IoTFeds

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

Π4.1 – Δοκιμαστική Εφαρμογή IoT και Προετοιμασία Πιλοτικής Εφαρμογής

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

Έργο	IoTFeds
Κωδικός Έργου	Τ2ΕΔΚ-02178
Ενότητα Εργασίας	ΕΕ4 (Δ4.1, Δ4.2)
Υπεύθυνος Ενότητας	ICOM
Υπεύθυνος Παραδοτέου	ICOM
Έκδοση Παραδοτέου	1.0
Ημερομηνία	10/10/2023

Συντονιστές Παραδοτέου
Νάσιου Απόστολος, ICOM
Λασκαράτος Δημήτριος, ICOM
Πιέτρη Ήλια, ICOM
Τσάκαλος Κωνσταντίνος, ICOM
Τσαρουχά Μαρία, ICOM
Χαραλαμπίδης Μάριος, ICOM

Συν-συγγραφείς
Αθανασάκης Ευάγγελος, ΕΚΕΤΑ
Γεωργιάδης Στέφανος, ΕΚΕΤΑ
Γκούσκος Στέλιος, TERRACOM
Γόγολος Γεώργιος, TERRACOM
Γκορίτσας Κώστας, TERRACOM
Ευθυμιάδης Παύλος, TERRACOM
Καλαμπόκης Κωνσταντίνος, TERRACOM
Καραδήμου Δήμητρα, AUEB
Νάκος Μιχαήλ, TERRACOM
Νταρζάνος Γεώργιος, AUEB
Ουμνόβα Ναταλία, TERRACOM
Παπαϊωάννου Αθανάσιος, AUEB
Πεχλιβανίδης Αναστάσιος, ΕΚΕΤΑ
Πετρίδης Κωνσταντίνος, TERRACOM
Πολυμένη Σοφία, ΕΚΕΤΑ
Σταμούλης Γεώργιος, AUEB
Τάγκας Αλέξανδρος, TERRACOM
Τοκμακίδου Παρασκευή, TERRACOM
Σακελλαρίου Ζήσης, ΕΚΕΤΑ
Σπανός Γεώργιος, ΕΚΕΤΑ

Ακρωνύμια

Ακρωνύμιο	Ορισμός
BaaS	Blockchain as a Service
BIM	Best-Practice Information Model
CO	Carbon Monoxide
CO_2	Carbon Dioxide
$PM_{2.5}$	Particulate Matter
CIM	Common Information Model
LoRaWAN	Long Range Wide Area Network
GUI	Graphical User Interface
IoT	Internet of Things
EE	Ενότητα Εργασίας
RAP	Resource Access Proxy

Σύνοψη Παραδοτέου

Το παραδοτέο αυτό περιγράφει την προετοιμασία του περιβάλλοντος για την υποστήριξη του σεναρίου δοκιμής, που αποτελείται από δύο στάδια: την προετοιμασία περιβάλλοντος των IoT πλατφορμών και την υλοποίηση της δοκιμαστικής εφαρμογής. Το πρώτο στάδιο έχει στόχο τη συνέργεια των τριών πιλοτικών IoT πλατφορμών για τη δημιουργία ομοσπονδίας IoT και τη διάθεση προϊόντος δεδομένων από πολλαπλές ομόσπονδες πηγές δεδομένων. Απαιτεί την διαμόρφωση κάθε IoT πλατφόρμας με την εγκατάσταση του απαραίτητου ενδιάμεσου λογισμικού ώστε να είναι δυνατή η σύνδεση της στο IoTFeds σύστημα, η διάθεση των επιλεγμένων IoT πόρων της και η απόκτηση δεδομένων από αυτήν. Το δεύτερο στάδιο αφορά την υλοποίηση της δοκιμαστικής IoT εφαρμογής (ενός παρόχου εφαρμογής) για την παροχή μίας ολοκληρωμένης λύσης που βασίζεται στον συνδυασμό και την αξιοποίηση δεδομένων καιρού, κυκλοφορίας και στάθμευσης που συλλέγονται από πολλαπλές IoT πηγές δεδομένων. Η πρόσβαση του παρόχου εφαρμογής σε αυτά τα δεδομένα αποκτάται με την σύνθεση ενός ομοσπονδιακού προϊόντος δεδομένων και την αγορά του μέσω μιας ομοσπονδιακής αγοράςIoTFeds στην οποία συμμετέχουν οι πάροχοι των τριών πιλοτικών πλατφορμών εγγράφοντας τις πηγές δεδομένων τους. Η δοκιμαστική εφαρμογή θα χρησιμοποιηθεί για την επίδειξη χρήσης των λειτουργιών της IoTFeds πλατφόρμας.

Συνεπώς, το παραδοτέο αυτό συνοδεύει την δοκιμαστική IoT εφαρμογή και αποτυπώνει όλα τα στάδια που ακολουθήθηκαν για την προετοιμασία της πιλοτικής εφαρμογής ως αποτέλεσμα των δράσεων Δ4.1 «Υλοποίηση Δοκιμαστικής Εφαρμογής IoT» και Δ4.2 «Προετοιμασία Περιβάλλοντος Πιλοτικής Εφαρμογής». Η δοκιμαστική εφαρμογή που υλοποιήθηκε και το περιβάλλον της πιλοτικής εφαρμογής που στήθηκε θα αποτελέσουν τη βάση για την Δράση Δ4.3 «Πιλοτική Εφαρμογή Συστήματος», εξασφαλίζοντας την ετοιμότητα του περιβάλλοντος και την ομαλή λειτουργία του συστήματος για την εκτέλεση της πιλοτικής εφαρμογής.

Πίνακας Περιεχομένων

1 Αντικείμενο και Δομή του Παραδοτέου 9

2 Περιγραφή σεναρίου δοκιμής (πιλοτική και δοκιμαστική εφαρμογή) 10

2.1.1 Πιλοτική επίδειξη – Δημιουργία προϊόντος από πολλαπλές ομόσπονδες πηγές δεδομένων 11

2.1.2 Δοκιμαστική εφαρμογή – Χρήση δεδομένων προϊόντος από τρίτη εφαρμογή 12

3 Προετοιμασία περιβάλλοντος πιλοτικής εφαρμογής 14

3.1 Διαμόρφωση πλατφορμών IoT 14

3.1.1 Πλατφόρμα Spotypal-Zastel και διάθεση δεδομένων κίνησης 14

3.1.1.1 Περιγραφή πλατφόρμας Spotypal-Zastel 14

3.1.1.2 Διασύνδεση πλατφόρμας Spotypal-Zastel στο IoTFeds σύστημα 15

3.1.2 Πλατφόρμα Smart Environment και διάθεση περιβαλλοντικών δεδομένων 19

3.1.2.1 Περιγραφή πλατφόρμας Smart Environment 19

3.1.2.2 Διασύνδεση πλατφόρμας Smart Environment στο IoTFeds σύστημα 22

3.1.3 Πλατφόρμα UiTOP και διάθεση δεδομένων έξυπνης στάθμευσης 26

3.1.3.1 Περιγραφή πλατφόρμας UiTOP 26

3.1.3.2 Διασύνδεση πλατφόρμας UiTOP στο IoTFeds σύστημα 26

3.2 Δημιουργία ομοσπονδίας και εγγραφή IoT πόρων σε αυτήν 32

3.3 Δημιουργία προϊόντων και διάθεση τους στην IoTFeds αγορά 37

4 Υλοποίηση δοκιμαστικής εφαρμογής 40

4.1 Περιγραφή δοκιμαστικής εφαρμογής IoT 40

4.2 Επαλήθευση της δοκιμαστικής εφαρμογής IoT 41

5 Συμπεράσματα 46

Βιβλιογραφία 47

Ευρετήριο Εικόνων

Εικόνα 1: Αρχιτεκτονική για την υλοποίηση του σεναρίου δοκιμής. 10

Εικόνα 2: Προετοιμασία πιλοτικής εφαρμογής. 11

Εικόνα 3: Νέα εφαρμογή που βασίζεται στην χρήση δεδομένων από την IoTFeds πλατφόρμα. . 12

Εικόνα 4: Δημιουργία χρήστη terracom. 15

Εικόνα 5: Καταχώρηση πλατφόρμας zastel-spotypal στο IoTFeds σύστημα. 16

Εικόνα 6: Επεξεργασία δεδομένων πλατφόρμας zasel-spotypal. 18

Εικόνα 7: Παράδειγμα καταχώρησης IoT πόρου της πλατφόρμας spotypal-zastel για δεδομένα κυκλοφοριακής κίνησης. 19

Εικόνα 8: Αρχιτεκτονική συστήματος Smart Environment. 20

Εικόνα 9: Εφαρμογή της IoT πλατφόρμας Smart Environment σε πραγματικό σενάριο χρήσης. 21

Εικόνα 10: Καταχώρηση πλατφόρμας Smart Environment στο IoTFeds σύστημα. 22

Εικόνα 11: Εγγραφή resource στην IoTFeds πλατφόρμα για την πόλη της Θεσσαλονίκης. 24

Εικόνα 12: Λαμβανόμενα δεδομένα από το CERTH API. 25

Εικόνα 13: Λήψη περιβαλλοντικών δεδομένων από το CERTH API. 25

Εικόνα 14: Πλατφόρμα UiTOP. 26

Εικόνα 15: Εγραφή χρήστη UiTOP στο IoTFeds σύστημα. 27

Εικόνα 16: Καταχώρηση πλατφόρμας uitop-platform2 στο IoTFeds σύστημα. 28

Εικόνα 17: Το API endpoint της Marketplace υπηρεσίας που χρησιμοποιείται για την εγγραφή πόρων στην πλατφόρμα IoT (uitop-platform2). 28

Εικόνα 18: Καταχώρηση “uitopSensors” πηγής δεδομένων πλατφόρμας uitop-platform2. 29

Εικόνα 19: Καταχώρηση “uitopVirtualSensors” πηγής δεδομένων πλατφόρμας uitop-platform2). 30

Εικόνα 20: Δεδομένα έξυπνης στάθμευσης “uitopSensors” της πλατφόρμας uitop-platform2. ... 31

Εικόνα 21: Η υλοποίηση της μεθόδου στο Rap-plugin που δημιουργεί τις παρατηρήσεις (observations) για τα δεδομένα έξυπνης στάθμευσης. 32

Εικόνα 22: Δημιουργία IoTFeds ομοσπονδίας. 32

Εικόνα 23: Προσθήκη οργανισμών στην ομοσπονδία. 33

Εικόνα 24: Δημιουργία ομοσπονδίας μεταξύ των πιλοτικών IoT πλατφορμών. 33

Εικόνα 25: Ορισμός κανόνων ομοσπονδίας. 34

Εικόνα 26: Διαμοιρασμός IoT πόρου uitopSensors (uitop-platform2) στην ομοσπονδία “fed20231117b”. 34

Εικόνα 27: Διαμοιρασμός IoT πόρου uitopVirtualSensors (uitop-platform2) στην ομοσπονδία “fed20231117b”. 35

Εικόνα 28: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “uitop-platform2”. 35

Εικόνα 29: Διαμοιρασμός IoT πόρου ath_weather (smart_environment) στην ομοσπονδία “fed20231117b”. 35

Εικόνα 30: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “smart_ environment”. 36

Εικόνα 31: Διαμοιρασμός IoT πόρου traffic-data-athens-realtime (zastel-spotypal) στην ομοσπονδία “fed20231117b”. 36

Εικόνα 32: Διαμοιρασμός IoT πόρου traffic-data-athens-historical (zastel-spotypal) στην ομοσπονδία “fed20231117b”. 36

Εικόνα 33: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “zastel-spotypal”. 37

Εικόνα 34: Δημιουργία προϊόντος που συνδυάζει πόρους και των τριών πλατφορμών IoT. 37

Εικόνα 35: Αγορά προϊόντος από το Marketplace API. 38

Εικόνα 36: Έλεγχος για την τιμή ενός προϊόντος από το Marketplace API. 38

Εικόνα 37:Πρόσβαση σε ένα προϊόν από το Marketplace API. 39

Εικόνα 38: JSON αντικείμενο της mobile εφαρμογής που βασίζεται στα δεδομένα που ελήφθησαν από την IoTFeds πλατφόρμα. 41

Εικόνα 39: Η εφαρμογή στο Playstore για την λήψη της εφαρμογής. 42

Εικόνα 40: Επιλογή περιοχή και λήψη δεδομένων. 42

Εικόνα 41: Εμφάνιση δεδομένων κίνησης και parking στο χάρτη. 43

Εικόνα 42:Εμφάνιση Δεδομένων Parking. 44

Εικόνα 43:Εμφάνιση καιρικών δεδομένων και menu λειτουργιών. 44

Εικόνα 44: Ρυθμίσεις λήψης ιστορικών δεδομένων. 45

Ευρετήριο Πινάκων

Πίνακας 1: Αισθητήρες της πλατφόρμας Smart Environment και οι αντίστοιχες λαμβανόμενες τιμές τους. 20

Πίνακας 2: APIs για τη λήψη των περιβαλλοντικών δεδομένων. 23

1 Αντικείμενο και Δομή του Παραδοτέου

Στο παραδοτέο αυτό παρουσιάζεται το σενάριο χρήσης που θα εκτελεστεί για την αξιολόγηση των λειτουργιών του συστήματος IoTFeds μέσω της χρήσης των υπηρεσιών δεδομένων IoTFeds (προϊόντων δεδομένων) από μία εφαρμογή στον τομέα της Έξυπνης Πόλης με στόχο την παροχή μίας νέας υπηρεσίας προς τους χρήστες της. Το σενάριο χρήσης βασίζεται στην ανάλυση των σεναρίων που παρουσιάστηκαν στο παραδοτέο Π1.2 «Απαιτήσεις Συστήματος και Ενδιάμεση Αρχιτεκτονική Συστήματος» και την ανάπτυξη μίας υπηρεσίας που θα επιτρέπει στους χρήστες της εφαρμογής να επιλέξουν τη βέλτιστη διαδρομή για τη μετάβαση στο χώρο εργασίας τους και την επιλογή του χώρου στάθμευσης του οχήματός τους.

Το κείμενο στο παραδοτέο αυτό ακολουθεί την παρακάτω δομή:

Στο κεφάλαιο 2 παρουσιάζεται το σενάριο χρήσης που αποτελείται από την πιλοτική εφαρμογή (δημιουργία προϊόντος μέσα από την ομοσπονδιακή αγορά της IoTFeds πλατφόρμας και την δοκιμαστική εφαρμογή στον τομέα της Έξυπνης Πόλης).

Στο κεφάλαιο 3 καταγράφονται τα στάδια προετοιμασίας της πιλοτικής εφαρμογής με την ενσωμάτωση των συμμετεχουσών IoT πλατφορμών, uiTOP (δεδομένα έξυπνης στάθμευσης), Smart Environment (περιβαλλοντικά δεδομένα) και Spotypal (δεδομένα κυκλοφοριακής κίνησης), και τον διαμοιρασμό IoT πόρων (πηγών δεδομένων) τους στην ομοσπονδιακή αγορά με στόχο τη δημιουργία προϊόντων και την παροχή ενοποιημένων ροών δεδομένων που προέρχονται από πολλαπλές IoT πηγές δεδομένων.

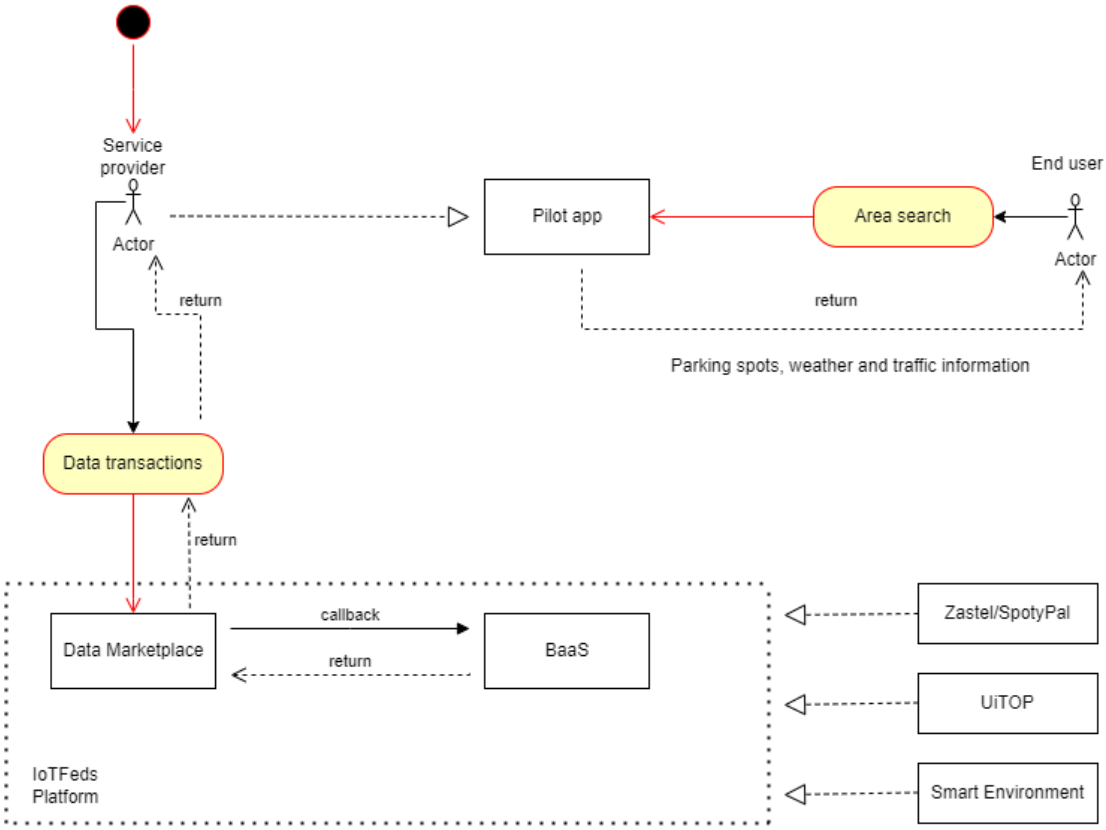
Στο κεφάλαιο 4 παρουσιάζεται η δοκιμαστική εφαρμογή στον τομέα της Έξυπνης Πόλης που συνδυάζει δεδομένα από τις τρεις πιλοτικές IoT πλατφόρμες και τεκμηριώνεται η υλοποίησή της.

Στο κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα.

2 Περιγραφή σεναρίου δοκιμής (πιλοτική και δοκιμαστική εφαρμογή)

Το σενάριο δοκιμής αποτελείται από την πιλοτική επίδειξη (επίδειξη χρήσης συστήματος IoTFeds) και την δοκιμαστική εφαρμογή (επίδειξη χρήσης εξωτερικής/τρίτης εφαρμογής). Στο παραδοτέο αυτό περιγράφεται η προετοιμασία περιβάλλοντος για την υποστήριξη του σεναρίου δοκιμής και τη δημιουργία προϊόντος από πολλαπλές ομόσπονδες πηγές δεδομένων (από πολλαπλούς παρόχους δεδομένων) καθώς και η υλοποίηση μίας εξωτερικής/τρίτης εφαρμογής (που ανήκει σε έναν πάροχο εφαρμογής), η οποία βασίζεται στην αξιοποίηση δεδομένων που αποκτήθηκαν μέσω της αγοράς ενός προϊόντος δεδομένων από μία ομοσπονδιακή αγορά IoTFeds, για την δημιουργία και παροχή νέων υπηρεσιών προς τους πελάτες της εφαρμογής.

