ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τμήμα Μηχανικών
Βιομηχανικής Σχεδίασης & Παραγωγής

Διπλωματική Εργασία

Τεχνολογία Blockchain και Ακαδημαϊκά Δεδομένα

του φοιτητή

Βέτσου Αναστάσιου

ΑΜ: 222017128

Διατριβή για λήψη *Βασικού* Διπλώματος

Επιβλέπουσα Καθηγήτρια:
ΛΕΛΙΓΚΟΥ ΕΛΕΝΗ ΑΙΚΑΤΕΡΙΝΗ

Αθήνα, ΙΟΥΝΙΟΣ, 2023

Δήλωση Συγγραφέα Διπλωματικής Εργασίας

Ο κάτωθι υπογεγραμμένος Βέτσος Αναστάσιος του Λεονάρντ, με αριθμό μητρώου 222017128 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι 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 πτυχίου μου».

Βέτσος Αναστάσιος

Εξεταστική Επιτροπή:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	**ΒΑΘΜΙΔΑ**	**ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ**
ΛΕΛΙΓΚΟΥ ΕΛ.-ΑΙΚ.	ΑΝΑΠΛΗΡΩΤΡΙΑ ΚΑΘΗΓΗΤΡΙΑ	Eleni Aikaterini Leligkou Digitally signed by Eleni Aikaterini Leligkou Date: 2023.07.19 11:15:01 +03'00'
ΔΡΟΣΟΣ Χ.	ΕΔΙΠ	Christos Drosos Digitally signed by Christos Drosos Date: 2023.07.20 22:15:14 +03'00'
ΚΑΝΤΖΟΣ Δ.	ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ	Dimitrios Kantzos Digitally signed by Dimitrios Kantzos Date: 2023.07.26 11:23:38 +03'00'

Πίνακας Περιεχομένων

Περίληψη

Με την μαζική ψηφιοποίηση σε πολλά μέρη της καθημερινότητας μας είναι εύκολο να παραλείψουμε το κομμάτι της ασφάλειας των πιο προσωπικών μας δεδομένων στον κυβερνοχώρο. Χρησιμοποιώντας τα σημερινά εργαλεία που έχουμε στην διάθεσή μας μπορούμε να χτίσουμε ασφαλή συστήματα για τα δεδομένα του χρήστη.

Σε αυτή την διπλωματική εργασία θα συνδέσουμε έναν oracle server σε μια ακαδημαϊκή εφαρμογή για την διατήρηση προσωπικών δεδομένων των φοιτητών. Για την ασφαλή επικοινωνία μεταξύ ενός smart contract και του ιδιωτικού oracle server θα σταλθούν αποκλειστικά ανώνυμα στοιχεία. Ο σκοπός του συστήματος είναι η επικύρωση και η εκτύπωση του διπλώματος του φοιτητή με την αμεταβλητότητα που μας διαθέτει το blockchain δίκτυο.

Για την σύνδεση ενός φοιτητή στην ακαδημαϊκή εφαρμογή, αλλά και για την ταυτοποίηση του από το ακαδημαϊκό σύστημα, θα χρειαστεί η χρήση ενός ηλεκτρονικού πορτοφολιού για την αλληλεπίδραση με το blockchain δίκτυο.

Λέξεις κλειδιά

Blockchain, Smart Contract, Oracle, Ethereum, Ψηφιακό Δίπλωμα, Ακαδημαϊκά Δεδομένα, ERC 721, dApp.

Abstract

With mass digitization being a large part of our daily lives, it's easy to ignore the security part of our most personal data in cyberspace. Using today's tools we can build secure systems for user’s data.

In this thesis we will connect an oracle server to an academic application for the retention of students' personal data. In order to achieve a secure communication between a smart contract and the private oracle server, only anonymous data will be sent. The purpose of the system is to validate and print the student's diploma with the invariance and immutability provided to us by the blockchain network.

In order to connect a student to the academic application, but also for the student to be identified by the academic system, an e-profile to interact with the blockchain network is required.

Key Words

Blockchain, Smart Contract, Oracle, Ethereum, Digital Degree, Academic Data, ERC 721, dApp.

Αναγνωρίσεις

Αρχικά, θα ήθελα να ευχαριστήσω την κυρία Ελένη Αικατερίνη Λελίγκου για την ευκαιρία που μου έδωσε να ασχοληθώ με αυτό το θέμα της διπλωματικής.

Στην συνέχεια, θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου Ιωάννη Χριστίδη για τη συνεχή υποστήριξη και καθοδήγηση καθ' όλη την διάρκεια την διπλωματικής μου εργασίας.

Επίσης, θα ήθελα να εκφράσω τη βαθύτατη εκτίμηση στους φίλους μου που διάβασαν και σχολίασαν τη διπλωματική εργασία μου και με βοήθησαν να καταλάβω καλυτέρα το θέμα από μια διαφορετική οπτική γωνία.

Ευχαριστώ τους γονείς μου που με στήριξαν για την ολοκλήρωση της διπλωματικής μου διατριβής καθώς και για την υπομονή, την ενθάρρυνση και την υποστήριξή τους.

Θεωρητικό Μέρος

Κεφάλαιο 1
Προϋποθέσεις Κατανόησης του Blockchain

1.1 Εισαγωγή

Το blockchain, συχνά αναφερόμενο σαν μια κατανεμημένη βάση δεδομένων, είναι μια δομή δεδομένων μιας πιο ευρύς τεχνολογίας κατανεμημένου καθολικού (Distributed Ledger Technology ή DLT).

1.2 Τεχνολογία Κατανεμημένου Καθολικού (DLT)

Καθολικό (ledger): *Μία συλλογή όλων των λογαριασμών μιας εταιρίας.*

Το καθολικό είναι αξιόπιστο και ανοιχτό για όλους τους συμμετέχοντες του συστήματος και για να εγγυηθούμε την αξιοπιστία του, το καθολικό πρέπει να είναι χρονικά ταξινομημένο και append only, δηλαδή νέες συναλλαγές μπορούν να προστεθούν αλλά οι ήδη υπάρχον συναλλαγές είναι αμετάβλητες. Για να επιβεβαιωθεί ότι το καθολικό δεν αλλοιώθηκε, πρέπει να χρησιμοποιηθεί μια κρυπτογραφική επιτομή (cryptographic digest). Μια επιτομή είναι μια συλλογή συμπυκνωμένων έργων που περιγράφεται σαν μια συμβολοσειρά μικρού μήκους, σε περίπτωση που ένα μέρος του καθολικού αλλάξει η προκύπτουσα επιτομή θα άλλαζε, και έτσι θα ανιχνευόταν η παραβίαση [1].
Συμφώνα με τον Κανονισμό του Ευρωπαϊκού Κοινοβουλίου το 2022 [6]:

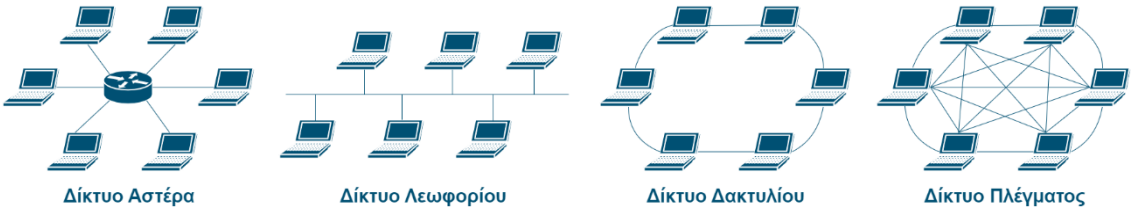
"*Ένα κατανεμημένο καθολικό σημαίνει ένα αποθετήριο πληροφοριών που κρατάει αρχεία συναλλαγών και το οποίο μοιράζεται και συγχρονίζεται μεταξύ ενός συνόλου κόμβων δικτύου DLT χρησιμοποιώντας έναν μηχανισμό συναίνεσης*"

Ο μηχανισμός συναίνεσης (consensus mechanism) χρησιμοποιείτε για να προσδιορίσει τη νομιμότητα μιας νέας συναλλαγής στο DLT. Κάθε προσθήκη στο

καθολικό από έναν συμμετέχοντα στο δίκτυο διαδίδεται σε όλους τους κόμβους (nodes) με αρχιτεκτονική peer-to-peer (P2P). Όταν η συναίνεση έχει επικυρωθεί από τους κόμβους προστίθεται στο DLT. Καταργώντας την ανάγκη σε μια κεντρική αρχή, οι πληροφορίες συγχρονίζονται στη συνέχεια σε όλο το δίκτυο σε κάθε κόμβο, γεγονός που εξασφαλίζει τη συνέπεια των δεδομένων [3].

1.3 Δίκτυο Ομότιμων Κόμβων (Peer-to-Peer Network)

Ένα peer-to-peer δίκτυο δεν έχει ένα κεντρικό εξυπηρετητή (server), αλλά αποτελείται από κόμβους (peers ή nodes) που μπορούν να αναλάβουν καθήκοντα εξυπηρετητή και πελάτη (client). Υπάρχουν διάφορες τοπολογίες του δικτύου, εκ των οποίων οι κυρίες είναι οι εξής:



Εικόνα 1.1.1: Διάφορες τοπολογίες peer-to-peer δικτύων. Δημιουργημένο από τον συγγραφέα.

Ένας κόμβος αναγνωρίζεται από την IP διεύθυνσή του ή από ένα διακριτό αναγνωριστικό κωδικό (GUID). Κάθε κόμβος έχει ένα σύνολο αναγνωριστικών για κάθε γειτονικό του κόμβο. Σε δυναμικά δίκτυα οι κομβόι μπορούν να συνδεθούν και να αποσυνδεθούν κατά βούληση [7]. Ο κάθε κόμβος μπορεί να επικοινωνήσει με έναν άλλο εφόσον ξέρει το αναγνωριστικό του μέσω TCP/IP και UDP. Σε περίπτωση που δεν το ξέρει, μπορεί να επικοινωνήσει με τους γειτονικούς κόμβους για το δικό τους σύνολο αναγνωριστικών και έτσι το αίτημα επικοινωνίας μεταδίδεται στο δίκτυο.

1.4 Κρυπτογραφική Συνάρτηση Κατακερματισμού (Cryptographic Hash Function)

Ο κατακερματισμός (hashing) είναι μια κρυπτογραφική συνάρτηση, που παίρνει ως είσοδο οποιουδήποτε μεγέθους δεδομένα και δημιουργεί μια συμβολοσειρά σταθερού μεγέθους γνωστή και ως επιτομή (digest). Επιτρέπει την είσοδο δεδομένων, κατακερματισμού αυτών των δεδομένων και την επίτευξη του ίδιου αποτελέσματος

ανεξαρτήτως του ατόμου, αποδεικνύοντας ότι τα δεδομένα ήταν αμετάβλητα. Μια εντελώς διαφορετική επιτομή εξόδου θα προκύψει ακόμη και από την αλλαγή ενός bit εισόδου.

Εικόνα 1.1.2: Λειτουργία κατακερματισμού. Δημιουργημένο από τον συγγραφέα.

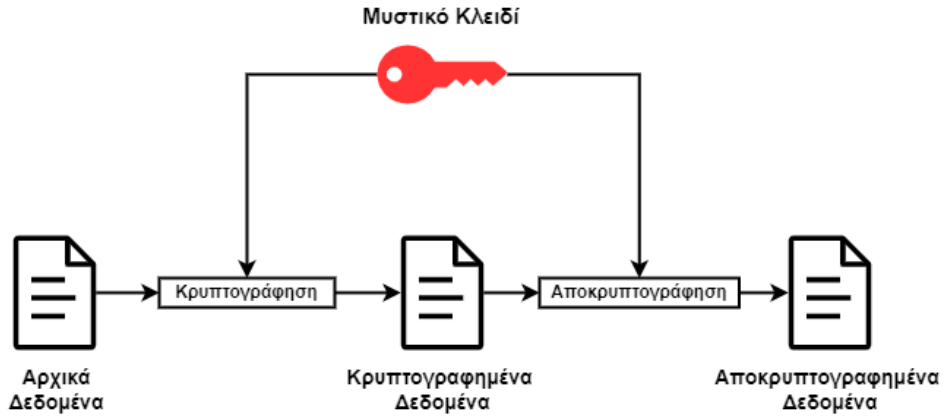
Οι συναρτήσεις κατακερματισμού (hash) έχουν τρεις ιδιότητες ασφάλειας [9]:

- Είναι preimage resistant, που σημαίνει ότι είναι μιας κατεύθυνσης. Είναι υπολογιστικά ανέφικτο να υπολογιστεί η σωστή τιμή εισόδου δεδομένης κάποιας τιμής εξόδου (π.χ. έχοντας μια επιτομή y, δεν μπορούμε να υπολογίσουμε την είσοδο x, της hash(x) = y).
- Είναι second preimage resistant, που σημαίνει ότι είναι υπολογιστικά ανέφικτο να βρεθεί μια είσοδος που κατακερματίζει σε μια συγκεκριμένη έξοδο (π.χ. έχοντας είσοδο x, να βρεθεί το y, ώστε να ισχύει hash(x) = hash(y)).
- Είναι collision resistant, που σημαίνει ότι είναι υπολογιστικά ανέφικτο να βρεθούν δύο είσοδοι που κατακερματίζονται στην ίδια έξοδο. (π.χ. να βρεθούν οι είσοδοι x και y, όπου hash(x) = hash(y)).

1.5 Συμμετρική Κρυπτογραφία (Symmetric Cryptography)

Γνωστή κα ως κρυπτογραφία μυστικού κλειδιού (secret key), αυτή η μέθοδος χρησιμοποιεί ένα κλειδί για κρυπτογράφηση και αποκρυπτογράφηση [10]. Τα

δεδομένα κρυπτογραφούνται με το κλειδί, μεταφέρονται στο δίκτυο και αποκρυπτογραφούνται με το ίδιο κλειδί, έτσι ώστε να διατηρούμε την εμπιστευτικότητα κατά την διάδοση των δεδομένων.

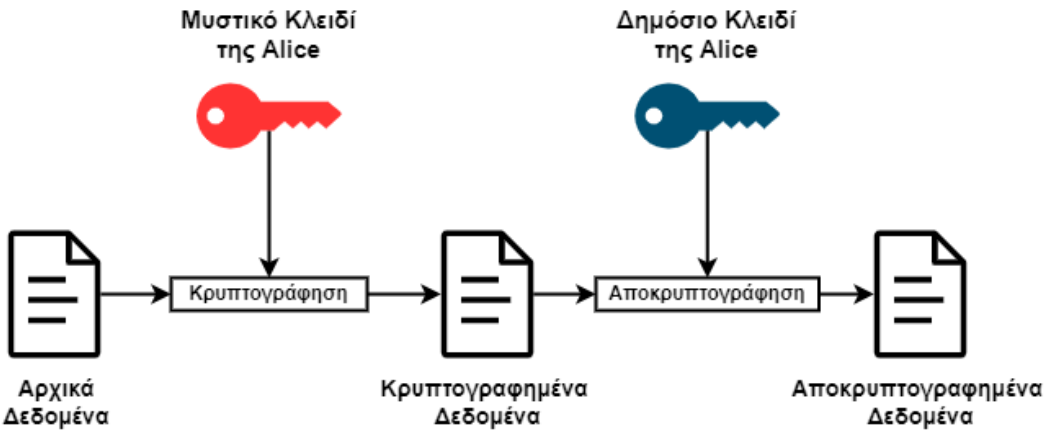


Εικόνα 1.1.3: Διαδικασία συμμετρικής κρυπτογραφίας. Δημιουργημένο από τον συγγραφέα.

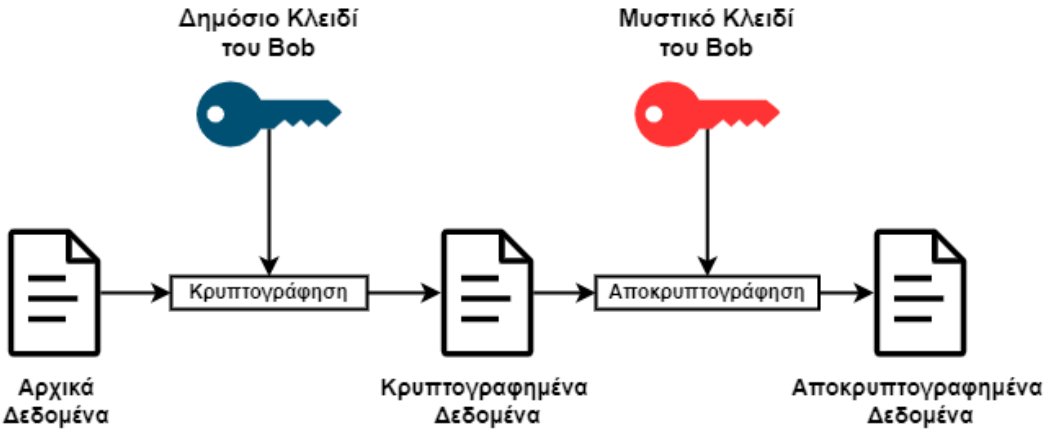
Η προφανής αδυναμία αυτής της μεθόδου είναι ότι σε περίπτωση που το κλειδί γίνει γνωστό από κάποιο κακόβουλο άτομο, θα μπορεί να αποκρυπτογραφήσει τα δεδομένα. Ένας σημαντικός αλγόριθμος κατά την διάρκεια του χρόνου είναι ο DES [11], του οποίου η κύρια αδυναμία του είναι ότι σήμερα το μέγεθος του κλειδιού (64 bit, 54 από αυτά χρησιμοποιούνται απευθείας από τον αλγόριθμο) θεωρείται ευάλωτο.

