

IoTFeds

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

Π1.4 - Ενδιάμεση Έκδοση Ολοκληρωμένου Συστήματος

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	T2EΔK-02178
Ενότητα Εργασίας	EE1 (Δ1.4)
Υπεύθυνος Ενότητας	ICOM
Υπεύθυνος Παραδοτέου	ICOM
Έκδοση Παραδοτέου	1.0
Ημερομηνία	31 Δεκεμβρίου, 2022

Συντονιστής Παραδοτέου

Απόστολος Νάσιου, ICOM

Έλια Πιέτρη, ICOM

Κωνσταντίνος Τσάκαλος, ICOM

Μάριος Χαραλαμπίδης, ICOM

Υπεύθυνοι Συγγραφής

Ευάγγελος Αθανασάκης, ΕΚΕΤΑ

Στέλιος Γκούσκος, TERRACOM

Σοφία Πολυμένη, ΕΚΕΤΑ

Ζήσης Σακελλαρίου, ΕΚΕΤΑ

Γεώργιος Σπανός, ΕΚΕΤΑ

Ακρωνύμια

Ακρωνύμιο	Ορισμός
API	Application Programming Interface
UI	User Interface
IoT	Internet of Things
BaaS	Blockchain as a Service
RAP	Resource Access Proxy
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
JSON	JavaScript Object Notation
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 περιγράφει την ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας που είναι αποτέλεσμα της ενσωμάτωσης των υπηρεσιών των στοιχείων λογισμικού που αναπτύχθηκαν στα πλαίσια της δράσης Δ2.1 «Διαχείριση Ομοσπονδίας IoT» και Δ2.2 «Σημασιολογική Διαλειτουργικότητα Πλατφορμών IoT».

Το παραδοτέο τεκμηριώνει και συνοδεύει τον κώδικα Ανοιχτής Πηγής του λογισμικού της πρώτης έκδοσης του ολοκληρωμένου συστήματος. Η ενδιάμεση αυτή έκδοση (πρώτη έκδοση) παρέχει τις βασικές λειτουργικότητες για τη διαχείριση της ομοσπονδίας, ενσωματώνοντας τις σχετικές υπηρεσίες των υποσυστημάτων symbIoTe και BaaS (Blockchain as a Service) που συνθέτουν την IoTFeds πλατφόρμα. Τα στοιχεία λογισμικού του symbIoTe για τη διαχείριση ομοσπονδιών, χρηστών και πλατφορμών (Administration) καθώς και για τη διαχείριση IoT πόρων (symbIoTeAPI) εξυπηρετούν τα αιτήματα του χρήστη εκκινώντας τις απαραίτητες ροές επικοινωνίας στο σύστημα με άλλα εμπλεκόμενα στοιχεία του symbIoTe καθώς και με το BaaS (στοιχεία λογισμικού για διαχείριση χρηστών, διαχείριση ομοσπονδιών και διαχείριση ψηφοφορίας στο BaaS).

Στο κείμενο αυτό περιγράφονται οι διεπαφές/APIs (Application Programming Interfaces) που απαιτούνται για την υποστήριξη της διαχείρισης της IoT ομοσπονδίας και την ενσωμάτωση των διαφορετικών στοιχείων λογισμικού της πλατφόρμας IoTFeds που αναπτύχθηκαν στις δράσεις Δ2.1 και Δ2.2 στο ολοκληρωμένο σύστημα. Η έκδοση αυτή θα αποτελέσει τη βάση, στην οποία θα βασιστούν οι μηχανισμοί των επόμενων δράσεων για την ανάπτυξη της τελικής έκδοσης του συστήματος της IoTFeds πλατφόρμας που θα περιγραφεί στο παραδοτέο Π1.5 «Τελική Έκδοση Ολοκληρωμένου Συστήματος».

Πίνακας Περιεχομένων

1	Αντικείμενο και Δομή του Παραδοτέου.....	9
2	Ενσωμάτωση του ολοκληρωμένου συστήματος IoTFeds.....	10
2.1	Αρχιτεκτονική του συστήματος για την διαχείριση της ομοσπονδίας	10
2.2	Κατασκευή και ανάπτυξη του έργου.....	13
2.3	Διαδικασία για την εγκατάσταση του λογισμικού του ενσωματωμένου συστήματος 14	
2.3.1	Εξωτερικά εργαλεία (External tools used)	14
2.3.2	symbIoTe.....	15
2.3.2.1	symbIoTe Core deployment	15
2.3.2.1.1	Δημιουργία φακέλου διαμόρφωσης	15
2.3.2.1.2	Δημιουργία πιστοποιητικού Core AAM certificate.....	15
2.3.2.1.3	Δημιουργία φακέλων για την ανάπτυξη (deployment) του symbIoTe core 16	
2.3.2.1.4	Δημιουργία docker volumes για την ανάπτυξη (deployment) σε Docker περιβάλλον	17
2.3.2.1.5	Διαμόρφωση μικρο-υπηρεσίας NGINX για την ανάπτυξη (deployment) σε Docker περιβάλλον.....	17
2.3.2.1.6	Ανάπτυξη σε περιβάλλον Docker Swarm	18
2.3.2.1.7	Ανάπτυξη σε περιβάλλον Kubernetes	18
2.3.2.2	symbIoTe Cloud deployment	19
2.3.2.2.1	Εγγραφή του ιδιοκτήτη της πλατφόρμας.....	19
2.3.2.2.2	Εγγραφή της IoT πλατφόρμας.....	19
2.3.2.2.3	Διαμόρφωση των απαιτούμενων αρχείων παραμέτρων	19
2.3.2.2.4	Δημιουργία φακέλου για την εγκατάσταση του symbIoTe Cloud με Docker Swarm.....	20
2.3.2.2.5	Δημιουργία φακέλου για την εγκατάσταση του symbIoTe Cloud σε Kubernetes.....	23
2.3.2.2.6	Δημιουργία RAP plugin	23
2.3.3	Blockchain as a Service (BaaS).....	27
3	Σχεδιασμός και δοκιμή του ολοκληρωμένου συστήματος IoTFeds – Ενδιάμεση έκδοση 29	
3.1	Υποσύστημα symbIoTe.....	29
3.1.1	Στοιχείο λογισμικού για τη διαχείριση ομοσπονδιών, χρηστών και πλατφορμών (Administration)	29
3.1.1.1	Περιγραφή	29
3.1.1.2	Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	30
3.1.1.3	Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	30
3.1.1.3.1	Διεπαφές για τη διαχείριση IoT ομοσπονδίας.....	30
3.1.1.4	Δοκιμή και επαλήθευση (Test cases)	31
3.1.2	Στοιχείο λογισμικού για τη διαχείριση IoT πόρων (symbIoTe API).....	45
3.1.2.1	Περιγραφή	45

3.1.2.2	Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	45
3.1.2.3	Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	45
3.1.2.4	Δοκιμή και επαλήθευση (Test cases)	46
3.2	Υποσύστημα BaaS	51
3.2.1	Στοιχείο λογισμικού για τη διαχείριση χρηστών	51
3.2.1.1	Περιγραφή	51
3.2.1.2	Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	51
3.2.1.3	Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	52
3.2.1.4	Δοκιμή και επαλήθευση (Test cases)	52
3.2.2	Στοιχείο λογισμικού για τη διαχείριση ομοσπονδιών.....	57
3.2.2.1	Περιγραφή	58
3.2.2.2	Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	58
3.2.2.3	Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος	58
3.2.2.4	Δοκιμή και επαλήθευση (Test cases)	59
4	Λειτουργική δοκιμή και επίδειξη συστήματος	69
5	Συμπεράσματα.....	96
6	Αναφορές.....	97

Εικόνα 1: Αρχιτεκτονική υποσυστήματος symbIoTe στο IoTFeds.	10
Εικόνα 2: Διαχείριση ομοσπονδιών στο BaaS.	13
Εικόνα 3: Πληροφορίες εγγεγραμμένης IoT πλατφόρμας.	19
Εικόνα 4: Διαμόρφωση παραμέτρων πλατφόρμας.	20
Εικόνα 5: Δομή φακέλου διαμόρφωσης και ανάπτυξης του symbIoTe Cloud σε Docker Swarm.	22
Εικόνα 6 Παράμετροι στο σώμα του ελέγχου BaaSAPI-1.A.	53
Εικόνα 7 Παράμετροι στο σώμα του ελέγχου BaaSAPI-2.A.	53
Εικόνα 8 Παράμετροι στο σώμα του ελέγχου BaaSAPI-3.A.	53
Εικόνα 9 Επιστροφή πληροφοριών χρήστη απο το BaaS.	54
Εικόνα 10 Επιστροφή πληροφοριών όλων των χρηστών από το BaaS.	54
Εικόνα 11 Παράμετροι στο σώμα του ελέγχου BaaSAPI-6.A.	54
Εικόνα 12 Παράμετροι στο σώμα του ελέγχου BaaSAPI-6.B.	54
Εικόνα 13 Παράμετροι στο σώμα του ελέγχου BaaSAPI-7.A.	55
Εικόνα 14 Παράμετροι στο σώμα του ελέγχου BaaSAPI-7.B.	55
Εικόνα 15 Παράμετροι στο σώμα του ελέγχου BaaSAPI-8.A.	55
Εικόνα 16 Παράμετροι στο σώμα του ελέγχου BaaSAPI-9.A.	56
Εικόνα 17 Παράμετροι στο σώμα του ελέγχου BaaSAPI-9.B.	56
Εικόνα 18 Παράμετροι στο σώμα του ελέγχου BaaSAPI-10.A.	56
Εικόνα 19 Αντικείμενο διαγραμμένου χρήστη στο BaaS.	56
Εικόνα 20 Παράμετροι στο σώμα του ελέγχου BaaSAPI-10.B.	57
Εικόνα 21: Παράμετροι στο σώμα του ελέγχου BaaSAPI-11.A.	59
Εικόνα 22: Επιστροφή πληροφοριών ομοσπονδίας απο το BaaS.	60
Εικόνα 23: Επιστροφή πληροφοριών όλων των ομοσπονδιών απο το BaaS.	60
Εικόνα 24 Παράμετροι στο σώμα του ελέγχου BaaSAPI-14.A.	61
Εικόνα 25: Παράμετροι στο σώμα του ελέγχου BaaSAPI-14.Γ.	61
Εικόνα 26: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.A.	62
Εικόνα 27 Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.B.	62
Εικόνα 28: Αντικείμενο που επιστρέφεται απο τον έλεγχο BaaSAPI-16.B.	62
Εικόνα 29: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.Γ.	63
Εικόνα 30: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.A.	63
Εικόνα 31: Αντικείμενο που επιστρέφεται απο τον έλεγχο BaaSAPI-17.A.	63
Εικόνα 32: Παράμετροι στο σώμα του ελέγχου BaaSAPI-17.B.	64
Εικόνα 33: Αρχικό σενάριο κανόνων της ομοσπονδίας που χρησιμοποιείται για τους ελέγχους της διεπαφής BaaSAPI-18.	64
Εικόνα 34: Παράμετροι στο σώμα του ελέγχου BaaSAPI-18.A.	65
Εικόνα 35: Αναμενόμενο αποτέλεσμα επιστροφής του ελέγχου BaaSAPI-18.B.	66
Εικόνα 36: Αντικείμενο που περιγράφει την ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B.	67
Εικόνα 37: Παράμετροι στο σώμα του ελέγχου BaaSAPI-20.A για καταχώρηση ψήφου στην ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B.	67

Πίνακας 1:Διεπαφές στοιχείου symbIoTe Administration για διαχείριση ομοσπονδίας.	30
Πίνακας 2:Περίπτωσης δοκιμής στοιχείου symbIoTe's Administration για διαχείριση ομοσπονδίας.	44
Πίνακας 3:Διεπαφές στοιχείου Administration για τη διαχείριση IoT πόρων.	46
Πίνακας 4:Περίπτωσης δοκιμής στοιχείου symbIoTe-API.	50
Πίνακας 5: Διεπαφές στοιχείου BaaS για την διαχείριση χρηστών.	52
Πίνακας 6:Περίπτωσης δοκιμής στοιχείου BaaS για την διαχείριση χρηστών.	57
Πίνακας 7:Διεπαφές στοιχείου BaaS.....	59
Πίνακας 8:Περίπτωσης δοκιμής στοιχείου BaaS για τη διαχείριση ομοσπονδιών.....	68

1 Αντικείμενο και Δομή του Παραδοτέου

Στο παραδοτέο αυτό περιγράφεται η ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας. Η έκδοση αυτή περιλαμβάνει τους μηχανισμούς για τη διαχείριση των IoT ομοσπονδιών και την επίτευξη της σημασιολογικής διαλειτουργικότητας που αναπτύχθηκαν στις δράσεις Δ2.1 και Δ2.2 αντίστοιχα επεκτείνοντας το ενδιάμεσο λογισμικό symbIoTe και ενσωματώνοντας μηχανισμούς blockchain. Οι λειτουργικότητες που καλύπτει η ενδιάμεση έκδοση του ολοκληρωμένου συστήματος συνοψίζονται παρακάτω:

- Εγγραφή χρήστη.
- Σύνδεση χρήστη.
- Διαγραφή χρήστη.
- Εγγραφή IoT πλατφόρμας.
- Διαγραφή IoT πλατφόρμας.
- Εγκαθίδρυση ομοσπονδίας.
- Προσθήκη μέλους ομοσπονδίας.
- Απομάκρυνση μελών από ομοσπονδία.
- Επιλογή και τροποποίηση κανόνων ομοσπονδίας.
- Διαγραφή ομοσπονδίας και των μελών της.
- Εγγραφή ή τροποποίηση IoT συσκευών/πόρων με βάση το μοντέλο πληροφορίας IoT-Feds για την Έξυπνη Πόλη.
- Διαμοιρασμός των IoT συσκευών/πόρων σε μία ομοσπονδία.
- Διαγραφή μίας IoT συσκευής/πόρου από μία ομοσπονδία ή το σύστημα.

Το παραδοτέο τεκμηριώνει την ενσωμάτωση των υπηρεσιών των στοιχείων λογισμικού που αναπτύχθηκαν ή και επεκτάθηκαν στα πλαίσια του έργου για την παροχή των λειτουργικοτήτων αυτών στο ολοκληρωμένο σύστημα.

Το κείμενο στο παραδοτέο αυτό ακολουθεί την παρακάτω δομή:

Στο κεφάλαιο 2 περιγράφεται η μεθοδολογία που ακολουθείται για την ενσωμάτωση του ολοκληρωμένου συστήματος, παρουσιάζοντας την αρχιτεκτονική του συστήματος και την διασύνδεση των υποσυστημάτων που το αποτελούν, την μεθοδολογία ανάπτυξης έργου και τα βήματα εγκατάστασης συστήματος, τα εξωτερικά εργαλεία που χρησιμοποιούνται και τη μεθοδολογία δοκιμής του ενσωματωμένου συστήματος.

Στο κεφάλαιο 3 παρουσιάζονται τα στοιχεία λογισμικού των υποσυστημάτων που συμμετέχουν στους νέους μηχανισμούς και την ενδιάμεση έκδοση του ολοκληρωμένου συστήματος καθώς και τις σχετικές διεπαφές που υλοποιήθηκαν και συνοδεύουν τον κώδικα Ανοιχτής Πηγής.

Στο κεφάλαιο 4 γίνεται η λειτουργική δοκιμή και επίδειξη της χρήσης του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας για την υποστήριξη της διαχείρισης της IoT ομοσπονδίας και την επίτευξη της διαλειτουργικότητας σε μία IoT ομοσπονδία.

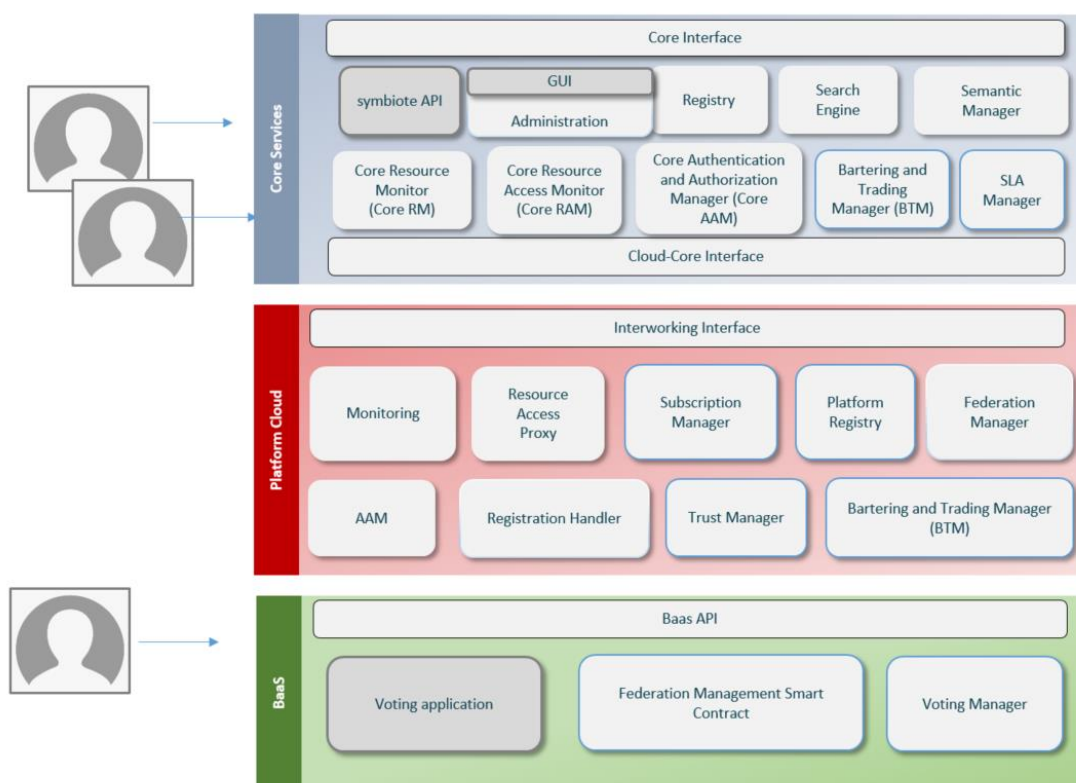
Τέλος, το κεφάλαιο 5 αναλύει τα συμπεράσματα και ολοκληρώνει το παραδοτέο.

2 Ενσωμάτωση του ολοκληρωμένου συστήματος IoTFeds

Στην ενότητα αυτή περιγράφεται η μεθοδολογία για την ενσωμάτωση της ενδιάμεσης έκδοσης του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας. Αρχικά συνοψίζεται η αρχιτεκτονική του συστήματος για την διαχείριση της IoT ομοσπονδίας και τη σημασιολογική διαλειτουργικότητα. Στη συνέχεια περιγράφονται οι οδηγίες και δίνονται οι πληροφορίες για την κατασκευή και ανάπτυξη του έργου (project build and deployment), τα εξωτερικά εργαλεία που χρησιμοποιούνται στο σύστημα και τέλος περιγράφεται η μεθοδολογία δοκιμής που ακολουθείται για την διασφάλιση της ομαλής λειτουργίας του συστήματος.

2.1 Αρχιτεκτονική του συστήματος για την διαχείριση της ομοσπονδίας

Το σύνολο των επιμέρους στοιχείων που εμπλέκονται στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος φαίνεται στην Εικόνα 1.



Εικόνα 1: Αρχιτεκτονική υποσυστήματος symbIoTe στο IoTFeds.

Πιο συγκεκριμένα, η διαχείριση των χρηστών στην IoTFeds πλατφόρμα όπως η εγγραφή ενός χρήστη στο σύστημα γίνεται μέσω της γραφικής διεπαφής Administration GUI. Επιπλέον, ένας χρήστης (οργανισμός) που θέλει να δημιουργήσει ή συμμετάσχει σε μία IoT ομοσπονδία πρέπει να εγγράψει την/τις IoT πλατφόρμα/-ες του στο σύστημα. Η διαχείριση των πλατφορμών γίνεται μέσω της γραφικής διεπαφής. Μετά την εγγραφή της πλατφόρμας στο σύστημα, ο χρήστης μπορεί να δημιουργήσει μία νέα ομοσπονδία, να ζητήσει να γίνει μέλος σε υπάρχουσες ομοσπονδίες ή κάποιο μέλος σε υπάρχουσα ομοσπονδία να τον καλέσει να συμμετάσχει στην ομοσπονδία αυτή. Συνεπώς όλα τα αιτήματα για τη διαχείριση της ομοσπονδίας γίνονται μέσω της γραφικής διεπαφής. Τα αιτήματα του χρήστη προωθούνται από το στοιχείο λογισμικού Administration του symbIoTe στα εμπλεκόμενα στοιχεία λογισμικού του symbIoTe όπως ο AAM (Authentication and Authorization Manager) για τον έλεγχο ταυτότητας και αυθεντικοποίησης και ο FM (Federation Manager) για τη διαχείριση των ομοσπονδιών, ενώ ενημερώνεται το υποσύστημα BaaS για την καταγραφή και τον έλεγχο ακεραιότητας πληροφορίας. Το υποσύστημα BaaS διεκπεραιώνει επίσης τις διαδικασίες ψηφοφορίας και την εκτέλεση έξυπνων συμβολαίων που εμπλέκονται στα σχετικά αιτήματα όπως η συμμετοχή νέου

μέλους σε υπάρχουσα ομοσπονδία ή η απομάκρυνση υπάρχοντος μέλους από μία ομοσπονδία. Η ψηφοφορία από κάθε χρήστη γίνεται μέσα από το στοιχείο Voting Application του BaaS. Τέλος, η διαχείριση και ο διαμοιρασμός των IoT πόρων/συσκευών και των δεδομένων τους γίνονται μέσω του RestFul API του στοιχείου λογισμικού symbIoTe API του symbIoTe. Το στοιχείο αυτό αναπτύχθηκε στα πλαίσια του έργου για να διευκολύνει την αλληλεπίδραση ενός χρήστη/πελάτη (όπως είναι μία εφαρμογή ή ένας πάροχος δεδομένων) με τις υπηρεσίες του symbIoTe από όπου εκκινούνται οι απαραίτητες ροές επικοινωνίας με τα εμπλεκόμενα στοιχεία του symbIoTe όπως με τον RH (Registration Handler) για την εγγραφή των πόρων στο σύστημα πόρου και τον RAP (Resource Access Proxy) για την πρόσβαση στα δεδομένα, ενώ η επικοινωνία του symbIoTe με το υποσύστημα BaaS πραγματοποιείται μέσα από το BaaS API. Το υποσύστημα symbIoTe αποτελείται από ένα σύνολο κεντροποιημένων υπηρεσιών (symbIoTe Core services) και ένα σύνολο κατανεμημένων υπηρεσιών στη μεριά της εκάστοτε πλατφόρμας/οργανισμού (symbIoTe Cloud services) για την ενοποίηση και επικοινωνία με αυτές τις IoT πλατφόρμες και την πρόσβαση στα δεδομένα παρατήρησης (μετρήσεις), τις υπηρεσίες και τους ενεργοποιητές που αυτές παρέχουν με ασφαλή και ενοποιημένο τρόπο. Οι κεντροποιημένες υπηρεσίες του symbIoTe είναι οι ακόλουθες:

- *symbIoTeAPI*: Το στοιχείο αυτό αναπτύχθηκε στα πλαίσια του έργου για να διευκολύνει την αλληλεπίδραση ενός χρήστη/πελάτη (όπως είναι μία εφαρμογή ή ένας πάροχος δεδομένων) με τις υπηρεσίες του symbIoTe από όπου εκκινούνται οι απαραίτητες ροές επικοινωνίας με τα εμπλεκόμενα στοιχεία του symbIoTe όπως με τον RH (Registration Handler) για την εγγραφή των πόρων στο σύστημα πόρου και τον RAP (Resource Access Proxy) για την πρόσβαση στα δεδομένα.
- *Administration*. Παρέχει τη γραφική διεπαφή χρήστη και την εξυπηρέτηση διαχειριστικών ενεργειών από τους χρήστες στο σύστημα σχετικά με την διαχείριση των χρηστών, πλατφορμών και ομοσπονδιών. Το στοιχείο αυτό έχει επεκταθεί στα πλαίσια του έργου για να υποστηρίξει τη διαχείριση χρηστών (οργανισμών), τη διαχείριση ομοσπονδιών και των κανόνων τους στην IoTFeds πλατφόρμα με χρήση έξυπνων συμβολαίων και την εκτέλεση ψηφοφοριών ενσωματώνοντας τις υπηρεσίες του BaaS.
- *Registry*. Αποθηκεύει δεδομένα σχετικά με τους καταχωρημένους πόρους στο σύστημα με βάση το μοντέλο πληροφορίας σε συνεργασία με τον Semantic Manager (SM).
- *Search Engine*. Εξυπηρετεί την αναζήτηση πόρων που καλύπτουν τα κριτήρια στα σχετικά ερωτήματα και ακολουθώντας τους περιορισμούς ασφάλειας του συστήματος. Το στοιχείο αυτό θα επεκταθεί στα πλαίσια του έργου στις επόμενες δράσεις. Οι επεκτάσεις θα συμπεριληφθούν στην τελική έκδοση του ολοκληρωμένου συστήματος στο Π1.5.
- *Semantic Manager (SM)*. Διαχειρίζεται τα μοντέλα πληροφορίας που χρησιμοποιούνται και επαληθεύει τη συμμόρφωση των πόρων με το μοντέλο πληροφορίας που χρησιμοποιείται. Το στοιχείο λογισμικού SM έχει επεκταθεί με προσθήκες για την υποστήριξη της σημασιολογικής διαλειτουργικότητας στα πλαίσια του έργου για να υποστηρίξει το μοντέλο πληροφορίας του IoTFeds για την Έξυπνη Πόλη.
- *Core Authentication and Authorization Manager (Core AAM)*. Εφαρμόζει 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 να διατηρήσει πληροφορίες δημοτικότητας πόρων.
- *Core Bartering and Trading Manager (Core BTM)*. Περιλαμβάνει όλες τις λειτουργίες ανταλλαγής και συναλλαγών που πρέπει να συγκεντρωθούν και να αναπτυχθούν εντός

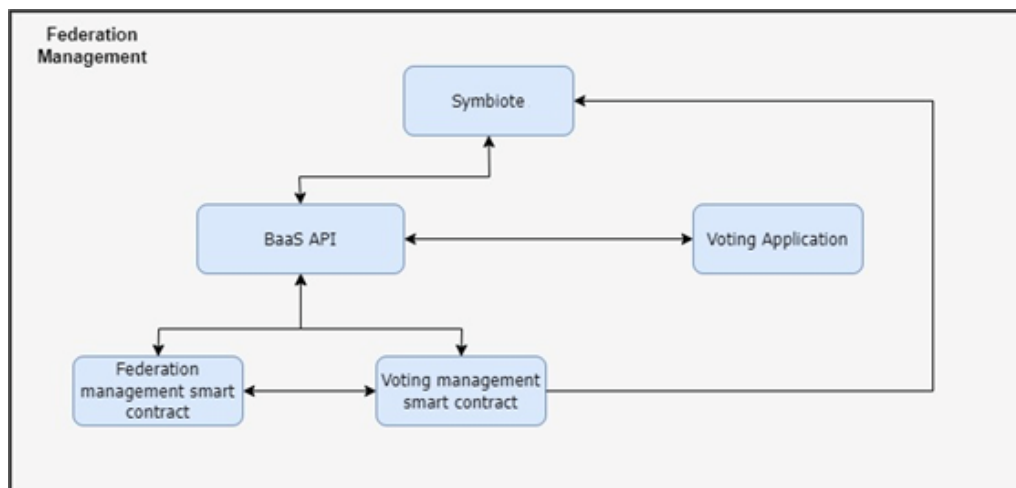
μίας εφαρμογής, τη διαχείριση κουπονιών και στατιστικών. Το στοιχείο αυτό θα επεκταθεί στα πλαίσια του έργου στις επόμενες δράσεις. Οι επεκτάσεις θα συμπεριληφθούν στην τελική έκδοση του ολοκληρωμένου συστήματος στο Π1.5.

- *Service Level Agreement Manager (SLA Manager)*. Διαχειρίζεται τον κύκλο ζωής των συμφωνιών επιπέδου υπηρεσιών (SLAs) για τις ομοσπονδίες των IoT πλατφορμών. Αξιολογεί, 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 της πλατφόρμας σε συνεργασία με τις κεντριοποιημένες 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.
- *Authentication and Authorization Manager (Platform AAM)*. Διαχειρίζεται την δυνατότητα πρόσβασης στους πόρους της πλατφόρμας και των συνεργατικών πλατφορμών.
- *Monitoring*. Είναι υπεύθυνο για τη συλλογή μετρήσεων που σχετίζονται με την κατάσταση των καταχωρημένων πόρων και την πρόσβαση σε αυτούς. Το στοιχείο αυτό θα επεκταθεί στα πλαίσια του έργου στις επόμενες δράσεις. Οι επεκτάσεις θα συμπεριληφθούν στην τελική έκδοση του ολοκληρωμένου συστήματος στο Π1.5.
- *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. Το στοιχείο αυτό θα επεκταθεί στα πλαίσια του έργου στις επόμενες δράσεις. Οι επεκτάσεις θα συμπεριληφθούν στην τελική έκδοση του ολοκληρωμένου συστήματος στο Π1.5.
- *Bartering and Trading Manager (BTM)*. Διαχειρίζεται τους αποκεντρωμένους μηχανισμούς ανταλλαγής και εμπορίας πόρων μεταξύ των ομόσπονδων πλατφορμών. Το στοιχείο αυτό θα επεκταθεί στα πλαίσια του έργου στις επόμενες δράσεις. Οι επεκτάσεις θα συμπεριληφθούν στην τελική έκδοση του ολοκληρωμένου συστήματος στο Π1.5.

Τέλος, στην Εικόνα 2 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 2: Διαχείριση ομοσπονδιών στο BaaS.

