ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

Μελέτη των Τεχνολογιών του Blockchain και Χρήση τους για τη Δημιουργία ενός Συστήματος Σύναψης Συμβολαίων μεταξύ Ασθενών και Ασφαλιστικών Εταιριών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευθυμίου Κ. Χονδρογιάννη

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ: Θεοδώρα Βαρβαρίγου, Καθηγήτρια ΕΜΠ

ΣΕΠΤΕΜΒΡΙΟΣ 2020

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

Μελέτη των Τεχνολογιών του Blockchain και Χρήση τους για τη Δημιουργία ενός Συστήματος Σύναψης Συμβολαίων μεταξύ Ασθενών και Ασφαλιστικών Εταιριών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευθυμίου Κ. Χονδρογιάννη

Συμβουλευτική Επιτροπή : Θεοδώρα Α. Βαρβαρίγου

Δημήτριος Ασκούνης

Εμμανουήλ Βαρβαρίγος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 29η Σεπτεμβρίου 2020.

Αθήνα, Σεπτέμβριος 2020

.....	.....	.....
Θεοδώρα Α. Βαρβαρίγου	Δημήτριος Ασκούνης	Εμμανουήλ Βαρβαρίγος
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

.....

Ευθύμιος Κ. Χονδρογιάννης

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

Πρόλογος

Στα πλαίσια της Διπλωματικής μου εργασίας για το Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών (ΔΠΜΣ) «Τεχνο-Οικονομικά Συστήματα» ασχολήθηκα με τη μελέτη των τεχνολογιών που χρησιμοποιήθηκαν για τη δημιουργία των κρυπτονομισμάτων και ειδικότερα του Bitcoin και του Ethereum. Η εργασία αυτή μου έδωσε τη δυνατότητα να μελετήσω την αρχιτεκτονική των συστημάτων αυτών, τις δομές δεδομένων τους καθώς επίσης και τους αλγορίθμους που χρησιμοποιούνται για την επίτευξη συναίνεσης μεταξύ μη έμπιστων χρηστών. Επίσης, με ώθησε να μελετήσω διάφορα αποκεντροποιημένα συστήματα που έχουν αναπτυχθεί χρησιμοποιώντας τις τεχνολογίες αυτές καθώς επίσης και τις αδυναμίες τους.

Επιπρόσθετα, μου έδωσε τη δυνατότητα να αναπτύξω ένα νέο σύστημα που επιτρέπει τη σύναψη συμβολαίων μεταξύ ασθενών και ασφαλιστικών εταιριών που δραστηριοποιούνται στον χώρο της υγείας. Τα συμβόλαια αυτά εγγυώνται την άμεση αποζημίωση των ασθενών, εφόσον πληρούνται οι όροι του συμβολαίου, μετά τη λεπτομερή εξέταση των δεδομένων των ασθενών. Ακόμη, μου επέτρεψε να συνδυάσω τη χρήση των τεχνολογιών του σημασιολογικού ιστού με τις τεχνολογίες του Blockchain για τη σημασιολογική αναπαράσταση τόσο των δεδομένων των ασθενών, όσο και των όρων των συμβολαίων.

Η εργασία που παρουσιάζεται στις επόμενες σελίδες εκπονήθηκε κατά τη διάρκεια του έτους 2020 υπό την επίβλεψη της κ. Θεοδώρας Βαρβαρίγου, την οποία και θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου για την υποστήριξη που μου παρείχε στην προσπάθεια αυτή.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη στήριξή τους όλα αυτά τα χρόνια.

Ευθύμιος Κ. Χονδρογιάννης

Σεπτέμβριος 2020

Η σελίδα αυτή είναι σκόπιμα λευκή

Πίνακας Περιεχομένων

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

Περίληψη

Ο αρμονικός συνδυασμός των ομότιμων συστημάτων (peer-to-peer systems) με την κρυπτογραφία δημόσιου κλειδιού και τους αλγορίθμους συναίνεσης επέτρεψαν την πραγματοποίηση οικονομικών συναλλαγών μεταξύ αναξιόπιστων χρηστών χωρίς την ανάγκη ύπαρξης ενός διαμεσολαβητή. Στην καρδιά του συστήματος αυτού βρίσκεται ένα ελευθέρως προσβάσιμο διανεμημένο βιβλίο (public distributed ledger aka Blockchain) που περιέχει τις συναλλαγές που έχουν πραγματοποιηθεί και συνεχώς επεκτείνεται με νέες συναλλαγές, χωρίς να υπάρχει κίνδυνος αμφισβήτησης ή τροποποίησής τους παρά, το γεγονός ότι δεν υπάρχει μια κεντρική αρχή ελέγχου. Σύντομα, η προσέγγιση αυτή χρησιμοποιήθηκε όχι μόνο για την καταγραφή των κρυπτονομισμάτων των χρηστών αλλά και προγραμμάτων τα οποία μπορούσαν να εκτελεστούν από τους κόμβους του δικτύου και να μεταβάλλουν την κατάσταση στην οποία αυτό βρίσκεται. Η προσέγγιση αυτή ώθησε στη δημιουργία μιας πλατφόρμας, η οποία επιτρέπει την ανάπτυξη εφαρμογών που βασίζονται στις τεχνολογίες του Blockchain και σύντομα βρήκε εφαρμογή σε διάφορους τομείς, συμπεριλαμβανομένου του χώρου της υγείας.

Στην εργασία αυτή παρουσιάζουμε ένα κατανεμημένο σύστημα που αναπτύχθηκε χρησιμοποιώντας τις τεχνολογίες του Blockchain, το οποίο επιτρέπει στους ασθενείς να συνάψουν συμβόλαια με ασφαλιστικές εταιρίες υγείας (health insurance companies), τα οποία εγγυώνται την άμεση αποζημίωση των ασθενών, εφόσον πληρούνται οι όροι του συμβολαίου. Για τη σημασιολογική αναπαράσταση τόσο των δεδομένων των ασθενών, όσο και των όρων του συμβολαίου βασιστήκαμε σε υπάρχοντα μοντέλα και συστήματα κωδικοποίησης, τα οποία ακολούθως εκφράσαμε, χρησιμοποιώντας τις τεχνολογίες του σημασιολογικού ιστού. Ωστόσο, τα δεδομένα των ασθενών, αποθηκεύονται εκτός της πλατφόρμας του Blockchain. Η προσέγγιση που ακολουθήσαμε δίνει τη δυνατότητα στον χρήστη να έχει άμεσο έλεγχο στα δεδομένα που αποθηκεύονται κατά την επίσκεψή του σε ένα κέντρο περίθαλψης και πώς αυτά χρησιμοποιούνται, καθώς επίσης και να συνάψει συμφωνίες με τις ασφαλιστικές εταιρίες για πιθανή εκδήλωση ασθενειών και την εξέταση των επιμέρους συνθηκών λαμβάνοντας υπόψη την σημασία των όρων.

Το σύστημα αυτό «υποφέρει» από τις αδυναμίες του Blockchain και των Smart Contracts, οι οποίες μπορούν, σε ορισμένες περιπτώσεις, να αποβούν καταστροφικές για το σύστημα. Επίσης, εξαρτάται σε μεγάλο βαθμό από τα Web Services που

αναλαμβάνουν τον έλεγχο των δεδομένων των ασθενών. Ενδεχομένως, η πιστοποίηση και ο περιορισμός των οντοτήτων που έχουν πρόσβαση στα δεδομένα να μπορεί να βελτιώσει την ασφάλεια και την αξιοπιστία του συστήματος, ενώ παράλληλα να συμβάλει στην βελτιστοποίηση του τρόπου εκτέλεσης των επιμέρους λειτουργιών. Ωστόσο, η προσέγγιση αυτή προϋποθέτει τη χρήση διαφορετικών εργαλείων που δίνουν μεγαλύτερη ελευθερία στον χρήστη κατά την παραμετροποίηση της πλατφόρμας του Blockchain και τον καθορισμό των αλγορίθμων που θα χρησιμοποιηθούν.

Λέξεις Κλειδιά

Πλατφόρμα Ethereum, Αλυσίδα από Μπλοκ, Έξυπνα Συμβόλαια, Ασφάλεια Υγείας, Οντολογίες

Abstract

The seamless combination of peer-to-peer systems (p2p) with public key encryption and consensus algorithms allowed financial transactions between unreliable users to take place without the need for intermediaries. At the heart of this system is a public distributed ledger (aka Blockchain) with the transactions committed, and is constantly being enriched with new ones, accepted by all entities being involved, despite the fact that there is no central authority. Soon, this approach was used not only to record users' cryptocurrencies (e.g., Bitcoin) but also programs that could run from network nodes and change its state (e.g., Ethereum). In this way, the system is actually a platform that allows users to develop their own applications based on Blockchain technologies and soon applied in various fields, including the systems developed in the health domain.

In this work, we present a distributed system (DApp) developed using Blockchain technologies which allows patients to sign a contract with a health insurance company. The latter guarantees the immediate compensation of patients provided that the terms of the contract are met. For the semantic representation of both patient data and contract terms, we have used existing models and international coding systems, which we have formally expressed using the Semantic Web technologies. However, data of each patient were stored off-Blockchain. The approach we have followed allows users to directly control the data produced during their visit to a healthcare entity, as well as come into agreements with insurance companies regarding their health status (e.g., the manifestation of a possible disease), which can temporarily access the patient data and accordingly examine the contract conditions, taking into account the meaning of clinical terms.

The developed system "suffers" from the weaknesses of Blockchain technology and Smart Contracts, which in some cases can negatively affect the whole system. Also, it depends on the external web services used for evaluating the smart contract conditions. The certification and restriction of the entities accessing the patient data can increase the security and reliability of the system. Meanwhile, it can also contribute to the improvement of specific Blockchain operations. However, this approach requires the use of different tools and platforms that give more freedom to the user when configuring the Blockchain platform and determining the algorithms used.

Keywords:

Ethereum, Blockchain, Smart Contracts, Health Insurance, Ontologies

1 Εισαγωγή

Το Bitcoin έφερε επανάσταση στον χώρο των κρυπτονομισμάτων, καθώς επέτρεψε την πραγματοποίηση οικονομικών συναλλαγών μεταξύ αναξιόπιστων χρηστών χωρίς την ανάγκη ύπαρξης ενός μεσολαβητή (π.χ., τράπεζας), όπως πρωτοπαρουσίασε ο Nakamoto το 2008 [1]. Το Bitcoin βασίζεται στην ύπαρξη ενός ελευθέρως προσβάσιμου διανεμημένου ψηφιακού βιβλίου, γνωστό ως public distributed ledger ή απλά Blockchain (λόγω της δομής οργάνωσης που χρησιμοποιείται), το οποίο περιέχει όλες τις συναλλαγές που έχουν λάβει χώρα και συνεχώς επεκτείνεται με νέες συναλλαγές, οι οποίες προστατεύονται χρησιμοποιώντας την κρυπτογραφία δημόσιου κλειδιού [2][3]. Σύντομα, διαπιστώθηκε ότι οι τεχνολογίες που χρησιμοποιούνται και ειδικότερα το Blockchain θα μπορούσε να χρησιμοποιηθεί όχι μόνο για την αποθήκευση δεδομένων, όπως τα χρήματα που ανήκουν στον κάθε χρήστη αλλά και για προγράμματα, τα οποία θα μπορούσαν να εκτελεστούν από τους κόμβους του δικτύου. Η πιο γνωστή πλατφόρμα, που υποστηρίζει την αποθήκευση και εκτέλεση προγραμμάτων χρησιμοποιώντας τις τεχνολογίες του Blockchain, είναι το Ethereum που ξεκίνησε το 2013 [4] και σύντομα διεκδίκησε σημαντικό μέρος της αγοράς των κρυπτονομισμάτων, αλλά και επέτρεψε τη δημιουργία κατανεμημένων εφαρμογών, γνωστών ως Distributed Applications (DApps), διευρύνοντας έτσι το πεδίο εφαρμογών των τεχνολογιών Blockchain. Ακολούθως, αναπτύχθηκαν διάφορες εφαρμογές, που εκμεταλλεύονται τα πλεονεκτήματα που προσφέρει η τεχνολογία αυτή για τη δημιουργία τοπικών νομισμάτων, που μπορούν να καλύψουν τις ανάγκες των εφαρμογών αλλά και τη σύναψη συμβολαίων μεταξύ των οντοτήτων, που είναι γνωστά ως έξυπνα συμβόλαια (smart contracts). Η τεχνολογία αυτή επηρέασε την ανάπτυξη των εφαρμογών σε διαφόρους τομείς μεταξύ των οποίων και τις εφαρμογές στον χώρο της υγείας [5].

Το Blockchain [6] μας επιτρέπει να τοποθετήσουμε τα δεδομένα μας στους κόμβους της αλυσίδας, αλλά η πρόσβαση στα δεδομένα αυτά να ελέγχεται εξολοκλήρου από τον χρήστη (Blockchain 1.0) ή από προγράμματα, γνωστά ως smart contracts (Blockchain 2.0). Οι αλγόριθμοι συναίνεσης που χρησιμοποιούνται μας διασφαλίζουν ότι τα δεδομένα που υπάρχουν στην αλυσίδα (τόσο τα χρήματα όσο και τα προγράμματα) δεν

μπορούν να τροποποιηθούν (immutable) και μας επιτρέπουν την πραγματοποίηση οικονομικών συναλλαγών (π.χ., αγορά ενός προϊόντος ή υπηρεσίας) χωρίς την ανάγκη ύπαρξης μιας κεντρικής αρχής ελέγχου. Βέβαια, όλοι οι χρήστες μπορούν να δουν τα δεδομένα που υπάρχουν στο Blockchain (transparency) και επομένως, να εντοπίσουν τα δεδομένα που ανήκουν σε έναν λογαριασμό και ενδεχομένως να φτάσουν ακόμη και στο πραγματικό πρόσωπο στο οποίο ανήκει ο λογαριασμός αυτός. Ωστόσο, αυτό μπορούμε να το εμποδίσουμε είτε διατηρώντας τα ευαίσθητα προσωπικά δεδομένα των χρηστών σε αποθήκες δεδομένων, που βρίσκονται εκτός της αλυσίδας (off Blockchain) [7][8], είτε αποθηκεύοντάς τα και αυτά στην αλυσίδα αλλά περιορίζοντας αισθητά τις οντότητες που θα μπορούσαν να έχουν πρόσβαση σε αυτήν, όπως στην περίπτωση του Medicalchain[1]. Συνεπώς, η τεχνολογία αυτή θα μπορούσε να χρησιμοποιηθεί για την αποτελεσματική αποθήκευση και διαχείριση των ιατρικών δεδομένων των χρηστών ή μέρος αυτών, τα οποία θα μπορούσαν να ελεγχθούν από ασφαλιστικές εταιρίες υγείας, προκειμένου να δουν εάν ο ασθενής ακολούθησε τη θεραπεία που έπρεπε και ενδεχομένως να τον αποζημιώσουν άμεσα, σε περίπτωση που ο ασθενής εκδήλωσε κάποια μη επιθυμητά συμπτώματα ή ασθένειες.

1.1 Στόχος Εργασίας

Στα πλαίσια της εργασίας αυτής θα ασχοληθούμε με τη δημιουργία ενός κατανεμημένου συστήματος (DApp), το οποίο μας επιτρέπει (α) να συνάψουμε συμφωνίες με ασφαλιστικές εταιρίες που δραστηριοποιούνται στον χώρο της υγείας και (β) να αποζημιωθούμε άμεσα από τις εταιρίες αυτές (καταθέτοντας το αντίστοιχο ποσό στον λογαριασμό μας), όταν πληρούνται οι όροι της συμφωνίας. Απαραίτητη βέβαια προϋπόθεση είναι τόσο εμείς, όσο και η ασφαλιστική εταιρία να μπορούν να έχουν πρόσβαση στα δεδομένα των ασθενών, προκειμένου να ελέγξουν εάν πληρούνται οι όροι του συμβολαίου και να προβούν στις αντίστοιχες ενέργειες. Για τον λόγο αυτό, κάθε φορά που επισκεπτόμαστε ένα κέντρο περίθαλψης, αυτό θα πρέπει να ενημερώνει το σύστημα για τα δεδομένα που έχουν προκύψει για έναν ασθενή, ενώ παράλληλα ο έλεγχος των δεδομένων που αποθηκεύονται και των οντοτήτων που έχουν πρόσβαση σε αυτά θα γίνεται εξ ολοκλήρου από τον χρήστη/ασθενή.

[1] Medicalchain, https://medicalchain.com/en/

Για τη διαλειτουργική αναπαράσταση τόσο των δεδομένων των ασθενών, όσο και των όρων ενός συμβολαίου θα βασιστούμε σε υπάρχοντα πρότυπα και κωδικοποιήσεις που είναι διεθνώς αποδεκτά και κατ' επέκταση διευκολύνουν την επικοινωνία των χρηστών με τα συστήματα αυτά. Για την αποθήκευση των δεδομένων των ασθενών που προκύπτουν κατά την επίσκεψή τους σε ένα κέντρο περίθαλψης καθώς και τη σύναψη των συμβολαίων των ασθενών με την ασφαλιστική εταιρία θα δημιουργήσουμε τα κατάλληλα smart contracts, τα οποία επιτρέπουν στον χρήστη/ασθενή να εντοπίσει τα δεδομένα που έχουν αποθηκευτεί καθώς και τις οντότητες που έχουν πρόσβαση σε αυτά. Ωστόσο, στην παρούσα εργασία, τα επιμέρους δεδομένα θα αποθηκεύονται σε μια σχεσιακή βάση που βρίσκεται εκτός της αλυσίδας. Η συμφωνία μεταξύ των ασθενών και των ασφαλιστικών εταιριών θα αποθηκεύεται στο Blockchain, έτσι ώστε να μην μπορεί να αλλαχθεί. Ωστόσο, για την επικοινωνία με τη βάση και τον έλεγχο των δεδομένων των χρηστών θα υλοποιηθούν τα απαραίτητα Web Services (γνωστά ως oracles), που θα εξετάζουν τη βάση και θα ενημερώνουν αντίστοιχα την αλυσίδα. Τέλος, να αναφέρουμε ότι ο χρήστης θα προκαταβάλει το ποσό που απαιτείται για τη σύναψη του συμβολαίου, ενώ, από τη μεριά της εταιρίας, θα γίνεται η απαραίτητη δέσμευση των χρημάτων της για την αποζημίωση του χρήστη, τα οποία η εταιρία θα μπορεί να πάρει πίσω μετά τη λήξη της ασφάλειας.

1.2 Δομή Εγγράφου

Στις ακόλουθες ενότητες αρχικά θα περιγράψουμε τις τεχνολογίες πάνω στις οποίες έχει βασιστεί το Blockchain και πώς αυτές έχουν χρησιμοποιηθεί για τη δημιουργία του κρυπτονομίσματος Bitcoin (Κεφάλαιο 2) καθώς επίσης και της πλατφόρμας εκτέλεσης προγραμμάτων Ethereum (Κεφάλαιο 3). Ειδικότερα, θα δούμε την αλληλεπίδραση του χρήστη με τα συστήματα αυτά, τις δομές δεδομένων που χρησιμοποιούνται καθώς επίσης και τους αλγορίθμους/πρωτόκολλα που τρέχουν σε κάθε κόμβο του δικτύου για την ενημέρωση της αλυσίδας και την εκτέλεση των προγραμμάτων. Ο στόχος των κεφαλαίων αυτών δεν είναι να δώσουμε στον αναγνώστη μια λεπτομερή και εξαντλητική περιγραφή όλων των θεμάτων που σχετίζονται με το Blockchain και τα Smart Contracts, αλλά να δείξουμε τις τεχνολογίες στις οποίες αυτά βασίζονται, τον τρόπο με τον οποίο συνδυάζονται μεταξύ τους, τα βασικά πεδία των δομών δεδομένων και γενικότερα τη φιλοσοφία που υπάρχει πίσω από τις έννοιες αυτές.

Σε αρκετές περιπτώσεις παραπέμπουμε τον χρήστη σε σχετικά βιβλία και δημοσιεύσεις σε διεθνή συνέδρια και περιοδικά, στην περίπτωση που θα ήθελε να εμβαθύνει σε κάποια από τα παραπάνω θέματα.

Ακολούθως θα προχωρήσουμε στην περιγραφή του συστήματος με το οποίο ασχοληθήκαμε στην εργασία αυτή, το οποίο δεν είναι άλλο από τη δημιουργία ενός «κατανεμημένου συστήματος» (DApp), το οποίο επιτρέπει στους ασθενείς να συνάψουν συμβόλαια με κάποια ασφαλιστική εταιρία υγείας, χρησιμοποιώντας τις τεχνολογίες του Blockchain. Ειδικότερα, στο Κεφάλαιο 4 θα αναφερθούμε στις υπηρεσίες που παρέχει το σύστημα που αναπτύχθηκε για κάθε μία από τις τρεις βασικές οντότητες τους συστήματος (ασθενείς, κέντρα περίθαλψης, ασφαλιστικές εταιρίες υγείας) καθώς επίσης την προσέγγιση που ακολουθήσαμε για την αναπαράσταση των δεδομένων των ασθενών και την έκφραση των όρων των συμβολαίων. Στο Κεφάλαιο 5 θα περιγράψουμε με λεπτομέρεια την αρχιτεκτονική του συστήματος και τον τρόπο λειτουργίας του καθώς επίσης και τα έξυπνα συμβόλαια, που αναπτύχθηκαν για την καταγραφή των δεδομένων των ασθενών και τη σύναψη συμβολαίων με τις ασφαλιστικές εταιρίες υγείας. Οι τεχνικές λεπτομέρειες υλοποίησης του συστήματος βρίσκονται στην τελευταία υποενότητα του κεφαλαίου αυτού, συμπεριλαμβανομένων των εργαλείων που χρησιμοποιήσαμε για την ανάπτυξη του συστήματος και την επικοινωνία του χρήστη με αυτό.

Στο Κεφάλαιο 6 θα περιγράψουμε τα σχετικά συστήματα που έχουν αναπτυχθεί στον χώρο της υγείας, χρησιμοποιώντας τις τεχνολογίες του Blockchain και χαίρουν ιδιαίτερης προσοχής (έχουν λάβει σημαντικό αριθμό αναφορών) από την επιστημονική κοινότητα. Επίσης, θα περιγράψουμε τους κινδύνους που εγκυμονεί η χρήση του Blockchain καθώς και τις αδυναμίες των Smart Contracts. Στο τέλος του κεφαλαίου αυτού υπάρχει μια συζήτηση για τον τρόπο λειτουργίας των παραπάνω συστημάτων και ειδικότερα της πλατφόρμας στην οποία έχουν τοποθετηθεί καθώς επίσης και μια μικρή εισαγωγή στην πλατφόρμα του Hyperledger. Τέλος, στο Κεφάλαιο 7 θα συνοψίσουμε τα κύρια σημεία της εργασίας αυτής.

Για την καλύτερη κατανόηση του περιεχομένου των επόμενων κεφαλαίων προτρέπουμε τους αναγνώστες, πριν προχωρήσουν στις επόμενες ενότητες, να διαβάσουν τα κεφάλαια 8.1 και 8.2 του παραρτήματος του εγγράφου αυτού, όπου περιγράφουμε συνοπτικά τις βασικές αρχές της κρυπτογραφίας δημόσιου κλειδιού (public key encryption) [9] καθώς και τον τρόπο λειτουργίας των ομότιμων (peer-to-peer)

συστημάτων [10]. Επίσης προτρέπουμε τους αναγνώστες να δουν το έγγραφο [11], όπου περιγράφεται ένας πρακτικός αλγόριθμος επίτευξης συναίνεσης μεταξύ αναξιόπιστων κόμβων ενός δικτύου, γνωστός ως Byzantine Fault Tolerance (BFT) Algorithm. Ο Αλγόριθμος αυτός επιτρέπει στα συστήματα να συνεχίσουν την ομαλή λειτουργία τους παρά την κακεντρεχή λειτουργία ορισμένων κόμβων του δικτύου (nodes act maliciously) ή την αδυναμία της μεταξύ τους επικοινωνίας (network problems) και το όνομά του προέρχεται από το γνωστό πρόβλημα επίτευξης συμφωνίας μεταξύ διαφορετικών στρατηγών, που μπορούν να επικοινωνούν μεταξύ τους με μηνύματα, γνωστό ως Byzantine Generals Problem [12].

Η σελίδα αυτή είναι σκόπιμα λευκή

2 Bitcoin & Blockchain

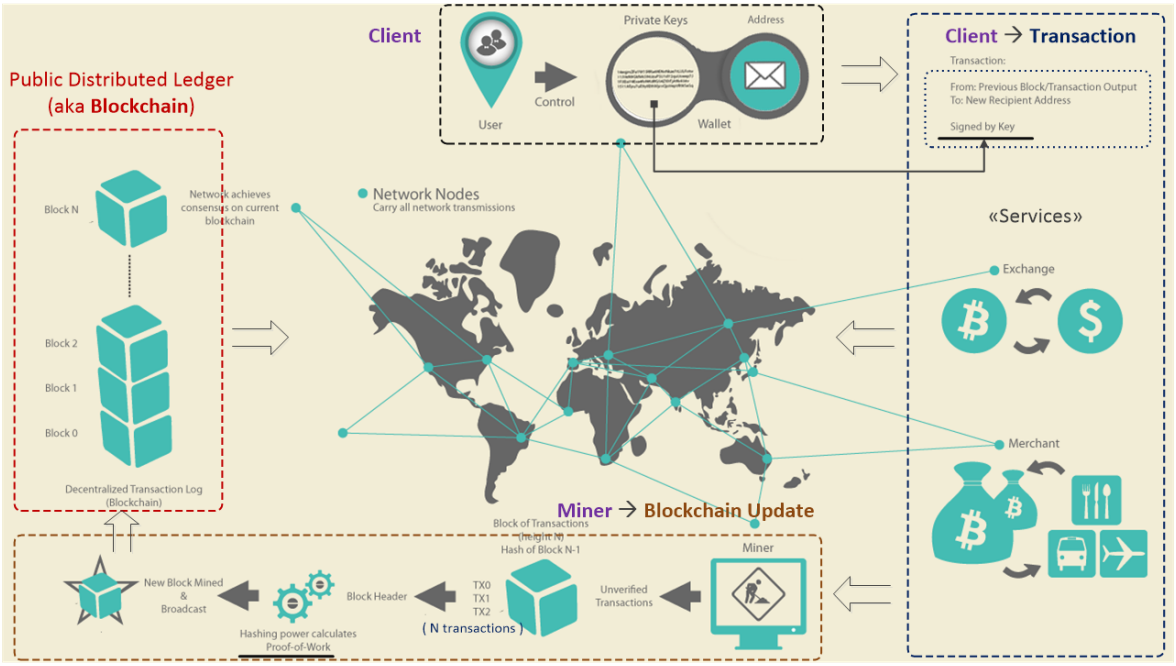
2.1 Εισαγωγή

Με τον όρο κρυπτονόμισμα αναφερόμαστε στην ψηφιακή μορφή ενός νομίσματος, το οποίο αποθηκεύεται σε ένα κατανεμημένο ηλεκτρονικό βιβλίο (distributed ledger), που περιέχει τις συναλλαγές που έχουν πραγματοποιηθεί μεταξύ των χρηστών, χρησιμοποιώντας αλγορίθμους κρυπτογράφησης και πρωτόκολλα συναίνεσης. Το πιο γνωστό κρυπτονόμισμα είναι το Bitcoin (BTC). Το κρυπτονόμισμα αυτό επιτρέπει σε μη έμπιστους χρήστες να πραγματοποιήσουν τις οικονομικές τους συναλλαγές χωρίς την ανάγκη ύπαρξης μιας κεντρικής αρχής ελέγχου. Βασίζεται στην ύπαρξη ενός αποκεντροποιημένου συστήματος επικοινωνίας μεταξύ των ομότιμων κόμβων του δικτύου [10], καθένας απ' τους οποίους διατηρεί μια αλυσίδα (γνωστή ως Blockchain) με τις συναλλαγές που έχουν λάβει χώρα. Για την πιστοποίηση της κατοχής των νομισμάτων των χρηστών και την μεταβίβαση της ιδιοκτησίας τους (ή μέρος αυτών) σε άλλους χρήστες του δικτύου γίνεται χρήση αλγορίθμων κρυπτογραφίας δημόσιου κλειδιού [3], η οποία βασίζεται στην ύπαρξη δύο διαφορετικών κλειδιών, ενός δημόσιου και ενός ιδιωτικού κλειδιού. Η εμπιστοσύνη του χρήστη προς το σύστημα αυτό πηγάζει από την χρήση έξυπνων αλγορίθμων συναίνεσης (consensus algorithms/models), οι οποίοι είναι υπεύθυνοι για τη διατήρηση της υπάρχουσας κατάστασης της αλυσίδας και των συναλλαγών που αυτή περιέχει και την ενημέρωσή της με νέους κόμβους/συναλλαγές. Περισσότερα για τα δεδομένα που αποθηκεύονται στους κόμβους της αλυσίδας και τους αλγορίθμους που χρησιμοποιούνται υπάρχουν στα επόμενα κεφάλαια της ενότητας αυτής.

2.2 Βασικές Οντότητες του κρυπτονομίσματος Bitcoin

Στο Σχήμα 1 παρουσιάζονται οι βασικές οντότητες του κρυπτονομίσματος Bitcoin και η σχέση που υπάρχει μεταξύ τους. Ο *χρήστης (user/client)* ο οποίος έχει στην κατοχή του το ιδιωτικό του κλειδί (private key), που «ξεκλειδώνει» τα χρήματα τα οποία έχουν καταχωρηθεί στην αντίστοιχη διεύθυνση (address), μπορεί να τα χρησιμοποιήσει, για να πραγματοποιήσει τις ηλεκτρονικές τους συναλλαγές (*transactions*), όπως για παράδειγμα

τη μεταφορά χρημάτων σε έναν άλλον λογαριασμό ή την αγορά ενός προϊόντος. Οι συναλλαγές που πραγματοποιούνται μεταδίδονται και στους υπόλοιπους κόμβους του δικτύου. Ωστόσο, οι συναλλαγές αυτές θεωρούνται έγκυρες μόνο, όταν προστεθούν στο ελευθέρως προσβάσιμο διαμοιραζόμενο καθολικό (public distributed ledger), γνωστό και ως αλυσίδα με μπλοκ (*Blockchain*). Η προσθήκη μιας συναλλαγής στο Blockchain γίνεται από ειδικούς κόμβους, που ονομάζονται *κόμβοι εξόρυξης (miners)* και οι οποίοι ελέγχουν περιοδικά τις δοσοληψίες που έχουν γίνει από διάφορους χρήστες, αλλά δεν έχουν ακόμη επισημοποιηθεί και ακολούθως τις προσθέτουν στο Blockchain, αφού πρώτα καταφέρουν να βρουν τη λύση σε ένα δύσκολο πρόβλημα. Η διαδικασία αυτή είναι γνωστή ως απόδειξη εργασίας (*proof of work – περισσότερα στην ενότητα 2.4.2*) και μας διασφαλίζει ότι τα δεδομένα (δοσοληψίες) που υπάρχουν στο μπλοκ και έχουν προστεθεί στην αλυσίδα δεν μπορούν εν γένει να αλλαχθούν. Ακολούθως, το νέο μπλοκ μεταδίδεται και στους υπόλοιπους κόμβους του δικτύου, οι οποίοι ελέγχουν την εγκυρότητά του και το προσθέτουν στην αλυσίδα τους.



Σχήμα 1: Οι βασικές οντότητες του κρυπτονομίσματος Bitcoin[2]

Όλοι οι κόμβοι του δικτύου είναι ίσοι. Ωστόσο, μπορεί να έχουν διαφορετικό ρόλο μέσα στο δίκτυο, ανάλογα με τις υπηρεσίες που αυτοί παρέχουν. Όπως είδαμε και στα προηγούμενα, κάποιοι κόμβοι είναι υπεύθυνοι για την ενημέρωση της αλυσίδας με νέους

[2] Βασισμένη στην εικόνα 2.1 του Βιβλίου Mastering Bitcoin του Αντρέα Μ. Αντωνόπουλου [13]

κόμβους (miners), ενώ οι κόμβοι που μας επιτρέπουν να έχουμε πρόσβαση στην αλυσίδα χρειάζεται να διαθέτουν ένα ψηφιακό πορτοφόλι (digital wallet). Οι κόμβοι αυτοί μπορεί να διατηρούν ένα αντίγραφο της αλυσίδας (Blockchain Database) και ονομάζονται Reference Nodes, ωστόσο κάτι τέτοιο δεν είναι εντελώς απαραίτητο (στην περίπτωση αυτή λέγονται Light-weight Wallet). Τέλος, όλοι οι κόμβοι έχουν τη δυνατότητα δρομολόγησης (routing) και μπορούν επίσης να ανακαλύπτουν νέους κόμβους του δικτύου, καθώς και να ελέγχουν τις συναλλαγές που καταφτάνουν σε αυτούς και να τις προωθούν στους γειτονικούς κόμβους. Σχεδόν κάθε κόμβος του δικτύου διατηρεί επίσης μια προσωρινή λίστα με τις συναλλαγές που έχουν πραγματοποιηθεί (transaction pool), οι οποίες είναι έγκυρες (ικανοποιούν μια λίστα από κριτήρια), αλλά δεν έχουν ακόμη επιβεβαιωθεί (δεν έχουν συμπεριληφθεί ακόμη στην αλυσίδα).

