ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ροή Δικτύων Υπολογιστών και Επικοινωνιών,

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ασφαλής διαχείριση IoT συσκευών

Γεώργιος Τσερεγκούνης
Α.Μ. 151056

Εισηγητής: Παναγιώτης Καρκαζής, Επ. Καθηγητής

(Κενό φύλλο)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ασφαλής διαχείριση IoT συσκευών

Γεώργιος Τσερεγκούνης
Α.Μ. 151056

Εξεταστική Επιτροπή:

Παναγιώτης Καρκαζής
Επ. Καθηγητής

Panagiotis Karkazis
Digitally signed by Panagiotis Karkazis
Date: 2021.07.28 16:08:16 +03'00'

Αθανάσιος Βουλόδημος
Επ. Καθηγητής

Athanasios Voulodimos
Digitally signed by Athanasios Voulodimos
Date: 2021.07.28 14:20:51 +03'00'

Ελένη – Αικατερίνη Λελίγκου
Αναπληρώτρια Καθηγήτρια

Eleni Aikaterini Leligkou
Digitally signed by Eleni Aikaterini Leligkou
Date: 2021.07.28 15:38:38 +03'00'

Ημερομηνία εξέτασης 26/7/2021

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Τσερεγκούνης Γεώργιος του Αλεξάνδρου, με αριθμό μητρώου 151056 φοιτητής/τρια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:
«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

(Κενό φύλλο)

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της διαχείρισης έξυπνων συσκευών. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

(Κενό φύλλο)

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει σαν σκοπό την καταγραφή των πρωτοκόλλων και της αρχιτεκτονικής του διαδικτύου των πραγμάτων (IoT). Το IoT αποτελείται από πολλά και διαφορετικά αντικείμενα με ενσωματωμένους αισθητήτες και ενεργοποιητές, τα οποία είναι συνδεδεμένα στο διαδίκτυο. Στο πλαίσιο της παρούσης εργασίας αναλύονται τα ζητήματα ασφάλειας που προκύπτουν από αυτή την τεχνολογία. Ακόμα καταγράφονται διάφορες διαθέσιμες πλατφόρμες για την διαχείριση IoT συσκευών και παρουσιάζεται σε αναλυτικά βήματα η ανάπτυξη και εφαρμογή μιας πλατφόρμας για την διαχείριση συσκευών του διαδικτύου των πραγμάτων.

ABSTRACT

The present thesis aims to record the protocols and architecture of the Internet of Things (IoT). The IoT consists of many different networks of objects, which are connected to the Internet and each object incorporates sensors. The security issues arising from this technology are analyzed. We are also preset several available platforms for managing IoT devices and we propose and deploy an opensouse platform for managing the Internet of Things.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Επικοινωνίες και Δίκτυα Ηλεκτρονικών Υπολογιστών
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Διαδίκτυο των Πραγμάτων, Ασφάλεια, Υπολογιστικό Νέφος,

Περιεχόμενα

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

IoT Internet of Things

ITU International Telecommunication Union

BMA Block Matching Algorithm

MTU Maximum transmission unit

IEEE Institute of Electrical and Electronics Engineers

EPC Electronic Product Code

RFID Radio-frequency identification

GSM Global System for Mobile Communications

OMG Object Management Group

M2M machine-to-machine

DDS Data Distribution Service

QoS Quality of service

DTLS Datagram Transport Layer Security

AMQP Advanced Message Queuing Protocol

MQTT Message Queue Telemetry Transport

MQTT-SN MQTT For Sensor Networks

XMPP eXtensible Messaging and Presence Protocol

REST Representational State Transfer

IETF Internet Engineering Task Force

DTLS Datagram Transport Layer Security

mDNS multicast Domain Name System

DNS-SD DNS Service Discovery

IID Interface Identifier

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks

MTU Maximum transmission unit

RPL Routing protocol for low power and lossy networks

WPANs wireless personal area networks

PLC low-power line communication

WSNs wireless sensor networks

LR-WPANs Low-Rate Wireless Personal Area Networks

FFD Fully functional devices

RFD Reduced functional devices

PAN Personal area network

LTE-A Long Term Evolution-Advanced

eMBMS evolved multimedia broadcast multicast service

EPCISEPCglobal Electronic Product Code Information Services

ONS object-naming services

CIA Confidentiality, Integrity, Availability

AAA Authentication, Authorization, Accounting

TCB Trusted Computing Base

ACL Access Control List

RSA Rivest-Shamir-Adleman

DES Data Encryption Standard

AES Advanced Encryption Standard

DoS Denial of Service

MITM Man-in-the-middle attack

GMM Gateway Management Module

Εισαγωγή

Ιστορική αναφορά και ορισμός IoT συσκευών

Η ραγδαία ανάπτυξη του διαδικτύου τα τελευταία χρόνια έχει οδηγήσει στην εισβολή του στην καθημερινότητα του ανθρώπου. Το Internet και οι εφαρμογές του διευκολύνουν τον άνθρωπο ειδικά σε δύσκολες για αυτόν καταστάσεις. Μια αρκετά ενδιαφέρουσα τεχνολογική εξέλιξη είναι και το Διαδίκτυο των Πραγμάτων (IoT). Η προσπάθεια των επιστημόνων για την επεξήγηση του ορισμού του IoT έχει ξεκινήσει από το 1994 όπου ο R.S. Raji σε άρθρο του στο επιστημονικό περιοδικό IEEE (Institute of Electrical and Electronics Engineers) Spectrum αναφέρεται στο IoT με την φράση "μεταφορά μικρών πακέτων δεδομένων σε ένα μεγάλο εύρος κόμβων, με στόχο την πλήρη ενσωμάτωση και αυτοματοποίηση διεργασιών από οικιακές εφαρμογές μέχρι και σε βιομηχανικές" [1]. Στις αρχές της χιλιετίας του 2000 η εταιρεία Auto-ID ξεκίνησε την προώθηση της ιδέας της διασύνδεσης των αντικειμένων, τα οποία ενσωμάτωναν το σύστημα EPC (Electronic Product Code), με το διαδίκτυο. Ο εκτελεστικός διευθυντής της εταιρείας (Kevin Ashton [2]) θεωρήθηκε ως ο πρώτος που χρησιμοποίησε τον όρο Internet of Things όπου το 1999 σε μια παρουσίασή του αναφέρθηκε στην προσπάθεια των εταιρειών να ελαχιστοποιήσουν την ανθρώπινη παρέμβαση στην εφοδιαστική αλυσίδα μέσω ενός τρόπου διασύνδεσης των αντικειμένων με το διαδίκτυο, όπου χρησιμοποιούταν και το πρότυπο RFID (Radio-frequency identification). Όμως σύμφωνα με τον καθηγητή Daniel Engel ο όρος IoT εμφανίζεται το 1997 σε μια δημοσίευση της Διεθνούς Ένωσης Τηλεπικοινωνιών - ITU (International Telecommunication Union) [3].

Για τον ακριβή ορισμό του IoT βλέπουμε ότι υπάρχουν διαφωνίες που επίκεινται στην οπτική γωνία του κάθε οργανισμού. Από την μια μεριά η αμερικάνικη εταιρεία πληροφορικής, έρευνας και συμβουλών Gartner δίνει τον ακόλουθο ορισμό: "The Internet of Things (IoT) is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment". Με άλλα λόγια θέτει το IoT ως το δίκτυο όπου τα φυσικά αντικείμενα εντός του εμπεριέχουν ενσωματωμένα συστήματα επικοινωνίας, αλληλεπίδρασης και αίσθησης τόσο με την εσωτερική τους κατάστασης όσο και με το εξωτερικό τους περιβάλλον. Ενώ ο IEEE θέτει το IoT [4] ως: "A network of items—each embedded with sensors—which are connected to the Internet". Δηλαδή ορίζει την περιγραφή του IoT ως ένα δίκτυο πραγμάτων, το καθένα ενσωματώνει αισθητήρες, τα οποία είναι συνδεδεμένα στο διαδίκτυο.

Η πρώτη προσπάθεια του ανθρώπου για την δημιουργία μιας πρώιμης μορφής IoT εφαρμογής αποτελεί το "έξυπνο" μηχάνημα αυτόματης πώλησης Coca-cola το 1982 στο πανεπιστήμιο Carnegie Mellon. Ορισμένοι φοιτητές προσάρμοσαν τον αυτόματο πωλητή έτσι ώστε να έχει την δυνατότητα αναφοράς του περιεχομένου του και την κατάσταση των προϊόντων (ζεστά η κρύα) [5]. Το Trojan Room coffee pot [6] αποτελεί ένα επιπλέον πρώιμο IoT project. Η εφαρμογή αυτή υλοποιήθηκε το 1989 και προέκυψε από την 'ανάγκη' του ανθρώπου για την 'καφεΐνη'. Οι ερευνητές του πανεπιστήμιου του Cambridge είχαν τοποθετήσει μια κάμερα που μετέδιδε την κατάσταση της καφετερίας (3 φορές ανά ένα λεπτό).

Συμπερασματικά έχουμε την δυνατότητα να περιγράψουμε το Διαδίκτυο των Πραγμάτων (Internet of Things) ως την δικτύωση διαφορετικών φυσικών αντικειμένων όπως είναι οι συσκευές και τα αντικείμενα που χρησιμοποιεί καθημερινά ο άνθρωπος. Τέτοιες συσκευές μπορεί να είναι μια τηλεόραση, ένα κλιματιστικό καθώς και οποιοδήποτε άλλο αντικείμενο όπως ένα αυτοκίνητο ή κτιριακές εγκαταστάσεις. Όλα τα μέρη του IoT αποτελούνται από

αισθητήρες που καταγράφουν οποιαδήποτε μεταβολή. Κάθε καταγραφή μιας μεταβολής αποτελεί και ένα δεδομένο το οποίο μετά από επεξεργασία από την ίδια την IoT συσκευή αποστέλνεται σε μια γειτονική. Η σύνθεση αυτών των δεδομένων γεννά την πληροφορία που επιθυμούμε να αντλήσουμε από το συγκεκριμένο δίκτυο. Ακόμα μέσω συγκεκριμένων πλατφορμών έχουμε την δυνατότητα να ρυθμίσουμε και να μεταβάλουμε την λειτουργία των συσκευών απομακρυσμένα.

Έξυπνες πόλεις (Smart Cities)

Πλέον η ανάπτυξη της επιστήμης της πληροφορικής έχει καταστήσει εφικτή την μετατροπή των πόλεων σε «έξυπνες». Σε μια τέτοια πόλη όλα τα αντικείμενα δύναται να έχουν έναν ενσωματωμένο επεξεργαστή, αισθητήρες και την δυνατότητα να επικοινωνούν μεταξύ τους. Ακόμα τα δεδομένα που συλλέγουν οι αισθητήρες αποστέλλονται σε εφαρμογές που τρέχουν στο νέφος, κατόπιν τα δεδομένα αναλύονται – διαχειρίζονται και ανατροφοδοτούν τους πολίτες με πληροφορίες που ενισχύσουν την ποιότητα της ζωής τους. Κατά αυτόν τον τρόπο το περιβάλλον μιας πόλης θα γίνει ασφαλέστερο και πιο πρακτικό [7].

Σιγκαπούρη

Η Σιγκαπούρη [8] είναι μια από τις πιο “έξυπνες πόλεις” διεθνώς, το πρόγραμμα που έχει εφαρμοστεί από την κυβέρνηση ονομάζεται Smart Nation (Έξυπνο Έθνος). Με δημόσια δαπάνη έχουν τοποθετηθεί μέσα στην πόλη διάφορα κιβώτια (Aggregation Gateway boxes) στα οποία ενσωματώνονται ποικίλοι αισθητήρες. Τα δεδομένα που συλλέγονται αφορούν την κυκλοφορία των οχημάτων και των πεζών, τα δεδομένα αυτά αποστέλλονται σε διάφορες υπηρεσίες με σκοπό την βελτίωση των παρεχόμενων υπηρεσιών τους και συνεπώς μέσα από υπηρεσίες ανοικτού κώδικα η πληροφορία διαχέεται στους πολίτες. Επειδή περίπου το 80% των κατοίκων της Σιγκαπούρης διαμένει σε δημόσιες κατοικίες, η κυβέρνηση έχει την δυνατότητα να πειραματίζεται με διάφορες οικιακές υπηρεσίες αυτοματοποίησης όπως την έξυπνη διαχείριση στην οικιακή ενέργεια, το νερό καθώς και τον έλεγχο συστημάτων για τους ηλικιωμένους. Ακόμα μηχανικοί αναλύουν δεδομένα που αφορούν την ροή του αέρα και της ηλιακής ενέργειας προκειμένου να κατασκευαστούν κτίρια σε καταλληλότερες περιοχές. Μέχρι το 2022 υπολογίζεται ότι θα έχει επιτευχθεί η εγκατάσταση έξυπνων λαμπών που σθντελούν στην εξοικονόμηση ενέργειας σε όλους τους δημόσιους δρόμους καθώς και ότι πάνω από 6.000 κτίρια θα έχουν ηλιακούς συλλέκτες. Επιπλέον το Εθνικό Ίδρυμα Έρευνας (National Research Foundation) έχει φτιάξει ένα τρισδιάστατο μοντέλο της πόλης κι το διαθέτει στο ευρύ κοινό για δοκιμές ιδεών και υπηρεσιών.

Όσλο

Η πόλη του Όσλο αναφέρεται διεθνώς ως πρότυπο ‘έξυπνης πόλης’ ειδικά για την έξυπνη και αποδοτικότερη διαχείριση της ενέργειας. Οι κτιριακές εγκαταστάσεις παγκοσμίως υπολογίζεται ότι καταναλώνουν περίπου το 40% της ενέργειας. Προκείμενου να μειωθεί αυτό το ποσοστό στην πρωτεύουσα της Νορβηγίας η κυβέρνηση έχει τοποθετήσει μια ευρεία γκάμα από αισθητήρες που ελέγχουν τον φωτισμό, την θέρμανση και την ψύξη της πόλης. Στόχος του Όσλο είναι να ελαττώσει τις “δαπάνες” σε ενέργεια ως 95% μέχρι το 2030 [9]. Για να επιτευχθεί ο παραπάνω στόχος οι αρχές της πόλης δημιουργούν ευκαιρίες για την ευρεία χρήση των ηλεκτρικών αυτοκινήτων, των έξυπνων δικτύων (smart grids) καθώς και των σταθμών φόρτισης ηλεκτρικών οχημάτων. Μέχρι στιγμής στο Όσλο υπάρχουν περισσότεροι από 2,000 σταθμοί φόρτισης ηλεκτρικών οχημάτων, οι κάτοχοι αυτών των

οχημάτων απολαμβάνουν μια σειρά από προνόμια όπως δωρεάν parking, φόρτιση, απαλλαγή από μεταφορικά σε πλοία καθώς κι μερική ελάφρυνση από φόρους πώλησης. Επιπροσθέτως η Νορβηγική κυβέρνηση έχει ανακοινώσει πως σχεδιάζει να χτίσει μια έξυπνη πόλη έκτασης σχεδόν ενός τετραφωνικού χιλιομέτρου γύρω από το αεροδρόμιου του Όσλο έτσι ώστε να δημιουργηθεί μια κοινωνία που θα βασίζεται στην τεχνολογία [10]. Ο σχεδιασμός είναι αυτή η πόλη να ηλεκτροδοτείται μόνο από ενέργεια που θα προέρχεται από ανανεώσιμες πηγές καθώς και όποια «αχρησιμοποίητη» μορφή ενέργειας να επιστρέφει πίσω στο δίκτυο. Σε αυτό το πλαίσιο λοιπόν συστήματα με ενσωματωμένους αισθητήρες θα διαχειρίζονται τόσο τον φωτισμό στον οδικό άξονα, στα κτίρια όσο και την επεξεργασία των αποβλήτων. Σε αυτήν την μικρή αλλά έξυπνη πόλη θα επιτρέπεται η κυκλοφορία μόνο ηλεκτρονικών αυτοκινήτων και στην συνεχεία μόνο αυτόματων αυτοκινήτων (self-driving).

Κοπεγχάγη

Η Κοπεγχάγη οδεύει ολοταχώς προς την ενσωμάτωση έξυπνων εφαρμογών στο δίκτυο και επακολούθως στην δραστική μείωση του διοξειδίου του άνθρακα στην πόλη μέχρι το 2025 [11]. Ήδη το 2014 το κρατικό ίδρυμα Copenhagen Solutions Lab έλαβε το βραβείο Έξυπνες Πόλεις Διεθνώς (World Smart Cities award) για την ανάπτυξη συστήματος (Copenhagen Connecting) [12] ελέγχου της οδικής κυκλοφορίας, της ποιότητας του αέρα και έξυπνης διαχείρισης των αστικών αποβλήτων, ενέργειας. Το σύστημα αυτό συνδέει τις υπηρεσίες στάθμευσης οχημάτων, τους σηματοδότες, την φωταγώγηση της πόλης, έξυπνης καταμέτρησης και τα σημεία φόρτισης των ηλεκτρονικών οχημάτων σε πραγματικό χρόνο και βελτιστοποιεί την διαχείριση της ενέργειας ανάλογα με την τιμή των υγρών καυσίμων, την κυκλοφοριακή συμφόρηση και τις καιρικές συνθήκες. Ο στόχος της ανάλυσης, μέτρησης και σύγκρισης όλων αυτών των δεδομένων είναι η αποτελεσματικότερη προσφορά υπηρεσιών στους πολίτες της πόλης. Αρκετοί κάτοικοι της Κοπεγχάγης χρησιμοποιούν το ποδήλατο ως βασικό μέσο μεταφοράς [13] για αυτό το λόγο έχει δημιουργηθεί μια εφαρμογή [14] όπου καθοδηγεί τους ποδηλάτες στην πόλη. Ακόμα η εφαρμογή τους ενημερώνει πόσο γρήγορα χρειάζεται να πηγαίνουν για να προλάβουν το επόμενο πράσινο φανάρι, μετράει την διανυόμενη απόσταση και υπολογίζει τις θερμίδες όπου έχουν καταναλώσει.

Οι εφαρμογές των έξυπνων πόλεων συνοψίζονται στις ακόλουθες κύριες κατηγόριες:

Έξυπνη διαχείριση ενέργειας

Στις μέρες μας έχει γίνει σαφές σε όλους ότι η βελτιστοποίηση της διαχείρισης της ενέργειας είναι απαραίτητη ειδάλλως η οικολογική κρίση σύντομα θα ξεφύγει από κάθε έλεγχο. Για αυτό το λόγο η υιοθέτηση IoT εφαρμογών για την διαχείριση της ενέργειας ειδικά σε αστικές περιοχές αποτελεί ένα πραγματικά χρήσιμο εργαλείο για την προστασία του πλανήτη. Η έξυπνη καταμέτρηση κατανάλωσης της ενέργειας και στην συνέχεια η ανάλυση των μετρήσεων αυτών σε συνδυασμό με την παροχή συμβουλών μείωσης της κατανάλωσης της ενέργειας μπορεί να βοηθήσει τους πολίτες να υιοθετήσουν πιο οικολογικούς τρόπους ζωής.

Επιπλέον βασική ιδέα [15] της έξυπνης διαχείρισης της ενέργειας είναι η ενσωμάτωση της σε όλες τις έξυπνες εφαρμογές μιας και με αυτό τον τρόπο η διαχείριση της ενέργειας θα είναι αποδοτικότερη.

Έξυπνο σύστημα υγείας

Την τελευταία διετία και με την πανδημία που προκάλεσε ο SARS-CoV-2 οι συζητήσεις για βελτιστοποίηση των συστημάτων υγείας έχουν αυξηθεί. Μια πιθανή λύση είναι και η χρήση εφαρμογών IoT όπου μπορούν να παρέχουν ακριβή, συνεχή ενημέρωση στον γιατρό για κάθε ασθενή και σε περίπτωση εμφάνισης μιας μη αναμενόμενης μέτρησης να γίνει αποστολή ειδοποίησης ανάλογα με την κρισιμότητα. Ακόμα σε περίπτωση άμεσης ανάγκης υπάρχει η δυνατότητα ειδοποίησης συγγενών.

Κατά την διάρκεια της πανδημίας [16] αρκετά κράτη είχαν προχωρήσει σε μια παρόμοια λύση όπου ο κάθε πολίτης με την λήψη, εγκατάσταση μιας εφαρμογής μπορούσε να ενημερωθεί σε περίπτωση που βρισκόταν σε κοντινή απόσταση με κάποιο κρούσμα της νόσου. Η εφαρμογή με την χρήση του πρωτοκόλλου bluetooth επικοινωνεί με άλλες συσκευές εντός της εμβέλειας του δικτύου και ανταλλάσσουν χρίσμα δεδομένα, εφόσον κάποιος χρήστης επιβεβαιώσει ότι υπήρξε κρούσμα τότε αποστέλλεται σχετική ειδοποίηση σε όλους όσους έχει γίνει μια ζεύξη εντός ενός σύντομου χρονικού διαστήματος.

Έξυπνο σπίτι

Οι εφαρμογές του IoT εντός του σπιτιού γνωρίζουν μια αυξητική πορεία τα τελευταία χρόνια μιας και μας βοηθούν στην καθημερινότητά μας. Σχεδόν όλες οι οικιακές συσκευές σήμερα μπορούν να έχουν ενσωματωμένους αισθητήρες και ελεγκτές (controllers). Κάθε συσκευή γίνεται πλέον κόμβος του δικτύου και μπορεί να ελέγχεται από μια άλλη συσκευή. Κατά αυτόν τον τρόπο έχουμε την δυνατότητα να εφαρμόσουμε διάφορες υλοποιήσεις όπως την ‘αυτόματη’ ενεργοποίηση του συναγερμού μόλις απομακρυνθούμε από το σπίτι, την διαχείριση του φωτισμού, την αυτόματη καθαριότητα μέχρι και την παραγγελία αγαθών που λείπουν από το ψυγείο.

Έξυπνη διαχείριση κυκλοφορίας

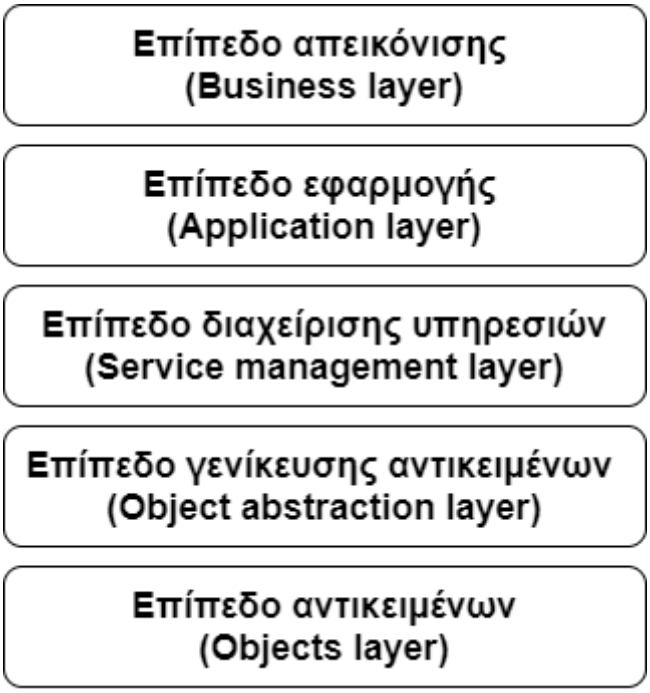
Η κυκλοφοριακή συμφόρηση στις πόλεις αποτελεί ένα τεράστιο πρόβλημα διεθνώς. Σημαντικά προβλήματα στον οδικό ιστό μιας μητρόπολης μπορούν να προκύψουν από ποικίλους παράγοντες. Οι πιο συνήθεις αιτίες για αυτά τα προβλήματα είναι υπερβολικά πολλά οχήματα στον δρόμο, ένα ατύχημα, αποτυχία των φωτεινών σηματοδοτών, ένα χαλασμένο όχημα καθώς και προβλήματα στο οδόστρωμα.

Με την εφαρμογή IoT λύσεων η εμφάνιση του προβλήματος της κυκλοφοριακής συμφόρησης στις πόλεις μπορεί να μειωθεί δραματικά. Ορισμένες λύσεις είναι η ανάλυση δεδομένων [17] από βίντεο, διάφορους αισθητήρες (υπέρυθρης ακτινοβολίας, RFID, ανίχνευσης τοποθεσίας, παρόντος καιρού, ορατότητας, ατμοσφαιρικού ηλεκτρικού πεδίου, υγρασίας), κατάσταση των φωτεινών σηματοδοτών. Μετά την ανάλυση των δεδομένων οι χρήστες του οδικού άξονα έχουν την δυνατότητα μέσω μιας εφαρμογής στο κινητό τους τηλέφωνο για συνεχή ενημέρωση της ‘εικόνας’ της διαδρομής τους μέχρι τον τελικό προορισμό. Η εφαρμογή αυτή ενδέχεται να προτείνει αλλαγές στην διαδρομή, να αποστέλλει ειδοποιήσεις στον χρήστη για καιρικά φαινόμενα που επηρεάζουν την οδήγηση. Ακόμα ένα έξυπνο σύστημα διαχείρισης της κυκλοφορίας ενδέχεται να αποτελείται από έξυπνα οχήματα [18] που θα αντιλαμβάνονται το περιβάλλον τους και θα περιηγούνται στον αστικό ιστό μόνα τους χωρίς να απαιτείται οποιαδήποτε παρέμβαση από άνθρωπο. Όλα τα προηγούμενα μέτρα μπορούν να οδηγήσουν στην αποφυγή σημείων συμφόρησης, δυστυχημάτων καθώς και σε πιο ξεκούραστη «εμπειρία» οδήγησης γεγονός που αυξήσει την ποιότητα ζωής των πολιτών μιας πόλης.

Αρχιτεκτονική και Τρόποι διαχείρισης IoT συσκευών

Η αρχιτεκτονική του IoT που πρόκειται να μελετήσουμε αποτελείται από πέντε επίπεδα (αναπαρίσταται στο σχήμα 1), τα οποία είναι:

- Επίπεδο αντικειμένων (Objects layer)
- Επίπεδο γενίκευσης αντικειμένων (Object abstraction layer)
- Επίπεδο διαχείρισης υπηρεσιών (Service management layer)
- Επίπεδο εφαρμογής (Application layer)
- Επίπεδο απεικόνισης (Business layer)



Σχήμα 1 Αρχιτεκτονική 5 επιπέδων

Επίπεδο αντικειμένων (Objects layer)

Το επίπεδο αντικειμένων συγκεντρώνει ουσιαστικά όλες τις φυσικές συσκευές που συλλέγουν δεδομένα από το περιβάλλον τους και στην συνέχεια συνθέτουν την πληροφορία που αξιοποιεί το IoT σύστημα. Οι φυσικές συσκευές αποτελούνται από διάφορους μικροελεγκτές όπως είναι το raspberry και το arduino και μπορούν να ενθυλακώνουν μια ευρεία γκάμα από αισθητήρες μέτρησης υγρασίας, θερμοκρασίας, θορύβου αλλά και πολλά άλλα στοιχεία του περιβάλλοντος. Τα δεδομένα που συλλέγονται χρειάζεται να μεταβούν στο επόμενο επίπεδο μέσα από τα ασφαλή κανάλια επικοινωνίας.

Επίπεδο αφαιρετικής διαχείρισης αντικειμένων (Object abstraction layer)

Το συγκεκριμένο επίπεδο αναλαμβάνει την μετάδοση της πληροφορίας από το επίπεδο αντικειμένων στο επίπεδο διαχείρισης υπηρεσιών. Η μετάδοση των μηνυμάτων μπορεί να γίνει μέσα από το cloud ή πιο παραδοσιακές τεχνολογίες όπως οι: 3G, GSM (Global System for Mobile Communications), Wi-Fi, Bluetooth low energy και ZigBee.

Επίπεδο διαχείρισης υπηρεσιών (Service management layer)

Το επίπεδο διαχείρισης υπηρεσιών μπορούμε να το παρομοιάσουμε με ένα router μιας και είναι υπεύθυνο για την αντιστοίχιση των διαφορών υπηρεσιών με τον αιτούντα της υπηρεσίας βασιζόμενο στην ονομασία και τις διευθύνσεις των δυο μερών. Με αυτήν την διαδικασία οι σχεδιαστές του πληροφοριακού συστήματος έχουν την δυνατότητα να «εξαλείψουν» τις διαφορές ανάμεσα στα δεδομένα που προέρχονται από ετερογενείς πήγες.

Επίπεδο εφαρμογής (Application layer)

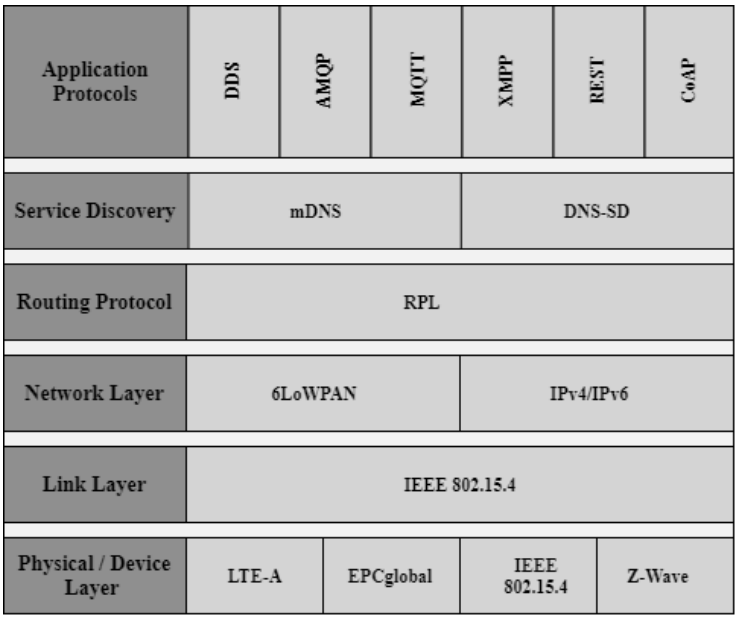
Το επίπεδο εφαρμογής μεταβιβάζει στον «καταναλωτή» την προσφερόμενη υπηρεσία του IoT περιβάλλοντος. Τα διάφορα IoT περιβάλλοντα περιλαμβάνουν τις «έξυπνες» πόλεις, την «έξυπνη» διαχείριση της ενέργειας, το έξυπνο σπίτι καθώς και πολλές άλλες εφαρμογές.

Επίπεδο απεικόνισης (Business layer)

Σε αυτό το επίπεδο συναντάται η συνολική εποπτεία των υπηρεσιών και των διαδικασιών εντός του IoT περιβάλλοντος. Αυτό το επίπεδο είναι υπεύθυνο για την σχεδίαση, ανάλυση, αξιολόγηση, έλεγχο και εφαρμογή των προϋποθέσεων του IoT συστήματος. Ακόμα σε αυτό το επίπεδο γίνεται η απεικόνιση των δεδομένων σε σχεδιαγράμματα, μοντέλα και διάφορα διαγράμματα.

Πρωτόκολλα IoT

Τα πρωτόκολλα στο IoT χωρίζονται στις κατηγορίες: πρωτοκολλά επιπέδου εφαρμογής, υπηρεσίας ανακάλυψης, πρωτόκολλα δρομολόγησης, πρωτόκολλα επιπέδου δικτύου, πρωτόκολλα επιπέδου ζεύξης καθώς και πρωτόκολλα φυσικού επιπέδου. Όλα τα πρωτόκολλα που αναφέρονται στο σχήμα 2 ενορχηστρώνουν όλα τα μέρη ενός IoT συτήματος.



