ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΕΦΑΡΜΟΓΕΣ ΤΕΧΝΟΛΟΓΙΩΝ BLOCKCHAIN ΚΑΙ ΜΗ ΑΝΤΑΛΛΑΞΙΜΩΝ ΔΙΑΚΡΙΤΩΝ»

ΣΤΕΡΓΙΑΝΟΥ Ε. ΕΛΕΝΗ

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ: ΕΛΕΝΗ ΑΙΚΑΤΕΡΙΝΗ ΛΕΛΙΓΚΟΥ

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2024

Εξεταστική Επιτροπή

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την κάτωθι τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από τη Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του Τμήματος.

Α/Α	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	Λελίγκου Ελένη Αικατερίνη	Eleni Aikaterini Leligkou Digitally signed by Eleni Aikaterini Leligkou Date: 2024.04.08 10:35:57 +03'00'
2	Κάντζος Δημήτριος	Dimitrios Kantzos Digitally signed by Dimitrios Kantzos Date: 2024.04.04 21:04:31 +03'00'
3	Δρόσος Χρήστος	 Digitally signed by CHRISTOS DROSOS Date: 2024.04.04 22:54:44 +03'00'

Δήλωση Συγγραφέα Διπλωματικής Εργασίας

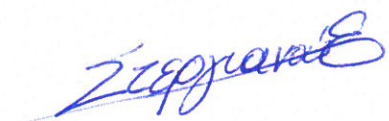
Η κάτωθι υπογεγραμμένη **Στεργιανού Ελένη** του **Ευστρατίου**, με αριθμό μητρώου **701252017117** φοιτήτρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

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

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Η Δηλούσα,

Στεργιανού Ελένη



Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε το χειμερινό εξάμηνο και δεν θα μπορούσε να είχε υλοποιηθεί χωρίς την πολύτιμη βοήθεια κάποιων ατόμων που θα ήθελα να ευχαριστήσω καθώς με στήριξαν ο καθένας με τον δικό του μοναδικό τρόπο.

Για αρχή θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια μου κα. Ελένη Αικατερίνη Λελίγκου, που μου ανέθεσε και επέβλεψε μέχρι τέλους την εκπόνηση της παρούσας διπλωματικής εργασίας. Την ευχαριστώ, ακόμη, που μου εμπιστεύτηκε το συγκεκριμένο θέμα και για την βοήθεια και τις γνώσεις που μου μετέδωσε.

Εν συνεχεία, ιδιαίτερες ευχαριστίες στον Χρηστίδη Ιωάννη για την ενθάρρυνση και την προθυμία του για συνεχείς συζητήσεις και τη βοήθεια του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Επίσης, ένα μεγάλο ευχαριστώ στον καθηγητή κ. Γκοτσόπουλο Αναστάσιο για τις γνώσεις που απέκτησα κατά τη διάρκεια φοίτησης στο τμήμα, για την υποστήριξη και την ενθάρρυνση.

Τέλος, θα ήθελα να εκφράσω την αγάπη μου στην οικογένειά μου για την συνεχή στήριξη, συμπαράσταση και κατανόηση σε όλη την διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία αποτελεί μια συνολική μελέτη πάνω στην τεχνολογία blockchain, εξερευνώντας την αρχιτεκτονική, τους τύπους αλυσίδων, τα χαρακτηριστικά, τα πλεονεκτήματα, τις εφαρμογές και τις διάφορες τεχνολογίες και εργαλεία που σχετίζονται με αυτή (Κεφάλαιο 1).

Στην συνέχεια, εξετάζεται το πρότυπο κρυπτονομίσματος NFT, οι αλγόριθμοι κοινής συναίνεσης, καθώς και ένα βασικό στοιχείο κάθε blockchain δικτύου, οι κομβόι (Κεφάλαια 2-4).

Στο τέλος του θεωρητικού σκέλους, γίνεται αναφορά στα δημοφιλέστερα blockchain δίκτυα, στην κρυπτογραφία και τέλος στα βασικά είδη κρυπτονομίσματος (Κεφάλαια 5-7). Ο σκοπός της εργασίας είναι να παρέχει μια ολοκληρωμένη κατανόηση του πεδίου.

Στο πρακτικό μέρος, η εργασία επικεντρώνεται στην εφαρμογή των Soulbound Tokens (SBTs) στον τομέα της αυθεντικοποίησης οντοτήτων, όπου αναπτύχθηκε ένα σύστημα που επιτρέπει σε εκπαιδευτικά ιδρύματα να δημιουργούν ψηφιακές ταυτότητες για φοιτητές και το προσωπικό τους (Κεφάλαια 8-14).

Το σύστημα χρησιμοποιεί δυο έξυπνα συμβόλαια. Το πρώτο ονόματι «CryptoPass», επιτρέπει την δημιουργία ψηφιακών ταυτοτήτων στην μορφή διακριτών SBT. Στα συγκεκριμένα SBTs, έχει αφαιρεθεί η ιδιότητα κατοχής πολλαπλών διακριτών του ιδίου είδους. Το δεύτερο έξυπνο συμβόλαιο, «AccessToken», παράγει NFTs μικρής διάρκειας, σε συνδυασμό με το επίπεδο πρόσβασης που προσδιορίζεται από τα μεταδεδομένα του SBT, ενισχύοντας την ασφάλεια και την αξιοπιστία του συστήματος. Τα παραγόμενα NFTs χρησιμοποιούνται κυρίως για την είσοδο σε φυσικές υπηρεσίες ή εγκαταστάσεις καθώς μπορούν να μετατρέπουν σε κωδικούς QR.

Η εργασία αναδεικνύει επίσης τη σημασία χρήσης ενός Web Server ως διαμεσολαβητή, προσφέροντας μια φιλική προς τον χρήστη προσέγγιση στην αλληλεπίδραση με την τεχνολογία blockchain.

Τέλος, αξίζει να αναφερθεί η βιβλιοθήκη «Web3Button» που παρέχει έναν εύκολο τρόπο ενσωμάτωσης του ανεπτυγμένου συστήματος σε υπάρχουσες ηλεκτρονικές υπηρεσίες, δίχως την ανάγκη ο προγραμματιστής να διαθέτει δεξιότητες πάνω στην τεχνολογία blockchain.

Abstract

This diploma thesis constitutes a comprehensive study on blockchain technology, exploring its architecture, types of chains, characteristics, advantages, applications, and various technologies and tools associated with it (Chapter 1).

Subsequently, the cryptocurrency standard NFT, consensus algorithms, as well as a fundamental element of every blockchain network, the nodes, are examined (Chapters 2-4).

At the end of the theoretical section, there is a reference to the most popular blockchain networks, cryptography, and finally to the basic types of cryptocurrencies (Chapters 5-7). The purpose of this work is to provide a comprehensive understanding of the field.

In the practical part, the work focuses on the application of Soulbound Tokens (SBTs) in the field of entity authentication, where a system was developed that allows educational institutions to create digital identities for students and their staff (Chapters 8-14).

The system uses two smart contracts. The first one, named "CryptoPass," allows the creation of digital identities in the form of distinct SBTs. For these specific SBTs, the property of owning multiple distinct tokens of the same kind has been removed. The second smart contract, "AccessToken," produces short-duration NFTs in conjunction with the access level determined by the SBT's metadata, enhancing the system's security and reliability. The produced NFTs are mainly used for entry into physical services or facilities as they can be converted into QR codes.

The work also highlights the importance of using a Web Server as an intermediary, offering a user-friendly approach to interacting with blockchain technology.

Finally, it is worth mentioning the "Web3Button" library, which provides an easy way to integrate the developed system into existing online services, without the need for the programmer to have skills in blockchain technology.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Κεφάλαιο 1: Τεχνολογία Blockchain

1.1. Αρχιτεκτονική Blockchain

Ένας κόμβος ξεκινά μια συναλλαγή σε ένα αποκεντρωμένο δίκτυο blockchain με τη χρήση μίας ψηφιακής υπογραφής και ταυτόχρονα με την κρυπτογράφηση τύπου ιδιωτικού κλειδιού. Ως συναλλαγή νοείται μια δομή δεδομένων που αντιπροσωπεύει τη μεταφορά ψηφιακών στοιχείων μεταξύ ομότιμων στο δίκτυο blockchain. Όλες οι συναλλαγές αποθηκεύονται σε ένα σύνολο μη επιβεβαιωμένων συναλλαγών και διαδίδονται στο δίκτυο με τη χρήση ενός πρωτόκολλο τύπου flooding, γνωστό και ως πρωτόκολλο Gossip. Στη συνέχεια, οι ομότιμοι κόμβοι πρέπει να επιλέξουν και να επικυρώσουν αυτές τις συναλλαγές με βάση ορισμένα προκαθορισμένα κριτήρια. Για παράδειγμα, οι κόμβοι προσπαθούν να επαληθεύσουν και να επικυρώσουν τις εν λόγω συναλλαγές ελέγχοντας εάν ένας εκκινητής διαθέτει όντως επαρκές υπόλοιπο για να ενεργοποιήσει μια συναλλαγή ή προσπαθεί να εξαπατήσει το σύστημα με την απόπειρα εγγραφής διπλής δαπάνης. Η διπλή δαπάνη καθίσταται δυνατή εφόσον υπάρξει χρήση του ίδιου ποσού εισροών για δύο ή περισσότερες διαφορετικές συναλλαγές [1]. Μόλις η συναλλαγή επαληθευτεί και επικυρωθεί από τον εξορύκτη αποτελεί πλέον μέρος μιας συστοιχίας (μπλοκ). Οι ομότιμοι κόμβοι που χρησιμοποιούν την υπολογιστική τους ισχύ για την εξόρυξη μπλοκ ονομάζονται miners (εξορύκτες) [2]. Οι κόμβοι εξόρυξης πρέπει να λύσουν έναν υπολογιστικό γρίφο και δαπανούν ένα μεγάλο μέρος των υπολογιστικών δυνατοτήτων τους για να καταστήσουν μια συστοιχία δημόσια. Ο εξορύκτης που θα λύσει πρώτος το γρίφο αποκτά την ευκαιρία να δημιουργήσει ένα νέο μπλοκ. Έτσι, η επιτυχής δημιουργία ενός νέου μπλοκ συνοδεύεται από ένα μικρό κίνητρο. Όλοι οι ομότιμοι κόμβοι στο δίκτυο επαληθεύουν στη συνέχεια το νέο μπλοκ χρησιμοποιώντας έναν μηχανισμό συναίνεσης, ο οποίος αποτελεί μια τεχνική που βοηθά ένα αποκεντρωμένο δίκτυο εναρμονιστεί σε διάφορα ζητήματα. Στη συνέχεια, το νέο μπλοκ προστίθεται στην υπάρχουσα αλυσίδα και στο τοπικό αντίγραφο των αμετάβλητων εγγραφών κάθε ομότιμου. Στο σημείο αυτό, η συναλλαγή είναι πλέον επιβεβαιωμένη. Το επόμενο μπλοκ συνδέεται με το μπλοκ που μόλις δημιουργήθηκε πρόσφατα χρησιμοποιώντας μια κρυπτογραφική συνάρτηση κατακερματισμού. Επομένως, το μπλοκ λαμβάνει την πρώτη επιβεβαίωση ενώ η συναλλαγή λαμβάνει τη δεύτερη επιβεβαίωση. Ομοίως, κάθε φορά που προστίθεται ένα νέο μπλοκ στην αλυσίδα συστοιχιών, η συναλλαγή επιβεβαιώνεται εκ νέου. Γενικά, μια συναλλαγή απαιτείται να λάβει έξι επιβεβαιώσεις στο δίκτυο προκειμένου να θεωρηθεί οριστική [3].

Μια συναλλαγή blockchain επομένως μπορεί να οριστεί ως μια μικρή μονάδα μιας εργασίας που αποθηκεύεται σε δημόσια αρχεία. Αυτές οι εγγραφές είναι επίσης γνωστές ως μπλοκ [4]. Αυτά τα μπλοκ εκτελούνται, υλοποιούνται και αποθηκεύονται σε blockchains προς επικύρωση από όλους τους εξορύκτες που συμμετέχουν στο δίκτυο blockchain. Κάθε προηγούμενη συναλλαγή μπορεί να ελεγχθεί ανά πάσα στιγμή, αλλά δεν μπορεί να καταστεί ενήμερη [5]. Το blockchain είναι η υποκείμενη τεχνολογία του κρυπτονομίσματος Bitcoin και διευκολύνει τις συναλλαγές που πραγματοποιούνται σε ένα παγκόσμιο δίκτυο διομοτίμων (peer to peer) με αποκεντρωμένο τρόπο. Αυτό κάνει το Bitcoin ένα ψηφιακό νόμισμα χωρίς σύνορα, ιδιαιτέρως δε ανθεκτικό στην κάθε είδους λογοκρισία. Το πλεονέκτημα της δημόσιας τεχνολογία blockchain αποτελεί το γεγονός δεν απαιτεί κανενός είδους επιπλέον εξασφάλιση κατά τη μεταβίβαση της ιδιοκτησίας των ψηφιακών περιουσιακών στοιχείων από έναν ομότιμο σε άλλον. Το blockchain είναι ένα σύστημα που παρέχει επαρκείς εγγυήσεις

ασφαλείας μέσω των λειτουργιών που διαδίδουν όλες τις ενέργειες εντός του δικτύου [6]. Η ασφάλεια είναι μια άλλη πτυχή που πρέπει να ληφθεί υπόψη ως πλεονέκτημα. Η εξόρυξη blockchain και οι μηχανισμοί συναίνεσης που βασίζονται σε μεγάλο βαθμό σε μια κρυπτογραφική συνάρτηση κατακερματισμού κρίνονται επαρκή για να αντιμετωπίσουν τα ανακύπτοντα ζητήματα ασφάλειας. Για παράδειγμα, το Bitcoin χρησιμοποιεί έναν ασφαλή αλγόριθμο κατακερματισμού 256 bit, γνωστό ως SHA-256 [7]. Το Bitcoin μπορεί να λάβει οποιονδήποτε τύπο εισαγωγής δεδομένων, όπως κείμενο, αριθμούς, συμβολοσειρά ή ακόμα και ένα αρχείο οποιουδήποτε μήκους και μεγέθους που δημιουργείται από υπολογιστή, για να παράξει 256 bit ή ένα αποτέλεσμα 64 χαρακτήρων που ονομάζεται κατακερματισμός [8]. Με την ίδια εισαγωγή, το μετατρεπόμενο αποτέλεσμα κατακερματισμού θα παραμένει πάντα ακριβώς ταυτόσημο. Ωστόσο, η οποιαδήποτε αλλαγή στην εισαγωγή θα αλλάξει εντελώς το αποτέλεσμα, η οποία ονομάζεται επίσης μονόδρομη συνάρτηση, που σημαίνει ότι εκ του αποτελέσματος, δεν είναι εφικτός ο υπολογισμός της εισαγωγής. Θα μπορούσε κάποιος μόνο να μαντέψει την ακριβή εισαγωγή των δεδομένων και οι πιθανότητες να συμβεί κάτι τέτοιο είναι απειροελάχιστες.

1.2. Τύποι αλυσίδων Blockchain

Υπάρχουν τέσσερις τύποι αλυσίδων blockchain:

1.2.1 Δημόσιες αλυσίδες Blockchains

Οι δημόσιες αλυσίδες συστοιχιών (blockchains) δεν υπόκεινται όπως φανερώνει η ονομασία σου σε κάποια άδεια, επιτρέπουν σε οποιονδήποτε να εγγραφεί σε αυτές και είναι πλήρως αποκεντρωμένες. Οι δημόσιες αλυσίδες συστοιχιών επιτρέπουν σε όλους τους κόμβους μιας συγκεκριμμένης blockchain να έχουν ίσα δικαιώματα πρόσβασης σε αυτή, να δημιουργούν νέες συστοιχίες δεδομένων και να τις επικυρώνουν.

Μέχρι σήμερα, οι δημόσιες blockchains χρησιμοποιούνται κυρίως για την ανταλλαγή και την εξόρυξη κρυπτονομισμάτων, όπως το Bitcoin, το Ethereum και το Litecoin. Σε αυτές τις δημόσιες αλυσίδες συστοιχιών, οι κόμβοι «εξορύσσονται» έναντι κρυπτονομισμάτων, δημιουργώντας συστοιχίες (blocks) για τις συναλλαγές που απαιτούνται από το δίκτυο μέσω της επίλυσης κρυπτογραφικών εξισώσεων. Σε αντάλλαγμα οι κόμβοι εξόρυξης κερδίζουν ένα μικρό ποσό κρυπτονομισμάτων. Οι εξορύκτες ουσιαστικά λειτουργούν ως οιονεί ταμίες τραπεζών που δημιουργούν μια συναλλαγή και λαμβάνουν (ή «εξορύσσουν») συγκεκριμμένο τίμημα για τις προσπάθειές τους.

1.2.2 Ιδιωτικές (ή διαχειριζόμενες) αλυσίδες συστοιχιών (blockchains)

Οι ιδιωτικές αλυσίδες συστοιχιών, οι οποίες ονομάζονται και διαχειριζόμενες αλυσίδες συστοιχιών, είναι αλυσίδες συστοιχιών που υπόκεινται σε άδεια και ελέγχονται από έναν μόνο οργανισμό/φορέα. Σε μια ιδιωτική blockchain, η κεντρική αρχή καθορίζει ποιος μπορεί να αποτελεί ένα κόμβο. Η κεντρική αρχή επίσης δεν εκχωρεί απαραίτητα σε κάθε κόμβο ίσα δικαιώματα για την εκτέλεση λειτουργιών εντός του δικτύου. Οι ιδιωτικές αλυσίδες συστοιχιών είναι μόνο εν μέρει αποκεντρωμένες επειδή η δημόσια πρόσβαση σε αυτές είναι περιορισμένη. Μερικά παραδείγματα ιδιωτικών blockchain είναι το επιχειρηματικό (business-

to-business) δίκτυο ανταλλαγής εικονικών νομισμάτων Ripple and Hyperledger, ένα έργο-ομπρέλα εφαρμογών blockchain ανοιχτού κώδικα.

Τόσο οι ιδιωτικές όσο και οι δημόσιες αλυσίδες συστοιχιών έχουν μειονεκτήματα. Οι δημόσιες blockchain τείνουν να έχουν μεγαλύτερους χρόνους επικύρωσης για νέα δεδομένα από τα ιδιωτικά blockchain και τα ιδιωτικά blockchain είναι πιο ευάλωτα σε απάτες και κακόβουλες ενέργειες. Για την αντιμετώπιση αυτών των μειονεκτημάτων, αναπτύχθηκαν οι κοινοπραξίες blockchains, καθώς και οι υβριδικές αλυσίδες συστοιχιών, που αναλύονται κάτωθι.

1.2.3 Blockchain κοινοπραξίας

Οι blockchain κοινοπραξίας είναι blockchain που υπόκεινται σε άδεια και τα οποία διακυβερνώνται από μια ομάδα οργανισμών/φορέων και όχι από έναν οργανισμό, όπως στην περίπτωση των ιδιωτικών blockchain. Ως εκ τούτου, οι blockchain κοινοπραξιών απολαμβάνουν μεγαλύτερη αποκέντρωση από τα ιδιωτικά blockchain, με αποτέλεσμα υψηλότερα επίπεδα ασφάλειας. Ωστόσο, η σύσταση κοινοπραξιών μπορεί να είναι μια δύσκολη διαδικασία, καθώς απαιτεί συνεργασία μεταξύ ορισμένων οργανισμών, η οποία παρουσιάζει υλικοτεχνικές προκλήσεις καθώς και δυνητικό αντιμονοπωλιακό κίνδυνο Ένα παράδειγμα blockchain κοινοπραξίας αποτελεί η CargoSmart, η οποία έχει αναπτύξει το Global Shipping Business Network Consortium, μια μη κερδοσκοπική κοινοπραξία blockchain που στοχεύει στην ψηφιοποίηση της ναυτιλιακής βιομηχανίας και στο να επιτρέπει στους φορείς της εν λόγω βιομηχανίας να συνεργάζονται πιο αποτελεσματικά.

1.2.4 Υβριδικές blockchains

Οι υβριδικές blockchains είναι αλυσίδες συστοιχιών που ελέγχονται από έναν μόνο οργανισμό, αλλά με ένα επίπεδο εποπτείας που πραγματοποιείται από δημόσια blockchain, η χρήση της οποίας απαιτείται για την εκτέλεση ορισμένων επικυρώσεων συναλλαγών. Ένα παράδειγμα υβριδικής blockchain είναι το IBM Food Trust, το οποίο αναπτύχθηκε για να βελτιώσει την αποτελεσματικότητα σε ολόκληρη την αλυσίδα εφοδιασμού τροφίμων.

1.3 Χαρακτηριστικά της τεχνολογίας Blockchain

1.3.1 Αποκέντρωση

Στα συμβατικά κεντρικά συστήματα συναλλαγών, κάθε συναλλαγή πρέπει να επικυρώνεται μέσω μιας κεντρικής υπηρεσίας που διαθέτει αξιοπιστία και χαίρει καθολικής αναγνώρισης (π.χ. της κεντρικής τράπεζας). Επομένως, η αποκέντρωση απαιτεί εμπιστοσύνη, που είναι το κύριο ζήτημα, μαζί με την ανθεκτικότητα συλλογής, τη διαθεσιμότητα και την ανακατεύθυνση. Στο σημείο αυτό η αποκεντρωμένη αρχιτεκτονική blockchain peer-to-peer θα μπορούσε να αποτελεί μια προτιμότερη λύση. Σε αντίθεση με ένα κεντρικό σύστημα, μια συναλλαγή στο δίκτυο blockchain μπορεί να πραγματοποιηθεί μεταξύ οποιωνδήποτε δύο

ομότιμων (P2P – peer to peer) χωρίς έλεγχο ταυτότητας από την κεντρική υπηρεσία. Με αυτόν τον τρόπο, το blockchain μπορεί να μειώσει την περιρρέουσα ανησυχία για ζητήματα εμπιστοσύνης χρησιμοποιώντας διάφορες διαδικασίες συναίνεσης. Επιπλέον, μπορεί να μειώσει το κόστος του διακομιστή (συμπεριλαμβανομένου του κόστους ανάπτυξης και του κόστους λειτουργίας) και να μετριάσει τα σημεία συμφόρησης απόδοσης στον κεντρικό διακομιστή. Αντίθετα, σε πολλές περιπτώσεις, το blockchain έχει κάποια μειονεκτήματα. Για παράδειγμα, σε μηχανισμούς συναίνεσης τύπου Proof of Work όπως τα κρυπτονομίσματα Bitcoin και Ethereum, το κόστος που αφορά στο διακομιστή (server) και στις δαπάνες ενέργειας είναι τάξεις μεγέθους υψηλότερες, ενώ η απόδοση είναι επίσης αρκετές τάξεις μεγέθους χαμηλότερη.

1.3.2 Ανθεκτικότητα – Διατηρησιμότητα

Η τεχνολογία blockchain παρέχει την κατάλληλη υποδομή για την πραγματοποίηση επαληθεύσεων [9] και επιτρέπει στους παραγωγούς καθώς και στους καταναλωτές να αποδείξουν ότι τα δεδομένα τους είναι αυθεντικά και δεν έχουν αλλοιωθεί. Για παράδειγμα, εάν ένα blockchain αποτελείται από 10 μπλοκ, τότε το μπλοκ αρ. 10 περιέχει τον κατακερματισμό του προηγούμενου μπλοκ της ακολουθίας και για τη δημιουργία ενός νέου μπλοκ, χρησιμοποιούνται τα δεδομένα του τρέχοντος μπλοκ. Επομένως, όλα τα μπλοκ συνδέονται και συνδέονται μεταξύ τους στην υπάρχουσα αλυσίδα. Ακόμη και οι συναλλαγές σχετίζονται με την προηγούμενη συναλλαγή. Η οποιαδήποτε ενημέρωση για οποιαδήποτε συναλλαγή θα αλλάξει σημαντικά τον κατακερματισμό του μπλοκ. Εάν κάποιος θέλει να τροποποιήσει οποιοδήποτε δεδομένο, θα πρέπει να αλλάξει όλα τα δεδομένα κατακερματισμού του προηγούμενου μπλοκ, κάτι που θεωρείται αστρονομικά δύσκολο έργο, λαμβάνοντας υπόψη τον όγκο της εργασίας που ενέχει μια τέτοια απόπειρα. Επιπλέον, μετά τη δημιουργία ενός μπλοκ από έναν εξορύκτη, αυτή επιβεβαιώνεται από τους λοιπούς χρήστες του δικτύου. Ως εκ τούτου, οποιαδήποτε παραποίηση ή πλαστογραφία δεδομένων θα εντοπιστεί από το δίκτυο. Για αυτόν τον λόγο, το blockchain θεωρείται σχεδόν απαραβίαστο ως τεχνολογία.

1.3.3 Ανωνυμία

Είναι δυνατή η αλληλεπίδραση με το δίκτυο blockchain μέσω μια τυχαίως γεννηθείσας διεύθυνσης (randomly generated address) [10]. Ένας χρήστης μπορεί να διαθέτει πολλές διευθύνσεις μέσα σε ένα δίκτυο Blockchain για να αποφύγει την έκθεση της ταυτότητάς του. Καθώς πρόκειται για ένα αποκεντρωμένο σύστημα, καμία κεντρική αρχή δεν παρακολουθεί ή καταγράφει τις προσωπικές πληροφορίες των χρηστών. Το blockchain παρέχει ένα ορισμένο ποσοστό ανωνυμίας μέσω του ιδιάζοντος περιβάλλοντός που δημιουργεί.

1.3.4 Ελεγξιμότητα

Όλες οι συναλλαγές που πραγματοποιούνται σε ένα δίκτυο blockchain καταγράφονται από ένα ψηφιακά κατανεμημένο αρχείο εγγραφών και επικυρώνονται από μια ψηφιακή χρονική σήμανση. Ως αποτέλεσμα, είναι δυνατός ο έλεγχος και η ανίχνευση προηγούμενων εγγραφών

με πρόσβαση σε οποιονδήποτε κόμβο στο δίκτυο [11]. Για παράδειγμα, όλες οι συναλλαγές θα μπορούσαν να ανιχνευθούν ως επαναλαμβανόμενες στο Bitcoin, κάτι που διευκολύνει τον έλεγχο και τη διαφάνεια της κατάστασης δεδομένων στο blockchain. Ωστόσο, με τη διοχέτευση χρημάτων σε πολλούς λογαριασμούς, γίνεται πολύ δύσκολο να εντοπιστεί η προέλευσή τους.

1.4 Πλεονεκτήματα και μειονεκτήματα της τεχνολογίας Blockchain

Τα πλεονεκτήματα της τεχνολογίας blockchain είναι τα κάτωθι:

- Ακεραιότητα δεδομένων: Οι τεχνολογίες blockchain έχουν σχεδιαστεί κατά τέτοιο τρόπο ώστε οποιαδήποτε συστοιχία ή ακόμα και μια συναλλαγή που προσθέτει στην αλυσίδα να μην μπορεί να εισαχθεί προς επεξεργασία, κάτι που τελικά παρέχει πολύ υψηλό εύρος ασφάλειας.
- Έλλειψη λογοκρισίας: Η τεχνολογία blockchain είναι απαλλαγμένη από λογοκρισία καθώς κανείς δεν έχει τον έλεγχο ενός μεμονωμένου μέρους της, αλλά αντιθέτως χρησιμοποιεί την έννοια των αξιόπιστων κόμβων για επικύρωση, καθώς και πρωτόκολλα συναίνεσης που επικυρώνουν συναλλαγές χρησιμοποιώντας έξυπνα συμβόλαια.
- Επαληθευσιμότητα: Η τεχνολογία blockchain χρησιμοποιείται για την αποθήκευση πληροφοριών με αποκεντρωμένο τρόπο, ώστε ο καθένας να μπορεί να επαληθεύσει την ορθότητα των πληροφοριών χρησιμοποιώντας απόδειξη τύπου μηδενικής γνώσης μέσω της οποίας ένα μέρος αποδεικνύει την ορθότητα των δεδομένων σε ένα άλλο μέρος χωρίς να αποκαλύπτει τίποτα σχετικά με τα δεδομένα αυτά καθαυτά.
- Εύρος διανομής: Δεδομένου ότι τα δεδομένα των blockchains αποθηκεύονται συχνά σε χιλιάδες συσκευές σε ένα κατανεμημένο δίκτυο κόμβων, το σύστημα και τα δεδομένα είναι εξαιρετικά ανθεκτικά σε τεχνικές βλάβες και κακόβουλες επιθέσεις. Κάθε κόμβος δικτύου μπορεί να αναπαραγάγει και να αποθηκεύσει ένα αντίγραφο της βάσης δεδομένων και, λόγω αυτού, δεν υπάρχει καμμία πιθανότητα αποτυχίας.
- Ιχνηλασιμότητα: Η μορφή μιας blockchain έχει σχεδιαστεί έτσι ώστε να δημιουργεί μια μη αναστρέψιμη διαδρομή ελέγχου, καθιστώντας εύκολο και προσιτό τον εντοπισμό οποιασδήποτε προσθήκης στην εν λόγω αλυσίδα.
- Αμεταβλητότητα: Τα δεδομένα δεν μπορούν να παραβιαστούν στην τεχνολογία blockchain λόγω της αποκεντρωμένης δομής της, επομένως οποιαδήποτε αλλαγή αντικατοπτρίζεται άμεσα σε όλους τους κόμβους, ώστε να μην μπορεί κανείς να προβεί στη διάπραξη απάτης.
- Προσβασιμότητα: Ένα από τα σημαντικότερα πλεονεκτήματα της τεχνολογίας blockchain είναι ότι είναι προσβάσιμη με κάθε τρόπο και ο οποιοσδήποτε μπορεί να συμμετέχει με τη συνεισφορά του στην τεχνολογία blockchain. Δεν χρειάζεται καμία άδεια από κανέναν για να συμμετέχει ο οποιοσδήποτε στο κατανεμημένο δίκτυο.
- Σταθερότητα: Από τη στιγμή που τα δεδομένα έχουν καταχωρηθεί στη blockchain, είναι εξαιρετικά δύσκολο να αφαιρεθούν ή να τροποποιηθούν. Αυτό καθιστά το blockchain μια εξαιρετική τεχνολογία για την αποθήκευση οικονομικών αρχείων ή οποιωνδήποτε άλλων δεδομένων όπου απαιτείται τήρηση αρχείου ελέγχου, επειδή

κάθε αλλαγή παρακολουθείται και καταγράφεται μόνιμα σε ένα κατανεμημένο και δημόσιο αρχείο.

- Ασφάλεια: Η τεχνολογία blockchain είναι εξαιρετικά ασφαλής καθώς σε κάθε μέλος του δικτύου blockchain παρέχεται μια μοναδική ταυτότητα που συνδέεται με τον λογαριασμό του. Επίσης, η κρυπτογράφηση των συστοιχιών εντός της αλυσίδας καθιστά πιο δύσκολο για κάθε επίδοξο ψηφιακό πειρατή να διαταράξει την παραδοσιακή δομή της αλυσίδας
- Ταχύτητα επεξεργασίας: Πριν από την εφεύρεση των blockchains, ο μέσος τραπεζικός οργανισμός χρειαζόταν πολύ χρόνο για την επεξεργασία και την εισαγωγή των συναλλαγών που πραγματοποιούνταν στο σύστημά του, αλλά μετά την τεχνολογία blockchain η ταχύτητα των συναλλαγών αυξήθηκε σε πολύ μεγάλο βαθμό. Πριν την εισαγωγή της τεχνολογίας blockchain, ο συνολικός απαιτούμενος χρόνος για την ολοκλήρωση μιας τραπεζικής πράξης ήταν περίπου τρεις ημέρες, αλλά μετά την εισαγωγή της τεχνολογίας blockchain, ο χρόνος αυτός μειώθηκε σε λίγα λεπτά ή και δευτερόλεπτα.
- Αδυναμία παρέμβασης τρίτων: Καμία κυβέρνηση ή χρηματοπιστωτικό ίδρυμα δεν έχει τον έλεγχο των κρυπτονομισμάτων που προκύπτουν στην τεχνολογία blockchain. Αυτό σημαίνει ότι καμία κυβέρνηση δεν μπορεί να επηρρεάσει την αξία κάποιου κρυπτονομίσματος.

Τα μειονεκτήματα της τεχνολογίας blockchain είναι τα κάτωθι:

- Υψηλή ενεργειακή κατανάλωση και κόστος: Η κατανάλωση ενέργειας στην τεχνολογία blockchain είναι συγκριτικά υψηλή λόγω των δραστηριοτήτων εξόρυξης. Η διατήρηση αρχείου σε πραγματικό χρόνο είναι ένας από τους λόγους αυτής της κατανάλωσης, επειδή κάθε φορά που δημιουργεί έναν νέο κόμβο, επικοινωνεί και με κάθε άλλο κόμβο ταυτόχρονα. Κάθε συναλλαγή κρυπτογράφησης έχει επίσης υψηλές ενεργειακές απαιτήσεις. Υπάρχουν πολύ μικρές πιθανότητες να επιλυθεί αυτό το ζήτημα με την πρόοδο της τεχνολογίας, όπως επίσης και το πρόβλημα αποθήκευσης του δικτύου που συναρτάται άμεσα με το υψηλό ενεργειακό κόστοςμπορεί να καλύπτεται από ενεργειακά ζητήματα που δεν μπορούν να επιλυθούν.
- Έλλειψη εξοικείωσης: Η τεχνολογία blockchain είναι μια τεχνολογία πρόσφατη, επομένως οι άνθρωποι δεν έχουν ακόμα μεγάλη εμπιστοσύνη σε αυτή, δεν είναι έτοιμοι να επενδύσουν σε αυτή. Αρκετές εφαρμογές της τεχνολογίας blockchain λειτουργούν ήδη εξαιρετικά σε διαφορετικούς κλάδους, αλλά εξακολουθεί να χρειάζεται να κερδηθεί η εμπιστοσύνη περισσότερων ανθρώπων ώστε καταστούν αναγνωρίσιμα τα οφέλη της πλήρους αξιοποίησής της.
- Νομικά ζητήματα: Σε όλες τις χώρες του κόσμου, τα σύγχρονα χρηματοοικονομικά προϊόντα ελέγχονται από την κεντρική κυβέρνηση της. Υπάρχουν νομικά εμπόδια για τα κρυπτονομίσματα ώστε να γίνουν αποδεκτό από τα προϋπάρχοντα παραδοσιακά χρηματοπιστωτικά ιδρύματα.
- Επιθέσεις τύπου 51%: Ο αλγόριθμος συναίνεσης Proof of Work που προστατεύει τα κρυπτονομίσματα όπως το Bitcoin στην τεχνολογία blockchain έχει αποδειχθεί πολύ αποτελεσματικός με την πάροδο του χρόνου. Ωστόσο, υπάρχουν ορισμένοι τύποι επιθέσεων που μπορούν να εκτελεστούν εναντίον δικτύων blockchain και οι επιθέσεις 51% είναι από τις πιο κοινές εξ αυτών. Μια τέτοια επίθεση μπορεί να

συμβεί εάν ένας οργανισμός καταφέρει να ελέγξει περισσότερο από το 50% της ισχύος κατακερματισμού του δικτύου, κάτι που θα του επέτρεπε τελικά να διακόψει το δίκτυο αποκλείοντας ή τροποποιώντας σκόπιμα τη ροή των συναλλαγών.

- Μη βιωσιμότητα δικτύου για χρήσεις ειδικού σκοπού: Από τη φύση της, η τεχνολογία blockchain χρησιμοποιεί μια αυστηρή επιχειρηματική λογική που δεν επιτρέπει τον επανασχεδιασμό της χωρίς την απώλεια των πλεονεκτημάτων της. Χρειάζεται η ανάπτυξη δυνατότητας πραγματοποίησης έλλογων επιχειρηματικών αλλαγών που να είναι ταυτόχρονα συμβατές με την τεχνολογία blockchain.
- Δυσκολία Ανάπτυξης: Η εφαρμογή πολύ περίπλοκων πρωτοκόλλων για να επιτευχθεί συναίνεση και να επιτρέπεται η κλιμάκωση των συναλλαγών από την αρχή είναι αναγκαία. Δεν μπορεί κανείς να εφαρμόσει βιαστικά μια ιδέα στην τεχνολογία blockchain με την ελπίδα να προσθέσει αργότερα νέες δυνατότητες και να επεκτείνει την εφαρμογή χωρίς μια σοβαρή αναδιάταξη του δικτύου ή και διακλάδωση του.
- Αναποτελεσματικότητα: Οι blockchains, ειδικά αυτές που χρησιμοποιούν αλγόριθμο τύπου Proof-of-Work, είναι εξαιρετικά αναποτελεσματικές. Δεδομένου ότι η εξόρυξη είναι εξαιρετικά ανταγωνιστική διαδικασία και ανακυρήσσεται μόνο ένας νικητής κάθε δέκα λεπτά, η προσπάθεια κάθε άλλου εξορύκτη δεν ανταμοίβεται καθόλου.
- Ζητήματα αποθήκευσης: Τα αρχεία των blockchain μπορούν να μεγαλώσουν πολύ σε όγκο με την πάροδο του χρόνου. Η συνεχής αύξηση στα μεγεθη των blockchains φαίνεται να ξεπερνά την ανάπτυξη στους σκληρούς δίσκους και το δίκτυο κινδυνεύει να χάσει κόμβους εάν το αρχείο μεγαλώσει σε σημείο που δεν μπορούν να το μεταφορτώσουν και να το αποθηκεύσουν τα μέλη του.
- Επεκτασιμότητα: Είναι ένα από τα μεγαλύτερα μειονεκτήματα της τεχνολογίας blockchain είναι η αδυναμία κλιμάκωσης λόγω του σταθερού μεγέθους ανα συστοιχία για την αποθήκευση δεδομένων. Το μέγεθος κάθε συστοιχίας είναι 1 MB και η χωρητικότητα αυτή είναι αρκετή για την αποθήκευση ελάχιστων συναλλαγών ανά συστοιχία.