Το πρώτο μέρος του σεναρίου δοκιμής αφορά τη λειτουργία μίας ομοσπονδιακής αγοράς με τη συνέργεια των πιλοτικών IoT πλατφορμών για τη δημιουργία IoT ομοσπονδίας και προϊόντων δεδομένων που προέρχονται από πολλαπλές πηγές δεδομένων. Τα προϊόντα που δημιουργήθηκαν διατίθενται στην ομοσπονδιακή αγορά όπου δημιουργήθηκαν για κατανάλωση από τα ομόσπονδα μέλη καθώς και στην καθολική αγορά για χρήση από τρίτους. Στο δεύτερο μέρος του σεναρίου δοκιμής ένας πάροχος εφαρμογής που συμμετέχει στο IoTFeds οικοσύστημα με τον ρόλο του καταναλωτή δεδομένων έχει τη δυνατότητα να αναζητήσει, να αγοράσει και να καταναλώσει τα επιθυμητά προϊόντα. Μετά την κατανάλωση οποιουδήποτε προϊόντος η IoTFeds πλατφόρμα επιτρέπει στον χρήστη την αξιολόγηση των προϊόντων που κατανάλωσε. Στο σενάριο δοκιμής ο πάροχος εφαρμογής είναι μέλος της ομοσπονδίας. Τα δεδομένα που αποκτήθηκαν από τον πάροχο της εφαρμογής μέσα από τη χρήση της IoTFeds πλατφόρμας χρησιμοποιούνται για την δημιουργία και παροχή νέων υπηρεσιών προς τους πελάτες της εφαρμογής. Η ροή του σεναρίου περιγράφεται στην παρακάτω εικόνα.

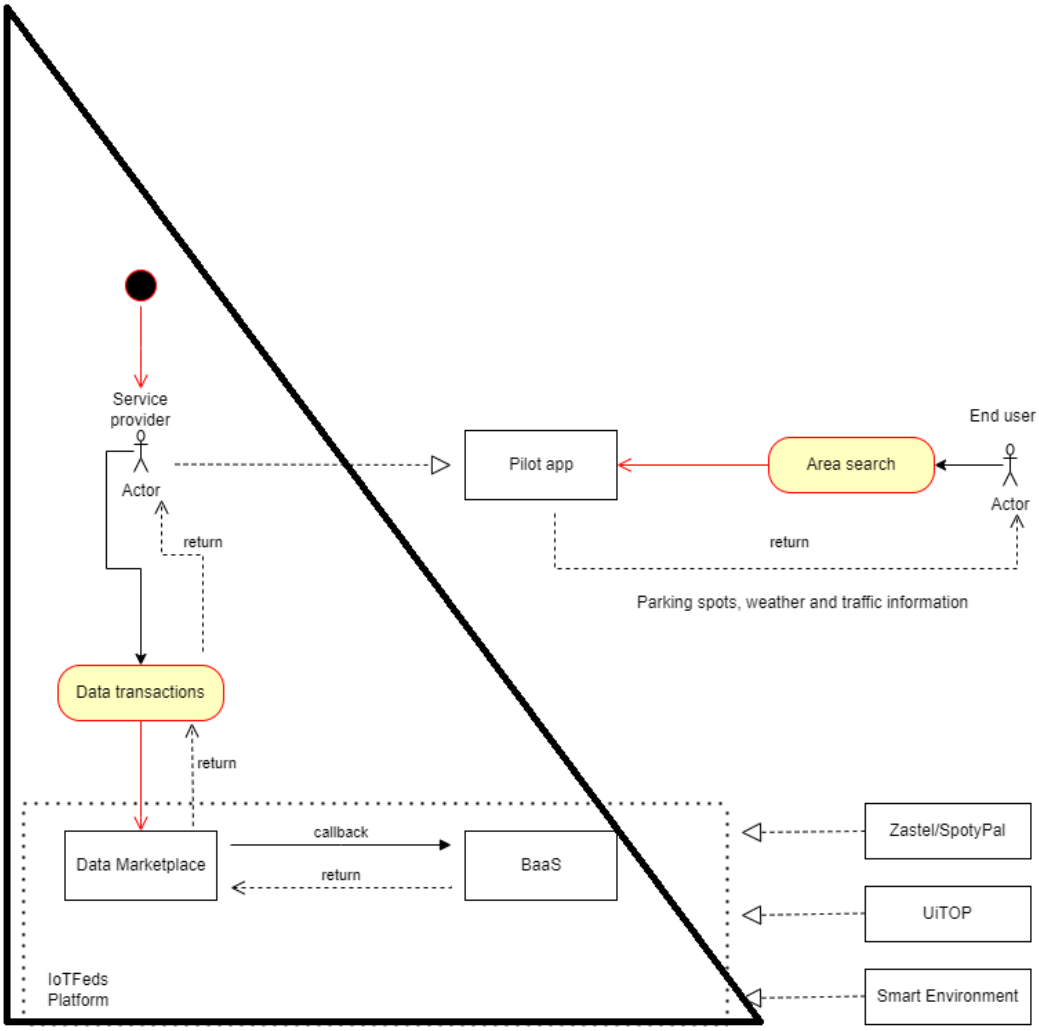


Εικόνα 1: Αρχιτεκτονική για την υλοποίηση του σεναρίου δοκιμής.

Στις επόμενες παραγράφους περιγράφονται αναλυτικά η πιλοτική επίδειξη και δοκιμαστική εφαρμογή με τη δημιουργία ομοσπονδίας για τη συνέργεια των IoT πλατφορμών και τη δημιουργία προϊόντων καθώς και την υλοποίηση της εφαρμογής όπως αναπτύχθηκε από τον πάροχο της εφαρμογής.

2.1.1 Πιλοτική επίδειξη – Δημιουργία προϊόντος από πολλαπλές ομόσπονδες πηγές δεδομένων

Το στάδιο αυτό αφορά την χρήση της IoTFeds πλατφόρμας και των σχετικών υπηρεσιών (λειτουργικοτήτων) της πλατφόρμας με στόχο την επαλήθευση της σωστής λειτουργίας της και την διασφάλιση της επιτυχούς εκτέλεσης του σεναρίου, όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 2: Προετοιμασία πιλοτικής εφαρμογής.

Κατά το στάδιο αυτό, οι τρεις πιλοτικές IoT πλατφόρμες διαμορφώνονται κατάλληλα ώστε να συνδεθούν στο IoTFeds σύστημα για να μπορούν να δημιουργήσουν ομοσπονδία και να διαμοιράσουν σε αυτήν τις πηγές δεδομένων που επιθυμούν. Συνεπώς αρχικά εξετάζονται οι λειτουργικότητες της IoTFeds πλατφόρμας όσον αφορά το επίπεδο της ομοσπονδίας (δημιουργία ομοσπονδίας, προσθήκη μέλους στην ομοσπονδία, ενημέρωση κανόνων ομοσπονδίας) και των πηγών δεδομένων (εγγραφή IoT συσκευών στο σύστημα, διαμοιρασμός συσκευών σε ομοσπονδία, ανακάλυψη IoT συσκευών και πρόσβαση στα δεδομένα). Στη συνέχεια εξετάζονται λειτουργικότητες σχετικά με την αγορά (marketplace). Στο επίπεδο αυτό δημιουργούνται προϊόντα από τον συνδυασμό πηγών δεδομένων των διαφορετικών πλατφορμών για διάθεση και χρήση στην ομοσπονδιακή αγορά. Πιο συγκεκριμένα,

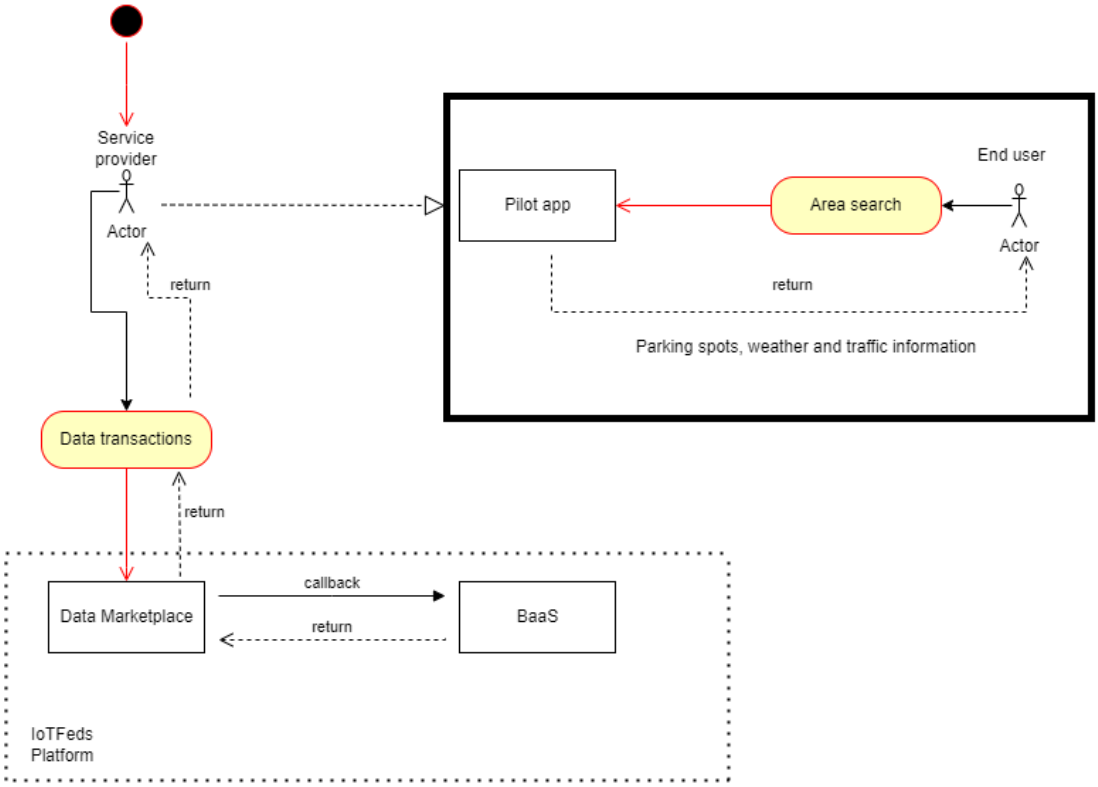
εξετάζονται λειτουργικότητες της IoTFeds πλατφόρμας σχετικά με την δημιουργία, αναζήτηση και αγορά προϊόντος καθώς και την πρόσβαση στα δεδομένα των πηγών που το απαρτίζουν. Στο Κεφάλαιο 3 παρουσιάζεται αναλυτικά η προετοιμασία της πιλοτικής εφαρμογής και δίνονται λεπτομέρειες σχετικά με τις IoT πλατφόρμες και τις πηγές δεδομένων τους, τη διαμόρφωση τους και την αξιοποίηση τους για τον έλεγχο των αντίστοιχων λειτουργικοτήτων του συστήματος.

2.1.2 Δοκιμαστική εφαρμογή – Χρήση δεδομένων προϊόντος από τρίτη εφαρμογή

Το δεύτερο στάδιο αφορά την υλοποίηση της δοκιμαστικής IoT εφαρμογής (ενός παρόχου εφαρμογής), όπως φαίνεται στο παρακάτω σχήμα, για την παροχή μίας ολοκληρωμένης λύσης που βασίζεται στην αξιοποίηση δεδομένων που ανακτήθηκαν από την χρήση της IoTFeds πλατφόρμας.

Για τη χρήση των δεδομένων του προϊόντος η δοκιμαστική εφαρμογή χρειάζεται να έχει ορίσει τις παραμέτρους χρήσης του προϊόντος, δηλαδή το χρονικό διάστημα προσπέλασης των δεδομένων, το πλήθος των προσπελάσεων και τη συχνότητα των προσπελάσεων. Μετά τον καθορισμό των παραμέτρων αυτών, η δοκιμαστική εφαρμογή μπορεί να προχωρήσει σε αγορά του προϊόντος προκειμένου να καταχωρίσει και επίσημα το δικαίωμά της για χρήση των δεδομένων. Από αυτό το σημείο και έπειτα, η εφαρμογή έχει τη δυνατότητα να προσπελάσει τα δεδομένα του προϊόντος. Με την ενέργεια αυτή θα μπορέσει να αποκτήσει πρόσβαση σε δεδομένα από πολλαπλούς παρόχους και να τα συνδυάσει προκειμένου να διαθέσει κάποιου είδους υπηρεσία προς τους χρήστες της

Πρόκειται για μία εφαρμογή που δίνει στον χρήστη την δυνατότητα να βρει εύκολα χώρους στάθμευσης και ελεύθερες θέσεις σε αυτά. Επίσης παρέχει επιπλέον πληροφορίες, όπως δεδομένα κίνησης, είτε ιστορικά είτε πραγματικού χρόνου και τέλος τα εκάστοτε καιρικά δεδομένα της περιοχής για κάθε χώρο στάθμευσης. Η υλοποίηση της δοκιμαστικής εφαρμογής, τα τεχνικά χαρακτηριστικά και περισσότερες λεπτομέρειες περιγράφονται στο Κεφάλαιο 4.



Εικόνα 3: Νέα εφαρμογή που βασίζεται στην χρήση δεδομένων από την IoTFeds πλατφόρμα.

3 Προετοιμασία περιβάλλοντος πιλοτικής εφαρμογής

Η δράση αυτή περιλαμβάνει την προετοιμασία των πλατφορμών IoT που συμμετέχουν στην πιλοτική ομοσπονδία IoT. Πιο συγκεκριμένα, κάθε πλατφόρμα IoT εγκαθιστά το ενδιάμεσο λογισμικό και επιλέγει ποιους πόρους θα κάνει διαθέσιμους στην ομοσπονδία, βάσει του σεναρίου χρήσης από την ΕΕ1.

3.1 Διαμόρφωση πλατφορμών IoT

Κατά το στάδιο αυτό οι πιλοτικές IoT πλατφόρμες διαμορφώθηκαν κατάλληλα για την διαλειτουργικότητα τους με την IoTFeds πλατφόρμα ακολουθώντας τις διαθέσιμες οδηγίες ενσωμάτωσης (integration) που περιγράφηκαν στο Π1.4 και εκτελώντας τα παρακάτω βήματα:

1. Δημιουργία χρήστη (οργανισμού) για τον πάροχο πλατφόρμας στην IoTFeds πλατφόρμα μέσω της γραφικής διεπαφής χρήστη administration GUI: https://symbiote-core.iotfeds.intracom-telecom.com/administration
2. Καταχώρηση της IoT πλατφόρμας στο σύστημα και λήψη των αρχείων διαμόρφωσης για την εγκατάσταση του ενδιάμεσου λογισμικού (symbIoTe cloud).
3. Εγκατάσταση και λειτουργία του symbIoTe cloud λογισμικού.
4. Υλοποίηση του RAP plugin για την εκάστοτε IoT πλατφόρμα με βάση το διαθέσιμο παράδειγμα[1]. Η μικρο-υπηρεσία RAP plugin αφορά την υλοποίηση των διεπαφών για ακρόαση γεγονότων (listeners) από το ενδιάμεσο λογισμικό του symbIoTe προς την μεριά της εκάστοτε IoT πλατφόρμας για την ανάγνωση και επεξεργασία των δεδομένων από τις πραγματικές πηγές δεδομένων της πλατφόρμας.
5. Ενσωμάτωση της μικρο-υπηρεσίας RAP plugin στο λειτουργικό symbIoTe cloud λογισμικό της IoT πλατφόρμας.
6. Καταχώρηση πηγών δεδομένων στο IoTFeds σύστημα.

3.1.1 Πλατφόρμα Spotypal-Zastel και διάθεση δεδομένων κίνησης

3.1.1.1 Περιγραφή πλατφόρμας Spotypal-Zastel

Το Spotypal ξεκίνησε το 2018 και αποτελεί ένα σύστημα monitoring, με κύριο στόχο τη διευκόλυνση των χρηστών να εντοπίσουν τα χαμένα αντικείμενα ή τα κατοικίδιά τους μέσω προηγμένης τεχνολογίας κινητών και cloud.

Πρόκειται για μια εφαρμογή που βασίζεται σε mobile application και απαιτεί τη χρήση της συσκευής Spotypal. Ο χρήστης μπορεί να κατεβάσει την εφαρμογή από το Play Store, και μετά τη δημιουργία λογαριασμού, πρέπει να συσχετίσει τη συσκευή με τον λογαριασμό του. Το σύστημα παρέχει εκτενείς δυνατότητες και χαρακτηριστικά. Στο πλαίσιο του IoTFeds, η βάση δεδομένων του Spotypal ενημερώνεται σχετικά με τις θέσεις των συσκευών με περιοδικά χρονικά διαστήματα, χρησιμοποιώντας ανωνυμοποιημένες διαδικασίες.

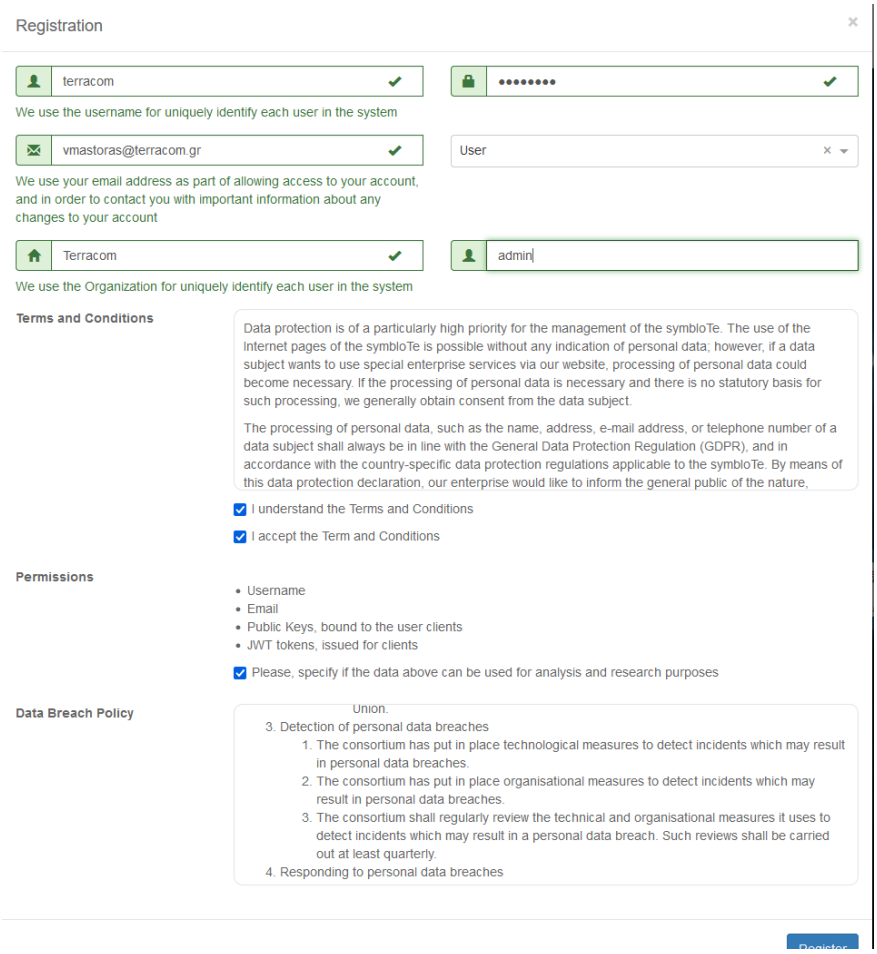
Τα δεδομένα αυτά, με τη χρήση του Spotypal API, αποστέλλονται στην IoT πλατφόρμα Zastel. Η Zastel ανήκει στη Terracom και στην περίπτωση του IoTFeds, αποτελεί το αποθετήριο των δεδομένων που συλλέγονται από την εφαρμογή Spotypal, παρέχοντας επίσης εργαλεία που έχουν αναπτυχθεί ειδικά για το IoTFeds και αναλύονται εκτενέστερα παρακάτω.

