Κατανεμημένα συστήματα ασφάλειας και εφαρμογές

Μάλαμας Ευάγγελος
bagmalamas@unipi.gr

Περιεχόμενα

Κλασσικές Συναλλαγές – Εξέλιξη τους

- Αρχικά γίνονταν μέσω ανταλλαγών φυσικών αντικειμένων (barter trade)
  π.χ. 2 βόδια για 10 πήλινα αγγεία
- Κατόπιν μέσω νομισμάτων (χρυσά, ασημένια, χάλκινα)
- Μετά μέσω «Παραστατικών νομισμάτων» (fiat currency)

Κλασσικές Συναλλαγές – Τράπεζες

- Όλα τα προηγούμενα απαιτούν την φυσική παρουσία των δύο μερών (για να υπάρχει «Πίστη»-Trust)
- Διαφορετικά χρειάζεται κάποιος ενδιάμεσος που να παρέχει την Πίστη
- Τράπεζες

Τράπεζα και Μειονεκτήματα

- Πρόκειται για κεντρικοποιημένο σύστημα (τράπεζα)
- Τρίτος (Μεσάζων) που εγγυάται την Πίστη των συναλλαγών αλλά προσθέτει:
  - Κόστος (για τις παρεχόμενες υπηρεσίες)
  - Χρόνο (για την εκκαθάριση συναλλαγών)
  - Έχει απόλυτη εξουσία

Οι Τράπεζες δεν είναι το μόνο Παράδειγμα

- Οίκοι εκκαθάρισης (clearing houses)
  - Χρηματιστηριακές Συναλλαγές
  - Διαχείριση πνευματικών δικαιωμάτων-Digital right management
- Υπηρεσίες Συμβολαίων
- Υπηρεσίες Μητρώων

Blockchain – Βασικά Χαρακτηριστικά (1)

- Από τη μεριά των επιχειρήσεων: Λογιστικό Βιβλίο («Καθολικό») συναλλαγών διαμοιραζόμενο, αποκεντρωμένο και ανοικτό
  - Η Βάση Δεδομένων που το υλοποιεί επιτρέπει εγγραφές τύπου append μόνο
  - Καμία εγγραφή δεν μπορεί να αλλάξει ή να διαγραφεί
  - Όλες οι εγγραφές αντιγράφονται σε όλα τα αντίγραφα που υπάρχουν στους διάφορους κόμβους

Blockchain – Βασικά Χαρακτηριστικά (2)

- Πρόκειται για ένα επίπεδο πάνω από το διαδίκτυο
- Συνυπάρχει στο Διαδίκτυο με άλλες τεχνολογίες
- Ένας από τους στόχους είναι η πλήρης αποκέντρωση στο πλαίσιο του ανοικτού TCP/IP

Νοήματα όρων - 1

- Κατανεμημένο Σύστημα
  - Συγκεντρωτικά Κατανεμημένο Σύστημα

    1 κύριος κόμβος που διαιρεί και διανέμει τις εργασίες ή τα δεδομένα στους υπόλοιπους κόμβους(π.χ. Hadoop)
  - Αποκεντρωμένα Κατανεμημένο Σύστημα

    Δεν υπάρχει κύριος κόμβος, η επεξεργασία είναι κατανεμημένη
    Το blockchain είναι κάτι τέτοιο

Νοήματα όρων - 2

- Συγκέντρωση VS Αποκέντρωση
  - Τεχνική Αρχιτεκτονική
  - Πολιτική Αντίληψη (ο έλεγχος που ένας/πολλοί έχουν σε ένα σύστημα)
  - Λογική Αντίληψη (Το πως φαίνεται να είναι)

Π.χ. αν το κόψουμε στην μέση τα επι μέρους συστήματα συνεχίζουν να λειτουργούν?

<u>Για παράδειγμα</u>

Εταιρεία: Συγκεντρωτική ως προς τους 3 τρόπους

BitTorrent: Αποκεντρωτικό ως προς τους 3 τρόπους

Δίκτυο Διανομής Περιεχομένου (CDN): Αποκεντρωτικό αρχιτεκτονικά και λογικά,
αλλά συγκεντρωτικό πολιτικά

Blockchain: Αποκεντρωτικό αρχιτεκτονικά και πολιτικά, αλλά συγκεντρωτικό λογικά
Υπάρχει μια μόνον κατάσταση (συμφωνημένη από κοινού) και το όλο σύστημα συμπεριφέρεται ως 1 Η/Υ

Μειονεκτήματα Συγκεντρωτικών Συστημάτων

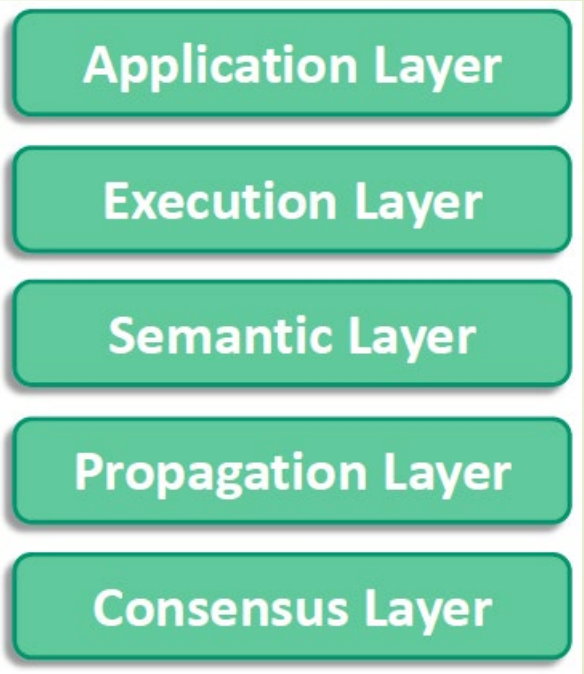
- Μοναδικό (κεντρικό) σημείο αποτυχίας – Single Point of Failure
- Πιο ευάλωτα σε επιθέσεις (άρα λιγότερο ασφαλή)
- Συγκέντρωση εξουσίας μπορεί να οδηγήσει σε αντιδεοντολογικές πρακτικές (π.χ. ένας malicious admin)
- Κλιμάκωση είναι συχνά δύσκολη

Πλεονεκτήματα Αποκεντρωτικών Συστημάτων

- Αν και πιο δύσκολα στον σχεδιασμό, υλοποίηση και διαχείριση:
  - Δεν έχουν ένα (κεντρικό) σημείο αποτυχίας
  - Άρα πιο σταθερά και ανεκτικά σε σφάλματα
  - Ανθεκτικότερα σε επιθέσεις
  - Συμμετρικά όπου όλοι οι κόμβοι έχουν την ίδια εξουσία, οπότε λιγότερες αντιδεοντολογικές πρακτικές και πιο δημοκρατική λειτουργία

Επίπεδα του Blockchain

- Δεν υπάρχει καθολικά αποδεκτή ταξινόμηση
- Αυτό που εμφανίζεται εδώ δεν πρέπει να συγχέεται με την στοίβα TCP/IP



Application Layer

- Εδώ ανήκει η κωδικοποίηση των επιθυμητών λειτουργιών
  σε ανάπτυξη εφαρμογής για τον τελικό χρήστη

- Περιλαμβάνει συνήθη στοίβα εργαλείων ανάπτυξης:
  Π.χ. Server/client προγραμματιστικές δομές, scripting, API, πλαίσια ανάπτυξης κ.ά.

Execution Layer

Εδώ εκτελούνται οι εντολές που ζητώνται από το Application Layer

- Απλές εντολές ή πολλαπλές (π.χ. για έξυπνα συμβόλαια)
- Bitcoin: απλά script που δεν είναι Turing-complete
- Ethereum και Hyperledger: πολύπλοκες εντολές
  - *Ethereum code σε Solidity και Ethereum Virtual Machine*
  - *Hyperledger chaincode σε Java ή Go και docker images*

Semantic Layer

- Επικύρωση των συναλλαγών που εκτελούνται παραπάνω
-
- Εδώ ορίζονται οι κανόνες
  - Bitcoin: δεν έχει λογαριασμούς
  - Ethereum: έχει λογαριασμούς
    - smart contract: ειδικός τύπος λογαριασμού με εκτελέσιμο κώδικα και καταστάσεις

- οι δομές δεδομένων
  π.χ. Merkle trees

- και πώς συνδέονται τα block μεταξύ τους

Propagation Layer

- Ασχολείται με την επικοινωνία P2P
- Αρκετά ζητήματα σχετικά με τη διάδοση των συναλλαγών/block:

  - Πώς οι κόμβοι βρίσκουν ο ένας τον άλλον
  - Πώς συνομιλούν (latency)
  - Πώς συγχρονίζονται (ως προς τις συναλλαγές)

Consensus Layer

- Ασχολείται με το πώς όλοι κόμβοι συμφωνούν σε μία συνεπή κατάσταση του «Καθολικού»

- Διάφοροι τρόποι επίτευξης τέτοιας ομοφωνίας:

  - Εξαρτώνται από το Σενάριο Χρήσης
  - Στα Ethereum και Bitcoin, μέσω του "mining"
  - Μηχανισμοί/πρωτόκολλα ομοφωνίας:
    - Proof of Work (PoW),
    - Proof of Stake (PoS),
    - delegated PoS (dPoS),
    - Practical Byzantine Fault Tolerance (PBFT)
    - Proof of Authority (PoA)
    - Proof of Elapsed Time

Τι συνδυάζεται στο Blockchain?

- Κρυπτογραφία
  (για ασφάλεια στις συναλλαγές)
- Θεωρία Παιγνίων
  (για να δώσει λύσεις στο πρόβλημα των Βυζαντινών Στρατηγών- Η βάση για την λειτουργία των μηχανισμών συναίνεσης)
- Επιστήμη των Υπολογιστών
  (παρέχει χρήσιμες δομές και μηχανισμούς)

Θεωρία Παιγνίων - 1

- Συνήθως αναφέρεται σε παιχνίδια με 2 ή περισσότερους παίκτες που έχουν κάποια στρατηγική συμπεριφορά

  - Δικηγόροι αντίπαλοι σε δικαστήριο
  - Πολιτική εκλογή
  - Σηματοδότες κυκλοφορίας (φανάρια)
  - Απόρριψη επιλογής για πρόσληψη λόγω διαφοράς μεταξύ προσφοράς και ζήτησης

Θεωρία Παιγνίων - 2

- Άρα παιχνίδι είναι οποιαδήποτε περίπτωση όπου υπάρχουν συσχετιζόμενες λογικές επιλογές

  - Οι διαθέσιμες προοπτικές για έναν παίκτη εξαρτώνται όχι μόνον από τις προσωπικές του προτιμήσεις, αλλά και από τις επιλογές που κάνουν οι υπόλοιποι σε μια δεδομένη κατάσταση

- Η Θεωρία παιγνίων αφορά τη μελέτη στρατηγικών για πολύπλοκα παιχνίδια

  - Η τέχνη του να κάνουμε τη βέλτιστη κίνηση ή να επιλέγουμε μια τέλεια στρατηγική σε μια δεδομένη κατάσταση με βάση κάποιο στόχο του παιχνιδιού (συνθήκες νίκης)

  - Για να το πετυχαίνουμε αυτό, πρέπει να καταλαβαίνουμε τη στρατηγική του αντιπάλου, καθώς και ποιά σκέφτεται ο αντίπαλος ότι θα είναι η επόμενη κίνησή μας

Θεωρία Παιγνίων - 3

- Παράδειγμα #1: 1 παγωτό κρέμα, 1 παγωτό βανίλια και δύο αδέλφια, όπου ο μεγάλος γνωρίζει ότι όποια επιλογή να κάνει, θα την επιλέξει και ο μικρός

  - Στρατηγική μεγάλου: Αν θέλει την κρέμα, επιλέγει την βανίλια, το ίδιο κάνει ο μικρός, οπότε ο μεγάλος «υποχωρεί» κερδίζοντας την πραγματική του επιθυμία.