1.5 Εφαρμογές της τεχνολογίας Blockchain

Η τεχνολογία blockchain έχει πολλαπλές εφαρμογές. Καταρχάς, είναι σημαντικό να κατανοήσουμε ότι το κρυπτονόμισμα bitcoin δεν είναι ταυτόσημο με την τεχνολογία blockchain. Αντίθετα, αποτελεί μια από τις πιο επιτυχημένες εφαρμογές της τεχνολογίας blockchain [12]. Το Bitcoin είναι ένα κρυπτογραφικό ψηφιακό νόμισμα, το οποίο συναλλάσσεται μέσω ενός ανοιχτού, δημόσιου και ανώνυμου δικτύου blockchain. Ωστόσο, οι ειδικοί υποστηρίζουν ότι αυτή η τεχνολογία μπορεί να εφαρμοστεί για την εύρεση λύσεων για πλειάδα έτερων τομέων, όπως η υγειονομική περίθαλψη, η εκπαίδευση, η διενέργεια ψηφοφοριών, η διαχείριση στοιχείων ταυτότητας, η διακυβέρνηση, η αλυσίδα εφοδιασμού, οι ενεργειακοί πόροι κ.λπ. Επίσης, ορισμένοι οραματιστές προβλέπουν ότι η τεχνολογία blockchain μπορεί να επηρεάσει την ψηφιακή σφαίρα παρόμοια με το διαδίκτυο [13]. Όταν πρωτοεμφανίστηκε το Διαδίκτυο, δεν είχαμε ιδέα πώς θα άλλαζε για πάντα τη ζωή μας. Από την κατασκευή των έξυπνων τηλέφωνων και τα μηνύματα κειμένου έως τη δυνατότητα προβολής ταινιών συνεχούς ροής (streaming) και βιντεοδιασκέψεων με αγαπημένα πρόσωπα, καθώς και η διαδικτυακή συμμετοχή σε συναντήσεις ή συνεντεύξεις, κανείς δεν γνώριζε πώς θα άλλαζε ο κόσμος με την εφεύρεση του Διαδικτύου. Αυτήν τη στιγμή βρισκόμαστε στα πρώτα στάδια του blockchain και υπάρχουν ακόμη πολλές δυνατότητες της τεχνολογίας που μπορούν να εξελιχθούν, όπως εκτίθεται στα παρακάτω παραδείγματα:

1.5.1 Υγειονομική περίθαλψη

Η τεχνολογία κατανεμημένων αρχείων έχει τη δυνατότητα να μεταμορφώσει τις υπηρεσίες υγείας [14]. Η τεχνολογία blockchain μπορεί να χρησιμοποιηθεί για την ιχνηλασιμότητα των φαρμάκων και τη διαχείριση ιατρικών δεδομένων των ασθενών. Η παραποίηση φαρμάκων αποτελεί ένα σημαντικό πρόβλημα για τη φαρμακοβιομηχανία. Εκθέσεις από τον οργανισμό Health Research Funding αποκάλυψαν ότι το 10% έως το 30% των φαρμάκων που πωλούνται στις αναπτυσσόμενες χώρες είναι πλαστά [15]. Υπολογίζεται από τον ΠΟΥ ότι το 16% των πλαστών φαρμάκων έχουν λανθασμένη σύσταση, ενώ το 17% περιέχει ένα ανακριβές επίπεδο βασικών συστατικών ουσιών. Ως εκ τούτου, τα εν λόγω φάρμακα μπορούν να θέσουν τη ζωή ενός ασθενούς σε κίνδυνο, καθώς δεν οδηγούν σε ίαση, αντίθετα είναι δυνατόν να προκαλέσουν δευτερογενείς παρενέργειες που μπορούν να οδηγήσουν σε θάνατο. Από οικονομικής άποψης, η παραποίηση φαρμάκων ευθύνεται για ετήσια απώλεια εκτιμώμενου ύψους 10,2 δισεκατομμυρίων ευρώ για τους ευρωπαϊκούς φαρμακευτικούς οργανισμούς [16]. Η τεχνολογία blockchain μπορεί να αποτελέσει λύση για την αντιμετώπιση αυτού του ζητήματος, επειδή όλες οι συναλλαγές που προστίθενται στο κατανεμημένο καθολικό αρχείο είναι αμετάβλητες και έχουν ψηφιακή χρονοσήμανση, γεγονός που καθιστά δυνατή την παρακολούθηση ενός προϊόντος και την προστασία των πληροφοριών. Η ασφαλής διαχείριση των δεδομένων των ασθενών είναι μία από τις κύριες ανησυχίες για τον κλάδο της υγειονομικής περίθαλψης [17]. Για την παροχή εξατομικευμένων θεραπειών, είναι απαραίτητο να υπάρχει πρόσβαση στο πλήρες ιατρικό ιστορικό κάθε ασθενούς. Είναι κοινά αποδεκτό ότι τα ιατρικά δεδομένα είναι ευαίσθητα και απαιτούν για τη διαχείρισή τους μια ασφαλή πλατφόρμα κοινής χρήσης. Το υπάρχον σύστημα τήρησης ιατρικών αρχείων στερείται ιδιωτικότητας καθώς και διαλειτουργικότητας. Επί του παρόντος, η τεχνολογία blockchain μπορεί να προσφέρει το εφαλτήριο για την ενοποίηση των ιατρικών αρχείων μεταξύ διαφορετικών δομών υγειονομικής περίθαλψης, καθώς και την απαιτούμενη ακεραιότητα δεδομένων, μέσω της δημιουργίας ενός ισχυρού και ασφαλούς πλαισίου αποθήκευσης ψηφιακών ιατρικών αρχείων που εγγυάται την προσφορά διαφανών και ποιοτικών υπηρεσιών υγείας για τους ασθενείς, καθώς και μείωση του κόστους θεραπείας τους.

1.5.2 Ενέργεια

Μία από τις κύριες χρήσεις του blockchain όσον αφορά στην ενέργεια μπορεί να πραγματοποιηθεί στα μικροδίκτυα. Ένα μικροδίκτυο είναι ένα τοπικό σύνολο πηγών ηλεκτρικής ενέργειας και φορτίων που ενσωματώνονται και αποτελούν αντικείμενο διαχείρισης με στόχο τη βελτίωση της απόδοσης και της αξιοπιστίας της παραγωγής και της κατανάλωσης ενέργειας [18]. Οι πηγές ηλεκτρικής ενέργειας μπορούν να είναι κατανεμημένες γεννήτριες ενέργειας, σταθμοί ανανεώσιμων πηγών ενέργειας και εξοπλισμός αποθήκευσης ενέργειας σε εγκαταστάσεις που δημιουργούνται και ανήκουν σε διαφορετικούς οργανισμούς ή παρόχους ενέργειας. Ένα από τα κύρια πλεονεκτήματα της τεχνολογίας μικροδικτύων είναι ότι όχι μόνο επιτρέπει στους κατοίκους και άλλους καταναλωτές ηλεκτρικής ενέργειας, όπως τους βιομηχανικούς πελάτες, να έχουν πρόσβαση στην απαιτούμενη ενέργεια, αλλά μπορούν επίσης να παράγουν και να πωλούν πλεονάζουσα ενέργεια στο δίκτυο. Η τεχνολογία blockchain μπορεί να χρησιμοποιηθεί για την καταγραφή

και την επικύρωση συναλλαγών πώλησης και αγοράς ενέργειας σε μικροδίκτυα [19]. Με παρόμοιο τρόπο, η τεχνολογία blockchain μπορεί να χρησιμοποιηθεί σε μεγαλύτερη κλίμακα για να επιτρέψει την εμπορία ενέργειας σε έξυπνα δίκτυα. Σε περίπτωση ανάπτυξης έξυπνων δικτύων εξοπλισμένων με αμφίδρομη ροή επικοινωνίας, το blockchain μπορεί να χρησιμοποιηθεί για να υποστηρίξει την ασφαλή παρακολούθηση της κατανάλωσης και την εμπορία ενέργειας με διατήρηση της ιδιωτικής ζωής χωρίς την ανάγκη ενδιάμεσων στην εν λόγω αγορά. Τα λεγόμενα «έξυπνα» συμβόλαια θα μπορούσαν να χρησιμοποιηθούν για να διασφαλιστούν οι προγραμματικές περιγραφές των αναμενόμενων βαθμών ευελιξίας ισχύος, η επικύρωση και η προσιτότητα των συμφωνιών ανταπόκρισης στη ζήτηση και η ισορροπία μεταξύ των αναγκών ενέργειας και της παραγωγής. Επιπλέον, το blockchain μπορεί να χρησιμοποιηθεί για να επιτρέψει την εμπορία ενέργειας στο βιομηχανικό Internet of Things (IoT) [20]. Γενικά, η χρήση του blockchain σε εφαρμογές που σχετίζονται με την ενέργεια έχει τη δυνατότητα να μειώσει το ενεργειακό κόστος και να αυξήσει την ανθεκτικότητα της αγοράς.

1.5.3 Χρηματοοικονομικές αγορές

Η τεχνολογία blockchain θα μπορούσε να λύσει προβλήματα στο κατακερματισμένο περιβάλλον μιας κεφαλαιαγοράς, όπως π.χ. η διαλειτουργικότητα, η εμπιστοσύνη και η διαφάνεια [21]. Λόγω του ρόλου των διαμεσολαβητών της αγοράς και του ρυθμιστικού πλαισίου, χρειάζονται περισσότερες από τρεις ημέρες για την ολοκλήρωση και την οριστικοποίηση όλων των σχετικών συναλλαγών. Ως αποτέλεσμα, οι επενδυτές υποβάλλονται σε μια δυσκίνητη διαδικασία. Το blockchain μπορεί να είναι η λύση από αυτή την άποψη. Μπορεί να κάνει το χρηματιστήριο αρτιότερο μέσω της αποκέντρωσης και της αυτοματοποίησης [22]. Εξαλείφοντας τους μεσάζοντες και επιταχύνοντας την εκκαθάριση των συναλλαγών, το blockchain μπορεί να συμβάλει στη μείωση του σχετικού κόστους. Επιπλέον, η τεχνολογία μπορεί να διευκολύνει την επίπονη γραφειοκρατία που ενέχει η εμπορική και νομική διαδικασία μεταβίβασης ιδιοκτησίας των άυλων τίτλων. Με την εισαγωγή των έξυπνων συμβολαίων, το blockchain απαλοίφει την ανάγκη οποιουδήποτε τρίτου προσώπου και αρχής ως ρυθμιστή της αγοράς, ενεργώντας αφ’εαυτού ως ρυθμιστής για το σύνολο των συναλλαγών.

1.5.4 Ψηφοφορίες

Η τεχνολογία blockchain μπορεί να χρησιμοποιηθεί σε διαφορετικά πεδία ως λύση σε προβλήματα που μπορεί να εμφανίζει μια τυπική βάση δεδομένων. Το 2019 αποκαλύφθηκε ότι ένας μεγάλος κατασκευαστής εκλογικών μηχανημάτων στις ΗΠΑ είχε εγκαταστήσει λογισμικό απομακρυσμένης πρόσβασης σε ορισμένα συστήματα του [23]. Αυτό το λογισμικό επέτρεπε την αλλαγή των ψήφων κατά την καταμέτρηση του συνόλου. Περιπτώσεις όπως αυτή δημιουργούν έλλειψη εμπιστοσύνης στο εκλογικό σύστημα της Αμερικής, όπως αποδεικνύεται σε σχετικές δημοσκοπήσεις, οι οποίες καταλήγουν ότι μόνο το ένα τέταρτο περίπου των Αμερικανών πολιτών αισθάνονται σίγουροι ότι η ψήφος τους μετράται. Το Blockchain θα έλυνε αυτό το ζήτημα παρέχοντας ένα κατανεμημένο ψηφιακό αρχείο που θα διασφάλιζε την καταμέτρηση των ψήφων, καθώς το σύνολο δεδομένων του αρχείου που κατέχει κάθε ψηφοφόρος είναι το ίδιο με αυτό που μετράει στο συνολικό αποτέλεσμα.

1.5.5 Εκπαίδευση

Προς το παρόν, τα ακαδημαϊκά ιδρύματα που έχουν υιοθετήσει την τεχνολογία blockchain τη χρησιμοποιούν κυρίως για την αποθήκευση και την κοινή χρήση ακαδημαϊκών αρχείων και διαπιστευτηρίων. Ωστόσο, η τεχνολογία θα μπορούσε να φέρει επανάσταση στην εκπαίδευση με διάφορους τρόπους: ενίσχυση των ευκαιριών για δια βίου μάθηση, δημιουργία μεγαλύτερης αποτελεσματικότητας για τους εκπαιδευτικούς μέσω έξυπνων συμβάσεων και δυνατότητα για τους μαθητές να έχουν την ιδιοκτησία των ακαδημαϊκών τους αρχείων, μεταξύ άλλων. Αν και οι δυνατότητες είναι ελπιδοφόρες, ζητήματα όπως η ασφάλεια δεδομένων, οι δυνατότητες επέκτασης/κλιμάκωσης, αλλά και το υποκείμενο κόστος αποτελούν πραγματικές προκλήσεις για την ευρεία υιοθέτηση της τεχνολογίας blockchain στον εκπαιδευτικό τομέα. Η τεχνολογία blockchain μπορεί επίσης να βελτιώσει την ποιότητα της διαδικτυακής εκπαίδευσης, ενισχύοντας τις διαδικασίες ακαδημαϊκής διαπίστευσης. Η σημερινή ευκολία προσφοράς ψηφιακής εκπαίδευσης μπορεί ενίοτε να οδηγήσει τους μαθητές σε μη διαπιστευμένους ή μη πιστοποιημένους φορείς εκπαίδευσης που λειτουργούν σε φαινομενικά ίσους όρους ανταγωνισμού με ακαδημαϊκός επαρκή ιδρύματα, θέτοντας τους έτσι σε κίνδυνο την αξιοπιστία των τίτλων εκπαίδευσής τους.

Κεφάλαιο 2: Non-Fungible Tokens - NFT

7.1 Ορισμός - Ιστορικό των NFT

Τα NFTs βασίζονται στην τεχνολογία blockchain και αποδίδονται στην ελληνική γλώσσα ως Μη Ανταλλάξιμα Διακριτικά, εναλλακτικά ως Μη Εναλλάξιμα Κρυπτοπαραστατικά. Αποτελούν μοναδικά αναγνωρίσιμες ψηφιακές αναπαραστάσεις φυσικών ή ψηφιακών αντικειμένων. Συνήθως, τα NFTs δεν μπορούν να διαιρεθούν σε μικρότερες μονάδες και αντιπροσωπεύουν δομημένα μεταδεδομένα που αναφέρονται σε φυσικά ή ψηφιακά αντικείμενα. Τα NFTs λειτουργούν ως ξεχωριστά αναγνωριστικά και συχνά δεν συνδέονται με τα αντικείμενα αυτά καθεαυτά.

Οι υποστηρικτές τους ισχυρίζονται ότι προωθούν τη διαλειτουργική εμπορευματοποίηση των ψηφιακών ή φυσικών αγαθών. Ήδη το 2012-2013, οι κατακερματισμοί αρχείων ή άλλων μορφών δεδομένων είχαν ενσωματωθεί στο κρυπτονόμισμα Bitcoin μέσω της τεχνολογίας blockchain για να παρέχουν απόδειξη της ύπαρξης ή της αυθεντικότητας σε ένα συγκεκριμένο χρονικό σημείο [24]. Η εξέλιξη αυτή βασίστηκε στη δημιουργία των λεγόμενων «χρωματιστών νομισμάτων», ένα είδος διακριτικού που προσδιορίζονται μοναδικά με την προσθήκη μεταδεδομένων στις συναλλαγές με Bitcoin και με το Namecoin, ένα ξεχωριστό blockchain που αναπτύσσει διακριτικά για την κατοχύρωση ονομάτων χώρου προκειμένου να εξυπηρετήσει τη δημιουργία ενός εναλλακτικού, αποκεντρωμένου συστήματος ονομάτων τομέα ανώτατου επιπέδου [25]. Ένα περαιτέρω πείραμα ήταν το Counterparty, το οποίο επεκτείνει τις δυνατότητες για πιο γενικής χρήσης εφαρμογές NFT στο Bitcoin και οι πρώτες κάρτες συναλλαγών που βασίζονται σε τεχνολογία blockchain [26].

Οι προδιαγραφές των περισσότερων υπαρχόντων NFT περιλαμβάνονται σε ένα τεχνικό πρότυπο που ονομάζεται ERC-721 (ERC-721 Non-Fungible Token Standard, 2018). Το πρότυπο αυτό περιγράφει τα απαιτούμενα μεταδεδομένα του NFT και τις εκτελέσιμες συναρτήσεις που πρέπει να υποστηρίζει το υποκείμενο έξυπνο συμβόλαιο για να συνεργαστεί με την υπάρχουσα υποδομή όπως ιστότοποι εμπορίας NFT και λοιπά πρωτόκολλα διεπαφών. Το πρότυπο βασίζεται στο Ethereum blockchain, το πιο δημοφιλές ως τώρα, αλλά πολλές άλλες υλοποιήσεις βασίζονται στο γενικότερο πρότυπο του Ethereum. Το ERC-721 βασίζεται στην επονομαζόμενη Ethereum Improvement Proposal (Πρόταση Βελτίωσης Ethereum - EIP) και οριστικοποιήθηκε το 2018, λίγο μετά το Cryptokitties [27], ένα παιχνίδι συλλογής ψηφιακών γατών, το οποίο έγινε για πρώτη φορά δημοφιλές το 2017.

Από το 2018 και μετά, τα NFT και οι δραστηριοποιούμενες στον τομέα αυτό εταιρείες επεκτάθηκαν ακόμη περισσότερο και διαφοροποίησαν την προσφορά των υπηρεσιών τους. Τα NFTs άρχισαν να παρεισφρύουν στην αγορά των καλών τεχνών όσον αφορά την τιμολόγηση, με το έργο του Beeples «First 5000 Days» να πωλείται για 69,3 εκατομμύρια δολλάρια Η.Π.Α. [28]. Λίγο αργότερα, διαφοροποιήθηκαν περαιτέρω με την δημιουργία NFTs από tweets [29], εξώφυλλα εφημερίδων (The Economist, 2021), ακόμη και από άρθρα νομικών επιθεωρήσεων [30]. Τροφοδοτούμενες από επιχειρηματικά κεφάλαια, επενδύσεις σε κρυπτονομίσματα και μια επιθετική διαφημιστική εκστρατεία, τα σημεία αγορών και εμπορίας NFTs, καθώς και οι σχετικές υποδομές επεκτάθηκαν μαζικά. Στο τέλος του 2021, ένας προγραμματιστής δημιούργησε το διακριτικό «Cryptogotchis», τον πιο ακριβό κλώνο Tamagotchi που υπήρξε μέχρι σήμερα (Cryptogotchi Home, 2021).

Ως αποτέλεσμα αυτής της επέκτασης, δημιουργήθηκαν NFTs από μουσικά τραγούδια, φυσικά αντικείμενα, ακαδημαϊκές εργασίες και πολλά άλλα αγαθά. Μερικές φορές αυτά τα NFTs ήταν απλώς πειράματα από ιδρυτές start-up εταιρειών ή επενδυτές που έψαχναν να τοποθετηθούν στην αναδυόμενη αγορά, ωστόσο άλλοι ισχυρίζονται ότι αυτή η διαδικασία της δημιουργίας διακριτικών (tokenization) θα οδηγήσει σε ένα νέο ιδιοκτησιακό σύστημα.

Καθώς η εξέλιξη των NFTs συνεχίζεται, ο κόσμος της τέχνης έχει ήδη προβεί σε συνεργασίες ανάμεσα σε καθιερωμένους θεσμούς στον κόσμο της τέχνης, όπως η Art Basel και εταιρείες τεχνολογίας. Αυτές οι συνεργασίες καθοδηγούνται εν μέρει από την προοπτική κέρδους με τους υποστηρικτές των κρυπτονομισμάτων να υπόσχονται καλύτερες αμοιβές στους καλλιτέχνες, αποδιαμεσολάβηση και ευκολότερη συμμόρφωση με την υφιστάμενη και επερχόμενη νομοθεσία κατά της νομιμοποίησης εσόδων από παράνομες δραστηριότητες.

7.2 Δημιουργία των NFT και σχετικά ζητήματα

Η λεγόμενη κοπή (minting) είναι η πράξη δημιουργίας ενός NFT. Σε αυτή τη διαδικασία, ένας χρήστης δημιουργεί ένα νέο σύνολο δεδομένων NFT με την αποστολή μιας συναλλαγής σε ένα υποκείμενο έξυπνο συμβόλαιο που υποστηρίζει NFTs, όπως περιγράφονται στο ERC-721. Του εκχωρείται μια διεύθυνση συμβολαίου blockchain και ένα tokenId, τα οποία συνδυαστικά συνθέτουν ένα παγκοσμίως μοναδικό αναγνωριστικό. Επιπλέον μετα-δεδομένα μπορούν να προστεθούν (προαιρετικά) σε ύστερο χρόνο. Η επεξεργασία του διακριτικού (token) δεν είναι απαραίτητη για την κοπή και δεν είναι αναγκαίο να αποθηκευτεί στο NFT ούτε ένας κατακερματισμός (hash) του έργου.

Υπάρχουν τρεις κύριοι τύποι NFT, ανάλογα με το πώς σχετίζονται με το ψηφιακό ή φυσικό περιουσιακό αγαθό που αντιπροσωπεύουν. Πρώτον, για ορισμένα NFT, το έργο μεταφορτώνεται στο blockchain. Αυτό, για παράδειγμα, μπορεί να συμβεί μέσω κώδικα δημιουργίας τέχνης ή διανυσματικής τέχνης. Αυτός ο τύπος NFT είναι σχετικά σπάνιος λόγω του υψηλού κόστους αποθήκευσης δεδομένων στο blockchain. Δεύτερον, άλλα NFTs ενσωματώνουν δικαιώματα ιδιοκτησίας, είτε προσδιορίζοντας τα στα μεταδεδομένα του NFT ή μέσω αναφοράς σε εξωτερικούς συμβατικούς όρους τρίτων (όπως στο Mintable). Και στις δύο περιπτώσεις, η ιδιοκτησία μπορεί να μεταβιβαστεί μέσω συναλλαγών blockchain [31]. Τέλος, ο πιο ευρύτερα χρησιμοποιούμενος τύπος NFT δεν παρέχει κανένα είδος δικαιώματος στον ιδιοκτήτη του ή βασίζεται σε μια άδεια τύπου “creative commons”, όπως η CC0, η οποία επίσης δεν παρέχει δικαιώματα στον κάτοχο του διακριτικού, καθώς τα σχετικά δικαιώματα εκχωρούνται ως ελευθέρας χρήσεως.

Τα NFT εγείρουν πολλά ζητήματα, τα πιο σημαντικά εκ των οποίων είναι η αβεβαιότητα σχετικά με το νομικό πλαίσιο και τα οικονομικά οφέλη που παρέχουν, καθώς και τις περιβαλλοντικές επιπτώσεις τους λόγω της υποκείμενης τεχνολογίας blockchain. Η ευκολία δημιουργίας «ψηφιακών εκδόσεων» είτε έργων τέχνης είτε συλλεκτικών αντικειμένων σε ανοιχτό και οικονομικά ρευστό δίκτυο που δημιουργήθηκε για τη μεταφορά υπεραξιών έχει εν μέρει δημιουργήσει νέες πηγές εσόδων για τους καλλιτέχνες, τα μουσεία και τις εταιρείες του χώρου. Ορισμένοι υποστηρικτές των NFTs προβάλλουν το επιχείρημα ότι «τα NFTs μπορεί να είναι σε θέση να εκδημοκρατίσουν την τέχνη» [32], καθώς επιτρέπουν σε ένα ευρύ φάσμα ανθρώπων να διαδώσουν την ψηφιακώς παραχθείσα τους τέχνη και να αμείβονται για αυτή τη διάδοση. Ωστόσο, σε αντίθεση με τους σχετικούς ισχυρισμούς, δεν υπάρχουν προς το

παρόν στοιχεία που να αποδεικνύουν ότι τα NFTs συμβάλλουν στον αγώνα των καλλιτεχνών να εξασφαλίσουν ένα αξιοπρεπές εισόδημα (με ορισμένες αξιοσημείωτες εξαιρέσεις) σε σύγκριση με άλλες μορφές διαδικτυακής αποτίμησης σε χρήμα.

Όσον αφορά στα πνευματικά δικαιώματα, τα NFTs δεν συμβάλλουν για να εξασφαλίσουν τα προς το ζην σε πολλούς καλλιτέχνες, δεδομένου ότι είναι ελεύθερα προσβάσιμα, και ως εκ τούτου ήδη καθιερωμένοι καλλιτέχνες) είναι αυτοί που και ευδοκιμούν στο νέο περιβάλλον. Σε σύγκριση με άλλες δημοφιλείς υπηρεσίες προβολής περιεχομένου, όπως οι υπηρεσίες συνεχούς ροής, τα NFTs δεν προστατεύονται από τη διαχείριση ψηφιακών δικαιωμάτων και επομένως μπορούν να νέμονται από οποιονδήποτε, ή και πολλούς κατόχους ταυτόχρονα. Αυτή η «μη ανταγωνιστική φύση» ευνοεί καλλιτέχνες με υφιστάμενη επιρροή στα καλλιτεχνικά δρώμενα, καθώς και αυτούς που ποντάρουν στη δημιουργία τεχνητών ελλείψεων [33], περιορίζοντας τεχνητά όχι το έργο αυτό καθαυτό, αλλά και τις αναφορές σε αυτό. Ωστόσο, το ίδιο το έργο μπορεί ακόμα να το απολαύσει και να το αντιγράψει οποιοσδήποτε. Με τη μη ύπαρξη δικαιωμάτων ιδιοκτησίας σε αυτά, ένα NFT είναι ουσιαστικά συχνά μόνο ένα μοναδικό καθολικό αναγνωριστικό για μια αναφορά σε κάποιο έργο.

Ως εκ τούτου, αφενός οι υποστηρικτές των NFTs περιγράφουν αυτά τα κομμάτια μεταδεδομένων ως την ανατολή ενός νέου οικονομικού συστήματος και απελευθέρωσης της τέχνης και των καλλιτεχνών από τις καταπιεστικές δυνάμεις της αγοράς της τέχνης, ενώ οι αντίπαλοι τους και οι σκεπτικιστές τα θεωρούν ως δείγμα ανεξέλεγκτου καπιταλισμού, λόγω της άμετρης εμπορευματοποίησης και της τιτλοποίησης της τέχνης. Η εμπορευματοποίηση αναφέρεται στην αντιμετώπιση της τέχνης ως ένα κοινό εμπορεύσιμο αγαθό αντί για που διατηρεί αξία αυτόφωτη και ανεξάρτητη από το χρήμα. Η τιτλοποίηση αναφέρεται στον κίνδυνο να μετατραπούν τα πάντα σε χρηματοπιστωτικό μέσο χρηματοοικονομικής κερδοσκοπίας, η οποία στη συνέχεια επιτρέπει επίσης την κλασματοποίηση (διαίρεση σε μετοχές) ενός περιουσιακού στοιχείου. Καταγγέλλοντας την αχαλίνωτη απάτη και την κερδοσκοπία, οι αντίπαλοι των NFTs ισχυρίζονται ότι τα οικονομικά μοντέλα που χρησιμοποιούνται από διάφορα έργα NFT δεν προσφέρουν μη καπιταλιστικά κίνητρα, όπως π.χ. ένα δικαιότερο οικονομικό σύστημα.

Όσον αφορά στις περιβαλλοντικές ανησυχίες, τα περισσότερα NFTs σήμερα δημιουργούνται σε blockchain τύπου Proof-of-Work, τα οποία απαιτούν τεράστιες ποσότητες ενέργειας για να τροφοδοτήσουν τη λειτουργία και ασφάλειά τους και κατακρίνονται λόγω του περιβαλλοντικού τους αποτυπώματος. Υπάρχουν blockchains που βασίζονται σε Proof-of-Stake συστήματα ή συστήματα δεύτερου επιπέδου, τα οποία είτε είναι υπό ανάπτυξη, είτε είναι ήδη διαθέσιμα για την άμβλυνση των παραπάνω περιβαλλοντικών επιπτώσεων. Ωστόσο, προς το παρόν, η απαιτούμενη ποσότητα ενέργειας είναι ένα ουσιαστικό επιχείρημα κατά της δημιουργίας των NFT.

Άλλα τεχνικά και κοινωνικο-νομικά ζητήματα που εγείρονται από τα NFTs αφορούν στην αποσύνθεση συνδέσμων (φαινόμενο «link rot»), καθώς και στις κατηγορίες για απάτη και ξέπλυμα βρώμικου χρήματος. Πολλά NFTs διαθέτουν μόνο έναν σύνδεσμο για το περιεχόμενο τους, επομένως η "αποσύνθεση συνδέσμων" είναι μια βάσιμη ανησυχία. Ο όρος αυτός περιγράφει την κατάσταση όπου ο υπερσύνδεσμος δεν οδηγεί πλέον στη διεύθυνση που περιέχει, γιατί δεν είναι πλέον διαθέσιμη μέσω της αντίστοιχης υπηρεσίας φιλοξενίας διακομιστών. Τέλος, υπάρχουν και αρκετές καταγγελίες για ξέπλυμα βρώμικου χρήματος. Συγκεκριμένα έργα, π.χ. ειδικά εκείνα που αφορούν σε συλλεκτικά αντικείμενα, μερικές

φορές εξαφανίζονται αμέσως μετά την πώληση όλων των NFTs που εκπορεύονται από αυτά [34] και τέτοια φαινόμενα δημιουργούν εύλογη ανησυχία.

7.3 NFT & Πνευματική ιδιοκτησία

Τα NFTs και η νομοθεσία περί πνευματικών δικαιωμάτων έχουν δύο σημαντικά σημεία αλληλεπίδρασης. Το πρώτο είναι επικεντρώνεται στην «κοπή» - δημιουργία των NFTs, και η δεύτερη επικεντρώνεται στη διασπορά των ψηφιοποιημένων έργων.

Αναμφίβολα, το περιεχόμενο πίσω από κάθε NFT μπορεί να υπόκειται σε προστασία πνευματικών δικαιωμάτων. Το όριο της πρωτοτυπίας (δηλαδή αν ένα έργο είναι αρκετά πρωτότυπο ώστε να προστατευτεί από τη σχετική νομοθεσία) αποτελεί προαπαιτούμενο απονομής της σχετικής προστασίας βάσει της νομοθεσίας περί πνευματικών δικαιωμάτων της Ευρωπαϊκής Ένωσης, και αυτό το όριο είναι χαμηλό σύμφωνα με τη νομολογία του Δικαστηρίου της Ευρωπαϊκής Ένωσης. Ως εκ τούτου, ακόμη και η τέχνη που βασίζεται σε εικονοστοιχεία - pixels (π.χ. CryptoPunks) μπορεί να πληροί αυτές τις απαιτήσεις. Ομοίως, πολλές άλλες παραδοσιακές έννοιες πνευματικών δικαιωμάτων εξακολουθούν να ισχύουν για ψηφιακά έργα τέχνης με διακριτικά NFTs.

Η χρήση των διακριτικών που αναφέρονται σε ένα έργο που προστατεύεται από πνευματικά δικαιώματα οδηγεί σε πιο ουσιαστικές προκλήσεις στον τομέα των πνευματικών δικαιωμάτων. Καταρχάς, η ανάρτηση μιας ψηφιακής εικόνας σε έναν ιστότοπο (π.χ. OpenSea) μπορεί παραβιάζει το οικονομικό δικαίωμα διάθεσης του έργου στο κοινό από τον δημιουργό. (Στην Ευρωπαϊκή Ένωση, το άρθρο 3 της Οδηγίας InfoSoc παρέχει αυτό το δικαίωμα στους δημιουργούς και στους κατόχους συγγενικών δικαιωμάτων όσον αφορά στη χρήση κατ' απαίτηση). Δεύτερον, δεν είναι καθόλου βέβαιο ότι η προσφορά προς «πώληση» του ίδιου του NFT αποτελεί «χρήση» αυτού υπό την παραδοσιακή έννοια των πνευματικών δικαιωμάτων. Είναι εύλογο ότι η μεταφορά των NFT δεν εμπίπτει στο πεδίο του δικαιώματος διανομής, καθώς η διανομή σχετίζεται κυρίως με τη μεταβίβαση κυριότητας απτών αντιγράφων των έργων. Αντιθέτως, η προσφορά πρόσβασης σε ψηφιακά αντίγραφα αντιμετωπίζεται ως διάθεση αυτού του αντιγράφου στο κοινό. Η απόφαση του ΔΕΕ επιβεβαίωσε τα παραπάνω στην υπόθεση Tom Kabinet [35]. Η ίδια απόφαση κατέληξε στο συμπέρασμα ότι το δόγμα της εξάντλησης θα πρέπει να παραμείνει ανεφάρμοστο στον ψηφιακό τομέα για έργα πλην λογισμικού.

Μια σταθερό σημείο τριβής μεταξύ των NFTs και της νομοθεσίας περί πνευματικών δικαιωμάτων αποτελεί η παραπλανητική χρήση ορολογίας που σχετίζεται με τα πνευματικά δικαιώματα. Η χρήση της ορολογίας πνευματικών δικαιωμάτων δημιουργεί την ψευδαίσθηση ότι τα NFTs αβίαστα ενσωματώνουν δικαιώματα ιδιοκτησίας. Επιπλέον, οι αξιώσεις περί αυθεντικότητας εγείρονται με βάση συνδέσμους προς ένα έργο, ακόμη και όταν δεν υπάρχει νομική σύνδεση μεταξύ του έργου και του εν λόγω NFT. Η απόκτηση δικαιωμάτων κυριότητας σπάνια συνδέεται με την απόκτηση ενός NFT και οι πλατφόρμες συχνά δεν καταβάλλουν καμμία προσπάθεια για να επαληθεύσουν την αυθεντικότητα του. Υπάρχει ακόμη και ένα έργο που επιτρέπει την αυτοματοποιημένη «κλωνοποίηση» ενός NFT, μέσω της δημιουργίας του από οποιονδήποτε χρήστη (Knockoff NFTs, 2021).

Στα NFTs, οι πωλητές μπορούν να θέσουν τους δικούς τους σχετικούς όρους. Αυτοί οι όροι μπορεί να συνίστανται σε παραδοσιακές μεταβιβάσεις δικαιωμάτων, δυνατότητα χρήσης του

NFT για ξεκλείδωμα πρόσθετου περιεχομένου ή ψηφιακά δικαιώματα μεταπώλησης. Αυτά τα δικαιώματα μπορούν να εκχωρηθούν είτε μέσω παραδοσιακής συμφωνίας αδειοδότησης ή μέσω της επισύναψης πρόσθετων όρων στο NFT. Σε κάθε περίπτωση, οι δημιουργοί και ιδιοκτήτες των NFTs δύνανται να ελέγχουν την τύχη των δημιουργιών τους.

Έχουν γίνει απόπειρες ακόμη και πριν από την αύξηση της δημοφιλίας των NFT για τη χρήση συστημάτων blockchain ώστε να δημιουργηθεί ένα σύστημα αρχείου εγγραφής για έργα που προστατεύονται από πνευματικά δικαιώματα—αλλά όλες απέτυχαν [36]. Μερικές ήταν πολύ πρώιμες ώστε να αποδώσουν καρπούς, άλλες ήταν απλώς πειραματικές (Ujo) και οι υφιστάμενοι ενδιαφερόμενοι φορείς, όπως οι εταιρείες συλλογικής διαχείρισης και εκδότες πιθανότατα έχουν ελάχιστα οφέλη με το να καταστήσουν τη διαδικασία της αδειοδότησης διαφανέστερη. Ωστόσο, για τους καταναλωτές και τους μικρότερης εμβέλειας καλλιτέχνες, η διαφάνεια σχετικά με το ποιος κερδίζει πόσα θα μπορούσε να είναι πολύ ωφέλιμη.

Τέλος, ένα ζήτημα δημόσιας πολιτικής είναι η κοπή-δημιουργία έργων ελευθέρας χρήσεως. Αυτού του είδους η δημιουργία NFT είναι εφικτό να μην απαγορευθεί, καθώς το πρωτότυπο έργο δεν είναι απαραίτητο για την κοπή του διακριτικού. Ωστόσο, υπάρχει περίπτωση να προκληθεί έντονη αντίδραση σε μέρος της κοινωνίας εφόσον θα αποκομίζονται από κάποιους κέρδη κατ’αυτό τον τρόπο από δημόσια ή δωρεάν έργα [37].

