

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

“Ανάπτυξη πλατφόρμας ομογενοποίησης δικτύων αποτελούμενο από
ετερογενείς τεχνολογίες”

Σπουδαστής: Λιάκου Αθανασία

Εισηγητής: Αντωνόπουλος Χρήστος

Πάτρα 2023

ΠΕΡΙΛΗΨΗ

Το θέμα της διπλωματικής μου εργασίας αφορά την ανάπτυξη πλατφόρμας ομογενοποίησης δικτύων αποτελούμενο από ετερογενείς τεχνολογίες.

Τα τελευταία χρόνια με την πρόοδο της τεχνολογίας έχει αναπτυχθεί με ραγδαίους ρυθμούς ο τομέας του Internet of Things, Διαδίκτυο των Πραγμάτων, σε όλα τα επίπεδα τόσο στο ερευνητικό μέρους του όσο και στο παραγωγικό. Το Internet of Things διαφέρει από τα κλασικά δίκτυα όσο αφορά τις διαφορετικές τεχνολογίες και πρωτόκολλα που χρησιμοποιούνται από την επιμέρους συσκευές. Συνεπώς δημιουργείται η ανάγκη υλοποίησης μια πλατφόρμα που θα λαμβάνει τα δεδομένα από την κάθε συσκευή και ότι πρωτόκολλο και να χρησιμοποιεί η κάθε συσκευή, τα δεδομένα να ομογενοποιούνται και στην συνέχεια να επεξεργάζονται και να αποθηκεύονται σε μία κοινή βάση δεδομένων.

Λέξεις – Κλειδιά: Internet of Things, MQTT, Data Aggregator, Αισθητήρες, Μηχανική Εκμάθηση, Serialization Format, IoT Gateway, Cassandra, Protocol Buffers.

ABSTRACT

The topic of my research papers is the development of a platform for network homogenization composed of heterogeneous technologies. In recent years, with the progress of technology, the Internet of Things (IoT) has developed rapidly at all levels, both in research and production. IoT differs from traditional networks in terms of the different technologies and protocols used by individual devices. Therefore, there is a need to implement a platform that will receive data from each device, regardless of the protocol used, homogenize the data, and then process and store it in a common database.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	6
ABSTRACT	7
ΚΕΦΑΛΑΙΟ 1 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	
1.1 Εισαγωγή στα Ασύρματα Δίκτυα.....	8
1.2 Internet of Things.....	8
1.3 Λειτουργία ΙοΤ.....	10
1.4 Συστατικά του ΙοΤ.....	10
1.4.1 Συσκευές.....	11
1.4.2 Αισθητήρες.....	12
1.4.3 Δίκτυο Συνδεσιμότητας.....	13
1.4.4 Πλατφόρμα ΙοΤ.....	13
1.5 Πρωτόκολλα Επικοινωνίας.....	14
1.5.1 MQTT.....	14
1.5.2 XMPP.....	16
1.5.3 CoAP.....	16
1.5.4 AMQP.....	17
1.5.5 ZigBee.....	18
1.5.6 LoRaWAN.....	18
1.5.7 Bluetooth.....	19
1.5.8 Wifi.....	19
1.6 Αρχιτεκτονική του ΙοΤ.....	20
1.7 Ασφάλεια στο ΙοΤ.....	21
1.7.1 Θέματα Ασφάλειας στο ΙοΤ.....	21
1.7.2 Προστασία Συστημάτων ΙοΤ.....	22
1.8 Ομογενοποίηση Δεδομένων.....	23
1.8.1 Διαδικασία Ομογενοποίησης Δεδομένων.....	23
1.9 ΙοΤ και Τεχνητή Νοημοσύνη.....	24
1.9.1 Αλγόριθμοι Μηχανικής Εκμάθησης στο ΙοΤ.....	25
ΚΕΦΑΛΑΙΟ 2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑΣ	
2.1 Περιγραφή Αρχιτεκτονικής.....	27

2.2 Environmental Application.....	28
2.3 Data Aggregator.....	29
2.4 Βάση Δεδομένων Cassandra.....	30
2.5 Serialization Format.....	31
ΚΕΦΑΛΑΙΟ 3 ΥΛΟΠΟΙΗΣΗ ΠΛΑΤΦΟΡΜΑΣ	
3.1 Εργαλεία.....	32
3.1.1 Python.....	32
3.1.2 IoT Gateway.....	33
3.1.3 MQTT Broker της EMQX.....	33
3.1.4 Βάση Δεδομένων Cassandra.....	34
3.1.5 Protocol Buffers.....	35
3.1.6 Sensor Node Εφαρμογή.....	36
3.2 Υλοποίηση.....	37
3.2.1 Υλοποίηση Environmental Application.....	37
3.2.2 Υλοποίηση Data Aggregator.....	39
ΣΥΜΠΕΡΑΣΜΑΤΑ	
4.1 Συνολική Αξιολόγηση.....	40
4.2 Μελλοντική Επέκταση της Πλατφόρμας.....	41
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	43
ΠΑΡΑΡΤΗΜΑ.....	44

ΕΙΣΑΓΩΓΗ

Η τεχνολογία του Internet of Things (IoT) έχει δημιουργήσει ένα νέο πεδίο εφαρμογών στον τομέα της τεχνολογίας των ασύρματων δικτύων και των αισθητήρων. Η ανάπτυξη εφαρμογών IoT συνήθως απαιτεί τη συλλογή και επεξεργασία δεδομένων από διάφορες πηγές και συσκευές. Ωστόσο, οι πηγές αυτές χρησιμοποιούν διαφορετικές τεχνολογίες και πρωτόκολλα επικοινωνίας, κάτι που δυσχεραίνει την ομογενοποίηση των δεδομένων και την ανάπτυξη ενιαίας πλατφόρμας διαχείρισης αυτών των δεδομένων.

Στην παρούσα διπλωματική εργασία παρουσιάζουμε την ανάπτυξη μιας πλατφόρμας ομογενοποίησης δεδομένων για το Internet of Things. Η πλατφόρμα αυτή αποτελείται από ετερογενείς τεχνολογίες στόχος μας είναι τα δεδομένα να ομαδοποιούνται και να αποθηκεύονται με τρόπο που να μπορούν να επεξεργαστούν. Κατά την υλοποίηση της πλατφόρμας, θα αναπτυχθούν μηχανισμοί που θα επιτρέπουν την αξιοποίηση των δεδομένων από διάφορες πηγές και την ομογενοποίηση τους. Η αρχιτεκτονική του συστήματος θα βασίζεται στη χρήση διαφορετικών εργαλείων με ανοιχτά πρότυπα και τεχνολογίες, όπως ο MQTT Broker της EMQX, η βάση δεδομένων Cassandra και η γλώσσα προγραμματισμού Python.

Πιο συγκεκριμένα, στο πρώτο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της διπλωματικής. Αρχικά γίνεται μια εισαγωγή στα Ασύρματα Δίκτυα, τα οποία αποτελούν την βάση για το Internet of Things και παρουσιάζεται η έννοια του Internet of Things και η σημασία του στην σύγχρονη τεχνολογία. Στην συνέχεια, αναλύονται οι βασικές αρχές λειτουργίας του IoT και παρουσιάζονται τα συστατικά που απαρτίζουν το IoT, όπως οι συσκευές, οι αισθητήρες, το δίκτυο συνδεσιμότητας και η πλατφόρμα IoT. Ακόμα, αναλύονται τα διάφορα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται στο IoT, όπως το MQTT, το XMPP, το CoAP, το AMQP, το ZigBee, το LoRaWAN, το Bluetooth και το Wifi. Στη συνέχεια, εξετάζεται η αρχιτεκτονική του IoT και η ομογενοποίηση δεδομένων, που αποτελεί σημαντικό παράγοντα για την επίτευξη της αξιοπιστίας και αποδοτικότητας στα δίκτυα IoT και διευκολύνει την επικοινωνία μεταξύ διαφορετικών συσκευών και πλατφορμών. Τέλος, αναλύονται οι σχέσεις του IoT με την τεχνητή νοημοσύνη και εξετάζεται η χρήση αλγορίθμων μηχανικής εκμάθησης στο IoT.

Στο δεύτερο κεφάλαιο αναλύεται η αρχιτεκτονική της πλατφόρμας του IoT την οποία θέλουμε να υλοποιήσουμε. Αρχικά, παρουσιάζεται μια περιγραφή της αρχιτεκτονικής και πώς αυτή επιτρέπει την αποδοτική λειτουργία της πλατφόρμας. Στη συνέχεια, αναλύεται η λειτουργία του Environmental Application και πώς αυτή συνεργάζεται με τα υπόλοιπα συστατικά της πλατφόρμας. Επιπλέον, αναλύεται ο ρόλος του Data Aggregator, που είναι υπεύθυνος για τη συλλογή των δεδομένων από τους αισθητήρες και τη μετάδοσή τους στη βάση δεδομένων της πλατφόρμας. Αναφέρεται επίσης η χρήση της βάσης δεδομένων Cassandra, η οποία επιτρέπει την αποθήκευση μεγάλου όγκου δεδομένων με υψηλή απόδοση και διαθεσιμότητα. Τέλος, αναφέρεται η χρήση του Serialization Format για τη μετατροπή των δεδομένων σε μια μορφή που μπορεί να μεταδοθεί μέσω του δικτύου στο Data Aggregator και στη βάση δεδομένων. Το Serialization Format το οποίο έχει επιλεγεί είναι τα Protocol Buffers και αναλύονται τα πλεονεκτήματα της χρήσης του.

Το τρίτο κεφάλαιο αναφέρεται στην υλοποίηση της πλατφόρμας IoT που περιγράφεται στο προηγούμενο κεφάλαιο. Παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν, μεταξύ των οποίων η γλώσσα προγραμματισμού Python, το IoT Gateway, ο MQTT Broker της EMQX, η βάση δεδομένων Cassandra και τα Protocol Buffers. Επιπλέον, παρουσιάζεται η μέθοδος υλοποίησης του Environmental Application και του Data Aggregator.

Τέλος, το τέταρτο και τελευταίο κεφάλαιο αποτελεί μια συνολική αξιολόγηση της υλοποίησης της πλατφόρμας και αναφέρεται σε μελλοντικές επεκτάσεις της πλατφόρμας. Πιο συγκεκριμένα, στην πρώτη υποενότητα, "Συνολική Αξιολόγηση", γίνεται μια ανασκόπηση της αρχιτεκτονικής της πλατφόρμας και των εργαλείων που χρησιμοποιήθηκαν για την υλοποίησή της. Παρέχεται μια συνολική εκτίμηση για τη λειτουργικότητα και την απόδοση της πλατφόρμας και περιγράφονται οι πιθανές βελτιώσεις που θα μπορούσαν να γίνουν. Ενώ στην δεύτερη υποενότητα, "Μελλοντική Επέκταση της Πλατφόρμας", περιγράφονται οι πιθανές επεκτάσεις της πλατφόρμας στο μέλλον.

ΚΕΦΑΛΑΙΟ 1

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΡΑΘΡΟ

1.1 Εισαγωγή στα Ασύρματα Δίκτυα

Τα ασύρματα δίκτυα αποτελούν δίκτυα επικοινωνίας που επιτρέπουν τη μετάδοση δεδομένων με ασύρματο τρόπο, χωρίς τη χρήση φυσικών συνδέσεων καλωδίων. Μπορούν να χρησιμοποιούνται σε διάφορες εφαρμογές, όπως η ασύρματη επικοινωνία μεταξύ συσκευών IoT, η ασύρματη σύνδεση στο διαδίκτυο, η ασύρματη επικοινωνία σε οχήματα και η ασύρματη σύνδεση στα ασύρματα δίκτυα κινητής τηλεφωνίας. Μερικά από τα πιο δημοφιλή ασύρματα δίκτυα περιλαμβάνουν το Wi-Fi, το Bluetooth, το Zigbee και το LoRaWAN. Κάθε ασύρματο δίκτυο έχει τα δικά του χαρακτηριστικά και επιλογές που το καθιστούν κατάλληλο για διαφορετικές εφαρμογές.

Τα ασύρματα δίκτυα είναι ζωτικής σημασίας για το Internet of Things (IoT), καθώς επιτρέπουν τη σύνδεση και την ανταλλαγή δεδομένων μεταξύ διαφόρων συσκευών και αισθητήρων στο IoT. Οι συσκευές και οι αισθητήρες στο IoT συνήθως είναι απομακρυσμένοι και διασκορπισμένοι, και επομένως, δεν είναι εφικτό να συνδεθούν με καλώδια. Αντ' αυτού, χρησιμοποιούν ασύρματα δίκτυα για να επικοινωνήσουν μεταξύ τους και με το διαδίκτυο.

Τέλος, τα ασύρματα δίκτυα στο IoT μπορούν να υποστηρίξουν διάφορα πρωτόκολλα, όπως το Wi-Fi, το Bluetooth, το Zigbee και το LoRaWAN, και κάθε τεχνολογία έχει τα δικά της πλεονεκτήματα και μειονεκτήματα ανάλογα με την εφαρμογή. Ωστόσο, όλες οι τεχνολογίες ασύρματων δικτύων στο IoT έχουν κοινό στόχο, να διευκολύνουν τη συνδεσιμότητα των συσκευών και των αισθητήρων στο διαδίκτυο και να επιτρέψουν την ανταλλαγή δεδομένων για να επιτευχθούν διάφορες εφαρμογές IoT.