Σχήμα 2 Κατηγοριοποίηση IoT Πρωτοκόλλων [19]

Πρωτόκολλα επιπέδου εφαρμογής

Κατηγορίες πρωτοκόλλων (Request / Response & Publish / Subscribe)

Τα πρωτόκολλα επικοινωνίας του επιπέδου εφαρμογής χωρίζονται κυρίως σε δυο κατηγορίες. Αυτές είναι το μοντέλο αιτήματος / απάντησης (Request / Response) και το μοντέλο δημοσίευσης / εγγραφής (Publish / Subscribe).

Το μοντέλο αιτήματος / απάντησης (Request / Response) αποτελεί τον πιο τυπικό τρόπο επικοινωνίας μεταξύ δυο συσκευών. Η συσκευή-πελάτης στέλνει στον εξυπηρετητή (server) ένα αίτημα και αυτός εκτελεί ορισμένες διεργασίες για την ικανοποίηση του αιτήματος και επιστρέφει στον πελάτη απάντηση με τα αποτελέσματα του αιτήματος του πρώτου.

Στο μοντέλο δημοσίευσης / εγγραφής (Publish / Subscribe) οι συσκευές αρχικά δημοσιεύουν τα δεδομένα σε μια κεντρική πηγή (broker). Ύστερα ο broker μεταβιβάζει τα δεδομένα στους πελάτες που έχουν εγγραφεί στην συγκεκριμένη πηγή. Με αυτόν τον τρόπο ο broker δεν αποθηκεύει δεδομένα απλά λειτουργεί σαν μεταγωγέας.

DDS (Data Distribution Service)

Το Data Distribution Service (DDS) [20] αποτελεί ένα πρωτόκολλο επίπεδου εφαρμογής αξιόπιστης επικοινωνίας M2M (machine-to-machine). Αρχικά το DDS ανακοινώθηκε το 2004 από την OMG (Object Management Group) και ως βασική τεχνική χρησιμοποιεί την publish - subscribe. Σύμφωνα με την OMG το DDS χρησιμοποιείτε κυρίως στην βιομηχανικό τομέα, π.χ. στις μεταφορές, στις έξυπνες εφαρμογές διαχείριση ενέργειας (smart energy), υγειονομικές υπηρεσίες και βιομηχανικό αυτοματισμό. Ο κύρος στόχος του DDS είναι η ανταλλαγή των κατάλληλων δεδομένων στο κατάλληλο χώρο και χρόνο. Η διαχείριση των διασυνδεδεμένων συσκευών γίνεται κεντρικά, γεγονός που ενισχύει την αξιοπιστία και την ασφάλεια του πρωτοκόλλου.

Πιο συγκεκριμένα το DDS παρέχει ένα κατανεμημένο χώρο πληροφοριών (domain) στον οποίο οι εφαρμογές μπορούν να δημοσιεύουν (publish) και να διαβάσουν (consume) δεδομένα με ασύγχρονο και αυτόνομο τρόπο.

Όπως είναι κατανοητό για να εξασφαλιστεί η ασφάλεια και η ποιότητα της υπηρεσίας απαιτείται υπολογιστική ισχύ από ισχυρά συστήματα τα οποία χρειάζεται να επικοινωνούν τόσο μεταξύ τους όσο και με συσκευές χαμηλής υπολογιστικής ισχύς. Για να λυθεί το παραπάνω ζήτημα έχει αναπτυχθεί και μια ‘ελαφριά’ έκδοση του DDS. Επομένως το DDS καταφέρνει να συνδυάσει όλα τα υποσυστήματα με τέτοιο τρόπο ώστε να επιτυγχάνεται η καλύτερη απόδοση και η μέγιστη αξιοπιστία.

Ο στόχος του πρωτοκόλλου μπορεί συνοψιστεί ως ασφαλή μεταφορά της ορθής πληροφορίας στο ιδανικό μέρος εντός αυστηρών χρονικών περιθωρίων.

Χαρακτηριστικά

Η δυναμική ανακάλυψη των δεδομένων στο DDS προσφέρει απομόνωση των εφαρμογών από την τοπολογία δικτύου. Γεγονός που ενισχύει την ανεξαρτησία σε επίπεδο λογισμικού. Ακόμα η τεχνική publish – subscribe προσφέρει μεγάλη ευελιξία την οποία μπορούν να εκμεταλλευτούν οι ‘αποστέλλεις’ και οι αποδέκτες των δεδομένων για να επικοινωνούν ασύγχρονα και ανώνυμα.

Επιπλέον το DDS υποστηρίζει διάφορες πολιτικές QoS (Quality of service) οι οποίες ενισχύουν την διάδοση των δεδομένων σε πραγματικό χρόνο και με μεγάλη αξιοπιστία κατά

την μετάδοση. Αξίζει να σημειωθεί πως τα διάφορα επίπεδα των πολιτικών εξασφάλισης της υπηρεσίας του πρωτόκολλου φτάνουν τα 23. Οι πολιτικές QoS σε συνδυασμό με το όνομα του topic και τη μορφή των δεδομένων ευνοούν την δυναμική ανακάλυψη που απαιτείτε σε peer-to-peer επικοινωνία. Την οποία αξιοποιεί το DDS για την μετάδοση των δεδομένων.

Παρότι το DDS βασίζεται στο publish – subscribe μοντέλο επικοινωνίας έχει μηχανισμούς που του επιτρέπουν να χρησιμοποιηθεί και σαν request – reply μοντέλο σε περιπτώσεις που κάτι τέτοιο είναι αναγκαίο, όπως η μεταβίβαση απομακρυσμένων εντολών.

Το DDS πρωτόκολλο μπορεί να βρίσκεται πάνω από UDP ή TCP ή DTLS (Datagram Transport Layer Security). Ωστόσο από το 2016 και μετά δύναται να χρησιμοποιηθεί “πάνω” από το πρωτόκολλο RTPS [21]. Το RTPS προσφέρει στο DDS authentication, access control, encryption/decryption, data tagging και security event logging.

Ακόμα ένα χαρακτηριστικό του DDS είναι ότι μόνο όσοι συμμετέχουν σε ένα domain μπορούν να επικοινωνούν μεταξύ τους, γεγονός που ενισχύει την ασφάλεια του πρωτοκόλλου. Η ασφάλεια των δεδομένων επιτυγχάνεται κυρίως με φίλτρα και queries.

Αρχιτεκτονική

Μέλη του δικτύου:

Μέλος Domain: Ως μέλος του domain ορίζεται η οντότητα η οποία διαχειρίζεται την διασύνδεση μεταξύ των εφαρμογών που χρησιμοποιούν το ίδιο DDS domain. Για παράδειγμα μπορεί να αποτελείτε από ένα διαχειριστή, ένα container και ένα “κατασκευαστή” για τα υπόλοιπα μέλη του DDS.

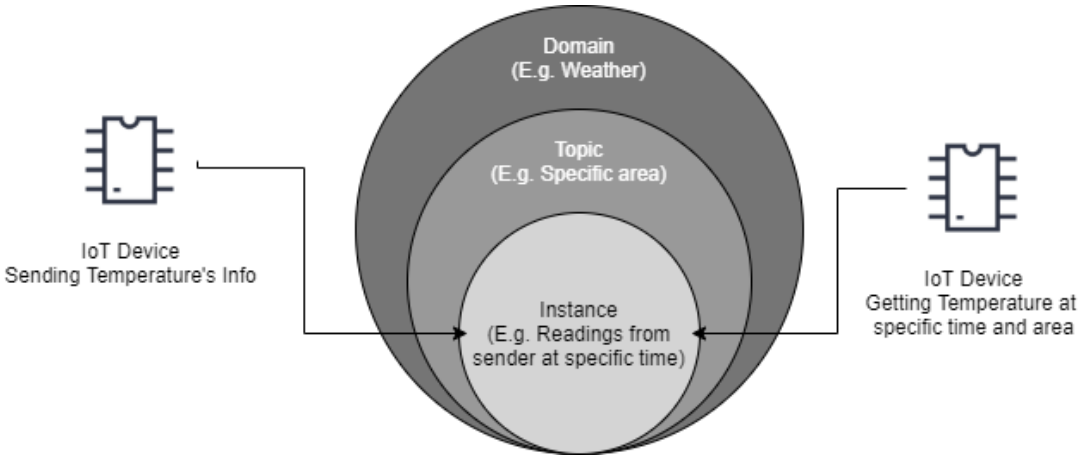
Data Writer and Publisher: Ως publisher ορίζεται η εφαρμογή η οποία δημιουργείται από ένα μέλος του domain και αρχικά χρησιμοποιείται σαν ‘καλούπι’ για την δημιουργία και κατόπιν την διαχείριση μιας ομάδας από data writers, οι οποίοι δημοσιεύουν δεδομένα στον ίδιο τομέα (partition) του domain. Οι data writers χρησιμοποιούνται από τις εφαρμογές, προκειμένου αυτές να δημοσιεύσουν δεδομένα στο domain. Μεταξύ των δυο οντοτήτων (data writers – publishers) υπάρχουν πολιτικές QoS έτσι ώστε η συμπεριφορά τους σαν οντότητες του DDS να είναι εξασφαλισμένη και καθορισμένη.

Subscriber and Data Reader: Ο subscriber δημιουργείτε από ένα μέλος του domain και αναλαμβάνει τον ρόλο της δημιουργίας και της διαχείρισης των data readers. Οι data readers χρησιμοποιούνται από τις εφαρμογές για την λήψη δεδομένων. Η λήψη των δεδομένων μπορεί να γίνει με δυο τρόπους. Ο πρώτος είναι μια listener-based προσέγγιση, η οποία παρέχει ένα μηχανισμό ασύγχρονης επικοινωνίας κατά την οποία η 'ανάγνωση’ δεδομένων γίνεται μέσω callbacks τα οποία ‘τρέχουν’ σε ξεχωριστό thread, με αυτό τον τρόπο η εφαρμογή δεν μπαίνει σε κατάσταση αναμονής. Η δεύτερη μέθοδος προσεγγίζεται με ένα wait-based τρόπο, ο οποίος παρέχει ένα μηχανισμό ‘αναμονής΄ της κύριας εφαρμογής μέχρι να πραγματοποιηθεί μια καθορισμένη τροποποίηση στα δεδομένα.

Topic: Το topic ουσιαστικά καθορίζει την σύνδεση ενός data writer με έναν data reader, και αντιστρόφως. Η σύνδεση επιτυγχάνεται όταν ένα topic που έχει δημοσιευτεί από έναν data writer ‘ταιριάζει’ με ένα topic στο οποίο ένας data reader έχει εγγραφεί. Η επικοινωνία με αυτόν τον τρόπο είναι ανώνυμη και διαφανής δηλαδή οι publishers και οι subscribers δεν χρειάζεται να ενδιαφέρονται ούτε για τον τρόπο ούτε με το ποιος λαμβάνει/γράφει τα δεδομένα μιας και αυτά τα διαχειρίζεται το ενδιάμεσο λογισμικό (middleware) του DD (DCPS).

Αξιολόγηση

Το κύριο πλεονέκτημα του DDS είναι το εύρος των πολιτικών QoS που παρέχει. Οι οποίες εξασφαλίζουν reliability, latency, liveliness, durability, destination order και ownership. Ακόμα παρέχει μηχανισμούς που εξασφαλίζουν την ασφάλεια τόσο των δεδομένων όσο και των μελών ενός domain καθώς και την ανωνυμία τους στο δίκτυο. Ωστόσο χρειάζεται αρκετό χρόνο εκτέλεσης (compile time). Επιπλέον οι επικεφαλίδες που απαιτούνται για την μεταφορά των πακέτων είναι αρκετά μεγάλες. Το προηγούμενο συμβάλει και στην απαίτηση του DDS για μεγάλό εύρος ζώνης. Οπότε γίνετε κατανοητό πως το πρωτόκολλο αδυνατεί να λειτουργήσει ιδανικά σε δίκτυα με ελάχιστους πόρους.



Σχήμα 3 Παράδειγμα Λειτουργίας DDS

AMQP (Advanced Message Queuing Protocol)

Το AMQP είναι ένα πρωτόκολλο επιπέδου εφαρμογής για προσαρμοσμένο ενδιάμεσο λογισμικό. Το AMQP βασίζεται στο μοντέλο εξυπηρετητή - πελάτη (Client – Server) πάνω από το πρωτόκολλο επιπέδου δικτύου TCP/IP, κάνει χρήση του μοντέλου μηνυμάτων δημοσίευσης / εγγραφής (Publish/ Subscribe). Το AMQP σχεδιάστηκε το 2003 από την εταιρεία JPMorgan Chase για χρήση κυρίως από τράπεζες. Για αυτό το λόγο διακρίνεται για την υψηλή απόδοση, αξιοπιστία και την διαλειτουργικότητα μεταξύ των άκρων του δικτύου.

Χαρακτηριστικά

Το AMQP χαρακτηρίζεται ως στρωματοειδές πρωτόκολλο, δηλαδή μια αλλαγή σε οποιοδήποτε επίπεδο δεν επηρεάζει την λειτουργία των υπόλοιπων επιπέδων.

Το πρωτόκολλο χωρίζεται σε σημαντικούς τομείς: το μοντέλο αναμονής καθώς και το μοντέλο μεταφοράς.

Η αυτοδυναμία στα μηνύματα που παρέχεται από το πρωτόκολλο του προσδίδει μεγάλη χρονική ευελιξία και ευελιξία στην διαχείριση μεγάλων ή μικρών σε μέγεθος μηνυμάτων.

Συστατικά του δικτύου

Ουρές (Queues): Οι ουρές ουσιαστικά αποτελούν τον κορμό του πρωτοκόλλου. Όλα τα μηνύματα αποθηκεύονται σε ουρές. Στις οποίες μπορούν να υλοποιηθούν αναδιάταξη, αναζήτηση, ανταλλαγές μηνυμάτων και προσφέρουν στον διαχειριστή του συστήματος την δυνατότητα να ορίσει τους μηχανισμούς εκείνους που εξασφαλίζουν την ποιότητα της προσφερόμενης υπηρεσίας (QoS).

Ανταλλαγές (Exchanges): Οι ανταλλαγές αποτελούν τον μηχανισμό που καθορίζει τον τρόπο παράδοσης κάθε μηνύματος στον παραλήπτη. Ουσιαστικά με τις ανταλλαγές ορίζεται εάν η παράδοση θα είναι έμμεση (μοντέλο publish/ Subscribe) ή άμεση. Το πρωτόκολλο υποστηρίζει την άμεση ανταλλαγή, την ανταλλαγή θέματος και την ανταλλαγή κεφαλίδων. Αξίζει να σημειωθεί ότι δεν γίνεται αποθήκευση των μηνυμάτων κατά την διαδικασία της ανταλλαγής αλλά μόνο των δεσμευτικών παραμέτρων.

Δεσμοί (Bindings): Οι παράμετροι είναι αναγκαία στοιχεία για την δρομολόγηση των μηνυμάτων στο AMQP. Οι δεσμοί είναι διαφορετικοί σε κάθε μηχανισμό ανταλλαγής του πρωτοκόλλου προκειμένου να εξυπηρετήσουν τις ανάγκες κάθε μηχανισμού.

Αξιολόγηση

Το AMQP ενσωματώνει τα πρωτόκολλα TLS/SSL γεγονός που του προσφέρει ασφάλεια κατά την μετάδοση των μηνυμάτων. Παρέχει μηχανισμούς QoS μέσω της διαχείρισης των ουρών.

Ωστόσο η διεπαφή του με τον χρήστη είναι αρκετά δύσκολη. Ακόμα ο σχεδιασμός του οδηγεί σε υλοποιήσεις με αρκετά ογκώδη κώδικα.

MQTT (Message Queue Telemetry Transport)

Το MQTT αρχικά είχε σχεδιαστεί από την IBM για χρήση σε συσκευές με περιορισμένους πόρους. Το πρωτόκολλο βασίζεται στο TCP/IP και στο μοντέλο publish/subscribe με χρήση broker για την μετάδοση μηνυμάτων. Μπορούμε να θεωρήσουμε το MQTT-SN (MQTT For Sensor Networks) σαν επέκταση το MQTT. Η επέκταση αυτή βασίζεται στο μοντέλο εξυπηρετητή - πελάτη (Client – server) πάνω από UDP και στο μοντέλο publish/subscribe.

Χαρακτηριστικά

Το πρωτόκολλο παρέχει την δυνατότητα για την διατήρηση της συνεδρίας (session) του χρήστη με τον εξυπηρετητή. Κατά την παράδοση του μηνύματος οι ενδιάμεσες συσκευές δεν έχουν την δυνατότητα αποθήκευσης και πρόσβασης της μεταφερόμενης πληροφορίας.

Το MQTT υποστηρίζει την ασύγχρονη μετάδοση μηνυμάτων. Ακόμα παρέχει τρία επίπεδα για την εξασφάλιση της ποιότητας της προσφερόμενης υπηρεσίας (QoS). Αυτά τα επίπεδα είναι η ‘το πολύ μια φορά’, η ‘τουλάχιστον μια φορά’, η ‘ακριβώς μια φορά’ μετάδοση της πληροφορίας.

Η «επέκταση» MQTT-SN προσδίδει το χαρακτηριστικό της ανακάλυψης πυλών του δικτύου (gateway discovery) καθώς και λειτουργίες όπως η κατάσταση ύπνου (sleep mode) και μικρού μεγέθους επικεφαλίδες.

Αξιολόγηση

Το MQTT λόγο της απλότητας του είναι ιδανικό για την M2M επικοινωνία συσκευών σε δίκτυα περιορισμένων πόρων. Μέσω των τριών αναφερόμενων επιπέδων QoS καθώς και την χρήση του πρωτοκόλλου TLS εξασφαλίζεται η ασφάλεια, η αξιοπιστία και διαλειτουργικότητα κατά την επικοινωνία των άκρων του δικτύου.

Το MQTT δεν συνίσταται για χρήση σε πολύπλοκα και διευρυμένων δυνατοτήτων δίκτυα μιας και η διατήρηση της απλότητας του δυσκολεύει εκθετικά με το μέγεθος το δικτύου.

Το MQTT-SN απαιτεί μικρή χρήση μπαταρίας λόγω του σχεδιασμού του.

	MQTT	MQTT-SN
Ιδανική Χρήση	Σε συσκευές με περιορισμένους πόρους	Σε ασύρματα δίκτυα αισθητήρων
Πρωτόκολλο επιπέδου Μεταφοράς	TCP	UDP
Τρόπος επικοινωνίας συσκευών	Topic Name	Topic ID
Τεχνικές εξοικονόμησης ενέργειας	Δεν παρέχει	Sleep mode, small payload size

Πίνακας 1 Διαφορές MQTT-SN και MQTT

XMPP (eXtensible Messaging and Presence Protocol)

Το XMPP σχεδιάστηκε το 1998 από την Jabber (κοινότητα ανοικτού κώδικα) με σκοπό την άμεση επικοινωνία μεταξύ των χρηστών και περιγράφεται από το [22]. Προκειμένου να πετύχει το στόχο του το πρωτόκολλο κάνει χρήση της γλώσσας XML και έτσι διευκολύνεται η άμεση ανταλλαγή μηνυμάτων, πληροφορίων, κλήσεις βίντεο και φωνής με την παρουσία των χρηστών.

Το XMPP βασίζεται στο μοντέλο αιτήματος / απάντησης (Request / Response) πάνω από το πρωτόκολλο TCP/IP. Ωστόσο δίνατε να γίνει χρήση του μοντέλου δημοσίευσης / εγγραφής με το πρότυπο που περιγράφεται στο [23].

Χαρακτηριστικά

Το XMPP δίνει την δυνατότητα σε έναν χρήστη να παραμένει συνδεδεμένος σε πολλαπλές συσκευές ταυτόχρονα. Το παραπάνω επιτυγχάνεται επειδή η διευθυνσιοδότηση γίνεται με βασικό άξονα την σύνδεση του κάθε χρήστη και της συσκευής που χρησιμοποιεί.

Τα άκρα του πρωτοκόλλου ανταλλάσσουν συνεχώς πληροφορίες σε XML σε συνεχή ροή (streaming).

Αξιολόγηση

Το XMPP παρέχει ασφάλεια και αξιοπιστία κατά την μετάδοση των μηνυμάτων μέσω του προτύπου TLS. Ακόμα με την χρήση κατάλληλων πυλών (gateways) και την διασύνδεση με XMPP εξυπηρετητές (servers) παρέχεται η δυνατότητα μετάδοσης πληροφοριών με άλλα συστήματα. Αποτελεί το πιο αξιόπιστο πρωτόκολλο για τον έλεγχο οικιακών συσκευών.

Το σημαντικό μειονέκτημα του πρωτοκόλλου είναι ότι δεν παρέχει μηχανισμούς εξασφάλισης της προσφερόμενης πληροφορίας (QoS).

REST (Representational State Transfer)

Το Representational State Transfer (REST) σχεδιάστηκε το 2000 από τον Roy Fielding [24]. Το REST βασίζεται στο μοντέλο αιτήματος / απάντησης (Request / Response) και αξιοποιεί τις μεθόδους του HTTP (GET,POST,PUT,DELETE) στην ανταλλαγή ασύγχρονων μηνυμάτων. Ο στόχος του REST στο IoT είναι η M2M επικοινωνία σε εφαρμογές όπως η έξυπνη ενέργεια και ο αυτοματισμός κτιρίων.

Χαρακτηριστικά

Το πρωτόκολλο μπορεί να αξιοποιήσει πλήρως τις δυνατότητες που δίνει το HTTP, δηλαδή την αυθεντικοποίηση, την προσωρινή αποθήκευση (caching) καθώς και τον ορισμό της μορφής των δεδομένων προς μεταφορά. Η μορφή των δεδομένων πρέπει να είναι είτε σε XML είτε σε JSON, την απόφαση για την μορφή των δεδομένων την «παίρνει» ο εξυπηρετητής. Χρησιμοποιείται ήδη στο IoT μιας και αποτελεί το βασικό χαρακτηριστικό σε πληθώρα διαδικτυακών εφαρμογών και είναι ανεξάρτητο από το λειτουργικό σύστημα καθώς και από το υλικό πάνω στο οποίο λειτουργεί.

Το REST λειτουργεί κυρίως πάνω από το TCP/IP. Ακόμα ενδέχεται να «τρέξει» πάνω στο HTTPS με την χρήση του TLS/SSL γεγονός που του προσδίδει ασφάλεια και αξιοπιστία.

Αξιολόγηση

Το REST όπως είπαμε χρησιμοποιείται σήμερα ευρέως ωστόσο η χρήση του HTTP το αποκλείει από τις πιο πολλές M2M πλατφόρμες, μιας και αυτές δεν υποστηρίζουν το HTTP(S).

Το πρωτόκολλο έχει αρκετά μεγάλες επικεφαλίδες στα πακέτα και εφόσον συνδυαστεί με το Long Polling οδηγούν σε μεγάλη κατανάλωση πόρων (μπαταρίας).

CoAP (Constrained Application Protocol)

Το CoAP είναι ένα πρωτόκολλο επικοινωνίας επιπέδου εφαρμογής που σχεδιάστηκε για χρήση σε συσκευές με περιορισμένους πόρους. Σχεδιάστηκε από την IETF (Internet Engineering Task Force) και το πρότυπο [25] περιγράφει το πρωτόκολλο. Ακόμα το πρωτόκολλο αποτελεί μια πιο «ελαφριά» έκδοση του REST επομένως βασίζεται στο μοντέλο αιτήματος / απάντησης (Request / Response). Ωστόσο με την επέκταση Observers μπορεί να κάνει χρήση και του μοντέλου μοντέλο δημοσίευσης / εγγραφής (Publish / Subscribe).

Χαρακτηριστικά

Το CoAP χρησιμοποιεί το πρωτόκολλο δικτύου επιπέδου μετάδοσης UDP. Η ασφάλεια κατά την μετάδοση μηνυμάτων εξασφαλίζεται με την χρήση του DTLS (Datagram Transport Layer Security).

Το πρωτόκολλο υποστηρίζει τα χαρακτηριστικά του HTTP μέσω διαμεσολαβητών (proxies). Επεκτείνεται με χαρακτηριστικά όπως την ανακάλυψη πόρων (resource discovery), την υποστήριξη πολλαπλής εκπομπής μηνυμάτων (multicast) και την δυνατότητα προσωρινής αποθήκευσης (caching).

Με βάση τα παραπάνω χρησιμοποιεί μικρές σε μήκος επικεφαλίδες overhead.

Αξιολόγηση

Το CoAP παρέχει την δυνατότητα της συνεχείς παρακολούθησης των μετρήσεων με την βοήθεια των Observers. Επομένως γίνεται αντιληπτό ότι υποστηρίζει και την ασύγχρονη ανταλλαγή μηνυμάτων.

Επίσης υποστηρίζει την αυτόματη αναγνώριση υπηρεσιών μέσω του πρωτοκόλλου Constrained RESTful Environments (CoRE) [26].

Το πρωτόκολλο μπορεί να εγγυηθεί μεγάλη αποδοτικότητα σε συσκευές με ελάχιστη χωρητικότητα τόσο σε μνήμη όσο και σε ενέργεια.

Το CoAP διαθέτει την δυνατότητα χαρακτηρισμού μηνυμάτων ως επιβεβαιώσιμα ή μη και αυτός είναι ο μοναδικός μηχανισμούς QoS που προσφέρει το COAP, γεγονός που το καθιστά μη αξιόπιστο πρωτόκολλο.

Στον πίνακα 2 γίνεται σύγκριση μεταξύ όλων των πρωτοκόλλων του επιπέδου εφαρμογής. Το κάθε πρότυπο είναι ιδανικό εφόσον τεθεί στο ορθό πλαίσιο και υλοποιηθεί για την λύση που έχει σχεδιαστεί. Επομένως ο σχεδιαστής του συστήματος οφείλει να λάβει υπόψιν όλα όσα μελετήθηκαν προηγουμένως και συνοψίζονται στον παρακάτω πίνακα.

	DDS	**AMQP**	**MQTT**	**XMPP**	**REST**	**CoAP**
Πρωτόκολλο Μεταφοράς	TCP/UDP	TCP	TCP	TCP	TCP	UDP
Μοντέλο εξυπηρέτησης	Υποστηρίζει και τα δυο	Υποστηρίζει και τα δυο	Δημοσίευσης /εγγραφής	Υποστηρίζει και τα δυο	Αιτήματος / απάντησης	Υποστηρίζει και τα δυο
QoS	Σχεδόν 23 επίπεδα	3 επίπεδα (at most once, at least once, Exactly one)	3 επίπεδα (at most once, at least once, Exactly one)	Κανένα	Κανένα	Επιβεβαιώσιμο ή μη
Ασφάλεια	TLS/DTLS/ Ενσωματωμένη	TLS/SSL	TLS/SSL	TLS/SSL	SSL in HTTPS	DTLS
Υποστήριξη RESTful	Όχι	Όχι	Όχι	Όχι	Ναι	Ναι
	Περίπλοκες εφαρμογές	Επικοινωνία μεταξύ των	Διασύνδεση συσκευών	Εφαρμογές	Έξυπνη ενέργεια	Διαχείριση συσκευών

Ιδανική χρήση	βιομηχανικού τομέα	άκρων δικτύων	με περιορισμένους πόρους	πραγματικού χρόνου		με περιορισμένους πόρους

Πίνακας 2 Διαφορές Πρωτοκόλλων επιπέδου εφαρμογής

Υπηρεσία Ανακάλυψης (Service Discovery)

Με το παραπάνω όρο εννοούμε την διαδικασία κατά την οποία οι συσκευές ανακαλύπτουν άλλες υπηρεσίες ή συσκευές που βρίσκονται στο ίδιο δίκτυο. Συγκεκριμένα θα μελετήσουμε τις υπηρεσίες mDNS και DNS-SD.

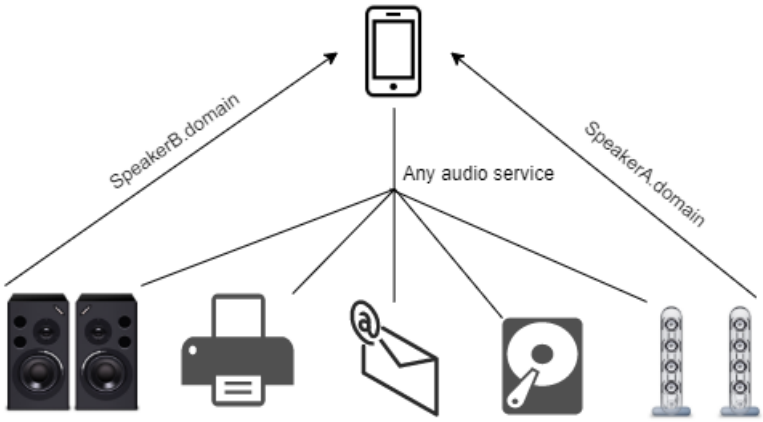
mDNS (multicast Domain Name System)

Η υπηρεσία multicast Domain Name System (mDNS) [27] εκτελεί παρόμοιες λειτουργίες με έναν απλό DNS server. Ακόμα αποτελεί μια απλή λύση μιας και μπορεί να εφαρμοστεί στις συσκευές που βρίσκονται στο ίδιο δίκτυο χωρίς να απαιτείται κάποιος επιπλέον εξοπλισμός ούτε παραμετροποίηση από τον χρήστη. Ακόμα η υπηρεσία αυτή παρέχει μεγάλη ανοχή σφαλμάτων αφού μπορεί να συνεχιστεί κανονικά η λειτουργία σε περίπτωση που κάποιες συσκευές πάψουν να λειτουργούν.

DNS-SD (DNS Service Discovery)

Το πρωτόκολλο DNS-SD σχεδιάστηκε από τους S. Cheshire, M. Krochmal το 2013 [28]. Σκοπός του πρωτοκόλλου είναι η ανακάλυψη των προσφερόμενων υπηρεσιών μέσα σε ένα τοπικό δίκτυο μέσα από μηνύματα DNS.

Το DNS-SD μπορεί να χρησιμοποιεί mDNS ή Unicast DNS. Εφόσον γίνει χρήση του mDNS το πρωτόκολλο παρέχει άμεση λειτουργία χωρίς την ανάγκη όποιας διαχείρισης ή ρύθμισης παραμέτρων στις διασυνδεδεμένες συσκευές του δικτύου. Η ανακάλυψη των προσφερόμενων υπηρεσιών αρχικά απαιτεί την εύρεση των host names που παρέχουν την επιθυμητή υπηρεσία και κατόπιν το ταίριασμα των host names με την IP διεύθυνσή τους. Εάν όμως το DNS-SD εγκατασταθεί με την χρήση του Unicast DNS τότε απαιτείται ρύθμιση των συσκευών που θα παρέχουν υπηρεσίες. Οι παράμετροι αυτές μπορεί να είναι το domain name όπου θα διαφημίζονται οι υπηρεσίες καθώς και η παραχώρηση των απαραίτητων δικαιωμάτων στον DNS εξυπηρετητή για συνεχής ενημέρωση όποιων μεταβολών στις συσκευές.



