι στο δίκτυο πρέπει να συμφωνήσουν ότι μια συναλλαγή είναι έγκυρη προτού καταγραφεί στο blockchain. Ο μηχανισμός συναίνεσης διασφαλίζει ότι τα δεδομένα στο blockchain είναι ακριβή και αξιόπιστα.

Αμετάβλητη: Μόλις καταγραφεί μια συναλλαγή στο blockchain, δεν μπορεί να τροποποιηθεί ή να διαγραφεί. Αυτό συμβαίνει επειδή η αλλαγή των δεδομένων ενός μπλοκ θα άλλαζε τον κατακερματισμό του και ο κατακερματισμός όλων των επόμενων μπλοκ θα άλλαζε επίσης. Αυτό καθιστά το blockchain ιδανική πλατφόρμα για ασφαλή και διαφανή τήρηση αρχείων σημαντικών δεδομένων.

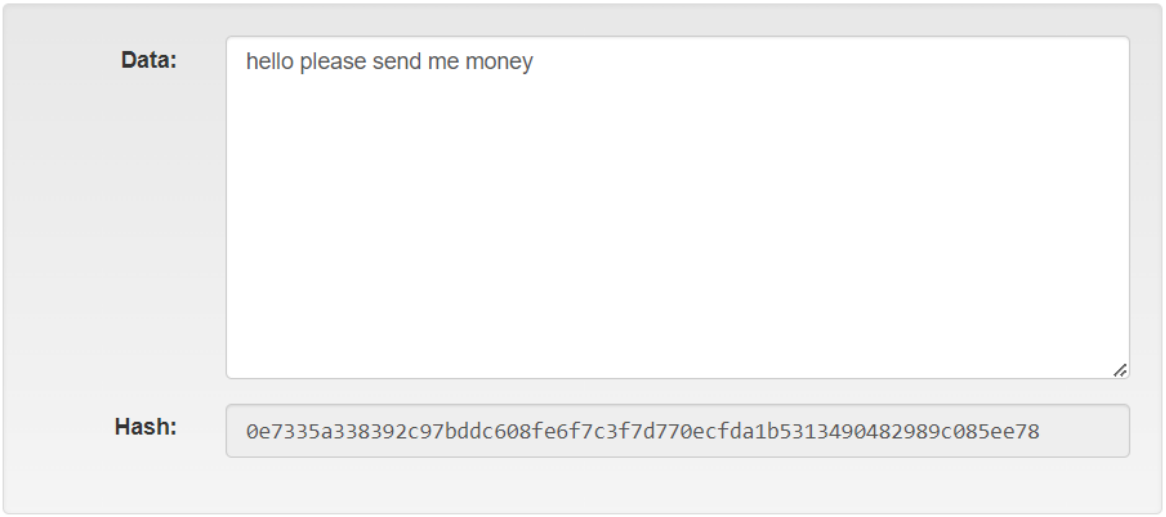
Έξυπνα συμβόλαια: Τα έξυπνα συμβόλαια είναι αυτοεκτελούμενα συμβόλαια που αποθηκεύονται στο blockchain γεγονός που τα καθιστά ασφαλή και αναλλοίωτα. Μπορούν να αυτοματοποιήσουν διάφορες διαδικασίες όπως την διαδικασία επαλήθευσης και εκτέλεσης των όρων μιας σύμβασης, μειώνοντας έτσι το κόστος και βελτιώνοντας την αποτελεσματικότητα.

Εικόνα 2.2: Χαρακτηριστικά Blockchain

2.3 Κατακερματισμός Blockchain

Στο έγγραφο, ο S. Nakamoto(2008) περιγράφει το blockchain ως "μια αλυσίδα μπλοκ, με κάθε μπλοκ να περιέχει έναν κατακερματισμό του προηγούμενου μπλοκ μέχρι το μπλοκ γένεσης της αλυσίδας" (p. 1). Αυτός ο ορισμός έχει γίνει ευρέως αποδεκτός και αναφέρεται συχνά σε ακαδημαϊκές και βιομηχανικές δημοσιεύσεις. Το hash ή κατακερματισμός όπως φαίνεται στην εικόνα 2.3(Brownworth, 2016) είναι μια μοναδική συμβολοσειρά χαρακτήρων σταθερού μήκους που δημιουργείται εκτελώντας μια μαθηματική συνάρτηση σε ένα κομμάτι δεδομένων μέσω ενός αλγορίθμου. Ο κατακερματισμός που προκύπτει είναι μοναδικός για το συγκεκριμένο τμήμα δεδομένων και λειτουργεί ως ψηφιακό δακτυλικό αποτύπωμα, επιτρέποντας την εύκολη αναγνώριση και επαλήθευση των δεδομένων (Narayanan, Arvind, et al, 2016, p 5-10).

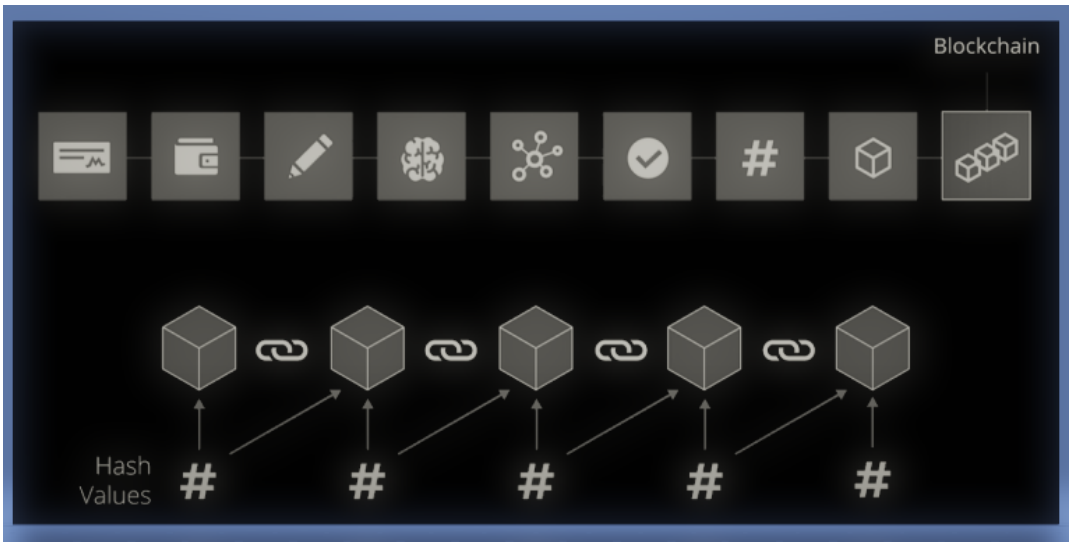
Οι συναρτήσεις κατακερματισμού είναι μονόδρομες συναρτήσεις, πράγμα που σημαίνει ότι είναι εύκολο να δημιουργηθεί ένας κατακερματισμός από τα δεδομένα εισόδου, αλλά σχεδόν αδύνατο να ανασχεδιαστούν τα αρχικά δεδομένα από τον κατακερματισμό. Αυτή η ιδιότητα κάνει τα hashes ένα ιδανικό εργαλείο για τη διασφάλιση της ακεραιότητας και της ασφάλειας των δεδομένων σε συστήματα blockchain. Στο blockchain, τα hashes χρησιμοποιούνται συνήθως για τη σύνδεση μπλοκ μεταξύ τους σε μια αλυσίδα. Κάθε μπλοκ περιέχει έναν κατακερματισμό-hash του προηγούμενου μπλοκ στην αλυσίδα, σχηματίζοντας μια ασφαλή και αδιάψευστη εγγραφή όλων των συναλλαγών και των δεδομένων που είναι αποθηκευμένα στο blockchain.



Εικόνα 2.3.1: block με κατακερματισμό SHA256 δεδομένων

Το "Hashing" είναι ένας κρυπτογραφικός αλγόριθμος που χρησιμοποιείται στην τεχνολογία blockchain για τη διασφάλιση της ακεραιότητας και της ασφάλειας των δεδομένων. Ο κατακερματισμός είναι μια μονόδρομη διαδικασία που μετατρέπει οποιαδήποτε δεδομένα εισόδου σε μια αλφαριθμητική συμβολοσειρά σταθερού μεγέθους που ονομάζεται κατακερματισμός(hash). Ο κατακερματισμός που δημιουργείται από τον αλγόριθμο κατακερματισμού είναι μοναδικός για τα δεδομένα εισόδου, πράγμα

που σημαίνει ότι οποιαδήποτε μικρή αλλαγή στα δεδομένα εισόδου θα έχει ως αποτέλεσμα έναν εντελώς διαφορετικό κατακερματισμό. Αυτό καθιστά τον κατακερματισμό χρήσιμο εργαλείο για την επαλήθευση της ακεραιότητας των δεδομένων που είναι αποθηκευμένα σε μια αλυσίδα μπλοκ.



Εικόνα 2.3.2: block, blockchain και hash values

Ένας κοινώς χρησιμοποιούμενος αλγόριθμος κατακερματισμού στην τεχνολογία blockchain είναι ο αλγόριθμος SHA-256, ο οποίος σημαίνει Secure Hash Algorithm. Ο αλγόριθμος SHA-256 χρησιμοποιείται για τη δημιουργία των κατακερματισμένων συμβολοσειρών για τα μπλοκ στην αλυσίδα μπλοκ του δικτύου Bitcoin. Ένας άλλος δημοφιλής αλγόριθμος κατακερματισμού που χρησιμοποιείται στην τεχνολογία blockchain είναι ο αλγόριθμος Keccak, ο οποίος χρησιμοποιείται στο blockchain Ethereum. Ο αλγόριθμος Keccak, γνωστός και ως SHA-3 επιλέχθηκε ως ο νικητής του διαγωνισμού του Εθνικού Ινστιτούτου Προτύπων και Τεχνολογίας (NIST) για ένα νέο πρότυπο συνάρτησης κατακερματισμού κρυπτογράφησης, επίσης γνωστό ως στις 2 Οκτωβρίου 2012. Ο διαγωνισμός ξεκίνησε το 2007 και έλαβε 64 υποβολές, οι οποίες στη συνέχεια περιορίστηκαν σε πέντε φιναλίστ πριν επιλεγεί ο Keccak ως νικητής. Η επιλογή του Keccak ως νικητή σηματοδότησε την πρώτη

αλλαγή στο Secure Hash Standard του NIST από την κυκλοφορία του SHA-2 το 2001. NIST(Kelsey et al.,2016)

Άλλος ένας αλγόριθμος κατακερματισμού που έχει τραβήξει το ενδιαφέρον είναι ο αλγόριθμος κατακερματισμού χιονοστιβάδας. Είναι μια σχετικά νέα οικογένεια κρυπτογραφικών συναρτήσεων κατακερματισμού. Ο όρος "φαινόμενο χιονοστιβάδας" χρησιμοποιήθηκε για πρώτη φορά Horst Feistel(1973) και υλοποιήθηκε το 2018 από μια ομάδα ερευνητών (Team Rocket, 2018). Ο αλγόριθμος έχει σχεδιαστεί για να είναι εξαιρετικά ανθεκτικός σε επιθέσεις, συμπεριλαμβανομένων των επιθέσεων σύγκρουσης και των επιθέσεων πριν από την εικόνα(Klarreich, 2018). Αυτό το καθιστά ελκυστική επιλογή για χρήση στην τεχνολογία blockchain, όπου η ασφάλεια είναι ζωτικής σημασίας.

Ο αλγόριθμος κατακερματισμού χιονοστιβάδας αντλεί το όνομά του από το "φαινόμενο χιονοστιβάδας", το οποίο αναφέρεται στην ιδιότητα των κρυπτογραφικών συναρτήσεων όπου μια μικρή αλλαγή στην είσοδο οδηγεί σε μεγάλη αλλαγή στην έξοδο (Boneh, 2019). Αυτή η ιδιότητα είναι σημαντική για τη διασφάλιση ότι μικρές αλλαγές στα δεδομένα εισόδου παράγουν σημαντικά διαφορετικά αποτελέσματα, γεγονός που καθιστά δύσκολο για τους εισβολείς να προβλέψουν την έξοδο ή να βρουν συγκρούσεις.

Ο αλγόριθμος κατακερματισμού χιονοστιβάδας βασίζεται σε μια μαθηματική κατασκευή που ονομάζεται κατασκευή σφουγγαριού, η οποία χρησιμοποιείται για να μετατρέψει μια είσοδο σταθερού μήκους σε έξοδο σταθερού μήκους (Tromp, 2018). Ο αλγόριθμος χρησιμοποιεί μια συνάρτηση μετάθεσης για να ανακατέψει τα δεδομένα εισόδου και, στη συνέχεια, εφαρμόζει μια συνάρτηση συμπίεσης για την παραγωγή της εξόδου. Η λειτουργία συμπίεσης έχει σχεδιαστεί για να διασφαλίζει ότι μικρές αλλαγές στην είσοδο προκαλούν μεγάλες αλλαγές στην έξοδο, γεγονός που ενισχύει την ασφάλεια του αλγορίθμου.

Από την εισαγωγή του το 2018, ο αλγόριθμος κατακερματισμού χιονοστιβάδας έχει κερδίσει δημοτικότητα ως ασφαλής και αποτελεσματικός αλγόριθμος κατακερματισμού. Έχει χρησιμοποιηθεί σε μια σειρά από έργα blockchain, συμπεριλαμβανομένου του Avalanche blockchain, το οποίο είναι μια αποκεντρωμένη πλατφόρμα που υποστηρίζει έξυπνα συμβόλαια (Avalanche, n.d.). Ωστόσο, εξακολουθεί να είναι ένας σχετικά νέος αλγόριθμος και απαιτείται περαιτέρω έρευνα για την αξιολόγηση της απόδοσης και των ιδιοτήτων ασφαλείας του με την πάροδο του χρόνου.

Συμπερασματικά, κάθε αλγόριθμος έχει τις δικές του μοναδικές ιδιότητες, όπως ταχύτητα, ασφάλεια και απαιτήσεις μνήμης, και η επιλογή του αλγορίθμου εξαρτάται από τη κάθε περίπτωση χρήσης. Ο κατακερματισμός είναι ένα κρίσιμο συστατικό της τεχνολογίας blockchain, καθώς επιτρέπει την αδιάσειστη και ασφαλή τήρηση των συναλλαγών. Με τη δημιουργία μιας ψηφιακής υπογραφής δεδομένων που είναι μοναδική και ουσιαστικά αδύνατη η αναστροφή της μηχανικής του αλγορίθμου, ο κατακερματισμός επιτρέπει τη δημιουργία ενός αμετάβλητου “αρχείου” συναλλαγών που είναι ανθεκτικό σε παραποίηση και απάτη. Η επιλογή του αλγορίθμου κατακερματισμού προσφέρει τα δικά του μοναδικά πλεονεκτήματα και αδυναμίες.

2.4 Μηχανισμοί συναίνεσης

Η συναίνεση είναι μια κρίσιμη πτυχή της τεχνολογίας blockchain που διασφαλίζει την ακεραιότητα και την ασφάλεια του δικτύου. Η συναίνεση αναφέρεται στη διαδικασία με την οποία ένα δίκτυο κόμβων συμφωνεί σχετικά με την τρέχουσα κατάσταση του καθολικού ή την ακολουθία των συναλλαγών που έχουν πραγματοποιηθεί στην αλυσίδα μπλοκ. Αυτή η διαδικασία είναι απαραίτητη για την αποτροπή διπλών δαπανών, απάτης και άλλων κακόβουλων δραστηριοτήτων στο δίκτυο. Υπάρχουν διάφοροι μηχανισμοί συναίνεσης που χρησιμοποιούνται στην τεχνολογία blockchain, ο καθένας με

τα δικά του πλεονεκτήματα και μειονεκτήματα. Μερικοί από τους πιο συνηθισμένους μηχανισμούς συναίνεσης είναι:

Proof of Work (PoW): Το PoW ή αλλιώς Απόδειξη Εργασίας, είναι ο πιο γνωστός και ευρέως χρησιμοποιούμενος μηχανισμός συναίνεσης στην τεχνολογία blockchain. Στο PoW, οι ανθρακωρύχοι(miners) ανταγωνίζονται για να λύσουν ένα κρυπτογραφικό παζλ και ο πρώτος ανθρακωρύχος που θα λύσει το παζλ ανταμείβεται με νέο κρυπτονόμισμα. Αυτός ο μηχανισμός είναι ενεργοβόρος και μπορεί να είναι αργός, αλλά έχει αποδειχθεί ότι είναι ασφαλής και ανθεκτικός σε επιθέσεις.