  - Συνολικό όμως αποτέλεσμα "win-win" αφού και οι δύο κέρδισαν αυτό που ήθελαν

Τύποι Παιχνιδιών

- Πολλοί τρόποι για να ταξινομήσουμε τα παιχνίδια
  - zero-sum/non-zero-sum, simultaneous/sequential, κλπ.

- Εδώ μας ενδιαφέρουν τα *cooperative*/***noncooperative***

  - Οι παίκτες συνεργάζονται μεταξύ τους (συμμαχίες) και μπορεί να υπάρχει μια εξωτερική δύναμη για να επιβεβαιώνει κάτι τέτοιο

  - Στον αντίποδα, παίζουν **μόνον** ως άτομα, κοιτώντας το στενό τους συμφέρον και δεν υπάρχει εξωτερική δύναμη για να τους υποχρεώσει σε συνεργασία

Να έχουμε κατά νου ότι ισχύει:

Σε οποιοδήποτε μη-συνεργατικό παίγνιο, όπου οι παίκτες γνωρίζουν τις στρατηγικές των άλλων, υπάρχει τουλάχιστον ένα σημείο ισορροπίας όπου όλοι οι παίκτες παίζουν με τις βέλτιστες στρατηγικές τους για τα μέγιστα κέρδη και κανένας παίκτης δεν θα κέρδιζε με το να άλλαζε την στρατηγική του

Πρόβλημα των Βυζαντινών Στρατηγών

- Ο Βυζαντινός στρατός επιτίθεται εναντίον μιας πόλης, αλλά είναι χωρισμένος σε 2 ή περισσότερες μονάδες (με 1 στρατηγό η κάθε μία)
- Για να νικήσει ο στρατός πρέπει όλες οι μονάδες να επιτεθούν ταυτόχρονα
- Το πρόβλημα είναι πως να φτάσουν σε ομοφωνία
  - είτε όλοι μαζί να επιτεθούν
  - είτε όλοι μαζί να υποχωρήσουν

Πρακτικά Ζητήματα

- Χρειάζεται ομοφωνία μεταξύ των στρατηγών
  θα πρέπει να επιτεθούν αν τουλάχιστον 3 από τους 5 επιθυμούν επίθεση, αλλιώς να υποχωρήσουν
- Αφού δεν υπάρχει κεντρικός συντονισμός, εάν 1 από τους 5 είναι προδότης,
  θα ψήφιζε (για να ηττηθούν):
  - επίθεση με τους στρατηγούς που επιθυμούν επίθεση
  - υποχώρηση με εκείνους που επιθυμούν υποχώρηση
- Τι θα συμβεί αν 2 στρατηγοί και ο προδότης ψηφίσουν επίθεση;

Πιο πολύπλοκα ζητήματα

- Εάν υπάρχουν περισσότεροι προδότες;
- Πώς θα μπορούσε να επιτευχθεί συντονισμός των στρατηγών μέσω μηνυμάτων;
- Εάν ο αγγελιοφόρος συλληφθεί, σκοτωθεί ή δωροδοκηθεί από το διοικητή της πόλης;
- Εάν ο στρατηγός-προδότης πλαστογραφήσει ένα διαφορετικό μήνυμα και εξαπατήσει τους υπόλοιπους;
- Πώς να διακρίνουμε τους τίμιους από τους προδότες;

Ανάλογα σενάρια

- Πώς καταλήγει σε ομοφωνία μια ομάδα ανθρώπων σχετικά με την ημερήσια διάταξη για ψηφοφορία;
- Πως διατηρείται η συνεπής κατάσταση σε μια κατανεμημένη ή αποκεντρωμένη βάση δεδομένων;
- Πως διατηρείται η συνεπής κατάσταση των αντιγράφων blockchain μεταξύ των κόμβων ενός δικτύου;

Οι λύσεις μπορεί να είναι πολύ διαφορετικές για διαφορετικές καταστάσεις

Σχετικά με το Blockchain

- Μια δομή δεδομένων:
  - Αλυσίδα από μπλοκ (block) συνδεδεμένα μεταξύ τους

- Τι είναι το “block”;
  - Δεδομένα για μια ή περισσότερες συναλλαγές

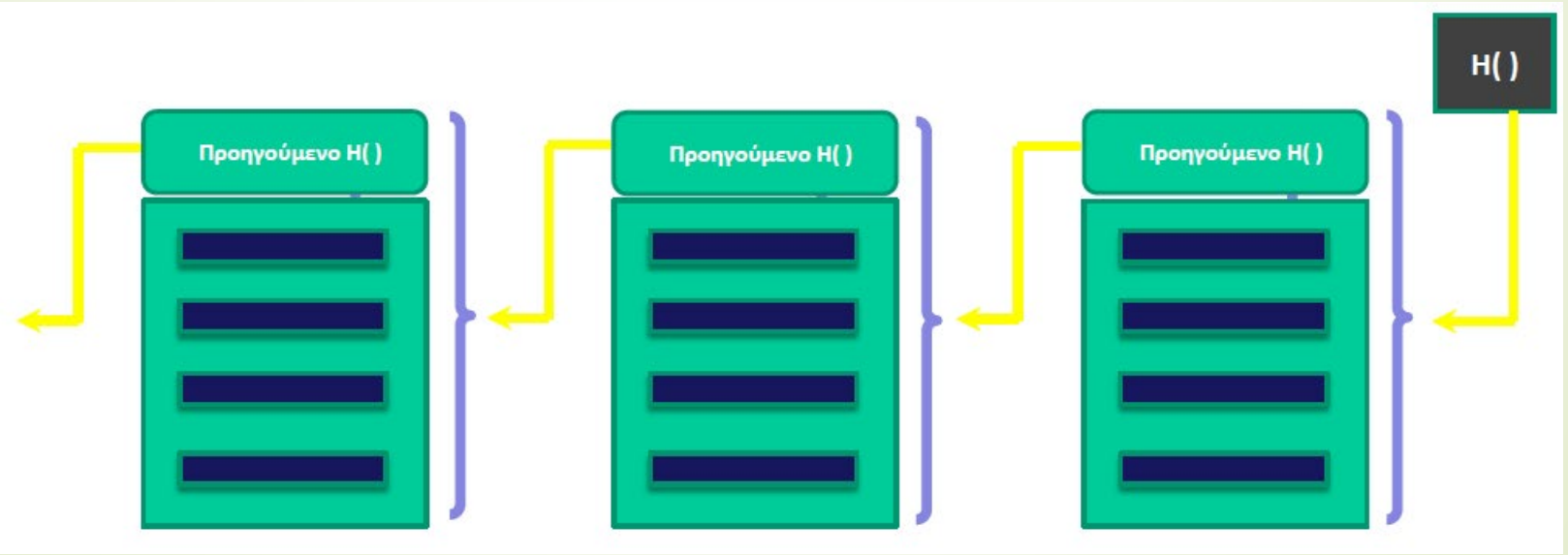
- Βασικό δομικό στοιχείο είναι οι δείκτες κατακερματισμού (hash pointer)

Τι είναι τα Hash Pointer ?

- Περίπου όπως οι δείκτες στις συνδεδεμένες λίστες
- Είναι κάτι σαν «αποτύπωμα» των δεδομένων
- Εδώ όμως δείχνουν στο προηγούμενο block και όχι στο επόμενο
- Επίσης παρέχουν εγγύηση ότι τα αντίστοιχα δεδομένα δεν έχουν αλλοιωθεί

Blockchain: Μοναδική πηγή αλήθειας

- Στη κεφαλίδα (header) κάθε block αποθηκεύεται η τιμή Hash του προηγούμενου block, ώστε να υπάρχει διασφάλιση για τη μη αλλοίωση των δεδομένων του προηγούμενου block



Παρατηρήσεις

- Η τιμή Hash() αφορά όλο το προηγούμενο block, μαζί με το προηγούμενο Hash()
-
- Κάθε block δείχνει στο προηγούμενο block που ονομάζεται “parent block” (γονικό)
- Το πρωταρχικό block ονομάζεται “genesis block”
- Είναι πρακτικά αδύνατο να αλλάξει κάποιος τα δεδομένα σε ένα block

Λεπτομέρειες

- Εάν κάποιος αλλάξει το περιεχόμενο ενός block;
  - *Δεν ταιριάζουν οι τιμές Hash()*

- Τι γίνεται εάν κάποιος αλλάξει και τις τιμές Hash();
  - *Θα σπάσει την «αλυσίδα», εκτός αν αλλάξει ολόκληρη την «αλυσίδα»!*

https://andersbrownworth.com/blockchain/

Δένδρα Merkle

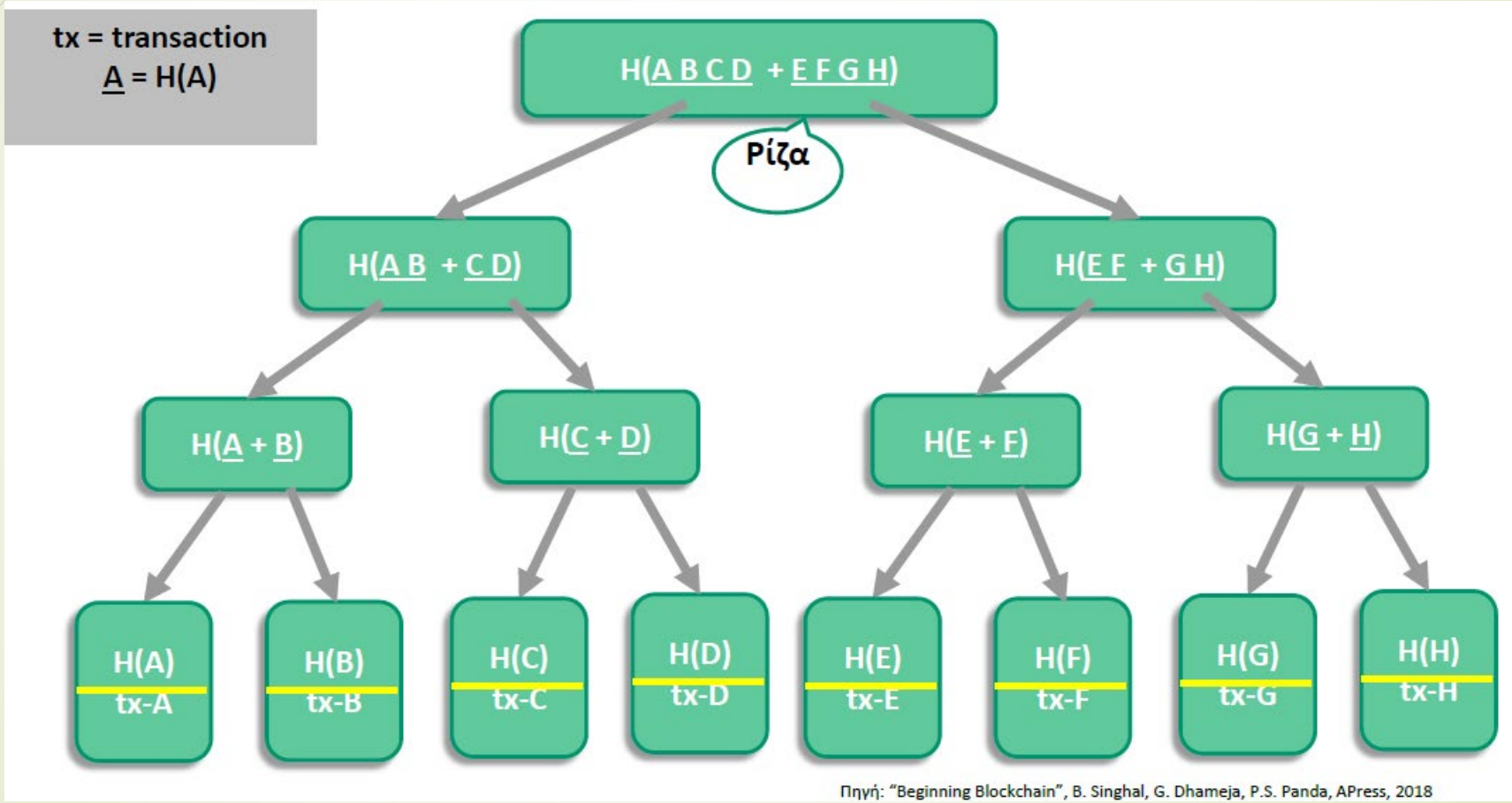
- Από τον Ralph Merkle που τα επινόησε
- Δένδρο Merkle: δυαδικό δένδρο με δείκτες hash (binary hash tree)
- Δομή δεδομένων που χρησιμοποιείται στο Bitcoin

