# IoTFeds

Κατανεμημένες Αγορές Δεδομένων για IoT Ομοσπονδίες
Distributed Data Marketplaces for IoT Federations

## Π2.1 - Αναφορά Μηχανισμών Διαχείρισης Ομοσπονδιών IoT και Σημασιολογικής Διαλειτουργικότητας

ΔΡΑΣΗ ΕΘΝΙΚΗΣ ΕΜΒΕΛΕΙΑΣ:
«ΕΡΕΥΝΩ-ΔΗΜΙΟΥΡΓΩ-ΚΑΙΝΟΤΟΜΩ Β' ΚΥΚΛΟΣ»
«ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ, ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ & ΚΑΙΝΟΤΟΜΙΑ» (ΕΠΑνΕΚ)
Ειδική Υπηρεσία Διαχείρισης Επιχειρησιακού Προγράμματος Ανταγωνιστικότητα
Επιχειρηματικότητα και Καινοτομία (ΕΥΔ ΕΠΑνΕΚ)
Ειδική Υπηρεσία Διαχείρισης και Εφαρμογής Δράσεων στους τομείς
Έρευνας, Τεχνολογικής Ανάπτυξης και Καινοτομίας
(ΕΥΔΕ ΕΤΑΚ)

| | |
|---|---|
| **Έργο** | IoTFeds |
| **Κωδικός Έργου** | Τ2ΕΔΚ-02178 |
| **Ενότητα Εργασίας** | ΕΕ2 (Δ2.1) |
| **Υπεύθυνος Ενότητας** | ICOM |
| **Υπεύθυνος Παραδοτέου** | ICOM |
| **Έκδοση Παραδοτέου** | 1.0 |
| **Ημερομηνία** | 30 Σεπτεμβρίου, 2022 |

**Συντονιστής Παραδοτέου**
Ήλια Πιέτρη, ICOM

**Υπεύθυνοι Συγγραφής**
Ευάγγελος Αθανασάκης, ΕΚΕΤΑ
Κωνσταντίνος Βασιλόπουλος, ICOM
Γεώργιος Γόγολος, TERRACOM
Στέλιος Γκούσκος, TERRACOM
Παύλος Ευθυμιάδης, TERRACOM
Κωνσταντίνος Καλαμπόκης, TERRACOM
Δήμητρα Καραδήμου, AUEB
Ελένη Ζαρογιάννη, ICOM
Ευάγγελος Μαρκάκης, AUEB
Απόστολος Νάσιου, ICOM
Γεώργιος Νταρζάνος, AUEB
Σοφία Πολυμένη, ΕΚΕΤΑ
Ζήσης Σακελλαρίου, ΕΚΕΤΑ
Γεώργιος Σπανός, ΕΚΕΤΑ
Ναταλία Ούμνοβα, TERRACOM
Αθανάσιος Γ. Παπαϊωάννου, AUEB
Γεώργιος Δ. Σταμούλης, AUEB
Αλέξανδρος Τάγκας, TERRACOM
Κωνσταντίνος Τσάκαλος, ICOM
Παρασκευή Τοκμακίδου, TERRACOM
Μάριος Χαραλαμπίδης, ICOM

# Ακρωνύμια

| Ακρωνύμιο | Ορισμός |
|---|---|
| **API** | Application Programming Interface |
| **PM** | Particulate Matter |
| **SAREF** | Smart Applications REFerence |
| **UI** | User Interface |
| **IoT** | Internet of Things |
| **DLTs** | Distributed Ledger Technologies |
| **CIM** | Core Information Model |
| **DAO** | Decentralized Autonomous Organization |
| **HLF** | Hyperledger Fabric |
| **ABAC** | Attribute Based Access Control |
| **RBAC** | Role Based Access Control |
| **BaaS** | Blockchain as a Service |
| **RAP** | Resource Access Proxy |
| **CA** | Certification Authority |
| **SDOs** | Standard Developing Organizations |
| **W3C** | World Wide Web Consortium |
| **oneM2M** | One Machine-to-Machine Partnership |
| **ETSI** | European Telecommunications Standards Institute |
| **OGC** | Open Geospatial Consortium |
| **RDF** | Resource Description Framework |
| **RDFS** | Resource Description Framework Schema |
| **OWL** | Web Ontology Language |
| **SSN** | Semantic Sensor Network |
| **SOSA** | Sensor, Observation, Sample and Actuators |
| **NGSI** | Next Generation Service Interface |
| **NGSI-LD** | Next Generation Service Interface - Linked Data |
| **JSON** | JavaScript Object Notation |
| **JSON-LD** | JavaScript Object Notation for Linked Data |

| | |
|---|---|
| **CO** | Carbon Monoxide |
| **NO2** | Nitrogen Dioxide |
| **SO2** | Sulphur Dioxide |
| **URI** | Uniform Resource Identifier |
| **BIM** | Best-practice Information Model |
| **PIM** | Platform-specific Information Model |
| **AAM** | Authentication and Authorization Manager |
| **SM** | Semantic Manager |
| **RAP** | Resource Access Proxy |
| **RH** | Registration Handler |

# Σύνοψη Παραδοτέου

Το παραδοτέο αυτό παρουσιάζει τον σχεδιασμό και την ανάπτυξη των μηχανισμών διαχείρισης ομοσπονδιών στο Διαδίκτυο των Πραγμάτων (Internet of Things – IoT). Οι μηχανισμοί αυτοί επιτρέπουν σε έναν σύνολο από παρόχους ή/και καταναλωτές δεδομένων να δημιουργήσουν ομοσπονδίες IoT ώστε να μπορούν να μοιραστούν IoT δεδομένα και να δημιουργήσουν νέα προϊόντα προς διάθεση στην αγορά, καλύπτοντας έτσι λειτουργικότητες απαραίτητες για την εγκαθίδρυση μιας ομοσπονδίας IoT, τον ορισμό/επεξεργασία κανόνων και πολιτικών, και την εγγραφή (ή απομάκρυνση) μελών σε αυτή. Ο καθορισμός των απαιτούμενων λειτουργικοτήτων για τις ομοσπονδίες IoT βασίστηκε στις απαιτήσεις και την αρχιτεκτονική του συστήματος καθώς και τις τεχνολογίες που περιγράφηκαν στα παραδοτέα Π1.2 «Απαιτήσεις Συστήματος και Ενδιάμεση Αρχιτεκτονική Συστήματος» και Π1.3 «Τελική Αρχιτεκτονική Συστήματος και Επιλεγμένες Τεχνολογίες». Επίσης, η σχεδίαση και ανάπτυξη των μηχανισμών αυτών βασίζεται στους υπάρχοντες μηχανισμούς του ενδιάμεσου λογισμικού του έργου symbIoTe (Yuste, et al., 2018), (Christoph, et al., 2018) και οι οποίοι συνδυάζονται στα πλαίσια του έργου IoTFeds με μηχανισμούς blockchain όπως είναι η λειτουργία της διαχείρισης ταυτότητας (identity management) μέσω τεχνολογιών κατανεμημένου καθολικού (Distributed Ledger Technologies – DLTs), οι Αποκεντρωμένοι Αυτόνομοι Οργανισμοί (Decentralized Autonomous Organizations –DAOs), τα έξυπνα συμβόλαια (smart contracts) κ.α.

Επίσης, στο παρόν έγγραφο περιγράφεται η προσέγγιση που ακολουθείται για την επίτευξη της σημασιολογικής διαλειτουργικότητας στα πλαίσια του έργου IoTFeds και παρουσιάζεται το μοντέλο πληροφορίας του συστήματος IoTFeds στο οποίο θα βασιστούν οι ομοσπονδίες IoTFeds για να ορίσουν τα δικά τους μοντέλα πληροφορίας και να επιτύχουν out-of-the-box ή by-mapping διαλειτουργικότητα μεταξύ ομόσπονδων μελών. Το μοντέλο πληροφορίας IoTFeds βασίζεται και επεκτείνει το βασικό μοντέλο πληροφορίας (Core Information Model – CIM) του ενδιάμεσου λογισμικού symbIoTe (Jacoby, et al., 2017).

Τέλος, το παραδοτέο αυτό καταγράφει την υλοποίηση των παραπάνω μηχανισμών και των σχετικών στοιχείων λογισμικού των υποσυστημάτων που εμπλέκονται στην διαχείριση μίας ομοσπονδίας IoT και συνοδεύουν τον αντίστοιχο κώδικα Ανοιχτής Πηγής.

Τα αποτελέσματα του παραδοτέου αυτού θα αποτελέσουν τη βάση για επόμενες δράσεις (όπως Δ2.4 «Ασφαλής Πρόσβαση σε Δεδομένα IoT»/Π2.2 «Αναφορά Μηχανισμών Διαφήμισης, Ανακάλυψης και Ασφαλούς Πρόσβασης σε Πόρους IoT») όπου και θα εμπλουτιστούν οι λειτουργικότητες για τη διαχείριση της ομοσπονδίας με νέους μηχανισμούς π.χ., τον έλεγχο ακεραιότητας των δικαιωμάτων του χρήστη.

# Πίνακας Περιεχομένων

# 1 Αντικείμενο και Δομή του Παραδοτέου

Στο παραδοτέο αυτό παρουσιάζονται ο σχεδιασμός και η ανάπτυξη των μηχανισμών διαχείρισης μίας ομοσπονδίας IoT (Δ2.1 «Διαχείριση Ομοσπονδίας») καθώς και η προσέγγιση που ακολουθείται για την επίτευξη της σημασιολογικής διαλειτουργικότητας στα πλαίσια του έργου IoTFeds με την ανάπτυξη του μοντέλου πληροφορίας του συστήματος IoTFeds (Δ2.2 «Σημασιολογική Διαλειτουργικότητα»). Οι μηχανισμοί διαχείρισης μίας ομοσπονδίας IoT βασίζονται στις απαιτήσεις και την αρχιτεκτονική του συστήματος καθώς και τις τεχνολογίες που περιγράφηκαν στα παραδοτέα Π1.2 «Απαιτήσεις Συστήματος και Ενδιάμεση Αρχιτεκτονική Συστήματος» και Π1.3 «Τελική Αρχιτεκτονική Συστήματος και Επιλεγμένες Τεχνολογίες», χτίζοντας πάνω στους υπάρχοντες μηχανισμούς του ενδιάμεσου λογισμικού του symbIoTe και την επέκταση τους με τεχνολογίες blockchain. Οι λειτουργικότητες που αναπτύχθηκαν καλύπτουν τα σενάρια που περιγράφηκαν στο Π1.2 και σχετίζονται με την διαχείριση μίας ομοσπονδίας IoT τα οποία συνοψίζονται παρακάτω:

1. Εγγραφή χρήστη
2. Σύνδεση χρήστη
3. Εγγραφή IoT πλατφόρμας
4. Εγγραφή IoT συσκευών
5. Εγκαθίδρυση ομοσπονδίας
6. Προσθήκη μέλους ομοσπονδίας
7. Απομάκρυνση μελών από ομοσπονδία
8. Διαγραφή ομοσπονδίας και μελών
9. Επιλογή κανόνων και τροποποίηση.

Για την ανάπτυξη των λειτουργικοτήτων αυτών αξιοποιήθηκαν οι αντίστοιχες λειτουργικότητες του symbIoTe οι οποίες τροποποιήθηκαν ώστε να ενσωματωθεί η επικοινωνία με τις υπηρεσίες του BaaS που αναπτύχθηκαν για την καταγραφή των στοιχείων (όσον αφορά τις λειτουργικότητες 1, 3, 4), τον έλεγχο ακεραιότητας πληροφορίας (λειτουργικότητες 1-9), τον ορισμό και εκτέλεση έξυπνων συμβολαίων (λειτουργικότητες 5-9), την εκκίνηση και διεκπεραίωση διαδικασιών ψηφοφορίας (λειτουργικότητες 6, 7, 9), την ενσωμάτωση του νέου μοντέλου πληροφορίας (λειτουργικότητα 4) κ.α.

Ο σχεδιασμός των τελικών λειτουργικοτήτων και η υλοποίησή τους ακολουθεί τις τεχνολογίες και την μεθοδολογία της Ενότητας Εργασίας 1 - ΕΕ1 «Απαιτήσεις, Αρχιτεκτονική και Ολοκλήρωση Συστήματος» που θέτει τα θεμέλια για τον σωστό σχεδιασμό και την ανάπτυξη του πρωτοτύπου.

Το κείμενο στο παραδοτέο αυτό ακολουθεί την παρακάτω δομή:

Στο κεφάλαιο 2 περιγράφεται ο σχεδιασμός για τη διαχείριση της IoT ομοσπονδίας (αρχιτεκτονική).

Στο κεφάλαιο 3 αναλύονται οι βασικοί μηχανισμοί διαχείρισης IoT ομοσπονδίας που χρησιμοποιούνται στο έργο IoTFeds και αφορούν τη χρήση τεχνολογιών blockchain, μηχανισμών αυθεντικοποίησης και εξουσιοδότησης καθώς και μηχανισμών Αποκεντρωμένων Αυτόνομων Οργανισμών (Decentralized Autonomous Organizations – DAOs).

Στο κεφάλαιο 4 παρουσιάζεται η προσέγγιση που ακολουθείται για την επίτευξη της σημασιολογικής διαλειτουργικότητας στο σύστημα IoTFeds και τα μοντέλο πληροφορίας. Στα πλαίσια αυτά εξετάζονται υπάρχοντα πρότυπα και οντολογίες στο Διαδίκτυο των Πραγμάτων (Internet of Things – IoT) και την Έξυπνη Πόλη (Smart City), περιγράφονται τα υπάρχοντα μοντέλα στο ενδιάμεσο λογισμικό symbIoTe και τις πιλοτικές πλατφόρμες του IoTFeds τα οποία

αποτέλεσαν τη βάση για την επέκταση του μοντέλο πληροφορίας και παρουσιάζεται το τελικό μοντέλο διαλειτουργικότητας στο σύστημα IoTFeds.

Στο κεφάλαιο 5 περιγράφονται οι μέθοδοι και τα στοιχεία λογισμικού των υποσυστημάτων που συμμετέχουν στη διαχείριση μίας ομοσπονδίας IoT καθώς και οι σχετικές διεπαφές για την αλληλεπίδραση με αυτά όπως αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια του έργου IoTFeds και τα οποία συνοδεύουν τον κώδικα Ανοιχτής Πηγής.

# 2 Προσέγγιση για τη διαχείριση της IoT ομοσπονδίας

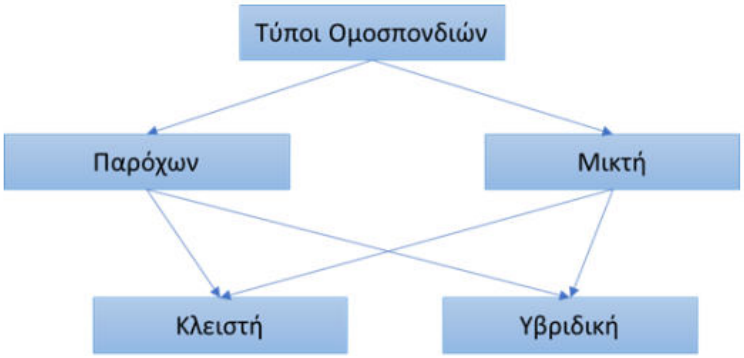
## *2.1 Αρχιτεκτονική Ομοσπονδιών και Αγορών της IoTFeds Πλατφόρμας*

Όπως περιγράφηκε στο Π1.3, στα πλαίσια του έργου IoTFeds, μια ομοσπονδία ορίζεται ως: «Ένα σύνολο φορέων που συνεργάζονται βάσει καλώς ορισμένων και από κοινού συμφωνηθέντων κανόνων και πολιτικών, για τη συλλογική παροχή υπηρεσιών δεδομένων». Σε κάθε ομοσπονδία «επισυνάπτεται» μια ομοσπονδιακή αγορά (federated marketplace) στην οποία τα μέλη της ομοσπονδίας εγγράφουν IoT πόρους (ή πηγές δεδομένων) και οποιοδήποτε μέλος της ομοσπονδίας έχει τη δυνατότητα να συνθέσει ένα προϊόν με πηγές δεδομένων (packaging) που προέρχονται από πολλαπλούς παρόχους/μέλη της ομοσπονδίας. Τα προϊόντα μπορούν να διατεθούν προς κατανάλωση από άλλα μέλη της ομοσπονδίας μέσα στην ομοσπονδιακή αγορά ή και εξωτερικά στην καθολική αγορά του IoTFeds για κατανάλωση από φορείς που δεν ανήκουν στην ομοσπονδία.

Η διαχείριση των ομοσπονδιών και των αγορών τους επιτυγχάνεται συνεργατικά από τα δύο βασικά υποσυστήματα της IoTFeds πλατφόρμας, το symbIoTe και το BaaS, ο σχετικός ρόλος και λειτουργικότητες των οποίων περιγράφεται στο Κεφάλαιο 5. Επίσης, η σημασιολογική διαλειτουργικότητα των δεδομένων που προέρχονται από διαφορετικές πηγές και παρόχους επιτυγχάνεται καθολικά αλλά και σε επίπεδο ομοσπονδίας. Η προσέγγιση που ακολουθείται για την επίτευξη της σημασιολογικής διαλειτουργικότητας περιγράφεται αναλυτικά παρακάτω στην Ενότητα 4.4.

Οι φορείς (χρήστες) που συμμετέχουν στις IoTFeds ομοσπονδίες μπορεί να έχουν τον ρόλο του *παρόχου IoT δεδομένων* (IoT data provider), του *καταναλωτή IoT δεδομένων* (IoT data consumer) ή και των δύο, δημιουργώντας δύο πιθανές κατηγορίες ομοσπονδιών στην IoTFeds πλατφόρμα (Εικόνα 1):

- *Ομοσπονδία παρόχων* (μόνο IoT data providers), όπου όλα τα μέλη της ομοσπονδίας θα πρέπει να παρέχουν υπηρεσίες δεδομένων και συνεπώς κάθε πάροχος είναι ταυτόχρονα και εν δυνάμει καταναλωτής των υπηρεσιών της ομοσπονδίας.
- *Μικτή ομοσπονδία,* όπου ένα μέλος μπορεί να έχει και μόνο το ρόλο του καταναλωτή IoT δεδομένων.




*Εικόνα 1. Κατηγορίες ομοσπονδιών της IoTFeds πλατφόρμας.*

Ένας καταναλωτής έχει πάντα πρόσβαση στην καθολική αγορά της IoTFeds πλατφόρμας, ενώ η πρόσβασή του στις ομοσπονδιακές αγορές εξαρτάται από τις ομοσπονδίες στις οποίες συμμετέχει. Η σύνθεση των προϊόντων (packaging) είναι δυνατή μόνο στις ομοσπονδιακές αγορές, ενώ στην καθολική αγορά ο καταναλωτής καλείται να επιλέξει από ένα σύνολο προκατασκευασμένων προϊόντων, χωρίς τη δυνατότητα σύνθεσης.

Ανάλογα με τον τύπο της αγοράς όπου μία ομοσπονδία διαθέτει τα προϊόντα (υπηρεσίες) της, η ομοσπονδία μπορεί επιπλέον να είναι είτε *κλειστή* είτε *υβριδική*:

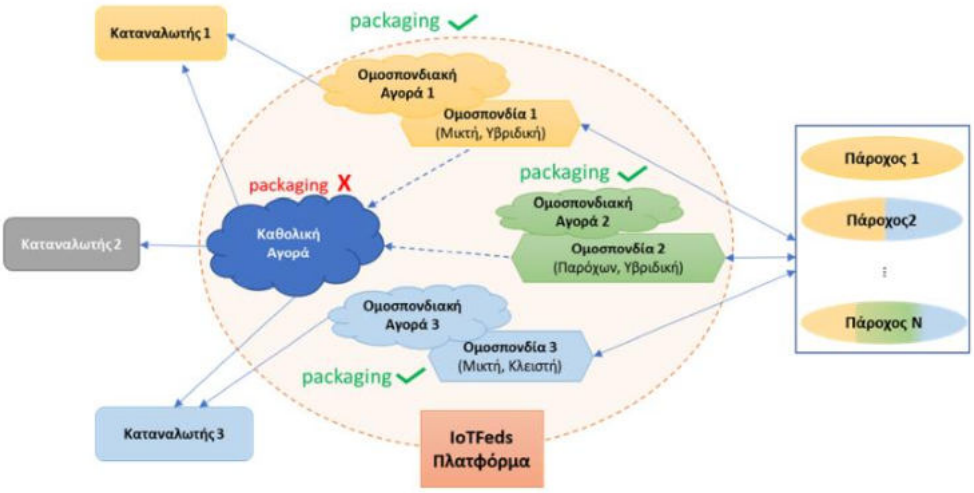
- Μια *κλειστή ομοσπονδία* ενδέχεται να επιλέξει μια πολιτική βάσει της οποίας τα συγκεκριμένα ομοσπονδιακά προϊόντα δεδομένων μπορούν να καταναλωθούν μόνο από τα μέλη της ομοσπονδίας, μέσω μιας ομοσπονδιακής αγοράς.
- Αντίθετα, μια *υβριδική ομοσπονδία* ενδέχεται να επιλέξει μια πολιτική όπου επιπρόσθετα επιτρέπει τη διάθεση αυτών των προϊόντων και σε μη μέλη της, μέσω μιας ανοιχτής καθολικής αγοράς (global marketplace) της IoTFeds πλατφόρμας.

Συνεπώς, και οι δύο τύποι αγοράς (καθολική αγορά καθώς και κάθε ομοσπονδιακή αγορά) διαθέτει δικούς της κανόνες και πολιτικές τιμολόγησης των υπηρεσιών δεδομένων. Στα πλαίσια της διαχείρισης της ομοσπονδίας που εξετάζεται στο παραδοτέο αυτό, παρέχεται η δυνατότητα η κάθε ομοσπονδία να ορίσει τους κανόνες που τη διέπουν καθώς και την τροποποίησή τους μέσα από κάποια διαδικασία ψηφοφορίας όπως θα αναλυθεί στην Ενότητα 3.3.

Συνολικά, τα τέσσερα διαφορετικά ήδη ομοσπονδιών που προκύπτουν και υποστηρίζονται από την IoTFeds πλατφόρμα είναι:

- Κλειστές ομοσπονδίες παρόχων (Data Providers-only Closed Federations)
- Υβριδικές ομοσπονδίες παρόχων (Data Providers-only Hybrid Federations)
- Κλειστές μικτές ομοσπονδίες (Mixed Closed Federation)
- Υβριδικές μικτές ομοσπονδίες (Mixed Hybrid Federation).

Στην παρακάτω εικόνα συνοψίζονται οι επιτρεπόμενες αλληλεπιδράσεις σε διαφορετικούς τύπους ομοσπονδίας: κλειστή και υβριδική ομοσπονδία παρόχων (πράσινο χρώμα) όπου συμμετέχουν μόνο πάροχοι με την υβριδική να επιτρέπει επιπρόσθετα τη διάθεση προϊόντων και στην καθολική αγορά, μικτή κλειστή ομοσπονδία (γαλάζιο χρώμα) όπου τα προϊόντα διατίθενται μόνο στην ομοσπονδιακή αγορά αλλά τόσο σε παρόχους όσο και σε καταναλωτές που συμμετέχουν σε αυτήν (όπως στον καταναλωτή 3), μικτή υβριδική (κίτρινο χρώμα) όπου καταναλωτές και πάροχοι μπορεί να είναι μέλη ή και μη της ομοσπονδίας (όπως οι καταναλωτές 1 και 2).

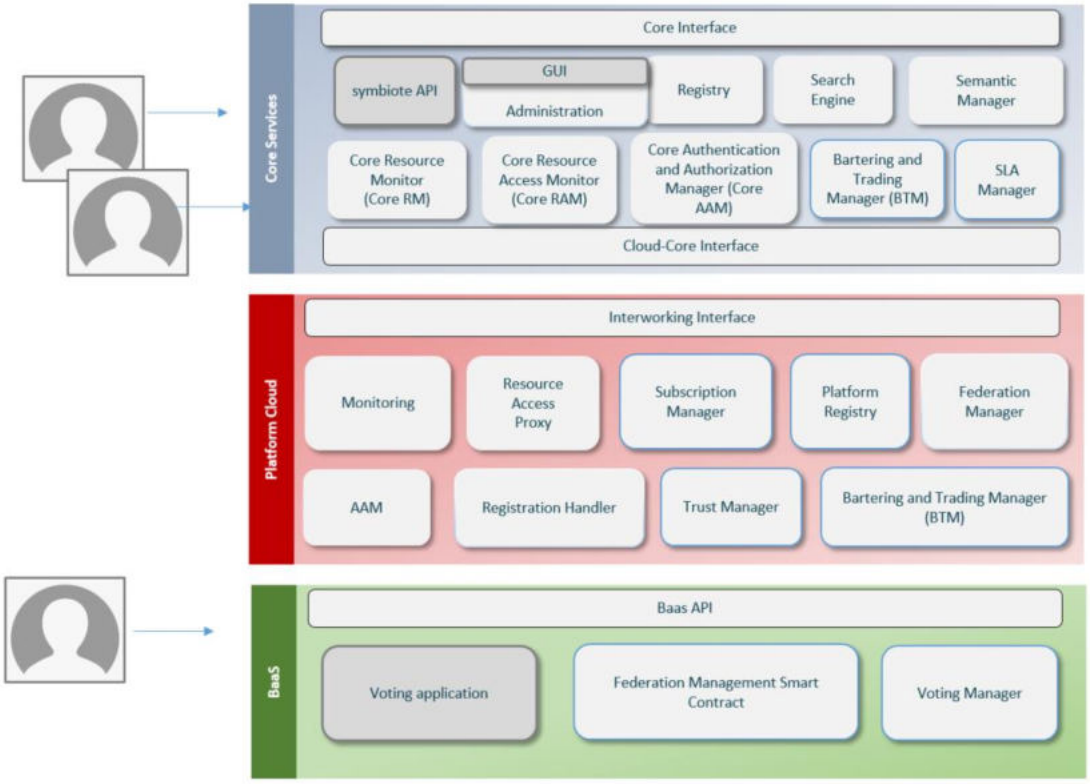



*Εικόνα 2: Αρχιτεκτονική ομοσπονδιών και αγοράς της IoTFeds πλατφόρμας.*

## 2.2 Διαχείριση Ομοσπονδιών

### 2.2.1 Αρχιτεκτονική Συστήματος

Το υποσύνολο των επιμέρους στοιχείων που κυρίως εμπλέκονται στη διαχείριση των ομοσπονδιών φαίνονται στην παρακάτω εικόνα.




*Εικόνα 3: Αρχιτεκτονική υποσυστήματος symbIoTe στο IoTFeds.*

Η αλληλεπίδραση του χρήστη με την IoTFeds πλατφόρμα για την διαχείριση ομοσπονδιών γίνεται κυρίως μέσω των στοιχείων Administration GUI και symbIoTe API του symbIoTe που έχουν επεκταθεί και υλοποιηθεί αντίστοιχα σύμφωνα με τις ανάγκες του έργου IoTFeds, ενώ η ψηφοφορία από κάθε χρήστη γίνεται μέσα από το στοιχείο Voting Application του BaaS. Πιο συγκεκριμένα, η διαχείριση των χρηστών, των πλατφορμών και των ομοσπονδιών επιτυγχάνεται μέσα από την γραφική διεπαφή χρήστη του Administration στοιχείου (Administration GUI) του symbIoTe παρέχοντας στον χρήστη κατάλληλες σελίδες και φόρμες για την καταχώρηση και ενημέρωση των σχετικών πληροφοριών. Από εκεί τα αιτήματα του χρήστη προωθούνται στα κατάλληλα σημεία πρόσβασης του επεκταμένου Administration στοιχείου (backend) του symbIoTe που εκκινεί την απαραίτητη ροή πληροφορίας για να ολοκληρωθεί το αίτημα. Επιπλέον, αιτήματα των χρηστών για τη διαχείριση των πόρων και τον διαμοιρασμό τους στην IoTFeds πλατφόρμα γίνεται μέσω του RestFul API του στοιχείου symbIoTe API που αναπτύχθηκε στα πλαίσια του IoTFeds έργου για να διευκολύνει την αλληλεπίδραση ενός χρήστη/πελάτη (όπως είναι μία εφαρμογή ή ένας πάροχος δεδομένων) με τις υπηρεσίες του symbIoTe. Στη συνέχεια, τα στοιχεία που εξυπηρετούν τα αιτήματα του χρήστη (Administration και symbIoTeAPI) εκκινούν τις απαραίτητες ροές επικοινωνίας με τα εμπλεκόμενα στοιχεία του symbIoTe όπως ο AAM (Authentication and Authorization Manager), FM (Federation Manager) και RH (Registration Handler), καθώς και το υποσύστημα BaaS για την καταγραφή των στοιχείων, τον έλεγχο ακεραιότητας πληροφορίας, τον ορισμό και εκτέλεση έξυπνων συμβολαίων, την εκκίνηση και διεκπεραίωση διαδικασιών ψηφοφορίας κ.α. Η επικοινωνία του symbIoTe με το υποσύστημα BaaS πραγματοποιείται μέσα από το BaaS API. Τέλος, ο χρήστης διεκπεραιώνει ψηφοφορίες στις οποίες εμπλέκεται μέσω του στοιχείου Voting Application του BaaS.

Παρακάτω περιγράφονται οι βασικές λειτουργικότητες των εμπλεκόμενων στοιχείων του symbIoTe. Οι κεντρικοποιημένες υπηρεσίες είναι οι ακόλουθες:

- *Administration.* Παρέχει τη γραφική διεπαφή χρήστη και την εξυπηρέτηση διαχειριστικών ενεργειών από τους χρήστες στο σύστημα σχετικά με την διαχείριση των χρηστών, πλατφορμών, ομοσπονδιών και των κανόνων τους. Στα πλαίσια της δράσης Δ2.1 το στοιχείο αυτό του λογισμικού έχει επεκταθεί για να υποστηρίξει τους κανόνες

και πολιτικές ομοσπονδίας του IoTFeds και την ενσωμάτωση των έξυπνων συμβολαίων του BaaS, όπως αναλύονται στην Ενότητα 3.3.

- *Registry*. Αποθηκεύει δεδομένα σχετικά με τους καταχωρημένους πόρους στο σύστημα με βάση το μοντέλο πληροφορίας σε συνεργασία με τον Sematic Manager (SM).
- *Search Engine*. Εξυπηρετεί την αναζήτηση πόρων που καλύπτουν τα κριτήρια στα σχετικά ερωτήματα και ακολουθώντας τους περιορισμούς ασφάλειας του συστήματος.
- *Semantic Manager (SM)*. Διαχειρίζεται τα μοντέλα πληροφορίας που χρησιμοποιούνται και επαληθεύει τη συμμόρφωση των πόρων με το μοντέλο πληροφορίας που χρησιμοποιείται. Στα πλαίσια της δράσης Δ2.2 «Σημασιολογική Διαλειτουργικότητα» το στοιχείο αυτό λογισμικού έχει επεκταθεί με την ενσωμάτωση του μοντέλου πληροφορίας έξυπνης πόλης όπως θα περιγραφεί στις επόμενες ενότητες.
- *Core Authentication and Authorization Manager (Core AAM)*. Χρησιμοποιεί τους περιορισμούς ασφαλείας του συστήματος με σκοπό την ασφαλή πρόσβαση και αναζήτηση των καταχωρημένων πόρων στο σύστημα.
- *Core Resource Monitor (CRM)*. Παρακολουθεί τη διαθεσιμότητα των καταχωρημένων στο σύστημα πόρων από το Registry.
- *Core Resource Access Monitor (CRAM)*. Λειτουργεί ως διακομιστής μεσολάβησης που ανακατευθύνει εφαρμογές και ενεργοποιεί τους πραγματικούς πόρους που προσφέρονται από τις πλατφόρμες. Συλλέγει στατιστικά στοιχεία πρόσβασης σε πόρους από το στοιχείο RAP των πλατφορμών προκειμένου για να διατηρήσει πληροφορίες δημοτικότητας πόρων.
- *Core Bartering and Trading Component (Core B&T)*. Περιλαμβάνει όλες τις λειτουργίες ανταλλαγής και συναλλαγών που πρέπει να συγκεντρωθούν και να αναπτυχθούν εντός μίας εφαρμογής, τη διαχείριση κουπονιών και στατιστικών.
- *Service Level Agreement Manager (SLA Manager)*. Διαχειρίζεται τον κύκλο ζωής των συμφωνιών επιπέδου υπηρεσιών (SLAs) για τις ομοσπονδίες των IoT πλατφορμών. *Α*ξιολογεί, με τη βοήθεια του στοιχείου Monitoring, εάν όλοι οι πόροι που μοιράζονται σε μια ομοσπονδία συμμορφώνονται με τους περιορισμούς QoS που ορίζονται στην ομοσπονδία.

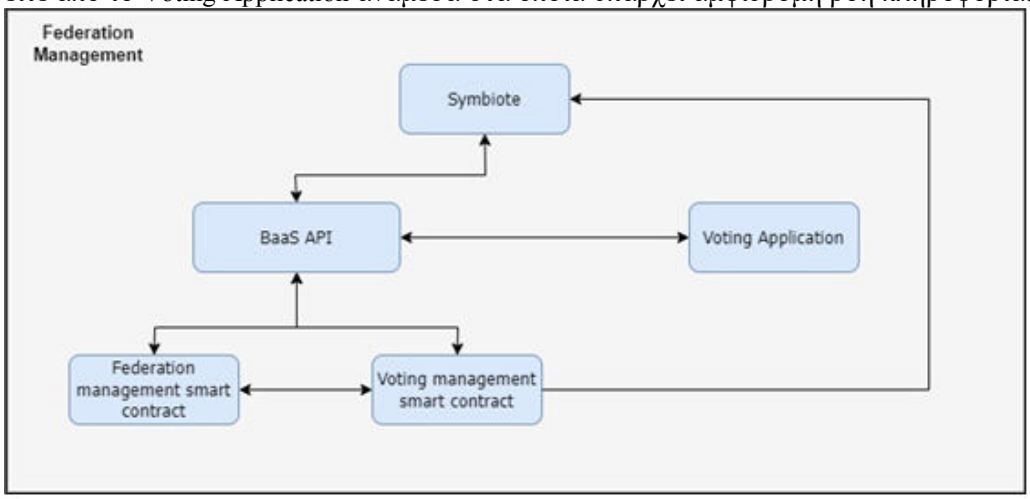
Τα στοιχεία του symbIoTe που συμμετέχουν στη διαχείριση των ομοσπονδιών στη μεριά της πλατφόρμας είναι τα παρακάτω:

- *Registration Handler (RH)*. Διαχειρίζεται την καταχώρηση των πόρων την πλατφόρμας σε συνεργασία με τις κεντρικοποιημένες υπηρεσίες του symbIoTe και του Platform Registry.
- *Resource Access Proxy (RAP)*. Διαχειρίζεται αιτήματα που αφορούν την προσπέλαση στους καταχωρημένους πόρους της πλατφόρμας και τους ομόσπονδους πόρους. Το στοιχείο αυτό του λογισμικού ενσωματώνει το νέο μοντέλο πληροφορίας Έξυπνης Πόλης καθώς καθορίζει τη δομή του αντικειμένου των μετρήσεων των πηγών που επιστρέφεται ως απόκριση σε μια κλήση.
- *Authentication and Authorization Manager (Platform AAM)*. Διαχειρίζεται την δυνατότητα πρόσβασης στους πόρους της πλατφόρμας και των συνεργατικών πλατφορμών resources
- *Monitoring*. Είναι υπεύθυνο για τη συλλογή μετρήσεων που σχετίζονται με την κατάσταση των καταχωρημένων πόρων και την πρόσβαση σε αυτούς.
- *Subscription Manager (SM)*. Προωθεί και διαχειρίζεται μηνύματα που αφορούν ενημερώσεις για τους πόρους στις ομοσπονδίες.
- *Platform Registry (PR)*. Διευκολύνει την αναζήτηση των ομόσπονδων πόρων που είναι καταχωρημένα τοπικά.
- *Federation Manager (FM)*. Διαχειρίζεται την κατάσταση των ομοσπονδιών πλατφορμών παίρνοντας ενημερώσεις από το Administration στοιχείο του symbIoTe.

- *Trust manager*. Υπολογίζει τις αξίες εμπιστοσύνης και φήμης στο επίπεδο των πόρων και των πλατφορμών.
- *Bartering and Trading Manager (BTM)*. Διαχειρίζεται τους αποκεντρωμένους μηχανισμούς ανταλλαγής και εμπορίας πόρων μεταξύ των ομόσπονδων πλατφορμών.

Επιπλέον, όπως αναφέρθηκε το στοιχείο λογισμικού symbIoTeAPI αναπτύχθηκε στα πλαίσια του έργου IoTFeds ώστε να διαχειρίζεται τα αιτήματα από έναν πελάτη (χρήστη) προς τα στοιχεία λογισμικού του υποσυστήματος symbIoTe ενσωματώνοντας την επικοινωνία με τις υπηρεσίες του BaaS για την καταγραφή των στοιχείων, τον έλεγχο ακεραιότητας πληροφορίας, τον ορισμό και εκτέλεση έξυπνων συμβολαίων, την εκκίνηση και διεκπεραίωση διαδικασιών ψηφοφορίας κ.α. Τροποποιήσεις σε στοιχεία του symbIoTe που αφορούν άλλες δράσεις θα περιγραφούν στα αντίστοιχα παραδοτέα.

Τέλος, στην Εικόνα 4 φαίνονται τα στοιχεία του BaaS που είναι υπεύθυνα για τη διαχείριση των ομοσπονδιών, αλλά και η ροή πληροφορίας μεταξύ αυτών. Τα δύο smart contracts επικοινωνούν μεταξύ τους ώστε να ανταλλάσσουν πληροφορίες σχετικά με τις ομοσπονδίες. Οι ενέργειες των smart contracts ενεργοποιούνται μέσα από το BaaS API, είτε αυτές καλούνται από το symbIoTe είτε από το Voting Application ανάμεσα στα οποία υπάρχει αμφίδρομη ροή πληροφορίας.




*Εικόνα 4: Διαχείριση ομοσπονδιών στο BaaS.*

Ο σχεδιασμός της αρχιτεκτονικής του συστήματος για την υποστήριξη των ομοσπονδιών βασίστηκε στην ανάλυση των απαιτήσεων στο Π1.3 και την περιγραφή των σεναρίων χρήσης του συστήματος στο Π1.2 που σχετίζονται με τη διαχείριση της ομοσπονδίας:

1. Εγγραφή χρήστη, η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και ενσωματώνει καταγραφή των στοιχείων στο BaaS.
2. Σύνδεση χρήστη, η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και ενσωματώνει έλεγχο ακεραιότητας στο BaaS.
3. Εγγραφή IoT πλατφόρμας, η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και ενσωματώνει καταγραφή των στοιχείων στο BaaS.
4. Εγγραφή IoT συσκευών, η οποία γίνεται μέσω της διεπαφής του symbIoTeAPI με καταγραφή των στοιχείων και έλεγχο ακεραιότητας στο BaaS.
5. Εγκαθίδρυση ομοσπονδίας, η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και ενσωματώνει καταγραφή των στοιχείων.
6. Προσθήκη μέλους ομοσπονδίας η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και κάνει χρήση ψηφοφορίας όπως θα αναλυθεί στην Ενότητα 3.3.
7. Απομάκρυνση μελών από ομοσπονδία η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και κάνει χρήση ψηφοφορίας όπως θα αναλυθεί στην Ενότητα 3.3.

8. Διαγραφή ομοσπονδίας και μελών η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και ενσωματώνει ενημέρωση των στοιχείων στο BaaS.

Επιλογή κανόνων και τροποποίηση η οποία γίνεται μέσα από τη γραφική διεπαφή του χρήστη (Administration GUI) και κάνει χρήση ψηφοφορίας όπως θα αναλυθεί στην Ενότητα 3.3..Στη συνέχεια παρουσιάζονται οι μηχανισμοί και διεπαφές του συστήματος που υλοποιήθηκαν ή επεκτάθηκαν για να καλύψουν τα παραπάνω σενάρια και την διαχείριση των ομοσπονδιών. Στοιχεία του συστήματος και μηχανισμοί που σχετίζονται με την αναζήτηση, ασφαλή πρόσβαση, φήμη/εμπιστοσύνη και την αγορά θα εμπλουτιστούν στις επόμενες σχετικές δράσεις.

# 3 Βασικοί μηχανισμοί διαχείρισης IoT ομοσπονδίας

Στην ενότητα αυτή αναλύονται οι βασικοί μηχανισμοί στους οποίους στηρίζεται η διαχείριση της ομοσπονδίας στο IoTFeds: (i) οι μηχανισμοί blockchain για την αποθήκευση και την ενημέρωση των πληροφοριών σχετικά με τις ομοσπονδίες IoTFeds και την χρήση έξυπνων συμβολαίων που ορίζουν τις επιτρεπόμενες ενέργειες σε αυτές, (ii) οι μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης για την πρόσβαση σε πόρους και στοιχεία με ασφαλή τρόπο και την επαλήθευση της αυθεντικότητας, και τέλος (iii) οι μηχανισμοί για την αποκεντρωμένη και αυτόνομη διακυβέρνηση των ομοσπονδιών IoTFeds ως αποκεντρωμένοι αυτόνομοι οργανισμοί.

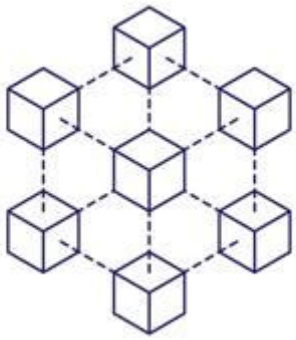
## *3.1 Τεχνολογίες Blockchain/DLT*

Οι τεχνολογίες του blockchain και του κατανεμημένου καθολικού (DLT) στα πλαίσια του IoTFeds εντάσσονται ώστε να εμπλουτίσουν και να επεκτείνουν την ήδη υπάρχουσα υποδομή του symbIoTe, με σκοπό την καταγραφή πληροφοριών, οι οποίες είναι απαραίτητο να καταγράφονται με τρόπο τέτοιο ώστε να μη μπορούν να παραποιηθούν, προσφέροντας έτσι ένα ακόμα επίπεδο ασφάλειας και εμπιστευτικότητας.

### *3.1.1 Τι είναι το blockchain*

Η τεχνολογία του blockchain (investopedia, n.d.) αποτελεί μια από τις πιο γρήγορα αναπτυσσόμενες τεχνολογίες των τελευταίων ετών, αν και είναι γνωστή από το 2008. Οι λειτουργίες και τα πλεονεκτήματα που προσφέρει, κάνουν την τεχνολογία του blockchain να βρίσκει εφαρμογές σε πολλούς τομείς, τόσο σε απλές εφαρμογές στην καθημερινότητα, όσο και σε τεχνολογικό επίπεδο.

Το blockchain είναι μια κατανεμημένη βάση δεδομένων ή αλλιώς ένα κατανεμημένο καθολικό (distributed ledger), και «τρέχει» σε κατανεμημένους κόμβους σε ένα δίκτυο υπολογιστών όπως φαίνεται στην Εικόνα 5[1]. Η αποθήκευση των δεδομένων στο blockchain γίνεται με τη μορφή blocks. Κάθε τέτοιο block από δεδομένα συνδέεται με το προηγούμενο block χρησιμοποιώντας μεθόδους κρυπτογραφίας, και πιο συγκεκριμένα την μέθοδο του κατακερματισμού (hashing). Σε κάθε block περιλαμβάνεται το hash και την ώρα δημιουργίας του προηγούμενου block, μαζί με το hash του ίδιου του block. Επομένως κάθε νέο block είναι συνδεδεμένο με ασφαλή τρόπο με το προηγούμενο block, σχηματίζοντας έτσι μια αλυσίδα από blocks (για αυτό και το όνομα blockchain, chain of blocks).



*Εικόνα 5: Ένα τυπικό blockchain.*

Μία από τις πιο συνηθισμένες χρήσεις του blockchain είναι για την καταγραφή συναλλαγών (οικονομικών και μη). Κάθε συναλλαγή, μαζί με τα δεδομένα που αυτή περιλαμβάνει, και γενικά οποιοδήποτε είδος ψηφιακής πληροφορίας θέλουμε να αποθηκευτεί στο blockchain συγκεντρώνεται σε blocks. Εξαιτίας της σύνδεσης των blocks μεταξύ τους, δεν είναι δυνατή οποιαδήποτε προσπάθεια παραποίησης ή επεξεργασίας κάποιας πληροφορίας που έχει ήδη

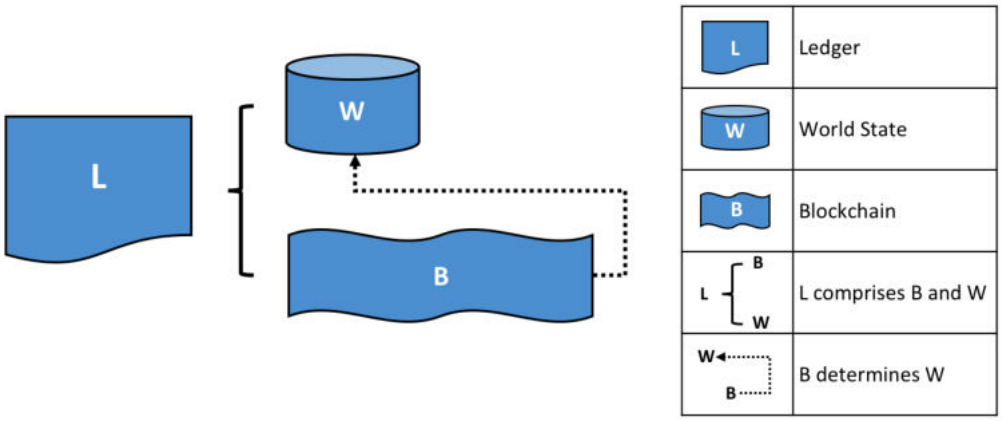
[1] https://www.istockphoto.com/illustrations/blockchain

αποθηκευτεί. Συνεπώς, όταν η πληροφορία αποθηκευτεί στον ledger, μπορούμε να κάνουμε λόγο για ένα αμετάβλητο καθολικό (immutable ledger). Μία από τις ψηφιακές πληροφορίες που θέλουμε να αποθηκεύσουμε στο blockchain στα πλαίσια του IoTFeds, είναι πληροφορίες σχετικά με τις ομοσπονδίες που θα δημιουργηθούν.

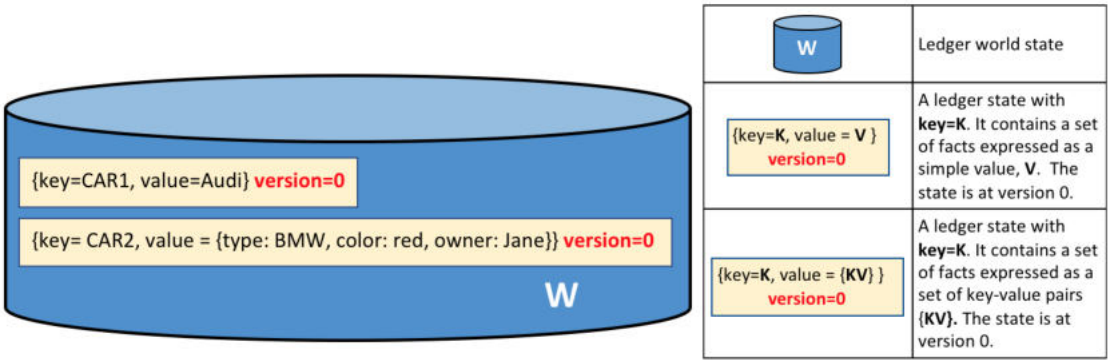
### *3.1.2 Hyperledger Fabric*

Το Hyperledger Fabric (HLF) (hyperledger-fabric) αποτελεί ένα πλαίσιο blockchain (framework), το οποίο προσφέρει μια πληθώρα λειτουργιών που είναι απαραίτητες για την υλοποίηση της IoTFeds πλατφόρμας και την διαχείριση των ομοσπονδιών στα πλαίσια αυτής. Σε μια πλατφόρμα όπως αυτή του IoTFeds, οι ομοσπονδίες που θα δημιουργούνται και οι συναλλαγές που θα λαμβάνουν χώρα μέσα σε αυτή θα πρέπει να καταγράφονται με αμετάβλητο τρόπο. Ωστόσο, ορισμένες πληροφορίες της ομοσπονδίας ενδέχεται να μεταβάλλονται με το χρόνο, όπως για παράδειγμα η πρόσθεση νέων μελών πέρα από τη στιγμή της δημιουργίας της ομοσπονδίας ή η αλλαγή σε ορισμένες πολιτικές ή κανόνες της.

Το κατανεμημένο καθολικό (distributed ledger) του HLF αποτελείται από δύο ξεχωριστά κομμάτια, το world state και το blockchain όπως φαίνεται στην Εικόνα 6. Το world state αποτελεί την κατανεμημένη βάση του HLF και καταγράφει τις πιο πρόσφατες τιμές των αποθηκευμένων δεδομένων σε μια χρονική στιγμή. Τα δεδομένα αποθηκεύονται με τη μορφή κλειδιού-τιμής (key-value pairs) και οι τιμές αυτών των δεδομένων μπορούν να μεταβληθούν και να αλλάξουν τιμή (Εικόνα 7). Αυτό γίνεται μέσα από συναλλαγές (transactions) οι οποίες έχουν πρόσβαση στο world state, μέσα από στοιχεία (SDKs) και τα Έξυπνα Συμβόλαια (Smart Contracts) που παρέχει το HLF, για την επικοινωνία με το καθολικό. Αυτές οι συναλλαγές πρώτα εγκρίνονται από τους συμμετέχοντες του δικτύου του blockchain και μόνο αν είναι έγκυρες θα αλλάξουν την τιμή (value) ενός συγκεκριμένου κλειδιού (key) και θα καταγραφούν στο καθολικό.

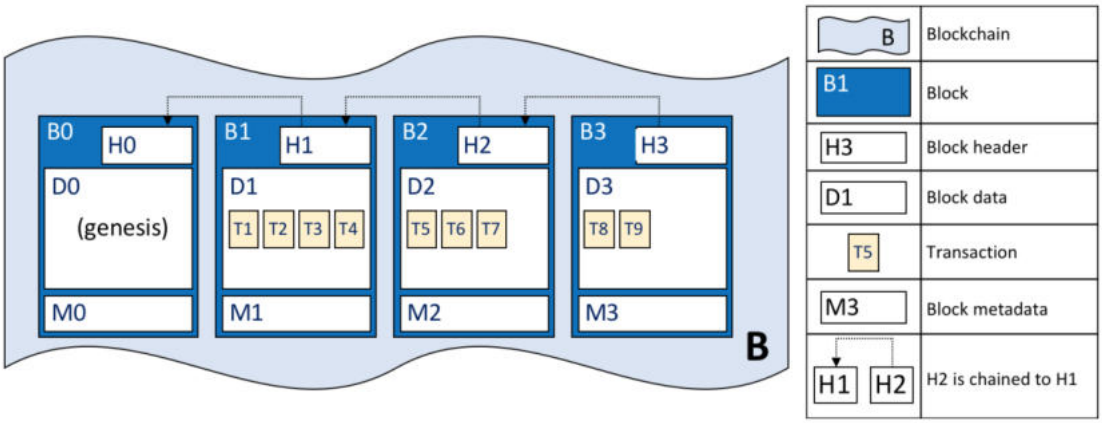



*Εικόνα 6. Το καθολικό (ledger) και τα περιεχόμενά του.*




*Εικόνα 7. Το world state στο HLF.*

Το blockchain μπορεί να θεωρηθεί ως ένα ιστορικό αρχείο που περιέχει όλες τις συναλλαγές και τα δεδομένα τους που περιγράφουν πως προέκυψαν οι πιο πρόσφατες τιμές των δεδομένων που είναι αποθηκευμένα στο world state. Αυτές οι συναλλαγές περιέχονται σε blocks και είναι αυτά που καταγράφονται με αμετάβλητο τρόπο στο καθολικό (Εικόνα 8).




*Εικόνα 8. Το blockchain στο HLF.*

### *3.1.3 Διαχείριση ομοσπονδιών στο Hyperledger Fabric*

Οι ομοσπονδίες του IoTFeds στα πλαίσια του blockchain θα αντιμετωπιστούν ως ψηφιακές πληροφορίες που θα αποθηκεύονται στο world state, έχοντας ένα μοναδικό αναγνωριστικό ως κλειδί (key) και ως τιμή (value) τις απαραίτητες για την ομοσπονδία πληροφορίες. Αυτές οι πληροφορίες περιλαμβάνουν, μεταξύ άλλων, τα μέλη της ομοσπονδίας, τους κανόνες της καθώς και τις πολιτικές χρέωσης και διαμοίρασης των κερδών.

Είναι σαφές ότι τέτοιες πληροφορίες πρέπει να μπορούν να αλλάζουν και να μην είναι αμετάβλητες. Για το λόγο αυτό, θα αναπτυχθούν Έξυπνα Συμβόλαια τα οποία θα ορίζουν τις απαραίτητες ενέργειες για την επεξεργασία αυτών των δεδομένων. Κάθε εκτέλεση μιας συνάρτησης (ενέργειας) που περιλαμβάνεται στα Έξυπνα Συμβόλαια αποτελεί ταυτόχρονα και ένα είδος συναλλαγής. Επομένως, κάθε φορά που θα εκτελείται μια τέτοια ενέργεια, τόσο η ενέργεια όσο και τα δεδομένα της θα καταγράφονται αμετάβλητα στο καθολικό, προσφέροντας έτσι διαφάνεια και ασφάλεια. Όλες οι απαραίτητες ενέργειες θα εκτελούνται μέσα από ένα API που θα παρέχει τα κατάλληλα endpoints για τη διαχείριση των ομοσπονδιών.

## *3.2 Μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης*

Οι μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης του IoTFeds, στηρίζονται πάνω στους υπάρχοντες μηχανισμούς της πλατφόρμας του symbIoTe και αποτελούν επέκταση αυτών με τον εμπλουτισμό και ενσωμάτωση μηχανισμών blockchain. Οι μηχανισμοί αυτοί θα εμπλουτιστούν και αναλυθούν πιο εκτενώς στο Π2.2.