Συνοψίζοντας, οι βασικές υπηρεσίες που περιλαμβάνονται στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας είναι οι παρακάτω:

- Εγγραφή χρήστη
- Σύνδεση χρήστη
- Διαγραφή χρήστη
- Εγγραφή IoT πλατφόρμας
- Διαγραφή IoT πλατφόρμας
- Εγκαθίδρυση ομοσπονδίας
- Προσθήκη μέλους ομοσπονδίας
- Απομάκρυνση μελών από ομοσπονδία
- Επιλογή και τροποποίηση κανόνων ομοσπονδίας
- Διαγραφή ομοσπονδίας και των μελών της
- Εγγραφή ή τροποποίηση IoT συσκευών/πόρων με βάση το μοντέλο πληροφορίας IoT-Feds για την Έξυπνη Πόλη
- Διαμοιρασμός των IoT συσκευών/πόρων σε μία ομοσπονδία
- Διαγραφή μίας IoT συσκευής/πόρου από μία ομοσπονδία ή το σύστημα

Οι παραπάνω λειτουργικότητες θα αποτελέσουν και τη βάση για την ανάπτυξη των μηχανισμών και τα αντίστοιχα τελικά σημεία (endpoints) στις επόμενες δράσεις όπως η Δ2.3 «Διαφήμιση και Ανακάλυψη Ομόσπονδων Πόρων IoT» και Δ2.4 «Ασφαλής Πρόσβαση σε Δεδομένα IoT».

2.2 Κατασκευή και ανάπτυξη του έργου

Κατά την υλοποίηση, η ομάδα ανάπτυξης του έργου ακολουθεί μεθοδολογία συνεχούς ολοκλήρωσης (Continuous Integration) για την διευκόλυνση της ανάπτυξης και ενσωμάτωσης των στοιχείων λογισμικού στο ολοκληρωμένο σύστημα της IoTFeds πλατφόρμας. Η ομάδα του έργου χρησιμοποιεί τα εργαλεία και την υποδομή που ορίστηκαν και αναπτύχθηκαν για τους σκοπούς του έργου IoTFeds, όπως περιγράφονται παρακάτω:

- Το εργαλείο Git χρησιμοποιείται ως σύστημα ελέγχου έκδοσης (version control system), επιτρέποντας στους προγραμματιστές της ομάδας ανάπτυξης να συνεργάζονται και να δουλεύουν παράλληλα στον πηγαίο κώδικα.
- Ο πηγαίος κώδικας είναι προσβάσιμος και διαθέσιμος στους προγραμματιστές της ομάδας σε ένα ιδιωτικό αποθετήριο GitLab (<https://colab-repo.intracom-telecom.com/colab-projects/iotfeds>) ενώ για την άμεση διάθεση του κώδικα ανοιχτής πηγής προς τρίτους αντίγραφο του κώδικα διατίθεται στο δημόσιο (public) αποθετήριο (<https://github.com/iotfedsproject>) το οποίο ενημερώνεται από τους συμμετέχοντες στο

έργο και περιέχει οδηγίες για τη δημιουργία και την ανάπτυξη κάθε απαραίτητου στοιχείου, όπως τα αρχεία yaml που αξιοποιούνται στην ανάπτυξη και ενσωμάτωση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας.

- Επιπλέον, έχει δημιουργηθεί ένα αποθετήριο (image repository) στο DockerHub: <https://hub.docker.com/repository/docker/iotfedsproject/> για τη διαχείριση των εικόνων (container images) που ενσωματώνουν τον κώδικα κάθε στοιχείου λογισμικού, όπου είναι δυνατόν.
- Για τις δραστηριότητες της ενσωμάτωσης του ολοκληρωμένου συστήματος και τη δοκιμή του, τα υποσυστήματα έχουν αναπτυχθεί στην υποδομή του εκάστοτε φορέα του υποσυστήματος (ICOM, EKETA): το υποσύστημα symbIoTe (symbIoTe core services) είναι εγκαταστημένο σε Kubernetes cluster στο παρακάτω url: <https://symbiote-core.iot-feds.intracom-telecom.com/administration/>, ενώ το υποσύστημα BaaS είναι διαθέσιμο στο παρακάτω url: <http://160.40.50.169:8800/baas/>. Η επικοινωνία μεταξύ των υποσυστημάτων καθώς και η πρόσβαση σε αυτά από τρίτους (όπως symbIoTe cloud services IoT πλατφορμών εγγεγραμμένων στο σύστημα) γίνεται μέσω HTTP(S). Περισσότερες λεπτομέρειες δίνονται στις ενότητες 3.1 και 3.2.

Τέλος, για την εξασφάλιση της επιτυχούς διασύνδεσης των διαφορετικών στοιχείων λογισμικού μεταξύ τους και της ομαλής λειτουργίας του ενσωματωμένου συστήματος, ορίζονται και εκτελούνται υποθέσεις και σενάρια δοκιμών (test cases) για τις υπηρεσίες κάθε υποσυστήματος που αποτελεί μέρος της πλατφόρμας IoTFeds και τις λειτουργικότητες του συστήματος. Για RESTful API, εκτελούνται κατάλληλα αιτήματα HTTP(S) που καλύπτουν το πλήρες σύνολο δοκιμών για κάθε τελικό σημείο που θα δοκιμαστεί και η απάντηση που λαμβάνεται επικυρώνεται με βάση το αναμενόμενο αποτέλεσμα (απάντηση). Οι επιτυχείς δοκιμές αναφέρονται ως PASS και οι αποτυχημένες ως FAIL. Σε περίπτωση αποτυχίας οι προγραμματιστές που εμπλέκονται με το υπό δοκιμή λογισμικό ενημερώνονται για τη διόρθωση τυχόν σφαλμάτων που εντοπίστηκαν. Όταν όλες οι υποθέσεις και σενάρια δοκιμών κάθε διεπαφής ολοκληρώνονται με επιτυχία, ο πηγαίος κώδικας με τις σχετικές λειτουργικότητες περιλαμβάνεται στην νέα έκδοση του ολοκληρωμένου συστήματος. Για τις γραφικές διεπαφές ορίζονται υποθέσεις και σενάρια δοκιμών και ελέγχονται τα επιμέρους βήματα για τον έλεγχο της επιτυχούς λειτουργίας. Το αποτέλεσμα της δοκιμαστικής διαδικασίας θεωρείται αποδεκτό μετά την επιτυχή ολοκλήρωση όλων των υποθέσεων και σεναρίων δοκιμών και της επικύρωσης της αναμενόμενης συμπεριφοράς του συστήματος.

2.3 Διαδικασία για την εγκατάσταση του λογισμικού του ενσωματωμένου συστήματος

Η ενότητα αυτή περιγράφει τις πληροφορίες και τη διαδικασία που απαιτείται για την ανάπτυξη του ενσωματωμένου συστήματος. Συνοψίζοντας, όπως αναφέρθηκε στη προηγούμενη ενότητα, ο κώδικας ανοιχτής πηγής είναι διαθέσιμος προς τρίτους στο ανοιχτό αποθετήριο: <https://github.com/iotfedsproject>, ενώ ενημερώνεται συνεχώς στο ιδιωτικό αποθετήριο: <https://colab-repo.intracom-telecom.com/colab-projects/iotfeds> σε όλη τη διάρκεια της υλοποίησης. Τα στοιχεία λογισμικού των υποσυστημάτων της IoTFeds πλατφόρμας είναι ενσωματωμένα σε Docker εικόνες (images) που είναι διαθέσιμες στο DockerHub: <https://hub.docker.com/repository/docker/iotfedsproject/>.

2.3.1 Εξωτερικά εργαλεία (External tools used)

Το symbIoTe υποσύστημα έχει σχεδιαστεί ακολουθώντας την αρχιτεκτονική μικρο-υπηρεσιών και έχει αναπτυχθεί ώστε να τρέχει σε περιβάλλοντα εικονικοποίησης (Virtualization) σε επίπεδο λειτουργικού συστήματος (containerized). Μπορεί να εγκατασταθεί σε περιβάλλον Docker (Docker 20.10.x, Docker compose 1.29.x) ή σε περιβάλλον Kubernetes. Το υποσύστημα symbIoTe επιπλέον χρησιμοποιεί βάση δεδομένων MongoDB (3.6+) για την αποθήκευση και ανάκτηση των δεδομένων από τα στοιχεία λογισμικού που το αποτελούν και RabbitMQ (3.6+) για την ασύγχρονη επικοινωνία μεταξύ των στοιχείων λογισμικού (όπως μεταξύ των υπηρεσιών

του symbIoTe Core ή του symbIoTe Cloud). Τέλος, η μικρο-υπηρεσία NGINX (1.12+) χρησιμοποιείται για να επιτρέψει την πρόσβαση στις υπηρεσίες των στοιχείων λογισμικού του symbIoTe από έξω.

Για την υλοποίηση του BaaS, εκτός από την τεχνολογία του blockchain Hyperledger Fabric (HLF) στην οποία βασίζεται, είναι απαραίτητη και μια διεπαφή για την επικοινωνία του blockchain με τα υπόλοιπα στοιχεία λογισμικού (components) του συστήματος. Όλες οι υπηρεσίες (services) που αποτελούν τον ενδιάμεσο κρίκο μεταξύ του ολοκληρωμένου συστήματος και του blockchain, φιλοξενούνται σε ένα RESTful API, που έχει υλοποιηθεί με χρήση του Node.js¹, ένα περιβάλλον ανοιχτού κώδικα για server (open source server environment) που επιτρέπει την εκτέλεση κώδικα JavaScript σε server. Οι υπηρεσίες παρέχονται ως τελικά σημεία (endpoints) του API, γραμμένα σε JavaScript. Παρέχονται ως HTTP requests ώστε να καλούνται από τα components του symbIoTe, εκτός από την διαδικασία της ψηφοφορίας, η οποία παρέχεται με δική της διεπαφή (front-end) από την οποία εκκινούν κλήσεις κατευθείαν στο BaaS. Κάθε υπηρεσία (endpoint) είναι υπεύθυνη για την σύνδεση και επικοινωνία με το blockchain component, καθώς και την κλήση των απαραίτητων Smart Contracts (SCs). Τα αποτελέσματα από την εκτέλεση των SCs επιστρέφονται μέσω των υπηρεσιών και του API στα κατάλληλα components κάθε φορά. Ολόκληρη, λοιπόν, η διεπαφή του API αποτελεί ένα εξωτερικό εργαλείο απαραίτητο για την ορθή ανταλλαγή πληροφοριών του ολοκληρωμένου συστήματος του IoTFeds.

2.3.2 symbIoTe

Το υποσύστημα symbIoTe βασίζεται σε αρχιτεκτονική μικρουπηρεσιών- (micro-service) και τα στοιχεία λογισμικού που το απαρτίζουν είναι ενσωματωμένα σε Docker² περιβάλλον. Η εγκατάσταση του υποσυστήματος βασίζεται σε περιβάλλον Kubernetes³ ή Docker Swarm⁴. Τα Docker images κάθε μικρο-υπηρεσίας είναι διαθέσιμα στο DockerHub⁵ καθώς και οι οδηγίες για την εκτέλεση και διαμόρφωση τους, οι οποίες περιγράφονται παρακάτω για Ubuntu Linux λειτουργικό σύστημα.

2.3.2.1 symbIoTe Core deployment

2.3.2.1.1 Δημιουργία φακέλου διαμόρφωσης

- Ο χρήστης δημιουργεί το φάκελο π.χ. symbiote-core στο μηχάνημα που θα χρησιμοποιηθεί: `mkdir symbiote-core`.
- Μεσα στον φάκελο *symbiote-core* πρέπει να δημιουργηθεί ένας φάκελος *configuration*:
`cd symbiote-core`
`mkdir configuration`

2.3.2.1.2 Δημιουργία πιστοποιητικού Core AAM certificate

Αρχικά πρέπει να δημιουργηθεί ένας φάκελος αποθήκευσης κλειδιών τύπου PKCS12 που να περιέχει ένα πιστοποιητικό με τα παρακάτω χαρακτηριστικά:

- * αυτουπογεγραμμένο (self signed).
- * με ενεργοποιημένη την ιδιότητα CA (Certificate Authority).
- * με τις ακόλουθες παραμέτρους κρυπτογράφησης:
 - SIGNATURE_ALGORITHM=SHA256 με το ECDSA
 - CURVE_NAME=secp256r1
 - KEY_PAIR_GEN_ALGORITHM=ECDSA
- * με την τιμή CN (Common Name) που έχει οριστεί σύμφωνα με το πεδίο CORE_AAM_INSTANCE_ID (τρέχουσα τιμή SymbIoTe_Core_AAM) και ορίζεται στο αρχείο AAMConstants.java.
- * με το όνομα καταχώρησης πιστοποιητικού "SymbIoTe_Core_AAM".

¹ <https://www.codecademy.com/article/what-is-node>

² <https://www.docker.com/>

³ <https://kubernetes.io/>

⁴ <https://docs.docker.com/engine/swarm/>

⁵ <https://hub.docker.com/>

Αυτός ο φάκελος (folder) κλειδιών θα χρησιμοποιηθεί για την αυτόματη εκκίνηση της μικρο-υπηρεσίας AAM ως Core AAM μικρο-υπηρεσίας. Για την επαλήθευση των χρηστών και εφαρμογών (third-party users), η υπηρεσία Core AAM αξιοποιεί το Java KeyStore, του οποίου η διαδικασία δημιουργίας περιγράφεται παρακάτω.

Έκδοση πιστοποιητικού SSL

Η έκδοση του πιστοποιητικού SSL επιτυγχάνεται από κάποια αρχή έκδοσης SSL πιστοποιητικού (CA) όπως το Let's Encrypt⁶ και για την έκδοσή του μπορεί να χρησιμοποιηθεί το εργαλείο certbot⁷.

- Εγκαταστήστε το certbot:

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get
sudo apt-get install certbot python-certbot-apache
```

- Αποκτήστε το πιστοποιητικό εκτελώντας την εντολή
certbot --apache certonly

Η θύρα Apache (80 από προεπιλογή) θα πρέπει να είναι προσβάσιμη από έξω στο τείχος προστασίας σας. Επιλέξτε αυτόνομο (επιλογή 2) και εισαγάγετε το όνομα τομέα σας.

- Μετά την επιτυχή εκτέλεση, πλοηγηθείτε στην τοποθεσία:

/etc/letsencrypt/live/<domain_name>/

όπου μπορείτε να βρείτε το πιστοποιητικό και το ιδιωτικό κλειδί σας (5 αρχεία συνολικά, cert.pem, chain.pem, fullchain.pem, privkey.pem, README).

Δημιουργία Java Keystore με το εκδοθέν πιστοποιητικό SSL, που απαιτείται για την ανάπτυξη Core AAM

Για τη δημιουργία Java Keystore που περιέχει το πιστοποιητικό, χρησιμοποιήστε την εφαρμογή KeyStore Explorer⁸.

- (προαιρετικά) Επιθεωρήστε τα ληφθέντα αρχεία επιλέγοντας από το μενού “Επιθεώρηση” --> “Επιθεώρηση αρχείου”.
- Επιλέξτε “Δημιουργήστε ένα νέο Keystore” --> “PKCS #12”.
- Επιλέξτε “Εργαλεία” --> “Εισαγωγή ζεύγους κλειδιών” --> “PKCS #12”.
- Αποεπιλέξτε το “Κρυπτογραφημένο ιδιωτικό κλειδί”.
Περιηγηθείτε και ορίστε το ιδιωτικό σας κλειδί (*privkey.pem*)
Περιηγηθείτε και ορίστε το πιστοποιητικό σας (*fullchain.pem*)
- Επιλέξτε “Εισαγωγή” --> πληκτρολογήστε ψευδώνυμο για το πιστοποιητικό για αυτόν τον χώρο αποθήκευσης κλειδιών.
- Εισαγάγετε τον κωδικό πρόσβασης.
- Επιλέξτε “Αρχείο” --> “Αποθήκευση” --> εισάγετε τον κωδικό πρόσβασης που έχετε ορίσει προηγουμένως --> <όνομα αρχείου>.p12 π.χ. core.p12.
Το όνομα αρχείου θα χρησιμοποιηθεί ως παράμετρος διαμόρφωσης του στοιχείου Platform AAM: server.ssl.key-store=classpath:core.p12
- Αντιγράψτε το core.p12 keystore αρχείο στο φάκελο configuration.

2.3.2.1.3 Δημιουργία φακέλων για την ανάπτυξη (deployment) του symbIoTe core

Από το αποθετήριο πρέπει να αντιγραφούν τα παρακάτω αρχεία στον φάκελο configuration:

- AuthenticationAuthorizationManager folder,
- CoreConfigProperties folder,

⁶ <https://letsencrypt.org/>

⁷ <https://certbot.eff.org/>

⁸ <http://keystore-explorer.org/downloads.html>

- *CoreInterface folder,*
- *bootstrap.properties file,*
- *nginx-ngrok.conf,*
- *nginx-prod.conf,*
- *nginx.conf.*

Οι παράμετροι που έχουν χαρακτηριστεί ως *FILL ME* πρέπει να συμπληρωθούν στα παρακάτω αρχεία:

- *CoreConfigProperties/application.properties*
- *AuthenticationAuthorizationManager/bootstrap.properties*
- *bootstrap.properties*

Στον φάκελο *CoreConfigProperties* πρέπει να γίνουν οι απαραίτητες αλλαγές για τα αρχεία *.properties (π.χ. MongoDB, rabbitMQ credentials):

- Κλωνοποιούμε το φάκελο:
git clone https://github.com/symbiote-h2020/SymbioteCore.git configuration/Cloud-ConfigProperties"
- κανουμε ότι αλλαγές απαιτούνται μέσα στο φάκελο CoreConfigProperties (cd Core-ConfigProperties)
- υποβάλλουμε τις αλλαγές (commit):
git commit -am "SymbIoTe Core configuration"
cd ..

2.3.2.1.4 Δημιουργία docker volumes για την ανάπτυξη (deployment) σε Docker περιβάλλον

Τα παρακάτω βήματα γίνονται μέσα στο φάκελο *configuration*.

Δημιουργούμε τα docker volumes για τη διατήρηση των CoreConfigProperties με την παρακάτω εντολή:

```
docker container run --rm -v $PWD/CoreConfigProperties:/source -v {docker stack name}_symbiote-vol-config:/home/CoreConfigProperties -w /source alpine cp -r . /home/CoreConfigProperties/
επιλέγοντας το όνομα {docker stack name} που θέλουμε να δώσουμε στο docker stack όπως "symbiote-core".
```

Στη συνέχεια πρέπει να αντιγραφούν στο φάκελο *configuration* τα docker-compose files που απαιτούνται και είναι διαθέσιμα στο αποθετήριο:

- *docker-compose-swarm-core.yml*
- *docker-compose-prod-swarm-core.yml*

Σε περίπτωση ύπαρξης διαμεσολαβητή (proxy) πρέπει να προστεθούν οι απαραίτητες ρυθμίσεις για τον διαμεσολαβητή (JAVA_HTTP_PROXY, JAVA_HTTPS_PROXY, JAVA_SOCKS_PROXY, JAVA_NON_PROXY_HOSTS) στο *docker-compose-swarm-core.yml* και *docker-compose-prod-swarm-core.yml* αρχεία.

2.3.2.1.5 Διαμόρφωση μικρο-υπηρεσίας NGINX για την ανάπτυξη (deployment) σε Docker περιβάλλον

Δημιουργούμε μέσα στον φάκελο *configuration* τον φάκελο *nginx-certificates* και αντιγράφουμε στον φάκελο **nginx-certificates** τα αρχεία πιστοποιητικών **fullchain.pem** and **privkey.pem** που δημιουργήσαμε προηγουμένως στον φάκελο /etc/letsencrypt/live/<domain_name>/ (ή εναλλακτικά με symbolic links):

```
sudo cp /etc/letsencrypt/live/{your domain}/fullchain.pem nginx-certificates/
sudo cp /etc/letsencrypt/live/{your domain}/privkey.pem nginx-certificates/
sudo chown -R {user}:{group} nginx-certificates
```

2.3.2.1.6 Ανάπτυξη σε περιβάλλον Docker Swarm

- Εκτελούμε την εντολή **docker swarm init** αν ο κόμβος δεν είναι swarm manager. Χρησιμοποιούμε swarm mode ώστε τα Docker secrets να είναι κρυπτογραφημένα και διαθέσιμα μόνο στα swarm services και όχι σε standalone containers.
- Η ανάπτυξη γίνεται με την εντολή **docker stack deploy -c docker-compose-swarm-core.yml -c docker-compose-prod-swarm-core.yml {docker stack name}**
όπου **{docker stack name}** είναι το όνομα που επιλέξαμε για το docker stack και χρησιμοποιήσαμε και παραπάνω για τη δημιουργία των volumes.
- Με την εντολή **docker stack ls** μπορούμε να δούμε το stack και τον αριθμό των υπηρεσιών που δημιουργήθηκαν.
- Με την εντολή **docker service ls** ελέγχουμε τη λίστα των μικρο-υπηρεσιών που τρέχουν στο **{docker stack name}** stack. Περιμένουμε λίγο μέχρι ο αριθμός των tasks (replicas) για κάθε υπηρεσία παραπάνω να μην είναι 0.
- Με την εντολή **docker logs <container id> -f** μπορούμε να δούμε την κατάσταση μίας μικρουπηρεσίας και αν έχει σηκωθεί επιτυχώς Σε αυτήν την περίπτωση θα εμφανίζεται το μήνυμα στα logs 'Started in ... seconds (JVM running for ...)'.
- Με την εντολή **docker stack rm {docker stack name}** κατεβάζουμε το stack και τις εφαρμογές που τρέχουν σε αυτό (Services, networks, secrets κλπ).
- Με την εντολή **docker swarm leave --force** μπορούμε να σταματήσουμε το swarm mode του κόμβου.

2.3.2.1.7 Ανάπτυξη σε περιβάλλον Kubernetes

Για την ανάπτυξη του συστήματος σε Kubernetes θα πρέπει να μετασχηματίσουμε τα αρχεία που διαμορφώσαμε στα προηγούμενα βήματα για την χρήση τους από το Kubernetes. Τα δεδομένα των αρχείων θα εισαχθούν στα αντικείμενα Secret, ConfigMap.

- Η δημιουργία του Secret με το πιστοποιητικό για το Core AAM γίνεται ως εξής:
kubectrl create secret opaque aamkeystore --from-file=aamkeystore=core.p12
- Η δημιουργία του Secret με το πιστοποιητικό για το NGINX γίνεται ως εξής:
kubectrl create secret tls certificate --cert=fullchain.pem --key=privkey.pem
- Τα αρχεία *.properties που επεξεργαστήκαμε και δημιουργήθηκαν προηγουμένως θα πρέπει να ενσωματωθούν σε αρχεία τύπου ConfigMap:
kubectrl create configmap bootstrap-properties --from-file=bootstrap.properties
Η παραπάνω διαδικασία για τη δημιουργία του ConfigMap θα πρέπει να γίνει για κάθε αρχείο τύπου *.properties.
- Στο αποθετήριο μπορούμε να βρούμε τα υπόλοιπα αρχεία απαραίτητα για την εγκατάσταση του συστήματος κάτω από το φάκελο **core**⁹. Για την εγκατάστασή τους εκτελούμε τις παρακάτω εντολές:
kubectrl apply -f deployments/.
kubectrl apply -f services/.
- Για την πρόσβαση των υπηρεσιών από το Internet θα πρέπει να δημιουργήσουμε Ingress κανόνες όπου στο **host** εισάγουμε τη διεύθυνση από όπου θέλουμε να είναι προσβάσιμα. Υπάρχουν παραδείγματα στο φάκελο Ingress όπου μπορούμε να τα επεξεργαστούμε ανάλογα με τις ανάγκες μας.
kubectrl apply -f ingress/.
kubectrl apply -f pvc/.

⁹ <https://github.com/iotfedsproject/IoTFeds-K8s-Deployment/tree/main/core>

- Με την εντολή **kubectl get pods** ελέγχουμε τη λίστα των μικρο-υπηρεσιών που τρέχουν στο *Kubernetes*.
- Με την εντολή **kubectl logs pod-name -f** μπορούμε να δούμε την κατάσταση μίας μικρουπηρεσίας και αν έχει σηκωθεί επιτυχώς. Σε αυτήν την περίπτωση θα εμφανίζεται το μήνυμα στα logs 'Started in ... seconds (JVM running for ...)'.

2.3.2.2 *symbIoTe Cloud deployment*

2.3.2.2.1 Εγγραφή του ιδιοκτήτη της πλατφόρμας

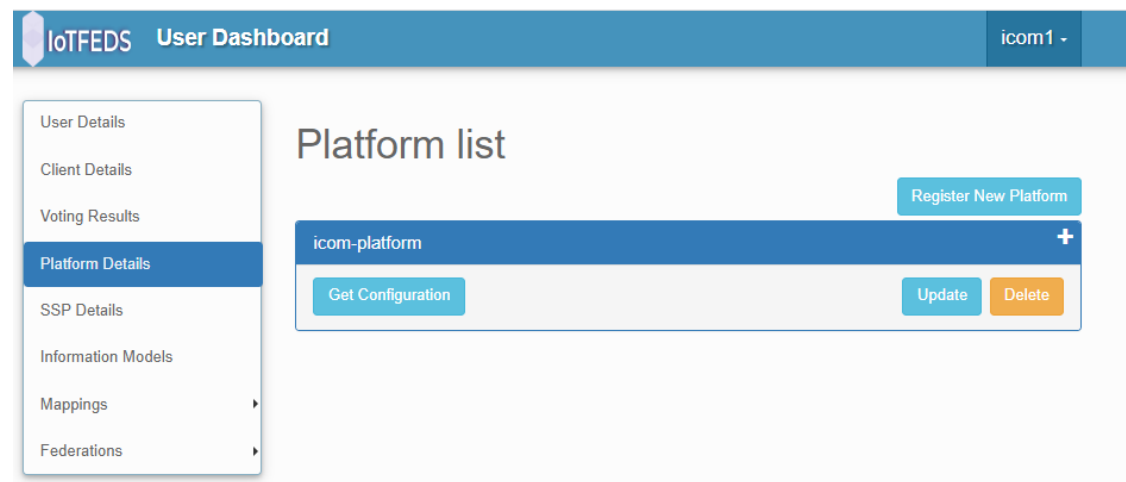
Ο ιδιοκτήτης της πλατφόρμας πρέπει να είναι εγγεγραμμένος χρήστης στην IoTFeds πλατφόρμα με τον ρόλο του “service owner”. Αναλυτικά τα βήματα παρουσιάζονται στο Κεφάλαιο 4 - AdminGUI-2.

2.3.2.2.2 Εγγραφή της IoT πλατφόρμας

Ο ιδιοκτήτης της πλατφόρμας μπορεί να συνδεθεί στο σύστημα μετά την επιτυχημένη εγγραφή του. Στη συνέχεια ο χρήστης πρέπει να εγγράψει την IoT πλατφόρμα του στο σύστημα. Αναλυτικά τα βήματα παρουσιάζονται στο Κεφάλαιο 4 - AdminGUI-4.

2.3.2.2.3 Διαμόρφωση των απαιτούμενων αρχείων παραμέτρων

Στη συνέχεια ο χρήστης μπορεί να κατεβάσει τα απαιτούμενα αρχεία για τη διαμόρφωση του υποσυστήματος (*symbIoTe Cloud*) για την επίτευξη επιπέδου διαλειτουργικότητας L1 (Πύλη IoT και μηχανή αναζήτησης) ή L2 (Ομοσπονδίες IoT), που περιγράφηκαν στο Π1.3, πατώντας στο πάνελ της πλατφόρμας το κουμπί “Get Configuration” όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 3: Πληροφορίες εγγεγραμμένης IoT πλατφόρμας.

Ο χρήστης συμπληρώνει τη φόρμα με τα επιθυμητά στοιχεία και κατεβάζει το αρχείο *configuration.zip* που περιέχει τα αρχεία παραμέτρων που απαιτούνται για την εγκατάσταση (deployment) των στοιχείων λογισμικού του *symbIoTe Cloud* σε *Docker Swarm*.

IoT FEDS User Dashboard

Get Configuration for platform icom-platform

Platform Admin Username <input type="text" value="Mandatory Field"/> The Platform Owner Username in your Cloud deployment	Platform Admin Password <input type="password" value="Mandatory Field"/> The Platform Owner Password in your Cloud deployment
Components Keystore Password <input type="text" value="Optional"/> The keystore password in your cloud components (apart from PAAM) Default: 'pass'	AAM Keystore File Name <input type="text" value="Optional"/> The name of the AAM keystore generated. Default: 'paam-keystore-platform_name'
AAM Keystore Password <input type="password" value="Optional"/> The password of the AAM keystore. MUST NOT be longer than 7 chars Default: 'pass'	Token Validity in milliseconds <input type="text" value="Optional"/> How long the tokens issued to your users (apps) are valid in ms. Think maybe of an hour, day, week? Default: 600000 (10 mins)
Type <input type="text" value="No"/> Use built-in plugin provided by rap. Default: No	Deployment Type <input type="text" value="Docker"/> Choose your deployment type
Compliance Level <input type="text" value="L1"/> Choose your compliance level	

Buttons: **Get Configuration** **Close**

Εικόνα 4: Διαμόρφωση παραμέτρων πλατφόρμας.

2.3.2.2.4 Δημιουργία φακέλου για την εγκατάσταση του symbIoTe Cloud με Docker Swarm

- Ο χρήστης πρέπει να δημιουργήσει τον φάκελο για την εγκατάσταση, π.χ. `iotfedsL2-cloud`:
mkdir iotfedsL2-cloud
- Μεταβαίνει στον φάκελο που δημιουργήθηκε:
cd iotfedsL2-cloud
- Γίνεται αποσυμπίεση στο φάκελο `iotfedsL2-cloud` του αρχείου `configuration.zip`.
- Μέσα στον φάκελο `CloudConfigProperties`:

Το αρχείο `CloudConfigProperties/application.properties` περιέχει τα URLs των διαλειτουργικών διεπαφών (interworking interfaces):

1. `coreInterface` για την northbound κίνηση προς το symbIoTe Core, π.χ.
symbIoTe.core.interface.url = <https://iotfeds-symbiotecore.intracom-telecom.com/coreInterface>
2. `cloudCoreInterface` για southbound κίνηση από τις IoT πλατφόρμες (που τρέχουν το symbIoTe Cloud) προς το symbIoTe Core:
symbIoTe.core.cloud.interface.url = <https://iotfeds-symbiotecore.intracom-telecom.com/cloudCoreInterface>

- Στον φάκελο *iotfedsL2-cloud*:
- Στο αρχείο *AuthenticationAuthorizationManager/cert.properties* υπάρχουν οι παράμετροι για τη διεύθυνση του symbIoTe core AAM (που είναι διαθέσιμος μέσα από το *coreInterface*):

coreAAMAddress = <https://iotfeds-symbiotecore.intracom-telecom.com/coreInterface>

- Υποβολή (commit) των αλλαγών:

Μέσα στον φάκελο *CloudConfigProperties* εκτελούμε τις παρακάτω εντολές:

```
rm -r .git
git init
git config user.email you@example.com
git config user.name "Your User Name"
git add .
git commit -m "Your platform's name"
```

- Δημιουργία Docker volume για την διατήρηση της παραμετροποίησης του φακέλου *CloudConfigProperties*:

Μέσα στον φάκελο *iotfedsL2-cloud* εκτελούμε την επόμενη εντολή Docker αντικαθιστώντας το **{docker stack name}** με το επιθυμητό όνομα για τη δημιουργία του Docker stack (π.χ., *iotfedsL2_cloud*).

