



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΝΟΣ ΕΞΥΠΝΟΥ ΣΠΙΤΙΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ (INTERNET OF THINGS)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΝΤΕΛΕΗΜΩΝ ΓΑΛΑΤΗΣ

Επιβλέπουσα: Βασιλική Καντερέ

Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2023



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΝΟΣ ΕΞΥΠΝΟΥ ΣΠΙΤΙΟΥ ΜΕ ΧΡΗΣΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΤΩΝ ΠΡΑΓΜΑΤΩΝ (INTERNET OF THINGS)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΝΤΕΛΕΗΜΩΝ ΓΑΛΑΤΗΣ

Επιβλέπουσα: Βασιλική Καντερέ

Επίκουρη Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ...

.....

Β. Καντερέ

Επίκουρη Καθηγήτρια Ε.Μ.Π.

.....

Ν. Κοζύρης

Καθηγητής Ε.Μ.Π.

.....

Σ. Παπαβασιλείου

Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2023

.....

Παντελεήμων Γαλάτης

Διπλωματούχος Ηλεκτρολόγος Μηχανολόγος και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Παντελεήμων Γαλάτης, 2023.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης εργασίας είναι ο σχεδιασμός της βασικής αρχιτεκτονικής σε ένα Διαδίκτυο των Πραγμάτων, με σκοπό να υποστηρίξει τις ανάγκες ενός έξυπνου σπιτιού. Για την επίτευξη του συγκεκριμένου στόχου, θα δοθεί η λίστα των δομικών στοιχείων ενός Διαδικτύου των Πραγμάτων, η οποία αποτελείται από την πλακέτα αισθητήρων (με επιλογές όπως Raspberry Pi, Arduino), το πρωτόκολλο επικοινωνίας (με επιλογές όπως Zigbee, AMQP, MQTT), το μεσίτη μηνυμάτων (με επιλογές όπως Mosquitto, RabbitMQ), τη βάση δεδομένων (με επιλογές όπως SQL, NoSQL, TSDB), το πρόγραμμα εισροής δεδομένων (με επιλογές όπως Apache NiFi, Node-RED) και το πρόγραμμα γραφικής οπτικοποίησης (με επιλογές όπως Grafana, Power BI), ενώ στη συνέχεια θα αναλυθούν οι διαθέσιμες επιλογές για κάθε δομικό στοιχείο, με συγκρίσεις μεταξύ αυτών. Έπειτα, θα επιλεγούν οι καταλληλότερες από αυτές, με σκοπό την πιο αποδοτική λειτουργία της τελικής εφαρμογής και με βάση τις συγκεκριμένες απαιτήσεις της εργασίας. Τέλος, θα υπάρξει μια σύντομη επίδειξη της λειτουργίας αυτής, μέσω εικόνων της εκτέλεσης των δυνατοτήτων της εφαρμογής.

Λέξεις-κλειδιά: Internet of Things, Apache Kafka, Apache , Raspberry Pi, Arduino, Zigbee, AMQP, MQTT, Mosquitto, RabbitMQ, SQL, NoSQL, TSDB, Apache NiFi, Node-RED, Grafana, Power BI

ABSTRACT

The aim of this project is to design the basic architecture of an Internet of Things, in order to support the needs of a smart home. To achieve this goal, the list of building blocks of an Internet of Things will be given, which consists of the sensor board (with options such as Raspberry Pi, Arduino), the communication protocol (with options such as Zigbee, AMQP, MQTT), the message broker (with options such as Mosquitto, RabbitMQ), the database (with options such as SQL, NoSQL, TSDB), the data ingestion program (with options such as Apache NiFi, Node-RED) and the graphical visualization program (with options such as Grafana, Power BI), and then the available options for each building block will be analyzed, with comparisons between them. Then, the most suitable of them will be selected, with the aim of making the final application run more efficiently and based on the specific requirements of the project. Finally, there will be a brief demonstration of this operation, through images of the execution of the application's capabilities.

Keywords: Internet of Things, Apache Kafka, Apache , Raspberry Pi, Arduino, Zigbee, AMQP, MQTT, Mosquitto, RabbitMQ, SQL, NoSQL, TSDB, Apache NiFi, Node-RED, Grafana, Power BI

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαιτέρως τον υποψήφιο διδάκτορα κ. Παρασκευά Κερασιώτη, ο οποίος ήταν πάντοτε βοηθητικός και πρόθυμος να βελτιώσει την ποιότητα της διπλωματικής μου.

Επίσης, ευχαριστώ τους καθηγητές και βοηθούς της σχολής Ηλεκτρολόγων Μηχανολόγων και Μηχανικών Υπολογιστών του Ε.Μ.Π. για τις γνώσεις που μου μετέφεραν στο αντικείμενο της επιστήμης υπολογιστών κατά τη διάρκεια των σπουδών μου.

Τέλος, ένα μεγάλο ευχαριστώ στους φίλους μου και την οικογένειά μου, που με στήριζαν και συνεχίζουν να με στηρίζουν μέχρι και σήμερα.

Περιεχόμενα

1. Εισαγωγή.....	10
1.1 Ορισμός.....	10
1.2 Αρχιτεκτονική του IoT.....	11
1.2.1 Αρχιτεκτονική 3 επιπέδων.....	11
1.2.2 Αρχιτεκτονική 5 επιπέδων.....	12
1.2.3 Αρχιτεκτονική 7 επιπέδων.....	12
1.2 Πλεονεκτήματα.....	13
1.2.1 Έλεγχος και παρακολούθηση.....	13
1.2.2 Εξοικονόμηση κόστους και ενέργειας.....	13
1.2.3 Επιπτώσεις στο περιβάλλον.....	13
1.2.4 Καλύτερη ασφάλεια.....	14
1.2.5 Άνεση.....	14
1.3 Εφαρμογές.....	15
2. Τεχνολογίες IoT.....	17
2.1 Εισαγωγή.....	17
2.1.1 Πλακέτα αισθητήρων.....	17
2.1.2 Βάση δεδομένων.....	18
2.1.3 Πλατφόρμα εισροής δεδομένων.....	19
2.1.4 Πρωτόκολλο επικοινωνίας.....	19
2.1.5 Μεσίτης μηνυμάτων.....	20
2.1.6 Εφαρμογή διαδραστικής οπτικοποίησης.....	20
2.2 Διαθέσιμες επιλογές.....	21
2.2.1 Πλακέτα Αισθητήρων.....	21
2.2.1.1 Raspberry Pi.....	21
2.2.1.2 Arduino.....	22
2.2.1.3 - Διαφορές μεταξύ Raspberry PI και Arduino.....	24

2.2.2 Βάση Δεδομένων.....	26
2.2.2.1 Προκλήσεις βάσης δεδομένων.....	26
2.2.2.2 Τύποι βάσεων δεδομένων.....	27
2.2.2.3 Πλεονεκτήματα βάσεων δεδομένων NoSQL.....	38
2.2.2.4 Μειονεκτήματα βάσεων δεδομένων NoSQL.....	40
2.2.3 Πλατφόρμα Εισροής Δεδομένων.....	42
2.2.3.1 Node-RED.....	42
2.2.3.2 Apache NiFi.....	43
2.2.3.2 StreamSets.....	43
2.2.3.2 Flogo.....	44
2.2.3.2 Διαφορές μεταξύ Node-RED και Apache NiFi.....	45
2.2.4 Πρωτόκολλο Επικοινωνίας.....	46
2.2.4.1 Zigbee.....	46
2.2.4.2 AMQP.....	57
2.2.4.3 MQTT.....	64
2.2.4.4 Η βασική διαφορά μεταξύ AMQP και MQTT.....	83
2.2.5 Message Brokers.....	84
2.2.5.1 RabbitMQ.....	84
2.2.5.2 HiveMQ.....	85
2.2.5.3 Mosquitto MQTT.....	86
2.2.6 ΕΦΑΡΜΟΓΗ ΔΙΑΔΡΑΣΤΙΚΗΣ ΟΠΤΙΚΟΠΟΙΗΣΗΣ.....	87
2.2.6.1 Grafana.....	87
2.2.6.2 Power BI.....	89
2.2.6.3 Zabbix.....	90
2.2.6.4 Tableau.....	91
2.2.6.5 Splunk.....	92
2.2.6.6 Datadog.....	93
3. Σχεδιασμός.....	95

3.1 Στόχοι της εφαρμογής.....	95
3.2 Τύπος δεδομένων IoT.....	96
3.3 Επιλογές IoT.....	96
3.3.1 MONGODB.....	96
3.3.2 MQTT.....	97
3.3.2 Mosquitto MQTT.....	97
3.3.3 Node-RED.....	98
3.3.4 Grafana.....	98
3.4 Παραδοχές.....	99
3.5 Αρχιτεκτονική.....	99
3.6 Σενάρια χρήσης.....	101
3.6.1 Εισαγωγή δεδομένων στο σύστημα μέσω αισθητήρων.....	101
3.6.2 Αποστολή εντολών του χρήστη.....	102
3.6.3 Ανανέωση δεδομένων του συστήματος μέσω κανόνων.....	102
4. Λειτουργίες και Αρχιτεκτονική.....	104
4.1 Ανάπτυξη Σεναρίων Χρήσης.....	104
4.1.1 Εισαγωγή δεδομένων στο σύστημα μέσω αισθητήρων.....	104
4.1.2 Αποστολή εντολών του χρήστη.....	107
4.1.3 Ανανέωση δεδομένων του συστήματος μέσω κανόνων.....	110
4.2 Ανάπτυξη REST API.....	111
5. Συμπεράσματα και Επεκτάσεις.....	117
5.1 Επίτευξη στόχων.....	117
5.2 Επεκτάσεις.....	118
5.2.1 Συστήματα ασφαλείας.....	118
5.2.2 Χρήση Apache NiFi αντί για Node-RED.....	118
5.2.3 Ισοζύγισι φορτίου (load balancing).....	118
5.2.4 Χρήση Apache Kafka αντί για MQTT.....	119
BIBΛΙΟΓΡΑΦΙΑ.....	120

1. Εισαγωγή

Σκοπός της εργασίας είναι η αρχιτεκτονική ενός έξυπνου σπιτιού, το οποίο θα μπορεί να ενημερώνει το χρήστη για διάφορες μετρήσεις (όπως π.χ. τη θερμοκρασία) ανά τακτά χρονικά διαστήματα, ενώ παράλληλα θα μπορεί να δέχεται και οδηγίες από το χρήστη για να εκτελέσει διάφορες ενέργειες (όπως π.χ. να ανοίξει το κλιματιστικό αν η θερμοκρασία ξεπεράσει κάποια όρια ή όταν ο χρήστης πατήσει ένα κουμπί). Για να δημιουργηθεί το συγκεκριμένο δίκτυο απαιτούνται μερικοί υπολογιστές (ή αντίστοιχες συσκευές με διεπαφή αλληλεπίδρασης), αρκετοί αισθητήρες για να τροφοδοτούν το σύστημα με τις κατάλληλες μετρήσεις και προφανώς τα κατάλληλα προγράμματα και πρωτόκολλα επικοινωνίας μεταξύ αυτών.

1.1 Ορισμός

Δεν υπάρχει επίσημος ορισμός της έννοιας "έξυπνο σπίτι". Αυτό που ξεκίνησε περισσότερο ως λέξη-κλειδί έχει εισέλθει στην κοινή χρήση, αλλά κανένα άτομο ή εταιρεία δεν εφηύρε τα έξυπνα σπίτια, ούτε μπορεί να διεκδικήσει την απόλυτη κυριαρχία στον τομέα. Ακόμη και μεταξύ εκείνων που την έχουν υιοθετήσει, ένα έξυπνο σπίτι μπορεί να μοιάζει πολύ διαφορετικό από ένα άλλο. Οι διαφορές περιλαμβάνουν το ποιες έξυπνες συσκευές υπάρχουν στο σπίτι και ακόμη και το πόσες από αυτές τις συσκευές είναι παρούσες.

Στον πυρήνα του, ένα έξυπνο σπίτι ορίζεται ίσως καλύτερα ως ένας χώρος διαβίωσης με συνδεδεμένη τεχνολογία που προορίζεται να βελτιώσει το σπίτι [1]. Αυτό θα μπορούσε να περιλαμβάνει έξυπνους θερμοστάτες, έξυπνα φώτα, έξυπνα βύσματα ή πρίζες, έξυπνα κουδούνια και κλειδαριές. Θα μπορούσε να είναι οποιοσδήποτε συνδυασμός των παραπάνω, όλα τα παραπάνω ή να είναι εντελώς άλλες έξυπνες συσκευές. Το στοιχείο που συνδέει όλα τα έξυπνα σπίτια είναι το ότι όλα, με τον ένα ή τον άλλο τρόπο, χρησιμοποιούν την έννοια του Internet of Things (IoT).

Το Διαδίκτυο των Πραγμάτων (IoT) περιγράφει το δίκτυο φυσικών αντικειμένων - "πραγμάτων" - που είναι ενσωματωμένα με αισθητήρες, λογισμικό και άλλες τεχνολογίες με σκοπό τη σύνδεση και την ανταλλαγή δεδομένων με άλλες συσκευές και συστήματα μέσω του Διαδικτύου. Οι συσκευές αυτές κυμαίνονται από συνηθισμένα οικιακά αντικείμενα έως εξελιγμένα βιομηχανικά εργαλεία.

1.2 Αρχιτεκτονική του IoT

Λόγω της μεγάλης ποικιλίας συσκευών και αναγκών χρηστών που εξυπηρετεί, στο IoT δεν υπάρχει συγκεκριμένο καθολικό διάγραμμα αρχιτεκτονικής. Ωστόσο, μια γενική εικόνα των στοιχείων του IoT θα ήταν η παρακάτω:

- Στρώμα συσκευών: Περιέχει έναν ή παραπάνω αισθητήρες με μικρό μέγεθος, κόστος και κατανάλωση ενέργειας.
- Στρώμα Επικοινωνίας: Περιέχει πρωτόκολλα όπως REST, MQTT ή HTTP.
- Στρώμα διαύλου: Περιέχει HTTP Server και συχνά MQTT Broker. Συγκεντρώνει και οργανώνει τα δεδομένα μέσω θυρών, ενώ παράλληλα διαμορφώνει τον τύπο των δεδομένων αυτών έτσι ώστε να μπορούν να χρησιμοποιηθούν από διαφορετικά πρωτόκολλα.
- Επεξεργασία γεγονότων: Προεπεξεργασία με εφαρμογή cloud και edge computing.
- Στρώμα εφαρμογής: Πρόκειται για μια μηχανή βασισμένη στο Web (Web based engine) για επικοινωνία με άλλα API (API Management), η οποία δίνει τη δυνατότητα επικοινωνίας και με συσκευές εκτός του δικτύου του IoT μέσω του dashboard.

1.2.1 Αρχιτεκτονική 3 επιπέδων

Η πιο βασική μορφή αρχιτεκτονικής είναι 3 επιπέδων.

- i. Στρώμα αισθητήρων: συγκεντρώνονται πληροφορίες για το περιβάλλον καθώς και τιμές μεταβλητών που σχετίζονται με αυτό.
- ii. Στρώμα δικτύου: σε αυτό το επίπεδο πραγματοποιείται η σύνδεση των συσκευών που χρησιμοποιούνται από το IoT και γενικά των συσκευών δικτύου. Επίσης, διαβιβάζονται και προεπεξεργάζονται τα δεδομένα που συλλέγονται από το στρώμα αισθητήρων.
- iii. Στρώμα εφαρμογής: Στο συγκεκριμένο στρώμα βρίσκεται το σύνολο των εφαρμογών οι οποίες χρησιμοποιούνται από το IoT. Ευθύνη του είναι η διανομή πληροφοριών, με τη χρήση υπηρεσιών, στους χρήστες, οι οποίοι τις διαχειρίζονται καταλλήλως.

1.2.2 Αρχιτεκτονική 5 επιπέδων

Η αρχιτεκτονική του Σχήματος 1.1 μπορεί να χωριστεί επιμέρους και να προκύψει αυτή των 5 στρωμάτων.

- (i) Στρώμα αισθητήρων: Ισχύουν όσα αναφέρθηκαν στην αρχιτεκτονική τριών στρωμάτων.
- (ii) Στρώμα μεταφοράς: Το συγκεκριμένο στρώμα μεταφέρει δεδομένα από τους αισθητήρες στα στρώματα εντός του δικτύου, όπως το wireless, το 3G, το 4G, το 5G, το LAN ή το Bluetooth.
- (iii) Στρώμα επεξεργασίας: Αποθηκεύει, επεξεργάζεται και αναλύει το μεγάλο όγκο δεδομένων που παράγονται στα προηγούμενα στρώματα, με τεχνικές όπως cloud computing και big data analysis.
- (iv) Στρώμα εφαρμογής: Ισχύουν όσα αναφέρθηκαν στην αρχιτεκτονική τριών στρωμάτων.
- (v) Στρώμα επιχείρησης: Πρόκειται για τη διαχείριση του IoT από ένα σύνολο διαφορετικών οπτικών και εξασφάλιση ασφάλειας και ιδιωτικότητας στον χρήστη. Έχει στενή σύνδεση με την εμπορική χρήση του IoT ως τεχνολογία εφαρμοσμένη σε παραγωγικές μονάδες.

1.2.3 Αρχιτεκτονική 7 επιπέδων

Είναι η αρχιτεκτονική η οποία χρησιμοποιείται πιο συχνά για την περιγραφή της δομής ενός συστήματος IoT και αποτελείται από τα εξής επίπεδα:

- (i) Αγορά: Πρόκειται για έξυπνο σπίτι, έξυπνο σύστημα υγείας κ.α.
- (ii) Απόκτηση πληροφορίας: Πρόκειται για αισθητήρες που συλλέγουν δεδομένα από το περιβάλλον, όπως αισθητήρες επιπέδου υγρασίας, θερμοκρασίας, κάμερες κ.α.
- (iii) Ενδοεπικοινωνία: Μεταφέρει τα δεδομένα τα οποία συλλέγουν οι σένσορες, συνήθως στο cloud ή σε ένα κέντρο δεδομένων. Αποτελούν μεγάλα δεδομένα.
- (iv) Ενσωμάτωση: Τα δεδομένα του προηγούμενου επιπέδου συγκεντρώνονται και ενσωματώνονται σε άλλα σύνολα δεδομένων.
- (v) Ανάλυση: Τα συνδυασμένα αυτά δεδομένα στη συνέχεια, αναλύονται, χρησιμοποιώντας τεχνικές μηχανικής μάθησης και εξόρυξης γνώσης από δεδομένα.
- (vi) Εφαρμογές: Το επίπεδο αυτό σχετίζεται με την υλοποίηση των κατανεμημένων εφαρμογών αυτών, όπως επίσης και με είδη λογισμικών όπως αρχιτεκτονική προσανατολισμένη προς υπηρεσίες, δικτύωση ορισμένη από το λογισμικό κ.α.
- (vii) Υπηρεσίες: Το επίπεδο αυτό παρέχει υπηρεσίες στο χρήστη, όπως διαχείριση ενέργειας, μεταφορές, εκπαίδευση κ.α.

Πέρα από τα 7 αυτά στρώματα, είναι απαραίτητο να διαχειρίζεται σωστά κάθε επίπεδο και να διατηρείται η ασφάλεια αυτού.

1.2 Πλεονεκτήματα

Μερικές από τα πλεονεκτήματα του IoT για έξυπνο σπίτι είναι τα εξής:

1.2.1 Έλεγχος και παρακολούθηση

Το IoT βελτιώνει σημαντικά τον τρόπο με τον οποίο μπορεί κανείς να ελέγχει και να παρακολουθεί όλες τις διαδικασίες που λαμβάνουν χώρα στο σπίτι του. Το ψυγείο, για παράδειγμα, μπορεί να ειδοποιήσει αν το γιαούρτι θα χαλάσει σε δύο ημέρες ή μπορεί να προσθέσει γάλα στη λίστα με τα ψώνια.

Κάθε μέρα η εφαρμογή IoT συλλέγει δεδομένα σχετικά με τον τρόπο λειτουργίας του σπιτιού, τα επεξεργάζεται και εμφανίζει τις σημαντικότερες πληροφορίες.

1.2.2 Εξοικονόμηση κόστους και ενέργειας

Με λεπτομερή στατιστικά στοιχεία σχετικά με την εργασία και την κατανάλωση ενέργειας κάθε έξυπνης συσκευής, μπορεί κανείς εύκολα να βελτιστοποιήσει τη χρήση τους και να προσαρμόσει τις ρυθμίσεις του έξυπνου σπιτιού του με πιο οικονομικό τρόπο. Μπορεί επίσης να επιτρέψει στο σύστημα έξυπνου σπιτιού να κάνει τα πάντα μόνο του. Το σύστημα θα απενεργοποιήσει τις συσκευές που δεν χρησιμοποιούνται και θα μειώσει την κατανάλωση ενέργειας εάν δεν υπάρχει κανείς στο σπίτι.

1.2.3 Επιπτώσεις στο περιβάλλον

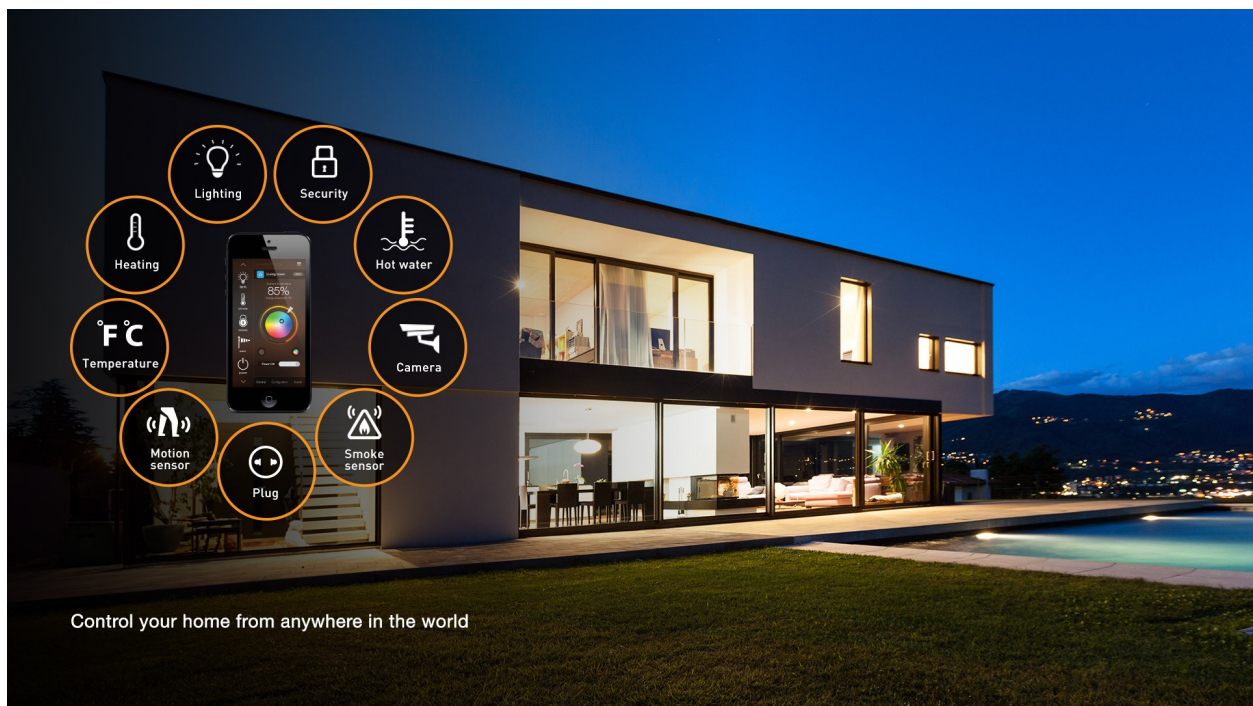
Εκτός από τη μείωση των λογαριασμών, η εξοικονόμηση ενέργειας συμβάλλει στη μείωση του αρνητικού αντίκτυπου στο περιβάλλον και σε μια πιο "πράσινη" ζωή.

1.2.4 Καλύτερη ασφάλεια

Το έξυπνο οικιακό σύστημα ασφαλείας είναι ένα εξαιρετικό εργαλείο για την προστασία της περιουσίας, καθώς παρέχει πλήρη έλεγχο των πάντων εντός και εκτός του σπιτιού. Κάμερες ασφαλείας, έξυπνες κλειδαριές, καθώς και αισθητήρες κίνησης, καπνού και άλλοι αισθητήρες - όλα αυτά τα gadgets συνεργάζονται και ειδοποιούν το χρήστη αν κάτι πάει στραβά. Μπορεί να ελέγχει τι συμβαίνει μέσα και γύρω από το σπίτι του από το γραφείο ή από μια παραλία στην άλλη άκρη του κόσμου [2].

1.2.5 Άνεση

Οι άνθρωποι πάντα ήθελαν να έχουν τα πάντα με μεγαλύτερη άνεση και ευκολία, και το IoT στα έξυπνα σπίτια, ίσως, είναι το μεγαλύτερο επίτευγμα προς αυτή την κατεύθυνση. Αντιμετωπίζει τέλεια το έργο της απλοποίησης και της βελτίωσης της ζωής : πολλές διαδικασίες στο σπίτι λειτουργούν αυτόνομα, ωστόσο ο συνολικός έλεγχος παραμένει στα χέρια του χρήστη.



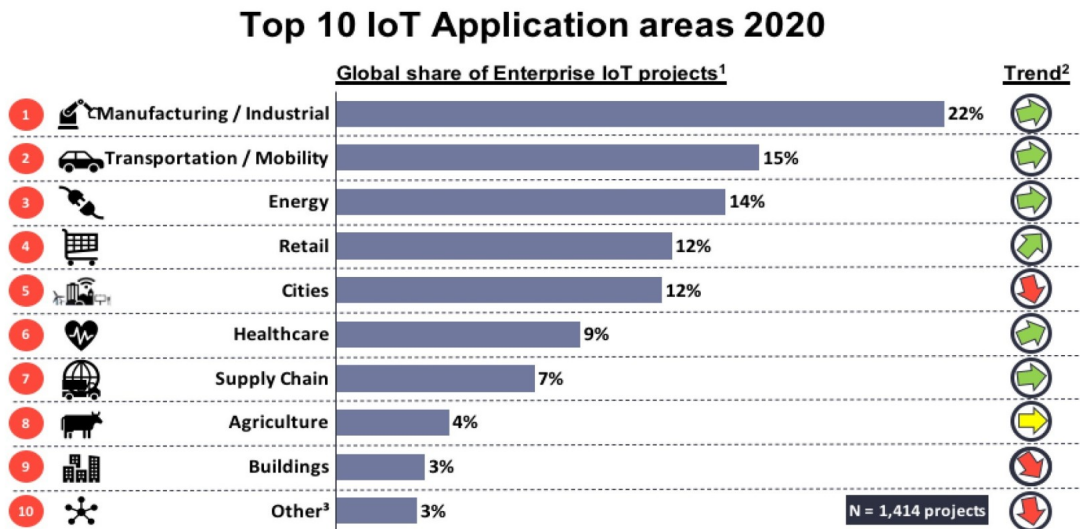
Σχήμα 1.1: Χρήση IoT σε έξυπνο σπίτι

Πηγή: <https://www.elitedevelopers.co.in/blog/2020/04/18/smart-homes/>

1.3 Εφαρμογές

Εφαρμογές του IoT συναντώνται σε τομείς όπως:

- **Wearable:** Τομέας με μεγάλη ανάπτυξη λόγω της αυξανόμενης ζήτησης της αγοράς. Ήδη προσελκύει πολλές εταιρείες κολοσσούς και startups.
- **Υγεία:** Αυξανόμενη ζήτηση για “έξυπνα” προϊόντα υγείας. Ήδη αρκετά προϊόντα αυτού του τομέα βρίσκονται στην αγορά με υπόσχεση για βελτίωση στο μέλλον. Έξυπνα αναπηρικά αμαξίδια, υπενθύμιση για φαρμακευτική αγωγή και γενικά παρακολούθηση υγείας από εφαρμογές αποτελούν τα πλέον διαδεδομένα παραδείγματα [3].
- **Γεωργία:** Έξυπνοι τρόποι για καλλιέργεια της γης και εκμετάλλευση της με έλεγχο υγρασίας και θρεπτικών συστατικών. Παρακολούθηση κτηνοτροφικών μονάδων με αισθητήρες στα ζώα και γεωγραφικό εντοπισμό τους.
- **Έξυπνες πόλεις:** Έργο που εκπονείται από την κυβέρνηση της Ινδίας με σκοπό την δημιουργία αυτοεξυπηρετούμενης έξυπνης πόλης και σπιτιών.
- **Αυτοκινητοβιομηχανία:** Μεγάλη αύξηση στην ζήτηση αυτοκινήτων δίχως οδηγό με επιπλέον συστήματα ασφαλείας γεωγραφικό εντοπισμό και δυνατότητα επικοινωνίας με άλλες συσκευές. Σε δοκιμαστικό ακόμα στάδιο από την Google.
- **Έξυπνο πλέγμα:** Προσπάθεια για έλεγχο ενέργειας και νερού του κάθε σπιτιού και ακινήτου μέσω αισθητήρων και του Internet.
- **Βιομηχανικό Internet:** Ήδη εν μέρη εφαρμοζόμενο με την αντικατάσταση όλο και περισσότερου έμβιου εργατικού δυναμικού με μηχανές για μικρότερη πιθανότητα λάθους και μικρότερη επικινδυνότητα και νομική ευθύνη της εταιρείας.

