ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Μελέτη της τρέχουσας τεχνολογικής στάθμισης για τη δόμηση και αξιοποίηση δεδομένων ιδρυμάτων ανώτατης εκπαίδευσης και διερεύνηση των προοπτικών αξιοποίησης των blockchains για τον ίδιο σκοπό

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΒΑΣΙΛΙΚΗΣ ΒΛΑΧΟΥ

Επιβλέπων: Δημήτριος Ασκούνης

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Μελέτη της τρέχουσας τεχνολογικής στάθμισης για τη δόμηση και αξιοποίηση δεδομένων ιδρυμάτων ανώτατης εκπαίδευσης και διερεύνηση των προοπτικών αξιοποίησης των blockchains για τον ίδιο σκοπό

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΒΑΣΙΛΙΚΗΣ ΒΛΑΧΟΥ

Επιβλέπων: Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Μαρτίου 2020.

.....	.....	.....
Δημήτριος Ασκούνης Καθηγητής Ε.Μ.Π.	Ιωάννης Ψαρράς Καθηγητής Ε.Μ.Π.	Χρυσόστομος Δούκας Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020

.....
Βασιλική Βλάχου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η πιστοποίηση των τίτλων σπουδών κατέχει κεντρικό ρόλο στο σημερινό ιδιαίτερα ανταγωνιστικό εκπαιδευτικό και εργασιακό περιβάλλον, αφού είναι απαραίτητη για την απόδειξη των προσόντων των εκπαιδευόμενων. Ωστόσο, ο τρόπος φύλαξης των πιστοποιητικών τα καθιστά ευάλωτα σε απώλεια και καταδολίευση, ενώ ταυτόχρονα προκαλεί καθυστερήσεις στην έκδοση και την επικύρωσή τους. Μία τεχνολογία, η οποία δίνει κατάλληλες λύσεις και μπορεί να εξαλείψει τα παραπάνω μειονεκτήματα είναι το blockchain.

Η παρούσα διπλωματική εργασία πραγματεύεται το πως μπορεί να χρησιμοποιηθεί το blockchain προκειμένου να εξαλειφθούν τα χρόνια προβλήματα που υπάρχουν στο χώρο της εκπαίδευσης και που σχετίζονται με την έκδοση και την επικύρωση πιστοποιητικών, καθώς και την ανάπτυξη μιας εφαρμογής που επιλύει τα εν λόγω προβλήματα.

Αρχικά, παρουσιάζονται τα προβλήματα που αφορούν την έκδοση και την πιστοποίηση των τίτλων σπουδών και γίνεται μια σύντομη εισαγωγή στην τεχνολογία blockchain. Στη συνέχεια, παρουσιάζεται η έρευνα που έχει προηγηθεί όσον αφορά τη χρήση του blockchain για την επικύρωση εκπαιδευτικών πιστοποιητικών, ενώ ακολουθεί αναλυτική ανάπτυξη της αρχιτεκτονικής που θα έπρεπε ιδανικά να έχει ένα blockchain που αφορά εκπαιδευτικά πιστοποιητικά. Έπειτα, πραγματοποιείται η ενδελεχής εξέταση και σύγκριση τριών διαφορετικών frameworks που θα μπορούσαν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογής έκδοσης και ελέγχου της εγκυρότητας πιστοποιητικών. Μέσω της σύγκρισης βγαίνει το συμπέρασμα ότι το Hyperledger Iroha αποτελεί το καταλληλότερο framework. Τέλος, περιγράφεται η αναλυτική διαδικασία που ακολουθήθηκε για την ανάπτυξη της εν λόγω εφαρμογής και παρουσιάζεται ο τρόπος χρήσης της εφαρμογής από τις διάφορες κατηγορίες χρηστών.

Λέξεις κλειδιά: Blockchain, Hyperledger Iroha, Εκπαιδευτικά Πιστοποιητικά, Ασφάλεια, GDPR, Αποκεντροποίηση, Διαλειτουργικότητα

Abstract

The verification of educational credentials is of great importance in today's highly competitive environment because it is essential for the demonstration of students' qualifications. However, the fact that higher education institutions keep student data in centralized databases that offer little or no interoperability causes them to be vulnerable to loss and fraud. At the same time, the centralized storage of credentials slows down their issuance and verification processes. Blockchain is a technology that gives appropriate solutions that can eliminate the aforementioned challenges in education.

The scope of this diploma thesis is not only to present how blockchain can be used to eliminate chronic challenges in the field of education related to the issuance and verification of certificates, but also the development of an application that will offer blockchain-enabled verification of credentials.

Initially, this thesis presents and analyzes the challenges that exist in the field of education and introduces the concept of blockchain technology. What follows is the presentation of the research that has already been conducted as far as the usage of blockchain for the verification of credentials is concerned. Furthermore, an appropriate methodology for the storage of credentials in an application based on blockchain is analyzed. Then follows a thorough examination and comparison of three different frameworks that could be used for the development of applications for certificate issuance and validation. It is concluded that Hyperledger Iroha is the most appropriate platform for the creation of such an application. Finally, this thesis describes how the application was developed and implemented and presents how it can be used by different user categories.

Keywords: Blockchain, Hyperledger Iroha, Educational Certificates, Security, GDPR, Decentralization, Interoperability

Ευχαριστίες

Η εκπόνηση της παρούσας διπλωματικής εργασίας πραγματοποιήθηκε στα πλαίσια του εργαστηρίου Συστημάτων Αποφάσεων και Διοίκησης του τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Ασκούνη Δημήτριο, για την επίβλεψη της παρούσας διπλωματικής εργασίας, καθώς και για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον επιστημονικό πεδίο.

Ιδιαιτέρως θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτωρ του εργαστηρίου Συστημάτων Αποφάσεων & Διοίκησης κ. Χρήστο Κοντζίνο, ο οποίος στάθηκε δίπλα μου από την αρχή, δίνοντας μου τις κατάλληλες συμβουλές και κατευθύνσεις, προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα.

Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτωρ Μιχάλη Κοντούλη για τη βοήθειά του σε τεχνικά ζητήματα.

Κλείνοντας θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την υποστήριξή τους όλα αυτά τα χρόνια.

Πίνακας Περιεχομένων

1 Εισαγωγή

1.1 Σκοπός

Είναι ευρέως αποδεκτό ότι ο σκοπός της Ανώτατης Εκπαίδευσης είναι να ανταποκριθεί στις μαθησιακές ανάγκες και τις προσδοκίες των ανθρώπων μέσα από την ανάπτυξη των πνευματικών τους ικανοτήτων. Άλλωστε, μέσω της Ανώτατης Εκπαίδευσης τα άτομα εξοπλίζονται με γνώσεις και δεξιότητες που τα βοηθούν να αξιοποιήσουν τα ταλέντα τους και να εκμεταλλευτούν τις ευκαιρίες που τους παρουσιάζονται. Ωστόσο, στο σημερινό ιδιαίτερα ανταγωνιστικό εκπαιδευτικό και εργασιακό περιβάλλον, τα εκπαιδευτικά ιδρύματα οφείλουν να εξασφαλίζουν, επίσης, ότι οι απόφοιτοί τους θα μπορούν να συνεχίσουν τις σπουδές τους σε κάποιο άλλο εκπαιδευτικό ίδρυμα ή να ενταχθούν στην αγορά εργασίας χωρίς καθυστερήσεις που οφείλονται σε έλλειψη τεχνογνωσίας από τη μεριά των πανεπιστημίων.

Η δυσκολία που υπάρχει όσον αφορά την πιστοποίηση των τίτλων σπουδών είναι σε σημαντικό βαθμό υπεύθυνη για τέτοιου είδους καθυστερήσεις. Η πιστοποίηση των τίτλων σπουδών κατέχει ιδιαίτερα σημαντικό ρόλο στον τομέα της εκπαίδευσης, αφού θεωρείται απαραίτητη για την απόδειξη και την επαλήθευση των προσόντων των εκπαιδευόμενων. Ωστόσο, τα εκπαιδευτικά πιστοποιητικά "αντιστέκονται" μέχρι στιγμής στην πρόοδο της τεχνολογίας. Αυτού του είδους τα έγγραφα διατίθενται ακόμα σε μορφές ευάλωτες σε απώλεια, φθορά ή και καταδολίευση, ενώ εξαρτώνται από τους ίδιους του εκπαιδευτικούς οργανισμούς ή τρίτους φορείς για την έκδοση και την επικύρωσή τους, γεγονός που απαιτεί συνήθως χρονοβόρες διαδικασίες. Αυτό οφείλεται στο γεγονός ότι οι δομές και οι βάσεις δεδομένων στις οποίες φυλάσσονται αυτά τα πιστοποιητικά είναι κεντρικές και η πρόσβαση σε αυτές είναι περιορισμένη στο προσωπικό του εκάστοτε εκπαιδευτικού ιδρύματος. Το γεγονός αυτό καταδεικνύει ένα ευρύτερο πρόβλημα στον τρόπο δόμησης των πανεπιστημιακών

δεδομένων, ο οποίος δεν επιτρέπει την ανάπτυξη καινοτόμων υπηρεσιών για τη διευκόλυνση των φοιτητών, των καθηγητών και του προσωπικού του ιδρύματος.

Επιπλέον, η αποθήκευση των εκπαιδευτικών πιστοποιητικών σε κεντρικές βάσεις δεδομένων καθιστά δυνατή την απώλεια, τη φθορά αλλά και την καταδολίευσή τους από τρίτους. Όμως, η απάτη και η διαφθορά όσον αφορά τα εκπαιδευτικά πιστοποιητικά οδηγεί στον κλονισμό της εμπιστοσύνης στο εκπαιδευτικό σύστημα. Συνεπώς, πρέπει να αναπτυχθούν τεχνικές που διασφαλίζουν την εγκυρότητα των πιστοποιητικών.

Προκειμένου να ενισχυθεί η αξιοπιστία των εκπαιδευτικών πιστοποιητικών και να διευκολυνθεί ο έλεγχος της εγκυρότητάς τους από άλλα ιδρύματα, απαιτούνται αλλαγές στον τρόπο λειτουργίας των εκπαιδευτικών ιδρυμάτων. Πιο συγκεκριμένα, πρέπει να υιοθετηθούν τεχνολογίες που οδηγούν στην καθολική δόμηση των πιστοποιητικών, την ανάπτυξη αυτοματοποιημένων διαδικασιών και την ενίσχυση της εμπιστοσύνης όσον αφορά τα εκπαιδευτικά πιστοποιητικά.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται στα εξής σημεία:

1. Αναλύθηκαν τα προβλήματα που αντιμετωπίζει ο τομέας της ανώτερης εκπαίδευσης όσον αφορά την επικύρωση και την αξιοπιστία των πιστοποιητικών.
2. Μελετήθηκαν ερευνητικές προσεγγίσεις που σχετίζονται με τη χρήση των blockchains σε ένα ευρύ πεδίο εφαρμογών.
3. Μελετήθηκαν ερευνητικές προσεγγίσεις βασισμένες στο blockchain που αφορούν την έκδοση και επικύρωση των πιστοποιητικών.
4. Αναπτύχθηκε η αρχιτεκτονική που θα έπρεπε ιδανικά να έχει ένα blockchain που αφορά τα εκπαιδευτικά πιστοποιητικά.
5. Εξετάστηκαν και αξιολογήθηκαν υπάρχοντα frameworks που θα μπορούσαν να χρησιμοποιηθούν για την ανάπτυξη μιας εφαρμογής που εξαλείφει τα προβλήματα του τομέα της εκπαίδευσης.
6. Υλοποιήθηκε μια εφαρμογή για την έκδοση και τον έλεγχο της εγκυρότητας των πιστοποιητικών.

1.2 Οργάνωση Κειμένου

Η παρούσα διπλωματική εργασία έχει οργανωθεί σε 8 κεφάλαια.

- Το κεφάλαιο 1 αποτελεί την εισαγωγή της εργασίας, όπου παρουσιάζεται ο σκοπός και η δομή της.

- Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση της εργασίας.
- Το κεφάλαιο 3 μελετά διάφορες ερευνητικές προσεγγίσεις που αφορούν τη χρήση των blockchains σε τομείς πέρα από την εκπαίδευση.
- Το κεφάλαιο 4 σχετίζεται με την προτυποποίηση του blockchain και στοχεύει στην ανάπτυξη της ιδανικής αρχιτεκτονικής που θα έπρεπε να έχει ένα blockchain που αφορά την εκπαίδευση.
- Στο κεφάλαιο 5 παρουσιάζονται και αξιολογούνται ορισμένα από τα σημαντικότερα blockchain frameworks.
- Το κεφάλαιο 6 αφορά την υλοποίηση και προσομοίωση της λειτουργίας μιας εφαρμογής έκδοσης και επικύρωσης εκπαιδευτικών πιστοποιητικών.
- Το κεφάλαιο 7 εξάγει τα γενικά συμπεράσματα της παρούσας εργασίας και προτείνει μελλοντικές επεκτάσεις.
- Το κεφάλαιο 8 παραθέτει τη βιβλιογραφία που χρησιμοποιήθηκε.

2 Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή στην τεχνολογία Blockchain

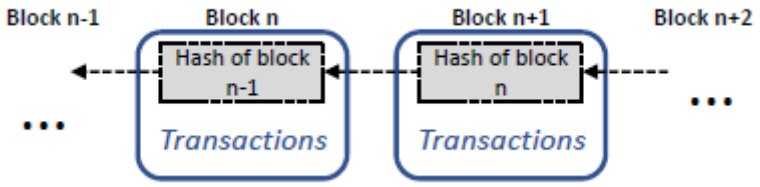
Το blockchain είναι μία τεχνολογία, η οποία δίνει τη δυνατότητα αποκεντρωμένης δόμησης δεδομένων σε μία κατανεμημένη συλλογή λογαριασμών. Το blockchain αποτελείται από blocks, τα οποία αντιπροσωπεύουν ένα σύνολο συναλλαγών (transactions) και είναι συνδεδεμένα μεταξύ τους. Τα blocks ενός blockchain δεν αποθηκεύονται σε μια κεντρική βάση δεδομένων, αλλά διανέμονται σε έναν αριθμό από κατανεμημένους peer-to-peer κόμβους (nodes) ενός δικτύου (network).

Κάθε block περιλαμβάνει τα εξής:

- δεδομένα ανάλογα με την εφαρμογή στην οποία χρησιμοποιείται το blockchain
- το hash του block
- το hash του προηγούμενου block
- τη χρονική στιγμή (timestamp) που δημιουργήθηκε το block

Το hash του block δημιουργείται μέσω της χρήσης κρυπτογραφικών συναρτήσεων κατακερματισμού (hash functions) και αποτελεί την αναπαράσταση των δεδομένων του block. Συνεπώς, εξαρτάται άμεσα από αυτά και άρα μπορεί να χρησιμοποιηθεί για την επαλήθευση της ακεραιότητας των συναλλαγών (transactions) που περιέχονται στο εκάστοτε block. Το hash κάθε block εξαρτάται και από το hash του προηγούμενου σε σειρά block, γεγονός που εξασφαλίζει ότι το blockchain είναι αμετάβλητο, καθώς η παραποίηση ενός block συνεπάγεται την αλλαγή του hash του συγκεκριμένου block και άρα και των hashes όλων των επόμενων blocks.

Στην ακόλουθη εικόνα παρουσιάζεται σχηματικά η δομή ενός blockchain.

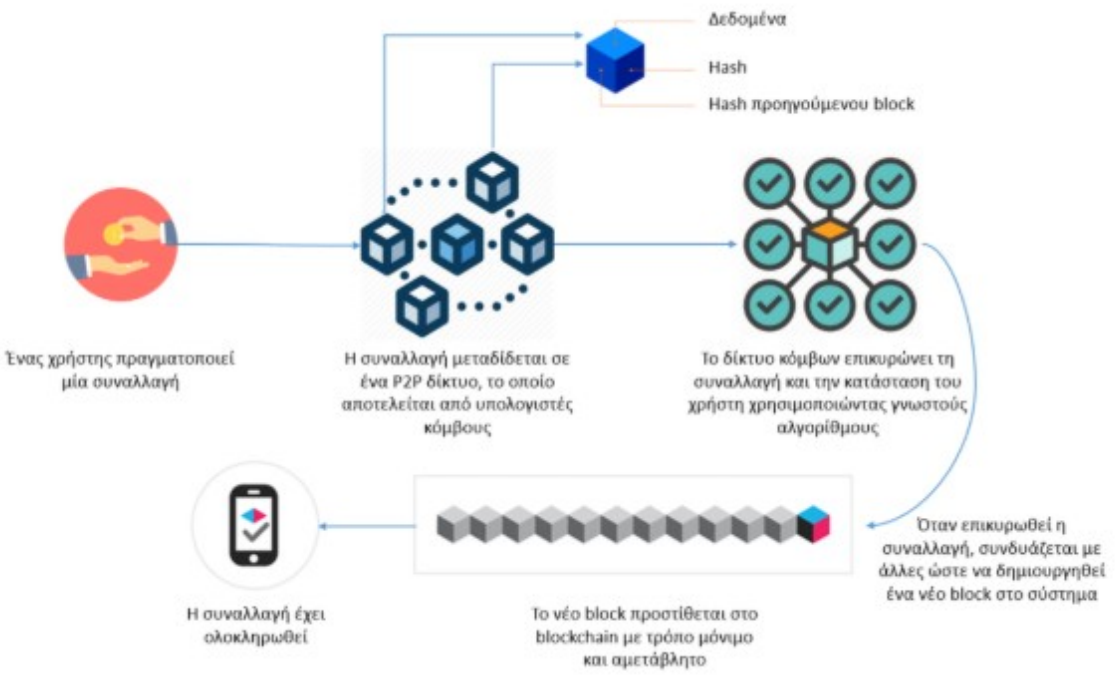


Εικόνα 1: Η δομή ενός blockchain

Το blockchain δε διαχειρίζεται από κάποια κεντρική αρχή, η οποία είναι υπεύθυνη για την επικύρωση και την επαλήθευση των συναλλαγών. Παρόλα αυτά, κάθε συναλλαγή του blockchain θεωρείται ότι είναι πλήρως επαληθευμένη. Αυτό συμβαίνει λόγω της ύπαρξης συναινετικού πρωτοκόλλου, το οποίο αποτελεί βασικό μέρος κάθε δικτύου blockchain και είναι υπεύθυνο για τη διατήρηση της ακεραιότητας του ιστορικού των συναλλαγών στο σύστημα.

Το blockchain είναι ευρέως γνωστό από τη χρήση του για συναλλαγές στον τομέα των κρυπτονομισμάτων, όπως το Bitcoin και το Ethereum. Μπορεί, όμως, να χρησιμοποιηθεί γενικότερα για τη δημιουργία μόνιμων, δημόσιων και διαφανών συστημάτων διαχείρισης δεδομένων. Τα τελευταία χρόνια, η προστιθέμενη αξία του blockchain έχει γίνει αντιληπτή από την ερευνητική κοινότητα και διάφορους οργανισμούς, οι οποίοι προτείνουν και αναπτύσσουν καινοτόμες λύσεις σε τομείς πέρα από τα κρυπτονομίσματα, όπως είναι η εκπαίδευση, η υγεία, η δημόσια διοίκηση και οι εφοδιαστικές αλυσίδες.

Στην ακόλουθη εικόνα παρουσιάζεται σχηματικά ο τρόπος λειτουργίας του blockchain.



Εικόνα 2: Τρόπος λειτουργίας του blockchain

Από την παραπάνω εικόνα γίνεται αντιληπτό ότι το blockchain μπορεί να αποτελέσει αποτελεσματική λύση σε κάθε τομέα που περιλαμβάνει συναλλαγές δεδομένων. Σε αυτό συμβάλλει σημαντικά και το γεγονός ότι η τεχνολογία blockchain δεν επηρεάζεται από το είδος των δεδομένων που αποθηκεύονται, με αποτέλεσμα να υπάρχει μεγάλη ελευθερία στις πιθανές εφαρμογές.

Κάθε δίκτυο blockchain διέπεται από τις ακόλουθες ιδιότητες:

- Συνεχής διαθεσιμότητα: Σε αντίθεση με παραδοσιακούς διακομιστές, το blockchain ουσιαστικά δεν σταματάει ποτέ να λειτουργεί, ούτε λόγω βλάβης ούτε λόγω συντήρησης.
- Αξιοπιστία: Το σύστημα ολοκληρώνει τις λειτουργίες του σταθερά και με επιτυχία, ενώ παράλληλα παρέχει εξηγήσεις για ενδεχόμενες αποτυχίες συναλλαγών.
- Ανοιχτό: Το blockchain δεν ξεχωρίζει συγκεκριμένους χρήστες ή υπολογιστές. Είναι ανοιχτό και προσβάσιμο από όλους.
- Ασφάλεια: Στο επίπεδο της κάθε συναλλαγής διασφαλίζει ότι η ιδιοκτησία μένει και μεταφέρεται στους σωστούς χρήστες. Όσον αφορά τη λειτουργία ολόκληρου του συστήματος, το blockchain προστατεύει τους χρήστες από κλοπές, μη εξουσιοδοτημένες προσβάσεις, διπλές πληρωμές (double spending) και ψεύτικες συναλλαγές.
- Ανθεκτικότητα: Ακόμα και υπό δύσκολες συνθήκες, το blockchain μπορεί να επιβεβαιώσει, αλλά και να μεταφέρει σωστά την ιδιοκτησία των δεδομένων του, ενώ είναι ανθεκτικό σε μεγάλο εύρος επιθέσεων.

- Τελική Σταθερότητα: Λόγω του τρόπου λειτουργίας του blockchain, υπάρχουν μερικές περιπτώσεις όπου οι απαντήσεις που δίνει το σύστημα δεν είναι σταθερές, αλλά με τη σύντομη πάροδο του χρόνου τελικά όλο το σύστημα επιστρέφει σταθερές απαντήσεις.
- Ακεραιότητα: Η συμπεριφορά του συστήματος δεν περιλαμβάνει λογικά λάθη. Το blockchain διατηρεί την ακεραιότητα των δεδομένων και βεβαιώνει την ασφάλεια των συναλλαγών, όπως και το ιστορικό τους.

Για την αύξηση της ασφάλειας και της εμπιστοσύνης στο δίκτυο, το blockchain διαθέτει διάφορους μηχανισμούς [1], μέσω των οποίων μπορεί να διασφαλιστεί μία συναλλαγή, η κατάσταση ενός block, αλλά και η κατάσταση ολόκληρου του συστήματος. Οι μηχανισμοί αυτοί είναι οι εξής:

- Απόδειξη ύπαρξης και μη ύπαρξης: Μπορεί να εξακριβωθεί εύκολα και σίγουρα αν ένα στοιχείο υπάρχει στο σύστημα.
- Απόδειξη χρόνου: Όταν αποθηκεύονται πληροφορίες στο blockchain, αποθηκεύεται και η ώρα κατά την οποία προστέθηκαν. Συνεπώς, είναι δυνατή η δημιουργία εφαρμογών που παρακολουθούν τη συχνότητα συμβάντων και διατηρούν την ιστορικότητά τους.
- Απόδειξη σειράς: Λόγω της απόδειξης χρόνου, σε περιπτώσεις συμφόρησης του δικτύου, μπορεί να φαίνεται η σειρά με την οποία πραγματοποιήθηκαν κάποιες αιτήσεις / συναλλαγές.
- Απόδειξη συγγραφής: Η εισαγωγή δεδομένων στο blockchain περιλαμβάνει και τα ψηφιακά στοιχεία του χρήστη που τα προσέθεσε. Ο μηχανισμός αυτός χρησιμεύει και για την ανίχνευση κακόβουλων επιθέσεων.
- Απόδειξη ιδιοκτησίας: Βασιζόμενο σε όλες τις υπόλοιπες αποδείξεις, φαίνεται πάντα με βεβαιότητα σε ποιόν ανήκει κάποιο στοιχείο μέσα στο blockchain.

2.2 Πανεπιστημιακά Δεδομένα και Blockchain

Στον τομέα της εκπαίδευσης, η προστιθέμενη αξια που μπορεί να προσφέρει το blockchain, ως δομή δεδομένων, είναι πολύπλευρη.

- Τα blocks στα οποία αποθηκεύονται τα δεδομένα είναι αμετάβλητα, το οποίο σημαίνει πως είναι πρακτικά αδύνατο να τροποποιηθούν τα εκπαιδευτικά πιστοποιητικά, που είναι αποθηκευμένα σε blockchain, μετά την έκδοσή τους από κάποιον κακόβουλο χρήστη. Συνεπώς, είναι σίγουρο ότι τα συγκεκριμένα πιστοποιητικά δεν έχουν μεταβληθεί.
- Είναι εφικτό να δοθεί μόνο σε εξουσιοδοτημένα εκπαιδευτικά ιδρύματα το δικαίωμα αποθήκευσης πιστοποιητικών στο blockchain. Επίσης, καταγράφεται, στο blockchain, πληροφορία που αφορά την ταυτότητα του πανεπιστημιακού ιδρύματος που εξέδωσε το εκάστοτε πιστοποιητικό. Συνεπώς, τόσο τα εκπαιδευτικά ιδρύματα όσο και οι εργοδότες μπορούν να είναι σίγουροι ότι τα συγκεκριμένα πιστοποιητικά είναι έγκυρα και αξιόπιστα.
- Το blockchain δεν εξαρτάται από κάποια κεντρική αρχή, αλλά κάθε χρήστης είναι ο ίδιος υπεύθυνος για τα δεδομένα του και μπορεί να δώσει σε τρίτους δικαιώματα πρόσβασης σε αυτά. Δίνοντας τον έλεγχο των πιστοποιητικών στους ίδιους τους σπουδαστές και όχι μόνο στο προσωπικό των εκπαιδευτικών ιδρυμάτων, τα πιστοποιητικά γίνονται προσβάσιμα σε ένα ευρύτερο κοινό, ενώ με αυτόν τον τρόπο διασφαλίζεται και η σωστή χρήση τους.

Έχουν ήδη γίνει προσπάθειες και ερευνητικές προσεγγίσεις, ώστε να χρησιμοποιηθεί η τεχνολογία του blockchain για την έκδοση και την επικύρωση εκπαιδευτικών πιστοποιητικών. Οι προσεγγίσεις αυτές παρουσιάζονται στο κεφάλαιο 3 της παρούσας διπλωματικής εργασίας.

2.3 Κανονισμός GDPR

Ο Γενικός Κανονισμός για την Προστασία των Δεδομένων (GDPR - General Data Protection Regulation) [2] 2016/679 είναι μια ρύθμιση στη νομοθεσία της Ευρωπαϊκής Ένωσης (ΕΕ) περί προστασίας των δεδομένων και της ιδιωτικής ζωής για όλα τα άτομα εντός της Ευρωπαϊκής Ένωσης και του Ευρωπαϊκού Οικονομικού Χώρου (ΕΟΧ). Αφορά, επίσης, την εξαγωγή δεδομένων προσωπικού χαρακτήρα εκτός των περιοχών της ΕΕ και του ΕΟΧ. Ο GDPR αποσκοπεί πρωτίστως να δώσει στους ιδιώτες τον έλεγχο των προσωπικών τους δεδομένων και να απλοποιήσει τους κανονισμούς για τις διεθνείς επιχειρήσεις, ενοποιώντας τις ρυθμίσεις εντός της ΕΕ (Council of the European Union 6/11/2015). Ο GDPR εκδόθηκε στις 14 Απριλίου 2016 και τέθηκε σε ισχύ στις 25 Μαΐου 2018.

Ως προσωπικά δεδομένα ορίζονται οι πληροφορίες οι οποίες περιγράφουν ένα άτομο, όπως στοιχεία αναγνώρισης (ονοματεπώνυμο, ηλικία, κατοικία, επάγγελμα, οικογενειακή κατάσταση), φυσικά χαρακτηριστικά, εκπαίδευση, εργασία (προϋπηρεσία, εργασιακή συμπεριφορά), οικονομική κατάσταση (έσοδα, περιουσιακά στοιχεία, οικονομική συμπεριφορά), ενδιαφέροντα, δραστηριότητες, συνήθειες. Επιπλέον, ως ευαίσθητα χαρακτηρίζονται τα προσωπικά δεδομένα ενός ατόμου που αναφέρονται στη φυλετική ή εθνική του προέλευση, στα πολιτικά του φρονήματα, στις θρησκευτικές ή φιλοσοφικές του πεποιθήσεις, στη συμμετοχή του σε συνδικαλιστική οργάνωση, στην υγεία του, στην κοινωνική του πρόνοια, στην ερωτική του ζωή, τις ποινικές διώξεις και καταδίκες του, καθώς και στη συμμετοχή του σε συναφείς με τα ανωτέρω ενώσεις προσώπων. Τα ευαίσθητα δεδομένα προστατεύονται από τον Νόμο με αυστηρότερες ρυθμίσεις από ότι τα απλά προσωπικά δεδομένα.

Σύμφωνα με τον GDPR, τα άτομα έχουν έναν αριθμό από δικαιώματα όσον αφορά τα προσωπικά τους δεδομένα, τα κυριότερα από τα οποία παρουσιάζονται παρακάτω:

- Δικαίωμα ενημέρωσης: Κάθε άτομο πρέπει να ενημερώνεται με σαφήνεια όσον αφορά τη μεταποίηση των δεδομένων του. Αυτό περιλαμβάνει το όνομα και τα στοιχεία επικοινωνίας του οργανισμού που τα επεξεργάζεται, το σκοπό της επεξεργασίας των δεδομένων, τη νομική βάση για την επεξεργασία, την προβλεπόμενη χρονική περίοδο που θα διατηρούνται τα δεδομένα του ατόμου, καθώς και τους αποδέκτες ή τις κατηγορίες αποδεκτών των δεδομένων.
- Δικαίωμα πρόσβασης: Κάθε άτομο πρέπει να έχει πρόσβαση στις διαδικασίες που σχετίζονται με τα προσωπικά του δεδομένα.
- Δικαίωμα διόρθωσης: Εάν τα δεδομένα είναι ανακριβή, τα άτομα μπορούν να ζητήσουν διόρθωση αυτών, την οποία ο οργανισμός οφείλει να ακολουθήσει.
- Δικαίωμα διαγραφής: Κάθε άτομο μπορεί να απαιτήσει διαγραφή των δεδομένων του όποτε το επιθυμεί.
- Δικαίωμα αντίρρησης: Ένα άτομο έχει το δικαίωμα να ζητήσει από έναν οργανισμό να περιορίσει την επεξεργασία των προσωπικών του δεδομένων.