```
docker container run --rm -v $PWD/CloudConfigProperties:/source -v
{docker stack name} _symbiote-vol-config:/home/CloudConfigProperties -w
/source alpine cp -r . /home/CloudConfigProperties/
```

- Διαμόρφωση των αρχείων Docker compose
 - Αντιγράφουμε στο φάκελο *iotfedsL2-cloud* τα παρακάτω Docker compose αρχεία που είναι διαθέσιμα στο αποθετήριο κώδικα:

```
docker-compose-swarm-L2.yml
docker-compose-prod-swarm-L2.yml
```

- Σε περίπτωση ύπαρξης διακομιστή μεσολάβησης (proxy) στο **docker-compose-swarm-L2.yml** πρέπει να καθοριστούν οι σχετικές παράμετροι (proxy settings) αναλόγως: **JAVA_HTTP_PROXY**, **JAVA_HTTPS_PROXY**, **JAVA_NON_PROXY_HOSTS**
- Διαμόρφωση της μικρο-υπηρεσίας NGINX που τρέχει στο symbIoTe Cloud L2 services docker stack
 - Η διαμόρφωση αφορά τα αρχεία **nginx.conf** και **nginx-prod.conf** που βρίσκονται στον φάκελο *iotfedsL2-cloud*, ενώ τα αρχεία **nginx.conf**, **nginx-prod.conf** and **nginx-ngrok** μπορούν να αφαιρεθούν.
 - Η παράμετρος **server_name** στα **nginx.conf** και **nginx-prod.conf** αρχεία πρέπει να ανταποκρίνεται στο όνομα της πλατφόρμας που δόθηκε κατά την εγγραφή της π.χ *test-platform*.
 - Απαιτείται στη συνέχεια η δημιουργία **fullchain.pem** και **privkey.pem** πιστοποιητικών¹⁰, όπως για παράδειγμα με χρήση *certbot*:
certbot --manual certonly --preferred-challenges dns -d <domain name> --email <admin email>
 όπου θα ζητηθεί να καταχωρηθεί η εγγραφή DNS TXT κάτω από ένα συγκεκριμένο domain name. Μετα τη δημιουργία των πιστοποιητικών το letsencrypt ενημερώνει τον χρήστη για ηγνη τοποθεσία των αρχείων. Από την τοποθεσία αυτή τα πιστοποιη-

¹⁰ <https://github.com/symbIoTe-h2020/symbIoTeCloud/wiki/2.1-Configuration-of-NGINX>

ητικά πρέπει να αντιγραφούν στην τοποθεσία που δηλώνεται στα αρχεία παραμέτρων του nginx, στον φάκελο δηλαδή nginx-certificates folder που πρέπει να δημιουργηθεί όπως παρακάτω:

Στο φάκελο **iotfedsL2-cloud** ο χρήστης πρέπει να δημιουργήσει τον φάκελο **nginx-certificates**, όπου το nginx έχει διαμορφωθεί για να βρίσκει τα αρχεία **full-chain.pem** και **privkey.pem files** και όπου πρέπει να αντιγραφούν τα πιστοποιητικά που δημιουργήθηκαν ωρίτερα:

```
mkdir nginx-certificates
```

```
sudo cp /etc/letsencrypt/live/{your domain}/fullchain.pem nginx-certificates/
```

```
sudo cp /etc/letsencrypt/live/{your domain}/privkey.pem nginx-certificates/
```

```
sudo chown -R {user}:{group} nginx-certificates
```

- Εκτέλεση του symbIoTe L1/L2 Cloud services Docker Stack
 - Η αναπτυξιά (deployment) γίνεται μέσα από τον φάκελο **iotfedsL2-cloud** που έχουμε δημιουργήσει και πρέπει να έχει την παρακάτω δομή:



Εικόνα 5: Δομή φακέλου διαμόρφωσης και ανάπτυξης του symbIoTe Cloud σε Docker Swarm.

- Εκτελούμε την εντολή **docker swarm init** αν ο κόμβος δεν είναι swarm manager. Χρησιμοποιούμε swarm mode ώστε τα Docker secrets να είναι κρυπτογραφημένα και διαθέσιμα μόνο στα swarm services και όχι σε standalone containers.
- Η ανάπτυξη γίνεται με την εντολή **docker stack deploy -c docker-compose-swarm-L2.yml -c docker-compose-prod-swarm-L2.yml {docker stack name}**
όπου *{docker stack name}* είναι το όνομα που θα δώσουμε στο docker stack και χρησιμοποιήσαμε και παραπάνω για τη δημιουργία των volumes.
- Με την εντολή **docker stack ls** μπορούμε να δούμε το stack και τον αριθμό των υπηρεσιών που δημιουργήθηκαν.
- Με την εντολή **docker ps** ελέγχουμε τη λίστα των μικρο-υπηρεσιών που τρέχουν στο *{docker stack name}* stack.
- Με την εντολή **docker image ls** ελέγχουμε ότι όλες οι εικόνες images έχουν δημιουργηθεί (π.χ. οι εικόνες γίνονται pull από το Docker Hub την πρώτη φορά). Περιμένουμε λίγο μέχρι ο αριθμός των tasks (replicas) για κάθε υπηρεσία παραπάνω να μην είναι 0.
- Με την εντολή **docker logs <container id> -f** μπορούμε να δούμε την κατάσταση μίας μικρουπηρεσίας και αν έχει σηκωθεί επιτυχώς. Σε αυτήν την περίπτωση θα εμφανίζεται το μήνυμα στα logs 'Started in ... seconds (JVM running for ...)'.
 • Με την εντολή **docker stack rm {docker stack name}** κατεβάζουμε το stack και τις εφαρμογές που τρέχουν σε αυτό (Services, networks, secrets κλπ).
- Με την εντολή **docker swarm leave --force** μπορούμε να σταματήσουμε το swarm mode του κόμβου.

2.3.2.2.5 Δημιουργία φακέλου για την εγκατάσταση του symbIoTe Cloud σε Kubernetes

Για την ανάπτυξη του υποσυστήματος σε Kubernetes θα πρέπει να μετασχηματίσουμε τα αρχεία που διαμορφώσαμε στα προηγούμενα βήματα για την χρήση τους από το Kubernetes. Τα δεδομένα των αρχείων θα εισαχθούν σε αντικείμενο ConfigMap. Ακόμη θα πρέπει να γίνει και η εφαρμογή των αρχείων απαραίτητων για την εγκατάστασης της πλατφόρμας.

- Στο αποθετήριο μπορούμε να βρούμε τα υπόλοιπα αρχεία απαραίτητα για την εγκατάσταση του συστήματος κάτω από το φάκελο **cloud**¹¹.
- Θα επεξεργαστούμε το αρχείο ingress-patch.json και θα αλλάξουμε τις τιμές του πεδίου value στη διεύθυνση από όπου θέλουμε να είναι προσβάσιμη η πλατφόρμα μας.
- Θα θέλαμε να δημιουργήσουμε και ένα ξεχωριστό namespace για αυτή τη πλατφόρμα: `kubectrl create namespace <my-namespace>`
Ακόμη αλλάζουμε και την τιμή namespace στο kustomization.yaml
- Για την εφαρμογή των αρχείων αλλά και τη δημιουργία οσων υπολοίπων εκτελούμαι στον φακέλο cloud την εντολή: `kubectrl apply -k .`
- Με την εντολή **kubectrl get pods** ελέγχουμε τη λίστα των μικρο-υπηρεσιών που τρέχουν στο *Kubernetes*.
- Με την εντολή **kubectrl logs pod-name -f** μπορούμε να δούμε την κατάσταση μίας μικρουπηρεσίας και αν έχει σηκωθεί επιτυχώς. Σε αυτήν την περίπτωση θα εμφανίζεται το μήνυμα στα logs 'Started in ... seconds (JVM running for ...)'.
 • Με την εντολή **docker swarm leave --force** μπορούμε να σταματήσουμε το swarm mode του κόμβου.

2.3.2.2.6 Δημιουργία RAP plugin

Οι πάροχοι δεδομένων πρέπει να υλοποιήσουν τον κώδικα που απαιτείται για την απόκτηση δεδομένων πόρων από την IoT πλατφόρμα τους. Το RAP plugin λειτουργεί ως μικρουπηρεσία, διασυνδέοντας το στοιχείο λογισμικού RAP και την ίδια την πλατφόρμα IoT για να επιτρέψει

¹¹ <https://github.com/iotfedsproject/IoTFeds-K8s-Deployment/tree/main/cloud>

την πρόσβαση συμβατή με L1 ή L2 σε δεδομένα πόρων εντός πλατφορμών IoT. Το RAP plugin για την εκάστοτε πλατφόρμα μπορεί να υλοποιηθεί σε Java ή οποιαδήποτε άλλη επιθυμητή γλώσσα προγραμματισμού¹². Οι οδηγίες για τη δημιουργία ενός RAP plugin βασίζονται στο [wiki](#) που είναι διαθέσιμο στο αποθετήριο κώδικα του symbIoTe. Τέλος, ο μηχανισμός ειδοποίησης μπορεί να ενεργοποιηθεί με την υλοποίηση των σχετικών διεπαφών. Ένα παράδειγμα μιας εφαρμογής που έχει εγγραφεί σε έναν πόρο (subscription) και οδηγίες ανάπτυξης για εφαρμογή υπάρχουν στο αποθετήριο κώδικα.

- Ο κώδικας (RapPluginExample project) είναι διαθέσιμος στο παρακάτω αποθετήριο κώδικα του symbIoTe: <https://github.com/symbiote-h2020/RapPluginExample>.
- Το RAP plugin πρέπει να προσαρμοστεί κατάλληλα για να επικοινωνεί με την εκάστοτε IoT πλατφόρμα τροποποιώντας το αρχείο RapPluginExampleApplication.java στο path:
src/main/java/eu/h2020/symbiote/rappluginexample/RapPluginExampleApplication.java
να ώστε:
 - Να διαχειρίζεται requests από το RAP (read resource, read resource history, actuate on a resource, invoke service, subscribe/unsubscribe),
 - Να έχει πρόσβαση στους πόρους και τα δεδομένα τους που είναι αποθηκευμένα στην IoT πλατφόρμα,
 - Να απαντάει στο στοιχείο λογισμικού RAP και τελικά σε μία τρίτη εφαρμογή του symbIoTe οικοσυστήματος,
 - Να μπορεί να εγγραφεί σε ή να απεγγραφεί από έναν IoT πόρο για παρακολούθηση μετρήσεων (subscribe/unsubscribe),
 - Να δημιουργεί ειδοποιήσεις για έναν εγγεγραμμένο πόρο.

Για να το επιτύχει αυτό, ο διαχειριστής της πλατφόρμας πρέπει να προσαρμόσει τις παρακάτω μεθόδους:

- *public String getResource(List resourceInfo)*: Για να εξυπηρετηθεί ένα αίτημα μίας τρίτης εφαρμογής για τη λήψη δεδομένων ενός IoT πόρου, το στοιχείο RAP θα στείλει ένα μήνυμα στο RAP plugin το οποίο θα ενεργοποιήσει τη μέθοδο *getResource* στον ResourceAccessListener σε περίπτωση διαβάσματος της τελευταίας τιμής. Η μέθοδος επιστρέφει την τελευταία μέτρηση του IoT πόρου που έχει μετατραπεί σε JSON και επιστραφεί ως String.
- *public String getResourceHistory(List resourceInfo, int top, Query filterQuery)*: Αυτή η μέθοδος καλείται να διαβάσει ιστορικές τιμές μετρήσεων. Τα ορίσματα είναι: το resourceInfo, που αντιπροσωπεύει τον πόρο, top, που είναι ο αριθμός των μετρήσεων (observations) που πρέπει να επιστραφούν (max =100) και το filterQuery, το οποίο αντιπροσωπεύει άλλα φίλτρα (αν ωστόσο η ιδιότητα rap-plugin.filtersSupported είναι ψευδής, θα να είναι πάντα μηδενική). Η μέθοδος επιστρέφει μετρήσεις, που μετατρέπονται σε JSON και επιστρέφονται ως String. Σε περίπτωση που δεν είναι δυνατή η ανάγνωση, η μέθοδος θα εμφανίσει RapPluginException.
- *public void actuateResource(String internalId, Map<String, Map<String, Value>> capabilities)*: Τα ορίσματα της μεθόδου περιλαμβάνουν το αναγνωριστικό εσωτερικού πόρου και τις δυνατότητες ενεργοποίησης (capabilities) που αντιστοιχούν σε παραμέτρους και τις αντίστοιχες τιμές. Οι τιμές μπορεί να είναι τύπου PrimitiveValue, PrimitiveValueArray, ****ComplexValue **** και ComplexValueArray. Η μέθοδος για την ενεργοποίηση συσκευών θα εμφανίσει RapPluginException όταν δεν μπορεί να εκτελεστεί η ενεργοποίηση.

¹² <https://github.com/symbiote-h2020/SymbioteCloud/wiki/4.3-Creating-RAP-plugin-in-other-languages>

- *public String invokeService(String internalId, Map<String, Value> parameters)*: Στην περίπτωση υπηρεσίας υπάρχει η μέθοδος *invokeService* στον *ServiceAccessListener* που δέχεται ως παραμέτρους το *id* του IoT πόρου (εγγεγραμμένη υπηρεσία) και ένα *Map* από παραμέτρους και τιμές όπως στην περίπτωση του *actuation*. Ομοίως, σε περίπτωση αποτυχίας εμφανίζεται *RapPluginException*.
- *public void subscribeResource(String resourceId)*: Η μέθοδος αυτή καλείται όταν γίνεται παρακολούθηση μετρήσεων/εγγραφή (*subscription*) σε έναν IoT πόρο και χρειάζεται μόνο το *internal Id* του IoT πόρου.
- *public void unsubscribeResource(String resourceId)*: Η μέθοδος αυτή καλείται όταν η παρακολούθηση των μετρήσεων ενός IoT πόρου σταματάει (*unsubscribe*). Ο ιδιοκτήτης της IoT πλατφόρμας πρέπει να επιλέξει πότε θα στέλνονται οι ειδοποιήσεις, που μπορεί να είναι περιοδικά ή με βάση ένα γεγονός (*event*). Η ειδοποίηση στέλνεται στον συνδρομητή (*subscriber*) χρησιμοποιώντας την μέθοδο *sendNotification* του *RAP plugin*, η οποία δέχεται το *Observation* και την στέλνει το στοιχείο λογισμικού *RAP*, το οποίο στη συνέχεια το προωθεί στον τελικό χρήστη μέσω ενός *web socket*.
- *private void sendPush()*.
- Διαμόρφωση παραμέτρων για την εγκατάσταση του *RAP plugin*

Απαιτείται η προσθήκη των αρχείου διαμόρφωσης παραμέτρων του *RAP plugin* στο φάκελο εγκατάστασης του *symbIoTe cloud*: *WORKING_DIR/RAPPluginStarter/application.properties* που έχει τις παρακάτω παραμέτρους:

- *spring.application.name = TheNameOfRapPlugin*: όπου ορίζουμε το όνομα του *RAP plugin* και το οποίο χρησιμοποιείται στη συνέχεια κατά την καταχώρηση του IoT πόρου στο σύστημα.
- Οι πληροφορίες για το *RabbitMQ* μέσω του οποίου επιτυγχάνεται η επικοινωνία μεταξύ του *RAP plugin* και του στοιχείου λογισμικού *RAP*.
rabbit.host=symbiote-rabbitmq
rabbit.username=guest
rabbit.password=guest
rabbit.replyTimeout=60000
- *rap-plugin.filtersSupported=false*: Αν είναι *false*, ο *RAP* φιλτράρει τις απαντήσεις του *RAP plugin* για να τις στείλει στους *listeners*, διαφορετικά το *RAP plugin* φιλτράρει τα δεδομένα που πρέπει να επιστραφούν με βάση το *query* (ιστορικά δεδομένα).
- *rap-plugin.notificationsSupported=false*: Αφορά τη δυνατότητα υποστήριξης ρολών δεδομένων (*streamingdata*) ή μη του *RAP plugin*. Η τιμή είναι *true* όταν μπορεί να δεχθεί συνδρομές (*subscriptions*) καθώς και να κάνει *push* δεδομένα όταν υπάρχει αλλαγή στη τιμή.

Τέλος, στο αρχείο *CloudConfigProperties/ResourceAccessProxy.properties* το *rap.enableSpecificPlugin=false* αφορά την ενημέρωση του *RAP* για την ύπαρξη/εγκατάσταση ή μη κάποιου *RAP plugin*.

- Δημιουργία *docker image* του *RAP plugin* και προσθήκη του στην *symbIoTe Cloud* εγκατάσταση.
 - Το εργαλείο **gradlew.bat** μέσα στον φάκελο μπορεί να χρησιμοποιηθεί για τη δημιουργία του αρχείου *build/libs/RapPluginExample-0.0.1-SNAPSHOT-run.jar*.

- Η δημιουργία του docker image γίνεται με τη χρήση του Dockerfile που δίνεται. Το Dockerfile πρέπει να βρίσκεται στον ίδιο φάκελο με το αρχείο jar που δημιουργήθηκε.

```
FROM openjdk:15-alpine
WORKDIR /home
ENV componentName "YourRapPlugin"
ENV componentVersion 1.0.0
ENV http_proxy http://xxx.xxx.xxx.xxx:yyy
RUN apk --no-cache add \
    _git \
    _unzip \
    _wget \
    _bash_
COPY RapPluginExample-0.0.1-SNAPSHOT-run.jar /home/Rap-
PluginExample-0.0.1-SNAPSHOT-run.jar
CMD java $JAVA_HTTP_PROXY $JAVA_HTTPS_PROXY
$JAVA_NON_PROXY_HOSTS -jar $(ls *run.jar)
```

- Το επόμενο βήμα αφορά τη δημιουργία του αρχείου docker-compose-swarm-rapplugin.yml, του yml αρχείου για την εγκατάσταση του RAP plugin σε περιβάλλον Docker Swarm (αντίστοιχη μετατροπή απαιτείται για εγκατάσταση σε περιβάλλον Kubernetes). Στο αρχείο αυτό η παράμετρος your_rap_plugin_image_name πρέπει να αντικατασταθεί με το όνομα του RAP plugin image που δημιουργήθηκε. Επιπλέον στην περίπτωση ύπαρξης proxy απαιτούνται proxy settings.

```
x-environment:
&default-environment
- JAVA_HTTP_PROXY=-Dhttp.proxyHost=xxx.xxx.xxx.xxx -
Dhttp.proxyPort=yyy
- JAVA_HTTPS_PROXY=-Dhttps.proxyHost=xxx.xxx.xxx.xxx -
Dhttps.proxyPort=yyyy
-JAVA_NON_PROXY_HOSTS=-Dhttp.nonProxyHosts=*symbiote-
h2020.eu/*intracom-telecom.com/symbiote-cloudconfig/symbiote-aam
services:
symbiote-rpe:
image: your_rap_plugin_image_name
depends_on:
- symbiote-aam
deploy:
resources:
limits:
memory: 350M
networks:
- symbioteL2
environment: *default-environment
secrets:
- source: rpeconfig
target: /home/application.properties
secrets:
rpeconfig:
file: ./RAPPluginStarter/application.properties
```

- Το τελευταίο βήμα αφορά την εντολή εγκατάστασης του symbIoTe cloud συμπεριλαμβάνοντας το RAP plugin: **docker stack deploy -c docker-**

**compose-swarm-L2.yml -c docker-compose-prod-swarm-L2.yml -c
docker-compose-swarm-raplugin.yml.**

2.3.3 Blockchain as a Service (BaaS)

Εξ' ορισμού το blockchain αποτελεί ένα αποκεντρωμένο σύστημα, με πολλαπλούς κόμβους που ο καθένας τρέχει ξεχωριστά και επικοινωνεί με τους υπόλοιπους. Στο Hyperledger Fabric αυτό επιτυγχάνεται με πολλαπλά Docker containers.

Τα βασικά δομικά στοιχεία του δικτύου του HLF είναι οι peers (κάθε peer αποτελεί και έναν κόμβο του δικτύου), τα Certificate Authorities (CAs) και οι Orderer peers (ο Orderer είναι απαραίτητο δομικό στοιχείο του Fabric). Κάθε peer και CA τρέχει ξεχωριστά στο δικό του container, εξασφαλίζοντας έτσι ότι αν ένας κόμβος του δικτύου δεν είναι διαθέσιμος εξαιτίας κάποιας αστοχίας, οι υπόλοιποι κόμβοι θα συνεχίσουν να λειτουργούν κανονικά. Επιπρόσθετα, κάθε Smart Contract (SC), ή στη γλώσσα του Fabric chaincode, τρέχει σε δικό του container. Πέραν αυτού, κάθε chaincode εγκαθίσταται σε όλους τους κόμβους του δικτύου έτσι ώστε διαφορετικές κλήσεις σε smart contracts να απαντώνται από έναν τυχαίο κόμβο. Επομένως, κάθε chaincode μπορεί να τρέχει σε πολλαπλά containers, ένα container δηλαδή για κάθε κόμβο στον οποίο εκτελείται.

Το δίκτυο με όλα τα components του, δηλαδή όλα τα Docker containers, τίθενται σε λειτουργία με τη βοήθεια του Docker Compose. Το Docker Compose είναι ένα εργαλείο που στόχο έχει τον ορισμό και την λειτουργία εφαρμογών που βασίζονται σε πολλαπλά containers. Με το Docker Compose είναι δυνατό να χρησιμοποιήσουμε ένα είδος αρχείου (.yaml) στο οποίο περιέχονται όλες τις απαραίτητες πληροφορίες για τη λειτουργία κ τον ορισμό των διαφόρων containers που θα αποτελέσουν το δίκτυο του blockchain και με μία εντολή να δημιουργηθεί όλο το δίκτυο.

Στην μέχρι τώρα υλοποίηση μας για τη διαχείριση των ομοσπονδιών και χρηστών, έχουμε έναν αριθμό από containers, τα οποία αναφέρονται παρακάτω:

- 2 containers για τα CAs.
- 5 containers για τους τέσσερις κόμβους (peers) του δικτύου.
- 4 containers για το SC που είναι υπεύθυνο για τη διαχείριση των χρηστών.
- 4 containers για το SC που είναι υπεύθυνο για τη διαχείριση των ομοσπονδιών.

Το σύστημα χρησιμοποιεί την έκδοση 2.3 του Hyperledger Fabric. Για την εγκατάστασή του πρέπει να ακολουθηθούν τα παρακάτω βήματα:

1. Εγκατάσταση των προ απαιτούμενων του fabric ακολουθώντας πιστά τις οδηγίες [στην βιβλιογραφία](#).
2. Λήψη του αποθετηρίου του BaaS με την εντολή «git clone <https://colab-repo.intracom-telecom.com/colab-projects/iotfeds/baas/baas.git>».
3. Εγκατάσταση των binaries και docker images όπως εξηγείται [στην βιβλιογραφία](#), κατά την ανάπτυξη του λογισμικού οι τελευταίες εκδόσεις ήταν Fabric v2.3.3 και Fabric CA v1.5.2, οπότε η κατάλληλη εντολή είναι «curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.3.3 1.5.2».
4. Στη συνέχεια κάποιος πρέπει να πλοηγηθεί στο ευρετήριο «test-network» από όπου πρέπει διαδοχικά να σηκωθούν τα containers των peers του δικτύου και έπειτα αυτά που περιέχουν τα chaincodes. Για διευκόλυνση υλοποιήθηκε βοηθητικό script που εκτελεί με την σωστή σειρά τις απαραίτητες εντολές, συνεπώς για τα παραπάνω αρκεί η εντολή «./net-up-and-contracts.sh».
5. Το επόμενο βήμα είναι να σηκωθεί το API του BaaS το οποίο βρίσκεται στο ευρετήριο «/asset-transfer-basic/application-javascript». Αφού πλοηγηθεί εκεί κάποιος σε ξεχωριστό terminal πρέπει να ελέγξει αν υπάρχει αρχείο με το όνομα «package-lock.json» καθώς και ευρετήρια με ονόματα «node_modules» και «wallet». Σε περίπτωση που υπάρχει κάποιο από αυτά πρέπει να διαγραφεί.
6. Για λόγους ασφαλείας, το αρχείο που περιέχει το private key με το οποίο υπογράφονται tokens για δικαίωμα ψήφου, δεν υπάρχει στο αποθετήριο και κάποιος πρέπει να το δη-

μιουργήσει πριν σηκωθεί η εφαρμογή. Στο ίδιο ευρετήριο πρέπει να δημιουργηθεί αρχείο με όνομα «.env» που περιέχει την παράμετρο «TOKEN_SECRET» η οποία θα πρέπει να έχει ως τιμή ένα ικανοποιητικά μεγάλο και τυχαίο hash, για παράδειγμα «TOKEN_SECRET=h234SKV59noweRN08vGTYklm34ropzg51hY»

7. Ο αυτοματοποιημένος nodemailer, θα πρέπει να ρυθμιστεί ώστε να στέλνει σωστό σύνδεσμο στους χρήστες για να μπορούν να ψηφίσουν. Προκειμένου να γίνει αυτό κάποιος πρέπει να βρει το παρακάτω αρχείο: `asset-transfer-basic/application-javascript/nodemailer/node_modules/nodemailer/sendmail.js` και στην γραμμή 35 να αντικαταστήσει την λέξη `localhost` με την διεύθυνση IP του υπολογιστή του, π.χ «`let url = "http:// 160.40.50.169:3000/";`» αντί για «`let url = "http://localhost:3000/";`».
8. Στο ευρετήριο «`/asset-transfer-basic/application-javascript`» πρέπει να τρέξουν η εντολή «`npm install`» και μετά την ολοκλήρωση της η εντολή «`node index.js`».
9. Το τελευταίο βήμα είναι να σηκωθεί η εφαρμογή που περιέχει το voting GUI. Σε ξεχωριστό terminal πρέπει να γίνει πλοήγηση στο ευρετήριο «`asset-transfer-basic/application-javascript/client`» και να τρέξει η εντολή «`npm start`».

3 Σχεδιασμός και δοκιμή του ολοκληρωμένου συστήματος IoTFeds – Ενδιάμεση έκδοση

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

3.1 Υποσύστημα symbIoTe

Το υποσύστημα symbIoTe¹³ επιτρέπει την καταγραφή, ανακάλυψη και διαμοιρασμό IoT συσκευών των IoT πλατφορμών που επιθυμούν να διαθέσουν IoT δεδομένα τους σε τρίτα συστήματα ή μεταξύ τους, καθώς επίσης προσφέρει μία ενοποιημένη διεπαφή για την πρόσβαση στα δεδομένα με ασφαλή τρόπο και μηχανισμούς για σημασιολογική διαλειτουργικότητα. Παρέχει λειτουργικότητες για τη διαχείριση χρηστών, ομοσπονδιών, πλατφορμών και των δεδομένων τους σε συνδυασμό με την τεχνολογία blockchain και την ενσωμάτωση των υπηρεσιών του BaaS υποσυστήματος.

Το symbIoTe υποσύστημα αποτελείται από ένα σύνολο από κεντριοποιημένες υπηρεσίες (symbIoTe core services) όπου οι εγγεγραμμένες IoT πλατφόρμες μπορούν να διαφημίσουν και να διαθέσουν τις IoT συσκευές τους σε άλλες πλατφόρμες του οικοσυστήματος IoTFeds ή προς τρίτους, παρόχους εφαρμογών.. Στα πλαίσια του έργου οι κεντριοποιημένες υπηρεσίες <symbIoTe-core> είναι διαθέσιμες στα παρακάτω urls:

<https://iotfeds-symbiotecore.intracom-telecom.com/administration/>, για την πιλοτική χρήση από τους φορείς του έργου,

<https://symbiote-core.iotfeds.intracom-telecom.com/administration/>, για τη δοκιμή του ενσωματωμένου συστήματος κατά την ανάπτυξη.

Τέλος, η πρόσβαση στα πραγματικά IoT δεδομένα των IoT συσκευών πραγματοποιείται από καταναεμημένες υπηρεσίες στη μεριά της IoT πλατφόρμας που τα προσφέρει (symbIoTe cloud services), καθώς και μηχανισμούς για τη διαχείριση των ομοσπονδιών της πλατφόρμας.

URL	
GitLab repository	https://colab-repo.intracom-telecom.com/colab-projects/iotfeds/symbIoTe , https://github.com/iotfedsproject/
Container registry	https://hub.docker.com/repository/docker/iotfedsproject/
Deployment yaml file	https://github.com/iotfedsproject/IoTFeds-K8s-Deployment , https://github.com/iotfedsproject/IoTFedsDockerSwarmDeployment

3.1.1 Στοιχείο λογισμικού για τη διαχείριση ομοσπονδιών, χρηστών και πλατφορμών (Administration)

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά, οι διεπαφές και οι δοκιμές τους για το στοιχείο λογισμικού Administration του symbIoTe για τη διαχείριση των IoT ομοσπονδιών, των χρηστών και των IoT πλατφορμών.

3.1.1.1 Περιγραφή

Η διαχείριση των ομοσπονδιών, των χρηστών και των πλατφορμών τους στο οικοσύστημα του symbIoTe γίνεται μέσα από τη γραφική διεπαφή χρήστη (Administration GUI). Τα εκάστοτε αιτήματα του χρήστη αποστέλλονται στο στοιχείο λογισμικού Administration του symbIoTe μέσω των αντίστοιχων endpoints του στοιχείου αυτού από το οποίο και ενημερώνεται το υποσύστημα BaaS.

¹³ <https://www.symbiote-h2020.eu/>

3.1.1.2 Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Τα παρακάτω χαρακτηριστικά προστέθηκαν στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας:

- Διαχείριση χρηστών – για την διαχείριση των χρηστών στο σύστημα.
- Διαχείριση πλατφορμών – για την καταχώρηση των IoT πλατφορμών.
- Διαχείριση ομοσπονδιών – για τη διαχείριση των ομοσπονδιών, των μελών και των κανόνων τους.

3.1.1.3 Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Το Administration παρέχει τη γραφική διεπαφή χρήστη που είναι προσβάσιμη στο παρακάτω url: <symbIoTe-core>/administration. Το symbIoTe Administration backend είναι προσβάσιμο μέσω HTTPS GET/POST REST αιτήματα από το γραφικό περιβάλλον.

3.1.1.3.1 Διεπαφές για τη διαχείριση IoT ομοσπονδίας

Στην ενότητα αυτή περιγράφονται οι διεπαφές του στοιχείου Administration του υποσυστήματος symbIoTe για τη διαχείριση των IoT ομοσπονδιών.

Πίνακας 1: Διεπαφές στοιχείου symbIoTe Administration για διαχείριση ομοσπονδίας.

Inter Face ID	Type	Feature	Interface	Request from	Request to	Release
SymbIoTeAdminFed-1	REST	Request to the administration component to create a core user. Effectively like registering a user from the administration GUI	/administration/register	IoTFeds User (Data consumer)	Administration	v1.0
SymbIoTeAdminFed-2	REST	Request for an organization/platform to be added to a federation	/administration/federation_vote_request /joinFederation	IoTFeds User (Data consumer)	Administration	v1.0
SymbIoTeAdminFed-3	REST	Returns the list of the join requests for a specific user	/administration/federation_vote_request /allForUser	IoTFeds User (Data consumer)	Administration	v1.0
SymbIoTeAdminFed-4	REST	Request for an organization/platform to be deleted from a federation	<iotfeds-url>/administration/federation_vote_request /removeUserFromFed	IoTFeds User (Data consumer)	Administration	v1.0

SymbIoTeAdminFed-5	REST	Request to update the Rules/Smart Contract of a federation	/administration/federation_vote_request/updateFederationRules	IoTFeds User (Data consumer)	Administration	v1.0
SymbIoTeAdminFed-6	REST	Receiving the result of the requests	/administration/federation_vote_request/result	Baas	Administration	v1.0
SymbIoTeAdminUser-1	REST	Invite a user to an existing federation	/administration/user/cpanel/federation_invite	IoTFeds User (Data producer)	Administration	v1.0
SymbIoTeAdminUser-2	REST	Leave from a federation you are currently in	/administration/user/cpanel/leave_federation	IoTFeds User	Administration	v1.0
SymbIoTeAdminUser-3	REST	Delete a federation	/administration/user/cpanel/delete_federation	IoTFeds User	Administration	v1.0
SymbIoTeAdminUser-4	REST	Request for token	/administration/user/login	IoTFeds User	Administration	v1.0
SymbIoTeAdminUser-5	REST	Register a platform to symbIoTe	/administration/user/cpanel/register_platform	IoTFeds User (Data Producer)	Administration	v1.0
SymbIoTeAdminUser-6	REST	Register a new federation to symbiote	/administration/user/cpanel/create_federation	IoTFeds User	Administration	v1.0
SymbIoTeAdminUser-7	REST	Return the list of federations registered in the system	/administration/user/cpanel/list_federations	IoTFeds User	Administration	v1.0

3.1.1.4 Δοκιμή και επαλήθευση (Test cases)

Στην ενότητα αυτή περιγράφονται

Διεπαφή SymbIoTeAdminFed -1. Η διεπαφή αυτή χρησιμοποιείται για το αίτημα δημιουργίας ενός χρήστη.

SymbIoTeAdminFed -1.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{ "accountNonExpired": true,
  "accountNonLocked": true,
  "analyticsAndResearchConsent": true,
  "authorities": [
```

```
{
  "authority": "string"
},
"conditionsAccepted": true,
"credentialsNonExpired": true,
"enabled": true,
"password": "string",
"recoveryMail": "string",
"role": "USER",
"termsAccepted": true,
"username": "user1",
"validPassword": "string",
"validUsername": "string"
}
```

Η επιστροφή του μηνύματος είναι η αντίστοιχη σελίδα του Administration GUI σε περίπτωση επιτυχούς ή μη εγγραφής του χρήστη.

