Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ethereum και Αποκεντρωμένες εφαρμογές – Χτίζοντας το μέλλον** **Ethereum Dapps – Building the Future**
Ονοματεπώνυμο Φοιτητή	**ΚΩΝΣΤΑΝΤΙΝΑ ΓΑΒΡΙΗΛΙΔΟΥ**
Πατρώνυμο	**ΗΛΙΑΣ**
Αριθμός Μητρώου	**ΜΠΠΛ / 17006**
Επιβλέπων	**ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΤΣΑΚΗΣ**

Ημερομηνία Παράδοσης **14 ΔΕΚΕΜΒΡΙΟΥ 2019**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Κωνσταντίνος Πατσάκης Επίκουρος Καθηγητής	Γεώργιος Τσιχριντζής Καθηγητής	Ευθύμιος Αλέπης Αν. Καθηγητής

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Πατσάκη Κωνσταντίνο για την εμπιστοσύνη που μου έδειξε με την ανάθεση της εργασίας αλλά και την υποστήριξή του στις στιγμές που χρειάστηκα την συνδρομή του.

Επίσης θα ήθελα να εκφράσω την ευγνωμοσύνη μου σε όλους όσους με στήριξαν κατά την διάρκεια της εκπόνησης της πτυχιακής μου εργασίας, που δεν είναι άλλοι από την οικογένεια μου, τον σύζυγό μου Γιάννη και τα παιδιά μου Γεωργία και Παναγιώτη.

Τους ευχαριστώ ολόψυχα για την κατανόηση και την υπομονή τους και τους αφιερώνω με αγάπη το πόνημά μου.

Περιεχόμενα

Περίληψη

Οι τεχνολογικές εξελίξεις μεταμορφώνονται και συνεχώς αναπτύσσονται με τόσο γρήγορο ρυθμό, ώστε όλοι να πρέπει να παραμείνουν σε εγρήγορση για να μπορούν να τις παρακολουθήσουν.

Το Blockchain είναι το νέο κύμα καινοτομίας που έχει ήδη αρχίσει να επανασχεδιάζει τον τρόπο εργασίας, τις κοινωνικές και πολιτικές αλληλεπιδράσεις και τους παραδοσιακούς τρόπους ανταλλαγής. Ωστόσο δεν είναι μόνο μια απλή αλλαγή, αλλά ένα ραγδαίως αναπτυσσόμενο φαινόμενο που ήδη βρίσκεται σε εξέλιξη. Κατά τη συγγραφή αυτή, περισσότερα από 40 κορυφαία χρηματοπιστωτικά ιδρύματα και πολλές διαφορετικές επιχειρήσεις σε διάφορες βιομηχανίες, άρχισαν να διερευνούν τις δυνατότητες του blockchain με σκοπό να μειώσουν το κόστος των συναλλαγών, να επιταχύνουν το χρόνο περάτωσής τους, να μειώσουν τον κίνδυνο απάτης και να εξαλείψουν τους μεσάζοντες ή τις υπηρεσίες διαμεσολάβησης.

Η τεχνολογική ιδέα πίσω από το Blockchain είναι πολύ πανομοιότυπη με αυτή μιας βάσης δεδομένων και μέσα από αυτήν την προσέγγιση, γίνεται η χρήση του στην ανάπτυξη της voting εφαρμογής, αποδεικνύοντας με ρεαλιστικό τρόπο ότι η αλυσίδα του Ethereum μπορεί να αναλάβει ηγετικό ρόλο στη νέα εποχή, που σκοπό έχει τη δημιουργία αποκεντρωμένων εφαρμογών.

Abstract

Technological advancements and innovation is constantly evolving and growing at such a fast rate that everyone is required to stay abreast of these advancements and innovations.

Blockchain is the new wave of disruption that has already started to redesign business, social and political interactions, and any other way of value exchange. Again, it is not just a change, but a rapid phenomenon that is already in motion. As of this writing, more than 40 top financial institutions and many different firms across industries have started to explore blockchain to lower transaction cost, speed up transaction time, reduce the risk of fraud, and eliminate the middleman or intermediary services.

The technological concept behind the Blockchain is interestingly closely identical to that of a database. The scope of this deliverable is to offer a realistic proof of concept that Ethereum blockchain can take the role of a secure back-end that boosts-up the creation of decentralized applications.

Εισαγωγή

Ο ενθουσιασμός που επικρατεί για τη νέα τεχνολογία, τα πλεονεκτήματα και τις δυνατότητες εφαρμογής της κάνουν πολλούς να μιλούν για επανάσταση αντίστοιχη με εκείνη του διαδικτύου, η οποία μέσα στα επόμενα χρόνια θα αλλάξει ριζικά τις δομές, τον τρόπο οργάνωσης και τη λειτουργία των σύγχρονων κοινωνιών. Το blockchain αποτελεί αναμφίβολα μια μεγαλοφυή εφεύρεση – το πνευματικό παιδί ενός ατόμου ή μιας ομάδας ατόμων γνωστοί με το ψευδώνυμο, Satoshi Nakamoto. [1]

Ήδη οι εφαρμογές της τεχνολογίας blockchain καλύπτουν όλα σχεδόν τα πεδία της οικονομίας, ενώ ολοένα και περισσότερες εταιρείες, οργανισμοί και δημόσιες αρχές επενδύουν σημαντικούς πόρους και εφαρμόζουν πιλοτικά τη νέα τεχνολογία.

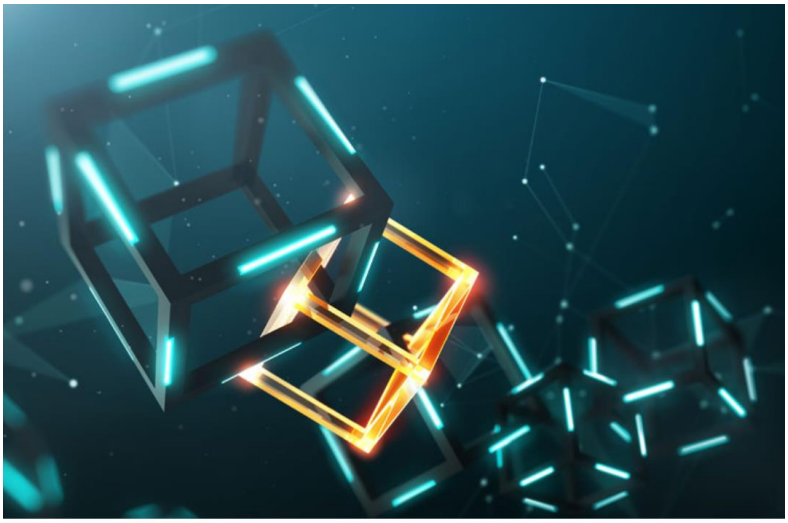
Πάνω σε αυτήν την κατεύθυνση, βοηθάει και η ύπαρξη του κατάλληλου νομοθετικού πλαισίου, καθώς τον Φεβρουάριο του 2018, η Ευρωπαϊκή Επιτροπή ανακοίνωσε [2] τη σύσταση παρατηρητηρίου και forum για την blockchain με σκοπό την παρακολούθηση των εξελίξεων και την προώθησή της νέας τεχνολογίας [3], δίνοντας έτσι και μια επίσημη αναγνώριση της δυναμικής του νέου «trend» και των λύσεων που αυτό αναμένεται να προσφέρει.

Τεχνολογία Blockchain

Τι είναι όμως τελικά το Blockchain;

Η τεχνολογία blockchain αποτελεί μια νέα τεχνολογία - αν και μετράει ήδη 10 χρόνια παρουσίας - η οποία συμπεριφέρεται ως μία δημόσια, μη δυνατόν να τροποποιηθεί το ιστορικό της, κατανεμημένη σειρά δεδομένων, ομαδοποιημένων σε χρονικά αριθμημένα «τμήματα» ή «συστοιχίες» τα επονομαζόμενα *blocks*. [4] θα μπορούσε να χαρακτηριστεί ως ένα δημόσιο ψηφιακό λογιστικό βιβλίο, στο οποίο καταγράφονται όλες οι συναλλαγές που έχουν εκτελεστεί από όλους τους χρήστες, καταχωρημένες σε ένα συγκεκριμένο μοτίβο δημιουργώντας έτσι ένα μητρώο δεδομένων και πληροφοριών (ledger).

Με απλά λόγια, θα μπορούσαμε να πούμε ότι, το blockchain απαρτίζεται από κόμβους (blocks) που συνδέονται μεταξύ τους, δημιουργώντας έτσι μια εικονική αλυσίδα (chain).



Εικόνα 1 https://www.ledgerinsights.com/ethereum-first-token-taxonomy-framework/

Η θεμελιώδης διαφορά από τα υφιστάμενα μητρώα και βάσεις δεδομένων είναι ότι για την τήρησή του δεν είναι αρμόδια μία κεντρική αρχή, αλλά οι λεγόμενοι κόμβοι – nodes, δηλαδή όλοι οι χρήστες του blockchain οι οποίοι, έχοντας εγκαταστήσει το απαιτούμενο λογισμικό, ενημερώνουν, ταυτόχρονα, το μητρώο για τις αλλαγές σε αυτό, ώστε ανά πάσα στιγμή όλοι να έχουν την ίδια ακριβώς την ίδια κατάσταση μητρώου [5].

Αυτό σημαίνει ότι η κατάρρευση οποιουδήποτε κόμβου, δεν προκαλεί την κατάρρευση του συστήματος, καθώς αντίγραφο του μητρώου, φυλάσσεται σε όλους τους κόμβους που συμμετέχουν στην αλυσίδα.

Η τεχνολογία Blockchain εισήχθη χρονικά κατά την κρίση του χρηματοπιστωτικού συστήματος το 2008 [6], για να υποστηρίξει έναν νέο τύπο ψηφιακού νομίσματος του γνωστού σε όλους μας, Bitcoin, προκειμένου να χρησιμοποιηθεί για την δημιουργία ασφαλών συναλλαγών σε ένα «peer-to-peer» δίκτυο αλλά σε ένα περιβάλλον χωρίς εξασφαλισμένη εμπιστοσύνη (trustless environment) μεταξύ των εμπλεκόμενων μερών (συμμετεχόντων).

Πριν την εμφάνιση της τεχνολογίας Blockchain, οι παραδοσιακές συναλλαγές απαιτούσαν την παρουσία ενός ενδιάμεσου φορέα (μεσάζων) που να λειτουργεί ως φορέας εμπιστοσύνης μεταξύ των εμπλεκόμενων μερών.

Παραδείγματος χάριν εάν η συναλλαγή αφορούσε σε μεταφορά χρημάτων μεταξύ δυο προσώπων A (Alice) και B (Bob), η διεξαγωγή και εκκαθάριση της συναλλαγής προϋπόθετε την παρουσία ενδιάμεσων χρηματοπιστωτικών ιδρυμάτων (Τράπεζες) που θα αναλάμβαναν την μεταφορά του ποσού μεταφοράς / κεφαλαίων μεταξύ των ενδιαφερόμενων μερών.

Εικόνα 2 https://www.cbinsights.com/research/what-is-blockchain-technology/

Με την είσοδο της τεχνολογίας blockchain δημιουργήθηκε μια νέα προσέγγιση.

Αντί για παράδειγμα η Τράπεζα μέσω του κεντρικού της συστήματος να επιβεβαιώνει τη μεταφορά χρημάτων από τον Α στον Β, η επαλήθευση αυτή επιτυγχάνεται από τους κόμβους (χρήστες) που συμμετέχουν στο δίκτυο, με την τήρηση και ταυτόχρονη ενημέρωση του μητρώου από όλους.

Έτσι με την επίτευξη συμφωνίας (consensus) ανάμεσα στους κόμβους, δημιουργείται εμπιστοσύνη για την ορθότητα των στοιχείων που καταχωρούνται στο μητρώο και το σημαντικότερο, όσο μεγαλύτερος είναι ο αριθμός των κόμβων που συμμετέχουν και τηρούν το μητρώο, τόσο μεγαλύτερος και ο βαθμός εμπιστοσύνης και ουδετερότητας επιτυγχάνεται.

Συνεπώς, το μητρώο σε μία πλατφόρμα blockchain δεν είναι απλά αποκεντρωμένο (decentralized) αλλά και διανεμημένο (distributed) με την έννοια ότι ολόκληρο το μητρώο συναλλαγών τηρείται από όλους τους κόμβους (χρήστες) και συγχρονίζεται ταυτόχρονα, ώστε όλοι οι κόμβοι να έχουν το ίδιο ενημερωμένο μητρώο (sync).

Εικόνα 3 https://www.cbinsights.com/research/what-is-blockchain-technology/

Τα μπλοκ αυτά είναι κρυπτογραφημένα με τον αλγόριθμό SHA-256 secure hash algorithm, ο οποίος κρυπτογραφεί μαθηματικά και κατακερματίζει την πληροφορία του μπλοκ με τρόπο ο οποίος να μην μπορεί να είναι αναστρέψιμος (one way encryption), πάντα με ένα σταθερό αποτέλεσμα είτε τα δεδομένα που καταχωρήσαμε είναι ένας χαρακτήρας είτε το κείμενο από ένα ολόκληρο βιβλίο.

Δείτε την σχηματική [7] απεικόνιση τμήματος ενός blockchain παρακάτω.

Εικόνα 4

Με αυτόν τον τρόπο επιτυγχάνεται η διανομή της πληροφορίας και η ενημέρωση όλων των κόμβων (nodes) με το τελευταίο αντίγραφο των μπλοκ καθώς επίσης και η ασφάλεια των δεδομένων από αλλαγές και αμφισβητήσεις.

Κάθε μπλοκ συνδέεται κρυπτογραφικά και υπογράφεται ψηφιακά από κάθε κόμβο με το προηγούμενο του και μια οποιαδήποτε προσπάθεια αλλαγής των δεδομένων ενός μπλοκ θα ήταν αδύνατη καθώς δεν θα μπορούσε να επιβεβαιωθεί κρυπτογραφικά από κανένα κόμβο στο σύνολο τους.

Επιβεβαίωση Αμεταβλητότητας Δεδομένων

Ας υποθέσουμε λοιπόν ότι από μια αλυσίδα μπλοκ δεδομένων η οποία αποτελείται από 10.000 μπλοκ, αφαιρέσουμε το μπλοκ Νο 345 θέλοντας να αρνηθούμε την ύπαρξη μιας συναλλαγής ή αλλάξουμε τα δεδομένα συναλλαγών ενός μπλοκ.

Τότε θα συμβούν τα παρακάτω:

1. Αλλοίωση της ταυτότητας του μπλοκ

Δεν θα μπορέσει να επαληθευτεί η αλυσίδα καθώς θα υπάρχει ασυμφωνία και οι συναλλαγές στο μπλοκ δεν θα ταιριάζουν με την ταυτότητα του μπλοκ (θυμηθείτε κάθε μπλοκ χαρακτηρίζεται από ένα hash που το καθιστά μοναδικό).

2. Αποτυχία επιβεβαίωσης σειράς της αλυσίδας

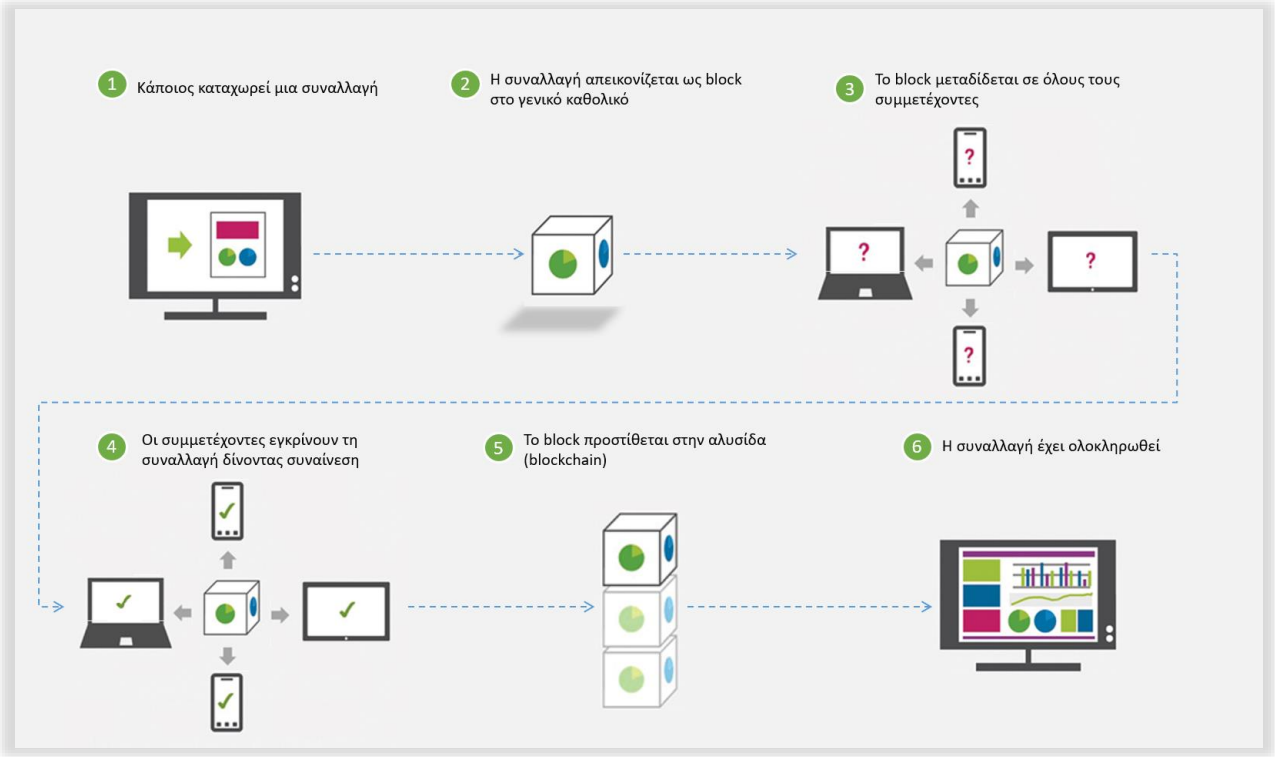
Θυμηθείτε ότι υπάρχει αλληλουχία μεταξύ των μπλοκ δεδομένων σε μία αλυσίδα. Έτσι λοιπόν αν λείπει ένα μπλοκ τότε η αλληλουχία αυτή δεν υφίσταται και θα πρέπει να γίνει υπολογισμός ξανά των ταυτοτήτων των μπλοκ ολόκληρης της αλυσίδας ώστε τα μπλοκ να συνδέονται με σωστή αλληλουχία. Αυτή η διαδικασία είναι εξαιρετικά δύσκολη, χρονοβόρα ενεργοβόρα και κοστοβόρα, γεγονός που την καθιστά πρακτικά αδύνατη να γίνει. Επιπροσθέτως θα πρέπει να γίνει και η ενημέρωση όλων των κόμβων που

διαμοιράζουν την λίστα με τις συναλλαγές καθώς και να γίνει ψηφιακή πιστοποίηση των συναλλαγών από όλα τα μέρη.

