

|  |
| --- |
| **Ασφαλής Διαχείριση Ιχνηλάτησης Εφοδιαστικής Αλυσίδας με χρήση Κατανεμημένων και Ενσωματωμένων Μηχανισμών Ασφάλειας  (Τ2ΕΔΚ-02836)** |
| **ΑΡΤΕΜΙΣ** |
| ***Αρχιτεκτονική κατανεμημένων υπηρεσιών ασφάλειας*** |

|  |  |
| --- | --- |
| ΕΝΟΤΗΤΑ ΕΡΓΑΣΙΑΣ ΕΡΓΟΥ: | **ΕΕ3** |
| ΚΩΔΙΚΟΣ ΠΑΡΑΔΟΤΕΟΥ: | **Π3.2** |
| ΕΚΔΟΣΗ: | 1.0 |
| ΚΑΤΑΣΤΑΣΗ: | **Επεξεργασία** |
| ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ: | **30.01.2023** |
| ΥΠΕΥΘΥΝΟΣ ΦΟΡΕΑΣ: | **ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ** |
| ΣΥΜΜΕΤΕΧΟΝΤΕΣ ΦΟΡΕΙΣ | **ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  **ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**  **iTrack Services ΕΠΕ** |

Πίνακας Περιεχόμενων

[Περίληψη 6](#_heading=h.1jlao46)

[1. Εισαγωγή 7](#_heading=h.40ew0vw)

[1.1. Σκοπός και στόχοι του παραδοτέου 7](#_heading=h.2fk6b3p)

[1.2. Δομή του παραδοτέου 8](#_heading=h.upglbi)

[**2. Αρχιτεκτονική συστήματος 9**](#_heading=h.7b0xdo9m67wb)

[2.1. Ανάλυση οντοτήτων συστήματος 9](#_heading=h.q880bs8gs41y)

[TMSC (trust management smart contract) 13](#_heading=h.t7sv771z9ae9)

[PSC (Proxy Smart Contract) 14](#_heading=h.q73hyc1j6a2f)

[LSC (Logging Smart Contract) 14](#_heading=h.bfe4m2nhq5d)

[ACSC (Access Control smart Contract) 15](#_heading=h.7lps0gffwib9)

[KSSC (Key Store Smart Contract) 15](#_heading=h.ke6ll3riebz1)

[DHSC (Data Handler Smart Contract) 16](#_heading=h.wvvmm2sordy)

[2.2. Αρχιτεκτονική συστήματος 19](#_heading=h.mypjdd5zyhvy)

[Διαγράμματα Ροής Δεδομένων (Data Flow Diagrams) 20](#_heading=h.243i4a2)

[Γενικό (context) Διάγραμμα – Επίπεδο 0 20](#_heading=h.j8sehv)

[Διάγραμμα – Επίπεδο 1 21](#_heading=h.338fx5o)

[3. Κατανεμημένη υποδομή και υπηρεσίες 21](#_heading=h.3ep43zb)

[3.1. Εισαγωγή 21](#_heading=h.1tuee74)

[3.2. Τεχνολογίες συστήματος 22](#_heading=h.4du1wux)

[3.3. Αρχιτεκτονική συστήματος 27](#_heading=h.3oy7u29)

[3.4. Σενάρια χρήσης 27](#_heading=h.1idq7dh)

[4. Κατανεμημένη βάση δεδομένων 30](#_heading=h.42ddq1a)

[5. Ανάλυση μηχανισμών ασφάλειας 32](#_heading=h.2hio093)

[5.1. Εισαγωγή 32](#_heading=h.wnyagw)

[6. Συμπεράσματα 33](#_heading=h.haapch)

[7. Αναφορές 34](#_heading=h.3gnlt4p)

[Οπισθόφυλλο 35](#_heading=h.1vsw3ci)

Πίνακας Πινάκων

Πίνακας Εικόνων

Τεχνικοί Όροι & Συντομογραφίες

# Περίληψη

Το παρόν παραδοτέο (Π3.2) της ενότητας εργασίας ΕΕ3 με τίτλο “Αρχιτεκτονική κατανεμημένων υπηρεσιών ασφάλειας” έχει σκοπό να μελετήσει εναλλακτικές τεχνολογίες και να προτείνει μια αρχιτεκτονική βασισμένη σε τεχνολογία κατανεμημένου καθολικού (blockchain) για την ανάπτυξη της πρωτότυπης πλατφόρμας, η οποία θα αποτυπώνει ρεαλιστικά τις επιχειρησιακές διαδικασίες ιχνηλάτησης στην ΕΑ. Παρουσιάζονται οι επιμέρους τεχνολογίες, που συνδυάζονται στα πλαίσια της αρχιτεκτονικής του υπό ανάπτυξη συστήματος, και στην συνέχεια αναλύεται η αρχιτεκτονική αυτή, καθώς και το πως συνεισφέρει στην επίτευξη των βασικών στόχων του σχεδιασμού μιας ασφαλούς κατανεμημένης πλατφορμας για την υποστήριξη/παρακολούθηση συστημάτων εφοδιαστικής αλυσίδας.

# Εισαγωγή

## Σκοπός και στόχοι του παραδοτέου

Στα οικοσυστήματα της αλυσίδας εφοδιασμού, βασιζόμαστε όλο και περισσότερο στα πληροφοριακά συστήματα για τις βασικές διαδικασίες των οργανισμών. Είναι σύνηθες αυτά τα πληροφοριακά συστήματα να διαχειρίζονται ευαίσθητες πληροφορίες ή να υλοποιούν ευαίσθητες λειτουργίες ή ενέργειες. Για παράδειγμα, η αλυσίδα εφοδιασμού μπορεί να βασίζεται σε πληροφοριακά συστήματα για τη διαχείριση ευαίσθητων πληροφοριών, όπως τα δεδομένα πελατών, ή για την υλοποίηση ευαίσθητων λειτουργιών, όπως η διαχείριση των επιπέδων αποθεμάτων.

Επιπλέον, στα πληροφοριακά συστήματα που συναντώνται στον τομέα της εφοδιαστικής αλυσίδας, υπάρχουν πολλοί ενδιαφερόμενοι που συμμετέχουν και επικοινωνούν μεταξύ τους. Από τον κατασκευαστή ενός προϊόντος και τον προμηθευτή που παρέχει τις πρώτες ύλες, μέχρι τον διανομέα και τον πωλητή λιανικής.

Η από κοινού διαχείριση τέτοιων συστημάτων από τα εμπλεκόμενα μέρη που τα απαρτίζουν είναι μια δύσκολη, επίπονη και χρονοβόρα διαδικασία, καθώς τα εμπλεκόμενα μέρη ανήκουν σε διαφορετικούς οργανισμούς, με διαφορετικές πολιτικές και απαιτήσεις. Θα πρέπει επίσης να σημειωθεί ότι η ανταλλαγή κρίσιμων πληροφοριών (όπως, για παράδειγμα, δεδομενα απο τις συσκευές μέτρησης υγρασίας και θερμοκρασίας κατα την μεταφορά και αποθήκευση των προϊόντων) δεν είναι απλή υπόθεση, καθώς, με τα τρέχοντα εργαλεία που διαθέτουν οι οργανισμοί, η ανταλλαγή αυτή απαιτεί κυρίως την κρυπτογράφηση των δεδομένων σε συγκεκριμένους παραλήπτες.

Αυτή η απαίτηση για κρυπτογράφηση των πληροφοριών είναι συνέπεια της ανάγκης διαβαθμισμένης πρόσβασης των δεδομένων απο τα εμπλεκόμενα μέρη. Όπως και με άλλες κατηγορίες πληροφοριών και δεδομένων, τα δεδομένα και οι πληροφορίες πρέπει να προστατεύονται από την υποκλοπή και τη δυνατότητα ανάγνωσης από τρίτους που θα μπορούσαν να επωφεληθούν από την κατοχή ή τη γνώση αυτών των πληροφοριών. Επιπλέον, ένα θεμελιώδες χαρακτηριστικό που ενδιαφέρει είναι η δυνατότητα εξασφάλισης της ακεραιότητας των δεδομένων ώστε αυτά να μην μπορούν να αλλοιωθούν ώστε να προσφέρουν μια πιστή αποτύπωση της τρέχουσας κατάστασης. Τα δεδομένα και οι πληροφορίες για τα προϊόντα, πολύ συχνά, χρησιμοποιούνται από οντότητες τρίτων, ώστε να διαπιστωθεί η σωστή εφαρμογή του κανονιστικού πλαισίου και των κανονισμών συμμόρφωσης που αφορούν μια ευρεία γκάμα προϊόντων (π.χ. τρόφιμα, φάρμακα). Ως εκ τούτου, η ανάγκη να διασφαλιστεί ότι τα δεδομένα που λαμβάνουν δεν είναι παραποιημένα (σκόπιμα ή μη) είναι εξαιρετικά σημαντική.

Ιδιαίτερα πρέπει να ληφθεί υπόψη, ότι ο σχεδιασμός μιας τέτοιας αρχιτεκτονικής για την ανταλλαγή δεδομένων καθίσταται εξόχως δυσκολότερη όταν μια σειρά παράμετροι του συστήματος αλλάζουν συχνά (π.χ. συχνές αλλαγές κατοχής των προϊόντων, από τον παραγωγό στον μεταφορέα κλπ).

Το παρόν παραδοτέο παρουσιάζει την λεπτομερή ανάλυση του σχεδιασμού του προτεινόμενου συστήματος που επιλύει τα παραπάνω ζητήματα.

Συγκεκριμένα, δίνει τη δυνατότητα σε πληροφοριακά συστήματα στα οποία εμπλέκονται πολλοί ενδιαφερόμενοι να ανταλλάσσουν ευαίσθητες πληροφορίες, προστατεύοντας την εμπιστευτικότητα, την ακεραιότητα και την αυθεντικότητα των πληροφοριών αυτών, ενώ ταυτόχρονα αξιοποιεί την τεχνολογία Blockchain, ώστε όλοι οι ενδιαφερόμενοι να έχουν συνολική εποπτεία των ενεργειών που έχουν πραγματοποιηθεί, όπως για παράδειγμα αιτήματα πρόσβασης σε συγκεκριμένα δεδομένα κλπ.

Επιτρέπει επίσης την αποστολή αιτημάτων και τη λήψη κρίσιμων πληροφοριών μόνο από οντότητες, χρήστες τους οποίους οι οργανισμοί έχουν δυναμικά ορίσει ότι μπορούν να έχουν πρόσβαση σε αυτές. Ως δυναμική αξίωση πρόσβασης ορίζεται η ικανότητα του συστήματος να επιτρέπει την ανάκτηση πληροφοριών και την πρόσβαση σε αυτές μόνο από οντότητες που έχουν λάβει την κατάλληλη έγκριση από τον ενδιαφερόμενο φορέα στον οποίο ανήκουν μέσω συγκεκριμένων, προκαθορισμένων πολιτικών πρόσβασης. Αυτές οι πολιτικές μπορούν να ανανεώνονται από οποιοδήποτε από τα εμπλεκόμενα μέρη ανά πάσα στιγμή, ανάλογα με τις ανάγκες που προκύπτουν.

Για παράδειγμα, ένας οδηγός φορτηγού που ανήκει σε έναν από τους συμμετέχοντες οργανισμούς του συστήματος (π.χ. ένας οργανισμός που προσφέρει υπηρεσίες μεταφοράς προϊόντων) μπορεί να λάβει τις πληροφορίες για τα προϊόντα που μεταφέρει, καθώς και τα δεδομένα που λαμβάνει από τους σενσορες του οχήματος μόνο εφόσον πληροί τουλάχιστον μία από τις ακόλουθες δύο προϋποθέσεις:

- Εάν ο οδηγός έχει εξουσιοδοτηθεί από τον προμηθευτή ή την εταιρία μεταφοράς ώστε να έχει πρόσβαση σε τέτοιες πληροφορίες.

- Εάν ο διανομέας έχει υπογράψει συμφωνία μη δημοσιοποίησης προμηθευτή ή την εταιρία μεταφοράς.

Οι δύο παραπάνω προϋποθέσεις εκτελούνται αυτόματα και ελέγχονται από το σύστημα μέσω των ατομικών πολιτικών κάθε εμπλεκόμενου μέρους. Σε αυτό το παράδειγμα, χρησιμοποιούνται οι πολιτικές του οργανισμού της αλυσίδας εφοδιασμού στον οποίο ανήκει ο εν λόγω οδηγός που επιθυμεί να έχει πρόσβαση στις πληροφορίες για το προϊόν.

Όπως γίνεται αντιληπτό, ένα εξαιρετικά σημαντικό σημείο της προτεινόμενης υλοποίησης είναι ότι οι πολιτικές αυτές δεν απαιτούν καμία ενέργεια από τα εμπλεκόμενα μέρη, εκτός από την ενημέρωσή τους, αφού αξιοποιούνται αυτόματα από το σύστημα, χωρίς να απαιτείται η παρουσία του ανθρώπινου παράγοντα κατά τη λήψη αποφάσεων και, γενικότερα, κατά την εκτέλεση κρίσιμων ενεργειών. Με τον τρόπο αυτό μειώνεται η πιθανότητα σφαλμάτων και καθυστερήσεων στη διαδικασία της εφοδιαστικής αλυσίδας και διασφαλίζεται ότι οι ευαίσθητες πληροφορίες είναι προσβάσιμες μόνο από εξουσιοδοτημένα μέρη, ενισχύοντας έτσι την ασφάλεια και την αξιοπιστία του συστήματος της εφοδιαστικής αλυσίδας.

## Δομή του παραδοτέου

# Αρχιτεκτονική συστήματος

Στο κεφάλαιο αυτό …

## Ανάλυση οντοτήτων συστήματος

Στο συγκεκριμένο κεφάλαιο θα γίνει παρουσίαση των τεχνολογιών που χρησιμοποιούνται για την υλοποίηση της υφιστάμενης υπηρεσίας ασφάλειας. Στην συνέχεια θα γίνει λεπτομερής ανάλυση του λογισμικού εφαρμογής χρήστη (Client Application), του λογισμικού διασύνδεσης του χρήστη με το σύστημα (Backend API) και του blockchain συστήματος με ιδιαίτερη αναφορά στα smart contracts. Έπειτα θα παρουσιαστεί η ανάλυση του λογισμικού διασύνδεσης blockchain συστημάτων (Inter-Blockchain API), του λογισμικού διασύνδεσης συστήματος – με την αποκεντρωμένη βάση δεδομένων (DBC API) και τέλος οι κρυπτογραφικές πράξεις που απαρτίζουν το σύστημα.

2.1.1 Λογισμικό Εφαρμογής Χρήστη (Client Application)

Η συγκεκριμένη οντότητα αποτελεί *πύλη* στο σύστημα. Κάθε ανθρώπινος ρόλος του συστήματος μπορεί να κάνει χρήση υπηρεσιών που υλοποιούνται και παρέχονται από το σύστημα (Πελάτης-Εξυπηρετητής), μέσω αυτής.

Shape

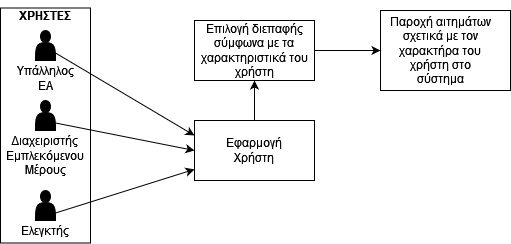
Description automatically generated with medium confidence

**Εικόνα 1: ΔΙΑΣΥΝΔΕΣΗ ΧΡΗΣΤΗ - ΣΥΣΤΗΜΑ**

Είναι, δηλαδή, η διεπαφή (User Interface - UI) ενός ανθρώπινου ρόλου, με το υπόλοιπο σύστημα.