Διεπαφή SymbIoTeAdminFed -2. Η διεπαφή αυτή χρησιμοποιείται για το αίτημα ένταξης ενός οργανισμού σε μία ομοσπονδία.

SymbIoTeAdminFed -2.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{
  "federationId": "Smart City",
  "memberToAdd": "user1"
}
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{
  "federationId": "Smart City",
  "handledDate": "2022-10-17T13:11:30.556Z",
  "requestDate": "2022-10-17T13:11:30.556Z",
  "requestingUser": {
    "accountNonExpired": true,
    "accountNonLocked": true,
    "analyticsAndResearchConsent": true,
    "authorities": [
      {
        "authority": "string"
      }
    ],
    "conditionsAccepted": true,
    "credentialsNonExpired": true,
    "enabled": true,
    "password": "string",
    "recoveryMail": "string",
    "role": "USER",
    "termsAccepted": true,
    "username": "user1",
    "validPassword": "string",
    "validUsername": "string"
  },
  "status": "PENDING",
}
```



```

"username": "string",
"voteAction": "ADD_MEMBER",
"votingId": "string",
"SmartContract": null
}

```

Διεπαφή SymbIoTeAdminFed -3. Η διεπαφή αυτή χρησιμοποιείται για το αίτημα της λίστας των αιτημάτων ενός χρήστη.

SymbIoTeAdminFed -3.A. Δεν απαιτούνται παράμετροι για την κλήση.

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

[[
  {
    "federationId": "Smart City",
    "handledDate": "2022-10-17T13:11:30.556Z",
    "requestDate": "2022-10-17T13:11:30.556Z",
    "requestingUser": {
      "accountNonExpired": true,
      "accountNonLocked": true,
      "analyticsAndResearchConsent": true,
      "authorities": [
        {
          "authority": "string"
        }
      ],
      "conditionsAccepted": true,
      "credentialsNonExpired": true,
      "enabled": true,
      "password": "string",
      "recoveryMail": "string",
      "role": "USER",
      "termsAccepted": true,
      "username": "string",
      "validPassword": "string",
      "validUsername": "string"
    },
    "status": "PENDING",
    "username": "string",
    "voteAction": "ADD_MEMBER",
    "votingId": "string",
    "SmartContract": null
  }
]]

```

Διεπαφή SymbIoTeAdminFed -4. Η διεπαφή αυτή χρησιμοποιείται για το αίτημα διαγραφής ενός οργανισμού από μία ομοσπονδία.

SymbIoTeAdminFed -4.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "federationId": "Smart City",
  "memberToAdd": "user1"
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

{
  "federationId": "Smart City",
  "handledDate": "2022-10-17T13:11:30.556Z",

```

```

"requestDate": "2022-10-17T13:11:30.556Z",
"requestingUser": {
  "accountNonExpired": true,
  "accountNonLocked": true,
  "analyticsAndResearchConsent": true,
  "authorities": [
    {
      "authority": "string"
    }
  ],
  "conditionsAccepted": true,
  "credentialsNonExpired": true,
  "enabled": true,
  "password": "string",
  "recoveryMail": "string",
  "role": "USER",
  "termsAccepted": true,
  "username": "user1",
  "validPassword": "string",
  "validUsername": "string"
},
"status": "PENDING",
"username": "string",
"voteAction": "REMOVE_MEMBER",
"votingId": "string",
"SmartContract": null
}

```

Διεπαφή SymbIoTeAdminFed -5. Η διεπαφή αυτή χρησιμοποιείται για το αίτημα ενημέρωσης των κανόνων μίας ομοσπονδίας.

SymbIoTeAdminFed -5.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "id": "fed_test",
  "name": "fed_test",
  "public": "true",
  "informationModel": {
    "id": "BIM",
    "uri": null,
    "name": null,
    "owner": null,
    "rdf": null,
    "rdfFormat": null
  },
  "members": [
    {
      "platformId": "icom-platform",
      "interworkingServiceURL": null
    }
  ],
  "SmartContract": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "user1",

```

```

    "user2"
  ],
  "Proposals": [
    "ChangeRule",
    "DeleteFed",
    "InviteMember",
    "JoinRequest",
    "RequestRemove"
  ],
  "VoteRules": {
    "Tokens": 50,
    "Type": {
      "ApprovalPercentage": 75,
      "Base": "Board"
    }
  },
  "FedMarketplace": {
    "ChargePolicy": "Free",
    "Coin": "IoTFeds",
    "FedProduct": "Packaging",
    "ProfitPolicy": "PerSource"
  },
  "FedTypeRules": {
    "DataAvailability": "Closed",
    "ServiceType": "Energy, Health, Environment",
    "SupportedOntologies": "Smart Environment, Smart City",
    "Type": "Providers"
  },
  "QualityAssuranceMetrics": {
    "QoSPercentage": 60,
    "Quality": {
      "MinValueFed": 6
    },
    "ReputationPercentage": 40,
    "Underperformance": "RequestRemove"
  }
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

{
  "federationId": "Smart City",
  "handledDate": "2022-10-17T13:11:30.556Z",
  "requestDate": "2022-10-17T13:11:30.556Z",
  "requestingUser": {
    "accountNonExpired": true,
    "accountNonLocked": true,
    "analyticsAndResearchConsent": true,
    "authorities": [
      {
        "authority": "string"
      }
    ]
  },

```

```

    "conditionsAccepted": true,
    "credentialsNonExpired": true,
    "enabled": true,
    "password": "string",
    "recoveryMail": "string",
    "role": "USER",
    "termsAccepted": true,
    "username": "string",
    "validPassword": "string",
    "validUsername": "string"
  },
  "status": "PENDING",
  "username": "string",
  "voteAction": "ADD_MEMBER",
  "votingId": "string",
  "SmartContract": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "user1",
          "user2"
        ],
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      }
    },
    "FedMarketplace": {
      "ChargePolicy": "Free",
      "Coin": "IoTFeds",
      "FedProduct": "Packaging",
      "ProfitPolicy": "PerSource"
    },
    "FedTypeRules": {
      "DataAvailaiability": "Closed",
      "ServiceType": "Energy, Health, Environment",
      "SupportedOntologies": "Smart Environment, Smart City",
      "Type": "Providers"
    },
    "QualityAssuranceMetrics": {
      "QoSPercentage": 60,
      "Quality": {
        "MinValueFed": 6
      },
      "ReputationPercentage": 40,

```

```

    "Underperformance": "RequestRemove"
  }
}
}
}

```

Διεπαφή SymbIoTeAdminFed -6. Η διεπαφή αυτή χρησιμοποιείται για τη λήψη της απάντησης ενός αιτήματος προς ψήφιση.

SymbIoTeAdminFed -6.A. Το τεστ εκτελείται για τη διεπαφή SymbIoTeAdminFed-6. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{ "votingId": "notAvalidVoteId", "status": "ACCEPTED" }
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

Success Message

Διεπαφή SymbIoTeAdminUser-1. Η διεπαφή αυτή χρησιμοποιείται για την πρόσκληση ενός χρήστη σε μία υπάρχουσα ομοσπονδία.

SymbIoTeAdminUser -1.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{
  "federationId": "fed1",
  "memberToAdd": "user1"
}
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{
  "fed1": {
    "id": " fed1",
    "name": " fed1",
    "public": "true",
    "informationModel": {
      "id": "BIM",
      "uri": null,
      "name": null,
      "owner": null,
      "rdf": null,
      "rdfFormat": null
    },
    "members": [
      {
        "platformId": "icom-platform",
        "interworkingServiceURL": null
      }
    ],
    "SmartContract": {
      "IoTFedsRules": {
        "FedGov": {
          "BoardGov": [
            "user1",
            "user2"
          ],
          "Proposals": [
            "ChangeRule",
            "DeleteFed",
            "InviteMember",
            "JoinRequest",

```

```

    "RequestRemove"
  ],
  "VoteRules": {
    "Tokens": 50,
    "Type": {
      "ApprovalPercentage": 75,
      "Base": "Board"
    }
  }
},
"FedMarketplace": {
  "ChargePolicy": "Free",
  "Coin": "IoTFeds",
  "FedProduct": "Packaging",
  "ProfitPolicy": "PerSource"
},
"FedTypeRules": {
  "DataAvailability": "Closed",
  "ServiceType": "Energy, Health, Environment",
  "SupportedOntologies": "Smart Environment, Smart City",
  "Type": "Providers"
},
"QualityAssuranceMetrics": {
  "QoSPercentage": 60,
  "Quality": {
    "MinValueFed": 6
  },
  "ReputationPercentage": 40,
  "Underperformance": "RequestRemove"
}
}
}
}
}
}

```

Διεπαφή SymbIoTeAdminUser-2. Η διεπαφή αυτή χρησιμοποιείται για την αποχώρηση ενός οργανισμού από μια ομοσπονδία.

SymbIoTeAdminUser -2.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "federationId": "fed1",
  "organization": "user1"
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

{
  "fed1": {
    "id": " fed1",
    "name": " fed1",
    "public": "true",
    "informationModel": {
      "id": "BIM",
      "uri": null,
      "name": null,
      "owner": null,

```

```

    "rdf": null,
    "rdfFormat": null
  },
  "members": [
    {
      "platformId": "icom-platform",
      "interworkingServiceURL": null
    }
  ],
  "SmartContract": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "user1",
          "user2"
        ],
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      },
      "FedMarketplace": {
        "ChargePolicy": "Free",
        "Coin": "IoTFeds",
        "FedProduct": "Packaging",
        "ProfitPolicy": "PerSource"
      },
      "FedTypeRules": {
        "DataAvailaiability": "Closed",
        "ServiceType": "Energy, Health, Environment",
        "SupportedOntologies": "Smart Environment, Smart City",
        "Type": "Providers"
      },
      "QualityAssuranceMetrics": {
        "QoSPercentage": 60,
        "Quality": {
          "MinValueFed": 6
        },
        "ReputationPercentage": 40,
        "Underperformance": "RequestRemove"
      }
    }
  }
}

```

Διεπαφή SymbIoTeAdminUser-3. Η διεπαφή αυτή χρησιμοποιείται για τη διαγραφή μίας ομοσπονδίας.

SymbIoTeAdminUser -3.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{
  "federationIdToDelete": "fed1",
}
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{
  "fed1": {
    "id": " fed1",
    "name": " fed1",
    "public": "true",
    "informationModel": {
      "id": "BIM",
      "uri": null,
      "name": null,
      "owner": null,
      "rdf": null,
      "rdfFormat": null
    },
    "members": [
      {
        "platformId": "icom-platform",
        "interworkingServiceURL": null
      }
    ],
    "SmartContract": {
      "IoTFedsRules": {
        "FedGov": {
          "BoardGov": [
            "user1",
            "user2"
          ],
        },
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      },
    },
    "FedMarketplace": {
      "ChargePolicy": "Free",
    },
  },
}
```



```

    "Coin": "IoTFeds",
    "FedProduct": "Packaging",
    "ProfitPolicy": "PerSource"
  },
  "FedTypeRules": {
    "DataAvailability": "Closed",
    "ServiceType": "Energy, Health, Environment",
    "SupportedOntologies": "Smart Environment, Smart City",
    "Type": "Providers"
  },
  "QualityAssuranceMetrics": {
    "QoSPercentage": 60,
    "Quality": {
      "MinValueFed": 6
    },
    "ReputationPercentage": 40,
    "Underperformance": "RequestRemove"
  }
}
}
}
}
}

```

Διεπαφή SymbIoTeAdminUser-4. Η διεπαφή αυτή χρησιμοποιείται για την είσοδο ενός χρήστη στο γραφικό περιβάλλον.

SymbIoTeAdminUser -4.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{username, password }
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{Redirect to main page }
```

Διεπαφή SymbIoTeAdminUser-5. Η διεπαφή αυτή χρησιμοποιείται για την εγγραφή μίας πλατφόρμας.

SymbIoTeAdminUser -5.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{ PlatformDetails platformDetails }
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{ }
```

Διεπαφή SymbIoTeAdminUser-6. Η διεπαφή αυτή χρησιμοποιείται για την εγγραφή μιας ομοσπονδίας.

SymbIoTeAdminUser -6.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "id": "fed_test",
  "name": "fed_test",
  "public": "true",
  "informationModel": {
    "id": "BIM",
    "uri": null,
    "name": null,

```

```

    "owner": null,
    "rdf": null,
    "rdfFormat": null
  },
  "members": [
    {
      "platformId": "icom-platform",
      "interworkingServiceURL": null
    }
  ],
  "SmartContract": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "evathana",
          "gspanos",
          "spolymeni"
        ],
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      },
      "FedMarketplace": {
        "ChargePolicy": "Free",
        "Coin": "IoTFeds",
        "FedProduct": "Packaging",
        "ProfitPolicy": "PerSource"
      },
      "FedTypeRules": {
        "DataAvailaiability": "Closed",
        "ServiceType": "Energy, Health, Environment",
        "SupportedOntologies": "Smart Environment, Smart City",
        "Type": "Providers"
      },
      "QualityAssuranceMetrics": {
        "QoSPercentage": 60,
        "Quality": {
          "MinValueFed": 6
        },
        "ReputationPercentage": 40,
        "Underperformance": "RequestRemove"
      }
    }
  }
}

```

```
}
```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
{ "message" : "Federation Registration was successful!" }
```

Διεπαφή SymbIoTeAdminUser-7. Η διεπαφή αυτή χρησιμοποιείται για την απόκτηση των εγγεγραμμένων ομοσπονδιών στο σύστημα.

SymbIoTeAdminUser -7.A. Δεν απαιτούνται παράμετροι για την κλήση.

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```
[{
  "id": "fed_test",
  "name": "fed_test",
  "public": "true",
  "informationModel": {
    "id": "BIM",
    "uri": null,
    "name": null,
    "owner": null,
    "rdf": null,
    "rdfFormat": null
  },
  "members": [
    {
      "platformId": "icom-platform",
      "interworkingServiceURL": null
    }
  ],
  "SmartContract": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "evathana",
          "gspanos",
          "spolymeni"
        ],
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      }
    },
    "FedMarketplace": {
      "ChargePolicy": "Free",
```

```
"Coin": "IoTFeeds",  
  "FedProduct": "Packaging",  
  "ProfitPolicy": "PerSource"  
},  
"FedTypeRules": {  
  "DataAvailability": "Closed",  
  "ServiceType": "Energy, Health, Environment",  
  "SupportedOntologies": "Smart Environment, Smart City",  
  "Type": "Providers"  
},  
"QualityAssuranceMetrics": {  
  "QoSPercentage": 60,  
  "Quality": {  
    "MinValueFed": 6  
  },  
  "ReputationPercentage": 40,  
  "Underperformance": "RequestRemove"  
}  
}  
}
```

Στον παρακάτω πίνακα συνοψίζονται οι δοκιμές και το αποτέλεσμα τους (Επιτυχές ή μη επιτυχές).

Πίνακας 2: Περίπτωσηι δοκιμής στοιχείου symbolTe's Administration για διαχείριση ομοσπονδίας.

Interface	Test case	Method	Call	Status
SymbIoTeAdminFed-1	SymbIoTeAdminFed-1.A	POST	<iotfeds-url>/administration/register	PASS
SymbIoTeAdminFed-2	SymbIoTeAdminFed-2.A	POST	<iotfeds-url>/administration/federation_vote_request/joinFederation	PASS
SymbIoTeAdminFed-3	SymbIoTeAdminFed-3.A	GET	<iotfeds-url>/administration/federation_vote_request/allForUser	PASS
SymbIoTeAdminFed-4	SymbIoTeAdminFed-4.A	DELETE	<iotfeds-url>/administration/federation_vote_request/removeUserFromFed	PASS
SymbIoTeAdminFed-5	SymbIoTeAdminFed-5.A	POST	<iotfeds-url>/administration/federation_vote_request/updateFederationRules	PASS
SymbIoTeAdminFed-6	SymbIoTeAdminFed-6.A	POST	<iotfeds-url>/administration/federation_vote_request/result	

SymbIoTeAdminUser-1	SymbIoTeAdminUser-1.A	POST	<iotfeds-url>/administration/user/cpanel/federation_invite	PASS
SymbIoTeAdminUser-2	SymbIoTeAdminUser-2.A	POST	<iotfeds-url>/administration/user/cpanel/leave_federation	PASS
SymbIoTeAdminUser-3	SymbIoTeAdminUser-3.A	POST	<iotfeds-url>/administration/user/cpanel/delete_federation	PASS
SymbIoTeAdminUser-4	SymbIoTeAdminUser-4.A	POST	<iotfeds-url>/administration/user/login	PASS
SymbIoTeAdminUser-5	SymbIoTeAdminUser-5.A	POST	<iotfeds-url>/administration/user/cpanel/register_platform	PASS
SymbIoTeAdminUser-6	SymbIoTeAdminUser-6.A	POST	<iotfeds-url>/administration/user/cpanel/create_federation	PASS
SymbIoTeAdminUser-7	SymbIoTeAdminUser-7.A	GET	<iotfeds-url>/administration/user/cpanel/list_federations	PASS

3.1.2 Στοιχείο λογισμικού για τη διαχείριση IoT πόρων (symbIoTe API)

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά, οι διεπαφές και οι δοκιμές του στοιχείου λογισμικού symbIoTeAPI για τη διαχείριση IoT πόρων.

3.1.2.1 Περιγραφή

Το στοιχείο λογισμικού symbIoTe API παρέχει τη δυνατότητα στους χρήστες να αλληλεπιδρούν με το IoTFeds σύστημα μέσω Restful API για την εγγραφή, γ.τροποποίηση/διαγραφή και διαμοιρασμό των IoT πόρων των πλατφορμών καθώς και την πρόσβαση στα δεδομένα τους.

3.1.2.2 Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Τα παρακάτω χαρακτηριστικά προστέθηκαν στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας:

- Εγγραφή και τροποποίηση IoT πόρων – καταγράφει έναν (καθολικό ή ομόσπονδο) IoT πόρο στην IoTFeds πλατφόρμα.
- Διαγραφή IoT πόρων – διαγράφει έναν IoT πόρο από μία ομοσπονδία ή την IoTFeds πλατφόρμα.
- Πρόσβαση στα IoT δεδομένα – επιστρέφει τα δεδομένα ενός IoT πόρου (καθολικού ή ομόσπονδου).
- Αναζήτηση IoT πόρων – αναζητάει τους IoT πόρους με βάση τα κριτήρια αναζήτησης.

3.1.2.3 Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Το symbIoTeAPI είναι προσβάσιμο μέσω Rest Get/Post HTTP αιτήματα στο παρακάτω uri:
symbIoTeAPI-url: <symbIoTe-core>/sybioteapi.

Πίνακας 3: Διεπαφές στοιχείου Administration για τη διαχείριση IoT πόρων.

Inter Face ID	Type	Feature	Interface	Request from	Request to	Release
SymbIoTeAPI-1	REST	Register a resource to symbiote	/symbioteapi/resources/add /11	IoTFeds User (Data Producer)	symbIoTe API	v1.0
SymbIoTeAPI-2	REST	Register a resource to symbiote	/symbioteapi/resources/add /12	IoTFeds User (Data Producer)	symbIoTe API	v1.0
SymbIoTeAPI-3	REST	Share a resource to a federation	/symbioteapi/resources/shareResource	IoTFeds User (Data Producer)	symbIoTe API	v1.0
SymbIoTeAPI-4	REST	Unshare (remove) a resource from a federation	/symbioteapi/resources/unshareResource	IoTFeds User (Data Producer)	symbIoTe API	v1.0
SymbIoTeAPI-5	REST	Delete L1 resource	/symbioteapi/resources/deleteL1	IoTFeds User (Data Producer)	symbIoTe API	v1.0
SymbIoTeAPI-6	REST	Delete L2 resource	/symbioteapi/resources/deleteL2	IoTFeds User (Data Producer)	symbIoTe API	v1.0

3.1.2.4 Δοκιμή και επαλήθευση (Test cases)

Στην ενότητα αυτή παρουσιάζονται οι δοκιμές για τις διεπαφές του στοιχείου λογισμικού symbIoTeAPI για τη διαχείριση IoT πόρων.

Διεπαφή symbIoTeAPI-1. Η διεπαφή αυτή χρησιμοποιείται για την εγγραφή ενός L1 πόρου.

symbIoTeAPI -1.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```
{
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",
    "clientId": "client"
  },
  "cloudResource": {
    "internalId": "vsd",
    "pluginId": "vds",
    "accessPolicy": "CAP",
    "filteringPolicy": "CAP",
    "resource": {
      "id": "resourceId",
      "name": "resourceName",
      "description": [
        "This is",
```

```

        "a description"
      ],
      "interworkingServiceURL": "thisisa.url"
    }
  }
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

{
  "key": [
    {
      "credentials": {
        "platformId": "test_platform",
        "username": "test_user",
        "password": "password",
        "clientId": "client"
      },
      "cloudResource": {
        "internalId": "cloud1",
        "pluginId": "plugin1",
        "accessPolicy": "CAP",
        "filteringPolicy": "CAP",
        "resource": {
          "id": "resourceId",
          "name": "resourceName",
          "description": [
            "This is",
            "a description"
          ],
          "interworkingServiceURL": "thisisa.url"
        }
      }
    }
  ]
}

```

Διεπαφή symbIoTeAPI-2. Η διεπαφή αυτή χρησιμοποιείται για την εγγραφή ενός L2 πόρου.

symbIoTeAPI -2.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",
    "clientId": "client"
  },
  "cloudResource": {
    "internalId": "vds",
    "pluginId": "vds",
    "accessPolicy": "CAP",
    "filteringPolicy": "CAP",
    "resource": {
      "id": "resourceId",
      "name": "resourceName",
      "description": [

```

```

    "This is",
    "a description"
  ],
  "interworkingServiceURL": "thisisa.url"
}
}
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

[
  {
    "cloudResource": {
      "internalId": " cloud1",
      "pluginId": " plugin1",
      "accessPolicy": "CAP",
      "filteringPolicy": "CAP",
      "resource": {
        "id": "resourceId",
        "name": "resourceName",
        "description": [
          "This is",
          "a description"
        ],
        "interworkingServiceURL": "thisisa.url"
      }
    }
  }
]

```

Διεπαφή symbIoTeAPI-3. Η διεπαφή αυτή χρησιμοποιείται για κοινοποίηση ενός πόρου σε μία ομοσπονδία.

symbIoTeAPI -3.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",
    "clientId": "client"
  },
  "resourceInternalId": "resourceId",
  "bartered": true,
  "federationId": "fed1"
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

[
  {
    "cloudResource": {
      "internalId": " cloud1",
      "pluginId": " plugin1",
      "accessPolicy": "CAP",
      "filteringPolicy": "CAP",
      "resource": {
        "id": "resourceId",
        "name": "resourceName",
        "description": [

```



```

    "This is",
    "a description"
  ],
  "interworkingServiceURL": "thisisa.url"
}
}
}
]

```

Διεπαφή symbIoTeAPI-4. Η διεπαφή αυτή χρησιμοποιείται για την άρση κοινοποίησης ενός πόρου σε ομοσπονδία.

symbIoTeAPI -4.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",
    "clientId": "client"
  },
  "resourceInternalId": "resourceId",
  "bartered": false,
  "federationId": "fed1"
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

[
  {
    "cloudResource": {
      "internalId": " cloud1",
      "pluginId": " plugin1",
      "accessPolicy": "CAP",
      "filteringPolicy": "CAP",
      "resource": {
        "id": "resourceId",
        "name": "resourceName",
        "description": [
          "This is",
          "a description"
        ],
        "interworkingServiceURL": "thisisa.url"
      }
    }
  }
]

```

Διεπαφή symbIoTeAPI-5. Η διεπαφή αυτή χρησιμοποιείται για τη διαγραφή ενός L1 πόρου.

symbIoTeAPI -5.A. Το τεστ εκτελείται για τη διεπαφή SymbIoTeAPI-5. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "resourceInternalId": "resourceId",
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",

```

```

    "clientId": "client"
  }
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

```

{
  "cloudResource": {
    "internalId": " cloud1",
    "pluginId": " plugin1",
    "accessPolicy": "CAP",
    "filteringPolicy": "CAP",
    "resource": {
      "id": "resourceId",
      "name": "resourceName",
      "description": [
        "This is",
        "a description"
      ],
      "interworkingServiceURL": "thisisa.url"
    }
  },
  "errors": ""
}

```

Διεπαφή symbIoTeAPI-6. Η διεπαφή αυτή χρησιμοποιείται για τη διαγραφή ενός L2 πόρου.

symbIoTeAPI -6.A. Το ακόλουθο σώμα για την κλήση θα πρέπει να οριστεί:

```

{
  "resourceInternalId": "resourceId",
  "credentials": {
    "platformId": "test_platform",
    "username": "test_user",
    "password": "password",
    "clientId": "client"
  }
}

```

Το επιθυμητό αποτέλεσμα φαίνεται παρακάτω:

Το κείμενο της επιστροφής του συγκεκριμένου αιτήματος είναι κενό.

Στον παρακάτω πίνακα συνοψίζονται οι περιπτώσεις δοκιμών και το αποτέλεσμα τους (επιτυχές ή μη) για το στοιχείο λογισμικού symbIoTeAPI.

Πίνακας 4:Περίπτωσης δοκιμής στοιχείου symbIoTe-API.

Interface	Test case	Method	Call	Status
SymbIoTeAPI-1	SymbIoTeAPI-1.A	POST	<iotfeds-url>/symbioteapi/resources/add/11	PASS
SymbIoTeAPI-2	SymbIoTeAPI-2.A	POST	<iotfeds-url>/symbioteapi/resources/add/12	PASS

SymbIoTeA PI-3	SymbIoTeA PI-3.A	POST	<iotfeds-url>/sym-bioteapi/re-sources/shareResource	PASS
SymbIoTeA PI-4	SymbIoTeA PI-4.A	POST	<iotfeds-url>/sym-bioteapi/re-sources/unshareResource	PASS
SymbIoTeA PI-5	SymbIoTeA PI-5.A	DELETE	<iotfeds-url>/sym-bioteapi/re-sources/deleteL1	PASS
SymbIoTeA PI-6	SymbIoTeA PI-6.A	DELETE	<iotfeds-url>/sym-bioteapi/re-sources/deleteL2	PASS

3.2 Υποσύστημα BaaS

URL	
GitLab repository	https://colab-repo.intracom-telecom.com/colab-projects/iotfeds/baas/baas
Container registry	fabric-ca , fabric-peer , fabric-orderer
Deployment yaml file	docker-compose-ca.yaml , docker-compose-test-net.yaml

3.2.1 Στοιχείο λογισμικού για τη διαχείριση χρηστών

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά, οι διεπαφές και οι περιπτώσεις δοκιμών για το στοιχείο λογισμικού BaaS για τη διαχείριση χρηστών.