Εικόνα 2.4.1 hash puzzle πηγή:
blog.varonis.com/the-definitive-guide-to-cryptographic-hash-functions-part-1/

Proof of Stake (PoS): Το PoS είναι ένας μηχανισμός συναίνεσης που χρησιμοποιεί μια διαφορετική προσέγγιση στην εξόρυξη. Στο PoS, οι miners-κόμβοι επιλέγονται με βάση το ποντάρισμά τους ή το ποσό του κρυπτονομίσματος που κατέχουν. Όσο περισσότερα κρυπτονομίσματα κατέχει ένας εξορύκτης, τόσο πιο πιθανό είναι να επιλεγεί για την εξόρυξη του επόμενου μπλοκ. Αυτός ο μηχανισμός είναι ταχύτερος και πιο ενεργειακά αποδοτικός από το PoW, αλλά εξακολουθεί να είναι σχετικά νέος και μη δοκιμασμένος.

Delegated Proof of Stake (DPoS): Το DPoS είναι μια παραλλαγή του PoS που χρησιμοποιεί μια πιο δημοκρατική προσέγγιση στην εξόρυξη. Στο DPoS, οι κάτοχοι “tokens” ψηφίζουν αντιπροσώπους που είναι υπεύθυνοι για την εξόρυξη του επόμενου μπλοκ. Αυτός ο μηχανισμός είναι ταχύτερος και πιο αποτελεσματικός από το PoW και το PoS, αλλά μπορεί να είναι ευάλωτος εάν μια μικρή ομάδα αντιπροσώπων ελέγχει το δίκτυο.

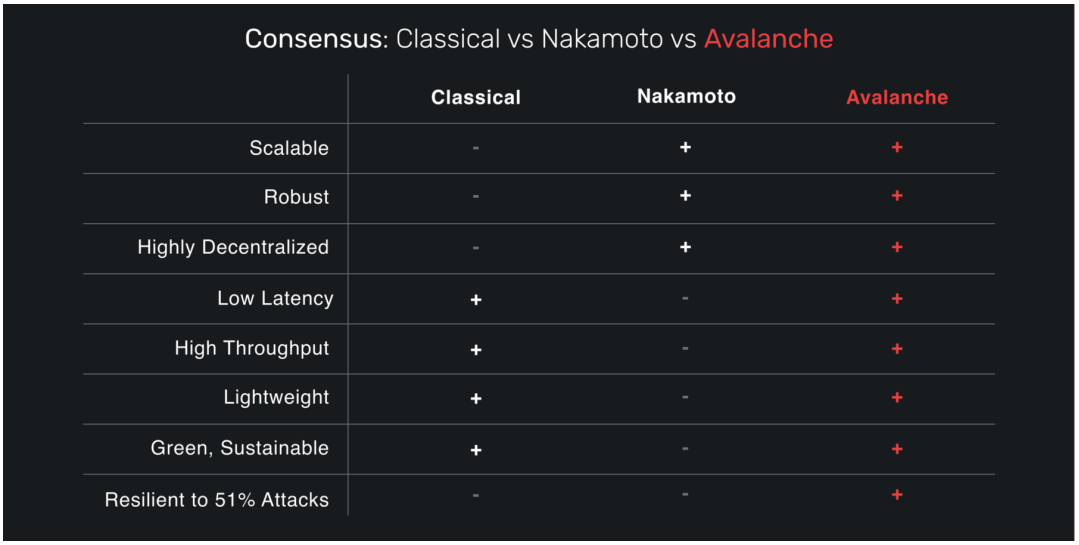
Byzantine Fault Tolerance (BFT): Το BFT είναι ένας συναινετικός μηχανισμός που έχει σχεδιαστεί για να είναι εξαιρετικά ανθεκτικός σε επιθέσεις. Στο BFT, οι κόμβοι επικοινωνούν μεταξύ τους για να επιτευχθεί συναίνεση σχετικά με την τρέχουσα κατάσταση του καθολικού. Αυτός ο μηχανισμός είναι γρήγορος και ασφαλής, αλλά μπορεί να είναι ευάλωτος σε επιθέσεις αν παραβιαστεί περισσότερο από το ένα τρίτο των κόμβων(Castro, 1999).

Πρακτική βυζαντινή ανοχή σφαλμάτων (PBFT): Το PBFT είναι μια παραλλαγή του BFT που έχει σχεδιαστεί για να είναι πιο πρακτική και αποτελεσματική. Στο PBFT, οι κόμβοι επικοινωνούν μεταξύ τους για να επιτύχουν συναίνεση σχετικά με την τρέχουσα κατάσταση του καθολικού. Αυτός ο μηχανισμός είναι ταχύτερος και πιο αποτελεσματικός από τον BFT, αλλά απαιτεί έναν σταθερό αριθμό κόμβων, που μπορεί να τον καταστήσει ευάλωτο σε επιθέσεις εάν ένας σημαντικός αριθμός κόμβων παραβιαστεί.

Απόδειξη γνώσης (PoK): Η συναίνεση του Proof of knowledge βασίζεται σε μια κρυπτογραφική έννοια που ονομάζεται αποδείξεις μηδενικής γνώσης (ZKPs), οι οποίες επιτρέπουν στους χρήστες να αποδείξουν ότι έχουν γνώση μιας συγκεκριμένης πληροφορίας χωρίς να αποκαλύπτουν αυτές τις πληροφορίες σε άλλους. Αυτό γίνεται μέσω μιας διαδικασίας που ονομάζεται ZKP, η οποία είναι μια μαθηματική απόδειξη που δείχνει ότι ένας χρήστης γνωρίζει ένα μυστικό χωρίς να αποκαλύπτει το ίδιο το μυστικό. Ένα από τα κύρια πλεονεκτήματα του PoK consensus είναι ότι δεν απαιτεί το ίδιο επίπεδο υπολογιστικής ισχύος με τους αλγόριθμους PoW ή PoS. Αυτό οφείλεται στο

γεγονός ότι η συναίνεση βασίζεται στην απόδειξη γνώσης και όχι στην επίλυση σύνθετων μαθηματικών προβλημάτων ή στην κατοχή μεγάλου όγκου κρυπτονομισμάτων. Αυτό σημαίνει ότι το PoK consensus είναι πιο ενεργειακά αποδοτικό και οικονομικά αποδοτικότερο από άλλους αλγόριθμους συναίνεσης. Ένα άλλο πλεονέκτημα του PoK consensus είναι η ασφάλειά του. Δεδομένου ότι οι χρήστες δεν χρειάζεται να αποκαλύπτουν τις μυστικές τους πληροφορίες σε άλλους, υπάρχει μικρότερος κίνδυνος υποκλοπής ή κλοπής των πληροφοριών. Αυτό καθιστά τη συναίνεση PoK πιο ασφαλή από τους αλγόριθμους PoW ή PoS, οι οποίοι μπορεί να είναι ευάλωτοι σε επιθέσεις εάν διακυβευτεί η υπολογιστική ισχύς ενός χρήστη(Pop et al, 2020).

Πρωτόκολλο Συναίνεσης Avalanche: Το πρωτόκολλο συναίνεσης Avalanche αναφέρεται συχνά ως μέλος της «οικογένειας Snow» των πρωτοκόλλων συναίνεσης, η οποία περιλαμβάνει τα Snowflake, Snowball και Avalanche. Αυτά τα πρωτόκολλα μοιράζονται ορισμένα κοινά χαρακτηριστικά και μηχανισμούς, όπως η χρήση επαναλαμβανόμενων γύρων ψηφοφορίας και η πιθανολογική δειγματοληψία, αλλά το Avalanche είναι το πιο προηγμένο και πολύπλοκο πρωτόκολλο μεταξύ τους. Έτσι, το Avalanche βασίζεται στις έννοιες και τις ιδέες των πρωτοκόλλων της οικογένειας Snow, αλλά προσθέτει πολλές πρόσθετες δυνατότητες και βελτιστοποιήσεις για να επιτύχει τις ιδιότητες υψηλής απόδοσης και ασφάλειας. Χρησιμοποιώντας τυχαία δειγματοληψία και μετασταθερότητα για την εξακρίβωση και διατήρηση των συναλλαγών, αντιπροσωπεύει μια νέα οικογένεια πρωτοκόλλων. Ο μηχανισμός συναίνεση του blockchain Avalanche είναι ιδιαίτερα περίπλοκος και αξίζει να μελετηθεί περεταίρω καθώς αποτελεί μια “πράσινη” λύση, γρήγορη και ασφαλή σε σχέση με άλλους τρόπους συναίνεσης δικτύων blockchain(Rocket Team et al, 2019).



Εικόνα 2.4.2: Σύγκριση συναίνεσης PoW Bitcoin με Avalanche του Avalanche, πηγή:https://docs.avax.network/

2.5 Έξυπνα Συμβόλαια

Τα έξυπνα συμβόλαια είναι αυτοεκτελούμενα προγράμματα που εκτελούνται σε blockchain. Έχουν σχεδιαστεί για να αυτοματοποιούν την εκτέλεση συμβάσεων, οι οποίες μπορεί να περιλαμβάνουν κανόνες και προϋποθέσεις για τη μεταφορά ψηφιακών στοιχείων ή την εκτέλεση συγκεκριμένων ενεργειών. Τα έξυπνα συμβόλαια επιτρέπουν την απρόσκοπτη και αποκεντρωμένη εκτέλεση συναλλαγών χωρίς την ανάγκη διαμεσολαβητών ή τρίτων. Δημιουργούνται χρησιμοποιώντας γλώσσες προγραμματισμού όπως η Solidity, η οποία χρησιμοποιείται για τη δημιουργία συμβολαίων σε blockchain που έχουν οριστεί με αυτή τη δυνατότητα. Τα έξυπνα συμβόλαια εκτελούνται αυτόματα μόλις εκπληρωθούν οι προκαθορισμένες προϋποθέσεις και η εκτέλεσή τους είναι ορατή σε όλους τους συμμετέχοντες στο blockchain.

Τα έξυπνα συμβόλαια έχουν πολλά πλεονεκτήματα σε σχέση με τα παραδοσιακά συμβόλαια, όπως αυξημένη διαφάνεια, μειωμένο κόστος συναλλαγών, αυξημένη ταχύτητα και αποτελεσματικότητα στην εκτέλεση των συμβάσεων. Ωστόσο, έχουν επίσης ορισμένους περιορισμούς, συμπεριλαμβανομένης της ανάγκης για εξειδικευμένες γνώσεις

προγραμματισμού και της έλλειψης νομικής αναγνώρισης σε πολλές δικαιοδοσίες.

Η διαδικασία ξεκινά με τη δημιουργία του έξυπνου συμβολαίου, το οποίο περιλαμβάνει τη σύνταξη κώδικα που καθορίζει τους κανόνες και τους κανονισμούς της συμφωνίας. Αυτός ο κώδικας είναι συνήθως γραμμένος σε μια γλώσσα προγραμματισμού που είναι συγκεκριμένη για την πλατφόρμα blockchain στην οποία θα εκτελεστεί το έξυπνο συμβόλαιο. Μόλις γραφτεί ο κώδικας, μεταφορτώνεται στο blockchain και αναπτύσσεται. Μόλις αναπτυχθεί, το έξυπνο συμβόλαιο μπορεί να εκτελεστεί αυτόματα όταν πληρούνται ορισμένες προϋποθέσεις. Για παράδειγμα, ένα έξυπνο συμβόλαιο θα μπορούσε να συνταχθεί για αυτόματη μεταφορά κεφαλαίων από το ένα μέρος στο άλλο όταν συμβαίνει ένα συγκεκριμένο συμβάν. Αυτό θα μπορούσε να προκληθεί από διάφορους παράγοντες, όπως η λήψη ενός συγκεκριμένου ποσού κρυπτονομίσματος ή η ολοκλήρωση μιας εργασίας.

Τα έξυπνα συμβόλαια μπορούν να χρησιμοποιηθούν σε διάφορους κλάδους και εφαρμογές, από τη χρηματοδότηση και την ασφάλιση έως τη διαχείριση της εφοδιαστικής αλυσίδας και την ακίνητη περιουσία. Είναι ιδιαίτερα χρήσιμα σε καταστάσεις όπου πολλά μέρη πρέπει να αλληλεπιδρούν και να συναλλάσσονται μεταξύ τους, αλλά όπου μπορεί να λείπει η εμπιστοσύνη και η διαφάνεια(Mohanta et al.,2018).

Ένα από τα βασικά πλεονεκτήματα των έξυπνων συμβολαίων είναι η ικανότητά τους να παρέχουν χρονική σήμανση(time-stamping, Haber et al., 1991) και μη απόρριψη(non-repudiation, Zhou et al.,1996). Η χρονική σήμανση αναφέρεται στην ικανότητα των έξυπνων συμβολαίων να καταγράφουν και να επαληθεύουν το χρόνο κατά τον οποίο έλαβε χώρα ένα συγκεκριμένο γεγονός ή συναλλαγή. Αυτό είναι σημαντικό για πολλές εφαρμογές, όπως νομικές συμβάσεις, όπου ο χρόνος των γεγονότων μπορεί να είναι κρίσιμος για την έκβαση της σύμβασης.

Η μη άρνηση, από την άλλη πλευρά, αναφέρεται στην αδυναμία ενός μέρους να αρνηθεί ότι έχει προβεί σε μια συγκεκριμένη ενέργεια ή ότι έχει κάνει μια συγκεκριμένη δήλωση. Τα έξυπνα συμβόλαια μπορούν να παρέχουν μη απόρριψη καταγράφοντας όλες τις ενέργειες και τις συναλλαγές στο blockchain. Αυτό σημαίνει ότι από τη στιγμή που μια συναλλαγή ή μια ενέργεια έχει καταγραφεί στο blockchain, δεν μπορεί να διαγραφεί ή να τροποποιηθεί, παρέχοντας υψηλό επίπεδο διαφάνειας και λογοδοσίας.

Εκτός από τη χρονική σήμανση και τη μη απόρριψη, τα έξυπνα συμβόλαια μπορούν να προσφέρουν μια σειρά από άλλα οφέλη, όπως η αυτοματοποίηση των διαδικασιών, η μείωση του κόστους και η αύξηση της αποτελεσματικότητας. Μπορούν επίσης να βοηθήσουν στην εξάλειψη της ανάγκης για μεσάζοντες ή έμπιστους τρίτους, γεγονός που μπορεί να μειώσει περαιτέρω το κόστος και να αυξήσει την ταχύτητα και την ασφάλεια.

Συνολικά, τα έξυπνα συμβόλαια είναι ένα ισχυρό εργαλείο για τη δημιουργία διαφανών, ασφαλών και αποτελεσματικών συστημάτων για ένα ευρύ φάσμα εφαρμογών και η ικανότητά τους να παρέχουν χρονική σήμανση και μη αποκήρυξη τα καθιστά ιδιαίτερα πολύτιμα για χρήση σε νομικές και οικονομικές εφαρμογές. Τα έξυπνα συμβόλαια χρησιμοποιούνται συνήθως σε αποκεντρωμένες εφαρμογές (dApps) που θα αναφερθούν παρακάτω και χτίζονται σε πλατφόρμες blockchain όπως το Ethereum.

Αυτα τα dApps μπορούν να χρησιμοποιηθούν για τη διευκόλυνση ενός ευρέος φάσματος συναλλαγών. Οι Συναλλαγές στα πρωτόκολλα blockchain που υποστηρίζουν έξυπνες συμβάσεις χωρίζονται σε δύο τύπους. Στις συναλλαγές κλήσης μηνυμάτων και στις συναλλαγές δημιουργίας συμβολαίων. Αυτή η διαφορά είναι σημαντική για το περιεχόμενο των δεδομένων των blocks. Μερικές από αυτές τις συναλλαγές μπορεί να είναι για εφαρμογές συμβολαίων που ρυθμίζουν δανεισμό peer-to-peer, εφαρμογές για τη διαχείριση της αλυσίδας εφοδιασμού, ιατρικά δεδομένα, επαλήθευση ψηφιακής ταυτότητας

και γενικά για όποια εφαρμογή εκτελείται στο blockchain. Συνολικά μπορούν να χρησιμοποιηθούν για ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένων των χρηματοοικονομικών συμβάσεων, δημόσιες μεταφορές, των συστημάτων ψηφοφορίας και άλλων.