1.6 Ασύμμετρης Κρυπτογραφία (Asymmetric Cryptography)

Γνωστή κα ως κρυπτογραφία δημοσίου κλειδιού (public key), αυτή η μέθοδος χρησιμοποιεί ένα ζεύγος κλειδιών που είναι μαθηματικά σχετικές, το ζεύγος κλειδιών αποτελείτε από ένα δημόσιο κλειδί και ένα μυστικό κλειδί. Παραδοσιακά, το μυστικό κλειδί χρησιμοποιείται για την κρυπτογράφηση των δεδομένων και δεν αποκαλύπτεται ποτέ σε άλλο άτομο, ενώ το δημόσιο κλειδί αποκρυπτογραφεί τα δεδομένα και μπορεί να διαφημιστεί τόσο ευρέως όσο θέλει ο ιδιοκτήτης. Δεν έχει σημασία όμως ποιο κλειδί εφαρμόζεται πρώτα από τα δυο. Ας πούμε ότι η Alice θέλει να στείλει ένα μήνυμα στον Bob.



Εικόνα 1.1.4: Η Alice θα κρυπτογραφήσει το μήνυμα χρησιμοποιώντας το μυστικό κλειδί της. Ο Bob αποκρυπτογραφεί το μήνυμα χρησιμοποιώντας το δημόσιο κλειδί της Alice. Δημιουργημένο από τον συγγραφέα.



Εικόνα 1.1.5: Η Alice θα κρυπτογραφήσει το μήνυμα χρησιμοποιώντας το δημόσιο κλειδί του Bob. Ο Bob θα αποκρυπτογραφήσει μήνυμα χρησιμοποιώντας το μυστικό κλειδί του. Δημιουργημένο από τον συγγραφέα.

Ο γνωστός αλγόριθμος RSA υλοποιεί ένα κρυπτοσύστημα δημόσιου κλειδιού, καθώς και ψηφιακές υπογραφές [5][12]. Ένα δημόσιο κλειδί περιγράφεται σαν ένα ζεύγος ακέραιων αριθμών (e, N), όπου $N = p \cdot q$, όπου p, q, είναι δυο μεγάλοι πρώτοι αριθμοί και ο e ο πρώτος ως προς τον αριθμό $\varphi(N) = (p - 1)(q - 1)$.

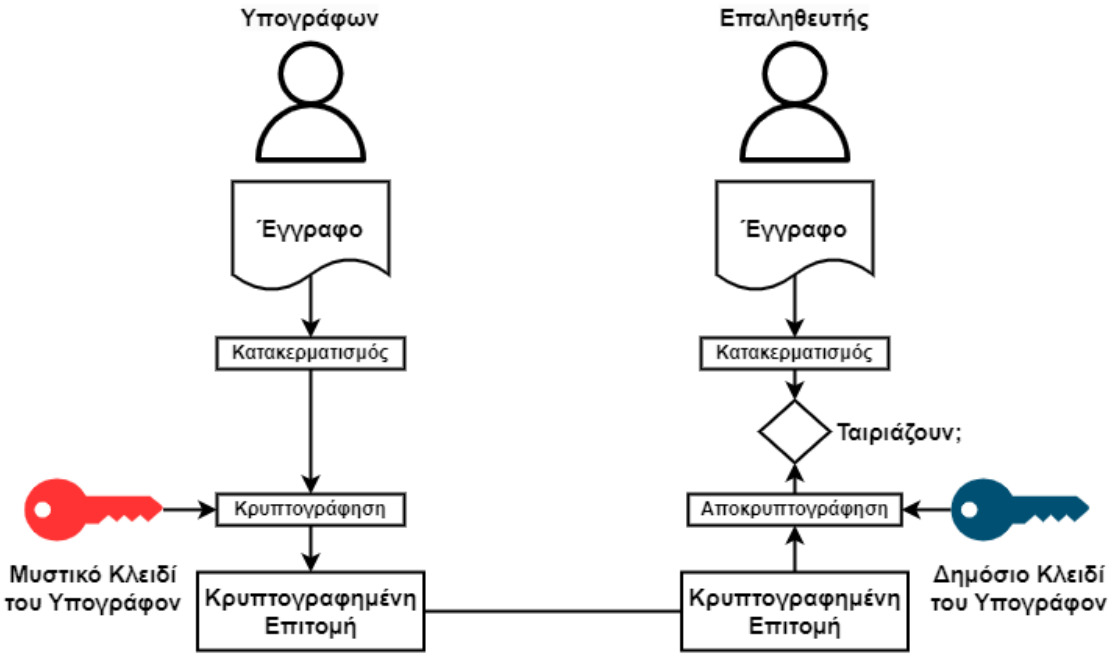
Εξίσωση κρυπτογράφησης: c = m^e mod N

Εξίσωση αποκρυπτογράφησης: m = c^d mod N

Όπου d το μυστικό κλειδί του παραλήπτη που υπολογίζεται από την σχέση:
d = e^{-1} mod φ(N)

1.6.1 Ψηφιακές Υπογραφές (Digital Signatures)

Οι ψηφιακές υπογραφές μπορούν να χρησιμοποιηθούν για την επαλήθευση της αυθεντικότητας των υπογεγραμμένων δεδομένων και την ταυτότητα του υπογράφοντος. Μια ψηφιακή υπογραφή μπορεί χρησιμοποιηθεί από τον παραλήπτη ενός υπογεγραμμένου μηνύματος για να αποδείξει στον ίδιο ή σε τρίτο μέλος ότι ο υπογράφων που δημιούργησε την υπογραφή το έκανε πραγματικά. Δεδομένου ότι ο υπογράφων δεν μπορεί ευκολά να αποκηρύξει την υπογραφή του σε αργότερη στιγμή, γνωστό και ως non-repudiation. Το ηλεκτρονικό ταχυδρομείο, οι ηλεκτρονικές χρηματοοικονομικές μεταφορές, η ηλεκτρονική ανταλλαγή δεδομένων, η αποθήκευση δεδομένων και άλλες εφαρμογές που απαιτούν διασφάλιση ακεραιότητας δεδομένων και επαλήθευση χρησιμοποιούν αλγόριθμους ψηφιακών υπογραφών [17].



Εικόνα 1.1.6: Η διαδικασία επαλήθευσης μιας ψηφιακής υπογραφής. Δημιουργημένο από τον συγγραφέα.

Η διαδικασία επαλήθευσης περιγράφεται ως εξής: Σε περίπτωση που ο επαληθευτής δεν έχει το έγγραφο (μήνυμα ή δεδομένα) που θα υπογραφεί, θα σταλθεί σε αυτόν. Ο υπογράφων κατακερματίζει το έγγραφο, κρυπτογραφεί την παραγομένη επιτομή με

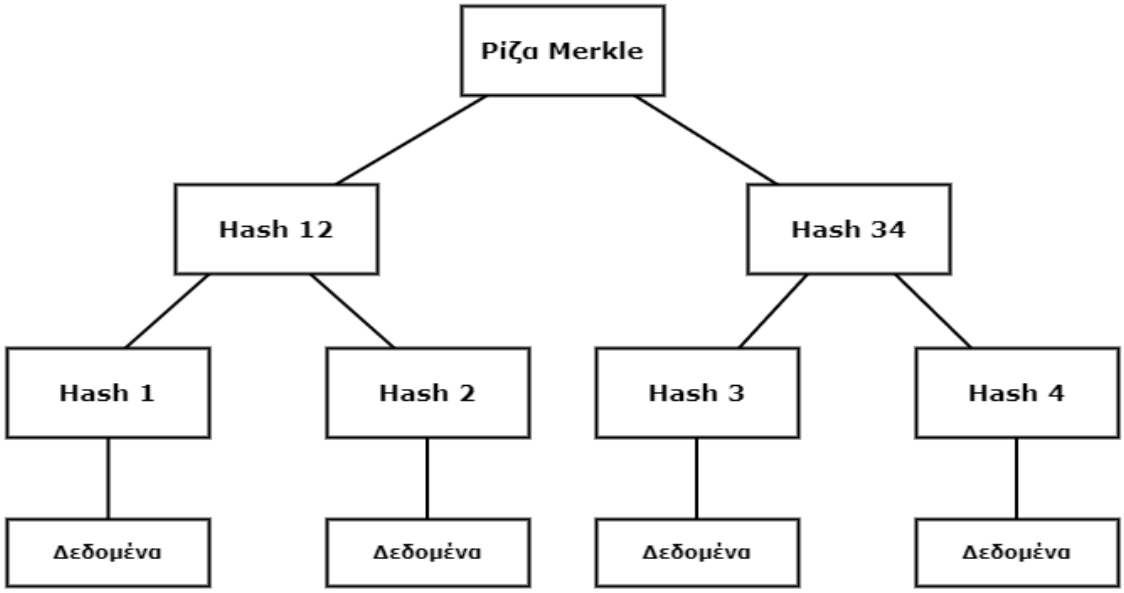
το μυστικό κλειδί του και θα το στείλει στον επαληθευτή. Στην συνέχεια ο επαληθευτής κατακερματίζει το έγγραφο του, αποκρυπτογραφεί την κρυπτογραφημένη επιτομή που έλαβε με το δημόσιο κλειδί του υπογράφων και ελέγχει αν οι δυο επίτομες είναι ιδίες.

Στην περίπτωση RSA, η ψηφιακή υπογραφής ενός μηνύματος m παράγεται ως εξής [5]:

- Υπολογισμός της κρυπτογραφικής επιτομής h(m).
- Υπολογισμός της υπογραφής s = $h(m)^d$ mod N
- Όπου d το μυστικό κλειδί του υπογράφοντα και (e, N) το δημόσιο κλειδί.

1.7 Merkle Trees

Τα Merkle Trees που ονομάζονται από τον Ralph Merkle [13], παράγονται με συνεχή υπολογισμό του ριζικού κατακερματισμού ή της ρίζας Merkle, η οποία είναι μια επιτομή των προηγουμένων ζευγών κατακερματισμού. Κατασκευάζεται χρησιμοποιώντας μια μέθοδο από κάτω προς τα πάνω στην οποία κάθε κόμβος κατακερματίζει την επιτομή των υπο-κόμβων (branch nodes) του και ούτω καθεξής, μέχρι να υπάρξει μια ενιαία επιτομή των αρχειακών δεδομένων (block data). Οι κόμβοι του πρώτου επιπέδου γνωστοί και ως leaf nodes κατακερματίζουν τα αρχικά δεδομένα.



Εικόνα 1.1.7: Δομή του Merkle Tree. Δημιουργημένο από τον συγγραφέα.

Μας επιτρέπει να αποδείξουμε αποτελεσματικά ότι κάποιο κομμάτι δεδομένων χρησιμοποιήθηκε για τη δημιουργία ρίζας Merkle χωρίς να χρειάζεται να έχουμε πρόσβαση ή να αποθηκεύσουμε όλα τα αρχικά block δεδομένων [14].

1.8 Ψηφιακές Διευθύνσεις

Ο David Chaum πρότεινε την τεχνική [15] που στην επικοινωνία μεταξύ χρηστών ο ανταποκριτής μπορεί να παραμείνει ανώνυμος, ενώ του επιτρέπει να ανταποκριθεί μέσω μιας διεύθυνσης επιστροφής. Μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία καταλόγων μη ανιχνεύσιμων ψηφιακών ψευδώνυμων από επιλεγμένες εφαρμογές.

Η διεύθυνση (address) είναι μια σύντομη αλφαριθμητική συμβολοσειρά η οποία λειτουργεί ως αναγνωριστικό για τον λογαριασμό ενός χρήστη. Υπάρχουν πολλές προσεγγίσεις για την δημιουργία μιας διεύθυνσης, αλλά μιλώντας πιο γενικευμένα, είναι η επιτομή ενός δημόσιου κλειδιού μετά από κατακερματισμό [5].

Κεφάλαιο 2
Blockchain

2.1 Εισαγωγή

Το blockchain είναι ένα κατανεμημένο καθολικό, αποτελούμενο από blocks. Κάθε block, εκτός του πρώτου (genesis block), περιέχει τη κρυπτογραφική επιτομή του προηγούμενου, σχηματίζοντας έτσι μια αλυσίδα. Η πρώτη σύγκλιση ιδεών, οι οποίες στο μέλλον θα είχαν τον ορισμό blockchain, προήρθε από εφευρέτη του Bitcoin με το ψευδώνυμο Satoshi Nakamoto το 2008 [2].

2.2 Κατηγοριοποίηση Blockchain

2.2.1 Δημόσια αλυσίδα (Public chain) ή Χωρίς Άδεια (Permissionless)

Σε μια δημόσια αλυσίδα κάθε κόμβος μπορεί να συνδεθεί και να αποσυνδεθεί κατά βούληση από το δίκτυο, συχνά το λογισμικό τους είναι ανοικτό προς όλους (open source). Οποιοσδήποτε κόμβος έχει το δικαίωμα να δημοσιεύει blocks, αυτό έχει ως αποτέλεσμα την ιδιότητα ότι ο καθένας μπορεί να διαβάσει το blockchain καθώς και να καταγράψει συναλλαγές [9].

2.2.2 Ιδιωτική αλυσίδα (Private chain) ή Με Άδεια (Permissioned)

Μια ιδιωτική αλυσίδα δεν είναι ανοιχτή δημόσια, ο κάθε κόμβος πρέπει να έχει αρχική πρόσβαση ή να εγκριθεί από κάποια αρχή για να συνδεθεί, γνώστη και ως federated chain [8]. Οι κομβόι μπορεί να έχουν διαφόρων ειδών πρόσβασης σε λειτουργίες [9].

2.3 Αρχιτεκτονική ενός Blockchain

Η τεχνολογία blockchain περιγράφεται σε μια αρχιτεκτονική πέντε επιπέδων [8]:

2.3.1 Επίπεδο Εφαρμογής (Application Layer)

Στο επίπεδο εφαρμογής είναι το σημείο αλληλεπίδρασης των τελικών χρηστών με το δίκτυο του blockchain (π.χ. Κρυπρονομίσματα, Έλεγχος Πρόσβασης, Internet of Things, Ιατρικοί Φάκελοι) [5]. Η έρευνα επικεντρώνεται κυρίως σε Cross-chain, τεχνολογίες με στόχος τη μεταφορά δεδομένων μεταξύ διαφορετικών blockchains [8].

2.3.2 Επίπεδο Συμβολαίου (Contract Layer)

Το επίπεδο συμβολαίου είναι το θεμέλιο των προγραμματιζόμενων χαρακτηριστικών του blockchain [8]. Σε αυτό το επίπεδο περιγράφεται το περιβάλλον των έξυπνων συμβολαίων (Smart Contracts), οπού είναι προγράμματα υπολογιστή που εκτελούν όρους μιας σύμβασης ή συμφωνίας με βάση την κατάσταση του συστήματος, θα διατρίβουμε περισσότερο σε αυτά στο *Κεφάλαιο 3*.

2.3.3 Επίπεδο Συναίνεσης (Consensus Layer)

Σε αυτό το επίπεδο χρησιμοποιούνται αλγόριθμοι συναίνεσης για την διασφάλισή της συνέπειας των πληροφοριών που κάθε κόμβος καταγράφει [8], στο *2.6* αναφέρονται μερικοί αλγόριθμοι συναίνεσης.

2.3.4 Επίπεδο Δικτύου (Network Layer)

Στο επίπεδο δικτύου χρησιμοποιούμε το μοντέλο peer-to-peer για την επικοινωνία των κόμβων, η λειτουργία του είναι η διάδοση των blocks και φροντίζει ότι όλοι οι κόμβοι μπορούν να λάβουν μηνύματα συναλλαγών [5][8].