Πώς λειτουργεί το Blockchain

Στη γλώσσα των υπολογιστών, το blockchain ανήκει στην κατηγορία δικτύων υπό τον τίτλο «δίκτυο ομότιμων κόμβων». Το «ομότιμος» σημαίνει εδώ ότι δεν υπάρχει κάποιο πρόσωπο του δικτύου που να υπερέχει έναντι κάποιου άλλου καθ' οποιονδήποτε τρόπο. Μπορεί τα πρόσωπα που συμμετέχουν στο δίκτυο να μην είναι ίδια, αλλά είναι ίσα μεταξύ τους αναφορικά με οποιαδήποτε διαδικασία εκλογής ή/και επιλογής μεταξύ αυτών. Όλα τα πρόσωπα του δικτύου blockchain, δημιουργούν και μοιράζονται από κοινού ένα πρώτο αρχείο. Η διαδικασία δημιουργίας και διαφύλαξης του αρχείου αυτού καθορίζεται και ελέγχεται από ένα «Σύνταγμα», που ονομάζεται πρωτόκολλο συναίνεσης.

Αλλά ας πάρουμε τα πράγματα από την αρχή, από το μπλοκ της… γένεσης καθώς όλα ξεκινούν από εδώ, από την πρώτη συναλλαγή, το λεγόμενο «genesis block» [8].



Εικόνα 5

Πάνω σε αυτό το πρώτο μπλοκ 1 προστίθενται από τους «miners» σε γραμμική, χρονολογική σειρά όλες οι επόμενες σειρές με μια λογική παζλ. Δηλαδή κάθε «μπλοκ» συνδέεται άρρηκτα με το προηγούμενο, ώστε να μη μπορεί να παρεμβληθεί ένα ενδιάμεσο και η αλυσίδα να μη μπορεί να μεταβληθεί.

Η αλυσίδα αυτή αποτελεί κοινή και καθολικά αποδεκτή βάση δεδομένων (θα δούμε σε επόμενο κεφάλαιο την διαφορά μεταξύ συμβατικής βάσης), που εξυπηρετείται από χιλιάδες ηλεκτρονικούς υπολογιστές ταυτόχρονα, ώστε να είναι άμεσα προσβάσιμη σε όλους στο Διαδίκτυο την ίδια χρονική στιγμή. Αυτός ο τρόπος λειτουργίας, δηλαδή το ότι δεν υπάρχει αποθήκευση σε μόνο μια κεντρική τοποθεσία, καθιστά τις κοινοποιήσεις δημόσιες, επαληθεύσιμες και μέχρι σήμερα αναλλοίωτες από κυβερνοεπιθέσεις.

Οι «σελίδες» του γράφονται με την επίλυση «γρίφων» από ανθρώπους πίσω από χιλιάδες ηλεκτρονικούς υπολογιστές, τους λεγόμενους «miners» (εξορύχους), ενώ για όσους καταφέρνουν να «ταιριάξουν» ένα κομμάτι του παζλ, υπάρχει οικονομικό κίνητρο, το οποίο αποτιμάται σε κρυπτονόμισμα (Ether).

Όταν η αλυσίδα ενημερώνεται με νέες κοινοποιήσεις, κάθε χρήστης που χρησιμοποιεί την πλατφόρμα πρέπει να συμφωνεί ότι η κοινοποίηση είναι έγκυρη. Άτομα τα οποία αναρτούν λανθασμένες πληροφορίες στις κοινοποιήσεις έχουν οικονομικές συνέπειες και αν «πιαστούν», η πιθανότητα επιλογής τους στο μέλλον μειώνεται ραγδαία. Με αυτόν τον τρόπο διασφαλίζεται ότι οι κοινοποιήσεις περιέχουν έγκυρες και ορθές πληροφορίες. Ο χρήστης μπορεί να έχει πρόσβαση μέσω ενός μοναδικού στοιχείου πρόσβασης (ας το σκεφτούμε ως κλειδί) που του έχει διατεθεί αρχικά. Εάν χαθεί το στοιχείο πρόσβασης, ο χρήστης δεν μπορεί να συνδεθεί καθώς δεν υπάρχει μηχανισμός αναπλήρωσης του στοιχείου.

Κατά τη διαδικασία της εξόρυξης (mining), υπολογιστές σε όλο το κόσμο ανταγωνίζονται πάνω στην πραγματοποίηση υπολογισμών, για τη λύση των οποίων αμείβονται με κρυπτονομίσματα. Αν μερικοί υπολογιστές μπορούν να κάνουν τους υπολογισμούς πιο γρήγορα, τότε μπορούν να δοκιμάσουν περισσότερες φορές την τύχη τους στο ίδιο χρονικό διάστημα.

Οι miners επιχειρούν να λύσουν αυτούς τους γρίφους χρησιμοποιώντας ειδικό λογισμικό, ώστε να εξασφαλίσουν την αμοιβή τους. Η δε πρόκληση θα μπορούσε να συγκριθεί με τις ολοένα δυσκολότερες «πίστες» που συναντά κάποιος σε ένα βιντεοπαιχνίδι, αφού το δίκτυο αυτόματα αλλάζει το επίπεδο δυσκολίας των μαθηματικών προβλημάτων, αναλόγως τού πόσο γρήγορα αυτά λύνονται με την πάροδο του χρόνου.

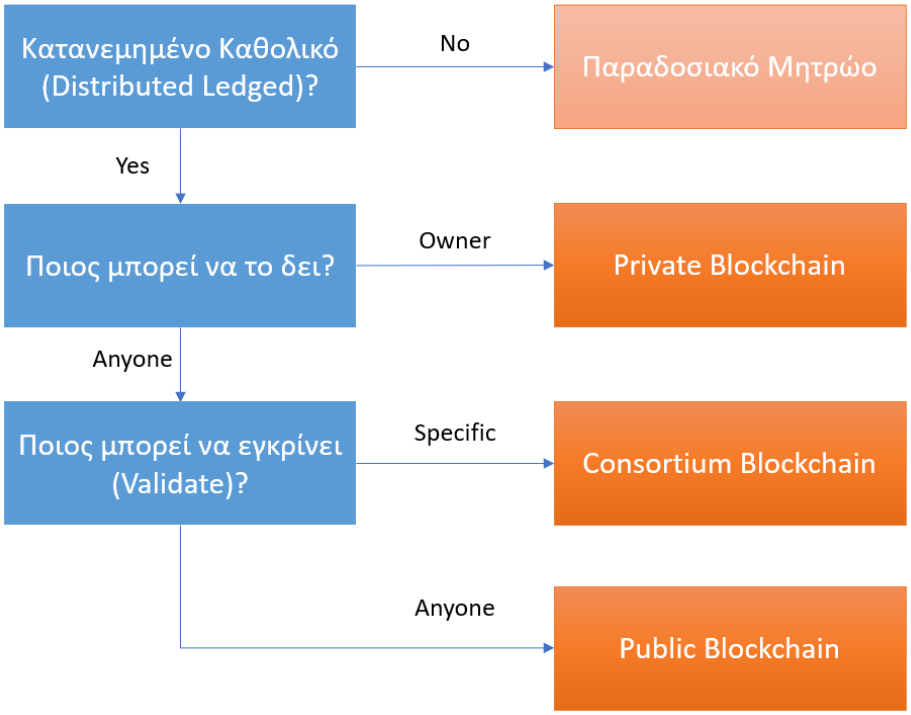
Το mining απαιτεί αυξημένη υπολογιστική ισχύ, η οποία καθορίζεται από την κάρτα γραφικών του κάθε Miner (GPU). Απαιτεί όμως και αυξημένη κατανάλωση ηλεκτρικού ρεύματος. Κάποτε η εξόρυξη γινόταν μέσω επεξεργαστών (CPU). Πλέον, τουλάχιστον στην περίπτωση του Bitcoin, η μέθοδος αυτή θεωρείται μάλλον ατελέσφορη στις περισσότερες περιπτώσεις, αφού οι κάρτες γραφικών προσφέρουν πολύ μεγαλύτερη ισχύ, λύνοντας τους αλγόριθμους σημαντικά ταχύτερα. Άρα, όποιος θέλει να ασχοληθεί με το Mining, καλό είναι να «επενδύσει» σε μια δυνατή κάρτα γραφικών, αλλά και να γνωρίζει ότι ένα αποδοτικό σύστημα εξόρυξης κρυπτονομισμάτων (mining rig) χρειάζεται υψηλής ποιότητας λογισμικό (hardware), αλλά και όσο το δυνατόν φθηνότερο ηλεκτρικό ρεύμα, ώστε να μην καταλήξει ασύμφορο…

Ανεξάρτητα όμως από την συγκεκριμένη διαπίστωση, με την είσοδο της τεχνολογίας blockchain δημιουργήθηκε ο κορμός για ένα νέο είδους δικτύου που δίνει τη δυνατότητα σε ψηφιακές πληροφορίες να διανέμονται αλλά να μην αντιγράφονται. Αν και αρχικά προορίστηκε για τα ψηφιακά νομίσματα, Bitcoin, τώρα η νέα αυτή τεχνολογική οντότητα έχει περισσότερες χρήσεις στην τεχνολογία.

Είδη Blockchain

Το Blockchain δημιουργήθηκε για να είναι δημόσιας χρήσης, στην πορεία, επιχειρήσεις και οι κυβερνητικοί οργανισμοί, θέλησαν να εκμεταλλευτούν τα πλεονεκτήματά του ορίζοντας όμως οι ίδιοι ποιος θα είναι ο validator.

Έτσι προέκυψε η ανάγκη δημιουργίας blockchain με ιδιωτική διαβάθμιση.



Εικόνα 6

Στην τρέχουσα φάση, αριθμούνται τρεις τύποι blockchain και είναι οι ακόλουθοι [9]:

Δημόσιοι Κατάλογοι Blockchains (Public Blockchains)

Είναι προσβάσιμα σε οποιονδήποτε «χρήστη».

Κάθε πρόσωπο μπορεί να πραγματοποιήσει μια συναλλαγή, να συμμετάσχει στη διαδικασία επικύρωσης των «μπλοκ» ή να αποκτήσει ένα αντίγραφο του Blockchain.

Άρα σε ένα δημόσιο κατάλογο:

1. Οποιοσδήποτε μπορεί να γράψει χωρίς άδεια από κάποια αρχή
2. Οποιοσδήποτε μπορεί να διαβάσει

Παράδειγμα: Ένα παράδειγμα public blockchain, είναι το κρυπτονόμισμα Bitcoin

Κοινοπρακτικοί Κατάλογοι Blockchains (Consortium Blockchains)

Είναι προσβάσιμα σε οποιονδήποτε χρήστη για «read» όμως σε συγκεκριμένες ομάδες για την εκτέλεση και επιβεβαίωση των συναλλαγών. Με αυτόν τον τύπο, επιτρέπεται η διαφάνεια των συναλλαγών

προς τρίτους (εποπτικές αρχές, επενδυτές κλπ) αλλά η διαδικασία της καταγραφής πάνω στο block ορίζεται από συγκεκριμένα άτομα.

Άρα σε ένα κοινοπρακτικό κατάλογο:

1. Συγκεκριμένες ομάδες / χρήστες μεταξύ των οργανισμών που συμμετέχουν μπορούν να εκτελέσουν συναλλαγές
2. Οποιοσδήποτε (εντός και εκτός οργανισμού) μπορεί να διαβάσει

Παράδειγμα: κοινοπραξία 10 τραπεζών που για την κατοχύρωση απόφασης απαιτείται η συγκατάθεση τουλάχιστον των 8 από τις 10 που συμμετέχουν.

Ιδιωτικοί Κατάλογοι Blockchains (Private Blockchains)

Βρίσκονται υπό τον έλεγχο ενός «χρήστη» ο οποίος αποκλειστικά διασφαλίζει τον έλεγχο της συμμετοχής και της επικύρωσης.

Εδώ όλοι οι συμμετέχοντες στην αλυσίδα είναι γνωστοί και έμπιστοι και οι αλυσίδες συνήθως είναι πιο μικρές και γρήγορες ενώ το proof of work σχετικά πιο απλό ανάλογα με την επιχειρηματική απόφαση.

Άρα σε έναν ιδιωτικό κατάλογο:

1. Συγκεκριμένη ομάδα / χρήστες εντός του οργανισμού μπορούν να εκτελέσουν συναλλαγές
2. Οποιοσδήποτε (εντός και εκτός οργανισμού) μπορεί να διαβάσει

Παράδειγμα: παραγγελίες εσωτερικά σε μια εταιρεία ή ένα οργανισμό όπως μια Τράπεζα.

Blockchain key points

- Το Blockchain είναι ένα «peer-to-peer» σύστημα για την υποστήριξη της δημιουργίας συναλλαγών σε ένα περιβάλλον χωρίς αμοιβαία εμπιστοσύνη [10]
- Αποτελεί ένα ανοικτό, αποκεντρωμένο και ανοικτό μητρώο συναλλαγών, το οποίο αντιγράφεται και τηρείται σε μεγάλο αριθμό κόμβων, δέχεται καταχωρήσεις (append-only) και δεν επιτρέπει την τροποποίηση των υφιστάμενων καταχωρήσεων, παραμένει ουσιαστικά αμετάβλητο
- Οποιαδήποτε νέα συναλλαγή ενημερώνεται ταυτόχρονα σε όλα τα αντίγραφα του μητρώου που φυλάσσονται στους κόμβους. Η διαδικασία αυτή είναι πιο πολύπλοκη από τον παραδοσιακό τρόπο Client-server αλλά συνάμα και πιο ασφαλής καθώς αν ένας κόμβος χαθεί το σύστημα θα συνεχίσει να λειτουργεί. Όσο πιο πολλοί κόμβοι τόσο πιο ασφαλές το δίκτυο αλλά και ταυτόχρονα και πιο αργό
- Κάθε κόμβος δεδομένων κρυπτογραφείται με τον αλγόριθμο SHA-256 και περιέχει εκτός των δεδομένων και την πληροφορία hash του προηγούμενου μπλοκ δεδομένων. Με αυτόν τον τρόπο πραγματοποιείται η σύνδεση των κόμβων από τον αρχικό κόμβο (genesis block) μέχρι τον τελευταίο.
- Δεν προϋποθέτει την παρουσία ενδιάμεσων για την επιβεβαίωση, διασφάλιση και εκκαθάριση της συναλλαγής, καθώς όλες οι συναλλαγές επιβεβαιώνονται (ή απορρίπτονται) από τους συμμετέχοντες κόμβους
- Αποτελεί ένα επιπλέον επίπεδο (layer) στο διαδίκτυο (Internet) και μπορεί να συνυπάρχει και με άλλες τεχνολογίες διαδικτύου
- Όπως ακριβώς το πρωτόκολλο TCP/IP (1. Process/Application Layer 2. Host-to-Host/Transport Layer 3. Internet Layer 4. Network Access/Link Layer) δημιουργήθηκε για να υποστηρίξει ένα ανοικτό σύστημα, έτσι και το Blockchain δημιουργήθηκε για να προσφέρει αποκέντρωση εφαρμογών. Για αυτόν το λόγο και οι δημιουργοί του Bitcoin διέθεσαν τον κώδικά τους (open-sourced) προκειμένου να εμπνεύσουν την δημιουργία και άλλων αποκεντρωμένων εφαρμογών.

Σύγκριση με παραδοσιακές βάσεις δεδομένων

Αποθήκευση δεδομένων σε παραδοσιακές βάσεις δεδομένων

Οι παραδοσιακές βάσεις δεδομένων αποθηκεύουν τα δεδομένα τους σε μια κεντρική τοποθεσία και για να προσπελάσουμε ή να δημιουργήσουμε δεδομένα πρέπει να συνδεθούμε με την συγκεκριμένη τοποθεσία. Ανεξάρτητα από την αρχιτεκτονική της βάσης, ανάλογα με το επίπεδο πρόσβασης του χρήστη οι δυνατότητες που υπάρχουν ως προς τα δεδομένα της βάσης είναι η δημιουργία (**C**reate), η προσπέλαση (**R**ead), η ενημέρωση (**U**pdate) και η διαγραφή (**D**elete) και όλες αυτές οι δυνατότητες - λειτουργίες είναι γνωστές ως **CRUD** λειτουργίες.

Πιο αναλυτικά [11]:

Create: Προσθέτει νέο record στη βάση δεδομένων, συνοδευόμενο συνήθως και με πληροφορίες σχετικές με τον χρήστη καταχώρησης (identification).

Read: Δίνει την δυνατότητα προσπέλασης των δεδομένων. Η αναζήτηση των δεδομένων από την βάση γίνεται συνήθως βάσει κάποιου κλειδιού ή index.

Update: Τροποποιεί υφιστάμενα δεδομένα από τη βάση, δίνοντας τη δυνατότητα αποθήκευσής τους με διαφορετική πληροφορία.