Στην παρούσα υλοποίηση, η συγκεκριμένη οντότητα:

* Αυθεντικοποιεί τους χρήστες του συστήματος με την *Αρχή Έκδοσης Πιστοποιητικών* (CAs) του εμπλεκόμενου μέρους στο οποίο ανήκουν.
* Δημιουργεί, επεξεργάζεται και αποστέλλει αιτήματα προς το σύστημα.
* Λαμβάνει, επεξεργάζεται και παρέχει στον χρήστη, απαντήσεις σε αιτήματα του ίδιου καθώς και ενημερώσεις ή αιτήματα που δημιουργούνται από τα υπόλοιπα εμπλεκόμενα μέρη ή το ίδιο το σύστημα .
* Είναι σχεδιασμένη με τέτοιο τρόπο όπου κάθε διαφορετική οντότητα της προτεινόμενης υλοποίησης (π.χ. Απλός χρήστης, Διαχειριστής ενός εμπλεκόμενου μέρους, Ελεγκτής) διαθέτει τη δική της διεπαφή (UI) η οποία χαρακτηρίζεται από την παροχή ενεργειών σχετικές με τις ενέργειες που αυτές οι οντότητες μπορούν να υλοποιήσουν.



**Εικόνα 2: ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΗ & ΤΥΠΟΙ ΧΡΗΣΤΩΝ**

Η οντότητα αυτή του συστήματος παρέχει τέσσερις (4) τύπους διεπαφής (UI), σύμφωνα με τους τύπους χρηστών που υποστηρίζονται από το σύστημα:

* **Διεπαφή Χρήστη** (Υπάλληλος ΕΑ(οδηγός, ελεγκτής κ.λπ.), Χρήστης Εφαρμογής κ.λπ.) – Είναι οποιαδήποτε οντότητα συνοδεύεται από ένα GID, δηλαδή ένα μοναδικό αναγνωριστικό
* **Διεπαφή Διαχειριστή εμπλεκόμενου μέρους** (Admin)
* **Διεπαφή Διαχειριστή Blockchain εμπλεκόμενου μέρους** (SCSS-Admin)
* **Διεπαφή Ελεγκτή** (Auditor)

**Διεπαφή Χρήστη**

Επιτρέπει τη δημιουργία μίας αίτησης προς τους εμπλεκόμενους οργανισμούς, τη λήψη παλαιότερων αιτήσεων οι οποίες δημιουργήθηκαν από το χρήστη και ολοκληρώθηκαν από το Blockchain σύστημα, τη λήψη των δεδομένων που συγκεντρώθηκαν από ένα ή περισσότερα εμπλεκόμενα μέρη και αφορούν μία αίτηση, καθώς και την ολική αποκρυπτογράφηση των δεδομένων, σε περίπτωση που τα δεδομένα μίας αίτησης προστατεύονται με τη μέθοδο της κρυπτογράφησης. Σημειώνουμε ότι η αποκρυπτογράφηση των πληροφοριών που παρέχονται από τα εμπλεκόμενα μέρη, υλοποιείται σε δύο σημεία: μία μερική αποκρυπτογράφηση των πληροφοριών η οποία εκτελείται στο επιχειρηματικό Blockchain και από τον οργανισμό που ανήκει ο αιτών ο οποίος τη δημιούργησε και μία πλήρης αποκρυπτογράφηση η οποία υλοποιείται από τον ίδιο τον κάτοχο του αιτήματος, μέσω του Electron Application και της επικοινωνίας του με το Hashicorp Vault.

**Διεπαφή Διαχειριστή Εμπλεκόμενου Μέρους (Admin)**

Επιτρέπει τη δημιουργία νέων ανθρώπινων οντοτήτων (π.χ. Χρήστη, Supply Cain System Security(SCSS)-Admin, Auditor), οι οποίες μπορούν να αλληλεπιδρούν με το σύστημα.

**Διεπαφή Διαχειριστή Blockchain εμπλεκόμενου μέρους (SCSS-Admin)**

Επιτρέπει την ανανέωση των στοιχείων αυτών τα οποία χαρακτηρίζουν ένα εμπλεκόμενο μέρος και τα οποία χρησιμοποιούνται από το σύστημα για τη δημιουργία ενός σημείου εμπιστοσύνης μεταξύ ενός οργανισμού και των χρηστών οι οποίοι ανήκουν σε αυτόν. Τα στοιχεία αυτά είναι 3 (τρία):

* Το *Πιστοποιητικό* (Root CA Certificate) μέσω του οποίου ο οργανισμός δημιουργεί οντότητες
* Η *Λίστα Ανάκλησης Πιστοποιητικών* (Certificate Revocation List, CRL), μέσω της οποίας ο οργανισμός μπορεί να κάνει ανάκληση των πιστοποιητικών που ανήκουν στους χρήστες του
* Η *Λίστα Ελέγχου Πρόβασης* (Access Control List, ACL), μέσω της οποίας οι χρήστες ενός οργανισμού μπορούν και λαμβάνουν συγκεκριμένες ιδιότητες, βάσει των ιδιαίτερων χαρακτηριστικών τους, στο σύστημα.

Επιπλέον, επιτρέπει τη δημιουργία ψηφοφοριών για την προετοιμασία της εισαγωγής ενός νέου οργανισμού στο σύστημα, καθώς και για την αφαίρεση ενός υφιστάμενου, ενώ, ακόμη, κάθε SCSS-Admin μπορεί να λαμβάνει μέρος στις ψηφοφορίες αυτές που δημιουργήθηκαν από άλλους SCSS-Admins ή Auditors του συστήματος, άλλων οργανισμών, και να επηρεάσει το τελικό αποτέλεσμα της ψηφοφορίας.

**Διεπαφή Ελεγκτή (Auditor)**

Επιτρέπει σε μία οντότητα με την ιδιότητα του Ελεγκτή να ζητάει από τα εμπλεκόμενα μέρη του συστήματος, πρόσβαση στις καταγραφές (Logs) του συστήματος. Η πρόσβαση ζητείται με τη μορφή ψηφοφορίας, την οποία μπορεί να δημιουργήσει ο ίδιος ο ελεγκτής. Κάθε ελεγκτής μπορεί να ζητάει πρόσβαση στις καταγραφές:

* Ολόκληρου του συστήματος
* Του επιχειρηματικού τομέα στον οποίο ανήκει
* Διαφορετικού επιχειρηματικού τομέα από αυτόν στον οποίο ανήκει

3.2.3 Λογισμικό διασύνδεσης Χρήστη – Σύστημα (Backend API)

Είναι το σημείο το οποίο λειτουργεί ως *μεσάζοντας* μεταξύ της εφαρμογής του χρήστη και του υπόλοιπου συστήματος. Γενικό του μέλημα είναι:

* Η λήψη και η προώθηση των αιτημάτων των χρηστών προς το υπόλοιπο σύστημα και, αντίστροφα, η λήψη και η προώθηση πληροφορίας από το σύστημα προς τους χρήστες.
* Ο έλεγχος του χρήστη που αποστέλλει αιτήματα στο σύστημα, προκειμένου να ρυθμίζονται οι οντότητες που επικοινωνούν με αυτό. Το σύστημα δαπανά επεξεργαστική ισχύ μόνο για οντότητες οι οποίες είναι διαπιστωμένο ότι έχουν δημιουργηθεί και ανήκουν σε ένα εμπλεκόμενο μέρος που συμμετέχει στο σύστημα.
* Ο έλεγχος και ο συντονισμός των ταυτόχρονων χρηστών που χρησιμοποιούν το σύστημα. Με την ύπαρξη ενός ανώτατου ορίου ταυτόχρονων χρηστών, το σύστημα παραμένει υγιές ώστε να εξυπηρετεί με τον ίδιο τρόπο και χωρίς διακρίσεις όλα τα εμπλεκόμενα μέρη που το απαρτίζουν.

Στην οντότητα αυτή υφίστανται και εκτελούν λειτουργίες τα ίδια τα εμπλεκόμενα μέρη του συστήματος. Κάθε ενέργεια που εκτελείται από έναν χρήστη, αρχικά συντονίζεται από το Backend API και τελικά παραλαμβάνεται και ολοκληρώνεται από το ίδιο το εμπλεκόμενο μέρος στο οποίο αυτός ο χρήστης ανήκει. Κάθε εμπλεκόμενο μέρος συμμετέχει, ως ξεχωριστή οντότητα, στη λειτουργία του Backend API.

Με την ύπαρξη των εμπλεκόμενων μερών σε αυτό το σημείο της υλοποίησης, το σύστημα αποκτά επιπλέον διαφάνεια ενώ κάθε εμπλεκόμενο μέρος είναι υπεύθυνο για την εξυπηρέτηση των δικών του χρηστών.

Πιο συγκεκριμένα, το σύστημα αυτό αναπτύσσει τις παρακάτω λειτουργίες:

|  |
| --- |
| **Δημιουργία ειδικού διαπιστευτηρίου Χρήστη** |
| Κάθε χρήστης ο οποίος επιθυμεί να επικοινωνήσει με το σύστημα, πρέπει πρώτα να λάβει ένα *ειδικό διαπιστευτήριο* (token) το οποίο μόνο η συγκεκριμένη οντότητα μπορεί να δημιουργήσει. Κατά την αυθεντικοποίηση του χρήστη με την *Αρχή έκδοσης πιστοποιητικών* του εμπλεκόμενου μέρους στο οποίο ανήκει, ο χρήστης αποστέλλει ένα αίτημα στο Backend API, προκειμένου να λάβει πίσω ένα τεκμήριο το οποίο θα μπορεί, μετέπειτα, να χρησιμοποιεί ώστε να κάνει χρήση των υπηρεσιών του συστήματος. Για τη δημιουργία αυτού του τεκμήριου, λαμβάνει μέρος και το εμπλεκόμενο μέρος στο οποίο ανήκει η οντότητα.  Shape  Description automatically generated with medium confidence |
| **Αίτηση επικοινωνίας με το Blockchain σύστημα** |
| Όταν ένας χρήστης επιθυμεί να επικοινωνήσει με το Blockchain σύστημα, αρχικά, και μέσω της Εφαρμογής Χρήστη (Client Application), δημιουργεί ένα αίτημα προς το Backend API στο οποίο   1. δηλώνει την επιθυμία του να επικοινωνήσει με το Blockchain και 2. ζητάει τη δημιουργία ενός εισιτηρίου (ticket), με το οποίο θα λάβει εξουσιοδότηση για επικοινωνία   Αφού το Backend API λάβει το αίτημα αυτό, ελέγχει τον χρήστη καθώς και το ειδικό τεκμήριο το οποίο έλαβε κατά την αυθεντικοποίηση του *(****1****)* και εισάγει το αίτημα σε μία ουρά αναμονής. Αφού το αίτημα βγει από την ουρά αναμονής, δηλαδή δεν υφίστανται άλλα αιτήματα τα οποία είναι προς ολοκλήρωση, η οντότητα Backend API δημιουργεί ένα ειδικό εισιτήριο για τον χρήστη, το οποίο θα μπορεί να χρησιμοποιηθεί για πολύ συγκεκριμένο χρόνο και πράξεις πάνω στο Blockchain σύστημα. Κάθε εισιτήριο είναι μοναδικό και αφορά τον χρήστη για τον οποίο δημιουργήθηκε.  Shape  Description automatically generated with medium confidence  Πρέπει να σημειωθεί, ότι δεν απαιτείται να δημιουργείται ένα εισιτήριο για κάθε ενέργεια ενός χρήστη προς το σύστημα. Τα εισιτήρια αυτά είναι απαραίτητα **μόνο** για την επικοινωνία των διαφόρων ανθρώπινων οντοτήτων με το Blockchain υποσύστημα (Proxy BC), το οποίο θα πρέπει να λαμβάνει συγκεκριμένο αριθμό ταυτόχρονων αιτημάτων, ώστε να μη διακυβεύεται η λειτουργία του. |
| **Επικοινωνία Χρήστη με Blockchain σύστημα** |
| Επιγραμματικά, υφίστανται δύο μεγάλες κατηγορίες ενεργειών, πάνω στα Blockchains του συστήματος:   * η ανάγνωση δεδομένων που είναι αποθηκευμένα και προστατεύονται από το Blockchain (*Evaluate*) * η εισαγωγή νέων δεδομένων και αποθήκευσή τους στο προστατευμένο περιβάλλον του Blockchain (*Submit*)   Όπως ήδη αναφέρθηκε, κάθε ενέργεια των χρηστών πάνω στο κυρίαρχο Blockchain του συστήματος (Proxy BC) οφείλει να συνοδεύεται και από ένα εισιτήριο που θα εξουσιοδοτεί την ενέργεια αυτή.  Αφού ο χρήστης λάβει ένα εισιτήριο από το Backend API, μπορεί να εκτελεί τις παραπάνω ενέργειες. Κάθε εισιτήριο είναι άρρηκτα συνδεδεμένο με την ενέργεια που ο χρήστης θέλει να εκτελέσει, συνεπώς κάθε ενέργεια του χρήστη πάνω στο Blockchain είναι συγκεκριμένη και δε μπορεί να υπάρξει δόλος. |

3.2.5 Smart Contracts

Στην υλοποίηση αναπτύσσονται έξη (6) Smart Contracts (Chaincodes), από τα οποία καθένα υλοποιεί συγκεκριμένη λειτουργικότητα ενώ για κάθε Blockchain, αναπτύσσονται συγκεκριμένα Smart Contracts, σύμφωνα με τις λειτουργίες και ανάγκες που το καθένα είναι επιφορτισμένο να υλοποιεί και έχει.

Συγκεκριμένα,

|  |  |
| --- | --- |
| Proxy Blockchain (PBC) | |
| TMSC, PSC, LSC, DHSC | |
| Organizational Domain Blockchain (ODBC) | **Supply Chain Blockchain(SCBC)** |
| ACSC, KSSC | ACSC, KSSC |

#### TMSC (trust management smart contract)

Το TMSC αποτελεί το Smart Contract του συστήματος στο οποίο αναπτύσσονται οι παρακάτω κύριες δυνατότητες:

* Μηχανισμοί προσθαφαίρεσης εμπλεκόμενων μερών
* Παροχή λειτουργιών δυναμικής ανανέωσης των στοιχείων τα οποία αφορούν ένα εμπλεκόμενο μέρος (Πιστοποιητικά, Λίστες ανάκλησης, Λίστες πρόσβασης)
* Αξίωση πρόσβασης χρήστη στο Blockchain σύστημα

Συγκεκριμένα,

**Μηχανισμοί Προσθαφαίρεσης Εμπλεκόμενων Μερών**

Το σύστημα επιτρέπει την προετοιμασία για την εισαγωγή ή την αφαίρεση ενός νέου ή υφιστάμενου εμπλεκόμενου μέρους, αντίστοιχα.

Πριν την συμμετοχή ενός οργανισμού στα επιμέρους Blockchain συστήματα (Proxy και Domain Blockchains), είναι απαραίτητη η εισαγωγή των στοιχείων αυτών που αφορούν τον οργανισμό αυτόν (Πιστοποιητικά, Λίστες ανάκλησης, Λίστες πρόσβασης). Για αυτό, το TMSC αναπτύσσει τη δυνατότητα ένα υπάρχον εμπλεκόμενο μέρος να είναι σε θέση να εισάγει αυτά τα δεδομένα και τα υπόλοιπα εμπλεκόμενα μέρη του συστήματος να αποφασίζουν, μέσω της διεξαγωγής ψηφοφορίας, εάν επιτρέπουν την εισαγωγή του νέου οργανισμού που προτάθηκε.

Επιπλέον, δίνεται η δυνατότητα ένα εμπλεκόμενο μέρος να ξεκινά μία διαδικασία αφαίρεσης ενός οργανισμού από τα εμπλεκόμενα μέρη. Μέσω της διαδικασίας αυτής, τα στοιχεία τα οποία συνθέτουν ένα εμπλεκόμενο μέρος (Πιστοποιητικά, Λίστες ανάκλησης, Λίστες πρόσβασης), έπειτα από τη διεξαγωγή ψηφοφορίας, γίνονται ανάκληση και οι χρήστες που ανήκουν στον ανακληθέν οργανισμό παύουν να έχουν πρόσβαση σε ευαίσθητες και σημαντικές λειτουργίες του συστήματος.

**Δυναμική Ανανέωση Στοιχείων Εμπλεκόμενου Μέρους**

Κάθε οντότητα που εκτελεί μία ενέργεια στο Blockchain σύστημα, ελέγχεται για τα χαρακτηριστικά της από τον οργανισμό στον οποίο ανήκει ή τους οργανισμούς τους οποίους η ενέργεια αυτή αφορά, ενώ κάθε οργανισμός είναι υπεύθυνος για την ανανέωση των πληροφοριών που χρησιμοποιούνται για τον έλεγχο των οντοτήτων αυτών.