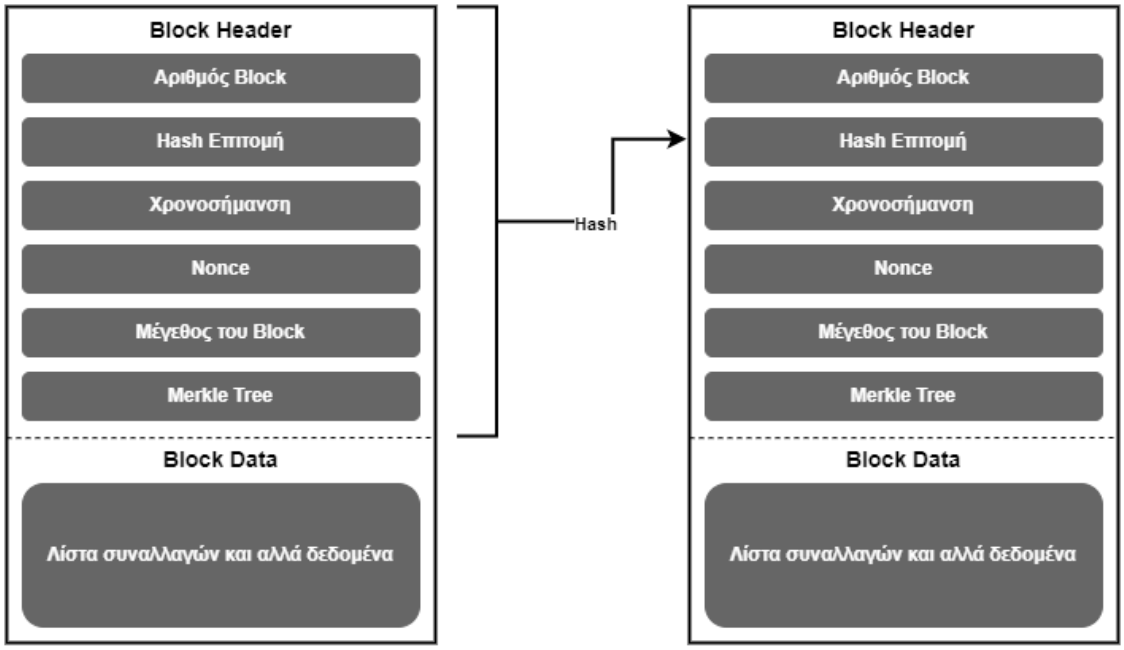
2.3.5 Επίπεδο Δεδομένων (Data Layer)

Το επίπεδο δεδομένων περιγράφετε σαν το κομμάτι DLT του blockchain μέσω μιας συγκεκριμένης δομής δεδομένων που αποτελείτε από blocks. Τα δεδομένα

αποθηκεύονται σε μεμονωμένους κόμβους, στην περίπτωση του Bitcoin ένα block αποθηκεύετε σαν ένα αρχείο που το όνομα του αντιστοιχεί στη σειρά του στην αλυσίδα [8].

2.4 Block

Το πιο σημαντικό στοιχείο σε ένα blockchain είναι το block. Κάθε block αποτελείτε από μια block κεφαλίδα (header) και από block δεδομένα (data) [5][9]. Σε κάθε σύστημα μπορεί να καθοριστούν διαφορετικά πεδία αλλά τυπικά περιλαμβάνονται τα ακόλουθα:



Εικόνα 1.2.1: Η δομή ενός block. Δημιουργημένο από τον συγγραφέα.

2.4.1 Block Header

Ο αριθμός block αντιπροσωπεύει τη σειρά του block στην αλυσίδα, η hash επιτομή του προηγουμένου block header σχηματίζοντας έτσι την αλυσίδα, μια χρονοσήμανση που ήταν μια από την κρίσιμες ιδέες της τεχνολογίας blockchain [1][16], μια τυχαία τιμή (nonce) που δημιουργείται από μια γεννήτρια τυχαίων συμβολοσειρών, το μέγεθος του block και το Merkle Tree τον δεδομένων του block (block data), που χρησιμοποιείται για την αποτελεσματική και ασφαλή κωδικοποίηση και κρυπτογράφηση δεδομένων κάθε block.

2.4.2 Block Data

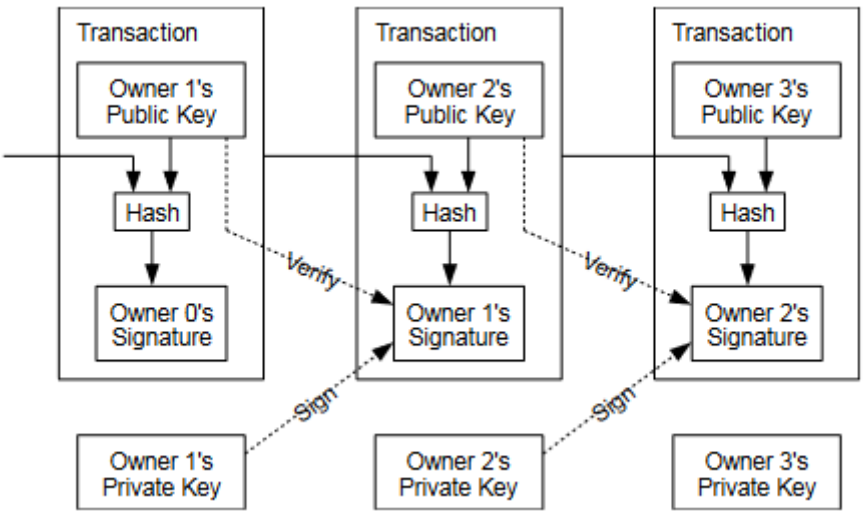
Μια λίστα συναλλαγών ή και συμβάντων που περιλαμβάνονται στο block ή και αλλά δεδομένα.

2.4.3 Genesis Block

Το genesis block είναι η ρίζα του blockchain, είναι το μόνο προρυθμισμένο block και κάθε block πρέπει να προστεθεί μετά από αυτό.

2.5 Συναλλαγές (Transactions)

Μια συναλλαγή είναι μια αναπαράσταση μιας αλληλεπίδρασης μεταξύ μερών. Στην περίπτωση των κρυπτονομισμάτων, μια συναλλαγή αντιπροσωπεύει μια μεταφορά του κρυπτονομίσματος μεταξύ χρηστών του δικτύου blockchain.



Εικόνα 1.2.2: Συνδεσιμότητα μεταξύ συναλλαγών. Διαθέσιμο [2].

Συχνά μια μεταφορά κρυπτονομίσματος χρειάζεται τις ακόλουθες πληροφορίες [9]:

- **Είσοδοι** μπορεί να είναι μια αναφορά μιας πηγής του ψηφιακού περιουσιακού στοιχείου (digital asset), είτε της προηγούμενη συναλλαγής στην οποία

δόθηκε στον αποστολέα ή και ένα νέο αμετάβλητο ψηφιακό περιουσιακό στοιχείο που μπορεί να χωριστεί σε πολλαπλά κομμάτια.

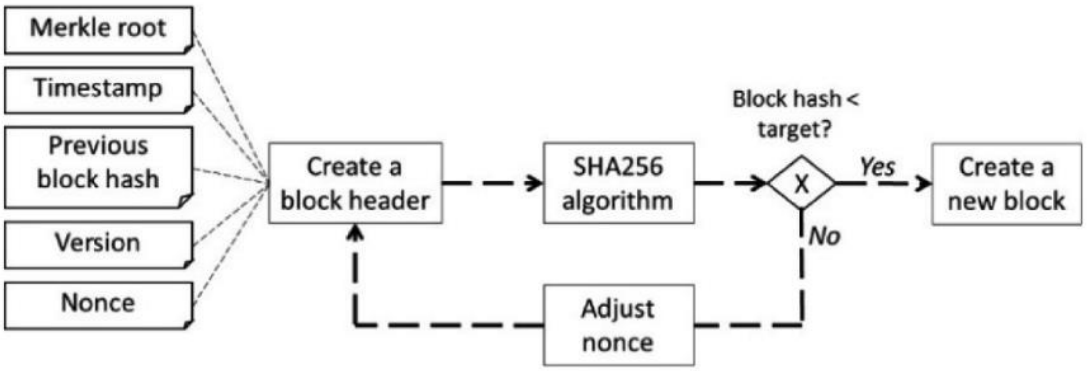
- Οι **έξοδοι** εξάγουν τα digital assets σε λογαριασμούς, αποτελούντα από το αναγνωριστικό των digital assets, την ψηφιακή διεύθυνση του νέου ιδιοκτήτη και ένα σύνολο προϋποθέσεων που πρέπει να πληρούν οι νέοι ιδιοκτήτες. Τα πρόσθετα ψηφιακά περιουσιακά πρέπει να επιστραφούν στον αποστολέα εάν είναι περισσότερα από όσα είναι απαραίτητα.

Με την ψηφιακή υπογραφή μιας επίτομης της προηγούμενης συναλλαγής, του δημόσιου κλειδιού του επόμενου κατόχου και την προσθήκη τους στο τέλος του κέρματος, κάθε ιδιοκτήτης μεταφέρει την ιδιοκτησία του κέρματος στο επόμενο. Ένας δικαιούχος μπορεί να ελέγξει τις υπογραφές για να επιβεβαιώσει την αλυσίδα ιδιοκτησίας [2]. Σε πολλές περιπτώσεις, σε μια συναλλαγή μπορεί να καταγράφονται προσθετά δεδομένα [5]. Μια συναλλαγή γενικά δεν αναγνωρίζεται ως επιβεβαιωμένη έως ότου δημοσιευθούν περισσότερα block πάνω από το blockμε την σχετική συναλλαγή [9].

2.6 Αλγόριθμοι Συναίνεσης

2.6.1 Απόδειξη Εργασίας (Proof-of-Work)

Ο αλγόριθμος απόδειξης εργασίας (PoW) επιλέγει ένα πρόβλημα που μπορεί να λυθεί μόνο μαντεύοντας την λύση για να δημιουργηθεί και να επικυρωθεί ένα πλήρες block. Η πρόκληση είναι να προσδιοριστεί μια τιμή nonce, έτσι ώστε, όταν χρησιμοποιηθούν τα δεδομένα συναλλαγής και την τιμή nonce ως εισόδους σε μια συνάρτηση κατακερματισμού [18], το πρόβλημα έχει λυθεί όταν η παράγωγη επιτομή εξόδου της συνάρτησης κατακερματισμού ικανοποιεί τη δυσκολία που έχει τεθεί από τον αλγόριθμο.



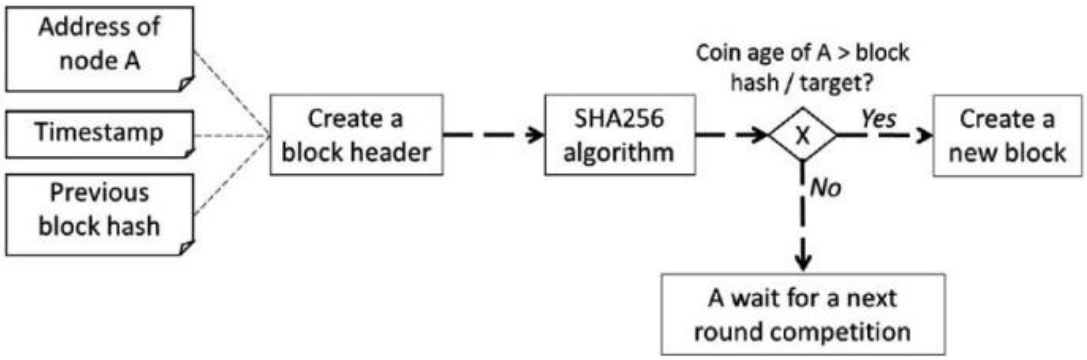
Εικόνα 1.2.3: Σύστημα Proof-of-Work. Διαθέσιμο [30].

Η δυσκολία είναι ανάλογη της συχνότητας δημοσίευσης των blocks, έτσι ώστε να είναι μια υπολογιστικά εντατική διαδικασία. Προς το παρόν δεν υπάρχει γνωστός τρόπος συντομεύσεις αυτής της διαδικασίας. Για να προσδιορίσουν την κατάλληλη τιμή nonce, οι κόμβοι πρέπει να επενδύσουν χρόνο, πόρους και υπολογιστική δύναμη, λειτουργία γνωστή και ως mining [9].

Ένα γνωστό πρόβλημα είναι οι υπολογιστικές απαιτήσεις και κατανάλωση ρεύματος που χρειάζεται ο αλγόριθμος απόδειξης εργασίας [20].

2.6.2 Απόδειξη Συμμετοχής (Proof-of-Stake)

Ο αλγόριθμος απόδειξης συμμετοχής (PoS) χρησιμοποιεί εικονικούς πόρους, όπως ως το μερίδιο συμμέτοχης (stake) ενός κόμβου, για να εκτελέσει την εκλογή αυτού που θα δημοσιεύσει το block και διατηρώντας την συναίνεση του δικτύου [19]. Ο καθοριστικός παράγοντας επιλογής είναι το μερίδιο ενός κόμβου στο σύστημα. Όσο περισσότερο το μερίδιο ενός χρήστη είναι στο σύστημα, τόσο πιο πιθανό είναι να θέλει το σύστημα να πετύχει [9].



Εικόνα 1.2.4: Σύστημα Proof-of-Stake. Διαθέσιμο [30].

Ο αλγόριθμος PoS είναι γρήγορος και έχει ως αποτέλεσμα χαμηλά έξοδα σε σχέση με αυτά του αλγόριθμου PoW. Τα πρωτόκολλα PoS αναπτύσσονται με σκοπό την παροχή μιας πιο αποτελεσματικής και φιλικής προς το περιβάλλον εναλλακτικής λύσης [19].

2.6.3 Αλγόριθμος Περασμένου Χρόνου (Proof-of-Elapsed-Time)

Στον αλγόριθμο περασμένου χρόνου κάθε πιθανός κόμβος ζητά έναν τυχαίο χρόνο αναμονής. Ο χρόνος αναμονής θα δημιουργείται από μια αξιόπιστη πλατφόρμα ασφαλών υπολογιστών και θα επιστραφεί στον κάθε κόμβο. Αφού περιμένουν την καθορισμένη περίοδο, ο πρώτος κόμβος που θα ολοκληρώσει την αναμονή είναι ο νικητής και μπορεί να δημοσιεύσει το νέο block [8][9].

2.6.4 Πρακτική Βυζαντινή Ανοχή Σφαλμάτων (Practical-Byzantine-Fault-Tolerance)

Η πρακτική Βυζαντινή ανοχή σφαλμάτων (PBFT) είναι η επίλυση σε ένα γνωστό πρόβλημα στρατηγών στο οποίο ορισμένοι στρατηγοί είναι ανέντιμοι, αλλά πρέπει να επιτευχθεί συναίνεση μεταξύ τους.

Με τον διαμοιρασμό μιας συναλλαγής, οι κόμβοι χρησιμοποιούν το δημόσιο κλειδί του υπογράφοντος για την διαδικασία ελέγχου. Όσο οι κακόβουλοι κόμβοι

αποτελούν λιγότερο από το ένα τρίτο όλων των κόμβων στο σύστημα, οι κόμβοι θα καταλήξουν σε συμφωνία για την παρούσα κατάσταση του blockchain [5][8].

2.6.5 Αλγόριθμος Χρονοπρογραμματισμού (Round-Robin)

Στον αλγόριθμο χρονοπρογραμματισμού (RR), οι κόμβοι εναλλάσσονται στην δημιουργία των blocks. Για να χειριστούν καταστάσεις όπου ένας κόμβος δεν είναι διαθέσιμος για την δημοσίευση ενός block στην σειρά του, τα συστήματα αυτά μπορεί να περιλαμβάνουν ένα χρονικό όριο για να επιστρέψουν στους διαθέσιμους κόμβους να δημοσιεύσουν blocks, έτσι ώστε οι μη διαθέσιμοι κόμβοι να μην προκαλέσουν διακοπή της δημοσίευσης νέων block [9].

2.6.6 Απόδειξη Αρχής (Proof-of-Authority)

Ο αλγόριθμος απόδειξης αρχής βασίζεται στην εμπιστοσύνη μεταξύ των κόμβων (αρχών) και την ταυτότητα τους. Οι ταυτότητες των κόμβων πρέπει να αποδεικνύονται και να επαληθεύονται μέσα στο δίκτυο blockchain. Εφόσον οι ταυτότητες είναι αναγνωρίσιμες, αποθαρρύνουν την κακόβουλη χρήση του δικτύου [5].

2.7 Λειτουργίες των Κόμβων

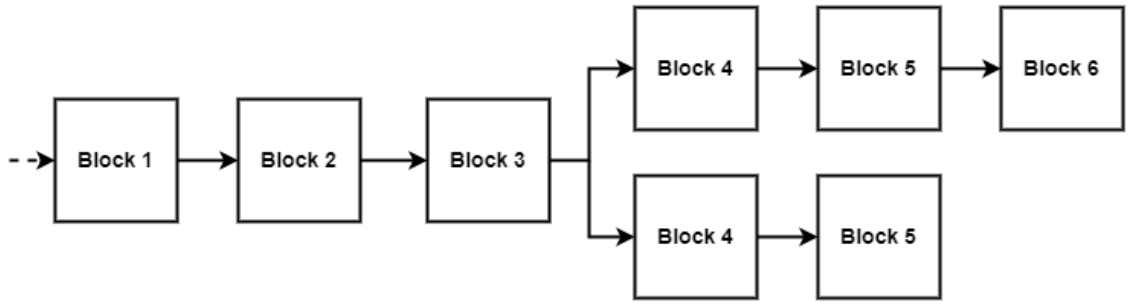
Οι κόμβοι «ακούνε» για επικυρωμένες συναλλαγές, τις εκπέμπουν στο δίκτυο και παράλληλα τις καταγράφουν σε ένα block. Όταν οι συναλλαγές καταγραφούν στο block και ο μηχανισμός συναίνεσης πληρείται από το κόμβο που ωθεί το block στο δίκτυο, οι συναλλαγές ή και τα δεδομένα στο block αποτελούν πλέον μέρος του δικτύου.

Ο κόμβος που δημοσίευσε το block ανταμείβεται από το σύστημα με ένα νέο ψηφιακό στοιχείο ή και ένα τέλος από τις συναλλαγές στο εν λόγω block. Έχοντας

μια συνεχή παροχή νέων block που δημοσιεύονται, αποτρέπει τους κακόβουλους χρήστες να προλάβουν και να κατασκευάσουν ένα μακρύτερο, αλλοιωμένο blockchain [9]. Οι κόμβοι θεωρούν πάντα ότι η μεγαλύτερη αλυσίδα είναι η σωστή και θα συνεχίσουν να εργάζονται για την επέκτασή της [2].

