



TANDEM

Ενσωμάτωση λειτουργιών και
συσκευών σε υπηρεσίες νέφους στις
παρυφές του δικτύου

Παραδοτέο Π2.2: Λογισμικό Συστήματος – Έκδοση 2^η

(Τεκμηρίωση, Οδηγίες Εγκατάστασης και Εγχειρίδιο Χρήσης)



Ευρωπαϊκή Ένωση
Ευρωπαϊκά Διαρθρωτικά
και Επενδυτικά Ταμεία



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΕΠΕΝΔΥΣΕΩΝ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΠΡΟΓΡΑΜΜΑΤΩΝ ΕΤΠΑ, ΤΣ & ΕΚΤ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑΝΕΚ

ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Έργο	TANDEM
Κωδικός Έργου	T2ΕΔΚ-02825
Ενότητα Εργασίας	ΕΕ2: Λεπτομερής Σχεδιασμός και Ανάπτυξη Συστήματος
Συντονιστής Ενότητας	INTRACOM TELECOM
Συντονιστής Παραδοτέου	INTRACOM TELECOM
Έκδοση Παραδοτέου	1.0
Ημερομηνία	31/08/2022

Πίνακας Περιεχομένων

Ακρωνύμια	9
Βασικές Έννοιες	12
Μεταφράσεις Βασικών Όρων	16
Σύνοψη Παραδοτέου	17
1 Εισαγωγή	19
1.1 Το έργο TANDEM	19
1.2 Αντικείμενο Παραδοτέου	21
1.3 Δομή Παραδοτέου	21
2 2 ^η Έκδοση του Λογισμικού Συστήματος TANDEM	23
3 Υλοποίηση Λογισμικού Κόμβων Άκρων (Παρυφών)	26
3.1 Διαχειριστής Πλατφόρμας Παρυφών του TANDEM (TANDEM Edge Platform Manager - ri-Edge)	26
3.1.1 Εισαγωγή	26
3.1.2 Αρχιτεκτονική και Ανάλυση του Διαχειριστή Πλατφόρμας Παρυφών	27
3.2 Edge Monitoring	33
3.3 Multi-access Edge Computing Platform	35
3.4 TANDEM Βασικές και Υποστηρικτικές Υπηρεσίες	44
3.4.1 IoT Data Analytics	46
3.4.2 TANDEM Object Detection Service	47
3.4.3 Frame Grabber	56
3.5 TANDEM App services	56
3.5.1 Αυτοματοποιημένη διαδικασία εγκατάστασης	56
3.5.2 Υπηρεσία Ευφυούς Στάθμευσης (Smart Parking Service)	57
3.5.3 Υπηρεσία Επόπτευσης & Ασφάλειας Χώρου (Surveillance & Security Service)	59
4 Υλοποίηση Λογισμικού Διαχειριστή Συστήματος TANDEM	62
4.1 TANDEM Portal	62
4.1.1 Σχεδιασμός	63
4.2 Service Orchestrator (Ενορχηστρωτής Υπηρεσιών)	63
4.2.1 Σχεδιασμός	64
4.2.2 Εργαλεία	64
4.2.3 Υλοποίηση Καταλόγου Υπηρεσιών (Service Catalogue)	65

4.2.4	Υλοποίηση Διαχειριστή Αλυσίδων Υπηρεσιών (Service Chain Manager – SCO)	71
4.2.5	Υλοποίηση Διαχειριστή Χρηστών (User Manager).....	73
4.2.6	Υλοποίηση Διαχειριστή Υποδομής (Infrastructure Manager).....	77
4.3	Ενορχηστρωτής Άκρων TANDEM (TANDEM Edge Orchestrator).....	78
5	Οδηγίες Εγκατάστασης	80
5.1	Εγκατάσταση Λογισμικού Κόμβων Άκρων (Παρυφών).....	80
5.1.1	Προετοιμασία των κόμβων TANDEM.....	80
5.1.2	Παροχή – εγκατάσταση βασικών (core) TANDEM PaaS υπηρεσιών	87
5.2	Εγκατάσταση Πλατφόρμας Διαχείρισης Συστήματος.....	96
5.2.1	Εγκατάσταση του TANDEM Service Catalogue	96
5.2.2	Εγκατάσταση του Nginx σαν reverse proxy και application server	99
5.2.3	Εγκατάσταση του TANDEM Infrastructure Manager	102
5.2.4	Εγκατάσταση του TANDEM User Manager	103
5.2.5	Εγκατάσταση του TANDEM Edge Orchestrator	104
6	Οδηγός Χρήσης.....	107
6.1	Κεντρική Σελίδα και Σύνδεση Χρήστη	107
6.2	Διαχείριση Εφαρμογών	108
6.2.1	Αναζήτηση Εφαρμογών.....	109
6.2.2	Αλλαγή Κατάστασης Εφαρμογής	110
6.2.3	Διαγραφή Εφαρμογής	110
6.2.4	Εκτέλεση Εφαρμογής	110
6.2.5	Παρακολούθηση Χρήσης Πόρων Εφαρμογής.....	111
6.3	Διαχείριση Υπηρεσιών (Κατάλογος Υπηρεσιών).....	112
6.3.1	Αναζήτηση Υπηρεσίας	113
6.3.2	Προβολή Λεπτομερειών Υπηρεσίας και Αλλαγή των Στοιχείων της.....	114
6.3.3	Αλλαγή Κατάστασης Υπηρεσίας.....	115
6.3.4	Διαγραφή Υπηρεσίας	116
6.3.5	Δημιουργία Προϊόντος από μια Υπηρεσία.....	116
6.3.6	Καταχώρηση Υπηρεσίας.....	117
6.3.7	Τροποποίηση Υπηρεσίας.....	120
6.4	Διαχείριση Αλυσίδων Υπηρεσιών	120
6.4.1	Δημιουργία Αλυσίδας Υπηρεσιών.....	120
6.4.2	Προσθήκη κόμβου.....	121

6.4.3	Αποθήκευση Αλυσίδας Υπηρεσιών	124
6.4.4	Επεξεργασία Υπάρχουσας Αλυσίδας	125
6.4.5	Διαγραφή Αλυσίδας Υπηρεσιών	125
6.5	Διαχείριση Χρηστών	126
6.5.1	Καταχώρηση Χρήστη	127
6.5.2	Τροποποίηση Χρήστη	129
6.5.3	Είσοδος Χρήστη	129
6.5.4	Αναζήτηση Χρηστών	129
6.5.5	Διαγραφή Χρήστη	130
6.6	Διαχείριση Υποδομής	130
6.6.1	Αναζήτηση Edge Clouds	131
6.6.2	Διαγραφή Edge Cloud	131
6.6.3	Παρακολούθηση χρήσης πόρων Edge Cloud	131
6.6.4	Προβολή Nodes ενός Edge Cloud	132
6.6.5	Καταχώρηση Νέου Node σε Edge Cloud	133
6.6.6	Διαγραφή Node	133
6.6.7	Παρακολούθηση χρήσης πόρων Node	134
6.6.8	Παρακολούθηση των services ενός Node	135
6.7	Διαχείριση Συσκευών	136
6.7.1	Αναζήτηση Συσκευών	137
6.7.2	Διαγραφή Συσκευής	137
6.8	Διαχείριση Προϊόντων (Κατάλογος Προϊόντων)	138
6.8.1	Αναζήτηση Προϊόντος	139
6.8.2	Διαγραφή Προϊόντος	139
6.8.3	Δημιουργία Προϊόντος από μια Υπηρεσία	139
6.8.4	Καταχώρηση Προϊόντος	139
6.8.5	Τροποποίηση Προϊόντος	141
6.9	Διαχείριση ρόλων χρηστών	141
6.9.1	Δυνατότητες πρόσβασης ρόλου	142
7	Πρωτότυπα Αλυσίδων Υπηρεσιών (Service Chains Prototypes)	144
7.1	Αλυσίδα Υπηρεσιών Διαδικτύου των Πραγμάτων (IoT Service Chain)	144
7.2	Αλυσίδα Υπηρεσιών Αναγνώρισης Αντικειμένων (Object Detection Service Chain)	147
8	TANDEM Application	150

8.1	Parking Application.....	150
9	Συμπεράσματα και Επόμενες Εκδόσεις Λογισμικού.....	151
10	Αναφορές	154
11	ΠΑΡΑΡΤΗΜΑ - Προετοιμασία & Εγκατάσταση	155
11.1	Προετοιμασία Υπολογιστικού Εξυπηρετητή (server).....	155
11.2	Δημιουργία Εικονικού-Νέφους για εγκατάσταση Kubernetes με Ένα(1) Master & N Workers	156
11.3	Εγκατάσταση Απαραίτητου Software στο Εικονικό-Νέφος.....	159
11.4	Εγκατάσταση Προϊόντος TANDEM σε Υποδομή KUBERNETES	175
11.5	Εισαγωγή Worker-Nodes στο Οικοσύστημα του Kubernetes	185
11.6	Εγκατάσταση Βοηθητικών Υπηρεσιών στο Οικοσύστημα Kubernetes.....	187

Ευρετήριο Εικόνων

Εικόνα 1:	Αρχιτεκτονική υψηλού επιπέδου του Διαχειριστή Πλαφρόσματος Άκρων	27
Εικόνα 2:	Μέθοδος αυθεντικοποίησης TANDEM χρήστη.....	29
Εικόνα 3:	Μέθοδοι διαχείρισης εγγραφών Service Function	29
Εικόνα 4:	Μέθοδοι διαχείρισης εγγραφών PaaS υπηρεσιών	30
Εικόνα 5:	Μέθοδος εύρεσης και εγγραφής κόμβων στον κατάλογο	31
Εικόνα 6:	Μέθοδος διαχείρισης και εγκατάστασης PaaS υπηρεσιών	32
Εικόνα 7:	Μέθοδος ενεργοποίησης TANDEM κόμβου	33
Εικόνα 8:	Αρχιτεκτονική του συστήματος παρακολούθησης "Promethues-Operator"	35
Εικόνα 9:	Υψηλού Επιπέδου Αρχιτεκτονική του MEC.....	36
Εικόνα 10:	Το EdgeXFoundry ανάμεσα στα Αντικείμενα και στο Πληροφοριακό Σύστημα ..	46
Εικόνα 11:	Η Αρχιτεκτονική του EdgeXFoundry.....	46
Εικόνα 12	- Δοκιμαστική Χρήση της Υπηρεσίας Smart Parking στο parking του κτιρίου A5 της INTRACOM TELECOM	59
Εικόνα 13-	Δοκιμαστική Χρήση της Υπηρεσίας Επόπτευσης και Ασφάλειας στον περιβάλλον χώρο της εταιρείας INTRACOM TELECOM –Εντοπισμός Ανθρώπου	61
Εικόνα 14	-Αυτόματη Ανάγνωση Πινακίδας Οχήματος για έλεγχο είσοδου -Εγκαταστάσεις INTRACOM TELECOM	61
Εικόνα 15:	Η Υλοποίηση του Διαχειριστή Συστήματος TANDEM	62
Εικόνα 16:	Η δομή των χρηστών στο Tandem	76
Εικόνα 17:	Η κεντρική ενημερωτική σελίδα της πλατφόρμας TANDEM	107
Εικόνα 18:	Το παράθυρο σύνδεσης του χρήστη στην πλατφόρμα TANDEM.....	108
Εικόνα 19:	Εικονίδιο "Details" στις διαθέσιμες ενέργειες για μία εφαρμογή	109
Εικόνα 20:	Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων εφαρμογών στην πλατφόρμα	109
Εικόνα 21:	Γραμμή αναζήτησης εφαρμογών	110

Εικόνα 22: Εικονίδιο “Next state” στις διαθέσιμες ενέργειες για μία εφαρμογή.....	110
Εικόνα 23: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία εφαρμογή.....	110
Εικόνα 24: Εικονίδιο “Run application” στις διαθέσιμες ενέργειες για μία εφαρμογή	111
Εικόνα 25: Η παρουσίαση των μετρήσεων της εφαρμογής κατά την εκτέλεσή της.....	111
Εικόνα 26: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για μία εφαρμογή	111
Εικόνα 27: Παρουσίαση μετρικών χρήσης πόρων των υπηρεσιών μίας εφαρμογής	112
Εικόνα 28: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων υπηρεσιών στην πλατφόρμα	113
Εικόνα 29: Γραμμή αναζήτησης υπηρεσιών	113
Εικόνα 30: Εικονίδιο “Edit” στις διαθέσιμες ενέργειες για μία υπηρεσία	114
Εικόνα 31: Λειτουργικότητα του εικονιδίου “Edit” στις διαθέσιμες ενέργειες για μία υπηρεσία	115
Εικόνα 32: Εικονίδιο “Change state” στις διαθέσιμες ενέργειες για μία υπηρεσία	115
Εικόνα 33: Λειτουργικότητα του εικονιδίου “Change state” στις διαθέσιμες ενέργειες για μία υπηρεσία	116
Εικόνα 34: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία υπηρεσία	116
Εικόνα 35: Εικονίδιο “Create product” στις διαθέσιμες ενέργειες για μία υπηρεσία	116
Εικόνα 36: Φόρμα εγγραφής προϊόντος στην πλατφόρμα.....	117
Εικόνα 37: Φόρμα εγγραφής υπηρεσίας στην πλατφόρμα.....	118
Εικόνα 38: Το περιβάλλον δημιουργίας αλυσίδας υπηρεσιών	121
Εικόνα 39: Η εισαγωγή του “Start” ως πρώτου κόμβου της αλυσίδας υπηρεσιών	121
Εικόνα 40: Το αναδυόμενο παράθυρο παραμέτρων ενός κόμβου	122
Εικόνα 41: Η ανανεωμένη ετικέτα ενός κόμβου μετά από επεξεργασία της από τον χρήστη	122
Εικόνα 42: Το παράθυρο συσχέτισης ενός κόμβου με μία υπηρεσία	123
Εικόνα 43: Το εικονίδιο βέλους “Add Link” σε έναν κόμβο για την προσθήκη συνδέσμου	123
Εικόνα 44: Η σύνδεση μεταξύ δύο κόμβων	124
Εικόνα 45: Το εικονίδιο βέλους “ Remove Links” σε έναν κόμβο για την αφαίρεση συνδέσμων	124
Εικόνα 46: Το εικονίδιο “Remove Node” σε έναν κόμβο για την αφαίρεσή του	124
Εικόνα 47: Το αναδυόμενο παράθυρο αποθήκευσης μίας αλυσίδας υπηρεσιών.....	125
Εικόνα 48: Το αναδυόμενο παράθυρο ανοίγματος και επεξεργασίας υπάρχουσας αλυσίδας υπηρεσιών.....	125
Εικόνα 49: Το αναδυόμενο παράθυρο διαγραφής μίας αλυσίδας υπηρεσιών	126
Εικόνα 50: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων χρηστών στην πλατφόρμα.....	127
Εικόνα 51: Φόρμα εγγραφής χρήστη στην πλατφόρμα	128
Εικόνα 52: Ο προαιρετικά εμφανιζόμενος πίνακας διεύθυνσης χρέωσης του χρήστη	129
Εικόνα 53: Εικονίδιο “Edit” στις διαθέσιμες ενέργειες για έναν χρήστη	129
Εικόνα 54: Μήνυμα χαιρετισμού του χρήστη μετά από είσοδό του στην πλατφόρμα	129
Εικόνα 55: Γραμμή αναζήτησης χρηστών	130
Εικόνα 56: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για έναν χρήστη	130
Εικόνα 57: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων νεφών άκρου στην πλατφόρμα	130
Εικόνα 58: Γραμμή αναζήτησης νεφών άκρου	131

Εικόνα 59: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για ένα νέφος άκρου	131
Εικόνα 60: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για ένα νέφος άκρου	131
Εικόνα 61: Παρουσίαση μετρικών χρήσης πόρων από ένα νέφος άκρου	132
Εικόνα 62: Εικονίδιο “Info” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου	133
Εικόνα 63: Το αναδυόμενο παράθυρο παρουσίασης των κόμβων ενός νέφους άκρου	133
Εικόνα 64: Εικονίδιο “Register Node” στις διαθέσιμες ενέργειες για ένα νέφος άκρου	133
Εικόνα 65: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για έναν κόμβο του νέφους άκρου	134
Εικόνα 66: Εικονίδιο “Monitor Node” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου	134
Εικόνα 67: Παρουσίαση μετρικών χρήσης πόρων για έναν κόμβο ενός νέφους άκρου	135
Εικόνα 68: Εικονίδιο “Monitor Services” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου	135
Εικόνα 69: Παρουσίαση μετρικών χρήσης πόρων από τις υπηρεσίες ενός νέφους άκρου	136
Εικόνα 70: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων συσκευών στην πλατφόρμα	137
Εικόνα 71: Γραμμή αναζήτησης συσκευών	137
Εικόνα 72: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία συσκευή	138
Εικόνα 73: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων προϊόντων στην πλατφόρμα	138
Εικόνα 74: Γραμμή αναζήτησης προϊόντων	139
Εικόνα 75: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για ένα προϊόν	139
Εικόνα 76: Φόρμα εγγραφής προϊόντος στην πλατφόρμα	140
Εικόνα 77 – Οθόνη Διαχείρισης των ρόλων χρηστών του συστήματος	142
Εικόνα 78 - Συσχέτιση Ρόλων Χρηστών με δικαιώματα πρόσβασης σε λειτουργίες	143
Εικόνα 79: Η ροή εργασιών του IoT (κάτω) και πως αναπαριστάται αναλυτικά στην σελίδα διαχείρισης αλυσίδων του TANDEM portal (πάνω).	144
Εικόνα 80: Τα δομικά στοιχεία του Argo framework σε περιβάλλον Kubernetes	144
Εικόνα 81 - Αλυσίδα Υπηρεσιών Αναγνώρισης Αντικειμένων	148

Ακρωνύμια

Ακρωνύμιο	Ορισμός
AR	Augmented Reality (Επαυξημένη Πραγματικότητα)
API	Application Programming Interface (Διεπαφή Προγραμματισμού Εφαρμογών)
AWS	Amazon Web Service
BaaS	Backend-as-a-Service (Το Backend ως Υπηρεσία)
BPMN	Business Process Model and Notation (Μοντέλο Επιχειρηματικής Διαδικασίας και Συμβολισμοί)
CDA	Confirmatory Data Analysis (Ανάλυση Επιβεβαιωτικών Δεδομένων)
CDN	Content Delivery Networks (Δίκτυο Διανομής Περιεχομένου)
CFS	Customer Facing Service (Εξυπηρέτηση πελατών)
CP	Connection Point (Σημείο Σύνδεσης)
CSP	Communication Service Provider (Πάροχος Υπηρεσιών Επικοινωνιών)
CV	Computer Vision (Μηχανική, Υπολογιστική ή Τεχνητή Όραση)
DAG	Directed Acyclic Graph (Κατευθυνόμενος Ακυκλικός Γράφος)
DC	Datacenter (Κέντρο δεδομένων)
DMN	Decision Model and Notation (Μοντέλο απόφασης και Συμβολισμοί)
DPI	Deep Packet Inspection (Εις βάθος εξέταση πακέτων)
EC	Edge Computing (Υπολογιστική στα άκρα του δικτύου)
EDA	Exploratory Data Analysis (Διερευνητική Ανάλυση Δεδομένων)
ETL	Extract-Transform-Load (Εξαγωγή-μετασχηματισμός-φόρτωση)
FaaS	Function-as-a-Service (Η Λειτουργία ως Υπηρεσία)
FQDN	Fully Qualified Domain Name (Πλήρως αναγνωρισμένο όνομα τομέα)
GTP	GPRS Tunneling Protocol (Πρωτόκολλο σήραγγας GPRS)
GUI	Graphical User Interface (Γραφικό Περιβάλλον Χρήστη)
HLS	HTTP Live Streaming (Ζωντανή μετάδοση HTTP)
ICN	Information-Centric Network (Δίκτυο προσανατολισμένο στην πληροφορία)
IOPS	Input/Output Operations Per Second (Αριθμός εισόδων/εξόδων ανά δευτερόλεπτο)
IoT	Internet of Things (Το Διαδίκτυο των Αντικειμένων)
ISP	Internet Service Provider (Πάροχος Υπηρεσιών Διαδικτύου)
KPI	Key Performance Indicator (Δείκτης απόδοσης)
μDC	Micro-datacenter (Μικρο-κέντρο δεδομένων)

MANO	Management and orchestration (Διαχείριση και ενορχήστρωση)
MCDM	Multi-criteria decision making (Λήψη αποφάσεων πολλαπλών κριτηρίων)
MEC	Multi-access edge computing (Υπολογιστική άκρων)
MIME	Multipurpose Internet Mail Extension (Επέκταση για πολλαπλές χρήσεις σε αλληλογραφία μέσω διαδικτύου)
NFV	Network Function Virtualization (Εικονικοποίηση δικτυακών λειτουργιών)
NFVO	Network Function Virtualization Orchestration (Ενορχήστρωση εικονικοποιημένων δικτυακών λειτουργιών)
NSD	Network Service Descriptor (Περιγραφέας δικτυακής υπηρεσίας)
NST	Network Slice Template (Πρότυπο δικτυακού τεμαχίου)
NSI	Network Slice Instance (Στιγμιότυπο δικτυακού τεμαχίου)
NSSI	Network Slice Subnet Instance (Στιγμιότυπο υποδικτύου δικτυακού τεμαχίου)
OS	Operating System (Λειτουργικό Σύστημα)
OU	Organizational Units (Οργανωτικές Μονάδες)
PNF	Physical network function (Φυσική δικτυακή λειτουργία)
POP	Point of Presence (Σημείο Πρόσβασης)
PPDR	Public protection and disaster relief (Δημόσια προστασία και ανακούφιση από καταστροφές)
QCI	QoS Class Identifier (Αναγνωριστικό κλάσης ποιότητας υπηρεσίας)
QoE	Quality of Experience (Ποιότητα Εμπειρίας Χρήστη)
QoS	Quality of Service (Ποιότητα Παροχής Υπηρεσίας)
PaaS	Platform as a Service (Πλατφόρμα ως Υπηρεσία)
PLC	Programmable Logic Controller (Προγραμματιζόμενος Λογικός Ελεγκτής)
RaaS	Robot as a Service (Ρομπότ ως Υπηρεσία)
RAN	Radio Access Network (Δίκτυο Πρόσβασης Ραδιοεπικοινωνίας)
SaaS	Software as a Service (Λογισμικό ως Υπηρεσία)
SDN	Software-defined networking (Δίκτυα οριζόμενα μέσω λογισμικού)
SLA	Service Level Agreement (Συμφωνητικό Παροχής Υπηρεσίας)
SLAM	Simultaneous localization and mapping (Ταυτόχρονος εντοπισμός και χαρτογράφηση)
SR	Source Routing (Πηγαία δρομολόγηση)
TCAM	Ternary Content Addressable Memory (Τριαδική μνήμη αναζήτησης περιεχομένου)
TTL	Time-to-Live (Χρόνος ζωής)
UML	Unified Modelling Language (Ενοποιημένη γλώσσα σχεδιασμού)

UE	User Equipment
URL	Uniform Resource Locator (Ενιαίος εντοπιστής πόρων)
vCache	Virtual cache (Εικονικοποιημένη κρυφή μνήμη)
VDU	Virtual Deployment Unit (Μονάδα Εικονικής Ανάπτυξης)
VIM	Virtualized Infrastructure Manager (Διαχειριστής Εικονικοποιημένης Υποδομής)
VLD	Virtual Link Descriptor (Περιγραφέας εικονικής σύνδεσης)
VNF	Virtual Network Function (Εικονικοποιημένη δικτυακή λειτουργία)
VNFD	Virtual Network Function Descriptor (Περιγραφέας εικονικοποιημένης δικτυακής λειτουργίας)
VoD	Video on Demand (Βίντεο κατά απαίτηση)
VNO	Virtual Network Operator (Εικονικός χειριστής δικτύου)
VR	Virtual Reality (Εικονική Πραγματικότητα)

Βασικές Έννοιες

Έννοια	Περιγραφή
Αναγνώριση Αντικειμένων (Object Recognition)	Είναι μια υπολογιστική τεχνολογία που σχετίζεται με την Τεχνητή Όραση (Computer Vision) και την Επεξεργασία Εικόνων (Image Processing) και ασχολείται με την ανίχνευση παρουσιών σημασιολογικών αντικειμένων μιας συγκεκριμένης κατηγορίας (όπως ανθρώπους, οχήματα, ζώα) σε ψηφιακές εικόνες και βίντεο. Έχει εφαρμογές σε πολλούς τομείς όπως στην Επιτήρηση Χώρου (Surveillance) και στην Ρομποτική βιομηχανία.
Εφαρμογή (Application)	Μία οργανωμένη ενότητα συνεργαζόμενων λειτουργιών που ικανοποιούν μια συγκεκριμένη ανάγκη του τελικού χρήστη (π.χ. εφαρμογή επιτήρησης χώρου ή ελέγχου λειτουργίας Data Room). Μπορεί να αντιστοιχεί σε μία Υπηρεσία Εφαρμογής ή να αποτελεί τη σύνθεση υπηρεσιών του TANDEM που μπορεί να λειτουργούν και σε διαφορετικούς κόμβους. Η εφαρμογή μπορεί να παρέχει ή να καταναλώσει υπηρεσίες του TANDEM. Λειτουργεί ως εικονικοποιημένη εφαρμογή (οι υπηρεσίες-συστατικά της υλοποιούνται σε εικονικές μηχανές (VMs) ή containers), πάνω από την υποδομή εικονικοποίησης που παρέχεται από το TANDEM. Μπορεί να έχει συσχετισμένους κανόνες και απαιτήσεις, όπως απαιτούμενους πόρους, μέγιστη καθυστέρηση, απαιτούμενες ή χρήσιμες υπηρεσίες κ.λπ. για κάθε συστατικό της. Αυτές οι απαιτήσεις επικυρώνονται από τη διαχείριση επιπέδου συστήματος TANDEM. Στην περίπτωση που αποτελεί το αποτέλεσμα σύνθεσης υπηρεσιών προδιαγράφεται με ένα δομημένο αρχείο κειμένου (με όλη τη λογική σύνδεσης των υπηρεσιών και των συσκευών) που περιέχει τις οδηγίες προς τις επιλεγμένες μηχανές διαχείρισης γεγονότων και ροών εργασιών (event & workflow engines) για την υλοποίηση της λογικής της εφαρμογής. Η ιεραρχία είναι: Εφαρμογή (Application) → Υπηρεσία (Service) → Function (Δομο-ενότητα ή Στοιχειώδης Υπηρεσία)
Διαδίκτυο των Αντικειμένων (Internet of Things – IoT)	Αποτελεί το δίκτυο επικοινωνίας πληθώρας φυσικών συσκευών και κάθε αντικειμένου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων με άλλες συσκευές μέσω του Διαδικτύου
Δομο-ενότητα (function)	Μια σειρά από λειτουργίες με συγκεκριμένο σκοπό που συνθέτουν τη μικρότερη δυνατή εγκαταστάσιμη μονάδα λογισμικού. Αντιστοιχεί στην έννοια της Εικονικοποιημένης Δικτυακής Λειτουργίας (Virtual Network Function) του NFV MANO. Σε αυτό το κείμενο χρησιμοποιείται ισοδύναμα με τον όρο Υπηρεσία (Service) και περιγράφει μια υπηρεσία που μπορεί να αποτελέσει δομική ενότητα άλλων πιο σύνθετων υπηρεσιών ή εφαρμογών (Στοιχειώδης Υπηρεσία). Η σύνθεση δομο-ενοτήτων μπορεί να οδηγήσει σε υπηρεσίες με τον ίδιο τρόπο που η σύνθεση απλών υπηρεσιών μπορεί να οδηγήσει σε πιο πολύπλοκες υπηρεσίες. Λειτουργεί ως εικονικοποιημένη υπηρεσία (σε εικονικές μηχανές (VMs) ή containers), πάνω από την υποδομή εικονικοποίησης που παρέχεται από το TANDEM. Μπορεί να έχει συσχετισμένους κανόνες και απαιτήσεις, όπως απαιτούμενους πόρους, μέγιστη καθυστέρηση, απαιτούμενες ή χρήσιμες υπηρεσίες κ.λπ. Αυτές οι απαιτήσεις επικυρώνονται από τη διαχείριση επιπέδου συστήματος TANDEM.

Εικονικοποιημένη Υποδομή (Virtualized Infrastructure)	Η εικονικοποιημένη υποδομή είναι μία συλλογή από συνιστώσες IT υπολογιστικού περιβάλλοντος που υλοποιούνται σε επίπεδο κώδικα. Μία τέτοια συνιστώσα που δημιουργείται με τη χρήση κατάλληλου κώδικα έχει τα ίδια χαρακτηριστικά με τον αντίστοιχο “χειροπιαστό” εξοπλισμό. Το πλεονέκτημα που έχει ένα διαχειριστής εικονοποιημένου συστήματος είναι η ανεξάρτηση του από συγκεκριμένους κατασκευαστές και προσόντα. Όμως το μεγαλύτερο όφελος είναι η καλύτερη χρήση των διαθέσιμων υποδομών - ελαχιστοποίηση του χρόνου που βρίσκεται ακριβός εξοπλισμός σε αχρηστία. Συνήθως ένα εικονοποιημένο υπολογιστικό κέντρο αποτελείται από τις ακόλουθες συνιστώσες: • Εικονικές Μηχανές (Virtual Machines) - προσφέρουν γενικά χαρακτηριστικά με συμβατικές δυνατότητες σε επεξεργασία, αποθήκευση και δίκτυο. • Εικονικές Επεξεργαστικές Μονάδες (Virtual Compute) - προσφέρουν servers με εξαιρετικές επεξεργαστικές ικανότητες που ξεπερνάνε τις δυνατότητες ενός μόνο μηχανήματος. • Εικονικές Αποθηκευτικές Μονάδες (Virtual Storage) - προσφέρουν servers με εξαιρετικές αποθηκευτικές ικανότητες που ξεπερνάνε τις δυνατότητες ενός μόνο μηχανήματος. • Εικονικά Δίκτυα (Virtualized Networks) - προσφέρουν δυνατότητες διασύνδεσης μεγάλης πολυπλοκότητας αποφεύγοντας την αγορά ακριβού εξοπλισμού.
Επιτήρηση (επίβλεψη) Χώρου (Surveillance)	Η Επιτήρηση-Χώρου είναι ένα ευρύ πεδίο-εφαρμογής υπολογιστών και προγραμματισμού με τη συμμετοχή πολύπλευρου εξοπλισμού. Συγκεκριμένα συνεργάζονται οι ακόλουθες οντότητες: • Κάμερες που βιντεοσκοπούν μία περιοχή με ιδιαίτερο ενδιαφέρον • Υπολογιστές που λαμβάνουν το σήμα της κάμερας και το επεξεργάζονται Το ζητούμενο είναι η εξαγωγή συμπεράσματος σε πραγματικό χρόνο για παράνομη παρουσία μη-εξουσιοδοτημένου προσωπικού σε ελεγχόμενο χώρο. Στη συνέχεια θα ενεργοποιηθεί άμεσα ένας μηχανισμός αντιμετώπισης περιστατικών. Η επεξεργασία του σήματος βίντεο στηρίζεται σε αλγόριθμους τεχνητής-όρασης. Το τελευταίο καιρό χρησιμοποιούνται με επιτυχία μέθοδοι μηχανικής-ευφυΐας που επιτρέπουν την εκπαίδευση μοντέλων για αναγνώριση πολύπλοκων αντικειμένων σε “πολυσύχναστους” χώρους.
Κόμβος TANDEM (TANDEM Node)	Κάθε υπολογιστική πλατφόρμα που μπορεί να φιλοξενήσει μία ή περισσότερες από τις μικροπηρεσίες (microservices) του TANDEM.
Πράγμα /Αντικείμενο (Thing/Object)	Κάθε συσκευή που διασυνδέεται με άλλες στο Διαδίκτυο των Αντικειμένων
Πύλη IoT (IoT Gateway)	Αποκαλείται η γέφυρα που διασυνδέει τα δεδομένα από τις διαφορετικές συσκευές με το υπολογιστικό νέφος και ενδεχομένως ενσωματώνει και κάποια υπολογιστική ευφυΐα για επεξεργασία των δεδομένων.
IoT Διεπαφή χρήστη (IoT User interface)	Πρόκειται για γραφικό περιβάλλον που προβάλλει στους χρήστες τα δεδομένα που συλλέχθηκαν από τις συσκευές και τους επιτρέπει να ζητήσουν ενέργειες προς εκτέλεση από τις συσκευές
Πακέτο Υπηρεσίας (Service Package)	Αποτελείται από (α) έναν Περιγραφέα Ανάπτυξης (Deployment Descriptor) που είναι ένα δομημένο αρχείο κειμένου με όλες τις πληροφορίες που χρειάζονται για τη δημιουργία στιγμιότυπου της υπηρεσίας στην πλατφόρμα του TANDEM, συμπεριλαμβανομένου ενός περιγραφέα εφαρμογής (κανόνες

	της εφαρμογής και απαιτήσεις από την πλατφόρμα TANDEM), πληροφοριών δήλωσης (manifest) και του URI της Εικόνας Λογισμικού (Software Image) και (β) της εικόνας λογισμικού (software image) εναλλακτικά της χρήσης του URI στο deployment descriptor.
Προϊόν (product)	Η υπηρεσία ή ο πόρος (υπολογιστικός, δικτυακός ή αποθήκευσης) με την προσθήκη των επιχειρηματικών πτυχών που σχετίζονται με το πως η υπηρεσία ή ο πόρος προσφέρεται στον πελάτη. Έτσι μια (τεχνική) υπηρεσία ή ένας πόρος μετασχηματίζεται σε εμπορεύσιμη υπηρεσία ή πόρο (δηλαδή προϊόν) με την προσθήκη στοιχείων όπως τιμολόγησης, άδειας χρήσης και Service Level Agreement (SLA).
Συνεργατική ή Ομοσπονδιακή Μάθηση (Federated Learning).	Είναι ένα σχήμα Μηχανικής Μάθησης στο οποίο ένα κοινό μοντέλο μπορεί να εκπαιδευτεί από κοινού από έναν αριθμό κατανεμημένων κόμβων χρησιμοποιώντας τα τοπικά αποθηκευμένα δεδομένα. Μπορεί να παρέχει καλύτερο απόρρητο δεδομένων, επειδή τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση δεν μεταφέρονται σε κεντρικό διακομιστή. Η ομοσπονδιακή εκμάθηση είναι κατάλληλη για εφαρμογές υπολογιστικών άκρων και μπορεί να αξιοποιήσει την υπολογιστική ισχύ των διακομιστών στα άκρα και τα δεδομένα που συλλέγονται σε ευρέως διασκορπισμένες συσκευές στα άκρα
Τεχνητή Νοημοσύνη (Artificial Intelligence)	Πρόκειται για αλγόριθμους που προσφέρουν μία μορφή ευφυΐας σε μηχανές. Συγκεκριμένα ένα συγκεκριμένο πρόβλημα διατυπώνεται σε ένα πλαίσιο που επιτρέπουν σε ένα υπολογιστή να εκπαιδευτεί στην επίλυση του και στη συνέχεια να αξιολογηθεί. Όταν οι επιδόσεις του κριθούν ικανοποιητικές μπορεί να αξιοποιηθεί σε ένα πραγματικό περιβάλλον - διαφορετικό από αυτό που έγινε η εκπαίδευση του. Η έννοια της Τεχνητής-Νοημοσύνης (AI) είναι αρκετά γενική και αποτέλεσε τη βάση για πιο συγκεκριμένες εφαρμογές, όπως: • Πρόβλεψη (Forecasting) • Εντοπισμός Ανωμαλιών (Anomaly Detection) • Κατανόηση φυσικής ομιλίας (Natural Language Processing) •
Τεχνητή Όραση (Machine-Vision)	Υπολογιστικό σύστημα που έχει ως στόχο τη κατανόηση του πραγματικού κόσμου μέσα από την επεξεργασία διαδοχικών εικόνων που παράγουν οι κάμερες. Χρησιμοποιεί Επεξεργασία Εικόνας (Image Processing) και Τεχνητή Νοημοσύνη (Artificial Intelligence) . Η Αναγνώριση Αντικειμένων (Object Recognition) είναι μία από τις λειτουργίες της Τεχνητής Όρασης.
Υπηρεσίες (Services)	Παραπομπή στην έννοια Δομο-ενότητα (function)
Υποστηρικτικές Υπηρεσίες (Supporting Services)	Οι υπηρεσίες που απαιτούνται για τη λειτουργία Υπηρεσιών Εφαρμογής ή στη δημιουργία Εφαρμογών (Applications), για παράδειγμα πρέπει να έχουμε την βοηθητική υπηρεσία πρόσβασης σε δεδομένα από εξωτερικές συσκευές, για να μπορέσουμε να χρησιμοποιήσουμε την υπηρεσία ανάλυσης αυτών των δεδομένων.
Υπηρεσίες Εφαρμογής (Application Services)	Οι βασικές υπηρεσίες προς τον πελάτη σε αντιδιαστολή με τις υποστηρικτικές υπηρεσίες. Είναι οι υπηρεσίες που θέλει να χρησιμοποιήσει ο πελάτης αλλά πολλές φορές μπορεί να χρειάζεται να συνδυαστούν με υποστηρικτικές υπηρεσίες για να προσφερθούν και να αποτελέσουν Εφαρμογή (Application) προς τον πελάτη.
Υπηρεσίες Εφαρμογής	Υπηρεσίες που εγκαθιστά ο Τελικός Χρήστης στις Υποδομές Τρίτου (π.χ. στις υποδομές του παρόχου του TANDEM). Οι υπηρεσίες αυτές μπορούν να

Χρήστη (User Application Services)	επικοινωνήσουν με τις άλλες υπηρεσίες της υποδομής για να χρησιμοποιούν λειτουργίες τους ή μπορούν να προσφέρουν λειτουργίες.
Υπολογιστική στα Άκρα του Δικτύου (Edge Computing)	Κατανεμημένη υπολογιστική αρχιτεκτονική που φέρνει την επεξεργασία και την αποθήκευση δεδομένων πιο κοντά στην τοποθεσία όπου απαιτείται, για τη βελτίωση των χρόνων απόκρισης και την εξοικονόμηση εύρους ζώνης
Υπολογιστικά Νέφη στα Άκρα του Δικτύου (Edge Clouds)	Ένα σύστημα cloud που περιλαμβάνει πόρους αποθήκευσης και επεξεργασίας που βρίσκονται στα άκρα του δικτύου, διασυνδέονται μεταξύ τους, και μπορούν να επεκταθούν και να προσαρμοστούν άμεσα σε μεταβαλλόμενες ανάγκες. Ένα Edge Cloud μπορεί να συνδέεται με την περιοχή που εξυπηρετεί και οι πόροι του να διαχειρίζονται από έναν Διαχειριστή Εικονικοποιημένης Υποδομής

Μεταφράσεις Βασικών Όρων

Όρος στα Αγγλικά	Όρος στα Ελληνικά
Agent	Πράκτορας
Alert	Συναγερμός
Bandwidth	Εύρος Ζώνης
Backend	Η πλευρά του εξυπηρετητή
Cloud Computing	Υπολογιστική Νέφους, Νεφοϋπολογιστική,
Cluster	Συστάδα
Edge	Άκρο, Παρυφή
Edge Computing	Υπολογιστική Άκρων, Υπολογιστική στα Άκρα του Δικτύου, Παρυφοϋπολογιστική
Descriptor	Περιγραφέας
Federeted Learning	Συνεργατική ή Ομοσπονδιακή Μάθηση
Feedback	Ανατροφοδότηση
Interface	Διεπαφή
Instance	Στιγμιότυπο, παρουσία
Metadata	Μεταδεδομένα
Load Balancing	Εξισορρόπηση Φορτίου
Microservices	Μικροϋπηρεσίες
Notification	Ειδοποίηση
Orchestrator	Ενορχηστρωτής
Portal	Πύλη
Process	Διεργασία
Provider	Πάροχος
Scaling	Κλιμάκωση, Επέκταση
Scalability	Επεκτασιμότητα
Server	Εξυπηρετητής, Διακομιστής
Service	Υπηρεσία
Scheduling	Χρονοπρογραμματισμός
Timestamp	Χρονοσφραγίδα
Virtual Machine	Εικονική Μηχανή
Virtualized	Εικονικοποιημένη
Virtualization	Εικονικοποίηση
Visualization	Οπτικοποίηση

Σύνοψη Παραδοτέου

Το παραδοτέο αυτό περιγράφει τις πτυχές της 2^{ης} τεχνικής υλοποίησης της πλατφόρμας του TANDEM και συνοδεύεται από την τεκμηρίωση λογισμικού, τις οδηγίες εγκατάστασης και τις οδηγίες χρήσης

Το έργο TANDEM έχει ως στόχο τη διαμόρφωση μιας ολοκληρωμένης λύσης για την υποστήριξη νέων μοντέλων υπολογιστικού νέφους στις παρυφές του δικτύου που θα επιτρέπει στους πελάτες - χρήστες του TANDEM να αναπτύσσουν, να εκτελούν και να διαχειρίζονται υπηρεσίες χωρίς το κόστος και την πολυπλοκότητα της οικοδόμησης και της συντήρησης της υποδομής ενώ παράλληλα μπορούν να αξιοποιούν και τις παρεχόμενες υπηρεσίες του TANDEM. Η ανάπτυξη και η λειτουργία των εφαρμογών φιλοξενείται σε υποδομές στις παρυφές των δικτύων.

Η λύση του TANDEM, θα δίνει τη δυνατότητα στους πελάτες να:

- Χρησιμοποιήσουν υπάρχουσες στοιχειώδεις υπηρεσίες του TANDEM (δομο-ενότητες ή λειτουργίες) οι οποίες ενεργοποιούνται μόνο μετά από την κλήση του πελάτη και μπορεί να χρεώνονται με βάσει το χρόνο εκτέλεσης
- Χρησιμοποιήσουν υπάρχουσες υπηρεσίες και εφαρμογές που έχουν τουλάχιστον μία διεργασία διακομιστή (στιγμιότυπο) που απαντά εξωτερικά αιτήματα πελατών με την κλιμάκωση να επιτυγχάνεται με την εκκίνηση περισσότερων διεργασιών διακομιστή.
- Εγκαταστήσουν μία δική τους υπηρεσία στην πλατφόρμα του TANDEM η οποία αξιοποιεί τους πόρους και τις συνδεδεμένες συσκευές
- Χρησιμοποιήσουν διαθέσιμες υπηρεσίες της πλατφόρμας, να τις συνδυάζουν χρησιμοποιώντας το γραφικό εργαλείο του TANDEM και να δημιουργούν σύνθετες αλυσίδες υπηρεσιών/λειτουργιών
- Αναπτύξουν εφαρμογές στο Web ή στα κινητά καλώντας τη Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface – API) υπηρεσιών του TANDEM ή υπηρεσιών που φιλοξενούνται στην πλατφόρμα του TANDEM

υποστηρίζοντας με αυτό τον τρόπο νέα **Function-as-a-Service (FaaS)** , **Platform-as-a-Service (Paas)** και **Backend-as-a-Service (Baas)** μοντέλων υπολογιστικού νέφους στις παρυφές του δικτύου.

Το έργο επικεντρώνεται στο σχεδιασμό, υλοποίηση και αξιολόγηση μιας ολοκληρωμένης πλατφόρμας που θα επιτυγχάνει τη δραστική απλοποίηση των διαδικασιών ανάπτυξης, διαχείρισης, επόπτευσης και επέκτασης υπηρεσιών σε περιβάλλοντα εγγενώς κατανεμημένα. Κεντρικό ρόλο κατέχουν (i) η δυνατότητα ενσωμάτωσης προϋπάρχοντων και προσαρμοζόμενων δομο-ενοτήτων/λειτουργιών σε ευρύτερες, σύνθετες αλυσίδες λειτουργίας (function chains), και (ii) η απλοποιημένη διασύνδεσή τους με τις τοπικές, σε επίπεδο κόμβου άκρων, συσκευές/πηγές δεδομένων, αλλά και με πιθανόν απομακρυσμένες και κεντριοποιημένες δομο-ενότητες π.χ., backends εφαρμογών στο κεντριοποιημένο υπολογιστικό νέφος.

Οι δυνατότητες αυτές έχουν ως στόχο τυπικά περιβάλλοντα υπολογιστικών νεφών στις παρυφές του δικτύου, με έμφαση στο Διαδίκτυο των Αντικειμένων, όπου η διασύνδεση και διαχείριση συσκευών αλλά και η υποστήριξη συγκεκριμένων τύπων επεξεργασίας δεδομένων, αποτελούν συχνή πρόκληση. Τυπικά παραδείγματα επεξεργασίας αποτελούν οι αναλύσεις δεδομένων μετρήσεων π.χ., μέσοι όροι τιμών, παρεκκλίνουσες τιμές, καθώς και ο εντοπισμός αντικειμένων σε ροές video π.χ., παρουσία ανθρώπων, αυτοκινήτων, κ.λπ.

Σκοπός του έργου είναι να δώσει ώθηση στην ανάπτυξη αντίστοιχων υπηρεσιών μέσω της διευκόλυνσης των παρόχων, τόσο στη τοπική διαχείριση και διασύνδεση τοπικών συσκευών και υπολογιστικών πόρων, όσο και στην εύκολη επαναχρησιμοποίηση λογισμικού για συγκεκριμένες, μεμονωμένες λειτουργίες, με άμεσο αντίκτυπο στην ευελιξία και ταχύτητα ανάπτυξης των επιθυμητών υπηρεσιών. Το έργο θα αναπτύξει την προτεινόμενη πλατφόρμα σχεδιάζοντας, υλοποιώντας και ενοποιώντας σημαντικές επεκτάσεις σε διαθέσιμες πλατφόρμες λογισμικού, ήτοι **Kubernetes [K8S]** και **EdgeXFoundry[EdgeX]**. Στοχεύοντας ένα πλήρως λειτουργικό πρωτότυπο σύστημα, το έργο θα προβεί στην υλοποίηση σημαντικών δομο-ενοτήτων επικεντρωμένων στην ανάλυση δεδομένων, τόσο από αισθητήρες, όσο και από κάμερες και στη διαχείριση και δρομολόγηση δεδομένων (ουρές δεδομένων). Τα δεδομένα από αισθητήρες θα αναλυθούν από διαφορετικές οπτικές γωνίας με χρήση **Αναλυτικής Επεξεργασίας (Analytical Processing)** καθώς και με χρήση τεχνολογιών **Μηχανικής Μάθησης (Machine Learning – ML)** και **Τεχνητής Νοημοσύνης (Artificial Intelligence – AI)** για την πρόβλεψη μελλοντικών τιμών σε μετρήσεις από αισθητήρες (Time Series Forecasting). Με τεχνολογίες Μηχανικής Μάθησης θα αναλυθούν και τα δεδομένα από τις κάμερες (οι ροές βίντεο) για την **Αναγνώριση Αντικειμένων (Object Recognition)**

Η αρχιτεκτονική του TANDEM επιτυγχάνει τη δραστική απλοποίηση των διαδικασιών ανάπτυξης, διαχείρισης, επόπτευσης και επέκτασης υπηρεσιών σε εικονικοποιημένες υποδομές, που βρίσκονται στις παρυφές του δικτύου.

1 Εισαγωγή

1.1 Το έργο TANDEM

Το έργο έχει ως στόχο τη διαμόρφωση μιας ολοκληρωμένης λύσης για την υποστήριξη νέων **BaaS/FaaS/PaaS** μοντέλων υπολογιστικού νέφους, στα άκρα του δικτύου. Το έργο θα επικεντρωθεί στο σχεδιασμό, υλοποίηση και αξιολόγηση μιας ολοκληρωμένης πλατφόρμας που θα επιτυγχάνει τη δραστική απλοποίηση των διαδικασιών ανάπτυξης, διαχείρισης, επόπτευσης και επέκτασης υπηρεσιών σε περιβάλλοντα εγγενώς κατανεμημένα.

Κεντρικό ρόλο κατέχουν:

- (i) η δυνατότητα ενσωμάτωσης προϋπάρχοντων και προσαρμοζόμενων δομο-ενοτήτων λογισμικού σε ευρύτερες, σύνθετες αλυσίδες λειτουργίας (function chains), και
- (ii) η απλοποιημένη διασύνδεσή τους με τις τοπικές, σε επίπεδο κόμβου άκρων, συσκευές και πηγές δεδομένων.

Στόχος είναι τυπικά περιβάλλοντα υπολογιστικών νεφών στις παρυφές του δικτύου, με έμφαση στο Διαδίκτυο των Αντικειμένων, όπου η διασύνδεση και διαχείριση συσκευών αλλά και η υποστήριξη συγκεκριμένων τύπων επεξεργασίας δεδομένων, αποτελούν συχνή πρόκληση.

Τυπικά παραδείγματα επεξεργασίας αποτελούν η ανάλυση δεδομένων μετρήσεων π.χ., μέσοι όροι τιμών, παρεκκλίνουσες τιμές, καθώς και η χρήση τεχνολογιών **Μηχανικής Μάθησης (Machine Learning – ML)** και **Τεχνητής Νοημοσύνης (Artificial Intelligence – AI)** για την πρόβλεψη μελλοντικών τιμών σε μετρήσεις από αισθητήρες (Time Series Forecasting) καθώς και ο εντοπισμός αντικειμένων σε ροές video π.χ., παρουσία ανθρώπων, αυτοκινήτων, όπλων, κ.λπ. Σκοπός του έργου είναι να δώσει ώθηση στην ανάπτυξη αντίστοιχων υπηρεσιών μέσω της διευκόλυνσης των παρόχων, τόσο στην τοπική διαχείριση και διασύνδεση τοπικών συσκευών και υπολογιστικών πόρων, όσο και στην εύκολη επαναχρησιμοποίηση λογισμικού για συγκεκριμένες, μεμονωμένες λειτουργίες, με άμεσο αντίκτυπο στην ευελιξία και ταχύτητα ανάπτυξης των επιθυμητών υπηρεσιών.

Η πλατφόρμα TANDEM βασίζεται σε δύο πυλώνες:

Κόμβοι άκρων

Παρέχουν υπολογιστικούς πόρους, ο διαμοιρασμός των οποίων βασίζεται στην τεχνολογία εικονικοποίησης **Docker Containers [Docker]**. Έμφαση θα δοθεί στην υποστήριξη διαφόρων τεχνολογιών διασύνδεσης τοπικών συσκευών π.χ., μέσω Bluetooth, Zigbee, LoRa, WiFi.

Επιδιωκόμενες λειτουργίες:

- Αυτόματη ανακάλυψη και ενσωμάτωση τοπικών συσκευών, όπως αισθητήρες, κάμερες, κ.λπ..
- Τοπική διαχείριση υπηρεσιών και αντίστοιχων πόρων. Περιλαμβάνει μεταξύ άλλων την αρχικοποίηση, κατάργηση, τροποποίηση, διασύνδεση και επόπτευση δομο-ενοτήτων λογισμικού, και αντίστοιχων πόρων.

Πλατφόρμα διαχείρισης

Περιλαμβάνει διαδικασίες για τη διαμόρφωση, διαχείριση, επόπτευση και επέκταση υπηρεσιών

Επιδιωκόμενες λειτουργίες:

1. **Λειτουργίες διαχείρισης και ενορχήστρωσης υπολογιστικού περιβάλλοντος** που περιλαμβάνουν λειτουργίες διαχείρισης κύκλου ζωής υπηρεσιών, τηλεμετρίας, διαχείρισης ευρωστίας και ελαστικότητας/κλιμακωσιμότητας, παροχής (γραφικών) διεπαφών σε επίπεδο διαχειριστή πόρων ή υπηρεσιών, αυτοματοποιημένης αναπαραγωγής ή διαφοροποίησης υπηρεσιών στους διαθέσιμους κόμβους, διασύνδεσης υπηρεσιών τόσο εντός του ίδιου κόμβου όσο και μεταξύ κόμβων, κ.λπ.
2. **Λειτουργίες διαχείρισης και ενορχήστρωσης δομο-ενοτήτων**, που περιλαμβάνουν:
 - **Κατάλογο δομο-ενοτήτων/υπηρεσιών (function/service catalogue).** Περιλαμβάνει λειτουργίες για τη διάθεση, αναζήτηση, διαχείριση και προβολή δομο-ενοτήτων σε επίπεδο container images. Το έργο θα υλοποιήσει συγκεκριμένες δομο-ενοότητες με σκοπό: (i) την ολοκληρωμένη υλοποίηση και αξιολόγηση της πλατφόρμας σε ρεαλιστικές περιπτώσεις χρήσης, (ii) την αρχικοποίηση καταλόγου που θα διατεθεί στην κοινότητα για χρήση και επέκταση. Οι δομο-ενοότητες αυτές θα είναι: (1) Επεξεργασία ροών δεδομένων του Διαδικτύου των Αντικειμένων, π.χ., υπολογισμός μέσων όρων, alerts, κ.λ.π., και πρόβλεψη μελλοντικών τιμών (2) Ανάλυση ροών video, για την αναγνώριση αντικειμένων, (3) Ουρές μηνυμάτων, περιλαμβάνοντας ευρέως διαδεδομένες λύσεις ανοιχτού-κώδικα π.χ. Kafka, κ.λ.π.
 - **Προσαρμογή δομο-ενοτήτων** στις ανάγκες της εκάστοτε υπηρεσίας, μέσω αντίστοιχων διεπαφών.
 - **Ενορχήστρωση δομο-ενοτήτων**, που θα βασίζεται (i) στην ανάπτυξη ενός πληροφοριακού μοντέλου για τη δομημένη περιγραφή των παρεχόμενων λειτουργιών καθώς και το σαφή ορισμό των διεπαφών, (ii) στην υποστήριξη γραφικού περιβάλλοντος για την τελική σύνθεση αλυσίδων λειτουργίας από τους διαχειριστές υπηρεσιών, (iii) στον έλεγχο και επικύρωση συμβατότητας των εμπλεκόμενων δομο-ενοτήτων σύνθετων υπηρεσιών. Η σύνθεση των δομοενοτήτων σε αλυσίδες, θα υποστηρίζει και το επίπεδο δικτύου και το επίπεδο εφαρμογής καθώς και τη διαχείριση του κύκλου ζωής τόσο των αρχικοποιημένων δομο-ενοτήτων όσο και των διαμορφωμένων αλυσίδων.

Υποστηρίζονται δύο σενάρια λειτουργίας:

1. Δεν είναι απαραίτητα να υπάρχουν εκτελούμενες διεργασίες για όλες τις υπηρεσίες αλλά αυτές δημιουργούνται μετά από αίτημα εφαρμογής πελάτη. Οι χρήστες μπορεί να πληρώνουν μόνο για το χρόνο εκτέλεσης των λειτουργιών επιτυγχάνοντας έτσι χαμηλότερο κόστος και υπάρχει δυναμικά προσαρμοζόμενη επεκτασιμότητα, με το κόστος της καθυστέρησης κάθε φορά για την ενεργοποίηση της αντίστοιχης διεργασίας. Γι' αυτό οι υπηρεσίες FaaS πρέπει να είναι στοιχειώδεις και σχεδιασμένες για να κάνουν μόνο μία ενέργεια σε απάντηση μιας αίτησης. Ο σκοπός του κώδικα πρέπει να είναι περιορισμένος, και ο κώδικας αποδοτικός και ελαφρύς (lightweight), ώστε οι λειτουργίες να φορτώνονται και να εκτελούνται γρήγορα (μοντέλο **Function-as-a-Service-FaaS**, η **Λειτουργία ως Υπηρεσία**)
2. Οι υπηρεσίες φιλοξενίας εφαρμογών έχουν τουλάχιστον μία διεργασία διακομιστή (στιγμιότυπο) που απαντά σε εξωτερικά αιτήματα πελατών. Η κλιμάκωση επιτυγχάνεται με την εκκίνηση περισσότερων διεργασιών διακομιστή, για τις οποίες

μπορεί να χρεώνεται άμεσα ο πελάτης (μοντέλο **Platform as a Service-PaaS**, η **Πλατφόρμα ως Υπηρεσία**).

Επιπλέον οι πελάτες θα μπορούν να αναπτύξουν εφαρμογές στο Web ή στα κινητά καλώντας το API υπηρεσιών του TANDEM ή υπηρεσιών που φιλοξενούνται στην πλατφόρμα του TANDEM. Ο πελάτης θα μπορούσε να πληρώσει με βάση τον αριθμό των κλήσεων API (μοντέλο **Backend as a Service-PaaS**, το **Backend ως Υπηρεσία**).

1.2 Αντικείμενο Παραδοτέου

Ο στόχος αυτού του παραδοτέου είναι να παρουσιάσει τις πτυχές της τεχνικής υλοποίησης της δεύτερης λειτουργικής έκδοσης του TANDEM που αναδεικνύει την φιλοσοφία λειτουργίας του. Συνοδεύεται από την τεκμηρίωση του λογισμικού, τις οδηγίες εγκατάστασης καθώς και οδηγίες χρήσης για τον Τελικό Χρήστη. Το λογισμικό μαζί με τα συνοδευτικά έγγραφα έχει αναρτηθεί στο αποθετήριο λογισμικού Github (<https://github.com/tandem-project/TANDEM-project>).

Η εργασία που παρουσιάζεται στο συγκεκριμένο παραδοτέο πραγματοποιείται στο πλαίσιο των εργασιών της **Ενότητας Εργασίας ΕΕ2 (Λεπτομερής Σχεδιασμός και Ανάπτυξη Συστήματος)** και συνδέεται στενά με τα παραδοτέα Π1.1, Π1.2 που αναλύουν τις απαιτήσεις και την αρχιτεκτονική του συστήματος TANDEM.

Η δεύτερη έκδοση του λογισμικού χρησιμοποιεί την ανατροφοδότηση από την πρώτη έκδοση (προβλήματα, σχόλια, παρατηρήσεις) και βασίζεται στην τελική αρχιτεκτονική του συστήματος (Παραδοτέο Π1.2.)

Τα προβλήματα που θα εντοπιστούν καθώς και τα σχόλια και οι παρατηρήσεις στην δεύτερη έκδοση του συστήματος από τις προϊόντικές διευθύνσεις, θα χρησιμοποιηθούν για τη βελτίωση του στην τρίτη.

1.3 Δομή Παραδοτέου

Το παραδοτέο Π2.2 περιγράφει τον δεύτερο κύκλο υλοποίησης της αρχιτεκτονικής του TANDEM που στόχο έχει τη δραστική απλοποίηση των διαδικασιών ανάπτυξης, διαχείρισης, επόπτευσης και επέκτασης υπηρεσιών σε εικονικοποιημένες υποδομές, που βρίσκονται στις παρυφές του δικτύου. Η δομή του παραδοτέου είναι η εξής:

Στην Ενότητα 1 παρουσιάζεται μια εισαγωγή στο έργο TANDEM και στο αντικείμενο του παραδοτέου καθώς και η βασική ορολογία.

Στην Ενότητα 2 παρουσιάζονται τα βασικά στοιχεία της 2^{ης} υλοποίησης του συστήματος TANDEM.

Στην Ενότητα 3 παρουσιάζεται η τεκμηρίωση της 2^{ης} υλοποίησης του Λογισμικού Κόμβων Άκρων του συστήματος TANDEM.

Στην Ενότητα 4 παρουσιάζεται η τεκμηρίωση της 2^{ης} υλοποίησης του Διαχειριστή Συστήματος TANDEM.

Στην Ενότητα 5 παρουσιάζονται οι οδηγίες εγκατάστασης του συστήματος TANDEM, τόσο του λογισμικού των Κόμβων Άκρων όσο και του Λογισμικού του Διαχειριστή Συστήματος TANDEM. Για το Kubernetes οι οδηγίες αφορούν την έκδοση MicroK8s

Η Ενότητα 6 αποτελεί έναν αναλυτικό Οδηγό Χρήσης του TANDEM Portal στην 2^η έκδοσή του.