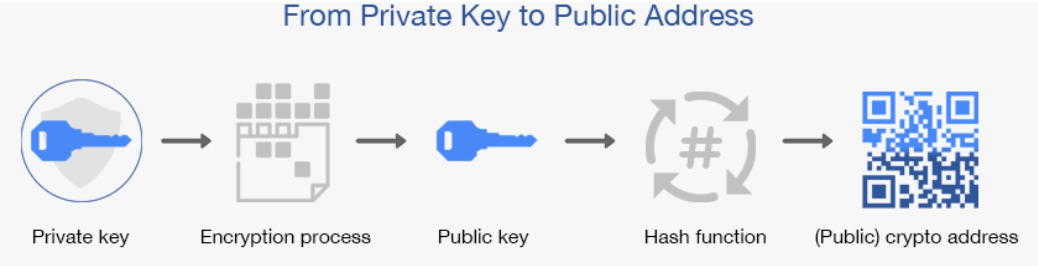
Η λειτουργία του κρυπτονομίσματος Bitcoin βασίζεται στην ύπαρξη της αλυσίδας με τα μπλοκ (Blockchain) καθώς επίσης και στους αλγορίθμους / πρωτόκολλα, που χρησιμοποιούνται για την ενημέρωση της λίστας αυτής με νέους κόμβους. Για να προστεθεί ένα *νέο μπλοκ* στην αλυσίδα, είναι απαραίτητη η εύρεση της λύσης σε ένα πολύ δύσκολο πρόβλημα. Απ' την άλλη, άπαξ και βρεθεί η λύση αυτή, όλοι οι άλλοι κόμβοι μπορούν εύκολα να ελέγξουν ότι η λύση που βρέθηκε είναι η σωστή. Ο Ανδρέας Μ. Αντωνόπουλος στο βιβλίο του [13] παρομοιάζει το πρόβλημα αυτό με τη λύση ενός πάζλ Sudoku. Όπως είναι εύκολα κατανοητό, η εύρεση της λύσης σε ένα τέτοιο πάζλ απαιτεί αρκετό χρόνο, καθώς πρέπει να γίνουν αρκετές δοκιμές, για να βρούμε τον σωστό συνδυασμό αριθμών. Ωστόσο, όταν βρεθεί η λύση, μπορούμε εύκολα να δούμε εάν αυτή είναι σωστή η όχι. Επίσης, κάθε ένα μπλοκ που προστίθεται στην αλυσίδα περιέχει ένα πεδίο που «δείχνει» ποιο είναι το προηγούμενο μπλοκ. Το αναγνωριστικό ενός μπλοκ προκύπτει με βάση τα δεδομένα που υπάρχουν σε αυτό (μπορούμε να το φανταστούμε σαν μια σύνοψη των δεδομένων που υπάρχουν σε αυτό και η οποία προσδιορίζει μοναδικά το μπλοκ). Κατά συνέπεια, οποιαδήποτε μεταβολή στα δεδομένα που υπάρχουν σε ένα από τα προηγούμενα μπλοκ συνεπάγεται αλλαγή του αναγνωριστικού του και κατά συνέπεια διάσπαση της αλυσίδας.

Για να μπορέσει κάποιος κακόβουλος χρήστης (κόμβος εξόρυξης γνώσης) να αλλάξει τα δεδομένα που υπάρχουν μέσα στην αλυσίδα, θα πρέπει να λύσει το προαναφερθέν δύσκολο πρόβλημα τόσο για το μπλοκ στο οποίο επιθυμεί να επιφέρει κάποια αλλαγή στις συναλλαγές που έχουν καταγραφεί, όσο και για τα μπλοκ που έπονται

και μάλιστα αυτό να γίνει ταυτόχρονα σε ένα σημαντικό αριθμό από κόμβους του δικτύου, ώστε να επικρατήσουν οι τροποποιήσεις που έχουν εσκεμμένα γίνει, καθώς οι υπόλοιποι κόμβοι του δικτύου γνωρίζουν ήδη την τρέχουσα κατάσταση του Blockchain και το επεκτείνουν συνεχώς με νέους κόμβους. Ωστόσο, αυτό είναι πρακτικά δύσκολο, εάν όχι αδύνατο, για αυτό και οι συναλλαγές που υπάρχουν πιο πίσω στην αλυσίδα πρακτικά δεν μπορούν να αλλάξουν. Αυτό μας διασφαλίζει ότι η πληροφορία που υπάρχει στο Blockchain και ειδικότερα τα χρήματα που ανήκουν σε κάθε χρήστη δεν μπορούν να αλλαχτούν ούτε μπορούν οι χρήστες να τα ξοδέψουν δύο φορές (double spent problem), παρά το γεγονός ότι δεν υπάρχει ένα κεντρικό σύστημα ελέγχου.

2.3 Ψηφιακά Πορτοφόλια (Digital Wallets) και Συναλλαγές (Transactions)

Για κάθε χρήστη υπάρχει μια *διεύθυνση (address)* καθώς επίσης και το αντίστοιχο *κλειδί (private key)*, το οποίο μπορούμε να χρησιμοποιήσουμε, για να πιστοποιήσουμε ότι τα χρήματα που «ανήκουν» στην παραπάνω διεύθυνση είναι δικά μας (βλέπε Κεφάλαιο 8.1.3) και ακολούθως να τα χρησιμοποιήσουμε για την πραγματοποίηση των συναλλαγών μας (π.χ., αποστολή χρημάτων σε μία άλλη διεύθυνση). Το ιδιωτικό κλειδί είναι ένας μεγάλος τυχαίος αριθμός. Το δημόσιο κλειδί προκύπτει από το ιδιωτικό κλειδί, χρησιμοποιώντας τον αλγόριθμο πολλαπλασιασμού ενός σημείου της ελλειπτικής καμπύλης [14]. Η διεύθυνση του χρήστη προκύπτει από το δημόσιο κλειδί, χρησιμοποιώντας έναν ή και παραπάνω αλγορίθμους σύνοψης, όπως φαίνεται και στο Σχήμα 2.



Σχήμα 2: Διαδικασία Παραγωγής Διεύθυνσης Χρήστη[3]

[3] https://www.bitira.com/public-key-vs-private-key-what-are-the-key-differences/

Το ιδιωτικό κλειδί αποθηκεύεται σε ένα *πορτοφόλι (digital wallet)* και συνήθως προστατεύεται επιπρόσθετα από κάποιον κωδικό. Στο σημείο αυτό αναφέρουμε ότι τα ψηφιακά πορτοφόλια περιέχουν μόνο το κλειδί, ενώ τα χρήματα που ανήκουν στον αντίστοιχο λογαριασμό υπάρχουν στην αλυσίδα (Blockchain). Επίσης, κάθε χρήστης μπορεί να έχει παραπάνω από ένα τέτοια ζευγάρια (ιδιωτικό κλειδί, διεύθυνση), τα οποία μπορούν να αποθηκεύονται στο ίδιο ψηφιακό πορτοφόλι. Αυτό είναι ιδιαίτερα χρήσιμο στις περιπτώσεις εκείνες, όπου ο χρήστης θα ήθελε να διατηρήσει την ανωνυμία του στο σύστημα, χρησιμοποιώντας παραπάνω από μία διευθύνσεις κατά την πραγματοποίηση των συναλλαγών του.

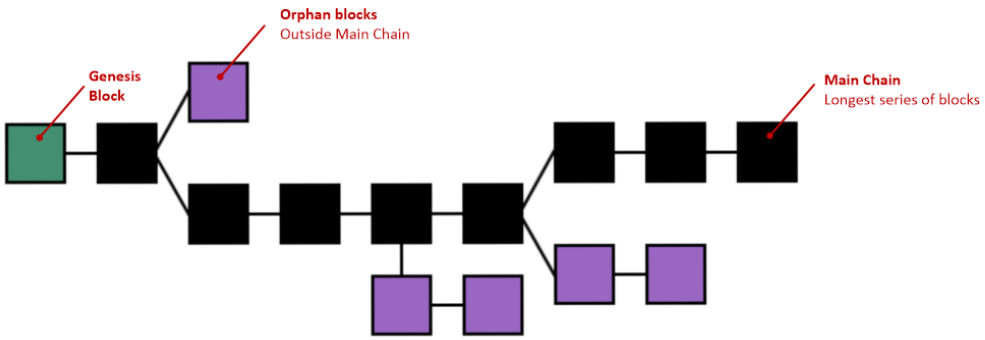
Μια *συναλλαγή (transaction)* ενημερώνει το δίκτυο ότι ένας συγκεκριμένος αριθμός από bitcoins ανήκουν σε έναν συγκεκριμένο χρήστη, ο οποίος έχει μάλιστα το δικαίωμα να μεταφέρει ένα μέρος αυτών ή και ολόκληρο το ποσό σε κάποιον άλλον χρήστη. Στο σημείο αυτό να αναφέρουμε ότι δεν αποθηκεύεται στο δίκτυο ο συνολικός αριθμός των χρημάτων που έχει κάποιος χρήστης στην κατοχή του. Αντ' αυτού καταγράφονται όλες οι συναλλαγές που έχουν λάβει χώρα, τις οποίες μπορεί ακολούθως ο χρήστης να τις συνοψίσει, για να βρει τα μη ξοδευμένα χρήματα που έχει στη διάθεσή του. Η υπηρεσία αυτή συνήθως παρέχεται από τον client (digital wallet).

Σε κάθε συναλλαγή υπάρχει η απόδειξη ότι ένα συγκεκριμένο ποσό ανήκει σε κάποιον χρήστη υπό την μορφή της ψηφιακής υπογραφής (digital signature), η οποία μπορεί να ελεγχθεί ως προς την εγκυρότητά της από οποιονδήποτε κόμβο του δικτύου, μέσω μιας προδιαγεγραμμένης διαδικασίας (unlocking script). Πολλές φορές το ποσό που απαιτείται για την πραγματοποίηση μιας συναλλαγής είναι διάσπαρτο σε παραπάνω από μια συναλλαγές που έχουν λάβει χώρα και κατά συνέπεια απαιτείται ένας ειδικός αλγόριθμος, προκειμένου να εντοπιστούν οι συναλλαγές του χρήστη που απαιτούνται για να καλύψουν ένα συγκεκριμένο ποσό. Αυτές ορίζονται ως «είσοδος» σε μια συναλλαγή (αναφορά σε προηγούμενες συναλλαγές που έχουν λάβει χώρα). Η «έξοδος» της συναλλαγής αποτελείται επίσης από ένα ή περισσότερα ποσά (π.χ., ποσό μεταφοράς και οικονομική επιβάρυνση πραγματοποίησης συναλλαγής), τα οποία είναι επίσης κλειδωμένα με τα αντίστοιχα ιδιωτικά κλειδιά, προκειμένου να είναι διαθέσιμα μόνο στους αντίστοιχους χρήστες. Αυτό γίνεται επίσης μέσω μιας προδιαγεγραμμένης διαδικασίας (locking script aka encumbrance), η οποία ορίζει τις συνθήκες που πρέπει να πληρούνται, προκειμένου ο χρήστης να μπορεί να ξοδέψει το ποσό αυτό. Η έξοδος μίας

συναλλαγής μπορεί να αποτελεί είσοδο σε μία άλλη συναλλαγή και με αυτόν τον τρόπο δημιουργείται μια αλυσίδα ιδιοκτησίας (chain of ownership), όπου ένα συγκεκριμένο ποσό μεταφέρεται από τον έναν χρήστη (για την ακρίβεια, την διεύθυνσή του) στον άλλον.

2.4 Αλυσίδα από Μπλοκ (Blockchain) και Μηχανισμοί Επέκτασης Αλυσίδας

Το Blockchain, όπως δηλώνει και το όνομά του, αποτελείται από μία αλυσίδα από μπλοκ (Σχήμα 3). Αρχικά η αλυσίδα αυτή αποτελείται από ένα μόνο μπλοκ, που ονομάζεται «genesis block» και ακολούθως ενημερώνεται (για την ακρίβεια, επεκτείνεται με την προσθήκη νέων μπλοκ) με βάση τις συναλλαγές που λαμβάνουν χώρα. Για την προσθήκη ενός νέου μπλοκ στην αλυσίδα ακολουθείται μια προδιαγεγραμμένη διαδικασία (γνωστή ως proof of work), η οποία μας διασφαλίζει ότι το μπλοκ που θα προστεθεί στην αλυσίδα, και ειδικότερα οι συναλλαγές που αυτό περιέχει είναι έγκυρες (confirmed), λαμβάνοντας υπόψη και τις άλλες συναλλαγές που έχουν καταγραφεί στα προηγούμενα μπλοκ της αλυσίδας. Στην ενότητα αυτή θα δώσουμε μια σύντομη περιγραφή της δομής ενός τέτοιου μπλοκ, καθώς επίσης και της διαδικασίας που ακολουθείται για την προσθήκη ενός νέου μπλοκ στην αλυσίδα. Για τη λεπτομερή τους περιγραφή παραπέμπουμε τους αναγνώστες στα κεφάλαια 7 και 8 του βιβλίου Mastering Bitcoin του Ανδρέα Αντωνόπουλου [13], όπου περιγράφονται αναλυτικά οι σχετικές τεχνολογίες και αλγόριθμοι.



Σχήμα 3: Γραφική απεικόνιση ενός Blockchain[4]

[4] Blockchain, https://en.wikipedia.org/wiki/Blockchain

2.4.1 Δομή ενός Μπλοκ της Αλυσίδας (Bitcoin Blockchain Data)

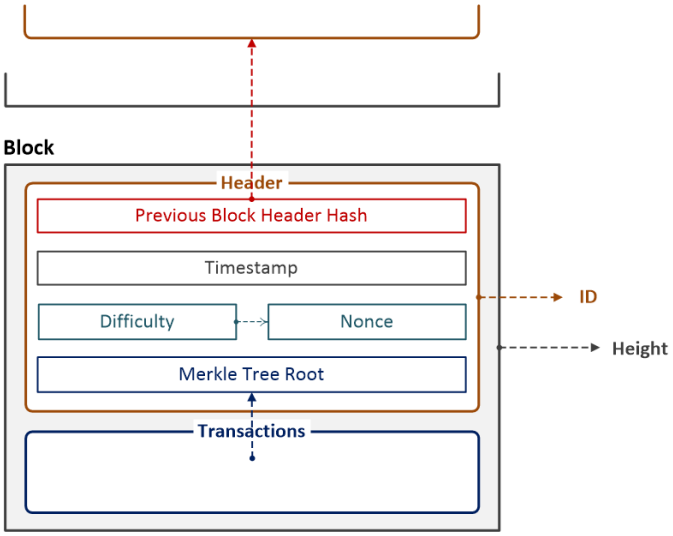
Σε κάθε μπλοκ, εκτός από το μέγεθός του και τον αριθμό των συναλλαγών που αυτό περιέχει, υπάρχει μια *επικεφαλίδα (header)* που περιέχει κάποια μεταδεδομένα (π.χ., ποιο είναι το προηγούμενο μπλοκ της αλυσίδας) καθώς επίσης και τις *συναλλαγές (transactions)* που έχουν γίνει (Σχήμα 4). Ειδικότερα, στην επικεφαλίδα υπάρχουν τα δεδομένα που παρουσιάζονται στον Πίνακας 1:

Πεδίο	**Σύντομη Περιγραφή**
Version	Η έκδοση του Blockchain (χρήσιμο για τη διαχείριση των αλλαγών)
Previous Block Hash	Η σύνοψη (hash) του προηγούμενου μπλοκ της αλυσίδας
Merkle Root	Η σύνοψη (hash) της ρίζας του Merkle Tree του μπλοκ
Timestamp	Η ημέρα και ώρα που δημιουργήθηκε το μπλοκ
Difficulty Target	Ένας αριθμός που μας δείχνει τη δυσκολία του proof of work
Nonce	Ένας αριθμός που μας δείχνει ότι έχει γίνει το proof of work

Πίνακας 1: Πεδία Επικεφαλίδας ενός μπλοκ της Bitcoin αλυσίδας

Από τα πεδία της επικεφαλίδας, αυτό που έχει ιδιαίτερη σημασία για τη δομή της αλυσίδας είναι το δεύτερο πεδίο με όνομα «previous block hash», καθώς μας επιτρέπει να εντοπίσουμε το προηγούμενο μπλοκ της αλυσίδας. Το Merkle Tree Root[5] προκύπτει από τη «συνεχή» σύνοψη των δεδομένων (π.χ., συναλλαγών), έως ότου «μείνει» ένας μόνο κόμβος, η ρίζα του δένδρου. Το δένδρο αυτό διευκολύνει την αναζήτηση των δεδομένων και ειδικότερα, εάν συμπεριλαμβάνεται αυτό που ψάχνουμε στα δεδομένα με σαφώς λιγότερους υπολογισμούς της τάξεως του λογάριθμου 2 (log2).

[5] Merkle Tree, https://en.wikipedia.org/wiki/Merkle_tree



Σχήμα 4: Γραφική απεικόνιση των πεδίων ενός μπλοκ της Bitcoin αλυσίδας

Στο σημείο αυτό είναι σημαντικό να τονίσουμε ότι το αναγνωριστικό (identifier aka ID) ενός μπλοκ είναι η συμβολοακολουθία που προκύπτει από την επικεφαλίδα, εάν εφαρμόσουμε δύο φορές διαδοχικά τον SHA256 αλγόριθμο σύνοψης και το αναγνωριστικό αυτό δεν συμπεριλαμβάνεται στο μπλοκ/επικεφαλίδα. Για κάθε μπλοκ υπάρχει επίσης ένας αριθμός (height) που μας δείχνει τη θέση που έχει το μπλοκ στην αλυσίδα (την απόσταση από τον αρχικό κόμβο), ο οποίος επίσης δεν υπάρχει στη δομή του μπλοκ. Ωστόσο, το ύψος του μπλοκ από μόνο του δεν προσδιορίζει πάντα το μπλοκ, καθώς θα μπορούσαν να υπάρχουν (για κάποιο έστω χρονικό διάστημα) παραπάνω από ένα μπλοκ που έχουν το ίδιο ύψος, όπως φαίνεται και στο Σχήμα 3 και οφείλεται στην ταυτόχρονη προσθήκη ενός μπλοκ (με διαφορετικές εν γένει συναλλαγές) στην αλυσίδα από διαφορετικούς κόμβους (όχι απαραίτητα κακόβουλους), όπως θα δούμε στο επόμενο κεφάλαιο.

Θα θέλαμε επίσης να τονίσουμε ότι το αναγνωριστικό ενός κόμβου προκύπτει από το header. Ωστόσο, σε αυτό υπάρχει πληροφορία από όλα τα δεδομένα που υπάρχουν στο μπλοκ, καθώς η επικεφαλίδα περιλαμβάνει τη ρίζα του Merkle Tree, η οποία προκύπτει, όπως είδαμε παραπάνω, από τις συναλλαγές που υπάρχουν στο μπλοκ. Κατά συνέπεια, η οποιαδήποτε αλλαγή στα περιεχόμενα ενός κόμβου επηρεάζει το αναγνωριστικό του κόμβου αυτού και κατά συνέπεια τη δομή ολόκληρης της αλυσίδας. Τα δύο τελευταία πεδία (difficult και nonce) έχουν ιδιαίτερη σημασία κατά τον έλεγχο της εγκυρότητας του περιεχομένου ενός μπλοκ και η σημασία τους θα γίνει περισσότερο κατανοητή στις επόμενες παραγράφους.

2.4.2 Προσθήκη νέου Μπλοκ – Απόδειξη Εργασίας (Proof of Work)

Μια συναλλαγή, για να είναι έγκυρη θα πρέπει να προστεθεί στην αλυσίδα. Αυτό γίνεται από ειδικούς κόμβους του δικτύου, που ονομάζονται miners. Οι κόμβοι αυτοί επιλέγουν έναν περιορισμένο αριθμό έγκυρων συναλλαγών που βρίσκονται στην λίστα τους (transactions pool) και δεν έχουν προστεθεί ακόμη στην αλυσίδα και ακολούθως κατασκευάζουν ένα μπλοκ (candidate block) με τις συναλλαγές αυτές, καθορίζοντας τα επιμέρους πεδία της επικεφαλίδας.

Στις προηγούμενες παραγράφους έχουμε ήδη δει πώς προκύπτουν οι τιμές των περισσότερων πεδίων εκτός βέβαια από δύο πεδία, difficulty και nonce. Το πρώτο πεδίο καθορίζει τη δυσκολία επίλυσης ενός μαθηματικού προβλήματος, μέσω του οποίου προκύπτει η τιμή του δεύτερου πεδίου. Ο σκοπός αυτού του πεδίου είναι να περιορίσει τον χρόνο που απαιτείται για την επίλυση του προβλήματος σε ένα ευρύτερο πλαίσιο (στην περίπτωση του Bitcoin είναι περίπου 10 λεπτά) και η τιμή του προκύπτει, λαμβάνοντας υπόψη την υπάρχουσα δυσκολία αλλά και τον χρόνο που απαιτήθηκε για την προσθήκη του τελευταίου μπλοκ της αλυσίδας.

Το πεδίο nonce περιέχει τη λύση του προβλήματος. Ειδικότερα ο κόμβος καλείται να βρει τον αριθμό εκείνον (nonce), έτσι ώστε η σύνοψη των πεδίων της επικεφαλίδας να είναι ένας αριθμός μικρότερος από αυτόν που ορίζεται στο προηγούμενο πεδίο (difficulty). Αυτό αποτελεί μια πολύ χρονοβόρα διαδικασία, καθώς το σύστημα θα πρέπει να δοκιμάσει έναν προς έναν διάφορους αριθμούς, καθώς η αλλαγή έστω και ενός ψηφίου στον αριθμό αυτό συνεπάγεται μια εντελώς διαφορετική σύνοψη επικεφαλίδας. Απ’ την άλλη μεριά, όταν βρεθεί ο αριθμός αυτός, μπορεί εύκολα να ελεγχθεί από τους άλλους κόμβους του συστήματος ότι ικανοποιεί το ζητούμενο (είναι μικρότερος από έναν δοσμένο αριθμό). Η διαδικασία εύρεσης του αριθμού nonce ονομάζεται *proof of work*.

Ο κόμβος που θα βρει πρώτος τον παραπάνω αριθμό είναι ο νικητής. Ακολούθως, προσθέτει τον κόμβο στην αλυσίδα και ενημερώνει τους υπόλοιπους κόμβους, οι οποίοι με τη σειρά τους ελέγχουν την εγκυρότητα του μπλοκ και το προσθέτουν στην αλυσίδα τους, ενώ παράλληλα βγάζουν από τη λίστα τους (transactions pool) τις συναλλαγές που έχουν ήδη προστεθεί στην αλυσίδα. Σημειώνουμε ότι ο κόμβος που θα βρει πρώτος τη λύση στο παραπάνω πρόβλημα ανταμείβεται με κάποιο ποσό. Ειδικότερα, η πρώτη από τις συναλλαγές που υπάρχουν στο μπλοκ (generation transaction) καθορίζει το ποσό της

ανταμοιβής του, το οποίο θα είναι έγκυρο, όταν βρεθεί ο ζητούμενος αριθμός και προστεθεί το μπλοκ στην αλυσίδα.

Εάν καταφέρουν ταυτόχρονα δύο κόμβοι να βρουν λύση στο πρόβλημα, δημιουργούνται δύο κλαδιά («fork»), τα οποία συνεχίζουν να επεκτείνονται ανεξάρτητα το ένα από το άλλο. Ωστόσο, ένα από αυτά «επιβιώνει» στην πορεία (για την ακρίβεια, το μακρύτερο). Ο λόγος που συμβαίνει αυτό οφείλεται στο γεγονός ότι η μεγαλύτερη αλυσίδα πιστοποιεί ότι έχει λάβει χώρα ήδη περισσότερη εργασία (proof of work) και τα δεδομένα που υπάρχουν σε αυτήν είναι πιο έγκυρα και ασφαλή (δεν μπορούν να αλλαχθούν), ειδικά αυτά που βρίσκονται πιο πίσω στην αλυσίδα. Πρακτικά, τα δεδομένα που βρίσκονται έξι μπλοκ πιο πίσω δεν μπορούν να αλλαχθούν.

Στο σημείο αυτό να αναφέρουμε ότι η λέξη *fork* (ιδιαίτερα γνωστή από τη γλώσσα προγραμματισμού C) χρησιμοποιείται τόσο, για να δηλώσει την ταυτόχρονη ύπαρξη δύο διαφορετικών versions της αλυσίδας, όσο και για να σηματοδοτήσει την πραγματοποίηση αλλαγών στην αλυσίδα για λόγους βελτίωσης ή επίλυσης κάποιου προβλήματος. Τις αλλαγές αυτές μπορούμε να τις διακρίνουμε σε δύο ευρύτερες κατηγορίες, soft και hard. Στην πρώτη περίπτωση, οι αλλαγές που έχουν γίνει δεν επηρεάζουν την εγκυρότητα των μπλοκ που υπάρχουν ήδη στην αλυσίδα (backward compatibility), ενώ στην δεύτερη περίπτωση το χαρακτηριστικό αυτό δεν ικανοποιείται και απαιτούνται σημαντικές αλλαγές τόσο στη δομή, όσο και στα προγράμματα που τρέχουν για τη συνέχιση της επέκτασης της αλυσίδας και γενικότερα τη λειτουργία του συστήματος.

3 Ethereum & Smart Contracts

3.1 Εισαγωγή

Το Ethereum δεν είναι μόνο ένα κρυπτονόμισμα. Το Ethereum είναι μια πλατφόρμα που μας επιτρέπει να αναπτύξουμε κατανεμημένες εφαρμογές (distributed applications), βασισμένες σε τεχνολογίες Blockchain. Τόσο το Ethereum, όσο και το Bitcoin βασίζονται στην ύπαρξη μιας ελευθέρως προσβάσιμης διανεμημένης αλυσίδας από μπλοκ (public distributed ledger aka Blockchain), η οποία ενημερώνεται από τους κόμβους του δικτύου με βάση τις συναλλαγές που λαμβάνουν χώρα. Ωστόσο, το Blockchain που χρησιμοποιείται στις δύο παραπάνω περιπτώσεις παρουσιάζει μικρές αλλά σημαντικές διαφορές, καθώς αυτό εξυπηρετεί διαφορετικούς σκοπούς. Το Blockchain που χρησιμοποιείται στην περίπτωση του Bitcoin έχει σχεδιαστεί, για να εξυπηρετεί την πραγματοποίηση ηλεκτρονικών οικονομικών συναλλαγών. Απ' την άλλη το Blockchain που χρησιμοποιείται στην περίπτωση του Ethereum μπορούμε να το δούμε ως μια προγραμματίσιμη υποδομή, που μας επιτρέπει να αναπτύξουμε τις εφαρμογές μας.

Για να καταλάβουμε καλύτερα τη διαφορά μεταξύ των δύο αυτών συστημάτων, αρκεί να σκεφτούμε ότι στους κόμβους, και ειδικότερα στις συναλλαγές που αυτοί περιέχουν, δεν υπάρχει μόνο πληροφορία σχετικά με το ποσό του κρυπτονομίσματος που κατέχει ο χρήστης (στην περίπτωση του Ethereum το κρυπτονόμισμα ονομάζεται Ether) αλλά και προγράμματα τα οποία είναι εκφρασμένα σε μία γλώσσα κατανοητή από την πλατφόρμα (γνωστή ως bytecode) και επομένως μπορούν να εκτελεστούν από όλους τους κόμβους του δικτύου, χρησιμοποιώντας το virtual machine της πλατφόρμας του Ethereum. Τα τμήματα αυτά κώδικα ονομάζονται *smart contracts* και είναι τμήματα "εκτελέσιμου" κώδικα (που έχει γίνει compile χρησιμοποιώντας συνήθως την γλώσσα solidity) που βρίσκεται στο Blockchain και προστατεύεται, χρησιμοποιώντας την κρυπτογραφία δημόσιου κλειδιού.

Στο σημείο αυτό να τονίσουμε ότι τα smart contracts ελέγχονται εξολοκλήρου από την πλατφόρμα του Ethereum σε αντίθεση με τους λογαριασμούς χρηστών που ελέγχονται από τον καθένα, χρησιμοποιώντας το ιδιωτικό τους κλειδί. Επίσης, να τονίσουμε ότι η

εκτέλεση του τμήματος κώδικα έχει κάποιο κόστος (όπως θα δούμε παρακάτω), το οποίο πρέπει να καταβάλει ο κάθε χρήστης για την εκτέλεση του αντίστοιχου τμήματος κώδικα.

3.2 Έξυπνα Συμβόλαια (Smart Contracts)

Ένα *συμβόλαιο (contract)* είναι μια συμφωνία μεταξύ των συμβαλλόμενων, οι οποίοι αναλαμβάνουν συγκεκριμένες δεσμεύσεις ο ένας απέναντι στον άλλο και η επιβολή τους διασφαλίζεται από τους υπάρχοντες νόμους. Σε ένα συμβόλαιο μεταξύ δύο συμβαλλόμενων οι δεσμεύσεις που υπάρχουν μεταξύ τους έχουν την εξής μορφή: εάν συμβεί αυτό, τότε ο συμβαλλόμενος υποχρεούται να κάνει ορισμένες ενέργειες, διαφορετικά θα πρέπει να κάνει κάποιες άλλες ενέργειες (if then else). Επίσης, στην περίπτωση που κάποιο από τα συμβαλλόμενα μέρη δεν είναι πρόθυμο να τηρήσει τους όρους της συμφωνίας, θα καταφύγουν στο δικαστήριο, προκειμένου να ελεγχθεί ποιος έχει δίκαιο και ακολούθως να γίνουν οι προδιαγεγραμμένες ενέργειες. Επομένως, στην περίπτωση αυτή, η επιβολή των απαραίτητων ενεργειών διασφαλίζεται από μια τρίτη οντότητα.

Από τα παραπάνω είναι εμφανής η σχέση που υπάρχει μεταξύ των συμβολαίων και των γλωσσών προγραμματισμού για την έκφραση των όρων και των ενεργειών που πρέπει να γίνουν σε μια μορφή που είναι κατανοητή από τον υπολογιστή. Επίσης, μέσω της χρήσης των τεχνολογιών Blockchain, μπορούμε να διασφαλίσουμε ότι τα συμβαλλόμενα μέρη δεν θα επιφέρουν οποιαδήποτε αλλαγή στο συμβόλαιο και μάλιστα να εξαλείψουμε την ανάγκη ύπαρξης μιας τρίτης οντότητας για την επιβολή του[6], καθώς ο

[6] Αρχικά υπήρχε η πεποίθηση ότι τα έξυπνα συμβόλαια (smart contracts) θα μπορούσαν να αντικαταστήσουν τις νομικές συμβάσεις (legal contracts). Κάποια μικροπροβλήματα που είχαν προκύψει από την εσφαλμένη λειτουργία κάποιων έξυπνων συμβολαίων θεωρήθηκαν ότι είναι εφάμιλλα με τα παραθυράκια που αφήνουν (εσκεμμένα ή μη) κάποιοι νόμοι. Ωστόσο, η παραπάνω πεποίθηση άρχισε σταδιακά να υποχωρεί μετά την πραγματοποίηση επιθέσεων στο Blockchain και ειδικότερα στην πλατφόρμα του Ethereum (DAO attach), οι οποίες είχαν ως αποτέλεσμα την κλοπή ενός σημαντικού ποσού, η οποία ανάγκασε την κοινότητα να μεταβεί σε μια προηγούμενη, κοινώς αποδεκτή κατάσταση του Blockchain, κάνοντας τις απαραίτητες αλλαγές (hard fork) τόσο στη δομή του, όσο και στους αλγορίθμους του συστήματος.

έλεγχος των συνθηκών και η επιβολή των απαραίτητων ενεργειών μπορεί να γίνει αυτόματα από το Blockchain σύστημα. Απαραίτητη βέβαια προϋπόθεση είναι το συμβόλαιο να είναι εκφρασμένο σε μία γλώσσα που να καταλαβαίνει το Blockchain σύστημα (για το Ethereum η γλώσσα είναι συνήθως η Solidity) και τα απαραίτητα για τη λειτουργία του δεδομένα να βρίσκονται στην αλυσίδα. Ένα τέτοιο συμβόλαιο, στο οποίο τόσο οι συνθήκες, όσο και οι ενέργειες είναι εκφρασμένες σε μια γλώσσα του υπολογιστή και η επιβολή του γίνεται αυτόματα από το σύστημα ονομάζενται *έξυπνο συμβόλαιο (smart contract).*

Τα έξυπνα συμβόλαια μπορούν να χρησιμοποιηθούν για τη δημιουργία μιας ευρείας γκάμας κατανεμημένων εφαρμογών (Distributed Applications aka DApps), που χρησιμοποιούν τις τεχνολογίες Blockchain, συμπεριλαμβανομένης της δημιουργίας ψηφιακών ενδείξεων (π.χ., τοπικά νομίσματα), έξυπνων ιδιοκτησιών (π.χ., αυτόματα διαχειριζόμενες συσκευές), κτλ. Ο όρος *«ψηφιακή ένδειξη» (digital token)* χρησιμοποιείται, για να αναφερθούμε είτε σε ένα ψηφιακό περιουσιακό στοιχείο (digital asset) είτε στην ψηφιακή αναπαράσταση ενός πραγματικού περιουσιακού στοιχείου (physical asset). Συνεπώς, ο όρος αυτός μπορεί να χρησιμοποιηθεί, για να αναφερθούμε σε οποιοδήποτε εμπορεύσιμο αγαθό, όπως ένα νόμισμα, πόντους που λαμβάνουμε ως ανταμοιβή, πιστοποιητικά κατοχής, κτλ. Ο όρος *«έξυπνη ιδιοκτησία» (smart property)* αναφέρεται τόσο σε ψηφιακά αντικείμενα (π.χ., πνευματικά δικαιώματα), όσο και στην ψηφιακή αναπαράσταση των αντικειμένων (π.χ., όχημα) και έχει να κάνει με τον έλεγχο της πρόσβασης σε αυτά, μέσω της χρήσης των τεχνολογιών Blockchain.