2.8 Forking

Πολλαπλά block μπορούν να εκδοθούν σχεδόν ταυτόχρονα σε διαφορά δίκτυα blockchain. Ένας κόμβος σε πολλά συστήματα κατανεμημένων δικτύων μπορεί να βρίσκεται πίσω σε πληροφορίες ή πιθανό και να έχουν διαφορετικές πληροφορίες. Οι αντιθέσεις συνήθως λύνονται σχετικά γρήγορα. Τα περισσότερα δίκτυα blockchain χρησιμοποιούν την μεγαλύτερη αλυσίδα σαν την πραγματική μέχρι να δημοσιευτεί το επόμενο block [9].



Εικόνα 1.2.5: Παράδειγμα Forking. Δημιουργημένο από τον συγγραφέα.

Τα forks είναι προσωρινές αντιθέσεις, όπως περιγράφονται παραπάνω, και πιο συχνά ο ορός χρησιμοποιείται σε περιπτώσεις τροποποιήσεων στο πρωτόκολλο και στις δομές δεδομένων ενός δικτύου blockchain. Υπάρχουν δυο κατηγορίες, τα soft forks και τα hard forks.

2.8.1 Soft Forks

Οι περιπτώσεις όπου μη ενημερωμένοι κόμβοι μπορούν να συνεχίσουν να πραγματοποιούν συναλλαγές με ενημερωμένους κόμβους ονομάζονται soft forks. Για

παράδειγμα, εάν ένα blockchain επέλεξε να μειώσει το μέγεθος του block. Οι μη ενημερωμένοι κόμβοι θα θεωρούσαν αυτά τα block ως γνήσια, ενώ οι ενημερωμένοι κόμβοι θα άλλαζαν το μέγεθος του block και θα συνέχιζαν να συναλλάσσονται ως συνήθως [9].

2.8.2 Hard Forks

Στην περίπτωση των hard forks, όλοι οι κόμβοι πρέπει να ενημερωθούν στο αναθεωρημένο πρωτόκολλο για να μπορούν να συνεχίσουν να πραγματοποιούν συναλλαγές [9].

2.10 Διάσημα Δίκτυα Blockchain

2.10.1 Bitcoin

Το πρώτο ηλεκτρονικό νόμισμα που πέτυχε εμπορική επιτυχία. Ο εφευρέτης του, Satoshi Nakamoto, πρότεινε μια λύση στο πρόβλημα διπλής δαπάνης (double-spending) χρησιμοποιώντας ένα δίκτυο peer-to-peer, μια αλυσίδα από block συναλλαγών χρονικά σφραγισμένες και αξιοποιώντας τον proof-of-work αλγόριθμο για τον μηχανισμό συναίνεσης [2]. Το Bitcoin ήταν ο καταλύτης για την εξάπλωση της τεχνολογίας blockchain προς το ευρύ κοινό, βασιζόμενο όμως σε ιδέες που εκτείνονται δεκαετίες πίσω [1].

2.10.2 Ethereum

Το 2013 ο Vitalik Buterin σχεδίασε το Ethereum [4], η πρόθεση του Ethereum ήταν να συγχωνευθεί και να βελτιώσει τις έννοιες του scripting, altcoins και μετα-πρωτόκολλα αλυσίδας, επιτρέποντας στους προγραμματιστές να δημιουργούν

αυθαίρετες εφαρμογές βασισμένες σε συναίνεση που έχουν την επεκτασιμότητα, τυποποίηση, πληρότητα χαρακτηριστικών, ευκολία ανάπτυξης και διαλειτουργικότητα. Για να επιτευχθούν αυτά, το Ethereum σχεδιάστηκε με μια Turing-complete ενσωματωμένη γλώσσα προγραμματισμού, επιτρέποντας σε οποιονδήποτε να γράψει smart contracts και αποκεντρωμένες εφαρμογές.

Το Ethereum έχει εφαρμόσει τέλη συναλλαγής (gas) σαν ανταμοιβή για τους κόμβους που δημοσίευσαν ένα block.

Τον Σεπτέμβριο 2022, το Ethereum υιοθέτησε τον αλγόριθμο συναίνεσης απόδειξης συμμετοχής [21].

2.10.3 Hyperledger fabric

Η IBM Blockchain πλατφόρμα χτίστηκε γύρω από το Hyperledger Fabric [29]. Προσφέρει μια νέα υπηρεσία, blockchain-as-a-service (BaaS), χρησιμοποιείται σε ιδιωτικά blockchain, είναι προσαρμόσιμο σε πολλά επίπεδα αρχιτεκτονικής του blockchain και υποστηρίζεται από εκατοντάδες οργανισμούς που προωθούν τις δυνατότητες του Fabric [22].

2.11 Ψηφιακά Πορτοφόλια

Τα ψηφιακά πορτοφόλια είναι εφαρμογές που μπορούν να διαχειριστούν διαφόρων ειδών νομισμάτων. Θα εστιάσουμε στα αποκεντρωμένα ψηφιακά πορτοφόλια, οπού διαφέρουν από τα πιο παραδοσιακά συστήματα τραπεζών που μεταφέρουν παραστατικά χρήματα εγγεγραμμένα σε μια πιστωτική κάρτα. Κάθε συναλλαγή στα αποκεντρωμένα ψηφιακά πορτοφόλια καταγράφεται και αποθηκεύεται στο blockchain, χρησιμοποιούν κλειδιά συμμετρικής κρυπτογραφίας και αναγνωρίζονται από μια ψηφιακή διεύθυνση.

Κεφάλαιο 3
Smart Contracts

3.1 Εισαγωγή

Η πρώτη αναφορά στον όρο smart contract (έξυπνα συμβόλαια) αναφέρθηκε από τον Nick Szabo το 1994 [23], εξήγησε ότι ένα έξυπνο συμβόλαιο είναι ένα μηχανογραφημένο πρωτόκολλο συναλλαγής που εκτελεί τους όρους μιας σύμβασης. Ο σκοπός του ήταν να πληρούν τους κοινούς συμβατικούς όρους, χωρίς μεσάζοντες και μειωμένα κόστη συναλλαγών και λειτουργείας. Η άνοδος της δημοτικότητας του Ethereum επέκτεινε την ευρεία γνώση και την λειτουργικότητα των smart contracts, ήταν ένα από τα πιο σημαντικά άλματα στο ιστορικό των smart contract [24]. Το Ethereum είναι το πιο δημοφιλές blockchain για την ανάπτυξη smart contract [25].

3.2 Smart Contracts στο Blockchain

Το πρόγραμμα είναι αμετάβλητο και επαληθευμένο χρησιμοποιώντας κρυπτογραφικά υπογεγραμμένες συναλλαγές, διασφαλίζοντας έτσι την αξιοπιστία του [9]. Το κύριο χαρακτηριστικό που κληρονομεί από την τεχνολογία blockchain είναι η αποκεντρωμένη φύση του. Η απουσία αξιόπιστου τρίτου μέρους μειώνει το κόστος συναλλαγής και την αρχή που επιβάλλουν οι κεντρικοί φορείς. Μόλις επιτευχθεί μια συγκεκριμένη προκαθορισμένη κατάσταση του blockchain, η προγραμματισμένη κατάσταση και η ροή των γεγονότων προορίζονται να την πραγματοποίησή τους. Μετά την αποδοχή από όλους τους συμμετέχοντες στο δίκτυο blockchain, η ακριβής κατάσταση θα καθοριστεί στο smart contract [24].

3.3 Τοποθέτηση και Εκτέλεση

Ένα έξυπνο συμβόλαιο έχει υπόλοιπο, ιδιωτικό χώρο αποθήκευσης και εκτελέσιμο κώδικα. Η κατάσταση της σύμβασης αποτελείται από τον χώρο αποθήκευσης και το υπόλοιπο του συμβολαίου. Η κατάσταση αποθηκεύεται στο blockchain και ενημερώνεται κάθε φορά που ενεργοποιείται το smart contract [25].

Η αποστολή μιας συναλλαγής στη διεύθυνση του συμβολαίου κατά την περιγραφή της συναρτήσεις που θα εκτελεστεί δεδομένου ενός συνόλου ορισμών που απαιτείται για την εκτέλεση. Αυτές οι συναρτήσεις, με τη σειρά τους, μπορούν να καλέσουν άλλα smart contracts εάν έχουν προγραμματιστεί για να το κάνουν [25].

Υπάρχουν πολλές τεχνικές διασύνδεσης ή επικοινωνίας με τα smart contracts, πολλά από αυτά είναι δημόσια και ο καθένας μπορεί να αλληλεπίδραση με αυτά, αλλά χρειάζονται αξιόπιστες πληροφορίες από κάποιο oracle, θα επικεντρωθούμε στην αργότερη περίπτωση στο *Κεφάλαιο 5*.

Κεφάλαιο 4
Tokens

4.1 Εισαγωγή

Τα κρυπτογραφικά tokens είναι προγραμματιζόμενες ψηφιακές μονάδες αξίας που αποθηκεύεται σε ένα blockchain ή σε άλλες τεχνολογίες κατανεμημένου καθολικού. Αξιοποιώντας τις δυνατότητες που παρέχονται από την πλατφόρμα στην οποία βασίζεται το περιουσιακό στοιχείο, η αναπαράσταση περιουσιακών στοιχείων ως tokens επιτρέπει στο δίκτυο blockchain να δημιουργήσει τη μοναδική κατάσταση και την ιδιοκτησία ενός αντικειμένου, καθώς και τη μεταβίβαση της ιδιοκτησίας. Εφόσον το καθολικό είναι ασφαλές, το περιουσιακό στοιχείο είναι αμετάβλητο και δεν μπορεί να μεταφερθεί χωρίς τη συγκατάθεση του ιδιοκτήτη [26].

4.2 Κατηγοριοποίηση των Tokens

Τα tokens μπορούν να διαφοροποιηθούν με βάση τα εναλλαξιμότητα τους.

- Τα **ανταλλάξιμα tokens** (fungible) έχουν την ίδια αξία με άλλα tokens της ίδιας τάξεις.
- **Μη ανταλλάξιμα tokens** (non-fungible ή NFT), τα tokens στην συγκεκριμένη κατηγορία δεν είναι εναλλάξιμα, έχουν την ίδια έννοια με περιουσιακά στοιχεία πραγματικού κόσμου όπως είδη τέχνης, περιουσίες, εισιτήρια κλπ.
- Τα **υβριδικά tokens** συνδυάζουν ανταλλάξιμα και μη ανταλλάξιμα χαρακτηριστικά, για παράδειγμα μπορεί να περιγράφεται σαν ένα ανταλλάξιμο περιουσιακό στοιχείο, όπως μια συγκεκριμένη πτήση αεροπορικής εταιρείας, και ένα μη ανταλλάξιμο περιουσιακό στοιχείο, όπως μια συγκεκριμένη θέση σε αυτή τη πτήση.

4.3 Χαρακτηριστικά των Tokens

Μερικές βασικά χαρακτηριστικά των tokens είναι τα εξής:

- **Κρυπτονομίσματα** (cryptocurrencies): χρησιμοποιούνται σαν τα περιουσιακά στοιχεία σε ένα blockchain δίκτυο.

- **Χρηστικά tokens:** είναι προγραμματιζόμενα περιουσιακά στοιχεία που παρέχουν στους χρήστες πρόσβαση σε ένα προϊόν ή μια υπηρεσία στο μέλλον.

- **Tokens πρωτοκόλλου:** ενθαρρύνουν και υποστηρίζουν την εκτέλεση λειτουργιών ενός συστήματος ή μιας εφαρμογής.

- **Tokens πληροφοριών:** είναι αρχεία υγειονομικής περίθαλψης, πνευματική ιδιοκτησία ή τίτλοι.

- **Μεταβιβάσιμα/μη μεταβιβάσιμα tokens**: είναι η ικανότητα ή η αδυναμία μεταβίβασης της ιδιοκτησίας ενός token μετά την έκδοσή του.

- **Υποδιαιρέσιμα tokens**: είναι τα δεκαδικά ψηφία στα οποία μπορεί να υποδιαιρεθεί ένα token.

- **Μοναδικά tokens**: μπορεί να υπάρχει μόνο μια ποσότητα ενός token ανά άτομο.

- **Mintable**: Η δυνατότητα δημιουργίας νέων token του ιδίου είδους.

- **Burnable**: Η δυνατότητα αφαίρεσης token από την κυκλοφορία.

4.4 Token Standards

Για την παροχή μιας κοινής μορφής token στο Ethereum, παρουσιάστηκε μια τυπική διεπαφή, αναφερομένη ως ERC (Ethereum Request for Comments).

4.4.1 ERC 20

Το πρότυπο ERC 20 είναι το πιο συχνά χρησιμοποιούμενο πρότυπο, παρέχοντας βασικές λειτουργίες για μεταφορές token και επιτρέποντας την πιστοποίησή τους, ώστε να μπορούν να δαπανηθούν από άλλο τρίτο μέρος της αλυσίδας.

4.4.2 ERC 721

Το πρότυπο ERC 721 καθορίζει τον τρόπο δημιουργίας μη ανταλλάξιμων ή μοναδικών token στο Ethereum blockchain, επιτρέποντας την παρακολούθηση των token. Η ιδιοκτησία κάθε περιουσιακού στοιχείου πρέπει να παρακολουθείται μεμονωμένα και ατομικά. Δεν περιορίζεται η προσθήκη ή η αφαίρεση επιπλέον λειτουργιών, πέρα από τις ελάχιστες απαιτήσεις διεπαφής που πρέπει να εφαρμόσει ένα smart contract για την λειτουργία του προτύπου.

4.4.3 ERC 777

Το πρότυπο ERC 777 καθορίζει βελτιωμένες δυνατότητες αλληλεπίδρασης token που είναι backwards compatible με το ERC 20. Ορίζει τους χειριστές για τη μεταφορά token για λογαριασμό μιας άλλης διεύθυνσης, καθώς και hooks για αποστολή και λήψη, για να δώσουν στους κατόχους διακριτικών πρόσθετο έλεγχο των token τους.

4.4.4 ERC 1155

Το ERC 1155 είναι ένα ανώτερο πρότυπο για token που επιτρέπει να δημιουργούνται ανταλλάξιμα, μη ανταλλάξιμα και ημι-ανταλλάξιμα tokens όλα μέσω ενός ενιαίου smart contract.

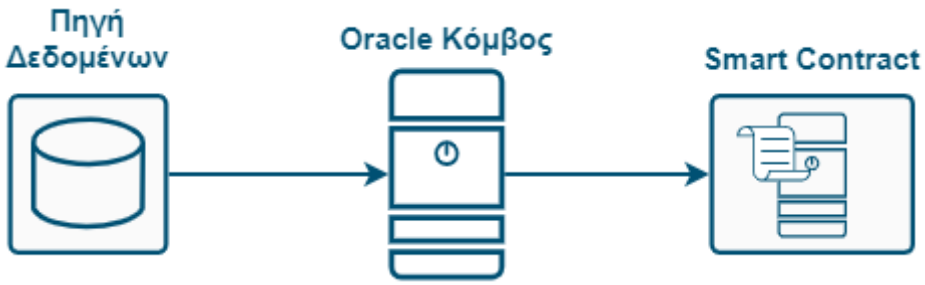
Κεφάλαιο 5
Oracles

5.1 Εισαγωγή

Τα oracles στο blockchain είναι υπηρεσίες που παρέχουν πρόσβαση εξωτερικών δεδομένων σε smart contracts. Λειτουργούν ως σημείο σύνδεσης μεταξύ του εξωτερικού κόσμου και του blockchain. Ένα oracle δεν είναι το ίδιο πηγή δεδομένων, αλλά το επίπεδο που αναζητά, επικυρώνει και πιστοποιεί άλλες πηγές δεδομένων πριν μεταδώσει αυτές τις πληροφορίες στο smart contract [27].

5.2 Το Oracle Οικοσύστημα

Το oracle οικοσύστημα περιλαμβάνει από τα ακόλουθα τρία μέρη [28]:



Εικόνα 1.5.1: Oracle οικοσύστημα. Δημιουργημένο από τον συγγραφέα.

- Την **πηγή δεδομένων,** αυτή είναι η εκτός αλυσίδας τοποθεσία όπου διατηρούνται τα δεδομένα. Δεν είναι απαραίτητο να χρησιμοποιηθούν από την αποκεντρωμένη εφαρμογή.
- Τον **oracle κόμβο**, παίρνει δεδομένα από μια πηγή δεδομένων και τα στέλνει σε ένα smart contract, ώστε να μπορούν να χρησιμοποιηθούν από αυτό.
- Το **smart contract**, αυτό το μέρος περιλαμβάνει τον κώδικα που καθορίζει τον τρόπο διαχείρισης των δεδομένων. Συνήθως έχει προκαθορισμένα πρότυπα για την αποδοχή ή την απόρριψη δεδομένων.

5.3 Τύποι Oracle

5.3.1 Oracles Λογισμικού (Software Oracles)

Τα Software Oracles συνδέονται με πηγές δεδομένων στο διαδίκτυο και τα στέλνουν στο smart contract. Τα δεδομένα ενδέχεται να προέρχονται από διαδικτυακές βάσεις δεδομένων, διακομιστές, ιστοσελίδες ή οποιαδήποτε άλλη πηγή δεδομένων στο Διαδίκτυο.