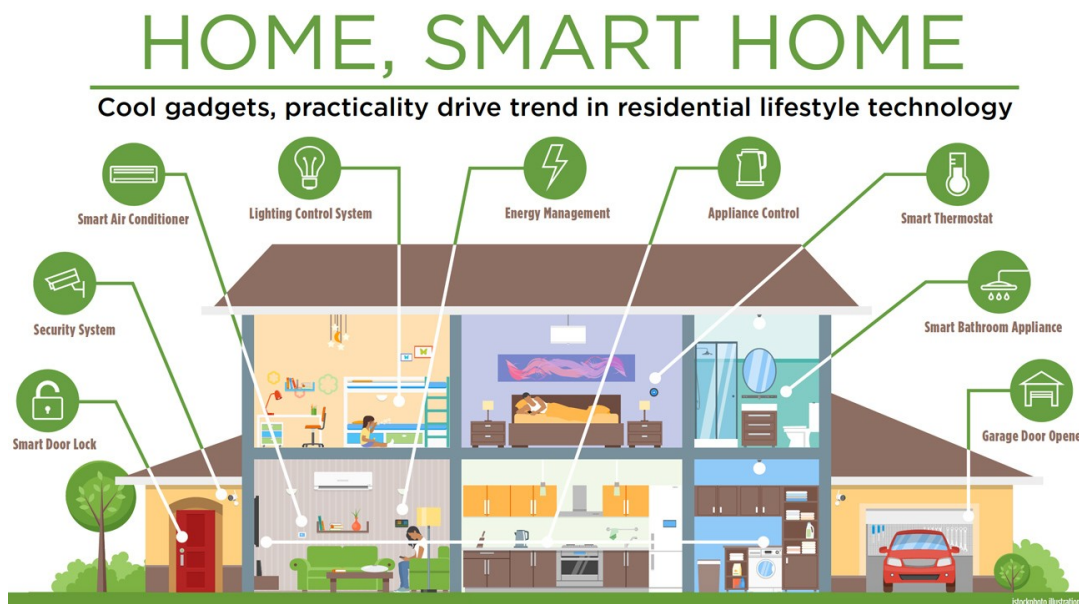


Σχήμα 1.2: Top 10 IoT applications in 2020.

Πηγή: <https://iot-analytics.com/top-10-iot-applications-in-2020/>

Πιο συγκεκριμένα, μερικές ενδεικτικές εφαρμογές του IoT σε έξυπνο σπίτι είναι οι εξής [4]:

- έξυπνοι θερμοστάτες
- έξυπνα ανεμιστήρια
- έξυπνοι ανεμιστήρες οροφής
- έξυπνος υγραντήρας ολόκληρου του σπιτιού
- έξυπνο ψυγείο
- έξυπνο πλυντήριο πιάτων
- έξυπνη συσκευή ενδοεπικοινωνίας
- έξυπνο κλειδί πόρτας / έξυπνη κλειδαριά
- έξυπνες κάμερες
- έξυπνο κουδούνι πόρτας
- έξυπνος αισθητήρας διαρροής νερού
- έξυπνος μηχανισμός γκαραζόπορτας
- έξυπνες ηλεκτρικές σκούπες
- αυτοκαθαριζόμενο πλακάκι
- έξυπνο μπάνιο
- έξυπνοι λαμπτήρες φωτισμού



Σχήμα 1.3: Αναπαράσταση έξυπνου σπιτιού

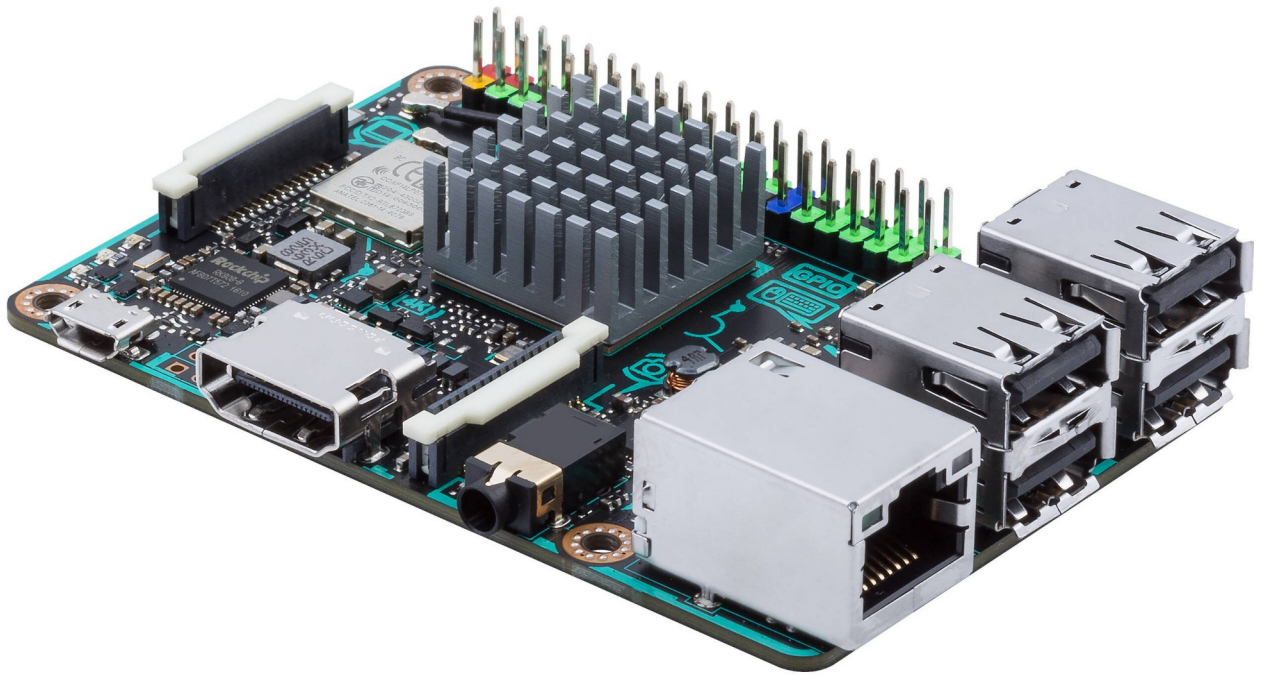
Πηγή: <https://www.littlegeckotech.com/smart-houses-role-technology>

2. Τεχνολογίες IoT

2.1 Εισαγωγή

Οι τεχνολογίες είναι οι εξής:

2.1.1 Πλακέτα αισθητήρων

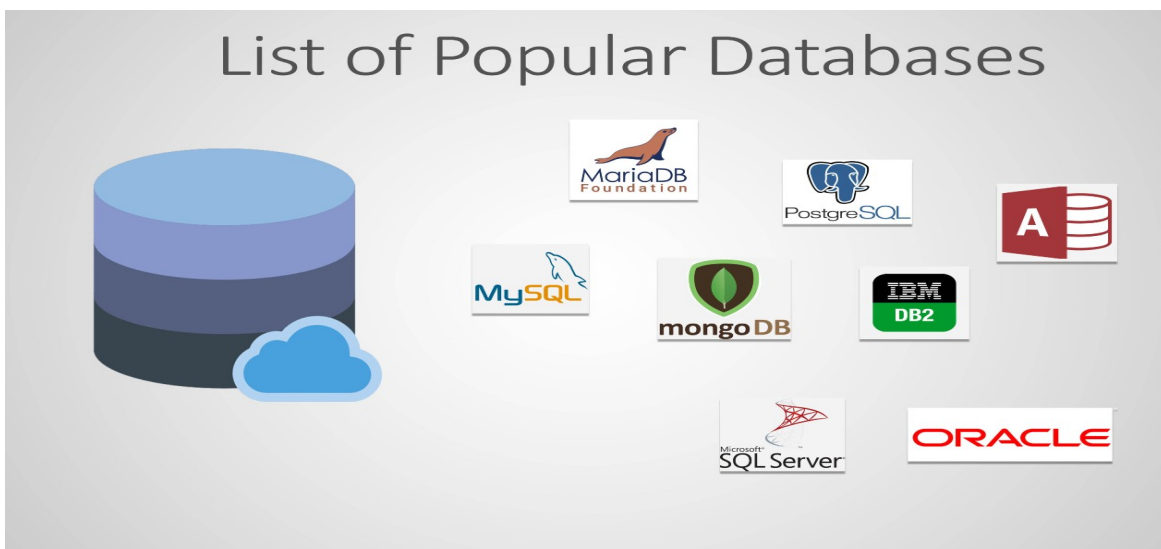


Σχήμα 2.1: Υπολογιστής μίας πλακέτας (SBC)

Πηγή: <https://www.electronics-lab.com/top-10-single-board-computers-of-2020/>

Ο υπολογιστής μίας πλακέτας (SBC) είναι ένας πλήρης υπολογιστής κατασκευασμένος σε μία μόνο πλακέτα κυκλώματος, με μικροεπεξεργαστή(ες), μνήμη, είσοδο/έξοδο (I/O) και άλλα χαρακτηριστικά που απαιτούνται για έναν λειτουργικό υπολογιστή [5]. Πολλοί τύποι οικιακών υπολογιστών ή φορητών υπολογιστών ενσωματώνουν όλες τις λειτουργίες τους σε μία μόνο πλακέτα τυπωμένου κυκλώματος. Σε αντίθεση με έναν επιτραπέζιο προσωπικό υπολογιστή, οι υπολογιστές μιας πλακέτας συχνά δεν βασίζονται σε υποδοχές επέκτασης για περιφερειακές λειτουργίες ή επέκταση. Οι υπολογιστές μιας πλακέτας έχουν κατασκευαστεί χρησιμοποιώντας ένα ευρύ φάσμα μικροεπεξεργαστών. Για να ολοκληρωθεί η επικοινωνία μεταξύ των αισθητήρων και του υπόλοιπου συστήματος, απαιτείται η ύπαρξη μιας πλακέτας, η οποία θα μετατρέπει τις αναλογικές μετρήσεις των αισθητήρων σε ψηφιακά δεδομένα και θα τα εισάγει στο IoT.

2.1.2 Βάση δεδομένων



Σχήμα 2.2: Λίστα δημοφιλών βάσεων δεδομένων

Πηγή: <https://www.trickyenough.com/most-popular-databases/>

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δομημένων πληροφοριών ή δεδομένων, που συνήθως αποθηκεύονται ηλεκτρονικά σε ένα σύστημα υπολογιστή. Μια βάση δεδομένων ελέγχεται συνήθως από ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Τα δεδομένα και το DBMS, μαζί με τις εφαρμογές που συνδέονται με αυτά, αναφέρονται ως σύστημα βάσεων δεδομένων, συχνά συντομευμένο σε βάση δεδομένων [6].

Η βάση δεδομένων είναι απαραίτητη στο σχεδιασμό του έξυπνου σπιτιού, καθώς είναι σημαντικό να αποθηκεύονται διάφορες μετρήσεις από τους αισθητήρες και να εμφανίζονται είτε ως μεμονωμένες μετρήσεις, είτε συνολικά ως μια συνάθροιση μετρήσεων (π.χ. για την περίπτωση της θερμοκρασίας, θα μπορούσε να δίνεται στο χρήστη η μέση θερμοκρασία το τελευταίο λεπτό). Συνήθως, οι αισθητήρες

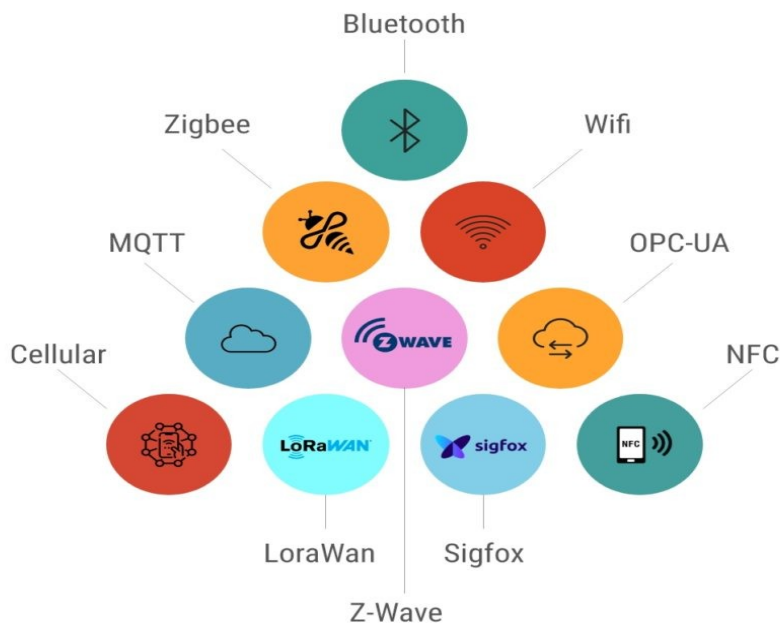
παρέχουν δεδομένα πολύ πιο συχνά από όσο ένας άνθρωπος ελέγχει το σύστημα, οπότε η συνάθροιση των δεδομένων αυτών σε χρονικά διαστήματα είναι ο συνηθισμένος τρόπος παρουσίασης.

Τα δεδομένα στους πιο συνηθισμένους τύπους βάσεων δεδομένων που λειτουργούν σήμερα συνήθως μοντελοποιούνται σε γραμμές και στήλες σε μια σειρά πινάκων για να είναι αποτελεσματική η επεξεργασία και η αναζήτηση δεδομένων. Τα δεδομένα μπορούν στη συνέχεια να είναι εύκολα προσβάσιμα, να διαχειρίζονται, να τροποποιούνται, να ενημερώνονται, να ελέγχονται και να οργανώνονται. Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν δομημένη γλώσσα ερωτημάτων (SQL) για τη συγγραφή και την αναζήτηση δεδομένων.

2.1.3 Πλατφόρμα εισροής δεδομένων

Η πλατφόρμα εισροής δεδομένων είναι ένα πρόγραμμα που χειρίζεται την είσοδο των δεδομένων στο σύστημα και την πορεία αυτών εντός του συστήματος. Συγκεκριμένα, πρέπει να συντονίζει την επικοινωνία της πλακέτας με τη βάση, της βάσης με το χρήστη και του χρήστη με την πλακέτα.

2.1.4 Πρωτόκολλο επικοινωνίας

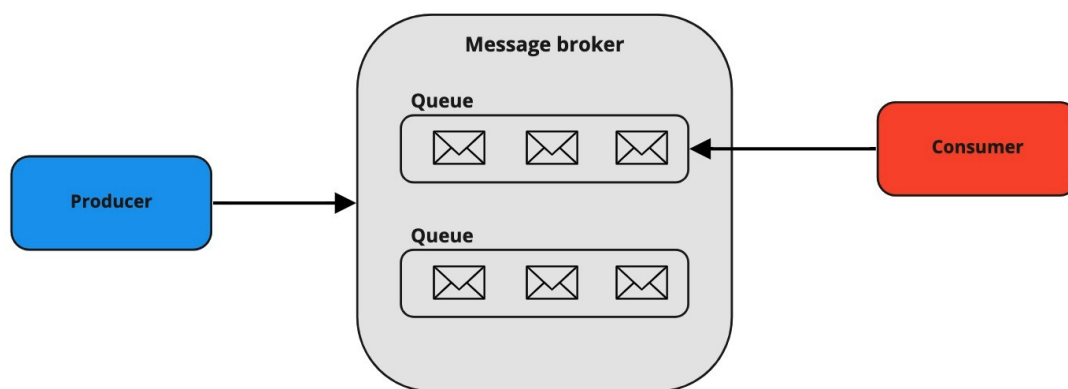


Σχήμα 2.3: Πιθανές επιλογές πρωτοκόλλου επικοινωνίας

Πηγή: <https://hashstudioz.com/blog/top-iot-communication-protocols-2020/>

Για την επικοινωνία μεταξύ των συσκευών πρέπει να χρησιμοποιηθεί το κατάλληλο πρωτόκολλο. Ένα πρωτόκολλο δικτύου όπως το HTTP είναι αρκετά σπάταλο σε δεδομένα που δεν χρειάζονται σε ένα εσωτερικό δίκτυο όπως σε ένα IoT. Ιδανικά προτιμάται ένα πιο απλό πρωτόκολλο, ώστε να μπορεί να εκτελεστεί γρήγορα στους μικροεπεξεργαστές που ελέγχουν τους αισθητήρες, αλλά παράλληλα και αξιόπιστο για την επικοινωνία των συσκευών.

2.1.5 Μεσίτης μηνυμάτων



miro

Σχήμα 2.4: Αναπαράσταση μεσίτη μηνυμάτων

Πηγή:<https://tsh.io/blog/message-broker/>

Ένας μεσίτης μηνυμάτων (επίσης γνωστός ως μεσίτης ολοκλήρωσης ή μηχανή διασύνδεσης) είναι μια ενδιάμεση μονάδα προγράμματος υπολογιστή που μεταφράζει ένα μήνυμα από το επίσημο πρωτόκολλο ανταλλαγής μηνυμάτων του αποστολέα στο επίσημο πρωτόκολλο ανταλλαγής μηνυμάτων του παραλήπτη [7]. Οι διαμεσολαβητές μηνυμάτων είναι στοιχεία σε δίκτυα τηλεπικοινωνιών ή υπολογιστών όπου οι εφαρμογές λογισμικού επικοινωνούν ανταλλάσσοντας τυπικά καθορισμένα μηνύματα.

2.1.6 Εφαρμογή διαδραστικής οπτικοποίησης

Μια εφαρμογή διαδραστικής οπτικοποίησης παρέχει διαγράμματα, γραφήματα και ειδοποιήσεις για τον ιστό όταν συνδέεται με υποστηριζόμενες πηγές δεδομένων. Στην περίπτωση ενός IoT, σκοπός του είναι να ενημερώνει εύκολα και γρήγορα το χρήστη για τυχόν αλλαγές στην κατάσταση του σπιτιού, καθώς επίσης να επιτρέπει την αποστολή οδηγιών από το χρήστη προς το IoT.

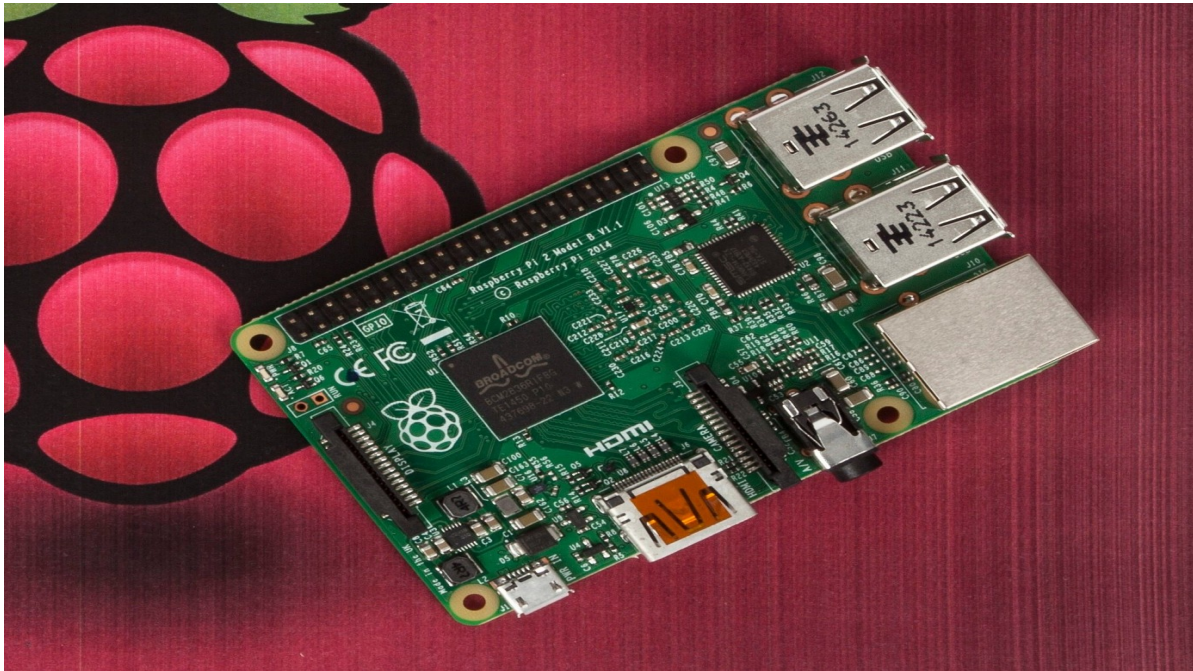
Στη συνέχεια θα αναλυθούν τα βασικά δομικά συστατικά ενός τέτοιου σχεδιασμού, με μια σύντομη αναφορά εναλλακτικών επιλογών για το κάθε ένα.

2.2 Διαθέσιμες επιλογές

2.2.1 Πλακέτα Αισθητήρων

Για την υλοποίηση του SBC θα μπορούσαν να χρησιμοποιηθούν πολλές διαφορετικές πλακέτες γενικής χρήσεως ή και προσαρμοσμένες πλακέτες που εφάπτονται συγκεκριμένα στην υλοποίηση του εκάστοτε IoT. Παρακάτω θα αναλυθούν δύο δημοφιλή και αξιόπιστα είδη πλακετών, το Raspberry και το Arduino.

2.2.1.1 Raspberry Pi



Σχήμα 2.5: Πλακέτα Raspberry Pi

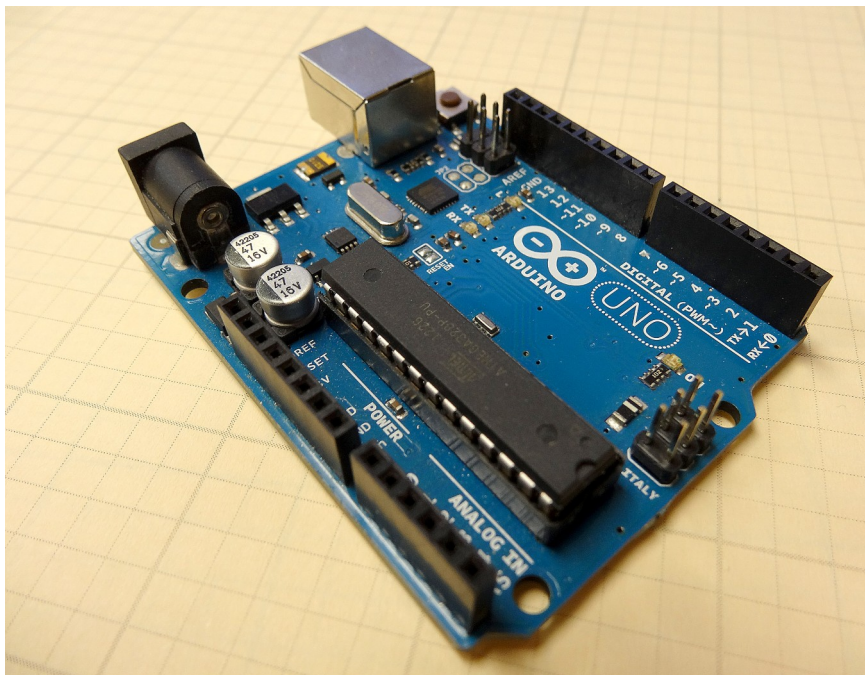
Πηγή: <http://www.expertreviews.co.uk/raspberry-pi-foundation/raspberry-pi-2>

Το Raspberry Pi αναπτύχθηκε από το ίδρυμα Raspberry Foundation στο Ηνωμένο Βασίλειο, το οποίο είναι μια σειρά από μικρούς υπολογιστές με βάση μια πλακέτα. Το αρχικό μοντέλο έγινε δημοφιλές

καθώς άρχισε να χρησιμοποιείται και στη ρομποτική και απέκτησε εκθετική δημοτικότητα. Τα περισσότερα Pis δημιουργούνται στο εργοστάσιο της Sony στην Ουαλία, αλλά ορισμένα κατασκευάζονται και στην Ιαπωνία και την Κίνα.

Πιστεύεται ότι είναι πιο περίπλοκο από το Arduino. Χρησιμοποιείται όταν οι απαιτήσεις είναι αυτές ενός ολοκληρωμένου υπολογιστή, καθώς μπορεί να οδηγήσει ένα πιο περίπλοκο ρομπότ ικανό να εκτελεί εργασίες ταυτόχρονα. Η πρωταρχική του χρήση ήταν να αναπτύξει την κατανόηση μεταξύ των μαθητών που σχετίζονται με την επιστήμη των υπολογιστών και τις τεχνολογίες της, ειδικά σε εκείνους που ανήκουν σε αναπτυσσόμενες χώρες.

2.2.1.2 Arduino

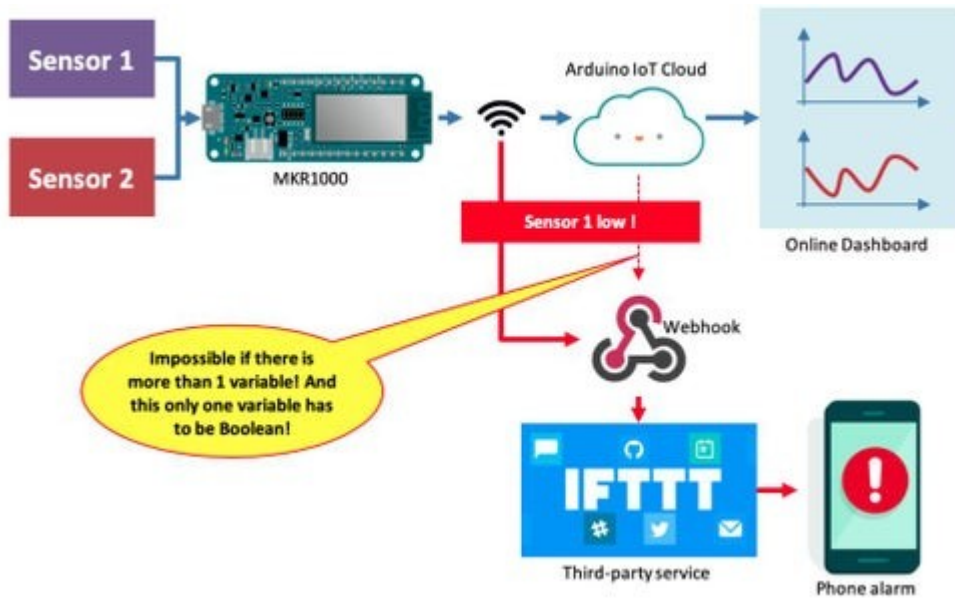


Σχήμα 2.6: Πλακέτα Arduino

Πηγή:<https://toptechboy.com/arduino-tutorial-1-getting-started-with-the-arduino-for-beginners/arduino/>

Το Arduino είναι πολύ απλό και εύκολο στη χρήση. Οι επαναλαμβανόμενες εργασίες, όπως το άνοιγμα ή το κλείσιμο μιας γκαραζόπορτας, η καταγραφή της εξωτερικής θερμοκρασίας και η αναφορά των δεδομένων σε κάποιο εργαλείο ή σε έναν ιστότοπο που οδηγείται από ένα απλό ρομπότ, μπορούν να γίνουν δυνατές με το Arduino. Αυτά τα κιτ χρησιμοποιούνται για τον έλεγχο και την αίσθηση των αντικειμένων του πραγματικού φυσικού και ψηφιακού κόσμου. Οι πλακέτες Arduino διατίθενται με τη μορφή DIY(Do-It-Yourself) κιτ ή στο εμπόριο σε προσαρμοσμένη μορφή.

Arduino IoT Cloud



Σχήμα 2.7: Αναπαράσταση Arduino IoT Cloud

Πηγή: <https://www.instructables.com/id/MKR1000-Arduino-IoT-Cloud-IFTT-Phone-Call/>



Το Arduino IoT Cloud είναι ο τρόπος με τον οποίο η Arduino εκδημοκρατίζει την ανάπτυξη του IoT, διευκολύνοντας όλους να δημιουργήσουν εφαρμογές για το διαδίκτυο των πραγμάτων. Παρέχει τη δυνατότητα στις συσκευές που βασίζονται στο IoT να ανταλλάσσουν δεδομένα μεταξύ τους και με το cloud όπου μπορεί να γίνει περαιτέρω επεξεργασία και να χρησιμοποιηθούν τα δεδομένα για την επίλυση συγκεκριμένων προβλημάτων. Η πλατφόρμα επιτρέπει την επικοινωνία μέσω πλήθους πρωτοκόλλων όπως: HTTP REST API, MQTT, εργαλεία γραμμής εντολών, Javascript και Websockets. Διαθέτει την τυπική "εύκολη στη χρήση" φύση του Arduino και διαθέτει επίσης εργαλεία που το καθιστούν ικανό να παράγει αυτόματα το σκίτσο/κώδικα για τη συσκευή, γεγονός που συμβάλλει στη μείωση του χρόνου ανάπτυξης από ώρες σε λεπτά. Στο πλαίσιο της τυπικής του φύσης ανοικτού κώδικα και της υποστήριξης άλλων κλώνων, το Arduino IoT Cloud επιτρέπει τη σύνδεση άλλων συσκευών και πλακετών που βασίζονται σε Linux με την πλατφόρμα [8].