Delete: Δίνει τη δυνατότητα οριστικής διαγραφής του δεδομένου από την βάση, με την εκτέλεση της συγκεκριμένης εντολής το δεδομένο παύει να υπάρχει και η ανάκτησή του δεν είναι εφικτή.

Στις παραδοσιακές βάσεις δεδομένων (ενδεικτικά Oracle, MSSQL Server, PostgreSQL, IBM DB2 κλπ) το κύριο πρόβλημα ήταν η ταυτόχρονη χρήση και επεξεργασία των δεδομένων από παραπάνω από έναν χρήστες που οδηγούσε στο κλασικό πρόβλημα συγχρονισμού (classic concurrency problem).

Παραδείγματος χάριν αν ο Α και ο Β επεξεργάζονταν ταυτόχρονα την ίδια πληροφορία, τότε στη βάση δεδομένων θα παραμείνει μόνο η εικόνα του χρήστη που έχει εκτελέσει τελευταίος την λειτουργία της ενημέρωσης (update) χάνοντας έτσι την ενημέρωση που προηγήθηκε χρονικά από τον Α.

Πάνω σε αυτό έχουν εφαρμοστεί διάφορες λύσεις για τα συστήματα διαχείρισης δεδομένων (Database management systems (DBMSs)) με κυριότερες τις κάτωθι:

- **Locking:** Σε αυτήν την προσέγγιση, η DBMS κλειδώνει το δεδομένο ή το σύνολο δεδομένων (lock) μέχρι ο χρήστης να ολοκληρώσει τις λειτουργίες του. Έτσι εφαρμόζει μια σειριακή εξυπηρέτηση των αιτημάτων τροποποίησης διατηρώντας στον χρήστη τοπικά αντίγραφα της πληροφορίας που τα αποδεσμεύει (unlock) στους υπόλοιπους χρήστες μόνο όταν ο αρχικός χρήστης - locker ολοκληρώσει τις λειτουργίες του. Αυτή η προσέγγιση ναι μεν λύνει το πρόβλημα του συγχρονισμού, οδηγεί όμως σε αναμονές για τους υπόλοιπους χρήστες καθώς η ολοκλήρωση του προηγούμενου αποτελεί προϋπόθεση για τις εργασίες του επόμενου.
- **Timestamp ordering:** όταν ο χρήστης θέλει να προσπελάσει (διαβάζει) τα δεδομένα η DBMS σημειώνει την χρονική στιγμή (timestamp) και τη συγκρίνει με το timestamp της συναλλαγής και «αποφασίζει» αν είναι ασφαλές να διαβάσει το δεδομένο. Σε αυτήν την προσέγγιση, αν πάμε να διαβάσουμε ένα δεδομένου το οποίο γίνεται update από άλλον χρήστη, υπάρχει πιθανότητα να οδηγηθούμε σε dead-end και αναγκαστικό τερματισμό της συναλλαγής που εμποδίζει την υλοποίηση user-friendly applications.
- **Optimistic concurrency control:** Οι δυο προηγούμενες προσεγγίσεις θεωρούν δεδομένο ότι θα συμβούν συγκρούσεις μεταξύ των ενημερώσεων ενώ αυτή η λύση θεωρεί ότι αν συμβούν θα είναι σπάνιες. Εδώ οι χρήστες μπορούν να διαβάζουν τα δεδομένα χωρίς πρόβλημα ή αναμονές και όταν ένας χρήστης επιχειρεί να εισάγει ένα νέο record, τότε το DBMS συγκρίνει πότε διαβάστηκε τελευταία το δεδομένο σε σχέση με το τρέχον αποθηκευμένο της βάσης και αν εντοπίσει διαφορά τότε καταλαβαίνει ότι έχει προκύψει ενημέρωση και η εγγραφή αποτυγχάνει. Αν το δεδομένο δεν έχει αλλάξει τότε η ενημέρωση ολοκληρώνεται με επιτυχία. Αυτή η τεχνική υποστηρίζει την κλιμακωτή σχεδίαση των εφαρμογών.

Οι παραδοσιακές βάσεις δεδομένων απλοποιούν το sharing των δεδομένων, την εκτέλεση των CRUD λειτουργιών και την διατήρηση της συνοχής των δεδομένων σε περιβάλλοντα υψηλών επιπέδων απόδοσης αν και σε επίπεδο εξασφάλισης του audit trail στις τροποποιήσεις των δεδομένων και του διαχειριστικού effort για την συντήρηση της υποδομής έχουν σημαντικά περιθώρια βελτίωσης. Ωστόσο το βασικό τους πλεονέκτημα είναι η απόδοση (performance) που έχουν, καθώς είναι σχεδιασμένα να χρησιμοποιούν store options (indexes) που μειώνουν σημαντικά τον χρόνο εντοπισμού και διαλογής των δεδομένων και για αυτόν τον λόγο επιλέγονται όταν απαιτείται μικρός χρόνος ανταπόκρισης και μεγάλη διακίνηση των δεδομένων (bulks).

Αποθήκευση δεδομένων στο blockchain

Το blockchain διαχειρίζεται διαφορετικά τα δεδομένα σε σχέση με παραδοσιακή βάση δεδομένων.

Μια από τις πρώτες και σημαντικότερες διαφορές είναι το γεγονός ότι δεν υποστηρίζει CRUD λειτουργίες. Οι μόνες λειτουργίες που υποστηρίζει είναι η Write που είναι η ίδια όπως η δημιουργία (Create) και η προσπέλαση (Read). Έτσι από τη στιγμή που τα δεδομένα τοποθετηθούν στο block και ενταχθούν στο blockchain γίνονται αμετάβλητα (immutable) καθώς το blockchain δεν υποστηρίζει τις λειτουργίες Update και Delete.

Επίσης μια ακόμα σημαντική διαφορά είναι η τοποθεσία αποθήκευσης, το φυσικό δηλαδή location της βάσης (database). Όπως αναφέρθηκε προηγουμένως όλες οι DBMS, ανεξαρτήτως αρχιτεκτονικής (n-tier, distributed processing, etc.) είναι κεντροποιημένες (centralized) δηλαδή βρίσκονται σε μοναδική τοποθεσία ενώ στο blockchain, ένα πλήρες αντίγραφο του διατηρείται σε κάθε κόμβο (node) και μάλιστα μια από τις δυσκολίες που έχει να αντιμετωπίσει ένα blockchain network είναι το να εξασφαλίσει ότι όλα τα blockchain nodes εμπεριέχουν τα ίδια δεδομένα. Η υλοποίηση του κάθε blockchain υιοθετεί αυστηρούς κανόνες για την διατήρηση και τον συγχρονισμό μέσα στο δίκτυο αλλά ταυτόχρονα αυτοί οι κανόνες βοηθούν στον εντοπισμό διαφορών μεταξύ των κόμβων εύκολα και γρήγορα.

Αυτή η ιδιαιτερότητα, στην διανεμημένη αποθήκευση, βοηθάει στο να θεωρείται αρκετά ευέλικτο καθώς τυχόν βλάβη σε έναν κόμβο έχει αμελητέα επίπτωση στο υπόλοιπο blockchain δίκτυο, από την άλλη όμως υστερεί σε θέματα απόδοσης (performance) στην αποθήκευση, καθώς δεν σχεδιάστηκε στο να υποστηρίζει γρήγορες προσπελάσεις δεδομένων κατά μήκος του block tree.

Στον πίνακα που ακολουθεί συνοψίζονται οι διαφορές μεταξύ της αποθήκευσης δεδομένων σε μια συμβατική βάση και στο blockchain [11].

Στοιχείο σύγκρισης	Παραδοσιακή βάση δεδομένων	Blockchain
Τοποθεσία (Location)	Υπάρχει online μια μοναδική έκδοση της βάσης (εκτός σεναρίου disaster server ή archiving)	Διαμοιρασμένη βάση δεδομένων. Ο κάθε κόμβος αποτελεί ένα πλήρες αντίγραφο του blockchain
Λειτουργίες (Operations)	Create, Read, Update, Delete (CRUD)	Read, Write
Απόδοση (Performance)	Έχουν σχεδιαστεί με άξονα την απόδοση καθώς έχουν εφαρμοστεί βελτιστοποίησεις προκειμένου να εξασφαλίζεται μικρός χρόνος ανταπόκρισης και μεγάλη διακίνηση των δεδομένων (bulks)	Δεν έχει σχεδιαστεί με άξονα την απόδοση
Ακεραιότητα (Integrity)	Εξαρτάται από την DBMS και από τους κανόνες στον κώδικα της εφαρμογής	Η συναίνεση (Consensus) και η αμεταβλητότητα (immutability) εξασφαλίζουν την ακεραιότητα της πληροφορίας
Διαφάνεια (Transparency)	Ορίζεται από την εκάστοτε DBMS	Ο κάθε κόμβος αποτελεί ένα πλήρες αντίγραφο του blockchain
Έλεγχος (Control)	Κεντροποιημένος (Centralized)	Αποκεντρωμένος (Decentralized)

Συνεπώς προτού επιλεγεί το blockchain ως δομή δεδομένων της πληροφορίας είναι σημαντικό να γνωρίζουμε ότι δεν εφαρμόζει σε όλες τις περιπτώσεις αλλά προορίζεται κυρίως για να προσφέρει λύσεις σε τομείς που εμπεριέχουν [12] α) την ανάγκη για καταχώρηση συναλλαγών και τήρηση αμετάβλητου μητρώου και β) την έλλειψη εμπιστοσύνης μεταξύ των εμπλεκόμενων μερών.

Τομείς εφαρμογής blockchain

Μερικές από τις βιομηχανίες που αναμένεται να επηρεάσει [13] είναι:

Α. Τράπεζες/ Χρηματοοικονομικά

Η τεχνολογία blockchain αναμένεται να επηρεάσει σημαντικά τον συγκεκριμένο τομέα, καθώς δύναται να εξασφαλίσει πρόσβαση σε χρηματοπιστωτικές υπηρεσίες σε δισεκατομμύρια ανθρώπους σε όλο τον κόσμο, ακόμα και στον τρίτο κόσμο, προσφέροντας ελαχιστοποίηση του χρόνου και του κόστους διακανονισμού καθώς και των cross-border πληρωμών. Για πολλούς το ψηφιακό χρήμα φαίνεται να αποτελεί τον φυσικό διάδοχο του πλαστικού χρήματος.

Κυβέρνηση

Η εφαρμογή συστημάτων βασισμένων σε blockchain μπορεί να μειώσει σημαντικά τη γραφειοκρατία και να αυξήσει την ασφάλεια, την αποτελεσματικότητα και τη διαφάνεια των κυβερνητικών πράξεων.

Ψηφοφορία

Η τεχνολογία blockchain μπορεί να χρησιμοποιηθεί για την διεξαγωγή ηλεκτρονικής ψηφοφορίας, εξασφαλίζοντας διαφάνεια και ταχύτητα αλλά και μειώνοντας σημαντικά το κόστος διεκπεραίωσης.

Εμπόριο

Η τεχνολογία blockchain μπορεί να υποστηρίξει αποκεντρωμένες υπηρεσίες λιανικής που συνδέουν αγοραστές και πωλητές χωρίς μεσάζοντα και συναφή τέλη. Σε αυτές τις περιπτώσεις, η εμπιστοσύνη προέρχεται από τα smart contract systems, το tracking των συναλλαγών και την ασφάλεια των ανταλλαγών μέσα από συστήματα διαχείρισης διαδικτυακής φήμης.

Ομοίως, θα μπορούσε να ωφελήσει οργανισμούς σχετικούς με την διαχείριση ενέργειας, την κυβερνοασφάλεια, την διαχείριση αλυσίδας εφοδιασμού, τις παροχές κοινωνικής πρόνοιας, την υγειονομική περίθαλψη, το crowdfunding αλλά και το φιλανθρωπικό έργο.

Blockchain SWOT

Ποια είναι όμως τα δυνατά σημεία, οι αδυναμίες και οι ευκαιρίες που παρουσιάζονται από την χρήση της Ethereum blockchain τεχνολογίας?

Έξυπνα Συμβόλαια (Smart Contracts) και Αποκεντρωμένες εφαρμογές (Dapps)

Με την δημιουργία της Blockchain πλατφόρμας Ethereum το 2015 έγινε η εισαγωγή της έννοιας των «Έξυπνων συμβολαίων» (Smart Contracts) [14].

Τα «Έξυπνα Συμβόλαια» είναι στην ουσία προγράμματα κώδικα τα οποία ενεργοποιούνται και εκτελούνται όταν ικανοποιηθούν συγκεκριμένες συνθήκες αυτόματα.

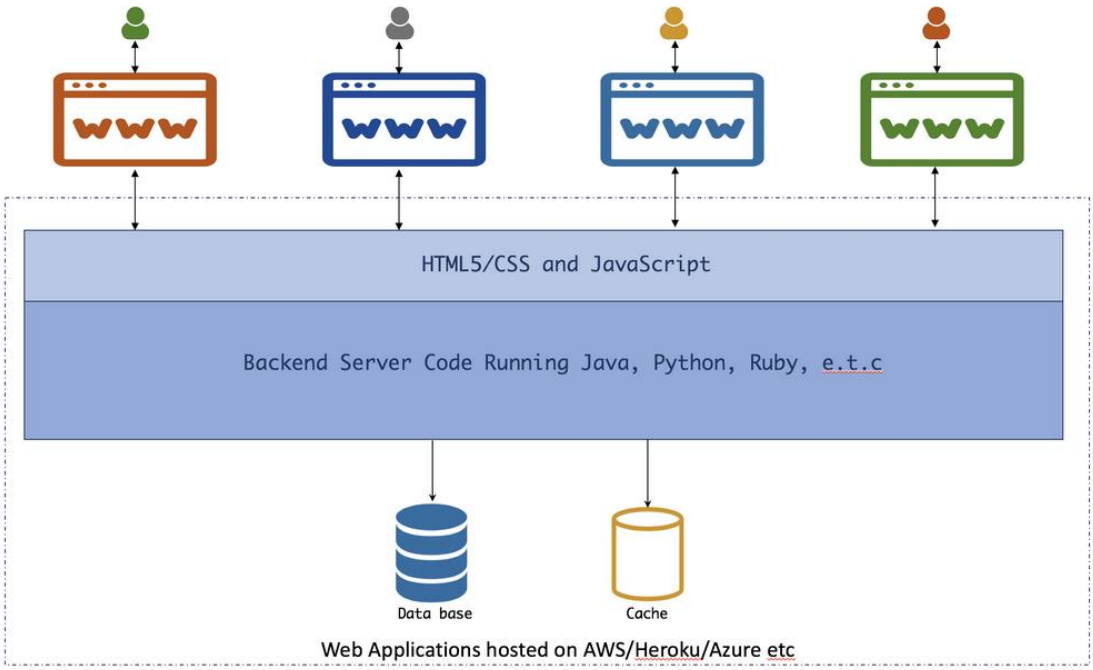
Το όνομα «Έξυπνα Συμβόλαια» το επινόησε ο Nick Szabo το 1996 [15], όμως με την εμφάνιση του Blockchain και με την ιδέα ότι προγράμματα μπορούν να εκτελούνται σε ένα ασφαλές περιβάλλον, τα «Έξυπνα Συμβόλαια» μετατράπηκαν σε εφαρμόσιμες λύσεις. Στα έξυπνα συμβόλαια οι κανόνες ψηφιοποιούνται και ορίζουν τι επρόκειτο να εκτελεστεί.

Θα μπορούσε να θεωρηθεί ως μια σύμβαση (όχι με την νομική της υπόσταση) που περιγράφει τους όρους της σχέσης μεταξύ διαφόρων μερών που συμμετέχουν στις συναλλαγές.

Η είσοδος των «Έξυπνων συμβολαίων» μεταμόρφωσε το προγραμματιστικό τοπίο και την δομή των υλοποιήσεων καθώς στον συμβατικό προγραμματισμό είχαμε την 3-tiered model δομή Client -> Application Server -> Database Server, όπου όταν ο Client στέλνει ένα αίτημα (request) διαμέσου του web browser (Chrome, Mozila κλπ) ο κώδικας που έχει υλοποιηθεί στον application server επικοινωνεί με τον Database server για να εξυπηρετήσει την CRUD λειτουργία.

Σε αυτήν την περίπτωση η web εφαρμογή φιλοξενείται κεντρικά σε κάποιον hosting provider και ο client (browser, widget, API κλπ) αλληλοεπιδρά με αυτήν την centralized εφαρμογή.

Προ – Ethereum εποχή:

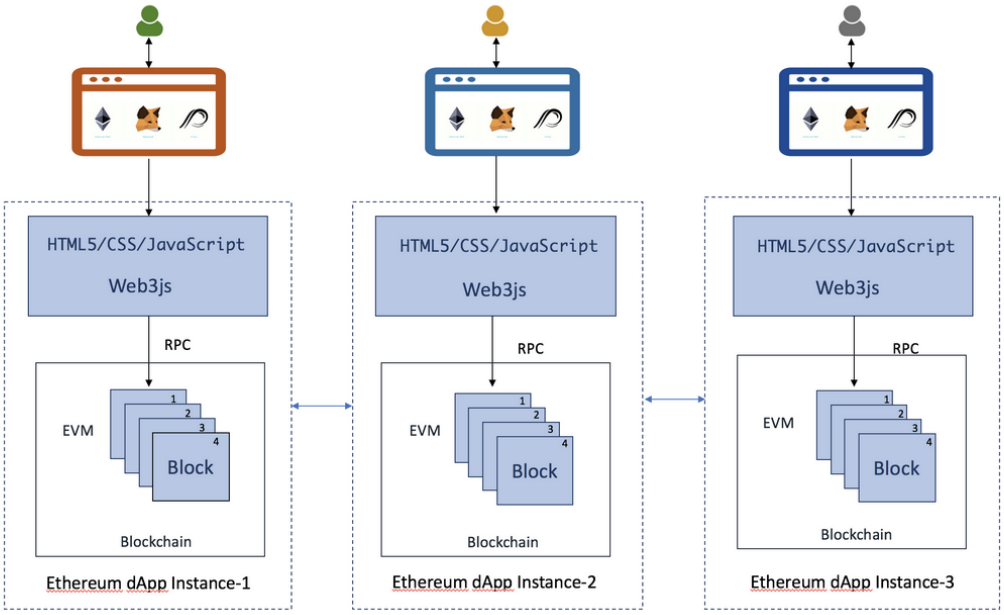


Εικόνα 7 https://medium.com/coinmonks/getting-started-with-ethereum-and-building-basic-dapp-ebb681fb3748

Στην Smart - Dapp εποχή, δεν υπάρχει κεντρικός server στον οποίο να συνδεθούν οι clients και ο κάθε client επικοινωνεί με το δικό του instance.