Κεφάλαιο 3: Αλγόριθμοι κοινής συναίνεσης

Αλγόριθμος Συναίνεσης

Αλγόριθμος συναίνεσης καλείται ο μηχανισμός μέσω του οποίου ένα δίκτυο blockchain συναινεί. Τα δημόσια blockchains δημιουργούνται ως κατανεμημένα συστήματα και δεν βασίζονται σε κάποια κεντρική αρχή, οι κατανεμημένοι κόμβοι πρέπει να συμφωνήσουν πάνω στο θέμα της εγκυρότητα των συναλλαγών. Έτσι ο αλγόριθμος συναίνεσης πιστοποιεί ότι τηρούνται οι κανόνες πρωτοκόλλου και εγγυώνται ότι όλες οι συναλλαγές πραγματοποιούνται με έναν βέβαιο και ασφαλή τρόπο. Για τον λόγο αυτό κιόλας το κάθε κρυπτονόμισμα μπορεί να χρησιμοποιηθεί μόνον μία φορά.

Πολλές φορές συγχέεται ο όρος «αλγόριθμος» με τον όρο «πρωτόκολλο». Είναι δύο ξεχωριστοί όροι που είναι αλληλένδετοι, εξού και η ταύτιση τους. Με τον όρο «πρωτόκολλο» εννοούμε τον ορισμό των κανόνων σε ένα δίκτυο blockchain, ενώ με τον όρο «αλγόριθμος» εννοούμε τον τρόπο με τον οποίο θα ακολουθηθούν – τηρηθούν οι κανόνες αυτοί. [38]

3.1 Proof of Work (PoW)

Το Proof of Work έκανε την εμφάνισή του στις αρχές του 1990 ως μέσο για τον περιορισμό των ανεπιθύμητων emails, βέβαια ο δημιουργός του Bitcoin εφάρμοσε το μοντέλο αυτό στο white paper του Bitcoin.
Το PoW αναζητά λύση σε σύνθετα μαθηματικά προβλήματα και στη συνέχεια τα επαληθεύει. Αυτό μας επιτρέπει να αναζητούμε τη λύση ενός προβλήματος με τη χρήση υπολογιστικών συστημάτων και όταν βρεθεί η λύση αυτή, τότε γίνεται επαλήθευση και έτσι μεγαλώνει η κρυπτοαλυσίδα. [39]

3.2 Proof of Stake (PoS)

Το Proof of Stake είναι μια παραλλαγή του PoW που εμφανίστηκε το 2012 για να μειωθεί η κατανάλωση ενέργειας. Στο PoS η συμμετοχή καθορίζεται από αυτούς που ελέγχουν μέρος του supply ενός νομίσματος. Ο αλγόριθμος αυτός επιλέγει τυχαία έναν κόμβο, ο οποίος θα πρέπει να προτείνει το επόμενο block στο blockchain. Όταν ένας κόμβος επιλέγεται, τότε θα πρέπει να επαληθευθεί η εγκυρότητα των συναλλαγών εντός του block, να υπογράφει και έπειτα να προτείνει το block στο δίκτυο για επικύρωση. [39]

3.3 Proof of Burn (PoB)

Το Proof of Burn είναι επίσης μια παραλλαγή του PoW για να μειωθεί η κατανάλωση ενέργειας. Ο αλγόριθμος PoB για να επικυρώσει το block, δεν απαιτεί τη χρήση ισχυρών υπολογιστικών συστημάτων, αλλά τα κρυπτονομίσματα καίγονται σκόπιμα ως τρόπος «επένδυσης» πόρων στο δίκτυο blockchain. Με αυτόν τον τρόπο οι χρήστες αποδεικνύουν τη

δέσμευσή τους στο δίκτυο και αποκτούν το δικαίωμα στο «mining» και στην επικύρωση συναλλαγών. [40]

3.4 Proof of Capacity (PoC)

Το Proof of Capacity είναι ένας μηχανισμός για να επιλεγεί κάποιος από τους συμμετέχοντες του δικτύου και να δημιουργήσει ένα νέο block για το blockchain. Το PoC θεωρείται ιδιαίτερα αποδοτικό ως προς την ενέργεια που καταναλώνει και ως προς τους πόρους που δεσμεύει, επειδή οι συμμετέχοντες παρέχουν προσωρινά χώρο αποθήκευσης στους σκληρούς τους δίσκους.

3.5 Proof Of Lapsed Time (PoET)

Το Proof of Lapsed Time ακολουθεί ένα σύστημα λοταρίας κατά κάποιον τρόπο. Ο αλγόριθμος αυτός χρησιμοποιεί έναν τυχαία παραγόμενο χρόνο που έχει παρέλθει, για να αποφασίσει τα δικαιώματα της εξόρυξης. Με την εκτέλεση ενός αξιόπιστου κώδικα σε ένα ασφαλές περιβάλλον, ο PoET εξασφαλίζει ότι τα αποτελέσματα που θα παραχθούν είναι επαληθεύσιμα από τους εξωτερικούς συμμετέχοντες. [41]

3.6 Byzantine Fault Tolerance (BFT)

Το Byzantine Fault Tolerance επινοήθηκε το 1982 ως ένα λογικό δίλημμα που δείχνει πώς μια ομάδα «Βυζαντινών Στρατηγών» μπορεί να έχει προβλήματα επικοινωνίας, όταν προσπαθεί να συμφωνήσει για την επόμενη κίνησή της. Κάθε στρατηγός έχει τον δικό του στρατό και κάθε ομάδα βρίσκεται σε διαφορετικές τοποθεσίες γύρω από την πόλη που σκοπεύουν να επιτεθούν. Οι στρατηγοί πρέπει να συμφωνήσουν ή για επίθεση ή για υποχώρηση. Και τα δύο δε μπορεί να γίνουν! [42]

3.7 Proof of Authority (PoA)

Στον αλγόριθμο Proof of Authority οι συναλλαγές και τα blocks επικυρώνονται από εγκεκριμένους επικυρωτές. Οι επικυρωτές «τρέχουν» ένα λογισμικό που τους επιτρέπει να τοποθετούν συναλλαγές σε blocks. Η διαδικασία αυτή είναι αυτοματοποιημένη και δεν είναι απαραίτητο οι επικυρωτές να παρακολουθούν συνεχώς τις οθόνες τους. Με τον αλγόριθμο PoA, τα άτομα κερδίζουν το δικαίωμα να γίνουν επικυρωτές, επομένως υπάρχει ένα *κίνητρο* να διατηρήσουν τη θέση τους.

3.8 Proof of Authentification (PoAh)

Το Proof of Authentification μπορεί να αναβαθμίσει τη διαχείριση του νομίσματος. Με την εγγραφή ενός συμβάντος σε ένα blockchain δίκτυο, αποδεικνύεται αυτόματα η γνησιότητά του. Αυτό συμβαίνει επειδή κάθε κρυπτονόμισμα έχει μια μοναδική διεύθυνση και γίνεται

κατάτμηση σε ένα δημόσιο chain. Βέβαια εκτελείται σε λίγα υπολογιστικά συστήματα και έχει καθυστέρηση τρία περίπου δευτερόλεπτα. [43]

3.9 Proof of Possession (PoP)

Το Proof of Possession χρησιμοποιείται για τον έλεγχο ταυτότητας και για να μετριάσουμε τον κίνδυνο κλοπής και χρήσης των διακριτικών ασφαλείας από έναν εισβολέα. Σε αυτή τη φάση γίνεται χρήση ενός κλειδιού κρυπτογραφίας. [44]

3.10 Proof of Importance (PoI)

Το Proof of Importance (PoI) είναι ένα σύστημα που χρησιμοποιείται για να καθορίσει ποιοι χρήστες είναι κατάλληλοι να εκτελέσουν τους απαραίτητους υπολογισμούς, για να προσθέσουν ένα νέο block δεδομένων σε μια κρυπτοαλυσίδα και τέλος να λάβουν τη σχετική πληρωμή. [45]

Κεφάλαιο 4: Κόμβοι

4.1. Κόμβος

Τον όρο «Κόμβος» ή «Node» τον συναντάμε κυρίως στα δίκτυα υπολογιστών και στα κατανεμημένα συστήματα. Στον χώρο των ψηφιακών νομισμάτων, ως nodes ορίζονται οι συσκευές ενός blockchain δικτύου και η κύρια λειτουργία τους είναι να επαληθεύουν την εγκυρότητα κάθε επόμενης δοσοληψίας, που ονομάζονται blocks. Όπως είναι αντιληπτό, κάθε κόμβος έχει και τη δική του διεύθυνση. Υπάρχουν αρκετών ειδών κόμβοι. Παρακάτω γίνεται μια σύντομη αναφορά στους Full nodes, Online and Offline nodes, Light nodes.

4.2. Full Nodes

Αυτός ο τύπος κόμβου διατηρεί όλες τις συναλλαγές του δικτύου και είναι υπεύθυνος για την επικύρωση των blocks, καθώς και για τις συναλλαγές. Είναι απαραίτητος για το δίκτυο blockchain, διότι παρέχει ασφάλεια. [46]

4.3. Online & Offline Nodes

Ο τύπος αυτός είναι πάντα online επειδή στέλνει συνεχώς ενημερώσεις στο δίκτυο, ενώ οι offline κόμβοι κάθε φορά που συνδέονται ξανά σε ένα δίκτυο λαμβάνουν ένα ενημερωμένο αντίγραφο του ledger (ψηφιακό πορτοφόλι). Και αυτό διότι πρέπει να διασφαλισθεί ο συγχρονισμός των offline nodes με το δίκτυο.

4.4. Light Nodes

Ο κόμβος τύπου Light Node χρειάζεται μόνον για τη λήψη των κεφαλίδων από τα blocks, ώστε να γίνει η επαλήθευση. Επίσης, χρησιμοποείται ως σημείο επικοινωνίας με το δίκτυο. Όπως καταλαβαίνουμε όλοι οι κόμβοι είναι απαραίτητοι για την ορθή λειτουργία του blockchain δικτύου, αφού εξασφαλίζουν την αξιοπιστία των δεδομένων και την ειλικρίνεια των ατόμων που συμμετέχουν σε ένα τέτοιο δίκτυο. [47]

Κεφάλαιο 5: Δημοφιλή Blockchain

5.1. Bitcoin

Το Bitcoin ή αλλιώς κρυπτονόμισμα είναι ψηφιακό νόμισμα, το οποίο δεν απαιτεί την παρέμβαση κάποιας τράπεζας για την έκδοσή του. Το ψηφιακό αυτό νόμισμα εφευρέθηκε το 2008 από την ομάδα Satoshi Nakamoto και η διαθεσή του ξεκίνησε το 2009, όταν ο πηγαίος κώδικάς αναρτήθηκε στο internet ως ελεύθερο λογισμικό.

Πίσω από αυτή την τεχνολογία κρύβεται ένα ολόκληρο σύστημα ασφάλειας! Η πιστότητα και η φερεγγυότητα των συναλλαγών διασφαλίζονται με τεχνικές που βασίζονται στην κρυπτογράφηση και καταχωρούνται σε ένα δημόσιο κατανεμημένο αρχείο που ονομάζεται blockchain. Τα bitcoins είναι επακόλουθο μιας υπολογιστικής διεργασίας που μπορεί να επιτευχθεί από οποιοδήποτε ισχυρό υπολογιστικό σύστημα. Η διεργασία συντίθεται από μια σειρά πολύπλοκων μαθηματικών υπολογισμών που καλείται εξόρυξη, στα αγγλικά mining.

Τα bitcoins εξαργυρώνονται μέσω ειδικών συναλλαγματικών πρακτορείων και χρησιμοποιούνται για να αγορά προϊόντων ή/και υπηρεσιών, αλλά μπορούν να χρησιμοποιηθούν και σαν επένδυση.

Τα κρυπτονομίσματα έχουν κατακριθεί για την χρήση τους σε παράνομες ενέργειες, για το γεγονός ότι καταναλώνουν αρκετή ηλεκτρική ενέργεια για τη την εξόρυξή τους, για τη μεταβλητότητα της τιμής τους και για τις υψηλές χρεώσεις για την μετατροπή τους σε άλλα νομίσματα. Αρκετοί οικονομολόγοι έχουν χαρακτηρίσει τοα bitcoins ως χρηματοπιστωτικό σύννεφο! Με άλλα λόγια για κάποιους είναι μόδα των καιρών!

Όπως καθετί έτσι και τα ψηφιακά νομίσματα έχουν τα θετικά και τα αρνητικά τους. Τα θετικά της υπόθεσης είναι: η γρήγορη ταχύτητα με την οποία γίνονται οι συναλλαγές, το γεγονός ότι δεν υπάρχει γεωγραφικός περιορισμός, το χαμηλό κόστος συναλλαγών και οι διαφάνεια στις συναλλαγές. Στα αρνητικά σημειώνονται: η απώλεια ιδιωτικών κλειδιών, το ασαφές νομικό πλαίσιο και τα τυχόν κενά στην ασφάλεια. [48]

5.2. Etherium

Το Ethereum είναι μια δημόσια πλατφόρμα blockchain ανοιχτού κώδικα που είναι βασισμένη στον παράλληλο προγραμματισμό, διαθέτοντας τη λειτουργικότητα έξυπνης σύμβασης, στα αγγλικά scripting.

Το Ether είναι κρυπτονόμισμα του οποίου το δίκτυο blockchain γεννιέται από τη πλατφόρμα Ethereum. Το Ether μπορεί να μεταφέρεται μεταξύ λογαριασμών και να χρησιμοποιηθεί για την αντιστάθμιση συμμετεχόντων κόμβων εξόρυξης για τους εκτελούμενους υπολογισμούς. Το Ethereum παρέχει μια αποκεντρωμένη Εικονική Μηχανή , η οποία μπορεί να εκτελέσει σενάρια χρησιμοποιώντας το διεθνές δίκτυο δημόσιων κόμβων. Ο Gas, μια λειτουργία τιμολόγησης εσωτερικών συναλλαγών, χρησιμοποιείται για να ελαττώσει τα spam και να μοιράσει τους πόρους του δικτύου.

Το 2016, το Ethereum χωρίστηκε σε δύο ξεχωριστά blockchains. Το Ethereum και το Ethereum Classic.

5.3. Διαφορές Bitcoin – Ethereum

Στο πρώτο κομμάτι αναλύσαμε τα bitcoins και έπειτα τα Ethereum νομίσματα. Αξίζει να σημειωθεί ότι πρόκειται για δύο από τα κορυφαία ψηφιακά νομίσματα και αρκετοί είναι εκείνοι που μπαίνουν στη διαδικασία να τα συγκρίνουν.

Από τη μια πλευρά το bitcoin χρησιμοποιείται ως εναλλακτικό νόμισμα και ως αποθήκη αξίας, επειδή υπάρχει περιορισμένη προσφορά. Εκτιμάται ότι θα υπάρξουν συνολικά 21.000 bitcoins. Θέλει αρκετό χρόνο εξόρυξης με αποτέλεσμα να είναι ασύμφορο ως προς την κατανάλωση ενέργειας. Επίσης, θέλει αρκετό χρόνο ώστε να απελευθερώσει κάποια blocks και αυτό γίνεται διότι δίνεται έμφαση στα θέματα ασφαλείας.

Από την άλλη πλευρά το ethereum, αναπτύχθηκε για ως πλατφόρμα κατασκευής εφαρμογών με νομίσματα και δεν υπάρχει περιορισμός στη προσφορά του. Δε σπαταλάται σημαντικός χρόνος στην εξόρυξη, γεγονός που δεν καταναλώνεται τόση ενέργεια. Επιπλέον, δε χρειάζεται τόσο χρόνο για να απελευθερώσει blocks. Επιπρόσθετα, θα πρέπει να αναφέρουμε ότι το ethereum είναι ακόμα νέο νόμισμα, καθώς μόλις το 2021 έγινε διαθέσιμη η δεύτερη έκδοση.

Τέλος, δεν υπάρχει καλύτερο ή χειρότερο ψηφιακό νόμισμα. Είναι μια ανούσια σύγκριση καθώς το καθένα αναπτύχθηκε για άλλο λόγο!

Κεφάλαιο 6: Κρυπτογραφία

6.1. Κρυπτογραφία και κρυπτογράφηση

Η κρυπτογραφία ασχολείται με την ασφαλή επικοινωνία. Η κρυπτογράφηση είναι η διαδικασία με την οποία γίνεται κωδικοποίηση μιας επικοινωνίας για να είναι ασφαλής, μέσω διαφόρων μηχανισμών ή ακόμα πιο σωστά μέσω αλγορίθμων κρυπτογράφησης. Οι αλγόριθμοι που έχουν να κάνουν με αναγραμματισμό μηνυμάτων ονομάζονται κλασσικοί αλγόριθμοι κρυπτογραφίας, ενώ οι αλγόριθμοι που έχουν να κάνουν με κλειδιά ονομάζομαι σύγχρονοι αλγόριθμοι κρυπτογράφησης. Παρακάτω γίνεται σύντομη αναφορά σε δύο απλούς αλγορίθμους κρυπτογράφησης.

6.2. Αλγόριθμος Καίσαρα

Ο αλγόριθμος αυτός ολισθαίνει ένα ένα τα γράμματα κάθε λέξης. Βέβαια αυτό γίνεται με τη σύμφωνη γνώμη των δύο άκρων που επικοινωνούν. Δηλαδή, τα δύο άκρα θα πρέπει να καταλήξουν στο πόσες θέσεις θα ολισθήσει κάθε γράμμα. Για παράδειγμα έχουμε τη λέξη ΚΑΛΗΜΕΡΑ. Τα δύο άκρα συμφωνούν ότι οι χαρακτήρες ολίσθησης θα είναι ο αριθμός 5. Άρα ως κρυπτογραφημένο κείμενο έχουμε την λέξη: ΟΖΠΜΠΚΧΖ .

6.3. Ασυμμετρική Κρυπτογράφηση

Ο αλγόριθμος αυτός βασίζεται στη χρήση ενός ιδιωτικού κλειδιού και ενός δημόσιο κλειδιού. Έτσι εξασφαλίζεται η εμπιστευτικότητα της μεταδιδόμενης πληροφορίας. Για παράδειγμα, αν δύο άκρα (Χ, Υ) θέλουν να ανταλλάξουν δεδομένα ο Χ θα πρέπει να κρυπτογραφήσει τα δεδομένα με το δημόσιο κλειδιού του Υ. Ο Υ με τη σειρά του θα αποκρυπτογραφήσει τη πληροφορία με το ιδιωτικό του κλειδί.

Κεφάλαιο 7: Βασικά είδη κρυπτονομισμάτων

Εισαγωγή

Στο blockchain του Ethereum, τα tokens αναγνωρίζονται ως ψηφιακά αντίγραφα. Αυτά τα ψηφιακά αντίγραφα έχουν τη δυνατότητα να αντιπροσωπεύουν τόσο πραγματικά αγαθά, που χρησιμοποιούνται ως μέσο ανταλλαγής για την αγορά ειδικών αγαθών ή υπηρεσιών, όσο και διάφορες άλλες ανταλλάξιμες αξίες. Αυτά τα tokens ουσιαστικά λειτουργούν ως μια μονάδα ανταλλαγής εντός του συστήματος για την αγορά και πώληση αγαθών ή υπηρεσιών. Τα tokens αυτά συνήθως κυκλοφορούν μέσω αρχικών δημόσιων προσφορών (Initial Coin Offerings-ICO). Το Ethereum το 2015 καθόρισε τις πρώτες τεχνικές προδιαγραφές για αυτά τα tokens, που ανταλλάσσονται στο δίκτυο. Αυτό περιλαμβάνει την τυποποίηση των smart contracts, που περιέχουν κοινές λειτουργίες και εφαρμογές, όπως transfer(address _to, uint256 _value), balanceOf(address _owner). Η κατοχή ενός token είναι ισοδύναμη με την πραγματική κατοχή ενός συγκεκριμένου αντικειμένου.

Η διαδικασία μεταφοράς και κατοχής ενός token κάτι που μπορείς να βρεις στο blockchain και μπορεί να αντιπροσωπεύει διάφορες αξίες όπως μετοχές εταιρειών, κουπόνια ή ακόμα και συστήματα επιβράβευσης με πόντους. Η βασική λογική πίσω από το σύστημα αυτό είναι η αφαίρεση των μονάδων X από τον A και η μεταφορά τους στον B, υπό τις προϋποθέσεις ότι (1) ο A διέθετε τουλάχιστον μονάδες X πριν τη συναλλαγή και (2) η συναλλαγή έχει την έγκριση του A.

Τα EIPs (Ethereum Improvement Proposals) και τα ERCs (Ethereum Request for Comments) είναι πρότυπα και προτάσεις στο οικοσύστημα του Ethereum. Περιγράφουν τεχνικές προδιαγραφές, πρότυπα και κατευθυντήριες γραμμές για το Ethereum, τα smart contracts και τα συναφή στοιχεία, πριν απο την υιοθετηση τους τίθενται ως προς ψηφιση και η κοινότητα αποφασίζει αν θα ενταχθούν στο πρωτοκολλο ή οχι, επίσης ο κάθε ένας μπορεί να δημιουργήσει μια πρόταση ως προς ψήφιση.

Τα πρότυπα token στο Ethereum blockchain είναι ένα σύνολο προκαθορισμένων κανόνων και προδιαγραφών που διέπουν τη δημιουργία, τη συμπεριφορά και τη λειτουργικότητα των ψηφιακών token. Αυτά τα πρότυπα παρέχουν ένα κοινό πλαίσιο για τους προγραμματιστές για τη δημιουργία και την αλληλεπίδραση με τα tokens, επιτρέποντας την απρόσκοπτη συμβατότητα και διαλειτουργικότητα μεταξύ διαφόρων αποκεντρωμένων εφαρμογών (DApps) και έξυπνων συμβολαίων στο Ethereum. Τα πιο ευρέως υιοθετημένα πρότυπα token στο Ethereum περιλαμβάνουν τα ERC-20, ERC-721 και ERC-1155 και παρουσιάζονται στη συνέχεια. [50] [51]

7.1 ERC-20

Το ERC-20, ένα κομβικό πρότυπο στον Ethereum, ανήκει στην κατηγορία fungible tokens που μοιάζουν αρκετά σαν τα νομίσματα και μπορούν να υποδιαιρεθούν σε μικρότερα. Αυτό το πρότυπο επιτρέπει την ύπαρξη ενός πλαισίου για τους προγραμματιστές, προκειμένου να δημιουργούν και να διαχειρίζονται αυτά τα tokens.

Τα tokens βασισμένα στο πρότυπο ERC-20, είναι πανομοιότυπα και ανταλλάξιμα, παρέχοντας την ιδανική βάση για την ανάπτυξη ενός ευρέως φάσματος εφαρμογών. Αυτά περιλαμβάνουν αποκεντρωμένες εφαρμογές (DApps), αρχικές προσφορές, και άλλες εφαρμογές εντός της πλατφόρμας Ethereum. Οι βασικές λειτουργίες που περιλαμβάνονται σε ένα ERC-20 token περιλαμβάνουν την transfer (για μεταφορές token), την balanceOf (για την

εύρεση του υπολοίπου token ενός λογαριασμού) και την approve (που επιτρέπει σε έναν άλλο λογαριασμό να χρησιμοποιήσει ένα συγκεκριμένο ποσό token). Αυτές οι λειτουργίες επιτρέπουν την αλληλεπίδραση μεταξύ των διάφορων διαδραστών και συστημάτων στο δίκτυο Ethereum.

Τα ERC-20 tokens ενσωματώνονται εύκολα στις υπάρχουσες υποδομές του Ethereum, όπως τα πορτοφόλια, τα ανταλλακτήρια και άλλες DApps, επιτρέποντας την άμεση διαπραγμάτευση σε αποκεντρωμένα χρηματιστήρια και τη χρήση σε πρωτόκολλα αποκεντρωμένης χρηματοδότησης (DeFi). Αυτή η τυποποιημένη φύση διευκολύνει την ανάπτυξη έξυπνων συμβολαίων και DApps, που μπορούν να αλληλεπιδρούν με πολλαπλά tokens, βασισμένα σε συνεπείς διεπαφές και συμπεριφορές.

Ωστόσο, είναι κρίσιμο να τονιστεί ότι, παρά την ευρεία υιοθέτηση του ERC-20, το πρότυπο αυτό δεν προσφέρει από μόνο του εγγυήσεις για ασφάλεια ή νομιμότητα των tokens. Από τεχνικής απόψεως, τα ERC-20 tokens απαιτούν ακριβή προγραμματισμό και προσοχή στις λεπτομέρειες για να διασφαλιστεί η συμβατότητα και η ασφάλεια. Οι προγραμματιστές πρέπει να είναι ενήμεροι για τα ζητήματα ασφαλείας και τις πρακτικές καλής ανάπτυξης για να δημιουργήσουν ασφαλή και αξιόπιστα tokens γιατί έχουν υπάρξει πολλά ζητήματα.

Στην πράξη, έχουν αναπτυχθεί πολλές εφαρμογές που χρησιμοποιούν τα ERC-20 tokens. Παραδείγματα περιλαμβάνουν την πλατφόρμα MakerDAO, η οποία χρησιμοποιεί το ERC-20 token DAI ως stablecoin, την πλατφόρμα Compound για αποκεντρωμένο δανεισμό και δανειστική στην αγορά DeFi, και το Uniswap, μια αποκεντρωμένη πλατφόρμα ανταλλαγής που επιτρέπει στους χρήστες να ανταλλάσσουν διάφορα ERC-20 tokens. Αυτές οι εφαρμογές αντιπροσωπεύουν την ποικιλία και τις δυνατότητες που προσφέρουν τα ERC-20 tokens στο οικοσύστημα του Ethereum. [52]

7.2 ERC-721

Το πρότυπο ERC-721ανήκει στην κατηγορία των non fungible tokens (NFTs). Αυτό το πρότυπο διαφέρει ριζικά από το ERC-20, καθώς κάθε ERC-721 token αποτελεί ένα μοναδικό ψηφιακό αντικείμενο (δεν μπορεί να χωριστεί σε μικρότερα), με τη δυνατότητα να έχει διαφορετική αξία και ιδιότητες από οποιοδήποτε άλλο token.

Το πρότυπο ERC-721 δίνει ένα πρότυπο για τη δημιουργία και διαχείριση αυτών των μοναδικών tokens στο Ethereum blockchain. Κάθε token έχει ένα μοναδικό αναγνωριστικό (token ID) και συνδέεται με μια συγκεκριμένη διεύθυνση, επιτρέποντας την αναπαράσταση ενός μοναδικού περιουσιακού στοιχείου, όπως μια ψηφιακή εικόνα, ένα κομμάτι μουσικής ή άλλου είδους ψηφιακό περιεχόμενο.

Το ERC-721 προσδιορίζει ένα σύνολο από λειτουργίες για την αλληλεπίδραση με αυτά τα tokens. Στις κύριες λειτουργίες περιλαμβάνονται η transferFrom(), για τη μεταφορά των tokens, η approve() για την εξουσιοδότηση κάποιου άλλου να χειριστεί το token στο πλαίσιο του ιδιοκτήτη, και η ownerOf() για τον εντοπισμό του κατόχου ενός συγκεκριμένου token. Επιπλέον, το πρότυπο υποστηρίζει γεγονότα όπως το Transfer και το Approval, που εκπέμπονται σε συγκεκριμένες λειτουργίες.

Αυτή η τεχνολογία έχει ανοίξει τον δρόμο για πληθώρα εφαρμογών που ξεπερνούν τον τομέα της ψηφιακής τέχνης και εκτείνονται σε αγορές όπως τα τυχερά παιχνίδια, τα εικονικά ακίνητα, και άλλες μορφές ψηφιακής ιδιοκτησίας. Τα NFTs προσφέρουν μια ξεχωριστή δυνατότητα για τη δημιουργία και τη διαπραγμάτευση μοναδικών ψηφιακών περιουσιακών στοιχείων, δίνοντας τη δυνατότητα στους δημιουργούς και τους συλλέκτες να αλληλεπιδρούν με τα αγαθά τους σε μια αποκεντρωμένη και διαφανή πλατφόρμα.

Οι εφαρμογές του ERC-721 έχουν χρησιμοποιηθεί στην τέχνη και τις συλλογές, παρέχοντας μια πλατφόρμα για την αναπαράσταση, αγορά, πώληση και ανταλλαγή μοναδικών ψηφιακών περιουσιακών στοιχείων. 'Ηδη υπάρχουν εφαρμογές όπως το CryptoKitties, όπου οι χρήστες συλλέγουν και αναπαράγουν εικονικές γάτες, και το Decentraland, μια εικονική πλατφόρμα όπου οι χρήστες μπορούν να αγοράζουν και να πωλούν εικονικά ακίνητα. Η δημιουργία αυτών των μοναδικών ψηφιακών αντικειμένων και η δυνατότητα αλληλεπίδρασης με αυτά μέσω του blockchain ανοίγει νέους δρόμους στην ψηφιακή ιδιοκτησία και τους συλλέκτες. [53]

7.3 ERC-1155

Το ERC-1155 στο Ethereum προσφέρει μια ευέλικτη λύση για τη δημιουργία και διαχείριση τόσο ανταλλάξιμων όσο και μη-ανταλλάξιμων tokens μέσα σε ένα smart contract. Αυτή η προσέγγιση επιτρέπει την αποδοτική διαχείριση πολλαπλών τύπων assets, μειώνοντας το κόστος για τους προγραμματιστές και βελτιστοποιώντας το gas cost. Το πρότυπο αυτό συνδυάζει τα χαρακτηριστικά των ERC-20 και ERC-721, προσφέροντας μεγαλύτερη ευελιξία στην αναπαράσταση διαφορετικών τύπων token σε ένα μόνο συμβόλαιο.

Το token περιλαμβάνει λειτουργίες όπως η safeTransferFrom(), η safeBatchTransferFrom() και η balanceOf(), οι οποίες διευκολύνουν την ασφαλή μεταφορά και την ανακάλυψη του υπολοίπου των tokens. Με το ERC-1155, ένας κάτοχος μπορεί να μεταφέρει πολλαπλά tokens σε διαφορετικούς παραλήπτες με μία μόνο συναλλαγή, εξοικονομώντας έτσι το gas και βελτιστοποιώντας την αποδοτικότητα του δικτύου.

Στον τομέα των τυχερών παιχνιδιών, τα ERC-1155 tokens έχουν χρησιμοποιηθεί για την αναπαράσταση διαφόρων εικονικών αντικειμένων με διαφορετικές αξίες και ιδιότητες, ενώ στον χώρο των ψηφιακών συλλεκτικών αντικειμένων, διευκολύνουν τη δημιουργία μοναδικών συλλογών με ελάχιστη πολυπλοκότητα. Στο πεδίο των DeFi, το ERC-1155 προσφέρει τη δυνατότητα ανταλλαγής διαφορετικών τύπων χρηματοοικονομικών assets μέσα από ένα ενιαίο συμβόλαιο, επιτρέποντας την αποδοτική διαχείριση και μεταβίβαση περιουσιακών στοιχείων. [54]

Στην εφοδιαστική αλυσίδα, τα ERC-1155 tokens μπορούν να αναπαριστούν διαφορετικά προϊόντα, παρέχοντας έναν αποτελεσματικό τρόπο για την παρακολούθηση και επαλήθευση της προέλευσης και γνησιότητας των αγαθών. Το πρότυπο επιτρέπει την ενσωμάτωση πληροφοριών, όπως προέλευση, πιστοποιήσεις και ποιοτικά χαρακτηριστικά σε κάθε token.

Τέλος, στον τομέα της παραγωγής, παραδείγματα εφαρμογών που χρησιμοποιούν το ERC-1155 περιλαμβάνουν το Enjin, μια πλατφόρμα που επιτρέπει στους χρήστες να δημιουργούν και να διαχειρίζονται τα δικά τους tokens για χρήση σε παιχνίδια και εφαρμογές, και το Sandbox, μια εικονική πλατφόρμα παιχνιδιού όπου τα assets και οι εμπειρίες δημιουργούνται ως ERC-1155 tokens. Αυτές οι εφαρμογές αποδεικνύουν την πολυμορφική δυναμική και την ευελιξία του ERC-1155, ανοίγοντας νέους δρόμους για την ανάπτυξη και την αλληλεπίδραση στον ψηφιακό κόσμο. [55]

7.4 Soulbound tokens

Τα Soulbound tokens (SBTs) αποτελούν μια ενδιαφέρουσα καινοτομία στον τομέα των blockchain tokens. Αυτή η νέα κατηγορία token εστιάζει στην αντιπροσώπευση μη ανταλλάξιμων προσωπικών χαρακτηριστικών ή διαπιστεύσεων σε μια blockchain. Τα Soulbound tokens διαφοροποιούνται από τα συνηθισμένα NFTs λόγω της μη ανταλλάξιμης φύσης τους και της στενής σύνδεσης με την ταυτότητα του κατόχου τους.

Στην ουσία, ένα Soulbound token λειτουργεί ως μια ψηφιακή, αδιάσπαστη και μη μεταβιβάσιμη διαπίστευση που συνδέεται με την ψηφιακή ταυτότητα ενός ατόμου ή οντότητας. Μπορεί να αναπαραστήσει εκπαιδευτικά πιστοποιητικά, επαγγελματικές διαπιστεύσεις, ακόμα και προσωπικά επιτεύγματα ή ιδιότητες. Αυτή η μοναδικότητα και η μη μεταβιβάσιμη φύση τους προσδίδουν μια νέα διάσταση στην ψηφιακή ταυτότητα και την εμπιστοσύνη στον ψηφιακό κόσμο.

Οι εφαρμογές των Soulbound tokens εκτείνονται σε διάφορους τομείς. Στην εκπαίδευση, για παράδειγμα, μπορούν να αναπαραστήσουν πιστοποιητικά και διπλώματα, παρέχοντας μια ασφαλή και αναμφισβήτητη απόδειξη των εκπαιδευτικών επιτευγμάτων ενός ατόμου. Στον επαγγελματικό τομέα, τα SBTs μπορούν να αντιπροσωπεύουν επαγγελματικές διαπιστεύσεις ή εμπειρία, ενισχύοντας την επαγγελματική εξέλιξη και την αγορά εργασίας. Επιπλέον, στον τομέα της υγείας, τα Soulbound tokens έχουν τη δυνατότητα να αποτελέσουν ένα ασφαλές μέσο για την αποθήκευση και τη διαχείριση ιατρικών δεδομένων και ιστορικού.

Στην πραγματική παραγωγή, οι εφαρμογές των Soulbound tokens ακόμα βρίσκονται σε πρώιμο στάδιο ανάπτυξης, αλλά υπάρχουν αρκετά έργα που εξερευνούν τις δυνατότητες τους. Παραδείγματα περιλαμβάνουν τη χρήση τους σε πλατφόρμες blockchain για την αποθήκευση εκπαιδευτικών πιστοποιητικών και την πιστοποίηση επαγγελματικών δεξιοτήτων. Αυτές οι εφαρμογές υπόσχονται να φέρουν μια νέα εποχή στην ψηφιακή αξιοπιστία και τη διαφάνεια, ανοίγοντας νέους δρόμους στην αξιοποίηση των προσωπικών δεδομένων και της ψηφιακής ταυτότητας. [56]

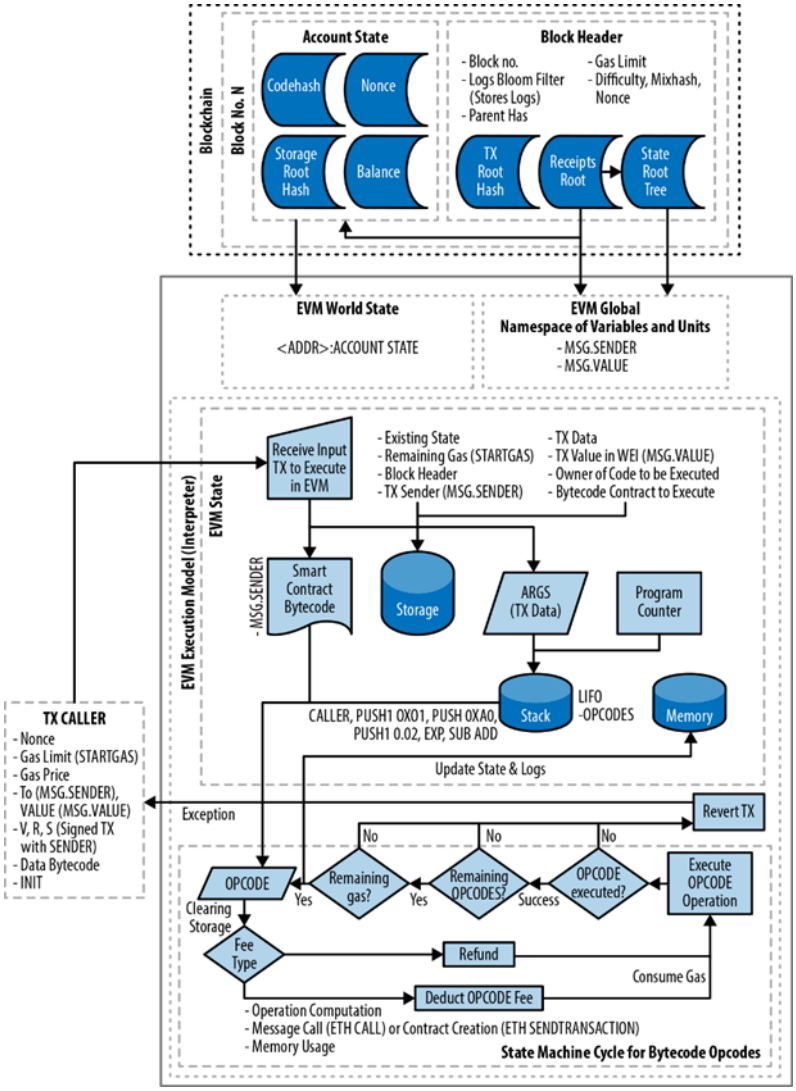
Υλοποιήσεις

Για τα token αυτά υπάρχουν διάφορες υλοποιήσεις όπως της OpenZeppelin και της ConsenSys.