Η Ενότητα 7 περιγράφει την πειραματική υλοποίηση Αλυσίδων Υπηρεσιών TANDEM.

Η Ενότητα 8 περιγράφει πειραματική εφαρμογή που υλοποιήθηκε με χρήση των υπηρεσιών TANDEM.

Η Ενότητα 9 παρέχει μια σύνοψη του παραδοτέου και αναφορά στα πλάνα για τις λειτουργίες που θα παρέχουν οι δύο επόμενες εκδόσεις του λογισμικού

Το Παράρτημα περιέχει αναλυτικές οδηγίες εγκατάστασης ξεκινώντας από την εγκατάσταση της εικονικοποιημένης υποδομής (δημιουργία VMs) στην περίπτωση που αυτό απαιτείται, στην εγκατάσταση του Kubernetes (K8s) και στην εγκατάσταση των υπηρεσιών Τεχνητής Νοημοσύνης

2 2^η Έκδοση του Λογισμικού Συστήματος TANDEM

Σε αυτή την Ενότητα παρουσιάζεται η δεύτερη έκδοση του Λογισμικού Συστήματος TANDEM.

Το έργο μέσω επαναλαμβανόμενων κύκλων σχεδιασμού, υλοποίησης, ελέγχου, θα παραδώσει συνολικά 4 εκδόσεις του λογισμικού με διαφορά τεσσάρων μηνών. Κάθε έκδοση του λογισμικού θα έχει σε σχέση με την προηγούμενη επιπλέον λειτουργίες και διορθώσεις/βελτιώσεις στις υπάρχουσες λειτουργίες. Στους κύκλους αυτούς συμμετέχουν και οι διαδικασίες ολοκλήρωσης και γενικού ελέγχου συστήματος (Δ3.1) της Ενότητας Εργασίας 3 (ΕΕ3).

Η τελική έκδοση του λογισμικού (4^η έκδοση) θα πρέπει να καλύπτει όλες τις προδιαγραφές συστήματος και είναι και η έκδοση που θα αξιολογηθεί εκτεταμένα στην τελευταία φάση του έργου στα πλαίσια της ΕΕ3 η οποία θα παραδώσει και την τελική αξιολόγηση της πλατφόρμας.

Η περιγραφή του Λογισμικού γίνεται βάσει των δύο πυλώνων του TANDEM:

1. Λογισμικό για τους **Κόμβους Άκρων** οι οποίοι παρέχουν υπολογιστικούς πόρους, ο διαμοιρασμός των οποίων βασίζεται στην τεχνολογία εικονικοποίησης Docker Containers. Το λογισμικό αυτό διαχειρίζεται τις υποδομές, τους πόρους και τις υπηρεσίες των κόμβων και παρέχει υπηρεσίες χρήσιμες στα άκρα (π.χ. υπηρεσίες για τη διαχείριση των IoT συσκευών)
2. Λογισμικό για την **Πλατφόρμα Διαχείρισης** που περιλαμβάνει διαδικασίες για τη διαμόρφωση, διαχείριση, επόπτευση και επέκταση υπηρεσιών.

Οι εκδόσεις του λογισμικού βασίζονται σε **Docker Containers** και **Kubernetes** ως ενορχηστρωτή της εγκατάστασης, και υποστήριξης κλιμάκωσης (scaling) και διαχείρισης υπηρεσιών σε μορφή Container. Η έκδοση αυτή υποστηρίζει ένα νέφος άκρων (cluster). Για την υποστήριξη IoT λειτουργιών και εξασφάλιση της διαλειτουργικότητας μεταξύ συσκευών και υπηρεσιών ενσωματώνεται το **EdgeXFoundry**.

Η πρώτη έκδοση του λογισμικού περιλάμβανε:

- Αρχική Υλοποίηση του **Διαχειριστή Πλατφόρμας Άκρων (TANDEM Edge Platform Manager ή pi-Edge)** που διευκολύνει την διαχείριση των υποδομών άκρων. Προσφέρει αυτοματοποιημένη διαχείριση του κύκλου ζωής των υπηρεσιών στα άκρα του δικτύου και κάνει διαθέσιμο ένα πλήθος από υπηρεσίες προστιθέμενης αξίας και έξυπνους μηχανισμούς στα άκρα, για τη διασφάλιση της αποδοτικής εγκατάστασης και της εύρυθμης λειτουργίας των αλυσίδων υπηρεσιών. Αυτοί οι μηχανισμοί περιλαμβάνουν λειτουργίες όπως *αυτό-κλιμάκωση* και *αυτό-ίαση* υπηρεσιών αναλόγως του φόρτου εργασίας και του γενικού πλαισίου λειτουργίας τους, την αυτόματη διασύνδεση των κόμβων άκρων (Edge nodes) με συσκευές Διαδικτύου των Πραγμάτων (IoT) και επέκταση σε αυτές υπολογιστικού περιβάλλοντος για το εκτέλεση από άκρη σε άκρη IoT εφαρμογών.
- Αρχική Υλοποίηση του **TANDEM Portal**. Σε αυτή την έκδοση έχει υλοποιηθεί το User Interface και για λειτουργίες που δεν υποστηρίζονται στην πρώτη έκδοση ώστε ο χρήστης να έχει συνολική εικόνα για την πλατφόρμα του TANDEM.
- Αρχική Υλοποίηση του **Καταλόγου Υπηρεσιών (Service Catalogue)** με λειτουργίες όπως καταχώρηση καινούργιας υπηρεσίας, τροποποίηση υπηρεσίας, διαγραφή υπηρεσίας, αναζήτησης υπηρεσίας με κριτήρια, γραφική παρακολούθηση εκτέλεσης

υπηρεσιών, κλπ. Το μοντέλο υπηρεσίας του TANDEM επεκτείνει το μοντέλο του **ETSI MEC[MEC]** με στοιχεία από το αντίστοιχα μοντέλα του **TM Forum [TMF633]** και του **Open Service Catalogue Manager[OSCM]** με στόχο την επιπλέον κάλυψη επιχειρηματικών πτυχών (πώς η υπηρεσία προσφέρεται στον πελάτη).

- Προκαταρκτική Υλοποίηση μηχανισμών αυθεντικοποίησης και εξουσιοδότησης χρηστών που αποτελούν μέρους του **Διαχειριστή Χρηστών (User Manager)** με χρήση της πλατφόρμας **Keycloak [keycloak]**.
- Αρχική Υλοποίηση της **Παρακολούθησης Συστήματος (System Monitoring)** που βασίζεται σε **Prometheus[Prometheus]** και **Grafana [Grafana]**. Ο χρήστης μπορεί να παρακολουθεί γραφικά σε πραγματικό χρόνο την κατάσταση λειτουργίας ενός edge cloud, ενός κόμβου του edge cloud, των υπηρεσιών μιας σύνθετης υπηρεσίας ή εφαρμογής, ή μιας συγκεκριμένης υπηρεσίας.
- Αρχική Υλοποίηση του **Διαχειριστή Αλυσίδων Υπηρεσιών (Service Chain Manager)** που είναι μέρος του **Ενορχηστρωτή Αλυσίδων Υπηρεσιών (Service Chain Orchestrator)**. Περιλαμβάνει γραφικό διαδραστικό εργαλείο για τη διαχείριση αλυσίδων δομο-ενοτήτων (functions) και υπηρεσιών (services). Ο χρήστης μπορεί να δημιουργήσει αλυσίδες υπηρεσιών, να τις τροποποιήσει, να τις διαγράψει, να τις αναζητήσει με διάφορα κριτήρια, να τις ορίσει ως εφαρμογές. Η περιγραφή των αλυσίδων υπηρεσιών αποθηκεύονται σε ένα δομημένο αρχείο κειμένου(json) το οποίο συντηρείτε σε Βάση Δεδομένων μαζί με τα μεταδεδομένα της Αλυσίδας (όπως όνομα, περιγραφή, κλπ).
- Αρχική Υλοποίηση της Υπηρεσίας Τεχνητής Νοημοσύνης **Αναγνώριση Αντικειμένων (Object Detection)** και της Υποστηρικτικής της Υπηρεσίας **Διαχειριστής Ζωντανού Σήματος Βίντεο (Video Stream Manager)**. Η Υπηρεσία Αναγνώρισης Αντικειμένων πραγματοποιεί αυτόματη αναγνώριση σε εισερχόμενο σήμα βίντεο συγκεκριμένων αντικειμένων, όπως άνθρωποι, αυτοκίνητα, μηχανές, σακίδια κοκ. Η φιλοσοφία της είναι να λειτουργεί στο Edge ώστε το βίντεο να μη χρειάζεται να ταξιδεύει στο δίκτυο και να το επιβαρύνει. Η κύρια λειτουργία του Διαχειριστή Ζωντανού Σήματος Βίντεο είναι να συγκεντρώνει στιγμιότυπα εικόνων από ροές βίντεο.
- Αρχική Υλοποίηση της Βασικής Υπηρεσίας **Ανάλυση Δεδομένων από IoT Συσκευές (IoT Data Analytics)** η οποία χρησιμοποιεί διάφορες υπηρεσίες για να συγκεντρώσει και να επεξεργαστεί δεδομένα από IoT αισθητήρες και συσκευές ώστε τελικά να τα παρουσιάσει γραφικά σε διάφορες μορφές με χρήση του *Grafana*.
- Αρχική Υλοποίηση δύο Υπηρεσιών Εφαρμογών που χρησιμοποιούν την υπηρεσία Αναγνώρισης Αντικειμένων, την **Υπηρεσία Έξυπνης Στάθμευσης (Smart Parking Service)** η οποία αυτόματα εντοπίζει τις διαθέσιμες θέσεις στάθμευσης με χρήση καμερών και την **Υπηρεσία Επόπτευσης & Ασφάλειας Χώρου (Surveillance & Security Service)** η οποία αυτόματα εντοπίζει αντικείμενα (π.χ. ξεχασμένες τσάντες, αυτοκίνητα, κλπ) και ανθρώπους που δε θα έπρεπε να υπάρχουν σε επιτηρούμενο χώρο με χρήση καμερών. Επίσης υλοποίηση κεντρικής πειραματικής εφαρμογής **Παρακολούθησης Χώρου Στάθμευσης (Parking Application)** που ενέχει ένα ρόλο “συγκεντρωτή” (aggregator) όλων των πληροφοριών από τις κατανεμημένες Υπηρεσίες Έξυπνης Στάθμευσης που εκτελούνται στα άκρα.

Οι υπηρεσίες που υλοποιήθηκαν στα πλαίσια της πρώτης έκδοσης λογισμικού χρησιμοποιήθηκαν για τη δημιουργία δύο πρωτοτύπων αλυσίδων υπηρεσιών από άκρο σε άκρο από τη συλλογή δεδομένων από IoT devices μέχρι τη ανάλυση τους, τη γραφική αναπαράστασή τους και τις ενέργειες που λαμβάνουν χώρα ανάλογα με τις τιμές τους. Ο σκοπός τους είναι η εξέταση του τρόπου υλοποίησης και των τεχνικών λεπτομερειών και η επίδειξη της φιλοσοφίας του TANDEM. Οι δύο αλυσίδες υπηρεσιών περιγράφονται στην Ενότητα 7.

Επιγραμματικά, η 1^η έκδοση του λογισμικού περιλαμβάνει αρχικές εκδόσεις των:

- Διαχειριστή Πλατφόρμας Άκρων (TANDEM Edge Platform Manager ή pi-Edge)
- TANDEM Portal
- Καταλόγου Υπηρεσιών (Service Catalogue)
- Υπηρεσίας Παρακολούθησης Συστήματος (System Monitoring)
- Ενορχηστρωτή Αλυσίδων Υπηρεσιών (Service Chain Orchestrator). .
- Υπηρεσίας Τεχνητής Νοημοσύνης Αναγνώρισης Αντικειμένων (Object Detection service) και της υποστηρικτικής της Υπηρεσίας Διαχειριστής Ζωντανού Σήματος Βίντεο (Video Stream Manager).
- Βασικής Υπηρεσίας Ανάλυση Δεδομένων από IoT Συσκευές (IoT Data Analytics).
- Δύο Υπηρεσιών Εφαρμογών που χρησιμοποιούν την υπηρεσία Αναγνώρισης Αντικειμένων, την Υπηρεσία Έξυπνης Στάθμευσης (Smart Parking Service) και την Υπηρεσία Ασφαλείας (Security Service)

Τα προβλήματα που εντοπίστηκαν καθώς και τα σχόλια και οι παρατηρήσεις στην πρώτη έκδοση του συστήματος και από τις προϊοντικές διευθύνσεις, χρησιμοποιήθηκαν για τη βελτίωση του στην δεύτερη.

Σύμφωνα με τον προγραμματισμό, η δεύτερη έκδοση του λογισμικού που περιγράφεται σε αυτό το παραδοτέο περιλαμβάνει:

- Την ολοκλήρωση της υλοποίησης του TANDEM portal με την προσθήκη όλων των λειτουργιών που έχουν προδιαγραφεί.
- Τη Δεύτερη Έκδοση του **Διαχειριστή Πλατφόρμας Άκρων (TANDEM Edge Platform Manager ή pi-Edge)**, με διορθώσεις στη λειτουργία του και επιπλέον αυτοματισμούς καθώς και την προσθήκη και υποστήριξη υπηρεσιών **MEC πλατφόρμας** στην γενική αρχιτεκτονική, με την χρησιμοποίηση της στο IoT σενάριο χρήσης. Επιπλέον προστέθηκε η αυτοματοποιημένη παραγωγή και ανανέωση των σελιδών που οπτικοποιούν την χρήση των πορών κάθε κόμβου αλλά και κάθε υπηρεσίας που τρέχει σε αυτούς.
- Τη Δεύτερη Έκδοση του **Καταλόγου Υπηρεσιών (Service Catalogue)** και της **Υπηρεσίας Παρακολούθησης Συστήματος (System Monitoring Service)**
- Την πρώτη λειτουργική έκδοση της Υπηρεσίας **Διαχειριστή Χρηστών (User Manager)** που θα περιλαμβάνει αυστηρούς μηχανισμούς εξουσιοδότησης σύμφωνα και με το ρόλο του χρήστη στην πλατφόρμα του TANDEM.
- Ολοκλήρωση της επικοινωνίας του γραφικού περιβάλλοντος του **TANDEM Portal** με τους μηχανισμούς διαχείρισης της υποδομής (αρχικοποίηση κόμβου TANDEM, ενεργοποίηση Βασικών Υπηρεσιών σε αυτό, κλπ.).

Επιπλέον του αρχικού προγραμματισμού, η έκδοση αυτή περιλαμβάνει και το MEC service “Location APIs” (αρχικός προγραμματισμός για τις εκδόσεις 3 και 4) που είναι ένα σύνολο από διεπαφές, οι οποίες παρέχονται από το **MEC Platform** και προσφέρουν πληροφορίες που αφορούν την τοποθεσία διαφόρων User Equipments, τον υπολογισμό αποστάσεων, καθώς και την εύρεση του/των κοντινότερων UE(s).

Η MEC αυτή υπηρεσία, χρησιμοποιήθηκε πειραματικά στην πρωτότυπη αλυσίδα υπηρεσιών που υλοποιήθηκε ως μέρος της πρώτης έκδοσης και αφορά τη διαδικασία από τη συλλογή δεδομένων από IoT devices (αισθητήρες θερμοκρασίας και υγρασίας) μέχρι τη ανάλυσή τους, τη γραφική αναπαράστασή τους και τις ενέργειες που λαμβάνουν χώρα ανάλογα με τις τιμές τους.

3 Υλοποίηση Λογισμικού Κόμβων Άκρων (Παρυφών)

3.1 Διαχειριστής Πλατφόρμας Παρυφών του TANDEM (TANDEM Edge Platform Manager - pi-Edge)

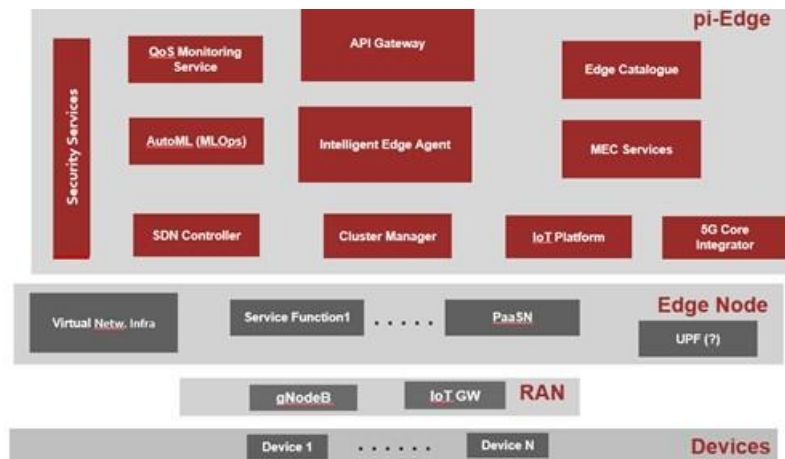
3.1.1 Εισαγωγή

Η πλατφόρμα pi-Edge υλοποιεί τη λειτουργικότητα του Διαχειριστή Πλατφόρμας Παρυφών του TANDEM και διευκολύνει την διαχείριση των υποδομών άκρων. Προσφέρει αυτοματοποιημένη διαχείριση του κύκλου ζωής των υπηρεσιών στα άκρα του δικτύου και κάνει διαθέσιμο ένα πλήθος από υπηρεσίες προστιθέμενης αξίας και έξυπνους μηχανισμούς στα άκρα, για τη διασφάλιση της αποδοτικής εγκατάστασης και της εύρυθμης λειτουργίας των αλυσίδων υπηρεσιών. Αυτοί οι μηχανισμοί περιλαμβάνουν λειτουργίες όπως *αυτό-κλιμάκωση* και *αυτό-ίαση* υπηρεσιών αναλόγως του φόρτου εργασίας και του γενικού πλαισίου λειτουργίας τους, την αυτόματη διασύνδεση των κόμβων άκρων (Edge nodes) με συσκευές Διαδικτύου των Πραγμάτων (IoT) και επέκταση σε αυτές υπολογιστικού περιβάλλοντος για το εκτέλεση από άκρη σε άκρη IoT εφαρμογών. Το κύρια κίνητρα πίσω από την δημιουργία της pi-Edge πλατφόρμας ήταν:

- Η διευκόλυνση της επικοινωνίας οποιαδήποτε NFV MANO πλατφόρμας με τις υποδομές των άκρων και τις υπηρεσίες που πρέπει να τρέχουν εκεί, χρησιμοποιώντας ένα PaaS μοντέλο που ενισχύει την αυτοματοποίηση, την διαλειτουργικότητα και την συντηρησιμότητα.
- Ελαχιστοποίηση των εξόδων διαχείρισης. Καθώς, η pi-Edge πλατφόρμα αναλαμβάνει να διαχειρίζεται τις υπηρεσίες που τρέχουν στα άκρα.
- Δεν χρειάζεται να συλλέγονται συνεχώς λεπτομερή και ακατέργαστα δεδομένα από όλες τις άκρες που αφορούν την κατάσταση των υπηρεσιών που τρέχουν εκεί, καθώς αυτά μπορούν να μελετώνται σε κάθε ακμή ξεχωριστά. Μέσω τις pi-Edge πλατφόρμας οι αποφάσεις διαχείρισης χαμηλού επιπέδου, μπορούν να ληφθούν τοπικά σε κάθε ακμή.
- Ελαχιστοποίηση την ταχύτητας εγκατάστασης κάθε υπηρεσίες καθώς και τόνωση της επαναχρησιμοποίησης εγκατεστημένων εφαρμογών. Σημαντικό ρόλο σε αυτό παίζουν οι υπηρεσίες που παρέχει η pi-Edge πλατφόρμα, προσφέροντας υπηρεσίες για εύκολο εντοπισμό και διαχείριση εγγεγραμμένων στον κατάλογο PaaS υπηρεσιών που μπορεί να αποτελούνται από 1 ή περισσότερα Service functions. Έτσι, ο κάθε χρήστης μπορεί να επαναχρησιμοποιήσει συνθέτες PaaS υπηρεσίες που έχουν δημιουργηθεί από άλλους χρήστες, δίχως να πρέπει να σπαταλήσει σημαντικό χρόνο για να τις δημιουργήσει.
- Παράλληλα, προσφέρεται ένα ανώτατου επιπέδου αφαιρετικό μοντέλο για την επεξήγηση και χρησιμοποίηση των PaaS υπηρεσιών αλλά και των Service Functions, διατηρώντας αρκετή ευελιξία και ευκολία στην χρήση.

3.1.2 Αρχιτεκτονική και Ανάλυση του Διαχειριστή Πλατφόρμας Παρυφών

Όπως απεικονίζεται στην Εικόνα 1: Αρχιτεκτονική υψηλού επιπέδου, η πλατφόρμα αποτελείται από μια σειρά από οντότητες (API Gateway, Intelligent Edge Agent, QoS Monitoring Service, Edge Catalogue, Cluster Manager, Security Services κ.), MEC Platform, όπου κάθε μία υλοποιεί μια ξεχωριστή λειτουργία. Στις επόμενες υπο-ενότητες αναλύονται οι σημαντικότερες οντότητες που χρησιμοποιούνται αυτή την στιγμή από το TANDEM για την υλοποίηση των σεναρίων που έχουν περιγραφεί.



Εικόνα 1: Αρχιτεκτονική υψηλού επιπέδου του Διαχειριστή Πλατφόρμας Άκρων

3.1.2.1 Edge Catalogue

Ο κατάλογος της πλατφόρμας περιέχει συλλογές των υποστηριζόμενων service functions, PaaS υπηρεσιών αλλά και πληροφορίες των κόμβων. Τα service functions αποτελούν την μικρότερη «μονάδα» του καταλόγου που μπορεί να εγκατασταθεί στο σύμπλεγμα των παρυφών του δικτύου. Συγκεκριμένα, οι εγγραφές service functions αποτελούνται από τα παρακάτω στοιχεία:

```
{
  "service_function_name": "string",
  "service_function_image": "string",
  "service_function_type": "string",
  "application_ports": [
    5601
  ],
  "autoscaling_policies": [
    {
      "policy": "string",
      "monitoring_metrics": [
        {
          "metric": "string",
          "limit": "string",
          "request": "string",
          "util_percent": Integer,
          "is_default": Boolean
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "required_volumes": [
    {
      "name": "string",
      "path": "string"
    }
  ],
  "required_env_parameters": [
    {
      "name": "string"
    }
  ]
]

```

όπου, περιέχουν πληροφορία για το online repository που υπάρχει το image αλλά και το όνομα του, τον τύπο του, τα application ports που περιγράφουν τις πόρτες την εφαρμογής αλλά και πληροφορίες σχετικά με το πότε πρέπει να εκτελείτε (αυτό-) κλιμάκωση, οι πόροι που απαιτούνται, καθώς και πληροφορίες με το αν απαιτούνται τομείς (volumes) και ποιες τιμές πρέπει να έχουν οι παράμετροι συστήματος (environment parameters/ variables).

Οι εγγραφές για τις PaaS υπηρεσίες αποτελούνται από τα παρακάτω στοιχεία:

```

{
  "paas_service_name": "string",
  "service_functions": [
    [
      {
        "service_function_identifier_name": "string",
        "volume_mounts": [
          {
            "storage": "string",
            "name": "string"
          }
        ],
        "autoscaling_metric": "cpu"
      },
      {
        "service_function_identifier_name": "string",
        "autoscaling_metric": "string",
        "env_parameters": [
          {
            "name": "string ",
            "value": "string"
          }
        ]
      }
    ]
  ]
}

```

Όπου, μια PaaS υπηρεσία μπορεί να αναφέρεται σε ένα συνδυασμό από service functions. Κάθε service function περιέχει το όνομα της, που χρησιμοποιείται ως το αναγνωριστικό (για την εύρεση του στον κατάλογο), καθώς και τις απαραίτητες πληροφορίες που χρειάζονται για την επιτυχή εγκατάσταση του (*volumes*: το όνομα αλλά και το μέγεθος του δίσκου που χρειάζονται, *env_parametes*: που είναι απαραίτητες για να εκτελεστεί το service, και *autoscaling_metric*: με τι μονάδα μέτρησης θα γίνονται οι (αυτό)-κλιμακώσεις).

Οι εγγραφές για τις πληροφορίες των κόμβων (υποδομές στις παρυφές του δικτύου) αποτελούνται από τα παρακάτω στοιχεία:

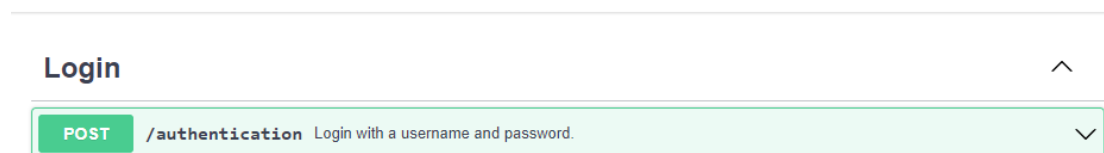
```
{
  "serial": "string",
  "name": "string",
  "location": "string",
  "id": "string"
}
```

όπου έχουμε την IP του μηχανήματος, την τοποθεσία του, αλλά και το όνομα του.

3.1.2.2 API Gateway & Intelligent Edge Agent

Παρακάτω παρατίθενται τα APIs που «προσφέρονται» από τον API Gateway της ri-Edge πλατφόρμας.

3.1.2.2.1 Μέθοδοι αυθεντικοποίησης



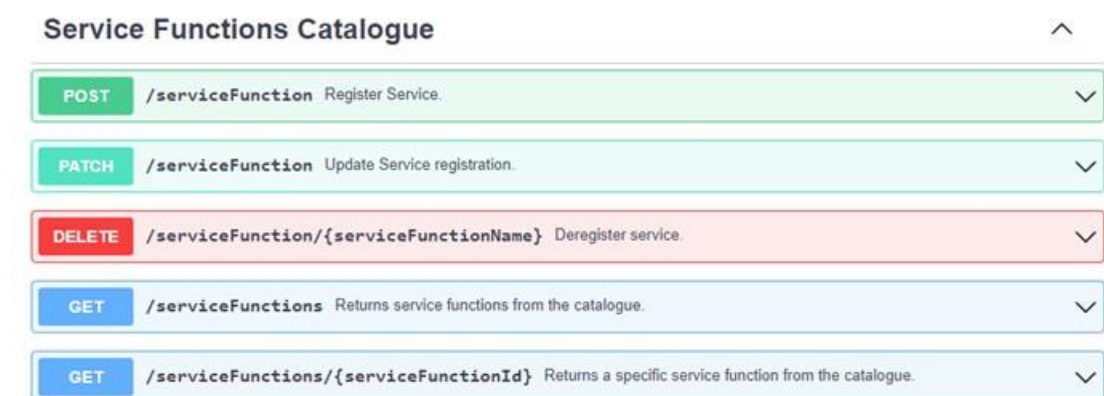
Εικόνα 2: Μέθοδος αυθεντικοποίησης TANDEM χρήστη

Χρησιμοποιώντας την παραπάνω μέθοδο, μπορεί ο χρήστης να αποκτήσει πιστοποίηση για χρήση της πλατφόρμας, χρησιμοποιώντας για σώμα το παρακάτω (παρατίθεται με παράδειγμα):

```
{
  "username": "admin",
  "password": "admin@ICOM_123!"
}
```

3.1.2.2.2 Μέθοδοι εγγραφής στον κατάλογο

Χρησιμοποιώντας τις παρακάτω μεθόδους, μπορεί ο χρήστης να διαχειριστεί τις εγγραφές στον κατάλογο της πλατφόρμας.



Εικόνα 3: Μέθοδοι διαχείρισης εγγραφών Service Function

Με την μέθοδο **POST /serviceFunction** μπορούμε να γράψουμε service function στον κατάλογο, χρησιμοποιώντας για σώμα το παρακάτω (παρατίθεται με παράδειγμα):

```
{
  "service_function_name": "Kibana",
  "service_function_image": "kibana:7.15.2",
  "service_function_type": "Container",
  "application_ports": [
    5601
  ],
  "autoscaling_policies": [
    {
      "policy": "maximize-performance",
      "monitoring_metrics": [
        {
          "metric": "cpu",
          "limit": "1000m",
          "request": "600m",
          "util_percent": 60,
          "is_default": true
        }
      ]
    }
  ],
  "required_volumes": [
    {
      "name": "volumeconfig",
      "path": "/data/config"
    }
  ],
  "required_env_parameters": [
    {
      "name": "ELASTICSEARCH_URL"
    }
  ]
}
```

παράλληλα υποστηρίζονται και λειτουργίες όπως εκσυγχρονισμός/ ενημέρωση του service function (**PATCH /serviceFunction**), διαγραφή του (**DELETE /serviceFunction/{serviceFunctionName}**) αλλά και απόκτηση πληροφοριών για τα/ το service function (**GET /serviceFunctions**, **GET /serviceFunctions/{serviceFunctionId}**).

PaaS Services Catalogue			^
POST	/paasService	Register PaaS.	▼
PATCH	/paasService	Update PaaS registration.	▼
DELETE	/paasService/{paasServiceName}	Deregister PaaS.	▼
GET	/paasServices	Returns PaaS Services from the catalogue.	▼
GET	/paasServices/{paasServiceId}	Returns a specific PaaS service from the catalogue.	▼

Εικόνα 4: Μέθοδοι διαχείρισης εγγραφών PaaS υπηρεσιών

Με την μέθοδο **POST /paasService** μπορούμε να γράψουμε PaaS στον κατάλογο, χρησιμοποιώντας για σώμα το παρακάτω (παρατίθεται με παράδειγμα):

```
{
  "paas_service_name": "Support-Security-Services",
  "service_functions": [
    {
      "service_function_identifier_name": "ElasticSearch",
      "volume_mounts": [
        {
          "storage": "500Mi",
          "name": "volume1"
        }
      ],
      "autoscaling_metric": "cpu"
    },
    {
      "service_function_identifier_name": "Kibana",
      "autoscaling_metric": "memory",
      "env_parameters": [
        {
          "name": "ESTASTICSEARCH_URL",
          "value": "http://Elasticname:..... INTKUBEDNS"
        }
      ]
    }
  ]
}
```

παράλληλα υποστηρίζονται και λειτουργίες όπως εκσυγχρονισμός/ ενημέρωση της PaaS υπηρεσίας (**PATCH /paasService**), διαγραφή της (**DELETE /paasService /{paasServiceName}**) αλλά και απόκτηση πληροφοριών για τις/ την PaaS υπηρεσία (**GET /paasServices, GET /paasServices/{paasServiceId}**).

Nodes		^
GET	/initNodes Initialize nodes. Scan nodes in the edge cluster.	✓ 🔒
POST	/activateTandemNode Request to create a "PaaS-enabled" node.	✓ 🔒
GET	/nodes Returns the edge nodes status.	✓ 🔒
GET	/nodes/{nodeName} Returns info of a specific edge node.	✓ 🔒

Εικόνα 5: Μέθοδος εύρεσης και εγγραφής κόμβων στον κατάλογο

Εκτελώντας την μέθοδο **GET /iniNodes** η ri-Edge πλατφόρμα αναζητάει τους κόμβους που είναι συνδεδεμένοι με τον κύριο κόμβο (master node) και αποθηκεύει τα παρακάτω στοιχεία τους στον κατάλογο (παρατίθεται με παράδειγμα):

```
{
  "serial": "146.124.106.179",
  "name": "compute1",
  "location": "Peania_19002_Athens",
  "id": "237d11c4-aca6-4845-9538-ba7b3e89c0b6"
}
```

Όπου μπορούμε να τα δούμε εκτελώντας α) την μέθοδο **GET /nodes** για να παρουμε γενικά τις πληροφορίες όλων των κόμβων ή β) **GET /nodes/{nodename}** για να παρουμε τις πληροφορίες του κόμβου της επιλογής μας. Οι πληροφορίες αυτές περιέχουν μετρήσεις και δείκτες για το τι PaaS υπηρεσίες τρέχουν στους κόμβους, τι πόρους απαιτούν, πόσοι πόροι είναι διαθέσιμοι συνολικά σε κάθε κόμβο κτλπ. Επίσης προσφέρουν URLs για την ζωντάνη παρακολούθηση των πόρων και των υπηρεσιών που τρέχουν, σε συνεργασία με το Edge Monitoring στοιχείο της πλατφόρμας που περιγράφεται παρακάτω.

3.1.2.2.3 Μέθοδοι εγκατάστασης PaaS υπηρεσιών

Χρησιμοποιώντας τις παρακάτω μεθόδους, μπορεί ο χρήστης να ζητήσει την εγκατάσταση PaaS υπηρεσίας στην παρυφή της επιλογής του, να δει την κατάσταση της δηλ. αν είναι σε “running” κατάσταση ή όχι, σε ποιο κόμβο έχει εγκατασταθεί κα.

PaaS Services Instances			^
POST	/deployPaas	Request to deploy a PaaS (from the catalogue) to an edge node.	▼
PATCH	/updateDeployedPaas	Request to update the status of a deployed PaaS.	▼
GET	/deployedPaas	Returns edge paas services status.	▼
GET	/deployedPaasServices/{deployedPaasServiceName}	Returns the requested edge paas service status.	▼
DELETE	/deployedPaas/{deployedPaasServiceName}	Deletes a deployed PaaS service.	▼

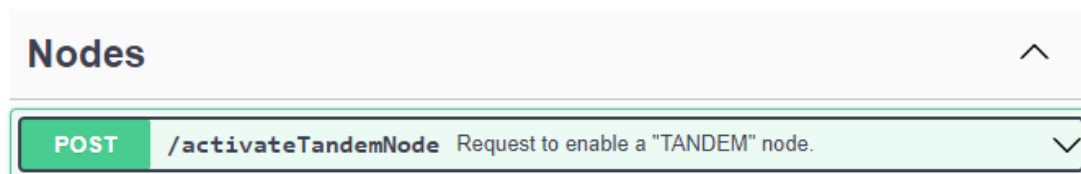
Εικόνα 6: Μέθοδος διαχείρισης και εγκατάστασης PaaS υπηρεσιών

Με την μέθοδο **POST /deployPaas** μπορούμε να ζητήσουμε από την ri-Edge πλατφόρμα να εγκαταστήσει μια PaaS υπηρεσία, χρησιμοποιώντας για σώμα το παρακάτω (παρατίθεται με παράδειγμα):

```
{
  "paas_service_name": "EdgeX",
  "paas_instance_name": "Edgex_ICOM_deployment",
  "autoscaling_type": "maximize_performance",
  "count_min": 1,
  "count_max": 5,
  "location": "Peania_Athens_node1",
  "all_node_ports": false,
  "node_ports": [
    90,
    8080
  ]
}
```

παράλληλα υποστηρίζονται και λειτουργίες όπως εκσυγχρονισμός/ ενημέρωση της εγκαταστημένης PaaS υπηρεσίας (**PATCH /updateDeployedPaas**), διαγραφή της (**DELETE /deoloyedPaas/{deployedPaasServiceName}**) αλλά και απόκτηση πληροφοριών για τις/ την εγκαταστημένη PaaS υπηρεσία service function (**GET /deployedPaasServices**, **GET /deployedPaasServices/{ deployedPaasServiceName}**).

3.1.2.2.4 Μέθοδοι ενεργοποίησης TANDEM κόμβων



Εικόνα 7: Μέθοδος ενεργοποίησης TANDEM κόμβου

Με την παραπάνω μέθοδο, ο χρήστης μπορεί να ζητήσει την ενεργοποίηση οποιουδήποτε TANDEM κόμβου. Με αυτό τον τρόπο ουσιαστικά ο χρήστης εγκαθιστά βασικές (core) PaaS υπηρεσιών στο σύμπλεγμα των κόμβων, χρησιμοποιώντας ως «σώμα» το παρακάτω (παρατίθεται με παράδειγμα για την ενεργοποίηση του κόμβου με τοποθεσία “Peania_Athens” και εγκατάσταση ως core PaaS υπηρεσίας την “EdgeX” πλατφόρμα):

```
{
  "location": "Peania_Athens",
  "paas_services": [
    {
      "paas_service_name": "EdgeX",
      "paas_instance_name": "Edgex_ICOM_deployment",
      "autoscaling_type": "maximize_performance",
      "count_min": 1,
      "count_max": 5,
      "location": "Peania_Athens_node1",
      "all_node_ports": false,
      "node_ports": [
        90,
        8080
      ]
    }
  ]
}
```

3.1.2.3 Cluster Manager

Ουσιαστικά αποτελεί το plugin για την διασύνδεση με τον Kubernetes master node. Χρησιμοποιείται η python βιβλιοθήκη (Kubernetes Python Client) αλλά και δικές μας υλοποιήσεις, όπου λειτουργούν ως client (πελάτης) προς τα υποστηριζόμενα APIs που προσφέρονται από το Kubernetes master node ώστε να εγκαθίστανται με οι υπηρεσίες στις παρυφές/ ακμές του δικτύου.

3.2 Edge Monitoring

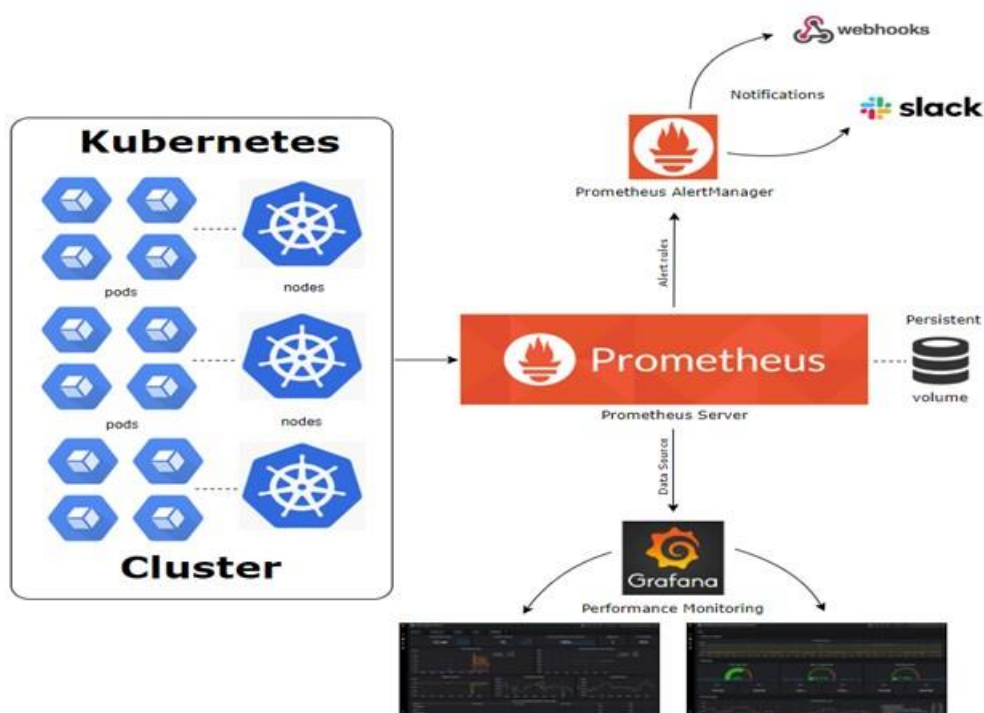
Στην ενότητα αυτή περιγράφουμε ένα ολοκληρωμένο σύνολο από υπηρεσίες ικανές να παρακολουθούν τους πόρους τόσο της φυσικής υποδομής που αποτελούν το σύμπλεγμα των μονάδων του Kubernetes ,όσο και των εφαρμογών που εγκαθίστανται στο συγκεκριμένο σύμπλεγμα. Οι υπηρεσίες αυτές αποτελούν κομμάτη της ri-Edge αρχιτεκτονικής, καθώς σε αυτό τον κύκλο εργασιών, έχουν υλοποιηθεί ολοι οι απαραίτητοι μηχανισμοί ενοποιήσεις τους. Οι πόροι που παρακολουθούνται μπορεί να είναι το ποσοστό χρήσης της επεξεργαστής

ισχύος (cpu), της μνήμης (ram) , του δίσκου (disk), η ταχύτητα διάδοσης δεδομένων μέσω της κάρτας δικτύου (bandwidth/throughput), κτλ.

Ειδικότερα το σύστημα παρακολούθησης που εγκαταστάθηκε βασίζεται στο ανοιχτού κώδικα λογισμικό **Prometheus-operator [Prometheus-Operator]** με κύριο στόχο να απλοποιήσει και να αυτοματοποιήσει τη διαμόρφωση και τη διαχείριση της στοίβας παρακολούθησης Prometheus που εκτελείται σε ένα σύμπλεγμα μονάδων Kubernetes. Συγκεκριμένα , οι υπηρεσίες που εγκαθίστανται είναι οι ακόλουθες:

- **prometheus-operator** : ο πυρήνας του συστήματος παρακολούθησης, υπεύθυνος για τη διαχείριση άλλων μικρό-υπηρεσιών του συστήματος.
- **kube-state-metrics exporter, node-exporter, blackbox-exporter** : υπηρεσίες υπεύθυνες για την εξαγωγή μετρήσεων σχετικά με τους πόρους χρήσης τόσο της φυσικής υποδομής του συμπλέγματος των μονάδων του Kubernetes, όσο και των μικρο-εφαρμογών που εγκαθίστανται στο σύμπλεγμα.
- **Prometheus Server/ promethues-adapter** : υπηρεσία υπεύθυνη για την συλλογή των μετρήσεων από τις υπηρεσίες εξαγωγής (**kube-state-metrics exporter, node-exporter, blackbox-exporter**) καθώς και την περαιτέρω αποθήκευση των δεδομένων σε σχετικές βάσεις δεδομένων (promethues TSBD).
- **Grafana** : υπηρεσία υπεύθυνη για την ζήτηση των μετρήσεων από τις βάσεις δεδομένων καθώς επίσης και των οπτικοποίηση και ανάλυση μέσω πολλαπλών διαδραστικών γραφημάτων.
- **Alertmanager** : υπηρεσία υπεύθυνη για την διαχείριση ειδοποιήσεων που αποστέλλονται από την υπηρεσία (**Prometheus-adapter /promethues-k8s**), εφόσον κάποια μέτρηση ξεπεράσει κάποιο προκαθορισμένο άνω όριο. Φροντίζει για την τη δρομολόγηση των ειδοποιήσεων στους σωστούς δέκτες μέσω email.

Οι αλληλεπιδράσεις των υπηρεσιών της στοίβας παρακολούθησης Prometheus παρουσιάζονται στην Εικόνα 9: Υψηλού Επιπέδου Αρχιτεκτονική του MEC..

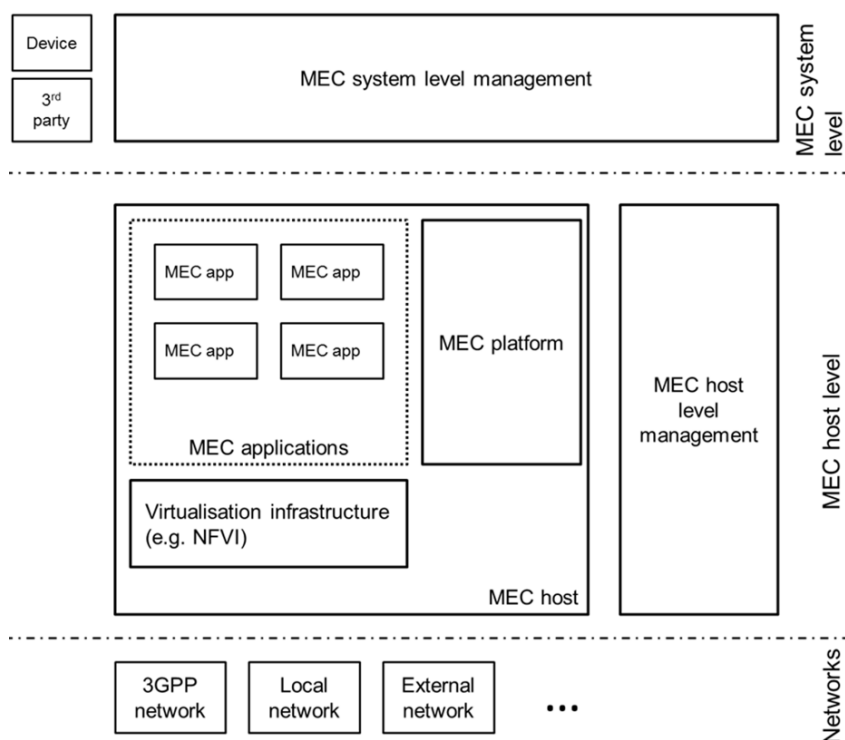


Εικόνα 8: Αρχιτεκτονική του συστήματος παρακολούθησης "Promethues-Operator"

3.3 Multi-access Edge Computing Platform

Στην πλατφόρμα του TANDEM έχουμε ενσωματώσει τη πλατφόρμα MEC, προκειμένου να εμπλουτίσουμε το TANDEM με μία πληθώρα υπηρεσιών. Η πλατφόρμα MEC αποτελεί το κυριότερο κομμάτι της αρχιτεκτονικής MEC όπως αυτή προτάθηκε από το ETSI. Στην παρούσα ενότητα θα περιγράψουμε εν συντομία τη συνολική δομή της MEC αρχιτεκτονικής και στη συνέχεια θα επικεντρωθούμε στην πλατφόρμα MEC και στις υλοποιημένες υπηρεσίες που αυτή υποστηρίζει.

Η αρχιτεκτονική Multi-access Edge Computing καθιστά δυνατή την υλοποίηση MEC εφαρμογών ως αμιγώς οντότητες λογισμικού, οι οποίες μπορούν να εκτελούνται πάνω σε μία Υποδομή Εικονικοποίησης (Virtualisation Infrastructure) κοντά ή στα άκρα του δικτύου. Η συνολική MEC αρχιτεκτονική συνοψίζεται στην Εικόνα 9: Υψηλού Επιπέδου Αρχιτεκτονική του MEC..



Εικόνα 9: Υψηλού Επιπέδου Αρχιτεκτονική του MEC..

Η παραπάνω αρχιτεκτονική χωρίζεται σε τρία επίπεδα, το επίπεδο MEC συστήματος (MEC system), το επίπεδο MEC εξυπηρετητή (MEC host) και το επίπεδο δικτύου (Networks). Στα πλαίσια της δικής μας υλοποίησης, υιοθετούμε τα δομικά στοιχεία του επιπέδου MEC εξυπηρετητή. Συγκεκριμένα, ενσωματώνοντας στην αρχιτεκτονική μας το MEC host, παρέχουμε μία οντότητα που εμπεριέχει την πλατφόρμα MEC και την Υποδομή Εικονικοποίησης, η οποία παρέχει τους απαραίτητους πόρους για να εκτελούνται οι MEC εφαρμογές. Τα τρία αυτά δομικά στοιχεία αναλύονται παρακάτω.

Η **Πλατφόρμα MEC** (MEC platform) είναι το κυριότερο δομικό στοιχείο του επιπέδου. Παρέχει τη δυνατότητα στις MEC εφαρμογές να ανακαλύψουν, καταναλώσουν, προωθήσουν ή να παρέχουν δικές τους υπηρεσίες. Επιπρόσθετα, η ίδια η πλατφόρμα διαθέτει ένα σύνολο από υπηρεσίες προς κατανάλωση από τις MEC εφαρμογές, όπως υπηρεσίες τοποθεσίας, όπως επίσης και πρόσβαση σε μόνιμα διαθέσιμη μνήμη και πληροφορίες σχετικά με την τρέχουσα ώρα της ημέρας. Προκειμένου οι διάφορες MEC εφαρμογές να έχουν πρόσβαση στις υπηρεσίες της πλατφόρμας, αυτή παρέχει μία σειρά από APIs, αφιερωμένα στην κάθε υπηρεσία ξεχωριστά.

Η **Υποδομή Εικονικοποίησης** παρέχει τους απαραίτητους υπολογιστικούς πόρους, χώρο αποθήκευσης και δικτυακούς πόρους που απαιτούνται για να λειτουργούν οι MEC εφαρμογές. Περιλαμβάνει ένα επίπεδο δεδομένων που εκτελεί τους κανόνες κυκλοφορίας που λαμβάνει από τη πλατφόρμα MEC και δρομολογεί την κίνηση μεταξύ εφαρμογών, υπηρεσιών, και άλλων δικτύων πρόσβασης.

Οι **MEC Εφαρμογές** εκτελούνται ως εικονικές εφαρμογές πάνω στην Υποδομή Εικονικοποίησης και επικοινωνούν με την Πλατφόρμα MEC προκειμένου να καταναλώσουν ή να παρέχουν κάποια υπηρεσία. Οι υπηρεσίες προς κατανάλωση δύναται να προέρχονται τόσο από την Πλατφόρμα MEC, όσο και από άλλες MEC Εφαρμογές. Οι εφαρμογές αυτές επικοινωνούν, επίσης, με την Πλατφόρμα προκειμένου να δηλώσουν/ανανεώσουν κάποια υπηρεσία που παρέχουν ή να αναζητήσουν διαθέσιμες υπηρεσίες. Τέλος, διαθέτουν ένα σύνολο από προδιαγραφές, όπως απαραίτητους πόρους, μέγιστη καθυστέρηση ή υπηρεσίες που πρέπει να είναι στη διάθεσή τους, οι οποίες προδιαγραφές θα πρέπει να ικανοποιούνται για την ομαλή εκτέλεση των MEC Εφαρμογών.

Αναφορικά με τα APIs που υλοποιήθηκαν, ακολουθούν τις προδιαγραφές που ορίζονται στην MEC Αρχιτεκτονική και διακρίνονται σε δύο κατηγορίες:

Location APIs: Ένα σύνολο από APIs που παρέχουν πληροφορίες τοποθεσίας για User Equipments (UEs):

1. UE Distance Lookup:

- Υπολογισμός απόστασης μεταξύ δύο UEs ή ενός UE και ενός σημείου, ορισμένου με γεωγραφικές συντεταγμένες.
- URL: [/queries/distance](#)
- Παράμετροι: διεύθυνση UE, διεύθυνση δευτέρου UE ή γεωγραφικές συντεταγμένες
- Επιστρέφει την απόσταση μεταξύ των δύο σημείων σε JSON μορφή που ακολουθεί το σχήμα:

```
{
  "terminalDistance": {
    "accuracy": 0,
    "distance": 0,
    "timestamp": {...}
  }
}
```

2. UE Location Lookup:

- Επιλογή ενός ή περισσότερων UE με βάση τα δοσμένα φίλτρα και προβολή πληροφοριών σχετικών με την τοποθεσία τους.
- URL: [/queries/users](#)
- Παράμετροι: διεύθυνση UE, αναγνωριστικό ζώνης στην οποία ανήκει, αναγνωριστικό σημείου πρόσβασης
- Επιστρέφει σε JSON μορφή μία λίστα από τα UEs που αντιστοιχούν στα δοσμένα φίλτρα, όπου κάθε στοιχείο της περιλαμβάνει μία σειρά από πληροφορίες σχετικές με την τοποθεσία τους:

```
{
  "userList": {
```

```

"resourceURL": "string",
"user": [
  {
    "accessPointId": "string",
    "address": "string",
    "ancillaryInfo": "string",
    "contextLocationInfo": "string",
    "locationInfo": {
      "accuracy": 0,
      "accuracyAltitude": 0,
      "accuracySemiMinor": 0,
      "altitude": 0,
      "confidence": 0,
      "includedAngle": 0,
      "innerRadius": 0,
      "latitude": [
        0
      ],
      "longitude": [
        0
      ],
      "offsetAngle": 0,
      "orientationMajorAxis": 0,
      "shape": 0,
      "timestamp": {...},
      "uncertaintyRadius": 0,
      "velocity": {...}
    },
    "resourceURL": "string",
    "timestamp": {...},
    "zoneId": "string"
  }
]
}

```

3. Closest UE:

- Εύρεση του UE που βρίσκεται πιο κοντά σε ένα άλλο UE ή σε ένα σημείο προσδιορισμένο από τις γεωγραφικές του συντεταγμένες.
- URL: </queries/closestUser>
- Παράμετροι: διεύθυνση UE ή γεωγραφικές συντεταγμένες, τύπος UE προς αναζήτηση
- Επιστρέφει σε JSON μορφή μία σειρά από πληροφορίες σχετικές με την τοποθεσία του κοντινότερου UE:

```

{
  "userList": {

```

```

"resourceURL": "string",
"user": [
  {
    "accessPointId": "string",
    "address": "string",
    "ancillaryInfo": "string",
    "contextLocationInfo": "string",
    "locationInfo": {
      "accuracy": 0,
      "accuracyAltitude": 0,
      "accuracySemiMinor": 0,
      "altitude": 0,
      "confidence": 0,
      "includedAngle": 0,
      "innerRadius": 0,
      "latitude": [
        0
      ],
      "longitude": [
        0
      ],
      "offsetAngle": 0,
      "orientationMajorAxis": 0,
      "shape": 0,
      "timestamp": {...},
      "uncertaintyRadius": 0,
      "velocity": {...}
    },
    "resourceURL": "string",
    "timestamp": {...},
    "zoneId": "string"
  }
]
}

```

4. Nearby UE(s):

- Εύρεση του/των UE(s) που βρίσκονται εντός μίας εμβέλειας από ένα άλλο UE ή ένα σημείο προσδιορισμένο από τις γεωγραφικές του συντεταγμένες.
- URL: [/queries/closeUsers](#)
- Παράμετροι: διεύθυνση UE ή γεωγραφικές συντεταγμένες, τύπος UE προς αναζήτηση, απόσταση από UE/σημείο
- Επιστρέφει σε JSON μορφή μία λίστα από τα UEs που βρίσκονται εντός της δοσμένης εμβέλειας, όπου κάθε στοιχείο της περιλαμβάνει μία σειρά από πληροφορίες σχετικές με την τοποθεσία τους:

```

{
  "userList": {
    "resourceURL": "string",
    "user": [
      {
        "accessPointId": "string",
        "address": "string",
        "ancillaryInfo": "string",
        "contextLocationInfo": "string",
        "locationInfo": {
          "accuracy": 0,
          "accuracyAltitude": 0,
          "accuracySemiMinor": 0,
          "altitude": 0,
          "confidence": 0,
          "includedAngle": 0,
          "innerRadius": 0,
          "latitude": [
            0
          ],
          "longitude": [
            0
          ],
          "offsetAngle": 0,
          "orientationMajorAxis": 0,
          "shape": 0,
          "timestamp": {...},
          "uncertaintyRadius": 0,
          "velocity": {...}
        },
        "resourceURL": "string",
        "timestamp": {...},
        "zoneId": "string"
      }
    ]
  }
}

```

Service Management APIs

1. Services:

- Ανάκτηση πληροφοριών για μία λίστα από όλες τις MEC υπηρεσίες που παρέχονται. Χρησιμοποιείται ως έλεγχος διαθεσιμότητας υπηρεσίας.
- URL: [/services](#)
- Παράμετροι: -
- Επιστρέφει σε JSON μορφή μία λίστα από όλες υπηρεσίες:

```
[
  {
    "serInstanceId": "string",
    "serName": "string",
    "serCategory": {
      "href": "string",
      "id": "string",
      "name": "string",
      "version": "string"
    },
    "version": "string",
    "state": "ACTIVE",
    "transportInfo": {...},
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": true,
    "isLocal": true,
    "livenessInterval": 0,
    "_links": {...}
  }
]
```

2. Service:

- Ανάκτηση πληροφοριών για μία συγκεκριμένη MEC υπηρεσία. Χρησιμοποιείται ως έλεγχος διαθεσιμότητας υπηρεσίας.
- URL: </service/{serviceld}>
- Παράμετροι: αναγνωριστικό της υπηρεσίας
- Επιστρέφει σε JSON μορφή όλες τις πληροφορίες για τη συγκεκριμένη υπηρεσία:

```
{
  "serInstanceId": "string",
  "serName": "string",
  "serCategory": {
    "href": "string",
    "id": "string",
    "name": "string",
    "version": "string"
  },
  "version": "string",
  "state": "ACTIVE",
  "transportInfo": {...},
  "serializer": "JSON",
  "scopeOfLocality": "MEC_SYSTEM",
  "consumedLocalOnly": true,
```

```

    "isLocal": true,
    "livenessInterval": 0,
    "_links": {...}
  }

```

3. GET Application Services:

- Ανάκτηση πληροφοριών για μία λίστα από όλες τις MEC υπηρεσίες που παρέχονται από μία δοσμένη MEC εφαρμογή. Χρησιμοποιείται ως έλεγχος διαθεσιμότητας υπηρεσίας.
- URL: [/applications/{applInstanceId}/services](#)
- Παράμετροι: αναγνωριστικό της εφαρμογής ή αναγνωριστικό υπηρεσίας ή όνομα υπηρεσίας, λοιποί παράμετροι σχετικοί με την υπηρεσία
- Επιστρέφει σε JSON μορφή μία λίστα από όλες υπηρεσίες που αντιστοιχούν στα κριτήρια:

```

[
  {
    "serInstanceId": "string",
    "serName": "string",
    "serCategory": {
      "href": "string",
      "id": "string",
      "name": "string",
      "version": "string"
    },
    "version": "string",
    "state": "ACTIVE",
    "transportInfo": {...},
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": true,
    "isLocal": true,
    "livenessInterval": 0,
    "_links": {...}
  }
]

```

4. GET Application Service:

- Ανάκτηση πληροφοριών για μία συγκεκριμένη MEC υπηρεσία που παρέχεται από μία δοσμένη MEC εφαρμογή. Χρησιμοποιείται ως έλεγχος διαθεσιμότητας υπηρεσίας.
- URL: [/applications/{applInstanceId}/services/{serviceId}](#)
- Παράμετροι: αναγνωριστικό της εφαρμογής, αναγνωριστικό της υπηρεσίας
- Επιστρέφει σε JSON μορφή όλες τις πληροφορίες για τη συγκεκριμένη υπηρεσία:

```

{
  "serInstanceId": "string",
  "serName": "string",

```

```

    "serCategory": {
      "href": "string",
      "id": "string",
      "name": "string",
      "version": "string"
    },
    "version": "string",
    "state": "ACTIVE",
    "transportInfo": {...},
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": true,
    "isLocal": true,
    "livenessInterval": 0,
    "_links": {...}
  }

```

5. POST Application Services:

- Εγγραφή καινούριας υπηρεσίας
- URL: [/applications/{applInstanceId}/services/{serviceId}](#)
- Παράμετροι: αναγνωριστικό της εφαρμογής
- Σώμα: Πληροφορίες σχετικά με την νέα εφαρμογή σε μορφή JSON.
- Επιστρέφει σε JSON μορφή όλες τις πληροφορίες για τη νέα υπηρεσία:

```

{
  "serInstanceId": "string",
  "serName": "string",
  "serCategory": {
    "href": "string",
    "id": "string",
    "name": "string",
    "version": "string"
  },
  "version": "string",
  "state": "ACTIVE",
  "transportInfo": {...},
  "serializer": "JSON",
  "scopeOfLocality": "MEC_SYSTEM",
  "consumedLocalOnly": true,
  "isLocal": true,
  "livenessInterval": 0,
  "_links": {...}
}

```

6. PUT Application Service:

- Ανανέωση υπάρχουσας υπηρεσίας

- URL: [/applications/{applInstanceId}/services](#)
- Παράμετροι: αναγνωριστικό της εφαρμογής , αναγνωριστικό της υπηρεσίας
- Σώμα: Πληροφορίες σχετικά με την εφαρμογή προς επεξεργασία σε μορφή JSON.
- Επιστρέφει σε JSON μορφή όλες τις πληροφορίες για τη ανανεωμένη υπηρεσία:

```
{
  "serInstanceId": "string",
  "serName": "string",
  "serCategory": {
    "href": "string",
    "id": "string",
    "name": "string",
    "version": "string"
  },
  "version": "string",
  "state": "ACTIVE",
  "transportInfo": {...},
  "serializer": "JSON",
  "scopeOfLocality": "MEC_SYSTEM",
  "consumedLocalOnly": true,
  "isLocal": true,
  "livenessInterval": 0,
  "_links": {...}
}
```

7. DELETE Application Service:

- Διαγραφή υπηρεσίας.
- URL: [/applications/{applInstanceId}/services](#)
- Παράμετροι: αναγνωριστικό της εφαρμογής , αναγνωριστικό της υπηρεσίας
- Επιστρέφει μήνυμα: *"No content"*

3.4 TANDEM Βασικές και Υποστηρικτικές Υπηρεσίες

Η πλατφόρμα TANDEM παρέχει μια σειρά από υπηρεσίες. Κάποιες υπηρεσίες αντιστοιχίζονται άμεσα σε απαιτήσεις τελικών χρηστών και ονομάζονται βασικές υπηρεσίες (core services), και μπορεί να είναι οι κεντρικές υπηρεσίες στις εφαρμογές των τελικών χρηστών πέρα από τη χρήση τους ως μέρος μιας αλυσίδας υπηρεσιών για τη σύνθεση πιο πολύπλοκων εφαρμογών. Αυτές οι υπηρεσίες είναι οι βασικές υπηρεσίες προς τους πελάτες, και ονομάζονται **Υπηρεσίες Εφαρμογής (Application Services)**. Βασικές υπηρεσίες της πλατφόρμας TANDEM που μπορεί να έχουν και το ρόλο των Υπηρεσιών Εφαρμογής είναι:

1. Η Ανάλυση Δεδομένων από IoT συσκευές (IoT Data Analytics)

2. Η Αναγνώριση Αντικειμένων σε ροές βίντεο (**Αναγνώριση Αντικειμένων - Object Recognition**)
3. Η Πρόβλεψη Μελλοντικών τιμών μετρήσεων από IoT συσκευές όπως από αισθητήρες (**Time Series Forecasting**).