2.6 Αποκεντρωμένες εφαρμογές DApps

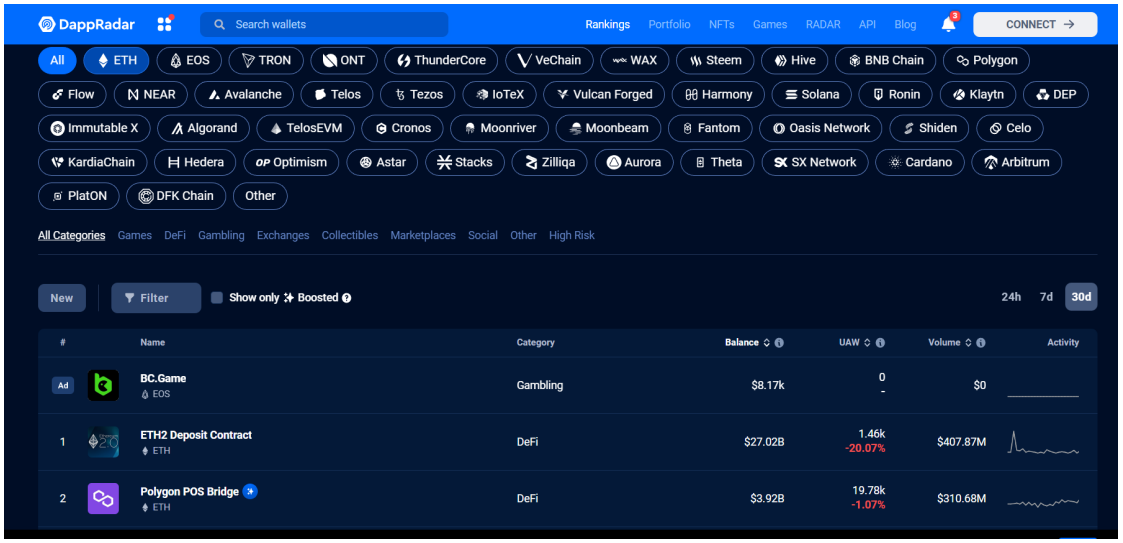
Οι αποκεντρωμένη εφαρμογή, ή σύντομα DApp, είναι μια εφαρμογή που έχει το backend(τη πίσω λειτουργία) κώδικα που εκτελείται σε ένα αποκεντρωμένο δίκτυο peer-to-peer. Στην περίπτωση του Ethereum που εισήγαγε τις αποκεντρωμένες εφαρμογές, αυτό το αποκεντρωμένο δίκτυο είναι ο παγκόσμιος υπολογιστής των πολλαπλών κόμβων που το υποστηρίζουν σε ξεχωριστούς υπολογιστές διαφόρων χρηστών ή miners όπως αναφέρθηκαν παραπάνω. Αυτό είναι διαφορετικό από τις τυπικές εφαρμογές γιατί όλες αυτές οι εφαρμογές εκτελούνται συνήθως σε κεντρικούς διακομιστές μιας εταιρείας ή ενός οργανισμού.

Με τον ίδιο τρόπο που προσπάθησε να βγάλει το bitcoin τους μεσάζοντες στις χρηματοοικονομικές συναλλαγές, το Ethereum επιχείρησε να το κάνει για ολόκληρες εφαρμογές με επιτυχία. Εφόσον η εικονική μηχανή Ethereum EVM είναι σε θέση να επεξεργάζεται τη λογική ολοκλήρωσης του Turing, αυτά τα DApps μπορούν να λειτουργήσουν παρόμοιες τυπικές εφαρμογές με τη διαφορά να είναι στο κατανεμημένο δίκτυο που τα υποστηρίζει. Υπάρχουν και άλλες πλατφόρμες στις οποίες μπορούν να δημιουργηθούν DApps όπως Avalanche, EOS και Neo, αλλά το Ethereum ακόμα είναι η κορυφαία πλατφόρμα στον κόσμο για τη δημιουργία κατανεμημένων εφαρμογών Gherghelas(2022).

Αφού δημιουργηθεί το back-end που συνδέεται με την πλατφόρμα Ethereum, η συνέχεια της δημιουργίας αυτής της εφαρμογής γίνεται με ήδη γνωστούς τρόπους. Χρησιμοποιώντας HTML, CSS, JavaScript, και οποιαδήποτε πλαίσια

ιστού ή εργαλεία στα οποία θα προτιμούσε κάποιος να εργαστεί. Μπορούν να δημιουργηθούν ολόκληρα μπροστινά άκρα(front-end) στην αποκεντρωμένη εφαρμογή με την οποία μπορούν να αλληλεπιδράσουν οι χρήστες. Το βασικό κομμάτι κώδικα που συνδέει τους χρήστες με ένα δίκτυο, όπως το δίκτυο Ethereum μέσω της διεπαφής, είναι το έξυπνο συμβόλαιο. Αυτό το έξυπνο συμβόλαιο καθορίζει τον σκοπό της εφαρμογής και επιτρέπει στους χρήστες να ορίζουν και να ανακτούν δεδομένα από το δίκτυο Ethereum.

Παρόμοια με το Play Store του λειτουργικού κινητών Android ή το Apple App Store, το Ethereum έχει ένα σύνολο εφαρμογών live. Μπορούν να αναζητηθούν σε πρόγραμμα περιήγησης, αλλά ένα χρήσιμο εργαλείο είναι το Dappradar(2022). Το Dappradar κατηγοριοποιεί πολλά από τα DApps και τα βαθμολογεί με βάση τους χρήστες και τον όγκο. Αυτό δίνει την ευκαιρία για αναζήτηση και διερεύνηση κάποιων από τις κορυφαίες εφαρμογές αυτήν τη στιγμή. Επίσης, στις κινητές συσκευές η περιήγηση μπορεί να γίνει μέσα από browser που διαθέτουν τα πορτοφόλια όπως Metamask ή Trust. Ωστόσο να τονιστεί ότι ο χρήστης θα πρέπει να μελετήσει τη λειτουργία των πορτοφολιών, να διερευνά και να γνωρίζει πριν κάνει οποιαδήποτε ενέργεια σε αυτά. Για να αλληλεπιδράσουν τα πορτοφόλια με το DApp χρειάζεται το αντίστοιχο τέλος συναλλαγής και να γνωρίζει ο χρήστης τι κάνει.



Εικόνα 2.6: DappRadar

ΚΕΦΑΛΑΙΟ 3

3. Ψηφιακή Επικοινωνία και Ανταλλαγή Μηνυμάτων

Στα σύγχρονα συστήματα επικοινωνίας, οι ψηφιακές τεχνικές χρησιμοποιούνται ευρέως για την κωδικοποίηση, τη μετάδοση και την αποκωδικοποίηση μηνυμάτων. Αυτές οι τεχνικές περιλαμβάνουν κρυπτογράφηση, ψηφιακές υπογραφές και λειτουργίες κατακερματισμού. Η κρυπτογράφηση είναι η διαδικασία μετατροπής ενός μηνύματος σε μη αναγνώσιμη μορφή, η οποία μπορεί να αποκωδικοποιηθεί μόνο από εξουσιοδοτημένα μέρη που διαθέτουν το κλειδί αποκρυπτογράφησης. Οι ψηφιακές υπογραφές, από την άλλη πλευρά, χρησιμοποιούν ασύμμετρη κρυπτογράφηση κλειδιού για τον έλεγχο ταυτότητας του αποστολέα του μηνύματος και τη διασφάλιση της ακεραιότητας του μηνύματος. Οι συναρτήσεις κατακερματισμού χρησιμοποιούνται για τη δημιουργία συνοπτικών μηνυμάτων σταθερού μήκους που αντιπροσωπεύουν μοναδικά τα δεδομένα εισόδου και μπορούν να χρησιμοποιηθούν για τον εντοπισμό τυχόν μη εξουσιοδοτημένων αλλαγών στο περιεχόμενο του μηνύματος(Zhang et al., 2021).

Μία από τις κύριες προκλήσεις της ψηφιακής επικοινωνίας είναι η διασφάλιση ότι τα μηνύματα μεταδίδονται με ασφάλεια και ότι δεν παραβιάζονται κατά τη μετάδοση. Η κρυπτογράφηση χρησιμοποιείται συχνά για την προστασία του απορρήτου των μηνυμάτων, ενώ ο κατακερματισμός χρησιμοποιείται για την επαλήθευση της ακεραιότητας του μηνύματος και τη διασφάλιση ότι δεν έχει παραβιαστεί. Οι ψηφιακές υπογραφές χρησιμοποιούνται για τον έλεγχο ταυτότητας του αποστολέα του μηνύματος και για την επαλήθευση ότι το μήνυμα δεν έχει τροποποιηθεί. Τα συστήματα κλειδιού(PKI) χρησιμοποιούνται για την παροχή ενός αξιόπιστου πλαισίου για την ανταλλαγή δημόσιων κλειδιών μεταξύ των μερών που εμπλέκονται στην επικοινωνία. Αυτό το πλαίσιο περιλαμβάνει τη χρήση ενός αξιόπιστου τρίτου μέρους, γνωστού ως

Certificate Authority (CA), το οποίο εκδίδει ψηφιακά πιστοποιητικά σε μέρη που εμπλέκονται στην επικοινωνία. Αυτά τα πιστοποιητικά περιέχουν το δημόσιο κλειδί του παραλήπτη, το οποίο χρησιμοποιείται για την κρυπτογράφηση των μηνυμάτων που αποστέλλονται σε αυτόν τον παραλήπτη.

`Οι γνωστές υπηρεσίες ανταλλαγής μηνυμάτων συνοπτικά χρησιμοποιούν διαδικασίες διασφάλισης της ασφάλειας και διαθεσιμότητας των μηνυμάτων των χρηστών οι οποίες έχουν αποδειχθεί πολύ αποτελεσματικές. Η εμπιστευτικότητα της πληροφορίας είναι ένα ζήτημα που αποδεδειγμένα προσφέρεται στις παραδοσιακές τεχνικές με την κρυπτογράφηση. Οι συμμετρικοί και οι μη συμμετρικοι αλγόριθμοι που χρησιμοποιούνται προσφέρουν υψηλά επίπεδα προστασίας. Για την ακεραιότητα των μηνυμάτων το ταξίδι των δεδομένων από τον αποστολέα στον παραλήπτη είναι το ίδιο επικίνδυνο τόσο για το blockchain όσο και για τις παραδοσιακές μεθόδους.

Η χρήση της τεχνολογίας blockchain χρησιμοποιεί όλες τις παραπάνω τεχνικές ασφαλείας(Menezes et al., 2018 p 489-534), παρέχοντας επιπλέον έναν αμετάβλητο και ασφαλή τρόπο καταγραφής δεδομένων μηνυμάτων. Η ανάγκη συστήματα επικοινωνίας που μπορούν να εγγυηθούν την ασφάλεια, την εμπιστευτικότητα, την ακεραιότητα και τη διαθεσιμότητα των μηνυμάτων είναι δεδομένη. Η τεχνολογία Blockchain έχει αναδειχθεί ως μια πολλά υποσχόμενη λύση σε αυτές τις προκλήσεις, παρέχοντας έναν αποκεντρωμένο, αξιόπιστο και διαφανή μηχανισμό για την καταγραφή και την επαλήθευση των συναλλαγών (Swan, 2015). Αυτό δεν θα ήταν αρκετό για να αντικαταστήσει γνωστές μεθόδους καθώς συνδυάζεται με επιπλέον προκλήσεις. Η τεχνολογία blockchain είναι μία καινούργια τεχνολογία και για τη χρήση της υπάρχει κόστος τόσο για τους προγραμματιστές όσο και για τους χρήστες. Επιπλέον, όπως έχει αναφερθεί υπάρχουν και αρκετοί τεχνικοί περιορισμοί που πιθανόν θα χρειαστεί χρόνος για να επιλυθούν.

Σε μια αλυσίδα μπλοκ, κάθε μπλοκ περιέχει μια λίστα συναλλαγών που συνδέονται κρυπτογραφικά με το προηγούμενο μπλοκ, σχηματίζοντας μια άθραυστη αλυσίδα εγγραφών. Οι κόμβοι στο δίκτυο διατηρούν ένα αντίγραφο του blockchain και χρησιμοποιούν αλγόριθμους συναίνεσης για να διασφαλίσουν ότι όλα τα αντίγραφα είναι συνεπή και ακριβή. Τα δεδομένα που καταγράφονται στο blockchain είναι αδιάβλητα με ταυτότητα και ψηφιακή υπογραφή στο κάθε μπλοκ που δεν τροποποιείται. Η αξιοποίηση αυτών των χαρακτηριστικών μπορεί να είναι κεντρικό σημείο που έχει νόημα να μελετηθεί η χρήση της τεχνολογίας blockchain στην ασφαλή ανταλλαγή μηνυμάτων.

Χρησιμοποιώντας ένα πλαίσιο ανταλλαγής μηνυμάτων που βασίζεται σε blockchain, τα μηνύματα μπορούν να αποθηκευτούν στο blockchain με τρόπο που διασφαλίζει ότι δεν μπορούν να παραβιαστούν ή να διαγραφούν. Τα έξυπνα συμβόλαια μπορούν επίσης να χρησιμοποιηθούν για να διασφαλιστεί ότι όλα τα μέρη που εμπλέκονται σε μια επικοινωνία συμμορφώνονται με προκαθορισμένους κανόνες και ότι τυχόν διαφορές που προκύπτουν κατά τη διαδικασία μπορούν να πιστοποιηθούν από συγκεκριμένες αρχές. Σύμφωνα με αυτά λοιπόν είναι σκόπιμο όταν υλοποιείται μία εφαρμογή αξιοποιώντας την τεχνολογία blockchain όταν η περίπτωση χρήσης της λύνει περισσότερα προβλήματα από αυτά που θα προκύψουν για την υλοποίησή της.

3.1 Συνεισφορά Blockchain στην Ανταλλαγή και μη Αποποίηση Μηνυμάτων

Η τεχνολογία Blockchain μπορεί να παρέχει μια ασφαλή και αξιόπιστη μέθοδο για την ανταλλαγή μηνυμάτων, με τη χρήση έξυπνων συμβολαίων για τη διασφάλιση της συμμόρφωσης με προκαθορισμένους κανόνες και την αμετάβλητη καταγραφή των δεδομένων στο blockchain. Η δημιουργία ενός συστήματος ανταλλαγής μηνυμάτων σε ένα blockchain περιλαμβάνει τη χρήση κόμβων για τη διατήρηση της ακεραιότητας του blockchain, καθώς και ενός

αποκεντρωμένου δικτύου για να διασφαλιστεί η αμεταβλητότητα των μηνυμάτων. Αυτό επιτρέπει τη δημιουργία μιας ασφαλούς πλατφόρμας ανταλλαγής μηνυμάτων που δεν μπορεί να παραβιαστεί ή να διαγραφεί, διατηρώντας τις τεχνικές ασφαλείας που υπάρχουν στα παραδοσιακά συστήματα ηλεκτρονικών μηνυμάτων προσφέροντας όμως και τα επιπλέον χαρακτηριστικά της χρονοσφράγισης και μη αποποίησης.