3.2.1.1 Περιγραφή

Το στοιχείο λογισμικού BaaS για τη διαχείριση χρηστών παρέχει τη δυνατότητα στους χρήστες να εγγράφονται στο blockchain και να καταχωρούν τις πλατφόρμες και συσκευές που έχουν εγγράψει στο symbIoTe. Για τις παραπάνω διεργασίες προσφέρονται και οι αντίστροφες λειτουργίες της αντίστοιχης διαγραφής. Επιπλέον προσφέρεται η δυνατότητα ενημέρωσης του υπολοίπου ενός χρήστη, και πρόσβασης σε πληροφορίες του χρήστη.

3.2.1.2 Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Τα παρακάτω χαρακτηριστικά προστέθηκαν στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας:

- registerUserToBc – εγγραφή ενός χρήστη στο BaaS
- registerPlatforms – εγγραφή μιας πλατφόρμας σε έναν χρήστη στο BaaS
- registerDevice – εγγραφή μιας συσκευής σε μια πλατφόρμα χρήστη στο BaaS
- getUserInfoU – πρόσβαση σε στοιχεία ενός χρήστη στο BaaS
- getAllUsers – πρόσβαση σε στοιχεία όλων των χρηστών στο BaaS
- updateUserBalance – ενημέρωση υπολοίπου χρήστη στο BaaS
- removePlatform – Αφαίρεση πλατφόρμας χρήστη από το BaaS
- removePlatformResources – Αφαίρεση πόρων πλατφόρμας χρήστη από το BaaS
- removeDevice – Αφαίρεση συσκευής χρήστη από το BaaS
- deleteUser – Αφαίρεση χρήστη από το BaaS

3.2.1.3 Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Το BaaS API για διαχείριση χρηστών είναι προσβάσιμο μέσω Rest Get/Post/Patch/Delete HTTP αιτήματα στο παρακάτω uri:

BaaS-API-url-user: http://160.40.50.169:8800/baas/user

Πίνακας 5: Διεπαφές στοιχείου BaaS για την διαχείριση χρηστών.

Inter Face ID	Type	Feature	Endpoint /Queue	Request from	Request to	Release
BaaSAPI-1	REST	registerUser ToBc	POST < BaaS-API-url-user l>/register_user	Symbiote	BaaS	v1.0
BaaSAPI-2	REST	regiterPlatform	POST < BaaS-API-url-user l>/register_platform	Symbiote	BaaS	v1.0
BaaSAPI-3	REST	registerDevice	POST < BaaS-API-url-user l>/register_device	Symbiote	BaaS	v1.0
BaaSAPI-4	REST	getUserInfo	GET < BaaS-API-url-user l>/get_user_info	Symbiote	BaaS	v1.0
BaaSAPI-5	REST	getAllUsers	GET < BaaS-API-url-user l>/get_all_users	Symbiote	BaaS	v1.0
BaaSAPI-6	REST	updateUser Balance	PATCH < BaaS-API-url-user l>/update_user_balance	Symbiote	BaaS	v1.0
BaaSAPI-7	REST	removePlatform	DELETE < BaaS-API-url-user l>/remove_platform	Symbiote	BaaS	v1.0
BaaSAPI-8	REST	removePlatformResources	DELETE < BaaS-API-url-user l>/remove_platform_resource	Symbiote	BaaS	v1.0
BaaSAPI-9	REST	removeDevice	DELETE < BaaS-API-url-user l>/remove_device	Symbiote	BaaS	v1.0
BaaSAPI-10	REST	deleteUser	DELETE < BaaS-API-url-user l>/delete_user	Symbiote	BaaS	v1.0

3.2.1.4 Δοκιμή και επαλήθευση (Test cases)

Στην ενότητα αυτή περιγράφονται οι περιπτώσεις δοκιμών του στοιχείου λογισμικού BaaS για τη διαχείριση χρηστών.

Διεπαφή BaaSAPI-1. Χρησιμοποιείται για την εγγραφή ενός χρήστη στο BaaS.

BaaSAPI-1.A. Αυτός ο έλεγχος επαληθεύει ότι ένας χρήστης γράφεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```

{
  ... "id": "grapadop",
  ... "role": "researcher",
  ... "mail": "grapadop@iti.gr",
  ... "organization": "CERTH"
}

```

Εικόνα 6 Παράμετροι στο σώμα του ελέγχου BaaSAPI-1.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!»

BaaSAPI-1.B. Αυτός ο έλεγχος επαληθεύει ότι ένας χρήστης δε μπορεί να δίπλο-εγγραφεί στο σύστημα, οι παράμετροι που πρέπει να οριστούν στο σώμα του αιτήματος είναι ίδιοι με τον προηγούμενο έλεγχο. Το αναμενόμενο αποτέλεσμα είναι status: 403 με μήνυμα που περιέχει «The user grapadop already exists...»

Διεπαφή BaaSAPI-2. Χρησιμοποιείται για την εγγραφή μιας πλατφόρμας ενός χρήστη στο BaaS.

BaaSAPI-2.A. Αυτός ο έλεγχος επαληθεύει ότι μια πλατφόρμα γράφεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```

{
  ... "platform_id": "vicinity",
  ... "assoc_user_id": "grapadop"
}

```

Εικόνα 7 Παράμετροι στο σώμα του ελέγχου BaaSAPI-2.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!»

BaaSAPI-2.B. Αυτός ο έλεγχος επαληθεύει ότι μια πλατφόρμα δε μπορεί να δίπλο-εγγραφεί στον ίδιο χρήστη, οι παράμετροι που πρέπει να οριστούν στο σώμα του αιτήματος είναι ίδιο με τον προηγούμενο έλεγχο. Το αναμενόμενο αποτέλεσμα είναι status: 400 με μήνυμα που περιέχει «Update failed...» και «The platform vicinity is already registered»

Διεπαφή BaaSAPI-3. Χρησιμοποιείται για την εγγραφή μιας συσκευής σε μια πλατφόρμα ενός χρήστη στο BaaS.

BaaSAPI-3.A. Αυτός ο έλεγχος επαληθεύει ότι μια συσκευή γράφεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```

{
  ... "platform_id": "vicinity",
  ... "device_id": "device1"
}

```

Εικόνα 8 Παράμετροι στο σώμα του ελέγχου BaaSAPI-3.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!»

BaaSAPI-3.B. Αυτός ο έλεγχος επαληθεύει ότι μια συσκευή δε μπορεί να δίπλο-εγγραφεί σε μια πλατφόρμα, οι παράμετροι που πρέπει να οριστούν στο σώμα του αιτήματος είναι ίδιες με τον προηγούμενο έλεγχο. Το αναμενόμενο αποτέλεσμα είναι status: 400 με μήνυμα που περιέχει «Registration failed» και «The device device1 is already registered...»

Διεπαφή BaaSAPI-4. Χρησιμοποιείται για την πρόσβαση σε πληροφορίες ενός χρήστη στο BaaS.

BaaSAPI-4.A. Αυτός ο έλεγχος επαληθεύει ότι οι πληροφορίες ενός χρήστη επιστρέφονται με επιτυχία, εφόσον το αναγνωριστικό του χρήστη περνιέται ως παράμετρος στο url του αιτήματος. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 9.

```
{
  "AssociatedPlatforms": {
    "vicinity": []
  },
  "Balance": 0,
  "ID": "gpapadop",
  "Mail": "gpapadop@iti.gr",
  "Organization": "CERTH",
  "Role": "researcher",
  "docType": "user"
}
```

Εικόνα 9 Επιστροφή πληροφοριών χρήστη απο το BaaS.

BaaSAPI-4.B. Αυτός ο έλεγχος επαληθεύει ότι επιστρέφεται το κατάλληλο μήνυμα για χρήστη που δεν υπάρχει, εφόσον το αναγνωριστικό του χρήστη περνιέται ως παράμετρος στο url του αιτήματος. Το αναμενόμενο αποτέλεσμα είναι status 403 και μήνυμα που περιέχει «Access Failed...» και «The user does not exist».

Διεπαφή BaaSAPI-5. Χρησιμοποιείται για την πρόσβαση πληροφοριών όλων των χρηστών στο BaaS.

BaaSAPI-5.A. Αυτός ο έλεγχος επαληθεύει ότι οι πληροφορίες όλων των χρηστών επιστρέφονται με επιτυχία. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 10.

```
[{"AssociatedPlatforms":{"vicinity":["device1"]},"Balance":0,"ID":"evathana","Mail":"evathana@iti.gr","Organization":"Dimos Thermis","Role":"Research associate","docType":"user"},{"AssociatedPlatforms":{"vicinity":[]},"Balance":0,"ID":"gpapadop","Mail":"gpapadop@iti.gr","Organization":"CERTH","Role":"researcher","docType":"user"},{"AssociatedPlatforms":{"vicinity":[]},"Balance":0,"ID":"gspanos","Mail":"gspanos@iti.gr","Organization":"CERTH","Role":"Postdoctoral Researcher","docType":"user"},{"AssociatedPlatforms":{"vicinity":[]},"Balance":0,"ID":"spolymeni","Mail":"spolymeni@iti.gr","Organization":"Dimos Kalamarias","Role":"Research associate","docType":"user"}]
```

Εικόνα 10 Επιστροφή πληροφοριών όλων των χρηστών από το BaaS.

Διεπαφή BaaSAPI-6. Χρησιμοποιείται για την ανανέωση υπολοίπου ενός χρήστη στο BaaS

BaaSAPI-6.A. Αυτός ο έλεγχος επαληθεύει ότι η ανανέωση υπολοίπου ενός χρήστη γίνεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```
{
  "id": "gpapadop",
  "transaction_fee": 300
}
```

Εικόνα 11 Παράμετροι στο σώμα του ελέγχου BaaSAPI-6.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» και σε εκτέλεση του ελέγχου BaaSAPI-4 για τον χρήστη αυτό πρέπει να φαίνεται η ενημέρωση του υπολοίπου του.

BaaSAPI-6.B. Αυτός ο έλεγχος επαληθεύει ότι η ανανέωση υπολοίπου ενός χρήστη που δεν υπάρχει δεν είναι εφικτή, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```
{
  "id": "gpapado2",
  "transaction_fee": 300
}
```

Εικόνα 12 Παράμετροι στο σώμα του ελέγχου BaaSAPI-6.B.

Το αναμενόμενο αποτέλεσμα είναι status: 403 με μήνυμα που περιέχει «Update failed...» και «The user gpadop2 does not exist».

Διεπαφή BaaSAPI-7. Χρησιμοποιείται για την διαγραφή πλατφόρμας ενός χρήστη στο BaaS

BaaSAPI-7.A. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή πλατφόρμας ενός χρήστη γίνεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```
{
  ... "platform_id": "vicinity",
  ... "assoc_user_id": "gpadop"
}
```

Εικόνα 13 Παράμετροι στο σώμα του ελέγχου BaaSAPI-7.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» και σε εκτέλεση του ελέγχου BaaSAPI-4 για τον χρήστη αυτόν δεν πρέπει να φαίνεται η διαγραμμένη πλατφόρμα.

BaaSAPI-7.B. Αυτός ο έλεγχος επαληθεύει εάν η προς διαγραφή πλατφόρμα ενός χρήστη δεν υπάρχει, το σύστημα ενημερώνει κατάλληλα, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```
1 {
2   ... "platform_id": "false_platform",
3   ... "assoc_user_id": "gpadop"
4 }
```

Εικόνα 14 Παράμετροι στο σώμα του ελέγχου BaaSAPI-7.B.

Το αναμενόμενο αποτέλεσμα είναι status: 404 με μήνυμα «Deletion failed, platform does not exist...».

Διεπαφή BaaSAPI-8. Χρησιμοποιείται για την διαγραφή όλων των πόρων μιας πλατφόρμας ενός χρήστη στο BaaS.

BaaSAPI-8.A. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή όλων των πόρων μιας πλατφόρμας ενός χρήστη γίνεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```
{
  ... "platform_id": "vicinity"
}
```

Εικόνα 15 Παράμετροι στο σώμα του ελέγχου BaaSAPI-8.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» και σε εκτέλεση του ελέγχου BaaSAPI-4 για τον χρήστη αυτόν δεν πρέπει να φαίνεται η πλατφόρμα αλλά τα περιεχόμενα της να είναι κενά.

BaaSAPI-8.B. Αυτός ο έλεγχος επαληθεύει ότι εάν η πλατφόρμα της οποίας οι πόροι είναι προς διαγραφή είναι κενή, το σύστημα ενημερώνει κατάλληλα, οι παράμετροι είναι ίδιοι με το προηγούμενο αίτημα ωστόσο αυτή τη φορά η πλατφόρμα είναι κενή από την προηγούμενη εκτέλεση. Το αναμενόμενο αποτέλεσμα είναι status: 404 με μήνυμα «Deletion failed, resources not found in platform».

Διεπαφή BaaSAPI-9. Χρησιμοποιείται για την διαγραφή ενός πόρου από μια πλατφόρμα ενός χρήστη στο BaaS

BaaSAPI-9.A. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή πόρου από πλατφόρμα ενός χρήστη γίνεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```

{
  "platform_id": "vicinity",
  "device_id": "device1"
}

```

Εικόνα 16 Παράμετροι στο σώμα του ελέγχου BaaSAPI-9.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» και σε εκτέλεση του ελέγχου BaaSAPI-4 για τον χρήστη αυτόν δεν πρέπει να φαίνεται η συσκευή μέσα στην αντίστοιχη πλατφόρμα.

BaaSAPI-9.B. Αυτός ο έλεγχος επαληθεύει ότι εάν η προς διαγραφή συσκευή δεν υπάρχει στην αντίστοιχη πλατφόρμα, το σύστημα ενημερώνει κατάλληλα. Οι παράμετροι φαίνονται στην Εικόνα 17.

```

1 {
2   "platform_id": "vicinity",
3   "device_id": "device2"
4 }

```

Εικόνα 17 Παράμετροι στο σώμα του ελέγχου BaaSAPI-9.B.

Το αναμενόμενο αποτέλεσμα είναι status: 404 με μήνυμα «Deletion failed, device not found in platform».

Διεπαφή BaaSAPI-10. Χρησιμοποιείται για την διαγραφή ενός χρήστη στο BaaS

BaaSAPI-10.A. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή ενός χρήστη γίνεται με επιτυχία, οι παρακάτω παράμετροι πρέπει να οριστούν στο σώμα του αιτήματος:

```

{
  "user_id": "gpapadop"
}

```

Εικόνα 18 Παράμετροι στο σώμα του ελέγχου BaaSAPI-10.A.

Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» καθώς και το αντικείμενο που περιγράφει τον διαγραφμένο χρήστη όπως στην Εικόνα 19 . Σε εκτέλεση του ελέγχου BaaSAPI-5 δεν πρέπει να φαίνεται ο χρήστης αυτός.

```

{
  "AssociatedPlatforms": {
    "vicinity": []
  },
  "Balance": 0,
  "ID": "gpapadop",
  "Mail": "gpapadop@iti.gr",
  "Organization": "CERTH",
  "Role": "researcher",
  "docType": "user"
}

```

Εικόνα 19 Αντικείμενο διαγραφμένου χρήστη στο BaaS.

BaaSAPI-10.B. Αυτός ο έλεγχος επαληθεύει ότι εάν ο προς διαγραφή χρήστης δεν υπάρχει, το σύστημα ενημερώνει κατάλληλα. Οι παράμετροι φαίνονται παρακάτω.


```
{
  ... "user_id": "unreal_user"
}
```

Εικόνα 20 Παράμετροι στο σώμα του ελέγχου BaaSAPI-10.B.

Το αναμενόμενο αποτέλεσμα είναι status: 400 με μήνυμα που περιέχει «Deletion failed ...» και «The user unreal_user does not exist».

Πίνακας 6: Περίπτωσης δοκιμής στοιχείου BaaS για την διαχείριση χρηστών.

Interface	Test case	Method	Call	Status
BaaSAPI-I-1	BaaSAPI-1.A	POST	<BaaS-API-URL-user> /register_user	PASS
	BaaSAPI-1.B	POST	<BaaS-API-URL-user> / register_user	PASS
BaaSAPI-I-2	BaaSAPI-2.A	POST	<BaaS-API-URL-user> /register_platform	PASS
	BaaSAPI-2.B	POST	<BaaS-API-URL-user> /register_platform	PASS
BaaSAPI-I-3	BaaSAPI-3.A	POST	<BaaS-API-URL-user> /register_device	PASS
	BaaSAPI-3.B	POST	<BaaS-API-URL-user> /register_device	PASS
BaaSAPI-I-4	BaaSAPI-4.A.	GET	<BaaS-API-URL-user> /get_user_info?user_id =gpapadop	PASS
	BaaSAPI-4.A.	GET	<BaaS-API-URL-user> /get_user_info?user_id =gpapadop2	PASS
BaaSAPI-I-5	BaaSAPI-5.A	GET	<BaaS-API-URL-user> /get_all_user_info	PASS
BaaSAPI-I-6	BaaSAPI-6.A	PATCH	<BaaS-API-URL-user> /update_user_balance	PASS
	BaaSAPI-6.B	PATCH	<BaaS-API-URL-user> /update_user_balance	PASS
BaaSAPI-I-7	BaaSAPI-7.A	DELETE	<BaaS-API-URL-user> /remove_platform	PASS
	BaaSAPI-7.B	DELETE	<BaaS-API-URL-user> /remove_platform	PASS
BaaSAPI-I-8	BaaSAPI-8.A	DELETE	<BaaS-API-URL-user> /remove_platform_resources	PASS
	BaaSAPI-8.B	DELETE	<BaaS-API-URL-user> /remove_platform_resources	PASS
BaaSAPI-I-9	BaaSAPI-9.A	DELETE	<BaaS-API-URL-user> /remove_device	PASS
	BaaSAPI-9.B	DELETE	<BaaS-API-URL-user> /remove_device	PASS
BaaSAPI-I-10	BaaSAPI-10.A	DELETE	<BaaS-API-URL-user> /delete_user	PASS
	BaaSAPI-10.B	DELETE	<BaaS-API-URL-user> /delete_user	PASS

3.2.2 Στοιχείο λογισμικού για τη διαχείριση ομοσπονδιών

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά, οι διεπαφές και οι περιπτώσεις δοκιμών για το στοιχείο λογισμικού BaaS για τη διαχείριση ομοσπονδιών.

3.2.2.1 Περιγραφή

Το στοιχείο λογισμικού BaaS για τη διαχείριση ομοσπονδιών παρέχει τη δυνατότητα εγγραφής ομοσπονδιών, πρόσβασης σε πληροφορίες αυτών, καθώς και αποχώρησης μελών από αυτές ή εξ ολοκλήρου διαγραφής τους. Επίσης, δίνεται σε χρήστες η δυνατότητα αιτήματος για πρόσκληση ή αφαίρεση μέλους καθώς και αλλαγή κανόνων ομοσπονδίας. Τα αιτήματα αυτά εκκινούν διαδικασίες ψηφοφορίας στις ομοσπονδίες. Για να υποστηριχτούν αυτές οι ψηφοφορίες παρέχεται επιπλέον γραφικό περιβάλλον ώστε ο εκάστοτε ψηφοφόρος να έχει πρόσβαση στην ψηφοφορία και να μπορεί να ψηφίσει.

3.2.2.2 Χαρακτηριστικά που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Τα παρακάτω χαρακτηριστικά προστέθηκαν στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας:

- registerFedToBc – δημιουργεί την εγγραφή μιας ομοσπονδίας στο BaaS
- getFedInfo – πρόσβαση σε πληροφορίες ομοσπονδίας στο BaaS
- getAllFeds – πρόσβαση σε πληροφορίες όλων των ομοσπονδιών στο BaaS
- leaveFed – αποχώρηση χρήστη από ομοσπονδία στο BaaS
- deleteFederation – διαγραφή ομοσπονδίας από το BaaS
- addFedMemberRequest – αίτημα προσθήκης μέλους σε ομοσπονδία στο BaaS
- removeFedMemberRequest – αίτημα αφαίρεσης μέλους από ομοσπονδία στο BaaS
- updateFedRules – αίτημα αλλαγής κανόνων σε ομοσπονδία στο BaaS
- getVotingDescription – πρόσβαση σε πληροφορίες ψηφοφορίας στο BaaS
- registerVote – καταγραφή ψήφου στο BaaS

3.2.2.3 Διεπαφές που συμπεριλαμβάνονται στην τρέχουσα έκδοση του ολοκληρωμένου συστήματος

Το BaaS API για διαχείριση ομοσπονδιών είναι προσβάσιμο μέσω Rest Get/Post/Patch/Delete HTTP αιτήματα στο παρακάτω uri:

BaaS-API-url-fed: <http://160.40.50.169:8800/baas/federation>.

Πίνακας 7: Διεπαφές στοιχείου BaaS.

Inter Face ID	Type	Feature	Endpoint /Queue	Request from	Request to	Release
BaaSAPI-11	REST	registerFedToBc	POST <BaaS-API-url-fed>/register_fed	sybmIoTe	BaaS	v1.0
BaaSAPI-12	REST	getFedInfo	GET <BaaS-API-url-fed>/get_fed_info	sybmIoTe	BaaS	v1.0
BaaSAPI-13	REST	getAllFeds	GET <BaaS-API-url-fed>/get_all_fed_info	sybmIoTe	BaaS	v1.0
BaaSAPI-14	REST	leaveFed	PATCH <BaaS-API-url-fed>/leave_fed	sybmIoTe	BaaS	v1.0
BaaSAPI-15	REST	deleteFederation	DELETE <BaaS-API-url-fed>/delete_fed	sybmIoTe	BaaS	v1.0
BaaSAPI-16	REST	addFedMemberRequest	POST <BaaS-API-url-fed>/add_fed_member_request	sybmIoTe	BaaS	v1.0
BaaSAPI-17	REST	removeFedMemberRequest	POST <BaaS-API-url-fed>/remove_fed_member_request	sybmIoTe	BaaS	v1.0
BaaSAPI-18	REST	updateFedRules	POST <BaaS-API-url-fed>/update_fed_rules_request	sybmIoTe	BaaS	v1.0
BaaSAPI-19	REST	getVotingDescription	GET <BaaS-API-url-fed>/get_voting_description	BaaS	ioTFeds user	v1.0
BaaSAPI-20	REST	registerVote	POST <BaaS-API-url-fed>/register_vote	IoTFeds user	BaaS	v1.0

3.2.2.4 Δοκιμή και επαλήθευση (Test cases)

Στην ενότητα αυτή περιγράφονται οι περιπτώσεις δοκιμών για τις διεπαφές που περιγράφηκαν παραπάνω.

Διεπαφή BaaSAPI-11. Χρησιμοποιείται για την εγγραφή μιας ομοσπονδίας στο BaaS.

BaaSAPI -11.A. Αυτός ο έλεγχος χρησιμοποιείται για την επαλήθευση της επιτυχούς εγγραφής μιας ομοσπονδίας στο BaaS. Οι παράμετροι που περνιούνται στο σώμα φαίνονται στην Εικόνα 21. Το αναμενόμενο αποτέλεσμα είναι status 200 με μήνυμα «OK...»

```

{
  "fed_id": "fed4",
  "creator_id": "gpapadop",
  "inf_model": "BIM",
  "related_applications": ["platform1"],
  "rules": {"first": "1st rule"}
}

```

Εικόνα 21: Παράμετροι στο σώμα του ελέγχου BaaSAPI-11.A.

BaaSAPI-11.B. Αυτός ο έλεγχος επαληθεύει ότι μια ομοσπονδία δε μπορεί να δίπλο-εγγραφεί στο BaaS, οι παράμετροι που πρέπει να οριστούν στο σώμα του αιτήματος είναι ίδιες με τον προηγούμενο έλεγχο. Το αναμενόμενο αποτέλεσμα είναι status: 403 με μήνυμα που περιέχει «Registration failed...» και «The federation with id fed4 already exists»

Διεπαφή BaaSAPI-12. Χρησιμοποιείται για την πρόσβαση σε πληροφορίες μίας ομοσπονδίας στο BaaS.

BaaSAPI-12.A. Αυτός ο έλεγχος επαληθεύει ότι οι πληροφορίες μιας ομοσπονδίας επιστρέφονται με επιτυχία, εφόσον το αναγνωριστικό της ομοσπονδίας και του δημιουργού της περνούν ως παράμετροι στο «url» του αιτήματος. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 22.

```
{
  "ID": "fed4",
  "creator_id": "gpapadop",
  "docType": "fed",
  "inf_model": "BIM",
  "members_ids": [
    "gpapadop"
  ],
  "related_applications": [
    "platform1"
  ],
  "rules": {
    "first": "1st rule"
  }
}
```

Εικόνα 22: Επιστροφή πληροφοριών ομοσπονδίας από το BaaS.

BaaSAPI-12.B. Αυτός ο έλεγχος επαληθεύει ότι επιστρέφεται το κατάλληλο μήνυμα για ομοσπονδία που δεν υπάρχει, εφόσον το αναγνωριστικό της ομοσπονδίας και του δημιουργού περνιέται ως παράμετρος στο «url» του αιτήματος. Το αναμενόμενο αποτέλεσμα είναι status 403 και μήνυμα «Access Failed...» και «id does not exist».

Διεπαφή BaaSAPI-13. Χρησιμοποιείται για την πρόσβαση πληροφοριών όλων των ομοσπονδιών στο BaaS.

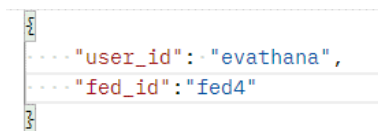
BaaSAPI-13.A. Αυτός ο έλεγχος επαληθεύει ότι οι πληροφορίες όλων των ομοσπονδιών επιστρέφονται με επιτυχία. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 23.

```
{
  "ID": "fed1", "creator_id": "evathana", "docType": "fed", "inf_model": "BIM", "members_ids": ["evathana", "gsanos", "spolymeri"], "related_applications": [
    "platform1"
  ], "rules": {
    "first": "1st rule"
  }
}, {
  "ID": "fed2", "creator_id": "spolymeri", "docType": "fed", "inf_model": "BIM", "members_ids": ["spolymeri"], "related_applications": [
    "platform1"
  ], "rules": {
    "first": "1st rule"
  }
}, {
  "ID": "fed3", "creator_id": "gsanos", "docType": "fed", "inf_model": "BIM", "members_ids": ["gsanos"], "related_applications": [
    "platform1"
  ], "rules": {
    "first": "1st rule"
  }
}, {
  "ID": "fed4", "creator_id": "evathana", "docType": "fed", "inf_model": "BIM", "members_ids": ["evathana"], "related_applications": [
    "platform1"
  ], "rules": {
    "first": "1st rule"
  }
}
```

Εικόνα 23: Επιστροφή πληροφοριών όλων των ομοσπονδιών από το BaaS.

Διεπαφή BaaSAPI-14. Χρησιμοποιείται για την αποχώρηση μέλους από ομοσπονδία στο BaaS. Για τους ελέγχους αυτής της διεπαφής δημιουργείται ομοσπονδία με id fed4 από τον χρήστη gpapadop στην οποία προσθέτουμε τον χρήστη evathana.

BaaSAPI-14.A. Αυτός ο έλεγχος επαληθεύει ότι μέλος μιας ομοσπονδίας μπορεί να αποχωρήσει με επιτυχία. Οι παράμετροι που περνιούνται στο σώμα είναι αυτοί στην Εικόνα 24. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 22.



```
{
  "user_id": "evathana",
  "fed_id": "fed4"
}
```

Εικόνα 24 Παράμετροι στο σώμα του ελέγχου BaaSAPI-14.A.

BaaSAPI-14.B. Αυτός ο έλεγχος επαληθεύει ότι δε μπορεί να αφαιρεθεί από ομοσπονδία χρήστη που δεν υπάρχει σε αυτήν. Το σώμα του ελέγχου είναι ίδιο με αυτό του προηγούμενου ελέγχου, όπου αυτή τη φορά ο προς αφαίρεσή χρήστη δεν υπάρχει στην ομοσπονδία ως μέλος. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα που περιέχει «User leaving failed ...» και «The user is not registered in federation fed4»

BaaSAPI-14.Γ. Αυτός ο έλεγχος επαληθεύει ότι δε μπορεί να αποχωρήσει από μια ομοσπονδία ο δημιουργός της. Οι παράμετροι που περνιούνται στο σώμα είναι αυτοί στην Εικόνα 25. Το αναμενόμενο αποτέλεσμα είναι status 400 και μήνυμα που περιέχει «User leaving failed» και «The creator of the federation is not allowed to leave it».



```
{
  "user_id": "gpapadop",
  "fed_id": "fed4"
}
```

Εικόνα 25: Παράμετροι στο σώμα του ελέγχου BaaSAPI-14.Γ.

Διεπαφή BaaSAPI-15. Χρησιμοποιείται για την διαγραφή μιας ομοσπονδίας στο BaaS.

BaaSAPI-15.A. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή μιας ομοσπονδίας δεν μπορεί να γίνει παρά μόνο από τον δημιουργό της. Οι παράμετροι είναι αυτές στην Εικόνα 24. Το αναμενόμενο αποτέλεσμα είναι status: 400 με μήνυμα που περιέχει «Deletion failed ...» και «Only the creator of the federation can delete it».

BaaSAPI-15.B. Αυτός ο έλεγχος επαληθεύει ότι εάν η ομοσπονδία περιέχει μέλη δε μπορεί να διαγραφεί. Εγγράφουμε στην ομοσπονδία fed4 τον χρήστη evathana και καλούμε την διεπαφή με τις παραμέτρους στην Εικόνα 25. Το αναμενόμενο αποτέλεσμα είναι μήνυμα που περιέχει «Deletion failed» και «This federation still contains members and cannot be deleted».

BaaSAPI-15.Γ. Αυτός ο έλεγχος επαληθεύει ότι η διαγραφή μιας ομοσπονδίας γίνεται με επιτυχία, οι παράμετροι που πρέπει να οριστούν στο σώμα του αιτήματος είναι ίδιες με την Εικόνα 25 αφού διαγράψουμε τον χρήστη evathana από την ομοσπονδία. Το αναμενόμενο αποτέλεσμα είναι status: 200 με μήνυμα «OK!» καθώς και το αντικείμενο που περιγράφει την διαγραμμένη ομοσπονδία όπως Εικόνα 22. Σε εκτέλεση του ελέγχου BaaSAPI-12 δεν πρέπει να φαίνεται η ομοσπονδία αυτή.

BaaSAPI-15.Δ. Αυτός ο έλεγχος επαληθεύει ότι εάν η προς διαγραφή ομοσπονδία δεν υπάρχει, το σύστημα ενημερώνει κατάλληλα. Οι παράμετροι είναι ξανά ίδιοι με αυτές στην Εικόνα 24. Το αναμενόμενο αποτέλεσμα είναι status: 400 με μήνυμα που περιέχει «Deletion failed ...» και «id is not defined».

Διεπαφή BaaSAPI-16. Χρησιμοποιείται για την έναρξη ψηφοφορίας για την προσθήκη νέου μέλους σε ομοσπονδία στο BaaS. Η πραγματοποίηση των ελέγχων γίνεται στην ομοσπονδία «fed2» με μοναδικό χρήστη «spolymeni».

BaaSAPI-16.A. Αυτός ο έλεγχος επαληθεύει ότι δεν δύναται η έναρξη ψηφοφορίας για προσθήκη χρήστη που υπάρχει ήδη στην ομοσπονδία. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 26. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα που περιέχει «Add new federation member request failed ...» και «This member is already a member of this federation».