Αρχικές εκδόσεις των δύο πρώτων είναι διαθέσιμες στην πρώτη έκδοση του λογισμικού TANDEM.

Πέρα από τις βασικές υπηρεσίες, υπάρχουν και οι υποστηρικτικές υπηρεσίες που απαιτούνται στη δημιουργία εφαρμογών, για παράδειγμα για να έχουμε πρόσβαση σε δεδομένα από εξωτερικές συσκευές. Αυτές οι υπηρεσίες είναι οι:

1. Υπηρεσίες Ανακάλυψης και Διαχείρισης Συσκευών
2. Υπηρεσίες Χρονοδρομολόγησης (Scheduling Services)
3. Υπηρεσίες Συναγερμών και Ειδοποιήσεων (Alerts & Notifications Services)
4. Υπηρεσίες Κανόνων (Rules Engine Services)

Οι υποστηρικτικές IoT υπηρεσίες του TANDEM βασίζονται σε αντίστοιχες υπηρεσίες που παρέχει η IoT πλατφόρμα **EdgeXFoundry** η οποία αποτελεί μέρος της αρχικής έκδοσης της πλατφόρμας TANDEM και προσφέρονται ενοποιημένα από την ri-Edge πλατφόρμα, καθώς αποτελεί μέρος της αρχιτεκτονικής της

Επίσης υποστηρικτική υπηρεσία είναι οι Ουρές Μηνυμάτων (Message Queues) για την επικοινωνία και ανταλλαγή δεδομένων μεταξύ των υπηρεσιών που αποτελούν μέρος μιας σύνθετης εφαρμογής.

Το Edge Host (μέσω τις ri-Edge πλατφόρμας) περιλαμβάνει επίσης Υπηρεσίες Συστήματος (System Services) που αντιστοιχούν στις Απαραίτητες **Υπηρεσίες MEC (MEC Services)** όπως η Υπηρεσία Τοποθεσίας (Location Service) και η Υπηρεσία Διαχείρισης Κυκλοφορίας (Traffic Management Service). Υπηρεσίες συστήματος που εν μέρη είναι διαθέσιμες σε αυτή την έκδοση λογισμικού, και θα εξελεχουν περαιτέρω στις επόμενες.

Παρακάτω παρουσιάζεται το **EdgexFoundry** που προσφέρει λειτουργίες IoT στην πλατφόρμα του TANDEM.

EdgexFoundry

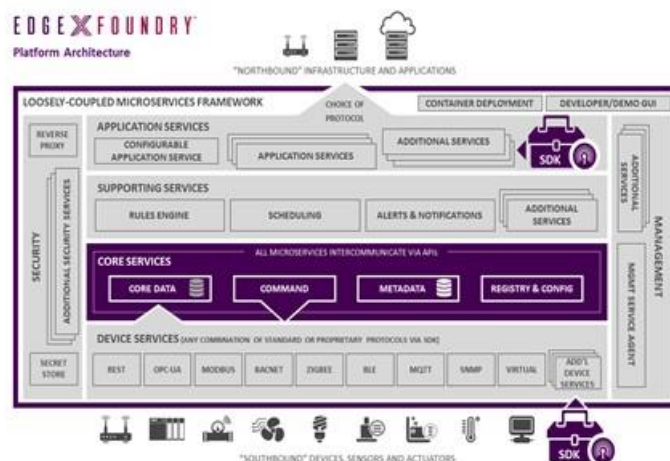
Το EdgeX Foundry είναι μία ευέλικτη και επεκτάσιμη πλατφόρμα ανοιχτού λογισμικού για χρήση στα άκρα του δικτύου, υποστηριζόμενη από το μη-κερδοσκοπικό οργανισμό Linux Foundation. Σκοπός του είναι να παρέχει μία ενοποιημένη ανοιχτή πλατφόρμα για IoT υπολογιστική στα άκρα του δικτύου, η οποία θα διευκολύνει την αλληλεπίδραση μεταξύ των συσκευών και των εφαρμογών στα άκρα του δικτύου.



Εικόνα 10: Το EdgeXFoundry ανάμεσα στα Αντικείμενα και στο Πληροφοριακό Σύστημα

Συγκεκριμένα, το EdgeXFoundry αλληλοεπιδρά με το φυσικό κόσμο των συσκευών, αισθητήρων, actuators και άλλων IoT αντικειμένων και δρα ως middleware μεταξύ αυτών και διάφορων πληροφοριακών συστημάτων και εφαρμογών, με τα οποία ανταλλάσσει δεδομένα (Εικόνα 10). Στον πυρήνα αυτού του project βρίσκεται ένα πλαίσιο αλληλεπίδρασης που φιλοξενείται μέσα σε μία πλατφόρμα λογισμικού αναφοράς η οποία είναι εντελώς ανεξάρτητη του υλικού και του λειτουργικού συστήματος. Με την αλληλεπιδραστικότητα που παρέχει, του EdgeX Foundry διευκολύνει την παρακολούθηση των αντικειμένων του φυσικού κόσμου, την αποστολή οδηγιών σε αυτά, τη συλλογή δεδομένων από αυτά, την προώθηση των δεδομένων σε υπολογιστικά νέφη όπου μπορεί να γίνει η αποθήκευση, συγκέντρωση, ανάλυσή τους, η λήψη αποφάσεων με βάση αυτά και η εφαρμογή των ανάλογων ενεργειών στα φυσικά αντικείμενα.

Η συγκεκριμένη πλατφόρμα ακολουθεί μία βαθμωτή IoT αρχιτεκτονική με χαλαρή σχέση μεταξύ των συστατικών μερών της. Η αρχιτεκτονική αποτελείται από βαθμίδες, κάθε μία από τις οποίες με τη σειρά της περιλαμβάνει ένα σύνολο μικροϋπηρεσιών (microservices) που επιτελούν συγκεκριμένες λειτουργίες στους υπολογιστικούς κόμβους στα άκρα του δικτύου και απεικονίζεται στην Εικόνα 11: Η Αρχιτεκτονική του EdgeXFoundry.



Εικόνα 11: Η Αρχιτεκτονική του EdgeXFoundry

3.4.1 IoT Data Analytics

Στην ενότητα αυτή περιγράφουμε μια βασική **Υπηρεσία Εφαρμογής**, την **“IoT Data Analytics”** που είναι υπεύθυνη για την Ανάλυση Δεδομένων από IoT συσκευές, καθώς και τις υπο-υπηρεσίες που εσωκλείει. Συγκεκριμένα η **IoT Data Analytics** έχει υλοποιηθεί σαν μια custom PaaS υπηρεσία η οποία έχει ως ευθύνη την οπτικοποίηση πληθώρας μετρήσεων που συλλέγονται από αισθητήρες IoT συσκευών οι οποίες είναι συνδεδεμένες στην EdgeX

PaaS υπηρεσία. Συγκεκριμένα η σύνθετη **IoT Data Analytics** PaaS υπηρεσία αποτελείται από τα παρακάτω υπηρεσίες:

1. **iotmonitoring-mqtt** : υπο-υπηρεσία υπεύθυνη να υλοποιεί την λειτουργικότητα ενός ανοιχτού κώδικα message broker που υλοποιεί τις εκδόσεις 5.0, 3.1.1 και 3.1 του πρωτοκόλλου MQTT. Συγκεκριμένα , το πρωτόκολλο MQTT είναι ένα ελαφρύ πρωτόκολλο που υποστηρίζει εφαρμογές του Internet of Things (IoT). Ο Eclipse Mosquitto Mqtt Broker [Mosquitto] που χρησιμοποιήθηκε, δίνει την δυνατότητα , ως διαμεσολαβητής, να επικοινωνούν μεταξύ τους οι πελάτες και οι συνδρομητές που είναι συνδεδεμένοι στον broker, μέσω της δημοσίευσης/συνδρομής (pub/sub) μηνυμάτων σε έναν δίαυλο (bus). Στην περίπτωση μας, η υπο-υπηρεσία συλλέγει τις μετρήσεις από αισθητήρες IoT συσκευών που υπάρχουν στην υπο-υπηρεσία **app-service-configurable-mqtt** της EdgeX PaaS υπηρεσίας η οποία έχει εγγραφεί στον MQTT-Broker σαν πελάτης. Μετά την συλλογή, αποθηκεύει τις μετρήσεις εσωτερικά στον δίαυλο (bus) ώστε μελλοντικά να διανεμηθούν στους υποψήφιους συνδρομητές όπως το **iotmonitoring-messenger/IoT Data monitoring service**.
2. **iotmonitoring-messenger/IoT Data monitoring service** : υπο-υπηρεσία υπεύθυνη να συλλέγει τις μετρήσεις που αποθηκεύονται στην υπο-υπηρεσίας **iotmonitoring-mqtt** και να τις εγγράφει σε μια **influxDB [InfluxDB]** βάση δεδομένων (**iotmonitoring-influxdb**). Επίσης, είναι υπεύθυνη εφόσον οι μετρήσεις ξεπεράσουν τα προβλεπόμενα άνω όρια να πυροδοτήσει διάφορους μηχανισμούς όπως “alarm” και “notification”. Η συγκεκριμένη υπο-υπηρεσία έχει υλοποιηθεί σε γλώσσα προγραμματισμού python.
3. **iotmonitoring-influxdb** : υπο-υπηρεσία υπεύθυνη να υλοποιεί την λειτουργικότητα μιας InfluxDb βάσης δεδομένων για την αποθήκευση των μετρήσεων που συλλέγονται από αισθητήρες IoT συσκευών. Συγκεκριμένα, η InfluxDB είναι μια πλατφόρμα χρονοσειρών ανοιχτού κώδικα, που περιλαμβάνει ένα API για την αποθήκευση και την αναζήτηση δεδομένων, την επεξεργασία τους στο παρασκήνιο για σκοπούς παρακολούθησης και ειδοποίησης, καθώς και πίνακες εργαλείων χρηστών και για εξερεύνηση των δεδομένων και πολλά άλλα. Στην συγκεκριμένη βάση δεδομένων τα δεδομένα γράφονται και διαβάζονται σε πραγματικό χρόνο καθώς επίσης εφαρμόζονται τεχνικές μηχανικής μάθησης για τον εντοπισμό διάφορων ανωμαλιών.
4. **iotmonitoring-grafana** : υπο-υπηρεσία υπεύθυνη να υλοποιεί την λειτουργικότητα μιας γραφικής πλατφόρμας για την ανάλυση δεδομένων, συγκεκριμένα της πλατφόρμας Grafana. Συγκεκριμένα η πλατφόρμα Grafana είναι μια διαδικτυακή εφαρμογή ανοιχτού κώδικα για την ανάλυση στοιχείων και μετρήσεων καθώς και για την διαδραστική οπτικοποίηση τους μέσω πολλαπλών γραφημάτων. Στην περίπτωση μας, ρόλος της είναι η ζήτηση των μετρήσεων από την βάση δεδομένων της υπο-υπηρεσίας **iotmonitoring-influxdb** για την οπτικοποίηση των αποθηκευμένων μετρήσεων.

3.4.2 TANDEM Object Detection Service

3.4.2.1 Εισαγωγή

Η υπηρεσία αυτή έχει υλοποιηθεί με χρήση containers και είναι διαθέσιμη στην πρώτη έκδοση του λογισμικού.

Πραγματοποιεί αυτόματη αναγνώριση σε εισερχόμενο σήμα βίντεο συγκεκριμένων αντικειμένων, όπως άνθρωποι/ αυτοκίνητα/ μηχανές/ σακίδια κοκ.

Εγκαθίσταται στους κόμβους άκρων ώστε να μη χρειάζεται το βίντεο να ταξιδεύει στο δίκτυο και να το επιβαρύνει.

Η αναλυτική παρουσίαση του εν λόγω συστήματος ξεκινάει από μία κρίσιμη υποδομή που ελέγχεται διαρκώς από μία (ή περισσότερες) κάμερες. Για παράδειγμα σε ένα αεροδρόμιο θα ελέγχονται όλοι οι χώροι για χειραποσκευές χωρίς ιδιοκτήτη. Ενώ σε ένα σενάριο ευφυούς καλλιέργειας θα ελέγχονται μεγάλες εκτάσεις για πιθανή προσβολή από κάποια ασθένεια.

Για την αδιάλειπτη επιτήρηση χρησιμοποιούνται κάμερες που παρακολουθούν 24 ώρες x 7 ημέρες την κινητικότητα χωρίς να υπάρχουν κρυφά σημεία. Το συνεχές σήμα της κάμερας εισάγεται σε ένα ιδιωτικό υποδίκτυο όπου έχει δυνατότητα πρόσβασης μία υπηρεσία επεξεργασίας εικόνας.

Πρόκειται για μία εξατομικευμένη υπηρεσία γραμμένη σε γλώσσα προγραμματισμού **Python**. Χρησιμοποιώντας τη βιβλιοθήκη **OpenCV[OpenCV]** είναι σε θέση να επεξεργαστεί το σήμα video και να εξάγει διαδοχικά frames που αναπαριστούν την εξέλιξη ενός φαινομένου σε διακριτές χρονικές στιγμές.

Στη συνέχεια οι εικόνες περνάνε από φίλτρα με σκοπό τη διακράτηση μόνο της χρήσιμης πληροφορίας και την απόρριψη του υπόλοιπου τμήματος. Ο γενικός κανόνας είναι ότι η οπτική πολυμεσική αποτύπωση είναι η πιο εύγλωττη όταν καταλήγει σε έναν άνθρωπο που θέλει να βγάλει κάποια συμπεράσματα. Αλλά όταν απευθύνεται σε ένα υπολογιστή που τρέχει μία προγραμματιστική εφαρμογή δεν είναι απαραίτητος όλος αυτός ο μεγάλος όγκος δεδομένων και επιβάλλεται για λόγους βέλτιστης αξιοποίησης υποδομών και χρόνου εκτέλεσης να απομακρυνθεί οτιδήποτε άσχετο με το τελικό αλγοριθμικό στόχο. Συνήθως οι αρχικές εικόνες έχουν χρωματική συνιστώσα και η πρώτη αποδόμηση που γίνεται είναι η μετατροπή τους στις αποχρώσεις του γκρι. Αυτό επιτυγχάνεται πάλι με τη βιβλιοθήκη OpenCV.

Το δεύτερο βήμα στην επεξεργασία εικόνας είναι σχετικό με τη γενική φιλοσοφία των συστημάτων αναγνώρισης που προτείνονται. Ακολουθούν τη φιλοσοφία των ΕΥΦΥΩΝ συστημάτων και ο σχεδιασμός τους επιβάλλει τη μετάβαση από τρεις φάσεις: εκπαίδευση (training) – έλεγχος (testing) και εφαρμογή με επαναλαμβανόμενη αξιολόγηση (validation). Για την ολοκλήρωση όλων αυτών των βημάτων με τρόπο ουσιαστικό απαιτείται ένας εξαιρετικά ΜΕΓΑΛΟΣ αριθμός από διαφορετικές εικόνες. Ο αριθμός αυτός είναι δύσκολο να συγκεντρωθεί με πραγματικές εικόνες που λαμβάνονται από ανθρώπινους συντελεστές που παρακολουθούν 24 ώρες όλες τις μέρες του χρόνου ένα γεγονός να εξελίσσεται. Για παράδειγμα η κίνηση σε ένα δρόμο πρέπει αποτυπωθεί από τα πρωί μέχρι το βράδυ επίσης πρέπει να συμπεριλάβουμε τις επιπτώσεις του καιρού δείχνοντας το ίδιο τοπίο με βροχή, χιόνια ή συννεφιά.

Η επίδραση του παράγοντα σχετικά με την ώρα της ημέρας αλλά και του συμπληρωματικού σχετικά με την εποχή του χρόνου μπορεί να προστεθεί με εργαλεία τεχνητής όρασης. Δεν

είναι απαραίτητη η αναμονή της παρέλευσης μηνών για την μετάβαση από το χειμώνα στο καλοκαίρι για παράδειγμα αλλά με τεχνητό τρόπο μπορεί να γίνει μετατροπή της αρχικής εικόνας σε χειμωνιάτικο ντεκόρ κρατώντας την σημαντική πληροφορία και αλλάζοντας το επίπεδο καιρικές συνθήκες. Ο προτεινόμενος κώδικας έχει τη δυνατότητα να απομακρύνει το τρέχον αποτύπωμα της ώρας και της εποχής. Στη συνέχεια ο χρήστης μπορεί να ζητήσει τη μετάβαση σε άλλη εποχή ή ώρα με καθαρά τεχνικά μέσα. Μία συλλογή από τέτοια εργαλεία υπάρχουν στη βιβλιοθήκη **Albumentations [ALBUMENTATIONS]** που χρησιμοποιείται σε πολλά σημεία του κώδικα.

Μετά τη διακριτοποίηση του σήματος video σε εικόνες και τον μετασχηματισμό τους στη συνέχεια σε αποχρώσεις του γκρι και σε πολλά επίπεδα εποχής και χρόνου σειρά έχει η απάντηση σε ένα ερώτημα σχετικό με την οικονομία σε δέσμευση υποδομών. Ένα σήμα video παράγει ένα πολύ ΜΕΓΑΛΟ αριθμό από εικόνες που η επεξεργασία του συνόλου θα απαιτούσε σημαντικό χρόνο και μέσα. Επιπλέον τα ενδιαφέροντα φαινόμενα που τυγχάνουν προσοχής έχουν μικρή διάρκεια και δεν έχουν προκαθορισμένη συχνότητα εμφάνισης. Άρα από τον παραγόμενο όγκο εικόνων μόνο ένα μικρό υποσύνολο έχει ουσιαστική αξία και πρέπει να επεξεργαστεί στα πλαίσια του γενικού προβλήματος αναγνώριση αντικειμένων. Η προτεινόμενη υπηρεσία για να είναι σε θέση να επιτύχει τα καλύτερα αποτελέσματα στο μικρότερο χρόνο με τα λιγότερα μέσα εφαρμόζει ένα κριτήριο ΠΡΟΕΠΙΛΟΓΗΣ. Αποφεύγει να στείλει όλες τις εικόνες για επεξεργασία. Είτε χρησιμοποιεί ένα σταθερό χρονικό βήμα, για παράδειγμα κάθε 2 δευτερόλεπτα, που είναι αρκετά μικρό και δεν υπάρχει κίνδυνος να συμβεί κάτι σοβαρό και να μην το αντιληφθεί η υπηρεσία αναγνώρισης. Είτε εξετάζει αν υπάρχουν ουσιαστικές διαφορές στο περιεχόμενο διαδοχικών στιγμιότυπων που να δικαιολογούν την επεξεργασία των καινούργιων διαφορετικά πρόκειται για επαναλαμβανόμενη γνώση και πρέπει να μείνουν στο παρασκήνιο.

Η εξειδικευμένη υπηρεσία που αναλύθηκε παραπάνω είχε σαν στόχο τον εντοπισμό και την προετοιμασία των σημαντικών εικόνων από ένα βίντεο επιτήρησης. Σειρά έχει η επεξεργασία τους με βάση τη μεθοδολογία των ευφυών μοντέλων. Πρόκειται για μία σύνθετη διαδικασία που ολοκληρώνεται σε τρία βήματα. Χρονικά προηγείται η ΕΚΠΑΙΔΕΥΣΗ του μοντέλου. Ένας μεγάλος αριθμός από εικόνες χρησιμοποιείται για τον προσδιορισμό του ζητούμενου αντικειμένου στο μοντέλο. Οι εικόνες πρέπει να καλύπτουν ένα μεγάλο φάσμα πιθανών εκφάνσεων ώστε οι προτάσεις του μοντέλου να είναι σωστές σε αποκλίνουσες καταστάσεις: μερική υπερκάλυψη ζητούμενου αντικειμένου, χαμηλός φωτισμός, διαφοροποιημένη θέση κάμερας, κοκ.

Μετά τη φάση της εκπαίδευσης ακολουθεί η φάση του ελέγχου (testing) όπου το προτεινόμενο μοντέλο για το πρόβλημα αναγνώρισης ανά χείρας βαθμολογείται με βάση τις επιδόσεις του σε μία σειρά από αποφάσεις με γνωστή απάντηση. Αν η επίδοση του κριθεί προβληματική με πολλές αποτυχίες το μοντέλο δεν προχωράει σε χρήση αλλά στέλνεται για επανεκπαίδευση. Μόνο όταν ελεγχθεί με ικανοποιητικά αποτελέσματα προωθείται αλλά και πάλι ανά τακτά χρονικά διαστήματα αξιολογείται μήπως έχει ξεπεραστεί από τις συνθήκες λειτουργίας του. Σε ένα κόσμο που αλλάζει κάτι που σχεδιάζεται με βάση το σήμερα χρειάζεται ενημέρωση σύντομα.

Η φάση της εκπαίδευσης είναι εξαιρετικά χρονοβόρα και επίπονη καθώς απαιτείται επεξεργασία ενός πολύ μεγάλου αριθμού από εικόνες. Χρειάζεται εξειδικευμένος εξοπλισμός που είναι δύσκολο να αποκτηθεί. Για αυτό το λόγο οι περισσότερες προσπάθειες εισαγωγής ευφύων μοντέλων σε προβλήματα αναγνώρισης αντικειμένων καταφεύγουν σε βιβλιοθήκες με έτοιμα μοντέλα. Έχουν σχεδιαστεί από μεγάλους οργανισμούς με παγκόσμιο κύρος που διατηρούν εκατομμύρια εικόνες από αντικείμενα καθημερινής χρήσης.

Στην προκειμένη περίπτωση οι βιβλιοθήκες που επιλέχθηκαν είναι οι: **TensorFlow [TensorFlow]** και **PyTorch [PyTorch]**. Πρόκειται για δύο από τις πιο διαδεδομένες που προσφέρουν πληθώρα εργαλείων για όλα τα ζητήματα εισαγωγής Ευφύων Μοντέλων για όλες τις πλατφόρμες: από μεγάλους servers με πολλαπλές GPU έως κινητές συσκευές περιορισμένων δυνατοτήτων.

Οι βιβλιοθήκες προσφέρουν τα βασικά εργαλεία για κάθε πρόβλημα Τεχνητής Νοημοσύνης. Εξειδικευμένα προβλήματα που έχουν σαν ζητούμενο την αναγνώριση αντικειμένων, όπως αυτοκίνητα ή άνθρωποι, απαιτούν τη χρήση συγκεκριμένων φίλτρων που εγκαθίστανται στο «υπέδαφος» των βιβλιοθηκών. Στη προτεινόμενη υλοποίηση δοκιμάστηκαν δύο «αρχιτεκτονικές»: **yolo [Yolo]** και **detectron2[Detectron]**.

Το YOLO ξεχωρίζει λόγω της ικανότητας του να εντοπίζει γρήγορα ζητούμενα αντικείμενα ακόμα και όταν δεν εμφανίζονται ολόκληρα σε «άγνωστες» εικόνες. Από την άλλη μεριά το detectron2 κάνει αναγνώριση και εντοπισμό αντικειμένων. Στο τέλος δίνει τη περιοχή της εικόνας που φιλοξενεί το αντικείμενο.

3.4.2.2 Αναγνώριση Αντικειμένων με Ευφυή Μοντέλα – Φάση ΕΚΠΑΙΔΕΥΣΗΣ (Training) και επιλογής ΚΑΛΥΤΕΡΟΥ

Μία θεμελιώδης επιλογή σχετικά με την υλοποίηση μίας ευφυούς υπηρεσίας είναι η γλώσσα προγραμματισμού και η αρχιτεκτονική φιλοξενίας. Η πιο διαδεδομένη λύση είναι η γλώσσα Python καθώς προσφέρει μεγάλη φιλικότητα στο προγραμματιστή αλλά το κυριότερο είναι ο μεγάλος αριθμός καταξιωμένων βιβλιοθηκών Τεχνητής-Νοημοσύνης που την συνοδεύουν.

Οι σύνθετες εφαρμογές που διαθέτουν μία μεγάλη δεξαμενή κώδικα είναι μια μεγάλη πρόκληση γιατί πρέπει να τρέχουν με βέλτιστο τρόπο ανεξάρτητα από τον εξυπηρετητή (server) που τις φιλοξενεί. Υπάρχουν περιπτώσεις που ο σχεδιασμός και ο έλεγχος της εφαρμογής γίνεται σε ένα εξυπηρετητή με αρχιτεκτονική A (για παράδειγμα windows) και η εμπορική εγκατάσταση γίνεται σε διαφορετική αρχιτεκτονική (για παράδειγμα Linux) με απρόβλεπτες επιπτώσεις. Για την αντιμετώπιση των προβλημάτων ασυμβατότητας έχει επιλεγεί η υλοποίηση σε Docker Containers αφού προσφέρει το πλεονέκτημα της «απομονωμένης» συνύπαρξης. Συγκεκριμένα ο κώδικας υλοποίησης του ευφύου μοντέλου και όλες οι υποστηρικτικές βιβλιοθήκες διανέμονται σε μία εικονική οντότητα που εξασφαλίζει αυτοτέλεια σε οποιοδήποτε εξυπηρετητή κληθεί να εγκατασταθεί.

Η φάση εκπαίδευσης ενός ευφύου μοντέλου απαιτεί τη δέσμευση σχεδόν όλων των αποθεμάτων σε μνήμη (RAM) και επεξεργαστική ισχύ (CPU). Το αποτέλεσμα είναι ο εξυπηρετητής να αρχίσει να υπολειτουργεί και να μην μπορεί να εξυπηρετήσει άλλες υπηρεσίες πέραν του Ευφύου μοντέλου. Για την βελτιστοποίηση της απόδοσης του μοντέλου και την απόκριση σε πραγματικό χρόνο αλλά και τον εξορθολογισμό της

λειτουργίας του εξυπηρετητή για να μπορεί να ικανοποιεί πολλαπλές υπηρεσίες έχει επιλεγεί η λύση της GPU. Στα συμβατικά προγραμματιστικά έργα όλος ο όγκος υπολογισμών γίνεται στην κεντρική-μονάδα (CPU) αλλά τα έργα Τεχνητής-Νοημοσύνης διακρίνονται από μεγάλο όγκο μαθηματικών πράξεων και το πιο κατάλληλο σημείο για την εκτέλεση τους είναι η κάρτα γραφικών (GPU). Η προτεινόμενη υλοποίηση χρησιμοποιεί κατάλληλες βιβλιοθήκες που επικοινωνούν απευθείας με τη GPU και αφήνουν «ελεύθερη» την CPU για το υπόλοιπο σύστημα. Η πιο διαδεδομένη λύση σε αυτή τη κατεύθυνση είναι οι κάρτες γραφικών της εταιρίας NVIDIA με πολλαπλές επιλογές ανάλογα με το διατιθέμενο ποσό αγοράς.

Αναφέρθηκε ότι ο κώδικας και οι υποστηρικτικές βιβλιοθήκες διανέμονται σε μορφή CONTAINERS με σκοπό την μέγιστη δυνατή ΑΥΤΑΡΚΕΙΑ. Για τη δημιουργία των containers χρησιμοποιούνται απλά αρχεία κειμένου που περιέχουν οδηγίες και λέγονται Dockerfiles. Ο ακολουθούμενος κύκλος εργασίας περιλαμβάνει ένα εξυπηρετητή εργασίας (Development server) όπου γίνεται η συγγραφή του κώδικα της ευφυούς-υπηρεσίας. Όταν φτάσει η ανάπτυξη σε ένα αποδεκτό επίπεδο ωριμότητας γίνεται αποθήκευση της σταθερής έκδοσης σε ένα εξωτερικό REPOSITORY όπως είναι το GitHub. Από αυτό το σημείο οποιοσδήποτε εγκεκριμένος συνεργάτης μπορεί να «κατεβάσει» το μοντέλο για δοκιμή.

Μία άλλη εναλλακτική επειδή χρησιμοποιούνται Docker-Containers είναι το «παγκόσμιο» σημείο συγκέντρωσης που διατηρεί η Docker με ηλεκτρονική διεύθυνση hub.docker.com. Ο μηχανικός ανάπτυξης κάθε φορά που υπάρχει μία «σταθερή» έκδοση φτιάχνει ένα Image με το κώδικα και τις απαραίτητες βιβλιοθήκες που ανεβάζει στο Hub. Ο μηχανικός διαχείρισης των εμπορικών εγκαταστάσεων κατεβάζει το τελευταίο Image και το τρέχει στα σημεία επιτήρησης για «επικίνδυνα» αντικείμενα.

3.4.2.3 Αναγνώριση Αντικειμένων με Ευφυή Μοντέλα – Φάση ΕΦΑΡΜΟΓΗΣ (Inference) σε άγνωστες εικόνες

Η φάση της εκπαίδευσης ενός ευφυούς μοντέλου είναι ιδιαίτερα χρονοβόρα και απαιτητική σε υποδομές και για αυτό το λόγο αποφεύγεται. Ο ενδιαφερόμενος περνάει απευθείας στην εφαρμογή χρησιμοποιώντας ένα έτοιμο μοντέλο που έχει εκπαιδευτεί από ένα έγκυρο οργανισμό. Για αυτό το σκοπό στη προτεινόμενη υπηρεσία αναγνώρισης αντικειμένων έχουν επιλεγεί δύο βιβλιοθήκες εγνωσμένης αξίας.

Οι βιβλιοθήκες Τεχνητής-Νοημοσύνης προσφέρουν τα απαραίτητα εργαλεία για την κατάστρωση ενός υπολογιστικού έργου. Κυρίως πρόκειται για βοηθήματα για την αναλυτική παρουσίαση με γραφήματα της ροής ελέγχου και την εξαγωγή συμπερασμάτων για την ορθολογιστική λειτουργία του μοντέλου. Επίσης προσφέρουν μια σειρά από μετρητικά απόδοσης που χρησιμεύουν για την «συνεκτική» παρουσίαση των ορθών και εσφαλμένων αποφάσεων σε συνολικό επίπεδο.

Η σημαντικότερη όμως συνεισφορά των βιβλιοθηκών Τεχνητής-Ευφυίας είναι οι τεράστιες δεξαμενές κώδικα που μπορεί να καλέσει ο μηχανικός-εφαρμογών για να επιτύχει μία γρήγορη και αξιόπιστη υλοποίηση του προβλήματος του. Προσφέρονται σε διάφορες εκδόσεις που αποσκοπούν να προσφέρουν λύσεις ανεξάρτητα με τη γλώσσα προγραμματισμού. Έτσι προσφέρουν μηχανισμούς διασύνδεσης των δεξαμενών κώδικα της βιβλιοθήκης με την εκάστοτε εφαρμογή είτε είναι γραμμένη σε Python, Java ή κάποια άλλη γλώσσα.

Τέλος ο μηχανικός ανάπτυξης που θα καταφύγει σε μία βιβλιοθήκη Τεχνητής-Νοημοσύνης πάντα αποσκοπεί να επωφεληθεί από μία συλλογή από έτοιμα μοντέλα. Συγκεντρώνονται σε ένα κεντρικό σημείο με κωδική ονομασία Hub ή Zoo και ο μηχανικός-ανάπτυξης μπορεί να τα «κατεβάσει» στην υπο-κατασκευή υπηρεσία με την συγγραφή ελάχιστου η καθόλου κώδικα διασύνδεσης.

Η υπηρεσία ευφυούς-αναγνώρισης μπορεί να εγκατασταθεί σε διάφορα σημεία ανάλογα με το βαθμό αποκέντρωσης που επιζητάτε στην υλοποίηση. Για παράδειγμα στη περίπτωση επιτήρησης ενός χώρου με IP-κάμερες και επεξεργασία σε ένα κεντρικό σημείο οι απαιτήσεις σε υπολογιστική ισχύ δεν είναι υπερβολικά δεσμευτικές. Το προτεινόμενο σύστημα επιτρέπει την συγκέντρωση βίντεο από πολλαπλά σημεία σε ένα τοπικό εξυπηρετητή με εξαιρετικές δυνατότητες σε CPU και GPU. Υπάρχει όμως και το εναλλακτικό σενάριο όπου δίπλα σε κάθε κάμερα επιτήρησης τοποθετείται ένας μικρο-υπολογιστής που τρέχει στο «τελευταίο-μίλι» τον κώδικα αναγνώρισης. Στη δεύτερη περίπτωση τα υπολογιστικά μέσα είναι σημαντικά περιορισμένα σε σχέση με τη πρώτη του εξυπηρετητή σε ένα data-room. Οι βιβλιοθήκες Τεχνητής-Ευφυίας είναι σε θέση να ανταπεξέλθουν σε όλες τις περιστάσεις καθώς διατηρούν εκδόσεις με «light» χαρακτηριστικά που καλύπτουν τις βασικές ανάγκες αλλά και «πλήρεις» εκδόσεις με αυξημένες ανάγκες.

Με αυτό το σκεπτικό και κυρίως τη δυνατότητα η υπηρεσία να μην περιορίζεται από τις αρχιτεκτονικές φιλοξενίας έχουν επιλεγεί οι βιβλιοθήκες **PyTorch** και **TensorFlow**. Ο Python κώδικας που υλοποιεί το έργο ευφυούς αναγνώρισης έχει πολλαπλές εκδόσεις που συνεργάζονται με τις πλήρεις και «light» εκδόσεις των βιβλιοθηκών. Όσων αφορά το τελικό στόχο που είναι η αναγνώριση αντικειμένων υλοποιείται με προ-εκπαιδευμένα μοντέλα που υπάρχουν στα αντίστοιχα Hubs. Ξεκινώντας από το TensorFlow οι μηχανικοί ανάπτυξης απευθύνθηκαν στην εισαγωγική σελίδα <https://tfhub.dev/>. Υπάρχουν έτοιμα μοντέλα για πολλαπλά επίπεδα υπηρεσιών: Αναγνώριση Κειμένου, Εικόνας, Βίντεο και Ήχου. Στη συγκεκριμένη περίπτωση ενδιαφέρουν κυρίως τα μοντέλα που χρησιμοποιούνται για επεξεργασία εικόνας και απαντάνε σε ερωτήματα ύπαρξης αλλά και εντοπισμού. Η πρώτη κατηγορία του Object-Detection απαντάει στο βασικό ερώτημα αν υπάρχει ένα ζητούμενο αντικείμενο σε μία ομάδα (<https://tfhub.dev/s?module-type=image-object-detection>). Στη περίπτωση του εντοπισμού (Image-Segmentation) η έρευνα είναι πιο εμπεριστατωμένη και προσδιορίζει σε πια ακριβώς περιοχή της εικόνας βρίσκεται το ζητούμενο αντικείμενο (<https://tfhub.dev/s?module-type=image-segmentation>).

3.4.2.4 Η οργανωμένη εκτέλεση έργου Τεχνητής-Νοημοσύνης σε περιβάλλον Μικρο-Υπηρεσιών

Αναφέρθηκαν παραπάνω τα πλεονεκτήματα των Containers και οι λόγοι που κρίθηκαν απαραίτητοι για την υλοποίηση του ευφυούς-μοντέλου αναγνώρισης αντικειμένων σε επιτηρούμενο χώρο. Το σημαντικότερο σημείο της επιχειρηματολογίας ήταν η απομόνωση που προσφέρει από το υπόλοιπο σύστημα. Ο κώδικας συσσωματώνεται μαζί με τις απαραίτητες βιβλιοθήκες σε μία κλειστή οντότητα που είναι έτοιμη για εκτέλεση.

Οι Containers συνήθως φιλοξενούν στοχευμένες μικρο-υπηρεσίες δηλαδή θεωρείτε ΛΑΘΟΣ επιλογή η συγκέντρωση σε ένα Container μίας μεγάλης δεξαμενής κώδικα που κάνει πολλά πράγματα. Γίνεται ένας κατακερματισμός του συνολικού έργου σε πολλά μικρότερα που

πρέπει να συντονιστούν για να έχουν ταυτόχρονη παρουσία στη λήψη αποφάσεων. Η ανταλλαγή μηνυμάτων ανάμεσα τους γίνεται διαμέσων κάποιου MessageQueue, όπως είναι το Redis.

Ίσως ο πιο γνωστός εντοχιστρωτής για containers είναι ο Kubernetes. Αποτελείται από τουλάχιστον δύο(2) εικονικές δομές (VMs) που είναι επιφορτισμένες με τους ρόλους διαχειριστή (Master) και εκτελεστή (Worker). Ο κώδικας που είναι ενθυλακωμένος τρέχει σε Pods στους εκτελεστές που συνήθως είναι περισσότεροι από ένας ώστε να μπορούν να εφαρμοστούν αρχιτεκτονικές FailOver και LoadBalancing.

Η στρατηγική που ακολουθείτε για το SetUp της απαραίτητης υποδομής σχετίζεται με την σχεδιαστική-αρχή Infrastructure-as-Code (IaC). Είναι αποδεκτό από όλους ότι όσες φορές επαναλαμβάνεται «χειροκίνητα» μια διαδικασία τόσο αυξάνεται ο κίνδυνος σφάλματος. Αντίθετα αν η διαδικασία αυτοματοποιηθεί με τη συγγραφή κάποιων scripts που τρέχουν αυτούσια κάθε φορά τότε δεν υπάρχουν διαφοροποιήσεις στο τελικό αποτέλεσμα. Στην αρχή της προσπάθειας δεσμεύεται ένας Linux server που έχει εγκατεστημένο ένα Vagrant-engine (<https://www.vagrantup.com/>). Πρόκειται για ένα «δοκιμασμένο» εργαλείο για τη δημιουργία Εικονικών-Δομών (VMs) και τη ρύθμιση λεπτομερειών σχετικά με τη διαδικτυακή διασύνδεση τους αλλά και CPU/ Memory που θα αντιστοιχηθεί. Οι οδηγίες που γράφονται σε ένα text-αρχείο με την ονομασία Vagrantfile περιγράφουν τις κατασκευαστικές απαιτήσεις για το Master-Node και τουλάχιστον ένα Worker-Node.

Μετά την εκκίνηση των VMs σειρά έχει η εγκατάσταση του απαραίτητου λογισμικού ευρείας κλίμακας – όχι η υπηρεσία τεχνητής νοημοσύνης. Αυτό επιτυγχάνεται με ένα software-automation-tool που ονομάζεται Ansible (<https://www.ansible.com/>). Πάλι σε ένα text αρχείο γράφονται οδηγίες για την πλήρωση των VMs με βασικά εργαλεία όπως είναι το Docker-Engine και τα Kubernetes-Binaries.

Ο **Kubernetes** (<https://www.ansible.com/>) προσφέρει μια ολοκληρωμένη σουίτα για τη διαχείριση προγραμματιστικών έργων που είναι υλοποιημένα με τις αρχές των micro-services. Η καρδιά του συστήματος είναι ένας MASTER κόμβος που συντονίζει τη λειτουργία πολλών WORKER κόμβων με προτεραιότητα στους Containers που φιλοξενούν το κώδικα της υπηρεσίας. Στο έργο ανά χείρας έχει αυτοματοποιηθεί η εγκατάσταση του Kubernetes ενώ προσφέρεται η δυνατότητα στο διαχειριστή να επιλέξει τον κατάλληλο αριθμό Worker κόμβων σε πραγματικό χρόνο. Μετά την «εκτέλεση» ενός linux-script παραδίδεται ένα πλήρως λειτουργικό οικοσύστημα και σειρά έχει η διανομή του προγραμματιστικού έργου «Αναγνώρισης-Αντικειμένων».

Όπως έχει αναφερθεί η υλοποίηση της προσφερόμενης υπηρεσίας γίνεται με τη βοήθεια των Containers που ενθυλακώνουν το πηγαίο κώδικα και τις απαραίτητες βιβλιοθήκες. Δημιουργούνται πολλοί συνεργαζόμενοι containers που ο καθένας έχει ένα συγκεκριμένο και σαφή ρόλο και επικοινωνούν μεταξύ τους με Message-Queues. Το πρώτο βήμα για την εκκίνηση της υπηρεσίας είναι η δημιουργία ενός αποθηκευτικού χώρου για Containers (Private-Registry) στα όρια του Kubernetes. Στη συνέχεια αυτός ο χώρος θα «γεμίσει» με τους containers της υπηρεσίας που συντάσσονται από τους μηχανικούς ανάπτυξης και οι ώριμες εκδόσεις οδηγούνται στο Docker-Hub για εύκολη πρόσβαση. Άρα σε οποιαδήποτε εγκατάσταση του Tandem με διασύνδεση στο παγκόσμιο ιστό οι επικαιροποιημένοι

containers με τις τελευταίες εκδόσεις κώδικα μεταφέρονται στο τοπικό Private-Registry για διανομή στη συνέχεια στους Worker κόμβους. Σε περίπτωση που η εγκατάσταση του Tandem γίνεται σε απομονωμένο σημείο χωρίς διαδικτυακή σύνδεση ο μηχανισμός είναι εφοδιασμένος με μία ώριμη έκδοση των παρεχόμενων υπηρεσιών ώστε να είναι εγγυημένη η λειτουργικότητα σε όλες τις περιπτώσεις.

Στο τελευταίο στάδιο διατίθενται μία σειρά από Linux-Scripts που καλούνε κατάλληλα διαμορφωμένα αρχεία με οδηγίες για το Kubernetes για κάθε παρεχόμενη υπηρεσία. Στην ορολογία του συγκεκριμένου ενορχηστρωτή στο υψηλότερο επίπεδο υπάρχει το “Kind:Application” που αντιπροσωπεύει την υπηρεσία σε επίπεδο συναλλαγής με τελικό χρήστη. Είναι ένα υπερσύνολο που περιλαμβάνει όλες τις απαραίτητες οντότητες με κυριότερη την έννοια του Template/ Container όπου καθορίζονται οι συνεργαζόμενες προγραμματιστικές δομές μικρής κλίμακας. Για παράδειγμα στην υπηρεσία «Αναγνώρισης-Αντικειμένων» το σύνθετο έργο αναλύεται σε άλλα μικρότερα όπως είναι τα «διαχείριση σήματος video και εξαγωγή στιγμιότυπων», η «επεξεργασία εικόνας και μετατροπή από έγχρωμη σε κλίμακα-γκρι», η «δημιουργία detector για την αναγνώριση αντικειμένων με βάση κάποιο προ-εκπαιδευμένο μοντέλο», η «εφαρμογή detector σε άγνωστες εικόνες», ο «εντοπισμός κρίσιμων αντικειμένων με συμπληρωματική πληροφορία ακριβή θέση σε εικόνα», η «ενημέρωση υπόλοιπου συστήματος για τα ευρήματα από την επεξεργασία των στιγμιότυπων», και η «αντίδραση συστήματος στην ύπαρξη επικίνδυνων αντικειμένων για παράδειγμα αποστολή μηνυμάτων» ...

Όλα τα παραπάνω υπολογιστικά έργα ΔΕΝ τρέχουν σε ένα μεγάλο container αλλά διαμοιράζονται με την αρχή των μικρο-υπηρεσιών (microservices). Ο κάθε container κάνει ένα μικρό αλλά σαφές έργο και ο αντίστοιχος ορισμός του γίνεται στο παράρτημα Templates/spec του εκάστοτε application. Λειτουργικές παράμετροι που καθορίζονται σε αυτό το σημείο είναι όρια για δέσμευση σε CPU & RAM ώστε να μην υπάρξει υπερκατανάλωση και στο τέλος εξάντληση. Επίσης καθορίζεται τοποθεσία για μόνιμη αποθήκευση σε περίπτωση που υπάρχουν δεδομένα που πρέπει να έχουν πρόσβαση πολλοί containers. Για παράδειγμα το module που κάνει «επεξεργασία-εικόνας» και το συνεργαζόμενο «εντοπισμός-αντικειμένων με χρήση ευφυούς detector» επεξεργάζονται τα στιγμιότυπα που αναλύεται το βίντεο επιτήρησης. Κατά τη διάρκεια της σύστασης των αντίστοιχων containers ορίζεται αυτός ο κοινός χώρος που έχει διάρκεια ζωής μεγαλύτερη από τις επιμέρους εργασίες διαχείρισης. Παρέχονται δύο δυνατότητες στην αρχιτεκτονική του συστήματος αρχείων (File-System) με την απλούστερη να είναι η δέσμευση τοπικού χώρου στο Worker-Node που τρέχει το Pod με το εξεταζόμενο container. Για μεγαλύτερη ευελιξία υπάρχει η επιλογή του δικτυακού συστήματος αρχείων (Network File System) που παρέχεται η δυνατότητα δέσμευσης απομακρυσμένου χώρου σε άλλα worker-nodes ώστε αν υπάρχει αστοχία του συγκεκριμένου τα δεδομένα να επιβιώσουν.

Επειδή τα πρωτογενή δεδομένα είναι ευμεγέθη και υπάρχει ο κίνδυνος ταχείας εξάντλησης του διατιθέμενου αποθηκευτικού χώρου λειτουργούν ανεξάρτητοι μηχανισμοί ελέγχου ευρυθμίας. Ο ρόλος τους προσομοιάζει τους αντιπροσώπους (Agents) τηλεμετρίας που είναι επιφορτισμένοι με τη συλλογή και αρχική επεξεργασία λειτουργικών παραμέτρων. Τρέχουν σε ανεξάρτητους containers και διαθέτουν τους κανόνες αντίδρασης όταν παρουσιαστεί ένα πρόβλημα. Έτσι για παράδειγμα όταν συγκεντρωθεί μεγάλος αριθμός από στιγμιότυπα στην

«ουρά»-αναμονής για επεξεργασία και ο Worker-Node που φιλοξενεί το αποθηκευτικό σύστημα κινδυνεύει με αστοχία, τότε ο telemetry-agent αναλαμβάνει τη διαγραφή των παλαιότερων εικόνων. Ο προσανατολισμός του είναι ξεκάθαρα η ευρυθμία των υποδομών. Σε περίπτωση απλής συλλογής λειτουργικών μετρητικών χωρίς τον εντοπισμό προβλημάτων ο κάθε agent προωθεί σε ένα κεντρικό κομμάτι την τοπική πληροφορία με τελικό παραλήπτη κάποιο Prometheus ή Grafana Server.

Τα μηνύματα τηλεμετρίας ή τα αποτελέσματα εντοπισμού «κρίσιμων» αντικειμένων σε έναν επιβλεπόμενο χώρο μπορούν να μεταφερθούν σε άλλες προσφερόμενες υπηρεσίες στο ίδιο εννοχρηστωτή. Λειτουργεί όμως και συμπληρωματικός μηχανισμός ανταλλαγής μηνυμάτων που επιτρέπει την επικοινωνία με άλλο εννοχρηστωτή που «τρέχει» σε διαφορετικό γεωγραφικό σημείο. Υποστηρίζονται πολλαπλά πρωτόκολλα επικοινωνίας, όπως IoT/MQTT και AMQP, και για την μεταφορά των μηνυμάτων χρησιμοποιούνται brokers όπως RabbitMQ και Mosquitto.

3.3.2.4 Σήματα Εισόδου και Εξόδου στην Υπηρεσία Αναγνώρισης Αντικειμένων

Η εφαρμογή της υπηρεσίας αναγνώρισης αντικειμένων γίνεται σε δύο(2) στάδια με κύριο μέλημα την προετοιμασία ενός ευφυούς μοντέλου.

Στη πρώτη φάση επιλέγεται βιβλιοθήκη Μηχανικής-Μάθησης (ML) με πρωταρχικό στόχο το PyTorch. Είναι εξαιρετικής σημασίας να επιλεγεί μία έκδοση της βιβλιοθήκης που δεν παρουσιάζει ασυμβατότητες με την κάρτα γραφικών (GPU) του εξυπηρετητή.

Στη συνέχεια ο έλεγχος της υπηρεσίας μεταβαίνει στο κέντρο-συγκέντρωσης (HUB) ευφυών μοντέλων που αποτελεί αναπόσπαστο κομμάτι κάθε βιβλιοθήκης ML. Εκεί υπάρχουν δοκιμασμένες επιλογές που είναι προ-εκπαιδευμένες και ενδεδειγμένες δοκιμασμένες. Για την υλοποίηση της υπηρεσίας Αναγνώρισης-Αντικειμένων του TANDEM έχουν επιλεγεί το **Detectron2** της Facebook και το **Yolo5**.

Για την εφαρμογή των ευφυών μοντέλων πρέπει να δοθούν μερικές κατευθυντήριες οδηγίες:

Προσδιορισμός του σημείου εκτέλεσης των υπολογισμών. Πρόκειται για ένα μεγάλο όγκο εργασίας που μπορεί να προωθηθεί στη CPU αλλά η βέλτιστη λύση είναι να υπάρχει μία ισχυρή και εξελιγμένη κάρτα γραφικών GPU που θα αναλάβει αυτό το δύσκολο ρόλο και θα απελευθερώσει τη CPU για τις υπόλοιπες εργασίες του μηχανήματος.

Προσδιορισμός του βαθμού επισφάλειας στην αναγνώριση αντικειμένων. Σε κάθε έργο τεχνητής-νοημοσύνης οι τελικές αποφάσεις χαρακτηρίζονται από κάποιο βαθμό βεβαιότητας για την ορθότητα τους. Οι αλγόριθμοι ποτέ δεν είναι απόλυτα σίγουροι άρα υπάρχει ένα αποδεχτό περιθώριο σφάλματος. Ένα ευφυές μοντέλο που ορίζεται με ένα ποσοστό 80% σημαίνει ότι οποιαδήποτε εκτίμηση έχει πιθανότητα σφάλματος μικρότερη του 20% θα επιλεγεί σαν σωστή.

Με το πέρας του ορισμού της αρχιτεκτονικής του μοντέλου αναγνώρισης και των βασικών παραμέτρων λειτουργίας του τίθεται σε ισχύ. Σε αυτό το σημείο ολοκληρώνεται η πρώτη φάση.

Στη δεύτερη φάση της αναγνώρισης αντικειμένων διακινούνται φωτογραφίες που είναι ζητούμενο η σύνθεση τους - για παράδειγμα αν υπάρχουν αυτοκίνητα ή άνθρωποι:

- Το επιλεγμένο ευφυές μοντέλο επεξεργάζεται μία σειρά από φωτογραφίες
- Εξάγει μια λίστα από αντικείμενα που μπόρεσε να αναγνωρίσει με αποδεκτό ποσοστό βεβαιότητας
- Η υπηρεσία επιλέγει τα εκάστοτε αντικείμενα που είναι το ζητούμενο σε κάθε περίπτωση. Η ταυτότητα εύρεσης που προωθείται στα υψηλότερα επίπεδα του TANDEM αποτελείται από { τύπος/ label, θέση/ bounding-box, probability}

3.4.3 Frame Grabber

Με γνώμονα το διαχωρισμό των υπηρεσιών σε στρατηγικές (βασικές) και υποστηρικτικές η συγκεκριμένα κατατάσσεται στη δεύτερη κατηγορία. Δεν προσφέρει μία λειτουργικότητα που φτάνει στο τελικό χρήστη αλλά είναι απαραίτητη η ευρυθμία της για να προσφέρεται από τη πλατφόρμα TANDEM η δυνατότητα αναγνώρισης αντικειμένων όπως αυτοκίνητα και άνθρωποι.

Η είσοδος (input) που δέχεται ο Frame Grabber είναι το συνεχές-σήμα βίντεο από μία ή περισσότερες κάμερες.

Ο κώδικας είναι συμβατός με το RTSP πρωτόκολλο, αρχικά από τις λέξεις Real-Time-Streaming-Protocol και υποστηρίζεται από πολλούς κατασκευαστές καμερών. Το αποτέλεσμα είναι να υπάρχει συνεργασία με τις περισσότερες υπάρχουσες υποδομές και να μην απαιτείται η αγορά καινούργιου ακριβού εξοπλισμού, όπως είναι οι κάμερες.

Η έξοδος αυτού του υποστηρικτικού module είναι μία σειρά από στιγμιότυπα που απεικονίζουν τη οπτική εξέλιξη ενός γεγονότος, όπως είναι η κινητικότητα σε ένα χώρο πάρκινγκ. Συγκεκριμένα υπάρχει ένα folder όπου αποθηκεύονται οι διαδοχικές εικόνες για να μπορέσουν να τις επεξεργαστούν άλλα “κομμάτια” όπως είναι η ευφυής αναγνώριση.

Επειδή η αποθήκευση εικόνων συνεπάγεται γρήγορη κατανάλωση χώρου σε σκληρούς δίσκους, ελέγχεται από ένα monitoring μηχανισμό. Τρέχει ένα “ρολόι” που ανά τακτά χρονικά διαστήματα μετράει τον “ελεύθερο” χώρο. Σε περίπτωση που υπάρχει ικανοποιητική διαθεσιμότητα τότε μπαίνει σε κατάσταση ύπνου. Διαφορετικά επιλαμβάνεται του προβλήματος και προχωράει στη διαγραφή των παλαιότερων αρχείων.

3.5 TANDEM App services

3.5.1 Αυτοματοποιημένη διαδικασία εγκατάστασης

Τόση η ίδια η πλατφόρμα TANDEM όσο και οι προφερόμενες υπηρεσίες υλοποιούνται σε Containers και ακολουθούν τις σχεδιαστικές αρχές των Micro-Services. Χρησιμοποιούνται μια σειρά από εργαλεία με κυρίαρχο τον ενορχηστρωτή Kubernetes.

Η εγκατάσταση αυτής της υποδομής είναι πολύ-βηματική και εξαιρετικά χρονοβόρα. Απαιτεί εξειδικευμένες γνώσεις και αποτελεί ένα σημαντικό εμπόδιο για τη προιοντική διανομή της πλατφόρμας.

Με γνώμονα την αυτοματοποίηση της εγκατάστασης που θα απομακρύνει την ανάγκη εξειδικευμένου προσωπικού αλλά ταυτόχρονα θα ελαχιστοποιήσει τη πιθανότητα σφάλματος ακολουθείται η αρχή του IaasC, δηλαδή Infrastructure as a Code. Γράφονται οδηγίες σε κατάλληλα αρχεία που διαχειρίζονται όλα τα ζητούμενα ξεκινώντας από τη φάση του virtualization και φτάνοντας στη διανομή των υπηρεσιών TANDEM.

Ο ενδιαφερόμενος στη πλατφόρμα αρκεί να προσέλθει με ένα αριθμό εξυπηρετητών, σε περιβάλλον Linux. Η διαδικασία αυτοματισμού το πρώτο πράγμα που θα κάνει είναι η εγκατάσταση μίας virtualization-engine, στη προκειμένη περίπτωση του Oracle Virtual Box (<https://www.virtualbox.org/>). Με τη βοήθεια αυτού του εργαλείου ο φυσικός εξοπλισμός μπορεί να λάβει εικονικοποιημένη υπόσταση και να επιτευχθεί μεγαλύτερη αξιοποίηση των δυνατοτήτων του.

Το δεύτερο εργαλείο που θα εγκατασταθεί είναι μία μηχανή σύνταξης οδηγιών για τη δημιουργία εικονικοποιημένων υπολογιστικών συστημάτων. Το προαναφερθέν Oracle/VirtualBox είναι σε θέση να δημιουργήσει VMs. Το πρόβλημα είναι ότι μία πλατφόρμα με ενорχηστρωτή και προσφερόμενες υπηρεσίες απαιτεί ένα μεγάλο αριθμό από συνεργαζόμενα VMs. Μία “χειροκίνητη” εγκατάσταση απαιτεί εξειδικευμένους διαχειριστές συστημάτων και κρύβει μεγάλες πιθανότητες σφάλματος. Η λύση είναι να μπουν όλες οι οδηγίες δημιουργίας των VMs σε ένα αρχείο και σε κάθε εγκατάσταση να τρέχει αυτότισιο. Για αυτό το σκοπό έχει επιλεγεί το HashiCorp/ Vagrant (<https://www.vagrantup.com/>). Στα αρχεία που έχουν συνταχθεί βρίσκονται αναλυτικές οδηγίες για αποδιδόμενη CPU και RAM σε κάθε VM αλλά και δικτυακή διασύνδεση μεταξύ των VM αλλά και με το φυσικό μηχάνημα που τα φιλοξενεί.

Τα VMs είναι μικροί εξυπηρετητές που το κύριο μέρος της λειτουργικότητας τους το οφείλουν στο software που υπάρχει εγκατεστημένο και τα κάνει χρήσιμα για κάποιες εργασίες. Για παράδειγμα η εγκατάσταση μίας βάσης δεδομένων ώστε να μπορούν να αποθηκεύονται δεδομένα σε βάθος χρόνου.

Το τρίτο εργαλείο που θα χρησιμοποιηθεί για τον αυτοματισμό της εγκατάστασης της πλατφόρμας TANDEM είναι το RedHat/ Ansible (<https://www.ansible.com/>). Πρόκειται για μία μηχανή που κατανοεί οδηγίες για ενημέρωση VMs με συγκεκριμένο software που στη προκειμένη περίπτωση έχει μεγάλη σημασία η εγκατάσταση του Container-Engine και των binaries του Kubernetes. Έχουν γραφεί αρχεία που αυτοματοποιούν τη διαδικασία που παίρνει απλά VMs και τα μετατρέπει σε Kubernetes Master και Worker Nodes.

Τέλος έχει απλοποιηθεί και η διανομή των προσφερόμενων υπηρεσιών TANDEM. Πρόκειται για οδηγίες που απευθύνονται στον Kubertes και διαχειρίζονται από τα πιο απλά μέχρι τα πιο σύνθετα προβλήματα όπως είναι το fail-over και load-balancing.

Ένα σημαντικό μέρος από όλα αυτά τα αρχεία αυτοματισμού παρατίθεται στο παράρτημα.