- Δικαίωμα κοινοποίησης: Κάθε άτομο πρέπει να ενημερώνεται όσον αφορά την κοινοποίηση των δεδομένων του σε τρίτους.
- Δικαίωμα αυτοματοποιημένης λήψης αποφάσεων και δημιουργίας προφίλ: Κάθε άτομο πρέπει να ενημερώνεται όσον αφορά τη λογική της αυτοματοποιημένης επεξεργασίας των δεδομένων του.
- Δικαίωμα φορητότητας των δεδομένων: Ένα άτομο έχει το δικαίωμα να διασφαλίσει ότι τα προσωπικά του δεδομένα αποθηκεύονται σε δομημένη, ευρέως χρησιμοποιούμενη και μηχανικά αναγνώσιμη μορφή. Όταν είναι τεχνικά εφικτό, ένα άτομο μπορεί να ζητήσει να μεταφέρει απευθείας τα προσωπικά του δεδομένα από έναν οργανισμό σε έναν άλλο.

2.4 GDPR και Blockchain

Η συμμόρφωση του blockchain με τον κανονισμό GDPR αποτελεί ένα φλέγον ζήτημα, το οποίο απασχολεί την Ευρωπαϊκή Επιτροπή και αφορά τον τρόπο με τον οποίο χρησιμοποιείται η συγκεκριμένη τεχνολογία σε διάφορες περιπτώσεις και εφαρμογές. Συνεπώς, αξίζει να αναφερθούν, στο πλαίσιο του GDPR, κάποια σημαντικά ζητήματα που σχετίζονται με την προστασία των προσωπικών δεδομένων σε ένα blockchain.

Πιο συγκεκριμένα, η δόμηση και η αποθήκευση των δεδομένων σε ένα blockchain γίνεται με τέτοιο τρόπο που δεν είναι εφικτή η διαγραφή ή η διόρθωσή τους μετά την καταχώρησή τους στο blockchain. Άλλωστε, η "μη μεταβλητότητα" των δεδομένων αποτελεί ένα από τα βασικότερα χαρακτηριστικά του blockchain. Συνεπώς, είναι αδύνατο να διαγραφεί ή να επικαιροποιηθεί το αρχείο μιας συναλλαγής (transaction) χωρίς να καταστραφεί το blockchain. Ωστόσο, η διόρθωση και η διαγραφή των προσωπικών δεδομένων αποτελούν δύο βασικά δικαιώματα των χρηστών σύμφωνα με το GDPR.

Δεν υπάρχει κάποια επίσημη οδηγία από την Ευρωπαϊκή Επιτροπή όσον αφορά την αποθήκευση προσωπικών δεδομένων σε ένα blockchain. Μέχρι στιγμής, οι οργανισμοί, οι οποίοι αναπτύσσουν blockchain εφαρμογές, χρησιμοποιούν τις εξής τεχνικές προκειμένου να εξασφαλίσουν ότι το blockchain τους θα είναι σύμφωνο με τον GDPR:

- Τεχνικές κρυπτογράφησης που συνδυάζονται με την καταστροφή κλειδιού: στη συγκεκριμένη τεχνική το κλειδί παρέχει πρόσβαση στα δεδομένα του block. Όταν το κλειδί καταστρέφεται δεν είναι πλέον δυνατή η πρόσβαση στα αντίστοιχα δεδομένα. Αν και αυτή η τεχνική δεν αποτελεί πραγματική διαγραφή των δεδομένων, μπορεί να θεωρηθεί διαγραφή υπό την ευρεία έννοια.
- Φύλαξη δεδομένων σε κεντρικές βάσεις: στη συγκεκριμένη τεχνική τα δεδομένα αποθηκεύονται σε κεντρικές βάσεις, ενώ το blockchain χρησιμοποιείται για την αποθήκευση των συναλλαγών ανάμεσα στους χρήστες του συστήματος. Αυτή η τεχνική επιτρέπει τη διαγραφή των δεδομένων των χρηστών, αφού μπορούν να διαγραφούν από τις κεντρικές βάσεις. Το γεγονός ότι δεν μπορούν να διαγραφούν δεδομένα από το blockchain δεν αποτελεί πρόβλημα, καθώς το μόνο που διατηρείται στο σύστημα είναι οι συναλλαγές, οι οποίες δεν μπορούν να αποκαλύψουν τα δεδομένα των χρηστών.

3 *Ερευνητικές προσεγγίσεις σχετικά με Semantic Blockchains*

Στο παρόν κεφάλαιο θα παρουσιαστούν διάφορες ερευνητικές προσεγγίσεις που αφορούν τη χρήση των semantic blockchains σε ένα ευρύ πεδίο εφαρμογών, όπως για παράδειγμα το Διαδίκτυο των Πραγμάτων (Internet of Things) και η πιστοποίηση της διαδικτυακής ταυτότητας των χρηστών.

Οι Ruta et al. [3] προτείνουν μία αρχιτεκτονική που συνδυάζει το Σημασιολογικό Ιστό (Semantic Web) με το Blockchain, ώστε να βελτιώσει την επεκτασιμότητα του Διαδικτύου των Πραγμάτων (Internet of Things). Πιο συγκεκριμένα οι Ruta et al. ενσωματώνουν ένα επίπεδο σημασιολογικής αναζήτησης πόρων σε μία βασική δομή blockchain. Ένα blockchain που σχετίζεται με το Διαδίκτυο των Πραγμάτων στοχεύει στο να επιτρέπει την υλοποίηση υπηρεσιών όπως η εγγραφή πόρων, η αναζήτηση, η επιλογή και η τελική εκτέλεση ή αλλιώς πληρωμή. Στο παρόν paper, οι παραπάνω υπηρεσίες υλοποιούνται μέσω της χρήσης έξυπνων συμβολαίων (smart contracts) που επιτρέπουν την κατανεμημένη εκτέλεση υπηρεσιών και την ύπαρξη εμπιστοσύνης στο σύστημα. Στη συγκεκριμένη αρχιτεκτονική, οι πράκτορες (agents) αντιπροσωπεύονται από το δημόσιο κλειδί τους και σχετίζονται με ένα λογαριασμό (account). Κάθε πράκτορας μπορεί να κάνει σημασιολογική αναζήτηση πόρων, ώστε να αποκτήσει την ιδιοκτησία κάποιου πόρου ή να τον μεταβιβάσει σε άλλο λογαριασμό. Ένα βασικό χαρακτηριστικό αυτού του συστήματος αποτελεί το γεγονός ότι τα αποτελέσματα της αναζήτησης πόρων μπορούν να εξηγηθούν λογικά μέσω της σημασιολογική ένωσης του αιτήματος (request) με τους πόρους (resources).

Οι Rashid et al. [4] προτείνουν μία υλοποίηση που βασίζεται σε blockchain για την κατανεμημένη αποθήκευση και επεξεργασία εκπαιδευτικών δεδομένων με ασφαλή τρόπο. Η αρχιτεκτονική που παρουσιάζουν βασίζεται στη χρήση έξυπνων συμβολαίων (smart contracts) και Minimal Service Model και στοχεύει στην αποθήκευση εκπαιδευτικών δεδομένων στο Ethereum blockchain.

Προκειμένου να γίνει εφικτή η αναζήτηση των κατανεμημένων δεδομένων, οι Rashid et al. προτείνουν τη χρήση μίας RESTful σημασιολογικής υπηρεσίας Διαδικτύου (semantic web service) που επιτρέπει την αναζήτηση και εύρεση έξυπνων συμβολαίων μέσω του Ομοιόμορφου Αναγνωριστικού Πόρου (Uniform Resource Identifier - URI). Ταυτόχρονα, το συγκεκριμένο σύστημα χρησιμοποιεί τη βιβλιοθήκη web3 του Ethereum, προκειμένου να παρακολουθεί κάθε block που προστίθεται στο blockchain και να ανακτά τις συναλλαγές που έχουν καταγραφεί σε αυτό.

Οι Faísca et al. [5] εξετάζουν μια αρχιτεκτονική που συνδυάζει το blockchain με το Σημασιολογικό Ιστό για την κατανεμημένη αυθεντικοποίηση των χρηστών, αλλά και για τη διαχείριση της ταυτότητας και της ιδιωτικότητας των δεδομένων των χρηστών από τους ίδιους τους χρήστες. Πιο συγκεκριμένα, προτείνουν ένα μηχανισμό ταυτοποίησης χρηστών που βασίζεται σε WebID, JSON Web Tokens και blockchain. Το προτεινόμενο σύστημα χρησιμοποιεί το Namecoin blockchain, το οποίο επιτρέπει την αποθήκευση σε αυτό της ταυτότητας του κάθε χρήστη, καθώς και του αναγνωριστικού URI για το προφίλ του. Το προφίλ του εκάστοτε χρήστη, το οποίο περιέχει την απαραίτητη πληροφορία για την πιστοποίησή του, υλοποιείται με WebID profile και αποθηκεύεται σε ένα peer-to-peer (P2P) σύστημα αποθήκευσης αρχείων. Τα JSON Web Tokens χρησιμοποιούνται για την κρυπτογράφηση της προσωπικής πληροφορίας που απαιτείται να μεταφερθεί μεταξύ δύο συστημάτων.

Οι English et al. [6] περιγράφουν πως το blockchain μπορεί να εξαλείψει τις αδυναμίες που παρουσιάζουν τα Ομοιόμορφα Αναγνωριστικά Πόρου (Uniform Resource Identifiers - URI). Μέσω της χρήσης του Namecoin blockchain, που βασίζεται στο Bitcoin blockchain, είναι εφικτή η επίτευξη συναίνεσης όσων αφορά την ιδιοκτησία πόρων URI. Έτσι, το Σύστημα Ονοματοδοσίας Διαδικτύου (Domain Name System - DNS) σταματά να λειτουργεί ως μεσάζοντας, γεγονός που συμβάλλει σημαντικά στην αποκεντροποίηση του διαδικτύου.
Στο ίδιο paper επισημαίνουν πως είναι εφικτή η δημιουργία μιας οντολογίας για την αποθήκευση δεδομένων σε blockchain. Συνεπώς, ο Σημασιολογικός Ιστός διευκολύνει την ύπαρξη μιας κοινής κατανόησης των εννοιών του blockchain μεταξύ των ανθρώπων. Ταυτόχρονα, εξηγούν το γεγονός ότι η χρήση μιας σημασιολογικής οντολογίας για τα δεδομένα που αποθηκεύονται στο blockchain επιτρέπει τη σύνδεση του blockchain με άλλα Διασυνδεδεμένα Δεδομένα (Linked Data). Αυτή η διασύνδεση των δεδομένων αυξάνει τη δυνατότητα ανάλυσής τους από ανθρώπινους χρήστες.

4 Προτεινόμενη υλοποίηση Blockchain στην εκπαίδευση

4.1 Ερευνητικές προσεγγίσεις αποθήκευσης εκπαιδευτικών πιστοποιητικών βασισμένες σε blockchain

Ο κύριος στόχος του παρόντος κεφαλαίου είναι να περιγράψει διάφορες ερευνητικές προσεγγίσεις και υλοποιήσεις, βασισμένες στο blockchain, που αφορούν την αποτελεσματική και ασφαλή έκδοση πτυχίων από εκπαιδευτικά ιδρύματα, καθώς και τον έλεγχο της εγκυρότητάς τους από τρίτους.

Οι Rashid et al. [4] προτείνουν μία υλοποίηση που στοχεύει στην ασφαλή και κατανεμημένη αποθήκευση εκπαιδευτικών δεδομένων στο Ethereum blockchain. Η αρχιτεκτονική που εξετάζουν χρησιμοποιεί έξυπνα συμβόλαια (smart contracts) για την δημιουργία Open Badges. Για την αναζήτηση κατανεμημένων δεδομένων και έξυπνων συμβολαίων γίνεται χρήση μια RESTFul semantic web service, ενώ χρησιμοποιείται η οντολογία EthOn του Ethereum. Μέσω της βιβλιοθήκης web3 του Ethereum είναι εφικτή η ανάκτηση των συναλλαγών που έχουν καταγραφεί σε κάθε καινούργιο block που προστίθεται στο blockchain.

Οι English et al. [6] προτείνουν μια αρχιτεκτονική, βασισμένη σε blockchain, η οποία επιτρέπει στα εκπαιδευτικά ιδρύματα να εκδίδουν πιστοποιητικά και κατορθώματα των σπουδαστών στο Ethereum

blockchain, ενώ ταυτόχρονα επιτρέπει στους σπουδαστές να έχουν πρόσβαση και να διαχειρίζονται τα "βραβεία" που έχουν λάβει. Πιο συγκεκριμένα, το προτεινόμενο μοντέλο αποτελείται από δύο εφαρμογές βασισμένες σε αποκεντρωμένα blockchains. Η μία εφαρμογή θα διαχειρίζεται την δημοσίευση και επικύρωση των ψηφιακά υπογεγραμμένων πιστοποιητικών στο blockchain. Οι υπογραφές δένουν τα πιστοποιητικά με το εκπαιδευτικό ίδρυμα που τα εξέδωσε και τον εκάστοτε σπουδαστή. Η δεύτερη εφαρμογή δίνει τη δυνατότητα στο σπουδαστή να διαχειρίζεται τα πιστοποιητικά του και να παραχωρεί σε τρίτους δικαιώματα πρόσβασης.

Οι Tom Saeys et al. [7] προτείνουν μία λύση μέσω της οποίας το εκπαιδευτικό ίδρυμα θα εκδίδει ένα πιστοποιητικό και θα αποθηκεύει μέσω μιας ιστοσελίδας το hash του στο blockchain, ενώ θα στέλνει το πιστοποιητικό στο σπουδαστή μέσω email. Στη συνέχεια, ο σπουδαστής θα μπορεί να αποστέλλει το πιστοποιητικό σε τρίτους, οι οποίοι θα ελέγχουν την εγκυρότητά του υπολογίζοντας το hash του πιστοποιητικού μέσω της ιστοσελίδας και ελέγχοντας αν υπάρχει στο blockchain.

Το πανεπιστήμιο της Λευκωσίας [8] αποτελεί το πρώτο πανεπιστημιακό ίδρυμα παγκοσμίως που εκδίδει ακαδημαϊκά πιστοποιητικά, των οποίων η αυθεντικότητα μπορεί να ελεγχθεί μέσω του Bitcoin blockchain. Για την έκδοση του πιστοποιητικού δημιουργείται ένα ψηφιακό αρχείο που περιέχει το πιστοποιητικό. Στη συνέχεια, το αρχείο υπογράφεται με το ιδιωτικό κλειδί του πανεπιστημίου και η υπογραφή προστίθεται στο πιστοποιητικό. Ύστερα, δημιουργείται το hash του τελικού αρχείου και αποθηκεύεται στο blockchain μέσω του ιδιωτικού κλειδιού, το οποίο επαναχρησιμοποιείται για να επιτευχθεί η συναλλαγή που δείχνει ότι το πιστοποιητικό εκδόθηκε για έναν συγκεκριμένο φοιτητή μία συγκεκριμένη χρονική στιγμή.

Τα MIT Media Lab και Learning Machine [9] έχουν επίσης υλοποιήσει ένα σύστημα (Blockcerts) για την έκδοση επίσημων πιστοποιητικών (πχ εκπαιδευτικών, εργασιακών ή πιστοποιητικών της ιδιότητας μέλους σε μία ομάδα) σε παραλήπτες και την αποθήκευση στοιχείου για την απόδειξη της εγκυρότητάς τους σε Blockchain. Συγκεκριμένα, χρησιμοποιούν το Bitcoin blockchain, αλλά η υλοποίησή τους μπορεί εύκολα να επεκταθεί σε οποιοδήποτε άλλο αξιόπιστο σύστημα, όπως το Ethereum Blockchain. Προκειμένου να ελέγξει κάποιος την εγκυρότητα ενός πιστοποιητικού πρέπει να υπολογίσει μέσω του αλγορίθμου SHA-256 το hash του πιστοποιητικού που του έχει σταλεί και να ανακτήσει από το blockchain το αποθηκευμένο hash του πιστοποιητικού, ώστε να ελέγξει την ισότητά τους. Στη συνέχεια, μπορεί να ελέγξει την υπογραφή για να πιστοποιήσει ότι το πιστοποιητικό έχει εκδοθεί από το MIT και δεν έχει ανακληθεί.

Είναι εύκολο να παρατηρηθεί από τις παραπάνω ερευνητικές προσεγγίσεις ότι η γενική περίπτωση χρήσης blockchain στο χώρο της εκπαίδευσης κινείται στα πλαίσια των τεχνικών που ήδη παρουσιάστηκαν. Δηλαδή, το blockchain χρησιμοποιείται για την αποκεντρωμένη αποθήκευση και τον έλεγχο της εγκυρότητας ενός πιστοποιητικού. Συμπεραίνεται, λοιπόν, ότι στο χώρο της εκπαίδευσης η χρήση του blockchain αποσκοπεί κυρίως στο να επιτρέψει την αυθεντικοποίηση πιστοποιητικών, να επιτρέψει στους εκπαιδευόμενους να διαχειρίζονται τα δικαιώματα πρόσβασης σε αυτά και να προσθέσει ένα στρώμα ασφαλείας στη διαχείριση πανεπιστημιακών δεδομένων. Συνεπώς, μέσω της τεχνολογίας Blockchain η επικύρωση των πιστοποιητικών παύει να εξαρτάται από εκπαιδευτικούς οργανισμούς ή τρίτους φορείς, ενώ ταυτόχρονα η αποθήκευσή τους αποκεντροποιείται. Το blockchain, λοιπόν, αποτελεί τη βάση πάνω στην οποία μπορούν να χτιστούν ποικίλα εργαλεία και εφαρμογές προστιθέμενης αξίας που θα εκμεταλλεύονται τα αποτελέσματα της ενσωμάτωσης του blockchain στην εκπαίδευση. Για αυτό το λόγο, κρίνεται ιδιαίτερα σημαντική η

εύρεση του συνδυασμού τεχνικών που δημιουργεί το καταλληλότερο blockchain για το χώρο της εκπαίδευσης.

4.2 Αποθήκευση Δεδομένων

Σύμφωνα με το Γενικό Κανονισμό για την Προστασία των Δεδομένων (GDPR - General Data Protection Regulation) της Ευρωπαϊκής Ένωσης, πρέπει τα ευαίσθητα προσωπικά δεδομένα κάθε ανθρώπου να προστατεύονται και να μην είναι προσβάσιμα προς τρίτους εκτός αν το ίδιο το άτομο εκχωρήσει δικαίωμα πρόσβασης στα δεδομένα του. Είναι επίσης απαραίτητο να είναι εφικτή η οριστική διαγραφή των δεδομένων σε περίπτωση που ζητηθεί.

Δύο από τα βασικότερα χαρακτηριστικά του blockchain είναι τα εξής:

- ανοιχτό, άρα τα δεδομένα που είναι αποθηκευμένα σε αυτό είναι προσβάσιμα από όλους
- αμετάβλητο, επομένως δεν είναι εφικτή η διαγραφή των δεδομένων που έχουν αποθηκευτεί σε αυτό.

Για αυτό το λόγο είναι σημαντικό να διασφαλιστεί ότι τα δεδομένα που αποθηκεύονται στο blockchain είναι είτε μη προσωπικά, είτε σχετιζόμενα με προσωπικά, αλλά με τρόπο που τα πραγματικά δεδομένα είναι αδύνατον να ανακτηθούν από το blockchain. Αυτό πρέπει να γίνει εξακολουθώντας να εξασφαλίζεται η αμεταβλητότητα των δεδομένων.

Παρατηρείται ότι στις ερευνητικές προσεγγίσεις που περιγράφηκαν παραπάνω τα πιστοποιητικά αποθηκεύονται εκτός του blockchain, ώστε τα συστήματα να λειτουργούν σύμφωνα με το Γενικό Κανονισμό για την Προστασία των Δεδομένων, ενώ η διασφάλιση της ακεραιότητάς τους επιτυγχάνεται μέσω του blockchain.

Η ίδια τακτική θα χρησιμοποιηθεί και στην παρούσα υλοποίηση. Δηλαδή, τα πιστοποιητικά που θα εκδίδονται θα αποθηκεύονται στην κεντρική βάση πιστοποιητικών του εκάστοτε εκπαιδευτικού ιδρύματος, αλλά μέσω του blockchain θα είναι εφικτή η επαλήθευση της ιστορικότητας και της αυθεντικότητας των πληροφοριών κάθε πιστοποιητικού από οποιονδήποτε έχει πρόσβαση σε αυτό.

Προκειμένου να διασφαλιστεί η ασφάλεια των προσωπικών δεδομένων τα πιστοποιητικά θα αποθηκεύονται κρυπτογραφημένα στην κεντρική βάση, ώστε να μπορούν να διαβαστούν μόνο από όσους γνωρίζουν το κλειδί της αποκρυπτογράφησης, δηλαδή όσους έχουν αποκτήσει δικαίωμα πρόσβασης στο πιστοποιητικό.

4.3 Δομή του Blockchain Block

Στο παρόν κεφάλαιο και λαμβάνοντας υπόψιν τις υπάρχουσες ερευνητικές προσεγγίσεις, θα παρουσιαστεί η πληροφορία που θα είναι αποθηκευμένη μέσα σε ένα οποιοδήποτε block του blockchain που θα χρησιμοποιηθεί για την αποθήκευση και διαχείριση των πιστοποιητικών.

Οι ερευνητικές προσεγγίσεις των Rashid et al. [4], English et al. [6], του πανεπιστημίου της Λευκωσίας [8] και των MIT Media Lab και Learning Machine [9] χρησιμοποιούν τα blockchains του Bitcoin και του Ethereum. Συνεπώς, οι αρχιτεκτονικές που προτείνουν στηρίζονται στα δέντρα Merkle.

Το blockchain που θα χρησιμοποιηθεί στηρίζεται επίσης στα δέντρα Merkle (Merkle trees) [10] για τη διασφάλιση της ακεραιότητας των συναλλαγών.
Στην κρυπτογραφία και την επιστήμη των υπολογιστών, ένα δέντρο Merkle ή δέντρο κατακερματισμού (hash tree) είναι ένα δέντρο στο οποίο κάθε κόμβος φύλλο περιέχει την ετικέτα κατακερματισμού (hash) ενός block δεδομένων, ενώ κάθε κόμβος που δεν είναι φύλλο περιέχει τον κρυπτογραφικό κατακερματισμό των ετικετών που έχουν οι κόμβοι παιδιά του.
Τα δέντρα Merkle είναι χρήσιμα στην αποτελεσματική επαλήθευση των δεδομένων που αποθηκεύονται, διαχειρίζονται και μεταφέρονται μεταξύ υπολογιστών. Μέσω της χρήσης τους είναι εφικτή η διασφάλιση του ότι τα block δεδομένων που λαμβάνονται από άλλους peers σε ένα peer-to-peer δίκτυο δεν έχουν μεταβληθεί.

Στο συγκεκριμένο σύστημα, κάθε κόμβος φύλλο του δέντρου Merkle περιέχει την ετικέτα κατακερματισμού μιας συναλλαγής που αντιπροσωπεύει την προσθήκη ενός πόρου
(δηλαδή τη δημιουργία και έκδοση ενός πιστοποιητικού) στο επίσημο αρχείο πανεπιστημιακών πιστοποιητικών.

Κάθε block αποτελείται από την επικεφαλίδα (header) και το σώμα (body) του block.

Η επικεφαλίδα του block περιέχει τα ακόλουθα μεταδεδομένα, τα οποία χρησιμοποιούνται για την επικύρωση (verification) του block:

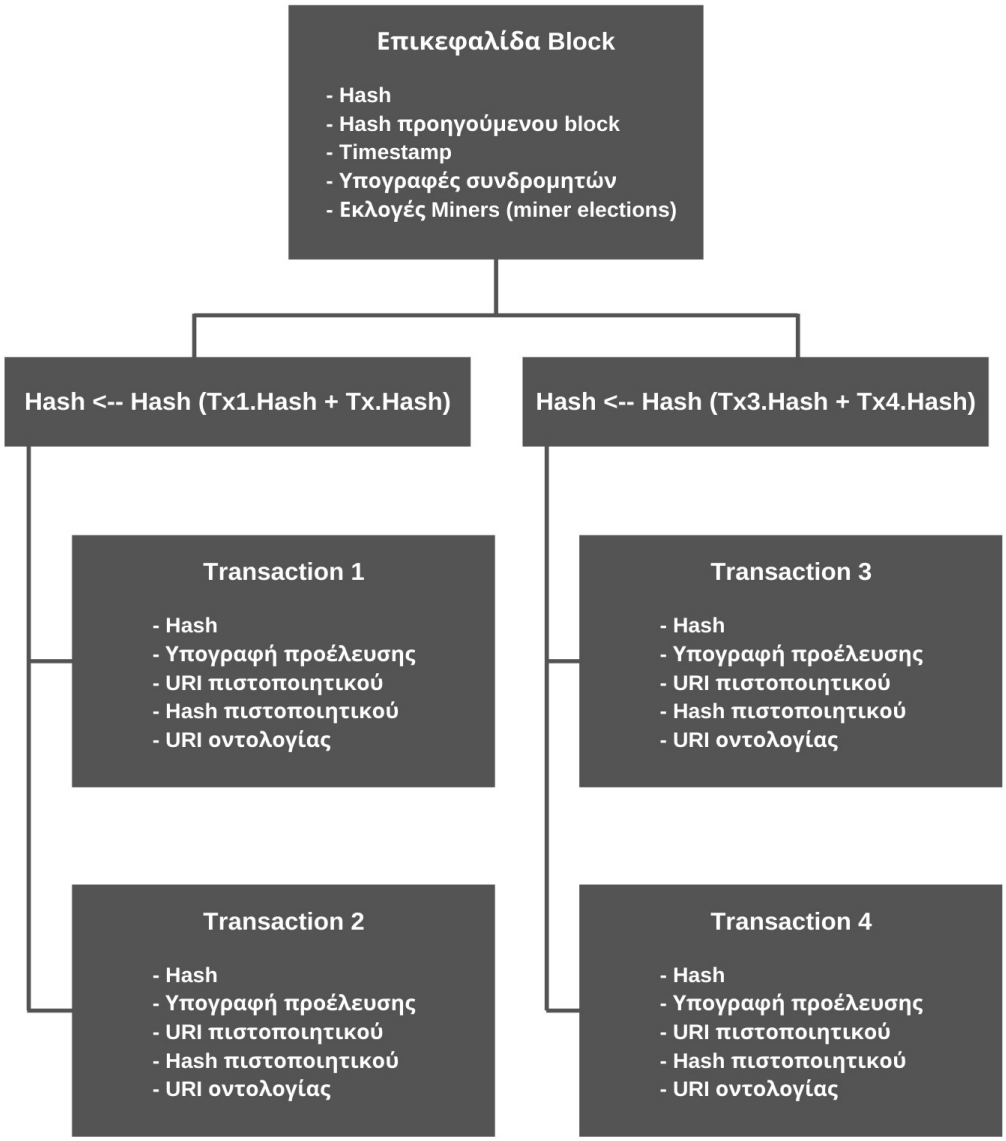
- Hash: το hash του block. Αν υποθέσουμε ότι η ρίζα του δέντρου Merkle έχει 2 κόμβους παιδιά, έστω c1 και c2, και ότι το προηγούμενο block είναι το bn-1, τότε το hash του νέου block bn θα προκύπτει από το hash της αλγοριθμικής σύνδεσης των hashes των bn-1, c1, c2. Το hash του προηγούμενου block χρησιμοποιείται ώστε να δημιουργηθεί η αλυσίδα των blocks και να μην είναι δυνατή η παραποίηση ενός μόνο block από κακόβουλους χρήστες χωρίς αυτό να συνεπάγεται μεταβολή και της υπόλοιπης αλυσίδας.
- Hash προηγούμενου block: το hash του προηγούμενου block ώστε να δημιουργηθεί η αλυσίδα των blocks.
- Timestamp: Η τωρινή χρονοσφραγίδα (εγγραφής της πληροφορίας μέσα στο blockchain).
- Υπογραφές συνδρομητών: Απαιτείται μια ψηφιακή υπογραφή από κάθε κόμβο που συνέβαλε στη δημιουργία του block. Οι υπογραφές διασφαλίζουν ότι το block θα παραμείνει έγκυρο αφού συναρμολογηθεί από τον miner.
- Εκλογές Miners (miner elections): Κάθε κόμβος που συνέβαλε στη δημιουργία του block χρειάζεται να παράσχει ένα τυχαίο αριθμό κρυπτογραφημένο με το ιδιωτικό του κλειδί. Αυτός ο αριθμός θα χρησιμοποιηθεί στη συνέχεια για την εκλογή του επόμενου miner.

Το σώμα του block (block body) περιέχει τις συναλλαγές (transactions) που πραγματοποιούνται.

- Hash: Το hash της συναλλαγής. Χρησιμοποιείται για την εξακρίβωση του περιεχομένου του πόρου (πιστοποιητικού) που είναι αποθηκευμένο εκτός του blockchain.
- Υπογραφή προέλευσης: η ψηφιακή υπογραφή του κόμβου προέλευσης

- URI πιστοποιητικού: Μία αναφορά στην πραγματική τοποθεσία του πόρου (που βρίσκεται αποθηκευμένο το πιστοποιητικό).
- Hash πιστοποιητικού: το hash του πιστοποιητικού που βρίσκεται αποθηκευμένο στην κεντρική βάση του εκπαιδευτικού ιδρύματος. Χρησιμοποιείται για την εύκολη εξακρίβωση του περιεχομένου του πιστοποιητικού από τρίτους.
- URI οντολογίας: Το URI της οντολογίας στην οποία ανταποκρίνεται αυτός ο πόρος (τι σημαίνουν τα δεδομένα).

Η επικεφαλίδα και το σώμα του block του blockchain που θα υλοποιηθεί παρουσιάζονται σχηματικά στην ακόλουθη εικόνα.



Εικόνα 3: Ένα block του Blockchain που σχετίζεται με την έκδοση και τον έλεγχο της εγκυρότητας εκπαιδευτικών πιστοποιητικών

Προκειμένου να καταγράφονται και άλλες συναλλαγές, όπως η χορήγηση άδειας για προβολή του πιστοποιητικού, πέρα από την έκδοση νέου πιστοποιητικού θα προστεθούν και συναλλαγές διαφορετικού είδους οι οποίες, για παράδειγμα, θα περιλαμβάνουν τη διεύθυνση (wallet address) του

παραλήπτη του πιστοποιητικού και το hash της συναλλαγής από την οποία ο αποστολέας έλαβε το πιστοποιητικό.

4.4 Ασφάλεια Συστήματος

4.4.1 Συναρτήσεις Κατακερματισμού

Είναι απαραίτητο να μπορεί να επαληθευτεί η ιστορικότητα και η αυθεντικότητα των πληροφοριών κάθε πιστοποιητικού από οποιονδήποτε έχει πρόσβαση σε αυτό. Για το σκοπό αυτό θα χρησιμοποιηθούν οι συναρτήσεις κατακερματισμού (hash functions). Οι κρυπτογραφικές συναρτήσεις κατακερματισμού επιτρέπουν τον υπολογισμό ενός hash με εύκολο τρόπο, ενώ είναι πρακτικά αδύνατον να υπολογιστούν τα στοιχεία από τα οποία προέκυψε. Δηλαδή, αν κάποιος διαθέτει ένα hash και κάποια δεδομένα, μπορεί να υπολογίσει το hash των δεδομένων και να το συγκρίνει με το αρχικό για να ελέγξει αν το hash προέρχεται από τα συγκεκριμένα δεδομένα. Ωστόσο, αν διαθέτει απλώς ένα hash δεν μπορεί να ανακαλύψει από που προέρχεται. Μία συνάρτηση κατακερματισμού μπορεί να εφαρμοστεί διαδοχικές φορές σε πρόσθετα κομμάτια δεδομένων προκειμένου να καταγραφεί η χρονολογία της ύπαρξης των δεδομένων.