7.5 Ethereum Virtual Machine και DApps

Το Ethereum Virtual Machine (EVM) αποτελεί τον πυρήνα της πλατφόρμας Ethereum. Το EVM είναι ένας εικονικός υπολογιστής όπως φαίνεται στην παρακάτω εικόνα, ένα περιβάλλον εκτέλεσης που λειτουργεί σε όλους τους κόμβους του δικτύου Ethereum, επιτρέποντας την εκτέλεση κώδικα και τη διαχείριση δεδομένων στο δίκτυο. Το EVM επιτρέπει τη δημιουργία και την εκτέλεση των λεγόμενων "έξυπνων συμβολαίων" (smart contracts), τα οποία είναι προγράμματα που εκτελούνται αυτόματα όταν πληρούνται ορισμένες προκαθορισμένες συνθήκες, παρέχοντας ένα υψηλό επίπεδο ασφάλειας, διαφάνειας και αξιοπιστίας. [57]

Το EVM αποτελεί τη βάση για τα Decentralized Apps (DApps), οι οποίες είναι εφαρμογές που λειτουργούν σε ένα peer-to-peer δίκτυο υπολογιστών αντί για έναν κεντρικό server. Διαφέρουν από τις παραδοσιακές εφαρμογές στο ότι δεν ελέγχονται από μία μόνο αρχή, αλλά λειτουργούν σε ένα αποκεντρωμένο δίκτυο nodes, προσφέροντας μεγαλύτερη διαφάνεια, ασφάλεια και ανθεκτικότητα στη λογοκρισία. Επιπλέον, τα DApps επωφελούνται από την αμεσότητα και την αποτελεσματικότητα των συναλλαγών μέσω blockchain, καθώς και από την δυνατότητα ενσωμάτωσης των έξυπνων συμβολαίων για αυτοματοποιημένες συναλλαγές. [58]

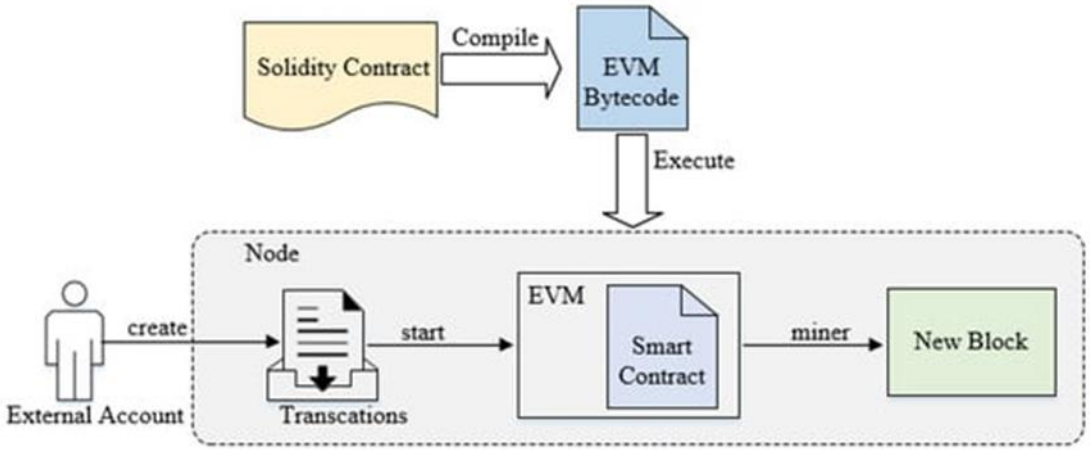


Εικόνα 1: Αρχιτεκτονική Ethereum Virtual Machine (EVM)

Διαθέσιμο Online: https://cypherpunks-core.github.io/ethereumbook/13evm.html

Ωστόσο, τα DApps αντιμετωπίζουν επίσης προκλήσεις. Η ανάπτυξη και διαχείριση DApps απαιτεί σημαντική τεχνική εμπειρία και κατανόηση των blockchain τεχνολογιών. Επιπλέον, οι ζητήματα απόδοσης του Ethereum μπορούν να οδηγήσουν σε αυξημένα κόστη συναλλαγών και καθυστερήσεις. [59]

Τα smart contracts στο Ethereum είναι μικρά προγράμματα που εκτελούνται στο EVM και είναι γραμμένα σε γλώσσες προγραμματισμού υψηλού επιπέδου, όπως το Solidity. Μεταγλωττίζονται σε bytecode, το οποίο το EVM μπορεί να αναγνωρίσει και να εκτελέσει. Τα smart contracts μπορούν να εκτελέσουν μια ποικιλία συναλλαγών, από τη μεταφορά κρυπτονομισμάτων έως την αυτοματοποιημένη διαχείριση συμφωνιών και τη δημιουργία πολύπλοκων DApps.



Εικόνα 2: Λειτουργία smart contract EVM

Διαθέσιμο Online: https://www.mdpi.com/2079-9292/12/10/2152

Υπάρχουν πολλές εφαρμογές που χρησιμοποιούν το Ethereum και το EVM για να δημιουργήσουν ενδιαφέρουσες DApps. Εφαρμογές όπως το Uniswap, ένα αποκεντρωμένο ανταλλακτήριο που επιτρέπει την ανταλλαγή διαφόρων κρυπτονομισμάτων, και το Compound, μια πλατφόρμα αποκεντρωμένης χρηματοδότησης (DeFi) που επιτρέπει στους χρήστες να δανείζονται και να δίνουν δάνεια κρυπτονομίσματα, έχουν αποδείξει τη δυναμική και το δυναμικό του Ethereum ως πλατφόρμα για τη δημιουργία ποικίλων και καινοτόμων DApps. Επίσης, πλατφόρμες όπως το MakerDAO, που επιτρέπει τη δημιουργία του stablecoin DAI, και το Decentraland, μια εικονική πλατφόρμα παιχνιδιού και κοινότητας, αποτελούν παραδείγματα της δυνατότητας του Ethereum να υποστηρίξει μια ποικιλία καινοτόμων εφαρμογών. Αυτές οι εφαρμογές αποδεικνύουν την ευελιξία και την πολυμορφία των DApps και πώς μπορούν να αξιοποιήσουν την τεχνολογία blockchain για να προσφέρουν καινοτόμες και ασφαλείς υπηρεσίες.

Κεφάλαιο 8: Τεχνολογίες και Εργαλεία

Εισαγωγή

Η εμφάνιση της τεχνολογίας blockchain έχει εισάγει μια νέα εποχή ψηφιακής καινοτομίας, η οποία μπορεί να εφαρμοστεί σε πολυάριθμους τομείς και βιομηχανίες. Ένας τέτοιος τομέας όπου η επίδραση του blockchain είναι έντονα αισθητή είναι στον τομέα της ψηφιακής επαλήθευσης ταυτότητας. Αυτό το κεφάλαιο εμβαθύνει στην πρακτική εφαρμογή του blockchain σε ένα ακαδημαϊκό περιβάλλον, εστιάζοντας στη δημιουργία ενός συστήματος ψηφιακής ταυτότητας κυρίως για τους φοιτητές αλλά και για το προσωπικό ενός πανεπιστημίου.

Ο θεμελιώδες στόχος αυτού του κεφαλαίου είναι να αναλυθούν με λεπτομέρεια οι πρακτικές πτυχές της αναπτυγμένης εφαρμογής, εξηγώντας τις περίπλοκες λεπτομέρειες της εφαρμογής blockchain σε ένα πρακτικό πλαίσιο. Ο στόχος είναι η παροχή μιας πλήρους περιγραφής του σχεδιασμού, της ανάπτυξης και των λειτουργικών δυνατοτήτων του συστήματος καθώς και τον τρόπο που αυτές σχετίζονται με την επαλήθευση και διαχείριση της ψηφιακής ταυτότητας.

Στο παρόν κεφαλαίο θα αναλυθούν τα ακόλουθα κύρια μέρη της αναπτυγμένης εφαρμογής:

- **Τεχνολογίες και Εργαλεία**: Ο σύγχρονος κόσμος του προγραμματισμού, προσφέρει μια πληθώρα τεχνολογιών, εργαλείων και βιβλιοθηκών που αυξάνουν σημαντικά την εμπειρία ανάπτυξης λογισμικού (Development Experience) καθώς μειώνουν το όγκο του κώδικα που πρέπει να γραφτεί. Καθώς η εφαρμογή αποτελείται από τέσσερα μέρη, για την βέλτιστη οργάνωση του εντύπου οι τεχνολογίες έχουν οργανωθεί βάση του κομματιού στο οποίο χρησιμοποιήθηκαν.

- **Έξυπνα Συμβόλαια** (Smart Contracts): Αποτελούν την ραχοκοκαλιά των αποκεντρωμένων εφαρμογών (DApps). Δημιουργήθηκαν 2 έξυπνα συμβόλαια, το *CryptoPass* και το *AccessToken* με τη χρήση της γλώσσας προγραμματισμού Solidity.

- **Η Βιβλιοθήκη Web3Button**: Η μετατροπή μιας παραδοσιακής διαδικτυακή εφαρμογής σε DApp (ή Web3 App) αποτελεί έργο ιδιαίτερα δύσκολο, ειδικά εάν η εκάστοτε ομάδα ανάπτυξης ή συντήρησης της εφαρμογής δεν γνωρίζει για την τεχνολογία blockchain. Γι' αυτόν τον λόγο δημιουργήσαμε ένα καινοτόμο module οπού αναλαμβάνει την σύνδεση, την ταυτοποίηση και επίσης ελέγχει εάν ο εκάστοτε χρήσης έχει τα απαραίτητα δικαιώματα για αποκτήσει πρόσβαση σε μια υπηρεσία.

- **Λογική Πλευράς Διακομιστή** (Server-Side Logic): Η υποδομή του διακομιστή είναι τροφοδοτούμενη από το web framework Hono [60], το οποίο συντονίζει την αλληλεπίδραση μεταξύ των έξυπνων συμβολαίων και της πρόσοψης της εφαρμογής (frontend). Παρόλο που η χρήση διακομιστών είναι μια δράση που γενικά αποθαρρύνεται κατά το σχεδιασμό των DApps, καθώς δημιουργεί κεντροποίηση και single points of failure, στην προκείμενη περίπτωση η χρήση ενός διακομιστή μας παρέχει αρκετά οφέλη και η φύση του συστήματος το επιτρέπει.

- **Διεπαφή χρήστη** (User Interface/Frontend): Η διεπαφή χρήστη που έχει δημιουργεί δεν αποτελεί μια ολοκληρωμένη εφαρμογή. Ο σκοπός της είναι να παρέχει έναν

οπτικό και διαδραστικό τρόπο παρουσίασης των διάφορων εξαρτημάτων και δυνατοτήτων της παρούσας εφαρμογής.

- **Προσομοίωση Δημιουργίας Ψηφιακού Πάσου**: Η ενότητα εξετάζει λεπτομερώς την αναπτυγμένη εφαρμογή, εστιάζοντας στις διαδικασίες εγκατάστασης και εκτέλεσης της εφαρμογής, με σκοπό την παρουσίαση τεσσάρων λειτουργιών: ενεργοποίηση υπηρεσιών, ταυτοποίηση χρήστη, δημιουργία Soulbound Token και παραγωγή Access Token. Ακόμα, παρέχει μια καθαρή και οργανωμένη παρουσίαση της διαδικασίας ανάπτυξης και της πρακτικής εφαρμογής των καινοτόμων λύσεων. Αναλύει το απαραίτητο λογισμικό και τα βήματα για την απόκτηση και εκτέλεση του πηγαίου κώδικα. Επιπλέον, παρουσιάζει τέσσερα σενάρια χρήσης, τα οποία περιγράφουν διαδοχικά τις διαδικασίες σχετικά με την εκκίνηση των υπηρεσιών, την χρήση της βιβλιοθήκης «*Web3Button*» για ταυτοποίηση, την δημιουργία ενός SBT για έναν φοιτητή, την δημιουργία και χρήση ενός Access Token.

Ως θεμέλιο της εφαρμογής, τα έξυπνα συμβόλαια λειτουργούν ως ο αμετάβλητος κατάλογος που καταγράφει και επαληθεύει κάθε συναλλαγή και αλληλεπίδραση. Η ανάπτυξη επικεντρώνεται στη χρήση μιας παραλλαγής των προτύπων ERC721 (γνωστά και ως NFTs), τα Soul Bound Tokens (SBTs)[61].

Η κυριά διαφορά ανάμεσα στα δυο πρότυπα, έγκειται στο γεγονός ότι, στα SBTs έχει αφαιρεθεί η ιδιότητα της ανταλλαξιμότητας καθώς και ότι ένα κρυπτοπορτοφόλι δεν μπορεί να έχει στην κατοχή του, την ιδιά στιγμή, πάνω από ένα SBT.

Στην εφαρμογή αυτό το πρότυπο κρυπτονομίσματος θα χρησιμοποιηθεί για να την ψηφιοποιήση των ταυτοτήτων των φοιτητών, παρέχοντας έτσι μια πύλη για ασφαλή και αποτελεσματική πρόσβαση σε μια πληθώρα ψηφιακών και φυσικών υπηρεσιών, δίχως την ανάγκη κατοχής ενός φυσικού αποδεικτικού ή την χρήση πολλαπλών λογαριασμών.

8.1 Έξυπνα Συμβόλαια

- **OpenZeppelin**: Ο οργανισμός OpenZeppelin έχει καθιερωθεί ως ένας θεμελιώδης πόρος στον κόσμο του blockchain και της ανάπτυξης έξυπνων συμβολαίων. Προσφέρει μια σειρά από ασφαλή και επαναχρησιμοποιήσιμα εργαλεία που έχουν υποστεί πολλαπλές δοκιμές και ελέγχους, θέτοντας τον ως πρότυπο για τον κώδικα που χρησιμοποιείται στην δημιουργία αποκεντρωμένων εφαρμογών στο Ethereum. Οι βιβλιοθήκες του OpenZeppelin είναι ιδιαίτερα εκτιμημένες για την αξιοπιστία τους και για την ευκολία με την οποία μπορούν να ενσωματωθούν σε προϊόντα blockchain, επιτρέποντας στους προγραμματιστές να αναπτύξουν λειτουργίες πιο γρήγορα και να μειώσουν τον κίνδυνο ασφαλείας στα έργα τους. Με μια ενεργή κοινότητα και συνεχή υποστήριξη, το OpenZeppelin αποτελεί ένα ζωτικό εργαλείο για κάθε αναπτυξιακό έργο που επιδιώκει να εκμεταλλευτεί τις δυνατότητες της τεχνολογίας blockchain.

- **Hardhat:** Το Hardhat είναι ένα ευέλικτο και επεκτάσιμο περιβάλλον ανάπτυξης για το Ethereum. Προτιμάται για το ολοκληρωμένο πλαίσιο δοκιμών του, το οποίο διευκολύνει τόσο τη συγγραφή σεναρίων όσο και την αλληλεπίδραση με το δίκτυο Ethereum. Το Hardhat παρέχει στους προγραμματιστές ένα τοπικό περιβάλλον

blockchain, όπου μπορούν να αναπτύσσουν συμβόλαια, να τρέχουν δοκιμές και να εντοπίζουν σφάλματα στον κώδικα.

- **Dotenv**: Η βιβλιοθήκη dotenv είναι ένα απαραίτητο εργαλείο στην ανάπτυξη εφαρμογών, καθώς παρέχει μια απλή και ασφαλή μέθοδο για τη διαχείριση ιδιωτικών και ευαίσθητων δεδομένων. Με τη χρήση της, οι προγραμματιστές μπορούν να φορτώσουν μεταβλητές περιβάλλοντος από ένα αρχείο .env στον χώρο εργασίας τους, κρατώντας τον κώδικα καθαρό από ευαίσθητες πληροφορίες και επιτρέποντας την εύκολη προσαρμογή των εφαρμογών σε διάφορα περιβάλλοντα. Αυτή η προσέγγιση βοηθάει στην πρόληψη της διαρροής σημαντικών δεδομένων και στη διευκόλυνση της διαδικασίας ανάπτυξης, ενώ συμβάλλει στην αυτοματοποίηση και την επεκτασιμότητα των προγραμμάτων. Η ευρεία υιοθέτηση και η υποστήριξη της βιβλιοθήκης dotenv αναδεικνύουν την αξία της στη σύγχρονη ανάπτυξη λογισμικού.

8.2 Web3 Button Module

- **Metamask/detect-provider**: Η βιβλιοθήκη *@metamask/detect-provider* παίζει έναν κρίσιμο ρόλο στην ενσωμάτωση των εφαρμογών DApp με το *MetaMask*, το οποίο είναι ένα από τα πιο διαδεδομένα πορτοφόλια Ethereum που λειτουργεί ως επέκταση περιηγητή. Αυτή η βιβλιοθήκη επιτρέπει στους προγραμματιστές να ανιχνεύσουν αξιόπιστα και αποτελεσματικά την παρουσία του παρόχου MetaMask στο περιβάλλον του χρήστη. Με τη χρήση της, μπορούν εύκολα να διασφαλίσουν ότι οι ενέργειες που απαιτούν την αλληλεπίδραση με το blockchain μπορούν να πραγματοποιηθούν μόνο εάν ο πάροχος είναι παρών, εξασφαλίζοντας μια ομαλή και άμεση εμπειρία χρήστη. Η @metamask/detect-provider αποτελεί λοιπόν μια θεμελιώδη διευκόλυνση στην ανάπτυξη των DApps, καθώς προσφέρει μια απλοποιημένη και σταθερή μέθοδο για την επίτευξη συνεργασίας μεταξύ του frontend και των κρυπτογραφικών πορτοφολιών.

- **Axios**: Η βιβλιοθήκη Axios είναι ένας δημοφιλής JavaScript HTTP client που λειτουργεί τόσο στο περιβάλλον του browser όσο και σε αυτό του Node.js, παρέχοντας μια απλή και σύγχρονη διεπαφή για την εκτέλεση αιτήσεων HTTP. Διακρίνεται για την ευελιξία και την ευκολία στη χρήση, προσφέροντας υποστήριξη για τις υποσχέσεις της JavaScript (promises) και πλούσιες δυνατότητες διαμόρφωσης αιτημάτων. Με την Axios, προγραμματιστές μπορούν να υλοποιούν αιτήσεις GET, POST και των υπολοίπων HTTP μεθόδων με μεγάλη ευκολία, να διαχειρίζονται τις απαντήσεις και να προσαρμόζουν τις αιτήσεις με τη χρήση interceptors. Η δημοτικότητα της Axios στην ανάπτυξη μοντέρνων web και Node.js εφαρμογών έχει εδραιωθεί χάρη στη σταθερότητα, την απόδοση και την ικανότητα ανταπόκρισης σε πολύπλοκες απαιτήσεις δικτύωσης.

- **Csstype**: Η βιβλιοθήκη "csstype" είναι ένας τύπος βοηθητικής βιβλιοθήκης για την JavaScript που παρέχει συγκεκριμένους ορισμούς τύπων για CSS properties και τιμές συμβατές με την TypeScript. Αυτή η βιβλιοθήκη είναι πολύτιμη για προγραμματιστές που θέλουν να υλοποιήσουν ισχυρούς τύπους δεδομένων στις inline CSS δηλώσεις τους, ενισχύοντας την ασφάλεια τύπου και μειώνοντας τα λάθη που συνδέονται με την ανάπτυξη του στυλ των εφαρμογών τους. Η "csstype" προσφέρει εκτεταμένη υποστήριξη για την αυτόματη συμπλήρωση και την επιβεβαίωση τύπων σε περιβάλλοντα που υποστηρίζουν TypeScript, διευκολύνοντας την ανάπτυξη και τη συντήρηση του CSS κώδικα μέσα στο JavaScript έργο. Εν συνεχεία, αυτό βελτιώνει

τη διαδικασία ανάπτυξης προσφέροντας μια πιο αυστηρή και οργανωμένη διαχείριση των στυλ.

- **Ethers.js**: Η βιβλιοθήκη "ethers.js" αποτελεί μία συλλογή από εργαλεία και βιβλιοθήκες που σχεδιάστηκαν για να απλοποιούν την αλληλεπίδραση με το Ethereum blockchain και τον προγραμματισμό των έξυπνων συμβολαίων. Είναι γραμμένη σε JavaScript και προορίζεται για χρήση τόσο σε περιβάλλοντα browser όσο και στο Node.js, προσφέροντας έναν εύχρηστο και ελαφρύ τρόπο για την εκτέλεση εργασιών όπως η σύνδεση με peers του Ethereum δικτύου, η δημιουργία και διαχείριση πορτοφολιών, η κωδικοποίηση και αποκωδικοποίηση δεδομένων και η αποστολή συναλλαγών. Επιπλέον, "ethers.js" είναι ευρέως αποδεκτή για την υποστήριξη που προσφέρει σε πρότυπα ασφάλειας και για την ενσωμάτωσή της με άλλες βιβλιοθήκες και πρότυπα του Ethereum, κάνοντας την μια απαραίτητη προσθήκη στο εργαλειοθήκη κάθε blockchain προγραμματιστή.

- **Toastify-js**: Η βιβλιοθήκη "toastify-js" είναι ένα ελαφρύ και ευέλικτο πακέτο για την παραγωγή ειδοποιήσεων στο περιβάλλον του web, που προσφέρει μια απλή μέθοδο για την προβολή προσαρμόσιμων toast μηνυμάτων. Χωρίς να επιβαρύνει την απόδοση της εφαρμογής, η "toastify-js" επιτρέπει στους προγραμματιστές να προσθέτουν εντυπωσιακές ειδοποιήσεις για άμεση ανατροφοδότηση στους χρήστες τους. Είτε πρόκειται για επιβεβαίωση ενός επιτυχημένου συμβάντος, είτε για προειδοποίηση για κάποιο σφάλμα, η "toastify-js" προσφέρει μια οπτικά ελκυστική και χρηστική λύση. Η χρήση της είναι απλή και διαισθητική, με ένα API που διευκολύνει την ταχεία ανάπτυξη και επιτρέπει εύκολες προσαρμογές στο στυλ και τη συμπεριφορά των toast μηνυμάτων, καθιστώντας την μια ιδανική επιλογή για τη βελτίωση της εμπειρίας χρήστη σε σύγχρονες ιστοσελίδες και web εφαρμογές.

- **TypeScript**: Η TypeScript είναι μια ανοιχτού κώδικα γλώσσα προγραμματισμού που αναπτύχθηκε από τη Microsoft και συνδυάζει την ευελιξία της JavaScript με ισχυρά χαρακτηριστικά τύπων δεδομένων, βοηθώντας στην παραγωγή πιο καθαρού και διαχειρίσιμου κώδικα. Προσφέρει προαιρετική στατική τυποποίηση και την δυνατότητα για την χρήση τελευταίων δυνατοτήτων του ECMAScript, καθιστώντας την ιδανική για την ανάπτυξη μεγάλων εφαρμογών ή όταν απαιτείται βελτιωμένη ασφάλεια τύπου. Η TypeScript ενσωματώνεται άριστα με υπάρχοντα JavaScript frameworks και βιβλιοθήκες, παρέχοντας ταυτόχρονα εργαλεία για την αποκάλυψη σφαλμάτων πριν από την εκτέλεση του κώδικα. Με την πλούσια κοινότητα και εκτενή τεκμηρίωση, η TypeScript έχει κερδίσει την προτίμηση αναπτυξιακών ομάδων παγκοσμίως, ενισχύοντας την αξιοπιστία και την παραγωγικότητα στη δημιουργία λογισμικού.

8.3 Λογική Πλευράς Διακομιστή

- **Hono Web Framework**: Η βιβλιοθήκη "Hono" είναι ένα ελαφρύ και αποδοτικό web framework για το Node.js, σχεδιασμένο για την ανάπτυξη ταχείας και κλιμακούμενης server-side λογικής. Αυτό το framework προσφέρει μια διαισθητική API για τη διαχείριση των δρομολογήσεων, των αιτήσεων και των αποκρίσεων, ενισχύοντας την ευκολία στην ανάπτυξη RESTful APIs. Χάρη στην ελαχιστοποιημένη προσέγγισή του, το "Hono" αποτελεί μία ελκυστική επιλογή για προγραμματιστές που αναζητούν μια γρήγορη και ευέλικτη λύση για την κατασκευή της back-end λογικής τους. Επίσης, το "Hono" προσφέρει τη δυνατότητα εύκολης ενσωμάτωσης με άλλες

βιβλιοθήκες και εργαλεία του Node.js ecosystem, καθιστώντας την ανάπτυξη πιο αποδοτική και την συντήρηση του κώδικα απλούστερη.

- **Τεχνολογίες που χρησιμοποιούνται αλλά έχουν ήδη αναφερθεί παραπάνω:**
  - Dotenv
  - Ethers.js
  - TypeScript

8.4 Διεπαφή χρήστη

- **React**: Η React είναι μια δημοφιλής βιβλιοθήκη JavaScript ανοικτού κώδικα που αναπτύχθηκε από το Meta (πρώην Facebook) για την κατασκευή δυναμικών και ευέλικτων διεπαφών χρήστη. Χαρακτηρίζεται από την αποτελεσματική δημιουργία διεπαφών μέσω των συνθετικών εξαρτημάτων της (components), τα οποία ενθαρρύνουν την επαναχρησιμοποίησης κώδικα στην ανάπτυξη εφαρμογών. Η React χρησιμοποιεί το εικονικό DOM (Virtual DOM) για να βελτιστοποιήσει την απόδοση, επιτρέποντας γρήγορες ανανεώσεις και αποδοτική αναδόμηση της UI κατά τη διάρκεια της αλλαγής καταστάσεων της εφαρμογής. Η δημοτικότητα της React προέρχεται επίσης από τη δυνατότητα εύκολης ενσωμάτωσης με πλήθος άλλων τεχνολογιών και το οικοσύστημά της, που περιλαμβάνει εργαλεία όπως το Redux για διαχείριση καταστάσεων και το React Router για δρομολόγηση, καθιστώντας την μία από τις προτιμώμενες λύσεις για την ανάπτυξη μοντέρνων web και native εφαρμογών.

- **React-dom**: Η βιβλιοθήκη "react-dom" είναι μια απαραίτητη επέκταση της βιβλιοθήκης React, σχεδιασμένη ειδικά για τη διαχείριση της αλληλεπίδρασης μεταξύ των React components και του DOM (Document Object Model) σε περιηγητές. Αυτή η βιβλιοθήκη επιτρέπει στους προγραμματιστές να εκτελούν δυναμικές ενημερώσεις της διεπαφής χρήστη, χωρίς να απαιτείται η πλήρης επαναφόρτωση της σελίδας. Το "react-dom" περιλαμβάνει σημαντικές μεθόδους όπως το ReactDOM.render, το οποίο συνδέει τα React elements με το DOM, καθιστώντας τα ορατά στην οθόνη του χρήστη. Επιπλέον, παρέχει λειτουργίες για την ασφαλή διαχείριση συμβάντων και για την επεξεργασία των στοιχείων του DOM με τρόπο που είναι συμβατός με την αρχιτεκτονική της React. Η χρήση της "react-dom" είναι θεμελιώδης για την ανάπτυξη οποιασδήποτε εφαρμογής που χρησιμοποιεί τη React στο web, καθώς διασφαλίζει την αποδοτική, γρήγορη και ομαλή αλληλεπίδραση μεταξύ των διεπαφών της εφαρμογής και της δομής των web σελίδων.

- **JsQR**: Η βιβλιοθήκη "jsQR" είναι ένας απλός και ισχυρός αναγνώστης QR κωδίκων γραμμένος σε JavaScript, ο οποίος επιτρέπει την εύκολη ανίχνευση και αποκωδικοποίηση QR κωδίκων από εικόνες σε ιστοσελίδες. Η "jsQR" είναι σχεδιασμένη να λειτουργεί αποδοτικά σε διάφορα περιβάλλοντα, παρέχοντας μια ευέλικτη API που δέχεται ως είσοδο τα δεδομένα εικόνας και επιστρέφει τα αποτελέσματα της ανάγνωσης. Αυτό την καθιστά ιδανική για εφαρμογές web που απαιτούν τη λειτουργία ανάγνωσης QR κωδίκων, όπως μηχανές ελέγχου εισιτηρίων, συστήματα πληρωμών, και άλλες εφαρμογές που σχετίζονται με την ταυτοποίηση ή την κοινή χρήση πληροφοριών. Με την απλότητά της και την υψηλή απόδοση, η "jsQR" αποτελεί μια πολύτιμη προσθήκη στην οικογένεια βιβλιοθηκών JavaScript,

παρέχοντας μια αξιόπιστη λύση για την ενσωμάτωση ανάγνωσης QR σε διαδικτυακές εφαρμογές.

- **React-qr-code**: Η βιβλιοθήκη "react-qr-code" είναι ένα εργαλείο βασισμένο στη React, το οποίο επιτρέπει την απλή και διαισθητική δημιουργία QR κωδίκων εντός των React εφαρμογών. Χρησιμοποιώντας τα React components, η "react-qr-code" παρέχει μια ομαλή ενσωμάτωση με την διεπαφή χρήστη, καθιστώντας την δημιουργία και παρουσίαση των QR κωδίκων μια απλή διαδικασία. Η βιβλιοθήκη αυτή είναι ιδιαίτερα χρήσιμη σε εφαρμογές που απαιτούν την γρήγορη και αποτελεσματική ανταλλαγή δεδομένων, όπως η αποστολή διευθύνσεων ή άλλων πληροφοριών με ασφαλή τρόπο. Το "react-qr-code" εκμεταλλεύεται την δύναμη της React για την δημιουργία δυναμικών, προσαρμόσιμων και υψηλής απόδοσης QR κωδίκων, προσφέροντας μια στιβαρή λύση σε αναπτυσσόμενες εφαρμογές που επιθυμούν να ενσωματώσουν την λειτουργία αυτή με ελάχιστο κόπο και χωρίς την ανάγκη για εξωτερικά dependencies.

- **Vite**: Η "Vite" είναι μία σύγχρονη και εξαιρετικά γρήγορη βιβλιοθήκη κατασκευής (build tool) για τον προγραμματισμό front-end, που επικεντρώνεται στην ταχεία ανάπτυξη και ευκολία χρήσης. Χρησιμοποιώντας την JavaScript API του κατασκευαστή εντολών (build commands) και την υποστήριξη για τις πιο δημοφιλείς γλώσσες και πλαίσια όπως React, Vue και Svelte, η "Vite" προσφέρει μια εμπειρία ανάπτυξης χωρίς τριβές, αυξάνοντας την απόδοση του hot-module replacement και προσφέροντας βελτιωμένη ανατροφοδότηση στον προγραμματιστή. Επίσης, διαθέτει ενσωματωμένη υποστήριξη για την TypeScript και την δυνατότητα για αυτόματο split code, βοηθώντας στην δημιουργία πιο αποδοτικών εφαρμογών. Με την "Vite", οι προγραμματιστές μπορούν να περιμένουν μειωμένους χρόνους φόρτωσης και απλοποιημένη διαχείριση των εξαρτήσεων τους, κάνοντας την ιδανική για επιχειρήσεις που απαιτούν μια σύγχρονη και αποδοτική ανάπτυξη workflow.

- **Eslint**: Η βιβλιοθήκη "ESLint" είναι ένα εξαιρετικά ισχυρό εργαλείο στατικής ανάλυσης κώδικα για γλώσσες βασισμένες στη JavaScript και JSX. Το ESLint βοηθά τους προγραμματιστές να εντοπίζουν και να διορθώνουν προβλήματα στον κώδικα τους, προάγοντας πρακτικές καλού προγραμματισμού και συμβατότητα με πρότυπα κωδικοποίησης. Με ένα πλούσιο οικοσύστημα προσαρμοσμένων κανόνων και προσθηκών, το ESLint επιτρέπει την προσαρμογή της ανάλυσης κώδικα στις ανάγκες του εκάστοτε έργου, ενώ παρέχει ενσωματωμένη υποστήριξη για την εντοπισμό κοινών σφαλμάτων και αντιπαραβάσεων των best practices. Η ευκολία χρήσης του ESLint, μαζί με την ευελιξία του να διαμορφώνει το coding style βάσει των προτιμήσεων της ομάδας ανάπτυξης, το καθιστά ένα απαραίτητο εργαλείο για σύγχρονους προγραμματιστές που επιδιώκουν υψηλή ποιότητα και συνέπεια στον κώδικα τους.

- **Τεχνολογίες που χρησιμοποιούνται αλλά έχουν ήδη αναφερθεί παραπάνω:**
  - Axios
  - Ethers.js
  - TypeScript

Κεφάλαιο 9: Έξυπνα Συμβόλαια

Εισαγωγή

Στο πλαίσιο της εφαρμογής μας για την ψηφιοποιήση του φοιτητικού πάσου, τα έξυπνα συμβόλαια είναι αυτόνομα προγράμματα που βρίσκονται στο δίκτυο blockchain. Για να εκτελεστεί ο κώδικας τους πρέπει να μια εξωτερική οντότητα να τα καλέσει. Η εφαρμογή μας απαιτεί την χρήση δύο έξυπνων συμβολαίων για να λειτουργήσει: το **CryptoPass** και το **AccessToken**. Αυτά τα συμβόλαια, γραμμένα σε Solidity, διέπουν τη δημιουργία και διαχείριση ψηφιακών ταυτοτήτων και διακριτικών πρόσβασης, αντίστοιχα.

9.1 Έξυπνο Συμβόλαιο: CryptoPass

9.1.1 Σκοπός και Σχεδίαση

Το έξυπνο συμβόλαιο CryptoPass είναι το κεντρικό σημείο για τη δημιουργία και διαχείριση των *Soulbound Tokens* (SBTs) που αντιπροσωπεύουν τα ψηφιακά πασά των φοιτητών και του προσωπικού του πανεπιστημίου. Το συμβόλαιο εκμεταλλεύεται τις δυνατότητες του διακριτικού *ERC-721*, το οποίο επιτρέπει τη δημιουργία μοναδικών, αλλά ανταλλάξιμων διακριτικών. Στην δική μας περίπτωση, επιθυμούμε την ύπαρξη μοναδικών διακριτών αλλά αυτά πρέπει είναι **μη-ανταλλάξιμα** και **ο κάθε χρήστης θα πρέπει να μπορεί να έχει στην κατοχή του μόνο ένα**. Για να επιτευχθεί αυτό χρειάστηκε να τροποποιήσουμε μερικές εσωτερικές συναρτήσεις του ERC-721. Παρακάτω αναλύονται οι βασικές λειτουργίες του έξυπνου συμβολαίου.

9.1.2 Κύριες Λειτουργίες

- **Δημιουργία SBTs**: Υπάρχει μια συνάρτηση, ονόματι '*createSBT'*, που σκοπός της είναι η δημιουργία ενός SBT καθώς και η σύνδεση του με ένα metadata, τον ρολό. Ο ρόλος χρησιμοποιείται για την διαχείριση της προσβασιμότητας των χρηστών, καθώς είναι πολύ πιθανόν μερικές υπηρεσίες να πρέπει να είναι διαθέσιμες μόνο σε συγκεκριμένες ομάδες χρηστών, όπως για παράδειγμα καθηγητές ή τεχνικό προσωπικό. Η χρήση αυτής της συνάρτησης είναι περιορισμένη σε εξουσιοδοτημένους λογαριασμούς για να διατηρηθεί η ακεραιότητα και ο έλεγχος της έκδοσης ταυτότητας. Αυτό επιτυγχάνεται μέσω της χρήσης ενός τροποποιητή (modifier), ονόματι *onlyAuthAcc*, ο οποίος δεν επιτρέπει να την εκτέλεση της συνάρτησης από διευθύνσεις στις οποίες δεν έχει δοθεί η απαραίτητη εξουσιοδότηση.