[1] https://github.com/symbiote-h2020/RapPluginExample

3.1.1.2 Διασύνδεση πλατφόρμας Spotypal-Zastel στο IoTFeds σύστημα

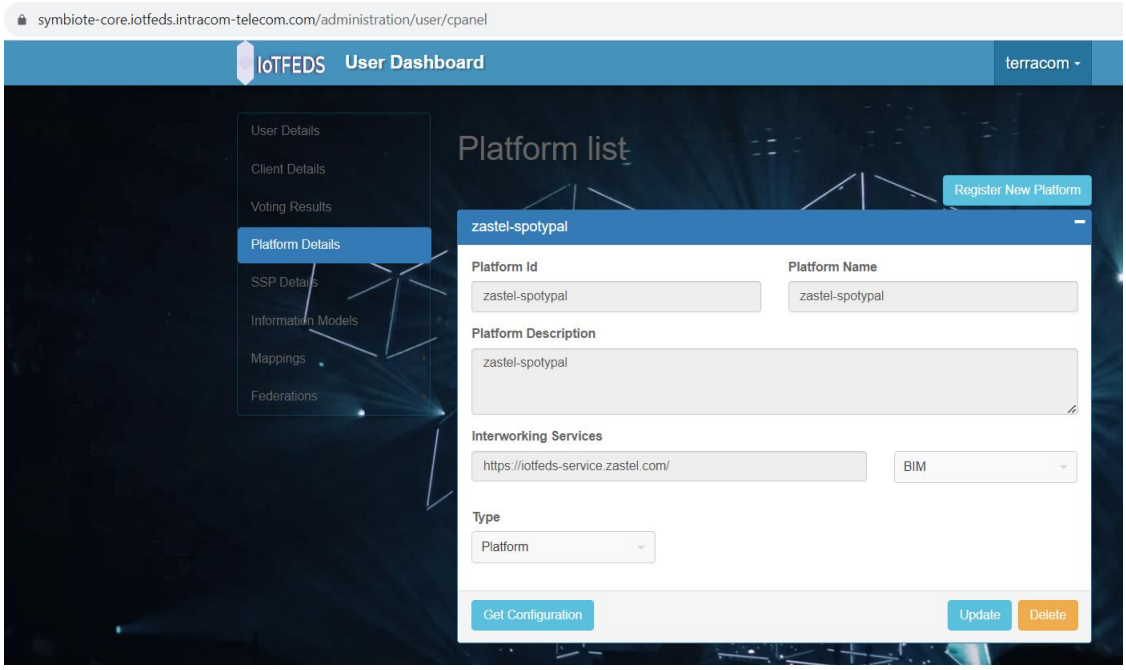
Στην ενότητα αυτή περιγράφονται τα βήματα για τη διαμόρφωση και λειτουργία της IoT πλατφόρμας spotypal-zastel για την διάθεση των δεδομένων κίνησης στο IoTFeds σύστημα.

1. Αρχικά δημιουργήθηκε ο χρήστης terracom για τον πάροχο πλατφόρμας zastel-spotypal στην IoTFeds πλατφόρμα μέσω της γραφικής διεπαφής χρήστη administration GUI: https://symbiote-core.iotfeds.intracom-telecom.com/administration.



Εικόνα 4: Δημιουργία χρήστη terracom.

2. Στη συνέχεια καταχωρήθηκε η spotypal-zastel πλατφόρμα στο σύστημα επιλέγοντας το μοντέλο πληροφορίας BIM (Best Practice Information Model) που περιέχει το μοντέλο smart city, όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα 5: Καταχώρηση πλατφόρμας zastel-spotypal στο IoTFeds σύστημα.

3. Από την καταχωρημένη εγγραφή της IoT πλατφόρμας στο σύστημα, έγινε λήψη των απαραίτητων αρχείων διαμόρφωσης (configuration) τα οποία αποτέλεσαν τη βάση για την εγκατάσταση του ενδιάμεσου λογισμικού symbIoTe cloud. Το βήμα αυτό περιλαμβάνει τη δημιουργία docker volumes, τα οποία αποτελούν δομές αποθήκευσης της πληροφορίας που διατηρεί το symbIoTe cloud, την αλλαγή ορισμένων URL και άλλων παραμέτρων στο configuration, όπως επίσης και της χρήσης των κατάλληλων διαπιστευτηρίων χρήστη, για την ομαλή επικοινωνία μεταξύ των διαφόρων υποσυστημάτων του symbIoTe cloud. Επιβεβαιώθηκε ότι το λογισμικό εγκαταστάθηκε επιτυχώς: https://iotfeds-service.zastel.com/paam/get_available_aams με την παρακολούθηση (monitoring) της κατάστασης των επιμέρους στοιχείων λογισμικού μέσω της εξέτασης του αρχείου καταγραφής (logs) για την επιβεβαίωση της ορθής λειτουργίας τους.

4. Ακολούθως, διεκπεραιώθηκε η υλοποίηση του RAP plugin για την πλατφόρμα zastel-spotypal με βάση το διαθέσιμο παράδειγμα[2]. Στο βήμα αυτό υλοποιήθηκαν όλες οι απαραίτητες μέθοδοι σύμφωνα με τις ανάγκες διάθεσης των δεδομένων κίνησης για την επίτευξη ολοκληρωμένης επικοινωνίας μεταξύ του ενδιάμεσου λογισμικού (στοιχείο λογισμικού symbIoTe RAP) και της πλατφόρμας SpotyPal, που επιτρέπει την πρόσβαση στα πραγματικά δεδομένα κίνησης που διαθέτει η πλατφόρμα. Οι μέθοδοι που αναπτύχθηκαν φαίνονται ενδεικτικά στις παρακάτω εικόνες:

[2] https://github.com/symbiote-h2020/RapPluginExample

```java
private JSONObject fetchTrafficDataFromZastel(int numberOfObservations, String resourceId) throws MalformedURLException, IOException, ProtocolException {
    URL x = new URL(zastelURL + "?observations=" + numberOfObservations + "&resourceId=" + resourceId);
    HttpURLConnection conn = (HttpURLConnection) x.openConnection();
    conn.setRequestProperty("x-access-token", zastelAccessToken);
    conn.setRequestMethod("GET");

    InputStream is = null;
    Scanner s = null;
    try {
        is = conn.getInputStream();
        s = new Scanner(is).useDelimiter("\\A");
        String result = s.hasNext() ? s.next() : "";

        return new JSONObject(result);
    } finally {
        is.close();
        s.close();
    }
}
```

Εικόνα 6: Λήψη δεδομένων κίνησης από την πλατφόρμα zastel-spotypal.

Όταν το RAP plugin λαμβάνει από το ενδιάμεσο λογισμικό κάποιο αίτημα για την προσπέλαση των κυκλοφοριακών δεδομένων, αποστέλλει ένα αίτημα προς την πλατφόρμα SpotyPal, μέσω της οποίας ανακτά τα δεδομένα, και αφού αναλύσει (parsing) την απάντηση της πλατφόρμας SpotyPal, διαμορφώνει την τελική απάντηση υπό τη μορφή ενός observation, όπως αναμένεται από το ενδιάμεσο λογισμικό. Το SpotyPal αξιοποιεί το built-in μοντέλο πληροφορίας για τα κυκλοφοριακά δεδομένα, επομένως δεν απαιτούνταν κάποια ιδιαίτερη μετατροπή για τη επίτευξη σημασιολογικής συμβατότητας με το υπόλοιπο οικοσύστημα του IoTFeds.

```java
private ArrayList<Observation> retrieveObservations(int numberOfObservations, String resourceId) {
  try {
    System.out.println("Fetching data from Zastel...");
    JSONObject trafficDataResponse = fetchTrafficDataFromZastel(numberOfObservations, resourceId);
    System.out.println("Data retrieved successfully from Zastel");

    return parseTrafficDataFromZastel(trafficDataResponse, resourceId);
  } catch (Exception e) {
    e.printStackTrace();
  }

  return null;
}
```

Εικόνα 8: Ανάκτηση observations.

```java
private ArrayList<Observation> parseTrafficDataFromZastel(JSONObject pinpointsDictionary, String resourceId) throws RapPluginException {
    System.out.println("Parsing data from Zastel");
    Location observationLocation = resourceId == ioanninaTrafficDataResourceId ?
      new WGS84Location(39.657126, 20.857933, 0, "Ioannina", Arrays.asList("City of Ioannina")) :
      new WGS84Location(37.975975, 23.725818, 0, "Athens", Arrays.asList("City of Athens"));
    TimeZone utcTimeZone = TimeZone.getTimeZone("UTC");
    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");
    dateFormat.setTimeZone(utcTimeZone);
    Date date = new Date();
    String timestamp = dateFormat.format(date);
    long ms = date.getTime() - 1000;
    date.setTime(ms);
    String samplet = dateFormat.format(date);
    ArrayList<Observation> observations = new ArrayList<Observation>();

    Iterator<String> dictionaryKeys = pinpointsDictionary.keys();
    int counter = pinpointsDictionary.length();
    int maxPinpointsValues = 1000;
    int numberOfPinpointsAdded = maxPinpointsValues;
    while (dictionaryKeys.hasNext()) {
        String dictionaryKey = dictionaryKeys.next();
        counter--;
        if (counter > 2) continue;

        JSONArray observationPinpoints = (JSONArray) pinpointsDictionary.get(dictionaryKey);
        List<ObservationValue> observationValues = new ArrayList<ObservationValue>();

        int loopLimit = Math.min(observationPinpoints.length(), numberOfPinpointsAdded);
        for (int i = loopLimit - 1; i >= 0; i--) {
            ObservationValue observationValue = new ObservationValue(
                observationPinpoints.get(i).toString(),
                new Property("-", "-", Arrays.asList("-")),
                null);

            observationValues.add(observationValue);
            numberOfPinpointsAdded--;
        }

        observations.add(new Observation(resourceId, observationLocation, timestamp, samplet, observationValues));
    }
    System.out.println("Parsed traffic data successfully. Returning " + (maxPinpointsValues - numberOfPinpointsAdded) + " pinpoints.");

    return observations;
}
```

Εικόνα 6: Επεξεργασία δεδομένων πλατφόρμας zasel-spotypal.

5. Έπειτα από την ολοκλήρωση της υλοποίησης του RAP plugin, δημιουργήθηκε το docker image για την εγκατάστασή του ως μικρο-υπηρεσία. Η δημιουργία του έγινε αξιοποιώντας το Dockerfile που χρησιμοποιεί τον πηγαίο κώδικά του. Το docker image του RAP plugin ενσωματώθηκε στο εγκαταστημένο symbIoTe cloud της πλατφόρμας spotypal-zastel και ελέγχθηκε η επιτυχής καταχώρηση των πηγών δεδομένων για τη διάθεση των κυκλοφοριακών δεδομένων κίνησης όπως περιγράφεται αναλυτικά στο επόμενο βήμα. Οι καταχωρημένες πηγές δεδομένων είναι διαθέσιμες: https://iotfeds-service.zastel.com/rh/resources.

```dockerfile
FROM openjdk:8-jre-alpine

WORKDIR /home

ENV componentName "RapPluginExample"
ENV componentVersion 2.0.0

RUN apk --no-cache add \
    git \
    unzip \
    wget \
    bash \
    && echo "Downloading $componentName $componentVersion" \
    && wget "https://jitpack.io/com/github/symbiote-h2020/$componentName/$componentVersion/$componentName-$componentVersion-run.jar"

CMD java $JAVA_HTTP_PROXY $JAVA_HTTPS_PROXY $JAVA_NON_PROXY_HOSTS -jar $(ls *run.jar)
```

Εικόνα 10:DockerFile.για το RAP plugin.

Η καταχώρηση των πηγών δεδομένων στο IoTFeds σύστημα έγινε με χρήση του marketplace/add/L1/resource. Ένα παράδειγμα με το σώμα (body) της κλήσης αυτής για την καταχώρηση IoT πόρου για δεδομένα κυκλοφοριακής κίνησης φαίνεται στην παρακάτω εικόνα. Ο IoT πόρος στη συνέχεια είναι προσβάσιμος από την μικρο-υπηρεσία marketplace.

```
{
  "cloudResource": {
    "internalId": "traffic-data-athens-historical",
    "pluginId": "zastel-spotypal",
    "accessPolicy": {
      "policyType": "PUBLIC",
      "requiredClaims": {}
    },
    "filteringPolicy": {
      "policyType": "PUBLIC",
      "requiredClaims": {}
    },
    "resource": {
      "@c": ".MobileSensor",
      "observesProperty": [

      ],
      "id": null,
      "name": "Traffic Data Athens Historical",
      "description": [
        "Traffic Data Athens Historical"
      ],
      "interworkingServiceURL": "https://iotfeds-service.zastel.com/",
      "locatedAt": {
        "@c": ".WGS84Location",
        "longitude": 37.960781,
        "latitude": 23.861427,
        "name": "Europe/Athens",
        "description": [
          "Athens"
        ]
      }
    }
  },
  "credentials": {
    "clientId": "terracom",
    "password": "******",
    "platformId": "zastel-spotypal",
    "username": "terracom"
  }
}
```


Εικόνα 7: Παράδειγμα καταχώρησης IoT πόρου της πλατφόρμας spotypal-zastel για δεδομένα κυκλοφοριακής κίνησης.

3.1.2 Πλατφόρμα Smart Environment και διάθεση περιβαλλοντικών δεδομένων

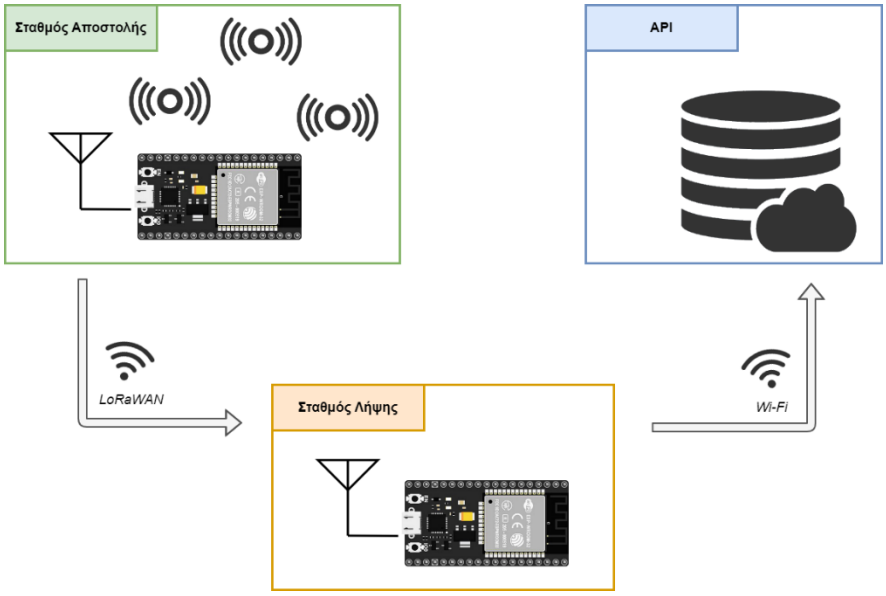
Στη συγκεκριμένη ενότητα παρουσιάζεται αναλυτικά τόσο η λειτουργία, όσο και η διαδικασία διαμόρφωσης της IoT πλατφόρμας Smart Environment για τη διάθεση περιβαλλοντικών δεδομένων (καιρού και ποιότητας αέρα) στο IoTFeds σύστημα.

3.1.2.1 Περιγραφή πλατφόρμας Smart Environment

Στα πλαίσια του έργου IoTFeds πραγματοποιήθηκε η υλοποίηση εφαρμογής Smart Environment, η οποία περιλαμβάνει τόσο την επιλογή του κατάλληλου υλικού (hardware), όσο και την υλοποίηση του αντίστοιχου λογισμικού (software) για τη συλλογή των περιβαλλοντικών δεδομένων. Η συγκεκριμένη υλοποίηση πραγματοποιήθηκε για τη λήψη δεδομένων καιρού και ποιότητας αέρα για την περιοχή της Θεσσαλονίκης και τοποθετήθηκε σε πραγματικές συνθήκες σε εξωτερικό περιβάλλον στην περιοχή της Νέας Κρήνης για τις ανάγκες των πιλοτικών δοκιμών του έργου.

Πιο συγκεκριμένα, το IoT σύστημα Smart Environment αποτελείται από 2 επιμέρους σταθμούς, το σταθμό αποστολής δεδομένων, ο οποίος αποτελεί το βασικό κύκλωμα και είναι κατάλληλα σχεδιασμένος για τοποθέτηση σε εξωτερικό χώρο για τη λήψη των μετρήσεων, και το σταθμό λήψης, ο οποίος λαμβάνει τα περιβαλλοντικά δεδομένα από τον εξωτερικό σταθμό, ενώ για την καταγραφή και την απεικόνιση των συλλεγόμενων δεδομένων υλοποιήθηκε το

CERTH API[3]. Μια γενική εικόνα της αρχιτεκτονικής της πλατφόρμας Smart Environment παρέχεται στο διάγραμμα στην Εικόνα 8.



Εικόνα 8: Αρχιτεκτονική συστήματος Smart Environment.

Όπως φαίνεται και στο διάγραμμα στην Εικόνα 8, ο σταθμός αποστολής συλλέγει τα περιβαλλοντικά δεδομένα από την τοποθεσία του και τα αποστέλλει ασύρματα μέσω LoRaWAN πρωτόκολλου επικοινωνίας στο σταθμό λήψης που βρίσκεται σε εσωτερικό χώρο. Ο συγκεκριμένος τύπος ασύρματων επικοινωνιών επιτρέπει την επικοινωνία των δύο σταθμών σε πολύ μακρινές αποστάσεις, σε έως και 5km σε αστικές περιοχές, κάτω από ιδανικές συνθήκες, καταναλώνοντας ελάχιστη ενέργεια κατά τη μετάδοση των δεδομένων, όντας ιδιαίτερα χρήσιμος σε IoT υλοποιήσεις. Τέλος, τα λαμβανόμενα δεδομένα προωθούνται μέσω Wi-Fi από το σταθμό λήψης στο CERTH API για οπτικοποίηση και αποθήκευση για μετέπειτα χρήση.

Όσον αφορά το υλικό της πλατφόρμας Smart Environment, χρησιμοποιήθηκαν 2 ESP NodeMCU-32s μικροελεγκτές, 2 SX1278 LoRa modules και 2 κεραίες 433MHz, ένα για κάθε σταθμό, για τη συλλογή, μετάδοση και λήψη των περιβαλλοντικών δεδομένων. Παράλληλα, στην περίπτωση του σταθμού αποστολής, το κύκλωμα περιλαμβάνει επιπρόσθετα 8 περιβαλλοντικούς αισθητήρες IoT χαμηλού κόστους για τη συλλογή 10 περιβαλλοντικών μετρήσεων. Ο Πίνακας 1 περιλαμβάνει αναλυτικά όλους τους αισθητήρες της IoT πλατφόρμας, καθώς και τις αντίστοιχες λαμβανόμενες τιμές από κάθε αισθητήρα. Αναλυτικότερες πληροφορίες σχετικά με τη συνδεσμολογία και τη λειτουργία του συστήματος Smart Environment παρέχονται στο (Polymeni, et al., 2023).