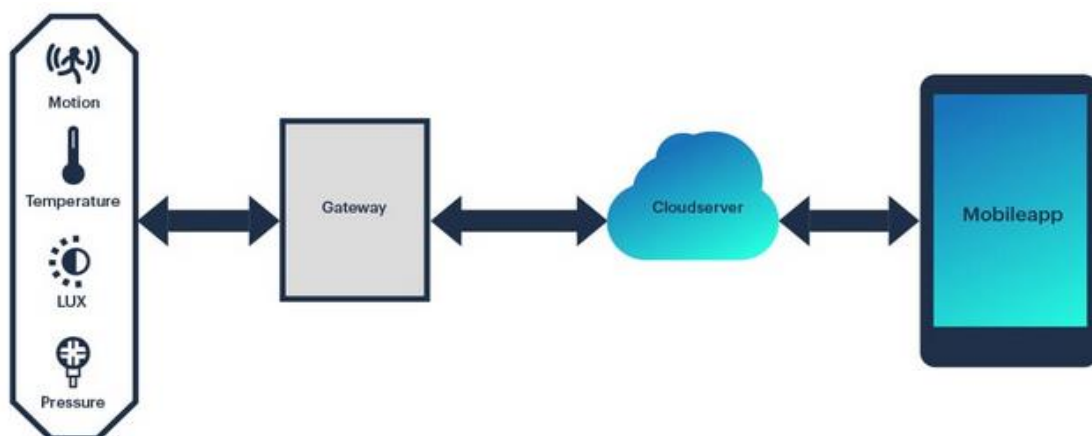
1.2 Internet of Things

Το Internet of Things (IoT) ή Διαδίκτυο των πραγμάτων αποτελεί ένα μεγάλο δίκτυο διασυνδεδεμένων συσκευών όπως φυσικές συσκευές, λογισμικό, αισθητήρες όπου επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων και πληροφορίας μεταξύ ενός τοπικού δικτύου και στο

διαδίκτυο. Δηλαδή, οι αισθητήρες που είναι συνδεδεμένοι στο διαδίκτυο συλλέγουν τα δεδομένα από μετράν οι αισθητήρες και είτε παρουσιάζουν τις πληροφορίες στο διαδίκτυο ή προσαρμόζεται κάποια μορφή τεχνητής νοημοσύνης ώστε αυτόματα να γίνεται κάποια ενέργεια η οποία έχει καθοριστεί. Επιτρέπει τη σύνδεση του φυσικού κόσμου με τον ψηφιακό κόσμο και τη δημιουργία νέων δυνατοτήτων σε διάφορους τομείς όπως η ενέργεια, η υγεία, οι μεταφορές και η παραγωγή. Οι συσκευές συλλέγουν και μεταφέρουν και επεξεργάζονται μεγάλες ποσότητες δεδομένων.

Η βασική ιδέα του IoT εμφανίστηκε περίπου το 1980 και το 1985 ο Peter T. Lewis έδωσε το ορισμό “IoT είναι η ενσωμάτωση ανθρώπων, διαδικασιών, συσκευών και της τεχνολογίας, σε ένα κοινό δίκτυο, για την απομακρυσμένη παρακολούθηση, χειρισμό και αξιολόγηση των τάσεων των συσκευών”.

Πρόκειται για μία ασύρματη τεχνολογία η οποία εφαρμόζεται σε πληθώρα τομέων της καθημερινότητας και αποτελείται από μία μεγάλη ποικιλία ετερογενών συσκευών όπως συστήματα ασφαλείας, κινητά τηλέφωνα, αυτοκίνητα, αισθητήρες, φώτα, κάμερες. Ο βασικός σκοπός είναι η διασύνδεση και επικοινωνία όλων των συστατικών μεταξύ τους δίνοντας την δυνατότητα στον χρήστη να τα ελέγχει και να αλληλοεπιδρά με το δίκτυο άμεσα και εύκολα μέσω πολλαπλών συσκευών όπως ένα κινητό ή υπολογιστή ή κινητό. Επιπλέον το IoT έχει σχεδιαστεί ώστε να κάνει την καθημερινότητά μας πιο εύκολη επιλύοντας προβλήματα που οι άνθρωποι δεν μπορούν να επιλύσουν και κατά συνέπεια βελτιώνεται η ζωή του ανθρώπου με την εισαγωγή καινοτόμων τρόπων.



Εικόνα 1.1: IoT Αρχιτεκτονική

1.3 Λειτουργία IoT

Η λειτουργία του IoT βασίζεται στο ότι κάθε συστατικό - συσκευή χρησιμοποιώντας έναν ενσωματωμένο αισθητήρα μπορεί να συνδέεται στο διαδίκτυο με σκοπό να λαμβάνει και να στέλνει δεδομένα μέσω διαφόρων πρωτοκόλλων, όπως για παράδειγμα το Wifi. Δηλαδή βασίζεται στη συνεχή συλλογή, αποθήκευση, ανάλυση και ανταλλαγή δεδομένων από τους αισθητήρες και τις συσκευές που συνδέονται σε αυτό. Οι συσκευές μπορούν να εντοπίσουν τις χρήσιμες πληροφορίες και να τις διαχειριστούν όπως έχει οριστεί.

Οι αισθητήρες εκπέμπουν συνεχώς δεδομένα σχετικά με το περιβάλλον και της λειτουργίας τους, όπως η θερμοκρασία, η υγρασία, η ταχύτητα, η τοποθεσία και άλλες παράμετροι που σχετίζονται με την κατάσταση και την λειτουργία τους. Αυτά τα δεδομένα στη συνέχεια μεταδίδονται στο διαδίκτυο, όπου αναλύονται και χρησιμοποιούνται για να ληφθούν αποφάσεις, να ελεγχθούν οι διαδικασίες και να βελτιωθεί η απόδοση του συστήματος.

Η πλατφόρμα του IoT αποτελείται από cloud server και βάσεις δεδομένων όπου ενσωματώνονται και επεξεργάζονται τα δεδομένα από τους αισθητήρες. Πιο συγκεκριμένα, αναλύει διεξοδικά τα δεδομένα και επιστρέφει πληροφορίες με βάση τα δεδομένα που παρέχονται. Επιπλέον, ένα σημαντικό στοιχείο της λειτουργίας του IoT είναι η αυτοματοποίηση της διαδικασίας λήψης αποφάσεων και ενεργειών με βάση τα δεδομένα που συλλέγονται. Για παράδειγμα, μπορεί να ρυθμιστεί μια συσκευή να ανοίγει αυτόματα το κλιματιστικό όταν ανιχνεύεται ότι η θερμοκρασία έχει αυξηθεί πάνω από ένα συγκεκριμένο όριο. Τέλος, τα επεξεργασμένα δεδομένα μοιράζονται στις υπόλοιπες συσκευές του συστήματος.

1.4 Συστατικά του IoT

Τα συστατικά του IoT είναι οι βασικές συνιστώσες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών IoT. Αν και οι συστατικές αυτές μπορεί να διαφέρουν ανάλογα με την εφαρμογή, μπορούν να περιλαμβάνουν τα παρακάτω

1.4.1 Συσκευές

Οι συσκευές Internet of Things (IoT) είναι φυσικά αντικείμενα που είναι συνδεδεμένα στο διαδίκτυο και είναι ικανά να συλλέγουν και να ανταλλάσσουν δεδομένα μεταξύ τους και με άλλες συσκευές ή συστήματα. Οι συσκευές IoT μπορούν να έχουν πολλές μορφές, από αισθητήρες και κάμερες ασφάλειας έως οικιακές συσκευές, οχήματα, έξυπνα ρολόγια και βιομηχανικά μηχανήματα.

Οι συγκεκριμένες συσκευές αποτελούνται στην πλειοψηφία τους από αισθητήρες οι οποίοι συλλέγουν πληροφορίες από το φυσικό περιβάλλον όπως διάφορες αλλαγές σε θερμοκρασία, κάποια ανίχνευση κίνησης, πίεση, υγρασία, φως και τις αποθηκεύουν. Η συνδεσιμότητα των συσκευών IoT με το διαδίκτυο και η δυνατότητα ανταλλαγής δεδομένων μεταξύ τους ανοίγει τον δρόμο για την ανάπτυξη πολλών εφαρμογών και υπηρεσιών σε πολλούς τομείς όπως η υγεία, η βιομηχανία, η ασφάλεια και η ενέργεια.

Στην συνέχεια θα αναφερθούν κάποιες βασικές IoT συσκευές

- **Z-Wave συσκευές:** Οι συσκευές Z-Wave είναι συσκευές του Internet of Things που λειτουργούν με ασύρματο τρόπο και χρησιμοποιούνται για τον έλεγχο και τη διαχείριση ενός χώρου. Η τεχνολογία Z-Wave χρησιμοποιεί ένα ασύρματο πρωτόκολλο επικοινωνίας χαμηλής ισχύος και μικρού εύρους ζώνης, που επιτρέπει στις συσκευές να επικοινωνούν μεταξύ τους και με έναν κεντρικό ελεγκτή, όπως έναν έξυπνο ηλεκτρονικό υπολογιστή ή ένα smartphone.
- Οι συσκευές Z-Wave περιλαμβάνουν αισθητήρες κίνησης, θερμοστάτες, κλειδαριές πόρτας, φώτα, ηχεία και άλλες συσκευές που μπορούν να ελέγχονται ασύρματα. Η τεχνολογία Z-Wave είναι διαθέσιμη σε μια πληθώρα συσκευών από διάφορους κατασκευαστές, και μπορεί να χρησιμοποιηθεί σε διάφορα συστήματα.
- **BLE συσκευές:** Οι συσκευές BLE (Bluetooth Low Energy) είναι συσκευές που χρησιμοποιούν το πρωτόκολλο Bluetooth για να ανταλλάξουν δεδομένα με άλλες συσκευές σε μικρή απόσταση και με ελάχιστη κατανάλωση ενέργειας. Αυτό το καθιστά ιδανικό για εφαρμογές στο Internet of Things, όπου οι συσκευές πρέπει να λειτουργούν για μεγάλο χρονικό διάστημα με μπαταρίες. Οι συσκευές BLE είναι

συνήθως αισθητήρες, περιφερειακά υγείας, wearables και άλλες συσκευές που μετρούν και αποστέλλουν δεδομένα για την κατάστασή τους, όπως η θερμοκρασία, η κίνηση, οι καρδιακοί παλμοί, η πίεση του αίματος κ.λπ. Τέλος, χρησιμοποιούνται συχνά στον κλάδο της έξυπνης υγείας, των φορητών συσκευών και των έξυπνων σπιτιών.

- **ROS based συσκευές:** Οι ROS based συσκευές είναι συσκευές που χρησιμοποιούν το λογισμικό ανοικτού κώδικα Robot Operating System (ROS) για τον έλεγχό τους. Το ROS είναι ένα σύστημα λογισμικού που παρέχει βιβλιοθήκες, εργαλεία και βασικές λειτουργίες για τη δημιουργία λογισμικού ρομποτικής εφαρμογής. Οι συσκευές που χρησιμοποιούν το ROS μπορούν να επικοινωνούν μεταξύ τους και με άλλες συσκευές μέσω διαφόρων πρωτοκόλλων επικοινωνίας, όπως το MQTT και το CoAP, για να ανταλλάσσουν δεδομένα και να συνεργάζονται σε εφαρμογές ρομποτικής και IoT.
- **KNX συσκευές:** Οι KNX συσκευές είναι συσκευές που χρησιμοποιούν το πρότυπο επικοινωνίας KNX για να επικοινωνήσουν μεταξύ τους και να αλληλοεπιδράσουν με ένα σύστημα κτιριακού αυτοματισμού. Μπορούν να είναι διάφορων τύπων συσκευές όπως διακόπτες, αισθητήρες φωτός, θερμοστάτες, αισθητήρες κίνησης, και πολλές άλλες συσκευές που ελέγχουν τις λειτουργίες του κτιρίου όπως φωτισμό, κλιματισμός, ασφάλεια, παράθυρα και πόρτες. Η επικοινωνία των KNX συσκευών γίνεται μέσω της τεχνολογίας του Bus που χρησιμοποιεί το KNX πρότυπο επικοινωνίας. Το Bus είναι ένα σύστημα διασύνδεσης πολλαπλών συσκευών μέσω ενός καλωδίου, και μπορεί να στείλει σήματα ελέγχου και πληροφορίες σε όλες τις συσκευές που συνδέονται σε αυτό.

1.4.2 Αισθητήρες

Όπως αναφέραμε ήδη, οι IoT συσκευές έχουν ενσωματωμένους αισθητήρες με σκοπό την λήψη δεδομένων από το περιβάλλον. Ως αισθητήρας ορίζεται η μονάδα που μετράει μία φυσική ποσότητα από το περιβάλλον στο οποίο βρίσκεται και την μετατρέπει σε ένα σήμα το οποίο έχει την δυνατότητα να επεξεργαστεί από τα υπολογιστικά συστήματα του δικτύου.

Οι αισθητήρες στις μέρες μας έχουν μεγάλο φάσμα εφαρμογών με ευρεία χρήση. Τα δεδομένα που μπορεί να συλλέξει ένας αισθητήρας είναι ποικίλα όπως τις μεταβλητές του περιβάλλοντος για παράδειγμα υγρασία και θερμοκρασίας αλλά και άλλες παραμέτρους ενός συστήματος όπως μία κίνηση. Τέλος το μέγεθος ενός αισθητήρα μπορεί να ποικίλει από κάποια εκατοστά έως ελάχιστα χιλιοστά

1.4.3 Δίκτυο Συνδεσιμότητας

Το δίκτυο συνδεσιμότητας στο IoT αποτελεί το σύνολο των διαφόρων τεχνολογιών και πρωτοκόλλων που χρησιμοποιούνται για τη μεταφορά δεδομένων μεταξύ των συσκευών IoT και των διακομιστών. Τα δίκτυα συνδεσιμότητας εξασφαλίζουν τη δυνατότητα σύνδεσης των συσκευών IoT στο Διαδίκτυο, καθώς και την ασφαλή και αξιόπιστη μεταφορά των δεδομένων που συλλέγονται από αυτές τις συσκευές.