```
{
  "fed_id": "fed2",
  "requestor_id": "spolymeni",
  "member_id": "spolymeni"
}
```

Εικόνα 26: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.A.

BaaSAPI-16.B. Αυτός ο έλεγχος επαληθεύει ότι η έναρξη ψηφοφορίας για προσθήκη χρήστη σε ομοσπονδία, ο οποίος είναι ήδη εγγεγραμμένος στο BaaS λειτουργεί κανονικά. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 27. Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 28, με πληροφορίες της ψηφοφορίας.

```
{
  "fed_id": "fed2",
  "requestor_id": "spolymeni",
  "member_id": "gspanos"
}
```

Εικόνα 27 Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.B.

```
{
  "ID": "1igowujn4c33icchr38twu",
  "docType": "voting",
  "federation": "fed2",
  "descr": {
    "votingType": "addition",
    "requestorID": "Dimos Kalamarias",
    "memberID": "CERTH",
    "fedID": "fed2"
  },
  "status": true,
  "startDate": 1673274890022,
  "endDate": 1673361290022,
  "votes": {
    "spolymeni": "pending"
  }
}
```

Εικόνα 28: Αντικείμενο που επιστρέφεται από τον έλεγχο BaaSAPI-16.B.

BaaSAPI-16.Γ. Αυτός ο έλεγχος επαληθεύει ότι δεν δύναται η έναρξη δεύτερης ψηφοφορίας για μια ομοσπονδία πριν λήξει η πρώτη. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 29 Εικόνα 26. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα που περιέχει «Add new federation member request failed ...» και «There is already an ongoing voting for this federation».

```

{
  "fed_id": "fed2",
  "requestor_id": "spolymeni",
  "member_id": "evathana"
}

```

Εικόνα 29: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.Γ.

Διεπαφή BaaSAPI-17. Χρησιμοποιείται για την έναρξη ψηφοφορίας για την αφαίρεση μέλους από ομοσπονδία στο BaaS. Η πραγματοποίηση των ελέγχων γίνεται στην ομοσπονδία «fed1» με χρήστες «spolymeni», «evathana» και «gsranos».

BaaSAPI-17.A. Αυτός ο έλεγχος επαληθεύει ότι η έναρξη ψηφοφορίας προτεινόμενη από μέλος της, για αφαίρεση χρήστη, ο οποίος είναι ήδη μέλος της ομοσπονδίας λειτουργεί κανονικά. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 30 όπου ο χρήστης «spolymeni» προτείνει την αφαίρεση του χρήστη «gsranos». Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 31, με πληροφορίες της ψηφοφορίας, όπως φαίνεται το αντικείμενο αυτό που επιστρέφεται, δεν εκθέτει τα IDs των οργανισμών αλλά μόνο τα ονόματά τους.

```

{
  "fed_id": "fed1",
  "requestor_id": "spolymeni",
  "user_id": "gsranos"
}

```

Εικόνα 30: Παράμετροι στο σώμα του ελέγχου BaaSAPI-16.A.

```

{
  "ID": "vksqymf8cbhvyq5wk9xosk",
  "docType": "voting",
  "federation": "fed1",
  "descr": {
    "votingType": "removal",
    "requestorID": "Dimos Kalamarias",
    "memberID": "CERTH",
    "fedID": "fed1"
  },
  "status": true,
  "startDate": 1673879446220,
  "endDate": 1673965846220,
  "votes": {
    "evathana": "pending",
    "spolymeni": "pending"
  }
}

```

Εικόνα 31: Αντικείμενο που επιστρέφεται από τον έλεγχο BaaSAPI-17.A.

BaaSAPI-17.B. Αυτός ο έλεγχος επαληθεύει ότι δεν δύναται η έναρξη ψηφοφορίας για αφαίρεση μέλους που δεν υπάρχει ήδη στην ομοσπονδία. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 32. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα « Request to remove a member from the federation failed » και «This user is not a member of this federation».

```

    ... "fed_id": "fed1",
    ... "requestor_id": "spolymeni",
    ... "user_id": "gropadap"
  }

```

Εικόνα 32: Παράμετροι στο σώμα του ελέγχου BaaSAPI-17.B.

Διεπαφή BaaSAPI-18. Χρησιμοποιείται για την έναρξη ψηφοφορίας για την αλλαγή κανόνων σε ομοσπονδία στο BaaS. Η πραγματοποίηση των ελέγχων γίνεται στην ομοσπονδία «fed1» με αρχικό σετ κανόνων αυτό στην Εικόνα 33.

```

"rules": {
  "IoTfedsRules": {
    "FedGov": {
      "BoardGov": [
        "evathana",
        "gspanos",
        "spolymeni"
      ],
      "Proposals": [
        "ChangeRule",
        "DeleteFed",
        "InviteMember",
        "JoinRequest",
        "RequestRemove"
      ],
      "VoteRules": {
        "Tokens": 50,
        "Type": {
          "ApprovalPercentage": 75,
          "Base": "Board"
        }
      }
    },
    "FedMarketplace": {
      "ChargePolicy": "Free",
      "Coin": "IoTfeds",
      "FedProduct": "Packaging",
      "ProfitPolicy": "PerSource"
    },
    "FedTypeRules": {
      "DataAvailability": "Closed",
      "ServiceType": "Energy, Health, Environment",
      "SupportedOntologies": "Smart Environment, Smart City",
      "Type": "Providers"
    },
    "QualityAssuranceMetrics": {
      "QoSPercentage": 60,
      "Quality": 5,
      "MinValueFed": 0,
      "ReputationPercentage": 40,
      "Underperformance": "RequestRemove"
    }
  }
}

```

Εικόνα 33: Αρχικό σετ κανόνων της ομοσπονδίας που χρησιμοποιείται για τους ελέγχους της διεπαφής BaaSAPI-18.

BaaSAPI-18.A. Αυτός ο έλεγχος επαληθεύει ότι δεν δύναται η έναρξη ψηφοφορίας για προτεινόμενους κανόνες ίδιους με τους υπάρχοντες. Οι παράμετροι στο σώμα του ελέγχου είναι αυτοί στην Εικόνα 34. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα που περιέχει «Request to update federation rules failed ...» και «Proposed rules are same as existing ones».


```

{
  "fed_id": "fed1",
  "requestor_id": "evathana",
  "new_rules": {
    "IoTFedsRules": {
      "FedGov": {
        "BoardGov": [
          "evathana",
          "gspanos",
          "spolymeni"
        ],
        "Proposals": [
          "ChangeRule",
          "DeleteFed",
          "InviteMember",
          "JoinRequest",
          "RequestRemove"
        ],
        "VoteRules": {
          "Tokens": 50,
          "Type": {
            "ApprovalPercentage": 75,
            "Base": "Board"
          }
        }
      },
      "FedMarketplace": {
        "ChargePolicy": "Free",
        "Coin": "IoTFeds",
        "FedProduct": "Packaging",
        "ProfitPolicy": "PerSource"
      },
      "FedTypeRules": {
        "DataAvailability": "Closed",
        "ServiceType": "Energy, Health, Environment",
        "SupportedOntologies": "Smart Environment, Smart City",
        "Type": "Providers"
      },
      "QualityAssuranceMetrics": {
        "QoSPercentage": 60,
        "Quality": {
          "MinValueFed": 6
        },
        "ReputationPercentage": 40,
        "Underperformance": "RequestRemove"
      }
    }
  }
}

```

Εικόνα 34: Παράμετροι στο σώμα του ελέγχου BaaSAPI-18.A.

BaaSAPI-18.B. Αυτός ο έλεγχος επαληθεύει ότι η έναρξη ψηφοφορίας για αλλαγή κανόνων σε ομοσπονδία, λειτουργεί κανονικά. Μπορούμε να το δοκιμάσουμε προτείνοντας αλλαγή στο πεδίο "BoardGov" αφαιρώντας τον χρήστη «spolymeni». Το αναμενόμενο αποτέλεσμα είναι status 200 και αντικείμενο σαν αυτό στην Εικόνα 35, με πληροφορίες της ψηφοφορίας.

```

{
  "ID": "n2nfb8iaf7je3w73zadky",
  "docType": "voting",
  "federation": "fed1",
  "descr": {
    "votingType": "rule change",
    "requestorID": "Dimos Thermis",
    "proposedRules": {
      "changedField": "BoardGov",
      "proposed": [
        "evathana",
        "gspanos"
      ],
      "current": [
        "evathana",
        "gspanos",
        "spolymeni"
      ]
    }
  },
  "fedID": "fed1"
},
{
  "status": true,
  "startDate": 1673880787931,
  "endDate": 1673967187931,
  "votes": {
    "evathana": "pending",
    "gspanos": "pending",
    "spolymeni": "pending"
  }
}

```

Εικόνα 35: Αναμενόμενο αποτέλεσμα επιστροφής του ελέγχου BaaSAPI-18.B.

Διεπαφή BaaSAPI-19. Χρησιμοποιείται για την πρόσβαση σε πληροφορίες μιας ψηφοφορίας από έναν χρήστη.

BaaSAPI-19.A. Αυτός ο έλεγχος επαληθεύει ότι πληροφορίες για μια ψηφοφορία είναι προσβάσιμες από έναν χρήστη με το token που εκδίδεται για αυτόν από τα endpoints που εκκινούν την διαδικασία ψηφοφορίας. Το token αυτό παρέχεται σε κάθε μέλος της ομοσπονδίας που έχει δικαίωμα ψήφου (BoardGov) και είναι ένα Json Web Token (JWT) που περνιέται ως παράμετρος στο url του αιτήματος και εφόσον είναι έγκυρο επιστρέφεται αντικείμενο με περιγραφή της ψηφοφορίας. Μετά την ψήφο του χρήστη το token αυτό εξαργυρώνεται. Για παράδειγμα για την ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B. το αντικείμενο θα είναι αυτό στην Εικόνα 36. Αξίζει να σημειωθεί πως στην κανονική ροή του συστήματος, η κλήση αυτής της διεπαφής γίνεται αυτόματα κατά το κλικ του συνδέσμου που λαμβάνει ο χρήστης στο e-mail του προκειμένου να ψηφίσει.

```

{
  "fedID": "fed1",
  "proposedRules": {
    "changedField": "BoardGov",
    "current": [
      "evathana",
      "gspanos",
      "spolymeni"
    ],
    "proposed": [
      "evathana",
      "gspanos"
    ]
  },
  "requestorID": "Dimos Thermis",
  "votingType": "rule change"
}

```

Εικόνα 36: Αντικείμενο που περιγράφει την ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B.

BaaSAPI-19.B. Αυτός ο έλεγχος επαληθεύει ότι η διεπαφή δε μπορεί να δώσει αποτέλεσμα αν το token δεν είναι έγκυρο. Καλούμε τη διεπαφή χωρίς να θέσουμε τιμή στην παράμετρο url του token. Το αναμενόμενο αποτέλεσμα είναι status 403 με μήνυμα «Invalid token».

Διεπαφή BaaSAPI- 20. Χρησιμοποιείται για την καταχώρηση ψήφου ενός χρήστη σε μια ψηφοφορία

BaaSAPI-20.A. Αυτός ο έλεγχος επαληθεύει ότι ένας χρήστης έχει δικαίωμα ψήφου σε μια ψηφοφορία μέσω του token του. Για παράδειγμα για την ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B. για θετική ψήφο του χρήστη «evathana», οι παράμετροι θα είναι αυτοί στην Εικόνα 37. Αξίζει να σημειωθεί πως στην κανονική ροή του συστήματος, η κλήση αυτής της διεπαφής γίνεται αυτόματα κατά το κλικ του πλήκτρου submit (μετά την επιλογή ψήφου) στην σελίδα που βλέπει ο χρήστης εφόσον επαληθευτεί το token του.