Σχήμα 4 Παράδειγμα λειτουργίας DNS Service Discovery

Επίπεδο Δικτύου (Network Layer)

Το επίπεδο δικτύου είναι υπεύθυνο για την διασύνδεση των διαφορετικών δικτύων επικοινωνίας. Το κάθε δίκτυο επικοινωνίας συλλέγει, μεταφέρει και επεξεργάζεται δεδομένα. Ουσιαστικά το επίπεδου δικτύου περιλαμβάνει τα ipv4 και ipv6 και 6LoWPAN, τα προηγούμενα χρησιμοποιούνται κατά κόρον σε όλα τα IoT περιβάλλοντα.

IPv4

Το IPv4 [29] αποτελεί ένα από τα παλιότερα πρωτόκολλα που χρησιμοποιούνται στις μέρες μας. Σχεδιάστηκε το Σεπτέμβριο του 1981. Το IPv4 διαχειρίζεται την διευθυνσιοδότηση του κάθε συστήματος σε ένα δίκτυο και αναλαμβάνει την δρομολόγηση των πακέτων στον τελικό προορισμό. Τις παραπάνω λειτουργίες τις επιτυγχάνει με ένα σύστημα διευθυνσιοδότησης με δυο μέρη. Το πρώτο μέρος, το οποίο ονομάζεται κεφαλίδα, περιλαμβάνει όλα τα απαραίτητα στοιχεία για να αποσταλθεί το πακέτο στον τελικό προορισμό του, π.χ. Διευθύνσεις αφετηρίας κι προορισμού, έκδοση, συνολικό μήκος πακέτου, time to live, id. Το δεύτερο μέρος αποτελείται από τα δεδομένα. Η διαδικασία κατά την οποία συνδυάζονται τα δυο μέρη ονομάζεται ενθυλάκωση.

Κάθε διεύθυνση έχει μήκος 32-bit όποτε το ipv4 μπορεί να δώσει συνολικά 4.294.967.296 (2^{32}) και ένα παράδειγμα ipv4 διεύθυνσης σε δεκαδικό σύστημα είναι 127.0.0.1 .

IPv6

Το IPv6 [30] ουσιαστικά αποτελεί εξέλιξη του IPv4. Το IPv6 είναι απολύτως απαραίτητο μιας και από το 2012 όλες οι διαθέσιμες διευθύνσεις του IPv4 έχουν διαμοιραστεί σε συσκευές. Το μήκος κάθε διεύθυνσης είναι 128-bit δηλαδή το ipv6 μπορεί να διαθέσει συνολικά 7.9 * 1028 (θεωρητικά περίπου 2128) διευθύνσεις και ένα παράδειγμα διεύθυνσης ipv6 σε δεκαεξαδικό σύστημα είναι: 2001:0db8:85a3:0000:0000:8a2e:9685:5879. Κάθε διεύθυνση αποτελείται από δυο μέρη το πρόθεμα του υποδικτύου (subnet prefix) και το μοναδικό αναγνωριστικό της διεπαφής (Interface Identifier “IID”). Το μήκος του κάθε μέρους ενδέχεται να διαφέρει αλλά ο πιο τυπικός διαχωρισμός είναι 64-bit ανά μέρος. Το μεγάλο εύρος του μοναδικού αναγνωριστικού επιτρέπει την επιλογή του κατά περίπτωση και ανάλογα το σενάριο χρήσης. Χάρη στο μεγάλο εύρος των διευθύνσεων του IPv6 πλέον υπάρχει η δυνατότητα κάθε συσκευή να συνδεθεί με τουλάχιστον μια υπηρεσία στο διαδίκτυο. Ένα ακόμα πλεονέκτημα του ipv6 είναι η μη αναγκαία χρήση του NAT γεγονός που αυξάνει την συνδεσιμότητα, την αξιοπιστία και την προσαρμοστικότητα του δικτύου.

Το IPv6 καθορίζει τις διευθύνσεις σε δυο εμβέλειες, η μια είναι τοπική, η δεύτερη καθολική και τουλάχιστον μια καθορίζεται για κάθε διεπαφή του κόμβου. Η τοπική διεύθυνση χρησιμοποιείται για την αυτόματη ανακάλυψη των άκρων του τοπικού δικτύου και για την αυτόματη παραμετροποίηση. Τα πακέτα που αποστέλλονται από μια τοπική IP δεν προωθούνται στο διαδίκτυο από τους δρομολογητές μιας και η IP ενδέχεται να μην είναι μοναδική. Η καθολική διεύθυνση αποτελεί μοναδικό αναγνωριστικό της εκάστοτε διεπαφής και για αυτό χρησιμοποιείται για την επικοινωνία στο διαδίκτυο. Όπως είναι σαφές από τα παραπάνω ένας κόμβος χρειάζεται τουλάχιστον μια καθολική διεύθυνση για μπορεί να συνδεθεί στο διαδίκτυο.

Οι διαφορές του IPv6 με το IPv4 αφορούν και την απλοποίηση των επικεφαλίδων κάθε πακέτου προκείμενου να μειωθούν τα κόστη επεξεργασίας και αναμετάδοσης των πακέτων.

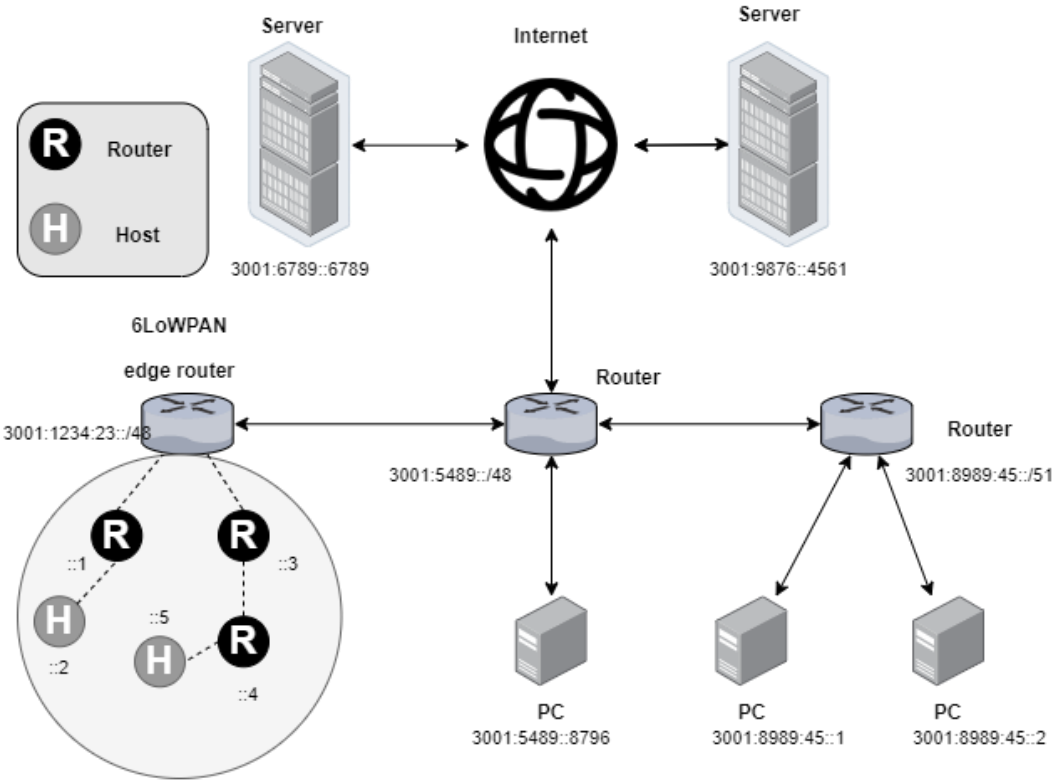
Επιπλέον οι αλλαγές στις επικεφαλίδες προσφέρουν επεκτασιμότητα, αποδοτικότερη προώθηση. Σε αυτό το πλαίσιο το IPv6 προσφέρει και την δυνατότητα εξασφάλισης της ποιότητας της προσφερόμενης υπηρεσίας (QoS) [31].

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks)

Το 6LoWPAN [32] εισηγήθηκε το 2007 μιας και υπήρχε επιτακτική ανάγκη για την δημιουργία ενός πρωτοκόλλου επικοινωνίας IoT συσκευών με περιορισμένους ενεργειακούς πόρους και καθιστά εύκολη την επικοινωνία με τα IP πρωτόκολλα «ipv4, ipv6» (IoT MTU “Maximum transmission unit” =127 bytes, IPv6 minimum MTU=1280 bytes). Το 6LoWPAN αποτελεί το πρωτόκολλο που ενισχύει ουσιαστικά την διασύνδεση κάθε συσκευής στο διαδίκτυο. Το πρότυπο είναι αρκετά ευέλικτο τόσο στην συχνότητα λειτουργίας (frequency) όσο και στο πρωτόκολλο «πάνω» από το οποίο θα λειτουργεί (ethernet, wifi, 802.15.4).

Μιας και τα δίκτυα του προτύπου μπορούν εξ ορισμού να επικοινωνούν άμεσα με τα IP δίκτυα ο μόνος απαραίτητος εξοπλισμός είναι ένας δρομολογητής. Τα 6LoWPAN δίκτυα γενικά μπορούν να περιγραφούν σαν «δίκτυα τσέπης» (stub networks) αφού λειτουργούν στα άκρα του δικτύου.

Ένα 6LoWPAN δίκτυο αποτελείται από τις εξής συσκευές: router και hosts. Οι hosts είναι συσκευές που δεν μπορούν να προβούν στην δρομολόγηση δεδομένων σε άλλες συσκευές, ωστόσο ενδέχεται να έχουν την δυνατότητα να ελέγχουν ανά τακτικά χρονικά διαστήματα εάν στα routers υπάρχουν δεδομένα που τις αφορούν. Τα routers ουσιαστικά δρομολογούν την κίνηση τόσο μεταξύ των συσκευών του 6LoWPAN δικτύου όσο κι μεταξύ των συσκευών με άλλα IP δίκτυα όπως το Wi-Fi, Ethernet, 5G ή 4G κ.α. Καθώς οι δρομολογητές λειτουργούν σε επίπεδου δικτυού δεν συγκρατούν την κατάσταση του επιπέδου εφαρμογής, γεγονός που συντελεί στην μείωση του φόρτου εργασίας κι στην χρήση των χαμηλού κόστους ενσωματωμένων συσκευών μέσω απλού λογισμικού ως router άκρου.



Σχήμα 5 Δίκτυο 6LoWPAN

RPL (Routing protocol for low power and lossy networks)

Στο επίπεδο δικτύου το IoT κάνει χρήση του πρωτόκολλου δρομολόγησης RPL (Routing protocol for low power and lossy networks). Το οποίο σχεδιάστηκε για χρήση στο IPv6. Όπως υποδεικνύει και το όνομά του το RPL χρησιμοποιείται κυρίως σε δίκτυα με περιορισμένους πόρους όπως τα WPANs (wireless personal area networks), PLC (low-power line communication) και WSNs (wireless sensor networks). Στο [33] αναφέρονται μερικά χαρακτηριστικά των δικτύων με περιορισμένους πόρους. Οι πιο πολλοί κόμβοι τέτοιων δικτύων έχουν περιορισμένη επεξεργαστική ισχύ, ελάχιστη διαθέσιμη μνήμη καθώς και χαμηλά αποθέματα ενέργειας. Το προηγούμενο χαρακτηριστικό των δικτύων με περιορισμένους πόρους συντελεί στην ασταθή δικτύωση με χαμηλά ποσοστά επιτυχώς παραδοτέων πακέτων. Ο τρόπος επικοινωνίας των nodes τέτοιων δικτύων δεν είναι point-to-point αλλά ενδέχεται να είναι και multi-point-to-point, γεγονός που αυξάνει την πολυπλοκότητα του δικτυού. Το σύνολο των διασυνδεμένων κόμβων ενδέχεται να φτάσει σε δεκάδες χιλιάδες. Ακόμα το κάθε δίκτυο ενδέχεται να ενσωματώνει διαφορετικές εκδόσεις του RPL, όπου κάθε έκδοση ενδέχεται να εξυπηρετεί διαφορετικούς και ”ανταγωνιστικούς” περιορισμούς ή κριτήρια απόδοσης.

Για να επιτύχει την ευρέα χρήση από πληθώρα δικτύων με περιορισμένους πόρους το RPL ξεχωρίζει τις διαδικασίες της επεξεργασίας και της προώθησης των πακέτων από την βελτιστοποίηση της δρομολόγησή τους. Το πρωτόκολλο ωστόσο απαιτεί την αμφίδρομη επικοινωνία μεταξύ των κόμβων και για αυτό το λόγο καθίσταται υποχρεωτική η επιβεβαίωση της σύνδεσης της εκάστοτε συσκευής με τον δρομολογητή πριν αυτός χρησιμοποιηθεί για την προώθηση πακέτων. Το πρότυπο ολοκληρώνει την παραπάνω λειτουργία κατά την φάση εύρεσης “γονέα" και αφού λάβει έναυσμα από έναν εξωτερικό μηχανισμό για την επιβεβαίωση των ιδιοτήτων της σύνδεσης του κόμβου με τον δρομολογητή καθώς και πληροφορίες για γειτονικούς κόμβους. Η προηγούμενη διαδικασία μπορεί να γίνει με διάφορους τρόπους (Neighbor Unreachability Detection, Bidirectional Forwarding Detection [34]), γενικά προτιμώνται μηχανισμοί που ενεργοποιούνται με την κίνηση στο δίκτυο προκειμένου να μειωθεί το κόστος της επιτήρησης συνδέσεων που ενδέχεται να μην είναι ενεργές.

Το πρότυπο αναμένει ακόμα την ενεργοποίηση ενός εξωτερικού μηχανισμού όπου θα επιδιώκει πρόσβαση και ακολούθως θα μεταφέρει ορισμένες πληροφορίες ελέγχου, οι οποίες ονομάζονται RPL Packet Information. Αυτές οι πληροφορίες αφορούν την σύνδεση ενός RPL πακέτου με ένα συγκεκριμένο πρότυπο (στιγμιότυπο) του RPL που εφαρμόζεται στο δίκτυο και καθιστούν εφικτή την επαλήθευση των καταστάσεων δρομολόγησης. Ένας τέτοιος μηχανισμός ονομάζεται RPL option, περιγράφεται στο [35] και είναι αναγκαίος για όλα τα πακέτα εκτός αυτών που επιβάλλουν αυστηρά κριτήρια δρομολόγησης, τα οποία εξ’ ορισμού αποφεύγουν τους ατέρμονους βρόγχους και περιορίζουν την ανάγκη χρήσης των πακέτων RPL Information.

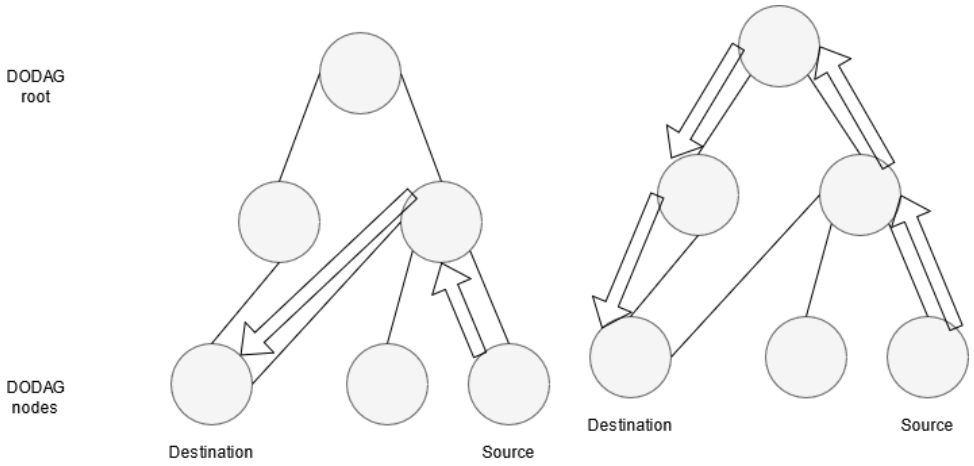
Το πρωτόκολλο RPL παρέχει έναν μηχανισμό διάδοσης της πληροφορίας σε δίκτυα που σχηματίζονται δυναμικά. Αυτός ο μηχανισμός καθιστά εφικτή την επικοινωνία των κόμβων του δικτύου με ελάχιστη παραμετροποίηση αυτών με αποτέλεσμα οι κόμβοι να μπορούν να λειτουργούν αυτόνομα. Η παραπάνω λειτουργία κάνει χρήση του αλγορίθμου Trickle [36] προκειμένου να βελτιστοποιήσει την διάδοση της πληροφορίας στους κόμβους του δικτύου.

Συχνά ένας κόμβος του δικτύου μπορεί να λειτουργεί σαν δέκτης και σαν δρομολογητής. Σαν δέκτης υλοποιεί τις διαδικασίες που αναφέρονται στα [37], [38], [39] και [40]. Ενώ σαν

δρομολογητής έχει την δυνατότητα να μεταδίδει πληροφορίες που είναι απαραίτητες για συγκεκριμένη σύνδεση σε άλλους κόμβους του δικτύου.

Οι δρομολογητές του δικτύου λειτουργούν με δυο τρόπους δηλαδή είτε με δρομολόγηση προς τα πάνω είτε προς τα κάτω. Με τον πρώτο τρόπο όλη η κίνηση μεταφέρεται από την πηγή προς τον κόμβο ρίζας (root node) και έπειτα κατευθύνεται προς τον κόμβο προορισμού. Με τον δεύτερο τρόπο τα πακέτα δρομολογούνται με κατεύθυνση προς τα κάτω με βάση την διεύθυνση του προορισμού.

Το πρότυπο συμμορφώνεται με την πολύ-επίπεδη αρχιτεκτονική του πρωτοκόλλου IP και επομένως δεν στηρίζεται σε συγκεκριμένα πρωτόκολλα του επιπέδου ζεύξης. Επομένως το RPL έχει την δυνατότητα να “συνεργάζεται" με διάφορα πρωτόκολλα χαμηλότερου επιπέδου σε αυτά συμπεριλαμβάνονται αυτά που αφορούν δίκτυα με περιορισμένους πόρους και ενδεχομένως με πολλές απώλειες.



Σχήμα 6 Δρομολόγηση προς τα κάτω (Αριστερά) και δρομολόγηση προς τα πάνω(δεξιά)

Επίπεδο Ζεύξης Δεδομένων (Link Layer)

Το επίπεδο ζεύξης δεδομένων αποτελείται από τα πρότυπα που καθορίζουν τη μετάδοση της πληροφορίας ανάμεσα σε συσκευές που βρίσκονται στο ίδιο τοπικό δίκτυο. Το πρωταρχικό πρωτόκολλο που χρησιμοποιείται σε IoT περιβάλλοντα είναι το IEEE 802.15.4.

IEEE 802.15.4

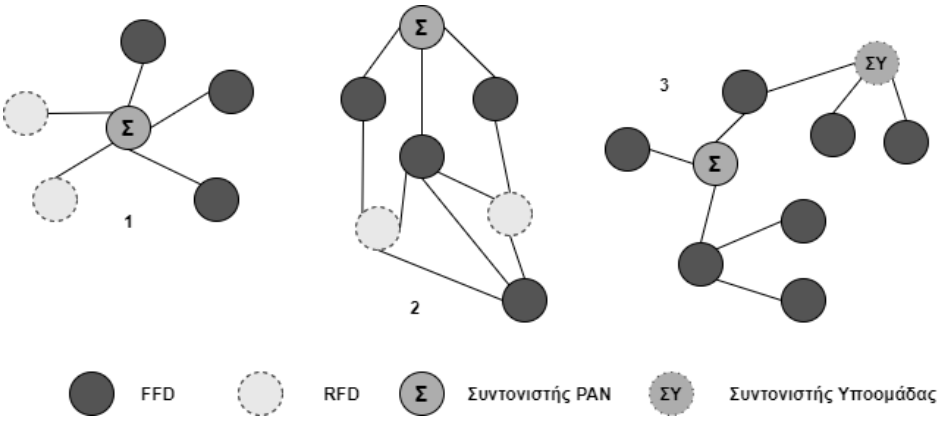
Το πρότυπο IEEE 802.15.4 [41] είναι επιφορτισμένο με την διαχείριση των συσκευών που ανήκουν σε ένα LR-WPANs (Low-Rate Wireless Personal Area Networks). Αρχικά σχεδιάστηκε το 2003 από την ομάδα IEEE 802 και ο στόχος της ήταν η επίτευξη επικοινωνίας μεταξύ συσκευών με εξαιρετικά ελάχιστο κόστος τόσο σε ενέργεια όσο και σε υπολογιστική ισχύ. Το πρωτόκολλο είναι ιδανικό για την διασύνδεση IoT συσκευών μιας και υποστηρίζει την σύνδεση μέχρι 65 χιλιάδων συσκευών σε ένα δίκτυο με μηχανισμούς ασφάλειας, κρυπτογράφησης καθώς και αυθεντικοποίησης. Ωστόσο το πρότυπο δεν παρέχει μηχανισμούς εξασφάλισης της προσφερόμενης υπηρεσίας (QoS).

Το πρωτόκολλο υποστηρίζει δυο ειδών συσκευές τις πλήρως λειτουργικές συσκευές «Fully functional devices (FFD)» και της ‘μειωμένης’ λειτουργικότητας συσκευές «Reduced functional devices (RFD)». Οι συσκευές FFD μπορούν να λειτουργήσουν είτε σαν συντονιστές του δικτύου είτε σαν απλά μέλη του δικτύου. Οι συντονιστές επιτελούν τις εξής λειτουργίες α) δημιουργία, β) έλεγχος λαθών, γ) συντήρηση του δικτύου, δ) εφαρμογή κρυπτογράφησης με το πρωτόκολλο MAC, ε) αποθήκευση πίνακα δρομολόγησης στην μνήμη τους καθώς και

ζ) επικοινωνία με άλλα άκρα του δικτύου με χρήση των τοπολογιών: 1) αστέρας, 2) Peer-to-Peer και 3) cluster-tree. Ενώ οι συσκευές τύπου RFD έχουν μόνο την δυνατότητα να λειτουργούν ως απλά άκρα του δικτύου, μιας και διαθέτουν ελάχιστους πόρους, καθώς και την επικοινωνία με συντονιστές του δικτύου με χρήση της τοπολογίας αστέρας.

Οι τοπολογίες που περιεγράφηκαν παραπάνω λειτουργούν ως εξής:

1. Αστέρας: Σε αυτήν την τοπολογία χρησιμοποιείτε τουλάχιστον μια συσκευή τύπου FFD και μια ή παραπάνω τύπου RFD. Η συσκευή τύπου FFD τοποθετείτε στο κέντρο του δικτύου PAN (Personal area network) και ελέγχει - διαχειρίζεται όλα τα υπόλοιπα άκρα του δικτύου επομένως γίνεται και ο συντονιστής του δικτύου.
2. Peer-to-Peer: Σε αυτήν την περίπτωση υπάρχει ένας συντονιστής του δικτύου PAN και τα υπόλοιπα άκρα επικοινωνούν άμεσα μαζί του ή με την 'βοήθεια' ενδιάμεσων άκρων από άλλα άκρα
3. Cluster-tree: Πρόκειται για μια ειδική μορφή peer-to-peer τοπολογίας όπου εκτός από έναν συντονιστή του δικτύου PAN και απλά άκρα του δικτύου περιέχει έναν συντονιστή υποομάδας (cluster head).



Σχήμα 7 (1-3) Τοπολογίες [42]

Το IEEE 802.15.4 έχει την δυνατότητα να λειτουργεί τόσο σε επίπεδο ζεύξης δεδομένων όσο και στο φυσικό επίπεδο των συσκευών. Στο Link layer αναλαμβάνει την εξασφάλιση της σύνδεσης μεταξύ δυο ή παραπάνω συσκευών όπως και την διόρθωση τυχόν λαθών του δικτύου. Στο φυσικό επίπεδο των συσκευών εξασφαλίζει την μετατροπή των bits σε σήμα το οποίο μπορεί να μεταδοθεί στον αέρα.

Φυσικό Επίπεδο Συσκευών (Physical/Device Layer)

LTE-A (Long Term Evolution-Advanced)

LTE ονομάζεται το ευρέως διαδεδομένο δίκτυο κινητής επικοινωνίας 4G το οποίο προσφέρει υψηλό ρυθμό μεταφοράς δεδομένων στις κινητές συσκευές. Το LTE αποτελεί μέρος μιας σειράς πρωτοτύπων που ονομάζονται eMBMS (**evolved multimedia broadcast multicast service**) και κατάφερε να αυξήσει την απόδοση των ασύρματων δικτυών έως και πενήντα φορές [43]. Το πρότυπο αποτελεί ένα δίκτυο μοναδικής συχνότητας. Μερικές από τις πιο διαδεδομένες χρήσεις του πρωτοκόλλου αφορούν την ζωντανή αναμετάδοση γεγονότων, την ροή τηλεοπτικών μέσων σε πραγματικό χρόνο καθώς κι την συνεχή ενημέρωση καιρικών συνθηκών, νεών κ.α.

5G

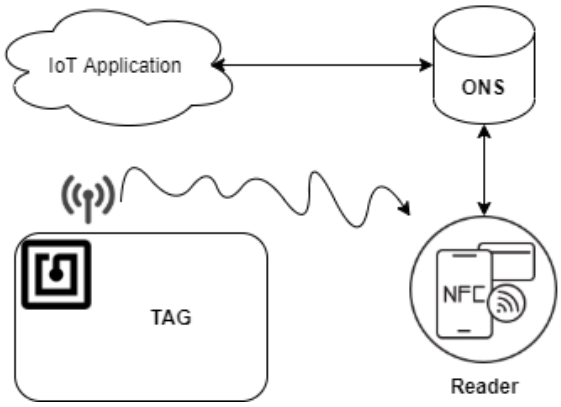
Το 5G αποτελεί το νεότερο πρότυπο επικοινωνίας των κινητών συσκευών και η πλήρης εφαρμογή του αναμένεται να αντιμετωπίσει κυρίως τρεις ανάγκες. Η πρώτη είναι η μείωση του χρόνου καθυστέρησης (latency) της μεταφοράς των δεδομένων σε λιγότερο από ένα δευτερόλεπτο. Η δεύτερη ανάγκη που αναμένεται να καλυφθεί είναι η συνεχόμενη ροή δεδομένων με τουλάχιστον 1 Gbps σε δεκάδες χιλιάδες ταυτόχρονα συνδεδεμένων χρηστών. Και τελικά το πρωτόκολλο 5G θα ενισχύσει σημαντικά την βελτίωση της ενεργειακής απόδοσης των συσκευών [44].

Από το παραπάνω γίνεται κατανοητό το 5G θα ενισχύσει το IoT σε αρκετούς τομείς [45]. Ο πρώτος είναι η αυτοματοποίηση στα εργοστάσια όπου απαιτείται συνεχής έλεγχος των μηχανημάτων σε πραγματικό χρόνο και με μηδενικές καθυστερήσεις. Ένας δεύτερος τομέας είναι τα smart grids όπου οι απαιτήσεις σε αξιοπιστία και σε μηδενικούς χρόνους καθυστέρησης είναι ακόμη μεγαλύτερες. Στο προηγούμενο τομέα εντάσσεται η αυτόνομη οδήγηση οχημάτων και η βελτιστοποίηση της οδικής κυκλοφορίας όπου το ένα δευτερόλεπτο αποτελεί κρίσιμο παράγοντα αποφυγής ατυχήματος.

EPCglobal

Το πρωτόκολλο αυτό κατασκευάστηκε για την ενορχήστρωση διαδικασιών μετάδοσης δεδομένων από συσκευές που βρίσκονται σε κοντινή απόσταση και αποτελούν μοναδικό κρίκο σε μια εφοδιαστική αλυσίδα [46]. Οι διαφορετικές διαδικασίες που επιτελεί το EPCglobal είναι η ανάγνωση, διαχείριση και ο έλεγχος των δεδομένων που «φέρει» η συσκευή. Οι διάφορες αλλαγές στα δεδομένα της κάθε συσκευής γίνονται με την χρήση EPCglobal Electronic Product Code Information Services (EPCIS). Τα EPCIS ουσιαστικά αποτελούν ποικίλες διαδικτυακές υπηρεσίες (web services) που εφαρμόζουν συμφωνημένα σχήματα και διεπαφές.

Προκειμένου να συμβούν τα παραπάνω κάθε συσκευή πρέπει να ανταποκρίνεται σε ορισμένα χαρακτηριστικά. Αρχικά μια συσκευή ή και προϊόν της εφοδιαστικής αλυσίδας πρέπει να διαθέτει έναν κωδικό που ονομάζεται EPC. Αυτός ο κωδικός αποτελεί μια μοναδική και αναγνωριστική δυαδική ακολουθία η οποία ενσωματώνεται σε μια RFID ετικέτα, ακολούθους καθίσταται δυνατή η ταυτοποίηση των αντικειμένων με την χρήση συσκευών αναγνώρισης RFID ετικετών. Η κάθε RFID ετικέτα αποτελείται από δυο κυρίως μέρη ένα ηλεκτρονικό τσιπ όπου αποθηκεύεται ο EPC κι μια κεραία που επιτρέπει την επικοινωνία με άλλες συσκευές μέσω της ανάγνωσης ετικετών. Η επικοινωνία αυτή επιτυγχάνεται μέσω κυμάτων ραδιοσυχνοτήτων. Τα δυο κυριότερα στοιχεία του RFID συστήματος αποτελούν οι κεραίες μετάδοσης ραδιοσυχνοτήτων (radio signal transporter) καθώς και οι συσκευές αναγνώρισης ετικετών (tag reader). Ο tag reader αναγνωρίζει τον κωδικό που αποστέλλει μια ετικέτα μέσω ραδιοκυμάτων. Στην συνέχεια η συσκευή αναγνώρισης μεταδίδει τον κωδικό της ετικέτας σε μια εφαρμογή. Η οποία ονομάζεται object-naming services (ONS) και έχει την δυνατότητα διεκπεραίωσης διαφόρων ενεργειών όπως η αναζήτηση του κατασκευαστή, η ανάγνωση της τυχόν περιγραφής.



Σχήμα 8 RFID

Z-Wave

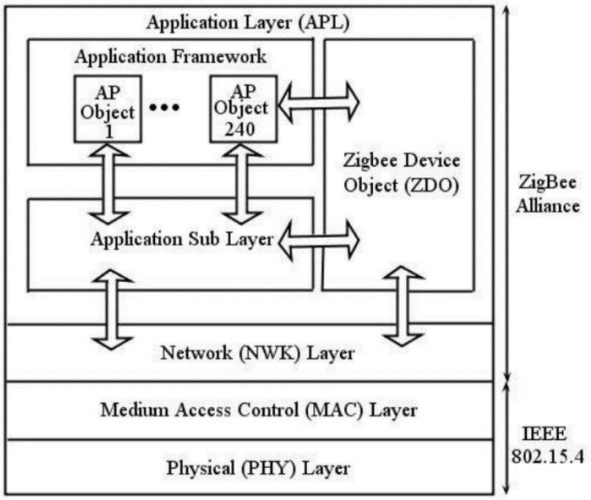
Το Z-Wave σχεδιάστηκε από την ZenSys, μετέπειτα βελτιώθηκε από την Z-Wave alliance και αποτελεί ένα πρότυπο ασύρματης επικοινωνίας με ελάχιστες απαιτήσεις σε ενέργεια. Για τον παραπάνω λόγο χρησιμοποιείται κυρίως σε οικιακά δίκτυα. Ένας παραπάνω λόγος της επέκτασης του πρωτόκολλου είναι ότι έχει την δυνατότητα να αναμεταδίδει τις διάφορες εντολές προσπερνώντας διάφορα εμπόδια όπως τοίχους και ορόφους μέσω της χρήσης ενδιάμεσων κόμβων για την δρομολόγηση των πακέτων. Το Z-Wave συνήθως λειτουργεί στην συχνότητα των 900 MHz, ακολουθεί τοπολογία πλέγματος όπου κάθε κόμβος μπορεί να λαμβάνει και να αποστέλλει δεδομένα.