Στο σημείο αυτό να αναφέρουμε ότι η λειτουργία των smart contracts βασίζεται στα δεδομένα που υπάρχουν καταγεγραμμένα στο Blockchain. Ορισμένες, όμως, φορές είναι απαραίτητο να έχουμε πρόσβαση σε δεδομένα εκτός του Blockchain, όπως για παράδειγμα για την εύρεση της αντιστοιχίας μεταξύ ευρώ και δολαρίου. Για αυτόν τον σκοπό υπάρχουν τα oracles. Με τον όρο *oracle* αναφερόμαστε σε υπηρεσίες, που έχουν σχεδιαστεί από τρίτους, με απώτερο σκοπό τη διευκόλυνση της λειτουργίας των smart contracts. Ένα oracle μπορούμε να το φανταστούμε ως έναν data provider, ο οποίος παρέχει/ανεβάζει τα σχετικά με τη λειτουργία ενός smart contract δεδομένα στο Blockchain, προκειμένου να συνεχίσει την ομαλή λειτουργία του.

3.3 Λογαριασμοί (Accounts) και Συναλλαγές (Transactions)

Το Ethereum [15] μπορούμε να το δούμε ως μια *μηχανή κατάστασης* βασισμένη σε συναλλαγές, όπου το σύστημα ξεκινάει από μία αρχική κατάσταση (genesis state), η οποία σταδιακά μεταβάλλεται μέσα από τη χρήση των συναλλαγών, ανάλογα με τα μηνύματα που ανταλλάσσονται μεταξύ των λογαριασμών. Με τον όρο κατάσταση αναφερόμαστε σε οτιδήποτε μπορεί να αναπαρασταθεί από τον υπολογιστή, συμπεριλαμβανομένου του υπολοίπου του εκάστοτε λογαριασμού (λογαριασμοί χρηστών), συμφωνίες μεταξύ οντοτήτων του πραγματικού κόσμου ή γενικότερα οποιαδήποτε δεδομένα, τα οποία αποτελούν την ψηφιακή αναπαράσταση μιας οντότητας του πραγματικού κόσμου (και αποτελούν μέρος ενός smart contract). Συνεπώς, οι συναλλαγές αποτελούν μια έγκυρη μετάβαση από τη μία κατάσταση στην άλλη.

Στο Ethereum υπάρχουν δύο διαφορετικοί τύποι «λογαριασμών». Οι *λογαριασμοί χρηστών (aka Externally Owned Accounts),* οι οποίοι ελέγχονται εξολοκλήρου από τον εκάστοτε χρήστη (μέσω του ιδιωτικού του κλειδιού) και περιέχουν τα χρήματα που έχει αυτός στην διάθεσή του. Τα *συμβόλαια (smart contracts)* είναι ένας άλλος τύπος λογαριασμού, ο οποίος μπορεί να συσχετίζεται με κάποιο υπόλοιπο, αλλά ελέγχεται εξ ολοκλήρου από την πλατφόρμα, η οποία αναλαμβάνει να διαχειριστεί και να εκτελέσει τον κώδικα που αυτό περιέχει. Τα συμβόλαια και κυρίως ο κώδικας που αυτά περιέχουν εκτελούνται από τον χρήστη με την αποστολή του κατάλληλου μηνύματος μέσω της χρήσης των συναλλαγών.

Στο Ethereum υπάρχουν τρεις τύποι συναλλαγών. Ο πρώτος τύπος περιλαμβάνει τις *«κλασικές» συναλλαγές*, που χρησιμοποιούνται για την μεταφορά χρημάτων από έναν λογαριασμό σε έναν άλλον. Οι άλλοι δύο τύποι συναλλαγών έχουν να κάνουν με τα smart contracts. Ειδικότερα, ο δεύτερος τύπος περιλαμβάνει τις συναλλαγές για τη *δημιουργία/αρχικοποίηση* των smart contracts, ενώ ο τρίτος τις συναλλαγές για την *εκτέλεση* μιας συνάρτησης/μεθόδου (δηλαδή, ενός τμήματος κώδικα) του συμβολαίου.

Για κάθε μία από τις παραπάνω συναλλαγές καταγράφουμε τα πεδία που φαίνονται στον Πίνακας 2.

Πεδίο	**Σύντομη Περιγραφή**
From	Διεύθυνση Αποστολέα (υπογεγραμμένη)
To	Διεύθυνση Παραλήπτη
Value	Το ποσό μεταφοράς, εκφρασμένο σε Wei (1 Ether = 10^{18} Wei)
Gas	Ένας αριθμός που εκφράζει το ποσό – «καύσιμο» – που μπορεί να χρησιμοποιηθεί για την εκτέλεση μιας συναλλαγής (σε συνδυασμό με το Gas Price)
Gas Price	Η τιμή του «καυσίμου» (στο κρυπτονόμισμα Ether) για τη συγκεκριμένη συναλλαγή
Data	Δεδομένα που χρειάζονται κατά την εκτέλεση μιας συναλλαγής - μηνύματος
Nonce	Ένας αριθμός

Πίνακας 2: Πεδία μιας Ethereum Συναλλαγής

Οι τιμές των παραπάνω πεδίων εξαρτώνται από τον τύπο της συναλλαγής. Στην περίπτωση μιας απλής «κλασικής» συναλλαγής για την αποστολή χρημάτων από έναν χρήστη σε έναν άλλον, το πεδίο Data είναι κενό. Στην περίπτωση δημιουργίας ενός συμβολαίου, το πεδίο του παραλήπτη είναι κενό, ενώ το πεδίο με τα δεδομένα περιέχει τις τιμές αρχικοποίησης του συμβολαίου. Το πεδίο αυτό μπορεί, επίσης, να περιέχει τα δεδομένα που θέλουμε να στείλουμε για την εκτέλεση ενός συγκεκριμένου τμήματος κώδικα (δηλαδή μία συνάρτηση) του συμβολαίου. Τα πεδία Gas και Gas Price έχουν να κάνουν με τα χρήματα που προσφέρουμε για την πραγματοποίηση της συναλλαγής μας (όπως θα δούμε παρακάτω) και η τιμή τους επηρεάζει τόσο την εκτέλεση του κώδικα (στην περίπτωση των smart contracts), όσο και τη διαδικασία πιστοποίησης των συναλλαγών, καθώς, όσα περισσότερα χρήματα προσφέρουμε, τόσο οι κόμβοι εξόρυξης γνώσης θα είναι περισσότερο πρόθυμοι να το συμπεριλάβουν στο αμέσως επόμενο μπλοκ.

Ένα smart contract μπορεί να στέλνει μηνύματα σε ένα άλλο smart contract. Τα δεδομένα αυτά τοποθετούνται στο πεδίο data της συναλλαγής και ακολούθως χρησιμοποιούνται από την πλατφόρμα του Ethereum, για τη δημιουργία του μηνύματος. Συνεπώς, το *μήνυμα (message)* είναι ένα εικονικό αντικείμενο που δημιουργείται κατά τη διαδικασία εκτέλεσης του τμήματος κώδικα του συμβολαίου, με βάση τα δεδομένα που υπάρχουν στη συναλλαγή.

3.4 Κώδικας Έξυπνου Συμβολαίου (Smart Contract) και Κόστος Εκτέλεσης

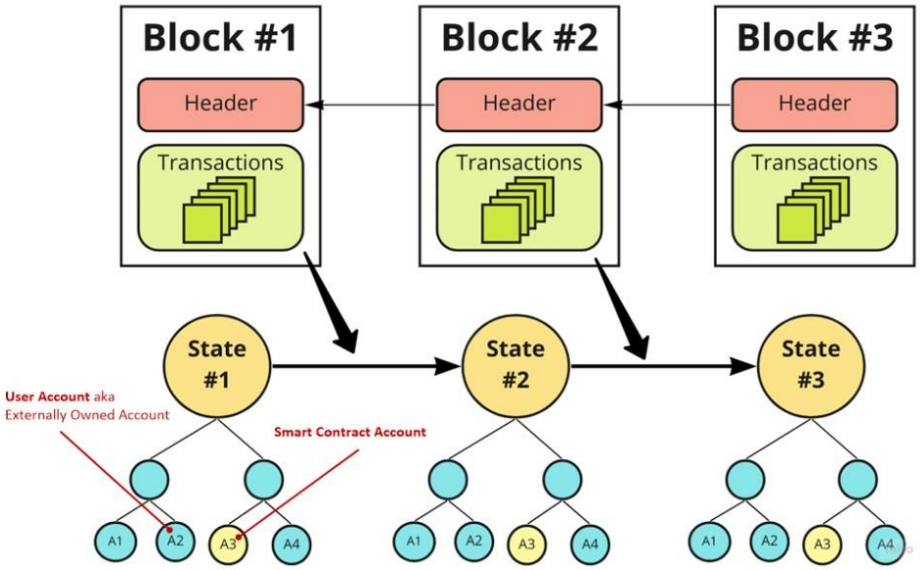
Ο κώδικας του έξυπνου συμβολαίου είναι μία *ακολουθία από bytes (Bytecode),* που αναπαριστά λειτουργίες που μπορούν να εκτελεστούν από την εικονική μηχανή του Ethereum (Ethereum Virtual Machine aka EVM). Ο κώδικας αυτός προκύπτει κατά τη διαδικασία μεταγλώττισης (compile) ενός Smart Contract, που είναι γραμμένο χρησιμοποιώντας τη *γλώσσα προγραμματισμού Solidity* και προηγείται της δημιουργίας του συμβολαίου (περισσότερες πληροφορίες υπάρχουν στο Παράρτημα, στο Κεφάλαιο 8.3). Κατά τη διάρκεια της εκτέλεσής του μπορεί να έχει πρόσβαση σε τέσσερις διαφορετικούς τύπους αποθήκευσης δεδομένων, συμπεριλαμβανομένης μιας στοίβας (stack - περιορισμένου μεγέθους), μιας προσωρινής μνήμης (memory – άπειρου μεγέθους) καθώς επίσης και της μνήμης αποθήκευσης (storage – key/value), οι τιμές της οποίας διατηρούνται και μετά την ολοκλήρωση της εκτέλεσης του προγράμματος.

Το συνολικό ποσό που απαιτείται (για την ακρίβεια, μπορεί να διατεθεί) για την εκτέλεση μιας συναλλαγής προκύπτει από τον πολλαπλασιασμό της τιμής του πεδίου *gas (limit)* με την τιμή του πεδίου *gas price*. Ωστόσο, εάν το κόστος εκτέλεσης δεν ξεπεράσει το ποσό αυτό, το υπόλοιπο επιστρέφεται στον χρήστη. Το κόστος εκτέλεσης του τμήματος του κώδικα κάθε συμβολαίου προκύπτει από τις εντολές που αυτό περιέχει. Με αυτόν τον τρόπο υπάρχει δικαιοσύνη ως προς τους υπολογιστικούς πόρους που αφιερώνονται για την εκτέλεση των επιμέρους εντολών των συναρτήσεων των smart contracts, έτσι ώστε πιο περίπλοκα συμβόλαια να απαιτούν παραπάνω χρήματα για την εκτέλεσή τους. Σημειώνουμε ότι υπάρχουν μεγάλες διαφορές στο κόστος των επιτρεπόμενων εντολών. Για παράδειγμα μια εντολή πρόσθεσης δύο αριθμών κοστίζει 3 gas, ενώ μια εντολή που απαιτεί πρόσβαση σε έναν άλλον λογαριασμό 25000 gas. Για περισσότερες λεπτομέρειες παραπέμπουμε τους αναγνώστες στον πίνακα 5-3 του βιβλίου Blockchain [16]. Εάν το ποσό καυσίμου που έχουμε ορίσει δεν είναι αρκετό, η εκτέλεση του κώδικα σταματάει και το σύστημα επιστρέφει στην προηγούμενη κατάσταση. Ωστόσο, το ποσό που αντιστοιχεί στην εκτέλεση του κώδικα αποδίδεται στον κόμβο που το έχει τρέξει. Η προσέγγιση αυτή προστατεύει το σύστημα από επιθέσεις άρνησης υπηρεσίας (Denial of Service attacks, aka DoS attacks), οι οποίες σταματούν, όταν εξαντληθούν τα διαθέσιμα καύσιμα.

3.5 Δεδομένα Αλυσίδας και Μηχανισμοί Επέκτασης

3.5.1 Δομή Μπλοκ Αλυσίδας (Ethereum Blockchain Data)

Κάθε μπλοκ της αλυσίδας περιλαμβάνει τα πεδία που είχαμε δει στην προηγούμενη ενότητα στο κεφάλαιο 2.4.1, όπου περιγράψαμε αναλυτικά τα πεδία που υπάρχουν σε ένα μπλοκ του Bitcoin Blockchain. Ωστόσο, περιλαμβάνει ορισμένα επιπρόσθετα πεδία, όπως τη θέση (number) που κατέχει κάθε κόμβος στην αλυσίδα καθώς επίσης και τον αριθμό gas, που έχουν ξοδευτεί για τις συναλλαγές που υπάρχουν στο block. Επίσης, στα blocks υπάρχει πληροφορία όχι μόνο για τις συναλλαγές που έχουν πραγματοποιηθεί αλλά και πληροφορία για την πιο πρόσφατη κατάσταση στην οποία βρίσκεται το σύστημα.



Σχήμα 5: Η δομή του Ethereum Blockchain[7]

Η κατάσταση στην οποία βρίσκεται το σύστημα αποθηκεύεται σε μία δενδρική μορφή και κατά την προσθήκη ενός νέου μπλοκ αναφέρουμε μόνο τη ρίζα του δέντρου αυτού στο μπλοκ (Σχήμα 5). Η νέα κατάσταση προκύπτει από την εκτέλεση των συναλλαγών που υπάρχουν στο block.

[7] https://kauri.io/manage-an-ethereum-account-with-java-and-web3j/925d923e12c543da9a0a3e617be963b4/a

3.5.2 Προσθήκη νέου Μπλοκ – Απόδειξη Πονταρίσματος (Proof of Stake)

Για την ενημέρωση της αλυσίδας με νέους κόμβους, το Ethereum (όπως και το Bitcoin) βασίζεται στον *Proof of Work* αλγόριθμο. Η δυσκολία επίλυσης του προβλήματος ορίζεται με τέτοιο τρόπο, ώστε να είναι εφικτή η δημιουργία ενός νέου μπλοκ κάθε περίπου 12 δευτερόλεπτα.

Η προσέγγιση αυτή (Proof of Work) απαιτεί την κατανάλωση σημαντικής ποσότητας ενέργειας, που ξοδεύεται «άσκοπα» από τους κόμβους του δικτύου (miners), υπό την έννοια ότι συμμετέχουν αρκετοί κόμβοι στη διαδικασία, αλλά μόνο η «εργασία» του κόμβου που θα καταφέρει να λύσει πρώτος το πρόβλημα θα ληφθεί πραγματικά υπόψη, ενώ οι υπόλοιποι κόμβοι θα εγκαταλείψουν εν τέλει αυτήν την προσπάθεια. Επίσης, αν και ο αλγόριθμος αυτός επιτρέπει στον κάθε χρήστη να μπορεί να συμμετέχει στη διαδικασία αυτή, η υψηλή υπολογιστική ισχύς, που απαιτείται για τη δημιουργία νέων κόμβων, οδήγησε στη δημιουργία ομάδων από τέτοιους κόμβους (mining pools), που συνεργάζονται στην προσπάθειά τους να λύσουν το πρόβλημα, μοιράζοντας μεταξύ τους τόσο την δουλειά, όσο και τα κέρδη (σε περίπτωση επιτυχίας) και κατ' επέκταση έχουν μεγαλύτερη πιθανότητα να καταφέρουν να λύσουν το πρόβλημα εγκαίρως. Το γεγονός αυτό περιορίζει κάπως τις οντότητες (mining pools) που είναι σε θέση να δημιουργήσουν έναν νέο κόμβο στην αλυσίδα, γεγονός που αυξάνει τους κινδύνους του συστήματος, καθώς η πιθανή συνεργασία ορισμένων εξ αυτών θα μπορούσε να κατέχει παραπάνω από το 51% της υπολογιστικής ισχύος του δικτύου και κατ' επέκταση να επιφέρουν ανεπιθύμητες αλλαγές σε αυτό.

Τα παραπάνω οδήγησαν την επιστημονική κοινότητα στην αναζήτηση νέων αλγορίθμων επίτευξης συναίνεσης (consensus algorithms), έτσι ώστε να μην γίνεται σπατάλη των διαθέσιμων υπολογιστικών πόρων. Ο πιο ευρέως γνωστός αλγόριθμος ονομάζεται *Proof of Stake* και βασίζεται στην ιδέα ότι όσο μεγαλύτερο μερίδιο (stake) κατέχει ένας χρήστης στο δίκτυο (για την ακρίβεια, έχει ποντάρει στο σύστημα το οποίο θα πάρει πίσω μαζί με κάποια ανταμοιβή για τη δημιουργία του νέου μπλοκ, εφόσον όλα εξελιχθούν ομαλά), τόσο λιγότερο θα θέλει να το βλάψει, καθώς θα χάσει το ποσό που κατέχει [17]. Στην προσέγγιση αυτή οι κόμβοι του δικτύου δεν αναλώνονται στην επίλυση δύσκολων προβλημάτων αλλά στον έλεγχο (validation) της εγκυρότητας των συναλλαγών

που θα συμπεριληφθούν σε κάθε μπλοκ. Η επιλογή του κόμβου που θα προσθέσει τον νέο κόμβο στην αλυσίδα καθορίζεται, λαμβάνοντας υπόψη την ποσότητα των κρυπτονομισμάτων που έχει ποντάρει ο κάθε χρήστης στο δίκτυο, έτσι ώστε οι χρήστες με περισσότερα χρήματα να είναι πιο πιθανόν να επιλεγούν. Η ανταμοιβή, που προσφέρεται στον κόμβο που πραγματοποιεί τους απαραίτητους ελέγχους και προσθέτει τον νέο κόμβο της αλυσίδας, προέρχεται συνήθως από το άθροισμα των κρυπτονομισμάτων που προσφέρονται (για την εκτέλεσή τους) από κάθε μία από τις συναλλαγές αυτές.

Υπάρχουν διάφορες παραλλαγές της προσέγγισης αυτής. Για παράδειγμα στη δημιουργία του νέου κόμβου της αλυσίδας μπορεί να συμμετέχουν αρκετοί χρήστες, οι οποίοι κατέχουν κάποιο σημαντικό μερίδιο του δικτύου, κατασκευάζοντας αρχικά το «δικό τους» υποψήφιο μπλοκ και ακολούθως ψηφίζοντας μεταξύ τους για το ποιο μπλοκ θα πρέπει να προστεθεί, έως ότου υπάρξει συναίνεση (byzantine fault tolerance proof of stake). Σε μια άλλη προσέγγιση λαμβάνεται υπόψη η ηλικία των χρημάτων που κατέχει ο κάθε χρήστης κατά την επιλογή του κόμβου που θα είναι υπεύθυνος για την προσθήκη του νέου κόμβου στην αλυσίδα (coin age proof of stake). Η προσέγγιση αυτή επιτρέπει μεν στους χρήστες με τα περισσότερα χρήματα να δημιουργούν πιο συχνά κόμβους, ωστόσο δεν τους αφήνει να κυριαρχήσουν έναντι των υπολοίπων κόμβων.

Όπως είναι εμφανές από τα παραπάνω, οι αλγόριθμοι αυτοί επιτρέπουν την επέκταση της αλυσίδας, χωρίς να είναι απαραίτητο να ξοδέψουμε τεράστιες ποσότητες ενέργειας. Ωστόσο, αξίζει να αναφέρουμε ότι ορισμένοι από τους αλγορίθμους αυτούς μπορεί να μην επιτρέψουν τον αποτελεσματικό χειρισμό μιας σύγκρουσης (conflict), η οποία έχει ως αποτέλεσμα τη δημιουργία δύο Blockchain αλυσίδων (fork). Ειδικότερα, είναι πιθανόν να συνεχίζουν την επέκταση των δύο αλυσίδων, έχοντας ως απώτερο σκοπό να μεγιστοποιήσουν το κέρδος τους, καθώς δεν έχουν τίποτα να χάσουν. Το πρόβλημα αυτό είναι γνωστό ως πρόβλημα στο οποίο δεν διακυβεύεται να χάσουν τίποτα, «nothing at stake». Επίσης, εάν ένας χρήστης καταφέρει να συγκεντρώσει παραπάνω από το 51% των κρυπτονομισμάτων που είναι διαθέσιμα στο δίκτυο, μπορεί να επιφέρει ανεπιθύμητες αλλαγές σε αυτό, ωστόσο κάτι τέτοιο είναι πρακτικά δύσκολο.

Όπως αναφέραμε και στην αρχή της ενότητας αυτής, το Ethereum έχει βασιστεί στην Proof of Work προσέγγιση για την ανανέωση της αλυσίδας. Ωστόσο, δεδομένων των θεμάτων που αναφέρθηκαν πιο πάνω, έχουν σκοπό να αντικαταστήσουν τον αλγόριθμό αυτό με κάποιον άλλον, ο οποίος είναι βασισμένος στον αλγόριθμο Proof of Stake.

Ειδικότερα, ο αλγόριθμος συναίνεσης που έχουν αναπτύξει ονομάζεται Casper[8] και η αλλαγή είναι προγραμματισμένη να γίνει μέσα στο 2020. Ο αλγόριθμος αυτός ξεπερνά το παραπάνω πρόβλημα που παρατηρήθηκε στην περίπτωση κάποιας σύγκρουσης, επιτρέποντας σε κάποιον validator να συμμετέχει σε μια από τις διαθέσιμες αλυσίδες.

[8] Ethereum Casper Explained, https://academy.binance.com/blockchain/ethereum-casper-explained

4 Σύστημα Σύναψης Συμβολαίων Υγείας

4.1 Περιγραφή Συστήματος

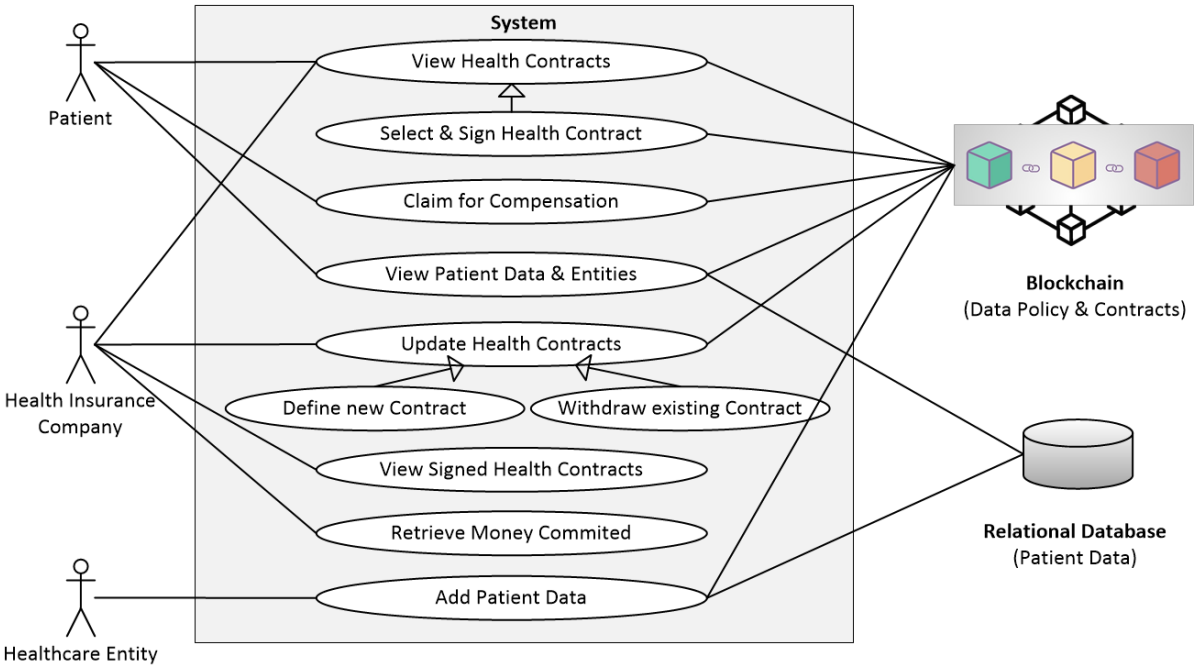
Στα πλαίσια της εργασίας αυτής δημιουργήσαμε μια κατανεμημένη εφαρμογή (DApp), η οποία βασίζεται στις τεχνολογίες του Blockchain και επιτρέπει στους χρήστες να συνάψουν μια συμφωνία με μια ασφαλιστική εταιρία, καταβάλλοντας ένα συγκεκριμένο χρηματικό ποσό. Η εταιρία, απ' τη μεριά της, είναι υπεύθυνη να αποζημιώσει τον ασθενή σε περίπτωση παρουσίασης κάποιου ιατρικού προβλήματος, με την προϋπόθεση ότι έχουν γίνει όλες οι απαιτούμενες ενέργειες από τη μεριά του ασθενή. Για τον σκοπό αυτό, τα δεδομένα του ασθενή, που προκύπτουν κατά τη διάρκεια της εξέτασής του από κάποιο κέντρο περίθαλψης και είναι απαραίτητα για την αποτίμηση των όρων του συμβολαίου, θα καταγράφονται στο σύστημα αλλά η πρόσβαση σε αυτά θα ελέγχεται εξ ολοκλήρου από τον χρήστη.

Για τις ανάγκες της εργασίας αυτής δημιουργήσαμε ένα γραφικό περιβάλλον που επιτρέπει στους ασθενείς να εξετάσουν τα διαθέσιμα συμβόλαια και ακολούθως να επιλέξουν/υπογράψουν το συμβόλαιο που τους καλύπτει (signed contract). Συνεπώς, κατά τη διάρκεια σύναψης της συμφωνίας θα πρέπει να καθορίσουμε σε ποιον ασθενή αναφερόμαστε, τη χρονική περίοδο που ισχύει η συμφωνία (start & end date), τα ιατρικά προβλήματα που καλύπτει (π.χ., εκδήλωση μιας ασθένειας), τις συνθήκες τις οποίες θα πρέπει να ικανοποιεί ο ασθενής, για να λάβει την αποζημίωση (π.χ., λήψη μιας συγκεκριμένης φαρμακευτικής ουσίας κατά τη διάρκεια μιας κλινικής μελέτης) και φυσικά το ποσό που θα λάβει στην περίπτωση αυτή. Για λόγους ασφάλειας, τα παραπάνω δεδομένα θα αποθηκεύονται στην πλατφόρμα του Blockchain και ο χρήστης θα επικοινωνεί απευθείας με αυτή, μέσω της χρήσης των έξυπνων συμβολαίων που υλοποιήθηκαν. Σημειώνουμε ότι ο ασθενής θα δικαιούται να λάβει μία μόνο φορά την αποζημίωση κατά την περίοδο ισχύος του συμβολαίου. Επίσης η εταιρία θα έχει την δυνατότητα να ακυρώσει κάποιο συμβόλαιο [18], καταβάλλοντας στο λογαριασμό του χρήστη το αντίστοιχο ποσό/ποινή (π.χ., το διπλάσιο ποσό από αυτό που έδωσε κατά τη

σύναψη της συμφωνίας, εφόσον δεν πληρούνται οι όροι του υπογεγραμμένου συμβολαίου, διαφορετικά όλο το ποσό που προέβλεπε να λάβει).

4.1.1 Παρεχόμενες Υπηρεσίες

Στις επόμενες τρεις παραγράφους μπορούμε να δούμε συνοπτικά τις υπηρεσίες που παρέχονται από το σύστημα που αναπτύχθηκε, για καθεμία από τις τρεις βασικές οντότητες του συστήματος. Οι υπηρεσίες αυτές παρουσιάζονται γραφικά και στο ακόλουθο UML[9] διάγραμμα χρήσης (Σχήμα 6).



Σχήμα 6: Διάγραμμα Χρήσης του Συστήματος

Μέσω του συστήματος *οι ασθενείς (patients)* έχουν τη δυνατότητα να εξετάσουν τα διαθέσιμα συμβόλαια μιας ασφαλιστικής εταιρίας υγείας, τα προβλήματα που καλύπτουν και τις προϋποθέσεις που πρέπει να ισχύουν για την αποζημίωσή τους, το κόστος της ασφάλειας και φυσικά το ποσό που θα λάβουν σε περίπτωση που ικανοποιούνται οι όροι του συμβολαίου. Επίσης, οι ασθενείς μπορούν να συνάψουν ένα νέο συμβόλαιο με την ασφαλιστική εταιρία Υγείας, καταβάλλοντας το απαιτούμενο ποσό. Επιπρόσθετα, οι ασθενείς μπορούν να αποζημιωθούν άμεσα με ένα συγκεκριμένο ποσό, το οποίο θα κατατίθεται στο λογαριασμό τους, εφόσον πληρούνται οι όροι του συμβολαίου. Τέλος οι ασθενείς μπορούν να εξετάσουν τα ιατρικά τους δεδομένα που

[9] Unified Modeling Language (UML) , https://en.wikipedia.org/wiki/Unified_Modeling_Language

αποθηκεύονται στο σύστημα καθώς επίσης και τις οντότητες που έχουν πρόσβαση σε αυτά.

Οι ασφαλιστικές εταιρίες υγείας (health insurance companies) έχουν τη δυνατότητα να εκφράσουν τους όρους ενός συμβολαίου και να το «ανεβάσουν» στην πλατφόρμα του συστήματος (ώστε να μπορεί ακολούθως να επιλεγεί από τους ανθρώπους). Επίσης, μπορούν να καταργήσουν ένα υπάρχον συμβόλαιο (εάν αυτό είναι εντελώς απαραίτητο), αποζημιώνοντας όλους τους πελάτες που το είχαν επιλέξει. Επιπλέον, μπορούν να δουν τα συμβόλαια που έχουν ήδη υπογραφεί και να ζητήσουν τη μεταφορά του ποσού της εταιρίας που είχε δεσμευτεί (για τις ανάγκες του συμβολαίου) στον λογαριασμό της.

Τα κέντρα περίθαλψης (healthcare entities/providers) είναι υπεύθυνα για την ενημέρωση του συστήματος με νέα δεδομένα που προκύπτουν από την επίσκεψη ενός ασθενή σε αυτά.

Στα πλαίσια της εργασίας αυτής θα θεωρήσουμε ότι υπάρχει μόνο μία ασφαλιστική εταιρία υγείας και ένα μόνο κέντρο περίθαλψης. Ωστόσο, η προσέγγιση που περιγράφουμε θα μπορούσε να χρησιμοποιηθεί, για παραπάνω από μία εταιρίες και παραπάνω από ένα κέντρα περίθαλψης, με μικρές αλλαγές. Επίσης, τα δεδομένα που αποθηκεύονται στο σύστημα θεωρούμε ότι καταγράφονται μόνο για τις ανάγκες αποτίμησης των συμβολαίων των ασφαλιστικών εταιριών. Κατά συνέπεια, πρόσβαση σε αυτά θα επιτρέπεται μόνο στις αντίστοιχες ασφαλιστικές εταιρίες και μόνο κατά τη διάρκεια ισχύος του συμβολαίου.