- Δομούνται με κατακερματισμό συζευγμένων δεδομένων (π.χ. συναλλαγές στο επίπεδο φύλλου), κατόπιν ξανά κατακερματισμό των αποτελεσμάτων μέχρι το κόμβο-ρίζα, που ονομάζεται ρίζα Merkle

  - Καταστρώνονται από-κάτω-προς-τα-επάνω (bottom-up)

  - Π.χ. στο Bitcoin τα φύλλα είναι πάντα συναλλαγές ενός μοναδικού block του blockchain

Παράδειγμα Δένδρου Merkle



Πηγή: "Beginning Blockchain", B. Singhal, G. Dhameja, P.S. Panda, APress, 2018

Χαρακτηριστικά των Δένδρων Merkle

- Είναι απαραβίαστα (tamper-proof)

  - Παραβίαση σε οποιοδήποτε επίπεδο θα οδηγούσε σε μη ταίριασμα των Hash που είναι αποθηκευμένα στα ανώτερα επίπεδα *και μέχρι τη ρίζα Merkle*

- Εγγυώνται την ορθή ακολουθία των συναλλαγών

  - Αφού πάλι οι τιμές Hash (έως και την ρίζα) θα άλλαζαν

Βασικά ερωτήματα

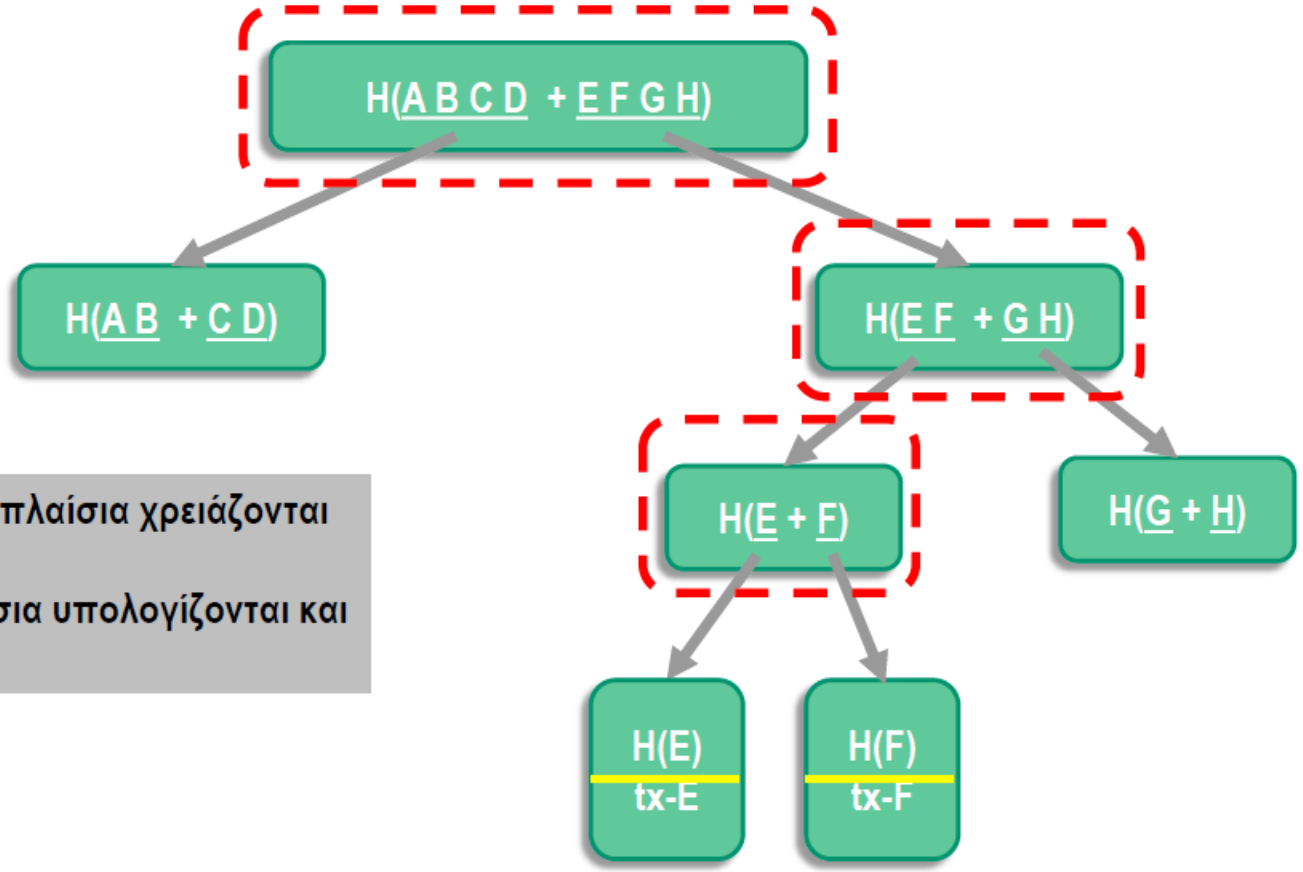
- *Εάν δεν υπάρχει ζυγό πλήθος συναλλαγών;*

  Λύση: επαναλαμβάνεται το Hash της τελευταίας συναλλαγής
  Παραμένει η ίδια συναλλαγή, άρα δεν υπάρχει double-spent
  ή επανάληψη συναλλαγής, οπότε έχουμε ισορροπημένο δένδρο

- *Για αναζήτηση παρελθούσας συναλλαγής απ' ευθείας ή μέσω της τιμής hash της;*

  Λύση: επιβεβαίωση αν η συναλλαγή (φύλλο) ανήκει στο δένδρο
  Merkle χρειαζόμαστε μόνον ένα μέρος του δένδρου
  Το πλήθος των υπολογισμών για *n* συναλλαγές είναι **log *n***

Παράδειγμα Επιβεβαίωσης Συναλλαγής F



Μόνο τα κανονικά πλαίσια χρειάζονται

Τα διάστικτα πλαίσια υπολογίζονται και ταιριάζονται

Πηγή: "Beginning Blockchain", B. Singhal, G. Dhameja, P.S. Panda, APress, 2018

Blockchain και Δένδρα Merkle

- Εάν οι συναλλαγές είναι πάρα πολλές, μπορούμε να έχουμε ταχύτερη επιβεβαίωσή τους; Λύση:

  - Κάθε μπλοκ ήδη περιλαμβάνει το Hash του προηγούμενου
  - Αποθήκευση μόνο της ρίζας Merkle root σε αυτό
    - Αντί να αποθηκεύονται τα hashes όλων των Tx

- Άρα, για επιβεβαίωση συναλλαγής στο μπλοκ 22456: παίρνουμε τις συναλλαγές του μπλοκ και με τη ρίζα Merkle
  - επιβεβαιώνουμε το δένδρο Merkle
  - αλλά και τη σειρά των συναλλαγών

Ελαφροί κόμβοι (light nodes)

- Κόμβοι που δεν περιέχουν ολόκληρα τα block συναλλαγών, αλλά μόνο τις επικεφαλίδες
- Επιβεβαίωση ότι κάποια συναλλαγή συνέβη στο παρελθόν:

  - Επιβεβαίωση της συναλλαγής ως μέρος ενός μπλοκ και Επιβεβαίωση του μπλοκ ως μέρους του blockchain,
  - Δεν ζητά να κατεβάσει όλες τις συναλλαγές από το δίκτυο, αλλά ζητά το Hash του μπλοκ και το Hash της συναλλαγής
  - Οι κόμβοι που έχουν αυτή την πληροφορία απαντούν με τη διαδρομή Merkle που σχετίζει τη συναλλαγή με το μπλοκ

- Σημαντική η ύπαρξη τους για λόγους ταχύτητας

Πλήρεις κόμβοι (Full nodes)

- Θεμελιώδεις για την ύπαρξη του συστήματος
- Περιέχουν το πλήρες ιστορικό των συναλλαγών από την απαρχή του συστήματος και άρα είναι οι πλέον ασφαλείς
- Δεν στηρίζονται στο δίκτυο για επικύρωση των συναλλαγών, αφού έχουν διαθέσιμο το πλήρες ιστορικό,
- Χρειάζονται μόνο τα προτεινόμενα block ώστε (αφού αυτά επικυρωθούν) να ενημερώσουν το ιστορικό τους

Επιθυμητές ιδιότητες των Λύσεων BC

- Αμεταβλητότητα (Immutability)
- Ανθεκτική σε Παραχάραξη (Forgery Resistant)
- Δημοκρατική (Democratic)
- Ανθεκτική σε Διπλοξόδεμα (Double-Spend Resistant)
- Συνεπής Κατάσταση του Καθολικού (Consistent State of the Ledger)
- Ανεκτική & Ελέγξιμη (Resilient & Auditable)

Αμεταβλητότητα-Immutability

- Η πιο επιθυμητή ιδιότητα προκειμένου να τηρείται η ατομικότητα (αμετάβλητο) των συναλλαγών blockchain

- Με τη μαζική δικτυακή μετάδοση των συναλλαγών, μετά από κάποιον χρόνο, προστίθενται περισσότερα block, καθιστώντας τες πλήρως αμετάβλητες

  - Πρακτικά ανέφικτη η μεταβολή πάρα πολλών block σε όλους τους κόμβους

Ανθεκτική σε Παραχάραξη (Forgery Resistant)

- Οι συναλλαγές είναι δημόσιες και το σύστημα αποκεντρωμένο, ευνοώντας επιθέσεις ειδικά όταν αφορά χρήμα/αξία
- Η απαιτούμενη προστασία επιτυγχάνεται με χρήση Hash (ακεραιότητα) και ψηφιακών υπογραφών (αυθεντικότητα και μη-αποποίηση)

Δημοκρατική (Democratic)

- Ως γενική αρχή, κάθε αποκεντρωμένο σύστημα P2P πρέπει να θεωρεί ισότιμους όλους τους συμμετέχοντες, με τις αποφάσεις να λαμβάνονται κατά πλειοψηφία

  - democratic by design
  - όχι πλήρως εφαρμόσιμο σε private blockchain

Ανθεκτική σε Διπλοξόδεμα (Double-Spend Resistant)

- Επιθέσεις Διπλοξοδέματος: σύνηθες φαινόμενο, ακόμα και για συναλλαγές χωρίς χρήματα.
  - Το ίδιο αντικείμενο πωλείται σε πολλαπλούς αγοραστές
  - Το υπόλοιπο λογαριασμού μας 100€, αλλά πληρώνουμε δύο ή περισσότερες αγορές μας από 90€

<u>Λύση Bitcoin: Η είσοδος κάθε συναλλαγής (όταν πληρώνουμε κάποιον) είναι έξοδος από κάποιαν άλλη</u>

- Εύκολα αντιμετωπίζεται σε συγκεντρωτικά συστήματα λόγω του ελέγχου των συναλλαγών από μια κεντρική αρχή

- Στα αποκεντρωμένα συστήματα blockchain η λύση έγκειται στον έλεγχο όλων των συναλλαγών του παρελθόντος, μέχρι το genesis block

Συνεπής κατάσταση του καθολικού (Consistent State of the ledger)

Π.χ., κόμβοι που δεν είναι συγχρονισμένοι και έχασαν κάποιες συναλλαγές, ή που επίτηδες προσπαθούν την ακύρωση κάποιας συναλλαγής

Λύση: Κατάλληλη μορφή ομοφωνίας (consensus)

- Δύσκολο αλλά κρίσιμο σημείο
- Διάφοροι μηχανισμοί

Ανεκτική (Resilient)

Το δίκτυο θα πρέπει να είναι αρκετά ανεκτικό σε (μεταξύ άλλων):

- Προσωρινές αστοχίες κόμβων
- Έλλειψη διαθεσιμότητας κάποιων κόμβων κάπου-κάπου
- Διακυμάνσεις δικτύου και απορρίψεις διακινούμενων πακέτων