3.5.2 Υπηρεσία Ευφυούς Στάθμευσης (Smart Parking Service)

Επιγραμματικά η υπηρεσία αυτή:

- Ζητάει από την υπηρεσία Αναγνώρισης Αντικειμένων να την ενημερώνει για τα αυτοκίνητα που εντοπίζει σε στιγμιότυπα (frames) που εξάγονται από ροές βίντεο
- Για κάθε ένα αυτοκίνητο η υπηρεσία ευφυούς-πάρκινγκ επικεντρώνει τη προσοχή της στο σημείο της εικόνας που εντοπίστηκε
- Η θέση εύρεσης συσχετίζεται με τις θέσεις παρκαρίσματος.
- Στο χάρτη αποτύπωσης διαθεσιμότητας η θέση που βρίσκεται εγγύτερα στο αυτοκίνητο και υπάρχει μία μερική επικάλυψη δηλώνεται σαν δεσμευμένη
- Ενημερώνονται τα ποσοστά πληρότητας του συγκεκριμένου χώρου στάθμευσης.
- Η πληροφορία που εξήχθει σε τοπικό επίπεδο μετά την προσέγγιση και κατάληψη μίας άδειας θέσης από ένα καινούργιο αυτοκίνητο μεταφέρεται κεντρικά για την ενημέρωση της υπηρεσίας ολιστικής διαχείρισης όλων των χώρων στάθμευσης.

Η συγκεκριμένη υπηρεσία έχει εξαιρετικές προοπτικές στο χώρο της Έξυπνης Πόλης (Smart City) καθώς και στα συστήματα C-ITS (Cooperative Intelligent Transport Systems) όπου η πληροφορία μεταφέρεται από το σημείο που δημιουργείται στους προσερχόμενους οδηγούς για να τους βοηθήσει σε "Εύκολη Στάθμευση".

Είναι μία υπηρεσία Στρατηγικής σημασίας για το TANDEM αφού επιτρέπει σε μία μεγάλη μάζα χρηστών να κινηθούν με άνεση και ασφάλεια. Η λειτουργία της είναι άμεσα συνυφασμένη με την υπηρεσία Αναγνώρισης Αντικειμένων:

- Σε χαμηλό επίπεδο τρέχει ο μηχανισμός που αναλύει διαδοχικά στιγμιότυπα από ένα βίντεο επιτήρησης (έχει αναλυθεί στην ενότητα 3.3)
- Τα αποτελέσματα δηλαδή {label, position, probability} μπαίνουν σε ένα δίαυλο επικοινωνίας (για παράδειγμα RabbitMQ) με την μέθοδο του publisher
- Η εν λόγω υπηρεσία παρακολουθεί το ίδιο κανάλι επικοινωνίας αλλά σαν subscriber δηλαδή λαμβάνει ειδοποιήσεις για κάθε καινούργιο αντικείμενο που εντοπίστηκε
- Επιλέγει μόνο τα αντικείμενα που την ενδιαφέρουν δηλαδή σταθμευμένα αυτοκίνητα
- Για κάθε ένα γνωρίζει την ακόλουθη πληροφορία ... βρέθηκε ένα σε ένα χώρο της εικόνας που ορίζεται από τις ακόλουθες συντεταγμένες {Κορυφή#1 - Κορυφή#2 - Κορυφή#3 -Κορυφή#4}
- Επίσης η υπηρεσία είναι κάτοχος της χαρτογράφησης του συνολικού χώρου .. δηλαδή η θέση P1 ορίζεται επίσης από μία τετράδα κορυφών.
- Γίνεται αντιπαραβολή της θέσης του αντικειμένου με αυτές του πάρκινγκ οπότε η υπηρεσία γνωρίζει ότι οι P-i όπου $i=1..k$ είναι δεσμευμένες ενώ οι υπόλοιπες P-J είναι διαθέσιμες για επερχόμενους οδηγούς



Εικόνα 12 - Δοκιμαστική Χρήση της Υπηρεσίας Smart Parking στο parking του κτιρίου A5 της INTRACOM TELECOM

Η Εικόνα 12 δείχνει τη δοκιμαστική χρήση της υπηρεσίας Smart Parking στο parking του κτιρίου A5 της INTRACOM TELECOM. Αποτελείται από δύο επίπεδα (layers). Στο παρασκήνιο υπάρχει ένα στιγμιότυπο από το χώρο ελεγχόμενης στάθμευσης και έχει αποτυπωθεί με IP-κάμερα, ενώ στο προσκήνιο έχουν τοποθετηθεί μερικές γραφικές λεπτομέρειες που απεικονίζουν τα αποτελέσματα του ευφυούς μοντέλου αναγνώρισης αυτοκινήτων.

Συγκεκριμένα με "πράσινο" περίγραμμα εμφανίζονται οι ΕΛΕΥΘΕΡΕΣ θέσεις παρκαρίσματος ενώ με κόκκινο οι κατειλημμένες. Επίσης κάθε αντικείμενο που αναγνωρίζεται αναγράφεται ο τίτλος του για παράδειγμα "ΑΝΘΡΩΠΟΣ".

Τέλος στο πάνω-δεξιά άκρο της εικόνας έχει παρεμβληθεί ένα κείμενο με "απολογιστικές" πληροφορίες, όπως το σύνολο των ελεύθερων θέσεων.

3.5.3 Υπηρεσία Επόπτευσης & Ασφάλειας Χώρου (Surveillance & Security Service)

Μία δεύτερη κατηγορία υπηρεσιών του TANDEM στρατηγικής σημασίας έχει να κάνει με ζητήματα ασφάλειας και ελέγχου χώρου. Επιγραμματικά η υπηρεσία αυτή:

- Είναι συνδεδεμένη με τη στρατηγική υπηρεσία της Αναγνώρισης-Αντικειμένων στην οποία έχει δηλώσει το ενδιαφέρον της για τις κατηγορίες "πινακίδες-κυκλοφορίας", "άνθρωπος", "αποσκευές", "σακίδια", κοκ.
- Ενημερώνεται αυτόματα για εμφανίσεις καθενός από τα "ενδιαφέροντα" αντικείμενα σε στιγμιότυπα από τον επιβλεπόμενο χώρο
- Ανάλογα με το αντικείμενο εφαρμόζει μία συγκεκριμένη λογική διαχείρισης. Για παράδειγμα σε πινακίδες-κυκλοφορίας η υπηρεσία ευφυούς-ελέγχου επικεντρώνει τη προσοχή της στα αριθμητικά στοιχεία της πινακίδας (οι λατινικοί χαρακτήρες έχουν δευτερεύοντα ρόλο). Μετά την εφαρμογή αλγορίθμων OCR (Optical Character Recognition) τα νούμερα διαβάζονται.

- Τα νούμερα της τρέχουσας πινακίδας ελέγχονται με τη λίστα των επιτρεπόμενων οχημάτων -- αυτή η λίστα εισέρχεται στη πλατφόρμα TANDEM κάθε μέρα.
- Εάν το αυτοκίνητο δεν είναι εξουσιοδοτημένο η υπηρεσία ασφάλειας στέλνει επείγουσες ειδοποιήσεις μέσω κατάλληλων-modules που είναι επιφορτισμένα με τη διαχείριση emails & sms
- Επίσης η υπηρεσία στέλνει εντολές εκκίνησης λειτουργίας σε αποτρεπτικούς μηχανισμούς όπως σειρήνες/ συναγερμούς.

Συγκεκριμένο προσφέρεται η δυνατότητα στο τελικό χρήστη για προκαταβολική ενημέρωση στη περίπτωση μη-εξουσιοδοτημένων αντικειμένων όπως:

- Ανάγνωση πινακίδων κυκλοφορίας αυτοκινήτων και Ειδοποίηση για μη-προβλεπόμενη είσοδο και
- Εντοπισμό εγκαταλειμμένων αντικειμένων όπως σακιδία.

Η λειτουργία των υπηρεσιών-ασφάλειας είναι άρρηκτα δεμένη με την θεμελιώδη-υπηρεσία Αναγνώρισης-Αντικειμένων - εκτελείται στο παρασκήνιο αλλά είναι προαπαιτούμενο για την ευφυή συμπεριφορά της πλατφόρμας.

Η είσοδος στις υπηρεσίες ασφάλειας είναι μια λίστα από αντικείμενα που εντοπίστηκαν σε μία εικόνα. Συγκεκριμένα υπάρχει ένας δίαυλος επικοινωνίας (message-queue/ RabbitMQ) όπου η μονάδα αναγνώρισης με το πέρας επεξεργασίας μίας άγνωστης εικόνας γνωστοποιεί τα ευρήματα της. Στον ίδιο δίαυλο και στο ίδιο κανάλι όπου γίνονται οι ανακοινώσεις (για παράδειγμα το κανάλι με ονομασία πινακίδες-αυτοκινήτων) είναι δηλωμένος σαν παρατηρητής “subscriber” η υπηρεσία-ασφάλειας. Με αυτό το τρόπο κάθε φορά που εμφανίζεται καινούργιο μήνυμα για εντοπισμό νέου αντικειμένου ενημερώνεται αυτόματα.

Η επεξεργασία που κάνει η υπηρεσία ασφάλειας αποσκοπεί στην έγκαιρη κινητοποίηση σε περιπτώσεις παρείσακτων. Έτσι όταν δημοσιοποιηθεί η ύπαρξη μίας εισερχόμενης πινακίδας αυτοκινήτου συγκρίνει το περιεχόμενο της με μία λίστα από επιτρεπόμενα οχήματα που ενημερώνεται καθημερινά από την αρμόδια υπηρεσία.

Στην περίπτωση που δεν υπάρχει εξουσιοδότηση η υπηρεσία ασφάλειας εναποθέτει στο δίαυλο-επικοινωνίας και στο κανάλι “επείγουσες ανακοινώσεις” ένα ενημερωτικό μήνυμα. Το μήνυμα αυτό θα καταλήξει στην υπηρεσία-ειδοποιήσεων που με τη σειρά της θα συντάξει sms και emails για το εξουσιοδοτημένο προσωπικό.

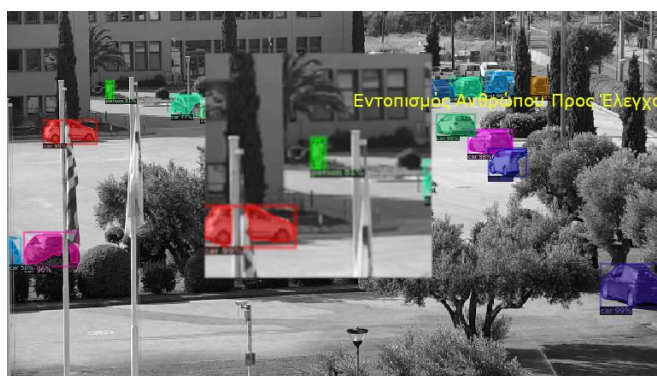
Τέλος σε περιπτώσεις που η παραβατικότητα κρίνεται μεγάλης σημασίας και επικινδυνότητας, η υπηρεσία-ασφάλειας θα κινητοποιήσει μηχανισμούς αντίδρασης. Για παράδειγμα θα στείλει START μήνυμα σε μία σειρήνα συναγερμού.

Στη συνέχεια παρατίθενται μερικές εικόνες που απεικονίζουν εύγλωττα τα κυρίαρχα σενάρια χρήσης στο πεδίο της “Επόπτευσης κ Ασφάλειας”.

Σε ένα χώρο ελέγχου τοποθετείται μία κάμερα ασφάλειας σε ένα ψηλό σημείο ώστε να έχει πανοραμική θέα και να παρατηρεί σε 24ωρη βάση. Όπως φαίνεται ο χώρος είναι “γεμάτος” με αντικείμενα εκ των οποίων άλλα είναι ενδιαφέροντα και πολλά απλά έχουν φυσική

παρουσία. Η Υπηρεσία-Ασφάλειας βρίσκεται σε κατάσταση διαρκούς επόπτευσης μέχρι τη χρονική στιγμή που εντοπίζεται ένας άνθρωπος. Τονίζεται η ικανότητα του ευφυούς-μοντέλου να διακρίνει σε πραγματικά σενάρια όπου ο άνθρωπος είναι εν μέρη-ορατός κινείται ανάμεσα σε άλλα αντικείμενα και βρίσκεται μακριά από το σημείο έδρασης της κάμερας (Εικόνα 13).

Μόλις εντοπιστεί ο άνθρωπος η υπηρεσία εισέρχεται σε κατάσταση Ασφάλειας με αναφορά σε κάποια πρωτόκολλα μη-εξουσιοδοτημένων ενεργειών. Για παράδειγμα αν ο εντοπισμός γίνεται σε κάποια χρονική στιγμή που δεν επιτρέπεται η παρουσία κανενός τότε στέλνεται ειδοποίηση σε μία κεντρική υπηρεσία για επιβεβαίωση και παρέμβαση.



Εικόνα 13-Δοκιμαστική Χρήση της Υπηρεσίας Επόπτευσης και Ασφάλειας στον περιβάλλον χώρο της εταιρείας INTRACOM TELECOM –Εντοπισμός Ανθρώπου

Σε ένα άλλο σενάριο (Εικόνα 14) ελέγχονται οι πινακίδες κυκλοφορίας των προσερχόμενων αυτοκινήτων σε μία κρίσιμη υποδομή. Η υπηρεσία ξεκινάει με τον εντοπισμό της πινακίδας στο μπροστινό μέρος του οχήματος. Στη συνέχεια γίνεται διαχωρισμός των λατινικών χαρακτήρων από τους αριθμητικούς που αποτελούν και τη κρίσιμη πληροφορία για την αναγνώριση του άγνωστου αυτοκινήτου. Στο τελευταίο στάδιο τα νούμερα από τη πινακίδα κυκλοφορίας αντιπαραβάλλονται με μία λίστα που συντάσσεται κεντρικά και περιλαμβάνει τα εξουσιοδοτημένα.

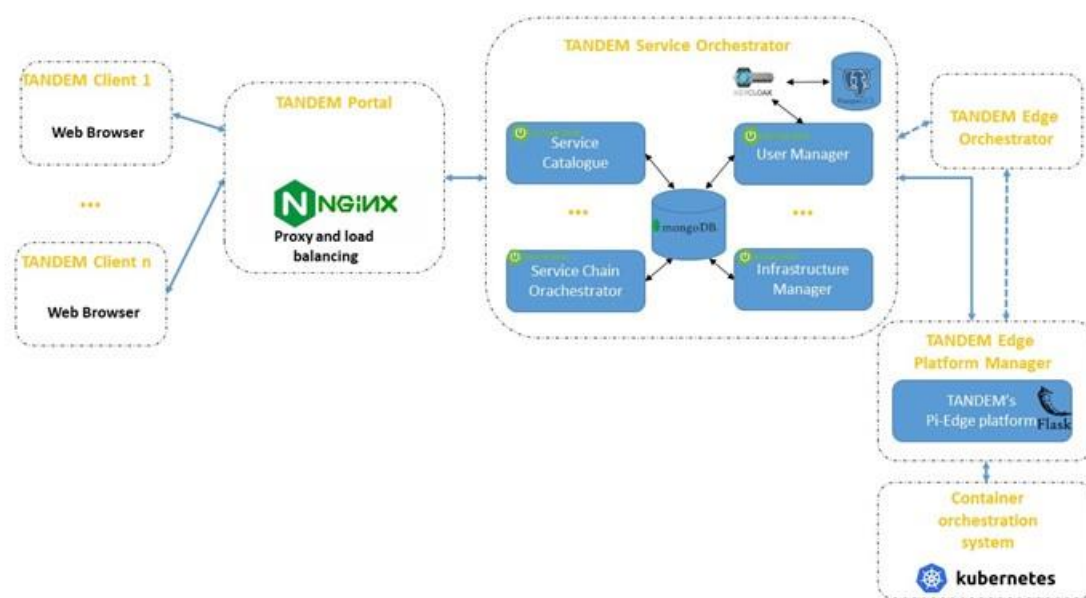


Εικόνα 14 -Αυτόματη Ανάγνωση Πινακίδας Οχήματος για έλεγχο είσοδου -Εγκαταστάσεις INTRACOM TELECOM

4 Υλοποίηση Λογισμικού Διαχειριστή Συστήματος TANDEM

Η Εικόνα 15 παρουσιάζει τα Βασικά Συστατικά του Διαχειριστή Συστήματος TANDEM, την επικοινωνία μεταξύ τους και την υλοποίησης τους στην 2^η έκδοση. Οι χρήστες χρησιμοποιούν web browser για να έχουν πρόσβαση μέσω του TANDEM portal σε λειτουργίες του TANDEM που παρέχονται από τον **Service Orchestrator (Ενορχηστρωτή Υπηρεσιών)**. Ο Service Orchestrator παρέχει λειτουργίες όπως κατάλογος υπηρεσιών, δημιουργία αλυσίδων υπηρεσιών, διαχείριση χρηστών και κεντρική Διαχείριση Υποδομής. Ο Edge Orchestrator (Ενορχηστρωτής Άκρων) που διαχειρίζεται την υποδομή, τους πόρους και την λειτουργία των υπηρεσιών όλων των edge clouds επικοινωνώντας με τον TANDEM Edge Platform Manager καθενός από αυτούς στη δεύτερη έκδοση του λογισμικού σύμφωνα με τον προγραμματισμό υποστηρίζει μόνο ένα edge cloud (cluster). Για τη Διαχείριση Υποδομής, ο Service Orchestrator επικοινωνεί με τον TANDEM Edge Platform Manager που διαχειρίζεται το συγκεκριμένο edge cloud του δεύτερου πρωτοτύπου (για κάποιες λειτουργίες μέσω της αρχικής έκδοσης του Edge Orchestrator).

Στα παρακάτω υποκεφάλαια παρουσιάζεται αναλυτικά η υλοποίηση των Συστατικών του Διαχειριστή Συστήματος TANDEM για το πρώτο πρωτότυπο.



Εικόνα 15: Η Υλοποίηση του Διαχειριστή Συστήματος TANDEM

4.1 TANDEM Portal

Σε αυτή την ενότητα περιγράφεται η υλοποίηση του **TANDEM Service Portal (Πύλη Υπηρεσιών TANDEM)** που είναι το μοναδικό σημείο εισόδου σε τρίτους σε λειτουργίες του TANDEM.

Το Portal δίνει πρόσβαση στις παρακάτω λειτουργίες

1. Έλεγχος της λειτουργίας του TANDEM Edge Orchestrator. Παροχή μιας πιο υψηλού επιπέδου διεπαφής στις λειτουργίες του Edge Orchestrator.

2. Διατήρηση συνολικής εικόνας του συστήματος άκρων (edge system) και των υπολογιστών του, των διαθέσιμων πόρων του, των διαθέσιμων υπηρεσιών του και της τοπολογίας του
3. Διαχείριση καταλόγου των υπηρεσιών που παρέχονται στους υπολογιστές άκρων καθώς και στοιχείων χρέωσης τους
4. Διαχείριση χρηστών και των δικαιωμάτων τους
5. Διαχείριση μοντέλων τιμολόγησης, στοιχείων χρέωσης και πληροφοριών SLA
6. Διαχείριση συνδεδεμένων συσκευών
7. Υποστήριξη εργαλείου ροής εργασίας (workflow engine) για τη δημιουργία σύνθετων υπηρεσιών με τη δημιουργία αλυσίδων υπηρεσιών (Service Chain Orchestrator)
8. Έλεγχος κύκλου ζωής υπηρεσιών και εφαρμογών,

4.1.1 Σχεδιασμός

Η εφαρμογή χρησιμοποιεί ένα σύνολο από τεχνολογίες που επιτρέπουν την παροχή αξιόπιστης και υψηλής ποιότητας εμπειρία (QoE) στον τελικό χρήστη. Παρακάτω αναφέρονται τα κυριότερα τεχνικά χαρακτηριστικά:

Για τη σχεδίαση και τη προσθήκη λειτουργικότητας στην πλευρά του περιηγητή του τελικού χρήστη, γίνεται χρήση των γλωσσών προγραμματισμού HTML5+, CSS και JavaScript καθώς και βιβλιοθηκών όπως **Bootstrap [Bootstrap]** και **jQuery [jquery]**

- Το **Bootstrap** αποτελεί μια πλήρη και ευρύτατα διαδεδομένη δομή ανοικτού κώδικα, που παρέχει έτοιμες και παραμετροποιήσιμες ρυθμίσεις αναπαράστασης στοιχείων στον τελικό χρήστη με χρήση της γλώσσα CSS. Κύριο μέλημα της δομής είναι η παροχή μιας συλλογής έτοιμων στοιχείων που προσαρμόζονται δυναμικά σε πολλές οθόνες ανεξαρτήτου μεγέθους και εφαρμογής περιήγησης.
- Η **jQuery** είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού με χαρακτηριστικά:
 - 1 DOM element επιλογές χρησιμοποιώντας την ανοικτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών Sizzle.
 - 2 χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
 - 3 Events
 - 4 AJAX
 - 5 Εργαλεία όπως πληροφορίες user-agent, ανίχνευση χαρακτηριστικών.

Το **Nginx [Nginx]** χρησιμοποιείται ως διακομιστής εξυπηρετώντας απευθείας αιτήματα στατικών αρχείων (application server).

4.2 Service Orchestrator (Ενορχηστρωτής Υπηρεσιών)

Ο **Service Orchestrator (Ενορχηστρωτής Υπηρεσιών)** διαχειρίζεται όλες τις λειτουργίες που αφορούν τις παρεχόμενες υπηρεσίες στους πελάτες του συστήματος.

Οι πελάτες έχουν πρόσβαση σε αυτόν μέσω του **TANDEM Service Portal (Πύλη Υπηρεσιών TANDEM)** που είναι το μοναδικό σημείο εισόδου σε τρίτους σε λειτουργίες του TANDEM. Μέσω του Service Orchestrator οι πελάτες μπορούν να χρησιμοποιήσουν κάποια υπηρεσία,

να δουν τις παρεχόμενες υπηρεσίες και την τιμολόγησή τους και να πάρουν πληροφορίες για τις υπηρεσίες που χρησιμοποιούν και την τρέχουσα χρέωσή τους.

Ο Διαχειριστής του Συστήματος χρησιμοποιεί τον **Service Orchestrator** για να διαχειρίζεται κεντρικά τις υπηρεσίες του TANDEM και τους αντίστοιχους πόρους.

Οι υπηρεσίες του Ενορχηστρωτή Υπηρεσιών που υλοποιήθηκαν για τη δεύτερη έκδοση λογισμικού είναι ο **Κατάλογος Υπηρεσιών (Service Catalogue)**, ο **Διαχειριστής Αλυσίδων Υπηρεσιών (Service Chain Manager)** που είναι τμήμα του **Ενορχηστρωτή Αλυσίδων Υπηρεσιών (Service Chain Orchestrator – SCO)**, ο **Διαχειριστής Χρηστών (User Manager)** και ο **Διαχειριστής Υποδομής (Infrastructure Manager)**.

4.2.1 Σχεδιασμός

Η εφαρμογή χρησιμοποιεί ένα σύνολο από τεχνολογίες που επιτρέπουν την παροχή αξιόπιστης και υψηλής ποιότητας εμπειρία (QoE) στον τελικό χρήστη. Παρακάτω αναφέρονται τα κυριότερα τεχνικά χαρακτηριστικά:

1. Ο πυρήνας της εφαρμογής είναι υλοποιημένος στην γλώσσα προγραμματισμού Java με τη βοήθεια του πλαισίου ανάπτυξη διαδικτυακών εφαρμογών **Spring Boot [Spring Boot]**.
2. Η **MongoDB [MongoDB]** είναι το επιλεγμένο Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS), στο οποίο αποθηκεύονται όλα τα απαιτούμενα δεδομένα σε μη συσχετισμένη(non-relational) μορφή.
3. Το **Nginx** χρησιμοποιείται ως αντίστροφος διακομιστής μεσολάβησης (reverse proxy) και εξισορρόπησης φορτίου αιτημάτων. Προσφέρει λειτουργίες όπως ο έλεγχος πρόσβασης τελικών χρηστών μέσω ψηφιακών πιστοποιητικών, υποστηρίζει το πρωτόκολλο HTTP2 και αποσυμφορίζει την κύρια εφαρμογή του διακομιστή εξυπηρετώντας απευθείας αιτήματα στατικών αρχείων
4. Το deployment όλων των παραπάνω επιτυγχάνεται με τη χρήση του **Kubernetes** ως ενορχηστρωτή (orchestrator). Η χρήση του ενορχηστρωτή επιτρέπει τη συνολική εποπτεία του συστήματος σε περιβάλλον παραγωγής καθώς και την δημιουργία αντιγράφων των εγκατεστημένων εφαρμογών ώστε να επιτυγχάνεται η εξυπηρέτηση μεγαλύτερου όγκου αιτημάτων από το διακομιστή εάν αυτό χρειαστεί (scale up).

4.2.2 Εργαλεία

Σχετικά εργαλεία που χρησιμοποιήθηκαν:

- Το *Spring Framework* είναι ένα δημοφιλές, ανοιχτού κώδικα πλαίσιο εταιρικού επιπέδου για τη δημιουργία αυτόνομων εφαρμογών ποιότητας παραγωγής που εκτελούνται στην **Java Virtual Machine (JVM)**. Το Spring Boot είναι ένα εργαλείο που κάνει την ανάπτυξη διαδικτυακών εφαρμογών και μικροϋπηρεσιών (microservices) με το Spring Framework ταχύτερη και ευκολότερη μιας και είναι εύκολα κατανοητός ο τρόπος λειτουργίας και χρήσης του. Επιτρέπει τη δημιουργία μιας εφαρμογής που βασίζεται στο Spring με σημαντικά λιγότερες ρυθμίσεις.

- Η *MongoDB* είναι μια μη σχεσιακή βάση δεδομένων ανοιχτού κώδικα, βασισμένη σε έγγραφα. Οι μη σχεσιακές βάσεις δεδομένων είναι πολύ χρήσιμες για εργασία με μεγάλα σύνολα κατανεμημένων δεδομένων. Το MongoDB είναι ένα εργαλείο που μπορεί να διαχειριστεί πληροφορίες προσανατολισμένες σε έγγραφα (document oriented), να αποθηκεύσει ή να ανακτήσει πληροφορίες. Γραμμένη σε C++, η MongoDB χρησιμοποιεί έγγραφα τύπου JSON με προαιρετικά σχήματα. Οι οργανισμοί συνήθως επιλέγουν τη Mongo DB για τα ad-hoc ερωτήματα, τη δημιουργία ευρετηρίου, την εξισορρόπηση φόρτου, τη συγκέντρωση, την εκτέλεση JavaScript από την πλευρά του διακομιστή και άλλες δυνατότητες.
- Το *Nginx* είναι μια εφαρμογή ανοιχτού κώδικα, BSD2-clause. Λειτουργεί ως διακομιστής που χρησιμοποιείται κυρίως ως εξισορροπιστής φόρτου αιτημάτων και αντίστροφος μεσολαβητής με στόχο της βελτίωση απόδοσης του διακομιστή. Υποστηρίζει ασύγχρονη επικοινωνία σε επίπεδο αιτήματος και τελικού χρήστη, γεγονός που του επιτρέπει να εξυπηρετεί ταυτόχρονα μεγαλύτερο αριθμό αιτημάτων σε σχέση με σύγχρονους διακομιστές χωρίς να απαιτούνται περισσότεροι πόροι. Το TANDEM χρησιμοποιεί το Nginx ως αντίστροφο μεσολαβητή διακομιστή που φιλτράρει τα αιτήματα προς το διακομιστή, επιτρέποντας την επικοινωνία μόνο σε τελικούς χρήστες που φέρουν έγκυρο πιστοποιητικό, ενώ η σύνδεση γίνεται μέσω διαύλου επικοινωνίας που χρησιμοποιεί κωδικοποίηση TLS και σε περιπτώσεις που υποστηρίζεται από τον περιηγητή του τελικού χρήστη χρησιμοποιείται το πρωτόκολλο HTTP2 για τη ταχύτερη μεταφορά δεδομένων.

4.2.3 Υλοποίηση Καταλόγου Υπηρεσιών (Service Catalogue)

Η εφαρμογή παρέχει τα κατάλληλα end points προς τη διεπαφή ανθρώπου-μηχανής, επιτρέποντας στον χρήστη να αποθηκεύει την πληροφορία για τα παρεχόμενα από αυτόν services και προϊόντα (products), καθώς και να πραγματοποιεί αναζήτηση στα ήδη παρεχόμενα από το TANDEM services. Ειδικότερα, παρέχονται οι ακόλουθες λειτουργίες:

- Δήλωση νέων υπηρεσιών και αποθήκευση των στοιχείων τους, μέσω της αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/create/service
```

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{
  "serId": "TANDEM_ser2",
  "serName": "Notification Service",
  "serType": "FaaSTest",
  "serProvider": "INTRACOM TELECOM S.A.",
  "serDescription": "The user defines the sender, the recipients and the text of the e-
```

```

mail message and the service responds whether the sending was successful or not",
  "serCategory": {
    "href": "Notification",
    "id": "serCat2",
    "name": "Notification Services",
    "version": "001"
  },
  "serVersion": "001",
  "state": "In TANDEM Test",
  "serAPIDescriptionURL": [
    "tandem-project.gr/Services/MonService/",
    "edgeservices.org/TandemMonService/"
  ],
  "serConfigParams": [
    {
      "serParamName": "smtp_server_port",
      "serParamType": "string",
      "serParamValue": "smtp.tandem.com:587",
      "serParamDescr": "e-mail server address and port",
      "serParamTypicalValue": "string"
    },
    {
      "serParamName": "password",
      "serParamType": "string",
      "serParamValue": "string",
      "serParamDescr": "sender password",
      "serParamTypicalValue": "string"
    }
  ],
  "serOperations": [
    {
      "serOperationName": "SendNotification",
      "serOperationEndPoint": "/ICOM/NotificationService/SendNotification",
      "serOperationDescription": "send email Notification",
      "serOperationType": "Synchronous",
      "serOperationInputParams": [
        {
          "serParamName": "sender_email",
          "serParamType": "string",
          "serParamValue": "TANDEM-notifications@intracom-telecom.com",
          "serParamDescr": "e-mail address of the Sender",
          "serParamTypicalValue": "string"
        },
        {
          "serParamName": "receiver_email1",
          "serParamType": "string",
          "serParamValue": "tandem-administrator@intracom-telecom.com",

```

```

        "serParamDescr": "e-mail address of the Recipient1",
        "serParamTypicalValue": "string"
    },
    {
        "serParamName": "receiver_email2",
        "serParamType": "string",
        "serParamValue": "string",
        "serParamDescr": "e-mail address of the Recipient2",
        "serParamTypicalValue": "string"
    },
    {
        "serParamName": "message_title",
        "serParamType": "string",
        "serParamValue": "string",
        "serParamDescr": "The title of the message",
        "serParamTypicalValue": "string"
    },
    {
        "serParamName": "message_body",
        "serParamType": "string",
        "serParamValue": "string",
        "serParamDescr": "e-mail text message",
        "serParamTypicalValue": "string"
    }
],
"serOperationOutputParams": [{
    "serParamName": "sending result",
    "serParamType": "number",
    "serParamValue": "201",
    "serParamDescr": "e-
mail sending result, success or failure codes",
    "serParamTypicalValue": "string"
}
]
}
],
"serComputeReq": {
    "serMemorySize": "5",
    "serMemorySizeMU": "Mb",
    "serCPUArchitecture": "x86",
    "serNumVirtualCPUs": "100",
    "serVirtualCPUClock": "string",
    "serNumVirtualCPUsMU": "millicores",
    "serNumVirtualGPUs": "0"
},
"serStorageReq": {
    "serTypeOfStorage": "string",
    "serSizeOfStorage": "0",
    "serSizeOfStorageMU": "Gb"
},
"serLatencyReq": {

```

```

        "serLatencyTimeUnit": "ms",
        "serLatencyValue": "1000"
    },
    "serThroughputReq": {
        "serThroughputMU": "mbps",
        "serThroughputValue": "0.01"
    },
    "serServiceReq": [
        {
            "serServiceId": "e-mail server"
        }
    ],
    "serServiceOptional": [
        {
            "serServiceId": "string"
        }
    ],
    "serSwImage": {
        "serSWImageId": "SWImage00015",
        "serSWImageName": "SWImage00015",
        "serSWImageContainerFormat": "Docker",
        "serSWImageSizeinMBs": "0,5",
        "serSWImageOS": "SWImage00015",
        "serSWImageURL": "/ICOM/SWimages"
    },
    "serializer": "JSON",
    "transportType": "REST_HTTP",
    "transportProtocol": "TCP",
    "scopeOfLocality": "TANDEM_SYSTEM",
    "consumedLocalOnly": true,
    "isLocal": true
}

```

Το σώμα μηνύματος περιλαμβάνει πληροφορίες για την υπηρεσία όπως αναγνωριστικό, όνομα, τύπος, πάροχος, περιγραφή, κατηγορία, έκδοση, κατάσταση, URL περιγραφής, παραμέτρους διαμόρφωσης, λειτουργίες και απαιτήσεις σε πόρους.

- Ενημέρωση των στοιχείων υπαρχόντων υπηρεσιών σε περίπτωση αλλαγής της κατάστασής τους, μέσω της αποστολής ενός HTTP PUT αιτήματος στο παρακάτω URL:

`http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/update/services/{id}`

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό της υπηρεσίας προς ανανέωση ({id}), ενώ παρέχεται και σώμα μηνύματος ίδιο με αυτό της δήλωσης και αποθήκευσης νέων υπηρεσιών.

- Ενημέρωση της κατάστασης μίας υπηρεσίας, μέσω της αποστολής ενός HTTP PUT αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/update/services/state
```

προσδιορίζοντας το αναγνωριστικό της υπηρεσίας προς αλλαγή κατάστασης καθώς και τη νέα της κατάσταση, όπως φαίνεται στο παρακάτω σώμα μηνύματος:

```
{
  "serId": "TANDEM_ser2",
  "state": "Launched"
}
```

- Αναζήτηση στη λίστα των παρεχόντων υπηρεσιών, με βάση κάποιο πεδίο της υπηρεσίας. Για την αναζήτηση με βάση το αναγνωριστικό της υπηρεσίας αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται το αναγνωριστικό της υπηρεσίας προς εύρεση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/get/services/{id}
```

Για αναζήτηση με βάση το όνομα, την έκδοση ή την κατάσταση της υπηρεσίας, αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται η τιμή της παραμέτρου με βάση την οποία θα γίνει η αναζήτηση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/search?{parameter_name}={value}
```

- Διαγραφή ανενεργών services, μέσω της αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/delete/services/{id}
```

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό της υπηρεσίας προς διαγραφή ({id}).

- Δήλωση νέων προϊόντων και αποθήκευση των στοιχείων τους, μέσω της αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/create/products
```

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{
  "productId": "pID",
  "productName": "pName",
  "productType": "FaaS",
  "productProvider": "string",
}
```

```

    "productDescription": "string",
    "productCategory": {
        "href": "Notification",
        "id": "string",
        "name": "string",
        "version": "string"
    },
    "productVersion": "string",
    "productLifeCycleStatus": "string",
    "productLifeCycleStatusReason": "string",
    "productIsBuddle": true,
    "productIsSellable": true,
    "productServiceId": "string",
    "productServiceName": "string",
    "productApplicationId": "string",
    "productApplicationName": "string",
    "productValidFrom": "string",
    "productValidTo": "string",
    "productLastUpdate": "string",
    "productPricingModel": "string",
    "productPricePerChargeUnit": [
        {
            "ChargeUnit": "string",
            "ProductPrice": "number"
        }
    ],
    "productPriceCurrency": "string",
    "productAvailabilityZones": [
        {
            "AvailabilityZoneId": "string",
            "AvailabilityZoneName": "string"
        }
    ],
    "productServiceLevelAgreementId": "string",
    "productServiceLevelAgreementName": "string",
    "productServiceLevelAgreementDescription": "string"
}

```

Το σώμα μηνύματος περιλαμβάνει πληροφορίες για το προϊόν όπως αναγνωριστικό, όνομα, τύπος, πάροχος, περιγραφή, κατηγορία, έκδοση, κατάσταση, ζώνη διαθεσιμότητας και τιμή ανά μονάδα.

- Διαγραφή προϊόντων, μέσω της αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/servicecatalogue/delete/products/{id}
```

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό του προϊόντος προς διαγραφή ({id}).

4.2.4 Υλοποίηση Διαχειριστή Αλυσίδων Υπηρεσιών (Service Chain Manager – SCO)

Η εφαρμογή παρέχει τα κατάλληλα end points προς τη διεπαφή ανθρώπου-μηχανής, επιτρέποντας στον χρήστη να αποθηκεύει την πληροφορία για τις παρεχόμενες από αυτόν Αλυσίδες Υπηρεσιών, καθώς και να πραγματοποιεί αναζήτηση στις υπάρχουσες. Ειδικότερα, παρέχονται οι ακόλουθες λειτουργίες:

- Δήλωση νέων Αλυσίδων Υπηρεσιών (workflows) και αποθήκευση της δομής τους μέσω αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

[http://\[TANDEM_Portal_IP\]:\[TANDEM_Portal_Port\]/workflows/create](http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/workflows/create)

Υποβάλλοντας στο σώμα του αιτήματος ένα json structure της παρακάτω άμορφης

```
{
  "name": "temperature and Humidity Monitoring1",
  "description": null,
  "json": {
    "prmters": [],
    "issync": false,
    "startnodeid": "0",
    "endnodeid": "1",
    "prmtersout": "",
    "prmtersin": [],
    "t001d": {
      "id": "0",
      "xpos": 100.92500305175781,
      "ypos": 83.59999084472656,
      "type": "START_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["2"],
      "p001s": []
    },
    {
      "id": "1",
      "xpos": 500.9250030517578,
      "ypos": 373,
      "type": "END_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": [],
      "p001s": ["5"]
    },
    {
      "id": "2",
      "xpos": 265.9250030517578,
      "ypos": 128,
      "type": "PAAS_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["3"],
      "p001s": ["0"],
      "description": "EDGE FOUNDRY",
      "properties": "",
      "service": "TANDEM_ser3",
      "servicedescr": "Notification Service"
    },
    {
      "id": "3",
      "xpos": 391.9250030517578,
      "ypos": 132,
      "type": "PAAS_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["6"],
      "p001s": ["2"],
      "description": "IOT MONITORING",
      "properties": "",
      "service": "tandem_custom_3",
      "servicedescr": "Trial 2"
    },
    {
      "id": "4",
      "xpos": 309.9250030517578,
      "ypos": 253,
      "type": "DEVICE_SERVICE",
      "truethen": "",
      "falsethen": "",
      "n001s": [],
      "p001s": ["6"],
      "description": "MEC SERVICES",
      "properties": "",
      "service": "tandem_custom_13",
      "servicedescr": "Trial 3"
    },
    {
      "id": "5",
      "xpos": 465.9250030517578,
      "ypos": 241,
      "type": "FAAS_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["1"],
      "p001s": ["6"],
      "description": "email",
      "properties": "",
      "service": "TANDEM_ser3",
      "servicedescr": "Notification Service"
    },
    {
      "id": "6",
      "xpos": 540.9250030517578,
      "ypos": 123.19999980926514,
      "type": "USERAPP_SERVICE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["5", "7", "4"],
      "p001s": ["3"],
      "description": "EVENT SENSOR",
      "properties": "",
      "service": "TANDEM_ser3",
      "servicedescr": "Notification Service"
    },
    {
      "id": "7",
      "xpos": 602.9250030517578,
      "ypos": 245,
      "type": "FAAS_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": [],
      "p001s": ["6"],
      "description": "alarm",
      "properties": "",
      "service": "TANDEM_ser3",
      "servicedescr": "Notification Service"
    }
  ],
  "maxnode": "7",
  "type": "1"
}
```

Όπου περιέχονται τα απαραίτητα στοιχεία έτσι ώστε να απεικονισθεί ο γράφος.

- Ενημέρωση της δομής των υπαρχόντων Αλυσίδων Υπηρεσιών σε περίπτωση αλλαγής της κατάστασής τους

μέσω αποστολής ενός HTTP PUT αιτήματος στο παρακάτω URL:

[http://\[TANDEM_Portal_IP\]:\[TANDEM_Portal_Port\]/workflows/update](http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/workflows/update)

Υποβάλλοντας στο σώμα του αιτήματος ένα json structure της παρακάτω μορφής

```
{
  "name": "temperature and Humidity Monitoring1",
  "description": null,
  "json": {
    "prmtrs": [],
    "issync": false,
    "startnodeid": "0",
    "endnodeid": "1",
    "prmtrsout": "",
    "prmtrsins": [],
    "t001d": {
      "id": "0",
      "xpos": 100.92500305175781,
      "ypos": 83.59999084472656,
      "type": "START_NODE",
      "truethen": "",
      "falsethen": "",
      "n001s": ["2"],
      "p001s": []
    },
    "id": "1",
    "xpos": 500.9250030517578,
    "ypos": 373,
    "type": "END_NODE",
    "truethen": "",
    "falsethen": "",
    "n001s": [],
    "p001s": ["5"]
  },
  "id": "2",
  "xpos": 265.9250030517578,
  "ypos": 128,
  "type": "PAAS_NODE",
  "truethen": "",
  "falsethen": "",
  "n001s": ["3"],
  "p001s": ["0"],
  "description": "EDGE FOUNDRY",
  "properties": "",
  "service": "TANDEM_ser3",
  "servicedescr": "Notification Service",
  "id": "3",
  "xpos": 391.9250030517578,
  "ypos": 132,
  "type": "PAAS_NODE",
  "truethen": "",
  "falsethen": "",
  "n001s": ["6"],
  "p001s": ["2"],
  "description": "IOT MONITORING",
  "properties": "",
  "service": "tandem_custom_3",
  "servicedescr": "Trial 2",
  "id": "4",
  "xpos": 309.9250030517578,
  "ypos": 253,
  "type": "DEVICE_SERVICE",
  "truethen": "",
  "falsethen": "",
  "n001s": [],
  "p001s": ["6"],
  "description": "MEC SERVICES",
  "properties": "",
  "service": "tandem_custom_13",
  "servicedescr": "Trial 3",
  "id": "5",
  "xpos": 465.9250030517578,
  "ypos": 241,
  "type": "FAAS_NODE",
  "truethen": "",
  "falsethen": "",
  "n001s": ["1"],
  "p001s": ["6"],
  "description": "email",
  "properties": "",
  "service": "TANDEM_ser3",
  "servicedescr": "Notification Service",
  "id": "6",
  "xpos": 540.9250030517578,
  "ypos": 123.19999980926514,
  "type": "USERAPP_SERVICE",
  "truethen": "",
  "falsethen": "",
  "n001s": ["5", "7", "4"],
  "p001s": ["3"],
  "description": "EVENT SENSOR",
  "properties": "",
  "service": "TANDEM_ser3",
  "servicedescr": "Notification Service",
  "id": "7",
  "xpos": 602.9250030517578,
  "ypos": 245,
  "type": "FAAS_NODE",
  "truethen": "",
  "falsethen": "",
  "n001s": [],
  "p001s": ["6"],
  "description": "alarm",
  "properties": "",
  "service": "TANDEM_ser3",
  "servicedescr": "Notification Service",
  "maxnode": "7",
  "type": "1"
}
```

Όπου περιέχονται τα απαραίτητα στοιχεία έτσι ώστε να απεικονισθεί ο γράφος.

- Αναζήτηση στη λίστα των διαθέσιμων Αλυσίδων Υπηρεσιών με χρήση του ονόματος τους

μέσω αποστολής ενός HTTP GET αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/workflows/{name}
```

Εισάγοντας στο όρισμα το όνομα της αλυσίδας υπηρεσιών

- Διαγραφή υπάρχουσας Αλυσίδας Υπηρεσιών σε περίπτωση που δε χρησιμοποιείται πια

μέσω αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/workflows/delete/{name}
```

Εισάγοντας στο όρισμα το όνομα της αλυσίδας υπηρεσιών.

4.2.5 Υλοποίηση Διαχειριστή Χρηστών (User Manager)

Η εφαρμογή παρέχει τα κατάλληλα end points προς τη διεπαφή ανθρώπου-μηχανής, επιτρέποντας την αποθήκευση πληροφοριών σχετικά με νέους χρήστες, καθώς και την αναζήτηση στους υπάρχοντες. Ειδικότερα, παρέχονται οι ακόλουθες λειτουργίες:

- Δήλωση νέων χρηστών και αποθήκευση των στοιχείων τους, μέσω της αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

`http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue/create/users`

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{
  "userId": "ICOM",
  "userFirstName": "Dimitris",
  "userLastName": "Christou",
  "userEmail": "info@intracom-telecom.com",
  "userName": "ICOMProfessional",
  "userPassword": "*****",
  "userAccountType": "Professional",
  "userCompanyName": "INTRACOM S.A. Telecom Solutions",
  "userPhoneNumber": "+30-210-6671449",
  "userPaymentInfo": {
    "userCardNumber": "**** * 4344",
    "userExpirationDate": "08/27",
    "userCardname": "INTRACOM",
    "userCVV": "****",
    "userPAN": "****"
  },
  "userPhysicalAddress": {
    "userCountry": "Greece",
    "userAddress": "19,7 Km Markopoulou Ave.",
    "userCity": "Peania, Athens",
    "userState": "Attika",
    "userPostalCode": "19002"
  },
  "userBillingAddress": {
    "userCountry": "Greece",
    "userAddress": "19,7 Km Markopoulou Ave.",
    "userCity": "Peania, Athens",
    "userState": "Attika",
    "userPostalCode": "19002"
  },
  "userRole": "Administrator",
}
```

```

    "userInterests": [{"interestId": "TANDEM_SerCat2", "interestName": "Internet of Things (IoT)"}, {"interestId": "TANDEM_SerCat5", "interestName": "Augmented Reality"}],
    "userOfferedServices": [{"offservID": "TANDEM_ser3", "offservName": "Object Detection Service"}, {"offservID": "TANDEM_ser1", "offservName": "Temperature and Humidity Monitoring Service"}],
    "userObtainedServices": [{"obtserID": "TANDEM_ser5", "obtserName": "EdgeX Foundry services"}, {"obtserID": "TANDEM_ser6", "obtserName": "Grafana Service"}],
    "userInstantiatedServices": [{"instserID": "TANDEM_ser3", "instserName": "Object Detection Service"}, {"instserID": "TANDEM_ser1", "instserName": "Temperature and Humidity Monitoring Service"}],
    "userBillingInfo": [{
        "billingURL": "/ICOM/Billing"
    }]
}

```

Το σώμα μηνύματος περιλαμβάνει πληροφορίες όπως αναγνωριστικό, ονοματεπώνυμο χρήστη, email, κωδικό και username χρήστη, εταιρεία χρήστη, τύπο λογαριασμού, τηλέφωνο χρήστη, πληροφορίες πληρωμής και διευθύνσεις χρήστη, ρόλο χρήστη καθώς και ένα σύνολο από ενδιαφέροντα, προσφερόμενες υπηρεσίες, αποκτηθείσες υπηρεσίες και ενεργοποιηθείσες υπηρεσίες του χρήστη.

- Ενημέρωση των στοιχείων υπαρχόντων χρηστών σε περίπτωση αλλαγής, μέσω της αποστολής ενός HTTP PUT αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue/update/users/{id}
```

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό του χρήστη προς ανανέωση ({id}), ενώ παρέχεται και σώμα μηνύματος ίδιο με αυτό της δήλωσης και αποθήκευσης νέων χρηστών.

- Αναζήτηση στη λίστα των υπαρχόντων χρηστών με βάση κάποιο πεδίο του χρήστη. Για την αναζήτηση με βάση το αναγνωριστικό του χρήστη αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται το αναγνωριστικό του χρήστη προς εύρεση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue/get/users/{id}
```

Για αναζήτηση με βάση το όνομα, το επίθετο, το όνομα χρήστη, τον τύπο λογαριασμού, το όνομα εταιρείας ή το ρόλο του χρήστη, αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται η τιμή της παραμέτρου με βάση την οποία θα γίνει η αναζήτηση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue/search?{parameter_name}={value}
```

- Λήψη των εγγεγραμμένων χρηστών, μέσω της αποστολής ενός HTTP GET αιτήματος στο παρακάτω URL:

`http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue /get/users`

- Διαγραφή χρηστών, μέσω της αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

`http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/usercatalogue/delete/users/{id}`

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό του χρήστη προς διαγραφή ({id}).

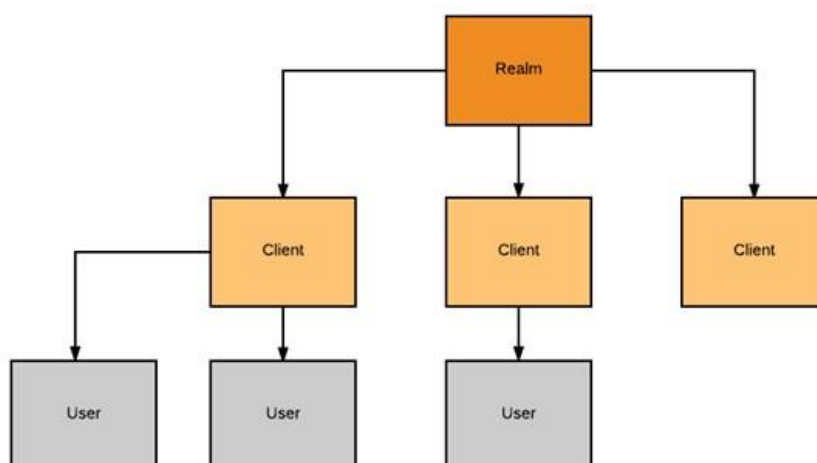
Για τους σκοπούς αυθεντικοποίησης το TANDEM χρησιμοποιεί την πλατφόρμα του **Keycloak [keycloak]**. Το Keycloak είναι μια λύση διαχείρισης ταυτότητας και πρόσβασης ανοιχτού κώδικα που στοχεύει σε σύγχρονες εφαρμογές και υπηρεσίες. Μερικές από τις υπηρεσίες που προσφέρει το keycloak είναι οι εξής:

- **Υποστήριξη πολλαπλών πρωτοκόλλων:** Προς το παρόν, το Keycloak υποστηρίζει τρία διαφορετικά πρωτόκολλα, συγκεκριμένα - OpenID Connect, OAuth 2.0 και SAML 2.0.
- **Single Sign on:** Το Keycloak έχει πλήρη υποστήριξη για Single Sign-On και Single Sign-Out.

Για τους σκοπούς υποστήριξης της προσβασιμότητας στο σύστημα, το σύστημα Tandem προσφέρει τα εξής:

- **Κονσόλα διαχειριστή:** Το Tandem portal προσφέρει GUI βασισμένο στο web όπου μπορούν να γίνουν όλες οι ρυθμίσεις που απαιτούνται από κάθε εφαρμογή.
- **Ταυτότητα χρήστη και προσβάσεις:** Το σύστημα Tandem λειτουργεί ως αυτόνομος διαχειριστής ταυτότητας χρήστη και πρόσβασης, επιτρέποντας τη δημιουργία μιας βάσης δεδομένων χρηστών με προσαρμοσμένους ρόλους και ομάδες. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν περαιτέρω για τον έλεγχο ταυτότητας των χρηστών εντός της εφαρμογής και την ασφάλεια τμημάτων της βάσει προκαθορισμένων ρόλων.
- **Clients:** Οι clients είναι οντότητες που μπορούν να ζητήσουν από το Keycloak τον έλεγχο ταυτότητας ενός χρήστη. Τις περισσότερες φορές, οι clients είναι εφαρμογές και υπηρεσίες που θέλουν να ασφαλιστούν και να παρέχουν μια ενιαία λύση σύνδεσης. Οι πελάτες μπορούν επίσης να είναι οντότητες που θέλουν απλώς να ζητήσουν πληροφορίες ταυτότητας ή ένα διακριτικό πρόσβασης, ώστε να μπορούν να επικαλούνται με ασφάλεια άλλες υπηρεσίες στο δίκτυο που είναι ασφαλείς από το Tandem system.
- **Roles:** Οι ρόλοι προσδιορίζουν έναν τύπο ή κατηγορία χρήστη. Διαχειριστής, χρήστης, διευθυντής και υπάλληλος είναι όλοι τυπικοί ρόλοι που μπορεί να υπάρχουν σε έναν οργανισμό. Οι εφαρμογές συχνά εκχωρούν πρόσβαση και δικαιώματα σε συγκεκριμένους ρόλους και όχι σε μεμονωμένους χρήστες, καθώς η συναλλαγή με χρήστες μπορεί να είναι πολύ λεπτή και δύσκολη στη διαχείριση.

- **Χρήστες:** Οι χρήστες είναι οντότητες που μπορούν να συνδεθούν στο σύστημά σας. Μπορούν να έχουν χαρακτηριστικά που σχετίζονται με τον εαυτό τους, όπως email, όνομα χρήστη, διεύθυνση, αριθμό τηλεφώνου και ημέρα γέννησης. Μπορεί να τους ανατεθεί η ιδιότητα μέλους ομάδας και να τους ανατεθούν συγκεκριμένοι ρόλοι.
- **Αντιστοίχιση ρόλου-χρήστη:** Ένας χρήστης μπορεί να συσχετιστεί με μηδέν ή περισσότερους ρόλους. Αυτές οι πληροφορίες αντιστοίχισης ρόλων μπορούν να ενσωματωθούν σε διακριτικά και ισχυρισμούς, έτσι ώστε οι εφαρμογές να μπορούν να αποφασίζουν τα δικαιώματα πρόσβασης σε διάφορους πόρους που διαχειρίζονται.



Εικόνα 16: Η δομή των χρηστών στο Tandem

Για την χρήση συγκεκριμένων λειτουργιών του καταλόγου υπηρεσιών, όπως για παράδειγμα την δημιουργία νέων υπηρεσιών, ένας χρήστης που έχει τον κατάλληλο ρόλο, χρειάζεται να αποθέσει τα διαπιστευτήριά του στο keycloak(όνομα και κωδικό χρήστη) και να λάβει ένα access token το οποίο και θα χρησιμοποιήσει κατά την κλήση των αντίστοιχων endpoints.

4.2.6 Υλοποίηση Διαχειριστή Υποδομής (Infrastructure Manager)

Η εφαρμογή παρέχει τα κατάλληλα end points προς τη διεπαφή ανθρώπου-μηχανής, επιτρέποντας στον χρήστη να αποθηκεύει και να αναζητά πληροφορίες για την υποδομή καθώς και τους κόμβους της. Ειδικότερα, παρέχονται οι ακόλουθες λειτουργίες:

- Δήλωση νέων υποδομών και αποθήκευση των στοιχείων τους, μέσω της αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/create/infrastructure
```

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{
  "edgeCloudId": "EC2",
  "edgeCloudName": "Edge Cloud 1",
  "edgeCloudAvailabilityZone": "Zone 21",
  "edgeCloudProvider": "INTRACOM S.A. Telecom Solutions",
  "piEdgeIP": "10.152.183.191",
  "piEdgePort": "8080"
}
```

Οι σημαντικότερες πληροφορίες είναι το αναγνωριστικό και το όνομα της νέας υποδομής, η ζώνη διαθεσιμότητας και ο πάροχός της καθώς και η διεύθυνση IP και η πόρτα της.

- Ενημέρωση των στοιχείων υπαρχόντων υποδομών σε περίπτωση αλλαγής, μέσω της αποστολής ενός HTTP PUT αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/update/infrastructure/{id}
```

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό της υποδομής προς ανανέωση ({id}), ενώ παρέχεται και σώμα μηνύματος ίδιο με αυτό της δήλωσης και αποθήκευσης νέων υποδομών.

- Αναζήτηση στη λίστα των υποδομών με βάση κάποιο πεδίο της υποδομής. Για την αναζήτηση με βάση το αναγνωριστικό της υποδομής αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται το αναγνωριστικό της υποδομής προς εύρεση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/get/infrastructure/{id}
```

Για αναζήτηση με βάση το όνομα, τη ζώνη διαθεσιμότητας ή τον πάροχο της υποδομής, αποστέλλεται ένα HTTP GET αίτημα στο παρακάτω URL, όπου προσδιορίζεται η τιμή της παραμέτρου με βάση την οποία θα γίνει η αναζήτηση:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/search?
{parameter_name}={value}
```

- Λήψη των υποδομών με ανανεωμένες μετρικές για τους κόμβους τους, μέσω της αποστολής ενός HTTP GET αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/get/infrastructure
```

- Διαγραφή υποδομών, μέσω της αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

```
http://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/infrastructurecatalogue/delete/infrastructure/{id}
```

Εντός του ίδιου του URL προσδιορίζεται και το αναγνωριστικό της υποδομής προς διαγραφή ({id}).

4.3 Εντοπισμός Έκδοσης TANDEM (TANDEM Edge Orchestrator)

Η εφαρμογή του Edge Orchestrator παρέχει τα κατάλληλα end points προς τη διεπαφή ανθρώπου-μηχανής, επιτρέποντας στον χρήστη να ζητά την εκτέλεση κάποιου PaaS service, καθώς και τον τερματισμό του. Ειδικότερα, παρέχονται οι ακόλουθες λειτουργίες:

- Αίτημα για εκτέλεση ενός PaaS service, μέσω της αποστολής ενός HTTP POST αιτήματος στο παρακάτω URL:

```
https://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/systemmanager/instantiate/service
```

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{
  "piEdgeIP": "10.152.183.191",
  "piEdgePort": "8080",
  "paas_service_name": "iotmonitoring",
  "paas_instance_name": "iotmonitoring",
  "autoscaling_type": "minimize_cost",
  "count_min": 1,
  "count_max": 2,
  "location": "Peania_Athens_19002_2",
  "all_node_ports": true
}
```

Οι σημαντικότερες πληροφορίες είναι η IP και η πόρτα του υπολογιστικού νέφους όπου θα εκτελεστεί η υπηρεσία, το όνομα της υπηρεσίας και του στιγμιότυπου της καθώς και η τοποθεσία της.

- Τερματισμός εκτέλεσης ενός PaaS service, μέσω της αποστολής ενός HTTP DELETE αιτήματος στο παρακάτω URL:

```
https://[TANDEM_Portal_IP]:[TANDEM_Portal_Port]/systemmanager/delete/service
```

προσδιορίζοντας τις απαραίτητες πληροφορίες που φαίνονται στο παρακάτω σώμα μηνύματος:

```
{  "piEdgeIP": "10.152.183.191",
  "piEdgePort": "8080",
  "serviceName": "iotmonitoringpaas"
}
```

5 Οδηγίες Εγκατάστασης

5.1 Εγκατάσταση Λογισμικού Κόμβων Άκρων (Παρυφών)

5.1.1 Προετοιμασία των κόμβων TANDEM

Για την δημιουργία του Kubernetes συμπλέγματος κόμβων χρησιμοποιήσαμε το **MicroK8s [MicroK8s]**, το οποίο αυτοματοποιεί σε μεγάλο βαθμό την εγκατάσταση του Kubernetes και είναι κατάλληλο για γρήγορες, μοναδικού πακέτου Kubernetes for προγραμματιστές, IoT (Διαδίκτυο των Πραγμάτων) και υποδομές άκρων. Συγκεκριμένα, ακολουθώντας το εγχειρίδιο εγκατάστασης.

Στις υποδομές μας (προς το παρόν) έχουμε 2 μηχανήματα (servers) με Ubuntu 20.04 λειτουργικό σύστημα. Το ένα, που υπερισχύει σε χαρακτηριστικά (με διεύθυνση 146.124.106.209) θα χρησιμοποιηθεί παράλληλα ως κύριος (master) κόμβος και κόμβος εργάτης (worker node), και το δεύτερο (με διεύθυνση 146.124.106.210) ως δεύτερος κόμβος εργάτης.