Η ασπίδα Arduino MKR ENV επιτρέπει στη σειρά πλακετών MKR Arduino να λαμβάνει περιβαλλοντικά δεδομένα μέσω μιας πληθώρας αισθητήρων που είναι ενσωματωμένοι στην πλακέτα. Οι ενσωματωμένοι αισθητήρες περιλαμβάνουν αισθητήρες τελευταίας γενιάς για τη μέτρηση:

-Ατμοσφαιρικής πίεσης

- Θερμοκρασίας και υγρασίας
- Έντασης υπεριώδους ακτινοβολίας UVA
- Έντασης υπεριώδους ακτινοβολίας UVB
- Δείκτη υπεριώδους ακτινοβολίας (υπολογισμένος)
- Έντασης φωτός (σε LUX)

2.2.1.3 - Διαφορές μεταξύ Raspberry Pi και Arduino

	Raspberry Pi 3	Arduino Uno Rev3
		
Advantages	<ul style="list-style-type: none"> • Stronger and quicker processor, multitasking available • Built in Ethernet port, Wi-Fi and Bluetooth capability • OS can be switched easily • Audio output, Camera port, USB ports, HDMI output all included • Great to start learning to code with its helpful learning programs already installed • Great for projects that need to connect online and have multiple activities going on at the same time 	<ul style="list-style-type: none"> • Easier to connect to analog sensors, motors and other electronic components • Variety of Shields that can add functionality • Long set-up not needed, just plug in and code will run • Price is cheaper (around \$20) and will not need much cables (standard A/B USB) • Great for projects that need to quickly get data from sensors and do one activity from that data
Disadvantages	<ul style="list-style-type: none"> • Long set up and will need extra components when first starting (HDMI cable, monitor, keyboard and mouse) • Might need to install programs to get simple actions going • Can be more expensive (around \$35, not including SD cards, cables, keyboards/mouse) 	<ul style="list-style-type: none"> • Can run one code at a time, so can't multitask activities, slower speed • No Internet connectivity right out the box (can add with Shield) • Bigger learning curve since it's C/C++ language and will need to get outside sources to learn

Σχήμα 2.8: Σύγκριση Arduino με Raspberry Pi

Πηγή: <https://www.arrow.com/en/research-and-events/articles/comparing-arduino-uno-and-raspberry-pi-3>

Το Raspberry Pi είναι ένας μίνι υπολογιστής γενικής χρήσης που συνήθως διαθέτει λειτουργικό σύστημα βασισμένο στο Linux και έχει τη δυνατότητα πολλαπλών εργασιών. Μπορεί να χρησιμοποιηθεί για την εκτέλεση εντατικών υπολογιστικών και μαθηματικών λειτουργιών, όπως για το Bitcoin ή τις τεχνολογίες κρυπτογράφησης. Από την άλλη πλευρά, το Arduino είναι μια εταιρεία λογισμικού και υλικού υπολογιστών ανοικτού κώδικα, κοινοτήτων χρηστών και έργων που κατασκευάζει και σχεδιάζει κιτ μικροελεγκτών μίας πλακέτας διαδραστικών αντικειμένων και κατασκευής ψηφιακών συσκευών. Ένα Arduino είναι πρωτίστως μια μητρική πλακέτα μικροελεγκτή, η οποία είναι ένας απλός υπολογιστής που χειρίζεται μόνο ένα πρόγραμμα κάθε φορά, επαναλαμβανόμενα.

Το Arduino είναι ένας μικροελεγκτής ενώ το Raspberry Pi είναι ένας μικροεπεξεργαστής. Παρόλο που το Raspberry Pi είναι πιο ισχυρό, απαιτεί λειτουργικό σύστημα, περισσότερη ισχύ και είναι επίσης πιο ακριβό από ένα Arduino. Ως εκ τούτου, δεν χρησιμοποιούνται για τα ίδια έργα [9].

Το πλεονέκτημα του Raspberry περιλαμβάνει την εύκολη σύνδεση με το διαδίκτυο, υπάρχει πλήρης υποστήριξη από την κοινότητα Linux και η επιλογή της γλώσσας προγραμματισμού μπορεί να είναι τεράστια, ενώ, στην περίπτωση του Arduino, τα πλεονεκτήματα είναι ότι είναι εύκολο να ξεκινήσει κανείς, μπορεί να χρησιμοποιηθεί για την κατασκευή εφαρμογών πραγματικού χρόνου που βασίζονται τόσο στο υλικό όσο και στο λογισμικό, δεν απαιτούνται βαθιές γνώσεις προγραμματισμού και είναι εύκολο να επεκταθεί και να έχει τεράστιες βιβλιοθήκες και σύνολα ασπίδων από τους χρήστες [10].

Τα μειονεκτήματα του Raspberry περιλαμβάνουν τη μη υποστήριξη υλικού πραγματικού χρόνου, η διασύνδεση μπορεί να καθυστερήσει εάν η CPU είναι απασχολημένη, δεν μπορούν να οδηγηθούν επαγωγικά φορτία, δεν υπάρχει αναλογικός σε ψηφιακό μετατροπέας και ο σχεδιασμός υλικού δεν είναι ανοικτού κώδικα, ενώ τα μειονεκτήματα για το Arduino περιλαμβάνουν ότι οι γλώσσες που πρέπει να χρησιμοποιηθούν είναι μόνο Arduino ή C ή C++ στις οποίες είναι δύσκολο να κωδικοποιηθεί, δεν είναι αρκετά ισχυρό εργαλείο όταν το φέρουμε σε σύγκριση με το Raspberry Pi και η σύνδεση με το διαδίκτυο είναι αρκετά δύσκολη.

2.2.2 Βάση Δεδομένων

2.2.2.1 Προκλήσεις βάσης δεδομένων

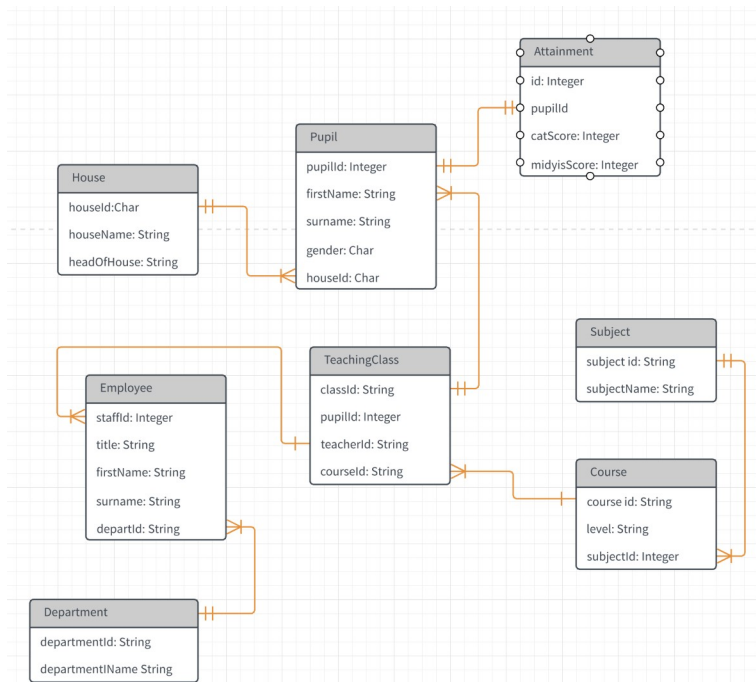
Οι σημερινές μεγάλες επιχειρησιακές βάσεις δεδομένων υποστηρίζουν συχνά πολύ σύνθετα ερωτήματα και αναμένεται να παρέχουν σχεδόν άμεσες απαντήσεις σε αυτά τα ερωτήματα. Ως αποτέλεσμα, οι διαχειριστές βάσεων δεδομένων καλούνται συνεχώς να χρησιμοποιούν μια μεγάλη ποικιλία μεθόδων για να συμβάλλουν στη βελτίωση της απόδοσης. Ορισμένες κοινές προκλήσεις που αντιμετωπίζουν περιλαμβάνουν:

- Απορρόφηση σημαντικών αυξήσεων στον όγκο δεδομένων: Η έκρηξη των δεδομένων που έρχονται από αισθητήρες, συνδεδεμένες μηχανές και δεκάδες άλλες πηγές κάνει τους διαχειριστές βάσεων δεδομένων να αγωνίζονται να διαχειριστούν και να οργανώσουν αποτελεσματικά τα δεδομένα των εταιρειών τους.
- Διασφάλιση της ασφάλειας των δεδομένων: Οι παραβιάσεις δεδομένων συμβαίνουν παντού αυτές τις μέρες και οι χάκερ γίνονται όλο και πιο εφευρετικοί. Είναι πιο σημαντικό από ποτέ να διασφαλιστεί ότι τα δεδομένα είναι ασφαλή αλλά και εύκολα προσβάσιμα από τους χρήστες.
- Να συμβαδίζουν με τη ζήτηση: Στο σημερινό ταχέως εξελισσόμενο επιχειρηματικό περιβάλλον, οι εταιρείες χρειάζονται πρόσβαση σε πραγματικό χρόνο στα δεδομένα τους για την υποστήριξη της έγκαιρης λήψης αποφάσεων και την αξιοποίηση νέων ευκαιριών.
- Διαχείριση και συντήρηση της βάσης δεδομένων και της υποδομής: Οι διαχειριστές βάσεων δεδομένων πρέπει να παρακολουθούν συνεχώς τη βάση δεδομένων για προβλήματα και να εκτελούν προληπτική συντήρηση, καθώς και να εφαρμόζουν αναβαθμίσεις και διορθώσεις λογισμικού. Καθώς οι βάσεις δεδομένων γίνονται όλο και πιο πολύπλοκες και ο όγκος των δεδομένων αυξάνεται, οι εταιρείες έρχονται αντιμέτωπες με το κόστος πρόσληψης πρόσθετων ταλέντων για την παρακολούθηση και τη ρύθμιση των βάσεων δεδομένων τους.
- Άρση των περιορισμών στην επεκτασιμότητα: Μια επιχείρηση πρέπει να αναπτύσσεται αν πρόκειται να επιβιώσει, και η διαχείριση των δεδομένων της πρέπει να αναπτύσσεται μαζί της. Αλλά είναι πολύ δύσκολο για τους διαχειριστές βάσεων δεδομένων να προβλέψουν πόση χωρητικότητα θα χρειαστεί η εταιρεία, ιδίως με τις τοπικές βάσεις δεδομένων.

2.2.2.2 Τύποι βάσεων δεδομένων

Υπάρχουν πολλοί διαφορετικοί τύποι βάσεων δεδομένων. Η καλύτερη βάση δεδομένων για έναν συγκεκριμένο οργανισμό εξαρτάται από τον τρόπο με τον οποίο ο οργανισμός σκοπεύει να χρησιμοποιήσει τα δεδομένα.

Σχεσιακές βάσεις δεδομένων



Σχήμα 2.9: Παράδειγμα οργάνωσης σχεσιακής βάσης δεδομένων

Πηγή: <https://stackoverflow.com/questions/51542341/relational-database-design-query>

Οι σχεσιακές βάσεις δεδομένων κυριάρχησαν τη δεκαετία του 1980. Τα στοιχεία σε μια σχεσιακή βάση δεδομένων οργανώνονται ως σύνολο πινάκων με στήλες και γραμμές. Η τεχνολογία σχεσιακών βάσεων δεδομένων παρέχει τον πιο αποτελεσματικό και ευέλικτο τρόπο πρόσβασης σε δομημένες πληροφορίες.

Ένα σχεσιακό μοντέλο οργανώνει τα δεδομένα σε έναν ή περισσότερους πίνακες (ή "σχέσεις") με στήλες και γραμμές, με ένα μοναδικό κλειδί που προσδιορίζει κάθε γραμμή. Οι σειρές ονομάζονται επίσης εγγραφές ή πλειάδες. Οι στήλες ονομάζονται επίσης χαρακτηριστικά. Γενικά, κάθε πίνακας/σχέση αντιπροσωπεύει έναν "τύπο οντότητας" (όπως ο πελάτης ή το προϊόν). Οι γραμμές αντιπροσωπεύουν περιπτώσεις αυτού του τύπου οντότητας (όπως "Lee" ή "καρέκλα") και οι στήλες αντιπροσωπεύουν τιμές που αποδίδονται σε αυτή την περίπτωση (όπως διεύθυνση ή τιμή).

Για παράδειγμα, κάθε γραμμή ενός πίνακα τάξης αντιστοιχεί σε μια τάξη και μια τάξη αντιστοιχεί σε πολλούς μαθητές, οπότε η σχέση μεταξύ του πίνακα τάξης και του πίνακα μαθητών είναι "ένα προς πολλά"

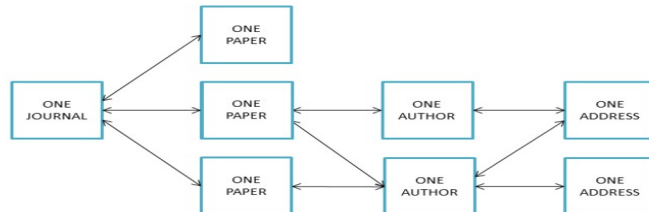
Κάθε γραμμή σε έναν πίνακα έχει το δικό της μοναδικό κλειδί. Οι γραμμές ενός πίνακα μπορούν να συνδεθούν με γραμμές άλλων πινάκων προσθέτοντας μια στήλη για το μοναδικό κλειδί της συνδεδεμένης γραμμής (τέτοιες στήλες είναι γνωστές ως ξένα κλειδιά).

Μέρος αυτής της επεξεργασίας περιλαμβάνει τη συνεπή δυνατότητα επιλογής ή τροποποίησης μίας και μόνο μίας γραμμής σε έναν πίνακα. Ως εκ τούτου, οι περισσότερες φυσικές υλοποιήσεις έχουν ένα μοναδικό πρωτεύον κλειδί (primary key - PK) για κάθε γραμμή σε έναν πίνακα. Όταν γράφεται μια νέα γραμμή στον πίνακα, δημιουργείται μια νέα μοναδική τιμή για το πρωτεύον κλειδί- αυτό είναι το κλειδί που χρησιμοποιεί το σύστημα κυρίως για την πρόσβαση στον πίνακα. Η απόδοση του συστήματος βελτιστοποιείται για PKs. Άλλα, πιο φυσικά κλειδιά μπορούν επίσης να προσδιοριστούν και να οριστούν ως εναλλακτικά κλειδιά (alternate key - AK). Συχνά απαιτούνται πολλές στήλες για να σχηματιστεί ένα AK (αυτός είναι ένας λόγος για τον οποίο μια μόνο στήλη ακέραίου αριθμού γίνεται συνήθως το PK). Τόσο τα PKs όσο και τα AKs έχουν την ικανότητα να προσδιορίζουν μοναδικά μια γραμμή μέσα σε έναν πίνακα. Πρόσθετη τεχνολογία μπορεί να εφαρμοστεί για την εξασφάλιση ενός μοναδικού αναγνωριστικού σε όλο τον κόσμο, ενός παγκόσμιου μοναδικού αναγνωριστικού, όταν υπάρχουν ευρύτερες απαιτήσεις του συστήματος.

Τα πρωτεύοντα κλειδιά εντός μιας βάσης δεδομένων χρησιμοποιούνται για τον καθορισμό των σχέσεων μεταξύ των πινάκων. Όταν ένα PK μεταφέρεται σε έναν άλλο πίνακα, γίνεται ξένο κλειδί στον άλλο πίνακα. Όταν κάθε κελί μπορεί να περιέχει μόνο μία τιμή και το PK μεταναστεύει σε έναν κανονικό πίνακα οντοτήτων, αυτό το πρότυπο σχεδίασης μπορεί να αντιπροσωπεύει είτε μια σχέση ένα προς ένα είτε μια σχέση ένα προς πολλά. Τα περισσότερα σχέδια σχεσιακών βάσεων δεδομένων επιλύουν τις σχέσεις πολλά-προς-πολλά δημιουργώντας έναν πρόσθετο πίνακα που περιέχει τα PKs και από τους δύο άλλους πίνακες οντοτήτων - η σχέση γίνεται οντότητα- ο πίνακας επίλυσης ονομάζεται στη συνέχεια κατάλληλα και τα δύο FKs συνδυάζονται για να σχηματίσουν ένα PK. Η μετανάστευση των PKs σε άλλους πίνακες είναι ο δεύτερος σημαντικός λόγος για τον οποίο χρησιμοποιούνται συνήθως ακέραιοι αριθμοί που έχουν οριστεί από το σύστημα ως PKs- συνήθως δεν υπάρχει ούτε αποτελεσματικότητα ούτε σαφήνεια στη μετανάστευση ενός συνόλου άλλων τύπων στηλών.

Relational database

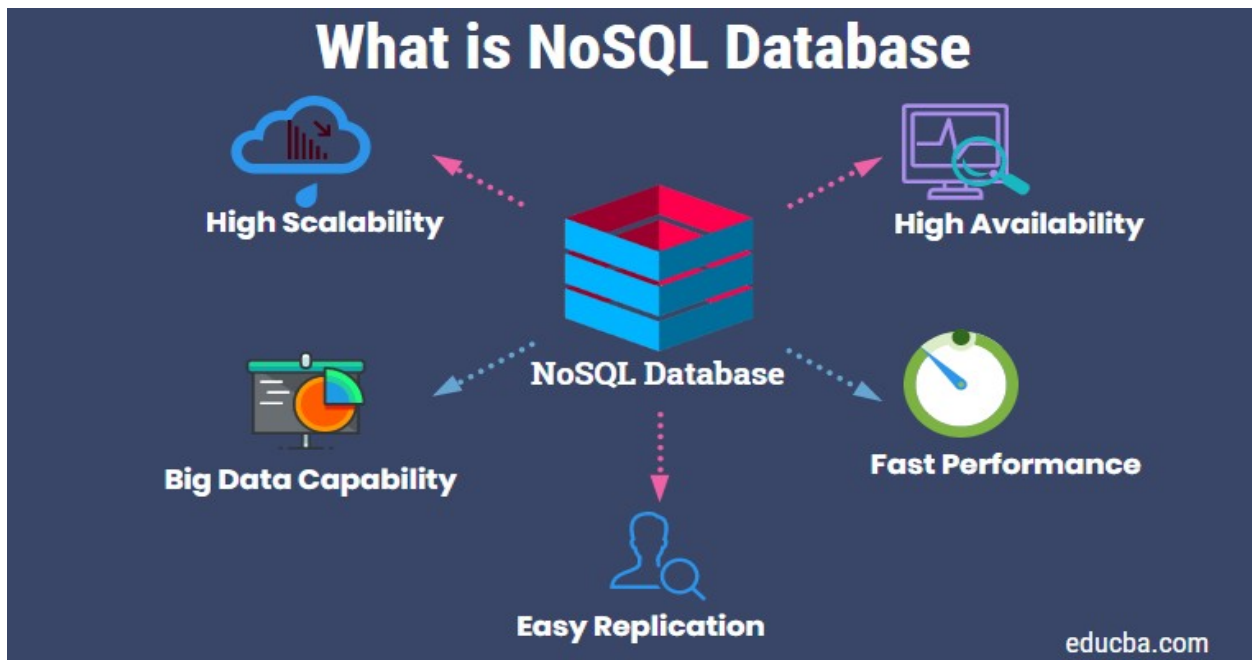
data in many tables linked by identifiers to core table(s)



Σχήμα 2.10: Βασική δομή σχεσιακής βάσης δεδομένων

Πηγή: <http://www.assignmentpoint.com/science/computer/object-relational-databases.html>

Βάσεις δεδομένων NoSQL



Σχήμα 2.11: Χαρακτηριστικά μη σχεσιακής βάσης δεδομένων

Πηγή: <https://www.educba.com/what-is-nosql-database/>

Μια NoSQL, ή μη σχεσιακή βάση δεδομένων, επιτρέπει την αποθήκευση και τον χειρισμό μη δομημένων και ημιδομημένων δεδομένων (σε αντίθεση με μια σχεσιακή βάση δεδομένων, η οποία ορίζει πώς πρέπει να συντίθενται όλα τα δεδομένα που εισάγονται στη βάση δεδομένων). Οι βάσεις δεδομένων NoSQL έγιναν δημοφιλείς καθώς οι διαδικτυακές εφαρμογές έγιναν πιο διαδεδομένες και πιο σύνθετες.

Οι βάσεις δεδομένων NoSQL προσφέρουν πολλά πλεονεκτήματα σε σχέση με τις σχεσιακές βάσεις δεδομένων. Οι βάσεις δεδομένων NoSQL διαθέτουν ευέλικτα μοντέλα δεδομένων, κλιμακώνονται οριζόντια, έχουν απίστευτα γρήγορα ερωτήματα και είναι εύκολο να εργαστούν οι προγραμματιστές.

Μια βάση δεδομένων NoSQL (που αρχικά σημαίνει "μη SQL" ή "μη σχεσιακή") παρέχει έναν μηχανισμό αποθήκευσης και ανάκτησης δεδομένων που μοντελοποιούνται με μέσα διαφορετικά από τις σχέσεις σε μορφή πίνακα που χρησιμοποιούνται στις σχεσιακές βάσεις δεδομένων. Οι βάσεις δεδομένων NoSQL χρησιμοποιούνται όλο και περισσότερο σε εφαρμογές μεγάλου όγκου δεδομένων και διαδικτυακών εφαρμογών πραγματικού χρόνου[5]. Τα συστήματα NoSQL αποκαλούνται επίσης μερικές φορές Not only SQL για να τονιστεί ότι μπορούν να υποστηρίξουν γλώσσες ερωτημάτων που μοιάζουν με SQL ή να βρίσκονται δίπλα σε βάσεις δεδομένων SQL σε πολυγλωσσικές αρχιτεκτονικές.

Τα κίνητρα για αυτή την προσέγγιση περιλαμβάνουν την απλότητα του σχεδιασμού, την απλούστερη "οριζόντια" κλιμάκωση σε συστάδες μηχανών (η οποία αποτελεί πρόβλημα για τις σχεσιακές βάσεις δεδομένων), τον λεπτότερο έλεγχο της διαθεσιμότητας και τον περιορισμό της αναντιστοιχίας αντικειμενικής-σχεσιακής εμπλοκής. Οι δομές δεδομένων που χρησιμοποιούνται από τις NoSQL βάσεις δεδομένων (π.χ. ζεύγος κλειδιών-τιμών, ευρεία στήλη, γράφος ή έγγραφο) είναι διαφορετικές από αυτές που χρησιμοποιούνται εξ ορισμού στις σχεσιακές βάσεις δεδομένων, καθιστώντας ορισμένες λειτουργίες ταχύτερες στις NoSQL. Η ιδιαίτερη καταλληλότητα μιας δεδομένης βάσης δεδομένων NoSQL εξαρτάται από το πρόβλημα που πρέπει να επιλύσει. Μερικές φορές οι δομές δεδομένων που χρησιμοποιούνται από τις βάσεις δεδομένων NoSQL θεωρούνται επίσης ως "πιο ευέλικτες" από τους πίνακες των σχεσιακών βάσεων δεδομένων.

Πολλές αποθήκες NoSQL συμβιβάζουν τη συνέπεια υπέρ της διαθεσιμότητας, της ανοχής κατατμήσεων και της ταχύτητας. Τα εμπόδια στην ευρύτερη υιοθέτηση των NoSQL καταστημάτων περιλαμβάνουν τη χρήση γλωσσών ερωτημάτων χαμηλού επιπέδου (αντί της SQL, για παράδειγμα), την έλλειψη δυνατότητας εκτέλεσης ad hoc ενώσεων μεταξύ πινάκων, την έλλειψη τυποποιημένων διεπαφών και τις τεράστιες προηγούμενες επενδύσεις σε υπάρχουσες σχεσιακές βάσεις δεδομένων.

Αντ' αυτού, οι περισσότερες βάσεις δεδομένων NoSQL προσφέρουν μια έννοια "ενδεχόμενης συνέπειας", σύμφωνα με την οποία οι αλλαγές στη βάση δεδομένων διαδίδονται σε όλους τους κόμβους "τελικά" (συνήθως εντός χιλιοστών του δευτερολέπτου), οπότε τα ερωτήματα για δεδομένα μπορεί να μην επιστρέφουν αμέσως ενημερωμένα δεδομένα ή να οδηγούν στην ανάγνωση δεδομένων που δεν είναι ακριβή, ένα πρόβλημα γνωστό ως "παλαιές αναγνώσεις". Επιπλέον, ορισμένα συστήματα NoSQL ενδέχεται να παρουσιάζουν χαμένες εγγραφές και άλλες μορφές απώλειας δεδομένων. Ορισμένα συστήματα NoSQL παρέχουν έννοιες όπως η καταγραφή με προγενέστερη εγγραφή για την αποφυγή απώλειας δεδομένων. Για την κατανομημένη επεξεργασία συναλλαγών σε πολλαπλές βάσεις

δεδομένων, η συνέπεια των δεδομένων αποτελεί ακόμη μεγαλύτερη πρόκληση που είναι δύσκολη τόσο για τις NoSQL όσο και για τις σχεσιακές βάσεις δεδομένων. Λίγα συστήματα διατηρούν τόσο τις συναλλαγές ACID όσο και τα πρότυπα X/Open XA για την κατανεμημένη επεξεργασία συναλλαγών. Οι διαδραστικές σχεσιακές βάσεις δεδομένων μοιράζονται τεχνικές ανάλυσης της συμμορφούμενης αναπαραγωγής ως κοινό χαρακτηριστικό. Οι περιορισμοί εντός του περιβάλλοντος διεπαφής ξεπερνιούνται με τη χρήση πρωτοκόλλων σημασιολογικής εικονικοποίησης, έτσι ώστε οι υπηρεσίες NoSQL να είναι προσβάσιμες στα περισσότερα λειτουργικά συστήματα.

Τύποι βάσεων δεδομένων NoSQL

Αποθήκευση εγγράφων

Η κεντρική έννοια μιας αποθήκης εγγράφων είναι αυτή του "εγγράφου". Αν και οι λεπτομέρειες αυτού του ορισμού διαφέρουν μεταξύ των βάσεων δεδομένων προσανατολισμένων στα έγγραφα, όλες υποθέτουν ότι τα έγγραφα ενθυλακώνουν και κωδικοποιούν δεδομένα (ή πληροφορίες) σε κάποιες τυποποιημένες μορφές ή κωδικοποιήσεις. Οι χρησιμοποιούμενες κωδικοποιήσεις περιλαμβάνουν XML, YAML και JSON και δυαδικές μορφές όπως BSON. Τα έγγραφα απευθύνονται στη βάση δεδομένων μέσω ενός μοναδικού κλειδιού που αντιπροσωπεύει το συγκεκριμένο έγγραφο. Ένα άλλο καθοριστικό χαρακτηριστικό μιας βάσης δεδομένων προσανατολισμένης στα έγγραφα είναι ένα API ή μια γλώσσα ερωτημάτων για την ανάκτηση εγγράφων με βάση το περιεχόμενό τους.

Διαφορετικές υλοποιήσεις προσφέρουν διαφορετικούς τρόπους οργάνωσης ή/και ομαδοποίησης εγγράφων:

- Συλλογές
- Ετικέτες
- Μη ορατά μεταδεδομένα
- Ιεραρχίες καταλόγου

Σε σύγκριση με τις σχεσιακές βάσεις δεδομένων, οι συλλογές θα μπορούσαν να θεωρηθούν ανάλογες με τους πίνακες και τα έγγραφα ανάλογες με τις εγγραφές. Όμως είναι διαφορετικές: κάθε εγγραφή σε έναν πίνακα έχει την ίδια ακολουθία πεδίων, ενώ τα έγγραφα σε μια συλλογή μπορεί να έχουν πεδία που είναι εντελώς διαφορετικά.