Στα δίκτυα συνδεσιμότητας περιλαμβάνονται οι ασύρματες τεχνολογίες όπως το MQTT, το Bluetooth, το ZigBee και το LoRaWAN, καθώς και οι ενσύρματες τεχνολογίες όπως το Ethernet. Κάθε τεχνολογία έχει τα δικά της πλεονεκτήματα και περιορισμούς, επομένως η επιλογή της κατάλληλης τεχνολογίας για ένα συγκεκριμένο σενάριο χρήσης είναι σημαντική. Επιπλέον, το δίκτυο συνδεσιμότητας συνδέει τις συσκευές IoT με τους διακομιστές cloud και τα συστήματα ανάλυσης δεδομένων, διευκολύνοντας έτσι την επεξεργασία και την ανάλυση των δεδομένων που συλλέγονται.

1.4.4 Πλατφόρμα IoT

Η πλατφόρμα IoT είναι ένα λογισμικό που επιτρέπει στους χρήστες να συλλέγουν, να αποθηκεύουν και να αναλύουν δεδομένα από συσκευές IoT. Σκοπός της πλατφόρμας είναι να διευκολύνει τη διαχείριση και την ανάπτυξη εφαρμογών IoT. Η πλατφόρμα μπορεί να περιλαμβάνει διάφορα εργαλεία και υπηρεσίες, όπως διαχείριση συσκευών, συλλογή δεδομένων, ανάλυση δεδομένων, ασφάλεια και επικοινωνία.

Παρέχει επίσης ένα περιβάλλον ανάπτυξης εφαρμογών για να διευκολύνει τη δημιουργία προσαρμοσμένων λύσεων για συγκεκριμένες ανάγκες. Η πλατφόρμα IoT είναι κρίσιμης

σημασίας για την αποτελεσματική και αξιόπιστη λειτουργία των συσκευών IoT και των συνδεδεμένων εφαρμογών.

1.5 Πρωτόκολλα Επικοινωνίας IoT

Οι IoT συσκευές για να επικοινωνήσουν μεταξύ τους απαιτούν την χρήση πρωτοκόλλων. Τα πρωτόκολλα επικοινωνίας (communication protocols) είναι δηλαδή οι κανόνες που διέπουν την ανταλλαγή δεδομένων μεταξύ των συσκευών. Τα κλασικά πρωτόκολλα, όπως το TCP/IP, δεν μπορούν να υποστηρίξουν τις συγκεκριμένες συσκευές λόγω περιορισμένων ενεργειακών πόρων, χαμηλής υπολογιστικής ισχύς και ενδεχόμενων διακοπών στην σύνδεση. Για αυτό το σκοπό έχουν αναπτυχθεί πρωτόκολλα τα οποία ανταποκρίνονται στις ανάγκες τα οποία παρουσιάζονται στην συνέχεια.

1.5.1 MQTT

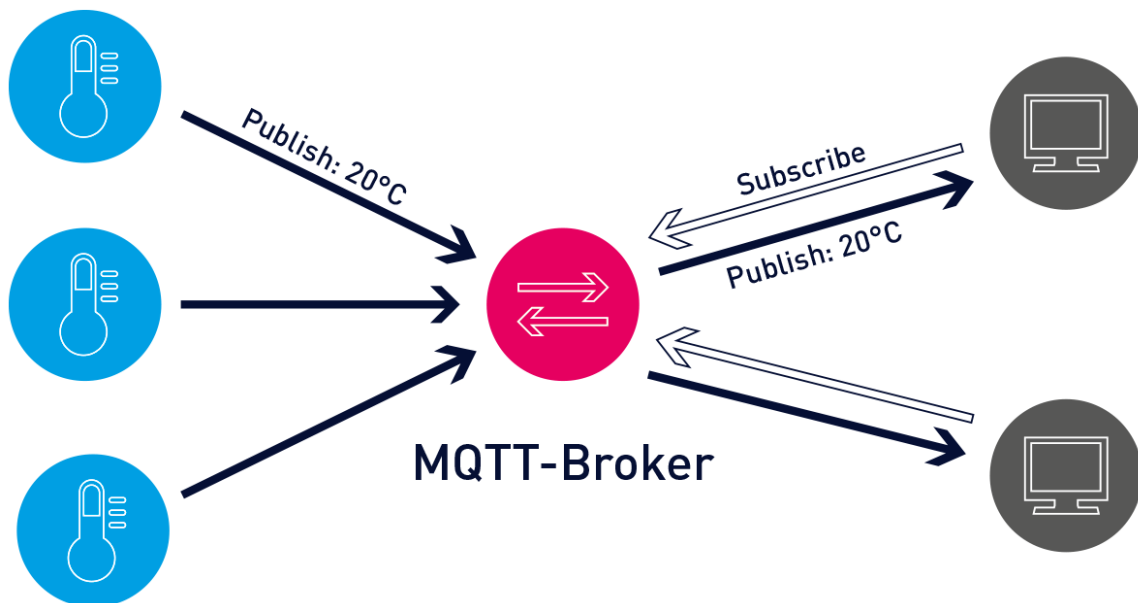
Το MQTT (Message Queuing Telemetry Transport) είναι ένα "ελαφρύ" πρωτόκολλο επικοινωνίας ανοικτού κώδικα για τη μεταφορά δεδομένων μεταξύ συσκευών IoT. Το MQTT πρωτόκολλο διαφέρει από την προσέγγιση του client/server και χρησιμοποιείται το πρότυπο publish/subscribe και επιτρέπει στις συσκευές να επικοινωνούν μεταξύ τους μέσω μιας μεσολαβητικής υπηρεσίας που ονομάζεται broker. Αναπτύχθηκε το 1999 από τους Andy Stanford-Clark και Arlen Nipper και στόχος ήταν να δημιουργηθεί ένα πρωτόκολλο επικοινωνίας χαμηλών απαιτήσεων σε ενέργεια και bandwidth όπου οι χρήστες έχουν την δυνατότητα να στέλνουν και να λαμβάνουν μηνύματα ως clients.

Οι συσκευές που επικοινωνούν μέσω MQTT μπορούν να είναι αισθητήρες, ενσωματωμένα συστήματα, υπολογιστές, κινητά τηλέφωνα και άλλες συσκευές IoT. Το MQTT είναι ελαφρύ, αποδοτικό και προσαρμόσιμο σε διάφορες συσκευές και πλατφόρμες. Κάποια από τα βασικά χαρακτηριστικά του είναι ότι αποτελεί ένα δυαδικό πρωτόκολλο μειώνοντας με αυτό τον τρόπο την επιβάρυνση που έχει το σύρμα. Επιπλέον, έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται το εύρος ζώνης και χρήση της μπαταρίας και έχει χαμηλές απαιτήσεις σε υπολογιστικούς πόρους. Το MQTT έχει σχεδιαστεί ώστε να είναι κατάλληλο για απομακρυσμένες συνδέσεις και είναι πολύ απλό να υλοποιηθεί. Βασικό ακόμα χαρακτηριστικό είναι πως το συγκεκριμένο

πρωτόκολλο είναι ασύγχρονο και συνεπώς δεν απαιτείται η ταυτόχρονη σύνδεση του subscriber με τον publisher.

Το MQTT χρησιμοποιεί την αρχή του publish-subscribe, όπου οι συσκευές δημοσιεύουν (publish) μηνύματα σε θέματα (topics) και άλλες συσκευές εγγράφονται (subscribe) σε αυτά τα θέματα για να λαμβάνουν τα μηνύματα που ενδιαφέρουν. Ο broker λειτουργεί ως μεσολαβητής μεταξύ των συσκευών, και διαχειρίζεται τα μηνύματα που διακινούνται μεταξύ τους. Μπορούν να υπάρξουν πολλοί "πελάτες" που είναι συνδεδεμένοι στο ίδιο broker. Οι πελάτες αυτοί μπορεί να είναι ή publisher, ή subscriber ή και τα δύο μαζί την ίδια στιγμή.

Πιο αναλυτικά, ένας publisher στέλνει το μήνυμα στον broker με ένα θέμα και οι subscribers εγγράφονται στον broker με ένα θέμα. Όταν ένα μήνυμα σταλεί στον broker, ο broker θα προωθήσει το μήνυμα σε όσους subscribers είχαν δηλώσει το ίδιο μήνυμα. Επομένως, οι πελάτες (publishers) δεν έχουν την δυνατότητα να γνωρίζουν την ύπαρξη των υπολοίπων αλλά ο broker έχει γνώση όσων είναι συνδεδεμένοι σε αυτόν. Τέλος, πρέπει να αναφέρουμε ότι επικοινωνία μέσω MQTT είναι αξιόπιστη και ασφαλής και χρησιμοποιείται ευρέως.



Εικόνα 1.2: Αρχιτεκτονική MQTT

1.5.2 XMPP

Το XMPP (Extensible Messaging and Presence Protocol) χαρακτηρίζεται ως πρωτόκολλο “ανοιχτής τεχνολογίας” και χρησιμοποιείται ευρέως ως σύστημα επικοινωνίας πραγματικού χρόνου και παρέχει ανταλλαγή μηνυμάτων και κλήσεις ήχου και εικόνας. Είναι βασισμένο σε ανοιχτά πρότυπα και προτείνεται ως ένα από τα βασικά πρωτόκολλα επικοινωνίας για το Internet of Things.

Η βασική ιδέα του XMPP είναι η χρήση ενός ανοιχτού, ασφαλούς και επεκτάσιμου πρωτοκόλλου που θα επιτρέπει την επικοινωνία σε πραγματικό χρόνο μεταξύ ετερογενών συσκευών. Το XMPP παρέχει μια αξιόπιστη και ασφαλή επικοινωνία σε πραγματικό χρόνο και υποστηρίζει πολλαπλά προφίλ χρήστη, διαφορετικά επίπεδα πρόσβασης και έλεγχο απορρίπτουν. Επίσης, έχει τη δυνατότητα να χρησιμοποιηθεί για τη μεταφορά δεδομένων σε μεγάλη κλίμακα, καθώς υποστηρίζει την αναζήτηση, την εγκαθίδρυση και τη διαχείριση συνόδους.

Θεωρείται ως ένα απλό πρωτόκολλο με υποδοχές TCP και μεταδίδονται μηνύματα XML. Βασίζεται στην αρχιτεκτονική του μοντέλου client – server και ένα από τα βασικά χαρακτηριστικά του είναι η δυνατότητα επέκτασης του με νέες λειτουργίες και πρωτόκολλα, επιτρέποντας στους προγραμματιστές να δημιουργήσουν προσαρμοσμένες εφαρμογές και λειτουργίες σύμφωνα με τις ανάγκες τους.

1.5.3 CoAP

Το CoAP (Constrained Application Protocol) αποτελεί ένα πρωτόκολλο επικοινωνίας που έχει σχεδιαστεί για τις περιορισμένες συσκευές του Internet των Πραγμάτων (IoT) και χρησιμοποιείται για την ανταλλαγή δεδομένων μεταξύ συσκευών. Το πρωτόκολλο CoAP έχει επηρεαστεί από το HTTP και υποστηρίζει απλές αιτήσεις και απαντήσεις μεταξύ των διαφορετικών συσκευών και των διακομιστών που υπάρχουν στο δίκτυο. Η διαφορά του CoAP από το HTTP είναι ότι το CoAP χρησιμοποιεί UDP (User Datagram Protocol) αντί για TCP (Transmission Control Protocol), το οποίο καθιστά το πρωτόκολλο πιο αποδοτικό και λειτουργικό για τις συσκευές με περιορισμένους πόρους.

Το CoAP παρέχει επίσης τη δυνατότητα επικοινωνίας σε περιβάλλοντα που έχουν περιορισμένη πρόσβαση στο διαδίκτυο, όπως τα δίκτυα ασύρματων αισθητήρων και άλλα δίκτυα που χρησιμοποιούν περιορισμένους πόρους. Το CoAP επίσης υποστηρίζει την ασφαλή μετάδοση δεδομένων με τη χρήση του DTLS (Datagram Transport Layer Security), το οποίο παρέχει κρυπτογραφημένη επικοινωνία μεταξύ των συσκευών και των διακομιστών.

1.5.4 AMQP

Το AMQP (Advanced Message Queuing Protocol) είναι ένα πρωτόκολλο επικοινωνίας που χρησιμοποιείται για την αποστολή και λήψη μηνυμάτων μεταξύ συσκευών και εφαρμογών σε ένα δίκτυο. Τα δεδομένα που μεταφέρει είναι σε δυαδική αναπαράσταση και παρέχει ασφάλεια και αξιοπιστία. Αναπτύχθηκε από την ομάδα του πρωτόκολλου AMQP στην οποία συμμετέχουν εταιρείες όπως η Red Hat, η Cisco και η Microsoft.

Το AMQP παρέχει μια αξιόπιστη, ασύγχρονη, πολυκαναλική και ασφαλή επικοινωνία μεταξύ συστημάτων. Το πρωτόκολλο χρησιμοποιεί ένα σύστημα ουρών για την αποθήκευση και διαχείριση των μηνυμάτων. Η εφαρμογή που στέλνει ένα μήνυμα το αποστέλλει στον broker του συστήματος και ο broker το αποθηκεύει στην ουρά. Η εφαρμογή που λαμβάνει το μήνυμα επικοινωνεί με τον broker και λαμβάνει το μήνυμα από την ουρά.

Το AMQP υποστηρίζει πολλά χαρακτηριστικά, όπως η δρομολόγηση μηνυμάτων με βάση πολλά κριτήρια, ο αναστολέας (pause) και ο επαναφορτιζόμενος (repayable) καταναλωτής, η δυνατότητα ανταλλαγής μηνυμάτων μεταξύ διαφορετικών πρωτοκόλλων και πολλά άλλα.

Η χρήση του AMQP στο IoT μπορεί να βελτιώσει την αξιοπιστία, την απόδοση και την αποτελεσματικότητα της επικοινωνίας μεταξύ των συσκευών. Επιπλέον, το AMQP παρέχει επίσης μια ευέλικτη και ασφαλή πλατφόρμα για τη διασύνδεση συσκευών στο IoT, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ διαφορετικών συσκευών και πλατφορμών.