Τα κύρια «συστατικά» του δικτύου είναι οι ελεγκτές (controllers) και οι εξαρτημένοι κόμβοι (slave nodes). Οι ελεγκτές έχουν την δυνατότητα δημιουργίας πινάκων δρομολόγησης και υπολογισμού της διαδρομής των πακέτων στο δίκτυο. Ακόμα οι ελεγκτές χωρίζονται σε πρωτεύοντες και δευτερεύοντες. Οι πρώτοι διατηρούν την συνολική «εικόνα» του δικτύου και ελέγχουν τα δεδομένα που εξέρχονται από το δίκτυο. Επίσης κατά την διαδικασία της εγγραφής ενός κόμβου στο Z-Wave δίκτυο οι πρωτεύοντες ελεγκτές του αναθέτουν ένα μοναδικό αναγνωριστικό. Σε κάθε Z-Wave δίκτυο υπάρχει μόνο ένας πρωτεύων ελεγκτής. Οι δεύτεροι περιέχουν τα αναγνωριστικά που έχουν δοθεί στις συσκευές και τον πίνακα δρομολόγησης.

Μερικές εφαρμογές «έξυπνου σπιτιού» που κάνουν χρήση του Z-Wave είναι οι έξυπνες λάμπες, κλειδαριές – πόρτες και θερμοστάτες.

ZigBee

Το ZigBee αρχικά σχεδιάστηκε το 1998 από την ZigBee alliance, αποτελεί ιδανικό πρότυπο για συσκευές με περιορισμένους ενεργειακούς πόρους (υποστήριξη λειτουργίας ύπνου στις συσκευές). Ακόμα το πρότυπο διακρίνεται για το χαμηλό κόστος, για την δυνατότητα υποστήριξης μεγάλων δικτύων με πάνω από 65.000 κόμβους [47] και για την τοπολογία πλέγματος (mesh topology) που λειτουργεί πάνω από το IEEE 802.15.4. Το τελευταίο βοηθά ενισχύει την ικανότητα του πρωτοκόλλου να επαναπροσδιορίζει διαδρομές που αποτυγχάνουν να συνδεθούν. Τα παραπάνω σε συνδυασμό με ειδικούς μηχανισμούς δρομολόγησης καθιστούν ένα δίκτυο ZigBee ικανό να καλύπτει 100 τετραγωνικά χιλιόμετρα. Οι ειδικοί μηχανισμοί δρομολόγησης, που συντελούν στο παραπάνω γεγονός, παρέχουν υπηρεσίες επανεμετάδοσης πακέτων μέσω των κόμβων του ZigBee δικτύου προς το κέντρο ελέγχου του δικτύου (Alexandrov & Monov, 2015).



Σχήμα 9 ZigBee architecture [48]

Από όλα τα παραπάνω πρωτόκολλα το ποιο θα είναι ιδανικό για την εφαρμογή που επιθυμεί να δημιουργήσει ο καθένας θα αποφασιστεί από τον ίδιο τον «αρχιτέκτονα» του δικτύου μιας και είναι αυτός γνωρίζει τις ανάγκες των συσκευών του δικτυού. Ο πιο σημαντικός παράγοντας στην επιλογή του πρωτοκόλλου είναι η κατανάλωση ενέργειας μιας και συχνά στα IoT περιβάλλοντα συναντάμε συσκευές που διαθέτουν ελάχιστα αποθέματα ενέργειας αλλά και έχουν μικρή επεξεργαστική ισχύς.

Ασφάλεια IoT συσκευών

Εισαγωγή

Η ασφάλεια σε όλα τα πληροφοριακά συστήματα ανεξαιρέτως αποτελεί ένα από τα πιο ζωτικά ζητήματα της λειτουργίας αυτών. Το παραπάνω γεγονός ενισχύετε στο IoT μιας και αυτό αποτελείται από διαφορετικές τεχνολογίες και ποίκιλα πληροφοριακά συστήματα (όπως οι βάσεις δεδομένων, cloud servers, mobile devices, διάφορους αισθητήρες και μικροελεγκτές). Αρκετή έμφαση πρέπει να δοθεί και στην ασφάλεια των συστημάτων που λειτουργούν με περιορισμένους πόρους (είτε λόγο ενέργειας είτε λόγο υπολογιστικής ισχύς). Από την προηγούμενη πρόταση σε συνδυασμό με την γνώση ότι η ‘διατήρηση’ της ασφάλειας των συστημάτων απαιτεί αρκετή υπολογιστική ισχύ οδηγούμαστε στο συμπέρασμα ότι οι επιλογές για την ασφάλεια των δεδομένων των εν λόγο συστημάτων μειώνονται δραματικά.

Ωστόσο για να φτάσουμε στο σημείο όπου θα μπορούμε να αντιληφθούμε τους διαθέσιμους μηχανισμούς ασφάλειας σε IoT συσκευές χρειάζεται να κατανοήσουμε τους βασικούς μηχανισμούς διατήρησης της ασφάλειας. Αυτοί είναι οι CIA (Confidentiality, Integrity, Availability) καθώς και το AAA Framework (Authentication, Authorization, Accounting).

Προυποθέσεις Ασφάλειας (Security Requirements)

Στην συνέχεια λοιπόν θα επιχειρήσουμε να αναλύσουμε συνοπτικά τους βασικότερους μηχανισμούς ασφάλειας. Εκτός από τους προαναφερόμενους (CIA & AAA Framework) θα αναφέρουμε την λειτουργία των Defense-in-Depth (Ασφάλεια σε βάθος), Κρυπτογράφηση και του Trusted Computing Base (TCB).

CIA (Confidentiality, Integrity, Availability)

Η εμπιστευτικότητα (Confidentiality), η ακεραιότητα (Integrity) και η διαθεσιμότητα (Availability) των δεδομένων αποτελούν βασικά στοιχεία που χρειάζεται να τηρούνται τόσο κατά το στάδιο της δημιουργίας ενός πληροφοριακού συστήματος όσο και για το στάδιο της ανάπτυξης αυτών.

Σχήμα 10 CIA (Confidentiality, Integrity, Availability)

Εμπιστευτικότητα (Confidentiality)

Σύμφωνα με την αρχή της εμπιστευτικότητας πρέπει να εξασφαλίζεται ότι μόνο οι εξουσιοδοτημένοι χρήστες θα έχουν πρόσβαση στις πληροφορίες που τους αφορούν.

Ακεραιότητα (Integrity)

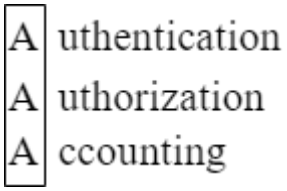
Η ακεραιότητα αποτελεί τον μηχανισμό που δεν επιτρέπει σε μη εξουσιοδοτημένους χρήστες να τροποποιούν με οποιονδήποτε τρόπο δεδομένα. Υπάρχουν τρεις τρόποι τροποποίησης της πληροφορίας η αλλαγή (update), η προσθήκη (write) και η διαγραφή (delete).

Διαθεσιμότητα (Availability)

Η διαθεσιμότητα εξασφαλίζει πως μόνο οι εξουσιοδοτημένοι χρήστες θα έχουν πρόσβαση στην «ακέραια» πληροφορία την στιγμή που θέλουν. Με άλλα λόγια ο μηχανισμός αυτός αφορά την περίπτωση της άρνησης στην προσπέλαση της ορθής πληροφορίας από εξουσιοδοτημένους χρήστες.

AAA Framework (Authentication, Authorization, Accounting)

Το AAA framework αποτελεί μια αρχιτεκτονική που ρυθμίζει με συνέπεια τρεις ανεξάρτητους μηχανισμούς ασφάλειας. Οι τρεις μηχανισμοί αυτοί είναι η αυθεντικοποίηση (Authentication), η εξουσιοδότηση (Authorization) και η χρέωση (Accounting).



Σχήμα 11 AAA Framework

Αυθεντικοποίηση (Authentication)

Η αυθεντικοποίηση αποτελεί την διαδικασία κατά την οποία εξασφαλίζεται ότι τα διαπιστευτήρια που παρέχει ο χρήστης στο πληροφοριακό σύστημα είναι έγκυρα. Με αυτήν την διαδικασία εξασφαλίζεται ότι οι χρήστες με μη έγκυρα διαπιστευτήρια δεν θα αποκτήσουν πρόσβαση σε οποιαδήποτε «προστατευόμενο» δεδομένο. Η πιο διαδομένη και εύκολη στην υλοποίηση τεχνική αυθεντικοποίησης είναι το όνομα χρήστη και ο κωδικός (username & password). Ωστόσο αυτή η τεχνική ελλοχεύει αρκετούς κινδύνους, όπως οι «αδύναμοι» κωδικοί, brute-force attacks, phishing κ.α.. Προκειμένου να ενισχύσουμε την ασφάλεια κατά την αυθεντικοποίηση μπορούμε να χρησιμοποιήσουμε άλλες τεχνικές αυθεντικοποίησης όπως η αυθεντικοποίηση δυο παραγόντων όπου πέρα από τον κωδικό ο χρήστης απαιτείτε να εισάγει ένα κωδικό μιας χρήσης, ένα βιομετρικό του στοιχείο ή και την αναγνώριση μιας συγκεκριμένης συσκευής (συνήθως usb stick).

Εξουσιοδότηση (Authorization)

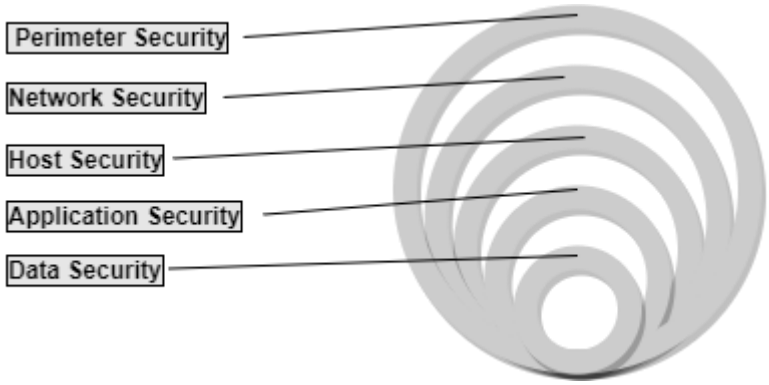
Η εξουσιοδότηση αποτελείται από το μηχανισμό που εξασφαλίζει ότι ένα συγκεκριμένο δεδομένο θα μπορεί να προσπελασθεί από συγκεκριμένους χρήστες, οι οποίοι θα μπορούν να πραγματοποιήσουν συγκεκριμένες ενέργειες στο δεδομένο. Το παραπάνω επιτυγχάνεται εφόσον δοθούν στο κατάλληλο τύπο χρηστών τα κατάλληλα δικαιώματα με βάση τον ρόλο τους. Υπάρχουν δυο είδη δικαιωμάτων που μπορούν να εκχωρηθούν σε χρήστες και αυτά είναι είτε μόνο ανάγνωση (Read-only) είτε ανάγνωση και εγγραφή (Read and write). Τελικά δημιουργείται ένας πίνακας που ονομάζεται λίστα ελέγχου πρόσβασης (Access Control List “ACL”), αυτός ο πίνακας περιέχει τα διάφορα δικαιώματα που έχουν διαφορετικοί χρήστες σε διαφορετικά δεδομένα.

Χρέωση (Accounting)

Η χρέωση αφορά τις διαδικασίες που διεκπεραιώνονται σποραδικά προκειμένου να ελεγχθεί η αποτελεσματικότητα των εφαρμοσμένων μηχανισμών ασφάλειας σε ένα υπολογιστικό σύστημα. Η διαδικασία της χρέωσης επιτυγχάνεται από την ανάλυση των αρχείων καταγραφής (audit logs) της δραστηριότητας των διαφορετικών χρηστών.

Ασφάλεια σε βάθος (Defense-in-Depth)

Με τον όρο Defense-in-Depth εννοούμε το σύνολο των πολιτικών που εξασφαλίζουν την ύπαρξη πολλών επιπέδων ασφάλειας σε ένα πληροφοριακό σύστημα. Ο κύριος ρόλος του μηχανισμού είναι ότι σε περίπτωση παραβίασης ή αποτυχίας ενός επιπέδου ασφαλείας τότε ο «επιτιθέμενος» δεν θα έχει την δυνατότητα να εισχωρήσει σε άλλα επίπεδα. Οπότε είναι φανερό πως εφαρμόζοντας αυτόν τον μηχανισμό σε ένα πληροφοριακό σύστημα όχι μόνο κερδίζουμε πολύτιμο χρόνο, όπου μπορούμε να εντοπίσουμε τον «επιτιθέμενο», ταυτοχρόνως εξασφαλίζουμε την ασφάλεια όλων των συστημάτων που βρίσκονται σε διαφορετικό επίπεδο ασφάλειας από το «παραβιασμένο».



Σχήμα 12 Defense-in-Depth

Κρυπτογράφηση (encryption)

Η κρυπτογράφηση αποτελεί την διαδικασία κατά την οποία τα δεδομένα μετατρέπονται με κάποιο αλγόριθμο κρυπτογράφησης σε μια ακατανόητη μορφή που δύσκολα θα παραβιαστεί ή θα παραμορφωθεί. Άρα μόνο ο εξουσιοδοτημένος χρήστης θα μπορέσει να προσπελάσει την πληροφορία που συνθέτουν τα κρυπτογραφημένα δεδομένα. Η διαδικασία που περιγράφεται στην προηγούμενη πρόταση, δηλαδή η μετατροπή των κρυπτογραφημένων δεδομένων στην αρχική τους μορφή, ονομάζεται αποκρυπτογράφηση των δεδομένων και εξασφαλίζει ότι κάποιος μη εξουσιοδοτημένος χρήστης δεν θα έχει την δυνατότητα ανάγνωσης και τροποποίησης των δεδομένων. Οι αλγόριθμοι κρυπτογράφησης χωρίζονται σε δυο κύριες κατηγορίες. Η πρώτη ονομάζεται συμμετρική κρυπτογραφία (Symmetric Encryption) και χρησιμοποιεί το ίδιο κλειδί για τις διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης. Και η δεύτερη κατηγορία λέγεται ασύμμετρη κρυπτογράφηση (Asymmetric Encryption) και χρησιμοποιεί άλλο κλειδί για την κρυπτογράφηση και διαφορετικό για την αποκρυπτογράφηση. Ακόμα υπάρχουν και υδριβικοί αλγόριθμοι κρυπτογράφησης που συνδυάζουν τις παραπάνω δυο μεθόδους.

Μερικοί από τους πιο γνωστούς αλγόριθμους κρυπτογράφησης είναι ο Rivest-Shamir-Adleman (RSA), ο Data Encryption Standard (DES),ο οποίος δεν θεωρείται πλέον ασφαλής οπότε και χρησιμοποιείται ο Triple DES (3DES). Άλλοι αλγόριθμοι είναι ο Advanced Encryption Standard (AES), ο Blowfish, ο RC2 και ο RC6. Οι προηγούμενοι αλγόριθμοι κρυπτογράφησης είναι οι σημαντικότεροι στην εξασφάλιση της ασφάλειας σε ένα πληροφοριακό σύστημα ωστόσο απαιτούν εξαιρετικά μεγάλα αποθέματα από επεξεργαστική ισχύ, μνήμη καθώς και ενέργεια. Οπότε δεν αποτελούν την ενδεδειγμένη λύση σε όλες τις εφαρμογές του διαδικτύου των πραγμάτων. Προκειμένου να αντιμετωπιστεί το παραπάνω πρόβλημα βρέθηκαν αλγόριθμοι που θα προσαρμοστούν στις ανάγκες των IoT συσκευών. Ορισμένα παραδείγματα «ελαφρών» (lightweight) αλγορίθμων κρυπτογράφησης αποτελούν ο DESL, μια ελαφριά έκδοση του DES, ο DESXL μια ελαφριά έκδοση του DESL, ο Curupira, ο Katan & Ktantan, ο Present, ο Hummingbird, ο Simon and Speck, ο LED (Light Encryption Device) κ.α. [49].

Trusted Computing Base (TCB)

Ουσιαστικά ο ορός trusted computing base αναφέρεται στα πιο «ευαίσθητα» σημεία ενός υπολογιστικού συστήματος, εφόσον εκτεθούν αυτά σε κάποιο κίνδυνο τότε διακυβεύεται η εύρυθμη λειτουργία ολόκληρου του συστήματος. Γίνεται, λοιπόν, εύκολα κατανοητό ότι πρέπει να δοθεί η δέουσα προσοχή κατά την διάρκεια του ορισμού των ευαίσθητων σημείων

του συστήματος. Ακόμη πρέπει να έχουμε στο νου μας ότι το IoT αποτελείται από μια ευρεία γκάμα από συσκευές, γεγονός που αυξάνει εκθετικά την δυσκολία του ορισμού των ευαίσθητων.

Απειλές Ασφάλειας

Με εφόδιο την προηγούμενη ανάλυση των κύριων μηχανισμών της ασφάλειας των υπολογιστικών συστημάτων έχουμε την δυνατότητα να κατανοήσουμε τις διαφορετικές ευπάθειες αυτών. Κάθε μια ευπάθεια πρέπει να αντιμετωπίζεται άμεσα διότι ενδέχεται να μετατραπεί σε πραγματική «απειλή» της ασφάλειας του συστήματος.

DoS (Denial of Service)

Η πρώτη απειλή που θα μελετήσουμε είναι η άρνηση της εξυπηρέτησης της υπηρεσίας (Denial of Service) αφού αποτελεί ίσως και την πιο διαδομένη επίθεση. Όπως αναφέρεται και στην ονομασία της εν λόγω απειλής το υπολογιστικό σύστημα δεν μπορεί να εξυπηρετήσει τους εξουσιοδοτημένους χρήστες αφού οι πόροι του δέχονται «επίθεση» κατά την οποία ένας χρήστης υπερ-φορτώνει μαζικά το σύστημα [50]. Όταν μια τέτοια επίθεση γίνεται από διαφορετικούς «χρήστες» τότε ονομάζεται κατανεμημένη άρνηση της εξυπηρέτησης της υπηρεσίας (Distributed Denial of Service). Οι απλές μέθοδοι τέτοιας επιθέσεις είναι η X-DoS (XML based) και η H-DoS (HTTP based).

Unauthorized subscription/publication

Η απειλή της μη εξουσιοδοτημένης εγγραφής-δημοσίευσης πραγματώνεται όταν ένας μη εξουσιοδοτημένος χρήστης εγγράφει δεδομένα στο υπολογιστικό σύστημα. Αυτή η διαδικασία ενδέχεται να προκαλέσει αρκετά μεγάλο πρόβλημα αφού εκτός από την τροποποίηση της πληροφορίας μπορεί να οδηγήσει σε επίθεση DoS ή και σε άλλες επιθέσεις. Ακόμη αυτού του είδους επίθεση συμβαίνει ανάμεσα στην επικοινωνία του publisher με ένα broker, ενός broker με άλλον broker, ενός broker με έναν subscriber καθώς και ενός publisher με έναν subscriber.

Πλαστογράφηση (Spoofing)

Η πλαστογράφηση (spoofing) αποτελεί την «κλοπή» της ταυτότητας ενός εξουσιοδοτημένου χρήστη. Εφόσον η απειλή του spoofing δεν αντιμετωπιστεί αμέσως ο επιτιθέμενος έχει την δυνατότητα να προχωρήσει σε άλλες απειλές όπως η man-in-the-middle, το routing redirect, source routing, blind spoofing, gps ή e-mail spoofing καθώς και το flooding.

Big Data Threats

Η ανάγκη χρήσης των δεδομένων μεγάλης κλίμακας (Big Data) φέρνει σαν συνεπακολουθούμενο την ανάγκη της αντιμετώπισης των δικών τους απειλών. Η μεγαλύτερη απειλή των big data είναι η τροποποίηση των αλγορίθμων παραγωγής των δεδομένων από μη εξουσιοδοτημένα άτομα. Με τον τρόπο που παράγονται τα δεδομένα μεγάλης κλίμακας είναι εξαιρετικά δύσκολο να εντοπιστεί ένας host ο οποίος παράγει-αποστέλλει λανθασμένα δεδομένα.

Ένα ακόμα σοβαρό ζήτημα που προκύπτει από την χρήση των δεδομένων μεγάλης κλίμακας είναι η αξιολόγηση της πηγής που αποστέλλει τα δεδομένα (data source validation). Ειδικά στον συνδυασμό του IoT με τα big data όπου οι πηγές είναι πολύ διαφορετικές και ανομοιογενείς μεταξύ τους η αξιολόγηση τους δυσκολεύει.

Ακόμα τα δεδομένα μεγάλης κλίμακας απαιτούν την χρήση μη σχεσιακών βάσεων δεδομένων. Το παραπάνω γεγονός μεταφέρει τους χειρισμούς ασφάλειας των βάσεων από τον «μηχανισμό» της κάθε βάσης στον σχεδιαστή της βάσης. Επομένως ο σχεδιαστής της βάσης καλείται να είναι πολύ προσεκτικός τόσο κατά την σχεδίαση και ανάπτυξη της βάσης όσο και κατά την «συντήρηση» αυτής.

Όσο ο όγκος των δεδομένων αυξάνεται εκθετικά τόσο αυξάνεται και η δυσκολία ασφαλούς αποθήκευσης και διαχείρισης αυτών. Οπότε απαιτείται να έχουμε μια προσέγγιση κλιμάκωσης των δεδομένων σε επίπεδα (storage tiering) έτσι ώστε η διαχείριση τους να γίνεται πιο εύκολη. Σε αυτήν την περίπτωση ο προγραμματιστής καλείται να διαβαθμίσει τα δεδομένα, είτε χειροκίνητα είτε με αυτοματοποιημένες διαδικασίες, με κύριο κριτήριο την συχνότητα της χρήσης τους. Σε αυτό το σημείο αξίζει να σημειωθεί ότι τα επίπεδα αυτά διαφέρουν μεταξύ τους σε κόστος, απόδοση, χωρητικότητα καθώς και τους μηχανισμούς ασφαλείας τους. Όποτε εάν δεν δοθεί η δέουσα προσοχή σε κάθε επίπεδο υπάρχει ο κίνδυνος να παρουσιαστούν κενά ασφαλείας τα οποία ενδέχεται να μετατραπούν σε απειλές για το σύστημα.

Τέλος όταν αναφερόμαστε στα big data χρειάζεται να σκεφτόμαστε και την ιδιωτικότητα των χρηστών. Ένα πολύ συχνό σφάλμα που ενδέχεται να απειλήσει και το σύστημα είναι η πρόσβαση τρίτων και μη αξιόπιστων εφαρμογών στα δεδομένα που έχουν συλλεχθεί και αφορούν κυρίως την δραστηριότητα των χρηστών στο σύστημα. Ενώ τα δεδομένα αυτά προσφέρουν καλύτερη εμπειρία στον χρήστη μπορούν και να τον βλάψουν.

Cloud Threats

Το περιβάλλον του IoT απαιτεί την χρήση του υπολογιστικού νέφους (cloud computing) όπου και αυτό μεταφέρει τις δικές του απειλές. Για να κατανοήσουμε τις απειλές αυτές πρέπει να γνωρίζουμε πως το νέφος αποτελείται τρία διαφορετικά επίπεδα, τα οποία είναι το επίπεδο της εφαρμογής (application layer), το επίπεδο της υπηρεσίας (platform layer) και το επίπεδο της υποδομής (infrastructure layer). Το κάθε επίπεδο αποτελείται από διαφορετικές ευπάθειες. Οι λύσεις αυτών θα δοθούν παρακάτω.

Ωστόσο οι μεγαλύτερες απειλές του cloud εντοπίζονται στην χρήση της τεχνολογίας multitenancy (application layer). Με τον όρο multitenancy (πολύ-μίσθωση) εννοούμε πως για την παροχή της ενοικιαζόμενης υπηρεσίας στους χρήστες αξιοποιούνται ίδιοι υλικοί πόροι. Αυτό επιτυγχάνεται από την εικονικοποίηση (virtualization) των πόρων αυτών. Επομένως δημιουργούνται ρίσκα τόσο για εμπιστευτικότητα όσο και για την ακεραιότητα των δεδομένων του υπολογιστικού συστήματος μιας και σε περίπτωση παραβίασης της ασφάλειας αυτού ο «επιτιθέμενος» έχει την δυνατότητα να διαχειριστεί εικονικούς πόρους άλλων χρηστών.

VM Threats

Η εικονικοποίηση της υποδομής όπως αναφέραμε από πάνω εγκυμονεί αρκετούς κινδύνους, οπότε χρειάζεται να τους μελετήσουμε όλους αναλυτικά εφόσον χρησιμοποιούμε εικονικά μηχανήματα (virtual machines).

Μια από τις πιο κρίσιμες απειλές της εικονικοποίησης είναι αυτή που ονομάζεται (κλοπή εικονικού μηχανήματος). Αυτή η απειλή πραγματώνεται όταν κάποιος αντιγράφει ή μετακινεί ένα εικονικό μηχάνημα σε ένα σημείο χωρίς να έχει καμία εξουσιοδότηση για αυτήν την ενέργεια. Με αυτήν την διαδικασία ο hacker ενδέχεται να αποκτήσει πρόσβαση σε απόρρητα δεδομένα άλλων χρηστών.

Μια παρόμοια απειλή με αυτήν της VM Theft είναι και η VM Escape (απόδραση εικονικού μηχανήματος). Ουσιαστικά αυτή η απειλή περιγράφει την διαδικασία κατά την οποία ένα εικονικό μηχάνημα «ξεφεύγει» από τον έλεγχο του "επόπτη" (hypervisor) και στην συνέχεια επηρεάζει την λειτουργία του. Επιπλέον το εικονικό μηχάνημα που έχει ξεφύγει από τον επόπτη δεν περιορίζεται από ελέγχους ασφάλειάς που υπάρχουν και μπορεί να ελέγξει τα υπόλοιπα εικονικά μηχανήματα που επηρεάζει ο hypervisor.

Άλλη μια απειλή κατά την οποία ένα παραβιασμένο εικονικό μηχάνημα ενδέχεται να επηρεάσει την λειτουργία των υπολοίπων που λειτουργούν στο ίδιο φυσικό υπολογιστικό μηχάνημα ονομάζεται Instant-On Gaps (άμεσο κενό). Ένα κενό, το οποίο μπορεί να οδηγήσει στην πραγμάτωση αυτής της απειλής, 'γεννιέται' κατά τις διαδικασίες πρόσθεσης, αφαίρεσης αλλά ακόμα και χρήσης των εικονικών μηχανημάτων. Αφού αυτός ο κύκλος επαναλαμβάνεται συχνά μπορεί υπάρξει ένα VM όπου για κάποιο λόγο παρεκκλίνει από τις εφαρμοσμένες τεχνικές ασφάλειας και οδηγήσει στην ύπαρξη τρωτών σημείων. Τις ευπάθειες αυτές λοιπόν μπορεί να τις εκμεταλλευτεί ένας επιτιθέμενος για να δημιουργήσει νέα 'ευάλωτα' εικονικά μηχανήματα τα οποία στην συνέχεια θα επηρεάσουν την συνολική φυσική υποδομή.

Μια ακόμα σοβαρή απειλή των εικονικών μηχανών ονομάζεται Hyperjacking. Η hyperjacking ουσιαστικά αποτελεί την εγκατάσταση ενός νέου hypervisor ή την υπονόμευση του προ-υπάρχοντος, ο επιτιθέμενος με αυτόν τον τρόπο παίρνει τον απόλυτο έλεγχο από της φυσικής υποδομής. Εφόσον έχουν συμβεί τα προηγούμενα η ευπάθεια αυτή δεν είναι εύκολο να ανακαλυφθεί, μιας και μπορεί να χαρακτηριστεί ως ευπάθεια επιπέδου rootkit. Όπως λοιπόν γνωρίζουμε τέτοιες ευπάθειες δεν είναι δυνατόν να ανακαλυφθούν από το λογισμικό και ο hacker διατηρεί πρόσβαση επιπέδου administrator έχοντας παρακάμψει όλους τους μηχανισμούς αυθεντικοποίησης και εξουσιοδότησης που παρέχει το λειτουργικό σύστημα.

Φυσικά δεν πρέπει να ξεχνάμε και την απειλή που ονομάζεται Inter-VM attack. Αυτή η απειλή πραγματώνεται όταν ένα εικονικό μηχάνημα παραβιάζεται και στην συνεχεία ο επιτιθέμενος παραβιάζει και αλλά virtual machines που φιλοξενούνται στον ίδιο φυσικό μηχάνημα.

Τέλος τα εικονικά μηχανήματα μπορούν να παρουσιάσουν και ευπάθειες ακόμα και όταν είναι κλειστά. Η αιτία του παραπάνω ζητήματος είναι ότι όταν βρίσκονται σε ανενεργή κατάσταση τότε δεν λαμβάνουν τις ενημερώσεις ασφαλείας καθώς και ότι τότε είναι διαθέσιμα ως αρχεία 'στιγμιότυπων' εικονικών μηχανημάτων (VM image files). Επομένως τα αρχεία αυτά εφικτό να τροποποιηθούν από κάποιο malware ή ακόμα και αντιγραφούν από κάποιον μη εξουσιοδοτημένο χρήστη. Το τελευταίο βέβαια προϋποθέτει ότι ορισμένοι μηχανισμοί ασφαλείας (θα μελετηθούν παρακάτω) δεν εφαρμόζονται. Αύτη η απειλή ονομάζεται μη εξουσιοδοτημένη τροποποίηση των αρχείων 'στιγμιότυπων' των εικονικών μηχανών (unauthorized alteration of virtual machine image files). Αυτή η ευπάθεια ενδέχεται να δημιουργηθεί και κατά την διάρκεια της μετακίνησης (migration) ενός εικονικού μηχανήματος εφόσον υπάρξουν επιθέσεις όπως η eavesdropping.

IoT Malware

Με τον όρο malware εννοούμε τα προγράμματα που εγκαθίστανται και λειτουργούν κακόβουλα χωρίς ο χρήστης να γνωρίζει την ύπαρξή τους καθώς κι τις λειτουργίες που αυτά επιτελούν. Τα πιο γνωστά malware είναι οι ιοί, trojans, worms, spyware και ransomware. Όλα τα προηγούμενα έχουν τα εξής κύρια χαρακτηριστικά: α) συγκεντρώνουν πληροφορίες από το πληροφοριακό σύστημα στο οποίο εκτελούνται, β) προσπαθούν να πολλαπλασιαστούν και να μολύνουν άλλα συστήματα που βρίσκονται στο ίδιο δίκτυο με το 'μολυσμένο', γ) ο επιτιθέμενος μπορεί να εκτελέσει ορισμένες εντολές απομακρυσμένα ή να πάρει τον απόλυτο έλεγχο του συστήματος.