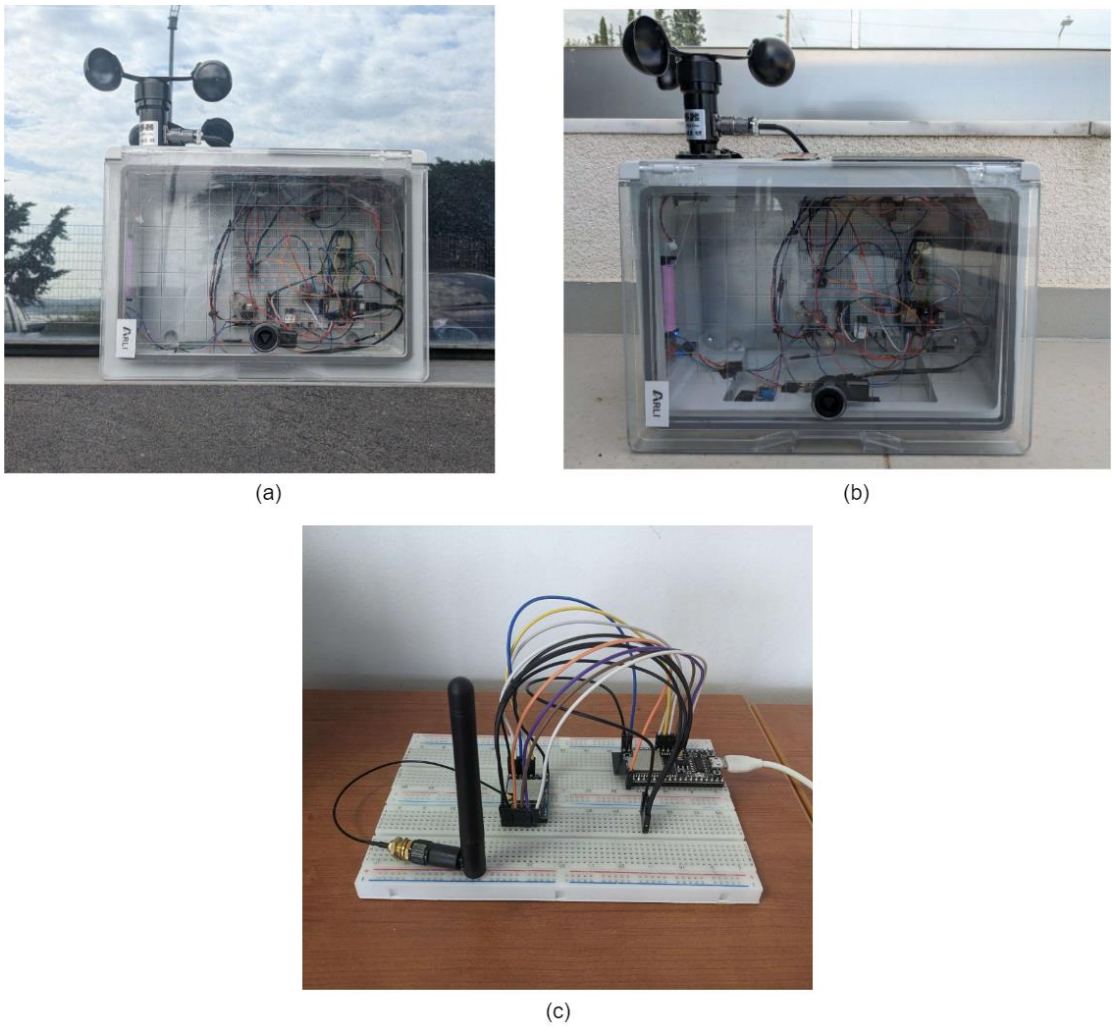
Πίνακας 1: Αισθητήρες της πλατφόρμας Smart Environment και οι αντίστοιχες λαμβανόμενες τιμές τους.

Αισθητήρας	Λαμβανόμενη τιμή
DHT22	Θερμοκρασία, υγρασία
BH1750FVI	Φωτεινότητα
Αναλογικό ανεμόμετρο	Ταχύτητα ανέμου
YL-83	Βροχόπτωση
BMP280	Ατμοσφαιρική πίεση
MQ-7	Συγκέντρωση CO

[3] http://balkanpi.iti.gr/iotfeds-api/v1/measurements

MQ-135	Συγκέντρωση CO_2
PMS5003	Συγκέντρωση $PM_{2.5}$

Όπως φαίνεται και από την Εικόνα 8, ο σταθμός αποστολής αποτελεί το βασικό σύστημα του μετεωρολογικού σταθμού Smart Environment, ο οποίος είναι κατάλληλα σχεδιασμένος ώστε να λειτουργεί αυτόνομα σε εξωτερικούς χώρους, χωρίς να είναι απαραίτητη η ενεργειακή τροφοδότησή του από το ηλεκτρικό δίκτυο. Ο συγκεκριμένος σταθμός, πέρα από τους προαναφερθέντες αισθητήρες, περιλαμβάνει τη χρήση ενός ηλιακού πάνελ για τη λειτουργία του, καθώς και 2 επαναφορτιζόμενες μπαταρίες τύπου 18650, ώστε να αποθηκεύεται η ηλιακή ενέργεια για μετέπειτα χρήση. Παράλληλα, για εξοικονόμηση όσο το δυνατόν περισσότερης ενέργειας, ο σταθμός αποστολής απενεργοποιείται μεταξύ των λαμβανόμενων μετρήσεων και ενεργοποιείται μόνο για τη λήψη της εκάστοτε μέτρησης, η οποία πραγματοποιείται κάθε 30’.



Εικόνα 9: Εφαρμογή της IoT πλατφόρμας Smart Environment σε πραγματικό σενάριο χρήσης.

Στην Εικόνα 9 παρουσιάζεται ο IoT σταθμός Smart Environment σε λειτουργία. Για την εύρεση του κατάλληλου σεναρίου χρήσης έγιναν διάφορες δοκιμές σχετικά τόσο με την κατάλληλη τοποθέτηση της εξωτερικής μονάδας (Εικόνα 9(a) και Εικόνα 9(b)) όσο και με την επαρκή τροφοδότησή της ώστε να είναι πλήρως αυτόνομη. Στο συγκεκριμένο πλαίσιο πραγματοποιήθηκαν διάφορες τροποποιήσεις και εντοπίστηκε πως είναι απαραίτητη η ενσωμάτωση ενός δεύτερου ηλιακού πάνελ, καθώς και 2 επιπλέον επαναφορτιζόμενων μπαταριών (σύνολο 4 μπαταρίες) για την επαρκή τροφοδότηση του σταθμού αποστολής.

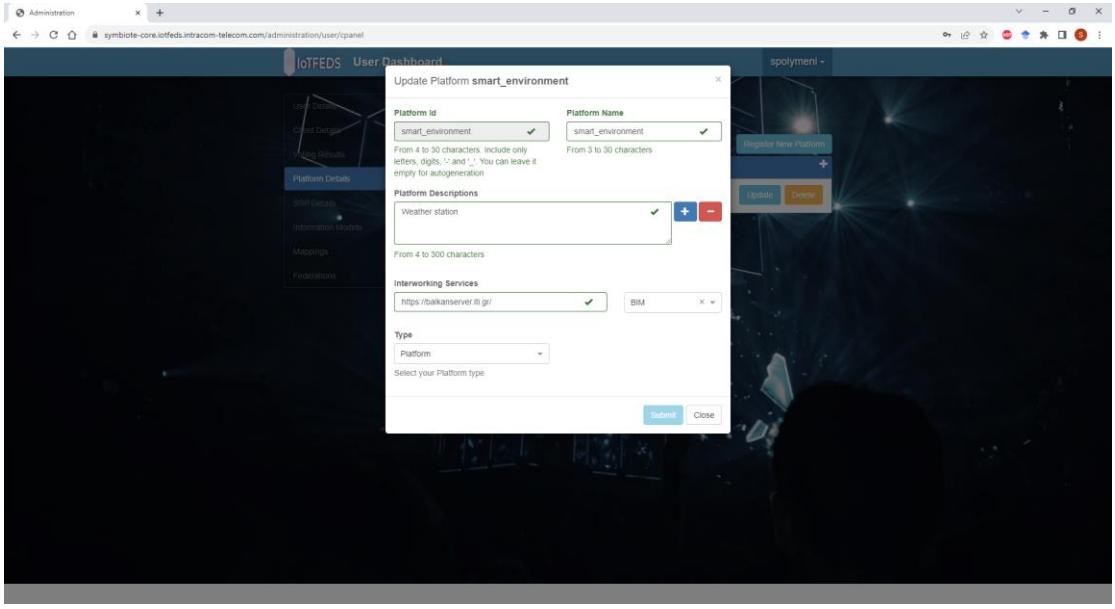
Τέλος, επιπρόσθετες μελέτες γίνονται σχετικά με το σταθμό λήψης και την ενδεχόμενη αντικατάσταση του NodeMCU-32s μικροελεγκτή με ένα Raspberry Pi 4, για τη παροχή μεγαλύτερης υπολογιστικής ισχύος, με στόχο την υλοποίηση εξελιγμένων μοντέλων πρόβλεψης περιβαλλοντικών δεδομένων.

3.1.2.2 Διασύνδεση πλατφόρμας Smart Environment στο IoTFeds σύστημα

Η IoT πλατφόρμα Smart Environment που περιγράφηκε παραπάνω ενσωματώνεται στο IoTFeds σύστημα για τη διάθεση των συλλεγόμενων περιβαλλοντικών δεδομένων στο marketplace. Για τη συγκεκριμένη διαδικασία, ακολουθήθηκε μια συγκεκριμένη ροή εργασιών από την αρχική εγκατάσταση του symbIoTe cloud μέχρι και την τελική διάθεση των δεδομένων στο σύστημα.

Αρχικά, το πρώτο βήμα της διαδικασίας περιλαμβάνει την εγγραφή του παρόχου της πλατφόρμας στη διαχειριστική διεπαφή του IoTFeds συστήματος. Για το σκοπό αυτό, δημιουργήθηκε ο χρήστης «spolymeni» ως πάροχος της πλατφόρμας Smart Environment στην IoTFeds πλατφόρμα μέσω του Administration GUI: https://symbiote-core.iotfeds.intracom-telecom.com/administration.

Στη συνέχεια, έγινε καταχώρηση της πλατφόρμας smart environment στο σύστημα από την καρτέλα *Platform Details* > *Register New Platform*, επιλέγοντας το μοντέλο πληροφορίας Smart City BIM, το μοντέλο πληροφορίας Έξυπνης Πόλης που περιγράφηκε στο Π2.1 «Αναφορά Μηχανισμών Διαχείρισης Ομοσπονδιών IoT και Σημασιολογικής Διαλειτουργικότητας» το οποίο αφορά την επέκταση του βασικού μοντέλου πληροφορίας του symbIoTe με οντολογίες στον τομέα της έξυπνης πόλης καθώς και προσθήκες με βάση τα μοντέλα πληροφορίας των πιλοτικών πλατφορμών. Η καταχώρηση της πλατφόρμας μέσω του Administration GUI παρουσιάζεται στην Εικόνα 10 που ακολουθεί.



Εικόνα 10: Καταχώρηση πλατφόρμας Smart Environment στο IoTFeds σύστημα.

Με την ολοκλήρωση της καταχώρησης της πλατφόρμας στο IoTFeds σύστημα, πραγματοποιήθηκε λήψη του .zip αρχείου που περιλαμβάνει όλα τα απαραίτητα αρχεία διαμόρφωσης της πλατφόρμας βάσει των οποίων επιτυγχάνεται η εγκατάσταση του ενδιάμεσου λογισμικού symbIoTe cloud. Ωστόσο, η εγκατάσταση του symbIoTe cloud προϋποθέτει επιπλέον και τη δημιουργία των docker volumes που περιέχουν τις απαραίτητες πληροφορίες για το symbIoTe cloud, την αλλαγή ορισμένων βασικών URLs και άλλων παραμέτρων στα αρχεία του configuration για την επιτυχή διασύνδεση του server της πλατφόρμας Smart

Environment με το symbIoTe cloud, καθώς επίσης και την ύπαρξη κατάλληλων διαπιστευτηρίων χρήστη για την επίτευξη της ομαλούς επικοινωνίας των δύο υποσυστημάτων. Οι συγκεκριμένες οδηγίες περιγράφονται αναλυτικά σε προηγούμενο παραδοτέο (Π1.4).

Με την ολοκλήρωση της επιτυχούς εγκατάστασης του symbIoTe cloud, ακολούθησε η υλοποίηση του RAP plugin για την πλατφόρμα Smart Environment, βάσει αντίστοιχου παραδείγματος που είναι διαθέσιμο online[4].

Για τις ανάγκες της διαμόρφωσης του RAP plugin για το Smart Environment, απαραίτητο βήμα αποτέλεσε η υλοποίηση μίας νέας υπηρεσίας (WeatherService) για την λήψη, το parsing και την αποστολή των περιβαλλοντικών δεδομένων πραγματικού χρόνου για 5 διαφορετικές πόλεις της Ελλάδας (Θεσσαλονίκη, Αθήνα, Ιωάννινα, Βόλος και Πάτρα), τα οποία λαμβάνονται από το CERTH API. Πιο συγκεκριμένα, όπως αναφέρθηκε και την προηγούμενη ενότητα 3.1.2.1 για την πόλη της Θεσσαλονίκης, τα δεδομένα που λαμβάνονται από το CERTH API αφορούν πραγματικές μετρήσεις από τον σταθμό Smart Environment. Οι περιβαλλοντικές τιμές για τις υπόλοιπες 4 πόλεις λαμβάνονται από εξωτερικό API, το OpenWeatherMap API, βάσει των συντεταγμένων κάθε περιοχής και στη συνέχεια αποθηκεύονται στην ίδια βάση με τα δεδομένα της Θεσσαλονίκης με διαφορετικό nodeID για κάθε πόλη. Ο Πίνακας 2 που ακολουθεί περιλαμβάνει αναλυτικά το API για τα δεδομένα κάθε περιοχής.

Πίνακας 2: APIs για τη λήψη των περιβαλλοντικών δεδομένων.

API	HTTP Request	Δεδομένα	Πόλη
CERTH API	GET http://balkanpi.iti.gr/iotfeds-api/v1/measurements	Καιρός και ποιότητα αέρα	Θεσσαλονίκη
OpenWeatherMap API	GET https://api.openweathermap.org/data/2.5/weather?lat={}&lon={}&appid={}&units=metric	Καιρός	Αθήνα, Ιωάννινα, Βόλος, Πάτρα
OpenWeatherMap API	GET http://api.openweathermap.org/data/2.5/air_pollution?lat={}&lon={}&appid={}&units=metric	Ποιότητα αέρα	Αθήνα, Ιωάννινα, Βόλος, Πάτρα

Στη συνέχεια, όπως προαναφέρθηκε, τα δεδομένα από κάθε περιοχή συγκεντρώνονται στο CERTH API με διαφορετικό nodeID για κάθε πόλη ως εξής: (i) «node: 0» για τη Θεσσαλονίκη, (ii) «node: 1» για την Αθήνα , (iii) «node: 2» για τα Ιωάννινα, (iv) «node: 3» για την Πάτρα και (v) «node: 4» για το Βόλο. Ως αποτέλεσμα, οι πιο πρόσφατες τιμές για κάθε περιοχή λαμβάνονται μέσω του iotfeds-api/v1/getlatest με το αντίστοιχο nodeID. Ενδεικτικά, για την περιοχή της Θεσσαλονίκης, η τελευταία τιμή από τα περιβαλλοντικά δεδομένα παρέχεται μέσω του: http://balkanpi.iti.gr/iotfeds-api/v1/getlatest/0.

Αντίστοιχα, για λόγους συνοχής, δημιουργήθηκαν συνολικά 5 επιμέρους περιβαλλοντικά resources, ένα για κάθε μία από τις προαναφερθείσες πόλεις ως εξής: (i) «skg_weather» για τη Θεσσαλονίκη, (ii) «ath_weather» για την Αθήνα, (iii) «ioa_weather» για τα Ιωάννινα, (iv) «vol_weather» για το Βόλο και (v) «gpa_weather» για την Πάτρα. Καθένα από τα resources αυτά περιλαμβάνει 10 περιβαλλοντικές τιμές και υπάρχει ήδη εγγεγραμμένο στο IoTFeds σύστημα πριν την εκκίνηση της διαδικασίας λήψη των δεδομένων. Στην Εικόνα 11 παρουσιάζεται ενδεικτικά το resource για την πόλη της Θεσσαλονίκης. Η καταχώρηση των resources έγινε μέσω του endpoint: https://symbiote-core.iotfeds.intracom-telecom.com/symbioteapi/resources/add/l1

[4] https://github.com/symbiote-h2020/RapPluginExample

```
"cloudResource": {
  "internalId": "skg_weather",
  "pluginId": "smart_environment",
  "accessPolicy": {
    "policyType": "PUBLIC",
    "requiredClaims": {}
  },
  "filteringPolicy": {
    "policyType": "PUBLIC",
    "requiredClaims": {}
  },
  "resource": {
    "@c": ".StationarySensor",
    "observesProperty": [
      "temperature",
      "humidity",
      "illuminance",
      "windSpeed",
      "co",
      "co2",
      "dry",
      "rainPercentage",
      "pm2_5",
      "atmosphericPressure"
    ],
    "id": null,
    "name": "Weather station/Thessaloniki",
    "description": [
      "Weather station at Thessaloniki"
    ],
    "interworkingServiceURL": "https://balkanserver.iti.gr",
    "locatedAt": {
      "@c": ".WGS84Location",
      "longitude": 22.994627,
      "latitude": 40.563547,
      "name": "Europe/Thessaloniki",
      "description": [
        "Weather station at Nea Krini, Thessaloniki"
      ]
    },
    "featureOfInterest": {
      "name": "Thessaloniki",
      "description": [
        "Weather data of Thessaloniki"
      ],
      "hasProperty": [
        "temperature",
        "humidity",
        "illuminance",
        "windSpeed",
        "co",
        "co2",
        "dry",
        "rainPercentage",
        "pm2_5",
        "atmosphericPressure"
      ]
    }
  }
},
```


Εικόνα 11: Εγγραφή resource στην IoTFeds πλατφόρμα για την πόλη της Θεσσαλονίκης.

Για τη λήψη των περιβαλλοντικών δεδομένων από το CERTH API, δημιουργήθηκε η μέθοδος «readResource», η οποία ελέγχει το ID από κάθε resource (π.χ. «skg_weather») και βάσει αυτού επιστρέφει την τελευταία τιμή των περιβαλλοντικών δεδομένων από την WeatherService. Στην Εικόνα 12 που ακολουθεί παρουσιάζεται η λίστα των περιβαλλοντικών δεδομένων που λαμβάνονται για κάθε περιοχή. Σε περίπτωση που το ID του resource δεν ταιριάζει με κάποιο από τα 5 ορισμένα resource IDs, η μέθοδος επιστρέφει μια κενή συμβολοσειρά (string).

```
ArrayList<ObservationValue> obsList = new ArrayList<>();
obsList.add(measurement.getTemperatureObs());
obsList.add(measurement.getHumidityObs());
obsList.add(measurement.getIlluminanceObs());
obsList.add(measurement.getwindSpeedObs());
obsList.add(measurement.getCoObs());
obsList.add(measurement.getCo2Obs());
obsList.add(measurement.getDryObs());
obsList.add(measurement.getRainPercentageObs());
obsList.add(measurement.getPM25Obs());
obsList.add(measurement.getAtmPressureObs());

Observation obs = new Observation(sensorId, loc, timestamp, samplet , obsList);
String response = objectMapper.writeValueAsString(obs);
return response;
```

Εικόνα 12: Λαμβανόμενα δεδομένα από το CERTH API.