- **Ανάθεση Ρόλων**: Το συμβόλαιο κατηγοριοποιεί τα SBTs (ή ψηφιακές ταυτότητές) αναθέτοντας ρόλους, όπως για παράδειγμα Φοιτητής ή Καθηγητής, μέσω ενός απαριθμημένου τύπου (Enum). Οι ρόλοι καθορίζουν το επίπεδο πρόσβασης και τα προνόμια που κατέχει κάθε διακριτικό ψηφιακής ταυτότητας. Για την βέλτιστη οργάνωση του κώδικα, δημιουργήθηκε ένα ξεχωριστό αρχείο, ονόματι Types.sol, οπού γίνεται χρήση της λέξης κλειδί *library* (βιβλιοθήκη) αντί για *contract* (συμβόλαιο) για να την δημιουργία μιας κλάσης (Εικόνα 3). Η κύρια διαφορά ανάμεσα σε μια βιβλιοθήκη και ένα συμβόλαιο, έγκειται στο γεγονός ότι οι βιβλιοθήκες δεν μπορούν να έχουν μεταβλητές αποθήκευσης (storage variables) και δεν μπορούν να προωθηθούν (deploy) αυτοτελώς. Χρησιμοποιούνται κατά κύριο

λόγο για να ομαδοποιήσουν συναρτήσεις που μπορούν να επαναχρησιμοποιηθούν σε διαφορετικά συμβόλαια.

Εικόνα 3: Ο απαριθμημένος τύπος δεδομένων για τους ρόλους.

- **Διαχείριση Κύκλου Ζωής Διακριτικού**: Το CryptoPass περιλαμβάνει λειτουργίες για τη διαχείριση ολόκληρου του κύκλου ζωής ενός SBT, συμπεριλαμβανομένης της δημιουργίας, της τροποποίησης ρόλων και της καύσης (διαγραφής), αν και η τελευταία συχνά είναι περιορισμένη στο πλαίσιο των SBTs.

9.1.3 Χαρακτηριστικά Ασφαλείας

- **Τροποποιητές (modifiers)**: Το συμβόλαιο χρησιμοποιεί τροποποιητές όπως το **onlyAuthAcc** για να περιορίσει ευαίσθητες λειτουργίες σε εξουσιοδοτημένους λογαριασμούς, διασφαλίζοντας ότι μόνο συγκεκριμένοι λογαριασμοί του πανεπιστημίου μπορούν να εκδώσουν ή να τροποποιήσουν τα SBTs. Ακόμα, υπάρχει και ο τροποποιητής **onlyOne**, οποίος έχει τοποθετηθεί στην συνάρτηση **safeMint** με σκοπό να είναι αδύνατη η δημιουργία ενός SBT εάν ο λογαριασμός για τον οποίο προορίζεται ήδη κατέχει ένα.

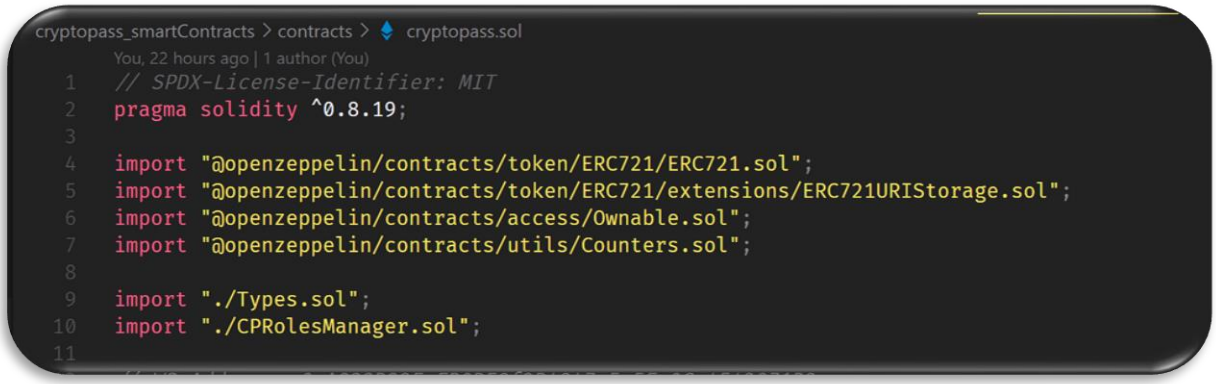
- **Μη-Μεταφερσιμότητα (Non-Transferable)**: Το συμβόλαιο τροποποιεί τη λειτουργία της συνάρτησης *_beforeTokenTransfer*, η οποία προέρχεται από το συμβόλαιο ERC-721, για να αποτρέψει τη μεταφορά των SBTs, διασφαλίζοντας ότι οι ψηφιακές ταυτότητες παραμένουν δεσμευμένες στον αρχικό κάτοχο.

9.1.4 Ανάλυση Κώδικα

Για να κατανοήσουμε σε βάθος τον τρόπο λειτουργίας του παρόντος έξυπνου συμβολαίου ας εξετάσουμε τα κυριά μέρη του κώδικα από τον οποίο απαρτίζεται:

Αρχικά, γίνεται χρήση της έκδοσης 0.8.19 και άνω της γλώσσας Solidity. Υστέρα παρατηρούμε ότι χρησιμοποιούνται *4 βιβλιοθήκες* από τον οργανισμό OpenZeppelin [62], όπως απεικονίζεται στην Εικόνα 4. Αυτές είναι εξής:

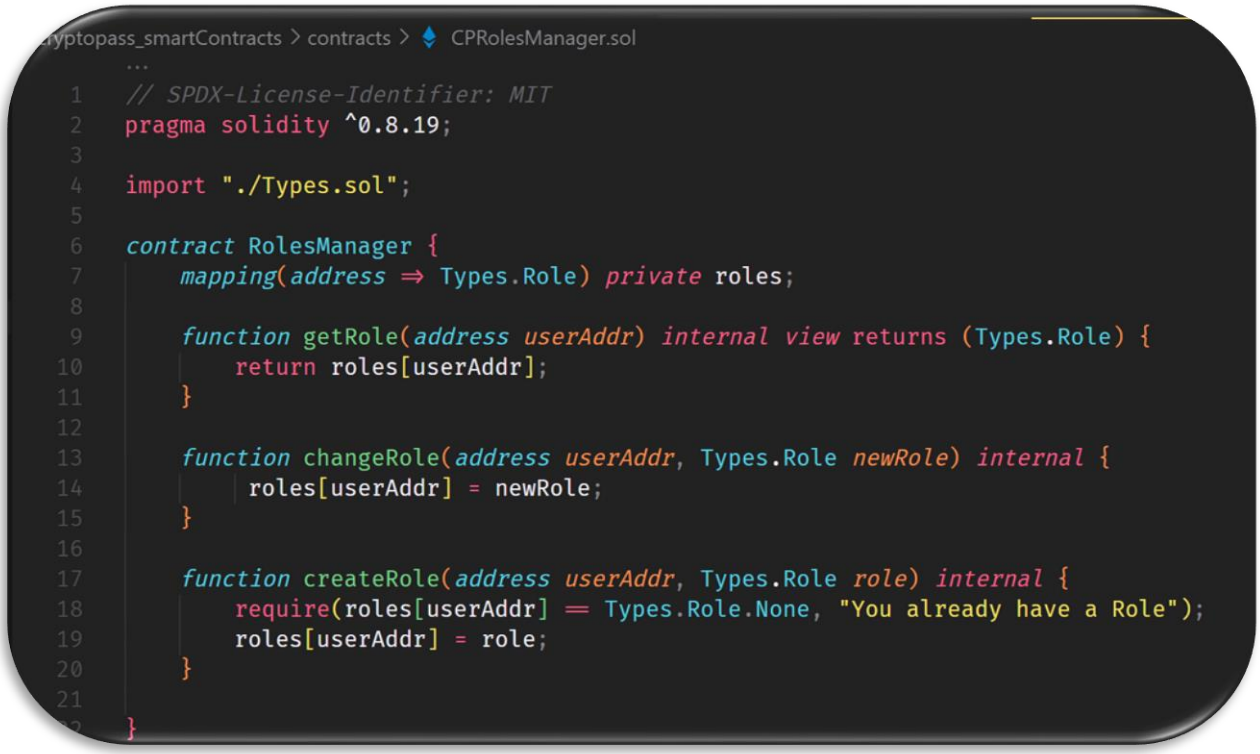
- ERC721 [63], Αυτή η βιβλιοθήκη υλοποιεί το πρότυπο ERC721, το οποίο χρησιμοποιείται ευρέως για τα μοναδικά διακριτικά (NFTs). Παρέχει τη βασική λειτουργικότητα για τη δημιουργία και διαχείριση μοναδικών διακριτικών.
- ERC721URIStorage [64], Αυτή η βιβλιοθήκη επεκτείνει τη λειτουργικότητα του ERC721 προσθέτοντας υποστήριξη για την αποθήκευση και διαχείριση μεταδεδομένων (metadata) που σχετίζονται με το κάθε διακριτικό. Επιτρέπει την συσχέτιση ενός μοναδικού URI (Uniform Resource Identifier) με κάθε διακριτικό, το οποίο μπορεί να χρησιμοποιηθεί για την ανάκτηση πρόσθετων πληροφοριών για το διακριτικό.
- Ownable [65], Αυτή η βιβλιοθήκη παρέχει έναν βασικό μηχανισμό ελέγχου πρόσβασης για συμβόλαια. Επιτρέπει τη δημιουργία ενός ιδιοκτήτη που έχει ειδικά προνόμια, όπως η δυνατότητα μεταβίβασης ιδιοκτησίας ή η εκτέλεση ορισμένων διοικητικών εργασιών. Παρέχει επίσης τροποποιητές (modifiers) που μπορούν να χρησιμοποιηθούν για τον περιορισμό της πρόσβασης σε συγκεκριμένες συναρτήσεις ή λειτουργίες.
- Counters [66], Αυτή η βιβλιοθήκη παρέχει έναν απλό τρόπο για την δημιουργία και να διαχείριση μετρητών (counters) στη Solidity. Οι μετρητές είναι χρήσιμοι για την παραγωγή μοναδικών αναγνωριστικών (IDs) ή την καταμέτρηση του αριθμού ορισμένων γεγονότων (events). Παρέχει λειτουργίες για την αύξηση, τη μείωση και την ανάκτηση της τρέχουσας τιμής ενός μετρητή.



Εικόνα 4: Οι απαιτούμενες βιβλιοθήκες για την λειτουργία του έξυπνου συμβολαίου CryptoPass.

Επιπλέον εισάγουμε και αλλά 2 αρχεία κατασκευασμένα από την συγγραφέα, το *Types.sol* (που είδαμε πιο πάνω) και *CPRolesManager.sol*. Έχοντας ήδη αναλύσει το Types, ας δούμε το CPRolesManager.

Το **CPRolesManager** είναι ένα μικρού μεγέθους συμβόλαιο σκοπός του οποίου είναι η διαχείριση των ρόλων οι οποίοι έχουν συνδεθεί με το κάθε SBT. Αποτελείται από τρεις συναρτήσεις για την *ανάκτηση*, *αλλαγή* και *δημιουργία* ενός ρόλου, όπως φαίνεται στην Εικόνα 5. Παρατηρώντας τους τροποποιητές των συναρτήσεων, βλέπουμε το τροποποιητή *internal* να είναι παρόν και στις τρεις συναρτήσεις. Αυτό υποδηλώνει οι αυτές οι συναρτήσεις έχουν σχεδιαστεί για να χρησιμοποιηθούν μονάχα από το συμβόλαιο CryptoPass ή κάποιο άλλο συμβόλαιο που κληρονομεί από αυτό.



Εικόνα 5: Το βοηθητικό συμβόλαιο RolesManager, σκοπός του είναι η διαχείριση των ρόλων.

Συνεχίζοντας με την ανάλυση του CryptoPass, το επόμενο κομμάτι που αξίζει να μελετήσουμε είναι η αρχικοποίηση των παγκοσμίων μεταβλητών (global variables), του κατασκευαστή της κλάσης (constructor) και τους δυο τροποποιητές (Εικόνα 6).

```
contract CryptoPass is ERC721, ERC721URIStorage, Ownable, RolesManager {
    using Counters for Counters.Counter;

    mapping(address => bool) public _authPersonal; // Make Private After testing

    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("CryptoPass", "CPT") {
        // Creating the Contract's Owner
        address owner = owner();

        // Making the Owner an Authorized Account
        _authPersonal[owner] = true;

        // Making the Owner an Admin
        createUserRole(owner, Types.Role.Admin);

        // We mint a SBT for the Owner,
        // Its the some as calling "createSBT" but we bypass some checks
        safeMint(owner);
    }

    // This modifier is used to check if the user already has a SBT AND
    // does not allow the creation of a 2nd one
    modifier onlyOne(address to) {
        require(
            balanceOf(to) == 0,
            "CryptoPass: You already possess a SBT and you may only have one"
        );
        _; // This is a placeholder for the code of the modified function
    }

    // This modifier is used to check if the user is authorized to create a SBT
    modifier onlyAuthAcc() {
        require(
            _authPersonal[msg.sender],
            "CryptoPass: You are NOT Authorized to create an SBT"
        );
        _;
    }
```

Εικόνα 6: CryptoPass, παγκόσμιες μεταβλητές, κατασκευαστής και τροποποιητές.

Αρχικά δηλώνουμε ότι θα χρησιμοποιήσουμε την μεταβλητή Counters για να αναφερθούμε στην βιβλιοθήκη Counters.Counter. Αυτό γίνεται για λογούς διευκόλυνσης, ώστε να γράφουμε λιγότερο κώδικα. Υστέρα δηλώνουμε μια χαρτογράφηση ή μεταβλητή αντιστοίχισής, ονόματι *_authPersonal*. Ο σκοπός της είναι η αποθήκευση των διευθύνσεων στις οποίες έχει δοθεί το δικαίωμα για την δημιουργία SBTs.

Έπειτα έχουμε τον κατασκευαστή (constructor), οποίος δημιουργεί μια μεταβλητή για να αποθηκεύσει την διεύθυνση του ιδιοκτήτη του έξυπνού συμβόλαιού. Όταν έναν συμβόλαιο κληρονομεί το *Ownable*, ο λογαριασμός που προώθησε (έκανε deploy) το συμβόλαιο αυτόματα γίνεται ο ιδιοκτήτης. Η συνάρτηση owner(),

προέρχεται από το συμβόλαιο Ownable.sol και επιστρέφει την διεύθυνση του ιδιοκτήτη. Στην παρούσα περίπτωση είναι το ίδιο με το να γράφαμε το εξής:

address owner = msg.sender;

Ύστερα, δίνουμε στον owner τα ακόλουθα:

1. Το δικαίωμα της δημιουργίας SBTs.
2. Το ρολό του Admin.
3. Του δημιουργούμε ένα SBT.

Έχοντας πλέον ολοκλήρωση την επεξήγηση του κατασκευαστή, ας προχωρήσουμε στους δυο τροποποιητές.

Ο πρώτος, ονόματι **onlyOne**, απαιτεί το υπόλοιπο (balance) του χρήστη, όσον αφορά τα SBTs, να είναι ίσο με το μηδέν. Αν αυτό δεν ισχύει, καταργεί την συναλλαγή και στέλνει το εξής μήνυμα σφάλματος: «*CryptoPass: You already possess a SBT and you may only have one*».

Ο **onlyAuthAcc**, απλώς χρησιμοποιεί την διεύθυνση του αποστολέα της συναλλαγής για να δει εάν υπάρχει στην χαρτογράφηση *_authPersonal*. Εάν υπάρχει, επιτρέπει στον υπόλοιπό κώδικα να εκτελεστεί αλλιώς ακυρώνει την συναλλαγή και στέλνει το εξής μήνυμα σφάλματος: «*CryptoPass: You are NOT Authorized to create an SBT*».

Πριν προχωρήσουμε στο έξυπνο συμβόλαιο *AccessToken*, αξίζει να δούμε τις τρεις πιο βασικές συναρτήσεις του CryptoPass.

Αρχικά, έχουμε την ***_beforeTokenTransfer*** (Εικόνα 7). Η συγκεκριμένη έχει κληρονομηθεί από το συμβόλαιο ERC721.sol των OpenZeppelin.

```
// This restricts the ability to transfer an SBT
// It is an internal function that is called by the ERC721 and here we override it
function _beforeTokenTransfer(
    address from,
    address to,
    uint256 tokenId,
    uint256 batchSize
) internal override(ERC721) {
    require(
        from == address(0),
        "CryptoPass: This is an SBT Token, Its not transferable."
    );
    super._beforeTokenTransfer(from, to, tokenId, batchSize);
}

// This function is used to create an NFT
```

Εικόνα 7: CryptoPass, η τροποποιημένη συνάρτηση _beforeTokenTransfer.

Η συγκεκριμένη συνάρτηση, στο συμβόλαιο των OpenZeppelin, είναι κενή και επίσης διαθέτει τον τροποποιητή *virtual*. Αυτός ο τροποποιητής παρέχεται από την Solidity, και χρησιμοποιείται όταν επιθυμούμε να επιτρέψουμε σε άλλους προγραμματιστές να αλλάξουν την συμπεριφορά μιας συνάρτησης. Το γεγονός ότι η αρχική συνάρτηση είναι κενή σημαίνει ότι αποτελεί μια συνάρτηση εργαλείο (utility function) και ο λόγος ύπαρξης της είναι να επιτρέψει την εισαγωγή επιπλέον λειτουργικότητας.

Στην δική μας περίπτωση, το μόνο που κάνουμε είναι να προσθέσουμε ένα περιορισμό, ο οποίος έμμεσα δεν επιτρέπει την μεταφορά του διακριτού. Αυτό το

καταφέρνουμε απαιτώντας πως η διεύθυνση από την οποία θα παρθεί το διακριτό για μεταφερθεί σε μια άλλη διεύθυνση πρέπει να είναι η μηδενική διεύθυνση. Κανένας δεν έχει πρόσβαση στην μηδενική διεύθυνση, οπότε η συνθήκη `from == address(0)`, δεν μπορεί ποτέ να είναι αληθής.

Τώρα, ας εξετάσουμε την συνάρτηση **safeMint**, απεικονίζεται από κάτω στην Εικόνα 8.

```
// This function is used to create an NFT
// The NFT becomes an SBT because of the resitrictions we have set above
function safeMint(address to) private onlyOne(to) {
    uint256 tokenId = _tokenIdCounter.current();
    _tokenIdCounter.increment();
    _safeMint(to, tokenId);
}
```

Εικόνα 8: CryptoPass, η συνάρτηση safeMint.

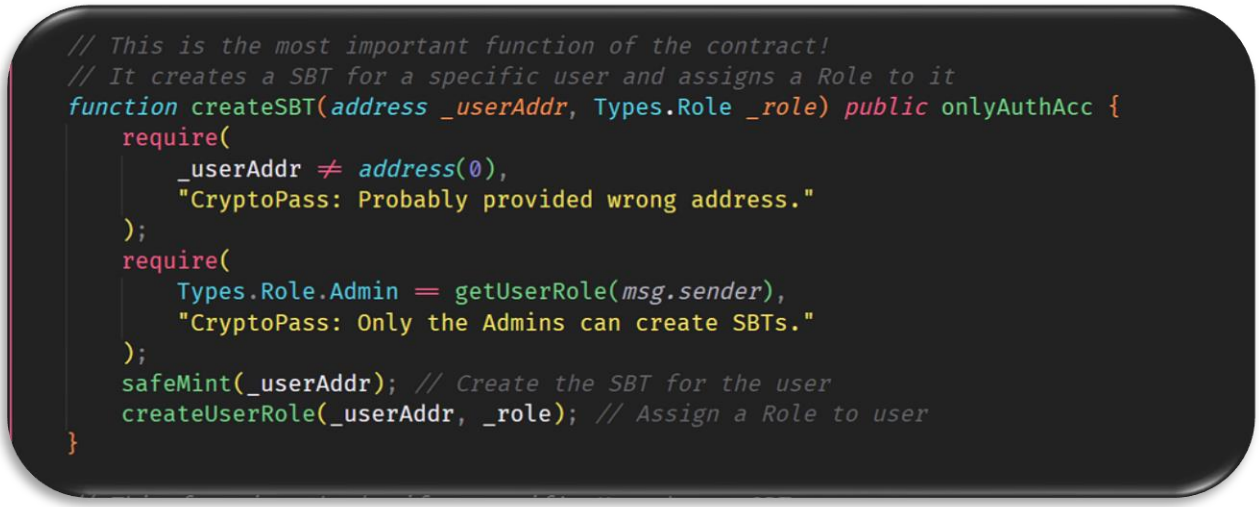
Παρατηρούμε πως διαθέτει τον τροποποιητή *onlyOne*, και του παραχωθεί την παράμετρο της ώστε ο τροποποιητής να ελέγξει εάν η συγκεκριμένη διεύθυνση διαθέτει ήδη ένα SBT. Στην περίπτωση που δεν έχει SBT, εκτελείται ο κώδικας μέσα στις αγκύλες. Επίσης, εδώ γίνεται χρήση του μετρητή που δημιουργήσαμε στην αρχή του συμβολαίου.

Αρχικά, αποθηκεύουμε την τωρινή τιμή του μετρητή σε μια τοπική μεταβλητή που την ονομάζουμε *tokenId*. Υστέρα, αυξάνουμε την τιμή του μετρητή, ώστε το επόμενο διακριτό να έχει διαφορετικό *ID*. Τέλος, καλούμε την συνάρτηση *_safeMint* που μας παρέχεται από το συμβόλαιο των OpenZeppelin για να δημιουργήσουμε το SBT.

Η τελευταία συνάρτηση που θα μελετήσουμε είναι και η πιο σημαντική. Η **createSBT** που παρουσιάζεται στην Εικόνα 9. Σε αντίθεση με την *safeMint*, η συγκεκριμένη μπορεί να καλεστεί από οντότητες εκτός του παρόντος συμβολαίου, όπως για παράδειγμα ένα *web server*. Σε τέτοιες περιπτώσεις πρέπει να αφιερώνεται ιδιαίτερη προσοχή στην προσβασιμότητα της συνάρτησης. Γι' αυτόν ακριβώς τον λόγο χρησιμοποιούμε τον τροποποιητή *onlyAuthAcc*, ώστε μόνο εξουσιοδοτημένοι χρήστες να μπορούν να την χρησιμοποιήσουν. Εκτός από τον τροποποιητή, έχει τοποθετηθεί ακόμα ένα μετρό ασφάλειας.

Μόνο χρήστες οι οποίοι διαθέτουν το ρολό "Admin" μπορούν να δημιουργήσουν SBTs. Δηλαδή, για να μπορέσει ένας λογαριασμός να αποκτήσει το δικαίωμα χορήγησης SBTs, θα πρέπει να τοποθετηθεί στην χαρτογράφηση *_authPersonal* και να διαθέτει τον ρολό του *Admin*.

Η υπόλοιπη λειτουργικότητα της συνάρτησης είναι ιδιαίτερα απλή, καθώς δημιουργεί το SBT καλώντας την safeMint παραχωρώντας την διεύθυνση του χρήστη για τον οποίο προορίζεται και τέλος του χορηγεί έναν ρολό, τον οποίο επιλεγεί ο χρήστης που καλεί την συνάρτηση, δηλαδή η γραμματεία.



```
// This is the most important function of the contract!
// It creates a SBT for a specific user and assigns a Role to it
function createSBT(address _userAddr, Types.Role _role) public onlyAuthAcc {
    require(
        _userAddr != address(0),
        "CryptoPass: Probably provided wrong address."
    );
    require(
        Types.Role.Admin == getUserRole(msg.sender),
        "CryptoPass: Only the Admins can create SBTs."
    );
    safeMint(_userAddr); // Create the SBT for the user
    createUserRole(_userAddr, _role); // Assign a Role to user
}
```

Εικόνα 9: CryptoPass, η συνάρτηση createSBT.

9.2 Έξυπνο Συμβόλαιο: AccessToken

9.2.1 Σκοπός και Σχεδίαση

Όπως και το CryptoPass, το έξυπνο συμβόλαιο AccessToken χρησιμοποιεί το πρότυπο ERC-721 των OpenZeppelin για την δημιουργία των διακριτών του. Η αρμοδιότητα του είναι η διαχείριση της έκδοσης προσωρινών διακριτικών πρόσβασης. Αυτά τα διακριτικά αποτελούν αποδεικτικά κατοχής ψηφιακής ταυτότητας με περιορισμένη διάρκεια. Όταν φτάσουν στην διεπαφή του χρήστη (client UI), αυτά τα διακριτά κωδικοποιούνται και μετατρέπονται σε QR κωδικούς με σκοπό να μπορούν ευκολά να επαληθευτούν από φυσικές οντότητες, ανθρώπινες ή μη.

9.2.2 Κύριες Λειτουργίες

- **Έκδοση Διακριτικών και Λήξη**: Το συμβόλαιο δημιουργεί διακριτικά πρόσβασης με έναν προκαθορισμένο αριθμό μπλοκ λήξης, μετά τον οποίο το διακριτικό δεν μπορεί πλέον να χρησιμοποιηθεί, αυξάνοντας την ασφάλεια με τον περιορισμό της περιόδου πρόσβασης. Εάν το διακριτό δεν χρησιμοποιηθεί αυτόματα καίγεται (διαγράφεται). Το ίδιο συμβαίνει και όταν το διακριτό χρησιμοποιηθεί από το χρήστη. Τέλος, όταν δημιουργείται ένα διακριτό, πυροδοτείτε ένα γεγονός (event), που μπορείς να χρησιμοποιηθεί για να ανακτηθούν πληροφορίες για τον συγκεκριμένο διακριτό ή για την εκτέλεση άλλων ενεργειών ως παρενέργειες σε αυτό το γεγονός.

- **Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους**: Το AccessToken επικοινωνεί με το συμβόλαιο CryptoPass πριν δημιουργήσει το προσωρινό διακριτό, για να επιβεβαίωση την ύπαρξη του SBT και επειδή χρειάζεται να λάβει τον ρολό του εκάστοτέ χρήστη για να τον ενσωματώσει στα δεδομένα που θα επιστέψει στον web server. Χρησιμοποιήθηκε αυτή η προσέγγιση καθώς διαφορετικά ο web server θα έπρεπε να καλέσει και τα δυο έξυπνα συμβόλαια και θα αυτό αύξανε την περιπλοκότητα κατά τον σχεδιασμό της εφαρμογής.

- **Ενσωμάτωση QR Κώδικα**: Τα διακριτικά πρόσβασης έχουν σχεδιαστεί ώστε να μετατρέπονται ευκολά σε κωδικούς QR από την εφαρμογή του χρήστη (frontend), παρέχοντας μια γέφυρα μεταξύ των ψηφιακών εξουσιοδοτήσεων και του φυσικού ελέγχου πρόσβασης.

9.2.3 Ανάλυση Κώδικα

Καθώς το *AccessToken* διαθέτει αρκετές ομοιότητες με το *CryptoPass* θα επικεντρωθούμε μονάχα στα κύρια χαρακτηριστικά του. Ας ξεκινήσουμε με τις παγκόσμιες μεταβλητές που βρίσκονται στην αρχή του αρχείο, όπως απεικονίζεται στην Εικόνα 10.

```
// Assuming CryptoPass contract has an interface like this:
interface ICryptoPass {
    enum Role {
        None,
        Student,
        Professor,
        Secretary,
        Admin
    }

    function getUserRole(address user) external view returns (Role);

    function hasSBT() external view returns (bool result);
}

// This "IERC721Receiver", allows the contract to receive the SoulBound NFT
contract AccessToken is ERC721, Ownable, IERC721Receiver {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIdCounter;

    struct TokenData {
        uint256 id; // The Token's ID
        ICryptoPass.Role role; // This is an Enum, see Types.sol
        uint256 exp; // expiration block number
    }

    mapping(uint256 => TokenData) private _tokenDetails;
    mapping(address => uint256) private _ownerToTokenId;

    ICryptoPass public cryptoPassContract;

    // Number of blocks roughly equivalent to 5 minutes depends on the block time. For Ethereum's ~15 second block time,
    it's around 20.
    uint256 public expirationBlocks = 20;

    event TokenMinted(uint256 tokenId, ICryptoPass.Role role, uint256 exp);
```

Εικόνα 10: AccessToken, παγκόσμιες μεταβλητές.

Αρχικά, παρατηρούμε την ύπαρξη μιας διεπαφή (interface) σκοπός της οποίας είναι να παρέχει στο παρόν συμβόλαιο τις απαραίτητες πληροφορίες που χρειάζεται για μπορεί να καλέσει τις αναφερόμενες συναρτήσεις από το CryptoPass.

Έπειτα, έχουμε την έναρξή του συμβολαίου. Ενδιαφέρον αποτελεί η χρήση της βιβλιοθήκης **IERC721Receiver** που επιτρέπει στο έξυπνο συμβόλαιο να δέχεται διακριτά που ακολουθούν το πρότυπο των NFTs.

Υστέρα έχουμε την δομή δεδομένων (struct) **TokenData**. Κατά μια έννοια ένα *struct* είναι σαν μια κλάση οπού όμως μπορούμε να έχουμε μόνο δεδομένα και όχι μεθόδους.

Ουσιαστικά, στην Solidity χρησιμοποιούμε τη λέξη-κλειδί `struct` για να δημιουργήσουμε ένα δικό μας τύπο δεδομένων. Ο συγκεκριμένος τύπος αποτελείται από τα στοιχεία.

- **id**: Το ID χρησιμοποιείται για μπορούμε να ξεχωρίζουμε τα διάφορα tokens.
- **role**: Αποτελεί τον ρολό που περνούμε από το CryptoPass και ο τύπος του πρέπει να είναι συγκεκριμένος, δηλαδή να ανήκει στο απαριθμημένο τύπο (enum) *Role*.
- **exp**: Είναι απλώς ένας μη-αρνητικός ακέραιος αριθμός μεγέθους 32-bytes (uint256) που υποδηλώνει μέχρι ποιο block θα ισχύει το token.

Στην συνέχεια έχουμε δυο χαρτογραφήσεις, την *_tokenDetails* και την *_ownerToTokenId*.

Η πρώτη, αντιστοιχίζει το id του κάθε token με τον τύπο δεδομένων που αναφέραμε πιο πάνω, δηλαδή τα δεδομένα του token. Με αυτόν το τρόπο, εάν γνωρίζουμε το ID του token, μπορούμε να ανακτήσουμε όλες του τις πληροφορίες του.

Η δεύτερη, αντιστοιχίζει την δημοσιά διεύθυνση των κάτοχων των tokens με το id αυτών των tokens. Με αυτόν τον τρόπο είναι πολύ εύκολο να αντλήσουμε τις πληροφορίες ενός token για έναν συγκεκριμένο χρήστη, καθώς πλέον χρειαζόμαστε μονάχα την διεύθυνση του. Ο λόγος που δεν αντιστοιχούμε κατευθείαν τα δεδομένα των tokens με τις διευθύνσεις των κατόχων τους, έγκειται στο γεγονός ότι υπάρχουν συναρτήσεις δεν έχουν πρόσβαση στην διεύθυνση του χρήστη αλλά πρέπει να αντλήσουν πληροφορίες για τα access tokens.

Περαιτέρω, έχουμε άλλες δυο παγκόσμιες μεταβλητές, την *cryptoPassContract* και την *expirationBlocks*. Η *cryptoPassContract* είναι απαραίτητη για να πραγματοποιηθεί η σύνδεση αναμεσά στα δυο έξυπνα συμβόλαια. Χρησιμοποιεί ως τύπο την διεπαφή (interface) που δημιουργήσαμε έξω από το συμβόλαιο. Μέσα στον κατασκευαστή (constructor) θα αναθέσουμε μια τιμή σε αυτήν την μεταβλητή, οπότε θα την μελετήσουμε αργότερα.

Η *expirationBlocks* δεν είναι πάρα μια σταθερά που δηλώνει μετά από ποσό χρόνο θα παύσει να ισχύει ένα token. Ο αριθμός 20 επιλέχθηκε με τον συμβιβασμό ότι πρέπει να περάσουν περίπου δεκαπέντε δευτερόλεπτα για να παραχθεί ένα νέο block. Οπότε ο αριθμός 20 αντιστοιχεί περίπου σε πέντε λεπτά.

Πριν προχωρήσουμε στην ανάλυση του υπολοίπου συμβολαίου, αξίζει να αναφέρουμε το γεγονός (event) *TokenMinted*. Αυτό το γεγονός πυροδοτείτε όταν οποιοσδήποτε χρήστης κάνει αίτηση για ένα προσωρινό access token. Στην συγκεκριμένη εφαρμογή δεν υπάρχει κώδικας που να εκμεταλλεύεται αυτήν την λειτουργικότητα αλλά τοποθετήθηκε καθώς θεωρείται καλή πρακτική (best practice).

Συνεχίζοντας στο υπόλοιπό συμβόλαιο, θα μελετήσουμε τον κατασκευαστή, και τις δυο πιο σημαντικότερές συναρτήσεις.

Ο **κατασκευαστής του AccessToken** (Εικόνα 11) είναι ιδιαίτερα απλώς αλλά ταυτόχρονα ενδιαφέρον, καθώς δέχεται μια διεύθυνση ως παράμετρο και υστέρα την παραχωρεί σε μια συνάρτηση οπού έχει το ίδιο ακριβώς όνομα με την διεπαφή που δημιουργήσαμε έξω από το συμβόλαιο.

```
constructor(
    address _cryptoPassAddress
) ERC721("CryptoPassAccessToken", "CPATK") {
    cryptoPassContract = ICryptoPass(_cryptoPassAddress); // Connecting t
    first!
}
```

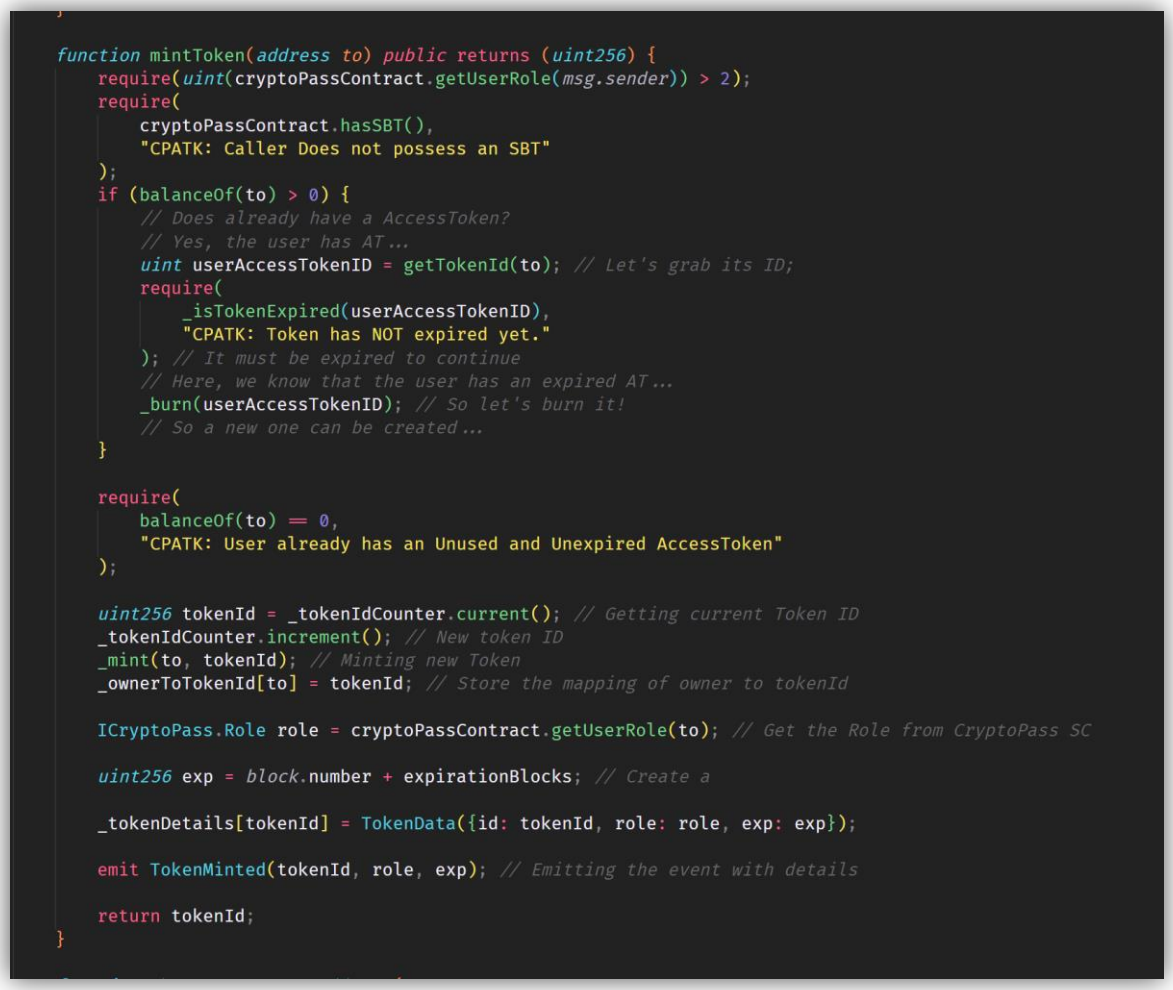
Εικόνα 11: AccessToken, κατασκευαστής (constructor) συμβολαίου.

Όταν κατασκευάζουμε μια διεπαφή, η Solidity αυτόματα δημιουργεί μια συνάρτηση που δέχεται ως επιχείρημα (argument) μια διεύθυνση. Αυτή η συνάρτηση λοιπόν επιστρέφει ένα αντικείμενο το οποίο μπορούμε να χρησιμοποιήσουμε για να καλέσουμε τις συναρτήσεις ενός αλλού έξυπνού συμβολαίου.