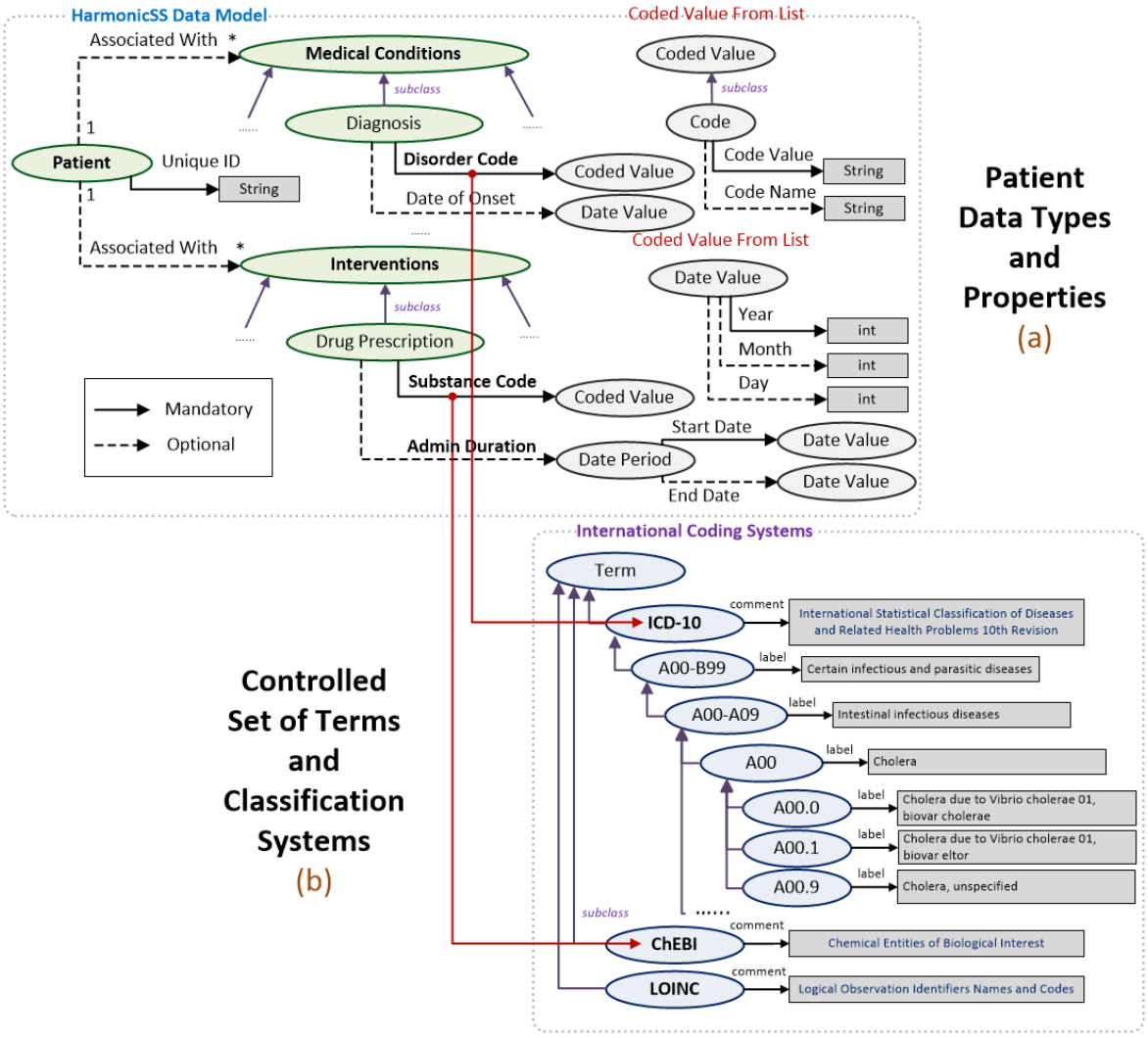
4.2 Αναπαράσταση και Αποθήκευση των Δεδομένων

4.2.1 Αναπαράσταση και Αποθήκευση των Δεδομένων των Ασθενών

Για τη διαλειτουργική αναπαράσταση τόσο των δεδομένων των ασθενών, όσο και των συνθηκών του εκάστοτε συμβολαίου, βασιστήκαμε σε υπάρχοντα *μοντέλα* και *διεθνείς κωδικοποιήσεις*. Ειδικότερα, χρησιμοποιήσαμε την οντολογική αναπαράσταση των δεδομένων των ασθενών (Σχήμα 7– (a)), που έχουμε ήδη αναπτύξει για την καταγραφή των δεδομένων των ασθενών με Σύνδρομο Σιόγκρεν στα πλαίσια του έργου HarmonicSS[10]. Σύμφωνα με το μοντέλο αυτό, ένας ασθενής μπορεί να συνδέεται με έναν

[10] HarmonicSS Project, https://www.harmonicss.eu/

ή περισσότερους τύπους δεδομένων (αρχιτεκτονική αστέρα), όπως είναι τα Δημογραφικά του Χαρακτηριστικά, οι Εργαστηριακές Εξετάσεις που έχει κάνει, τα Ιατρικά Προβλήματα που έχει παρουσιάσει και τα Φάρμακα που έχει λάβει. Για καθέναν από αυτούς τους τύπους δεδομένων έχουμε, επίσης, ορίσει τις παραμέτρους που μας ενδιαφέρουν καθώς επίσης και τις επιτρεπόμενες τιμές.



Σχήμα 7: Μοντέλο Αναφοράς (Reference Model) Δεδομένων Ασθενών

Για τη διεύρυνση του πεδίου εφαρμογών του μοντέλου αυτού, χρησιμοποιήσαμε διεθνή συστήματα κωδικοποίησης για την καταγραφή των ασθενειών, των φαρμάκων και των εργαστηριακών εξετάσεων. Ειδικότερα, χρησιμοποιήσαμε το Διεθνές Σύστημα Κωδικοποίησης των Ασθενειών (ICD[11]), τη Βάση με τις Χημικές Ουσίες Βιολογικού

[11] International Classification of Diseases (ICD), https://www.who.int/classifications/icd/en/

Ενδιαφέροντος (ChEBI[12]) καθώς επίσης και τη Βάση με τα Αναγνωριστικά Ονόματα και τους Κωδικούς των εξετάσεων (LOINC[13]). Για καθένα από τα συστήματα αυτά δημιουργήσαμε μια οντολογική αναπαράσταση των όρων του (σε περίπτωση που δεν ήταν άμεσα διαθέσιμη) και ακολούθως τα συμπεριλάβαμε στο μοντέλο μας (Σχήμα 7 – (b)), ώστε να έχουμε συγκεντρωμένη όλη την πληροφορία που απαιτείται για την έκφραση των δεδομένων των ασθενών σε μια μόνο οντολογία.

Τα *δεδομένα των ασθενών* μπορούν εύκολα να εκφραστούν, χρησιμοποιώντας τους όρους του παραπάνω μοντέλου. Για παράδειγμα, ο ασθενής Χ είναι ένας Άνθρωπος ο οποίος το 2000 διαγνώστηκε με Οξύ Έμφραγμα του Μυοκαρδίου (Acute Myocardial Infarction – ICD-10 κωδικός: I21). Σημειώνουμε ότι τόσο η οντολογία που αναπτύχθηκε (λόγω μεγέθους), όσο και τα δεδομένα των ασθενών (λόγω ευαίσθητων προσωπικών δεδομένων) αποθηκεύονται εκτός της αλυσίδας (off Blockchain). Ωστόσο, καταγράφουμε στο Blockchain δείκτες προς αυτά, έτσι ώστε να είναι εφικτό να εντοπίσουμε τα δεδομένα που ανήκουν σε έναν ασθενή, να βεβαιωθούμε ότι δεν τροποποιήθηκαν (τόσο τα δεδομένα, όσο και το μοντέλο αναφοράς) και ακολούθως να ελέγξουμε τις συνθήκες (εάν πληρούνται οι όροι του συμβολαίου).

Στα πλαίσια της εργασίας αυτής θα θεωρήσουμε ότι τα δεδομένα αποθηκεύονται σε μία *σχεσιακή βάση*, χρησιμοποιώντας τους όρους της οντολογίας. Ειδικότερα, το σύνολο των δηλώσεων που αφορούν την περιγραφή μιας συγκεκριμένης οντότητας, όπως για παράδειγμα τα δεδομένα που αφορούν μία συγκεκριμένη διάγνωση, θα καταγράφονται στο ίδιο κελί, το οποίο θα έχει ένα αναγνωριστικό, το οποίο θα χρησιμοποιούμε, για να αναφερθούμε σε αυτό. Το αναγνωριστικό αυτό θα προκύπτει από τον αλγόριθμο σύνοψης του περιεχομένου του κελιού και θα αποθηκεύεται επίσης στο Blockchain, έτσι ώστε να μπορούμε ακολούθως να εντοπίσουμε τα δεδομένα που ανήκουν σε έναν ασθενή και να βεβαιωθούμε ότι δεν έχουν αλλαχθεί. Σημειώνουμε ότι, κατά την αποθήκευση του παραπάνω αναγνωριστικού στο Blockchain, θα αποθηκεύουμε επίσης και τις οντότητες που έχουν πρόσβαση σε αυτό (smart contracts) μέσω της καταγραφής των διευθύνσεών τους.

Κατά την επίσκεψη ενός ασθενή σε κάποιο *Κέντρο Περίθαλψης* για την πραγματοποίηση των απαιτούμενων εξετάσεων και τη διάγνωση πιθανών προβλημάτων,

[12] Chemical Entities of Biological Interest (ChEBI), https://www.ebi.ac.uk/chebi/
[13] Logical Observation Identifiers Names and Codes (LOINC), https://loinc.org/

θα ενημερώνεται η βάση του συστήματος. Αυτό θα γίνεται μέσω ενός εξειδικευμένου web service, το οποίο θα είναι υπεύθυνο να ενημερώσει τόσο τη βάση, όσο και το Blockchain. Στο σημείο αυτό να τονίσουμε ότι τα δεδομένα που καταγράφονται στις βάσεις αυτών των οργανισμών μπορεί να είναι εκφρασμένα με διαφορετικό τρόπο από αυτόν που απαιτείται για τη λειτουργία του συστήματός μας. Ωστόσο, τα δεδομένα που καταγράφονται θα πρέπει να εκφραστούν με βάση τους όρους του μοντέλου που έχουμε αναπτύξει και τις κωδικοποιήσεις που έχουν επιλεχθεί. Για τον σκοπό αυτό μπορούν να χρησιμοποιηθούν εργαλεία και συστήματα που έχουν αναπτυχθεί στο παρελθόν [19] και επιτρέπουν στους χρήστες να καθορίσουν τον τρόπο συσχέτισης των δικών τους μοντέλων με τους όρους που υπάρχουν στο μοντέλο αναφοράς και ακολούθως να χρησιμοποιήσουν τις συσχετίσεις αυτές για την αυτόματη έκφραση των δεδομένων τους, χρησιμοποιώντας τους όρους της παραπάνω οντολογίας. Να τονίσουμε ότι ο οργανισμός δεν είναι απαραίτητο να αποστέλλει όλα τα δεδομένα που καταγράφονται, παρά μόνο αυτά που είναι απαραίτητα για την αποτίμηση των όρων των συμβολαίων (με άλλα λόγια, αυτά που μπορούν να εκφραστούν με βάση τους όρους του μοντέλου μας).

Σημειώνουμε ότι, κατά την καταγραφή των δεικτών στο Blockchain, θα καταγράφουμε επίσης και τις οντότητες που μπορεί να έχουν πρόσβαση σε αυτά, έτσι ώστε ο χρήστης να είναι ενήμερος για τα δεδομένα που καταγράφονται και πώς αυτά χρησιμοποιούνται. Ειδικότερα, θα διατηρούμε μια λίστα με τις διευθύνσεις των λογαριασμών που επιτρέπεται να έχουν πρόσβαση σε αυτά. Αρχικά, αυτή η λίστα θα είναι κενή, ενώ θα ενημερώνεται ανάλογα με τα συμβόλαια που υπογράφονται, έτσι ώστε αυτά να έχουν πρόσβαση στα απαραίτητα για τους ελέγχους τους δεδομένα, ενώ η πρόσβαση σε αυτά θα εμποδίζεται μετά τη λήξη του συμβολαίου, αφαιρώντας τον αντίστοιχο λογαριασμό/διεύθυνση από τη λίστα.

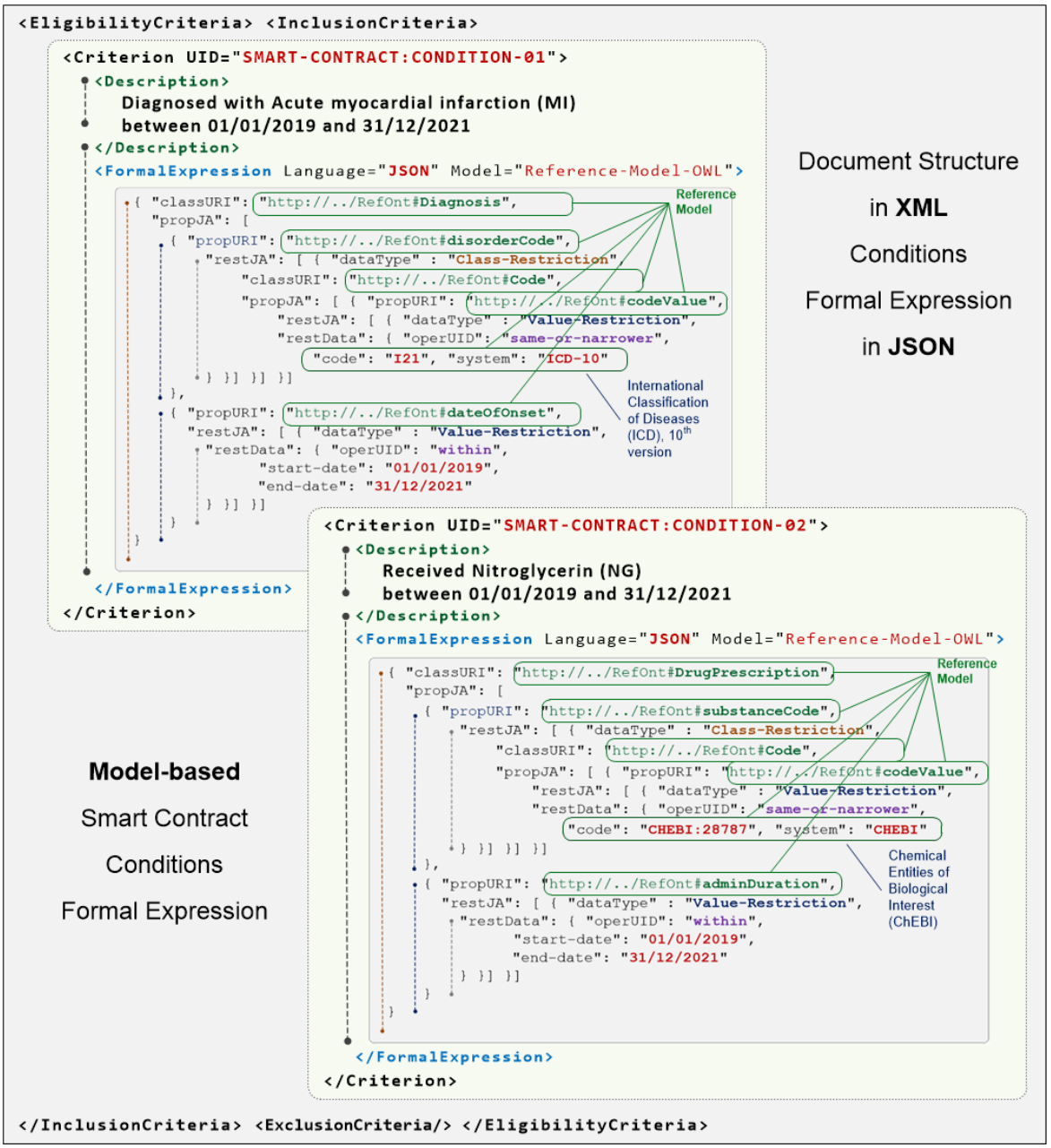
4.2.2 Έκφραση και Αποτίμηση των Όρων των Συμβολαίων

Για τη λεπτομερή *έκφραση των όρων του συμβολαίου* χρησιμοποιήσαμε το μοντέλο που έχουμε ήδη αναπτύξει για την έκφραση των κριτηρίων καταλληλότητας (eligibility criteria aka inclusion/exclusion criteria) μιας κλινικής δοκιμής [20]. Το μοντέλο αυτό μας επιτρέπει να εκφράσουμε περιορισμούς όσον αφορά τις τιμές που μπορούν να πάρουν οι παράμετροι που καταγράφουμε για κάθε ασθενή, σύμφωνα με την παραπάνω οντολογία και κατά συνέπεια είναι ιδανικό για την έκφραση των όρων του συμβολαίου. Για

παράδειγμα (Σχήμα 8), ο ασθενής έλαβε το φάρμακο Νιτρογλυκερίνη (ChEBI: 28787), αλλά παρ' όλα αυτά παρουσίασε Οξύ Έμφραγμα του Μυοκαρδίου (ICD-10: I21) κατά τη διάρκεια ισχύος του συμβολαίου του (η ακριβής ημερομηνία έναρξης και λήξης του συμβολαίου ορίζεται κατά την υπογραφή του) και συνεπώς θα πρέπει να αποζημιωθεί. Επίσης, οι συνθήκες που είναι εκφρασμένες με βάση το παραπάνω μοντέλο, χρησιμοποιώντας την γλώσσα JSON[14] για την έκφραση των περιορισμών που θα πρέπει να ικανοποιούν οι τιμές των παραμέτρων, μπορούν εύκολα να εκφραστούν με τη μορφή ενός SPARQL ερωτήματος [21] και κατ' επέκταση να χρησιμοποιηθούν για τον έλεγχο των δεδομένων των ασθενών. Κατά την εξέταση των δεδομένων των ασθενών, λαμβάνουμε επίσης υπόψη την κατηγοριοποίηση των όρων του μοντέλου αναφοράς μας, για να δούμε εάν όντως ο ασθενής ικανοποιεί τις συνθήκες ή όχι, ακόμη και εάν δεν υπάρχει η αντίστοιχη δήλωση στα δεδομένα που έχουν καταγραφεί. Αυτό μας επιτρέπει να εξάγουμε σημασιολογικά σωστά αποτελέσματα, λαμβάνοντας υπόψη τη σημασία των δεδομένων που έχουν αποθηκευτεί και ειδικότερα των κατασκευαστών που χρησιμοποιούνται για την οργάνωση των όρων σε ευρύτερες κατηγορίες. Δεδομένου ότι τα συμβόλαια που συνάπτονται και ειδικά οι όροι της συμφωνίας θέλουμε να είναι ευρέως γνωστά και να μην μπορούν να τροποποιηθούν, θα τα αποθηκεύουμε στο Blockchain ως Smart Contracts, έτσι ώστε να μπορούν να εκτελούνται γρήγορα και ο χρήστης να αποζημιώνεται άμεσα με το αντίστοιχο ποσό, εφόσον κάτι τέτοιο το δικαιούται.

Όπως γνωρίζουμε, τα συμβόλαια αυτά βασίζονται στα δεδομένα που υπάρχουν ήδη στην αλυσίδα. Για τον σκοπό αυτό υλοποιήθηκε κάποιο επιπλέον service (γνωστό ως oracle), το οποίο είναι υπεύθυνο να εξετάσει τις συνθήκες και κατά πόσο αυτές πληρούνται και ακολούθως να ενημερώσει το Blockchain, καταγράφοντας το γεγονός αυτό, χωρίς ωστόσο να περιέχει αυτό προσωπικά δεδομένα των ασθενών. Αυτό, βέβαια, προϋποθέτει ότι τόσο ο χρήστης, όσο και ο πάροχος εμπιστεύονται το service αυτό. Για λόγους βελτίωσης της αξιοπιστίας του συστήματος μπορούμε επιπρόσθετα να καταγράφουμε παραπάνω πληροφορία στο Blockchain όσον αφορά το πότε έτρεξε τελευταία φορά το service και να χρησιμοποιήσουμε αλγορίθμους κρυπτογράφησης συμμετρικού κλειδιού τόσο για την επικοινωνία με το service, όσο και για την αποθήκευση των δεδομένων. Ωστόσο, αυτά θα μπορούσαμε να πούμε ότι είναι βελτιστοποιήσεις του συστήματος και δεν παρέχονται στην τρέχουσα έκδοσή του.

[14] JavaScript Object Notation (JSON), https://www.json.org/json-en.html



Σχήμα 8: Τυπική Έκφραση των Όρων/Συνθηκών ενός Συμβολαίου Υγείας

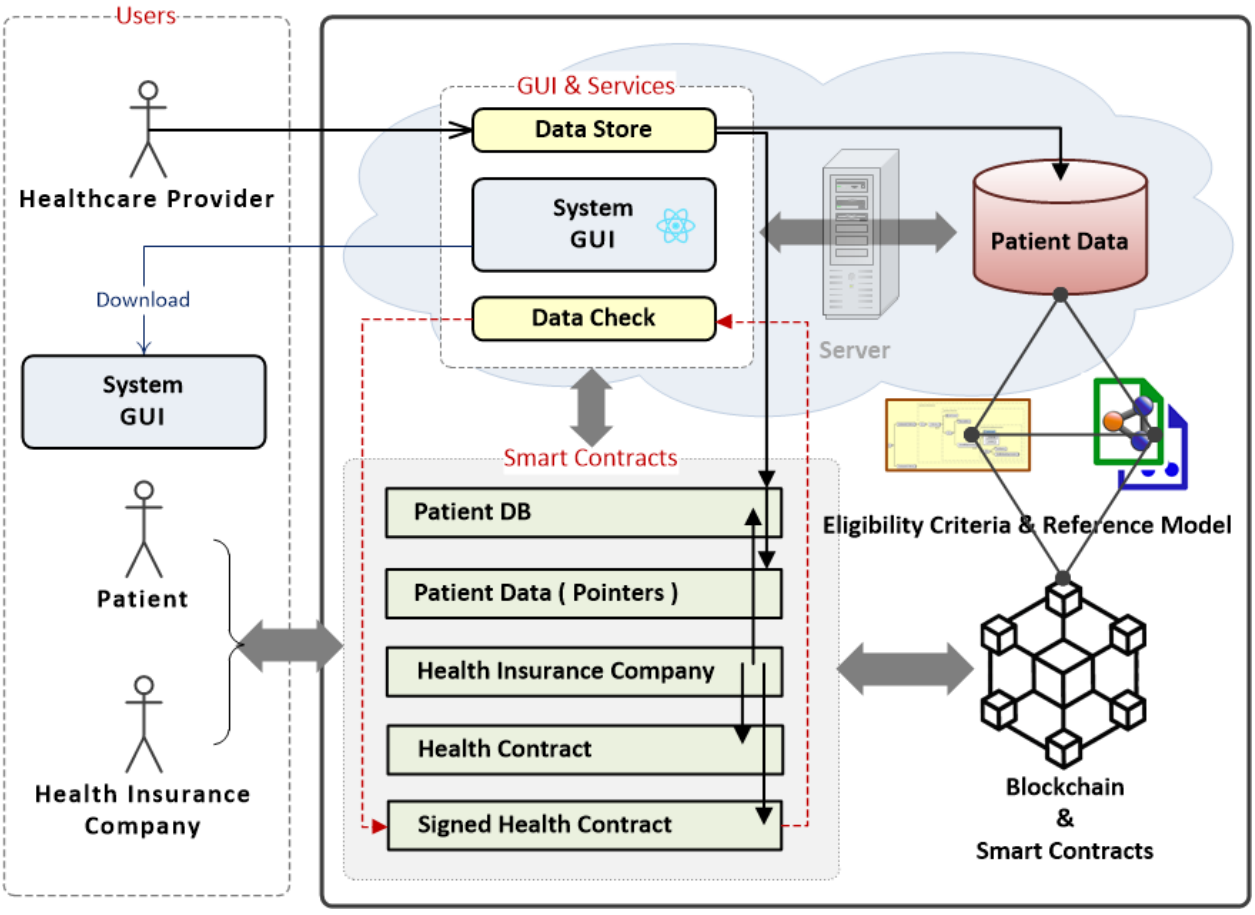
5 *Υλοποίηση Συστήματος και Τεχνικές Λεπτομέρειες*

5.1 Αρχιτεκτονική Συστήματος και Αλληλεπίδραση μεταξύ Οντοτήτων

Η επικοινωνία του χρήστη με το σύστημα γίνεται μέσω ενός γραφικού περιβάλλοντος, το οποίο αρχικά προτρέπει τον χρήστη να συνδεθεί στον λογαριασμό του και ακολούθως τον διευκολύνει στη διεκπεραίωση των συναλλαγών του, καλώντας τις αντίστοιχες μεθόδους των smart contracts που έχουν υλοποιηθεί. Στην περίπτωση στην οποία απαιτείται πρόσβαση στη βάση, όπου είναι αποθηκευμένα τα δεδομένα των χρηστών, αυτό γίνεται μέσω εξειδικευμένων web services, τα οποία μπορούν να καλέσουν συγκεκριμένοι χρήστες/οντότητες/λογαριασμοί, οι οποίοι έχουν αναλάβει τον ρόλο αυτό και κατά συνέπεια μπορούν να καλύψουν το κόστος πραγματοποίησης των συναλλαγών. Στο Blockchain αποθηκεύονται οι δείκτες προς τα αντίστοιχα δεδομένα των χρηστών καθώς και τα υπογεγραμμένα έξυπνα συμβόλαια, τα οποία έχουν πρόσβαση σε αυτά.

Στο Σχήμα 9 παρουσιάζεται *η αρχιτεκτονική του συστήματος*, όπου μπορούμε να διακρίνουμε τους τρεις διαφορετικούς χρήστες καθώς και τα επιμέρους τμήματα του συστήματος. Το Γραφικό Περιβάλλον υπάρχει κυρίως για την αλληλεπίδραση των ασθενών με το σύστημα. Μέσω της εφαρμογής αυτής οι χρήστες μπορούν να δουν τα διαθέσιμα συμβόλαια καθώς και να διαλέξουν αυτό που τους ταιριάζει, καταβάλλοντας το αντίστοιχο ποσό. Τα δύο web service χρησιμεύουν για την αποθήκευση και τον έλεγχο των δεδομένων των ασθενών. Τόσο το γραφικό περιβάλλον της πλατφόρμας, όσο και τα web services αλληλεπιδρούν με τα smart contracts που έχουν υλοποιηθεί, για να παρέχουν το επιθυμητό αποτέλεσμα. Αξίζει να αναφέρουμε ότι ο χρήστης μιλάει απευθείας με την πλατφόρμα του Blockchain, καθώς η εφαρμογή αρχικά μεταφέρεται στον υπολογιστή του χρήστη και ακολούθως χρησιμοποιείται για την επικοινωνία με τα smart contracts. Τα web services παραμένουν στον Application Server που βρίσκεται στο cloud και χρησιμοποιούνται για τη διευκόλυνση της πρόσβασης στα δεδομένα των ασθενών. Τα

δεδομένα των ασθενών καταλήγουν σε μία σχεσιακή βάση, η οποία περιέχει έναν πίνακα με το hash των δεδομένων των χρηστών καθώς και τα δεδομένα αυτά (RDF/XML), εκφρασμένα με βάση τους όρους του Μοντέλου Αναφοράς που είδαμε στο προηγούμενο κεφάλαιο. Όπως φαίνεται και στο σχήμα, τα δεδομένα των χρηστών που αποθηκεύονται στη βάση, τα δεδομένα των χρηστών που αποθηκεύονται στο Blockchain, το Μοντέλο Αναφοράς και τα Κριτήρια Καταλληλότητας που χρησιμοποιούνται για την τυπική έκφραση των όρων του συμβολαίου συνδέονται στενά μεταξύ τους και η χρήση όλων αυτών είναι απαραίτητη για τον εντοπισμό των δεδομένων των ασθενών, την ερμηνεία τους και τον έλεγχο των συνθηκών του συμβολαίου.



Σχήμα 9: Αρχιτεκτονική του Συστήματος

5.1.1 Αρχικοποίηση Συστήματος

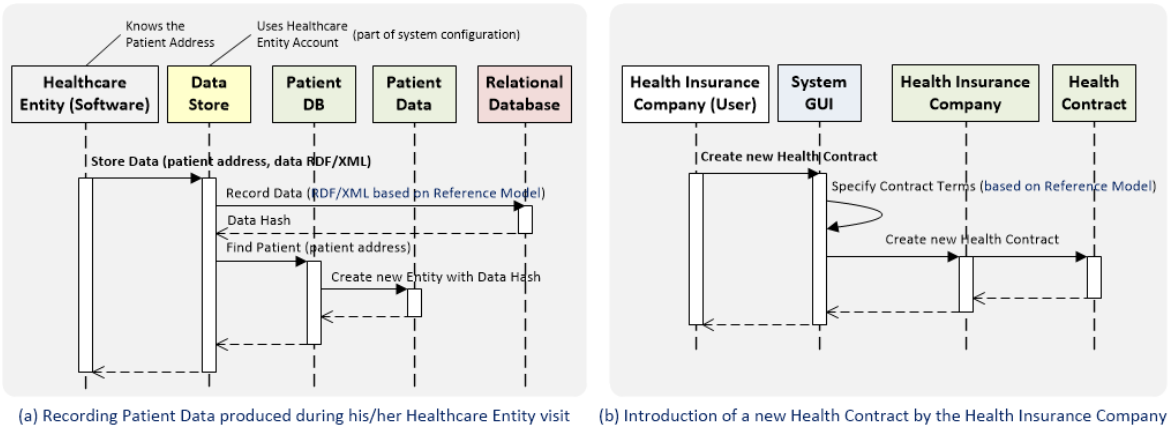
Για την ορθή λειτουργία του συστήματος απαιτείται μια φάση αρχικοποίησης, σύμφωνα με την οποία θα πρέπει το Κέντρο Περίθαλψης Ασθενών να ανανεώσει τα συστήματά του, έτσι ώστε τα δεδομένα που καταγράφονται για κάθε ασθενή κατά την επίσκεψή του σ'

αυτό να παρέχονται και στην πλατφόρμα που έχουμε αναπτύξει, καλώντας το αντίστοιχο Web Service (Data Store). Ειδικότερα, θα πρέπει να παρέχει στο web service την public διεύθυνση του χρήστη (εναλλακτικά θα μπορούσε να ήταν κάποιο άλλο αναγνωριστικό, το οποίο θα μας βοηθούσε να εντοπίσουμε τον λογαριασμό του ασθενούς) καθώς και τα δεδομένα που θέλουμε να αποθηκευτούν, εκφρασμένα με βάση τους όρους του μοντέλου που έχουμε αναπτύξει. Ακολούθως, το σύστημα θα κατευθύνει τα δεδομένα αυτά στη σχεσιακή βάση, αλλά και θα εντοπίζει και θα ανανεώνει τον λογαριασμό του χρήστη στην Blockchain πλατφόρμα, καταγράφοντας τον δείκτη (hash) που παραπέμπει στα αντίστοιχα δεδομένα που βρίσκονται στη σχεσιακή βάση. Σημειώνουμε ότι το web service για τον έλεγχο των δεδομένων θα επικοινωνεί και αυτό αρχικά με το Blockchain, έτσι ώστε να εντοπίσει τα δεδομένα του χρήστη (pointers) και ειδικότερα αυτά στα οποία επιτρέπεται να έχει πρόσβαση, ώστε ακολούθως να ελέγξει κατά πόσο πληρούνται οι συνθήκες του συμβολαίου.

Κατά την αρχικοποίηση του συστήματος συνίσταται επίσης ο καθορισμός των διαθέσιμων συμβολαίων, έτσι ώστε να μπορούν ακολούθως να επιλεγούν από τον ασθενή. Κατά τη διαδικασία αυτή η ασφαλιστική εταιρία καλείται να εκφράσει τους όρους του συμβολαίου (τα κριτήρια που πρέπει να ικανοποιεί ο ασθενής, για να λάβει την αποζημίωση) με βάση τους όρους του μοντέλου αναφοράς που έχουμε αναπτύξει, τη διάρκεια του συμβολαίου καθώς και τα χρήματα που θα πρέπει να καταβάλει ο χρήστης για την σύναψη του συμβολαίου καθώς και το ποσό που θα πάρει πίσω ως αποζημίωση, σε περίπτωση που ικανοποιούνται οι όροι του συμβολαίου. Για τον σκοπό αυτό, κατά τη σύναψη του συμβολαίου, θα μεταφέρεται ένα ποσό από τον λογαριασμό της ασφαλιστικής εταιρίας στο smart contract που έχει υπογραφεί, έτσι ώστε να μπορεί να καλύψει τον πελάτη, σε περίπτωση που ικανοποιούνται οι συνθήκες. Τα χρήματα αυτά θα μπορεί η εταιρία να τα πάρει πρακτικά πίσω, όταν λήξει το συμβόλαιο. Υπενθυμίζουμε ότι τα έξυπνα συμβόλαια ελέγχονται από την πλατφόρμα του Blockchain συστήματος (σε αντίθεση με τους λογαριασμούς των χρηστών) και επομένως τα χρήματα που τοποθετούνται στα έξυπνα συμβόλαια είναι εν γένει ασφαλή και μέρος αυτών μπορεί να μεταφερθεί είτε στον ασθενή είτε στην εταιρία μέσω των κατάλληλων μεθόδων, εφόσον πληρούνται οι όροι του συμβολαίου.