1.5.5 ZigBee

Το Zigbee αποτελεί πρωτόκολλο ασύρματων δικτύων που χρησιμοποιείται στο IoT για τη σύνδεση και την επικοινωνία μεταξύ των συσκευών. Αναπτύχθηκε από το Zigbee Alliance, μια εταιρεία που αποτελείται από πολλούς κατασκευαστές και προμηθευτές, και έχει σχεδιαστεί ειδικά για τις απαιτήσεις του IoT, δηλαδή παρέχει χαμηλή κατανάλωση ενέργειας, μικρή εμβέλεια και ασφάλεια και αξιοπιστία.

Το Zigbee χρησιμοποιεί μια μορφολογία δικτύου mesh, όπου οι συσκευές μπορούν να επικοινωνούν απευθείας μεταξύ τους μέσω μεταξύ τους συνδεδεμένων συσκευών. Αυτό το σχήμα δικτύου επιτρέπει την αυξημένη κάλυψη και αξιοπιστία του δικτύου, καθώς και την αποκέντρωση της επικοινωνίας. Το Zigbee μπορεί να χρησιμοποιείται ευρέως σε πολλές εφαρμογές IoT, όπως έξυπνα σπίτια, έξυπνες πόλεις, κτίρια.

1.5.6 LoRaWAN

Το LoRaWAN (Long Range Wide Area Network) είναι ένα ασύρματο πρωτόκολλο επικοινωνίας το οποίο είναι σχεδιασμένο για τη σύνδεση IoT συσκευών με μεγάλη εμβέλεια, χαμηλή κατανάλωση ισχύος και χαμηλό κόστος. Το LoRaWAN χρησιμοποιείται συνήθως για συνδεδεμένες συσκευές που δεν χρειάζονται να στέλνουν συχνά δεδομένα, αλλά τα δεδομένα πρέπει να στέλνονται σε μεγάλη απόσταση.

Η τεχνολογία LoRaWAN βασίζεται σε μια συγκεκριμένη αρχιτεκτονική δικτύου, που αποτελείται από τρεις βασικούς τύπους σταθμών: τους LoRaWAN end-nodes, τις LoRaWAN gateways και το LoRaWAN network server. Οι end-nodes είναι οι συσκευές που είναι συνδεδεμένες στο δίκτυο και στέλνουν τα δεδομένα. Οι gateways είναι οι σταθμοί που λαμβάνουν τα δεδομένα από τα end-nodes και τα προωθούν στο δίκτυο, ενώ το LoRaWAN network server είναι υπεύθυνο για τη διαχείριση του δικτύου.

Τα δίκτυα LoRaWAN χρησιμοποιούνται σε διάφορους τομείς του IoT, όπως για παράδειγμα στη βιομηχανία, την ενέργεια, την αγροτική παραγωγή, την υγεία και την πόλη έξυπνων πόλεων. Η τεχνολογία αυτή επιτρέπει την ανάπτυξη δικτύων που καλύπτουν μεγάλες περιοχές με

ελάχιστο κόστος και δυνατότητα σύνδεσης εκατομμυρίων συσκευών IoT. Με το LoRaWAN, οι συσκευές IoT μπορούν να επικοινωνούν μεταξύ τους και με το cloud, στέλνοντας και λαμβάνοντας δεδομένα σε πραγματικό χρόνο.

1.5.7 Bluetooth

Το Bluetooth αποτελεί μια ασύρματη τεχνολογία επικοινωνίας που χρησιμοποιείται ευρέως στο IoT για την ασύρματη διασύνδεση διαφορετικών συσκευών. Μπορεί να χρησιμοποιηθεί για μικρές αποστάσεις επικοινωνίας, μέχρι 10 μέτρα περίπου, και χρησιμοποιείται συνήθως για τη σύνδεση μικρών συσκευών όπως αισθητήρες, ηλεκτρονικές συσκευές και κινητά τηλέφωνα. Η τεχνολογία Bluetooth στο IoT χρησιμοποιείται συνήθως για να συνδέει αισθητήρες και άλλες συσκευές με κινητά τηλέφωνα και άλλες φορητές συσκευές. Με τη βοήθεια της τεχνολογίας Bluetooth, οι χρήστες μπορούν να λαμβάνουν δεδομένα από τους αισθητήρες του IoT και να ελέγχουν και να διαχειρίζονται τις συσκευές τους από απόσταση.

Επιπλέον, η τεχνολογία Bluetooth χρησιμοποιείται επίσης για τη δημιουργία μικρών δικτύων στο IoT, γνωστών ως δίκτυα Bluetooth mesh, που επιτρέπουν τη σύνδεση πολλών συσκευών μεταξύ τους και τη μεταφορά δεδομένων από το ένα σημείο στο άλλο.

Η τελευταία και πιο διαδεδομένη έκδοση είναι η Bluetooth Low Energy (BLE), η οποία είναι σχεδιασμένη για να καλύπτει τις ανάγκες των IoT συσκευών, οι οποίες συνήθως δεν χρειάζονται μεγάλο εύρος ζώνης ή μεγάλο εύρος συχνοτήτων. Επιπλέον, η τεχνολογία BLE καταναλώνει λιγότερη ενέργεια από τις παλαιότερες εκδόσεις του Bluetooth, κάτι που την καθιστά ιδανική για συσκευές που λειτουργούν με μπαταρίες και έχουν μικρό μέγεθος. Οι συσκευές που χρησιμοποιούν το BLE μπορούν να επικοινωνούν μεταξύ τους μέσω ασύρματης σύνδεσης και μπορούν επίσης να συνδεθούν με ένα smartphone ή μια άλλη συσκευή IoT που έχει δυνατότητα Bluetooth σύνδεσης.

1.5.8 Wifi

Το Wi-Fi είναι μια ασύρματη τεχνολογία δικτύωσης που χρησιμοποιεί συχνότητες ραδιοκυμάτων για τη μετάδοση δεδομένων σε κοντινές αποστάσεις μεταξύ συσκευών. Η

τεχνολογία Wi-Fi χρησιμοποιείται ευρέως στα σπίτια, τα γραφεία, τα δημόσια χώρα και σε καταστήματα για τη σύνδεση συσκευών στο διαδίκτυο. Οι συσκευές που υποστηρίζουν Wi-Fi μπορούν να συνδεθούν σε δίκτυα Wi-Fi, όπου μπορούν να ανταλλάσσουν δεδομένα με άλλες συσκευές στο δίκτυο.

Η ταχύτητα μετάδοσης δεδομένων του Wi-Fi εξαρτάται από την έκδοση του πρωτοκόλλου Wi-Fi που χρησιμοποιείται. Οι πιο συνηθισμένες εκδόσεις πρωτοκόλλου είναι η 802.11n, η 802.11ac και η 802.11ax, με την τελευταία να παρέχει ταχύτητες έως και 10Gbps στα πιο σύγχρονα δίκτυα Wi-Fi 6. Επίσης, η συχνότητα λειτουργίας του Wi-Fi μπορεί να επηρεάσει την ταχύτητα και την απόδοση του δικτύου. Οι συχνότητες οι οποίες λειτουργεί είναι τα 2.4GHz και 5GHz.

Στο IoT, το WiFi χρησιμοποιείται συχνά ως μέσο επικοινωνίας μεταξύ των συσκευών. Συνήθως, οι συσκευές IoT που χρησιμοποιούν WiFi συνδέονται με έναν δρομολογητή (router) που είναι συνδεδεμένος στο διαδίκτυο ή σε ένα ιδιωτικό δίκτυο. Οι συσκευές IoT μπορούν να επικοινωνήσουν μεταξύ τους μέσω ενός ασύρματου δικτύου WiFi, αν και αυτό μπορεί να είναι λιγότερο αξιόπιστο από άλλες επιλογές σύνδεσης, όπως οι τεχνολογίες LPWAN. Ωστόσο, η χρήση του WiFi στο IoT είναι ευρέως διαδεδομένη και προσφέρει στις συσκευές IoT πρόσβαση στο διαδίκτυο και στα cloud services.

1.6 Αρχιτεκτονική του IoT

Η αρχιτεκτονική ενός IoT συστήματος βασίζεται σε τρεις διαστάσεις οι οποίες περιγράφονται αναλυτικά στην συνέχεια:

- Συστατικά πληροφοριών, όπου περιλαμβάνονται όλα τα συστατικά τα οποία συνδέονται με το IoT περιβάλλον και μπορούν να ανιχνεύουν τα αντικείμενα, τα στοιχεία ελέγχου και τις οντότητες.
- Ανεξάρτητο δίκτυο, όπου πρόκειται για το δίκτυο άμεσης σύνδεσης το οποίο χρησιμοποιείται και περιλαμβάνει τους μηχανισμούς αυτοπροστασίας, αυτοβελτίωσης, αυτοδιαμόρφωσης και αυτοπροσαρμογής.

- Έξυπνες εφαρμογές, όπου λαμβάνουν και επεξεργάζονται τα δεδομένα που παράγονται από IoT συσκευές.

1.7 Ασφάλεια στο IoT

Η ασφάλεια στο IoT αποτελεί μια πολύ σημαντικό ζήτημα και μεγάλη πρόκληση, καθώς οι συσκευές οι οποίες περιλαμβάνονται στα συγκεκριμένα συστήματα συνήθως είναι περιορισμένες σε υπολογιστικούς πόρους και επικοινωνία μεταξύ τους γίνεται μέσω ασύρματων δικτύων, όπως για παράδειγμα το Wi-Fi, το Bluetooth και το Zigbee. Αυτό έχει ως αποτέλεσμα οι συσκευές του IoT να είναι ευάλωτες σε διάφορες μορφές επιθέσεων και κακόβουλων ενεργειών.

1.7.1 Θέματα Ασφάλειας στο IoT

Τα IoT συστήματα αντιμετωπίζουν πολλούς κινδύνους ασφάλειας και ορισμένοι από τους κινδύνους αυτούς είναι οι εξής:

- Κακόβουλες επιθέσεις: Οι εισβολείς μπορούν να χρησιμοποιήσουν τα IoT συστήματα για να εκτελέσουν επιθέσεις στο δίκτυο ή σε άλλες συσκευές του δικτύου.
- Διαρροή δεδομένων: Οι κακόβουλοι χρήστες μπορούν να κλέψουν προσωπικά δεδομένα που συλλέγονται από τα IoT συστήματα, όπως πληροφορίες που σχετίζονται με την υγεία, τον τρόπο ζωής και την κατανάλωση ενέργειας.
- DoS επιθέσεις: Οι επιθέσεις DoS μπορούν να προκαλέσουν αδυναμία λειτουργίας του δικτύου ή των συσκευών του IoT.
- Αδυναμία ενημέρωσης των συσκευών: Οι συσκευές του IoT πρέπει να ενημερώνονται συχνά για τη βελτίωση της ασφάλειας του δικτύου.
- Εκμετάλλευση ευπάθειας των συσκευών: Οι κακόβουλοι χρήστες μπορούν να εκμεταλλευτούν τις ευπάθειες των συσκευών IoT, όπως οι κωδικοί πρόσβασης που δεν έχουν αλλάξει οι χρήστες ή προβλήματα ασφάλειας λογισμικού, για να έχουν πρόσβαση σε δεδομένα.

1.7.2 Προστασία Συστημάτων IoT

Τα θέματα ασφάλειας στο IoT απαιτούν προληπτικές ενέργειες και συνήθως χρησιμοποιούνται συνδυαστικά. Ακολουθούν μερικές από τις κύριες προληπτικές και δράσης προστασίας και αντιμετώπισης που μπορούν να ενισχύσουν την ασφάλεια των συστημάτων IoT.

- i. Αποτροπή επιθέσεων: Η καλύτερη πρακτική είναι να προλαμβάνετε επιθέσεις στο IoT σύστημά σας. Αυτό μπορεί να περιλαμβάνει τη χρήση ισχυρών κωδικών πρόσβασης, τη διαχείριση των ενημερώσεων λογισμικού, τη χρήση λογισμικού ανίχνευσης επιθέσεων και την ενημέρωση και εκπαίδευση του προσωπικού σας για τις βασικές πρακτικές ασφάλειας.
- ii. Περιορισμός πρόσβασης: Περιορίστε την πρόσβαση στο IoT σύστημά σας μόνο σε όσους έχουν ανάγκη να τη χρησιμοποιήσουν. Αυτό μπορεί να γίνει μέσω διαφόρων τεχνολογιών αυθεντικοποίησης και ελέγχου πρόσβασης, όπως πιστοποιητικά, κλειδιά κρυπτογράφησης και διαχείρισης ταυτότητας.
- iii. Ενημέρωση και διαχείριση ασφαλείας: Οι κατασκευαστές των συσκευών IoT πρέπει να παρέχουν ενημερώσεις και διαχείριση ασφαλείας για τις συσκευές τους. Αυτό περιλαμβάνει τη διόρθωση των αδυναμιών ασφαλείας και την αναβάθμιση του λογισμικού ασφαλείας.
- iv. Απομόνωση συσκευών: Οι συσκευές IoT μπορούν να απομονωθούν σε ένα ξεχωριστό δίκτυο, ώστε να περιοριστεί η επίδραση μιας ενδεχόμενης παραβίασης.
- v. Συνεχής εκπαίδευση και ενημέρωση των χρηστών: Η προώθηση βέλτιστων πρακτικών και η ευαισθητοποίηση των χρηστών και των επαγγελματιών για τους κινδύνους ασφαλείας μπορούν να μειώσουν αισθητά την πιθανότητα επιθέσεων στα IoT συστήματα.
- vi. Τέλος, ο συνδυασμός αυτών των πρακτικών με μια αυστηρή νομοθεσία μπορεί να βοηθήσει στη διασφάλιση της ασφάλειας στα IoT συστήματα. Η δημιουργία κανόνων και κανονισμών που ορίζουν τις ελάχιστες απαιτήσεις ασφαλείας που πρέπει να πληρούν οι συσκευές IoT, καθώς και η επιβολή κυρώσεων σε περίπτωση παραβίασης των κανόνων αυτών μπορούν να δώσουν μεγαλύτερη ασφάλεια στους χρήστες των IoT συσκευών.