Στο περιβάλλον του IoT δεν είναι εύκολο να αντιμετωπιστεί η απειλή των malwares [51]. Τον παραπάνω προβληματισμό μπορούμε να τον κατανοήσουμε διότι κάθε συνδεδεμένη συσκευή στο οικοσύστημα του IoT διαφέρει αρχιτεκτονικά, επομένως κάθε άκρο του συστήματος είναι αντιμέτωπη με μια ευρεία γκάμα από κακόβουλα λογισμικά. Ακόμα το IoT ενδέχεται να απειληθεί συνολικά από ένα malware. Στην προηγούμενη περίπτωση ο επιτιθέμενος αφού έχει μολύνει μια συσκευή που είναι συνδεδεμένη σε ένα IoT σύστημα τότε προσπαθεί να μολύνει όλα τα άκρα του δικτύου και να αποκτήσει τον έλεγχο των πιο ευάλωτων συσκευών. Το πρώτο malware που έχει απειλήσει IoT περιβάλλον είναι το Linux.Darlloz [52] και είχε ανακαλυφθεί το 2013. Το Linux.Darlloz στόχευε δρομολογητές, κάμερες ασφαλείας και set-top boxes (έξυπνες συσκευές πολυμέσων) και εφόσον ολοκληρωνόταν επιτυχώς η εγκατάσταση του στον στόχο ξεκίναγε το mining κρυπτονομισμάτων όπως το Mincoin και το Dogecoin.

Man-in-the-middle attack (MITM)

Ο επιτιθέμενος σε αυτήν την απειλή εισβάλει ανάμεσα στην επικοινωνία δυο υπολογιστικών συστημάτων και ελέγχει την ροή της επικοινωνίας τους, από την παραπάνω διαδικασία δικαιολογείται κι η ονομασία αυτής της απειλής. Ουσιαστικά αποτελεί επίθεση τύπου spoofing διότι ο επιτιθέμενος στέλνει δεδομένα υποδυόμενος κάποιον άλλον σε ένα άκρο του δικτύου. Αυτή η απειλή εάν γίνει σε ένα κρυπτογραφημένο δίκτυο ενδέχεται να φαίνεται ακίνδυνη όμως δεν είναι γιατί ο επιτιθέμενος μπορεί να ξεγελάσει κάποιον χρήστη ώστε ο τελευταίος να αποκαλύψει στον πρώτο απόρρητες πληροφορίες. Ακόμα αυτή η τύπου επίθεση συχνά χρησιμοποιείται προκειμένου ο hacker να ανακαλύψει κάποιο άλλο ευπαθής σημείο του συστήματος και να το εκμεταλλευτεί. Σύμφωνα με τα παραπάνω αυτή η απειλή μπορεί να επηρεάσει την εμπιστευτικότητα και την ακεραιότητα του υπολογιστικού συστήματος.

Replay attack

Η replay attack αποτελεί ουσιαστικά μια μορφή της απειλής man-in-the-middle attack όπου ο «επιτιθέμενος» λαμβάνει τα μηνύματα του πομπού και τα αναμεταδίδει υποδυόμενος τον πομπό σε επόμενη χρονική στιγμή. Ο στόχος του hacker μπορεί είναι η απόκτηση πρόσβασης σε απόρρητα δεδομένα μέσω της απάντησης του δέκτη ή και η επανάληψη κάποιας ενέργειας έτσι ώστε ο επιτιθέμενος να ανιχνεύσει και ύστερα να εκμεταλλευτεί κάποια άλλη ευπάθεια του συστήματος.

Φυσικές επιθέσεις (Physical attacks)

Οι κλασικές μέθοδοι των φυσικών επιθέσεων περιλαμβάνουν την καταστροφή του υλικού όπου λειτουργεί ένα πληροφοριακό σύστημα είτε ακούσια είτε εκούσια. Οι ακούσιες απειλές περιλαμβάνουν την αλλοίωση του υλικού από φυσικά φαινόμενα (πλημύρες, βροχές, φωτιές, φυσική φθορά, απότομη αύξηση της τάσης του ρεύματος, κ.α.) καθώς και από διάφορα ατυχήματα όπως πτώση υγρών. Ενώ οι εκούσιες ζημιές περιλαμβάνουν την πρόσβαση μη εξουσιοδοτημένου ατόμου στο υλικό (server room), καταστροφή του υλικού (πρόκληση πυρκαγιάς ή άλλων γεγονότων) καθώς και η κλοπή μέρους του υλικού.

Στο περιβάλλον του IoT μπορούμε να προσθέσουμε και την φυσική απειλή κατά την οποία ο επιτιθέμενος αποκτά νόμιμα μια συσκευή και στην συνέχεια την χρησιμοποιεί για να ανακαλύψει τρωτά σημεία στην προσφερόμενη υπηρεσία. Αυτό μπορεί να επιτευχθεί εάν ο hacker προβεί σε reverse engineering τεχνικές πάνω στην συσκευή, ακολούθως πραγματοποιήσει μια σειρά από δοκιμαστικές επιθέσεις έτσι ώστε να προχωρήσει σε κανονική επίθεση στο πληροφοριακό σύστημα.

Πρακτικές εξασφάλισης της ασφάλειας (Security Solutions)

DoS (Denial of Service)

Η επίθεση τύπου DoS [53] μπορεί να αντιμετωπιστεί με ποικίλους τρόπους. Η πιο απλή μέθοδος υλοποιείτε και από απλά router τα οποία ανιχνεύουν την πιθανώς κακόβουλη IP, στην συνέχεια την τοποθετούν σε μια «μαύρη λίστα» (blacklist) και τελικά απορρίπτουν κάθε αίτημα που προέρχεται από πιθανώς κακόβουλες IP διευθύνσεις. Επιπλέον οι δρομολογητές έχουν την δυνατότητα να απορρίπτουν πακέτα τα οποία έχουν μη επιτρεπτές τιμές στο πεδίο source address έτσι ώστε να μην είναι δυνατή η αποστολή πακέτων με «παραπλανητικές» διευθύνσεις αποστολής. Ακόμα οι δρομολογητές με την χρήση timestamp μπορούν να απορρίπτουν πακέτα που αποστέλλονται για πολλοστή φορά με την ίδια «χρονοσήμανση».

Όπως είδαμε πιο πάνω ένα router έχει περιορισμένες λειτουργίες άμυνας εναντίον επιθέσεων DoS ωστόσο η τοποθέτηση ενός firewall (software ή hardware) αυξάνει σημαντικά το πλήθος των διαθέσιμων λειτουργιών άμυνας. Το firewall μπορεί να «φιλτράρει» τα πακέτα που δέχεται και να τα απορρίπτει σε περίπτωση που δεν πληρούνται συγκεκριμένα κριτήρια. Επιπρόσθετα στην παραπάνω μέθοδο το τείχος προστασίας (firewall) αξιοποιώντας δυναμικές λίστες απόρριψης (blacklists) παραμένει ενημερωμένο για καινούργια nodes που τείνουν να στέλνουν πολλά μη επιτρεπτά αιτήματα προς το διαδίκτυο. Ενδεικτικά αναφέρονται μερικοί γνωστοί πάροχοι blacklist όπως οι access.redhawk.org, all.s5h.net, b.barracudacentral.org, bl.spamcop.net, bl.tiopan.com και blackholes.wirehub.net.

Τέλος υπάρχουν κι ορισμένα εργαλεία που προσφέρουν προστασία ενάντια σε DoS επιθέσεις τα εργαλεία αυτά συνδυάζουν τις παραπάνω τεχνικές. Ονομαστικά αναφέρονται ορισμένα τέτοια εργαλεία όπως είναι το imperva securesphere web application firewall, το AWS Shield, το Incapsula καθώς και το black lotus.

Πλαστογράφηση (Unauthorized subscription/publication & Spoofing)

Οι παραπάνω απειλές μπορούν να αντιμετωπιστούν μερικώς με τους τρόπους αντιμετώπισης των DoS επιθέσεων. Όμως στις περιπτώσεις των unauthorized subscription/publication και spoofing επιθέσεων ο αμυνόμενος πρέπει να ελέγχει κι εάν ο επιτιθέμενος διαθέτει τα κατάλληλα διαπιστευτήρια προκειμένου να προβεί σε

οποιαδήποτε ενέργεια για παράδειγμα τροποποίηση δεδομένων. Δηλαδή ο αμυνόμενος χρειάζεται να θέσει σε εφαρμογή τα κριτήρια του AAA framework (περιγράφεται παραπάνω).

Big Data Threats

Όπως αναλύσαμε προηγουμένως τα δεδομένα μεγάλης κλίμακας αντιμετωπίζουν αρκετούς κινδύνους. Επομένως ένα πληροφοριακό σύστημα που χρησιμοποιεί μόνο ένα μέρος των διαφορετικών μηχανισμών ασφαλείας δεν δύναται να χαρακτηριστεί ως ασφαλές αλλά απαιτείται το σύνολο των διαθέσιμων μηχανισμών εξασφάλισης της «ακεραιότητας» ενός συστήματος έτσι ώστε αυτό να διακρίνεται ως «αξιόπιστο».

Απόρροια λοιπόν του γεγονότος της μη εξασφαλισμένης «παραγωγής» των δεδομένων είναι ότι απαιτείται η χρήση ποικίλων αλγορίθμων οι οποίοι θα είναι ικανοί να εξασφαλίσουν την ποιότητα των «εισαγόμενων» στο σύστημα δεδομένων. Από την προηγούμενη πρόταση καταλαβαίνουμε ότι δεν γίνεται να υπάρξει ένα ενιαίο εργαλείο όπου θα πραγματώνει τους κατάλληλους αλγορίθμους σε όλα τα περιβάλλοντα IoT, έτσι απαιτείται η συμβολή αυτών που σχεδίασαν το πληροφοριακό σύστημα για να υλοποιηθούν οι κατάλληλοι αλγόριθμοι που θα λειτουργούν σαν φίλτρα στα εισερχόμενα δεδομένα.

Στο παραπάνω πρόβλημα πρέπει να προστεθεί και η αξιολόγηση της πηγής των δεδομένων όπου δύσκολα μπορεί να επιτευχθεί σε ένα IoT περιβάλλον. Η λύση σε αυτήν την απειλή είναι η αντιστοίχιση κάθε συσκευής-πομπού με ένα μοναδικό κλειδί και η απόρριψη όσων δεν έχουν αντιστοιχηθεί. Αυτή η αντιστοίχιση δεν είναι πάντα εφικτή λόγο της ανομοιογένειας των συσκευών που τροφοδοτούν τα IoT περιβάλλοντα οπότε οι διαχειριστές χρειάζεται να θέσουν ορισμένα άλλα κριτήρια για την απόρριψη ή όχι των πηγών.

Ένα επιπλέον πρόβλημα που θα πρέπει να έχουν στο νου τους οι σχεδιαστές του πληροφοριακού συστήματος είναι η διαχείριση των μη σχεσιακών βάσεων δεδομένων. Σε αυτό το θέμα πάλι δεν μπορεί να βρεθεί ένας μηχανισμός που θα είναι ίδιος σε όλα τα περιβάλλοντα IoT οπότε οι διαχειριστές του περιβάλλοντος είναι υπεύθυνοι για την συνεχή επιτήρηση και ανανέωση των κατάλληλων πολιτικών ασφαλείας που θα εξασφαλίζουν την απρόσκοπτη λειτουργία του συστήματος.

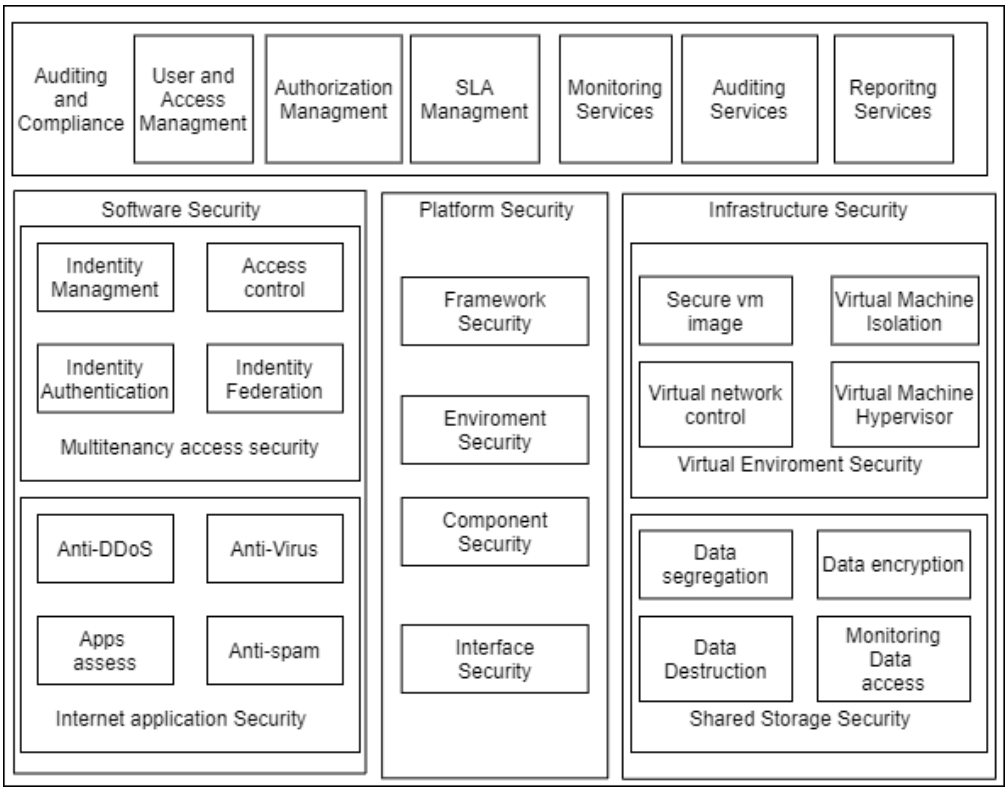
Η διαβάθμιση των δεδομένων (data tiering) όπως αναφέραμε προηγουμένως κρύβει αρκετούς κινδύνους. Οι διαθέσιμες πρακτικές για την καλύτερη προστασία της ιεραρχικής κατάτμησης των δεδομένων (πυραμίδα τα πιο σημαντικά στην κορυφή με μεγαλύτερη προστασία) εξαρτώνται από τους σχεδιαστές του συστήματος μιας και αφορούν την υλοποίηση πολιτικών που θα τρέχουν είτε αυτόματα από το σύστημα είτε μηχανικά από κάποιον διαχειριστή.

Cloud Threats

Οι απειλές που περιέχονται στην πολύ-μίσθωση των υποδομών μπορούν να λυθούν με τρεις τρόπους: τον διαχωρισμό των εικονικών μηχανών (VM segmentation), τον ενεργό κι διαρκή έλεγχο των εικονικών μηχανών (virtual machine introspection) και την κατάτμηση των βάσεων δεδομένων (database segmentation).

Οι απειλές που ελλοχεύουν τα διαφορετικά επίπεδα του υπολογιστικού νέφους μας αναγκάζουν να προσχωρήσουμε σε αυστηρή εφαρμογή ορισμένων μέτρων. Τα μέτρα αυτά μας προσφέρουν μεγάλη αξιοπιστία και προστασία του συστήματος. Το παρακάτω διάγραμμα θα βοηθήσει να κατανοήσουμε την λειτουργία των μηχανισμών ασφάλειας στα

τρία διαφορετικά επίπεδα που πρέπει να έχουμε υπόψιν μας στον σχεδιασμό cloud υλοποιήσεων.



Σχήμα 13 Cloud Security

VM Threats

Σε αυτό το σημείο θα μελετήσουμε τις μεθόδους προστασίας έναντι των απειλών που ελλοχεύει η χρήση των εικονικών μηχανών.

Προκειμένου να προστατεύσουμε το υπολογιστικό σύστημα έναντι της απειλής της «κλοπής» του εικονικού μηχανήματος (VM theft) πρέπει να δημιουργήσουμε τέτοιους περιορισμούς έτσι ώστε το εικονικό μηχάνημα να «δένεται» με τον host στον οποίο φιλοξενείται κι να μην υπάρχει η δυνατότητα μεταφοράς του σε τρίτο άγνωστο host.

Η απειλή VM Escape μπορεί να αντιμετωπιστεί αποτελεσματικά εάν πραγματοποιούμε συνεχείς ελέγχους στον hypervisor και ενισχύοντας συνεχώς την ασφάλειά του.

Η απειλή του Instant-on Gap μπορεί να αντιμετωπιστεί επαρκώς εφόσον σε κάθε φυσικό μηχάνημα υπάρχει ένα εικονικό το οποίο φροντίζει και ελέγχει εάν όλα τα υπόλοιπα εικονικά μηχανήματα παραμένουν ενημερωμένα.

Μια από τις πιο σοβαρές απειλές που χρειάζεται να έχουμε στο νου μας κατά την χρήση των εικονικών μηχανών είναι αυτή του Hyperjacking (έχει αναλυθεί σε προηγούμενο σημείο). Μιας και δεν είναι εύκολο να αντιληφθούμε την πραγμάτωση αυτής της απειλής είμαστε υποχρεωμένοι να ενορχηστρώσουμε ορισμένα αντίμετρα στο σύστημά μας. Αυτά τα αντίμετρα περιλαμβάνουν την ασφαλή εκκίνηση (secure boot) του hypervisor, η ασφαλής εκκίνηση θα πρέπει να εξασφαλίζεται από το υλικό χρησιμοποιώντας την τεχνική TCB. Ένα ακόμη αντίμετρο εναντίον του Hyperjacking είναι η σάρωση κι ο έλεγχος της ακεραιότητας του hardware όπου βρίσκεται ο hypervisor.

Τα μετρά που μπορούμε να λάβουμε εναντίον της απειλής της Inter-VM attack περιλαμβάνουν την τοποθέτηση firewall ανάμεσα στα VM καθώς και μηχανισμούς ανίχνευσης κακόβουλων δραστηριοτήτων. Ακόμα μπορούμε να εφαρμόσουμε τεχνικές απομόνωσης των εικονικών μηχανών.

Παραπάνω είχαμε μελετήσει τους κινδύνους που επιφέρουν τα εικονικά μηχανήματα όταν είναι ανενεργά είτε λόγω migration είτε έπειτα από ενέργειες του χρήστη. Για να εξαλείψουμε τις απειλές που ελλοχεύουν στην ανενεργή κατάσταση των VM μπορούμε να προβούμε στην κρυπτογράφηση τους. Με τον προηγούμενο τρόπο προστατεύουμε τα VM από την απειλή μη εξουσιοδοτημένης τροποποίησης των αρχείων 'στιγμιότυπων' των εικονικών μηχανών (unauthorized alteration of virtual machine image files).

Ως προς την απειλή της VM escape το μόνο μέτρο που μπορούμε να πάρουμε είναι η καλύτερη προστασία του hypervisor από εξωτερικές απειλές (firewall, fully control every running app, anti-virus κλπ.).

IoT Malware

Η απειλή του malware [54] στο IoT είναι αρκετά δύσκολο να αντιμετωπιστεί μιας οι συμβατές λύσεις προστασίας από malware δεν μπορούν να εφαρμοστούν σε συσκευές που έχουν περιορισμένες δυνατότητες. Με άλλα λόγια οι μικροσυσκευές του περιβάλλοντος του IoT αδυνατούν να τρέχουν κάποιο anti-virus ή/και anti-malware πρόγραμμα που θα τρέχει σε πραγματικό χρόνο και θα ανιχνεύει άμεσα ένα κακόβουλο αρχείο. Εάν συνυπολογίσουμε την ευρεία γκάμα των αρχιτεκτονικών όλων των συσκευών του IoT τότε καταλαβαίνουμε ότι δεν εφικτό να δημιουργηθεί ένα ενιαίο εργαλείο προστασίας. Επομένως το πιο ισχυρό όπλο ενάντια στο malware αποτελούν η κάθε περίπτωση εξέταση ευπαθειών και η άμεση επίλυση τους καθώς η περαιτέρω έρευνα του θέματος.

Τα παραπάνω γίνονται αντιληπτά μιας και η λύση που δόθηκε για το Linux.Darlloz ήταν οι παροτρύνσεις για αλλαγές σε κώδικα, καλύτερο έλεγχο των συνδέσεων των συσκευών καθώς και συχνή ενημέρωση των κωδικών.

Man-in-the-middle attack (MITM)

Στην περίπτωση των MITM επιθέσεων η αυθεντικοποίηση των άκρων του δικτύου καθώς και η κρυπτογράφηση των δεδομένων αποτελούν τις βέλτιστες πρακτικές που εξασφαλίζουν την ευστάθεια του πληροφοριακού συστήματος [55].

Replay attack

Ενάντια στην απειλή των replay attacks ισχύουν και τα μέτρα προστασίας που ισχύουν στις MITM επιθέσεις. Επιπρόσθετα υπάρχουν αρκετά πρωτόκολλα τα οποία χρησιμοποιούν χρονοσύμανση (timestamp) ή hash function ή την αναπαραγωγή τυχαίων αριθμών έτσι ώστε να εξασφαλιστεί η ανίχνευση κι η επαρκή προστασία από τέτοιου είδους επιθέσεις [56].

Φυσικές επιθέσεις (Physical attacks)

Η τελευταία κατηγορία απειλών που πρέπει να αντιμετωπιστεί αποτελείται από τις φυσικές επιθέσεις που αναλύθηκαν προηγουμένως. Το κυριότερο μέτρο ασφαλείας εναντίον τέτοιων απειλών είναι η αντιγραφή των δεδομένων (backup) σε πολλά σημεία τα οποία θα είναι και διάσπαρτα στον κόσμο. Οι υποδομές που θα φιλοξενούν τα δεδομένα πρέπει να λαμβάνουν κυρίως μέτρα προστασίας όπως η τοποθέτηση πορτών ασφαλείας, κάμερες, συναγερμό άλλα και μέτρα που θα εξασφαλίζουν την προστασία των υποδομών από φυσικές καταστροφές αντιπλημμυρικά έργα και συστήματα πυρόσβεσης κ.α. .

Προκειμένου να εμποδιστεί η απειλή του reverse engineering σε νόμιμη συσκευή πρέπει να ακολουθούνται οι γενικοί κανόνες περί προγραμματισμού στις κινητές συσκευές (όπως ο

«ευάλωτος» κώδικας πρέπει να αποθηκεύεται κι να τρέχει σε προστατευμένους servers, τμηματοποίηση κώδικα κ.α.).

Τύπος Επίθεσης	Ευπάθεια συστήματος	Προτεινόμενη αντιμετώπιση
DoS (Denial of Service)	Availability	Firewall, blacklists
Unauthorized subscription/publication	Confidentiality, Integrity	Firewall, blacklists, tokens
Spoofing (πλαστογράφηση)	Integrity	Firewall, blacklists, tokens
Τροποποίηση των αλγορίθμων παραγωγής των δεδομένων	Integrity	Κατάλληλοι αλγόριθμοι φιλτραρίσματος των εισερχόμενων δεδομένων
Data source validation	Integrity, Availability	Tokens to sources
Μη σχεσιακές βάσεις δεδομένων	CIA	Ιδιαίτερη προσοχή στην σχεδίαση του πληροφοριακού συστήματος
Ιδιωτικότητα των χρηστών	Confidentiality	Έλεγχος κατά την αξιοποίηση τρίτων εφαρμογών
VM Theft	Απομονωμένο CIA	“Δέσιμο” VM με τον host
VM Escape	CIA όλου του συστήματος	Hypervisor patching
Instant-On Gaps	Integrity, Availability	Ένα VM ελέγχει τα υπόλοιπα
Hyperjacking	CIA όλου του συστήματος	Secure boot, hardware scan & monitoring
Unauthorized alteration of virtual machine image files	Integrity	Encryption
IoT Malware	CIA	Συνεχής εξέταση ευπαθειών και άμεση επίλυση
Man-in-the-middle attack (MITM)	Confidentiality, Integrity	Authentication & Encryption
Replay attack	Confidentiality, Integrity	Encryption with timestamps or hash functions or random numbers

Physical attacks (Φυσικές επιθέσεις)	Integrity, Availability	Backup plan, basic programming rules on mobile devices

Πίνακας 3 Κατηγοριοποίηση Απειλών IoT και Λύσεις

Αυθεντικοποίηση

Η αυθεντικοποίηση όπως την αναλύσαμε παραπάνω αποτελεί μια από τις σημαντικότερες διαδικασίες όπου εξασφάλισης της ασφάλειας του συστήματός μας. Σύμφωνα με το [57] οι τεχνικές αυθεντικοποίησης μπορούν να κατανεμηθούν στις έξις κατηγορίες: κεντρική (χρήση ενός κεντρικού server όπου θα αυθεντικοποιεί κάθε οντότητα), την κατανεμημένη (κατανεμημένη αυθεντικοποίηση μεταξύ των άκρων), ιεραρχική (χρήση ιεραρχικής αρχιτεκτονικής κατά την διαδικασία αυθεντικοποίησης) και επίπεδη (όπου δεν υπάρχει καμία ιεραρχία κατά την αυθεντικοποίηση).

Σύμφωνα με την ίδια έρευνα τα χαρακτηριστικά των ποικίλλων πρωτοκόλλων αυθεντικοποίησης μπορούν να συνοψιστούν στα παρακάτω: διπλή-αυθεντικοποίηση (κοινή ή αυθεντικοποίηση μιας διαδρομής), χρήση επιπρόσθετου υλικού (εάν υπάρχει ανάγκη επιπλέον υλικού για να επιτευχθεί η αυθεντικοποίηση, π.χ. rfid tag), πολλαπλά διαπιστευτήρια (πολλαπλά επίπεδα αυθεντικοποίησης με διαφορετικά διαπιστευτήρια), φάση εγγραφής (διαδικασίας εγγραφής του χρήστη) καθώς και η φάση εκτός σύνδεσης (διαδικασία κατά την οποία προετοιμάζεται το απαιτούμενο δίκτυο).

Αυθεντικοποίηση Συσκευών

Η αυθεντικοποίηση των συσκευών που συμμετέχουν σε ένα IoT περιβάλλον αποτελεί μια ιδιαιτέρως πολύπλοκη διαδικασία. Ο κυριότερος λόγος που συμβάλει σε αυτήν την κατάσταση είναι οι περιορισμοί που θέτονται από το υλικό των μικροσυσκευών και ουσιαστικά απαγορεύουν την χρήση κλασσικών μεθόδων αυθεντικοποίησης. Μιας και αυτοί απαιτούν αρκετή ενέργεια και επεξεργαστική ισχύς.

Μια από τις πιο απλές μεθόδους αυθεντικοποίησης των συσκευών αποτελεί η χρήση ενός προκαθορισμένου token σε συνδυασμό με κάποιο id (π.χ. όνομα της συσκευής, id χρήστη) έτσι ώστε ο server να δέχεται δεδομένα μόνο από την συσκευή που έχει εγγραφεί με συγκεκριμένο token και να απορρίπτει άλλες συσκευές. Αυτός ο τρόπος αυθεντικοποίησης μπορεί να ενισχυθεί εάν το token παράγεται με βάση κάποιο μοναδικό χαρακτηριστικό της εκάστοτε συσκευής όπως είναι η mac address, οι συντεταγμένες της συσκευής (εάν δεν είναι κινητή). Ενναλακτικά θα μπορούσαν να χρησιμοποιηθούν συγκεκριμένα credentials τα οποία θα είναι αποθηκευμένα σε ένα ασφαλές μέρος της συσκευής προκειμένου να επιτευχθεί η σύνδεση της συσκευής με το υπόλοιπο πληροφοριακό σύστημα. Τα παραπάνω μπορούν να συνδυαστούν με ένα βιομετρικό στοιχείο του χρήστη, με μια smart card, ένα rfid tag. Αυτά τα δεδομένα μπορούν να απαιτούνται κατά την εγγραφή του χρήστη στο σύστημα κι στην συνέχεια να αντιστοιχίζονται με συγκεκριμένη συσκευή.

Εφόσον μας το επιτρέπουν οι διαθέσιμου πόροι της συσκευής υπάρχουν αρκετοί αλγόριθμοι και πρωτόκολλα αυθεντικοποίησης [58]. Τα πρωτόκολλα αυτά μπορούν να χωριστούν σε τρεις κατηγορίες την συμμετρική κρυπτογράφηση, την ασύμμετρη κρυπτογράφηση καθώς και την υβριδική.

Αυθεντικοποίηση Χρήστη

Η πιο ευρέως διαδεδομένη και ταυτοχρόνως λιγότερη ασφαλής μέθοδο αυθεντικοποίησης χρηστών αποτελείται από τον συνδυασμό ενός username και ενός κωδικού. Ωστόσο αυτή η μέθοδος μπορεί να ενισχυθεί σημαντικά εφόσον συνδυαστεί με βιομετρικά χαρακτηριστικά των χρηστών ή με έναν μοναδικό κι τυχαίο κωδικό που μπορεί να αποσταλεί στον χρηστή κι να ισχύει για λίγα λεπτά. Ακόμα υπάρχει η δυνατότητα αυθεντικοποίησης χρηστών μέσω τρίτων υλικών συσκευών π.χ. Rfid-tag, Smart card. Όμως συνδυασμός δυο ή παραπάνω μεθόδων αποτελεί την ασφαλέστερη λύση.

Μέθοδος Αυθεντικοποίησης	Εξασφάλιση Ασφάλειας (1-5)	Παρατηρήσεις
Username – Password	1-2	Η ασφάλεια εξαρτάται αποκλειστικά από την πολυπλοκότητα του κωδικού
Βιομετρικά στοιχεία	4	Είναι σχεδόν αδύνατο δυο άνθρωποι να έχουν τα ίδια βιομετρικά στοιχεία, υπάρχει η δυνατότητα υποκλοπής όμως
Random Password	3	Εξαρτάται από τον τρόπο παραγωγής και μετάδοσης στον χρήστη
Rfid-Tag, Smart Card	3	Κίνδυνος υποκλοπής
Αυθεντικοποίηση τριών ή παραπάνω παραγόντων	5	Ελάχιστη η πιθανότητα παραβίασης

Πίνακας 4 Κατηγοριοποίηση Μεθόδων Αυθεντικοποίησης

Πλατφόρμες IoT

Azure IoT Hub

Η Microsoft προσφέρει την πλατφόρμα Azure IoT Hub ως ένα εργαλείο ανάπτυξης IoT εφαρμογών. Η υπηρεσία αύτη της Microsoft παρέχει την δυνατότητα αμφίδρομης επικοινωνίας μεταξύ συσκευών και άλλων υπηρεσιών της Microsoft στο cloud. Το παραπάνω γεγονός καθιστά τον δίαυλο επικοινωνίας μεταξύ συσκευών και χρηστών με την πλατφόρμα αξιόπιστο και ασφαλή. Κάθε αυθεντικοποιημένη συσκευή αποστέλλει μηνύματα προς το Hub, το οποίο στην συνέχεια μεταβιβάζει κάθε μήνυμα προς μια ή περισσότερα άκρα άλλων υπηρεσιών. Σε κάθε συσκευή αντιστοιχείται μια ψηφιακή της απεικόνιση (όπου ονομάζεται και δίδυμη συσκευή) στο υπολογιστικό νέφος. Κάθε «δίδυμη συσκευή» περιέχει τις επιθυμητές πληροφορίες (καθορίζονται από την εφαρμογή, «διαβάζονται» από την συσκευή) και τις αναφερόμενες πληροφορίες (καθορίζονται από την συσκευή, «διαβάζονται» από την εφαρμογή).