Ο υπάρχοντας μηχανισμός αυθεντικοποίησης και εξουσιοδότησης στο symbIoTe βασίζεται στην παροχή διακριτικών (access tokens) και πιστοποιητικών (certificates) που επιτρέπουν στις εφαρμογές να αναζητούν και να έχουν πρόσβαση σε πόρους και στοιχεία με ασφαλή τρόπο καθώς και μία μέθοδο πρόκλησης-απόκρισης για την επαλήθευση της αυθεντικότητας. Υπεύθυνο για την παροχή των υπηρεσιών αυτών στο symbIoTe είναι το στοιχείο λογισμικού Authentication Authorization Manager – AAM που έχει υλοποιηθεί ως μία μικρο-υπηρεσία (microservice), καθώς το symbIoTe ακολουθεί την αρχιτεκτονική μικρο-υπηρεσιών (microservices architecture). Η μικρο-υπηρεσία AAM βασίζεται σε βιβλιοθήκες πελάτη, τον χειριστή ασφάλειας (Security Handler - SH) που έχει υλοποιηθεί στα πλαίσια του symbIoTe και χρησιμοποιείται από τα διαφορετικά στοιχεία λογισμικού (μικρο-υπηρεσίες) του symbIoTe. Περιλαμβάνει μεθόδους που επιτρέπουν από τη μία μεριά στους πελάτες να αποκτούν διαπιστευτήρια εξουσιοδότησης και από την άλλη πλευρά στις υπηρεσίες να αξιολογούν τα ληφθέντα διαπιστευτήρια τόσο για την εξουσιοδότηση λειτουργιών όσο και τον έλεγχο ταυτότητας των πελατών. Τέλος η μικρο-υπηρεσία AAM μπορεί να διαμορφωθεί ανάλογα με την πλατφόρμα στην οποία γίνεται η ανάπτυξη (deployment). Στα πλαίσια του symbIoTe και κατ' επέκταση στο IoTFeds η μικρο-υπηρεσία AAM είναι διαμορφωμένη σε δύο τύπους πλατφορμών: τον κεντρικό διαχειριστή αυθεντικοποίησης και εξουσιοδότησης (Core Authentication and Authorization Manager -Core AAM) και τον διαχειριστή αυθεντικοποίησης και εξουσιοδότησης στη μεριά της εκάστοτε απομακρυσμένης IoT πλατφόρμας (Platform Authentication and Authorization Manager - PAAM) (Yuste, και συν., 2018), (Skočir, et al., 2017).

### 3.2.1 Μηχανισμός για έλεγχο πρόσβασης

Ο μηχανισμός ασφαλείας στο symbIoTe βασίζεται στον έλεγχο πρόσβασης με βάση τα χαρακτηριστικά (Attribute Based Access Control – ABAC). Στη μεθοδολογία ABAC, ως χαρακτηριστικό (attribute) ορίζεται μια συγκεκριμένη ιδιότητα, ρόλος ή άδεια που σχετίζεται με μια οντότητα στο σύστημα και εκχωρείται μετά από μια διαδικασία ελέγχου ταυτότητας από τον διαχειριστή του συστήματος. Η πρόσβαση σε πόρους ελέγχεται μέσω των πολιτικών ελέγχου πρόσβασης. Η πολιτική πρόσβασης ορίζει ένα συγκεκριμένο συνδυασμό χαρακτηριστικών που απαιτούνται για την παραχώρηση πρόσβασης σε έναν πόρο και εκχωρείται σε κάθε πόρο από τον κάτοχό του. Επομένως, μια εφαρμογή ενός καταναλωτή δεδομένων μπορεί να λάβει πρόσβαση σε έναν πόρο μόνο εφόσον διαθέτει ένα σύνολο χαρακτηριστικών που ταιριάζουν με την προκαθορισμένη, από τον κάτοχο του πόρου, πολιτική πρόσβασης.

Στο symbIoTe:

- Τα «χαρακτηριστικά» αποθηκεύονται σε διακριτικά πρόσβασης (access tokens) και δημιουργούνται από τις αντίστοιχες μικρο-υπηρεσίες (microservices) αυθεντικοποίησης και εξουσιοδότησης (Authentication Authorization Manager - AAM) κάθε συμμορφούμενης με το symbIoTe, IoT πλατφόρμας. Τα διακριτικά πρόσβασης είναι συνεπώς ψηφιακά αντικείμενα που περιέχουν μία λίστα στοιχείων που σχετίζονται με τον μηχανισμό ασφαλείας που χρησιμοποιείται στα πλαίσια του symbIoTe, τα χαρακτηριστικά που παραχωρούνται στον κάτοχο του διακριτικού, και χρησιμοποιούνται για σκοπούς ελέγχου ταυτότητας και εξουσιοδότησης.
- Κάθε διακριτικό εξασφαλίζει την αυθεντικότητα των «χαρακτηριστικών» που περιέχει.
- Τα «χαρακτηριστικά» μπορούν να τροποποιηθούν όσον αφορά την πρόσβαση σε μια απομακρυσμένη IoT πλατφόρμα.
- Οι πολιτικές πρόσβασης ελέγχονται από την μικρο-υπηρεσία που είναι υπεύθυνη για την διαχείριση της πρόσβασης στα δεδομένα των IoT πόρων (Resource Access Proxy - RAP).

Ένα παράδειγμα του ABAC μηχανισμού του symbIoTe[2] όπου επιτρέπεται η πρόσβαση σε έναν πόρο μίας πλατφόρμας fed_h σε μία συγκεκριμένη ομοσπονδία fed_id από χρήστες που ανήκουν σε πλατφόρμες που εμπλέκονται στην ομοσπονδία φαίνεται παρακάτω Στο παράδειγμα αυτό, οι χρήστες με έγκυρο εσωτερικό διακριτικό, όπως υποδεικνύεται από το πεδίο req_loc ή αλλιώς requireAllLocalTokens, από οποιοδήποτε μέλος της ομοσπονδίας (UiTOP, Zastel) μπορούν να έχουν πρόσβαση με βάση τους κανόνες τα ομοσπονδίας:

```
{
   "policyType":"SFTAP",
   "requiredClaims":{
    "fed_id":"fedSmartCity",
    "fed_h":"UiTOP",
    "fed_s" : "2",
    "req_loc" : "false",
    "fed_m_1" : "Zastel",
    "fed_m_2" : "UiTOP"
   }
}
```

, ενώ η παρακάτω πολιτική μπορεί να περιορίσει την πρόσβαση μόνο στον χρήστη marketplace της πλατφόρμας symbioTe_Core_AAM:

```
{
   "policyType":"SLHTIBAP",
   "requiredClaims":{
    "sub":"marketplace",
    "iss":"SymbIoTe_Core_AAM"
   }
}
```

[2] https://github.com/symbiote-h2020/SymbIoTeSecurity#attribute-based-access-control

Στα πλαίσια του έργου IoTFeds, το ABAC θα συνδυαστεί με τους μηχανισμούς του BaaS για να διαπιστωθεί αν ένας χρήστης είναι εξουσιοδοτημένος να εκτελέσει συγκεκριμένες ενέργειες όπως για την αγορά προϊόντων στις επόμενες δράσεις (Δ2.4). Πιο συγκεκριμένα, η πολιτική πρόσβασης `SLHTIBAP` μπορεί να αξιοποιηθεί για να περιοριστεί η πρόσβαση στα δεδομένα πηγών δεδομένων ώστε να επιτρέπεται μόνο μέσω του στοιχείου λογισμικού αγοράς (marketplace) για προϊόντα που έχουν αγοραστεί. Ο εμπλουτισμένος μηχανισμός στα πλαίσια του έργου θα παρουσιαστεί στο παραδοτέο Π2.2.

### 3.2.2 Χρήση πιστοποιητικών για πιστοποίηση (authentication) και εξουσιοδότηση (authorization)

Ένας χρήστης του IoTFeds μπορεί να είναι εγγεγραμμένος είτε στην κεντρική πλατφόρμα του symbIoTe ή και σε οποιαδήποτε πλατφόρμα IoT συμβατή με το symbIoTe, ανάλογα με τον ρόλο που καλείται να επιτελέσει στο IoTFeds οικοσύστημα. Για παράδειγμα, οι πάροχοι των IoT δεδομένων και υπηρεσιών, αντιστοιχίζονται σε χρήστες IoT πλατφορμών του symbIoTe (στο PAAM), ενώ κάποιοι καταναλωτές των IoT δεδομένων της IoTFeds αγοράς (Global Marketplace) είναι πιθανό να είναι εγγεγραμμένοι μόνο στην κεντρική πλατφόρμα του symbIoTe (Core AAM) και όχι σε κάποια από τις IoT πλατφόρμες. Επομένως, ένας χρήστης μπορεί να εκτελέσει μια διαδικασία σύνδεσης με την παράδοση των δικών του διαπιστευτηρίων (όνομα χρήστη και κωδικό πρόσβασης).

Στον υπάρχοντα μηχανισμό του symbIoTe, χωρίς απώλεια γενικότητας, κάθε χρήστης μπορεί να διαχειρίζεται πολλαπλούς πόρους. Ενώ ο χρήστης διαθέτει μόνο ένα σύνολο διαπιστευτηρίων, κάθε πόρος έχει στην κατοχή του ένα μοναδικό ζεύγος δημόσιου-ιδιωτικού κλειδιού και ένα έγκυρο Πιστοποιητικό X.509 για κάθε χρήστη. Το Πιστοποιητικό X.509 εκδίδεται από τον κεντρικό διαχειριστή αυθεντικοποίησης και εξουσιοδότησης (Core AAM) ή εκείνον κάθε πλατφόρμας (PAAM), ανάλογα με το πού είναι εγγεγραμμένος ο χρήστης. Επομένως, για κάθε πόρο, ο χρήστης δημιουργεί ένα ζεύγος κλειδιών (που αποτελείται από δημόσιο κλειδί και ιδιωτικό κλειδί) και στέλνει ένα αίτημα υπογραφής πιστοποιητικού στον αντίστοιχο διαχειριστή αυθεντικοποίησης και εξουσιοδότησης (AAM). Στη συνέχεια, ο διαχειριστής αυθεντικοποίησης και εξουσιοδότησης δημιουργεί και παραδίδει ένα έγκυρο Πιστοποιητικό στον χρήστη. Τέλος, το πιστοποιητικό αποθηκεύεται από τον εκάστοτε πόρο. Προκειμένου να καταστήσει τη διαδικασία για τη δημιουργία όλων των πιστοποιητικών επεκτάσιμη και αποτελεσματική, το symbIoTe χρησιμοποιεί μια αρχιτεκτονική αλυσίδας αξιοπιστίας πιστοποιητικών. Σύμφωνα με τη λογική ABAC που περιγράφηκε προηγουμένως, οι ιδιότητες του χρήστη κωδικοποιούνται από χαρακτηριστικά που είναι αποθηκευμένα σε μια αποκλειστική δομή δεδομένων που αναφέρεται ως διακριτικό (token). Στο symbIoTe, ένα διακριτικό μπορεί να δημιουργηθεί μόνο από έναν διαχειριστή αυθεντικοποίησης και εξουσιοδότησης. Επιπλέον, περιέχει κυρίως τη λίστα των χαρακτηριστικών που έχουν εκχωρηθεί (και ισχύει) για έναν χρήστη εντός της κεντρικοποιημένης πλατφόρμας του symbIoTe ή σε οποιαδήποτε άλλη IoT πλατφόρμα.

Στη αλυσίδα αξιοπιστίας πιστοποιητικών του symbIoTe, εμπλέκονται 3 κατηγορίες πιστοποιητικών: (α) το πιστοποιητικό symbIoTe Core που έχε τον ρόλο του πιστοποιητικού ρίζας (Root Certificate) και από το οποίο ξεκινάει η αλυσίδα πιστοποίησης (certificate chain), (β) το πιστοποιητικό IoT πλατφόρμας του symbIoTe που έχει τον ρόλο του Ενδιάμεσου Πιστοποιητικού, μίας ενδιάμεσης αρχής έκδοσης πιστοποιητικών που μπορεί να υπογράψει πιστοποιητικά για λογαριασμό της αρχής έκδοσης πιστοποιητικών ρίζας και (γ) το πιστοποιητικό εφαρμογής (καταναλωτή δεδομένων) που είναι ένα πιστοποιητικό Τέλους (Leaf) και συνδέει μια τιμή δημόσιου κλειδιού με έναν χρήστη ή μια εφαρμογή, με κάθε χρήστη να απαιτείται να λάβει ένα πιστοποιητικό για κάθε συσχετισμένο πόρο.

Στα πλαίσια της επέκτασης του μηχανισμού για τις ανάγκες του έργου, το BaaS δρα ως ένα δεύτερο στρώμα αυθεντικοποίησης. Κάθε χρήστης του IoTFeds πρέπει να είναι εγγεγραμμένος και στο blockchain. Συνεπώς, κατά τη διαδικασία εγγραφής στην κεντρική πλατφόρμα του symbIoTe τα στοιχεία του χρήστη καταχωρούνται και στη βάση του blockchain. Κάθε φορά που ένας νέος χρήστης εγγράφεται στο blockchain, το Certification Authority (CA) που χρησιμοποιείται στο blockchain εκδίδει ένα Πιστοποιητικό X.509, που αναγνωρίζει μοναδικά κάθε χρήστη. Εκτός από το CA που παρέχεται από το HLF, υπάρχει και ένα Membership Service

Provider (MSP). Το MSP είναι υπεύθυνο για την αυθεντικοποίηση του χρήστη με την έννοια ότι μπορεί το Πιστοποιητικό X.509 να είναι πράγματι έγκυρο αλλά να μην έχει εκδοθεί από ένα CA το οποίο να εμπιστεύεται το MSP. Συνδυάζοντας, λοιπόν, αυτά τα δύο στοιχεία του HLF παρέχεται ένας επιπλέον μηχανισμός αυθεντικοποίησης του χρήστη.

### 3.2.3 Μέθοδος πρόκλησης απόκρισης (challenge response) για επαλήθευση αυθεντικότητας (authenticity verification).

Η διαδικασία πρόκλησης-απόκρισης στο symbIoTe επαληθεύει ότι μία οντότητα όπως χρήστης ή εφαρμογή που χρησιμοποιεί ένα διακριτικό είναι πραγματικά εκείνη για την οποία έχει εκδοθεί αυτό το διακριτικό από την μικρο-υπηρεσία AAM. Η διαδικασία βασίζεται σε κρυπτογράφηση δημόσιου κλειδιού με τον κάτοχο του διακριτικού να έχει στην κατοχή του ένα ιδιωτικό κλειδί που σχετίζεται με ένα δημόσιο κλειδί το οποίο είναι αποθηκευμένο στο διακριτικό. Ο κάτοχος του κλειδιού εκτελεί ορισμένες κρυπτογραφικές λειτουργίες χρησιμοποιώντας το ιδιωτικό του κλειδί και εκτελείται μια αντίθετη λειτουργία χρησιμοποιώντας το δημόσιο κλειδί για να επαληθευτεί η αυθεντικότητα της οντότητας. Περισσότερες λεπτομέρειες θα περιγραφούν στο Π2.2.

### 3.2.4 Μηχανισμοί ασφαλείας στα πλαίσια της IoT ομοσπονδίας

Για την επίτευξη της αυθεντικοποίησης, οι ομοσπονδίες χρησιμοποιούν πλήρως την υποδομή δημοσίου κλειδιού symbIoTe (Public Key Infrastructure - PKI) που περιγράφηκε στην προηγούμενη παράγραφο για να διαχειριστούν τις ταυτότητες των πελατών και των υπηρεσιών που συμμετέχουν σε κατανεμημένα σενάρια (για την απευθείας επικοινωνία μεταξύ των πλατφορμών και την αμοιβαία πιστοποίηση χωρίς την εμπλοκή επικοινωνίας με τις κεντρικοποιημένες υπηρεσίες του symbIoTe). Πιο συγκεριμένα, οι κεντρικοποιημένες υπηρεσίες του symbIoTe προσφέρουν μια Αρχή Πιστοποίησης Ρίζας (CA) και οι εγγεγραμμένες πλατφόρμες, στις υπηρεσίες νέφους τους, έχουν ενδιάμεσες εξουσίες αρχής πιστοποίησης για την έκδοση διαπιστευτηρίων για τους χρήστες τους.

Η υποδομή δημοσίου κλειδιού που ορίζεται από το symbIoTe παρέχει την υποστήριξη για τον έλεγχο ταυτότητας των φορέων και υπηρεσιών σε σενάρια ομοσπονδίας δημιουργώντας αλυσίδες πιστοποίησης και επαληθεύοντας εάν τα μέρη που επικοινωνούν προέρχονται από την ίδια ρίζα πιστοποίησης (τις κεντρικές υπηρεσίες του symbIoTe), γεγονός που αποδεικνύει την αυθεντικότητά τους.

Εν κατακλείδι, το κρυπτογραφικό υλικό που δημιουργείται σε αυτό το επίπεδο χρησιμοποιείται ώστε να παρέχονται υπογραφές για πακέτα δεδομένων (payloads) στο επίπεδο εξουσιοδότησης. Με αυτόν τον τρόπο επιτυγχάνεται ένας μηχανισμός αμοιβαίας επαλήθευσης ταυτότητας, όπου τόσο ο πελάτης (η οντότητα που κάνει το αίτημα) όσο και η υπηρεσία μπορούν να υπογράψουν τα πακέτα δεδομένων του οργανισμού και να επικυρώνονται από τον παραλήπτη της επικοινωνίας.

Όσον αφορά την επικύρωση διαπιστευτηρίων, η επικύρωση των διακριτικών, των κλειδιών που είναι αποθηκευμένα σε αυτά και της υπογραφής είναι ένα από τα πιο σημαντικά εργαλεία ασφαλείας στο symbIoTe. Οι μηχανισμοί ασφαλείας καθώς και η ενσωμάτωση με τους μηχανισμούς του Blockchain θα περιγραφούν εκτενώς στο Π2.2.

Συνοπτικά, στα πλαίσια του symbIoTe η διαδικασία που ακολουθείται είναι η παρακάτω:

- Ένα ενοποιημένο API εκτίθεται στους προγραμματιστές εφαρμογών για την επικύρωση διακριτικών, που είναι διαθέσιμα σε μία βοηθητική βιβλιοθήκη ασφαλείας, προκειμένου να ελέγξει την επικύρωση. Το επίπεδο ασφαλείας του symbIoTe επαληθεύει πρώτα το περιεχόμενο συμβολοσειράς του διακριτικού, για να προσδιορίσει εάν έχει καταστραφεί κατά τη μετάδοση ή εάν έχει σωστή υπογραφή και δεν έχει λήξει ακόμη.
- Στη συνέχεια, ο διαχειριστής αυθεντικοποίησης και εξουσιοδότησης ελέγχει το διακριτικό. Οι διαχειριστές αυθεντικοποίησης και εξουσιοδότησης της κεντρικοποιημένης πλατφόρμας (Core AAM) ή των IoT πλατφορμών (PAAMs), ελέγχουν εάν ο εκδότης του διακριτικού υπάρχει στο οικοσύστημα symbIoTe, εάν το δημόσιο κλειδί του εκδότη ή του υποκειμένου έχει ανακληθεί ή εάν το πιστοποιητικό του εκδότη ή του υποκειμένου έχει λήξει. Επιπλέον, οι διαχειριστές αυθεντικοποίησης και εξουσιοδότησης ελέγχουν στις βάσεις δεδομένων τους, εάν το διακριτικό δεν έχει

ανακληθεί λόγω παραβίασης ασφαλείας. Υπάρχουν διαφορετικοί τύποι διακριτικών: το εσωτερικό διακριτικό, home token, που εκδίδεται για έναν χρήστη που έχει λογαριασμό στον συγκεκριμένο εκδότη (AAM), το εξωτερικό διακριτικό (foreign token) που αφορά την έκδοση διακριτικού με χρήση άλλου διακριτικού και το διακριτικό επισκέπτη (guest token) για διακριτικό που εκδόθηκε για χρήστη χωρίς διαπιστευτήρια (επισκέπτη).

- Στην περίπτωση εξωτερικού διακριτικού, διακριτικού που έχει εκδοθεί από άλλη πλατφόρμα, γίνεται επικύρωση συμμετοχής στην ομοσπονδία. Εάν μια πλατφόρμα έχει αφαιρεθεί από τουλάχιστον μία ομοσπονδία μετά την δημιουργία του εξωτερικού της διακριτικού, το διακριτικό θεωρείται άκυρο και ο πελάτης πρέπει να αποκτήσει νέο.
- Επιπρόσθετα, εάν ένας διαχειριστής αυθεντικοποίησης και εξουσιοδότησης μιας ομόσπονδης πλατφόρμας παρουσιαστεί με εξωτερικό διακριτικό που έχει εκδόσει ο ίδιος, προσεγγίζει την πλατφόρμα που εξέδωσε το εσωτερικό διακριτικό για να ελέγξει εάν η προέλευση των διαπιστευτηρίων (το προαναφερθέν εσωτερικό διακριτικό και τα δημόσια κλειδιά πελατών) δεν ανακλήθηκαν, γεγονός που θα προκαλούσε ανάκληση του εξωτερικού διακριτικού που χρησιμοποιήθηκε στην προσπάθεια εξουσιοδότησης.

Λόγω των απαιτήσεων των ομοσπονδιών, το symbIoTe επιτρέπει στους ακόλουθους φορείς να ανακαλέσουν τα ακόλουθα:

- Ο διαχειριστής του symbIoTe μπορεί να ανακαλέσει τα πιστοποιητικά στοιχείων λογισμικού του κεντρικοποιημένου διαχειριστή αυθεντικοποίησης και εξουσιοδότησης.
- Ο διαχειριστής του διαχειριστή αυθεντικοποίησης και εξουσιοδότησης μιας πλατφόρμας μπορεί να ανακαλέσει τα πιστοποιητικά των δικών του στοιχείων λογισμικού.
- Όλοι οι φορείς όλων των διαχειριστών αυθεντικοποίησης και εξουσιοδότησης μπορούν να ανακαλέσουν τα εξωτερικά διακριτικά.

Όλοι οι φορείς θα πρέπει να παρουσιάσουν τα διαπιστευτήριά τους (όνομα χρήστη και κωδικό πρόσβασης) για να εξουσιοδοτηθούν οι ενέργειές τους. Σε περίπτωση ανάκλησης εξωτερικού διακριτικού, απαιτείται μόνο το αντίστοιχο εσωτερικό διακριτικό.
Σε επίπεδο εξουσιοδότησης, αξιοποιείται ο μηχανισμός ABAC που περιγράφηκε νωρίτερα. Αυτό επιτρέπει στους κατόχους των πλατφορμών να αντιστοιχήσουν τα εγγενή σχήματα εξουσιοδότησής τους, π.χ. πολιτικές ελέγχου πρόσβασης βάσει ταυτότητας πάνω από το ABAC που είναι διαθέσιμο εξ αρχής από το symbIoTe.
Το symbIoTe χρησιμοποιεί την τεχνολογία JSON διακριτικού ιστού (JSON web token[3]). Όλα τα διαπιστευτήρια εξουσιοδότησης παραδίδονται στους φορείς από τους διαχειριστές αυθεντικοποίησης και εξουσιοδότησης, υπογεγραμμένα με τα πιστοποιητικά τους. Ο παραπάνω μηχανισμός του symbIoTe για την εξουσιοδοτημένη πρόσβαση στα δεδομένα των πηγών δεδομένων θα εμπλουτιστεί με ένα επιπλέον επίπεδο ασφαλείας στο BaaS για την εξουσιοδοτημένη πρόσβαση στο επίπεδο των προϊόντων δεδομένων. Ο συνδυασμός των δύο μηχανισμών αποτελεί αντικείμενο επόμενου παραδοτέου (Π2.2) και θα διασφαλίσει την πρόσβαση στα δεδομένα μέσω επιλεγμένων λύσεων αγορών IoTFeds (Π3.2 - «Αναφορά μηχανισμών Αγοράς IoT»).

## *3.3 IoTFeds Ομοσπονδίες ως Αποκεντρωμένοι Αυτόνομοι Οργανισμοί*

Στα πλαίσια αυτής της ενότητας μελετάμε πως μπορεί να επιτευχθεί η αποκεντρωμένη και αυτόνομη διακυβέρνηση των ομοσπονδιών IoTFeds μέσω της δημιουργίας Αποκεντρωμένων Αυτόνομων Οργανισμών (Decentralized Autonomous Organizations - DAOs).
Η έννοια των DAOs εισήχθη από τον Christoph Jentzsch (Christoph J. , November 2016) και έκτοτε έχουν δοθεί πολλαπλοί διαφορετικοί ορισμοί (Buterin, 2014) – (Wang, 2019). Ένας

[3] https://www.rfc-editor.org/rfc/rfc7519

DAO θα μπορούσε να περιγραφεί ως ένας οργανισμός (ή μια κοινότητα) του οποίου η διακυβέρνηση επιτυγχάνεται χωρίς την ύπαρξη κάποια κεντρικής αρχής ή ιεραρχίας, αλλά μέσω συλλογικών αποφάσεων των οντοτήτων που συμμετέχουν σε αυτόν. Η αποκεντρωμένη και αυτοματοποιημένη διακυβέρνηση των DAOs επιτυγχάνεται μέσω ενός συνόλου κανόνων και πολιτικών που ορίζουν τον τρόπο λειτουργίας του οργανισμού, οι οποίοι καταγράφονται και εφαρμόζονται στο blockchain με την μορφή έξυπνων συμβολαίων. Συνεπώς, οι DAOs αποτελούν έναν ισχυρό μηχανισμό για ένα «δημοκρατικό» τρόπο διακυβέρνησης οργανισμών ή κοινοτήτων που εξυπηρετεί τα συμφέροντα του συνόλου των μελών της, μετριάζοντας έτσι προβλήματα τύπου «προϊσταμένου - υφισταμένου» (Grossman & Hart, 1992) ή «τραγωδίας των κοινών» (Hardin, 1968). Σε αυτού του τύπου τα προβλήματα μια οντότητα (π.χ. «προϊστάμενος») καλείται να λάβει αποφάσεις για την διαχείριση ενός συνόλου κοινών πόρων. Οι αποφάσεις αυτές θα πρέπει εξυπηρετούν τα συμφέροντα ενός συνόλου υφισταμένων (ή το κοινό καλό) που έχουν συνεισφορά σε αυτούς τους κοινούς πόρους. Από την άλλη, τα απαραίτητα κίνητρα θα πρέπει να παρέχονται στο «προϊστάμενο», ώστε οι αποφάσεις του να είναι προς το συμφέρον όλων. Σε περίπτωση που δεν παρέχονται τα κατάλληλα κίνητρα, ενδέχεται να υπάρξουν προβλήματα σύγκρουσης συμφερόντων και ηθικού κινδύνου, και η λήψη αποφάσεων να μη γίνεται με γνώμονα το κοινό καλό.

Οποιαδήποτε απόφαση σε ένα DAO λαμβάνεται μετά από ψηφοφορία των μελών του, η οποία διενεργείται στο blockchain μέσω έξυπνων συμβολαίων. Στην πιο απλή μορφή των DAOs, κάθε μέλος του οργανισμού έχει την δυνατότητα να καταθέσει μια πρόταση προς εξέταση από τα υπόλοιπα μέλη. Αυτή η πρόταση μπορεί π.χ. να σχετίζεται με μια λειτουργική ή διαχειριστική δράση του οργανισμού. Αφού αυτή η πρόταση γίνει γνωστή σε όλα τα μέλη του οργανισμού, στην συνέχεια καλείται μια διαδικασία ψηφοφορίας για την αποδοχή της πρότασης ή όχι. Το απαιτούμενο ποσοστό θετικών ψήφων για την αποδοχή μιας πρότασης ορίζεται από τους κανόνες του DAO και καταγράφεται στο σχετικό έξυπνο συμβόλαιο. Τόσο για την κατάθεση μιας πρότασης, όσο και για την συμμετοχή ενός μέλους σε οποιαδήποτε ψηφοφορία απαιτείται η κατοχή *κουπονιών διακυβέρνησης (governance tokens)* (Staff, 2022). Σε μια απλή υλοποίηση DAO, θα μπορούσε κάθε ένα από τα μέλη του οργανισμού να διαθέτει από ένα κουπόνι διακυβέρνησης. Ωστόσο, σε άλλες υλοποιήσεις, κάποια από τα μέλη μπορεί να διαθέτουν περισσότερα κουπόνια διακυβέρνησης από άλλα και άρα να έχουν μεγαλύτερη «ισχύ»/συμμετοχή στην διακυβέρνηση του οργανισμού. Τα κουπόνια διακυβέρνησης θα μπορούσαν να παρέχονται από το σύστημα σε κάθε μέλος που εισέρχεται. Εναλλακτικά, θα μπορούσαν να αποκτώνται μέσω της ενεργής συμμετοχής του μέλους στις δραστηριότητες της κοινότητας. Ένας τέτοιος μηχανισμός ενδέχεται να είναι συνδεδεμένος και με μία μετρική φήμης για το εύρος και την ποιότητα δραστηριότητας κάθε μέλους. Ο ορισμός των μετρικών φήμης στο IoTFeds καθώς και οι μέθοδοι υπολογισμού αυτών παρουσιάζονται στο παραδοτέο Π2.3. Τέλος, κάποιος DAO μπορεί να επιτρέπει στα μέλη του να επιλέγουν να αντιπροσωπεύονται από ένα άλλο μέλος στις ψηφοφορίες. Η υιοθέτηση οποιασδήποτε εκ των παραπάνω εναλλακτικών ορίζεται από τους κανόνες της εκάστοτε ομοσπονδίας και υλοποιείται μέσω έξυπνων συμβολαίων. Οι κανόνες των ομοσπονδιών IoTFeds και τα έξυπνα συμβόλαια που διασφαλίζουν την εφαρμογή τους παρουσιάζονται στις Ενότητες 3.3.1 και 3.3.2 αντίστοιχα.

Όσον αφορά την υλοποίηση του συστήματος, ένας DAO θα μπορούσε είτε να υλοποιηθεί εξ αρχής, με τα μέλη του να αναπτύσσουν τα απαραίτητα έξυπνα συμβόλαια, είτε αξιοποιώντας υπάρχουσες πλατφόρμες που παρέχουν DAO ως υπηρεσία (Faqir, Youssef, Arroyo, & Hassan, 2020), (Faqir-Rhazoui, Youssef, Arroyo, & Hassan, 2021). Παραδείγματα τέτοιων πλατφορμών αποτελούν οι Aragon[4], DAOhaus[5], DAOstack[6] και Colony[7]. Η πλατφόρμα Aragon παρέχει ένα στατικό πρότυπο έξυπνο συμβόλαιο για την δημιουργία DAO. Ωστόσο, δίνει και την δυνατότητα στους χρήστες της να προσαρμόσουν τους κανόνες και τις πολιτικές του DAO στις δικές τους ανάγκες προσθέτοντας ή αφαιρώντας ένα σύνολο από έξυπνα συμβόλαια. Αντίθετα, η πλατφόρμα DAOstack δεν προσφέρει την δυνατότητα προσαρμογής των συμβολαίων.

[4] https://aragon.org/
[5] https://daohaus.club/
[6] https://daostack.io/
[7] https://colony.io/

Συγκεκριμένα, όλοι οι DAOs στο DAOstack ακολουθούν το σύστημα ψηφοφορίας Holographic Consensus (Faqir-Rhazoui, Youssef, Arroyo, & Hassan, 2021) (υποστηρίζεται και από την πλατφόρμα Aragon) το οποίο περιγράφεται παρακάτω. Η πλατφόρμα DAOhaus βασίζεται στα συμβόλαια που έχουν αναπτυχθεί από το Moloch DAO (Soleimani, 2019) και επιτρέπει την δημιουργία νέων DAOs που επιθυμούν να ακολουθήσουν την λογική του Moloch DAO που επίσης περιγράφεται παρακάτω. Τέλος, η πλατφόρμα Colony καθιστά δυνατή την δημιουργία DAOs ή αλλιώς «αποικιών» όπου τα μέλη έχουν κοινό σκοπό ο οποίος μεταφράζεται σε δραστηριότητες που πρέπει να εκτελεστούν από τα μέλη. Τα μέλη αυξάνουν την φήμη και άρα επιρροή τους στον DAO ολοκληρώνοντας δραστηριότητες, και μέσω αυτού λαμβάνουν κουπόνια διακυβέρνησης.
Στην συνέχεια περιγράφουμε τους βασικούς μηχανισμούς ψηφοφορίας που χρησιμοποιούνται από τους υπάρχοντες DAOs στις προαναφερθείσες πλατφόρμες.

- **Holographic Consensus** (Faqir-Rhazoui, Youssef, Arroyo, & Hassan, 2021)**:** Το σύστημα ψηφοφορίας Holographic Consensus αποσκοπεί στην επίτευξη κλιμακωσιμότητας (scalability) ενός DAO ως προς τον αριθμό των μελών και αριθμό των ψηφοφοριών που μπορούν να υποστηριχθούν. Το Holographic Consensus ορίζει ότι μια απόφαση μπορεί να εγκριθεί από την απόλυτη πλειοψηφία των μελών του DAO, αλλά υπό συγκεκριμένες προϋποθέσεις η σχετική πλειοψηφία (δηλαδή > 51% αυτών που συμμετείχαν στην ψηφοφορία) είναι αρκετή. Συγκεκριμένα, εισάγει την έννοια των παικτών (stakers), οι οποίοι στοιχηματίζουν ειδικά *κουπόνια* υπέρ ή κατά μιας πρότασης που κατατίθεται από ένα μέλος. Οι stakers ενδέχεται να μην είναι μέλη του DAO, αλλά να ποντάρουν τα κουπόνια αποσκοπώντας σε κέρδη από αυτά τα πονταρίσματα (όταν μια πρόταση γίνει απόδεκτή). Αν το συνολικό ποντάρισμα για μία πρόταση φτάσει ένα συγκεκριμένο ύψος, τότε η σχετική πλειοψηφία είναι αρκετή για την λήψη απόφασης.
- **Μηχανισμοί Moloch DAO** (Soleimani, 2019): Για την κατάθεση μιας πρότασης στο Moloch DAO, το μέλος που καταθέτει την πρόταση θα πρέπει να καταβάλει κάποιο φόρο (tribute). Η κάθε πρόταση που τίθεται σε ψηφοφορία μπορεί να εγκριθεί από την σχετική πλειοψηφία, ωστόσο το Moloch DAO, εισάγει έναν μηχανισμό «οργισμένης αποχώρησης» (“rage quit”). Συγκεκριμένα, ένα μέλος του DAO που διαφωνεί με την απόφαση που λήφθηκε από μία ψηφοφορία έχει την δυνατότητα να αποχωρήσει από τον DAO αποσύροντας τους πόρους του εντός μια ορισμένης χρονικής περιόδου μετά την ψηφοφορία η οποία ονομάζεται «περίοδος χάριτος» (“grace period”). Αν ένα ποσοστό μεγαλύτερο του 33% των μελών του DAO επιλέξει να κάνει rage quit, τότε το αποτέλεσμα της ψηφοφορίας θεωρείτε άκυρο. Η έκδοση v2 του Moloch DAO δίνει την δυνατότητα απομάκρυνσης μελών, ενώ επιτρέπει την δυνατότητα κατάθεσης προτάσεων από μη μέλη όταν αυτές λαμβάνουν την υποστήριξη (sponsorship) κάποιου μέλους του DAO.
- **Ψηφοφορία Πεποίθησης - Conviction Voting (CV)** (Emmett, 2019): Η Ψηφοφορία Πεποίθησης λαμβάνει υπόψη την αθροιστική προτίμηση των μελών του DAO σε προτάσεις, η οποία εκφράζεται συνεχώς και όχι σε ένα παράθυρο ψηφοφορίας. Δηλαδή, ένα μέλος εκφράζει την προτίμηση (ψήφο) του σε μια πρόταση και όσο περισσότερο κρατά την ψήφο του σε αυτή την συγκεκριμένη πρόταση, τόσο αυξάνεται και η «πεποίθηση» (“conviction”) σε αυτή την πρόταση. Ένα μέλος μπορεί να αλλάξει την ψήφο του οποιαδήποτε στιγμή. Η ψήφος/προτίμηση των μελών δηλώνεται με κουπόνια διακυβέρνησης (governance tokens) περιορισμένου αριθμού ανά μέλος. Όσο περισσότερο κρατά κάποιος τα κουπόνια σε μία πρόταση, τόσο μεγαλύτερη είναι η πιθανότητα η πρόταση να φτάσει τον απαιτούμενο αριθμό ψήφων και να εγκριθεί.

Οι περιπτώσεις χρήσης των DAOs είναι πολλαπλές. Οι περισσότερες τρέχουσες υλοποιήσεις DAOs αφορούν πλατφόρμες που δραστηριοποιούνται στον τομέα Decentralized Finance (DeFi) (Zetzsche, A., Arner, & Buckley, 2020) και επικεντρώνονται στους τομείς συλλογικής χρηματοδότησης δράσεων (crowdfunding) (Jentzsch, 2016), στη συλλογική διαχείριση

κεφαλαίων και επενδύσεων κυρίως για την υποστήριξη έργων που αξιοποιούν την τεχνολογία blockchain (π.χ. Moloch DAO, LAO DAO[8], κτλ.). Ένας διαφορετικού τύπου DAO είναι ο Steemit[9], ένα κοινωνικό μέσο δικτύωσης που υλοποιείται σε τεχνολογία blockchain και σκοπός του είναι ο δίκαιος διαμοιρασμός της ανταμοιβής (σε STEEM) των δημιουργών περιεχομένου βάσει των αξιολογήσεων των χρηστών της πλατφόρμας.

Σε ένα γενικότερο πλαίσιο, οποιοσδήποτε τύπος DAO έχει σαν επίκεντρο την βάσει κανόνων, αποκεντρωμένη, συλλογική διαχείριση ενός συλλογικού ή κοινού αγαθού, με στόχο την μεγιστοποίηση του οφέλους για τα μέλη του οργανισμού. Στην περίπτωση του IoTFeds, κάθε ομοσπονδία Παρόχων (και ενδεχομένως Καταναλωτών) δεδομένων μπορεί να θεωρηθεί ως ένας DAO. Στόχος του IoTFeds DAO θα είναι η αποκεντρωμένη διαχείριση ομοσπονδιών Παρόχων/Καταναλωτών δεδομένων IoT για την διασφάλιση της υψηλής ποιότητας ομοσπονδιακών υπηρεσιών δεδομένων και το δίκαιο διαμοιρασμό των κερδών που δημιουργούνται από την παροχή αυτών των υπηρεσιών.

### *3.3.1 Κανόνες και Πολιτικές Ομοσπονδίας – IoTFeds DAO*

Σε αυτή την υπό-ενότητα ορίζουμε ένα σύνολο από κανόνες και πολιτικές για την διακυβέρνηση των IoTFeds DAOs. Συνεπώς οι κανόνες και πολιτικές που αναλύονται στην ενότητα αυτή αποτελούν τη βάση για την υλοποίηση και επέκταση των ομοσπονδιών στα πλαίσια του έργου IoTFeds. Οι κανόνες αυτοί παρέχονται ως ένα πρότυπο για την δημιουργία ενός νέου IoTFeds DAO επιτρέποντας ένα σύνολο από εναλλακτικές επιλογές. Κάποιοι από τους κανόνες είναι απαραίτητοι, ενώ κάποιοι άλλοι είναι προαιρετικοί. Οι κανόνες που συζητώνται παρακάτω ορίζονται κατά την δημιουργία της ομοσπονδίας, ενώ δύναται να αλλάξουν μετά την δημιουργία της μέσω συλλογικών διαδικασιών της ομοσπονδίας που επίσης. *(Το σύνολο των κανόνων είναι υπό διερεύνηση και αναμένεται να εμπλουτιστεί περαιτέρω στην διάρκεια του έργου.)*

***Τύπος ομοσπονδίας.*** *Αυτή η ομάδα κανόνων περιλαμβάνει τον ορισμό χαρακτηριστικών που έχουν να κάνουν με τους ρόλους που πρέπει να έχουν τα μέλη της ομοσπονδίας εντός αυτής, τον τύπο των υπηρεσιών δεδομένων που υποστηρίζονται, τις οντολογίες που χρησιμοποιούνται και τον βαθμός έκθεσης των υπηρεσιών της ομοσπονδίας.

- *Ρόλοι μελών*: Θεωρούμε δύο βασικές κατηγορίες ομοσπονδιών βάσει των ρόλων των μελών τους, την ομοσπονδία ***Παρόχων*** και την ***Μικτή*** ομοσπονδία. Σε μια ομοσπονδία Παρόχων όλα τα μέλη της ομοσπονδίας θα πρέπει να παρέχουν υπηρεσίες δεδομένων. Φυσικά, αυτοί οι Πάροχοι θα είναι ταυτόχρονα και εν δυνάμει Καταναλωτές των υπηρεσιών της ομοσπονδίας. Σε μια Μικτή ομοσπονδία κάποιο μέλος μπορεί να έχει μόνο το ρόλο του Καταναλωτή ομόσπονδων υπηρεσιών δεδομένων.
- *Βαθμός έκθεσης ομόσπονδων υπηρεσιών δεδομένων*: Θεωρούμε δύο βασικές κατηγορίες ομοσπονδιών βάσει του βαθμού έκθεσης των υπηρεσιών δεδομένων, την ***Κλειστή*** και την ***Υβριδική*** ομοσπονδία. Σε μια Κλειστή ομοσπονδία, όλες οι υπηρεσίες που συν-δημιουργούνται από κοινού από μέλη της ομοσπονδίας παρέχονται μέσω μιας *ομοσπονδιακής αγοράς (federated marketplace)* για κατανάλωση μόνο από τα μέλη της ομοσπονδίας. Η πολιτική που εφαρμόζεται για την διάθεση των ομόσπονδων υπηρεσιών ορίζεται από τους κανόνες αυτής της ομοσπονδιακής αγοράς (περιγράφεται στην συνέχεια της ενότητας). Μια Υβριδική ομοσπονδία, πέρα από την υποστήριξη της ομοσπονδιακής αγοράς, επιτρέπει την έκθεση των προϊόντων υπηρεσιών δεδομένων που δημιουργούνται συνεργατικά από τα μέλη της ομοσπονδίας σε μία *ανοικτή καθολική αγορά* (*open global marketplace*). Σε αυτή στην καθολική αγορά έχουν πρόσβαση όλοι οι χρήστες της πλατφόρμας IoTFeds και τα προϊόντα παρέχονται βάσει των πολιτικών που ορίζονται από αυτή. Σε κάθε περίπτωση η κατανάλωση μέσω της ομοσπονδιακής αγοράς θα έχει ευνοϊκότερους όρους, οπότε αναμένεται τα μέλη της ομοσπονδίας να προτιμούν την κατανάλωση προϊόντων μέσω της ομοσπονδιακής

[8] https://www.thelao.io/
[9] https://steemit.com/

αγοράς, ενώ η καθολική αγορά να επικεντρώνεται στην κατανάλωση υπηρεσιών από μη μέλη.

- *Συνδρομή μέλους*: Ορίζει αν τα μέλη πρέπει να καταβάλουν κάποια μηνιαία/ετήσια συνδρομή μέλους, καθώς και την τιμή αυτής σε *κουπόνια χρησιμότητας του IoTFeds*, δηλαδή *IoTFeds utility tokens*.Η έννοια των IoTFeds (utility) tokens εισήχθη στο παραδοτέο Π1.3 και αφορά κουπόνια τα οποία χρησιμοποιούνται εσωτερικά στο IoTFeds σαν *εναλλακτικό νόμισμα* για μικροπληρωμές που έχουν να κάνουν με την διαχείριση των ομοσπονδιών. Η χρήση αυτών των κουπονιών ως νόμισμα για την διενέργεια αγορών στις ομοσπονδιακές ή καθολική αγορές αποτελεί αντικείμενο μελέτης της ενότητας εργασίας ΕΕ3.
- *Τύπος υποστηριζόμενων υπηρεσιών*: Αυτός ο κανόνας ορίζει τον τύπο των δεδομένων που είναι σχετικά με τις υπηρεσίες στις οποίες επικεντρώνεται η ομοσπονδία. Θεωρώντας ότι το IoTFeds επιτρέπει την κατηγοριοποίηση των δεδομένων βάσει των υπηρεσιών ή των καθετοποιημένων τομέων που εξυπηρετούν (π.χ., Έξυπνη Πόλη, Βιομηχανία, Γεωργία, Ενέργεια, Υγεία, κτλ.), κάθε ομοσπονδία μπορεί να ορίσει το υποσύνολο των τομέων που εξυπηρετεί.
- Υποστηριζόμενες οντολογίες: Κάθε ομοσπονδία θα ορίζει την οντολογία/ες (.ttl) που θα χρησιμοποιήσει για την σημασιολογική διαλειτουργικότητα των πόρων δεδομένων που παρέχονται από τα μέλη της ομοσπονδίας. Για την συμβατότητα των οντολογιών (μοντέλο πληροφορίας) που επιλέγονται από κάθε ομοσπονδία με το Core Information Model (Βασικό Μοντέλο Πληροφορίας) του IoTFeds, ενδέχεται να απαιτείται η ανάπτυξη κάποιου μηχανισμού Mapping (αντιστοίχισης) από τα μέλη της ομοσπονδίας.

***Διακυβέρνηση Ομοσπονδίας.*** Αυτό το σύνολο των κανόνων περιλαμβάνει τον ορισμό των διαδικασιών που ακολουθούνται για την διακυβέρνηση της ομοσπονδίας, καθώς και τον τρόπο λήψης αποφάσεων για αυτές τις διαδικασίες λαμβάνοντας υπόψη ότι οι ομοσπονδίες είναι DAOs.

- *Επιτροπή Διακυβέρνησης*: Ορίζει το σύνολο των μελών της ομοσπονδίας που συμμετέχουν στην λήψη αποφάσεων σχετικά με την διακυβέρνηση της ομοσπονδίας. Οι δύο ακραίες περιπτώσεις είναι όλα τα μέλη της ομοσπονδίας να συμμετέχουν στην λήψη αποφάσεων ή ένα μόνο μέλος ορισμένο σαν «διαχειριστής» να λαμβάνει τις αποφάσεις για όλους. Φυσικά, θα μπορούσε ένα υποσύνολο μελών να είναι υπεύθυνο για την λήψη αποφάσεων, π.χ. μόνο τα μέλη που παρέχουν υπηρεσίες στην περίπτωση Μικτής ομοσπονδίας. Τα μέλη που συμμετέχουν στην Επιτροπή Διακυβέρνησης, όπως και κάθε κανόνας την ομοσπονδίας, μπορούν να αλλάξουν μέσω της διαδικασίας λήψης αποφάσεων που ορίζεται παρακάτω.
- *Κουπόνια Διακυβέρνησης (Governance Token)*: Τα μέλη της Επιτροπής Διακυβέρνησης διαθέτουν κουπόνια διακυβέρνησης (governance tokens) τα οποία μπορούν να αξιοποιήσουν στην διαδικασία λήψης αποφάσεων. Κάθε μέλος της Επιτροπής Διακυβέρνησης μπορεί να διαθέτει ένα ή πολλαπλά κουπόνια διακυβέρνησης. Στην τρέχουσα έκδοση των κανόνων θεωρούμε ότι κάθε μέλος διαθέτει ένα κουπόνι διακυβέρνησης. Ωστόσο, σε μελλοντικές υλοποιήσεις μπορούμε να θεωρήσουμε ότι κάθε μέλος διαθέτει έναν αριθμό κουπονιών που είναι ανάλογος με το αριθμό των πόρων που διαθέτει στην ομοσπονδία ή την «φήμη» του. Επίσης, ένα μέλος της Επιτροπής Διακυβέρνησης θα έχει την δυνατότητα να εξουσιοδοτήσει ένα άλλο μέλος ώστε να συμμετέχει στην λήψη αποφάσεων εκ μέρους του (governance token delegation).
- *Τύποι Προτάσεων*: Ο παρών κανόνας ορίζει τους τύπους τον προτάσεων που μπορούν να κατατεθούν προς εξέταση από την Επιτροπή Διακυβέρνησης μιας ομοσπονδίας τα μέλη ή μη μέλη της. Μια ομοσπονδία μπορεί να επιτρέπει όλους τους παρακάτω τύπους προτάσεων ή ένα υποσύνολο από αυτές.

  - Πρόσκληση νέου μέλους στην ομοσπονδίας. (Πρόταση από μέλος)
  - Αίτημα συμμετοχής στην ομοσπονδία. (Πρόταση από μη μέλος)
  - Αίτημα απομάκρυνσης μέλους από την ομοσπονδία. (Πρόταση από μέλος)
  - Αίτημα αλλαγής κανόνα ή πολλαπλών κανόνων της ομοσπονδίας. (Πρόταση από μέλος)
  - Αίτημα εγγραφής ομοσπονδιακού προϊόντος στην καθολική αγορά του IoTFeds (Πρόταση από μέλος)
- *Διαδικασία Λήψης Αποφάσεων*: Όλες οι αποφάσεις της ομοσπονδίας λαμβάνονται έπειτα από ψηφοφορία της Επιτροπής Διακυβέρνησης. Η διαδικασία λήψης αποφάσεων που υιοθετείται θα μπορεί να διαφέρει από ομοσπονδία σε ομοσπονδία και να εμπνέεται από ένα από τα μοντέλα τύπου Holographic Consensus, Moloch DAO, κτλ., που παρουσιάστηκαν νωρίτερα. Στα πρώτα στάδια της υλοποίησης του έργου, το IoTFeds θα παρέχει ένα *στατικό πρότυπο* της διαδικασίας ψηφοφορίας το οποίο θα δίνει την δυνατότητα σε κάθε ομοσπονδία DAO να παραμετροποιήσει συγκεκριμένα πεδία του. Μελλοντικά, σκοπεύουμε να διευρύνουμε τους τύπους διαδικασιών λήψης αποφάσεων που υποστηρίζονται από το IoTFeds, δίνοντας την δυνατότητα στις ομοσπονδίες να ορίσουν οποιαδήποτε διαδικασία επιθυμούν.

  Το στατικό πρότυπο της διαδικασία λήψης αποφάσεων θα ορίζει τα παρακάτω:
  - *Φόρος πρότασης:* Ορίζει την ποσότητα *IoTFeds tokens* που απαιτούνται να καταβληθούν από το μέλος (ή μη μέλος) που καταθέτει μια πρόταση.
  - *Τύπος Πλειοψηφίας:* Ορίζει αν απαιτείται απόλυτη, σχετική ή κάποια άλλη ειδική πλειοψηφία για την λήψη μιας απόφασης. Για να μπορεί να υποστηριχθεί οποιοδήποτε είδος πλειοψηφίας, ο παρών κανόνας ορίζεται από δύο πεδία, το ποσοστό ψήφων που απαιτείται για την κατοχύρωση μια απόφασης και την βάση στην οποία εξετάζεται η πλειοψηφία. Αυτή η βάση μπορεί να είναι είτε το σύνολο των μελών την Επιτροπής Διακυβέρνησης ή στο σύνολο των μελών που συμμετείχαν στην ψηφοφορία.

***Κανόνες Διασφάλισης Ποιότητας.*** Αυτό το σύνολο των κανόνων επικεντρώνεται στον ορισμό του τύπου των μετρικών ποιότητας υπηρεσίας που λαμβάνονται υπόψη από την ομοσπονδία, καθώς και τις τιμές που θα πρέπει να επιτυγχάνουν τα μέλη της ομοσπονδίας ως προς αυτές τις μετρικές.

- *Μετρικές ποιότητας*: Η ποιότητα της υπηρεσίας που παρέχει ένα μέλος μπορεί να αξιολογείται είτε βάσει των αντικειμενικών μετρικών Ποιότητας Υπηρεσίας (Quality of Service - QoS) που υποστηρίζονται από το σύστημα IoTFeds, είτε βάσει της φήμης του κάθε μέλους που προκύπτει από τον συνυπολογισμό των αντικειμενικών μετρικών με την υποκειμενική αξιολόγηση των υπηρεσιών από τους Καταναλωτές. Θεωρώντας ότι και οι δύο διαστάσεις μετρικών (αντικειμενικές και υποκειμενικές) λαμβάνουν τιμές σε ένα κοινό εύρος τιμών (π.χ. [1, 5]), οποιαδήποτε μετρική ποιότητας μπορεί να επιλεγεί ορίζοντας τα επιθυμητά βάρη-ποσοστά (0-100%) βάσει των οποίων κάθε μία από τις δύο διαστάσεις θα λαμβάνεται υπόψη. Οι μετρικές ποιότητας θα αναλυθούν σε επόμενη δράση (Δ2.5).
- *Επίπεδο Ποιότητας*: Ορίζει την ελάχιστη τιμή ποιότητας για τις υπηρεσίες που παρέχονται από την ομοσπονδία συνολικά αλλά και κάθε ένα από τα μέλη.
- *Διαχείριση Αστοχίας Ποιότητας*: Ορίζει αν υπάρχει μηχανισμός διαχείρισης αστοχίας ποιότητας. Δηλαδή, αν "πυροδοτείται" μια διαδικασία λήψης απόφασης που έχει να κάνει με την απομάκρυνση τους μέλους όταν αυτό δεν επιτυγχάνει την απαιτούμενη ποιότητα.

***Κανόνες και Πολιτικές Ομοσπονδιακής Αγοράς.*** Το σύνολο αυτών των κανόνων και πολιτικών ορίζει την λειτουργία της ομοσπονδιακής αγοράς, δηλαδή την διαδικασία δημιουργίας

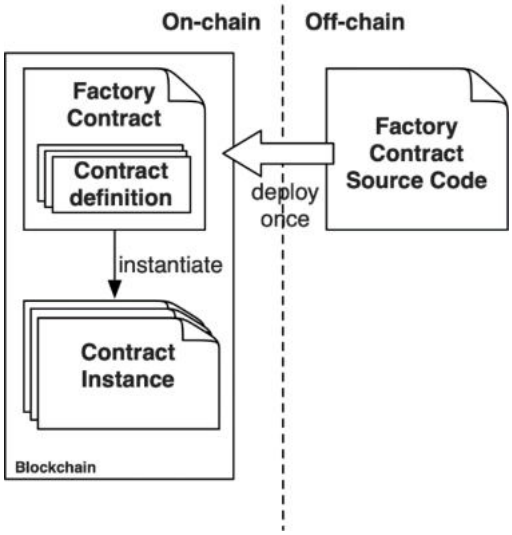
ομοσπονδιακού προϊόντος, την πολιτική τιμολόγησης των προϊόντων και διαμοιρασμού των κερδών, καθώς και το νόμισμα που χρησιμοποιείται για τις συναλλαγές.