Με βάση τα παραπάνω λοιπόν η χρήση της τεχνολογίας blockchain στην ασφαλή ανταλλαγή μηνυμάτων μπορεί να χρησιμοποιηθεί για τη διασφάλιση της μη αποποίησης με απλούς τρόπους. Ο κύριος σκοπός μιας τέτοιας εφαρμογής θα ήταν για τη διασφάλιση:

1. Της ταυτότητας του αποστολέα. Ο αποστολέας αφού σταλεί το μήνυμα μεταφέρεται στον παραλήπτη και καταχωρείται στο block της αλυσίδας. Είναι αποδεδειγμένα αυτός που ισχυρίζεται.
2. Το περιεχόμενο του μηνύματος. Ένα μήνυμα δεν μπορεί να χαθεί συνολικά και το περιεχόμενο του δεν μπορεί να αλλάξει.
3. Τη χρονική στιγμή της αποστολής. Ο χρόνος της αποστολής καταγράφεται με ακρίβεια και δεν μπορεί να τροποποιηθεί ούτε να διαγραφεί. Ακόμα και να οριστεί από ένα έξυπνο συμβόλαιο κάποιος άλλος μηχανισμός για την εμφάνισή ή όχι το block της καταγραφής δεν μπορεί να τροποποιηθεί αφού καταχωρηθεί στην αλυσίδα.
4. Την ταυτότητα του παραλήπτη. Ο παραλήπτης είναι συγκεκριμένος και μόνο ο χρήστης που προορίζεται για αυτόν το μήνυμα μπορεί να το παραλάβει. Ισχύουν για αυτόν τα ίδια που ισχύουν για την ταυτότητα του αποστολέα.
5. Τον έλεγχο από τις αρχές του συστήματος. Οι αρχές μπορούν να ελέγξουν και κατ’ επέκταση να πιστοποιήσουν την ορθότητα των ισχυρισμών του αποστολέα ή του παραλήπτη για την μεταξύ τους αλληλεπίδραση. Στην προκειμένη περίπτωση η ορθότητα των

ισχυρισμών αφορά την αποστολή ενός μηνύματος από ένα χρήστη σε έναν άλλον με συγκεκριμένο περιεχόμενο και προκαθορισμένο χρόνο.

Η ενέργεια των χρηστών που απαιτείται για να χρησιμοποιήσουν ένα τέτοιο σύστημα ανταλλαγής μηνυμάτων είναι να δημιουργήσουν ένα πορτοφόλι του λογαριασμού τους στο blockchain δίκτυο που φιλοξενεί την εφαρμογή ή να χρησιμοποιήσουν κάποιο που έχουν ήδη και απλά να προσθέσουν, αν δεν είναι στο ίδιο δίκτυο blockchain, το δίκτυο που είναι χτισμένη πάνω η εφαρμογή. Η διεύθυνση του πορτοφολιού του χρήστη αποτελεί και την ταυτότητα του.

3.2 Σεναρίων Χρήσης Εφαρμογής Ανταλλαγή Μηνυμάτων σε Blockchain.

Υπάρχουν πολλά σενάρια όπου μια εφαρμογή ανταλλαγής μηνυμάτων που βασίζεται σε blockchain μπορεί να είναι χρήσιμη για μη απόρριψη και χρονική σήμανση. Ακολουθούν μερικά παραδείγματα:

Νομικές συμβάσεις: Στις νομικές συμβάσεις, είναι σημαντικό να υπάρχει μια χρονική σήμανση και εγγραφή του πότε τα μέρη συμφώνησαν και με ποιους όρους. Χρησιμοποιώντας μια εφαρμογή ανταλλαγής μηνυμάτων που βασίζεται σε blockchain, τα μέρη μπορούν να καταγράψουν τη συμφωνία τους, η οποία στη συνέχεια φέρει χρονική σήμανση και δεν μπορεί να τροποποιηθεί ή να απορριφθεί.

Πνευματική ιδιοκτησία: Μια εφαρμογή ανταλλαγής μηνυμάτων που βασίζεται σε blockchain μπορεί να χρησιμοποιηθεί για την καταγραφή της πνευματικής ιδιοκτησίας, όπως διπλώματα ευρεσιτεχνίας ή εμπορικά σήματα. Αυτό θα διασφάλιζε σε μια επικοινωνία ποιος κατέχει την ιδιοκτησία με χρονοσήμανση και δεν μπορεί να αλλοιωθεί ή να απορριφθεί.

Ιατρικά αρχεία: Οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain μπορούν να χρησιμοποιηθούν για την καταγραφή ιατρικών αρχείων, συμπεριλαμβανομένων των αποτελεσμάτων των εξετάσεων, των θεραπειών και των συνταγών. Αυτό διασφαλίζει ότι οι πληροφορίες έχουν χρονική σήμανση και δεν μπορούν να τροποποιηθούν ή να απορριφθούν, κάτι που μπορεί να είναι κρίσιμο σε νομικές διαδικασίες.

Δημόσιες Υπηρεσίες: Στις κρατικές υπηρεσίες, οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain μπορούν να χρησιμοποιηθούν για την επικοινωνία μεταξύ πολιτών και δημοσίων φορέων για την ανταλλαγή και αποστολή πιστοποιητικων γέννησης, άδειες γάμου και έγγραφα ιδιοκτησίας διασφαλίζοντας ότι έχουν χρονοσήμανση και δεν μπορούν να τροποποιηθούν ή να απορριφθούν. Αυτό μπορεί να βοηθήσει στη βελτίωση της αποτελεσματικότητας των κρατικών υπηρεσιών και στην ενίσχυση της εμπιστοσύνης μεταξύ των πολιτών και των κυβερνητικών θεσμών.

Για την κατανόηση της χρήσης μιας τέτοιας εφαρμογής θα πάρουμε παράδειγμα τη διαδικασία παραγωγής μιας εταιρείας λογισμικού. Ένας προγραμματιστής ή μία εταιρεία συνήθως δεσμεύεται για αυτό που θα παραδώσει στον ενδιαφερόμενο. Σε αυτή τη δέσμευση δεσμεύονται και οι δύο πλευρές και υπάρχουν ακόμα και ποινές στις καθυστερήσεις. Από την αρχή μέχρι το τέλος της παράδοσης του λογισμικού μπορεί να χρειαστεί αρκετές φορές η αλληλεπίδραση μεταξύ των χρηστών. Τόσο για τη δημιουργία όσο και για τους χρόνους της παράδοσης πραγματοποιείται μία συμφωνία μεταξύ των μερών. Τι είναι αυτό που παραδίδεται, σε τι χρονικό διάστημα, τι έχει συμφωνηθεί για το έργο που παραδίδεται. Αν η επικοινωνία για ένα έργο γίνει μέσω ενός τέτοιου συστήματος ότι έχει επικοινωνηθεί σε επιμέρους συζητήσεις μπορεί να εξακολουθήσει να είναι αποδεικτικό συμφωνίας. Στην παραγωγή λογισμικού πολύ συχνά προκύπτουν ζητήματα που είτε ο προγραμματιστής δεν έχει λάβει γνώση για κάποιο δεδομένο,είτε δεν κατανόησε, είτε δεν του ζητήθηκαν εξαρχής και για την υλοποίηση αυτών είναι

επιπλέον χρόνος. Υπάρχουν αρκετα διαφορετικά εργαλεία και προγράμματα. Μία τέτοια εφαρμογή όμως θα έκανε αυτή τη διαδικασία συγκεντρωτική και απόλυτα δεσμευτική ότι πράγματι συνέβη ακόμα και στην περίπτωση που υπάρχει διαφωνία μεταξύ των μερών. Σε αυτό το παράδειγμα η επικοινωνία πραγματοποιείται ανάμεσα σε έναν προγραμματιστή μιας εταιρείας πού είναι ο υπεύθυνος για την ανάπτυξη, με τον προϊστάμενο των πληροφοριακών συστημάτων μιας Τράπεζας, πού είναι ο υπεύθυνος να μεταφέρει τα ζητούμενα στην εταιρεία που έχει αναλάβει το έργο. Σε κάποιο στάδιο των παραδόσεων υπάρχει διαφωνία μεταξύ των μερών για ένα μέρος. Ο προϊστάμενος της Τράπεζας ισχυρίζεται ότι το είχε ζητήσει στον προγραμματιστή και ήταν και αυτό στα πλαίσια του πιθανού συμβολαίου που έχει η τράπεζα με την εταιρεία προγραμματισμού για το έργο. Ο προγραμματιστής διαφωνεί. Σε αυτή την περίπτωση θα μπορούσε η αρχή authority να είναι κάποιο πρόσωπο ανώτερης βαθμίδας από την τράπεζα και από την εταιρεία προγραμματισμού όπως ο project manager. Στην περίπτωση που θα διαφωνούσαν και αυτοί οι δύο θα μπορούσε να μπει ως ελεγκτικός μηχανισμός είτε οι δικηγόροι τους ή ακόμα και κάποια δικαστική αρχή.

Στο δημόσιο τομέα. Στη θέση του πελάτη θα μπορούσε να είναι ένας δημόσιος φορέας που είναι υπεύθυνος για κάποιο έργο που έχει δοθεί με τη διαδικασία του διαγωνισμού. Μία κρατική οντότητα μπορεί να ενημερώνει τους πολίτες της για δικαιώματα που πρέπει να ασκηθούν σε συγκεκριμένα χρονικά πλαίσια ή υποχρεώσεις οι οποίες πρέπει να καλυφθούν, επίσης, σε συγκεκριμένα χρονικά πλαίσια. Ένα δεσμευτικό σύστημα ανταλλαγής μηνυμάτων που υλοποιείται μέσω της τεχνολογίας blockchain μπορεί να πιστοποιεί, ενώπιον άλλης αρχής (π.χ. δικαστικής) την αποστολή της ενημέρωσης και την αποστολή ή μη της απάντησης καθώς και το περιεχόμενό τους.

Σε συστήματα παραγγελιών. Επικοινωνία θα μπορούσε να συμβαίνει μέσα από μία τέτοια εφαρμογή ώστε να πραγματοποιείται η παραγγελία αλλά να αποτελεί ταυτόχρονα και μία δεσμευτική προσφορά με αποδοχή της από τον

έναν προς τον άλλον. Επιπλέον, δέσμευση για τον χρόνο και τρόπο παράδοσης. Αφού έχει αποσταλεί η παραγγελία με ένα απλό μήνυμα ο αποστολέας μπορεί να επιβεβαιώσει ότι πράγματι την έχει στείλει τη στιγμή που την έστειλε και έχει ενημερώσει σε συγκεκριμένο χρόνο. Επιπλέον θα μπορούσαν να αποστέλλονται και έγγραφα μέσα από ένα τέτοιο σύστημα. Το σύστημα με αυτό τον τρόπο αποτελεί ένα ισχυρό αποδεικτικό όχι μόνο τον εγγράφων αλλά και της ενέργειας της αποστολής και παραλαβής του ενός μέρους από το άλλο μέρος όπως ένα τιμολόγιο στη συγκεκριμένη περίπτωση.

Συνολικά, οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain μπορούν να είναι χρήσιμες σε οποιοδήποτε σενάριο που απαιτείται μια εγγραφή επικοινωνίας με χρονική σήμανση και μη άρνηση στην αλληλεπίδραση των χρηστών. Συνοψίζοντας μια τέτοια υπηρεσία μπορεί να χρησιμοποιηθεί για να παρέχει:

- Δεσμευτικές προσφορές
- Δεσμευτικούς χρόνους
- Δεσμευτικές αποφάσεις
- Δεσμευτικές ενημερώσεις
- Δεσμευτικές επιβεβαιώσεις
- Σύντομη επίλυση διαμάχης σε διαφωνίες που προκύπτουν κατά τη διάρκεια των δεσμεύσεων που πραγματοποιήθηκαν κατά την επικοινωνία.

3.3 Πλαίσια ανταλλαγής μηνυμάτων που βασίζονται σε blockchain

Έχουν προταθεί αρκετά πλαίσια ανταλλαγής μηνυμάτων που βασίζονται σε blockchain, όπως το Bitmessage, το Obsidian και το Dust. Το Bitmessage είναι ένα σύστημα ανταλλαγής μηνυμάτων peer-to-peer που χρησιμοποιεί ένα αποκεντρωμένο δίκτυο και έναν μηχανισμό συναίνεσης Proof-of-Work για τη διασφάλιση της αυθεντικότητας των μηνυμάτων. Ένα ακόμα ζήτημα στον

τεχνολογικό χώρο του blockchain είναι πως όσο πιο μικρό είναι ένα δίκτυο τόσο πιο επιρρεπές είναι. Το Bitmessage έχει παρουσιάσει ένα από τα πιο σοβαρά θέματα που θα μπορούσαν να προκύψουν σε μια εφαρμογή blockchain, αυτό της απομακρυσμένης εκτέλεσης κώδικα. Το Obsidian είναι μια πλατφόρμα ανταλλαγής μηνυμάτων με επίκεντρο το απόρρητο που χρησιμοποιεί ένα κατανεμημένο δίκτυο και ένα σύστημα για την πρόληψη ανεπιθύμητων μηνυμάτων και τη διασφάλιση της παράδοσης μηνυμάτων. Το Dust είναι μια άλλη πλατφόρμα ανταλλαγής μηνυμάτων που βασίζεται σε blockchain που χρησιμοποιεί ένα αποκεντρωμένο δίκτυο και κρυπτογράφηση από άκρο σε άκρο για την παροχή ασφαλών μηνυμάτων.

Ένα παράδειγμα ενός πλαισίου ανταλλαγής μηνυμάτων που βασίζεται σε blockchain είναι το πρωτόκολλο Whisper, το οποίο αποτελεί μέρος της πλατφόρμας blockchain Ethereum. Το Whisper παρέχει ένα ασφαλές, αποκεντρωμένο σύστημα ανταλλαγής μηνυμάτων που επιτρέπει στους χρήστες να επικοινωνούν ιδιωτικά χωρίς την ανάγκη κεντρικού διακομιστή. Τα μηνύματα που αποστέλλονται μέσω του Whisper είναι κρυπτογραφημένα και ο αποστολέας και ο παραλήπτης παραμένουν ανώνυμοι. Το Whisper παρέχει επίσης ένα μέσο μετάδοσης μηνυμάτων σε πολλούς παραλήπτες, το οποίο είναι χρήσιμο για εφαρμογές όπως οι ροές ειδήσεων.

Ακόμα μια εφαρμογή είναι το σύστημα Enigma, το οποίο έχει ένα κατανεμημένο δίκτυο κόμβων για την ροή των μηνυμάτων και την πρόληψη της μη εξουσιοδοτημένης πρόσβασης. Το σύστημα διασφαλίζει ότι μόνο τα εξουσιοδοτημένα μέρη μπορούν να προβάλλουν και να αλληλεπιδρούν με τα μηνύματα, εμποδίζοντας παράλληλα τυχόν μη εξουσιοδοτημένα μέρη να έχουν πρόσβαση στις πληροφορίες.

Το Session είναι μια εφαρμογή ανταλλαγής μηνυμάτων που εστιάζει στο απόρρητο και αξιοποιεί τεχνολογίες blockchain και κρυπτονομισμάτων για να παρέχει μια ασφαλή και ανώνυμη πλατφόρμα επικοινωνίας. Το Session είναι

χτισμένο στο Δίκτυο Loki, το οποίο είναι ένα σύστημα επικοινωνίας που βασίζεται στο blockchain που εστιάζει στο απόρρητο. Ένα από τα βασικά χαρακτηριστικά του Session είναι το ανώνυμο πρωτόκολλο επικοινωνίας του. Όταν οι χρήστες στέλνουν μηνύματα μέσω του Session, η ταυτότητά τους αποκρύπτεται μέσω της χρήσης προηγμένης κρυπτογράφησης και δρομολόγησης onion. Αυτό σημαίνει ότι τα μηνύματα κρυπτογραφούνται πολλές φορές και δρομολογούνται μέσω πολλών κόμβων δικτύου, καθιστώντας σχεδόν αδύνατο για κανέναν να εντοπίσει την προέλευση του μηνύματος ή να παρακολουθήσει τον αποστολέα.

Όπως βλέπουμε λοιπόν οι υλοποιήσεις των υπαρκτών εφαρμογών για ανταλλαγή μηνυμάτων εστιάζουν στο απόρρητο και την ανωνυμία που μπορεί να παρέχει το blockchain. Στόχος αυτής της έρευνας είναι η αξιοποίηση της ψηφιακής επικοινωνίας με την τεχνολογία blockchain με την κατεύθυνση της διαφάνειας και της προστασίας σε υπάρχουσες διαδικασίες κάνοντας αυτές πιο απλές, πιο σύντομες και πιο λειτουργικές. Επομένως οι απαιτήσεις δεν καλύπτονται για το ζητούμενο που παρουσιάζεται.