Για τη λήψη των δεδομένων βάσει του αναγνωριστικού του κόμβου που λαμβάνει τις περιβαλλοντικές μετρήσεις από το CERTH API ορίστηκε η μέθοδος «getLastMeasurementByNodeId», η οποία παρουσιάζεται στην Εικόνα 13. Η συγκεκριμένη μέθοδος δημιουργεί ένα σημείο εισόδου (webClient) για την υποβολή αιτημάτων HTTP GET στο CERTH API, τα οποία περιλαμβάνουν το αναγνωριστικό του κόμβου και τις απαραίτητες παραμέτρους του API ενσωματωμένες στο URL του αιτήματος. Τέλος, γίνεται εξαγωγή και καταγραφή των δεδομένων, καθώς και επιστροφή του αντικειμένου (object) της μέτρησης που περιέχει τα λαμβανόμενα δεδομένα.

```
public WeatherMeasurementDto getLastMeasurementByNodeId(int nodeId)
{
    log.debug("Fetching measurements from CERTH API: " + CERTHAPI_GETMEASUREMENTS + "/" + LOCATIONS[nodeId]);

    WeatherMeasurementDto measurement = new WeatherMeasurementDto();

    try {

        WebClient webClient = WebClient.create(CERTHAPI_URL);
        measurement = webClient.get()
                        .uri(CERTHAPI_GETMEASUREMENTS + "/" + LOCATIONS[nodeId])
                        .retrieve()
                        .bodyToMono(WeatherMeasurementDto.class)
                        .block();

        log.debug("Last measurement: " + measurement.toString());

    } catch (Exception e) {
        log.error("Error fetching data: " + e.getMessage());
        e.printStackTrace();
    }

    return measurement;
}
```

Εικόνα 13: Λήψη περιβαλλοντικών δεδομένων από το CERTH API.

Μετά την υλοποίηση του RAP plugin, επόμενο βήμα αποτελεί η δημιουργία του αντίστοιχου docker image για την εγκατάστασή του ως μικρο-υπηρεσία στο ήδη υπάρχον stack του symbIoTe cloud, βάσει της διαδικασίας που αναφέρεται στην προηγούμενη ενότητα 3.1.1. Οι καταχωρημένες πηγές περιβαλλοντικών δεδομένων είναι διαθέσιμες μέσω του: https://balkanserver.iti.gr/rh/resources.

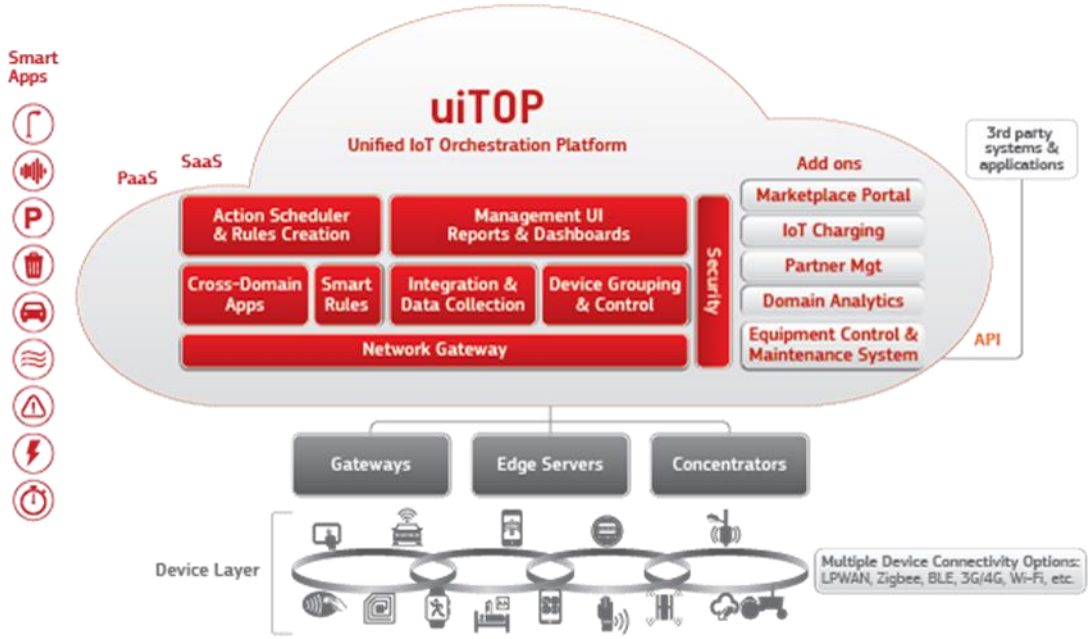
Τέλος, όλα τα εγγεγραμμένα resources στο IoTFeds σύστημα είναι προσβάσιμα μέσω της μικρο-υπηρεσίας marketplace βάσει του resource ID. Συνεπώς, η πρόσβαση π.χ. στο resource

της Αθήνας επιτυγχάνεται μέσω του: https://symbiote-core.iotfeds.intracom-telecom.com/symbioteapi/resources/access/l1/ath_weather.

3.1.3 Πλατφόρμα UiTOP και διάθεση δεδομένων έξυπνης στάθμευσης

3.1.3.1 Περιγραφή πλατφόρμας UiTOP

Όπως παρουσιάστηκε στο Π1.3 «Τελική Αρχιτεκτονική Συστήματος», η πλατφόρμα UiTOP έχει αναπτυχθεί για τη διαχείριση και τον έλεγχο ευφυών συστημάτων της Έξυπνης Πόλης (οδοφωτισμός, στάθμευση, διαχείριση απορριμμάτων, έλεγχος επιπέδων θορύβου κ.ά.). Η πλατφόρμα επιτρέπει την παρακολούθηση και τον έλεγχο αισθητήρων και συσκευών από διαφορετικούς κατασκευαστές χρησιμοποιώντας διαφορετικά πρωτόκολλα και δίκτυα διασύνδεσης και δίνοντας τη δυνατότητα ανάπτυξης και φιλοξενίας έξυπνων IoT εφαρμογών. Στην εικόνα που ακολουθεί (Εικόνα 14) παρουσιάζεται η αρχιτεκτονική της UiTOP[5] πλατφόρμας.



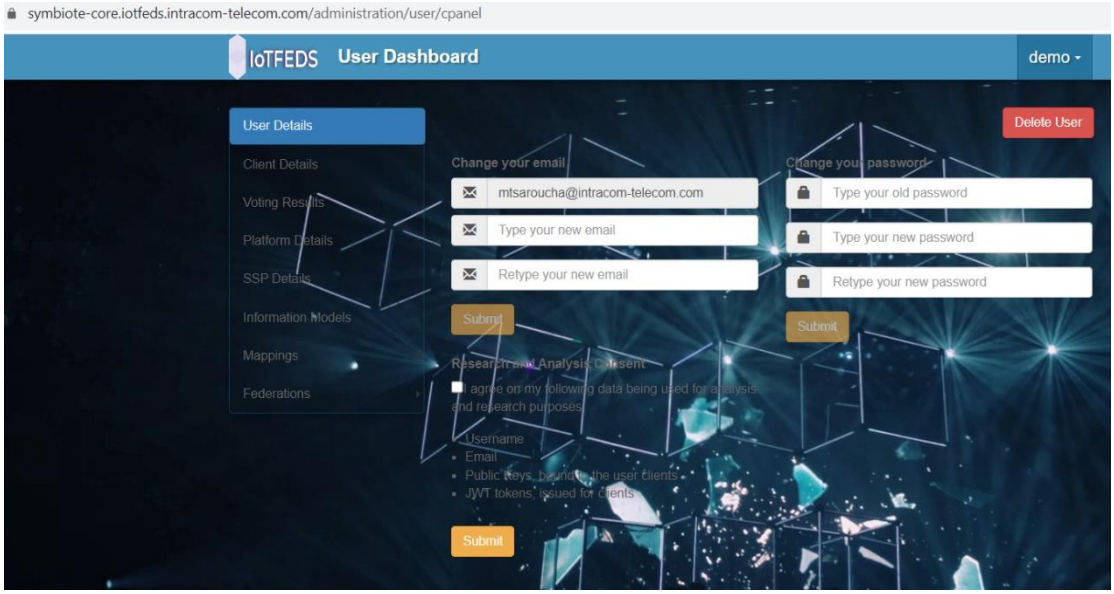
Εικόνα 14: Πλατφόρμα UiTOP.

3.1.3.2 Διασύνδεση πλατφόρμας UiTOP στο IoTFeds σύστημα

Στην ενότητα αυτή περιγράφονται σε πιο αναλυτικά βήματα η διαμόρφωση της IoT πλατφόρμας UiTOP (Intracom Telecom Unified IoT Orchestration platform) για τη διάθεση δεδομένων έξυπνης στάθμευσης στο IoTFeds σύστημα. Η πλατφόρμα UiTOP λειτουργεί ανεξάρτητα και προσφέρει πρόσβαση στα δεδομένα της μέσω υπηρεσιών API (Application Programming Interface).

Το πρώτο βήμα είναι η καταχώρηση του χρήστη και της IoT πλατφόρμας στο σύστημα. Η εγγραφή του γίνεται μέσω της γραφικής διεπαφής administration GUI στον ιστότοπο https://symbiote-core.iotfeds.intracom-telecom.com/administration. Όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 15), δημιουργείται και αποθηκεύεται ο χρήστης demo με τον ρόλο του ιδιοκτήτη υπηρεσιών (service owner).

[5] https://www.intracom-telecom.com/en/products/telco_software/IoT/IoT_orchestration_Platform.htm

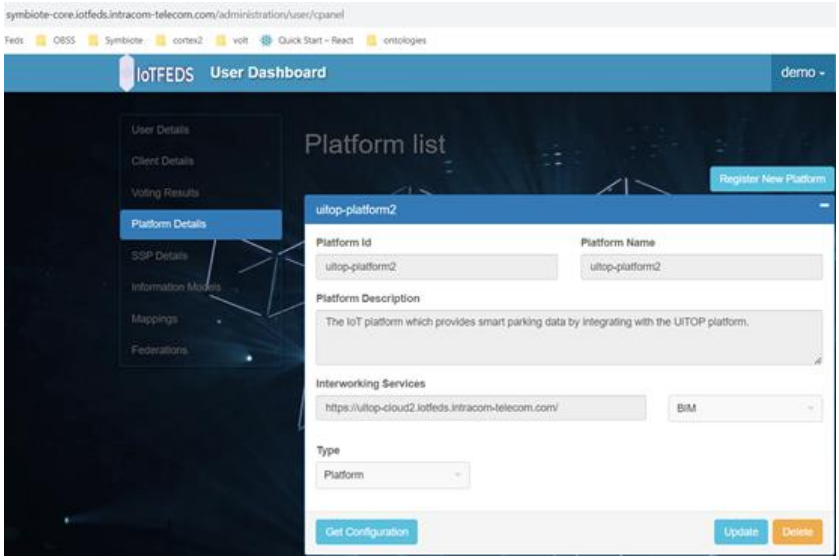


Εικόνα 15: Εγραφή χρήστη UiTOP στο IoTFeds σύστημα.

Το όνομα της πλατφόρμας IoT είναι uitop-platform2. Ο χρήστης demo είναι πιστοποιημένος χρήστης της uitop-platform2 και έχει όλα τα δικαιώματα ενός ιδιοκτήτη υπηρεσιών, δηλαδή μπορεί να δημιουργήσει, να εγγράψει και να προσπελάσει τους πόρους της uitop-platform2.

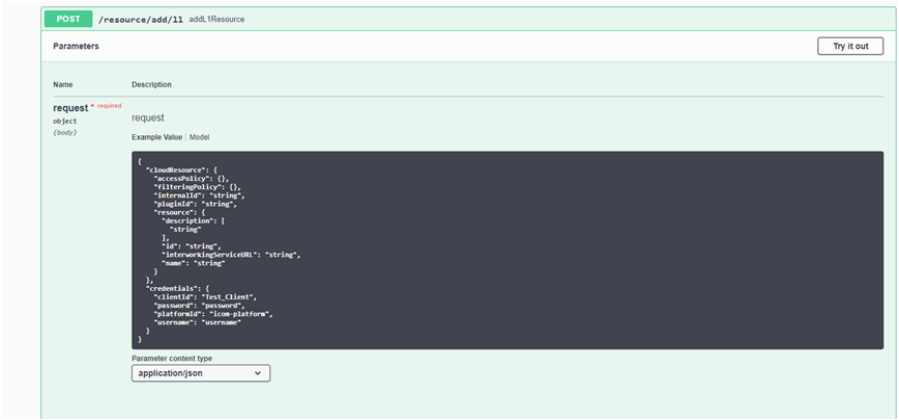
Ακολουθεί η ανάπτυξη της μικροϋπηρεσίας Rap plugin. Το Rap plugin είναι ένα λογισμικό που υλοποιεί μια διεπαφή η οποία χειρίζεται μηνύματα. Αυτά τα μηνύματα αφορούν στην ανάγνωση και την επεξεργασία δεδομένων. Υλοποιώντας το δικό της Rap plugin η uitop-platform2, εγγράφει την δική της διεπαφή μέσω της οποίας προμηθεύεται τα δεδομένα έξυπνης στάθμευσης από την πλατφόρμα UiTOP και τα εσωκλείει μέσα στους (cloud) πόρους της uitop-platform2. Στο τέλος αυτής της ενότητας εξηγούνται αναλυτικά τα δεδομένα έξυπνης στάθμευσης.

Τέλος, γίνεται η εγκατάσταση της πλατφόρμας uitop-platform2 καθώς και η λειτουργία της σαν αυτοματοποιημένη εφαρμογή. Όπως έχει ήδη αναφερθεί και έχει περιγραφεί αναλυτικά στο παραδοτέο Π1.4, στις IoT πλατφόρμες (cloud platforms) εγκαθίσταται κάποιες μικροϋπηρεσίες (symbIoTe cloud). Οι μικροϋπηρεσίες της uitop-platform2 τρέχουν σε περιβάλλον Kubernetes. Η λειτουργία της uitop-platform2 βασίζεται σε μια εξωτερικοποιημένη διαμόρφωση παραμέτρων (configuration) και το σύνολο αυτής της ρύθμισης μπορεί να αποκτηθεί στην ακόλουθη τοποθεσία του GUI, όπως φαίνεται στην *Εικόνα 5*. Το περιεχόμενο του configuration μπαίνει σε εξωτερικά αρχεία «application.properties» στα οποία βασίζονται οι μικρουπηρεσίες (containers) της (cloud) εγκατάστασης για την uitop-platform2 που τρέχουν στο Kubernetes περιβάλλον.



Εικόνα 16: Καταχώρηση πλατφόρμας uitop-platform2 στο IoTFeds σύστημα.

Για την προσπέλαση των δεδομένων στάθμευσης που διαθέτουν οι πηγές δεδομένων της IoT πλατφόρμας (uitop-platform2) απαιτείται η καταχώρηση των IoT πόρων (πηγών δεδομένων) στην uitop-platform2 στο IoTFeds σύστημα μέσα του marketplace API: https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/resource/add/l1:



Εικόνα 17: Το API endpoint της Marketplace υπηρεσίας που χρησιμοποιείται για την εγγραφή πόρων στην πλατφόρμα IoT (uitop-platform2).

Στην Εικόνα 6 φαίνονται τα μεταδεδομένα από την επιτυχή εγγραφή του IoT πόρου “uitopSensors”, ο οποίος περιέχει δεδομένα στάθμευσης σε πραγματική τοποθεσία εντός των εγκαταστάσεων της Intracom Telecom. Επιπλέον στην *Εικόνα 7* φαίνεται η επιτυχής εγγραφή του IoT πόρου “uitopVirtualSensors” ενός εικονικού (μη πραγματικού) χώρου στάθμευσης ο οποίος περιέχει δεδομένα στάθμευσης σε μια εικονική τοποθεσία.


```
{
 "cloudResource": {
  "internalId": "uitopSensors",
  "pluginId": "uitop-platform2",
  "accessPolicy": {
   "policyType": "PUBLIC",
   "requiredClaims": {}
  },
  "filteringPolicy": {
   "policyType": "PUBLIC",
   "requiredClaims": {}
```

```
    },
    "resource": {
      "@c": ".StationarySensor",
      "observesProperty": [
        "state"
      ],
      "id": null,
      "name": "Parking area",
      "description": [
        "Parking sensors"
      ],
      "interworkingServiceURL": "https://uitop-cloud2.iotfeds.intracom-telecom.com/",
      "locatedAt": {
        "@c": ".WGS84Location",
        "longitude": 23.874748,
        "latitude": 37.940128,
        "name": "Europe/Athens",
        "description": [
          "non-restricted parking area"
        ]
      },
      "featureOfInterest": {
        "name": "parking spot",
        "description": [
          "parking availability, 4 parking spots"
        ],
        "hasProperty": [
            "state"
        ]
      }
    }
  },
  "credentials": {
    "clientId": "****",
    "password": "****",
    "platformId": "uitop-platform2",
    "username": "demo"
  }
}
```


Εικόνα 18: Καταχώρηση “uitopSensors” πηγής δεδομένων πλατφόρμας uitop-platform2.


```
{
  "cloudResource": {
    "internalId": "uitopVirtualSensors",
    "pluginId": "uitop-platform2",
    "accessPolicy": {
      "policyType": "PUBLIC",
      "requiredClaims": {}
    },
    "filteringPolicy": {
      "policyType": "PUBLIC",
      "requiredClaims": {}
    },
    "resource": {
      "@c": ".StationarySensor",
      "observesProperty": [
        "state"
      ],
      "id": null,
      "name": "Parking area",
      "description": [
        "Parking sensors"
      ],
      "interworkingServiceURL": "https://uitop-cloud2.iotfeds.intracom-telecom.com/",
      "locatedAt": {
        "@c": ".WGS84Location",
        "longitude": 37.909735,
```

```
        "latitude": 23.877304,
        "name": "Europe/Athens",
        "description": [
          "non-restricted parking area"
        ]
      },
      "featureOfInterest": {
        "name": "parking spot",
        "description": [
          "parking availability, 4 parking spots"
        ],
        "hasProperty": [
            "state"
        ]
      }
    }
  },
  "credentials": {
    "clientId": "****",
    "password": "****",
    "platformId": "uitop-platform2",
    "username": "demo"
  }
}
```


Εικόνα 19: Καταχώρηση “uitopVirtualSensors” πηγής δεδομένων πλατφόρμας uitop-platform2).

Μετά την επιτυχή τους εγγραφή στην IoT πλατφόρμα uitop-platform2, οι δύο πόροι έχουν λάβει από έναν αναγνωριστικό αριθμό ο καθένας. Αντίστοιχα ο πόρος “uitopSensors” πήρε το αναγνωριστικό “6566e7fd7ba94a00010c462f” και ο πόρος “uitopVirtualSensors” πήρε το αναγνωριστικό “6566f13d7ba94a00010c4638”.

Στην συνέχεια γίνεται η συλλογή των δεδομένων τους. Τον ρόλο του συλλέκτη δεδομένων αναλαμβάνει το λειτουργικό Rap-Plugin της πλατφόρμας IoT, το οποίο εκτελώντας REST (Representational state transfer) κλήσεις στην πλατφόρμα UiTOP, συγκεντρώνει τα δεδομένα στάθμευσης και γεμίζει τους πόρους uitopSensors και uitopVirtualSensors.