Αυτό σημαίνει ότι οποιοσδήποτε θελήσει να αλληλοεπιδράσει με μια εφαρμογή που έχει υλοποιηθεί σε περιβάλλον Blockchain (Decentralized Application) θα χρειαστεί να κατεβάσει τοπικά ένα πλήρες αντίγραφο του block.

Μετά – Ethereum εποχή:



Η εκτέλεση ενός συμβολαίου απαιτεί κόστος συναλλαγής σε ETH που εξαρτάται από την ποσότητα υπολογιστικής ισχύος που απαιτείται. Τι σημαίνει όμως αυτό;

Στην ενότητα που ακολουθεί, αναλύονται οι έννοιες Gas limit και Gas price που οδηγούν στο κόστος της συναλλαγής (Transaction cost).

Mining Ether και GAS

- **Gas limit**
- **Gas price**
- **Transaction Cost**

Gas (αέριο) είναι η ονομασία του κόστους εκτέλεσης μια συναλλαγής, την οποία οφείλουν να καταβάλουν όσοι προκαλούν μια νέα συναλλαγή στο Ethereum blockchain.

Η ονομασία gas έχει προκύψει από την προσέγγιση ότι το gas συμπεριφέρεται ως η κινητήριος δύναμη (cryptofuel) που οδηγεί την διάχυση των έξυπνων συμβολαίων (smart contracts) μέσα στην αλυσίδα του Ethereum [16].

Το Gas αγοράζεται με ether και τις υποδιαιρέσεις του, δηλαδή με το ψηφιακό νόμισμα του Ethereum, από τους εξορύχους (miners) οι οποίοι εκτελούν τον κώδικα.

Παρακάτω είναι μια λίστα με τις υποδιαιρέσεις του Ether και την αξία τους με βάση το Wei.

Unit	*Wei Value*	*Wei*
Wei	1 Wei	1
Kwei	1e3 Wei	1,000
Mwei	1e6 Wei	1,000,000
Gwei	1e9 Wei	1,000,000,000
Microether	1e12 Wei	1,000,000,000,000
Milliether	1e15 Wei	1,000,000,000,000,000
Ether	1e18 Wei	1,000,000,000,000,000,000

Την τιμή του gas (gas price), την ορίζουν οι ίδιοι οι miners, καθώς μπορούν να απορρίψουν μια συναλλαγή αν εκτιμήσουν ότι το ποσό ether που τους αποδίδει, είναι κάτω από το όριο (gas limit) που έχουν ορίσει για τις συναλλαγές τους. Ως gas price ορίζεται το κόστος σε Wei για κάθε μονάδα gas για την οποία οι λειτουργίες VM κοστολογούνται.

Σε αυτήν την κατεύθυνση, οι miners, επηρεάζουν τους νόμους της προσφοράς και ζήτησης για την συγκεκριμένη συναλλαγή (gas price) μέσα στο Ethereum blockchain, ενώ το gas limit λειτουργεί ως μια «ασπίδα» για τους ίδιους καθώς θέτοντας ένα όριο για την συναλλαγή, προστατεύονται από αστοχίες (bugs) του κώδικα (smart contact) που μπορούν να οδηγήσουν σε ατέρμονους βρόγχους και εξάντληση του υπολοίπου από τα διαθέσιμα ethers του λογαριασμού τους.

Ωστόσο, αξίζει να αναφερθεί, ότι στην περίπτωση όπου στον blockchain account δεν υπάρχει διαθέσιμη η απαιτούμενη ποσότητα σε ether τότε η διαδικασία ματαιώνεται και ενεργοποιείται ο μηχανισμός του roll-back (undo) όπου επιστρέφει η κατάσταση στην προγενέστερη εικόνα της. Άρα είναι σημαντικό να διατυπωθεί ότι οποιαδήποτε συναλλαγή μέσα στον κόσμο του Ethereum, δημιουργεί ένα κόστος το οποίο αποπληρώνεται με την χρήση ether, διαφορετικά δεν υφίσταται ως συναλλαγή.

Όμως τελικά οι έννοιες Gas και Ether πού διαφέρουν, μήπως είναι οι ίδιες?

Η απάντηση είναι όχι, καθώς το gas αντικατοπτρίζει το κόστος μιας συγκεκριμένης συναλλαγής και είναι κάτι σταθερό και δεδομένο από την αρχή, ενώ το Ether αποτελεί την «συναλλαγματική μετάφραση»,

την «ισοτιμία» δηλαδή του κόστους συναλλαγής, το οποίο μεταβάλλεται καθώς καθορίζεται από τους νόμους της αγοράς ψηφιακού χρήματος (digit money market).

Πώς υπολογίζουμε το κόστος στα Smart Contracts [17]

Gas

Κάθε ενέργεια σε επίπεδο EVM (Ethereum Virtual Machine) ονομάζεται OPCODE και περιλαμβάνει ενέργειες όπως ADD (πρόσθεση δυο integer), BALANCE (έλεγχος υπολοίπου λογαριασμού) και CREATE (δημιουργία νέου contract).

Κάθε OPCODE έχει ένα μέγεθος (κόστος) το οποίο ονομάζεται Gas. Gas είναι ένας αριθμός που απεικονίζει τον βαθμό πολυπλοκότητας των OPCODE ενεργειών. Για παράδειγμα η ενέργεια ADD χρησιμοποιεί 3 Gas ενώ η ενέργεια MUL (multiply) απαιτεί 5 Gas.

Η ποσότητα του Gas που απαιτεί κάθε OPCODE έχει υπολογιστεί ακολούθως [18]:

OPCODE_Gas_Costs.csv

Η βάση όλων των συναλλαγών είναι 21000 Gas, άρα αν δεν αλληλοεπιδράς με το contract η ελάχιστη ποσότητα σε Gas είναι 21000, ενώ αν αλληλοεπιδράς, η συναλλαγή κοστίζει 21000 Gas συν το Gas που προκύπτει.

Παραγόμενο: **gas_used**

Gas price

Όταν ο χρήστης στέλνει μια συναλλαγή.

Παραγόμενο: **gas_price**

Transaction Cost

Οι λειτουργίες στο Ethereum κοστίζουν **gas_price * gas_used**.

Παραγόμενο: **transaction_cost** = **gas_price * gas_used**

Για περισσότερες λεπτομέρειες αναφορικά με τον υπολογισμό του κόστους συναλλαγής μπορείτε να ανατρέξετε στο https://ethervm.io/

Συνοπτική περιγραφή συστήματος

Χρησιμοποιώντας λοιπόν την τεχνολογία Ethereum Blockchain, θα επιχειρηθεί (PoC) μέσα στα πλαίσια της εργασίας η υλοποίηση ενός έξυπνου συμβολαίου και η χρήση του ως back-end σε μια εφαρμογή ψηφοφορίας που αξιοποιεί τις δυνατότητες του Ethereum.

Σε αυτή την ενότητα θα καταγραφούν λεπτομέρειες που αφορούν τον σκοπό του συστήματος, τις λειτουργικές απαιτήσεις του συστήματος, τις παραδοχές που έγιναν, καθώς και τις κατηγορίες των χρηστών.

Σκοπός του συστήματος

Σκοπός του συστήματος είναι να υποστηρίξει την ψηφιακή ψηφοφορία με τη χρήση του Ethereum Blockchain [19] και να καλύπτει τις κάτωθι βασικές απαιτήσεις:

1. Απόκρυψη προσωπικών δεδομένων
2. Ένας ψήφος ανά φυσικό πρόσωπο
3. Διαφάνεια και αξιοπιστία

Βασικοί ρόλοι συστήματος

Οι βασικοί ενεργοποιοί (actors) που προδιαγράφονται είναι:

4. ο actor του επόπτη (Chairman)
5. ο actor του υποψήφιου (Candidate)
6. ο actor του ψηφοφόρου (Voter)

Πιο αναλυτικά:

1. Chairman:

Είναι ο owner του συστήματος και θα έχει τις δυνατότητες:

1. Να κάνει την δημιουργία του smart contract (create)
2. Να προσθέσει candidates (add candidates)
3. Να εκκινήσει την ψηφοφορία (start voting)
4. Να λήξει την ψηφοφορία (end voting)
5. Να μεταφέρει την ιδιοκτησία του contract (transfer ownership) σε άλλη διεύθυνση

2. Voter

Είναι ο βασικός χρήστης του συστήματος και θα έχει τις δυνατότητες:

1. Να κάνει registration (κατά το registration λαμβάνει στο mail του ένα auto-generated key (auth_token) που θα χρησιμοποιήσει στη διαδικασία της ψηφοφορίας)
2. Να ψηφίσει χρησιμοποιώντας για την καταχώρηση της ψήφους το auto-generated key (auth_token) που έλαβε από την φάση του registration

3. Υποψήφιος

Καταχωρείται από τον owner του contract με βάσει την διεύθυνση, το όνομα και την χώρα την οποία εκπροσωπεί.

Σχεδιασμός και υλοποίηση Συστήματος

Υλοποίηση λειτουργικών απαιτήσεων

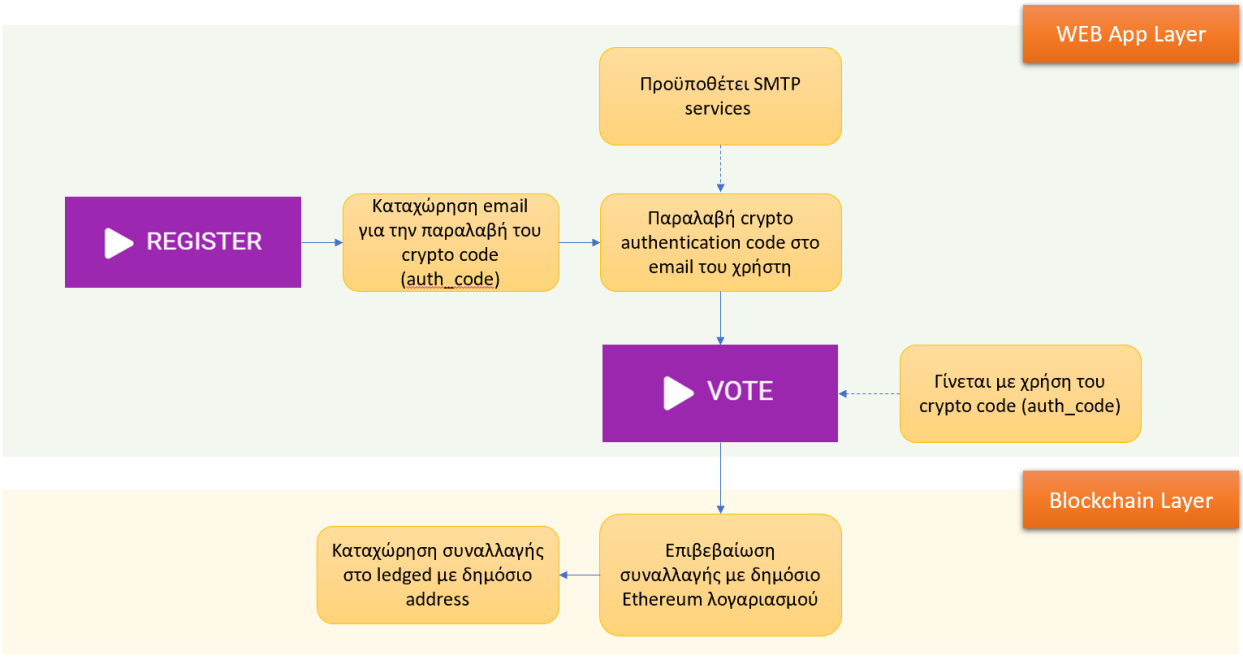
1. Απόκρυψη προσωπικών δεδομένων

Η χρήση του blockchain από τη φύση της, δεν διασφαλίζει την ανωνυμία του χρήστη αλλά την ψευδωνυμία του (GDPR concern) [20].

Για την συγκεκριμένη προσέγγιση προτείνεται η παραγωγή ενός auto-generated κλειδιού (crypto auth_token) το οποίο θα αποστέλλεται αυτόματα στο mail καταχώρησης του χρήστη, ώστε ο χρήστης να γίνει eligible, να έχει δηλαδή τη δυνατότητα συμμετοχής στη διαδικασία της ψηφοφορίας μέσα από τη web εφαρμογή.

Έτσι κατά το identification [21] του χρήστη στην διαδικασία του Voting, μέσα από τη web εφαρμογή, δεν θα χρησιμοποιείται η Euthereum διεύθυνση (address) που τον μοναδικοποιεί και ταυτοποιεί μέσα στο Ethereum αλλά θα γίνεται η χρήση του crypto generated κωδικού που θα παράγει με λογική μήτρας το ίδιο το contract - και θα λαμβάνει ο χρήστης μέσω SMTP services στην ηλεκτρονική του διεύθυνση (email).

Ως επέκταση και επιπλέον δικλείδα, θα μπορούσε επίσης να προταθεί, η επιβεβαίωση (Confirmation process μέσω του Metamask) να μην πραγματοποιείται από Ethereum λογαριασμό του χρήστη αλλά να πραγματοποιείται από μια κοινή / δημόσια Ethereum διεύθυνση – που θα γνωστοποιεί στους χρήστες ο εκάστοτε φορέας - ώστε να αποφεύγεται με οιονδήποτε τρόπο, η ταυτοποίηση του ψηφοφόρου με συγκεκριμένο Ethereum λογαριασμό, όπως απεικονίζεται διαγραμματικά ακολούθως:



Με αυτόν τον τρόπο, εισάγοντας ένα «ενδιάμεσο» layer στην αυθεντικοποίηση του χρήστη και στο επίπεδο του blockchain, εμποδίζεται η ιχνηλασιμότητα του προσώπου που υπάρχει πίσω από την Euthereum διεύθυνση και εξασφαλίζεται ότι για την διεξαγωγή της ψηφοφορίας, δεν θα γίνει logged το πραγματικό Ethereum identity του χρήστη.

Αυτή όμως η προσέγγιση θα είχε πραγματικά νόημα, μόνο σε υλοποιήσεις που θέλουμε να διασφαλίζεται πλήρως η ανωνυμία, όπως συμβαίνει στην περίπτωση μιας κρατικής ψηφοφορίας και όχι σε εφαρμογές που σχετίζονται με τη διασφάλιση ποιότητας ή την διαχείριση αλυσίδας εφοδιασμού όπως πχ το smart farming, όπου εκεί η ιχνηλασιμότητα πρέπει να διαδραματίζει πρωταγωνιστικό ρόλο.

2. Ένας ψήφος ανά φυσικό πρόσωπο

Για την συγκεκριμένη απαίτηση, η λίστα των voters θα τηρεί την πληροφορία των ψηφοφόρων που ψήφησαν, μεταβάλλοντας την κατάσταση (status) σε voted ώστε να εξασφαλίζεται ότι δεν θα υπάρχει δυνατότητα να ψηφίσει το ίδιο πρόσωπο – ο ίδιος χρήστης παραπάνω από μια φορά.

Έτσι ακόμα και στην περίπτωση που επιχειρήσει ο χρήστης να ψηφίσει για $2^η$ φορά, ο ψήφος του δεν θα συνυπολογιστεί στο counting των αποτελεσμάτων, καθώς θα έχει ήδη λάβει την ένδειξη ότι έχει γίνει voted.

3. Διαφάνεια και αξιοπιστία

Στο blockchain οποιαδήποτε κίνηση – συναλλαγή καταγράφεται και καταχωρείται στο γενικό καθολικό με τέτοιο τρόπο που την καθιστά immutable (αμετάβλητη). Επίσης οι κανόνες του smart contract, από τη στιγμή που έχει μεταπέσει στο γενικό καθολικό (deployed) δεν μπορούν να μεταβληθούν και έτσι παραμένουν αδιάβλητοι.

Άρα, όπως αναφέρθηκε και στη σελ. 8 του παρόντος, η ίδια η χρήση της τεχνολογίας εξασφαλίζει την διαφάνεια και την αξιοπιστία, καθώς εξασφαλίζει την φυσική συνέχεια των καταχωρήσεων καθώς και δεν υποστηρίζει δυνατότητες διαγραφής (delete) ή ενημέρωσης (update) στα δεδομένα μητρώου που έχουν καταχωρηθεί.