Συγκεκριμένα, για την εγκατάσταση του microk8s σε κάθε μηχανήμα εκτελούμε:

Έπειτα, μετά την επιτυχή εγκατάσταση σε κάθε κόμβο, εκτελέσαμε τις παρακάτω εντολές

```
sudo snap install microk8s --classic
```

ώστε να ενώσουμε τα 2 μηχανήματα σε ένα σύμπλεγμα(cluster) κόμβων:

Μηχάνημα 1:

```
microk8s add-node
```

λαμβάνοντας την απάντηση:

```
Use the '--worker' flag to join a node as a worker not running  
the control plane, eg:
```

```
microk8s join  
146.124.106.209:25000/1a1e5376d5dcbb42a52c9d4bee2e2c5e/18e948d  
829be --worker
```

έτσι, εκτελούμε την παρακάτω εντολή στο μηχανήμα 2:

```
microk8s join  
146.124.106.209:25000/1a1e5376d5dcbb42a  
52c9d4bee2e2c5e/18e948d829be --worker
```

Μετά, για την ολοκλήρωση τις προετοιμασίας του συμπλεγμάτων κόμβων, πρέπει να εγκαταστήσουμε τα απαραίτητα πακέτα που χρειαζόμαστε, δηλαδή

1. DNS: Σύστημα ονοματολογίας συμπλέγματος
2. Dashboard: Γραφικό περιβάλλον παρακολούθησης συστήματος
3. Storage: Σύστημα αποθήκευσης
4. Prometheus: Σύστημα παρακολούθησης μονάδων συμπλέγματος.

Πρώτα πρέπει να δημιουργήσουμε τα kubernetes namespaces που θα χρειαστούμε.

Έτσι εκτελούμε στον κύριο κόμβο:

```
kubectl create namespace monitoring
```

```
kubectl create namespace pi-edge
```

```
kubectl create namespace pi-edge-system
```

```
microk8s enable dns dashboard storage
```

```
kubectl create namespace argo
```

```
microk8s enable promethues
```

Συγκεκριμένα, η τελευταία εντολή εγκαθιστά το ολοκληρωμένο σύνολο από υπηρεσίες παρακολούθησης πόρων που περιγράψαμε στην ενότητα 3.2.

Επίσης, με δεδομένο ότι και οι 2 κόμβοι ανήκουν στο ίδιο kubernetes σύμπλεγμα, και στην ίδια τοποθεσία, τοποθετούμε «επιγραφή» και στους 2 kubernetes κόμβους με την παρακάτω εντολή:

Για τον κύριο (και εργάτη) κόμβο:

```
kubectl label node k8smaster  
location=Peania_Athens_19002
```

για τον εργάτη κόμβο:

Στην συνέχεια πρέπει να εγκαταστήσουμε τα προγράμματα που θα διαχειρίζονται τις PaaS

```
kubectl label node k8ssecondary  
location=Peania_Athens_19002
```

(πλατφόρμα ως υπηρεσία) υπηρεσίες και FaaS (λειτουργία σαν υπηρεσία) υπηρεσίες στις υποδομές μας. Συγκεκριμένα, για τις PaaS χρησιμοποιούμε την ri-Edge πλατφόρμα η οποία χαρακτηρίζεται ως διαχειριστική πλατφόρμα του άκρου (edge platform manage), και έχει αναπτυχθεί από την ομάδα μας (ICOM R&D). Χρησιμοποιώντας το παρακάτω yaml αρχείο με όνομα ri-edge.yaml,

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  annotations:  
    kompose.cmd: kompose convert  
    kompose.version: 1.26.0 (40646f47)  
  creationTimestamp: null  
  labels:  
    io.kompose.service: kubempcontr  
  name: kubempcontr  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      io.kompose.service: kubempcontr  
  strategy: {}  
  template:
```

```

metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    #io.kompose.network/netEMPkub: "true"
    io.kompose.service: kubempcontr
spec:
  nodeSelector:
    location: Peania_Athens_19002
  containers:
    - env:
        - name: KUBERNETES_MASTER_HOST
          value: https://146.124.106.209:16443
        - name: KUBERNETES_MASTER_TOKEN
          value: TVVGV1FJb3l0alNUOUdEeVRvbnZKdGhZS3FXUXg2aG5mMGVxekRqRnZhWTOK
        - name: EMP_STORAGE_URI
          value: mongodb://mongopiedge:27017
      image: nikospsar/pi_edge_controller:1.2.1
      name: kubempcontr
      ports:
        - containerPort: 8080
      resources: {}
      imagePullPolicy: Always
    restartPolicy: Always
status: {}
---
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: kubempcontr
  name: kubempcontr
spec:
  type: NodePort
  ports:
    - name: "8080"
      nodePort: 32410
      port: 8080
      targetPort: 8080
  selector:
    io.kompose.service: kubempcontr
status:
  loadBalancer: {}
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv-volume
  labels:

```

```
  io.kompose.service: mongo-db
spec:
  storageClassName: manual
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  creationTimestamp: null
  labels:
    io.kompose.service: mongo-db
  name: mongo-db
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
status: {}
---
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: mongopiedge
  name: mongopiedge
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: mongopiedge
  strategy:
    type: Recreate
  template:
    metadata:
      annotations:
        kompose.cmd: kompose convert
        kompose.version: 1.26.0 (40646f47)
      creationTimestamp: null
      labels:
        #io.kompose.network/netEMPkub: "true"
        io.kompose.service: mongopiedge
    spec:
      nodeSelector:
        location: Peania_Athens_19002
      containers:
```

```
- image: mongo
  name: mongopiedge
  ports:
    - containerPort: 27017
  resources: {}
  volumeMounts:
    - mountPath: /data/db
      name: mongo-db
  restartPolicy: Always
  volumes:
    - name: mongo-db
      persistentVolumeClaim:
        claimName: mongo-db
status: {}
---
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: mongopiedge
  name: mongopiedge
spec:
  type: NodePort
  ports:
    - name: "27017"
      nodePort: 32411
      port: 27017
      targetPort: 27017
  selector:
    io.kompose.service: mongopiedge
status:
  loadBalancer: {}
```

εκτελούμε στον κύριο κόμβο,

```
kubectl apply -f pi-edge.yaml -n pi-edge-  
system
```

Στην συνέχεια, για τις FaaS υπηρεσίες χρησιμοποιούμε το Argo που έχει αναπτυχθεί για την διαχείριση εφαρμογών που αποτελούνται από ροή εργασιών. Έτσι, πραγματοποιούμε με την σειρά στον κύριο κόμβο:

```
kubectl apply -f  
https://raw.githubusercontent.com/argoproj/  
argo-events/stable/manifests/install.yaml
```

```
kubectl apply -f  
https://raw.githubusercontent.com/argoproj/  
argo-events/stable/manifests/install-  
validating-webhook.yaml
```

```
kubectl -n argo-events apply -f  
https://raw.githubusercontent.com/argoproj/  
argo-  
events/stable/examples/eventbus/native.yaml
```

```
kubectl apply -n argo -f  
https://raw.githubusercontent.com/argoproj/  
argo-workflows/stable/manifests/install.yaml
```

Αν όλα έχουν εγκατασταθεί με επιτυχία, πρέπει εκτελώντας την παρακάτω εντολή:
να λάβουμε:

```
tanmaster@k8smaster:~$ kubectl get deploy -A
```

NAMESPACE	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
kube-system	calico-kube-controllers	1/1	1	1	4d
kube-system	hostpath-provisioner	1/1	1	1	4d
kube-system	coredns	1/1	1	1	4d
kube-system	kubernetes-dashboard	1/1	1	1	4d
kube-system	metrics-server	1/1	1	1	4d
kube-system	dashboard-metrics-scraper	1/1	1	1	4d
pi-edge-system	mongopiedge	1/1	1	1	3d23h
monitoring	prometheus-adapter	2/2	2	2	3d22h
monitoring	prometheus-operator	1/1	1	1	3d22h
monitoring	grafana	1/1	1	1	3d22h
monitoring	kube-state-metrics	1/1	1	1	3d22h
monitoring	blackbox-exporter	1/1	1	1	3d22h
argo-events	controller-manager	1/1	1	1	3d22h
argo-events	events-webhook	1/1	1	1	3d22h
argo	workflow-controller	1/1	1	1	3d21h
argo	argo-server	1/1	1	1	3d21h
pi-edge-system	kubempctr	1/1	1	1	3d23h
argo-events	webhook-eventsources-44n55	1/1	1	1	2d22h
argo-events	webhook-sensor-znnff	1/1	1	1	2d22h

5.1.2 Παροχή – εγκατάσταση βασικών (core) TANDEM PaaS υπηρεσιών

Εφόσον έχει ολοκληρωθεί επιτυχημένα η προετοιμασία των κόμβων TANDEM που περιεγράφηκε παραπάνω, μπορούμε να προχωρήσουμε στην εγκατάσταση των βασικών TANDEM PaaS υπηρεσιών χρησιμοποιώντας την pi-Edge πλατφόρμα. Αυτή η διαδικασία ονομάζεται ενεργοποίηση TANDEM κόμβου.

Συγκεκριμένα, εφόσον έχουν εγγραφεί επιτυχώς στον pi-Edge κατάλογο τα Service Functions αλλά και οι PaaS υπηρεσίες που αποτελούνται από 1 ή παραπάνω Service Functions, ο χρήστης μπορεί να της εγκαταστήσει σε όποια τοποθεσία επιθυμεί core PaaS υπηρεσία.

Για παράδειγμα,

Για να εγκαταστήσουμε την EdgeX πλατφόρμα στο kubernetes σύμπλεγμα που αναφέρεται παραπάνω, πρέπει να πραγματοποιήσουμε τα παρακάτω βήματα:

Βήμα 1: Επιβεβαίωση ότι το EdgeX είναι γραμμένο στον κατάλογο σαν PaaS υπηρεσία:

```
curl http://146.124.106.209:32410/piedge-connector/2.0.0/paasServices/
```

Όπου πρέπει να λάβουμε απάντηση που να περιέχει το EdgeX. Η πληροφορία που θα ληφθεί θα εμπεριέχει και πληροφορίες για τα ποσοστά των μετρήσιμων μονάδων (cpu, μνήμη κ.λπ..) που πρέπει να κλιμακώνεται το κάθε service (scale up/ scale down). Τους κανόνες της κλιμάκωσης αυτής αναλαμβάνει να την εγκαταστήσει η pi-Edge πλατφόρμα.

Η EdgeX πλατφόρμα αποτελείται από τα παρακάτω service functions:

1. sys-mgmt-agent
2. app-service-configurable-mqtt
3. app-service-configurable-rules
4. core-command

5. core-consul
6. core-data
7. core-metadata
8. device-rest
9. kuiper
10. redis
11. support-notifications
12. support-scheduler

Παράδειγμα απάντησης βήματος 1 (παρατίθεται η απάντηση μόνο με ένα service function):

```
{
  "_id": "62c460cd398c76a53e2f9b1e",
  "name": "edgex",
  "service_functions": [
    {
      "autoscaling_metric": "cpu",
      "env_parameters": [
        {
          "name": "Clients_Command_Host",
          "value_ref": "core-command"
        },
        {
          "name": "Clients_CoreData_Host",
          "value_ref": "core-data"
        },
        {
          "name": "Clients_Data_Host",
          "value_ref": "core-data"
        },
        {
          "name": "Clients_Metadata_Host",
          "value_ref": "core-metadata"
        },
        {
          "name": "Clients_Notifications_Host",
          "value_ref": "support-notifications"
        },
        {
          "name": "Clients_RulesEngine_Host",
          "value_ref": "kuiper"
        },
        {
          "name": "Clients_Scheduler_Host",
```

```

        "value_ref": "support-scheduler"
      },
      {
        "name": "Databases_Primary_Host",
        "value_ref": "redis"
      },
      {
        "name": "EDGEX_SECURITY_SECRET_STORE",
        "value": "false"
      },
      {
        "name": "EDGEX_STARTUP_DURATION",
        "value": "60"
      },
      {
        "name": "ExecutorPath",
        "value": "/sys-mgmt-executor"
      },
      {
        "name": "Logging_EnableRemote",
        "value": "false"
      },
      {
        "name": "MetricsMechanism",
        "value": "executor"
      },
      {
        "name": "Registry_Host",
        "value_ref": "core-consul"
      },
      {
        "name": "Service_Host",
        "value_ref": "sys-mgmt-agent"
      }
    ],
    "service_function_identifier_name": "sys-mgmt-agent",
    "volume_mounts": [
      {
        "name": "edgex-sys",
        "storage": "100Mi"
      },
      {.....
      .....
      .....}
    ]

```

```
}  
]  
}
```

Μπορούμε να ελέγξουμε και στον κατάλογο με τι πληροφορίες (image κλπ.) είναι εγγεγραμμένο κάθε service function. Για παράδειγμα για το service function με όνομα "sys-mgmt-agent", εκτελούμε:

```
curl http://146.124.106.209:32410/piedge-  
connector/2.0.0/serviceFunctions/62c307ea69  
960314ec93614f
```

και λαμβάνουμε την παρακάτω απάντηση:

```
{  
  "_id": "62c307ea69960314ec93614f",  
  "application_ports": [  
    48090  
  ],  
  "autoscaling_policies": [  
    {  
      "monitoring_metrics": [  
        {  
          "is_default": null,  
          "limit": "0.5Gi",  
          "metric": "memory",  
          "request": "0.2Gi",  
          "util_percent": 80  
        },  
        {  
          "is_default": true,  
          "limit": "500m",  
          "metric": "cpu",  
          "request": "150m",  
          "util_percent": 80  
        }  
      ],  
      "policy": "minimize_cost"  
    },  
    {  
      "monitoring_metrics": [  
        {  
          "is_default": null,
```

```

        "limit": "1Gi",
        "metric": "memory",
        "request": "0.5Gi",
        "util_percent": 50
    },
    {
        "is_default": true,
        "limit": "500m",
        "metric": "cpu",
        "request": "200m",
        "util_percent": 40
    }
],
"policy": "maximize_performance"
}
],
"image": "edgexfoundry/docker-sys-mgmt-agent-go:1.2.1",
"name": "sys-mgmt-agent",
"required_env_parameters": [
    {
        "name": "Clients_Command_Host"
    },
    {
        "name": "Clients_CoreData_Host"
    },
    {
        "name": "Clients_Data_Host"
    },
    {
        "name": "Clients_Metadata_Host"
    },
    {
        "name": "Clients_Notifications_Host"
    },
    {
        "name": "Clients_RulesEngine_Host"
    },
    {
        "name": "Clients_Scheduler_Host"
    },
    {
        "name": "Databases_Primary_Host"
    },
    {
        "name": "EDGEX_SECURITY_SECRET_STORE"
    },
    {
        "name": "EDGEX_STARTUP_DURATION"
    },
],

```

```
{
  "name": "ExecutorPath"
},
{
  "name": "Logging_EnableRemote"
},
{
  "name": "MetricsMechanism"
},
{
  "name": "Registry_Host"
},
{
  "name": "Service_Host"
}
],
"required_volumes": [
  {
    "name": "edgex-sys",
    "path": "/var/run/docker.sock"
  }
],
"type": "Container"
}
```

Έπειτα, ο χρήστης μπορεί να χρησιμοποιήσει την ri-Edge πλατφόρμα για την ενεργοποίηση το TANDEM κόμβου (επιλέγοντας στο παράδειγμα μας να εγκαταστήσει μόνο την EdgeX πλατφόρμα ως κύρια (core) PaaS υπηρεσία και την τοποθεσία του κόμβου που θέλουμε να ενεργοποιηθεί με το label "location"):

```
curl --location --request POST
'http://146.124.106.209:32410/piedge-
connector/2.0.0/activateTandemNode --
header 'Content-Type: text/plain' --data-raw
'{
  "location": "Peania_Athens_19002",
  "paas_services": [
    {
      "paas_service_name": "edgex",
      "paas_instance_name": "edgex",
      "autoscaling_type":
        "maximize_performance",
      "count_min": 1,
      "count_max": 2,
      "location": "Peania_Athens_node1",
      "all_node_ports": true
    }
  ]
}'
```

λαμβάνοντας την απάντηση (αν όλα εγκαταστάθηκαν επιτυχώς):

```
" \nPaaS service edgex-app-service-configurable-mqtt deployed successfully \nPaaS service
edgex-app-service-configurable-rules deployed successfully \nPaaS service edgex-core-
command deployed successfully \nPaaS service edgex-core-consul deployed successfully
\PaaS service edgex-core-data deployed successfully \nPaaS service edgex-core-metadata
deployed successfully \nPaaS service edgex-device-rest deployed successfully \nPaaS service
edgex-kuiper deployed successfully \nPaaS service edgex-redis deployed successfully \nPaaS
service edgex-support-notifications deployed successfully \nPaaS service edgex-support-
scheduler deployed successfully \nPaaS service edgex-sys-mgmt-agent deployed successfully"
```

Μπορούμε να επιβεβαιώσουμε ότι εγκαταστάθηκε επιτυχώς εκτελώντας:

```
http://146.124.106.209:32410/piedge-connector/2.0.0/deployedPaasServices/edgex
```

Όπου πρέπει να λάβουμε απάντηση με όλα τα services σε “running” κατάσταση.

Εναλλακτικά, μπορεί να χρησιμοποιηθεί και η κονσόλα του κύριου κόμβου (master node) για τον έλεγχο.

```
kubectl get deploy -n pi-edge
```

```
tanmaster@k8smaster:~$ kubectl get deploy -n pi-edge
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
edgex-sys-mgmt-agent                1/1      1              1            3d4h
edgex-core-data                     1/1      1              1            3d4h
edgex-core-consul                   1/1      1              1            3d4h
edgex-core-metadata                 1/1      1              1            3d4h
edgex-device-rest                   1/1      1              1            3d4h
edgex-kuiper                        1/1      1              1            3d4h
edgex-app-service-configurable-rules 1/1      1              1            3d4h
edgex-app-service-configurable-mqtt 1/1      1              1            3d4h
edgex-support-scheduler              1/1      1              1            3d4h
edgex-support-notifications          1/1      1              1            3d4h
edgex-redis                         1/1      1              1            3d4h
edgex-core-command                  1/1      1              1            3d4h
```

```
kubectl get svc -n pi-edge
```

```
tanmaster@k8smaster:~$ kubectl get svc -n pi-edge
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
edgex-app-service-configurable-mqtt	NodePort	10.152.183.17	<none>	48101:30773/TCP	3d5h
edgex-app-service-configurable-rules	NodePort	10.152.183.228	<none>	48100:31062/TCP	3d5h
edgex-core-command	NodePort	10.152.183.216	<none>	48082:32351/TCP	3d5h
edgex-core-consul	NodePort	10.152.183.91	<none>	8400:30552/TCP,8500:30288/TCP	3d5h
edgex-core-data	NodePort	10.152.183.103	<none>	48080:31613/TCP,5563:31869/TCP	3d5h
edgex-core-metadata	NodePort	10.152.183.164	<none>	48081:31420/TCP	3d5h
edgex-device-rest	NodePort	10.152.183.154	<none>	49986:31547/TCP	3d5h
edgex-kulper	NodePort	10.152.183.112	<none>	48075:31657/TCP,20498:30044/TCP	3d5h
edgex-redis	NodePort	10.152.183.106	<none>	6379:30143/TCP	3d5h
edgex-support-notifications	NodePort	10.152.183.253	<none>	48060:31437/TCP	3d5h
edgex-support-scheduler	NodePort	10.152.183.234	<none>	48085:31344/TCP	3d5h
edgex-sys-ngmt-agent	NodePort	10.152.183.200	<none>	48090:31214/TCP	3d5h

```
kubectl get pvc -n pi-edge
```

```
tanmaster@k8smaster:~$ kubectl get pvc -n pi-edge
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
edgex-core-consul-consul-config	Bound	pvc-582796be-cfc0-468f-a80d-68c1b36c54be	100Mi	RWX	microk8s-hostpath	3d5h
edgex-redis-db-data	Bound	pvc-50557510-7ca4-494e-b377-e3bc52fa5a75	100Mi	RWX	microk8s-hostpath	3d5h
edgex-sys-ngmt-agent-edgex-sys	Bound	pvc-05a25f27-afc5-400b-9fb1-a72ac5ec05a	100Mi	RWX	microk8s-hostpath	3d5h
edgex-core-consul-consul-data	Bound	pvc-6abfa0a2-858c-48e0-b78e-45ab5edd4890	100Mi	RWX	microk8s-hostpath	3d5h

```
kubectl get hpa -n pi-edge
```

```
tanmaster@k8smaster:~$ kubectl get hpa -n pi-edge
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
edgex-app-service-configurable-rules	Deployment/edgex-app-service-configurable-rules	0%/80%	1	1	1	3d5h
edgex-app-service-configurable-mqtt	Deployment/edgex-app-service-configurable-mqtt	0%/80%	1	1	1	3d5h
edgex-kulper	Deployment/edgex-kulper	0%/80%	1	1	1	3d5h
edgex-core-data	Deployment/edgex-core-data	8%/80%	1	1	1	3d5h
edgex-redis	Deployment/edgex-redis	1%/80%	1	1	1	3d5h
edgex-core-consul	Deployment/edgex-core-consul	4%/80%	1	1	1	3d5h
edgex-support-scheduler	Deployment/edgex-support-scheduler	3%/80%	1	1	1	3d5h
edgex-device-rest	Deployment/edgex-device-rest	0%/80%	1	1	1	3d5h
edgex-sys-ngmt-agent	Deployment/edgex-sys-ngmt-agent	0%/80%	1	1	1	3d5h
edgex-support-notifications	Deployment/edgex-support-notifications	5%/80%	1	1	1	3d5h
edgex-core-command	Deployment/edgex-core-command	11%/80%	1	1	1	3d5h
edgex-core-metadata	Deployment/edgex-core-metadata	5%/80%	1	1	1	3d5h

Επομένως, πλέον με δεδομένο ότι όλα είναι σε “running” κατάσταση μπορούμε να είμαστε βέβαιοι ότι έχει ενεργοποιηθεί ο TANDEM κόμβος.

5.2 Εγκατάσταση Πλατφόρμας Διαχείρισης Συστήματος

5.2.1 Εγκατάσταση του TANDEM Service Catalogue

Για την εγκατάσταση του καταλόγου των services του TANDEM σε Kubernetes έχουν χρησιμοποιηθεί τα παρακάτω Kubernetes yaml αρχεία:

Name	Date modified	Type	Size
keycloak	21/7/2022 11:50 πμ	File folder	
serviceCatalogue	21/7/2022 11:59 πμ	File folder	
keycloak-deployment.yaml	1/6/2022 4:08 μμ	Yaml Source File	2 KB
keycloak-service.yaml	1/6/2022 4:25 μμ	Yaml Source File	1 KB
postgres-data-persistentvolumeclaim.ya...	18/5/2022 12:32 μμ	Yaml Source File	1 KB
postgres-deployment.yaml	18/5/2022 12:32 μμ	Yaml Source File	2 KB
postgres-service.yaml	19/5/2022 4:23 μμ	Yaml Source File	1 KB
service-catalogue-configmap.yaml	1/6/2022 1:16 μμ	Yaml Source File	2 KB
service-catalogue-deployment.yaml	19/7/2022 1:47 μμ	Yaml Source File	2 KB
service-catalogue-mongo-deployment.y...	18/5/2022 12:32 μμ	Yaml Source File	1 KB
service-catalogue-mongo-service.yaml	19/5/2022 12:34 μμ	Yaml Source File	1 KB
service-catalogue-networkpolicy.yaml	19/5/2022 12:34 μμ	Yaml Source File	1 KB
service-catalogue-service.yaml	19/5/2022 1:05 μμ	Yaml Source File	1 KB

Παρακάτω δίνεται ένα παράδειγμα των αντίστοιχων yaml:

Service-catalogue-deployment.yaml

```

apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: servicecatalogue
  name: servicecatalogue
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: servicecatalogue
  strategy: {}
  template:
    metadata:
      annotations:
        kompose.cmd: kompose convert
        kompose.version: 1.26.0 (40646f47)
      creationTimestamp: null
      labels:
        io.kompose.service: servicecatalogue

```

```
spec:
  containers:
    - image: marievixezonaki/service-catalogue:latest
      name: servicecatalogue
      ports:
        - containerPort: 8080
      resources: {}
      hostname: servicecatalogue
      restartPolicy: Always
  status: {}
---
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: servicecatalogue
  name: servicecatalogue
spec:
  type: NodePort
  ports:
    - name: "8082"
      port: 8082
      targetPort: 8082
  selector:
    io.kompose.service: servicecatalogue
status:
  loadBalancer: {}
```

Για την εγκατάσταση των υπηρεσιών του TANDEM, εκτελούμε την παρακάτω εντολή:

```
Kubectl apply -f service-catalogue-deployment  
-n service-catalogue
```

Όπου service-catalogue-deployment είναι ο φάκελος στον οποίο βρίσκονται όλα τα παραπάνω yaml αρχεία.

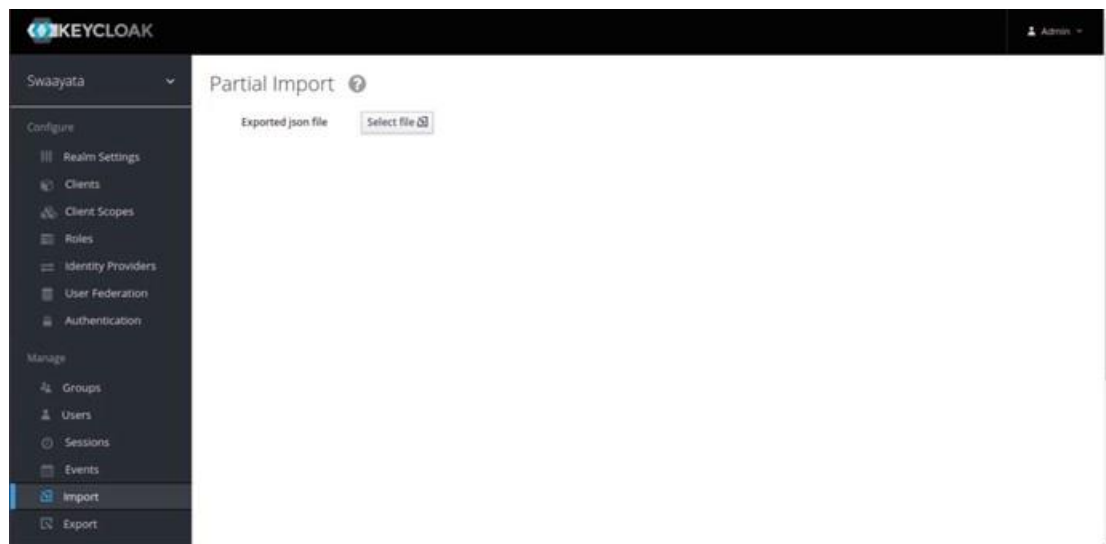
Με το πέρας της παραπάνω εντολής, μπορούμε να εξακριβώσουμε το επιτυχές deployment:

```
Kubectl get deploy -n service-catalogue
```

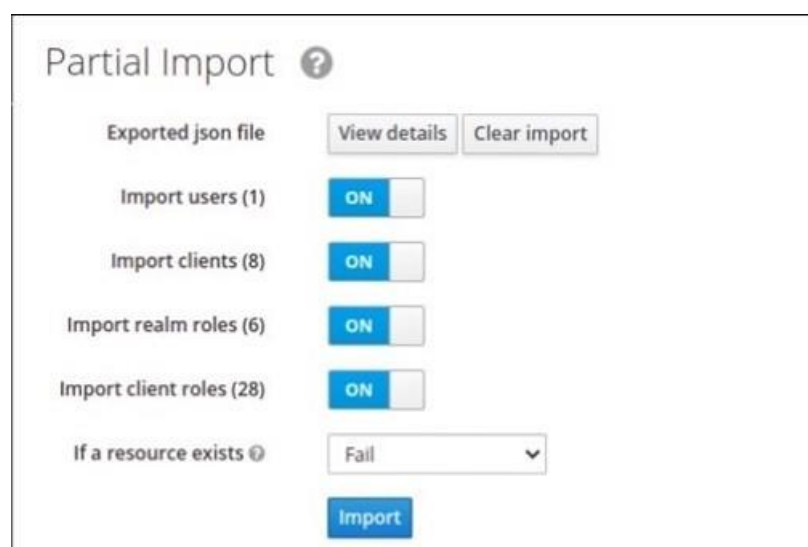
```
tanmaster@k8smaster:~$ kubectl get deploy -n service-catalogue
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
service-catalogue-mongo            1/1      1              1            7d20h
postgres                           1/1      1              1            7d20h
keycloak                           1/1      1              1            7d20h
service-catalogue                  1/1      1              1            7d20h
```

Για το configuration του keycloak, πρέπει να συνδεθούμε στο service του keycloak μέσω του UI στη διεύθυνση : <https://146.124.106.209:31443/auth/admin/master/console/#/realms/master/partial-import>

Και επιλέγουμε import χρησιμοποιώντας το αρχείο realm-export.json, στον φάκελο keycloak:



Επιλέγουμε να κανουμε import όλα τα στοιχεία:

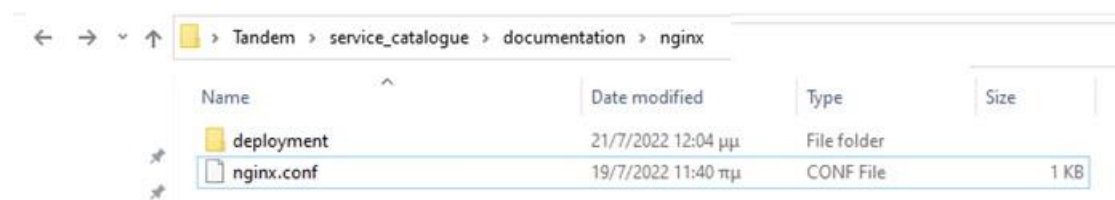


5.2.2 Εγκατάσταση του Nginx σαν reverse proxy και application server

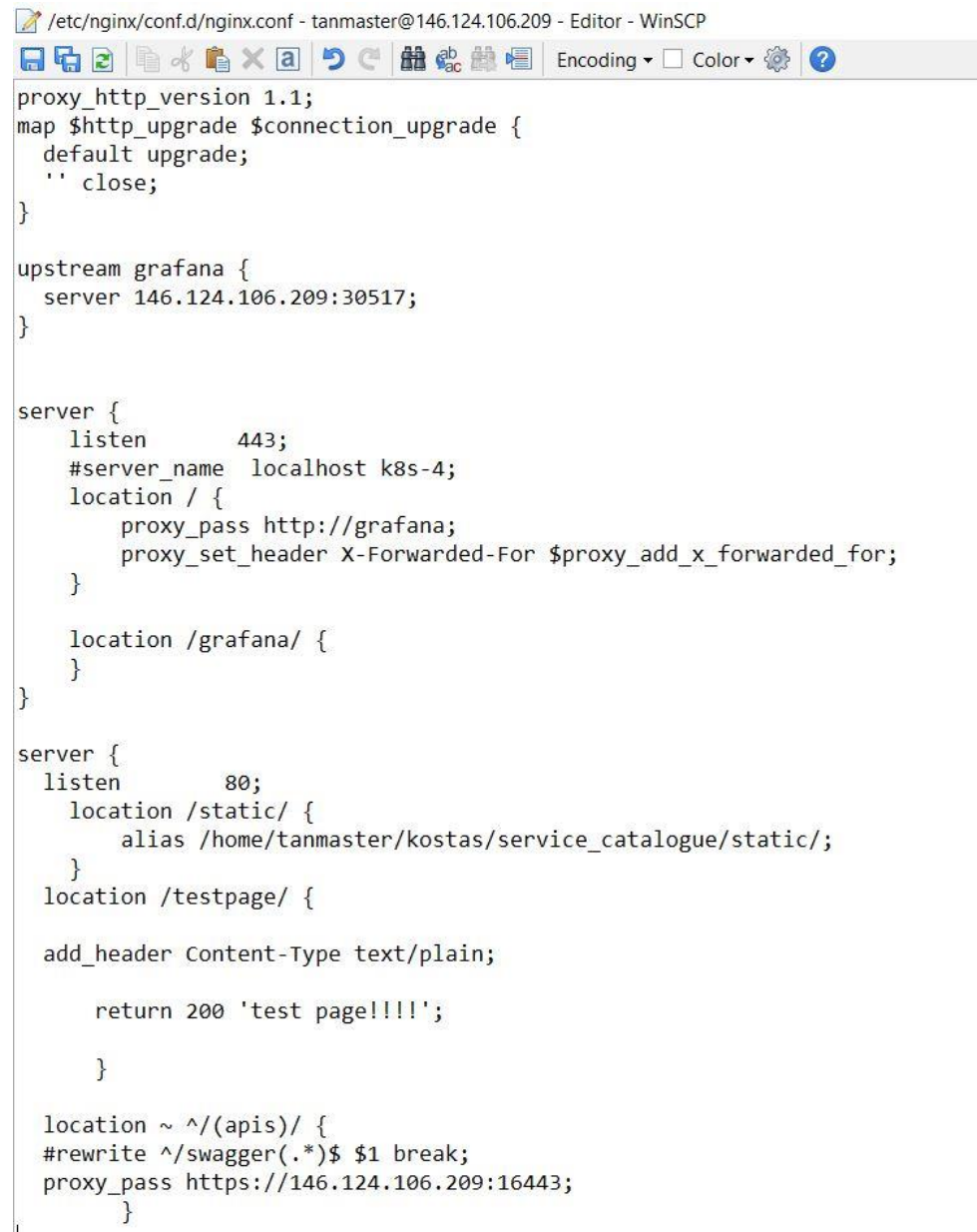
Για την εγκατάσταση του Nginx σαν reverse proxy και application server, χρησιμοποιούμε τις παρακάτω εντολές

```
sudo apt update  
sudo apt install nginx
```

Μετά το πέρας της εγκατάστασης, στον φάκελο etc/nginx/conf, κάνουμε override το αρχείο nginx.conf με το αρχείο nginx.conf που βρίσκεται στον φάκελο service_catalogue->documentation->nginx.



Τα περιεχόμενα του αρχείου φαίνονται παρακάτω:



```
/etc/nginx/conf.d/nginx.conf - tanmaster@146.124.106.209 - Editor - WinSCP
proxy_http_version 1.1;
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

upstream grafana {
    server 146.124.106.209:30517;
}

server {
    listen 443;
    #server_name localhost k8s-4;
    location / {
        proxy_pass http://grafana;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /grafana/ {
    }
}

server {
    listen 80;
    location /static/ {
        alias /home/tanmaster/kostas/service_catalogue/static;
    }
    location /testpage/ {
        add_header Content-Type text/plain;

        return 200 'test page!!!!';
    }

    location ~ ^/(apis)/ {
        #rewrite ^/swagger(.*)$ $1 break;
        proxy_pass https://146.124.106.209:16443;
    }
}
```

```

location ~ ^/(apis)/ {
#rewrite ^/swagger(.*)$ $1 break;
proxy_pass https://146.124.106.209:16443;
}

location ~ ^/(servicecatalogue|workflows)/ {
proxy_pass http://10.152.183.53:8082;
}

location ~ ^/(infrastructurecatalogue)/ {
proxy_pass http://10.152.183.117:8081;
}

location ~ ^/(usercatalogue)/ {
#rewrite ^/swagger(.*)$ $1 break;
proxy_pass http://10.152.183.210:8080;
}

location ~ ^/(systemmanager)/ {
#rewrite ^/swagger(.*)$ $1 break;
proxy_pass http://10.152.183.254:8083;
}

location ~ ^/(piedge-connector)/ {
proxy_pass http://10.152.183.191:8080;
}

location ~ ^/kubeflow/ {
rewrite ^/(.*) /$1 break;
proxy_pass http://146.124.106.232:80;
}
}

server {
listen 8443; #keycloak https
server_name keycloak_https;
location / {
proxy_pass https://10.152.183.84:31443;
}
}

server {
listen 8180; #keycloak http
server_name keycloak_http;
location / {
proxy_pass http://10.152.183.84:8180; #keycloak service clusterIP
}
}

```

Το directory `/home/tanmaster/application/service_catalogue/static/` πρέπει να αντικατασταθεί από το directory όπου ο χρήστης επιθυμεί να αποθέσει τα στατικά αρχεία.

Με το πέρας των παραπάνω βημάτων, εκτελούμε την παρακάτω εντολή:

```
sudo systemctl start nginx
```

5.2.3 Εγκατάσταση του TANDEM Infrastructure Manager

Για την εγκατάσταση του καταλόγου/διαχειριστή της υποδομής του TANDEM σε Kubernetes χρησιμοποιήθηκε το παρακάτω Kubernetes yaml αρχείο:

```
infrastructurecatalogue.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: infrastructurecatalogue
  name: infrastructurecatalogue
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: infrastructurecatalogue
  strategy: {}
  template:
    metadata:
      annotations:
        kompose.cmd: kompose convert
        kompose.version: 1.26.0 (40646f47)
      creationTimestamp: null
      labels:
        io.kompose.service: infrastructurecatalogue
    spec:
      containers:
        - image: marievixezonaki/infrastructure-catalogue:latest
          name: infrastructurecatalogue
          ports:
            - containerPort: 8080
          resources: {}
          hostname: infrastructurecatalogue
          restartPolicy: Always
      status: {}
---
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
```

```
creationTimestamp: null
labels:
  io.kompose.service: infrastructurecatalogue
name: infrastructurecatalogue
spec:
  type: NodePort
  ports:
    - name: "8081"
      port: 8081
      targetPort: 8081
      # nodePort: 30007

  selector:
    io.kompose.service: infrastructurecatalogue
status:
  loadBalancer: {}
```

Για την εγκατάσταση του καταλόγου/διαχειριστή υποδομής TANDEM, εκτελούμε την παρακάτω εντολή:

```
kubectl apply -f infrastructurecatalogue.yaml -n tandem-system
```

5.2.4 Εγκατάσταση του TANDEM User Manager

Για την εγκατάσταση του καταλόγου/διαχειριστή των χρηστών του TANDEM σε Kubernetes χρησιμοποιήθηκε το παρακάτω Kubernetes yaml αρχείο:

```
usercatalogue.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    kompose.cmd: kompose convert
    kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: usercatalogue
  name: usercatalogue
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: usercatalogue
  strategy: {}
  template:
    metadata:
      annotations:
```

```

  kompose.cmd: kompose convert
  kompose.version: 1.26.0 (40646f47)
  creationTimestamp: null
  labels:
    io.kompose.service: usercatalogue
  spec:
    containers:
      - image: marievixezonaki/user-catalogue:latest
        name: usercatalogue
        ports:
          - containerPort: 8081
        resources: {}
      hostname: usercatalogue
      restartPolicy: Always
    status: {}
  ---
  apiVersion: v1
  kind: Service
  metadata:
    annotations:
      kompose.cmd: kompose convert
      kompose.version: 1.26.0 (40646f47)
    creationTimestamp: null
    labels:
      io.kompose.service: usercatalogue
  name: usercatalogue
  spec:
    type: NodePort
    ports:
      - name: "8081"
        port: 8080
        targetPort: 8080
    selector:
      io.kompose.service: usercatalogue
  status:
    loadBalancer: {}

```

Για την εγκατάσταση του καταλόγου/διαχειριστή χρηστών TANDEM, εκτελούμε την παρακάτω εντολή:

```
kubectl apply -f usercatalogue.yaml -n
tandem-system
```

5.2.5 Εγκατάσταση του TANDEM Edge Orchestrator

Για την εγκατάσταση του ενορχηστρωτή άκρων του TANDEM σε Kubernetes χρησιμοποιήθηκε το παρακάτω Kubernetes yaml αρχείο:

```
systemmanager.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    io.kompose.service: systemmanager
  name: systemmanager
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: systemmanager
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        #io.kompose.network/netEMPkub: "true"
        io.kompose.service: systemmanager
    spec:
      containers:
        - env:
            - name: EMP_STORAGE_URI
              value: mongodb://mongopiedge:27017
          image: marievixezonaki/system-manager:2.2
          name: systemmanager
          ports:
            - containerPort: 8083
          resources: {}
          imagePullPolicy: Always
        restartPolicy: Always
status: {}
---
```

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    io.kompose.service: systemmanager
  name: systemmanager
spec:
  type: NodePort
  ports:
    - name: "8083"
      nodePort: 30003
      port: 8083
      targetPort: 8083
  selector:
    io.kompose.service: systemmanager
status:
  loadBalancer: {}
```

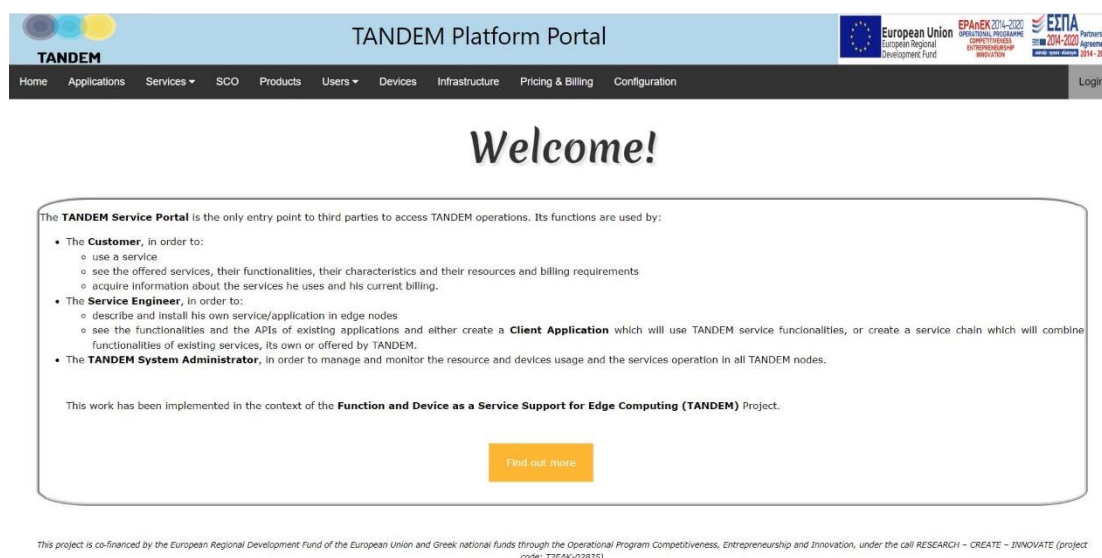
Για την εγκατάσταση του ενορχηστρωτή άκρων TANDEM, εκτελούμε την παρακάτω εντολή:

```
kubectl apply -f systemmanager.yaml -n  
tandem-system
```

6 Οδηγός Χρήσης

6.1 Κεντρική Σελίδα και Σύνδεση Χρήστη

Είτε ο χρήστης είναι συνδεδεμένος είτε όχι, όταν οδηγηθεί στο TANDEM, η εφαρμογή εμφανίζει την κεντρική ενημερωτική σελίδα (Εικόνα 17: Η κεντρική ενημερωτική σελίδα της πλατφόρμας TANDEM). Στην περίπτωση που ο χρήστης δεν είναι συνδεδεμένος, στο μενού που εμφανίζεται στη μπάρα της άνω πλευράς της σελίδας του δίνεται η επιλογή “Login” (“Σύνδεση”) ώστε να συνδεθεί στο λογαριασμό του. Εάν ο χρήστης επιλέξει να συνδεθεί σε υπάρχοντα λογαριασμό, τότε το παράθυρο που εμφανίζεται (Εικόνα 18: Το παράθυρο σύνδεσης του χρήστη στην πλατφόρμα TANDEM) ζητά τα διαπιστευτήρια του. Εάν ο χρήστης δεν είναι εγγεγραμμένος στο σύστημα, τότε μπορεί να κάνει αίτηση εγγραφής του πατώντας τον σύνδεσμο ‘register me’



Εικόνα 17: Η κεντρική ενημερωτική σελίδα της πλατφόρμας TANDEM

Μόλις ο χρήστης συνδεθεί επιτυχώς στην εφαρμογή, ανακατευθύνεται στην Αρχική σελίδα της Εφαρμογής όπου μπορεί να βρει μια λίστα με τις διαθέσιμες λειτουργίες στις οποίες έχει πρόσβαση βάσει του ρόλου του.

Εικόνα 18: Το παράθυρο σύνδεσης του χρήστη στην πλατφόρμα TANDEM

6.2 Διαχείριση Εφαρμογών

Όταν ο χρήστης επιλέξει την ενέργεια **“Applications”** (“Εφαρμογές”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των καταχωρημένων εφαρμογών στο TANDEM (Εικόνα 20: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων εφαρμογών στην πλατφόρμα). Για να δει την λίστα, ο χρήστης πρέπει να πατήσει το κουμπί «Submit» που εμφανίζεται στην δεξιά φόρμα, χωρίς απαραίτητα να εισάγει κάποια φίλτρα. Στην συνέχεια οι εφαρμογές εμφανίζονται σε μορφή πίνακα. Σε κάθε γραμμή του πίνακα παρουσιάζεται μία εφαρμογή, και πιο συγκεκριμένα τα εξής χαρακτηριστικά:

- **Όνομα (Name)** της εφαρμογής
- **Κατηγορία (Category)** στην οποία ανήκει η εφαρμογή, π.χ. Location-based services, IoT, Video Analytics κλπ.
- **Περιγραφή (Description)**, για την οποία εμφανίζονται οι 100 πρώτοι χαρακτήρες, ενώ η πλήρης περιγραφή εμφανίζεται όταν ο χρήστης περάσει πάνω από τις τρεις κουκίδες
- **Πάροχος (Provider)** της εφαρμογής π.χ. “INTRACOM S.A. Telecom Solutions”
- **Υπηρεσίες Εφαρμογής (Application Services)**, που αποτελούν τις κύριες υπηρεσίες που χρησιμοποιεί η συγκεκριμένη εφαρμογή για τη λειτουργία της
- **Υποστηρικτικές υπηρεσίες (Support Services)** που παρέχονται από την πλατφόρμα TANDEM, που αποτελούν βοηθητικές υπηρεσίες για τη λειτουργία της συγκεκριμένης εφαρμογής
- **Αλυσίδα υπηρεσιών (Service Chain)** η οποία δίνει το όνομα του διαγράμματος έργου (workflow) που περιγράφει την σύνδεση εμπλεκόμενων υπηρεσιών

- **Κατάσταση (State)** στην οποία βρίσκεται η εφαρμογή π.χ. “In Design”, “Approved (Active)”, “Instantiated”, “Rejected” κλπ.
- Διαθέσιμες **Ενέργειες (Actions)**. Για παράδειγμα, πατώντας το μπλε εικονίδιο “Details”, που φαίνεται στην Εικόνα 19: Εικονίδιο “Details” στις διαθέσιμες ενέργειες για μία εφαρμογή, ο χρήστης μπορεί να δει περισσότερες πληροφορίες για μία εφαρμογή.



Εικόνα 19: Εικονίδιο “Details” στις διαθέσιμες ενέργειες για μία εφαρμογή

TANDEM

TANDEM Platform Portal

European Union

European Regional Development Fund

EPANeK 2014-2020

OPERATIONAL PROGRAMME

COMPETITIVENESS AND INNOVATION

ΕΣΠΑ

2014-2020

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ

Hello, Thodoris!

HomeApplicationsServices▼SCOProductsUsers▼DevicesInfrastructurePricing & BillingConfiguration

Search for Application names..

Applications Filters

Name

Application name..

Categ.

Select▼

Provider

Select▼

State

Select▼

Submit

Name!#	Category!#	Description	Provider!#	Application Services	Support Services	Service Chain	State	Actions
Smart Parking	Smart City	It provides real-time information about the available parking spots of a parking lot. It includes a --	INTRACOM TELECOM S.A.	Parking Service	Object Detection Service, Camera 1 service	Object Detection SC1	Approved	
Physical Security	Surveillance	It uses cameras and Artificial Intelligence technologies for the automatic surveillance of an area (--)	INTRACOM TELECOM S.A.	Physical Security Service	Object Detection Service, Camera 2 service	Object Detection SC2	Rejected	
Temperature		It provides real-time monitoring of the	INTRACOM	IoT Data	Temperature and Humidity Monitoring Service, EdgeX Foundation			

Εικόνα 20: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων εφαρμογών στην πλατφόρμα

Η λίστα των εφαρμογών εμφανίζεται όταν ο χρήστης καθορίσει στην περιοχή φίλτρων στο δεξί μέρος της σελίδας τα επιθυμητά φίλτρα ανάλογα με τις εφαρμογές που θέλει να δει (ή κενά φίλτρα για εμφάνιση όλων) και πατήσει το κουμπί “Submit”. Τα διαθέσιμα φίλτρα που μπορεί να προσδιορίσει ο χρήστης είναι τα εξής:

- Όνομα
- Κατηγορία
- Πάροχος
- Κατάσταση

6.2.1 Αναζήτηση Εφαρμογών

Ο χρήστης μπορεί να αναζητήσει εφαρμογές με το όνομά τους μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα εφαρμογών, όπως φαίνεται στην Εικόνα 21: Γραμμή αναζήτησης εφαρμογών. Εναλλακτικά μπορεί να προσδιορίσει το όνομα της εφαρμογής στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.



Εικόνα 21: Γραμμή αναζήτησης εφαρμογών

6.2.2 Αλλαγή Κατάστασης Εφαρμογής

Ο χρήστης μπορεί να αλλάξει την κατάσταση μίας εφαρμογής πατώντας το πράσινο εικονίδιο “Next state” στη στήλη “Actions” της εφαρμογής προς αλλαγή κατάστασης, όπως φαίνεται στην Εικόνα 22: Εικονίδιο “Next state” στις διαθέσιμες ενέργειες για μία εφαρμογή.



Εικόνα 22: Εικονίδιο “Next state” στις διαθέσιμες ενέργειες για μία εφαρμογή

6.2.3 Διαγραφή Εφαρμογής

Ο χρήστης μπορεί να διαγράψει εφαρμογές πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” της εφαρμογής προς διαγραφή, όπως φαίνεται στην Εικόνα 23: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία εφαρμογή.



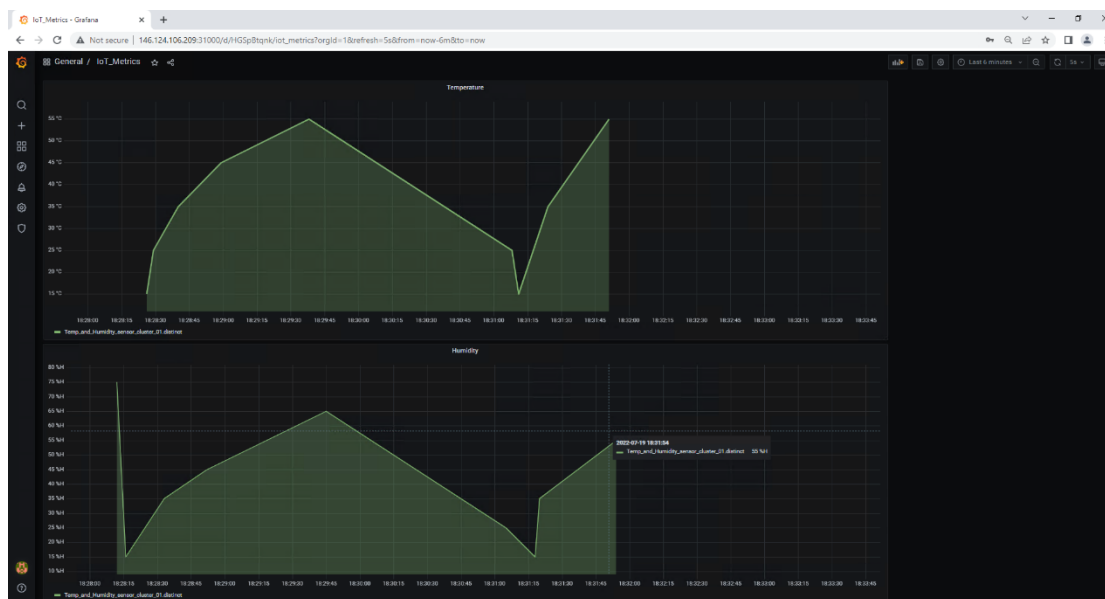
Εικόνα 23: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία εφαρμογή

6.2.4 Εκτέλεση Εφαρμογής

Ο χρήστης μπορεί να εκτελέσει εφαρμογές πατώντας το μπλε εικονίδιο “Run application” στη στήλη “Actions” της εφαρμογής προς εκτέλεση (Εικόνα 24: Εικονίδιο “Run application” στις διαθέσιμες ενέργειες για μία εφαρμογή). Με το πάτημα του εικονιδίου ανοίγει σε νέο παράθυρο ένα περιβάλλον Grafana το οποίο παρουσιάζει τις μετρήσεις της εφαρμογής, όπως φαίνεται στην Εικόνα 25: Η παρουσίαση των μετρήσεων της εφαρμογής κατά την εκτέλεσή της.



Εικόνα 24: Εικονίδιο “Run application” στις διαθέσιμες ενέργειες για μία εφαρμογή



Εικόνα 25: Η παρουσίαση των μετρήσεων της εφαρμογής κατά την εκτέλεσή της

6.2.5 Παρακολούθηση Χρήσης Πόρων Εφαρμογής

Ο χρήστης μπορεί να παρακολουθήσει τη χρήση πόρων από μία εφαρμογή πατώντας το μοβ εικονίδιο “Monitor” στη στήλη “Actions” της εφαρμογής προς παρακολούθηση (Εικόνα 26: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για μία εφαρμογή). Με το πάτημα του εικονιδίου ανοίγει σε νέο παράθυρο ένα περιβάλλον Grafana το οποίο παρουσιάζει την εικονικοποίηση των μετρικών χρήσης πόρων όλων των υπηρεσιών που αποτελούν την εφαρμογή, όπως φαίνεται στην Εικόνα 27: Παρουσίαση μετρικών χρήσης πόρων των υπηρεσιών μίας εφαρμογής.



Εικόνα 26: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για μία εφαρμογή



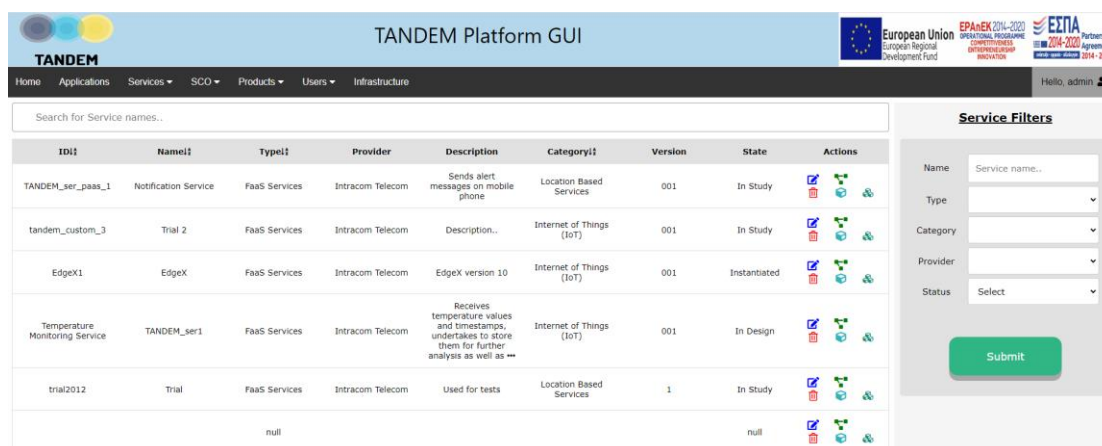
Εικόνα 27: Παρουσίαση μετρικών χρήσης πόρων των υπηρεσιών μίας εφαρμογής

6.3 Διαχείριση Υπηρεσιών (Κατάλογος Υπηρεσιών)

Όταν ο χρήστης επιλέξει την ενέργεια “Services” → “Management” (“Υπηρεσίες” → “Διαχείριση”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των καταχωρημένων υπηρεσιών στο TANDEM (Εικόνα 28: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων υπηρεσιών στην πλατφόρμα). Σε κάθε γραμμή του πίνακα παρουσιάζεται μία υπηρεσία, και πιο συγκεκριμένα τα εξής χαρακτηριστικά της:

- **Μοναδικό αναγνωριστικό (ID)** της υπηρεσίας
- **Όνομα (Name)** της υπηρεσίας
- **Τύπος (Type)** της υπηρεσίας, π.χ. FaaS Services, Device Services κλπ.
- **Πάροχος (Provider)** της υπηρεσίας π.χ. “INTRACOM S.A. Telecom Solutions”

- **Περιγραφή (Description)**, για την οποία εμφανίζονται οι 100 πρώτοι χαρακτήρες, ενώ η πλήρης περιγραφή εμφανίζεται όταν ο χρήστης περάσει πάνω από τις τρεις κουκίδες
- **Κατηγορία (Category)** στην οποία ανήκει η υπηρεσία, π.χ. Location-based services, IoT, Video Analytics κλπ.
- **Έκδοση (Version)** της υπηρεσίας
- **Κατάσταση (State)** στην οποία βρίσκεται η υπηρεσία, π.χ. “In Design”, “Approved (Active)”, “Instantiated”, “Rejected” κλπ.
- Διαθέσιμες **Ενέργειες (Actions)**, π.χ. Αλλαγή (Edit), Διαγραφή (Delete) κλπ.



ID#	Name#	Type#	Provider	Description	Category#	Version	State	Actions
TANDEM_ser_paaas_1	Notification Service	FaaS Services	Intracom Telecom	Sends alert messages on mobile phone	Location Based Services	001	In Study	[Edit] [Delete] [Refresh]
tandem_custom_3	Trial 2	FaaS Services	Intracom Telecom	Description...	Internet of Things (IoT)	001	In Study	[Edit] [Delete] [Refresh]
EdgeX1	EdgeX	FaaS Services	Intracom Telecom	EdgeX version 10	Internet of Things (IoT)	001	Instantiated	[Edit] [Delete] [Refresh]
Temperature Monitoring Service	TANDEM_ser1	FaaS Services	Intracom Telecom	Receives temperature values and timestamps, undertakes to store them for further analysis as well as ...	Internet of Things (IoT)	001	In Design	[Edit] [Delete] [Refresh]
trial2012	Trial	FaaS Services	Intracom Telecom	Used for tests	Location Based Services	1	In Study	[Edit] [Delete] [Refresh]
							null	[Edit] [Delete] [Refresh]

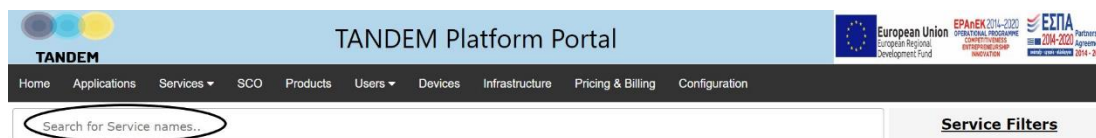
Εικόνα 28: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων υπηρεσιών στην πλατφόρμα

Η λίστα των υπηρεσιών εμφανίζεται όταν ο χρήστης επιλέξει την σελίδα από το μενού. Επίσης, ο χρήστης μπορεί να καθορίσει, στην περιοχή φίλτρων στο δεξί μέρος της σελίδας, τα επιθυμητά φίλτρα ανάλογα με τις υπηρεσίες που θέλει να δει (ή κενά φίλτρα για εμφάνιση όλων). Τα φίλτρα εφαρμόζονται όταν πατήσει το κουμπί “Submit”. Τα διαθέσιμα φίλτρα που μπορεί να προσδιορίσει ο χρήστης είναι τα εξής:

- Όνομα
- Τύπος
- Κατηγορία
- Πάροχος
- Κατάσταση

6.3.1 Αναζήτηση Υπηρεσίας

Ο χρήστης μπορεί να αναζητήσει υπηρεσίες με το όνομά τους μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα υπηρεσιών, όπως φαίνεται στην Εικόνα 29. Εναλλακτικά μπορεί να προσδιορίσει το όνομα της υπηρεσίας στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.



Εικόνα 29: Γραμμή αναζήτησης υπηρεσιών

6.3.2 Προβολή Λεπτομερειών Υπηρεσίας και Αλλαγή των Στοιχείων της

Ο χρήστης μπορεί να δει περισσότερες πληροφορίες για μία υπηρεσία καθώς και να μεταβάλλει τα χαρακτηριστικά της πατώντας το μπλε εικονίδιο “Edit” στη στήλη “Actions” της υπηρεσίας προς προβολή, όπως φαίνεται στην Εικόνα 30.



Εικόνα 30: Εικονίδιο “Edit” στις διαθέσιμες ενέργειες για μία υπηρεσία

Επιλέγοντας το παραπάνω εικονίδιο, θα ανοίξει, σε νέα καρτέλα στο ίδιο παράθυρο, η σελίδα που παρουσιάζει όλα τα δεδομένα της υπηρεσίας, όπως φαίνεται στην Εικόνα 31. Αυτή η σελίδα επιτρέπει την μεταβολή των στοιχείων της υπηρεσίας. Οι αλλαγές αποθηκεύονται όταν ο χρήστης πατήσει το κουμπί “Submit” στο τέλος της σελίδας.

TANDEM

TANDEM Platform GUI

European Union
European Regional Development Fund

EPANEX 2014-2020
OPERATIONAL PROGRAMME
INTELLIGENT GROWTH
INNOVATION

ΕΣΠΑ
2014-2020
Partnership Agreement

Home Applications Services SCO Products Users Infrastructure

Hello, admin

Search for Service names...

Idi:	Name:	Type:	Provider	Description	Category:	Version	State	Actions
TANDEM_ser_paas_1	Notification Service	FaaS Services	Intracom Telecom	Sends alert messages on mobile phone	Location Based Services	001	In Study	
tandem_custom_3	Trial 2	FaaS Services	Intracom Telecom	Description...	Internet of Things (IoT)	001	In Study	
EdgeX1	EdgeX	FaaS Services	Intracom Telecom	EdgeX version 10	Internet of Things (IoT)	001	Instantiated	
Temperature Monitoring Service	TANDEM_ser1	FaaS Services	Intracom Telecom	Receives temperature values and timestamps, undertakes to store them for further analysis as well as ...	Internet of Things (IoT)	001	In Design	
trial2012	Trial	FaaS Services	Intracom Telecom	Used for tests	Location Based Services	1	In Study	
							null	

Service Filters

Name:

Type:

Category:

Provider:

Status:

TANDEM Platform Portal

Home Applications Services SCD Products Users Infrastructure

European Union European Regional Development Fund
ΕΡΑΝΕΚ 2014-2020
ΕΣΠΑ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

Service Registration / Editing

Service Name: Notification Service
Service Id: TANDEM_ser_paaS_1
Service Category: Location Based Services
Service Description: Sends alert messages on mobile phone
Service Type: FaaS
Service Version: 001
Service Provider: Intracom Telecom
Service State: In Study

Service Description & APIs URLs

URL	Action
tandem-project.gr/Services/NonService	
edgeservices.org/TandemNonService	

[+ Add URL](#)

Configuration Parameters

Parameter Name	Parameter Type	Parameter Typical Value	Parameter Description	Action
Smtp_server	string	string	E-mail server address	

[+ Add Config Parameters](#)

Operations

Operation Name	Operation Description	Operation Type	Endpoint	Actions
SendNotification	Send Notifications	Synchronous	/TCOM/NotificationService/SendNotification	

[+ Add Operation](#)

Computational & Storage Requirements

Requirement	Value	Measurement Unit
Memory Size	5	Gb
Number of Virtual CPUs	50	millicores
Number of Virtual GPUs	0	millicores
Storage Requirements	0	Gb

Communication Requirements

Requirement	Value	Measurement Unit
Latency	1000	ms
Throughput	0.01	mbps

Required Services

Name	Description	Action
e-mail server	e-mail server	

[+ Add Service](#)

Optional Services

Name	Description	Action
E-mail server	E-mail server	

[+ Add Service](#)

Software Image Parameters

Parameter Name	Parameter Value
SW Image Name	SWImage00015
SW Image Container Format	Docker
SW Image Size in Mb	0.5
SW Image OS	Ubuntu 22.04 LTS
SW Image Architecture	
SW Image URL	/TCOM/SWImages

Consumed Local: Yes
Is Local: Yes
Scope of Locality: Internal use

[Submit](#)

Εικόνα 31: Λειτουργικότητα του εικονιδίου "Edit" στις διαθέσιμες ενέργειες για μία υπηρεσία

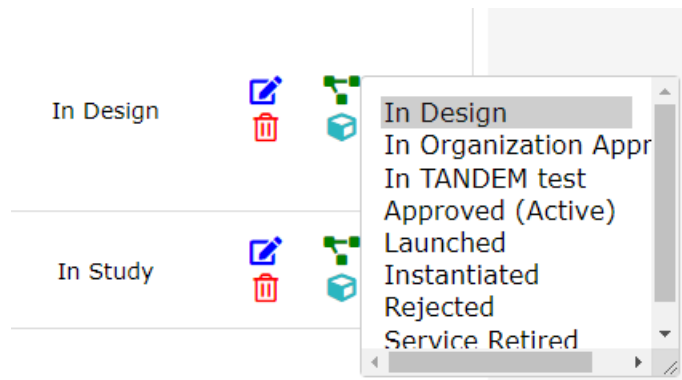
6.3.3 Αλλαγή Κατάστασης Υπηρεσίας

Ο χρήστης μπορεί να αλλάξει την κατάσταση μίας υπηρεσίας πατώντας το πράσινο εικονίδιο "Change state" στη στήλη "Actions" της υπηρεσίας προς αλλαγή κατάστασης, όπως φαίνεται στην Εικόνα 32: Εικονίδιο "Change state" στις διαθέσιμες ενέργειες για μία υπηρεσία.



Εικόνα 32: Εικονίδιο "Change state" στις διαθέσιμες ενέργειες για μία υπηρεσία

Ανάλογα με την τρέχουσα κατάσταση της υπηρεσίας, το “Change state” εμφανίζει μια λίστα με τις πιθανές καταστάσεις στις οποίες μπορεί να μεταβεί η υπηρεσία, όπως φαίνεται στην Εικόνα 33. Για να πραγματοποιηθεί η αλλαγή, ο χρήστης πρέπει να επιλέξει τη νέα κατάσταση και μετά να ξαναπατήσει το εικονίδιο “Change state”.



Εικόνα 33: Λειτουργικότητα του εικονιδίου “Change state” στις διαθέσιμες ενέργειες για μία υπηρεσία

6.3.4 Διαγραφή Υπηρεσίας

Ο χρήστης μπορεί να διαγράψει υπηρεσίες πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” της υπηρεσίας προς διαγραφή, όπως φαίνεται στην Εικόνα 34: *Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία υπηρεσία.*



Εικόνα 34: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία υπηρεσία

6.3.5 Δημιουργία Προϊόντος από μια Υπηρεσία

Ο χρήστης μπορεί να επιλέξει μια υπηρεσία από τον πίνακα υπηρεσιών και, με βάση αυτήν, να δημιουργήσει το αντίστοιχο προϊόν πατώντας το γαλάζιο εικονίδιο “Create product” στη στήλη “Actions” της υπηρεσίας, όπως φαίνεται στην Εικόνα 35.



Εικόνα 35: Εικονίδιο “Create product” στις διαθέσιμες ενέργειες για μία υπηρεσία

Πατώντας αυτό το κουμπί, ανοίγει, σε νέα καρτέλα στο ίδιο παράθυρο, η σελίδα δημιουργίας προϊόντος, στην οποία κάποια πεδία που σχετίζονται με την υπηρεσία είναι

προσυμπληρωμένα (Εικόνα 36). Η σελίδα αυτή περιγράφεται με περισσότερες λεπτομέρειες στην Ενότητα 6.8.5.

TANDEM Platform Portal

Home Applications Services SCO Products Users Infrastructure Hello, admin

Product Registration / Editing

Product Name * Product name...

Product Category * Internet of Things (IoT)

Product Provider * Intracom Telecom

Product Service * TANDEM_ser1

Product Description: Receives temperature values and timestamps, undertakes to store them for further analysis as well as to send notifications or to activate an alarm depending on the temperature values

Is Buddle ☐ Is Sellable ☐

Product Price Per Charge Unit * Select

Valid for * mm/dd/yyyy

Product Service Level Agreement (SLA) Select

Product Id * Product id...

Product Type * Faas

Product Version *

Product Application * Select

Life Cycle Status * Select

Product Pricing Model * Select

Currency * Select

Add SLA

Εικόνα 36: Φόρμα εγγραφής προϊόντος στην πλατφόρμα

6.3.6 Καταχώρηση Υπηρεσίας

Ο χρήστης έχει τη δυνατότητα να καταχωρήσει μία νέα υπηρεσία, επιλέγοντας την ενέργεια **“Services”** → **“Registration ”** (“Υπηρεσίες” → “Εγγραφή”) από το κεντρικό μενού. Η εφαρμογή τότε εμφανίζει τη σελίδα που φιλοξενεί τη φόρμα εγγραφής υπηρεσιών στο TANDEM (Εικόνα 37).

TANDEM Platform Portal

Home Applications Services SCO Products Users Infrastructure Hello, admin

Service Registration / Editing

Service Name *
Service ID *
Service Category *
Service Description

Service name...
Service ID...
Select
Description...

Service Type *
Service Version
Service Provider *
Service State *

Select
Service Version...
Select
Select

Service Description & APIs URLs	
URL	Action
tandem-project.gr/Services/NonService	
edgeservices.org/TandemNonService	

+ Add URL

Configuration Parameters				
Parameter Name	Parameter Type	Parameter Typical Value	Parameter Description	Action
smtp_server	string	smtp.tandem.com:587	E-mail server address	

+ Add Config Parameters

Operations				
Operation Name	Operation Description	Operation Type	Endpoint	Actions
SendNotification	Send Notifications	Synchronous	/TCOM/NotificationSer	

+ Add Operation

Computational & Storage Requirements		
Requirement	Value	Measurement Unit
Memory Size	5	Gb
Number of Virtual CPUs	50	millicores
Number of Virtual GPUs	0	millicores
Storage Requirements	0	Gb

Communication Requirements		
Requirement	Value	Measurement Unit
Latency	1000	ms
Throughput	0.01	Mbps

Required Services		
Name	Description	Action
E-mail server	E-mail Server	

+ Add Service

Optional Services		
Name	Description	Action
E-mail server	E-mail Server	

+ Add Service

Software Image Parameters	
Parameter Name	Parameter Value
SW Image Name	SWImage00015
SW Image Container Format	Docker
SW Image Size in Mb	0.5
SW Image OS	Ubuntu 22.04 LTS
SW Image Architecture	x86-64
SW Image URL	/TCOM/SWimages

Consumed Local *
Is Local *
Scope of Locality

Select
Select
Locality scope...

Submit

Εικόνα 37: Φόρμα εγγραφής υπηρεσίας στην πλατφόρμα

Τα στοιχεία προς συμπλήρωση στη φόρμα εγγραφής νέας υπηρεσίας είναι τα ακόλουθα:

- Όνομα (Service Name) της υπηρεσίας
- Τύπος (Service Type) της υπηρεσίας
- Μοναδικό αναγνωριστικό (Service Id) της υπηρεσίας
- Έκδοση (Service Version) της υπηρεσίας
- Κατηγορία (Service Category) στην οποία ανήκει η υπηρεσία
- Πάροχος (Service Provider) της υπηρεσίας
- Περιγραφή (Service Description) της υπηρεσίας
- Κατάσταση (Service State) στην οποία βρίσκεται η υπηρεσία
- Σύνδεσμοι περιγραφών και διεπαφών υπηρεσιών (Service Description & APIs URLs), οι οποίοι καταχωρούνται σε δικό τους πίνακα και μπορούν να διαγραφούν ή να προστεθούν νέοι
- Παράμετροι Διαμόρφωσης (Configuration Parameters) της υπηρεσίας, οι οποίες καταχωρούνται σε δικό τους πίνακα και αποτελούνται από:
 - Όνομα (Parameter Name) της παραμέτρου

- Τύπο (Parameter Type) της παραμέτρου, ο οποίος μπορεί να είναι συμβολοσειρά (string), αριθμός (number), δυαδικός (boolean), κενό (null), αντικείμενο (object) ή πίνακας (array)
- Τυπική τιμή (Parameter Typical Value) της παραμέτρου
- Περιγραφή (Parameter Description) της παραμέτρου
- Ενέργεια Διαγραφής (Delete Action) της παραμέτρου
- **Λειτουργίες (Operations)** της υπηρεσίας, οι οποίες καταχωρούνται σε δικό τους πίνακα και αποτελούνται από:
 - Όνομα (Operation Name) της λειτουργίας
 - Περιγραφή (Operation Description) της λειτουργίας
 - Τύπο (Operation Type) της λειτουργίας, ο οποίος μπορεί να είναι Σύγχρονος (Synchronous) ή Ασύγχρονος (Asynchronous)
 - Τερματικό σημείο (Endpoint) της λειτουργίας, που αποτελεί το σημείο πρόσβασης στη συγκεκριμένη λειτουργία
 - Ενέργειες (Actions) εμφάνισης λεπτομερειών για τη λειτουργία καθώς και διαγραφή της
 - Το εικονίδιο εμφάνισης λεπτομερειών ("Details") επιτρέπει τον ορισμό παραμέτρων εισόδου και εξόδου για την λειτουργία
- **Υπολογιστικές και Αποθηκευτικές Απαιτήσεις (Computational & Storage Requirements)** της υπηρεσίας, καθεμία από τις οποίες προσδιορίζεται από την Απαίτηση (Requirement), την Τιμή (Value) και τη Μονάδα Μέτρησης (Measurement Unit). Οι απαιτήσεις που πρέπει να προσδιοριστούν είναι:
 - Μέγεθος Μνήμης (Memory Size) που απαιτεί η υπηρεσία σε Gb
 - Αριθμός εικονικών Μονάδων Επεξεργασίας (Number of Virtual CPUs) που απαιτεί η υπηρεσία σε millicores
 - Αριθμός εικονικών Μονάδων Επεξεργασίας Γραφικών (Number of Virtual GPUs) που απαιτεί η υπηρεσία σε millicores
 - Αποθηκευτικές Απαιτήσεις (Storage Requirements) της υπηρεσίας σε Gb
- **Απαιτήσεις Επικοινωνίας (Communication Requirements)** της υπηρεσίας, καθεμία από τις οποίες προσδιορίζεται από την Απαίτηση (Requirement), την Τιμή (Value) και τη Μονάδα Μέτρησης (Measurement Unit). Οι απαιτήσεις που πρέπει να προσδιοριστούν είναι:
 - Καθυστερήση (Latency) σε ms
 - Ρυθμός δεδομένων (Throughput) σε Mbps
- **Απαιτούμενες Υπηρεσίες (Required Services)** της υπηρεσίας, οι οποίες καταχωρούνται σε δικό τους πίνακα και αποτελούνται από:
 - Όνομα (Name) της απαιτούμενης υπηρεσίας
 - Περιγραφή (Description) της απαιτούμενης υπηρεσίας
 - Ενέργεια Διαγραφής (Delete Action) της απαιτούμενης υπηρεσίας
- **Προαιρετικές Υπηρεσίες (Optional Services)** της υπηρεσίας, οι οποίες καταχωρούνται σε δικό τους πίνακα και αποτελούνται από:
 - Όνομα (Name) της προαιρετικής υπηρεσίας
 - Περιγραφή (Description) της προαιρετικής υπηρεσίας
 - Ενέργεια Διαγραφής (Delete Action) της προαιρετικής υπηρεσίας
- **Παράμετροι Εικόνας Λογισμικού (Software Image Parameters)** της υπηρεσίας, οι οποίες καταχωρούνται σε δικό τους πίνακα και αποτελούνται από:
 - Όνομα Εικόνας Λογισμικού (SW Image Name)
 - Μορφή Container Εικόνας Λογισμικού (SW Image Container Format)
 - Μέγεθος Εικόνας Λογισμικού σε Mb (SW Image Size in Mb)
 - Λειτουργικό Σύστημα Εικόνας Λογισμικού (SW Image OS)
 - Αρχιτεκτονική Εικόνας Λογισμικού (SW Image Architecture)

- Σύνδεσμος Εικόνας Λογισμικού (SW Image URL)
- **Τοπική Κατανάλωση (Consumed Local)** της υπηρεσίας, που μπορεί να παίρνει καταφατική ή αρνητική τιμή
- **Τοπικότητα (Is Local)** της υπηρεσίας, που μπορεί να παίρνει καταφατική ή αρνητική τιμή
- **Πλαίσιο Τοπικότητας (Scope of Locality)** της υπηρεσίας.

6.3.7 Τροποποίηση Υπηρεσίας

Ο χρήστης έχει τη δυνατότητα να τροποποιήσει υπάρχουσες υπηρεσίες με δύο τρόπους:

1. Επιλέγοντας την ενέργεια **“Services”** → **“Registration”** (“Υπηρεσίες” → “Εγγραφή”) από το κεντρικό μενού. Σε αυτήν την περίπτωση θα πρέπει να γωρίζει το μοναδικό αναγνωριστικό (Id) της υπηρεσίας και να το εισάγει στο κατάλληλο πεδίο.
2. Μεταβαίνοντας στην σελίδα **“Services”** → **“Management”** (“Υπηρεσίες” → “Διαχείριση”) και επιλέγοντας την ενέργεια **“Edit”** για την υπηρεσία που θέλει να τροποποιήσει.

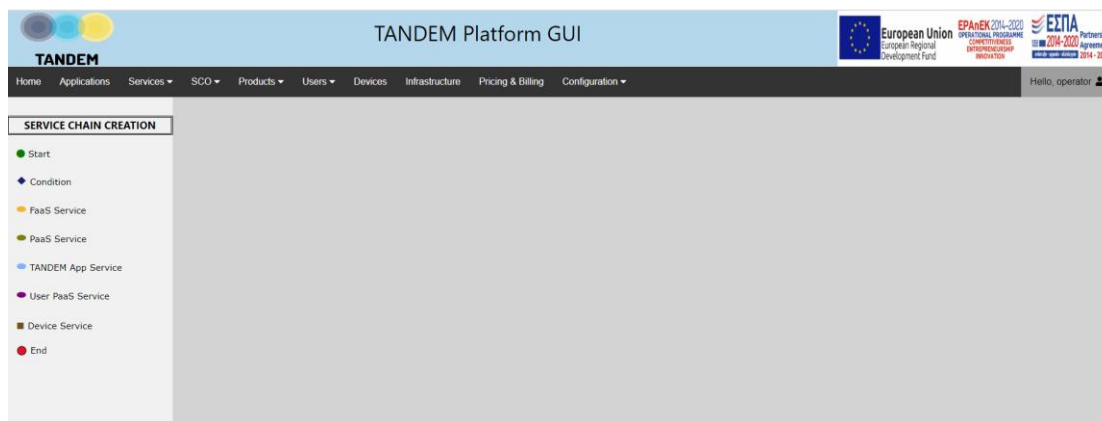
Και στους δύο τρόπους, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη φόρμα τροποποίησης υπηρεσιών στο TANDEM, η οποία είναι ίδια με τη φόρμα εγγραφής υπηρεσίας και εμφανίζεται στην Εικόνα 37: Φόρμα εγγραφής υπηρεσίας στην πλατφόρμα.

6.4 Διαχείριση Αλυσίδων Υπηρεσιών

Στην πλατφόρμα TANDEM υπάρχει δυνατότητα δημιουργίας αλυσίδων υπηρεσιών, δηλαδή συνδέσεων επιλεγμένων υπηρεσιών με σκοπό την δημιουργία μιας ενιαίας εφαρμογής. Ο Ενορχηστρωτής Αλυσίδων Υπηρεσιών (Service Chain Orchestrator - SCO) αναλαμβάνει να πραγματοποιήσει τη σύνδεση των επιλεγμένων υπηρεσιών καθώς και την ενορχήστρωσή τους. Πραγματοποιεί αναζήτηση στον Κατάλογο Υπηρεσιών για να βρει τις υπηρεσίες που χρειάζεται για να υλοποιήσει μια συγκεκριμένη εφαρμογή. Στην συνέχεια, σύμφωνα με τον τρόπο που ορίζονται οι υπηρεσίες και τα APIs τους, είναι σε θέση να πραγματοποιήσει την σύνθεση υπηρεσιών (service chains) που επιθυμεί.

6.4.1 Δημιουργία Αλυσίδας Υπηρεσιών

Όταν ο χρήστης επιλέξει την ενέργεια **“SCO”** (“Ενορχηστρωτής Αλυσίδων Υπηρεσιών”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί το περιβάλλον δημιουργίας αλυσίδας υπηρεσιών και ο χρήστης βρίσκεται σε κατάσταση άμεσης δημιουργίας αλυσίδας (Εικόνα 38).



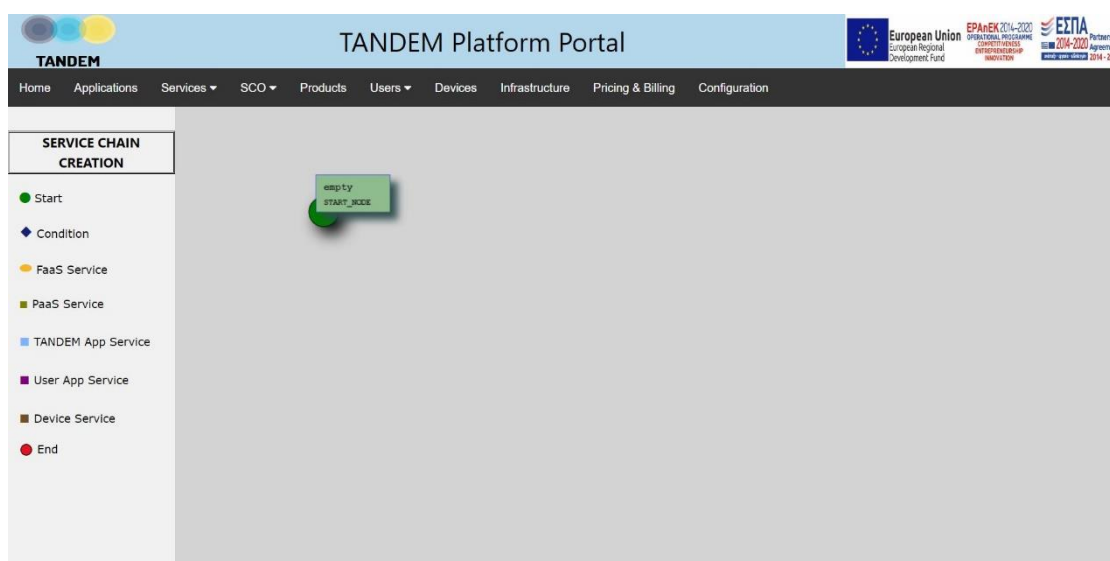
Εικόνα 38: Το περιβάλλον δημιουργίας αλυσίδας υπηρεσιών

6.4.2 Προσθήκη κόμβου

Για να προσθέσει έναν κόμβο στην αλυσίδα, ο χρήστης πρώτα επιλέγει έναν από τους διαθέσιμους κόμβους στο μενού που βρίσκεται στα αριστερά της σελίδας, και στη συνέχεια τον εμφανίζει με “κλικ” στο σημείο που επιθυμεί στο σκούρο γκρι πλαίσιο στα δεξιά της σελίδας. Οι διαθέσιμοι τύποι κόμβων είναι:

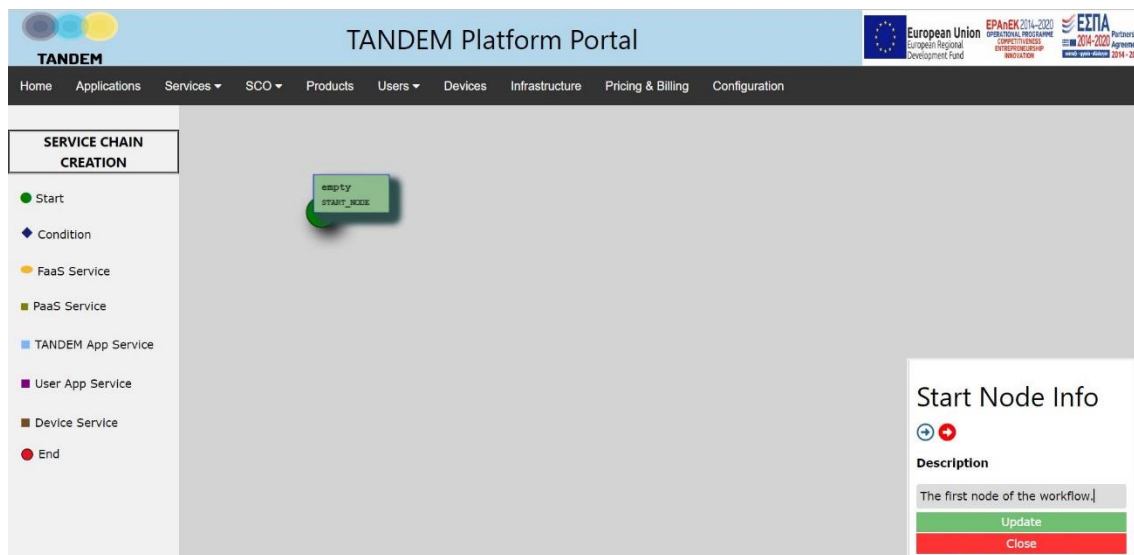
- Start
- Condition
- FaaS Service
- PaaS Service
- TANDEM App Service
- User PaaS Service
- Device Service
- End

Ο πρώτος κόμβος που εισάγεται στην κάθε αλυσίδα πρέπει απαραίτητως να είναι το “Start” (Εικόνα 39).

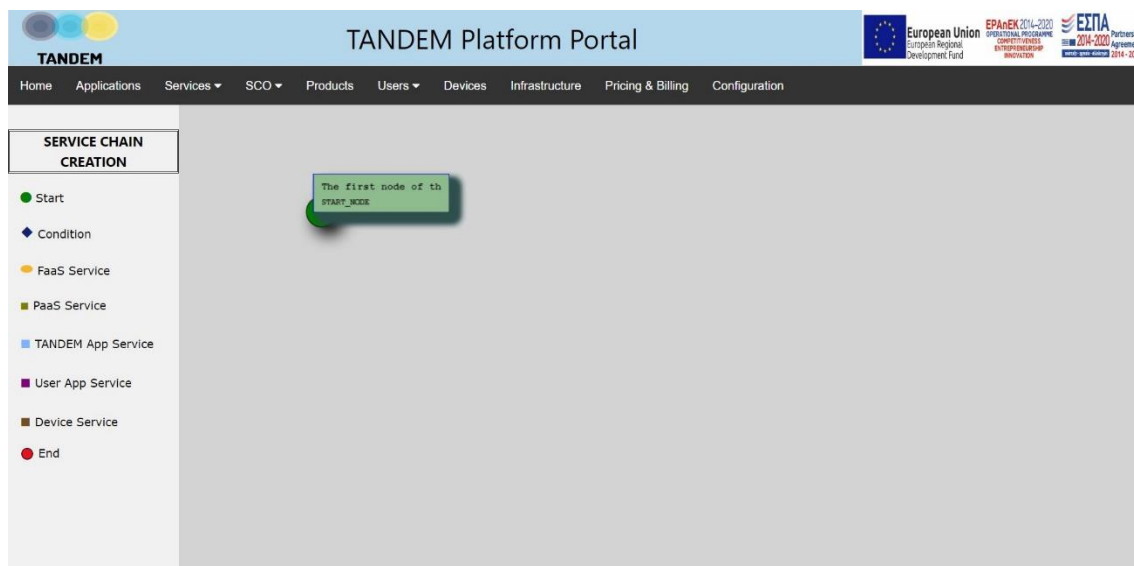


Εικόνα 39: Η εισαγωγή του “Start” ως πρώτου κόμβου της αλυσίδας υπηρεσιών

Ο κάθε κόμβος διαθέτει τη δική του ετικέτα, με μία σύντομη περιγραφή και το όνομά του. Όπως φαίνεται στην Εικόνα 39: Η εισαγωγή του “Start” ως πρώτου κόμβου της αλυσίδας υπηρεσιών, αρχικά η περιγραφή του κόμβου είναι κενή. Πατώντας πάνω σε ένα κόμβο, εμφανίζεται στο κάτω δεξιά μέρος της οθόνης σχετικό παράθυρο όπου ο χρήστης μπορεί να προσδιορίσει παραμέτρους σχετικές με τον κόμβο. Στην Εικόνα 40: Το αναδυόμενο παράθυρο παραμέτρων ενός κόμβου προσδιορίζεται η περιγραφή του κόμβου “Start” και στην Εικόνα 41 εμφανίζεται η ανανεωμένη περιγραφή στην ετικέτα του κόμβου.



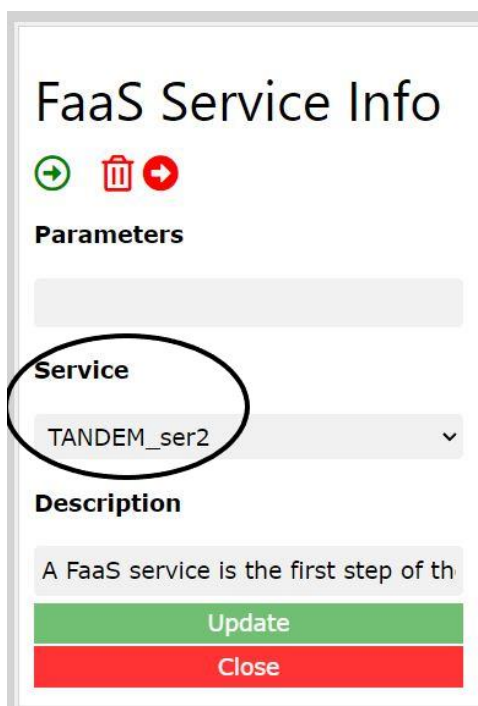
Εικόνα 40: Το αναδυόμενο παράθυρο παραμέτρων ενός κόμβου



Εικόνα 41: Η ανανεωμένη ετικέτα ενός κόμβου μετά από επεξεργασία της από τον χρήστη

6.4.2.1 Συσχέτιση κόμβου με υπηρεσία

Μετά την προσθήκη ενός κόμβου στην αλυσίδα, υπάρχει η δυνατότητα συσχετισμού του με μία από τις καταχωρημένες υπηρεσίες στην πλατφόρμα TANDEM. Ο συσχετισμός πραγματοποιείται πατώντας πάνω στον κόμβο και επιλέγοντας την κατάλληλη υπηρεσία από τη λίστα “Service” στο παράθυρο που εμφανίζεται στο κάτω δεξιά μέρος της σελίδας (Εικόνα 42).



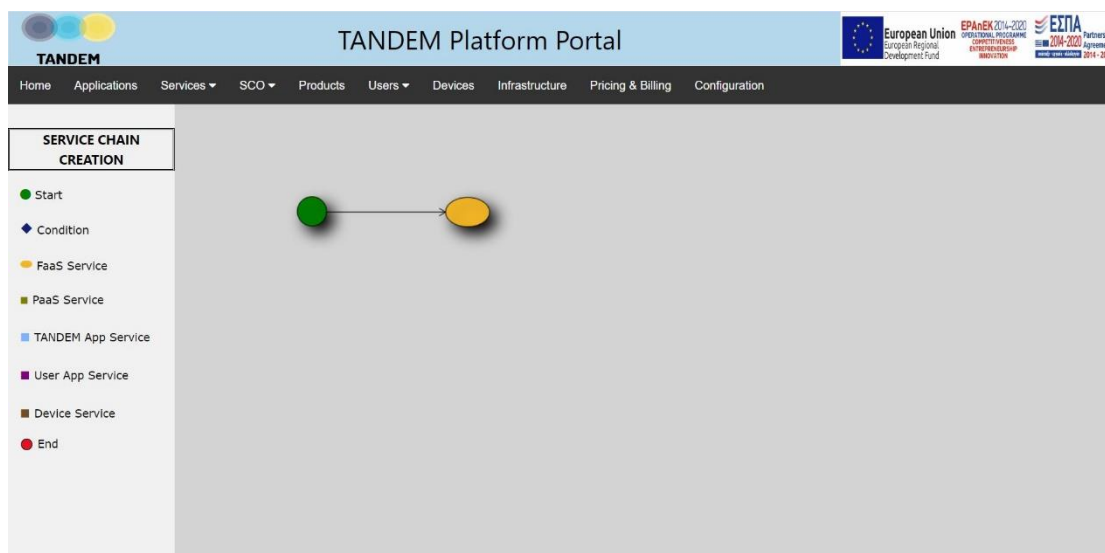
Εικόνα 42: Το παράθυρο συσχέτισης ενός κόμβου με μία υπηρεσία

6.4.2.2 Συσχέτιση κόμβου με άλλον κόμβο

Μετά την προσθήκη κόμβων στην αλυσίδα, πρέπει να τοποθετηθούν και οι απαραίτητοι σύνδεσμοι μεταξύ τους. Η διαδικασία ξεκινάει με την επιλογή του κόμβου από τον οποίο θα προέρχεται ο σύνδεσμος, στη συνέχεια με την επιλογή του εικονιδίου βέλους “Add Link” όπως φαίνεται στην Εικόνα 43 και τέλος την επιλογή του κόμβου στον οποίο θα καταλήγει ο σύνδεσμος. Το αποτέλεσμα της διαδικασίας είναι η προσθήκη συνδέσμου μεταξύ των δύο κόμβων (Εικόνα 44).



Εικόνα 43: Το εικονίδιο βέλους “Add Link” σε έναν κόμβο για την προσθήκη συνδέσμου



Εικόνα 44: Η σύνδεση μεταξύ δύο κόμβων

6.4.2.3 Διαγραφή Συσχέτισης κόμβων

Για να διαγραφεί η συσχέτιση μεταξύ δύο κόμβων, ο χρήστης πρέπει να επιλέξει οποιοδήποτε από τα δύο άκρα του συνδέσμου και να πατήσει το κόκκινο εικονίδιο βέλους “Remove Links” στο αναδυόμενο παράθυρο που εμφανίζεται (Εικόνα 45).



Εικόνα 45: Το εικονίδιο βέλους “Remove Links” σε έναν κόμβο για την αφαίρεση συνδέσμων

6.4.2.4 Διαγραφή Κόμβου

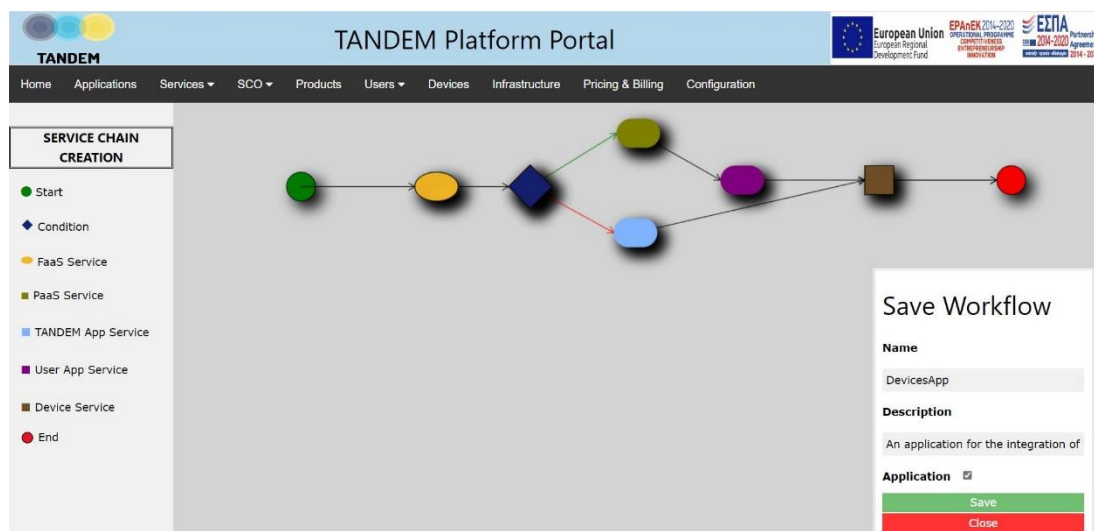
Ο χρήστης έχει τη δυνατότητα να διαγράψει έναν κόμβο από την αλυσίδα υπηρεσιών, επιλέγοντας τον κόμβο προς διαγραφή και πατώντας το κόκκινο εικονίδιο “Remove Node” στο αναδυόμενο παράθυρο που εμφανίζεται (Εικόνα 46).



Εικόνα 46: Το εικονίδιο “Remove Node” σε έναν κόμβο για την αφαίρεσή του

6.4.3 Αποθήκευση Αλυσίδας Υπηρεσιών

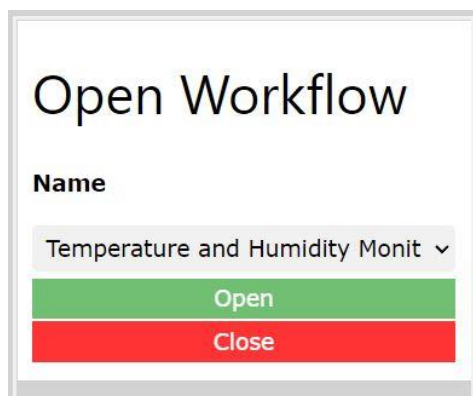
Αφού έχει ολοκληρωθεί η δημιουργία μίας αλυσίδας υπηρεσιών, ο χρήστης μπορεί να την αποθηκεύσει για μελλοντική χρήση στην πλατφόρμα TANDEM επιλέγοντας την ενέργεια “SCO” → “Save” (“Ενορχηστρωτής Αλυσίδων Υπηρεσιών” → “Αποθήκευση”) από το κεντρικό μενού. Σε σχετικό παράθυρο επιλογών αποθήκευσης της αλυσίδας που εμφανίζεται στο κάτω δεξιό μέρος της σελίδας, ο χρήστης μπορεί να προσδιορίσει το όνομα και μία περιγραφή της αλυσίδας, καθώς και να σημειώσει εάν πρόκειται για εφαρμογή ή όχι (Εικόνα 47). Μόλις πατήσει το κουμπί “Save” στο παράθυρο, η αλυσίδα υπηρεσιών αποθηκεύεται και εμφανίζεται σχετικό μήνυμα.



Εικόνα 47: Το αναδυόμενο παράθυρο αποθήκευσης μίας αλυσίδας υπηρεσιών

6.4.4 Επεξεργασία Υπάρχουσας Αλυσίδας

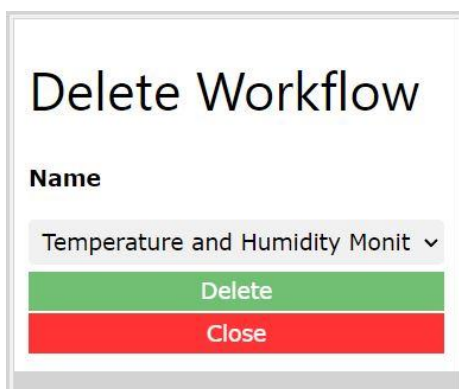
Ο χρήστης έχει τη δυνατότητα να ανοίξει στην οθόνη του μία αποθηκευμένη αλυσίδα υπηρεσιών και να την επεξεργαστεί. Όταν επιλεγεί η ενέργεια **“SCO”** → **“Open”** (“Ενορχηστρωτής Αλυσίδων Υπηρεσιών” → “Ανοιγμα”) από το κεντρικό μενού, η εφαρμογή εμφανίζει παράθυρο επιλογής της αλυσίδας προς εμφάνιση στο κάτω δεξιά μέρος της σελίδας (Εικόνα 48). Μόλις ο χρήστης επιλέξει την επιθυμητή αλυσίδα και πατήσει το κουμπί **“Open”**, αυτή εμφανίζεται στη σελίδα.



Εικόνα 48: Το αναδυόμενο παράθυρο ανοίγματος και επεξεργασίας υπάρχουσας αλυσίδας υπηρεσιών

6.4.5 Διαγραφή Αλυσίδας Υπηρεσιών

Ο χρήστης έχει τη δυνατότητα να διαγράψει μία αποθηκευμένη αλυσίδα υπηρεσιών. Όταν επιλεγεί η ενέργεια **“SCO”** → **“Delete”** (“Ενορχηστρωτής Αλυσίδων Υπηρεσιών” → “Διαγραφή”) από το κεντρικό μενού, η εφαρμογή εμφανίζει παράθυρο επιλογής της αλυσίδας προς διαγραφή στο κάτω δεξιά μέρος της σελίδας (Εικόνα 49). Μόλις ο χρήστης επιλέξει την επιθυμητή αλυσίδα και πατήσει το κουμπί **“Delete”**, αυτή διαγράφεται από την πλατφόρμα TANDEM.



Εικόνα 49: Το αναδυόμενο παράθυρο διαγραφής μίας αλυσίδας υπηρεσιών

6.5 Διαχείριση Χρηστών

Όταν επιλεγεί η ενέργεια “Users” → “Management” (“Χρήστες” → “Διαχείριση”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των καταχωρημένων χρηστών στο TANDEM (Εικόνα 50). Σε κάθε γραμμή του πίνακα παρουσιάζεται ένας χρήστης, και πιο συγκεκριμένα τα εξής χαρακτηριστικά του:

- **Μοναδικό αναγνωριστικό (ID)** του χρήστη
- **Μικρό Όνομα (First Name)** του χρήστη
- **Επίθετο (Last Name)** του χρήστη
- **Ηλεκτρονική Διεύθυνση (E-mail)** του χρήστη
- **Όνομα Χρήστη (Username)** του χρήστη
- **Όνομα Εταιρείας (Company Name)** του χρήστη
- **Τηλέφωνο (Phone Number)** του χρήστη
- **Τύπος Λογαριασμού (Account Type)** του χρήστη, π.χ. Προσωπικός (Personal), Επαγγελματικός (Professional)
- **Στοιχεία Πληρωμής (Payment Info)** του χρήστη, που περιλαμβάνουν τον αριθμό κάρτας, την ημερομηνία λήξης της, το όνομα που αναγράφεται στην κάρτα, τον κωδικό CVV της κάρτας και τον πρωτεύοντα αριθμό λογαριασμού
- **Φυσική Διεύθυνση (Physical Address)** του χρήστη, που προσδιορίζεται από τη χώρα, τη διεύθυνση, την πόλη, την περιφέρεια και τον ταχυδρομικό κωδικό της κατοικίας
- **Διεύθυνση Χρέωσης (Billing Address)** του χρήστη, που προσδιορίζεται από τη χώρα, τη διεύθυνση, την πόλη, την περιφέρεια και τον ταχυδρομικό κωδικό της χρέωσης
- **Ρόλος (Role)** του χρήστη, π.χ. Απλός Χρήστης (Standard User), Διαχειριστής (Administrator) κλπ.
- **Ενδιαφέροντα (Interests)** του χρήστη
- **Προσφερόμενες Υπηρεσίες (Offered Services)** από τον χρήστη στο σύστημα, για χρήση και από άλλα μέλη της πλατφόρμας
- **Αποκτηθείσες Υπηρεσίες (Obtained Services)** του χρήστη
- **Στιγμιότυποποιημένες Υπηρεσίες (Instantiated Services)** του χρήστη
- **Λεπτομέρειες Χρέωσης (Billing Info)** του χρήστη, που περιγράφονται από αντίστοιχο σύνδεσμο
- **Ενέργειες (Actions)** διαγραφής ενός χρήστη.

The screenshot shows the TANDEM Platform Portal interface. At the top, there is a navigation bar with links: Home, Applications, Services, SCO, Products, Users, Devices, Infrastructure, Pricing & Billing, and Configuration. Below the navigation bar is a search bar labeled "Search for users last names..". To the right of the search bar is a "Users Filters" sidebar. The main area displays a table of users with the following columns: ID#, First Name, Last Name, Email, Username, Company Name, Phone Number, Account Type, Payment Info, Physical Address, and Bill Address. Two users are listed: ICOM and EdgeServices. The ICOM user is Dimitris Christou, with email info@intracom-telecom.com, username ICOMProfessional, company INTRACOM S.A. Telecom Solutions, phone +30-210-6671449, account type Professional, payment info **** 4344 08/27 INTRACOM ****, physical address Greece 19,7 Km Markopoulou Ave, Peania, Athens Attika 19002, and bill address Greece 19,7 Km Markopoulou Ave, Peania, Athens Attika 19002. The EdgeServices user is John Marets, with email info@edge-services.com, username EdgeServicesProfessional, company IEdge Service S.A., phone +30-211-6471452, account type Professional, payment info **** 4344 06/26 EdgeServices ****, physical address Greece Grevenw 32, Patisia, Athens Attika 14322, and bill address Greece Grevenw 32, Patisia, Athens Attika 14322. The "Users Filters" sidebar on the right contains input fields for First Name, Last Name, Username, and Account Type, and dropdown menus for Company Name and Role. A "Submit" button is at the bottom of the sidebar.

Εικόνα 50: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων χρηστών στην πλατφόρμα

Η λίστα των χρηστών εμφανίζεται όταν καθοριστούν στην περιοχή φίλτρων στο δεξί μέρος της σελίδας τα επιθυμητά φίλτρα ανάλογα με τους χρήστες που θέλουμε να εμφανιστούν (ή κενά φίλτρα για εμφάνιση όλων) και πατηθεί το κουμπί “Submit”. Τα διαθέσιμα φίλτρα που μπορεί να προσδιοριστούν είναι τα εξής:

- Μικρό Όνομα
- Επίθετο
- Όνομα Χρήστη
- Τύπος Λογαριασμού
- Όνομα Εταιρείας
- Ρόλος

6.5.1 Καταχώρηση Χρήστη

Υπάρχει δυνατότητα καταχώρησης ενός νέου χρήστη στην πλατφόρμα TANDEM επιλέγοντας την ενέργεια “Users” → “Registration” (“Χρήστες” → “Εγγραφή”) από το κεντρικό μενού. Η εφαρμογή τότε εμφανίζει τη σελίδα που φιλοξενεί τη φόρμα εγγραφής χρηστών στο TANDEM (Εικόνα 51: Φόρμα εγγραφής χρήστη στην πλατφόρμα).

TANDEM Platform Portal

Home Applications Services SCO Products Users Infrastructure

User Registration

ID * User ID... Username * Username... First Name * First Name... Last Name * Last Name... Password * Password... Repeat Password Repeat Password... E-mail... E-mail... Company Name * Company Name... ICOM... Account Type Select User Role Select

Physical Address

Country	Greece
Address	19-71m Marousiou Ave.
City	Pearlia, Athens
State	Attika
Postal Code	15002

+ Add Billing Address

Interests

Name	Action
Internet of Things (IoT)	☐
Augmented Reality	☐

+ Add Interest

Submit

Εικόνα 51: Φόρμα εγγραφής χρήστη στην πλατφόρμα

Τα στοιχεία προς συμπλήρωση στη φόρμα εγγραφής νέου χρήστη είναι τα ακόλουθα:

- **Μοναδικό αναγνωριστικό (ID)** του χρήστη
- **Όνομα χρήστη (Username)** του χρήστη
- **Μικρό Όνομα (First Name)** του χρήστη
- **Επίθετο (Last Name)** του χρήστη
- **Κωδικός πρόσβασης (Password)** του χρήστη
- **Επιβεβαίωση Κωδικού πρόσβασης (Repeat Password)**
- **Ηλεκτρονική διεύθυνση (E-mail)** του χρήστη
- **Όνομα Εταιρείας (Company Name)** του χρήστη
- **Τύπος Λογαριασμού (Account Type)** του χρήστη π.χ. Προσωπικός (Personal), Επαγγελματικός (Professional)
- **Ρόλος (Role)** του χρήστη, π.χ. Απλός Χρήστης (Standard User), Διαχειριστής (Administrator) κλπ.
- **Φυσική Διεύθυνση (Physical Address)** του χρήστη, τα στοιχεία της οποίας καταχωρούνται σε δικό τους πίνακα και είναι τα εξής:
 - Χώρα (Country)
 - Διεύθυνση (Address)
 - Πόλη (City)
 - Περιφέρεια (State)
 - Ταχυδρομικός κωδικός (Postal Code)
- **Διεύθυνση Χρέωσης (Billing Address)** του χρήστη, τα στοιχεία της οποίας καταχωρούνται σε δικό τους πίνακα που εμφανίζεται μόνο με πάτημα του κουμπιού “Add Billing Address” (“Προσθήκη Διεύθυνσης Χρέωσης”) και αποτελούνται από τα ίδια πεδία με τη Φυσική Διεύθυνση (Εικόνα 52: Ο προαιρετικά εμφανιζόμενος πίνακας διεύθυνσης χρέωσης του χρήστη)
- **Ενδιαφέροντα (Interests)** του χρήστη, τα οποία επιλέγονται από μία προκαθορισμένη λίστα και έχουν τη δυνατότητα διαγραφής.

Billing Address	
Country	Greece
Address	19.7km Markopoulou Ave.
City	Peania, Athens
State	Attika
Postal Code	19002

Εικόνα 52: Ο προαιρετικά εμφανιζόμενος πίνακας διεύθυνσης χρέωσης του χρήστη

6.5.2 Τροποποίηση Χρήστη

Υπάρχει δυνατότητα τροποποίησης χρηστών πατώντας το μαύρο εικονίδιο “Edit” στη στήλη “Actions” του χρήστη προς τροποποίηση, όπως φαίνεται στην Εικόνα 53.



Εικόνα 53: Εικονίδιο “Edit” στις διαθέσιμες ενέργειες για έναν χρήστη

6.5.3 Είσοδος Χρήστη

Η είσοδος ενός χρήστη στην πλατφόρμα TANDEM γίνεται μέσω της αρχικής σελίδας της εφαρμογής, και συγκεκριμένα πατώντας το κουμπί “Login” που εμφανίζεται στο δεξί άκρο του κεντρικού μενού για να συνδεθεί στο λογαριασμό του. Με το πάτημα του κουμπιού αυτού, εμφανίζεται στο χρήστη παράθυρο για εισαγωγή και έλεγχο των διαπιστευτηρίων του (Εικόνα 18: Το παράθυρο σύνδεσης του χρήστη στην πλατφόρμα TANDEM).

Εάν η είσοδος του χρήστη πραγματοποιηθεί επιτυχώς, τότε στις διάφορες οθόνες της εφαρμογής το κουμπί εισόδου αντικαθίσταται από κατάλληλο μήνυμα χαιρετισμού του χρήστη, όπως φαίνεται στην Εικόνα 54: Μήνυμα χαιρετισμού του χρήστη μετά από είσοδό του στην πλατφόρμα .

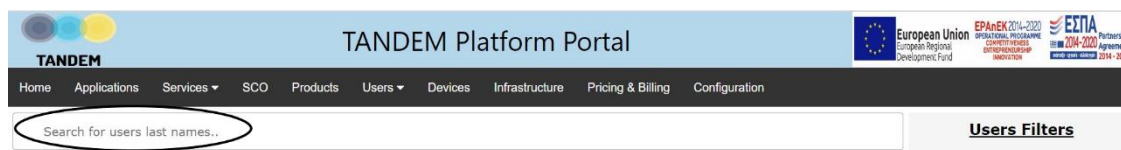


Εικόνα 54: Μήνυμα χαιρετισμού του χρήστη μετά από είσοδό του στην πλατφόρμα

6.5.4 Αναζήτηση Χρηστών

Υπάρχει δυνατότητα αναζήτησης ενός χρήστη με το επίθετό του μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα χρηστών, όπως φαίνεται στην Εικόνα 55.

Εναλλακτικά μπορεί να προσδιοριστεί το επίθετο του χρήστη στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.



Εικόνα 55: Γραμμή αναζήτησης χρηστών

6.5.5 Διαγραφή Χρήστη

Υπάρχει δυνατότητα διαγραφής χρηστών πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” του χρήστη προς διαγραφή, όπως φαίνεται στην Εικόνα 56.

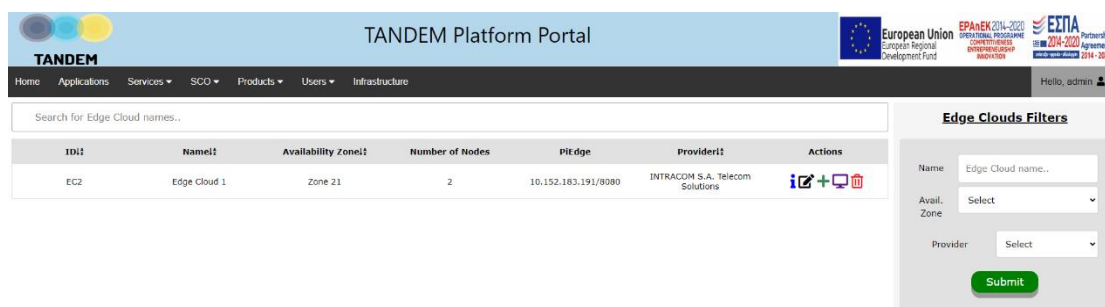


Εικόνα 56: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για έναν χρήστη

6.6 Διαχείριση Υποδομής

Όταν επιλεγεί η ενέργεια “Infrastructure” (“Υποδομή”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των καταχωρημένων υπολογιστικών νεφών άκρου (edge clouds) στο TANDEM (Εικόνα 57). Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων νεφών άκρου στην πλατφόρμα. Σε κάθε γραμμή του πίνακα παρουσιάζεται ένα νέφος άκρου, και πιο συγκεκριμένα τα εξής χαρακτηριστικά:

- **Μοναδικό αναγνωριστικό (ID)** του νέφους
- **Όνομα (Name)** του νέφους
- **Ζώνη Διαθεσιμότητας (Availability Zone)** του νέφους
- **Αριθμός Κόμβων (Number of Nodes)** του νέφους
- Τερματικό σημείο επικοινωνίας (endpoint) με το **PIEdge**
- **Πάροχος (Provider)** του νέφους π.χ. “INTRACOM S.A. Telecom Solutions”
- Διαθέσιμες **Ενέργειες (Actions)** που μπορούν να εφαρμοστούν στο νέφος.



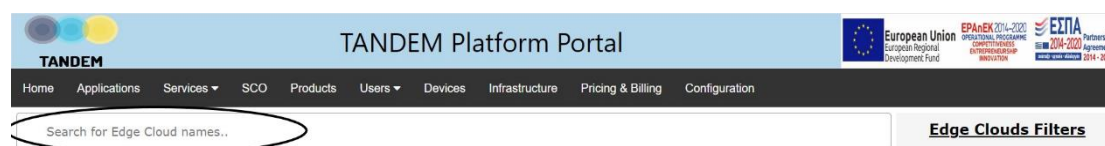
Εικόνα 57: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων νεφών άκρου στην πλατφόρμα

Η λίστα των υπολογιστικών νεφών εμφανίζεται όταν καθοριστούν στην περιοχή φίλτρων στο δεξί μέρος της σελίδας τα επιθυμητά φίλτρα ανάλογα με τα νέφη που θέλουμε να εμφανιστούν (ή κενά φίλτρα για εμφάνιση όλων) και πατηθεί το κουμπί “Submit”. Τα διαθέσιμα φίλτρα που μπορεί να προσδιοριστούν είναι τα εξής:

- Όνομα
- Ζώνη Διαθεσιμότητας
- Πάροχος

6.6.1 Αναζήτηση Edge Clouds

Ο χρήστης μπορεί να αναζητήσει υπολογιστικά νέφη άκρου με το όνομά τους μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα υπολογιστικών νεφών, όπως φαίνεται στην Εικόνα 58. Εναλλακτικά μπορεί να προσδιορίσει το όνομα του υπολογιστικού νέφους στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.



Εικόνα 58: Γραμμή αναζήτησης νεφών άκρου

6.6.2 Διαγραφή Edge Cloud

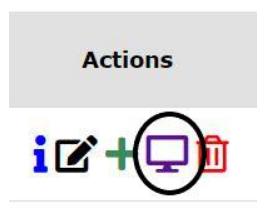
Υπάρχει δυνατότητα διαγραφής υπολογιστικών νεφών πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” του νέφους προς διαγραφή, όπως φαίνεται στην Εικόνα 59.



Εικόνα 59: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για ένα νέφος άκρου

6.6.3 Παρακολούθηση χρήσης πόρων Edge Cloud

Υπάρχει δυνατότητα παρακολούθησης της χρήσης πόρων των υπολογιστικών νεφών άκρου πατώντας το μοβ εικονίδιο “Monitor” στη στήλη “Actions” του νέφους προς παρακολούθηση (Εικόνα 60: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για ένα νέφος άκρου). Με το πάτημα του εικονιδίου ανοίγει σε νέο παράθυρο ένα περιβάλλον Grafana το οποίο παρουσιάζει τις μετρικές για όλους τους κόμβους του συσκευασμένου νέφους, όπως φαίνεται στην Εικόνα 61: Παρουσίαση μετρικών χρήσης πόρων από ένα νέφος άκρου.



Εικόνα 60: Εικονίδιο “Monitor” στις διαθέσιμες ενέργειες για ένα νέφος άκρου



Εικόνα 61: Παρουσίαση μετρικών χρήσης πόρων από ένα νέφος άκρου

6.6.4 Προβολή Nodes ενός Edge Cloud

Όταν ο χρήστης πατήσει το μπλε εικονίδιο “Info” στη στήλη “Actions” ενός νέφους (Εικόνα 62: Εικονίδιο “Info” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου), η εφαρμογή εμφανίζει το παράθυρο προβολής της συγκεντρωτικής λίστας των καταχωρημένων κόμβων (nodes) του συγκεκριμένου νέφους στο TANDEM, όπως φαίνεται στην Εικόνα 63. Σε κάθε γραμμή του πίνακα παρουσιάζεται ένας κόμβος, και πιο συγκεκριμένα τα εξής χαρακτηριστικά:

- **Μοναδικό αναγνωριστικό (ID)** του κόμβου
- **Όνομα (Name)** του κόμβου
- **Τοποθεσία (Location)** του κόμβου
- **Πάροχος (Provider)** του κόμβου, π.χ. “INTRACOM S.A. Telecom Solutions”
- **Διευθύνσεις (Addresses)** του κόμβου, οι οποίες περιλαμβάνουν το όνομα υπολογιστή του κόμβου, την εξωτερική διεύθυνση IP και την εσωτερική διεύθυνση IP
- **Συνθήκες (Conditions)** του κόμβου, οι οποίες περιλαμβάνουν την ετοιμότητα του κόμβου, την ύπαρξη πίεσης στο δίσκο του, την ύπαρξη πίεσης στη μνήμη του, την ύπαρξη πίεσης από μεγάλο αριθμό διεργασιών στον κόμβο και τη (μη) διαθεσιμότητά του
- **Χωρητικότητα (Capacity)** του κόμβου, και πιο συγκεκριμένα η χωρητικότητα της CPU, η χωρητικότητα της μνήμης, η χωρητικότητα του αποθηκευτικού χώρου καθώς και ο μέγιστος αριθμός pods στον κόμβο
- **Πόροι προς Διάθεση (Allocatable Resources)** του κόμβου, και πιο συγκεκριμένα η CPU, η μνήμη και ο αποθηκευτικός χώρος προς διάθεση

- **Γενικές Πληροφορίες (General Info)** του κόμβου, συμπεριλαμβανομένων του λειτουργικού συστήματος, της έκδοσης Kubernetes στο σύστημα, της έκδοσης πυρήνα του συστήματος, της αρχιτεκτονικής του συστήματος και την έκδοση του τρέχοντος container
- **Χρήση Πόρων (Usage)** του κόμβου, και πιο συγκεκριμένα το ποσοστό και τον ακριβή αριθμό της CPU και της μνήμης του κόμβου που χρησιμοποιείται την συγκεκριμένη στιγμή
- Διαθέσιμες **Ενέργειες (Actions)** που μπορούν να εφαρμοστούν στον του κόμβο.



Εικόνα 62: Εικονίδιο “Info” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου

TANDEM Platform Portal									
EDGE CLOUD NODES									
ID	Name	Location	Provider	Addresses	Conditions	Capacity	Allocatable Resources	General Info	Usage
k8smaster	k8smaster	Pearlia_Athens_19002	INTRACOM S.A. Telecom Solutions	Hostname: k8smaster CPU_IP: 146.124.106.209 Int_IP: 10.1.16.128	Ready: true Disk_Press: false CPU_Press: false Mem_Press: false NVMe_Usage: false	CPU: 12 Memory: 32862000 Ki Storage: 76368672 Ki Max_Pods_Number: 110	CPU Alloc.: 12 Mem Alloc.: 32779600 Ki Stor. Alloc.: 7520096 Ki	Node OS: Ubuntu 20.04.2 LTS Kubernetes Version: v1.24.8-2+1dda18a15eea38 Kernel Version: 5.4.0-113-generic Architecture: amd64 Container Runtime Version: containerd://1.5.13	CPU: 1377 m CPU %: 0.115 Memory: 11817388 Ki Memory %: 0.359
k8ssecondary	k8ssecondary	Pearlia_Athens_19002_2	INTRACOM S.A. Telecom Solutions	Hostname: k8ssecondary CPU_IP: 146.124.106.210 Int_IP: 10.1.162.128	Ready: true Disk_Press: false CPU_Press: false Mem_Press: false NVMe_Usage: false	CPU: 8 Memory: 16392640 Ki Storage: 153766352 Ki Max_Pods_Number: 110	CPU Alloc.: 8 Mem Alloc.: 16290940 Ki Stor. Alloc.: 152717776 Ki	Node OS: Ubuntu 20.04.2 LTS Kubernetes Version: v1.24.8-2+1dda18a15eea38 Kernel Version: 5.4.0-96-generic Architecture: amd64 Container Runtime Version: containerd://1.5.13	CPU: 742 m CPU %: 0.093 Memory: 4501676 Ki Memory %: 0.275

Εικόνα 63: Το αναδυόμενο παράθυρο παρουσίασης των κόμβων ενός νέφους άκρου

6.6.5 Καταχώρηση Νέου Node σε Edge Cloud

Υπάρχει δυνατότητα καταχώρησης νέου κόμβου σε ένα υπολογιστικό νέφος πατώντας το πράσινο εικονίδιο “Register Node” στη στήλη “Actions” του νέφους προς καταχώρηση κόμβου, όπως φαίνεται στην Εικόνα 64: Εικονίδιο “Register Node” στις διαθέσιμες ενέργειες για ένα νέφος άκρου.



Εικόνα 64: Εικονίδιο “Register Node” στις διαθέσιμες ενέργειες για ένα νέφος άκρου

6.6.6 Διαγραφή Node

Υπάρχει δυνατότητα διαγραφής κόμβων ενός υπολογιστικού νέφους πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” του κόμβου προς διαγραφή, όπως φαίνεται στην Εικόνα 65.



Εικόνα 65: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για έναν κόμβο του νέφους άκρου

6.6.7 Παρακολούθηση χρήσης πόρων Node

Υπάρχει δυνατότητα παρακολούθησης της χρήσης πόρων ενός κόμβου υπολογιστικού νέφους πατώντας το σκούρο μπλε εικονίδιο “Monitor Node” στη στήλη “Actions” του κόμβου προς παρακολούθηση (Εικόνα 66). Με το πάτημα του εικονιδίου ανοίγει σε νέο παράθυρο ένα περιβάλλον Grafana το οποίο παρουσιάζει την εικονικοποίηση μετρικών για τον συγκεκριμένο κόμβο, όπως φαίνεται στην Εικόνα 67.



Εικόνα 66: Εικονίδιο “Monitor Node” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου



Εικόνα 67: Παρουσίαση μετρικών χρήσης πόρων για έναν κόμβο ενός νέφους άκρου

6.6.8 Παρακολούθηση των services ενός Node

Υπάρχει δυνατότητα παρακολούθησης των υπηρεσιών ενός κόμβου υπολογιστικού νέφους πατώντας το μοβ εικονίδιο “Monitor Services” στη στήλη “Actions” του κόμβου προς παρακολούθηση (Εικόνα 68: Εικονίδιο “Monitor Services” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου). Με το πάτημα του εικονιδίου ανοίγει σε νέο παράθυρο ένα περιβάλλον Grafana το οποίο παρουσιάζει την εικονικοποίηση μετρικών για τις υπηρεσίες του συγκεκριμένου κόμβου, όπως φαίνεται στην Εικόνα 69.



Εικόνα 68: Εικονίδιο “Monitor Services” στις διαθέσιμες ενέργειες για έναν κόμβο ενός νέφους άκρου



Εικόνα 69: Παρουσίαση μετρικών χρήσης πόρων από τις υπηρεσίες ενός νέφους άκρου

6.7 Διαχείριση Συσκευών

Όταν επιλεγεί η ενέργεια **“Devices”** → **“Discovery”** (“Συσκευές” → “Ανακάλυψη”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των καταχωρημένων συσκευών στο TANDEM (Εικόνα 70: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων συσκευών στην πλατφόρμα). Σε κάθε γραμμή του πίνακα παρουσιάζεται μία συσκευή, και πιο συγκεκριμένα τα εξής χαρακτηριστικά της:

- **Ημερομηνία Δημιουργίας (Creation Date)** της συσκευής
- **Όνομα (Name)** της συσκευής
- **Περιγραφή (Description)** της συσκευής
- **Ετικέτες (Labels)** της συσκευής, δηλαδή χαρακτηριστικές λέξεις-κλειδιά που συνδέονται με τη λειτουργία της
- **Λειτουργική Κατάσταση (Operating State)** της συσκευής, η οποία μπορεί να είναι είτε Ενεργοποιημένη (Enabled) είτε Απενεργοποιημένη (Disabled)
- **Διαχειριστική Κατάσταση (Admin State)** της συσκευής, η οποία μπορεί να είναι είτε Κλειδωμένη (Locked) είτε Ξεκλειδωμένη (Unlocked)
- **Διεύθυνση IP και Θύρα (IP:Port)** της συσκευής όπου είναι προσβάσιμη από το χρήστη
- Διαθέσιμες **Ενέργειες (Actions)** που μπορούν να εφαρμοστούν στη συσκευή.

The screenshot shows the TANDEM Platform Portal interface. At the top, there is a navigation bar with links: Home, Applications, Services, SCO, Products, Users, Devices, Infrastructure, Pricing & Billing, and Configuration. Below the navigation bar is a search bar labeled "Search for Device names..". To the right of the search bar is a "Device Filters" panel with input fields for Name, Labels, Operating State, and Admin State, and a "Submit" button. The main content area displays a table of devices with columns: Creation Date, Name, Description, Labels, Operating State, Admin State, IP:Port, and Actions. Two devices are listed:

Creation Date	Name	Description	Labels	Operating State	Admin State	IP:Port	Actions
Tue Jan 20 1970 06:40:07 GMT+0200 (Eastern European Standard Time)	360°CameraDevice	A 360° camera used for surveillance and safety purposes in a variety of vertical industries.	camera, surveillance, 360°	ENABLED	UNLOCKED	146.124.106.208:8080	[Edit] [Delete]
Tue Jul 12 2022 13:36:20 GMT+0300 (Eastern European Summer Time)	Temp_and_Humidity_sensor_cluster_01	Raspberry Pi sensor cluster	Humidity sensor, Temperature sensor, DHT11	ENABLED	UNLOCKED	localhost:80	[Edit] [Delete]

Εικόνα 70: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων συσκευών στην πλατφόρμα

Η λίστα των συσκευών εμφανίζεται όταν καθοριστούν στην περιοχή φίλτρων στο δεξί μέρος της σελίδας τα επιθυμητά φίλτρα ανάλογα με τις συσκευές που θέλουμε να εμφανιστούν (ή κενά φίλτρα για εμφάνιση όλων) και πατηθεί το κουμπί "Submit". Τα διαθέσιμα φίλτρα που μπορεί να προσδιοριστούν είναι τα εξής:

- Όνομα
- Ετικέτες
- Λειτουργική Κατάσταση
- Διαχειριστική Κατάσταση

6.7.1 Αναζήτηση Συσκευών

Ο χρήστης μπορεί να αναζητήσει συσκευές με το όνομά τους μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα συσκευών, όπως φαίνεται στην Εικόνα 71. Εναλλακτικά μπορεί να προσδιορίσει το όνομα της συσκευής στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.

This screenshot is similar to the previous one, but the search bar labeled "Search for Device names.." is circled in red to highlight it.

Εικόνα 71: Γραμμή αναζήτησης συσκευών

6.7.2 Διαγραφή Συσκευής

Υπάρχει δυνατότητα διαγραφής συσκευών πατώντας το κόκκινο εικονίδιο "Delete" στη στήλη "Actions" της συσκευής προς διαγραφή, όπως φαίνεται στην Εικόνα 72: Εικονίδιο "Delete" στις διαθέσιμες ενέργειες για μία συσκευή.



Εικόνα 72: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για μία συσκευή

6.8 Διαχείριση Προϊόντων (Κατάλογος Προϊόντων)

Όταν ο χρήστης επιλέξει **“Products”** → **“Management”** (“Προϊόντα” → “Διαχείριση”) από το κεντρικό μενού, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη συγκεντρωτική λίστα των προϊόντων (Εικόνα 73). Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων υπηρεσιών στην πλατφόρμα). Σε κάθε γραμμή του πίνακα παρουσιάζεται ένα προϊόν, και πιο συγκεκριμένα τα εξής χαρακτηριστικά του:

- **Μοναδικό αναγνωριστικό (ID)** του προϊόντος
- **Όνομα (Name)** του προϊόντος
- **Τύπος (Type)** του προϊόντος, π.χ. FaaS Services, Device Services κλπ.
- **Πάροχος (Provider)** του προϊόντος, π.χ. “Intracom Telecom”
- **Περιγραφή (Description)**, για την οποία εμφανίζονται οι 100 πρώτοι χαρακτήρες, ενώ η πλήρης περιγραφή εμφανίζεται όταν ο χρήστης περάσει πάνω από τις τρεις κουκίδες
- **Κατηγορία (Category)** στην οποία ανήκει το προϊόν, π.χ. Location-based services, IoT, Video Analytics κλπ.
- **Έκδοση (Version)** του προϊόντος
- **Κατάσταση (State)** στην οποία βρίσκεται το προϊόν, π.χ. “In Design”, “Approved (Active)”, “Instantiated”, “Rejected” κλπ.
- Διαθέσιμες **Ενέργειες (Actions)**, π.χ. Αλλαγή (Edit), Διαγραφή (Delete) κλπ.

TANDEM Platform GUI								
Home Applications Services SCO Products Users Devices Infrastructure Pricing & Billing Configuration Search for Product names... Products Filters Name Product name... Type Category Provider Status Submit								
ID#	Name#	Type#	Provider	Description	Category#	Version#	State	Actions
ObjectDetectionProduct	Object Detection Product	FaaS Services	Intracom Telecom	Object Detection Product	Video Analytics	0.1	In Study	 
pro4	Tandem product 4	FaaS Services	Intracom Telecom	For testing purposes	Location Based Services	0.1	In Study	 
pro2	Tandem product 2	FaaS Services	Intracom Telecom	Description...	Location Based Services	0.1	In Design	 
pro1	Tandem product 1	FaaS Services	Intracom Telecom	Description...	Internet of Things (IoT)		In Study	 
pro3	Tandem product 3	FaaS Services	Intracom Telecom	Description...	Location Based Services	0.1	In Study	 
pro5	Temperature Monitoring	FaaS Services	Intracom Telecom	Monitoring the temperature in a room	Internet of Things (IoT)	0.1	In Study	 

Εικόνα 73: Η σελίδα παρουσίασης της συγκεντρωτικής λίστας καταχωρημένων προϊόντων στην πλατφόρμα

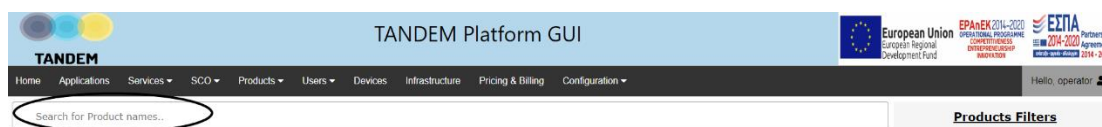
Η λίστα των προϊόντων εμφανίζεται όταν φορτώνεται η σελίδα. Επίσης, ο χρήστης μπορεί να καθορίσει, στην περιοχή φίλτρων στο δεξί μέρος της σελίδας, τα επιθυμητά φίλτρα ανάλογα με τα προϊόντα που θέλει να δει (ή κενά φίλτρα για εμφάνιση όλων). Τα φίλτρα εφαρμόζονται

όταν πατήσει το κουμπί “Submit”. Τα διαθέσιμα φίλτρα που μπορεί να προσδιορίσει ο χρήστης είναι τα εξής:

- Όνομα
- Τύπος
- Κατηγορία
- Πάροχος
- Κατάσταση

6.8.1 Αναζήτηση Προϊόντος

Ο χρήστης μπορεί να αναζητήσει προϊόντα με το όνομά τους μέσω της γραμμής αναζήτησης ακριβώς πάνω από τον πίνακα προϊόντων, όπως φαίνεται στην Εικόνα 74. Εναλλακτικά μπορεί να προσδιορίσει το όνομα του προϊόντος στην περιοχή φίλτρων, αφήνοντας κενά όλα τα υπόλοιπα φίλτρα.



Εικόνα 74: Γραμμή αναζήτησης προϊόντων

6.8.2 Διαγραφή Προϊόντος

Ο χρήστης μπορεί να διαγράψει προϊόντα πατώντας το κόκκινο εικονίδιο “Delete” στη στήλη “Actions” του προϊόντος προς διαγραφή, όπως φαίνεται στην Εικόνα 75.



Εικόνα 75: Εικονίδιο “Delete” στις διαθέσιμες ενέργειες για ένα προϊόν

6.8.3 Δημιουργία Προϊόντος από μια Υπηρεσία

Ο χρήστης μπορεί να επιλέξει μια υπηρεσία από τον πίνακα υπηρεσιών και, με βάση αυτήν, να δημιουργήσει το αντίστοιχο προϊόν, όπως περιγράφεται αναλυτικά στο Κεφάλαιο 6.3.5.

6.8.4 Καταχώρηση Προϊόντος

Ο χρήστης έχει τη δυνατότητα να καταχωρήσει ένα νέο προϊόν, επιλέγοντας την ενέργεια “Products” → “Registration ” (“Προϊόντα” → “Εγγραφή”) από το κεντρικό μενού. Η εφαρμογή τότε εμφανίζει τη σελίδα που φιλοξενεί τη φόρμα εγγραφής προϊόντων στο TANDEM (Εικόνα 76).

TANDEM Platform Portal

Home Applications Services SCO Products Users Devices Infrastructure Pricing & Billing Configuration Hello, operator

Product Registration / Editing

Product Name * Product name... Product Id * Product id...

Product Category * Select Product Type * Select

Product Provider * Select Product Version * Select

Product Service * Select Product Application * Select

Life Cycle Status * Select

Product Description

Is Buddle ☐ Is Sellable ☐ Product Pricing Model * Select

Product Price Per Charge Unit * Select Currency * Select

Valid for * mm/dd/yyyy mm/dd/yyyy

Product Service Level Agreement (SLA) Select [Add SLA](#)

TANDEM Locations	
Location	Select
Pania area1	☐
Pania area2	☐
Stavros Attikis 1	☐
Melissia area 1	☐
Melissia area 2	☐
Melissia area 3	☐
Chalandri area 1	☐

[Submit](#)

Εικόνα 76: Φόρμα εγγραφής προϊόντος στην πλατφόρμα

Τα στοιχεία προς συμπλήρωση στη φόρμα εγγραφής νέου προϊόντος είναι τα ακόλουθα:

- **Όνομα (Product Name)** του προϊόντος
- **Μοναδικό αναγνωριστικό (Product Id)** του προϊόντος
- **Κατηγορία (Product Category)** στην οποία ανήκει το προϊόν
- **Τύπος (Product Type)** του προϊόντος
- **Πάροχος (Product Provider)** του προϊόντος
- **Έκδοση (Product Version)** του προϊόντος
- **Υπηρεσία (Product Service)** ή **Εφαρμογή (Product Application)** για την οποία δημιουργείται το προϊόν
- **Περιγραφή (Product Description)** του προϊόντος
- **Κατάσταση (Life Cycle Status)** στην οποία βρίσκεται το προϊόν
- **Ομαδοποιημένο (Is Buddle)**, το οποίο επιλέγεται αν το προϊόν αποτελεί κομμάτι ενός σύνθετου προϊόντος
- **Εμπορεύσιμο (Is Sellable)**, το οποίο επιλέγεται αν το προϊόν μπορεί να πουληθεί μόνο του
- **Μοντέλο τιμολόγησης (Product Pricing Model)** του προϊόντος
- **Τιμή ανά μονάδα χρέωσης (Product Price Per Charge Unit)** του προϊόντος, η οποία προσδιορίζεται από την Μονάδα Χρέωσης και την Τιμή της
- **Νόμισμα (Currency)** της τιμής του προϊόντος
- **Περίοδος εγκυρότητας (Valid for)** του προϊόντος, η οποία ορίζεται από τις ημερομηνίες έναρξης και λήξης του προϊόντος
- **Συμφωνία σε επίπεδο υπηρεσιών (Product Service Level Agreement)** του προϊόντος

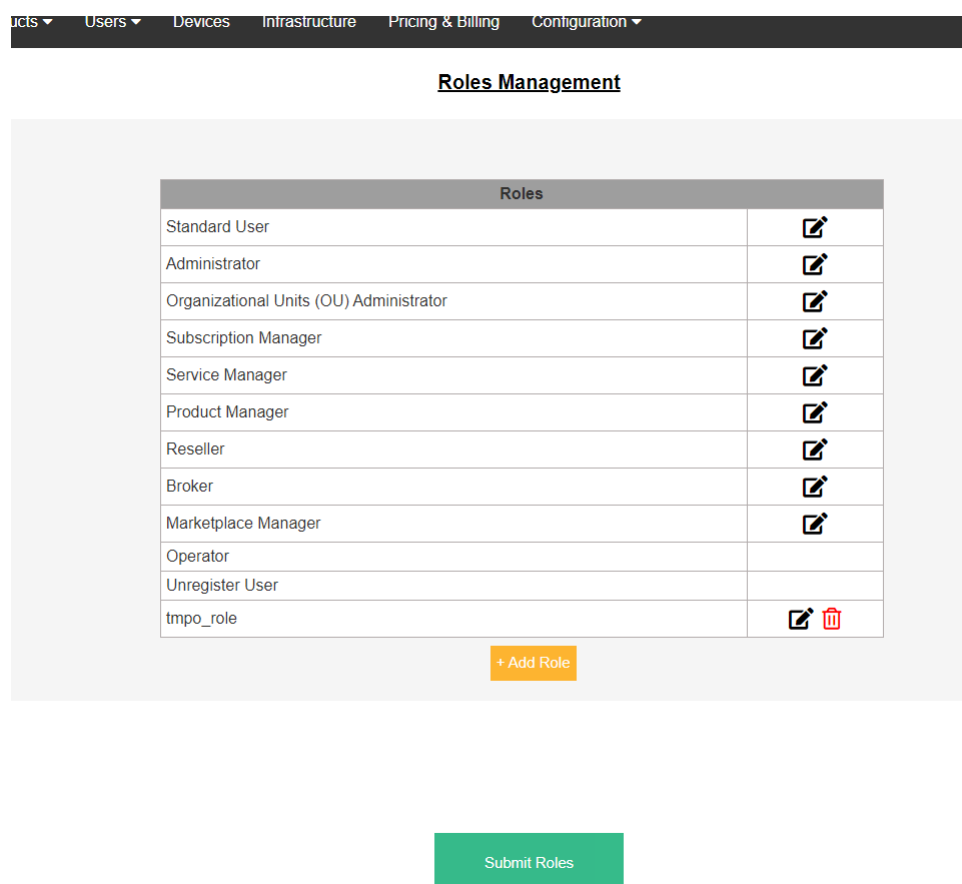
- **Τοποθεσίες (TANDEM locations)** του προϊόντος, οι οποίες αποτελούν τις θέσεις των κόμβων του TANDEM και καταχωρούνται σε δικό τους πίνακα, από όπου ο χρήστης μπορεί να επιλέξει σε ποιες τοποθεσίες θα μπορεί να υπάρχει το προϊόν

6.8.5 Τροποποίηση Προϊόντος

Ο χρήστης έχει τη δυνατότητα να τροποποιήσει ένα υπάρχον προϊόν επιλέγοντας την ενέργεια **“Products”** → **“Registration”** (“Προϊόντα” → “Εγγραφή”) από το κεντρικό μενού. Έπειτα, η εφαρμογή εμφανίζει τη σελίδα που φιλοξενεί τη φόρμα τροποποίησης προϊόντων στο TANDEM, η οποία είναι ίδια με τη φόρμα εγγραφής προϊόντος και εμφανίζεται στην Εικόνα 76. Σε αυτήν την περίπτωση, ο χρήστης θα πρέπει να γωρίζει το μοναδικό αναγνωριστικό (Id) του προϊόντος και να το εισάγει στο κατάλληλο πεδίο. Επίσης, θα πρέπει να συμπληρώσει σωστά όλα τα πεδία της φόρμας, γιατί όταν πατήσει το κουμπί **“Submit”**, η εφαρμογή θα ενημερώσει όλα τα χαρακτηριστικά του προϊόντος.

6.9 Διαχείριση ρόλων χρηστών

Το σύστημα δίνει τη δυνατότητα διαχείρισης ρόλων χρηστών και σύνδεση αυτών με συγκεκριμένα δικαιώματα πρόσβασης σε λειτουργίες του συστήματος. Ο χρήστης με ρόλο ‘Operator’ είναι προκαθορισμένος, έχει τη δυνατότητα πρόσβασης σε όλες τις λειτουργίες του συστήματος. Είναι ο ρόλος που έχει την δυνατότητα να καταχωρήσει ένα καινούργιο ρόλο στο σύστημα ή να τροποποιήσει τις ιδιότητες ενός ήδη καταχωρημένου ρόλου επιλέγοντας την ενέργεια **“Configuration”** → **“Roles Management”** (“Διαμόρφωση” → “Διαχείριση Ρόλων”)



Εικόνα 77 – Οθόνη Διαχείρισης των ρόλων χρηστών του συστήματος

Στην Εικόνα 77 εμφανίζεται μια λίστα με τους ρόλους χρηστών του συστήματος.

Πατώντας το κουμπί '+Add Role' ο χρήστης έχει την δυνατότητα να προσθέσει ένα νέο ρόλο.

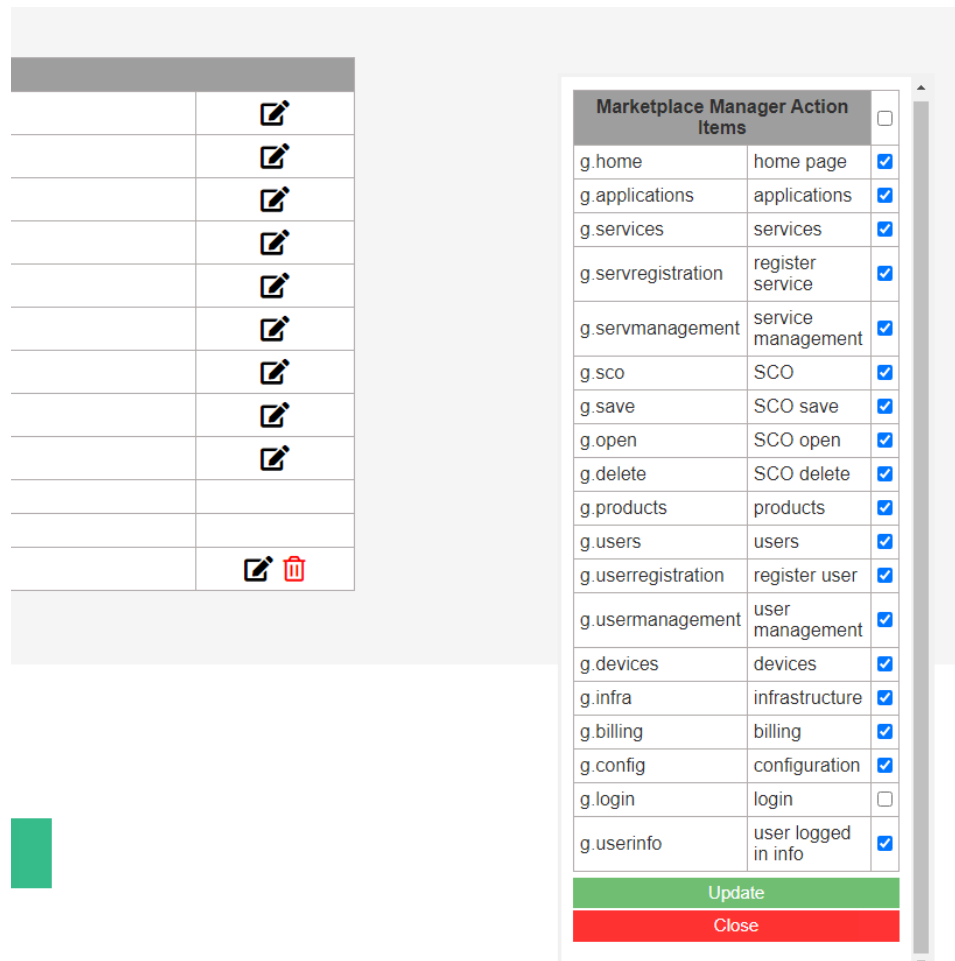
Πατώντας το κουμπί διαγραφής όπου το σύστημα του επιτρέπει, ο χρήστης μπορεί να διαγράψει τον ρόλο από το σύστημα.

Ο χρήστης έχει την δυνατότητα να καταχωρήσει όλες τις αλλαγές που έχει κάνει στους ρόλους και στις ιδιότητές τους, πατώντας το πλήκτρο 'Submit Roles'

6.9.1 Δυνατότητες πρόσβασης ρόλου

Ο χρήστης μπορεί να πατήσει το κουμπί διόρθωσης όπου του επιτρέπεται από το σύστημα, ώστε να διαμορφώσει τις δυνατότητες πρόσβασης ενός ρόλου.

Για τον σκοπό αυτό, εμφανίζεται η πιο κάτω οθόνη:



Marketplace Manager Action Items		
g.home	home page	☒
g.applications	applications	☒
g.services	services	☒
g.servregistration	register service	☒
g.servmanagement	service management	☒
g.sco	SCO	☒
g.save	SCO save	☒
g.open	SCO open	☒
g.delete	SCO delete	☒
g.products	products	☒
g.users	users	☒
g.userregistration	register user	☒
g.usermanagement	user management	☒
g.devices	devices	☒
g.infra	infrastructure	☒
g.billing	billing	☒
g.config	configuration	☒
g.login	login	☐
g.userinfo	user logged in info	☒

Update

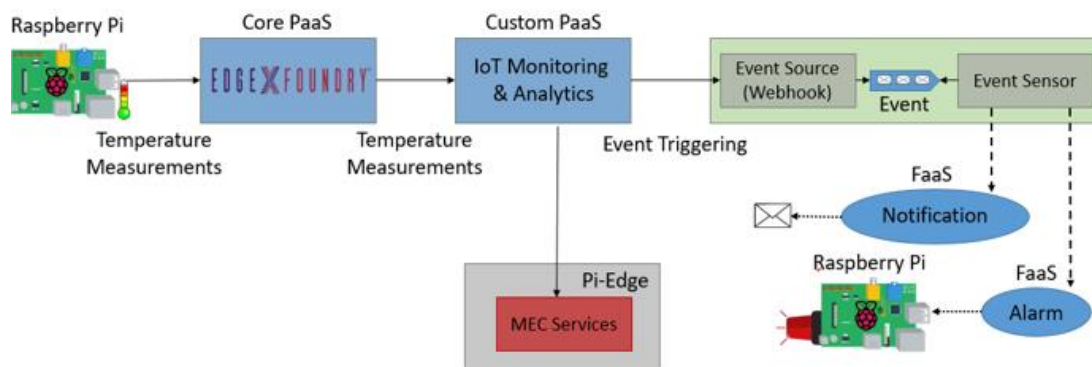
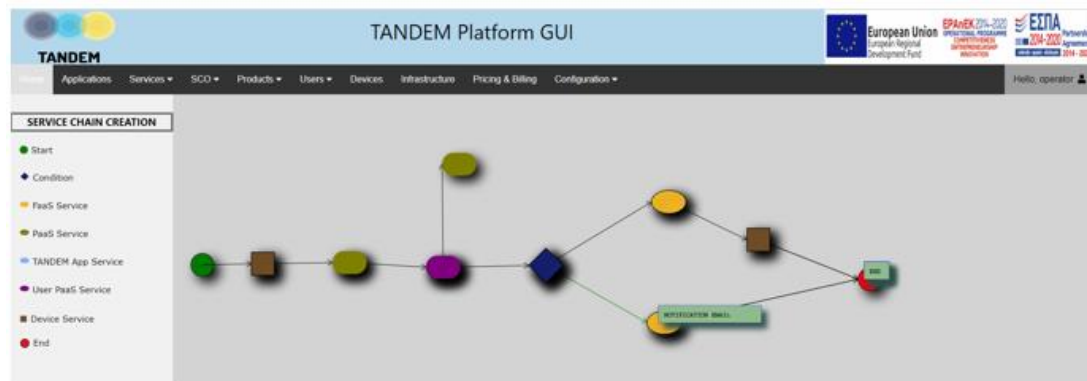
Close

Εικόνα 78 - Συσχέτιση Ρόλων Χρηστών με δικαιώματα πρόσβασης σε λειτουργίες

Στην Εικόνα 78 ο χρήστης έχει την δυνατότητα να δηλώσει πατώντας κατάλληλα το checkbox, ποιά λειτουργία θα μπορεί να είναι προσπελάσιμη από χρήστες που έχουν τον ρόλο που επιλέχθηκε. Με το πλήκτρο 'Update' ο χρήστης επιβεβαιώνει τις αλλαγές και επιστρέφει στην προηγούμενη οθόνη. Με το πλήκτρο 'Close' ο χρήστης επιστρέφει στην προηγούμενη σελίδα χωρίς να διασώσει τις όποιες αλλαγές έχει κάνει.

7 Πρωτότυπα Αλυσίδων Υπηρεσιών (Service Chains Prototypes)

7.1 Αλυσίδα Υπηρεσιών Διαδικτύου των Πραγμάτων (IoT Service Chain)



Εικόνα 79: Η ροή εργασιών του IoT (κάτω) και πως αναπαριστάται αναλυτικά στην σελίδα διαχείρισης αλυσίδων του TANDEM portal (πάνω).