Ελέγξιμη (Auditable)

- Υπάρχει από το σχεδιασμό του Blockchain ως αλυσίδα από hash

- Χρειάζεται όμως διασφάλιση ότι:
  - παραμένει η δυνατότητα ελέγχου με κάθε κόστος και
  - η επιβεβαίωση προηγούμενων συναλλαγών γίνεται ταχύτερα

Συναλλαγές στο BC - 1

- Για μεγάλο πλήθος συναλλαγών/δευτερόλεπτο επιλέγεται η ομαδοποίηση πολλών συναλλαγών σε block για
  - Σταθερότητα στο δίκτυο
  - Αντιμετώπιση του Sybil Attack

Συναλλαγές στο BC - 2

Στάδια κάθε νέας συναλλαγής (βασική μορφή):

1. Εκπέμπεται στο δίκτυο ώστε όλοι οι κόμβοι να γνωρίζουν πότε έλαβε χώρα (για αποφυγή double-spending)

2. Ελέγχεται από τους κόμβους η αυθεντικότητά της (επικυρώνεται ή απορρίπτεται)

3. Εκπέμπεται πάλι ως μέλος ομάδας (block) πολλαπλών δοσοληψιών για να ενταχθεί στο blockchain

4. Ποιος κόμβος αποφασίζει το block που ομαδοποιεί ορισμένες συναλλαγές; Ανάγκη για ομοφωνία

5. Καθώς δεν υπάρχει η έννοια του καθολικού χρόνου, η χρονοσήμανση αφορά τη σειρά άφιξης και πρόσθεσης κάθε block στο blockchain

6. Μόλις όλοι οι κόμβοι αποδεχθούν ένα block ομόφωνα, τοποθετείται στο blockchain μαζί με το Hash του αμέσως προηγουμένου block στην αλυσίδα

Συναλλαγές στο BC - 3

Σημαντικές λεπτομέρειες:

- Τα δένδρα Merkle και η όλη δομή blockchain βοηθούν σε έναν κόμβο τον αποδοτικό έλεγχο εγκυρότητας μίας συναλλαγής

- Οι υπόλοιποι κόμβοι δεν χρειάζεται να μοιράζονται πλήρη μπλοκ δεδομένων για να επιβεβαιώνεται η συμμετοχή μιας συναλλαγής σε ένα μπλοκ

- Χρειάζεται αποδοτικός τρόπος αποθήκευσης (για τις συναλλαγές και τα μεταδεδομένα τους)

Μηχανισμοί συναίνεσης (consensus mechanisms) - 1

- Για πρακτικούς λόγους εκπέμπεται στο δίκτυο, όχι κάθε μια νέα συναλλαγή, αλλά ένα block προς επικύρωση και ένταξη στο blockchain
- Ποιος κόμβος όμως προτείνει ένα block;

  - Εάν γίνεται τυχαία, τότε πρέπει να μεσολαβεί ένα ικανό χρονικό διάστημα μεταξύ προτάσεων, ώστε να μην "πέφτουν" όλες μαζί

  - Ανάγκη για μηχανισμό ομοφωνίας block by block

Μηχανισμοί συναίνεσης (consensus mechanisms) - 2

- Μηχανισμοί ομοφωνίας από τη Θεωρία Παιγνίων
- Διασφάλιση ότι οι κόμβοι κερδίζουν τα μέγιστα σεβόμενοι τους κανόνες:
  - Επιβράβευση των τιμίων και τιμωρία των κακών
  - Τι γίνεται στο Bitcoin όπου μπορεί κάποιος να έχει πολλαπλές ταυτότητες (με πραγματική ανωνυμία);
  - Δύσκολη η τιμωρία, αλλά εύκολη (και επιθυμητή!) η επιβράβευση
- Δεν μπορεί να γνωρίζει το σύστημα αν ένας επιλεγμένος κόμβος είναι κακοπροαίρετος ή τίμιος

Proof of Work (PoW)

- Αρκετά παλιός και δημοφιλής λόγω Bitcoin. Χρησιμοποιείται και στο Ethereum.
- Κεντρική ιδέα:

  - Γίνεται μια εργασία εξόρυξης (mining) πριν προταθεί ένα μπλοκ συναλλαγών στο δίκτυο

  - Δύσκολη στο να γίνει (υπολογιστικά και χρονικά), αλλά εύκολη στην επικύρωση

Στο blockchain βοηθάει με δυο τρόπους:

1. Θα χρειαστεί σίγουρα κάποιος χρόνος
2. Αν κάποιος κόμβος προσπαθήσει να εισάγει μια ψεύτικη δοσοληψία, τότε η απόρριψη του block από τους υπόλοιπους θα είναι πολύ δαπανηρή για αυτόν και εύκολα εξακριβώσιμη από τους υπόλοιπους

Proof of Work (PoW)-λεπτομέρειες

- Η δυσκολία της απαιτούμενης εργασίας θα πρέπει να είναι προσαρμοζόμενη, για να ελέγχεται η ταχύτητα δημιουργίας νέων block
- Φυσιολογικά, όσο πιο μεγάλη η υπολογιστική προσπάθεια κάποιου κόμβου, τόσο περισσότερες οι πιθανότητες να είναι ο πρώτος που θα προτείνει ένα block
- Εάν πάλι γίνεται με τυχαίο τρόπο η επιλογή, ίσως δεν θα έχουν κίνητρο οι «ισχυροί»

Είναι σημαντικό να γνωρίζουμε ότι αυτός ο μηχανισμός εφαρμόζεται στο BTC και έχει αντέξει σε κυβερνοεπιθέσεις ~10 χρόνια

Proof of Stake (PoS)

- Κάθε ενδιαφερόμενος κλειδώνει ορισμένα κεφάλαια (stake) και επιλέγονται τυχαία για να κλείσουν ένα νέο μπλοκ (και να πάρουν την ανταμοιβή).
- Με αυτόν τον τρόπο ξοδεύεται λίγη ηλεκτρική ενέργεια και η εξόρυξη δεν χρειάζεται πολύ υπολογιστή ισχύ.
- Όποιος τοποθετεί περισσότερα κεφάλαια έχει μεγαλύτερη πιθανότητα επιλογής
- Είναι ο πιο γνωστός μηχανισμός συναίνεσης μετά το PoW είναι σε διαδικασία ενσωμάτωσης και στο Ethereum 2.0

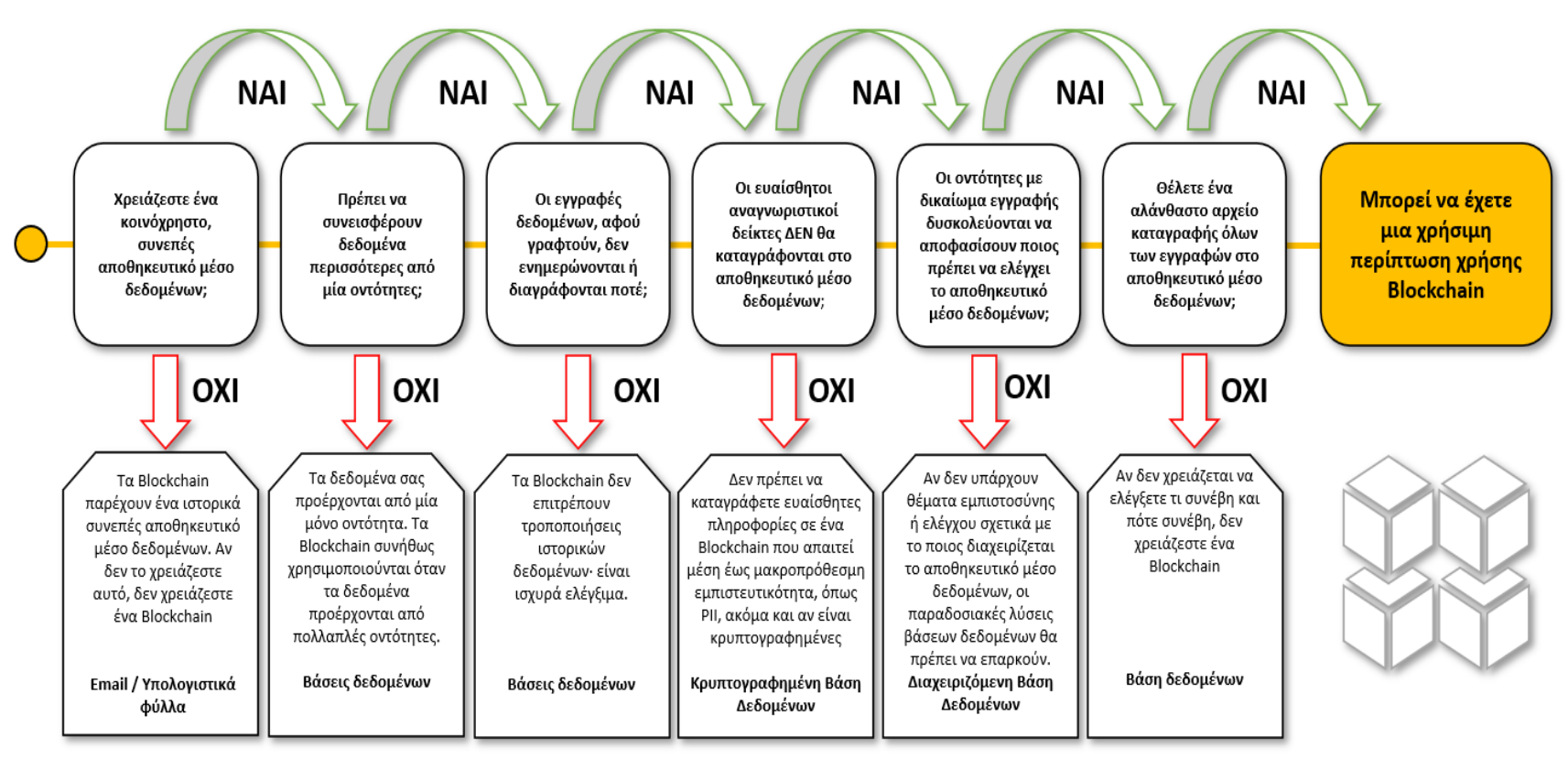
Proof of Authority (PoA)

- στα δίκτυα που βασίζονται σε PoA, οι συναλλαγές και τα μπλοκ επικυρώνονται από εγκεκριμένους λογαριασμούς, γνωστούς ως επικυρωτές (Validators)

- Ένας κόμβος επιλέγεται για επικυρωτής (validator) με βάση ένα σύστημα «πόντων» φήμης

Ανάγκη για Blockchain

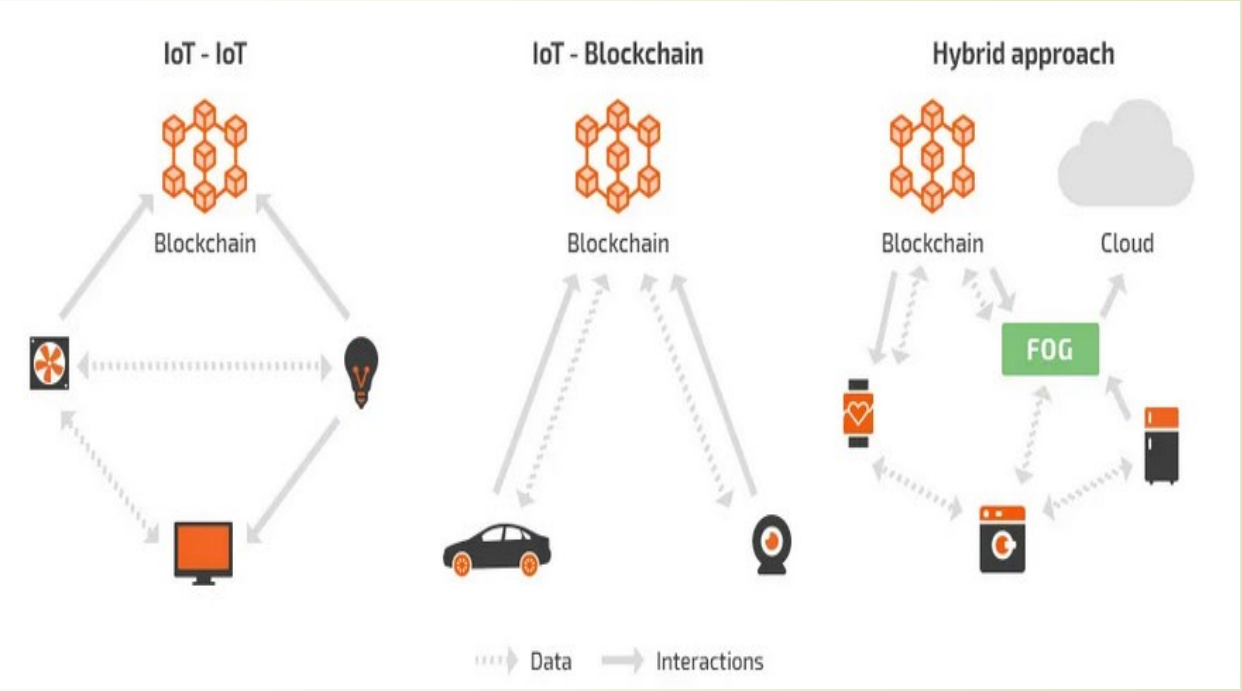
- Είναι το blockchain χρήσιμο για κάθε είδους σύστημα?