Το IoT Hub υποστηρίζει τα πρωτόκολλα AMQP με προαιρετική την υποστήριξη από WebSocket, το MQTT καθώς και HTTP πάνω από TLS. Ακόμα η πλατφόρμα μέσω του Azure IoT protocol gateway [59] έχει την δυνατότητα να ενσωματώσει και διαφορά άλλα πρότυπα. Την ασφάλεια της πλατφόρμας την εγγυάται η χρήση του Azure Active Directory [60] για την αυθεντικοποίηση του χρήστη. Η αυθεντικοποίηση των συσκευών γίνεται μέσω του Azure Hub Identity Registry [61] όπου αποθηκεύονται όλα τα μοναδικά αναγνωριστικά της κάθε

συσκευής και η σύνδεση πάντα προκαλείται από την συσκευή με χρήση του προτύπου TLS με πιστοποιητικά X.509 [62].

AWS IoT

Η λύση που προσφέρει η Amazon για την διαχείριση IoT συσκευών ονομάζεται AWS IoT. Οι συσκευές που συνδέονται με την πλατφόρμα αποκτούν μια ψηφιακή απεικόνιση που ονομάζεται device shadow [63] προκείμενου οι πληροφορίες της συσκευής να μπορούν να προσπεραστούν ανεξαρτήτως της κατάστασης της σύνδεσης (ενεργή ή ανενεργή) της συσκευής. Η υπηρεσία AWS IoT Device Management [64] αναλαμβάνει την διαχείριση, παρακολούθηση και την ταξινόμηση των συσκευών. Ακόμα η πλατφόρμα προσφέρει πλήρη διασύνδεση με όλες τις υπηρεσίες της Amazon γεγονός που συμβάλει στην προστασία του συστήματος από εξωτερικές εφαρμογές και πηγές.

Το AWS IoT ενσωματώνει τα πρωτόκολλα επικοινωνίας MQTT, HTTPS και MQTT πάνω από Secure WebSockets. Προκείμενου να εξασφαλίσει την ασφάλεια της η πλατφόρμα χρησιμοποίει την υπηρεσία Amazon Cognito [65] να πιστοποιήσει την αυθεντικότητα κινητών συσκευών. Η σύνδεση των συσκευών με την πλατφόρμα χρησιμοποιεί υποχρεωτικά TLS με υποστήριξη και για πιστοποιητικά τύπου X.509.

Google Cloud IoT Core

Η Google ονομάζει την δικιά της IoT πλατφόρμα Cloud IoT Core. Η πλατφόρμα διαχωρίζεται σε δυο κύρια συστήματα [66], το ένα αφορά την διαχείριση των συσκευών (ονομάζεται device manager) και το δεύτερο την «ενοποίηση» των πρωτοκόλλων (ονομάζεται protocol bridge). Οι λειτουργίες του πρώτου συστήματος αποτελούνται από την εγγραφή, αυθεντικοποίηση και την εξουσιοδότηση (authorization) των συσκευών στην πλατφόρμα. Οι λειτουργίες αυτές επιτυγχάνονται μέσω μητρώου (registry). Οι συσκευές αντιστοιχίζονται με ένα μοναδικό κλειδί (ID) και μπορούν να ταυτοποιηθούν με ένα όνομα (full resource name). Το δεύτερο σύστημα αναλαμβάνει την αποστολή δεδομένων από τις συσκευές στην πλατφόρμα μέσω των πρωτοκόλλων επικοινωνίας HTTP και MQTT. Η δομή των δεδομένων ενδέχεται να προκύπτει αυθαίρετα από την χρήστη, λειτουργία που οι προηγούμενες πλατφόρμες δεν προσφέρουν [67].

Το Cloud IoT Core προσφέρει διασύνδεση με όλες τις υπηρεσίες της Google γεγονός που βελτιώνει την ασφάλεια και αξιοπιστία του συστήματος. Ακόμα υποστηρίζει την αυθεντικοποίηση με ζεύγος δημόσιου και ιδιωτικού κλειδιού ανά συσκευή με την χρήση JSON Web Tokens [68] και την ενσωμάτωση RSA ή Elliptic Curve αλγορίθμων κρυπτογράφησης. Η χρήση του πρωτόκολλου TLS 1.2 είναι απαραίτητη για την επικοινωνία με συνδέσεις του προτύπου MQTT. Ακόμη μέσω της Cloud Identity [69] πραγματοποιείται η αυθεντικοποίηση και ο έλεγχος των συνδέσεων προς την πλατφόρμα Cloud IoT Core.

Kinetic IoT

Η Cisco προσφέρει την πλατφόρμα Kinetic IoT [70] για την διαχείριση IoT συσκευών. Η πλατφόρμα αποτελείται από τρία κύρια μέρη, το πρώτο ονομάζεται Edge and Fog Processing Module (EFM) [71] και αναλαμβάνει τον διαμοιρασμό των δεδομένων σε κατανεμημένους κόμβους του δικτύου προκειμένου να λάβουν άμεσες αποφάσεις κοντά στο πεδίο δράσης των αισθητήρων. Το δεύτερο μέρος ονομάζεται Data Control Module (DCM) [72] και είναι υπεύθυνο για την μεταφορά των δεδομένων από τις συνδεδεμένες συσκευές σε κατάλληλους κόμβους οι οποίοι επιτελούν λειτουργίες που έχει ορίσει ο χρήστης. Το

τελευταίο μέρος αναλαμβάνει την απεικόνιση όλων των δεδομένων του χρήστη και ονομάζεται Gateway Management Module (GMM).

Η πλατφόρμα υποστηρίζει τα πρωτόκολλα MQTT καθώς και AMQP. Κάθε επικοινωνία διασυνδεμένων εφαρμογών με την πλατφόρμα υποχρεωτικά γίνεται πάνω από TLS και με κλειδιά τα οποία είναι μοναδικά σε κάθε χρήστη.

Kaa

Η πλατφόρμα Kaa υποστηρίζεται από την εταιρία KaaIoT και αποτελεί μια λύση ανοικτή κώδικα για την διαχείριση συσκευών με ενσωματωμένους αισθητήρες. Το πλεονέκτημα αυτής της πλατφόρμας είναι ότι παρέχει πλήρη ελευθερία σε προγραμματιστές για την ανάπτυξη του κώδικα, την ενσωμάτωση πρωτόκολλών καθώς και την ενσωμάτωση τρίτων υπηρεσιών. Ωστόσο το προηγούμενο χαρακτηριστικό εγκυμονεί κινδύνους για την ασφάλεια του συστήματος εφόσον προκύψει κάποιο λάθος κατά την σχεδίαση του συστήματος και την ενσωμάτωση τρίτων υπηρεσιών. Η ευρεία γκάμα γνωστών τρίτων υπηρεσιών ανοικτού κώδικα που υποστηρίζονται από την Kaa ενδεικτικά είναι η kafka για την διαχείριση της ροής των δεδομένων, kubernetes για την ενορχήστρωση των containers, redis για την διαχείριση των cache δεδομένων, cassandra και mongoDB για την αποθήκευση των δεδομένων σε βάσεις δεδομένων καθώς και του keycloak για την διαχείριση των χρηστών [73].

Η πλατφόρμα υποστηρίζει τα πρωτόκολλα MQTT, CoAP, HTTP για την επικοινωνία με έξυπνες συσκευές καθώς και τα πρότυπα TLS ή DTLS για την εξασφάλιση της ασφάλειας του συστήματος. Η αυθεντικοποίηση των συσκευών μπορεί να γίνει από μια ευρεία γκάμα μηχανισμών που αποτελείται από κανέναν μέχρι και την αυθεντικοποιήση με τον έλεγχο προσυμφωνημένων κλειδιών για την επιβεβαίωση των TLS πιστοποιητικών του "πελάτη" [74]. Επιπλέον διαθέτει cloud έκδοση για cloud based υλοποιήσεις.

OpenRemote

Η OpenRemote [75] είναι μια ακόμη λύση ανοικτού κώδικα για την διαχείριση έξυπνων συσκευών που ξεκίνησε να αναπτύσσεται από το 2009. Στίχος της πλατφόρμας είναι η αντιμετώπιση προβλημάτων που προκύπτουν από την ενοποίηση διαφορετικών πρωτοκόλλων. Τα κύρια μέρη είναι ένας διαδικτυακός σχεδιαστής (online designer), ελεγκτής διασυνδέσεων (OpenRemote Controller) και μια διεπαφή (panel) ή μια εφαρμογή για Android/iOS για την απεικόνιση των δεδομένων. Ο OpenRemote Controller είναι υπεύθυνος για την συνεχή επικοινωνία των συσκευών με τις προσφερόμενες υπηρεσίες του συστήματος καθώς και για την εκτέλεση των αυτοματοποιημένων διεργασιών. Ο online designer αναλαμβάνει την διαχείριση των συσκευών, των διαδικτυακών υπηρεσιών, των μακροεντολών και των κανόνων αυτοματοποίησης του συστήματος. Ακόμα αυτό το "εργαλείο" παρέχει την δυνατότητα τροποποίησης της απεικόνισης των δεδομένων στην διεπαφή με τον χρήστη μέσω ενός προγράμματος περιήγησης και στις εφαρμογές για Android/iOS. Ο σκοπός του τρίτου μέρους (panel) είναι η απεικόνιση και ο έλεγχος των διεργασιών που εκτελούνται από τα άλλα δυο μέρη από τους χρήστες του συστήματος [76].

Η πλατφόρμα υποστηρίζει τα πρωτόκολλα MQTT, WebSocket, HTTP για επικοινωνία με έξυπνες συσκευές καθώς και το πρότυπο TLS. Η αυθεντικοποίηση και η εξουσιοδότηση των χρηστών στο OpenRemote [77] διαχειρίζονται από την ενσωματωμένη υπηρεσία Keycloak [78].

Thinger.io

Η πλατφόρμα Thinger.io [79] αποτελεί μια ανοικτού κώδικα λύση που προσφέρει μια ευρεία γκάμα εργαλείων διαχείρισης συνδεδεμένων συσκευών. Χάρη στην κλιμακωσιμότητα που διαθέτει μπορεί να ανακτά δεδομένα από χιλιάδες συσκευές την χρονική στιγμή που απαιτείται και αυτό να γίνεται με χαμηλό υπολογιστικό κόστος. Η Thinger.io αποτελείται από δυο κύρια μέρη, το Backend που ουσιαστικά είναι ο IoT server και Frontend που αποτελεί μια web-based αναπαράσταση των δεδομένων αλλά και του τρόπου διαχείρισης των συσκευών. Η επικοινωνία των συσκευών με την πλατφόρμα είναι αμφίδρομη και δύναται να χρησιμοποιεί τα πρότυπα Websocket, HTTP και MQTT πάνω από το πρωτόκολλο TLS [80].

Η πλατφόρμα ενσωματώνει και ποίκιλες τρίτες εφαρμογές όπως το Grafana (ανάλυση δεδομένων), VS Code (online idle), Node-RED(διαχείριση ροών δεδομένων). Ακόμα διαθέτει cloud έκδοση για cloud based εφαρμογές.

ThingsBoard

Η πλατφόρμα ανοικτού κώδικα ThingsBoard [81] παρέχει στους χρήστες ένα απλό γραφικό περιβάλλον διαχείρισης έξυπνων συσκευών, αναπαράστασης δεδομένων και διαχείρισης της ροής των δεδομένων. Το πρώτο μέρος του συστήματος, το οποίο ονομάζεται ThingsBoard Core, είναι υπεύθυνο για την υπεύθυνο για την διαχείριση της ενεργής κατάστασης της κάθε συσκευής και για την διακίνηση των δεδομένων μέσω κλήσεων τύπου REST API και WebSockets εγγραφών. Ένα άλλο μέρος του συστήματος ονομάζεται ThingsBoard Rule Engine και αποτελεί τον κορμό του συστήματος μιας είναι υπεύθυνο για την επεξεργασία των εισερχόμενων μηνυμάτων και για την αποστολή αυτών προς την καθορισμένη κατεύθυνση. Το τρίτο μέρος λέγεται ThingsBoard Web UI και προσφέρει ένα ελαφρύ και απλό στην χρήση στατικό γραφικό περιβάλλον στο οποίο εκτελούνται διάφορες εφαρμογές που επικοινωνούν με το ThingsBoard Core. Το μέρος που ολοκληρώνει το σύστημα είναι το ThingsBoard Transports και φέρει την ευθύνη της διαχείρισης της επικοινωνίας όλου του συστήματος με της συνδεδεμένες συσκευές.

Η πλατφόρμα υποστηρίζει την επικοινωνία με τις έξυπνες συσκευές με τα πρωτόκολλα MQTT, HTTP καθώς και CoAP. Για την εξασφάλιση της ασφάλειας της επικοινωνίας υποστηρίζει το πρότυπο TLS καθώς και την χρήση πιστοποιητικών τύπου X.509. Ακόμα προσφέρει την δυνατότητα ενσωμάτωσης τρίτων εφαρμογών διαχείρισης της ταυτότητας και της πρόσβασης των χρηστών όπως το keyckloak, Auth0, Google Cloud Platform κ.α.

Πλατφόρμα	Πάροχος	Χρήση	Πρωτόκολλα επικοινωνίας	Ασφάλεια
Cloud IoT Core	Google	Έξυπνες πόλεις, κτίρια και παρακολούθηση στοιχείων σε πραγματικό χρόνο	HTTP, MQTT	TLS, Google Cloud IAM, X.509 Certificates
Azure IoT Hub	Microsoft	Υγειονομική περίθαλψη, Λιανικό εμπόριο, Κατασκευαστικές εργασίες	HTTP, MQTT, AMQP over Web Sockets	Azure Hub Identity Registry, TLS, X.509 Certificates

AWS IoT	Amazon	Έξυπνη πόλη, Έξυπνο σπίτι, Γεωργία	HTTP, MQTT, MQTT over Secure WebSockets	Amazon Cognito
Kinetic IoT	Cisco	Έξυπνα οχήματα, Κατασκευαστικές εργασίες, Έξυπνη πόλη	MQTT	TLS, TLS, X.509 Certificates
Kaa	Ελεύθερο Λογισμικό	Έξυπνες πόλεις, Έξυπνη ενέργεια και Έξυπνα οχήματα	MQTT, CoAP, HTTP	TLS, DTLS, Keycloak
Thinger.io	Ελεύθερο Λογισμικό	Έξυπνες πόλεις, Γεωργία, Κατασκευαστικές εργασίες	HTTP, MQTT	TLS
OpenRemote	Ελεύθερο Λογισμικό	Έξυπνες πόλεις, Έξυπνη ενέργεια και Έξυπνα οχήματα	MQTT, WebSocket, HTTP	TLS, Keycloak
ThingsBoard	Ελεύθερο Λογισμικό	Έξυπνη ενέργεια, Γεωργία, Έξυπνα οχήματα	MQTT, CoAP, HTTP	TLS, X.509 certificates, OAuth integration

Πίνακας 5 Πλατφόρμες IoT

Ο παραπάνω πίνακας αποτυπώνει τις διαφορές των πλατφορμών που αναφέρθηκαν προηγουμένως. Όλες οι πλατφόρμες που δεν είναι ανοικτού κώδικα εκτελούνται με cloud based υλοποιήσεις, δεν προσφέρουν on premise εφαρμογές και αφού ο κώδικας του δεν είναι παραμετροποιήσιμος που δεν προσφέρουν την ανάλογη ελευθερία στους σχεδιαστές σε σχέση με τις πλατφόρμες ανοικτού κώδικα.

Λαμβάνοντας υπόψιν την παραπάνω σύγκριση, η πλατφόρμα της Google ενδείκνυται για χρήση σε έξυπνες πόλεις σε αντίθεση με την αντίστοιχη της Microsoft που προτιμάται σε εφαρμογές σχετικές με την υγεία. Η πλατφόρμα AWS IoT αποτελεί ιδανική λύση για το εφαρμογές σε οικιακό περιβάλλον ενώ η Kinetic IoT συνίσταται για υλοποιήσεις διαχείρισης οχημάτων. Η πλατφόρμα Kaa προτιμάται σε εφαρμογές σχετικές με έξυπνες πόλεις ενώ η Thinger.io για εφαρμογές διαχείρισης γεωργικών εργασιών. Τέλος η OpenRemote συνίσταται για open source υλοποιήσεις με σκοπό την διαχείριση έξυπνων οχημάτων ενώ η πλατφόρμα ThingsBoard προτιμάται σε εφαρμογές σχετικές με την διαχείριση της ενέργειας.

Περιγραφή προτεινόμενης λύσης

Λαμβάνοντας υπόψη όσα έχουν μελετηθεί στην παρούσα διπλωματική εργασία και έπειτα από σύγκριση ανάμεσα σε διαθέσιμες πλατφόρμες διαχείρισης IoT συσκευών ανοικτού κώδικα προτείνεται η παρακάτω λύση. Σημαντικά κριτήρια στην αναζήτηση της βέλτιστης πλατφόρμας ήταν η ύπαρξη πλούσιας τεκμηρίωσης, η πρόσφατη ενημέρωση της πλατφόρμας καθώς και τα υποστηριζόμενα πρωτόκολλα επικοινωνίας με τις συσκευές. Ένα ακόμη σημαντικό κριτήριο ήταν η διαχείριση των χρηστών του συστήματος.

Στον παρακάτω πίνακα απεικονίζονται οι διαθέσιμες πλατφόρμες που συμμετείχαν στην σύγκριση.

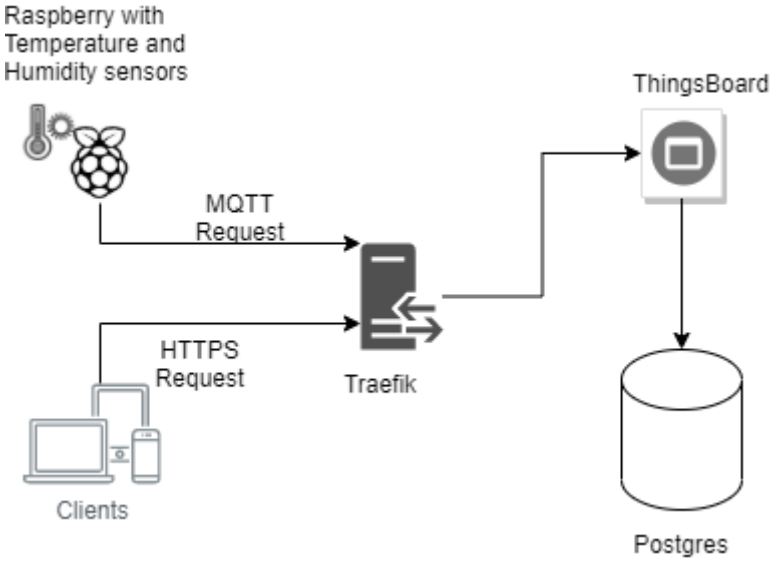
Πλατφόρμα	Διαδικασία-Εγκατάστασης	Υποστήριξη Docker	Πρωτόκολλα Επικοινωνίας	Προϋποθέσεις συστήματος	Διαχείριση Χρηστών
ThingsBoard	Πολύ εύκολη	Ναι	MQTT, Coap, HTTP	RAM: Postgres: 2-4GB Cassandra:4GB	Ενσωματωμένη role-based διαχείριση & Ενσωματωμένη υποστήριξη OAuth
Kaa	Μέτρια	Όχι	MQTT, CoAP, HTTP	64-bit OS 4 Gb RAM	Ενσωματωμένη υποστήριξη Keycloak
Devicehive	Εύκολη	Ναι	Websocket, MQTT	4 CPU cores 8 GB of RAM At least 16 GB of disk.	Ενσωματωμένη role-based διαχείριση & Ενσωματωμένη υποστήριξη OAuth
SiteWhere	Εύκολη	Ναι	MQTT, WebSocket	16GB RAM 4 CPUs cores 100GB Hard Disk/SSD	Ενσωματωμένη role-based διαχείριση

Πίνακας 6 Σύγκριση open-source IoT πλατφορμών

Λαμβάνοντας υπόψιν τον πίνακα 6 επιλέχθηκε η πλατφόρμα ThingsBoard για την συγκεκριμένη υλοποίηση μιας και προσφέρει πλούσια υποστήριξη στην κοινότητα με αρκετά μεγάλο όγκο τεκμηρίωσης και συνεχώς μέχρι σήμερα ανανεώνεται. Ακόμα είναι από τις ελάχιστες open source IoT πλατφόρμες που υποστηρίζουν τρίτες πλατφόρμες αυθεντικοποίησης. Επιπλέον παρέχει σύνδεση με τουλάχιστον τρία πρωτόκολλα MQTT, Coap, HTTP και ενσωματώνει την βάση δεδομένων Postgres και άλλα εργαλεία caching. Τελευταίος λόγος επιλογής της συγκεκριμένης πλατφόρμας αποτελεί ότι η χρήση της αποδείχθηκε αρκετά εύκολη.

Ακόμα επιλέχθηκε η πλατφόρμα αυθεντικοποίησης χρηστών auth0 μιας και υπήρχαν προβλήματα ασυμβατότητας του ThingsBoard με τις πλατφόρμες keycloak και την αντίστοιχη της google.

Τέλος επιλέχθηκε η υπηρεσία reverse proxy traefik μιας και αποτελεί μια αξιόπιστη λύση και προσφέρει την δυνατότητα χρήσης https με naitive self signed SSL πιστοποιητικά.



Σχήμα 14 Αρχιτεκτονική υλοποίησης

Εγκατάσταση:

Αρχικά για το συγκεκριμένο project χρειάζεται να πραγματοποιηθεί η εγκατάσταση του docker-compose και του docker όπως περιγράφονται στα [82] και [83] αντίστοιχα.

Για την επιβεβαίωση ότι ο docker έχει εγκατασταθεί επιτυχώς αρκεί να εκτελεστεί η εντολή:

```
sudo docker run hello-world
```

Και κατόπιν να εμφανίζεται ένα παρόμοιο αποτέλεσμα με:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:df5f5184104426b65967e016ff2ac0bfcd44ad7899ca3bbcf8e44e4461491a9e
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started
```

Ακόμη απαιτείτε η δημιουργία του εσωτερικού δικτύου που είναι αναγκαίο για την επικοινωνία του traefik με την πλατφόρμα thingsboard, το παραπάνω γίνεται με τις εντολές:

```
docker network create transit_idp
docker network create transit_thingsb
```

Το αποτέλεσμα θα είναι παρόμοιο με:

```
5c51cf08ba3f41e7ebafd1d8f0f0e57aa14ca8d13dd3c9210442f9a48192a09b
6c7e1b7b01f802366c582cf0ba7fc17beb8ec80e73ee1c8bb78eb41e85d5765f
```

Στην συνέχεια απαιτείται η προσαρμογή των αρχείων docker-compose.yml του treafik και της πλατφόρμας thingsboard αντίστοιχα. Τα δυο αρχεία αυτά πρέπει να αποθηκευτούν σε ξεχωριστούς φακέλους.

Αρχικά πραγματοποιείται η προσαρμογή του docker-compose.yml αρχείου του thingsboard. Υπενθυμίζεται ότι για την αποφυγή προβλημάτων ασυμβατότητας της έκδοσης γίνεται χρήση της έκδοσης `tb-postgres:2.5.4` που ενσωματώνει και την βάση δεδομένων(postgres). Για να προωθεί η πλατφόρμα τα αιτήματα από την πόρτα 9090 στην 8080 που ακούει ο treafik απαιτείται αλλαγή στο κατάλληλο σημείο. Στα δίκτυα γίνεται η κατάλληλη αλλαγή και ο ορισμός των παραπάνω δικτύων και κατόπιν ορίζεται το domain name της πλατφόρμας.

```
1.  version: '2'
2.  services:
3.  app:
4.  restart: "always"
5.  image: "thingsboard/tb-postgres:2.5.4"
6.  ports:
        -   "8080:9090"
        -   "1883:1883"
        -   "5683:5683/udp"
7.  environment:
8.  TB_QUEUE_TYPE: in-memory
9.  volumes:
        -   ~/.mytb-data:/data
        -   ~/.mytb-logs:/var/log/thingsboard
10. networks:
11. transit_thingsb:
12. labels:
        -   "traefik.backend=thingsboard"
        -   "traefik.port=9090"
```

```
      - "traefik.frontend.rule=Host:app.tsere.com"
      - "traefik.protocol=http"
      - "traefik.docker.network=transit_thingsb"
13. networks:
14. transit_thingsb:
15. external: true
```

Κατόπιν γίνεται αναπροσαρμογή του docker-compose.yml αρχείου του treafik. Υπενθυμίζεται ότι για την αποφυγή προβλημάτων ασυμβατότητας της έκδοσης καθορίζεται η έκδοση: `traefik:v1.7.4-alpine.` Για να δέχεται η υπηρεσία τα αιτήματα στην πόρτα 8080 απαιτείται αλλαγή στο κατάλληλο σημείο. Στα δίκτυα γίνεται η κατάλληλη αλλαγή και ο ορισμός των παραπάνω δικτύων και κατόπιν ορίζεται το domain name της υπηρεσίας. Ακόμα χρειάζεται να οριστεί αν η υπηρεσία θα δέχεται HTTPS αιτήματα (γραμμές 18 και 19)

```
1.  version: "3.6"

2.  services:
3.  traefik:
4.  image: traefik:v1.7.4-alpine
5.  hostname: traefik.tsere.com
6.  environment:
7.  TZ: Europe/Athens
8.  ports:
9.  "80:80"
10.   "443:443"
11.   volumes:
12.   /var/run/docker.sock:/var/run/docker.sock
13.   command: >
14.   --logLevel='DEBUG'
15.   --api.dashboard=true
16.   --InsecureSkipVerify=true
17.   --entryPoints='Name:http Address::80 Redirect.EntryPoint:https'
18.   --entryPoints='Name:https Address::443 TLS'
19.   --defaultentrypoints='http,https'
20.   --docker
21.   --docker.exposedbydefault=true
22.   --docker.watch=true
23.   --docker.swarmmode=false
24.   --docker.endpoint='unix:///var/run/docker.sock'
25.   networks:
26.   transit_idp:
27.   transit_thingsb:
28.   labels:
29.   "traefik.enable: true"

30.   networks:
31.   transit_idp:
32.   external: true
33.   transit_thingsb:
34. external: true
```

Από τον φάκελο που έχει αποθηκευτεί το docker-compose.yml αρχείο του treafik εκτελείται η εντολή:

```
sudo docker-compose up -d
```

Και τα αναμενόμενα αποτελέσματα είναι:

```
Pulling traefik (traefik:v1.7.4-alpine)...
v1.7.4-alpine: Pulling from library/traefik
4fe2ade4980c: Pull complete
8d9593d002f4: Pull complete
ab998c9a7159: Pull complete
a0da39a06413: Pull complete
Digest: sha256:652c9548e75f581f1fe5a7349cb96ceb3244debc4146cdaba02d61737aadbc1c
Status: Downloaded newer image for traefik:v1.7.4-alpine
Creating traefik_traefik_1 ... done
```

Με την ίδια εντολή αλλά στον φάκελο που έχει αποθηκευτεί το docker-compose.yml αρχείο του thingsboard, τα αναμενόμενα αποτελέσματα είναι:

```
Pulling app (thingsboard/tb-postgres:2.5.4)...
2.5.4: Pulling from thingsboard/tb-postgres
c0c53f743a40: Pull complete
66997431d390: Pull complete
0ea865e2909f: Pull complete
584bf23912b7: Pull complete
29a215b62e0a: Pull complete
44ba1e08385b: Pull complete
2172d131bfd4: Pull complete
6d9a54c8d9d9: Pull complete
124821e2a4c8: Pull complete
7b63bf35b688: Pull complete
04b8a2c2e373: Pull complete
7f4c239ed6d2: Pull complete
d10d1bc38cb8: Pull complete
be28a30f21ae: Pull complete
ff5e2bd74f65: Pull complete
c5e5b1bc7154: Pull complete
6c96dbb3f2cd: Pull complete
a8e29380d7ab: Pull complete
94364da7bd6a: Pull complete
e09587d71e04: Pull complete
2ded10dc5478: Pull complete
534e60894139: Pull complete
Digest: sha256:63a2b533c72f6c296f68a59ca409ef8375a26507d9e14e6a8093208b7ccd6fb4
Status: Downloaded newer image for thingsboard/tb-postgres:2.5.4
Creating thingsboard_app_1 ... done
```

Με την εντολή:

```
sudo docker ps
```

Γίνεται έλεγχος των containers των docker που εκτελούνται την δεδομένη χρονική στιγμή:

```
CONTAINER ID   IMAGE                         COMMAND                  CREATED          STATUS
PORTS                                                                                  NAMES
69ca369a02cc   traefik:v1.7.4-alpine         "/entrypoint.sh --lo…"   About a minute ago   Up About
a minute   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
traefik_traefik_1
0a2aa6df9a96   thingsboard/tb-postgres:2.5.4   "start-tb.sh"            2 minutes ago       Up 2
minutes       0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 0.0.0.0:5683->5683/udp, :::5683-
>5683/udp, 0.0.0.0:8080->9090/tcp, :::8080->9090/tcp   thingsboard_app_1
```

Στην συνέχεια χρειάζεται η προσθήκη στο τοπικό αρχείο /etc/hosts των παρακάτω εγγραφών έτσι ώστε να ανακατευθυνθούν τα αιτήματα προς την σωστή κατεύθυνση:

```
127.0.0.1       app.tsere.com
127.0.0.1       traefik.tsere.com
```

Ακολουθώντας τις οδηγίες που περιγράφονται στο [84] πραγματοποιείται η εγγραφή στην υπηρεσία Auth0. Από εκεί είναι απαραίτητα τα: clientName, clientId, clientSecret, accessTokenUri, authorizationUri, jwkSetUri και userInfoUri

Μετά απαιτείται να γίνει καθορισμός των παραπάνω στοιχείων στο αρχείο thingsboard.yml της πλατφόρμας (το αρχείο αυτό είναι υπεύθυνο για την λειτουργία της πλατφόρμας) με την παρακάτω εντολή γίνεται αντιγραφή του προηγούμενου αρχείου που βρίσκεται μέσα στον docker και στην τοποθεσία second-app_app_1:usr/share/thingsboard/conf/ σε ένα τοπικό αρχείο temp.yml.

```
sudo docker cp second-app_app_1:usr/share/thingsboard/conf/thingsboard.yml temp.yml
```

Μετά το άνοιγμα του αρχείου temp.yml απαιτούνται ορισμένες αλλαγές στις γραμμές 109 – 165. Παρακάτω φαίνεται το αρχικό αρχείο.