1.8 Ομογενοποίηση Δεδομένων

Στα IoT συστήματα τα δεδομένα τα οποία διακινούνται παράγονται από διαφορετικές πηγές και ετερογενείς τεχνολογίες. Η ομογενοποίηση των δεδομένων, data aggregation, αναφέρεται στη διαδικασία προσαρμογής των δεδομένων σε ένα κοινό μοντέλο, προκειμένου να μπορούν να αξιοποιηθούν από το σύστημα χωρίς δυσκολίες. Επιπλέον, αποτελεί μια διαδικασία ταυτοποίησης, προσαρμογής και ενσωμάτωσης δεδομένων από διαφορετικές πηγές και μορφές σε μια κοινή μορφή και δομή, ώστε να μπορούν να αναλυθούν και να χρησιμοποιηθούν από εφαρμογές και συστήματα πληροφορικής. Η ομογενοποίηση δεδομένων είναι σημαντική στη διαχείριση μεγάλων όγκων δεδομένων και στη δημιουργία ενιαίων πλατφορμών και εφαρμογών που χρησιμοποιούν δεδομένα από πολλαπλές πηγές, κάνοντας εύκολη την αξιοποίηση τους από το σύστημα και την αποφυγή προβλημάτων όπως η διαφορετική δομή των δεδομένων ή η διαφορετική σημασία τους ανάλογα με την πηγή από την οποία προέρχονται.

Η ομογενοποίηση δεδομένων περιλαμβάνει την επεξεργασία και τον συντονισμό των δεδομένων από διάφορες πηγές, έτσι ώστε να δημιουργηθεί ένα κοινό μοντέλο δεδομένων που θα είναι εύκολο να αναλυθεί και να χρησιμοποιηθεί από τα διάφορα συστήματα και εφαρμογές. Τέλος, επιτρέπει την ανάπτυξη ενός ενιαίου μοντέλου δεδομένων για την ανάλυση και την εξαγωγή συμπερασμάτων από τα δεδομένα, καθιστώντας τα πιο χρήσιμα και εύκολα στην επεξεργασία τους.

1.8.1 Διαδικασία Ομογενοποίησης Δεδομένων

Η Ομογενοποίηση των Δεδομένων αποτελεί μια σύνθετη διαδικασία η οποία περιλαμβάνει διάφορα βήματα, από τη συλλογή δεδομένων από διαφορετικές πηγές έως την ενσωμάτωσή τους σε ένα ενιαίο σύστημα αποθήκευσης και διαχείρισης. Τα βήματα αυτά περιλαμβάνουν τη συλλογή και επεξεργασία δεδομένων, τη μετατροπή τους σε ένα κοινό μορφότυπο, τη σύγκριση των δεδομένων και την εύρεση διαφορών ή συμβιβασμών, καθώς και την ενσωμάτωσή τους σε ένα ενιαίο σύστημα αποθήκευσης και διαχείρισης και παρουσιάζονται στην συνέχεια αναλυτικά:

1. Συλλογή δεδομένων: Η διαδικασία ξεκινά με τη συλλογή δεδομένων από διάφορες πηγές, όπως αισθητήρες, βάσεις δεδομένων, συσκευές κ.λπ.

2. Προεπεξεργασία δεδομένων: Με την ολοκλήρωση της συλλογή των δεδομένων, απαιτείται η προεπεξεργασία τους, όπως δηλαδή η απομάκρυνση των επαναλαμβανόμενων, κατεστραμμένων ή ακατάλληλων δεδομένων και η αντικατάσταση των τιμών που δεν έχουν καταγραφεί σωστά. Με αυτόν τον τρόπο επιτυγχάνεται η βελτιστοποίηση της ποιότητας των δεδομένων.
3. Ομογενοποίηση των δεδομένων: Στο συγκεκριμένο βήμα, οι διαφορετικοί τύποι δεδομένων πρέπει να ομογενοποιηθούν ώστε να είναι συμβατοί μεταξύ τους. Αυτό μπορεί να γίνει μέσω της μετατροπής των δεδομένων σε ένα κοινό μορφότυπο και μέσω της συνένωσης των δεδομένων από διαφορετικές πηγές. Το συγκεκριμένο βήμα αποτελεί ένα καίριο σημείο για την εξασφάλιση της συνοχής και της ενότητας των δεδομένων που έχουν συλλεχθεί.
4. Σύγκριση των δεδομένων και εύρεση διαφορών ή συμβιβασμών. Κατά τη διαδικασία της ομογενοποίησης δεδομένων, μπορεί να προκύψει η ανάγκη σύγκρισης των δεδομένων που έχουν συλλεχθεί από διαφορετικές πηγές, προκειμένου να εντοπιστούν διαφορές μεταξύ τους ή ακόμα και συμβιβασμοί. Η σύγκριση αυτή μπορεί να γίνει με διάφορους τρόπους, ανάλογα με τα χαρακτηριστικά των δεδομένων και την απαιτούμενη ακρίβεια της σύγκρισης.
5. Ενσωμάτωση των δεδομένων σε ένα ενιαίο σύστημα αποθήκευσης και διαχείρισης. Στο συγκεκριμένο βήμα, τα δεδομένα πρέπει να μεταφερθούν σε μια κοινή πλατφόρμα ή βάση δεδομένων, όπου μπορούν να αποθηκευτούν, να διαχειριστούν, να τροποποιηθούν και αναζητηθούν από τους χρήστες. Αυτή η πλατφόρμα θα πρέπει να είναι σχεδιασμένη έτσι ώστε να υποστηρίζει την αποθήκευση και την αναζήτηση των δεδομένων σε μια ενιαία μορφή, ανεξάρτητα από την προέλευσή τους.

1.9 IoT και Τεχνητή Νοημοσύνη

Η χρήση τεχνητής νοημοσύνης (Artificial Intelligent - AI) στα συστήματα Internet of Things μπορεί να βελτιώσει σημαντικά τη λειτουργικότητα, την απόδοση και την ασφάλεια των συσκευών και των συστημάτων. Με τη χρήση τεχνητής νοημοσύνης, οι συσκευές IoT μπορούν να επεξεργάζονται και να αναλύουν τα δεδομένα τους πιο αποτελεσματικά και να λαμβάνουν αποφάσεις που βασίζονται στα δεδομένα που συλλέγονται κάνοντας πρόβλεψη.

Η τεχνητή νοημοσύνη μπορεί να χρησιμοποιηθεί σε πολλούς τομείς των συστημάτων IoT, όπως η παρακολούθηση, η ασφάλεια, η διαχείριση ενέργειας, η αναγνώριση προτύπων και η πρόβλεψη συμπεριφοράς. Για παράδειγμα, ένα σύστημα IoT που χρησιμοποιεί τεχνητή νοημοσύνη μπορεί να επεξεργάζεται δεδομένα από αισθητήρες κίνησης και κατανάλωσης ενέργειας για να καταλάβει πότε ένας χώρος χρειάζεται ενεργειακή απόδοση και να ρυθμίσει τα συστήματα φωτισμού και κλιματισμού ανάλογα.

Καθώς το IoT αποτελεί μια τεχνολογία που επιτρέπει στις συσκευές να συνδεθούν μεταξύ τους και με το διαδίκτυο για να ανταλλάσσουν δεδομένα συλλέγοντας μεγάλες ποσότητες δεδομένων από διάφορες πηγές, όπως αισθητήρες, συσκευές και λογισμικό, με σκοπό την ανάλυση και την εξαγωγή πληροφοριών και να υλοποιούν λειτουργίες χωρίς ανθρώπινη παρέμβαση η τεχνητή νοημοσύνη μπορεί να βελτιώσει σημαντικά την απόδοση και την αξιοπιστία των IoT συστημάτων. Η χρήση της Τεχνητής Νοημοσύνης στο IoT μπορεί να προσφέρει πολλά πλεονεκτήματα, όπως η βελτίωση της ενεργειακής απόδοσης και η μείωση των λαθών στη λειτουργία των συστημάτων καθώς αποτελεί ένα σημαντικό πεδίο έρευνας και ανάπτυξης στο τομέα του IoT.

Επιπλέον, η Τεχνητή Νοημοσύνη αποτελεί έναν ευρύτερο όρο που αναφέρεται στην ικανότητα των υπολογιστών να εκτελούν εργασίες που απαιτούν ανθρώπινη νοημοσύνη, όπως αναγνώριση φωνής, αναγνώριση εικόνας και αντιμετώπιση προβλημάτων λογικής, περιλαμβάνοντας πολλές υποκατηγορίες, μεταξύ των οποίων και η μηχανική εκμάθηση.

1.9.1 Αλγόριθμοι Μηχανικής Εκμάθησης στο IoT

Η Μηχανική Εκμάθηση (Machine Learning) είναι ένα πεδίο της Τεχνητής Νοημοσύνης που αφορά την ανάπτυξη αλγορίθμων και τεχνικών για την εκπαίδευση υπολογιστικών συστημάτων, ώστε να μπορούν να αναλύουν δεδομένα και να προβλέπουν τα μελλοντικά αποτελέσματα. Στο πλαίσιο της Τεχνολογίας των Πραγμάτων (IoT), οι αλγόριθμοι μηχανικής εκμάθησης μπορούν να εκμεταλλευτούν τα δεδομένα που συλλέγονται από διαφορετικές πηγές στο IoT για να παράγουν αναλύσεις και προβλέψεις.

Οι αλγόριθμοι μηχανικής εκμάθησης χωρίζονται συνήθως στις παρακάτω τρεις βασικές κατηγορίες:

- i. **Επιβλεπόμενη μάθηση (Supervised Learning):** Οι αλγόριθμοι μηχανικής μάθησης επιβλεπόμενης μάθησης χρησιμοποιούνται όταν υπάρχει διαθέσιμο ένα σύνολο δεδομένων με ετικέτες. Ο στόχος είναι να βρεθεί ένας κανόνας που να προβλέπει τις ετικέτες για νέα δεδομένα που δεν έχουν χρησιμοποιηθεί κατά τη διαδικασία εκπαίδευσης. Η συγκεκριμένη κατηγορία μπορεί να εφαρμοστεί σε διάφορα προβλήματα πρόβλεψης (regression) και ταξινόμησης (classification).
- ii. **Ανεπιβλεπόμενη μάθηση (Unsupervised Learning):** Οι αλγόριθμοι ανεπιβλεπόμενης μάθησης χρησιμοποιούνται όταν δεν υπάρχουν διαθέσιμες ετικέτες στα δεδομένα εκπαίδευσης. Ο στόχος είναι να βρεθούν κρυφά πρότυπα στα δεδομένα. Αυτή η κατηγορία μπορεί να χρησιμοποιηθεί σε προβλήματα συσταδοποίησης (clustering).
- iii. **Ενισχυτική μάθηση (Reinforcement Learning):** Σε αυτήν την κατηγορία, ο αλγόριθμος μάθησης βασίζεται στην αρχή της αλληλεπίδρασης με ένα περιβάλλον μέσω μίας αλυσίδας ενεργειών-ανταμοιβών. Σε αυτό το πλαίσιο, ένας αλγόριθμος ενισχυτικής μάθησης προσπαθεί να εκπαιδεύσει έναν πράκτορα (ή αυτόνομο σύστημα λογισμικού) να πάρει τις σωστές αποφάσεις σε μία σειρά από καταστάσεις, με στόχο να μεγιστοποιήσει μία συνολική ανταμοιβή. Στην ενισχυτική μάθηση, ο πράκτορας αλληλοεπιδρά με το περιβάλλον και λαμβάνει ένα σήμα ανταμοιβής για τις ενέργειες που κάνει.

ΚΕΦΑΛΑΙΟ 2

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑΣ

2.1 Περιγραφή Αρχιτεκτονικής

Στο συγκεκριμένο κεφάλαιο γίνεται αναλυτική περιγραφή της αρχιτεκτονικής της πλατφόρμας που σχεδιάστηκε και χρησιμοποιήθηκε για την εκπόνηση της διπλωματικής εργασίας. Σε κάθε στάδιο αναλύεται ο τρόπος λειτουργίας, ο κώδικας και τα αποτελέσματα.