Κάθε εμπλεκόμενο μέρος του συστήματος, είναι υπεύθυνο για την ανανέωση των στοιχείων αυτών τα οποία το αφορούν. Τα στοιχεία αυτά είναι τρία (3):

* Πιστοποιητικό (Root CA Certificate) μέσω του οποίου δημιουργούνται τα πιστοποιητικά των χρηστών του οργανισμού για το σύστημα
* Λίστα ανάκλησης (Certificate Revocation List, CRL) μέσω της οποίας ο οργανισμός ακυρώνει τα πιστοποιητικά των χρηστών του τα οποία δημιούργησε σε προγενέστερο χρόνο
* Λίστα πρόσβασης (Access Control List, ACL) μέσω της οποίας δηλώνονται τα ιδιαίτερα χαρακτηριστικά και πληροφορίες ενός χρήστη (π.χ. ρόλος, δικαιώματα πρόσβασης κλπ)

**Αξίωση Πρόσβασης Χρήστη στο Blockchain σύστημα**

Ένας χρήστης μπορεί να λάβει πρόσβαση στο σύστημα και να εκτελέσει ευαίσθητες και σημαντικές ενέργειες, μόνο έπειτα από τον έλεγχο των γνωρισμάτων και των χαρακτηριστικών του. Για αυτό, στο Proxy BC και το TMSC αναπτύσσονται λειτουργίες που κάνουν χρήση των πληροφοριών του οργανισμού (Πιστοποιητικά, Λίστες ανάκλησης, Λίστες πρόσβασης) και προσφέρουν πρόσβαση στο σύστημα μόνο σε πιστοποιημένους χρήστες οι οποίο διαθέτουν τα απαραίτητα δικαιώματα και τεκμήρια για την ενέργεια που εκτελούν.

#### PSC (Proxy Smart Contract)

Βασικό μέλημα του PSC αποτελεί η διαχείριση της πληροφορίας που διακινείται στο σύστημα (λειτουργίες πρόσβασης, λήψης κ.ο.κ.). Θεμελιώδεις λειτουργίες του, είναι:

* Η αίτηση πρόσβασης στα δεδομένα ενός ή περισσότερων εμπλεκόμενων μερών, από τους χρήστες του συστήματος καθώς και η λήψη των δεδομένων που ελήφθησαν από ένα ή περισσότερα εμπλεκόμενα μέρη, από τους χρήστες του συστήματος
* Η έναρξη, η διαχείριση και ο τερματισμός των ψηφοφοριών που εκτελούνται στο σύστημα

**Λήψη και Πρόσβαση σε Δεδομένα**

Όταν ένας χρήστης επιθυμεί τη λήψη πληροφοριών που κατέχει ένα, ή περισσότερα του ενός, εμπλεκόμενα μέρη, δημιουργεί ένα αίτημα προς το PSC. Το PSC ελέγχει τον χρήστη, έπειτα από επικοινωνία του με το TMSC, καταγράφει το αίτημα, μέσω του LSC, και εκτελεί την ενέργεια που του ανατέθηκε. Συγκεκριμένα, συγκεντρώνει όλα τα ιδιαίτερα χαρακτηριστικά του χρήστη (π.χ. ρόλος, δικαιώματα πρόσβασης κ.λπ.), μέσω της Λίστας Πρόσβασης και τα αποστέλλει στο αντίστοιχο επιχειρηματικό Blockchain που ανήκει ο οργανισμός του χρήστη, για περαιτέρω επεξεργασία.

**Ψηφοφορίες Συστήματος**

Στο σύστημα αναπτύσσονται τρία (3) είδη ψηφοφορίας:

* Προσθήκη ενός νέου οργανισμού ως εμπλεκόμενο μέρος
* Αφαίρεση ενός οργανισμού από τα εμπλεκόμενα μέρη
* Αξίωση πρόσβασης στις καταγραφές (Logs) του συστήματος

Το PSC είναι υπεύθυνο να εκκινεί, να ρυθμίζει, να επεξεργάζεται και να τερματίζει τις ψηφοφορίες αυτές καθώς και να ενημερώνει το σύστημα συνολικά σύμφωνα με το αποτέλεσμα.

#### LSC (Logging Smart Contract)

Ευθύνη του LSC είναι η καταγραφή και διατήρηση των ενεργειών που εκτελέστηκαν στο σύστημα από τους χρήστες του.

Για παράδειγμα, ένας εκλεκτικός φορέας αιτήθηκε την λήψη των πληροφοριών ενός προϊόντος ΕΑ. Το LSC θα καταγράψει το αίτημα, το οποίο περιλαμβάνει τον οργανισμό που αφορά, τα στοιχεία του προϊόντος (π.χ. κωδικός προϊόντος, παρτίδα, barcode) καθώς επίσης και τα στοιχεία του εταιρίας που μεταφέρει το προϊόν στα πλαίσια της ΕΑ και θα τα αποθηκεύσει στο Proxy BC.

Μέσω του LSC δημιουργείται ένα σύστημα καταγραφής το οποίο, με την προϋπόθεση ότι η πλειοψηφία των εμπλεκόμενων μερών είναι έμπιστη, θα παρέχει πληροφορίες σχετικές με τις ενέργειες που έχουν εκτελεστεί και ολοκληρωθεί και οι οποίες δεν μπορούν να αλλοιωθούν από κανένα από τα εμπλεκόμενα μέρη.

#### ACSC (Access Control smart Contract)

Με την επιτυχή επεξεργασία ενός αιτήματος από το Proxy BC και την προώθηση του, μέσω του Λογισμικού Διασύνδεσης Blockchain Συστημάτων (Inter-Blockchain API), στο κατάλληλο Domain BC, το αίτημα προετοιμάζεται για την ολοκλήρωση του.

Όμως, πριν ένα αίτημα ολοκληρωθεί, είναι απαραίτητο να εφαρμοστούν σε αυτό οι πολιτικές του εμπλεκόμενου μέρους ή οι πολιτικές των εμπλεκόμενων μερών που αυτό αφορά.

Για αυτό, το ACSC ελέγχει εάν ο χρήστης ανταποκρίνεται στις προϋποθέσεις που έχει θέσει το καθένα εμπλεκόμενο μέρος, μέσω των πληροφοριών που συγκεντρώθηκαν, για τον χρήστη που δημιούργησε το αίτημα, από το TMSC και τη Λίστα πρόσβασης (ACL) του εμπλεκόμενου μέρους που αφορά το αίτημα.

Πιο συγκεκριμένα, σε κάθε αίτημα ενός χρήστη, το TMSC συγκεντρώνει για αυτόν τις πληροφορίες που υπάρχουν στη Λίστα πρόσβασης (ACL) του εμπλεκόμενου μέρους και τις εισάγει στο αίτημα. Το ACSC, αφού παραλάβει το αίτημα, ελέγχει τις πληροφορίες αυτές προκειμένου να διαπιστώσει αν ο χρήστης έχει τη δυνατότητα να λάβει τα δεδομένα που αιτήθηκε.

Diagram

Description automatically generated

**Εικόνα 3: ΑΞΙΩΣΗ ΠΡΟΣΒΑΣΗΣ ΣΕ ΠΛΗΡΟΦΟΡΙΕΣ ΜΕ ΤΗ ΧΡΗΣΤΗ ΠΟΛΙΤΙΚΩΝ**

#### KSSC (Key Store Smart Contract)

Όπως έχει αναφερθεί, η προστασία της πληροφορίας που λαμβάνεται από τα εμπλεκόμενα μέρη και διακινείται στο σύστημα, αποτελεί σημείο υψίστης προτεραιότητας και σημασίας.

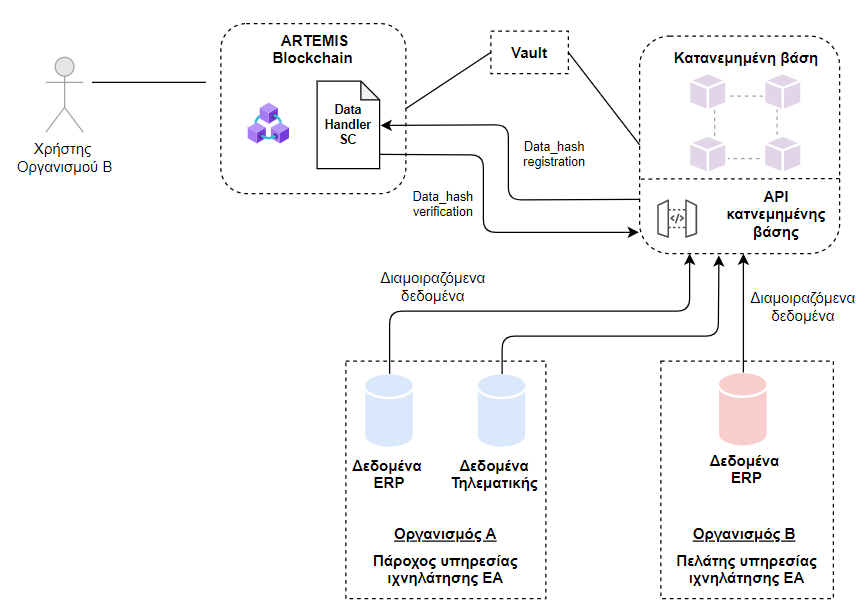
Κάθε ευαίσθητη πληροφορία, αποκρυπτογραφείται σε δύο (νοητά) επίπεδα. Ένα επίπεδο το οποίο αφορά τον οργανισμό και ένα επίπεδο το οποίο αφορά τον χρήστη ο οποίος αιτείται πρόσβασης στην πληροφορία. Αυτό επιτυγχάνεται με τη χρήση της τεχνολογίας του Attribute-Based Encryption (ABE Cryptography).

Αφού το ACSC επιτρέψει την πρόσβαση στα δεδομένα ενός εμπλεκόμενου μέρους, για ένα αίτημα που δημιουργήθηκε από έναν χρήστη, αποκρυπτογραφεί, μερικώς, με τη βοήθεια του KSSC, τα δεδομένα που έλαβε με ένα ABE Attribute το οποίο κατέχει κάθε οργανισμός ξεχωριστά (και στο οποίο οι χρήστες του συστήματος δεν έχουν πρόσβαση), ενώ ταυτόχρονα, κατά τη διαδικασία αυτή της μερικής αποκρυπτογράφησης, τα μερικώς αποκρυπτογραφημένα δεδομένα *δένονται* με τον χρήστη και το μοναδικό αναγνωριστικό που τον χαρακτηρίζει (GID) ώστε μόνον αυτός να μπορεί να τα αποκρυπτογραφήσει πλήρως. Δηλαδή, διαφορετικοί χρήστες δε μπορούν να συνδυάσουν τα κλειδιά τους προκειμένου να αποκρυπτογραφήσουν πληροφορία της οποίας κανένας χρήστης δε διαθέτει το σύνολο των κλειδιών που απαιτούνται από την ABE Πολιτική (ABE Policy) σύμφωνα με την οποία αυτή κρυπτογραφήθηκε.

Συνοψίζοντας, η διαδικασία της αποκρυπτογράφησης των δεδομένων που λαμβάνονται από τα εμπλεκόμενα μέρη, συμβαίνει σε δύο σημεία. Το ένα από αυτά, υλοποιείται στο KSSC και κάθε μερική αποκρυπτογράφηση συμβαίνει με τέτοιο τρόπο ώστε να είναι μοναδική για κάθε χρήστη, υποστηρίζοντας περαιτέρω την ασφάλεια και διαφάνεια του συστήματος.

#### DHSC (Data Handler Smart Contract)

Το Έξυπνο Συμβόλαιο που αναλαμβάνει την διαχείριση των δεδομένων. Μηχανισμός ελέγχου ακεραιότητας δεδομένων



**Εικόνα 4: Έξυπνο Συμβολαιο Διαχειρισης Δεδομένων (DHSC)**

3.2.6 Λογισμικό Διασύνδεσης Blockchain Συστημάτων (Inter-Blockchain API)

Για τη διασύνδεση των δύο διαφορετικών Blockchain οντοτήτων (Proxy BC – Domain BC), είναι αναγκαία η ύπαρξη ενός μεσάζοντα ο οποίος θα:

* λαμβάνει πληροφορίες από το ένα άκρο και θα τις μεταφέρει στο άλλο, δηλαδή από το Proxy BC προς ένα Domain BC και από ένα Domain BC προς το Proxy BC
* Θα ενορχηστρώνει τον τρόπο με τον οποίο οι δύο αυτές οντότητες θα αντλούν και θα ανταλλάσσουν πληροφορίες αναμεταξύ τους. Δηλαδή, θα ελέγχει και θα επεξεργάζεται τις ροές, τους αποστολείς και τους παραλήπτες των πληροφοριών

Όπως και στο Backend API, το οποίο, υπενθυμίζουμε, λαμβάνει το ρόλο της μετάδοσης της πληροφορίας από έναν χρήστη προς το κύριο Blockchain και αντίστροφα, έτσι και σε αυτό το λογισμικό, επίσης συμμετέχουν όλα τα εμπλεκόμενα μέρη που συμμετέχουν στο σύστημα. Κάθε επικοινωνία που συμβαίνει μεταξύ των κόμβων των δύο δικτύων (Proxy BC – Domain BC), συμβαίνει μέσω των οντοτήτων που έχει διαθέσει το καθένα εμπλεκόμενο μέρος στο Inter-Blockchain API.

A picture containing graphical user interface

Description automatically generated

**Εικόνα 5: ΕΜΠΛΕΚΟΜΕΝΑ ΜΕΡΗ & ΠΡΟΩΘΗΣΗ ΑΙΤΗΜΑΤΟΣ**

Συγκεκριμένα, στο Inter-Blockchain API έχουν ανατεθεί οι παρακάτω αρμοδιότητες:

* Μεταβίβαση της πληροφορίας μεταξύ των Blockchains του συστήματος
* Διαχείριση λειτουργιών οι οποίες αναπτύσσονται στο Blockchain, όμως υπάρχει η ανάγκη για μία εξωτερική οντότητα η οποία θα τις πλαισιώνει, με την έννοια ότι θα υποβοηθά το Blockchain προκειμένου να τις φέρνει σε πέρας (π.χ. Ψηφοφορίες)

**Inter-Blockchain API, Blockchains και Συμβάντα**

Κάθε σημαντική ενέργεια που συμβαίνει στο Blockchain και η οποία μπορεί να επιφέρει αλλαγές στην κατάσταση του συστήματος συνολικά, συνοδεύεται από ένα συμβάν το οποίο εκπέμπεται από το ίδιο το Blockchain.

Γενικά, το Hyperledger Fabric μας δίνει τη δυνατότητα για κάθε συναλλαγή που εκτελείται στο δίκτυο του, να εκπέμπεται μέχρι και ένα (1) συμβάν (event). Στην παρούσα υλοποίηση, εκμεταλλευόμαστε αυτήν τη δυνατότητα που παρέχεται και τη χρησιμοποιούμε προς όφελος του συστήματος.

Το Inter-Blockchain API, *αναμένει & λαμβάνει* συνεχώς και αδιαλείπτως συμβάντα τα οποία εκπέμπονται από το Proxy Blockchain.

Σημαντικά συμβάντα είναι:

|  |
| --- |
| Μεταφορά αιτήματος από το Proxy BC σε ένα Domain BC |
| Ολοκλήρωση αιτήματος |
| Έναρξη ή Λήξη Ψηφοφορίας |
| Υποβολή ψήφου από ένα εμπλεκόμενο μέρος, σε μία ψηφοφορία |
| Η αποδοχή ή η απόρριψη ενός αιτήματος, σύμφωνα με τις Πολιτικές του συστήματος και του εμπλεκόμενου μέρους που αφορά |
| Η αίτηση ενός Ελεγκτή να λάβει εξουσιοδότηση για πρόσβαση στις καταγραφές του συστήματος |

**Μεταβίβαση Πληροφορίας**

Όπως παρουσιάζεται και παραπάνω, το Inter-Blockchain API αναμένει συνεχώς για συμβάντα από το Proxy BC τα οποία ζητούν τη μεταφορά ενός αιτήματος σε ένα Domain BC. Μέσα στα συμβάντα αυτά, υπάρχει εμφωλευμένη η συνολική πληροφορία που εμπεριέχει το αίτημα το οποίο πρέπει να διαβιβαστεί.