```
  109   oauth2:
  110     # Enable/disable OAuth 2 login functionality
  111     # For details please refer to https://thingsboard.io/docs/user-guide/oauth-2-support/
  112     enabled: "${SECURITY_OAUTH2_ENABLED:true}"
  113     # Redirect URL where access code from external user management system will be
processed
  114     loginProcessingUrl:
"${SECURITY_OAUTH2_LOGIN_PROCESSING_URL:/login/oauth2/code/}"
  115     # List of SSO clients
  116     clients:
  117       default:
  118         # Label that going to be show on login button - 'Login with {loginButtonLabel}'
  119         loginButtonLabel: "${SECURITY_OAUTH2_DEFAULT_LOGIN_BUTTON_LABEL:Auth0}"
  120         # Icon that going to be show on login button. Material design icon ID
(https://material.angularjs.org/latest/api/directive/mdIcon)
  121         loginButtonIcon: "${SECURITY_OAUTH2_DEFAULT_LOGIN_BUTTON_ICON:mdi:shield-
account}"
```

```
  122       clientName: "${SECURITY_OAUTH2_DEFAULT_CLIENT_NAME:My App}"
  123       clientId: "${SECURITY_OAUTH2_DEFAULT_CLIENT_ID:
vPYD5ajgab2YKs89Q0hO1OWxYEWMzJfM}"
  124       clientSecret: "${SECURITY_OAUTH2_DEFAULT_CLIENT_SECRET:
g6XeJuccYbclVQF94W7BqQ-osPVFT0f_V0og929tWKLfKygWQxHscvnuwon4CwyC}"
  125       accessTokenUri: "${SECURITY_OAUTH2_DEFAULT_ACCESS_TOKEN_URI:
https://tsere.eu.auth0.com/oauth/token}"
  126       authorizationUri: "${SECURITY_OAUTH2_DEFAULT_AUTHORIZATION_URI:
https://tsere.eu.auth0.com/authorize}"
  127       scope: "${SECURITY_OAUTH2_DEFAULT_SCOPE:openid,email,profile}"
  128       # Redirect URL that must be in sync with 'security.oauth2.loginProcessingUrl', but
domain name added
  129       redirectUriTemplate:
"${SECURITY_OAUTH2_DEFAULT_REDIRECT_URI_TEMPLATE:http://app.tsere.com/login/oauth2/c
ode/}"
  130       jwkSetUri: "${SECURITY_OAUTH2_DEFAULT_JWK_SET_URI:
https://tsere.eu.auth0.com/.well-known/jwks.json}"
  131       # 'authorization_code', 'implicit', 'refresh_token' or 'client_credentials'
  132       authorizationGrantType:
"${SECURITY_OAUTH2_DEFAULT_AUTHORIZATION_GRANT_TYPE:authorization_code}"
  133       clientAuthenticationMethod:
"${SECURITY_OAUTH2_DEFAULT_CLIENT_AUTHENTICATION_METHOD:post}" # basic or post
  134       userInfoUri: "${SECURITY_OAUTH2_DEFAULT_USER_INFO_URI:
https://tsere.eu.auth0.com/userinfo}"
  135       userNameAttributeName:
"${SECURITY_OAUTH2_DEFAULT_USER_NAME_ATTRIBUTE_NAME:email}"
  136       mapperConfig:
  137         # Allows to create user if it not exists
  138         allowUserCreation:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_ALLOW_USER_CREATION:true}"
  139         # Allows user to setup ThingsBoard internal password and login over default Login
window
  140         activateUser: "${SECURITY_OAUTH2_DEFAULT_MAPPER_ACTIVATE_USER:false}"
  141         # Mapper type of converter from external user into internal - 'basic' or 'custom'
  142         type: "${SECURITY_OAUTH2_DEFAULT_MAPPER_TYPE:basic}"
  143         basic:
  144           # Key from attributes of external user object to use as email
  145           emailAttributeKey:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_EMAIL_ATTRIBUTE_KEY:email}"
  146           firstNameAttributeKey:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_FIRST_NAME_ATTRIBUTE_KEY: email}"
  147           lastNameAttributeKey:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_LAST_NAME_ATTRIBUTE_KEY:email}"
  148           # Strategy for generating Tenant from external user object - 'domain', 'email' or
'custom'
  149           # 'domain' - name of the Tenant will be extracted as domain from the email of the
user
  150           # 'email' - name of the Tenant will email of the user
  151           # 'custom' - please configure 'tenantNamePattern' for custom mapping
  152           tenantNameStrategy:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_TENANT_NAME_STRATEGY:domain}"
```

```
  153           # %{attribute_key} as placeholder for attribute value of attributes of external user
object
  154           tenantNamePattern:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_TENANT_NAME_PATTERN:}"
  155           # If this field is not empty, user will be created as a user under defined Customer
  156           # %{attribute_key} as placeholder for attribute value of attributes of external user
object
  157           customerNamePattern:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_CUSTOMER_NAME_PATTERN:}"
  158           # If this field is not empty, user will be created with default defined Dashboard
  159           defaultDashboardName:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_DEFAULT_DASHBOARD_NAME:}"
  160           # If this field is set 'true' along with non-empty 'defaultDashboardName', user will
start from the defined Dashboard in fullscreen mode
  161           alwaysFullScreen:
"${SECURITY_OAUTH2_DEFAULT_MAPPER_BASIC_ALWAYS_FULL_SCREEN:false}"
  162         custom:
  163           url: "${SECURITY_OAUTH2_DEFAULT_MAPPER_CUSTOM_URL:}"
  164           username: "${SECURITY_OAUTH2_DEFAULT_MAPPER_CUSTOM_USERNAME:}"
  165           password: "${SECURITY_OAUTH2_DEFAULT_MAPPER_CUSTOM_PASSWORD:}"
```

Από το παραπάνω αρχείο αφαιρούνται από όλα τα πεδία το κείμενο "${XXXXXXXXXXXX:}".

Κατόπιν το πεδίο enabled στην γραμμή 112 γίνεται true, στην γραμμή 119 είναι προαιρετικός ο ορισμός ενός ονόματος εμφάνισης της πλατφόρμας αυθεντικοποίησης και στην 121 ο ορισμός ενός αντίστοιχου εικονιδίου. Στην γραμμή 122 ορίζουμε το clientName, στην 123 και 124 το clientId και το clientSecret αντίστοιχα. Στις γραμμές 125, 126, 129, 130 και 134 ορίζονται τα κατάλληλα uri. Τέλος από την γραμμή 135 και έπειτα ορίζονται οι πληροφορίες σχετικά με τους χρήστες.

Το τελικό αρχείο είναι παρόμοιο με:

```
  109   oauth2:
  110     # Enable/disable OAuth 2 login functionality
  111     # For details please refer to https://thingsboard.io/docs/user-guide/oauth-2-support/
  112     enabled: true
  113     # Redirect URL where access code from external user management system will be
processed
  114     loginProcessingUrl: /login/oauth2/code/
  115     # List of SSO clients
  116     clients:
  117       default:
  118         # Label that going to be show on login button - 'Login with {loginButtonLabel}'
  loginButtonLabel: Auth0
  120         # Icon that going to be show on login button. Material design icon ID
(https://material.angularjs.org/latest/api/directive/mdIcon)
  121         loginButtonIcon: mdi:shield-account
  122         clientName: App
  123         clientId: Id
  124         clientSecret: Secret
  125         accessTokenUri: https://tsere.eu.auth0.com/oauth/token
  126         authorizationUri: https://tsere.eu.auth0.com/authorize
  127         scope: openid,email,profile
```

```
  128       # Redirect URL that must be in sync with 'security.oauth2.loginProcessingUrl', but
domain name added
  129       redirectUriTemplate: http://app.tsere.com/login/oauth2/code/
  130       jwkSetUri: https://tsere.eu.auth0.com/.well-known/jwks.json
  131       # 'authorization_code', 'implicit', 'refresh_token' or 'client_credentials'
  132       authorizationGrantType: authorization_code
  133       clientAuthenticationMethod: post # basic or post
  134       userInfoUri: https://tsere.eu.auth0.com/userinfo
  135       userNameAttributeName: email
  136       mapperConfig:
  137         # Allows to create user if it not exists
  138         allowUserCreation: true
  139         # Allows user to setup ThingsBoard internal password and login over default Login
window
  140         activateUser: false
  141         # Mapper type of converter from external user into internal - 'basic' or 'custom'
  142         type: basic
  143         basic:
  144           # Key from attributes of external user object to use as email
  145           emailAttributeKey: email
  146           firstNameAttributeKey: email
  147           lastNameAttributeKey: email
  148           # Strategy for generating Tenant from external user object - 'domain', 'email' or
'custom'
  149           # 'domain' - name of the Tenant will be extracted as domain from the email of the
user
  150           # 'email' - name of the Tenant will email of the user
  151           # 'custom' - please configure 'tenantNamePattern' for custom mapping
  152           tenantNameStrategy: domain
  153           # %{attribute_key} as placeholder for attribute value of attributes of external user
object
  154           tenantNamePattern:
  155           # If this field is not empty, user will be created as a user under defined Customer
  156           # %{attribute_key} as placeholder for attribute value of attributes of external user
object
  157           customerNamePattern:
  158           # If this field is not empty, user will be created with default defined Dashboard
  159           defaultDashboardName:
  160           # If this field is set 'true' along with non-empty 'defaultDashboardName', user will
start from the defined Dashboard in fullscreen mode
  161           alwaysFullScreen: false
  162         custom:
  163           url:
  164           username:
  165           password:
```

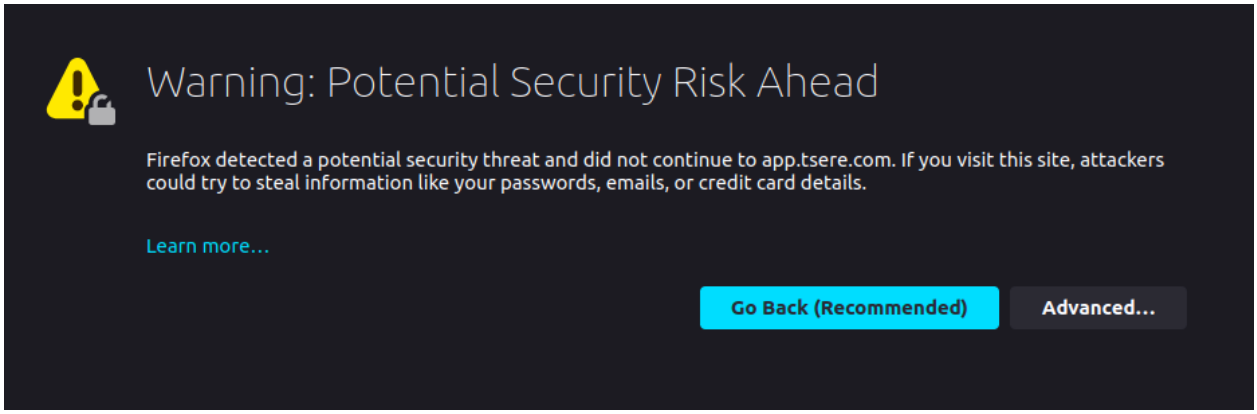
Με την επόμενη εντολή και αφού έχουν αποθηκευτεί οι αλλαγές στο τοπικό yml αρχείο πραγματοποιείται η μεταφορά του αρχείου temp.yml στον docker second-app_app_1 και στο σημείο: usr/share/thingsboard/conf/thingsboard.yml

```
sudo docker cp temp.yml second-app_app_1:usr/share/thingsboard/conf/thingsboard.yml
```

Τέλος με την επόμενη εντολή πραγματοποιείται η επανεκκίνηση του docker προκείμενου να καθιερωθούν οι αλλαγές στο σύστημα.

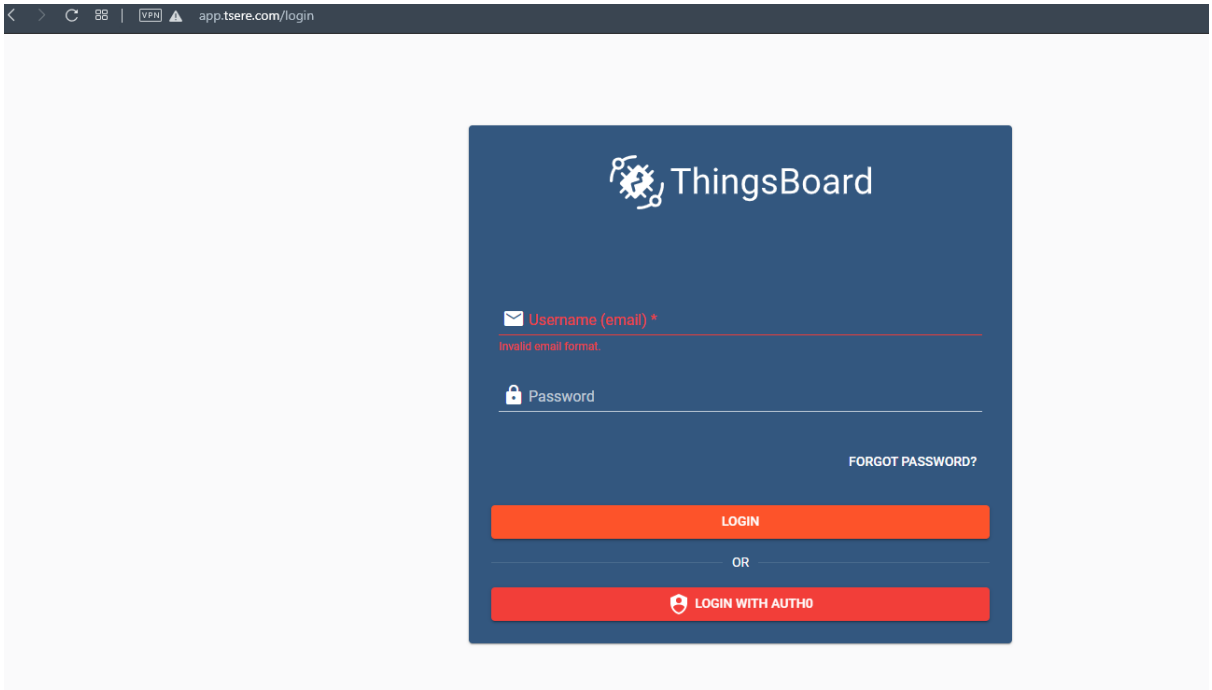
```
sudo docker restart second-app_app_1
```

Κατόπιν μέσω κάποιου προγράμματος περιήγησης γίνεται μετάβαση στο url app.tsere.com. Το παρακάτω μήνυμα (σχήμα 15) φαίνεται αφού το SSL πιστοποιητικό δεν αναγνωρίζεται ως έγκυρο μιας και είναι self-signed. Ωστόσο γίνεται αγνόηση του προχωρώντας με το κουμπί Advanced και Accept the risk and Continue.



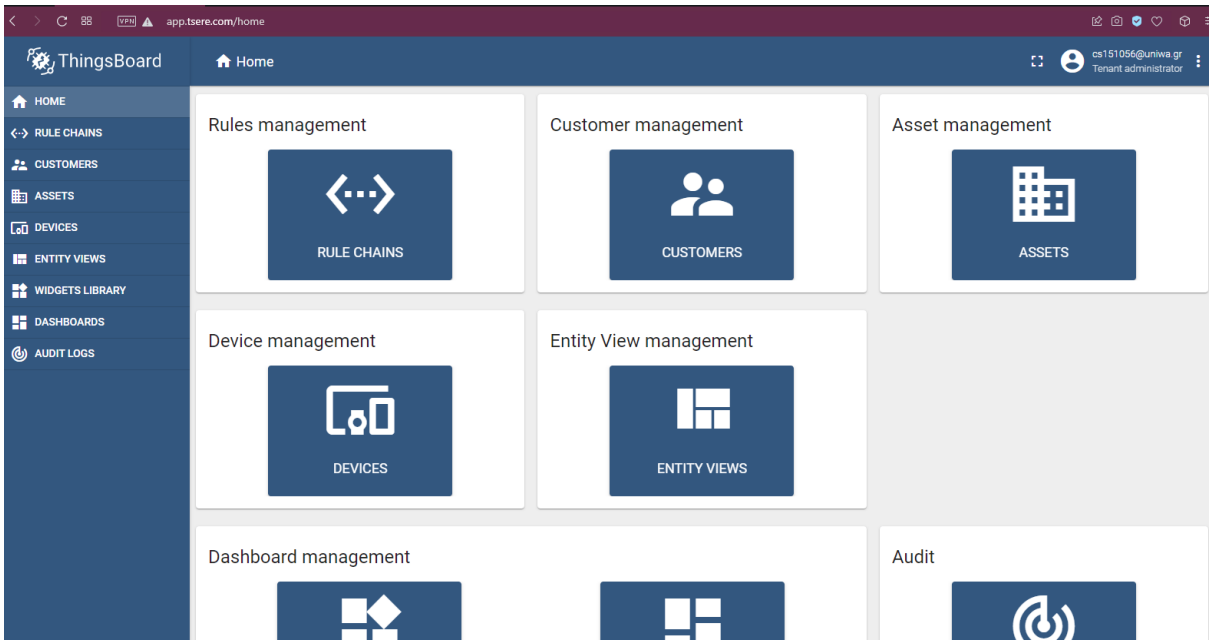
Σχήμα 15 Self signed Certificate

Μεταβαίνοντας στην σελίδα εισόδου στην πλατφόρμα (σχήμα 16) και επιλέγοντας το Login with Auth0 πραγματοποιείται ανακατεύθυνση στην σελίδα https://tsere.eu.auth0.com/login όπου είτε γίνεται είσοδος του χρήστη είτε εγγραφή.



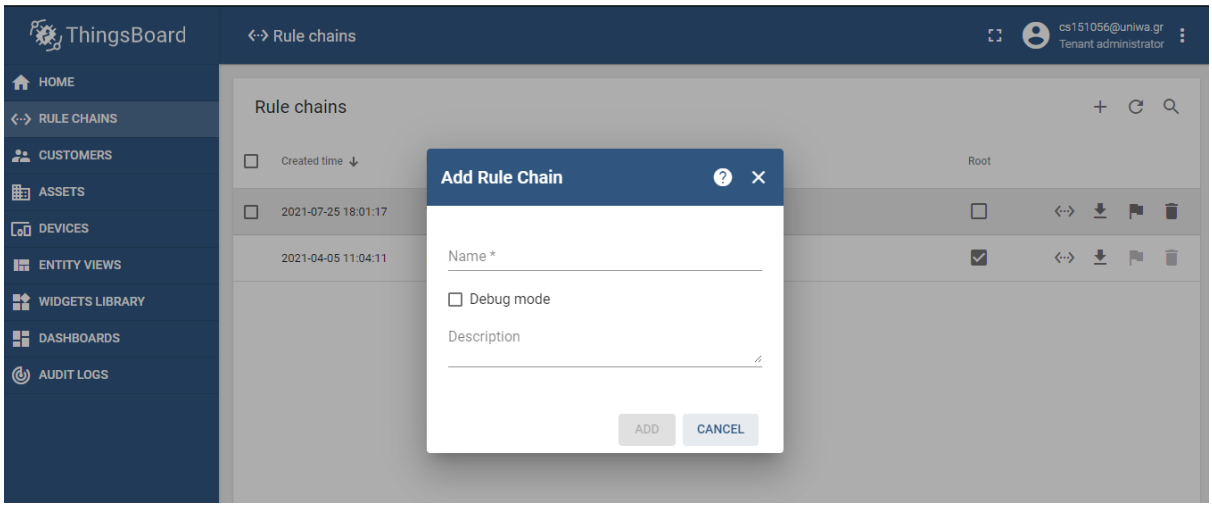
Σχήμα 16 Είσοδος στην πλατφόρμα

Μετά την είσοδο στην πλατφόρμα εμφανίζεται η αρχική οθόνη όπως φαίνεται στο σχήμα 17.



Σχήμα 17 Αρχική οθόνη

Προκειμένου να δημιουργηθεί ένας κανόνας που ελέγχει αν η θερμοκρασία βρίσκεται ανάμεσα στους 8 και 35 βαθμούς κελσίου και σε περίπτωση που βρίσκεται εκτός αυτού του ορίου να δημιουργεί μια ειδοποίηση, χρειάζεται από η μετάβαση στην καρτέλα Rule chains και να πατώντας το κουμπί + εμφανίζεται ένα αναδυόμενο πλαίσιο όπου δίνεται ένα κατάλληλο όνομα με το κουμπί add (όπως φαίνεται στο σχήμα 18) και δημιουργείται ο κανόνας.

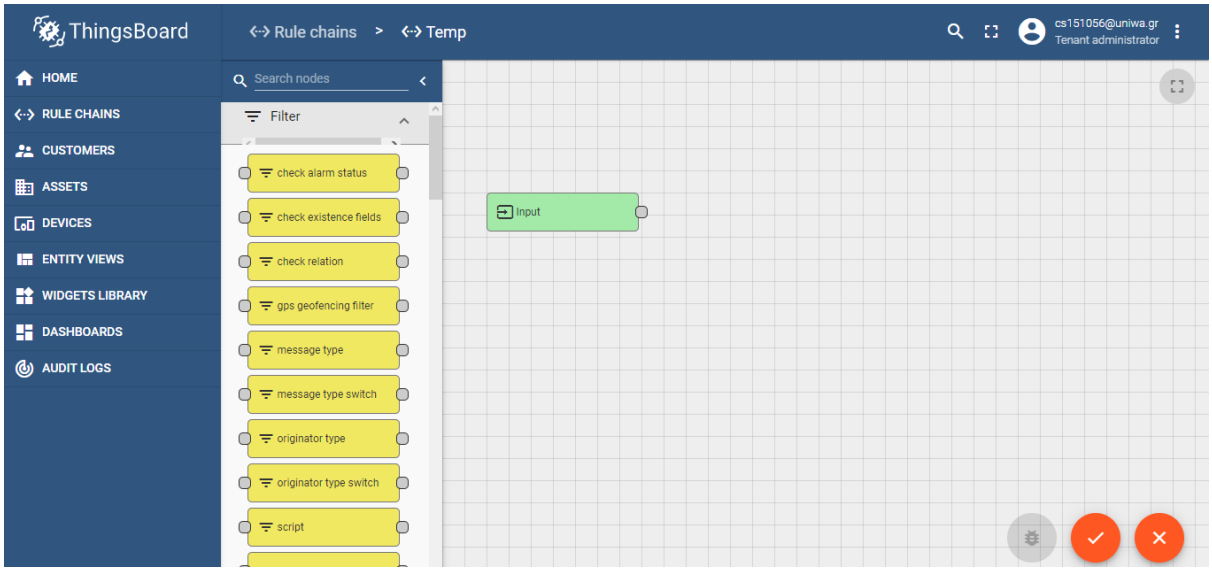


Σχήμα 18 Καρτέλα Rule chains

Κατόπιν γίνεται επιλογή του νέου κανόνα και με το κουμπί Open rule chain εμφανίζεται η καρτέλα προσαρμογής του κανόνα. Στην αναζήτηση όπως φαίνεται στο σχήμα 19 και με την τοποθέτηση του κειμένου «script» και με drag & drop πραγματοποιείται η τοποθέτηση του πλαίσιο μετά το input. Δίνοντας ένα όνομα και τοποθετώντας τον παρακάτω κώδικα στο

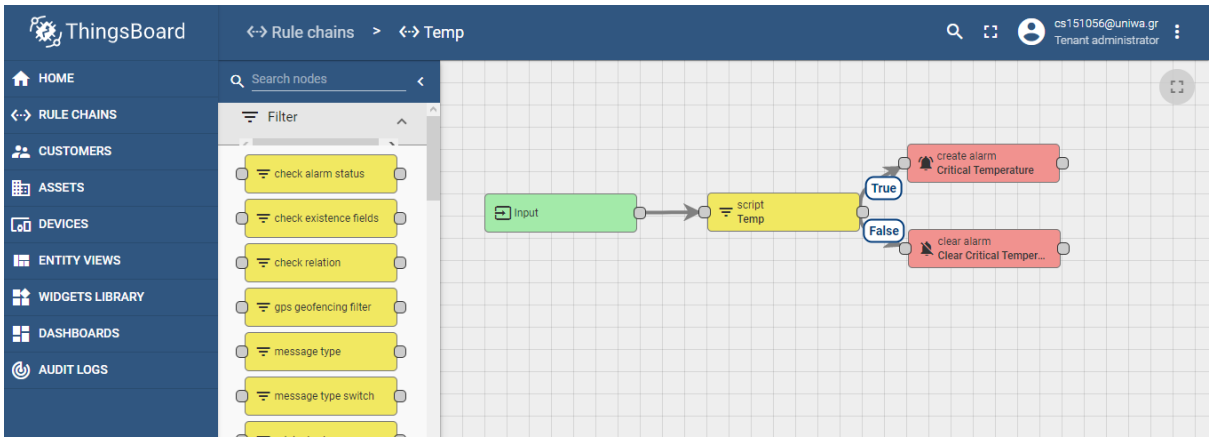
ανάλογο πλαίσιο γίνεται η τοποθέτηση του κώδικα που ελέγχει τις τιμές εισόδου και επιστρέφει true ή false αναλόγως εάν η τιμή βρίσκεται εντός ή εκτός του εύρους που έχει οριστεί:

```
return msg.temperature < 8 || msg.temperature > 35;
```



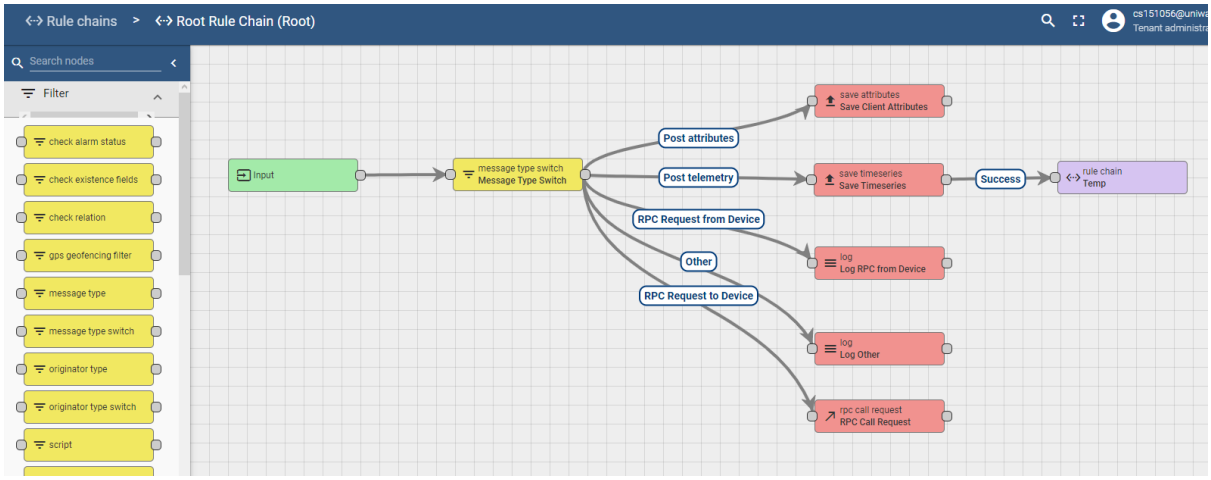
Σχήμα 19 Δημιουργία Κανόνα-α

Στην συνέχεια γίνεται αναζήτηση των πλαισίων create alarm, clear alarm και τοποθετηση τους μετά το πλαίσιο του script, δίνοντας και τα κατάλληλα ονόματα στα πλαίσια αυτά και αφού δημιουργηθούν οι κατάλληλες συνδέσεις όπως φαίνονται στο σχήμα 20 ολοκληρώνεται η δημιουργία του κανόνα.



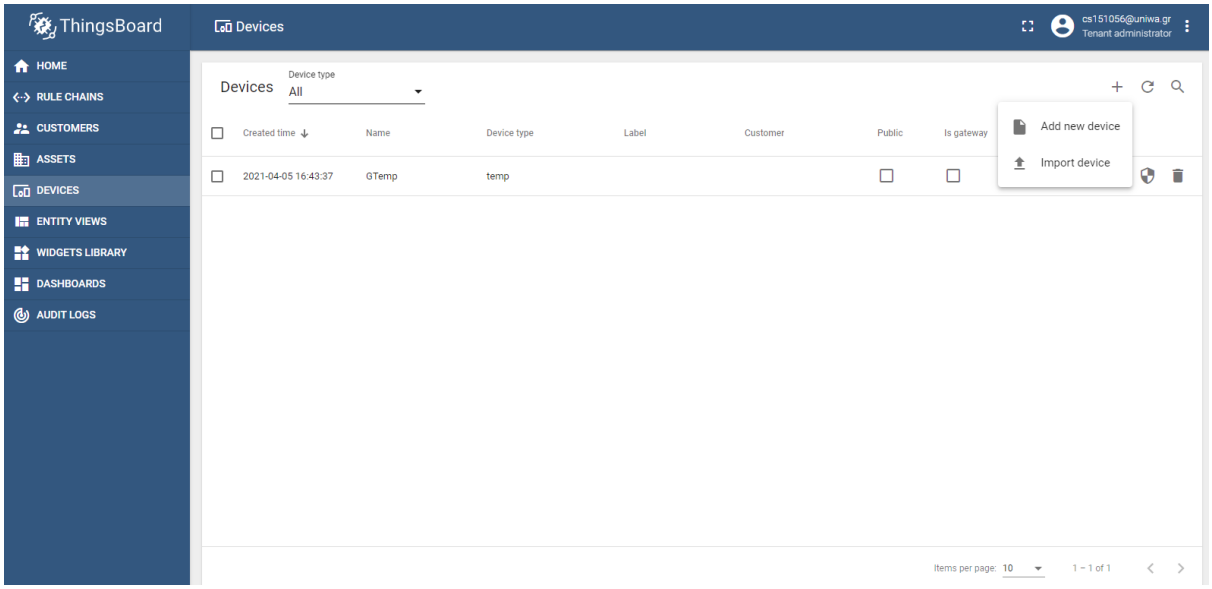
Σχήμα 20 Δημιουργία Κανόνα-β

Τέλος απατείτε να προσαρμοστεί κατάλληλα και ο Root Rule Chain. Οπότε γίνεται επιστροφή στην καρτέλα Rule Chains και άνοιγμα του Root Rule Chain. Στην συνέχεια απαιτείται η επιλογή ενός πλαισιου rule chain και η σύνδεση του με τον rule chain «Save Timeseries» όπως φαίνεται στο σχήμα 21.



Σχήμα 21 Δημιουργία Κανόνα-γ

Στην συνέχεια είναι αναγκαία η δημιουργία μιας νέας συσκευή στο σύστημα μεταβαίνοντας στην καρτέλα devices. Πατώντας το κουμπί + και επιλέγοντας το κουμπί add new device όπως φαίνεται στο παρακάτω σχήμα, στο αναδυόμενο παράθυρο δίνεται το όνομα της συσκευής όπως και ένας τύπος συσκευής.



Σχήμα 22 Δημιουργία συσκευής

Ο κώδικας που χρείαζεται για να συνδεθεί η συσκευή στην πλατφόρμα είναι:

```
1  import glob
2  import time
3  import os
4  import sys
5  import paho.mqtt.client as mqtt
6  import json
7  import RPi.GPIO as GPIO
8  import dht11
9  # initialize GPIO
10 GPIO.setwarnings(False)
11 GPIO.setmode(GPIO.BCM)
```

```
GPIO.cleanup()
instance = dht11.DHT11(pin = 24)