Το blockchain, λόγω της ασφάλειας και της μη μεταβλητότητας των δεδομένων που είναι αποθηκευμένα σε αυτό, μπορεί να λειτουργήσει αποτελεσματικά ως ένας χώρος για την αποθήκευση των hashes δεδομένων που έχουν αποθηκευτεί εκτός αυτού. Στο blockchain οι συναρτήσεις κατακερματισμού προσδιορίζουν τη μοναδική κατάσταση της αλυσίδας κάθε χρονική στιγμή. Ουσιαστικά τα blocks είναι συνδεδεμένοι κατάλογοι δεδομένων, αφού όπως προαναφέρθηκε η επικεφαλίδα κάθε block περιέχει το hash του προηγούμενου block. Συνεπώς, οποιαδήποτε αλλαγή γίνει στα δεδομένα προκαλεί αναντοιστοιχία με το hash που είναι αποθηκευμένο στο blockchain και άρα για να τροποποιηθεί οποιοδήποτε δεδομένο ενός block πρέπει να τροποποιηθεί το hash του block και άρα και τα hashes όλων των επόμενων blocks του blockchain, κάτι που είναι πρακτικά αδύνατο.

Οι πιο γνωστές κρυπτογραφικές συναρτήσεις είναι η SHA-1, SHA-2 (ή αλλιώς SHA-256), SHA-3, MD5 και Blake2.

Η SHA-256 είναι η συνάρτηση κατακερματισμού που χρησιμοποιείται από το Bitcoin blockchain, ενώ το Ethereum blockchain χρησιμοποιεί την Keccak-256. Συνεπώς, οι ερευνητικές προσεγγίσεις που αναλύθηκαν στην αρχή του παρόντος κεφαλαίου χρησιμοποιούν αυτές τις 2 συναρτήσεις κατακερματισμού.

Στη συγκεκριμένη υλοποίηση θα χρησιμοποιηθεί ιδανικά η συνάρτηση SHA-256, η οποία χρησιμοποιείται ήδη σε blockchains που χρειάζονται συναρτήσεις κατακερματισμού υψηλής ασφαλείας, όπως για παράδειγμα το blockchain του bitcoin. Η SHA-256 είναι ασφαλέστερη από τις MD5 και SHA-1, ενώ έχει καλύτερη επίδοση από τη SHA-3. Αν και η BLAKE2 έχει υψηλότερη ταχύτητα από τη SHA-256, δε θα χρησιμοποιηθεί στην παρούσα υλοποίηση, καθώς η διαφορά στην ταχύτητα που έχουν οι δύο συναρτήσεις δεν παίζει ρόλο στην έκδοση πιστοποιητικών.

4.4.2 Διαχείριση Κλειδιών

Η κρυπτογράφηση δημοσίου κλειδιού (Public Key Cryptography) ή αλλιώς ασύμμετρου κλειδιού (Asymmetric Key Cryptography) επινοήθηκε στο τέλος της δεκαετίας του 1970 από τους Whitfield Diffie και Martin Hellman και παρέχει ένα εντελώς διαφορετικό μοντέλο διαχείρισης των κλειδιών κρυπτογράφησης από την προγενέστερη κρυπτογράφηση συμμετρικού κλειδιού (Symmetric Key Cryptography) [11]. Η βασική ιδέα αυτού του είδους κρυπτογράφησης είναι ότι ο αποστολέας και ο παραλήπτης δεν μοιράζονται ένα κοινό μυστικό κλειδί όπως στην περίπτωση της κρυπτογράφησης συμμετρικού κλειδιού, αλλά διαθέτουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες.

Πιο συγκεκριμένα κάθε χρήστης διαθέτει δύο κλειδιά κρυπτογράφησης: το ιδιωτικό (private key) και το δημόσιο (public key). Το δημόσιο κλειδί είναι φανερό είτε σε όλη τη διαδικτυακή κοινότητα, είτε σε συγκεκριμένους παραλήπτες. Αντίθετα, κάθε χρήστης θα πρέπει να προφυλάσσει το ιδιωτικό του κλειδί και να το κρατάει κρυφό από τους υπόλοιπους χρήστες. Τα δύο αυτά κλειδιά έχουν μαθηματική σχέση μεταξύ τους. Δηλαδή, αν το ένα έχει χρησιμοποιηθεί για την κρυπτογράφηση κάποιου μηνύματος, το άλλο θα χρησιμοποιηθεί για την αποκρυπτογράφηση αυτού. Σημαντικός παράγοντας για την επιτυχία αυτού του είδους κρυπτογραφικού αλγορίθμου είναι το γεγονός ότι η γνώση του δημοσίου κλειδιού δεν επιτρέπει με κανέναν τρόπο τον υπολογισμό του ιδιωτικού κλειδιού.

Πέρα από το ζευγάρι δημόσιου - ιδιωτικού κλειδιού κάθε χρήστης του blockchain έχει και μία διεύθυνση πορτοφολιού (wallet address), η οποία προκύπτει ως το hash του δημοσίου κλειδιού. Η διεύθυνση αυτή ουσιαστικά αποτελεί την εικονική τοποθεσία του χρήστη.

Η ύπαρξη κρυπτογραφίας δημόσιου κλειδιού είναι απαραίτητη προϋπόθεση για το μοντέλο ασφαλείας του blockchain. Τόσο οι ταυτότητες των χρηστών όσο και των συναλλαγών προέρχονται και επιβεβαιώνονται από πιστοποιητικά δημοσίου κλειδιού. Συνεπώς, προκειμένου να εξασφαλιστεί η ασφάλεια του blockchain είναι απαραίτητο να υλοποιηθεί ασφαλής διαχείριση των κλειδιών κρυπτογράφησης.

Το ζεύγος δημόσιου - ιδιωτικού κλειδιού καθώς και η διεύθυνση αποθηκεύονται τοπικά στο πορτοφόλι (wallet) κάθε χρήστη. Η διεύθυνση του πορτοφολιού είναι ορατή από όλους. Το δημόσιο κλειδί κάθε χρήστη μπορεί είτε να αποθηκευτεί σε κάποιο server ώστε να είναι προσβάσιμο από τρίτους χρήστες, είτε να ανακτηθεί μέσω της υπογραφής ενός πιστοποιητικού ή μιας συναλλαγής. Η ανάκτηση του δημόσιου κλειδιού μπορεί να επιτευχθεί στην περίπτωση που χρησιμοποιηθεί ο αλγόριθμος ψηφιακής υπογραφής ελλειπτικής καμπύλης (Elliptic Curve Digital Signature Algorithm - ECDSA) για την υπογραφή των πιστοποιητικών - συναλλαγών.

4.4.3 Πολιτική πρόσβασης στο δίκτυο του blockchain

Ένα δίκτυο blockchain είναι χωρίς άδεια (permission-less) όταν δεν απαιτείται άδεια για να γίνει κάποιος μέλος του δικτύου και να συμβάλει στη συντήρησή του, δηλαδή την επικύρωση των συναλλαγών και τη δημιουργία νέων block. Ένα δίκτυο blockchain είναι με άδεια (permissioned) όταν υπάρχει λίστα με τους κόμβους που μπορούν να έχουν πρόσβαση σε αυτό και αυτοί οι κόμβοι έχουν μοναδικό αναγνωριστικό.

Η επιλογή του είδους του δικτύου παίζει σημαντικό ρόλο στο σχεδιασμό του blockchain. Τα χωρίς άδεια blockchains απαιτούν αυστηρότερες μεθόδους για επίτευξη συναίνεσης. Επίσης πρέπει να ανταμείβουν τους συμμετέχοντες στην περίπτωση που καταναλώνουν σημαντικό επίπεδο υπολογιστικών πόρων, ώστε οι συμμετέχοντες να έχουν κίνητρο να παραμένουν ειλικρινείς. Για παράδειγμα, το Bitcoin επιτρέπει στους κόμβους να δημιουργούν και να κρατούν νέο κρυπτονόμισμα όταν επικυρώνουν νέο block συναλλαγών. Αντίθετα, τα blockchains που θέλουν άδεια συνήθως υιοθετούνται σε συστήματα όπου η συνεργασία είναι ελεγχόμενη και η ίδια η πρόσβαση αποτελεί ανταμοιβή αφού επιτρέπει την "αγορά" και "πώληση" υπηρεσιών.

Τα δίκτυα που χρησιμοποιούνται για τα Bitcoin και Ethereum blockchains είναι χωρίς άδεια (permission-less). Επομένως, είναι εύκολο να βγει το συμπέρασμα ότι οι Rashid et al. [4], οι English et al. [6], το πανεπιστήμιο της Λευκωσίας [8] και τα MIT Media Lab και Learning Machine [9] χρησιμοποιούν δίκτυα blockchain χωρίς άδεια στις προτεινόμενες αρχιτεκτονικές τους.

Για το blockchain που θα υλοποιηθεί υπάρχουν οι εξής εκδοχές:

- να χρησιμοποιηθεί permission-less δίκτυο: δε θα υπάρχει συγκεκριμένη λίστα από κόμβους, αλλά οποιοσδήποτε θέλει θα μπορεί να συμβάλει στην επικύρωση. Σε αυτή την περίπτωση, το blockchain θα μπορεί να χρησιμοποιηθεί πιο εύκολα και για άλλες εφαρμογές, καθώς τα περισσότερα blockchains χρησιμοποιούν αυτό το είδος δικτύου.
- να χρησιμοποιηθεί permissioned δίκτυο: τα εκπαιδευτικά ιδρύματα θα είναι οι κόμβοι που θα επικυρώνουν τα block και τις συναλλαγές. Σε αυτή την περίπτωση, ουσιαστικά προστίθεται ένα ακόμα επίπεδο ασφαλείας στο σύστημα, καθώς μόνο εξουσιοδοτημένα και άρα εγκεκριμένα εκπαιδευτικά ιδρύματα θα μπορούν να εκδώσουν ένα νέο πιστοποιητικό.

Όσον αφορά την παρούσα υλοποίηση κρίνεται σκοπιμότερο να χρησιμοποιηθεί permissioned blockchain, ώστε να υπάρχει έλεγχος ως προς τα εκπαιδευτικά ιδρύματα που μπορούν να καταχωρήσουν πιστοποιητικά στο blockchain. Έτσι θα αποφευχθεί ακόμα και η ελάχιστη πιθανότητα δημιουργίας εικονικών εκπαιδευτικών ιδρυμάτων για την έκδοση μη γνήσιων πιστοποιητικών.

Εννοείται όμως ότι το σύστημα που θα υλοποιηθεί μπορεί να λειτουργήσει εξίσου καλά και σε perimission-less περιβάλλον.

4.4.4 Επίτευξη συναίνεσης

Το blockchain είναι ένα κατανεμημένο αποκεντρωμένο δίκτυο που διέπεται από τις ιδιότητες της ασφάλειας, της αξιοπιστίας, της διαφάνειας και της ακεραιότητας. Δεν υπάρχει κάποια κεντρική αρχή για την επικύρωση και την επαλήθευση των συναλλαγών, αλλά κάθε συναλλαγή του blockchain θεωρείται ότι είναι πλήρως επαληθευμένη. Αυτό είναι εφικτό λόγω της ύπαρξης συναινετικού πρωτοκόλλου, το οποίο αποτελεί βασικό μέρος κάθε δικτύου blockchain.

Ένας αλγόριθμος συναίνεσης (consensus algorithm) είναι μία διαδικασία μέσω της οποίας όλοι οι κόμβοι ενός δικτύου καταλήγουν σε μία συμφωνία για την παρούσα κατάσταση του κατανεμημένου blockchain. Δηλαδή οι μηχανισμοί συναίνεσης εξασφαλίζουν ότι όλοι οι μη-ελαττωματικοί χρήστες του δικτύου εκτελούν τις ίδιες ανανεώσεις κατάστασης του συστήματος με τη σειρά που συνέβησαν τα γεγονότα που άλλαξαν την κατάστασή του.

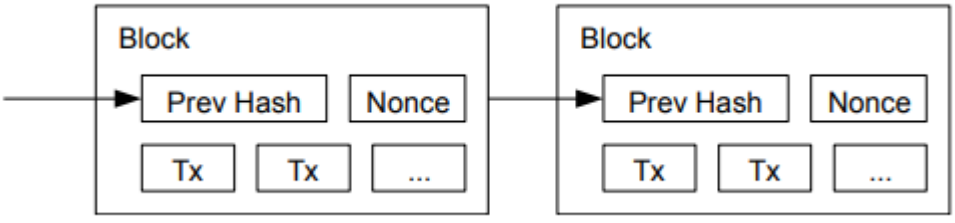
Υπάρχουν πολλοί αλγόριθμοι συναίνεσης, ενώ συνεχώς γίνονται προσπάθειες για δημιουργία νέων ή συνδυασμό των ήδη υπαρχόντων αλγορίθμων με στόχο την αύξηση της ασφάλειας και της απόδοσης. Οι πιο γνωστοί αλγόριθμοι συναίνεσης είναι οι εξής: Proof of Work (PoW), Proof of Stake (PoS) και Yet Another Consensus (YAC).

Στη συνέχεια θα αναλυθεί το πως χρησιμοποιούνται οι παραπάνω μηχανισμοί προκειμένου να επιτευχθεί συναίνεση σε ένα blockchain.

- **Proof of Work (PoW)**

  Ο συγκεκριμένος αλγόριθμος ουσιαστικά περιλαμβάνει την επίλυση ενός υπολογιστικού προβλήματος προκειμένου να προστεθούν νέα blocks συναλλαγών στο blockchain. Το υπολογιστικό πρόβλημα που θα χρησιμοποιηθεί πρέπει να είναι υπολογιστικά δύσκολο να επιλυθεί, αλλά ταυτόχρονα απαιτείται να είναι εύκολο να ελεγχθεί η εγκυρότητα μιας δοσμένης λύσης του. Η διαδικασία προσθήκης νέων block στο blockchain ονομάζεται mining (εξόρυξη) και οι κόμβοι τους δικτύου που καταβάλλουν προσπάθεια για την επίτευξή της ονομάζονται miners. Η επίλυση ενός τέτοιου προβλήματος απαιτεί την κατανάλωση σημαντικής υπολογιστικής δύναμης. Για αυτό το λόγο, δίνεται οικονομικό κίνητρο στους miners προκειμένου να ασχοληθούν με τη διαδικασία της εξόρυξης.

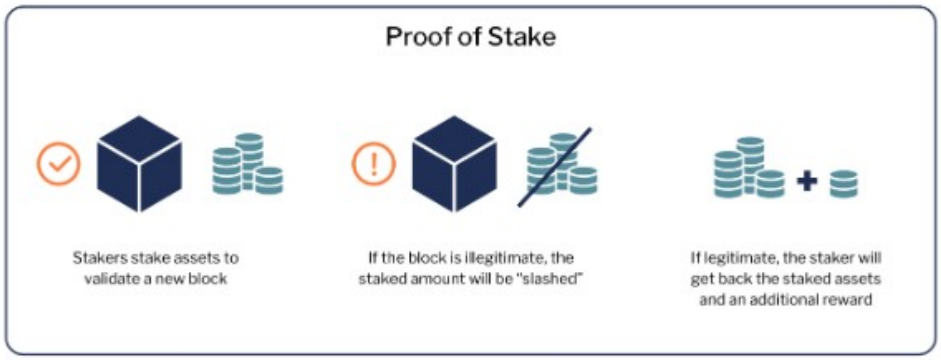
  Ο μηχανισμός Proof of Work χρησιμοποιείται από το blockchain του Bitcoin [12]. Προκειμένου να προστεθεί νέο block στην αλυσίδα του Bitcoin, πρέπει οι miners να βρουν έναν αριθμό, γνωστό ως nonce, που σε συνδυασμό με το περιεχόμενο του block δίνει hash που ξεκινάει από ένα συγκεκριμένο αριθμό μηδενικών ή που είναι μικρότερο από μία συγκεκριμένη τιμή.
  Ο miner που καταφέρνει να βρει ένα τέτοιο nonce προσθέτει το block στο blockchain και λαμβάνει την οικονομική αμοιβή, η οποία είναι 12,5 bitcoins.



Εικόνα 4: Proof of Work

- **Proof of Stake (PoS)**

  Ο μηχανισμός Proof of Stake δημιουργήθηκε ως μια εναλλακτική του μηχανισμού Proof of Work. Σε αυτόν τον αλγόριθμο συναίνεσης, οι επικυρωτές (validators) των νέων blocks αντί να επενδύσουν σε ακριβό εξοπλισμό προκειμένου να λύσουν ένα υπολογιστικά δύσκολο πρόβλημα, επενδύουν τα κρυπτονομίσματά τους στο σύστημα. Πιο συγκεκριμένα, κάθε validator ψηφίζει τα blocks που θεωρεί ότι μπορούν να προστεθούν στο blockchain κλειδώνοντας έναν αριθμό από τα κρυπτονομίσματα που διαθέτει σαν "στοίχημα". Συνεπώς, η ψήφος του κάθε validator είναι ανάλογη του ποσού που κλείδωσε. Το block που θα προστεθεί στο blockchain είναι αυτό που θα συγκεντρώσει τις περισσότερες ψήφους. Όπως και στο μηχανισμό Proof of Work, έτσι και στον Proof of Stake, δίνεται οικονομικό κίνητρο στους validators προκειμένου να είναι ειλικρινείς. Όσοι validators ψήφισαν το προστιθέμενο block λαμβάνουν οικονομική αμοιβή ανάλογη της ψήφου τους. Στην περίπτωση που κάποιος validator ψηφίσει δόλια ένα block που δεν είναι έγκυρο, κρατείται από το σύστημα το ποσό που κλείδωσε.



Εικόνα 5: Proof of Stake

Αν και στο παρελθόν το Ethereum blockchain χρησιμοποιούσε τον αλγόριθμο Proof of Work, πλέον γίνεται προσπάθεια ώστε να χρησιμοποιεί τον Proof of Stake.

- **Yet Another Consensus (YAC)**

  Ο Yet Another Consensus είναι ένας πρακτικός αποκεντρωμένος αλγόριθμος συναίνεσης, ο οποίος επιχειρεί να επιλύσει κλασικά προβλήματα που υπάρχουν σε αλγορίθμους συναίνεσης που βασίζονται στη Βυζαντινή Ανοχή Σφάλματος (Byzantine Fault Tolerance) [13]. Παράδειγμα τέτοιου προβλήματος αποτελεί η μη αποδοτική μετάδοση μηνυμάτων καθώς και η κατάληψη του συστήματος από "ισχυρούς ηγέτες", δηλαδή από χρήστες που καταλαμβάνουν μεγάλο μέρος των κόμβων του δικτύου του blockchain. Οι ισχυροί ηγέτες

μπορεί να έχουν νόημα σε ένα blockchain που αφορά κρυπτονομίσματα, ωστόσο σε ένα blockchain που σχετίζεται με την έκδοση και την επικύρωση πιστοποιητικών από εκπαιδευτικά ιδρύματα πρέπει να υπάρχει μεγαλύτερη ανεξαρτησία των χρηστών από τους "ηγέτες" του συστήματος.

Στη συνέχεια θα παρουσιαστεί ο τρόπος που λειτουργεί ο μηχανισμός Yet Another Consensus.

Οι τυπικοί συμμετέχοντες του YAC είναι οι εξής:

- Πελάτης (client): Κάθε πελάτης είναι ουσιαστικά ένας χρήστης που έχει ένα δημόσιο κλειδί καταχωρημένο στο σύστημα blockchain. Σε γενικές γραμμές, ο ρόλος του πελάτη είναι να δημιουργήσει συναλλαγές και να τις στείλει στην υπηρεσία παραγγελιών (ordering service). Ο πελάτης αναπτύσσει επίσης έξυπνα συμβόλαια, για να ορίσει τα δικαιώματα άλλων χρηστών στα δεδομένα του και να διενεργήσει ερωτήματα στους υπόλοιπους peers.
- Peer: Ένας peer είναι ένας από τους συμμετέχοντες του δικτύου που είναι υπεύθυνοι για την επικύρωση και την επίτευξη συμφωνίας όσον αφορά τις συναλλαγές και την αποθήκευση τους στα blocks του δικτύου. Οι peers διατηρούν το πλήρες ιστορικό συναλλαγών για την επικύρωση των προτάσεων (είτε δημιουργίας νέου block είτε μεταβολής της κατάστασης του δικτύου).
- Υπηρεσία παραγγελιών (Ordering Service): Η υπηρεσία παραγγελιών είναι μία μονάδα η οποία είναι υπεύθυνη για τη λήψη ενός συνόλου συναλλαγών και τη δημιουργία προτάσεων για νέα blocks. Μία πρόταση block περιέχει μία λίστα συναλλαγών η οποία πρέπει να συμφωνηθεί (μέσω επικύρωσης και ψηφίσματος) ανάμεσα στους peers.

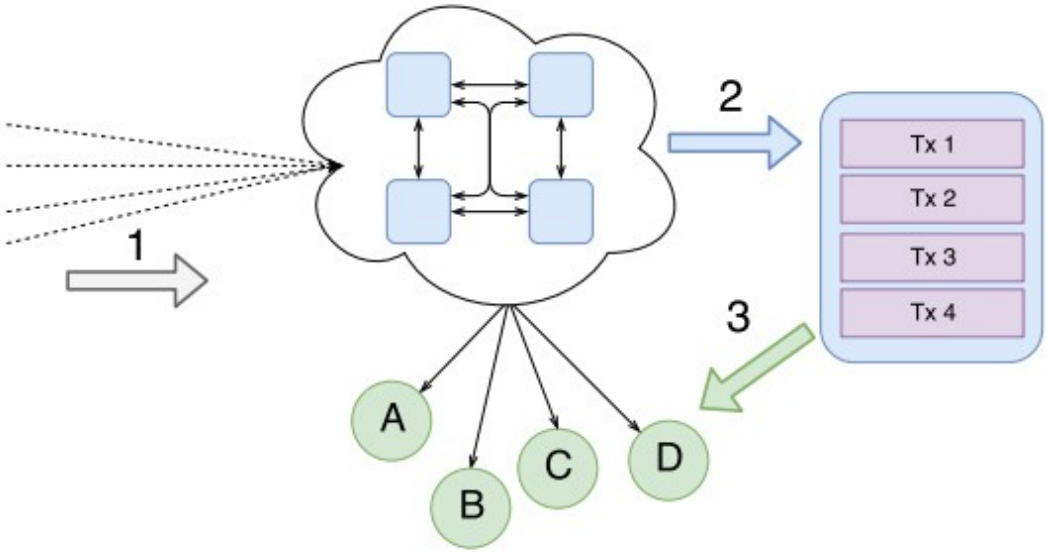
Προκειμένου να παρουσιαστεί ο τρόπος λειτουργίας του αλγορίθμου, χρειάζεται να γίνει ένας αριθμός από υποθέσεις.

- Θεωρείται ότι ο πελάτης (client) είναι γνωστός στους peers καθώς και ότι κάθε πελάτης (client) διαθέτει μία λίστα από peers με τους οποίους μπορεί να αλληλεπιδράσει.
- Ο πελάτης έχει το δικό του ζεύγος δημόσιου - ιδιωτικού κλειδιού αποθηκευμένα σε κάποια συσκευή, για παράδειγμα στον προσωπικό του υπολογιστή.
- Ο πελάτης έχει δικαιώματα εκτέλεσης ενός συγκεκριμένου υποσυνόλου εντολών / έξυπνων συμβολαίων, όπως η έκδοση ενός πιστοποιητικού και η παραχώρηση δικαιωμάτων πρόσβασης σε ένα πιστοποιητικό του από άλλους χρήστες.

Η γενική ροή λειτουργίας ενός γύρου του συστήματος μπορεί να περιγραφεί με τα ακόλουθα βήματα:

1. Ένας πελάτης (client) δημιουργεί μία συναλλαγή, μέσω των εντολών που έχει δικαίωμα να εκτελέσει, και την υπογράφει με το ιδιωτικό του κλειδί.
2. Ο πελάτης (client) στέλνει τη συναλλαγή σε έναν peer. Ο peer τη λαμβάνει, την επικυρώνει ώστε να βεβαιωθεί ότι έχει τη σωστή μορφή (όπως αυτή ορίζεται από το σύστημα) και τη μεταβιβάζει στην υπηρεσία παραγγελιών.
3. Η υπηρεσία παραγγελιών συγκεντρώνει τις συναλλαγές που μπορεί να συμπεριλάβει σε μία καινούρια πρόταση. Η πρόταση σχηματίζεται όταν η υπηρεσία παραγγελιών έχει συλλέξει ένα συγκεκριμένο αριθμό συναλλαγών ή έχει περάσει ένα συγκεκριμένο χρονικό διάστημα. Η πρόταση που σχηματίζεται από την υπηρεσία παραγγελιών περιέχει μια λίστα συναλλαγών που δυνητικά θα προστεθεί στο

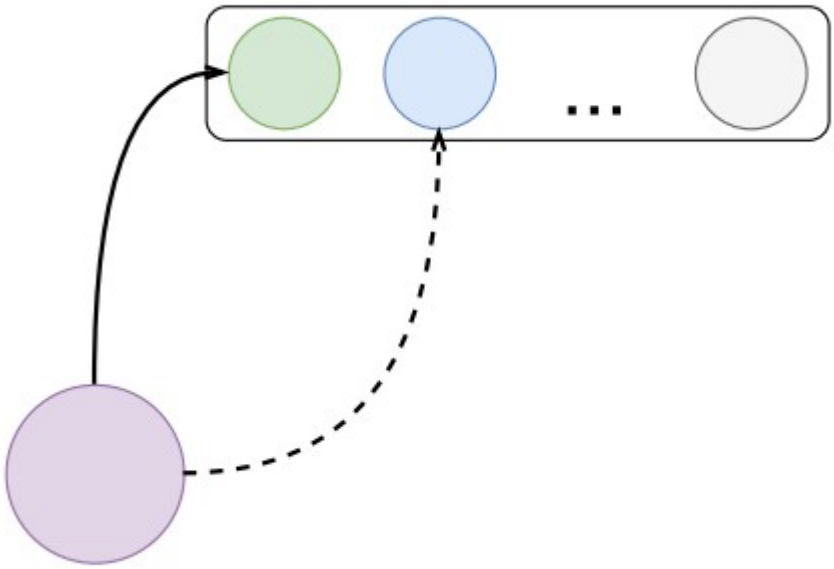
Blockchain σε αυτόν το γύρο. Στη συνέχεια η πρόταση αποστέλλεται στο σύνολο των peers του συστήματος.
Αυτή η διαδικασία παρουσιάζεται σχηματικά στην ακόλουθη εικόνα.



Εικόνα 6: Φάση παραγγελίας: 1) Οι πελάτες στέλνουν συναλλαγές σε έναν peer μέσω της υπηρεσίας παραγγελιών, 2) Η υπηρεσία παραγγελιών σχηματίζει μία πρόταση, 3) Η υπηρεσία παραγγελιών μοιράζεται την πρόταση με όλους τους peers, οι οποίοι θα πρέπει να ψηφίσουν για την εγκυρότητά της ώστε να προστεθεί ως νέο block στο Blockchain

4. Οι peers ανταλλάσσουν ψήφους μέσω του συστήματος και εν τέλει αποφασίζουν για το νέο block. Στην ουσία, σε αυτό το στάδιο, κάθε peer πρέπει να υπολογίσει μία επικυρωμένη πρόταση από την αρχική πρόταση που έλαβε από την υπηρεσία παραγγελιών. Μια επικυρωμένη πρόταση είναι ένα υποσύνολο των συναλλαγών που περιλαμβάνει η αρχική πρόταση, το οποίο έχει επικυρωθεί από τον peer. Το block που δημιουργείται από τον peer περιέχει ένα προτεινόμενο hash, τις συναλλαγές της επικυρωμένης πρότασης και επιπλέον μεταδεδομένα τα οποία απαιτούνται για την κρυπτογραφική επικύρωση της αλυσίδας. Το προτεινόμενο hash ορίζει μία μοναδική πρόταση για κάθε γύρο συνεργασίας. Το hash του block αντιπροσωπεύει την πρόθεση του peer να αποθηκεύσει ένα υποσύνολο των συναλλαγών στο blockchain. Αυτά τα hashes είναι απαραίτητα καθώς κάθε peer μπορεί να υπολογίσει διαφορετικά blocks από μία πρόταση.
   Στη συνέχεια, οι peers πρέπει να στείλουν μέσω μηνύματος την ψήφο τους. Το μήνυμα αυτό περιλαμβάνει τα προαναφερθέντα hashes και μία υπογραφή που πιστοποιεί τον peer. Η σειρά με την οποία οι peers θα ψηφίσουν για ένα block hash καθορίζεται από μία ειδική συνάρτηση (permutation function). Κατά τη διαδικασία ψηφοφορίας λοιπόν, το μήνυμα αποστέλλεται στον πρώτο peer της λίστας και στη συνέχεια στον επόμενο είτε αφού ο πρώτος στείλει την ψήφο του, είτε μετά το πέρας συγκεκριμένου χρονικού διαστήματος.

Η διαδικασία της ψηφοφορίας παρουσιάζεται σχηματικά στην ακόλουθη εικόνα.



Εικόνα 7: Διαδικασία ψηφοφορίας. Το μήνυμα αποστέλλεται στον πρώτο peer της λίστας (συμπαγής γραμμή), και στη συνέχεια στον επόμενο (διακεκομμένη γραμμή) μετά το πέρας συγκεκριμένου χρονικού διαστήματος

Η διαδικασία αυτή συνεχίζεται μέχρι το δίκτυο να λάβει ένα έγκυρο μήνυμα για την αποδοχή ή την απόρριψη των αλλαγών. Η πλειοψηφία συμβαίνει όταν τουλάχιστον τα ⅔ των peers του δικτύου αποδέχονται το block hash. Όταν ένας peer λάβει αποδοχή από την πλειοψηφία του δικτύου, μεταδίδει στους υπόλοιπους ένα μήνυμα εμπιστοσύνης.

5. Ο peer αποθηκεύει το block στο τοπικό του αντίγραφο (commit).

Ο συγκεκριμένος μηχανισμός επιτυγχάνει μικρή αδράνεια του συστήματος, η οποία οδηγεί σε υψηλή απόδοση συναλλαγών.

Ο αλγόριθμος συναίνεσης Yet Another Consensus χρησιμοποιείται από το blockchain του Hyperledger Iroha.