```

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRHRZvdGluZyI6Im4ybWZ1OG1hZjdqZTN3NzN6YWRreSIsIklEdm90ZXIiOiJldmF0aGFuYSIsIm1hdCI6MTY3Mzg4MDc5MCwiZXhwIjoxNjc3OTY3MTkwfQ.NdU6Wimr1gKuk00noDX-nzm7iFb0Uad2PW6qLL8AdU4",
  "vote": "yes"
}

```

Εικόνα 37: Παράμετροι στο σώμα του ελέγχου BaaSAPI-20.A για καταχώρηση ψήφου στην ψηφοφορία που εκκίνησε στον έλεγχο BaaSAPI-18.B.

BaaSAPI-20.B. Αυτός ο έλεγχος επαληθεύει ότι ένας χρήστης δεν μπορεί να ψηφίσει δυο φορές για την ίδια ψηφοφορία. Οι παράμετροι στο σώμα είναι ίδιες με αυτές του προηγούμενου ελέγχου. Το αναμενόμενο αποτέλεσμα είναι status 400 με μήνυμα που περιέχει «Register vote of user failed» και «Voter evathana has already voted and cannot vote again».

Πίνακας 8: Περίπτωσης δοκιμής στοιχείου BaaS για τη διαχείριση ομοσπονδιών.

Interface	Test case	Method	Call	Status
BaaSAPI-I-11	BaaSAPI-11.A	POST	<BaaS-API-url-fed>/register_fed	PASS
	BaaSAPI-11.B	POST	<BaaS-API-url-fed>/register_fed	PASS
BaaSAPI-I-12	BaaSAPI-12.A	GET	<BaaS-API-url-fed>/get_fed_info?creator_id=evathana&fed_id=fed4	PASS
	BaaSAPI-12.B	GET	<BaaS-API-url-fed>/get_fed_info?creator_id=evathana&fed_id=fed6	PASS
BaaSAPI-I-13	BaaSAPI-13.A	GET	<BaaS-API-url-fed>/get_all_fed_info	PASS
BaaSAPI-I-14	BaaSAPI-14.A	PATCH	<BaaS-API-url-fed>/leave_fed	PASS
	BaaSAPI-14.B	PATCH	<BaaS-API-url-fed>/leave_fed	PASS
	BaaSAPI-14.Γ	PATCH	<BaaS-API-url-fed>/leave_fed	PASS
BaaSAPI-I-15	BaaSAPI-15.A	DELETE	<BaaS-API-url-fed>/delete_fed/	PASS
	BaaSAPI-15.B	DELETE	<BaaS-API-url-fed>/delete_fed/	PASS
	BaaSAPI-15.Γ	DELETE	<BaaS-API-url-fed>/delete_fed/	PASS
	BaaSAPI-15.Δ	DELETE	<BaaS-API-url-fed>/delete_fed/	PASS
BaaSAPI-I-16	BaaSAPI-16.A	POST	<BaaS-API-url-fed>/new_member_request	PASS
	BaaSAPI-16.B	POST	<BaaS-API-url-fed>/new_member_request	PASS
	BaaSAPI-16.B	POST	<BaaS-API-url-fed>/new_member_request	PASS
BaaSAPI-I-17	BaaSAPI-17.A	POST	<BaaS-API-url-fed>/remove_member_request	PASS
	BaaSAPI-17.B	POST	<BaaS-API-url-fed>/remove_member_request	PASS
BaaSAPI-I-18	BaaSAPI-18.A	POST	<BaaS-API-url-fed>/update_fed_rules_request	PASS
	BaaSAPI-18.B	POST	<BaaS-API-url-fed>/update_fed_rules_request	PASS
BaaSAPI-I-19	BaaSAPI-19.A	GET	<BaaS-API-url-fed>/get_voting_description	PASS
	BaaSAPI-19.B	GET	<BaaS-API-url-fed>/get_voting_description	PASS
BaaSAPI-I-20	BaaSAPI-20.A	POST	<BaaS-API-url-fed>/voting/vote	PASS
	BaaSAPI-20.B	POST	<BaaS-API-url-fed>/voting/vote	PASS

4 Λειτουργική δοκιμή και επίδειξη συστήματος

Στην ενότητα αυτή παρουσιάζεται η αλληλεπίδραση του χρήστη με την πλατφόρμα του IoTFeds για την διεκπεραίωση αιτημάτων του χρήστη μέσω της γραφικής διεπαφής του συστήματος. Τα αιτήματα που υποστηρίζονται αφορούν τις παρακάτω λειτουργικότητες:

- Εγγραφή ενός χρήστη στην IoTFeds πλατφόρμα.
- Σύνδεση ενός χρήστη στη γραφική διεπαφή χρήστη.
- Εγγραφή μίας IoT πλατφόρμας.
- Δημιουργία IoT ομοσπονδίας.
- Τροποποίηση κανόνων ομοσπονδίας.
- Προσθήκη νέου μέλους σε υπάρχουσα ομοσπονδία.
- Αίτηση για συμμετοχή σε υπάρχουσα ομοσπονδία.
- Απομάκρυνση μέλους από ομοσπονδία.
- Αποχώρηση από ομοσπονδία.
- Διαγραφή ομοσπονδίας.
- Διαγραφή IoT πλατφόρμας.
- Διαγραφή χρήστη.

Περίπτωση Δοκιμής

Κατάσταση
Εκτέλεσης

AdminGUI-1: Πρόσβαση στον ιστότοπο της IoTFeds πλατφόρμας.

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ο χρήστης έχει πρόσβαση στον ιστότοπο της IoTFeds πλατφόρμας.

Ο χρήστης επισκέπτεται την κεντρική σελίδα του ιστότοπου της IoTFeds πλατφόρμας.

Προαπαιτούμενο:

Γνώση της διεύθυνσης του ιστότοπου της IoTFeds πλατφόρμας:

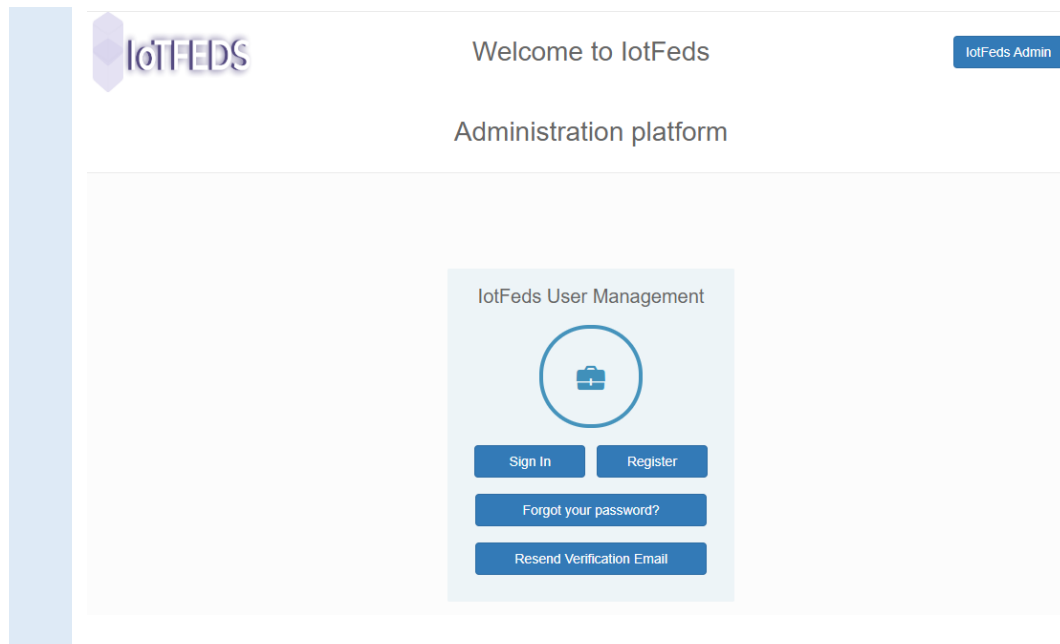
<https://iotfeds-symbiotecore.intracom-telecom.com/administration/>.

Αναμενόμενα αποτελέσματα:

Η αρχική σελίδα με το κεντρικό μενού της IoTFeds πλατφόρμας εμφανίζεται στην οθόνη του χρήστη.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στον χρήστη.

**AdminGUI-2: Εγγραφή χρήστη στην IoTFeds πλατφόρμα.****PASS**

Η δοκιμή αυτή εξασφαλίζει ότι ένας χρήστης μπορεί να εγγραφεί επιτυχώς στην IoTFeds πλατφόρμα.

Ο χρήστης κάνει εγγραφή στην κεντρική σελίδα του ιστότοπου της IoTFeds πλατφόρμας.

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-1

Αναμενόμενα αποτελέσματα:

Η φόρμα για τη συμπλήρωση των υποχρεωτικών στοιχείων του χρήστη εμφανίζεται όταν ο χρήστης πατήσει από το κεντρικό μενού της περίπτωσης δοκιμής AdminGUI-1 το κουμπί “Register”.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στην οθόνη του χρήστη πατώντας το κουμπί “Register”.

Registration

Username

We use the username for uniquely identify each user in the system

Password

Email

We use your email address as part of allowing access to your account, and in order to contact you with important information about any changes to your account

Choose your User Role

Organization

We use the Organization for uniquely identify each user in the system

lot feds role

Terms and Conditions

Data protection is of a particularly high priority for the management of the symbloTe. The use of the Internet pages of the symbloTe is possible without any indication of personal data; however, if a data subject wants to use special enterprise services via our website, processing of personal data could become necessary. If the processing of personal data is necessary and there is no statutory basis for such processing, we generally obtain consent from the data subject.

The processing of personal data, such as the name, address, e-mail address, or telephone number of a data subject shall always be in line with the General Data Protection Regulation (GDPR), and in accordance with the country-specific data protection

☐ I understand the Terms and Conditions

☐ I accept the Term and Conditions

Permissions

• Username

• Email

• Public Keys, bound to the user clients

• JWT tokens, issued for clients

☐ Please, specify if the data above can be used for analysis and research purposes

Data Breach Policy

1. Introduction

1. This policy sets out the policies and procedure of symbloTe (the "consortium") with respect to GDPR Art. 33 Notification of a personal data breach to the supervisory authority and Art. 34 Communication of a personal data breach to the data subject.

2. When dealing with personal data breaches, the consortium and all consortium members must focus on protecting individuals and their personal data, as well as protecting the interests of the consortium.

2. Definition

Ο χρήστης συμπληρώνει τα απαιτούμενα στοιχεία ορθώς.

Αναμενόμενα αποτελέσματα:

Η εγγραφή του χρήστη ολοκληρώνεται με επιτυχία εφόσον συμπληρώνονται όλα τα απαιτούμενα στοιχεία. Στη δοκιμή αυτή το κουμπί “Register” της φόρμας συμπλήρωσης παραμένει ανενεργό μέχρι ο χρήστης συμπληρώσει όλα τα υποχρεωτικά στοιχεία και αποδεχτεί τους όρους και προϋποθέσεις της IoTFeds πλατφόρμας.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στην οθόνη του χρήστη στην περίπτωση επιτυχούς εγγραφής του στην IoTFeds πλατφόρμα:

Registration Successful!

Please, login with your new credentials

Login

Ο χρήστης προσπαθεί να εγγραφεί στην IoTFeds πλατφόρμα επιλέγοντας ένα όνομα χρήστη (user name) που ήδη υπάρχει στην πλατφόρμα.

Αναμενόμενα αποτελέσματα:
Ο χρήστης ενημερώνεται ότι το όνομα χρήστη (user name) που έχει εισάγει στην φόρμα εγγραφής έχει ήδη καταχωρηθεί από άλλον χρήστη της IoTFeds πλατφόρμας.

Αποτελέσματα που παρατηρήθηκαν:
Το παρακάτω μήνυμα λάθους εμφανίζεται στην οθόνη του χρήστη.

Registration

An error occurred: Username exists!

iotfedsdc1

✓

We use the username for uniquely identify each user in the system

✓

iotfedsdc1@icom.com

✓

We use your email address as part of allowing access to your account, and in order to contact you with important information about any changes to your account

User

✕

icom

✓

We use the Organization for uniquely identify each user in the system

organization

✓

AdminGUI-3: Σύνδεση χρήστη στην IoTFeds πλατφόρμα.

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ένας χρήστης μπορεί να συνδεθεί επιτυχώς στην IoTFeds πλατφόρμα.

Ο χρήστης επιθυμεί πρόσβαση στην κεντρική σελίδα του ιστότοπου της IoTFeds πλατφόρμας.

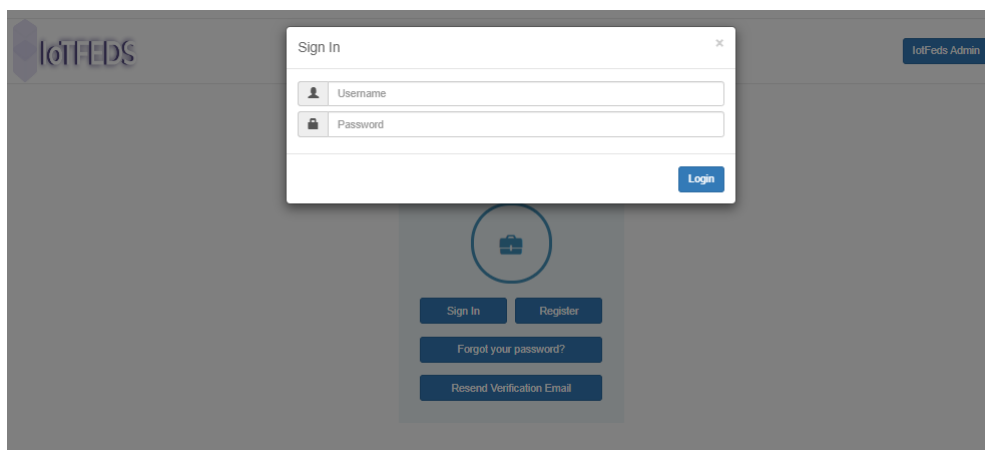
Προαπαιτούμενη περίπτωση δοκιμής:
AdminGUI-2

Αναμενόμενα αποτελέσματα:

Η φόρμα για τη συμπλήρωση των διαπιστευτηρίων του χρήστη εμφανίζεται όταν ο χρήστης πατήσει το κουμπί “Sign in” από το κεντρικό μενού.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στην οθόνη του χρήστη.



The image shows the IoTFeds login interface. A modal window titled 'Sign In' is centered, containing fields for 'Username' and 'Password', and a 'Login' button. In the background, there are buttons for 'Sign In', 'Register', 'Forgot your password?', and 'Resend Verification Email'. The top right corner has a 'IoTFeds Admin' link.

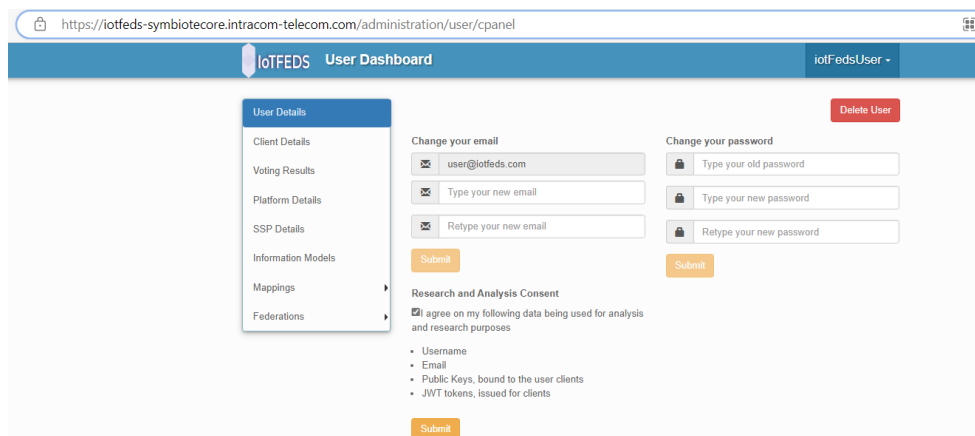
Ο χρήστης εισάγει τα διαπιστευτήρια του στην φόρμα και πατάει το κουμπί “Login”.

Αναμενόμενα αποτελέσματα:

Η σύνδεση του χρήστη ολοκληρώνεται με επιτυχία σε περίπτωση εισαγωγής σωστών διαπιστευτηρίων εμφανίζοντας στην οθόνη την ιστοσελίδα User Dashboard που αφορά αποκλειστικά τις δραστηριότητες του χρήστη στην IoTFeds πλατφόρμα.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στον χρήστη.



The image shows the 'User Dashboard' of the IoTFeds platform. The browser address bar shows the URL: <https://iotfeds-symbiotecore.intracom-telecom.com/administration/user/cpanel>. The dashboard has a blue header with the 'IoTFeds User Dashboard' title and a 'Delete User' button. A left sidebar contains a menu with options: 'User Details', 'Client Details', 'Voting Results', 'Platform Details', 'SSP Details', 'Information Models', 'Mappings', and 'Federations'. The main content area includes sections for 'Change your email' (with fields for current, new, and retype email, and a 'Submit' button), 'Change your password' (with fields for old, new, and retype password, and a 'Submit' button), and 'Research and Analysis Consent' (with a checkbox for agreement and a 'Submit' button). A list of items is shown below the consent section: Username, Email, Public Keys, bound to the user clients, and JWT tokens, issued for clients.

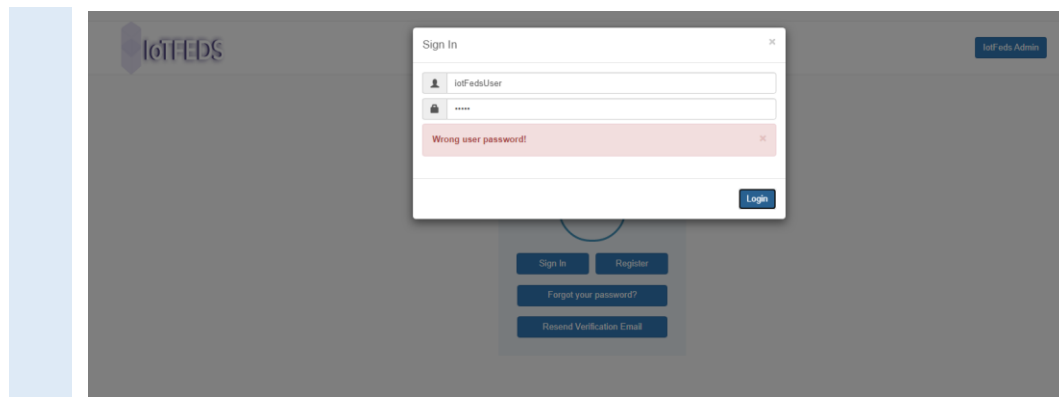
Ο χρήστης εισάγει λανθασμένα διαπιστευτήρια.

Αναμενόμενα αποτελέσματα:

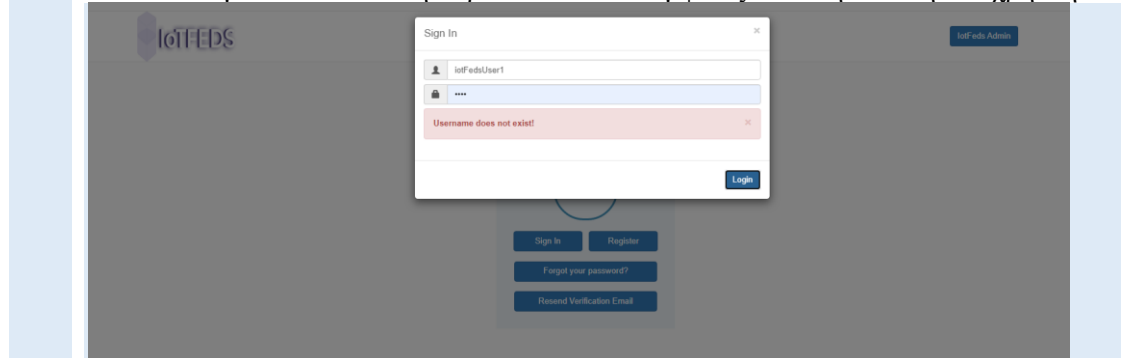
Σε περίπτωση λανθασμένων στοιχείων όπως κωδικό ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στον χρήστη.



Για λανθασμένο user name η παρακάτω σελίδα εμφανίζεται στην οθόνη του χρήστη:



AdminGUI-4: Εγγραφή IoT πλατφόρμας στο IoTFeds σύστημα.

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ένας εγγεγραμμένος χρήστης μπορεί να εγγράψει μία IoT πλατφόρμα ιδιοκτησίας του στο IoTFeds σύστημα.

Ο χρήστης ζητάει να κάνει εγγραφή της IoT πλατφόρμας μέσα από το μενού “Platform Details”.

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-3 (Σύνδεση χρήστη στην IoTFeds πλατφόρμα με ρόλο “Service Owner”)

Αναμενόμενα αποτελέσματα:

Η φόρμα για τη συμπλήρωση των στοιχείων της IoT πλατφόρμας εμφανίζεται στην οθόνη όταν ο χρήστης πατάει το κουμπί “Register New Platform”.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στο χρήστη για να εισάγει τα στοιχεία της IoT πλατφόρμας του.

The screenshot shows the 'IoT FEDS User Dashboard' with a sidebar menu containing: User Details, Client Details, Voting Results, Platform Details (selected), SSP Details, Information Models, Mappings, and Federations. A 'Platform Registration' modal is open, containing the following fields:

- Platform Id:** 'Enter preferred platform id'. Note: 'From 4 to 30 characters. Include only letters, digits, '-' and '_'. You can leave it empty for autogeneration.'
- Platform Name:** 'Enter the platform name'. Note: 'From 3 to 30 characters'.
- Platform Descriptions:** 'Enter the platform description'. Note: 'From 4 to 300 characters'.
- Interworking Services:** 'Enter a valid https url' and a dropdown for 'Information Model'.
- Type:** A dropdown menu currently showing 'Platform'. Note: 'Select your Platform type'.

Buttons at the bottom of the modal are 'Submit' and 'Close'.

Η εγγραφή της IoT πλατφόρμας επιτυγχάνεται όταν τα στοιχεία είναι ορθά.

Τα στοιχεία που εισάγει ο εγγεγραμμένος χρήστης είναι το όνομα της IoT πλατφόρμας (platform name), μια μοναδική ταυτότητα της πλατφόρμας (platform id), μια σύντομη περιγραφή της πλατφόρμας (Platform Description), τον τύπο της IoT πλατφόρμας (Platform ή Enabler type), το μοντέλο οντολογίας το οποίο χρησιμοποιείται από την IoT πλατφόρμα (Information Model) και την μοναδική διεύθυνση προσπέλασης (http://, https://) της IoT πλατφόρμας (Interworking Services).

Η επιτυχημένη εγγραφή της IoT πλατφόρμας στο IoT FEDS σύστημα εμφανίζει στην οθόνη του χρήστη το μήνυμα: “Registration of platform ... was successful!”. Μετά την επιτυχή εγγραφή της πλατφόρμας, ο χρήστης μπορεί να κατεβάσει τα αρχεία παραμέτρων για την διαμόρφωση και την εγκατάσταση του symbIoTe-cloud που περιγράφηκε στο Κεφάλαιο 2.3.2.2 πατώντας το κουμπί «Get Configuration». Στη συνέχεια ο χρήστης μπορεί να εγγράψει τους IoT πόρους που επιθυμεί να διαμοιράσει στο σύστημα.

The screenshot shows the 'IoT FEDS User Dashboard' with the same sidebar menu. The main area is titled 'Platform list' and displays a green success message: 'Registration of platform "test-platform" was successful!'. Below the message is a button 'Register New Platform'. A table lists the registered platform:

Platform Name	Actions
test-platform	Get Configuration, Update, Delete

Σε περίπτωση λανθασμένων στοιχείων το σύστημα επιστρέφει το αντίστοιχο μήνυμα λάθους στον χρήστη.

Στο παρακάτω παράδειγμα αποτυγχάνει η εγγραφή IoT πλατφόρμας με τιμή Interworking Services ή platform Id που ήδη έχουν καταχωρηθεί στην IotFeds πλατφόρμα.

The image displays two screenshots of the IoT FEDS User Dashboard, specifically the Platform Registration form. The dashboard has a sidebar with navigation links: User Details, Client Details, Voting Results, Platform Details (selected), SSP Details, Information Models, Mappings, and Federations. The top header shows 'IoT FEDS User Dashboard' and the user 'iotFedsUser'.

Top Screenshot: The Platform Registration form is shown with a red error message at the top: "AAM threw CommunicationException: Service gateway address already in use". The form fields are: Platform Id (test-platform1), Platform Name (test-platform), Platform Descriptions (test-platform), Interworking Services (https://iotfeds-symbiotecloud.intracom-telecom.cor), and Type (Platform). The form is titled "Platform Registration".

Bottom Screenshot: The Platform Registration form is shown with a red error message at the top: "AAM threw CommunicationException: Service with this instance id already exists". The form fields are: Platform Id (test-platform), Platform Name (test-platform), Platform Descriptions (test-platform), Interworking Services (https://iotfeds-symbiotecloud.intracom-telecom.cor), and Type (Platform). The form is titled "Platform Registration".

AdminGUI-5: Δημιουργία ομοσπονδίας.**PASS**

Η δοκιμή αυτή εξασφαλίζει ότι ένας χρήστης (με ρόλο Service Owner) μπορεί να δημιουργήσει μία ομοσπονδία στο IoTFeds σύστημα.

- 1 Ο χρήστης ζητάει να κάνει εγγραφή μία ομοσπονδίας μέσα από την επιλογή "Federation List" του μενού "Federations".

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-3 (Σύνδεση χρήστη με ρόλο “Service Owner”).

AdminGUI-4 (Εγγραφή IoT πλατφόρμας στο IoTFeds σύστημα).

Αναμενόμενα αποτελέσματα:

Η φόρμα για τη συμπλήρωση των στοιχείων της ομοσπονδίας εμφανίζεται όταν ο χρήστης πατάει το κουμπί “Register New Federation”.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στην οθόνη του χρήστη.

The screenshot shows the 'Federation Registration' form with the following sections:

- Federation Id**: Text input for the federation ID (4-30 characters).
- Federation Name**: Text input for the federation name (3-30 characters).
- Information Model**: Dropdown menu for selecting the federation information model.
- Public**: Dropdown menu for 'Yes' or 'No' (Has the federation public visibility?).
- Federation Members**: Text input for the organization name (4-30 characters).
- QoS Constraints**: Section for defining Quality of Service constraints with fields for Metric Type, Comparator, Threshold, Duration, and Resource Type.
- Federation government rules**: Section for defining rules, including:
 - Board government**: Text input for adding board government members.
 - Insert the supported proposals**: Text input for adding supported proposals.
 - Vote Rules Base**: Dropdown menu for selecting the vote rules base.
 - Tokens**: Text input for inserting the token.
 - Vote Rules Approval %**: Dropdown menu for selecting the approval percent.
- Quality assurance**: Section for defining quality assurance metrics, including:
 - Qos %**: Text input for inserting the qos %.
 - Reputation %**: Text input for inserting the reputation %.
 - Min Number of Federations**: Text input for inserting the minimum value fed.
 - Under Performance**: Dropdown menu for selecting the Under performance value.
- Federation rule types**: Section for defining rule types, including:
 - Rule Type**: Dropdown menu for selecting the type of rules.
 - Data Availability**: Dropdown menu for selecting the data availability.
 - Service Type**: Text input for inserting the type of service.
 - Supported Ontologies**: Text input for inserting the ontology.
- Marketplace policy**: Section for defining marketplace policies, including:
 - Charge Policy**: Dropdown menu for selecting the Charge Policy.
 - Federation Product**: Dropdown menu for selecting the supported Federation Product.
 - Profit Policy**: Dropdown menu for selecting the supported Profit Policy.
 - Coin**: Dropdown menu for selecting the supported Coin.

At the bottom right, there are 'Submit' and 'Close' buttons.

Ο χρήστης συμπληρώνει και υποβάλλει τη φόρμα.

Ο χρήστης συμπληρώνει όλα τα υποχρεωτικά πεδία της φόρμας και πατάει το κουμπί “Submit” το οποίο μέχρι να συμπληρωθούν όλα τα υποχρεωτικά πεδία της φόρμας παραμένει απενεργοποιημένο. Τα πεδία στο “QoS Constraints” δεν είναι υποχρεωτικά για συμπλήρωση. Σημειώνεται ότι κάποια πεδία της φόρμας μπορεί να τροποποιηθούν και εμπλουτιστούν με βάση τους μηχανισμούς που θα αναπτυχθούν στις επόμενες δράσεις. Η εγγραφή της ομοσπονδίας επιτυγχάνεται όταν οι τιμές που έχουν εισαχθεί στη φόρμα είναι σωστές.

Federation Registration

Federation Id
fed20 ✓
From 4 to 30 characters. Include only letters, digits, '-' and '_'. You can leave it empty for autogeneration

Federation Name
fed20 ✓
From 3 to 30 characters

Information Model
smart city x
Select the federation information model

Public
Yes x
Has the federation public visibility?

Federation Members
test-platform ✓ + -
From 4 to 30 characters. Include only letters, digits, '-' and '_'

QoS Constraints +

Availability x >= x 50 ✓ -
Mandatory Mandatory Mandatory
50 ✓ type 1 ✓
Optional Optional

Federation rule types

Rule Type
Providers x

Data Availability
Closed x

Service Type
smart city ✓

Supported Ontologies
smart city ontology ✓

Federation government rules

Board government
user1 ✓ + -
From 4 to 30 characters. Include only letters, digits, '-' and '_'

Insert the supported proposals
remove member ✓ + -
From 4 to 30 characters. Include only letters, digits, '-' and '_'

Vote Rules Base
Board x

Tokens
50 ✓

Vote Rules Approval %
100 x

Quality assurance

Qos %
50 ✓

Reputation %
50 ✓

Min Number of Federations
1 ✓

Under Performance
RequestRemove x

Marketplace policy

Charge Policy
PerProduct x

Federation Product
Packaging x

Profit Policy
PerSource x

Coin
IoTFeds x

Submit **Close**

Το αποτέλεσμα της επιτυχούς εγγραφής της ομοσπονδίας στο IoTFeds σύστημα συνοδεύεται με την εμφάνιση του μηνύματος: “Registration of federation fed20 was successful”.

IoTFEDS User Dashboard **iotFedsUser**

Register New Federation

Registration of federation "fed20" was successful! x