5.3.2 Oracles Υλικού (Hardware Oracles)

Ορισμένα smart contracts χρειάζονται να αλληλεπιδρούν με τον έξω κόσμο. Τα hardware oracles συλλέγουν δεδομένα από τον φυσικό κόσμο και τα καθιστούν διαθέσιμα σε smart contracts. Τέτοια δεδομένα μπορεί να κοινοποιούνται από ηλεκτρονικούς αισθητήρες, IoT, σαρωτές barcodes, RFID, ρομπότ και άλλες συσκευές ανάγνωσης δεδομένων.

5.3.3 Ανθρώπινα Oracles (Human Oracles)

Τα ανθρώπινα oracles είναι άτομα με εξειδικευμένες γνώσεις/δεξιότητες σε έναν συγκεκριμένο τομέα μπορούν μερικές φορές να λειτουργήσουν ως oracles. Μπορούν να κάνουν έρευνα και να αξιολογήσουν την εγκυρότητα των πληροφοριών από πολλές πηγές πριν τα στείλουν σε smart contracts. Τα ανθρώπινα oracles δεν είναι μόνο ικανά να μεταδίδουν προβλέψιμα δεδομένα, αλλά και να ανταποκρίνονται σε τυχαία αιτήματα, κάτι που ένας υπολογιστής μπορεί να δυσκολευτεί.

5.3.4 Υπολογιστικά Oracles (Computation Oracles)

Τα oracles μπορούν επίσης να χρησιμοποιηθούν για τη διεξαγωγή αυθαίρετων υπολογισμών εκτός αλυσίδας, κάτι που είναι ιδιαίτερα ευεργετικό δεδομένου του block gas limit του Ethereum και του σχετικά υψηλού κόστους υπολογισμού.

5.3.5 Εισερχόμενα/Εξερχόμενα Oracles (Inbound/Outbound Oracles)

Τα inbound oracles παρέχουν δεδομένα από εξωτερικές πηγές σε smart contracts, ενώ τα outbound oracles μεταφέρουν δεδομένα από smart contracts στον έξω κόσμο.

5.3.6 Oracles Ξεχωριστά σε Συμβόλαιο (Contract-specific Oracles)

Ένα ξεχωριστό σε συμβόλαιο oracle χρησιμοποιείται μόνο από ένα smart contract. Αυτό σημαίνει ότι εάν πρόκειται να αναπτυχθούν πολλά smart contracts, πρέπει να κατασκευαστεί ένας αντίστοιχος αριθμός από oracles.

5.3.7 Oracles Βάσει Συναίνεσης (Consensus-based Oracles)

Τα βάσει συναίνεσης oracles, σε αντίθεση με τα Software Oracles, δεν βασίζονται σε μια μόνο πηγή αλλά έναν αριθμό από αποκεντρωμένα oracles. Τα oracles που βασίζονται στη συναίνεση είναι πιο αργά, λόγο της χρονικής διάρκειας που απαιτείται για να φτάσουν σε συναίνεση.

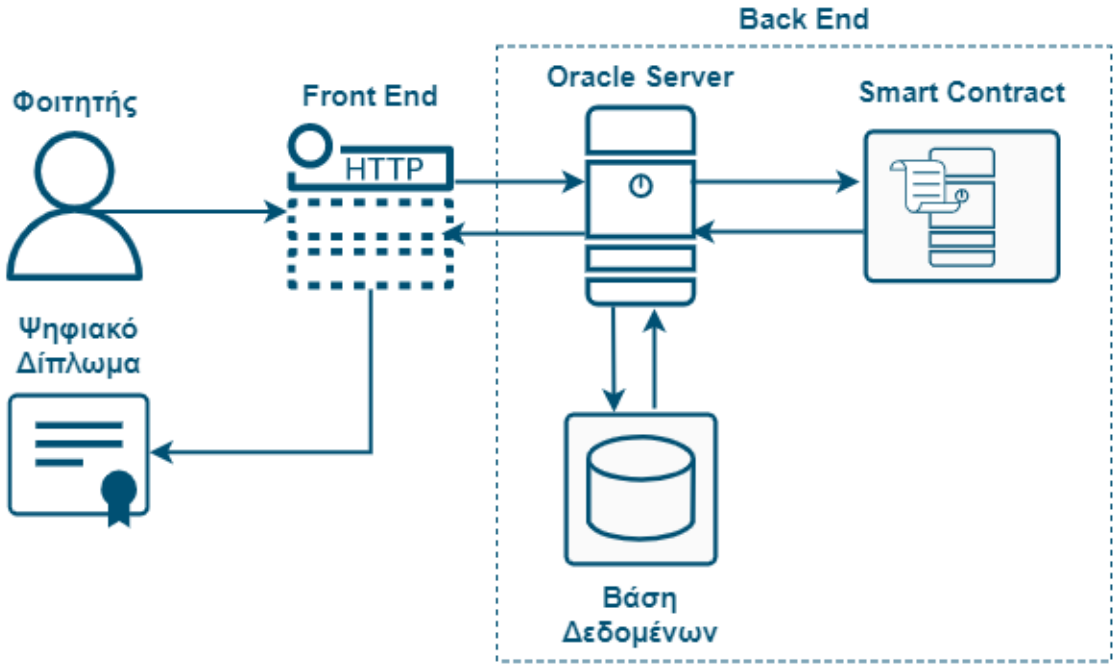
Πρακτικό Μέρος

Κεφάλαιο 1
Εισαγωγή

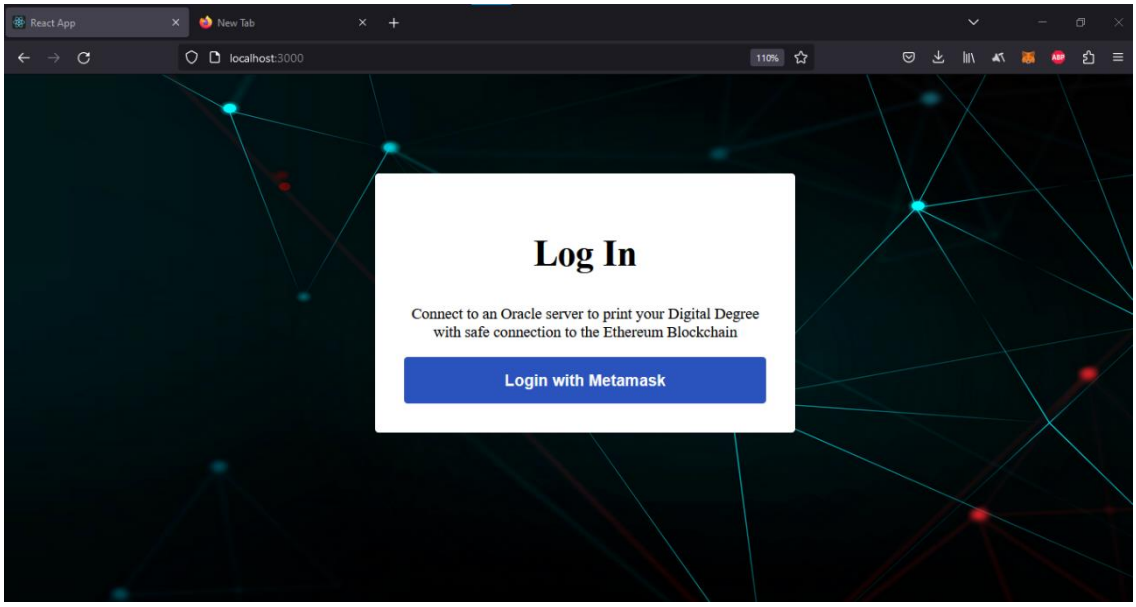
1.1 Εισαγωγή

Σε αυτό το κομμάτι της διπλωματικής εργασίας θα διατρίβουμε στο τεχνικό μέρος της εφαρμογής. Ο σκοπός της παρούσας διπλωματικής εργασίας είναι απόδειξη της ιδέας, ότι ένα ακαδημαϊκό σύστημα ενός ιδρύματος μπορεί να χρησιμοποιήσει ένα ERC721 token δεσμευμένο σε μια ψηφιακή διεύθυνση ενός φοιτητή μέσω του Ethereum blockchain για να εκτυπώσει αυτόματα ένα αμετάβλητο ψηφιακό δίπλωμα, χτίζοντας μια ακαδημαϊκή εφαρμογή γύρω από αυτή την ιδέα.

1.2 Περιγραφή Εφαρμογής

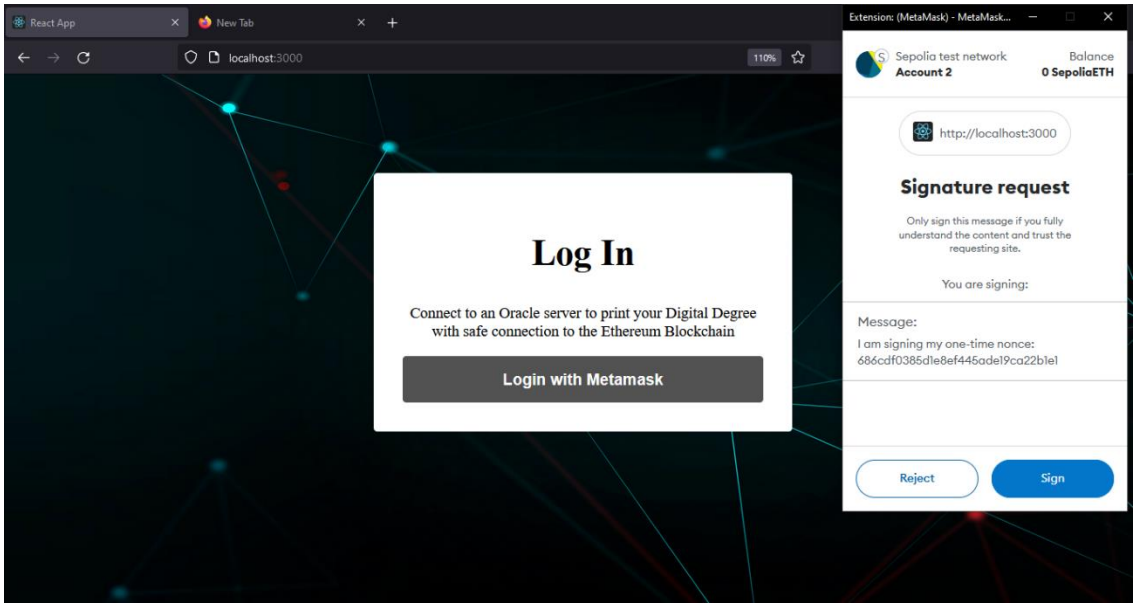


Εικόνα 2.1.1: Διάγραμμα λειτουργιών της εφαρμογής. Δημιουργημένο από τον συγγραφέα.



Εικόνα 2.1.2: Σελίδα σύνδεσης της εφαρμογής. Δημιουργημένο από τον συγγραφέα.

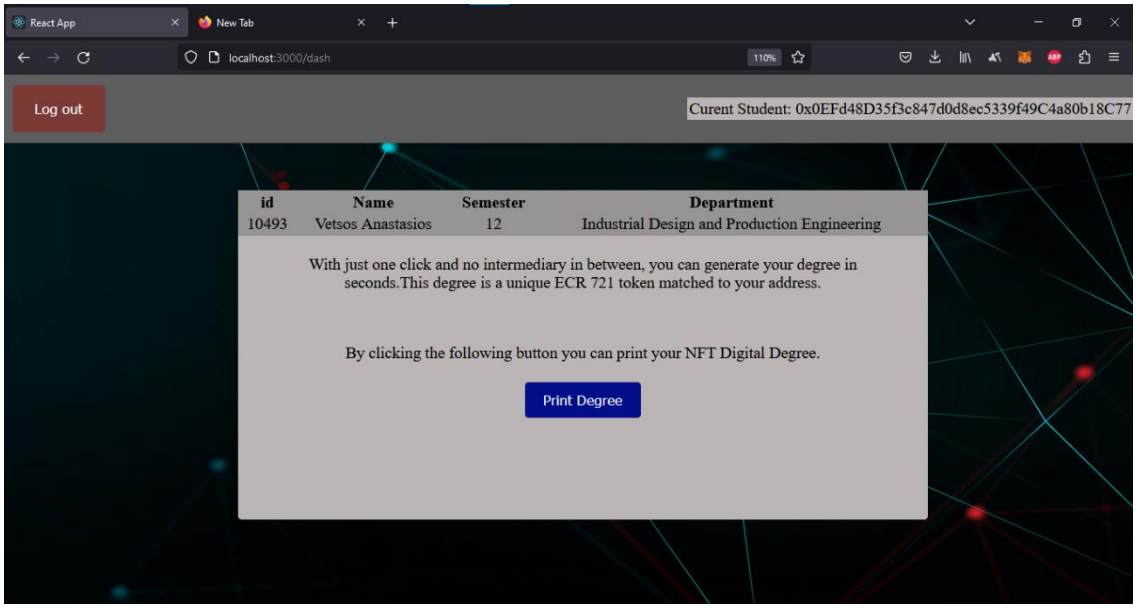
Αυτή είναι η αρχική σελίδα της εφαρμογής, όταν ο φοιτητής πατήσει το κουμπί *"Log in with MetaMask"* σε περίπτωση που ο φοιτητής δεν είναι ήδη συνδεδεμένος στο MetaMask θα εμφανιστεί ένα παράθυρο που ζητάει στον φοιτητή για κωδικό του MetaMask του που κατέχει.



Εικόνα 2.1.3: Αίτημα υπογραφής μηνύματος από το MetaMask. Δημιουργημένο από τον συγγραφέα.

Έπειτα θα εμφανιστεί ένα παρόμοιο παράθυρο που ζητάει από τον φοιτητή να υπογράψει ένα μήνυμα (η διαδικασία γίνεται αυτόματα με το μυστικό κλειδί του φοιτητή όπου διαχειρίζεται το MetaMask) όπου περιέχει μια τυχαία τιμή nonce που προήρθε από την αντίστοιχη εγγραφή στην βάση δεδομένων.

Με το πάτημα του κουμπιού *"Sign"*, θα σταλθούν στον server η ψηφιακή διεύθυνση του φοιτητή (όπου είναι η διεύθυνση του MetaMask) και η ψηφιακή υπογραφή του μηνύματος. Στην συνέχεια θα λάβει τον χρήστη στη βάση δεδομένων που αντιστοιχεί στη ψηφιακή διεύθυνση και στη συνέχεια φέρνει το σχετικό nonce. Με το nonce, τη διεύθυνση και την υπογραφή, ο server μπορεί να επαληθεύσει κρυπτογραφικά ότι το μήνυμα που περιείχε το nonce υπογράφηκε με επιτυχία από τον χρήστη [54]. Σε αυτή την περίπτωση, ο χρήστης έχει αποδείξει την ιδιοκτησία της διεύθυνσης και μπορούν να δημιουργηθούν δυο JWT (token πρόσβασης και token ανανέωσης). Επιπλέον, δημιουργείται ένα νέο nonce για τον φοιτητή και η αποστολή του token πρόσβασης στο front end.

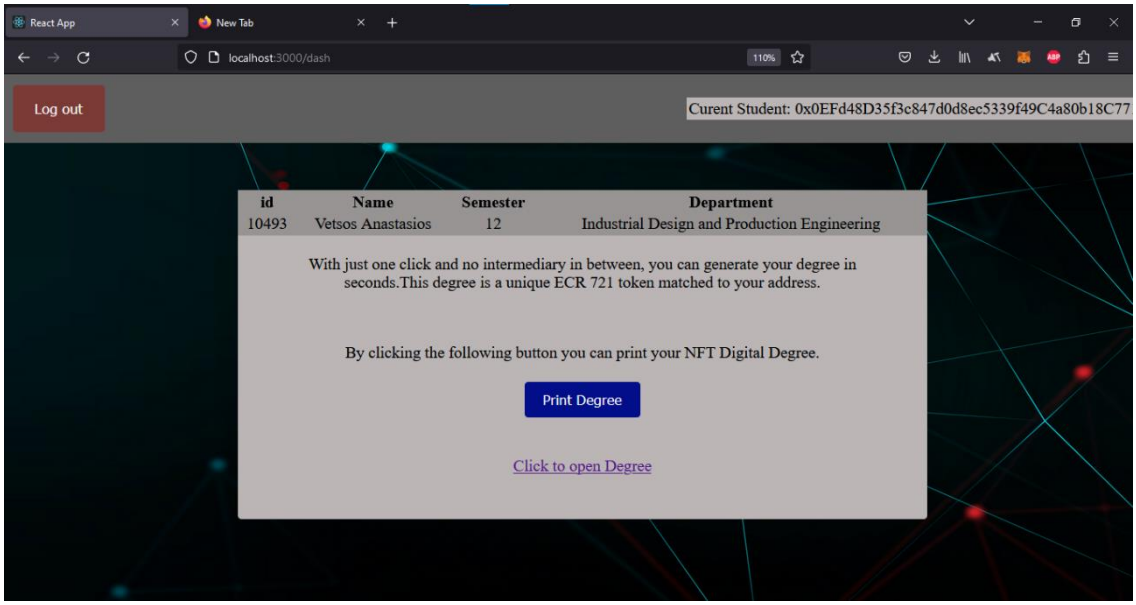


Εικόνα 2.1.4: Αρχική σελίδα της εφαρμογής. Δημιουργημένο από τον συγγραφέα.