Οι Rashid et al. [4] και οι English et al. [6] αξιοποιούν το ήδη υπάρχον Ethereum blockchain για τις προτεινόμενες υλοποιήσεις τους. Συνεπώς, σύντομα θα χρησιμοποιούν τον αλγόριθμο Proof of Stake (PoS) για την επίτευξη συναίνεσης στα συστήματά τους.

Αντίθετα, το πανεπιστήμιο της Λευκωσίας [8] και τα MIT Media Lab και Learning Machine [9] στηρίζουν τις υλοποιήσεις τους στο blockchain του Bitcoin. Επομένως, στις προτεινόμενες αρχιτεκτονικές τους η συναίνεση επιτυγχάνεται μέσω του αλγορίθμου Proof of Work (PoW).
Στην παρούσα υλοποίηση θεωρείται προτιμότερο να χρησιμοποιηθεί ο μηχανισμός Yet Another Consensus (YAC). Αυτό στηρίζεται στα εξής:

- Στην περίπτωση που ένα blockchain σχετίζεται με την έκδοση και την επικύρωση πιστοποιητικών από εκπαιδευτικά ιδρύματα, πρέπει να υπάρχει μεγαλύτερη ανεξαρτησία των χρηστών από τους "ηγέτες" του συστήματος. Αυτό ακριβώς επιτυγχάνει ο συγκεκριμένος μηχανισμός συναίνεσης.
- Οι μηχανισμοί Proof of Work (PoW) και Proof of Stake (PoS) είναι ιδανικοί για blockchains που σχετίζονται με κρυπτονομίσματα, καθώς οι miners και οι validators αντίστοιχα έχουν οικονομικό κίνητρο για την προσθήκη έγκυρων νέων blocks στην αλυσίδα. Κάτι τέτοιο δε θα συμβαίνει στην υλοποίηση του blockchain για τα εκπαιδευτικά ιδρύματα, καθώς τα ίδια τα ιδρύματα είναι υπεύθυνα για τη δημιουργία νέων blocks.

4.5 Έκδοση και εξακρίβωση πιστοποιητικού

Στο παρόν κεφάλαιο παρουσιάζεται η διαδικασία για την έκδοση και την πιστοποίηση του περιεχομένου ενός πιστοποιητικού.

Κάθε χρήστης του blockchain διαθέτει ένα ζευγάρι δημόσιου και ιδιωτικού κλειδιού που χρησιμοποιεί για να υπογράφει τις συναλλαγές. Έστω ότι ένα πανεπιστημιακό ίδρυμα (issuer) θέλει να εκδώσει ένα νέο πιστοποιητικό. Σε αυτή την περίπτωση υπογράφει με το ιδιωτικό του κλειδί (private key) το hash του περιεχομένου του πιστοποιητικού. Η υπογραφή δένει το πιστοποιητικό με το εκπαιδευτικό ίδρυμα που το εξέδωσε. Στη συνέχεια, προσθέτει την υπογραφή στο τέλος του πιστοποιητικού και το αποθηκεύει στην κεντρική βάση. Το blockchain βρίσκεται σε επικοινωνία με τη βάση για την παραγωγή του τελικού hash που θα χρειαστεί, ώστε να αποθηκευτούν οι λεπτομέρειες της συναλλαγής σε ένα block.

Είναι απαραίτητο να μπορεί να επαληθευτεί η ιστορικότητα και η αυθεντικότητα των πληροφοριών κάθε πιστοποιητικού από οποιονδήποτε έχει πρόσβαση σε αυτό. Για το σκοπό αυτό θα χρησιμοποιηθούν οι συναρτήσεις κατακερματισμού (hash functions).

Έστω ότι κάποιος χρήστης θέλει να ελέγξει ότι το πιστοποιητικό είναι έγκυρο (δηλαδή δεν έχει τροποποιηθεί το περιεχόμενό του). Στην περίπτωση αυτή χρειάζεται να ελέγξει ότι το hash του πιστοποιητικού σε συνδυασμό με την απόδειξη Merkle (Merkle proof) του πιστοποιητικού και το hash του προηγούμενου block δίνουν τη ρίζα του δέντρου Merkle του block στο οποίο ανήκει η συναλλαγή έκδοσης του πιστοποιητικού ή απλούστερα ότι το υπολογισμένο από αυτόν hash του πιστοποιητικού ταυτίζεται με το hash του πιστοποιητικού που περιέχεται στη συναλλαγή (transaction).

Έστω ότι κάποιος χρήστης θέλει να επιβεβαιώσει ότι ένα δεδομένο πιστοποιητικό προέρχεται από ένα συγκεκριμένο εκπαιδευτικό ίδρυμα. Σε αυτή την περίπτωση πρέπει να ακολουθήσει τα εξής βήματα:

I. αποκρυπτογραφεί το πιστοποιητικό που βρίσκεται στην κεντρική βάση
II. αποκρυπτογραφεί την υπογραφή με το δημόσιο κλειδί (public key) του εκπαιδευτικού ιδρύματος που εξέδωσε το πιστοποιητικό

III. ελέγχει την ισότητα της αποκρυπτογραφημένης υπογραφής με το hash του υπόλοιπου πιστοποιητικού
IV. ελέγχει ότι το hash του δημοσίου κλειδιού ισούται με τη διεύθυνση πορτοφολιού (wallet address) του εκπαιδευτικού ιδρύματος.

5 Blockchain Frameworks

Στο παρόν κεφάλαιο θα παρουσιαστούν τα frameworks Ethereum, Hyperledger Fabric και Hyperledger Iroha, τα οποία αποτελούν ορισμένα από τα σημαντικότερα blockchain frameworks. Στη συνέχεια, θα γίνει η συγκριτική τους ανάλυση, ώστε να αποφασιστεί το καταλληλότερο framework για την έκδοση και την επικύρωση εκπαιδευτικών πιστοποιητικών.

5.1 Ethereum

Το Ethereum [14] είναι ένα framework που στοχεύει στη δημιουργία ενός πρωτοκόλλου για την υποστήριξη αποκεντρωμένων εφαρμογών. Μέσω της ανοιχτής πλατφόρμας του Ethereum blockchain, μπορεί ο καθένας να δημιουργήσει και να χρησιμοποιήσει αποκεντρωμένες εφαρμογές που στηρίζονται στην τεχνολογία του blockchain. Το Ethereum δεν ελέγχεται από κάποια κεντρική αρχή, αλλά είναι ένα έργο ανοιχτού κώδικα (open-source project) που συντηρείται και βελτιώνεται από πολλούς ανθρώπους παγκοσμίως. Το συγκεκριμένο framework έχει σχεδιαστεί ώστε να είναι εύκολα προσαρμόσιμο και ευέλικτο. Η δημιουργία νέων εφαρμογών μέσω του Ethereum είναι ιδιαίτερα εύκολη και η χρήση αυτών των εφαρμογών θεωρείται απολύτως ασφαλής.

5.1.1 Χαρακτηριστικά Ethereum

Στην παρούσα ενότητα θα περιγραφούν τα κυριότερα χαρακτηριστικά του Ethereum blockchain.

5.1.1.1 Έξυπνα Συμβόλαια (Smart Contracts)

Το Ethereum δίνει στους χρήστες τη δυνατότητα να δημιουργήσουν smart contracts (έξυπνα συμβόλαια) για την υλοποίηση λειτουργιών μέσα στη δομή του blockchain. Τα smart contracts είναι κομμάτια κώδικα που εκτελούν αυτόματα συναλλαγές στο blockchain όταν ικανοποιούνται συγκεκριμένες συνθήκες. Είναι αποθηκευμένα στο blockchain και εκτελούνται χωρίς να είναι αναγκαία η παρέμβαση μεσάζοντα. Στο Ethereum, οι προγραμματιστές μπορούν να δημιουργήσουν

έξυπνα συμβόλαια με τη χρήση γλωσσών προγραμματισμού, οι οποίες βασίζονται σε υφιστάμενες γλώσσες.

5.1.1.2 Συνάρτηση Κατακερματισμού

Το Ethereum blockchain χρησιμοποιεί τη συνάρτηση κατακερματισμού Keccak-256 για τον υπολογισμό των απαραίτητων hashes.

5.1.1.3 Διαχείριση Κλειδιών

Κάθε λογαριασμός (account) ορίζεται από ένα ζεύγος δημόσιου - ιδιωτικού κλειδιού. Οι λογαριασμοί ευρετηριοποιούνται μέσω της διεύθυνσης (address) τους, η οποία είναι τα τελευταία 20 bytes του hash του δημοσίου κλειδιού.

5.1.1.4 Πολιτική πρόσβασης στο δίκτυο του Ethereum

Το Ethereum blockchain χρησιμοποιεί δίκτυο χωρίς άδεια (permission-less). Ωστόσο, το framework του Ethereum επιτρέπει τη δημιουργία ιδιωτικών (permissioned) blockchains, καθώς πολλές εφαρμογές απαιτούν τη χρήση blockchains στα οποία έχουν πρόσβαση και μπορούν να συντηρήσουν μόνο εξουσιοδοτημένοι χρήστες. Αν και τα permissioned blockchains δε σχετίζονται με το δημόσιο Ethereum blockchain, συμβάλλουν στην πρόοδο του Ethereum καθώς ενισχύουν την ανάπτυξη του λογισμικού του.

5.1.1.5 Αλγόριθμος συναίνεσης

Ο αλγόριθμος συναίνεσης που χρησιμοποιείται στο Ethereum από τότε που δημιουργήθηκε είναι ο Proof of Work (PoW). Γίνονται όμως προσπάθειες ώστε το Ethereum blockchain να αρχίσει να χρησιμοποιεί τον Proof of Stake (PoS) για την επίτευξη συναίνεσης. Αυτό συμβαίνει, διότι ο Proof of Work απαιτεί μεγάλη κατανάλωση ενέργειας για την επίλυση του δύσκολου υπολογιστικού προβλήματος.
Αντίθετα, ο Proof of Stake αποτελεί μία πιο οικολογική εναλλακτική για την επίτευξη συναίνεσης σε ένα δίκτυο blockchain.

5.1.2 Λειτουργία Ethereum και Ethereum Virtual Machine (EVM)

Το Ethereum είναι ένα προγραμματιζόμενο Blockchain. Αντί να δώσει στους χρήστες ένα σύνολο προκαθορισμένων λειτουργιών, όπως για παράδειγμα Bitcoin συναλλαγές, το Ethereum επιτρέπει στους χρήστες να δημιουργήσουν τις δικές τους λειτουργίες μέσα στη δομή blockchain. Οι λειτουργίες αυτές μπορούν να είναι οποιασδήποτε πολυπλοκότητας επιθυμούν οι χρήστες.

Το Ethereum ουσιαστικά αναφέρεται σε μία σειρά πρωτοκόλλων που ορίζουν μια πλατφόρμα για αποκεντρωμένες εφαρμογές. Στην καρδιά αυτής της πλατφόρμας βρίσκεται η Εικονική Μηχανή Ethereum (Ethereum Virtual Machine - EVM), η οποία μπορεί να εκτελέσει κώδικα αυθαίρετης αλγοριθμικής πολυπλοκότητας. Η Εικονική Μηχανή του Ethereum αποτελεί το περιβάλλον εκτέλεσης των έξυπνων συμβολαίων στο Ethereum. Είναι πλήρως απομονωμένη από το δίκτυο, γεγονός που σημαίνει ότι ο κώδικας που τρέχει εντός της εικονικής μηχανής δεν έχει καμία πρόσβαση στο δίκτυο, το σύστημα αρχείων ή άλλες διαδικασίες. Μάλιστα τα έξυπνα συμβόλαια έχουν περιορισμένη πρόσβαση ακόμα και όσον αφορά άλλα έξυπνα συμβόλαια. Οι προγραμματιστές μπορούν να δημιουργήσουν έξυπνα συμβόλαια με τη χρήση φιλικών γλωσσών προγραμματισμού, οι οποίες βασίζονται σε υφιστάμενες γλώσσες όπως JavaScript και Python.

Όπως κάθε blockchain, έτσι και το Ethereum στηρίζεται σε ένα peer-to-peer δίκτυο. Η βάση δεδομένων του Ethereum blockchain συντηρείται και ενημερώνεται από πολλούς κόμβους που συνδέονται στο δίκτυο. Κάθε κόμβος του δικτύου τρέχει την Εικονικής Μηχανή Ethereum και εκτελεί τις ίδιες οδηγίες. Αυτή η μέθοδος καθιστά τους υπολογισμούς στο Ethereum πολύ πιο αργούς και ακριβούς από ότι σε έναν παραδοσιακό υπολογιστή. Ωστόσο, έτσι εξασφαλίζεται η ύπαρξη συναίνεσης σε ολόκληρο το blockchain, χωρίς να είναι αναγκαία η ύπαρξη μιας τρίτης έμπιστης αρχής (trusted third party).

5.1.3 Ether και κόστος συναλλαγών

Το Ethereum παρέχει τη δυνατότητα δημιουργίας ενός κρυπτονομίσματος που ονομάζεται “Ether”. Το Ether μπορεί να μεταφερθεί μεταξύ λογαριασμών και χρησιμοποιείται ως εξής:

- για συναλλαγές ανάμεσα σε χρήστες του Ethereum blockchain.
- για την πληρωμή των υπολογισμών που γίνονται εντός της Εικονικής Μηχανής του Ethereum (EVM). Δηλαδή χρησιμοποιείται ως μέσο κοστολόγησης της υπολογιστικής ισχύος που καταναλώνει η πλατφόρμα Ethereum.
  Όπως περιγράφηκε παραπάνω, το Ethereum υλοποιεί ένα περιβάλλον εκτέλεσης έξυπνων συμβολαίων μέσω της Εικονικής Μηχανής του Ethereum. Κάθε κόμβος που συμμετέχει στο δίκτυο εκτελεί τους ίδιους υπολογισμούς προκειμένου να επικυρώσει ένα νέο block. Το γεγονός ότι τα έξυπνα συμβόλαια εκτελούνται πολλές φορές, τα καθιστά δαπανηρά και άρα πρέπει να υπάρχει ένα κίνητρο ώστε να μη χρησιμοποιείται η πλατφόρμα για υπολογισμούς που μπορούν να γίνουν εκτός αυτής. Το κίνητρο είναι το κόστος σε Ether που έχει κάθε λειτουργία που πραγματοποιείται στο Ethereum blockchain.

5.1.4 Εφαρμογές

Από τα παραπάνω γίνεται σαφές ότι η πλατφόρμα του Ethereum μπορεί να υποστηρίξει πολλές και διαφορετικού είδους αποκεντρωμένες εφαρμογές που στηρίζονται σε blockchain για την υλοποίησή τους. Το συγκεκριμένο framework είναι κατάλληλο για εφαρμογές που αυτοματοποιούν την αλληλεπίδραση μεταξύ peers ή διευκολύνουν τη συντονισμένη δράση ομάδας σε ένα δίκτυο. Τέτοιες εφαρμογές αποτελούν, για παράδειγμα, ο συντονισμός της λειτουργίας ενός peer-to-peer marketplace και η αυτοματοποίηση περίπλοκων οικονομικών συμβολαίων. Το Ethereum, όπως και το Bitcoin,

επιτρέπει στους χρήστους του να ανταλλάσσουν χρήματα (“Ether”) χωρίς τη διαμεσολάβηση ενός τρίτου προσώπου, όπως χρηματοπιστωτικά ιδρύματα, τράπεζες και κυβερνήσεις. Αυτό είναι εφικτό, διότι οι οικονομικές αλληλεπιδράσεις μεταξύ των χρηστών μπορούν να πραγματοποιηθούν αυτόματα και με αξιοπιστία χρησιμοποιώντας κώδικα που εκτελείται στο Ethereum. Γενικότερα το Ethereum είναι κατάλληλο για εφαρμογές στις οποίες οι ιδιότητες της ασφάλειας, της εμπιστοσύνης και της αμετοβλητότητας των δεδομένων είναι ιδιαίτερα σημαντικές. Εφαρμογές που αφορούν τη καταχώρηση περιουσιακών στοιχείων, την ψηφοφορία, τη διακυβέρνηση και το Διαδίκτυο των Πραγμάτων θα μπορούσαν να ωφεληθούν σημαντικά από το Ethereum.

5.2 Hyperledger Fabric

Το Hyperledger Fabric [15] είναι ένα framework που στοχεύει στη δημιουργία μιας πλατφόρμας για την υποστήριξη αποκεντρωμένων εφαρμογών που βασίζονται σε blockchains. Το Hyperledger Fabric είναι ένα έργο ανοιχτού κώδικα, η δημιουργία του οποίου ξεκίνησε από το ίδρυμα Linux (Linux Foundation). Πολλοί προγραμματιστές από πολυάριθμους οργανισμούς συμβάλλουν στη συντήρηση και τη βελτίωση της πλατφόρμας. Η αρχιτεκτονική του συγκεκριμένου framework είναι διαμορφώσιμη, γεγονός που επιτρέπει την καινοτομία και την ευελιξία, ώστε η πλατφόρμα να μπορεί να προσαρμόζεται στις ανάγκες της εκάστοτε εφαρμογής. Η δημιουργία νέων εφαρμογών μέσω του Hyperledger Fabric είναι ιδιαίτερα εύκολη, ενώ είναι ιδανικό για χρήση από επιχειρήσεις.

5.2.1 Χαρακτηριστικά Hyperledger Fabric

Στην παρούσα ενότητα θα περιγραφούν τα κυριότερα χαρακτηριστικά του Hyperledger Fabric.

5.2.1.1 Έξυπνα Συμβόλαια (Smart Contracts) και Chaincode

Το Hyperledger Fabric παρέχει στους χρήστες του τη δυνατότητα να δημιουργήσουν έξυπνα συμβόλαια για την υλοποίηση λειτουργιών. Τα έξυπνα συμβόλαια ορίζουν τη λογική πίσω από όλες τις συναλλαγές και ουσιαστικά συνθέτουν την επιχειρηματική λογική μιας εφαρμογής που στηρίζεται σε blockchain. Λόγω της ντετερμινιστικής του φύσης, το Hyperledger Fabric δίνει στους προγραμματιστές τη δυνατότητα να δημιουργήσουν έξυπνα συμβόλαια με τη χρήση γλωσσών προγραμματισμού γενικού σκοπού, όπως Java και Go, γεγονός ιδιαίτερα σημαντικό αφού σημαίνει ότι οι επιχειρήσεις έχουν ήδη την τεχνογνωσία που απαιτείται για την ανάπτυξη έξυπνων συμβολαίων. Τα έξυπνα συμβόλαια μπορούν να ομαδοποιηθούν σε chaincodes, ενώ ένα έξυπνο συμβόλαιο μπορεί να ανήκει σε περισσότερα από ένα chaincodes. Τα chaincodes μπορούν να χρησιμοποιηθούν για την ανάπτυξη επιχειρηματικών συμβολαίων, των ορισμό πόρων και τη διαχείριση αποκεντρωμένων εφαρμογών.

5.2.1.2 Πολιτική πρόσβασης στο δίκτυο του Hyperledger Fabric

Το Hyperledger Fabric χρησιμοποιεί δίκτυο με άδεια (permissioned). Οι συμμετέχοντες του δικτύου της συγκεκριμένης πλατφόρμας δεν είναι ανώνυμοι, αλλά εγγράφονται στην πλατφόρμα μέσω ενός έμπιστου παρόχου υπηρεσιών συνδρομής (Membership Service Provider - MSP) και "γνωρίζουν" ο ένας τον άλλο. Αυτό σημαίνει ότι αν και οι συμμετέχοντες μπορεί να μην έχουν πλήρη εμπιστοσύνη ο ένας στον άλλο, διότι για παράδειγμα μπορεί να είναι ανταγωνιστές, το δίκτυο μπορεί να λειτουργήσει σύμφωνα με ένα μοντέλο διακυβέρνησης που βασίζεται στην όποια εμπιστοσύνη υπάρχει μεταξύ τους λόγω για παράδειγμα κάποιας νομικής συμφωνίας. Επίσης, στο Hyperledger Fabric, ο κίνδυνος κάποιος συμμετέχοντας να εισάγει στο blockchain κακόβουλο κώδικα μέσα από έξυπνα συμβόλαια έχει εξαλειφθεί. Αυτό συμβαίνει καθώς ο εντοπισμός ενός τέτοιου χρήστη είναι εύκολος αφού όλες οι συναλλαγές καταγράφονται στο blockchain.

5.2.1.3 Διαχείριση ταυτότητας (Identity Management)

Προκειμένου να εξασφαλίσει ότι το blockchain λειτουργεί πράγματι με permissioned τρόπο, το Hyperledger Fabric framework παρέχει μια υπηρεσία διαχείρισης ταυτότητας (Identity Management Service), η οποία διαχειρίζεται τα IDs των χρηστών και πιστοποιεί την ταυτότητα όσων συμμετέχουν στο δίκτυο. Λίστες ελέγχου πρόσβασης μπορούν να χρησιμοποιηθούν για την παροχή πρόσθετων στρωμάτων άδειας όσον αφορά συγκεκριμένες λειτουργίες του δικτύου. Για παράδειγμα, ένας συγκεκριμένος χρήστης μπορεί να έχει τη δυνατότητα να εκτελεί κάποιο έξυπνο συμβόλαιο, αλλά να μην επιτρέπεται να προσθέτει ένα νέο έξυπνο συμβόλαιο στην πλατφόρμα.

5.2.1.4 Ιδιωτικότητα και Εμπιστευτικότητα

Η συγκεκριμένη πλατφόρμα εξασφαλίζει την ιδιωτικότητα και την εμπιστευτικότητα μέσω της αρχιτεκτονικής καναλιών της (channel architecture). Τα κανάλια είναι περιορισμένες διαδρομές μηνυμάτων (messaging paths) που χρησιμοποιούνται για την ιδιωτικότητα των συναλλαγών συγκεκριμένου υποσυνόλου των μελών του δικτύου. Δηλαδή, μόνο οι χρήστες που συμμετέχουν σε ένα συγκεκριμένο κανάλι έχουν πρόσβαση στα δεδομένα των συναλλαγών και τα έξυπνα συμβόλαια αυτού του καναλιού. Ουσιαστικά, τα κανάλια χρησιμοποιούνται ώστε να δίνεται πρόσβαση σε συγκεκριμένες συναλλαγές και έξυπνα συμβόλαια μόνο στους χρήστες που ανήκουν στο εκάστοτε κανάλι. Το Hyperledger Fabric επιτρέπει στους χρήστες του να συμμετέχουν ταυτόχρονα σε πολλά διαφορετικά δίκτυα blockchain μέσω των καναλιών.

5.2.1.5 Ιδιωτικά Δεδομένα (Private Data)

Το Hyperledger Fabric δίνει τη δυνατότητα δημιουργίας συλλογών ιδιωτικών δεδομένων (private data collections), οι οποίες επιτρέπουν σε ένα συγκεκριμένο υποσύνολο των χρηστών ενός καναλιού να δημιουργεί και να αναζητά δεδομένα που δεν είναι ορατά από τους υπόλοιπους χρήστες του καναλιού. Έτσι αποφεύγεται η δημιουργία επιπλέον καναλιών στην περίπτωση που χρειάζεται να αποκρύβονται λίγα μόνο δεδομένα από κάποιους χρήστες.

5.2.1.6 Διαχείριση Κλειδιών

Κάθε χρήστης της πλατφόρμας διαθέτει ένα ζεύγος δημόσιου - ιδιωτικού κλειδιού, το οποίο χρησιμοποιείται για τη δημιουργία και την επαλήθευση ψηφιακών υπογραφών.

5.2.1.7 Συνάρτηση Κατακερματισμού

Η πλατφόρμα του Hyperledger Fabric χρησιμοποιεί κατά κύριο λόγο τη συνάρτηση κατακερματισμού Shake256, η οποία ανήκει στην οικογένεια των συναρτήσεων κατακερματισμού SHA-3, για τον υπολογισμό των απαραίτητων hashes.

5.2.1.8 Κρυπτονόμισμα (cryptocurrency)

Το Hyperledger Fabric δεν δημιουργήθηκε με κάποιο εγγενές κρυπτονόμισμα (native currency). Ωστόσο, η δημιουργία νομίσματος είναι εφικτή στη συγκεκριμένη πλατφόρμα μέσω της χρήσης των έξυπνων συμβολαίων.

5.2.1.9 Αλγόριθμος συναίνεσης

Ο αλγόριθμος συναίνεσης που χρησιμοποιείται στο Hyperledger Fabric δεν είναι σταθερός. Δηλαδή, ο δημιουργός κάθε δικτύου μπορεί να επιλέξει τον αλγόριθμο συναίνεσης που ταιριάζει καλύτερα στις ανάγκες της εφαρμογής. Επειδή η πλατφόρμα δεν έχει δικό της κρυπτονόμισμα, μπορούν να χρησιμοποιηθούν και μηχανισμοί συναίνεσης που δεν προϋποθέτουν την ύπαρξη κρυπτονομίσματος, γεγονός που σημαίνει ότι η πλατφόρμα μπορεί να υλοποιηθεί με περίπου τα ίδια λειτουργικά κόστη που έχει ένα κατανεμημένο σύστημα.

5.2.2 Εφαρμογές

Οι εφαρμογές που χρησιμοποιούνται από επιχειρήσεις έχουν τις ακόλουθες απαιτήσεις:

- πρέπει οι συμμετέχοντες να μπορούν να αναγνωριστούν εύκολα.
- τα δίκτυα που χρησιμοποιούνται πρέπει να είναι με άδεια (permissioned), ώστε να έχουν πρόσβαση στην εφαρμογή μόνο εξουσιοδοτημένοι από την επιχείρηση χρήστες.
- πρέπει να τηρείται η ιδιωτικότητα των συναλλαγών και των δεδομένων που αφορούν τις επιχειρηματικές συναλλαγές.

Το Hyperledger Fabric έχει σχεδιαστεί από την αρχή με στόχο τη χρησιμοποίησή του σε επιχειρηματικές εφαρμογές. Για αυτό το λόγο, χρησιμοποιεί permissioned δίκτυο και υπηρεσία διαχείρισης της ταυτότητας των χρηστών. Ταυτόχρονα, χρησιμοποιεί κανάλια (channels) και ιδιωτικά δεδομένα (private data) προκειμένου να εξασφαλίσει ότι τηρείται η ιδιωτικότητα των συναλλαγών.

Πιο συγκεκριμένα, το συγκεκριμένο framework είναι κατάλληλο για χρήση σε εφαρμογές, στις οποίες η ιδιωτικότητα είναι βασική απαίτηση. Λόγω της προσαρμόσιμης αρχιτεκτονικής του, μπορεί να διαμορφωθεί ώστε να ικανοποιήσει τις απαιτήσεις πολλών επιχειρησιακών εφαρμογών. Συνεπώς,

η πλατφόρμα μπορεί να υποστηρίξει εφαρμογές που σχετίζονται με τομείς όπως τα χρηματοοικονομικά, η ιατροφαρμακευτική περίθαλψη, ή ασφάλιση και η διαχείριση εφοδιαστικής αλυσίδας.

5.3 Hyperledger Iroha

Το Hyperledger Iroha [16] είναι ένα framework που στοχεύει στη δημιουργία μιας πλατφόρμας για την ανάπτυξη αξιόπιστων, ασφαλών και αποκεντρωμένων εφαρμογών που βασίζονται σε blockchain. Το Hyperledger Iroha είναι ένα έργο ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί δωρεάν, ενώ συντηρείται και βελτιώνεται από πολλούς προγραμματιστές παγκοσμίως. Το συγκεκριμένο framework μπορεί να εγκατασταθεί και να λειτουργήσει πολύ εύκολα σε Linux και Mac OS. Η δημιουργία νέων εφαρμογών μπορεί να υλοποιηθεί εύκολα μέσω της χρήσης έτοιμων βιβλιοθηκών που περιέχουν εντολές (commands) και ερωτήματα (queries). Το Hyperledger Iroha είναι ιδανικό για εφαρμογές που αφορούν την έκδοση πιστοποιητικών και τη διαχείριση πόρων.

5.3.1 Χαρακτηριστικά Hyperledger Iroha

Στην παρούσα ενότητα θα περιγραφούν τα κυριότερα χαρακτηριστικά του Hyperledger Iroha.

5.3.1.1 Βασικές Έννοιες

5.3.1.1.1 Λογαριασμός (Account)

Ένας λογαριασμός (account) είναι μια οντότητα του Hyperledger Iroha που έχει την εξουσιοδότηση να εκτελέσει ένα συγκεκριμένο σύνολο ενεργειών. Κάθε λογαριασμός ανήκει σε έναν από τους υπάρχοντες τομείς (domains) και έχει συγκεκριμένους ρόλους (roles).

5.3.1.1.2 Στοιχείο (Asset)

Το συγκεκριμένο framework χρησιμοποιεί assets για την αναπαράσταση κάθε μετρήσιμου αγαθού ή κάποιας αξίας. Δηλαδή, ένα asset μπορεί να αντιπροσωπεύει οποιαδήποτε τέτοιου είδους μονάδα, όπως ένα νόμισμα, μπάρες χρυσού ή ακόμα και μονάδα ακίνητης περιουσίας.

5.3.1.1.3 Τομέας (Domain)

Η ύπαρξη του τομέα (domain) στο Hyperledger Iroha στοχεύει στην ομαδοποίηση των accounts και assets. Για παράδειγμα, ένας τομέας μπορεί να αντιπροσωπεύει τους χρήστες και τα περιουσιακά στοιχεία ενός συγκεκριμένου οργανισμού του δικτύου του Iroha.

5.3.1.1.4 Δικαίωμα (Permission) και Ρόλος (Role)

Το Hyperledger Iroha χρησιμοποιεί ένα σύστημα ελέγχου πρόσβασης, το οποίο βασίζεται σε ρόλους, προκειμένου να περιορίσει τις ενέργειες των χρηστών του. Αυτό το σύστημα συμβάλλει σημαντικά στην υλοποίηση εφαρμογών στις οποίες συμμετέχουν ομάδες χρηστών που έχουν διαφορετικά επίπεδα πρόσβασης. Η συντήρηση της συγκεκριμένης πλατφόρμας περιλαμβάνει τη δημιουργία ρόλων και δικαιωμάτων. Αυτό μπορεί να συμβεί είτε κατά το πρώτο βήμα της ανάπτυξης του συστήματος Iroha, είτε κατά τη διάρκεια της λειτουργίας του.

Ο ρόλος (role) είναι μια αφαίρεση (abstraction) που περιέχει ένα σύνολο δικαιωμάτων (permissions).

Το δικαίωμα (permission) είναι ένας κανόνας που δίνει το δικαίωμα σε χρήστες να εκτελέσουν ένα συγκεκριμένο σύνολο εντολών και ερωτημάτων. Τα permissions, με εξαίρεση τα grantable permissions, δεν μπορούν να δοθούν απευθείας σε ένα λογαριασμό. Αντίθετα, στους λογαριασμούς δίνονται ρόλοι, δηλαδή συλλογές δικαιωμάτων.