ΚΕΦΑΛΑΙΟ 4

4. Μεθοδολογία και Υλοποίηση

Για την ανάπτυξη της παρουσιαζόμενης εφαρμογής επιλέχθηκε αρχικά το βασικό δίκτυο blockchain για υλοποίηση αποκεντρωμένων εφαρμογών. Διερευνήθηκαν και χρησιμοποιήθηκαν διάφορα εργαλεία και βιβλιοθήκες. Τα βασικά από αυτα για προγραμματισμό έξυπνων συμβάσεων είναι:

- Etherscan
- Metamask
- Remix
- web3.js
- Infura
- Ganache
- Truffle
- Geth

Στην πορεία της υλοποίησης αφού δοκιμάστηκαν αρχικά τα Truffle, Ganache, Remix, Metamask και αλληλεπιδράσεις μεταξύ τους καθώς συνέπεσε με την αναβάθμιση του δικτύου Ethereum κρίθηκε καλή επιλογή το επόμενο συμβατό blockchain. Διερευνήθηκαν τα χαρακτηριστικά, οι μηχανισμοί, οι μηχανισμοί συναίνεσης, η ασφάλεια η απόδοση, η αρχιτεκτονική των δεδομένων block καθώς και η συμβατότητα. Οι αναφορές συμπεριλαμβάνονται στη βιβλιογραφία για το πρωτόκολλο που επιλέχθηκε αλλά η ανάλυση του θα αποτελούσε από μόνη της μια ξεχωριστή εργασία. Ωστόσο, η μελέτη του προτείνεται καθώς παρουσιάζει αρκετά στοιχεία με τεχνολογικό και ακαδημαϊκό ενδιαφέρον.

Η τεχνολογία Blockchain Avalanche (Rocket, 2019) που επιλέχθηκε προτάθηκε για πρώτη φορά το 2018 από μια ομάδα επιστημόνων υπολογιστών από το Πανεπιστήμιο Cornell. Ο στόχος ήταν να δημιουργηθεί ένα νέο πρωτόκολλο blockchain που θα μπορούσε να λύσει τα ζητήματα κλιμάκωσης και ασφάλειας που αντιμετωπίζουν οι γνωστές τεχνολογίες blockchain. Το δίκτυο Avalanche αν και έχει μικρό διάστημα λειτουργίας έχει τη δυνατότητα να εξυπηρετεί περίπου 4.500TPS και έχει δοκιμαστεί σε μεγάλη συμφόρηση αλλά όχι στο χρόνο. Οι ιδρυτές της Avalanche οραματίστηκαν ένα blockchain που θα μπορούσε να υποστηρίξει υψηλή διακίνηση συναλλαγών διατηρώντας παράλληλα ισχυρές εγγυήσεις ασφαλείας. Φαίνεται πως για τη ταχύτητα του πράγματι η απόδοση του σε σχέση με το Ethereum είναι εξαιρετική. Όσο για τα προβλήματα που θα προκύψουν στο μέλλον η πορεία του στο χρόνο θα δείξει τις δυνατότητες του. Συνολικά, ο μοναδικός μηχανισμός συναίνεσης, η αρχιτεκτονική των δεδομένων του Avalanche(Amores et al., 2022) και η υποστήριξη για συμβατότητα με τις έξυπνες συμβάσεις Ethereum το έχουν τοποθετήσει ως ένα πολλά υποσχόμενο νέο πρωτόκολλο blockchain με τη δυνατότητα να είναι μια ακόμα καινοτομία στον αποκεντρωμένο χώρο.

Το Avalanche με τον αλγόριθμο συναίνεσης της χιονοστιβάδας(Avalanche) είναι μια πολλά υποσχόμενη εξέλιξη στην τεχνολογία blockchain που επιτρέπει υψηλή ταχύτητα και ασφαλείς συναλλαγές, καθιστώντας την κατάλληλη για πλαίσια μηνυμάτων με βάση τη τεχνολογία blockchain. Χρησιμοποιεί μια νέα προσέγγιση στη συναίνεση που εξασφαλίζει την επεκτασιμότητα και τη σταθερότητα του δικτύου, ενώ παράλληλα διασφαλίζει ότι οι συναλλαγές επιβεβαιώνονται γρήγορα και με ασφάλεια. Σε σύγκριση με το δίκτυο ethereum που πραγματοποιεί 15-45 συναλλαγές το δευτερόλεπτο το Avalanche πραγματοποιεί περίπου 4500 TPS. Επιπλέον το κόστος συναλλαγής είναι ελάχιστο. Ακόμα και στα δοκιμαστικά δίκτυα χρειάζεται δαπάνη για την αλληλεπίδραση. Με το Avalanche αυτή η διαδικασία είναι σχεδόν στιγμιαία. Επιπλέον το Avalanche χρησιμοποιεί ένα ξεχωριστό και ιδιαίτερο τρόπο για την αναπαράσταση των δεδομένων του(Amores et al.,

2022), το λεγόμενο DAG. Το DAG (Directed Acyclic Graph) είναι μια δομή δεδομένων που μπορεί να χρησιμοποιηθεί για την αναπαράσταση διαφόρων τύπων δεδομένων, συμπεριλαμβανομένων των συναλλαγών blockchain. Στην τεχνολογία blockchain, ένα DAG μπορεί να χρησιμοποιηθεί για την αποθήκευση και την οργάνωση δεδομένων συναλλαγών με πιο αποτελεσματικό και επεκτάσιμο τρόπο από μια γραμμική αλυσίδα. Το Avalanche χρησιμοποιεί ένα DAG για την αποθήκευση και την οργάνωση δεδομένων συναλλαγών, επιτρέποντας ταχύτερη και αποτελεσματικότερη επεξεργασία των συναλλαγών.

Συνοψίζοντας καθώς δεν επηρεάστηκε η διαδικασία υλοποίησης, χρησιμοποιήθηκε το περιβάλλον Remix για τον προγραμματισμό του έξυπνου συμβολαίου σε συνδυασμό με τον πάροχο Metamask και επιλέχθηκε το Avalanche σύμφωνα με τα κριτήρια που αναφέρθηκαν. Η ανάπτυξη του συμβολαίου δοκιμάστηκε με το Metamask όπου στο Metamask προστέθηκε το δοκιμαστικό δίκτυο Avalanche-Fuji-C-Chain. Στη συνέχεια ο κώδικας που παράχθηκε (abi) καθώς και η διεύθυνση του συμβολαίου που έγινε deploy χρησιμοποιήθηκαν στα αντίστοιχα αρχεία της εφαρμογής για τη σύνδεση με το front-end. Αυτή η διαδικασία μπορεί να γίνει με οποιοδήποτε άλλη τεχνολογία blockchain είναι συμβατή με Solidity και υποστηρίζει EVM.

4.1 Επισκόπηση της Έξυπνης Σύμβασης

Το έξυπνο συμβόλαιο ορίζει ένα απλό σύστημα ανταλλαγής μηνυμάτων με δυνατότητα εγγραφής χρήστη, προσθήκης επαφής και δυνατότητα αποστολής μηνυμάτων. Επιπλέον έχει οριστεί η δυνατότητα για προσθήκη “Authority” όπου θα μπορεί κάποια διεύθυνση να δει μια συνομιλία μεταξύ δύο χρηστών. Το συμβόλαιο είναι γραμμένο σε γλώσσα προγραμματισμού Solidity στο περιβάλλον Remix και χρησιμοποιεί τη βιβλιοθήκη συμβάσεων του OpenZeppelin για βοηθητικά προγράμματα όπως λειτουργίες συμβολοσειρών.

Για την αποστολή μηνύματος απαιτείται αρχικά να είναι εγγεγραμμένοι οι χρήστες στην εφαρμογή. Πρέπει δηλαδή να έχουν συνδεθεί με το πορτοφόλι τους. Το ίδιο ισχύει και για την επαφή που θέλουν να στείλουν μήνυμα.

Με το “require” στον κώδικα δηλώνεται στη solidity ο κανόνας που περιορίζει τους χρήστες στους κανόνες της έξυπνης σύμβασης. Η εγγραφή ουσιαστικά δεν είναι παραδοσιακή όπως είναι σε μια πλατφόρμα ηλεκτρονικής αλληλογραφίας ή μιας εφαρμογής μηνυμάτων αλλά είναι η σύνδεση με τη ταυτότητα του χρήστη που στο blockchain είναι η διεύθυνση κατακερματισμού του ηλεκτρονικού του πορτοφολιού.

Παράδειγμα κώδικα:

```
// Sends a new message to a given friend
    function sendMessage(address friend_key, string calldata _msg) external {
      require(checkUserExists(msg.sender), "Create an account first!");
      require(checkUserExists(friend_key), "User is not registered!");
      require(checkAlreadyFriends(msg.sender,friend_key), "You are not friends
with the given user");
      bytes32 chatCode = _getChatCode(msg.sender, friend_key);
      message memory newMsg = message(msg.sender, block.timestamp, _msg);
      allMessages[chatCode].push(newMsg);
    }
```

Το παράδειγμα είναι η υλοποίηση της συνάρτησης αποστολής μηνύματος. Για να πραγματοποιηθεί η αποστολή μηνύματος το συμβόλαιο εξετάζει:

- αν ο χρήστης που πάει να στείλει μήνυμα υπάρχει στο blockchain και έχει οριστεί από το συμβόλαιο
- αν ο χρήστης που θα παραλάβει το μήνυμα υπάρχει στο blockchain και έχει οριστεί από το συμβόλαιο
- αν είναι επαφές μεταξύ τους

Ο πίνακας 4.1 περιέχει τις βασικές συναρτήσεις με μια σύντομη περιγραφή.

Πίνακας 4.1: Συναρτήσεις και περιγραφή τους

Συνάρτηση	Περιγραφή
`function checkUserExists(address pubkey) public view returns(bool)`	Ελέγχει από χάρτη χρηστών αν υπάρχει ο χρήστης. Επιστρέφει αληθές ή ψευδές. Απαραίτητο για κανόνα.
`function _addFriend(address me, address friend_key, string memory name) internal`	Εσωτερική. Απαραίτητη προϋπόθεση για κανόνα.
`function addFriend(address friend_key, string calldata name) external`	Εξωτερική. Καλείται από την εφαρμογή για τη καταχώρηση του χρήστη.
`function checkAlreadyFriends(address pubkey1, address pubkey2) internal view returns(bool)`	Απαραίτητη για κανόνα. Κάθε κωδικός συνομιλίας είναι μοναδικός.
`function _getChatCode(address pubkey1, address pubkey2) internal pure returns(bytes32)`	Η πιο σημαντική συνάρτηση της λειτουργίας της εφαρμογής. Επιστρέφει τον μοναδικό κωδικό συνομιλίας από τις διευθύνσεις των δύο χρηστών σε μορφή 32bytes.
`function sendMessage(address friend_key, string calldata _msg) external`	Αποστολή του μηνύματος. Δέχεται τη διεύθυνση του παραλήπτη και το περιεχόμενο. Καλεί κανόνες για να εκτελεστεί. Το “timestamp” ορίζεται εσωτερικά από το block μόλις εγκριθεί η αποστολή
`function readMessage(address friend_key) external view returns(message[] memory)`	Επιστρέφει ολόκληρη τη συνομιλία. Αυτό δεν αποτελεί πρόβλημα για τέλη εξόδων καθώς η κλήση διαβάσματος δεδομένων δεν απαιτεί χρέωση στο blockchain
`function authReadMessage(address personA, address personB) external view returns(message[] memory)`	Διάβασμα συνομιλία από “Αρχή”. Απαιτεί δικαιώματα.

Οι κανόνες που υλοποιεί το συμβόλαιο:

#Κανόνας 1: Οι χρήστες που χρησιμοποιούν την εφαρμογή πρέπει να έχουν συνδεθεί μια φορά σε αυτήν. Με αυτό το τρόπο αποδέχονται τους κανόνες της συμφωνίας.

#Κανόνας 2: Για να στείλει μήνυμα ο ένας χρήστης σε έναν άλλο είναι απαραίτητη προϋπόθεση να έχει προσθέσει ο ένας τον άλλο. Με αυτό τον τρόπο αποφεύγεται η ανεπιθύμητη λήψη μηνυμάτων.

#Κανόνας 3: μόνο ο αποστολέας που αλληλεπιδρα μπορεί να προσθέσει την επαφή του. Η προσθήκη του ίδιου ως επαφή επιτρέπεται από το συμβόλαιο.

#Κανόνας 4: τα μηνύματα μπορούν να διαβαστούν μόνο από τον αποστολέα τον παραλήπτη, τον δημιουργό του έξυπνου συμβολαίου ή τις “Αρχές.

#Κανόνας 5: ο δημιουργός του συμβολαίου μπορεί να δει ποιές διευθύνσεις είναι αρχές.

Άλλο ένα βασικό σημείου του συμβολαίου είναι ο ορισμός των μηνυμάτων. Τα μηνύματα κωδικοποιούνται με τη μορφή συνομιλίας/αλληλεπίδρασης χρηστών. Έστω ότι είναι δύο χρήστες.

A:0x30783c7f290DF8d161DA996faedD8C793dDef092
B:0x12345c7f290DF8d161DA996faedD8C793dDef092

Το αποτέλεσμα για κάθε συνομιλία είναι μια μοναδική ταυτότητα όπως είναι αυτή των χρηστών. Αντίστροφα και η αποκωδικοποίηση της συνομιλίας γίνεται με αυτό τον τρόπο. Η συνάρτηση ορίζεται ως pure καθώς δεν πρέπει να αλλάζει η κατάσταση των δεδομένων. Επιπλέον έχει οριστεί ως εσωτερική που σημαίνει ότι δεν μπορεί να καλεστεί εκτός του συμβολαίου. Η καταγραφή στο

block της συνομιλίας μπορεί να κωδικοποιηθεί και να αποκωδικοποιηθεί μόνο από το έξυπνο συμβόλαιο.

```
bytes32 chatCode = _getChatCode(msg.sender, friend_key);//call
```

//method

```
function _getChatCode(address pubkey1, address pubkey2) internal pure
returns(bytes32) {
        if(pubkey1 < pubkey2)
            return keccak256(abi.encodePacked(pubkey1, pubkey2));
        else
            return keccak256(abi.encodePacked(pubkey2, pubkey1));
 }
```

Το έξυπνο συμβόλαιο για την ανάπτυξη του χρησιμοποιεί ένα αρχείο script:

deploy_with_web3.ts

```
import { deploy } from './web3-lib'

(async () => {
    try {
        const result = await deploy('MessengerPof', [])
        console.log(`address: ${result.address}`)
    } catch (e) {
        console.log(e.message)
    }
})()
```

Για την ανάπτυξη μπορούν να χρησιμοποιηθούν διαφορετικοί πάροχοι.

- Injected Provider - MetaMask
- Dev - Ganache Provider

Για την δοκιμή με λογαριασμούς avalanche επιλέχθηκε ο injected provider Metamask. Έτσι με το deploy εμφανίζεται το μήνυμα για να πληρωθεί το κόστος συναλλαγής και να πληρωθεί στο δίκτυο.