Αποθήκευση κλειδιού-τιμής

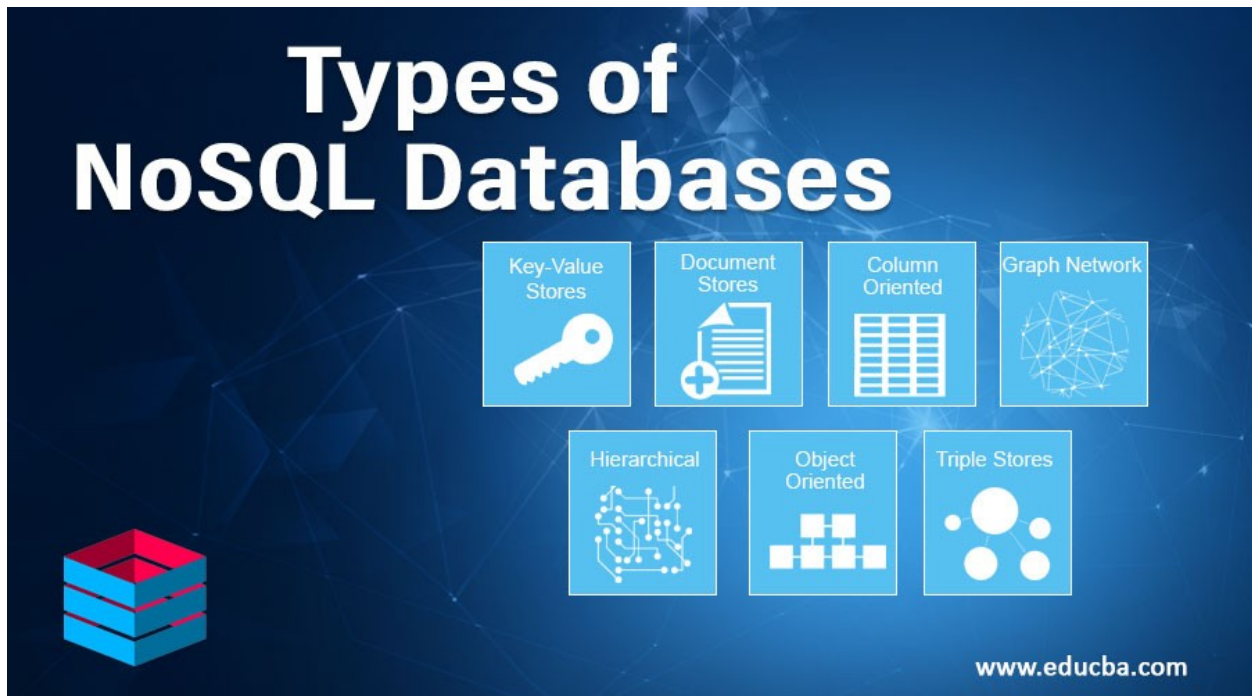
Τα καταστήματα κλειδί-τιμή (key-value - KV) χρησιμοποιούν τον συσχετιστικό πίνακα (που ονομάζεται επίσης χάρτης ή λεξικό) ως το θεμελιώδες μοντέλο δεδομένων τους. Σε αυτό το μοντέλο, τα δεδομένα αναπαρίστανται ως μια συλλογή από ζεύγη κλειδιού-τιμής, έτσι ώστε κάθε πιθανό κλειδί να εμφανίζεται το πολύ μία φορά στη συλλογή.

Το μοντέλο κλειδιού-τιμής είναι ένα από τα απλούστερα μη τριτογενή μοντέλα δεδομένων, και πλουσιότερα μοντέλα δεδομένων υλοποιούνται συχνά ως επέκταση αυτού. Το μοντέλο κλειδιού-τιμής μπορεί να επεκταθεί σε ένα μοντέλο διακριτής διάταξης που διατηρεί τα κλειδιά σε λεξιγραφική σειρά. Αυτή η επέκταση είναι υπολογιστικά ισχυρή, καθώς μπορεί να ανακτήσει αποτελεσματικά επιλεκτικές σειρές κλειδιών.

Οι αποθήκες κλειδιών-τιμών μπορούν να χρησιμοποιούν μοντέλα συνέπειας που κυμαίνονται από την ενδεχόμενη συνέπεια έως τη σειριοποιησιμότητα. Ορισμένες βάσεις δεδομένων υποστηρίζουν την ταξινόμηση των κλειδιών. Υπάρχουν διάφορες υλοποιήσεις υλικού, και ορισμένοι χρήστες αποθηκεύουν τα δεδομένα στη μνήμη (RAM), ενώ άλλοι σε δίσκους στερεάς κατάστασης (SSD) ή σε περιστρεφόμενους δίσκους (γνωστούς και ως σκληρό δίσκο (HDD)).

Γράφημα

Οι βάσεις δεδομένων γράφων έχουν σχεδιαστεί για δεδομένα των οποίων οι σχέσεις αναπαρίστανται καλά ως γράφος που αποτελείται από στοιχεία που συνδέονται με πεπερασμένο αριθμό σχέσεων. Παραδείγματα δεδομένων περιλαμβάνουν κοινωνικές σχέσεις, συνδέσεις δημόσιων μεταφορών, οδικούς χάρτες, τοπολογίες δικτύων κ.λπ.



Σχήμα 2.12: Τύποι μη σχεσιακών βάσεων δεδομένων

Πηγή: <https://www.educba.com/what-is-nosql-database/>

Χειρισμός σχεσιακών δεδομένων

Δεδομένου ότι οι περισσότερες NoSQL βάσεις δεδομένων δεν έχουν τη δυνατότητα σύνδεσης σε ερωτήματα, το σχήμα της βάσης δεδομένων πρέπει γενικά να σχεδιαστεί διαφορετικά. Υπάρχουν τρεις κύριες τεχνικές για το χειρισμό σχεσιακών δεδομένων σε μια βάση δεδομένων NoSQL. (Ανατρέξτε στον πίνακα Join και ACID Support για τις βάσεις δεδομένων NoSQL που υποστηρίζουν joins).

Ενσωμάτωση δεδομένων

Με τις βάσεις δεδομένων εγγράφων όπως η MongoDB είναι σύνηθες να τοποθετούνται περισσότερα δεδομένα σε μικρότερο αριθμό συλλογών. Για παράδειγμα, σε μια εφαρμογή ιστολογίου, μπορεί κανείς να επιλέξει να αποθηκεύσει τα σχόλια μέσα στο έγγραφο της ανάρτησης του ιστολογίου, έτσι ώστε με μια απλή ανάκτηση να πάρει όλα τα σχόλια. Έτσι σε αυτή την προσέγγιση ένα μόνο έγγραφο περιέχει όλα τα δεδομένα που χρειάζονται για μια συγκεκριμένη εργασία.

Προσωρινή αποθήκευση, αντιγραφή και μη κανονικοποιημένα δεδομένα

Αντί να αποθηκεύονται μόνο ξένα κλειδιά, είναι σύνηθες να αποθηκεύονται οι πραγματικές ξένες τιμές μαζί με τα δεδομένα του μοντέλου. Για παράδειγμα, κάθε σχόλιο ιστολογίου μπορεί να περιλαμβάνει το όνομα χρήστη εκτός από ένα αναγνωριστικό χρήστη, παρέχοντας έτσι εύκολη πρόσβαση στο όνομα χρήστη χωρίς να απαιτείται άλλη αναζήτηση. Όταν όμως το όνομα χρήστη αλλάξει, αυτό θα πρέπει τώρα να αλλάξει σε πολλά σημεία της βάσης δεδομένων. Έτσι, αυτή η προσέγγιση λειτουργεί καλύτερα όταν οι αναγνώσεις είναι πολύ πιο συχνές από τις εγγραφές [11].

Πολλαπλά ερωτήματα

Αντί να ανακτηθούν όλα τα δεδομένα με ένα ερώτημα, είναι σύνηθες να γίνονται πολλά ερωτήματα για να ληφθούν τα επιθυμητά δεδομένα. Τα ερωτήματα NoSQL είναι συχνά ταχύτερα από τα παραδοσιακά ερωτήματα SQL, οπότε το κόστος των πρόσθετων ερωτημάτων μπορεί να είναι αποδεκτό. Εάν θα ήταν απαραίτητος υπερβολικός αριθμός ερωτημάτων, μια από τις άλλες δύο προσεγγίσεις είναι πιο κατάλληλη.

Παρακάτω ακολουθεί ένας πίνακας αντιστοίχισης τύπων βάσεων δεδομένων NoSQL με χαρακτηριστικά παραδείγματα της κάθε κατηγορίας.

Κρυφή μνήμη κλειδιών-τιμών	Apache Ignite, Couchbase, Coherence, eXtreme Scale, Hazelcast, Infinispan, Memcached, Redis, Velocity
Αποθήκευση κλειδιού-τιμής	Azure Cosmos DB, ArangoDB, Amazon DynamoDB, Aerospike, Couchbase, ScyllaDB
Αποθήκη Tuple	Apache River, GigaSpaces, Tarantool, TIBCO ActiveSpaces, OpenLink Virtuoso
Triplestore	AllegroGraph, MarkLogic, Ontotext-OWLIM, Oracle NoSQL database, Profium Sense, Virtuoso Universal Server
Βάση δεδομένων αντικειμένων	Objectivity/DB, Perst, ZopeDB, db4o, GemStone/S, InterSystems Caché, JADE, ObjectDatabase++, ObjectDB, ObjectStore, ODABA, Realm, OpenLink Virtuoso, Versant Object Database, ZODB
Αποθήκευση εγγράφων	Azure Cosmos DB, ArangoDB, BaseX, Clusterpoint, Couchbase, CouchDB, DocumentDB, eXist-db, IBM Domino, MarkLogic, MongoDB, RavenDB, Qizx, RethinkDB, Elasticsearch, OrientDB
Αποθήκευση ευρείας στήλης	Azure Cosmos DB, Amazon DynamoDB, Bigtable, Cassandra, Google Cloud Datastore, HBase, Hypertable, ScyllaDB
Βάση δεδομένων γραφημάτων	Azure Cosmos DB, AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso

Top 4 NoSQL Databases	MongoDB	Cassandra	Elasticsearch	Couchbase
Description	One of the most popular document stores	Wide-column store based on ideas of BigTable and DynamoDB	A modern search and analytics engine based on Apache Lucene	JSON-based document store derived from CouchDB with a Memcached-compatible interface
Database model	Document store	Wide Column store	Search engine	Document store
Developer	MongoDB, Inc.	Apache Software Foundation	Elastic	Couchbase, Inc.
Release	2009	2008	2010	2011
Language	C++	Java	Java	C, C++ and Erlang
Server-side scripts	JavaScript	No	Yes	View functions in JavaScript
Replication methods	Master-slave replication	Selectable replication factor	Yes	Master-master replication, Master-slave replication
Best use	If you need dynamic queries. If you prefer to define indexes, not map and reduced functions. If you need good performance on a big DB and when your data changes too much	When data you need to store doesn't fit on server, but requires friendly familiar interface to it	When you have objects with flexible fields, and you need "advanced search" functionality	Any application that requires low-latency data access, high concurrency support and high availability

Σχήμα 2.13: Σύγκριση των 4 πιο δημοφιλών μη σχεσιακών βάσεων δεδομένων

Πηγή: <https://dzone.com/articles/top-4-nosql-databases>

Υποστήριξη ACID και join

Μια βάση δεδομένων χαρακτηρίζεται ως υποστηρικτική των ιδιοτήτων ACID (Atomicity, Consistency, Isolation, Durability) ή των λειτουργιών join, εάν η τεκμηρίωση της βάσης δεδομένων υποστηρίζει αυτόν τον ισχυρισμό. Ωστόσο, αυτό δεν σημαίνει απαραίτητα ότι η ικανότητα υποστηρίζεται πλήρως με τρόπο παρόμοιο με τις περισσότερες βάσεις δεδομένων SQL.

Παρακάτω ακολουθεί ένας πίνακας κατηγοριοποίησης γνωστών βάσεων δεδομένων NoSQL με βάση τη διαθεσιμότητα ιδιοτήτων ACID ή join.

Βάση δεδομένων	ACID	Joins
Aerospike	Ναι	Όχι
Apache Ignite	Ναι	Ναι
ArangoDB	Ναι	Ναι

Couchbase	Ναι	Ναι
CouchDB	Ναι	Ναι
IBM Db2	Ναι	Ναι
InfinityDB	Ναι	Όχι
LMDB	Ναι	Όχι
MarkLogic	Ναι	Ναι
MongoDB	Ναι	Ναι
OrientDB	Ναι	Ναι

Βάσεις δεδομένων χρονοσειρών

Μια βάση δεδομένων χρονοσειρών (TSDB) είναι μια βάση δεδομένων βελτιστοποιημένη για δεδομένα χρονοσήμανσης ή χρονοσειρών. Τα δεδομένα χρονοσειρών είναι απλώς μετρήσεις ή γεγονότα που παρακολουθούνται, παρακολουθούνται, υποβαθμίζονται και συγκεντρώνονται με την πάροδο του χρόνου. Αυτά θα μπορούσαν να είναι μετρήσεις διακομιστή, παρακολούθηση της απόδοσης εφαρμογών, δεδομένα δικτύου, δεδομένα αισθητήρων, γεγονότα, κλικ, συναλλαγές σε μια αγορά και πολλοί άλλοι τύποι δεδομένων ανάλυσης [12].

Μια βάση δεδομένων χρονοσειρών έχει κατασκευαστεί ειδικά για το χειρισμό μετρικών και γεγονότων ή μετρήσεων που φέρουν χρονοσήμανση. Μια TSDB είναι βελτιστοποιημένη για τη μέτρηση αλλαγών με την πάροδο του χρόνου. Οι ιδιότητες που καθιστούν τα δεδομένα χρονοσειρών πολύ διαφορετικά από άλλους φόρτους εργασίας δεδομένων είναι η διαχείριση του κύκλου ζωής των δεδομένων, η σύνοψη και η σάρωση μεγάλου εύρους πολλών εγγραφών.

Τα σύνολα δεδομένων χρονολογικών σειρών είναι σχετικά μεγάλα και ομοιόμορφα σε σύγκριση με άλλα σύνολα δεδομένων - συνήθως αποτελούνται από μια χρονοσφραγίδα και συναφή δεδομένα. Τα σύνολα δεδομένων χρονολογικών σειρών μπορούν επίσης να έχουν λιγότερες σχέσεις μεταξύ καταχωρήσεων δεδομένων σε διαφορετικούς πίνακες και δεν απαιτούν απεριόριστη αποθήκευση καταχωρήσεων. Οι μοναδικές ιδιότητες των συνόλων δεδομένων χρονολογικών σειρών σημαίνουν ότι οι βάσεις δεδομένων χρονολογικών σειρών μπορούν να παρέχουν σημαντικές βελτιώσεις στον αποθηκευτικό χώρο και την απόδοση σε σχέση με τις βάσεις δεδομένων γενικού σκοπού. Για παράδειγμα, λόγω της ομοιομορφίας των δεδομένων χρονοσειρών, οι εξειδικευμένοι αλγόριθμοι

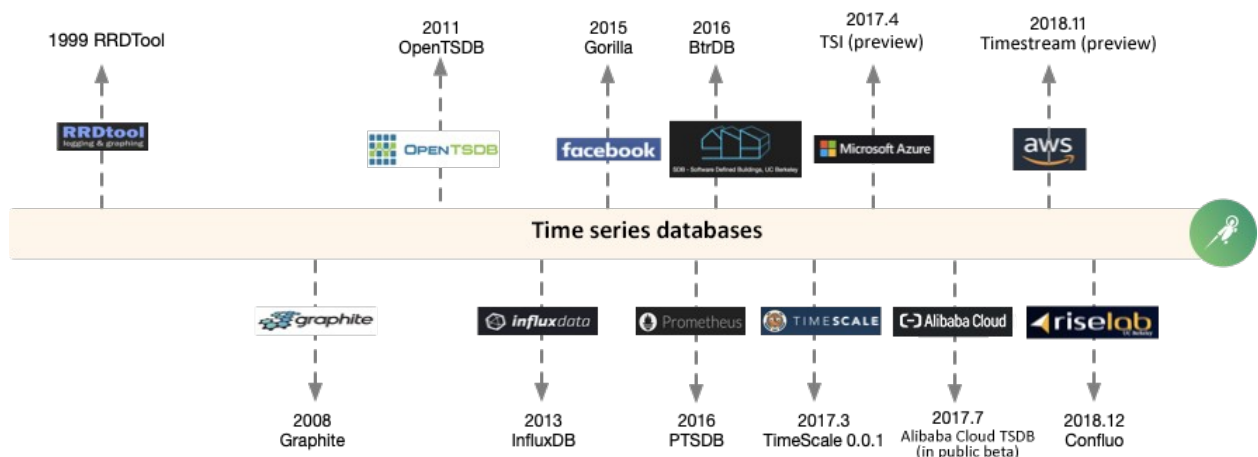
συμπίεσης μπορούν να παρέχουν βελτιώσεις σε σχέση με τους συνήθεις αλγόριθμους συμπίεσης που έχουν σχεδιαστεί για να λειτουργούν σε λιγότερο ομοιόμορφα δεδομένα. Οι βάσεις δεδομένων χρονοσειρών μπορούν επίσης να διαμορφωθούν ώστε να διαγράφουν τα παλιά δεδομένα, σε αντίθεση με τις συνήθεις βάσεις δεδομένων που έχουν σχεδιαστεί για να αποθηκεύουν δεδομένα επ' αόριστον. Οι ειδικοί δείκτες των βάσεων δεδομένων μπορούν επίσης να παρέχουν ώθηση στην απόδοση των ερωτημάτων.

Σε πολλές περιπτώσεις, τα αποθετήρια δεδομένων χρονοσειρών θα χρησιμοποιούν αλγόριθμους συμπίεσης για την αποτελεσματική διαχείριση των δεδομένων. Αν και είναι δυνατή η αποθήκευση δεδομένων χρονοσειρών σε πολλούς διαφορετικούς τύπους βάσεων δεδομένων, ο σχεδιασμός αυτών των συστημάτων με το χρόνο ως βασικό δείκτη είναι σαφώς διαφορετικός από τις σχεσιακές βάσεις δεδομένων που μειώνουν τις διακριτές σχέσεις μέσω αναφορικών μοντέλων.

Οι βάσεις δεδομένων χρονοσειρών δεν είναι καινούργιες, αλλά οι βάσεις δεδομένων χρονοσειρών πρώτης γενιάς επικεντρώνονταν κυρίως στην εξέταση χρηματοοικονομικών δεδομένων, στη μεταβλητότητα των χρηματιστηριακών συναλλαγών και στα συστήματα που κατασκευάστηκαν για την επίλυση των συναλλαγών. Όμως τα χρηματοοικονομικά δεδομένα δεν είναι πλέον σχεδόν η μόνη εφαρμογή των δεδομένων χρονοσειρών - στην πραγματικότητα, είναι μόνο μία από τις πολυάριθμες εφαρμογές σε διάφορους κλάδους. Οι θεμελιώδεις συνθήκες της πληροφορικής έχουν αλλάξει δραματικά την τελευταία δεκαετία. Τα πάντα έχουν γίνει διαμερισματοποιημένα (compartmentalized). Τα μονολιθικά mainframes έχουν εξαφανιστεί και έχουν αντικατασταθεί από serverless servers, microservers και containers [13].

Σήμερα, ό,τι μπορεί να γίνει εξάρτημα είναι εξάρτημα. Επιπλέον, γινόμαστε μάρτυρες της ενοργάνωσης κάθε διαθέσιμης επιφάνειας στον υλικό κόσμο - δρόμοι, αυτοκίνητα, εργοστάσια, δίκτυα ηλεκτρικής ενέργειας, καλύμματα πάγου, δορυφόροι, ρούχα, τηλέφωνα, μικροκύματα, δοχεία γάλακτος, πλανήτες, ανθρώπινα σώματα. Τα πάντα έχουν ή θα έχουν έναν αισθητήρα. Έτσι, τώρα, τα πάντα μέσα και έξω από την εταιρεία εκπέμπουν μια αδιάκοπη ροή μετρήσεων και γεγονότων ή δεδομένων χρονοσειράς.

Αυτό σημαίνει ότι οι υποκείμενες πλατφόρμες πρέπει να εξελιχθούν για να υποστηρίξουν αυτούς τους νέους φόρτους εργασίας - περισσότερα σημεία δεδομένων, περισσότερες πηγές δεδομένων, περισσότερη παρακολούθηση, περισσότεροι έλεγχοι. Αυτό που παρακολουθούμε και αυτό που απαιτούν οι καιροί είναι μια παραδειγματική αλλαγή στον τρόπο με τον οποίο προσεγγίζουμε την υποδομή δεδομένων και τον τρόπο με τον οποίο προσεγγίζουμε την κατασκευή, την παρακολούθηση, τον έλεγχο και τη διαχείριση των συστημάτων [14]. Αυτό που χρειαζόμαστε είναι μια αποδοτική, κλιμακούμενη, ειδικά κατασκευασμένη βάση δεδομένων χρονοσειρών.



Σχήμα 2.14: Οι βάσεις δεδομένων χρονοσειρών με την πάροδο του χρόνου

Πηγή: https://www.alibabacloud.com/blog/what-are-time-series-databases_595165

2.2.2.3 Πλεονεκτήματα βάσεων δεδομένων NoSQL

Ευέλικτα μοντέλα δεδομένων

Οι βάσεις δεδομένων NoSQL έχουν συνήθως πολύ ευέλικτα σχήματα. Ένα ευέλικτο σχήμα κάνει εύκολες τις αλλαγές στη βάση δεδομένων καθώς αλλάζουν οι απαιτήσεις [15]. Νέα χαρακτηριστικά της εφαρμογής μπορούν να επαναλαμβάνονται γρήγορα και να ενσωματώνονται συνεχώς για να παρέχουν αξία στους χρήστες ταχύτερα.

Οριζόντια κλιμάκωση

Οι περισσότερες βάσεις δεδομένων SQL απαιτούν την κάθετη κλιμάκωση (μετάβαση σε μεγαλύτερο, ακριβότερο διακομιστή) όταν υπερβαίνουν τις απαιτήσεις χωρητικότητας του τρέχοντος διακομιστή. Αντίθετα, οι περισσότερες βάσεις δεδομένων NoSQL επιτρέπουν την οριζόντια κλιμάκωση, πράγμα που σημαίνει ότι μπορούν να προσθεθούν φθηνότεροι, βασικοί διακομιστές όποτε χρειάζεται.

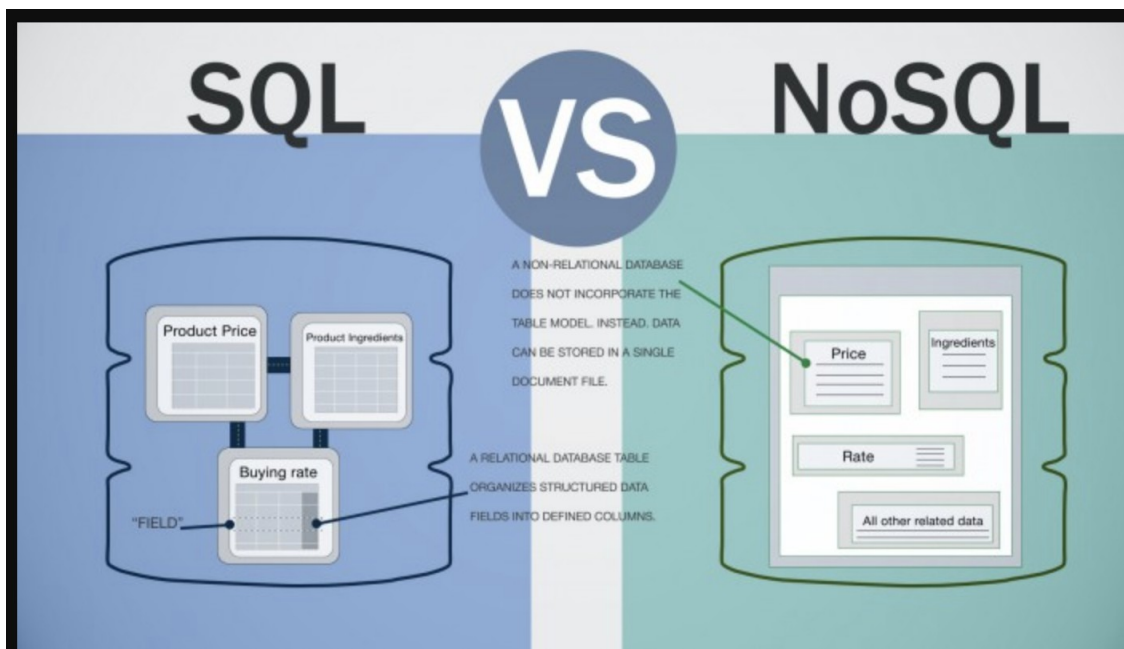
Γρήγορα ερωτήματα

Τα ερωτήματα στις βάσεις δεδομένων NoSQL μπορούν να είναι ταχύτερα από τις βάσεις δεδομένων SQL. Τα δεδομένα στις βάσεις δεδομένων SQL είναι συνήθως κανονικοποιημένα, οπότε τα ερωτήματα

για ένα μόνο αντικείμενο ή οντότητα απαιτούν να ενωθούν δεδομένα από πολλούς πίνακες. Καθώς οι πίνακες αυξάνονται σε μέγεθος, οι ενώσεις μπορεί να γίνουν ακριβές. Ωστόσο, τα δεδομένα στις βάσεις δεδομένων NoSQL αποθηκεύονται συνήθως με τρόπο που είναι βελτιστοποιημένος για ερωτήματα [16]. Ο γενικός κανόνας όταν χρησιμοποιηθεί μια μη σχεσιακή βάση δεδομένων είναι: Τα δεδομένα είναι που προσπελούνται μαζί θα πρέπει να αποθηκεύονται μαζί. Τα ερωτήματα συνήθως δεν απαιτούν joins, οπότε τα ερωτήματα είναι πολύ γρήγορα.

Εύκολο για τους προγραμματιστές

Ορισμένες βάσεις δεδομένων NoSQL, όπως η MongoDB, αντιστοιχούν τις δομές δεδομένων τους σε αυτές των δημοφιλών γλωσσών προγραμματισμού. Αυτή η αντιστοίχιση επιτρέπει στους προγραμματιστές να αποθηκεύουν τα δεδομένα τους με τον ίδιο τρόπο που τα χρησιμοποιούν στον κώδικα της εφαρμογής τους. Αν και μπορεί να φαίνεται σαν ένα ασήμαντο πλεονέκτημα, αυτή η αντιστοίχιση μπορεί να επιτρέψει στους προγραμματιστές να γράψουν λιγότερο κώδικα, οδηγώντας σε ταχύτερο χρόνο ανάπτυξης και λιγότερα σφάλματα.



Σχήμα 2.15: Σύγκριση SQL με NoSQL

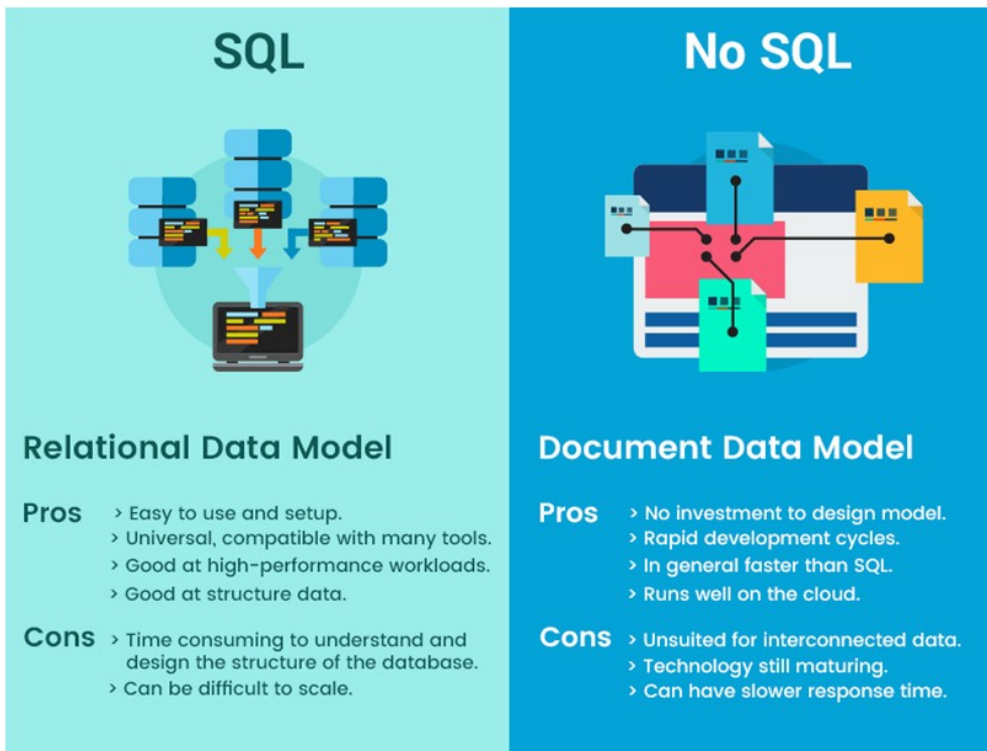
Πηγή: <https://medium.com/@mark.rethana/introduction-to-nosql-databases-c5b43f3ca1cc>

2.2.2.4 Μειονεκτήματα βάσεων δεδομένων NoSQL

Ένα από τα πιο συχνά αναφερόμενα μειονεκτήματα των βάσεων δεδομένων NoSQL είναι ότι δεν υποστηρίζουν συναλλαγές ACID (ατομικότητα, συνέπεια, απομόνωση, ανθεκτικότητα) σε πολλαπλά έγγραφα. Με τον κατάλληλο σχεδιασμό σχήματος, η ατομικότητα μιας εγγραφής είναι αποδεκτή για πολλές εφαρμογές. Ωστόσο, εξακολουθούν να υπάρχουν πολλές εφαρμογές που απαιτούν ACID σε πολλαπλές εγγραφές.