- *Πολιτική Τιμολόγησης*: Αυτός ο κανόνας ορίζει πως τιμολογούνται τα προϊόντα δεδομένων που καταναλώνονται μέσω της ομοσπονδιακής αγοράς. Συγκεκριμένα, ορίζει βάσει ποιου μοντέλου θα γίνεται η τιμολόγηση, π.χ. δωρεάν, πληρωμή-ανά-συναλλαγή (δηλαδή αγορά προϊόντος), πληρωμή-ανά-χρήση, κτλ. Η δωρεάν διάθεση προϊόντων έχει νόημα για ομοσπονδίες που εφαρμόζουν μια πολιτική συνδρομής μέλους. Οι πολιτικές τιμολόγησης αποτελούν μέρος της μελέτης της ενότητας εργασίας ΕΕ3 και θα αναλυθούν στο παραδοτέο Π3.2.
- *Δημιουργία ομοσπονδιακού προϊόντος*: Ο κανόνας αυτός ορίζει τον τρόπο με τον οποίο δημιουργείται ένα προϊόν στην ομοσπονδιακή αγορά. Το IoTFeds θα παρέχει δύο εναλλακτικούς τρόπους: *(i)* την *«χειροκίνητη» σύνθεση προϊόντος (packaging)* από ένα μέλος της ομοσπονδίας ή *(ii)* την *«αυτοματοποιημένη» σύνθεση προϊόντος (matching)* μέσω ενός μηχανισμού ταιριάσματος της ζήτησης για υπηρεσίες δεδομένων και της προσφοράς πόρων δεδομένων στην ομοσπονδία.
    - *Χειροκίνητη σύνθεση προϊόντος (packaging):* Στην περίπτωση του packaging, κάθε μέλος της ομοσπονδίας με τον ρόλο του Παρόχου δεδομένων έχει την δυνατότητα να εγγράψει πηγές δεδομένων στην ομοσπονδιακή αγορά. Από την άλλη κάθε μέλος της ομοσπονδίας με τον ρόλο του Καταναλωτή έχει την δυνατότητα αναζητήσει και να συνδυάσει πηγές δεδομένων (από διαφορετικούς Παρόχους) για την δημιουργία ενός ομοσπονδιακού προϊόντος που ο ίδιος θα καταναλώσει. Η πολιτική αυτή είναι συμβατή για τις Κλειστές αλλά και για τις Υβριδικές ομοσπονδίες που επιπρόσθετα επιθυμούν να εκθέσουν τα ομοσπονδιακά προϊόντα τους στην ανοιχτή καθολική αγορά του IoTFeds.
    - *Αυτοματοποιημένη σύνθεση προϊόντος (matching):* Στην περίπτωση του matching, ομοίως με την προηγούμενη περίπτωση, τα μέλη εγγράφουν πηγές δεδομένων στην ομοσπονδιακή αγορά. Ένας Καταναλωτής δεν θα έχει την δυνατότητα να αναζητήσει και να επιλέξει πηγές δεδομένων. Αντίθετα, θα καταθέτει ένα αίτημα για μια υπηρεσία δεδομένων δηλώνοντας συγκεκριμένες απαιτήσεις ως προς τον τύπο της υπηρεσίας, το επίπεδο ποιότητας, τη γεωγραφική κάλυψη, κτλ., που επιθυμεί. Στην συνέχεια, μια αυτοματοποιημένη διαδικασία της ομοσπονδιακής αγοράς θα αποφασίζει ποιες πηγές δεδομένων θα αντιστοιχίζονται σε ποια αιτήματα υπηρεσιών.

  Οι δύο ανωτέρω πολιτικές θα μελετηθούν περαιτέρω στα πλαίσια της ενότητας εργασίας ΕΕ3 και θα αναλυθούν στο παραδοτέο Π3.2. Σε κάθε μία από τις δύο πολιτικές, και αν δεν ακολουθείται η δωρεάν πολιτική τιμολόγησης, κατά την εγγραφή μιας πηγής δεδομένων ένας Πάροχος θα πρέπει να αναθέτει μια ζητούμενη τιμή με βάση την πολιτική τιμολόγησης που έχει οριστεί.
- *Πολιτική Διαμοιρασμού κερδών*: Αυτός ο κανόνας ορίζει πώς διαμοιράζονται τα κέρδη στα μέλη της ομοσπονδίας που συνεισφέρουν στην παροχή των προϊόντων. Οι δύο εναλλακτικές που ορίζονται είναι ο διαμοιρασμός *βάσει τιμής εγγραφής* και ο διαμοιρασμός βάσει *συνολικής συνεισφοράς*. Στην πρώτη περίπτωση, η τιμή μιας πηγής δεδομένων που έχει ορισθεί κατά την εγγραφή μιας πηγής δεδομένων αποδίδεται στον Πάροχο όταν αυτή πηγή συμπεριληφθεί σε ένα προϊόν. Η δεύτερη περίπτωση, είναι συμβατή για ομοσπονδίες που παρέχουν τα προϊόντα τους δωρεάν αλλά απαιτούν μια συνδρομή μέλους από τα μέλη τους. Σε αυτή την περίπτωση οι συνδρομές των μελών διαμοιράζονται στα μέλη της ομοσπονδίας βάσει της συνολικής συνεισφοράς τους στην παροχή προϊόντων. Σε αυτή την περίπτωση οι περίοδοι συνδρομών θα πρέπει να συνδυαστούν με τις περιόδους εκκαθάρισης στην ομοσπονδία. Οι παραπάνω πολιτικές θα μελετηθούν περαιτέρω στο παραδοτέο Π3.2.

### 3.3.2 *Νόμισμα*: Ο κανόνας αυτός ορίζει αν οι συναλλαγές στην ομοσπονδιακή αγορά θα εκτελούνται σε IoTFeds tokens ή σε παραστατικό χρήμα (fiat money).Έξυπνα Συμβόλαια για την Διαχείριση των Ομοσπονδιών - IoTFeds DAOs

Στην αρχή αυτής της ενότητας περιγράφουμε τη θεωρία πάνω στην οποία βασίστηκε η υλοποίηση για τη διαχείριση των ομοσπονδιών στο IoTFeds και στη συνέχεια περιγράφουμε τη διαδικασία που ακολουθήσαμε. Σύμφωνα με αυτή τη θεωρία, δημιουργείται ένα έξυπνο συμβόλαιο ανά ομοσπονδία, ωστόσο μια υλοποίηση που επιτυγχάνει αυτή την συνολική λειτουργικότητα με πολλαπλά έξυπνα συμβόλαια είναι επίσης δυνατό να εφαρμοστεί.
Στην προηγούμενη υποενότητα περιγράψαμε το σύνολο από τους κανόνες και πολιτικές που χαρακτηρίζουν μια ομοσπονδία στο IoTFeds. Κατά την δημιουργία μιας ομοσπονδίας το σύνολο αυτών των κανόνων και πολιτικών, καθώς και των άλλων λειτουργιών της ομοσπονδίας πρέπει να αρχικοποιηθούν με βάση τα χαρακτηριστικά της εκάστοτε ομοσπονδίας. Σύμφωνα με την λογική του εργοστασίου παραγωγής έξυπνων συμβολαίων (smart contract factory pattern[10]), η οποία παρέχεται στα πλαίσια του Ethereum, δημιουργείται ένα πρότυπο (template) έξυπνο συμβόλαιο (factory contact) βάσει του οποίου μπορούν να δημιουργηθούν πολλαπλά έξυπνα συμβόλαια. Η παραπάνω λογική παρουσιάζεται στην Εικόνα 9. Το factory contract καλείται κατά την δημιουργία μιας ομοσπονδίας και δημιουργεί το έξυπνο συμβόλαιο (contract instance) για την διαχείριση αυτής της ομοσπονδίας.




*Εικόνα 9: Smart contract factory pattern*[7].

***Factory contract:*** Το factory contact μπορεί καλείται κατά την δημιουργία μιας ομοσπονδίας. Οι χρήστες δεν αλληλεπιδρούν απευθείας με το factory contract. Επίσης, υπάρχει η δυνατότητα ανανέωσης του factory contact αν χρειαστεί. Όπως κάθε έξυπνο συμβόλαιο, έτσι και το factory contract αποτελείται από δεδομένα και συναρτήσεις.
***Contract Instance:*** Το contract instance είναι αποτέλεσμα της κλήσης της συνάρτησης *create federation* στο factory contract και στην ουσία αποτελεί το συμβόλαιο που είναι υπεύθυνο για την συνολική διαχείριση μιας ομοσπονδίας.
Όπως έχει ήδη αναφερθεί πιο πάνω, στο IoTFeds, χρησιμοποιούμε το Hyperledger Fabric, στο οποίο η παραπάνω λογική δεν βρίσκει εφαρμογή. Η ύπαρξη πολλαπλών συμβολαίων για την αναπαράσταση κάθε μίας ομοσπονδίας ξεχωριστά, θα δημιουργούσε θέματα απόδοσης και συντήρησης του ίδιου του δικτύου αλλά και του κώδικα. Η ψηφιακή πληροφορία στο HLF εγγράφεται στο κατανεμημένο καθολικό με τη μορφή key-value. Επομένως, κάθε ομοσπονδία αρχικοποιείται στο καθολικό ως αντικείμενο (object) και περιέχει όλες τις απαραίτητες πληροφορίες που διέπουν τη λειτουργία της, για παράδειγμα τα μέλη της και οι κανόνες της. Όλες οι ομοσπονδίες, όμως, διαχειρίζονται από συγκεκριμένες συναρτήσεις, οι οποίες διαφοροποιούν τη λειτουργία τους με βάση τους κανόνες της εκάστοτε ομοσπονδίας, διατηρώντας ίδια τη γενική λειτουργία της κάθε συνάρτησης.

[10] https://research.csiro.au/blockchainpatterns/general-patterns/contract-structural-patterns/factory-contract/

Σύμφωνα λοιπόν με τα παραπάνω, η διαχείριση μιας ομοσπονδίας ως DAO επιτυγχάνεται μέσω ενός έξυπνου συμβολαίου που εκτελείται στο BaaS και το οποίο υλοποιεί τη λογική του DAO καθώς και όλες τις απαραίτητες λειτουργίες για την διαχείριση μιας IoTFeds ομοσπονδίας. Για τις διαδικασίες που αφορούν τη διεξαγωγή μιας ψηφοφορίας στα πλαίσια μιας ομοσπονδίας, έχει αναπτυχθεί ένα ξεχωριστό smart contract, που περιέχει τις σχετικές συναρτήσεις και επικοινωνεί με το smart contract που είναι υπεύθυνο για τη διαχείριση των ομοσπονδιών.
Οι βασικές λειτουργίες για τη διαχείριση των ομοσπονδιών στο IoTFeds είναι οι ακόλουθες:

- Εγγραφή ομοσπονδίας
- Ανάκτηση πληροφοριών ομοσπονδίας
- Ανάκτηση πληροφοριών όλων των ομοσπονδιών
- Αποχώρηση από ομοσπονδία
- Διαγραφή ομοσπονδίας

Οι βασικές λειτουργίες για τη διαχείριση των ψηφοφοριών στις IoTFeds ομοσπονδίες είναι:

- Αίτημα προσθήκης νέου μέλους σε ομοσπονδία
- Αίτημα αφαίρεσης μέλους από ομοσπονδία
- Αίτημα αλλαγής κανόνων ομοσπονδίας
- Πρόσβαση σε περιγραφή ψηφοφορίας
- Καταγραφή ψήφου

Η περιγραφή των endpoints που επιτυγχάνουν τις παραπάνω λειτουργίες και υλοποιήθηκαν στα πλαίσια του έργου βρίσκεται στο κεφάλαιο 5.
Τα έξυπνα συμβόλαια που διαπραγματεύονται την εξασφάλιση ποιότητας υπηρεσίας αλλά και την λειτουργικότητα της αγοράς βάσει των ορισμένων πολιτικών, θα μελετηθούν σε σχετικά μελλοντικά παραδοτέα Π2.2 «Αναφορά Μηχανισμών Διαφήμισης, Ανακάλυψης και Ασφαλούς Πρόσβασης σε Πόρους IoT», Π3.1 «Αναφορά Μηχανισμών Καταγραφής και Ελέγχου Συναλλαγών», Π3.2 «Αναφορά Μηχανισμών Αγοράς IoT», αντίστοιχα.

### 3.3.3 Τεχνικό σχήμα κανόνων και πολιτικών ομοσπονδίας

Για τους κανόνες και τις πολιτικές μιας ομοσπονδίας στα πλαίσια του IoTFeds όπως περιγράφηκαν παραπάνω, υπάρχει η ανάγκη να περιγραφούν και με έναν πιο τεχνικό τρόπο, ο οποίος είναι απαραίτητος για την υλοποίηση και την επικοινωνία μεταξύ των επιμέρους στοιχείων, και πιο συγκεκριμένα μεταξύ της επεκταμένης γραφικής διεπαφής χρήστη Administration GUI και του BaaS. Ακολουθεί μια πρώτη προσέγγιση αυτών των κανόνων και πολιτικών σε μορφή JSON. Καθώς οι κανόνες και οι πολιτικές των ομοσπονδιών είναι υπό διερεύνηση, είναι πιθανό η προτεινόμενη δομή του JSON να αλλάξει και οι τελικές τροποποιήσεις θα παρουσιαστούν σε επόμενες σχετικές δράσεις (στην Δ2.5/Π2.3 για τις μετρικές φήμης).

```
{
  "IoTFedsRules":{
    "FedTypeRules": {
      "Type": "Providers/Mixed",
      "DataAvailability": "Closed/Hybrid",
      "ServiceType": "Energy, Health, Environment",
      "SupportedOntologies": [String]
    },

    "FedGov": {
      "BoardGov": ["user1", "user2"],
                  "Proposals": ["InviteMember", "JoinRequest", "RequestRemove", "ChangeRule",
       "DeleteFed"],
    "VoteRules": {
      "Tokens": 50,
```

```
        "Type": {
            "ApprovalPercentage": 100/75/50,
            "Base": "Board/Voters"
          }
        }
      },

      "QualityAssurance": {
        "Metrics": {
          "ObjectiveMetricsPercentage": 60,
          "SubjectiveMetricsPercentage": 40,
          "Reputation": {
            "MinValueFed": 5
          },
          "Underperformance": "None/RequestRemove"
        }

       },

       "FedMarketplace": {
        "ChargePolicy": "Free/PerProduct/PerUsage/Subscription",
        "FedProduct": "Packaging/Matching/",
        "ProfitPolicy": "PerSource/ Contribution/Auction",
        "Coin": "IoTFeds/Euro/ Both"
      }

    }
}
```

# 4 Μοντέλα πληροφορίας και σημασιολογική διαλειτουργικότητα

Η ενότητα αυτή εστιάζει στη σημασιολογική διαλειτουργικότητα και τις σχετικές τεχνολογίες εξετάζοντας σε μεγαλύτερη λεπτομέρεια τα τελευταία πρότυπα μοντέλων πληροφορίας και οντολογιών στο Διαδίκτυο των Πραγμάτων και την Έξυπνη Πόλη. Τέλος, παρουσιάζει τη προσέγγιση που ακολουθείται και το μοντέλο πληροφορίας για την επίτευξη της διαλειτουργικότητας στις IoTFeds ομοσπονδίες.

## *4.1 Εισαγωγή στη σημασιολογία και την διαλειτουργικότητα*

Η διαλειτουργικότητα αφορά στην δυνατότητα των διαφόρων συσκευών και εφαρμογών να επικοινωνούν αρμονικά μεταξύ τους και να ανταλλάσσουν πληροφορίες. Υπάρχουν τρία διαφορετικά είδη διαλειτουργικότητας:

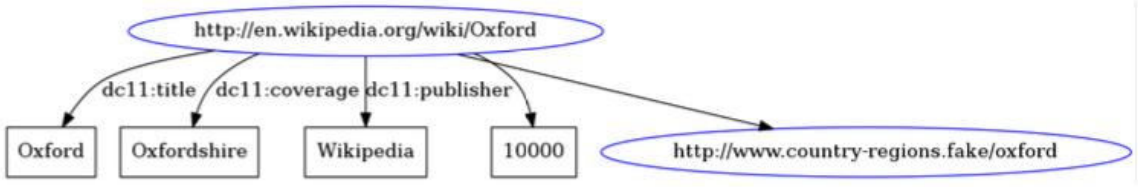
- Τεχνική διαλειτουργικότητα (technical interoperability), η οποία αφορά την πραγματική μεταφορά της πληροφορίας (bits μέσω ενσύρματου ή ασύρματου μέσου) ώστε αυτή να γίνεται με ομοιογενή και αποτελεσματικό τρόπο.
- Συντακτική διαλειτουργικότητα (syntactic interoperability), η οποία εξετάζει και διασφαλίζει την χρήση μιας κοινής γλώσσας (σύμβολα και έννοιες) για την επικοινωνία μεταξύ συσκευών και εφαρμογών.
- Σημασιολογική διαλειτουργικότητα (semantic interoperability), η οποία αφορά στην διασφάλιση της ακριβούς έννοιας και σημασίας των ανταλλασσόμενων πληροφοριών ώστε να είναι κατανοητή από οποιαδήποτε εφαρμογή.

Το παραδοτέο αυτό εστιάζει στη σημασιολογική διαλειτουργικότητα και τις σχετικές τεχνολογίες εξετάζοντας σε μεγαλύτερη λεπτομέρεια τα τελευταία πρότυπα μοντέλων πληροφορίας και οντολογιών στο Διαδίκτυο των Πραγμάτων.

## *4.2 Τεχνολογίες για σημασιολογική διαλειτουργικότητα*

Παρακάτω παρουσιάζονται τα πιο σημαντικά εργαλεία και τεχνολογίες για την σύνταξη μοντέλων πληροφοριών που διασφαλίζουν την σημασιολογική διαλειτουργικότητα.

***RDF***: Το Resource Description Framework (RDF) ((W3C), 2014) αποτελεί το βασικό μοντέλο αναπαράστασης δομών και προτύπων στον τομέα του σημασιολογικού ιστού (semantic web). Η βασική δήλωση ενός RDF (RDF statement) επιτυγχάνεται μέσω του τρίπτυχου «υποκείμενο-κατηγορούμενο-αντικείμενο» (“subject – predicate – object”). Τα δεδομένα ή οι πληροφορίες στο RDF συχνά απεικονίζονται σαν σημασμένα, κατευθυνόμενα γραφήματα όπως στην παρακάτω εικόνα**,** όπου τα οβάλ και τετράγωνα σχήματα αντιπροσωπεύουν πόρους (*resources)* και αλφαβητικά *(literals)* αντιστοίχως και τα σημασμένα βέλη αντιπροσωπεύουν σχέσεις (*relations)* ή κατηγορούμενα (*predicates).*




*Εικόνα 10: Παράδειγμα ενός RDF statement σαν γράφημα.*

Το μοντέλο RDF είναι ένα μοντέλο μεταδεδομένων (meta-data model), το οποίο έχει την δυνατότητα υποστήριξης πολλαπλών τρόπων σειριοποίησης (serialization), όπως RDF/XML (W3C, RDF 1.1 XML Syntax, 2014) , N-Triples (W3C, RDF 1.1 N-Triples, 2014), Turtle (W3C, RDF 1.1 Turtle, 2014), RDFa (W3C, RDFa 1.1 Primer - Third Edition, 2015), Notation3 (N3) (W3C, Notation3 (N3): A readable RDF syntax, 2011) και JSON-LD (W3C, JSON-LD 1.1,

2020). Τέλος η δομή των δεδομένων ορίζεται με το RDF Schema (RDFS) ενώ η γλώσσα OWL περιγράφει τις σημασιολογικές σχέσεις.

- ***RDF Schema*** (W3C, RDF Schema 1.1, 2014): Το σχήμα αυτό αποτελεί το λεξικό των RDF δεδομένων, το οποίο παρέχει το σημασιολογικό πλαίσιο στα RDF δεδομένα. Το RDF Schema ορίζει τις έννοιες της κλάσης (*class)*, ιδιότητας (*property)*, του τομέα (*domain)* και του εύρους (*range)* όπως φαίνονται στους παρακάτω πίνακες (Πίνακας 1 και Πίνακας 2).

*Πίνακας 1: RDFS οντότητες*

| Class Name | Description |
|---|---|
| rdfs:Resource | all things declared by RDF are resources |
| rdfs:Class | describes the concepts of a class |
| rdfs:Literal | describes the concept of a literal |
| rdfs:Property | the class for properties |

*Πίνακας 2: RDFS ιδιότητες.*

| Property Name | Description |
|---|---|
| rdfs:domain | defines to which subjects a property applies |
| rdfs:range | defines the set of values a property can accept |
| rdf:type | used to state that a resource is an instance of a class |
| rdfs:subClass Of | defines one class as a subclass of another class |
| rdfs:label | provides a human-readable version of resource's name |
| rdfs:comment | provides a human-readable description of resource |
| rdfs:seeAlso | link to another resource that might provide additional information |

- **Web Ontology Language (OWL):** Η OWL αποτελεί μια σημασιολογική γλώσσα για το (δια)δίκτυο, με στόχο την αναπαράσταση πολύπλοκης γνώσης και συσχετίσεων μεταξύ των πραγμάτων (things). Τα OWL μοντέλα είναι μοντέλα οντολογιών (ontologies) τα οποία συντάσσονται χρησιμοποιώντας RDF statements. Οι πιο σημαντικές δηλώσεις της OWL γλώσσας παρουσιάζονται παρακάτω.

| Property Name | Description |
|---|---|
| owl:Class | used for all class descriptions and is defined as a subclass of rdfs:Class. |
| owl:NamedIndividual | used to declare named (in contrast to anonymous), individual entities. |
| owl:equivalentClass | used to declared that 2 classes are equivalent. |
| owl:equivalentProperty | used to declare equivalent properties of classes. |