Βασικές λειτουργίες συστήματος

Περιγράφονται συνοπτικά οι βασικές λειτουργίες του συστήματος:

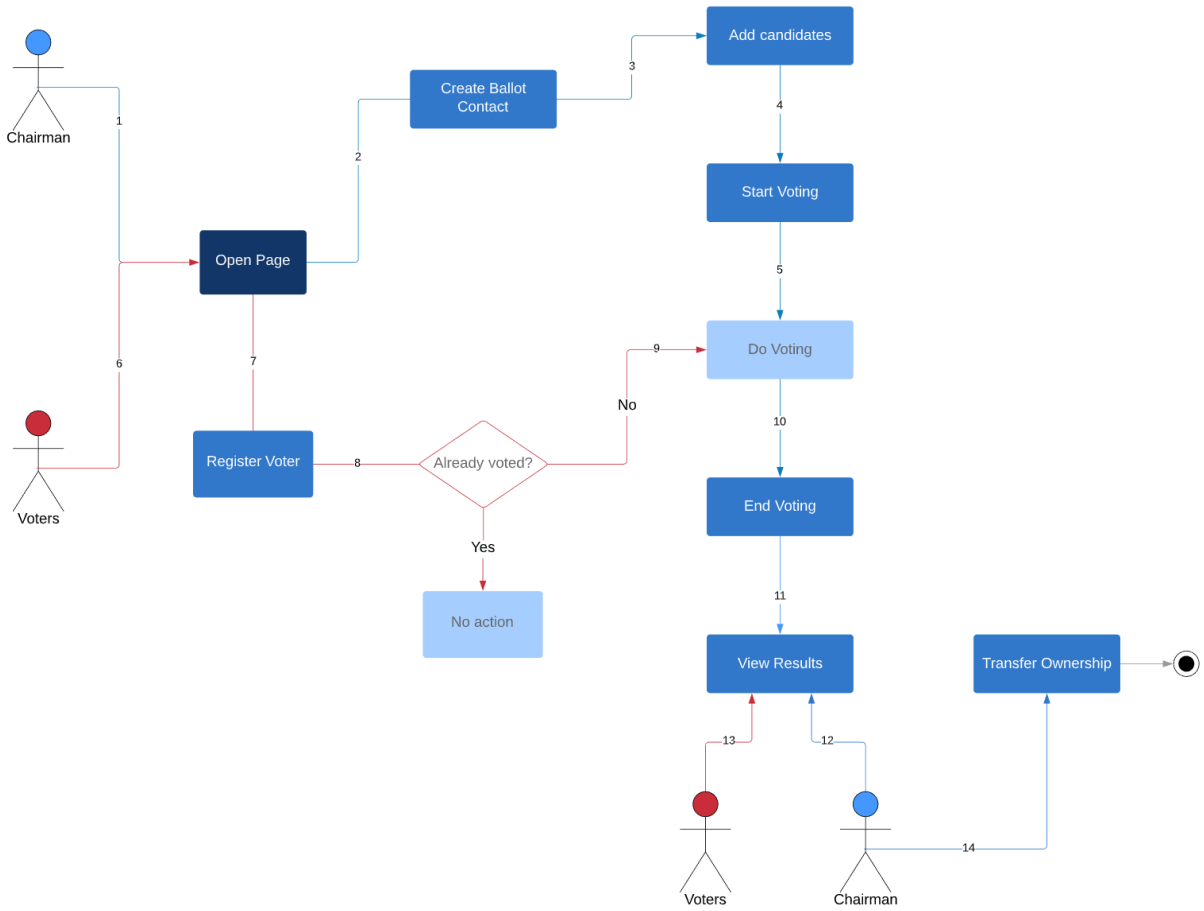
1. **Create**: οδηγεί στην δημιουργία του smart contract με τους κανόνες της ψηφοφορίας
2. **Start**: εκκινεί χρονικά τη διαδικασία της ψηφοφορίας
3. **Stop**: σταματάει τη διαδικασία της ψηφοφορίας
4. **State**: παρουσιάζει την τρέχουσα κατάσταση της ψηφοφορίας (Created, Voting, Ended) και χρησιμοποιείται και ως trigger για την ενεργοποίηση της λειτουργίας των διαδικασιών του συστήματος
5. **Vote**: δίνει τη δυνατότητα καταχώρησης ψήφου υπέρ συγκεκριμένου candidate και άρσης του δικαιώματος προσμέτρηση δεύτερου ψήφου – στην περίπτωση που το ίδιο «auth_token»
6. **Transfer address**: δίνει τη δυνατότητα μεταφοράς της ιδιοκτησίας σε άλλη διεύθυνση
7. **Results**: παρουσιάζει συγκεντρωτικά ανά υποψήφιο/χώρα το σύνολο των ψήφων που έχουν ληφθεί
8. **Winner**: αποκαλύπτει τον candidate ή τους candidates (σε περίπτωση ισοψηφίας) που έχουν συγκεντρώσει τους περισσότερους ψήφους

Περιγράφονται συνοπτικά οι βασικές οντότητες του συστήματος:

9. **About**: Εμφανίζει τα στοιχείων του owner του smart contact (name, address)
10. **Proposal**: Αποτελεί το motto της ψηφοφορίας
11. **Candidates**: Εμφανίζει σε λίστα τους υποψήφιους ανά όνομα και χώρα καθώς και τους ψήφους που λαμβάνουν κατά την διαδικασία της ψηφοφορίας
12. **Voters**: Εκπροσωπεί τον κόσμο που θα συμμετάσχει στην διαδικασία της ψηφοφορίας
13. **Results**: Εμφανίζει τα αποτελέσματα (sync) βάσει της ψηφοφορίας

Σχεδιασμός υλοποίησης

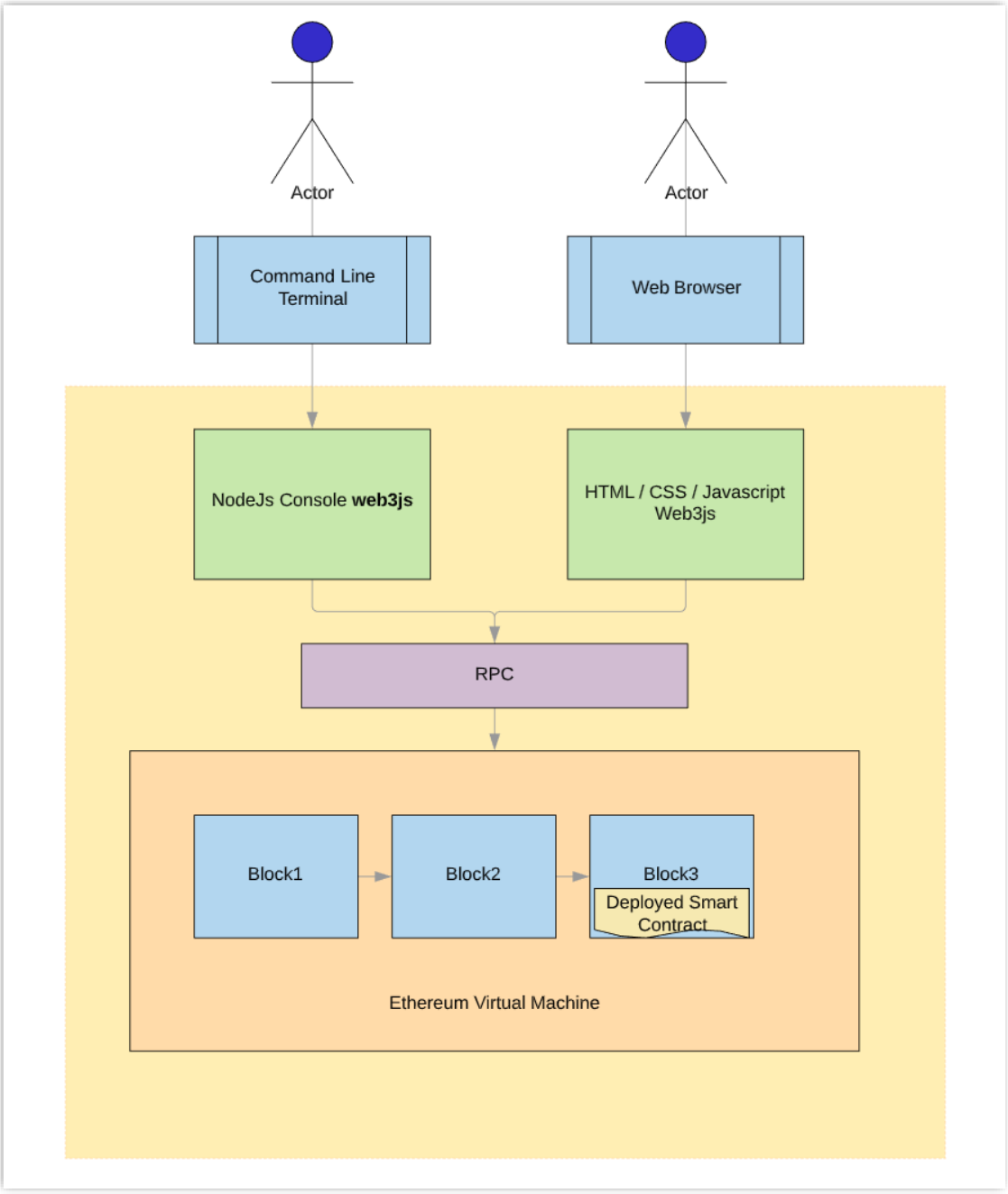
Βασικό workflow



Σημειώσεις:

- Ο Chairman ανοίγει την εφαρμογή χρησιμοποιώντας την Ethereum address και το password που διαθέτει και εκτελεί τη δημιουργία του Contract.
- Με την δημιουργία του Contract στο Ethereum Blockchain, και την απόδοση συγκεκριμένου address στο συμβόλαιο, εισάγει τους υποψήφιους και ενεργοποιεί την εκκίνηση της ψηφοφορίας.
- Με την εκκίνηση της ψηφοφορίας όλοι οι voters ανοίγουν τη σελίδα και εκτελούν την ενέργεια του registration προκειμένου να τους αποσταλεί το crypto key με το οποίο θα πραγματοποιήσουν την διαδικασία της ψηφοφορίας.
- Για την ολοκλήρωση της ψηφοφορίας ο Chairman, εκτελεί την ενέργεια Stop voting και «παγώνει» τη δυνατότητα καταχώρησης νέου ψήφου.
- Με την ολοκλήρωση της ψηφοφορίας, αποκαλύπτονται τα τελικά αποτελέσματα και δίνεται η δυνατότητα μεταφοράς του ownership σε νέα διεύθυνση.

Αρχιτεκτονική υλοποίησης



Εργαλεία και τεχνολογίες

Στο κεφάλαιο αυτό παρουσιάζονται τα κυριότερα εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής της παρούσας διπλωματικής εργασίας.

1. **Truffle**

Έκδοση που χρησιμοποιήθηκε: 5.0.2

Το Truffle αποτελεί μέρος της Truffle Suite (Truffle, Ganache, Drizzle) και χρησιμοποιείται για την ανάπτυξη έξυπνων συμβολαίων σε Ethereum VM περιβάλλον αλλά και ως testing framework. Έχει ενσωματωμένες λειτουργικότητες για μεταγλώττιση και μετάπτωση (deployment) του συμβολαίου σε δημόσια και ιδιωτικά δίκτυα. Χρησιμοποιήθηκε κυρίως κατά την πρώτη φάση της ανάπτυξης, για την δημιουργία του smart contract.

2. **Ganache CLI**

Έκδοση που χρησιμοποιήθηκε: 2.1.1

Το Ganache CLI (γνωστό και ως **TestRPC**) είναι ένα βασισμένο στο Node.js Ethereum client για δοκιμή και ανάπτυξη. Χρησιμοποιεί την συλλογή βιβλιοθηκών ethereumjs για να προσομοιώσει πλήρως έναν πραγματικό Ethereum κόμβο, χωρίς όμως την επιβάρυνση που συνεπάγεται η εκτέλεση του πραγματικού Ethereum κόμβου.

Δημιουργεί δηλαδή ένα ιδιωτικό Ethereum blockchain με 10 dummy Ethereum λογαριασμούς/διευθύνσεις, που ακούει στην τοπική πόρτα 8545 για γρήγορη δοκιμή και ανάπτυξη νέων έξυπνων συμβολαίων και κατανεμημένων εφαρμογών DApp (Decentralized application).

3. **NodeJs**

Έκδοση που χρησιμοποιήθηκε: **v12.13.0** με τα κάτωθι dependencies:

bcrypt: ^3.0.7,
body-parser: ^1.19.0,
ethers: ^4.0.40,
express: ^4.17.1,
ganache-cli: ^6.7.0,
solc: ^0.5.3

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού (κυρίως διακομιστών) χτισμένη σε περιβάλλον Javascript. Στόχος του Node είναι να παρέχει ένα εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών.

4. Solidity

Έκδοση που χρησιμοποιήθηκε: 0.5.13

Η solidity είναι η γλώσσα υψηλού επιπέδου που χρησιμοποιείται για την συγγραφή έξυπνων συμβολαίων (smart contracts) για το Ethereum blockchain και λειτουργεί με την λογική IFTTT (IF-This-Then-That).

Έχει παρόμοιο συντακτικό με αυτό της JavaScript, οπότε είναι εύκολα κατανοήσιμη από έναν πολύ μεγάλο αριθμό προγραμματιστών. Είναι μία στατικού τύπου γλώσσα και υποστηρίζει την κληρονομικότητα με παρόμοιο τρόπο με άλλες γλώσσες προγραμματισμού (πχ. C++). Για την σύνταξή της έχει επηρεαστεί από τις C++, Python και JavaScript και έχει σχεδιαστεί με σκοπό το Ethereum Virtual Machine (EVM)

5. Visual Studio Code

Έκδοση που χρησιμοποιήθηκε: 1.38.1 με ενεργοποίηση του extension Ethereum Solidity Language for Visual Studio Code για διευκόλυνση της σύνταξης του με το smart contract και την δημιουργία του .sol αρχείου.

6. MetaMask

Το MetaMask είναι ένα plugin διαθέσιμο για τους φυλλομετρητές Google Chrome, Mozilla Firefox, Opera και Brave. Είναι στην ουσία μία γέφυρα η οποία δίνει στον browser πρόσβαση στις κατανεμημένες εφαρμογές (DApps), που για την λειτουργία τους βασίζονται στο δίκτυο Ethereum.

Το μεγάλο πλεονέκτημα που προσφέρει είναι το γεγονός ότι με την χρήση του, η πρόσβαση στις εφαρμογές αυτές, δεν απαιτεί την εκτέλεση του πλήρους Ethereum κόμβου στο μηχάνημα του χρήστη. Συμπεριλαμβάνει μία ασφαλή κρύπτη (secure identity vault) και παρέχει στον χρήστη μία διεπαφή, μέσω της οποίας αυτός μπορεί να διαχειρίζεται τους Ethereum λογαριασμούς/διευθύνσεις του, ώστε να αλληλοεπιδρά με τις ιστοσελίδες, να στέλνει ή και να υπογράφει συναλλαγές και να υπογράφει δεδομένα.

7. Web3.js

Η βιβλιοθήκη web3.js είναι το επίσημο Ethereum Javascript API και χρησιμοποιείται για την αλληλεπίδραση με τα έξυπνα συμβόλαια. Είναι μια συλλογή από modules με συγκεκριμένη λειτουργικότητα στο οικοσύστημα του Ethereum

8. Template For Website

Material Design Template https://www.creative-tim.com/product/material-kit

Βασικά Σημεία Υλοποίησης Smart Contract

Για την δημιουργία του έξυπνου συμβολαίου [22] χρησιμοποιήθηκε ως γλώσσα προγραμματισμού η Solidity και για αυτό δημιουργούμε ένα αρχείο με κατάληξη .sol. Για το όνομα του αρχείου, ορίζουμε, το ίδιο όνομα με το όνομα του contract που θα δημιουργήσουμε.

Έτσι στην περίπτωσή μας το αρχείο θα ονομαστεί Ballot.sol όπως βλέπουμε να έχει ονομαστεί και το `contract Ballot`

Επίσης, ορίζουμε την έκδοση με την οποία θα γίνει η μεταγλώττιση (compilation) του συμβολαίου (εδώ έχει χρησιμοποιηθεί η 0.5.13 που στην παρούσα φάση είναι μια από τις τελευταίες διαθέσιμε εκδόσεις).

Στο σημείο αυτό αξίζει να αναφερθεί ότι η γλώσσα προγραμματισμού Solidity είναι μια διαρκώς μεταβαλλόμενη και εξελισσόμενη γλώσσα με αποτέλεσμα μεταξύ των διαφόρων εκδόσεων, να παρουσιάζονται σημαντικές διαφοροποιήσεις και καλό είναι αν κάποιος θέλει να αποφύγει εκπλήξεις με την εκτέλεση του προγράμματός του να ορίζει συγκεκριμένη έκδοση πχ `pragma solidity 0.5.13;` αποφεύγοντας την επιλογή ^ που υποδηλώνει από 0.5.13 και οποιαδήποτε νεότερη

```
pragma solidity ^0.5.13;

contract Ballot{

  address public ballotOfficialAddress;
  string public ballotOfficialName;
  string public ballotOfficialProposal;
  bytes private secretNounce;
  string public getLatestAuthCode;
  string public errorMessage;
  //initialization
  uint flag=0;
  //Used to define the current status
  enum State { Created, Voting, Ended }

  State public state;
```

Για την παρακολούθηση της κατάστασης του contract έχει δημιουργηθεί η μεταβλητή state η οποία λαμβάνει τις τιμές Created, Voting, Ended και γίνεται αρχικοποίηση με :

```
uint flag=0;
```

Για όσο διάστημα είναι ενεργή η ψηφοφορία ενώ γυρίζει σε

```
flag=1;
```

Όταν ολοκληρωθεί η διαδικασία.

Να σημειώσουμε ότι στο blockchain, λόγω του ότι δεν υπάρχει η έννοια του «end» είναι σημαντικό να ορίσουμε μέσα στον κώδικα του contract πότε παύει να ισχύει η λειτουργία του.

Με την δεσμευμένη λέξη `struct` δημιουργούνται οι τύποι `Candidate` και `Voter`

```
  //Structs allow you to define new types. In our case we define Candidate and Voter
  struct Candidate {
      string name;
      string country;
      uint256 candidate_id;
      uint voteCount;
  }

  struct Voter{
      string auth_code;
      bool authorized;
      uint candidate_id;
      string email;
      bool voted;
      address voter_address;
  }
```

Για την δημιουργία (κατά το deployment) του smart contract καλούμαστε να ορίσουμε τα πεδία που ορίζονται στον Constructor.

Στην συγκεκριμένη υλοποίηση έχουν οριστεί η διεύθυνση (address) αυτού που θα δημιουργήσει το contract καθώς και δυο επιπλέον string πεδία με την ονομασία της ψηφοφορίας (ballotOfficialName) και την πρόταση – motto (proposal) για την έναρξη της ψηφοφορίας.