Αφού το Inter-Blockchain API παραλάβει ένα συμβάν το οποίο αφορά τη μετάδοση ενός αιτήματος από το Proxy BC, πρώτη ενέργεια που εκτελεί είναι να εξασφαλίσει ότι το δίκτυο δε βρίσκεται στα όρια του. Αυτό το πραγματοποιεί βάζοντας το αίτημα που έλαβε σε μία ουρά αιτημάτων, η οποία θα βεβαιώσει πως υπάρχει περιορισμός στον αριθμό τον ταυτόχρονων αιτημάτων που εκτελούνται στο σύστημα.

Στο σύστημα υφίστανται δύο (2) μεγάλοι τύποι ουρών.  
Μία ουρά η οποία αφορά το Proxy BC και από μία ουρά για κάθε επιχειρηματικό Blockchain (HDBC, MDBC).

Shape

Description automatically generated with medium confidence

**Εικόνα 6: INTER-BLOCKCHAIN API & ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ ΟΓΚΟΥ ΤΩΝ ΑΙΤΗΜΑΤΩΝ ΜΕ ΟΥΡΕΣ**

Μία *υψηλού επιπέδου* απεικόνιση των όσων συμβαίνουν όταν το Inter-Blockchain API λάβει ένα αίτημα, παρουσιάζεται παραπάνω.

**Διαχείριση Ψηφοφοριών**

Όπως αναφέρθηκε, το Inter-Blockchain API χρησιμοποιείται και για λειτουργίες οι οποίες υποβοηθούν το Blockchain σύστημα, συνολικά (Proxy BC – Domain BCs) να υλοποιεί και να ολοκληρώνει διάφορες λειτουργίες του.

Λειτουργία η οποία ολοκληρώνεται σε δυνατότητες μέσω του Inter-Blockchain API, είναι οι *Ψηφοφορίες* που συμβαίνουν στο σύστημα και η διαχείριση αυτών.

Μέσω του συγκεκριμένου λογισμικού, το Proxy Blockchain και οι ψηφοφορίες που υλοποιούνται, μπορούν και εκτελούνται σε πραγματικό χρόνο. Τυχόν απουσία της συγκεκριμένης λειτουργίας, θα σήμαινε ότι η παρουσία του ανθρώπινου παράγοντα για την ολοκλήρωση των διαδικασιών που αφορούν τις ψηφοφορίες του συστήματος, θα ήταν απαραίτητη.

Στόχος του Inter-Blockchain API είναι η συνεχής ενημέρωση του Blockchain για τη λήξη μίας ψηφοφορίας (λόγω συμπλήρωσης ψήφων ή λήξης χρόνου). Για αυτό, το Inter-Blockchain API λαμβάνει, συγκρατεί και επεξεργάζεται όλες τις πληροφορίες που λαμβάνει από το Blockchain σχετικά με τις ψηφοφορίες που συμβαίνουν σε αυτό.

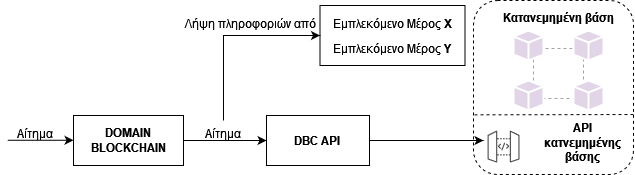
3.2.7 Λογισμικό Διασύνδεσης Συστήματος – Βάσεων Δεδομένων (DBC API)

Τα εμπλεκόμενα μέρη του συστήματος δεν αποθηκεύουν πληροφορίες που διατηρούν, άμεσα στο σύστημα.

Αντιθέτως, κάθε αίτημα που λαμβάνεται από το σύστημα και για το οποίο ζητείται η πληροφορία που κατέχει ένα εμπλεκόμενο μέρος (π.χ. πληροφορίες για την κατάσταση ενός προϊόντος(θερμοκρασία), πληροφορίες το μέσο μεταφοράς (ακριβής τοποθεσία φορτηγού)), λαμβάνεται σε πραγματικό χρόνο από τις βάσεις δεδομένων του κάθε εμπλεκόμενου μέρους.

Με τη διαδικασία της ολοκλήρωσης της ταυτοποίησης του χρήστη και της προώθησης του αιτήματος στο κατάλληλο Domain Blockchain, το Domain Blockchain και το Smart Contract το οποίο έχει αναλάβει τη διαδικασία αυτή (ACSC), επικοινωνεί με το συγκεκριμένο λογισμικό προκειμένου να ληφθούν οι πληροφορίες που ζητούνται από το αίτημα, από τη βάση ή τις βάσεις δεδομένων των εμπλεκόμενων μερών που αυτό αφορά.

Συνοπτικά, μέλημα του συγκεκριμένου λογισμικού είναι η διασύνδεση του συστήματος με τις βάσεις δεδομένων των οργανισμών οι οποίοι αποτελούν τα εμπλεκόμενα μέρη του συστήματος.

****

**Εικόνα 7: ΛΗΨΗ ΠΛΗΡΟΦΟΡΙΩΝ ΑΠΟ ΤΑ ΕΜΠΛΕΚΟΜΕΝΑ ΜΕΡΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ**

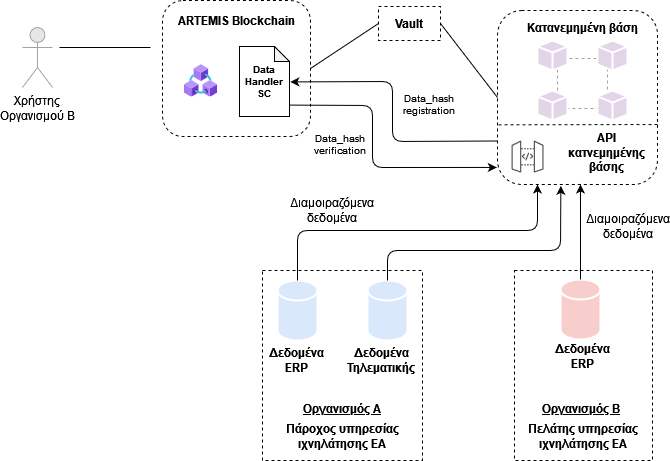
## Αρχιτεκτονική συστήματος

Το κεφάλαιο αυτό ασχολείται με την αρχιτεκτονική σχεδίαση του συστήματος.

Θα παρουσιαστούν τα μέρη (components) από τα οποία αποτελείται το σύστημα, οι διασυνδέσεις που δημιουργούνται μεταξύ τους καθώς και οι οντότητες οι οποίες μπορούν και υλοποιούν ενέργειες.

Η παρούσα υλοποίηση έχει αναπτυχθεί βάσει της αρθρωτής αρχιτεκτονικής (modular architecture). Όπως παρουσιάστηκε κατά την προηγούμενη ενότητα, κάθε επιμέρους στοιχείο του συστήματος επιτελεί συγκεκριμένη λειτουργία. Όλες οι λειτουργίες συνολικά, συντελούν στο σχεδιασμό του τελικού συστήματος.

Όλα τα λογισμικά του συστήματος εκτελούνται σε containerized περιβάλλον και η διαχείριση αυτών επιτυγχάνεται με τη χρήση του λογισμικού *Kubernetes*.



## Διαγράμματα Ροής Δεδομένων (Data Flow Diagrams)

Στην υποενότητα αυτή θα γίνει ανάπτυξη των διαγραμμάτων ροής δεδομένων του συστήματος.

Η ανάπτυξη θα γίνει σε δύο επίπεδα,

* **Επίπεδο 0**: Θα απεικονιστεί το σύστημα ως σύνολο
* **Επίπεδο 1**: Θα παρουσιαστούν συνολικά όλα τα μέρη από τα οποία αποτελείται η υλοποίηση. Επίσης, θα απεικονιστούν οι σχέσεις που έχουν τα διάφορα μέρη του συστήματος αναμεταξύ τους, ενώ ακόμη θα φανούν τα εσωτερικά (internal) κύρια στοιχεία από τα οποία αποτελείται κάθε λογισμικό του συστήματος.

### Γενικό (context) Διάγραμμα – Επίπεδο 0

Όπως απεικονίζεται στο διάγραμμα Επιπέδου 0, κάθε λογισμικό το οποίο εκτελείται στο σύστημα, εκτελείται μέσα σε ένα απομονωμένο container, ενώ για τη διαχείριση αυτών των containers (έλεγχος, αποσφαλμάτωση, λειτουργία κ.λπ.), υπεύθυνο είναι το Kubernetes.

Diagram

Description automatically generated

**Εικόνα 8:** ΔΙΑΓΡΑΜΜΑ ΕΠΙΠΕΔΟΥ 0

### Διάγραμμα – Επίπεδο 1

Στο διάγραμμα *επιπέδου 1*, παρουσιάζονται όλα τα μέρη που συμμετέχουν στη λειτουργία της υλοποίησης καθώς και οι αναμεταξύ τους σχέσεις που δημιουργούνται, ανά στρώμα (Layer) λειτουργικότητας.

|  |
| --- |
| Στρώμα Παρουσίασης (Presentation Layer) |
| * Λογισμικό Εφαρμογής Χρήστη (Electron Application - UI) |
| Στρώμα Υπηρεσιών (Service Layer) |
| * Λογισμικό Εφαρμογής Χρήστη (Electron Application) * Λογισμικό Διασύνδεσης Χρήστη – Σύστημα (Backend API) * Λογισμικό Διασύνδεσης Blockchain Συστημάτων (Inter-Blockchain API) * Blockchain Συστήματα (Proxy BC, Domain BCs) * Λογισμικό Διασύνδεσης Συστήματος – Βάσεων Δεδομένων (DBC API) * Αρχή έκδοσης πιστοποιητικών (CA) εμπλεκόμενου μέρους * Hashicorp’s Vault ABE Plugin |
| Στρώμα Δεδομένων (Data Layer) |
| * Βάσεις Δεδομένων των εμπλεκόμενων μερών |

# Κατανεμημένη υποδομή και υπηρεσίες

## Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστούν, θα αναπτυχθούν και θα αναλυθούν όλες οι οντότητες από τις οποίες αποτελείται η υφιστάμενη υπηρεσία ασφάλειας.

Κάθε οντότητα αναλαμβάνει σαφή ρόλο και επιτελεί συγκεκριμένο έργο. Όλες μαζί συντελούν μία ολότητα, μέσω της οποίας ικανοποιούνται ορισμένες απαιτήσεις που έχουν παρουσιαστεί σε προηγούμενα παραδοτέα [].

Το σύστημα απαρτίζεται από συνολικά επτά (7) κύριες οντότητες. Ονομαστικά, οι οντότητες αυτές είναι:

* [Λογισμικό εφαρμογής χρήστη του συστήματος](#_heading=h.4fsjm0b) (Client Application**1**)
* [Λογισμικό διασύνδεσης της εφαρμογής χρήστη με το υπόλοιπο σύστημα](#_heading=h.2uxtw84) (Backend API**2**)
* [Το κύριο Blockchain σύστημα (Proxy BC**3**) καθώς και τα επιμέρους επιχειρηματικά Blockchains](#_heading=h.1a346fx) (Domain BCs**4**)
* [Λογισμικό διασύνδεσης του Proxy Blockchain με τα Blockchains κάθε επιχειρηματικού τομέα](#_heading=h.3u2rp3q) (Inter-Blockchain API**5**)
* [Λογισμικό διασύνδεσης του συστήματος με τις Βάσεις Δεδομένων των εμπλεκόμενων μερών](#_heading=h.2981zbj) (DBC API**6**)
* [Εφαρμογή που ασχολείται με την κρυπτογράφηση και την αποκρυπτογράφηση ευαίσθητων δεδομένων που διακινούνται στο σύστημα](#_heading=h.odc9jc) (Hashicorp Vault**7**)

Πρακτικά, στο σύστημα υφίσταται και μία όγδοη (8η) οντότητα, η οποία είναι η *Αρχή έκδοσης πιστοποιητικών* (*Certificate Authority*, CA). Κάθε εμπλεκόμενο μέρος του συστήματος (Υπεύθυνο υπουργείο, Οργανισμοί), διαθέτει από μία τέτοια οντότητα, της οποίας μέλημα είναι η παραγωγή, η ανάκληση και ο έλεγχος των πιστοποιητικών των οντοτήτων του συστήματος (ανθρωπίνων, υλικών ή άυλων οντοτήτων). Κάθε εμπλεκόμενο μέρος είναι υπεύθυνο για την κατασκευή, τη συντήρηση και τον έλεγχο της συγκεκριμένης οντότητας.

## Τεχνολογίες συστήματος

Πριν γίνει εκτενής ανάλυση της κάθε οντότητας, των στοιχείων της, καθώς και του τρόπου που διασυνδέεται και επηρεάζει το σύστημα, θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν στο σύστημα.

**NodeJS**

*Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine*Εικόνα που περιέχει κείμενο, clipart, ανυσματικά γραφικά

Περιγραφή που δημιουργήθηκε αυτόματα

[nodejs.org](http://www.nodejs.org/)

Η NodeJS είναι μία πλατφόρμα που υποστηρίζει την κατασκευή εφαρμογών που εκτελούνται σε περιβάλλον server και στηρίζεται στη γλώσσα προγραμματισμού JavaScript.

Σύμφωνα με το OpenJS Foundation, η NodeJS είναι:

* Ένα JavaScript runtime το οποίο εκτελείται πάνω στην V8 JavaScript engine του Chrome
* Χρησιμοποιεί ένα event-driven, non-blocking I/O μοντέλο, το οποίο την υποστηρίζει έτσι ώστε να έχει μικρές υπολογιστικές απαιτήσεις (lightweight) και να είναι αποτελεσματική (efficient)
* Το οικοσύστημα πακέτων το οποίο είναι διαθέσιμο (npm), είναι το μεγαλύτερο οικοσύστημα βιβλιοθηκών (libraries) ανοικτού λογισμικού που υπάρχει στον κόσμο
* Λόγω του τρόπου που λειτουργεί (asynchronous event driven JS runtime), μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαδικτυακών εφαρμογών που μπορούν να επεκταθούν πολύ εύκολα (scalable network applications)

**ExpressJS**

*Fast, unopinionated, minimalist web framework for Node.js*

[expressjs.com](https://expressjs.com/) Company name

Description automatically generated with medium confidence

Το ExpressJS αποτελεί ένα από τα πιο διαδεδομένα frameworks για το σχεδιασμό και την ανάπτυξη διαδικτυακών εφαρμογών, τα οποία μπορούν να χρησιμοποιούνται από τον browser του χρήστη. Μόνη απαίτηση για την εκτέλεση του ExpressJS, είναι η ύπαρξη της JavaScript. Χρησιμοποιείται για την κατασκευή APIs τα οποία μπορούν να χρησιμοποιούνται από τους χρήστες ενός συστήματος. Ανήκει στην κατηγορία του ανοικτού λογισμικού.

**ReactJS**Logo

Description automatically generated

*A JavaScript library for building user interfaces*

[reactjs.com](https://reactjs.com/)

Η ReactJS είναι ένα library το οποίο χρησιμοποιείται για την κατασκευή front-end υλοποιήσεων, με τη χρήση της γλώσσας προγραμματισμού JavaScript. Ανήκει στην κατηγορία του ανοικτού λογισμικού.

**Hyperledger Fabric**

*Enteprise grade permissioned distributed ledger platform that offers*Text

Description automatically generated with medium confidence

*modularity and versatility for a broad set of industry use cases*

[hyperledger.org](https://hyperledger.org/use/fabric)

Το Hyperledger Fabric είναι ένα blockchain framework το οποίο αναπτύσσεται σύμφωνα με την έννοια και τη λογική της αρθρωτής αρχιτεκτονικής και στόχο έχει την παροχή ενός εργαλείου για την ανάπτυξη blockchain συστημάτων τα οποία μπορούν και ικανοποιούν τις απαιτήσεις που μπορεί να υπάρξουν από πλειάδα επιχειρηματικών τομέων τα οποία έχουν την ανάγκη για την ύπαρξη ενός ιδιωτικού blockchain δικτύου. Ανήκει στην κατηγορία του ανοικτού λογισμικού.