Δεδομένου ότι τα μοντέλα δεδομένων στις βάσεις δεδομένων NoSQL είναι συνήθως βελτιστοποιημένα για ερωτήματα και όχι για τη μείωση της επανάληψης δεδομένων, οι βάσεις δεδομένων NoSQL μπορεί να είναι μεγαλύτερες από τις βάσεις δεδομένων SQL. Η αποθήκευση είναι επί του παρόντος τόσο φθηνή που οι περισσότεροι θεωρούν αυτό το μειονέκτημα ήσσονος σημασίας, και ορισμένες βάσεις δεδομένων NoSQL υποστηρίζουν επίσης συμπίεση για τη μείωση του αποτυπώματος αποθήκευσης.

Ανάλογα με τον τύπο της βάσης δεδομένων NoSQL που θα επιλεγεί, ενδέχεται να μην μπορεί να επιτύχει όλες τις περιπτώσεις χρήσης σε μία μόνο βάση δεδομένων. Για παράδειγμα, οι βάσεις δεδομένων γράφων είναι εξαιρετικές για την ανάλυση των σχέσεων στα δεδομένα, αλλά μπορεί να μην παρέχουν αυτό που χρειάζεται για την καθημερινή ανάκτηση των δεδομένων, όπως ερωτήματα εύρους (range queries).



Σχήμα 2.16: Πλεονεκτήματα και μειονεκτήματα βάσεων SQL και NoSQL

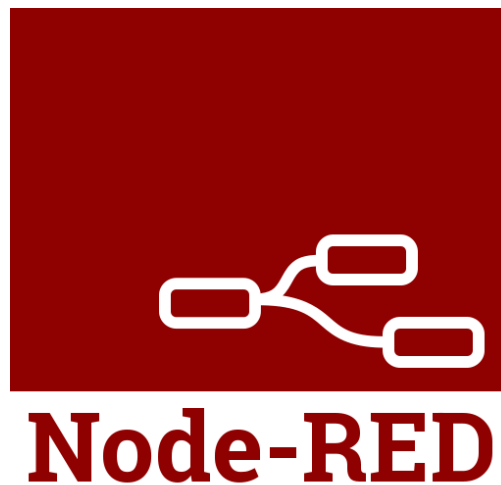
Πηγή: <https://www.clariontech.com/blog/harvest-nosql-speed-with-the-combination-of-php>

2.2.3 Πλατφόρμα Εισροής Δεδομένων

Στη συνέχεια θα αναλυθούν μερικά προγράμματα ενσωμάτωσης δεδομένων.

2.2.3.1 Node-RED

Το Node-RED είναι ένα διαδικτυακό εργαλείο προγραμματισμού ανοικτού κώδικα που χρησιμοποιεί την προσέγγιση με βάση τη ροή. Η μέθοδος αυτή βασίζεται στους λεγόμενους κόμβους, με κάθε κόμβο να έχει τη δική του συγκεκριμένη λειτουργία. Ανάλογα με τη λειτουργία του, κάθε κόμβος επεξεργάζεται στη συνέχεια τα δεδομένα εισόδου και τα αποστέλλει στον επόμενο κόμβο στη σειρά [17]. Η ίδια η συμπεριφορά της εφαρμογής δημιουργείται στη συνέχεια με την απλή τοποθέτηση κόμβων και την αλυσιδωτή σύνδεσή τους σε ροές. Εκτός από τη φιλικότητα προς το χρήστη, αυτή η προσέγγιση καθιστά επίσης τα προγράμματα που προκύπτουν πολύ σαφή και απλοποιεί σημαντικά την αντιμετώπιση προβλημάτων.



Σχήμα 2.17: Node-RED logo

Πηγή: <https://nodered.org/about/resources/>

2.2.3.2 Apache NiFi

Το Apache NiFi είναι μια πλατφόρμα εισαγωγής δεδομένων ανοικτού κώδικα σε πραγματικό χρόνο, η οποία έχει σχεδιαστεί για τη διαχείριση της μεταφοράς δεδομένων μεταξύ διαφορετικών πηγών και συστημάτων προορισμού. Το Apache NiFi υποστηρίζει ένα ευρύ φάσμα μορφών δεδομένων, όπως αρχεία καταγραφής, δεδομένα γεωγραφικής θέσης, κοινωνικές ροές και άλλα [18]. Ο Apache μπορεί να χειριστεί οτιδήποτε μπορεί να προσπελαστεί μέσω HTTPS. Το Apache NiFi υποστηρίζει πολλά διαφορετικά πρωτόκολλα, συμπεριλαμβανομένων των HTTP/S, SFTP, HDFS, καθώς και πολλά διαφορετικά συστήματα ανταλλαγής μηνυμάτων, όπως το Apache Kafka ή το ActiveMQ, και τις περισσότερες μεγάλες βάσεις δεδομένων. Αυτό σημαίνει ότι υποστηρίζεται μια μεγάλη ποικιλία πηγών και πρωτοκόλλων δεδομένων, γεγονός που καθιστά τη συγκεκριμένη πλατφόρμα δημοφιλή μεταξύ των επαγγελματιών της πληροφορικής που ασχολούνται με τεράστιες λίμνες δεδομένων και πολύπλοκες ροές δεδομένων.



Σχήμα 2.18: Apache NiFi logo

Πηγή: <https://noderedguide.com/node-red-lecture-8-advanced-flows-with-node-red/>

2.2.3.2 StreamSets

Το StreamSets είναι μια σύγχρονη πλατφόρμα ολοκλήρωσης δεδομένων αφιερωμένη στην κατασκευή των έξυπνων αγωγών δεδομένων που απαιτούνται για την τροφοδοσία των DataOps σε υβριδικές αρχιτεκτονικές και αρχιτεκτονικές πολλαπλών cloud. Αυτοματοποιώντας όσο το δυνατόν περισσότερο και αφαιρώντας το "πώς" της υλοποίησης του αγωγού δεδομένων, το StreamSets μεταφέρει το χρόνο και τους πόρους της μηχανικής δεδομένων στο "τι" των δεδομένων, ώστε οι ομάδες δεδομένων να ξοδεύουν λιγότερο χρόνο διορθώνοντας και περισσότερο χρόνο κάνοντας [19].

Το καλοκαίρι του 2021, το StreamSets ενσωμάτωσε όλες τις λειτουργίες του Control Hub και των Data Collector και Transformer Engines σε μια πλήρως διαχειρίσιμη υπηρεσία. Η StreamSets DataOps Platform είναι μια end-to-end πλατφόρμα μηχανικής δεδομένων σχεδιασμένη για να παρέχει συνεχή δεδομένα στην επιχείρηση, σχεδιασμένη για να λύνει τα προβλήματα του μηχανικού δεδομένων:

- Ελαχιστοποίηση του χρόνου εκκίνησης που απαιτείται για νέες τεχνολογίες και εύκολη επέκταση της μηχανικής δεδομένων για πιο σύνθετες λειτουργίες
- Ανάπτυξη σωληνώσεων δεδομένων που είναι ανθεκτικές στις πιο συνηθισμένες μορφές παρέκκλισης δεδομένων με όσο το δυνατόν λιγότερη παρέμβαση
- Παροχή επιχειρησιακής ορατότητας και ελέγχου διαχείρισης όλων των αγωγών, σε υβριδικές εγκαταστάσεις και εγκαταστάσεις πολλαπλών υπολογιστικών νεφών



Σχήμα 2.19: StreamSets logo

Πηγή:<https://www.eweek.com/enterprise-apps/streamsets-banks-20-million-in-vc-funding-to-control-dataflow-chaos/>

2.2.3.2 Flogo

Το Project Flogo είναι ένα αποδοτικό σε πόρους, βασισμένο στην Go οικοσύστημα ανοιχτού κώδικα για την κατασκευή εφαρμογών που βασίζονται σε συμβάντα. Σχεδιάστηκε για να αφαιρέσει το παράδειγμα της επεξεργασίας συμβάντων από την κατανάλωση συμβάντων επιτρέποντας στους προγραμματιστές να δημιουργούν μία φορά, να καταναλώνουν από οπουδήποτε και να

επεξεργάζονται χρησιμοποιώντας οποιαδήποτε από τις υποστηριζόμενες ενέργειες. Πρόκειται για ένα εξαιρετικά ελαφρύ πρόγραμμα ανοικτού κώδικα, βασισμένο στη Golang, το οποίο δίνει έμφαση στην ευελιξία ανάπτυξης, στη μητρική μηχανική μάθηση και στις λειτουργίες χωρίς διακομιστή.

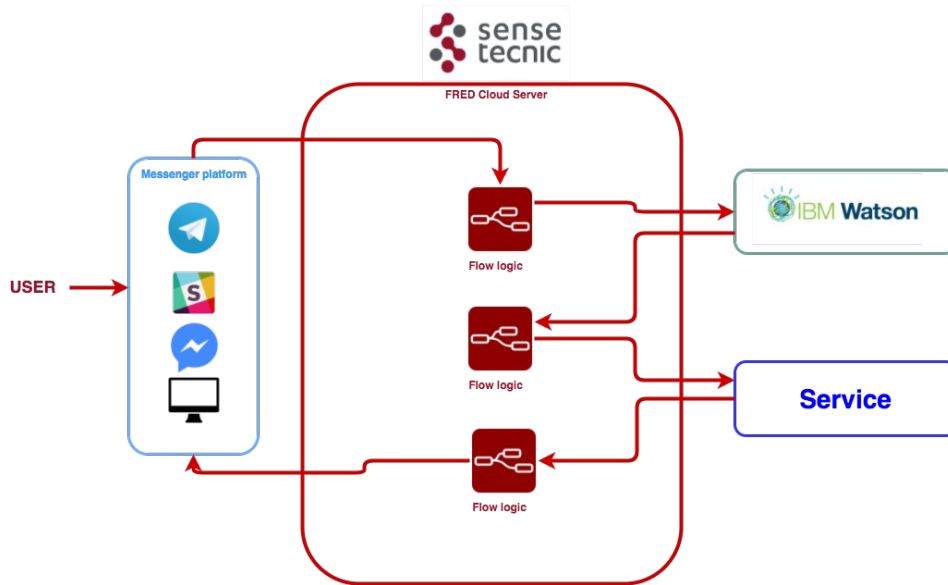


Σχήμα 2.20: Flogo logo

Πηγή:<https://www.eweek.com/enterprise-apps/streamsets-banks-20-million-in-vc-funding-to-control-dataflow-chaos/>

2.2.3.2 Διαφορές μεταξύ Node-RED και Apache NiFi

Το Node-Red προορίζεται για πιο κλασικές εργασίες και για έργα ΙΟΤ. Είναι αυτόνομο και δεν έχει σχεδιαστεί για να διανέμεται. Το Apache NiFi είναι λίγο περίπλοκο, αλλά είναι κατασκευασμένο για επεκτασιμότητα. Είναι δυνατόν να υπάρχουν πολλαπλοί κόμβοι nifi με υψηλότερη απόδοση. Το NiFi χρησιμοποιεί πολύ περισσότερους πόρους από το Node Red [20].



Σχήμα 2.21: Αναπαράσταση τρόπου λειτουργίας NODE-RED

Πηγή: <https://noderedguide.com/node-red-lecture-8-advanced-flows-with-node-red/>

2.2.4 Πρωτόκολλο Επικοινωνίας

Για το σκοπό της μεταβίβασης μηνυμάτων θα αναλυθούν τα παρακάτω πρωτόκολλα.

2.2.4.1 Zigbee

Το Zigbee είναι μια προδιαγραφή βασισμένη στο πρότυπο IEEE 802.15.4 για μια σειρά πρωτοκόλλων επικοινωνίας υψηλού επιπέδου που χρησιμοποιούνται για τη δημιουργία προσωπικών δικτύων περιοχής με μικρά, χαμηλής ισχύος ψηφιακά ραδιόφωνα, όπως για τον οικιακό αυτοματισμό, τη συλλογή δεδομένων από ιατρικές συσκευές και άλλες ανάγκες χαμηλής ισχύος και χαμηλού εύρους ζώνης, σχεδιασμένες για έργα μικρής κλίμακας που χρειάζονται ασύρματη σύνδεση. Ως εκ τούτου, το Zigbee είναι ένα ασύρματο ad hoc δίκτυο χαμηλής ισχύος, χαμηλού ρυθμού μετάδοσης δεδομένων και εγγύτητας (δηλ. προσωπικής περιοχής).

Το Zigbee είναι εκ κατασκευής φθηνότερο και απλούστερο από άλλα WPAN (ασύρματα προσωπικά δίκτυα περιοχής) όπως το WiFi και το Bluetooth. Αυτό το καθιστά κατάλληλο για επικοινωνία χαμηλής ταχύτητας και χαμηλού κόστους μεταξύ έξυπνων συσκευών [21].

Η τεχνολογία που ορίζεται από την προδιαγραφή Zigbee προορίζεται να είναι απλούστερη και λιγότερο δαπανηρή από άλλα ασύρματα δίκτυα προσωπικής περιοχής (WPAN), όπως το Bluetooth ή γενικότερα ασύρματα δίκτυα όπως το Wi-Fi. Οι εφαρμογές περιλαμβάνουν ασύρματους διακόπτες φωτισμού, οικιακούς μετρητές ενέργειας, συστήματα διαχείρισης της κυκλοφορίας και άλλο καταναλωτικό και βιομηχανικό εξοπλισμό που απαιτεί ασύρματη μεταφορά δεδομένων μικρής εμβέλειας και χαμηλού ρυθμού.

Η χαμηλή κατανάλωση ενέργειας περιορίζει τις αποστάσεις μετάδοσης σε 10-100 μέτρα οπτικής επαφής, ανάλογα με την ισχύ εξόδου και τα περιβαλλοντικά χαρακτηριστικά. Οι συσκευές Zigbee μπορούν να μεταδίδουν δεδομένα σε μεγάλες αποστάσεις περνώντας τα δεδομένα μέσω ενός δικτύου πλέγματος ενδιάμεσων συσκευών για να φτάσουν σε πιο απομακρυσμένες συσκευές. Το Zigbee χρησιμοποιείται συνήθως σε εφαρμογές χαμηλού ρυθμού δεδομένων που απαιτούν μεγάλη διάρκεια ζωής της μπαταρίας και ασφαλή δικτύωση. (Τα δίκτυα Zigbee ασφαλιζονται με συμμετρικά κλειδιά κρυπτογράφησης 128 bit.) Το Zigbee έχει καθορισμένο ρυθμό 250 kbit/s, που είναι ο καταλληλότερος για διαλείπουσες μεταδόσεις δεδομένων από έναν αισθητήρα ή μια συσκευή εισόδου.

Το Zigbee είναι ένα πρότυπο ασύρματου δικτύου πλέγματος χαμηλής ισχύος που απευθύνεται σε συσκευές που λειτουργούν με μπαταρία σε εφαρμογές ασύρματου ελέγχου και παρακολούθησης. Το Zigbee παρέχει επικοινωνία χαμηλής καθυστέρησης. Τα τσιπ Zigbee είναι συνήθως ενσωματωμένα με ραδιοφωνικά συστήματα και με μικροελεγκτές. Το Zigbee λειτουργεί στις βιομηχανικές, επιστημονικές και ιατρικές (ISM) ραδιοζώνες: 2,4 GHz στις περισσότερες δικαιοδοσίες παγκοσμίως- αν και ορισμένες συσκευές χρησιμοποιούν επίσης 784 MHz στην Κίνα, 868 MHz στην Ευρώπη και 915 MHz στις ΗΠΑ και την Αυστραλία, ωστόσο ακόμη και αυτές οι περιοχές και χώρες εξακολουθούν να χρησιμοποιούν 2,4 GHz για τις περισσότερες εμπορικές συσκευές Zigbee για οικιακή χρήση. Οι ρυθμοί δεδομένων ποικίλλουν από 20 kbit/s (ζώνη 868 MHz) έως 250 kbit/s (ζώνη 2,4 GHz).

Το Zigbee βασίζεται στο φυσικό στρώμα και τον έλεγχο πρόσβασης στα μέσα που ορίζονται στο πρότυπο IEEE 802.15.4 για ασύρματα δίκτυα προσωπικής περιοχής (WPAN) χαμηλού ρυθμού. Η προδιαγραφή περιλαμβάνει τέσσερα πρόσθετα βασικά στοιχεία: επίπεδο δικτύου, επίπεδο εφαρμογής, αντικείμενα συσκευής Zigbee (ZDOs) και αντικείμενα εφαρμογής που καθορίζονται από τον κατασκευαστή. Τα ZDOs είναι υπεύθυνα για ορισμένες εργασίες, όπως η παρακολούθηση των ρόλων των συσκευών, η διαχείριση των αιτήσεων για ένταξη σε ένα δίκτυο, καθώς και η ανακάλυψη συσκευών και η ασφάλεια.

Το επίπεδο δικτύου Zigbee υποστηρίζει εγγενώς τόσο τα δίκτυα αστέρα όσο και τα δίκτυα δέντρων, καθώς και τη γενική δικτύωση πλέγματος. Κάθε δίκτυο πρέπει να διαθέτει μία συσκευή συντονιστή. Στα δίκτυα αστέρα, ο συντονιστής πρέπει να είναι ο κεντρικός κόμβος. Τόσο τα δέντρα όσο και τα πλέγματα επιτρέπουν τη χρήση δρομολογητών Zigbee για την επέκταση της επικοινωνίας σε επίπεδο δικτύου. Ένα άλλο καθοριστικό χαρακτηριστικό του Zigbee είναι οι εγκαταστάσεις για τη διεξαγωγή ασφαλών επικοινωνιών, την προστασία της δημιουργίας και της μεταφοράς κρυπτογραφικών κλειδιών, την κρυπτογράφηση πλαισίων και τον έλεγχο της συσκευής. Βασίζεται στο βασικό πλαίσιο ασφαλείας που ορίζεται στο IEEE 802.15.4.



Σχήμα 2.22: ZigBee logo

Πηγή: <http://buildyoursmarthome.co/home-automation/protocols/zigbee/>

Ραδιοεξοπλισμός

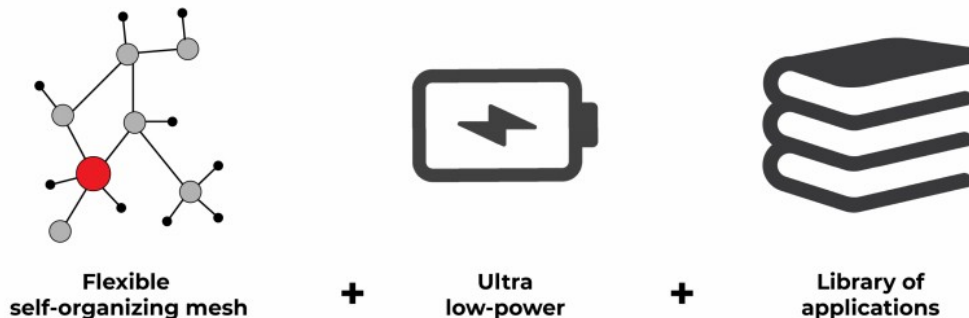
Ο σχεδιασμός του ραδιοφώνου που χρησιμοποιείται από το Zigbee έχει λίγα αναλογικά στάδια και χρησιμοποιεί ψηφιακά κυκλώματα όπου είναι δυνατόν. Διατίθενται προϊόντα που ενσωματώνουν το ραδιόφωνο και τον μικροελεγκτή σε μια ενιαία μονάδα[20].

Η διαδικασία πιστοποίησης του Zigbee περιλαμβάνει πλήρη επικύρωση των απαιτήσεων του φυσικού επιπέδου. Όλα τα ραδιόφωνα που προέρχονται από το ίδιο επικυρωμένο σύνολο μάσκας ημιαγωγών θα έχουν τα ίδια χαρακτηριστικά RF. Τα ραδιόφωνα Zigbee έχουν πολύ αυστηρούς περιορισμούς όσον αφορά την ισχύ και το εύρος ζώνης. Ένα μη επικυρωμένο φυσικό στρώμα που παρουσιάζει δυσλειτουργία μπορεί να αυξήσει την κατανάλωση ενέργειας άλλων συσκευών σε ένα δίκτυο Zigbee. Έτσι, οι ασύρματοι δοκιμάζονται με βάση τις οδηγίες που δίνονται από τη ρήτρα 6 του προτύπου 802.15.4-2006.

Αυτό το πρότυπο καθορίζει τη λειτουργία στις μη αδειοδοτημένες ζώνες ISM 2,4 έως 2,4835 GHz (παγκοσμίως), 902 έως 928 MHz (Αμερική και Αυστραλία) και 868 έως 868,6 MHz (Ευρώπη). Δεκαέξι κανάλια διατίθενται στη ζώνη των 2,4 GHz, σε απόσταση 5 MHz μεταξύ τους, χρησιμοποιώντας όμως μόνο 2 MHz εύρους ζώνης το καθένα. Οι ασύρματοι χρησιμοποιούν κωδικοποίηση απλωμένου φάσματος άμεσης ακολουθίας, η οποία διαχειρίζεται από την ψηφιακή ροή στον διαμορφωτή. Στις ζώνες 868 και 915 MHz χρησιμοποιείται η δυαδική κωδικοποίηση μετατόπισης φάσης (BPSK), ενώ στη ζώνη 2,4 GHz χρησιμοποιείται η τετραγωνική κωδικοποίηση μετατόπισης φάσης (OQPSK) που μεταδίδει δύο bit ανά σύμβολο.

Ο ακατέργαστος ρυθμός μετάδοσης δεδομένων πάνω από τον αέρα είναι 250 kbit/s ανά κανάλι στη ζώνη των 2,4 GHz, 40 kbit/s ανά κανάλι στη ζώνη των 915 MHz και 20 kbit/s στη ζώνη των 868 MHz. Η πραγματική απόδοση δεδομένων θα είναι μικρότερη από τον μέγιστο καθορισμένο ρυθμό μετάδοσης

λόγω της επιβάρυνσης των πακέτων και των καθυστερήσεων επεξεργασίας. Για εφαρμογές σε εσωτερικούς χώρους στα 2,4 GHz η απόσταση μετάδοσης είναι 10-20 m, ανάλογα με τα υλικά κατασκευής, τον αριθμό των τοίχων που πρέπει να διαπεραστούν και την ισχύ εξόδου που επιτρέπεται στη συγκεκριμένη γεωγραφική θέση. Η ισχύς εξόδου των ασυρμάτων είναι γενικά 0-20 dBm (1-100 mW).



Σχήμα 2.23: Χαρακτηριστικά του ZigBee

Πηγή: <https://zigbeealliance.org/solution/zigbee/>

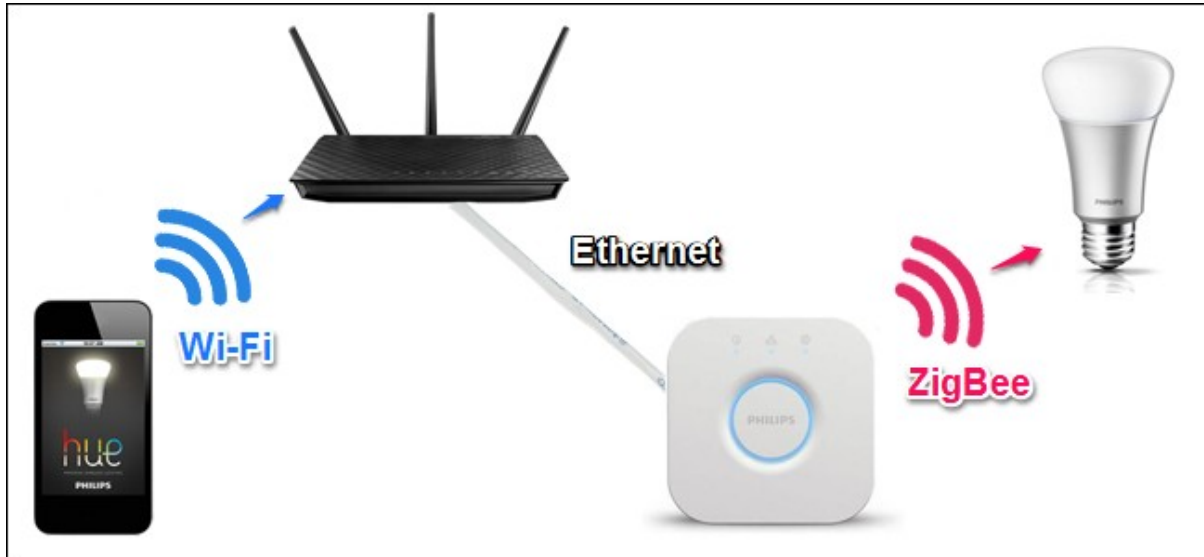
Τύποι συσκευών και τρόποι λειτουργίας

Υπάρχουν τρεις κατηγορίες συσκευών Zigbee:

1. Συντονιστής Zigbee (Zigbee Coordinator - ZC): Ο συντονιστής αποτελεί τη ρίζα του δέντρου του δικτύου και μπορεί να γεφυρώσει με άλλα δίκτυα. Υπάρχει ακριβώς ένας συντονιστής Zigbee σε κάθε δίκτυο, καθώς είναι η συσκευή που ξεκίνησε αρχικά το δίκτυο (η προδιαγραφή Zigbee LightLink επιτρέπει επίσης τη λειτουργία χωρίς συντονιστή Zigbee, καθιστώντας την πιο εύχρηστη για οικιακά προϊόντα που βρίσκονται στο ράφι). Αποθηκεύει πληροφορίες σχετικά με το δίκτυο, συμπεριλαμβανομένου του ότι ενεργεί ως κέντρο εμπιστοσύνης και αποθετήριο για τα κλειδιά ασφαλείας.
2. Δρομολογητής Zigbee (Zigbee Router - ZR): Εκτός από την εκτέλεση μιας λειτουργίας εφαρμογής, οι συσκευές δρομολογητή μπορούν να λειτουργούν ως ενδιάμεσοι δρομολογητές, διαβιβάζοντας δεδομένα σε άλλες συσκευές. Αυτού του είδους τα προϊόντα Zigbee τροφοδοτούνται συνήθως από το δίκτυο, ώστε να είναι πάντα διαθέσιμα στο δίκτυο.

Οι συσκευές δρομολογητή Zigbee ονομάζονται μερικές φορές επαναλήπτες Zigbee ή επεκτάσεις εμβέλειας Zigbee.

3. Τελική συσκευή Zigbee (Zigbee End Device - ZED): Περιέχει μόνο αρκετή λειτουργικότητα για να μιλάει με τον γονικό κόμβο (είτε τον συντονιστή είτε έναν δρομολογητή)- δεν μπορεί να αναμεταδίδει δεδομένα από άλλες συσκευές. Αυτή η σχέση επιτρέπει στον κόμβο να κοιμάται ένα σημαντικό μέρος του χρόνου δίνοντας έτσι μεγάλη διάρκεια ζωής της μπαταρίας. Αυτού του είδους τα προϊόντα συσκευών Zigbee λειτουργούν συχνά με μπαταρία. Ένα ZED απαιτεί τη μικρότερη ποσότητα μνήμης και συνεπώς μπορεί να είναι λιγότερο ακριβό στην κατασκευή από ένα ZR ή ZC.



Σχήμα 2.24: Οπτική αναπαράσταση επικοινωνίας μέσω ZigBee

Πηγή: <https://www.howtogeek.com/250614/what-are-zigbee-and-z-wave-smarthome-products/>

Τα τρέχοντα πρωτόκολλα Zigbee υποστηρίζουν δίκτυα με και χωρίς ραδιοφάρο. Στα δίκτυα που δεν διαθέτουν beacon-enabled, χρησιμοποιείται ένας μηχανισμός πρόσβασης στο κανάλι CSMA/CA χωρίς υποδοχές. Σε αυτόν τον τύπο δικτύου, οι δρομολογητές Zigbee έχουν συνήθως τους δέκτες τους συνεχώς ενεργούς, απαιτώντας έτσι πρόσθετη ισχύ. Ωστόσο, αυτό επιτρέπει ετερογενή δίκτυα στα οποία ορισμένες συσκευές λαμβάνουν συνεχώς, ενώ άλλες εκπέμπουν όταν είναι απαραίτητο. Το τυπικό παράδειγμα ενός ετερογενούς δικτύου είναι ένας ασύρματος διακόπτης φωτισμού: Ο κόμβος Zigbee στον λαμπτήρα μπορεί να λαμβάνει συνεχώς, καθώς τροφοδοτείται αξιόπιστα από την παροχή ρεύματος στον λαμπτήρα, ενώ ένας διακόπτης φωτός που τροφοδοτείται από μπαταρία θα παρέμεινε σε κατάσταση ύπνου μέχρι να πεταχτεί ο διακόπτης. Σε αυτή την περίπτωση, ο διακόπτης ξυπνάει, στέλνει μια εντολή στη λάμπα, λαμβάνει μια επιβεβαίωση και επιστρέφει στον ύπνο. Σε ένα τέτοιο δίκτυο ο κόμβος του λαμπτήρα θα είναι τουλάχιστον ένας δρομολογητής Zigbee, αν όχι ο συντονιστής Zigbee- ο κόμβος του διακόπτη είναι συνήθως μια τελική συσκευή Zigbee. Σε δίκτυα με δυνατότητα beacon, οι δρομολογητές Zigbee μεταδίδουν περιοδικά beacons για να επιβεβαιώσουν την παρουσία

τους σε άλλους κόμβους του δικτύου. Οι κόμβοι μπορούν να κοιμούνται μεταξύ των beacons, επεκτείνοντας έτσι τη διάρκεια ζωής της μπαταρίας τους. Τα χρονικά διαστήματα φάρου εξαρτώνται από το ρυθμό δεδομένων- μπορεί να κυμαίνονται από 15,36 χιλιοστά του δευτερολέπτου έως 251,65824 δευτερόλεπτα στα 250 kbit/s, από 24 χιλιοστά του δευτερολέπτου έως 393,216 δευτερόλεπτα στα 40 kbit/s και από 48 χιλιοστά του δευτερολέπτου έως 786,432 δευτερόλεπτα στα 20 kbit/s. Τα μεγάλα διαστήματα φάρου απαιτούν ακριβή χρονισμό, ο οποίος μπορεί να είναι δαπανηρός για την υλοποίηση σε προϊόντα χαμηλού κόστους.