Με το token πρόσβασης ο φοιτητής έχει πρόσβαση στην αρχική σελίδα. Εδώ εμφανίζονται κάποια στοιχεία του φοιτητή, ένα *"Log out"* κουμπί και ένα *"Print Degree"* κουμπί.

Αν ο φοιτητής πατήσει το *"Print Degree"* κουμπί θα σταλθεί η διεύθυνσή του στον oracle server (είναι ο ίδιος server, αλλά λόγο της επικοινωνίας με ένα smart contract τον περιγράφουμε έτσι). Τα χαρακτηριστικά του oracle είναι ένα κεντρωμένο inbound και outbound, software και επιπλέων το είναι ένα υπολογιστικό oracle. Πραγματοποιούνται οι έλεγχοι απαιτήσεων αποφοίτησης και ο υπολογισμός του τελικού βαθμού. Αν ο φοιτητής πληροί τις προϋποθέσεις, ο oracle server θα επικοινωνήσει με το smart contract που ελέγχει αν υπάρχει ήδη ένα ERC721 token που αντιστοιχεί στην διεύθυνση του φοιτητή. Αν όχι, κάνει mint ένα και το αντιστοιχεί στην εν λόγω διεύθυνση. Σε κάθε περίπτωση το smart contract θα εκπέμψει το αναγνωριστικό του token πίσω στον oracle server.

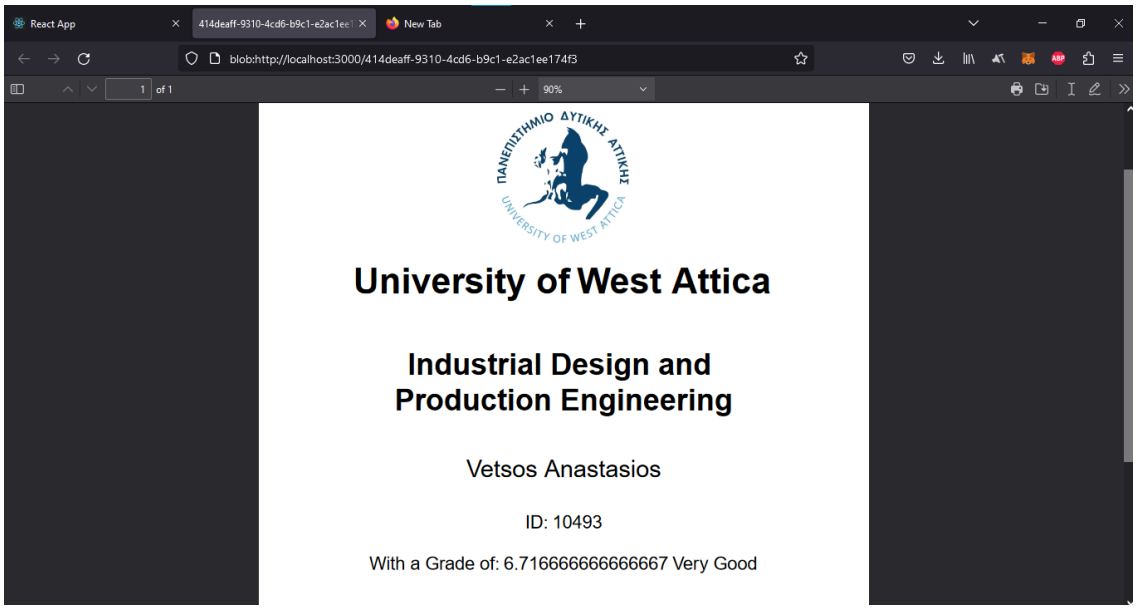
Ύστερα, ο server θα δημιουργήσει το δίπλωμα σαν ένα pdf έγγραφο που αποθηκεύεται στο server με τα απαραίτητα στοιχεία του αναφερόμενα μέσα στο δίπλωμα, το αναγνωριστικό του token σαν όνομα και τέλος στέλνει το όνομα στο front end.



Εικόνα 2.1.5: Σύνδεσμος του διπλώματος. Δημιουργημένο από τον συγγραφέα.

Όταν ολοκληρωθούν οι παραπάνω διαδικασίες θα εμφανιστεί ένας σύνδεσμος *("Click to open Degree"*) που όταν τον πατήσει ο φοιτητής, το pdf έγγραφο θα ανοίξει σε μια

νέα καρτέλα περιήγησης ως απάντηση από το server και ο φοιτητής δεν χρειάζεται να κατεβάσει κάτι υποχρεωτικά.



Εικόνα 2.1.6: Δίπλωμα σαν pdf Έγγραφο. Δημιουργημένο από τον συγγραφέα.

Ο φοιτητής έχει την δυνατότητα να κλείσει την καρτέλα και να την ξανανοίξει εξακολουθώντας να είναι συνδεδεμένος για έξη ώρες, μετά από αυτή την διάρκεια θα χρειαστεί να επανασυνδεθεί.

Αν ο φοιτητής θέλει να αποσυνδεθεί μπορεί να πατήσει το *"Log out"* κουμπί, όπου καθαρίζει όλα τα cookies στο front end και server, επίσης διαγράφει το pdf έγγραφο που είναι αποθηκευμένο στο server.

Κεφάλαιο 2
Ανάλυση Εφαρμογής

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα διευκρινίσουμε το τεχνικό πλαίσιο της παρούσας εφαρμογής, περιλαμβάνοντας στοιχεία λογισμικού και πακέτα δεδομένων που αποτελούν την ουσία αυτού του συστήματος, καθώς και μια ολοκληρωμένη ανάλυση της υποκείμενης αρχιτεκτονικής του συστήματος.

2.2 Λογισμικά και Πακέτα

2.2.1 Node

Το Node.js είναι ένα JavaScript runtime environment. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί για την εκτέλεση κώδικα JavaScript χωρίς την ανάγκη για ένα browser και έχει σχεδιαστεί για την δημιουργία κλιμακωτών εφαρμογών δικτύου. Πολλές άλλες βιβλιοθήκες έχουν σχεδιαστεί για να εκτελούνται σε αυτό το περιβάλλον.

2.2.2 Solidity

Η Solidity είναι μια αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού που χρησιμοποιείται για τη δημιουργία smart contracts στο Ethereum blockchain [32].

Τα συμβόλαια στην Solidity είναι ανάλογα με τις τάξεις σε αντικειμενοστραφείς γλώσσες προγραμματισμού. Τα συμβόλαια μπορούν επίσης να κληρονομήσουν από άλλα συμβόλαια. Υπάρχουν και άλλοι τύποι συμβολαίων, γνωστοί ως βιβλιοθήκες και διεπαφές.

2.2.3 OpenZeppelin

Το Open Zeppelin είναι μια βιβλιοθήκη ανάπτυξης smart contracts. Υποστηρίζει πρότυπα όπως το ERC20 και το ERC721, καθώς και έναν ευέλικτο μηχανισμό αδειοδότησης (permissioning), βασισμένο σε ρόλους και επαναχρησιμοποιήσιμα κομμάτια της Solidity για την κατασκευή προσαρμοσμένων συμβολαίων και εξελιγμένων αποκεντρωμένων συστημάτων.

2.2.4 React

Η React είναι μια βιβλιοθήκη που επιτρέπει την δημιουργία διεπαφών αλληλεπίδρασης ανθρώπου-μηχανής από ανεξάρτητα components. Τα components είναι συναρτήσεις JavaScript που λαμβάνουν δεδομένα και επιστρέφουν ό,τι πρέπει να εμφανίζεται στην οθόνη. Υπάρχει η δυνατότητα να σταλθούν νέα δεδομένα ως αντίδραση σε μια αλληλεπίδραση μεταξύ ανθρώπου και μηχανής, σε αυτή την περίπτωση η React θα αλλάξει την οθόνη ώστε να αντικατοπτρίζει τα νέα δεδομένα.

2.2.5 Redux Toolkit

Το Redux είναι μια βιβλιοθήκη για τη διατήρηση και την ενημέρωση καταστάσεων εφαρμογής (states) μέσω συμβάντων (events) γνωστών ως δράσεις (actions). Το Redux χρησιμοποιεί κανόνες που εγγυούνται την τροποποίηση των πληροφοριών με προβλέψιμο τρόπο. Λειτουργεί ως κεντρικό αποθετήριο για states που μπορούν να χρησιμοποιηθούν σε όλη την εφαρμογή.

Το **Redux Toolkit** είναι ένα πακέτο που αναδιπλώνεται πάνω από το κύριο πακέτο Redux και περιλαμβάνει μεθόδους API και κοινά dependencies που απαιτούνται για την ανάπτυξη μιας εφαρμογής React/Redux. Είναι ο τυπικός τρόπος έκφρασης της λογικής Redux με απλούστερες περιπτώσεις χρήσης.

2.2.6 MongoDB Atlas

Το MongoDB Atlas είναι μια βάση δεδομένων εγγράφων NoSQL χωρίς σχήματα. Σε αυτή την βάση δεδομένων μπορούν να αποθηκευτούν αντικείμενα JSON. Λόγω της ευέλικτης δομής της, σε αντίθεση με τις βάσεις δεδομένων SQL, μπορούν να αποθηκευτούν έγγραφα με διαφορετικές δομές. Αυτό είναι ένα πλεονέκτημα της χρήσης της NoSQL, καθώς μειώνει την πολυπλοκότητα της ανάπτυξης και επιταχύνει την ανάπτυξη εφαρμογών.

2.2.7 Mongoose

Το Mongoose είναι μια MongoDB και Node.js ODM (Object Data Modeling) βιβλιοθήκη. Χειρίζεται συσχετίσεις δεδομένων, κάνει επικύρωση σημάτων και χρησιμοποιείται για τη μετάφραση μεταξύ αντικειμένων στον κώδικα και της αναπαράστασής τους στο MongoDB.

2.2.8 Ethers

Το Ethers.js είναι μια βιβλιοθήκη γενικής χρήσης για την αλληλεπίδραση με το οικοσύστημα του Ethereum blockchain. Οι λειτουργίες της παρέχουν μια πλήρη λειτουργικότητα για όλες τις ανάγκες στο Ethereum, όπως εισαγωγή και εξαγωγή ψηφιακών πορτοφολιών και ασφαλή διατήρηση των ιδιωτικών κλειδιών στο front end.

2.2.9 Hardhat

Το Hardhat.js είναι ένα περιβάλλον ανάπτυξης λογισμικού για smart contracts και στο Ethereum. Αποτελείται από πολλά στοιχεία που συνεργάζονται για να προσφέρουν

ένα πλήρες περιβάλλον ανάπτυξης για την επεξεργασία, την κατασκευή, την ανάπτυξη των smart contracts και τον εντοπισμό σφαλμάτων.

Το Hardhat περιλαμβάνει επίσης το δίκτυο Hardhat, έναν τοπικό δοκιμαστικό κόμβο δικτύου Ethereum. Δίνει τη δυνατότητα για την ανάπτυξη smart contracts, την εκτέλεση δοκιμών και διόρθωση κώδικα σε έναν τοπικό υπολογιστή.

2.2.10 Alchemy

Το Alchemy είναι μια blockchain πλατφόρμα εργαλείων για προγραμματιστές. Θα χρησιμοποιήσουμε το Alchemy Supernode, το οποίο είναι ένα από τα πιο συχνά χρησιμοποιούμενα Ethereum APIs . Προσφέρει μια υποδομή για την εκτέλεση όλων των δραστηριοτήτων ενός κόμβου με βελτιωμένη διαθεσιμότητα και χρόνο λειτουργίας.

2.2.11 Sepolia

Το Sepolia είναι ένα δοκιμαστικό δίκτυο (testnet) απόδειξης συμμετοχής (PoS) που κατασκευάστηκε και συντηρείται από τους προγραμματιστές του Ethereum core απο τον Οκτώβριο του 2021.

Τα δοκιμαστικά δίκτυα είναι blockchains που προορίζονται να μιμηθούν τη λειτουργία ενός κύριου δικτύου (mainnet) αλλά βρίσκονται σε ξεχωριστό καθολικό. Αυτά τα testnets επιτρέπουν στους προγραμματιστές να δοκιμάσουν τις εφαρμογές και τα smart contracts τους σε ένα περιβάλλον χωρίς κίνδυνο πριν τα τοποθετήσουν στο mainnet του Ethereum.

Σε σύγκριση με άλλα testnets όπως το Goerli, το συνολικό ποσό των testnet tokens του Sepolia είναι απεριόριστο, πράγμα που σημαίνει ότι οι προγραμματιστές που

χρησιμοποιούν το Sepolia είναι λιγότερο πιθανό να υποφέρουν από έλλειψη testnet tokens.

2.2.12 Etherscan

Το Etherscan είναι το κορυφαίο εργαλείο εξερεύνησης και αναζήτησης blocks στο Ethereum.

Θα το χρησιμοποιήσουμε ως μέσο προβολής μιας αυθεντικής αναπαράστασης της δομής ενός ενός block.

2.2.13 web3

Το web3 είναι μια συλλογή βιβλιοθηκών που επιτρέπουν την αλληλεπίδραση με έναν τοπικό ή απομακρυσμένο κόμβο Ethereum.

2.2.14 MetaMask

Το MetalMask είναι ένα κρυπτογραφικό ψηφιακό πορτοφόλι. Το MetalMask API μπορεί να ζητήσει τους λογαριασμούς Ethereum των χρηστών, να ανακτήσει δεδομένα από συνδεδεμένα blockchains, να ζητήσει στους χρήστες να υπογράψουν μηνύματα και συναλλαγές και να πραγματοποιήσουν άλλες λειτουργίες.

2.2.15 eth-sig-util

Ένα σύνολο λειτουργιών για την υπογραφή και την επικύρωση δεδομένων χρησιμοποιώντας Ethereum κλειδιά.

2.2.16 Express

Το Express είναι ένα πλαίσιο εφαρμογών ιστού για το Node.js που παρέχει μια ισχυρή συλλογή λειτουργιών για εφαρμογές ιστού, φορητές και API εφαρμογές. Το Express προσθέτει ένα λεπτό στρώμα βασικών λειτουργιών εφαρμογών ιστού διατηρώντας παράλληλα τον χαρακτηριστικά του Node.js.

2.2.17 dotenv

Το dotenv είναι εάν zero-dependency module που φορτώνει μεταβλητές περιβάλλοντος από ένα .env αρχείο στο process.env. Η τεχνική Twelve-Factor App χρησιμοποιείται για την αποθήκευση ρυθμίσεων στο περιβάλλον ξεχωριστά από τον κώδικα.

2.2.18 js-cookie

Ένα JavaScript API για το χειρισμό των cookies του browser.

2.2.19 JSON Web Token

Το JSON Web Token είναι ένα ανοιχτό πρότυπο που προσφέρει μια συνοπτική και αυτόνομη μέθοδο για την ασφαλή παράδοση πληροφοριών ως ένα αντικείμενο JSON. Τα JWT μπορούν να υπογραφούν είτε με συμμετρικές είτε με ασύμμετρες κρυπτογραφικές μεθόδους. Το JWTs μπορεί να χρησιμοποιηθεί ως σύστημα ελέγχου ταυτότητας καθώς και για την ανταλλαγή δεδομένων.

2.2.20 cors

Το cors είναι ένας πακέτο Node.js που μπορεί να χρησιμοποιηθεί για να ενεργοποιήσει το Cross-Origin Resource Sharing (CORS) με διάφορες επιλογές.

Το CORS είναι ένας μηχανισμός που βασίζεται σε κεφαλίδες HTTP και επιτρέπει σε έναν server να προορίσει προελεύσεις (domains, schemes, ports).

2.2.21 PDF kit

Το PDF Kit είναι μια βιβλιοθήκη δημιουργίας εγγράφων PDF για το Node.js.

2.3 Back End

Το back end κάθε συστήματος παρέχει λειτουργικότητα από την πλευρά του server, διαχειρίζει την ενσωμάτωση με εξαρτήματα του front-end και βάσεις δεδομένων. Είναι ένα κρίσιμο στοιχείο της ανάπτυξης εφαρμογών, καθώς χειρίζεται αιτήματα και παρέχει πληροφορίες στους χρήστες.

2.3.1 OracleContract

Το συμβόλαιο είναι ένα ERC721 εισαγόμενο από την τοπικά ενσωματωμένη openzeppelin βιβλιοθήκη στο server, όπου έχει αφαιρεθεί η transfer συνάρτηση επειδή δεν θέλουμε τα tokens να μπορούν να μεταφερθούν σε άλλες διευθύνσεις.

Θέτουμε τις ακόλουθες μεταβλητές που θα χρησιμοποιήσουμε:

- Την **διεύθυνση** του oracle.
- Μια **Boolean τιμή** (*canGraduate*) που περιγράφει την κατάσταση πληρότητας αποφοίτησης του φοιτητή.
- Μια συνάρτηση που λειτουργεί ως **μετρητής των αναγνωριστικών** των ERC721 tokens.
- Ένα event που θα χρησιμοποιηθεί για να **εκπέμψει το αναγνωριστικό** token στο oracle.