Ο λόγος για τον οποίο πρέπει να πάρουμε αυτήν την διεύθυνση από την παράμετρο του κατασκευαστή είναι επειδή για να αποκτήσουμε την διεύθυνση του CryptoPass, πρέπει πρώτα το CryptoPass να προωθηθεί (να γίνει deploy) στο δίκτυο blockchain. Θα δούμε πιο αναλυτικά πως δουλεύει αυτό όταν θα μελετήσουμε τον κώδικα JavaScript που ευθύνη του είναι να προωθήσει τα έξυπνα συμβόλαια στο δίκτυο blockchain.

Ας εξετάσουμε τώρα τις δυο πιο σημαντικές συναρτήσεις του συμβολαίου, την *mintToken* και την *useToken*.

Ξεκινώντας από την πρώτη (**mintToken**), που είναι και η πιο περιπλοκή και φαίνεται στην Εικόνα 12, παρατηρούμε τις δυο δηλώσεις (statements) `require`. Η πρώτη ελέγχει εάν ο αποστολέας της συναλλαγής διαθέτει τουλάχιστον ρολό «*Professor*» (αυτό υποδηλώνει το νούμερο 2). Η δεύτερη ελέγχει εάν διαθέτει SBT, πρακτικά όλες οι συναλλαγές θα γίνονται από τον web server καθώς η διεπαφή χρήστη είναι σχεδιασμένη με τέτοιο τρόπο και επίσης είναι ο μόνος που διαθέτει ETH, αλλά το συγκεκριμένο `require` τοποθετήθηκε σαν μια εξτρά δικλείδα ασφάλειας.



```
function mintToken(address to) public returns (uint256) {
    require(uint(cryptoPassContract.getUserRole(msg.sender)) > 2);
    require(
        cryptoPassContract.hasSBT(),
        "CPATK: Caller Does not possess an SBT"
    );
    if (balanceOf(to) > 0) {
        // Does already have a AccessToken?
        // Yes, the user has AT...
        uint userAccessTokenID = getTokenId(to); // Let's grab its ID;
        require(
            _isTokenExpired(userAccessTokenID),
            "CPATK: Token has NOT expired yet."
        ); // It must be expired to continue
        // Here, we know that the user has an expired AT...
        _burn(userAccessTokenID); // So let's burn it!
        // So a new one can be created...
    }

    require(
        balanceOf(to) == 0,
        "CPATK: User already has an Unused and Unexpired AccessToken"
    );

    uint256 tokenId = _tokenIdCounter.current(); // Getting current Token ID
    _tokenIdCounter.increment(); // New token ID
    _mint(to, tokenId); // Minting new Token
    _ownerToTokenId[to] = tokenId; // Store the mapping of owner to tokenId

    ICryptoPass.Role role = cryptoPassContract.getUserRole(to); // Get the Role from CryptoPass SC

    uint256 exp = block.number + expirationBlocks; // Create a

    _tokenDetails[tokenId] = TokenData({id: tokenId, role: role, exp: exp});

    emit TokenMinted(tokenId, role, exp); // Emitting the event with details

    return tokenId;
}
```

Εικόνα 12: AccessToken, η συνάρτηση mintToken.

Έπειτα ακολουθούν και άλλοι έλεγχοι. Ο πρώτος βλέπει εάν το υπόλοιπό, σε access tokens, του λογαριασμού που θα του χορηγηθεί το access token είναι μηδέν. Εάν δεν είναι, δηλαδή έχει στην κατοχή του ένα access token, χρησιμοποιούμε την διεύθυνση για να αντλήσουμε τα δεδομένα αυτού το token. Έπειτα, ελέγχουμε εάν έχει λήξει. Στην περίπτωση που δεν έχει λήξει, ακυρώνουμε την συναλλαγή και στέλνουμε το εξής μήνυμα: «*CPATK: Token has NOT expired yet.*». Όμως στην περίπτωση που έχει λήξει, καίμε (διαγράφουμε) το συγκεκριμένο access token. Αυτό αλλάζει το υπόλοιπό του χρήστη από ένα σε μηδέν. Στην συνέχεια, υπάρχει ένας τελευταίος έλεγχος. Ο συγκεκριμένος απαιτεί το υπόλοιπό του χρήστη να είναι ίσο με μηδέν.

Συνοπτικά, ο σκοπός αυτών των δηλώσεων `require` είναι να επιβάλουν την κατοχή μόνο ενός access token την φορά και στην περίπτωση που κάποιο έχει λήξει, την διαγραφή του παλιό και την χορήγηση νέου. Ο υπόλοιπος κώδικας της συνάρτησης χρησιμοποιεί το κλασσικό μοτίβο για την παραγωγή ενός διακριτού, έπειτα αποθηκεύει το id του στην χαρτογράφηση *_ownerToTokenId,* υπολογίζει σε μια μεταβλητή τον αριθμό του block στο οποίο θα λήξει το access token, αποθηκεύει τα δεδομένα του νέου token στην χαρτογράφηση *_tokenDetails* και τέλος πυροδοτεί το γεγονός *TokenMinted*.

Ας μελετήσουμε τώρα την συνάρτηση **useToken**, που εμφανίζεται στην Εικόνα 13.

```
function useToken(uint256 tokenId) external {
    // require(ownerOf(tokenId) == msg.sender, "Not the owner of this token");
    require(
        uint(cryptoPassContract.getUserRole(msg.sender)) > 1,
        "CPATK: Insufficient Access Level"
    ); // Restricting Access
    require(!_isTokenExpired(tokenId), "CPATK: Token has expired");
    require(
        balanceOf(ownerOf(tokenId)) == 1,
        "CPATK: User does not possess an AccessToken"
    );
    address tokenOwner = ownerOf(tokenId);
    _burn(tokenId); // Token can only be used once
    delete _ownerToTokenId[tokenOwner]; // Clear the mapping
}
```

Εικόνα 13: AccessToken, η συνάρτηση useToken.

Σε σχέση με την *mintToken*, η συγκεκριμένη είναι αρκετά απλούστερη. Συνοπτικά, πραγματοποιούνται τρεις έλεγχοι πριν το access token χρησιμοποιηθεί.

Αρχικά, ο αποστολέας της συναλλαγής πρέπει να έχει ρολό τουλάχιστον ίσο με «Student» (αυτό υποδηλώνει το 1). Στην συνέχεια, το access token πρέπει να μην έχει λήξει. Ο τελευταίος έλεγχος εξετάζει εάν ο χρήστης έχει ένα ακριβώς access token, καθώς εάν έχει 0 ή περισσότερα από ένα υπάρχει πρόβλημα.

Στην περίπτωση που δεν ακυρωθεί η συναλλαγή, καταλήγουμε στο συμπέρασμα ότι ο χρήστης διαθέτει ένα έγκυρο access token και επόμενος μπορούμε με ασφάλεια να το διαγράψουμε. Ο web server είναι προγραμματισμένος να θεωρήσει επιτυχής την χρήση ενός access token εάν δεν παραχθεί κάποιο σφάλμα κατά την εκτελέσει της συναλλαγής.

Συμπεράσματα

Τα έξυπνα συμβόλαια CryptoPass και AccessToken είναι κρίσιμα για τη λειτουργικότητα της εφαρμογής, παρέχοντας ένα ασφαλές και αποκεντρωμένο πλαίσιο για την επαλήθευση της ψηφιακής ταυτότητας. Η ανάπτυξή τους στο δίκτυο Ethereum διασφαλίζει τη διαφάνεια, την ασφάλεια και την αμεταβλητότητα, ενώ οι προσεκτικά κατασκευασμένες λειτουργίες και τροποποιητές τους εξασφαλίζουν ότι μόνο εξουσιοδοτημένες και προβλεπόμενες ενέργειες μπορούν να εκτελεστούν, αντανακλώντας μια ισχυρή τήρηση των αρχών της τεχνολογίας blockchain και της ανάπτυξης έξυπνων συμβολαίων.

Κεφάλαιο 10: Περιβάλλον Ανάπτυξης Έξυπνων Συμβολαίων

Εισαγωγή

Ανεξάρτητα το είδος της εφαρμογής, το περιβάλλον ανάπτυξης (development environment) αποτελεί υψίστης σημασίας σε όλον τον κύκλο ζωής μιας εφαρμογής. Στην περίπτωση του blockchain και της ανάπτυξης έξυπνων συμβολαίων, το περιβάλλον ανάπτυξης παρέχει τα απαραίτητα εργαλεία για την δημιουργία, δοκιμή και ανάπτυξη ενός ή πολλαπλών έξυπνων συμβολαίων. Αυτή η ενότητα εξετάζει τις λεπτομέρειες του περιβάλλοντος **Hardhat**, το οποίο χρησιμοποιείται για την κατασκευή, ανάπτυξη, δοκιμή και προώθηση των έξυπνων συμβολαίων CryptoPass και AccessToken.

10.1 Κύρια Μέρη του Περιβάλλοντος

10.1.1 Το Σενάριο Προώθησης (myDeploys.js):

- **Σκοπός**: Η προώθηση των δυο έξυπνων συμβολαίων στο δίκτυο blockchain που έχει τεθεί στο αρχείο *hardhat.config.js*. Ακόμα, χρησιμοποιεί διάφορες βοηθητικές συναρτήσεις και σενάρια (scripts) για να αρχικοποιήσει την κατάσταση αυτών των συμβολαίων.

- **Λειτουργικότητα**: Το συγκεκριμένο σενάριο πραγματοποιεί *τέσσερεις βασικές λειτουργίες*:
  - Προωθεί τα δυο έξυπνα συμβόλαια ξεκινώντας από το CryptoPass. Μόλις προωθηθεί το CryptoPass, προωθεί το AccessToken παραχωρώντας του την διεύθυνση του CryptoPass.
  - Αποθηκεύει σε ένα αρχείο json τις πληροφορίες των δυο συμβολαίων με σκοπό η υπόλοιπη εφαρμογή να έχει δυναμική πρόσβαση σε αυτές. Αυτό βοηθάει ιδιαίτερα κατά την ανάπτυξη και δοκιμή της εφαρμογής.
  - Παραχωρεί δικαιώματα χορήγησης SBT και κατασκευάζει ένα SBT για το κρυπτοπορτοφόλι του web server αλλά και για το συμβόλαιο AccessToken.
  - Στέλνει 250 ETH στο κρυπτοπορτοφόλι του web server, ώστε ο web server να μπορεί να πραγματοποιήσει συναλλαγές στο δίκτυο blockchain.

- **Αλληλεπίδραση**: Αλληλοεπιδρά με το περιβάλλον εκτέλεσης Hardhat (hre) για την πρόσβαση στο δίκτυο και τα δεδομένα των λογαριασμών που χρησιμοποιούνται κατά την προώθηση των συμβολαίων.

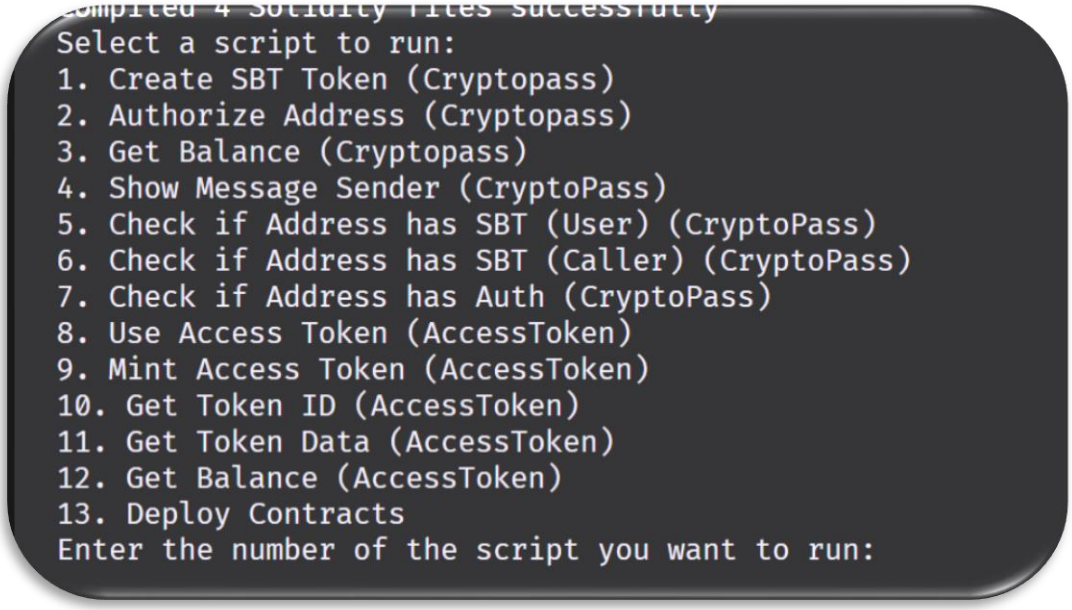
10.1.2 Ο Διαχειριστής Σεναρίων (scriptRunner.js):

- **Σκοπός**: Να παρέχει μια απλή διεπαφή κονσόλας (CLI) οπού ο προγραμματιστής να μπορεί με ευκολία να εκτελεί διαφορά σενάρια που καλούν συναρτήσεις από τα έξυπνα συμβόλαια.

- **Λειτουργικότητα**: Για την εκτέλεση του *scriptRunner*, πρέπει να χρησιμοποιηθεί η εντολή:

  npx hardhat run --network localhost scripts/actions/scriptRunner.ts

Έπειτα, θα εμφανιστούν οι ακόλουθες επιλογές στην κονσόλα του χρήστη (Εικόνα 14):



Εικόνα 14: Hardhat, οι διαθέσιμες επιλογές που παρέχονται από την διεπαφή κονσόλας του *scriptRunner*.

Όπως μπορείτε να παρατηρήσετε από την εικόνα, το scriptRunner μας παρέχει μια διεπαφή για να καλέσουμε τις συναρτήσεις των προωθημένων συμβολαίων. Ακόμα, αναφέρει σε ποιο συμβόλαιο αντιστοιχεί το κάθε σενάριο. Η τελευταία επιλογή (13) εκτελεί το σενάριο που αναφέραμε πιο πάνω για την προώθηση των έξυπνων συμβολαίων στο blockchain.

10.1.3 Το αρχείο ρυθμίσεων Hardhat (hardhat.config.js):

- **Σκοπός**: Λειτουργεί ως το κεντρικό αρχείο ρυθμίσεων για το περιβάλλον Hardhat.

- **Ρυθμιζόμενα Στοιχεία**: Περιλαμβάνει τις ρυθμίσεις δικτύου, ρυθμίσεις μεταγλωττιστή (compiler), προσθήκες πρόσθετων (plugins) και ορισμούς προσαρμοσμένων εργασιών (tasks). Καθώς η παρούσα εφαρμογή είναι ιδιαίτερα απλή, οι περισσότερες προεπιλεγμένες ρυθμίσεις (defaults) του Hardhat καλύπτουν τις ανάγκες της εφαρμογής. Αυτό έχει ως αποτέλεσμα το αρχείο ρυθμίσεων να είναι ιδιαίτερα μικρό, όπως απεικονίζεται στην Εικόνα 15 παρακάτω.

Εικόνα 15: Hardhat, το αρχείο ρυθμίσεων (*hardhat.config.js*)

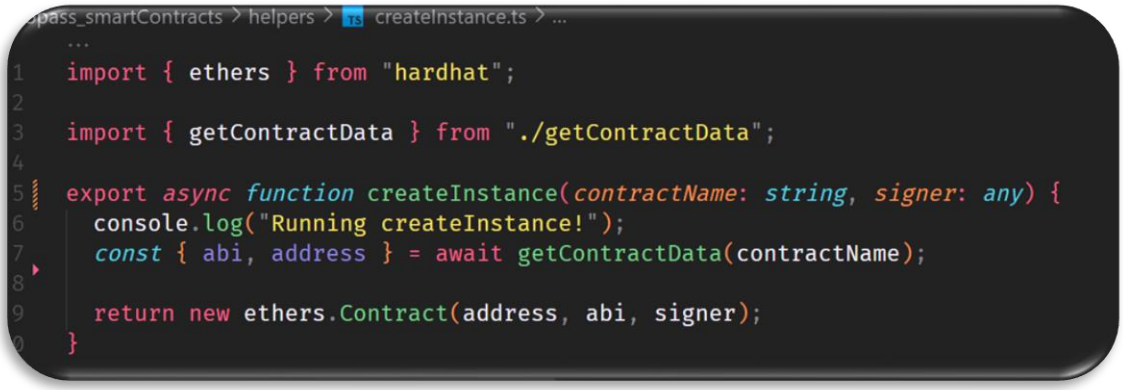
Σε μορφή σχολίων (comments) φαίνονται οι βασικές ρυθμίσεις που απαιτούνται εάν επιθυμούσαμε να προωθήσουμε τα συμβόλαια της εφαρμογής σε κάποιο άλλο δίκτυο και εάν ο προωθητής (deployer) πρέπει να είναι κάποιος συγκεκριμένος λογαριασμός.

- **Επεκτασιμότητα**: Το αρχείο ρυθμίσεων υποστηρίζει την ένταξη πρόσθετων και επιπλέον εργαλείων για την ενίσχυση της ροής εργασίας.

10.1.4 Βοηθητικές Συναρτήσεις (helper functions)

Εκτός από τα κύρια σενάρια και τις ρυθμίσεις, υπάρχει ένας φάκελος, ονόματι *helpers*, που περιέχει χρήσιμες συναρτήσεις που απλοποιούν τη διαδικασία ανάπτυξης:

- **createInstance (**Εικόνα 16**)**: Δημιουργεί αντικείμενο το οποίο είναι βασισμένο σε ένα έξυπνο συμβόλαιο που έχει ήδη προωθηθεί στο δίκτυο blockchain. Απαιτεί ως είσοδο (input) δυο επιχειρήματα, το όνομα του έξυπνου συμβολαίου και έναν λογαριασμό. Το παραγόμενο αντικείμενο μπορεί να χρησιμοποιηθεί για την κλήση συναρτήσεων από το προωθημένο έξυπνο συμβόλαιο.



Εικόνα 16: Hardhat, η βοηθητική συνάρτηση *createInstanse*

- **deployContract (**Εικόνα 17**)**: Αυτοματοποιεί την ανάπτυξη έξυπνων συμβολαίων, αφαιρώντας τα επαναλαμβανόμενα τμήματα κωδικοποίησης, αυξάνοντας έτσι την αποδοτικότητα και την οργάνωση του κώδικα.

Εικόνα 17: Hardhat, η βοηθητική συνάρτηση *deployContract*

- **storeContractData**: Αποθηκεύει τις πληροφορίες του συμβολαίου, όπως τη διεύθυνσή του και το Application Binary Interface (ABI), σε ένα αρχείο json, διευκολύνοντας την εύκολη πρόσβαση για εφαρμογές frontend ή για σκοπούς επαλήθευσης.

- **getContractData (**Εικόνα 18**)**: Ανακτά τις αποθηκευμένες πληροφορίες του συμβολαίου από ένα αρχείο json, οι οποίες είναι κρίσιμες για σενάρια ή εφαρμογές που χρειάζεται να αλληλεπιδράσουν με το αναπτυγμένο συμβόλαιο.

```
export const getContractData = async (contractName: string): Promise<any> => {
  console.log("Running getContractData!");
  const allContractData: ContractData = await import("../contractData.json");
  return findNestedObj(allContractData, contractName.toLowerCase());
};

function findNestedObj<T extends Record<string, any>>(
  obj: T,
  keyToFind: string
): any {
  if (obj.hasOwnProperty(keyToFind)) {
    return obj[keyToFind];
  }

  for (let key in obj) {
    if (obj[key] && typeof obj[key] === "object") {
      let result = findNestedObj(obj[key], keyToFind);
      if (result) {
        return result;
      }
    }
  }
  return null;
}
```

Εικόνα 18: Hardhat, η βοηθητική συνάρτηση *getContractData*

Συμπεράσματα

Το παρόν περιβάλλον ανάπτυξης Hardhat, ενισχυμένο με προσαρμοσμένα σενάρια και βοηθητικές λειτουργίες, παρέχει μια στιβαρή βάση για την ανάπτυξη έξυπνων συμβολαίων. Αυτό το περιβάλλον ενδυναμώνει τους προγραμματιστές με τα απαραίτητα εργαλεία για αποτελεσματική ανάπτυξη και διαχείριση συμβολαίων, διασφαλίζοντας μια ομαλή ροή εργασίας από την ανάπτυξη έως την παραγωγή. Τα παρεχόμενα εργαλεία και σενάρια είναι απαραίτητα για την ικανότητα της εφαρμογής να αναπτύσσει και να διαχειρίζεται με επιτυχία τα έξυπνα συμβόλαια CryptoPass και AccessToken εντός του οικοσυστήματος του Ethereum.

Κεφάλαιο 11: Αρχιτεκτονική του Web Server

Εισαγωγή

Η παρούσα ενότητα περιλαμβάνει την περιγραφή της αρχιτεκτονικής του web server και των συναφών λειτουργικών χειριστών (handlers). Γίνεται χρήση του web framework Hono, έναν ελαφρύ και ευέλικτο αντικαταστάτη του Express.js. Ο web server αποτελείται από τρία κυριά μέρη.

11.1 Αρχικό Σημείο Εισόδου (Entry point)

Η εφαρμογή του web server ξεκινάει από το αρχείο *index.js*, που βρίσκεται στον φάκελο src. Παρακάτω αναφέρονται τα κυριά χαρακτηριστικά του αρχείου:

1. **Εισαγωγή Απαραίτητων Βιβλιοθηκών**: Το αρχείο ξεκινάει με την εισαγωγή (import) των απαραίτητων βιβλιοθηκών για τη λειτουργία του web server, όπως η βιβλιοθήκη *Hono* για το routing και τη διαχείριση των αιτημάτων, και το middleware *cors* για την επεξεργασία αιτημάτων από διαφορετικές πηγές [67] (cross-origin).

2. **Ρύθμιση Περιβάλλοντος**: Χρησιμοποιεί τη βιβλιοθήκη *dotenv* για να φορτώσει μεταβλητές περιβάλλοντος από ένα αρχείο .env, προκειμένου να διαχειριστεί διαφορετικές ρυθμίσεις ανάλογα με το αν ο server τρέχει σε περιβάλλον παραγωγής ή δοκιμών.

3. **Προσδιορισμός Διαδρομών**: Ορίζει διάφορες διαδρομές (endpoints) για τον web server, με κάθε μία να αντιστοιχεί σε μια συγκεκριμένη λειτουργία, όπως πιστοποίηση χρήστη μέσω web3 authentication, ανάκτηση ρόλου χρήστη, δημιουργία QR κωδίκων, και άλλα.

4. **Περιβάλλοντα Εργασίας**: Ανάλογα με το αν η μεταβλητή περιβάλλοντος IS_PRODUCTION έχει την τιμή "yes" ή όχι, ο server είτε ορίζει τις διαδρομές για να διαχειρίζεται πραγματικά αιτήματα (παραγωγή) είτε για δοκιμαστικά αιτήματα (testing).

5. **Εκκίνηση του Server**: Τέλος, το αρχείο χρησιμοποιεί τη συνάρτηση serve για να ξεκινήσει τον web server και να ακούσει για αιτήματα στη θύρα 8787. Εμφανίζει επίσης στην κονσόλα τις διευθύνσεις URL για τις δοκιμές των διαφορετικών λειτουργιών που παρέχει.

Συνοπτικά, το αρχείο αυτό ρυθμίζει έναν web server ώστε να είναι έτοιμος να δεχθεί και να διαχειριστεί διάφορα είδη αιτημάτων, ανάλογα με το αν βρίσκεται σε περιβάλλον παραγωγής ή δοκιμών, και προσφέρει διάφορες web λειτουργίες μέσω των ορισμένων διαδρομών.

11.2 Χειριστές Διάδρομων (Route Handlers)

Οι χειριστές είναι ασύγχρονες συναρτήσεις που ανταποκρίνονται σε συγκεκριμένα HTTP αιτήματα [68] (requests) που στέλνονται στον server.

Συνολικά, εκθέτονται προς δημοσιά χρήση έξι διαδρομές (routes) που η κάθε μια είναι συνδεδεμένη με ένα χειριστή. Τα HTTP αιτήματα είναι τύπου POST [69] για όλες τις διαθέσιμες διαδρομές. Οι διαδρομές έχουν ως εξής:

- **createSBT**:
  - **Σκοπός:** Χειρίζεται τη δημιουργία ενός SBT (Soulbound Token) για τον χρήστη.
  - **Δεδομένα Εισόδου:** Διεύθυνση χρήστη και έναν ρολό.
  - **Επιστρέφει:** Μια τιμή για την επιβεβαίωση επιτυχής διεργασίας ή μήνυμα που αιτιολογεί τον λόγο αποτυχίας.

- **getRole**:
  - **Σκοπός:** Ανακτά τον ρόλο του χρήστη από το συμβόλαιο CryptoPass.
  - **Δεδομένα Εισόδου:** Διεύθυνση χρήστη.
  - **Επιστρέφει:** Έναν αριθμό που αντιστοιχεί στο ρολό του χρήστη. Για παράδειγμα, 1 ισούται με Φοιτητής (Student) και 4 με Διαχειριστής (Admin).

- **getTokenData**:
  - **Σκοπός:** Ανακτά τα δεδομένα ενός access token από το συμβόλαιο AccessToken.
  - **Δεδομένα Εισόδου:** Το ID του access token.
  - **Επιστρέφει:** Ένα αντικείμενο που περιλαμβάνει τις πληροφορίες του επιθυμητού token.

- **qrCodeCreator**:
  - **Σκοπός:** Μετατρέπει τα δεδομένα ενός access token σε ένα αντικείμενο JavaScript ώστε ο κώδικας στην διεπαφή του χρήστη να μπορεί να ευκολία να το μετατρέψει σε κωδικό QR. Ο λόγος που επιλέχθηκε αυτή η προσέγγιση είναι επειδή το μέγεθος ενός αντρειεμένου QR είναι πολλές φορές μεγαλύτερο από ένα αντικείμενο JavaScript.
  - **Δεδομένα Εισόδου:** Διεύθυνση χρήστη.
  - **Επιστρέφει:** Αντικείμενο JavaScript που περιλαμβάνει τις πληροφορίες του εκάστοτε access token.

- **qrCodeValidator**:
  - **Σκοπός:** Επικυρώνει τη χρήση ενός access token με βάση έναν QR κώδικα.
  - **Δεδομένα Εισόδου:** Αντικείμενο JavaScript που περιλαμβάνει τις πληροφορίες του access token.
  - **Επιστρέφει:** Μια τιμή για την επιβεβαίωση επιτυχής διεργασίας ή μήνυμα που αιτιολογεί τον λόγο αποτυχίας.

- **web3auth**:
  - **Σκοπός:** Χειρίζεται την πιστοποίηση χρήστη χρησιμοποιώντας το την υπογραφή που παρέχει ο χρήστης. Ουσιαστικά, ελέγχει ένα ο αποστολέας της αίτησης είναι πράγματι ο κάτοχος του κρυπτοπορτοφολιού.
  - **Δεδομένα Εισόδου:** Ένα μήνυμα (το περιεχόμενο είναι αμελητέο), τη διεύθυνση του χρήστη και ένα υπογεγραμμένο μήνυμα με το ιδιωτικό κλειδί του κρυπτοπορτοφολιού.
  - **Επιστρέφει:** Μια τιμή για την επιβεβαίωση επιτυχής διεργασίας ή μήνυμα που αιτιολογεί τον λόγο αποτυχίας.

11.3 Δημιουργία Πορτοφολιού και Ρυθμίσεις Web3

Για να μπορέσει ο web server να επικοινωνήσει με το blockchain και επομένως με τα έξυπνα συμβόλαια, πρέπει να χρησιμοποιήσει ένα κρυπτοπορτοφόλι.

Αυτό επιτυγχάνεται με την χρήση της βιβλιοθήκης *ethers.js*. Η οποία παρέχει ένα μεγάλο ευρέως εργαλείων και συναρτήσεων για την επικοινωνία μιας εφαρμογής γραμμένη σε JavaScript με EVM-compatible δίκτυα.

Ο κώδικας που είναι υπεύθυνος για την δημιουργία του κρυπτοπορτοφολιού, την αρχικοποίηση του παρόχου (provider) του blockchain δικτύου και την δημιουργία αντικειμένων που αντιπροσωπεύουν τα έξυπνα συμβόλαια βρίσκεται στο αρχείο *contracts.js* μέσα στον φάκελο src.

Αξίζει να εξετάσουμε τον κώδικα μέσα σε αυτό τον αρχείο για δούμε την δημιουργία και την χρήση ενός κρυπτοπορτοφολιού από ένα web server.

Ως συνηθώς θα ξεκινήσουμε τις απαιτούμενες βιβλιοθήκες και τις παγκόσμιες μεταβλητές όπως φαίνεται στην Εικόνα 19.

```
You, 1 second ago | 1 author (You)
import { ethers } from "ethers";

let provider: ethers.Provider;
let providerInitSuccessful = false;

const hardhat = "http://127.0.0.1:8545/";
const sepolia =
  "https://sepolia.infura.io/v3/" + process.env.INFURA_SEPOLIA_API_KEY;
```

Εικόνα 19: Web Server, απαιτούμενες βιβλιοθήκες και παγκόσμιες μεταβλητές του αρχείο *contracts.js*

Η μονή βιβλιοθήκη που χρειαζόμαστε είναι η *ethers*.

Έπειτα έχουμε δυο παγκόσμιες μεταβλητές, την *provider* και την *providerInitSuccessful*. Την πρώτη απλώς την δηλώνουμε χωρίς να την αρχικοποιούμε, ενώ στην δεύτερη δίνουμε την τιμή false.

Από κάτω τους, έχουμε 2 σταθερές. Το hardhat οπού περιέχει το *URL* στο οποίο ακούει ο πάροχος του τοπικού blockchain δικτύου όταν χρησιμοποιούμε το Hardhat. Ακόμα, έχουμε την μεταβλητή sepolia, οπού και εδώ δίνουμε ένα *URL* παρόχου αλλά αυτήν την φορά προσθέτουμε στο URL ένα κλειδί API. Το process.env.INFURA_SEPOLIA_API_KEY είναι απλώς ένας τρόπος για πάρουμε ευαίσθητα δεδομένα που είναι αποθηκευμένα σε περιβαλλοντικές μεταβλητές. Δεν θα επιμείνουμε στο θέμα καθώς είναι εκτός του πεδίου της παρούσας εργασίας.

Συνεχίζοντας, ερχόμαστε σε επαφή με την ασύγχρονη συνάρτηση *main* (Εικόνα 20). Το πρώτο πράγμα που παρατηρούμε είναι οι δηλώσεις αρκετών τοπικών μεταβλητών.

Εικόνα 20: Web Server, η αρχή της συνάρτησής *main* του αρχείο *contracts.js*

Αρχικά έχουμε το privateKey, που απλώς του δίνουμε την τιμή που έχουμε αποθήκευση στις περιβαλλοντικές μεταβλητές.

Μετά, χρησιμοποιούμε τον κατασκευαστή της κλάσης Wallet, που μας παρέχει η βιβλιοθήκη ethers, για να δημιουργήσουμε ένα κρυπτοπορτοφόλι βάση του ιδιωτικού κλειδιού.

Υστέρα, αποθηκεύουμε την δημοσιά διεύθυνση του κρυπτοπορτοφολιού που μόλις κατασκευάσαμε.

Έπειτα, δημιουργούμε ένα αντικείμενο signer, στην ουσία η διαφορά του signer από το wallet είναι ότι στο αντικείμενο signer έχει ενσωματωθεί ένας πάροχος οπότε έχει την δυνατότητα να γραφεί δεδομένα δημιουργώντας συναλλαγές, αλλά και να διαβάζει δεδομένα από το blockchain.

Στην συνέχεια περνούμε τις διευθύνσεις και τα ABIs των έξυπνων συμβολαίων από το αρχείο contractData.json. Προσοχή, πρέπει να έχουν προωθηθεί τα έξυπνα συμβόλαια προτού ξεκινήσουμε τον web server. Εάν δεν γίνει αυτό το αρχείο *contractData.json* δεν θα υπάρχει και το Hono θα εμφανίσει πρόβλημα.

Στο τελευταίο κομμάτι του κώδικα χρησιμοποιούμε τα δεδομένα από το *contractData.json* για να δημιουργήσουμε τα δυο αντικείμενα που εκπροσωπούν τα έξυπνα συμβόλαια, όπως απεικονίζεται στην Εικόνα 21. Μέσα από αυτά μπορούμε να τα επικοινωνήσουμε με πραγματικά έξυπνα συμβόλαια.

Ας αναλύσουμε το υπόλοιπο μέρος της συνάρτησης main.

```
console.log("The WS Wallet Public Addr: ", walletAddr);
// Fetch the balance
provider.getBalance(walletAddr).then((balnaceInWei: ethers.BigNumberish) => {
  const _balanceEther = ethers.formatEther(balnaceInWei);
  console.log(`Balance of The WS Wallet is: ${_balanceEther} ETH`);
  if (parseFloat(_balanceEther) < 2) {
    console.log("::: YOU MUST SENT ETH → SERVER :::");
  }
});

cryptoPass
  ._authPersonal(walletAddr)
  .then((hasAuth) => {
    if (!hasAuth)
      console.log("[BAD]: WS is NOT Authorized by the Contract! ");
    if (hasAuth) console.log("[GOOD]: WS is Authorized by the Contract! ");
  })
  .catch((error) => {
    if (error.code === "BAD_DATA") {
      console.log(
        "——— AN ERROR WAS THROUGH WHILE TRYING TO SEE IF WS HAS AUTH ———"
      );
      console.error(error);
    } else {
      throw error;
    }
  });
providerInitSuccessful = true; // Set this to true at the end of the function
```

Εικόνα 21: Web Server, η συνέχεια της συνάρτησής *main* του αρχείο *contracts.js*

Εδώ χρησιμοποιούμε τον πάροχο για μάθουμε ποσά ETH διαθέτει το κρυπτοπορτοφόλι του web server. Στο σενάριο (script) *myDeploys.js*, εάν θυμάστε στέλνουμε 250 ETH στον λογαριασμό του web server.

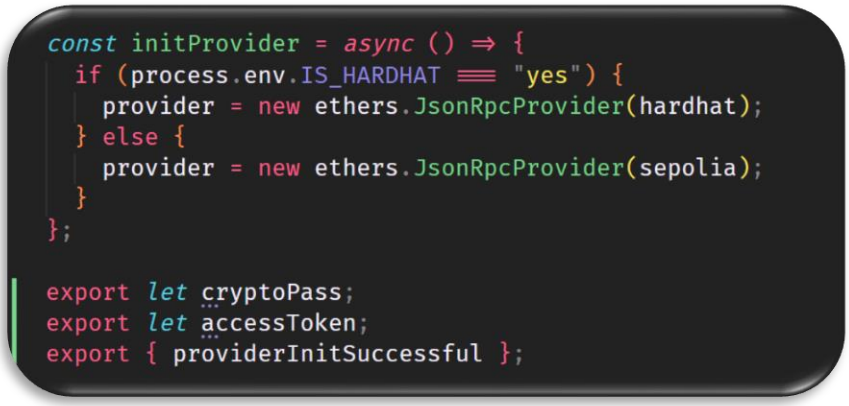
Στην περίπτωση που ο server διαθέτει κάτω από 2 ETH, εμφανίζεται ένα μήνυμα που σκοπός του είναι να υπενθυμίσει στον συντηρητή της εφαρμογής να στείλει μερικά ETH στον server καθώς σε λίγο δεν θα έχει αρκετά για να πραγματοποιεί συναλλαγές.

Τέλος ελέγχουμε, εάν στον web server έχουν δοθεί τα απαραίτητα δικαιώματα για την χορήγηση SBTs.

Συνοπτικά, το δεύτερο μέρος της συνάρτησης *main* αποτελεί έναν έλεγχο ότι όλα έχουν γίνει όπως πρέπει και ο server είναι έτοιμος και ικανός να δεχτεί αιτήματα.

Το τελευταίο αξιόλογο κομμάτι κώδικα αποτελείται από μια συνάρτηση που ελέγχει εάν έχουμε επιλέξει ότι θα χρησιμοποιήσουμε το Hardhat ως το δίκτυο blockchain μας και αναλόγως αρχικοποιεί την παγκόσμια μεταβλητή *provider* (Εικόνα 22).

Τέλος, εξάγουμε (κάνουμε export) το τα δυο αντικείμενα των έξυπνων συμβολαίων καθώς και την μεταβλητή *providerInitSuccessful*, καθώς τα χρησιμοποιούμε στους χείριστες των διάδρομων.



Εικόνα 22: Web Server, το τέλος του κώδικα του αρχείο *contracts.js*

Συμπεράσματα

Στην παρούσα ενότητα, περιγράψαμε τη δομή του web server που χρησιμοποιεί το framework Hono για το περιβάλλον Node.js. Το αρχικό σημείο εισόδου του server, το αρχείο index.js, φορτώνει τις απαραίτητες βιβλιοθήκες, ρυθμίζει μεταβλητές περιβάλλοντος μέσω της dotenv, και ορίζει διάφορες διαδρομές για λειτουργίες όπως πιστοποίηση μέσω web3 και δημιουργία QR κωδίκων. Οι χειριστές ανταποκρίνονται σε HTTP αιτήματα και διαχειρίζονται λειτουργίες όπως η δημιουργία Soulbound Tokens (SBT), η ανάκτηση ρόλων χρήστη και η επαλήθευση QR κωδίκων. Η επικοινωνία με το blockchain επιτυγχάνεται μέσω του παρόχου και των smart contracts που δημιουργούνται και ρυθμίζονται στο αρχείο contracts.js, ενώ ο server εκκινείται με τη χρήση της συνάρτησης serve και ακούει στη θύρα 8787.