Σχήμα 2.25:Αναπαράσταση επικοινωνίας IoT μέσω ZigBee

Πηγή: <https://www.ineltek.com/en/industrys-first-zigbee-pro-with-green-power-certified-platform-unveiled-by-microchip/>

Σε γενικές γραμμές, τα πρωτόκολλα Zigbee ελαχιστοποιούν το χρόνο που ο ασύρματος είναι ενεργοποιημένος, ώστε να μειωθεί η χρήση ενέργειας. Στα δίκτυα beaconing, οι κόμβοι χρειάζεται να είναι ενεργοί μόνο όσο μεταδίδεται ένα beacon. Στα δίκτυα χωρίς ραδιοφάρο, η κατανάλωση ισχύος είναι σαφώς ασύμμετρη: ορισμένες συσκευές είναι πάντα ενεργές, ενώ άλλες περνούν τον περισσότερο χρόνο τους σε κατάσταση ύπνου.

Εκτός από το Smart Energy Profile 2.0, οι συσκευές Zigbee πρέπει να συμμορφώνονται με το πρότυπο IEEE 802.15.4-2003 Low-rate Wireless Personal Area Network (LR-WPAN). Το πρότυπο καθορίζει τα κατώτερα στρώματα πρωτοκόλλου - το φυσικό στρώμα (PHY) και το τμήμα ελέγχου

πρόσβασης στα μέσα του στρώματος σύνδεσης δεδομένων. Ο βασικός τρόπος πρόσβασης στο κανάλι είναι η πολλαπλή πρόσβαση με αίσθηση φέροντος και αποφυγή συγκρούσεων (CSMA/CA). Δηλαδή, οι κόμβοι επικοινωνούν με έναν τρόπο κάπως ανάλογο με τον τρόπο που συνομιλούν οι άνθρωποι: ένας κόμβος ελέγχει για λίγο αν οι άλλοι κόμβοι δεν μιλούν πριν ξεκινήσει. Το CSMA/CA δεν χρησιμοποιείται σε τρεις αξιοσημείωτες εξαιρέσεις:

- Επιβεβαιώσεις μηνυμάτων
- Τα σήματα αποστέλλονται με ένα σταθερό χρονοδιάγραμμα.
- Οι συσκευές σε δίκτυα με ραδιοφάρο που έχουν απαιτήσεις χαμηλής καθυστέρησης και πραγματικού χρόνου μπορούν επίσης να χρησιμοποιούν εγγυημένες χρονοθυρίδες.

Επίπεδο δικτύου

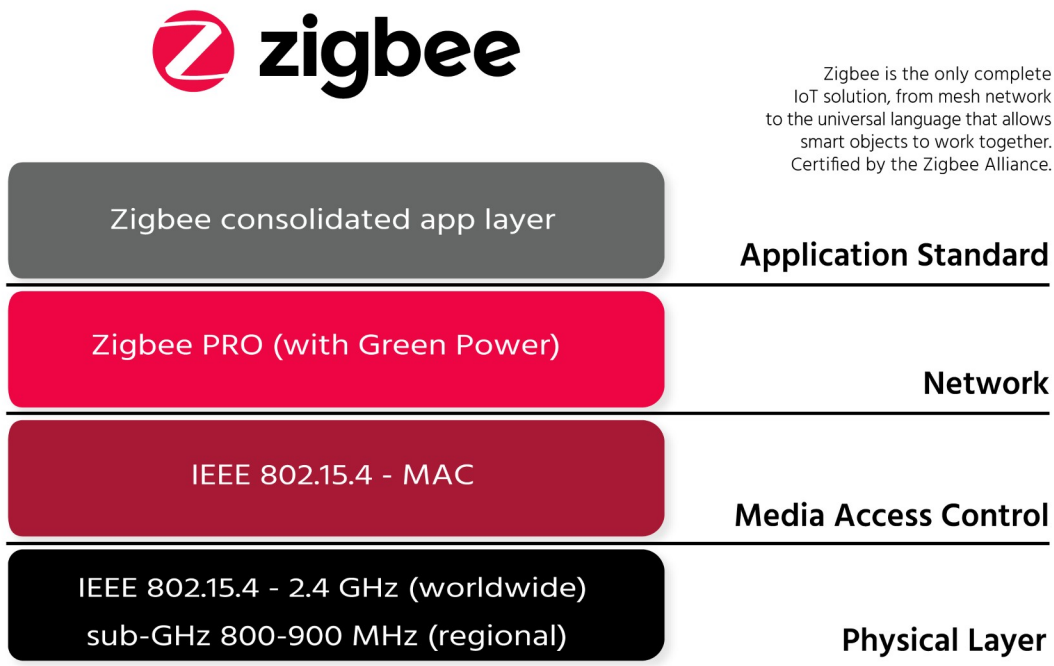
Οι κύριες λειτουργίες του στρώματος δικτύου είναι να διασφαλίζει τη σωστή χρήση του υποστρώματος MAC και να παρέχει μια κατάλληλη διεπαφή για χρήση από το επόμενο ανώτερο στρώμα, δηλαδή το στρώμα εφαρμογής. Το επίπεδο δικτύου ασχολείται με λειτουργίες δικτύου, όπως η σύνδεση, η αποσύνδεση και η δημιουργία δικτύων. Μπορεί να δημιουργήσει ένα δίκτυο, να διαθέσει διευθύνσεις και να προσθέσει και να αφαιρέσει συσκευές. Αυτό το επίπεδο κάνει χρήση τοπολογιών αστέρα, πλέγματος και δέντρου.

Η οντότητα δεδομένων του επιπέδου μεταφοράς δημιουργεί και διαχειρίζεται μονάδες δεδομένων πρωτοκόλλου με την καθοδήγηση του επιπέδου εφαρμογής και εκτελεί δρομολόγηση σύμφωνα με την τρέχουσα τοπολογία. Η οντότητα ελέγχου χειρίζεται τη διαμόρφωση νέων συσκευών και δημιουργεί νέα δίκτυα. Μπορεί να καθορίσει αν μια γειτονική συσκευή ανήκει στο δίκτυο και ανακαλύπτει νέους γείτονες και δρομολογητές.

Το πρωτόκολλο δρομολόγησης που χρησιμοποιείται από το επίπεδο δικτύου είναι το AODV[26]. Για την εύρεση μιας συσκευής προορισμού, το AODV χρησιμοποιείται για τη μετάδοση ενός αιτήματος δρομολόγησης σε όλους τους γείτονές του. Στη συνέχεια, οι γείτονες μεταδίδουν το αίτημα στους γείτονές τους και συνεχίζουν μέχρι να φτάσουν στον προορισμό. Μόλις επιτευχθεί ο προορισμός, μια απάντηση διαδρομής αποστέλλεται μέσω μονοαπευθυντικής μετάδοσης ακολουθώντας τη διαδρομή χαμηλότερου κόστους πίσω στην πηγή. Μόλις η πηγή λάβει την απάντηση, ενημερώνει τον πίνακα δρομολόγησης με τη διεύθυνση προορισμού του επόμενου άλματος στη διαδρομή και το σχετικό κόστος διαδρομής.

Επίπεδο εφαρμογής

Το επίπεδο εφαρμογής είναι το υψηλότερο επίπεδο που ορίζεται από τις προδιαγραφές και αποτελεί την πραγματική διεπαφή του συστήματος Zigbee με τους τελικούς χρήστες. Περιλαμβάνει την πλειονότητα των στοιχείων που προστίθενται από την προδιαγραφή Zigbee: τόσο το ZDO (αντικείμενο συσκευής Zigbee) όσο και οι διαδικασίες διαχείρισής του, μαζί με τα αντικείμενα εφαρμογής που ορίζονται από τον κατασκευαστή, θεωρούνται μέρος αυτού του στρώματος. Αυτό το στρώμα δεσμεύει πίνακες, στέλνει μηνύματα μεταξύ δεσμευμένων συσκευών, διαχειρίζεται διευθύνσεις ομάδων, ανασυνθέτει πακέτα και επίσης μεταφέρει δεδομένα. Είναι υπεύθυνο για την παροχή υπηρεσιών σε προφίλ συσκευών Zigbee.



Σχήμα 2.26: Επίπεδα επικοινωνίας ZigBee

Πηγή: https://zigbeealliance.org/news_and_articles/the-zigbee-alliance-introduces-first-multi-band-iot-mesh-network-technology-for-massive-iot-deployments/

Κύρια συστατικά

Το ZDO (Zigbee device object), ένα πρωτόκολλο στη στήβα πρωτοκόλλων Zigbee, είναι υπεύθυνο για τη συνολική διαχείριση της συσκευής, τα κλειδιά ασφαλείας και τις πολιτικές. Είναι υπεύθυνο για τον ορισμό του ρόλου μιας συσκευής είτε ως συντονιστή είτε ως τελικής συσκευής, όπως αναφέρθηκε παραπάνω, αλλά και για την ανακάλυψη νέων συσκευών στο δίκτυο και τον προσδιορισμό των προσφερόμενων υπηρεσιών τους. Στη συνέχεια μπορεί να προχωρήσει στη δημιουργία ασφαλών συνδέσεων με εξωτερικές συσκευές και να απαντήσει ανάλογα σε αιτήσεις δέσμευσης.

Το υποεπίπεδο υποστήριξης εφαρμογών (APS) είναι το άλλο κύριο τυποποιημένο στοιχείο της στήβας και ως τέτοιο προσφέρει μια σαφώς καθορισμένη διεπαφή και υπηρεσίες ελέγχου. Λειτουργεί ως γέφυρα μεταξύ του στρώματος δικτύου και των άλλων στοιχείων του στρώματος εφαρμογών: διατηρεί ενημερωμένους πίνακες δεσμεύσεων με τη μορφή βάσης δεδομένων, οι οποίοι μπορούν να χρησιμοποιηθούν για την εύρεση κατάλληλων συσκευών ανάλογα με τις υπηρεσίες που απαιτούνται και αυτές που προσφέρουν οι διάφορες συσκευές. Ως ένωση μεταξύ των δύο καθορισμένων στρωμάτων, δρομολογεί επίσης μηνύματα μεταξύ των στρωμάτων της στήβας πρωτοκόλλων.

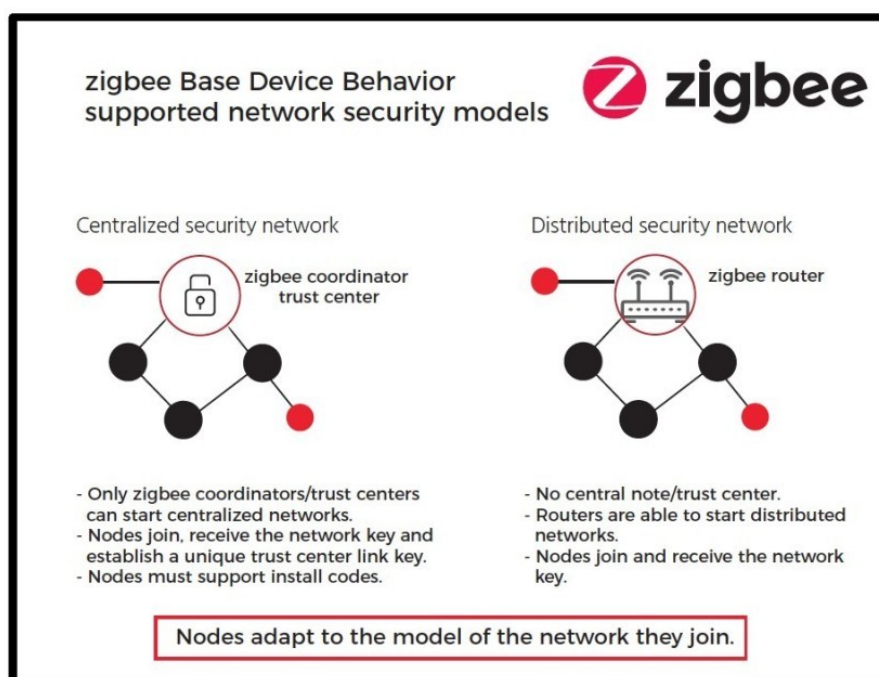
Υπηρεσίες ασφαλείας

Ως ένα από τα καθοριστικά χαρακτηριστικά του, το Zigbee παρέχει εγκαταστάσεις για τη διεξαγωγή ασφαλών επικοινωνιών, την προστασία της δημιουργίας και της μεταφοράς κρυπτογραφικών κλειδιών και την κρυπτογράφηση δεδομένων. Βασίζεται στο βασικό πλαίσιο ασφαλείας που ορίζεται στο IEEE 802.15.4.

Βασικό μοντέλο ασφάλειας

Ο βασικός μηχανισμός για τη διασφάλιση της εμπιστευτικότητας είναι η επαρκής προστασία όλου του υλικού κλειδιών. Τα κλειδιά αποτελούν τον ακρογωνιαίο λίθο της αρχιτεκτονικής ασφάλειας- ως εκ τούτου, η προστασία τους είναι υψίστης σημασίας και τα κλειδιά δεν πρέπει ποτέ να μεταφέρονται μέσω ενός μη ασφαλούς καναλιού. Μια στιγμιαία εξαίρεση σε αυτόν τον κανόνα συμβαίνει κατά την αρχική φάση της προσθήκης στο δίκτυο μιας προηγουμένως μη ρυθμισμένης συσκευής. Κατά την αρχική εγκατάσταση των κλειδιών, καθώς και κατά την επεξεργασία των πληροφοριών ασφαλείας, πρέπει να υπάρχει εμπιστοσύνη. Το μοντέλο του δικτύου Zigbee πρέπει να λαμβάνει ιδιαίτερη μέριμνα για τα ζητήματα ασφάλειας, καθώς τα ad hoc δίκτυα μπορεί να είναι φυσικά προσβάσιμα σε εξωτερικές συσκευές. Επίσης, η κατάσταση του περιβάλλοντος εργασίας δεν μπορεί να προβλεφθεί.

Μέσα στη στοίβα πρωτοκόλλων, τα διάφορα στρώματα δικτύου δεν διαχωρίζονται κρυπτογραφικά, οπότε απαιτούνται πολιτικές πρόσβασης και προϋποτίθεται συμβατικός σχεδιασμός. Το ανοιχτό μοντέλο εμπιστοσύνης εντός μιας συσκευής επιτρέπει την κοινή χρήση κλειδιών, η οποία μειώνει σημαντικά το πιθανό κόστος. Παρ' όλα αυτά, το επίπεδο που δημιουργεί ένα πλαίσιο είναι υπεύθυνο για την ασφάλειά του. Καθώς μπορεί να υπάρχουν κακόβουλες συσκευές, κάθε ωφέλιμο φορτίο στρώματος δικτύου πρέπει να κρυπτογραφείται, ώστε η μη εξουσιοδοτημένη κυκλοφορία να μπορεί να αποκοπεί αμέσως. Εξάιρεση αποτελεί, και πάλι, η μετάδοση του κλειδιού δικτύου, το οποίο προσδίδει ένα ενιαίο στρώμα ασφαλείας στο δίκτυο, σε μια νέα συσκευή σύνδεσης.



Σχήμα 2.27: Μοντέλα ασφαλείας δικτύου στο ZigBee

Πηγή: <https://research.kudelskisecurity.com/2017/11/08/zigbee-security-basics-part-2/>

Αρχιτεκτονική ασφαλείας

Η αρχιτεκτονική ασφαλείας του Zigbee βασίζεται στο CCM, το οποίο προσθέτει χαρακτηριστικά κρυπτογράφησης και ακεραιότητας μόνο στη λειτουργία CCM. Το Zigbee χρησιμοποιεί κλειδιά 128 bit για την υλοποίηση των μηχανισμών ασφαλείας του. Ένα κλειδί μπορεί να συσχετιστεί είτε με ένα δίκτυο, όντας χρησιμοποιήσιμο από τα επίπεδα Zigbee και το υποεπίπεδο MAC, είτε με μια σύνδεση,

αποκτώντας το μέσω προεγκατάστασης, συμφωνίας ή μεταφοράς. Η καθιέρωση των κλειδιών συνδέσεων βασίζεται σε ένα κύριο κλειδί που ελέγχει την αντιστοιχία κλειδιών συνδέσεων. Τελικά, τουλάχιστον, το αρχικό κύριο κλειδί πρέπει να λαμβάνεται μέσω ασφαλούς μέσου (μεταφορά ή προεγκατάσταση), καθώς από αυτό εξαρτάται η ασφάλεια ολόκληρου του δικτύου. Τα κλειδιά σύνδεσης και τα κύρια κλειδιά είναι ορατά μόνο στο επίπεδο εφαρμογής. Διαφορετικές υπηρεσίες χρησιμοποιούν διαφορετικές μονόδρομες παραλλαγές του κλειδιού σύνδεσης για να αποφεύγονται διαρροές και κίνδυνοι ασφαλείας.

Η διανομή κλειδιών είναι μία από τις σημαντικότερες λειτουργίες ασφαλείας του δικτύου. Ένα ασφαλές δίκτυο θα ορίσει μια ειδική συσκευή, το κέντρο εμπιστοσύνης, το οποίο εμπιστεύονται οι άλλες συσκευές για τη διανομή των κλειδιών ασφαλείας. Ιδανικά, οι συσκευές θα έχουν προεγκατεστημένη τη διεύθυνση του κέντρου εμπιστοσύνης και το αρχικό κύριο κλειδί- εάν επιτρέπεται μια στιγμιαία ευπάθεια, θα αποστέλλεται όπως περιγράφεται παραπάνω. Τυπικές εφαρμογές χωρίς ιδιαίτερες ανάγκες ασφαλείας θα χρησιμοποιούν για την επικοινωνία ένα κλειδί δικτύου που παρέχεται από το κέντρο εμπιστοσύνης (μέσω του αρχικά μη ασφαλούς καναλιού).

Έτσι, το κέντρο εμπιστοσύνης διατηρεί και το κλειδί δικτύου και παρέχει ασφάλεια από σημείο σε σημείο. Οι συσκευές θα δέχονται μόνο επικοινωνίες που προέρχονται από ένα κλειδί που παρέχεται από το κέντρο εμπιστοσύνης, εκτός από το αρχικό κύριο κλειδί. Η αρχιτεκτονική ασφαλείας κατανέμεται μεταξύ των επιπέδων δικτύου ως εξής:

Το υποεπίπεδο MAC είναι ικανό για αξιόπιστες επικοινωνίες ενός βήματος. Κατά κανόνα, το επίπεδο ασφαλείας που χρησιμοποιεί καθορίζεται από τα ανώτερα στρώματα.

Το επίπεδο δικτύου διαχειρίζεται τη δρομολόγηση, επεξεργάζεται τα ληφθέντα μηνύματα και είναι ικανό να μεταδίδει αιτήματα. Εάν είναι διαθέσιμο, τα εξερχόμενα πλαίσια χρησιμοποιούν το κατάλληλο κλειδί σύνδεσης σύμφωνα με τη δρομολόγηση. Διαφορετικά, χρησιμοποιείται το κλειδί δικτύου.

Το επίπεδο εφαρμογών προσφέρει υπηρεσίες δημιουργίας κλειδιών και μεταφοράς τόσο στο ZDO όσο και στις εφαρμογές.

2.2.4.2 AMQP

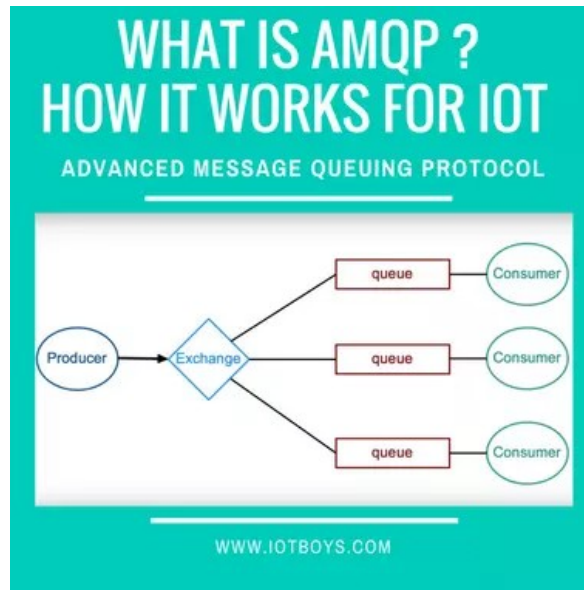


Σχήμα 2.28: AMQP logo

Πηγή: <https://www.survivingwithandroid.com/2016/08/iot-protocols-list.html>

Το Advanced Message Queuing Protocol (AMQP) είναι ένα ανοικτό πρότυπο πρωτόκολλο επιπέδου εφαρμογής για ενδιάμεσο λογισμικό προσανατολισμένο σε μηνύματα. Τα καθοριστικά χαρακτηριστικά του AMQP είναι ο προσανατολισμός των μηνυμάτων, η ουρά αναμονής, η δρομολόγηση (συμπεριλαμβανομένων των σημείων από σημείο σε σημείο και της δημοσίευσης και συνδρομής), η αξιοπιστία και η ασφάλεια.

Το AMQP επιβάλλει τη συμπεριφορά του παρόχου μηνυμάτων και του πελάτη σε βαθμό που οι υλοποιήσεις από διαφορετικούς προμηθευτές να είναι διαλειτουργικές, με τον ίδιο τρόπο που το SMTP, το HTTP, το FTP κ.λπ. έχουν δημιουργήσει διαλειτουργικά συστήματα. Οι προηγούμενες τυποποιήσεις του ενδιάμεσου λογισμικού συνέβησαν σε επίπεδο API (π.χ. JMS) και επικεντρώθηκαν στην τυποποίηση της αλληλεπίδρασης των προγραμματιστών με διαφορετικές υλοποιήσεις ενδιάμεσου λογισμικού και όχι στην παροχή διαλειτουργικότητας μεταξύ πολλαπλών υλοποιήσεων. Σε αντίθεση με το JMS, το οποίο ορίζει ένα API και ένα σύνολο συμπεριφορών που πρέπει να παρέχει μια υλοποίηση ανταλλαγής μηνυμάτων, το AMQP είναι ένα πρωτόκολλο σε επίπεδο καλωδίου. Ένα πρωτόκολλο σε επίπεδο καλωδίου είναι μια περιγραφή της μορφής των δεδομένων που αποστέλλονται μέσω του δικτύου ως ροή bytes. Κατά συνέπεια, κάθε εργαλείο που μπορεί να δημιουργήσει και να ερμηνεύσει μηνύματα που συμμορφώνονται με αυτή τη μορφή δεδομένων μπορεί να διαλειτουργήσει με οποιοδήποτε άλλο συμβατό εργαλείο, ανεξάρτητα από τη γλώσσα υλοποίησης.



Σχήμα 2.29: Χρήση του AMQP σε IoT

Πηγή: <https://iotboys.com/what-is-amqp-how-amqp-works-for-internet-of-things/>

Το AMQP είναι ένα δυαδικό πρωτόκολλο επιπέδου εφαρμογής, σχεδιασμένο να υποστηρίζει αποτελεσματικά μια μεγάλη ποικιλία εφαρμογών ανταλλαγής μηνυμάτων και προτύπων επικοινωνίας. Παρέχει ελεγχόμενη ροή, επικοινωνία προσανατολισμένη σε μηνύματα με εγγυήσεις παράδοσης μηνυμάτων, όπως *at-most-once* (όπου κάθε μήνυμα παραδίδεται μία φορά ή ποτέ), *at-least-once* (όπου κάθε μήνυμα είναι βέβαιο ότι θα παραδοθεί, αλλά μπορεί να παραδοθεί πολλές φορές) και *exactly-once* (όπου το μήνυμα θα φτάσει πάντα σίγουρα και μόνο μία φορά), και πιστοποίηση και/ή κρυπτογράφηση με βάση το SASL και/ή το TLS. Προϋποθέτει ένα υποκείμενο αξιόπιστο πρωτόκολλο επιπέδου μεταφοράς, όπως το Transmission Control Protocol (TCP) [22].

Η προδιαγραφή AMQP ορίζεται σε διάφορα επίπεδα: (i) ένα σύστημα τύπων, (ii) ένα συμμετρικό, ασύγχρονο πρωτόκολλο για τη μεταφορά μηνυμάτων από μια διεργασία σε μια άλλη, (iii) μια τυποποιημένη, επεκτάσιμη μορφή μηνυμάτων και (iv) ένα σύνολο τυποποιημένων αλλά επεκτάσιμων "δυνατοτήτων ανταλλαγής μηνυμάτων".

Το AMQP λύνει από μόνο του πολλά προβλήματα: το πρωτόκολλο (σε συνεργασία με έναν μεσίτη μηνυμάτων) εξασφαλίζει μια ισχυρή μεταφορά δεδομένων, ενώ παράλληλα επιτρέπει την αποθήκευση μηνυμάτων σε μια ουρά. Αυτό με τη σειρά του επιτρέπει μια ασύγχρονη επικοινωνία: ο αποστολέας και ο παραλήπτης δεν χρειάζεται να εργάζονται ο ένας γύρω από τον άλλον. Ο παραλήπτης (καταναλωτής) του μηνύματος δεν χρειάζεται να δεχτεί άμεσα τις πληροφορίες, να τις επεξεργαστεί και να επιβεβαιώσει την παραλαβή στον αποστολέα (παραγωγό). Αντ' αυτού, συλλέγει το μήνυμα από την ουρά όταν έχει διαθέσιμη χωρητικότητα. Εν τω μεταξύ, αυτό δίνει την ευκαιρία στον παραγωγό να συνεχίσει να εργάζεται - έτσι δεν υπάρχει χρόνος αδράνειας.

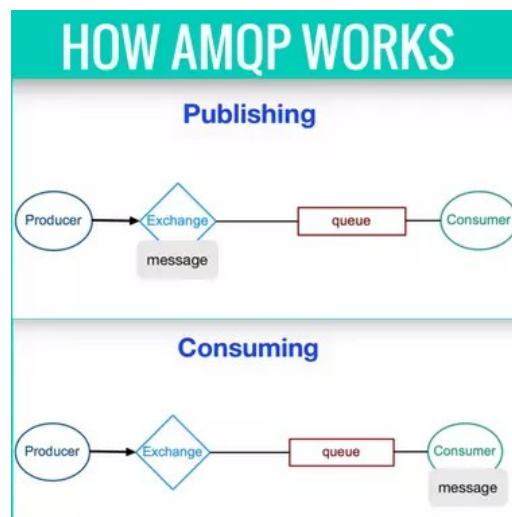
Η επιτυχία του σχετικά πρόσφατου πρωτοκόλλου έχει επίσης να κάνει με τη διαλειτουργικότητα. Το Advanced Message Queuing Protocol καθιερώνει μια δική του κοινή βάση. Αυτό επιτρέπει ακόμη και τη συγγραφή διαφορετικών εφαρμογών σε διαφορετικές γλώσσες προγραμματισμού. Με αυτόν τον τρόπο, ακόμη και προγράμματα διαφορετικών οργανισμών μπορούν εύκολα να επικοινωνούν μεταξύ τους. Και καθώς το AMQP είναι ελεύθερα διαθέσιμο, κάθε εταιρεία μπορεί να χρησιμοποιήσει το πρωτόκολλο χωρίς επιπλέον κόστος.