Η OWL γλώσσα εμπερικλείει όλες τις δηλώσεις της RDFS, προσθέτοντας επιπλέον δηλώσεις οντοτήτων όπως owl:equivalentClass, owl:equivalentProperty καθώς και επιπλέον δηλώσεις που αφορούν το μετα-μοντέλο, όπως owl:imports (που χρησιμοποιείται για να εισάγει ένα ήδη υπάρχον μοντέλο πληροφορίας) ή το owl:versionInfo (που χρησιμοποιείται για να δηλώσει την έκδοση ενός μοντέλου πληροφορίας).
Στο παρακάτω παράδειγμα δίνεται η περιγραφή δήλωσης μια νέας RDF οντότητας με όνομα SensorX. Στην αρχή της δήλωσης, δίνονται οι απαραίτητοι σύνδεσμοι για οντότητες που ορίζονται μέσα στο RDF και που δηλώνονται με βάση το πρόθεμά τους (prefix).

```
@prefix : <http://www.example.com/ontology/sensorX#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix core: <http://www.symbiote-h2020.eu/ontology/core#> .
:sensorX a owl:NamedIndividual ;
a core:StationarySensor ;
core:name "name of sensor X" ;

Όπως περιγράψαμε και πιο πάνω η οντότητα sensorX ακολουθεί την σύμβαση δήλωσης μιας RDF οντότητας με βάση το τρίπτυχο «subject-predicate-object». Συνεπώς η δήλωση "sensorX a owl:NamedIndividual", δηλώνει ότι το υποκείμενο sensorX ανήκει στην owl:NamedIndividual οντότητα, δηλαδή αποτελεί μια επώνυμη περίπτωση μίας οντότητας. Οι επόμενες δηλώσεις μας δίνουν επιπρόσθετες πληροφορίες σε σχέση με την νέα, δηλωθείσα οντότητα. Για παράδειγμα, η δήλωση "sensorX a core:StationarySensor" δηλώνει ότι το υποκείμενο sensorX αποτελεί έναν StationarySensor, τη δήλωση του οποίου θα βρούμε στο core (prefix)[11]. Και στις δύο περιπτώσεις το predicate a είναι απλά ένα κατηγορούμενο χωρίς ιδιαίτερη σημασία για αυτό και του αποδίδεται τυχαία το a. Τέλος, η δήλωση "sensorX core:name "name of Sensor X" δηλώνει ότι η οντότητα sensor, έχει την ιδιότητα (κατηγορούμενο) core:name (το name προέρχεται από το μοντέλο πληροφορίας core) και παίρνει σαν τιμή το "name of Sensor X".

## *4.3 Σημασιολογική διαλειτουργικότητα στον τομέα του Διαδικτύου των Πραγμάτων (IoT)*

Το Διαδίκτυο των Πραγμάτων (Internet of Things - IoT) ορίζεται ως ένα δίκτυο επικοινωνίας πληθώρας διασυνδεδεμένων συσκευών (φυσικών ή εικονικών) καθώς και κάθε αντικειμένου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων.

Στο τωρινό τοπίο στο Διαδίκτυο των Πραγμάτων επικρατούν πληθώρα καθέτων IoT silos, τα οποία σπανίως είναι διασυνδεδεμένα. Ωστόσο, γίνονται προσπάθειες για να επιτευχθεί γενική διαλειτουργικότητα μεταξύ των διαφορετικών συστημάτων με την υιοθέτηση των διαφορετικών επιπέδων διαλειτουργικότητας (τεχνικής, συντακτικής και σημασιολογικής διαλειτουργικότητας) που περιγράφηκαν παραπάνω. Προς την κατεύθυνση αυτή, διάφοροι διεθνείς οργανισμοί ασχολούνται με την επίτευξη και προώθηση τυποποιήσεων (Standard Developing Organizations - SDOs) στα πλαίσια εφαρμογής του IoT, όπως ο World Wide Web Consortium (W3C[12]), ο One Machine-to-Machine Partnership (oneM2M[13]), ο European Telecommunications Standards Institute (ETSI[14]) και ο Open Geospatial Consortium (OGC[15]).

Συγκεκριμένα, ο οργανισμός τυποποίησης W3C αναπτύσσει πρότυπα και οντολογίες ανοιχτές προς όλους, όπως η οντολογία Semantic Sensor Network ontology (SSN) (W3C, Semantic Sensor Network Ontology, 2017) και η Sensor, Observation, Sample and Actuators (SOSA), για την μακροπρόθεσμη ανάπτυξη του Διαδικτύου των Πραγμάτων, καθώς και του Δικτύου των Πραγμάτων (Web of Things). Οι οντολογίες SSN και SOSA αποσκοπούν στην αναπαράσταση των οντοτήτων, των σχέσεων και των δραστηριοτήτων που εμπλέκονται στην ανίχνευση (sensing), τη δειγματοληψία (sampling) και την ενεργοποίηση (actuation). Η οντολογία SSN περιγράφει τους αισθητήρες (sensors), τις παρατηρήσεις τους, τις εμπλεκόμενες διαδικασίες, τα χαρακτηριστικά που μελετώνται, τα δείγματα που χρησιμοποιήθηκαν και τις ιδιότητες που παρατηρούνται, καθώς και τους ενεργοποιητές (actuators). Το SSN συμπεριλαμβάνει την αυτοτελή οντολογία SOSA για τις στοιχειώδεις κλάσεις και τις ιδιότητές τους. Το OGC υλοποιεί πρότυπα όπως το SensorThings API[16], τα οποία χρησιμοποιούνται αποκλειστικά για την αναπαράσταση οντοτήτων στο πλαίσιο εφαρμογών του IoT παρέχοντας έναν ανοιχτό,

[11] http://www.symbiote-h2020.eu/ontology/core#
[12] https://www.w3.org
[13] https://www.onem2m.org
[14] https://www.etsi.org
[15] https://www.ogc.org
13 https://www.ogc.org/standards/sensorthings

γεωχωρικό και ενοποιημένο τρόπο διασύνδεσης IoT συσκευών, δεδομένων και εφαρμογών μέσω του Παγκόσμιου Ιστού. Ο Ευρωπαϊκός οργανισμός πιστοποιήσεων και προτύπων ETSI δραστηριοποιείται σε ένα πλήθος πεδίων εφαρμογών, όπως είναι τα Δίκτυο 5G, η Ηλεκτρονική Υγεία (eHealth) και το Διαδίκτυο των Πραγμάτων (IoT). Πιο συγκεκριμένα στον τομέα του IoT, η οντολογία SAREF (Smart Application Reference Ontology)[17] του ETSI διευκολύνει την αντιστοίχιση υπαρχόντων στοιχείων (όπως προτύπων, πρωτοκόλλων και μοντέλων δεδομένων) στον τομέα των έξυπνων συσκευών. Τέλος, ένα από τα ευρύτερα χρησιμοποιούμενα πρότυπα στον τομέα του IoT αποτελούν το NGSI (Next Generation Service Interface)[18] και NGSI-LD (NGSI Linked-Data)[19], τα οποία αφορούν στην επεξεργασία, σημασιολογική σύνταξη και διαμοιρασμό IoT πληροφοριών.

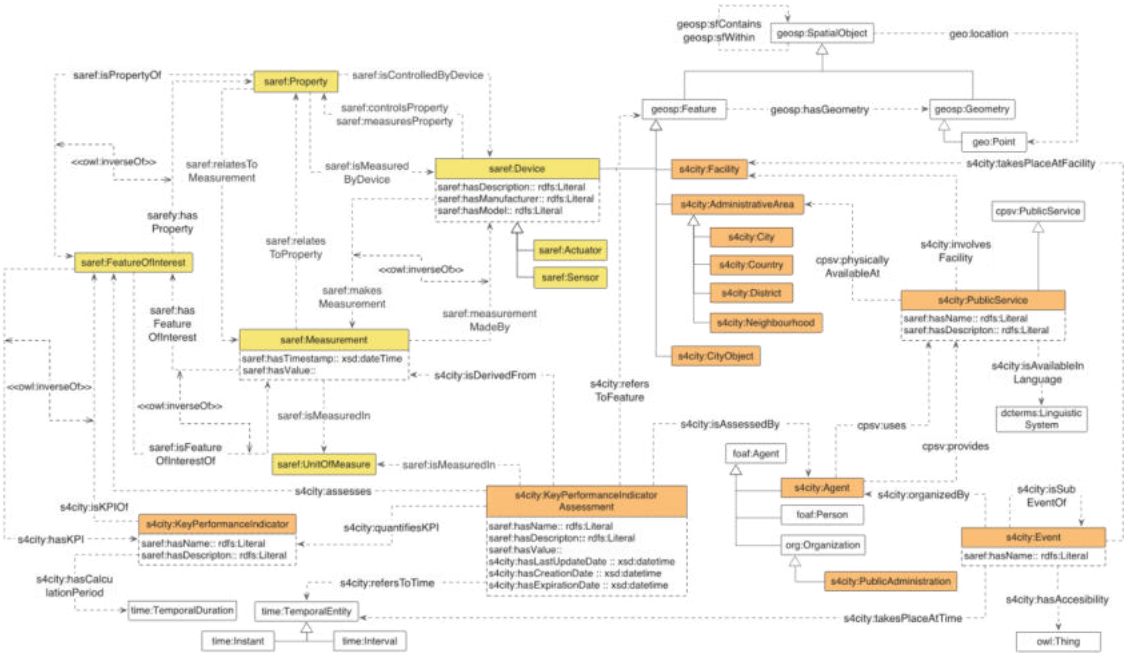
### 4.3.1 Σημασιολογική διαλειτουργικότητα στην Έξυπνη Πόλη

Στα πλαίσια ορισμού ενός μοντέλου πληροφορίας για την Έξυπνη Πόλη (Smart City), πραγματοποιήσαμε μια επισκόπηση των ήδη υπαρχουσών οντολογιών που έχουν εφαρμογή στο πεδίο ενδιαφέροντος. Μια αξιόλογη προσπάθεια για να συγκεντρωθεί το σύνολο των οντολογιών που βρίσκουν εφαρμογή στην Έξυπνη Πόλη έχει πραγματοποιηθεί από το έργο READY4SMARTCITIES FP7 CSA[20]. Από αυτές, αναλύσαμε και συγκεντρώσαμε τις βασικότερες και πιο σχετικές οντολογίες όπως παρουσιάζονται παρακάτω.

***SAREF4CITY:*** Η οντολογία SAREF4CITY[21] αποτελεί επέκταση της οντολογίας SAREF η οποία αποσκοπεί στην επίτευξη της διαλειτουργικότητας μεταξύ των διαφόρων συστημάτων και συσκευών (devices) που προέρχονται από ποικίλους παρόχους οι οποίοι δραστηριοποιούνται στον τομέα του IoT. Η SAREF και κατ' επέκταση η SAREF4CITY οντολογίες εκδίδονται από την τεχνική επιτροπή της ETSI SmartM2M[22].

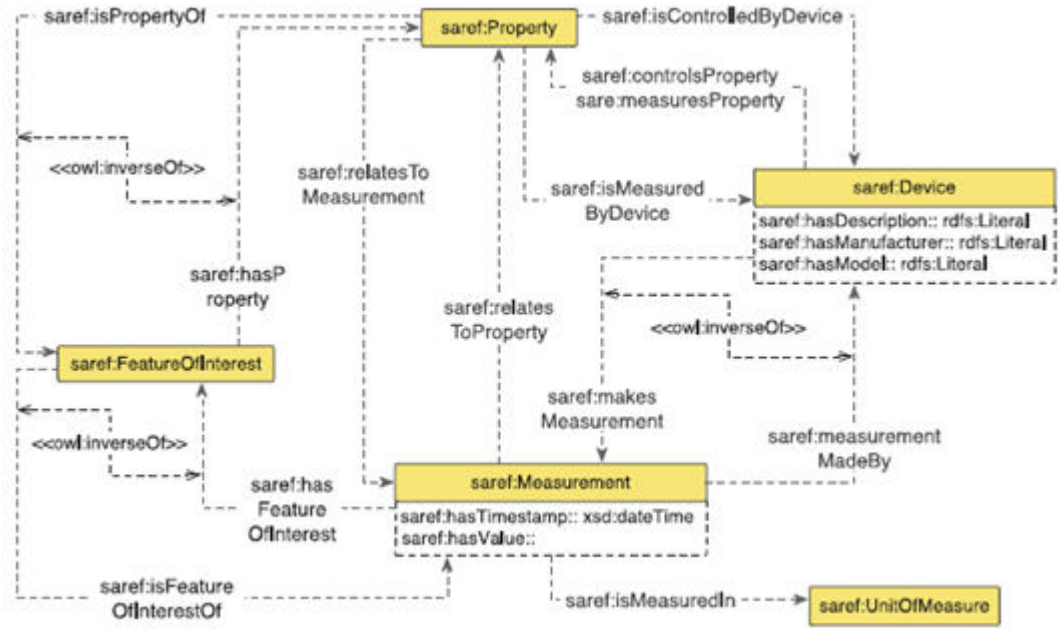
Το μοντέλο πληροφορίας του SAREF4CITY που φαίνεται στην παρακάτω εικόνα δίνει την δυνατότητα αντιστοίχισης μιας διοικητικής περιοχής (s4city:AdministrativeArea) σε περικλειόμενες υπο-περιοχές (όπως για παράδειγμα της γειτονιάς - s4city:Neighbourhood), μέσω «κληρονομικότητας» από την οντότητα geosp:SpatialObject και της geosp:Feature. Ομοίως, όλες οι υπο-οντότητες του s4city:AdministrativeArea, συγκεκριμένα οι: s4city:City, s4city:Country, s4city:District και s4city:Neighbourhood αποτελούν υπο-οντότητες του geosp:SpatialObject και «κληρονομούν» τις geosp:sfContains and geosp:sfWithin ιδιότητες.

17 https://saref.etsi.org
15 https://www.fiware.org/2016/06/08/fiware-ngsi-version-2-release-candidate/
16 https://fiware-datamodels.readthedocs.io/en/latest/ngsi-ld_howto/index.html
20 http://smartcity.linkeddata.es/
21 https://saref.etsi.org/saref4city/v1.1.2/
22 https://www.etsi.org/committee/smartm2m




*Εικόνα 11: Το μοντέλο πληροφορίας του SAREF4CITY, ως επέκταση του SAREF.*

Η μοντελοποίηση μετρήσεων στο SAREF4CITY βασίζεται στις οντότητες που ορίζονται στο SAREF (Εικόνα 12) και συγκεκριμένα από τις saref:Measurement και saref:UnitOfMeasure, οι οποίες μοντελοποιούν τις μετρήσιμες τιμές που αφορούν μια ιδιότητα (saref:Property) και την μονάδα μέτρησης αυτής της τιμής αντίστοιχα.
Παρόμοια με τους ορισμούς του symbIoTe CIM το οποίο περιγράφεται παρακάτω στην ενότητα 474.4.2, το SAREF ορίζει και τις οντότητες saref:Device (που διαχωρίζονται σε saref:Actuator και saref:Sensor), καθώς και την οντότητα saref:Property.




*Εικόνα 12: Το μοντέλο πληροφορίας SAREF.*

***cityGML Οντολογία:*** Η cityGML οντολογία[23] επεκτείνει το μοντέλο πληροφορίας/οντολογία του OGC με σκοπό την αναπαράσταση τρισδιάστατων μοντέλων πόλεων (3D city). Η

23 https://www.ogc.org/standards/citygml

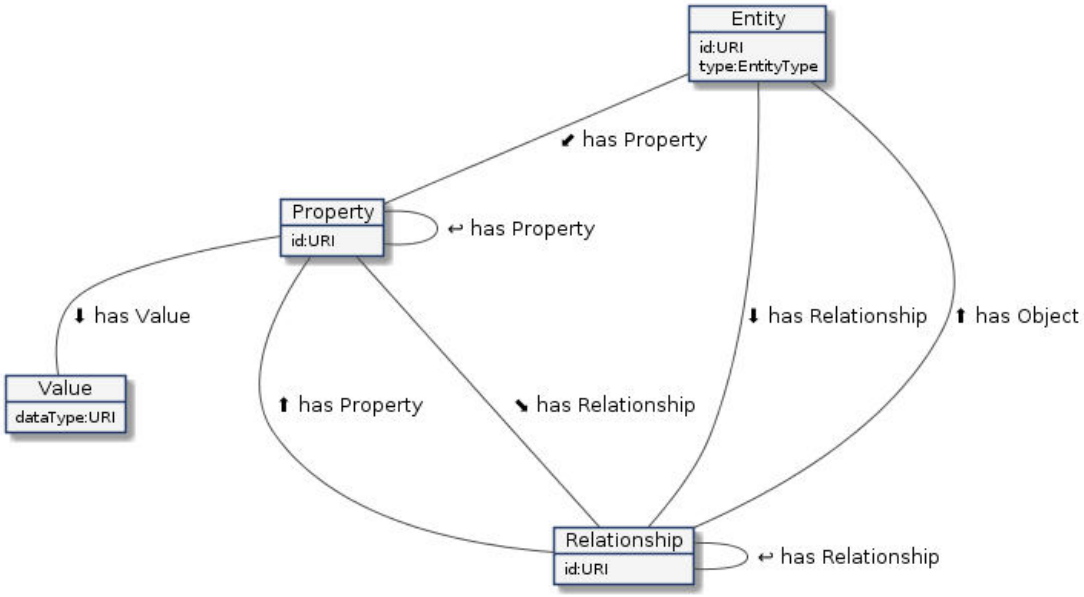
συγκεκριμένη οντολογία εξυπηρετεί την τρισδιάστατη αναπαράσταση αντικειμένων που αφορούν τον αστικό ιστό, όπως κτίρια, δρόμους, δέντρα κλπ.

***Km4city Οντολογία:*** Η οντολογία km4city[24] επιτρέπει την περιγραφή «Εξυπνων Πόλεων», αναπαριστώντας τις διάφορες οντότητες, συσχετίσεις και διασυνδέσεις που τις απαρτίζουν, συλλέγοντας δεδομένα από ποικίλες διαφορετικές πηγές, στις περιοχές της Τοσκάνης (MIIC, Muoversi Toscana, Osservatorio dei Trasporti), καθώς και από υπάρχουσες οντολογίες Open Data and Linked Data, οι οποίες διατίθενται από διάφορες δημοτικές κοινότητες (κυρίως της Φλωρεντίας).

***NSGI-LD:*** Το NGSI-LD αποτελείται από ένα μοντέλο πληροφορίας καθώς και ένα API για την δημοσίευση, την εγγραφή, τον ανοιχτό διαμοιρασμό και την υποβολή ερωτημάτων εύρεσης συναφούς, δομημένης πληροφορίας (context information). Οι βασικές δομές του μοντέλου NGSI-LD είναι:

- *Entity*: η οντότητα αυτή αποτελεί την σημασιολογική αναπαράσταση ενός αντικειμένου (*referent*), το οποίο υπάρχει στον φυσικό κόσμο. Το αντικείμενο αυτό δεν παρουσιάζει κατά ανάγκη μόνο φυσική υπόσταση (θα μπορούσε να είναι μια νομική ή μια διαχειριστική οντότητα), ή περιοριστική φύση (θα μπορούσε να είναι μια κατανεμημένη, συστημική δομή). Κάθε διακριτή περίπτωση της οντότητας Entity προσδιορίζεται μοναδικά από ένα URI (Uniform Resource Identifier), καθώς και από έναν τύπο (entity type).
- *Property*: η οντότητα αυτή εξυπηρετεί τον συσχετισμό ενός χαρακτηριστικού (NGSI-LD Value), προς ένα NGSI-LD Entity ή προς μια NGSI-LD Relationship ή προς ένα άλλο διαφορετικό NGSI-LD Property.
- *Relationship*: η οντότητα αυτή εξυπηρετεί την ευθεία σύνδεση μεταξύ ενός υποκειμένου (subject), το οποίο θα μπορούσε να είναι ένα NGSI-LD Entity, είτε ένα NGSI-LD Property, ή κάποιο άλλο NGSI-LD Relationship, και ενός αντικειμένου (object), το οποίο είναι ένα NGSI-LD Entity.
- Value: αποτελεί μια οντότητα που αναπαριστά μία JSON τιμή (π.χ. string, number, κλπ.), μιας JSON-LD τύπου τιμής ή μίας JSON-LD δομημένης τιμής (π.χ. set, list, κλπ.).

Στην εικόνα που ακολουθεί παρουσιάζεται το μοντέλο πληροφορίας του NGSI-LD, με όλες τις βασικές δομές που περιγράφηκαν παραπάνω.

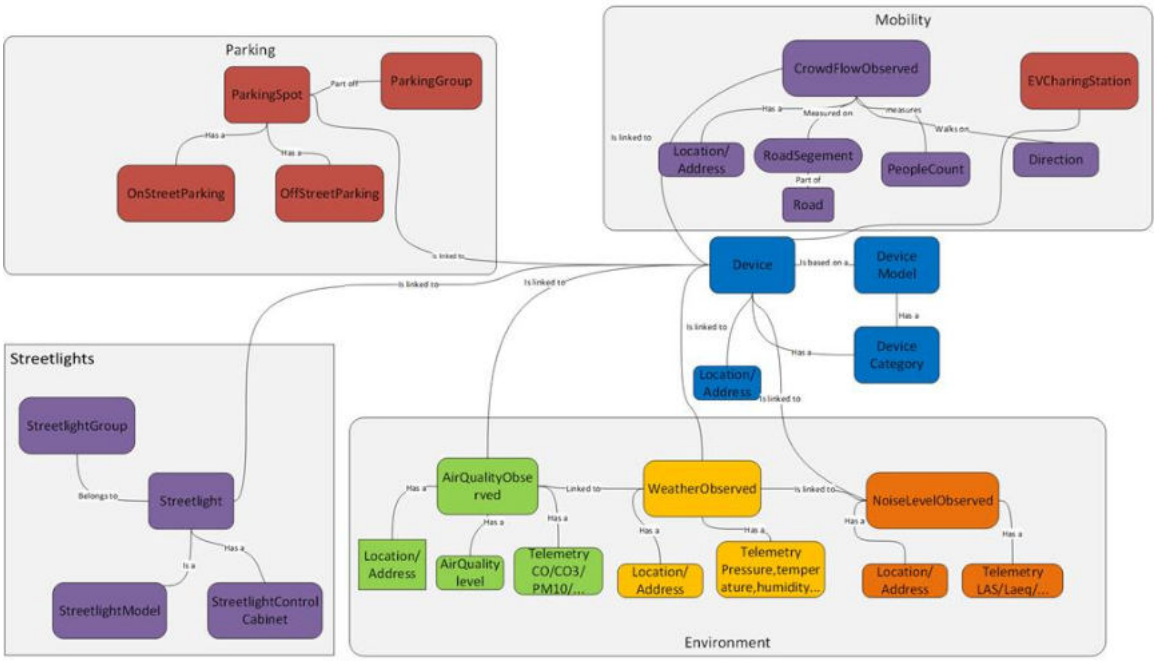



*Εικόνα 13: Μοντέλο πληροφορίας του NGSI-LD.*

24 https://www.km4city.org/

Το NGSI-LD περιέχει οντότητες για Έξυπνες Πόλεις και τις εφαρμογές τους όπως η Έξυπνη Στάθμευση.

***NGSI-LD for smart cities:*** Όπως φαίνεται στην εικόνα που ακολουθεί, το Smart Cities NGSI-LD[25] μοντέλο απαρτίζεται από επιμέρους οντότητες που αφορούν τα έξυπνα κτίρια (dataModel.Building), στάθμευση (dataModel.Parking), πάρκα (dataMdel.ParksAndGardens), φωτισμό (dataModel.Streetlighting), λιμάνια (dataModel.Ports), άλλα σημεία ενδιαφέροντος (dataModel.PointOfInterest), μεταφορά (dataModel.Transportation), απορρίμματα (dataModel.WasteManagement), αστική κινητικότητα (dataModel.UrbanMobility), μετεωρολογικά δεδομένα (dataModel.Weather) κλπ.




*Εικόνα 14:Παράδειγμα υποσυνόλου οντοτήτων του NGSI-LD μοντέλου.*

## 4.4 Μοντέλο πληροφορίας για το IoTFeds

### 4.4.1 Τεχνική προσέγγιση για την επίτευξη της σημασιολογικής διαλειτουργικότητας

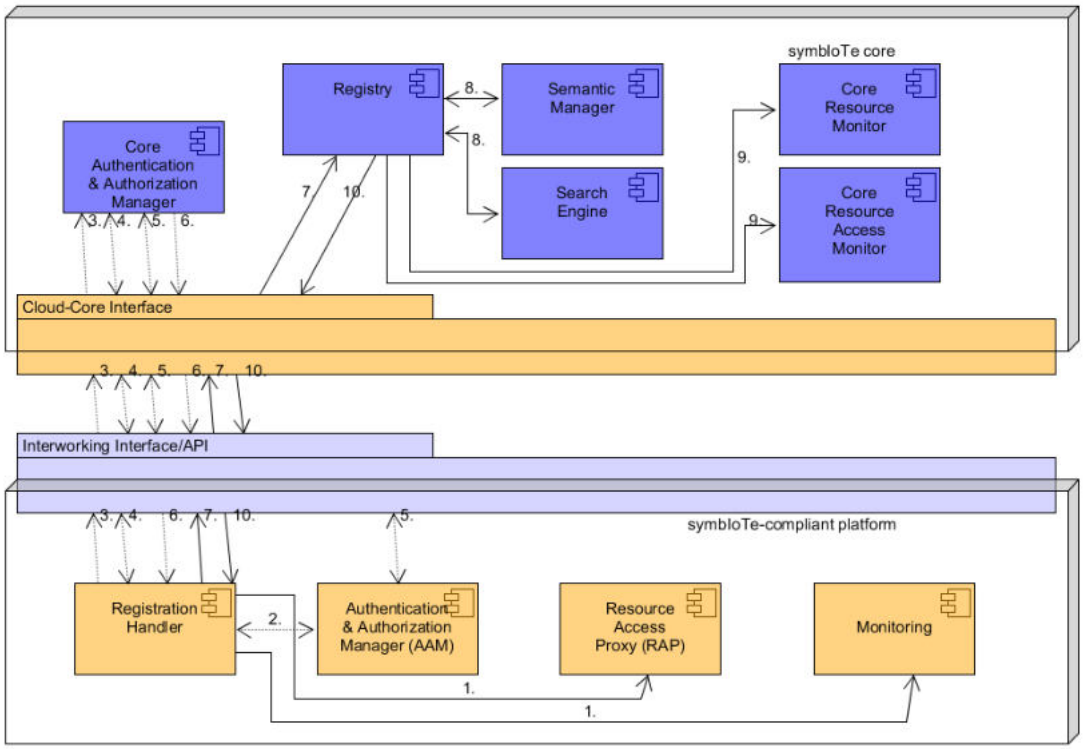
Στα πλαίσια του έργου IoTFeds η τεχνική προσέγγιση για την επίτευξη της σημασιολογικής διαλειτουργικότητας βασίστηκε στη υπάρχουσα λύση του symbIoTe που περιγράφεται παρακάτω, ενώ εμπλουτίστηκε το υπάρχον μοντέλο πληροφορίας ενσωματώνοντας οντολογίες για την κάλυψη αναγκών της Έξυπνης Πόλης. Για την επίτευξη της σημασιολογικής διαλειτουργικότητας, το symbIoTe βασίζεται στον ορισμό ενός βασικού μοντέλου πληροφορίας (Core Information Model - CIM), μίας οντολογίας όπου ορίζονται οι βασικές οντότητες και η βασική πληροφορία σχετικά με τους πόρους (resources), τους ενεργοποιητές (actuators) και τις υπηρεσίες (services) και στις οποίες θα πρέπει να βασίζεται η εγγραφή κάθε συσκευής στο σύστημα. Το μοντέλο πληροφορίας του symbIoTe έχει σχεδιαστεί με στόχο να είναι όσο πιο γενικό γίνεται αλλά και από την άλλη να είναι όσο λεπτομερές χρειάζεται. Επιπρόσθετες λεπτομέρειες για τις συσκευές μπορούν να περιγραφούν χρησιμοποιώντας μοντέλα πληροφορίας συγκεκριμένα για μία πλατφόρμα (platform specific information models - PIMs) για να παρέχει ευελιξία στη μεριά του παρόχου της πλατφόρμας. Ωστόσο, στην προσέγγιση αυτή, Core Information Model with Extensions (Michael Jacoby, 2017), όλα τα μοντέλα πληροφορίας πρέπει να συμμορφώνονται στο βασικό μοντέλο πληροφορίας όπου ορίζονται οι βασικές κλάσεις και οι σχέσεις μεταξύ τους και μπορούν να επεκτείνουν το μοντέλο αυτό για

25 https://github.com/smart-data-models/SmartCities

τις δικές τους ανάγκες χρησιμοποιώντας απευθείας τις κλάσεις αυτές ή ορίζοντας υποκλάσεις αυτών για επεκτάσεις του μοντέλου συγκεκριμένες για την περίπτωσή τους όπως επιπρόσθετα χαρακτηριστικά (properties). Με τον τρόπο αυτό το symbIoTe παρέχει στις πλατφόρμες δύο τύπους διαλειτουργικότητας: άμεση διαλειτουργικότητα (out-of-the-box interoperability ή αλλιώς interoperability by standardization), όπου κάθε πλατφόρμα χρησιμοποιεί (μερικώς) το ίδιο λεξιλόγιο για να περιγράψει τις συσκευές της μέσω της συμμόρφωσης και στοίχισης (alignment) με το CIM και διαλειτουργικότητα μέσω αντιστοίχισης (interoperability by mapping) όπου δύο πλατφόρμες χρησιμοποιούν δύο τελείως διαφορετικά λεξιλόγια (πέρα από το CIM που επεκτείνουν) αλλά συσχετίζονται. Στην περίπτωση αυτή μπορεί να οριστεί μία συσχέτιση μεταξύ των λεξιλογίων τους (mapping) για να μπορούν να κατανοούν επιπρόσθετα χαρακτηριστικά (properties) που ορίζουν με διαφορετικό τρόπο στα λεξιλόγιά τους. Η συσχέτιση αυτή μεταξύ των δύο λεξιλογίων (alignment) αποθηκεύεται και αξιοποιείται στο symbIoTe για την επίτευξη της διαλειτουργικότητας μεταξύ τους.
Διαφορές στις οντολογίες (ontologies mismatches) μπορεί να συμβούν πολύ συχνά ακόμα και αν οι οντολογίες που χρησιμοποιούνται από κάθε πλατφόρμα καλύπτουν την ίδια περιοχή ενδιαφέροντος, καθώς μπορεί η ταξινομία (taxonomy) να αφορά άλλο εύρος (scope) ή βαθμό λεπτομέρειας (granularity) ή μπορεί να χρησιμοποιείται η ίδια ταξινόμηση αλλά σε διαφορετική ορολογία ή γλώσσα. Η σημασιολογική αντιστοίχιση (semantic mapping) προσπαθεί να λύσει τις αναντιστοιχίες ορίζοντας δηλώσεις και κανόνες για το πώς τα δεδομένα που εκφράζονται χρησιμοποιώντας μια οντολογία μπορούν να μεταφραστούν στους όρους μιας άλλης οντολογίας. Μια τέτοια δήλωση ή κανόνας ονομάζεται μοτίβο αντιστοιχίας (correspondence pattern) και αποτελείται από μια οντολογία πηγής και μια οντολογία στόχου και κάποιες πληροφορίες αντιστοιχίας/μετασχηματισμού. Όλα τα μοτίβα αντιστοιχίας που έχουν την ίδια οντολογία πηγής και προορισμού μαζί σχηματίζουν μια ευθυγράμμιση μεταξύ των δύο οντολογιών (alignment). Για αναντιστοιχίες όπου η μετάφραση δεν είναι δυνατή δεν μπορεί να επιτευχθεί διαλειτουργικότητα. Συνεπώς, μόνο τα δεδομένα που μοντελοποιούνται και στις δύο οντολογίες, δηλαδή ένα υποσύνολο των δύο οντολογιών, μπορούν να μεταφραστούν με ασφάλεια μεταξύ τους σε πολλές περιπτώσεις.
Τέλος, στο symbIoTe παρέχεται το Best-Practice Information Model (BIM) που είναι μία ειδική κατηγορία PIM το οποίο καλύπτει κάποιες περιοχές ενδιαφέροντος (use case domains).
Οι βασικές υπηρεσίες και τα επιμέρους στοιχεία λογισμικού του symbIoTe που εμπλέκονται στην επίτευξη της σημασιολογικής διαλειτουργικότητας περιγράφονται παρακάτω:

- Εγγραφή πόρου (resource registration): Ο χρήστης εγγράφει μία νέα συσκευή στέλνοντας αίτημα στον Registration Handler (RH), στοιχείο λογισμικού στην μεριά της πλατφόρμας (symbIoTe cloud). Στα πλαίσια του IoTFeds η εγγραφή της συσκευής από τον χρήστη γίνεται μέσα από το αντίστοιχο endpoint του symbIoTeAPI το οποίο παρέχει το σύνολο των υπηρεσιών για έναν client, ενώ η συσκευή εγγράφεται και στη βάση του BaaS συστήματος. Εσωτερικά στο symbIoTe η αλληλουχία βημάτων που ακολουθείται για την ενημέρωση των εμπλεκόμενων στοιχείων λογισμικού περιγράφεται στην Εικόνα 15Εικόνα 6 (Jacoby, και συν., 2017). Ο RH ειδοποιεί το Resource Access Proxy (RAP) που είναι υπεύθυνο για την πρόσβαση στους πόρους στη μεριά της πλατφόρμας καθώς και το Monitoring στοιχείο λογισμικού που παρακολουθεί την κατάσταση των συσκευών της πλατφόρμας για τις αλλαγές και στέλνει αίτημα στο Registry για την εγγραφή, διαγραφή ή τροποποίηση της συσκευής στο κεντρικοποιημένο σύστημα (symbIoTe Core) με την συσκευή να αντιστοιχίζεται σε κάποιο από τους διαθέσιμους τύπους πόρου που ορίζονται στο CIM. Στην περίπτωση που χρησιμοποιείται κάποιο PIM (το οποίο έχει ήδη καταχωρηθεί στο symbIoTe μέσω του Administration GUI), το JSON αντικείμενο που στέλνεται περιέχει επιπλέον κάποια RDF περιγραφή του πόρου. Το μήνυμα αυτό στη συνέχεια προωθείται στον Semantic Manager (SM) το οποίο ελέγχει την εγκυρότητα της περιγραφής με βάση το μοντέλο που χρησιμοποιείται (PIM/CIM/BIM). Στην περίπτωση που δεν ορίζεται κάποιο PIM επιπρόσθετα μετατρέπεται το JSON αντικείμενο και στην αντίστοιχη RDF περιγραφή.

Η RDF περιγραφή στη συνέχεια καταλήγει και αποθηκεύεται στο Search στοιχείο λογισμικού. Τέλος, ενημερώνονται τα στοιχεία λογισμικού Core Resource Monitor (CRM) and Core Resource Access Monitor (CRAM) που παρακολουθούν και καταγράφουν την λειτουργία του πόρου και την πρόσβαση σε αυτόν. Η γενική αρχιτεκτονική του symbIoTe και τα επιμέρους στοιχεία λογισμικού έχουν περιγραφεί στο παραδοτέο Π1.3.



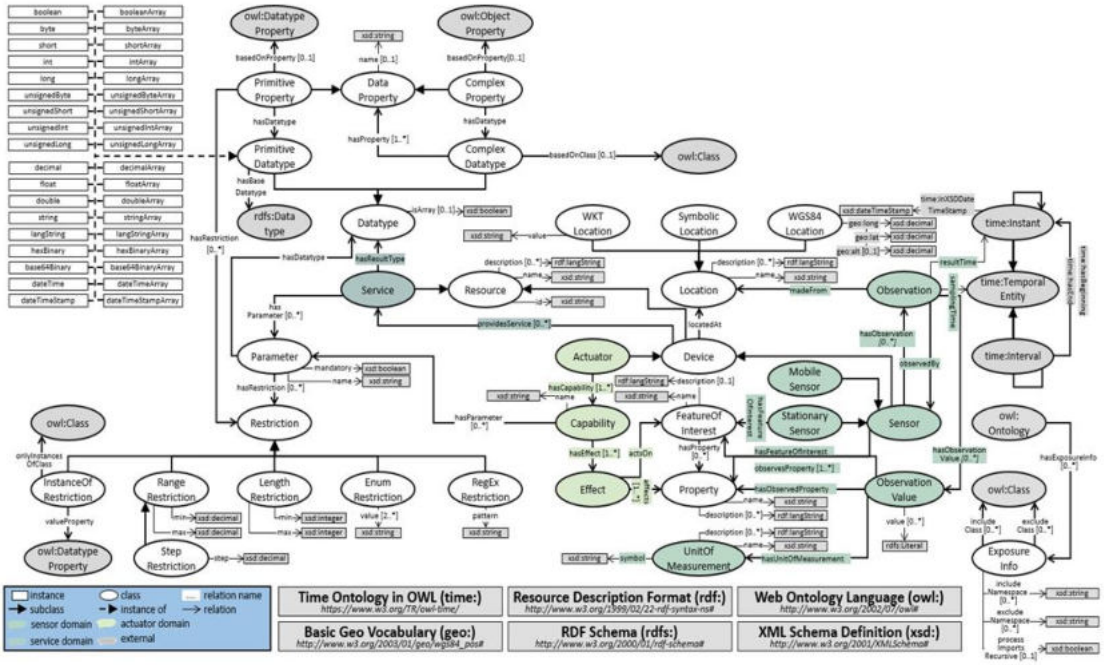

*Εικόνα 15: Εσωτερική επικοινωνία μεταξύ στοιχείων του symbIoTe υποσυστήματος για την εγγραφή IoT πόρου.*

- Ανακάλυψη πόρων και χρήση SPARQL Query Rewriting: Η ανακάλυψη πόρων παρέχεται από τις κεντρικοποιημένες υπηρεσίες του Search στοιχείου λογισμικού. Στα πλαίσια του IoTFeds η υπηρεσία καλείται από το αντίστοιχο symbIoTeAPI endpoint. Όταν ένα αίτημα απαιτεί χρήση query rewriting, το στοιχείο λογισμικού Search κοιτάζει τις υπάρχουσες (αποθηκευμένες) συσχετίσεις (mappings) που σχετίζονται με το αίτημα του χρήστη. Στη συνέχεια, η μηχανή αναζήτησης (Search) προωθεί το ερώτημα (query) και τις αντιστοιχίσεις (mappings) στον Semantic Manager (SM) ζητώντας να ξαναγράψει το ερώτημα με βάση τις αντιστοιχίσεις. Η μηχανή αναζήτησης εκτελεί στη συνέχεια τα εκ νέου ερωτήματα που της επιστράφηκαν από τον SM και παίρνει τη λίστα με τα αποτελέσματα τα οποία ξαναπερνάει στον SM για να τα μεταφράσει και να λάβει τα επανεγγραφέντα αποτελέσματα ώστε να απαντήσει το ερώτημα του χρήστη.
- Πρόσβαση στα δεδομένα (Resource Access): Η διεπαφή για την πρόσβαση στα δεδομένα βασίζεται στις οντότητες που ορίζονται στο CIM για τον καθορισμό των URLs μοτίβων που παρέχει το RAP. Επίσης, το μοντέλο πληροφορίας που χρησιμοποιείται (BIM ή PIM) καθορίζει τη δομή του αντικειμένου JSON (το observation) που επιστρέφεται στο URL. Καθώς οποιοδήποτε PIM είναι επέκταση του CIM, τα χαρακτηριστικά (properties) του αντικειμένου JSON που ορίζονται στο CIM είναι κατανοητά από τον οποιονδήποτε. Οποιεσδήποτε άλλες επιπρόσθετες πληροφορίες (properties) στο PIM μπορούν να αγνοηθούν σε περίπτωση που δεν είναι εφικτή η κατανόησή τους ή να μεταφραστούν ώστε να ταιριάζουν στο PIM που χρησιμοποιείται μέσω των βιβλιοθηκών του symbIoTe (symbIoTe Libraries).

Στην περίπτωση των ομοσπονδιών, κατά τη δημιουργία μίας ομοσπονδίας μπορεί να επιλεγεί το μοντέλο BIM που παρέχει το symbIoTe ως το μοντέλο που πρέπει να χρησιμοποιηθεί σε αυτήν την ομοσπονδία ή να συμφωνηθεί και να εγγραφεί στο symbIoTe οποιοδήποτε άλλο κοινό μοντέλο πληροφοριών PIM που θα διευκολύνει τις συναλλαγές δεδομένων και μεταδεδομένων εντός της ομοσπονδίας και είναι επέκταση του CIM. Η προσέγγιση της διαλειτουργικότητας μέσω αντιστοίχισης δεν υποστηρίζεται στην περίπτωση των ομοσπονδιών από τα στοιχεία λογισμικού διαχείρισης ομοσπονδιών (compliance level L2) παρά μόνο από τις κεντρικοποιημένες υπηρεσίες του symbIoTe (L1 core services). Εάν οι λειτουργικότητες της επικύρωσης του μοντέλου ομοσπονδίας και του semantic mapping είναι επιθυμητές, οι πλατφόρμες της ομοσπονδίας θα πρέπει να χρησιμοποιούν τις κεντρικοποιημένες υπηρεσίες και να αναπτύξουν τις δικές τους προσθήκες (plugins) για την υποστήριξη του semantic mapping. Στα πλαίσια του IoTFeds κάθε ομοσπονδία έχει τη δυνατότητα εγγραφής του δικού της PIM (Federation PIM) που θα πρέπει τα ομόσπονδα μέλη να ακολουθούν. Το μοντέλο αυτό καταχωρείται στο σύστημα και επιλέγεται κατά την δημιουργία της ομοσπονδίας. Πλατφόρμες που θέλουν να χρησιμοποιήσουν διαφορετικό PIM θα πρέπει να ορίσουν τις αντιστοιχίες μεταξύ των επιθυμητών μοντέλων (mappings).

### 4.4.2 Βασικό μοντέλο πληροφορίας του symbIoTe (CIM)

Το βασικό μοντέλο πληροφορίας του symbIoTe, Core Information Model (CIM), φαίνεται στην παρακάτω εικόνα:

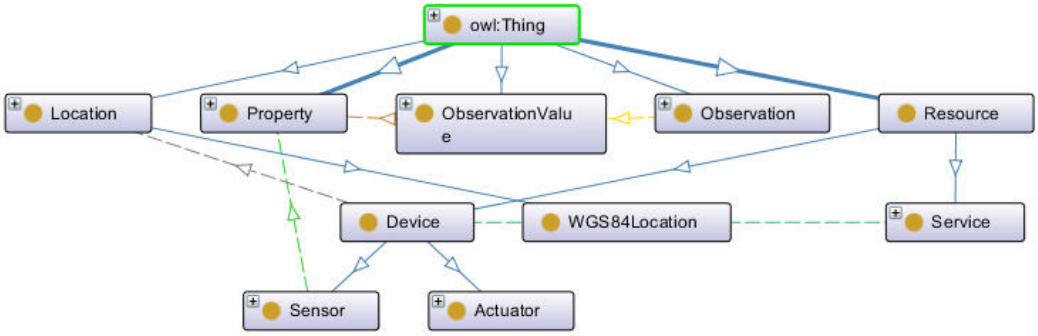



*Εικόνα 16: Το βασικό μοντέλο πληροφορίας του symbIoTe – CIM.*

Το CIM βασίζεται στις οντολογίες SSN και SOSA. Η βασική οντότητα του CIM είναι ο Πόρος (*Resource)*, το οποίο ανήκει σε μία πλατφόρμα (*Platform*). Έπειτα το *Resource* αποτελείται από τις υπο-οντότητες Υπηρεσία (Service) και Συσκευή (*Device)*, η οποία διακλαδώνεται στις υπο-οντότητες Αισθητήρας (*Sensor)* και Ενεργοποιητής (*Actuator) της οντολογίας SOSA*.
Από την SSN/SOSA οντολογία, το CIM δανείζεται επίσης τις οντότητες:

- *Observation:* οντότητα που περιγράφει μια μετρήσιμη ποιότητα ενός device, sensor, actuator κλπ.
- *ObservationValue:* οντότητα που περιγράφει τις τιμές (values) που αφορούν ένα Observation από ένα device, sensor, actuator κλπ.
- *Property*: οντότητα που περιγράφει ένα ποιοτικό χαρακτηριστικό μιας άλλης οντότητας, π.χ. ενός Sensor, ή Device.

Σημαντική επίσης για το CIM είναι η οντότητα *Location,* η οποία βασίζεται στην *WGS84Location*[26] οντολογία. Οι οντότητες αυτές και οι σχέσεις μεταξύ τους συνοψίζονται στην παρακάτω εικόνα.




*Εικόνα 17: Βασικές οντότητες του symbIoTe CIM.*

Οι παραπάνω οντότητες του symbIoTe CIM θα αποτελέσουν την βάση για την δημιουργία ενός εκτεταμένου μοντέλου πληροφορίας στο πεδίο εφαρμογής της Έξυπνης Πόλης εντάσσοντας οντότητες και από άλλες ευρέως χρησιμοποιούμενες οντολογίες για Smart Cities, όπως η SAREF4CITY.

### 4.4.3 Μοντέλα πληροφορίας πιλοτικών πλατφορμών

Στην ενότητα αυτή παρουσιάζονται τα μοντέλα πληροφορίας PIM που χρησιμοποιούνται στις υποστηρικτικές πλατφόρμες Smart Environment, Spotypal/Zastel και UiTOP οι οποίες θα αξιοποιηθούν στα πλαίσια του πιλοτικού του έργου IoTFeds με στόχο την επαλήθευση της κάλυψης των αναγκών των πιλοτικών πλατφορμών από το επεκταμένο μοντέλο πληροφορίας του IoTFeds.

#### *4.4.3.1 Μοντέλο Πληροφορίας Smart Environment*

Το μοντέλο πληροφορίας του συστήματος Smart Environment αποτελείται από 10 βασικές οντότητες, 13 object properties, καθώς και 2 data properties, τα οποία αναλύονται ακολούθως. Από τις 10 οντότητες που έχουν οριστεί για την οντολογία του συστήματος, οι 5 (Device, Function, Measurement, Property και Unit of Measure) χρησιμοποιήθηκαν από την οντολογία SAREF (Smart Applications REFerence), με κάποιες προσθήκες σε επιπλέον αισθητήρες που χρησιμοποιούνται από το προτεινόμενο σύστημα. Οι υπόλοιπες 5 οντότητες (Wireless Sensor Network, OpenWeatherMap API, Cloud, Event και Smart Environment UI) δημιουργήθηκαν βάσει των σχεδιαστικών αναγκών του συστήματος.

Πιο συγκεκριμένα, οι κύριες οντότητες που ορίστηκαν για το σύστημα Smart Environment, καθώς και οι μεταξύ τους συσχετίσεις, παρουσιάζονται και περιγράφονται ως εξής:

1. SAREF:Device: οντότητα που περιγράφει οποιαδήποτε συσκευή του συστήματος. Πιο συγκεκριμένα, το σύστημα Smart Environment περιλαμβάνει μόνο ένα σύνολο αισθητήρων, οπότε χρησιμοποιήθηκε μόνο η υποκλάση «Sensor», ορίζοντας 4 διαφορετικούς τύπους αισθητήρων (temperature sensor, humidity sensor, air quality sensor και light sensor). Επιπλέον, κάθε αισθητήρας του συστήματος έχει ως βασική λειτουργία (*has function*) τη λήψη μετρήσεων από το περιβάλλον του, ενώ αντίστοιχα κάθε συγκεκριμένος τύπος αισθητήρα λαμβάνει ένα συγκεκριμένο τύπο μέτρησης (*measures property*).
2. SAREF:Function: οντότητα που περιγράφει τη λειτουργία που επιτελεί κάθε συσκευή του συστήματος. Πιο συγκεκριμένα, επειδή το σύστημα Smart Environment χρησιμοποιεί συσκευές για τη λήψη μετρήσεων από το περιβάλλον, χρησιμοποιήθηκε μόνο η υποκλάση «Sensing Function». Κάθε λειτουργία λαμβάνει μια συγκεκριμένη περιβαλλοντική τιμή (*has sensing range*) και σχετίζεται με ένα συγκεκριμένο τύπο αισθητήρα (*has sensor type*).

26 http://www.w3.org/2003/01/geo/wgs84_pos

3. SAREF:Measurement: οντότητα που σχετίζεται με την τιμή της μέτρησης που λαμβάνεται από κάθε αισθητήρα του συστήματος Smart Environment. Πιο συγκεκριμένα, κάθε τιμή έχει μια συγκεκριμένη μονάδα μέτρησης (*is measured in*) και σχετίζεται με μια συγκεκριμένη ιδιότητα (*relates to property*).
4. SAREF:Property: ιδιότητα που σχετίζεται με τον τύπο της λαμβανόμενης μέτρησης. Κάθε μέτρηση που λαμβάνεται από τους αισθητήρες του συστήματος συλλέγεται από ένα συγκεκριμένο αισθητήρα (*is collected by device*) και σχετίζεται με μία συγκεκριμένη μέτρηση, inferred από την κλάση «Measurement» (inferred object property *relates to measurement*).
5. SAREF:Unit of Measure: οντότητα που σχετίζεται με τη μονάδα μέτρησης της λαμβανόμενης τιμής.
6. Wireless Sensor Network: οντότητα που χρησιμοποιείται για την περιγραφή της λειτουργίας του Smart Environment. Πιο συγκεκριμένα, είναι η οντότητα που επικοινωνεί τόσο με το «OpenWeatherMap API» για τη συλλογή επιπλέον περιβαλλοντικών τιμών (*requests data from*) όσο και με το «Cloud» για την αποθήκευση των δεδομένων (*stores data to*).
7. OpenWeatherMap API: οντότητα που σχετίζεται με το OpenWeatherMap. Η συγκεκριμένη κλάση περιλαμβάνει μόνο μία ιδιότητα, η οποία σχετίζεται με την επικοινωνία με το «Wireless Sensor Network» (*established communication*) για την παροχή επιπρόσθετων περιβαλλοντικών τιμών (wind speed, CO, NO2, SO2).
8. Cloud: οντότητα που αναπαριστά το αποθετήριο του συστήματος Smart Environment. Πιο συγκεκριμένα, περιλαμβάνει ιδιότητες που σχετίζονται με την αποθήκευση των δεδομένων που παρέχονται από το «Wireless Sensor Network» (inferred object property *stores data from*), καθώς και με την επεξεργασία των δεδομένων αυτών για την εξαγωγή μοντέλων πρόγνωσης (*offers prediction for*). Τέλος, αποστέλλει τα αποτελέσματα των προγνώσεων στο «Smart Environment UI» για απεικόνιση (*sends prediction result to*).
9. Event: οντότητα που περιλαμβάνει τα μοντέλα πρόβλεψης κατηγοριοποιημένα ανά γεγονός (πρόγνωση καιρού, πρόγνωση ποιότητας αέρα και πρόβλεψη ρίσκου εκδήλωσης πυρκαγιάς).
10. Smart Environment UI: οντότητα που σχετίζεται με τη διεπαφή που χρησιμοποιείται από το σύστημα του Smart Environment για την απεικόνιση των αποτελεσμάτων πρόβλεψης των μοντέλων. Διαθέτει μόνο μία ιδιότητα η οποία είναι inferred από το «Cloud» (inferred object property *receives prediction result from*).

Η γενική εικόνα της οντολογίας του συστήματος Smart Environment παρουσιάζεται στην Εικόνα 18 που ακολουθεί. Οι διακεκομμένες γραμμές αφορούν συσχετίσεις υποκλάσεων. Επιπλέον, οι βασικές οντότητες του συστήματος παρουσιάζονται με γαλάζιο χρώμα, οι υποκλάσεις με κίτρινο, ενώ οι υποκλάσεις της υποκλάσης «Sensor» με γκρι.


*Εικόνα 18: Οντολογία του συστήματος Smart Environment βάσει της οντολογίας SAREF.*

Τέλος, στους πίνακες που ακολουθούν παρουσιάζονται αναλυτικά όλες οι ιδιότητες του συστήματος Smart Environment. Πιο συγκεκριμένα, ο Πίνακας 3 περιλαμβάνει όλες τις συσχετίσεις των οντοτήτων του συστήματος που αναλύθηκαν παραπάνω.

*Πίνακας 3: Object properties για το σύστημα Smart Environment.*

| Object properties | | | |
|---|---|---|---|
| **Name** | **Domain** | **Range** | **Inverseof** |
| has function | Sensor | Sensing Function | - |
| measures property | Temperature sensor<br>Humidity sensor<br>Air Quality sensor<br>Light sensor | Temperature<br>Humidity<br>PM2.5<br>Light | - |
| sends data to | Sensor | Wireless Sensor Network | obtains data by |
| has sensing range | Function | Measurement | - |
| has sensor type | Function | Property | - |
| is measured in | Measurement | Unit of Measure | - |
| relates to property | Measurement | Property | relates to measurement |
| is measured by device | Property | Device | - |
| requests data from | Wireless Sensor Network | OpenWeatherMap API | - |
| established communication | OpenWeatherMap API | Wireless Sensor Network | - |

| | | | |
|---|---|---|---|
| stores data to | Wireless Sensor Network | Cloud | stores data from |
| offers prediction for | Cloud | Event | - |
| sends prediction result to | Cloud | Smart Environment UI | receives prediction result from |

Αντίστοιχα, ο Πίνακας 4 παρουσιάζει τα data properties του μοντέλου πληροφορίας του συστήματος Smart Environment. Οι συγκεκριμένες ιδιότητες είναι δύο και περιλαμβάνουν την «*has value*», που σχετίζεται τόσο με όλες τις μετρήσεις που λαμβάνονται από τους αισθητήρες του συστήματος, όσο και με τα προβλεπόμενα γεγονότα, και είναι τιμές τύπου *float*, και την «*has timestamp*», η οποία υποδεικνύει τη χρονική στιγμή λήψης κάθε μέτρησης και είναι τύπου *dateTime*.

*Πίνακας 4: Data properties για το σύστημα Smart Environment.*

| Data Properties | | |
|---|---|---|
| **Name** | **Domain** | **Range** |
| has value | - | float |
| has timestamp | Measurement | dateTime |

#### 4.4.3.2 SpotyPal/Zastel

Το μοντέλο πληροφορίας της πλατφόρμας SpotyPal/Zastel είναι βασισμένο στις οντολογίες Semantic Sensor Network (SSN) και Sensor Observation Sampling Actuator (SOSA). Οι εξωτερικές οντότητες προερχόμενες από τις οντολογίες αυτές είναι οι εξής:

- Sensor
- Observation
- Result
- Property
- Platform

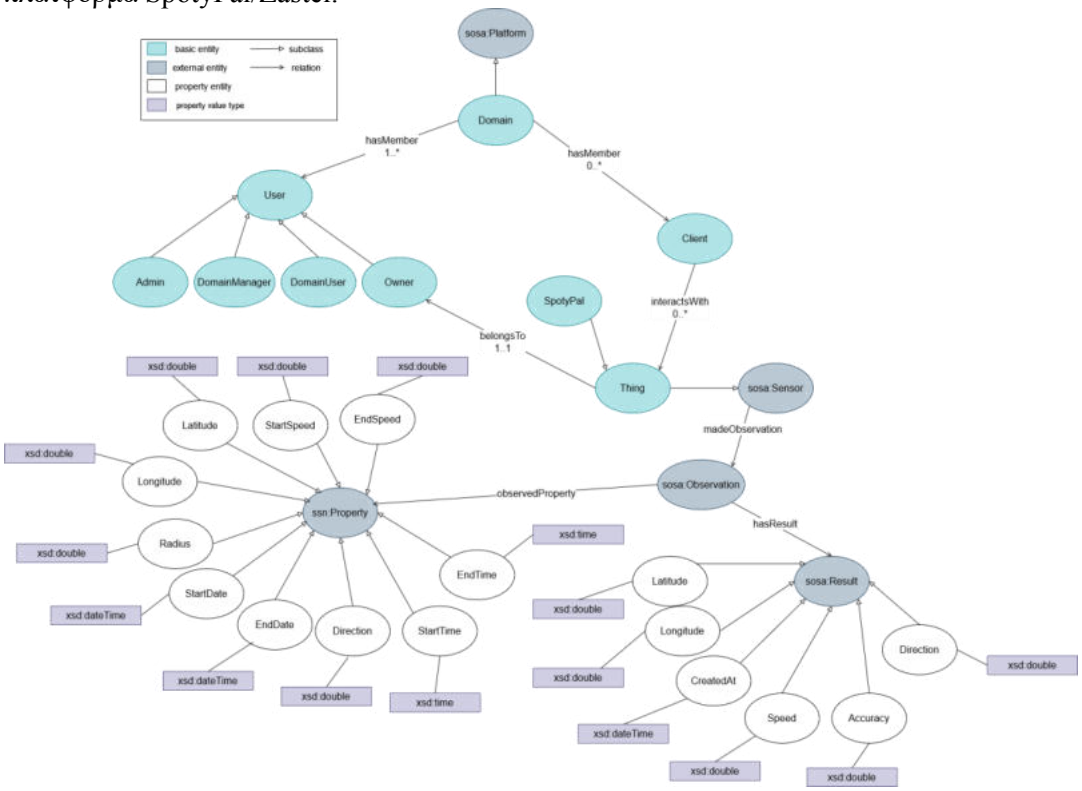
Οι βασικές εσωτερικές οντότητες οι οποίες σχετίζονται άμεσα με την πλατφόρμα SpotyPal/Zastel, είναι οι ακόλουθες:

- Domain: Με τον όρο αυτό αναπαρίσταται η οντότητα η οποία αξιοποιεί την πλατφόρμα SpotyPal/Zastel, προκειμένου να έχει στη διάθεσή της τις διαχειριστικές λειτουργίες όπως επίσης και τις λειτουργίες που αφορούν την παραγωγή δεδομένων από IoT συσκευές. Πρόκειται στην ουσία για το πεδίο χρήσης του SpotyPal/Zastel, ως ένα επιπρόσθετο, αφηρημένο επίπεδο IoT λειτουργιών.
- User: Η οντότητα αυτή αναπαριστά γενικά τους χρήστες ενός Domain οι οποίοι συνδέονται με τις διαχειριστικές λειτουργίες ή λειτουργίες υποδομής για το υποκείμενο IoT δίκτυο. Οι χρήστες αυτοί διακρίνονται με βάση τον ρόλο τους, με τον κάθε ρόλο να είναι επιφορτισμένος με συγκεκριμένα καθήκοντα μέσα στο πλαίσιο του Domain, τα οποία αφορούν λειτουργίες όπως η διαχείριση συσκευών και προσθήκη νέων χρηστών.
- Owner: Η οντότητα αυτή αποτυπώνει τον χρήστη εκείνον ο οποίος είναι υπεύθυνος για τη διαχείριση των συσκευών (Things) οι οποίες συμμετέχουν στο υποκείμενο δίκτυο IoT ενός Domain. Πιο συγκεκριμένα, ένας Owner έχει τη δυνατότητα να εγγράψει ένα νέο Thing, να το ενημερώσει και να το διαγράψει. Σε κάθε Owner ανήκει ένα υποσύνολο από τα συνολικά Things που περιλαμβάνονται στο Domain, και συγκεκριμένα αυτά τα οποία έχει εγγράψει. Επιπρόσθετα, ο Owner μπορεί να οργανώσει τα Things σε λογικές ομάδες, ταξινομημένες ιεραρχικά.
- DomainUser: Η οντότητα αυτή αναπαριστά έναν χρήστη ο οποίος επιτελεί διαχειριστικές λειτουργίες πάνω στους υπόλοιπους χρήστες ενός Domain. Είναι υπεύθυνος για την εγγραφή των Owners του Domain, όπως επίσης και των Clients. Επιπλέον, έχει δυνατότητες όπως η διαγραφή ενός Client.
- DomainManager: Με την οντότητα αυτή αποτυπώνεται ο διαχειριστής ενός Domain. Πρόκειται για τον υψηλότερο ιεραρχικά ρόλο εντός του Domain. Ο DomainManager μπορεί να εκτελέσει, πέραν των λειτουργιών τις οποίες εκτελούν και οι DomainUsers, επιπρόσθετες λειτουργίες που αφορούν στο Domain συνολικά. Τέτοιες λειτουργίες

είναι, για παράδειγμα, η διαγραφή ενός Owner ή ενός DomainUser, ή η αλλαγή παραμέτρων λειτουργίας του Domain.

- Admin: Η οντότητα αυτή περιγράφει τον διαχειριστή της πλατφόρμας SpotyPal/Zastel. Αυτός είναι και ο ύψιστος ρόλος για την πλατφόρμα, και έχει δικαίωμα σε λειτουργίες οι οποίες υπερβαίνουν τα όρια ενός μόνο Domain, σε αντίθεση με όλους τους υπόλοιπους χρήστες.
- Thing: Με την οντότητα αυτή αναπαρίσταται οποιαδήποτε συσκευή λαμβάνει μετρήσεις στο πλαίσιο λειτουργίας του υποκείμενου δικτύου IoT. Ένα Thing μπορεί να είναι ένα SpotyPal, ένα beacon, ένα smartphone, κάποια φορέσιμη συσκευή ή άλλου είδους συσκευή.
- Client: Αυτή η οντότητα αποτυπώνει τους χρήστες ενός Domain οι οποίοι συμμετέχουν σε διαδράσεις με τα εγγεγραμμένα Things του Domain. Για παράδειγμα, είναι οδηγοί οι οποίοι κατέχουν συσκευές SpotyPal και συλλέγουν κυκλοφοριακά δεδομένα, ή κάποιος ασθενής ο οποίος φοράει συσκευή για μέτρηση καρδιακών παλμών. Σημειώνεται ότι οι Clients είναι διαφορετικοί από τους Users, υπό την έννοια ότι δεν εκτελούν λειτουργίες στο επίπεδο του Domain, παρά μόνο τη διάδραση με τα Things. Τους Clients μπορούν να εγγράψουν και να διαγράψουν οι DomainUsers όπως επίσης και ο DomainManager.
- SpotyPal: Η οντότητα αυτή αποτυπώνει τις συσκευές SpotyPal οι οποίες μπορούν να χρησιμοποιηθούν και ως ένα είδος αισθητήρων, συλλέγοντας δεδομένα όπως για παράδειγμα η τοποθεσία ή η ταχύτητα του οχήματος στο οποίο έχουν τοποθετηθεί.

Στην παρακάτω εικόνα απεικονίζεται συνολικά το σχήμα του μοντέλου πληροφορίας για την πλατφόρμα SpotyPal/Zastel.




*Εικόνα 19: Συνολικό σχήμα του μοντέλου πληροφορίας Spotypal/Zastel.*

Στον παρακάτω πίνακα αναγράφονται οι συσχετίσεις μεταξύ των οντοτήτων του μοντέλου πληροφορίας που παρουσιάστηκε παραπάνω.

*Πίνακας 5:Object properties για το σύστημα Spotypal/Zastel.*

| Object properties | | | |
|---|---|---|---|
| **Name** | **Domain** | **Range** | **Inverseof** |
| hasMember | Domain | Admin<br>DomainManager | - |

| | | | |
|---|---|---|---|
| | | DomainUser<br>Owner<br>Client | |
| interactsWith | Client | Thing | - |
| belongsTo | Thing | Owner | - |
| hasObservation | Sensor | Observation | - |
| hasResult | Observation | Result | - |
| hasObservedProperty | Observation | StartTime<br>EndTime<br>Direction<br>StartDate<br>EndDate<br>Radius<br>Latitude<br>Longitude<br>StartSpeed<br>EndSpeed | - |

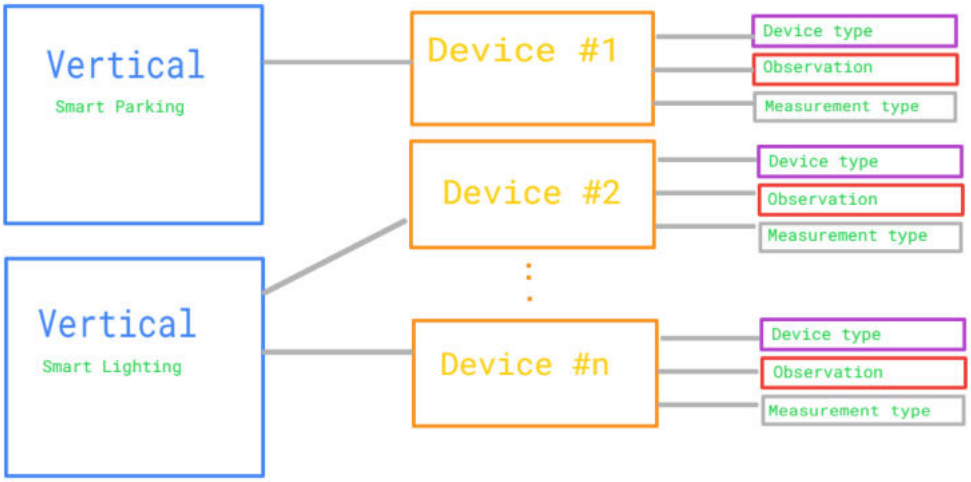
Στον επόμενο πίνακα παρουσιάζονται τα data properties του μοντέλου πληροφορίας

*Πίνακας 6: Data properties για το σύστημα Spotypal/Zastel.*

| Data Properties | | |
|---|---|---|
| **Name** | **Domain** | **Range** |
| has value | - | Double<br>DateTime<br>Time |

### *4.4.3.3 Μοντέλο Πληροφορίας UiTOP*

Στα πλαίσια του έργου IoTFeds η πλατφόρμα UiTOP συμμετέχει με δύο verticals της έξυπνης πόλης (Smart City), τον έξυπνο φωτισμό (Smart Lighting) και την έξυπνη στάθμευση (Smart Parking). Και τα δύο verticals βασίζονται σε ένα κοινό μοντέλο πληροφορίας (ontology), το UiTOP μοντέλο, που καλύπτει την έξυπνη πόλη (Smart City). Στο μοντέλο που χρησιμοποείται όλες οι συσκευές μοντελοποιούνται με την οντότητα devices που ανήκουν σε κάποιο τύπο συσκευής, device type, και ενημερώνουν κάποιους τύπους μετρήσεων, measurement types όπως φαίνεται στο παρακάτω σχήμα.




*Εικόνα 20: Βασικές οντότητες στο μοντέλο πληροφορίας του UiTOP.*

Σε κάθε vertical αντιστοιχεί ένας αριθμός από Devices**.** Η πληροφορία σχετικά με τον κάτοχο της συσκευής, tenant, καθορίζεται κατά την καταχώρηση του device στην πλατφόρμα.

Σε κάθε vertical υπάρχουν και εξειδικευμένες οντότητες. Στην περίπτωση του έξυπνου φωτισμού, οι εξειδικευμένες οντότητες για τα devices είναι ενδεικτικά: το pillar και ο controller. Στην περίπτωση της έξυπνης στάθμευσης, οι εξειδικευμένες device οντότητες είναι ενδεικτικά η συσκευή αισθητήρας με δυνατότητα αναγνώρισης του σταθμευμένου οχήματος.

### 4.4.4 Μοντέλο πληροφορίας IoTFeds

Σύμφωνα με τα παραπάνω, το μοντέλο πληροφορίας του IoTFeds αποτελείται από όλες τις οντότητες που απαρτίζουν το symbIoTe CIM μοντέλο (με τις βασικότερες να είναι αυτές που περιεγράφηκαν προηγουμένως), με προσθήκες για την Έξυπνη Πόλη και διαλειτουργικότητα με άλλα πρότυπα. Το μοντέλο πληροφορίας IoTFeds έχει καταχωρηθεί στο symbIoTe υποσύστημα του IoTFeds και αποτελεί επέκταση του υπάρχοντος μοντέλου CIM με τις προσθήκες που περιγράφονται παρακάτω.

Για την επέκταση του symbIoTe CIM μοντέλου με σχετικές οντότητες που εμφανίζονται στα μοντέλα του SAREF και NGSI-LD που είναι τα πιο κοντινά και χρησιμοποιούνται ευρέως εξετάζονται οι ομοιότητες και διαφορές στον ορισμό οντοτήτων που υπάρχουν ανάμεσα σε αυτές και στο symbIoTe CIM μοντέλο.

***Αντιστοίχιση symbIoTe CIM και SAREF***

Στο πλαίσιο αυτό, εντοπίσαμε τις παρακάτω οντότητες που αποτελούν μέρος του SAREF μοντέλου και εμφανίζονται ταυτόχρονα και στο symbIoTe CIM:

- *Device*. Ως saref:Device ορίζεται μία συσκευή, η οποία είναι σχεδιασμένη να επιτελεί κάποιο έργο (Εικόνα 12). Ομοίως, ως core:Device στο symbIoTe CIM ορίζουμε ένα αντικείμενο που σχετίζεται με την καταγραφή μια μέτρησης. Ως ιδιότητες του core:Device ορίζονται οι: name και description (Εικόνα 16), ενώ αυτές του saref:Device είναι οι: hasDescription, hasManufacturer, hasModel. Και στα δύο μοντέλα πληροφορίας η οντότητα Device έχει σαν υπο-οντότητες τις Sensor και Actuator.
- *Sensor*. Ως saref:Sensor ορίζεται μια συσκευή, η οποία ανιχνεύει και ανταποκρίνεται σε γεγονότα ή αλλαγές που εντοπίζονται στο φυσικό περιβάλλον, όπως για παράδειγμα το φώς, η κίνηση, η θερμοκρασία, κλπ. Το symbIoTe CIM δίνει ίδιο ορισμό στην οντότητα Sensor (Και προέρχεται από την SSN οντολογία), και διαθέτει τις ιδιότητες hasObservation και observesProperty, που την συνδέουν με τις οντότητες Observation και Property αντιστοίχως.
- *saref:Actuator.* Ως saref:Actuator ορίζεται ένα είδος Device, το οποίο έχει την δυνατότητα να ελέγξει έναν μηχανισμό ή ένα σύστημα εκτελώντας μία λειτουργία. Ο ίδιος ορισμός ισχύει και για το core:Actuator του symbIoTe CIM, τον οποίο δανείζεται από την οντολογία SOSA, και διαθέτει επιπλέον την ιδιότητα hasCapability, για να δηλώσει την ακριβή λειτουργία του core:Actuator.
- *saref:Property.* Ως saref:Property ορίζεται η ποιότητα ενός χαρακτηριστικού ή μιας ιδιότητας, η οποία μπορεί να μετρηθεί. Όμοιος είναι και ο ορισμός του core:Property που ορίζεται στο CIM, και είναι δανεική οντότητα από την οντολογία SSN. Ωστόσο, η διαφοροποίηση μεταξύ των δύο οντοτήτων έγκειται στο ότι το cim:Property ορίζει επιπλέον σαν ιδιότητες το name και description.
- *saref:Measurement.* Ως saref:Measurement ορίζεται η μετρήσιμη τιμή ενός saref:Property, ενώ επίσης συνδέεται με την οντότητα saref:UnitOfMeasure, μέσω της ιδιότητας saref:isMeasuredIn (Σχήμα 4). Στην ίδια λογική, η αντίστοιχη οντότητα του symbIoTe CIM είναι η core:ObservationValue, η οποία συνδέεται με την core:Observation, μέσω της hasObservationValue.
- *saref:UnitOfMeasure.* Ως saref:UnitOfMeasure ορίζεται η οντότητα που αφορά το πρότυπο μέτρησης μια ποσότητας ενός χαρακτηριστικού ή ενός Property. Η αντίστοιχη οντότητα στο symbIoTe CIM είναι ή core:UnitOfMeasurement, η οποία εκτός του ονόματος και της περιγραφής (name, description αντίστοιχα) ορίζει και το symbol σαν επιπλέον χαρακτηριστικό σχέσης με την saref:UnitOfMeasure.

Για τις οντότητες εκείνες που παρουσιάζουν ομοιότητες μεταξύ των μοντέλων saref, saref4city και symbIoTe CIM (συγκεκριμένα για τις οντότητες Device, Sensor, Actuator, Property, Measurement και UnitOfMeasure), πραγματοποιήθηκαν δηλώσεις του τύπου rdfs:equivalentClass ή rdfs:subclassOf, ώστε να δηλώσουμε ρητά ισότιμες οντότητες, ή υπο-οντότητες αντίστοιχα. Παρακάτω, ακολουθεί ένα απόσπασμα του μοντέλου πληροφορίας που αποτυπώνει τις παραπάνω αντιστοιχίες μεταξύ του saref και του symbIoTe CIM μοντέλου.

```
###  https://w3id.org/saref#Device

saref:Device owl:equivalentClass core:Device .

###  https://w3id.org/saref#Sensor

saref:Sensor owl:equivalentClass core:Sensor .

###  https://w3id.org/saref#Actuator

saref:Actuator owl:equivalentClass core:Actuator .

###  https://w3id.org/saref#Property

saref:Property owl:equivalentClass core:Property .

###  https://w3id.org/saref#UnitOfMeasure
saref:Measurement owl:equivalentClass core:ObservationValue .

###  https://w3id.org/saref#UnitOfMeasure
saref:UnitOfMeasure owl:equivalentClass core:UnitOfMeasurement .
```

*Εικόνα 21: Αντιστοιχίες saref-CIM στο τελικό μοντέλο πληροφορίας.*

Τέλος, από το μοντέλο saref4city εισαγεται η οντότητα city. Στην Εικόνα 22 δίνεται το απόσπασμα του μοντέλου πληροφορίας που αποτυπώνει την εισαγωγή της νέα οντότητας.

- **City**: οντότητα που σχετίζεται με την έννοια της πόλης και εισάγεται από το μοντέλο πληροφορίας saref4city. Η οντότητα saref4city:City κληρονομεί από τις οντότητες geosp:SpatialObject και της geosp:Feature, του μοντέλου πληροφορίας Geosparql (http://www.opengis.net/ont/geosparql ).

```
###################################################
### City definition.
###################################################

###  https://w3id.org/def/saref4city#City
:City a owl:Class ;
    rdfs:comment "A city is a large human settlement(https://en.wikipedia.org/wiki/City)"@en ;
    rdfs:label "City"@en;
        rdfs:subClassOf saref4city:AdministrativeArea;
        rdfs:subClassOf core:Location,
                        [ rdf:type owl:Restriction ;
                          owl:onProperty core:name ;
                          owl:onDataRange xsd:string ;
                          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                        ],
                        [ rdf:type owl:Restriction ;
                          owl:onProperty core:description ;
                          owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
                          owl:onDataRange rdf:langString
                        ] .
```

*Εικόνα 22: Η οντότητα City στο τελικό μοντέλο πληροφορίας.*

Σύμφωνα λοιπόν με την παραπάνω μελέτη των αντιστοιχιών μεταξύ saref και saref4city με το symbIoTe CIM, το symbIoTe CIM εμπλουτίζεται με την εισαγωγή ή αντιστοίχιση των οντοτήτων από το saref4city ή saref όπως περιγράφηκε παραπάνω.

***Αντιστοίχιση symbIoTe CIM και NGSI-LD***

Τέλος, στην προσπάθεια επέκτασης του symbiote CIM μοντέλου, η χρήση του μοντέλου NGSI-LD δεν κρίθηκε εφικτή, καθώς οι δηλώσεις αντίστοιχων μοντέλων παρουσιάζουν βασικές

διαφορές σειριοποίησης (serialization). Ωστόσο, μελετήθηκαν παραδείγματα για την αντιστοίχιση μεταξύ των δυο οντοτήτων, κάποια από τα οποία παρουσιάζονται παρακάτω. Συνεπώς, η μεθοδολογία semantic mapping μπορεί να αξιοποιηθεί για την αντιστοίχιση μοντέλων πληροφορίας πλατφορμών (ή και ομοσπονδιών) που ακολουθούν το μοντέλο NGSI-LD με το μοντέλο του IoTFeds για την επίτευξη by-mapping διαλειτουργικότητας.
Το μοντέλο NGSI-LD για την έξυπνη στάθμευση καλύπτει μια γκάμα από use cases για τα πιθανά σενάρια στάθμευσης σε μια έξυπνη πόλη. Όπως είδαμε παραπάνω, ορίζονται πέντε διαφορετικά μοντέλα για το Smart Parking. Συγκεκριμένα το OffStreetParking, OnStreetParking, ParkingAccess, ParkingGroup και ParkingSpot.
Εδώ θα παρουσιάσουμε τις αντιστοιχίες μεταξύ του μοντέλου NGSI-LD για το ParkingSpot και του βασικού μοντέλου CIM του symbIoTe.
Τα βασικά και συγχρόνως υποχρεωτικά πεδία του ParkingSpot είναι τα ακόλουθα:

- *id*
- *type*
- *location*
- *status*
- *category*
- *refParkingSite*

*To ParkingSpot* μπορεί να αντιστοιχηθεί στο πεδίο core:Location, ή να δηλωθεί σαν subclass αυτού *(rdfs:subClassOf core:Location).* Με αυτόν τον τρόπο μπορούμε να επιτύχουμε την αντιστοιχία των επιμέρους πεδίων του μοντέλου.
Συγκεκριμένα, το *id,* το οποίο αποτελεί ένα ξεχωριστό αναγνωριστικό της περιγραφόμενης οντότητας. αντιστοιχίζεται στο πεδίο , id του core:Location του symbIoTe CIM μοντέλου.
Στο NGSI-LD μοντέλο εξ 'ορισμού το πεδίο *type* παίρνει την τιμή ParkingSpot. Εφόσον το ParkingSpot δηλωθεί ως core:Location, υπάρχει ένα προς ένα αντιστοιχία.
Το *location* πεδίο είναι σε geo:json μορφή (κατηγοριοποίηση), και δέχεται δύο παραμέτρους/συνιστώσες για γεωγραφικό μήκος και πλάτος αντίστοιχα (longitude, latitude). Αντίστοιχα, το symbIoTe CIM διαθέτει την οντότητα WGS84Location, που διαθέτει τα προαναφερθέντα πεδία-συνιστώσες καθώς και αυτό για το γεωγραφικό υψόμετρο (altitude).
Στο NGSI-LD το πεδίο *status* μπορεί να δεχθεί τις ακόλουθες fixed τιμές: "closed", "free", "occupied", "unknown". Ουσιαστικά το πεδίο αυτό συνδυάζεται με το πεδίο *refDevice* του ParkingSpot μοντέλου, το οποίο μπορεί να αποδοθεί στο symbIoTe μοντέλο σαν υπο-οντότητα ενός Sensor (για να μπορεί να δέχεται observations) και ενός actuator, ώστε να μπορούν να του αποδίδονται Capabilities (on, off, free, occupied κλπ.). Το πεδίο *category* του NGSI-LD μοντέλου περιγράφει το είδος του σημείου στάθμευσης και μπορεί να πάρει την τιμή "onStreet" ή "offStreet". Ομώνυμα πεδία status και category έχουν μοντελοποιηθεί για το μοντέλο πληροφορίας του IoTFeds, ώστε να αντανακλά αυτές τις πληροφορίες, με την διαφορά ότι ορίζονται σαν string και δεν επιδέχονται συγκεκριμένες τιμές όπως στο NGSI-LD μοντέλο.
Τέλος το *refParkingSite*, το οποίο περιγράφει μια την ευρύτερηπεριοχή, η οποία εμπερικλείει πολλάParkingSpot. Επομένως, στο μοντέλο μας εντάξαμε μια οντότητα με την ίδια ονομασία (*refParkingSite*), η οποία συνδέεται με την οντότητα ParkingSpot μέσω του object propery *hasParkingSpots* (Εικόνα 16).
Για να καταστεί πιο σαφής η αντιστοίχιση των επιμέρους πεδίων μεταξύ των δύο μοντέλων, παρακάτω παρουσιάζονται δύο παραδείγματα του NGSI-LD ParkingSpot και του symbIoTe CIM (σε πρώτο στάδιο, πριν την αντιστοίχιση των μοντέλων).

```
{
  "id": "santander:daoiz_velarde_1_5:3",
  "type": "ParkingSpot",
  "name": "A-13",
  "location": {
    "type": "Point",
    "coordinates": [-3.80356167695194, 43.46296641666926]
  },
  "status": "free",
  "category": ["onStreet"],
  "refParkingSite": "santander:daoiz_velarde_1_5"
}
```


*Εικόνα 23: Παράδειγμα ενός NGSI-LD ParkingSpot σχήματος.*


```
{
   "@id" : "http://www.symbiote-h2020.eu/ontology/platforms/Platform1/location/santander:daoiz_velarde_1_5:3",
    "@type" : "core:WGS84Location",
    "name" : "A-13",
    "description" : "onStreet",
    "alt" : " ",
    "lat" : "-3.80356167695194",
    "long" : "43.46296641666926"
 |}
```


*Εικόνα 24: Παράδειγμα ενός symbIoTe CIM Location.*

Πέρα από αυτά τα βασικά πεδία, το ParkingSpot μοντέλο ορίζει και τα ακόλουθα: width, length, refParkingGroup και timeInstant. Τα πεδία για width και length, τα οποία ορίζουν το πλάτος και μήκος της θέσης στάθμευσης αντίστοιχα, θα μπορούσαν να δηλωθούν επεκτείνοντας το symbIoTe CIM μοντέλο και δηλώντας τα σαν τυπου core:Property (συγκεκριμένα ως rdfs:subClassOf core:Property). Το refParkingGroup, το οποίο υποδηλώνει μια συστάδα θέσεων στάθμευσης, θα μπορούσε να εξισωθεί με το core:WKTLocation του CIM, και τέλος το time:Instant υπάρχει σαν οντότητα ήδη στο symbIoTe CIM, αφού εισάγεται από το time:ontology[27].

Στα πλαίσια του NGSI-LD μοντέλου για την έξυπνη πόλη, υπάρχουν και μοντέλα πληροφορίας που καλύπτουν συσκευές και υπηρεσίες έξυπνου φωτισμού[28]. Για τις ανάγκες του έργου IoTFeds πραγματοποιήσαμε μια έρευνα για την αντιστοιχία του Streetlight json μοντελου του NGSI-LD[29] με το symbIoTe CIM και τα μοντέλα πληροφορίας BIM του symbIoTe που καλύπτουν τις ανάγκες για το έξυπνο σπίτι (Smart Residence) και την έξυπνη μετακίνηση (Smart Mobility)[30].

Τα υποχρεωτικά πεδία του μοντέλου Streetlight είναι τα:

- *id, που* αντιπροσωπεύει ένα ξεχωριστό αναγνωριστικό της περιγραφόμενης οντότητας. Αντίστοιχο σε αυτό πεδίο είναι το Resource.id του symbIoTe CIM.
- *type,* που εξ'ορισμού εδώ παίρνει την τιμή Streetlight.
- *location,* που όπως είδαμε παραπάνω αποτελεί αναφορά στο geo:json μοντέλο και στο symbIoTe CIM η αντίστοιχη οντότητα είναι η WGS84Location
- *status,* που αποτελεί αναφορά στην γενική κατάσταση του φωτισμού σε μία περιοχή ή γεωγραφικό σημείο και μπορεί να πάρει μια από τις ακόλουθες τιμές "brokenLantern", "columnIssue","defectiveLamp", "ok". Ομοίως με την περίπτωση του ParkingSpot, το Streetlight μοντέλο διαθέτει ένα πεδίο με το όνομα *refDevice*, που αντιστοιχεί στην συσκευή ή συσκευές που χρησιμοποιούνται για να ελέγχουν μια περιοχή φωτισμού (streetlight). Συνεπώς το refDevice αντιστοιχίζεται απευθείας στην Device_οντότητα

[27] http://www.w3.org/2006/time
[28] https://github.com/smart-data-models/dataModel.Streetlighting
[29] https://github.com/smart-data-models/dataModel.Streetlighting/blob/6e8080344a857e0ddf291c308cab1c4aa2969452/Streetlight/schema.json
[30] http://www.symbiote-h2020.eu/ontology

του symbIoTe CIM, η οποία είναι υπερ-οντότητας στις οντότητες Sensor και Actuator. Στο μοντέλο πληροφορίας του IoTFeds δημιουργήσαμε μία οντότητα με το όνομα LightSensor, η οποία σημασιολογικά αντιστοιχεί στο Streetlight του NGSI-LD. Για να αναπαραστήσουμε την ιδιότητα του *status*, δημιουργήσαμε ένα ομώνυμο πεδίο στην οντότητα LightSensor, για να καταγράφει την κατάσταση (status) του φωτός. Με τον ίδιο τρόπο, κρίθηκε ότι είναι επίσης σημαντικό να αναπαρασταθεί ευθέως στο νέο μοντέλο πληροφορίας και το πεδίο *powerState* του NGSI-LD μοντέλου, το οποίο περιγράφει την κατάσταση (από πλευράς ισχύος ρεύματος) του Streetlight, και μπορεί να πάρει τις τιμές "bootingup"*, "low", "off" και "on"*. Στο μοντέλο πληροφορίας του IoTFeds το πεδίο *powerState* μπορεί να πάρει οποιαδήποτε τιμή string.
Εν συντομία, τα υπόλοιπα πεδία του Streetlight μοντέλου μπορούν να αντιστοιχηθούν ή να αναπαρασταθούν ως εξής με το symbIoTe CIM.

- *refStreetlightControlCabinet,* ως core:WKTLocation, καθώς υποδηλώνει την τοποθεσία του πίνακα ελέγχου που αντιστοιχεί στην θέση φωτισμού.
- *refStreetlightGroup,* ως core:WKTLocation, καθώς υποδηλώνει την συστάδα φωτισμού στην οποία ανήκει η συγκεκριμένη θέση.
- *locationCategory,* ως core:WKTLocation, καθώς υποδηλώνει την τοποθεσία της πηγής φωτισμού (π.χ. πάρκο, γέφυρα, παιδότοπος κλπ.).
- Τα πεδία *dateLastLampChange, dateLastSwitchingOn, dateLastSwitchingOff, observationDateTime* ως time:Instant οντότητες, καθώς αφορούν date-time οντότητες.
- Τα πεδία *lanternHeight, illuminanceLevel, powerConsumption, current, powerRating, powerFactor, voltage, feederPillarNum, streetPoleNum, feeder, deviceInfo ως υπο-οντότητες του core:Property.*
- Το πεδίο municipalityInfo και όλα τα επιμέρους υπο-πεδία του μπορούν να αναπαρασταθούν και αυτά σαν subClassOf core:Property ή να εξισωθούν στο s4city:AdministrativeArea (Εικόνα 11).

Για λόγους οικονομίας χώρου, οι παραπάνω ιδιότητες, οι οποίες υποδεικνύεται ότι θα μπορούσαν να αναπαρασταθούν ως υπο-οντότητες του core:Property δεν αναπαρίστανται στην εικόνα του μοντέλου πληροφορίας IoTFeds (Εικόνα 28).
Τέλος, μελετήσαμε και το μοντέλο πληροφορίας για σημεία ξεχωριστού ενδιαφέροντος σε μια έξυπνη πόλη, και συγκεκριμένα το PointOfInterest[31].
Και εδώ τα απαραίτητα πεδία στην δήλωση του μοντέλου και των επιμέρους resources μετέπειτα είναι τα

- *id*,
- *type* (εξορισμού παίρνει την τιμή PointOfInterest),
- *category*, που μπορεί να πάρει οποιαδήποτε τιμή (string) που να ανταποκρίνεται σε μία νοητή ταξινομία σημείων αναφοράς,
- *name*, όπου μπορεί να πάρει οποιαδήποτε τιμή (π.χ. «coffeeshop nearby»).

Η PointOfInterest οντότητα μπορεί να αντιστοιχηθεί στην οντότητα του core:Location του symbIoTe CIM (Εικόνα 11), καθώς διαθέτει τα πεδία για id, name και description, τα οποία αντιστοιχίζονται στα *PointOfInterest.id, PointOfInterest.name* και *PointOfInterest.category* πεδία αντίστοιχα.
Τα πεδία warId, zoneId, refSeeAlso και additionalInfoURL μπορούν να αναπαρασταθούν σαν υπο-οντότητες του core:Property του μοντέλου symbIoTe CIM.

[31] https://github.com/smart-data-models/dataModel.PointOfInterest

Τέλος, μελετήσαμε και τα μοντέλα πληροφορίας που καλύπτουν την περιγραφή των καιρικών συνθηκών καθώς και την πρόβλεψη καιρικών φαινομένων, δηλαδή τα WeatherObserved[32] και WeatherForecast[33] NGSI-LD μοντέλα αντίστοιχα.
Τα βασικά πεδία του WeatherObserved μοντέλου και οι αντιστοιχίες του με το symbIoTe-CIM μοντέλο περιγράφονται παρακάτω.

- *id, που* αντιπροσωπεύει ένα ξεχωριστό αναγνωριστικό της περιγραφόμενης οντότητας και αντιστοιχεί στο πεδίο είναι το Resource.id του symbIoTe CIM.
- *type,* που αντιστοιχεί στον τύπο της οντότητας (εδώ WeatherObserved), και μπορεί να αντιστοιχηθεί στο Resource:type του symbIoTe CIM
- *dateObserved*, που προσδιρίζει την ημερομηνία καταγραφής του καιρικού φαινομένου και αντοιστοιχεί στην time:instant οντότητα του symbIoTe CIM
- *location,* που προσδιορίζει την τοποθεσία αναφορά των μετρήσεων καιρού και αντιστοιχεί στο core:Location του symbIoTe CIM.

Το πεδίο *refDevice* του WeatherObserved μοντέλου αποτελεί αναφορά στην συσκευή ή αισθητήρα που πραγματοποίησε τις μετρήσεις για τις παραμέτρους καιρού (αυτές αναφέρονται παρακάτω). Συνεπώς, το refDevice αντιστοιχίζεται στο Device του symbIoTe CIM και μπορεί να δεχτεί observations και properties που καταγράφει μία συσκευή. Επομένως με αυτόν τον τρόπο, τα πεδία id, typr, dateObserved και location αντιστοιχίζονται σε ήδη υπάρχουσες οντότητες του symbIoTe CIM μοντέλου, όπως περιγράφηκε πάνω.
Τα πεδία *temperature*, *feelsLikesTemperature, relativeHumidity, precipitation, solarRadiation, pressureTendency, dewPoint, streamGauge, snowHeight, uVIndexMax, aqiMajorPollutant, aqiMajorPollutantForecast, airTemperatureForecast, precipitationForecast, airQualityIndex, relativeHumidityForecast, illuminance, airQualityIndexForecast, airTemperatureTSA* του NGSI-LD WeatherObserved μοντέλου θα αναπαρασταθούν σαν υπο-οντότητες του core:Property του symbIoTe CIM μοντέλου.
Το μοντέλο πληροφορίας WeatherForecast αποτελείται από τα ακόλουθα πεδία, για τα οποία ακολουθεί σύντομη περιγραφή της αντιστοιχίας ή αναπαράστασής του στο symbIoTe CIM.

- *id, που* αντιπροσωπεύει ένα ξεχωριστό αναγνωριστικό της περιγραφόμενης οντότητας και αντιστοιχεί στο πεδίο είναι το Resource.id του symbIoTe CIM.
- *type,* που αντιστοιχεί στον τύπο της οντότητας (εδώ WeatherForecast), και μπορεί να αντιστοιχηθεί στο Resource:type του symbIoTe CIM
- *dateIssued*, που προσδιρίζει την ημερομηνία καταγραφής του καιρικού φαινομένου και αντοιστοιχεί στην time:instant οντότητα του symbIoTe CIM
- *address,* που προσδιορίζει την τοποθεσία αναφορά στον τόπο πρόβλεψης και αντιστοιχεί στο core:Location του symbIoTe CIM.

Τα πεδία *dateRetrieved*, *dateIssued, validFrom, validTo* του NGSI-LD μοντέλου πληροφορίας αποτελούν date-time δομές οπότε μπορούν να αποδοθούν ως time:Instant οντότητες του symbIoTe CIM μοντέλου.
Τα πεδία *dayMaximum, dayMinimum, uVIndexMax*, *precipitation* του NGSI-LD μοντέλου μπορούν να αναπαρασταθούν σαν υπο-οντότητες του core:Property του symbIoTe CIM μοντέλου.
Πιο συγκεκριμένα, το νέο μοντέλο πληροφορίας IoTFeds εμπλουτίζεται με τις ακόλουθες προσθήκες:

- **ParkingSite:** οντότητα που ορίζει μια περιοχή που αποτελείται από πολλές θέσης για στάθμευση και ορίζεται έτσι μια ζώνη στάθμευσης.
- **ParkingSpot:** οντότητα που αναπαριστά μια θέση στάθμευσης.
- **LightPoint:** οντότητα που αναπαριστά ένα σημείο στο οποίο διατίθεται κάποιο σύστημα φωτισμού.

[32] https://github.com/smart-data-models/dataModel.Weather/
[33] https://github.com/smart-data-models/dataModel.Weather/tree/7a607dfeaebd3b7164400f46095f86f256062f6b/WeatherForecast

- **LightPoint:** οντότητα που αναπαριστά ένα σημείο στο οποίο διατίθεται κάποιο σύστημα φωτισμού.

Παρακάτω ακολουθούν οι αποδόσεις των παραπάνω οντοτήτων στο μοντέλο πληροφορίας.