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

THINGSBOARD_HOST = 'YOUR_THINGSBOARD_HOST'
ACCESS_TOKEN = 'YOUR_ACCESS_TOKEN'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        #temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c

#INTERVAL=3600




#next_reading = time.time()

client = mqtt.Client()

# Set access token
client.username_pw_set(ACCESS_TOKEN)

# Connect to ThingsBoard using default MQTT port and 60 seconds keepalive interval
client.connect(THINGSBOARD_HOST, 1883, 60)

client.loop_start()
i=0
try:
#    while True:
        temperature = read_temp() ; sensor_data = {'temperature':
0,'temperature_dht':0,'humidity_dht':0}
        print(u"Temperature: {:g}\u00b0C".format(temperature))
        result = instance.read()
        while result.is_valid() == False or i<10:
```

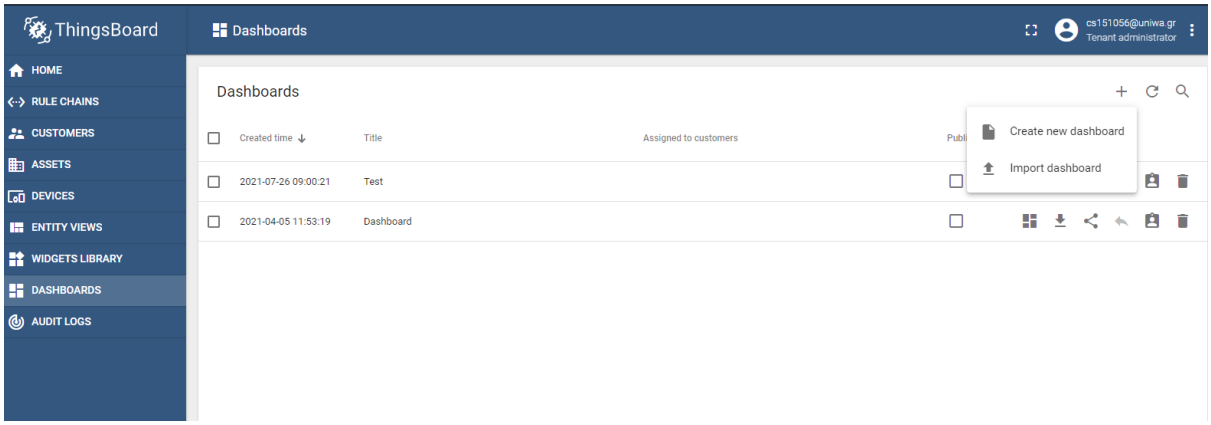
```
62          if result.is_valid():
63            break;
64          time.sleep(1)
65          result = instance.read()
66          i =+ 1
67        #print (result.is_valid())
68        print("\nDHT: Temperature: {:.1f} C    Humidity: {}% \n".format(result.temperature,
result.humidity))
69        sensor_data['temperature'] = temperature
70        sensor_data['temperature_dht'] = result.temperature
71        sensor_data['humidity_dht'] = result.humidity
72        # Sending temperature data to ThingsBoard
73        client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
74
75        #next_reading += INTERVAL
76        #sleep_time = next_reading-time.time()
77        #if sleep_time > 0:
78        #    time.sleep(sleep_time)
79  except KeyboardInterrupt:
80     pass
81
82  client.loop_stop()
83  client.disconnect()
```

Προυπόθεση εκτέλεσης του παραπάνω κώδικα αποτελεί η λήψη των βιβλιοθηκών glob, time, os, sys, paho.mqtt.client as mqtt, json, RPi.GPIO as GPIO, dht11.

Στην καρτέλα της συσκευής βρίσκεται το ACCESS_TOKEN της συσκευής προκείμενου να γίνει η αυθεντικοποίησή της από την πλατφόρμα. Πραγματοποιείται τροποποίηση της γραμμής 19 κατάλληλα ώστε να κατευθύνει τα αιτήματα προς την πλατφόρμα και στην 20 θέτουμε το ACCESS_TOKEN της συσκευής. Στην γραμμή 52 γίνεται η σύνδεση με την πλατφόρμα και στις γραμμές 58 – 66 πραγματοποιείται η ανάγνωση των δεδομένων από τους αισθητήρες και ο έλεγχος των δεδομένων. Στις γραμμές 69 – 73 γίνεται η αποστολή των δεδομένων στην πλατφόρμα. Οι γραμμές 40, 44, 57, και 75 – 79 που είναι σε σχόλια χρησιμοποιούνται εάν είναι επιθυμητή η εκτέλεση του κώδικα συνεχώς, η μετάβαση του κατάσταση sleep για ένα χρονικό σημείο Χ που καθορίζεται στην γραμμή 40 και η εκτέλεση του ξανά μετά την πάροδο του παραπάνω ορισμένου διαστήματος. Ωστόσο η καλύτερη πρακτική είναι η τοποθέτηση της παρακάτω εντολής στο crontab και επομένως η εκτέλεση όλου του κώδικα κάθε ένα Χ χρονικό διάστημα (π.χ. κάθε ώρα το λεπτό 00) που θα οριστεί.

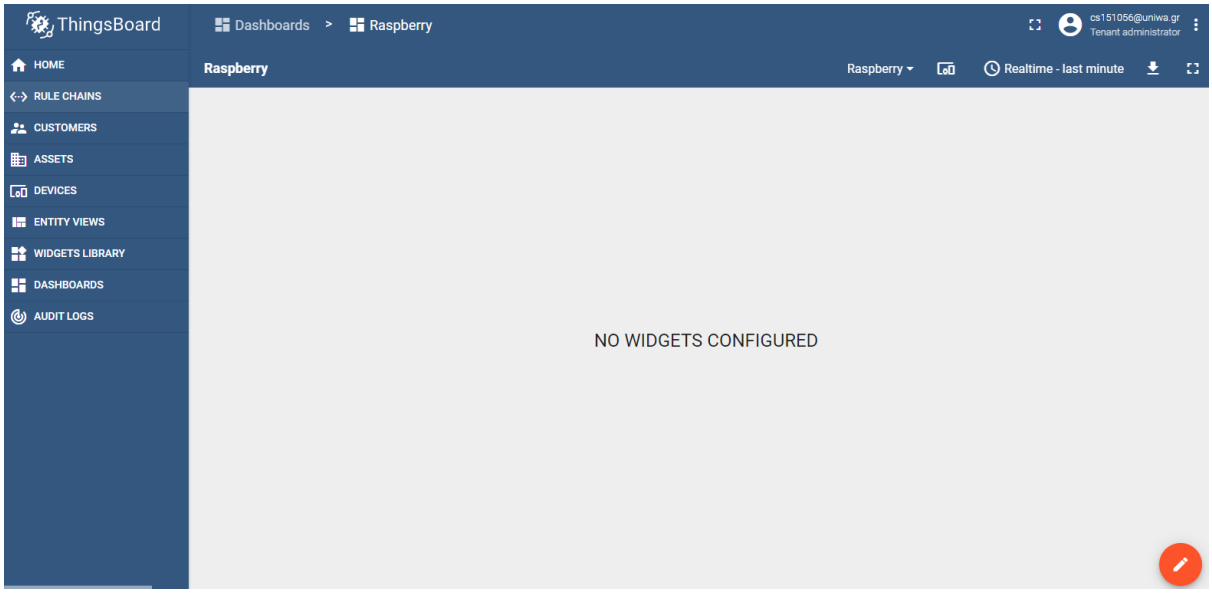
```
0 * * * * /usr/bin/python3 /home/pi/temp.py
```

Στην συνέχεια αναλύεται η δημιουργία ενός dashboard μέσω της καρτέλας dashboards. Πατώντας το κουμπί + και επιλέγοντας το create new dashboard (σχήμα 23) στο αναδυόμενο πλαίσιο ορίζεται ένα όνομα.



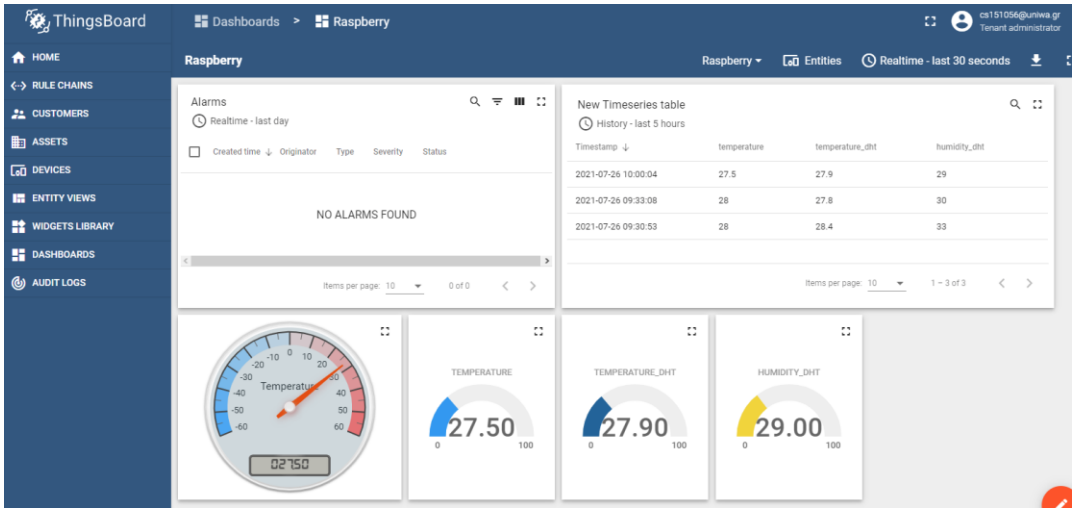
Σχήμα 23 Δημιουργία dashboard

Και στην συνέχεια επιλέγοντας το dashboard που μόλις δημιουργήθηκε και το κουμπί Open dashboard πραγματοποιείται μετάβαση στο dashboard. Κατόπιν επιλέγοντας το μολυβάκι παρέχεται η δυνατότητα επεξεργασίας του dashboard και από το κουμπί + και create new widget δίνεται η δυνατότητα να προσθήκης μιας ευρείας γκάμας από widget όπως alram, timeseries, gauge κ.α.



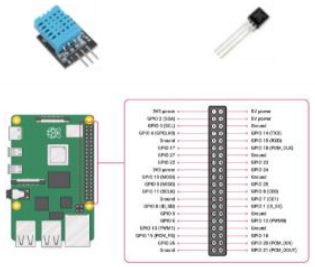
Σχήμα 24 Δημιουργία widget-1

Μετά από κατάλληλη προσαρμογή των widget εμφανίζεται η παρακάτω εικόνα (σχήμα 25).



Σχήμα 25 Δημιουργία widget-2

Οι συσκευές που χρησιμοποιήθηκαν είναι: Raspberry Pi 4 - 4 GB, Temperature and Humidity Sensor DHT11 Module, Temperature Sensor DS18B20 καθώς και ένα εικονικό μηχάνημα που παρέχεται από την υπηρεσία Cyclades του okeanos.grnet. Στο σχήμα 26 απεικονίζονται το Raspberry Pi και οι αισθητήρες που χρησιμοποιήθηκαν.



Σχήμα 26 Raspberry Pi 4 και αισθητήρες

Συμπεράσματα και προοπτικές

Σύνοψη της διπλωματικής εργασίας

Από την εκπόνηση της παρούσας διπλωματικής εργασίας γίνεται αντιληπτό πως η εξασφάλιση της ασφάλειας ενός συστήματος διαχείρισης IoT συσκευών αποτελεί μια πολύπλοκη διαδικασία. Μια πλατφόρμα IoT είναι ένας συνδυασμός ανόμοιων αντικειμένων και τεχνολογιών όπως το cloud, big data, κινητές συσκευές κ.α.

Το πρωταρχικό βήμα είναι η εφαρμογή του συνόλου των μηχανισμών εξασφάλισης της ασφάλειας του συστήματος που μελετήθηκαν σε προηγούμενο κεφάλαιο. Κατόπιν απαιτείται συνεχής και πλήρης έλεγχος όλων των μερών του συστήματος αφού μια ευπάθεια εκθέτει το σύστημα σε κίνδυνο αν δεν αντιμετωπιστεί άμεσα.

Τέλος είναι σημαντικό να τονιστεί ότι δεν αρκεί μια καθολική εφαρμογή ορισμένων προτύπων για την πλήρη υλοποίηση μιας IoT πλατφόρμας. Αντίθετα χρειάζεται η μελέτη των αναγκών του συστήματος και στην συνέχεια η εφαρμογή των ανάλογων λύσεων.

Προοπτικές

Το IoT έχει ήδη εφαρμοστεί σε πολλές πτυχές της καθημερινότητας και στα επόμενα έτη αναμένεται να ενσωματωθεί σε σχεδόν όλες τις καθημερινές διαδικασίες διευκολύνοντας

έτσι την ζωή των ανθρώπων. Σε αυτό το πλαίσιο είναι αναγκαίο να μελετηθούν ακόμα περισσότερο τα πρότυπα που θα εξασφαλίζουν την ασφαλή και πλήρη λειτουργία όλων των μερών του συστήματος.

Μια από τις μεγαλύτερες προκλήσεις στην ασφάλεια του IoT αποτελεί η επιτυχής αντιμετώπιση των malware που στοχεύουν τις συσκευές με περιορισμένους πόρους.

Βιβλιογραφία

[1] Raji, R., 1994. Smart networks for control. [online] Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/document/284793 [Accessed 18 July 2021].

[2] RFID JOURNAL. n.d. That 'Internet of Things' Thing. [online] Available at: <https://www.rfidjournal.com/that-internet-of-things-thing [Accessed 20 July 2021].

[3] Minerva, R., Biru, A. and Rotondi, D., 2015. Towards a definition of the Internet of Things (IoT). [online] Institute of Electrical and Electronics Engineers. Available at: <https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf [Accessed 20 July 2021].

[4] IEEE. 2019. Heterogeneous Integration Roadmap 2019 Edition. [online] Available at: <https://eps.ieee.org/images/files/HIR_2019/HIR1_ch03_iot.pdf> [Accessed 23 July 2021].

[5] Teicher, J., 2018. The little-known story of the first IoT device. [online] IBM. Available at: <https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/> [Accessed 20 July 2021].

[6] Fraser, Q., 1995. Trojan Room Coffee Pot Biography. [online] University of Cambridge. Available at: <https://www.cl.cam.ac.uk/coffee/qsf/coffee.html> [Accessed 20 July 2021].

[7] Asme.org. 2021. Top 10 Smart Cities in the World - ASME. [online] Available at: <https://www.asme.org/topics-resources/content/top-10-growing-smart-cities [Accessed 16 May 2021].

[8] Studio, 2., 2021. Smart Nation | Lucky Joint Construction Pte Ltd - Structured Cabling, FRS, FRC, Optical Fiber Cable, Home Networking. [online] Luckyjoint.com.sg. Available at: https://www.luckyjoint.com.sg/smart-nation [Accessed 10 May 2021]

[9] Ec.europa.eu. 2021. [online] Available at: https://ec.europa.eu/environment/europeangreencapital/wp-content/uploads/2016/12/Oslo_EGCA2019_finalist-presentation.pdf [Accessed 16 May 2021].

[10] Architect. 2021. Oslo Airport City Masterplan. [online] Available at: https://www.architectmagazine.com/project-gallery/oslo-airport-city-masterplan_o [Accessed 16 May 2021].

[11] Intelligent Transport. 2021. Copenhagen's drive to become connected and carbon neutral. [online] Available at: https://www.intelligenttransport.com/transport-articles/95032/copenhagens-drive-to-become-connected-and-carbon-neutral/ [Accessed 23 May 2021].

[12] Privacyshield.gov. 2021. Denmark - Smart Cities | Privacy Shield. [online] Available at: https://www.privacyshield.gov/article?id=Denmark-Smart-Cities [Accessed 23 May 2021].

[13] Thales Group. 2021. Secure, sustainable smart cities and the IoT. [online] Available at: https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/inspired/smart-cities [Accessed 2 June 2021].

[14] Gehl. 2021. Smart City Copenhagen. [online] Available at: https://gehlpeople.com/blog/smart-copenhagen-whats/ [Accessed 23 May 2021].

[15] Lund, H., Østergaard, P., Connolly, D. and Mathiesen, B., 2021. Smart energy and smart energy systems.

[16] Li, J. and Guo, X., 2021. COVID-19 Contact-tracing Apps: a Survey on the Global Deployment and Challenges. [online] arXiv.org. Available at: https://arxiv.org/abs/2005.03599 [Accessed 28 May 2021].

[17] Lanke, N., & Koul, S. (2013). Smart traffic management system. International Journal of Computer Applications, 75(7).

[18] Krasniqi, X. and Hajrizi, E., 2021. Use of IoT Technology to Drive the Automotive Industry from Connected to Full Autonomous Vehicles.

[19] Raj, P. and Raman, A., 2017. The Internet of things. 1st ed. CRC Press.

[20] Omg.org. 2015. OMG Data Distribution Service (DDS) Version 1.4. [online] Available at: https://www.omg.org/spec/DDS/1.4/PDF [Accessed 30 June 2021].

[21] Omg.org. 2018. DDS Security Version 1.1. [online] Available at: https://www.omg.org/spec/DDS-SECURITY/1.1 [Accessed 30 June 2021].

[22] Saint-Andre, P., 2011. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc6121 [Accessed 20 July 2021].

[23] Millard, P., Saint-Andre, P. and Meijer, R., 2006. XEP-0060: Publish-Subscribe. [online] Xmpp.org. Available at: https://xmpp.org/extensions/attic/xep-0060-1.9.html [Accessed 20 July 2021].

[24] Fielding, R., 2000. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. UNIVERSITY OF CALIFORNIA, IRVINE.

[25] Shelby, Z., Hartke, K. and Bormann, C., 2014. The Constrained Application Protocol (CoAP). [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc7252 [Accessed 20 July 2021].

[26] Shelby, Z., 2012. CoRE Link Format. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/RFC6690 [Accessed 20 July 2021].

[27] Cheshire, S. and Krochmal, M., 2013. Multicast DNS. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc6762 [Accessed 20 July 2021].

[28] Cheshire, S. and Krochmal, M., 2013. DNS-Based Service Discovery. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc6763 [Accessed 20 July 2021].

[29] Internet Engineering Task Force. 1981. INTERNET PROTOCOL. [online] Available at: https://datatracker.ietf.org/doc/html/rfc791 [Accessed 30 June 2021].

[30] Deering, S. and Hinden, R., 1998. Internet Protocol, Version 6 (IPv6) Specification. [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/html/rfc2460 [Accessed 30 June 2021].

[31] Deering, S. and Hinden, R., 1998. Internet Protocol, Version 6 (IPv6) Specification rfc2460. [online] Datatracker.ietf.org. Available at: https://datatracker.ietf.org/doc/html/rfc2460 [Accessed 4 June 2021].

[32] Hui, J. and Thubert, P., 2011. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. [online] Datatracker.ietf.org. Available at: https://datatracker.ietf.org/doc/html/rfc6282 [Accessed 27 June 2021].

[33] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP. and Alexander, R., 2012. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. [online] Tools.ietf.org. Available at: https://tools.ietf.org/pdf/rfc6550.pdf [Accessed 4 June 2021].

[34] Katz, D. and Ward, D., 2010. Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop). [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc5881 [Accessed 4 June 2021].

[35] Hui, J. and Vasseur, J., 2012. rfc6553. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc6553 [Accessed 13 June 2021].

[36] Levis, P., Clausen, T., Hui, J., Gnawali, O. and Ko, J., 2011. The Trickle Algorithm. [online] Internet Engineering Task Force (IETF). Available at: https://datatracker.ietf.org/doc/html/rfc6206 [Accessed 13 June 2021].

[37] Draves, R. and Thaler, D., 2021. Default Router Preferences and More-Specific Routes. [online] Institute of Electrical and Electronics Engineers. Available at: https://datatracker.ietf.org/doc/html/rfc4191 [Accessed 20 July 2021].

[38] Narten, T., Nordmark, E., Simpson, W. and Soliman, H., 2007. Neighbor Discovery for IP version 6 (IPv6). [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/html/RFC4861 [Accessed 20 July 2021].

[39] Thomson, S., Narten, T. and Jinmei, T., 2005. IPv6 Stateless Address Autoconfiguration. [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/html/RFC4862 [Accessed 20 July 2021].

[40] Perkins, C., Johnson, D. and Arkko, J., 2011. Mobility Support in IPv6. [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/html/RFC6275 [Accessed 20 July 2021].

[41] IEEE Std 802.15.4x-2019 (Amendment to IEEE 802.15.4-2015 as amended by IEEE 802.15.4n-2016, IEEE 802.15.4q-2016, IEEE 802.15.4u-2016, IEEE 802.15.4t-2017, IEEE 802.15.4v-2017, IEEE 802.15.4s-2018, and IEEE 802.15.4-2015/Cor. 1-2018, 2019. 802.15.4x-2019 - IEEE Standard for Low-Rate Wireless Networks - Amendment 7: Defining Enhancements to the Smart Utility Network (SUN) Physical Layers (PHYs) Supporting up to 2.4 Mb/s Data Rates. [online] Available at: https://ieeexplore.ieee.org/document/8700703 [Accessed 27 June 2021].

[42] Devadiga, K., n.d. IEEE 802.15.4 and the Internet of things. [online] Aalto University Wiki. Available at: https://wiki.aalto.fi/download/attachments/59704179/devadiga-802-15-4-and-the-iot.pdf?version=1 [Accessed 20 July 2021].

[43] Khan, A., Qadeer, M., Ansari, M. and Waheed, S., 2009. 4G as a Next Generation Wireless Network. In: 2009 International Conference on Future Computer and Communication. [online] Kuala Lumpar, Malaysia: Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/document/5189800 [Accessed 18 July 2021].

[44] Hajlaoui, E., Zaier, A., Khlifi, A., Ghodhbane, J., Hamed, M. and Sbita, L., 2020. 4G and 5G technologies: A Comparative Study. In: 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). [online] Sousse, Tunisia: Institute of Electrical and Electronics Engineers, pp.1-6. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9231605 [Accessed 18 July 2021].

[45] Schulz, P., Matthe, M., Klessig, H., Simsek, M., Fettweis, G., Ansari, J., Ashraf, S., Almeroth, B., Voigt, J., Riedel, I., Puschmann, A., Mitschele-Thiel, A., Muller, M., Elste, T. and Windisch, M., 2017. Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. [online] Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/abstract/document/7842415 [Accessed 18 July 2021].

[46] Mealling, M., 2008. A Uniform Resource Name Namespace for the EPCglobal Electronic Product Code (EPC) and Related Standards. [online] Datatracker.ietf.org. Available at: https://datatracker.ietf.org/doc/html/rfc5134 [Accessed 27 June 2021].

[47] Zigbee Alliance. n.d. When to Use Zigbee - Zigbee Alliance. [online] Available at: https://zigbeealliance.org/why-zigbee/when-to-use/ [Accessed 27 June 2021].

[48] Mohanty, S., 2010. Energy Efficient Routing Algorithms for Wireless Sensor Networks and Performance Evaluation of Quality of Service for IEEE 802.15.4 Networks. [online] ResearchGate. Available at: https://www.researchgate.net/publication/47737848_Energy_Efficient_Routing_Algorithms

_for_Wireless_Sensor_Networks_and_Performance_Evaluation_of_Quality_of_Service_for_IEEE_802154_Networks [Accessed 20 July 2021].

[49] Surendran, S., Nassef, A. and Beheshti, B., 2018. A survey of cryptographic algorithms for IoT devices. In: 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). [online] Farmingdale, NY, USA: Institute of Electrical and Electronics Engineers, pp.1-8. Available at: https://ieeexplore.ieee.org/abstract/document/8378034 [Accessed 18 July 2021].

[50] Zhao, K. and Ge, L., 2013. A Survey on the Internet of Things Security. In: 2013 Ninth International Conference on Computational Intelligence and Security. [online] Emeishan, China: Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/document/6746513 [Accessed 27 June 2021].

[51] Zhang, Z., Cheng Yi Cho, M., Wang, C., Shieh, S., Hsu, C. and Chen, C., 2021. IoT Security: Ongoing Challenges and Research Opportunities. In: 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications. [online] Matsue, Japan: Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/document/6978614 [Accessed 27 June 2021].

[52] Community.broadcom.com. 2014. IoT Worm Used to Mine Cryptocurrency. [online] Available at: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=00fcdbad-954d-42ff-af50-4d74001bdcbb&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments [Accessed 27 June 2021].

[53] Wang, Q., Dunlap, T., Cho, Y. and Qu, G., 2017. DoS attacks and countermeasures on network devices. In: 26th Wireless and Optical Communication Conference. [online] Newark: Institute of Electrical and Electronics Engineers, pp.1-6. Available at: https://ieeexplore.ieee.org/document/7928974 [Accessed 27 June 2021].

[54] Ngo, Q., Nguyen, H., Le, V. and Nguyen, D., 2020. A survey of IoT malware and detection methods based on static features. [online] ScienceDirect. Available at: https://reader.elsevier.com/reader/sd/pii/S2405959520300503?token=56256B9FB82882CD91E6B8E574B9B0DFFF410EB01A42B7F953CF2780C768DE9FE0AE444E6B766258602AB807C8434856&originRegion=eu-west-1&originCreation=20210705094443 [Accessed 5 July 2021].

[55] Ferrag, M., Maglaras, L., Janicke, H., Jiang, J. and Shu, L., 2017. Authentication Protocols for Internet of Things: A Comprehensive Survey. Security and Communication Networks, [online] 2017, pp.1-41. Available at: https://doi.org/10.1155/2017/6562953 [Accessed 27 June 2021].

[56] Ferrag, M., Maglaras, L., Janicke, H., Jiang, J. and Shu, L., 2017. Authentication Protocols for Internet of Things: A Comprehensive Survey. Security and Communication Networks,

[online] 2017, pp.1-41. Available at: https://doi.org/10.1155/2017/6562953 [Accessed 27 June 2021].

[57] Saadeh, M., Sleit, A., Qatawneh, M. and Almobaideen, W., 2016. Authentication Techniques for the Internet of Things: A Survey. In: 2016 Cybersecurity and Cyberforensics Conference (CCC). [online] Amman, Jordan: Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/document/7600206 [Accessed 20 July 2021].

[58] Ferrag, M., Maglaras, L., Janicke, H., Jiang, J. and Shu, L., 2017. Authentication Protocols for Internet of Things: A Comprehensive Survey. [online] Hindawi. Available at: https://www.hindawi.com/journals/scn/2017/6562953/ [Accessed 18 July 2021].

[59] Docs.microsoft.com. 2017. Azure IoT protocol gateway. [online] Available at: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-protocol-gateway [Accessed 21 June 2021].

[60] Docs.microsoft.com. 2020. What is Azure Active Directory? - Azure Active Directory. [online] Available at: https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis [Accessed 21 June 2021].

[61] Docs.microsoft.com. 2021. Understand the Azure IoT Hub identity registry. [online] Available at: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-identity-registry [Accessed 21 June 2021].

[62] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and Polk, W., 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. [online] Datatracker.ietf.org. Available at: https://datatracker.ietf.org/doc/html/rfc5280 [Accessed 21 June 2021].

[63] Docs.aws.amazon.com. n.d. AWS IoT Device Shadow service - AWS IoT Core. [online] Available at: https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html [Accessed 21 June 2021].

[64] Docs.aws.amazon.com. n.d. Managing devices with AWS IoT - AWS IoT Core. [online] Available at: https://docs.aws.amazon.com/iot/latest/developerguide/iot-thing-management.html [Accessed 21 June 2021].

[65] Docs.aws.amazon.com. 2021. Amazon Cognito Federated Identities. [online] Available at: https://docs.aws.amazon.com/cognitoidentity/latest/APIReference/Welcome.html [Accessed 21 June 2021].

[66] Google Cloud. 2019. Cloud IoT Core overview. [online] Available at: https://cloud.google.com/iot/docs/concepts/overview [Accessed 24 June 2021].

[67] Pierleoni, P., Concetti, R., Belli, A. and Palma, L., 2019. Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison. [online] Ieeexplore.ieee.org. Available at:

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8938723 [Accessed 27 June 2021].

[68] Jones, M., Bradley, J. and Sakimura, N., 2015. JSON Web Token (JWT). [online] Internet Engineering Task Force (IETF). Available at: https://www.rfc-editor.org/rfc/rfc7519.txt [Accessed 27 June 2021].

[69] Google Cloud. n.d. Cloud Identity|Google Cloud. [online] Available at: https://cloud.google.com/identity [Accessed 27 June 2021].

[70] Cisco. n.d. Cisco Kinetic IoT Platform - Cisco IoT Solutions. [online] Available at: https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-kinetic.html [Accessed 5 July 2021].

[71] Cisco.com. n.d. DATASHEET EDGE & FOG PROCESSING MODULE. [online] Available at: https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-datasheet-efm.pdf [Accessed 5 July 2021].

[72] Cisco.com. n.d. DATASHEET DATA CONTROL MODULE. [online] Available at: https://www.cisco.com/c/dam/en/us/solutions/collateral/internet-of-things/kinetic-datasheet-dcm.pdf [Accessed 5 July 2021].

[73] Kaaproject.github.io. 2021. Contribution guide · Kaa. [online] Available at: https://kaaproject.github.io/kaa/docs/v0.10.0/How-to-contribute/ [Accessed 7 July 2021].

[74] Docs.kaaiot.io. 2021. Kaa IoT Platform. [online] Available at: https://docs.kaaiot.io/KAA/docs/current/Welcome/ [Accessed 7 July 2021].

[75] OpenRemote. n.d. Documentation to build your own open source IoT platform | OpenRemote. [online] Available at: https://openremote.io/developers/ [Accessed 12 July 2021].

[76] In: Intelligent Systems Conference. 2017. Drivers, Standards and Platforms for the IoT: Towards a Digital VICINITY. [online] London: Institute of Electrical and Electronics Engineers. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8324287 [Accessed 12 July 2021].

[77] Turner, R., 2020. Openremote: User Guide: Realms, users and roles. [online] GitHub. Available at: https://github.com/openremote/openremote/wiki/User-Guide%3A-Realms%2C-users-and-roles [Accessed 14 July 2021].

[78] Keycloak.org. 2021. Keycloak - Documentation. [online] Available at: https://www.keycloak.org/documentation.html [Accessed 14 July 2021].

[79] Docs.thinger.io. n.d. Thinger.io OVERVIEW. [online] Available at: https://docs.thinger.io/ [Accessed 14 July 2021].

[80] Luis Bustamante, A., Patricio, M. and Molina, J., 2019. Thinger.io: An Open Source Platform for Deploying Data Fusion Applications in IoT Environments. [online] National

Center for Biotechnology Information. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427624/ [Accessed 14 July 2021].

[81] ThingsBoard. n.d. ThingsBoard architecture. [online] Available at: https://thingsboard.io/docs/reference/ [Accessed 18 July 2021].

[82] Docker Documentation. n.d. Install Docker Compose. [online] Available at: https://docs.docker.com/compose/install/ [Accessed 24 July 2021].

[83] Docker Documentation. n.d. Install Docker Engine on Ubuntu. [online] Available at: https://docs.docker.com/engine/install/ubuntu/ [Accessed 24 July 2021].

[84] ThingsBoard. n.d. OAuth 2.0 Support. [online] Available at: https://thingsboard.io/docs/user-guide/oauth-2-support/#login-with-auth0] [Accessed 24 July 2021].