Περιγραφή του AMQP 1.0

Σύστημα τύπου

Το AMQP ορίζει ένα αυτοπεριγραφόμενο σύστημα κωδικοποίησης που επιτρέπει τη διαλειτουργική αναπαράσταση ενός ευρέος φάσματος ευρέως χρησιμοποιούμενων τύπων. Επιτρέπει επίσης να επισημανθούν τα δεδομένα τύπου με πρόσθετο νόημα, για παράδειγμα, μια συγκεκριμένη τιμή συμβολοσειράς μπορεί να επισημανθεί έτσι ώστε να μπορεί να γίνει κατανοητή ως διεύθυνση URL. Ομοίως, μια τιμή χάρτη που περιέχει ζεύγη κλειδιών-τιμών για "όνομα", "διεύθυνση" κ.λπ., μπορεί να σχολιαστεί ως αναπαράσταση ενός τύπου "πελάτη".

Το σύστημα τύπων χρησιμοποιείται για τον ορισμό μιας μορφής μηνύματος που επιτρέπει την έκφραση και κατανόηση τυποποιημένων και εκτεταμένων μεταδεδομένων από οντότητες επεξεργασίας. Χρησιμοποιείται επίσης για τον ορισμό των πρωταρχικών μέσων επικοινωνίας μέσω των οποίων ανταλλάσσονται μηνύματα μεταξύ αυτών των οντοτήτων, δηλαδή των σωμάτων πλαισίων AMQP.



Σχήμα 2.30: Λειτουργία του AMQP

Πηγή: <https://iotboys.com/what-is-amqp-how-amqp-works-for-internet-of-things/>

Επιδόσεις και το πρωτόκολλο σύνδεσης

Η βασική μονάδα δεδομένων στο AMQP είναι ένα πλαίσιο. Έχουν οριστεί εννέα σώματα πλαισίων AMQP τα οποία χρησιμοποιούνται για την έναρξη, τον έλεγχο και τη διακοπή της μεταφοράς μηνυμάτων μεταξύ δύο ομοτίμων. Αυτά είναι τα εξής:

- open (η σύνδεση)
- έναρξη (η σύνοδος)
- attach (η σύνδεση)
- μεταφορά
- ροή
- διάθεση
- αποσύνδεση (του συνδέσμου)
- τέλος (της συνεδρίας)
- κλείσιμο (της σύνδεσης)

Το πρωτόκολλο σύνδεσης βρίσκεται στην καρδιά του AMQP.

Ένα σώμα πλαισίου attach αποστέλλεται για να ξεκινήσει μια νέα σύνδεση, ενώ ένα detach για να διαλύσει μια σύνδεση. Οι συνδέσεις μπορούν να δημιουργηθούν για τη λήψη ή την αποστολή μηνυμάτων.

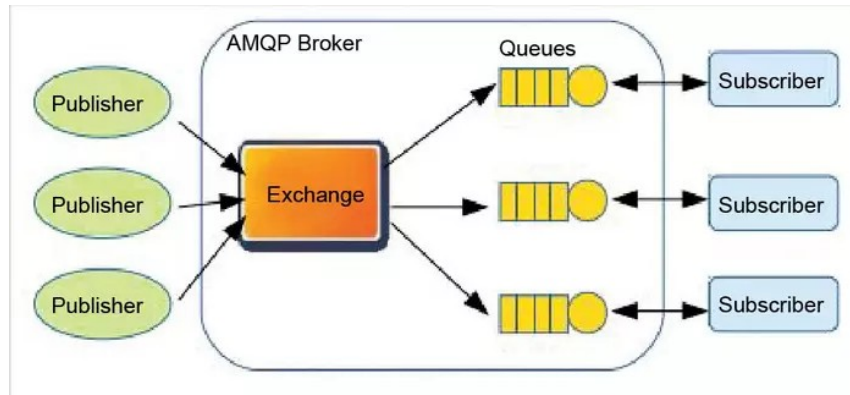
Τα μηνύματα αποστέλλονται μέσω ενός εγκατεστημένου συνδέσμου χρησιμοποιώντας το πλαίσιο μεταφοράς. Τα μηνύματα σε μια σύνδεση ρέουν προς μία μόνο κατεύθυνση.

Οι μεταφορές υπόκεινται σε ένα σύστημα ελέγχου ροής με βάση την πίστωση, το οποίο διαχειρίζεται με τη χρήση πλαισίων ροής. Αυτό επιτρέπει σε μια διεργασία να προστατεύεται από την υπερφόρτωση από πολύ μεγάλο όγκο μηνυμάτων ή πιο απλά να επιτρέπει σε μια συνδρομητική σύνδεση να τραβάει μηνύματα όπως και όταν το επιθυμεί.

Κάθε μεταφερόμενο μήνυμα πρέπει τελικά να διακανονίζεται. Ο διακανονισμός εξασφαλίζει ότι ο αποστολέας και ο παραλήπτης συμφωνούν για την κατάσταση της μεταφοράς, παρέχοντας εγγυήσεις αξιοπιστίας. Οι αλλαγές στην κατάσταση και το διακανονισμό για μια μεταφορά (ή ένα σύνολο μεταφορών) κοινοποιούνται μεταξύ των ομοτίμων εταίρων χρησιμοποιώντας το πλαίσιο διάθεσης. Με αυτόν τον τρόπο μπορούν να επιβληθούν διάφορες εγγυήσεις αξιοπιστίας: at-most-once, at-least-once και exactly-once.

Πολλαπλές συνδέσεις, και προς τις δύο κατευθύνσεις, μπορούν να ομαδοποιηθούν σε μια σύνοδο. Μια σύνοδος είναι μια αμφίδρομη, διαδοχική συνομιλία μεταξύ δύο ομοτίμων που ξεκινά με ένα πλαίσιο έναρξης και τερματίζεται με ένα πλαίσιο λήξης. Μια σύνδεση μεταξύ δύο ομοτίμων μπορεί να έχει πολλαπλές συνόδους πολυπλεγμένες πάνω της, κάθε μία από τις οποίες είναι λογικά ανεξάρτητη. Οι

συνδέσεις ξεκινούν με ένα πλαίσιο ανοίγματος στο οποίο εκφράζονται οι δυνατότητες του αποστέλλοντος ομότιμου και τερματίζονται με ένα πλαίσιο κλεισίματος.



Σχήμα 2.31: Πρωτόκολλο επικοινωνίας AMQP

Πηγή: <https://www.electronicshobby.com/technology-trends/popular-iot-protocols/2>

Μορφή μηνύματος

Το AMQP ορίζει ως γυμνό μήνυμα, το μέρος του μηνύματος που δημιουργείται από την εφαρμογή αποστολής. Αυτό θεωρείται αμετάβλητο καθώς το μήνυμα μεταφέρεται μεταξύ μιας ή περισσότερων διεργασιών.

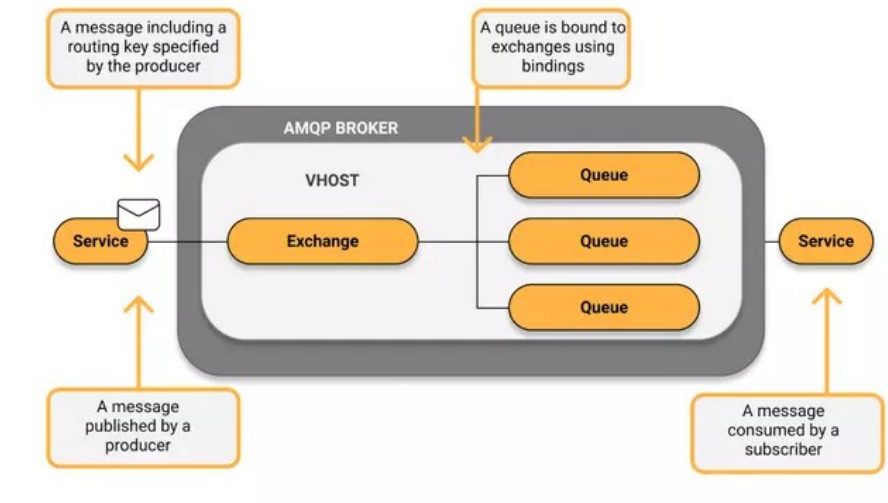
Η διασφάλιση ότι το μήνυμα όπως αποστέλλεται από την εφαρμογή είναι αμετάβλητο επιτρέπει την από άκρο σε άκρο υπογραφή ή/και κρυπτογράφηση του μηνύματος και διασφαλίζει ότι τυχόν έλεγχοι ακεραιότητας (π.χ. hashes ή digests) παραμένουν έγκυροι. Το μήνυμα μπορεί να σχολιαστεί από ενδιάμεσους φορείς κατά τη μεταφορά, αλλά οι σχολιασμοί αυτοί διατηρούνται διακριτοί από το αμετάβλητο γυμνό μήνυμα [23]. Οι σχολιασμοί μπορούν να προστεθούν πριν ή μετά το γυμνό μήνυμα.

Η επικεφαλίδα είναι ένα τυποποιημένο σύνολο σχολίων που σχετίζονται με την παράδοση, τα οποία μπορούν να ζητηθούν ή να υποδειχθούν για ένα μήνυμα και περιλαμβάνουν χρόνο ζωής, διάρκεια, προτεραιότητα.

Το ίδιο το γυμνό μήνυμα είναι δομημένο ως ένας προαιρετικός κατάλογος τυποποιημένων ιδιοτήτων (αναγνωριστικό μηνύματος, αναγνωριστικό χρήστη, χρόνος δημιουργίας, απάντηση σε, θέμα, αναγνωριστικό συσχέτισης, αναγνωριστικό ομάδας κ.λπ.), ένας προαιρετικός κατάλογος ιδιοτήτων που σχετίζονται με την εφαρμογή (δηλ. εκτεταμένες ιδιότητες) και ένα σώμα, το οποίο το AMQP αποκαλεί δεδομένα εφαρμογής.

Οι ιδιότητες προσδιορίζονται στο σύστημα τύπων AMQP, όπως και οι σχολιασμοί. Τα δεδομένα της εφαρμογής μπορούν να έχουν οποιαδήποτε μορφή και σε οποιαδήποτε κωδικοποίηση επιλέγει η

εφαρμογή. Μια επιλογή είναι η χρήση του συστήματος τύπων AMQP για την αποστολή δομημένων, αυτοπεριγραφόμενων δεδομένων.



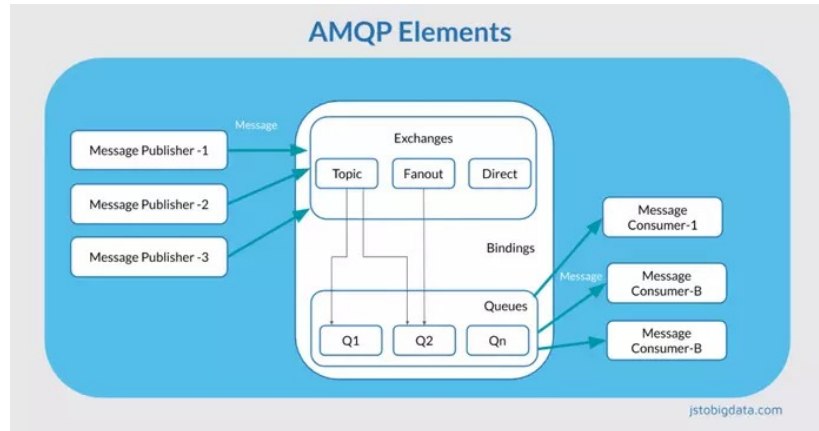
Σχήμα 2.32: Τρόπος επικοινωνίας μέσω AMQP

Πηγή: <https://www.cloudamqp.com/blog/2019-11-21-what-is-amqp-and-why-is-it-used-in-rabbitmq.html>

Δυνατότητες ανταλλαγής μηνυμάτων

Το πρωτόκολλο σύνδεσης μεταφέρει μηνύματα μεταξύ δύο κόμβων, αλλά προϋποθέτει ελάχιστα ως προς το ποιοι είναι αυτοί οι κόμβοι ή πώς υλοποιούνται.

Μια βασική κατηγορία είναι οι κόμβοι που χρησιμοποιούνται ως σημείο συνάντησης μεταξύ αποστολέων και παραληπτών μηνυμάτων (π.χ. ουρές ή θέματα). Η προδιαγραφή AMQP αποκαλεί τέτοιους κόμβους κόμβους διανομής και κωδικοποιεί ορισμένες κοινές συμπεριφορές.



Σχήμα 2.33: Στοιχεία AMQP

Πηγή: <https://jstobigdata.com/rabbitmq/elements-of-amqp/>

Αυτό περιλαμβάνει:

- κάποια τυποποιημένα αποτελέσματα για τις μεταφορές, μέσω των οποίων οι παραλήπτες των μηνυμάτων μπορούν για παράδειγμα να αποδέχονται ή να απορρίπτουν τα μηνύματα.

- έναν μηχανισμό για την ένδειξη ή την αίτηση ενός από τα δύο βασικά πρότυπα διανομής, των ανταγωνιστικών και μη ανταγωνιστικών καταναλωτών, μέσω των τρόπων διανομής *move* και *copy* αντίστοιχα.

- τη δυνατότητα δημιουργίας κόμβων κατά παραγγελία, π.χ. για προσωρινές ουρές απάντησης

- τη δυνατότητα βελτίωσης του συνόλου των μηνυμάτων που ενδιαφέρουν έναν παραλήπτη μέσω φίλτρων.

Αν και το AMQP μπορεί να χρησιμοποιηθεί σε απλά συστήματα ομότιμων χρηστών, ο ορισμός αυτού του πλαισίου για τις δυνατότητες ανταλλαγής μηνυμάτων επιτρέπει επιπλέον τη διαλειτουργικότητα με ενδιαμέσους φορείς ανταλλαγής μηνυμάτων (μεσίτες, γέφυρες κ.λπ.) σε μεγαλύτερα, πλουσιότερα δίκτυα ανταλλαγής μηνυμάτων. Το καθορισμένο πλαίσιο καλύπτει τις

βασικές συμπεριφορές, αλλά επιτρέπει την ανάπτυξη επεκτάσεων που μπορούν να κωδικοποιηθούν και να τυποποιηθούν περαιτέρω.

2.2.4.3 MQTT



Σχήμα 2.34: MQTT logo

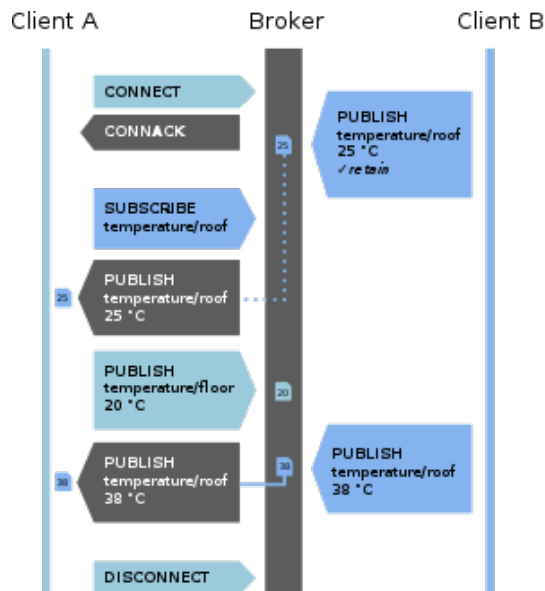
Πηγή: <https://mqtt.org/mqtt-specification/>

Το MQTT (Message Queuing Telemetry Transport) είναι ένα ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων για χρήση σε περιπτώσεις όπου οι πελάτες χρειάζονται μικρό αποτύπωμα κώδικα και είναι συνδεδεμένοι σε αναξιόπιστα δίκτυα ή δίκτυα με περιορισμένο εύρος ζώνης. Χρησιμοποιείται κυρίως για την επικοινωνία μεταξύ μηχανών (M2M) ή για συνδέσεις τύπου Internet of Things.

Το MQTT είναι ένα ελαφρύ, publish-subscribe πρωτόκολλο δικτύου που μεταφέρει μηνύματα μεταξύ συσκευών. Το πρωτόκολλο εκτελείται συνήθως μέσω TCP/IP, ωστόσο, οποιοδήποτε πρωτόκολλο δικτύου που παρέχει διατεταγμένες, χωρίς απώλειες, αμφίδρομες συνδέσεις μπορεί να υποστηρίξει το MQTT. Έχει σχεδιαστεί για συνδέσεις με απομακρυσμένες τοποθεσίες όπου υπάρχουν περιορισμοί πόρων ή το εύρος ζώνης του δικτύου είναι περιορισμένο. Το πρωτόκολλο είναι ανοικτό πρότυπο OASIS και σύσταση ISO (ISO/IEC 20922).

Το πρωτόκολλο MQTT ορίζει δύο τύπους οντοτήτων δικτύου: έναν μεσίτη μηνυμάτων και έναν αριθμό πελατών. Ένας μεσίτης MQTT είναι ένας διακομιστής που λαμβάνει όλα τα μηνύματα από τους πελάτες και στη συνέχεια δρομολογεί τα μηνύματα στους κατάλληλους πελάτες προορισμού[16].[17] Ένας πελάτης MQTT είναι οποιαδήποτε συσκευή (από έναν μικροελεγκτή μέχρι έναν πλήρως εξοπλισμένο διακομιστή) που εκτελεί μια βιβλιοθήκη MQTT και συνδέεται σε έναν μεσίτη MQTT μέσω δικτύου[17].

Οι πληροφορίες οργανώνονται σε μια ιεραρχία θεμάτων. Όταν ένας εκδότης έχει ένα νέο στοιχείο δεδομένων προς διανομή, στέλνει ένα μήνυμα ελέγχου με τα δεδομένα στον συνδεδεμένο μεσίτη. Στη συνέχεια, ο μεσίτης διανέμει τις πληροφορίες σε όλους τους πελάτες που έχουν εγγραφεί στο συγκεκριμένο θέμα. Ο εκδότης δεν χρειάζεται να έχει δεδομένα σχετικά με τον αριθμό ή τις θέσεις των συνδρομητών και οι συνδρομητές, με τη σειρά τους, δεν χρειάζεται να έχουν διαμορφωθεί με δεδομένα σχετικά με τους εκδότες.



Σχήμα 2.35: Ανταλλαγή μηνυμάτων στο MQTT

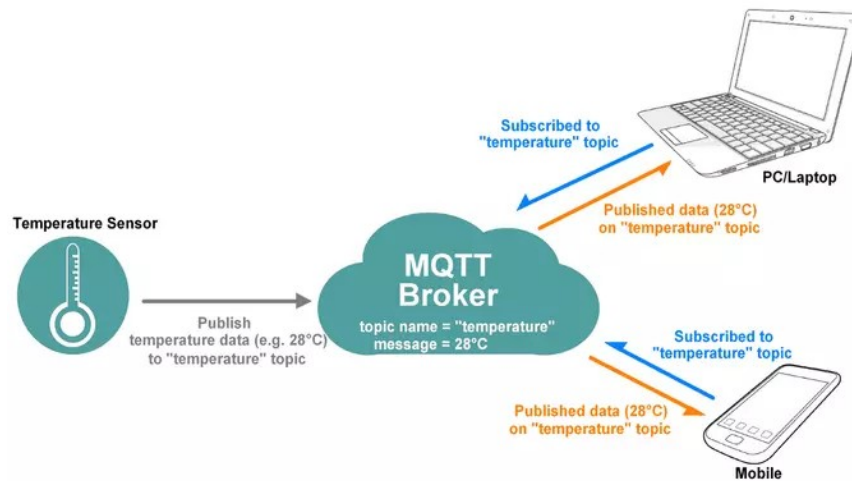
Πηγή: <https://en.wikipedia.org/wiki/MQTT>

Εάν ένας διαμεσολαβητής λάβει ένα μήνυμα για ένα θέμα για το οποίο δεν υπάρχουν τρέχουσες συνδρομές, ο διαμεσολαβητής απορρίπτει το μήνυμα, εκτός εάν ο εκδότης του μηνύματος έχει ορίσει το μήνυμα ως διατηρούμενο μήνυμα. Ένα διατηρημένο μήνυμα είναι ένα κανονικό μήνυμα MQTT με τη σημαία *retained* να έχει τεθεί σε *true*. Ο διαμεσολαβητής αποθηκεύει το τελευταίο διατηρημένο μήνυμα και το αντίστοιχο QoS για το επιλεγμένο θέμα. Κάθε πελάτης που εγγράφεται σε ένα μοτίβο θέματος που ταιριάζει με το θέμα του διατηρούμενου μηνύματος λαμβάνει το διατηρούμενο μήνυμα αμέσως μετά την εγγραφή του. Ο μεσίτης αποθηκεύει μόνο ένα διατηρημένο μήνυμα ανά θέμα [24]. Αυτό επιτρέπει στους νέους συνδρομητές ενός θέματος να λαμβάνουν την πιο πρόσφατη τιμή αντί να περιμένουν την επόμενη ενημέρωση από έναν εκδότη.

Όταν ένας πελάτης δημοσίευσης συνδέεται για πρώτη φορά με τον μεσίτη, μπορεί να ρυθμίσει ένα προεπιλεγμένο μήνυμα που θα αποστέλλεται στους συνδρομητές εάν ο μεσίτης ανιχνεύσει ότι ο πελάτης δημοσίευσης έχει απροσδόκητα αποσυνδεθεί από τον μεσίτη.

Οι πελάτες αλληλεπιδρούν μόνο με έναν διαμεσολαβητή, αλλά ένα σύστημα μπορεί να περιέχει πολλούς διακομιστές διαμεσολαβητών που ανταλλάσσουν δεδομένα με βάση τα τρέχοντα θέματα των συνδρομητών τους.

Το MQTT εκτελείται πάνω στο TCP/IP χρησιμοποιώντας μια τοπολογία PUSH/SUBSCRIBE. Στην αρχιτεκτονική του MQTT, υπάρχουν δύο τύποι συστημάτων: οι πελάτες και οι μεσίτες. Ο μεσίτης είναι ο διακομιστής με τον οποίο επικοινωνούν οι πελάτες. Ο μεσίτης λαμβάνει επικοινωνίες από τους πελάτες και στέλνει αυτές τις επικοινωνίες σε άλλους πελάτες. Οι πελάτες δεν επικοινωνούν απευθείας μεταξύ τους, αλλά συνδέονται με τον μεσίτη. Κάθε πελάτης μπορεί να είναι είτε εκδότης, είτε συνδρομητής, είτε και τα δύο.



Σχήμα 2.36: Χρήση του MQTT για την επικοινωνία του IoT

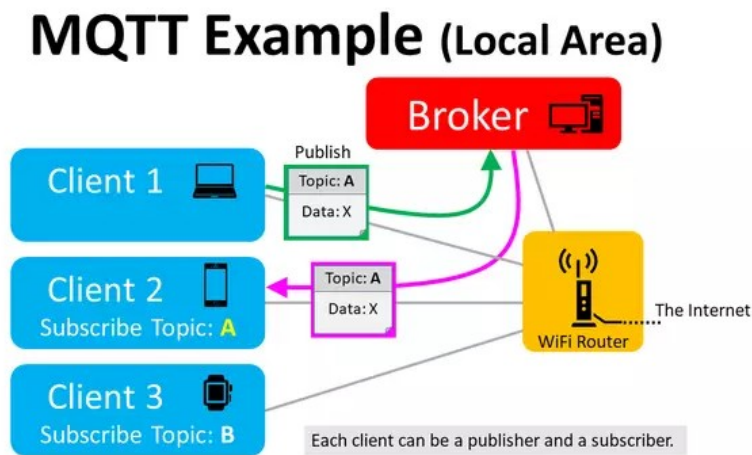
Πηγή: <https://medium.com/@jaydev.dave93/what-is-mqtt-protocol-c6a0cafffa8c>

Το MQTT είναι ένα πρωτόκολλο με γνώμονα τα συμβάντα. Δεν υπάρχει περιοδική ή συνεχής μετάδοση δεδομένων. Αυτό διατηρεί τη μετάδοση στο ελάχιστο. Ένας πελάτης δημοσιεύει μόνο όταν υπάρχουν πληροφορίες προς αποστολή, και ένας μεσίτης στέλνει πληροφορίες στους συνδρομητές μόνο όταν φθάνουν νέα δεδομένα.

Τα μηνύματα στο MQTT δημοσιεύονται ως θέματα. Τα θέματα είναι δομές σε μια ιεραρχία που χρησιμοποιεί τον χαρακτήρα slash (/) ως διαχωριστικό. Αυτή η δομή μοιάζει με αυτή ενός δέντρου καταλόγου σε ένα σύστημα αρχείων υπολογιστή. Μια δομή όπως sensors/OilandGas/Pressure/ επιτρέπει σε έναν συνδρομητή να καθορίσει ότι θα πρέπει να του αποστέλλονται δεδομένα μόνο από πελάτες που δημοσιεύουν στο θέμα Pressure, ή για μια ευρύτερη άποψη, ίσως όλα τα δεδομένα από πελάτες που δημοσιεύουν σε οποιοδήποτε θέμα sensors/OilandGas. Τα θέματα δεν δημιουργούνται ρητά στο MQTT. Εάν ένας διαμεσολαβητής λαμβάνει δεδομένα που δημοσιεύονται σε ένα θέμα που

δεν υπάρχει επί του παρόντος, το θέμα απλώς δημιουργείται και οι πελάτες μπορούν να εγγραφούν στο νέο θέμα.

Όταν οι επικοινωνίες είναι αναξιόπιστες, είναι πιθανό ένας εκδότης να αποσυνδεθεί από το δίκτυο χωρίς προειδοποίηση. Ένας εκδότης μπορεί να καταχωρίσει ένα μήνυμα που θα αποστέλλεται στους συνδρομητές σε περίπτωση που ο εκδότης αποσυνδεθεί απροσδόκητα. Αυτό το μήνυμα αποθηκεύεται στην προσωρινή μνήμη του διαμεσολαβητή και αποστέλλεται στους συνδρομητές σε περίπτωση που ο εκδότης αποσυνδεθεί αντικανονικά. Συνήθως, ένα τέτοιο μήνυμα περιλαμβάνει πληροφορίες που επιτρέπουν τον εντοπισμό του αποσυνδεδεμένου εκδότη, ώστε να μπορούν να γίνουν οι κατάλληλες ενέργειες.



Σχήμα 2.37: Παράδειγμα χρήσης MQTT σε τοπικό επίπεδο

Πηγή: <https://take6shin.blogspot.com/2019/07/lets-try-mqtt-communication.html>

Ο αρχικός στόχος του πρωτοκόλλου MQTT ήταν η μικρότερη δυνατή και αποδοτικότερη μετάδοση δεδομένων μέσω ακριβών και αναξιόπιστων γραμμών επικοινωνίας. Ως εκ τούτου, η ασφάλεια δεν αποτελούσε πρωταρχικό μέλημα κατά τον σχεδιασμό και την υλοποίηση του MQTT.

Ωστόσο, υπάρχουν ορισμένες επιλογές ασφάλειας με κόστος τη μεγαλύτερη μετάδοση δεδομένων και μεγαλύτερο αποτύπωμα.

Ασφάλεια δικτύου - Εάν το ίδιο το δίκτυο μπορεί να ασφαλιστεί, τότε η μετάδοση μη ασφαλών δεδομένων MQTT είναι άνευ σημασίας. Σε αυτή την περίπτωση, τα ζητήματα ασφάλειας θα πρέπει να προκύψουν από το εσωτερικό του ίδιου του δικτύου, ίσως μέσω ενός κακού παράγοντα ή άλλης μορφής διείσδυσης στο δίκτυο.

Όνομα χρήστη και κωδικός πρόσβασης - Το MQTT επιτρέπει ονόματα χρήστη και κωδικούς πρόσβασης για έναν πελάτη να εγκαθιδρύσει μια σύνδεση με έναν διαμεσολαβητή. Δυστυχώς, προκειμένου να διατηρηθεί η ελαφρότητα των επικαλύψεων, τα ονόματα χρήστη και οι κωδικοί πρόσβασης μεταδίδονται σε καθαρό κείμενο. Το 1999, αυτό ήταν κάτι παραπάνω από αρκετό, διότι η υποκλοπή μιας δορυφορικής επικοινωνίας για μια ουσιαστικά ασήμαντη ένδειξη αισθητήρα θα ήταν απαγορευτικά δύσκολη. Ωστόσο, σήμερα, η υποκλοπή πολλών τύπων επικοινωνιών ασύρματων δικτύων είναι ασήμαντη, καθιστώντας μια τέτοια πιστοποίηση σχεδόν άχρηστη.

Πολλές περιπτώσεις χρήσης απαιτούν ένα όνομα χρήστη και έναν κωδικό πρόσβασης όχι ως προστασία από κακόπιστους παράγοντες, αλλά ως τρόπο αποφυγής ακούσιων συνδέσεων.

SSL/TLS - Λειτουργώντας πάνω από το TCP/IP, η προφανής λύση για την ασφάλεια των μεταδόσεων μεταξύ πελατών και διαμεσολαβητών είναι η εφαρμογή του SSL/TLS. Δυστυχώς, αυτό προσθέτει σημαντική επιβάρυνση στις κατά τα άλλα ελαφριές επικοινωνίες.

Μεσίτης MQTT

Ο MQTT broker είναι ένα κομμάτι λογισμικού που εκτελείται σε έναν υπολογιστή (που λειτουργεί στις εγκαταστάσεις ή στο cloud) και μπορεί να είναι αυτοδημιούργητο ή να φιλοξενείται από τρίτο μέρος. Διατίθεται τόσο σε υλοποιήσεις ανοικτού κώδικα όσο και σε ιδιόκτητες υλοποιήσεις.