Κεφάλαιο 12: Η Διεπαφή Χρήστη (User Interface)

Εισαγωγή

Στην συγκεκριμένη ενότητα θα εξετάσουμε τα κυριά μέρη ενός από τα τέσσερα κομμάτια που απαρτίζουν το συνολική εφαρμογή, τη διεπαφή χρήστη. Η εφαρμογή έχει κατασκευαστεί με React, μια δημοφιλής και ισχυρή βιβλιοθήκη JavaScript για τη δημιουργία διεπαφών χρήστη, μαζί με το Vite, ένα σύγχρονο εργαλείο που βελτιώνει σημαντικά την εμπειρία ανάπτυξης. Για την διευκόλυνση του αναγνωστικού κοινού και καθώς πλέον η χρήση αγγλικών ορών είναι συνήθης σε ερευνητικά και επιστημονικά κείμενα, η συγγραφέας θα αναφέρεται στην διεπαφή χρήση ως frontend ή UI.

12.1 Σκοπός

Η συγκεκριμένη frontend εφαρμογή έχει ως κύριο σκοπό να παρουσιάσει με οπτικό και διαδραστικό τρόπο την λειτουργικότητα των υπολοίπων εξαρτημάτων που απαρτίζουν την συνολική εφαρμογή.

12.2 Ανάλυση

Η εφαρμογή αποτελείται από 3 βασικά σημεία:

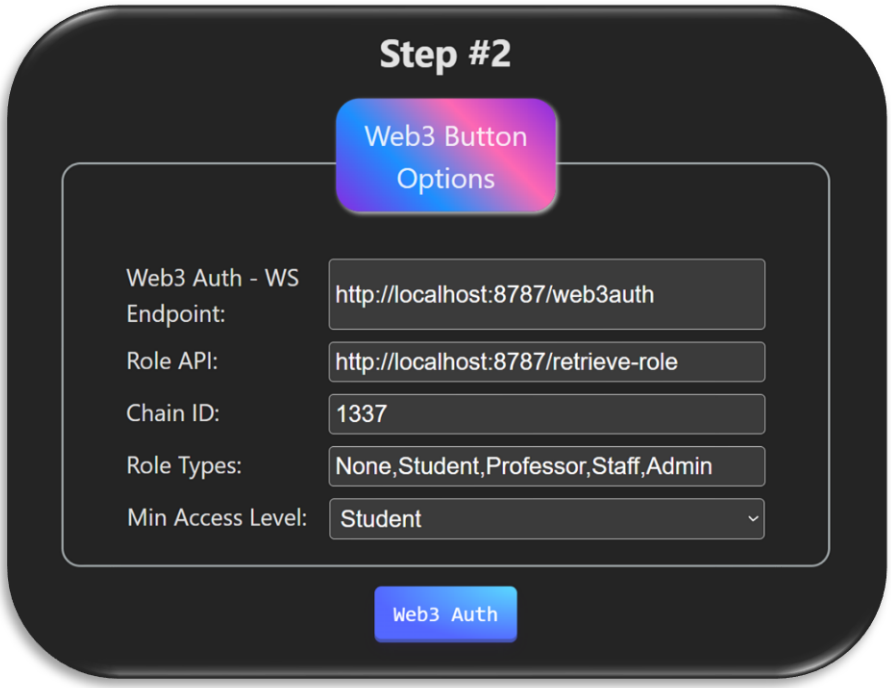
1. **Έλεγχος και κατάσταση υπηρεσιών**: Καθώς ξεκινάει η εκτέλεση του κώδικα στο frontend πραγματοποιούνται διάφοροι έλεγχοι για να επιβεβαιωθεί η ομαλή λειτουργία και η σωστή αρχικοποίηση των υπηρεσιών που απαρτίζουν την συνολική εφαρμογή (Εικόνα 23).

Εικόνα 23: Frontend, Μέρος #1: έλεγχος και κατάσταση υπηρεσιών.

2. **Web3 Button**: Παρέχεται ένα UI (Εικόνα 24) για την επίδειξη των παραμέτρων που χρειάζεται η *βιβλιοθήκη Web3Button* ώστε να μπορέσει να συνδεθεί με το web server, με το δικτυο blockchain και με τα έξυπνα συμβόλαια. Πρέπει να αναφερθεί ότι αλλάζοντας την τελευταια επιλογή (Min Access Level) μπορούμε να περιορίσουμε την πρόσβαση βάση του ρολού που κατέχει ο χρήστης. Πατώντας το

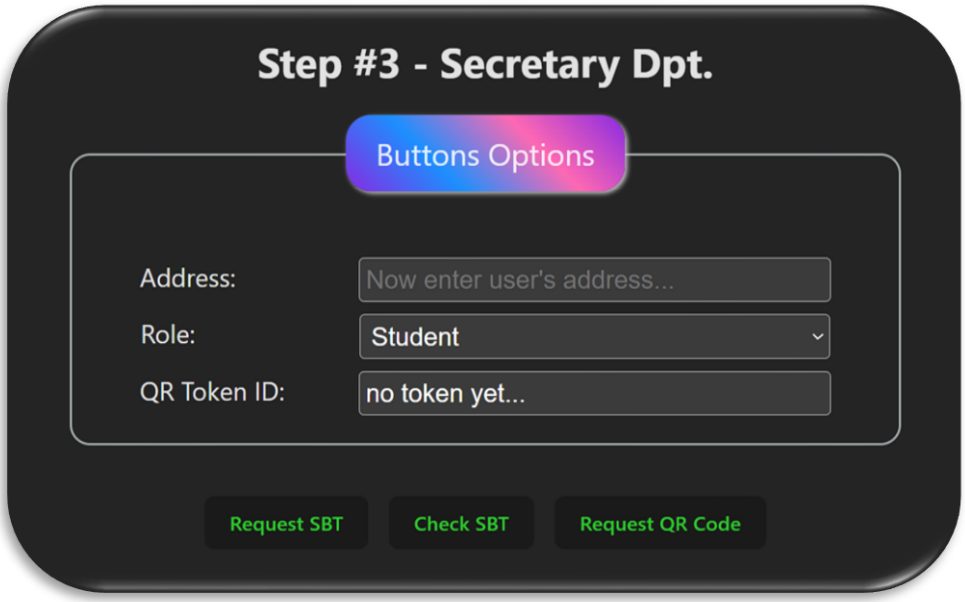
κουμπί *«Web3 Auth»* ξεκινάει η διαδικασία *«log in»* οπού το *Web3Button* θα πραγματοποιήσει τα ακόλουθα βήματα:

a. Θα ζητήσει από τον χρήστη να υπογράψει ένα τυχαίο μνήμα με τον κρυπτοπορτοφόλι του, ώστε να το στείλει το υπογεγραμμένο μήνυμα στο web server μαζί με την διεύθυνση του χρήστη για να γίνει η εξακρίβωση της ιδιοκτησίας του πορτοφολιού.
b. Υστέρα, θα ελέγξει εάν το χρήστης έχει στην κατοχή ένα SBT, εάν δεν έχει εμφανίζει αντίστοιχο μήνυμα στο πάνω κεντρικό μέρος της οθόνης.
c. Εάν έχει, θα ανακτήσει τον ρολό που είναι συνδεδεμένος με το SBT και θα τον συγκρίνει με την παράμετρο *Min Access Level*. Εάν είναι μεγαλύτερος ή ίσος θα επιτρέψει την πρόσβαση αλλιώς θα εμφανίζει αντίστοιχο μήνυμα. Για να επιδείξουμε την επιτυχή πρόσβαση στην ψηφιακή υπηρεσία, το συγκεκριμένο κομμάτι του UI θα πραγματοποιήσει μια κίνηση και το φωτάκι *«Logged In»* θα γίνει πράσινο.



Εικόνα 24: Frontend, Μέρος #2: Βιβλιοθήκη Web3Button.

3. **Διαχείριση SBTs και Access Tokens**: Σε αυτό το σημείο του frontend μπορούμε να εισάγουμε την διεύθυνση ενός λογαριασμού για να πραγματοποιήσουμε διάφορες ενέργειές (Εικόνα 25), αλλά πρώτα πρέπει να έχουμε επιτυχώς συνδεθεί στην εφαρμογή. Για να επιτευχθεί αυτό πρέπει να επιλέξουμε την διεύθυνση που έχουμε αναθέσει στον web server (στην περίπτωση μας, τον 3^{ο} λογαριασμό από αυτούς που μας προσφέρει το Hardhat) και να πιέσουμε το κουμπί *«Web3 Auth»* με το μπλε χρώμα.



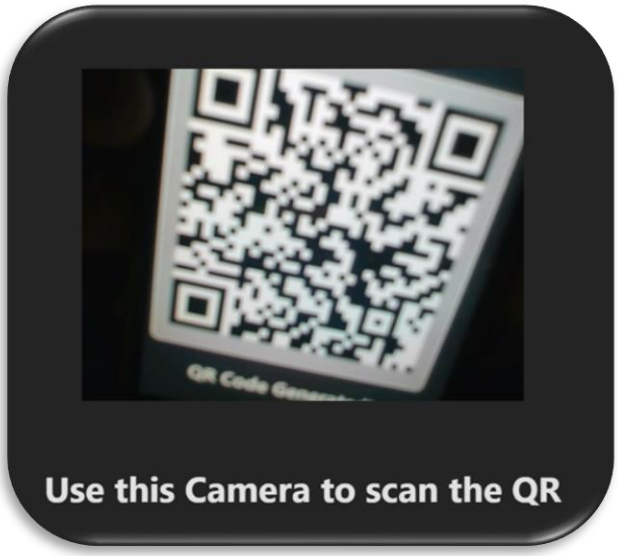
Εικόνα 25: Frontend, Μέρος #3: Διαχείριση SBTs και access tokens.

Μόλις συνδεθούμε, θα παρατηρήσουμε ότι τα γράμματα των κουμπιών στο 3ο σημείο του frontend (Step #3 – Secretary Dpt.) θα αλλάξουν από κοκκίο χρώμα σε πράσινο. Έπειτα μπορούμε να πραγματοποιήσουμε τις εξής ενεργείς:

a. **Request SBT**: Εφόσον έχουμε τοποθετήσει μια διεύθυνση στο πεδίο *Address*, μπορούμε να αναθέσουμε ένα ρολό σε αυτήν την διεύθυνση καθώς δημιουργούμε το SBT πατώντας το κουμπί *«Request SBT»*. Θα εμφανιστεί ένα μήνυμα που θα γραφεί: *«Submitting Transaction...»* καθώς περιμένουμε να ολοκληρωθεί η συναλλαγή. Εάν είναι επιτυχής το περιεχόμενο του μηνύματος θα μετατραπεί σε: *«□ SoulBound Token Successfully Created!»*.
b. **Check SBT**: Πατώντας αυτό το κουμπί, θα καλέσουμε μια συνάρτηση τύπου view από το έξυπνο συμβόλαιο *CryptoPass*. Αυτός ο τύπος υποδηλώνει πως απλώς διαβάζουμε δεδομένα από το blockchain και δεν γράφουμε όπως κάναμε με την προηγουμένη συναλλαγή. Γι' αυτόν τον λόγο η ανταπόκριση είναι εξαιρετικά γρήγορη. Όσον αφορά την λειτουργία του κουμπιού, απλώς παίρνει την διεύθυνση που έχουμε εισάγει στο πεδίο *Address* και επιστρέφει τον ρολό. Στην περίπτωση που βάλουμε μια διεύθυνση που δεν έχει SBT, θα μας επιστρέψει το εξής: *«The User's Role is: [None]»*.
c. **Request QR Code**: Το συγκεκριμένο κουμπί αλληλοεπιδρά, μέσω του web server, με το έξυπνο συμβόλαιο *AccessToken*. Παραχωρεί την διεύθυνση που έχει τεθεί στο πεδίο *Address* και περιμένει να λάβει την ανταπόκριση από τον server. Μόλις την λάβει, την μετατρέπει σε έναν κωδικό QR με την βοήθεια της βιβλιοθήκης *«react-qr-code»*, όπως απεικονίζεται στην Εικόνα 26.

Εικόνα 26: Frontend, Μέρος #3.1: Κατασκευή κωδικού QR.

Ακόμα, ζητά άδεια από τον χρήστη για να ενεργοποιήσει την κάμερα. Η κάμερα του υπολογιστή θα χρησιμοποιηθεί ως scanner. Αυτή η ενέργεια επιλέχθηκε καθώς παρατηρήθηκε πως είναι πολύ ευκολότερο να χρησιμοποιηθεί η κάμερα ενός κινητού τηλεφώνου για την λήψη του κωδικού QR και έπειτα η οθόνη του τηλεφώνου για να σκαναριστεί ο κωδικός από την εφαρμογή (Εικόνα 27). Εάν η διαδικασία είναι επιτυχής, κάτω από κωδικό QR εμφανίζεται ένα νέο κουμπί και με πράσινο χρώμα το εξής μήνυμα: *«QR Code Received Successfully!»*, όπως εμφανίζεται στην Εικόνα 28.



Εικόνα 27: Frontend, Μέρος #3.2: Καθάρισμα κωδικού QR.

Εικόνα 28: Frontend, Μέρος #3.3: Επιτυχής σάρωση κωδικού QR.

Το κουμπί που εμφανίζεται, ονόματι *«Use Access Token»*, θα αποκωδικοποιήσει τον QR κωδικό (που σκαναρίστηκε πριν) και θα χρησιμοποιήσει τα δεδομένα που θα παραχθούν για να προσπαθήσει να χρησιμοποιήσει το access token. Σε μια πραγματική περίπτωση αυτό θα γινόταν κατευθείαν μόλις το scanner αναγνώριζε τον κωδικό QR. Για την χρήση του access token δημιουργείται μια συναλλαγή. Εάν όλα πάνε καλά, ο κωδικός QR καθώς και η κάμερα θα αφαιρεθούν από το UI και θα μείνει μονάχα ένα μήνυμα που θα γραφεί: *«The QR Access Code was USED Successfully!»*

Κεφάλαιο 13: Βιβλιοθήκη Web3Button

Εισαγωγή

Το blockchain, παρότι αποτελεί μια σχετικά πρόσφατη τεχνολογία φέρει σημαντικά πλεονεκτήματα, με τα κυριότερα να είναι η αυξημένη διαφάνεια και η ενισχυμένη ασφάλεια. Ωστόσο, πολλοί προγραμματιστές δεν έχουν ακόμη αποκτήσει εμπειρία στο πώς λειτουργεί, τα οφέλη που προσφέρει, καθώς και τον τρόπο ενσωμάτωσής της στις εφαρμογές τους.

Η συγκεκριμένη βιβλιοθήκη δημιουργήθηκε με σκοπό να καταστήσει την τεχνολογία blockchain πιο προσιτή σε ένα ευρύτερο φάσμα προγραμματιστών, παρέχοντας μία απλοποιημένη διαδικασία για την επικοινωνία μεταξύ της διεπαφής χρήστη (frontend) και των κρυπτοπορτοφολιών.

13.1 Σκοπός

Ο σκοπός της βιβλιοθήκης είναι να καταστήσει την ενσωμάτωση και την αξιοποίηση της τεχνολογίας blockchain πιο εύκολη και προσιτή για προγραμματιστές. Παρέχει μια απλούστερη διαδικασία για την πιστοποίηση και την επικοινωνία μεταξύ των frontend εφαρμογών και των κρυπτοπορτοφολιών.

13.2 Χρήση

Η βιβλιοθήκη *Web3Button* έχει την μορφή ενός node module [70], με αποτέλεσμα η εγκατάσταση και η χρήση της να είναι ιδιαίτερα εύκολη και προσιτή σε όλα τα JavaScript frameworks [71]. Ουσιαστικά, αποτελείται από μια κλάση (class) που πρέπει να αρχικοποιηθεί με ορισμένες τιμές και έπειτα προσφέρει διαφόρους μεθόδους για τον έλεγχο της συμπεριφοράς της. Στην επόμενη υπό-ενότητα θα αναλυθούν περαιτέρω τα χαρακτηριστικά της.

13.3 Απαιτούμενες Υπηρεσίες

Για την λειτουργία της βιβλιοθήκης πρέπει να έχει δημιουργηθεί ένας web server ο οποίος θα είναι υπεύθυνος για την επικοινωνία μεταξύ της διεπαφής χρήστη και του blockchain. Ακόμα, θα πρέπει να διαθέτει μια διαδρομή (route) η οποία, χρησιμοποιώντας τα εργαλεία που προσφέρει η βιβλιοθήκη *ethers.js*, να ελέγχει εάν ο αποστολέας της αίτησης (request) είναι ο πραγματικός κάτοχος του πορτοφολιού.

13.4 Αρχικοποίηση

Κατά την αρχικοποίηση της βιβλιοθήκης, ο προγραμματιστής οφείλει να παραχωρήσει τα πέντε ακόλουθα δεδομένα:

- Μια διαδρομή για την πιστοποίηση της ιδιοκτησίας του κρυπτοπορτοφολιού (π.χ., *http://localhost:8787/web3auth*).
- Μια διαδρομή για την απόκτηση του ρολού του χρήστη (π.χ., *http://localhost:8787/retrieve-role*).
- Το Chain ID του δίκτυο blockchain στο οποίο πρέπει να είναι συνδεδεμένο το κρυπτοπορτοφόλι του χρήστη (π.χ., 1337).

- Όλους τους διαθέσιμους ρόλους διαχωρισμένους με κόμματα και χωρίς κενά, η σειρά είναι σημαντική (π.χ., *None,Student,Professor,Staff,Admin*).
- Να επιλέξει, χρησιμοποιώντας νούμερο, τον ελάχιστο ρολό που πρέπει να διαθέτει ο χρήστης για να του παραχωρηθεί πρόσβαση.

13.5 Χαρακτηριστικά και Δυνατότητες

Κατά κύριο λόγο η βιβλιοθήκη πραγματοποιεί τους απαραίτητους ελέγχους για ένα τυπικό σενάριο ταυτοποίησης όταν γίνεται χρήση της τεχνολογίας blockchain. Μόλις ο χρήστης πιέσει το κουμπί, ονόματι **web3 auth**, ξεκινάει μια σειρά από ελέγχους, εάν κάποιος αποτύχει παρουσιάζεται στο UI ένα μήνυμα (toast notification) που αναφέρει την αίτια. Η βιβλιοθήκη *toastify-js* έχει χρησιμοποιηθεί για την διαχείριση αυτών των μηνυμάτων.

Οι **έλεγχοι** είναι οι ακόλουθοι και πραγματοποιούνται με την αναφερόμενη σειρά:

1. Ύπαρξη κρυπτοπορτοφολιού.
2. Ύπαρξη συνδεδεμένου λογαριασμού.
3. Σύνδεση στο επιθυμητό δικτύου.
4. Απόδειξη της γνησιότητας της ιδιοκτησία του παρόντος κρυπτοπορτοφολιού.
5. Κατοχή SoulBound Token (SBT).
6. Κατοχή ρολού που να ικανοποιεί τους περιορισμούς πρόσβασης.

Για την **διαχείριση της συμπεριφοράς** της εφαρμογής **μετά από μια επιτυχής ταυτοποίηση ή όχι**, η βιβλιοθήκη προσφέρει δυο συναρτήσεις, την *onSuccess* και την *onFailure*. Αυτές οι συναρτήσεις δέχονται ως παραμέτρους άλλες συναρτήσεις. Η συγκεκριμένη αρχιτεκτονική χρησιμοποιείται για δώσει στον προγραμματιστή τον πλήρη έλεγχο, επιτρέποντας του να διαχειριστεί τι θα συμβεί σε κάθε μια από τις δυο πιθανές περιπτώσεις.

Τέλος, η βιβλιοθήκη παρέχει και την συνάρτηση *render*. Αυτή είναι απαραίτητή για την **εμφάνιση του κουμπιού** *«web3 auth»* **στην διεπαφή του χρήστη**. Δέχεται ως είσοδο ένα HTML στοιχείο και τοποθετεί μέσα του το κουμπί. Δηλαδή, για να χρησιμοποιηθεί από έναν προγραμματιστής, θα πρέπει πρώτα να δημιουργηθεί ένα στοιχείο τύπου δοχείου (container) σαν το ακόλουθο:

<div *id='web3button-container'* />

Έπειτα πρέπει να επιλεχθεί και να αποθηκευτεί σε μια μεταβλητή χρησιμοποιώντας JavaScript. Τέλος θα πρέπει να παραχωρηθεί ως επιχείρημα (argument) στην μέθοδο *render*.

Συμπεράσματά

Η βιβλιοθήκη Web3Button αναπτύχθηκε για να διευκολύνει τους προγραμματιστές να ενσωματώσουν λειτουργίες blockchain στις εφαρμογές τους, επιτρέποντας εύκολη ταυτοποίηση μέσω κρυπτοπορτοφολιών. Απλοποιεί την επικοινωνία μεταξύ frontend διεπαφών και blockchain, και παρέχει μεθόδους για διάφορες λειτουργίες, όπως την επαλήθευση ταυτότητας και την παρακολούθηση των δικαιωμάτων χρήστη.

Κεφάλαιο 14: Προσομοίωση Δημιουργίας Ψηφιακού Πάσου

Εισαγωγή

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, θα εξεταστεί ενδελεχώς η αναπτυγμένη εφαρμογή, αποδίδοντας ιδιαίτερη προσοχή στις **υποκείμενες τεχνολογίες και** στα **απαραίτητα λογισμικά εργαλεία**. Συγκεκριμένα, θα παρουσιαστούν οι διαδικασίες για την εγκατάσταση και την εκτέλεση της εφαρμογής, περιλαμβάνοντας:

1. Τις οδηγίες απόκτησης του πηγαίου κώδικα μέσω GitHub.
2. Τα βήματα εκκίνησης και επαλήθευσης της λειτουργικότητας της εφαρμογής σε τοπικό υπολογιστή.

Επιπλέον, θα **αναλυθούν τέσσερα σενάρια χρήσης**, που αποσκοπούν στην παρουσίαση των εξής λειτουργιών:

1. Την ενεργοποίηση υπηρεσιών και τον έλεγχο της απρόσκοπτης λειτουργίας τους μέσα από τη διεπαφή χρήστη.
2. Τη διαδικασία ταυτοποίησης του χρήστη με χρήση της βιβλιοθήκης "Web3Button".
3. Την δημιουργία και την επιβεβαίωση ενός Soulbound Token (SBT) για έναν φοιτητή.
4. Την παραγωγή και την εφαρμογή ενός Access Token από τον φοιτητή που έχει λάβει το SBT.

Με στόχο την παροχή μιας καθαρής και οργανωμένης παρουσίασης, η ενότητα αυτή αποσκοπεί στην διευκόλυνση του αναγνώστη να κατανοήσει την διαδικασία ανάπτυξης και την πρακτική εφαρμογή των καινοτόμων λύσεων που προτείνονται στην διπλωματική εργασία.

14.1 Απαραίτητο Λογισμικό

Ο πηγαίος κώδικας είναι αναρτημένος σε μια δημοσιά αποθήκη (repository) στο GitHub. Ο παρακάτω σύνδεσμος (link) μπορεί να χρησιμοποιηθεί για την μετάβαση στο συγκεκριμένο repository.

https://github.com/elblock/CryptoID

Ακολουθούν τα απαραίτητα λογισμικά για την απόκτηση και εκτέλεση του πηγαίου κώδικα:

1. **Web Browser Chrome**, καθώς η εφαρμογή αποτελεί μια διαδικτυακή εφαρμογή, η ύπαρξη ενός περιηγητή είναι απαραίτητη. Προτείνεται ο Chrome καθώς η εφαρμογή έχει δοκιμαστεί πάνω στον συγκεκριμένο περιηγητή.

2. **Κρυπτοπορτοφόλι MetaMask**, αναπόσπαστο κομμάτι κάθε αποκεντρωμένης εφαρμογής (DApp) είναι το κρυπτοπορτοφόλι. Κατά την ανάπτυξη και την δοκιμή

της εφαρμογής το MetaMask ήταν το επιλεγμένο κρυπτοπορτοφόλι. Γι' αυτόν τον λόγο προτείνεται η χρήση του συγκεκριμένου.

3. **NodeJS**, είναι ένα περιβάλλον που επιτρέπει την εκτέλεση κώδικα JavaScript εκτός του περιηγητή. Χωρίς αυτό δεν θα ήταν εφικτή η χρήση της JavaScript για την ανάπτυξη των εξάπτων συμβολαίων, του web server και της βιβλιοθήκης Web3Button.

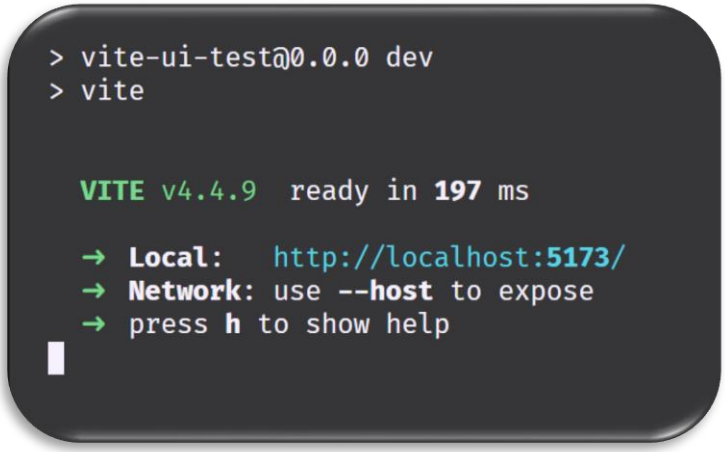
4. **Git**, αυτό το λογισμικό είναι απαραίτητο για την επικοινωνία του GitHub με τον τοπικό υπολογιστή.

14.2 Παρουσίαση Σεναρίων

Το **πρώτο σενάριο** αφορά την εκκίνηση όλων των υπηρεσιών που απαιτούνται για την ομαλή λειτουργία της αναπτυγμένης εφαρμογής. Η σειρά με την οποία θα εκκινηθούν είναι σημαντική.

Για την έναρξη των υπηρεσιών θα πρέπει να μεταβούμε στο φάκελο ρίζα (root directory) οπού θα χρειαστεί να ανοίξουμε τέσσερα τερματικά (terminals) που θα τρέχουν τις υπηρεσίες. Αυτά είναι τα εξής:

- Για το **Frontend** (Εικόνα 29): Το τερματικό πρέπει να δείχνει στον φάκελο *«../../cryptopass_frontend/vite-ui-test»*. Έπειτα πρέπει να εκτελέσουμε την ακόλουθη εντολή:
  npm run dev
  Μόλις εκτελεστεί η εντολή, θα πρέπει να εμφανιστεί η παρακάτω διεπαφή:

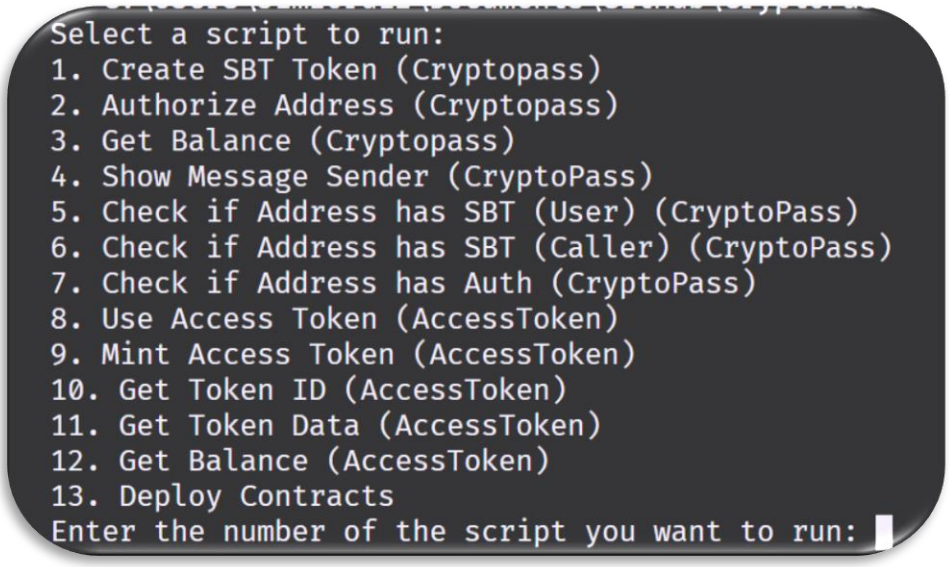


Εικόνα 29: Σενάριο 1ο, αποτέλεσμα έναρξης της διεπαφή χρήστη στο τερματικό.

- Για το **δίκτυο blockchain (**Εικόνα 30**)**: Το τερματικό πρέπει να έχει επιλέξει τον φάκελο *«../../cryptopass_smartContracts»*. Στην συνέχεια, τρέχουμε την εξής εντολή:
  npx hardhat node
  Αυτή η εντολή θα ξεκινήσει ένα τοπικό blockchain δίκτυο καθώς επίσης θα δημιουργήσει και έναν server που θα λειτουργεί ως πάροχος του blockchain. Η αναμενόμενη διεπαφή της κονσόλας πρέπει να μοιάζει με την παρακάτω:

Εικόνα 30: Σενάριο 1ο, αποτέλεσμα έναρξης του τοπικού δικτύου blockchain στο τερματικό.

- Για την **προώθηση των έξυπνων συμβολαίων (**Εικόνα 31**)**: Εφόσον υπάρχει blockchain, η προώθηση των έξυπνων συμβολαίων είναι εφικτή. Για την υλοποίηση αυτής της διαδικασίας πρέπει να ανοίξουμε ένα τερματικό στο ίδιο φάκελο με πριν, δηλαδή στον *«../../cryptopass_smartContracts»*. Έπειτα, πρέπει να εισάγουμε την εξής εντολή:
  npx hardhat run --network localhost scripts/actions/scriptRunner.ts
  Μόλις τρέξει η εντολή θα εμφανιστεί η ακόλουθη διεπαφή.



Εικόνα 31: Σενάριο 1ο, αποτέλεσμα εκτέλεσης του σεναρίου (script) για την προώθηση των συμβολαίων στο δίκτυο blockchain.

Η επιλογή *«13»* πρέπει να επιλεχθεί για την προώθηση των συμβολαίων. Μόλις ολοκληρωθεί η διαδικασία, μπορούμε να προχωρήσουμε στη επόμενη και τελευταία υπηρεσία.

- Για την έναρξη του **Web Server (**Εικόνα 32**)**: Η τελευταία υπηρεσία είναι ο web server. Για την εκκίνηση της πρέπει να μεταβούμε στην τοποθεσία *«../../cryptopass_ws»* και ανοίξουμε ένα ακόμα τερματικό. Η εντολή που πρέπει να χρησιμοποιήσουμε είναι η ακόλουθη:
  npm start
  Η διεπαφή που περιμένουμε να αντικρίσουμε είναι η παρακάτω.

```
> cryptopass_ws@1.0.0 start
> tsx --tsconfig ./tsconfig.json src/index.ts

You are in [PRODUCTION MODE]

For Web3 Auth Testing: http://localhost:8787/web3auth
For QR Code Testing: http://localhost:8787/qrCodeCreator

Running on: http://localhost:8787/
The WS Wallet Public Addr:  0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC
Balance of The WS Wallet is: 10250.0 ETH
[GOOD]: WS is Authorized by the Contract!
```

Εικόνα 32: Σενάριο 1ο, αποτέλεσμα έναρξης του web server στο τερματικό.

Για την ολοκλήρωση του 1ου σεναρίου πρέπει μονάχα να εισάγουμε το URL *«http://localhost:5173/»* στον περιηγητή μας. Εάν στο Βήμα #1 (Step #1) όλα τα φωτάκια είναι πράσινα, με μοναδική εξαίρεση το τελευταίο (Logged In), όλες οι υπηρεσίες είναι ενεργές και έχουν αρχικοποιηθεί σωστά. Όπως απεικονίζεται στην Εικόνα Εικόνα 33**.**

Εικόνα 33: Σενάριο 1ο, επιβεβαίωση σωστής αρχικοποίησης υπηρεσιών.

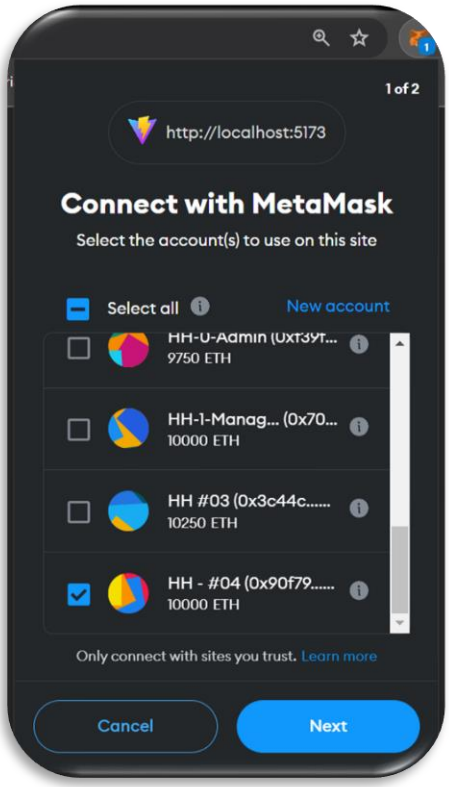
Το **2ο σενάριο** αφορά την χρήση της βιβλιοθήκης *«Web3Button»* ως εναλλακτική επιλογή για την ταυτοποίηση ενός μέλους της γραμματείας με αποστερώ σκοπό την πρόσβαση στην εφαρμογή που θα επιτρέπει σε ένα εξουσιοδοτημένο χρήστη να δημιουργήσει SBTs (ψηφιακές ταυτότητες).

Καθώς έχουν ήδη αναλυθεί, σε προηγουμένη ενότητα, οι λειτουργείες και δυνατότητες της βιβλιοθήκης, τώρα θα επικεντρωθούμε στην εμπειρία του χρήστη. Θεωρούμε ότι το κρυπτοπορτοφόλι MetaMask είναι ήδη εγκαταστημένο και ο χρήστης έχει συνδεθεί σε αυτό παρέχοντας τον κωδικό του. Πατώντας το μπλε κουμπί *«web3 auth»* ανοίγει το ακόλουθο παράθυρο.

Εδώ το MetaMask ρωτάει τον χρήστη με ποιον λογαριασμό επιθυμεί να συνδεθεί στην διαδικτυακή εφαρμογή. Στην προκείμενη περίπτωση διαλέγουμε τον *«HH - #04»* (δηλ. τον 4ο από τους λογαριασμούς που μας παρέχει το Hardhat) όπως παρουσιάζεται την Εικόνα 34.

Στην συνέχεια, το MetaMask θα εμφανίσει ένα ακόμη παράθυρο οπού ζητάει την υπογραφή ενός μνήματος που αποτελείται από έναν τυχαίο 9-ψηφιο ακέραιο αριθμό, όπως φαίνεται στην Εικόνα 35.

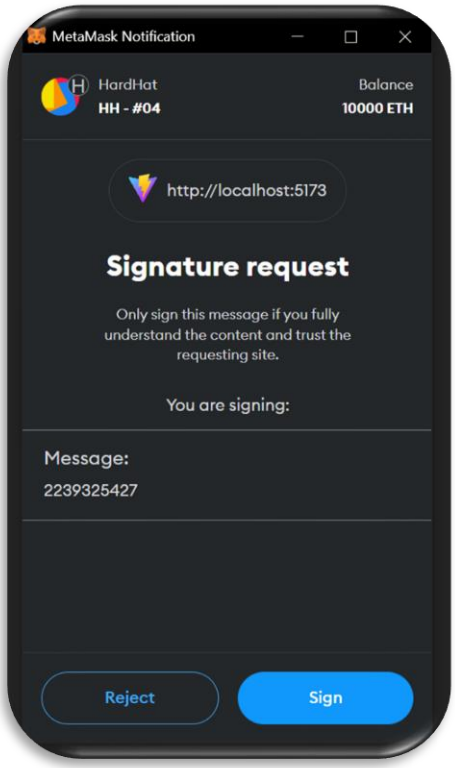
Αυτή η διαδικασία είναι τυπική σε αποκεντρωμένες εφαρμογές, καθώς θεωρείται καλή πρακτική να αποδεικνύεται πάντα η γνησιότητα της ιδιοκτησίας ενός λογαριασμού πριν αυτός χρησιμοποιηθεί από την εφαρμογή.