Στην Εικόνα 8 φαίνεται το περιεχόμενο μίας παρατήρησης (observation) του πόρου uitopSensors μέσω του marketplace API: *https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/resource/access/l1/uitopSensors* που επαληθεύει την επιτυχή πρόσβαση στα δεδομένα κίνησης της πλατφόρμας uitop-platform2. Στη φυσική περιοχή στάθμευσης (τοποθεσία εντός της εταιρείας Intracom Telecom) έχουν εγκατασταθεί τέσσερις αισθητήρες, ένας αισθητήρας σε κάθε θέση στάθμευσης. Τα ονόματα των αισθητήρων αυτών παρουσιάζονται σαν «Parking Sensor 1», «Parking Sensor 2», «Parking Sensor 3» και «Parking Sensor 4». Ανάλογα με τη διαθεσιμότητα της κάθε θέσης έξυονης στάθμευσης, η κατάσταση “state” μπορεί να είναι "sensor error", "free" ή "occupied" που αντιστοιχεί στις τιμές [-1, 0, 1] για το πεδίο “value”.


```
[
  {
    "resourceId": "655cbb8a7ba94a00010c4606",
    "location": {
      "@c": ".WGS84Location",
      "longitude": 23.874748,
      "latitude": 37.940128,
      "altitude": -1.0,
      "name": "",
      "description": null
    },
    "resultTime": "2017-08-10 08:52:08.0",
    "samplingTime": "2023-11-28T07:53:22",
    "obsValues": [
      {
        "value": "1",
        "obsProperty": {
          "name": "state",
          "iri": "IRI",
          "description": [
```

```
                "Parking Sensor 1",
                "Occupied parking."
            ]
        },
        "uom": null,
        "featureOfInterest": null
    },
    {
        "value": "1",
        "obsProperty": {
            "name": "state",
            "iri": "IRI",
            "description": [
                "Parking Sensor 2",
                "Occupied parking."
            ]
        },
        "uom": null,
        "featureOfInterest": null
    },
    {
        "value": "1",
        "obsProperty": {
            "name": "state",
            "iri": "IRI",
            "description": [
                "Parking Sensor 3",
                "Occupied parking."
            ]
        },
        "uom": null,
        "featureOfInterest": null
    },
    {
        "value": "1",
        "obsProperty": {
            "name": "state",
            "iri": "IRI",
            "description": [
                "Parking Sensor 4",
                "Occupied parking."
            ]
        },
        "uom": null,
        "featureOfInterest": null
    }
  ]
 }
]
```


Εικόνα 20: Δεδομένα έξυπνης στάθμευσης “uitopSensors” της πλατφόρμας uitop-platform2.

Τα δεδομένα από την εικονική περιοχή στάθμευσης μπορούν να προσπελαστούν με παρόμοιο τρόπο. Η εικονική περιοχή στάθμευσης αφορά δεκαέξι εικονικές θέσεις στάθμευσης σε κοντινό γεωγραφικό σημείο. Ο σκοπός που προστίθενται εικονικά δεδομένα στάθμευσης είναι για να βοηθήσουν στην εκτέλεση του πιλοτικού σεναρίου όπου δίνονται διάφορες επιλογές για θέσεις στάθμευσης στον χρήστη της κινητής συσκευής.

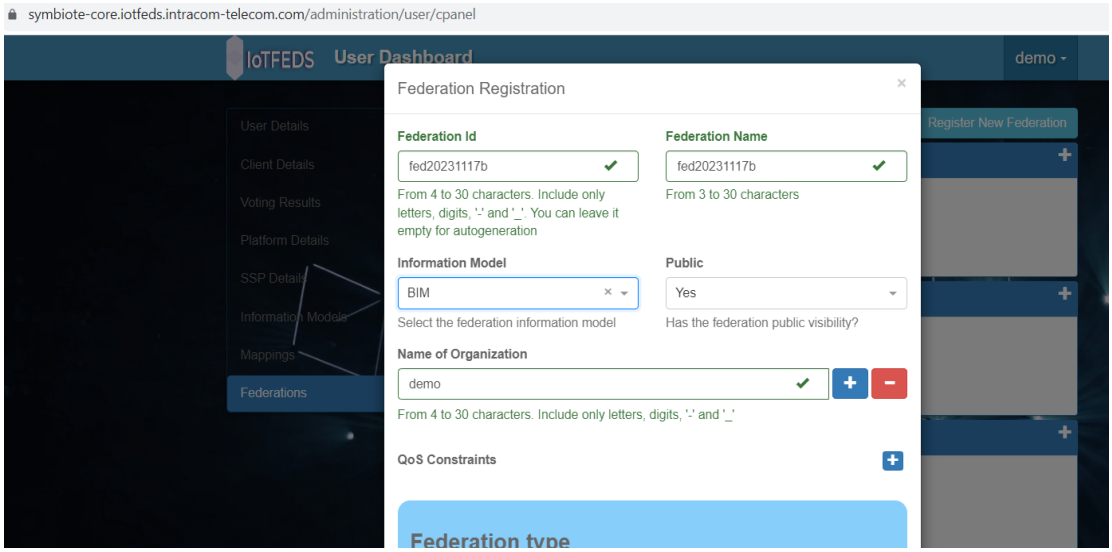
Η προσπέλαση των δεδομένων στάθμευσης και για τους δύο πόρους γίνεται από την μέθοδο «readResource» της διεπαφής Rap-plugin. Ουσιαστικά, η μικροϋπηρεσία Rap μετά από μια κλήση για προσπέλαση ενός πόρου καλεί την «readResource» για να διαβάσει το περιεχόμενο του. Είσοδος της διεπαφής είναι το αναγνωριστικό του πόρου και έξοδος της είναι το περιεχόμενο του πόρου σε μορφή JSON. Στην Εικόνα 11 φαίνεται ο κώδικας που υλοποιεί την μέθοδο «readResource».

```java
@Override
public String readResource(String resourceId) {
    try {
        if("uitopSensors".equals(resourceId)) {
            ParkingAreaDto parkingArea = parkingService.getParkingArea();
            WGS84Location loc = parkingArea.getLocation();
            String sampleTime = parkingArea.getObservationDate();
            String timestamp = parkingArea.getSensorTime();
            List<ObservationValue> obsList = parkingArea.getObsValues();
            Observation obs = new Observation( resourceId: "uitopSensors", loc, timestamp, sampleTime, obsList);
            return objectMapper.writeValueAsString(obs);
        } else if("uitopVirtualSensors".equals(resourceId)) {
            ParkingAreaDto parkingArea = parkingService.getVirtualParkingArea();
            WGS84Location loc = parkingArea.getLocation();
            String sampleTime = parkingArea.getObservationDate();
            String timestamp = parkingArea.getSensorTime();
            List<ObservationValue> obsList = parkingArea.getObsValues();
            Observation obs = new Observation( resourceId: "uitopVirtualSensors", loc, timestamp, sampleTime, obsList);
            return objectMapper.writeValueAsString(obs);
        } else {
            return "";
        }
    } catch (JsonProcessingException exc) {
        String message = String.format("Can not process observation properties of resourceId %s. ", resourceId);
        throw new RapPluginException(HttpStatus.INTERNAL_SERVER_ERROR.value(), message , exc);
    } catch (Exception exc) {
        throw new RapPluginException(HttpStatus.NOT_FOUND.value(), "Parking sensor(s) not found.");
    }
}
```

Εικόνα 21: Η υλοποίηση της μεθόδου στο Rap-plugin που δημιουργεί τις παρατηρήσεις (observations) για τα δεδομένα έξυπνης στάθμευσης.

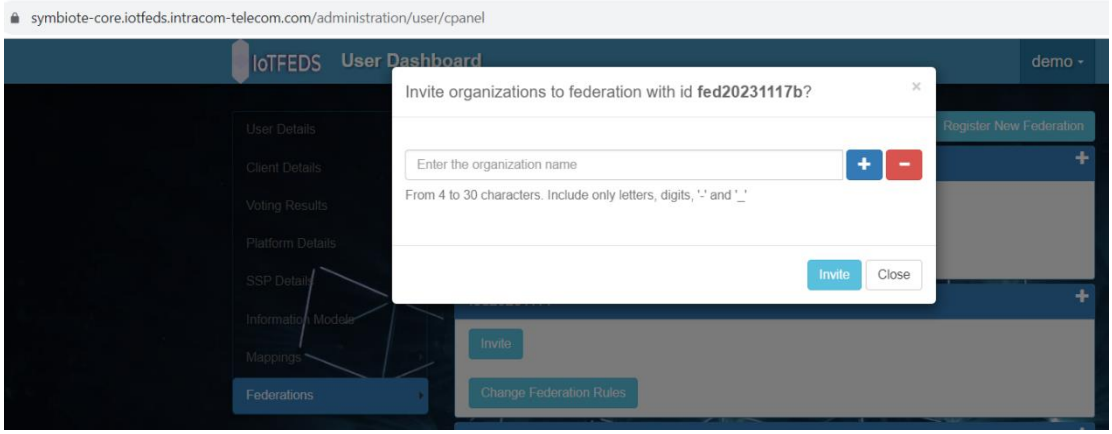
3.2 Δημιουργία ομοσπονδίας και εγγραφή IoT πόρων σε αυτήν

Στο στάδιο αυτό δημιουργήθηκε ομοσπονδία μεταξύ των τριών πιλοτικών IoT πλατφορμών, ορίζοντας τους κανόνες ομοσπονδίας και το μοντέλο πληροφορίας που αξιοποιούν τα μέλη στα πλαίσια της ομοσπονδίας για άμεση διαλειτουργικότητα. Για το σκοπό αυτό ο οργανισμός demo (ICOM/UiTOP) δημιούργησε μία ομοσπονδία μέσω του administration GUI, όπως φαίνεται στην παρακάτω εικόνα:



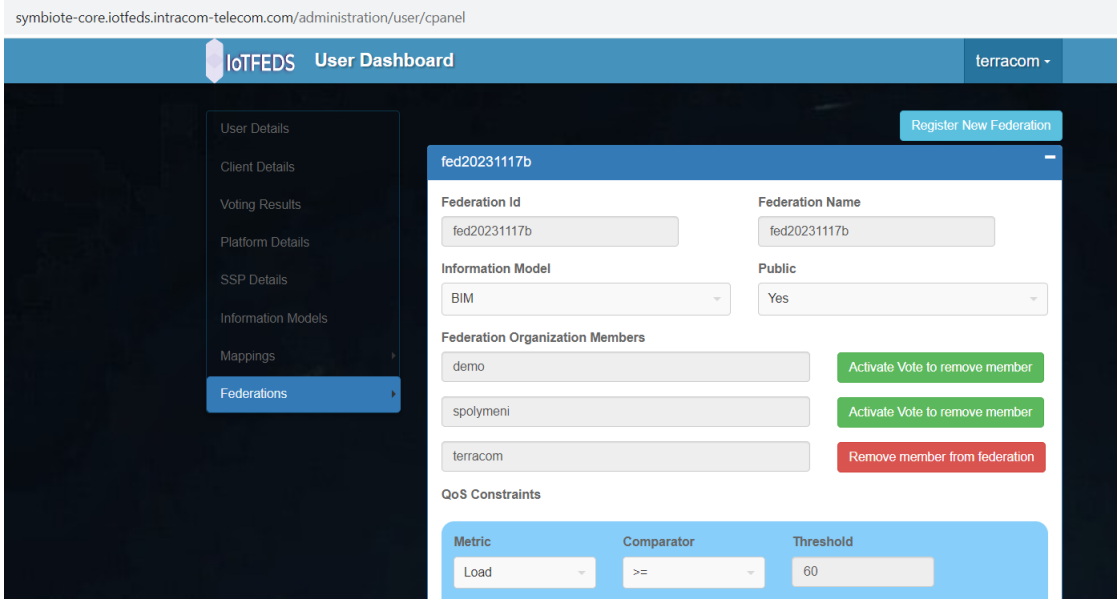
Εικόνα 22: Δημιουργία IoTFeds ομοσπονδίας.

Στη συνέχεια, ο χρήστης ζήτησε τη συμμετοχή των δύο άλλων οργανισμών/πλατφορμών, όπως φαίνεται στην παρακάτω εικόνα:



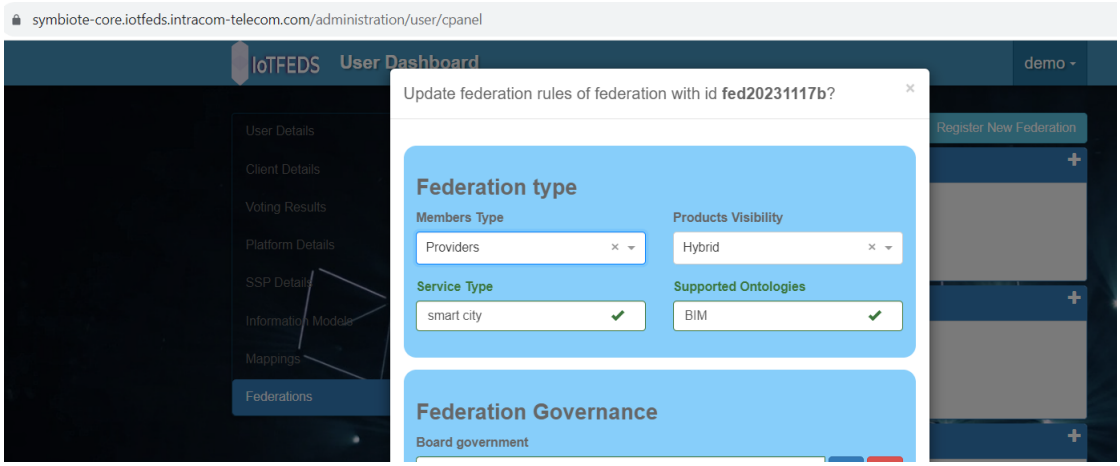
Εικόνα 23: Προσθήκη οργανισμών στην ομοσπονδία.

Η ομοσπονδία που δημιουργήθηκε με τους τρεις οργανισμούς (μέλη) φαίνεται στην παρακάτω εικόνα (Εικόνα 24):



Εικόνα 24: Δημιουργία ομοσπονδίας μεταξύ των πιλοτικών IoT πλατφορμών.

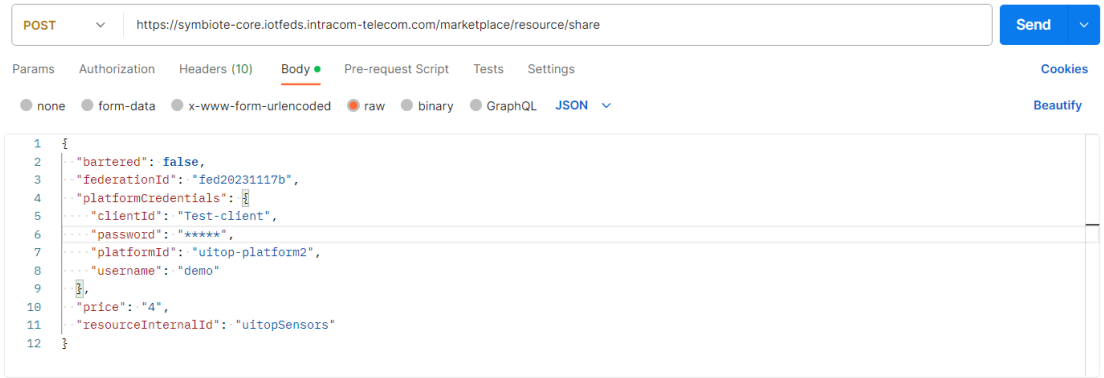
Κατά την εγγραφή της ομοσπονδίας ορίστηκαν και οι κανόνες της ομοσπονδίας οι οποίοι μπορούν στη συνέχεια να τροποποιηθούν κατόπιν σχετικού αιτήματος και με χρήση ψηφοφορίας. Μεταξύ αυτών, καθορίζεται και η προώθηση ή μη των προϊόντων που συντίθενται σε αυτήν. Ο κανόνας αυτός ("products visibility") παίρνει τις τιμές «public" ή "closed" για την διάθεση ή μη των προϊόντων της ομοσπονδιακής αγοράς στην καθολική αγορά και μπορεί να τροποποιηθεί με αίτημα αλλαγής κανόνων ομοσπονδίας. Ένα τέτοιο αίτημα αλλαγής κανόνων φαίνεται στην παρακάτω εικόνα (Εικόνα 25):



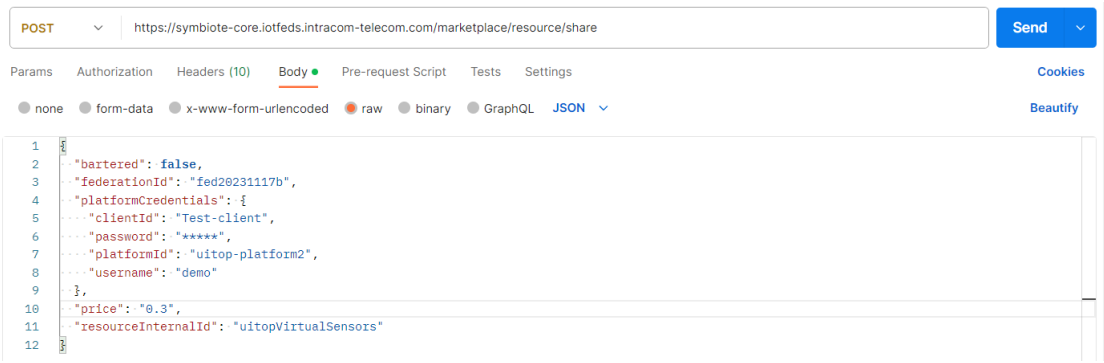
Εικόνα 25: Ορισμός κανόνων ομοσπονδίας.

Ο κανόνας αυτός ελέγχεται κατά τη δημιουργία προϊόντος, που περιγράφεται στην επόμενη ενότητα, για να προωθηθούν ή όχι στην καθολική αγορά τα προϊόντα που δημιουργούνται. Ωστόσο, στο επόμενο βήμα που ακολουθεί (πριν την σύνθεση προϊόντος), απαιτείται ο διαμοιρασμός των πηγών δεδομένων (IoT πόρων) από κάθε συμμετέχουσα IoT πλατφόρμα και στις οποίες βασίζεται η σύνθεση προϊόντων δεδομένων για διάθεση στην αγορά (ομοσπονδιακή ή και καθολική). Ο διαμοιρασμός αφορά πηγές δεδομένων που έχουν ήδη καταχωρηθεί στην IoTFeds πλατφόρμα και επιτυγχάνεται αξιοποιώντας το REST API endpoint του Marketplace: https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/resource/share. Στο σώμα της κλήσης, η συμμετέχουσα πλατφόρμα IoT βάζει τα πιστοποιημένα στοιχεία της καθώς και το όνομα της ομοσπονδίας και το αναγνωριστικό του πόρου που θέλει να διαμοιραστεί σε αυτήν. Μετά την επιτυχή καταχώρηση ή και διαμοιρασμό μίας πηγής δεδομένων, οι σχετικές πληροφορίες εμφανίζονται στο παρακάτω url της πλατφόρμας: «{platform_url}/rh/resources». Οι πληροφορίες για κάθε καταχωρημένη πηγή δεδομένων (IoT πόρο) περιλαμβάνει πληροφορίες σχετικά με τις ομοσπονδίες στις οποίες έχει διαμοιραστεί με επιτυχία.

Παρακάτω φαίνεται ο τρόπος που η IoT πλατφόρμα “uitop-platform2” διαμοίρασε τους πόρους “uitopSensors” και “uitopVirtualSensors” αντίστοιχα στην ομοσπονδία “fed20231117b”.

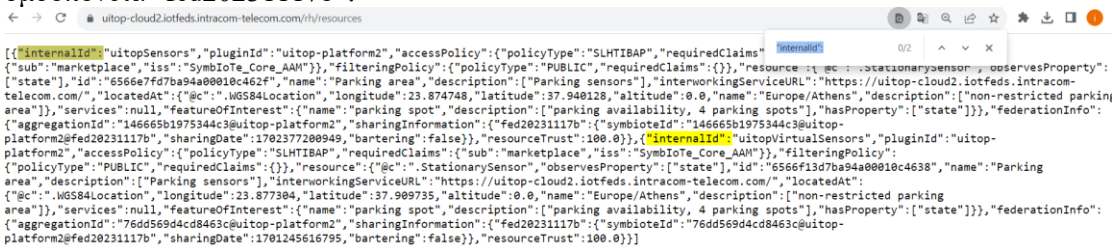


Εικόνα 26: Διαμοιρασμός IoT πόρου uitopSensors (uitop-platform2) στην ομοσπονδία “fed20231117b”.