```
################################################
### smart parking
################################################

###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#refParkingSite

:refParkingSite rdf:type owl:Class ;
                      rdfs:subClassOf core:Location,
                      rdfs:subClassOf core:WGS84Location ,
                                      [ rdf:type owl:Restriction ;
                                        owl:onProperty core:name ;
                                        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                        owl:onDataRange xsd:string
                                      ],
                                      [ rdf:type owl:Restriction ;
                                        owl:onProperty :hasParkingSpots ;
                                        owl:onClass :ParkingSpot ;
                                        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                                      ]  .

### http://www.symbiote-h2020.eu/ontology/smartcity_v1#ParkingSpot
:ParkingSpot rdf:type owl:class ;
         rdfs:subClassOf :refParkingSite ,
                         [ rdf:type owl:Restriction ;
                             owl:onProperty :id ;
                             owl:onDataRange xsd:string ;
                             owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                           ],
                           [ rdf:type owl:Restriction ;
                             owl:onProperty :status ;
                             owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
                             owl:onDataRange rdf:langString
                           ],
                           [ rdf:type owl:Restriction ;
                             owl:onProperty :category ;
                             owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
                             owl:onDataRange rdf:langString
                           ].
```

*Εικόνα 25: Οντότητες smart parking στο τελικό μοντέλο πληροφορίας.*

```
       ######################################################
       ### Smart lights
       ######################################################


       ###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#LightSensor

       :LightSensor rdf:type owl:Class ;

                            rdfs:subClassOf core:Sensor;
                            rdfs:subClassOf core:Actuator,
                                      [ rdf:type owl:Restriction ;
                                    owl:onProperty core:status ;
                                    owl:onDataRange xsd:string ;
                                    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                                        ],
                                        [ rdf:type owl:Restriction ;
                                    owl:onProperty core:powerState ;
                                    owl:onDataRange xsd:string ;
                                    owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                                        ] .

       ###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#LightPoint
       :LightPoint rdf:type owl:Class ;
                   rdfs:subClassOf core:WKTLocation    .
```

*Εικόνα 26: Οντότητες για smart lighting στο τελικό μοντέλο πληροφορίας.*

***Προσθήκη άλλων οντοτήτων στο τελικό μοντέλο πληροφορίας***

Επιπλέον ορίζονται οι ακόλουθες οντότητες, οι οποίες αποτελούν υποκλάσεις (subclasses) της βασικής οντότητας του symbIoTe CIM μοντέλου Sensor:

- **AirQualitySensor,** οντότητα για την καταγραφή της ποιότητας του αέρα.
- **DustSensor,** οντότητα για την καταγραφή/καταμέτρηση σκόνης.
- **HumiditySensor,** οντότητα για την καταγραφή της υγρασίας.
- **LightSensor,** οντότητα που είναι ταυτόχρονα αισθητήρας και ενεργοποιητής για το φως.
- **TemperatureSensor,** οντότητα για την καταγραφή της θερμοκρασίας.
- **SmokeSensor,** οντότητα για την ανίχνευση καπνού.

Η παρακάτω εικόνα παρουσιάζει τις οντότητες που δημιουργήθηκαν για να αναπαραστήσουν τους παραπάνω αισθητήρες.

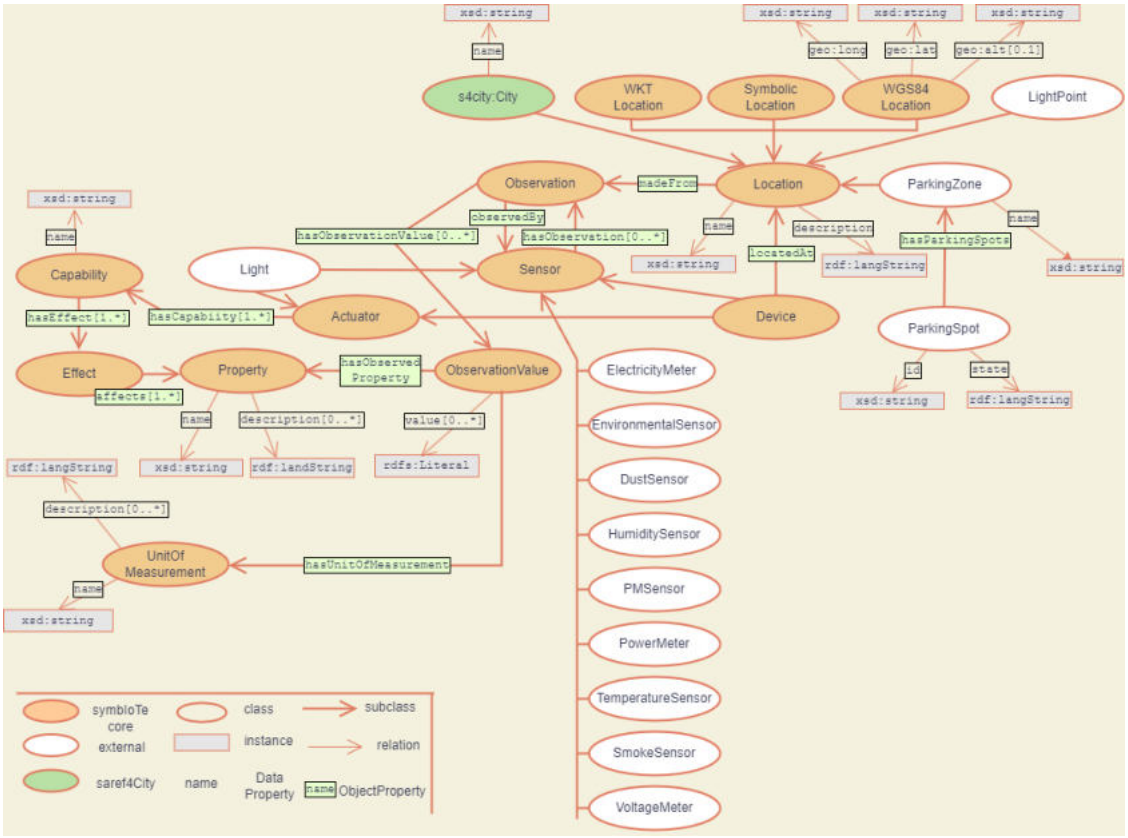
```
#################################################################
#
#    Classes
#
#################################################################


###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#AirQualitySensor

:AirQualitySensor rdf:type owl:Class ;

                          rdfs:subClassOf core:Sensor .

###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#HumiditySensor

:HumiditySensor rdf:type owl:Class ;

                      rdfs:subClassOf core:Sensor .

###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#LightSensor

:LightSensor rdf:type owl:Class ;

                      rdfs:subClassOf core:Sensor;
                      rdfs:subClassOf core:Actuator,
                                [ rdf:type owl:Restriction ;
                            owl:onProperty core:status ;
                            owl:onDataRange xsd:string ;
                            owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                                ],
                                [ rdf:type owl:Restriction ;
                            owl:onProperty core:powerState ;
                            owl:onDataRange xsd:string ;
                            owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
                                ] .


###  http://www.symbiote-h2020.eu/ontology/smartcity_v1#TemperatureSensor

:TemperatureSensor rdf:type owl:Class ;