Στην τοποθέτηση (deploy) του smart contract θα εκτελεστεί η συνάρτηση κατασκευαστή (contractor) που θέτει την τιμή της μεταβλητής *canGraduate* σε false και την τιμή της διεύθυνσης του oracle με αυτή την διεύθυνση που κάλεσε την τοποθέτηση του στο blockchain.

Θέτουμε ένα modifier που είναι κώδικας που μπορεί να προστεθεί σε μια συνάρτηση, με την προϋπόθεση ότι μόνο η διεύθυνση oracle μπορεί να καλέσει την συνάρτηση όπου προστέθηκε.

Καλούμε την συνάρτηση *graduationCheck* για να θέσουμε την τιμή της *canGraduate* μεταβλητής με την boolean παράμετρο που καλέστηκε.

Η συνάρτηση *safeMint* με απαραίτητη παράμετρο της διεύθυνσης ενός φοιτητή, έχει την προϋπόθεση ότι η *canGraduate* μεταβλητή πρέπει να είναι Boolean τιμή true. Ο ρόλος της είναι να αυξάνει τον μετρητή των αναγνωριστικών των tokens κατά ένα, να αντιστοιχεί την διεύθυνση του φοιτητή με το αναγνωριστικό του token, να θέτει την τιμή της *canGraduate* σε false και να εκπέμπει το αναγνωριστικό του token.

Η συνάρτηση *hasToken* με απαραίτητη παράμετρο της διεύθυνσης ενός φοιτητή, ψάχνει για το αναγνωριστικό του token που αντιστοιχεί στην διεύθυνση της παραμέτρου και εκπέμπει το αναγνωριστικό του token. Σε περίπτωση που δεν υπάρχει αντίστοιχο αναγνωριστικό, εκπέμπει τον αριθμό μηδέν μιας και κανένα αναγνωριστικό δεν μπορεί να έχει αυτή την τιμή, γιατί το αναγνωριστικό της πρώτης εκτέλεσης της *safeMint* συνάρτησης ισούται με ένα.

2.3.6 Deploy

Για να χρησιμοποιηθεί το smart contract πρέπει να τοποθετηθεί πρώτα στο blockchain. Χρησιμοποιώντας την βιβλιοθήκη hardhat, δημιουργούμε ένα instance

(στιγμιότυπο) του smart contract αποθηκευμένο στο server με όνομα *"OracleContract"*. Τοποθετούμε το smart contract με την διεύθυνση του oracle που έχουμε θέσει (σε αυτή τη διεύθυνση διέρχονται όλες οι συναλλαγές και τα τέλη για την επικοινωνία με το smart contract) σε ένα τεστ δικτύου του Ethereum το Sepolia.

Για μια καλύτερη πιθανότητα αποφυγής σφαλμάτων και την προβολή του κώδικα από το Etherscan, έχουμε την επιλογή να το επαληθεύσουμε προγραμματιστικά χρησιμοποιώντας την συνάρτηση run του hardhat.

2.3.2 Server

Στο βασικό αρχείο του server (*server.js*) χρησιμοποιούμε τις βασικές απαιτήσεις για την λειτουργία του server και την επικοινωνία με το front end. Καταγράφουμε κάθε επικοινωνία με τον server σε log αρχεία, ορίζουμε την επιτρεπόμενη προέλευση με το cors, τον χειρισμό των cookies, τα αιτήματα JSON και διάφορους μεθόδους χειρισμού σφαλμάτων.

Υπάρχουν οι εξής δρομολογήσεις (routes) στο server:

- Ρίζας ή root ("/")
- Αυθεντικοποίησης ("/auth")
- Φοιτητών ("/students")
- Oracle ("/oracle")
- Διπλωμάτων ("/degrees")

Μόλις αρχίσει η λειτουργία του, ο server θα προσπαθήσει να συνδεθεί στην προκαθορισμένη βάση δεδομένων (MongoDB) του πανεπιστημίου που περιέχει τα δεδομένα των φοιτητών.

2.3.3 Μοντέλα mongoose

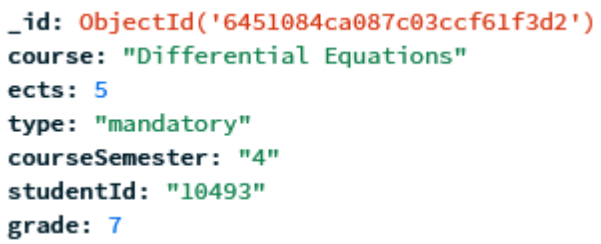
Τα μοντέλα που χρησιμοποιεί η εφαρμογή είναι το ***student*** που ορίζει την δομή των δεδομένων του φοιτητή και το ***course*** που ορίζει την δομή των δεδομένων ενός περασμένου βαθμού ενός φοιτητή και απαραίτητα δεδομένα του μαθήματος, όλα τα πεδία σε αυτά τα μοντέλα είναι υποχρεωτικά.

2.3.3.1 student

Εικόνα 2.1.7: Το μοντέλο student στην πλατφόρμα MongoDB Atlas. Δημιουργημένο από τον συγγραφέα.

- ***address***: Η ψηφιακή διεύθυνση του φοιτητή, τύπου συμβολοσειράς.
- ***id***: Ο αριθμός μητρώου του φοιτητή, τύπου συμβολοσειράς.
- ***name***: Το ονοματεπώνυμο του φοιτητή, τύπου συμβολοσειράς.
- ***department***: Το τμήμα του φοιτητή, τύπου συμβολοσειράς.
- ***semester***: Το τρέχον εξάμηνο του φοιτητή, τύπου νούμερο, με χαμηλότερη πιθανή τιμή 1, ακέραια τιμή.
- ***nonce***: Η τυχαία τιμή nonce μιας χρήσης τύπου συμβολοσειράς.

2.3.3.2 course



Εικόνα 2.1.8: Το μοντέλο course στην πλατφόρμα MongoDB Atlas.Δημιουργημένο από τον συγγραφέα.

- ***course***: Το εν λόγω μάθημα, τύπου συμβολοσειράς.
- ***ects***: Το ευρωπαϊκό σύστημα μεταφοράς και συσσώρευσης ακαδημαϊκών μονάδων, τύπου νούμερο, ακέραια τιμή.
- ***type***: Ο τύπος του μαθήματος, υποχρεωτικό ή κατ' επιλογή, τύπου συμβολοσειράς.
- ***courseSemester***: Το εξάμηνο του μαθήματος, τύπου συμβολοσειράς και ακέραια τιμή μεταξύ 1 με 9.
- ***studentId***: Ο αριθμός μητρώου του μαθητή, τύπου συμβολοσειράς.
- ***grade***: Ο βαθμός του μαθητή στο συγκεκριμένο μάθημα, τύπου νούμερο, μικρότερη τιμή 0 και μέγιστη 10.

2.3.4 Δρομολογήσεις (Routes)

Ορίζουμε διαδρομές και ενδιάμεσα λογισμικά για εισερχόμενα αιτήματα, οι διαδρομές ορίζονται χρησιμοποιώντας συμβολοσειρές και μπορούν να χρησιμοποιήσουν παραμέτρους από τη διεύθυνση URL. Μπορούμε να υπαγορεύσουμε μια μονάδα από το αντίστοιχο αρχείο χειριστών (*controller*), θα επικαλεσθεί ανάλογα με ποια μέθοδο HTTP ζητήθηκε από το front end. Οι μέθοδοι που θα χρησιμοποιήσουμε είναι η GET και POST.

Για να αποτρέψουμε μη εξουσιοδοτημένους χρήστες να επικοινωνούν με τις προστατευμένες διαδρομές (/student, /oracle, /degrees), θα χρησιμοποιήσουμε το

module *verifyJWT*. Ελέγχει εάν η κεφαλίδα αιτήματος περιέχει ένα έγκυρο JWT. Εάν το JWT δεν είναι έγκυρο, η συνάρτηση στέλνει μια απάντηση σφάλματος. Εάν είναι έγκυρο συνεχίζονται οι λειτουργίες της διεύθυνσης.

2.3.4.1 /root

Σε περίπτωση κάποιου λάθους της μεριάς του server, ο χρήστης θα ανακατευθυνθεί σε μια σελίδα σφάλματος 404.

2.3.4.2 /auth/

Σε αυτή την διαδρομή, οπού είναι η ριζά της /auth, ορίζουμε τις λειτουργίες σύνδεσης της εφαρμογής.

Με την μέθοδο POST καλούμε το ενδιάμεσο λογισμικό *loginLimiter*, το οποίοαποτρέπει πολλά αιτήματα από μια διεύθυνση IP σε μικρό χρονικό διάστημα, και την μονάδα login.

Η μονάδα login είναι μια ασύγχρονη συνάρτηση όπου πιάνει τυχόν σφάλματα. Περιμένει συγκεκριμένα μια ψηφιακή διεύθυνση και μια ψηφιακή υπογραφή από το σώμα, του αιτήματος, δεδομένα που αποστέλλονται από τον front end στο server. Ψάχνει στην βάση δεδομένων για μια εγγραφή όπου η διεύθυνση είναι ίδια με αυτή που λάβαμε από την αίτηση.

Χρησιμοποιούμε βιβλιοθήκες κρυπτογράφησης για την αποκρυπτογράφηση της διεύθυνσης από την ψηφιακή υπογραφή που λάβαμε. Ελέγχουμε αν είναι ίδια με την διεύθυνση που λάβαμε. Αν είναι ίδιες, δημιουργείται ένα token πρόσβασης με την διεύθυνση του φοιτητή, μια μικρή χρονική περίοδο λήξεις δεκαπέντε λεπτών (για λόγους προστασίας σε περίπτωση κλοπής του token) και ένα token ανανέωσης με την

διεύθυνση του φοιτητή και με χρονική περίοδο λήξεις έξι ημερών. Δημιουργούμε ένα νέο nonce για τον φοιτητή και ανανεώνουμε την βάση δεδομένων με αυτό. Στο τέλος, απαντάμε στο front end με ένα cookie που περιέχει το token ανανέωσης με όνομα *"jwt"* και το token πρόσβασης.

2.3.4.3 /auth/nonce

Η συγκεκριμένη διαδρομή χρησιμοποιεί την μέθοδο GET, χρειάζεται και μια παράμετρο (/auth/nonce/:address) της διεύθυνσης του φοιτητή για να αναζητήσει και να απαντήσει με το αντίστοιχο nonce.

2.3.4.4 /auth/refresh

Αυτή η διαδρομή χρησιμοποιεί την μέθοδο POST, η μονάδα χειρισμού λαμβάνει τα cookies από το front end, στη συνέχεια ελέγχει εάν το JWT υπάρχει στα cookies. Αν το token υπάρχει, επιβεβαιώνεται, αν η επιβεβαίωση είναι επιτυχής ανακτώνται οι πληροφορίες του μαθητή από το αποκωδικοποιημένο token και αναζητά έναν μαθητή με αυτήν τη διεύθυνση. Στη συνέχεια δημιουργείται ένα νέο token πρόσβασης και αποστέλλεται πίσω στο front end.

2.3.4.5 /auth/delete

Η διαδρομή ακούει για την μέθοδο POST, αφού λάβει το αναγνωριστικό του token του φοιτητή, θα διαγράψει το αντίστοιχο pdf αρχείο στην τοποθεσία του αποθετηριού στο server.

2.3.4.6 /auth/logout

Με την μέθοδο POST καλούμε την μονάδα χειρισμού logout που όταν λάβει τα cookies, θα τα διαγράψει και θα στείλει ένα JSON μήνυμα στο front end ότι τα cookies διαγραφτήκαν.

2.3.4.7 /student/

Η συγκεκριμένη διαδρομή χρησιμοποιεί την μέθοδο GET, χρειάζεται και μια παράμετρο της διεύθυνσης του φοιτητή για να αναζητήσει και να απαντήσει την εγγραφή που αντιστοιχεί στην διεύθυνση.

2.3.4.8 /oracle/

Την συγκεκριμένη διαδρομή μπορούμε να την περιγράψουμε σαν το oracle κομμάτι του server λόγω της επικοινωνίας με το smart contract, χρησιμοποιεί την μέθοδο POST και χρειάζεται μια παράμετρο της διεύθυνσης του φοιτητή για να αναζητήσει την εγγραφή που αντιστοιχεί στην διεύθυνση. Παράλληλα, συνδέεται ασύγχρονα με το πορτοφόλι που ταυτοποιείται από το oracle, με τον Alchemy κόμβο και παραλαμβάνει ένα instance του smart contract για να μπορεί να αλληλοεπιδράσει με αυτό (για τις παραπάνω λειτουργίες χρησιμοποιήθηκε η βιβλιοθήκη ethers δια μέσω της βιβλιοθήκης hardhat).

Ορίζεται μια Boolean τιμή false για την μεταβλητή που περιγράφει την κατάσταση πληρότητας των προϋποθέσεων για την αποφοίτηση του φοιτητή (*canGraduate*). Χρησιμοποιώντας τον αριθμό μητρώου του φοιτητή λαμβάνει τις απαραίτητες πληροφορίες για τον έλεγχο των προϋποθέσεων για την αποφοίτηση του φοιτητή και τον υπολογισμό του τελικού βαθμού [55].

Μετά από όλους τους ελέγχουν αν ο φοιτητής πληροί όλες τις προϋποθέσεις, οριζεται μια boolean τιμή true στην μεταβλητή *canGraduate* και καλείται η συνάρτηση που περιέχει όλες τις αλληλεπιδράσεις με το smart contract (*smartContractCalls*) με παραμέτρους το instance του smart contract, την διεύθυνση του φοιτητή και την μεταβλητή *canGraduate*.

Πρώτα καλούμε την συνάρτηση *hasToken* και περιμένουμε την απάντηση από το event. Αν η απάντηση που πήραμε είναι 0, τότε καλούμε την μεταβλητή *graduationCheck* για να θέσουμε την αντίστοιχη τιμή *canGraduate* στο smart contract. Έτσι, έχουμε τη δυνατότητα να καλέσουμε την *safeMint* συνάρτηση, όπου κάνει mint ένα νέο ERC 721 token και περιμένουμε την απάντηση από το event.

Αν υπάρχει η τιμή του token, δημιουργείται το δίπλωμα του φοιτητή ως ένα pdf έγγραφο, αποθηκεύεται στο αποθετήριο του server με το αριθμό του token σαν όνομα του εγγράφου και ανταποκρίνεται στο front end με αυτό.

2.3.4.9 /degrees/

Αυτή η διαδρομή χρησιμοποιεί την μέθοδο GET και χρειάζεται μια παράμετρο του ονόματος του pdf εγγράφου που αντιστοιχεί στον φοιτητή για να αναζητήσει και να διαγράψει το έγγραφο στο αποθετήριο του server.

2.3.5 Αποθετήριο του Server (Degrees storage)

Είναι ένας φάκελος στο server όπου αποθηκεύονται όλα τα διπλώματα σε μορφή pdf εγγράφων.

2.4 Front End

Χρησιμοποιώντας την React τεχνολογία ανάπτυξης εφαρμογών, έχει χτιστεί το κομμάτι αλληλεπίδρασης του φοιτητή με την εφαρμογή.

2.4.1 API Κομμάτι

Αυτό το μέρος επικοινωνεί με το back end μέσω API calls αξιοποιώντας την βιβλιοθήκη redux toolkit.

2.4.1.1 Redux Store

Το Redux store είναι το κύριο, κεντρικό αποθετήριο που αποθηκεύει όλες τις καταστάσεις της εφαρμογής.

Στην εφαρμογή έχουν οριστεί διάφοροι reducers (apiSlice, auth, oracle, degrees) και ένας *listener* για τον διαχειρισμό απεσταλμένων ενέργειων

Οι reducers είναι συναρτήσεις της Redux που τροποποιούν την κατάσταση μιας εφαρμογής με βάση τις ενέργειες που λαμβάνουν.

2.4.1.1 apiSlice

Το *apiSlice* ορίζει ένα *baseQuery* όλων των RTK Queries, απολήξεις που μιλάνε με το back end, το οποίο χειρίζεται κλήσεις API με αυτόματη αυθεντικοποίηση. Χρησιμοποιούμε τη συνάρτηση *fetchBaseQuery* από το Redux Toolkit για να δημιουργήσουμε ένα *baseQuery* και στη συνέχεια το εμφωλεύουμε σε μια προσαρμοσμένη συνάρτηση *baseQueryWithReauth*. Αυτή η προσαρμοσμένη συνάρτηση ελέγχει εάν το token του χρήστη έχει λήξει και εφόσον έχει λήξει, στέλνει ένα αίτημα ανανέωσης του token. Στη συνέχεια, δημιουργούμε ένα *apiSlice* για τις

κλήσεις API. Επισημαίνεται ότι χρησιμοποιούμε την μέθοδο *injectEndpoints*, η οποία μας επιτρέπει να χωρίσουμε τις απολήξεις και βοηθάει στην οργάνωση του κώδικα και στην μελλοντική ανάπτυξη μιας εφαρμογής).

2.4.1 Σελίδα Σύνδεσης (Log In Page)

Αυτό είναι το μέρος του front end που επιτρέπει στους φοιτητές να συνδεθούν στην εφαρμογή. Στη συγκεκριμένη εφαρμογή δεν χρησιμοποιούνται πεδία εισόδου, αλλά μια σύνδεση με το ψηφιακό πορτοφόλι MetaMask. Υπάρχουν διαφορά μηνύματα αναγνώρισης σφαλμάτων και κατά την μετάβαση στην αρχική σελίδα.