Blockchain και IoT

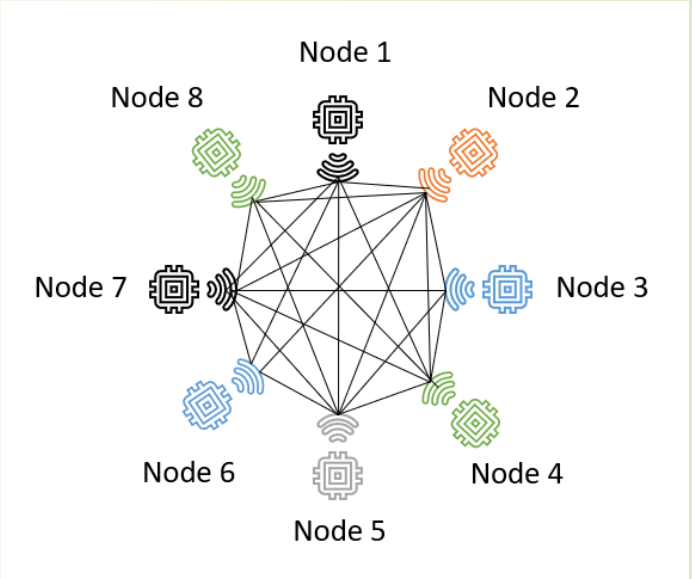
Διάφορα είδη διατάξεων:

- **IoT-to-IoT** (οι συσκευές λειτουργούν ως κόμβοι)
- **IoT-to-BC** (οι συσκευές επικοινωνούν με ένα BC σύστημα)
- **Hybrid** (αξιοποίηση πλεονεκτημάτων από δύο κόσμους BC και Cloud)



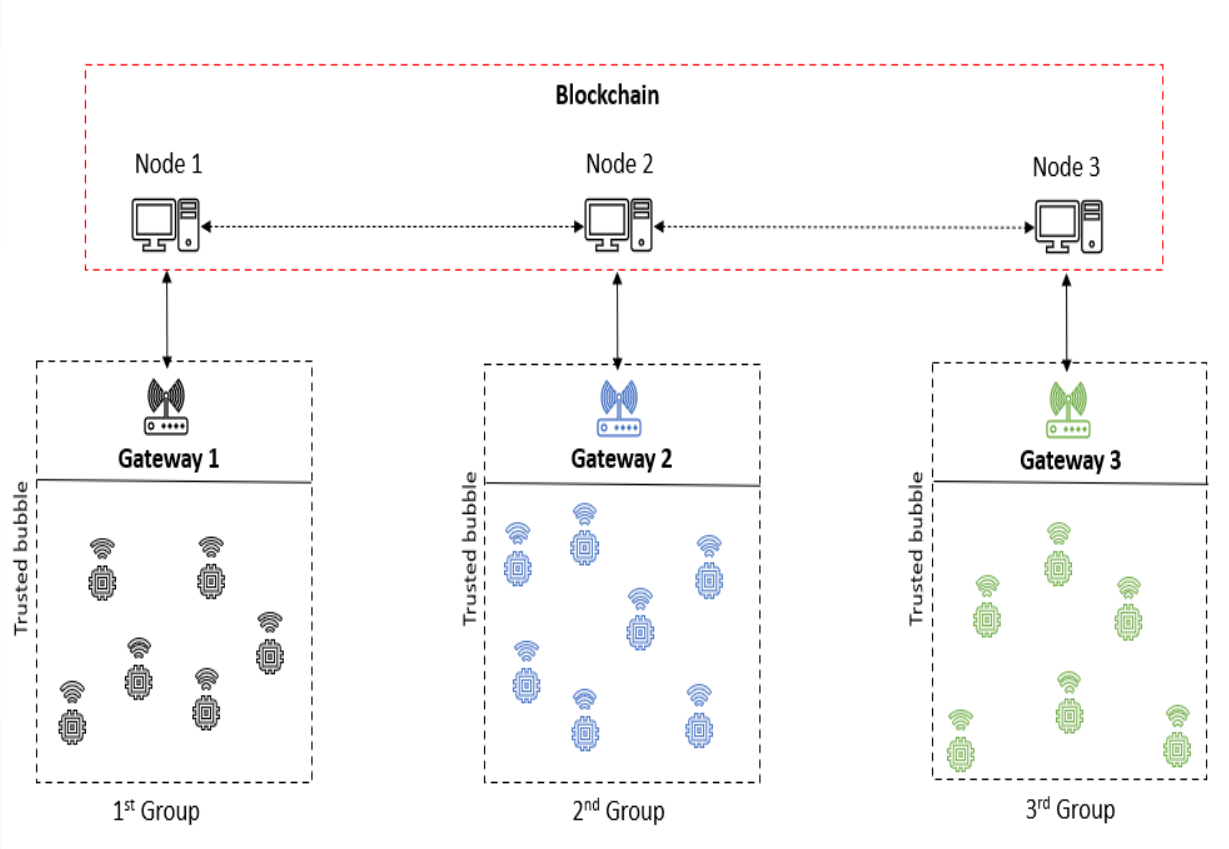
Blockchain και IoT (IoT-to-IoT)

- Οι ίδιες οι συσκευές λειτουργούν ως κόμβοι του blockchain συστήματος
- Απευθείας επικοινωνία μεταξύ των IoT συσκευών μέσω κάποιου πρωτοκόλλου
  (πχ ZigBee, BT)
- Οι συσκευές πρέπει να έχουν ένα ελάχιστο επίπεδο υπολογιστικών δυνατοτήτων (αυτό θέτει και περιορισμούς))
- Συνήθως χρησιμοποιείται σε private blockchains (δλδ με προκαθορισμένο αριθμό
  γνωστών από τα πριν συμμετεχόντων)



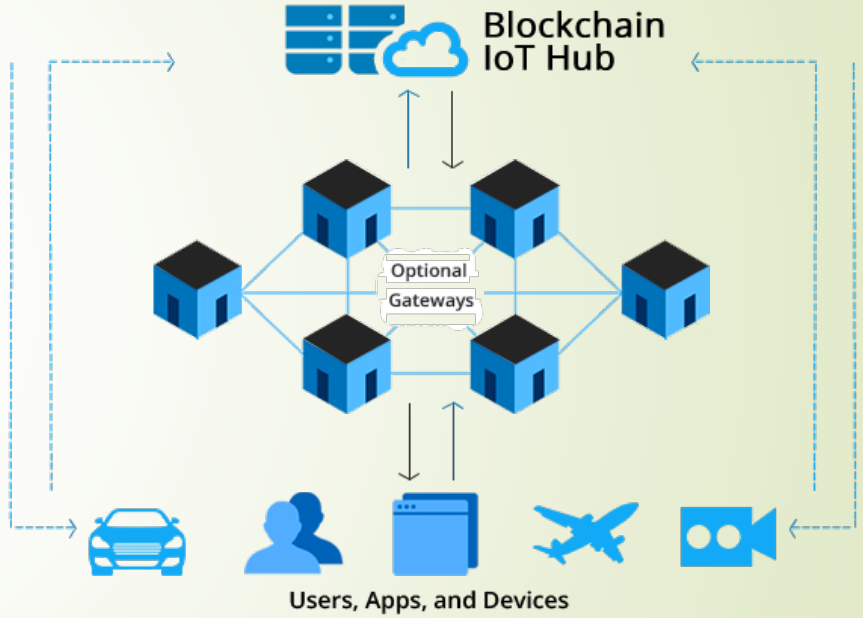
Blockchain και IoT (IoT-to-BC)

- Ένας κεντρικός κόμβος επικοινωνεί με πολλές IoT συσκευές
- Το βασικό πλεονέκτημα είναι ότι δεν υπάρχουν περιορισμοί στο είδος των IoT συσκευών
- Ομαδοποίηση συσκευών



Blockchain και IoT (Hybrid)

- Το σύστημα επικοινωνεί και με μια BC υποδομή και με μια cloud υποδομή
- Αξιοποίηση του BC για ένα μέρος των υπηρεσιών (π.χ. Data integrity check)
- Εύκολα εφαρμόσιμο σε υφιστάμενες λύσεις



Blockchain και IoT (εφαρμογές)

- **Εφοδιαστική Αλυσίδα.** *Σύνθετες και πολύπλοκες διαδικασίες που περιλαμβάνουν αρκετούς συμμετέχοντες. Ένα BC-IoT σύστημα μπορεί να κρατά αποθηκευμένα στοιχεία (θερμοκρασία, θέση, ώρα άφιξης κλπ). Το BC εξασφαλίζει ότι όλοι εμπιστεύονται τα δεδομένα.*

- **Ιατρικό Οικοσύστημα** (νοσοκομεία, κατασκευαστές IoMT, ασφαλιστικές εταιρείες). *Η κάθε ομάδα μπορεί να εφαρμόζει τις δικές της πολιτικές ασφάλειας και παρ/λα να εξασφαλίζει το αμετάβλητο των συναλλαγών.*

- **Ιχνηλασιμότητα προϊόντων.** *Με το BC μπορούμε να «χτίσουμε» μια αλυσίδα προέλευσης του προϊόντος. Όλοι οι χρήστες μπορούν να δουν αυτά τα δεδομένα, όπως και οι ρυθμιστικές αρχές.*

- **Συντήρηση υλικού και λογισμικού συσκευών.** *Οι συσκευές IoT παρακολουθούν την κατάσταση κρίσιμων συσκευών (π.χ. ανελκυστήρας, μηχανές) και αποθηκεύουν δεδομένα κατάστασης στο BC που είναι προσβάσιμα στις αντίστοιχες εταιρείες.*

Ασφάλεια σε IoT (Συστήματα Διαχείρισης Εμπιστοσύνης)

- Όλα τα συστήματα διαχείρισης εμπιστοσύνης χρησιμοποιούν έναν έμπιστο τρίτο (trusted third party) ο οποίος αναλαμβάνει την εξακρίβωση των στοιχείων εκείνου που θέλει να αυθεντικοποιηθεί.
- Γίνεται συνήθως με χρήση πιστοποιητικών (πχ X.509) που είναι ψηφιακά υπογεγραμμένα από κάποια αρχή πιστοποίησης.
- Με την χρήση του BC μπορεί να υλοποιηθεί με μεγαλύτερη ασφάλεια ένα σύστημα διαχείρισης εμπιστοσύνης ή ελέγχου ταυτότητας χωρίς την χρήση ενδιάμεσου μειώνοντας το επιχειρησιακό κόστος και αυξάνοντας την ασφάλεια

Ασφάλεια σε IoT (Συστήματα Διαχείρισης Εμπιστοσύνης)