**RabbitMQ Message Broker**

*RabbitMQ is the most widely deployed open source message broker*Icon

Description automatically generated

[rabbitmq.com](https://rabbitmq.com)

Το RabbitMQ είναι ένα από τα πιο διαδεδομένα message brokers. Χρησιμοποιείται για την ομαλή και αποτελεσματική διαχείριση της πληροφορίας που διακινείται σε ένα σύστημα και τα διάφορα μέρη από τα οποία αυτό αποτελείται. Ανήκει στην κατηγορία του ανοικτού λογισμικού.

**Hashicorp Vault**Shape

Description automatically generated with low confidence

*Manage Secrets & Protect Sensitive Data*

[vaultproject.io](https://vaultproject.io)

Το Hashicorp Vault είναι ένα εργαλείο για τη διαχείριση της ευαίσθητης πληροφορίας που υπάρχει και διακινείται σε ένα περιβάλλον στο οποίο η εμπιστοσύνη δεν αποτελεί προϋπόθεση. Χρησιμοποιείται για την αποθήκευση και ανάκτηση ευαίσθητης πληροφορίας καθώς και για ευαίσθητες ενέργειες (π.χ. κρυπτογραφικές πράξεις). Ανήκει στην κατηγορία του ανοικτού λογισμικού, με την εταιρία που το υποστηρίζει να παρέχει και επιχειρηματικές λύσεις.

**Uvicorn**

*Uvicorn is a lightning-fast ASGI server implementation,*

*using uvloop and httptools*

[uvicorn.org](https://uvicorn.org)

Το Uvicorn ανήκει στην κατηγορία των web server εργαλείων. Στηρίζεται στο ASGI. Ανήκει στην κατηγορία του ανοικτού λογισμικού.

**FastAPI**

*FastAPI framework, high performance, easy to learn,*

*fast to code, ready for production*

[fastapi.tiangolo.com](https://fastapi.tiangolo.com/)

Το FastAPI είναι ένα γρήγορo, υψηλών επιδόσεων, web framework για την κατασκευή APIs μέσω της Python 3.6+. Ανήκει στην κατηγορία του ανοικτού λογισμικού

3.2.1 Λογισμικό Εφαρμογής Χρήστη (Client Application)

**Χαρακτηριστικά ανάπτυξης**

Για την ανάπτυξη των διαφόρων διεπαφών χρησιμοποιήθηκε η ReactJS library, ενώ για την επικοινωνία του χρήστη με το σύστημα, χρησιμοποιείται τόσο η ReactJS όσο και η NodeJS. Υπάρχουν σημεία του συστήματος τα οποία απαιτούν, για τη δημιουργία των αιτημάτων και την επεξεργασία τους, τεχνολογίες, μετασχηματισμούς και λειτουργίες ανώτερες από αυτές που προσφέρονται από την ReactJS. Με την συνεργασία των δύο αυτών τεχνολογιών (ReactJS, NodeJS) το σύστημα μπορεί και ικανοποιεί όλους τους περιορισμούς που έχουν τεθεί.

Προκειμένου να συνδυαστούν όλες αυτές οι τεχνολογίες σε ένα σημείο, να συνυπάρχουν και να συνεργάζονται, ώστε να λειτουργούν ως μία ολότητα χωρίς να διαφαίνονται τα διαφορετικά δομικά μέρη από τα οποία η τελική οντότητα αποτελείται, χρησιμοποιείται η τεχνολογία του Electron framework. Μέσω αυτού του framework δημιουργείται ένα σημείο πρόσβασης το οποίο μπορεί να εκτελεί κάθε ανθρώπινη οντότητα στον προσωπικό της υπολογιστή ώστε να επικοινωνεί με το σύστημα.

Shape

Description automatically generated with medium confidence

**Εικόνα 9: ΠΑΡΟΥΣΙΑΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΧΡΗΣΤΗ**

Λογισμικό διασύνδεσης Χρήστη – Σύστημα (Backend API)

**Χαρακτηριστικά ανάπτυξης**

Η οντότητα, έχει αναπτυχθεί με χρήση της NodeJS πλατφόρμας για τη λειτουργικότητά της, ενώ το API το οποίο είναι διαθέσιμο στους χρήστες του συστήματος αναπτύχθηκε με το ExpressJS framework, μέσω του οποίου:

* η δημιουργία εισιτηρίων
* οι ταυτοποιήσεις των χρηστών του συστήματος
* η διασύνδεση των χρηστών με το βασικό Blockchain υποσύστημα, Proxy BC

υλοποιούνται.

Επιπλέον των προηγούμενων και για την ομαλοποίηση των αιτημάτων, χρησιμοποιείται το RabbitMQ message broker, μέσω του οποίου καθίσταται δυνατή η δημιουργία και η διατήρηση σταθερής και διαχειρίσιμης ροής αιτημάτων δημιουργίας εισιτηρίων για την επικοινωνία με το Proxy BC.

3.2.4 Blockchain Συστήματα

**Χαρακτηριστικά ανάπτυξης**

Για την ανάπτυξη των Blockchains, τόσο για το κύριο, όσο και για όλα τα επιμέρους επιχειρηματικά Blockchains, χρησιμοποιήθηκε το **Hyperledger Fabric**.

Η συγκεκριμένη υλοποίηση βασίζεται στην έννοια της *αρθρωτής αρχιτεκτονικής* (modular) και υποστηρίζει την ανάπτυξη προϊόντων και λύσεων που οι ανάγκες τους μπορούν να καλυφθούν πλήρως από τα χαρακτηριστικά και τον τρόπο λειτουργίας των Blockchain συστημάτων.

Η ανάγκη της παρούσας υλοποίησης για τη δημιουργία ενός ιδιωτικού και κατανεμημένου (private & distributed) δικτύου, στο οποίο πολλές οντότητες μπορούν να συμμετέχουν και να διαθέτουν κρίσιμες και ευαίσθητες πληροφορίες στους χρήστες του συστήματος, ικανοποιείται πλήρως από το προαναφερθέν framework και η υλοποίηση του απαραίτητου Private Blockchain, μέσω του οποίου καλύπτονται όλες οι προδιαγραφές, είναι δυνατή.

Δηλαδή, δίνεται η δυνατότητα για την ανάπτυξη ενός Blockchain, το οποίο ανήκει στην κατηγορία των Private Blockchains, όπου δεν υφίσταται κεντρικός κυρίαρχος (Central Authority), αλλά όλα τα διαφορετικά εμπλεκόμενα μέρη, τα οποία διαθέτουν από έναν κόμβο στο Blockchain περιβάλλον, το αναπτύσσουν, το συντηρούν και το μετασχηματίζουν μέσα από τις ενέργειες τους και, όταν αυτό απαιτείται, υπό την ύπαρξη της πλειοψηφίας.

**Συνολικά, μέσω του Hyperledger Fabric και της Blockchain τεχνολογίας**, το δίκτυο μέσα από το οποίο συνεργάζονται τα διάφορα εμπλεκόμενα μέρη, υποστηρίζει την έννοιας της:

* ***Διαφάνειας*** (Transparent), αφού κάθε συναλλαγή (Transaction) που υλοποιείται, μπορεί να είναι προσβάσιμη από τα υπόλοιπα ενδιαφερόμενα/εμπλεκόμενα μέρη
* ***Ασφάλειας* και *Αμεταβλητότητας*** (Secure & Immutable), αφού επιτρέπει τη διαχείριση ήδη κρυπτογραφημένης πληροφορίας, ενώ τα Blocks τα οποία δημιουργούνται και μέσα στα οποία αποθηκεύονται οι συναλλαγές που πραγματοποιούνται, δε μπορούν να αλλοιωθούν ή να αμφισβητηθούν. Αλλοίωση στα Blocks του δικτύου, άμεσα σημαίνει ότι υφίσταται εμπλεκόμενο μέρος το οποίο δεν είναι έντιμο. Μεταποίηση ενός Block μπορεί να συμβεί μόνο με τη σύμφωνη γνώμη όλων των εμπλεκόμενων μερών
* ***Διαθεσιμότητας* (Available)**, αφού ακόμη και εάν ένα εμπλεκόμενο μέρος πάψει να υφίσταται ή σταματήσει τη λειτουργία του λόγω σφάλματος, τα υπόλοιπα μέρη μπορούν να καλύψουν αυτήν την, προσωρινή ή μη, απώλεια
* ***Αντιγραψιμότητας* (Replicable)**, αφού κάθε ενέργεια που συμβαίνει στο δίκτυο, καταγράφεται από όλα τα συμμετέχοντα μέρη, ή καταγράφεται από τα μέρη που αφορά και μπορεί να επηρεάσει

3.2.5 Smart Contracts

**Χαρακτηριστικά ανάπτυξης**

3.2.6 Λογισμικό Διασύνδεσης Blockchain Συστημάτων (Inter-Blockchain API)

**Χαρακτηριστικά ανάπτυξης**

Για την ανάπτυξη του συγκεκριμένου λογισμικού, χρησιμοποιήθηκε η NodeJS.

Για τη δημιουργία και τη διατήρηση σταθερής ροής αιτημάτων προς τα Blockchain συστήματα, χρησιμοποιείται το RabbitMQ message broker.

3.2.7 Λογισμικό Διασύνδεσης Συστήματος – Βάσεων Δεδομένων (DBC API)

**Χαρακτηριστικά ανάπτυξης**

Για την ανάπτυξη του λογισμικού χρησιμοποιήθηκε η Python 3.

Η υλοποίηση του διακομιστή (server) με τον οποίο επικοινωνούν τα Domain Blockchains αναπτύχθηκε με το Uvicorn, ενώ για τη δημιουργία του API, χρησιμοποιήθηκε το FastAPI framework.

3.2.8 Hashicorp Vault & Κρυπτογραφικές Πράξεις

Η ασφάλεια των ευαίσθητων δεδομένων που λαμβάνονται, διακινούνται και μεταβιβάζονται στις οντότητες του συστήματος, αποτελεί πρώτο και κύριο μέλημα της υλοποίησης.

Όπως κάθε κρίσιμη ενέργεια λαμβάνεται σοβαρά υπόψιν από το σύστημα (καταγραφή, απαίτηση πλειοψηφίας, κατά περιπτώσεις, για την εκτέλεση της κ.λπ.), έτσι και κάθε ευαίσθητη πληροφορία, πρέπει να λαμβάνει την απαραίτητη προσοχή από αυτό.

Για την εξασφάλιση ότι τα ευαίσθητα δεδομένα, στην περίπτωση που ληφθούν από οντότητα η οποία δεν έχει τα απαιτούμενα τεκμήρια, θα είναι άχρηστα, χρησιμοποιείται το Hashicorp Vault και το ειδικά ανεπτυγμένο, για τις απαιτήσεις της παρούσας μελέτης και υλοποίησης, πρόσθετου (Plugin) το οποίο εκτελεί τις απαραίτητες κρυπτογραφικές πράξεις, οι οποίες βασίζονται στην τεχνολογία του Attribute-Based Encryption (ABE).

Το κρυπτογραφικό κλειδί με το οποίο κρυπτογραφήθηκαν τα δεδομένα που διακινούνται στο σύστημα προστατεύεται, με τη μέθοδο της ABE κρυπτογράφησης, σε δύο (νοητά) επίπεδα βάσει των ιδιαίτερων χαρακτηριστικών (Attributes) των οντοτήτων που λαμβάνουν μέρος σε αυτό.

Πιο συγκεκριμένα, ένα επίπεδο συστήματος και ένα επίπεδο οντότητας χρήστη (βλ. [Διαφύλαξη Δεδομένων Συστήματος - Κρυπτογραφικές Ενέργειες](#_heading=h.38czs75)).

Κατά την μερική αποκρυπτογράφηση, τα δεδομένα αυτά αντιστοιχίζονται με τον χρήστη ο οποίος ξεκίνησε τη διαδικασία.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα 18 ΠΑΡΟΥΣΙΑΣΗ ΕΠΙΠΕΔΩΝ ΑΠΟΚΡΥΠΤΟΓΡΑΦΗΣΗΣ ΔΕΔΟΜΕΝΩΝ (ΕΠΙΠΕΔΟ ΣΥΣΤΗΜΑΤΟΣ & ΧΡΗΣΤΗ)

Κάθε οργανισμός ο οποίος είναι μέρος του συστήματος, λαμβάνει πρόσβαση σε ένα Attribute τύπου Σύστημα (System). Το Attribute αυτό το χρησιμοποιεί ο εμπλεκόμενος οργανισμός για την ολοκλήρωση του 1ου σταδίου.

Αφού το πρώτο στάδιο ολοκληρωθεί και παραληφθούν τα *μερικώς αποκρυπτογραφημένα δεδομένα*, κάθε χρήστης χρησιμοποιεί τα κατάλληλα κλειδιά, τα οποία αντιστοιχούν στα χαρακτηριστικά (Attributes) που κατέχει και ολοκληρώνει τη διαδικασία της αποκρυπτογράφησης.

Κάθε οργανισμός είναι υπεύθυνος για τη δημιουργία αυτών των κλειδιών για τον χρήστη, με την παραχώρηση σε αυτόν των απαραίτητων χαρακτηριστικών και αδειών μέσω του πρόσθετου (Plugin).

## Αρχιτεκτονική συστήματος

Diagram

Description automatically generated

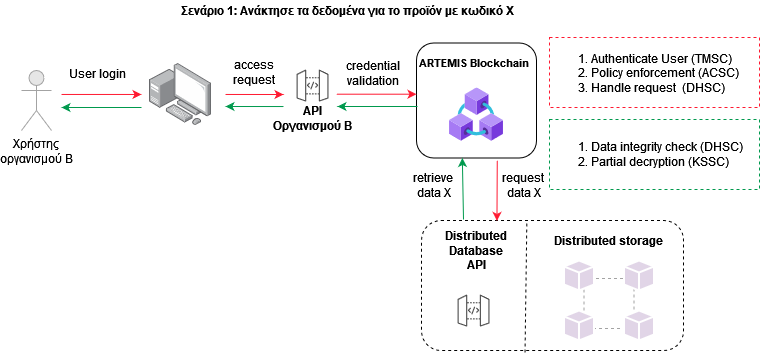
## Σενάρια χρήσης

Προτού γίνει αποτύπωση των διαφόρων σεναρίων χρήσης που μπορούν να αναπτυχθούν στο σύστημα, θα αναλυθούν οι λειτουργικές απαιτήσεις που υπάρχουν.