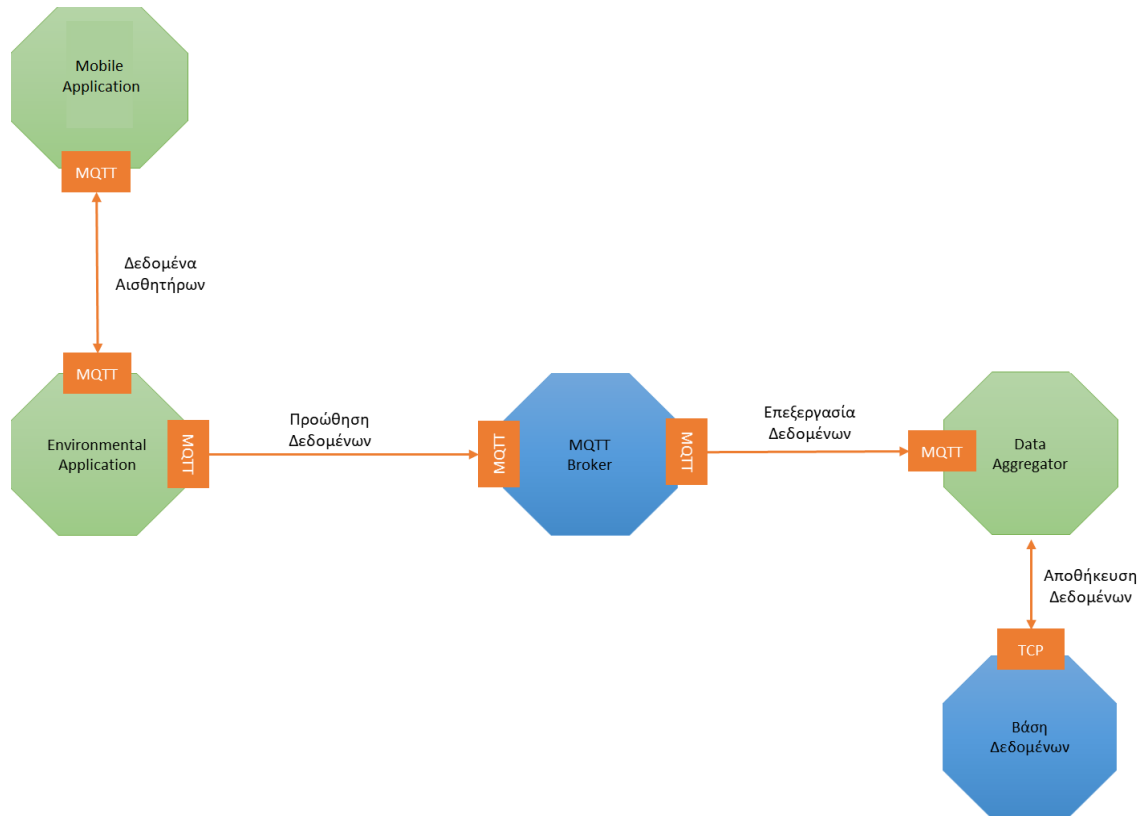
Αρχικά υλοποιήθηκε μία εφαρμογή με γλώσσα προγραμματισμού python, μέσω της οποίας γίνεται η εισαγωγή δεδομένων στην πλατφόρμα. Δηλαδή, μέσω της εφαρμογής συλλέγονται τα δεδομένα από τους αισθητήρες του κινητού τηλεφώνου με την χρήση της εφαρμογής Android Sensor Node, που είναι εγκατεστημένη στην κινητή συσκευή και τα αποστέλλει στον MQTT Broker της EMQX .

Εγκαθιστούμε την Android εφαρμογή Sensor Node στο κινητό τηλέφωνο που θα λειτουργήσει ως ασύρματος αισθητήρας, ο οποίος θα συλλέγει δεδομένα και θα τα στέλνει στον MQTT Broker της EMQX. Τα δεδομένα που έχουμε επιλέξει να στέλνει η Android εφαρμογή είναι αυτά που συλλέγονται από τον αισθητήρα της φωτεινότητας. Ο client, στην περίπτωση μας το κινητό τηλέφωνο, μέσω της εφαρμογής στέλνει, κάνει publish δηλαδή, ένα μήνυμα σε ένα topic το οποίο έχουμε ορίσει, από την εφαρμογή όπου περιέχει τις τιμές του αισθητήρα της φωτεινότητας. Το topic που έχουμε ορίσει πρέπει να είναι το ίδιο που έχει οριστεί στον Broker ώστε να "ακούει".

Στην συνέχεια, ο MQTT Broker, ο οποίος κάνει subscribe στο ίδιο topic, λαμβάνει τα δεδομένα και τα αποστέλλει σε μία άλλη εφαρμογή που έχουμε υλοποιήσει. Ο MQTT Broker είναι ουσιαστικά ένας ενδιάμεσος κόμβος ο οποίος μεταφέρει τα δεδομένα.

Η δεύτερη εφαρμογή είναι σχεδιασμένη να λαμβάνει τα δεδομένα που ανταλλάσσονται μέσω του MQTT Broker, ο οποίος λειτουργεί ως διαμεσολαβητής (middleware) μεταξύ των συσκευών που συνδέονται στο δίκτυο. Στην συνέχεια τα δεδομένα στέλνονται και αποθηκεύονται στην βάση δεδομένων της Cassandra.

Το serialization format που χρησιμοποιούμε για την μετάδοση των δεδομένων από τους αισθητήρες μέχρι την βάση δεδομένων Cassandra είναι η Protocol Buffers. Για της υλοποίηση της πλατφόρμας έχουν δημιουργηθεί το αρχείο sensor.proto το οποίο περιέχει τα δεδομένα του αισθητήρα.



Εικόνα 2.1. Αρχιτεκτονικής Πλατφόρμας

2.2 Environmental Application

Το Environmental Application είναι μία εφαρμογή Python που χρησιμοποιείται για την συλλογή δεδομένων από μια εφαρμογή Android Sensor Node και την εισαγωγή τους στην πλατφόρμα IoT. Η εφαρμογή αποτελείται από τρία κύρια μέρη: τον κώδικα εισαγωγής δεδομένων, τις βιβλιοθήκες για τη σύνδεση με τον MQTT broker και τον κώδικα επεξεργασίας των δεδομένων πριν την εισαγωγή τους στη βάση δεδομένων.

Η εφαρμογή επικοινωνεί με την εφαρμογή Android Sensor Node μέσω του πρωτοκόλλου MQTT, στέλνοντας ερωτήματα για τα δεδομένα που πρέπει να συλλεχθούν και λαμβάνοντας

τα απαντήσεις με τις μετρήσεις. Έπειτα, τα δεδομένα που συλλέγονται από την εφαρμογή Android Sensor Node αποθηκεύονται σε μορφή Protocol Buffers και αποστέλλονται μέσω του πρωτοκόλλου MQTT στον MQTT broker της EMQX.

Για τη μετατροπή των δεδομένων σε μορφή που μπορεί να αναγνωστεί από τη βάση δεδομένων Cassandra, χρησιμοποιείται τα Protocol Buffers. Τα Protocol Buffers αποτελούν αποδοτικό και συμπαγές format, επιτρέποντας την αποθήκευση και τη μεταφορά των δεδομένων με μικρότερη χρήση χώρου και χρόνου από ό,τι άλλες μορφές.

Συνολικά, η αρχιτεκτονική του συστήματος περιλαμβάνει τη χρήση πολλών εργαλείων και τεχνολογιών για τη δημιουργία ενός ολοκληρωμένου συστήματος IoT. Η Python εφαρμογή Environmental Application, οι διακομιστές IoT Gateway και ο MQTT Broker της EMQX συνεργάζονται για την αποτελεσματική συλλογή, μεταφορά και αποθήκευση των δεδομένων. Η χρήση αυτών των τεχνολογιών βελτιώνει την απόδοση και την αξιοπιστία του συστήματος, καθώς επίσης και τη δυνατότητα ανάλυσης και χρήσης των δεδομένων στο μέλλον.

2.3 Data Aggregator

Η Python εφαρμογή, την οποία αναπτύξαμε στην συνέχεια, με το όνομα Data Aggregator αποτελεί έναν κρίσιμο κομμάτι της πλατφόρμας καθώς είναι υπεύθυνη για την λήψη των δεδομένων από τον MQTT broker και την αποθήκευσή τους στην βάση δεδομένων Cassandra. Η επικοινωνία μεταξύ του Data Aggregator και της Βάσης Δεδομένων της Cassandra γίνεται μέσω της Cassandra Python Driver, η οποία παρέχει μια εύχρηστη διεπαφή για την αλληλεπίδραση με τη Βάση Δεδομένων.

Για την μετάδοση των δεδομένων από τους αισθητήρες στο Data Aggregator, χρησιμοποιείται το serialization format των Protocol Buffers. Αυτό επιτρέπει την αποτελεσματική και αποδοτική μετάδοση των δεδομένων μεταξύ τους. Έχει δημιουργηθεί το αρχείο sensor.proto, το οποίο περιέχει την περιγραφή των δεδομένων που στέλνονται από τον αισθητήρα. Μέσω αυτού του αρχείου, δημιουργούνται οι κατάλληλες κλάσεις στην εφαρμογή Data Aggregator, οι οποίες χρησιμοποιούνται για την αποκωδικοποίηση των δεδομένων που λαμβάνονται από τον MQTT broker.

Η αρχιτεκτονική του Data Aggregator περιγράφει τη διαδικασία που ακολουθείται για τη συλλογή και την αποθήκευση των δεδομένων αισθητήρων στη βάση δεδομένων Cassandra. Μέσω της χρήσης του serialization format των Protocol Buffers, τα δεδομένα μεταδίδονται με μικρό κόστος σε χρόνο και χρήση πόρων. Η χρήση του Cassandra εξασφαλίζει την αντοχή και ανθεκτικότητα του συστήματος.

2.4 Βάση Δεδομένων Cassandra

Η βάση δεδομένων ορίζεται ως μία συλλογή από μορφοποιημένα και σχετιζόμενα δεδομένα στην οποία είναι δυνατή η αναζήτηση δεδομένων κατά απαίτηση και η ανάκτηση τους. Συνεπώς αναφερόμαστε σε οργανωμένες ομάδες δεδομένων οι οποίες αποθηκεύονται ψηφιακά. Ιδιαίτερα στα IoT δίκτυα είναι απαραίτητη η χρήση μιας βάσης δεδομένων καθώς εκεί συγκεντρώνονται και ομαδοποιούνται όλα τα δεδομένα και οι μετρήσεις που διακινούν οι συσκευές του δικτύου.

Η βάση δεδομένων που επιλέχθηκε στην παρούσα διπλωματική εργασία είναι η Apache Cassandra, όπου αποτελεί μια διανεμημένη βάση δεδομένων NoSQL και έχει σχεδιαστεί έτσι ώστε να διαχειρίζεται μεγάλους όγκους δεδομένων σε πολλές περιοχές και κατανεμημένες διευθύνσεις IP. Είναι ανθεκτική σε αποτυχίες και παρέχει υψηλή διαθεσιμότητα και κλιμακωσιμότητα αλλά και ασφάλεια.

Η Cassandra χρησιμοποιεί ένα σύστημα κατανεμημένου κλειδιού-τιμής και επιτρέπει τη γρήγορη ανάκτηση δεδομένων με χαμηλή λανθασμένη απόκριση. Αυτό την καθιστά κατάλληλη για εφαρμογές με υψηλό ρυθμό εγγραφής και ανάγνωσης δεδομένων, όπως η καταγραφή γεγονότων, η αποθήκευση καταγραφών μετρήσεων από αισθητήρες IoT και άλλες εφαρμογές που απαιτούν υψηλές επιδόσεις και διαθεσιμότητα.

2.5 Serialization Format

Το serialization format αποτελεί έναν τρόπο για την αποθήκευση και τη μεταφορά δεδομένων από ένα πρόγραμμα σε ένα άλλο ή από έναν υπολογιστή σε έναν άλλο μέσω δικτύου. Στην ουσία, είναι η μετατροπή ενός αντικειμένου σε μια μορφή που μπορεί να αποθηκευτεί ή να μεταδοθεί μέσω διαφορετικών μέσων χωρίς να χαθούν οι πληροφορίες του. Διάφορα serialization formats χρησιμοποιούνται σε διάφορες εφαρμογές και πλατφόρμες, και κάθε ένα έχει τα δικά του πλεονεκτήματα και μειονεκτήματα σε σχέση με άλλα.

Το Protocol Buffer είναι ένα από τα πιο δημοφιλή serialization formats και χρησιμοποιείται συχνά σε διανεμημένα συστήματα για την αποθήκευση και τη μεταφορά δεδομένων. Η κύρια του ιδιότητα είναι η αποδοτικότητα στο μέγεθος των δεδομένων και η ταχύτητα στον τρόπο που αναγνωρίζεται και αναλύεται από τις συστάδες του συστήματος.

Η δομή των δεδομένων στο protocol buffer περιγράφεται σε ένα αρχείο κειμένου .proto με καθορισμένη μορφή. Το αρχείο αυτό περιέχει μια συλλογή από μηνύματα, τα οποία περιγράφουν τη δομή των δεδομένων που θα αποθηκευτούν ή μεταδοθούν. Η μορφή του αρχείου είναι κατανοητή από τον compiler του protocol buffer, ο οποίος μετατρέπει το αρχείο σε κώδικα σε διαφορετικές γλώσσες προγραμματισμού. Επιπλέον, τα αρχεία κειμένου .proto επιτρέπουν στους προγραμματιστές να ορίσουν μια κοινή γλώσσα ανταλλαγής δεδομένων μεταξύ διαφορετικών εφαρμογών, ανεξαρτήτως της γλώσσας προγραμματισμού ή της πλατφόρμας στην οποία λειτουργούν οι εφαρμογές αυτές. Αυτό επιτρέπει την εύκολη επικοινωνία μεταξύ διαφορετικών συστημάτων και εφαρμογών, εξαλείφοντας τον κίνδυνο των ασυμβατοτήτων των δομών δεδομένων και των τύπων δεδομένων.

Τέλος, το protocol buffer παρέχει επίσης τη δυνατότητα επαλήθευσης της δομής των δεδομένων και της συμβατότητάς τους μεταξύ διαφορετικών εκδόσεων του ίδιου προγράμματος. Αυτό διευκολύνει την ανάπτυξη και συντήρηση μεγάλων συστημάτων με πολλά ετερογενή μηνύματα και δεδομένα.

ΚΕΦΑΛΑΙΟ 3

ΥΛΟΠΟΙΗΣΗ ΠΛΑΤΦΟΡΜΑΣ

3.1 Εργαλεία

Στο συγκεκριμένο κεφάλαιο της διπλωματικής, γίνεται αρχικά ανάλυση και παρουσίαση των εργαλείων και των τεχνολογιών που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής. Πιο συγκεκριμένα, παρουσιάζονται και αξιολογούνται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, όπως η γλώσσα προγραμματισμού Python, η βάση δεδομένων Apache Cassandra και ο MQTT Broker της EMQX, καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανταλλαγή δεδομένων, όπως το Protocol Buffers. Σκοπός αυτής της ανάλυσης είναι η κατανόηση της λειτουργίας της κάθε τεχνολογίας και η εξήγηση του ρόλου που διαδραματίζει στην υλοποίηση της πλατφόρμας.