- Κάθε IoT μπορεί να λειτουργεί σαν κόμβος του BC το οποίο με την εγγραφή στο σύστημα αποκτά ένα μοναδικό blockchain_id
- Το blockchain_id λειτουργεί σαν αναγνωριστικό για κάθε συσκευή μέσα στο σύστημα
- Για να συνδεθεί μια IoT συσκευή με μια άλλη μπορεί να χρησιμοποιήσει το id της και το τοπικό blockchain wallet (που λειτουργεί σαν μέσω αποθήκευσης της υπογραφής του) για να στείλει ένα αίτημα σύνδεσης υπογράφοντας το ψηφιακά
- Το BC πιστοποιεί την εγκυρότητα

Blockchain in digital forensics (1)

- Το BC μπορεί είτε να χρησιμοποιηθεί το ίδιο σαν εργαλείο καταγραφής ψηφιακών στοιχείων είτε σαν εργαλείο αποθήκευσης και διαχείρισης ψηφιακών στοιχείων

- Παράδειγμα 1: Μια υποδομή που χρησιμοποιεί το BC για την καταγραφή των αιτημάτων πρόσβασης των χρηστών

    - ❑ Είναι σημαντικό το με ποιο τρόπο γίνεται η διαχείριση των ταυτοτήτων ώστε να μπορέσουν να ανασυρθούν στοιχεία (π.χ. Bitcoin - anonymity)
    - ❑ Εξίσου σημαντικό είναι το ποιος μπορεί να έχει πρόσβαση σε αυτά τα δεδομένα

Blockchain in digital forensics (2)

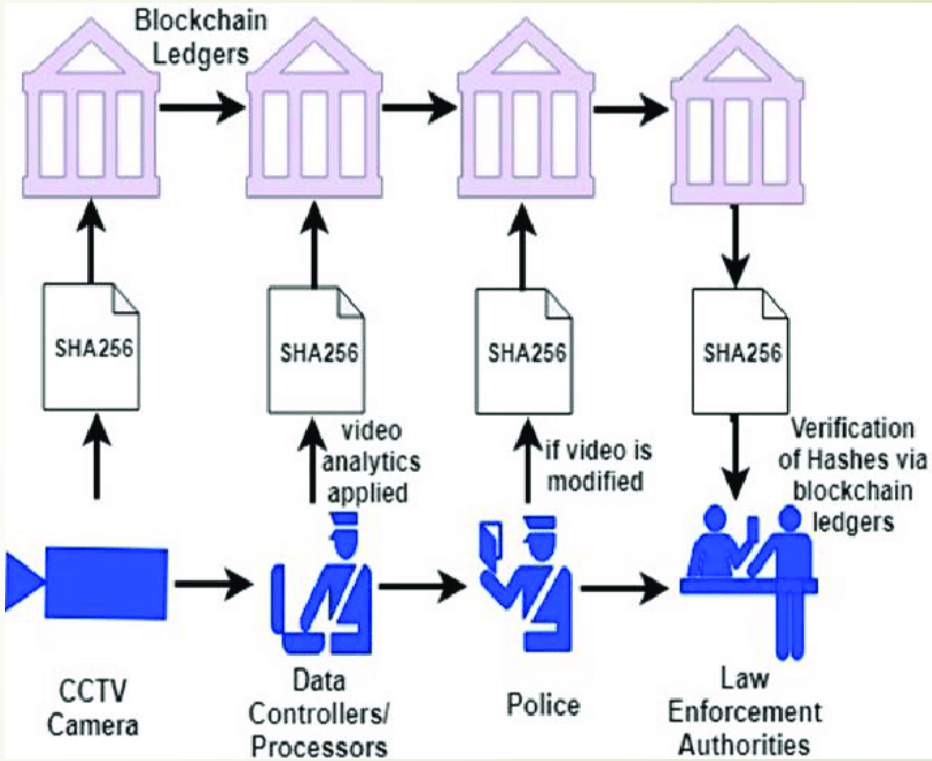
- Παράδειγμα 2: Μια υποδομή που χρησιμοποιεί το BC για την αποθήκευση δεδομένων από sensors
  - ❑ Έλεγχος αν οι αισθητήρες έστελναν δεδομένα στους προκαθορισμένους χρόνους
  - ❑ Έλεγχος αν τα δεδομένα είναι ψηφιακά υπογεγραμμένα από αυθεντικοποιημένους αισθητήρες

Blockchain in digital forensics (3)

- Στην περίπτωση που χρησιμοποιούμε το BC για την αποθήκευση ψηφιακών πειστηρίων έχουμε 2 πλεονεκτήματα.

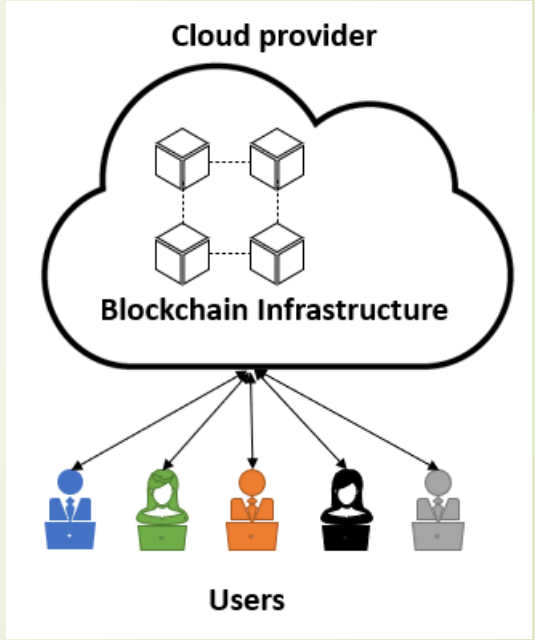
1ον εξασφαλίζουμε ότι αυτά δεν αλλοιώνονται μετά την συλλογή τους

2ον μπορούμε να ελέγξουμε ποιος απέκτησε πρόσβαση σε αυτά



Το Blockchain σαν υπηρεσία (BaaS)-1

- Το Blockchain-as-a-Service είναι ένα προϊόν που προσφέρεται από όλους τους γνωστούς cloud providers (Amazon, Microsoft κλπ)
- Στην πραγματικότητα ο πάροχος αναλαμβάνει την υποδομή και ανάπτυξη των κόμβων του συστήματος σε μια διάταξη εικονικών κόμβων.
- Ο χρήστης μπορεί να αλληλεπιδρά με το BC όπως έκανε με το cloud. Χρησιμοποιώντας κάποιο API για να «μιλά» με το BC
- Κάτι αντίστοιχο με την φιλοξενία μια ιστοσελίδας σε έναν web-hosting provider



Το Blockchain σαν υπηρεσία (BaaS)-2

- Βασικό πλεονέκτημα είναι το κόστος
- Το κόστος για την ιδιοκτησία ενός blockchain δικτύου θεωρείται σχετικά υψηλό (υποδομή, προσωπικό, λογισμικό, hardware, κλπ) όπως και για την λειτουργία του
- Με την χρήση του BaaS μπορεί κανείς να αξιοποιήσει την υπηρεσία pay-as-you-go, πληρώνοντας μόνο για ότι και όσο χρησιμοποίησε
- Η αναβάθμιση της υποδομής γίνεται με κλικ μεταβαίνοντας σε μεγαλύτερο πρόγραμμα

Ψηφιακά πορτοφόλια (digital wallets)

- Επιτρέπουν την αποθήκευση, διαχείριση και ανταλλαγή κρυπτονομισμάτων και άλλων ψηφιακών αγαθών
- Υπάρχουν δύο βασικά είδη: Hot και Cold Wallets
- Τα **Hot Wallets** είναι εφαρμογές είτε διαδικτυακές (π.χ. Metamask) είτε τοπικές (π.χ. Electrum)
- Τα **Cold Wallets** είναι μορφές κρυπτογραφημένων αποσπώμενων δίσκων (π.χ. ένα usb stick ή ένα token key)

Ψηφιακά πορτοφόλια (digital wallets)

- Ανεξάρτητα από το είδος η λειτουργία είναι ίδια
- Όταν δημιουργούμε ένα λογαριασμό αυτόματα δημιουργείται ένα ζεύγος δημόσιου-ιδιωτικού κλειδιού που συνδέεται με αυτόν.
- Το δημόσιο κλειδί είναι γνωστό σε όλους ενώ το ιδιωτικό το γνωρίζουμε μόνο εμείς
- Όποιος θέλει να στείλει κάτι σε εμάς χρησιμοποιεί το δημόσιο κλειδί που είναι στην ουσία η διεύθυνση του λογαριασμού

Ψηφιακά πορτοφόλια (digital wallets)

- Αν θέλουμε να στείλουμε κάτι από τον λογαριασμό θα πρέπει να χρησιμοποιήσουμε το ιδιωτικό κλειδί
- Κάτι αντίστοιχο με τον λογαριασμό email
- Για να μας στείλουν κάτι αρκεί να γνωρίζουν την διεύθυνση email
- Ενώ για να στείλουμε θα πρέπει να γνωρίζουμε τα συνθηματικά

Metamask

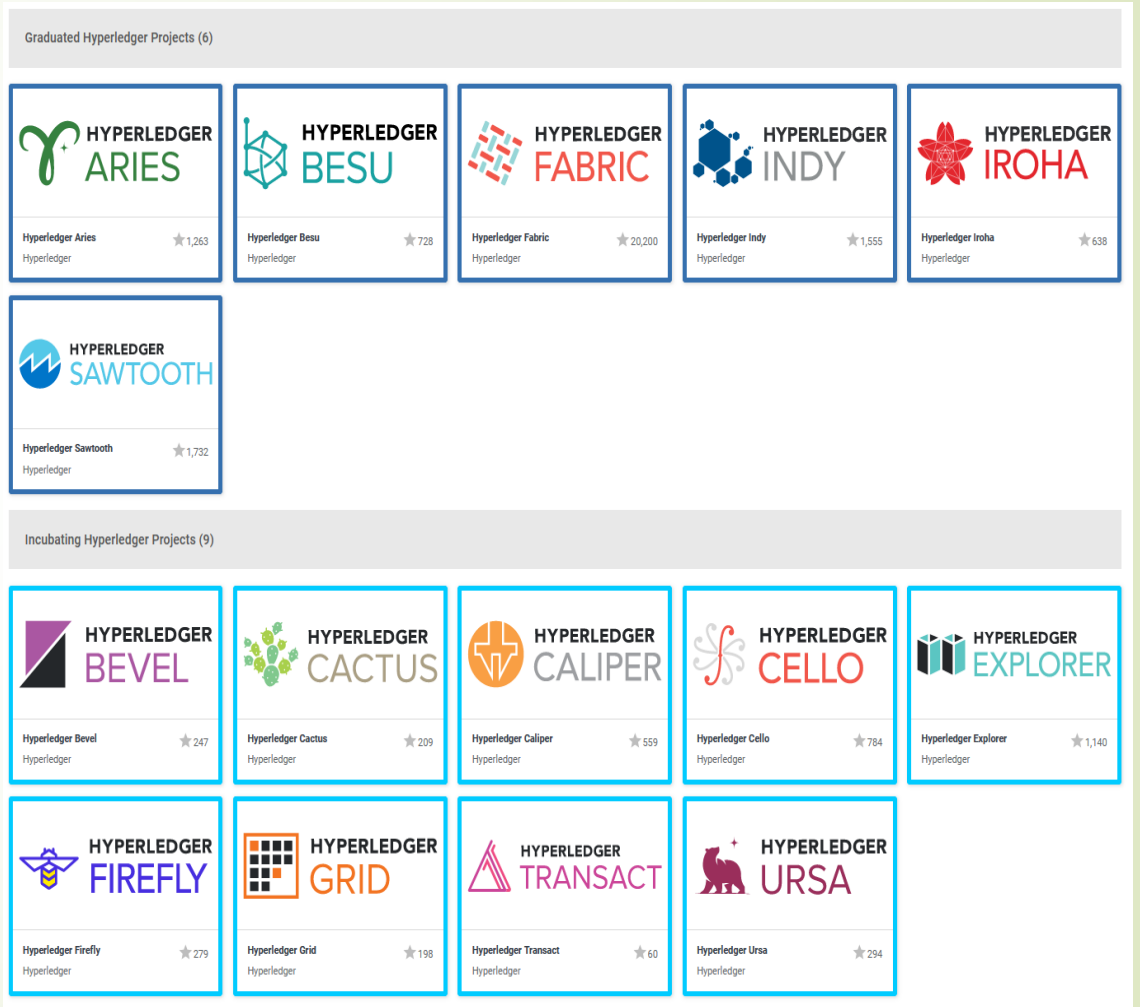
- Ηλεκτρονικό πορτοφόλι web-based για την αποθήκευση κρυπτονομισμάτων
- Δημιουργεί ένα Public-Secret Key pair (όπου το Public είναι η διεύθυνση του λογαριασμού)
- Υπάρχει σαν plugin στο Chrome
- Δυνατότητα σύνδεσης με το Ethereum network για πραγματοποίηση συναλλαγών
- Δυνατότητα δημιουργία test λογαριασμών