```
tanmaster@k8smaster:~$ kubectl -n argo-events get pods
NAME                                READY   STATUS    RESTARTS   AGE
controller-manager-565678585b-zx7jt 1/1     Running   0           7d2h
events-webhook-cf669c854-x54vd        1/1     Running   0           7d2h
eventbus-default-stan-0               2/2     Running   0           7d2h
eventbus-default-stan-1               2/2     Running   0           7d2h
eventbus-default-stan-2               2/2     Running   0           7d2h
webhook-eventsource-44n55-5889c4989-pqbxz 1/1     Running   0           6d2h
webhook-sensor-znnff-7c4986b99-9tnpd 1/1     Running   0           6d2h
```

Εικόνα 80: Τα δομικά στοιχεία του Argo framework σε περιβάλλον Kubernetes

Στην Εικόνα 79: Η ροή εργασιών του IoT (κάτω) και πως αναπαριστάται αναλυτικά στην σελίδα διαχείρισης αλυσίδων του TANDEM portal (πάνω).

φαίνεται το IoT workflow όπως υλοποιήθηκε στο έργο. Το Device Service στα αριστερά λειτουργεί ως ενεργοποιητής, μετρώντας τη θερμοκρασία και την υγρασία ενός δωματίου και στέλνοντας τη μέτρηση στο Core Data, που αποτελεί δομικό στοιχείο του EdgeX Foundry.

Στη συνέχεια, το δομικό στοιχείο Rules Engine στέλνει την μέτρηση στον Message Broker (κομμάτι του IoT Monitoring & Analytics). Από εκεί το Temperature Monitoring Service, που καταναλώνει τα μηνύματα από τον Message Broker, εκτελεί δύο λειτουργίες: πρώτον αποθηκεύει τη μέτρηση σε μια βάση δεδομένων, και δεύτερον, αναλόγως το ύψος της μέτρησης, στέλνει ένα HTTP POST request σε ένα από τα δύο endpoints του Event Source χρησιμοποιώντας τα MEC Services. Συγκεκριμένα, για το πρώτο endpoint εκτελεί τη μέθοδο Notification Service που αποστέλλει ένα e-mail στο UE (κινητή συσκευή) που βρίσκεται στην κοντινότερη απόσταση από την IoT συσκευή, όπως αυτά παρέχονται από τα υποστηριζόμενα MEC Services. Το email αυτό περιέχει πληροφορίες για την τοποθεσία του UE, της IoT συσκευής καθώς και για το ύψος της θερμοκρασίας. Το δεύτερο endpoint εκτελεί τη μέθοδο Alarm Service ζητώντας από το Command Service του EdgeX να ενεργοποιήσει το συναγερμό για μια διάρκεια Χ δευτερολέπτων που είναι επίσης παραμετροποιήσιμη από το χρήστη. Το κομμάτι του παραπάνω workflow, από το Event Source και μετά, έχει υλοποιηθεί με τη χρήση του Argo Events framework [Argo]. Όπως ίσως γίνεται αντιληπτό από την ονομασία, οι μέθοδοι εκτελούνται κατά την πυροδότηση συγκεκριμένων γεγονότων (events). Κατ' αυτή την έννοια, υπάρχει ένας "παραγωγός" κι ένας "καταναλωτής" γεγονότων, καθώς κι ένας δίαυλος (bus) για την αποθήκευση/κατανάλωση των γεγονότων. Στην περίπτωση του IoT workflow, τα γεγονότα, αποτελούν τα POST requests που αναφέρθηκαν παραπάνω. Στην Εικόνα 80 φαίνονται τα δομικά στοιχεία του framework σε περιβάλλον Kubernetes που είναι απαραίτητα για τις λειτουργίες που μόλις περιγράφηκαν. Μπορεί να διακριθεί ο δίαυλος των γεγονότων με τρία αντίγραφα (event bus), ο παραγωγός (webhook-eventsource) και τέλος, ο καταναλωτής (webhook-sensor).

Παρακάτω περιγράφεται πιο αναλυτικά το σενάριο παρακολούθησης δύο μετρητικών μεγεθών (θερμοκρασίας και υγρασίας) σαν παράδειγμα των Faas και PaaS υπηρεσιών που υποστηρίζει το TANDEM.

Επισκόπηση

Το σενάριο περιγράφει την ανάπτυξη μιας σειράς υπηρεσιών PaaS και FaaS που αποτελούν ένα ολοκληρωμένο σενάριο χρήσης της TANTEM πλατφόρμας, και των αυτοματοποιήσεων που αυτή προσφέρει, για την δημιουργία μιας εφαρμογής που θα τρέχει αποκλειστικά στις παρυφές του δικτύου. Στόχος είναι:

1. Η διαρκής παρακολούθηση δύο μετρητικών μεγεθών (θερμοκρασίας, υγρασίας) στο server room των εγκαταστάσεων της Intracom.
2. Η δυνατότητα της διαλειτουργικότητας των μετρητικών οργάνων και actuators ανάμεσα σε διάφορους χρήστες της πλατφόρμας του TANDEM.
3. Η αποθήκευση των μετρήσεων των για μελλοντική χρήση σε αλγορίθμους forecasting
4. Η δυνατότητα live παρακολούθησης των μετρήσεων.
5. Η χρήση των MEC Services, και συγκεκριμένα του LOCATION API.
6. Η δυνατότητα εκτέλεσης workflows σε συνάρτηση με συγκεκριμένες τιμές (thresholds) των μετρήσεων.

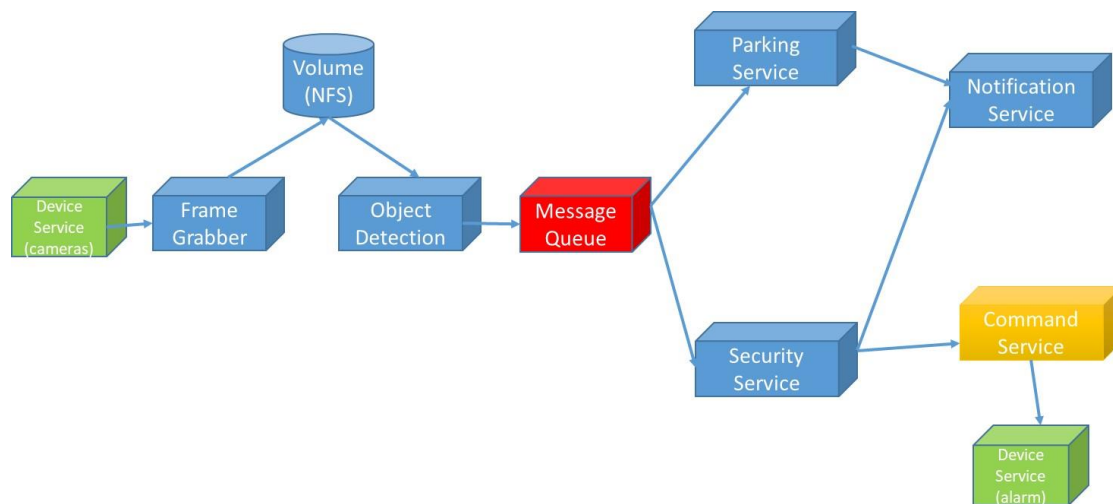
Το σενάριο χρησιμοποιεί ένα σύνολο από services που επιτρέπουν την ικανοποίηση των παραπάνω απαιτήσεων. Παρακάτω αναφέρονται τα κυριότερα services:

1. **Device services:** Το ρόλο των μετρητικών οργάνων και actuators, σε αυτό το σενάριο παίζει ένας υπολογιστής μοναδικής πλακέτας (single-board computer, SBC) Raspberry Pi 4. Με τη χρήση 3rd party αισθητήρων αλλά και actuators (ADAFRUIT-DHT11 sensor και LOUDITY-loudspeaker 0.5W 8Ω - 28 x 4.7mm) προσομοιώνει ένα σύνολο «έξυπνων» συσκευών που το TANDEM φιλοδοξεί να διασυνδέσει στα πλαίσια των edge monitoring services. Οι συσκευές αυτές είναι ένας cluster αισθητήρων (θερμοκρασίας-υγρασίας) καθώς και μία προειδοποιητική σειρήνα.
2. **EdgeXFoundry:** Το EdgeXFoundry επιτυγχάνει τη διαλειτουργικότητα των συνδεδεμένων device services, επιτρέποντας σε εξωτερικούς χρήστες να ενημερώνονται για τα information models που χρησιμοποιούν οι διάφορες συνδεδεμένες συσκευές ή actuators καθώς και να κάνουν access τις μετρήσεις των συνδεδεμένων συσκευών ή να χειρίζονται τους συνδεδεμένους actuators. Το EdgeXFoundry έχει βασιστεί στην αρχιτεκτονική των microservices. Τα microservices που χρησιμοποιούνται στο σενάριο είναι τα παρακάτω:
 - i. Edgex-Core-Metadata: Το service αυτό διατηρεί τα μεταδεδομένα των συνδεδεμένων συσκευών και ενεργοποιητών, επιτρέποντας σε 3^{ους} χρήστες να ενημερώνονται για τον τρόπο με τον οποίο μπορούν να έχουν πρόσβαση στις συσκευές αυτές
 - ii. Edgex-Core-Data: Το service αυτό διατηρεί τις μετρήσεις που στέλνονται από τις συσκευές στο EdgeXFoundry και δίνει πρόσβαση σε αυτές.
 - iii. Edgex-Core-Command: Το service αυτό διατηρεί τα μεταδεδομένα των εντολών που δέχονται οι δηλωμένοι actuators(σειρήνα).
 - iv. Edgex- Kuiper: Η rules engine του Edgex, χρησιμοποιείται για την προώθηση των χρήσιμων δεδομένων από το edge σε μια κεντρική τοποθεσία.
 - v. Edgex-app-service-configurable-mqtt: Το service αυτό χρησιμοποιείται από την rules engine για την προώθηση των δεδομένων μέσω ενός mqtt Broker (Mosquitto).
 - vi. Edgex-device-rest: Το service αυτό χρησιμοποιείται από το Edgex για την ενεργοποίηση των actuators(σειρήνα).
3. **MEC Services:** Πρόκειται για ένα σύνολο από υπηρεσίες, οι οποίες όπως αναλύθηκε στην υποενότητα 3.3, δίνουν τη δυνατότητα σε εξωτερικές εφαρμογές να παρέχουν και να χρησιμοποιήσουν τις υπηρεσίες άλλων εφαρμογών, όπως και να καταναλώνουν APIs που προσφέρουν πληροφορίες σχετικά με την τοποθεσία των εγγεγραμμένων χρηστών/συσκευών (UE). Τα services αυτά μπορούν να διακριθούν στις εξής δύο κατηγορίες:
 - i. Service Management APIs: Ένα σύνολο διεπαφών που δίνει τη δυνατότητα στις εφαρμογές να προωθήσουν τις υπηρεσίες που προσφέρουν, να μεταβάλλουν την κατάστασή τους ή να τις διαγράψουν, ενώ μπορούν επίσης να κάνουν αναζήτηση άλλων υπηρεσιών και έλεγχο διαθεσιμότητας μίας ή περισσότερων υπηρεσιών που παρέχονται από άλλες εφαρμογές.
 - ii. Location APIs: Ένα σύνολο από διεπαφές, οι οποίες παρέχονται από το MEC Platform και προσφέρουν πληροφορίες που αφορούν την τοποθεσία διαφόρων User Equipments, τον υπολογισμό αποστάσεων, καθώς και την εύρεση του/των κοντινότερων UE(s).
4. **IoT Data Analytics:** Υπηρεσία, όπως περιγράφηκε στο 3.3.1, η οποία έχει ως ευθύνη την οπτικοποίηση των μετρήσεων υγρασίας και θερμοκρασίας που συλλέγονται

από τους αισθητήρες της πλακέτας RaspberryPi που είναι συνδεδεμένη στην EdgeX PaaS υπηρεσία

- i. Message broker: Την ευθύνη για την μεταφορά των δεδομένων από το edge (δηλαδή από το edgex) σε μια πιο κεντρική τοποθεσία αναλαμβάνει ο MQTT message broker Eclipse Mosquitto. Ο Mosquitto λαμβάνει με την μορφή mqtt messages, όλες τις χρήσιμες μετρήσεις που φτάνουν στο edgex. Στο συγκεκριμένο σενάριο, MQTT message broker συλλέγει τις μετρήσεις υγρασίας & θερμοκρασίας της πλακέτας RaspberryPi.
 - ii. Temperature and Humidity monitoring service: Το Service αυτό είναι επιφορτισμένο με το συλλέγει τις μετρήσεις που αποθηκεύονται στον παραπάνω message broker και να αποθηκεύει τις μετρήσεις σε μια κεντρική βάση (influxDB). Επιπρόσθετα, το service εφαρμόζει έναν ή παραπάνω κανόνες στις μετρήσεις υγρασίας & θερμοκρασίας που δέχεται, ώστε να ενεργοποιεί μια σειρά από workflows ανάλογα με τις μετρήσεις που φτάνουν σε αυτό. Για παράδειγμα μπορεί να στείλει ένα ενημερωτικό mail (“notification action”) σε περίπτωση που η θερμοκρασία/υγρασία στο server room είναι μη ικανοποιητική, ή ενεργοποιήσει την σειρήνα (“alarm action”) σε περίπτωση που η θερμοκρασία/υγρασία φτάσει σε επικίνδυνα επίπεδα.
 - iii. IoT data DB: Στο σενάριο γίνεται χρήση της Influx DB για να αποθήκευση των IoT δεδομένων και συγκεκριμένα των μετρήσεων υγρασίας & θερμοκρασίας που συλλέγει η Temperature and Humidity monitoring service.
 - iv. IoT Data Monitoring : Στο σενάριο χρησιμοποιείται η γραφική πλατφόρμας για την ανάλυση δεδομένων, Grafana, όπου μέσω διαφόρων διαγραμμάτων οπτικοποιεί τις μετρήσεις θερμοκρασίας και υγρασίας που έχουν αποθηκευτεί στην IoT data DB.
5. Υποσύστημα Argo: Το υποσύστημα Argo υλοποιεί την ιδέα του FaaS όπως έχει αναφερθεί και παραπάνω. Προς αυτή την κατεύθυνση, χρησιμοποιεί έναν μηχανισμό publish/subscribe, ώστε τα FaaS να εκτελούνται ασύγχρονα με την πυροδότηση κάποιου γεγονότος. Παραγωγός αυτών των μηνυμάτων είναι το webhook-eventsource, ενώ το webhook-sensor είναι ο καταναλωτής που εκτελεί στο τέλος τα FaaS αναλόγως το μήνυμα που κατανάλωσε.

7.2 Αλυσίδα Υπηρεσιών Αναγνώρισης Αντικειμένων (Object Detection Service Chain)



Εικόνα 81 - Αλυσίδα Υπηρεσιών Αναγνώρισης Αντικειμένων

Στην εισαγωγή αυτής της ενότητας θα παρατεθεί μία επιγραμματική περιγραφή όλων των βημάτων που συγκροτούν το διάγραμμα ροών ενεργειών στην υπηρεσία αναγνώρισης αντικειμένων. Συγκεκριμένα η λίστα των απαραίτητων βημάτων τοποθετημένων με χρονική σειρά αποτελείται από τα ακόλουθα:

1. Μία (ή περισσότερες) κάμερες που παράγουν το βίντεο επιτήρησης
2. Μία μικρο-υπηρεσία (micro-service) που επεξεργάζεται το συνεχές σήμα και εξάγει στιγμιότυπα (frames) που τα αποθηκεύει σε ένα ΚΟΙΝΟ δικτυακό χώρο
3. Η καρδιά της προσφερόμενης ευφυούς λειτουργικότητας είναι το μοντέλο αναγνώρισης αντικειμένων. Ανασύρει τα στιγμιότυπα από το κοινό χώρο και εντοπίζει λίστες εμφανιζόμενων οντοτήτων : { κατηγορία για παράδειγμα αυτοκίνητο, θέση στην εικόνα, βεβαιότητα ορθότητας αναγνώρισης / καλό είναι να είναι μεγαλύτερη του 70%}
4. Η παραπάνω λίστα εξάγεται από το ευφύες μοντέλο σε ένα δίαυλο επικοινωνίας (message-queue)
5. Οι υπηρεσίες TANDEM υψηλού επιπέδου ενημερώνονται αυτόματα κάθε φορά που αναρτάται ένα καινούργιο αντικείμενο. Η Υπηρεσία Ευφυούς Στάθμευσης δραστηριοποιείται στην ύπαρξη καινούργιου σταθμευμένου οχήματος. Ενώ η Επόπτευση & Ασφάλεια όταν βρεθεί ένα εγκαταλειμμένο αντικείμενο για παράδειγμα σακίδιο
6. Υπάρχει ο μηχανισμός ειδοποιήσεων που εκκινεί την αποστολή μηνυμάτων όταν η κατάσταση είναι ιδιαίτερη. Ενώ όταν η κατάσταση γίνει κρίσιμη υπάρχει μηχανισμός συνεγερμού.

Η ροή ενεργειών στο σενάριο Εντοπισμού-Αντικειμένων περνάει από τις ακόλουθες φάσεις:

ΦΑΣΗ 1: Προετοιμασία Οικοσυστήματος Τεχνητής Νοημοσύνης και Ενεργοποίηση Προεκπαιδευμένου Μοντέλου

- Ανάγνωση χαρακτηριστικών κάρτας γραφικών (GPU)
- Επιλογή και εγκατάσταση συμβατού οικοσυστήματος τεχνητής-νοημοσύνης:

- Στη προκειμένη περίπτωση PyTorch
- Για την ελαχιστοποίηση των “προστριβών” με υπάρχουσες εκδόσεις εγκατεστημένων εργαλείων, η εγκατάσταση του οικοσυστήματος γίνεται σε ένα CONTAINER που εξασφαλίζει απομόνωση και αυτονομία
- Μετάβαση στο Pre-Trained Model Hub και επιλογή μίας αρχιτεκτονικής που ταιριάζει με το πρόβλημα εντοπισμού ανά-χείρας αλλά και τους διατιθέμενους υπολογιστικούς πόρους:
- Αρχιτεκτονική 1η: YoloV5 καθώς χαρακτηρίζεται από γρήγορες ταχύτητες εντοπισμού. Επίσης αντιμετωπίζει αποδοτικά προβλήματα υπερκαλύψεων και τμηματικών αποκρύψεων
- Αρχιτεκτονική 2η: Facebook/ Detectron2 καθώς προσφέρει μία σειρά από “mask-based” φίλτρα. Πρόκειται για δυναμικές επιφάνειες που μπορούν να κινηθούν από το δυναμικό μίας εικόνας και να εντοπίσουν την ακριβή περιοχή που υπάρχει ένα ζητούμενο αντικείμενο. Με αυτό το τρόπο απαντώνται ταυτόχρονα δύο κρίσιμα ερωτήματα: “Αν υπάρχει ;” και “Σε ποιο σημείο βρίσκεται !”
- Καθορισμός των απαραίτητων λειτουργικών παραμέτρων του μοντέλου:
- Σημείου εγκατάστασης CPU or GPU
- Ποσοστό Ακρίβειας
- Εκκίνηση οικοσυστήματος τεχνητής-νοημοσύνης στο κατάλληλο mode:
- Inference ... υπάρχει εκπαιδευμένο μοντέλο αναγνώρισης έτοιμο να διαχειριστεί άγνωστες εικόνες σε “προιοντικές” εγκαταστάσεις

ΦΑΣΗ 2: Λειτουργία Μηχανισμού Αναγνώρισης Αντικειμένων

- Υπάρχουν πολλές κατανεμημένες κάμερες που παρακολουθούν κρίσιμες υποδομές
- Ο Frame-Grabber διακριτοποιεί το συνεχές σήμα βίντεο σε διαδοχικά frames που αποθηκεύονται σε ένα χώρο με δυνατότητες πρόσβασης από οποιοδήποτε σημείο του TANDEM
- Ο αναγνώστης της υπηρεσίας διαβάζει τις καινούργιες εικόνες και τις αποστέλλει για ευφυή επεξεργασία από το ενεργοποιημένο μοντέλο αναγνώρισης
- Στο τέλος της επεξεργασίας κοινοποιείται μία λίστα με όλα τα αντικείμενα ... αντικείμενο-X { Τίτλος, Περιοχή εντοπισμού, Πιθανότητα Σφάλματος }
- Η λίστα αποστέλλεται στο κεντρικό δίαυλο επικοινωνίας για ενημέρωση των συνεργαζόμενων υπηρεσιών υψηλού επιπέδου, όπως ευφυούς-πάρκινγκ.

8 TANDEM Application

8.1 Parking Application

Η υπηρεσία Ευφυούς Στάθμευσης, που αναλύθηκε σε προηγούμενη ενότητα, είναι μία υπηρεσία που εκτελείται στα άκρα και είναι υπεύθυνη για την επόπτευση ενός συγκεκριμένου χώρου με κύριο μέλημα την διαρκή ενημέρωση σε πραγματικό χρόνο των μετρητικών {πλήθος από ΕΛΕΥΘΕΡΕΣ θέσεις, ΚΩΔΙΚΟΙ αναφοράς τους}. Με άλλα λόγια σε μία περιοχή με πέντε(5) ανεξάρτητους χώρους παρκαρίσματος θα υπάρχουν πέντε(5) υπηρεσίες που θα παρακολουθούν αδιάλειπτα τις ελεύθερες θέσεις που μπορούν να δεσμευτούν για προσερχόμενα αυτοκίνητα.

Η εφαρμογή αυτή ενέχει ένα ρόλο “συγκεντρωτή” (aggregator) όλων των πληροφοριών από τα κατανεμημένα services. Είναι φανερό ότι βρίσκεται σε ένα εκτελεστικό επίπεδο πιο υψηλό και ο ρόλος του είναι διπλός. Πρώτον εποπτεύει τη διαθεσιμότητα θέσεων παρκαρίσματος σε περιοχές και δεύτερον παρέχει ολοκληρωμένες οδηγίες σε οδηγούς που πλησιάζουν και αναζητούν ένα σημείο στάθμευσης.

Το σήμα είσοδου στην εφαρμογή είναι ένα αίτημα (request) - ο οδηγός χρησιμοποιώντας τη κινητή συσκευή του συνδέεται με την υπηρεσία που μπορεί να γίνει προσβάσιμη από το διαδίκτυο. Στη συνέχεια ενημερώνει για τις προθέσεις του: “θα ήθελα να παρκάρω στην οδό Ηπείρου 44 στο Περιστέρι”. Μέσω σήματος GPS το αίτημα του οδηγού εμπλουτίζεται με τη τρέχουσα θέση του.

Όταν η εφαρμογή δεχθεί ένα αίτημα εκτιμάει ποιος χώρος στάθμευσης βρίσκεται κοντύτερα στο ζητούμενο σημείο. Στη συνέχεια ελέγχει τι διαθεσιμότητα εμφανίζει και εάν κριθεί κατάλληλος τότε επιλέγεται και δεσμεύεται μία θέση για ένα χρονικό διάστημα.

9 Συμπεράσματα και Επόμενες Εκδόσεις Λογισμικού

Το παραδοτέο αυτό παρουσιάζει τις πτυχές της τεχνικής υλοποίησης της δεύτερης λειτουργικής έκδοσης του TANDEM που αναδεικνύει την φιλοσοφία λειτουργίας του. Συνοδεύεται από την τεκμηρίωση του λογισμικού, τις οδηγίες εγκατάστασης καθώς και Οδηγίες Χρήσης για τον Τελικό Χρήστη.

Η δουλειά που παρουσιάζεται στο συγκεκριμένο παραδοτέο πραγματοποιείται στο πλαίσιο των εργασιών της ενότητας εργασίας EE2 (Λεπτομερής Σχεδιασμός και Ανάπτυξη Συστήματος) και συνδέεται στενά με τα παραδοτέα Π1.1, Π1.2 που αναλύουν τις απαιτήσεις και την αρχιτεκτονική του συστήματος TANDEM.

Το έργο μέσω επαναλαμβανόμενων κύκλων σχεδιασμού, υλοποίησης, ελέγχου, θα παραδώσει συνολικά 4 εκδόσεις του λογισμικού με διαφορά τεσσάρων μηνών. Κάθε έκδοση του λογισμικού θα έχει σε σχέση με την προηγούμενη επιπλέον λειτουργίες και διορθώσεις/βελτιώσεις στις υπάρχουσες λειτουργίες. Στους κύκλους αυτούς συμμετέχουν και οι διαδικασίες ολοκλήρωσης και γενικού ελέγχου συστήματος (Δ3.1) της Ενότητας Εργασίας 3 (EE3).

Η τελική έκδοση του λογισμικού (4^η έκδοση) θα καλύπτει όλες τις προδιαγραφές συστήματος και είναι και η έκδοση που θα αξιολογηθεί εκτεταμένος στην τελευταία φάση του έργου στα πλαίσια της EE3 η οποία θα παραδώσει και την τελική αξιολόγηση της πλατφόρμας.

Οι εκδόσεις του λογισμικού βασίζονται σε **Docker Containers** και **Kubernetes** ως ενорχηστρωτή της εγκατάστασης, και υποστήριξης κλιμάκωσης (scaling) και διαχείρισης υπηρεσιών σε μορφή Container. Η έκδοση αυτή υποστηρίζει ένα νέφος άκρων (με ένα cluster). Για την υποστήριξη IoT λειτουργιών και εξασφάλιση της διαλειτουργικότητας μεταξύ συσκευών και υπηρεσιών ενσωματώνεται το **EdgeXFoundry**.

Όπως αναφέρθηκε στην Ενότητα 2, η 1^η έκδοση του λογισμικού περιλάμβανε αρχικές εκδόσεις των:

- Διαχειριστή Πλατφόρμας Άκρων (TANDEM Edge Platform Manager ή pi-Edge)
- TANDEM Portal
- Καταλόγου Υπηρεσιών (Service Catalogue)
- Υπηρεσίας Παρακολούθησης Συστήματος (System Monitoring)
- Ενорχηστρωτή Αλυσίδων Υπηρεσιών (Service Chain Orchestrator). .
- Υπηρεσίας Τεχνητής Νοημοσύνης Αναγνώρισης Αντικειμένων (Object Detection service) και της υποστηρικτικής της Υπηρεσίας Διαχειριστής Ζωντανού Σήματος Βίντεο (Video Stream Manager).
- Βασικής Υπηρεσίας Ανάλυση Δεδομένων από IoT Συσκευές (IoT Data Analytics).
- Δύο Υπηρεσιών Εφαρμογών που χρησιμοποιούν την υπηρεσία Αναγνώρισης Αντικειμένων, την Υπηρεσία Έξυπνης Στάθμευσης (Smart Parking Service) και την Υπηρεσία Ασφαλείας (Security Service)

Τα προβλήματα που εντοπίστηκαν καθώς και τα σχόλια και οι παρατηρήσεις στην πρώτη έκδοση του συστήματος από τις προιοντικές διευθύνσεις, χρησιμοποιήθηκαν για τη βελτίωση του στην δεύτερη.

Σύμφωνα με τον προγραμματισμό, η δεύτερη έκδοση του λογισμικού που περιγράφηκε σε αυτό το παραδοτέο, περιλαμβάνει επιπλέον:

- Την ολοκλήρωση της υλοποίησης του TANDEM portal με την προσθήκη όλων των λειτουργιών που έχουν προδιαγραφεί.
- Τη Δεύτερη Έκδοση του **Διαχειριστή Πλατφόρμας Άκρων (TANDEM Edge Platform Manager ή pi-Edge)**, με διορθώσεις στη λειτουργία του και επιπλέον αυτοματισμούς
- Τη Δεύτερη Έκδοση του **Καταλόγου Υπηρεσιών (Service Catalogue)** και της **Υπηρεσίας Παρακολούθησης Συστήματος (System Monitoring Service)**
- Την πρώτη λειτουργική έκδοση της Υπηρεσίας **Διαχειριστή Χρηστών (User Manager)** που θα περιλαμβάνει αυστηρούς μηχανισμούς εξουσιοδότησης σύμφωνα και με το ρόλο του χρήστη στην πλατφόρμα του TANDEM.
- Ολοκλήρωση της επικοινωνίας του γραφικού περιβάλλοντος του **TANDEM Portal** με τους μηχανισμούς διαχείρισης της υποδομής (αρχικοποίηση κόμβου TANDEM, ενεργοποίηση Βασικών Υπηρεσιών σε αυτό, κλπ.).

Επιπλέον του αρχικού προγραμματισμού, η έκδοση αυτή περιλαμβάνει και το MEC service “Location APIs” (αρχικός προγραμματισμός για τις εκδόσεις 3 και 4) που είναι ένα σύνολο από διεπαφές, οι οποίες παρέχονται από το **MEC Platform** και προσφέρουν πληροφορίες που αφορούν την τοποθεσία διαφόρων User Equipments, τον υπολογισμό αποστάσεων, καθώς και την εύρεση του/των κοντινότερων UE(s).

Η MEC αυτή υπηρεσία, χρησιμοποιήθηκε πειραματικά στην πρωτότυπη αλυσίδα υπηρεσιών που υλοποιήθηκε ως μέρος της πρώτης έκδοσης και αφορά τη διαδικασία από τη συλλογή δεδομένων από IoT devices (αισθητήρες θερμοκρασίας και υγρασίας) μέχρι τη ανάλυση τους, τη γραφική αναπαράστασή τους και τις ενέργειες που λαμβάνουν χώρα ανάλογα με τις τιμές τους.

Η επόμενη έκδοση (η Τρίτη) του συστήματος, σύμφωνα με τον προγραμματισμό πέρα από τις διορθώσεις και βελτιώσεις στη Δεύτερη έκδοση, θα περιλαμβάνει:

- Την υποστήριξη περισσότερων από ένα νέφη άκρων (edge clouds) με την υλοποίηση του **Ενορχηστρωτή Άκρων (Edge Orchestrator)** που διαχειρίζεται την υποδομή, τους πόρους και την λειτουργία των υπηρεσιών όλων των edge clouds επικοινωνώντας με τον Διαχειριστή Πλατφόρμας Άκρων καθενός από αυτούς.
- Την υλοποίηση του **Διαχειριστή Συσκευών (Device Manager)** που θα είναι υπεύθυνος για τη διαχείριση των συσκευών όλων των edge clouds κεντρικά
- Τους μηχανισμούς πρόβλεψης της μελλοντικής χρήσης των πόρων των κόμβων με χρήση Τεχνητής Νοημοσύνης οι οποίοι θα ενσωματωθούν στον Διαχειριστή Πλατφόρμας Άκρων.
- Την προσθήκη μηχανισμών πρόβλεψης μελλοντικών τιμών σε μετρήσεις από αισθητήρες στην **Ανάλυση Δεδομένων από IoT Συσκευές (IoT Data Analytics)**
- Αυτοματοποίηση της Εκτέλεσης Αλυσίδων Υπηρεσιών. Για τα το σκοπό αυτό θα γίνει αρχική υλοποίηση της αυτόματης εξαγωγής (α) της εσωτερικής αναπαράστασης μιας Αλυσίδας Υπηρεσιών στην πλατφόρμα TANDEM στην αναπαράσταση που υποστηρίζεται από το Workflow εργαλείο Argo και (β) πληροφοριών που θα

χρειαστεί ο Διαχειριστής Πλατφόρμας Άκρων σχετικά με τους πόρους και τις υπηρεσίες που απαιτούνται για την εκτέλεσή της.

- Υλοποίηση σεναρίου Προσωρινής Αποθήκευση βίντεο (video caching) στα άκρα του δικτύου σε Δίκτυα Διανομής Περιεχομένου (CDN) με εφαρμογή στο προϊόν της εταιρείας **fs|cdn™ Anywhere (Full Service Content Delivery Network)**

Στη διάρκεια υλοποίησης της τρίτης και τέταρτης έκδοσης, προγραμματίζεται να χρησιμοποιηθεί και αξιοποιηθεί περισσότερο το ETSI Location Service που υλοποιήθηκε και να εξεταστεί και η χρήση και άλλων ETSI MEC Service APIs (<https://forge.etsi.org/rep/mec>).

Η τελική έκδοση του λογισμικού (4^η έκδοση), πέρα από τις διορθώσεις και βελτιώσεις στη τρίτη έκδοση, θα πρέπει να καλύπτει όλες τις προδιαγραφές του συστήματος. Είναι η έκδοση που θα αξιολογηθεί εκτεταμένα στην τελευταία φάση του έργου στα πλαίσια της Ενότητας Εργασίας 3 (ΕΕ3). Η ενότητα αυτή θα παραδώσει και την τελική αξιολόγηση της πλατφόρμας.

Έτσι για την τελική έκδοση του λογισμικού συστήματος:

- Θα υλοποιηθούν οι λειτουργίες μετατροπής μιας υπηρεσίας ή μιας εφαρμογής σε προϊόν με την προσθήκη των στοιχείων συμφωνίας με τον πελάτη (SLA) καθώς και των στοιχείων τιμολόγησης. Έτσι το σύστημα θα έχει δύο επιπλέον λειτουργίες: την **Διαχείριση Προϊόντος (Product Manager)** και τη **Διαχείριση Χρέωσης (Billing Manager)**
- Θα υλοποιηθεί ο **Διαχειριστής Υπηρεσιών (Service Manager)** ο οποίος θα χρησιμοποιεί την υπηρεσία **Τεχνητής Νοημοσύνης (Artificial Intelligence)** που προβλέπει την μελλοντική χρήση των πόρων σε κάθε κόμβο και αποφασίζει για τα στιγμιότυπα των υπηρεσιών που θα ενεργοποιούνται σε κάθε έναν από αυτούς και ποιος θα είναι ο κύκλος ζωής τους, λαμβάνοντας επιπλέον υπόψη τα στοιχεία προσβασιμότητας στις συσκευές IoT από κάθε κόμβο, τις απαιτήσεις σε πόρους κάθε υπηρεσίας, τα στοιχεία της συμφωνίας με τον πελάτη (SLA) και τη συμμετοχή μιας υπηρεσίας σε μία αλυσίδα υπηρεσιών που επικοινωνούν.
- Θα υλοποιηθούν τα σενάρια επίδειξης του TANDEM σε περιβάλλον Βιομηχανικού IoT με χρήση συνεργαζόμενων ρομπότς όπως αυτά περιγράφονται στο παραδοτέο Π.1.1.

10 Αναφορές

[ALBUMENTATIONS] Albumentations – computer vision tool, [Ηλεκτρονικά], <https://albumentations.ai/>

[Argo] Argo Projects, <https://argoproj.github.io>

[Bootstrap] Βιβλιοθήκη HTML, CSS και JS, [Ηλεκτρονικά], <https://www.getbootstrap.com/>

[Detectron2] detectron2, [Ηλεκτρονικά], <https://github.com/facebookresearch/detectron2/>

[Docker] Docker, [Ηλεκτρονικά], <https://www.docker.com/>

[EdgeX] EdgeXFoundry, [Ηλεκτρονικά], <https://www.edgexfoundry.org/>

[Grafana] Grafana, [Ηλεκτρονικά], <https://grafana.com/>

[InfluxDB] Influx Time Series Data Base, [Ηλεκτρονικά], <https://www.influxdata.com/>

[jQuery] Βιβλιοθήκη Javascript, [Ηλεκτρονικά], <https://www.jquery.com/>

[K8S] Kubernetes, [Ηλεκτρονικά], <https://kubernetes.io/>

[Keycloak] <https://www.keycloak.org/>

[MEC] ETSI MEC, <https://www.etsi.org/technologies/multi-access-edge-computing>

[MicroK8s] Lightweight Kubernetes, [Ηλεκτρονικά], <https://microk8s.io/>

[MongoDB] <https://www.mongodb.com/>

[Mosquitto] Eclipse Mosquitto Mqtt Broker, [Ηλεκτρονικά], <https://mosquitto.org/>

[Nginx] <https://www.nginx.com/>

[OpenCV] Open Source Computer Vision Library, [Ηλεκτρονικά], <https://opencv.org/>

[OSCM] Open Service Catalogue Manager, <https://openservicecatalogmanager.org/>

[PyTorch] PyTorch, [Ηλεκτρονικά], <https://pytorch.org>

[Prometheus] Prometheus, [Ηλεκτρονικά], <https://prometheus.io/>

[Prometheus-Operator] Prometheus Operator, [Ηλεκτρονικά], <https://github.com/prometheus-operator/prometheus-operator>

[Spring Boot] <https://spring.io/projects/spring-boot>

[TensorFlow], TensorFlow, <https://tensorflow.org>

[TMF633], “Service Catalog Management API REST Specification”, TMF633, Release 18.5.0, January 2019

[Yolo] Yolo <https://github.com/ultralytics/yolov5>

11 ΠΑΡΑΡΤΗΜΑ - Προετοιμασία & Εγκατάσταση

Στο παράρτημα παρατίθενται αναλυτικές οδηγίες που καλύπτουν όλο το φάσμα ενεργειών για εγκατάσταση της πλατφόρμας TANDEM. Αναφέρονται πρωταρχικά ζητήματα όπως επιλογή κατάλληλου εξυπηρετητή (server) σε επίπεδο φυσικού μηχανήματος και λειτουργικού συστήματος (OS).

Στη συνέχεια παρουσιάζεται η στρατηγική επιλογή του IaC (Infrastructure-as-a-Code) που επιτρέπει την πλήρη αυτοματοποίηση της διαδικασίας “κατακερματισμού” των φυσικών πόρων σε εικονικοποιημένους (Virtuals). Με τη βοήθεια scripts και κατάλληλων εργαλείων γράφονται όλες οι οδηγίες σε μορφή κώδικα και εκτελούνται κάθε φορά απaráλλαχτες με εγγυημένα αποτελέσματα.

Όταν δημιουργηθούν όλο το απαραίτητο οικοσύστημα εγκαθίσταται ένας ενορχηστρωτής Kubernetes με πολλούς κατανεμημένους κόμβους-εργάτες που είναι έτοιμοι να φιλοξενήσουν υπηρεσίες της πλατφόρμας TANDEM.

Στο τέλος δίνεται μία ενδελεχής ματιά στην εγκατάσταση και λειτουργία των υπηρεσιών τεχνητής-ευφυίας. Γίνεται παρουσίαση των αρχείων διανομής της υπηρεσίας αναγνώρισης-αντικειμένων σε περιβάλλον Kubernetes που αποτελεί τη “ραχοκοκαλιά” για τις υπηρεσίες υψηλού επιπέδου: Έξυπνο-πάρκινγκ & Ασφάλεια-χώρου.

11.1 Προετοιμασία Υπολογιστικού Εξυπηρετητή (server)

Η προσπάθεια ξεκινάει με ένα εξυπηρετητή που τρέχει σε περιβάλλον Linux/ Ubuntu (20.04) LTS.

Ένα βασικό προαπαιτούμενο είναι η εγκατάσταση του μηχανισμού δημιουργίας και διαχείρισης εικονικών-μηχανών (VMs) [cross-platform virtualization software]. Η λύση που προκρίθηκε είναι το VirtualBox (www.virtualbox.org) από την Oracle. Για την εγκατάσταση του χρησιμοποιούνται τα επίσημα σημεία διανομής πακέτων της Ubuntu:

```
$ sudo apt update
```

```
$ sudo apt install virtualbox virtualbox-ext-pack
```

Μετά την εγκατάσταση του συγκεκριμένου πακέτου μπορεί να γίνει εύκολα κατακερματισμός του παρεχόμενου hardware σε εικονικές-μηχανές με διαφορετικά λειτουργικά συστήματα και υπολογιστικές δυνατότητες.

Σειρά έχει ο μηχανισμός αυτοματοποίησης του έργου δημιουργίας εικονικών υποδομών. Κάθε φορά που επαναλαμβάνεται η καταχώρηση εντολών στο σύστημα υπάρχει ο κίνδυνος λαθών με απρόβλεπτες συνέπειες. Για αυτό γίνεται αποτύπωση της διαδικασίας σε ένα αρχείο που τρέχει απaráλαχτο. Το εργαλείο με το όνομα Vagrant μπορεί να καταλάβει τις οδηγίες που μπαίνουν σε τέτοια αρχεία και να προχωρήσει στη δημιουργία των VMs.

Για την εγκατάσταση του Vagrant θα μπορούσαν να χρησιμοποιηθούν τα επίσημα σημεία διανομής της Ubuntu αλλά συνήθως δεν έχουν τη πιο πρόσφατη και ώριμη έκδοση του εν-λόγω εργαλείου. Για αυτό το σκοπό το πιο σωστό είναι να αποταθεί κανείς στη κατασκευάστρια εταιρία και να κάνει download το LATEST παράγωγο:

```
$ curl -O https://releases.hashicorp.com/vagrant_2.2.9_x86_64.deb
```

```
$ sudo apt install ./vagrant_2.2.9_x86_64.deb
```

```
$ vagrant --version
```

Η παραπάνω ακολουθία εντολών αποθήκευσε στο τοπικό εξυπηρετητή ένα αντίγραφο του προγράμματος εγκατάστασης στη συνέχεια το έτρεξε και στο τέλος δοκίμασε να δει αν η απόπειρα ήταν πετυχημένη. Το Vagrant είναι παρόν και έτοιμο προς χρήση.

11.2 Δημιουργία Εικονικού-Νέφους για εγκατάσταση Kubernetes με Ένα(1) Master & N Workers

Το Vagrant προσφέρει τη δυνατότητα στο διαχειριστή να γράψει οδηγίες για εικονικές-μηχανές μία φορά και να τις τρέξει «άπειρες» με εγγυημένο τελικό αποτέλεσμα. Το αρχείο με τις οδηγίες έχει μία απλή text-μορφή που είναι εύκολο στην ανάγνωση και φέρει τη χαρακτηριστική ονομασία Vagrantfile. Ένα μέρος της προσπάθειας εγκατάστασης του Kubernetes με αυτοματοποιημένο και δυναμικό τρόπο αποτέλεσε η συγγραφή του κατάλληλου Vagrantfile. Το βασικό ζητούμενο ήταν η διαστασιολόγηση του εικονικού-νέφους να γίνεται σε πραγματικό χρόνο, δηλαδή ο αριθμός των Worker-nodes να μην είναι πάντα ο ίδιος. Στη συνέχεια επισυνάπτεται ένα τμήμα του:

```
IMAGE_NAME = "bento/ubuntu-16.04"
```

```
#IMAGE_NAME="ubuntu/focal64"
```

```
N = 2
```

```
RABBITMQ_PORT = 31960
```

```
RABBITMQ_HOST_PORT = 18000
```

```
REDIS_PORT = 31961
```

```
REDIS_HOST_PORT = 18001
```

```
CONSUL_UI_PORT = 31962
```

```
CONSUL_UI_HOST_PORT = 18002
```

```
PORT_ARGO_UI = 31963
```

```
PORT_ARGO_UI_HOST = 18003
```

```
METRICS_SERVER_PORT = 31964
```

```
METRICS_SERVER_HOST_PORT = 18004
```

```
STACK_PROMETHEUS_PORT = 31972
```

```
STACK_PROMETHEUS_HOST_PORT = 18005
```

```
STACK_GRAFANA_PORT = 31974
```

```
STACK_GRAFANA_HOST_PORT = 18006
```

```
STACK_ALERTMANAGER_PORT = 31976
```

```
STACK_ALERTMANAGER_HOST_PORT = 18007
```

```
KUBEVIEW_PORT = 31978
```

```
KUBEVIEW_HOST_PORT = 18009
```

```
AIRFLOW_PORT = 31980
```

```
AIRFLOW_HOST_PORT = 18008
```

```
JAEGER_PORT = 31982
```

```
JAEGER_HOST_PORT = 18010
```

```
$script = <<-SCRIPT
```

```
    sudo http_proxy=http://icache.intracomtel.com:80 apt-get update
```

```
    sudo http_proxy=http://icache.intracomtel.com:80 apt-get install -y emacs lynx
```

```
SCRIPT
```

```
Vagrant.configure("2") do |config|

  config.ssh.insert_key = false

  config.vm.provider "virtualbox" do |v|

    v.memory = 2048

    v.cpus = 2

  end

  if Vagrant.has_plugin?("vagrant-proxyconf")

    config.proxy.http = "http://icache.intracomtel.com:80"

    config.proxy.https = "http://icache.intracomtel.com:80"

    config.proxy.no_proxy = "localhost,127.0.0.1,192.168.0.0/16,10.0.0.0/24"

  end

  # Enable NFS Synced Folders ... Host-Server {VS} All VM-Client

  config.vm.synced_folder "/var/nfs/general/", "/data", type: "nfs", nfs_export: false

  #
```

Στις πρώτες γραμμές γίνεται επιλογή του λειτουργικού-συστήματος (OS) που θα τρέχει σε κάθε VM. Οι επιλογές είναι πολλές με προεξέχουσες τις διάφορες εκδόσεις του Linux Ubuntu, Centos, ...

Στη συνέχεια ακολουθεί ο προσδιορισμός μίας σημαντικής λειτουργικής παραμέτρου με την ονομασία N και αποτυπώνει το πλήθος των worker-nodes που θα έχει το Kubernetes-νέφος.

Οι επόμενες οδηγίες αντιστοιχούν στο Port-Forwarding δηλαδή στην αυτόματη διασύνδεση κάποιων LISTENing-Ports στα VMs με κάποιες στο εξυπηρετητή-φιλοξενίας. Αυτό γίνεται γιατί τα VMs βρίσκονται σε μία ιδιωτική-δικτύωση (private-network) και δεν δέχονται απομακρυσμένα requests από άλλους εξυπηρετητές. Η λύση είναι η συσχέτιση των «εικονικών-θυρών» με πραγματικές ώστε να επιτυγχάνεται η προώθηση των μηνυμάτων.

Οι σημαντικότερες οδηγίες του Vagrantfile βρίσκονται στη παράγραφο **Vagrant.configure("2") do |config| !**

Είναι το σημείο που προσδιορίζονται το IP της εικονικής-μηχανής αλλά και λειτουργικά χαρακτηριστικά όπως CPU & Memory.

Η δυνατότητα του κοινού αποθηκευτικού χώρου ανάμεσα στα VMs

Οι εικονικές-μηχανές έχουν εφήμερο χαρακτήρα δηλαδή αν «σταματήσουν» θα χαθούν όλα τα δεδομένα τους. Επίσης λειτουργούν σε ένα πλαίσιο απομόνωσης δηλαδή δεν έχουν τη δυνατότητα ανταλλαγής δεδομένων με γειτονική δομή.

Για αυτό το λόγο γίνεται δέσμευση αποθηκευτικού χώρου στο εξυπηρετητής-φιλοξενίας που «συστήνεται» στα VMs με συγκεκριμένα προνόμια διαχείρισης. Για παράδειγμα η ακόλουθη οδηγία δίνει πρόσβαση στο VM : «τοπικός υποκατάλογος» /var/nfs/general/ , εμφανίζεται σαν «vm-filesystem» /data , «μηχανισμός υλοποίησης» NFS

```
config.vm.synced_folder "/var/nfs/general/", "/data", type: "nfs", nfs_export: false
```

Η επιτυχία αυτού του μηχανισμού έγκειται στη δυνατότητα που παρέχει σε διαφορετικά VMs να έχουν ένα σταθερό χώρο αποθήκευσης και να ανταλλάσσουν αρχεία. Το VM-01 μπορεί να παράγει ένα μεγάλο όγκο πληροφοριών που τα «αποστέλλει» σε ένα αρχείο στο κοινό χώρο ενώ το VM-02 τα διαβάζει και τα επεξεργάζεται.

11.3 Εγκατάσταση Απαραίτητου Software στο Εικονικό-Νέφος

Μετά την δημιουργία των απαραίτητων εικονικών-μηχανών ακολουθεί η «στελέχωση» τους με τα κατάλληλα εργαλεία. Ο στόχος είναι διπλός πρώτον η εγκατάσταση του ενορχηστρωτή Kubernetes και δεύτερον η εκκίνηση των υπηρεσιών τεχνητής-νοημοσύνης στο οικοσύστημα του ενορχηστρωτή.

Αναφέρθηκε στην αρχή η αξία ενός εργαλείου αυτοματισμού δημιουργίας της υποδομής και παρουσιάστηκε η τρέχουσα επιλογή Vagrant. Σε αυτή τη φάση θα χρησιμοποιηθεί ένα παρεμφερές εργαλείο αυτοματισμού αλλά αυτή τη φορά για την εγκατάσταση πακέτων/εργαλείων στα VMs. Το όνομα του είναι Ansible (<https://www.ansible.com/>) και είναι σε θέση να διεκπεραιώσει ζητήματα προετοιμασίας μηχανών από άποψη χρηστών, υπηρεσιών, προσωποποίησης και αυθεντικοποίησης, κοκ. Οι οδηγίες για τις απαραίτητες επεμβάσεις που πρέπει να γίνουν σε ένα καινούργιο VM αναγράφονται σε ένα απλό αρχείο-κειμένου. Σε μία εγκατάσταση Kubernetes απαιτούνται τουλάχιστον δύο εικονικές-μηχανές. Μία που λέγεται Master και έχει κεντρικό ρόλο καθώς συντονίζει τη λειτουργία των άλλων. Και μία ομάδα από workers που φιλοξενούν τις containerized-applications που έχουν διανεμηθεί. Επόμενο είναι η κάθε VM ανάλογα με το ρόλο της να απαιτεί και διαφορετικό συσχετισμό εργαλείων.

Στη συνέχεια ακολουθεί ένα απόσπασμα από τις οδηγίες που έχουν γραφτεί για το Ansible ώστε να προετοιμάσει το κόμβο Master:

\$ emacs master-playbook.yml

```
---
- hosts: all
```

```
become: false

tasks:
- name: Install packages that allow apt to be used over HTTPS

apt:
  name: "{{ packages }}"
  state: present
  update_cache: yes

environment:
  http_proxy: http://icache.intracomtel.com:80
  https_proxy: http://icache.intracomtel.com:80

vars:
  packages:
    - apt-transport-https
    - ca-certificates
    - curl
    - gnupg-agent
    - software-properties-common
    - sshpass
    - openssh-server

become: true

- name: Verify SSH service is running
  command: systemctl ssh status
  failed_when: false
  become: true

- name: Enable Password Authentication to access SSH server
```

```
ansible.builtin.lineinfile:
```

```
  path: /etc/ssh/sshd_config
```

```
  search_string: 'PasswordAuthentication no'
```

```
  line: 'PasswordAuthentication yes'
```

```
  owner: root
```

```
  group: root
```

```
become: true
```

```
- name: Restart SSH service
```

```
ansible.builtin.systemd:
```

```
  state: restarted
```

```
  daemon_reload: yes
```

```
  name: sshd
```

```
become: true
```

```
- name: Add an apt signing key for Docker
```

```
apt_key:
```

```
  url: https://download.docker.com/linux/ubuntu/gpg
```

```
  state: present
```

```
become: true
```

```
- name: Add apt repository for stable version
```

```
apt_repository:
```

```
  repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable
```

```
  state: present
```

```
become: true
```

- name: Install docker and its dependencies

apt:

name: "{{ packages }}"

state: present

update_cache: yes

environment:

http_proxy: http://icache.intracomtel.com:80

https_proxy: http://icache.intracomtel.com:80

vars:

packages:

- docker-ce

- docker-ce-cli

- containerd.io

- python3-pexpect

become: true

notify:

- docker status

- name: Add vagrant user to docker group

user:

name: vagrant

group: docker

become: true

- name: Define HTTP_PROXY, HTTPS_PROXY, and NO_PROXY for Docker Daemon

become: true

ansible.builtin.script: /home/vibm/kubernetes/kubernetes-setup/docker-httpsProxy.sh

```
register: docker_proxy_setup

- ansible.builtin.debug:

  var: docker_proxy_setup

- name: Google Cloud SDK Installation

  become: true

  ansible.builtin.script:      /home/vibm/kubernetes/kubernetes-setup/google-cloud-
installation.sh

- name: Google Cloud Account Authentication

  become: true

  ansible.builtin.command: "{{ item }}"

  with_items:

    - gcloud config set proxy/type http

    - gcloud config set proxy/address icache.intracomtel.com

    - gcloud config set proxy/port 80

    - gcloud auth activate-service-account learning-cloud@appspot.gserviceaccount.com --key-
file=learning-cloud-226142d8e000.json

- name: Configure Docker with google-cloud authentication

  ansible.builtin.expect:

    command: gcloud auth configure-docker

    responses:

      (.*)Do you want to continue(.*) : "Y"

- name: "failed to run Kubelet: misconfiguration: kubelet cgroup driver: \"systemd\" is
different from docker cgroup driver: \"cgroupfs\""

  ansible.builtin.command: "{{ item }}"
```

```
with_items:

  - mv daemon.json /etc/docker/

  - systemctl daemon-reload

  - systemctl restart docker

become: yes

- name: Remove swapfile from /etc/fstab

mount:

  name: "{{ item }}"

  fstype: swap

  state: absent

with_items:

  - swap

  - none

become: true

- name: Disable swap

  command: swapoff -a

  when: ansible_swaptotal_mb > 0

  become: true

- name: Copy shell-script to ... install-argo-workflows helmChart

  ansible.builtin.copy:

    src: /home/vibm/kubernetes/kubernetes-setup/install-argo-workflows.sh

    dest: /home/vagrant/install-argo-workflows.sh

    mode: u+x

- name: Setup NFS server on master node and export /opt/certs & /opt/registry as nfs share
```

```
ansible.builtin.include_tasks:

  file: "setup-nfs-server-on-master-node.yaml"

- name: prepare Camera4Me in master-node

  ansible.builtin.include_tasks:

    file: "prepare-camera4Me-k8s.yaml"

- name: setup private docker registry in Kubernetes

  ansible.builtin.include_tasks:

    file: "setup-private-docker-registry-k8s.yaml"

- name: Copy all shell-scripts for deployments automation

  ansible.builtin.copy:

    src: "{{ item }}"

    dest: "/home/vagrant/tandem/"

    mode: "u+x"

  with_items:

    - /home/vibm/kubernetes/kubernetes-setup/create-ns-tandem.sh

    - /home/vibm/kubernetes/kubernetes-setup/deploy-private-repository.sh

    - /home/vibm/kubernetes/kubernetes-setup/create-registry-svc.sh

    - /home/vibm/kubernetes/kubernetes-setup/run-tandem-pods.sh

    - /home/vibm/kubernetes/kubernetes-setup/delete-tandem-pods.sh

    - /home/vibm/kubernetes/kubernetes-setup/run-watchdog-pod.sh

    - /home/vibm/kubernetes/kubernetes-setup/run-system-monitoring.sh

    - /home/vibm/kubernetes/kubernetes-setup/run-grabber-pod.sh

    - /home/vibm/kubernetes/kubernetes-setup/create-registry-push-tandem-images.sh

    - /home/vibm/kubernetes/kubernetes-setup/download-new-images-run-again-pods.sh
```

```
- /home/vibm/kubernetes/kubernetes-setup/download-new-watchdog-image.sh

#- /home/vibm/kubernetes/kubernetes-setup/download-new-grabber-image.sh

- /home/vibm/kubernetes/kubernetes-setup/deploy-grafana-k8s.sh

- /home/vibm/kubernetes/kubernetes-setup/grafana-manifest.yaml

- /home/vibm/kubernetes/kubernetes-setup/deploy-redis-k8s.sh

- /home/vibm/kubernetes/kubernetes-setup/redis-official-full.yaml

- /home/vibm/kubernetes/kubernetes-setup/deploy-prometheus-stack-svc.sh

- /home/vibm/kubernetes/kubernetes-setup/prometheus-stack-manifest.yaml

- /home/vibm/kubernetes/kubernetes-setup/send-HTTP-requests-stack.sh

- /home/vibm/kubernetes/kubernetes-setup/persistent-volume-local-manifest.yaml

- /home/vibm/kubernetes/kubernetes-setup/apply-persistent-volume-local.sh


- name: Copy registry-certificate file to master node

  ansible.builtin.command:

    cmd: "{{ item }}"

  with_items:

    - cp /opt/certs/registry.crt /usr/local/share/ca-certificates/

    #- update-ca-certificates

    #- systemctl restart docker

  become: true

  ignore_errors: false


- name: Hold new certificates

  ansible.builtin.command:

    cmd: update-ca-certificates

  become: true

  register: response_certs
```

```
- ansible.builtin.debug:

  var: response_certs

- name: Restart docker after SSL cert renewal

  ansible.builtin.command:

    cmd: systemctl restart docker

  become: true

# - name: Download Node-Exporter-Full Dashboard

#  ansible.builtin.get_url:

#    url:      https://raw.githubusercontent.com/rfrail3/grafana-
dashboards/master/prometheus/node-exporter-full.json

#  dest: /home/vagrant/node-exporter-full.json
```

Το πρώτο βήμα στην προετοιμασία του Master-Node περιλαμβάνει την ενημέρωση των εγκατεστημένων πακέτων και την προσθήκη καινούργιων όπως είναι το «curl».

Στη συνέχεια γίνεται έλεγχος της λειτουργίας σημαντικών υπηρεσιών στο VM όπως είναι το «ssh». Είναι απαραίτητη η επιβεβαίωση ότι τρέχει και είναι αποκρίσιμη αλλά και ο έλεγχος της λίστας των χρηστών με άδεια εισόδου.

Το Docker έχει κομβικό ρόλο στη λειτουργία του Kubernetes αλλά και των εφαρμογών που φιλοξενεί. Πρέπει πριν από οποιαδήποτε απόπειρα εγκατάστασης να υπάρχει διαθέσιμη μία μηχανή containerization – όχι απαραίτητα ο Docker. Δεν αρκεί μόνο μία εγκατάσταση της αλλά πρέπει να είναι ενημερωμένη με μία πρόσφατη ώριμη έκδοση. Το καλύτερο σημείο να κατεβάσει κανείς binaries για την αρχιτεκτονική του σερβερ-φιλοξενίας είναι μέσω του επίσημου site του Docker. Η ταυτόχρονη λειτουργία του Kubernetes με το containerization-engine απαιτεί τη συμμόρφωση τους σε κάποιος κοινούν κανόνες όπως είναι τα CGROUPS (control-groups). Το οικοσύστημα του Docker περιλαμβάνει μία σειρά από εργαλεία όπως ένα backend-engine και ένα command-line-inteface. Επίσης εγκαθίσταται ένας Linux-daemon για να μπει στην ακολουθία των συστημικών υπηρεσιών που ξεκινάνε και σταματάνε από το λειτουργικό-σύστημα σε κάθε εκκίνηση και τερματισμό.

Η επόμενη ενέργεια που κάνει το Ansible είναι η μεταφορά από το διαδίκτυο στο τοπικό εξυπηρετητή και στη συνέχεια στο VM των binaries του Kubernetes για αρχιτεκτονική Linux και amd64. Για την ολοκλήρωση της απαιτείται η μετάβαση σε ένα repository της Google και με την επίδειξη του απαραίτητου πιστοποιητικού να αιτηθεί τη τελευταία έκδοση του

ενορχηστρωτή. Για την σωστή εγκατάσταση και εύρυθμη λειτουργία του Kubernetes απαιτείται η επανεκκίνηση κάποιων υπηρεσιών του Linux με καινούργιο configuration όπως είναι το swap-disk.

Το τελευταίο βήμα περιλαμβάνει τη μεταφορά κάποιων αρχείων στο VM που τοποθετούνται σε ένα υποκατάλογο με τη κωδική ονομασία «tandem». Όλη η προσπάθεια εγκατάστασης και εκκίνησης σε ένα EDGE του προϊόντος έχει αυτοματοποιηθεί με αποκλειστικό σκοπό την διευκόλυνση του εκάστοτε διαχειριστή. Υπάρχουν μία σειρά από εκτελέσιμα scripts που ξεδιπλώνουν τη διαδικασία και στο τέλος αφήνουν μια εγκατάσταση του TANDEM έτοιμη να δεχθεί αιτήματα από χρήστες.

\$ emacs worker-playbook.yml

```
---

- hosts: all

  become: true

  tasks:

    - name: Install packages that allow apt to be used over HTTPS

      apt:

        name: "{{ packages }}"

        state: present

        update_cache: yes

      environment:

        http_proxy: http://icache.intracomtel.com:80

        https_proxy: http://icache.intracomtel.com:80

      vars:

        packages:

          - apt-transport-https

          - ca-certificates

          - curl

          - gnupg-agent

          - software-properties-common
```

```
- sshpass

- name: Add an apt signing key for Docker

apt_key:

  url: https://download.docker.com/linux/ubuntu/gpg

  state: present

environment:

  http_proxy: http://icache.intracomtel.com:80

  https_proxy: http://icache.intracomtel.com:80

- name: Add apt repository for stable version

apt_repository:

  repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable

  state: present

environment:

  http_proxy: http://icache.intracomtel.com:80

  https_proxy: http://icache.intracomtel.com:80

- name: Install docker and its dependencies

apt:

  name: "{{ packages }}"

  state: present

  update_cache: yes

environment:

  http_proxy: http://icache.intracomtel.com:80
```

```
https_proxy: http://icache.intracomtel.com:80

vars:

packages:

- docker-ce

- docker-ce-cli

- containerd.io

- python3-pexpect

notify:

- docker status


- name: Add vagrant user to docker group

user:

  name: vagrant

  group: docker


- name: Define HTTP_PROXY, HTTPS_PROXY, and NO_PROXY for Docker Daemon

become: yes

ansible.builtin.script: /home/vibm/kubernetes/kubernetes-setup/docker-httpsProxy.sh


- name: Google Cloud SDK Installation

become: yes

ansible.builtin.script: /home/vibm/kubernetes/kubernetes-setup/google-cloud-
installation.sh


- name: Google Cloud Account Authentication

become: yes
```

```
ansible.builtin.command: "{{ item }}"

with_items:

- gcloud config set proxy/type http

- gcloud config set proxy/address icache.intracomtel.com

- gcloud config set proxy/port 80

- gcloud auth activate-service-account learning-cloud@appspot.gserviceaccount.com --key-
file=learning-cloud-226142d8e000.json


- name: Configure Docker with google-cloud authentication

ansible.builtin.expect:

  command: gcloud auth configure-docker

  responses:

    (.*)Do you want to continue(. *): "Y"


- name: "failed to run Kubelet: misconfiguration: kubelet cgroup driver: \"systemd\" is
different from docker cgroup driver: \"cgroupfs\""

ansible.builtin.command: "{{ item }}"

with_items:

- mv daemon.json /etc/docker/

- systemctl daemon-reload

- systemctl restart docker

become: yes


- name: Remove swapfile from /etc/fstab

mount:

  name: "{{ item }}"
```

```
fstype: swap

state: absent

with_items:

  - swap

  - none

- name: Disable swap

  command: swapoff -a

  when: ansible_swaptotal_mb > 0

- name: Add an apt signing key for Kubernetes

  apt_key:

    url: https://packages.cloud.google.com/apt/doc/apt-key.gpg

    state: present

  environment:

    http_proxy: http://icache.intracomtel.com:80

    https_proxy: http://icache.intracomtel.com:80

- name: Add apt repository for kubernetes

  apt_repository:

    repo: deb https://apt.kubernetes.io/ kubernetes-xenial main

    state: present

    filename: kubernetes.list

  environment:

    http_proxy: http://icache.intracomtel.com:80
```

```
https_proxy: http://icache.intracomtel.com:80

- name: Install kubernetes binaries

apt:

  name: "{{ packages }}"

  state: present

  update_cache: yes

environment:

  http_proxy: http://icache.intracomtel.com:80

  https_proxy: http://icache.intracomtel.com:80

vars:

  packages:

    - kubelet

    - kubeadm

    - kubectl

- name: Configure node ip

lineinfile:

  path: /etc/default/kubelet

  line: KUBELET_EXTRA_ARGS=--node-ip={{ node_ip }}

  create: yes

- name: Restart kubelet

service:

  name: kubelet
```

```
daemon_reload: yes

state: restarted


- name: Copy the join command to server location

  copy:    src="/home/vibm/kubernetes/join-command"    dest="/tmp/join-command.sh"
mode=0777


- name: Join the node to cluster

  command: echo HELLO #sh /tmp/join-command.sh


- name: Install NFS-functionality on client

  ansible.builtin.include_tasks:

    file: "setup-nfs-client-on-worker-node.yaml"


- name: Copy registry certificate file to worker node

  ansible.builtin.command:

    cmd: "{{ item }}"

  with_items:

    - cp /opt/certs/registry.crt /usr/local/share/ca-certificates/

    #- update-ca-certificates

    # - systemctl restart docker

  become: true

  ignore_errors: false


- name: Hold new certificates

  ansible.builtin.command:
```

```
cmd: update-ca-certificates

become: true

register: response_certs

- ansible.builtin.debug:

    var: response_certs

- name: Restart docker after SSL cert renewal

  ansible.builtin.command:

    cmd: systemctl restart docker

    become: true

- name: Create "grafana" folder in /home/vagrant/

  ansible.builtin.file:

    path: /home/vagrant/grafana/

    state: directory

    owner: 472

    group: 472
```

11.4 Εγκατάσταση Προϊόντος TANDEM σε Υποδομή KUBERNETES

Τα δεδομένα ανά χείρας σε αυτή τη φάση είναι ένα Master-VM με τη χαρακτηριστική ονομασία «k8s-master» και πολλά Worker-Nodes «node-XX», όπου XX=1, 2, ...

Εκκινεί μία σύνδεση με το Master μέσω SSH πρωτόκολλου :

\$ vagrant ssh k8s-mastrer

Ο έλεγχος μεταφέρεται στο Master-VM και στη συνέχεια οποιαδήποτε ενέργεια θα επηρεάσει μόνο αυτό. Η πρώτη δράση περιλαμβάνει την εγκατάσταση του ενορχηστρωτή Kubernetes και συγκεκριμένα του φορέα «Master». Με άλλα λόγια σε αυτό το VM

εγκατασταθεί ο κεντρικός έλεγχος του ενορχηστρωτή. Για την αποφυγή λεπτομερειών που θα προβληματίσουν το διαχειριστή όλη η αλληλουχία κινήσεων έχει αποθηκευτεί σε ένα αρχείο και θα «ξετυλιχτεί» αυτόματα η βηματική προετοιμασία:

```
$ vagrant ssh k8s-mastrer
```

```
$ ./initialize-kubernetes-cluster.sh
```

Για εποπτικούς λόγους παρατίθεται το περιεχόμενο του παραπάνω Linux-script, ώστε ο αναγνώστης να αποκτήσει μία αίσθηση του «βάθους» των ενεργειών για την εγκατάσταση του Master:

```
#!/bin/sh

KUBE_VERSION=v1.24.2

echo "STEP 1: Define Environment Variables for Intracom Proxy"
export HTTP_PROXY=http://icache.intracomtel.com:80
export HTTPS_PROXY=http://icache.intracomtel.com:80
export NO_PROXY=192.168.0.0/16,10.96.0.0/16

# Forwarding IPv4 and letting iptables see bridged traffic
#echo "--> Is BR_NETFILTER Module Loaded ? "$(lsmod | br_netfilter)
#echo "--> Load Manually ..."$(sudo modprobe br_netfilter)

# Update SYSCTL config files ...
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter
```

```
# systemctl params required by steup, params persist across reboots

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

net.ipv4.ip_forward = 1

EOF

# Apply systctl params without reboot

sudo sysctl --system

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

sudo curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg \
  >/usr/share/keyrings/kubernetes-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] \
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee \
/etc/apt/sources.list.d/kubernetes.list

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] \
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

echo "STEP 2: Check Verison Number of Current Stable Release"

curl https://storage.googleapis.com/kubernetes-release/release/stable-1.txt -O
```

```
echo "\nLatest Stable Kubernetes Version is:"

cat stable-1.txt

echo "\nCurrent Install Version is: ${KUBE_VERSION}"

echo "\n"


echo "!!! Ubuntu Init System --> SYSTEMD !!!"

echo "... ContainerRuntime &*& KUBELET must use same CGROUP DRIVER ..."


echo "Container Runtime --> CONTAINERD"

#sudo systemctl status containerd.service --no-pager


sudo systemctl disable containerd.service


wget      https://github.com/containerd/containerd/releases/download/v1.6.4/containerd-1.6.4-linux-amd64.tar.gz

sudo tar Czxvf /usr/local containerd-1.6.4-linux-amd64.tar.gz


wget https://raw.githubusercontent.com/containerd/containerd/main/containerd.service

#sudo mv containerd.service /usr/lib/systemd/system/

sudo mv containerd.service /lib/systemd/system/


sudo systemctl daemon-reload

sudo systemctl enable --now containerd


sudo systemctl status containerd


sudo systemctl status docker.service --no-pager

sudo systemctl cat docker --no-pager
```

```
#ctr --help

ctr --version

#runc --help

runc --version


sudo mkdir -p /opt/cni/bin/

sudo curl -fsSL https://github.com/containernetworking/plugins/releases/download/v1.1.1/cni-plugins-linux-amd64-v1.1.1.tgz

sudo tar -xzf /opt/cni/bin/cni-plugins-linux-amd64-v1.1.1.tgz


echo "Configure the SYSTEMD CGROUP Driver\n-----"

sudo bash -c 'echo -e "[plugins.\"io.containerd.grpc.v1.cri\".containerd.runtimes.runc]\" >> /etc/containerd/config.toml'

sudo bash -c 'echo \"  version = 2\" >> /etc/containerd/config.toml'

sudo bash -c 'echo -e \"[plugins.\"io.containerd.grpc.v1.cri\".containerd.runtimes.runc.options]\" >> /etc/containerd/config.toml'

sudo bash -c 'echo \"  SystemdCgroup = true\" >> /etc/containerd/config.toml'


# Other way to append line into file

#echo \" | sudo tee -a /etc/containerd/config.toml


sudo cat /etc/containerd/config.toml


echo 'Apply changes into SYSTEMD service'

sudo systemctl restart containerd

sudo systemctl status containerd --no-pager
```

```
echo 'Make Sure that CRI is Not Included in DISABLED_PLUGINS'

sudo sed -i 's/"cri"/' /etc/containerd/config.toml

sudo systemctl restart containerd

sudo systemctl status containerd --no-pager


sudo cat /etc/containerd/config.toml


#cictl ps

#sudo critcl --debug pull nginx:latest


#git clone https://github.com/Mirantis/cri-dockerd.git


#cat /etc/cictl.yaml

#ls /var/run/dockershim.sock

#ls /var/run/docker.sock


#sudo kubeadm config images pull --cri-socket-path unix:///var/run/dockershim.sock


#kubeadm config images pull


printf '*****\nGO language install in Ubuntu
20\n*****\n'

#sudo apt update

#sudo apt upgrade -y


#sudo apt search golang-go

#sudo apt search gccgo-go
```

```
#sudo apt install golang-go -y

sudo curl -fsSLo go1.18.3.linux-amd64.tar.gz https://go.dev/dl/go1.18.3.linux-amd64.tar.gz

#sudo curl -fsSLo go1.13.linux-amd64.tar.gz https://go.dev/dl/go1.13.linux-amd64.tar.gz

sudo rm -rf /usr/local/go/

sudo tar -C /usr/local -xzf go1.18.3.linux-amd64.tar.gz

export PATH=$PATH:/usr/local/go/bin


# Verify Installation

go version

printf '\n'


echo '!!!! Setup GOPATH environment variable !!!'

sudo mkdir -p go/bin

sudo mkdir -p go/pkg

sudo mkdir -p go/src

sudo chmod -R a=rwx /home/vagrant/go/

# SOS

export GOPATH='/home/vagrant/go/'


printf '*****\nCRI      Interface      for      Docker
Engine\n*****\n'

sudo git clone https://github.com/Mirantis/cri-dockerd.git

cd cri-dockerd && ls


echo "***** go build *****"
```

```
go tidy

sudo chmod a=wrx -R /home/vagrant/cri-dockerd/

go build -o bin/cri-dockerd

sudo install -o root -g root -m 0755 bin/cri-dockerd /usr/local/bin/cri-dockerd

sudo cp -a packaging/systemd/* /etc/systemd/system

sudo sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service

sudo systemctl daemon-reload

sudo systemctl enable cri-docker.service

sudo systemctl enable --now cri-docker.socket


# CRI-Listens @

varListen=$(sudo systemctl status cri-docker.socket | grep Listen:)

echo "CRI-Docker-Socket ... ${varListen} VS '/run/cri-dockerd.sock'"


cat <<EOF | sudo tee /etc/crictl.yaml

runtime-endpoint: unix:///run/cri-dockerd.sock

image-endpoint: unix:///run/cri-dockerd.sock

EOF


# Pull Image(s)

echo 'Ready to Pull Busybox Image'

docker login -u vibm69 -p firewind


crictl pull busybox

crictl images
```

```
# List Containers

crictl -r /run/cri-dockerd.sock -i /run/cri-dockerd.sock ps

#echo "Get GODBUS Release ... v5.0.3"

#go get github.com/probonopd/go-appimage

#cd /home/vagrant/go/src/github.com/probonopd/go-appimage/

#go mod init

#go mod tidy

#grep godbus go.mod

#echo '---> Get dbus@v5 <---'

#go get github.com/godbus/dbus@v5

# Build Errors with Import Statements

# *****

#echo '!!! cannot find package "context" in any of:'

#sed -i 's/"context"/"golang.org/x/net/context"/' ~/go/src/github.com/Mirantis/cri-
dockerd/core/container_*.go

#sed -i 's/"context"/"golang.org/x/net/context"/' ~/go/src/github.com/Mirantis/cri-
dockerd/core/docker_service.go

#find ~/go/src/ -type f -exec sed -i 's/"context"/"golang.org/x/net/context"/' {} \;

echo "STEP 3: Initialize Kubernetes Cluster - Setup Control Plane"

#sudo kubeadm init --apiserver-advertise-address="192.168.50.10" --apiserver-cert-extra-
sans="192.168.50.10" --node-name k8s-master --pod-network-cidr=192.168.0.0/16 --service-
cidr="10.96.0.0/12" --kubernetes-version v1.23.1
```

```
#sudo kubeadm init --apiserver-advertise-address="192.168.50.10" --apiserver-cert-extra-sans="192.168.50.10" --node-name k8s-master --pod-network-cidr=10.244.0.0/16 --kubernetes-version v1.23.6

sudo kubeadm init --cri-socket unix:///run/cri-dockerd.sock --apiserver-advertise-address="192.168.56.10" --apiserver-cert-extra-sans="192.168.56.10" --node-name k8s-master --pod-network-cidr=10.244.0.0/16 --kubernetes-version ${KUBE_VERSION}

echo "STEP 4: Enable Vagrant User to Access Kubernetes Cluster"

sudo mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

echo "STEP 5: Setup Container Networking Provider and Policy Engine"

sudo curl https://docs.projectcalico.org/manifests/calico.yaml -o /home/vagrant/calico.yaml

kubectl apply -f /home/vagrant/calico.yaml

#kubectl          apply          -f          https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml

#echo "STEP 6: Kubectl Shows Ready ?"

#watch kubectl get nodes

echo "STEP 7: Create a bootstrap token for Worker-Nodes to Join"

sudo touch /home/vagrant/bootstrap-token-for-worker.sh

sudo chmod a=rwx /home/vagrant/bootstrap-token-for-worker.sh

kubeadm token create --print-join-command >> /home/vagrant/bootstrap-token-for-worker.sh
```

Το αποτέλεσμα από την «εκτέλεση» του script είναι η δημιουργία μίας βοηθητικής οδηγίας που χρησιμεύει στο ακόλουθο σενάριο:

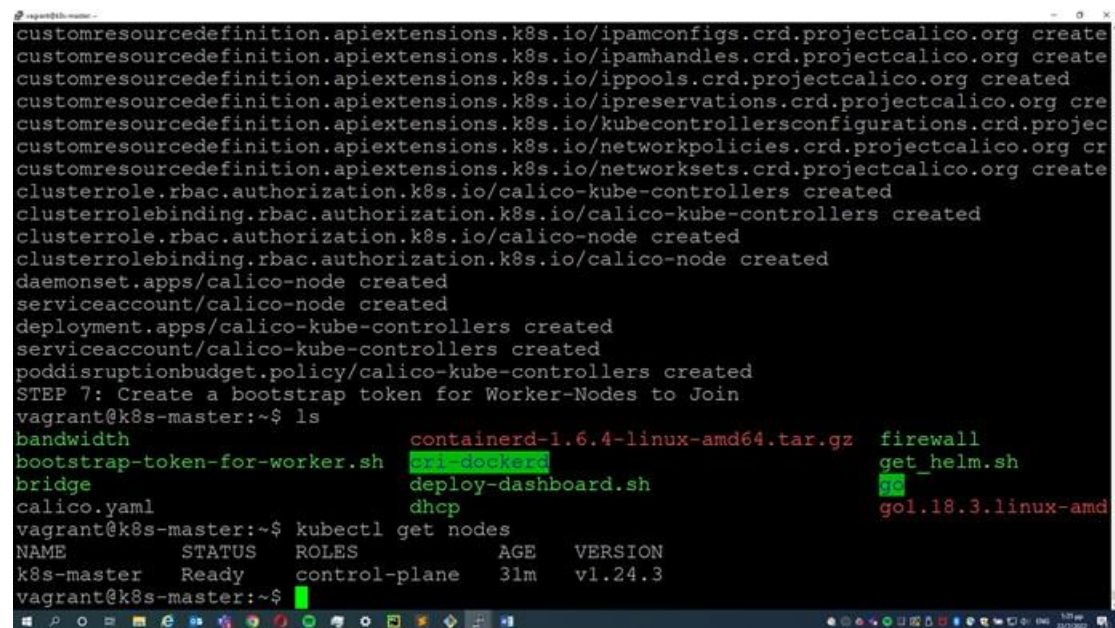
Worker-Node ... Handshake with ... Master Node

Συγκεκριμένα εμφανίζεται το αρχείο:

```
$ ls
```

[bootstrap-token-for-worker.sh](#)

```
$ kubectl get nodes
```



```
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org create
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org create
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org cre
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projec
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org cr
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org create
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
serviceaccount/calico-node created
deployment.apps/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
poddisruptionbudget.policy/calico-kube-controllers created
STEP 7: Create a bootstrap token for Worker-Nodes to Join
vagrant@k8s-master:~$ ls
bandwidth                               containerd-1.6.4-linux-amd64.tar.gz    firewall
bootstrap-token-for-worker.sh           cri-dockerd                           get_helm.sh
bridge                                  deploy-dashboard.sh                   go
calico.yaml                             dhcp                                  gol.18.3.linux-amd
vagrant@k8s-master:~$ kubectl get nodes
NAME             STATUS    ROLES    AGE     VERSION
k8s-master       Ready    control-plane   31m    v1.24.3
vagrant@k8s-master:~$
```

Στην παραπάνω εικόνα εμφανίζεται το αρχείο με τις οδηγίες ώστε ένα VM να «κερδίσει» την εμπιστοσύνη του Master και να συνδεθεί στο οικοσύστημα του Kubernetes σαν Worker. Επίσης ζητάτε από το σύστημα να κάνει μία απαρίθμηση των συνιστωσών του και εμφανίζει μόνο το Master που είναι λογικό καθώς δεν έχει γίνει κάποιο handshake.

11.5 Εισαγωγή Worker-Nodes στο Οικοσύστημα του Kubernetes

Το πρώτο βήμα περιλαμβάνει τη μετάβαση σε ένα VM που προορίζεται για worker:

```
$ vagrant ssh node-1
```

```
$ sudo ./join-worker-node.sh
```

Τα ενημερωτικά μηνύματα του worker-located script για join φαίνονται στην ακόλουθη εικόνα:

```

GLOBAL OPTIONS:
  --config value, -c value      Location of the client config file. If not spec
  --debug, -D                  Enable debug mode (default: false)
  --help, -h                    show help (default: false)
  --image-endpoint value, -i value Endpoint of CRI image manager service (default:
  --runtime-endpoint value, -r value Endpoint of CRI container runtime service (defa
  //var/run/cni-dockerd.sock]). Default is now deprecated and the endpoint should be se
  --timeout value, -t value     Timeout of connecting to the server in seconds
  --version, -v                 print the version (default: false)

FATA[0000] flag provided but not defined: -address
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

vagrant@node-1:~$

```

Η παραπάνω διαδικασία που εκτελέστηκε στο κόμβο #1 θα επαναληφθεί σε δύο ακόμα κόμβους. Το τελικό αποτέλεσμα είναι το Kubernetes οικοσύστημα να αποτελείται από ένα(1) Master και τρεις(3) Workers.

Για μία οπτική επιβεβαίωση αυτής της διαμόρφωσης συνδέομαι πάλι στο κεντρικό κόμβο και ζητώ μία λίστα από συμμετέχοντες στο οικοσύστημα. Τη τελευταία φορά η αναφορά περιλάμβανε μόνο τον Master, ενώ τώρα:

```

clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
serviceaccount/calico-node created
deployment.apps/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
poddisruptionbudget.policy/calico-kube-controllers created
STEP 7: Create a bootstrap token for Worker-Nodes to Join
vagrant@k8s-master:~$ ls
bandwidth          containerd-1.6.4-linux-amd64.tar.gz  firewall
bootstrap-token-for-worker.sh  cri-dockerd                          get_helm.sh
bridge             deploy-dashboard.sh                  gc
calico.yaml        dhcp                                gol.18.3.linux-amd
vagrant@k8s-master:~$ kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
k8s-master       Ready     control-plane   31m   v1.24.3
vagrant@k8s-master:~$ kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
k8s-master       Ready     control-plane   60m   v1.24.3
node-1           Ready     <none>        10m   v1.24.3
vagrant@k8s-master:~$ kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
k8s-master       Ready     control-plane   78m   v1.24.3
node-1           Ready     <none>        28m   v1.24.3
vagrant@k8s-master:~$

```

Μετά την παρέλευση μικρού χρονικού διαστήματος η δεξαμενή διαθεσίμων του Kubernetes έχει μεγαλώσει και άλλο:

```

bootstrap-token-for-worker.sh  cri-dockerd
bridge                         deploy-dashboard.sh
calico.yaml                    dhcp
vagrant@k8s-master:~$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
k8s-master    Ready     control-plane   31m    v1.24.3
vagrant@k8s-master:~$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
k8s-master    Ready     control-plane   60m    v1.24.3
node-1        Ready     <none>          10m    v1.24.3
vagrant@k8s-master:~$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
k8s-master    Ready     control-plane   78m    v1.24.3
node-1        Ready     <none>          28m    v1.24.3
vagrant@k8s-master:~$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
k8s-master    Ready     control-plane   83m    v1.24.3
node-1        Ready     <none>          33m    v1.24.3
node-2        Ready     <none>          19s    v1.24.3
vagrant@k8s-master:~$

```

Το οικοσύστημα έχει αρχίσει να αποκτάει κόμβους-«εργάτες» το οποίο σημαίνει ότι μπορεί να γίνει διασπορά εφαρμογών να δημιουργηθούν οι απαραίτητες οντότητες και να ξεκινήσει η ικανοποίηση αιτημάτων του τελικού χρήστη, για παράδειγμα αυτόματη αναγνώριση αντικειμένων με ευφυή μοντέλα.

11.6 Εγκατάσταση Βοηθητικών Υπηρεσιών στο Οικοσύστημα Kubernetes

Φτάνοντας σε αυτό το σημείο όλη η υποδομή υπολογιστικών-πόρων είναι διαθέσιμη τόσο σε επίπεδο φυσικών και εικονικών οντοτήτων. Έχει εγκατασταθεί το κατάλληλο λογισμικό με σημαντικότερη προσθήκη τον εντοπιστή Kubernetes. Μετά το γενικό διαχειριστικό πλαίσιο, έχει σειρά η ανάρτηση στο Edge των υπηρεσιών που προσφέρει η πλατφόρμα TANDEM για παράδειγμα Object-Detection.

Για την υλοποίηση των προσφερόμενων υπηρεσιών ακολουθείται η σχεδιαστική πρακτική των micro-services. Δηλαδή η ζητούμενη λειτουργικότητα κατακερματίζεται σε πολλά μικρά και καλά ορισμένα έργα που φιλοξενούνται σε containers που βρίσκονται σε διαρκή επικοινωνία για την ανταλλαγή μηνυμάτων και δεδομένων.

Γίνεται λοιπόν φανερό ότι πριν από την εγκατάσταση της TANDEM-υπηρεσίας θα προηγηθεί η εκκίνηση κάποιων βοηθητικών με κυριότερη ένα διαχειριστή μηνυμάτων (message-queue). Ο ρόλος του είναι κεντρικός στο σύστημα γιατί λαμβάνει μηνύματα από containers-παραγωγούς και τα προωθεί σε containers-καταναλωτές και με αυτό τον τρόπο ο έλεγχος προχωράει και οι υπηρεσίες οδεύουν από τα αρχικά στάδια στα καταληκτικά.

Ο έλεγχος μεταφέρεται στο Master-κόμβο και ασχολείται με τη δημιουργία ενός namespace ώστε να υπάρχει ένας ιδιωτικός χώρος για τις υπηρεσίες του TANDEM:

```
$ vagrant ssh k8s-master
```

```
$ ./tandem/create-ns-tandem.sh
```

```

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
Last login: Fri Jul 22 09:34:42 2022 from 10.0.2.2
vagrant@k8s-master:~$ tandem
tandem
vagrant@k8s-master:~$ tandem/create-ns-tandem.sh
namespace/tandem created
NAME          STATUS    AGE
default       Active   154m
kube-node-lease  Active   154m
kube-public    Active   154m
kube-system    Active   154m
tandem         Active    0s
vagrant@k8s-master:~$

```

Μία ενδιαφέρουσα παρατήρηση έχει να κάνει με την ύπαρξη έτοιμων scripts που διευκολύνουν την εγκατάσταση κάθε υπηρεσίας του TANDEM αλλά και κάθε βοηθητικής απομακρύνοντας από το διαχειριστή την υποχρέωση να ασχοληθεί με λεπτομέρειες. Όλα αυτά τα αρχεία έχουν συγκεντρωθεί στον υποκατάλογο TANDEM.

Έστω ότι θα χρησιμοποιηθεί είτε το Redis είτε το RabbitMQ σαν message-queue με το πρώτο να είναι απλούστερο και πιο ελαφρύ.

```

kube-system      Active   154m
tandem           Active    0s
vagrant@k8s-master:~$ ls tandem/
apply-persistent-volume-local.sh
create-ns-tandem.sh
create-registry-push-tandem-images.sh
create-registry-svc.sh
delete-tandem-pods.sh
deploy-grafana-k8s.sh
deployments
deploy-private-repository.sh
deploy-prometheus-stack-svc.sh
deploy-redis-k8s.sh
download-new-images-run-again-pods.sh
download-new-watchdog-image.sh
grafana-manifest.yaml
IoT
persistent-volume-local-manifest.yaml
prometheus-stack-manifest.yaml
redis-official-full.yaml
run-grabber-pod.sh
run-system-monitoring.sh
run-tandem-pods.sh
run-watchdog-pod.sh
send-HTTP-requests-stack.sh
vagrant@k8s-master:~$

```

Ανάμεσα στα άλλα προγράμματα-διανομής υπάρχει το «deploy-redis-k8s.sh» που με τη σειρά του καλεί το εξαιρετικά σημαντικό αρχείο «redis-official-full.yaml»:

```

---
kind: PersistentVolume

```

```
apiVersion: v1
metadata:
  name: redis-pv
spec:
  storageClassName: manual
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/home/vagrant/redis-store"
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: redis-pvc
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
labels:
  app: redis
  name: redis
spec:
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
        - name: redis
          image: bitnami/redis:7.0
          imagePullPolicy: IfNotPresent
          env:
            - name: REDIS_PORT_NUMBER
              value: '7000'
            - name: ALLOW_EMPTY_PASSWORD
              value: 'yes'
          ports:
            - containerPort: 7000
              name: "tcp-redis"
              protocol: TCP
          # readinessProbe:
          #   failureThreshold: 3
```

```
# httpGet:

#   path: /robots.txt

#   port: 3000

#   scheme: HTTP

#   initialDelaySeconds: 30

#   periodSeconds: 30

#   successThreshold: 1

#   timeoutSeconds: 2

# livenessProbe:

#   failureThreshold: 3

#   initialDelaySeconds: 30

#   periodSeconds: 10

#   successThreshold: 1

# tcpSocket:

#   port: 3000

#   timeoutSeconds: 1

resources:

  requests:

    cpu: 250m

    memory: 750Mi

  volumeMounts:

    - mountPath: /var/lib/redis

      name: redis-pv

securityContext:

  fsGroup: 472

supplementalGroups:

- 0
```

```
volumes:
  - name: redis-pv
    persistentVolumeClaim:
      claimName: redis-pvc
---
apiVersion: v1
kind: Service
metadata:
  name: redis-svc
spec:
  ports:
    - port: 7000
      protocol: TCP
      targetPort: "tcp-redis"
      nodePort: 30000
  selector:
    app: redis
  sessionAffinity: None
```