Στον παρακάτω πίνακα αναπτύσσονται οι Λειτουργικές απαιτήσεις του Συστήματος. Οι απαιτήσεις αυτές θα αναπτυχθούν σαν ολοκληρωμένες λειτουργίες και θα αναφέρονται ως *OpReqXX*. Για κάθε απαίτηση, θα αναφέρεται ο ρόλος του συστήματος (Backend API, Inter-Blockchain API κ.ο.κ.) ή ο Ανθρώπινος Ρόλος (Admin, CA-Admin, Auditor, Client) που θα μπορεί να εκτελέσει τη λειτουργία.

|  |  |  |  |
| --- | --- | --- | --- |
| **Απαίτηση** | **Σχέση** | **Απαίτηση** | **Προϋπόθεση** |
| **ADMIN** | | | |
| **OpReq00** | Certificate Authority | Να υπάρχει η δυνατότητα δημιουργίας λογαριασμών (CA-Admin, Auditor, Client) για κάθε εμπλεκόμενο |  |
| **OpReq01** | Hashicorp’s Vault | Να μπορεί να παρέχει προσωπικά γνωρίσματα (Attributes) στους χρήστες (Clients) του συστήματος και στο Hashicorp Vault ABE Plugin, προκειμένου οι χρήστες να μπορούν να αποκρυπτογραφούν την ήδη κρυπτογραφημένη, από τον κάθε εμπλεκόμενο, πληροφορία | Ο οργανισμός να διαθέτει το αντίστοιχο ABE Key το οποίο σχετίζεται με το attribute |
| **CA-ADMIN** | | | |
| **OpReq02** | Backend API - Proxy BC | Να είναι δυνατή η ανανέωση των στοιχείων ενός εμπλεκόμενου (Certificate, CRL, ACL) | * Ο οργανισμός να μην έχει ανακληθεί * Τα Certificate, CRL να είναι μεταγενέστερα των υπαρχόντων |
| **OpReq03** | Backend API - Proxy BC | Να μπορεί να λαμβάνει μέρος σε ψηφοφορίες (για την εισαγωγή/αφαίρεση εμπλεκόμενου, για την έγκριση άδειας σε έναν Ελεγκτή ώστε να λάβει είσοδο (access) στις καταγραφές (Logs) του συστήματος) | Ο οργανισμός να μην έχει ανακληθεί |
| **OpReq04** | Backend API - Proxy BC | Να μπορεί να δημιουργεί ψηφοφορίες για την εισαγωγή ενός νέου εμπλεκόμενου, στο σύστημα | Ο οργανισμός να μην έχει ανακληθεί |
| **OpReq05** | Backend API - Proxy BC | Να μπορεί να δημιουργεί ψηφοφορίες για την αφαίρεση ενός εμπλεκόμενου από το σύστημα | Ο οργανισμός να μην έχει ανακληθεί |
| **AUDITOR** | | | |
| **OpReq06** | Backend API - Proxy BC | Να μπορεί να λάβει πρόσβαση καθώς και να ελέγξει τις αιτήσεις του για είσοδο στις καταγραφές (Logs) του συστήματος | Ο οργανισμός να μην έχει ανακληθεί |
| **OpReq07** | Backend API - Proxy BC | Να μπορεί να βλέπει τις καταγραφές με την πρόσβαση που του έχει δοθεί από μία αίτηση για Έλεγχο (Audit) | * Ο οργανισμός να μην έχει ανακληθεί * Η πρόσβαση να μην έχει ανακληθεί |
| **OpReq08** | Backend API - Proxy BC | Να μπορεί να ζητάει πρόσβαση, είτε συνολικά στο σύστημα (Proxy BC), από όλα τα εμπλεκόμενα μέρη, είτε ανά επιχειρηματικό τομέα (Domain BC), στις καταγραφές (Logs) του συστήματος | Ο οργανισμός να μην έχει ανακληθεί |
| **CLIENT** | | | |
| **OpReq09** | Backend API - Inter-Blockchain API - Proxy BC | Να μπορεί να δημιουργεί μία αίτηση (Request) για τη λήψη πληροφοριών από τη Βάση Δεδομένων ενός ή περισσότερων εμπλεκόμενων που ανήκουν στον ίδιο ή διαφορετικό επιχειρηματικό τομέα με αυτόν | Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **OpReq10** | Hashicorp’s Vault | Να μπορεί να αποκρυπτογραφεί (decrypt) τα δεδομένα που ελήφθησαν από τη βάση δεδομένων (ή τις βάσεις δεδομένων) των εμπλεκόμενων και τα οποία αντιστοιχούν στην αίτησή του | Ο χρήστης να διαθέτει τα απαραίτητα ABE attributes με τα οποία έχει κρυπτογραφηθεί η πληροφορία |
| **OpReq11** | Backend API - Proxy BC | Να μπορεί να διαχειρίζεται τις αιτήσεις του (Requests), καθώς και να μπορεί να ελέγξει την πορεία ολοκλήρωσης τους (Ολοκληρωμένες, Επιτυχημένες) | Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **OpReq12** | Certificate Authority | Να έχει τη δυνατότητα, σε περίπτωση που αυτό απαιτείται από μία αίτηση, να συνδυάζει Ρόλους που έχει λάβει μέσω ταυτοποίησης στους διάφορους εμπλεκόμενους φορείς | * Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί * Το GID να είναι κοινό για όλες τις οντότητες του χρήστη |
| **OpReq13** | Backend API - Proxy BC | Να μπορεί να έχει επισκόπηση επί των ενεργών του ρόλων | Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **INTER-BLOCKCHAIN API** | | | |
| **OpReq14** | Proxy BC | Να ελέγχει σε τακτά χρονικά διαστήματα, ή έπειτα από τη λήψη ενός συμβάντος (event) από το Blockchain, την πορεία των ψηφοφοριών |  |
| **OpReq15** | Proxy BC | Να ενημερώνει το Blockchain για τη λήξη μίας ψηφοφορίας |  |
| **OpReq16** | Proxy BC | Να εισάγει αυτόματα νέες ψηφοφορίες στη λίστα με τις ψηφοφορίες που πρέπει να ελέγχει |  |
| **OpReq17** | Proxy BC – Domain BC | Να μεταβιβάζει συμβάντα (events) που έλαβε από το Proxy Blockchain στο κατάλληλο Domain Blockchain και αντίστροφα | Ο οργανισμός να συμμετέχει στο Inter-Blockchain API |
| **BACKEND API** | | | |
| **OpReq18** | Client Application – Proxy BC | Να μεταβιβάζει τις αιτήσεις από τους χρήστες (CA-Admins, Auditors, Clients) του συστήματος, στο Proxy Blockchain και αντίστροφα |  |
| **OpReq19** | Client Application | Να δημιουργεί τεκμήριο μέσω του οποίου ένας Client θα μπορεί να χρησιμοποιήσει το Backend-API για τη μεταφορά πληροφορίας προς το Proxy Blockchain | Ο οργανισμός να συμμετέχει στο Backend API |
| **PROXY BLOCKCHAIN** | | | |
| **OpReq20** | Client Application – Backend API – Inter-Blockchain API | Να δέχεται αιτήματα από χρήστη του συστήματος, μέσω του Backend API, να επικυρώνει τον χρήστη, να συγκεντρώνει τα χαρακτηριστικά και στοιχεία (Ρόλοι και Προσωρινοί ρόλοι) που τον αφορούν και αναφέρονται στις πολιτικές των εμπλεκόμενων, να αποθηκεύει το αίτημα στις εγγραφές του συστήματος και να προωθεί το αίτημα, μαζί με τα χαρακτηριστικά του που συγκεντρώθηκαν από τις πολιτικές, στο Inter-Blockchain API | * Ο οργανισμός να συμμετέχει στα Backend / Inter-Blockchain APIs * Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **OpReq21** | Inter-Blockchain API | Να καταχωρεί στις καταγραφές του συστήματος, τα αιτήματα που έχουν δημιουργηθεί από τους χρήστες. Όταν η διαδικασία του αιτήματος ολοκληρώνεται, να λαμβάνει από το Inter-Blockchain API τις πληροφορίες και δεδομένα που αφορούν το αίτημα και να ενημερώνει τις καταγραφές για την ολοκλήρωσή του. |  |
| **OpReq22** |  | Να λαμβάνει αίτημα για τη δημιουργία ψηφοφορίας, να ελέγχει εάν υπάρχει (ίδια) υφιστάμενη ψηφοφορία που εκτελείται, να δημιουργεί τα ψηφοδέλτια για κάθε εμπλεκόμενο που η ψηφοφορία αφορά | Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **OpReq23** | Client Application – Backend API | Να επιτρέπει την ανανέωση των εξατομικευμένων πολιτικών | Ο οργανισμός, ή ο χρήστης να μην έχει ανακληθεί |
| **DOMAIN BLOCKCHAIN** | | | |
| **OpReq24** | Hashicorp’s Vault | Να μπορεί να εκτελεί τη μερική αποκρυπτογράφηση των δεδομένων τα οποία ελήφθησαν από την επικοινωνία του Συστήματος με τις βάσεις δεδομένων των εμπλεκόμενων μερών | Ο οργανισμός να μπορεί να κάνει χρήση του *SA* ABE Attribute |
| **OpReq25** | Proxy BC- Inter-Blockchain API – DBC API | Να λαμβάνει αιτήματα (Requests) από το Proxy BC, διαμέσω του Inter-Blockchain API, και να αξιώνει πρόσβαση στη λήψη πληροφοριών και δεδομένων από τα εμπλεκόμενα μέρη, σύμφωνα με τις πολιτικές που έχουν θεσπισθεί από αυτά. Να επιστρέφει τα αποτελέσματα στο Proxy BC, διαμέσω του Inter-Blockchain API |  |

Τα σενάρια χρήσης ….



# Κατανεμημένη βάση δεδομένων

## Εισαγωγή

## Μηχανισμοί κατανεμημένης αποθήκευσης δεδομένων

## Μηχανισμοί commitment για σειρές δεδομένων

## Σύστημα αποθήκευσης δεδομένων

Προκειμένου να υπάρχει αλληλεπίδραση μεταξύ των οντοτήτων καθώς και πρόσβαση στα δεδομένα που απαιτούνται για την λειτουργία του συστήματος ARTEMIS, έχουν σχεδιαστεί τόσο οι βάσεις δεδομένων για κάθε οντότητα, καθώς επίσης και οι κατάλληλες διεπαφές προγραμματισμού (APIs). Η χρήση των διεπαφών διευκολύνει τόσο την καταχώρηση των δεδομένων στις βάσεις κάθε οντότητας όσο και την λήψη δεδομένων από αυτές. Στα επόμενα υποκεφάλαια θα αναλυθούν οι δομές αποθήκευσης δεδομένων καθώς και η αναλυτική περιγραφή των διεπαφών.

* + 1. Data layer

Ο πάροχος τηλεματικής θα πρέπει να έχει αποθηκευμένο το πελατολόγιο. Η αποθήκευση των πελατών γίνεται σε έναν πίνακα με ενδεικτικές τιμές:

* Ένα μοναδικό αναγνωριστικό για κάθε πελάτη (client\_id)
* Το όνομα του πελάτη (client\_name)

Σε κάθε πελάτη αποδίδεται ένα πλήθος συσκευών που ονομάζονται πύλες (gateways) οι οποίες είναι υπεύθυνες για την καταγραφή διαφόρων δεδομένων τηλεματικής. Ενδεικτικά τα δεδομένα που περιέχει το κάθε gateway είναι:

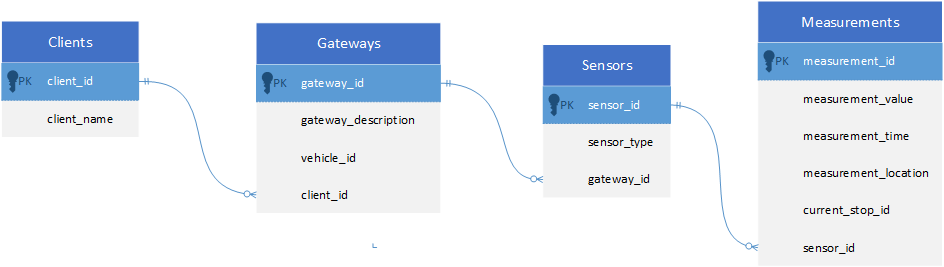
* Ένα μοναδικό αναγνωριστικό για κάθε πύλη (gateway\_id)
* Μια σύντομη περιγραφή για την ύπαρξη της πύλης (gateway\_description)
* Το αναγνωριστικό που οχήματος του πελάτη στο οποίο έχει εγκατασταθεί η κάθε πύλη (vehicle\_id)
* Το αναγνωριστικό του πελάτη στον οποίο έχει αποδοθεί η κάθε πύλη (client\_id)

Κάθε ένα gateway περιέχει ένα πλήθος αισθητήρων (sensors) που είναι υπεύθυνοι για τη καταγραφή μετρήσεων διαφόρων παραγόντων. Οι αισθητήρες έχουν ενδεικτικά τρεις παραμέτρους:

* Ένα μοναδικό αναγνωριστικό για κάθε αισθητήρα (sensor\_id)
* Τον τύπο των μετρήσεων που καταγράφει (sensor\_type)
* Το αναγνωριστικό της πύλης που είναι υπεύθυνη για τον συγκεκριμένο αισθητήρα (gateway\_id)

Ο κάθε αισθητήρας καταγράφει πολλαπλές μετρήσεις (measurements). Οι μετρήσεις έχουν τις παρακάτω παραμέτρους:

* Ένα μοναδικό αναγνωριστικό για κάθε τύπο μέτρησης (measurement\_id)
* Την τιμή της μέτρησης (measurement\_value)
* Την χρονική στιγμή που καταγράφηκε η κάθε μέτρηση (measurement\_time)
* Την γεωγραφική θέση που καταγράφηκε κάθε μέτρηση (measurement\_location)
* Ένα πεδίο που λαμβάνει τιμή όταν το όχημα σταματήσει μέσα στα γεωγραφικά όρια μίας δεδομένης στάσης που έχει καθοριστεί για το εκάστοτε δρομολόγιο (current\_stop\_id)
* Το αναγνωριστικό του αισθητήρα που είναι υπεύθυνο για τις συγκεκριμένες μετρήσεις (sensor\_id)



Η δεύτερη οντότητα για την οποία θα αναλυθεί η δομή δεδομένων αφορά κάθε πελάτη του παρόχου υπηρεσιών τηλεματικής. Κάθε πελάτης έχει με τη σειρά του ένα πλήθος πελατών (customers) που αναλαμβάνει να τους προμηθεύσει με προϊόντα. Ο πίνακας με τα χαρακτηριστικά του κάθε ένα από αυτούς τους πελάτες είναι:

* Ένα μοναδικό αναγνωριστικό για κάθε έναν πελάτη του προμηθευτή (customer\_id)
* Το όνομα του πελάτη (customer\_name)

Κάθε πελάτης του προμηθευτή μπορεί να περιέχει παραπάνω από ένα σημείο παραλαβής προϊόντων (stops). Ο πίνακας «stops» περιέχει:

* Το αναγνωριστικό πελάτη του προμηθευτή (customer\_id)
* Ένα μοναδικό αναγνωριστικό για κάθε σημείο παραλαβής που έχει ο πελάτης (stop\_id)
* Την γεωγραφική τοποθεσία κάθε σημείου παραλαβής (location)

Επίσης, κάθε πελάτης του παρόχου υπηρεσιών τηλεματικής, κατέχει ένα πλήθος οχημάτων που αναλαμβάνουν την μεταφορά προϊόντων. Τα οχήματα είναι καταγεγραμμένα σε έναν πίνακα που περιέχει ένα μοναδικό αναγνωριστικό για κάθε όχημα (vehicle\_id).

Το κάθε όχημα αναλαμβάνει να κάνει μία αποστολή (shipment). Ενδεικτικά η κάθε αποστολή αποτελείται από τις παρακάτω τιμές:

* Ένα μοναδικό αναγνωριστικό για κάθε αποστολή (shipment\_id)
* Την ημερομηνία εκτέλεσης της κάθε αποστολής (shipment\_date)
* Το αναγνωριστικό του οχήματος που αναλαμβάνει την διεξαγωγή της κάθε αποστολής (vehicle\_id)

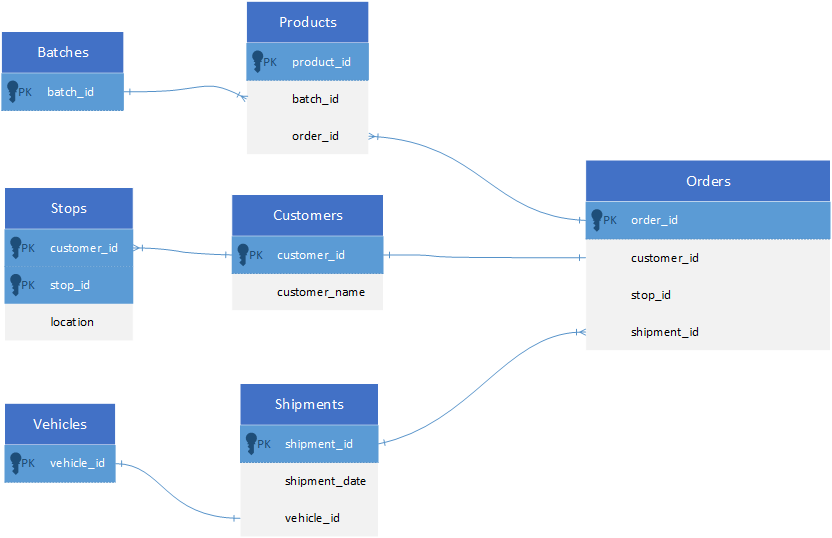
Κάθε αποστολή περιέχει πολλαπλές παραγγελίες (orders) που πρέπει να παραδοθούν σε πελάτες του προμηθευτή. Κάθε παραγγελία περιέχει τα εξής δεδομένα:

* Ένα μοναδικό αναγνωριστικό παραγγελίας (order\_id)
* Το αναγνωριστικό πελάτη για τον οποίο προορίζεται η παραγγελία (customer\_id)
* Το σημείο παραλαβής της παραγγελίας (stop\_id)
* Το αναγνωριστικό της αποστολής που περιέχει την εκάστοτε παραγγελία (shipment\_id)

Κάθε μία από τις παραγγελίες περιέχει πολλαπλά προϊόντα (products). Κάθε προϊόν περιέχει τα παρακάτω δεδομένα:

* Ένα μοναδικό αναγνωριστικό προϊόντος (product\_id)
* Το αναγνωριστικό της παρτίδας από την οποία προέρχεται το προϊόν (batch\_id)
* Το αναγνωριστικό της παραγγελίας που περιέχει το εκάστοτε προϊόν (order\_id)

Επιπροσθέτως, ο πελάτης του παρόχου υπηρεσιών τηλεματικής, καταγράφει σε έναν ξεχωριστό πίνακα τις παρτίδες (batch\_id) για τα προϊόντα του.