Τα επιχορηγούμενα δικαιώματα (grantable permissions) μπορούν να δοθούν απευθείας σε έναν λογαριασμό. Ένας λογαριασμός που έχει ένα τέτοιο δικαίωμα επιτρέπεται να πραγματοποιήσει μια συγκεκριμένη ενέργεια εκ μέρους ενός άλλου λογαριασμού. Για παράδειγμα, αν ο account a δώσει στον account b άδεια να μεταφέρει τα assets του, τότε ο account b μπορεί να μεταφέρει τα assets του account a σε οποιοδήποτε άλλο account.

5.3.1.1.5 Εντολές (Commands)

Ένα από τα σημαντικότερα χαρακτηριστικά του Hyperledger Iroha framework είναι το ότι επιτρέπει στους χρήστες του να εκτελούν λειτουργίες, όπως η δημιουργία και η μεταφορά ψηφιακών assets και η εγγραφή λογαριασμών, χρησιμοποιώντας προ-ενσωματωμένες εντολές (commands) του συστήματος. Αυτό απαλλάσσει τους developers του συστήματος από την ανάγκη να χρησιμοποιούν έξυπνα συμβόλαια, γεγονός που τους επιτρέπει να ολοκληρώνουν απλές εργασίες γρηγορότερα.

Μια εντολή (command) ουσιαστικά έχει ως πρόθεση το να αλλάξει την κατάσταση (World State View) του συστήματος, μέσω της εκτέλεσης ενεργειών πάνω σε μία οντότητα (asset, account) του συστήματος.

Για παράδειγμα:

- Η εντολή Create Account (Δημιουργία Λογαριασμού) στοχεύει στη δημιουργία μιας οντότητας που θα μπορεί να αποθηκεύει υπογραφές και προσωπικά δεδομένα και να κάνει συναλλαγές (transactions) και ερωτήματα (queries).

- Η εντολή Grant Permission (Χορήγηση Δικαιώματος) στοχεύει στο να δώσει σε κάποιον άλλο λογαριασμό το δικαίωμα να εκτελέσει κάποια ενέργεια πάνω στον λογαριασμό του χρήστη που εκτέλεσε τη συγκεκριμένη εντολή.

Κάθε εντολή πρέπει να συμπεριληφθεί σε μια συναλλαγή (transaction) προκειμένου να εκτελεστεί.

5.3.1.1.6 Ερώτημα (Query)

Το ερώτημα (query) είναι ένα αίτημα (request) προς το δίκτυο του Hyperledger Iroha που δεν αλλάζει την κατάσταση του συστήματος. Δηλαδή, το query δεν μπορεί να τροποποιήσει το περιεχόμενο του blockchain. Μέσω της εκτέλεσης ενός ερωτήματος, ένας χρήστης μπορεί να ζητήσει δεδομένα, όπως τα assets του λογαριασμού του και το ιστορικό των συναλλαγών του.

Για παράδειγμα:

- Το ερώτημα GetTransactions στοχεύει στην ανάκτηση πληροφοριών που σχετίζονται με συγκεκριμένες συναλλαγές, οι οποίες προσδιορίζονται από τα hashes τους.
- Το ερώτημα GetAccountDetail στοχεύει στην ανάκτηση των λεπτομερειών (details) ενός λογαριασμού. Τα details ενός λογαριασμού αποτελούνται από ζεύγη κλειδιών - τιμών.

5.3.1.1.7 Συναλλαγή (Transaction)

Η συναλλαγή είναι ένα ταξινομημένο σύνολο εντολών που εφαρμόζεται ατομικά (atomically) στο blockchain. Στην περίπτωση που κάποια από τις εντολές της συναλλαγής δεν είναι έγκυρη, απορρίπτεται ολόκληρη η συναλλαγή.

Το Hyperledger Iroha επιτρέπει την ύπαρξη παρτίδων συναλλαγών (transaction batch), οι οποίες επιτρέπουν την αποστολή πολλών συναλλαγών στο δίκτυο, διατηρώντας ταυτόχρονα τη σειρά εκτέλεσής τους. Τα transaction batches μπορούν να περιέχουν συναλλαγές που προέρχονται από διαφορετικούς λογαριασμούς.

Μια συναλλαγή μπορεί να απαιτεί μία ή και περισσότερες υπογραφές προκειμένου να εκτελεστεί.

5.3.1.2 Πολιτική πρόσβασης στο δίκτυο του Hyperledger Iroha

Το Hyperledger Iroha blockchain χρησιμοποιεί δίκτυο με άδεια (permissioned). Αυτό σημαίνει ότι μόνο εξουσιοδοτημένα άτομα μπορούν να έχουν πρόσβαση στο σύστημα και να αλληλεπιδράσουν με αυτό. Ουσιαστικά, το συγκεκριμένο framework διαθέτει ένα ισχυρό σύστημα δικαιωμάτων, καθώς απαιτείται άδεια όχι μόνο για την ένταξη των χρηστών στο δίκτυο, αλλά και για την εκτέλεση εντολών και ερωτημάτων.

5.3.1.3 Διαχείριση Κλειδιών

Κάθε χρήστης της πλατφόρμας διαθέτει ένα ζεύγος δημόσιου - ιδιωτικού κλειδιού, το οποίο χρησιμοποιείται για τη δημιουργία και την επαλήθευση ψηφιακών υπογραφών.

Η διαχείριση των κλειδιών στη συγκεκριμένη πλατφόρμα γίνεται ως εξής:

- πρέπει να γίνει αλλαγή των προεπιλεγμένων κωδικών πρόσβασης που χρησιμοποιήθηκαν κατά τη διάρκεια της εγκατάστασης.
- χρειάζεται να εξασφαλιστεί ότι τα νέα κλειδιά παράγονται σε ένα ασφαλές περιβάλλον και ότι μόνο ο διαχειριστής του συστήματος έχει πρόσβαση σε αυτά τα ζεύγη κλειδιών.
- μετά την ανάπτυξη κλειδιών στους κόμβους του Iroha, πρέπει να διαγραφούν τα ιδιωτικά κλειδιά από τον κεντρικό υπολογιστή που χρησιμοποιήθηκε για την ανάπτυξη.

5.3.1.4 Κρυπτονόμισμα (cryptocurrency)

Το Hyperledger Iroha δεν δημιουργήθηκε με κάποιο εγγενές κρυπτονόμισμα (native cryptocurrency). Ωστόσο, η δημιουργία νομίσματος είναι εφικτή στη συγκεκριμένη πλατφόρμα μέσω της δημιουργίας νέου asset που θα αντιπροσωπεύει το κρυπτονόμισμα.

5.3.1.5 Βιβλιοθήκες (Libraries)

Το συγκεκριμένο framework διαθέτει βιβλιοθήκες που παρέχουν στους developers εργαλεία για την ανάπτυξη του συστήματος. Για παράδειγμα, οι βιβλιοθήκες περιέχουν εντολές (commands), ερωτήματα (queries) και τρόπους για την αποστολή μηνυμάτων στους κόμβους του Iroha.

Μέχρι στιγμής υπάρχουν οι εξής βιβλιοθήκες:

- Java Library
- Javascript Library
- Python Library
- iOS Swift Library

οι οποίες χρησιμοποιούνται για την ανάπτυξη κώδικα στην αντίστοιχη γλώσσα προγραμματισμού.

5.3.1.6 Αλγόριθμος συναίνεσης

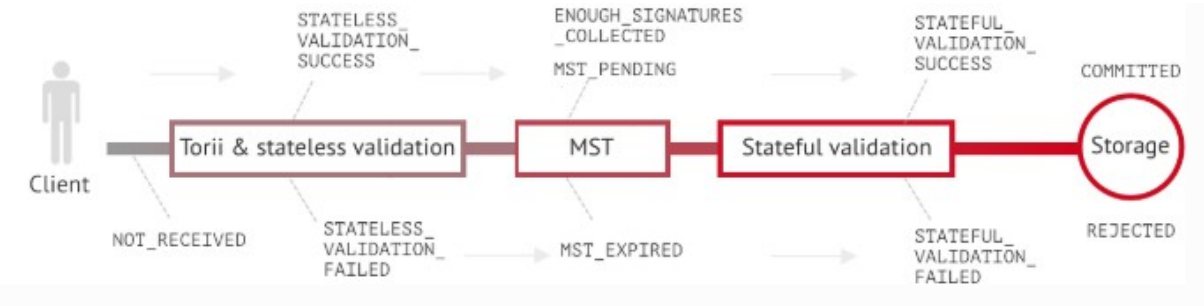
Ο αλγόριθμος συναίνεσης που χρησιμοποιείται στο Hyperledger Iroha είναι ο Yet Another Consensus (YAC). Ο συγκεκριμένος μηχανισμός επιτυγχάνει μικρή αδράνεια του συστήματος, η οποία οδηγεί σε υψηλή απόδοση συναλλαγών, ενώ ταυτόχρονα έχει μεγάλη δυνατότητα κλιμάκωσης και ανοχή σφάλματος σε σύγκρουση (Crash Fault Tolerance - CFT).

5.3.2 Λειτουργία Hyperledger Iroha

Μια συναλλαγή (transaction) πρέπει να περάσει με επιτυχία τα εξής στάδια προκειμένου να εκτελεστεί και να ενταχθεί στο blockchain:

- Stateless validation: ελέγχεται αν η συναλλαγή έχει διαμορφωθεί σωστά.
- Συγκέντρωση υπογραφών: στην περίπτωση που απαιτούνται περισσότερες από μία υπογραφές για την εκτέλεση της συναλλαγής, γίνεται έλεγχος για το αν έχει συγκεντρωθεί ο απαιτούμενος αριθμός υπογραφών.
- Stateful validation: ελέγχεται αν η συναλλαγή είναι σύμφωνη με την κατάσταση του συστήματος (World State View) τη δεδομένη χρονική στιγμή. Για παράδειγμα, σε μία συναλλαγή μεταφοράς χρημάτων ελέγχεται αν το ποσό που πρόκειται να μεταφερθεί υπάρχει στο λογαριασμό.

Τα στάδια από τα οποία πρέπει να περάσει μια συναλλαγή φαίνονται σχηματικά στην ακόλουθη εικόνα:



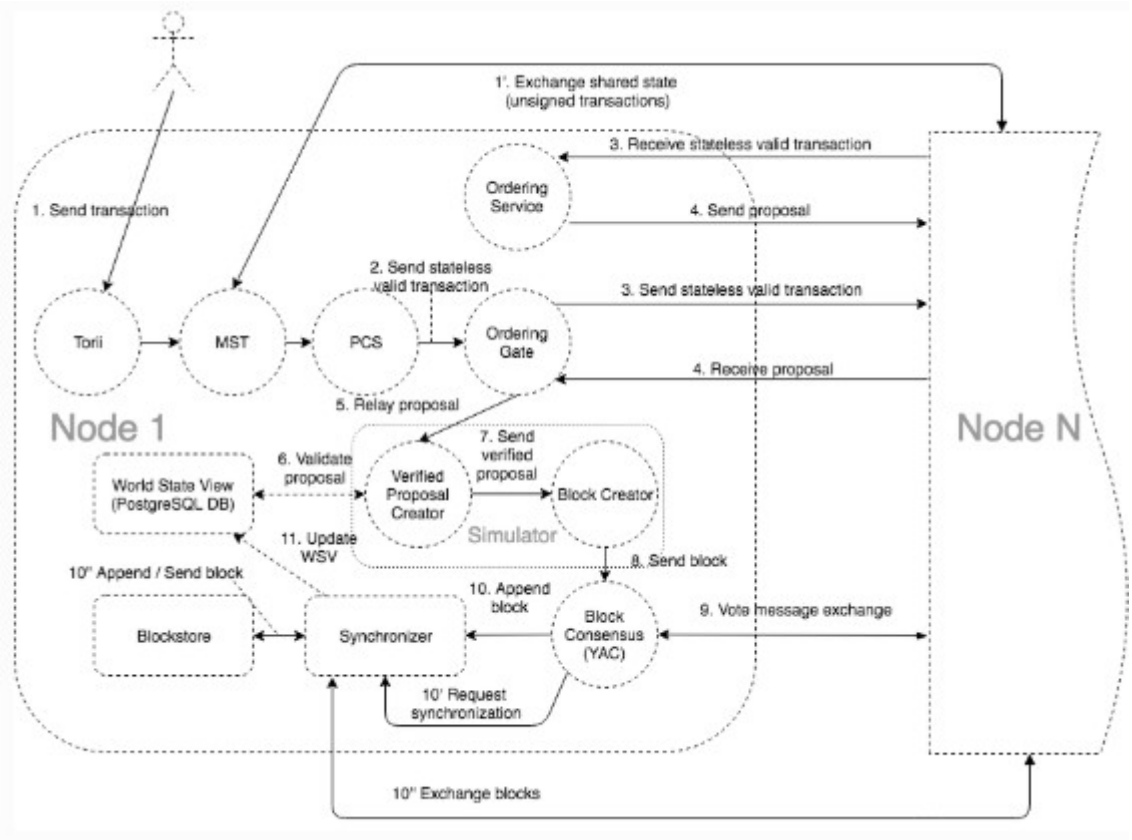
Εικόνα 8: Μια συναλλαγή πρέπει να περάσει με επιτυχία τα στάδια: 1) stateless validation, 2) συγκέντρωση υπογραφών, 3) stateful validation, προκειμένου να εκτελεστεί.

Στη συνέχεια περιγράφονται τα βασικά components του Hyperledger Iroha που παρέχουν την επικοινωνία μεταξύ των κόμβων του και συμβάλλουν στην πραγματοποίηση των συναλλαγών:

- Torii: είναι το component μέσω του οποίου οι clients αλληλεπιδρούν με το Iroha
- Multisignature Transactions Processor (MST Processor): ο επεξεργαστής συναλλαγών πολλαπλών υπογραφών στέλνει στους κόμβους (nodes) συναλλαγές πολλαπλών υπογραφών (multisignature transactions) μέχρι να επιτευχθεί ο απαραίτητος αριθμός υπογραφών.
- Peer Communication Service: λειτουργεί σαν ένας διαμεσολαβητής που μεταδίδει συναλλαγές από το Torii στο Ordering Gate μέσω του Multisignature Transactions Processor.
- Ordering Gate: μεταφέρει στο Ordering Service τις συναλλαγές που έχουν περάσει επιτυχώς το στάδιο της stateless validation. Στη συνέχεια λαμβάνει από το Ordering Service proposals (προτάσεις για blocks) και τις στέλνει στον Simulator προκειμένου να περάσουν το στάδιο της stateful validation.
- Ordering Service: συνδυάζει συναλλαγές που έχουν περάσει με επιτυχία το στάδιο της stateless validation σε μία πρόταση (proposal). Κάθε κόμβος (node) έχει τη δικιά του Ordering Service.
- Verified Proposal Creator: εκτελεί stateful validation για τις συναλλαγές που περιέχονται στην πρόταση που λαμβάνει από το Ordering Service. Με βάση τις συναλλαγές που περνάνε επιτυχώς την stateful validation, ο Verified Proposal Creator δημιουργεί και στέλνει μια επικυρωμένη πρόταση στον Block Creator.
- Block Creator: δημιουργεί ένα block από συναλλαγές που έχουν περάσει επιτυχώς τα στάδια των stateless και stateful validation. Ο Block Creator σε συνδυασμό με τον Verified Proposal Creator αποτελούν τον προσομοιωτή (Simulator).

- Block Consensus (YAC): εξασφαλίζει τη συναίνεση των κόμβων του δικτύου όσον αφορά την τωρινή κατάσταση του συστήματος.
- Synchronizer: προσθέτει στις αλυσίδες που έχουν αποθηκευμένες οι κόμβοι τα blocks που μπορεί να λείπουν από αυτές.
- World State View (WSV): αντικατοπτρίζει την τρέχουσα κατάσταση του συστήματος. Ουσιαστικά αποτελεί ένα στιγμιότυπο (snapshot) του συστήματος. Για παράδειγμα, περιέχει πληροφορία σχετικά με τα assets που έχει ένας λογαριασμός αυτή τη στιγμή, αλλά δεν περιέχει καμία πληροφορία σχετικά με τις συναλλαγές που έχουν πραγματοποιηθεί.
- Ametsuchi Blockstore: αποθηκεύει τα blocks και το World State View.

Τα components του Hyperledger Iroha που περιγράφηκαν καθώς και οι αλληλεπιδράσεις τους παρουσιάζονται σχηματικά στην ακόλουθη εικόνα.



Εικόνα 9: Τα components από τα οποία αποτελείται το δίκτυο του Hyperledger Iroha και οι μεταξύ τους αλληλεπιδράσεις.

5.3.3 Εφαρμογές

Από τα παραπάνω γίνεται σαφές ότι το Hyperledger Iroha μπορεί να υποστηρίξει πολλές και διαφορετικού είδους αποκεντρωμένες εφαρμογές που στηρίζονται σε blockchain για την υλοποίησή τους. Πιο συγκεκριμένα, αυτό το framework είναι κατάλληλο για τη διαχείριση ψηφιακών στοιχείων και ταυτοτήτων και τη σειριοποίηση δεδομένων (data serialization).

Στη συνέχεια περιγράφονται μερικά παραδείγματα τομέων και εφαρμογών που μπορούν να υλοποιηθούν επιτυχώς μέσω του Hyperledger Iroha.

- Υγειονομική Περίθαλψη και Εκπαιδευτικά Πιστοποιητικά

  Το Hyperledger Iroha μπορεί να ενσωματώσει στο σύστημα πολλαπλές αρχές πιστοποίησης, όπως εκπαιδευτικά ιδρύματα και νοσοκομεία. Το μοντέλο δικαιωμάτων του συγκεκριμένου framework μπορεί να εξασφαλίσει ότι η χορήγηση πιστοποιητικών πραγματοποιείται μόνο από εξουσιοδοτημένες αρχές πιστοποίησης.

  Στην περίπτωση της ιατροφαρμακευτικής περίθαλψης, ένα νοσοκομείο μπορεί να καταχωρηθεί στο σύστημα ως ένα αυτόνομο domain. Αυτό το domain θα έχει εργαζομένους που ο καθένας θα έχει ορισμένους ρόλους, όπως για παράδειγμα γιατρός και νοσοκόμα. Κάθε ασθενής του νοσοκομείου θα έχει ένα account που θα περιέχει το πλήρες ιατρικό ιστορικό του με μορφή key - value. Τα permissions θα υπάγονται σε ρόλους με τρόπο που θα εξασφαλίζει ότι μόνο οι πιστοποιημένοι γιατροί και ο ίδιος ο ασθενής θα έχουν πρόσβαση στο ιστορικό του. Στην περίπτωση που ένας ασθενής θελήσει να μοιραστεί το ιστορικό του με ένα άλλο ιατρικό ίδρυμα, μπορεί να παραχωρήσει δικαιώματα πρόσβασης στο λογαριασμό του στο συγκεκριμένο ίδρυμα.

  Με παρόμοιο τρόπο μπορεί να γίνει η έκδοση πιστοποιητικών από εκπαιδευτικά ιδρύματα και η παραχώρηση δικαιωμάτων πρόσβασης σε αυτά σε πιθανούς εργοδότες.

- Μεταβίβαση περιουσιακών στοιχείων
  Στην περίπτωση που ο χρήστης a θέλει να μεταβιβάσει ένα asset στο χρήστη b έναντι αμοιβής, μπορεί να δημιουργήσει μία multi-signature συναλλαγή που θα περιέχει δύο εντολές και θα πρέπει να υπογραφεί και από τους δύο χρήστες. Οι εντολές αφορούν τη μεταφορά του asset από το χρήστη a στο χρήστη b και τη μεταφορά ενός ποσού από το χρήστη b στο χρήστη a.

- Εφοριακό σύστημα
  Οι multi-signatures συναλλαγές μπορούν να συμβάλλουν σημαντικά στην εξάλειψη της φοροδιαφυγής. Κάθε συναλλαγή που αφορά ένα συγκεκριμένο τομέα μπορεί να πραγματοποιηθεί ως συναλλαγή πολλαπλών υπογραφών, όπου η μία υπογραφή προέρχεται από το χρήστη, ενώ η δεύτερη από ειδικούς κόμβους φορολόγησης.

5.4 Συγκριτική ανάλυση και συμπεράσματα

Στην παρούσα ενότητα θα γίνει συγκριτική ανάλυση των frameworks Ethereum, Hyperledger Fabric και Hyperledger Iroha, ώστε να αποφασιστεί το καταλληλότερο framework για την έκδοση και επικύρωση πιστοποιητικών από εκπαιδευτικά ιδρύματα.

Στον ακόλουθο πίνακα φαίνονται οι πιο σημαντικές διαφορές των παραπάνω frameworks.

	Ethereum	**Hyperledger Fabric**	**Hyperledger Iroha**
Έξυπνα συμβόλαια	✔	✔	—
Ενσωματωμένες εντολές	—	—	✔
Δίκτυο	permission-less	permissioned	permissioned
Αλγόριθμος Συναίνεσης	Αρχικά: Proof of Work (PoW) Σύντομα: Proof of Stake (PoS)	ο δημιουργός κάθε δικτύου μπορεί να επιλέξει τον αλγόριθμο συναίνεσης που ταιριάζει καλύτερα στις ανάγκες της εκάστοτε εφαρμογής	Yet Another Consensus (YAC)
Native currency	Ether	όχι αλλά μπορεί να δημιουργηθεί μέσω των έξυπνων συμβολαίων	όχι αλλά μπορεί να δημιουργηθεί μέσω των assets
Γλώσσες προγραμματισμού	γλώσσες προγραμματισμού που βασίζονται σε υφιστάμενες γλώσσες	γλώσσες προγραμματισμού γενικού σκοπού, π.χ. Java και Go	Java, Javascript, Python και Swift
Δικαιώματα	μέσω των έξυπνων συμβολαίων	Channels και Private Data	Permissions και Roles
Εφαρμογές	κυρίως εφαρμογές που αυτοματοποιούν την αλληλεπίδραση μεταξύ peers ή διευκολύνουν τη συντονισμένη δράση ομάδας σε ένα δίκτυο, π.χ. ο συντονισμός της λειτουργίας ενός peer-to-peer marketplace και η αυτοματοποίηση περίπλοκων οικονομικών συμβολαίων	επιχειρηματικές εφαρμογές, π.χ. χρηματοοικονομικά και διαχείριση εφοδιαστικής αλυσίδας	κυρίως εφαρμογές που αφορούν διαχείριση ψηφιακών στοιχείων, π.χ. ιατρικά και εκπαιδευτικά πιστοποιητικά και μεταβίβαση περιουσιακών στοιχείων

Πίνακας 1: Συγκριτική ανάλυση Ethereum, Hyperledger Fabric και Hyperledger Iroha

Από τον παραπάνω πίνακα γίνεται σαφές ότι το Hyperledger Iroha είναι το καταλληλότερο framework για την υλοποίηση εφαρμογής που αφορά την έκδοση και επικύρωση πιστοποιητικών από εκπαιδευτικά ιδρύματα με χρήση της blockchain τεχνολογίας.
Αυτό συμβαίνει για τους ακόλουθους λόγους:

- το Hyperledger Iroha είναι κατάλληλο για εφαρμογές που σχετίζονται με τη διαχείριση ψηφιακών στοιχείων.
- το δίκτυο είναι permissioned και άρα είναι εφικτό να έχουν πρόσβαση σε αυτό μόνο εξουσιοδοτημένα εκπαιδευτικά ιδρύματα, σπουδαστές και εργοδότες.

- ο αλγόριθμος συναίνεσης που χρησιμοποιείται είναι ο Yet Another Consensus, ο οποίος είχε θεωρηθεί ο καταλληλότερος για τη συγκεκριμένη εφαρμογή στο κεφάλαιο "Προτεινόμενη υλοποίηση Blockchain στην εκπαίδευση" της παρούσας διπλωματικής εργασίας.
- οι χρήστες θα μπορούν να δίνουν πρόσβαση στα πιστοποιητικά τους σε πιθανούς μελλοντικούς εργοδότες μέσω των ενσωματωμένων εντολών.
- η ανάπτυξη νέων εντολών θα είναι αρκετά εύκολη μέσω της χρήσης των υφιστάμενων γλωσσών προγραμματισμού Java, Javascript, Python και Swift.

6 Υλοποίηση Εφαρμογής

Στο σημείο αυτό έχουν συζητηθεί λεπτομερώς τόσο η αρχιτεκτονική της εφαρμογής που θα υλοποιηθεί, όσο και το framework που θα χρησιμοποιηθεί για την ανάπτυξή της. Ο κύριος στόχος του παρόντος κεφαλαίου είναι να περιγράψει την υλοποίηση της εν λόγω εφαρμογής. Πιο συγκεκριμένα, το κεφάλαιο αυτό αποτελεί έναν οδηγό για την εγκατάσταση του Hyperledger Iroha, την ανάπτυξη της εφαρμογής με χρήση της γλώσσα προγραμματισμού Python και τον έλεγχο της λειτουργίας της εφαρμογής.

6.1 Εγκατάσταση Hyperledger Iroha

Σε αυτήν την ενότητα παρουσιάζεται η διαδικασία που πρέπει να ακολουθηθεί για την εγκατάσταση του Hyperledger Iroha μέσω της πλατφόρμας Docker.

6.1.1 Docker

Το Docker [17] είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που υλοποιεί Εικονοποίηση (Virtualization) σε επίπεδο λειτουργικού συστήματος (operating system - OS). Ουσιαστικά, το Docker προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες Περιοχές Χρήστη (User Spaces) που ονομάζονται Software Containers. Επίσης, επιτρέπει σε ανεξάρτητα software containers να εκτελούνται στο ίδιο λειτουργικό σύστημα. Συνεπώς είναι πιο "ελαφρύ" από τις εικονικές μηχανές (virtual machines).

Για την εγκατάσταση του Hyperledger Iroha πρέπει να εγκατασταθούν τα docker και docker-compose. Το Docker Desktop, το οποίο είναι διαθέσιμο για Windows και Mac, περιλαμβάνει το docker-compose.

Για την εγκατάσταση του Docker Desktop πρέπει να γίνει λήψη αυτού από την ιστοσελίδα του Docker[1].

Εικόνα 10: Λήψη Docker Desktop

Για να ελεγχθεί ότι η εγκατάσταση ολοκληρώθηκε με επιτυχία μπορούν να πραγματοποιηθούν τα ακόλουθα βήματα μέσα από ένα τερματικό (terminal):

- Ο έλεγχος της version του Docker γίνεται μέσω της εντολής: docker --version.

Εικόνα 11: Έλεγχος version του Docker

- Ο έλεγχος της version του Docker-compose γίνεται μέσω της εντολής: docker-compose --version.

Εικόνα 12: Έλεγχος version του Docker- compose

1 https://www.docker.com/products/container-runtime

- Για να ελεγχθεί η σωστή λειτουργία του Docker, μπορεί να χρησιμοποιηθεί η εντολή: docker run hello-world. Αν η απάντηση είναι "Hello from Docker! This message shows that your installation appears to be working correctly.", τότε το Docker λειτουργεί σωστά.

```
silvia@MBP-Silbia ~ % docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Εικόνα 13: Εντολή: docker run hello-world

6.1.2 Python

Το Hyperledger Iroha framework είναι συμβατό με έκδοση Python 3 και νεότερη. Συνεπώς, πρέπει να εγκατασταθεί αντίστοιχη έκδοση της γλώσσα προγραμματισμού Python.

6.1.3 Hyperledger Iroha

Μετά την εγκατάσταση του Docker, πρέπει να ακολουθηθούν τα εξής βήματα για την εγκατάσταση του Hyperledger Iroha:

1) Πρέπει να γίνει λήψη του master branch του Iroha από το Iroha Repository[2].
2) Στο φάκελο του iroha master χρειάζεται να δημιουργηθεί ο νέος φάκελος: conf.
3) Πρέπει να μετακινηθούν τα εξής αρχεία από το φάκελο "example" στο φάκελο "conf":
    - admin@test.priv
    - admin@test.pub
    - config.docker
    - genesis.block
    - node0.priv
    - node0.pub
    - test@test.priv
    - test@test.pub
4) Στο φάκελο "conf", πρέπει να δημιουργηθεί ένα αρχείο με όνομα "iroha.yaml" και τον ακόλουθο κώδικα.

```
version: '3.4'
```

2 https://github.com/hyperledger/iroha

```
services:

services:

 some-postgres:
  image: postgres:9.5
  hostname: some-postgres
  restart: always
  ports:
   - 5432:5432
  volumes:
   - iroha-db:/var/lib/postgresql/data
  environment:
   POSTGRES_USER: postgres
   POSTGRES_PASSWORD: mysecretpassword
   PGDATA: /tmp
  command: -c 'max_prepared_transactions=100'
  networks:
   - iroha-network

 iroha:
  image: hyperledger/iroha:1.0.0_rc3
  hostname: iroha
  restart: always
  tty: true
  working_dir: /opt/iroha
  ports:s
   - 50051:50051
   - 55552:55552
   - 20000:20000
  volumes:
   - blockstore2:/tmp/block_store
   - /home/user/conf:/opt/iroha:delegated
  environment:
   IROHA_POSTGRES_HOST: some-postgres
   IROHA_POSTGRES_PASSWORD: mysecretpassword
   IROHA_POSTGRES_USER: postgres
   IROHA_POSTGRES_PORT: 5432
   KEY: node0
  cap_add:
   - SYS_PTRACE
  security_opt:
   - seccomp:unconfined
  depends_on:
   - some-postgres
  networks:
   - iroha-network


networks:
```

```
  iroha-network:
    driver: bridge

volumes:
  blockstore2:
  iroha-db:
```

Το κομμάτι /home/user/conf της γραμμής /home/user/conf:/opt/iroha:delegate πρέπει να αλλάξει ώστε να περιέχει το σωστό path.

5) Πρέπει να εκτελεστεί η εντολή: sudo docker-compose -f iroha.yaml up -d. Η εντολή αυτή πρέπει να εκτελεστεί μετά τη δημιουργία του genesis block που θα περιγραφεί σε επόμενη ενότητα.