Οι πλατφόρμες Hyperledger Fabric και Ethereum

- Δύο από τις πιο γνωστές πλατφόρμες blockchain με μεγάλες κοινότητες και πολλές βιβλιοθήκες
- Συνήθως το Ethereum χρησιμοποιείται για public και permissionless εφαρμογές (π.χ. crowdfunding)
- Συνήθως το Hyperledger χρησιμοποιείται για private και permissioned εφαρμογές (π.χ. Corporate dApps)

Hyperledger Fabric

- Υποστηρίζεται από το Linux Foundation
- Έχει ευέλικτη αρχιτεκτονική που μπορεί να συνδυάζει διαφορετικά υποσυστήματα
- Χρησιμοποιεί ένα μοντέλο ανοιχτών «Έξυπνων Συμβολαίων» που μπορούν να υποστηρίζουν διαφορετικούς τύπους δεδομένων
- Χρησιμοποιεί τον Raft consensus



Ethereum - 1

- Δημιουργήθηκε το 2013 και υποστηρίζεται από την Intel, Microsoft, JPMorgan κλπ
- Το μέγεθος του είναι συγκρίσιμο με το bitcoin network
- Έχει το δικό του κρυπτονόμισμα – Ether
- Μας δίνει την δυνατότητα να αναπτύξουμε και να δημοσιεύσουμε αποκεντρωμένες εφαρμογές (dApps) – εφαρμογές που «τρέχουν» απευθείας στο BC
- Έχουμε την δυνατότητα να επιλέξουμε ανάμεσα σε private και public υλοποιήσεις

Ethereum - 2

- Έχει πληθώρα εργαλείων που μας δίνουν την δυνατότητα να αναπτύξουμε «Έξυπνα Συμβόλαια» (π.χ. Remix, Truffle, Visual Studio)
- Για την ανάπτυξη «Έξυπνων Συμβολαίων» χρησιμοποιεί την γλώσσα Solidity (μοιάζει με javascript)
- Χρησιμοποιεί το EVM (Ethereum Virtual Machine)
- Χρησιμοποιεί PoW ενώ ετοιμάζεται η μετάβαση στο PoS (Ethereum 2.0)
- Τα βασικά μειονεκτήματα ο χρόνος ολοκλήρωσης των συναλλαγών και αυξημένα κόστη σε σχέση με άλλες πλατφόρμες

Ethereum Virtual Machine

- Είναι μια sandboxed εικονική μηχανή (vm) που τρέχει σε κάθε κόμβο
- Ας το σκεφτούμε σαν μεταγλωτιστή
- Οι κόμβοι φτάνουν σε συναίνεση εκτελώντας όλα τα transactions
- Μόνο ο miner ανταμοίβεται ενώ οι υπόλοιποι κόμβοι απλά επιβεβαιώνουν

Ethereum Denominations

1 Ether = 1000000000000000000 Wei (10^18)
1 Ether = 1000000000000000 Kwei, Ada, Femtoether (10^15)
1 Ether = 1000000000000 Mwei, Babbage, Picoether (10^12)
1 Ether = 1000000000 Gwei, Shannon, Nanoether, Nano (10^9)
1 Ether = 1000000 Szabo, Microether, Micro (10^6)
1 Ether = 1000 Finney, Milliether, Milli (10^3)
1 Ether = 0.001 Kether, Grand, Einstein (10^-3)
1 Ether = 0.000001 Mether (10^-6)
1 Ether = 0.000000001 Gether (10^-9)
1 Ether = 0.000000000001 Tether (10^-12)

Addresses and Keys

- Μια διεύθυνση στο Ethereum είναι «δεμένη» με κάποιο ιδιωτικό κλειδί
- Το «ιδιωτικό» κλειδί είναι μήκους 64 χαρακτήρων (32bytes)
- Το δημόσιο κλειδί (PK)που προκύπτει είναι η διεύθυνση
- Έχουμε δύο είδη λογαριασμών:

  1. Externally Owned Account-EOA). Λογαριασμός που ελέγχεται με κάποιο ιδιωτικό κλειδί (SK)
  2. Η διευθύνσεις των Smart Contracts δεν ελέγχονται από κάποιο SK

https://andersbrownworth.com/blockchain/public-private-keys/

Συναλλαγές - Transactions

- Κάθε συναλλαγή έχει υποχρεωτικά τα πεδία:

  - nonce: τυχαίος αριθμός
  - gasprice: τιμή/αξία συναλλαγής
  - startgas: ανώτατο όριο συναλλαγής
  - to: διεύθυνση παραλήπτη(EoA or contract address)
  - value: Λεφτά που μεταφέρονται (σε Ether)
  - data: δεδομένα που μεταφέρονται
  - v, r, s: ECDA υπογραφή

Τα διάφορα δίκτυα του Ethereum

0: Olympic, Ethereum public pre-release testnet
1: Frontier, Homestead, Metropolis, the Ethereum public main network
2: Morden, the public Ethereum testnet, now Ethereum Classic testnet
3: Ropsten, the public cross-client Ethereum testnet
4: Rinkeby, the public Geth PoA testnet
8: Ubiq, the public Gubiq main network with flux difficulty chain ID 8
42: Kovan, the public Parity PoA testnet
77: Sokol, the public POA Network testnet
99: Core, the public POA Network main network
7762959: Musicoin, the music blockchain

Souce: https://ethereum.stackexchange.com/questions/17051/how-to-select-a-network-id-or-is-there-a-list-of-network-ids/17101#17101

Προγραμματισμός σε Ethereum

Γλώσσες υψηλού επιπέδου:

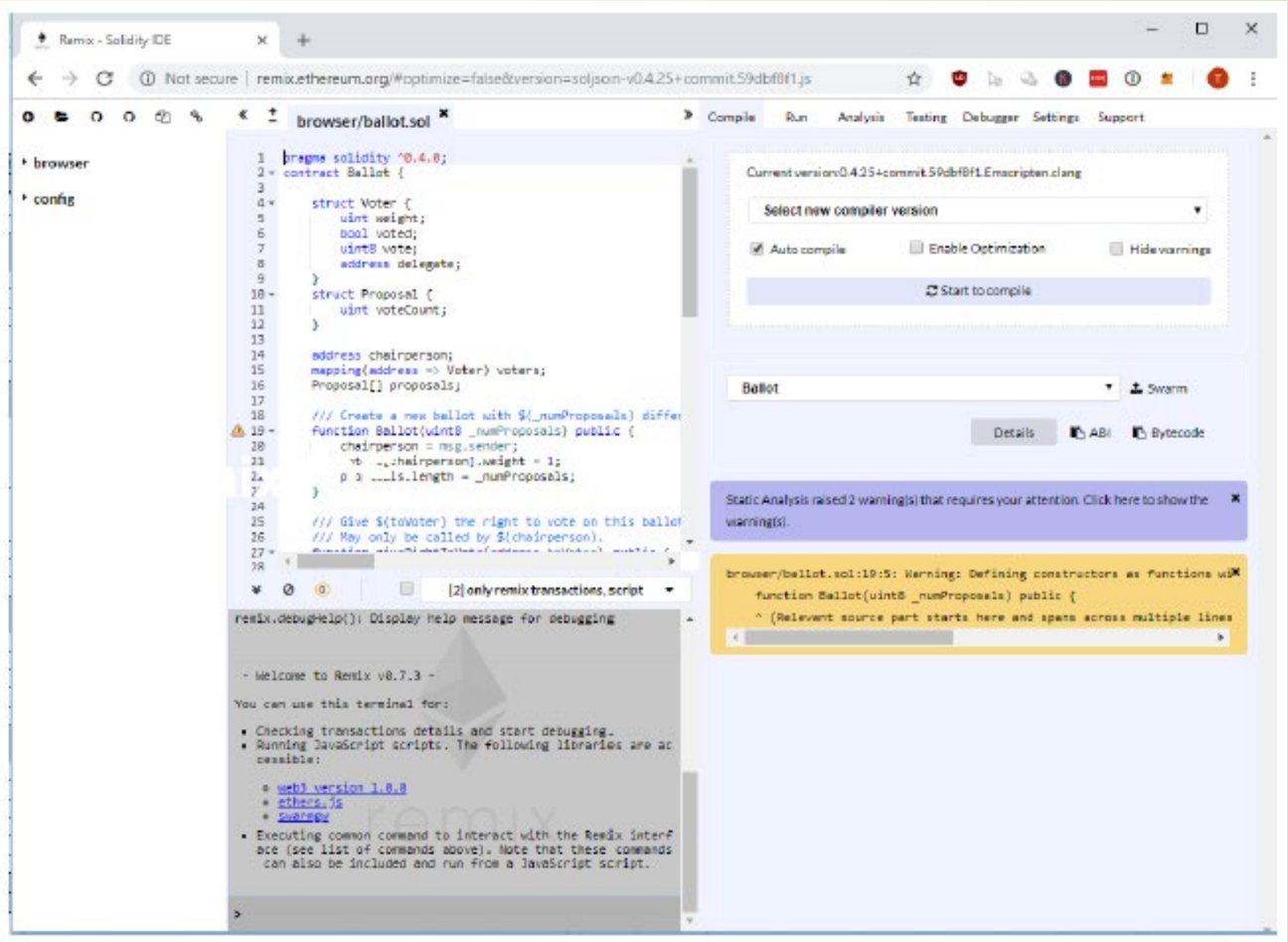
Solidity – Η πιο δημοφιλής (μοιάζει με javascript)
Vyper – Έμφαση στην ασφάλεια
LLL – "Low Level Lisp-like Language"
Mutan – Golang-like
Serpend, Python-like