Στη Solidity, τα συγκεκριμένα πεδία τα ενημερώνουμε μόνο μια φορά - κατά το deployment του contract στο blockchain - ενώ σε περίπτωση απουσίας constructor από τον κώδικα, τότε ο compiler δημιουργεί τον default.

```
constructor(string memory newAddress, string memory _ballotOfficialName, string
memory _proposal) public {
    ballotOfficialAddress = parseAddr(newAddress);
    ballotOfficialName = _ballotOfficialName;
    ballotOfficialProposal=_proposal;
    state = State.Created;
}
```

Στο contract έχουν δημιουργηθεί οι function :

```
1. addCandidates(string memory _name, string memory _country) public
   inState(State.Created)
2. getCandidateList() public view returns(string memory)
3. startVote()public inState(State.Created)
4. addVoter(string memory _email) public returns (string memory)
5. random(string memory _email)  private view returns (string memory)
6. getVoter (string memory _email) public view returns (string memory)
7. doVote(string memory _auth_code, uint _can_id) public inState(State.Voting)
   returns (string memory text)
8. endVote() public inState(State.Voting)
9. result() public view returns (string memory)
10. ResultCountry() public view returns (string memory)
11. getWinnerName() public view returns (string memory)
12. getWinnerCountry() public view returns (string memory)
13. getWinnerCount() public view returns(uint)
14. transferOwnership(string memory newAddressString) inState(State.Ended)
```

και έχουν χρησιμοποιηθεί οι δημοσιευμένες functions

```
1. parseAddr(string memory a) internal pure returns (address parsedAddress)
2. uint2str(uint _i) internal pure returns (string memory _uintAsString)
```

Πιο αναλυτικά :

addCandidate – για την προσθήκη υποψηφίων στην λίστα με τους candidates

```
function addCandidates(string memory _name, string memory _country) public
inState(State.Created)  {
    totalCandidates= totalCandidates + 1;
    candidates[totalCandidates] = Candidate(
        {name: _name,
        country: _country,
        candidate_id:
        totalCandidates,
        voteCount:0});
}
```

addVoter για την δημιουργία (registration) του ψηφοφόρου. Εδώ έχει γίνει η παραδοχή ότι έχουν χρησιμοποιηθεί SMTP services ώστε ο κάθε χρήστης να λαμβάνει για τον δικό του λογαριασμό, στο mail του, το autogenarated key (auth_code) που θα χρησιμοποιήσει ως κλειδί για την καταχώρηση του ψήφου του κατά την doVote, δηλαδή για την διαδικασία της ψηφοφορίας, ώστε να εξασφαλίζεται η ανωνυμία του.

```
function addVoter(string memory _email) public returns (string memory) {
  if (emailExists[_email] == false) {
        string memory key = random(_email);
        emailExists[_email] = true;
        emailKeys[_email] = key;
        voters[key] = Voter({auth_code:key, authorized:true, candidate_id:0,
email:_email, voted:false, voter_address:msg.sender});
        return key;
    }
}
```

Για την παραγωγή του authentication code έχει χρησιμοποιηθεί η function uint2str [23] που ουσιαστικά μετατρέπει την είσοδο του string σε κρυπτογραφημένο int

```
function uint2str(uint _i) internal pure returns (string memory _uintAsString) {
    if (_i == 0) {
        return "0";
    }
    uint j = _i;
    uint len;
    while (j != 0) {
        len++;
        j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;
    while (_i != 0) {
        bstr[k--] = byte(uint8(48 + _i % 10));
        _i /= 10;
    }
```

```
    return string(bstr);
}
```

doVote για την διαδικασία της ψηφοφορίας από τον χρήστη - που έχει ολοκληρώσει το registration - και έχει λάβει το `_auth_code`. Εδώ εφαρμόζονται διάφοροι έλεγχοι τόσο για την κατάσταση της ψηφοφορίας (αν είναι ενεργή ή όχι) όσο και για το αν έχει ήδη χρησιμοποιηθεί το _auth_code ώστε να μην επιτρέπεται η προσμέτρηση πάνω από έναν ψήφο για τον ίδιο χρήστη.

```
function doVote(string memory _auth_code, uint _can_id) public inState(State.Voting)
returns (string memory text) {
    if (flag==0) {
        if (voters[_auth_code].voted == true) {
          errorMessage="Already Voted";
        } else {
            if (voters[_auth_code].authorized == false) {
                errorMessage="Voter Not Authorized";
            } else {
                if (voters[_auth_code].voter_address!=msg.sender) {
                    errorMessage="Voting with Wrong Address";
                } else {
                    voters[_auth_code].voted = true;
                    voters[_auth_code].candidate_id = _can_id;
                    candidates[_can_id].voteCount = candidates[_can_id].voteCount +
1;
                    if (candidates[_can_id].voteCount > winnerCandidateVotes) {
                        winnerCandidateVotes = candidates[_can_id].voteCount;
                    }
                    totalVotes = totalVotes + 1;
                    errorMessage="Vote Successful";
                }
            }
        }
    } else {
        errorMessage="Election over";
    }
    return string(errorMessage);
}
```

result για την συγκέντρωση των αποτελεσμάτων ανά υποψήφιο / χώρα. Με την χρήση της abi.encodePacked() γίνεται concatenation των inputs data σε bytes array

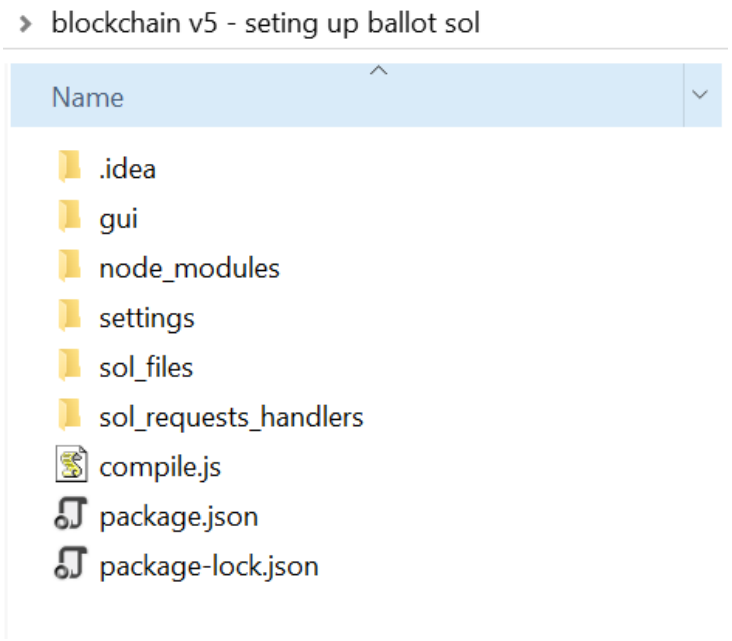
```
function result() public view returns (string memory){
    string memory names;

    for(uint i=1; i<=totalCandidates; i++){
      string memory id =
string(abi.encodePacked('"id":',uint2str(candidates[i].candidate_id)));
      string memory name = string(abi.encodePacked('"name":"',candidates[i].name,
'"'));
      string memory country =
string(abi.encodePacked('"country":"',candidates[i].country, '"'));
      string memory voteCount =
string(abi.encodePacked('"voteCount":',uint2str(candidates[i].voteCount)));
      if (i < totalCandidates) {
        names =
string(abi.encodePacked(names,'{',id,',',name,',',country,',',voteCount,'},'));
```

```
      } else {
        names =
string(abi.encodePacked(names,'{',id,',',name,',',country,',',voteCount,'}'));
      }
    }
    names = string(abi.encodePacked('[',names,']'));
    return names;
}
```

transferOwnerShip για την μεταφορά την «ιδιοκτησίας» του contract σε μια άλλη διεύθυνση, όταν ολοκληρωθεί η διαδικασία της ψηφοφορίας και μεταβληθεί η κατάσταση σε Ended `inState(State.Ended)`.

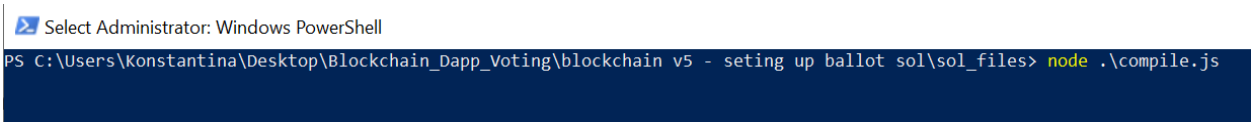
```
function transferOwnership(string memory newAddressString) inState(State.Ended)
public {
    address newAddress = parseAddr(newAddressString);
    require(newAddress != address(0));
    address oldAddress = ballotOfficialAddress;
    ballotOfficialAddress = newAddress;
    emit OwnershipTransferred(oldAddress, newAddress);
}
```

Παραδοτέα υλοποίησης συστήματος



Σημειώσεις:

1. Προϋποθέσεις:
   a. Εγκατάσταση του Nodejs με τα dependencies που έχουν αναφερθεί
   b. Δημιουργία του solidity αρχείου ή/και τοποθέτησή του μέσα στον φάκελο \blockchain v5 - seting up ballot sol\sol_files
2. Από τον φάκελο \blockchain v5 - seting up ballot sol\ και από File -> ανοίγουμε με ρόλο administrator το Windows PowerShell και εκτελούμε την εντολή node .\compile.js

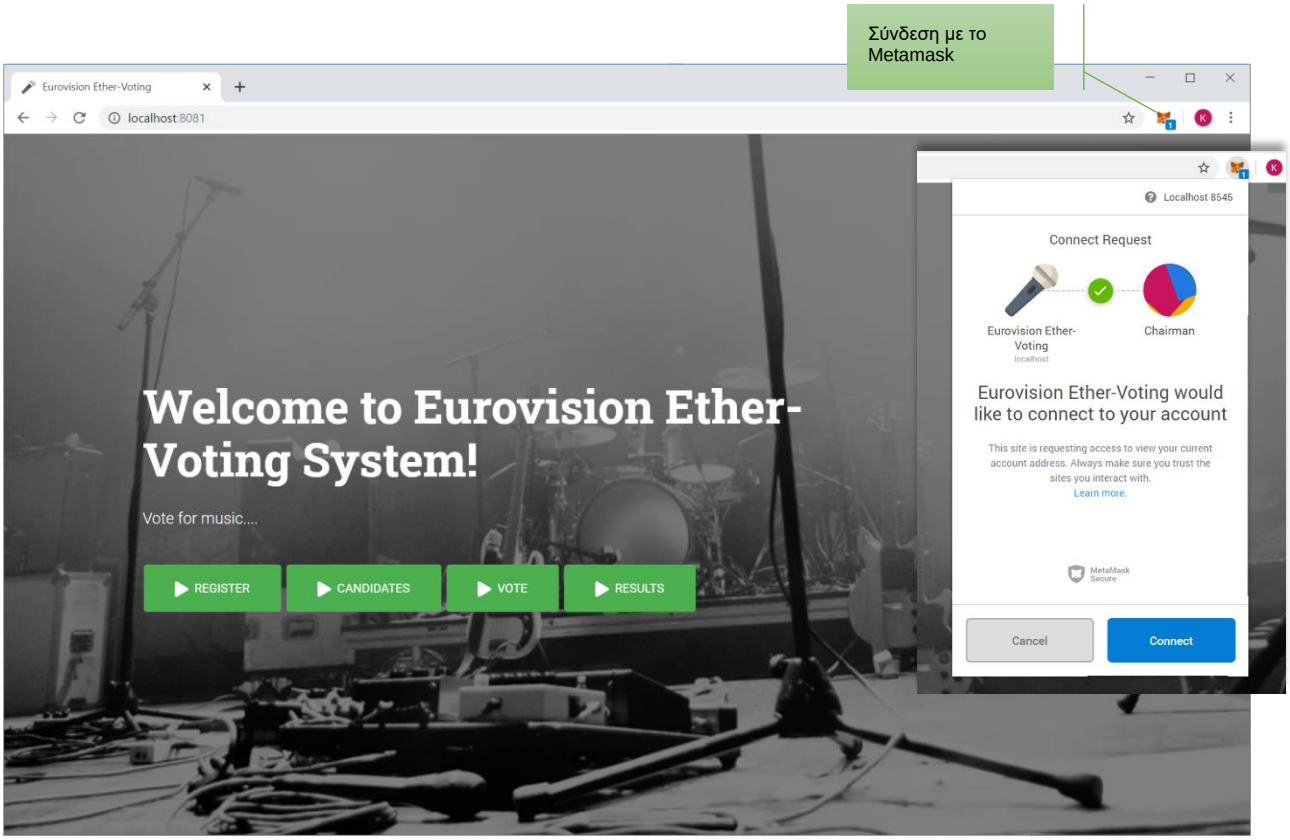


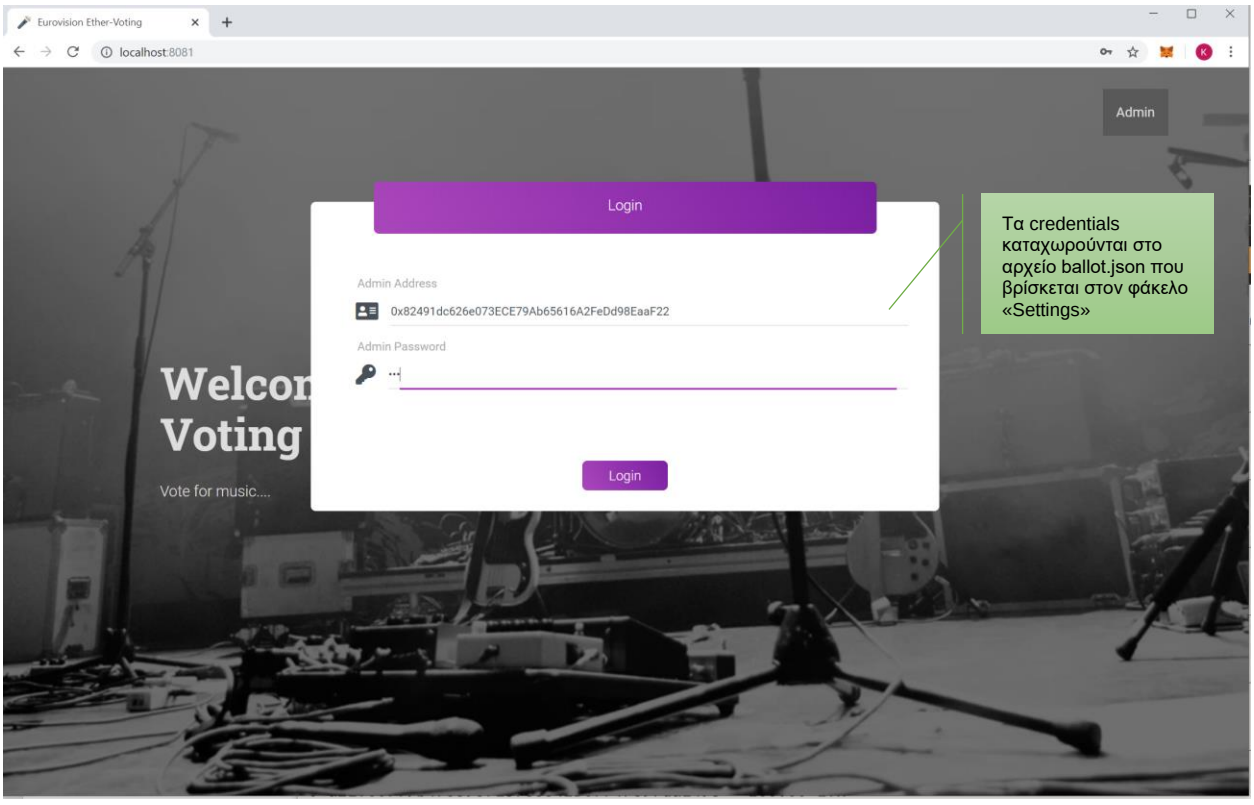
3. Στο ερώτημα «Which Sol file do you want to load?» ορίζουμε το Ballot (αξίζει να αναφερθεί ότι κατά την διαδικασία της υλοποίησης χρειάστηκαν παραπάνω από ένα version του .sol αρχείου οπότε στήθηκε μηχανισμός για την διαχείριση παραπάνω της μιας έκδοσης)
4. Όταν ολοκληρωθεί η διαδικασία του complilation επιστρέφονται στην οθόνη το ABI (metadata) και το Bytecode από την μεταγλώτισση του smart contract.
5. Για την παρουσίαση της υλοποίησης δημιουργήθηκαν κάποιες ιστοσελίδες, αυτές οι σελίδες έχουν δηλωθεί στο .js αρχείο που βρίσκεται στον φάκελο «sol_requests_handlers».