```
silvia@MBP-Silbia conf % sudo docker-compose -f iroha.yaml up -d
Creating network "conf_iroha-network" with driver "bridge"
Creating volume "conf_blockstore2" with default driver
Creating volume "conf_iroha-db" with default driver
Pulling some-postgres (postgres:9.5)...
9.5: Pulling from library/postgres
619014d83c02: Pull complete
7ec0fe6664f6: Pull complete
9ca7ba8f7764: Pull complete
9e1155d037e2: Pull complete
febcfb7f8870: Pull complete
8c78c79412b5: Pull complete
5a35744405c5: Pull complete
27717922e067: Pull complete
cebd6ec05a24: Pull complete
3a12e41a30c5: Pull complete
7e6fd4a8e561: Pull complete
9c470cc1fc33: Pull complete
d8895387da1b: Pull complete
d3ac611f0469: Pull complete
Digest: sha256:54a9f2c39fa0c7903bfb4d32568820a45bceb60a491e6378d35cc74b65be7b1c
Status: Downloaded newer image for postgres:9.5
Pulling iroha (hyperledger/iroha:1.0.0_rc3)...
1.0.0_rc3: Pulling from hyperledger/iroha
7b722c1070cd: Pull complete
5fbf74db61f1: Pull complete
ed41cb72e5c9: Pull complete
7ea47a67709e: Pull complete
ed74c74439fe: Pull complete
bc56df06977e: Pull complete
8be2885b968c: Pull complete
31f58f6fbcd0: Pull complete
8d3d0729709b: Pull complete
022a0038b384: Pull complete
Digest: sha256:8111fe62f292f91050eca33363ea2bd12cc0b7b5d3b9508301fcf2ad57232ae7
Status: Downloaded newer image for hyperledger/iroha:1.0.0_rc3
Creating conf_some-postgres_1 ... done
Creating conf_iroha_1         ... done
silvia@MBP-Silbia conf %
```

Εικόνα 14: Εντολή: sudo docker-compose -f iroha.yaml up -d

6.1.4 Pip και Iroha package

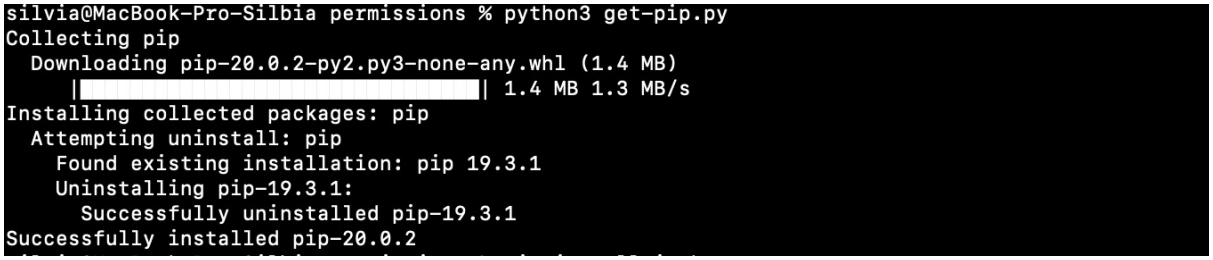
Το pip [18] είναι ένα πρότυπο σύστημα διαχείρισης πακέτων που χρησιμοποιείται για την εγκατάσταση και τη διαχείριση πακέτων λογισμικού που είναι γραμμένα σε Python.

Προκειμένου να εγκατασταθεί το pip πρέπει να πραγματοποιηθούν τα ακόλουθα βήματα μέσα από ένα τερματικό (terminal):

- curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

Εικόνα 15: Εντολή: curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

- python3 get-pip.py



Εικόνα 16: Εντολή: python3 get-pip.py

Στη συνέχεια, προκειμένου να εγκατασταθεί το package για το Iroha πρέπει να εκτελεστεί η εξής εντολή:

- pip install iroha

Εικόνα 17: Εντολή: pip install iroha

6.2 Επανεκκίνηση Hyperledger Iroha

Σε αυτήν την ενότητα παρουσιάζεται η διαδικασία που πρέπει να ακολουθηθεί σε περίπτωση που χρειαστεί να γίνει επανεκκίνηση του blockchain κατά τη διάρκεια ανάπτυξης και υλοποίησης της εφαρμογής.

Πιο συγκεκριμένα, προκειμένου να διαγραφεί το υπάρχον blockchain και να δημιουργηθεί εκ νέου, πρέπει να εκτελεστούν οι εξής εντολές μέσα από ένα τερματικό (terminal):

- Μέσω της εντολής sudo docker-compose -f iroha.yaml down απεγκαθίσταται το υπάρχον Hyperledger Iroha blockchain.

```
silvia@MacBook-Pro-Silbia conf % sudo docker-compose -f iroha.yaml down
Password:
Stopping conf_iroha_1         ... done
Stopping conf_some-postgres_1 ... done
Removing conf_iroha_1         ... done
Removing conf_some-postgres_1 ... done
Removing network conf_iroha-network
```

Εικόνα 18: Εντολή: sudo docker-compose -f iroha.yaml down

- Μέσω της εντολής docker volume ls καταγράφονται τα volumes του συστήματος που εξακολουθούν να είναι ενεργά.

```
silvia@MacBook-Pro-Silbia conf % docker volume ls
DRIVER              VOLUME NAME
local               conf_blockstore2
local               conf_iroha-db
```

Εικόνα 19: Εντολή: docker volume ls

- Για τη διαγραφή των παραπάνω volumes χρησιμοποιούνται οι εντολές:
  - docker volume rm conf_blockstore2
  - docker volume rm conf_iroha-db

```
silvia@MacBook-Pro-Silbia conf % docker volume rm conf_blockstore2
conf_blockstore2
silvia@MacBook-Pro-Silbia conf % docker volume rm conf_iroha-db
conf_iroha-db
```

Εικόνα 20: Εντολές: docker volume rm conf_blockstore2 και docker volume rm conf_iroha-db

- Μέσω της εντολής sudo docker-compose -f iroha.yaml up -d δημιουργείται το νέο blockchain.

```
silvia@MacBook-Pro-Silbia conf % sudo docker-compose -f iroha.yaml up -d
Creating network "conf_iroha-network" with driver "bridge"
Creating volume "conf_blockstore2" with default driver
Creating volume "conf_iroha-db" with default driver
Creating conf_some-postgres_1 ... done
Creating conf_iroha_1         ... done
```

Εικόνα 21: Εντολή: sudo docker-compose -f iroha.yaml up -d

6.3 Ανάπτυξη και Λεπτομέρειες Υλοποίησης

Στην παρούσα ενότητα θα περιγραφεί η υλοποίηση της εν λόγω εφαρμογής καθώς και η λογική πίσω από την ανάπτυξή της.

6.3.1 Λειτουργία Εφαρμογής

Σε αυτή την ενότητα θα παρουσιαστούν οι βασικότερες λειτουργικότητες της εφαρμογής που υλοποιήθηκε.

6.3.1.1 Λειτουργίες Χρηστών Συστήματος

Η συγκεκριμένη εφαρμογή αφορά τα εξής:

- την έκδοση επικυρωμένων πιστοποιητικών από εκπαιδευτικά ιδρύματα σε σπουδαστές.
- την παραχώρηση δικαιωμάτων πρόσβασης στα πιστοποιητικά καθώς και στην απόδειξη της γνησιότητάς τους, από σπουδαστές σε πιθανούς μελλοντικούς εργοδότες.

Προκειμένου να εκδώσει ένα πιστοποιητικό σε έναν σπουδαστή, το πανεπιστήμιο πρέπει να πραγματοποιήσει τα ακόλουθα βήματα:

1) Να μεταφέρει στο σπουδαστή ένα asset με όνομα την κατηγορία του πιστοποιητικού, για παράδειγμα bachelor, και ποσότητα 1, το οποίο θα φανερώνει την κατοχή αντίστοιχου πιστοποιητικού από το σπουδαστή.
2) Να θέσει τις λεπτομέρειες του λογαριασμού (account details) του φοιτητή ως εξής:
    - Detail που αφορά το uri στο οποίο μπορεί να βρεθεί το πιστοποιητικό και θα έχει ως τιμή (value) το εν λόγω uri .
    - Detail που αφορά το hash του πιστοποιητικού και θα έχει ως τιμή (value) το hash.
    - Detail που αφορά το hash της συναλλαγής (transaction) κατά την οποία το πανεπιστήμιο μετέφερε το asset στο σπουδαστή και θα έχει ως τιμή (value) το αντίστοιχο transaction hash.

Προκειμένου να παραχωρήσει δικαιώματα πρόσβασης στο πιστοποιητικό σε έναν πιθανό μελλοντικό εργοδότη και να αποδείξει τη γνησιότητά του, ο σπουδαστής πρέπει να πραγματοποιήσει τα εξής βήματα:

1) Να ελέγξει τις λεπτομέρειες του λογαριασμού του (account details) που έχει θέσει το εκάστοτε εκπαιδευτικό ίδρυμα, ώστε να βρει το hash της συναλλαγής έκδοσης του πιστοποιητικού.
2) Να δώσει στον εκάστοτε πιθανό εργοδότη το hash της συγκεκριμένης συναλλαγής. Για το πετύχει μπορεί να θέσει τις λεπτομέρειες του λογαριασμού (account details) του εργοδότη ως εξής:
    - Detail που αφορά το hash της συναλλαγής (transaction) κατά την οποία το πανεπιστήμιο μετέφερε το asset στο σπουδαστή και θα έχει ως τιμή (value) το αντίστοιχο transaction hash.

Προκειμένου να ελέγξει το hash του πιστοποιητικού, ο εργοδότης πρέπει να πραγματοποιήσει τα εξής βήματα:

1. Να ελέγξει τις λεπτομέρειες του λογαριασμού του (account details) που έχει θέσει ο εκάστοτε σπουδαστής, ώστε να βρει το hash της συναλλαγής έκδοσης του πιστοποιητικού.
2. Να κάνει query με το hash της συναλλαγής, ώστε να του την επιστρέψει το σύστημα.

Βέβαια, προκειμένου η εφαρμογή να είναι ολοκληρωμένη, οι χρήστες του συστήματος θα πρέπει να έχουν τη δυνατότητα να εκτελέσουν και ορισμένες επιπλέον λειτουργίες, οι οποίες θα περιγραφούν σε επόμενη ενότητα. Για παράδειγμα, θα μπορούν να ορίζουν τις λεπτομέρειες του λογαριασμού τους, να βλέπουν τα assets που έχουν στην κατοχή τους και να έχουν πρόσβαση σε όλες τις συναλλαγές που έχουν εκτελέσει στο παρελθόν.

6.3.1.2 Λειτουργίες Διαχειριστή Συστήματος

Ο διαχειριστής (administrator) θα είναι υπεύθυνος για την εκτέλεση λειτουργιών που αφορούν την εύρυθμη λειτουργία του συστήματος.

Ο βασικότερος ρόλος του είναι η δημιουργία λογαριασμών για νέους χρήστες. Για παράδειγμα, όταν ένας μη εγγεγραμμένος χρήστης κάνει αίτηση για δημιουργία λογαριασμού, ο διαχειριστής θα πρέπει να εγκρίνει το νέο λογαριασμό και να τον προσθέσει στο blockchain. Η διαδικασία αυτή βέβαια μπορεί να γίνεται και αυτοματοποιημένα με χρήση του ζεύγους κλειδιών του διαχειριστή.

Ο διαχειριστής του συστήματος θα έχει τη δυνατότητα να εκτελέσει επιπρόσθετες λειτουργίες, όπως δημιουργία νέου asset, νέου τομέα (domain) και νέου ρόλου (role). Οι λειτουργίες αυτές θα περιγραφούν αναλυτικά σε επόμενη ενότητα.

6.3.2 Βασικά Στοιχεία Εφαρμογής

Σε αυτή την ενότητα θα παρουσιαστούν τα βασικά στοιχεία της εφαρμογής που αφορά την έκδοση εκπαιδευτικών πιστοποιητικών.

6.3.2.1 Λογαριασμοί (Accounts)

Με βάση τη λειτουργικότητα της εφαρμογής, οι βασικοί χρήστες του συστήματος μπορούν να ομαδοποιηθούν ως εξής:

- εκπαιδευτικά ιδρύματα (universities)
- σπουδαστές (students)
- εργοδότες (employers)

Όπως αναφέρθηκε παραπάνω, χρειάζεται, επίσης, να υπάρχει χρήστης με την ιδιότητα του διαχειριστή του συστήματος, ο οποίος θα είναι υπεύθυνος για την εύρυθμη λειτουργία του συστήματος.

6.3.2.2 Στοιχεία (Assets)

Στη συγκεκριμένη εφαρμογή, τα πιστοποιητικά (certificates) μπορούν να αναπαρασταθούν μέσω των assets. Για κάθε διαφορετικό είδος πιστοποιητικού θα δημιουργηθεί ένα διαφορετικό asset με όνομα το είδος του πιστοποιητικού, όπως το master.

Επειδή τα βασικότερα πιστοποιητικά που μπορεί να εκδώσει ένα εκπαιδευτικό ίδρυμα αφορούν την περάτωση των προπτυχιακών, μεταπτυχιακών και διδακτορικών σπουδών, το σύστημα θα ξεκινήσει τη λειτουργία του με τα εξής assets:

- bachelor
- master
- phd,

ενώ θα δίνεται η δυνατότητα τόσο στα πανεπιστημιακά ιδρύματα όσο και στους διαχειριστές του συστήματος να δημιουργήσουν νέα assets.

6.3.2.3 Δικαιώματα (Permissions)

Κάθε ομάδα χρηστών έχει το δικαίωμα να εκτελέσει ένα συγκεκριμένο σύνολο λειτουργιών. Συνεπώς, θα πρέπει να έχει και ένα συγκεκριμένο σύνολο δικαιωμάτων (permissions) που αφορούν την εκτέλεση εντολών (commands).

Τα εκπαιδευτικά ιδρύματα θα πρέπει να έχουν τα ακόλουθα permissions, προκειμένου να μπορούν να εκτελέσουν όλες τις απαραίτητες ενέργειες για την έκδοση πιστοποιητικών και τη διαχείριση του λογαριασμού τους:

- **can_get_all_txs:** επιτρέπει την εύρεση μιας συναλλαγής με βάση το hash της
- **can_get_all_acc_detail:** επιτρέπει την πρόσβαση στις λεπτομέρειες (details) ενός λογαριασμού (account)
- **can_get_my_account:** επιτρέπει την πρόσβαση σε όλες τις πληροφορίες του λογαριασμού του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast:** επιτρέπει την εύρεση των assets που διαθέτει ο χρήστης που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query) και που σχετίζονται με ένα συγκεκριμένο asset
- **can_grant_can_set_my_account_detail:** επιτρέπει στο χρήστη που διαθέτει αυτό το δικαίωμα, να δώσει σε άλλους χρήστες ή να ανακαλέσει το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details)
- **can_transfer:** επιτρέπει στο χρήστη να μεταφέρει assets από το λογαριασμό του σε έναν άλλο
- **can_receive:** επιτρέπει στο χρήστη να λαμβάνει assets από άλλους λογαριασμούς
- **can_get_my_acc_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query)
- **can_add_asset_qty:** επιτρέπει στο χρήστη που δημιούργησε τη συναλλαγή να αυξήσει την ποσότητα ενός συγκεκριμένου asset στο λογαριασμό του
- **can_create_asset:** επιτρέπει τη δημιουργία ενός νέου asset

Οι φοιτητές θα πρέπει να έχουν τα ακόλουθα permissions, προκειμένου να μπορούν να εκτελέσουν όλες τις απαραίτητες ενέργειες για την παραχώρηση δικαιωμάτων πρόσβασης στα πιστοποιητικά τους σε άλλους χρήστες και τη διαχείριση του λογαριασμού τους:

- **can_get_all_txs:** επιτρέπει την εύρεση μιας συναλλαγής με βάση το hash της
- **can_get_my_acc_detail:** επιτρέπει την πρόσβαση στις λεπτομέρειες (details) του λογαριασμού (account) του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_my_account:** επιτρέπει την πρόσβαση σε όλες τις πληροφορίες του λογαριασμού του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast:** επιτρέπει την εύρεση των assets που διαθέτει ο χρήστης που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query) και που σχετίζονται με ένα συγκεκριμένο asset
- **can_grant_can_set_my_account_detail:** επιτρέπει στο χρήστη που διαθέτει αυτό το δικαίωμα, να δώσει σε άλλους χρήστες ή να ανακαλέσει το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details)
- **can_transfer:** επιτρέπει στο χρήστη να μεταφέρει assets από το λογαριασμό του σε έναν άλλο
- **can_receive:** επιτρέπει στο χρήστη να λαμβάνει assets από άλλους λογαριασμούς
- **can_get_my_acc_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query)

Οι εργοδότες θα πρέπει να έχουν τα ακόλουθα permissions, προκειμένου να μπορούν να βλέπουν τα πιστοποιητικά και να ελέγχουν την εγκυρότητά τους, καθώς και να διαχειρίζονται το λογαριασμό τους:

- **can_get_all_txs:** επιτρέπει την εύρεση μιας συναλλαγής με βάση το hash της
- **can_get_my_acc_detail:** επιτρέπει την πρόσβαση στις λεπτομέρειες (details) του λογαριασμού (account) του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_my_account:** επιτρέπει την πρόσβαση σε όλες τις πληροφορίες του λογαριασμού του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast:** επιτρέπει την εύρεση των assets που διαθέτει ο χρήστης που δημιουργεί το ερώτημα (query)
- **can_get_my_acc_ast_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query) και που σχετίζονται με ένα συγκεκριμένο asset
- **can_grant_can_set_my_account_detail:** επιτρέπει στο χρήστη που διαθέτει αυτό το δικαίωμα, να δώσει σε άλλους χρήστες ή να ανακαλέσει το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details)
- **can_transfer:** επιτρέπει στο χρήστη να μεταφέρει assets από το λογαριασμό του σε έναν άλλο
- **can_receive:** επιτρέπει στο χρήστη να λαμβάνει assets από άλλους λογαριασμούς
- **can_get_my_acc_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query)

Οι διαχειριστές του συστήματος θα πρέπει να έχουν τα ακόλουθα permissions, προκειμένου να μπορούν να διασφαλίζουν την εύρυθμη λειτουργία του συστήματος:

- **can_create_account:** επιτρέπει τη δημιουργία ενός νέου λογαριασμού

- **can_create_domain:** επιτρέπει τη δημιουργία ενός νέου τομέα
- **can_create_asset:** επιτρέπει τη δημιουργία ενός νέου asset
- **can_get_all_txs:** επιτρέπει την εύρεση μιας συναλλαγής με βάση το hash της
- **can_get_all_acc_detail:** επιτρέπει την πρόσβαση στις λεπτομέρειες (details) ενός λογαριασμού (account)
- **can_get_my_account:** επιτρέπει την πρόσβαση σε όλες τις πληροφορίες του λογαριασμού του χρήστη που δημιουργεί το ερώτημα (query)
- **can_get_roles:** επιτρέπει την εύρεση όλων των ρόλων που υπάρχουν στο σύστημα καθώς και την εύρεση των δικαιωμάτων κάθε ρόλου
- **can_read_assets:** επιτρέπει την πρόσβαση στις πληροφορίες που αφορούν ένα asset
- **can_grant_can_set_my_account_detail:** επιτρέπει στο χρήστη που διαθέτει αυτό το δικαίωμα, να δώσει σε άλλους χρήστες ή να ανακαλέσει το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details)
- **can_transfer:** επιτρέπει στο χρήστη να μεταφέρει assets από το λογαριασμό του σε έναν άλλο
- **can_receive:** επιτρέπει στο χρήστη να λαμβάνει assets από άλλους λογαριασμούς
- **can_add_asset_qty:** επιτρέπει στο χρήστη που δημιούργησε τη συναλλαγή να αυξήσει την ποσότητα ενός συγκεκριμένου asset στο λογαριασμό του
- **can_set_detail:** επιτρέπει στο χρήστη που έχει αυτό το δικαίωμα να θέσει τις λεπτομέρειες του λογαριασμού ενός άλλου χρήστη
- **can_add_peer:** επιτρέπει την προσθήκη peer στο δίκτυο
- **can_append_role:** επιτρέπει την προσθήκη ρόλων (roles) σε άλλο λογαριασμό
- **can_create_role:** επιτρέπει τη δημιουργία ενός νέου ρόλου
- **can_detach_role:** επιτρέπει την ανάκληση ενός ρόλου από ένα λογαριασμό
- **can_get_my_acc_txs:** επιτρέπει την εύρεση των συναλλαγών που δημιούργησε ο χρήστης που δημιουργεί το ερώτημα (query)
- **can_get_all_accounts:** επιτρέπει την πρόσβαση σε όλες τις πληροφορίες όλων των λογαριασμών

6.3.2.4 Ρόλοι (Roles)

Επειδή τα permissions δεν μπορούν να δοθούν απευθείας σε ένα λογαριασμό, θα δημιουργηθούν ρόλοι για κάθε ομάδα χρηστών, οι οποίοι θα περιέχουν ένα σύνολο δικαιωμάτων.

Θα δημιουργηθεί, επίσης, ένας ρόλος με το όνομα “certificate”, ο οποίος θα δοθεί στην κατηγορία που αφορά τα πιστοποιητικά.

Το σύστημα θα ξεκινήσει τη λειτουργία του με τους εξής ρόλους:

- university: ο ρόλος αυτός θα περιέχει τα permissions που πρέπει να έχουν τα εκπαιδευτικά ιδρύματα, όπως συζητήθηκε στην ενότητα “3.2.3) Δικαιώματα (Permissions)” του παρόντος κεφαλαίου.
- student: ο ρόλος αυτός θα περιέχει τα permissions που πρέπει να έχουν οι φοιτητές, όπως συζητήθηκε στην ενότητα “3.2.3) Δικαιώματα (Permissions)” του παρόντος κεφαλαίου.
- employer: ο ρόλος αυτός θα περιέχει τα permissions που πρέπει να έχουν οι εργοδότες, όπως συζητήθηκε στην ενότητα “3.2.3) Δικαιώματα (Permissions)” του παρόντος κεφαλαίου.

- admin: ο ρόλος αυτός θα περιέχει τα permissions που πρέπει να έχουν οι διαχειριστές του συστήματος, όπως συζητήθηκε στην ενότητα "3.2.3) Δικαιώματα (Permissions)" του παρόντος κεφαλαίου.
- certificate: ο ρόλος αυτός μπορεί είτε να είναι κενός, δηλαδή να μην περιέχει κανένα permission, είτε να περιέχει κάποια από τα πιο συνηθισμένα permissions, όπως: can_get_all_txs, can_get_my_acc_detail, can_get_my_account, can_transfer και can_receive, τα οποία έχουν ήδη περιγραφεί.

6.3.2.5 Τομείς (Domains)

Όπως έχει ήδη αναφερθεί, οι τομείς στο Hyperledger Iroha στοχεύουν στην ομαδοποίηση των accounts και των assets.

Στην παρούσα εφαρμογή, θα χρησιμοποιηθούν τομείς για την ομαδοποίηση των χρηστών σε εκπαιδευτικά ιδρύματα, σπουδαστές, εργοδότες και διαχειριστές, καθώς και την ομαδοποίηση των assets σε πιστοποιητικά.

Συνεπώς, το σύστημα θα ξεκινήσει τη λειτουργία του με τους εξής τομείς:

- uni: θα έχει σαν default ρόλο το "university" και θα αναπαριστά τα εκπαιδευτικά ιδρύματα.
- stud: θα έχει σαν default ρόλο το "student" και θα αναπαριστά τους σπουδαστές.
- empl: θα έχει σαν default ρόλο το "employer" και θα αναπαριστά τους εργοδότες.
- bc: θα έχει σαν default ρόλο τον "admin" και θα αναπαριστά τους διαχειριστές του συστήματος. Οι διαχειριστές πρέπει να έχουν τα δικαιώματα που έχουν όλοι οι χρήστες, ώστε να μπορούν να δημιουργούν accounts σε όλους τους τομείς. Επομένως, σε κάθε διαχειριστή θα προστίθενται οι υπόλοιποι ρόλοι που υπάρχουν στο σύστημα ("university", "student", "employer", "certificate"). Η ονομασία του τομέα "bc" προέρχεται από τον όρο "blockchain".
- cert: θα έχει σαν ρόλο το "certificate", ο οποίος στην προκειμένη περίπτωση θα είναι κενός, και θα αναπαριστά τα πιστοποιητικά.

6.3.3 Ανάπτυξη Εφαρμογής

Στην παρούσα ενότητα θα παρουσιαστεί ο κώδικας και οι λεπτομέρειες της εφαρμογής.

6.3.3.1 Genesis Block

Αρχικά, πρέπει να δημιουργηθεί το πρώτο block του blockchain (genesis block), το οποίο θα υλοποιεί όλα όσα περιγράφηκαν στην προηγούμενη ενότητα.

Δηλαδή το genesis block θα εκτελεί τα εξής:

- θα δημιουργεί τους ρόλους "university", "student", "employer", "certificate" και "admin".

- θα δημιουργεί τους τομείς “uni”, “stud”, “empl”, “bc” και “cert” με default ρόλους όπως περιγράφηκαν στην προηγούμενη ενότητα.
- θα δημιουργεί τα assets “bachelor”, “master” και “phd”.
- θα δημιουργεί, επίσης, ένα account για κάθε τομέα με τα εξής account_id:
  - admin@bc, στον οποίο θα προστεθούν όλοι οι ρόλοι
  - student1@stud
  - asep@empl
  - ntua@uni
- θα δημιουργεί, επίσης, peers για το δίκτυο

Στη συνέχεια, παρουσιάζεται ο κώδικας του genesis block, ο οποίος πρέπει να αντικαταστήσει τον κώδικα που υπάρχει στο αρχείο “genesis.block” του φακέλου “conf”.

```
{
  "block_v1":{
    "payload":{
      "transactions":[
        {
          "payload":{
            "reducedPayload":{
              "commands":[
                {
                  "addPeer":{
                    "peer":{
                      "address":"127.0.0.1:10001",

"peerKey":"bddd58404d1315e0eb27902c5d7c8eb0602c16238f005773df406bc191308929"
                    }
                  }
                },
                {
                  "createRole":{
                    "roleName":"admin",
                    "permissions":[
                      "can_create_account",
                      "can_create_domain",
                      "can_create_asset",
                      "can_get_all_txs",
                      "can_get_all_acc_detail",
                      "can_get_my_account",
                      "can_get_roles",
                      "can_read_assets",
                      "can_grant_can_set_my_account_detail",
                      "can_transfer",
                      "can_receive",
                      "can_add_asset_qty",
                      "can_set_detail",
                      "can_add_peer",
                      "can_append_role",
                      "can_create_role",
                      "can_detach_role",
                      "can_get_my_acc_txs",
                      "can_get_all_accounts"
                    ]
                  }
```

```
    },
    {
      "createRole":{
        "roleName":"university",
        "permissions":[
          "can_get_all_txs",
          "can_get_all_acc_detail",
          "can_get_my_account",
          "can_get_my_acc_ast",
          "can_get_my_acc_ast_txs",
          "can_grant_can_set_my_account_detail",
          "can_transfer",
          "can_receive",
          "can_get_my_acc_txs",
          "can_add_asset_qty",
          "can_create_asset"
        ]
      }
    },
    {
      "createRole":{
        "roleName":"student",
        "permissions":[
          "can_get_all_txs",
          "can_get_my_acc_detail",
          "can_get_my_account",
          "can_get_my_acc_ast",
          "can_get_my_acc_ast_txs",
          "can_grant_can_set_my_account_detail",
          "can_transfer",
          "can_receive",
          "can_get_my_acc_txs"
        ]
      }
    },
    {
      "createRole":{
        "roleName":"employer",
        "permissions":[
          "can_get_all_txs",
          "can_get_my_acc_detail",
          "can_get_my_acc_txs",
          "can_get_my_acc_ast",
          "can_get_my_acc_ast_txs",
          "can_get_my_account",
          "can_grant_can_set_my_account_detail",
          "can_transfer",
          "can_receive"
        ]
      }
    },
    {
      "createRole":{
        "roleName":"certificate",
        "permissions":[
        ]
      }
    },
```

```
{
  "createDomain":{
    "domainId":"bc",
    "defaultRole":"admin"
  }
},
{
  "createDomain":{
    "domainId":"empl",
    "defaultRole":"employer"
  }
},
{
  "createDomain":{
    "domainId":"stud",
    "defaultRole":"student"
  }
},
{
  "createDomain":{
    "domainId":"uni",
    "defaultRole":"university"
  }
},
{
  "createDomain":{
    "domainId":"cert",
    "defaultRole":"certificate"
  }
},
{
  "createAsset":{
    "assetName":"bachelor",
    "domainId":"cert",
    "precision":0
  }
},
{
  "createAsset":{
    "assetName":"master",
    "domainId":"cert",
    "precision":0
  }
},
{
  "createAsset":{
    "assetName":"phd",
    "domainId":"cert",
    "precision":0
  }
},
{
  "createAccount":{
    "accountName":"admin",
    "domainId":"bc",

"publicKey":"313a07e6384776ed95447710d15e59148473ccfc052a681317a72a69f2a49910"
  }
```

```
                },
                {
                  "createAccount":{
                    "accountName":"asep",
                    "domainId":"empl",

"publicKey":"90b86fb8a2889e6780b0a6d31008e2317298b591b4450d4dca72a8aa8f37519b"
                  }
                },
                {
                  "createAccount":{
                    "accountName":"ntua",
                    "domainId":"uni",

"publicKey":"8135e2242155e26c3941f94bdae34bf647c00aaafa333e41dacc291af090a8a5"
                  }
                },
                {
                  "createAccount":{
                    "accountName":"student1",
                    "domainId":"stud",

"publicKey":"bfec40d9d951ce59129276ed207ca810d5c6cf5e1ce83cd20959e9bd0d44df73"
                  }
                },
                {
                  "appendRole":{
                    "accountId":"admin@bc",
                    "roleName":"employer"
                  }
                },
                {
                  "appendRole":{
                    "accountId":"admin@bc",
                    "roleName":"student"
                  }
                },
                {
                  "appendRole":{
                    "accountId":"admin@bc",
                    "roleName":"university"
                  }
                },
                {
                  "appendRole":{
                    "accountId":"admin@bc",
                    "roleName":"certificate"
                  }
                }
              ],
              "quorum":1
            }
          }
        }
      ],
      "txNumber":1,
      "height":"1",
```

```
"prevBlockHash":"0000000000000000000000000000000000000000000000000000000000000000"
    }
  }
}
```


Εννοείται ότι τα κλειδιά που δίνονται στο genesis block μπορούν και πρέπει να αλλάξουν, αλλά πρέπει πάντα να είναι σύμφωνα με όσα περιγράφονται στην επόμενη ενότητα.

6.3.3.2 Αποθήκευση Ζεύγους Κλειδιών

Προκειμένου να είναι εφικτός ο τοπικός έλεγχος της εφαρμογής θα δημιουργηθούν οι φάκελοι “public_keys” και “private_keys” μέσα στο φάκελο “conf”, στους οποίους θα αποθηκεύονται αντίστοιχα τα δημόσια και τα ιδιωτικά κλειδιά των χρηστών.

Οι ονομασίες των αρχείων που θα περιέχουν τα κλειδιά αποτελούνται από το account_id του χρήστη και το “.priv” ή “.pub”. Για παράδειγμα τα αρχεία που περιέχουν τα κλειδιά του χρήστη “ntua@uni” ονομάζονται “ntua@uni.priv” και “ntua@uni.pub”.

Υπενθυμίζεται ότι κανονικά τα ιδιωτικά κλειδιά των χρηστών θα αποθηκεύονται τοπικά στο wallet τους, ενώ τα δημόσια κλειδιά θα είναι αποθηκευμένα σε server.

Σε αυτό το σημείο, χρειάζεται να αποθηκευτούν τα κλειδιά των χρηστών που έχουν ήδη δημιουργηθεί μέσω του genesis block.

Στον ακόλουθο πίνακα φαίνονται τα αρχεία που πρέπει να δημιουργηθούν.

Όνομα αρχείου
ntua@uni.priv
ntua@uni.pub
asep@empl.priv
asep@empl.pub
student1@stud.priv
student1@stud.pub
admin@bc.priv
admin@bc.pub

Πίνακας 2: Αρχεία κλειδιών που πρέπει να δημιουργηθούν για τους χρήστες που δημιουργήθηκαν στο Genesis Block

Τα ιδιωτικά κλειδιά παράγονται μέσω της συνάρτησης “private_key()” της βιβλιοθήκης “IrohaCrypto”, ενώ τα δημόσια κλειδιά παράγονται από τα ιδιωτικά μέσω της συνάρτησης “derive_public_key()” της ίδιας βιβλιοθήκης.

6.3.3.3 Βιβλιοθήκες (Libraries)

Ο κώδικας της εφαρμογής απαιτεί ορισμένες βιβλιοθήκες, οι οποίες μπορούν να εισαχθούν στο σύστημα ως εξής:

```
import os
import binascii
from iroha import IrohaCrypto
from iroha import Iroha, IrohaGrpc
from iroha import primitive_pb2
import sys
import datetime
import requests, hashlib
import webbrowser
from iroha.primitive_pb2 import *
```

6.3.3.4 Ορισμός Δικτύου (Network)

Για τον ορισμό του δικτύου στο port 50051 χρησιμοποιείται η εντολή:

```
net = IrohaGrpc('{}:{}'.format('localhost', '50051'))
```

6.3.3.5 Προαπαιτούμενες Συναρτήσεις

Σε αυτό το σημείο θα αναπτυχθεί ο κώδικας συναρτήσεων που χρησιμοποιούνται για την επιτέλεση των λειτουργιών της εφαρμογής.

- **send_transaction_and_print_status**

Η συγκεκριμένη συνάρτηση αφορά την αποστολή συναλλαγών στο δίκτυο και την εγγραφή τους στο blockchain. Επιστρέφει το hash της συναλλαγής και το status της, δηλαδή αν έγινε committed ή προέκυψε κάποιο stateless ή stateful σφάλμα.