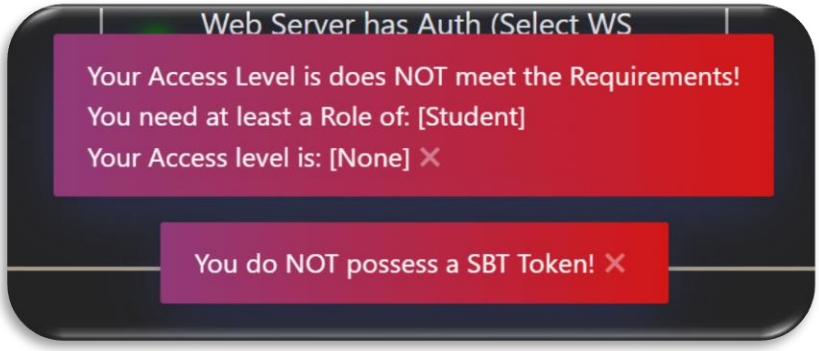
Εικόνα 34: Σενάριο 2ο, σύνδεση κρυπτοπορτοφολιού με διαδικτυακή εφαρμογή.

Για την πραγματοποίηση της επαλήθευση είναι απαραίτητη η χρήση ενός web server, για αυτό η διεπαφή χρήστη περιμένει την ανταπόκριση του server πριν συνεχίσει την εκτέλεση του υπολοίπου κώδικα.

Μόλις επιστρέψει η απάντηση από τον server εμφανίζονται δυο κόκκινα μηνύματα στο πάνω μέρος της εφαρμογής, όπως είναι απεικονισμένα στην Εικόνα 36. Αυτό συμβαίνει επειδή ο συγκεκριμένος λογαριασμός δεν διαθέτει ένα SBT, με αποτέλεσμα ούτε ρολό.

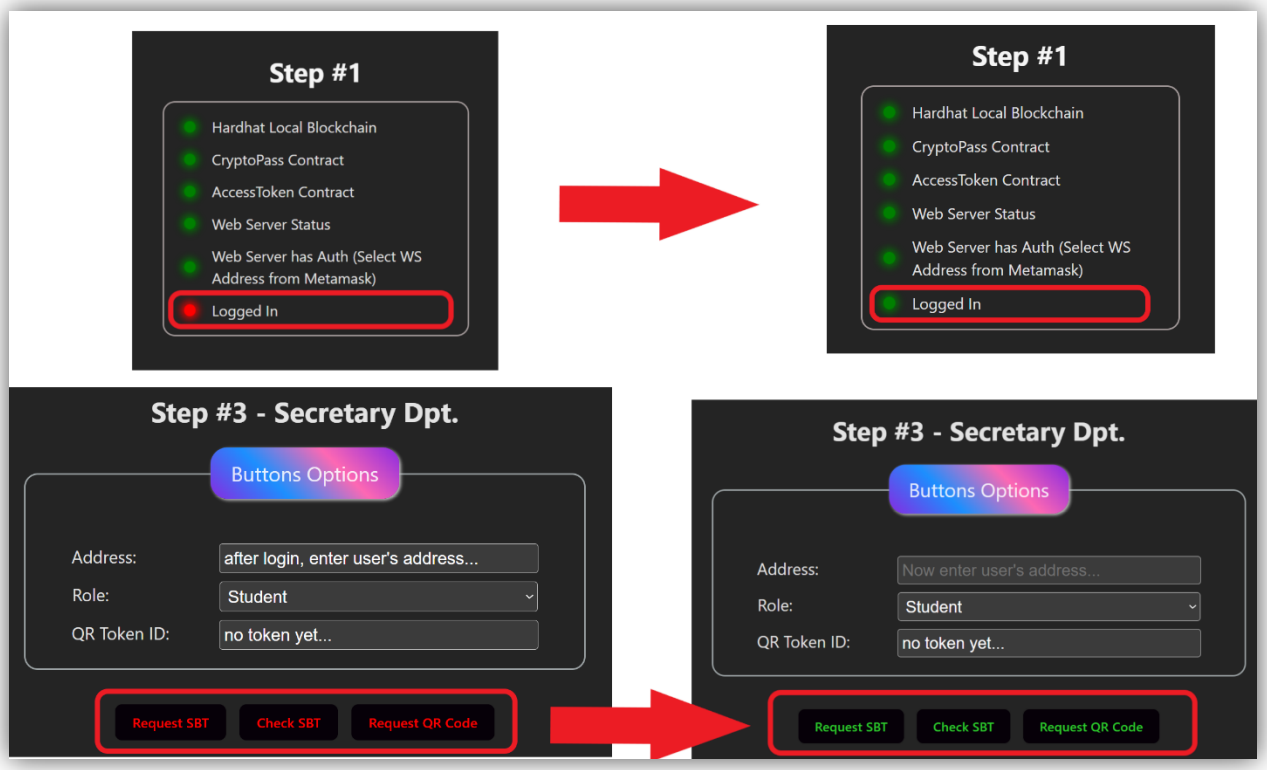


Εικόνα 35: Σενάριο 2ο, υπογραφή μηνύματος για την πιστοποίηση της πραγματικής ιδιοκτησίας του κρυπτοπορτοφολιού.



Εικόνα 36: Σενάριο 2ο, μηνύματα σφάλματος κατά την ταυτοποίηση του χρήστη.

Για να μας επιτρέψει η βιβλιοθήκη *«Web3Button»* να συνδεθούμε στην εφαρμογή, πρέπει να χρησιμοποιήσουμε τον μόνο λογαριασμό που έχει στην κατοχή ένα *SBT* και *δικαιώματα Admin*. Αυτός είναι το λογαριασμός *«HH #03»*. Επιλέγοντας τον από τη διεπαφή του MetaMask, μπορούμε να αποκτήσουμε πρόσβαση και να συνδεθούμε στην εφαρμογη. Όταν συμβεί αυτό, τα γράμματα των κουμπιών στο τμήμα *«Step #3 – Secretary Dpt.»*, θα μετατραπούν από κόκκινα σε πράσινα και επιπλέον στο *«Step #1»* το φωτάκι *Logged In* από κόκκινο θα γίνει πράσινο, όπως φαίνεται στην Εικόνα 37.

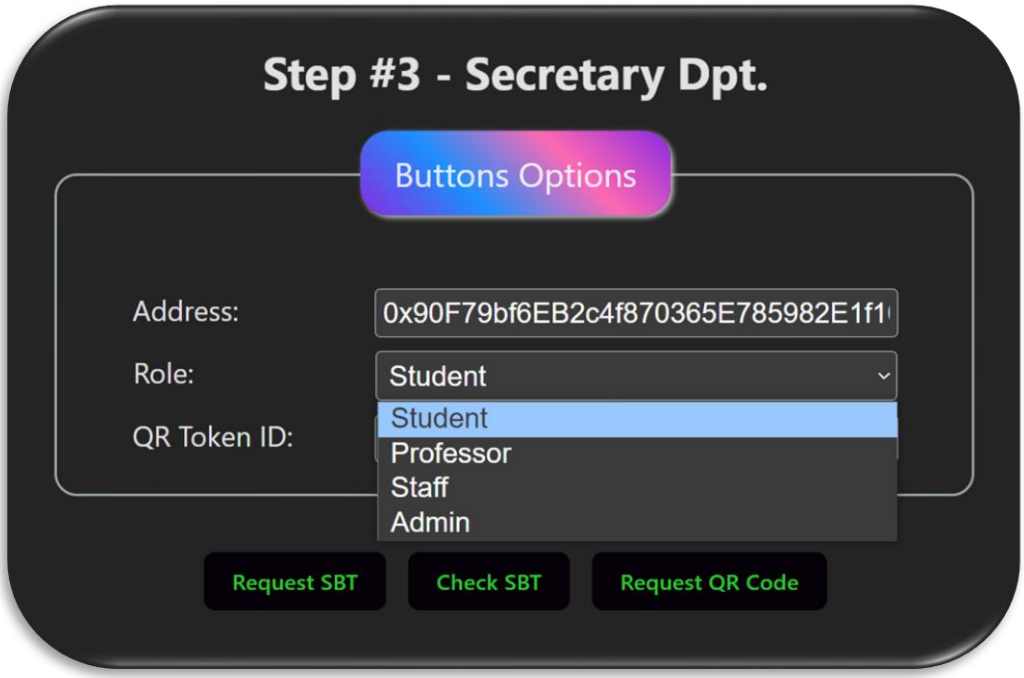


Εικόνα 37: Σενάριο 2ο, ενδείξεις επιτυχής σύνδεσης στην εφαρμογή.

Συνεχίζουμε στο **3ο σενάριο**, οπού θα πραγματοποιηθεί η δημιουργία ενός SBT για τον λογαριασμό *«HH - #04»*. Το πρώτο βήμα είναι η επικόλληση της διεύθυνσης του *HH - #04* στο πεδίο *«Address»*. Υστέρα ακολουθεί η αναθέσει ρολού, οι διαθέσιμες επιλογές είναι οι εξής, όπως αναγράφεται στην Εικόνα 38:

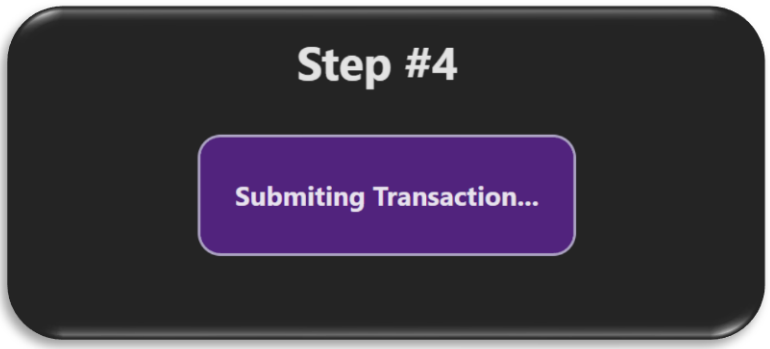
- Student
- Professor
- Staff
- Admin

Το πεδίο «QR Token ID» δεν επιτρέπει εισαγωγή κειμένου (δηλ. είναι read-only) και απλώς δείχνει το αναγνωριστικό (ID) του ενεργού Access Token. Η τιμή του θα αλλάξει στο 4ο σενάριο.



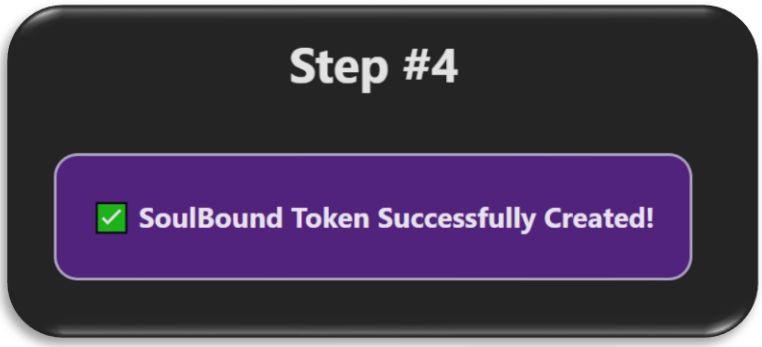
Εικόνα 38: Σενάριο 3ο, οι διαθέσιμοι ρολόι προς ανάθεση για μια ψηφιακη ταυτότητα.

Τώρα, για την δημιουργία της ψηφιακής ταυτότητα το πάτημα του κουμπιού *«Request SBT»* είναι αρκετό. Θα εμφανιστεί ένα μήνυμα που θα αναγράφει *«Submitting Transaction…»*, όπως απεικονίζεται την Εικόνα 39.



Εικόνα 39: Σενάριο 3ο, μήνυμα που ενημερώνει τον χρήστη για την κατάσταση της συναλλαγής.

Μόλις η συναλλαγή επικυρωθεί από το δίκτυο (συνήθως χρειάζεται 3-4 δευτερόλεπτα) το προηγούμενο μήνυμα θα αντικατασταθεί από το παρακάτω (Εικόνα 40).



Εικόνα 40: Σενάριο 3ο, μήνυμα που ενημερώνει τον χρήστη για την επιτυχή δημιουργία της ψηφιακής ταυτότητας.

Ακόμα, προσφέρεται η δυνατότητα ανάκτησης του ρολού μιας διεύθυνσης μέσω του κουμπιού *«Check SBT»*. Επιστρέφει τον ρολό της διεύθυνσης που έχει εισαχθεί στο πεδίο *«Address»*. Για παράδειγμα, εάν πατηθεί το κουμπί κρατώντας την ιδιά διεύθυνση, εμφανίζεται το ακόλουθο μήνυμα (Εικόνα 41).

Εικόνα 41: Σενάριο 3ο, μήνυμα που αναφέρει τον ρολό της διεύθυνσης που έχει τοποθετηθεί στο πεδίο *«Address»*.

Φτάσαμε στο **4ο και τελευταίο σενάριο**, για την πλήρη κατανόηση του πρέπει να γίνει ξεκάθαρο ότι πλέον δεν αναφερόμαστε στην ιδιά ομάδα ατόμων, ούτε στην ιδιά θεωρητική εφαρμογή. Η λογική και η λειτουργικότητα που έχει παρουσιαστεί έως τώρα, αφορούσε εξουσιοδοτημένο προσωπικό που αρμοδιότητα τους είναι η δημιουργία ψηφιακών ταυτοτήτων για φοιτητές, εάν και μπορεί να επεκταθεί για το υπόλοιπο προσωπικό του πανεπιστημίου.

Το κοινό για το οποίο προορίζεται η λειτουργικότητα που ακολουθεί, αφορά κυρίως τους φοιτητές και το υπόλοιπο προσωπικό του πανεπιστήμιου.

Εφόσον πλέον, ο φοιτητής που διαθέτει τον λογαριασμό *«HH - #04»* έχει στην κατοχή ένα SBT μπορεί να πραγματοποιήσει δυο ενεργείς:

- **Είσοδο** σε **ψηφιακές υπηρεσίες** που υποστηρίζουν την χρήση κρυπτοπορτοφολιού.
- **Είσοδο** σε **φυσικές υπηρεσίες** που υποστηρίζουν το σύστημα που έχει αναπτυχθεί σε αυτήν την εργασία.

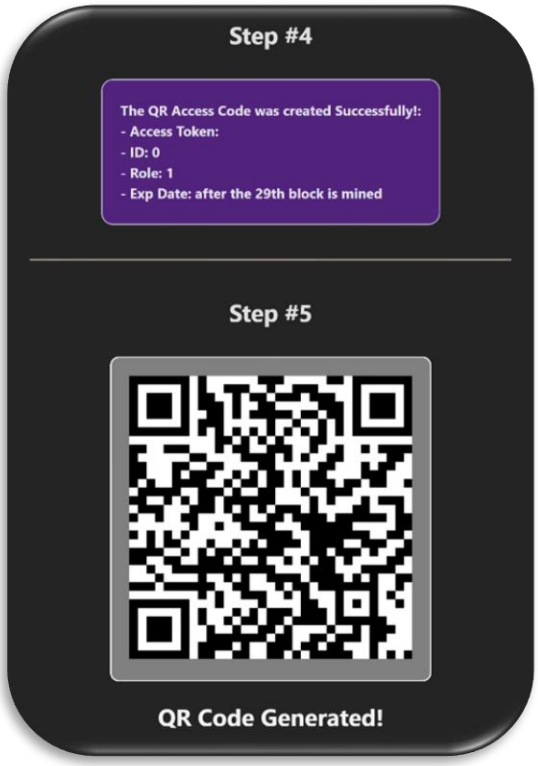
Για την είσοδο σε **ψηφιακές υπηρεσίες**, το μόνο που πρέπει να κάνει ο φοιτητής είναι να συνδεθεί στο κρυπτοπορτοφόλι του. Έπειτα, είναι μονάχα μερικά κλικ μακριά από την είσοδο σε οποιαδήποτε υπηρεσία.

Για να αποκτήσει ισχύ η συγκεκριμένη εναλλακτική ταυτοποίησης χρήστη, θα πρέπει να διαδοθεί η εφαρμογή της τεχνολογίας blockchain. Εάν συμβεί αυτό, οι φοιτητές θα μπορούν να χρησιμοποιούν το κρυπτοπορτοφόλι τους για να συνδέονται σε μια μεγάλη πληθώρα εφαρμογών δίχως την ανάγκη για πολλαπλούς λογαριασμούς, ενώ ταυτόχρονα έχουν τον πλήρη έλεγχο των προσωπικών τους δεδομένων.

Όσον αφορά την είσοδο σε **φυσικές υπηρεσίες**, η προσέγγιση που χρησιμοποιήθηκε είναι η κωδικοποίηση των *access token* σε *κωδικούς QR*. Με αυτόν τον τρόπο, μια φυσική υπηρεσία, όπως η σίτιση ή η βιβλιοθήκη, μπορεί με ευκολία και μέγιστη αξιοπιστία να αποκτήσει την πληροφορία ότι ένας φοιτητής έχει στην κατοχή του φοιτητικό πάσο.

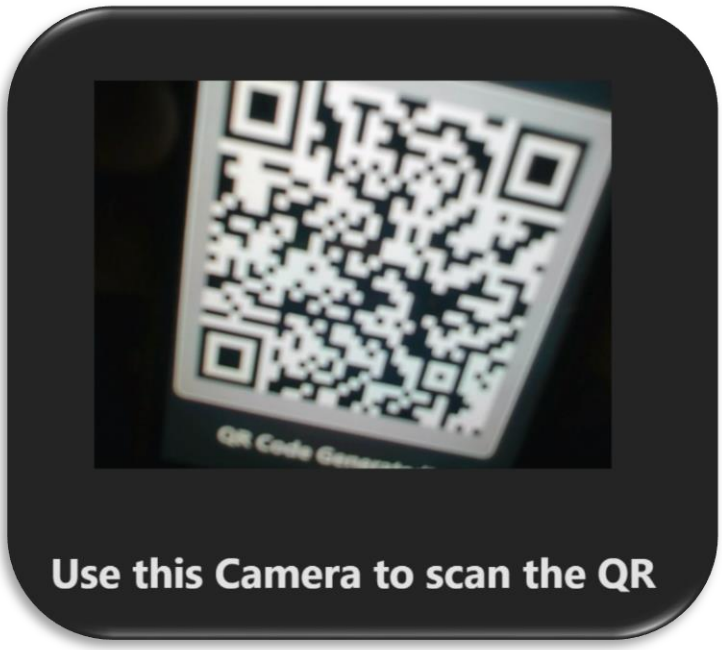
Επιστρέφοντας στην παρουσίαση της εφαρμογής, μπορούμε να φανταστούμε ότι το κουμπί *«Request QR Code»* θα βρίσκεται στην εφαρμογή που προορίζεται για τους φοιτητές. Μόλις πιεστεί, η διεπαφή χρήστη επικοινωνεί με τον web server για την απόκτηση ενός *access token*. Με την σειρά του ο web server επικοινωνεί με το συμβόλαιο *AccessToken*, οπού πραγματοποιούνται και όλοι οι απαραίτητοι έλεγχοι.

Μόλις η συναλλαγή επαληθευτεί η διεπαφή του χρήστη παίρνει την ακολουθεί μορφή (Εικόνα 42).



Εικόνα 42: Σενάριο 4ο, μήνυμα που αναφέρει της λεπτομερείς του access token και το παραγόμενος κωδικός QR.

Για την προσομοίωση του scanner, θα γίνει χρήση της κάμερας του υπολογιστή της συγγραφέα. Ακόμα, θα χρησιμοποιηθεί η κάμερα ενός κινητού τηλεφώνου για την φωτογράφηση του κωδικού QR από την οθόνη του υπολογιστή και έπειτα η οθόνη του κινητού ώστε να σκαναριστεί από την κάμερα του υπολογιστή, όπως φαίνεται στην Εικόνα 43.

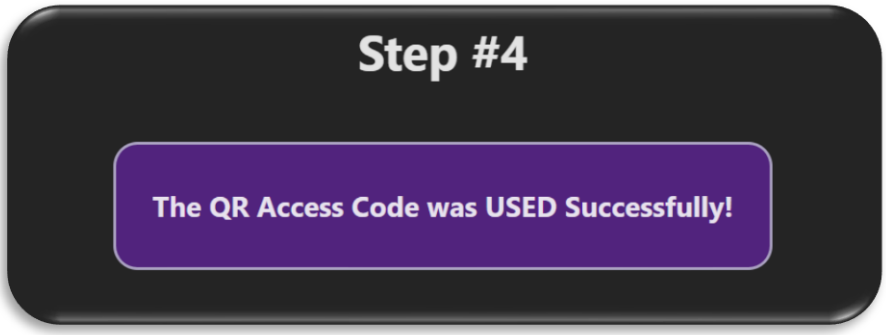


Εικόνα 43: Σενάριο 4ο, χρήση της οθόνης της ενός κινητού τηλεφώνου και της κάμερας του υπολογιστή για προσομοίωση του σκαναρίσματος ενός κωδικού QR.

Μόλις η κάμερα του υπολογιστή ανιχνεύσει τον κωδικό QR, η διεπαφή του χρήστη θα ανανεωθεί. Ένα μήνυμα με πράσινα γράμματα θα εμφανιστεί για να ενημερώσει τον χρήστη για την επιτυχή ανίχνευση. Ακόμα, θα φανερωθεί ένα κουμπί, ονόματι *«Use Access Token»*, όπως απεικονίζεται στην Εικόνα 44.

Σε ένα πραγματικό σενάριο, ο λειτουργία αυτού του κουμπιού θα ενεργοποιούνταν την στιγμή που θα αναγνωριζόταν ο κωδικός QR. Η λειτουργία είναι αρκετά απλή, απλώς στέλνει στο έξυπνο συμβόλαιο *AccessToken*, μέσω του web server, τα δεδομένα του access token που αποκωδικοποιήθηκαν από τον κωδικό QR. Εάν είναι έγκυρα, η κάμερα και ο κωδικός QR εξαφανίζονται και την θέση τους παίρνει το μήνυμα στην Εικόνα 45. Στην περίπτωση κάποιου προβλήματος, ο χρήστης ενημερώνεται μέσω των μηνυμάτων σφάλματος.

Εικόνα 44: Σενάριο 4ο, επιτυχής ανίχνευση κωδικού QR και η εμφάνιση του κουμπιού *«Use Access Token»*.



Εικόνα 45: Σενάριο 4ο, επιτυχής χρήση του access token.

Τέλος, καθώς το access token χρησιμοποιήθηκε ο χρήστης μπορεί να επαναλάβει την διαδικασία για την απόκτηση ενός καινούργιου.

Συμπεράσματα

Η τεχνολογία blockchain έχει αποδειχθεί επαναστατική στον τρόπο με τον οποίο αντιλαμβανόμαστε και διαχειριζόμαστε τα δεδομένα, προσφέροντας μια νέα διάσταση στην ασφάλεια και την αξιοπιστία σε διάφορους τομείς.

Μέσω της παρούσας διπλωματικής εργασίας, αναπτύχθηκε ένα σύστημα που εκμεταλλεύεται τα πλεονεκτήματα των Soulbound Tokens (SBTs) για τη δημιουργία ψηφιακών ταυτοτήτων, αποδεικνύοντας τις δυνατότητες της blockchain στην αυθεντικοποίηση οντοτήτων σε εκπαιδευτικά ιδρύματα.

Ωστόσο, η τρέχουσα υλοποίηση του συστήματος δεν είναι απαλλαγμένη από ατέλειες. Η ενσωμάτωση του παρόντος συστήματος, τόσο σε φυσικούς όσο και σε ψηφιακούς χώρους, απαιτεί πόρους. Συγκεκριμένα, για φυσικές εγκαταστάσεις απαιτείται ένας μίνι υπολογιστής, όπως ένα Raspberry Pi, και ένας σαρωτής QR κωδικών, ενώ για ψηφιακά περιβάλλοντα απαιτείται η πρόσληψη ενός προγραμματιστή για την σύνδεση της υπηρεσίας με το blockchain μέσω της βιβλιοθήκης Web3Button.

Επιπλέον, υπάρχουν μερικά κενά ασφαλείας, όπως η απουσία περιορισμού στον αριθμό των QR κωδικών που μπορεί να δημιουργήσει ένας εξουσιοδοτημένος χρήστης, επιτρέποντας την διανομή τους σε άλλους χρήστες με αποτέλεσμα την παραχώρηση πρόσβασης σε μη εξουσιοδοτημένα μέλη.

Για την επέκταση του συστήματος, θα μπορούσαν να εξεταστούν τρεις διαφορετικοί τρόποι:

1. Ενσωμάτωση ενός μηχανισμού που θα περιορίζει τον αριθμό των QR κωδικών που μπορεί να δημιουργήσει ένας χρήστης, αυξάνοντας έτσι την ασφάλεια του συστήματος.

2. Ανάπτυξη ενός πιο σύνθετου συστήματος επικύρωσης για τη ψηφιακή χρήση, που θα ελαχιστοποιεί την ανάγκη για εξωτερική προγραμματιστική εμπειρία.

3. Ανάπτυξη ενός συστήματος όπου θα πραγματοποιεί αυτόματα την δημιουργία των SBTs δίχως την ανάγκη ανθρωπίνου δυναμικού.

Η έρευνα στον τομέα της blockchain είναι δυναμική και συνεχώς εξελισσόμενη, προσφέροντας ανεξάντλητες δυνατότητες για καινοτομία και βελτίωση σε πληθώρα εφαρμογών. Η παρούσα εργασία αποτελεί μόνο την αφετηρία για περαιτέρω έρευνα και ανάπτυξη στη χρήση της blockchain, καθώς ανοίγει το δρόμο για νέες εξερευνήσεις και ανακαλύψεις στον συναρπαστικό κόσμο της τεχνολογίας.

Βιβλιογραφία

[1] G. Karame, E. Androulaki και S. Capkun, "Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin", IACR Cryptol. ePrint Arch., 2012

[2] J. A. Kroll, I. C. Davey και E. W. Felten, "The economics of bitcoin mining or bitcoin in the presence of adversaries", Proc. WEIS, 2013

[3] M. del Castillo, Chain is Now Working on Six ‘Citi-Sized’ Blockchain Networks, 2017.

[4] F. Tschorsch και B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies", IEEE Commun. Surveys Tuts., 2016.

[5] Z. Zheng, S. Xie, H. Dai, X. Chen και H. Wang, "An overview of blockchain technology: Architecture consensus and future trends", Proc. IEEE Int. Congr. Big Data (BigData Congr.), 2017.

[6] F. Glaser, "Pervasive decentralisation of digital infrastructures: A framework for blockchain enabled system and use case analysis", Proc. 50th Hawaii Int. Conf. Syst. Sci., 2017.

[7] K. J. O’Dwyer και D. Malone, "Bitcoin mining and its energy footprint", Proc. 25th IET Irish Signals Syst. Conf., 2014.

[8] Manimuthu, R. V. Sreedharan, R. G και D. Marwaha, "A literature review on bitcoin: Transformation of crypto currency into a global phenomenon", IEEE Eng. Manage. Rev., 2019.

[9] D. Shrier, W. Wu και A. Pentland, Blockchain & infrastructure (identity data security), Cambridge, MA, USA, 2016.

[10] Q. Wang, X. Li και Y. Yu, "Anonymity for bitcoin from secure escrow address", IEEE Access, 2018.

[11] H. Yu, Z. Yang και R. O. Sinnott, "Decentralized big data auditing for smart city environments leveraging blockchain technology", IEEE Access, 2019.

[12] M. Crosby, P. Pattanayak, S. Verma και V. Kalyanaraman, "Blockchain technology: Beyond bitcoin", Appl. Innov., 2016.

[13] R. L. Twesige, "A simple explanation of bitcoin and blockchain technology", 2015.

[14] T. McGhin, K.-K. R. Choo, C. Z. Liu και D. He, "Blockchain in healthcare applications: Research challenges and opportunities", J. Netw. Comput. Appl., 2019.

[15] B. D. Glass, "Counterfeit drugs and medical devices in developing countries", Res. Rep. Tropical Med., 2014.

[16] Counterfeit of Medicines Causes 37 000 Job Losses in EU Pharma Industry—ECA Academy, 2019.

[17] Azaria, A. Ekblaw, T. Vieira και A. Lippman, "MedRec: Using blockchain for medical data access and permission management", Proc. 2nd Int. Conf. Open Big Data (OBD), 2016.

[18] R. H. Lasseter και P. Piagi, "Microgrid: A conceptual solution", Proc. IEEE 35th Annual Power Electron. Spec. Conf., 2004.

[19] Cohn, T. West και C. Parker, "Smart after all: Blockchain smart contracts parametric insurance and smart energy grids", Georgetown Law Technol. Rev., 2017.

[20] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng και Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things", IEEE Trans. Ind. Informat., 2018.

[21] L. Lee, "New kids on the blockchain: How bitcoin’s technology could reinvent the stock market", Hastings Bus. Law J., 2015.

[22] D. Tapscott και A. Tapscott, "How blockchain will change organizations", MIT Sloan Manage. Rev., 2017.
[23] Fair Fight Donate Via Actblue, Μάιος 2019
[24] de Beauchesne, Q. NFT Month—History of NFTs. Ownest.Io. Μάϊος 2021.
[25] Namecoin. Namecoin. https://www.namecoin.org/. Ιανουάριος 2022.
[26] Portion.io. The History of NFTs & How They Got Started. Portionio Blog. Ιούλιος 2021.
[27] CryptoKitties. CryptoKitties | Collect and breed digital cats! CryptoKitties. https://www.cryptokitties.co. 2021.
[28] Christie's. Beeple (b. 1981), EVERYDAYS: THE FIRST 5000 DAYS. Christie's. Νοέμβριος 2021.
[29] Howcroft, E. Twitter boss Jack Dorsey's first tweet sold for $2.9 million as an NFT. Reuters. Μάρτιος 2021.
[30] Newsham, J. A law professor made $65,000 selling NFTs of papers he writes in his bathtub. Here's how he set his prices and what he's doing with the money. Business Insider. Οκτώβριος 2021.
[31] Foo, T. (n.d.). How Do NFT Copyrights Work? Mintable. Αύγουστος 2021.
[32] Gibson, J. The thousand-and-second tale of NFTs, as foretold by Edgar Allan Poe. Queen Mary Journal of Intellectual Property, τ. 11, αρ. 3,σ. 249–269, 2021.
[33] Brekke, J. K. και Fischer, A. Digital scarcity. Internet Policy Review, τ. 10, αρ. 2, 2021.
[34] Department of Justice. Two defendants charged in Non-Fungible Token ("NFT") fraud and money laundering scheme. Department of Justice U.S. Attorney's Office Southern District of New York. 2022.
[35] C-163/18 Tom Kabinet, ECLI:EU:C:2019:1111. European Court of Justice, 2019.
[36] Bodó, B., Gervais, D. και Quintais, J. P. Blockchain and smart contracts: The missing link in copyright licensing? International Journal of Law and Information Technology, τ. 26, αρ. 4,σ. 311–336, 2018.
[37] Guadamuz, A. The treachery of images: Non-fungible tokens and copyright. Journal of Intellectual Property Law & Practice, τ. 16, αρ. 12, σ. 1367–1385, 2021.
[38] Γκούβερης, Α. (n.d.). ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΜΗΧΑΝΙΣΜΟΙ ΣΥΝΑΙΝΕΣΗΣ ΚΡΥΠΤΟΝΟΜΙΣΜΑΤΩΝ – Cryptonomisma. https://cryptonomisma.com/%CE%B1%CE%BB%CE%B3%CF%8C%CF%81%CE%B9%CE%B8%CE%BC%CE%BF%CE%B9-%CE%BA%CE%B1%CE%B9-%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CF%83%CE%BC%CE%BF%CE%AF-%CF%83%CF%85%CE%BD%CE%B1%CE%AF%CE%BD%CE%B5%CF%83%CE%B7/
[39] Proof of Stake • Greepto. (2024, February 19). Greepto. https://greepto.gr/education/docs/blockchain/proof-of-work-vs-proof-of-stake/
[40] BaseCoin.Gr, & Αραμπατζής, Α. (2021, May 14). Τι είναι το Proof of Burn – Πλεονεκτήματα / Μειονεκτήματα. Basecoin. https://www.basecoin.gr/ti-einai-to-proof-of-burn-pleonektimata-meionektimata/
[41] Frankenfield, J. (2023, August 18). Proof of Elapsed Time (POET) Definition, Purposes, vs. POW. Investopedia. https://www.investopedia.com/terms/p/proof-elapsed-time-cryptocurrency.a

[42] Academy, B. (2023, August 17). Byzantine fault tolerance explained. Binance Academy. https://academy.binance.com/en/articles/byzantine-fault-tolerance-explained

[43] Puthal, D., & Mohanty, S. P. (2019). Proof of authentication: IoT-Friendly blockchains. IEEE Potentials, 38(1), 26–29. https://doi.org/10.1109/mpot.2018.2850541

[44] https://ldapwiki.com/wiki/Proof-of-Possession

[45] https://www.techopedia.com/definition/33599/proof-of-importance-poi

[46] Academy, B. (2023b, August 17). What are nodes? Binance Academy. https://academy.binance.com/en/articles/what-are-nodes

[47] CoinMarketCap. (2021, August 21). What is a node? CoinMarketCap Academy. https://coinmarketcap.com/academy/article/what-is-a-node#toc-what-is-a-blockchain-node-

[48] Frankenfield, J. (2024, February 24). What is bitcoin? How to mine, buy, and use it. Investopedia. https://www.investopedia.com/terms/b/bitcoin.asp

[49] (2018). Dylan Yaga Peter Mell, Nik Roby, Karen Scarfone, ‘‘NISTIR 8202 Blockchain

[50] Technology Overview’’, U.S. Department of Commerce

[51] “Technically, ‘Token’ Is Just Another Word for ‘Cryptocurrency’ or ‘Cryptoasset.’ But Increasingly It Has Taken on a Couple of More Specific Meanings Depending on Context.” https://www.coinbase.com/learn/crypto-basics/what-is-a-token.

[52] “Tokens” https://docs.openzeppelin.com/contracts/2.x/tokens

[53] “Consensys Tokens” https://github.com/Consensys/Tokens

[54] “ERC-20 Token Standard.” https://ethereum.org/developers/docs/standards/tokens/erc-20.

[55] “ERC-721: Non-Fungible Token Standard.” https://eips.ethereum.org/EIPS/eip-721.

[56] “ERC-1155: Multi Token Standard.” https://eips.ethereum.org/EIPS/eip-1155.

[57] ERC1155 - OpenZeppelin Docs.” https://docs.openzeppelin.com/contracts/3.x/erc1155.

[58] “What Is a SoulBound Token? Ledger.” https://www.ledger.com/academy/topics/blockchain/what-is-a-soulbound-token.

[59] “The Ethereum Virtual Machine” https://cypherpunks-core.github.io/ethereumbook/13evm.html

[60] “Ethereum Virtual Machine (EVM).” https://ethereum.org/developers/docs/evm.

[61] “Introduction to Dapps.” https://ethereum.org/developers/docs/dapps

[62] Hono contributors, "*Hono: A fast, minimalist web framework for Node.js,*" 2023. [Online]. Διαθέσιμο: https://github.com/honojs/hono. [Accessed: 27/11/2023].

[63] E.G. Weyl, P. Ohlhaver, and V. Buterin, "*Decentralized Society: Finding Web3's Soul*," SSRN Electronic Journal, May 10, 2022. [Online]. διαθέσιμο: SSRN: https://ssrn.com/abstract=4105763. [Accessed: 27/11/2023].

[64] OpenZeppelin. (n.d.). OpenZeppelin. [Online]. Διαθέσιμο: https://www.openzeppelin.com/. [Accessed: 27/11/2023].

[65] Ethereum Foundation. (n.d.). "*EIP-721: ERC-721 Non-Fungible Token Standard*," Ethereum Improvement Proposals. [Online]. Διαθέσιμο: https://eips.ethereum.org/EIPS/eip-721. [Accessed: 27/11/2023].

[66] OpenZeppelin. (n.d.). "*ERC721URIStorage,"* OpenZeppelin Docs. [Online]. Διαθέσιμο: https://docs.openzeppelin.com/contracts/4.x/api/token/erc721#ERC721URIStorage. [Accessed: 27/11/2023].

[67] OpenZeppelin. (n.d.). "*Ownable*," OpenZeppelin Docs. [Online]. Διαθέσιμο: https://docs.openzeppelin.com/contracts/4.x/access-control. [Accessed: 27/11/2023].

[68] OpenZeppelin. (n.d.). "*Counters*," OpenZeppelin Docs. [Online]. Διαθέσιμο: https://docs.openzeppelin.com/contracts/4.x/utilities#misc. [Accessed: 27/11/2023].

[69]] Mozilla Developer Network. (n.d.). "*Cross-Origin Resource Sharing (CORS) - HTTP*," Mozilla Developer Network. [Online]. Διαθέσιμο: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS. [Accessed: 27/11/2023].

[70] Mozilla Developer Network. (n.d.). "*An overview of HTTP*, " Mozilla Developer Network. [Online]. Διαθέσιμο: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview. [Accessed: 27/11/2023].

[71] Mozilla Developer Network. (n.d.). "*The HTTP POST Method*," Mozilla Developer Network. [Online]. Διαθέσιμο: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST. [Accessed: 27/11/2023].

[72] OpenJS Foundation. (n.d.). "*Modules: CommonJS modules*," Node.js v21.2.0 Documentation, Node.js. [Online]. Διαθέσιμο: https://nodejs.org/api/modules.html#modules-commonjs-modules. [Accessed: 27/11/2023].

[73] Mozilla Developer Network. (n.d.). "*Understanding client-side JavaScript frameworks*," Mozilla Developer Network. [Online]. Διαθέσιμο: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks. [Accessed: 27/11/2023].