Όταν ο φοιτητής πατήσει το κουμπί *"Log in with MetaMask"* και συνδεθεί με το MetaMask, μπορούμε να ορίσουμε μια μεταβλητή με τιμή της διεύθυνσης του φοιτητή. Ύστερα,ρησιμοποιώντας αυτή την διεύθυνση σαν παράμετρο, στέλνουμε ένα αίτημα στην /auth/nonce διαδρομή του server για να παραλάβουμε την nonce τιμή. Λόγω της ασύγχρονης λειτουργίας πρέπει να χρησιμοποιήσουμε *asyncThunk*, μια συνάρτηση που καθυστερεί την εκτέλεση μιας συνάρτησης ή block κώδικα μέχρι να πάρει απάντηση.

Όταν παραλάβουμε το nonce, χρησιμοποιώντας την συνάρτηση *sign* της web3 βιβλιοθήκης θα υπογράψουμε ένα μήνυμα που περιέχει το nonce με την διεύθυνση του φοιτητή.

Στην συνέχεια στέλνουμε το αίτημα σύνδεσης στην /auth/ διαδρομή του server με την διεύθυνση και την υπογραφή σαν παραμέτρους. Αν τα δεδομένα είναι έγκυρα, ο server θα απάντηση με το token πρόσβασης. Έπειτα θέτουμε μια κατάσταση με την τιμή του token πρόσβασης και περιηγούμαι τον φοιτητή στην αρχική σελίδα.

2.4.1 Αρχική Σελίδα (Logged In Page)

Αυτό είναι το μέρος του front end που είναι προσβάσιμο μόνο σε πιστοποιημένους φοιτητές. Εμφανίζει πληροφορίες του συνδεδεμένου φοιτητή και παρέχει πρόσβαση σε πόρους.

Μόλις ο χρήστης έχει πιστοποιηθεί, θα σταλεί ένα αίτημα στην διαδρομή /students στο server με την διεύθυνση του φοιτητή ως παράμετρο. Χρησιμοποιώντας την table HTML ετικέτα οργανώνουμε και προβάλουμε τα ληφθέντα δεδομένα.

2.4.1.1 Εκτύπωση Διπλώματος

Με το πάτημα του κουμπιού *"Print Degree"* θα σταλεί ένα αίτημα στον server για την εκτύπωση του ψηφιακού διπλώματος με διαδρομή /oracle και με την διεύθυνση του φοιτητή ως παράμετρο χρησιμοποιώντας την *asyncThunk* συνάρτηση για την αποστολή του αιτήματος. Όταν ο server επικοινωνήσει με το smart contract, θα απαντήσει με το αναγνωριστικό του ERC721 token που αποθηκεύεται σε ένα cookie με όνομα "*studentTokenId"*. Αν η διαδικασία είναι επιτυχής, θα εμφανιστεί ένας σύνδεσμος με όνομα *"Click to open Degree"* στην σελίδα.

Αν ο φοιτητής πατήσει το σύνδεσμο θα σταλεί ένα αίτημα με την τιμή του *studentTokenId* cookie σαν παράμετρο που αντιστοιχεί στο όνομα του εγγράφου που δημιουργήθηκε στο server. Η απάντηση που θα ληφθεί είναι σε μορφή blob (Binary Large Object). Χρησιμοποιώντας την *createObjectURL* συνάρτηση, δημιουργείται ένα νέο URL που εμφανίζει το pdf έγγραφο. Το έγγραφο θα ανοίξει σε μια νέα καρτέλα και ο φοιτητής θα περιηγηθεί αυτόματα σε αυτή.

2.4.1.1 Διατήρηση Δεδομένων (Data Persistency)

Ο φοιτητής μπορεί να ανανεώσει τη σελίδα ή και να την κλήσει και να την ξανανοίξει σε ύστερη στιγμή, μένοντας συνδεδεμένος για έξι ημέρες. Υπεύθυνο για αυτό είναι το token ανανέωσης που ανανεώνει το token πρόσβασης σε κάθε ανανέωση της σελίδας αλλά και κάθε δεκαπέντε λεπτά (η ζωή του token πρόσβασης). Σε περίπτωση που δεν υπάρχει έγκυρο token πρόσβασης, ο φοιτητής θα περιηγηθεί στην σελίδα σύνδεσης. Όσο ο είναι συνδεδεμένος ο φοιτητής, τα δεδομένα που είναι αποθηκευμένα σε καταστάσεις ή cookies είναι διαθέσιμα.

2.4.1.1 Αποσύνδεση (Log out)

Στην αρχική σελίδα ο φοιτητής έχει την επιλογή να αποσυνδεθεί από την εφαρμογή πατώντας το κουμπί *"Log out"*. Αν ο φοιτητής πατήσει το κουμπί και υπάρχει τιμή στο *studentTokenId* cookie, αρχικά θα σταλθεί ένα αίτημα στην διαδρομή /auth/logout και στην συνέχεια θα σταλθεί ένα αίτημα στην διαδρομή /auth/delete κουβαλώντας την τιμή του σαν παράμετρο που αντιστοιχεί στο όνομα του εγγράφου που είναι αποθηκευμένο στο server. Μετά από αυτό θα καταργηθούν όλες οι καταστάσεις και τα cookies και ο φοιτητής θα περιηγηθεί στην σελίδα σύνδεσης.

2.3 GitHub

Όλος ο κώδικας υπάρχει στο ακόλουθο αποθετήριο:
https://github.com/tasosve/Blockchain_Technology_and_Academic_Data_Thesis.git

Συμπεράσματα

Με τις βασικές γνώσεις των blockchain, smart contracts και των oracles, χτίστηκε μια ακαδημαϊκή εφαρμογή που χρησιμοποιήσει ένα ERC721 token δεσμευμένο σε μια ψηφιακή διεύθυνση ενός φοιτητή μέσω του Ethereum blockchain για να εκτυπωθεί αυτόματα ένα αμετάβλητο ψηφιακό δίπλωμα κάθε τμήματος .

Σε θέμα χρόνου εκτέλεσης και στα τέλη συναλλαγών του Ethereum, το oracle ήταν ευεργετικό, καθώς υπολογίζοντας εκεί τις προϋποθέσεις αποφοίτησης η διαδικασία είναι αρκετά πιο γρήγορη.

Επιπλέον, υπάρχει μια τεράστια προσαρμοστικότητα στο τρόπο αρχιτεκτονικής μιας παρόμοιας εφαρμογής, συγκεκριμένα στην επικοινωνία μεταξύ oracle και smart contract, από την κατασκευή του token στην μεταφορά του μέχρι και στην δυνατότητα να δημιουργήσω ή να διαγράψω υπάρχουσες συναρτήσεις.

Μελλοντική Εργασία

Θα μπορούσε να υπάρχει και μελλοντικό έργο που επεκτείνει το εύρος των θεμάτων της παρούσας διπλωματικής εργασίας.

Το σύστημα θα μπορούσε να περιλαμβάνει ταυτόχρονα πολλά ακαδημαϊκά ιδρύματα, πράγμα που σημαίνει ότι ο φοιτητής θα μπορούσε να έχει διπλώματα και μεταπτυχιακά ακόμη και από αλλά πανεπιστήμια.

Τα διπλώματα θα μπορούσαν να αποθηκεύονται σε ένα IPFS (αποθήκευση αρχείων σε ένα κατανεμημένο σύστημα) με την επικοινωνία με το smart contract ή κάποιο άλλο κομμάτι της εφαρμογής.

Κατάλογος Αναφορών

[1] A. Narayanan, J. Clark, “Bitcoin's Academic Pedigree,” Communications of the ACM, Vol. 60 No. 12, Pages 36-45, December 2017, [Online], Available: https://cacm.acm.org/magazines/2017/12/223058-bitcoins-academic-pedigree/fulltext#top

[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” October 2008, [Online], Available: https://bitcoin.org/bitcoin.pdf

[3] Commission de Surveillance du Secteur Financier, “Distributed Ledger Technologies & Blockchain,” January 2022, [Online], Available: https://www.cssf.lu/en/2022/01/white-paper-distributed-ledger-technologies-dlt-blockchain/

[4] V. Buterin, “A Next Generation Smart Contract & Decentralized Application Platform,” 2013, [Online], Available: https://ethereum.org/en/whitepaper/

[5] Σ. Κ. Κάτσικας, Σ. Γκρίτζαλης, Κ. Λαμπρινουδάκης, “Ασφάλεια Πληροφοριών & Συστημάτων στον Κυβερνοχώρο,” Εκδόσεις Νέων Τεχνολογιών, Vol. 8, Pages 234-275, Vol. 15, Pages 425-451, March 2021, ISBN: 978-960-578-064-7

[6] Regulation of the European Parliament and of the Council on a pilot regime for market infrastructures based on distributed ledger technology, and amending Regulations, (EU) 2022/858, May 2022, [Online], Available: https://eurlex.europa.eu/legalcontent/EN/TXT/?uri=CELEX%3A32022R0858&qid=1684318170493

[7] M. Bawa, H. Garcia-Molina, A. Gionis, R. Motwani, “Estimating Aggregates on a Peer-to-Peer Network,” Computer Science Department Stanford University Stanford, 2003, [Online], Available: http://infolab.stanford.edu/~bawa/Pub/aggregates.pdf

[8] W. Changjing, J. Huiwen, Z. Jingshan, Y. Min, H. Qing, Z. Zhengkang, “A Review of Blockchain Layered Architecture and Technology Application Research,” Wuhan University, 2021, [Online], Available: https://www.researchgate.net/publication/364060600_A_Review_of_Blockchain_Layered_Architecture_and_Technology_Application_Research

[9] D. Yaga, P. Mell, N. Roby, K. Scarfone, “Blockchain Technology Overview,” National Institute of Standards and Technology, October 2018, [Online], Available: https://doi.org/10.6028/NIST.IR.8202

[10] G. C. Kessler, “An Overview of Cryptography,” March 2023, [Online], Available: https://www.garykessler.net/library/crypto.html#skc

[11] National Institute of Standards and Technology, “Data Encryption Standard (DES),” U.S. Department of Commerce, October 1999, [Online], Available: https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-

25/documents/fips46-3.pdf

[12] E. Milanov, “The RSA Algorithm,” June 2009, [Online], Available: https://sites.math.washington.edu/~morrow/336_09/papers/Yevgeny.pdf

[13] R. C. Merkle, “A Signature Based System on a Conventional Encryption Function,” 1988, [Online], Available: https://people.eecs.berkeley.edu/~raluca/cs261-f15/readings/merkle.pdf

[14] H. Sáez, O. Borde, “An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries,” Department of Engineering, University of Cambridge, February 2022, [Online], Available: https://www.researchgate.net/publication/358740207_An_Overview_of_Trees_in_Blockchain_Technology_Merkle_Trees_and_Merkle_Patricia_Tries

[15] D. Chaum, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,” Communications of the ACM, Vol. 24 Num. 2, February 1981, [Online], Available: https://dl.acm.org/doi/10.1145/358549.358563

[16] S. Haber, W. S. Stornetta, “How to Time-Stamp a Digital Document,” Springer-Verlag Berlin Heidelberg, 1991, [Online], Available: https://link.springer.com/article/10.1007/bf00196791

[17] National Institute of Standards and Technology, “Digital Signature Standard (DSS),” Information Technology Laboratory, Gaithersburg, MD 20899-8900, July 2013, [Online], Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

[18] G. Huaqun, Y. Xingjie, “A Survey on Blockchain Technology and its Security,” year, [Online], Available: https://www.sciencedirect.com/science/article/pii/S2096720922000070

[19] W. Li, S. Andreina, J. Bohli, G. Karame, “Securing Proof-of-Stake Blockchain Protocols,” NEC Laboratories Europe, Germany, September 2017, [Online], Available: https://www.researchgate.net/publication/319647471_Securing_Proof-of-Stake_Blockchain_Protocols

[20] K.J. O’Dwyer, D. Malone, ” Bitcoin Mining and its Energy Footprint,” National University of Ireland Maynooth, January 2014, [Online], Available: https://www.researchgate.net/publication/271467748_Bitcoin_Mining_and_its_Energy_Footprint

[21] E. Kapengut, B. Mizrach, “An Event Study of the Ethereum Transition to Proof-of-Stake,” Rutgers University, February 2023, [Online], Available: https://arxiv.org/pdf/2210.13655.pdf

[22] IBM, “IBM Blockchain Platform Technical Overview,” April 2022, [Online], Available: https://www.ibm.com/downloads/cas/Q9DGBLV7

[23] N. Szabo, “Smart Contracts,” 1994, [Online], Available: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/L

OTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[24] T. M. Hewa, Y. Hu, M. Liyanage, S. Kanhare, M. Ylianttila, "Survey on Blockchain based Smart Contracts: Technical Aspects and Future Research," University of Oulu, University College Dublin, University of New South Wales, March 2021, [Online], Available: https://www.researchgate.net/publication/350018207_Survey_on_Blockchain-Based_Smart_Contracts_Technical_Aspects_and_Future_Research

[25] M. Alharby, A. van Moorsel, "Blockchain-Based Smart Contracts : A Systematic Mapping Study," Newcastle University, Taibah University, November 2018, [Online], Available: https://ieeexplore.ieee.org/document/8756390

[26] F. Founier, I. Skarbovsky, W. Shama, C. Ozturan, A. Sen, T. Teixeira, D. Messina, D. Drakoulis, "Blockchain Tokenization and Smart Contracts - I," IBM, BOUN, ENG, GFT, HPE, November 2020, [Online], Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5d66b7c11&appId=PPGMS

[27] A. Beniiche, "A Study of Blockchain Oracles," INRS, Montr´eal, QC, Canada, July 2020, [Online], Available: https://arxiv.org/abs/2004.07140

[28] G. Caldarelli, "Overview of Blockchain Oracle Research," University of Verona, Department of Business Administration, May 2022, [Online], Available: https://www.mdpi.com/1999-5903/14/6/175

[29] Hyperledger Foundation, "A Distributed Operating System for Permissioned Blockchains," April 2018, [Online], Available: https://arxiv.org/abs/1801.10228

[30] H. Virani, M. Kyada, "A Systematic Literature Review on Smart Contracts Security," Department of Engineering University of Guelph, December 2022, [Online], Available: https://arxiv.org/pdf/2212.05099.pdf

[31] K. Mammadzada, M.Iqbal, F. Milani, L. García-Bañuelos, R. Matulevičius, "Blockchain Oracles: A Framework for Blockchain- Based Applications," Raimundas Matulevičius's Lab, September 2020, [Online], Available: https://www.researchgate.net/publication/344079826_Blockchain_Oracles_A_Framework_for_Blockchain-Based_Applications

[32] "Solidity Documentation," [Online], Available: https://docs.soliditylang.org/en/latest/index.html

[33] "OpenZeppelin Documentation," [Online], Available: https://docs.openzeppelin.com/

[34] "React Documentation," [Online], Available: https://react.dev/learn

[35] "Redux Documentation," [Online], Available: https://redux.js.org/tutorials/essentials/part-1-overview-concepts

[36] “Redux Toolkit: Overview,” [Online], Available: https://redux.js.org/redux-toolkit/overview/

[36] “RTK Query Overview,” [Online], Available: https://redux-toolkit.js.org/rtk-query/overview

[37] “Ethers Documentation,” [Online], Available: https://docs.ethers.org/v5/

[39] “Hardhat Documentation,” [Online], Available: https://hardhat.org/docs

[40] “web3.js - Ethereum JavaScript API,” [Online], Available: https://web3js.readthedocs.io/en/v1.10.0/

[41] “MetaMask Documentation,” [Online], Available: https://docs.metamask.io

[42] “eth-sig-util,” [Online], Available: https://github.com/MetaMask/eth-sig-util

[43] “Express Documentation,” [Online], Available: https://expressjs.com/en/5x/api.html

[44] “dotenv,” [Online], Available: https://github.com/motdotla/dotenv

[45] “JavaScript Cookie,” [Online], Available: https://github.com/js-cookie/js-cookie

[46] “jsonwebtoken,” [Online], Available: https://github.com/auth0/node-jsonwebtoken

[47] “Cross-Origin Resource Sharing (CORS),” [Online], Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

[48] “PDFKit,” [Online], Available: https://pdfkit.org/

[49] “About Node.js,” [Online], Available: https://nodejs.org/en/about

[50] G. Weston, “An Introduction to Alchemy – A Web3 Development Platform,” January 2023, [Online], Available: https://101blockchains.com/alchemy-web3/

[51] “What is the Sepolia Testnet?,” [Online], Available: https://www.alchemy.com/overviews/sepolia-testnet

[52] “MongoDB Documentation,” [Online], Available: https://www.mongodb.com/docs/atlas/getting-started/

[53] N. Karnik, “Introduction to Mongoose for MongoDB,” February 2018, [Online], Available: https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/

[54] A. Martiny, “One-click Login With Blockchain: A MetaMask Tutorial,” [Online], Available: https://www.toptal.com/ethereum/one-click-login-flows-a-metamask-tutorial

[55] Ε. Α. Λελίγκου, Ε. Συμεωνάκη, “Οδηγός Σπουδών Ακαδημαϊκού Έτους 2022-2023,” Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, Cha. 4, Pages 52-68, Μάρτιος 2023, [Online], Available: https://idpe.uniwa.gr/studies/student-handbook