5.1.2 Αλληλεπίδραση μεταξύ των Βασικών Οντοτήτων του Συστήματος

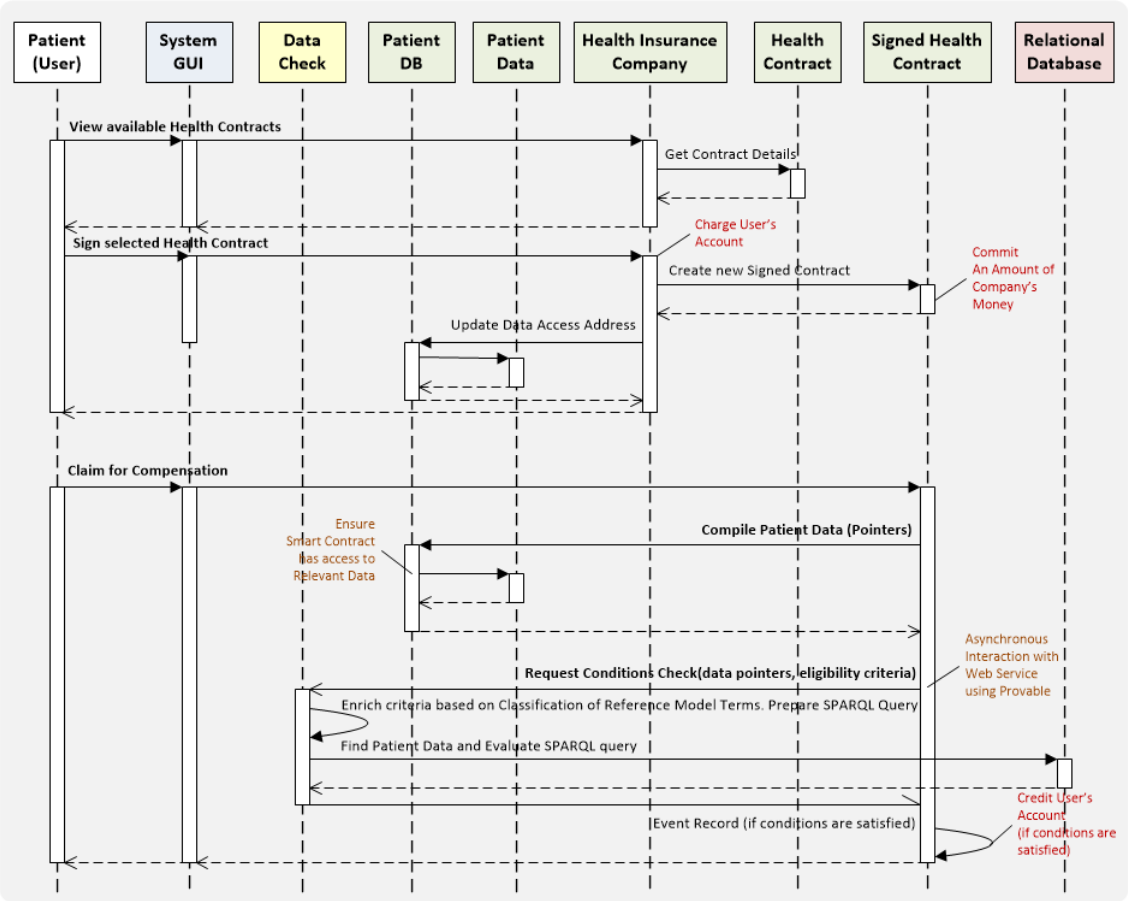
Στο Σχήμα 10 μπορούμε να δούμε την αλληλεπίδραση της Μονάδας Περίθαλψης (Healthcare Entity) καθώς και της Ασφαλιστικής Εταιρίας Υγείας (Health Insurance Company) με τις υπόλοιπες οντότητες του συστήματος, συμπεριλαμβανομένων των Smart Contracts, του Γραφικού Περιβάλλοντος, της Βάσης Δεδομένων και των Λοιπών Υπηρεσιών. Όπως φαίνεται και στο σχήμα (a), αρχικά τα δεδομένα καταχωρούνται στη σχεσιακή βάση και ακολούθως ενημερώνεται το Blockchain. Στην περίπτωση δημιουργίας ενός νέου συμβολαίου (b), αφού καθοριστούν οι επιμέρους παράμετροι, τα δεδομένα αποθηκεύονται κατευθείαν στο Blockchain.



Σχήμα 10: Αλληλεπίδραση (α) του Κέντρου Περίθαλψης και (β) της Ασφαλιστικής Εταιρίας Υγείας με τις υπόλοιπες οντότητες του συστήματος

Ακολούθως ο χρήστης μπορεί να επικοινωνήσει με το σύστημα, να δει τα διαθέσιμα συμβόλαια, να υπογράψει αυτό που επιθυμεί και να αιτηθεί της λήψης αποζημίωσης, η οποία ικανοποιείται άμεσα, εφόσον πληρούνται οι όροι. Η αλληλεπίδραση των ασθενών με τις υπόλοιπες οντότητες του συστήματος φαίνεται στο παρακάτω ακολουθιακό διάγραμμα (Σχήμα 11). Όπως έχουμε αναφέρει και πιο πάνω, κατά την υπογραφή του συμβολαίου ο λογαριασμός του χρήστη χρεώνεται άμεσα με το αντίστοιχο ποσό, ενώ παράλληλα γίνεται δέσμευση ενός ποσού της ασφαλιστικής εταιρίας, για να καλύψει τις ανάγκες αποζημίωσής του, εάν αυτό κριθεί απαραίτητο. Επίσης, για τον έλεγχο των συνθηκών, το σύστημα επικοινωνεί τόσο με τα Smart Contracts που υπάρχουν στο Blockchain, όσο και με τη βάση, για να εντοπίσει τα δεδομένα των ασθενών και ακολούθως χρησιμοποιεί την τυπική έκφραση των όρων των συνθηκών, για να εξετάσει κατά πόσο πληρούνται οι όροι του συμβολαίου. Κατά τη

διαδικασία αυτή λαμβάνει υπόψη τη σημασία των ιατρικών όρων και ειδικότερα την κατηγοριοποίησή τους για τη σημασιολογικά σωστή αποτίμηση των συνθηκών. Ειδικότερα, αφού πρώτα εντοπίσει όλους τους όρους με την ίδια σημασία, δημιουργεί το κατάλληλο ASK SPARQL[15] ερώτημα, το οποίο ακολούθως χρησιμοποιεί, για να εξετάσει εάν τα υπάρχοντα δεδομένα των χρηστών ικανοποιούν τους όρους του συμβολαίου. Στην περίπτωση αυτή, το προσυμφωνηθέν ποσό κατατίθεται στον λογαριασμό του χρήστη.



Σχήμα 11: Αλληλεπιδράσεις του Ασθενή με τις υπόλοιπες οντότητες του συστήματος

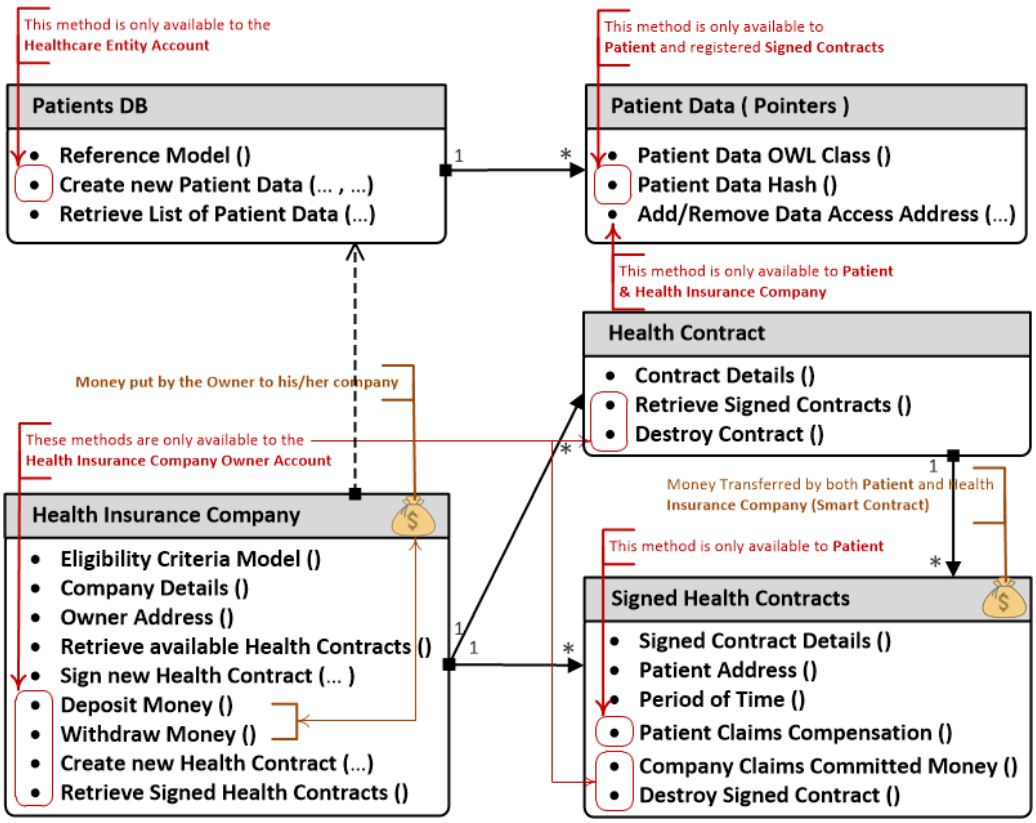
5.2 Έξυπνα Συμβόλαια

Στην ενότητα αυτή υπάρχει λεπτομερής περιγραφή των έξυπνων συμβολαίων που έχουν υλοποιηθεί, συμπεριλαμβανομένων των δεδομένων που αποθηκεύονται, των επιμέρους συναρτήσεων και της αλληλεξάρτησης μεταξύ τους (Σχήμα 12).

[15] SPARQL Query Language for RDF (SPARQL), https://www.w3.org/TR/rdf-sparql-query/

- **Βάση (Patient DB) και Δεδομένα Ασθενών (Patient Data)**

Για τη διαχείριση των δεδομένων των ασθενών έχουν υλοποιηθεί δύο διαφορετικά έξυπνα συμβόλαια. Το πρώτο συμβόλαιο ονομάζεται Patient DB. Το συμβόλαιο αυτό γίνεται deploy μία μόνο φορά κατά την αρχικοποίηση του συστήματος και μας βοηθάει να εντοπίσουμε τα δεδομένα των ασθενών καθώς επίσης και το μοντέλο, με βάση το οποίο είναι εκφρασμένα τα δεδομένα αυτά. Η πληροφορία για τα επιμέρους δεδομένα ενός ασθενή βρίσκεται σε ένα (ή παραπάνω) smart contract, που ονομάζεται Patient Data και στο οποίο καταγράφεται ο δείκτης προς τα δεδομένα που υπάρχουν στη σχεσιακή βάση (hash), η κλάση του μοντέλου αναφοράς στην οποία ανήκουν τα δεδομένα, καθώς επίσης και μια λίστα με τις διευθύνσεις των χρηστών (έξυπνων συμβολαίων ασφαλιστικών εταιριών), που μπορούν να έχουν πρόσβαση στα δεδομένα αυτά. Σημειώνουμε ότι μία οντότητα (instance) του "συμβολαίου" αυτού δημιουργείται μέσα από το προηγούμενο συμβόλαιο (Patient DB), το οποίο μας βοηθά να εντοπίσουμε τα δεδομένα που υπάρχουν για κάθε χρήστη, καθώς και να ελέγξουμε εάν μια οντότητα έχει πρόσβαση σε αυτά. Αυτό γίνεται εξετάζοντας τη διεύθυνση του χρήστη και κατά πόσο η διεύθυνση αυτή ανήκει στις οντότητες που επιτρέπεται να έχουν πρόσβαση στα δεδομένα.



Σχήμα 12: Έξυπνα Συμβόλαια Εφαρμογής

- **Εταιρία Ασφάλειας Υγείας (Health Insurance Company)**

Το έξυπνο αυτό συμβόλαιο γίνεται deploy μία μόνο φορά (από τον λογαριασμό του υπευθύνου της εταιρίας) και χρησιμοποιείται για την καταγραφή των βασικών παραμέτρων της ασφαλιστικής εταιρίας, όπως είναι το όνομα και η διεύθυνσή της. Στο συμβόλαιο αυτό καταγράφουμε επίσης και τη διεύθυνση του χρήστη που είναι υπεύθυνος για την εταιρία, έτσι ώστε να χρησιμοποιήσουμε αργότερα την πληροφορία αυτή, για να περιορίσουμε την πρόσβαση στις μεθόδους που μπορούν να δημιουργήσουν ή να καταστρέψουν ένα υπάρχον συμβόλαιο (υπογεγραμμένο ή μη). Επίσης, ο λογαριασμός αυτός θα πρέπει να διαθέτει ένα σημαντικό ποσό χρημάτων, για να μας εξασφαλίσει ότι θα μπορέσουμε να καταβάλουμε τα απαραίτητα χρήματα στους ασθενείς, εάν αυτό χρειαστεί. Συνεπώς, ο χρήστης (ιδιοκτήτης) μπορεί να καταθέσει (deposit) κάποια χρήματα από τον λογαριασμό του σε αυτό το smart contract για τη λειτουργία της επιχείρησης. Επίσης, έχει τη δυνατότητα να αποσύρει (withdraw) τα χρήματά του από την επιχείρηση ή μέρος αυτών ή ακόμη και να την κλείσει/καταστρέψει (destroy).

Μέσω του έξυπνου αυτού συμβολαίου ο ασθενής μπορεί να εντοπίσει τα διαθέσιμα συμβόλαια και να υπογράψει αυτό που καλύπτει τις ανάγκες του. Κατά τη σύναψη ενός νέου συμβολαίου ελέγχουμε τα χρήματα που υπάρχουν στον λογαριασμό της εταιρίας και εάν αυτά επαρκούν, για να καλύψουν τις ανάγκες της (αποζημίωση ασθενών). Για να βεβαιωθούμε ότι ο ιδιοκτήτης της εταιρίας δεν θα σπαταλήσει τα χρήματα αυτά, κατά την υπογραφή ενός νέου συμβολαίου με έναν ασθενή, γίνονται τα εξής: (α) ο ασθενής μεταφέρει στον λογαριασμό/έξυπνο συμβόλαιο που εκφράζει τη συμφωνία που έχει πραγματοποιηθεί το απαιτούμενο ποσό (π.χ. 5 Ether), με την συγκατάθεση του χρήστη (β) η εταιρία μεταφέρει σε αυτό το έξυπνο συμβόλαιο το απαραίτητο ποσό (π.χ., άλλα 5 Ether) αυτόματα, έτσι ώστε το ποσό που έχει συνολικά κατατεθεί (δλδ, 10 Ether) να μπορεί να καλύψει/πληρώσει τον ασθενή, σε περίπτωση που ικανοποιούνται οι όροι του συμβολαίου. Η εταιρία μπορεί να πάρει πίσω τα χρήματα αυτά μετά τη λήξη του συμβολαίου. Σημειώνουμε ότι η εταιρία μπορεί επίσης να καταργήσει ένα συμβόλαιο (υπογεγραμμένο ή μη), αποζημιώνοντας όλους τους εμπλεκόμενους χρήστες (στην περίπτωση ενός υπογεγραμμένου συμβολαίου). Στην περίπτωση αυτή ο ασθενής θα λάβει είτε το διπλάσιο των χρημάτων που κατέβαλε ή ολόκληρο το ποσό που δικαιούται (εφόσον πληρούνται οι όροι του συμβολαίου).

- **Όροι Συμβολαίου Υγείας (Health Contract)**

Το έξυπνο αυτό συμβόλαιο περιέχει τις βασικές παραμέτρους της ασφάλειας υγείας, συμπεριλαμβανομένων του κόστους της, της χρονικής της διάρκειας, των προβλημάτων που καλύπτει και των επιπρόσθετων συνθηκών που θα πρέπει να ικανοποιεί ο ασθενής, τα οποία είναι εκφρασμένα χρησιμοποιώντας το μοντέλο έκφρασης των κριτηρίων καταλληλότητας, λαμβάνοντας υπόψη τους όρους που έχουμε ορίσει στο μοντέλο αναφοράς, όπως είδαμε στο προηγούμενο κεφάλαιο. Το συμβόλαιο αυτό μπορεί ο χρήστης να το υπογράψει, καταβάλλοντας το απαιτούμενο ποσό. Στην περίπτωση αυτή, δημιουργείται ένα άλλο έξυπνο συμβόλαιο, στο οποίο καταγράφονται οι όροι του συμβολαίου και αποθηκεύεται το απαραίτητο ποσό χρημάτων. Τη διαδικασία υπογραφής ενός συμβολαίου την αναλαμβάνει το έξυπνο συμβόλαιο της εταιρίας που είδαμε πιο πάνω, που έχει άμεση πρόσβαση στα χρήματα που έχουν κατατεθεί. Αυτό το συμβόλαιο μας δίνει, όμως, τη δυνατότητα να καταγράψουμε τα συμβόλαια που έχουν δημιουργηθεί και ακολούθως υπογραφεί από τον χρήστη, χρησιμοποιώντας αυτό ως πρότυπο. Τέλος, δίνει τη δυνατότητα στην εταιρία να το απενεργοποιήσει, αποζημιώνοντας όλους τους χρήστες που έχουν συνάψει κάποια συμφωνία με βάση αυτό.

- **Επιλεγμένο/Υπογεγραμμένο Συμβόλαιο Υγείας (Signed Health Contract)**

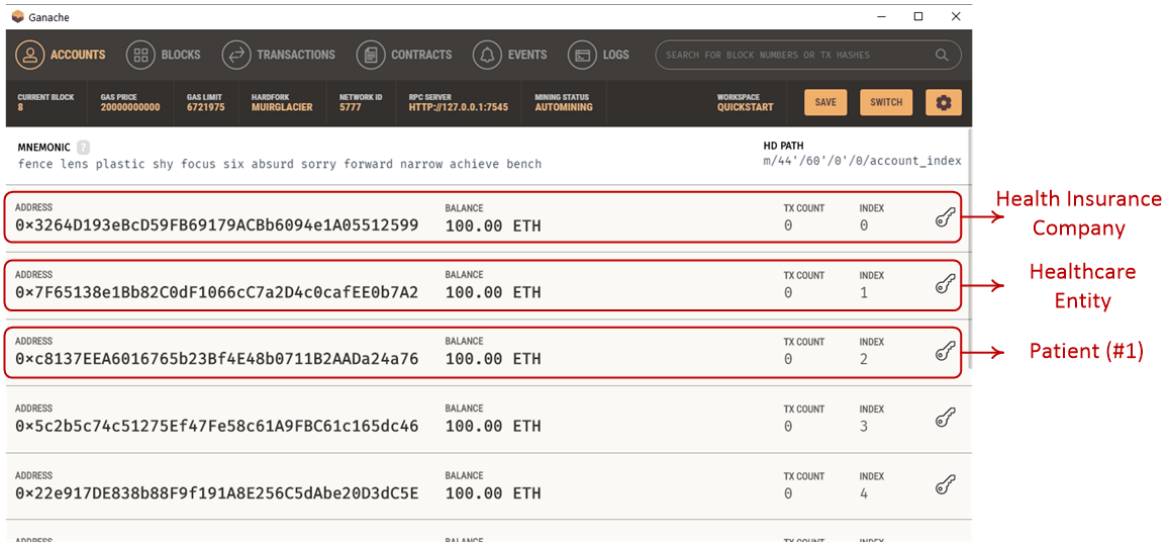
Στο «υπογεγραμμένο» συμβόλαιο αποθηκεύουμε τη διεύθυνση του χρήστη, τους όρους του συμβολαίου, τη συγκεκριμένη χρονική διάρκεια καθώς και μια μεταβλητή που δείχνει, εάν έχουν ποτέ εκπληρωθεί οι όροι του συμβολαίου ή όχι. Η μεταβλητή αυτή χρησιμοποιείται, έτσι ώστε ο ασθενής να μπορεί να αιτηθεί μία μόνο φορά την αποζημίωση, καλώντας την αντίστοιχη μέθοδο. Στην περίπτωση αυτή (patient claims compensation), το σύστημα εντοπίζει τα δεδομένα του ασθενή και ειδικότερα τους δείκτες προς αυτά και ακολούθως «καλεί» ασύγχρονα ένα εξωτερικό service, παρέχοντας τους δείκτες και τα κριτήρια που έχουν οριστεί, προκειμένου αυτό να ελέγξει εάν πληρούνται οι όροι και ακολούθως να ενημερώσει το smart contract με την καταγραφή του κατάλληλου γεγονότος. Το smart contract επιτρέπει στην εταιρία να πάρει πίσω τα χρήματα που της είχαν δεσμευτεί (είτε όλο το ποσό είτε μέρος αυτού), εφόσον το συμβόλαιο έχει λήξει και/ή ο χρήστης έχει αποζημιωθεί. Στην περίπτωση αυτή, το ποσό που υπάρχει στο υπογεγραμμένο συμβόλαιο μεταφέρεται στο έξυπνο συμβόλαιο που διαχειρίζεται ο ιδιοκτήτης της εταιρίας (απ’ όπου μπορεί να βάλει τα χρήματα στον λογαριασμό του, εάν κάτι τέτοιο επιθυμεί) και το υπογεγραμμένο αυτό συμβόλαιο

καταστρέφεται. Ωστόσο, η εταιρία μπορεί να ζητήσει τη «βίαιη» λύση του συμβολαίου, αποζημιώνοντας τον χρήστη με το απαραίτητο ποσό/ποινή (το ποσό της εταιρίας που έχει δεσμευτεί θα πρέπει να είναι ικανό να καλύψει το χειρότερο σενάριο).

5.3 Τεχνικές Λεπτομέρειες

Για την υλοποίηση της εφαρμογής μας χρησιμοποιήσαμε αρκετά διαφορετικά εργαλεία, τα οποία μας επιτρέπουν να φτιάξουμε μία τοπική έκδοση του Ethereum, να συνδεθούμε σε αυτό και ακολούθως να καλέσουμε τις συναρτήσεις των smart contracts μέσα από την εφαρμογή μας.

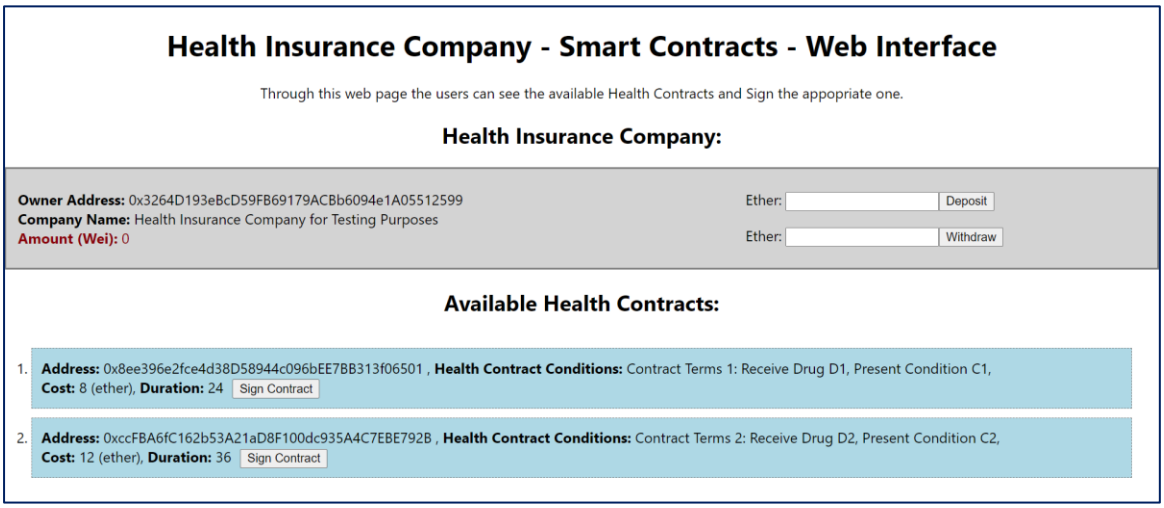
Για να εγκαταστήσουμε μια τοπική/προσωπική έκδοση του Ethereum Blockchain, χρησιμοποιήσαμε το εργαλείο Ganache[16], το οποίο είναι μέρος της Truffle[17] πλατφόρμας. Το εργαλείο αυτό μας παρέχει άμεσα 10 διαφορετικούς λογαριασμούς χρηστών, καθένας από τους οποίους έχει 100 «ψεύτικα» Ether στον λογαριασμό του. Από αυτούς τους λογαριασμούς θεωρήσαμε ότι ο πρώτος είναι ο λογαριασμός του ανθρώπου που είναι υπεύθυνος για τη λειτουργία της ασφαλιστικής εταιρίας, ο δεύτερος είναι ο λογαριασμός του ανθρώπου που είναι υπεύθυνος για τη λειτουργία του κέντρου περίθαλψης ασθενών και ο τρίτος είναι ο λογαριασμός ενός ασθενή (Σχήμα 13).



Σχήμα 13: Ganache Ethereum Blockchain Accounts

[16] Ganache , https://www.trufflesuite.com/ganache
[17] Truffle Suite, https://www.trufflesuite.com/

Τα έξυπνα συμβόλαια τα εκφράσαμε στη γλώσσα solidity[18], τα κάναμε compile, χρησιμοποιώντας τα εργαλεία της Truffle πλατφόρμας, ώστε να προκύψει ο bytecode (και ABI) και ακολούθως τα κάναμε deploy στο Blockchain. Ειδικότερα, το deployment των συμβολαίων σχετικά με την καταγραφή/αναφορά των δεδομένων των ασθενών έγινε από τον λογαριασμό του Κέντρου Περίθαλψης, ενώ το deployment των συμβολαίων σχετικών με την Ασφάλιση των ασθενών έγινε deploy από το λογαριασμό της Ασφαλιστικής Εταιρίας. Σημειώνουμε ότι οι δημόσιες διευθύνσεις των δύο συμβολαίων (Patient DB και Health Insurance Company) καταγράφηκαν σε ένα αρχείο, έτσι ώστε να μπορούν έπειτα να χρησιμοποιηθούν από το Γραφικό περιβάλλον του συστήματος. Αυτό αναπτύχθηκε χρησιμοποιώντας το framework React[19], το οποίο μας επιτρέπει να φτιάχνουμε πολύπλοκα αλληλεπιδραστικά γραφικά περιβάλλοντα. Για την επικοινωνία του χρήστη με το Blockchain/smart contracts χρησιμοποιήθηκε η βιβλιοθήκη web3[20]. Για τη σύνδεση του χρήστη με το Blockchain χρησιμοποιήσαμε το Google Chrome Web Browser και ειδικότερα το MetaMask Extension[21] (Chrome plugin), το οποίο μας επιτρέπει να διαχειριστούμε τους λογαριασμούς που έχουμε στο τοπικό Blockchain και να τους «περάσουμε» στην εφαρμογή/σελίδα, έτσι ώστε να χρησιμοποιηθούν ακολούθως για την επικοινωνία με το Blockchain και τα Smart Contracts. Ένα πολύ μικρό τμήμα του κώδικα που αναπτύχθηκε υπάρχει στο παράρτημα, στο Κεφάλαιο 8.4.

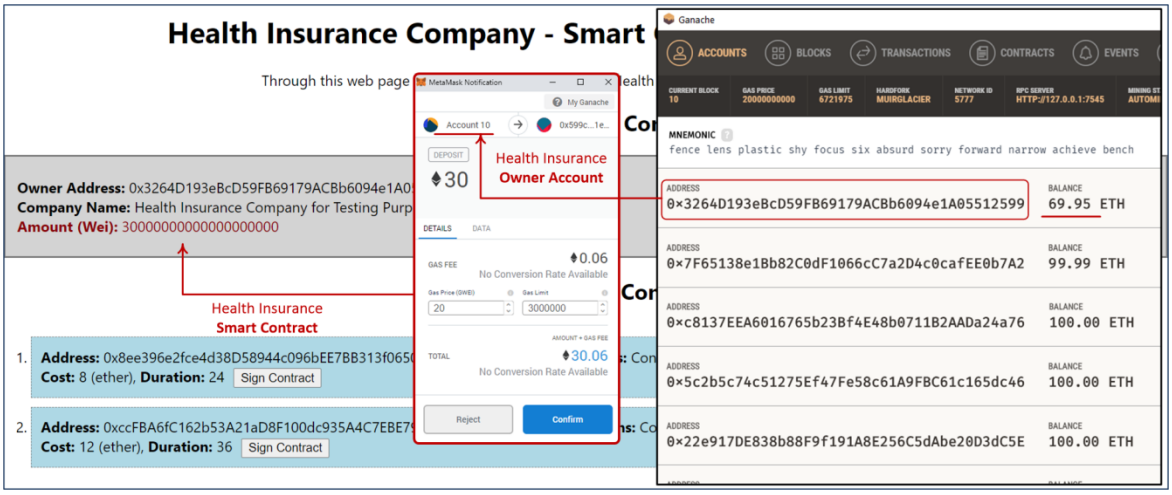


Σχήμα 14: Γραφικό Περιβάλλον για τη Διαχείριση των Χρημάτων της Εταιρίας

[18] Solidity Documentation, https://solidity.readthedocs.io/
[19] React, https://reactjs.org/
[20] Web3.js, https://web3js.readthedocs.io/
[21] MetaMask, https://metamask.io/

Στο Σχήμα 14 μπορούμε να δούμε τη σελίδα της εφαρμογής μας, αφού, όμως, πρώτα συνδεθήκαμε στο σύστημα (με την βοήθεια του MetaMask), χρησιμοποιώντας τον πρώτο λογαριασμό που αντιστοιχεί στον λογαριασμό της Ασφαλιστικής Εταιρίας. Μέσω της σελίδας αυτής μπορούμε να καταθέσουμε χρήματα στον λογαριασμό της εταιρίας (Smart Contract), αλλά και να πάρουμε χρήματα από αυτόν. Τα χρήματα αυτά είναι απαραίτητα, προκειμένου να διασφαλίσουμε τους πελάτες (ασθενείς) και ειδικότερα, ότι υπάρχουν τα διαθέσιμα χρήματα, τα οποία θα μπορέσουν να καλύψουν τις ανάγκες της εταιρίας (αποζημίωση ασθενών), εφόσον πληρούνται οι όροι των συμβολαίων.

Στο Σχήμα 15 μπορούμε να δούμε την εικόνα του συστήματος μετά την κατάθεση 30 Ether στον λογαριασμό της εταιρίας. Ειδικότερα, στο αριστερό μέρος της εικόνας μπορούμε να δούμε τα χρήματα που έχουν κατατεθεί στο Smart Contract (αφού βέβαια δώσαμε τη συγκατάθεσή μας στη μεταφορά αυτή), ενώ στο δεξί μέρος μπορούμε να δούμε το υπόλοιπο των λογαριασμών του συστήματος. Όπως μπορούμε να δούμε, το υπόλοιπο του πρώτου λογαριασμού είναι κάτι λιγότερο από 70 Ether, καθώς χρησιμοποιήθηκαν κάποια επιπρόσθετα χρήματα για το deployment των σχετικών contracts, αλλά και την καταγραφή της νέας δοσοληψίας (κατάθεση 30 ether) στον λογαριασμό της εταιρίας.

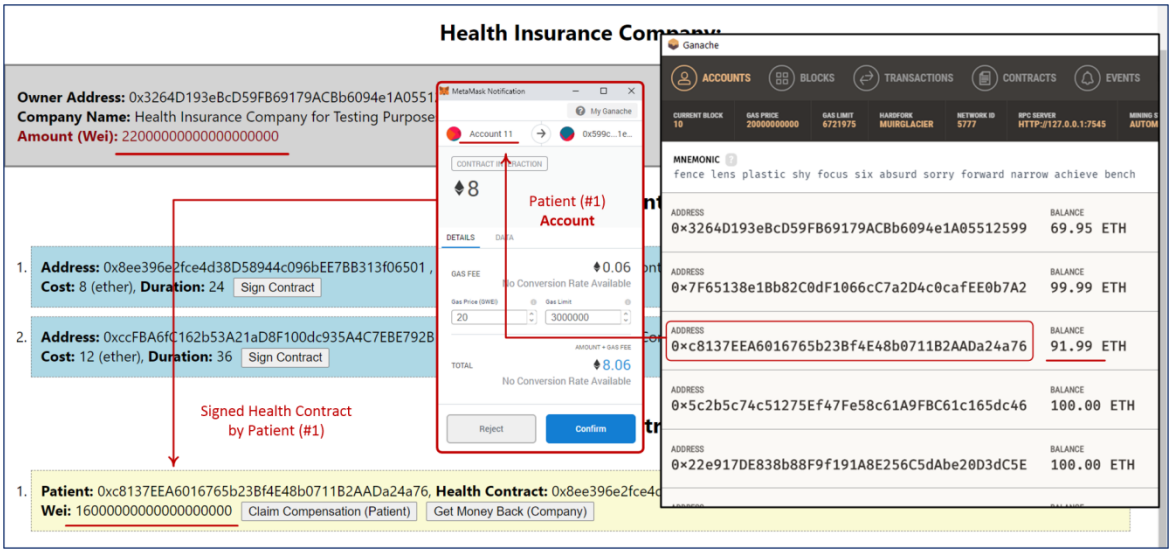


Σχήμα 15: Γραφικό Περιβάλλον και Υπόλοιπο Λογαριασμών μετά την Κατάθεση 30 Ether

Μέσω της σελίδας αυτής, μπορούμε να δούμε επίσης τα διαθέσιμα συμβόλαια και να υπογράψουμε το επιθυμητό, καταβάλλοντας το απαιτούμενο ποσό. Σημειώνουμε ότι ο ασθενής μπορεί να δει το ποσό που υπάρχει στον λογαριασμό της εταιρίας (smart

contract), αλλά δεν έχει κάποιο άλλο δικαίωμα σε αυτόν, καθώς η κατάθεση ή απόσυρση των χρημάτων που υπάρχουν στον λογαριασμό αυτό μπορεί να γίνει μόνο από τον διαχειριστή της ασφαλιστικής εταιρίας. Επίσης, ρυθμίσαμε το σύστημα κατά τέτοιο τρόπο, ώστε τα διαθέσιμα συμβόλαια να μπορούν να επιλεχθούν/υπογραφούν από οποιονδήποτε χρήστη, πλην του διαχειριστή του συστήματος και του κέντρου περίθαλψης. Συνεπώς, για να μπορέσουμε να υπογράψουμε ένα συμβόλαιο, συνδεθήκαμε στο σύστημα χρησιμοποιώντας τον τρίτο λογαριασμό, οποίος αντιστοιχεί σε έναν ασθενή, όπως αναφέραμε και πιο πάνω.