```
def send_transaction_and_print_status(transaction):
    hex_hash = binascii.hexlify(IrohaCrypto.hash(transaction))
    print('Transaction hash = {}, creator = {}'.format(
        hex_hash, transaction.payload.reduced_payload.creator_account_id))
    net.send_tx(transaction)
    to_send_back = []
    to_send_back.append(hex_hash)
    for status in net.tx_status_stream(transaction):
        to_send_back.append(status)
    print(to_send_back)
    return to_send_back
```

- **account_private_key**

Αυτή η συνάρτηση χρησιμοποιείται για την εύρεση του ιδιωτικού κλειδιού ενός λογαριασμού (account).

```
def account_private_key(self):
    path = "/Users/username/Documents/iroha-master/conf/private_keys/" + self + ".priv"
    with open (path, "r") as priv_key_file:
        key = priv_key_file.readlines()
    return key[0]
```

- **user_pub_priv_key**

Η συνάρτηση αυτή δημιουργεί ένα νέο ζεύγος δημόσιου - ιδιωτικού κλειδιού χρησιμοποιώντας τις συναρτήσεις που αναφέρθηκαν στην ενότητα “3.3.2) Αποθήκευση Ζεύγους Κλειδιών” του παρόντος κεφαλαίου.

```
def user_pub_priv_key():
    user_private_key = IrohaCrypto.private_key()
    user_public_key = IrohaCrypto.derive_public_key(user_private_key)
    return user_private_key, user_public_key
```

- **certificate_hash**

Η συνάρτηση αυτή υπολογίζει το hash ενός πιστοποιητικού που είναι αποθηκευμένο σε μορφή pdf τοπικά στον υπολογιστή. Σε σχόλια δίνεται ο τρόπος υπολογισμού ενός online πιστοποιητικού. Για τον υπολογισμό του hash χρησιμοποιείται η συνάρτηση SHA-256.

```
def certificate_hash(file):
    #for url
    #pdf = requests.get(file)
    #hash = hashlib.sha256(pdf.content).hexdigest()
    #for saved file
    with open(file, 'rb') as f:
        hash = hashlib.sha256(f.read()).hexdigest()
    return hash
```

- **open_file**

Η συνάρτηση αυτή ανοίγει ένα αρχείο που είναι αποθηκευμένο τοπικά στον υπολογιστή σε μια νέα καρτέλα. Σε σχόλια φαίνεται πως μπορεί να ανοίξει μια ιστοσελίδα, της οποίας το uri δίνεται ως παράμετρος στη συνάρτηση.

```
def open_file(file):
    #gia site
    #path = file
    path = 'file://' + file
    webbrowser.open_new(path)
```

- **save_to_file**

Η συγκεκριμένη συνάρτηση δημιουργεί ένα νέο αρχείο και προσθέτει περιεχόμενο σε αυτό.

```
def save_to_file(file_path, name, content):
    final_path = file_path + '/' + name
    with open(final_path, 'x') as f:
        f.write(content)
```

- **proper_date**

Η συγκεκριμένη συνάρτηση επιστρέφει την τωρινή ημερομηνία ως ένα string που αποτελείται από το έτος, το μήνα και τη μέρα. Για παράδειγμα η ημερομηνία 15/2/2020 θα επιστραφεί ως “2020215”.

```
def proper_date():
    x = datetime.date.today()
    return str(datetime.date.today().year) + str(datetime.date.today().month) +
str(datetime.date.today().day)
```

6.3.3.6 Βασικές λειτουργίες εφαρμογής

Σε αυτό το σημείο θα αναπτυχθεί ο κώδικας που αφορά τις σημαντικότερες από τις ενέργειες που μπορούν να εκτελέσουν οι χρήστες της εφαρμογής.

- **grant_can_set_my_account_detail_tx**

Μέσω αυτής της συνάρτησης ένας χρήστης, ο οποίος έχει το δικαίωμα “can_grant_can_set_my_account_detail”, μπορεί να δώσει σε άλλους χρήστες το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details).

Η συνάρτηση αυτή χρησιμοποιείται από:

- τους σπουδαστές, προκειμένου να επιτρέψουν στα εκπαιδευτικά ιδρύματα να ολοκληρώσουν τις ενέργειες έκδοσης ενός πιστοποιητικού.
- τους εργοδότες, προκειμένου να επιτρέψουν στους σπουδαστές να τους δώσουν τις απαραίτητες πληροφορίες ώστε να ελέγξουν την εγκυρότητα ενός πιστοποιητικού.

```
def grant_can_set_my_account_detail_tx(self, account):
    iroha = Iroha(self)
    commands = [
        iroha.command('GrantPermission', account_id = account, permission =
primitive_pb2.can_set_my_account_detail)
    ]
    tx = IrohaCrypto.sign_transaction(
        iroha.transaction(commands), account_private_key(self))
    send_result = send_transaction_and_print_status(tx)
```

```
    hex_hash = send_result[0]
    status = send_result[1:]
    return hex_hash, status
```

- **revoke_can_set_my_account_detail_tx**

Μέσω αυτής της συνάρτησης ένας χρήστης, ο οποίος έχει το δικαίωμα "can_grant_can_set_my_account_detail", μπορεί να ανακαλέσει από άλλους χρήστες το δικαίωμα να θέτουν τις λεπτομέρειες του λογαριασμού του (account details).

Η συνάρτηση αυτή χρησιμοποιείται από:

- τους σπουδαστές, προκειμένου να απαγορεύσουν στα εκπαιδευτικά ιδρύματα να θέτουν τις λεπτομέρειες του λογαριασμού τους μετά την έκδοση του πιστοποιητικού.
- τους εργοδότες, προκειμένου να απαγορεύσουν στους σπουδαστές να θέτουν τις λεπτομέρειες του λογαριασμού τους μετά τη λήψη των απαραίτητων πληροφοριών από αυτούς.

```
def revoke_can_set_my_account_detail_tx(self, account):
    iroha = Iroha(self)
    commands = [
        iroha.command('RevokePermission', account_id = account, permission =
primitive_pb2.can_set_my_account_detail)
    ]
    tx = IrohaCrypto.sign_transaction(
        iroha.transaction(commands), account_private_key(self))
    send_result = send_transaction_and_print_status(tx)
    hex_hash = send_result[0]
    status = send_result[1:]
    return hex_hash, status
```

- **get_account_assets**

Η συγκεκριμένη συνάρτηση επιτρέπει στον εκάστοτε χρήστη να δει το είδος και την ποσότητα των assets που διαθέτει. Για να μπορέσει να λάβει αυτή την πληροφορία, ο χρήστης πρέπει να έχει το δικαίωμα "can_get_my_acc_ast".

Η συνάρτηση αυτή χρησιμοποιείται από τους σπουδαστές προκειμένου να δουν τα πτυχία που έχουν εκδοθεί μέσω του blockchain. Για παράδειγμα, ένας σπουδαστής μπορεί να δει ότι έχει 1 bachelor και 2 master.

```
def get_account_assets(self):
    iroha = Iroha(self)
    query = iroha.query('GetAccountAssets', creator_account = self, account_id = self)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **get_my_account_detail**

Μέσω αυτής της συνάρτησης ένας χρήστης, ο οποίος έχει το δικαίωμα "can_get_my_acc_detail" ή "can_get_all_acc_detail", μπορεί να δει όλες τις λεπτομέρειες του λογαριασμού του (account details).

Μπορεί να χρησιμοποιηθεί από σπουδαστές και εργοδότες.

```
def get_my_account_detail(self):
    iroha = Iroha(self)
    query = iroha.query('GetAccountDetail', creator_account = self,  account_id = self)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **get_my_account_detail_by_writer**

Αυτή η συνάρτηση επιτρέπει στο χρήστη να δει τις λεπτομέρειες του λογαριασμού του (account details) που έχει θέσει ένας άλλος χρήστης. Προκειμένου να μπορέσει να λάβει αυτή την πληροφορία, ο χρήστης πρέπει να έχει το δικαίωμα "can_get_my_acc_detail" ή "can_get_all_acc_detail".

Η συνάρτηση χρησιμοποιείται από:

- τους σπουδαστές, προκειμένου να δουν τις λεπτομέρειες που τους έχουν προσθέσει τα εκπαιδευτικά ιδρύματα.
- τους εργοδότες, προκειμένου να δουν τις λεπτομέρειες που τους έχουν προσθέσει οι σπουδαστές.

```
def get_my_account_detail_by_writer(self, writer):
    iroha = Iroha(self)
    query = iroha.query('GetAccountDetail', creator_account = self,  account_id = self, writer =
writer)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **get_account_detail_by_self**

Μέσω της συγκεκριμένης συνάρτησης, ένας χρήστης, ο οποίος έχει το δικαίωμα "can_get_all_acc_detail", μπορεί να δει τις λεπτομέρειες που έχει θέσει σε έναν άλλο χρήστη.

Χρησιμοποιείται από τα εκπαιδευτικά ιδρύματα για τον έλεγχο των λεπτομερειών που έχουν θέσει στους σπουδαστές κατά τη διαδικασία έκδοσης ενός πιστοποιητικού.

```
def get_account_detail_by_self(self, acc_id):
    iroha = Iroha(self)
    query = iroha.query('GetAccountDetail', creator_account = self,  account_id = acc_id, writer =
```

```
self)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **issue_certificate**

Αυτή η συνάρτηση επιτρέπει στα εκπαιδευτικά ιδρύματα να εκδίδουν πιστοποιητικά σε σπουδαστές.

Για την εκτέλεση της συγκεκριμένης συνάρτησης το εκπαιδευτικό ίδρυμα πρέπει να έχει το δικαίωμα να θέτει τις λεπτομέρειες του εκάστοτε σπουδαστή, καθώς και το δικαίωμα "can_transfer", ενώ ο σπουδαστής πρέπει να έχει το δικαίωμα "can_receive".

Έστω ότι το εκπαιδευτικό ίδρυμα "ntua@uni" θέλει να εκδώσει ένα master στον σπουδαστή "student@stud". Σε αυτή την περίπτωση θα κληθεί η συνάρτηση "issue_certificate" η οποία πραγματοποιεί τις ακόλουθες ενέργειες:

- αυξάνει την ποσότητα του asset: master#cert που διαθέτει το εκπαιδευτικό ίδρυμα κατά 1.
- μεταφέρει ποσότητα 1 του συγκεκριμένου asset στο σπουδαστή.
- προσθέτει το uri του πιστοποιητικού στις λεπτομέρειες του λογαριασμού του σπουδαστή. Το όνομα της λεπτομέρειας προκύπτει από την ένωση του ονόματος του asset, του string "_uri_" και της ημερομηνίας έκδοσης. Για παράδειγμα, αν το συγκεκριμένο πτυχίο εκδόθηκε στις 15/2/2020, τότε το όνομα της λεπτομέρειας θα είναι "master_uri_2020215".
- προσθέτει το hash του πιστοποιητικού στις λεπτομέρειες του λογαριασμού του σπουδαστή. Το όνομα της λεπτομέρειας προκύπτει από την ένωση του ονόματος του asset, του string "_hash_" και της ημερομηνίας έκδοσης. Για παράδειγμα, αν το συγκεκριμένο πτυχίο εκδόθηκε στις 15/2/2020, τότε το όνομα της λεπτομέρειας θα είναι "master_hash_2020215".
- προσθέτει το hash της συναλλαγής έκδοσης του πιστοποιητικού, η οποία περιλαμβάνει τις παραπάνω ενέργειες, στις λεπτομέρειες του λογαριασμού του σπουδαστή. Το όνομα της λεπτομέρειας προκύπτει από την ένωση του ονόματος του asset, του string "_transaction_hash_" και της ημερομηνίας έκδοσης. Για παράδειγμα, αν το συγκεκριμένο πτυχίο εκδόθηκε στις 15/2/2020, τότε το όνομα της λεπτομέρειας θα είναι "master_transaction_hash_2020215".

```
def issue_certificate(self, student, asset, uri, domain_name=None):
    iroha = Iroha(self)
    domain = "#"
    domain = domain + (domain_name or "cert")
    cert_hash = certificate_hash(uri)
    commands = [
        iroha.command('AddAssetQuantity',
                      asset_id = asset + domain, amount = "1"),
        iroha.command('TransferAsset',
                      src_account_id = self,
                      dest_account_id = student,
```

```
                asset_id = asset + domain,
                amount = "1",
                description = asset + "issuance"),
        iroha.command('SetAccountDetail',
                account_id = student,
                key = asset + "_uri_" + proper_date(),
                value = uri),
        iroha.command('SetAccountDetail',
                account_id = student,
                key = asset + "_hash_" + proper_date(),
                value = cert_hash)
    ]
    tx = IrohaCrypto.sign_transaction(
        iroha.transaction(commands), account_private_key(self))
    result = send_transaction_and_print_status(tx)
    hex_hash_tx1 = result[0]
    status_tx1 = result[1:]
    if (status_tx1 == [('ENOUGH_SIGNATURES_COLLECTED', 9, 0),
('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]):
        commands = [
            iroha.command('SetAccountDetail',
                    account_id = student,
                    key = asset + "_transaction_hash_" + proper_date(),
                    value = result[0])
        ]
        tx = IrohaCrypto.sign_transaction(
            iroha.transaction(commands), account_private_key(self))
        result = send_transaction_and_print_status(tx)
        hex_hash_tx2 = result[0]
        status_tx2 = result[1:]
        return hex_hash_tx1, status_tx1, hex_hash_tx2, status_tx2
```

- **get_transaction_by_hash**

Η συνάρτηση αυτή επιτρέπει στους χρήστες, οι οποίοι έχουν το δικαίωμα “can_get_all_txs”, να αποκτήσουν πρόσβαση σε μία συναλλαγή εφόσον γνωρίζουν το hash της.

Η συνάρτηση χρησιμοποιείται από τα εκπαιδευτικά ιδρύματα, τους σπουδαστές και τους εργοδότες προκειμένου να δουν τη συναλλαγή έκδοσης ενός πιστοποιητικού, ώστε να ελέγξουν το hash του.

```
def get_transaction_by_hash(self, hash):
    iroha = Iroha(self)
    hashes = hash.encode('utf-8')
    query = iroha.query('GetTransactions', creator_account = self, tx_hashes=[hashes])
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **get_uri_and_hash_of_certificate_by_transaction_hash**

Η συνάρτηση αυτή επιτρέπει στους χρήστες, οι οποίοι έχουν το δικαίωμα "can_get_all_txs", να αποκτήσουν πρόσβαση στο uri και το hash ενός πιστοποιητικού, καθώς και το εκπαιδευτικό ίδρυμα που το εξέδωσε, εφόσον γνωρίζουν το hash της συναλλαγής έκδοσής του. Ταυτόχρονα, ανοίγει το πιστοποιητικό σε νέα καρτέλα.

Η συνάρτηση χρησιμοποιείται από τα εκπαιδευτικά ιδρύματα, τους σπουδαστές και τους εργοδότες.

```
def get_uri_and_hash_of_certificate_by_transaction_hash(self, hash):
    iroha = Iroha(self)
    hashes = hash.encode('utf-8')
    query = iroha.query('GetTransactions', creator_account = self, tx_hashes=[hashes])
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)

    #find uri and hash of issued certificate
    response_string = str(response)
    res = response_string.split("commands")
    res_uri = res[3].splitlines()[4].split("            ")[1].split("\"")[1]
    res_hash = res[4].splitlines()[4].split("            ")[1].split("\"")[1]
    res_creator = res[4].splitlines()[7].split("        ")[1].split("\"")[1]
    print("Certificate uri: " + res_uri)
    print("Certificate hash: " + res_hash)
    print("Certificate creator: " + res_creator)
    #open certificate
    open_file(res_uri)
    return  res_uri, res_hash, res_creator
```

- **get_account_transactions_by_asset**

Η συγκεκριμένη συνάρτηση επιτρέπει στο χρήστη, ο οποίος έχει το δικαίωμα "can_get_my_acc_ast_txs", να δει όλες τις συναλλαγές του που σχετίζονται με ένα συγκεκριμένο asset.

Χρησιμοποιείται από:

- τα εκπαιδευτικά ιδρύματα προκειμένου να δουν όλους τους φοιτητές στους οποίους έχουν εκδώσει ένα συγκεκριμένο πιστοποιητικό, όπως για παράδειγμα το bachelor.
- τους σπουδαστές για την πρόσβαση στις συναλλαγές που αφορούν ένα συγκεκριμένο είδος πιστοποιητικών.

```
def get_account_transactions_by_asset(self, asset, domain_name=None):
    iroha = Iroha(self)
    domain = "#"
    domain = domain + (domain_name or "cert")
    query = iroha.query('GetAccountAssetTransactions', creator_account = self, account_id = self,
                        asset_id = asset + domain, page_size=20)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    print(response)
    return response
```

- **get_certificate_hash_and_set_detail**

Αυτή η συνάρτηση επιτρέπει στους σπουδαστές να δώσουν στους εργοδότες την απαραίτητη πληροφορία για τον έλεγχο της εγκυρότητας ενός πιστοποιητικού. Επίσης, επιτρέπει στους σπουδαστές να αποκτήσουν πρόσβαση στο uri και το hash ενός πιστοποιητικού, καθώς και το εκπαιδευτικό ίδρυμα που το εξέδωσε, εφόσον γνωρίζουν το hash της συναλλαγής έκδοσής του, ενώ ταυτόχρονα, ανοίγει το πιστοποιητικό σε νέα καρτέλα.

Για την εκτέλεση της συγκεκριμένη συνάρτησης ο σπουδαστής πρέπει να έχει το δικαίωμα "can_get_all_txs", καθώς και το δικαίωμα να θέτει τις λεπτομέρειες του εκάστοτε εργοδότη.

Έστω ότι ο σπουδαστής "student@stud" θέλει να γνωστοποιήσει στον εργοδότη "employer@empl" το hash της συναλλαγής έκδοσης ενός εκ των πιστοποιητικών του. Σε αυτή την περίπτωση θα κληθεί η συνάρτηση "get_certificate_hash_and_set_detail", η οποία πραγματοποιεί τις ακόλουθες ενέργειες:

- βρίσκει τη συναλλαγή έκδοσης του πιστοποιητικού με βάση το hash.
- βρίσκει από τη συναλλαγή το uri και το hash του πιστοποιητικου, καθώς και το εκπαιδευτικό ίδρυμα που το εξέδωσε, ώστε να τα επιστρέψει στον σπουδαστή.
- ανοίγει το πιστοποιητικό σε νέα καρτέλα.
- προσθέτει το hash της συναλλαγής έκδοσης του πιστοποιητικού στις λεπτομέρειες του λογαριασμού του εργοδότη.

```
def get_certificate_hash_and_set_detail(self, hash, acc_id, key):
    iroha = Iroha(self)
    hashes = hash.encode('utf-8')
    query = iroha.query('GetTransactions', creator_account = self, tx_hashes=[hashes])
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)

    #find uri and hash of issued certificate
    response_string = str(response)
    res = response_string.split("commands")
    res_uri = res[3].splitlines()[4].split("            ")[1].split("\"")[1]
    res_hash = res[4].splitlines()[4].split("          ")[1].split("\"")[1]
    res_creator = res[4].splitlines()[7].split("       ")[1].split("\"")[1]
    print("Certificate uri: " + res_uri)
    print("Certificate hash: " + res_hash)
    print("Certificate creator: " + res_creator)
    #open certificate
    open_file(res_uri)
    commands = [
        iroha.command('SetAccountDetail', account_id = acc_id, key = key, value = hash)
    ]
    tx = IrohaCrypto.sign_transaction(
       iroha.transaction(commands), account_private_key(self))
    send_result = send_transaction_and_print_status(tx)
    hex_hash = send_result[0]
    status = send_result[1:]
    return res_uri, res_hash, res_creator, hex_hash, status
```

- **create_account**

Η συνάρτηση αυτή επιτρέπει στους χρήστες, οι οποίοι έχουν το δικαίωμα “can_create_account”, να δημιουργήσουν νέους λογαριασμούς.

Μπορεί να χρησιμοποιηθεί από το διαχειριστή του συστήματος για τη δημιουργία λογαριασμών που αφορούν εκπαιδευτικά ιδρύματα, σπουδαστές, εργοδότες αλλά και νέους διαχειριστές.

Έστω ότι ο διαχειριστής “admin@bc” θέλει να δημιουργήσει ένα λογαριασμό για τον σπουδαστή newstudent. Σε αυτή την περίπτωση θα κληθεί η συνάρτηση “create_account”, η οποία πραγματοποιεί τις ακόλουθες ενέργειες:

- δημιουργεί ένα καινούριο ζεύγος δημόσιου - ιδιωτικού κλειδιού για το νέο χρήστη.
- δημιουργεί τον καινούριο λογαριασμό
- αποθηκεύει τα κλειδιά σε νέα αρχεία σύμφωνα με όσα περιγράφονται στην ενότητα “3.3.2) Αποθήκευση Ζεύγους Κλειδιών” του παρόντος κεφαλαίου.

```
def create_account(self, name, domain):
    iroha = Iroha(self)
    account_id = name + '@' + domain
    private_key, public_key = user_pub_priv_key()
    commands = [
        iroha.command('CreateAccount', account_name = name, domain_id = domain, public_key =
public_key),
    ]
    tx = IrohaCrypto.sign_transaction(
        iroha.transaction(commands), account_private_key(self))
    send_result = send_transaction_and_print_status(tx)
    hex_hash = send_result[0]
    status = send_result[1:]
    query = iroha.query('GetAccount', creator_account = self, account_id = name + "@" + domain)
    IrohaCrypto.sign_query(query, account_private_key(self))
    response = net.send_query(query)
    if (str(response).split("{")[0] == "account_response "):
        save_to_file("/Users/silvia/Documents/iroha-master/conf/private_keys/", account_id + ".priv",
private_key.decode('utf-8'))
        save_to_file("/Users/silvia/Documents/iroha-master/conf/public_keys", account_id + ".pub",
public_key.decode('utf-8'))
    return hex_hash, status
```

6.3.3.7 Δευτερεύουσες λειτουργίες εφαρμογής

Οι παραπάνω λειτουργίες ναι μεν υλοποιούν τη βασική λειτουργικότητα της εφαρμογής, αλλά δεν την καθιστούν ολοκληρωμένη. Για το λόγο αυτό, έχουν αναπτυχθεί επιπλέον λειτουργίες. Στη συνέχεια, αναφέρονται οι επιπλέον εντολές που μπορεί να εκτελέσει κάθε ομάδα χρηστών.

1) Διαχειριστής

Ο διαχειριστής τους συστήματος μπορεί επίσης να:

- δημιουργήσει ένα νέο τομέα
- δημιουργήσει ένα νέο asset
- δει τις πληροφορίες του λογαριασμού του

- δει τις συναλλαγές που έχει πραγματοποιήσει
- δει τις πληροφορίες που αφορούν ένα asset
- προσθέσει ποσότητα από ένα συγκεκριμένο asset στο λογαριασμό του
- μεταφέρει asset σε άλλο λογαριασμό
- δει όλους τους ρόλους που υπάρχουν στο σύστημα
- δει τα δικαιώματα (permissions) που έχει ένας συγκεκριμένος ρόλος
- δημιουργήσει ένα νέο ρόλο
- προσθέσει έναν ρόλο σε έναν άλλο χρήστη
- αφαιρέσει ένα ρόλο από έναν άλλο χρήστη

2) Εκπαιδευτικό ίδρυμα

Ένα εκπαιδευτικό ίδρυμα μπορεί επίσης να:

- δει τις πληροφορίες του λογαριασμού του
- δει τις συναλλαγές που έχει πραγματοποιήσει
- δημιουργήσει ένα νέο asset
- προσθέσει ποσότητα από ένα συγκεκριμένο asset στο λογαριασμό του
- μεταφέρει asset σε άλλο λογαριασμό
- θέσει τις λεπτομέρειες του λογαριασμού ενός σπουδαστή που του έχει δώσει το αντίστοιχο δικαίωμα

3) Σπουδαστής

Ένας σπουδαστής μπορεί επίσης να:

- δει τις πληροφορίες του λογαριασμού του
- δει τις συναλλαγές που έχει πραγματοποιήσει
- υπολογίσει το hash ενός πιστοποιητικού
- θέσει τις λεπτομέρειες του λογαριασμού του
- θέσει τις λεπτομέρειες ενός άλλου λογαριασμού

4) Εργοδότης

Ένας εργοδότης μπορεί επίσης να:

- δει τις πληροφορίες του λογαριασμού του
- δει τις συναλλαγές που έχει πραγματοποιήσει
- υπολογίσει το hash ενός πιστοποιητικού
- θέσει τις λεπτομέρειες του λογαριασμού του

6.3.3.8 Command Line Interface

Για τη χρήση της εν λόγω εφαρμογής από τις διάφορες ομάδες χρηστών του συστήματος έχει δημιουργηθεί ένα command line interface. Το interface αυτό διευκολύνει τους χρήστες να εκτελέσουν τις ενέργειες που επιθυμούν.

Μέσα από το command line interface, οι χρήστες επιλέγουν τις λειτουργίες που επιθυμούν να πραγματοποιήσουν και δίνουν στο σύστημα τις απαραίτητες για αυτές τις λειτουργίες παραμέτρους, χωρίς να πρέπει να γνωρίζουν τις συναρτήσεις που χρησιμοποιούνται στο παρασκήνιο.

Σε κάθε χρήστη εμφανίζεται ένα διαφορετικό σύνολο λειτουργιών για επιλογή, ανάλογα με τον τομέα στον οποίο ανήκει. Οι λειτουργίες που μπορεί να επιλέξει κάθε κατηγορία χρηστών περιγράφονται στις ενότητες "3.3.6) Βασικές λειτουργίες εφαρμογής" και "3.3.7) Δευτερεύουσες λειτουργίες εφαρμογής" του παρόντος κεφαλαίου.

Στη συνέχεια παρουσιάζεται ο τρόπος με τον οποίο οι χρήστες μπορούν να χρησιμοποιήσουν το command line interface.

- Αρχικά ζητείται από τον εκάστοτε χρήστη να εισάγει το account id του και γίνεται έλεγχος για την επαλήθευση της ύπαρξης του συγκεκριμένου account id. Σε περίπτωση που ο χρήστης έχει προσπαθήσει να συνδεθεί με λάθος id, εμφανίζεται μήνυμα ότι το id ήταν λάθος και το σύστημα περιμένει για την εισαγωγή νέου account id.

```
Account Id: ntu@uni
Wrong Account Id. Please enter existing id: ntua@uni
```

Εικόνα 22: Εισαγωγή account id στην εφαρμογή

- Όταν ο χρήστης εισάγει υπάρχον account id, ελέγχεται ο τομέας (domain) στον οποίο ανήκει ο χρήστης. Ανάλογα με τον τομέα του, εμφανίζεται στο χρήστη ένα σύνολο λειτουργιών. Για τα εκπαιδευτικά ιδρύματα εμφανίζονται οι ακόλουθες λειτουργίες.

```
1. Get your account information
2. Get your transactions
3. Get your transactions related to a certain asset
4. Create asset
5. Add asset quantity to your account
6. Transfer asset
7. Issue certificate
8. Set student's account detail with transaction hash
9. Get transaction by hash
10. Get uri and hash of certificate by transaction hash
11. Get student's account details set by you
12. Set student's account details
```

Εικόνα 23: Λειτουργίες εκπαιδευτικών ιδρυμάτων

- Στη συνέχεια, ζητείται από το χρήστη να εισάγει τον αριθμό της εντολής που θέλει να εκτελέσει. Σε περίπτωση που χρήστης εισάγει λάθος αριθμό, εμφανίζεται μήνυμα ότι ο αριθμός ήταν λάθος και το σύστημα περιμένει για την εισαγωγή νέου αριθμού.

```
Enter number of command and press enter: 15
Wrong number. Enter right number of command and press enter: 4
```

Εικόνα 24: Επιλογή εντολής

- Αν η εντολή, στην οποία αντιστοιχεί ο αριθμός που πληκτρολόγησε ο χρήστης, απαιτεί την εισαγωγή παραμέτρων, τότε ζητείται από τον χρήστη να εισάγει τις απαραίτητες παραμέτρους. Διαφορετικά, εμφανίζεται απευθείας το αποτέλεσμα της εντολής.

```
Name of asset: test_create_asset
Domain of asset: cert
Precision of asset: 0
Transaction hash = b'626415093384019c7adf3d1beda6372c5dbb05fe13de6784ecaca2913e61e473', creator = ntua@uni
[b'626415093384019c7adf3d1beda6372c5dbb05fe13de6784ecaca2913e61e473', ('ENOUGH_SIGNATURES_COLLECTED', 9, 0), ('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]
```

Εικόνα 25: Εισαγωγή παραμέτρων

- Στη συνέχεια ζητείται από το χρήστη να δηλώσει αν θέλει να πραγματοποιήσει κάποια επιπλέον ενέργεια ή όχι. Αν ο χρήστης απαντήσει θετικά, εμφανίζονται και πάλι οι εντολές που μπορεί να εκτελέσει. Αν ο χρήστης απαντήσει αρνητικά, τότε κλείνει η εφαρμογή.

```
Do you want to execute another transaction? Enter Y or N:
```

Εικόνα 26: Παραμονή ή έξοδος από την εφαρμογή

6.4 Προσομοίωση λειτουργίας εφαρμογής

Στην παρούσα ενότητα θα γίνει η προσομοίωση των βασικότερων λειτουργιών της εφαρμογής.