                          rdfs:subClassOf core:Sensor .
```

*Εικόνα 27: Υποκλάσεις της οντότητας sensor στο τελικό μοντέλο πληροφορίας.*

***Τελικό μοντέλο πληροφορίας IoTFeds***

Το τελικό μοντέλο πληροφορίας του IoTFeds (Smart City BIM)[34] αποτελείται από το symbIoTe CIM και τις επεκτάσεις με βάση το μοντέλα που περιγράφηκαν παραπάνω όπως παρουσιάζονται στην παρακάτω εικόνα:

[34] http://www.symbiote-h2020.eu/ontology/smartcity_v1




*Εικόνα 28:Βασικές οντότητες στο μοντέλο πληροφορίας του IoTFeds.*

Συνοψίζοντας οι προσθήκες στο νέο μοντέλο πληροφορίας όπως παρουσιάστηκαν στην ενότητα αυτή αφορούν την αντιστοίχιση των βασικών οντοτήτων του SAREF ή SAREF4CITY μοντέλου με τις αντίστοιχες του symbIoTe CIM, οντότητες για την έξυπνη στάθμευση και έξυπνο φωτισμό από την αντιστοίχιση του NGSI-LD καθώς και νέους τύπους αισθητήρων στον τομέα της Έξυπνης Πόλης που δεν καλύπτονται από τον υπάρχον μοντέλο πληροφορίας ενσωματώνοντας και τις αντίστοιχες ιδιότητες (properties) που αφορούν τις μετρικές που παρακολουθούν και καταγράφουν. Το νέο μοντέλο πληροφορίας είναι διαθέσιμο στην ιστοσελίδα του symbIoTe (http://www.symbiote-h2020.eu/ontology/bim/smartcity) καθώς και στην ιστοσελίδα του έργου IoTFeds (http://www.iotfeds.gr/#Information-Models ).

# 5 Αλληλεπίδραση και Διεπαφές των Επιμέρους Στοιχείων Λογισμικού

Στην ενότητα αυτή περιγράφεται η υλοποίηση των λειτουργικοτήτων διαχείρισης ομοσπονδίας και των εμπλεκόμενων στοιχείων λογισμικού των υποσυστημάτων που συνοδεύουν τον αντίστοιχο κώδικα Ανοιχτής Πηγής, εστιάζοντας στις τροποποιήσεις και την ανάπτυξη των νέων διεπαφών που έγινε για τις ανάγκες του έργου IoTFeds.

## *5.1 Μέθοδοι για τη διαχείριση ομοσπονδίας*

Στην ενότητα αυτή παρουσιάζονται οι διεπαφές που χρησιμοποιούνται για την διαχείριση της ομοσπονδίας στα επιμέρους υποσυστήματα.

### 5.1.1 Μέθοδοι σχετικά με το symbIoTe

Στον παρακάτω πίνακα συνοψίζονται οι διεπαφές που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια του symbIoTe και της επέκτασης του για το έργο IoTFeds και οι οποίες απαιτούνται για την υποστήριξη των IoT ομοσπονδιών και την διαχείριση τους (λειτουργικότητες για τους χρήστες, τις ομοσπονδίες και τους πόρους σε αυτές):

*Πίνακας 7: Endpoints του symbIoTe υποσυστήματος για τη διαχείριση ομοσπονδίας.*

| # | Inter-face | Name | Request from | Request to | Method | Endpoint /Queue | Payload | Response | Description |
|---|---|---|---|---|---|---|---|---|---|
| 1 | REST | Register user | IoTFeds User (Data consumer) | Administration | POST | /administration/register | CoreUser | Map<String, Object> | Request to the administration component to create a core user. Effectively like registering a user from the administration GUI |
| 2 | REST | User sign in | IoTFeds User | Administration | POST | /administration/user/login | (String, String) | Success message | Request for token |
| 3 | REST | Register platform | IoTFeds User (Data Producer) | Administration | POST | /administration/user/cpanel/register_platform | PlatformDetails | HttpHeaders | Register a platform to symbIoTe |
| 4 | REST | IoT resource registration | IoTFeds User (Data Producer) | symbIoTe API | POST | /symbioteapi/resources/add/l1 | iotFedsApiCloudResourceL1 | CloudResource | Register a resource to symbiote |
| | | | | | POST | /symbioteapi/resources/shareResource | shareResourceModel | CloudResource | Share a resource to a federation |
| | | | | | POST | /symbioteapi/resources/unshareResource | shareResourceModel | CloudResource | Unshare (remove) a resource from a federation |
| 5 | REST | IoT resource deletion | IoTFeds User (Data Producer) | symbIoTe API | DELETE | /symbioteapi/resources/deleteL1 | String | CloudResource | Delete L1 resource |
| | | | | | DELETE | /symbioteapi/resources/deleteL2 | String | CloudResource | DeleteL2 reosurce |
| | | | | | POST | /administration/user/cpanel/create_federation | Federation | Map<String, Federation> | Register a new federation to symbiote |
| 6 | REST | Fed creation | IoTFeds User (Data Producer) | Administration | GET | /administration/user/cpanel/list_federations | | List<Federation> | Return the list of federations registered in the system |
| 7 | REST | Join fed | IoTFeds User | Administration | POST | /administration/federation_vote_request /joinFederation | JoinFederationRequest(Username, FederationId) | Success message | Request from an organization/ platform to join a federation |
| | | | | | GET | /administration/federation_vote_request /allForUser | | List<JoinFederationRequest> | Returns the list of the join requests |
| | | | | | DELETE | /administration/federation_vote_request /removeUserFromFed | (String, String) | FederationVoteRequest | Returns the vote request that was created |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 8 | REST | Remove fed member | IoTFeds User | Administration | POST | /administration/federation_vote_request/updateFederationRules | UpdateRules | FederationVoteRequest | Returns the vote request that was created |
| 9 | REST | Update fed rules | IoTFeds User | Administration | POST | /administration/federation_vote_request/result | (String, Object) | Success message | The vote request is handled in symbiote |
| 10 | REST | Voting result | Baas | Administration | POST | /administration/user/cpanel/federation_invite | Invitation Request | <Federation ID, FederationWithInvitations> | Invite a user to an existing federation |
| 11 | REST | Invite to fed | IoTFeds User (Data producer) | Administration | POST | /administration/user/cpanel/leave_federation | RequestParams:federationId, organization | Success message | Leave from a federation you are currently in |
| 12 | REST | Leave fed | IoTFeds User | Administration | POST | /administration/user /cpanel/delete_federation | RequestParam: federationIdToDelete | Success message | Delete a federation |
| 13 | REST | Delete fed | IoTFeds User | Administration | | | | | |

### 5.1.2 Μέθοδοι σχετικά με το BaaS

Στη συγκεκριμένη ενότητα του παραδοτέου παρουσιάζονται αναλυτικά όλες οι μέθοδοι που σχετίζονται με τις λειτουργίες του BaaS, περιλαμβάνοντας λειτουργίες που σχετίζονται με το χρήστη, τις ομοσπονδίες, καθώς και τις ψηφοφορίες. Τα αντίστοιχα endpoints για κάθε επιμέρους λειτουργία παρουσιάζονται αναλυτικά στον ακόλουθο πίνακα (Πίνακας 8).

*Πίνακας 8: Endpoints που περιγράφουν όλες τις λειτουργίες του BaaS.*

| # | Inter-face | Name | Request from | Request to | Method | Endpoint /Queue | Payload | Response | Description |
|---|---|---|---|---|---|---|---|---|---|
| 1 | REST | registerUserToBc | symbiote | BaaS | POST | /baas/user/register_user | (String, String, String, String) | Message | Request for registering a user to BaaS |
| 2 | REST | getUserInfo | symbiote | BaaS | GET | /baas/user/get_user_info | String | (Object, Message) | Request user information from BaaS |
| 3 | REST | getAllUsers | symbIoTe | BaaS | GET | /baas/user/get_all_users | - | (Array of objects, Message) | Request information of all users on BaaS |
| 4 | REST | updateUserBalance | symbIoTe | BaaS | PATCH | /baas/user/update_user_balance | (String, Number) | Message | Request to update a user's balance |
| 5 | REST | registerPlatform | symbIoTe | BaaS | PATCH | /baas/user/register_platform | (String, String) | Message | Request to register a platform to BaaS by updating the associated user fields |
| 6 | REST | removePlatform | symbIoTe | BaaS | PATCH | /baas/user/remove_platform | (String, String) | Message | Request to remove a platform from a user |
| 7 | REST | removePlatformResource | symbIoTe | BaaS | DELETE | /baas/user/remove_platform_resources | String | Message | Request to remove all the resources of a platform |
| 8 | REST | registerDevice | symbIoTe | BaaS | PATCH | /baas/user/register_device | (String, String) | Message | Request to register a device to BaaS by updating the associated user fields |
| 9 | REST | removeDevice | symbIoTe | BaaS | PATCH | /baas/user/remove_device | (String, String) | Message | Request to remove a device from a user's platform |
| 10 | REST | deleteUser | symbIoTe | BaaS | DELETE | /baas/user/delete_user | String | (Object, Message) | Request to delete a user from BaaS |
| 11 | REST | registerFedToBc | symbIoTe | BaaS | POST | /baas/federation/register_fed | (String, String, String, Array[String], Object) | Message | Request to register a federation to BaaS |
| 12 | REST | getFedInfo | symbIoTe | BaaS | GET | /baas/federation/get_fed_info | String | (Object, Message) | Request federation information from BaaS |

*Εικόνα 39: Διεπαφές υποσυστήματος symbIoTe.*



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 13 | REST | getAllFeds | symbIoTe | BaaS | GET | /baas/federation/get_all_feds | - | (Array of Objects, Message) | Request information of all federations on BaaS |
| 14 | REST | leaveFed | symbIoTe | BaaS | PATCH | /baas/federation/leave_fed | (String, String) | (Object, Message) | User request to voluntarily leave a federation |
| 15 | REST | deleteFederation | symbIoTe | BaaS | DELETE | /baas/federation/delete_federation | (String, String) | (Object, Message) | Request to remove a federation from BaaS |
| 16 | REST | addFedMemberRequest | symbIoTe | BaaS | POST | /baas/voting/add_fed_member_request | (String, String, String) | Message | Request to add a member to a federation through voting |
| 17 | REST | removeFedMemberRequest | symbIoTe | BaaS | POST | /baas/voting/remove_fed_member_request | (String, String, String) | Message | Request to remove a member from a federation through voting |
| 18 | REST | updateFedRules | symbIoTe | BaaS | POST | /baas/voting/update_fed_rules | (String, String, Object) | Message | Request the update of the federation rules. A voting procedure is initialized. |
| 19 | REST | getVotingDescription | BaaS | IoTFeds voting user | GET | /baas/voting/get_voting_description | String | (Object, Message) | Request the description of the voting procedure |
| 20 | REST | registerVote | IoTFeds voting user | BaaS | POST | /baas/voting/register_vote | (String, String) | Message | Request to register the vote of a user to a procedure that they were invited to vote in |

## 5.2 Περιγραφή των επιμέρους στοιχείων λογισμικού και λειτουργικοτήτων

### 5.2.1 Υποσύστημα symbIoTe

Στα πλαίσια του IoTFeds ο χρήστης (όπως πάροχος ή καταναλωτής δεδομένων) αλληλεπιδρά με το σύστημα μέσω της γραφικής διεπαφής χρήστη του symbIoTe (Administration GUI) είτε μέσω Restful API του στοιχείου symbIoTeAPI. Το στοιχείο αυτό (symbIoTeAPI) αναπτύχθηκε στα πλαίσια του IoTFeds για να διευκολύνει την Restful επικοινωνία του χρήστη με τα εσωτερικά στοιχεία λογισμικού του symbIoTe και την ενσωμάτωση της επικοινωνίας με το BaaS. Στην ενότητα αυτή περιγράφονται τα σημεία πρόσβασης (endpoints) που εμπλέκονται και έχουν επεκταθεί για την υλοποίηση των ομοσπονδιών και των βασικών λειτουργικοτήτων για τη διαχείριση των ομοσπονδιών στην πλατφόρμα IoTFeds. Τα σημεία αυτά πρόσβασης

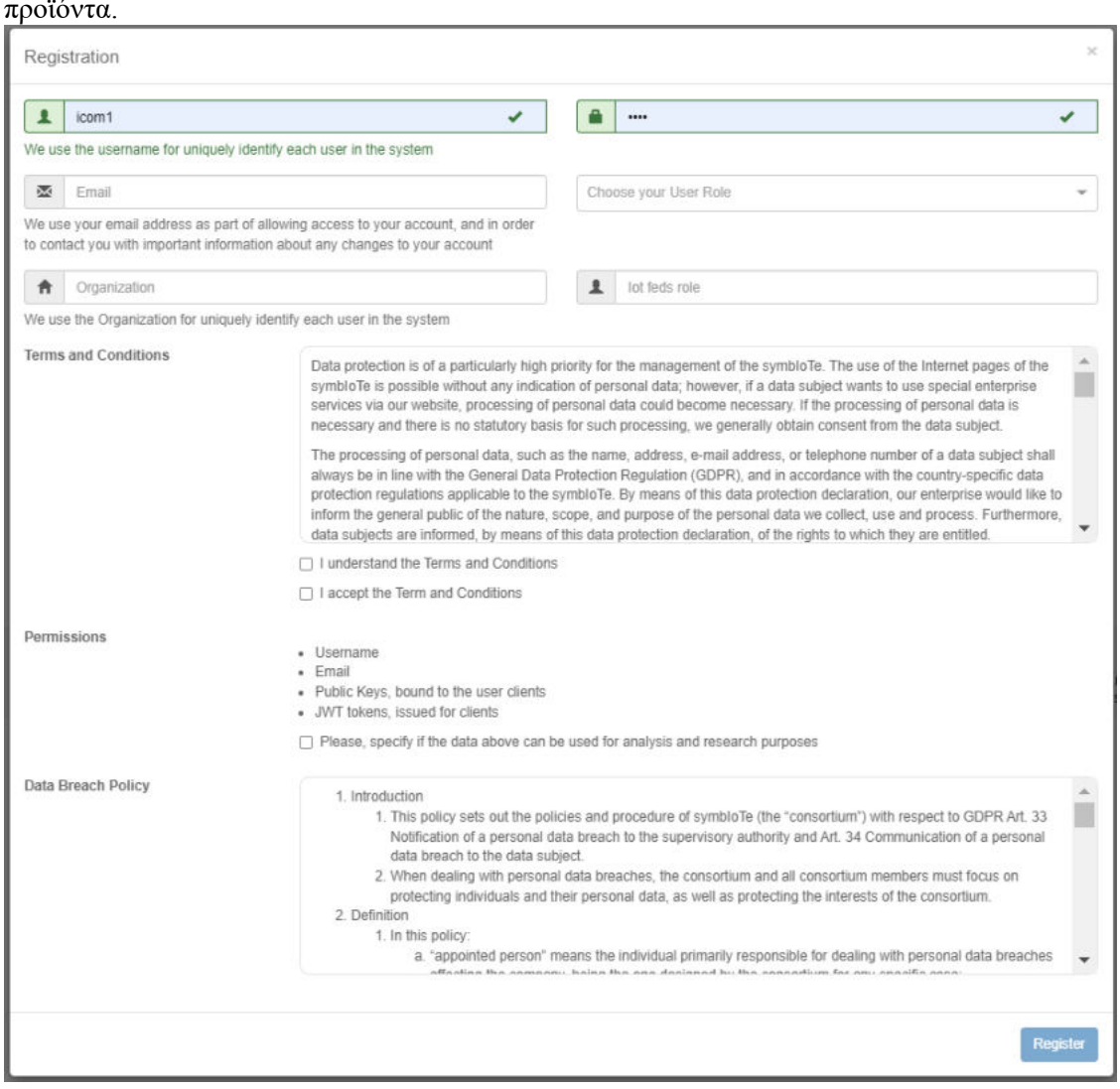
αφορούν κυρίως τη διαχείριση των χρηστών, πλατφορμών ή αλλιώς παρόχων δεδομένων και των πόρων τους καθώς και των ομοσπονδιών, των μελών και των κανόνων τους.

#### *5.2.1.1 Στοιχείο λογισμικού για τη διαχείριση ομοσπονδιών, χρηστών και πλατφορμών*

Η διαχείριση των ομοσπονδιών, των χρηστών και των πλατφορμών τους στο οικοσύστημα του symbIoTe γίνεται μέσα από τη γραφική διεπαφή χρήστη (Administration GUI)..

##### 5.2.1.1.1 Εγγραφή χρήστη στο symbIoTe

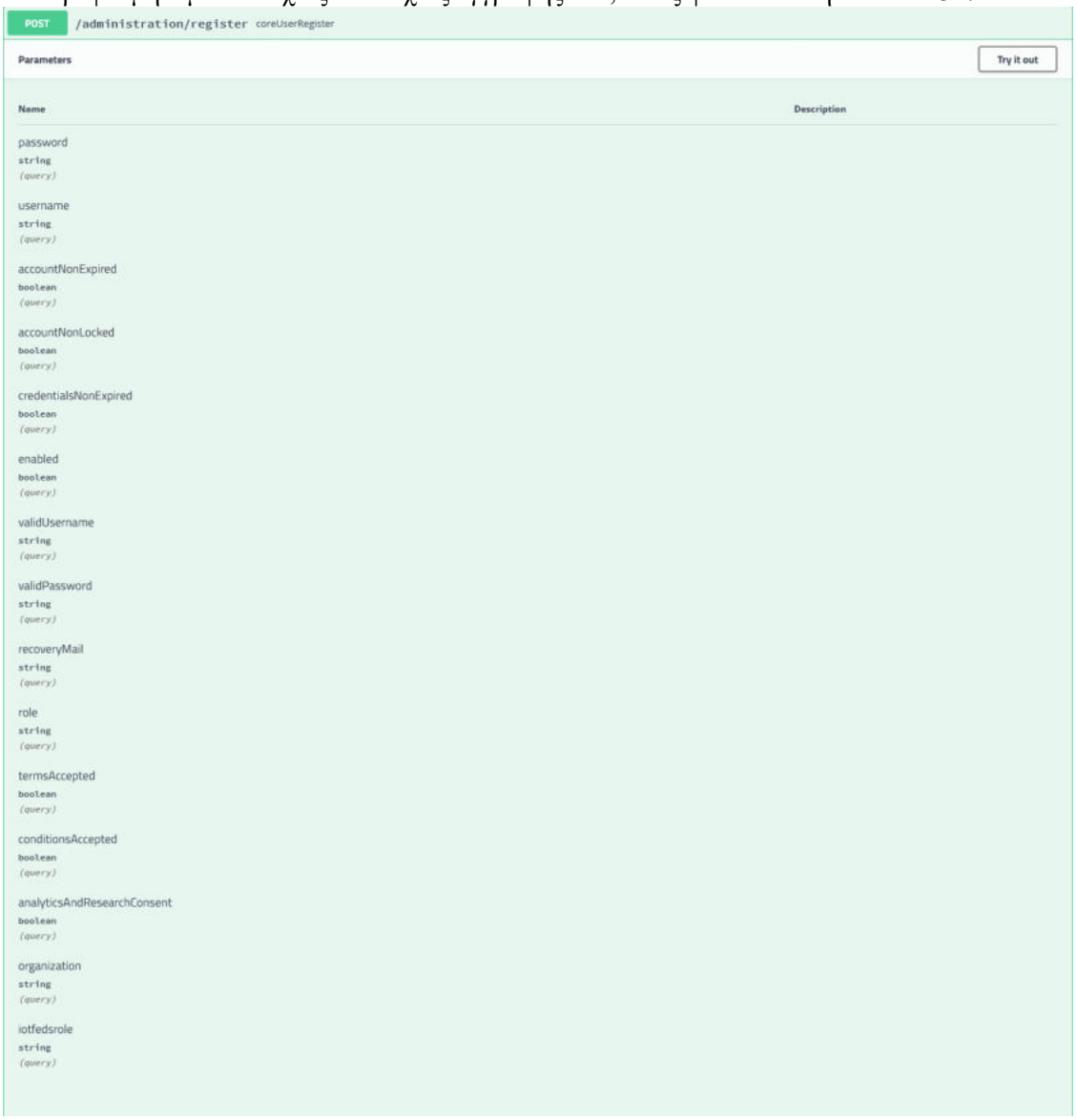
Η εγγραφή ενός χρήστη στο IoTFeds σύστημα γίνεται μέσω της γραφικής διεπαφής χρήστη του symbIoTe (Administration GUI). Στη παρακάτω εικόνα παρουσιάζεται η φόρμα εγγραφής του χρήστη, η οποία έχει επεκταθεί με την παράμετρο "IoTFeds role" που απαιτείται στο υποσύστημα BaaS. Χρήστες της IoTFeds πλατφόρμας μπορεί να είναι πάροχοι ή καταναλωτές δεδομένων (ένας οργανισμός). Ο ρόλος "service owner" απαιτείται στον ρόλο χρήστη (user role) για τη δημιουργία λογαριασμού κατόχου υπηρεσίας στο symbIoTe και συνεπώς την συμμετοχή στις ομοσπονδιακές αγορές της IoTFeds πλατφόρμας. Ο ρόλος "User" ως User Role για τη δημιουργία λογαριασμού καταναλωτή δεδομένων στο symbIoTe αρκεί για τη συμμετοχή του στην καθολική αγορά της IoTFeds πλατφόρμας και την πρόσβαση στα προκατασκευασμένα προϊόντα.




*Εικόνα 31:Εγγραφή χρήστη στο IoTFeds σύστημα.*

Τα στοιχεία που συμπληρώνει ο χρήστης προς εγγραφή αποστέλλονται στο επεκταμένο στοιχείο λογισμικού Administration του symbIoTe από το οποίο και ενημερώνεται το υποσύστημα BaaS για την καταγραφή των στοιχείων του χρήστη. Το endpoint του στοιχείου

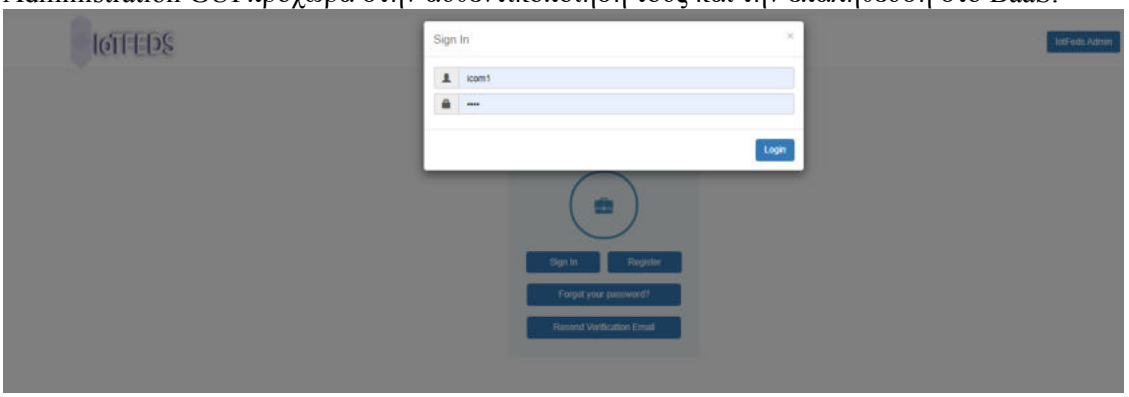
αυτού δέχεται από το Administration GUI τις παραμέτρους που έχει επιλέξει ο χρήστης και επιστρέφει μήνυμα επιτυχίας/αποτυχίας εγγραφής του, όπως φαίνεται στην Εικόνα 32.




*Εικόνα 32: Παράμετροι πληροφορίας χρήστη στο symbIoTe.*

#### 5.2.1.1.2 Σύνδεση χρήστη στο symbIoTe

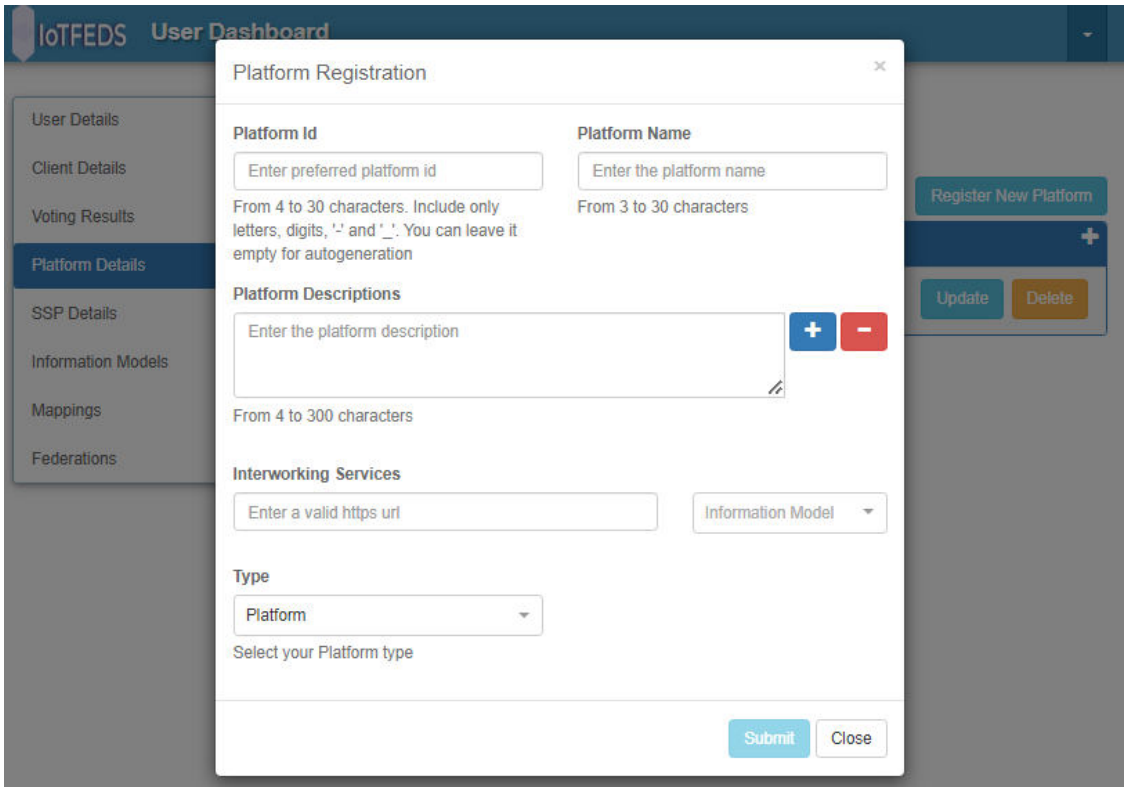
Αφορά τη σύνδεση ενός χρήστη στο IoTFeds σύστημα μέσω του Administration GUI. Ο χρήστης εισάγει τα διαπιστευτήριά του (credentials) στη φόρμα σύνδεσης χρήστη. Το Administration GUI προχωρά στην αυθεντικοποίηση τους και την επαλήθευση στο BaaS.




*Εικόνα 33: Σύνδεση χρήστη στην γραφική διεπαφή χρήστη (Administration GUI).*

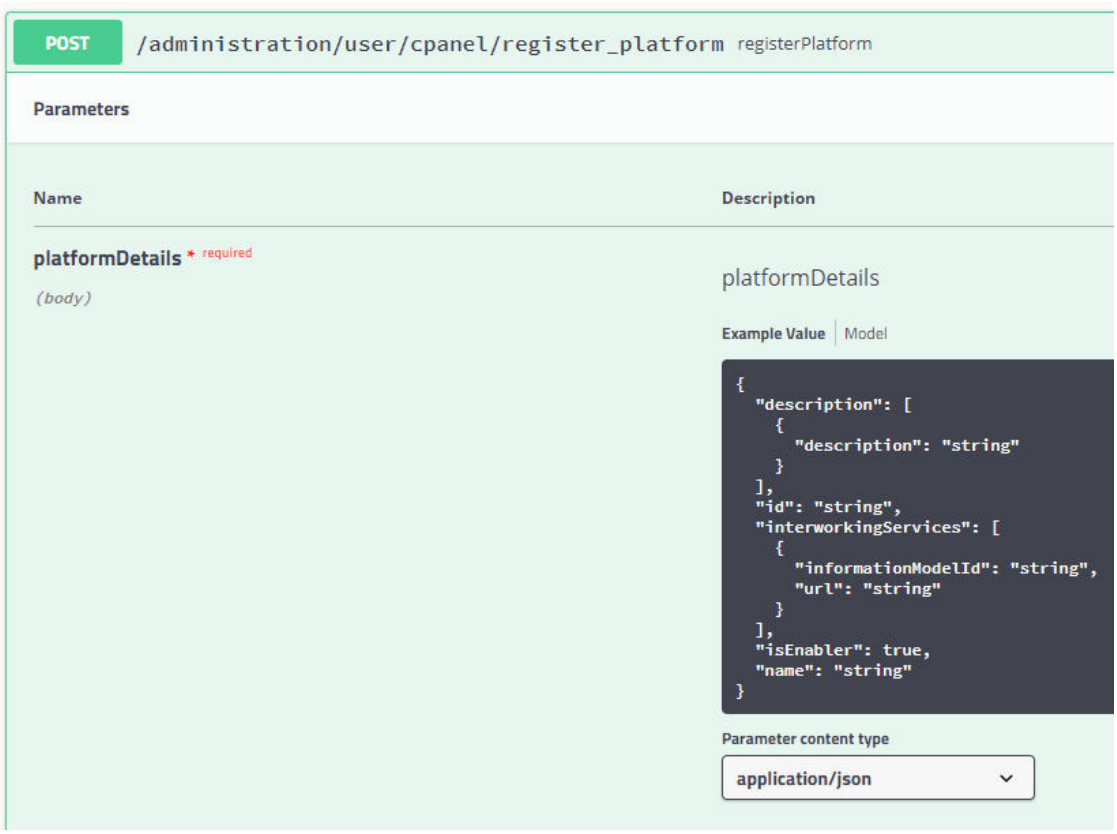
#### 5.2.1.1.3 Εγγραφή πλατφόρμας στο symbIoTe

Η εγγραφή IoT πλατφόρμας γίνεται μέσω της υπάρχουσας φόρμας στο Administration GUI του symbIoTe. Κάθε πάροχος δεδομένων στην IoTFeds πλατφόρμα απαιτείται να εγγράψει την πλατφόρμα του με το interworking service url (όπου θα σηκωθούν τα symbIoTe cloud services) για να συμμετάσχει στις IoTFeds ομοσπονδίες και να διαθέσει τα δεδομένα του. Ένας χρήστης μπορεί να εγγράψει πολλαπλές πλατφόρμες. Ένα τέτοιο σενάριο μπορεί να αφορά έναν οργανισμό (όπως ένα πανεπιστήμιο) που διαμοιράζει του IoT πόρους του μέσω δύο διαφορετικών πλατφορμών όπως UiTOP και Smart Environment για να καλύψει για παράδειγμα διαφορετικούς τομείς όπως έξυπνη στάθμευση και περιβάλλον και επιθυμεί την καταγραφή δύο ξεχωριστών πλατφορμών στο σύστημα.




*Εικόνα 34: Εγγραφή πλατφόρμας στο symbIoTe.*

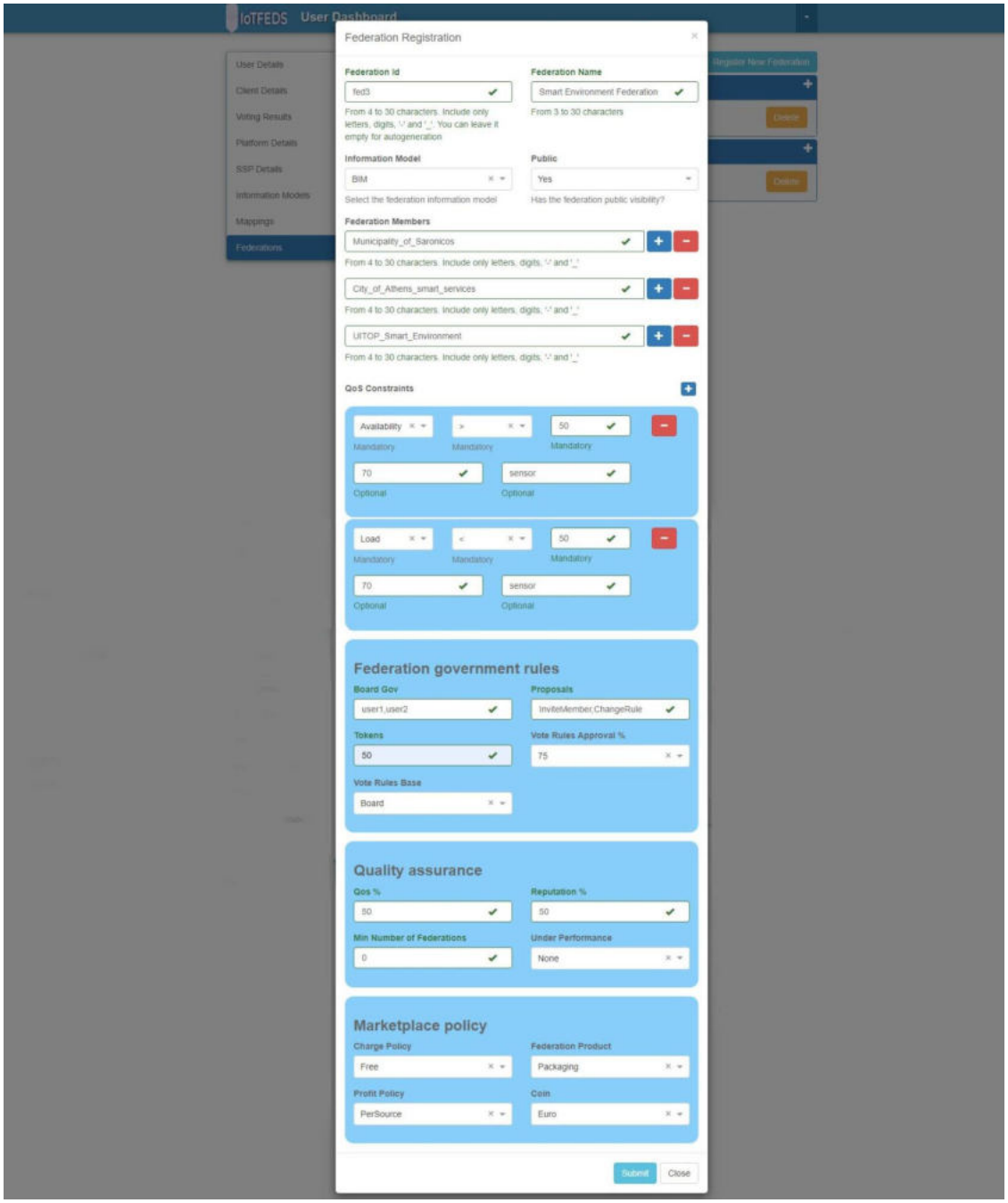
Οι πληροφορίες της πλατφόρμας προς εγγραφή περνάνε στο Administration (backend) στοιχείο για την εγγραφή της πλατφόρμας στο υποσύστημα symbIoTe και την ενημέρωση του BaaS. Όπως φαίνεται στην Εικόνα 35, το αντίστοιχο endpoint του στοιχείου αυτού λαμβάνει το αντικείμενο (object) που αποτελεί την πληροφορία μιας πλατφόρμας προς εγγραφή και επιστρέφει μήνυμα επιτυχίας/αποτυχίας εγγραφής του.




*Εικόνα 35: Πληροφορία εγγραφής μιας πλατφόρμας στο σύστημα του symbIoTe.*

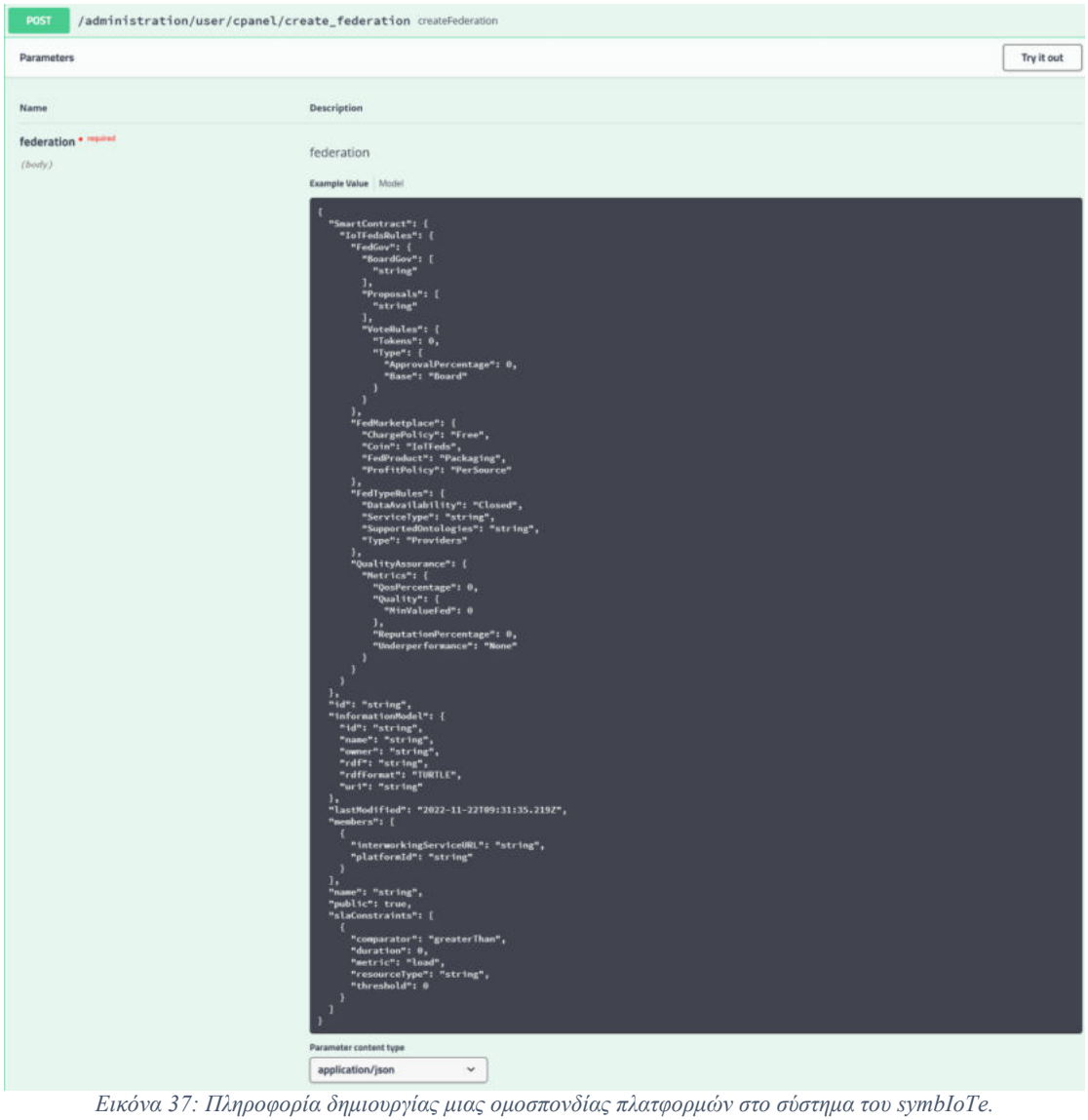
#### 5.2.1.1.4 Δημιουργία ομοσπονδίας στο symbIoTe

Αφορά την δημιουργία μιας IoTFeds ομοσπονδίας πλατφορμών μέσα από το Administration GUI του υποσυστήματος symbIoTe. Στα πλαίσια του IoTFeds έργου, οι πληροφορίες της ομοσπονδίας περιέχουν τους κανόνες ομοσπονδίας όπως ορίστηκαν στην Ενότητα Τεχνικό σχήμα κανόνων και πολιτικών ομοσπονδίας3.3.3:




*Εικόνα 36: Δημιουργία IoT ομοσπονδίας στο σύστημα.*

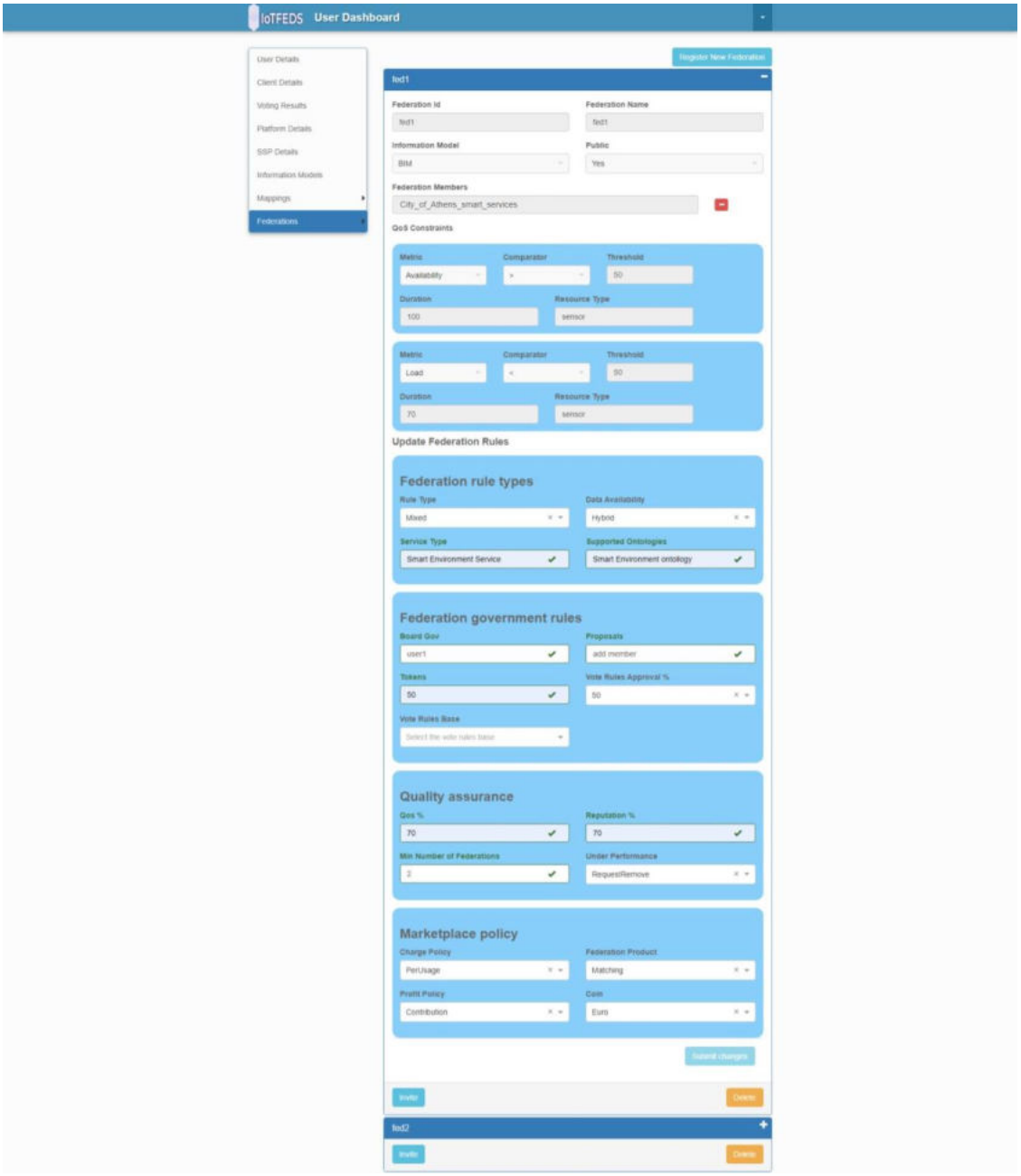
Στη συνέχεια καλείται το αντίστοιχο endpoint στο Administration (backend) που δέχεται ως είσοδο το αντικείμενο (object) που αποτελεί την πληροφορία μιας ομοσπονδίας προς δημιουργία όπως φαίνεται στην Εικόνα 37 και επιστρέφει μήνυμα επιτυχίας/αποτυχίας.




*Εικόνα 37: Πληροφορία δημιουργίας μιας ομοσπονδίας πλατφορμών στο σύστημα του symbIoTe.*

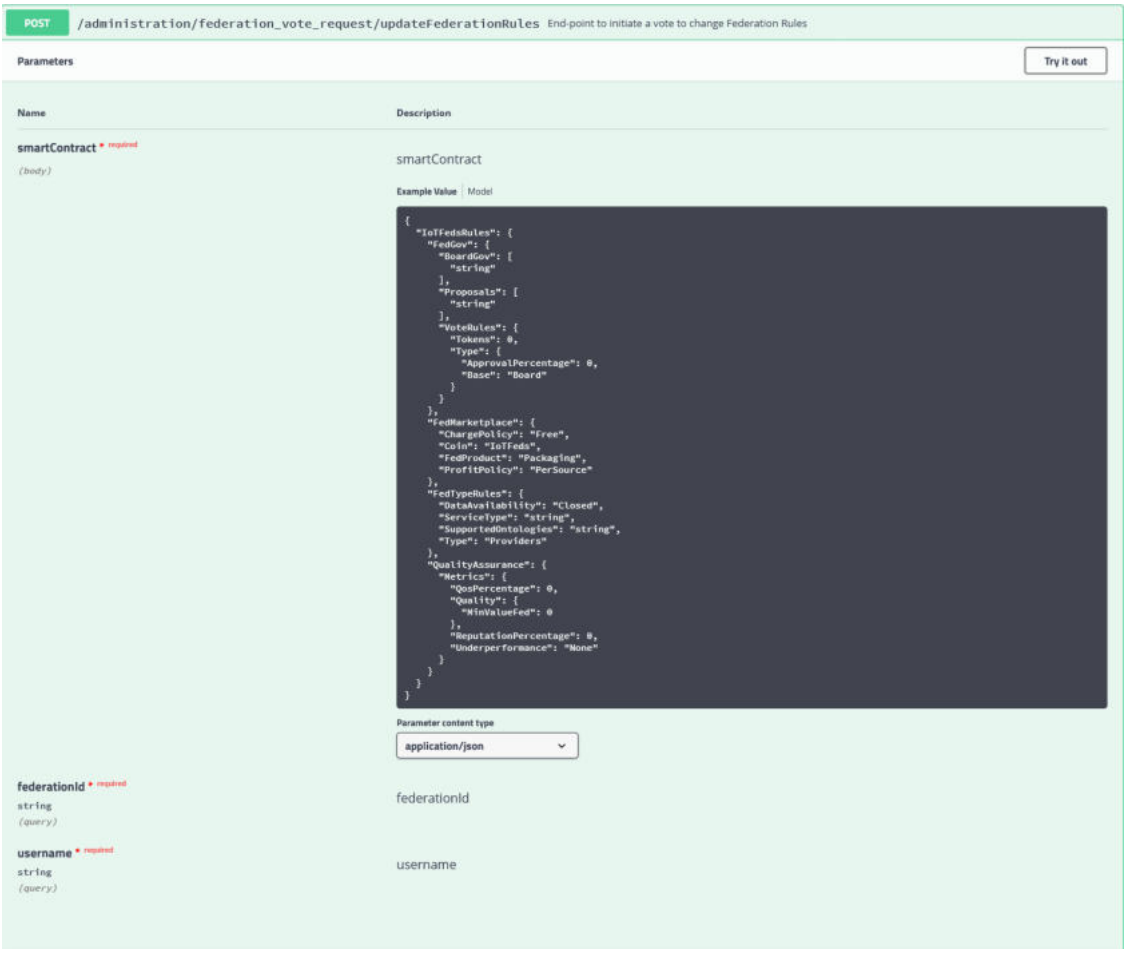
#### 5.2.1.1.5 Καταχώρηση αιτήματος ενημέρωσης κανόνων ομοσπονδίας

Αφορά την καταχώρηση αιτήματος για την ενημέρωση των κανόνων υπάρχουσας IoTFeds ομοσπονδίας όπως υλοποιήθηκε στα πλαίσια του έργου. Το αίτημα γίνεται από τη σελίδα της λίστας ομοσπονδιών που συμμετέχει ο χρήστης.




*Εικόνα 38: Τροποποίηση κανόνων ομοσπονδίας.*

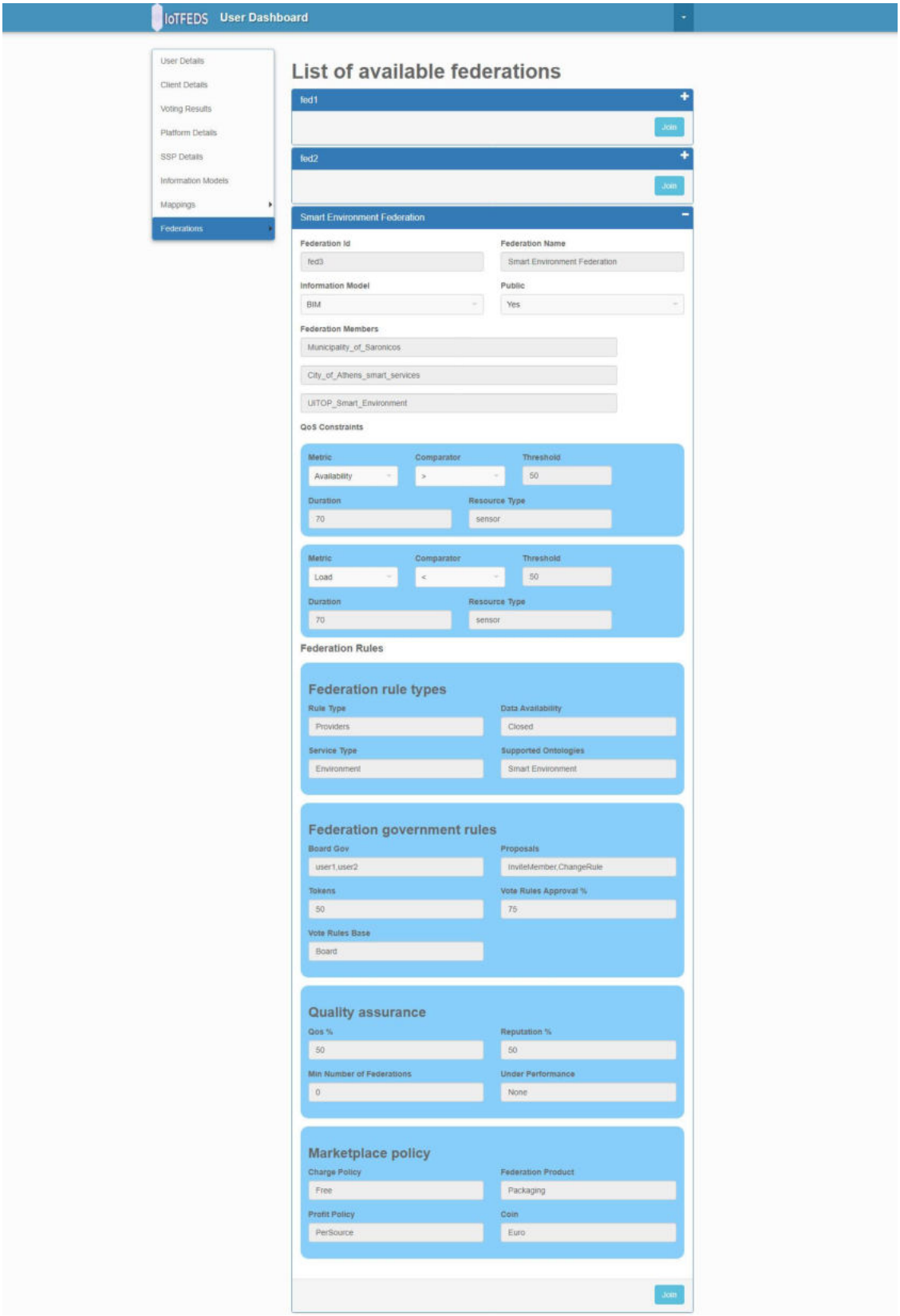
Επιλέγοντας την τροποποίηση κανόνων, το αίτημα περνάει στο αντίστοιχο endpoint που δέχεται ως είσοδο το αντικείμενο (object) που αποτελεί τους κανόνες της ομοσπονδίας όπως φαίνεται στην Εικόνα 39. Η διαδικασία αυτή σηματοδοτεί την έναρξη μίας διαδικασίας ψηφοφορίας στο BaaS για την ανανέωση των κανόνων της ομοσπονδίας.




*Εικόνα 39: Καταχώρηση αιτήματος ενημέρωσης κανόνων ομοσπονδίας.*

#### 5.2.1.1.6 Ανάκτηση λίστας ομοσπονδιών

Αφορά την ανάκτηση μιας λίστας όλων των ομοσπονδιών που είναι εγγεγραμμένες στο IoTFeds σύστημα. Επιστρέφει μια λίστα με όλες τις ομοσπονδίες και τις ενημερωμένες πληροφορίες τους όπως αυτές φαίνονται στην Εικόνα 40.




*Εικόνα 40: Λίστα ομοσπονδιών.*

#### 5.2.1.1.7 Καταχώρηση αιτήματος συμμετοχής σε ομοσπονδία

Αφορά την καταχώρηση αιτήματος για συμμετοχή σε υπάρχουσα ομοσπονδία στο σύστημα η οποία εκκινεί διαδικασία ψηφοφορίας σύμφωνα με τους κανόνες της ομοσπονδίας που έχουν οριστεί. Το αίτημα από τον χρήστη γίνεται από την σελίδα με τη λίστα ομοσπονδιών στην

προηγούμενη παράγραφο όπου ο χρήστης μπορεί να δει τις διαθέσιμες ομοσπονδίες και πληροφορίες για αυτές. Ο χρήστης έχει τη δυνατότητα να ζητήσει οι πλατφόρμες του να εισέλθουν σε κάποια ομοσπονδία επιλέγοντας το κουμπί "Join" για την επιθυμητή ομοσπονδία. Αυτή η ενέργεια σηματοδοτεί την εκκίνηση της ροής ψηφοφορίας για τη συμμετοχή στην ομοσπονδία. Η διαδικασία της ψηφοφορίας παρέχεται από το BaaS.
Πιο συγκεκριμένα, το αίτημα του χρήστη περνάει στο αντίστοιχο endpoint του Administration (backend) που παίρνει σαν παράμετρο το id της ομοσπονδίας και επιστρέφει μήνυμα επιτυχούς/ανεπιτυχούς καταχώρησης του αιτήματος όπως φαίνεται στην Εικόνα 41. Στη συνέχεια ενημερώνεται το BaaS για να ξεκινήσει η διαδικασία ψηφοφορίας για την συμμετοχή ή όχι του μέλους στην ομοσπονδία.