Δημιουργήθηκαν πολλαπλοί λογαριασμοί όπου έγινε εισαγωγή δοκιμαστικών coins. Το συμβόλαιο βρίσκεται στη διεύθυνση:

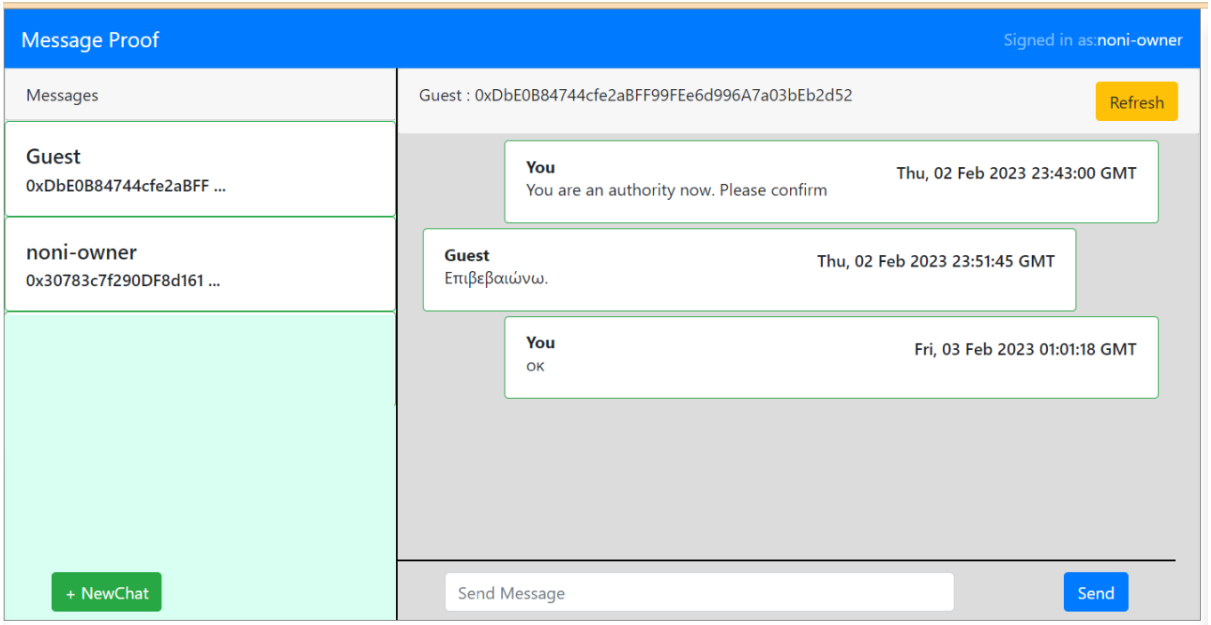
https://testnet.snowtrace.io/address/0x5CE77A9B70bB0089D6e1b9204aca05F85Ae861B9

4.2 Η Ασφαλής Εφαρμογή Επικοινωνίας

Για την εφαρμογή χρησιμοποιήθηκαν:

- Metamask - Ο πάροχος web3 για τη δημιουργία πορτοφολιών.
- Node.js
- ├── bootstrap@4.6.2
- ├── ethers@5.7.2
- ├── gh-pages@3.2.3
- ├── react-bootstrap@1.6.6
- ├── react-dom@17.0.2
- ├── react-scripts@4.0.3
- ├── react@17.0.2
- └── web-vitals@1.1.2

Η εφαρμογή που υλοποιήθηκε είναι λειτουργική. Είναι μια πρότυπη διεπαφή για την χρήση του smart contract και την επίδειξη των δυνατοτήτων της ανταλλαγής μηνυμάτων.



Εικόνα 4.2: η διεπαφή χρήστη που επικοινωνεί με το έξυπνο συμβόλαιο.

Οι λειτουργίες της εφαρμογής είναι:

- Σύνδεση της εφαρμογής μέσω κουμπιού με τον πάροχο-πορτοφόλι Metamask "connect with metamask"
- Για καινούργιο χρήστη με τη σύνδεση ζητείται username και προτείνετε το "Guest".
- Στην περίπτωση που ο χρήστης έχει ξανά συνδεθεί εμφανίζονται όλες οι προηγούμενες συνομιλίες του στο αριστερό παράθυρο της διεπαφής. Οι συνομιλίες είναι επιλέξιμες για προβολή στο κεντρικό παράθυρο πού εμφανίζεται ολόκληρη η συνομιλία.
- Ο χρήστης μέσα από τη διεπαφή μπορεί να προσθέσει νέο φίλο "New Chat".
- Να στείλει μήνυμα επιλέγοντας από τις συνομιλίες που βρίσκονται αριστερά. Τόσο οι κενές όσο και οι παλιές εμφανίζονται στο ίδιο μέρος.

Περισσότερα στιγμιότυπα βρίσκονται στο “ΠΑΡΑΡΤΗΜΑ ΙΙ”.

Τα δεδομένα της εφαρμογής μεταφέρονται προβλεπόμενα. Η Γραφική διεπαφή δεν αποτελεί στόχο της υλοποίησης αλλά χρησιμοποιεί εργαλεία και βιβλιοθήκες που αντικατοπτρίζουν την ανάπτυξη πραγματικών εφαρμογών. Δεν έχουν διαχειριστεί όλα τα σημεία σφάλματος

Για την επικοινωνία με το smart contract έγινε χρήση του ether.js
Η εφαρμογή είναι φτιαγμένη με React.

4.3 Επισκόπηση Σημείων Επικοινωνίας με το Έξυπνο Συμβόλαιο

Η βιβλιοθήκη που χρησιμοποιούνται για την πραγματοποίηση της σύνδεσης της διεπαφής με το δίκτυο που βρίσκεται το έξυπνο συμβόλαιο είναι η βιβλιοθήκη Javascript ethers.js Moore(2021).

```
import { ethers } from "ethers";
```

Άλλο ένα βασικό δομικό στοιχεία που απαιτείται για τη σύνδεση είναι το αρχείο “abi” Το ABI είναι η διεπαφή που ορίζει το τυπικό σχήμα για τον τρόπο κλήσης συναρτήσεων στο έξυπνο συμβόλαιο και για την ανάκτηση δεδομένων.

```
import { abi } from "./abi";
```

Η διεύθυνση στο δίκτυο που είναι το έξυπνο συμβόλαιο είναι ένα ακόμα σημαντικό δεδομένο που χρειάζεται για την υλοποίηση της εφαρμογής.

```
const CONTRACT_ADDRESS ="0x5CE77A9B70bB0089D6e1b9204aca05F85Ae861B9"
 const contractABI = abi;
```

Ο πάροχος επικοινωνίας όπως έχει οριστεί από την πρόταση βελτίωσης από το EIP-1193.

```
provider = new ethers.providers.Web3Provider( window.ethereum );
```

Δήλωση του υπογραφέα signer,το χρήστη δηλαδή που αλληλεπιδρά με το συμβόλαιο στο blockchain.

```
signer = provider.getSigner();
```

Η σταθερά contract χρησιμοποιείται πλέον από την εφαρμογή για την αλληλεπίδραση στο συμβόλαιο. Με μια εντολή χρησιμοποιείται η βιβλιοθήκη ethers.js που συνδέει τη διεύθυνση της σύμβασης, τον δυαδικό κώδικα της εφαρμογής και τη μεταβλητή που περιέχει το λογαριασμό-ταυτότητα σύνδεσης του χρήστη.

```
const contract = new ethers.Contract( CONTRACT_ADDRESS, contractABI, signer );
```

Το όρισμα της εντολής:

```
(alias) new Contract(addressOrName: string, contractInterface:
ethers.ContractInterface, signerOrProvider?: ethers.providers.Provider |
ethers.Signer | undefined): ethers.Contract
export Contract
```

αφού έχει οριστεί το συμβόλαιο πλέον ως μια σταθερά μπορούν να καλεστουν οι δημόσιες μεταβλητές και συναρτήσεις του συμβολαίου.

```
let present = await contract.checkUserExists( address );
```

Όπως φαίνεται στον παρακάτω κώδικα, μόνο το κείμενο και ο παραλήπτης αποστέλλεται στο συμβόλαιο. Ο χρόνος αποστολής ορίζεται από το block με την εκτέλεση της μεθόδου του συμβολαίου. Αν μια συνθήκη δεν ικανοποιείται δε θα εκτελεστεί η μέθοδος και δεν θα καταχωρηθεί το μπλοκ.

```
await myContract.sendMessage( recieverAddress, data );
```

5. Συμπέρασμα και Μελλοντικοί Στόχοι

Οι αναρίθμητες πρακτικές εφαρμογές που έχουν υλοποιηθεί(Navadkar et al.,2018), οι βιβλιογραφικές και δημοσιευμένες έρευνες, η όλο και αυξανόμενη υιοθέτηση της τεχνολογία blockchain σε πολλές χώρες(Jun, 2018) καταδεικνύουν ότι η τεχνολογία blockchain μπορεί να αποτελέσει μια πραγματική επαναστατική τεχνολογική εξέλιξη με την κατάλληλη αξιοποίηση της.

Οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain έχουν τη δυνατότητα να μεταμορφώσουν τον τρόπο με τον οποίο διαχειριζόμαστε και ασφαλίζουμε πληροφορίες, συναλλαγές και διαδικασίες. Παρέχοντας δυνατότητες μη απόρριψης και χρονικής σφράγισης, οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain μπορούν να βελτιώσουν την ακρίβεια και την ακεραιότητα στην αλληλεπίδραση της επικοινωνίας μεταξύ των μερών, να αποτρέψουν απάτες και διαφορές, να αυξήσουν την εμπιστοσύνη μεταξύ των μερών και να βελτιώσουν τη συμμόρφωση με τους κανονισμούς.

Συνοπτικά, οι εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain έχουν τη δυνατότητα να παρέχουν μεγαλύτερη διαφάνεια, ασφάλεια και αποτελεσματικότητα σε ένα ευρύ φάσμα τομέων, συμπεριλαμβανομένων των οικονομικών, της εφοδιαστικής αλυσίδας, του δημόσιου τομέα, της εκπαίδευσης, της ιατρικής, του νόμου, του ηλεκτρονικού εμπορίου και πολλών άλλων. Καθώς η τεχνολογία blockchain συνεχίζει να εξελίσσεται και να υιοθετείται ευρύτερα, μπορούμε να περιμένουμε να δούμε περαιτέρω καινοτομίες και νέες περιπτώσεις χρήσης για εφαρμογές ανταλλαγής μηνυμάτων που βασίζονται σε blockchain στο μέλλον.

Ενώ η έρευνα παρέχει πολύτιμες γνώσεις σχετικά με τη χρήση της τεχνολογίας blockchain στην ασφαλή ανταλλαγή μηνυμάτων, υπάρχει ακόμη πολύς χώρος για περαιτέρω έρευνα. Ένας τομέας για μελλοντική έρευνα θα μπορούσε να είναι η ενσωμάτωση πρόσθετων λειτουργιών σε πλαίσια ανταλλαγής μηνυμάτων που βασίζονται σε blockchain, όπως η ασφαλής κοινή χρήση αρχείων. Η μέθοδος που μπορεί να συμβεί είναι με τη χρήση μιας τεχνολογίας όπως είναι αυτή του IPFS. Το IPFS (Inter Planetary File System)είναι ένα πρωτόκολλο και ένα δίκτυο σχεδιασμένο να δημιουργεί ένα κατανεμημένο σύστημα αρχείων με βάση τη τεχνολογία Blockchain.

Επιπλέον, υπάρχει ανάγκη για έρευνα σχετικά με την επεκτασιμότητα των λύσεων ανταλλαγής μηνυμάτων που βασίζονται σε blockchain. Καθώς περισσότεροι χρήστες υιοθετούν αυτές τις τεχνολογίες, θα είναι σημαντικό να διασφαλιστεί ότι μπορούν να χειριστούν μεγάλους όγκους κίνησης χωρίς συμβιβασμούς στην ασφάλεια και την απόδοση. Η έρευνα θα μπορούσε να εστιαστεί και στους κανόνες της έξυπνης σύμβασης. Προτάθηκει ένα πλαίσιο μηνυμάτων που έχουν προσδιοριστεί κάποιοι κανόνες. Οι κανόνες αυτοί ανταποκρίνονται στις απαιτήσεις παροχής χρονοσήμανσης και μη αποποίησης περιεχομένου, αποστολέα, παραλήπτη. Επιπλέον έχουν υλοποιηθεί κανόνες για να εκχωρούνται δικαιώματα σε αρχές επίβλεψης. Για τους κανόνες επίβλεψης κρίνεται απαραίτητο να διερευνηθούν ερωτήματα που ξεπερνούν την τεχνολογική σκοπιά. Η πρόταση που παρουσιάστηκε για την υλοποίηση της έξυπνης σύμβασης είναι μια καλή βάση αλλά κρίνεται πως υπάρχει ακόμη πολύς χώρος για περαιτέρω έρευνα.

ΠΑΡΑΡΤΗΜΑ Ι

Στο παράρτημα αυτό παρατίθενται οι δοκιμαστικοί λογαριασμοί

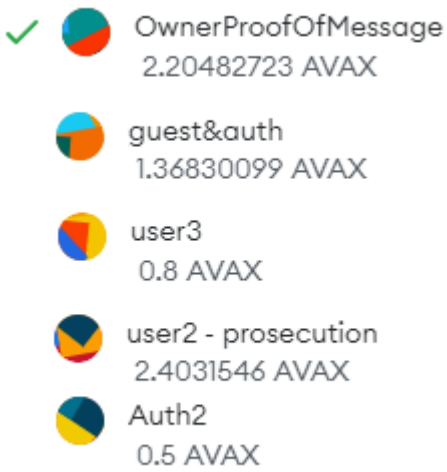
Account1-owner noni
0x30783c7f290DF8d161DA996faedD8C793dDef092

Account2 user - guest &auth
0xDbE0B84744cfe2aBFF99FEe6d996A7a03bEb2d52

Account3 user2 - prosecution
0x5Ff85C7b9F377d902C6d30EeddD7D56DD0268E0c

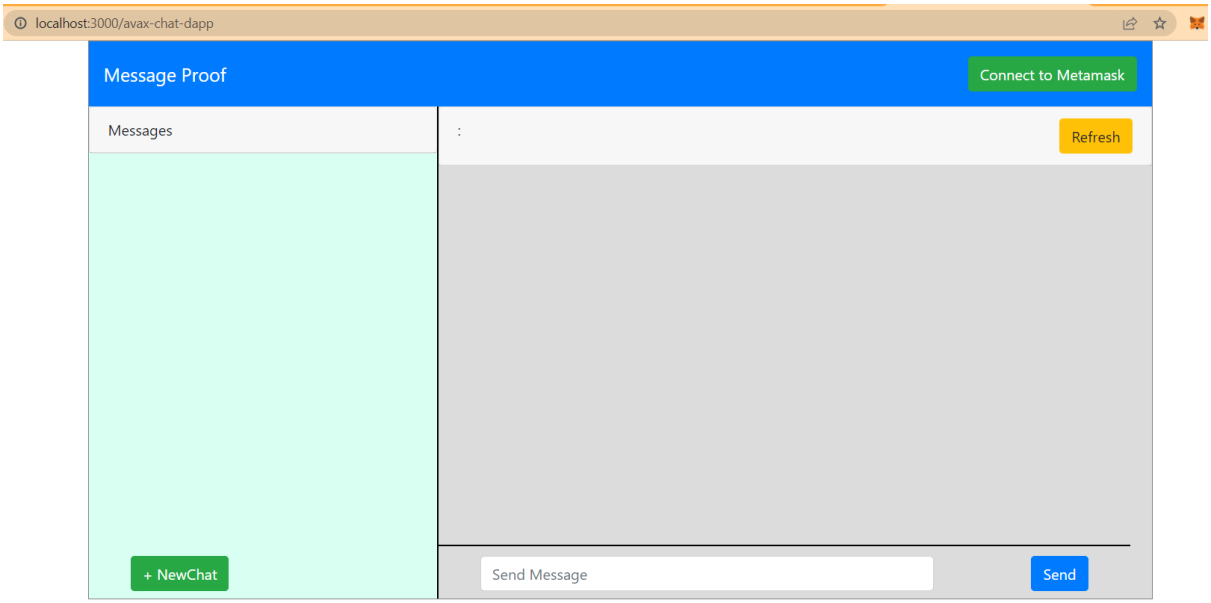
Account 4 - user3
0x785097386E5D6d41AF2Baf46cA36Ad69dD542007