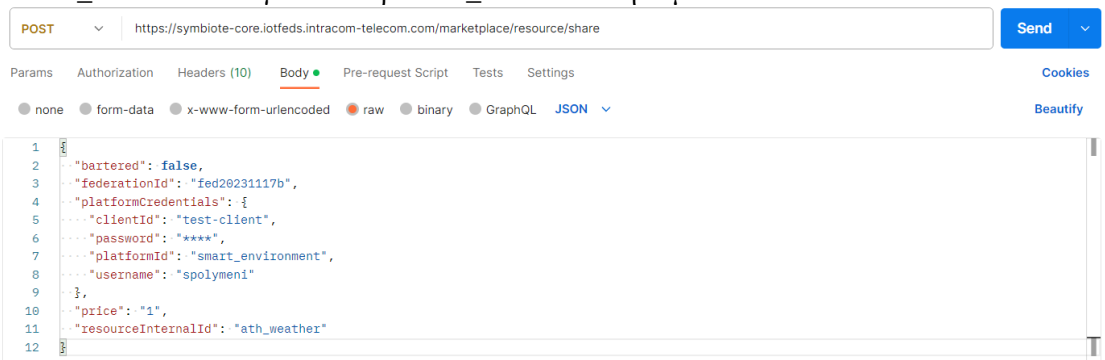
Εικόνα 27: Διαμοιρασμός IoT πόρου uitopVirtualSensors (uitop-platform2) στην ομοσπονδία “fed20231117b”.

Στην παρακάτω εικόνα (Εικόνα 28) φαίνονται οι καταχωρημένοι IoT πόροι της “uitop-platform2” που περιλαμβάνουν τους IoT πόρους που έχουν επίσης διαμοιραστεί στην ομοσπονδία “fed20231117b”.



Εικόνα 28: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “uitop-platform2”.

Στην παρακάτω εικόνα φαίνεται το αίτημα διαμοιρασμού της IoT πλατφόρμας “smart_environment” για τον πόρο “ath_weather” στην ομοσπονδία “fed20231117b”.



Εικόνα 29: Διαμοιρασμός IoT πόρου ath_weather (smart_environment) στην ομοσπονδία “fed20231117b”.

Παρακάτω φαίνονται οι πόροι που έχουν αποθηκευτεί στην “smart_environment” και φαίνεται επίσης ότι ο πόρος “ath_weather” είναι μέρος της ομοσπονδίας “fed20231117b”.

```
balkanserver.iti.gr/rh/resources
[{"internalId":"skg_weather","pluginId":"smart_environment","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"market
{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".StationarySensor","observesProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"],"id":"655b32247ba94a00010c45e4","name":"Weather
station/Thessaloniki","description":["Weather station at Thessaloniki"],"interworkingServiceURL":"https://balkanserver.iti.gr","locatedAt":
{"@c":".WGS84Location","longitude":22.994627,"latitude":40.563547,"altitude":0.0,"name":"Europe/Thessaloniki","description":["Weather station at Nea Krini,
Thessaloniki"]},"services":null,"featureOfInterest":{"name":"Thessaloniki","description":["Weather data of Thessaloniki"],"hasProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"]}},"federationInfo":
{"aggregationId":"50008d4f121e4855@smart_environment","sharingInformation":{"fed20231117b":
{"symbioteId":"50008d4f121e4855@smart_environment@fed20231117b","sharingDate":1701245713348,"bartering":false}},"resourceTrust":null}},
{"internalId":"ath_weather","pluginId":"smart_environment","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".StationarySensor","observesProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"],"id":"655b37747ba94a00010c45ea","name":"Weather
station/Athens","description":["Weather station at Athens"],"interworkingServiceURL":"https://balkanserver.iti.gr","locatedAt":
{"@c":".WGS84Location","longitude":23.8700195,"latitude":37.9423059,"altitude":0.0,"name":"Europe/Athens","description":["Weather station at Paiania,
Athens"]},"services":null,"featureOfInterest":{"name":"Athens","description":["Weather data of Athens"],"hasProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"]}},"federationInfo":
{"aggregationId":"55e610d7dbdd4cd3@smart_environment","sharingInformation":{"fed20231117b":
{"symbioteId":"55e610d7dbdd4cd3@smart_environment@fed20231117b","sharingDate":1701246043186,"bartering":false}},"resourceTrust":null}},
{"internalId":"ioa_weather","pluginId":"smart_environment","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".StationarySensor","observesProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"],"id":"656ef80d7ba94a00010c463d","name":"Weather
station/Ioannina","description":["Weather station at Ioannina"],"interworkingServiceURL":"https://balkanserver.iti.gr","locatedAt":
{"@c":".WGS84Location","longitude":20.853746,"latitude":39.665028,"altitude":0.0,"name":"Europe/Ioannina","description":["Weather station at City Center,
Ioannina"]},"services":null,"featureOfInterest":{"name":"Ioannina","description":["Weather data of Ioannina"],"hasProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"]}},"federationInfo":null},
{"internalId":"gpa_weather","pluginId":"smart_environment","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".StationarySensor","observesProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"],"id":"656ef8337ba94a00010c4640","name":"Weather
station/Patrai","description":["Weather station at Patrai"],"interworkingServiceURL":"https://balkanserver.iti.gr","locatedAt":
{"@c":".WGS84Location","longitude":21.732801,"latitude":38.242981,"altitude":0.0,"name":"Europe/Patrai","description":["Weather station at City Center,
Patrai"]},"services":null,"featureOfInterest":{"name":"Patrai","description":["Weather data of Patrai"],"hasProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"]}},"federationInfo":null},
{"internalId":"vol_weather","pluginId":"smart_environment","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".StationarySensor","observesProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"],"id":"656ef85d7ba94a00010c4643","name":"Weather
station/Volos","description":["Weather station at Volos"],"interworkingServiceURL":"https://balkanserver.iti.gr","locatedAt":
{"@c":".WGS84Location","longitude":22.9267206,"latitude":39.3652368,"altitude":0.0,"name":"Europe/Volos","description":["Weather station at City Center,
Volos"]},"services":null,"featureOfInterest":{"name":"Volos","description":["Weather data of Volos"],"hasProperty":
["temperature","humidity","illuminance","windSpeed","co","co2","dry","rainPercentage","pm2_5","atmosphericPressure"]}},"federationInfo":null}]
```


Εικόνα 30: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “smart_ environment”.

Επιπλέον, στις επόμενες εικόνες φαίνονται τα αιτήματα διαμοιρασμού της “zastel-spotypal” για τους πόρους “traffic-data-athens-historical” και “ traffic-data-athens-realtime” στην ομοσπονδία “fed20231117b”.


```
POST https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/resource/share
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
{
  "bartered": false,
  "federationId": "fed20231117b",
  "platformCredentials": {
    "clientId": "test-client",
    "password": "****",
    "platformId": "zastel-spotypal",
    "username": "terracom"
  },
  "price": "1.5",
  "resourceInternalId": "traffic-data-athens-realtime"
}
```


Εικόνα 31: Διαμοιρασμός IoT πόρου traffic-data-athens-realtime (zastel-spotypal) στην ομοσπονδία “fed20231117b”.


```
POST https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/resource/share
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
{
  "bartered": false,
  "federationId": "fed20231117b",
  "platformCredentials": {
    "clientId": "test-client",
    "password": "****",
    "platformId": "zastel-spotypal",
    "username": "terracom"
  },
  "price": "0.8",
  "resourceInternalId": "traffic-data-athens-historical"
}
```


Εικόνα 32: Διαμοιρασμός IoT πόρου traffic-data-athens-historical (zastel-spotypal) στην ομοσπονδία “fed20231117b”.

Τέλος, στην Εικόνα 33 φαίνονται οι πόροι που έχουν καταχωρηθεί από την “zastel-spotypal” καθώς και οι πληροφορίες σχετικά με τον διαμοιρασμό τους σε ομοσπονδίες.

iotfeds-service.zastel.com/rh/resources

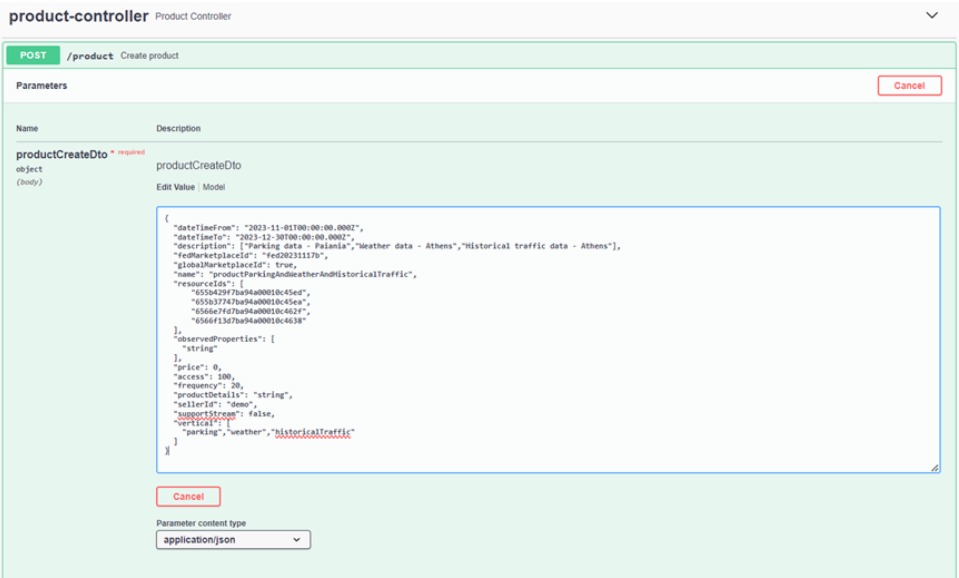
internalId 0/3

:{"internalId":"traffic-data-athens-historical","pluginId":"zastel-spotypal","accessPolicy":{"policyType":"SLHTIBAP
:"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":{"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".MobileSensor","observesProperty":
:],"id":"655b429f7ba94a00010c45ed","name":"Traffic Data Athens Historical","description":["Traffic Data Athens Historical"],"interworkingServiceURL":"https://iotfeds-
:ervice.zastel.com/","locatedAt":{"@c":".WGS84Location","longitude":37.960781,"latitude":23.861427,"altitude":0.0,"name":"Europe/Athens","description":
:"Athens"]},"services":null},"federationInfo":{"aggregationId":"7c9f98be58be44a2@zastel-spotypal","sharingInformation":{"fed20231117b":
:"symbioteId":"7c9f98be58be44a2@zastel-spotypal@fed20231117b","sharingDate":1700483335822,"bartering":false}},"resourceTrust":null}},{"internalId":"traffic-data-athens-
:ealtime","pluginId":"zastel-spotypal","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
:"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".MobileSensor","observesProperty":[],"id":"655b42b87ba94a00010c45f0","name":"Traffic Data Athens
:ealtime","description":["Traffic Data Athens Realtime"],"interworkingServiceURL":"https://iotfeds-service.zastel.com/","locatedAt":
:"@c":".WGS84Location","longitude":37.960781,"latitude":23.861427,"altitude":0.0,"name":"Europe/Athens","description":["Athens"]},"services":null},"federationInfo":
:"aggregationId":"5d1789a79b224560@zastel-spotypal","sharingInformation":{"fed20231117b":{"symbioteId":"5d1789a79b224560@zastel-
:potypal@fed20231117b","sharingDate":1700483358849,"bartering":false}},"resourceTrust":null}},{"internalId":"traffic-data-ioannina","pluginId":"zastel-
:potypal","accessPolicy":{"policyType":"SLHTIBAP","requiredClaims":{"sub":"marketplace","iss":"SymbIoTe_Core_AAM"}},"filteringPolicy":
:"policyType":"PUBLIC","requiredClaims":{}},"resource":{"@c":".MobileSensor","observesProperty":[],"id":"655b42e87ba94a00010c45f3","name":"Traffic Data
:Ioannina","description":["Traffic Data Ioannina"],"interworkingServiceURL":"https://iotfeds-service.zastel.com/","locatedAt":
:"@c":".WGS84Location","longitude":39.657126,"latitude":20.857933,"altitude":0.0,"name":"Europe/Ioannina","description":["Ioannina"]},"services":null},"federationInfo":
:"aggregationId":"0c0660dd89b24351@zastel-spotypal","sharingInformation":{"fed20231117b":{"symbioteId":"0c0660dd89b24351@zastel-
:potypal@fed20231117b","sharingDate":1700483375137,"bartering":false}},"resourceTrust":null}}]

Εικόνα 33: Οι καταχωρημένοι (και διαμοιρασμένοι σε ομοσπονδία) πόροι της “zastel-spotypal”.

3.3 Δημιουργία προϊόντων και διάθεση τους στην IoTFeds αγορά

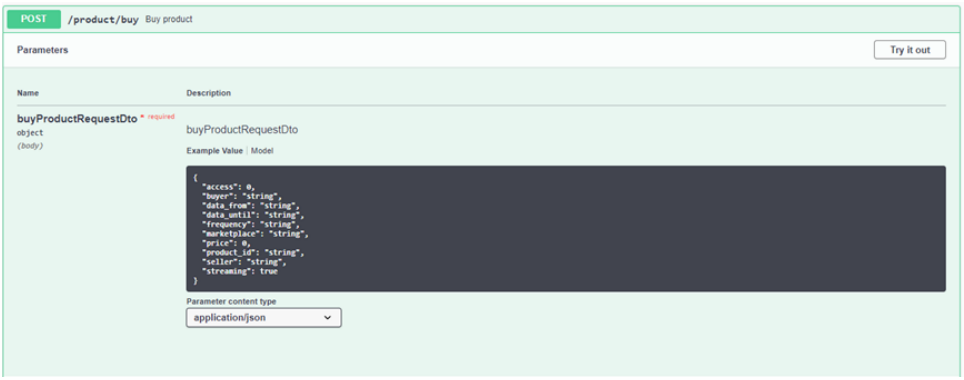
Ένα προϊόν δημιουργείται εκτελώντας μια κλήση στο REST API endpoint της Marketplace υπηρεσίας: https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/product. Ο δημιουργός του προϊόντος φαίνεται στο πεδίο «sellerId». Οι πόροι που συνθέτουν το προϊόν δηλώνονται σε μια λίστα με τους αναγνωριστικούς κωδικούς τους, «resourceIds». Το νέο προϊόν διατίθεται στην ομοσπονδιακή αγορά με αναγνωριστικό «fedMarketplaceId» για το χρονικό διάστημα που καθορίζεται από τα πεδία «dateTimeFrom» και «dateTimeTo». Η κλήση επιστρέφει τον αναγνωριστικό κωδικό του νέου προϊόντος σε περίπτωσης επιτυχούς δημιουργίας ή μήνυμα λάθους. Ο κάτοχος της πλατφόρμας uitop-platform2 δημιούργησε προϊόντα για διάθεση προς την καθολική αγορά. Στην Εικόνα 34 φαίνεται το προϊόν που επιδεικνύεται στην πιλοτική εφαρμογή. Το όνομα του προϊόντος είναι «productParkingAndWeatherAndHistoricalTraffic» και ο αναγνωριστικός κωδικός του είναι "6566f66164967575d4ddb64a". Ο δημιουργός του προϊόντος δίνει δικαίωμα στον αγοραστή του για εκατό προσπελάσεις στους πόρους του προϊόντος, μέσα στο χρονικό διάστημα, από τις "2023-11-01T00:00:00.000Z" έως και τις "2023-12-30T00:00:00.000Z". Το προϊόν αυτό αποτελείται από πόρους των τριών πιλοτικών IoT πλατφορμών (ο πόρος "655b429f7ba94a00010c45ed" περιέχει ιστορικά δεδομένα κυκλοφοριακής κίνησης για την Αθήνα, ο πόρος "655b37747ba94a00010c45ea" περιέχει περιβαλλοντικά δεδομένα για την Αθήνα και οι πόροι "6566e7fd7ba94a00010c462f" και "6566f13d7ba94a00010c4638” περιέχουν δεδομένα στάθμευσης για την Αθήνα).



Εικόνα 34: Δημιουργία προϊόντος που συνδυάζει πόρους και των τριών πλατφορμών IoT.

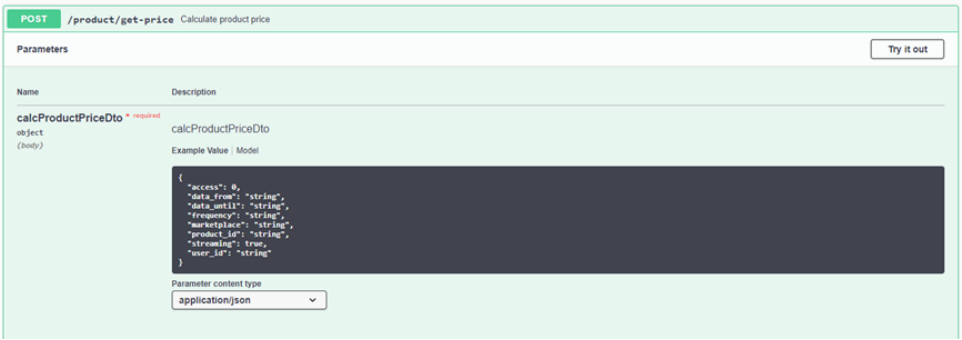
Για την επιτυχή αγορά ενός προϊόντος, είναι προϋπόθεση ο αγοραστής να γνωρίζει τον αναγνωριστικό κωδικό αυτού του προϊόντος, καθώς και να έχει επαρκές υπόλοιπο στον

λογαριασμό του για την αγορά. Στην Εικόνα 35φαίνεται το REST API endpoint «https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/product/buy» της Marketplace υπηρεσίας που πραγματοποιεί την αγορά ενός προϊόντος. Ο πωλητής του προϊόντος φαίνεται στο πεδίο «seller» και ο αγοραστής στο πεδίο «buyer». Το αποτέλεσμα της επιτυχούς αγοράς είναι πως το προϊόν ανήκει στον χρήστη που το αγόρασε, ο οποίος μπορεί να προσπελάσει τους πόρους του και τα δεδομένα αυτών.



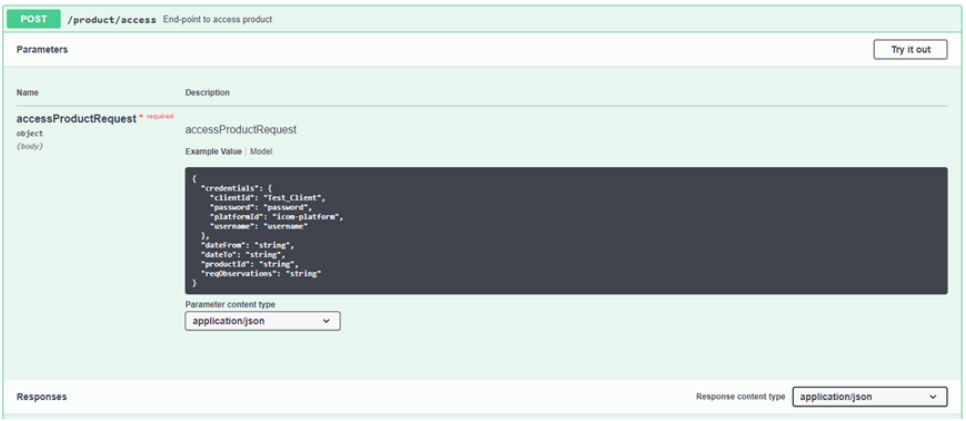
Εικόνα 35: Αγορά προϊόντος από το Marketplace API.