Για την λειτουργία του συστήματος θα πρέπει να υπάρχει μια ελεγχόμενη πρόσβαση στα δεδομένα μεταξύ των οντοτήτων. Ο διαμοιρασμός δεδομένων μεταξύ των οντοτήτων γίνεται με την βοήθεια μίας κοινής βάσης δεδομένων η οποία είναι αποκεντρωμένη. Δεδομένου ότι ο πάροχος υπηρεσιών τηλεματικής πρέπει να γνωρίζει:

* Το αναγνωριστικό του κάθε οχήματος το οποίο αναλαμβάνει να εκτελέσει μία αποστολή
* Το αναγνωριστικό της κάθε παραγγελίας προκειμένου να συσχετιστούν οι μετρήσεις
* Το αναγνωριστικό του κάθε πελάτη του προμηθευτή καθώς και τα αναγνωριστικά των σημείων παραλαβής για κάθε έναν από αυτούς τους πελάτες
* Την ημερομηνία για την οποία είναι προγραμματισμένη κάθε αποστολή

Ο πελάτης του παρόχου τηλεματικής θα πρέπει να διαμοιράσει στην κοινόχρηστη βάση:

* Την ημερομηνία αποστολής (shipment\_date)
* Το αναγνωριστικό της παραγγελίας για την συγκεκριμένη ημερομηνία (order\_id)
* Το αναγνωριστικό πελάτη του προμηθευτή (customer\_id)
* Το αναγνωριστικό του σημείου παραλαβής για τον συγκεκριμένο πελάτη (stop\_id)
* Το αναγνωριστικό του οχήματος που έχει αναλάβει την αποστολή (vehicle\_id)

Ο πάροχος υπηρεσιών τηλεματικής μπορεί να διαμοιράσει τις μετρήσεις του στην κοινή αποκεντρωμένη βάση δεδομένων προκειμένου να είναι προσβάσιμες για τις υπόλοιπες οντότητες. Λόγω του ότι τα δεδομένα είναι προσβάσιμα από όλους όσους απαρτίζουν την αποκεντρωμένη βάση δεδομένων, θα πρέπει να υπάρχει περιορισμένη πρόσβασή στα δεδομένα που μπορεί να έχει κάθε οντότητα (με την χρήση κρυπτογραφίας). Με αυτό τον τρόπο, κάθε οντότητα θα έχει πρόσβαση μόνο στα δεδομένα που την αφορούν. Έτσι, ο πελάτης του παρόχου τηλεματικής θα πρέπει να διαμοιράσει έναν πίνακα με τα παρακάτω δεδομένα:

* Το αναγνωριστικό της παραγγελίας (order\_id)
* Την ημερομηνία που θα εκτελεστεί η αποστολή (shipment\_date)
* Το αναγνωριστικό πελάτη του προμηθευτή σε κρυπτογραφημένη μορφή (enc\_customer\_id)
* Το σημείο παραλαβής σε κρυπτογραφημένη μορφή (enc\_stop\_id)
* Το όχημα που έχει αναλάβει την αποστολή (vehicle\_id)
* Το συμμετρικό κλειδί με το οποίο κρυπτογραφήθηκε το αναγνωριστικό του πελάτη (enc\_customer\_id) και το σημείο παραλαβής (enc\_stop\_id), σε κρυπτογραφημένη μορφή με την χρήση κρυπτογράφησης βάσει χαρακτηριστικών (Attribute Based Encryption). Ο πελάτης θα πρέπει να αποδώσει χαρακτηριστικά στον πάροχο τηλεματικής τέτοια ώστε να του επιτρέπουν την αποκρυπτογράφηση του συμμετρικού κλειδιού.

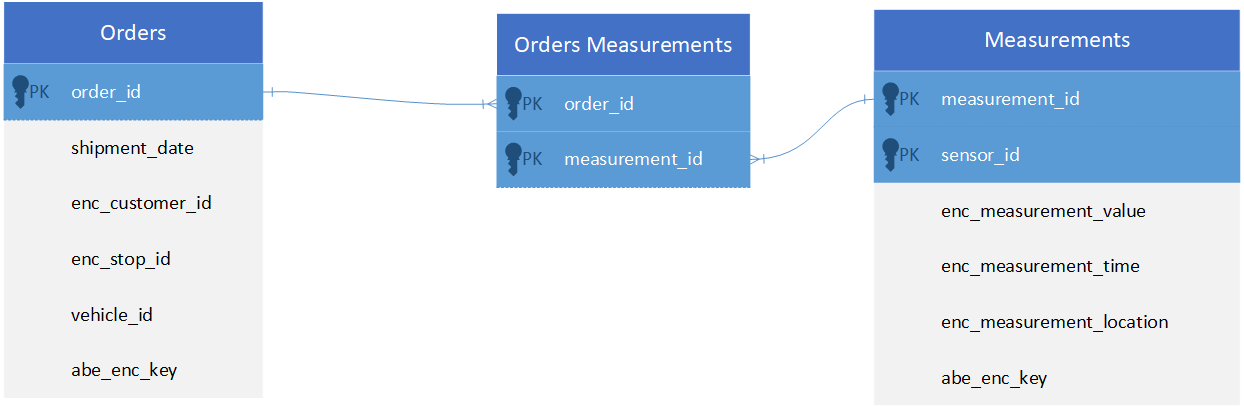
Έτσι, ο πάροχος υπηρεσιών τηλεματικής μπορεί να ελέγξει στην κοινόχρηστη βάση αν για κάποιο όχημα (vehicle\_id) από αυτά στα οποία έχουν εγκατασταθεί πύλες (gateway\_id), έχει αποστολή κάποια συγκεκριμένη ημερομηνία (shipment\_date). Αν υπάρχει, τότε θα αποκρυπτογραφήσει το συμμετρικό κλειδί με το κλειδί της κρυπτογράφησης βάσει χαρακτηριστικών (Αttribute Βased Εncryption key), προκειμένου να αποκρυπτογραφήσει με το συμμετρικο κλειδί, το αναγνωριστικό πελάτη του προμηθευτή (customer\_id) και το σημείο παραλαβής (stop\_id).

Κάθε παραγγελία διαθέτει πολλαπλές μετρήσεις, οπότε ο πάροχος υπηρεσιών τηλεματικής θα πρέπει να αποθηκεύει επίσης στη κοινόχρηστη βάση την σύνδεση μεταξύ των παραγγελιών και των μετρήσεων. Ουσιαστικά ο πίνακας που είναι υπεύθυνος για αυτή τη πληροφορία, θα περιέχει τα εξής δεδομένα:

* Το αναγνωριστικό της παραγγελίας (order\_id)
* Το αναγνωριστικό της κάθε μέτρησης για την παραγγελία (measurement\_id)

Τα δεδομένα της κάθε μέτρησης θα πρέπει να βρίσκονται και αυτά στη κοινόχρηστη βάση δεδομένων. Δηλαδή, για κάθε μέτρηση θα πρέπει να αποθηκεύονται σε έναν πίνακα τα παρακάτω δεδομένα:

* Το αναγνωριστικό της κάθε μέτρησης (measurement\_id)
* Το αναγνωριστικό του αισθητήρα που είναι υπεύθυνος για την μέτρηση (sensor\_id)
* Την τιμή της μέτρησης, σε κρυπτογραφημένη μορφή (enc\_measurement\_value)
* Την χρονική στιγμή που έγινε η μέτρηση, σε κρυπτογραφημένη μορφή (enc\_measurement\_time)
* Την γεωγραφική θέση στην οποία έγινε η μέτρηση, σε κρυπτογραφημένη μορφή (enc\_measurement\_location)
* Το συμμετρικό κλειδί με το οποίο κρυπτογραφήθηκε η τιμή της μέτρησης (enc\_measurement\_value), η χρονική στιγμή (enc\_measurement\_time) και η γεωγραφική θέση (enc\_measurement\_location) σε κρυπτογραφημένη μορφή με την χρήση κρυπτογράφησης βάσει χαρακτηριστικών (Attribute Based Encryption). Ο πάροχος τηλεματικών υπηρεσιών θα πρέπει να αποδώσει χαρακτηριστικά στους πελάτες του προμηθευτή τέτοια ώστε να του επιτρέπουν την αποκρυπτογράφηση του συμμετρικού κλειδιού για τα δεδομένα που τους αφορούν.



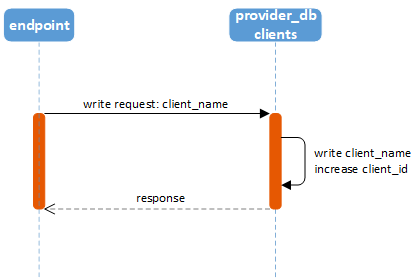
Η συνολική εικόνα των βάσεων δεδομένων απεικονίζεται στην παρακάτω εικόνα:



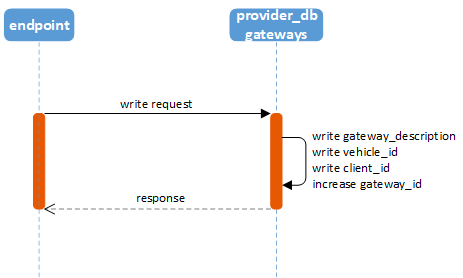
* + 1. API layer

Προκειμένου να αποθηκεύονται δεδομένα στις παραπάνω βάσεις δεδομένων για κάθε οντότητα, χρησιμοποιούνται Διεπαφές Προγραμματισμού Εφαρμογών (API).

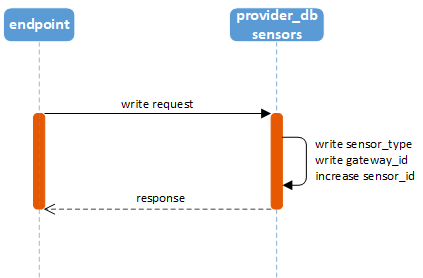
Αρχικά, ο πάροχος υπηρεσιών τηλεματικής, θα πρέπει να εγγράψει τους πελάτες του κάνοντας κλήση στο τελικό σημείο (endpoint) που είναι υπεύθυνο για την διαδικασία αυτή.



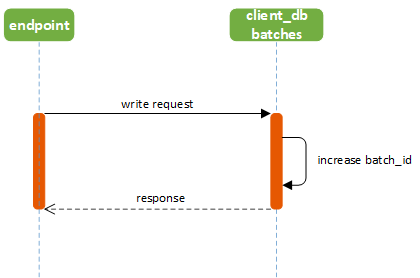
Προκειμένου να καταχωρήσει τις πληροφορίες για τις πύλες (gateways) θα πρέπει να καλέσει το τελικό σημείο (endpoint) που είναι υπεύθυνο για την εγγραφή του πίνακα «Gateways».



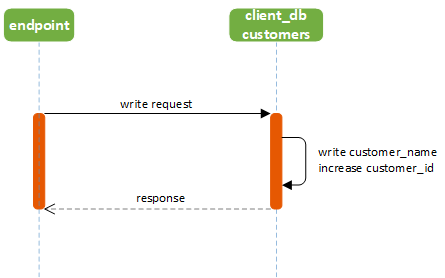
Προκειμένου να καταχωρηθούν τα χαρακτηριστικά του αισθητήρα, γίνεται κλήση στο τελικό σημείο και γίνεται αντίστοιχα η διαδικασία που περιγράφεται στο παρακάτω διάγραμμα ροής.



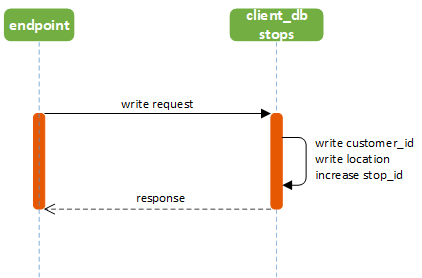
Ο πελάτης του παρόχου θα πρέπει να καταγράψει τις αρχικές πληροφορίες στους πίνακες της βάσης δεδομένων του με την κλήση μεθόδων σε δική του διεπαφή (API) που διαχειρίζεται την βάση του. Ενδεικτικά, η κλήση για την καταγραφή των παρτίδων (batches) γίνεται όπως παρακάτω.



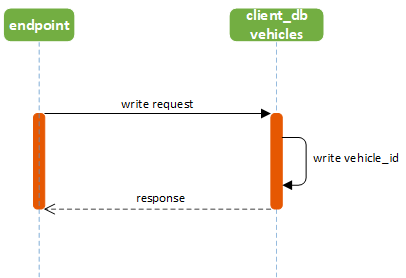
Επίσης για την εγγραφή των τελικών πελατών, γίνεται κλήση στο τελικό σημείο που είναι υπεύθυνο για την εγγραφή στοιχείων στον πίνακα «Customers».



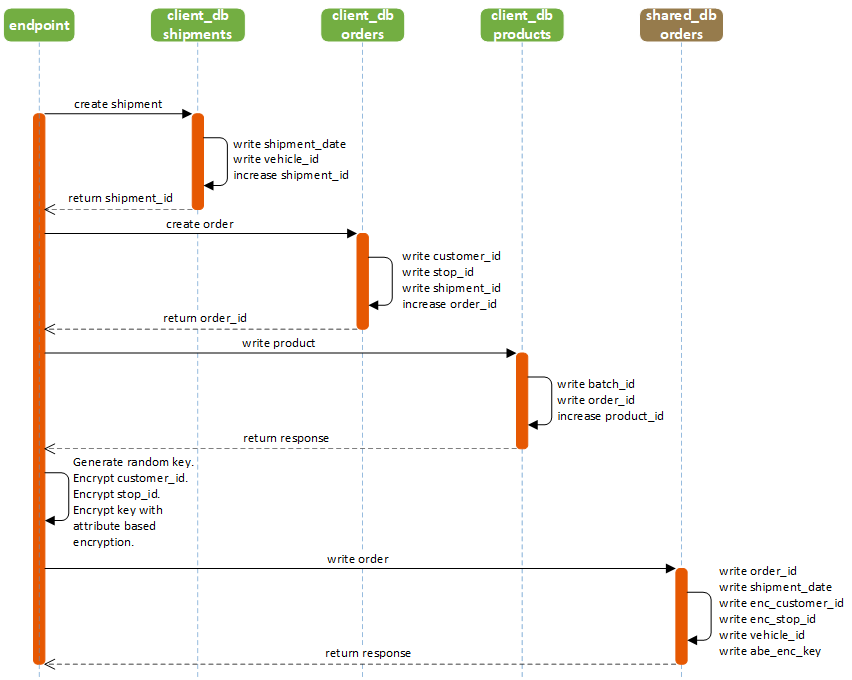
Για κάθε τελικό πελάτη, θα πρέπει να καταχωρηθούν τα σημεία παραλαβής καθώς και η γεωγραφικές τους τοποθεσίες.



Επίσης, ο πελάτης του παρόχου τηλεματικών υπηρεσιών, θα πρέπει να καταχωρήσει τα οχήματα που έχει στην κατοχή του και είναι υπεύθυνα για την αποστολή παραγγελιών.



Από τη στιγμή που υπάρχουν καταχωρημένα αρχικά δεδομένα στους πίνακες που αναφέραμε. Ο πελάτης του παρόχου μπορεί να δημιουργήσει μία παραγγελία προς αποστολή καλώντας το τελικό σημείο (endpoint) που πραγματοποιεί τις παρακάτω διαδικασίες.