Στο Σχήμα 16 μπορούμε να δούμε την κατάσταση του συστήματος μετά την υπογραφή του πρώτου συμβολαίου από τον ασθενή. Σημειώνουμε ότι, για να συμβεί αυτό, ο χρήστης/ασθενής έδωσε την συγκατάθεσή του για την μεταφορά των απαιτούμενων χρημάτων (8 Ether) στον λογαριασμό της εταιρίας (όπως φαίνεται και στο σχήμα που είναι στο δεξί μέρος). Ωστόσο, όπως μπορούμε να παρατηρήσουμε, το ποσό της εταιρίας δεν αυξήθηκε, αλλά μειώθηκε. Αυτό συμβαίνει, καθώς τόσο τα χρήματα του πελάτη, όσο και το αντίστοιχο ποσό από τη μεριά της εταιρίας μεταφέρθηκαν στο υπογεγραμμένο συμβόλαιο (16 ether), προκειμένου να μπορέσουν να χρησιμοποιηθούν για την αποζημίωση του πελάτη, εφόσον πληρούνται οι όροι του συμβολαίου. Τα χρήματα αυτά θα μπορέσει η εταιρία να τα πάρει πίσω, μετά τη λήξη του συμβολαίου, εφόσον ο πελάτης δεν παρουσίασε κάποιο πρόβλημα στη διάρκεια ισχύος του συμβολαίου που θα του επέτρεπε να λάβει τα χρήματα αυτά.



Σχήμα 16: Γραφικό Περιβάλλον και Υπόλοιπο Λογαριασμών μετά την Υπογραφή ενός Συμβολαίου

Τα δεδομένα των χρηστών καταλήγουν σε μια Σχεσιακή Βάση που υλοποιήσαμε σε MySQL[22], μέσω ενός JAVA Rest Service που αναπτύξαμε και το οποίο κάναμε deploy στον Apache Tomcat[23] Application Server που βάλαμε στο VM μας στο cloud. Το service αυτό κατά την επικοινωνία του με το Blockchain χρησιμοποιεί τον δεύτερο λογαριασμό, που αντιστοιχεί στον λογαριασμό του κέντρου περίθαλψης ασθενών, καθώς το αντίστοιχο smart contract είχε παραμετροποιηθεί κατά το deployment, ώστε να δέχεται εντολές μόνο από τον δημιουργό του. Το δεύτερο Web Service, που εξετάζει εάν ο ασθενής ικανοποιεί τους όρους του συμβολαίου, υλοποιήθηκε επίσης σε Java. Το service αυτό παίρνει ως input μια λίστα με τους δείκτες που αφορούν τα δεδομένα ενός ασθενή και τα κριτήρια καταλληλότητας που εκφράζουν τους όρους του συμβολαίου και εξετάζει κατά πόσο αυτοί ικανοποιούνται, λαμβάνοντας υπόψη την κατηγοριοποίηση των όρων. Στο server αυτό ανεβάσαμε επίσης το Γραφικό Περιβάλλον, ώστε να μπορούμε ακολούθως να το κατεβάσουμε και να το χρησιμοποιήσουμε για την επικοινωνία μας με το Blockchain.

Σημειώνουμε ότι ο κώδικας του συστήματος του Κέντρου Περίθαλψης Ασθενών έγινε update, έτσι ώστε να στέλνονται τα αντίστοιχα δεδομένα και στην πλατφόρμα μας, καλώντας σύγχρονα το πρώτο service. Για τις ανάγκες της εργασίας αυτής υλοποιήσαμε ένα Desktop Application, που ενημερώνει το σύστημα, καταγράφοντας ότι ο ασθενής διαγνώστηκε με ορισμένα προβλήματα, για να υποστηρίξουμε το σενάριο που θέλαμε να παρουσιάσουμε. Για τον έλεγχο των συνθηκών και ειδικότερα για την ασύγχρονη επικοινωνία των smart contracts με το δεύτερο web service χρησιμοποιήθηκε το λογισμικό πακέτο Provable[24] (διάδοχος του Oraclize) κατά την συγγραφή του σχετικού smart contract, καθώς αυτοματοποιεί τη διαδικασία αυτή, περιορίζοντας την έκταση του κώδικα και βελτιώνοντας την ασφάλεια του συστήματος.

[22] MySQL, https://www.mysql.com/
[23] Apache Tomcat, http://tomcat.apache.org/
[24] Provable, https://provable.xyz/

Η σελίδα αυτή είναι σκόπιμα λευκή

6 State of the Art

6.1 Εφαρμογές βασισμένες σε τεχνολογίες Blockchain

6.1.1 Διαχείριση Προσωπικών Δεδομένων

Ιδιωτικοί και δημόσιοι οργανισμοί (π.χ., Μικροβιολογικά Εργαστήρια) συγκεντρώνουν σημαντικό όγκο προσωπικών δεδομένων (συμπεριλαμβανομένων ευαίσθητων προσωπικών δεδομένων), για τα οποία οι ίδιοι οι άνθρωποι έχουν ελάχιστη ή καθόλου γνώση και έλεγχο για τα δεδομένα που αποθηκεύονται και πώς αυτά χρησιμοποιούνται. Στο έγγραφο αυτό [7] οι συγγραφείς παρουσιάζουν μια προσέγγιση βασισμένη στις τεχνολογίες του Blockchain, η οποία επιτρέπει στους ανθρώπους να έχουν πλήρη έλεγχο στα δεδομένα που τους αφορούν και να μπορούν να επιτρέπουν σε τρίτους να έχουν πρόσβαση σε αυτά. Για τον σκοπό αυτό το Blockchain χρησιμοποιείται για τον έλεγχο της πρόσβασης στα δεδομένα των χρηστών, τα οποία, όμως, αποθηκεύονται εκτός του Blockchain (off Blockchain). Ειδικότερα, τα δεδομένα των χρηστών κρυπτογραφούνται, χρησιμοποιώντας την κρυπτογραφία συμμετρικού κλειδιού και ακολούθως αποθηκεύονται διάσπαρτα στους κόμβους ενός peer-to-peer συστήματος (οι κόμβοι του οποίου μπορεί να είναι διαφορετικοί από τους κόμβους του Blockchain δικτύου), ενώ στο Blockchain αποθηκεύεται μόνο ένας δείκτης (το hash των δεδομένων) προς αυτά. Ακολούθως, μια τρίτη οντότητα (π.χ., κάποιος οργανισμός/υπηρεσία), που θα θέλει να έχει πρόσβαση σε αυτά, θα πρέπει πρώτα να συμβουλευτεί (να ρωτήσει) το Blockchain, έτσι ώστε να δει, εάν επιτρέπεται να έχει πρόσβαση στα δεδομένα του χρήστη και ακολούθως να εντοπίσει το «link» προς αυτά. Με αυτόν τον τρόπο ο χρήστης έχει απόλυτο έλεγχο τόσο στα δεδομένα που αποθηκεύονται, όσο και στις οντότητες που έχουν πρόσβαση σε αυτά, ενώ μπορεί να αλλάξει την πολιτική του ανά πάσα στιγμή.

Από άποψη ιδιωτικότητας και ασφάλειας, το γεγονός ότι η πολιτική που θα ακολουθήσουμε αποθηκεύεται στο Blockchain, μας διασφαλίζει από πιθανές επιθέσεις που μπορεί να λάβουν χώρα, με απώτερο σκοπό να εξασφαλίσουν οι επίδοξοι χρήστες πρόσβαση στα δεδομένα των χρηστών. Επίσης, τα δεδομένα των ανθρώπων

αποθηκεύονται εκτός της αλυσίδας, ενώ στις συναλλαγές που υπάρχουν στους ελευθέρους προσβάσιμους κόμβους του Blockchain υπάρχουν μόνο δείκτες προς αυτά. Απ' την άλλη, η προσέγγιση αυτή υποφέρει από τις αδυναμίες ενός Blockchain συστήματος, οι οποίες έχουν να κάνουν με το τύπο του Blockchain που θα χρησιμοποιηθεί, το πλήθος των κόμβων που συμμετέχουν στη διαδικασία ενημέρωσης της αλυσίδας και τον αλγόριθμο συναίνεσης που χρησιμοποιείται, ο οποίος επηρεάζει άμεσα και τον χρόνο που απαιτείται για την προσθήκη ενός νέου μπλοκ στην αλυσίδα.

Σχετικά με τα δεδομένα των ασθενών, το γεγονός ότι αποθηκεύονται διάσπαρτα μέσα στο σύστημα (εκτός της αλυσίδας) σε συνδυασμό με τη χρήση της κρυπτογραφίας συμμετρικού κλειδιού (διαφορετικού από του δημοσίου ή ιδιωτικού κλειδιού των χρηστών) εμποδίζει πιθανά κακόβουλα λογισμικά που θα ήθελαν να εντοπίσουν τα δεδομένα των χρηστών ή να έχουν πρόσβαση σε αυτά. Αυτό γίνεται επίσης κατανοητό, εάν λάβουμε υπόψη ότι ο χρήστης διατηρεί την ανωνυμία του στο σύστημα και μάλιστα για κάθε οντότητα που θα ήθελε να έχει πρόσβαση στα δεδομένα χρησιμοποιείται διαφορετικό αναγνωριστικό. Κατά συνέπεια, ακόμη και εάν καταφέρουν να οργανώσουν τα δεδομένα ενός χρήστη (μέσω της ανάλυσης των δεδομένων του Blockchain) και να τα αποκρυπτογραφήσουν, θα έχουν πρόσβαση μόνο σε ένα μέρος των δεδομένων των ανθρώπων. Βέβαια, αξίζει να αναφέρουμε ότι η προσέγγιση που ακολουθήθηκε έχει και κάποιες αδυναμίες. Για παράδειγμα, το γεγονός ότι ένα service μπορεί να έχει πρόσβαση σε ένα μέρος των δεδομένων των ασθενών, δεν μας εξασφαλίζει ότι το service αυτό δεν θα τα αποθηκεύσει τοπικά τα δεδομένα για μελλοντική χρήση. Μια καλύτερη προσέγγιση ίσως ήταν να μην επιτρέπεται στα services να έχουν πρόσβαση στα δεδομένα, αλλά να τους δίνουμε, όμως, τη δυνατότητα να εκτελέσουν ερωτήματα και να λάβουν μόνο τις απαντήσεις.

6.1.2 Συγκέντρωση και Διαχείριση Δεδομένων Ασθενών

Οι ασθενείς κατά τη διάρκεια της ζωής τους αφήνουν τα δεδομένα τους διάσπαρτα σε διάφορους οργανισμούς και ιδρύματα. Η συνολική εξέταση των ιστορικών δεδομένων των ασθενών θα μπορούσε να είναι επωφελής τόσο για τον ίδιο τον ασθενή (π.χ., παροχή καλύτερων υπηρεσιών υγείας), όσο και για την έρευνα γύρω από τους παράγοντες με τους οποίους σχετίζεται ο κάθε ασθενής (π.χ., καλύτερη μελέτη της επίδρασης των φαρμάκων) και κατ' επέκταση τη βελτίωση του συστήματος υγείας. Στο έγγραφο [8] οι συγγραφείς

παρουσιάζουν ένα σύστημα (MedRec) για την οργάνωση των δεδομένων των ασθενών και τον έλεγχο της πρόσβασης σε αυτά, λαμβάνοντας υπόψη τις τεχνολογίες Blockchain. Η δημιουργία του συστήματος αυτού έχει εμφανώς επηρεαστεί από την εργασία [7] και υλοποιήθηκε χρησιμοποιώντας τεχνολογίες αιχμής, όπως είναι το Ethereum και τα Smart Contracts. Ειδικότερα, για τις ανάγκες της εργασίας αυτής οι συγγραφείς υλοποίησαν τρία διαφορετικά έξυπνα συμβόλαια, τα οποία επιτρέπουν στον χρήστη να καταγράψει το αναγνωριστικό του (όπως είναι το όνομά του ή τον αριθμό κοινωνικής ασφάλισης), τα δεδομένα που του ανήκουν και τους χρήστες/οντότητες που μπορεί να έχουν πρόσβαση σε αυτά.

Τα δεδομένα των χρηστών που συλλέγονται από τους διάφορους οργανισμούς διατηρούνται σε ξεχωριστές βάσεις δεδομένων (π.χ., μια σχεσιακή βάση προσβάσιμη μέσω SQL[25]), ενώ στο Blockchain υπάρχει μόνο αναφορά στα δεδομένα που υπάρχουν σε αυτές καθώς και οι χρήστες (οι διευθύνσεις τους) που μπορούν να έχουν πρόσβαση σε αυτά. Πιο συγκεκριμένα, στο έξυπνο συμβόλαιο καταγράφεται για παράδειγμα το SQL ερώτημα που θα φέρει τα δεδομένα του ασθενούς, μία σύνοψη αυτών (για λόγους ασφάλειας) καθώς και τα στοιχεία επικοινωνίας με την αντίστοιχη βάση, προκειμένου να μπορούν ακολούθως να αξιοποιηθούν για τον έλεγχο της πρόσβασης σε αυτά. Επίσης, για κάθε οργανισμό (διεύθυνση) καταγράφονται τα δεδομένα στα οποία μπορούν να έχουν πρόσβαση μέσω του αντίστοιχου ερωτήματος (π.χ., SQL). Συνεπώς, ο κάθε χρήστης μπορεί να ελέγξει άμεσα τα δεδομένα που χρησιμοποιούνται από κάθε ινστιτούτο και να αλλάξει / περιορίσει την πρόσβαση σε αυτά. Σημειώνουμε ότι τα παραπάνω συμβόλαια είναι αποθηκευμένα στο Blockchain και μπορούν να εκτελεστούν από την πλατφόρμα του Ethereum.

Για την αποτελεσματική διαχείριση των δεδομένων των ασθενών (π.χ., έλεγχος αποθηκευμένων δεδομένων και οντοτήτων που έχουν πρόσβαση σε αυτά) έχουν υλοποιηθεί επιπρόσθετα κάποια προγράμματα, τα οποία επιτρέπουν στον χρήστη να επικοινωνήσει με την πλατφόρμα του Ethereum, είτε για να ορίσει την πολιτική που επιθυμεί (ως κάτοχος), είτε για να εξετάσει, εάν μπορεί να έχει πρόσβαση στα δεδομένα του ασθενή (ως χρήστης) και πώς. Ειδικότερα, για την επικοινωνία με τις βάσεις των ασθενών, υπάρχουν τοπικοί server, οι οποίοι συμβουλεύονται πρώτα το Blockchain και ακολούθως εκτελούν τα αντίστοιχα SQL ερωτήματα, εφόσον κάτι τέτοιο επιτρέπεται.

[25] Structured Query Language (SQL), https://en.wikipedia.org/wiki/SQL

Σημειώνουμε ότι στην τρέχουσα έκδοση του συστήματος, τα δεδομένα που αποθηκεύονται στο Blockchain δεν είναι κρυπτογραφημένα, αλλά κάτι τέτοιο μπορεί να γίνει σε επόμενες εκδόσεις του συστήματος. Επίσης, στην εργασία αυτή έμφαση δίνεται στη συγκέντρωση και διαχείριση των δεδομένων των χρηστών, τα οποία μπορούν να εξακολουθούν να βρίσκονται στις παραδοσιακές βάσεις ή συστήματα. Ωστόσο, αυτά θα πρέπει να συνοδεύονται από αντίστοιχους μηχανισμούς ασφάλειας.

Το σύστημα που υλοποιήθηκε υποφέρει από τις γνωστές αδυναμίες του Blockchain και των Smart Contracts καθώς επίσης και τις αδυναμίες των υπαρχόντων συστημάτων (π.χ., εξυπηρέτηση ερωτημάτων από έναν εξυπηρετητή, ασφάλεια βάσεων δεδομένων, κτλ.). Στο σύστημα αυτό έμφαση δίνεται στη σύνδεση διαφορετικών πηγών δεδομένων, υπό την έννοια ότι ο χρήστης θα μπορεί να εντοπίσει τις πηγές και τα δεδομένα που αποθηκεύονται σε αυτές. Ωστόσο, ο διαφορετικός τρόπος αναπαράστασης των δεδομένων αυτών (μοντέλα, κωδικοποιήσεις) αποτελεί ένα πολύ σημαντικό παράγοντα που περιορίζει σημαντικά την επικοινωνία των πληροφοριακών συστημάτων με τις πηγές δεδομένων.

6.1.3 Απομακρυσμένη Παρακολούθηση Ασθενών

Η απομακρυσμένη παρακολούθηση των ασθενών κερδίζει συνεχώς έδαφος. Για τον σκοπό αυτό ο ασθενής είναι εξοπλισμένος με φορητές (wearable) ή εμφυτευμένες (implanted) ηλεκτρονικές συσκευές, οι οποίες παρακολουθούν σε πραγματικό χρόνο την κατάστασή του και αναφέρουν τα αποτελέσματα σε κάποια κύρια συσκευή (master device), όπως για παράδειγμα ένα κινητό τηλέφωνο, ενώ μπορούν επίσης να προβούν και στις απαραίτητες ενέργειες με απώτερο στόχο την εύρυθμη λειτουργία του οργανισμού. Στο έγγραφο [22] οι συγγραφείς παρουσιάζουν ένα σύστημα, το οποίο βασίζεται σε τεχνολογίες Blockchain, για την αποτελεσματική διαχείριση των δεδομένων των ασθενών που προέρχονται από τους αισθητήρες (sensors) και την άμεση λήψη των απαραίτητων ενεργειών, όταν αυτό απαιτείται. Για τον σκοπό αυτό, τα δεδομένα των ασθενών αποθηκεύονται σε ξεχωριστές βάσεις δεδομένων, ενώ στο Blockchain καταγράφονται μόνο τα γεγονότα που λαμβάνουν χώρα και όχι τα ευαίσθητα προσωπικά δεδομένα των ασθενών, έτσι ώστε να είναι συμβατό με όσα προβλέπει ο Νόμος περί φορητότητας και λογοδοσίας για την ασφάλιση υγείας (HIPAA).

Για τις ανάγκες του συστήματος αυτού έχει χρησιμοποιηθεί ένα ιδιωτικό Blockchain, στο οποίο η ενημέρωση της αλυσίδας μπορεί να γίνει από επιλεγμένους κόμβους. Επίσης, έχουν υλοποιηθεί δύο διαφορετικά έξυπνα συμβόλαια, τα οποία είναι γραμμένα στην γλώσσα Solidity και μπορούν να εκτελεστούν χρησιμοποιώντας την πλατφόρμα του Ethereum. Το πρώτο συμβόλαιο λειτουργεί σαν σημείο επικοινωνίας της κύριας συσκευής με το Blockchain και είναι κοινό για όλους τους χρήστες. Αυτό με τη σειρά του αναλαμβάνει να δημιουργήσει τα απαραίτητα έξυπνα συμβόλαια (factory pattern), ανάλογα με τον τύπο των δεδομένων του χρήστη. Τα έξυπνα αυτά συμβόλαια είναι υπεύθυνα για τον έλεγχο της κατάστασης του χρήστη και τη λήψη των απαραίτητων μέτρων. Δεδομένου ότι η λειτουργία των έξυπνων συμβολαίων βασίζεται στα δεδομένα που υπάρχουν ήδη στην αλυσίδα, τα δεδομένα των χρηστών που καταφτάνουν από τους αισθητήρες παρέχονται (από ένα service που «τρέχει» στην κύρια συσκευή) άμεσα στις μεθόδους του αντίστοιχου συμβολαίου, για να ελεγχθούν (συμπεριλαμβανομένων των φυσιολογικών τους τιμών). Στην περίπτωση που η τιμή τους είναι μη φυσιολογική, το γεγονός αυτό καταγράφεται στο Blockchain και ακολούθως ενημερώνονται οι κατάλληλοι άνθρωποι. Στην περίπτωση στην οποία είναι απαραίτητη η λήψη κάποιων μέτρων, είτε μετά την άμεση ανάλυση των δεδομένων είτε κατόπιν επικοινωνίας με τους ειδικούς, το γεγονός αυτό καταγράφεται επίσης στο σύστημα. Τα δεδομένα των χρηστών οδηγούνται σε ξεχωριστή βάση για την αποθήκευση των δεδομένων των ασθενών.

Η παραπάνω προσέγγιση επιτρέπει την ασφαλή επεξεργασία των δεδομένων των χρηστών και την άμεση λήψη των απαραίτητων ενεργειών, οι οποίες καταγράφονται σε ένα «βιβλίο» το οποίο δεν μπορεί να αλλαχθεί (immutable). Για τον σκοπό αυτό, οι χρήστες επικοινωνούν με το Blockchain από το κινητό τους (από το κατάλληλο λογισμικό) χρησιμοποιώντας τη διεύθυνσή τους, διατηρώντας έτσι την ανωνυμία τους. Επίσης, στο Blockchain παρέχονται μόνο τα εντελώς απαραίτητα δεδομένα για τις ανάγκες του ελέγχου, ενώ τα δεδομένα των χρηστών οδηγούνται σε ξεχωριστή βάση εκτός της αλυσίδας. Συνεπώς, στο Blockchain αποθηκεύονται κυρίως πληροφορίες για τα γεγονότα που έχουν λάβει χώρα από το αντίστοιχο συμβόλαιο.

Στο σημείο αυτό, να αναφέρουμε ότι η προσέγγιση που ακολουθείται για τη δημιουργία των συμβολαίων, που θα επεξεργαστούν τα δεδομένα από ένα και μόνο συμβόλαιο, δίνει τη δυνατότητα για τον αποτελεσματικό χειρισμό των αλλαγών από τον

διαχειριστή του συστήματος (π.χ., «αντικατάσταση» ενός συμβολαίου με ένα άλλο), εάν αυτό κριθεί απαραίτητο.

6.2 Αδυναμίες του Blockchain και των Smart Contracts

6.2.1 Κίνδυνοι Ασφαλείας του Blockchain

Στα παραπάνω συστήματα (συμπεριλαμβανομένου του συστήματος που εμείς υλοποιήσαμε), οι συναλλαγές που έχουν πραγματοποιηθεί (είτε μεταφορά χρημάτων μεταξύ δυο χρηστών, στην περίπτωση του Bitcoin είτε καταγραφή της αλληλεπίδρασης μεταξύ των χρηστών και των smart contracts, στην περίπτωση του Ethereum) καταγράφονται στους κόμβους (μπλοκ) της αλυσίδας (Blockchain). Επίσης, όλοι οι χρήστες του δικτύου μπορούν να διατηρούν ένα αντίγραφο της αλυσίδας αυτής και κατ' επέκταση να έχουν πρόσβαση στην πληροφορία που είναι αποθηκευμένη, καθώς επίσης και να συμμετέχουν στη διαδικασία επέκτασης της αλυσίδας με νέους κόμβους και συναλλαγές. Δεδομένης της απουσίας μιας κεντρικής αρχής ελέγχου του συστήματος και των συναλλαγών που λαμβάνουν χώρα, ή εμπιστοσύνη του χρήστη προς το σύστημα πηγάζει από την αρχιτεκτονική του συστήματος και τους αλγορίθμους συναίνεσης που χρησιμοποιούνται για την επικοινωνία μεταξύ των χρηστών και την αποδοχή νέων συναλλαγών. Οι αλγόριθμοι αυτοί μας εξασφαλίζουν ότι οι συναλλαγές που πραγματοποιούμε καταγράφονται στο Blockchain και δεν μπορούν να αλλαχτούν (immutable), και ότι τα χρήματα των χρηστών παραμένουν ασφαλή μέσω της χρήση κρυπτογραφίας δημόσιου κλειδιού και δεν μπορούν να ξοδευτούν δύο φορές (double-spending problem). Ωστόσο, υπό ορισμένες συνθήκες, τα παραπάνω μπορούν να παρακαμφθούν, θέτοντας σε κίνδυνο όλο το σύστημα και μειώνοντας την αξιοπιστία των χρηστών προς αυτό.

Στην εργασία [23] παρουσιάζονται ορισμένες καταστάσεις, οι οποίες μπορούν να θέσουν σε κίνδυνο την αλυσίδα (Blockchain) και τα δεδομένα που αυτή περιέχει. Στον Πίνακας 3 παρουσιάζονται συνοπτικά οι κίνδυνοι που υπάρχουν.

No.	**Κίνδυνος Ασφαλείας**	**Παράγοντας που Ευθύνεται**
1	Κατοχή του 51%	Αλγόριθμος Συναίνεσης
2	Απώλεια Ιδιωτικού Κλειδιού	Μηχανισμός Παραγωγής και Διατήρησης Κλειδιών

3	Διπλή Δαπάνη (Double Spending)	Αλγόριθμος Συναίνεσης / Πιστοποίηση Συναλλαγών
4	Διαρροή Δεδομένων	Σχεδιασμός Συστήματος / Blockchain
5	Χρήση για Παράνομες Δραστηριότητες	Γενικότερη Λειτουργία του Συστήματος

Πίνακας 3: Κίνδυνοι Ασφαλείας του Blockchain

Ο πρώτος κίνδυνος σχετίζεται άμεσα με τον αλγόριθμο συναίνεσης που χρησιμοποιείται. Δύο από τους πλέον γνωστούς αλγορίθμους συναίνεσης είναι οι αλγόριθμοι Proof of Work (PoW) και Proof of Stake (PoS). Στην πρώτη περίπτωση (PoW), εάν ένας κόμβος (ή συνεργασία αυτών) κατέχει το 51% της υπολογιστικής ισχύος του δικτύου, μπορεί να επιφέρει ανεπιθύμητες αλλαγές στην αλυσίδα ή ακόμη και γενικότερα να επηρεάσει την ομαλή λειτουργία του συστήματος. Στην δεύτερη περίπτωση (PoS), εάν ένας κόμβος κατέχει το 51% των χρημάτων του δικτύου, μπορεί επίσης να βλάψει το Blockchain.

Ο δεύτερος κίνδυνος έχει να κάνει με το ιδιωτικό κλειδί και τον τρόπο που αυτό παράγεται και αποθηκεύεται. Το ιδιωτικό κλειδί παράγεται από τον κάθε χρήστη και είναι απαραίτητο για την πραγματοποίηση των συναλλαγών. Απώλειά του σημαίνει ότι ο χρήστης χάνει τον έλεγχο του λογαριασμού και των χρημάτων που αυτός περιέχει. Όταν δεν έχει γίνει χρήση αρκετής τυχαιότητας κατά τη δημιουργία του, κάποιος κακόβουλος χρήστης ή λογισμικό μπορεί να καταφέρει να βρει το ιδιωτικό κλειδί από τα δεδομένα που υπάρχουν ήδη στην αλυσίδα (διεύθυνση, ψηφιακή υπογραφή), λόγω μιας αδυναμίας του αλγόριθμου που χρησιμοποιείται κατά την παραγωγή του.

Ο τρίτος κίνδυνος έχει να κάνει με τον τρόπο με τον οποίο προστίθενται οι συναλλαγές στους κόμβους της αλυσίδας και ειδικότερα με τον αλγόριθμο συναίνεσης. Όπως έχουμε ήδη αναφέρει, το Blockchain (τόσο στην περίπτωση του Bitcoin, όσο και του Ethereum) δεν μας επιτρέπει να ξοδέψουμε τα χρήματα μας δύο φορές, καθώς όλοι οι κόμβοι του συστήματος έχουν πρόσβαση σε αυτό και επομένως μπορούν να αντιληφθούν μια τέτοια προσπάθεια. Όμως, στην αλυσίδα υπάρχουν οι συναλλαγές που έχουν ήδη εγκριθεί. Δεδομένου ότι μεσολαβεί κάποιος σημαντικός χρόνος για την ενημέρωση της αλυσίδας (ειδικά στην περίπτωση του PoW), αυτό δίνει τη δυνατότητα σε κακόβουλους χρήστες να ξοδέψουν στο μεσοδιάστημα τα χρήματά τους δύο φορές.

Ο τέταρτος κίνδυνος πηγάζει από το γεγονός ότι τα δεδομένα της αλυσίδας είναι ευρέως διαθέσιμα σε όλους τους κόμβους του δικτύου. Κατ' επέκταση κάποιος θα μπορούσε να αναλύσει τα δεδομένα αυτά και να εντοπίσει τα χρήματα ή τα δεδομένα καθώς και τις διευθύνσεις των χρηστών οι οποίοι εμπλέκονται και ακόμη να καταφέρει να φτάσει σε αυτούς. Η χρήση περισσότερων του ενός λογαριασμών από τους ίδιους τους χρήστες μπορεί κάπως να περιορίσει τον κίνδυνο. Ωστόσο, αυτός ο κίνδυνος είναι υπαρκτός.

Τέλος, αναφέρουμε ότι, δεδομένης της ανωνυμίας που υπάρχει μεταξύ των χρηστών, το σύστημα αυτό μπορεί να χρησιμοποιηθεί για την πραγματοποίηση παράνομων συναλλαγών, όπως για παράδειγμα, για να ζητήσουμε την καταβολή λύτρων, όπως στην περίπτωση μιας Ransomware[26] επίθεσης ή γενικότερα να χρησιμοποιηθεί για την πραγματοποίηση παράνομων δραστηριοτήτων.

6.2.2 Τρωτά Σημεία των Έξυπνων Συμβολαίων (Smart Contracts)

Τα Ethereum Smart Contracts είναι προγράμματα, τα οποία μπορούν να εκτελεστούν από τους κόμβους της πλατφόρμας του Ethereum και να αλλάξουν την κατάσταση του συστήματος. Τα προγράμματα αυτά είναι γραμμένα σε μια γλώσσα χαμηλού επιπέδου, γνωστή ως Bytecode, η οποία είναι μία Turing-complete γλώσσα προγραμματισμού, που υποστηρίζεται από την Εικονική Μηχανή της πλατφόρμας του Ethereum. Ωστόσο, τα έξυπνα συμβόλαια είναι συνήθως γραμμένα σε μια γλώσσα υψηλού επιπέδου, όπως είναι η Solidity (περισσότερες πληροφορίες υπάρχουν στο παράρτημα, στο Κεφάλαιο 8.3) και ο «εκτελέσιμος» από την εικονική μηχανή της Ethereum πλατφόρμας Bytecode, προκύπτει κατά τη διάρκεια της διαδικασίας της μεταγλώττισης. Το γεγονός ότι η γλώσσα Solidity είναι μια αρκετά εκφραστική γλώσσα (π.χ., υποστηρίζει δομές διακλάδωσης και επανάληψης), δίνει τη δυνατότητα στους προγραμματιστές να πληκτρολογήσουν αρκετά περίπλοκα συμβόλαια, προκειμένου να καλύψουν τις ανάγκες τους. Επίσης, οι προγραμματιστές έχουν τη δυνατότητα να δημιουργήσουν άλλα συμβόλαια, τα οποία μπορούν να τα καλέσουν, ασχέτως εάν αυτά τα είχαν δημιουργήσει οι ίδιοι ή κάποιος άλλος προγραμματιστής. Η ελευθερία που παρέχεται μέσω της γλώσσα αυτής κατά την έκφραση των συμβολαίων μπορεί να βάλει σε κίνδυνο τους λογαριασμούς των χρηστών του συστήματος και ειδικότερα αυτούς που έρχονται σε άμεση επικοινωνία με τις

[26] Ransomware, https://en.wikipedia.org/wiki/Ransomware

εφαρμογές που αναπτύσσονται. Στην ενότητα αυτή θα μελετήσουμε τα προβλήματα που έχουν να κάνουν με την ασφάλεια των έξυπνων συμβολαίων. Αυτά, συνήθως, πηγάζουν από τη "λάθος" από τη μεριά του προγραμματιστή αντίληψη του τρόπου λειτουργίας των επιμέρους εντολών των smart contracts, λόγω των ιδιαιτεροτήτων που παρουσιάζει τόσο η γλώσσα αυτή, όσο και ο τρόπος εκτέλεσης των έξυπνων συμβολαίων από την εικονική μηχανή της πλατφόρμας του Ethereum.

Οι συγγραφείς του έργου [24] έχουν οργανώσει τις αδυναμίες ασφαλείας των smart contracts σε τρεις ευρύτερες κατηγορίες, ανάλογα με το επίπεδο στο οποίο αυτές εντοπίζονται (Πίνακας 4).