Ο υποψήφιος αγοραστής ενός προϊόντος μπορεί να αναζητήσει την τιμή του προϊόντος που θέλει να αγοράσει, ώστε να γνωρίζει εκ των προτέρων εάν το υπόλοιπο του λογαριασμού του επαρκεί για να πραγματοποιήσει την αγορά. Η αντίστοιχη κλήση στο API endpoint της Marketplace υπηρεσίας φαίνεται στην Εικόνα 36. Ο υποψήφιος αγοραστής εισάγει τον κωδικό του προϊόντος που επιθυμεί να αγοράσει, τον κωδικό της αγοράς που διαθέτει το συγκεκριμένο προϊόν, καθώς και των αριθμό των προσπελάσεων που τον ενδιαφέρουν μέσα στο χρονικό διάστημα από «dataFrom» έως και «dataTo». Το παρακάτω endpoint απαντάει με την τιμή του προϊόντος στην αγορά που ζητήθηκε.



Εικόνα 36: Έλεγχος για την τιμή ενός προϊόντος από το Marketplace API.

Ο ιδιοκτήτης ενός προϊόντος μπορεί να προσπελάσει ένα προϊόν που του ανήκει καλώντας το API endpoint της Marketplace υπηρεσίας όπως φαίνεται στην Εικόνα 37. Σε αυτήν την κλήση δηλώνεται ο ιδιοκτήτης, ο αναγνωριστικός κωδικός του προϊόντος και ο αριθμός των προσπελάσεων που θα κάνει στους πόρους του προϊόντος.



Εικόνα 37:Πρόσβαση σε ένα προϊόν από το Marketplace API.

4 Υλοποίηση δοκιμαστικής εφαρμογής

Η δράση αυτή ασχολείται με την υλοποίηση μιας mobile δοκιμαστικής εφαρμογής IoT για android η οποία οπτικοποιεί τις παρεχόμενες λειτουργίες της ομοσπονδίας και θα επικυρώσει τις ενέργειες που λαμβάνουν χώρα στο ενδιάμεσο λογισμικό της IoT Ομοσπονδίας. Δεδομένου ότι η Ομοσπονδία IoT είναι φανερή μόνο στις συμμετέχουσες πλατφόρμες IoT, η δοκιμαστική εφαρμογή IoT που αναπτύχθηκε χρησιμοποιείται για λόγους επίδειξης των αποτελεσμάτων της πιλοτικής λειτουργίας της IoTFeds πλατφόρμας.

4.1 Περιγραφή δοκιμαστικής εφαρμογής IoT

Η δοκιμαστική εφαρμογή, αναλαμβάνει την εμφάνιση των δεδομένων στον χρήστη που εστιάζουν στην περιοχή που ορίζεται από την θέση που είναι ο ίδιος μέχρι τους χώρους στάθμευσης με σκοπό την εύκολη μετάβαση του. Τα δεδομένα απεικονίζονται σε ένα χάρτη και αφορούν τα δεδομένα κίνησης (με ποια ταχύτητα είναι η κίνηση στην πόλη), τα δεδομένα των χώρων στάθμευσης εκείνη την ώρα (πόσες ελεύθερες θέσεις υπάρχουν και ποιες) καθώς και τα τρέχοντα μετεωρολογικά δεδομένα.

Η mobile εφαρμογή αναπτύχθηκε με το framework Flutter της Google το οποίο χρησιμοποιεί τη γλώσσα Dart. Η όλη εφαρμογή στηρίζεται σε συγκεκριμένο endpoint από την IoT πλατφόρμα zastel, το οποίο παίρνει πρόσβαση από το ανάλογο product από το IoTFeds σύστημα. Συγκεκριμένα για να έχουμε τα δεδομένα που επιθυμούμε χρησιμοποιούμε το endpoint του zastel:

https://zastel.com/api/v1/messages/traffic-data?region=athens

με body για ιστορικά δεδομένα:

{ "latitude": 37.977818, "longitude": 23.727906, "startTimestamp": 1687754761556,

"radius": 10000, "minSpeed": 0, "maxSpeed": 500, "startHour": 0, "endHour": 23 }.

Αυτό το endpoint επιστρέφει στην mobile εφαρμογή τα δεδομένα κίνησης, parking και καιρού με την εξής μορφή:


```
9   {
10    "trafficData": {
11      "length": length,
12      "data": []
13    },
14    "parkingData": {
15      "length": 1,
16      "data": []
17    },
18    "weatherData":{
19      "dt": "2023-07-04 13:02:40",
20      "temperature": "35.5",
21      "humidity": 26,
22      "luminosity": "3775.00",
23      "wind_speed": "0.00",
24      "co": "2.58",
25      "co2": "6.82",
26      "dry": 1,
27      "rain_percentage": 0,
28      "atmospheric_pressure": "999.20"
29    }
30  };
```


Εικόνα 38: JSON αντικείμενο της mobile εφαρμογής που βασίζεται στα δεδομένα που ελήφθησαν από την IoTFeds πλατφόρμα.

Για real time είναι το ίδιο χωρίς το startTimestamp, το οποίο με τη σειρά του καλεί από το Marketplace του IoTFeds ρο ακόλουθο endopoint:

https://symbiote-core.iotfeds.intracom-telecom.com/marketplace/product/access

με body:

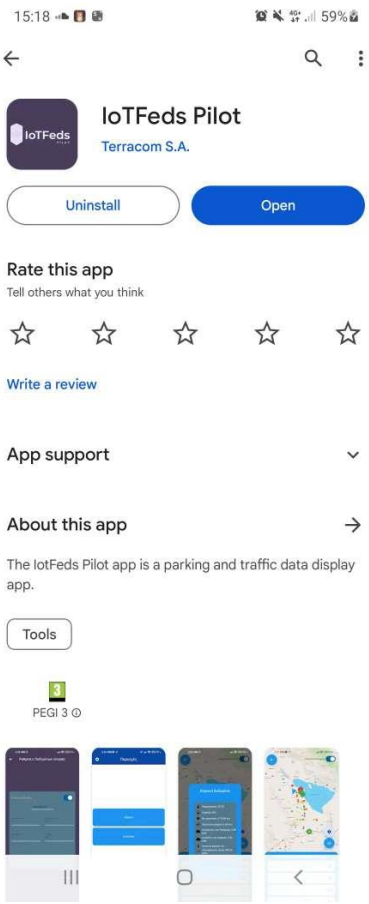
{ "credentials": { "clientId": "clientId", "password": "password", "platformId": "platformid", "username": "username"},

"productId": "productId", “dateFrom": "2023-11-22T00:00:00.000Z", "dateTo": "2023-12-30T00:00:00.000Z", "reqObservations": "3"}

Ο χρήστης της mobile εφαρμογής με αυτόν τον τρόπο έχει τα χρήσιμα δεδομένα για τον ίδιο μέσω του IoTFeds και έτσι η εφαρμογή είναι έτοιμη προς χρήση χωρίς περαιτέρω βήματα. Τα δεδομένα χρήζουν ανανέωσης για να ενημερώνεται ο χρήστης για αυτό και καλούνται οι προηγούμενες διαδικασίες ανά λεπτό.

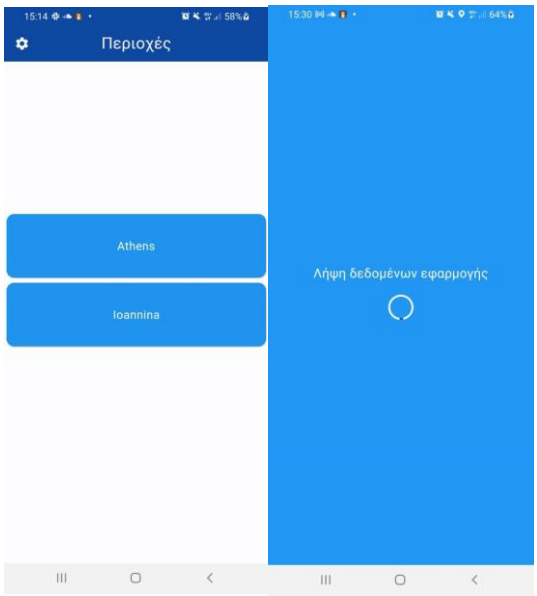
4.2 Επαλήθευση της δοκιμαστικής εφαρμογής IoT

Στο πλαίσιο της δοκιμής της υλοποίησης ο χρήστης android πρέπει να κατεβάσει την εφαρμογή από το playstore: https://play.google.com/store/apps/details?id=com.iotFeds.app.



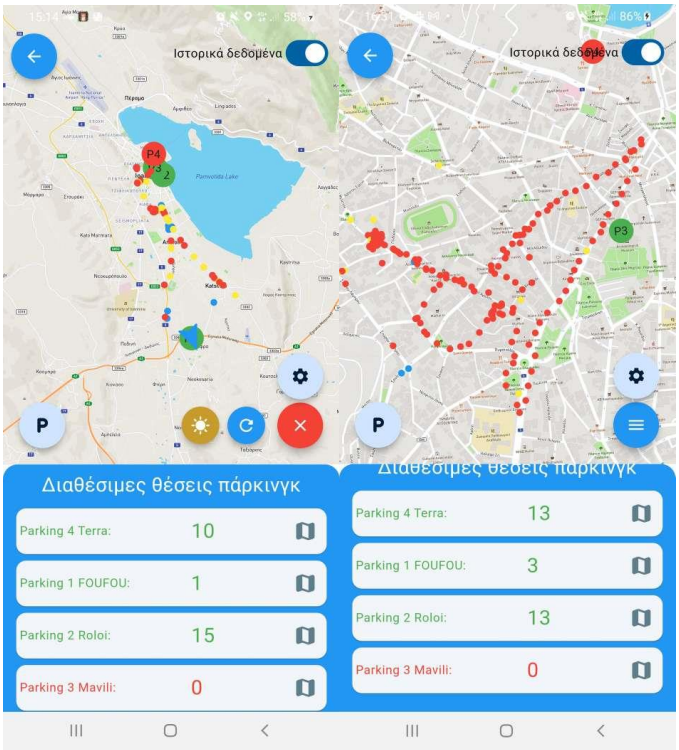
Εικόνα 39: Η εφαρμογή στο Playstore για την λήψη της εφαρμογής.

Το πρώτο βήμα του χρήστη είναι να επιλέξει την περιοχή στην οποία βρίσκεται. Σε αυτήν την περίπτωση έχουμε θέσει τις δύο γενικές περιοχές με βάση την πόλη, την Αθήνα αλλά και τα Ιωάννινα. Ακολούθως περιμένουμε για λίγα δευτερόλεπτα μέχρι ο χρήστης να μπορεί να έχει τα δεδομένα με βάση την επιλογή του.



Εικόνα 40: Επιλογή περιοχή και λήψη δεδομένων.

Στη συνέχεια ο χρήστης θα εισέλθει στην κεντρική σελίδα όπου παρατηρούμε με το μπλε βελάκι την θέση του καθώς με τους μικρότερους κύκλους απεικονίζονται οι διάφορες θέσεις που είχαμε με βάση την ταχύτητα. Στην περίπτωση που η ταχύτητα είναι μικρότερη ή ίση με 10 χλμ/ώρα απεικονίζεται με κόκκινο χρώμα ενώ στην περίπτωση που είναι μεγαλύτερη αλλά μικρότερη ή ίση με 25 χλμ/ώρα με κίτρινο. Αν είναι ακόμα μεγαλύτερη αλλά μικρότερη ή ίση με 40 χλμ/ώρα απεικονίζεται με μπλε ενώ σε ακόμα μεγαλύτερες ταχύτητες με πράσινο.

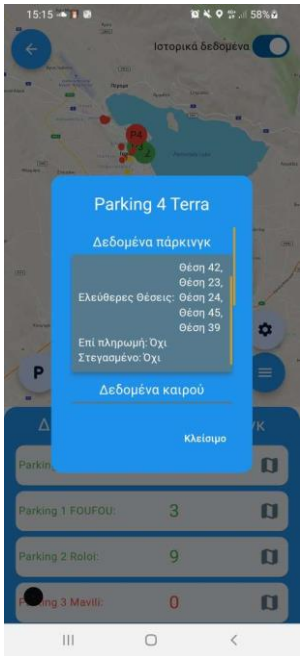


Εικόνα 41: Εμφάνιση δεδομένων κίνησης και parking στο χάρτη.

Στην κεντρική σελίδα παρατηρούμε στο κάτω μέρος της οθόνης μια λίστα με τα parkings. Όλα έχουν πράσινο χρώμα στην περιγραφή και με τις διαθέσιμες θέσεις στην διπλανή στήλη εκτός από εκείνα που δεν έχουν ελεύθερες θέσεις και για αυτό το λόγο είναι με κόκκινο χρώμα. Η ίδια λογική με τα χρώματα ακολουθείται και στην απεικόνιση τους στο χάρτη.

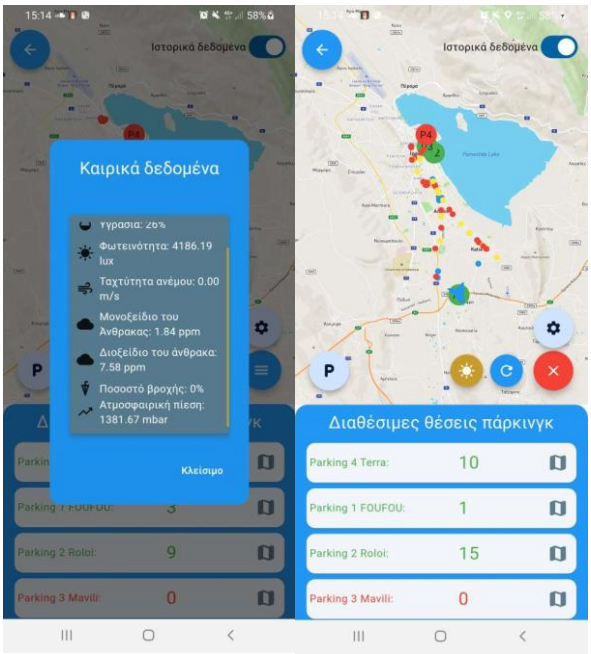
Στα δεξιά της γραμμής που αφορά κάθε ένα parking υπάρχει ένα εικονίδιο με έναν χάρτη, με το οποίο κατευθύνεται ο χρήστης στο google maps για να αποκτήσει οδηγίες για το πως μπορεί να φτάσει εκεί.

Όταν ο χρήστης εισέρχεται στο χώρο του parking εμφανίζεται αυτόματα ένα πλαίσιο διαλόγου με τις πληροφορίες του parking. Αναλυτικότερα πόσες είναι οι ελεύθερες θέσεις parking, ποιες είναι, αν είναι επί πληρωμή καθώς και αν είναι στεγασμένες. Αφού κλείσει τον διάλογο και για όσο βρίσκεται μέσα στο Parking εμφανίζεται ακριβώς πάνω από τη λίστα στα αριστερά ένα εικονίδιο με το γνωστό σύμβολο «P» με το οποίο μπορεί να ανοίξει ξανά τον διάλογο.



Εικόνα 42:Εμφάνιση Δεδομένων Parking.

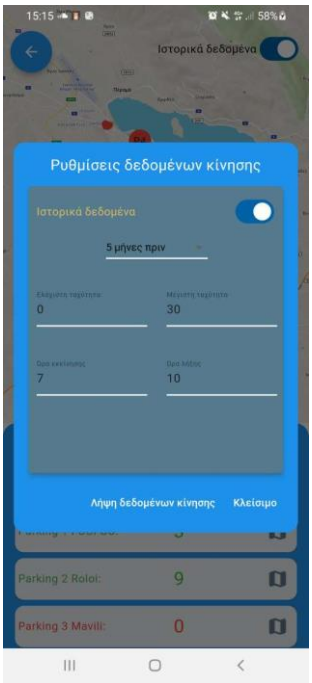
Κάτω από τα αναλυτικά στοιχεία για το κάθε parking υπάρχουν τα περιβαλλοντικά δεδομένα. Τα περιβαλλοντικά δεδομένα έχουν πληροφορία σχετικά με τη θερμοκρασία, την υγρασία, τη φωτεινότητα, την ταχύτητα του ανέμου, το μονοξείδιο και το διοξείδιο του άνθρακα, το ποσοστό της βροχής καθώς και την ατμοσφαιρική πίεση. Σε αυτή τη σελίδα ο χρήστης μπορεί να φτάσει και από τα δεξιά πάνω ακριβώς από τη λίστα με τα parkings πατώντας το κουμπί για το μενού και ακολούθως το κίτρινο εικονίδιο με τον ήλιο.



Εικόνα 43:Εμφάνιση καιρικών δεδομένων και menu λειτουργιών.

Στο μεσαίο εικονίδιο με το γνωστό σύμβολο για το «Refresh» υπάρχει η δυνατότητα ο χρήστης να καλέσει ξανά τα δεδομένα κίνησης πέρα από το περιοδικό χρόνο του ενός λεπτού που καλούνται. Πάνω δεξιά υπάρχει η επιλογή αν τα δεδομένα θα είναι πραγματικού χρόνου ή θα βασίζονται σε ιστορικά δεδομένα, δηλαδή αν βασίζονται σε μια σειρά ρυθμίσεων που μπορεί

ο χρήστης να θέσει όταν επιλέξει το σύμβολο των ρυθμίσεων που απεικονίζεται με το γρανάζι στα δεξιά.



Εικόνα 44: Ρυθμίσεις λήψης ιστορικών δεδομένων.

Σε αυτό το παράθυρο ο χρήστης μπορεί να επιλέξει το διάστημα για τα ιστορικά δεδομένα μεταξύ 5 επιλογών από μια εβδομάδα μέχρι 5 μήνες πριν, την μέγιστη και την ελάχιστη ταχύτητα καθώς και την ώρα εκκίνησης και λήξης.

5 Συμπεράσματα

Στο κείμενο αυτό περιγράφηκε η προετοιμασία του περιβάλλοντος των IoT πλατφορμών και η υλοποίηση της δοκιμαστικής εφαρμογής που απαιτούνται για την υλοποίηση και διεξαγωγή του σεναρίου δοκιμής. Οι τρεις πιλοτικές IoT πλατφόρμες (Zastel/spotypal, Smart environment, uiTOP) που συλλέγουν κυκλοφοριακά δεδομένα, περιβαλλοντικά δεδομένα και δεδομένα έξυπνης στάθμευσης, δημιουργούν IoT ομοσπονδία για τη διάθεση προϊόντος δεδομένων από τις ομόσπονδες πηγές δεδομένων τους. Για την σύνδεση των πλατφορμών στο σύστημα, πραγματοποιήθηκε η διαμόρφωση κάθε IoT πλατφόρμας με την εγκατάσταση του απαραίτητου ενδιάμεσου λογισμικού και έγινε η διάθεση των σχετικών IoT πηγών δεδομένων για τη διάθεση δεδομένων από αυτήν μέσω του IoTFeds συστήματος. Τέλος, υλοποιήθηκε μία δοκιμαστική IoT εφαρμογή (ενός παρόχου εφαρμογής) για την παροχή μίας ολοκληρωμένης λύσης που συνδυάζει τα δεδομένων IoT από τις πηγές δεδομένων (προϊόντος δεδομένων) των τριών πιλοτικών εφαρμογών μέσω της χρήσης της πλατφόρμας IoTFeds. Η εφαρμογή αυτή αποτελεί τη βάση για την επίδειξη χρήσης των λειτουργιών της IoTFeds πλατφόρμας που θα περιγραφεί στη Δράση Δ4.3 «Πιλοτική Εφαρμογή Συστήματος».

Βιβλιογραφία

Polymeni, S., Spanos, G., Tsiktsiris, D., Athanasakis, E., Votis, K., Tzovaras, D., & Kormentzas, G. (2023). everWeather: A Low-Cost and Self-Powered AIoT Weather Forecasting Station for Remote Areas. *EnviroInfo.* Springer.