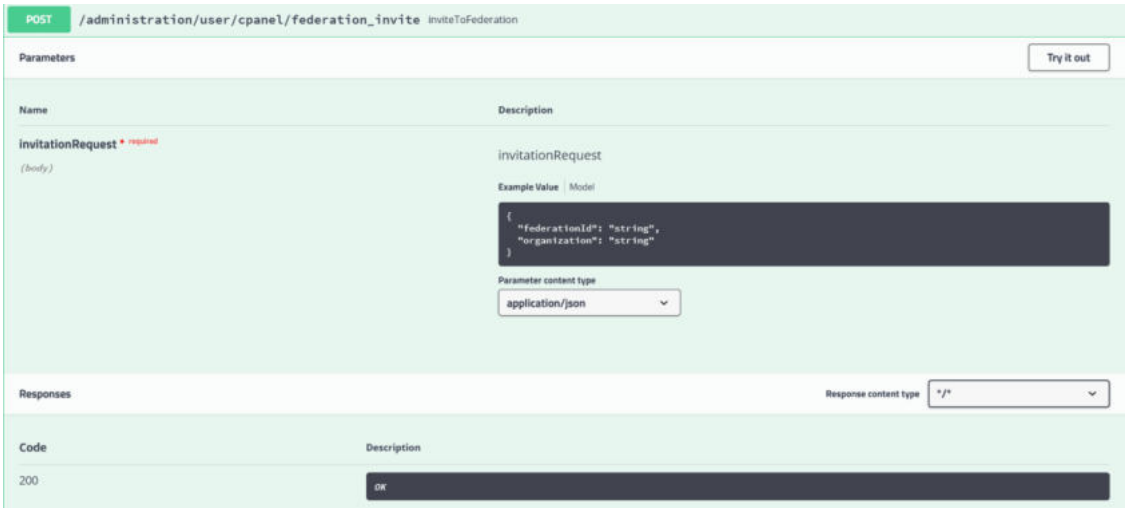

*Εικόνα 41: Καταχώρηση αιτήματος συμμετοχής σε ομοσπονδία του symbIoTe.*

#### 5.2.1.1.8 Πρόσκληση μέλους σε ομοσπονδία

Αφορά την πρόσκληση ενός μέλους σε υπάρχουσα ομοσπονδία του symbIoTe, μέσα από το administration GUI όπως φαίνεται στην Εικόνα 42. Για να προσκαλέσει ένα άλλο μέλος σε μια ομοσπονδία, ένας χρήστης πρέπει να είναι ο ίδιος ενεργό μέλος της ομοσπονδίας.


*Εικόνα 42: Πρόσκληση μέλους σε ομοσπονδία.*

Το αίτημα του χρήστη περνάει στο αντίστοιχο endpoint του Administration (backend) που παίρνει σαν παράμετρο το αναγνωριστικό της ομοσπονδίας και του χρήστη όπως φαίνεται στην παρακάτω εικόνα.

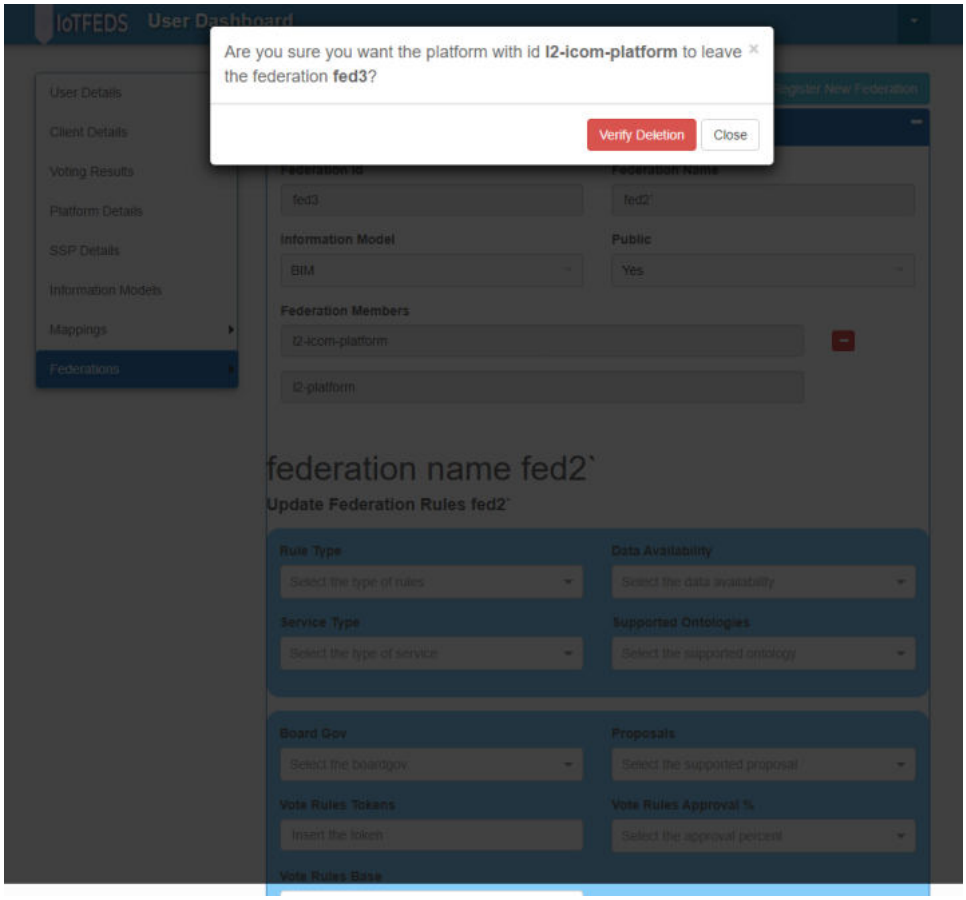



*Εικόνα 43: Καταχώρηση αιτήματος πρόσκλησης μέλους στο symbIoTe.*

Το αίτημα πρόσκλησης μέλους πυροδοτεί διαδικασία ψηφοφορίας με βάση τους κανόνες που ορίστηκαν κατά τη δημιουργία της ομοσπονδίας.

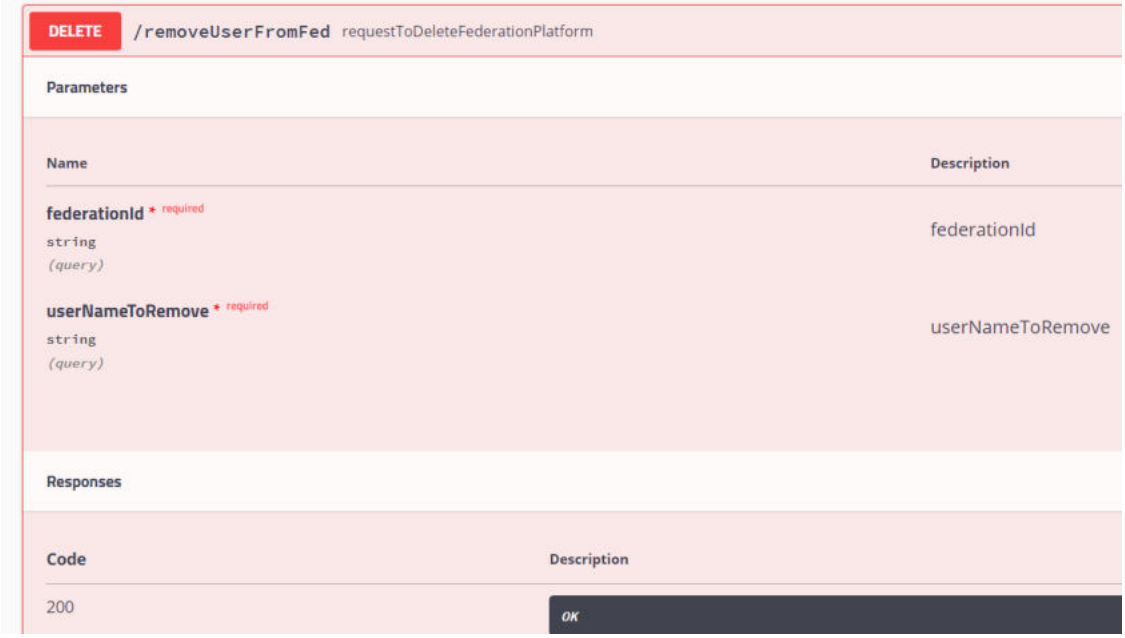
#### 5.2.1.1.9 Καταχώρηση αιτήματος απομάκρυνσης μέλους ομοσπονδίας

Αφορά την καταχώρηση αιτήματος για την απομάκρυνση ενός μέλους από υπάρχουσα ομοσπονδία πυροδοτώντας διαδικασία ψηφοφορίας ακολουθώντας τους κανόνες που έχουν καθοριστεί στην ομοσπονδία. Το αίτημα γίνεται από τη σελίδα της λίστας ομοσπονδιών που συμμετέχει ο χρήστης.




*Εικόνα 44: Αίτημα αποχώρησης από ομοσπονδία.*

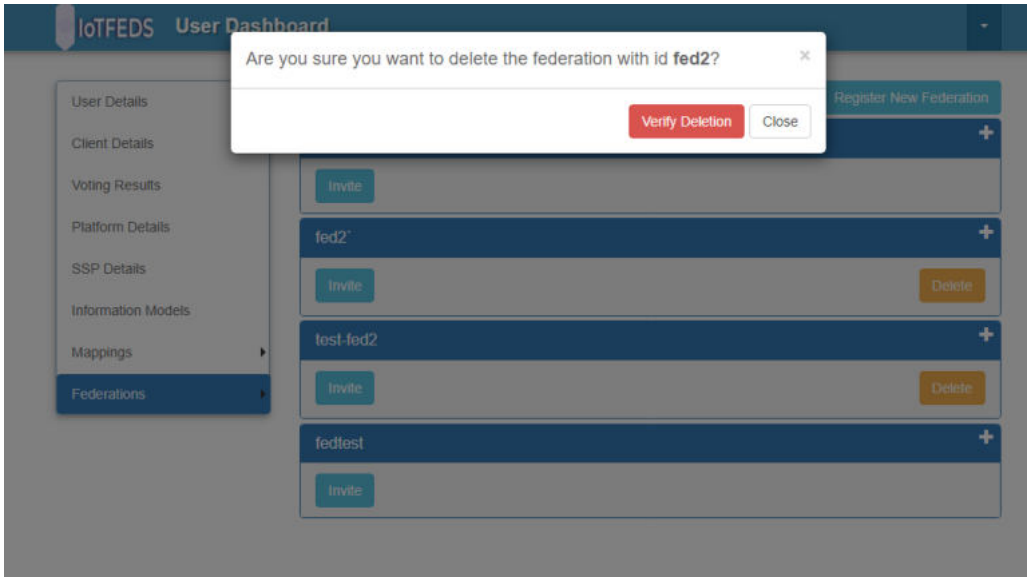
Το αίτημα προωθείται στο endpoint του Administration (backend) που παίρνει σαν παράμετρο το id της ομοσπονδίας και το username του προς απομάκρυνση χρήστη και επιστρέφει μήνυμα επιτυχούς/ανεπιτυχούς καταχώρησης του αιτήματος όπως φαίνεται στην Εικόνα 45. Η διαδικασία αυτή σηματοδοτεί την έναρξη μίας ψηφοφορίας για την απομάκρυνση ή όχι του μέλους από την ομοσπονδία στο BaaS.




*Εικόνα 45: Καταχώρηση αιτήματος απομάκρυνσης μέλους από ομοσπονδία του symbIoTe.*

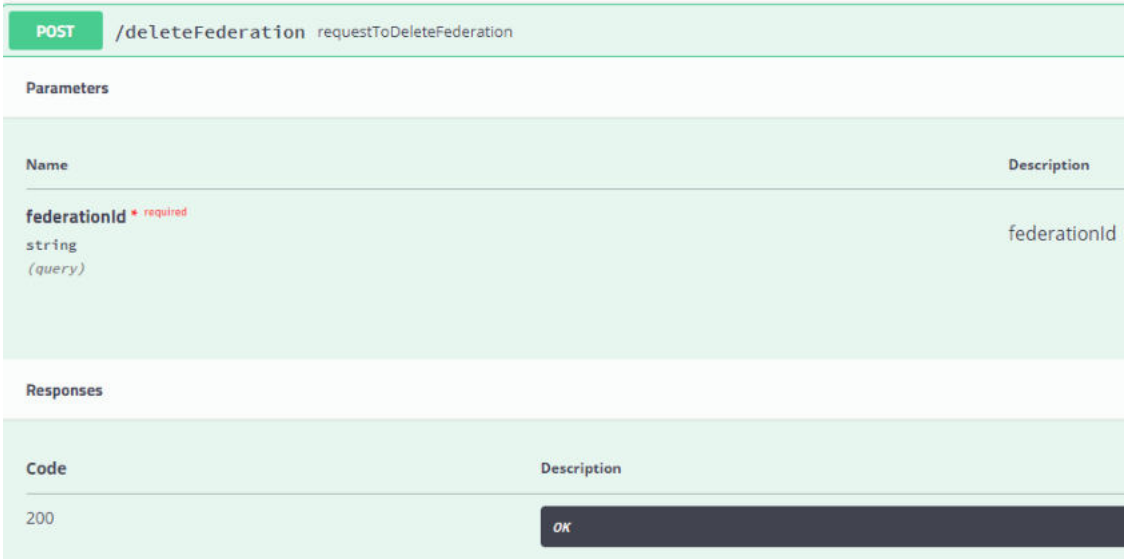
#### 5.2.1.1.10 Καταχώρηση αιτήματος διαγραφής ομοσπονδίας

Αφορά την καταχώριση αιτήματος για την διαγραφή μιας υπάρχουσας ομοσπονδίας. Ξεκινάει από την σελίδα της λίστας ομοσπονδιών που συμμετέχει ο χρήστης. Ένας χρήστης μπορεί να διαγράψει την ομοσπονδία, μόνο στην περίπτωση που αποτελεί το μόνο εναπομείναν μέλος της ομοσπονδίας, μετά από αποχώρηση των υπολοίπων μελών.




*Εικόνα 46: Διαγραφή ομοσπονδίας του symbIoTe.*

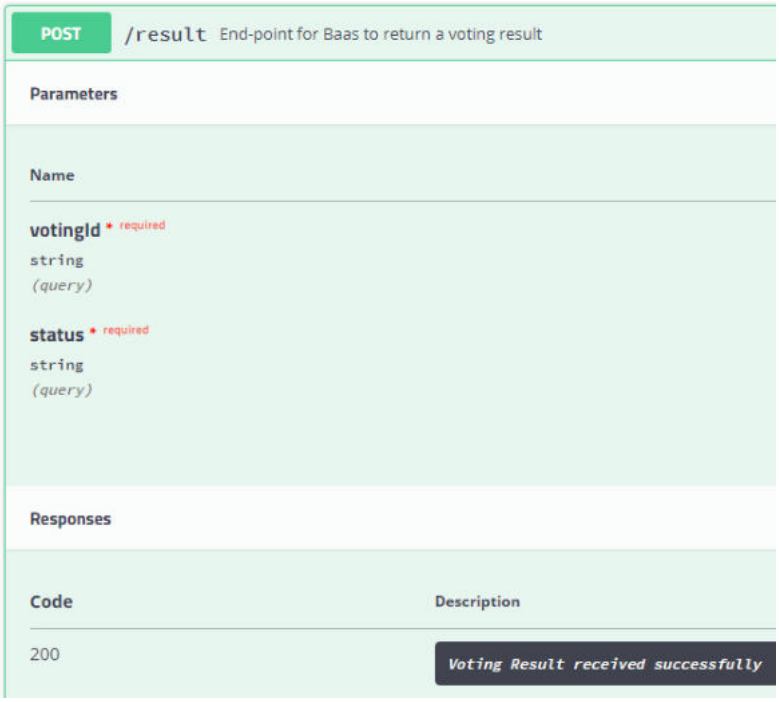
Το αίτημα του χρήστη περνάει στο endpoint του Administration (backend) που δέχεται ως παράμετρο το id της ομοσπονδίας προς διαγραφή και επιστρέφει μήνυμα επιτυχούς/ανεπιτυχούς καταχώρησης του αιτήματος όπως φαίνεται στην Εικόνα 47. Η διαδικασία περιλαμβάνει την ενημέρωση του BaaS.




*Εικόνα 47: Καταχώρηση αιτήματος διαγραφής ομοσπονδίας στο symbIoTe.*

#### 5.2.1.1.11 Διεκπεραίωση αιτήματος ψηφοφορίας

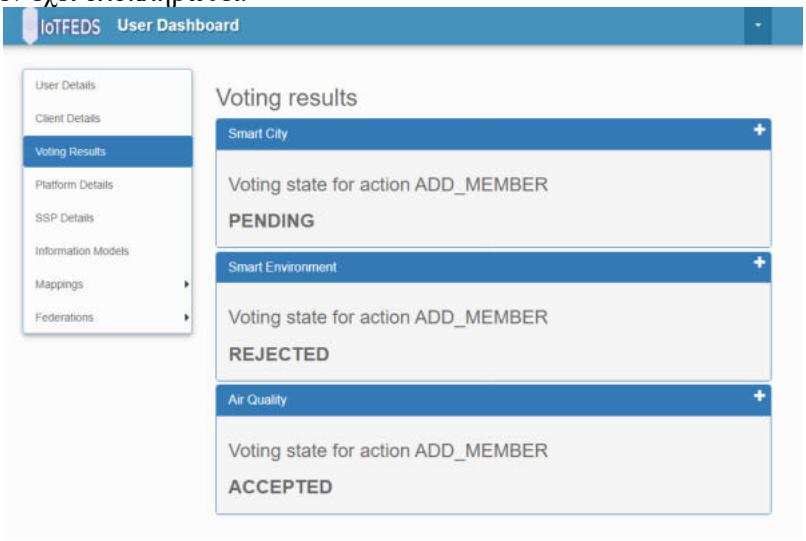
Αφορά την διεκπεραίωση ενός αιτήματος ψηφοφορίας για οποιαδήποτε από τις επιτρεπόμενες ενέργειες (συμμετοχή ή προσθήκη μέλους σε ομοσπονδία, αφαίρεση μέλους από ομοσπονδία, διαγραφή ομοσπονδίας, ενημέρωση κανόνων ομοσπονδίας) μετά την περαίωση της διαδικασίας ψηφοφορίας. Παίρνει σαν είσοδο το αναγνωριστικό (id) του αιτήματος ψηφοφορίας και το object που περιέχει το αποτέλεσμα της ψηφοφορίας όπως φαίνεται στην Εικόνα 48. Η διαδικασία αυτή σηματοδοτεί το τέλος μίας διαδικασίας ψηφοφορίας και την ενημέρωση από το BaaS στο symbIoTe.




*Εικόνα 48: Διεκπεραίωση αιτήματος ψηφοφορίας.*

#### 5.2.1.1.12 Λίστα αιτημάτων χρήστη για ψηφοφορία

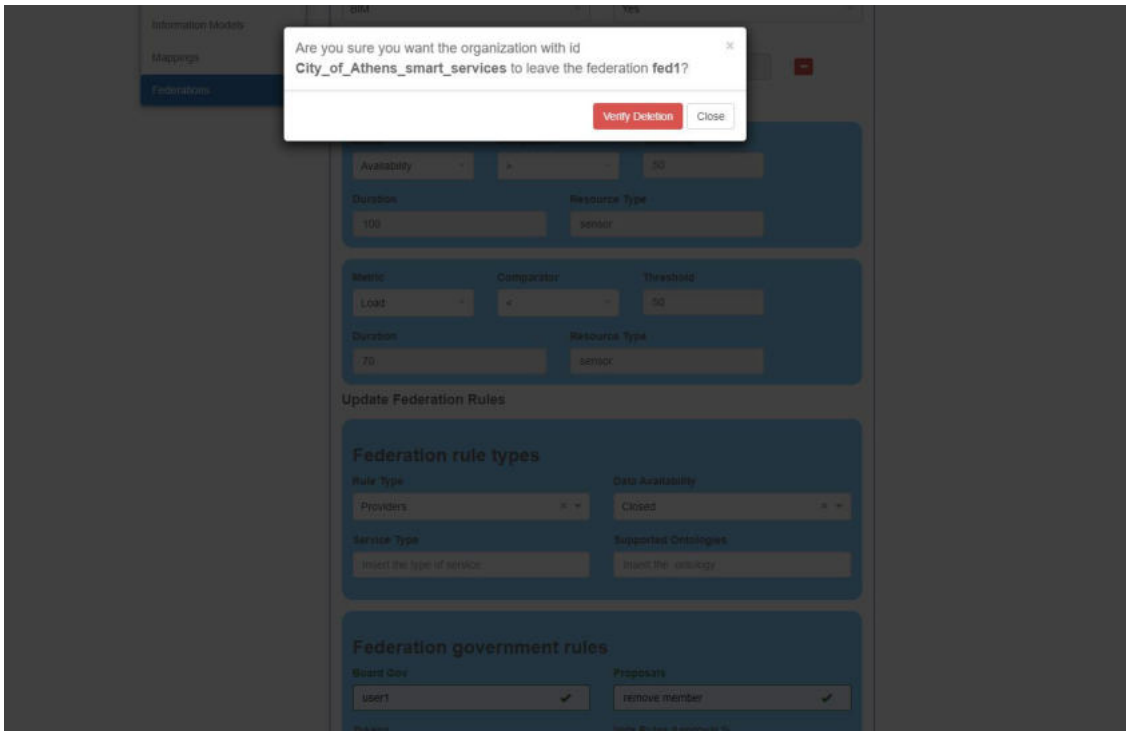
Από αυτή τη σελίδα ένας χρήστης μπορεί να δει μία λίστα των αιτημάτων για διαδικασία ψηφοφορίας που έχει πραγματοποιήσει στο παρελθόν, καθώς και την κατάσταση που βρίσκονται(PENDING, ACCEPTED ή REJECTED) όπως φαίνεται στην Εικόνα 49. Για τα αιτήματα που βρίσκονται σε κατάσταση «PENDING», μία διαδικασία ψηφοφορίας είναι σε εξέλιξη και δεν έχει ολοκληρωθεί.




*Εικόνα 49: Λίστα αιτημάτων για διαχείριση ομοσπονδίας και η κατάστασή τους.*

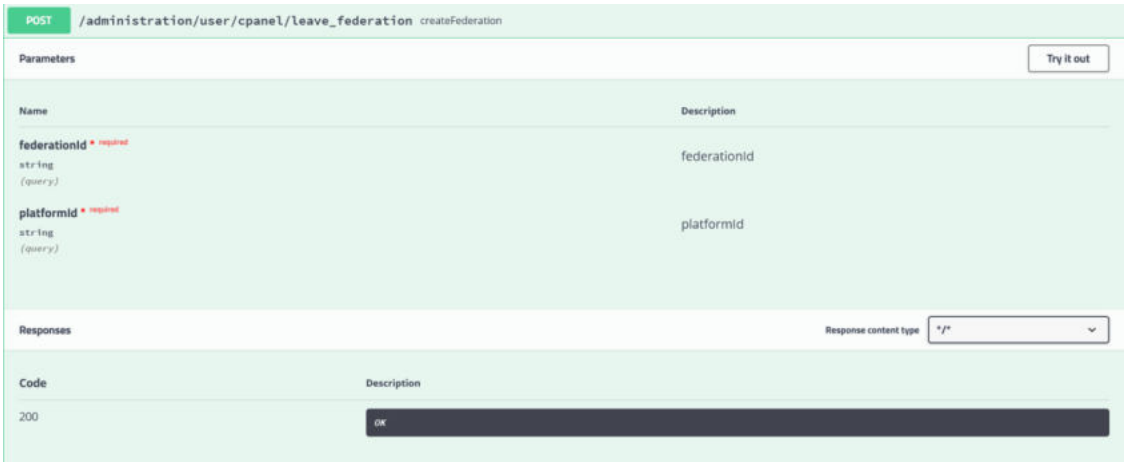
#### 5.2.1.1.13 Αποχώρηση από ομοσπονδία

Αφορά την αποχώρηση ενός μέλους από υπάρχουσα ομοσπονδία στην οποία συμμετέχει ήδη, μέσα από το administration GUI όπως φαίνεται στην Εικόνα 50. Μόνο ο service owner έχει τη δυνατότητα να ζητήσει την αποχώρηση από μια ομοσπονδία.




*Εικόνα 50: Αποχώρηση από ομοσπονδία στο symbIoTe.*

Το αίτημα προωθείται στο παρακάτω endpoint:



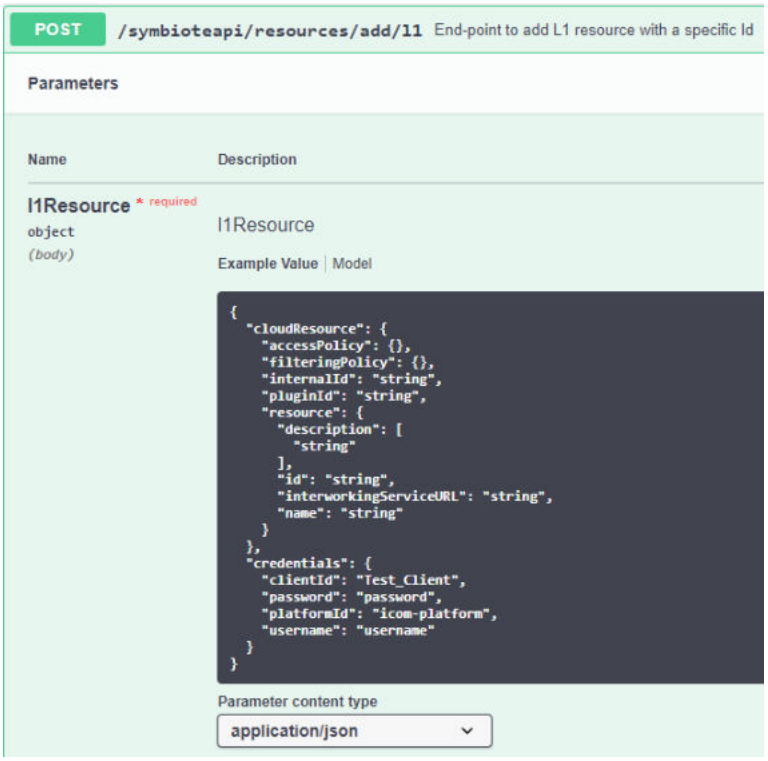

*Εικόνα 51: Καταχώρηση αιτήματος αποχώρησης από ομοσπονδία στο symbIoTe.*

### 5.2.1.2 Στοιχείο λογισμικού για τη διαχείριση IoT πόρων

Η διαχείριση των πόρων στο οικοσύστημα του symbIoTe γίνεται μέσω Restful APIs. Τα επιμέρους endpoints αναφέρονται παρακάτω.

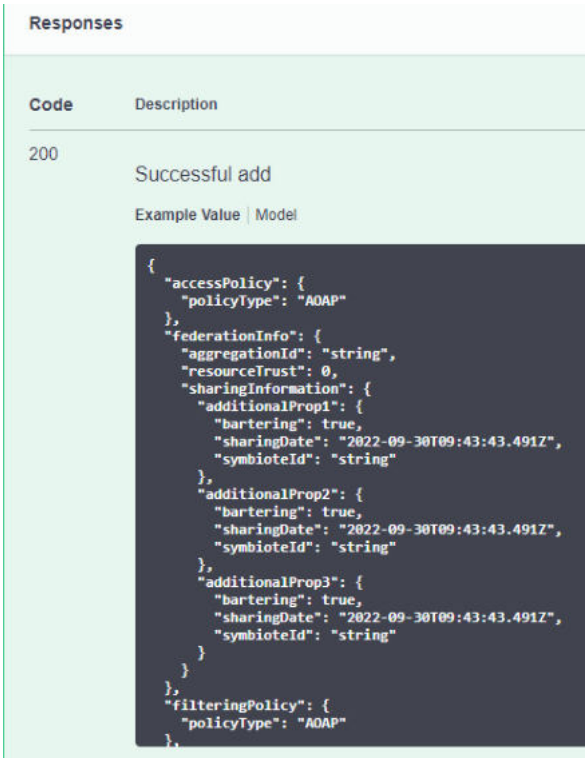
#### 5.2.1.2.1 Εγγραφή πόρων στο symbIoTe

Αφορά την εγγραφή πηγών δεδομένων στο σύστημα IoTFeds. Δέχεται ως είσοδο το αντικείμενο (object) που αποτελεί την πληροφορία ενός πόρου προς εγγραφή όπως φαίνεται στην Εικόνα 52 και επιστρέφει μήνυμα επιτυχίας/αποτυχίας εγγραφής του.



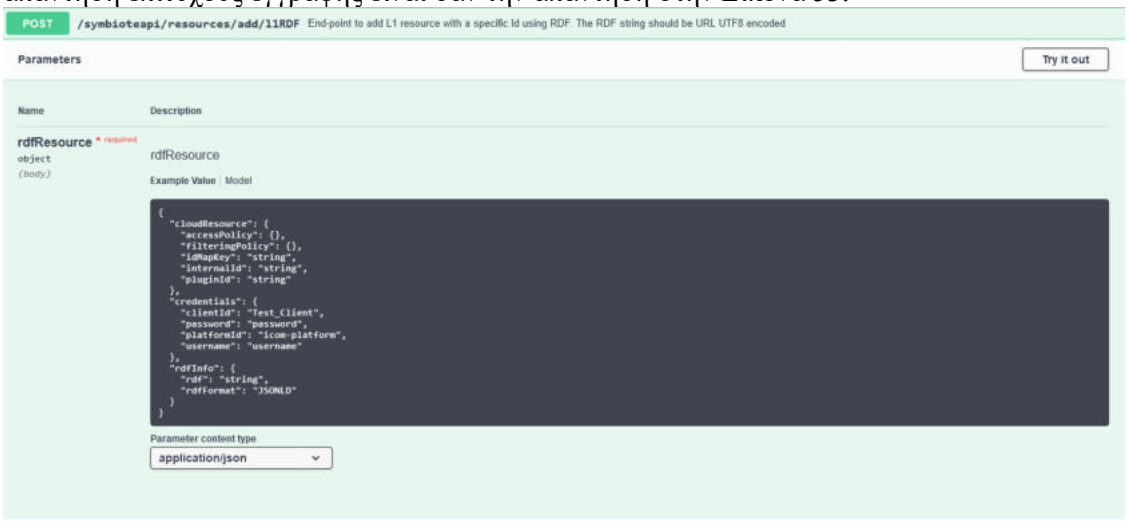

*Εικόνα 52: Πληροφορία εγγραφής πόρων στο σύστημα του SymbIoTe.*

Σε περίπτωση επιτυχίας, επιστρέφει και τη συνολική πληροφορία για τον πόρο που έχει δημιουργηθεί όπως φαίνεται στην Εικόνα 53.




*Εικόνα 53: Απάντηση επιτυχούς εγγραφής πόρου στο symbIoTe.*

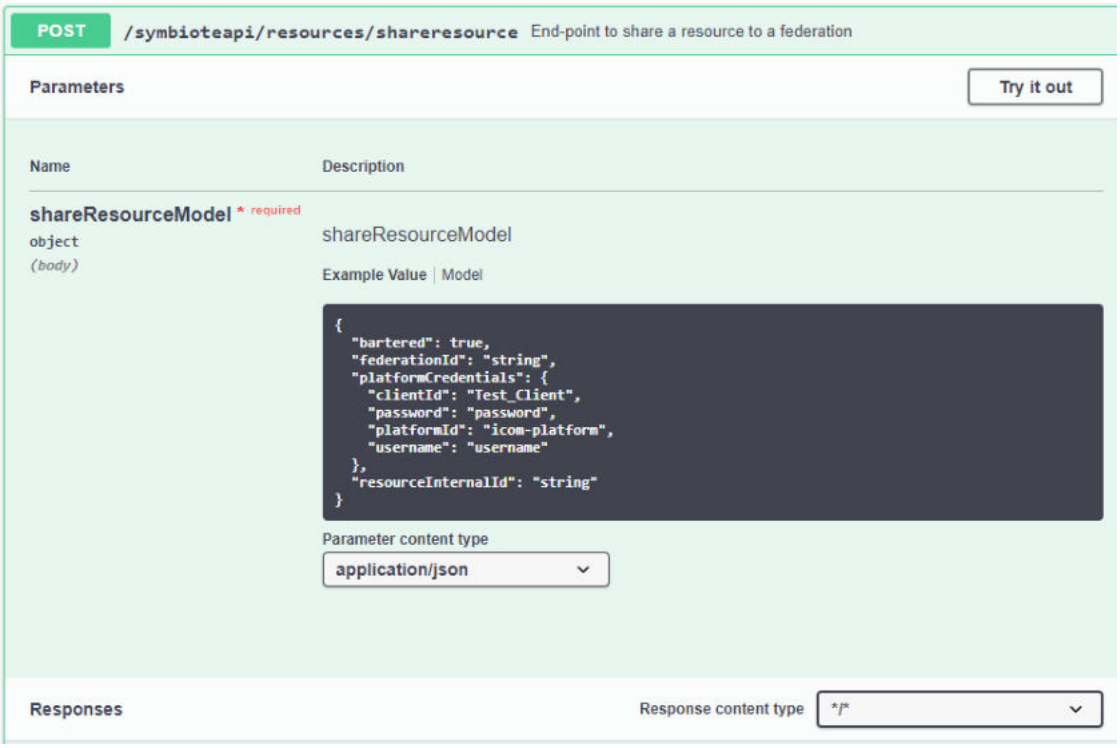
Παρακάτω φαίνεται το endpoint για καταχώρηση IoT πόρου χρησιμοποιώντας RDF. Η απάντηση επιτυχούς εγγραφής είναι σαν την απάντηση στην Εικόνα 53.




*Εικόνα 54: Καταχώρηση IoT πόρου στο symbIoTe με χρήση RDF.*

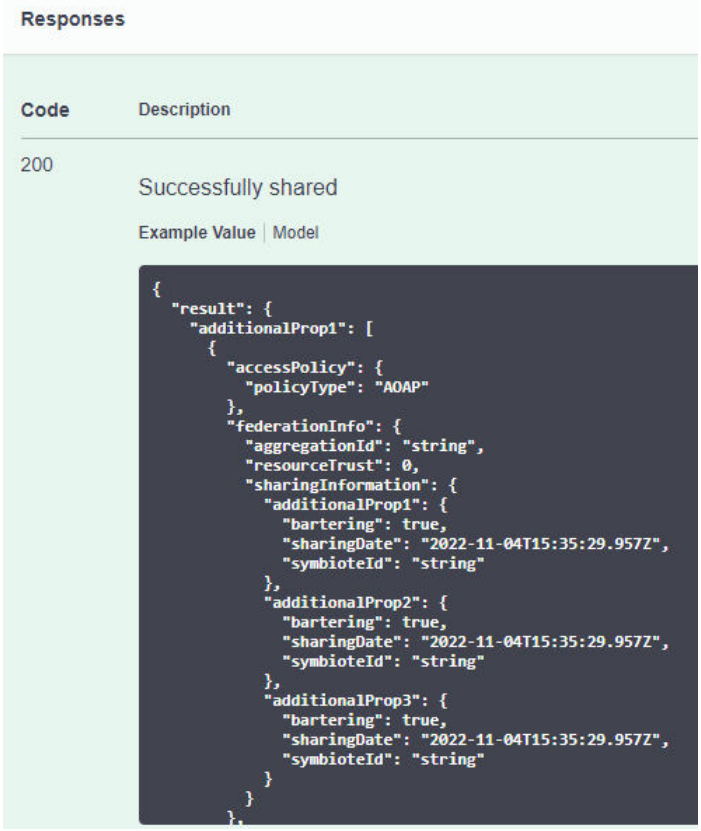
#### 5.2.1.2.2 Κοινοποίηση πόρων σε ομοσπονδία

Αφορά την κοινοποίηση πόρων σε υπάρχουσα ομοσπονδία στο σύστημα. Δέχεται σαν είσοδο ένα αντικείμενο που περιέχει την πληροφορία σχετικά με το ποιο αντικείμενο (πόρος) πρόκειται να κοινοποιηθεί σε ποια ομοσπονδία, καθώς και τα διαπιστευτήρια ενός χρήστη της πλατφόρμας στην οποία ανήκει ο πόρος όπως φαίνεται στην Εικόνα 55.




*Εικόνα 55: Πληροφορία κοινοποίησης πόρων σε ομοσπονδία του symbIoTe.*

Σε περίπτωση επιτυχίας, επιστρέφει το αποτέλεσμα όπως φαίνεται στην Εικόνα 56.

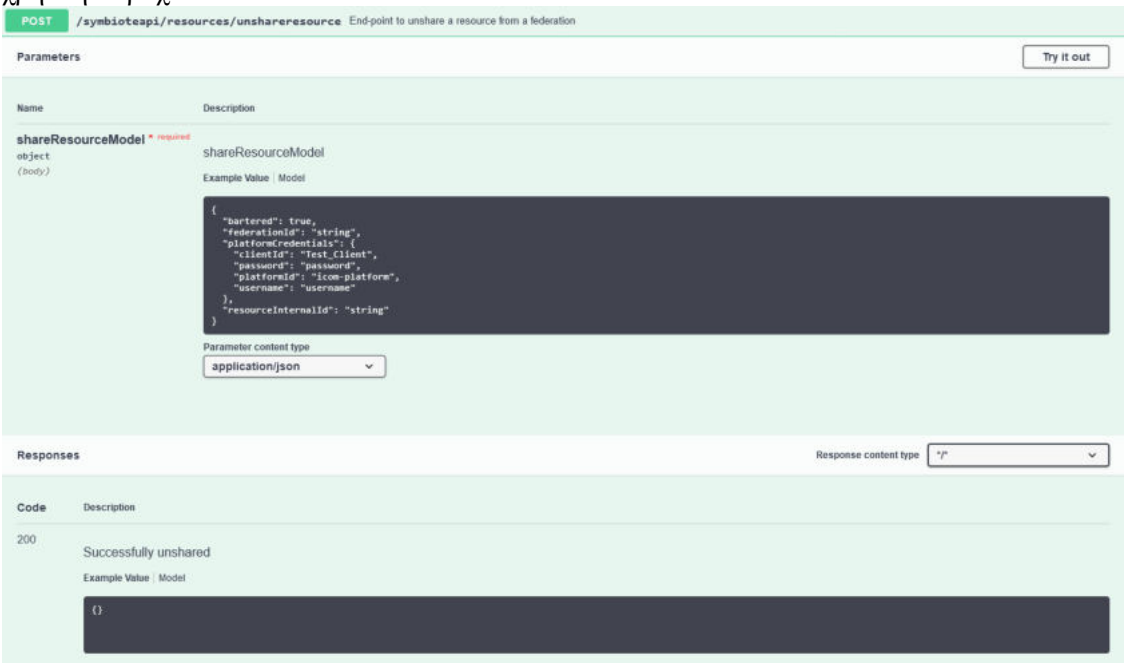



*Εικόνα 56: Πληροφορία επιτυχούς κοινοποίησης πόρου σε ομοσπονδία.*

#### 5.2.1.2.3 Διαγραφή πόρων στο symbIoTe

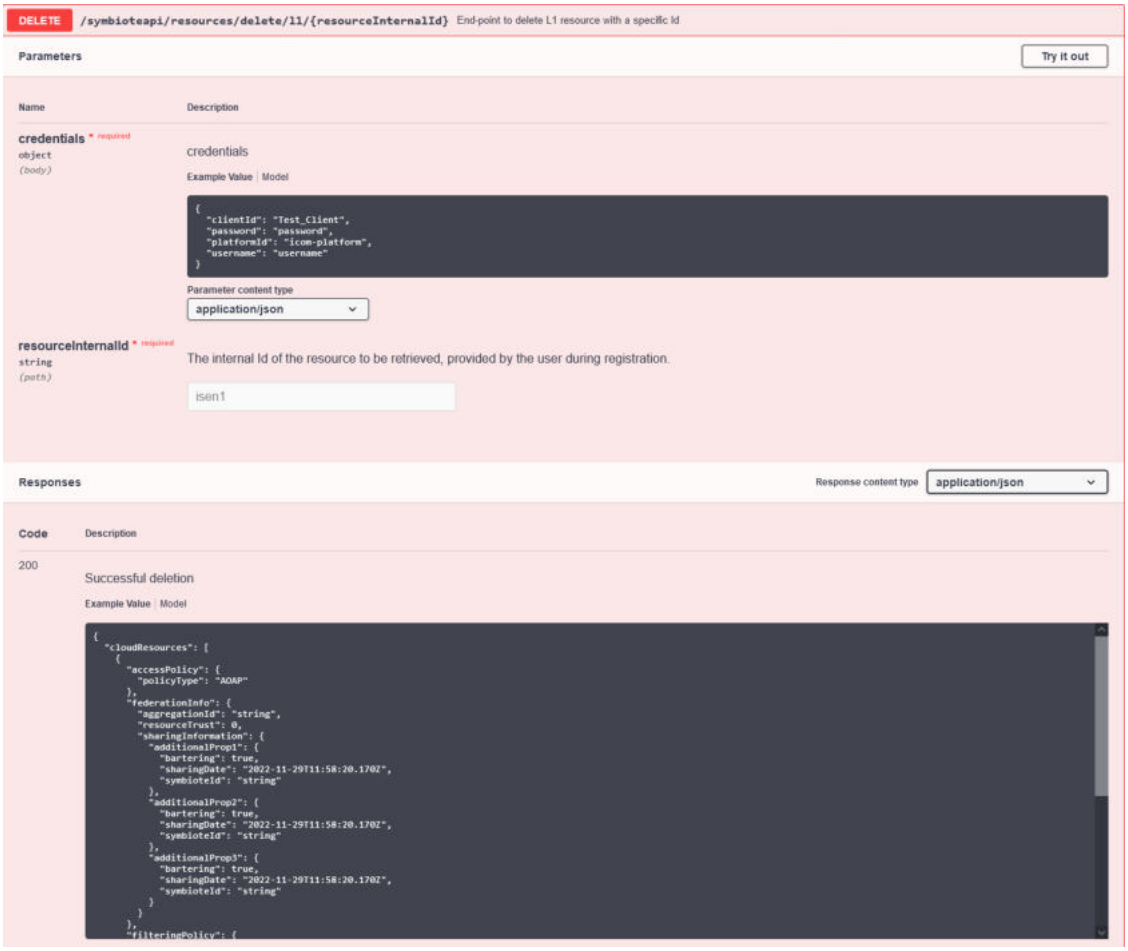
Τέλος, παρέχεται η δυνατότητα αφαίρεσης ενός IoT πόρου από μία IoTFeds ομοσπονδία ή και από το IoTFeds σύστημα.

Όπως φαίνεται στην παρακάτω εικόνα, μία πηγή δεδομένων αφαιρείται από μία ομοσπονδία καθορίζοντας στο αντικείμενο που δέχεται ως είσοδο το αναγνωριστικό του πόρου καθώς και της ομοσπονδίας από την οποία θέλει ο πάροχος να αφαιρεθεί ο πόρος. Τα διαπιστευτήρια του χρήστη/παρόχου απαιτούνται.

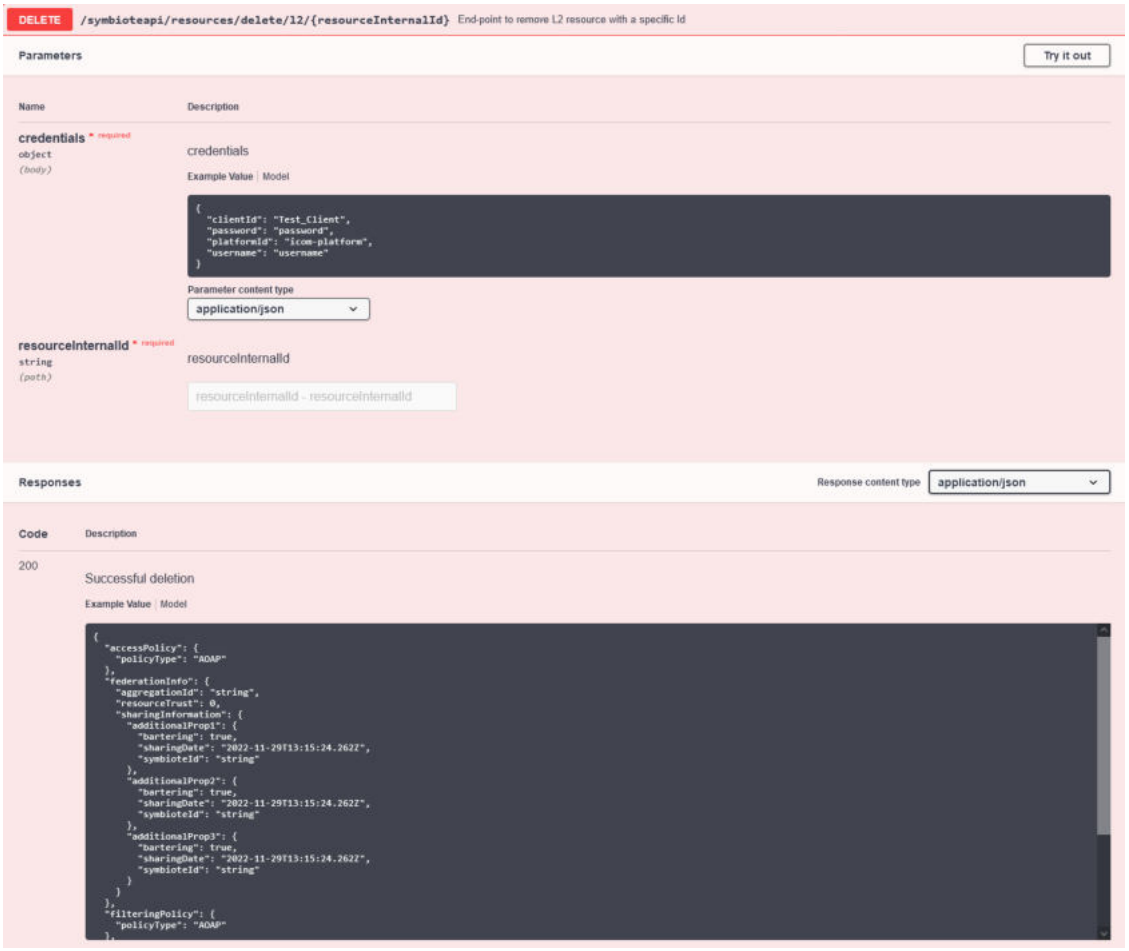



*Εικόνα 57: Πληροφορία αφαίρεσης πόρου από ομοσπονδία.*

Παρακάτω δίνεται η δυνατότητα αφαίρεσης ενός IoT πόρου από το σύστημα. Στην πρώτη εικόνα ο πόρος διαγράφεται από τις κεντρικοποιημένες υπηρεσίες του symbIoTe (πόροι που έχουν αφαιρεθεί από ομοσπονδίες ή έχουν διατεθεί εκτός ομοσπονδίας) ενώ η δεύτερη εικόνα αφορά ομόσπονδους πόρους. Οι παράμετροι που απαιτούνται να καθοριστούν είναι το αναγνωριστικό του IoT πόρου και τα διαπιστευτήρια του χρήστη (παρόχου).




*Εικόνα 58:Διαγραφή πόρου από το σύστημα.*




*Εικόνα 59: Διαγραφή ομόσπονδου πόρου από το σύστημα.*

### 5.2.2 Υποσύστημα BaaS

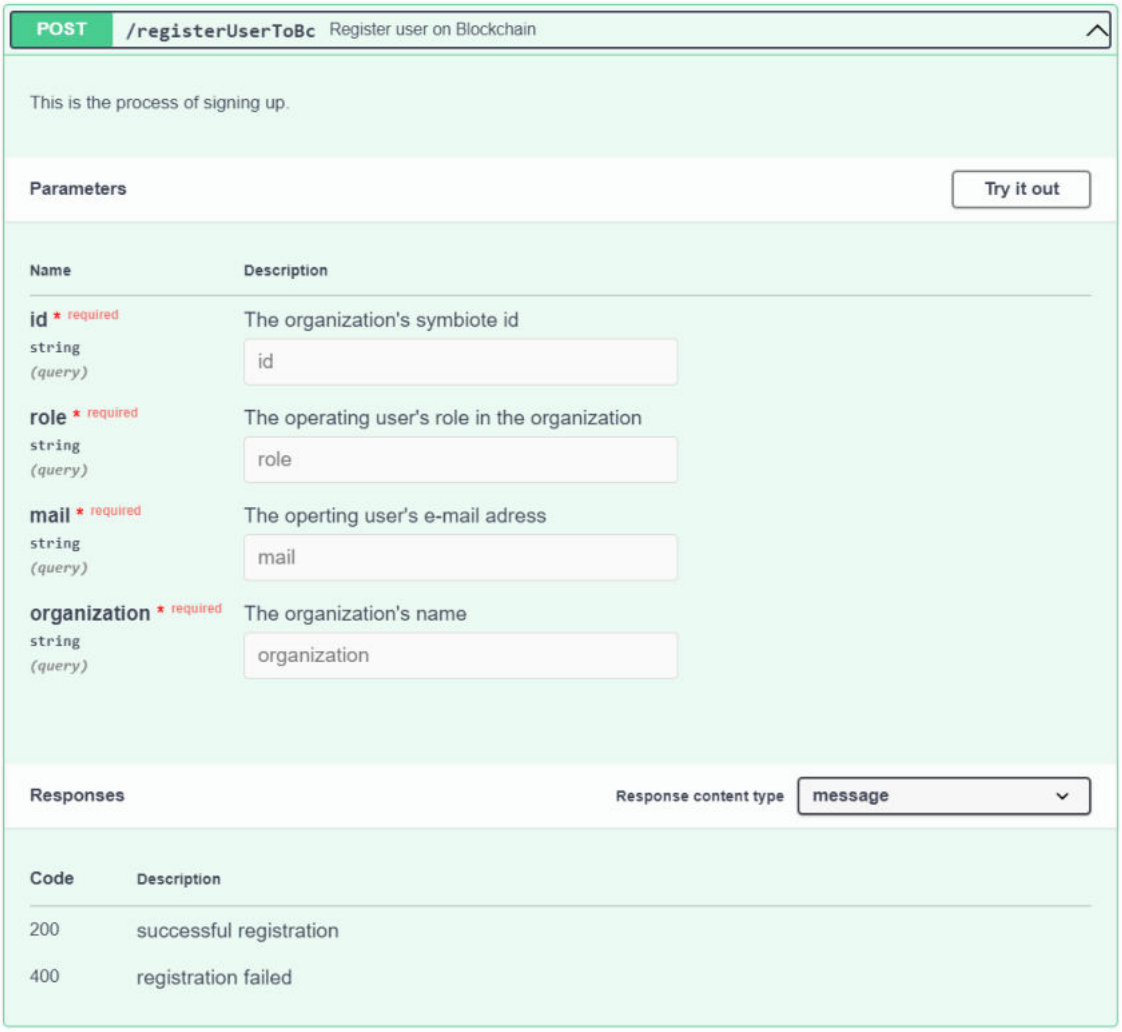
Για την αρχική υλοποίηση και τις βασικές λειτουργίες των ομοσπονδιών μέσω της πλατφόρμας IoTFeds, το υποσύστημα BaaS «εκθέτει» συγκεκριμένα σημεία πρόσβασης (endpoints) μέσω REST-API. Τα endpoints αυτά διαχωρίζονται σε λειτουργίες που αφορούν την διαχείριση χρηστών στο BaaS και σε άλλες που αφορούν την διαχείριση ομοσπονδιών και τις ψηφοφορίες. Οι ενότητες που ακολουθούν έχουν δημιουργηθεί με βάση αυτή την κατηγοριοποίηση.

#### *5.2.2.1 Στοιχείο λογισμικού BaaS για διαχείριση χρηστών*

Το στοιχείο αυτό αφορά όλα τα endpoints που έχουν να κάνουν με την διαχείριση χρηστών στο BaaS σύστημα του IoTFeds. Ακολουθεί μία σύντομη περιγραφή για το καθένα από αυτά.

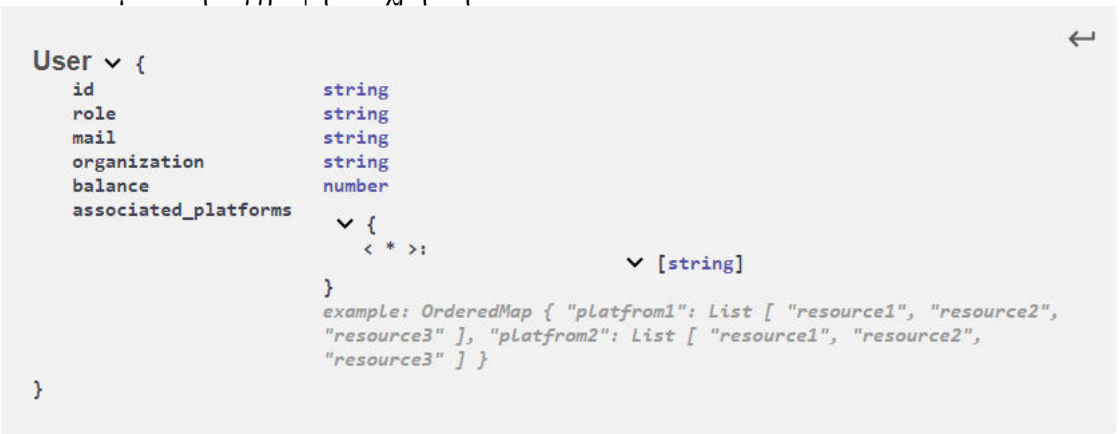
##### 5.2.2.1.1 Εγγραφή χρήστη στο BaaS

Αφορά την εγγραφή ενός χρήστη στο BaaS σύστημα. Δέχεται ως είσοδο το αντικείμενο (object) με τις πληροφορίες ενός χρήστη και επιστρέφει μήνυμα επιτυχίας/αποτυχίας εγγραφής του, όπως φαίνεται στην Εικόνα 60.




*Εικόνα 60: Παράμετροι του endpoint εγγραφής χρήστη στο BaaS και απόκριση συστήματος.*

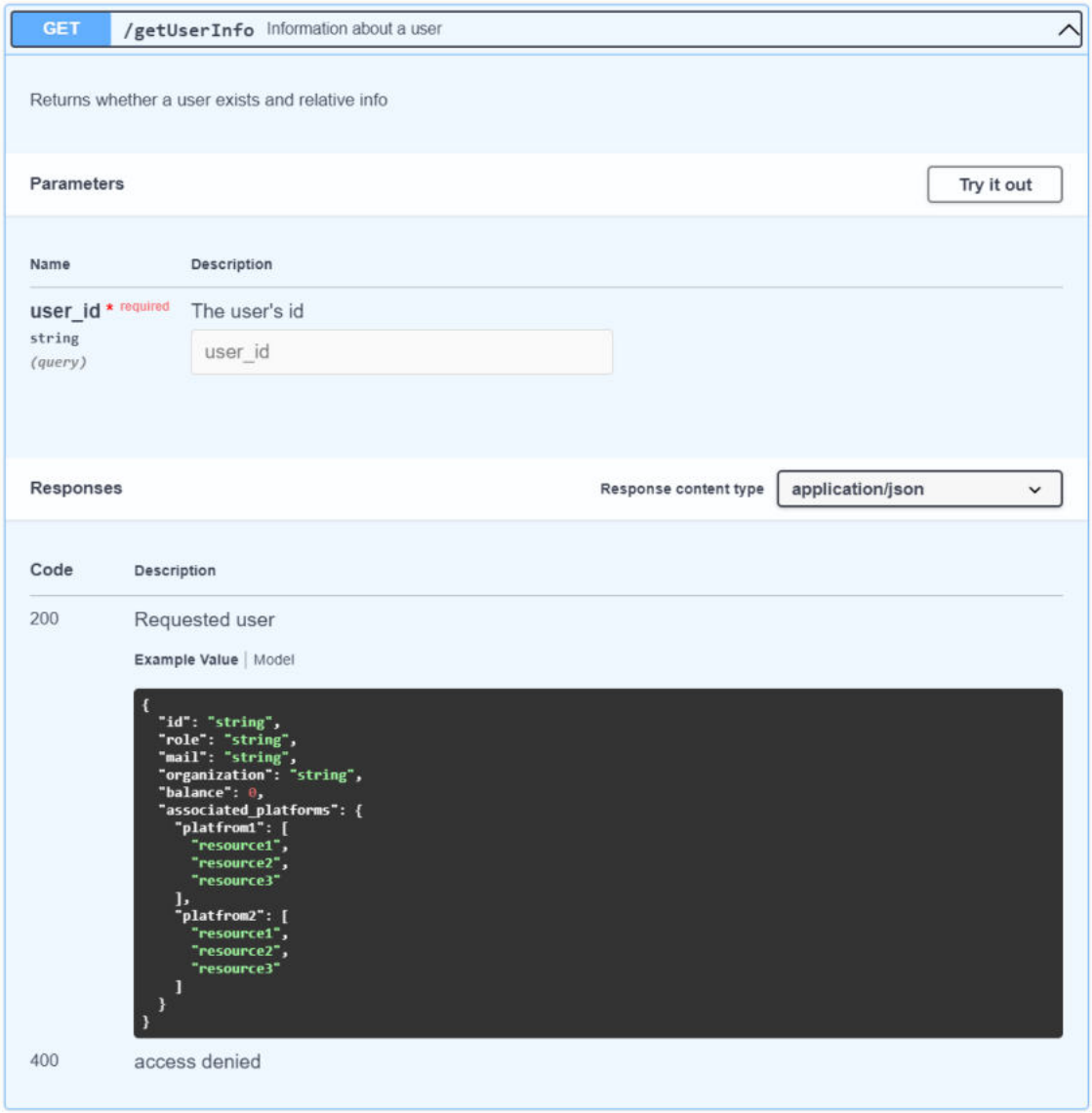
Αντίστοιχα, στην Εικόνα 61 παρουσιάζεται το αντικείμενο που επιστρέφεται από το σύστημα του BaaS μετά την εγγραφή του χρήστη σε αυτό.




*Εικόνα 61: Αντικείμενο που επιστρέφεται από το BaaS μετά την εγγραφή χρήστη*

#### 5.2.2.1.2 Ανάκτηση πληροφοριών χρήστη από το BaaS

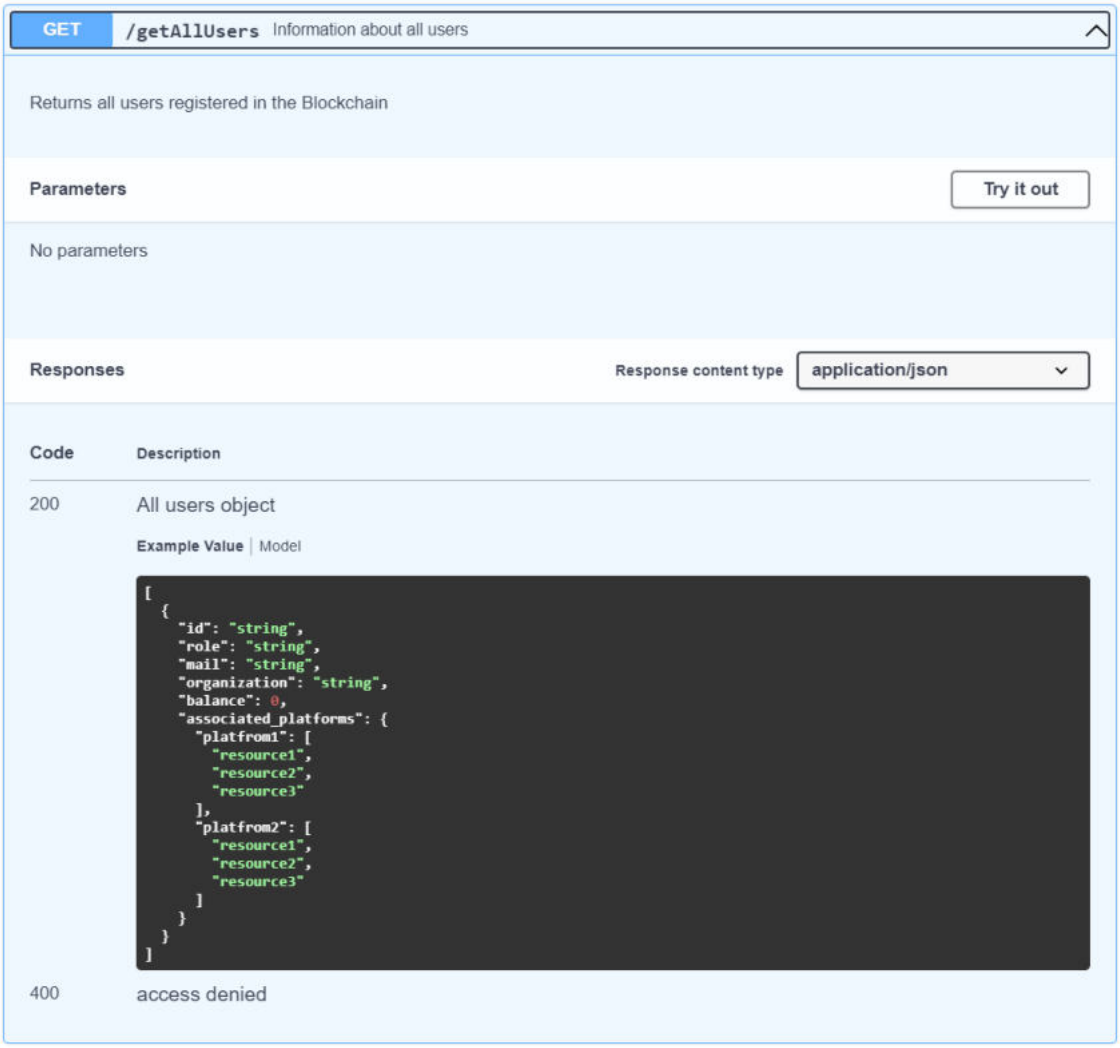
Αφορά την ανάκτηση πληροφοριών ενός χρήστη, όπως αυτές είναι αποθηκευμένες στο BaaS, από άλλα συστήματα του IoTFeds. Δέχεται ως είσοδο το αναγνωριστικό ενός χρήστη (user_id) και επιστρέφει μήνυμα επιτυχίας ή αποτυχίας συνοδευόμενο από object με τις πληροφορίες του χρήστη, όπως φαίνεται στην Εικόνα 62.




*Εικόνα 62: Παράμετροι του endpoint ανάκτησης πληροφοριών χρήστη από το BaaS και απόκριση συστήματος.*

#### 5.2.2.1.3 Ανάκτηση πληροφοριών όλων των χρηστών του BaaS

Πρόκειται για endpoint που αφορά την πρόσβαση σε πληροφορίες σχετικά με όλους τους χρήστες του BaaS και παρουσιάζεται αναλυτικά στην Εικόνα 63. Το endpoint αυτό μπορεί να αξιοποιηθεί για την ανακάλυψη χρηστών από διαχειριστές ομοσπονδίας και την πρόσκλησή τους για συμμετοχή στην ομοσπονδία τους.



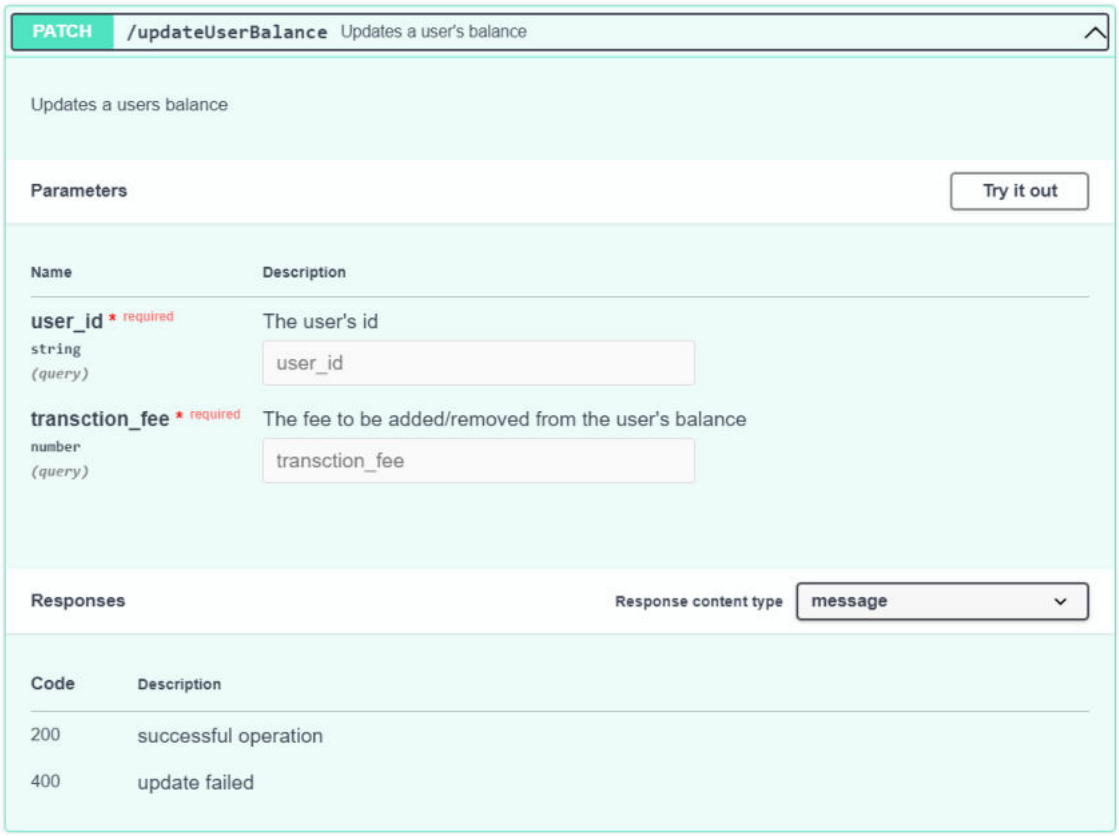

*Εικόνα 63: Παράμετροι του endpoint ανάκτησης πληροφοριών όλων των χρηστών από το BaaS και απόκριση συστήματος.*

Το συγκεκριμένο endpoint, χωρίς να δέχεται κάτι ως είσοδο, επιστρέφει μονοδιάστατο πίνακα από αντικείμενα, όπου κάθε ένα περιγράφει έναν χρήστη του BaaS, όπως παρουσιάζεται στην Εικόνα 64Εικόνα 64.


*Εικόνα 64: Μονοδιάστατος πίνακας από objects που επιστρέφεται στην επιτυχή ανάκτηση πληροφοριών χρηστών από το BaaS.*

#### 5.2.2.1.4 Ενημέρωση υπολοίπου χρήστη στο BaaS

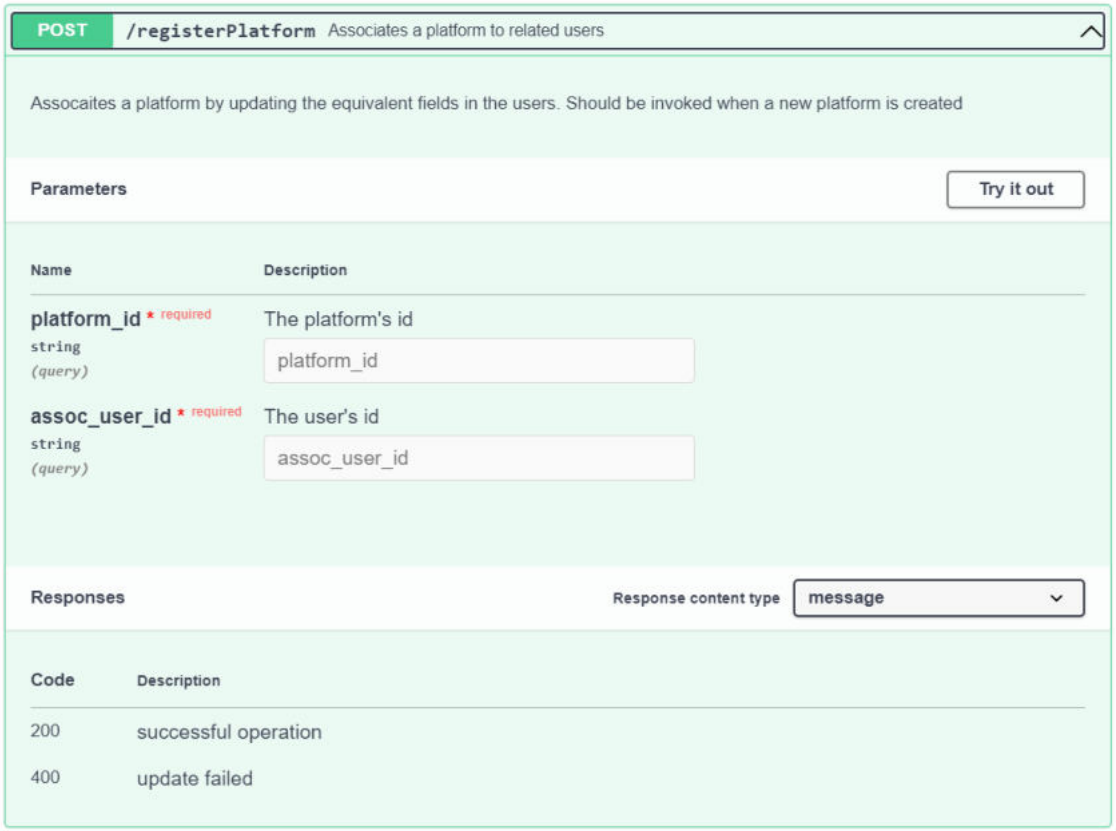
Το συγκεκριμένο endpoint καλείται για την αλλαγή του υπολοίπου ενός χρήστη μετά από κάποια συναλλαγή. Όπως φαίνεται στην Εικόνα 65, δέχεται ως εισόδους το αναγνωριστικό (user_id) του χρήστη, μαζί με το ποσό της συναλλαγής (transaction_fee), συνοδευόμενο από το κατάλληλο πρόσημο (+ για προσθήκη και – για αφαίρεση χρημάτων από τον λογαριασμό του χρήστη) και επιστρέφει μήνυμα επιτυχίας/αποτυχίας.




*Εικόνα 65: Παράμετροι του endpoint ενημέρωσης υπολοίπου χρήστη στο BaaS και απόκριση συστήματος.*

#### 5.2.2.1.5 Εγγραφή IoT πλατφόρμας στο BaaS

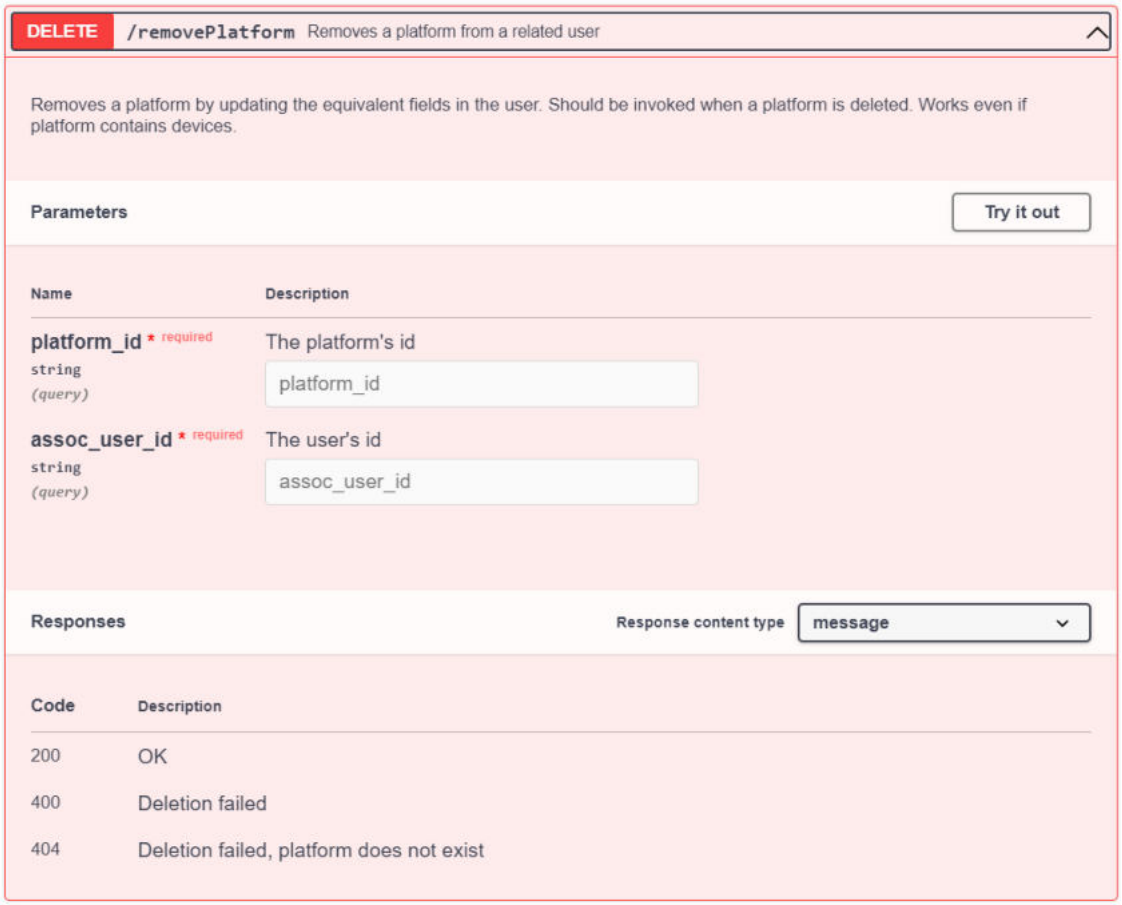
Χρησιμοποιείται για την εγγραφή μιας IoT πλατφόρμας στο BaaS. Καθώς η πληροφορία αυτή εγγράφεται στο μοντέλο πληροφορίας του κάθε χρήστη, το endpoint αυτό ενεργοποιεί κατάλληλες συναρτήσεις οι οποίες εγγράφουν την πλατφόρμα μαζί με τα resources που διαθέτει ο κάθε χρήστης σε αυτήν, στο μοντέλο πληροφορίας του κάθε χρήστη. Όπως φαίνεται στην Εικόνα 66, το συγκεκριμένο endpoint δέχεται ως είσοδο το αναγνωριστικό της πλατφόρμας (platform_id) καθώς και το αναγνωριστικό του χρήστη που σχετίζεται με τη συγκεκριμένη πλατφόρμα (assoc_user_id) και εμφανίζει μήνυμα επιτυχούς ή ανεπιτυχούς εγγραφής.




*Εικόνα 66: Παράμετροι του endpoint εγγραφής IoT πλατφόρμας στο BaaS και απόκριση συστήματος.*

#### 5.2.2.1.6 Αφαίρεση IoT πλατφόρμας από το BaaS

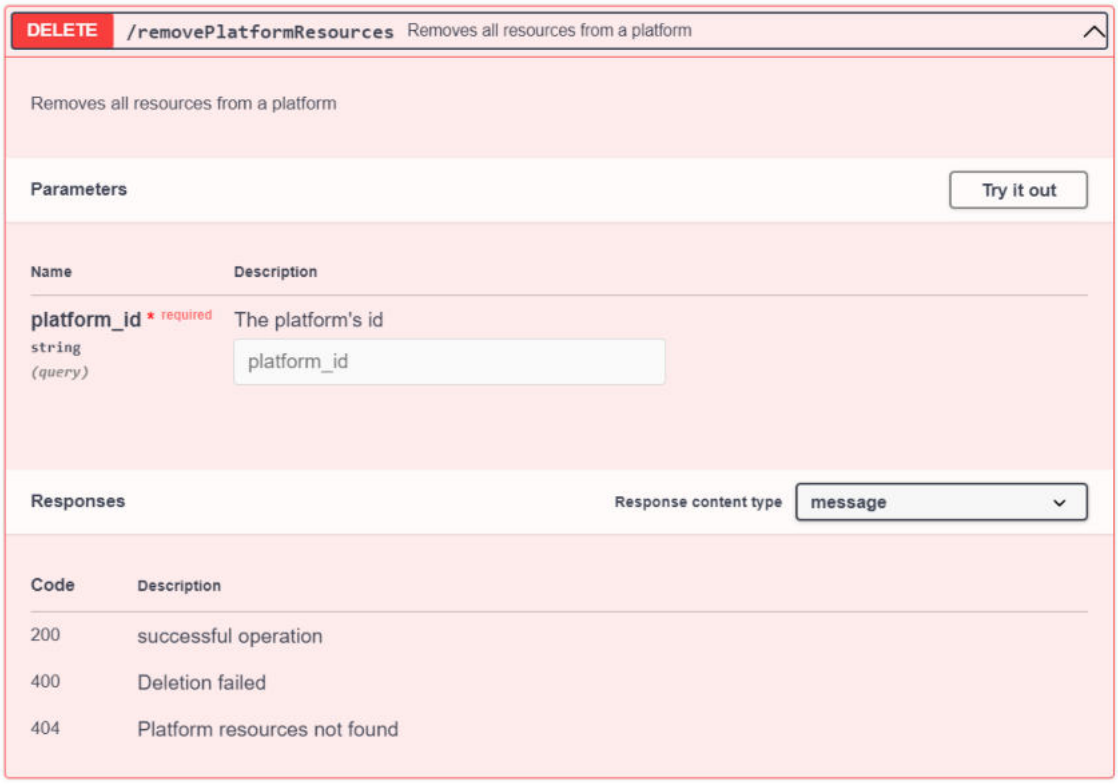
Χρησιμοποιείται για την αφαίρεση μιας πλατφόρμας από το BaaS και δέχεται τις ίδιες εισόδους με το προηγούμενο endpoint, όπως φαίνεται στην Εικόνα 67. Η εν λόγω πλατφόρμα, μαζί με όλες τι πιθανές περιεχόμενες συσκευές, διαγράφεται από το αντίστοιχο πεδίο του χρήστη.




*Εικόνα 67: Παράμετροι του endpoint αφαίρεσης IoT πλατφόρμας από το BaaS και απόκριση συστήματος.*

#### 5.2.2.1.7 Αφαίρεση συσκευών IoT πλατφόρμας από το BaaS

Χρησιμοποιείται για την αφαίρεση των περιεχομένων συσκευών μιας πλατφόρμας από το BaaS και δέχεται ως είσοδο το αναγνωριστικό της πλατφόρμας αυτής (platform_id), όπως φαίνεται στην Εικόνα 68. Οι συσκευές της εν λόγω πλατφόρμας, διαγράφονται από το αντίστοιχο πεδίο του χρήστη.

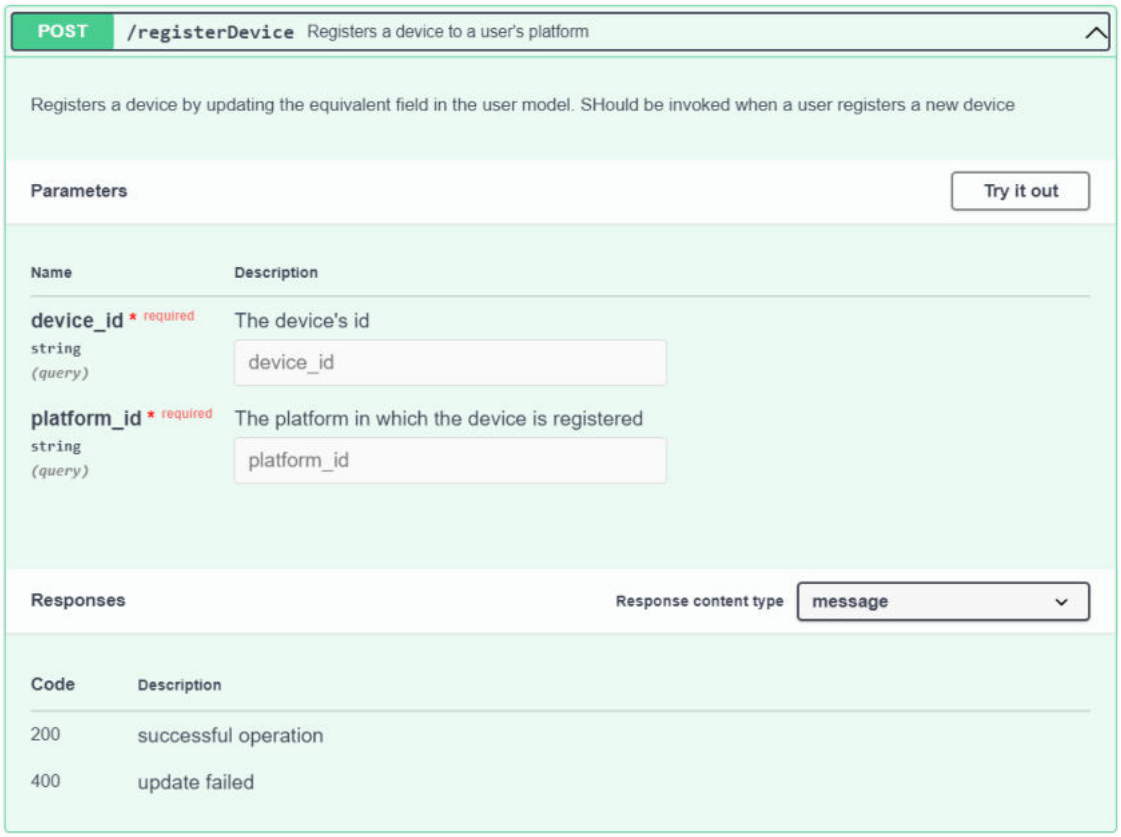



*Εικόνα 68: Παράμετροι του endpoint αφαίρεσης συσκευών IoT πλατφόρμας από το BaaS και απόκριση συστήματος.*

#### 5.2.2.1.8 Εγγραφή IoT συσκευής στο BaaS

Αφορά την εγγραφή μιας IoT συσκευής στο BaaS. Επειδή οι συσκευές εγγράφονται στο μοντέλο πληροφορίας του χρήστη, και εντός των συσχετιζόμενων πλατφορμών, το συγκεκριμένο endpoint αναζητά το αντίστοιχο πεδίο στο μοντέλο πληροφορίας κάθε χρήστη και γράφει εντός αυτού. Προϋπόθεση για να πετύχει, είναι οι πλατφόρμες που συσχετίζονται με την εγγραφόμενη συσκευή να έχουν ήδη συσχετιστεί με τους επιμέρους χρήστες μέσω του endpoint της προηγούμενης υπο-ενότητας.

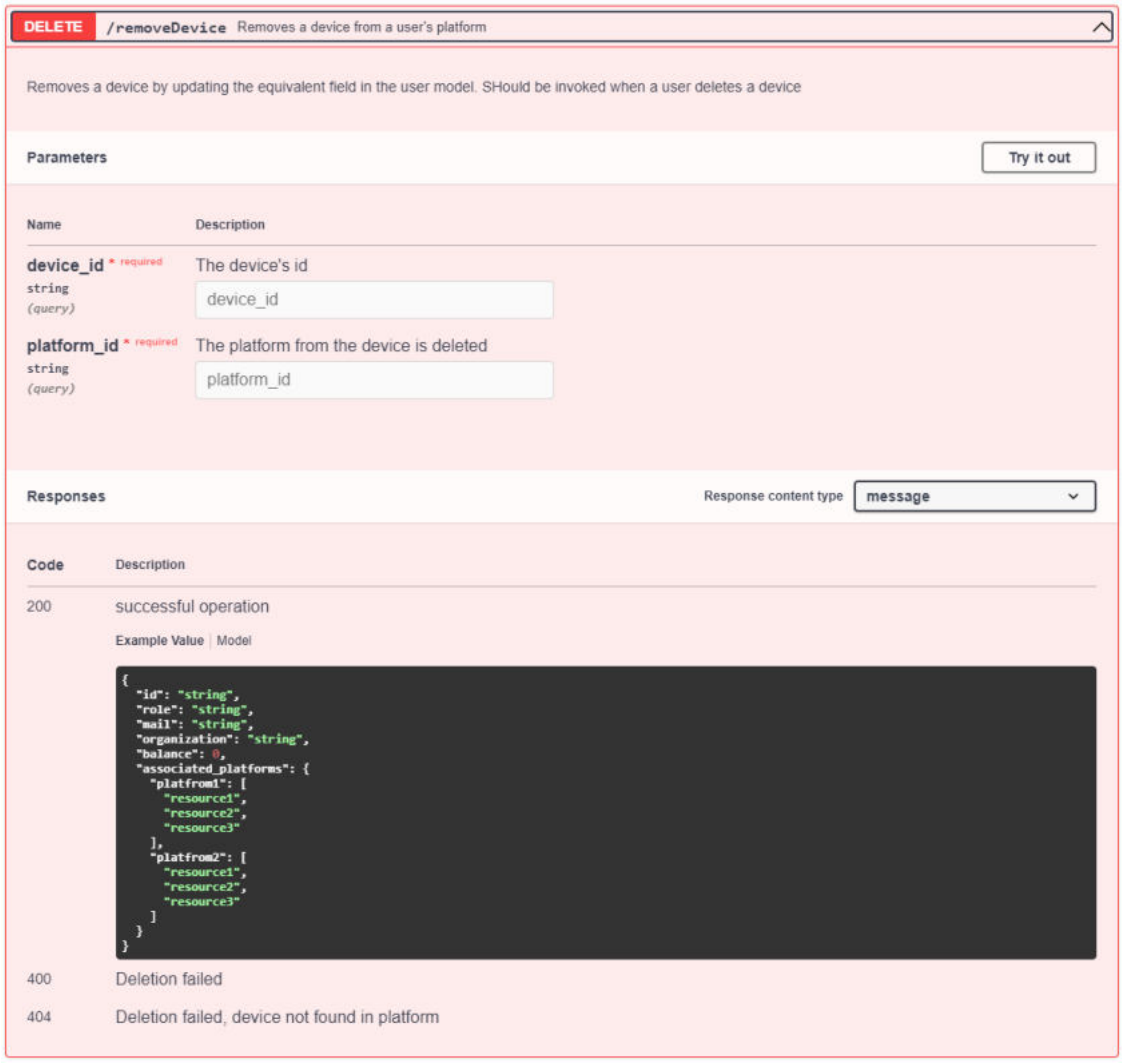
Όπως φαίνεται στην Εικόνα 69, το endpoint αυτό δέχεται ως είσοδο το αναγνωριστικό της συσκευής (device_id) και το αναγνωριστικό της συσχετιζόμενης πλατφόρμας (platform_id) και επιστρέφει μήνυμα επιτυχούς ή ανεπιτυχούς εγγραφής της συσκευής.




*Εικόνα 69: Παράμετροι του endpoint εγγραφής IoT συσκευής στο BaaS και απόκριση συστήματος.*

#### 5.2.2.1.9 Αφαίρεση IoT συσκευής από πλατφόρμα στο BaaS

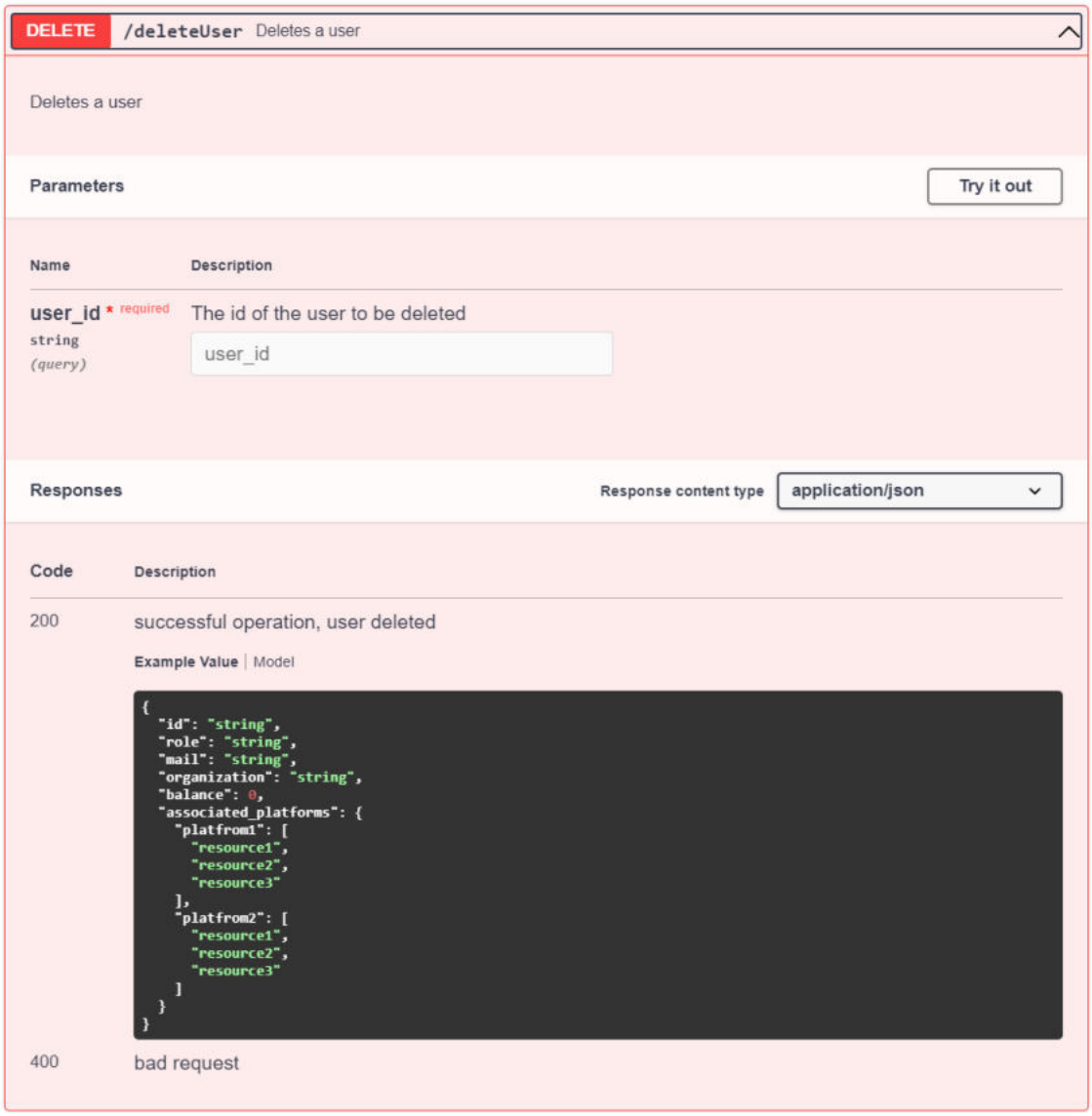
Αφορά την αφαίρεση μιας συσκευής από μια πλατφόρμα στο BaaS και, όπως φαίνεται στην Εικόνα 70, δέχεται τις ίδιες εισόδους με το προηγούμενο endpoint. Αρχικά, εντοπίζεται ο χρήστης-κάτοχος της πλατφόρμας και, στη συνέχεια, από τον πίνακα συσκευών στο υπό-πεδίο της συγκεκριμένης πλατφόρμας και στο πεδίο των συσχετιζόμενων πλατφορμών αφαιρείται η εν λόγω συσκευή.




*Εικόνα 70: Παράμετροι του endpoint αφαίρεσης IoT συσκευής από πλατφόρμα στο BaaS και απόκριση συστήματος.*

#### 5.2.2.1.10 Διαγραφή χρήστη από το BaaS

Το συγκεκριμένο endpoint αφορά την διαγραφή ενός χρήστη από το BaaS. Όπως φαίνεται στην Εικόνα 71, δέχεται ως είσοδο το αναγνωριστικό τους χρήστη (user_id) και επιστρέφει μήνυμα επιτυχίας/αποτυχίας ή αδυναμίας διαγραφής του χρήστη για συγκεκριμένους λόγους (ανήκει σε ομοσπονδία ή έχει οικονομικές υποχρεώσεις). Σε περίπτωση επιτυχίας, το μήνυμα συνοδεύεται από αντικείμενο που περιγράφει τον διαγεγραμμένο χρήστη.



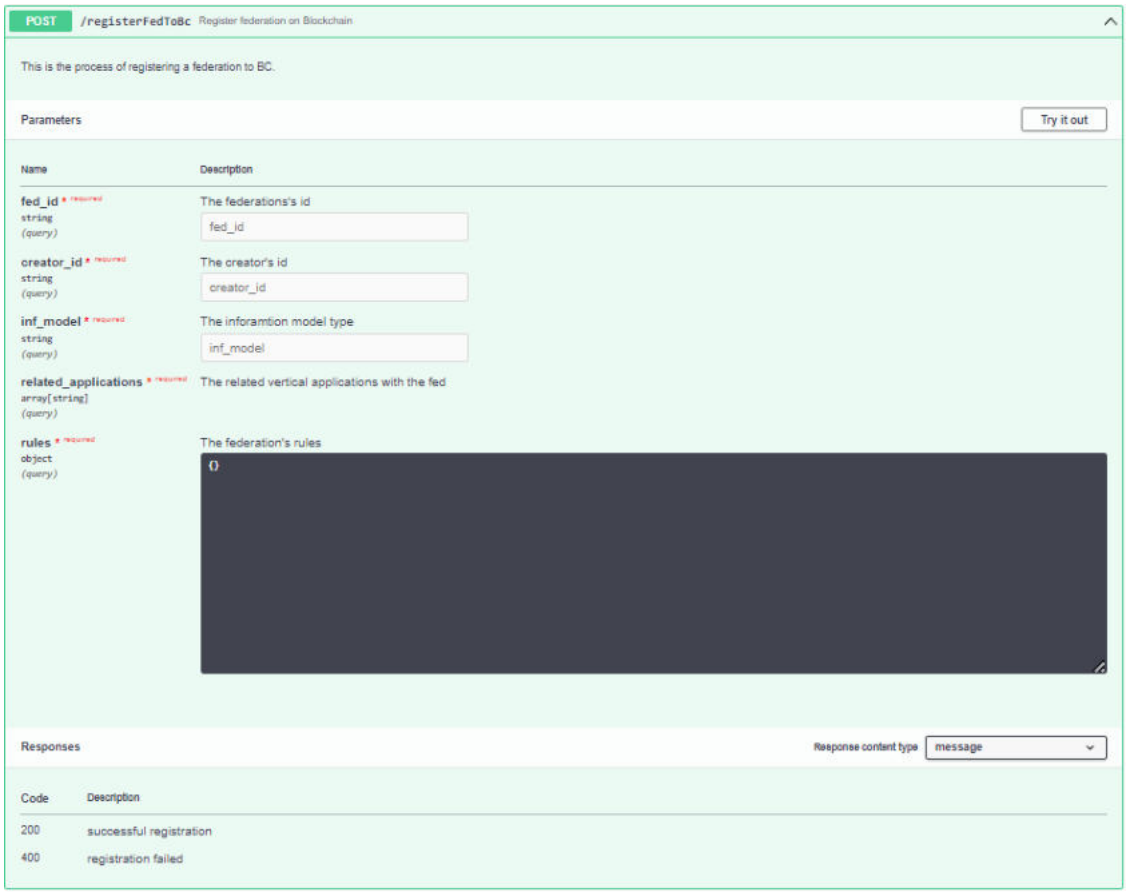

*Εικόνα 71: Παράμετροι του endpoint διαγραφής χρήστη από το BaaS και απόκριση συστήματος.*

### *5.2.2.2 Στοιχείο λογισμικού blockchain για διαχείριση ομοσπονδιών*

Το στοιχείο αυτό αφορά όλα τα endpoints που σχετίζονται με την διαχείριση ομοσπονδιών στο BaaS σύστημα του IoTFeds. Ακολουθεί μία σύντομη περιγραφή για το καθένα από αυτά.

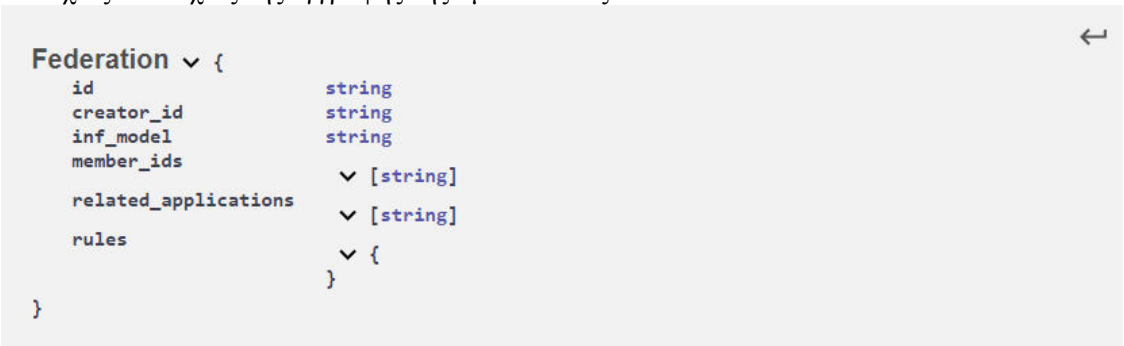
#### 5.2.2.2.1 Εγγραφή ομοσπονδίας στο BaaS

Το συγκεκριμένο endpoint σχετίζεται με την εγγραφή μιας ομοσπονδίας στο BaaS και παρουσιάζεται αναλυτικά στην Εικόνα 72. Ως είσοδο δέχεται το αναγνωριστικό της ομοσπονδίας (fed_id), το αναγνωριστικό του IoTFeds χρήστη που δημιούργησε τη συγκεκριμένη ομοσπονδία (creator_id), έναν πίνακα από String που περιλαμβάνει τις συσχετιζόμενες πιλοτικές πλατφόρμες (related_applications), καθώς και ένα αντικείμενο με τους κανόνες της ομοσπονδίας (rules).




*Εικόνα 72: Παράμετροι του endpoint εγγραφής ομοσπονδίας στο BaaS και απόκριση συστήματος.*

Ενδεικτικό παράδειγμα ενός τέτοιου αντικειμένου φαίνεται στην Εικόνα 73. Επιστρέφει μήνυμα επιτυχίας/αποτυχίας της εγγραφής της ομοσπονδίας.



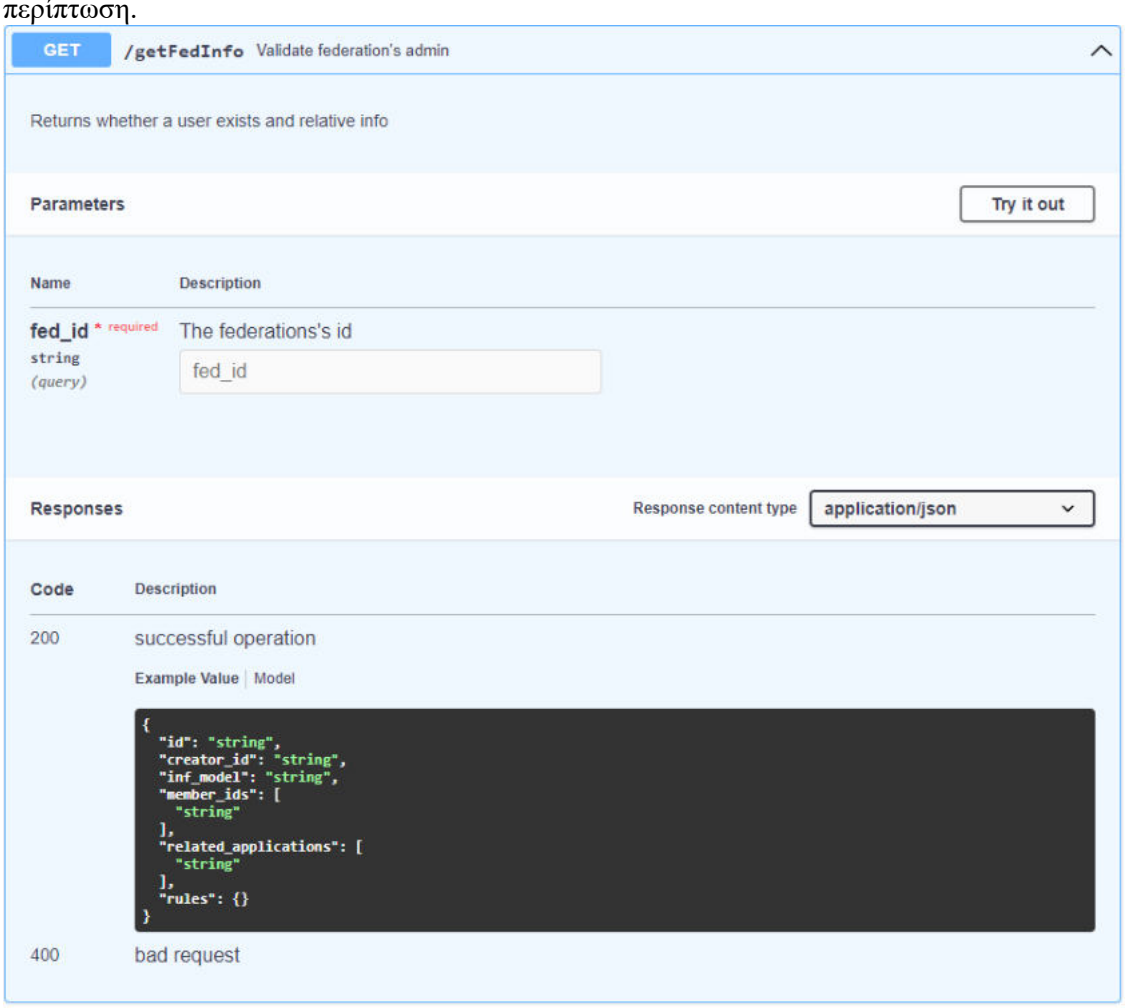

*Εικόνα 73: Αντικείμενο ομοσπονδίας στο BaaS.*

Αξίζει να σημειωθεί πως για λόγους συνάφειας το σύστημα δεν θα επιτρέπει την αρχικοποίηση μιας ομοσπονδίας με αρχικά μέλη πέραν του δημιουργού. Επιπλέον, το πεδίο «related_applications» αφορά πιθανές εφαρμογές που η ομοσπονδία είναι σε θέση να παρέχει μέσω του IoTFeds. Ενδεικτικά παραδείγματα αυτών των εφαρμογών είναι οι πιλοτικές πλατφόρμες (Smart Environment, Zastel/Spotypal και UiTOP) που θα υλοποιηθούν στο τελευταίο στάδιο του έργου. Τέλος, το πεδίο «rules» αφορά δομημένα αντικείμενα που ορίζουν το πλαίσιο κανόνων με το οποίο λειτουργεί μια ομοσπονδία. Η ανάλυση των διαφορετικών επιλογών για αυτό το πεδίο έχει γίνει στο υποκεφάλαιο 3.3.3.

#### 5.2.2.2.2 Ανάκτηση πληροφοριών ομοσπονδίας από το BaaS

Αφορά την πρόσβαση σε πληροφορίες μιας ομοσπονδίας, όπως αυτές είναι αποθηκευμένες στο BaaS, από άλλα συστήματα της IoTFeds πλατφόρμας και δέχεται ως είσοδο το αναγνωριστικό της ομοσπονδίας (fed_id). Όπως φαίνεται στην Εικόνα 74, επιστρέφει μήνυμα

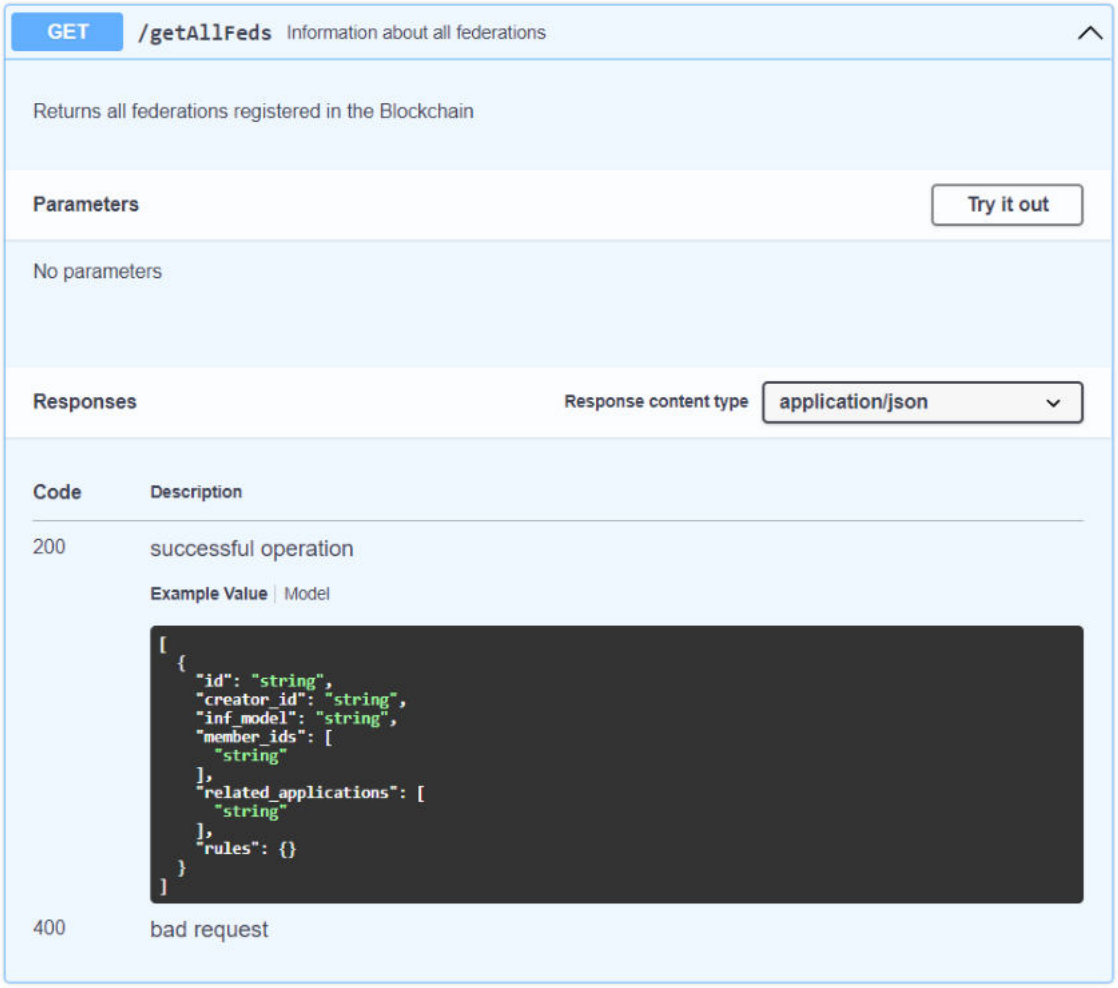
επιτυχίας/αποτυχίας συνοδευόμενο από αντικείμενο σαν αυτό στην Εικόνα 73 για την πρώτη περίπτωση.




*Εικόνα 74: Παράμετροι του endpoint ανάκτησης πληροφοριών ομοσπονδίας από το BaaS και απόκριση συστήματος.*

#### 5.2.2.2.3 Ανάκτηση πληροφοριών όλων των ομοσπονδιών από το BaaS

Πρόκειται για το endpoint που σχετίζεται με την ανάκτηση των πληροφοριών όλων των ομοσπονδιών που βρίσκονται εγγεγραμμένες στο BaaS και παρουσιάζεται αναλυτικά στην Εικόνα 75.




*Εικόνα 75: Παράμετροι του endpoint ανάκτησης πληροφοριών όλων των ομοσπονδιών από το BaaS και απόκριση συστήματος.*

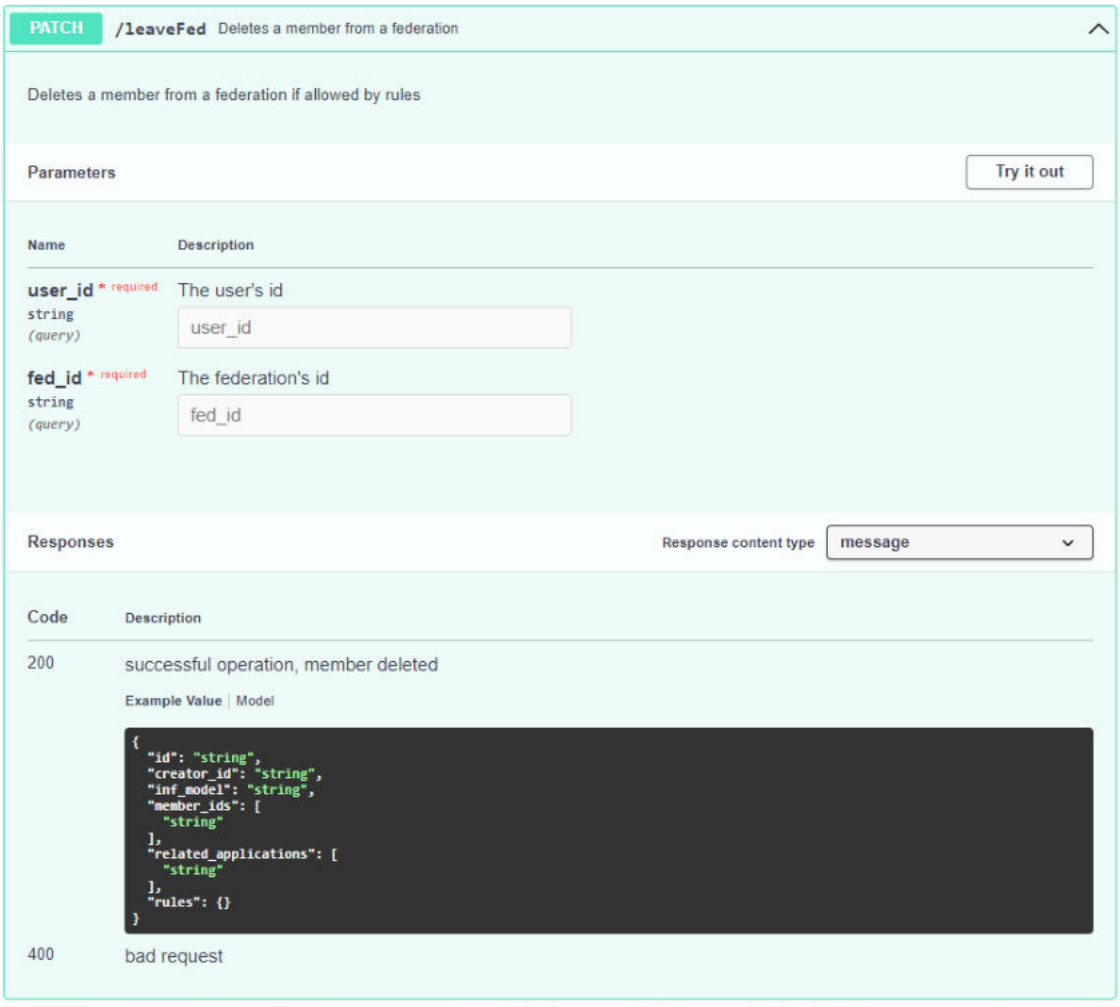
Χωρίς να δέχεται κάποια είσοδο, επιστρέφει μονοδιάστατο πίνακα με αντικείμενα σαν αυτό στην Εικόνα 76. Οπότε ο πίνακας αυτός περιέχει πληροφορίες για όλες τις ομοσπονδίες στο BaaS. Το endpoint αυτό μπορεί να αξιοποιηθεί για ανακάλυψη ομοσπονδιών από χρήστη ώστε να αιτηθεί την συμμετοχή του σε αυτές.