Account 5-authority2
0x8521573fb8c84887c04f7d0818a1beCf4bdbA96b

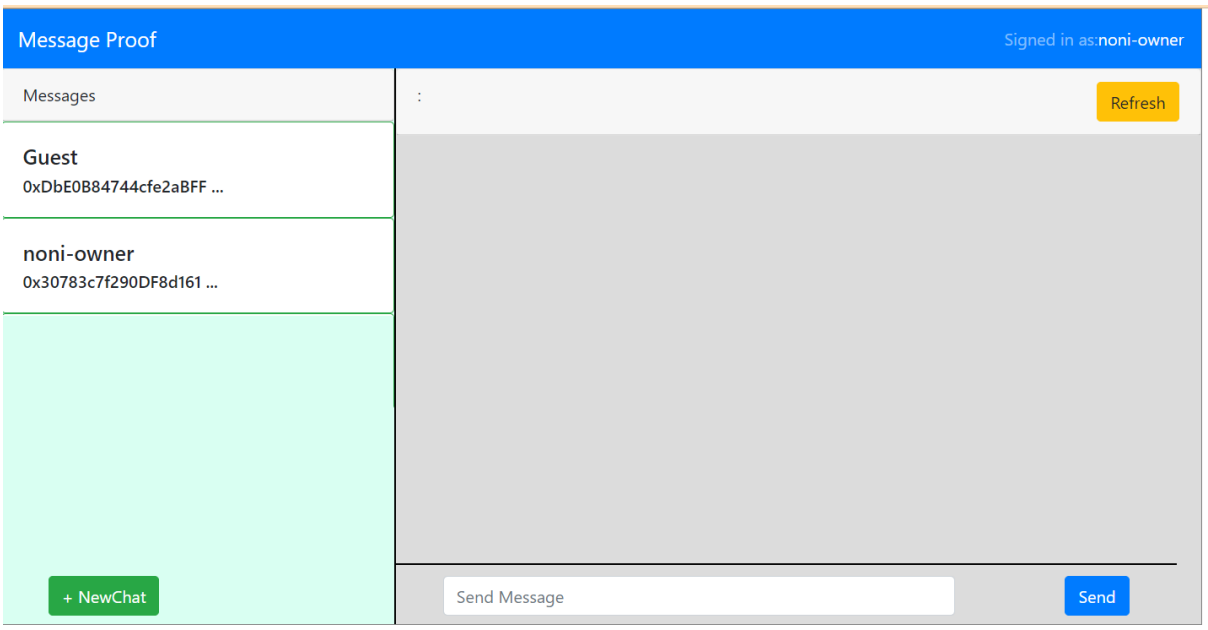


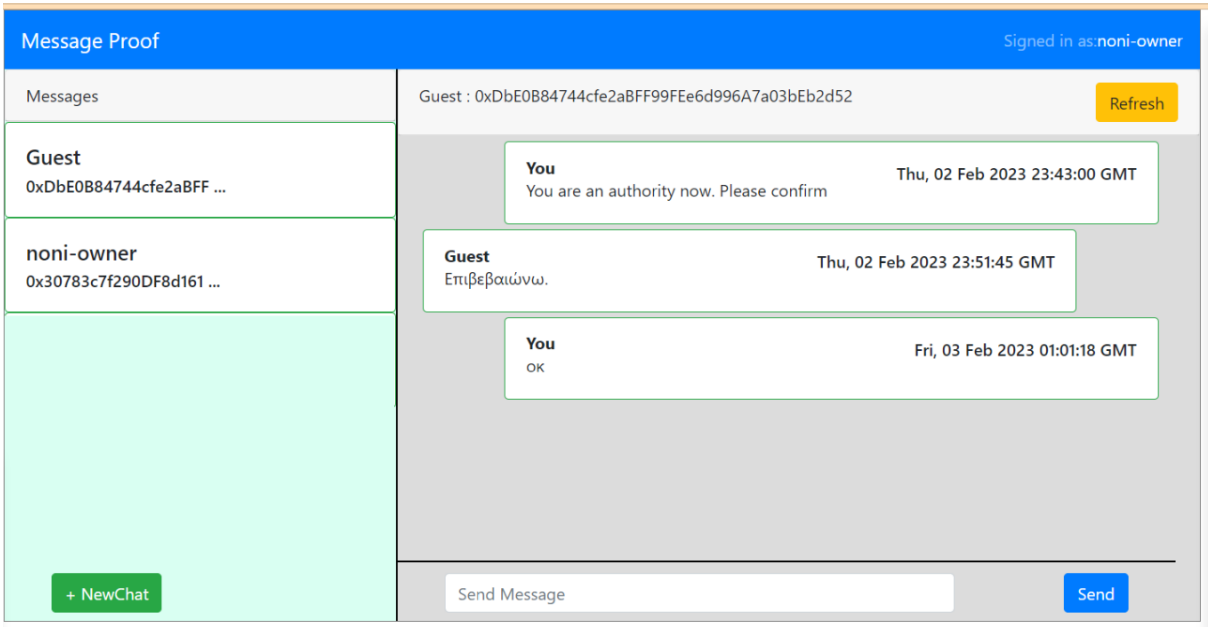
ΠΑΡΑΡΤΗΜΑ ΙΙ

Στο παράρτημα αυτό παρατίθενται στιγμιότυπα της εφαρμογής.

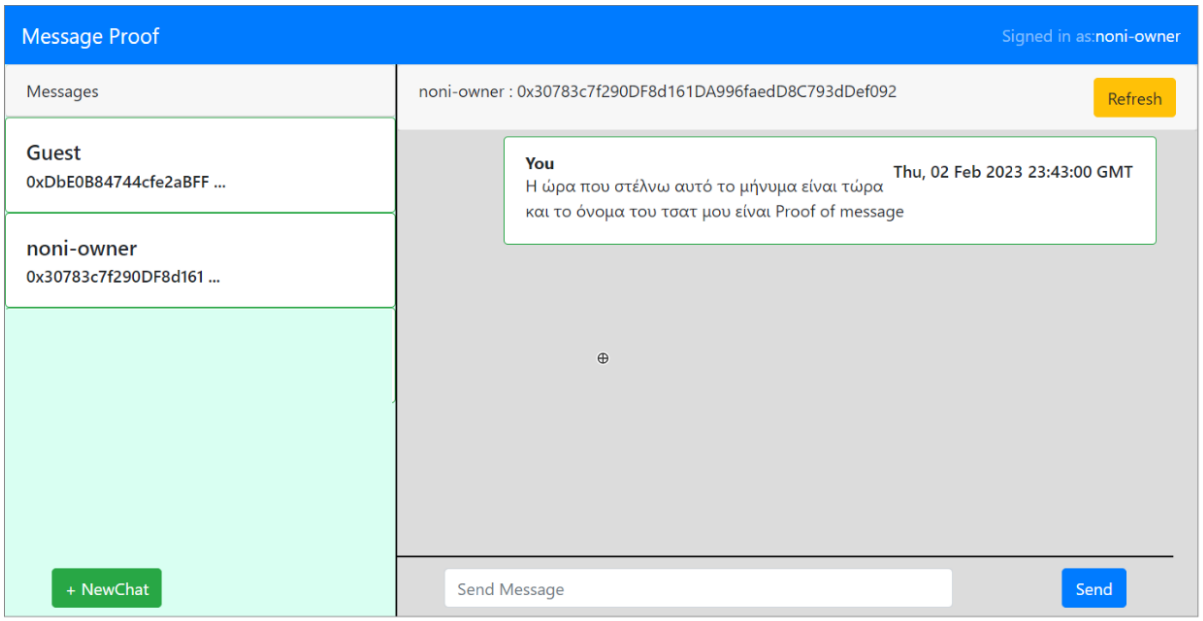


Η αρχική εφαρμογή χωρίς είσοδο.





συνομιλία με τον λογαριασμό δύο που το έχουν καταχωρηθεί δικαιώματα “authority” από το περιβάλλον remix.



Προβολή συνομιλίας από τον χρήστη προς τον ίδιο χρήστη ως μέθοδος διασφάλισης πνευματικής ιδιοκτησίας ονόματος με χρονοσφραγίδα.

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Ο κώδικας Solidity για το έξυπνο συμβόλαιο

```
pragma solidity >=0.7.0 <0.9.0;
import "@openzeppelin/contracts/utils/Strings.sol";
import "hardhat/console.sol";
/**
 * @title Database
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script ./scripts/deploy_with_web3.ts
 */
contract Database {

uint authcount=0;
address public Owner; //contract owner/public for test reasons
only
//ts's not efficient for gas ->better map
address[] public authorities;


    struct user {
        string name;
        friend[] friendList;
    }


    // Each friend is identified by its address and name assigned
by the second party
    struct friend {
        address pubkey;
        string name;
    }


    // message construct stores the single chat message and its
metadata
    struct message {
        address sender;
        uint256 timestamp;
        string msg;
    }


    // Collection of users registered on the application
```

```
    mapping(address => user) userList;
    // Collection of messages communicated in a channel between
two users
    mapping(bytes32 => message[]) allMessages; // key :
Hash(user1,user2)


        constructor() {
        //owner of contract
        Owner = msg.sender;
        //test conole.log
         console.log("set proof of message owner contract:",
            msg.sender    );
    }

    // It checks whether a user(identified by its public key)
    // has created an account on this application or not
    function checkUserExists(address pubkey) public view
returns(bool) {
        return bytes(userList[pubkey].name).length > 0;
    }


    // Registers the caller(msg.sender) to our app with a
non-empty username
    function createAccount(string calldata name) external {
        require(checkUserExists(msg.sender)==false, "User already
exists!");
        require(bytes(name).length>0, "Username cannot be
empty!");
        userList[msg.sender].name = name;
    }


// Returns the default name provided by an user
    function getUsername(address pubkey) external view
returns(string memory) {
        require(checkUserExists(pubkey), "User is not
registered!");
        return userList[pubkey].name;
    }

     // A helper function to update the friendList
    function _addFriend(address me, address friend_key, string
memory name) internal {
        friend memory newFriend = friend(friend_key,name);
        userList[me].friendList.push(newFriend);
    }
```

```
    // Adds new user as your friend with an associated nickname
    function addFriend(address friend_key, string calldata name)
external {
        require(checkUserExists(msg.sender), "Create an account
first!");
        require(checkUserExists(friend_key), "User is not
registered!");
     //   require(msg.sender!=friend_key, "Users cannot add
themselves as friends!"); oh yes they can
        require(checkAlreadyFriends(msg.sender,friend_key)==false,
"These users are already friends!");
        _addFriend(msg.sender, friend_key, name);
        _addFriend(friend_key, msg.sender,
userList[msg.sender].name);
    }


    // Checks if two users are already friends or not
    function checkAlreadyFriends(address pubkey1, address pubkey2)
internal view returns(bool) {
        if(userList[pubkey1].friendList.length >
userList[pubkey2].friendList.length)
        {
            address tmp = pubkey1;
            pubkey1 = pubkey2;
            pubkey2 = tmp;
        }
        for(uint i=0; i<userList[pubkey1].friendList.length; ++i)
        {
            if(userList[pubkey1].friendList[i].pubkey == pubkey2)
                return true;
        }
        return false;
    }


    // Returns list of friends of the sender
    function getMyFriendList() external view returns(friend[]
memory) {
        return userList[msg.sender].friendList;
    }


    // Returns a unique code for the channel created between the
two users
```

```
    // Hash(key1,key2) where key1 is lexicographically smaller
than key2
    function _getChatCode(address pubkey1, address pubkey2)
internal pure returns(bytes32) {
        if(pubkey1 < pubkey2)
            return keccak256(abi.encodePacked(pubkey1, pubkey2));
        else
            return keccak256(abi.encodePacked(pubkey2, pubkey1));
    }


    // Sends a new message to a given friend
    function sendMessage(address friend_key, string calldata _msg)
external {
        require(checkUserExists(msg.sender), "Create an account
first!");
        require(checkUserExists(friend_key), "User is not
registered!");
        require(checkAlreadyFriends(msg.sender,friend_key), "You
are not friends with the given user");
        bytes32 chatCode = _getChatCode(msg.sender, friend_key);
        message memory newMsg = message(msg.sender,
block.timestamp, _msg);
        allMessages[chatCode].push(newMsg);
    }


    // Returns all the chat messages communicated in a channel
    function readMessage(address friend_key) external view
returns(message[] memory) {
        bytes32 chatCode = _getChatCode(msg.sender, friend_key);
        return allMessages[chatCode];
    }

    //--------------------AUTHORITIES----------------------
 function checkAlreadyAuth(address pubkey) public view
returns(bool) {
     for (uint i; i < authorities.length; i++) {
        if (authorities[i] == pubkey) {
            // corresponding item found - update quantity and
early return
            return true;
        }
    }
        return false;
    }
```

```
  function addAuthority(address givenaddress) external {
        require(msg.sender==Owner, "YOU HAVE NOT PERMISSION");
        require(checkAlreadyAuth(givenaddress)==false, "Already
Authority");
        if (!checkUserExists(givenaddress))
            userList[givenaddress].name ="authority";
        authorities.push(givenaddress);
        authcount++;

    }


   function authReadMessage(address personA, address personB)
external view returns(message[] memory) {
        require(CanRead(msg.sender), "PERMISSION DENIED");
        bytes32 chatCode = _getChatCode(personA, personB);
        return allMessages[chatCode];
   }

    function getAuthorities() external view returns(address[]
memory) {
        //only for testin or another contract
        require(msg.sender==Owner, "YOU HAVE NOT PERMISSION");
        return authorities;
    }


//internal
      function CanRead (address WHO) internal view returns (bool){
        if (WHO==Owner)
            return true;
        for (uint i = 0; i < authorities.length; i++ ){
            if (WHO == authorities[i])
                return true;
        }
        return false;
    }
}
```