Κατηγορία	**Αδυναμία Ασφάλειας**
Solidity	Κλήση στο Άγνωστο
	Εσφαλμένη Διαχείριση Ειδικών Καταστάσεων (Exceptions)
	Έλλειψη Καυσίμων (Gas)
	Προσαρμογή Τύπων
	Επανάκληση Μεθόδου
	Διαχείριση Μυστικών/Ιδιωτικών Δεδομένων
Εκτέλεση Bytecode από την Εικονική Μηχανή (EVM)	Χάσιμο Χρημάτων
	Ξεπέρασμα ορίου στοίβας (εσωτερικής δομής δεδομένων)
	Διαχείριση Λαθών (Bugs)
Ενημέρωση Blockchain (Mining Process)	Απρόβλεπτες Καταστάσεις
	Χρήση τυχαιότητας
	Διαχείριση Χρονικών Περιορισμών

Πίνακας 4: Αδυναμίες Ασφάλειας Έξυπνων Συμβολαίων

Στην πρώτη κατηγορία ανήκουν οι περιπτώσεις εκείνες που αφορούν την έκφραση του συμβολαίου στη γλώσσα Solidity. Η γλώσσα αυτή παρέχει τρεις διαφορετικούς τρόπους με τους οποίους μπορούμε να καλέσουμε ένα άλλο συμβόλαιο, καθένας απ' τους οποίους συμπεριφέρεται λίγο διαφορετικά, όσον αφορά τη διαχείριση κάποιων ειδικών καταστάσεων (exceptions). Επίσης, για την εκτέλεση των προγραμμάτων απαιτούνται

χρήματα/καύσιμα, τα οποία, εάν δεν είναι αρκετά, κάνουν την εκτέλεση του προγράμματος να αποτύχει. Επιπλέον, ο κακός σχεδιασμός μιας μεθόδου μπορεί να οδηγήσει στον εσφαλμένο τρόπο λειτουργίας του συστήματος σε περίπτωση επανάκλησης αυτής. Τέλος, αναφέρουμε ότι τα δεδομένα που αποθηκεύονται στα συμβόλαια, ακόμη και εάν είναι ιδιωτικά και πρέπει να μείνουν κρυφά (έστω και για κάποιο μικρό χρονικό διάστημα), είναι δυνατόν να τα εντοπίσουμε.

Στη δεύτερη κατηγορία ανήκουν οι περιπτώσεις εκείνες που έχουν να κάνουν με την εκτέλεση του Bytecode που προκύπτει κατά τη διαδικασία της μεταγλώττισης από την Εικονική Μηχανή του Ethereum. Ο κώδικας αυτός, κατά την αποστολή χρημάτων σε έναν λογαριασμό, δεν ελέγχει εάν ο αποδέκτης, όντως, υπάρχει (είναι π.χ., κάποιος χρήστης), με αποτέλεσμα να κινδυνεύουν τα χρήματα να χαθούν. Επίσης, η εσωτερική στοίβα που χρησιμοποιείται για την εκτέλεση των προγραμμάτων έχει περιορισμένο μέγεθος, το οποίο, εάν το ξεπεράσουμε, κάνει τον κώδικά μας να αποτύχει. Σημειώνουμε ότι, στην περίπτωση που εντοπίσουμε μια αδυναμία στον κώδικα, αυτό είναι μερικές φορές δύσκολο να το διορθώσουμε, καθώς αυτός αποθηκεύεται στην αλυσίδα και δεν μπορεί εύκολα να αλλάξει (ειδικά, εάν δεν υπάρχει απ' την δική μας τη μεριά ο κατάλληλος σχεδιασμός).

Στην τρίτη κατηγορία ανήκουν οι αδυναμίες που έχουν να κάνουν με την ενημέρωση της αλυσίδας (Blockchain). Δεδομένου ότι η χρονική στιγμή κατά την οποία καλούμε μια συνάρτηση και η χρονική στιγμή που αυτή εκτελείται διαφέρει (λόγω της οργάνωσης των συναλλαγών και ακολούθως την προσθήκη τους στην αλυσίδα από τους κόμβους εξόρυξης) δεν μας επιτρέπει να γνωρίζουμε την ακριβή κατάσταση του συστήματος, τη στιγμή της εκτέλεσης, το οποίο μπορεί να οδηγήσει σε προβλήματα. Επίσης, για τη διαχείριση της τυχαιότητας και τον υπολογισμό των χρονικών διαστημάτων, πολλές φορές χρησιμοποιούμε το timestamp του τελευταίου μπλοκ της αλυσίδας, το οποίο, όμως, μπορεί να επηρεαστεί από κακόβουλα λογισμικά που προσπαθούν να επιφέρουν αλλαγές σε αυτήν.

Στο σημείο αυτό θα πρέπει να αναφέρουμε ότι τα παραπάνω αποτελούν μία πολύ σύντομη περιγραφή (και χωρίς να μπούμε σε μεγάλο βάθος εξετάζοντας τον κώδικα συγκεκριμένων συμβολαίων) των αδυναμιών που πολλές φορές συναντάμε σε υπάρχοντα smart contracts, οι οποίες (είτε μεμονωμένα, είτε συνδυασμός αυτών) μπορούν να οδηγήσουν πιθανούς κακόβουλους χρήστες στην πραγματοποίηση επιθέσεων στα

αντίστοιχα συμβόλαια και εφαρμογές (π.χ., δημιουργία ενός νέου κακοήθους συμβολαίου που τα καλεί, προσπαθώντας να εκμεταλλευτεί τις αδυναμίες τους), με απώτερο σκοπό να κλέψουν εικονικά νομίσματα από τους χρήστες που εμπλέκονται ή πληροφορίες που υπάρχουν αποθηκευμένες στο σύστημα. Για αυτόν τον λόγο προτείνουμε στους αναγνώστες να διαβάσουν το παραπάνω έγγραφο [24], στο οποίο υπάρχουν και συγκεκριμένα παραδείγματα (συμβόλαια στη γλώσσα Solidity) καθώς και αναφορές σε γνωστές επιθέσεις (π.χ., DAO attack).

6.2.3 Περαιτέρω Συζήτηση – Εισαγωγή στο Hyperledger Fabric

6.2.3.1 Το Μοντέλο Ταξινόμησης-Εκτέλεσης (Order-Execute Architecture)

Τα συστήματα που εξετάσαμε στις ενότητες 2 (Bitcoin) και 3 (Ethereum) βασίζονται στο *μοντέλο ταξινόμησης-εκτέλεσης (order-execute architecture)* για τη διαχείριση των συναλλαγών [25]. Σύμφωνα με το μοντέλο αυτό, οι εργασίες ταξινομούνται και ακολούθως εκτελούνται σειριακά, η μία μετά την άλλη, ενώ η δημιουργία του μπλοκ που θα προστεθεί στην αλυσίδα (συμπεριλαμβανομένων της δημιουργίας της επικεφαλίδας και του υπολογισμού των επιμέρους πεδίων, όπως τον υπολογισμό της τιμή nonce / αποτέλεσμα του αλγορίθμου Proof of Work) έπεται της διαδικασίας αυτής. Κατά συνέπεια, η σειριακή εκτέλεση των επιμέρους συναρτήσεων των smart contracts επηρεάζεται άμεσα από την εκτέλεση των υπολοίπων συναλλαγών. Αυτό γίνεται εύκολα αντιληπτό, εάν αναλογιστούμε την ύπαρξη ενός κακοσχεδιασμένου smart contract, το οποίο «μπαίνει» μέσα σε ένα ατέρμονα βρόχο και κατά συνέπεια δεν τερματίζει ποτέ. Στην περίπτωση του Ethereum, το πρόβλημα αυτό αντιμετωπίζεται μέσω της λεπτομερούς κοστολόγησης των επιμέρους εντολών, έτσι ώστε η συνεχής εκτέλεση ενός προγράμματος, σαν αυτό που αναφέρθηκε, να διακόπτεται αυτόματα, λόγω της εξάντλησης των διαθέσιμων χρημάτων (gas). Με αυτόν τον τρόπο το σύστημα προστατεύεται από τέτοιου είδους προβλήματα (ηθελημένα ή μη). Ωστόσο, η προσέγγιση αυτή δεν παύει να αυξάνει σημαντικά τον συνολικό χρόνο που απαιτείται για την εκτέλεση των συναλλαγών, επηρεάζοντας έτσι την ρυθμαπόδοση του συστήματος (throughput).

Επίσης, ο τρόπος λειτουργίας των συστημάτων αυτών, και ειδικότερα της επίτευξης της συναίνεσης, βασίζεται στο γεγονός ότι όλοι οι κόμβοι του δικτύου χρειάζεται να έχουν πρόσβαση στα δεδομένα, τον κώδικα του συμβολαίου και τα

αποτελέσματα της εκτέλεσής τους. Το γεγονός αυτό περιορίζει κάπως τις εφαρμογές που μπορούμε να αναπτύξουμε χρησιμοποιώντας την τεχνολογία αυτή, καθώς σε ορισμένες περιπτώσεις χρειάζεται ορισμένες μόνο οντότητες να έχουν πρόσβαση στα παραπάνω δεδομένα (confidentiality). Αυτό θα μπορούσε, βέβαια, να γίνει, χρησιμοποιώντας αλγόριθμους κρυπτογράφησης ή μεθόδους απόδειξης μηδενικής γνώσης (zero-knowledge proof)[27], ωστόσο, η προσέγγιση αυτή αυξάνει την πολυπλοκότητα του συστήματος και απαιτεί επιπρόσθετους υπολογισμούς και κατ' επέκταση υπολογιστικούς πόρους. Επιπλέον, τα προγράμματα που βρίσκονται στο Blockchain πρέπει να είναι ντετερμινιστικά (να παράγουν πάντα το ίδιο αποτέλεσμα), έτσι ώστε η ανεξάρτητη εκτέλεση των προγραμμάτων από τους κόμβους του δικτύου να οδηγεί πάντα το σύστημα στην ίδια κατάσταση. Για την αποφυγή συγγραφής μη ντετερμινιστικών προγραμμάτων (π.χ., καλούμε άμεσα ή έμμεσα μία συνάρτηση που μας επιστρέφει έναν τυχαίο αριθμό), τα συστήματα αυτά δεν μας επιτρέπουν τη χρήση μιας γλώσσας γενικού σκοπού, όπως είναι η C++ ή η Java. Αντ' αυτού, ο χρήστης θα πρέπει να εκφράσει τα προγράμματά του (smart contracts) σε μία διαφορετική γλώσσα, η οποία είναι συγκεκριμένη για το κάθε σύστημα (π.χ., για το Ethereum συνιστάται η χρήση της γλώσσας Solidity), η οποία πολλές φορές συμπεριφέρεται διαφορετικά απ' ό,τι αναμέναμε. Το γεγονός αυτό απαιτεί επιπρόσθετη προσπάθεια από τη μεριά του προγραμματιστή, ο οποίος καλείται να μάθει μία νέα γλώσσα προγραμματισμού.

6.2.3.2 Το Μοντέλο Εκτέλεσης-Ταξινόμησης-Ελέγχου (Execute-Order-Validate Architecture)

Τα παραπάνω θέματα οδήγησαν στη δημιουργία ενός διαφορετικού μοντέλου για τον τρόπο εκτέλεσης των συναλλαγών και την επίτευξη της συναίνεσης, γνωστό ως *εκτέλεση-ταξινόμηση-έλεγχος (execute-order-validate)*, το οποίο «σπάει» τη διαδικασία σε τρία επιμέρους βήματα (εκτέλεση, ταξινόμηση, έλεγχο), τα οποία μπορούν να εκτελεστούν από διαφορετικές οντότητες. Η προσέγγιση αυτή δίνει μια ώθηση στην παράλληλη εκτέλεση των συναλλαγών και ακολούθως την ταξινόμησή τους, λαμβάνοντας υπόψη τις αλληλεξαρτήσεις που έχουν ανιχνευτεί κατά τη διάρκεια της εκτέλεσης. Η μεθοδολογία αυτή έχει χρησιμοποιηθεί στην πλατφόρμα του Hyperledger Fabric[28], το οποίο μας επιτρέπει να δημιουργήσουμε αλυσίδες, για τις οποίες θα πρέπει να λάβουμε κάποια άδεια, προκειμένου να συμμετέχουμε σε αυτές (*permissioned blockchains*), σε αντίθεση

[27] Zero-knowledge proof, https://en.wikipedia.org/wiki/Zero-knowledge_proof
[28] Hyperledger Fabric, https://www.hyperledger.org/use/fabric

με τις αλυσίδες και τα συστήματα που εξετάσαμε στις προηγούμενες ενότητες (*permissionless blockchains*), στα οποία καθένας θα μπορούσε να έχει πρόσβαση και να συμμετέχει στην επέκταση της αλυσίδας. Τα συστήματα αυτά είναι ιδανικά για τις περιπτώσεις εκείνες, στις οποίες θα θέλαμε να συμμετέχουν κάποιοι πιστοποιημένοι οργανισμοί, οι οποίοι έχουν έναν κοινό σκοπό, ωστόσο δεν υπάρχει πλήρης εμπιστοσύνη μεταξύ τους. Τα παραπάνω μπορούν να θεωρηθούν σαν μια σύντομη εισαγωγή στην περιγραφή της πλατφόρμας του Hyperledger. Για περισσότερες λεπτομέρειες γύρω από την πλατφόρμα του Hyperledger και τους μηχανισμούς που χρησιμοποιούνται παραπέμπουμε τους αναγνώστες στο σχετικό έγγραφο [25] που αναφέρεται στην αρχή της υποενότητας αυτής. Σημειώνουμε ότι η πλατφόρμα αυτή μας επιτρέπει, επίσης, να γράψουμε έξυπνα συμβόλαια, τα οποία ονομάζονται chaincode. Ωστόσο, τα προγράμματα αυτά θα πρέπει να είναι γραμμένα σε κάποια στάνταρ γλώσσα προγραμματισμού, γενικού σκοπού, όπως είναι η Go.

Η γλώσσα προγραμματισμού Go (aka Golang)[29] είναι μια συναρτησιακή γλώσσα προγραμματισμού, που έχει σχεδιαστεί από την Google. Είναι αρκετά απλή γλώσσα, που μοιάζει με την C και υποστηρίζει την ανάπτυξη μεγάλου εύρους εφαρμογών, συμπεριλαμβανομένων της ανάπτυξης διαδικτυακών προγραμμάτων (network programming), της ανάλυσης μεγάλου όγκου δεδομένων (big data) και των μηχανισμών μηχανικής μάθησης (machine learning). Τα προγράμματα που είναι γραμμένα στην γλώσσα αυτή μεταγλωττίζονται σε κώδικα μηχανής και επομένως δεν χρειάζονται κάποιον διερμηνέα για να εκτελεστούν. Κατά συνέπεια, τα προγράμματα αυτά εκτελούνται γρήγορα.

[29] The Go Programming Language, https://golang.org/

Η σελίδα αυτή είναι σκόπιμα λευκή

7 Σύνοψη και Συμπεράσματα

Στα πλαίσια της εργασίας αυτής, αρχικά, μελετήσαμε τα συστήματα Bitcoin και Ethereum. Και τα δύο αυτά συστήματα βασίζονται στην ύπαρξη ενός ελευθέρως προσβάσιμου διανεμημένου ψηφιακού βιβλίου, γνωστό ως Blockchain, στο οποίο καταγράφονται οι συναλλαγές που έχουν πραγματοποιηθεί, οργανωμένες σε μπλοκ. Οι αλγόριθμοι συναίνεσης που χρησιμοποιούνται, σε συνδυασμό με τη χρήση της κρυπτογραφίας δημόσιου κλειδιού, μας διασφαλίζουν ότι τα χρήματά μας δεν θα τροποποιηθούν, παρά το γεγονός ότι δεν υπάρχει μια κεντρική αρχή ελέγχου, ενώ παράλληλα μπορούμε να τα χρησιμοποιήσουμε, για να πραγματοποιήσουμε με ασφάλεια τις συναλλαγές μας. Το Ethereum, επιπρόσθετα, μας δίνει τη δυνατότητα να αναπτύξουμε τα δικά μας προγράμματα υπό την μορφή των έξυπνων συμβολαίων (Smart Contracts), τα οποία αποθηκεύονται, επίσης, στο Blockchain και μπορούν να εκτελεστούν από τους κόμβους του δικτύου. Αυτό ώθησε στη δημιουργία ενός σημαντικού εύρους εφαρμογών (συμπεριλαμβανομένων εφαρμογών στον χώρο της υγείας), οι οποίες βασίζονται στα δεδομένα και τα προγράμματα που αποθηκεύονται στους κόμβους του δικτύου, για να παρέχουν με ασφάλεια τις υπηρεσίες τους.

Στην εργασία αυτή παρουσιάσαμε μια κατανεμημένη εφαρμογή, η οποία βασίζεται στις τεχνολογίες του Blockchain και επιτρέπει στους χρήστες να συνάψουν οικονομικές συμφωνίες με ασφαλιστικές εταιρίες, που δραστηριοποιούνται στον χώρο της υγείας, αναφορικά με την κατάστασή τους και την πιθανή εκδήλωση κάποιας ασθένειας. Ωστόσο, η εφαρμογή αυτή βασίζεται στα δεδομένα που παρέχονται από μια τρίτη οντότητα, κατά την επίσκεψη ενός ασθενή σε κάποιο κέντρο περίθαλψης. Τα δεδομένα των ασθενών αποθηκεύονται σε μια σχεσιακή βάση που βρίσκεται εκτός της αλυσίδας, ενώ η πληροφορία για τα δεδομένα που ανήκουν σε έναν ασθενή και τις οντότητες που μπορούν να έχουν πρόσβαση σε αυτά αποθηκεύεται στο Blockchain υπό την μορφή δεικτών. Τα έξυπνα συμβόλαια που υπογράφονται μεταξύ των ασθενών και των ασφαλιστικών εταιριών αποθηκεύονται επίσης στο Blockchain, το οποίο και συμβουλεύονται, για να εντοπίσουν τα δεδομένα των χρηστών και να εξετάσουν εάν πληρούνται οι όροι του συμβολαίου. Ωστόσο, ο έλεγχος αυτός γίνεται μέσω της χρήσης ενός service που

βρίσκεται εκτός του Blockchain και αναλαμβάνει να εξετάσει τα δεδομένα των ασθενών και ακολούθως να ενημερώσει την αλυσίδα με τα κατάλληλα γεγονότα, έτσι ώστε έπειτα να αποζημιώσει τον χρήστη, εφόσον πληρούνται οι όροι.

Το γεγονός ότι τα έξυπνα συμβόλαια έχουν αποθηκευτεί στο Blockchain μάς εξασφαλίζει ότι δεν θα τροποποιηθούν, καθώς επίσης και ότι θα αποζημιωθούν άμεσα οι χρήστες, εφόσον ικανοποιούνται οι συνθήκες και υπάρχει καταγεγραμμένο το αντίστοιχο γεγονός. Όμως, το σύστημα εξαρτάται σε μεγάλο βαθμό από το service που εξετάζει τα δεδομένα, καθώς κακή λειτουργία του service αυτού θα είχε άμεση επίπτωση στην ομαλή λειτουργία του συστήματος. Περαιτέρω βελτίωση της προσέγγισης αυτής θα ήταν να καταγράφεται επιπρόσθετη πληροφορία όσον αφορά την χρήση του παραπάνω Web Service στο Blockchain, ώστε να έχουμε καλύτερο έλεγχο της υπηρεσίας (π.χ. να γνωρίζουμε εάν έτρεξε και μάλιστα πότε ήταν η τελευταία φορά). Επίσης, και η χρήση κρυπτογραφίας κατά την ανταλλαγή των δεδομένων θα προστατεύσει τους δείκτες των δεδομένων ενός ασθενή από πιθανούς κακόβουλους χρήστες. Ωστόσο, δεν θα πάψουμε να εξαρτόμαστε σε κάποιο βαθμό από αυτό. Επίσης, τα δεδομένα των χρηστών θα μπορούσαν να είναι διάσπαρτα στους κόμβους του δικτύου (διαφορετικούς από τους κόμβους του Ethereum), έτσι ώστε, ακόμη και εάν κάποιο κακόβουλο λογισμικό καταφέρει να αποκτήσει πρόσβαση στη βάση, να έχει πρόσβαση μόνο σε ένα μέρος αυτών. Ένας άλλος παράγοντας έχει να κάνει με τη μονάδα περίθαλψης και κατά πόσο αυτή συμπεριφέρεται σωστά. Παράληψη ή λανθασμένη καταγραφή των γεγονότων επηρεάζει άμεσα το σύστημα. Ωστόσο, κάτι τέτοιο δεν μπορούμε να το αποφύγουμε παρά μόνο μέσω χρήσης ελέγχων των δεδομένων, για να εντοπίσουμε έγκαιρα ασυνέπειες μεταξύ αυτών, που θα μπορούσαν να οδηγήσουν σε απάτη.

Τέλος δεν θα πρέπει να παραλείψουμε να αναφέρουμε τις αδυναμίες που έχει η ίδια η πλατφόρμα του Ethereum και ειδικότερα η χρήση των Smart Contracts. Σημειώνουμε ότι η πλατφόρμα αυτή βασίζεται στην ύπαρξη ενός δημόσιου Blockchain, στο οποίο μπορεί να έχει πρόσβαση οποιοσδήποτε χρήστης, ο οποίος έχει μάλιστα τη δυνατότητα να δει και τα δεδομένα που υπάρχουν σε αυτό και να αλληλεπιδράσει με τα υπάρχοντα προγράμματα. Εναλλακτικά θα μπορούσαμε να χρησιμοποιήσουμε κάποιο Blockchain, στο οποίο να υπάρχει περιορισμός ως προς τις οντότητες που συμμετέχουν σε αυτό, έτσι ώστε να διασφαλίσουμε την ταυτότητα των εταιριών και των ασθενών που συμμετέχουν σε αυτό καθώς επίσης και των κέντρων περίθαλψης. Μια τέτοια προσέγγιση

θα μας επέτρεπε να διαχειριστούμε καλύτερα τα δεδομένα των χρηστών, να βελτιώσουμε εν γένει τον χρόνο που απαιτείται για την εκτέλεση προγραμμάτων και γενικότερα να έχουμε μια αλυσίδα που είναι πιο κοντά στις ανάγκες της εφαρμογής και στην οποία μπορούν να έχουν πρόσβαση συγκεκριμένοι χρήστες.

Σε κάθε περίπτωση, κατά την ανάπτυξη μιας εφαρμογής, θα πρέπει να σκεφτούμε προσεκτικά εάν, όντως, χρειαζόμαστε τη χρήση των τεχνολογιών αυτών, καθώς, όπως είδαμε, αυξάνουν την πολυπλοκότητα του συστήματος και περιορίζουν την ταχύτητα εκτέλεσης των εφαρμογών. Για αυτόν τον λόγο υπάρχουν και κάποιες εργασίες στη βιβλιογραφία, όπως αυτή [26] η οποία μας θέτει κάποια ερωτήματα τα οποία μας βοηθούν να αποφασίσουμε εάν, όντως, χρειάζεται να χρησιμοποιήσουμε την τεχνολογία αυτή ή όχι.

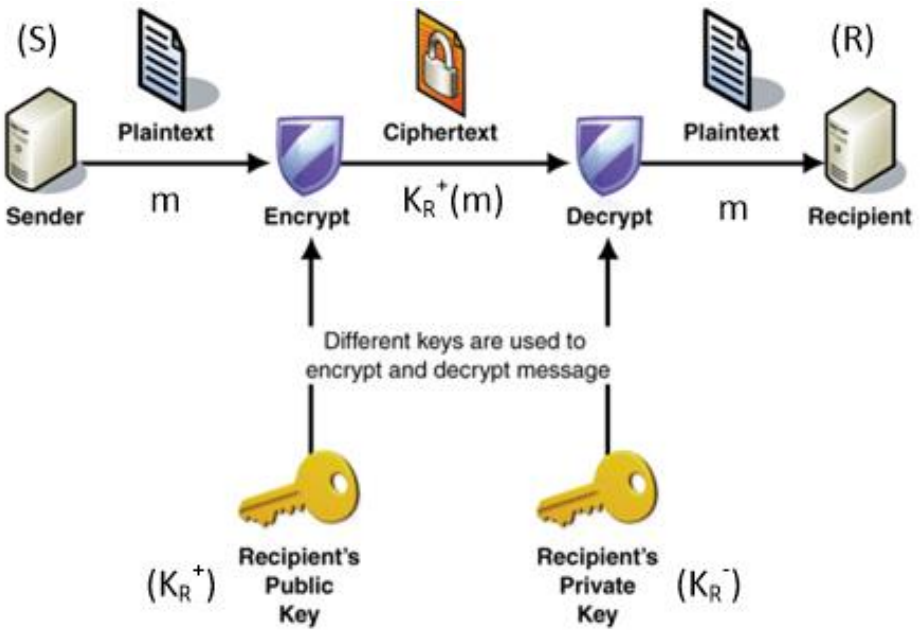
Η σελίδα αυτή είναι σκόπιμα λευκή

8 Παράρτημα

8.1 Κρυπτογράφηση Δημόσιου Κλειδιού

8.1.1 Ανταλλαγή Κρυπτογραφημένων Μηνυμάτων

Η κρυπτογράφηση δημόσιου κλειδιού βασίστηκε στην εργασία των Diffie και Hellman, που δημοσιεύτηκε το 1976 [2]. Η βασική ιδέα 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 [9]. Στην κρυπτογραφία δημόσιου κλειδιού είναι απαραίτητη η ύπαρξη ενός δημόσιου (K^+) και ενός ιδιωτικού (K^-) κλειδιού για καθέναν από τους δύο συμμετέχοντες. Το δημόσιο κλειδί είναι γνωστό σε όλους, σε αντίθεση με το ιδιωτικό κλειδί που είναι γνωστό μόνο από τον αντίστοιχο συμμετέχοντα.



Σχήμα 17: Η Διαδικασία της Κρυπτογράφησης και της Αποκρυπτογράφησης[30]

[30] https://www.tutorialspoint.com/cryptography/public_key_encryption.htm

Κατά τη διαδικασία της κρυπτογράφησης (Σχήμα 17) ο αποστολέας χρησιμοποιεί το δημόσιο κλειδί του παραλήπτη (K_R^+) για την κρυπτογράφηση ενός μηνύματος (m). Ακολούθως, ο παραλήπτης χρησιμοποιεί το ιδιωτικό του κλειδί (K_R^-) για την αποκρυπτογράφηση του κρυπτογραφημένου μηνύματος $K_R^+(m)$ που λαμβάνει από τον αποστολέα. Στο ακόλουθο σχήμα αναπαρίσταται γραφικά η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης ενός μηνύματος (m) από τον αποστολέα (S) στον παραλήπτη (R).

Για να λειτουργήσει σωστά η κρυπτογράφηση δημόσιου κλειδιού, θα πρέπει τόσο το δημόσιο, όσο και το ιδιωτικό κλειδί του καθενός να επιλεχθεί/παραχθεί με τέτοιο τρόπο, ώστε η κρυπτογράφηση του μηνύματος, χρησιμοποιώντας το ιδιωτικό κλειδί του αποστολέα και ακολούθως η αποκρυπτογράφησή του χρησιμοποιώντας το ιδιωτικό του κλειδί, να δίνει το αρχικό μήνυμα και αντιστρόφως.

$$K_R^-\,[\,K_R^+(m)\,] = K_R^+\,[\,K_R^-(m)\,] = m$$

8.1.2 RSA Αλγόριθμοι Κρυπτογράφησης και Αποκρυπτογράφησης

Ο αλγόριθμος RSA (από τα αρχικά των ονομάτων των Rivest, Shamir, και Adleman που δημοσίευσαν το σχετικό έγγραφο το 1977) [3] είναι αρκετά διαδεδομένος στην κρυπτογράφηση δημόσιου κλειδιού. Η παραγωγή τόσο του δημόσιου, όσο και του ιδιωτικού κλειδιού γίνεται ακολουθώντας κάποια συγκεκριμένα βήματα, μέσω των οποίων προκύπτουν οι αριθμοί m, e, d τέτοιοι, ώστε να είναι δύσκολο να προσδιορίσουμε το e ή το d με βάση τους άλλους δύο αριθμούς. Το δημόσιο κλειδί αποτελείται από το ζεύγος (n, e), ενώ το ιδιωτικό κλειδί από το ζεύγος (n, d).

Έστω ότι ο αποστολέας θέλει να στείλει στον παραλήπτη ένα μήνυμα (m). Για να γίνει αυτό εφικτό, χρειάζεται ο παραλήπτης να του στείλει το δημόσιό του κλειδί (n, e). Το κρυπτογραφημένο μήνυμα (c) υπολογίζεται από τον τύπο:

$$c = m^e \bmod n$$

Η αποκρυπτογράφηση του κρυπτογραφημένου μηνύματος (c) γίνεται χρησιμοποιώντας τον ακόλουθο τύπο:

$$m = c^d \bmod n$$

8.1.3 Ακεραιότητα Μηνύματος / Σύνοψη Μηνύματος και Ψηφιακή Υπογραφή

Η κρυπτογράφηση δημόσιου κλειδιού επιτρέπει την ασφαλή επικοινωνία μεταξύ δύο οντοτήτων, χωρίς να είναι απαραίτητο να χρησιμοποιήσουν ένα κοινό μυστικό κλειδί, όπως στην περίπτωση της κρυπτογράφησης συμμετρικού κλειδιού. Ωστόσο, η κρυπτογράφηση (από μόνη της) δεν μας εγγυάται ότι το μήνυμα δεν έχει αλλαχθεί, ούτε μας εξασφαλίζει ότι προέρχεται από έναν συγκεκριμένο αποστολέα. Τα προβλήματα αυτά μπορούν να λυθούν μέσω της χρήσης της ψηφιακής υπογραφής.

Για τον σκοπό αυτό ο αποστολέας χρειάζεται να στείλει επιπλέον πληροφορία που θα διασφαλίζει ότι το μήνυμα δεν έχει αλλαχθεί (ακεραιότητα μηνύματος), καθώς και ότι προέρχεται από αυτόν. Αυτό μπορεί να επιτευχθεί με τη χρήση του ιδιωτικού του κλειδιού για την «υπογραφή» του μηνύματος (δηλαδή την αποστολή όχι μόνο του αρχικού μηνύματος αλλά και του κρυπτογραφημένου με το δικό του ιδιωτικό κλειδί μηνύματος). Ακολούθως, ο παραλήπτης (μετά την αποκρυπτογράφηση του μηνύματος) μπορεί να χρησιμοποιήσει το δημόσιο κλειδί του αποστολέα, για να διαπιστώσει ότι το μήνυμα προέρχεται από αυτόν.

$$K_S^+ [K_S^- (m)] = m$$

Μια πιο αποδοτική προσέγγιση από την παραπάνω (λόγω του υπολογιστικού κόστους που εισάγει η επιπλέον κρυπτογράφηση ολόκληρου του μηνύματος) περιλαμβάνει τη δημιουργία μιας σύνοψης ενός μηνύματος (H(m)) και ακολούθως την κρυπτογράφησή του από τον αποστολέα χρησιμοποιώντας το ιδιωτικό του κλειδί. Αυτή είναι η ψηφιακή υπογραφή. Ο παραλήπτης ακολούθως μπορεί να αποκρυπτογραφήσει την κρυπτογραφημένη σύνοψη του μηνύματος και να ελέγξει εάν είναι ίδια με αυτή που προκύπτει από το μήνυμα που έχει σταλεί.

$$K_S^+ [K_S^- (H(m))] = H(m)$$

Απαραίτητη προϋπόθεση είναι ο αλγόριθμος που θα χρησιμοποιηθεί για τη δημιουργία της σύνοψης ενός μηνύματος να είναι τέτοιος, ώστε να είναι υπολογιστικά ανέφικτο να μπορούν να προκύψουν δύο ίδιες συνόψεις από διαφορετικά μηνύματα.

8.1.4 Secure Hash Algorithm (SHA)

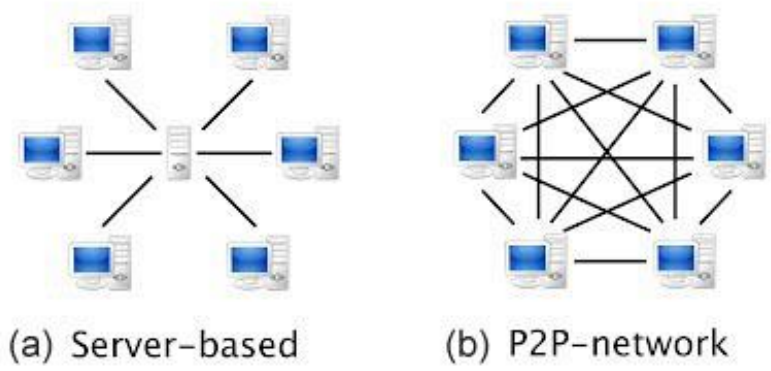
Μια συνάρτηση κρυπτογραφικού κατακερματισμού (cryptographic hash function) είναι μία μαθηματική συνάρτηση, η οποία αντιστοιχίζει τα δεδομένα μας, τα οποία μπορεί να είναι οποιουδήποτε μεγέθους σε έναν πίνακα με προεπιλεγμένο μέγεθος (fixed size). Η συνάρτηση αυτή είναι μονόδρομη, που σημαίνει ότι είναι πρακτικά ανέφικτο να μπορέσουμε να βρούμε το αρχικό κείμενο με βάση το αποτέλεσμα της συνάρτησης αυτής. Βέβαια, το γεγονός ότι υπάρχει ένα πεπερασμένο πεδίο τιμών δίνει την εντύπωση ότι για διαφορετικά δεδομένα θα μπορούσε να προκύψει η ίδια έξοδος. Ωστόσο, αυτό είναι πρακτικά πάρα πολύ δύσκολο και σπάνιο να το συναντήσουμε.