```
[
  {
    "id": "string",
    "creator_id": "string",
    "inf_model": "string",
    "member_ids": [
      "string"
    ],
    "related_applications": [
      "string"
    ],
    "rules": {}
  }
]
```


*Εικόνα 76: Πίνακας από objects που επιστρέφεται μετά την επιτυχή ανάκτηση των πληροφοριών των ομοσπονδιών από το BaaS.*

#### 5.2.2.2.4 Αποχώρηση από ομοσπονδία στο BaaS

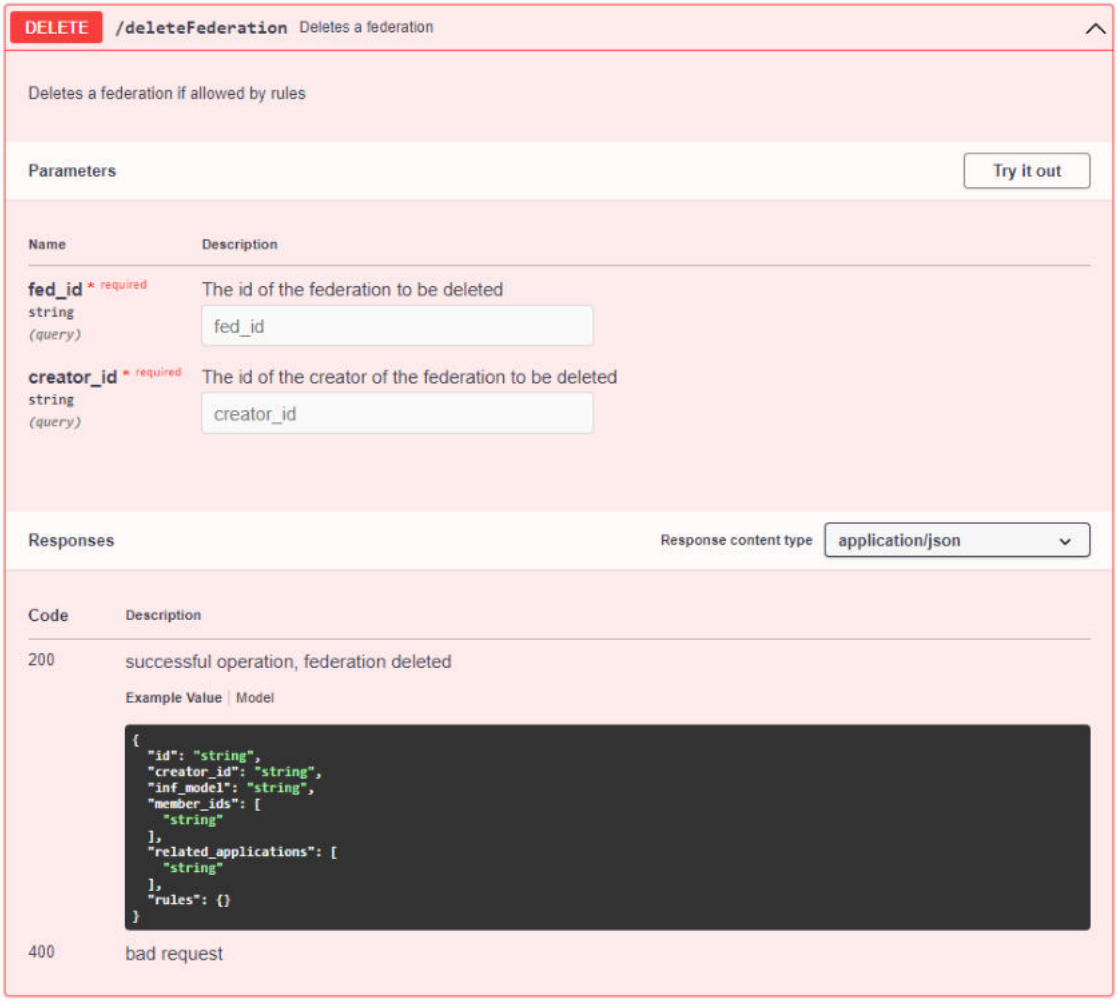
Πρόκειται για endpoint που αφορά την οικειοθελή αποχώρηση ενός χρήστη από μια ομοσπονδία του IoTFeds στο BaaS. Όπως φαίνεται στην Εικόνα 77, δέχεται ως είσοδο το αναγνωριστικό του χρήστη και της ομοσπονδίας και υπό συνθήκες, διαγράφει τον χρήστη από την λίστα μελών της ομοσπονδίας.




*Εικόνα 77: Παράμετροι του endpoint αποχώρησης από ομοσπονδία στο BaaS και απόκριση συστήματος.*

#### 5.2.2.2.5 Διαγραφή ομοσπονδίας από το BaaS

Το συγκεκριμένο endpoint αφορά τη διαγραφή μιας ομοσπονδίας από το BaaS και, όπως φαίνεται στην Εικόνα 78, δέχεται ως είσοδο το αναγνωριστικό της ομοσπονδίας που πρόκειται να διαγραφεί (fed_id) καθώς και το αναγνωριστικό του IoTFeds χρήστη που τη δημιούργησε (creator_id).




*Εικόνα 78: Παράμετροι του endpoint διαγραφής ομοσπονδίας από το BaaS και απόκριση συστήματος.*

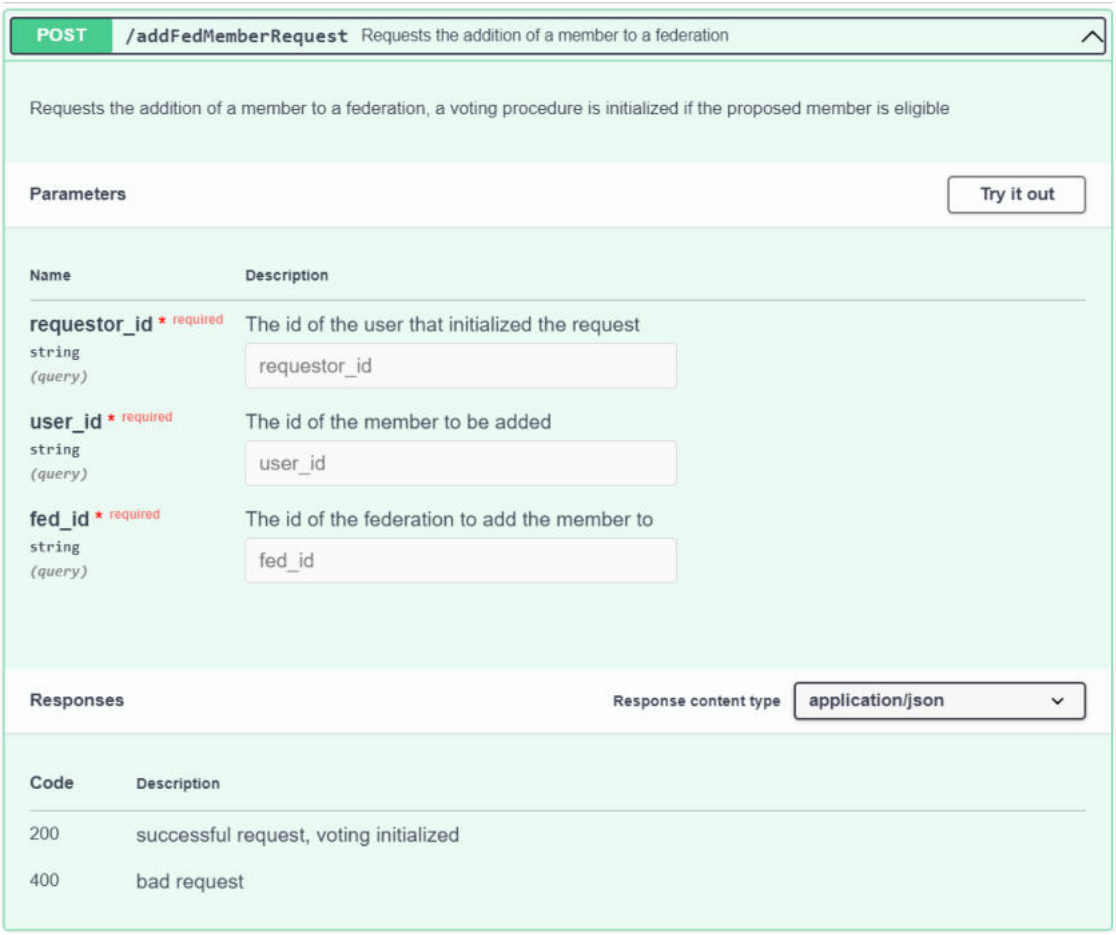
Σε περίπτωση επιτυχίας/αποτυχίας επιστρέφεται αντίστοιχο μήνυμα από το σύστημα. Πιο συγκεκριμένα, στην πρώτη περίπτωση επιστρέφεται αντικείμενο σαν αυτό στην Εικόνα 73 που περιέχει τις πληροφορίες της ομοσπονδίας που διεγράφη.

#### *5.2.2.3 Στοιχείο λογισμικού BaaS για διαχείριση ψηφοφορίας σε ομοσπονδίες*

Πρόκειται για στοιχείο λογισμικού που αφορά την διαχείριση ψηφοφοριών σε ομοσπονδίες στο BaaS. Τα επιμέρους endpoints περιγράφονται παρακάτω.

##### 5.2.2.3.1 Αίτημα προσθήκης νέου μέλους σε ομοσπονδία στο BaaS

Το συγκεκριμένο endpoint αφορά την προσθήκη νέου μέλους σε ομοσπονδία μέσα από πρόσκληση. Όπως φαίνεται στην Εικόνα 79, δέχεται ως είσοδο το αναγνωριστικό του χρήστη που εκκίνησε την πρόσκληση, του υπό προσθήκη χρήστη καθώς και της εν λόγω ομοσπονδίας.




*Εικόνα 79: Παράμετροι του endpoint για αίτημα προσθήκης νέου μέλους σε ομοσπονδία στο BaaS και απόκριση συστήματος.*

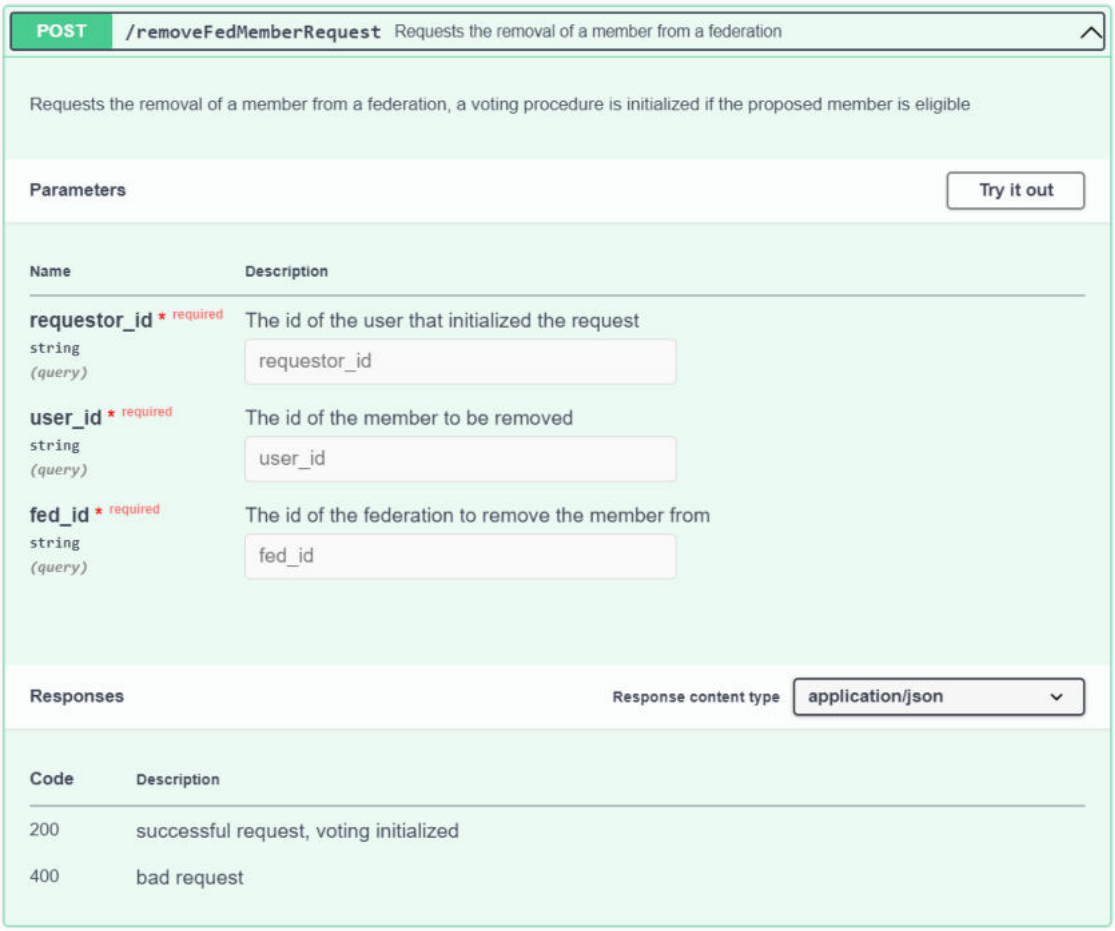
Σε περίπτωση επιτυχίας/αποτυχίας επιστρέφεται από το σύστημα αντίστοιχο μήνυμα. Πιο συγκεκριμένα, σε περίπτωση επιτυχίας εκκινείται διαδικασία ψηφοφορίας και δημιουργείται αντικείμενο σαν αυτό στην Εικόνα 80. Επίσης δημιουργείται ένα κουπόνι (token) για κάθε ψηφοφόρο για την πιο ασφαλή εξουσιοδότηση του. Το κουπόνι αυτό περνιέται ως παράμετρος στον προσωποποιημένο σύνδεσμο που αποστέλλεται μέσω «mail» κάθε χρήστη προκειμένου να ψηφίσει και καταναλώνεται αυτόματα από τα endpoints που υλοποιούν εξουσιοδότηση του χρήστη.


*Εικόνα 80: Αντικείμενο ψηφοφορίας στο BaaS*

#### 5.2.2.3.2 Αίτημα αφαίρεσης μέλους από ομοσπονδία στο BaaS

Το συγκεκριμένο endpoint αφορά την αφαίρεση μέλους από ομοσπονδία μέσα από πρόταση. Όπως φαίνεται στην Εικόνα 81, δέχεται ως είσοδο το αναγνωριστικό του χρήστη που εκκίνησε την πρόταση, του υπό αφαίρεση χρήστη καθώς και της εν λόγω ομοσπονδίας. Επιστρέφει

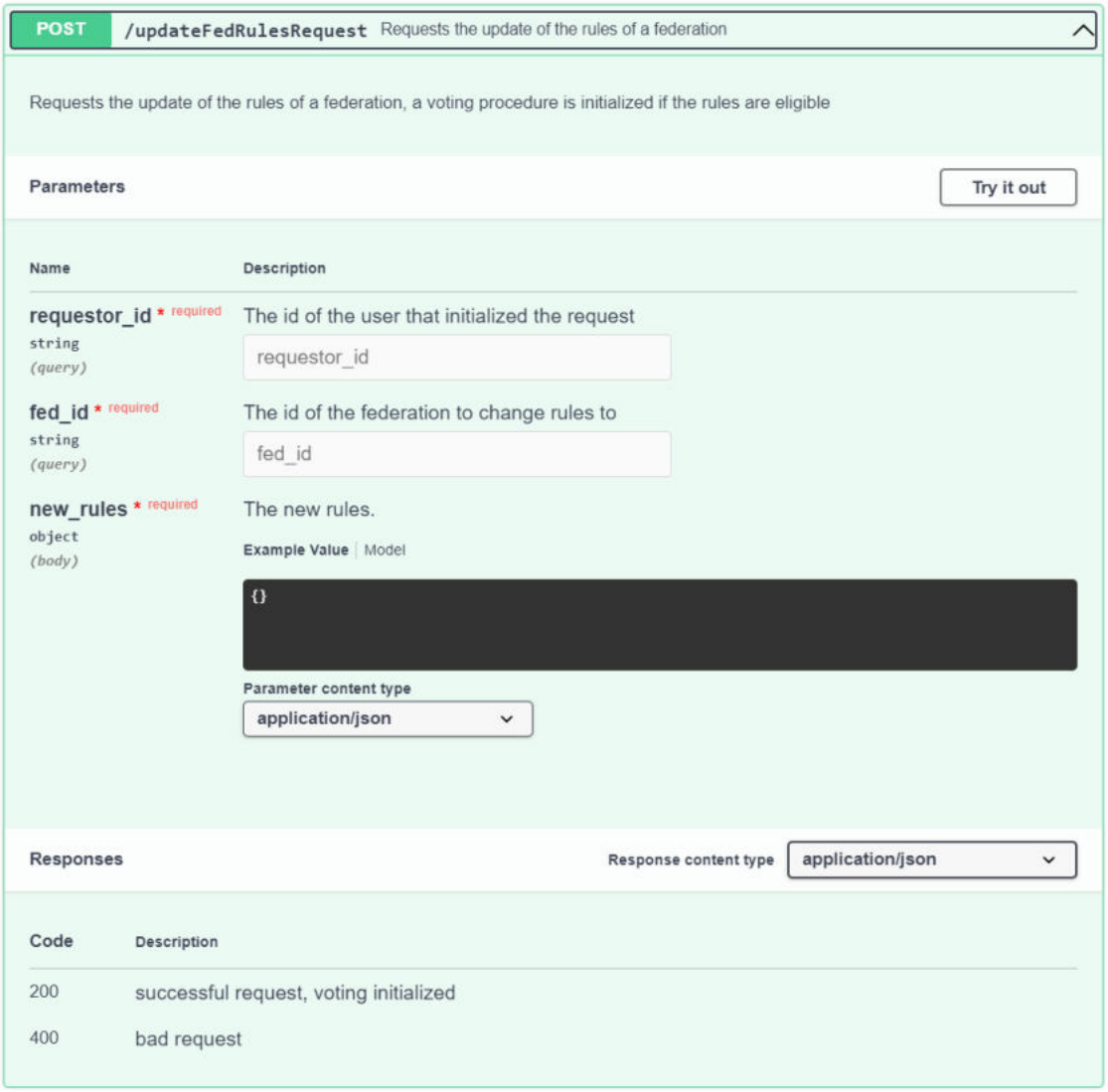
μήνυμα επιτυχίας/αποτυχίας, ενώ σε περίπτωση επιτυχίας εκκινείται διαδικασία ψηφοφορίας και δημιουργείται αντικείμενο σαν αυτό στην Εικόνα 80. Επίσης δημιουργείται ένα κουπόνι (token) για κάθε ψηφοφόρο για την πιο ασφαλή εξουσιοδότηση του. Το κουπόνι αυτό περνιέται ως παράμετρος στον προσωποποιημένο σύνδεσμο που αποστέλλεται μέσω «mail» κάθε χρήστη προκειμένου να ψηφίσει και καταναλώνεται αυτόματα από τα endpoints που υλοποιούν εξουσιοδότηση του χρήστη.




*Εικόνα 81: Παράμετροι του endpoint για αίτημα αφαίρεσης μέλους από ομοσπονδία στο BaaS και απόκριση συστήματος.*

#### 5.2.2.3.3 Αίτημα αλλαγής κανόνων ομοσπονδίας στο BaaS

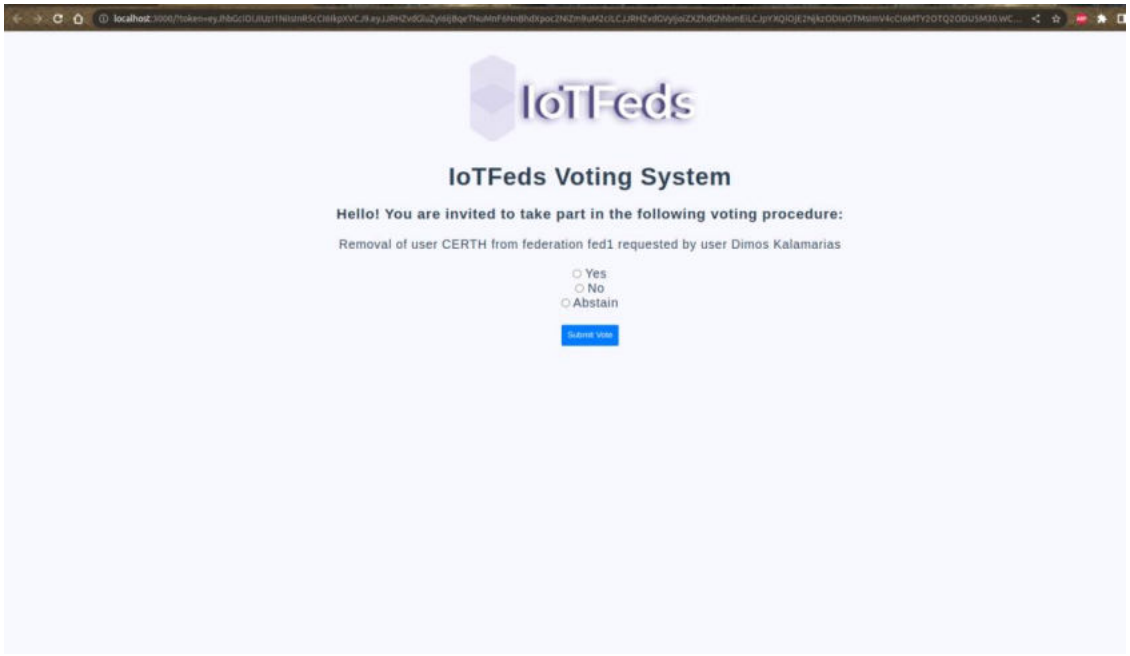
Το συγκεκριμένο endpoint αφορά την αλλαγή κανόνων σε ομοσπονδία μέσα από πρόταση. Όπως φαίνεται αναλυτικά στην Εικόνα 82, δέχεται ως είσοδο το αναγνωριστικό του χρήστη που εκκίνησε την πρόταση, της εν λόγω ομοσπονδίας καθώς και του προτεινόμενου σετ κανόνων. Επιστρέφει μήνυμα επιτυχίας/αποτυχίας, ενώ σε περίπτωση επιτυχίας εκκινείται διαδικασία ψηφοφορίας και δημιουργείται αντικείμενο σαν αυτό στην Εικόνα 80. Επίσης δημιουργείται ένα κουπόνι (token) για κάθε ψηφοφόρο για την πιο ασφαλή εξουσιοδότηση του. Το κουπόνι αυτό περνιέται ως παράμετρος στον προσωποποιημένο σύνδεσμο που αποστέλλεται μέσω «mail» κάθε χρήστη προκειμένου να ψηφίσει και καταναλώνεται αυτόματα από τα endpoints που υλοποιούν εξουσιοδότηση του χρήστη.



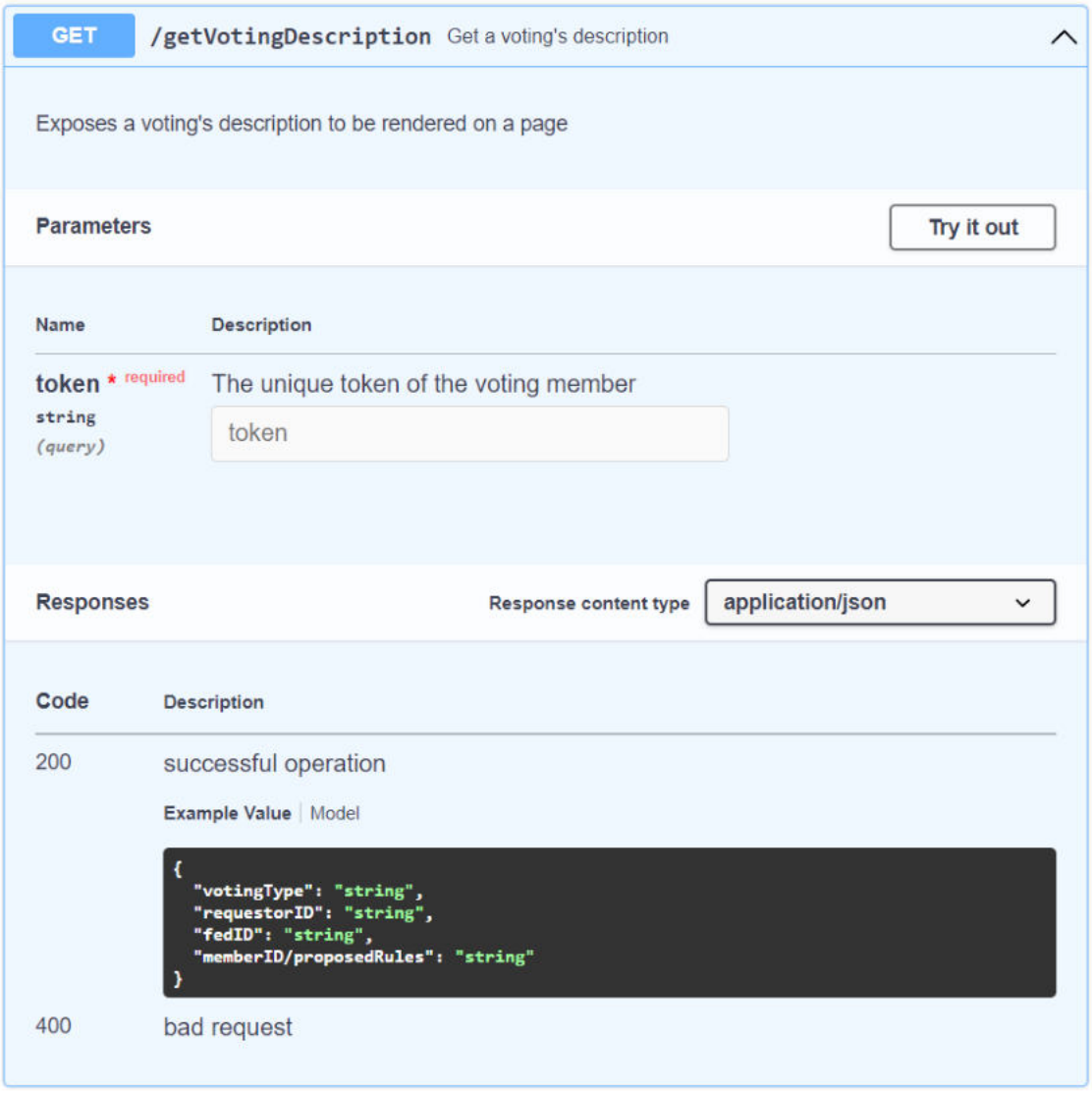

*Εικόνα 82: Παράμετροι του endpoint για αίτημα αλλαγής κανόνων στο BaaS και απόκριση συστήματος.*

#### 5.2.2.3.4 Πρόσβαση σε περιγραφή ψηφοφορίας στο BaaS

Το συγκεκριμένο endpoint αφορά την λήψη της περιγραφής μιας ψηφοφορίας. Η περιγραφή αυτή μπορεί να παρέχεται στον ψηφοφόρο-χρήστη προκειμένου να γνωρίζει το θέμα για το οποίο ψηφίζει. Όπως φαίνεται στην Εικόνα 84, δέχεται ως είσοδο το μοναδικό κουπόνι που διαθέτει κάθε χρήστης για ψηφοφορία και επιστρέφει την περιγραφή της που δημιουργείται αυτόματα από το smart contract κατά την αρχικοποίηση της ψηφοφορίας. Όπως προαναφέρθηκε το κουπόνι αυτό καταναλώνεται αυτόματα, αυτό σημαίνει ότι όταν ο ψηφοφόρος κάνει κλικ στον σύνδεσμο στο «mail» του, παραπέμπεται σε μια μια αρχική σελίδα η οποία αυτομάτως καλεί το endpoint αυτό διαβάζοντας το κουπόνι από τις παραμέτρους του συνδέσμου. Έπειτα από την επιτυχή κλήση του «endpoint» η περιγραφή που επιστρέφεται υλοποιείται σε μια ιστοσελίδα σαν αυτή της στην Εικόνα 83, μέσω της οποίας ο χρήστης δύναται να ψηφίσει αξιοποιώντας το επόμενο «endpoint». Παράδειγμα του κουπονιού είναι εμφανές στον σύνδεσμο στην κορυφή της εικόνας αυτής.



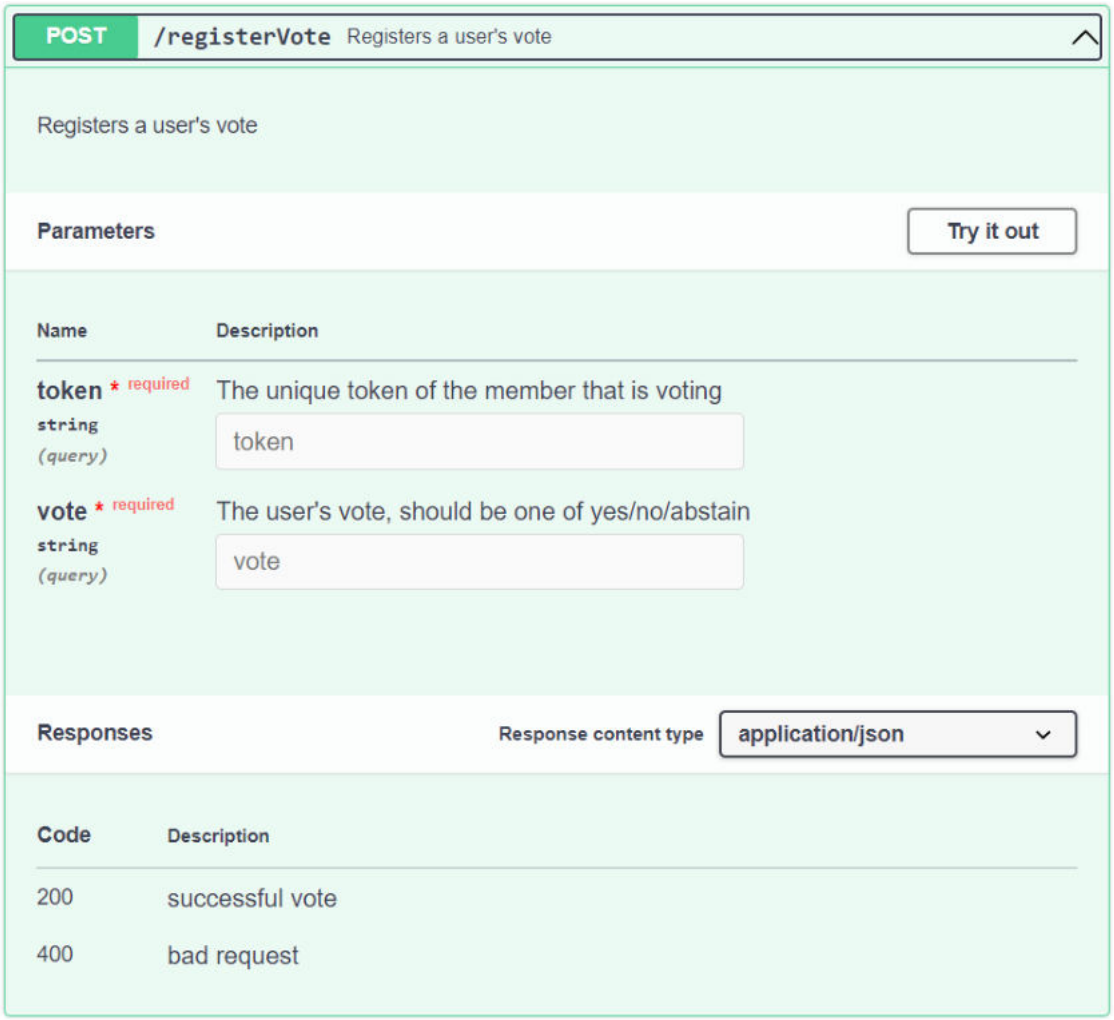

*Εικόνα 83: Παράδειγμα σελίδας ψηφοφορίας στο BaaS.*



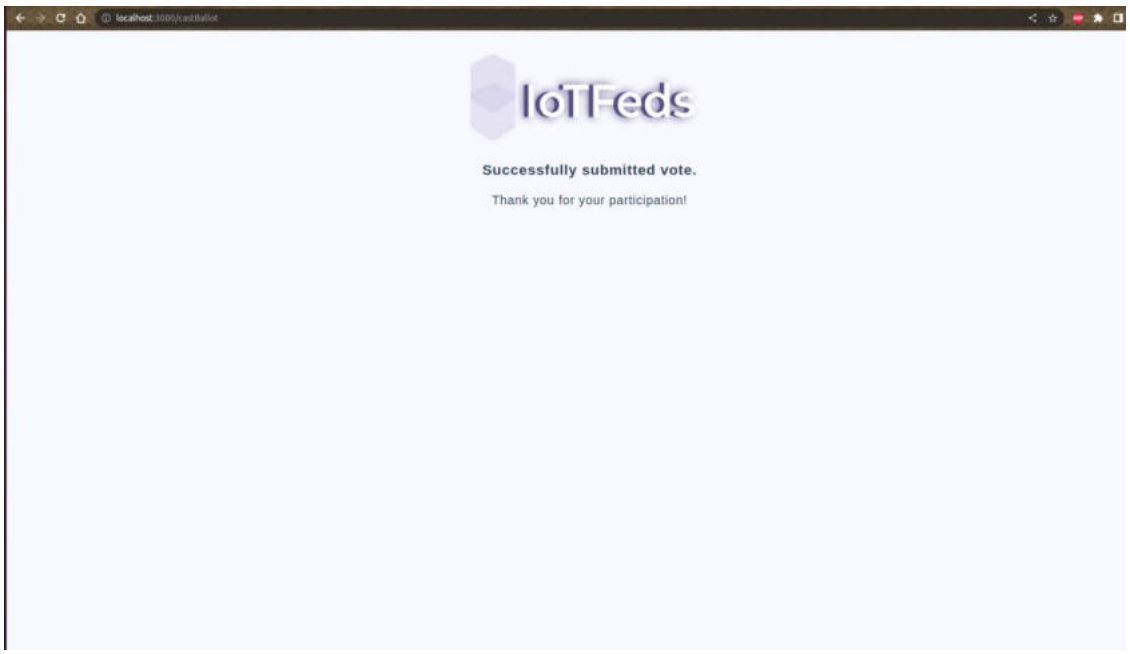

*Εικόνα 84: Παράμετροι του endpoint για πρόσβαση σε περιγραφή ψηφοφορίας στο BaaS και απόκριση συστήματος.*

#### 5.2.2.3.5 Καταγραφή ψήφου στο BaaS

Πρόκειται για endpoint που αφορά την καταγραφή ψήφου ενός χρήστη, για ψηφοφορία στην οποία έχει κληθεί να λάβει μέρος. Όπως φαίνεται στην Εικόνα 85, δέχεται ως είσοδο το το μοναδικό κουπόνι που διαθέτει κάθε χρήστης για ψηφοφορία καθώς και την ψήφο του (vote) με πιθανές τιμές yes/no/abstain και ενημερώνει το αντίστοιχο αντικείμενο ψηφοφορίας με την ψήφο του χρήστη. Όσον αφορά το «interface», το «endpoint» αυτό καλείται κατά την υποβολή της ψήφου από τον χρήστη (χρήση του πλήκτρου «Submit Vote» στην Εικόνα 83). Έπειτα ο χρήστης παραπέμπεται σε δεύτερη σελίδα με σχετικό περιεχόμενο επιτυχίας ή αποτυχίας της απόπειρας ψήφου του όπως φαίνεται στην .




*Εικόνα 85: Παράμετροι του endpoint καταγραφής ψήφου στο BaaS και απόκριση συστήματος.*




*Εικόνα 86 Παράδειγμα σελίδας έπειτα απο επιτυχή ψηφοφορία στο BaaS*

# 6 Συμπεράσματα

Στο παραδοτέο αυτό παρουσιάζονται η προσέγγιση που ακολουθείται και οι μηχανισμοί για τη διαχείριση μίας ομοσπονδίας στα πλαίσια του έργου IoTFeds. Η προσέγγιση αυτή βασίστηκε στις απαιτήσεις και την αρχιτεκτονική του συστήματος καθώς και τις τεχνολογίες που περιγράφηκαν σε προηγούμενα παραδοτέα (Π1.2 και Π1.3) αξιοποιώντας τους υπάρχοντες μηχανισμούς του ενδιάμεσου λογισμικού του symbIoTe και επεκτείνοντας τους με μηχανισμούς blockchain. Ο επιμέρους αρχιτεκτονικός σχεδιασμός για τη διαχείριση των IoTFeds ομοσπονδιών συνοψίζεται στο Κεφάλαιο 2.

Στο κείμενο που ακολουθεί (Κεφάλαιο 3) αναλύονται οι βασικοί μηχανισμοί στους οποίους στηρίζεται η διαχείριση της ομοσπονδίας. Αυτοί είναι οι μηχανισμοί blockchain για την αποθήκευση και την ενημέρωση των πληροφοριών σχετικά με τις ομοσπονδίες IoTFeds και την χρήση έξυπνων συμβολαίων που ορίζουν τις επιτρεπόμενες ενέργειες σε αυτές, οι μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης για την πρόσβαση σε πόρους και στοιχεία με ασφαλή τρόπο και την επαλήθευση της αυθεντικότητας και τέλος, οι μηχανισμοί για την αποκεντρωμένη και αυτόνομη διακυβέρνηση των ομοσπονδιών IoTFeds ως αποκεντρωμένοι αυτόνομοι οργανισμοί.

Στη συνέχεια (Κεφάλαιο 4) περιγράφεται η προσέγγιση που ακολουθείται για την επίτευξη της σημασιολογικής διαλειτουργικότητας και την ανάπτυξη του μοντέλου πληροφορίας του συστήματος IoTFeds το οποίο θα αποτελέσει την βάση για τον ορισμό των μοντέλων ομοσπονδιών IoTFeds. Για την ανάπτυξη του μοντέλου εξετάστηκαν τα τελευταία πρότυπα μοντέλων πληροφορίας και οντολογιών στο Διαδίκτυο των Πραγμάτων και την Έξυπνη Πόλη όπως NGSI-LD και SAREF/SAREF4CITY, καθώς και τα μοντέλα των πιλοτικών πλατφορμών. Οι βασικές τροποποιήσεις που προέκυψαν αφορούν την προσθήκη νέων οντοτήτων στο μοντέλο ή αντιστοιχιών υπαρχουσών οντοτήτων στο μοντέλο με οντότητες των μοντέλων που εξετάσθηκαν (SAREF/SAREF4CITY), ενώ όπου δεν ήταν δυνατή η εισαγωγή των οντοτήτων και αντιστοιχιών ως έχουν (NGSI-LD), δηλώσεις νέων οντοτήτων με βάση την ανάλυση των αντιστοιχιών που πραγματοποιήθηκε εισήχθησαν στο μοντέλο ώστε να καλυφθούν σχετικοί κάθετοι τομείς όπως η έξυπνη στάθμευση και ο έξυπνος φωτισμός που διαπραγματεύεται το έργο.

Τέλος, στο Κεφάλαιο 5 καταγράφεται η υλοποίηση των λειτουργικοτήτων διαχείρισης ομοσπονδίας και των εμπλεκόμενων στοιχείων λογισμικού των υποσυστημάτων που συνοδεύουν τον αντίστοιχο κώδικα Ανοιχτής Πηγής.

Στις δράσεις που ακολουθούν οι λειτουργικότητες για τη διαχείριση της ομοσπονδίας θα εμπλουτιστούν με νέες λειτουργικότητες που θα αναπτυχθούν στα πλαίσια των επόμενων δράσεων όπως είναι για τον έλεγχο ακεραιότητας των δικαιωμάτων του χρήστη.

# 7 Αναφορές

(n.d.). Ανάκτηση από investopedia: https://www.investopedia.com/terms/b/blockchain.asp

(n.d.). Ανάκτηση από hyperledger-fabric: https://hyperledger-fabric.readthedocs.io/en/release-2.3/ledger/ledger.html

(W3C), W. W. (2014, 2 25). *Rdf 1.1 concepts and abstract syntax.* Ανάκτηση από W3C: https://www.w3.org/TR/rdf11-concepts/

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform. Wihte Paper.*

Christoph, J. (November 2016). *Decentralized autonomous organization to automate governance.* White paper.

Christoph, R., Antonio, S. J., Pietro, T., Giuseppe, P., Gennaro, B., Giuseppe, B., . . . João, &. G. (2018). *D3.3 - Complete Federation Environment.* Zenodo. https://doi.org/10.5281/zenodo.1302897.

Emmett, J. (2019, 7 3). *Conviction Voting: A Novel Continuous Decision Making Alternative to Governance.* Ανάκτηση από Medium: https://medium.com/giveth/conviction-voting-a-novelcontinuous-decision-making-alternative-to-governance-aa746cfb9475

Faqir, E., Youssef, Arroyo, J., & Hassan, S. (2020). *An overview of decentralized autonomous organizations on the blockchain.*

Faqir-Rhazoui, Youssef, Arroyo, J., & Hassan, S. (2021). A comparative analysis of the platforms for decentralized autonomous organizations in the Ethereum blockchain. *Journal of Internet Services and Applications 12.1*, 1-20.

Faqir-Rhazoui, Youssef, Arroyo, J., & Hassan, S. (2021). *A scalable voting system: Validation of holographic consensus in daostack.*

Grossman, S., & Hart, O. (1992). An Analysis of the Principal-Agent Problem. Στο *Foundations of Insurance Economics. Huebner International Series on Risk, Insurance and Economic Security, vol 14.* Springer, Dordrecht. https://doi.org/10.1007/978-94-015-7957-5_16.

Hardin, G. (1968). The tragedy of the commons: the population problem has no technical solution; it requires a fundamental extension in morality. *Science*, 1243-1248.

Jacoby, M., Garcia, J., Paradell, A., Santis, L. d., Pardi, M., Kreiner, K., . . . Žarko, I. P. (2017). *D2.4 - Revised Semantics for IoT and Cloud Resources.* Zenodo. https://doi.org/10.5.

Jentzsch, C. (2016). *Decentralized autonomous organization to automate governance. White Paper.*

Michael Jacoby, J. G. (2017). *D2.4 - Revised Semantics for IoT and Cloud Resources.* Zenodo. https://doi.org/10.5.

Singh, M., & Kim, S. (2019). Chapter Four - Blockchain technology for decentralized autonomous organizations. Στο *Advances in Computers.* Elsevier https://doi.org/10.1016/bs.adcom.2019.06.001.

Skočir, P., Žarko, I. P., Katsaros, K., Pardi, M., Fraia, M. D., Kovacs, G., . . . Caldarola, D. (2017). *D1.4 - Final Report on System Requirements and Architecture.* Zenodo. https://doi.org/10.5281/zenodo.830156.

Soleimani, A. e. (2019). *The Moloch DAO.*

Staff, C. (2022, 3 10). *DeFi Governance in Action.* Ανάκτηση από https://www.gemini.com/cryptopedia/defi-solutions-decentralized-governance-meaning

W3C. (2011, 03 28). *Notation3 (N3): A readable RDF syntax.* Ανάκτηση από W3C: https://www.w3.org/TeamSubmission/n3/

W3C. (2014, 2 25). *RDF 1.1 N-Triples.* Ανάκτηση από W3C: https://www.w3.org/TR/n-triples/

W3C. (2014, 2 25). *RDF 1.1 Turtle.* Ανάκτηση από W3C: https://www.w3.org/TR/turtle/

W3C. (2014, 2 25). *RDF 1.1 XML Syntax.* Ανάκτηση από W3C: https://www.w3.org/TR/rdf-syntax-grammar/

W3C. (2014, 02 25). *RDF Schema 1.1.* Ανάκτηση από W3C: https://www.w3.org/TR/rdf-schema/

W3C. (2015, 3 17). *RDFa 1.1 Primer - Third Edition.* Ανάκτηση από W3C: https://www.w3.org/TR/rdfa-primer/

W3C. (2017, 12 8). *Semantic Sensor Network Ontology.* Ανάκτηση από W3C: https://www.w3.org/TR/vocab-ssn/

W3C. (2020, 07 16). *JSON-LD 1.1.* Ανάκτηση από W3C: https://www.w3.org/TR/json-ld/

Wang, S. e. (2019). Decentralized autonomous organizations: Concept, model, and applications. *IEEE Transactions on Computational Social Systems 6.5*, 870-878.

Yuste, J. I., Sanchez, J. A., Garcia, J., Tedeschi, P., Piro, G., Boggia, G., . . . Skocir, P. (2018). *D3.2 - Resource Trading, Security and Federation* . Zenodo. https://doi.org/10.5281/zenodo.1135955.

Zetzsche, A., D., Arner, D. W., & Buckley, R. P. (2020). Decentralized finance. *Journal of Financial Regulation 6.2*, 172-203.