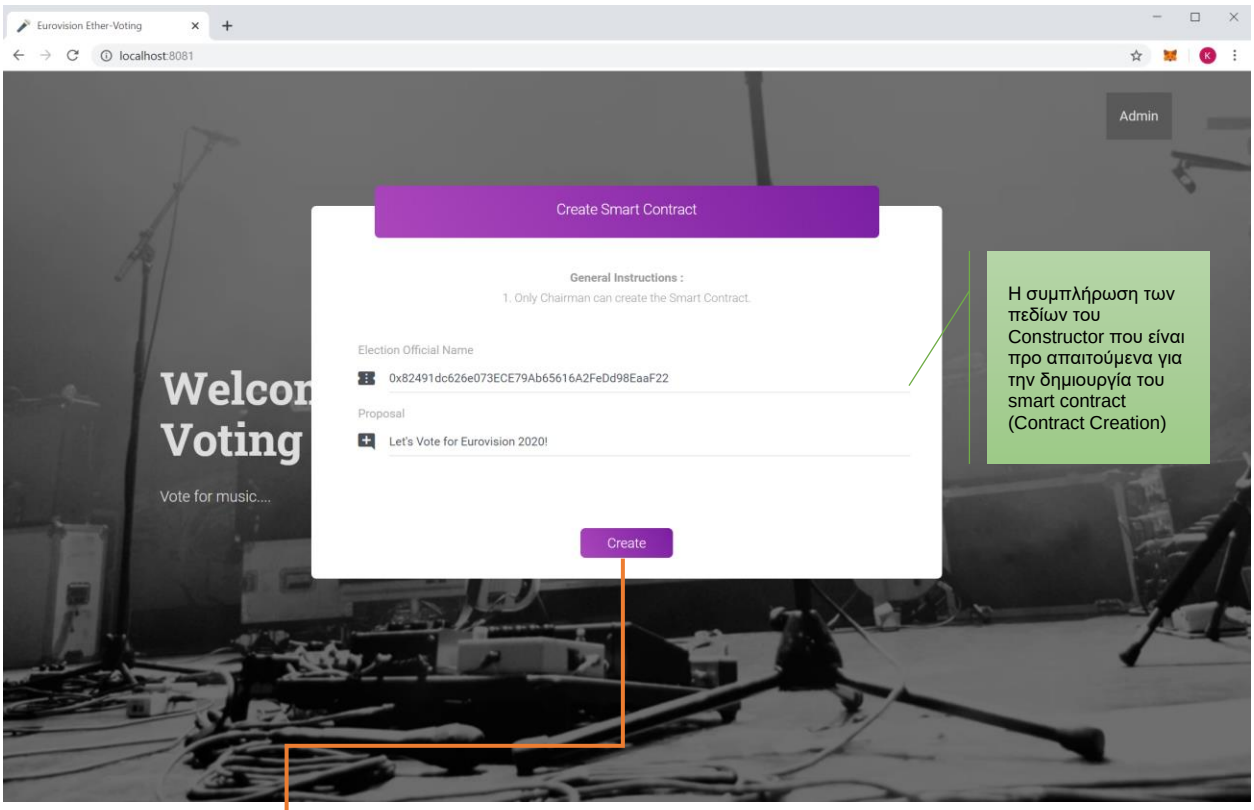
   Να σημειωθεί ότι το .js αρχείο θα πρέπει να έχει την ίδια ονομασία με το .sol αρχείο (Ballot.js).
6. Στον φάκελο «Settings» υπάρχει το αρχείο ballot.json με τις ρυθμίσεις του application (έχουν γίνει comment τα σημεία του encryption για να μπορεί να επαναχρησιμοποιηθεί) ενώ ως ρυθμίσεις έχουν οριστεί η blockchain διεύθυνση του admin και το password.
7. Ο φάκελος «idea» έχει δημιουργηθεί από το πρόγραμμα JetBrains WebStorm (αφορά στα αρχεία της συγκεκριμένης εφαρμογής).
8. Ο φάκελος «node_modules» εμπεριέχει όλα τα προγράμματα τα οποία εγκαταστάθηκαν στο nodejs

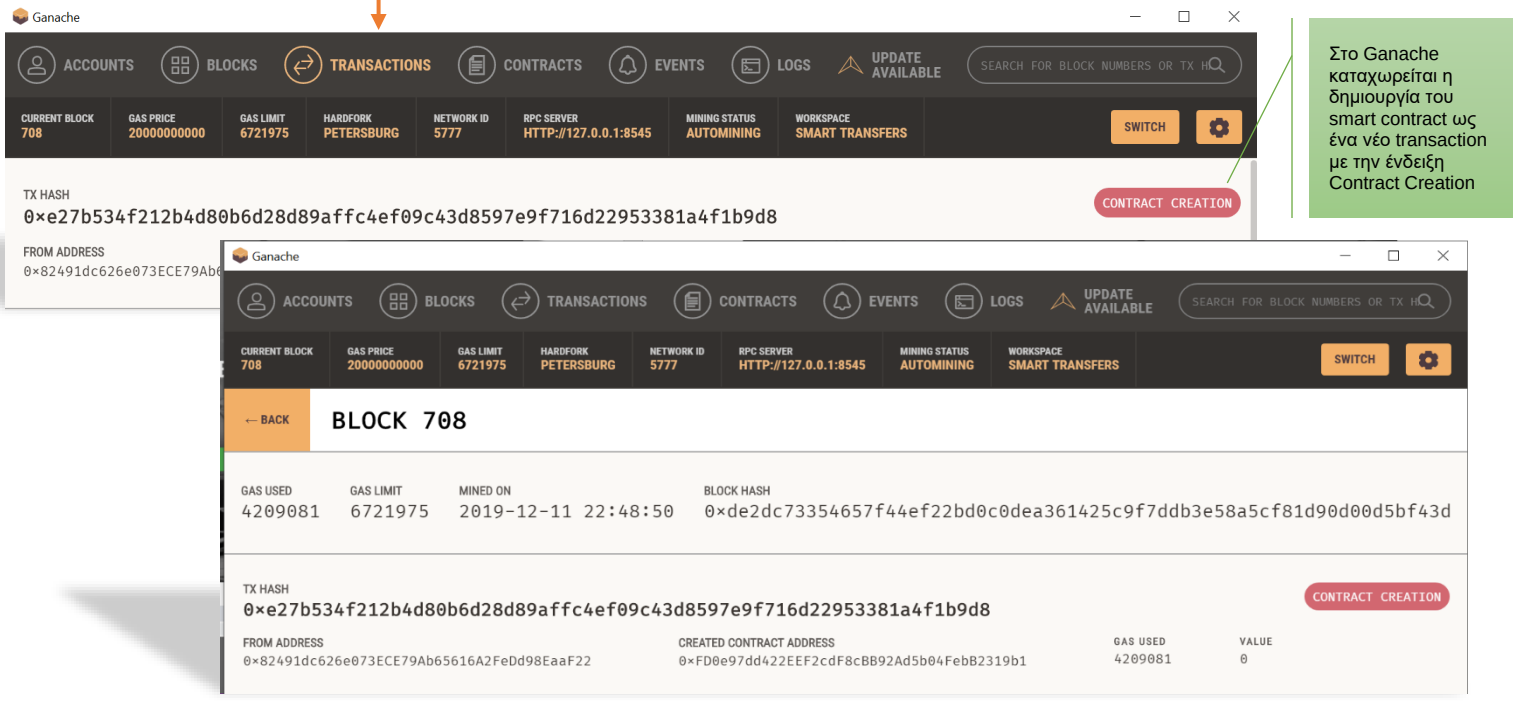
9. Τα αρχεία package.json και package-lock.json είναι τα εργαλεία τα οποία περιέχουν τα προγράμματα τα οποία χρησιμοποιήθηκαν στο nodejs, δηλαδή τον solidity compiler, τον express web server κλπ
10. Για την εκτέλεση του προγράμματος:
    a. Ανοίγουμε το Ganache
    b. Κάνουμε log-in στο Metamask επιλέγοντας το Localhost 8545
    c. Ανοίγουμε τη σελίδα http://localhost:8081/ με τον ρόλο του administrator (τα στοιχεία του admin για τις ανάγκες αναπαραγωγής της υλοποίησης, τα ορίζουμε στο αρχείο ballot.json που βρίσκεται στον φάκελο φάκελο «Settings») και εκκινούμε την διαδικασία της ψηφοφορίας με την δημιουργία του smart contract στο Ethereum blockchain.

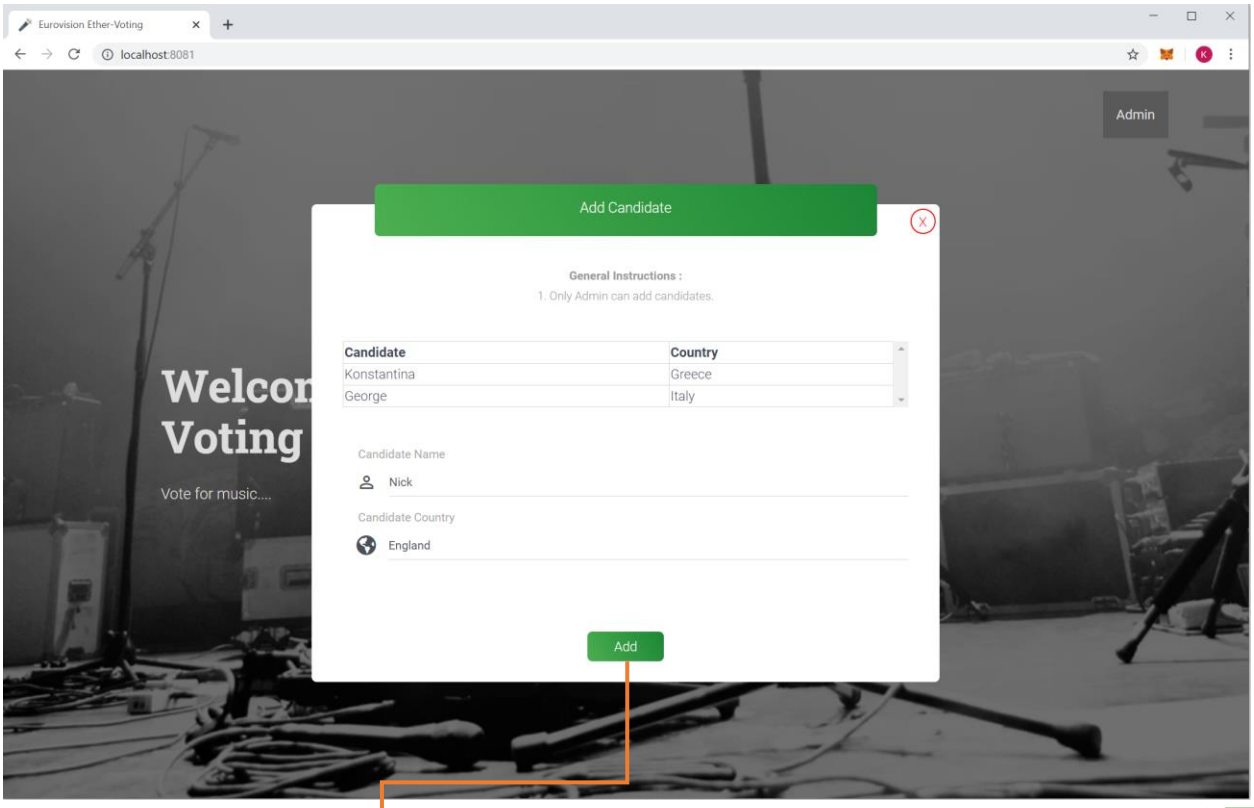
Επίδειξη λειτουργικότητας εφαρμογής

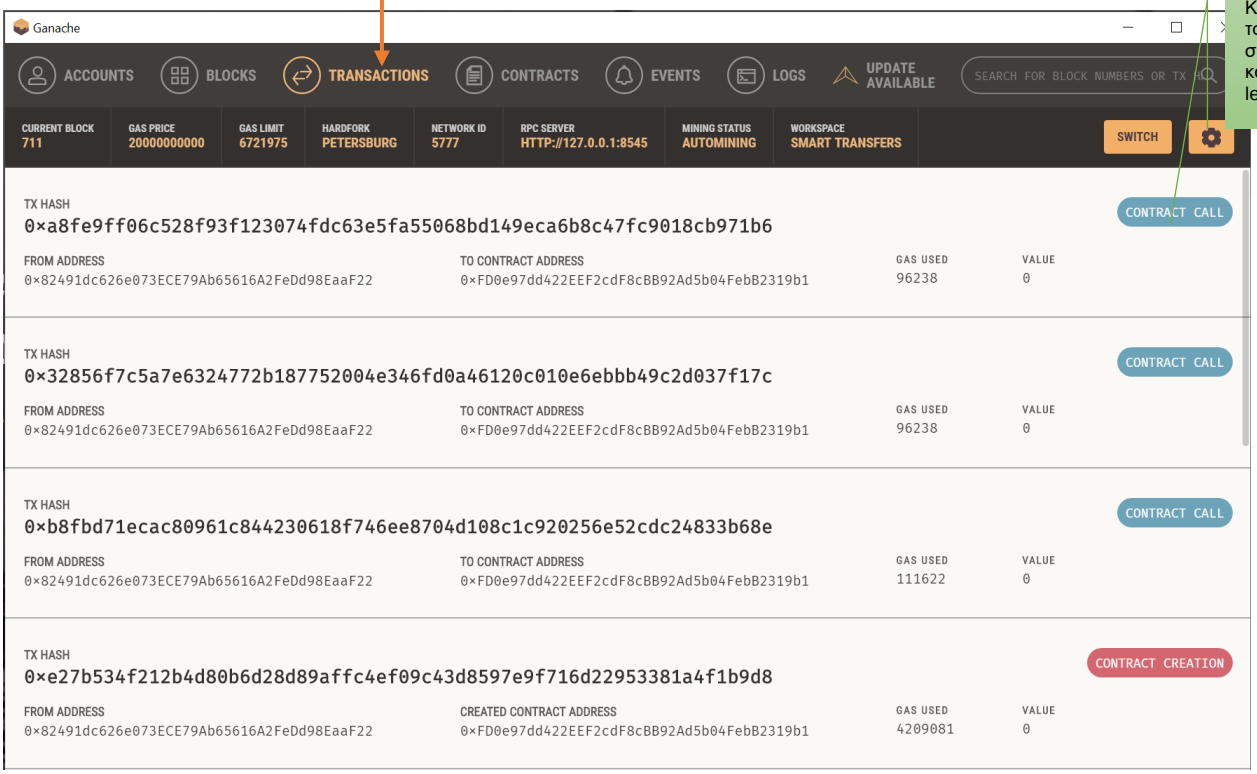




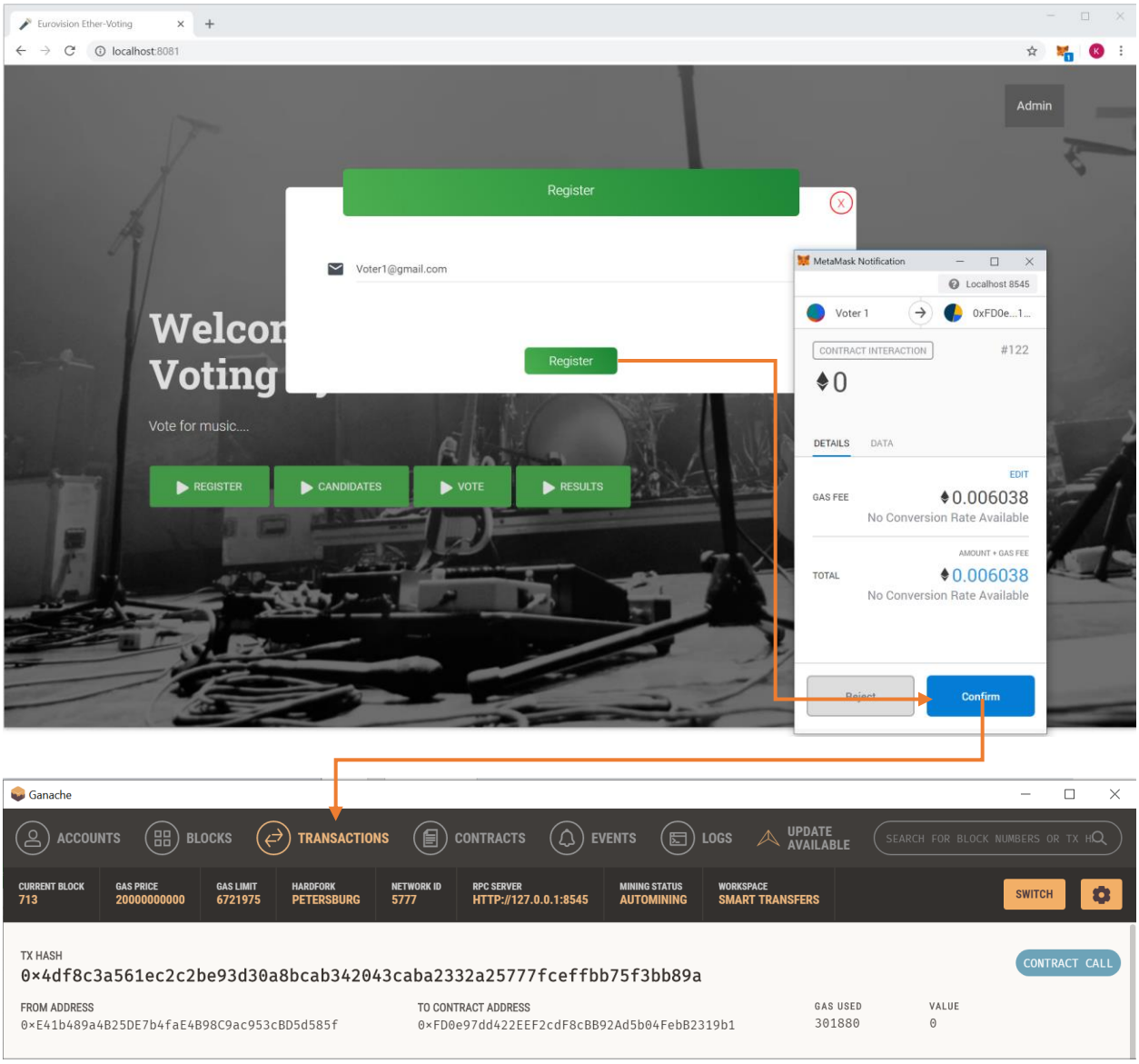

Eurovision Ether-Voting
localhost:8081
Admin
Welcon
Voting
Vote for music....
Create Smart Contract
General Instructions :
1. Only Chairman can create the Smart Contract.
Election Official Name
0x82491dc626e073ECE79Ab65616A2FeDd98EaaF22
Proposal
Let's Vote for Eurovision 2020!
Create
Η συμπλήρωση των πεδίων του Constructor που είναι προ απαιτούμενα για την δημιουργία του smart contract (Contract Creation)