ΠΑΡΑΡΤΗΜΑ IV

Ο κώδικας React του βασικού αρχείου React App.jsx

```
import React from "react";
import { useState, useEffect } from "react";
import { Container, Row, Col, Card, Form, Button } from
'react-bootstrap';
import { NavBar, ChatCard, Message, AddNewChat } from
'./components/Components.js';
import { ethers } from "ethers";
import { abi } from "./abi";

// Add the contract address inside the quotes
const CONTRACT_ADDRESS
="0x5CE77A9B70bB0089D6e1b9204aca05F85Ae861B9"

export function App( props ) {
    const [friends, setFriends] = useState(null);
    const [myName, setMyName] = useState(null);
    const [myPublicKey, setMyPublicKey] = useState(null);
    const [activeChat, setActiveChat] = useState({ friendname:
null, publicKey: null });
    const [activeChatMessages, setActiveChatMessages] =
useState(null);
    const [showConnectButton, setShowConnectButton] =
useState("block");
    const [myContract, setMyContract] = useState(null);

    // Save the contents of abi in a variable
    const contractABI = abi;
    let provider;
    let signer;

    // Login to Metamask and check the if the user exists else
creates one
    async function login() {
        let res = await connectToMetamask();
        try {
        if( res === true ) {
            console.log("login function")
            provider = new ethers.providers.Web3Provider(
window.ethereum );
            signer = provider.getSigner();
            try {
```

```
                    //αποθήκευση του contract για χρήση δεδομένων
του blockchain
                    const contract = new ethers.Contract(
CONTRACT_ADDRESS, contractABI, signer );
                    alert("?CONTRACT_ "+contract.name);
               setMyContract( contract );
                    const address = await signer.getAddress();
               alert("?CONTRACT_adress "+address);
                    let present = await contract.checkUserExists(
address );
                    let username;
                    if( present )
                         username = await contract.getUsername(
address );
                    else {
                         username = prompt('Enter a username',
'Guest');
                         if( username === '' ) username =
'Guest';
                         await contract.createAccount( username
);
                    }
                    setMyName( username );
                    setMyPublicKey( address );
                    setShowConnectButton( "none" );
               } catch(err) {
               console.log(err);
                    alert("?CONTRACT_ADDRESS not set
properly!"+err);
               }
        } else {
            alert("Couldn't connect to Metamask");
        }
        } catch(err) {
            console.log(err)
        alert("general error!");
        }
    }

    // Check if the Metamask connects
    async function connectToMetamask() {
        try {
            console.log("connection call");
            await window.ethereum.enable();
            return true;
        } catch(err) {
            console.log(err);
            return false;
```

```
        }
    }

    // Add a friend to the users' Friends List
    async function addChat( name, publicKey ) {
        try {
                let present = await myContract.checkUserExists(
publicKey );
                if( !present ) {
                     alert("Given address not found: Ask him to
join the app :)");
                     return;
                }
                try {
                     await myContract.addFriend( publicKey, name
);
                     const frnd = { "name": name, "publicKey":
publicKey };
                     setFriends( friends.concat(frnd) );
                } catch(err) {
                     alert("Friend already Added! You can't be
friend with the same person twice ;P");
                }
           } catch(err) {
                alert("Invalid address!")
           }
    }

    // Sends messsage to an user
    async function sendMessage( data ) {
        if( !( activeChat && activeChat.publicKey ) ) return;
        const recieverAddress = activeChat.publicKey;
        await myContract.sendMessage( recieverAddress, data );
    }

    // Fetch chat messages with a friend
    async function getMessage( friendsPublicKey ) {
        let nickname;
        let messages = [];
        friends.forEach( ( item ) => {
            if( item.publicKey === friendsPublicKey )
                nickname = item.name;
        });
        // Get messages
        const data = await myContract.readMessage(
friendsPublicKey );
        data.forEach( ( item ) => {
```

```
            const timestamp = new Date( 1000*item[1].toNumber()
).toUTCString();
            messages.push({ "publicKey": item[0], "timeStamp":
timestamp, "data": item[2] });
        });
        setActiveChat({ friendname: nickname, publicKey:
friendsPublicKey });
        setActiveChatMessages( messages );
    }

    // This executes every time page renders and when myPublicKey
or myContract changes
    useEffect( () => {
        async function loadFriends() {
            let friendList = [];
            // Get Friends
            try {
                const data = await myContract.getMyFriendList();
                data.forEach( ( item ) => {
                friendList.push({ "publicKey": item[0], "name":
item[1] });
                })
            } catch(err) {
                friendList = null;
            }
            setFriends( friendList );
        }
        loadFriends();
    }, [myPublicKey, myContract]);

    // Makes Cards for each Message
    const Messages = activeChatMessages ? activeChatMessages.map(
( message ) => {
        let margin = "5%";
        let sender = activeChat.friendname;
        if( message.publicKey === myPublicKey ) {
            margin = "15%";
            sender = "You";
        }
        return (
            <Message marginLeft={ margin } sender={ sender }
data={ message.data } timeStamp={ message.timeStamp } />
        );
    }) : null;

    // Displays each card
    const chats = friends ? friends.map( ( friend ) => {
     return (
```

```
         <ChatCard publicKey={ friend.publicKey } name={
friend.name } getMessages={ ( key ) => getMessage( key ) } />
     );
    }) : null;

    return (
        <Container style={{ padding:"0px", border:"1px solid grey"
}}>
            {/* This shows the navbar with connect button */}
            <NavBar username={ myName } login={ async () =>
login() } showButton={ showConnectButton } />
            <Row>
                {/* Here the friends list is shown */}
                <Col style={{ "paddingRight":"0px",
"borderRight":"2px solid #000000" }}>
                    <div style={{ "backgroundColor":"#D9FFF2",
"height":"100%", overflowY:"auto" }}>
                        <Row style={{ marginRight:"0px" }}  >
                            <Card style={{ width:'100%',
alignSelf:'center', marginLeft:"15px" }}>
                              <Card.Header>
                                  Messages
                              </Card.Header>
                            </Card>
                        </Row>
                        { chats }
                        <AddNewChat myContract={ myContract }
addHandler={ ( name, publicKey ) => addChat( name, publicKey )} />
                    </div>
                </Col>
                <Col xs={ 8 } style={{ "paddingLeft":"0px" }}>
                    <div style={{ "backgroundColor":"#DCDCDC",
"height":"100%" }}>
                        {/* Chat header with refresh button,
username and public key are rendered here */}
                        <Row style={{ marginRight:"0px" }}>
                            <Card style={{ width:'100%',
alignSelf:'center', margin:"0 0 5px 15px" }}>
                              <Card.Header>
                                  { activeChat.friendname } : {
activeChat.publicKey }
                                  <Button style={{ float:"right"
}} variant="warning" onClick={ () => {
                                      if( activeChat &&
activeChat.publicKey )
                                          getMessage(
activeChat.publicKey );
                                  } }>
```

```
                                        Refresh
                                    </Button>
                                </Card.Header>
                            </Card>
                        </Row>
                        {/* The messages will be shown here */}
                        <div className="MessageBox" style={{
height:"400px", overflowY:"auto" }}>
                            { Messages }
                        </div>
                        {/* The form with send button and message
input fields */}
                        <div className="SendMessage"  style={{
borderTop:"2px solid black", position:"relative", bottom:"0px",
padding:"10px 45px 0 45px", margin:"0 95px 0 0", width:"97%" }}>
                            <Form onSubmit={ (e) => {
                                    e.preventDefault();
                                    sendMessage(
document.getElementById( 'messageData' ).value );
                                    document.getElementById(
'messageData' ).value = "";
                                }}>
                                <Form.Row
className="align-items-center">
                                    <Col xs={9}>
                                        <Form.Control
id="messageData" className="mb-2"  placeholder="Send Message" />
                                    </Col>
                                    <Col >
                                      <Button className="mb-2"
style={{ float:"right" }} onClick={ () => {
                                          sendMessage(
document.getElementById( 'messageData' ).value );
                                          document.getElementById(
'messageData' ).value = "";
                                      }}>
                                        Send
                                      </Button>
                                    </Col>
                                </Form.Row>
                            </Form>
                        </div>
                    </div>
                </Col>
            </Row>
        </Container>
    );
}
```

Βιβλιογραφία

1. Amores-Sesar, I., Cachin, C., & Tedeschi, E. (2022). When is Spring coming? A Security Analysis of Avalanche Consensus. arXiv preprint arXiv:2210.03423.
2. Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.".
3. Back, A. (2002). Hashcash-a denial of service counter-measure.
4. Basel Committee on Banking Supervision. (2022, December). Prudential treatment of cryptoasset exposures. https://www.bis.org/bcbs/publ/d545.pdf
5. Bank for International Settlements. (2022). The future trillions to banks. Retrieved from https://www.bis.org/publ/othp34.htm January 2023
6. Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*. white paper, 3(37), 2-1.
7. Bočánek, M. (2021). First draft of crypto-asset regulation (MICA) with the European Union and potential implementation. Financial Law Review, (22 (2)), 37-53.
8. Boneh, D. (2019). Avalanche Hashing. Presented at the Cryptography and Computer Security Seminar, Stanford University, January 16, 2019.
9. Brownworth, A. . Hash. Anders Brownworth's Blockchain Demo. https://andersbrownworth.com/blockchain/hash, 2016
10. Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI), 173-186.
11. Chauhan, A., Malviya, O. P., Verma, M., & Mor, T. S. (2018, July). Blockchain and scalability. In 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C) (pp. 122-128). IEEE.
12. Chowdhury, S. (2021). Uttam Kumar: A Life in Cinema. Bloomsbury Publishing.
13. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. IEEE Access, 4, 2292-2303.
14. CNBC. (2022, September 15). Ethereum's massive software upgrade just went live. Here's what it does. Retrieved from

https://www.cnbc.com/2022/09/15/ethereums-massive-software-upgrade-just-went-live-heres-what-it-does.html December 2022

15. Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2, 6-10.
16. Coblenz, M., Oei, R., Etzel, T., Koronkevich, P., Baker, M., Bloem, Y., ... & Aldrich, J. (2020). Obsidian: Typestate and assets for safer blockchain programming. ACM Transactions on Programming Languages and Systems (TOPLAS), 42(3), 1-82.
17. Dai, W. (1998). B-money. Wei Dai. Retrieved from http://www.weidai.com/bmoney.txt, January 2023
18. 2018. DappRadar - Ranked list of blockchain dapps. https://dappradar.com/. (2018)
19. De Vries, A. (2023). Cryptocurrencies on the road to sustainability: Ethereum paving the way for Bitcoin. Patterns, 4(1).
20. Ethereum Improvement Proposals. (2015, November 19). EIP-1: Ethereum Improvement Proposal 1 (EIP-1): Ethereum Improvement Process. Retrieved from https://eips.ethereum.org/EIPS/eip-1 October 2022
21. europa.eu. (2018, August 23). In full force from today GDPR: Now you decide about your digital privacy. Europa. Retrieved from https://europa.eu/newsroom/highlights/special-coverage/eu-data-protection-rules_en January 2023
22. European Council. (2021, December 21). Distributed ledger technology: Member States endorse agreement reached with European Parliament. Retrieved from https://www.consilium.europa.eu/en/press/press-releases/2021/12/21/distributed-ledger-technology-member-states-endorse-agreement-reached-with-european-parliament/ January 2023
23. European Papers, Vol. 7, 2022, No 2, European Forum, Insight of 24 September 2022, pp. 601-613 ISSN 2499-8249 - doi: 10.15166/2499-8249/581. Retrieved from: europeanpapers.eu/es/europeanforum/decentralized-finance-eu-law-regulation-on-pilot-regime-for-market-infrastructures-based-on-distributed-ledger-technology January 2023

24. European Parliament. (2022, December 21). Markets in crypto-assets: Parliament adopts new rules for crypto-assets. European Parliament Press Release. Retrieved from: https://www.europarl.europa.eu/news/en/press-room/20221216IPR40205/markets-in-crypto-assets-parliament-adopts-new-rules-for-crypto-assets February 2023
25. Ethereum.org
26. Ethereum.org(2020). Introducing Ethereum 2.0. Retrieved from https://ethereum.org/en/eth2/ (February 4, 2023),
27. Finley, Klint (27 January 2014). "Out in the open: Teenage hacker transforms web into one giant Bitcoin network". Wired. Retrieved from https://www.wired.com/2014/01/ethereum/ January 2023.
28. Forbes. (2022, January 17). BIS report urges banks to embrace blockchain, advanced analytics and collaboration with fintech firms. Retrieved from https://www.forbes.com/sites/michaeldelcastillo/2022/01/17/bis-report-urges-banks-to-embrace-blockchain-advanced-analytics-and-collaboration-with-fintech-firms/?sh=512aa3af7e32 January 2023
29. Feistel, Horst. "Cryptography and computer privacy." Scientific american 228.5 (1973): 15-23.
30. Gherghelas Sara(2022, December 21). Dapp Industry Report 2022. Dappradar Retrieved 20 January 2023 from https://dappradar.com/blog/dapp-industry-report-2022-dapp-industry-proves-resilient-in-crypto-winter.
31. Jani, S. (2017). An overview of ethereum & its comparison with bitcoin. Int. J. Sci. Eng. Res, 10(8), 1-6.
32. Jun, M. (2018). Blockchain government-a next form of infrastructure for the twenty-first century. Journal of Open Innovation: Technology, Market, and Complexity, 4(1), 7.
33. Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document (pp. 437-455). Springer Berlin Heidelberg.
34. Haque, A. B., Islam, A. N., Hyrynsalmi, S., Naqvi, B., & Smolander, K. (2021). GDPR compliant blockchains–a systematic literature review. IEEE Access, 9, 50593-50606.

35. Hildenbrandt, E., Saxena, M., Zhu, X., Rodrigues, N., Daian, P., Guth, D., & Roşu, G. (2017). Kevm: A complete semantics of the ethereum virtual machine.
36. Iansiti, M., & Lakhani, K. R. (2017). The truth about blockchain. Harvard Business Review, 95(1), 118-127.
37. Kiayias, A., & Lazos, P. (2022). SoK: blockchain governance. arXiv preprint arXiv:2201.07188.
38. Kelsey, J., Chang, S. J., & Perlner, R. (2016). SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash. NIST special publication, 800, 185.
39. Li, J., Shetty, S., & Yang, W. (2017). Blockchain and smart contract based healthcare information system. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 811-814). IEEE.
40. Li, K., Lau, W. F., Au, M. H., Ho, I. W. H., & Wang, Y. (2020). Efficient message authentication with revocation transparency using blockchain for vehicular networks. Computers & Electrical Engineering, 86, 106721.
41. Lynch, G. S. (2009). Single point of failure: The 10 essential laws of supply chain risk management. John Wiley and Sons.
42. Lu, Y., Qi, Q., & Chen, X. (2023). A Framework of Transaction Packaging in High-throughput Blockchains. arXiv preprint arXiv:2301.10944.
43. Makarov, I., & Schoar, A. (2022). Cryptocurrencies and Decentralised Finance (No. 1061). Bank for International Settlements.
44. Mendi, A. F., Erol, T., & Şafak, E. (2020). Generating a Blockchain Smart Contract Application Framework. Advances in Science Technology and Engineering Systems Journal.
45. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). Handbook of applied cryptography. CRC press.
46. Mettler, M. (2016). Blockchain technology in healthcare: The revolution starts here. In Proceedings of the 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom) (pp. 1-3). IEEE.
47. Mohanta, B. K., Panda, S. S., & Jena, D. (2018, July). An overview of smart contract and use cases in blockchain technology. In 2018 9th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-4). IEEE.

48. Moore, R. (2019-2021). Ethers.js Documentation. Retrieved from https://docs.ethers.io/v5/ February 2023
49. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* Decentralized business review, 21260.
50. Nakamoto, S. (2009). *Bitcoin: A peer-to-peer electronic cash system.* Bitcoin. org. Disponible en https://bitcoin. org/en/bitcoin-paper.
51. Narayanan, Arvind, et al. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.
52. Navadkar, V. H., Nighot, A., & Wantmure, R. (2018). Overview of blockchain technology in government/public sectors. International Research Journal of Engineering and Technology, 5(6), 2287-2292.
53. Nelson, P. (2018). Primer on Blockchain. USAID: Washington, DC, USA. Retrieved from https://www.usaid.gov/sites/default/files/2022-05/USAID-Primer-Blockchain.pdf December 2022
54. Pop, C. D., Antal, M., Cioara, T., Anghel, I., & Salomie, I. (2020). Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy. Sensors, 20(19), 5678.
55. Reuters. (2022, January 17). Digital platforms pose biggest challenge to banks - BIS report. Retrieved from https://www.reuters.com/markets/us/digital-platforms-pose-biggest-challenge-banks-bis-report-2022-01-17/ (January 2023)
56. Rocket, T., Yin, M., Sekniqi, K., van Renesse, R., & Sirer, E. G. (2019). Scalable and probabilistic leaderless BFT consensus through metastability. arXiv preprint arXiv:1906.08936.
57. Shanker, M. (2019). Use Case: Smart Contract for Lease Agreements using Blockchain Technology. International Journal of Scientific Research in Computer Science and Engineering, 7(6), 1-09.
58. Stavroulakis, P., & Stamp, M. (Eds.). (2010). Handbook of information and communication security. Springer Science & Business Media.
59. Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.
60. Szabo, N. (1997). Formalizing and securing relationships on public networks. First monday.

61. Team Rocket. (2018). Avalanche: A Novel Metastable Consensus Protocol Family for Cryptocurrencies. Retrieved from https://avalabs.org/avalanche-paper/
62. Tsampas, G. (2022). Survey on Decentralized Applications (Doctoral dissertation, University of Piraeus (Greece)).
63. Tikhomirov, S. (2018). Ethereum: state of knowledge and research perspectives. In Foundations and Practice of Security: 10th International Symposium, FPS 2017, Nancy, France, October 23-25, 2017, Revised Selected Papers 10 (pp. 206-221). Springer International Publishing.
64. Tromp, J. (2018). Cuckoo Cycle: A new memory-hard proof-of-work with less memory. https://eprint.iacr.org/2018/1050.pdf
65. Truong, N. B., Sun, K., Lee, G. M., & Guo, Y. (2019). Gdpr-compliant personal data management: A blockchain-based solution. IEEE Transactions on Information Forensics and Security, 15, 1746-1761.
66. Vogelsteller F., R. Ghods ,V. Maia , M. Garreau , E. Marks(2018, June 30). EIP-1193: Ethereum Provider JavaScript API. Retrieved from https://www.bis.org/publ/othp34.pdf November 2022
67. Working Group established by the Asian Consultative Council of the Bank for International Settlements. (2020, November 27). Capital flows, exchange rates and policy frameworks in emerging Asia. BIS. Retrieved from https://www.bis.org/publ/othp34.pdf January 2023
68. Zhang, J., Liu, X., & Shen, C. (2021). Digital Communication Techniques and Their Security in Communication Systems. In Handbook of Communications Security (pp. 1-24). Springer, Cham.
69. Zhang, S., & Lee, J. H. (2020). Analysis of the main consensus protocols of blockchain. ICT express, 6(2), 93-97.
70. Zhou, J., & Gollman, D. (1996, May). A fair non-repudiation protocol. In Proceedings 1996 IEEE Symposium on Security and Privacy (pp. 55-61). IEEE.
71. Zhou, J. (1997). *Cryptanalysis of the Cramer-Shoup cryptosystem using algebraic and lattice methods*. Advances in Cryptology — EUROCRYPT ’97, 1233, 250-266.

72. Zhou, J. (2002). *A new paradigm of signature schemes based on zero-knowledge proof*. Journal of Computer Science and Technology, 17(4), 373-379.
73. Zohar, A. (2015). Bitcoin: Under the hood. Communications of the ACM, 58(9), 104-113.