Οι αλγόριθμοι ασφαλούς κρυπτογραφικού κατακερματισμού (Secure Hash Algorithms) έχουν σχεδιαστεί από τον Οργανισμό Εθνικής Ασφάλειας των Ηνωμένων Πολιτειών (NSA). Η δεύτερή τους έκδοση (SHA-2)[31] περιλαμβάνει έξι αλγορίθμους που ακολουθούν την ίδια λογική, αλλά διαφέρουν ως προς το μέγεθος της εξόδου και έχουν βελτιώσει κάποιες από τις αδυναμίες που υπήρχαν στην πρώτη έκδοση. Για περισσότερες πληροφορίες σχετικά με τον τρόπο λειτουργίας των αλγορίθμων αυτών, παραπέμπουμε τους αναγνώστες στη σχετική βιβλιογραφία.

8.2 Peer to Peer Computing

Ο όρος peer-to-peer (P2P) [10] αναφέρεται σε ένα σύνολο από συστήματα και εφαρμογές που χρησιμοποιούν διάσπαρτους πόρους, για να παρέχουν μία συγκεκριμένη υπηρεσία μέσα σε ένα αποκεντροποιημένο περιβάλλον. Τα συστήματα αυτά περιορίζουν την ανάγκη για την απόκτηση και διατήρηση μιας κοστοβόρας υποδομής (που θα ήταν απαραίτητη για την παροχή των επιθυμητών υπηρεσιών), επιτρέποντας την απευθείας επικοινωνία μεταξύ των χρηστών (Σχήμα 18 – (b)), χωρίς την ανάγκη ύπαρξης ενός κεντρικού εξυπηρετητή, όπως στην περίπτωση του μοντέλου πελάτης-εξυπηρετητής (client-server) (Σχήμα 18 – (a)). Κατά συνέπεια, τα συστήματα που έχουν αναπτυχτεί, ακολουθώντας το παραπάνω μοντέλο, μπορούν εύκολα να επεκταθούν (scalability) με νέους κόμβους, οι οποίοι έχουν πρόσβαση σε ένα μεγάλο σύνολο από πόρους, μέσω της δυνατότητας της συσσώρευσης που παρέχεται στους κόμβους του δικτύου.

[31] Secure Hash Algorithm 2 (SHA-2), https://en.wikipedia.org/wiki/SHA-2



Σχήμα 18: (a) Μοντέλο Πελάτη/Εξυπηρετητή (b) Δίκτυο Ομότιμων Κόμβων (p2p)[32]

Σε ένα peer-to-peer δίκτυο οι *κόμβοι του δικτύου* (ομότιμοι κόμβοι) επικοινωνούν απευθείας μεταξύ τους, έχοντας ως απώτερο σκοπό τη λήψη κάποιων πόρων που υπάρχουν διάσπαρτοι στο δίκτυο, αλλά και να προσφέρουν πόρους σε αυτό. Με απλά λόγια οι κόμβοι αυτοί λειτουργούν τόσο ως πελάτες, όσο και ως εξυπηρετητές. Οι πόροι που παρέχονται είναι συνήθως κάποια αρχεία. Ωστόσο, η προσέγγιση αυτή μπορεί να χρησιμοποιηθεί για την παροχή υπολογιστικών κύκλων, προκειμένου να πετύχουμε παράλληλη επεξεργασία δεδομένων ή τη δημιουργία ενός συστήματος που βασίζεται στην συνεργασία μεταξύ των κόμβων του, για την παροχή των προδιαγεγραμμένων υπηρεσιών.

Οι ομότιμοι κόμβοι ενός peer-to-peer δικτύου λειτουργούν αυτόνομα και μπορούν ελεύθερα να συμμετέχουν σε ένα τέτοιο σύστημα, αλλά και να φεύγουν από αυτό οποιαδήποτε χρονική στιγμή. Συνεπώς, οι κόμβοι του δικτύου, παρά το γεγονός ότι βασίζονται στους πόρους που παρέχονται από τους υπόλοιπους κόμβους του δικτύου, προκειμένου να πετύχουν τον σκοπό τους ή να προσφέρουν μία υπηρεσία, δεν εξαρτώνται (ή τουλάχιστον έτσι θα πρέπει) απόλυτα από τους γύρω κόμβους, καθώς το δίκτυο είναι δυναμικό και ανανεώνεται συνεχώς. Επίσης, κάποιοι από τους κόμβους του δικτύου μπορεί να μην είναι έμπιστοι.

Ένα σημαντικό θέμα στα peer-to-peer δίκτυα είναι η μεθοδολογία που ακολουθείται για την *ανακάλυψη νέων κόμβων* και κυρίως των πόρων που αυτοί παρέχουν στο δίκτυο. Στην περίπτωση ενός «καθαρού» peer-to-peer μοντέλου, το αίτημα μεταδίδεται σταδιακά σε όλους τους κόμβους του δικτύου. έως ότου βρεθεί ο κόμβος που μπορεί να απαντήσει. Η προσέγγιση αυτή, αν και διευκολύνει την επεκτασιμότητα του

[32] https://sites.google.com/site/cis3347cruzguzman014/module-2/client-server-and-peer-to-peer-networking

συστήματος, απαιτεί τη χρήση ενός σημαντικού μέρους του εύρους ζώνης του δικτύου και κατά συνέπεια δεν είναι και τόσο αποτελεσματική, όταν υπάρχουν πολλοί κόμβοι. Στην περίπτωση του «υβριδικού» μοντέλου (λόγω της χρήσης τεχνολογιών πελάτη-εξυπηρετητή στην αρχική φάση της επικοινωνίας) υπάρχει ένας κεντρικός κόμβος, στον οποίον οι υπόλοιποι κόμβοι αρχικά δηλώνουν τους πόρους που μπορούν να παρέχουν στο δίκτυο και ακολούθως τον συμβουλεύονται, προκειμένου να εντοπίσουν τους κόμβους που παρέχουν τον πόρο που χρειάζονται. Έπειτα, όμως, επικοινωνούν κατευθείαν με τον αντίστοιχο κόμβο, για τη λήψη του επιθυμητού πόρου.

Ένας από τους στόχους ενός P2P συστήματος είναι να επιτρέψει στους χρήστες να έχουν πρόσβαση σε διάφορους πόρους, χωρίς να ανησυχούν για νομικές ή άλλες επιπτώσεις. Για τον σκοπό αυτό μπορεί τόσο ο αποστολέας, όσο και ο παραλήπτης να διατηρούν την *ανωνυμία* τους στο δίκτυο. Ωστόσο, υπάρχουν διάφορα επίπεδα ανωνυμίας (π.χ., ανάλογα με τον αλγόριθμο αναζήτησης που χρησιμοποιείται, οι χρήστες μπορεί να είναι πιθανώς εκτεθειμένοι και άρα ανιχνεύσιμοι), τα οποία έχουν να κάνουν και με τους αλγορίθμους που χρησιμοποιούνται για τον εντοπισμό των πόρων.

Η ασφάλεια (security) είναι από τις μεγαλύτερες προκλήσεις στις P2P εφαρμογές. Αυτό γίνεται εύκολα κατανοητό, αν αναλογιστούμε ότι οι κόμβοι ενός τέτοιου συστήματος λειτουργούν όχι μόνο ως πελάτες αλλά και ως εξυπηρετητές, δημιουργώντας επιπρόσθετους κινδύνους για τους χρήστες. Για τη διασφάλιση του κάθε κόμβου, το επιπλέον λογισμικό που απαιτείται σε κάθε κόμβο για τη λειτουργία του συστήματος, είναι απομονωμένο από τους κρίσιμους πόρους του κόμβου (η στρατηγική αυτή είναι γνωστή ως sandboxing[33]), έτσι ώστε να μην βάλει σε κίνδυνο τη λειτουργία του συστήματος του κάθε κόμβου. Επίσης, τεχνικές κρυπτογράφησης, όπως η κρυπτογράφηση δημόσιου κλειδιού, μπορούν να χρησιμοποιηθούν κατά την ανταλλαγή των μηνυμάτων, ενώ οι ψηφιακές υπογραφές για την προστασία της πνευματικής ιδιοκτησίας. Τέλος, η φήμη του πόσο καλός ή χρήσιμος είναι ο κάθε κόμβος, μπορεί επίσης να χρησιμοποιηθεί για την προστασία του όλου συστήματος.

Η αξιοπιστία (reliability) ενός P2P συστήματος είναι ένα αρκετά δύσκολο πρόβλημα. Η προσέγγιση που συνήθως ακολουθείται είναι η χρήση πλεονάζουσας πληροφορίας, η οποία είναι ιδιαίτερα χρήσιμη στην περίπτωση αστοχίας του συστήματος. Για παράδειγμα, στην περίπτωση ενός P2P συστήματος για τον διαμοιρασμό αρχείων, θα

[33] Sandbox (computer security), https://en.wikipedia.org/wiki/Sandbox_(computer_security)

μπορούσαμε να διατηρούμε αντίγραφα σε παραπάνω από έναν κόμβους, έτσι ώστε να διευκολύνουμε την εξυπηρέτηση των πελατών και τη διαθεσιμότητα των αρχείων στο σύστημα, λαμβάνοντας υπόψη τον δυναμικό χαρακτήρα αυτών των συστημάτων.

8.3 Solidity

8.3.1 Χαρακτηριστικά της Γλώσσας Solidity

Η Solidity[34] είναι μία αντικειμενοστραφής υψηλού επιπέδου γλώσσα προγραμματισμού, που είναι χρήσιμη για τη δημιουργία των Smart Contracts. Αυτά είναι προγράμματα, τα οποία μπορούν να εκτελεστούν από την πλατφόρμα του Ethereum[35] και να μεταβάλλουν την κατάσταση του Blockchain συστήματος.

Για τη διευκόλυνση της κατανόησης των Smart Contracts, θα μπορούσαμε να τα φανταστούμε ως Java Classes, από τις οποίες μπορούμε να δημιουργήσουμε ένα instance (π.χ., κατά το deployment) και ακολούθως να καλέσουμε τις μεθόδους του. Ωστόσο, για να καλέσουμε τις αντίστοιχες μεθόδους (συμπεριλαμβανομένου του constructor που καλείται κατά τη δημιουργία του), θα πρέπει να καταβάλουμε ένα ποσό, που είναι αντίστοιχο με το κόστος των εντολών, που περιλαμβάνει κάθε μία από αυτές τις συναρτήσεις.

Κατά τη δημιουργία ενός Smart Contract μπορούμε να ορίσουμε τις *μεταβλητές* που μας ενδιαφέρουν. Ωστόσο, για κάθε μία από αυτές θα πρέπει να ορίσουμε τον τύπο και την εμβέλειά τους. Η γλώσσα αυτή περιλαμβάνει ορισμένους "primitive" data types για την αποθήκευση ακέραιων αριθμών, συμβολοακολουθιών και διευθύνσεων (address). Ωστόσο, μπορούμε να δημιουργήσουμε και δικούς μας τύπους δεδομένων, χρησιμοποιώντας την λέξη κλειδί *struct*. Επίσης, μπορούμε να περιορίσουμε την εμβέλεια των μεταβλητών, χρησιμοποιώντας τα προσδιοριστικά public, internal, και private. Σημειώνουμε ότι, στην περίπτωση των public μεταβλητών, δημιουργείται αυτόματα μια getter συνάρτηση με ακριβώς το ίδιο όνομα, την οποία μπορούμε να χρησιμοποιήσουμε, για να πάρουμε την τιμή της. Επίσης, οι τιμές των μεταβλητών ορίζουν την *κατάσταση* (state), στην οποία βρίσκεται το συμβόλαιο και μπορούν να μεταβληθούν είτε κατά την αρχικοποίηση είτε καλώντας την κατάλληλη συνάρτηση (εφόσον υπάρχει).

[34] Solidity, https://solidity.readthedocs.io/
[35] Ethereum, https://ethereum.org/en/

Variables Definition:

Type Scope Name π.χ., address private owner

Οι *συναρτήσεις* αποτελούν τμήματα κώδικα, τα οποία μπορούμε να καλέσουμε είτε ως χρήστες είτε μέσω μιας άλλης μεθόδου ή προγράμματος, εφόσον κάτι τέτοιο είναι εφικτό. Αυτό εξαρτάται από τα προσδιοριστικά που συνοδεύουν τον ορισμό της συνάρτησης και τα οποία δηλώνουν κατά πόσο μπορούμε να έχουμε πρόσβαση σε αυτήν, προϋποθέσεις που πρέπει να πληρούνται, εάν χρειάζεται να αλλάξουμε την τρέχουσα κατάσταση ή όχι (π.χ., pure function) ή και ακόμη εάν μπορούμε να στείλουμε χρήματα στο έξυπνο αυτό συμβόλαιο κατά την εκτέλεση της μεθόδου (payable).

Function Definition:

Function Name (Input-Variables-List) modifiers returns (Output-Variables-List) { /* … */ }

π.χ. function createToken(bytes32 name) public returns (OwnedToken tokenAddress) { /* … */ }

Το κυρίως μέρος της κάθε συνάρτησης περιγράφει βήμα προς βήμα τη διαδικασία που πρέπει να ακολουθήσουμε, για να παρέχουμε το επιθυμητό αποτέλεσμα. Στο σημείο αυτό να τονίσουμε ότι η γλώσσα solidity είναι *turing complete*. Αυτό σημαίνει ότι παρέχει τις βασικές δομές ελέγχου, μέσω των οποίων μπορούμε να εκφράσουμε διακλαδώσεις (if then else) καθώς και επαναλήψεις (loops).

Μέσω των συναρτήσεων μπορούμε να εκφράσουμε κάποια σημαντικά *γεγονότα (events)*, τα οποία μπορούμε ακολούθως να τα εντοπίσουμε μέσω της πλατφόρμας του Ethereum και ειδικότερα του Virtual Machine και κατ' επέκταση να τα επεξεργαστούμε.

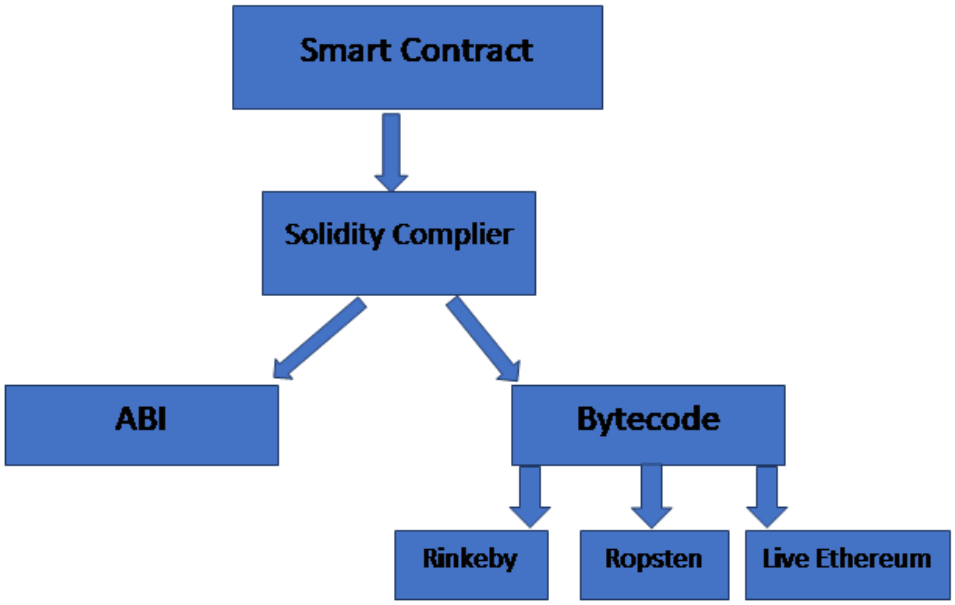
Event Definition:

event Event-Name(Variables-List);

Η τρέχουσα *έκδοση* της γλώσσας Solidity είναι η v.0.7.1. Σημειώνουμε ότι η γλώσσα αυτή δεν έχει πάρει ακόμη την τελική της μορφή, ενώ παρουσιάζει σημαντικές διαφορές από προηγούμενες εκδόσεις της, με αποτέλεσμα προγράμματα τα οποία είναι γραμμένα σε προηγούμενες εκδόσεις να μην μπορούν να μεταγλωττιστούν από πιο σύγχρονες εκδόσεις. Για τον λόγο αυτό, σε κάθε συμβόλαιο χρειάζεται να δηλώσουμε την έκδοση του solidity compiler που απαιτείται, χρησιμοποιώντας την λέξη κλειδί pragma.

8.3.2 Solidity Compiler

Τα προγράμματα που είναι γραμμένα χρησιμοποιώντας τη γλώσσα solidity πρέπει να γίνουν compile και ακολούθως να ανέβουν στο Blockchain. Κατά τη διαδικασία αυτή δημιουργείται ο «εκτελέσιμος» κώδικας, γνωστός ως Bytecode, ο οποίος μπορεί ακολούθως να εκτελεστεί από το Ethereum Virtual Machine (EVM). Εκτός από τον κώδικα αυτόν, παράγει και ένα JSON αρχείο (Application Binary Interface aka ABI), το οποίο περιγράφει το συμβόλαιο που θα γίνει deploy και τις συναρτήσεις που αυτό περιέχει, έτσι ώστε να μπορέσουμε αργότερα να το καλέσουμε είτε εκτός Blockchain είτε μέσα από κάποιο άλλο Contract (Σχήμα 19).



Σχήμα 19: Μεταγλώττιση Έξυπνων Συμβολαίων[36]

8.3.3 Ethereum Smart Contracts Notes

Τα προγράμματα που είναι γραμμένα, για να εκτελεστούν από την πλατφόρμα του Ethereum, μοιάζουν με κλασικά προγράμματα που είναι γραμμένα σε μια γλώσσα γενικού σκοπού, όπως είναι η Java και η C++. Ωστόσο, θα θέλαμε να επισημάνουμε κάποιους παράγοντες, που θα πρέπει να λάβουμε υπόψη, οι οποίοι διαφοροποιούν σημαντικά τα smart contracts από τις παραδοσιακές Web εφαρμογές.

[36] https://medium.com/@munishkohli/solidity-compiler-using-node-js-to-compile-smart-contracts-ce961899731f

Καταρχήν, όταν μιλάμε για solidity, αυτόματα έρχεται στο μυαλό μας η έννοια του κόστους. Τόσο για να δημιουργήσουμε ένα instance ενός smart contract, όσο και για να καλέσουμε κάποια συνάρτηση, η οποία μεταβάλει την κατάσταση του συστήματος, θα πρέπει να καταβάλουμε κάποιο χρηματικό ποσό (Ether). Επίσης, η εκτέλεση μιας συνάρτησης μπορεί να αποτύχει, μόνο και μόνο επειδή δεν έφτασαν τα χρήματα (gas), κάτι που πολλές φορές δεν μπορούμε να ξέρουμε εξ αρχής. Επίσης, τα smart contracts και ειδικότερα οι εντολές που τα απαρτίζουν δεν έχουν όλες το ίδιο κόστος. Ειδικότερα, υπάρχουν σημαντικές διαφορές μεταξύ τους και κατά συνέπεια θα πρέπει να είμαστε πολύ προσεκτικοί κατά τη συγγραφή του κώδικά μας, καθώς περιττές εντολές μπορούν να μας βάλουν σε έξτρα έξοδα. Αυτό συμβάλλει στη δημιουργία «καλύτερου», υπό την έννοια του πιο προσεγμένου κώδικα, ωστόσο, εισάγει έναν επιπλέον βαθμό δυσκολίας κατά τη συγγραφή του έξυπνου συμβολαίου.

Ένας άλλος παράγοντας που θα πρέπει να λάβουμε υπόψη είναι ότι τα προγράμματα αυτά αποθηκεύονται στο Blockchain, το οποίο συνεπάγεται τα εξής. Πρώτον και βασικότερον είναι ότι, άπαξ και ανέβουν, δεν μπορούν να αλλάξουν. Συνεπώς, οποιοδήποτε σφάλμα ή ατέλεια θέτει το πρόγραμμά μας σε κίνδυνο και δύσκολα μπορεί αυτό να αλλάξει. Στην καλύτερη περίπτωση να απενεργοποιήσουμε το συμβόλαιο αυτό. Υπάρχουν, βέβαια, και προσεγγίσεις που επιτρέπουν στον χρήστη να αλλάξει το περιεχόμενο του συμβολαίου, αλλά υπό την έννοια αυτή το συμβόλαιο ξεφεύγει του αρχικού σχεδιασμού του και δημιουργεί ερωτηματικά ως προς την εγκυρότητά του, καθώς μπορεί πρακτικά να τροποποιηθεί (ηθελημένα/ελεγχόμενα ή μη, λόγω κάποιου προγραμματιστικού σφάλματος). Επίσης, κάθε φορά που φτιάχνουμε ένα συμβόλαιο ή καλούμε κάποια συνάρτηση, αυτή εκτελείται μεν μια φορά από έναν κόμβο, αλλά το ίδιο πρόγραμμα θα εκτελεστεί πολλές φορές από διαφορετικούς κόμβους (κόμβοι/miners που συμμετέχουν στη διαδικασία/διασφάλιση ύπαρξης και ακεραιότητας του Blockchain) και θα πρέπει να παράγουν κάθε φορά το ίδιο αποτέλεσμα / κατάσταση. Κατά συνέπεια, συναρτήσεις, όπως η random integer generator, που έχουμε συνηθίσει να χρησιμοποιούμε σε γλώσσες προγραμματισμού, όπως η Java, δεν έχουν πλέον νόημα. Βέβαια, αυτό μπορούμε κάπως να το προσεγγίσουμε, χρησιμοποιώντας πληροφορία από το τελευταίο αποδεκτό block, όπως το timestamp, που θα μπορούσαμε να πούμε ότι είναι τυχαίο, για να παράγουμε, ακολούθως, τυχαίους αριθμούς, που είναι, όμως, ίδιοι σε όλους τους κόμβους, αν και αυτή η προσέγγιση εγκυμονεί κινδύνους.

Επίσης, θα πρέπει να λάβουμε υπόψη ότι τα προγράμματα αυτά τρέχουν στην πλατφόρμα που βρίσκεται το Blockchain και βασίζονται στα δεδομένα που υπάρχουν σε αυτό. Συνεπώς, εάν θέλουμε να πάρουμε κάποια πληροφορία από τον έξω κόσμο, θα πρέπει να ακολουθήσουμε μια διαφορετική προσέγγιση (τύπου oracles) και όχι να «χτυπήσουμε» κατευθείαν το service ή τη βάση. Επίσης, η εκτέλεση ενός τμήματος κώδικα και η πιστοποίησή του, απέχουν χρονικά, καθώς μεσολαβεί κάποιο χρονικό διάστημα, μέχρι να προστεθούν τα transactions, που ουσιαστικά συμβάλλουν στην αλλαγή καταστάσεων στο Blockchain. Αυτό δίνει τη δυνατότητα σε πιθανά κακόβουλα συμβόλαια να επεξεργαστούν τα δεδομένα που υπάρχουν ήδη και να διαφοροποιήσουν τη συμπεριφορά τους. Επίσης, τα transactions που εκτελούνται δεν μπαίνουν με μία προδιαγεγραμμένη σειρά στα μπλοκ, με ό,τι αυτό συνεπάγεται.

Τέλος τα exceptions, που μπορούν να προκύψουν κατά τη διάρκεια της εκτέλεσης (είτε λόγω σφάλματος είτε λόγω εξάντλησης χρημάτων) θέλουν λίγο διαφορετική αντιμετώπιση από αυτήν που συνήθως έχουμε σε γενικού σκοπού γλώσσες προγραμματισμού, καθώς μπορεί να προκαλέσουν ένα μπαράζ ανεπιθύμητων ενεργειών. Σημειώνουμε ότι, όταν προκύπτουν τέτοιου είδους προβλήματα, μπορεί το συμβόλαιο να επανέρχεται στην αρχική του κατάσταση, αλλά τα χρήματά μας θα έχουν ήδη ξοδευτεί, άσχετα αν δεν πετύχαμε τον στόχο μας. Επίσης, προσοχή θέλει με το τι γίνεται, όταν ένα συμβόλαιο καλεί ένα άλλο. Τέλος, η γλώσσα solidity μπορεί να είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού, ωστόσο, σε ορισμένες περιπτώσεις πρέπει να λάβουμε υπόψη θέματα που έχουν να κάνουν με γλώσσες χαμηλότερου επιπέδου, όπως η αφαίρεση ενός αριθμού από το 0.

Τα παραπάνω δείχνουν ότι ο προγραμματισμός των smart contacts διαφέρει σημαντικά από τη συγγραφή μιας διαδικτυακής εφαρμογής ή ενός web service. Δεδομένου ότι πολλοί software developers έχουν αρκετή εμπειρία στην ανάπτυξη τέτοιων εφαρμογών, η δημιουργία κατανεμημένων εφαρμογών (DApps) και ειδικότερα η δημιουργία και εκτέλεση των smart contracts, χρησιμοποιώντας την γλώσσα solidity, μπορεί να αποδειχθεί ιδιαίτερα περίπλοκη και σε ορισμένες περιπτώσεις επικίνδυνη για την ομαλή λειτουργία του συστήματος.

8.4 Τμήματα Κώδικα

8.4.1 Τμήμα Κώδικα Χρήστη (Client-Side React Code)

Στις ακόλουθες γραμμές μπορούμε να δούμε ένα πολύ μικρό τμήμα του Client-side React κώδικα, που αναπτύχθηκε για την επικοινωνία του χρήστη με το σύστημα και ειδικότερα, με τα smart contracts.

```
import getWeb3 from "./getWeb3";
import HealthInsuranceCompany from "./contracts/HealthInsuranceCompany.json";
// ...
// Get network provider and web3 instance.
const web3 = await getWeb3();
// Use web3 to get the user's accounts.
const accounts = await web3.eth.getAccounts();
// Get reference to Smart Contract using ABI
const networkId = await web3.eth.net.getId();
const deployedNetwork = HealthInsuranceCompany.networks[networkId];
const contract = new web3.eth.Contract(
        HealthInsuranceCompany.abi,
        deployedNetwork && deployedNetwork.address,
);
// Invoke Smart Contract Method
contract.methods.deposit().send(
    { from: accounts[0], gas: 3000000, value: web3.utils.toWei( depositvalue, 'ether') }
);
// ...
```

Το παραπάνω τμήμα του κώδικα καλεί τη μέθοδο deposit() του Health Insurance Company smart contract, στέλνοντας ένα συγκεκριμένο ποσό από Ether, αφού πρώτα το μετατρέψει σε wei. Για την επικοινωνία του χρήστη με τα smart contracts χρησιμοποιείται η web3 JavaScript βιβλιοθήκη καθώς επίσης και το ABI του smart contract, το οποίο έχει προκύψει κατά τη διαδικασία της μεταγλώττισης.

8.4.2 Τμήμα Κώδικα Έξυπνου Συμβολαίου (Smart Contract Solidity Code)

Στις ακόλουθες γραμμές μπορούμε να δούμε ένα πολύ μικρό τμήμα του έξυπνου συμβολαίου, που αναπτύχθηκε για την αποθήκευση των χρημάτων της ασφαλιστικής εταιρίας, έτσι ώστε ακολούθως να χρησιμοποιηθούν για τη σύναψη των συμβολαίων και την αποζημίωση των ασθενών.

```
/** Health Insurance Company Smart Contract that enable users/patients to sign a new Health
  * Contract regarding their Health Status */
contract HealthInsuranceCompany {
        // ...
        address payable internal owner;
        uint public weiAmount = 0;
         /** Store the account/owner created this smart contract */
        constructor() public {
                owner = msg.sender;
        }
        /** Access is restricted to the owner of the smart contract */
        modifier restricted() {
                if (msg.sender == owner) _;
        }
        /** Store an amount of money (WEI) to the smart contract */
        function deposit() public restricted payable {
                require(weiAmount + msg.value > weiAmount, "Safe Add");
                weiAmount += msg.value;
        }
        // ...
}
```

Η μέθοδος deposit() είναι διαθέσιμη μόνο στον χρήστη που έκανε deploy το έξυπνο αυτό συμβόλαιο. Επίσης, το payable modifier μας εξασφαλίζει ότι μπορούμε να στείλουμε χρήματα στο smart contract, όταν καλούμε τη μέθοδο αυτή. Σημειώνουμε ότι και η διεύθυνση του χρήστη είναι επίσης payable, έτσι ώστε να μπορέσουμε αργότερα να καταθέσουμε στον λογαριασμό αυτό τα διαθέσιμα/εναπομείναντα χρήματα.

Η σελίδα αυτή είναι σκόπιμα λευκή

9 Συντομογραφίες

Στον παρακάτω πίνακα παραθέτουμε τους όρους και συντομογραφίες που χρησιμοποιήθηκαν στη εργασία αυτή:

Όρος	Ερμηνεία
aka	also known as
ABI	Application Binary Interface
BFT	Byzantine Fault Tolerance
BTC	Bitcoin
ChEBI	Chemical Entities of Biological Interest
DAO	Decentralized Autonomous Organization
Dapp	Distributed Application
DoS attacks	Denial of Service attacks
EVM	Ethereum Virtual Machine
HIPAA	Health Insurance Portability and Accountability Act
ICD	International Classification of Diseases
JSON	JavaScript Object Notation
LOINC	Logical Observation Identifiers Names and Codes
P2P	Peer-to-Peer
PoS	Proof of Stake
PoW	Proof of Work
RSA	Initials of names: Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm

SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
UML	Unified Modeling Language

10 Βιβλιογραφικές Αναφορές

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.(2008).

[2] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE transactions on Information Theory, 22(6), 644-654.

[3] Rivest, R. L., Shamir, A., & Adleman, L. (1977). On Digital Signatures and Public-Key Cryptosystems (No. MIT/LCS/TM-82). MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE.

[4] Buterin, V. (2013). Ethereum white paper. GitHub repository, 1, 22-23.

[5] Kuo, T. T., Kim, H. E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. Journal of the American Medical Informatics Association, 24(6), 1211-1220.

[6] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, June). An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress) (pp. 557-564). IEEE.

[7] Zyskind, G., & Nathan, O. (2015, May). Decentralizing privacy: Using blockchain to protect personal data. In 2015 IEEE Security and Privacy Workshops (pp. 180-184). IEEE.

[8] Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016, August). Medrec: Using blockchain for medical data access and permission management. In 2016 2nd International Conference on Open and Big Data (OBD) (pp. 25-30). IEEE.

[9] Kurose, J. F., & Ross, K. W. (2012). Computer Networking: A Top-Down Approach . 6th. Harlow, UK: Pearson Education Ltd.

[10] Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., ... & Xu, Z. (2002). Peer-to-peer computing.

[11] Castro, M., & Liskov, B. (1999, February). Practical Byzantine fault tolerance. In OSDI (Vol. 99, No. 1999, pp. 173-186).

[12] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. ACM Trans. Program. Lang. Syst. 4, 3 (pp. 382–401).

[13] Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.".

[14] Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). Guide to elliptic curve cryptography. Springer Science & Business Media.

[15] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014), 1-32.

[16] Bambara, J. J., Allen, P. R., Iyer, K., Madsen, R., Lederer, S., & Wuehler, M. (2018). Blockchain: A practical guide to developing business, law, and technology solutions. McGraw Hill Professional.

[17] Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). Blockchain technology overview. arXiv preprint arXiv:1906.11078.

[18] Bartoletti, M., & Pompianu, L. (2017, April). An empirical analysis of smart contracts: platforms, applications, and design patterns. In International conference on financial cryptography and data security (pp. 494-509). Springer, Cham.

[19] Chondrogiannis, E., Andronikou, V., Karanastasis, E., & Varvarigou, T. A. (2014, December). An Intelligent Ontology Alignment Tool Dealing with Complicated Mismatches. In SWAT4LS.

[20] Chondrogiannis, E., Andronikou, V., Tagaris, A., Karanastasis, E., Varvarigou, T., & Tsuji, M. (2017). A novel semantic representation for eligibility criteria in clinical trials. Journal of Biomedical Informatics, 69, 10-23.

[21] Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. ACM Transactions on Database Systems (TODS), 34(3), 1-45.

[22] Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A., & Hayajneh, T. (2018). Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. Journal of medical systems, 42(7), 130.

[23] Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2020). A survey on the security of blockchain systems. Future Generation Computer Systems, 107, 841-853.

[24] Atzei, N., Bartoletti, M., & Cimoli, T. (2017, April). A survey of attacks on ethereum smart contracts (sok). In International conference on principles of security and trust (pp. 164-186). Springer, Berlin, Heidelberg.

[25] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Muralidharan, S. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).

[26] Wüst, K., & Gervais, A. (2018, June). Do you need a blockchain?. In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 45-54). IEEE.

Η σελίδα αυτή είναι σκόπιμα λευκή