3.1.1 Python

Η γλώσσα προγραμματισμού Python δημιουργήθηκε το 1991 από τον Guido van Rossum και αποτελεί μια υψηλού επιπέδου γλώσσα προγραμματισμού, η οποία είναι απλή στην σύνταξη της, εύκολη στη χρήση της και μοιάζει σε πολλά σημεία με την γλώσσα προγραμματισμού C. Παράλληλα παρέχει ισχυρές δυνατότητες προγραμματισμού και καλύπτει τις ανάγκες που έχουν δημιουργηθεί. Επιπλέον, η Python είναι δωρεάν και ανοιχτού κώδικα και αποτελείται από μια μεγάλη και ενεργή κοινότητα η οποία υποστηρίζει την συνεχή ανάπτυξη της και παρέχει πληθώρα βιβλιοθηκών, πακέτων και κώδικα για διάφορες μορφές εφαρμογών και υλοποιήσεων.

Η Python εκτελείται σε ένα σύστημα διερμηνέα, πράγμα που σημαίνει ότι ο κώδικας που έχουμε γράψει μπορεί να εκτελεστεί άμεσα. Με αυτό τον τρόπο η δημιουργία πρωτοτύπων λογισμικού μπορεί να είναι πολύ γρήγορη. Χρησιμοποιώντας την συγκεκριμένη γλώσσα παρέχεται η δυνατότητα να αναπτυχθούν προγράμματα ακολουθώντας το διαδικαστικό μοντέλο προγραμματισμού ή της αντικειμενοστρέφειας. Η υλοποίηση της διπλωματικής εργασίας γίνεται με την χρήση της Python.

3.1.2 IoT Gateway

Το IoT Gateway είναι μια συσκευή ή λογισμικό που χρησιμοποιείται για τη σύνδεση διαφορετικών συσκευών και πρωτοκόλλων στο Internet of Things (IoT). Καθώς οι συσκευές που συνδέονται στο IoT μπορεί να έχουν διαφορετικά πρωτόκολλα επικοινωνίας και προδιαγραφές, το IoT Gateway λειτουργεί ως μεσολαβητής για να διευκολύνει την επικοινωνία μεταξύ αυτών των συσκευών και να ενισχύσει την ασφάλεια και την ευκολία χρήσης του συστήματος.

Το IoT Gateway μπορεί να παρέχει διάφορες λειτουργίες, όπως τη συλλογή, επεξεργασία και αποθήκευση δεδομένων από διάφορες συσκευές, τη μετατροπή δεδομένων από διαφορετικά πρωτόκολλα σε κοινά πρότυπα και τη διαχείριση του δικτύου συνολικά. Το IoT Gateway μπορεί επίσης να λειτουργεί ως δρομολογητής για τα δεδομένα που ανταλλάσσονται μεταξύ των συσκευών και των εφαρμογών του IoT.

3.1.3 MQTT Broker της EMQX

Το EMQ X είναι ένας ελεύθερος και ανοιχτού κώδικα MQTT broker, που επιτρέπει τη μεταφορά δεδομένων σε πραγματικό χρόνο μεταξύ συσκευών Internet of Things (IoT). Ο EMQ X υποστηρίζει τα πρωτόκολλα MQTT, MQTT-SN και CoAP και παρέχει λειτουργίες όπως ασφάλεια, επεκτασιμότητα και ευελιξία. Επιπλέον, προσφέρει δυνατότητες διαχείρισης, παρακολούθησης και ανάλυσης δεδομένων για το Internet of Things.

Για την υλοποίηση της διπλωματικής εργασίας έχουμε "στήσει" στον τοπικό υπολογιστή τον ενδιάμεσο κόμβο ο οποίος θα μεταφέρει τα δεδομένα τα οποία αφού αναλυθούν στην συνέχεια θα αποθηκευτούν στην βάση δεδομένων της Cassandra. Στην συνέχεια αναλύονται τα βήματα για την εγκατάσταση του MQTT Broker.

1. Κατεβάζουμε το EMQX Broker από την επίσημη ιστοσελίδα του EMQX.
2. Αποσυμπιέζουμε το αρχείο σε έναν φάκελο στον υπολογιστή σου.
3. Ανοίγουμε το παράθυρο εντολών των Windows (Command Prompt).

4. Μεταβαίνουμε στον φάκελο που δημιουργήθηκε μετά την αποσυμπίεση του αρχείου και στην συνέχεια στον υποφάκελο bin.
5. Εκτελούμε την εντολή "emqx install".
6. Αφού ολοκληρωθεί η εγκατάσταση, μπορούμε να ξεκινήσουμε τον MQTT Broker του EMQX εκτελώντας την εντολή "emqx start" ή "emqx console".

Στην συνέχεια, ρυθμίζουμε τον MQTT Broker ανάλογα με τις ανάγκες της πλατφόρμας. Τέλος, μπορούμε να προηγηθούμε στο Dashboard της EMQX εισάγοντας σε ένα browser το URL "localhost:18083"

```
PS C:\emqx> bin/emqx console
EMQX_LOG__FILE_HANDLERS__DEFAULT__ENABLE [log.file_handlers.default.enable]: false
EMQX_LOG__CONSOLE_HANDLER__ENABLE [log.console_handler.enable]: true
EMQX_NODE__DB_ROLE [node.db_role]: core
EMQX_NODE__DB_BACKEND [node.db_backend]: mnesia

C:\emqx>C:\emqx\erts-12.3.2.10\bin\erl.exe -mode embedded -boot "C:\emqx\releases\5.0.21\start" -config "C:\emqx\data\conf\app.2023.04.11.11.24.28.config" -args_file "C:\emqx\data\conf\app.2023.04.11.11.24.28.args" -mnesia dir 'c:/emqx/data/mnesia/emqx@127.0.0.1'
Listener ssl:default on 0.0.0.0:8883 started.
Listener tcp:default on 0.0.0.0:1883 started.
Listener ws:default on 0.0.0.0:8083 started.
Listener wss:default on 0.0.0.0:8084 started.
Listener http:dashboard on :18083 started.
EMQX 5.0.21 is running now!
Eshell V12.3.2.10 (abort with ^G)
v5.0.21(emqx@127.0.0.1)1>
```

Εικόνα 3.1: Εκκίνηση MQTT Broker

3.1.4 Βάση Δεδομένων Cassandra

Η υλοποίηση στην συγκεκριμένη διπλωματική εργασία της Βάσης Δεδομένων θα γίνει μέσω του DevCenter, το οποίο αποτελεί το επίσημο εργαλείο της Apache Cassandra για τη διαχείριση και την ανάπτυξη εφαρμογών. Πρόκειται για ένα γραφικό περιβάλλον που παρέχει πρόσβαση στο σύστημα διαχείρισης βάσεων δεδομένων Cassandra μέσω ενός χρήστη φιλικού προς τον χρήστη περιβάλλοντος. Με το DevCenter, οι προγραμματιστές μπορούν να δημιουργούν, να επεξεργάζονται και να διαχειρίζονται τα δεδομένα Cassandra με ευκολία, επιτρέποντάς τους να εστιάσουν στην ανάπτυξη της εφαρμογής τους αντί να ασχολούνται με τη διαχείριση της βάσης δεδομένων.

Στην συνέχεια θα παρουσιαστεί η αναλυτικά η διαδικασία εγκατάστασης και χρήσης της Βάσης Δεδομένων της Cassandra.

1. Αρχικά πρέπει να εγκαταστήσουμε την βάση δεδομένων Cassandra στον υπολογιστή. Κατεβάζουμε την Cassandra από την επίσημη σελίδα και αφού αποσυμπιέσουμε το αρχείο κάνουμε την εγκατάσταση. Προϋπόθεση είναι να είναι εγκαταστημένη η Java.
2. Ανοίγουμε το αρχείο `cassandra.yaml` στο φάκελο `conf` και ενημερώνουμε τις τιμές για το `rpc_port` και το `native_transport_port` σε 9160 και 9042 αντίστοιχα.
3. Ορίζουμε στον σύστημα του υπολογιστή τις απαραίτητες τιμές περιβάλλοντος των PATH για την Java, την Python και την Cassandra.
4. Αφού επιβεβαιώσουμε ότι η εγκατάσταση έχω ολοκληρωθεί σωστά τρέχουμε την παρακάτω εντολή για να ενεργοποιηθεί η βάση δεδομένων της Cassandra.
`Cassandra -f`
5. Τέλος, πλέον μπορούμε να συνδεθούμε στην βάση δεδομένων της Cassandra χρησιμοποιώντας το εργαλείο `cqlsh`, το οποίο είναι ένα `command line interface` για τη διεπαφή CQL της Cassandra. Απλώς εκτέλεσε την παρακάτω εντολή `cqlsh <ip> <port>`, όπου `<ip>` και `<port>` αντιστοιχούν στη διεύθυνση IP και τη θύρα που έχει οριστεί στο αρχείο `cassandra.yaml`. Η `ip` που έχουμε ορίσει είναι η `localhost` (127.0.0.1) και το `port` είναι η `native_transport_port` την οποία έχουμε ορίσει ως 9042.

Η βάση δεδομένων Cassandra είναι έτοιμη για να αποθηκεύσει τα δεδομένα που στέλνονται από τον MQTT Broker.

3.1.5 Protocol Buffers

Τα Protocol Buffers ή `protobuf` είναι μία μορφή για την αναπαράσταση δεδομένων (`serializing data`) σε μια πλατφόρμα η οποία έχει αναπτυχθεί από την Google. Πρόκειται για ένα `lightweight binary format` που επιτρέπει στις εφαρμογές να αποθηκεύουν και να ανακτούν δεδομένα σε μια γρήγορη, αποτελεσματική και ευέλικτη μορφή.

Τα Protocol Buffers είναι χρήσιμα σε περιβάλλοντα δικτύωσης και σε εφαρμογές που χρειάζονται να μεταφέρουν μεγάλα όγκους δεδομένων μεταξύ ετερογενών τεχνολογιών. Μερικά πλεονεκτήματα των Protocol Buffers αποτελούν η μικρότερη κατανάλωση μνήμης και

εύρος ζώνης σε σχέση με τα αντίστοιχα μορφότυπα όπως το XML ή το JSON και η ευκολία στη χρήση.

Στην συνέχεια παραθέτω ένα απλό παράδειγμα ενός αρχείου Protocol Buffer, το οποίο περιέχει τον ορισμό μας κλάσεις για την αποθήκευση δεδομένων ενός αισθητήρα. Ο αισθητήρας συλλέγει δεδομένα για την θερμοκρασία και την υγρασία του περιβάλλοντος.

```
header {
  requestId: "b94a8da9-1845-4435-b442-8f1fb90004d2"
  requestTime {seconds: 1574351845}
}
payload [{
  type: "temperature"
  value: 19.23
},{
  type: "humidity"
  value: 67.12
}]
```

3.1.6 Sensor Node Εφαρμογή

Η εφαρμογή Sensor Node για Android είναι μια εφαρμογή που μας επιτρέπει να χρησιμοποιήσουμε το κινητό μας τηλέφωνο ως αισθητήρα για να λάβουμε διάφορα περιβαλλοντικά δεδομένα, όπως θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, φωτεινότητα, περιστροφή, επιτάχυνση. Υποστηρίζονται διάφορα πρωτόκολλα επικοινωνίας όπως MQTT και HTTP. Η εφαρμογή χρησιμοποιεί τους ενσωματωμένους αισθητήρες του smartphone για τη μέτρηση των δεδομένων.

Η εφαρμογή είναι χρήσιμη για πειραματικούς σκοπούς και για τη δημιουργία πρωτότυπων έργων IoT χρησιμοποιώντας ένα κινητό τηλέφωνο ως αισθητήρα. Επιπλέον, η εφαρμογή είναι δωρεάν και ανοικτού κώδικα, δίνοντας τη δυνατότητα στους χρήστες να τροποποιήσουν και να προσαρμόσουν την εφαρμογή στις ανάγκες τους.

Τέλος, οι περισσότερες εφαρμογές MQTT Client δεν παρέχουν πρόσβαση στους αισθητήρες του κινητού τηλεφώνου, αλλά επιτρέπουν την απλή αποστολή μηνυμάτων. Παρόλα αυτά, μπορούν να κάνουν και publish και subscribe σε Brokers, πράγμα το οποίο δεν είναι εφικτό με τη SensorNode. Η εφαρμογή Sensor Node υλοποιεί μόνο publish μηνύματα. Τέλος, η δωρεάν έκδοση της εφαρμογής δεν επιτρέπει τη πρόσβαση σε κάποιους από τους αισθητήρες όπως GPS, ποσοστό μπαταρίας και θερμοκρασία.

3.2 Υλοποίηση

Το συγκεκριμένο κεφάλαιο περιγράφει αναλυτικά την μεθοδολογία η οποία έχει επιλεγεί για την υλοποίηση της πλατφόρμας ομογενοποίησης καθώς επίσης αναλύεται και ο λόγος για τον οποίο έχει επιλέγει η συγκεκριμένη μεθοδολογία. Η ανάλυση και η επιλογή μεθοδολογίας βασίστηκε σε πολλούς παράγοντες, όπως οι απαιτήσεις του έργου, οι περιορισμοί χρόνου και πόρων, καθώς και οι εξελίξεις στο πεδίο της τεχνολογίας. Η μέθοδος που επιλέχθηκε θεωρήθηκε η πλέον κατάλληλη και αποτελείται από ένα σύνολο από αρχές, πρακτικές και τεχνικές, τις οποίες οι προγραμματιστές και οι ομάδες ανάπτυξης ακολουθούν για τη διασφάλιση της αξιοπιστίας και της επιτυχίας του έργου.

3.2.1 Υλοποίηση Environmental Application

Στον πρώτο κώδικα python που πρέπει να υλοποιήσουμε, επιδιώκουμε να συλλέξουμε δεδομένα από το android κινητό τηλέφωνο μέσω της εφαρμογής Android Sensor Node Free. Τα δεδομένα που έχουμε επιλέξει να συλλέξουμε είναι η ένταση της φωτεινότητας, light intensity, και στην συνέχεια στέλνονται, δηλαδή γίνονται publish στον MQTT Broker της EMQX.