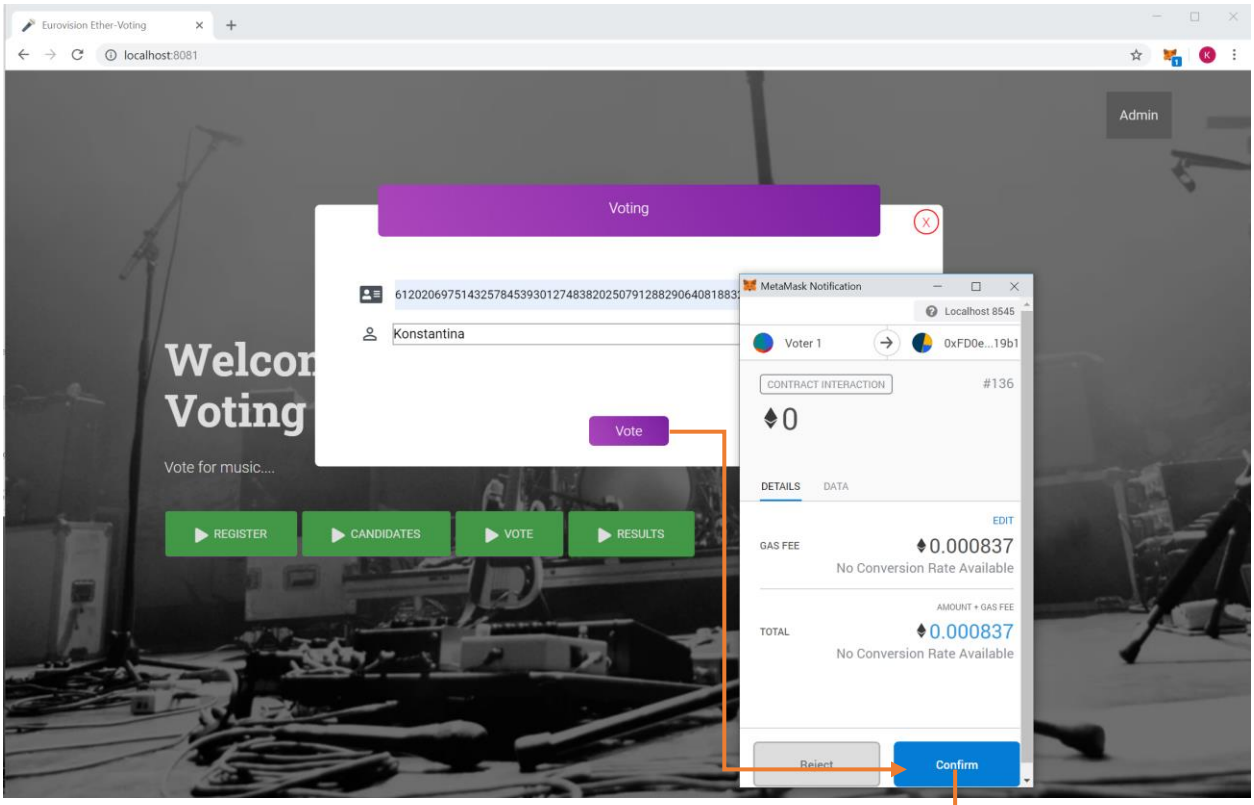

Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
UPDATE AVAILABLE
SEARCH FOR BLOCK NUMBERS OR TX HA
CURRENT BLOCK 708
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE SMART TRANSFERS
SWITCH
TX HASH
0×e27b534f212b4d80b6d28d89affc4ef09c43d8597e9f716d22953381a4f1b9d8
CONTRACT CREATION
FROM ADDRESS
0×82491dc626e073ECE79Ab
Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
UPDATE AVAILABLE
SEARCH FOR BLOCK NUMBERS OR TX HA
CURRENT BLOCK 708
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE SMART TRANSFERS
SWITCH
← BACK
BLOCK 708
GAS USED 4209081
GAS LIMIT 6721975
MINED ON 2019-12-11 22:48:50
BLOCK HASH 0×de2dc73354657f44ef22bd0c0dea361425c9f7ddb3e58a5cf81d90d00d5bf43d
TX HASH
0×e27b534f212b4d80b6d28d89affc4ef09c43d8597e9f716d22953381a4f1b9d8
CONTRACT CREATION
FROM ADDRESS
0×82491dc626e073ECE79Ab65616A2FeDd98EaaF22
CREATED CONTRACT ADDRESS
0×FD0e97dd422EEF2cdF8cBB92Ad5b04FebB2319b1
GAS USED
4209081
VALUE
0
Στο Ganache καταχωρείται η δημιουργία του smart contract ως ένα νέο transaction με την ένδειξη Contract Creation

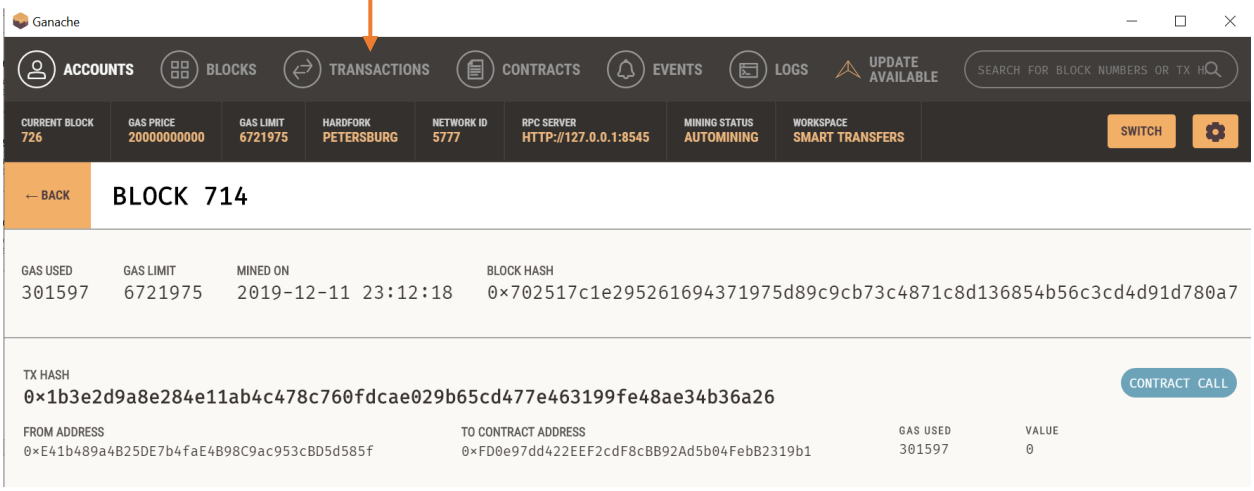


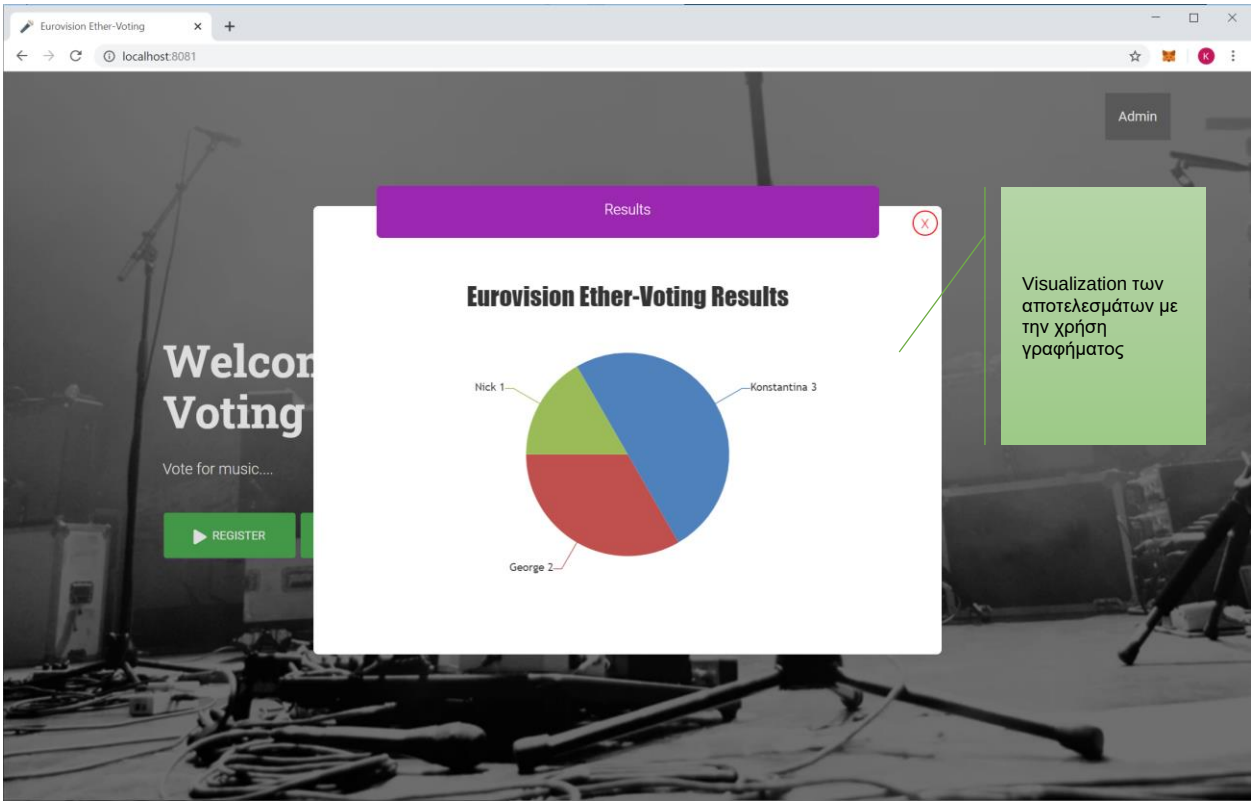


Κατά την «χρήση» του συγκεκριμένου συμβολαίου καταχωρούνται στο ledged contract calls


Eurovision Ether-Voting
localhost:8081
Admin
Register
Voter1@gmail.com
Register
Welcon
Voting
Vote for music....
REGISTER
CANDIDATES
VOTE
RESULTS
MetaMask Notification
Localhost 8545
Voter 1
0xFD0e...1...
CONTRACT INTERACTION
#122
0
DETAILS
DATA
EDIT
GAS FEE
0.006038
No Conversion Rate Available
AMOUNT + GAS FEE
TOTAL
0.006038
No Conversion Rate Available
Reject
Confirm
Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
UPDATE AVAILABLE
SEARCH FOR BLOCK NUMBERS OR TX H
CURRENT BLOCK 713
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE SMART TRANSFERS
SWITCH
TX HASH
0x4df8c3a561ec2c2be93d30a8bcab342043caba2332a25777fceffbb75f3bb89a
CONTRACT CALL
FROM ADDRESS
0xE41b489a4B25DE7b4faE4B98C9ac953cBD5d585f
TO CONTRACT ADDRESS
0xFD0e97dd422EEF2cdF8cBB92Ad5b04FebB2319b1
GAS USED
301880
VALUE
0


Eurovision Ether-Voting
localhost:8081
Admin
Voting
Konstantina
Vote
Welcon
Voting
Vote for music....
REGISTER
CANDIDATES
VOTE
RESULTS
MetaMask Notification
Localhost 8545
Voter 1
0xFD0e...19b1
CONTRACT INTERACTION
#136
0
DETAILS
DATA
EDIT
GAS FEE
0.000837
No Conversion Rate Available
AMOUNT + GAS FEE
TOTAL
0.000837
No Conversion Rate Available
Reject
Confirm


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
UPDATE AVAILABLE
SEARCH FOR BLOCK NUMBERS OR TX HA
CURRENT BLOCK 726
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE SMART TRANSFERS
SWITCH
← BACK
BLOCK 714
GAS USED 301597
GAS LIMIT 6721975
MINED ON 2019-12-11 23:12:18
BLOCK HASH 0x702517c1e2952616943719 75d89c9cb73c4871c8d136854b56c3cd4d91d780a7
TX HASH
0x1b3e2d9a8e284e11ab4c478c760fdcae029b65cd477e463199fe48ae34b36a26
CONTRACT CALL
FROM ADDRESS
0xE41b489a4B25DE7b4faE4B98C9ac953cBD5d585f
TO CONTRACT ADDRESS
0xFD0e97dd422EEF2cdF8cBB92Ad5b04FebB2319b1
GAS USED
301597
VALUE
0


Eurovision Ether-Voting
localhost:8081
Admin
Welcon
Voting
Vote for music....
REGISTER
Results
Eurovision Ether-Voting Results
Nick 1
Konstantina 3
George 2
Visualization των αποτελεσμάτων με την χρήση γραφήματος

Συμπέρασμα

Η χρήση του Ethereum ως πλατφόρμα ανάπτυξης αποκεντρωμένων εφαρμογών αναμένεται να μεταβάλει το αρχιτεκτονικό μοντέλο.

Είναι ακόμα στα αρχικά του στάδια κρίνοντας όμως από την ταχύτητα με την οποία μεταβάλλονται οι εκδόσεις της solidity και την συνεχώς αυξανόμενη βιβλιογραφία εκτιμώ ότι εντάσσεται μέσα στα πλαίσια του next big think και ότι είναι από τις τεχνολογίες που θα μας απασχολήσουν τα επόμενη έτη.

Το Blockchain ξεκίνησε για την υποστήριξη της λειτουργίας του Bitcoin αλλά ήδη έχουμε ήδη δει ότι δεν υποστηρίζει μόνο αυτό, καθώς με την είσοδο του Ethereum Blockchain και την ενσωμάτωση των Έξυπνων Συμβολαίων (Smart Contracts) είδαμε να υποστηρίζεται και μια εναλλακτική λειτουργία, αντίστοιχη με αυτή του back-end μιας οποιαδήποτε εφαρμογής.

Η υλοποίηση της εφαρμογής του voting που παρουσιάστηκε, είναι μια απλή περίπτωση proof of concept που όμως αποδεικνύει ότι μπορεί να λειτουργήσει άνετα ως μια βάση δεδομένων στην οποία καταχωρούνται σε χρονολογική σειρά και με κρυπτογραφημένη μορφή όλες οι συναλλαγές που πραγματοποιούνται από το frond end - και αυτή είναι μια από τις αναγνωρισμένες δυνατότητες.

Υπάρχουν ακόμα σημεία βελτίωσης τα οποία η κοινότητα του Blockchain αντιμετωπίζει καθημερινά όμως για εμένα προσωπικά αποτελεί μια ενδιαφέρουσα πρόκληση, καθώς φαίνεται να μπορεί να αποτελέσει, με ασφάλεια, μια πλατφόρμα υλοποίησης εφαρμογών και «εκτός συνόρων».

Βιβλιογραφικές αναφορές

[1] **Bitcoin: A Peer-to-Peer Electronic Cash System,** URL: https://bitcoin.org/en/bitcoin-paper

[2] **Η Ευρωπαϊκή Επιτροπή εγκαινιάζει το παρατηρητήριο - φόρουμ της ΕΕ για την τεχνολογία blockchain URL :** https://ec.europa.eu/commission/presscorner/detail/el/IP_18_521

[3] **Virtual currency schemes – a further analysis,** URL: https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemesen.pdf

[4] **Ethereum for Dummies,** by Michael G. Solomon, ISBN: 9781119474128

[5] **Ethereum is a global, open-source platform for decentralized applications.** URL : https://ethereum.org/

[6] **Bitcoin: A Peer-to-Peer Electronic Cash System,** by S. Nakamoto, 2008. Available on: https://bitcoin.org/bitcoin.pdf.

[7] **Blockchain for Dummies, 2nd Edition**, By: Tiana Laurence, John Wiley & Sons (US) © 2019, ISBN 9781119555018

[8] **A Survey on Long-Range Attacks for Proof of Stake Protocols**, Evangelos Deirmentzoglou, Georgios Papakyriakopoulos, Constantinos Patsakis, IEEE Access 7: 28712-28725 (2019)

[9] **Business Innovation Through Blockchain: The B3 Perspective,** By: Vincenzo Morabito, Springer © 2017, ISBN 9783319484778

[10] **Exploring Blockchain, presentation,** Instructor Jamie Campbell, viewed on percipio.com

[11] **Blockchain-Internet of Transaction: A Handbook for Blockchain Beginners,** by: Dipender Bhamah, Gursimran Oberoi, BPB Publications © 2018, ISBN: 9789386551962

[12] **A systematic literature review of blockchain-based applications: Current status, classification and open issues**, Fran Casino, Thomas K. Dasaklis, Constantinos Patsakis, Telematics and Informatics 36: 55-81 (2019)

[13] https://www.epixeiro.gr/article/65273

[14] **Getting Started With Ethereum and Building Basic Dapp using Truffle, Metamask & Ganache-CLI,** URL : https://medium.com/coinmonks/getting-started-with-ethereum-and-building-basic-dapp-ebb681fb3748

[15] **Smart Contracts Described by Nick Szabo 20 Years Ago Now Becoming Reality** URL : https://www.nasdaq.com/articles/smart-contracts-described-nick-szabo-20-years-ago-now-becoming-reality-2016-04-26

[16] **Gas Costs from Yellow Paper** -- EIP-150 Revision (1e18248 - 2017-04-12), URL : https://github.com/djrtwo/evm-opcode-gas-costs

[17] https://hackernoon.com/ether-purchase-power-df40a38c5a2f

[18] **Implementing Smart Contracts Using Ethereum**, presentation, Instructor Niranjan Pandey, viewed on percipio.com

[19] **Building Ethereum Dapps: A Beginner's Guide to Building Blockchain Solutions**, by Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Publisher: Apress © 2018, ISBN: 9781484234433

[20] **Towards self-sovereign identity using blockchain technology**, Baars, D. S.. MS thesis. University of Twente, 2016.

[21] **Blockchain for identity management**, Jacobovitz, Ori. The Lynne and William Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva Google Scholar 1 (2016): 9.

[22] **Building Ethereum Dapps : Decentralized Applications on the Ethereum Blockchain**, Roberto Infante, Foreword by Thomas Bertani, March 2019, ISBN 9781617295157

[23] https://gist.github.com/wmh/dcf988314d0ca5e325a1bb504e7e3323