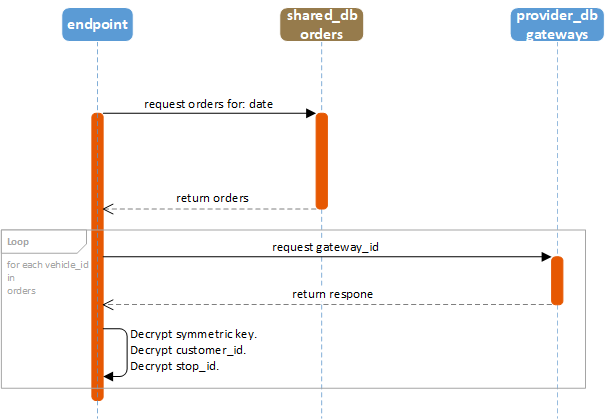


Όπως βλέπουμε στο παραπάνω διάγραμμα, η διαδικασία που ακολουθείται όταν κληθεί το τελικό σημείο για τη δημιουργία μίας παραγγελίας, είναι η εξής:

1. Δημιουργείται αποστολή και εισάγονται τα παρακάτω στον πίνακα «Shipments»:
   * Ημερομηνία αποστολής (shipment\_date)
   * Αναγνωριστικό του οχήματος που αναλαμβάνει την αποστολή (vehicle\_id)
   * Αυτόματη συμπλήρωση του αναγνωριστικού αποστολής (shipment\_id)
2. Επιστρέφεται η τιμή του αναγνωριστικού από τη δημιουργία της αποστολής
3. Δημιουργείται παραγγελία και εισάγονται τα παρακάτω στον πίνακα «Orders»:
   * Το αναγνωριστικό του τελικού πελάτη (customer\_id)
   * Το αναγνωριστικό του σημείου παραλαβής (stop\_id)
   * Το αναγνωριστικό αποστολής που επιστράφηκε ως απάντηση προηγουμένως (shipment\_id)
   * Αυτόματη συμπλήρωση του αναγνωριστικού παραγγελίας (order\_id)
4. Επιστρέφεται η τιμή του αναγνωριστικού από τη δημιουργία της παραγγελίας
5. Δημιουργείται εγγραφή προϊόντος με την εισαγωγή των παρακάτω δεδομένων στον πίνακα «Products»:
   * Το αναγνωριστικό της παρτίδας (batch\_id)
   * Το αναγνωριστικό της παραγγελίας (order\_id)
   * Αυτόματη συμπλήρωση του αναγνωριστικού του προιόντος (product\_id)
6. Επιστρέφεται η τελική απάντηση (αν η εισαγωγή των δεδομένων ήταν επιτυχής)
7. Δημιουργείται ένα τυχαίο συμμετρικό κλειδί
8. Κρυπτογραφείται το αναγνωριστικό του τελικού πελάτη (customer\_id)
9. Κρυπτογραφείται το αναγνωριστικό του σημείου παραλαβής (stop\_id)
10. Κρυπτογραφείται το συμμετρικό κλειδί με την χρήση κρυπτογράφησης βασισμένη σε χαρακτηριστικά (Attribute Based Encryption)
11. Δημιουργείται εγγραφή στον πίνακα «Orders» της κοινόχρηστης βάσης δεδομένων με τα παρακάτω δεδομένα:
    * Το αναγνωριστικό της παραγγελίας (order\_id)
    * Η ημερομηνία αποστολής (shipment\_date)
    * Το κρυπτογραφημένο αναγνωριστικό τελικού πελάτη (enc\_customer\_id)
    * Το κρυπτογραφημένο αναγνωριστικό σημείου παραλαβής (enc\_stop\_id)
    * Το κρυπτογραφημένο συμμετρικό κλειδί (abe\_enc\_key)

Όπως φαίνεται στο βήμα 11, αποθηκεύουμε τις ευαίσθητες πληροφορίες σε κρυπτογραφημένη μορφή. Προκειμένου να αποκρυπτογραφηθούν θα πρέπει πρώτα να αποκρυπτογραφηθεί το συμμετρικό κλειδί. Η αποκρυπτογράφηση αυτού, γίνεται μόνο από τις οντότητες που κατέχουν τα χαρακτηριστικά για τα οποία επέτρεψε ο προμηθευτής να γίνεται η αποκρυπτογράφηση.

Ο πάροχος υπηρεσιών τηλεματικής πλέον, μπορεί να ελέγξει για κάποια συγκεκριμένη μέρα αν υπάρχει παραγγελία προς παράδοση από κάποιον πελάτη του διαβάζοντας τον πίνακα «Orders» από την κοινόχρηστη βάση δεδομένων.



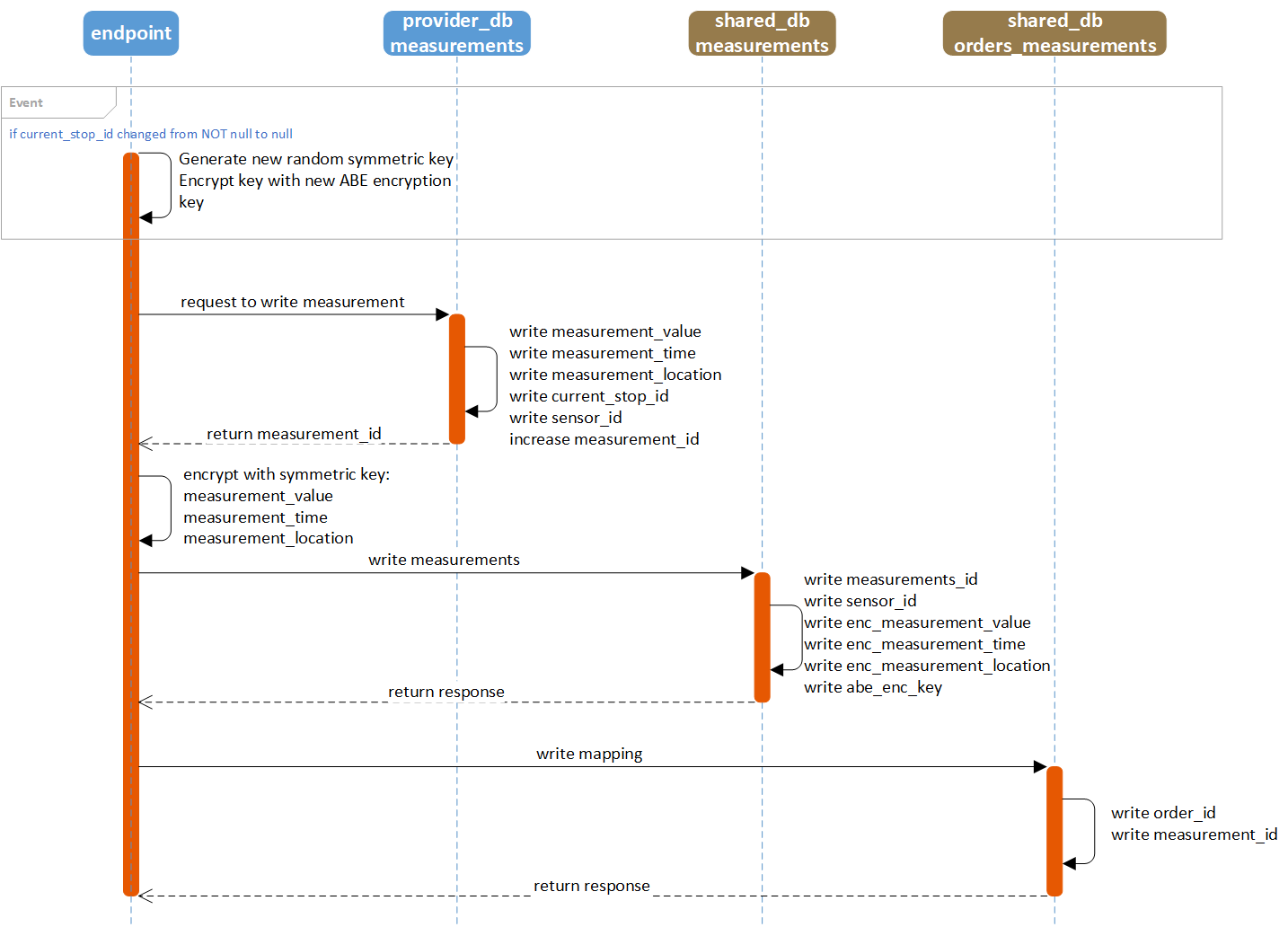
Στο παραπάνω διάγραμμα ο πάροχος τηλεματικών υπηρεσιών ελέγχει αν έχουν καταχωρηθεί παραγγελίες στην κοινόχρηστη βάση δεδομένων για μια συγκεκριμένη ημερομηνία «date». Έπειτα, για κάθε παραγγελία, αιτείται το αναγνωριστικό της πύλης (gateway\_id) που είναι εγκατεστημένο στο όχημα (vehicle\_id) της εκάστοτε παραγγελίας, από τον πίνακα στη βάση του (Gateways). Ύστερα, αποκρυπτογραφεί το συμμετρικό κλειδί βάσει ενός χαρακτηριστικού που του έχει προσδοθεί (Έχει φροντίσει ο πελάτης του για αυτό). Τέλος αποκρυπτογραφεί το αναγνωριστικό του τελικού πελάτη, καθώς και το αναγνωριστικό του σημείου παραλαβής.

Πλέον, ο πάροχος έχει όλες τις απαραίτητες πληροφορίες προκειμένου να καταγράψει, οργανώσει και κοινοποιήσει τα δεδομένα τηλεματικής.

Μετά την κλήση του τελικού σημείου που αναφέραμε παραπάνω, o πάροχος είναι σε θέση να γνωρίζει:

* Το αναγνωριστικό της παραγγελίας (order\_id)
* Το αναγνωριστικό του τελικού πελάτη (customer\_id)
* Το αναγνωριστικό του οχήματος που έχει αναλάβει την αποστολή (vehicle\_id)

Βάσει του αναγνωριστικού του οχήματος (vehicle\_id), γνωρίζει το αναγνωριστικό πύλης (gateway\_id) και κατά συνέπεια το αναγνωριστικό του αισθητήρα (sensor\_id) που θα αναλάβει να καταγράψει τις μετρήσεις. Έτσι, η πύλη, μπορεί να κάνει κλήση ενός τελικού σημείου (endpoint) που είναι υπεύθυνο για την εγγραφή των μετρήσεων.



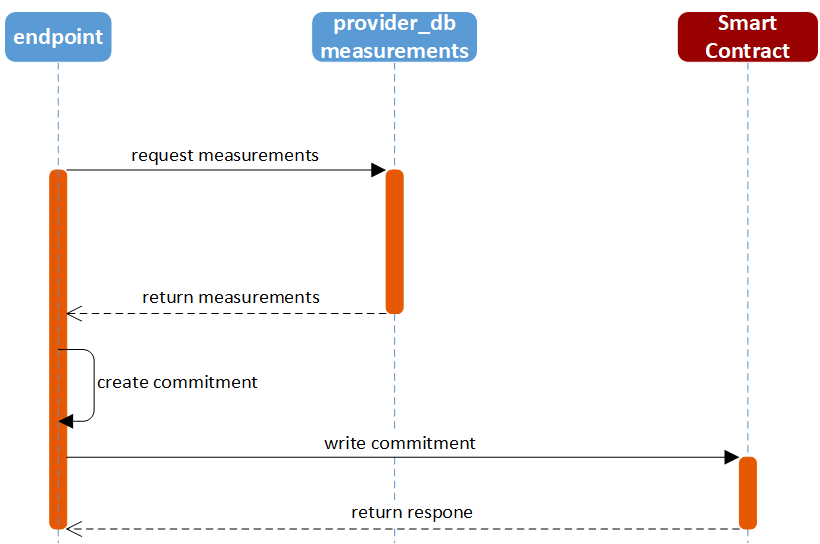
Η διαδικασία που εκτελεί το τελικό σημείο όπως φαίνεται στο παραπάνω διάγραμμα είναι η εξής:

1. Αρχικά δημιουργείται ένα τυχαίο συμμετρικό κλειδί ενώ παράλληλα δημιουργείται και ένα κρυπτογραφικό κλειδί με τη μέθοδο κρυπτογράφησης βάσει χαρακτηριστικών (ABE). Το ABE κλειδί μπορεί να χρησιμοποιηθεί από τους τελικούς πελάτες και είναι αυτό με το οποίο κρυπτογραφείται το συμμετρικό κλειδί.
   * Το βήμα 1 επαναλαμβάνεται μόλις προκληθεί ένα γεγονός. Το γεγονός αυτό αποτελείται από δύο στάδια:
     1. Το όχημα εισέρχεται σε σημείο παραλαβής. Άρα το τελικό σημείο θα γράψει το παρόν (stop\_id) στο πεδίο current\_stop\_id
     2. Το όχημα εξέρχεται από το σημείο παραλαβής. Άρα το τελικό σημείο θα γράψει στο πεδίο current\_stop\_id μία ουδέτερη τιμή

Αυτό θα σηματοδοτήσει την επανάληψη του βήματος 1 με την δημιουργία ABE κρυπτογραφικού κλειδιού που μπορούν να χρησιμοποιήσουν μόνο οι υπόλοιποι τελικοί πελάτες που δεν έχουν παραδοθεί ακόμη οι παραγγελίες της αποστολής.

1. Γίνεται η εγγραφή της μέτρησης στην βάση του παρόχου. Δηλαδή καταχωρούνται τα παρακάτω:
   * Η τιμή της μέτρησης (measurement\_value)
   * Η χρονική στιγμή που πάρθηκε η μέτρηση (measurement\_time)
   * Η γεωγραφική τοποθεσία που πάρθηκε η μέτρηση (measurement\_location)
   * Το παρόν σημείο παραλαβής (αν βρίσκεται) (current\_stop\_id)
   * Το αναγνωριστικό του αισθητήρα (sensor\_id)
   * Αυτόματη συμπλήρωση του αναγνωριστικού μέτρησης (measurement\_id)
2. Επιστρέφεται το αναγνωριστικό μέτρησης που δημιουργήθηκε (measurement\_id)
3. Γίνεται η κρυπτογράφηση των παρακάτω πεδίων με το συμμετρικό κλειδί:
   * Η τιμή της μέτρησης (measurement\_value)
   * Η χρονική στιγμή της μέτρησης (measurement\_time)
   * Η τοποθεσία της μέτρησης (measurement\_location)
4. Γίνεται η εγγραφή των δεδομένων της μέτρησης στην κοινόχρηστη βάση δεδομένων. Πιο συγκεκριμένα εγγράφονται:
   * Το αναγνωριστικό της μέτρησης (measurement\_id)
   * Το αναγνωριστικό του αισθητήρα (sensor\_id)
   * Η κρυπτογραφημένη τιμή της μέτρησης (enc\_measurement\_value)
   * Η κρυπτογραφημένη χρονική στιγμή που πάρθηκε η μέτρηση (enc\_measurement\_time)
   * Η κρυπτογραφημένη γεωγραφική τοποθεσία που πάρθηκε η μέτρηση (enc\_measurement\_location)
   * Το κρυπτογραφημένο συμμετρικό κλειδί (abe\_enc\_key)
5. Γίνεται επίσης η αντιστοίχιση της μέτρησης με την εκάστοτε παραγγελία. Δηλαδή, γίνεται η παρακάτω εγγραφή στην κοινόχρηστη βάση δεδομένων:
   * Το αναγνωριστικό της παραγγελίας (order\_id)
   * Το αναγνωριστικό της μέτρησης (measurement\_id)
6. Τέλος επιστρέφεται το τελικό αποτέλεσμα της εκτέλεσης διαδικασίας

Επιπρόσθετα, ο πάροχος υπηρεσιών τηλεματικής μπορεί να δημιουργήσει την δέσμευση (commitment) για κάποιο συγκεκριμένο πλήθος μετρήσεων ούτως ώστε να αποθηκευτεί στο έξυπνο συμβόλαιο (smart contract) προς μελλοντική χρήση για επαλήθευση από κάποιον πελάτη του.



# Ανάλυση μηχανισμών ασφάλειας

## Εισαγωγή

# Συμπεράσματα

# 

# Αναφορές

# Οπισθόφυλλο

«Η εργασία υλοποιήθηκε στο πλαίσιο της Δράσης ΕΡΕΥΝΩ – ΔΗΜΙΟΥΡΓΩ - ΚΑΙΝΟΤΟΜΩ και συγχρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση και εθνικούς πόρους μέσω του Ε.Π. Ανταγωνιστικότητα, Επιχειρηματικότητα & Καινοτομία (ΕΠΑνΕΚ) (κωδικός έργου:Τ2ΕΔΚ-02836)»

This research has been co‐financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code:T2EDK-02836)