Φυσικά μπορούν να χρησιμοποιηθούν και Java, C++ κλπ

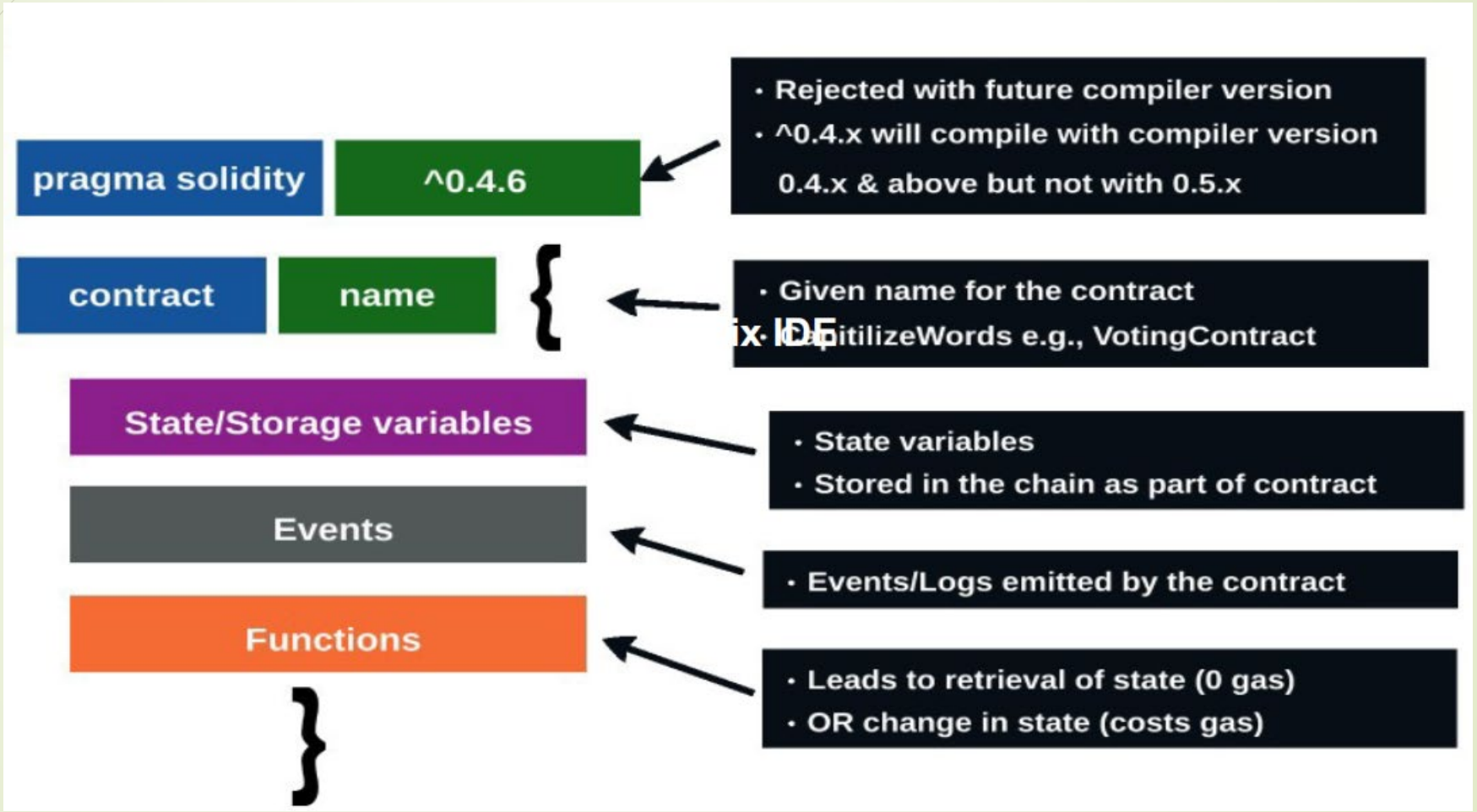
Remix IDE

Το Remix IDE είναι ένα web Framework που μας δίνει την δυνατότητα να αναπτύξουμε και να τρέξουμε «Έξυπνα Συμβόλαια».

Μπορούμε μέσα από τον debugger να δούμε όλες τις ενέργειες που συμβαίνουν στο blockchain



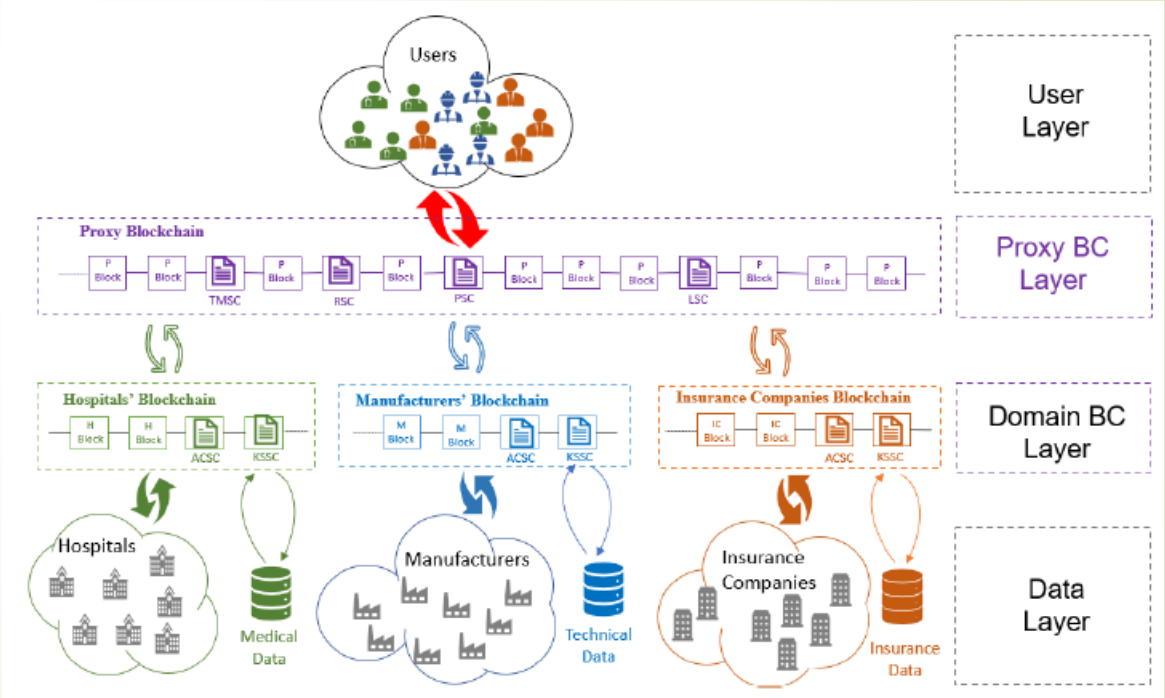
Η Δομή ενός Smart Contract



Ερευνητικό παράδειγμα

"A hierarchical multi blockchain for fine grained access to medical data"
Malamas, V., Kotzanikolaou, P., Dasaklis, T. K., & Burmester, M. (2020). *IEEE Access*

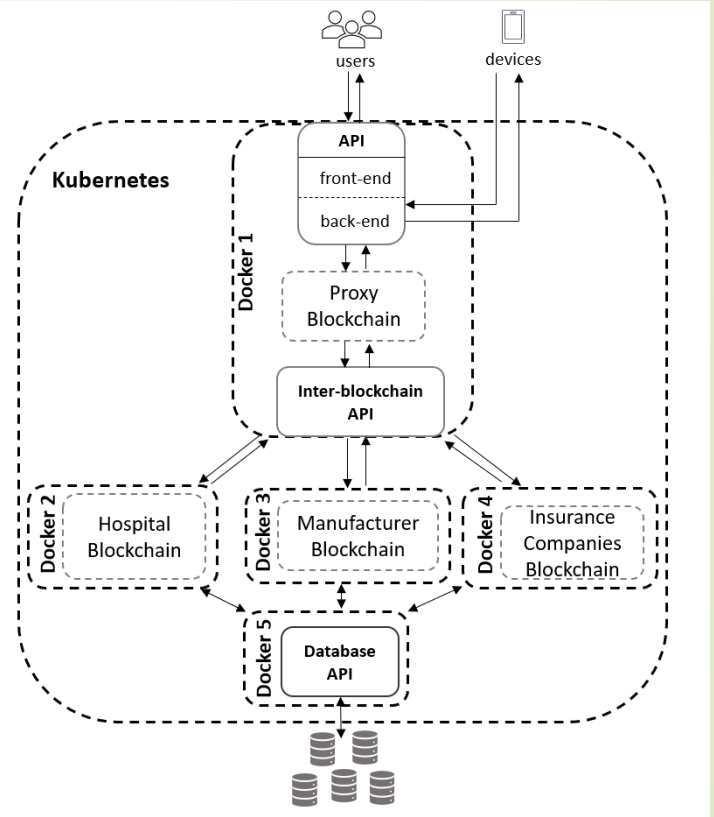
- Το σενάριο βασίζεται στη δομή του ιατρικού οικοσυστήματος (πολλαπλές συσκευές, πολλαπλοί χρήστες και πολλοί διαφορετικοί ιδιοκτήτες δεδομένων)
- Ποιο πρόβλημα θέλουμε να λύσουμε?



Ερευνητικό παράδειγμα

"A hierarchical multi blockchain for fine grained access to medical data"
Malamas, V., Kotzanikolaou, P., Dasaklis, T. K., & Burmester, M. (2020). *IEEE Access*

- Δομή της αρχιτεκτονικής
- Τεχνική υλοποίηση



Ερευνητικό παράδειγμα

"A hierarchical multi blockchain for fine grained access to medical data"
Malamas, V., Kotzanikolaou, P., Dasaklis, T. K., & Burmester, M. (2020). *IEEE Access*

- Έξυπνα συμβόλαια που αναπτύχθηκαν
  - Στο PBC (PSC, TMSC, LSC)
  - Στο DBC (ACSC, KSSC)

- Στο καθένα από αυτά ενσωματώνονται διαφορετικές λειτουργίες που εκτελούνται όταν «κάποιος» ή «κάτι» τις ενεργοποιήσει (χρήστης ή άλλο SC)

Ερευνητικό παράδειγμα

"A hierarchical multi blockchain for fine grained access to medical data"
Malamas, V., Kotzanikolaou, P., Dasaklis, T. K., & Burmester, M. (2020). *IEEE Access*

TABLE 2. Proxy Blockchain SCs main functions.

Function	Smart Contract	Actuator	Input	Output	Description
constructor	All	-	-	True/ False	Initialize and store information relevant to the SC
majorityConsent	PSC	*addCA* *removeCA* *retrieveLog*	authoritySign	True/ False	Sends and receives signed challenges from the Authorities
updateTrustAnchors	TMSC	CA Admin	new Cert new CRL new TempACL	updated pointer	Updates the Trust Anchors and updates the pointer to the last object of the linked lists. Triggers the *updateLog* from the LSC
getUserValidation	TMSC	PSC	userCert	True/ False, userRole	Checks the validity of the certificate by comparing it with the Ind_{CERT}
validateUser	PSC	User	userCert	True/ False, userRole	Takes as input the given user certificate and triggers the *getUserValidation* from the TMSC
addCA	TMSC	CA Admin	majorityConsent new CA_{CERT}, new CA_{CRL}, new $CA_{TempACL}$	updated pointer	Updates the Trust Anchors and sets a new pointer to the last object of the linked lists. Triggers the *updateLog* from the LSC.
removeCA	TMSC	CA Admin	majorityConsent revocate CA_{CERT}, revocate CA_{CRL}, revocate $CA_{TempACL}$	updated pointer	Updates the Trust Anchors and sets a new pointer to the last object of the linked lists. Triggers the *updateLog* from the LSC.
userRegistration/ deviceRegistration	RSC	User / device	userCert / devCert	userAddress/ devAddress	Links a user/device certificate with a unique Ethereum address (GID). Triggers the *registrationLog* of the LSC.
requestAccess	PSC	User	userCert,data_id	userRole, data_id	First, it triggers the *requestLog* from the LSC to log the request and then calls the *getUserValidation* from the TMSC to validate the user, if the function returns true it forwards the request to the DBC.
requestLog	LSC	PSC	request details	True/ False	Triggered by the PSC when a request is made. Returns a logging verification variable
updateLog	LSC	TMSC	update details	True/ False	Triggered by the TMSC when a request is made. Returns a logging verification variable
registrationLog	LSC	RSC	registration details	True/ False	Triggered by the RSC when a request is made. Returns a logging verification variable
retrieveLog	LSC	Auditor	majorityConsent, retrieve details	Logs	When an Auditor request access to the Logs, Authorities grant access through majority voting. It is a threshold enabled function.

Ερευνητικό παράδειγμα

"A hierarchical multi blockchain for fine grained access to medical data"
Malamas, V., Kotzanikolaou, P., Dasaklis, T. K., & Burmester, M. (2020). *IEEE Access*

TABLE 3. Domain Blockchain SCs main functions.

Function	Smart Contract	Actuator	Input	Output	Description
constructor	All	-	-	True/ False	Initialize and store information relevant to the SC
policyEnf	ACSC	PSC	verified attributes, data_id	True/False	Enforces the predefined policy according to the given verified attributes
requestData	KSSC	ACSC	data_id	partially decrypted data	It is activated when temporal attributes are verified for the user. The function retrieves the data from the database and performs partial decryption with the Temporal keys before it sends them to the PSC and then to user.
accessLog	ACSC	*policyEnf*	policy enforcement details	Logs	Triggered by the ACSC when policy enforcement over a request is made. Logs the transaction details on the Domain BC.
keystoreLog	KSSC	*requestData*	partial decryption details	Logs	Triggered by the KSSC when a partial decryption request is made. Logs the partial decryption details on the Domain BC.