fedq1 +

Invite

Change Federation Rules

Delete

fed20 +

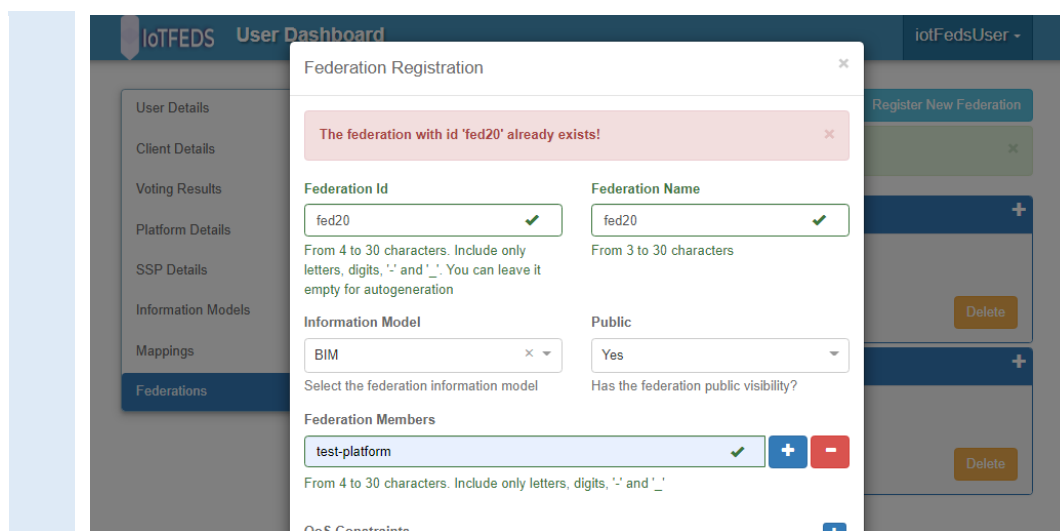
Invite

Change Federation Rules

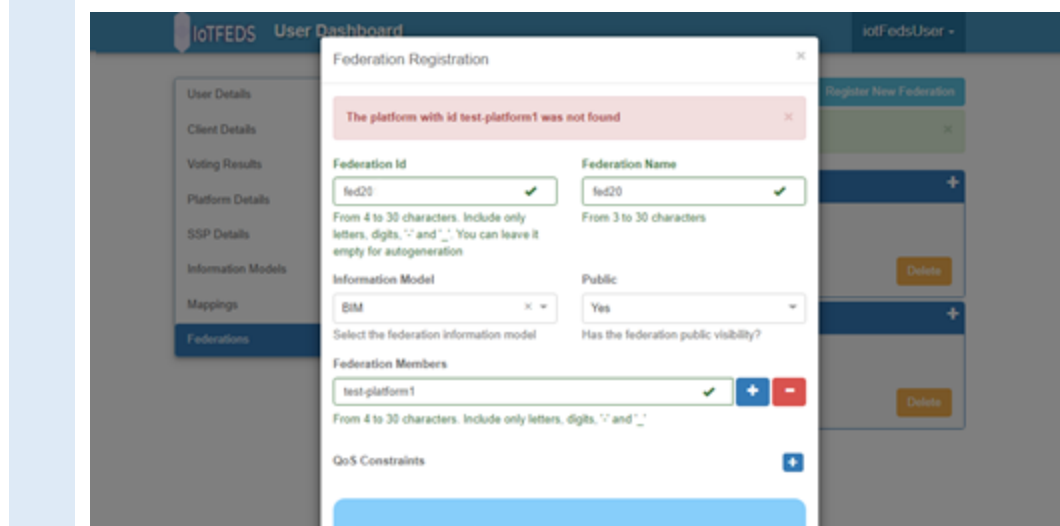
Delete

Σε περίπτωση σφάλματος εμφανίζεται αντίστοιχο μήνυμα.

Στην περίπτωση όπου ο χρήστης επιλέξει ταυτότητα ομοσπονδίας (federation id) που έχει καταχωρηθεί στο σύστημα εμφανίζεται μήνυμα λάθους: The federation with id “fed20” already exists.



Στην περίπτωση όπου ο χρήστης επιλέξει μέλος ομοσπονδίας (federation member) που δεν υπάρχει (πχ test-platform1) τότε το σύστημα εμφανίζει μήνυμα λάθους: The platform with id “test-platform1” was not found.



AdminGUI-6:

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ένα ομόσπονδο μέλος μπορεί να ζητήσει την συμμετοχή ενός νέου μέλους στην ομοσπονδία.

Ο χρήστης προσκαλεί ένα νέο μέλος στην επιθυμητή ομοσπονδία από τη λίστα “Federations List” του μενού “Federations”.

Προαπαιτούμενη περίπτωση δοκιμής:

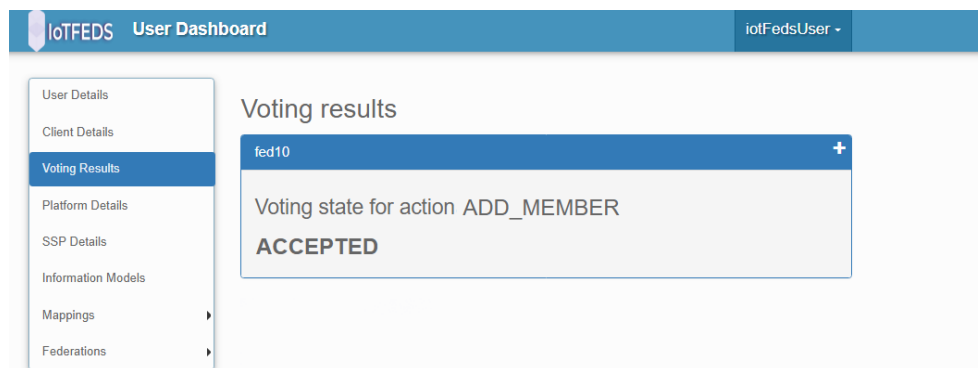
AdminGUI-5 (Εγγραφή ομοσπονδίας στο IotFeds σύστημα).

Προαπαιτούμενο: Ο χρήστης γνωρίζει το όνομα του μέλους που πρόκειται να προσκληθεί ως νέο μέλος στην ομοσπονδία. Το μέλος έχει εγγραφεί στο IotFeds σύστημα με τη διαδικασία που περιγράφεται στην περίπτωση δοκιμής AdminGUI-4: https://iotfeds-symbiotecore.intracom-telecom.com/coreInterface/get_available_aams.

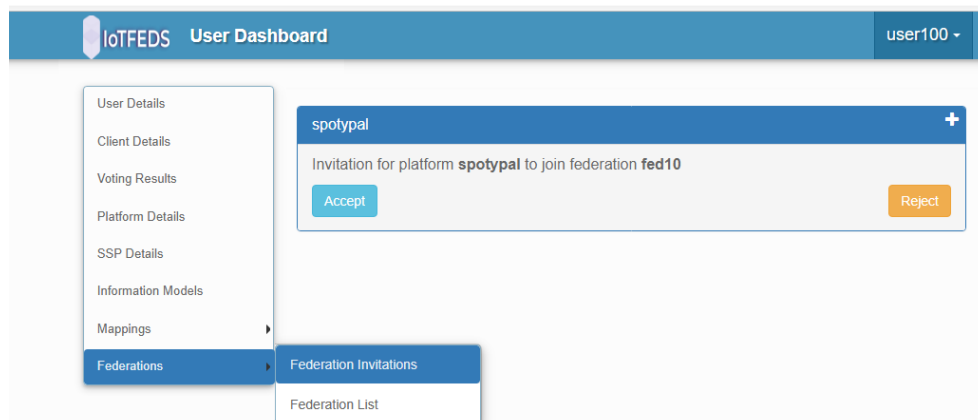
Αναμενόμενα αποτελέσματα:

Η φόρμα για τη συμπλήρωση των στοιχείων του οργανισμού που πρόκειται να προστεθεί στην ομοσπονδία εμφανίζεται στην οθόνη όταν ο χρήστης πατάει το κουμπί “Invite”.

Στην περίπτωση που το αποτέλεσμα της ψηφοφορίας για ένταξη στην ομοσπονδία είναι θετικό τότε εμφανίζεται στην οθόνη του χρήστη η παρακάτω ενημέρωση:



Στη συνέχεια το νέο μέλος της ομοσπονδίας δέχεται την πρόσκληση στο λογαριασμό του στο IoTFeds μέσα από το μενού Federations->Federation Invitations.



Ο χρήστης αποδεχόμενος την πρόσκληση πατάει το κουμπί “Accept” και εμφανίζεται τότε στην οθόνη του το παρακάτω μήνυμα:

The invitation to platform with id "spotypal" to join the federation with id "fed10" was accepted successfully

Παράλληλα ο χρήστης που έστειλε την πρόσκληση ενημερώνεται για την αποδοχή της πρόσκλησης με το ίδιο μήνυμα στο μενού Federations->Federation Invitations.

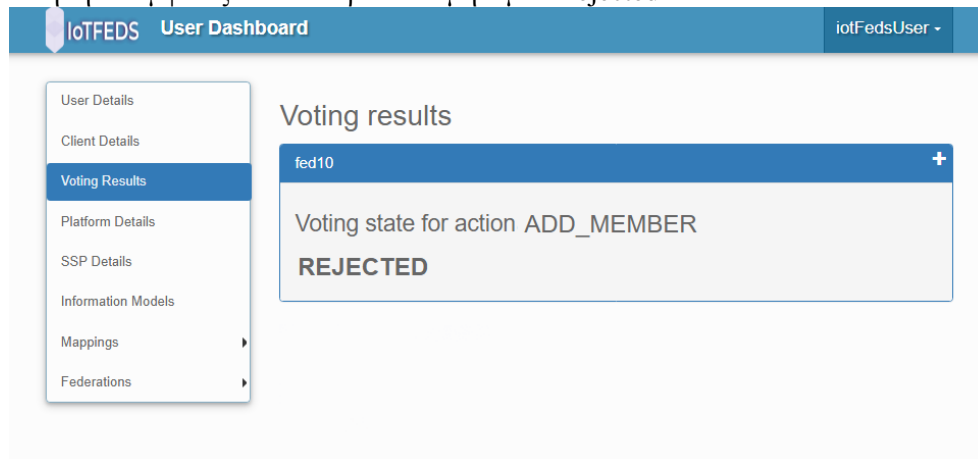
The invitation to platform with id "spotypal" to join the federation with id "fed10" was accepted successfully

Στην περίπτωση που ο χρήστης που δέχεται την πρόσκληση δεν επιθυμεί την ένταξη του οργανισμού στην ομοσπονδία, τότε πατάει το κουμπί “Reject” και τότε εμφανίζεται το παρακάτω μήνυμα στην οθόνη του:

The invitation to platform with id "spotypal" to join the federation with id "fed10" was rejected successfully

Ταυτόχρονα το ίδιο μήνυμα εμφανίζεται στην οθόνη του χρήστη που έκανε την πρόσκληση.

Στην περίπτωση όπου η αίτηση για ένταξη στην ομοσπονδία απορρίφθηκε στην ψηφοφορία τότε στη σελίδα Voting Results στο λογαριασμό του χρήστη που έκανε την αίτηση θα εμφανίζεται το παρακάτω μήνυμα “Rejected”.



AdminGUI-7:

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ένα μέλος της IoTFeds πλατφόρμας μπορεί να ζητήσει την συμμετοχή του σε μία υπάρχουσα ομοσπονδία.

Ο χρήστης έχει πρόσβαση στις διαθέσιμες ομοσπονδίες.

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-4 (Ο χρήστης έχει κάνει εγγραφή οργανισμού στο IoTFeds σύστημα).

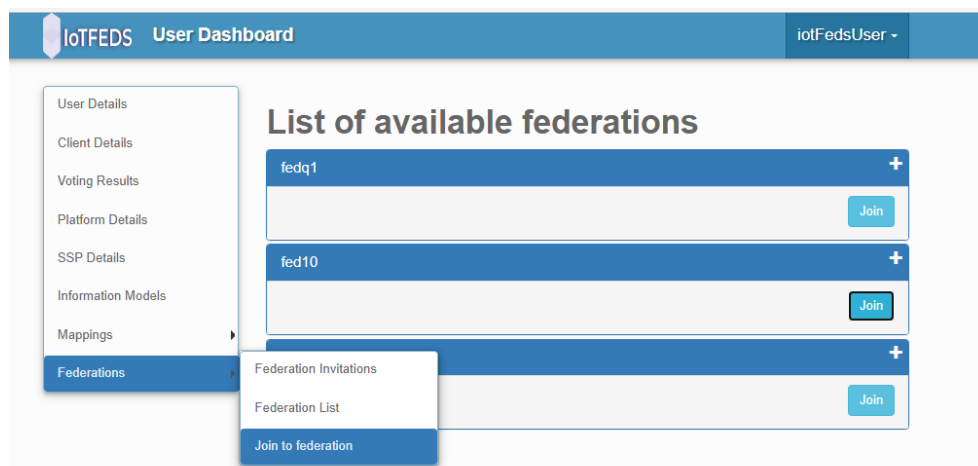
AdminGUI-5 (Άλλοι χρήστες της IoTFeds πλατφόρμας έχουν εγγράψει ομοσπονδίες και είναι έτοιμες να υποδεχτούν νέα μέλη).

Αναμενόμενα αποτελέσματα:

Η λίστα με τις ομοσπονδίες εμφανίζεται στην οθόνη του χρήστη με την επιλογή “Join to federation” στο μενού “Federations”.

Αποτελέσματα που παρατηρήθηκαν:

Στην οθόνη του χρήστη εμφανίζεται μια λίστα από ομοσπονδίες στις οποίες μπορεί να ενταχθεί.



Ο χρήστης έχει πρόσβαση στις πληροφορίες της ομοσπονδίας.

Ο χρήστης πατώντας το “+” κουμπί βλέπει στην οθόνη του τους κανόνες του Smart Contract που ισχύει για κάθε ομοσπονδία καθώς και ποια μέλη (οργανισμοί) την απαρτίζουν.

IoTFEDS

User Dashboard

iotFedsUser

User Details

Client Details

Voting Results

Platform Details

SSP Details

Information Models

Mappings

Federations

List of available federations

fedq1

Federation Id

fedq1

Federation Name

fedq1

Information Model

BIM

Public

Yes

Federation Members

icom-platform

QoS Constraints

Metric

Load

Comparator

>=

Threshold

50

Duration

1000

Resource Type

public

Federation Rules

Federation rule types

Rule Type

Providers

Data Availability

Closed

Service Type

smart environment

Supported Ontologies

smart city ontology

Federation government rules

Board Gov

user1

Proposals

remove member

Tokens

50

Vote Rules Approval %

100

Vote Rules Base

Voters

Quality assurance

Qos %

50

Reputation %

50

Min Number of Federations

1

Under Performance

None

Marketplace policy

Charge Policy

Free

Federation Product

Matching

Profit Policy

PerSource

Coin

IoTFeds

Join

fed10

+

Join

Ο χρήστης στέλνει το αίτημα συμμετοχής στην επιθυμητή ομοσπονδία (fedq1 στη δοκιμή) που εκκινεί διαδικασία ψηφοφορίας.

Ο χρήστης πατώντας το “Join” κουμπί στέλνει το αίτημα συμμετοχής στην επιθυμητή ομοσπονδία (fedq1 στη δοκιμή) και ξεκινά η διαδικασία ψηφοφορίας από μέλη που συμμετέχουν στην ομοσπονδία fedq1.

Αναμενόμενα αποτελέσματα:

Ο χρήστης μπορεί να παρακολουθήσει το αίτημα του στη λίστα των “Pending requests”.

Αποτελέσματα που παρατηρήθηκαν:

Στην περίπτωση που η ψηφοφορία είναι σε εξέλιξη ο χρήστης στην οθόνη του έχει την παρακάτω ενημέρωση:

The screenshot shows the 'IoTFEDS User Dashboard'. On the left is a sidebar menu with options: User Details, Client Details, Voting Results (highlighted), Platform Details, SSP Details, Information Models, Mappings, and Federations. The main content area is titled 'Voting results' and contains a table with one row for 'fedq1'. The table shows the 'Voting state for action ADD_MEMBER' as 'PENDING'.

fedq1
Voting state for action ADD_MEMBER PENDING

Κάθε μέλος της ομοσπονδίας ενημερώνεται για το αίτημα και την ψηφοφορία.

Αναμενόμενα αποτελέσματα:

Κάθε ψηφοφόρος αποκτάει πρόσβαση στην ψηφοφορία για να υποβάλει την ψήφο του.

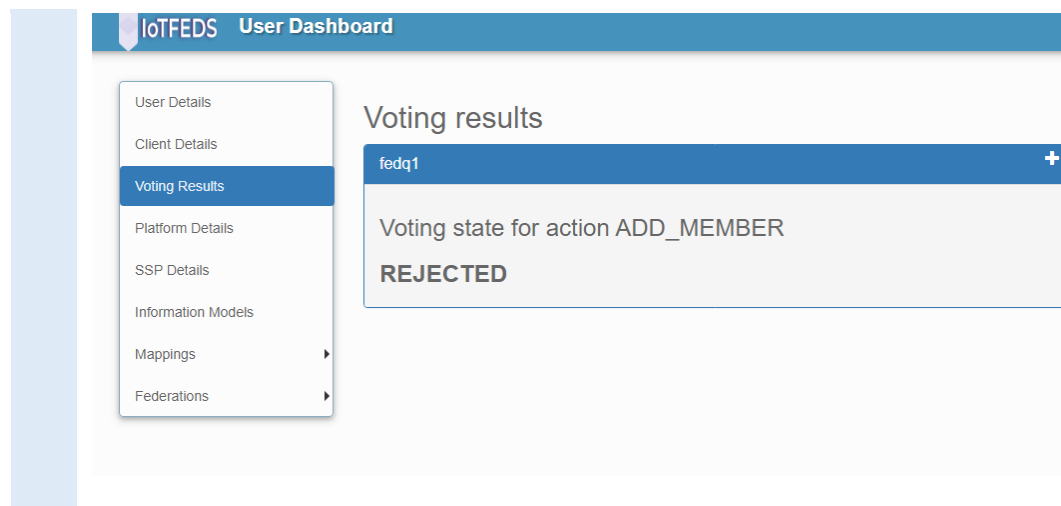
Αποτελέσματα που παρατηρήθηκαν:

Στην περίπτωση που το αποτέλεσμα της ψηφοφορίας είναι θετικό ο χρήστης στην οθόνη του έχει την παρακάτω ενημέρωση:

The screenshot shows the 'IoTFEDS User Dashboard' with the same sidebar menu. The 'Voting results' section now shows the 'Voting state for action ADD_MEMBER' as 'ACCEPTED'.

fedq1
Voting state for action ADD_MEMBER ACCEPTED

Στην περίπτωση που το αποτέλεσμα της ψηφοφορίας είναι αρνητικό ο χρήστης στην οθόνη του έχει την παρακάτω ενημέρωση:

**AdminGUI-8:****PASS**

Η δοκιμή αυτή εξασφαλίζει ότι ένα ομόσπονδο μέλος (οργανισμός) μπορεί να αποχωρήσει από την ομοσπονδία στην οποία ανήκει.

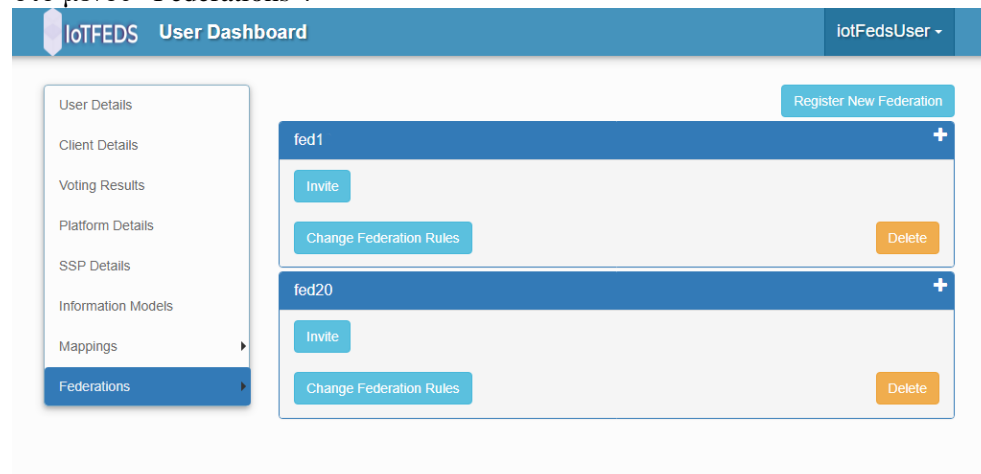
Ο χρήστης έχει πρόσβαση στις διαθέσιμες ομοσπονδίες του.

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-6,
AdminGUI-7

Αναμενόμενα αποτελέσματα:

Οι ομοσπονδίες στις οποίες ανήκει ο χρήστης εμφανίζονται στη λίστα “Federations List” στο μενού “Federations”.



Ο χρήστης επιλέγει το μέλος προς αποχώρηση από την ομοσπονδία.

Ο χρήστης πατάει το “+” κουμπί για να δει τη λίστα των μελών μιας ομοσπονδίας (fed1 στη δοκιμή). Από τη λίστα επιλέγει το όνομα του οργανισμού του (icom-platform στη δοκιμή) που επιθυμεί να απομακρύνει από την ομοσπονδία fed1 .

Στη συνέχεια επιλέγει το “Remove member from federation” κουμπί για να ενεργοποιήσει τη διαδικασία αποχώρησης του μέλους icom-member από την ομοσπονδία fed1.

Αποτελέσματα που παρατηρήθηκαν:

Με το πάτημα του κουμπιού “Remove member from federation” εμφανίζεται στην οθόνη του χρήστη μήνυμα επιβεβαίωσης:

Ο χρήστης στέλνει το αίτημα για αποχώρηση από την ομοσπονδία.

Αναμενόμενα αποτελέσματα:

Ο χρήστης ενημερώνεται για το αποτέλεσμα του αιτήματος και ανανεώνεται η λίστα ομοσπονδιών που του εμφανίζεται σε περίπτωση επιτυχίας.

Αποτελέσματα που παρατηρήθηκαν:

Στην οθόνη του χρήστη εμφανίζεται το παρακάτω μήνυμα:

The platform with id "icom-platform" left the federation "fed1" successfully!

Με το ίδιο μήνυμα ενημερώνονται και τα υπόλοιπα μέλη της ομοσπονδίας για την απομάκρυνση του οργανισμού icom-platform.

Η λίστα των ομοσπονδιών ανανεώνεται και εμφανίζεται χωρίς πλέον το μέλος icom-platform:

The screenshot shows the 'IoTFEDS User Dashboard' with a sidebar menu on the left containing: User Details, Client Details, Voting Results, Platform Details, SSP Details, Information Models, Mappings, and Federations (highlighted). The main content area is titled 'fed1' and includes a 'Register New Federation' button. The configuration fields are as follows:

- Federation Id:** fed1
- Federation Name:** fed1
- Information Model:** BIM
- Public:** Yes
- Federation Members:** test-platform (with an 'Activate Vote to remove member' button)
- QoS Constraints:**
 - Metric:** Load
 - Comparator:** >=
 - Threshold:** 50
 - Duration:** 1000
 - Resource Type:** public
- Federation Rules:**
 - Federation rule types:**

Rule Type	Data Availability

AdminGUI-10:**PASS**

Η δοκιμή αυτή εξασφαλίζει ότι ένα ομόσπονδο μέλος της IoTFeds πλατφόρμας ζητάει την απομάκρυνση ενός άλλου ομόσπονδου μέλους της ίδιας ομοσπονδίας.

Ο χρήστης έχει πρόσβαση στις ομοσπονδίες που συμμετέχει και στις πληροφορίες τους.

Προαπαιτούμενη περίπτωση δοκιμής:

AdminGUI-6

AdminGUI-7

Αναμενόμενα αποτελέσματα:

Τα μέλη της ομοσπονδίας εμφανίζονται στη φόρμα στοιχείων της ομοσπονδίας και υπάρχει επιλογή για την απομάκρυνση ενός άλλου μέλους με τη διαδικασία της ψηφοφορίας.

Αποτελέσματα που παρατηρήθηκαν:

Στην οθόνη του χρήστη εμφανίζεται η σελίδα που περιέχει τη λίστα των μελών της ομοσπονδίας (Federation Members). Από τη λίστα, τα μέλη (οργανισμοί) που η απομάκρυνση τους απαιτεί την ενεργοποίηση ψηφοφορίας συνοδεύονται με το κουμπί "Activate Vote to remove member".

The screenshot shows the 'IoT FEDS User Dashboard' with a sidebar menu on the left containing: User Details, Client Details, Voting Results, Platform Details, SSP Details, Information Models, Mappings, and Federations (highlighted). The main content area is titled 'fed1' and includes a 'Register New Federation' button. It contains several sections: 'Federation Id' and 'Federation Name' (both 'fed1'), 'Information Model' (BIM) and 'Public' (Yes), 'Federation Members' (icom-platform and test-platform), and 'QoS Constraints' (Metric: Load, Comparator: >=, Threshold: 50). For the 'test-platform' member, there are two buttons: 'Remove member from federation' and 'Activate Vote to remove member'.

Ο χρήστης επιλέγει το μέλος για αποχώρηση από την ομοσπονδία και εκκινεί τη διαδικασία ψηφοφορίας.

Ο χρήστης επιλέγει το κουμπί “Activate Vote to remove member” στο μέλος που επιθυμεί να αποχωρήσει.

Αναμενόμενα αποτελέσματα:

Το μήνυμα επιβεβαίωσης για απομάκρυνση του μέλους από την ομοσπονδία εμφανίζεται στην οθόνη του χρήστη.

This screenshot shows the same dashboard as before, but with a confirmation dialog box overlaid. The dialog box contains the text: 'Are you sure you want to activate a voting procedure to remove the member with id test-platform from the federation fed1?'. It has two buttons: 'Activate voting to remove the member' (in red) and 'Close'.

Αποτελέσματα που παρατηρήθηκαν:

Πατώντας το κουμπί “Activate voting to remove the member” το αίτημα εμφανίζεται στη λίστα με τα αιτήματα ψηφοφορίας του χρήστη με την ένδειξη ότι η ψηφοφορία είναι σε εξέλιξη.

The screenshot shows the 'IoT FEDS User Dashboard' for a user named 'iotFedsUser'. On the left is a sidebar menu with options: User Details, Client Details, Voting Results (highlighted), Platform Details, SSP Details, Information Models, Mappings, and Federations. The main content area is titled 'Voting results' and contains a table with one entry for 'fed1'. The entry shows the voting state for the action 'REMOVE_MEMBER' as 'PENDING'.

Κάθε μέλος της ομοσπονδίας ενημερώνεται για το αίτημα και αποκτάει πρόσβαση στην ψηφοφορία.

Αναμενόμενα αποτελέσματα:

Ο χρήστης ενημερώνεται για την ψηφοφορία και την υποβολή της ψήφου του στο σύστημα

Αποτελέσματα που παρατηρήθηκαν:

Dear IoT FEDS User,

Please take a moment to cast your vote through this link: <http://160.40.50.169:3000/?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRHZvdGluZyI6InIaG83ZjR6dXB0c2VvYWZyYWJofiwiaUR2b3Rlcil6ImJlbjxwvJWfWBAiqh9glYe2ZvTC6morMePho62Q4>

Μετά την ολοκλήρωση της ψηφοφορίας, κάθε μέλος της ομοσπονδίας ενημερώνεται για το αίτημα.

Αναμενόμενα αποτελέσματα:

Στην περίπτωση θετικής ψήφου ο χρήστης ενημερώνεται για την απομάκρυνσή του και η λίστα μελών της ομοσπονδίας ενημερώνεται.

Αποτελέσματα που παρατηρήθηκαν:

Στην περίπτωση που το αποτέλεσμα της ψηφοφορίας είναι θετικό για την απομάκρυνση του μέλους τότε εμφανίζεται στην οθόνη της σελίδας “Voting Results” το παρακάτω μήνυμα αποτελέσματος της ψηφοφορίας:

This screenshot is similar to the first one, showing the 'IoT FEDS User Dashboard' for 'iotFedsUser'. The 'Voting Results' section now shows the status for the 'REMOVE_MEMBER' action as 'ACCEPTED'.

Οι πληροφορίες της ομοσπονδίας εμφανίζονται χωρίς το μέλος που απομακρύνθηκε:

The screenshot displays the 'IoT FEDS User Dashboard' for a user named 'iotFedsUser'. The left sidebar contains a menu with the following items: User Details, Client Details, Voting Results, Platform Details, SSP Details, Information Models, Mappings, and Federations (highlighted). The main content area is titled 'fed1' and contains several sections:

- Federation Id:** fed1
- Federation Name:** fed1
- Information Model:** BIM
- Public:** Yes
- Federation Members:** icom-platform (with a 'Remove member from federation' button)
- QoS Constraints:**
 - Metric:** Load
 - Comparator:** >=
 - Threshold:** 50
 - Duration:** 1000
 - Resource Type:** public
- Federation Rules:**
 - Federation rule types:**
 - Rule Type:** Providers
 - Data Availability:** Closed
 - Service Type:** smart environment
 - Supported Ontologies:** smart city

 A 'Register New Federation' button is located at the top right of the dashboard.

Στην περίπτωση που το αποτέλεσμα της ψηφοφορίας είναι αρνητικό για την απομάκρυνση του μέλους τότε εμφανίζεται στην οθόνη της σελίδας “Voting Results” το παρακάτω μήνυμα αποτελέσματος της ψηφοφορίας:

The screenshot displays the 'IoT FEDS User Dashboard' for a user named 'iotFedsUser'. The left sidebar contains a menu with the following items: User Details, Client Details, Voting Results (highlighted), Platform Details, SSP Details, Information Models, Mappings, and Federations. The main content area is titled 'Voting results' and shows a message for 'fed1':

- Voting state for action REMOVE_MEMBER**
- REJECTED**

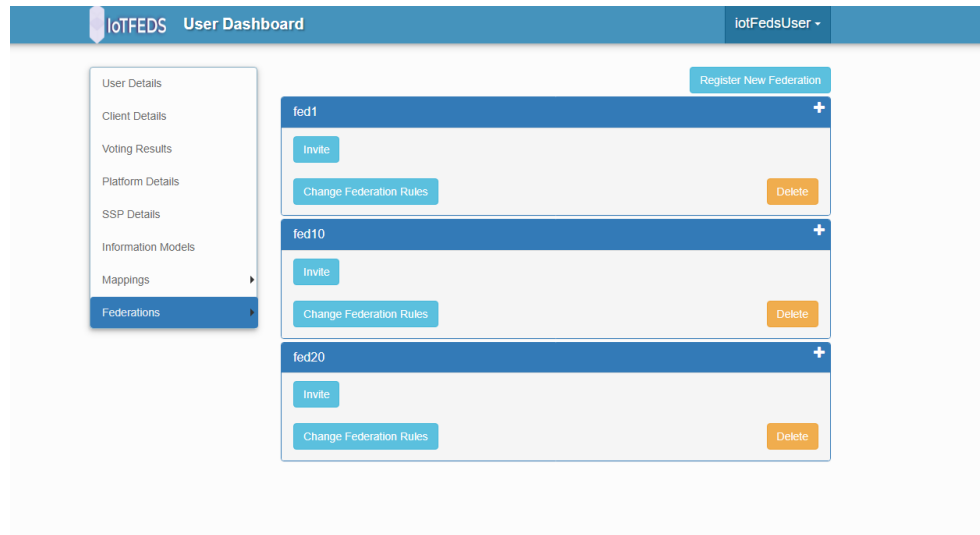
AdminGUI-11:**PASS**

Η δοκιμή αυτή εξασφαλίζει ότι ένα ομόσπονδο μέλος της IoTFeds πλατφόρμας μπορεί να ζητήσει την τροποποίηση κανόνων σε ομοσπονδίες που συμμετέχει.

Ο χρήστης έχει πρόσβαση στις ομοσπονδίες που συμμετέχει και στις πληροφορίες τους.

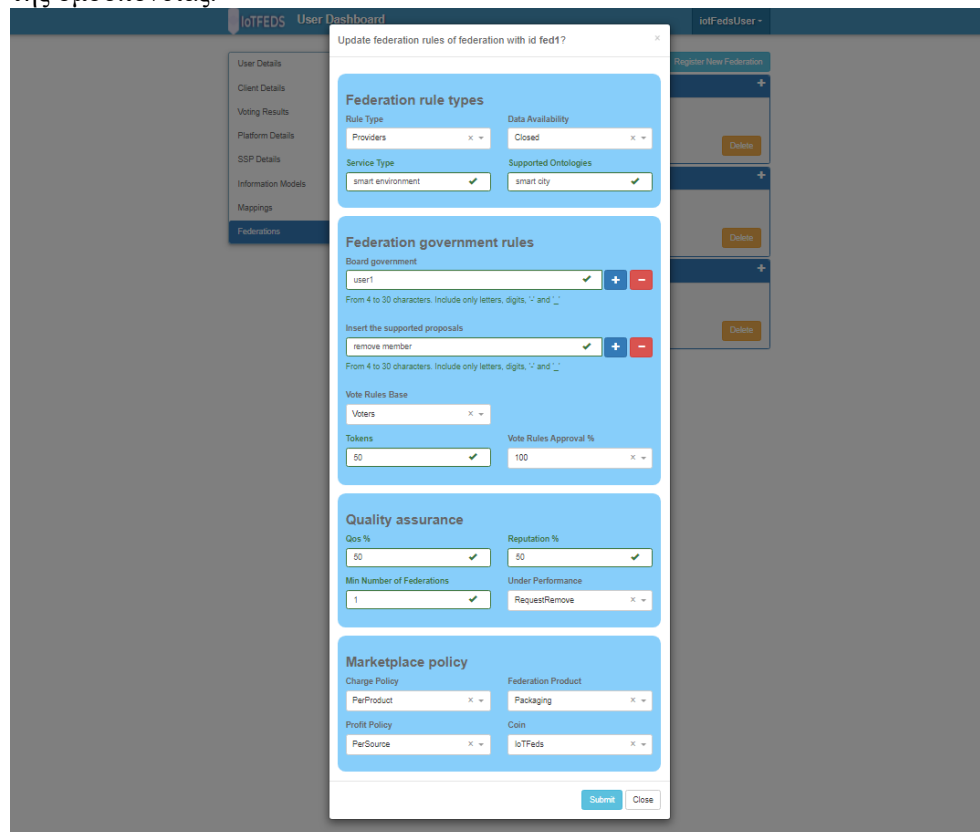
Προαπαιτούμενη περίπτωση δοκιμής:
AdminGUI-5 (Εγγραφή ομοσπονδίας στο IotFeds σύστημα).

Αναμενόμενα αποτελέσματα:
Ο χρήστης εισέρχεται στη σελίδα με τη λίστα των ομοσπονδιών:



Στη συνέχεια επιλέγει την ομοσπονδία στην οποία επιθυμεί να γίνει τροποποίηση κανόνων.

Αποτελέσματα που παρατηρήθηκαν:
Πατώντας το κουμπί “Change Federation Rules” εμφανίζεται η φόρμα με τους κανόνες της ομοσπονδίας.



Ο χρήστης επιλέγει την τροποποίηση των κανόνων της ομοσπονδίας.

Αναμενόμενα αποτελέσματα:

Ο χρήστης αφού ολοκληρώσει την τροποποίηση των κανόνων πατάει το κουμπί “Submit” για να προωθηθεί η τροποποίηση των κανόνων σε ψηφοφορία από τα μέλη της ομοσπονδίας.

Αποτελέσματα που παρατηρήθηκαν:

Το παρακάτω μήνυμα ενημερώνει το χρήστη για την επιτυχημένη προώθηση της αίτησης για τροποποίηση κανόνων της ομοσπονδίας.

Your change federation rules request is forwarded to voting procedure

Στη σελίδα Voting Results αρχικά εμφανίζεται το μήνυμα ότι η ψηφοφορία για τροποποίηση κανόνων είναι σε εξέλιξη.

IoT FEDS

User Dashboard

iotFedsUser +

User Details

Client Details

Voting Results

Platform Details

SSP Details

Information Models

Mappings

Federations

Voting results

fedq1

Voting state for action UPDATE_FEDERATION_RULES

PENDING

Κάθε μέλος της ομοσπονδίας ενημερώνεται για το αίτημα και αποκτάει πρόσβαση στην ψηφοφορία.

Αναμενόμενα αποτελέσματα:

Ο χρήστης υποβάλλει τη ψήφο του στο σύστημα.

Αποτελέσματα που παρατηρήθηκαν:

Dear IoTFeds User,

Please take a moment to cast your vote through this link: <http://160.40.50.169:3000/>

[token=eyJhbGciOiJIUzI1NiIsInR5cE6kcpXVCJ9.eyJRHRzdGluzYl6lnIa8ZjR6dXB0c2VvYWZYZWJoIiwiaWF0eQ6mcmMePho62O4](#)

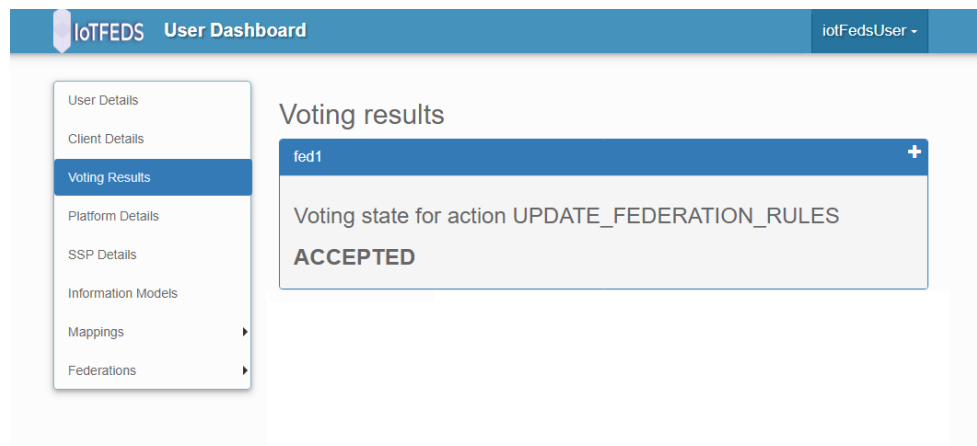
Μετά την ολοκλήρωση της ψηφοφορίας σε περίπτωση θετικής ψήφου, οι κανόνες της ομοσπονδίας τροποποιούνται και ο χρήστης ενημερώνεται με τα αποτελέσματα της ψηφοφορίας.

Αναμενόμενα αποτελέσματα:

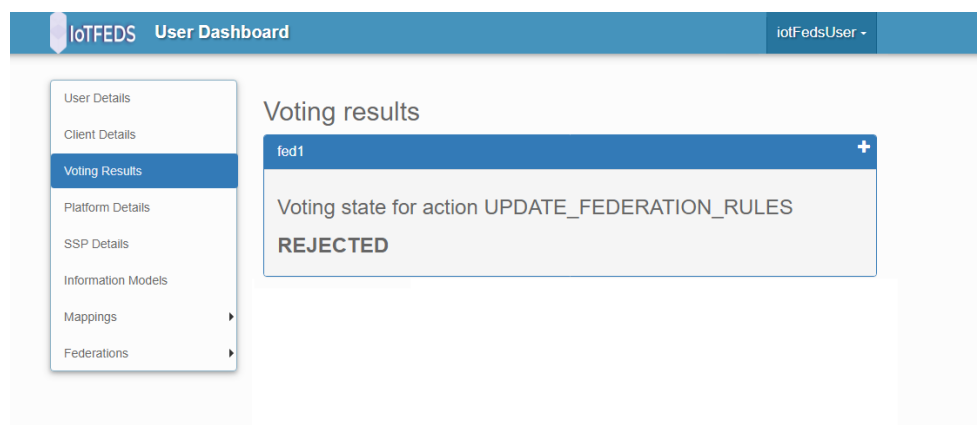
Ο χρήστης ενημερώνεται με το αποτέλεσμα της ψηφοφορίας και οι κανόνες της ομοσπονδίας στη φόρμα στοιχείων της εμφανίζονται τροποποιημένοι μόνο μετά από θετική ψήφο.

Αποτελέσματα που παρατηρήθηκαν:

Στην περίπτωση που η τροποποίηση των κανόνων έχει εγκριθεί από την ψηφοφορία, τότε στη σελίδα Voting Results εμφανίζεται το παρακάτω ενημερωτικό μήνυμα:



Στην περίπτωση που η τροποποίηση των κανόνων δεν έχει εγκριθεί από την ψηφοφορία, τότε στη σελίδα Voting Results εμφανίζεται το παρακάτω ενημερωτικό μήνυμα:



AdminGUI-12

PASS

Η δοκιμή αυτή εξασφαλίζει ότι ένας χρήστης μπορεί να διαγράψει μία πλατφόρμα του από το IoTFeds σύστημα.

Ο χρήστης έχει πρόσβαση στις πλατφόρμες του και τις πληροφορίες τους.

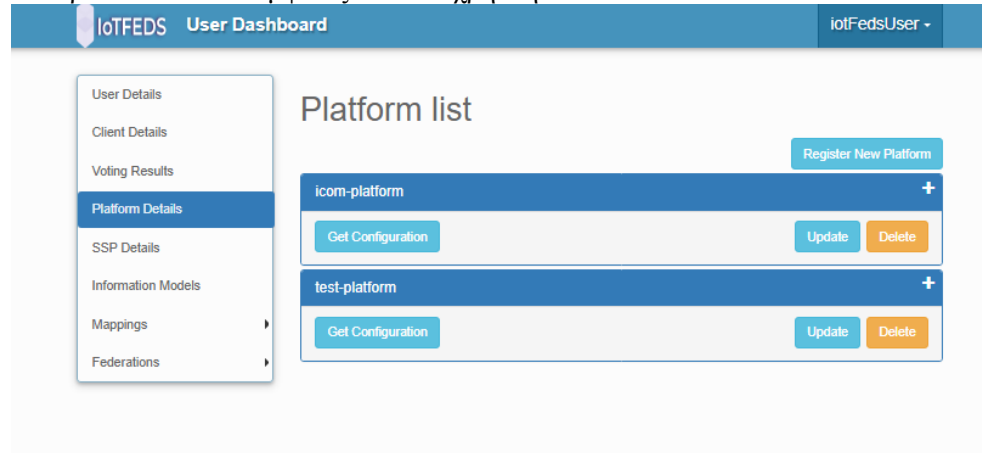
Προαπαιτούμενη περίπτωση δοκιμής:
AdminGUI-4

Αναμενόμενα αποτελέσματα:

Οι πλατφόρμες του χρήστη εμφανίζονται σε λίστα στο μενού “Platform Details”.

Αποτελέσματα που παρατηρήθηκαν:

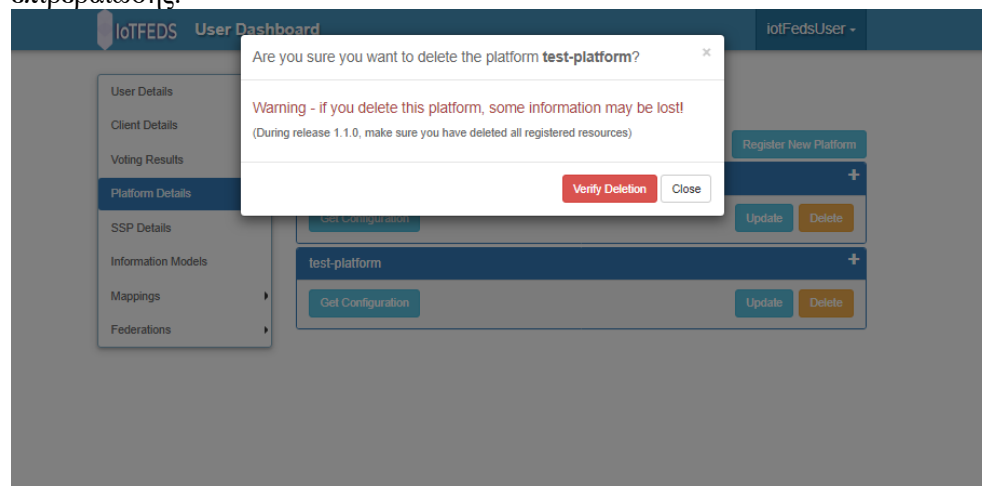
Η παρακάτω σελίδα εμφανίζεται στο χρήστη:



Ο χρήστης επιλέγει τη διαγραφή της πλατφόρμας του.

Αναμενόμενα αποτελέσματα:

Πατώντας το “Delete” κουμπί εμφανίζεται στην οθόνη του χρήστη μήνυμα επιβεβαίωσης.



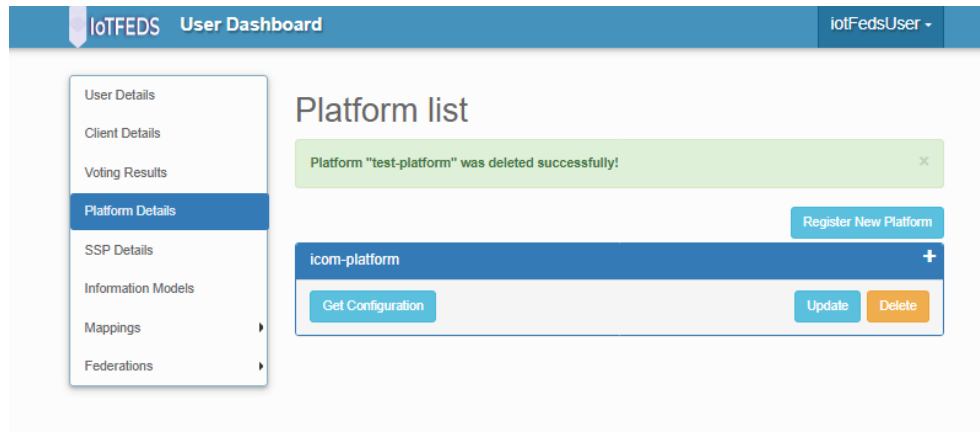
Στη συνέχεια ο χρήστης επιβεβαιώνει την πρόθεσή του για διαγραφή της πλατφόρμας. Ο χρήστης πατάει το “Verify Deletion” κουμπί.

Αναμενόμενα αποτελέσματα:

Το IoTFeds σύστημα επιβεβαιώνει την επιτυχή διαγραφή της πλατφόρμας εμφανίζοντας μήνυμα στην οθόνη του χρήστη. Το μενού “Platform Details” ενημερώνεται εμφανίζοντας μια ανανεωμένη λίστα από πλατφόρμες χωρίς την πλατφόρμα που διαγράφηκε.

Αποτελέσματα που παρατηρήθηκαν:

Η παρακάτω σελίδα εμφανίζεται στο χρήστη.



5 Συμπεράσματα

Στο παραδοτέο αυτό περιγράφεται η ενδιάμεση έκδοση του ολοκληρωμένου συστήματος της IoTFeds πλατφόρμας. Επιπλέον, τεκμηριώνεται η ενσωμάτωση των υπηρεσιών των στοιχείων λογισμικού που αναπτύχθηκαν ή και επεκτάθηκαν στα πλαίσια του έργου για τη διαχείριση των IoT ομοσπονδιών και την επίτευξη της σημασιολογικής διαλειτουργικότητας στο ολοκληρωμένο σύστημα (Δ2.1/Δ2.2).

Πιο συγκεκριμένα, αρχικά περιγράφεται η μεθοδολογία που ακολουθείται για την ενσωμάτωση και τη δοκιμή του ολοκληρωμένου συστήματος. Στη συνέχεια παρουσιάζονται τα στοιχεία λογισμικού των υποσυστημάτων που συμμετέχουν στην υλοποίηση των μηχανισμών αυτών και οι αντίστοιχες διεπαφές που συμπεριλήφθηκαν στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος.

Τέλος, ακολουθεί η λειτουργική δοκιμή της IoTFeds πλατφόρμας και η επίδειξη της χρήσης της όπως υλοποιήθηκε στην ενδιάμεση έκδοση του ολοκληρωμένου συστήματος. Η τελική έκδοση της IoTFeds πλατφόρμας θα παρουσιαστεί στο παραδοτέο που ακολουθεί - Π1.5 «Τελική Έκδοση Ολοκληρωμένου Συστήματος».

6 Αναφορές

There are no sources in the current document.