Ο κώδικας αυτός συνδέεται στον MQTT broker στη διεύθυνση 192.168.2.3 στη θύρα 1883. Στη συνάρτηση main() δημιουργείται το αντικείμενο τύπου Sensor για την ένταση του ήχου. Επιπλέον, τα δεδομένα πρέπει να μετατρέπουν σε string για να μπορούν να σταλούν μέσω του MQTT broker. Το protocol buffers είναι ένα binary format και δεν μπορεί να αποσταλεί απευθείας μέσω του MQTT broker. Έτσι, γίνεται η μετατροπή των δεδομένων σε μια μορφή

που μπορεί να αποσταλεί ως αλφαριθμητικό μέσω του broker. Όταν το μήνυμα ληφθεί από τον παραλήπτη, θα πρέπει να γίνει ξανά η μετατροπή των δεδομένων από το string στο format του protocol buffers.

Για να υλοποιήσουμε την MQTT σύνδεση θα πρέπει να χρησιμοποιήσουμε τη βιβλιοθήκη paho-mqtt και την εγκαθιστούμε με την παρακάτω εντολή.

```
pip install paho mqtt
```

Το serialization format που χρησιμοποιούμε για την μετάδοση των δεδομένων είναι τα protocol buffers και χρησιμοποιούμε στον κώδικα την αντίστοιχη βιβλιοθήκη protobuf. Η εγκατάστασή της γίνεται με την παρακάτω εντολή.

```
pip install protobuf
```

Επίσης, για να μπορέσουμε να υλοποιήσουμε την MQTT σύνδεση αλλά και με το αρχείο sensor.proto, όπου έχουμε καθορίσει τη δομή των δεδομένων που θα σταλούν στον broker, να δημιουργήσουμε το αρχείο Sensor_pb2 θα πρέπει να εισάγουμε τις παρακάτω βιβλιοθήκες στην αρχή του κώδικα.

```
import paho.mqtt.client as mqtt  
import Sensor_pb2
```

Στον python κώδικα, η συνάρτηση on_connect εκτελείται όταν ο client συνδέεται στον MQTT broker και η συνάρτηση on_publish εκτελείται όταν ο client εκδίδει ένα μήνυμα προς δημοσίευση. Η σύνδεση με τον broker γίνεται μέσω της συνάρτησης connect και το μήνυμα που προς δημοσίευση ανήκει στον τύπο sensor_pb2.Sensor.

Ο κώδικας python όπου υλοποιεί το Environmental Application παρατίθεται στο Παράρτημα της Διπλωματικής.

3.2.2 Υλοποίηση Data Aggregator

Στην συνέχεια υλοποιούμε την δεύτερη εφαρμογή σε python η οποία λαμβάνει τα δεδομένα από τον MQTT Broker της EMQX και τα στέλνει στην Βάση Δεδομένων Cassandra όπου και αποθηκεύονται.

Πιο αναλυτικά, η εφαρμογή εγκαθιστά έναν MQTT client και συνδέεται στον MQTT broker. Κάνει subscribe σε ένα topic όπου η android εφαρμογή δημοσιεύει τα δεδομένα. Όταν λαμβάνει ένα μήνυμα σε αυτό το topic, ο client μετατρέπει το φορτίο από δυαδική μορφή σε ένα μήνυμα protobuf όπου περιλαμβάνει τα δεδομένα του αισθητήρα ήχου και τα εκτυπώνει. Στη συνέχεια, τα δεδομένα αποθηκεύονται στην βάση δεδομένων της Cassandra. Ο broker επαναλαμβάνεται την συγκεκριμένη διαδικασία απεριόριστα.

Οι βιβλιοθήκες που πρέπει να εγκατασταθούν είναι οι ίδιες με την πρώτη εφαρμογή, δηλαδή η paho-mqtt και η protobuf καθώς επίσης η βιβλιοθήκη Cassandra-driver που μας επιτρέπει να συνδεθούμε με την Cassandra και να αποθηκεύουμε τα δεδομένα. Η εγκατάσταση του driver της Cassandra στο περιβάλλον της Python υλοποιείται με την εντολή.

```
pip install cassandra-driver
```

Ο κώδικας python όπου υλοποιεί το Data Aggregator στο Παράρτημα της Διπλωματικής.

ΣΥΜΠΕΡΑΣΜΑΤΑ

4.1 Συνολική Αξιολόγηση

Στην διπλωματική εργασία αναπτύξαμε μία πλατφόρμα, η οποία συλλέγει και αποθηκεύει δεδομένα από αισθητήρες μέσω ενός MQTT broker και τα αποθηκεύει στην βάση δεδομένων της Cassandra. Ο σχεδιασμός και η υλοποίηση αυτού του συστήματος βασίζεται σε μια σειρά τεχνολογιών και μεθοδολογιών, όπως τον MQTT πρωτόκολλο επικοινωνίας, τη χρήση της βάσης δεδομένων Cassandra, και το serialization format των δεδομένων του αισθητήρα σε Protocol Buffers.

Αρχικά, επιλέχθηκαν τα κατάλληλα εργαλεία και τεχνολογίες για την υλοποίηση του συστήματος, τα οποία ενσωματώνουν τις πιο πρόσφατες εξελίξεις στο χώρο του IoT και της ομογενοποίησης δεδομένων. Η χρήση της Apache Cassandra ως βάσης δεδομένων είναι επίσης μια καλή επιλογή λόγω της κατανομημένης φύσης της και της ικανότητάς της να διαχειρίζεται μεγάλες όγκους δεδομένων.

Επιπλέον, η χρήση του Protocol Buffers ως serialization format αποτελεί μια πολύ καλή επιλογή για τη μεταφορά των δεδομένων, καθώς είναι ελαφρύ, γρήγορο και αποδοτικό σε σχέση με άλλες μορφές, όπως το JSON και το XML. Αυτό επιτρέπει στο σύστημα να μεταφέρει με ασφάλεια μεγάλους όγκους δεδομένων σε πραγματικό χρόνο, ενώ επιτρέπει την εξοικονόμηση ενέργειας στα συστήματα IoT που λειτουργούν με μπαταρίες.

Με βάση τα αποτελέσματα, μπορούμε να συμπεράνουμε ότι η ανάπτυξη μιας πλατφόρμας IoT βασισμένης στην Python, το Apache Cassandra και το MQTT protocol μπορεί να επιτευχθεί με επιτυχία. Η χρήση του Python ως βασικής γλώσσας προγραμματισμού μας επέτρεψε να αναπτύξουμε γρήγορα μια λειτουργική πλατφόρμα, ενώ η επιλογή του Apache Cassandra και του MQTT protocol εξασφάλισε την αξιοπιστία και την ευελιξία του συστήματος.

Ωστόσο, υπάρχουν κάποιες περιοριστικές παράμετροι που πρέπει να ληφθούν υπόψη σε μια πραγματική εφαρμογή. Το Cassandra είναι ένα σύστημα βασισμένο στη μνήμη, που απαιτεί μεγάλη ποσότητα μνήμης RAM για να λειτουργήσει αποτελεσματικά. Επιπλέον, η

εγκατάσταση και η διαμόρφωση του Cassandra μπορεί να αποτελέσει πρόκληση για ανεμπόδιστη εκτέλεση της εφαρμογής. Τέλος, η απόδοση του συστήματος εξαρτάται σε μεγάλο βαθμό από τη διαμόρφωση του broker MQTT και τη δικτυακή υποδομή.

Όσον αφορά τις βελτιώσεις που μπορούν να γίνουν στη διπλωματική εργασία, μπορεί να εξεταστεί η χρήση ενός συστήματος ανταλλαγής μηνυμάτων ανοιχτού κώδικα, όπως το Apache Kafka, αντί για τον MQTT broker, η χρήση ενός πιο πρόσφατου serialization format όπως το Google's FlatBuffers ή το Apache Avro.

4.2 Μελλοντική Επέκταση του Συστήματος

Βασιζόμενοι στα αποτελέσματα της παρούσας διπλωματικής, μπορούμε να εξετάσουμε πιθανές μελλοντικές επεκτάσεις του συστήματος. Μία από αυτές τις επεκτάσεις είναι η χρήση αλγορίθμων μηχανικής εκμάθησης για την ανάλυση των δεδομένων που συλλέγονται από τους αισθητήρες στο περιβάλλον.

Οι αλγόριθμοι μηχανικής εκμάθησης μπορούν να εξάγουν συσχετίσεις και πρότυπα από τα δεδομένα, και να προβλέπουν μελλοντικές εξελίξεις στο περιβάλλον. Για παράδειγμα, ένας αλγόριθμος μηχανικής εκμάθησης μπορεί να εκπαιδευτεί να ανιχνεύει μοτίβα στα δεδομένα του περιβάλλοντος και να αντιλαμβάνεται τότε οι τιμές των αισθητήρων ξεφεύγουν από τα αναμενόμενα επίπεδα.

Η προσθήκη ενός αλγορίθμου μηχανικής εκμάθησης στο σύστημα είναι μια εξαιρετικά χρήσιμη επέκταση που θα βελτιώσει την απόδοσή του συστήματος και θα το καταστήσει πιο αποτελεσματικό στην παροχή ενημερώσεων και προειδοποιήσεων. Ο αλγόριθμος μηχανικής εκμάθησης θα εκμεταλλεύεται την πληροφορία που συλλέγεται από το σύστημα για να μάθει πώς να αναγνωρίζει πρότυπα στα δεδομένα και να κάνει αποφάσεις βάσει αυτών των προτύπων.

Επιπλέον, θα αναλύει την πληροφορία που συλλέγεται από το σύστημα και θα δημιουργεί μοντέλα που θα χρησιμοποιούνται για να προβλέπουν τις μελλοντικές εξελίξεις. Αυτά τα μοντέλα θα είναι εξαιρετικά χρήσιμα για τη διάγνωση και την πρόληψη πιθανών προβλημάτων που μπορεί να εμφανιστούν στο μέλλον.

Ένα ακόμα βήμα θα ήταν η ανάπτυξη ενός εύχρηστου διασύνδεσης χρήστη, user interface, για την παρακολούθηση των δεδομένων αισθητήρων και των αποτελεσμάτων των αλγορίθμων πρόβλεψης. Αυτή η διεπαφή θα πρέπει να επιτρέπει στους χρήστες να προβάλλουν τα δεδομένα των αισθητήρων, να παρακολουθούν την κατάσταση του συστήματος, και να εκτελούν ενέργειες από απόσταση.

Το user interface μπορεί να υλοποιηθεί με πολλούς τρόπους, αλλά μια δημοφιλής επιλογή είναι η δημιουργία μιας web εφαρμογής που θα εκτελείται σε έναν web server. Οι χρήστες θα μπορούν να αποκτήσουν πρόσβαση στη διεπαφή χρησιμοποιώντας έναν web browser και θα είναι σε θέση να παρακολουθούν τα δεδομένα των αισθητήρων και να εκτελούν ενέργειες από απόσταση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Mattern Friedemann, Floerkemeier Christian (2010). *From the Internet of Computers to the Internet of Things*

Arshdeep Bahga, Vijay Madisetti (2014). *Internet of Things: A Hands-On Approach*

Gaston C. Hillar MQTT Essentials (2017). *A Lightweight IoT Protocol*

Wes McKinney (2022). *Python for Data Analysis*

Jeff Carpenter και Eben Hewitt (2016). *Cassandra: The Definitive Guide*

https://cassandra.apache.org/_/index.html

<https://www.iotforall.com/>

https://en.wikipedia.org/wiki/Internet_of_things

<https://techvidvan.com/tutorials/how-iot-works/>

<https://en.wikipedia.org/wiki/MQTT>

<https://mqtt.org>

<https://www.emqx.io/>

<https://iot.ieee.org/>

ΠΑΡΑΡΤΗΜΑ

Environmental Application

```
import paho.mqtt.client as mqtt
import Sensor_pb2

# Define the IP address and port of the MQTT broker
broker_ip = "192.168.2.3"
broker_port = 1883

# Define the keep alive time in seconds
keep_alive = 60

# Create a MQTT client object
client = mqtt.Client()

# Subscribe to the topic "local"
client.subscribe("sensor/light")

# Define the function to handle the connection
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to broker")
    else:
        print("Connection failed")

# Define the function to handle incoming messages
def on_message(client, userdata, msg):
    # Deserialize the message to a SensorData object
    data = Sensor_pb2.SensorData()
    data.ParseFromString(msg.payload)
```

```
print("Received message: " + str(msg.payload))

# Set the functions to handle the connection and incoming messages
client.on_connect = on_connect
client.on_message = on_message

# Connect to the broker
client.connect(broker_ip, broker_port, keep_alive)

# Create a SensorData message
data = Sensor_pb2.SensorData()

# Populate the message with data
#data.light_intensity = 500

# Serialize the message to bytes
serialized_data = data.SerializeToString()

# Publish the message to the topic "sensor/light"
client.publish("sensor/light", serialized_data)

# Start the loop to process incoming messages
client.loop_forever()
```

Data Aggregator

```
import paho.mqtt.client as mqtt
from cassandra.cluster import Cluster
import Sensor_pb2

# Set up Cassandra connection
cluster = Cluster(['127.0.0.1'])
```

```

session = cluster.connect('sensor_data')

# Set up MQTT client
client = mqtt.Client()
client.connect('192.168.2.3', 1883)

# Define callback function for when a message is received
def on_message(client, userdata, message):
    # Parse the message using Protocol Buffers
    sensor_data = Sensor_pb2.Sensor()
    sensor_data.ParseFromString(message.payload)

    # Extract the light intensity value from the sensor data
    light_intensity = sensor_data.light_intensity

    # Insert the sensor data into Cassandra
    session.execute("INSERT INTO sensor_data (timestamp, light_intensity) VALUES
(now(), %s)", (light_intensity,))

# Subscribe to the MQTT topic
client.subscribe("sensor/light")

# Set up callback function for when a message is received
client.on_message = on_message

# Start the MQTT client loop
client.loop_forever()

```