Πιο συγκεκριμένα, θα παρουσιαστούν τα εξής:

- οι ενέργειες που πρέπει να εκτελεστούν για την έκδοση ενός πιστοποιητικού από ένα εκπαιδευτικό ίδρυμα σε ένα σπουδαστή και οι ενέργειες που πρέπει να εκτελέσει ένας σπουδαστής για να δει το πιστοποιητικό του.
- οι ενέργειες που πρέπει να εκτελεστούν για να δώσει ένας σπουδαστής σε έναν εργοδότη την απαραίτητη πληροφορία για τον έλεγχο της γνησιότητας ενός εκ των πιστοποιητικών του και οι ενέργειες που πρέπει να εκτελέσει ένας εργοδότης προκειμένου να ελέγξει τη γνησιότητα ενός πιστοποιητικού.

6.4.1 Έκδοση πιστοποιητικού

Έστω ότι το εκπαιδευτικό ίδρυμα "ntua@uni" θέλει να εκδώσει ένα πιστοποιητικό κατηγορίας bachelor στον σπουδαστή "student1@stud". Για την έκδοση του πιστοποιητικού πρέπει να πραγματοποιηθούν τα εξής:

- Ο "student1@stud" πρέπει να παραχωρήσει στο πανεπιστήμιο το δικαίωμα να μπορεί να θέτει τις λεπτομέρειες του λογαριασμού του. Αυτή η ενέργεια θα πραγματοποιείται κατά τη διαδικασία αίτησης έκδοσης του πιστοποιητικού.

Για να εκτελέσει τη συγκεκριμένη ενέργεια, ο σπουδαστής κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 14, η οποία αντιστοιχεί στην παραχώρηση του συγκεκριμένου δικαιώματος, και εισάγει το εκπαιδευτικό ίδρυμα στο οποίο επιθυμεί να δώσει αυτό το δικαίωμα. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
Account Id: student1@stud
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your assets
5. Get your transactions
6. Get your transactions related to a certain asset
7. Get hash of certificate
8. Get transaction by hash
9. Get uri and hash of certificate by transaction hash
10. Set hash of certificate issuance transaction to employer
11. Set your account details
12. Set another account's details
13. Transfer asset
14. Grant another account permission to set your details
15. Revoke from another account the permission to set your details
Enter number of command and press enter: 14
Account: ntua@uni
Transaction hash = b'631bd8807d8bf895d8a36e8b486287ac5bb11d226eaad4eda6e18aebb576722a', creator = student1@stud
[b'631bd8807d8bf895d8a36e8b486287ac5bb11d226eaad4eda6e18aebb576722a', ('ENOUGH_SIGNATURES_COLLECTED', 9, 0), ('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]
Do you want to execute another transaction? Enter Y or N:
```

Εικόνα 27: Ο σπουδαστής παραχωρεί σε πανεπιστήμιο το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του

- Ο χρήστης "ntua@uni" πρέπει να εκδώσει το πιστοποιητικό και να ελέγξει ότι η διαδικασία ολοκληρώθηκε με επιτυχία.

Για την έκδοση του πιστοποιητικού, κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 7, η οποία αντιστοιχεί στην έκδοση του πιστοποιητικού, και εισάγει το φοιτητή που αφορά το πιστοποιητικό, την κατηγορία του πιστοποιητικού καθώς και το uri του. Η παραπάνω διαδικασία φαίνεται στην ακόλουθη εικόνα.

Εικόνα 28: Έκδοση πιστοποιητικού

Για να ελέγξει ότι η παραπάνω διαδικασία ολοκληρώθηκε με πλήρη επιτυχία, το εκπαιδευτικό ίδρυμα μπορεί να δει τις λεπτομέρειες που έθεσε στον σπουδαστή. Προκειμένου να προχωρήσει σε αυτή την ενέργεια πρέπει να δηλώσει ότι θέλει να εκτελέσει νέα συναλλαγή, να επιλέξει την εντολή 11, η οποία αντιστοιχεί στον έλεγχο των λεπτομερειών του σπουδαστή, και να εισάγει το account id του φοιτητή. Στην ακόλουθη εικόνα παρουσιάζεται η παραπάνω διαδικασία.

Εικόνα 29: Έλεγχος λεπτομερειών σπουδαστή από πανεπιστήμιο

- Ο “student1@stud” πρέπει, στη συνέχεια, να ελέγξει ότι έχει πρόσβαση στο πιστοποιητικό του και να ανακαλέσει από το εκπαιδευτικό ίδρυμα το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του.

Αρχικά, πρέπει να δει τις λεπτομέρειες του λογαριασμού του που έχει θέσει το εκπαιδευτικό ίδρυμα. Για να εκτελέσει αυτή την ενέργεια, ο σπουδαστής κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 3, μέσω της οποίας μπορεί να δει τις λεπτομέρειες που του έθεσε ένας άλλος χρήστης, και εισάγει το εκπαιδευτικό ίδρυμα που εξέδωσε το πιστοποιητικό. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

Εικόνα 30: Ο φοιτητής ελέγχει τις λεπτομέρειες που του έθεσε το πανεπιστήμιο

Ο σπουδαστής μπορεί να δει τη συναλλαγή έκδοσης του πιστοποιητικού του εκτελώντας τις ακόλουθες ενέργειες. Παίρνει από τις λεπτομέρειες του λογαριασμού του, τις οποίες είδε μέσω της προηγούμενης εντολής, το hash της συναλλαγής έκδοσης του πιστοποιητικού. Η ετικέτα του hash περιέχει το "transaction_hash". Στο συγκεκριμένο παράδειγμα το hash έχει ετικέτα bachelor_transaction_hash_2020224. Το 2020224 αφορά την ημερομηνία έκδοσης του πιστοποιητικού. Στη συνέχεια, δηλώνει ότι θέλει να εκτελέσει νέα εντολή, επιλέγει την εντολή 8 και εισάγει το hash της συναλλαγής. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.

Εικόνα 31: Ο φοιτητής βλέπει τη συναλλαγή έκδοσης του πιστοποιητικού

Ο σπουδαστής μπορεί να δει απευθείας το πιστοποιητικό (σε καρτέλα), το uri και το hash του πιστοποιητικού, καθώς και το account id του πανεπιστημίου που το εξέδωσε. Για να το πετύχει, δηλώνει ότι θέλει να εκτελέσει νέα εντολή, επιλέγει την εντολή 9 και εισάγει το hash της συναλλαγής. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.

```
Do you want to execute another transaction? Enter Y or N: y
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your assets
5. Get your transactions
6. Get your transactions related to a certain asset
7. Get hash of certificate
8. Get transaction by hash
9. Get uri and hash of certificate by transaction hash
10. Set hash of certificate issuance transaction to employer
11. Set your account details
12. Set another account's details
13. Transfer asset
14. Grant another account permission to set your details
15. Revoke from another account the permission to set your details
Enter number of command and press enter: 9
Hash of transaction: 4bf07676f60d3a039afdd46324e6c09749fe0e61af2e936095d6ff1e77ee00cc
Certificate uri: /Users/silvia/Documents/iroha-master/conf/cert.pdf
Certificate hash: 72aad62224cb3ab37d119f7dd4c10e4919053f5f12c055ef9be999ae229516a4
Certificate creator: ntua@uni
Do you want to execute another transaction? Enter Y or N:
```

Εικόνα 32: Ο φοιτητής βλέπει το πιστοποιητικό, το uri και το hash του πιστοποιητικού, καθώς και το account id του πανεπιστημίου που το εξέδωσε

Για να ανακαλέσει από το πανεπιστήμιο το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του, πρέπει ο σπουδαστής να δηλώσει ότι θέλει να εκτελέσει νέα εντολή, να επιλέξει την εντολή 15 και να εισάγει το εκπαιδευτικό ίδρυμα. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
Do you want to execute another transaction? Enter Y or N: y
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your assets
5. Get your transactions
6. Get your transactions related to a certain asset
7. Get hash of certificate
8. Get transaction by hash
9. Get uri and hash of certificate by transaction hash
10. Set hash of certificate issuance transaction to employer
11. Set your account details
12. Set another account's details
13. Transfer asset
14. Grant another account permission to set your details
15. Revoke from another account the permission to set your details
Enter number of command and press enter: 15
Account: ntua@uni
Transaction hash = b'3076f9c49fac0866a711caf1bc5453ff12717b67a14f25670df1b171836dc86a', creator = student1@stud
[b'3076f9c49fac0866a711caf1bc5453ff12717b67a14f25670df1b171836dc86a', ('ENOUGH_SIGNATURES_COLLECTED', 9, 0), ('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]
```

Εικόνα 33: Ο σπουδαστής ανακαλεί από πανεπιστήμιο το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του

6.4.2 Έλεγχος γνησιότητας πιστοποιητικού

Έστω ότι ο φοιτητής “student1@stud” θέλει να δώσει στον εργοδότη “employer@empl” τις απαραίτητες πληροφορίες, ώστε να μπορεί να ελέγξει τη γνησιότητα του bachelor πιστοποιητικού του. Θα πρέπει να πραγματοποιηθούν τα εξής:

- Ο “employer@empl” πρέπει να παραχωρήσει στο φοιτητή το δικαίωμα να μπορεί να θέτει τις λεπτομέρειες του λογαριασμού του.

Για να εκτελέσει τη συγκεκριμένη ενέργεια, ο εργοδότης κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 9, η οποία αντιστοιχεί στην παραχώρηση του συγκεκριμένου δικαιώματος, και εισάγει το σπουδαστή στον οποίο επιθυμεί να δώσει αυτό το δικαίωμα. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
Account Id: employer@empl
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your transactions
5. Get hash of certificate
6. Get transaction by hash
7. Get uri and hash of certificate by transaction hash
8. Set your account details
9. Grant another account permission to set your details
10. Revoke from another account the permission to set your details
Enter number of command and press enter: 9
Account: student1@stud
Transaction hash = b'878f1c12e84667d4446ce1f9626e1477b44152ef64e02457e7c23b7be1d61115', creator = employer@empl
[b'878f1c12e84667d4446ce1f9626e1477b44152ef64e02457e7c23b7be1d61115', ('ENOUGH_SIGNATURES_COLLECTED', 9, 0), ('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]
Do you want to execute another transaction? Enter Y or N:
```

Εικόνα 34: Ο εργοδότης παραχωρεί σε σπουδαστή το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του

- Ο χρήστης “student1@stud” πρέπει να δώσει στον εργοδότη τις απαραίτητες πληροφορίες.

Αρχικά, πρέπει να βρει το hash της συναλλαγής έκδοσης του πιστοποιητικού. Για να εκτελέσει αυτή την ενέργεια, ο σπουδαστής κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 3, μέσω της οποίας μπορεί να δει τις λεπτομέρειες που του έθεσε ένας άλλος χρήστης, και εισάγει το εκπαιδευτικό ίδρυμα που εξέδωσε το πιστοποιητικό. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your assets
5. Get your transactions
6. Get your transactions related to a certain asset
7. Get hash of certificate
8. Get transaction by hash
9. Get uri and hash of certificate by transaction hash
10. Set hash of certificate issuance transaction to employer
11. Set your account details
12. Set another account's details
13. Transfer asset
14. Grant another account permission to set your details
15. Revoke from another account the permission to set your details
Enter number of command and press enter: 3
Account: ntua@uni
account_detail_response {
  detail: "{\"ntua@uni\" : {\"bachelor_uri_2020224\": \"/Users/silvia/Documents/iroha-master/conf/cert.pdf\", \"bachelor_hash_2020224\": \"72aad62224cb3ab37d119f7dd4c10e491905
3f5f12c055ef9be999ae229516a4\", \"bachelor_transaction_hash_2020224\": \"4bf07676f60d3a039afdd46324e6c09749fe0e61af2e936095d6ff1e77ee00cc\"}}"
}
query_hash: "18a1633232f1bb42c39acd7b22c3f9b1a725b58efe67a543440241e0eb0edf30"
```

Εικόνα 35: Ο φοιτητής ελέγχει τις λεπτομέρειες που του έθεσε το πανεπιστήμιο

Για να δώσει στον εργοδότη το hash της συναλλαγής έκδοσης του πιστοποιητικού πρέπει να πάρει από τις λεπτομέρειες του λογαριασμού του, τις οποίες είδε μέσω της προηγούμενης εντολής, το hash της συναλλαγής έκδοσης του πιστοποιητικού. Η ετικέτα του hash περιέχει το “transaction_hash”. Στο συγκεκριμένο παράδειγμα το hash έχει ετικέτα

bachelor_transaction_hash_2020224. Στη συνέχεια, δηλώνει ότι θέλει να εκτελέσει νέα εντολή, επιλέγει την εντολή 10 και εισάγει το hash της συναλλαγής, το account id του εργοδότη και την ετικέτα της λεπτομέρειας που θέλει να θέσει. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.

```
Do you want to execute another transaction? Enter Y or N: y
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your assets
5. Get your transactions
6. Get your transactions related to a certain asset
7. Get hash of certificate
8. Get transaction by hash
9. Get uri and hash of certificate by transaction hash
10. Set hash of certificate issuance transaction to employer
11. Set your account details
12. Set another account's details
13. Transfer asset
14. Grant another account permission to set your details
15. Revoke from another account the permission to set your details
Enter number of command and press enter: 10
Hash of transaction: 4bf07676f60d3a039afdd46324e6c09749fe0e61af2e936095d6ff1e77ee00cc
Account: employer@empl
Name of detail: student1_bachelor
Certificate uri: /Users/silvia/Documents/iroha-master/conf/cert.pdf
Certificate hash: 72aad62224cb3ab37d119f7dd4c10e4919053f5f12c055ef9be999ae229516a4
Certificate creator: ntua@uni
Transaction hash = b'4124af9dce50cb1f9d507b870dc65ec550bdf4a2a972d991af33119458519323', creator = student1@stud
[b'4124af9dce50cb1f9d507b870dc65ec550bdf4a2a972d991af33119458519323', ('ENOUGH_SIGNATURES_COLLECTED', 9, 0), ('STATEFUL_VALIDATION_SUCCESS', 3, 0), ('COMMITTED', 5, 0)]
```

Εικόνα 36: Ο φοιτητής δίνει το hash της συναλλαγής έκδοσης του πιστοποιητικού στον εργοδότη

- Ο “employer@empl” πρέπει, στη συνέχεια, να ελέγξει ότι έχει πρόσβαση στο πιστοποιητικό, να ανακαλέσει από το φοιτητή το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του και να ελέγξει το hash του πιστοποιητικού.

Αρχικά, πρέπει να δει τις λεπτομέρειες του λογαριασμού του που έχει θέσει ο φοιτητής. Για να εκτελέσει αυτή την ενέργεια, κάνει login με το account id του στην εφαρμογή, επιλέγει την εντολή 3, μέσω της οποίας μπορεί να δει τις λεπτομέρειες που του έθεσε ένας άλλος χρήστης, και εισάγει το account id του σπουδαστή. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
Account Id: employer@empl
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your transactions
5. Get hash of certificate
6. Get transaction by hash
7. Get uri and hash of certificate by transaction hash
8. Set your account details
9. Grant another account permission to set your details
10. Revoke from another account the permission to set your details
Enter number of command and press enter: 3
Account: student1@stud
account_detail_response {
  detail: "{\"student1@stud\" : {\"student1_bachelor\": \"4bf07676f60d3a039afdd46324e6c09749fe0e61af2e936095d6ff1e77ee00cc\"}}"
}
query_hash: "3df67d9f6a1f0e4b5b38ac79d58422557b731c9be2a14c0efc514488df8bcc27"

Do you want to execute another transaction? Enter Y or N:
```

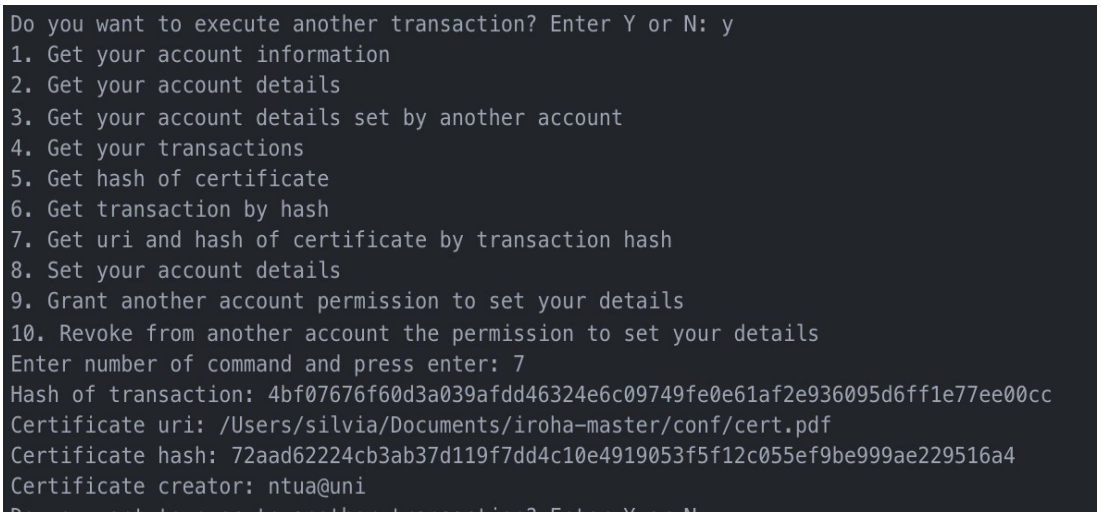
Εικόνα 37: Ο εργοδότης ελέγχει τις λεπτομέρειες που του έθεσε ο φοιτητής

Ο εργοδότης μπορεί να δει τη συναλλαγή έκδοσης του πιστοποιητικού εκτελώντας τις ακόλουθες ενέργειες. Παίρνει από τις λεπτομέρειες του λογαριασμού του, τις οποίες είδε μέσω της προηγούμενης εντολής, το hash της συναλλαγής έκδοσης του πιστοποιητικού. Στη συνέχεια, δηλώνει ότι θέλει να εκτελέσει νέα εντολή, επιλέγει την εντολή 6 και εισάγει το hash της συναλλαγής. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.

```
Enter number of command and press enter: 6
Hash of transaction: 4bf07676f60d3a039afdd46324e6c09749fe0e61af2e936095d6ff1e77ee00cc
transactions_response {
  transactions {
    payload {
      reduced_payload {
        commands {
          add_asset_quantity {
            asset_id: "bachelor#cert"
            amount: "1"
          }
        }
        commands {
          transfer_asset {
            src_account_id: "ntua@uni"
            dest_account_id: "student1@stud"
            asset_id: "bachelor#cert"
            description: "bachelorissuance"
            amount: "1"
          }
        }
        commands {
          set_account_detail {
            account_id: "student1@stud"
            key: "bachelor_uri_2020224"
            value: "/Users/silvia/Documents/iroha-master/conf/cert.pdf"
          }
        }
        commands {
          set_account_detail {
            account_id: "student1@stud"
            key: "bachelor_hash_2020224"
            value: "72aad62224cb3ab37d119f7dd4c10e4919053f5f12c055ef9be999ae229516a4"
          }
        }
        creator_account_id: "ntua@uni"
        created_time: 1582556104686
        quorum: 1
      }
    }
    signatures {
      public_key: "8135e2242155e26c3941f94bdae34bf647c00aaafa333e41dacc291af090a8a5"
      signature: "13a99f6df353e757773a4d062de986d9f7811978c0334d39db8bf58c8de8453de218916f50b7675429c2c438c7e9028c3b5d08070cad298b8c24bc568623ec07"
    }
  }
}
query_hash: "fb61f6318a5c6b4b79bf337ad66b9687e52cb2837eee2f14d9cd357fa15c5034"
```

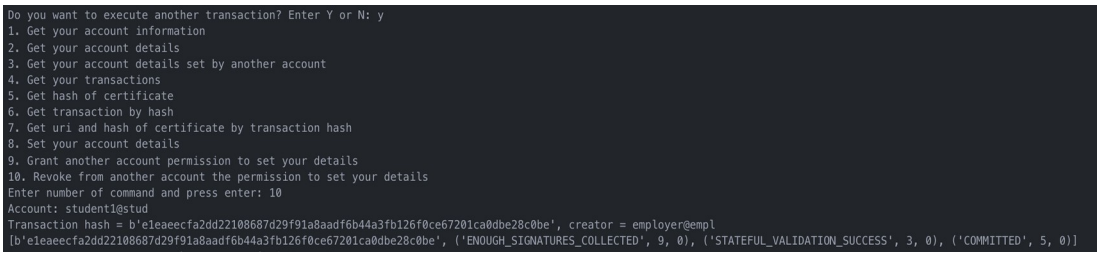
Εικόνα 38: Ο εργοδότης βλέπει τη συναλλαγή έκδοσης του πιστοποιητικού

Ο εργοδότης μπορεί να δει απευθείας το πιστοποιητικό (σε καρτέλα), το uri και το hash του πιστοποιητικού, καθώς και το account id του πανεπιστημίου που το εξέδωσε. Για να το πετύχει, δηλώνει ότι θέλει να εκτελέσει νέα εντολή, επιλέγει την εντολή 7 και εισάγει το hash της συναλλαγής. Η διαδικασία αυτή φαίνεται στην ακόλουθη εικόνα.



Εικόνα 39: Ο εργοδότης βλέπει το πιστοποιητικό, το uri και το hash του πιστοποιητικού, καθώς και το account id του πανεπιστημίου που το εξέδωσε

Για να ανακαλέσει από το φοιτητή το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του, πρέπει ο εργοδότης να δηλώσει ότι θέλει να εκτελέσει νέα εντολή, να επιλέξει την εντολή 10 και να εισάγει το account id του φοιτητή. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.



Εικόνα 40: Ο εργοδότης ανακαλεί από το φοιτητή το δικαίωμα να θέτει τις λεπτομέρειες του λογαριασμού του

Για να ελέγξει το hash του πιστοποιητικού, πρέπει ο εργοδότης να δηλώσει ότι θέλει να εκτελέσει νέα εντολή, να επιλέξει την εντολή 5 και να εισάγει το uri του πιστοποιητικού. Η όλη διαδικασία φαίνεται στην ακόλουθη εικόνα.

```
Do you want to execute another transaction? Enter Y or N: y
1. Get your account information
2. Get your account details
3. Get your account details set by another account
4. Get your transactions
5. Get hash of certificate
6. Get transaction by hash
7. Get uri and hash of certificate by transaction hash
8. Set your account details
9. Grant another account permission to set your details
10. Revoke from another account the permission to set your details
Enter number of command and press enter: 5
Uri: /Users/silvia/Documents/iroha-master/conf/cert.pdf
72aad62224cb3ab37d119f7dd4c10e4919053f5f12c055ef9be999ae229516a4
Do you want to execute another transaction? Enter Y or N:
```

Εικόνα 41: Ο εργοδότης ελέγχει το hash του πιστοποιητικού

7 *Συμπεράσματα και Μελλοντικές Προοπτικές*

Όπως έχει αναφερθεί στο "Κεφάλαιο 1: Εισαγωγή", η πιστοποίηση των τίτλων σπουδών κατέχει κεντρικό ρόλο στον τομέα της εκπαίδευσης, αφού είναι απαραίτητη για την απόδειξη των προσόντων των εκπαιδευόμενων. Ωστόσο, οι δομές στις οποίες φυλάσσονται τα πιστοποιητικά είναι κεντρικές και η πρόσβαση σε αυτές περιορίζεται στο προσωπικό του εκάστοτε εκπαιδευτικού ιδρύματος. Συνεπώς, όχι μόνο είναι τα πιστοποιητικά ευάλωτα σε απώλεια και καταδολίευση, αλλά απαιτούνται και χρονοβόρες διαδικασίες για την έκδοση και την επικύρωσή τους.

Για την ενίσχυση της αξιοπιστίας των εκπαιδευτικών πιστοποιητικών και τη διευκόλυνση του ελέγχου της εγκυρότητάς τους, είναι απαραίτητο να γίνουν αλλαγές στον τρόπο δόμησης και έκδοσης των πιστοποιητικών. Πιο συγκεκριμένα, κρίνεται αναγκαία η υιοθέτηση τεχνολογιών που επιτρέπουν την καθολική δόμηση των πιστοποιητικών και την αυτοματοποίηση των διαδικασιών έκδοσης και επικύρωσής τους.

Το blockchain δίνει λύση στα παραπάνω χρόνια προβλήματα που σχετίζονται με το χώρο της εκπαίδευσης, αφού δίνει τη δυνατότητα αποκεντρωμένης δόμησης δεδομένων σε μία κατανεμημένη συλλογή λογαριασμών, ενώ ταυτόχρονα, δεν επιτρέπει τη μεταβολή των δεδομένων που έχουν αποθηκευτεί σε αυτό.

Τα τελευταία χρόνια, έχει γίνει αντιληπτό ότι η εφαρμογή της τεχνολογίας blockchain σε τομείς πέρα από τα κρυπτονομίσματα έχει ιδιαίτερη αξία. Τόσο η ερευνητική κοινότητα, όσο και διάφοροι οργανισμοί έχουν ήδη αρχίσει να μελετούν και να χρησιμοποιούν το blockchain προκειμένου να αναπτύξουν λύσεις για προβλήματα του χώρου της εκπαίδευσης.

Ωστόσο, η συγκεκριμένη τεχνολογία δεν είναι ακόμα γνωστή στο ευρύ κοινό. Συνεπώς, η ανάπτυξη λύσης, που ικανοποιεί το εκάστοτε πρόβλημα στο βέλτιστο δυνατό βαθμό, απαιτεί την ενδελεχή έρευνα των διαφόρων πλατφόρμων που κυκλοφορούν. Όσον αφορά το παρόν πρόβλημα, θεωρήθηκε ότι το καταλληλότερο framework είναι το Hyperledger Iroha, χωρίς αυτό να αποκλείει τη δυνατότητα

χρήσης διαφορετικών frameworks.

Η ανάπτυξη εφαρμογής έκδοσης και ελέγχου της εγκυρότητας εκπαιδευτικών πιστοποιητικών επιβεβαιώνει την καταλληλότητα του blockchain για την επίλυση των χρόνιων προβλημάτων που σχετίζονται με το χώρο της εκπαίδευσης. Ταυτόχρονα, καταδεικνύει το γεγονός ότι η δημιουργία ασφαλών εφαρμογών, οι οποίες χρησιμοποιούν τη συγκεκριμένη τεχνολογία, δεν είναι ιδιαίτερα δύσκολη.

Τόσο το blockchain ως τεχνολογία, όσο και η χρήση του σε τομείς πέρα από τα κρυπτονομίσματα έχουν περαιτέρω περιθώρια μελέτης και εξέλιξης, καθώς η μελέτη τους ξεκίνησε πρόσφατα. Για το λόγο αυτό, προτείνονται οι εξής μελλοντικές επεκτάσεις:

- Η υλοποίηση της εφαρμογής με χρήση των άλλων δύο frameworks που παρουσιάστικαν (Ethereum και Hyperledger Fabric), καθώς και η μελέτη για το αν θα μπορούσαν να χρησιμοποιηθούν άλλες υπάρχουσες πλατφόρμες κρυπτονομισμάτων.
- Η επέκταση της εφαρμογής ώστε να δίνεται στα εκπαιδευτικά ιδρύματα η δυνατότητα να προσθέτουν και badges στους σπουδαστές τα οποία θα αντιστοιχούν στα μαθήματα που έχουν παρακολουθήσει επιτυχώς.
- Η επέκταση της εφαρμογής ώστε να μπορούν να εκδοθούν πιστοποιητικά και από άλλους φορείς, τα οποία μπορεί να αφορούν διπλώματα, εγκεκριμένα online courses ή ακόμα και την εργασιακή εμπειρία ενός ατόμου.
- Η εφαρμογή της μεθοδολογίας που παρουσιάστηκε στην παρούσα διπλωματική εργασία σε τομείς πέρα από την εκπαίδευση.
- Η προσθήκη οντολογιών για τη διασύνδεση των δεδομένων που είναι αποθηκευμένα σε blockchain με άλλα διασυνδεδεμένα δεδομένα (Linked Data).

8 *Βιβλιογραφία*

1. Drescher, D.: Blockchain Basics: A Non-technical Introduction in 25 Steps, 1st edn. Apress, Frankfurt am Main (2017)
2. Christos Kontzinos, Michael Kontoulis, Panagiotis Kapsalis, Ourania Markaki, Spiros Mouzakitis, Rano Manta, Thelma Androutsou, Ioannis Kouris, H. Karanikas, A. Billiris, A. Christodoulakis, E. Thireos: Methodology for secure storage and information exchange of medical data based on blockchain
3. Michele Ruta, Floriano Scioscia, Saverio Ieva, Giovanna Capurso, Eugenio Di Sciascio: Semantic Blockchain to Improve Scalability in the Internet of Things (2017)
4. Umar Rashid, Allan Third, John Domingue: Web Service for Semantic Negotiation of Smart Contracts
5. José G. Faísca, José Q. Rogado: Decentralized Semantic Identity
6. Matthew English, Soren Auer and John Domingue: Block Chain Technologies & The Semantic Web: A Framework for Symbiotic Development
7. https://herug2019.de/storage/app/uploads/public/5ce/7d8/5e4/Blockchain-based%20Digital%20Credentials%20-%20Saeys,%20Okamoto,%20Jonkers.pdf (2019)
8. https://www.unic.ac.cy/iff/blockchain-certificates/ (2019)
9. http://certificates.media.mit.edu/ (2019)
10. https://en.wikipedia.org/wiki/Merkle_tree (2019)
11. https://en.wikipedia.org/wiki/Public-key_cryptography (2019)
12. Satoshi Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System
13. Fedor Muratov, Andrei Lebedev, Nikolai Iushkevich, Bulat Nasrulin, Makoto Takemiya: YAC: BFT Consensus Algorithm for Blockchain (2018)
14. https://ethereum-homestead.readthedocs.io/en/latest/index.html (2020)
15. https://hyperledger-fabric.readthedocs.io/en/release-1.4/index.html (2020)

16. https://iroha.readthedocs.io/en/latest/index.html (2020)
17. https://en.wikipedia.org/wiki/Docker_(software) (2020)
18. https://en.wikipedia.org/wiki/Pip_(package_manager) (2020)