Ο διαμεσολαβητής λειτουργεί ως ταχυδρομείο. Οι πελάτες MQTT δεν χρησιμοποιούν απευθείας διεύθυνση σύνδεσης του προοριζόμενου παραλήπτη, αλλά χρησιμοποιούν τη γραμμή θέματος που ονομάζεται "Θέμα". Όποιος εγγράφεται λαμβάνει αντίγραφο όλων των μηνυμάτων για το συγκεκριμένο θέμα. Πολλοί πελάτες μπορούν να εγγραφούν σε ένα θέμα από έναν μόνο μεσίτη (δυνατότητα ένας προς πολλούς) και ένας μόνο πελάτης μπορεί να εγγράψει εγγραφές σε θέματα με πολλούς μεσίτες (πολλοί προς έναν).

Κάθε πελάτης μπορεί τόσο να παράγει όσο και να λαμβάνει δεδομένα τόσο με τη δημοσίευση όσο και με την εγγραφή, δηλαδή οι συσκευές μπορούν να δημοσιεύουν δεδομένα αισθητήρων και να είναι σε θέση να λαμβάνουν τις πληροφορίες διαμόρφωσης ή τις εντολές ελέγχου (το MQTT είναι ένα πρωτόκολλο αμφίδρομης επικοινωνίας). Αυτό βοηθά τόσο στην ανταλλαγή δεδομένων, όσο και στη διαχείριση και τον έλεγχο των συσκευών. Ένας πελάτης δεν μπορεί να μεταδώσει τα ίδια δεδομένα σε μια σειρά από θέματα και πρέπει να δημοσιεύσει πολλαπλά μηνύματα στον μεσίτη, το καθένα με ένα μόνο θέμα που δίνεται.

Με την αρχιτεκτονική του μεσίτη MQTT, οι συσκευές-πελάτες και η εφαρμογή διακομιστή αποσυνδέονται. Με αυτόν τον τρόπο, οι πελάτες διατηρούνται σε άγνοια για τις πληροφορίες του άλλου. Το MQTT, εάν έχει ρυθμιστεί έτσι, μπορεί να χρησιμοποιήσει κρυπτογράφηση TLS με

πιστοποιητικά, όνομα χρήστη και κωδικό πρόσβασης για συνδέσεις που προστατεύονται. Προαιρετικά, η σύνδεση μπορεί να απαιτεί πιστοποίηση, με τη μορφή ενός αρχείου πιστοποιητικού που παρέχει ο πελάτης και πρέπει να ταιριάζει με το αντίγραφο του διακομιστή.

Σε περίπτωση αποτυχίας, το λογισμικό διαμεσολαβητή και οι πελάτες μπορούν να παραδίδουν αυτόματα σε έναν πλεονάζοντα/αυτόματο εφεδρικό διαμεσολαβητή. Οι εφεδρικοί μεσίτες μπορούν επίσης να ρυθμιστούν ώστε να μοιράζονται το φορτίο των πελατών σε πολλούς διακομιστές στο χώρο, στο νέφος ή σε συνδυασμό αυτών.

Ο μεσίτης μπορεί να υποστηρίζει τόσο το τυπικό MQTT όσο και το MQTT για συμβατές προδιαγραφές, όπως το Sparkplug[25].

Ο διαμεσολαβητής παρακολουθεί όλες τις πληροφορίες της συνεδρίας καθώς η συσκευή ενεργοποιείται και απενεργοποιείται, σε μια λειτουργία που ονομάζεται "μόνιμες συνεδρίες". Σε αυτή την κατάσταση, ο μεσίτης θα αποθηκεύσει τόσο τις πληροφορίες σύνδεσης για κάθε πελάτη, τα θέματα στα οποία κάθε πελάτης έχει εγγραφεί, όσο και τυχόν μηνύματα για ένα θέμα με QoS 1 ή 2.

Τα κύρια πλεονεκτήματα του μεσίτη MQTT είναι:

1. Εξαλείφει τις ευάλωτες και ανασφαλείς συνδέσεις πελατών, εάν έχει ρυθμιστεί έτσι
2. Μπορεί εύκολα να κλιμακωθεί από μία συσκευή σε χιλιάδες
3. Διαχειρίζεται και παρακολουθεί όλες τις καταστάσεις σύνδεσης πελάτη, συμπεριλαμβανομένων των διαπιστευτηρίων ασφαλείας και των πιστοποιητικών, εάν έχει ρυθμιστεί έτσι
4. Μειωμένη επιβάρυνση του δικτύου χωρίς να διακυβεύεται η ασφάλεια, εάν ρυθμιστεί σε (κυψελοειδές ή δορυφορικό δίκτυο)

Τύποι μηνυμάτων

Connect

Περιμένει να δημιουργηθεί μια σύνδεση με τον διακομιστή και δημιουργεί μια σύνδεση μεταξύ των κόμβων.

Disconnect

Περιμένει να τελειώσει ο πελάτης MQTT τις εργασίες που πρέπει να κάνει και να αποσυνδεθεί η σύνδεση TCP/IP.

Publish

Επιστρέφει αμέσως στο νήμα της εφαρμογής μετά τη διαβίβαση της αίτησης στον πελάτη MQTT.

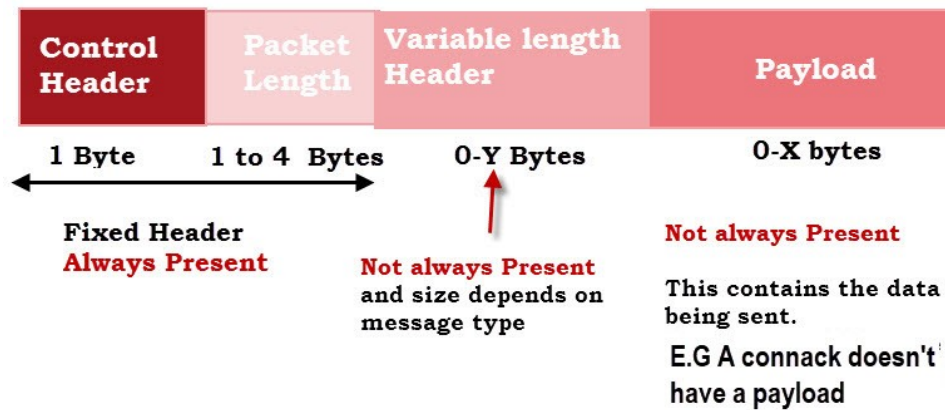
Το MQTT χρησιμοποιεί μια μορφή εντολής και επιβεβαίωσης εντολής.

Αυτό σημαίνει ότι κάθε εντολή έχει μια σχετική επιβεβαίωση.

Τα ονόματα θεμάτων, το αναγνωριστικό πελάτη, τα ονόματα χρηστών και οι κωδικοί πρόσβασης κωδικοποιούνται ως συμβολοσειρές UTF-8.

Το ωφέλιμο φορτίο, εξαιρουμένων των πληροφοριών του πρωτοκόλλου MQTT, όπως το αναγνωριστικό πελάτη κ.λπ., είναι δυαδικά δεδομένα και το περιεχόμενο και η μορφή τους είναι ειδικά για κάθε εφαρμογή.

Η μορφή του πακέτου ή του μηνύματος MQTT αποτελείται από μια σταθερή επικεφαλίδα 2 byte (υπάρχει πάντα) + μεταβλητή επικεφαλίδα (δεν υπάρχει πάντα)+ ωφέλιμο φορτίο (δεν υπάρχει πάντα).



MQTT Standard Packet Structure

Σχήμα 2.38: Δομή πακέτου MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Οι πιθανές μορφές πακέτων είναι:

Σταθερή επικεφαλίδα (πεδίο ελέγχου + μήκος) - Παράδειγμα CONNACK

Σταθερή επικεφαλίδα (πεδίο ελέγχου + μήκος) + μεταβλητή επικεφαλίδα - Παράδειγμα PUBACK

Σταθερή επικεφαλίδα (πεδίο ελέγχου + μήκος) + μεταβλητή επικεφαλίδα + ωφέλιμο φορτίο - Παράδειγμα CONNECT

Το πεδίο σταθερής επικεφαλίδας αποτελείται από το πεδίο ελέγχου και το πεδίο μεταβλητού μήκους μήκους του πακέτου.

Το ελάχιστο μέγεθος του πεδίου μήκους πακέτου είναι 1 byte, το οποίο ισχύει για μηνύματα με συνολικό μήκος μικρότερο από 127 bytes (χωρίς τα πεδία ελέγχου και μήκους).

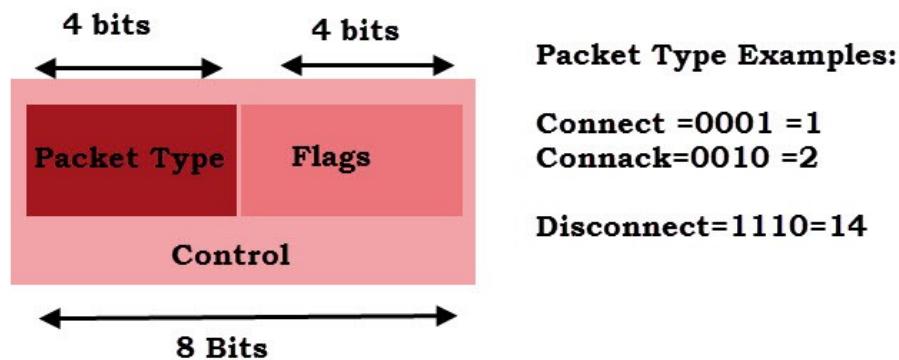
Το μέγιστο μέγεθος πακέτου είναι 256MB. Μικρά πακέτα μικρότερα από 127 bytes έχουν πεδίο μήκους πακέτου 1 byte.

Πακέτα μεγαλύτερα από 127 και μικρότερα από 16383 θα χρησιμοποιούν 2 bytes. κ.λπ.

Σημείωση: Χρησιμοποιούνται 7 bit με το 8ο bit να είναι ένα bit συνέχισης.

Το ελάχιστο μέγεθος πακέτου είναι μόλις 2 bytes με ένα πεδίο ελέγχου ενός byte και ένα πεδίο μήκους πακέτου ενός byte. Π.χ. το μήνυμα αποσύνδεσης είναι μόνο 2 bytes.

Πεδίο Ελέγχου



MQTT Control Field Structure

Σχήμα 2.39: Δομή πεδίου ελέγχου MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Το πεδίο ελέγχου των 8 bit είναι το πρώτο byte της σταθερής επικεφαλίδας των 2 byte. Χωρίζεται σε δύο πεδία των 4 bit και περιέχει όλες τις εντολές και τις απαντήσεις του πρωτοκόλλου.

Τα 4 πρώτα πιο σημαντικά bit είναι το πεδίο τύπου εντολής ή μηνύματος και τα άλλα 4 bit χρησιμοποιούνται ως σημαίες ελέγχου.

Σημαίες ελέγχου

Αν και υπάρχουν 16 πιθανές σημαίες, πολύ λίγες χρησιμοποιούνται στην πραγματικότητα.

Το μήνυμα δημοσίευσης κάνει τη μεγαλύτερη χρήση αυτών των σημαιών, όπως φαίνεται στον παρακάτω πίνακα:

MQTT Control Flags (Partial)

Table 2.2 - Flag Bits

Control Packet	Fixed header flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP ¹	QoS ²	QoS ²	RETAIN ³
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0

Duplicate message

Quality of Service
00,01,10 =QOS 0,1,2

Retain Message

Note: taken from the OASIS MQTT 3.1.1 specification document

Σχήμα 2.40: Σημαίες ελέγχου στο MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Η σημαία duplicate χρησιμοποιείται κατά την εκ νέου δημοσίευση ενός μηνύματος με QOS ή 1 ή 2.

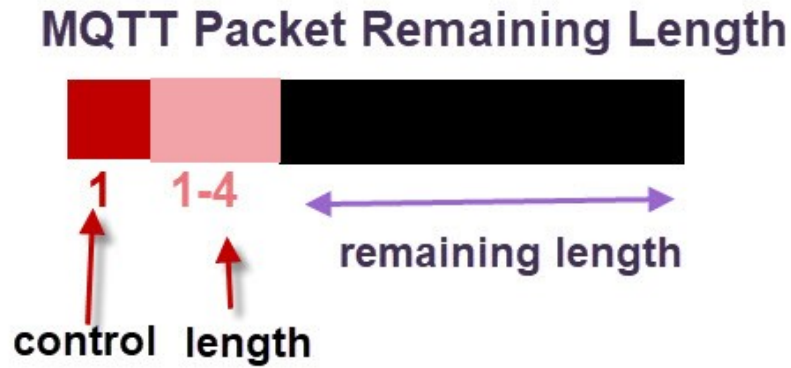
Οι σημαίες QOS χρησιμοποιούνται κατά τη δημοσίευση και υποδεικνύουν το επίπεδο QOS -0,1,2

Η σημαία διατήρησης μηνύματος χρησιμοποιείται επίσης κατά τη δημοσίευση.

Υπολειπόμενο μήκος

Είναι μεταβλητού μήκους μεταξύ 1 και 4 bytes. Κάθε byte χρησιμοποιεί 7 bits για το μήκος με το MSB να χρησιμοποιείται ως σημαία συνέχισης.

Το εναπομένον μήκος είναι ο αριθμός των bytes που ακολουθούν το πεδίο μήκους, περιλαμβάνει επικεφαλίδα μεταβλητού μήκους και ωφέλιμο φορτίο, όπως απεικονίζεται παρακάτω:

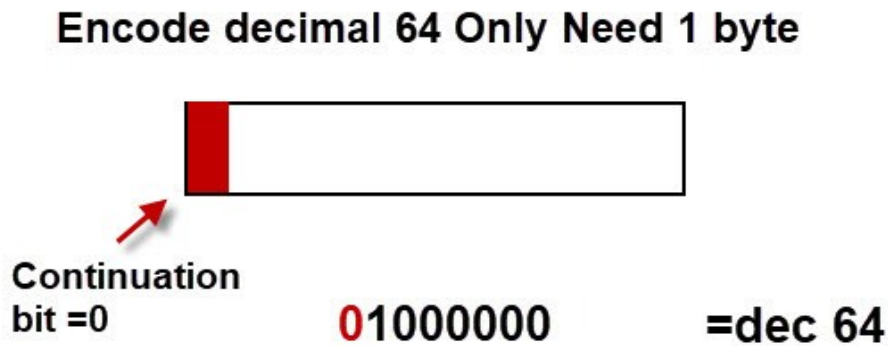


Σχήμα 2.41: Υπολοιπό μήκος πακέτου στο MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Ακολουθεί η απεικόνιση του πεδίου μήκους για μέγεθος πακέτου 64 και 321 bytes

Υπόλοιπο μήκος πακέτου 64 bytes του απαιτεί μόνο 1 byte:

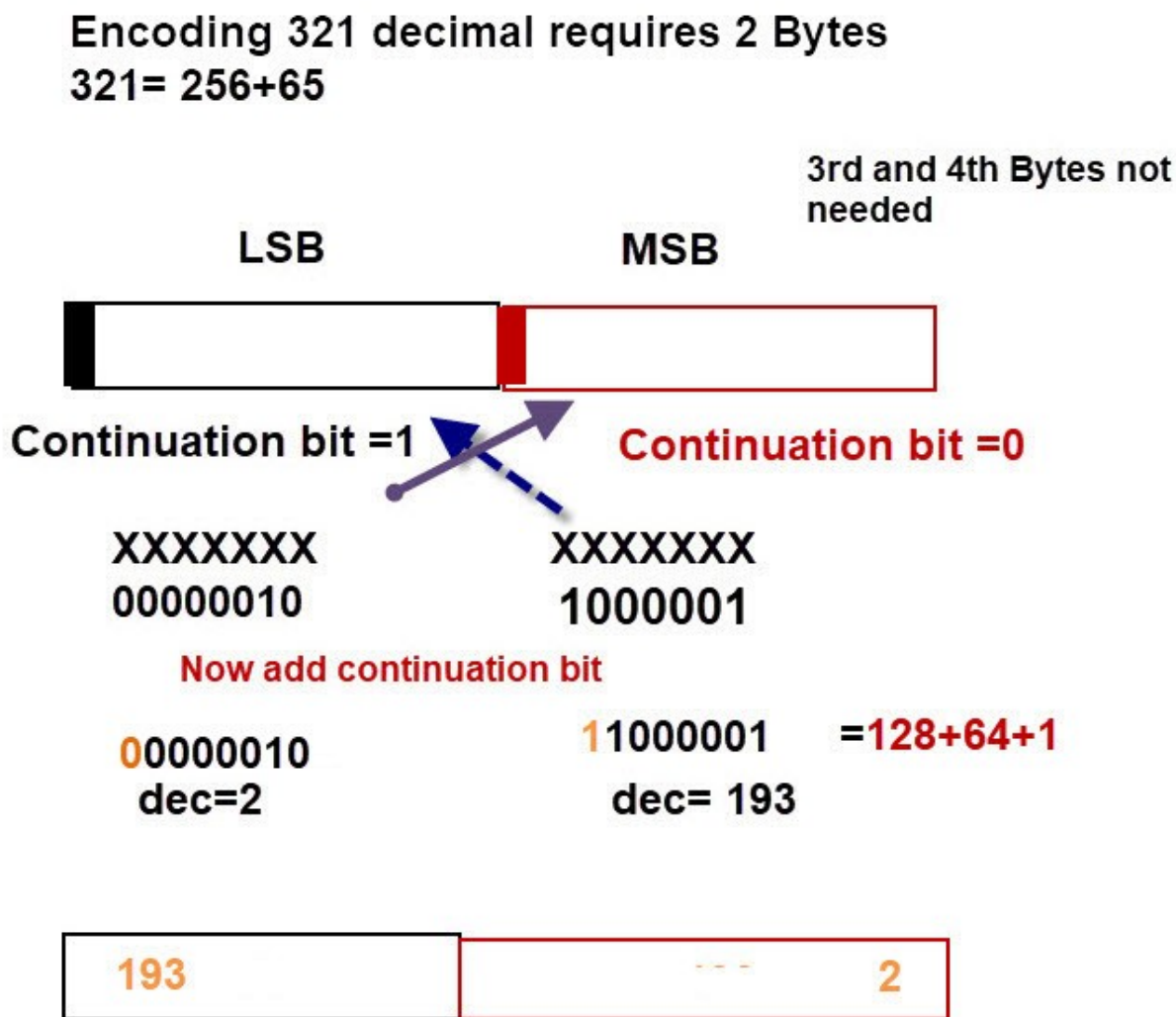


Encoding Remain Length Field Example 1

Σχήμα 2.42: Παράδειγμα 1 πεδίου υπολοιπούμενου μήκους στο MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Μήκος πακέτου 321 bytes απαιτεί ένα πεδίο μήκους 2 bytes:



Encoding Remain Length Field Example 2

Σχήμα 2.43: Παράδειγμα 2 πεδίου υπολοιπούμενου μήκους στο MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Κεφαλίδα μεταβλητού μήκους

Όπως αναφέρθηκε προηγουμένως, το πεδίο επικεφαλίδας μεταβλητού μήκους δεν είναι πάντα παρόν σε ένα μήνυμα MQTT.

Ορισμένοι τύποι μηνυμάτων MQTT ή εντολές απαιτούν τη χρήση αυτού του πεδίου για τη μεταφορά πρόσθετων πληροφοριών ελέγχου.

Το πεδίο επικεφαλίδας μεταβλητού μήκους είναι παρόμοιο, αλλά όχι το ίδιο για όλους τους τύπους μηνυμάτων.

Παράδειγμα μηνύματος σύνδεσης και αποσύνδεσης MQTT

Ως παράδειγμα θα εξετάσουμε τώρα τις λεπτομέρειες του πακέτου για ένα μήνυμα σύνδεσης.

Παρακάτω παρουσιάζεται ένα πραγματικό παράδειγμα σύνδεσης και αποσύνδεσης πελάτη που δείχνει τις πραγματικές τιμές byte για τα δεδομένα που αποστέλλονται και λαμβάνονται.

Ο κωδικός ελέγχου CONNECT =0x10

Ο κωδικός ελέγχου CONNACK =0x20

Πακέτο MQTT =έλεγχος + μήκος + όνομα πρωτοκόλλου + επίπεδο πρωτοκόλλου + σημαίες σύνδεσης + keep alive +Payload

```
>>>
connecting client = python_test clean session = True
sending command 0x10 sending flags = 0
sending bytearray(b'\x10\x17\x00\x04MQTT\x04\x02\x00<\x00
\x0bpython_test')
received command 0x20 flags binary 0b0
received data b' \x02\x00\x00'
received data decimal [32, 2, 0, 0]

connected
disconnecting
sending command 0xe0 sending flags = 0
sending b'\xe0\x00'
>>> |
```

Total message length =23 bytes

length=2

Simple 2 byte Disconnect

MQTT Message Example - 1

Σχήμα 2.43: Παράδειγμα μηνύματος MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Σημειώσεις:

- Παρατηρήστε τους κωδικούς ελέγχου σύνδεσης (0x10) και επιβεβαίωσης σύνδεσης (0x20).
- Παρατηρήστε το συνολικό μήκος hex17 ή 23 bytes χωρίς να περιλαμβάνονται τα πεδία ελέγχου και μήκους. Το πεδίο μήκους είναι μόνο 1 byte.
- Θα πρέπει επίσης να μπορείτε να δείτε το αναγνωριστικό πελάτη (python_test) στο πακέτο που αποστέλλεται.
- Όταν εξετάζετε τα πραγματικά bytes του πακέτου, η Python εκτυπώνει τις δεκαεξαδικές τιμές εκτός αν μπορεί να ταιριάξει με έναν χαρακτήρα ASCII. στο παραπάνω παράδειγμα το πεδίο keep alive είναι x00x3C αλλά εμφανίζεται ως x00<. Όπου Ascii < =0x3C

MQTT Connect Message Structure

Connect Clean Session True Client ID =PYTON1

Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Meaning	Header	Remaining Length	Length of protocol name		Protocol Name +Version					Connect Flags	Keep Alive	Length									
Hex	0x10	0x13	0x0	0x4	0x4d	0x51	0x54	0x54	0x4	0x2	0x0	0x3c	0x0	0x7	0x70	0x79	0x74	0x68	0x6F	6E	0x31
Ascii		19		4	M	Q	T	T	4			60		7	P	Y	T	H	O	N	1

Notes:

Remaining Length = bytes 3 to 14
 Length of protocol name=4 =MQTT
 length in bytes 13-14 -payload length =7
 Connect Flags show Clean Session =True

Connect Flags

User name flag = bit 7
 Password Flag = bit 6
 Will Retain = bit 5
 Will QOS = bit 5
 Will QOS = bit 4
 Will Flag = bit 2
 Clean Session = bit 1
 Reserved = bit 0

Σχήμα 2.44:Δομή μηνύματος σύνδεσης στο MQTT

Πηγή: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>

Σημειώσεις:

- Το πεδίο ID πελάτη αποστέλλεται ως το πρώτο μέρος του ωφέλιμου φορτίου και όχι ως μέρος της επικεφαλίδας.
- Το αναγνωριστικό πελάτη ακολουθείται από ένα πεδίο μήκους

- Οι σημαίες Connect υποδεικνύουν ότι ζητείται καθαρή σύνδεση.
- Οι σημαίες σύνδεσης αποτελούν μέρος της επικεφαλίδας μεταβλητού μήκους και χρησιμοποιούνται για να υποδείξουν την παρουσία ή την απουσία των πεδίων username,password και will message στο ωφέλιμο φορτίο. Περιέχει επίσης τη σημαία καθαρής συνόδου και το Will QOS.

Σύνοψη πίνακα πακέτων ελέγχου

Πακέτο ελέγχου	Μεταβλητή Επικεφαλίδα	Ωφέλιμο φορτίο
CONNECT	Απαιτούμενο	Απαιτούμενο
CONNACK	Κανένα	Κανένα
PUBLISH	Απαιτούμενο	Προαιρετικό
PUBACK	Απαιτούμενο	Κανένα
PUBREC	Απαιτούμενο	Κανένα
PUBREL	Απαιτούμενο	Κανένα
PUBCOMP	Απαιτούμενο	Κανένα
SUBSCRIBE	Απαιτούμενο	Απαιτούμενο
SUBACK	Απαιτούμενο	Απαιτούμενο
UNSUBSCRIBE	Απαιτούμενο	Απαιτούμενο
UNSUBACK	Απαιτούμενο	Απαιτούμενο
PINGREQ	Κανένα	Κανένα
PINGRESP	Κανένα	Κανένα

DISCONNECT	Κανένα	Κανένα
------------	--------	--------

Επίπεδα QoS MQTT

Το MQTT έχει σχεδιάσει 3 επίπεδα QoS.

Το πολύ μία φορά (0)

Τουλάχιστον μία φορά (1)

Ακριβώς μία φορά (2)

Το QoS 0 είναι ένας τρόπος αποστολής μηνυμάτων "πυροβόλησε και ξέχνα": Αφού ένας αποστολέας (πιθανώς Εκδότης ή Διαμεσολαβητής) στείλει ένα μήνυμα, δεν ενδιαφέρεται πλέον για το αν θα σταλεί στο άλλο μέρος ή αν θα δημιουργήσει μηχανισμό επαναποστολής.

Ο QoS 1 περιλαμβάνει έναν απλό μηχανισμό επαναποστολής. Αφού ο αποστολέας στείλει ένα μήνυμα, περιμένει το ACK του παραλήπτη. Εάν δεν λάβει το ACK, στέλνει εκ νέου το μήνυμα. Αυτός ο τρόπος λειτουργίας μπορεί να εγγραφεί ότι το μήνυμα μπορεί να φτάσει τουλάχιστον μία φορά, αλλά δεν μπορεί να εγγραφεί ότι το μήνυμα επαναλαμβάνεται.

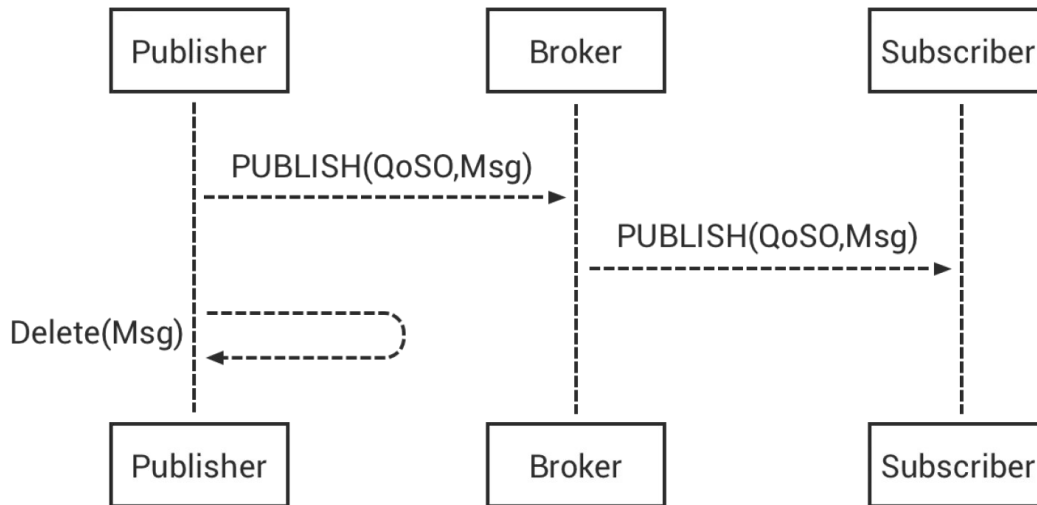
Το QoS 2 σχεδίασε έναν μηχανισμό ανακάλυψης επαναποστολής και επανάληψης μηνυμάτων για να διασφαλίσει ότι το μήνυμα θα φτάσει μόνο μία φορά.

QoS	Publisher	Subscriber
0: το πολύ μία φορά	Θα στείλει ένα μήνυμα μόνο μία φορά.	Μπορεί να λάβει ή να μην λάβει το μήνυμα.
1: τουλάχιστον μία φορά	Θα στείλει ένα μήνυμα τουλάχιστον μία φορά, εφόσον λάβει επιβεβαίωση ή λάβει την εντολή τερματισμού της μετάδοσης.	Είναι πιθανό να λάβει το μήνυμα τουλάχιστον μία φορά (είναι πιθανό το μήνυμα να ληφθεί περισσότερες από μία φορές).
2: ακριβώς μία φορά	Θα στείλει ένα μήνυμα μόνο μία φορά.	Θα λάβει το μήνυμα μόνο μία φορά.

QoS 0 - Δημοσίευση το πολύ μία φορά

Όταν το QoS είναι 0, η δημοσίευση των μηνυμάτων εξαρτάται από τις δυνατότητες του υποκείμενου δικτύου. Ο εκδότης θα δημοσιεύσει το μήνυμα μόνο μία φορά, ο παραλήπτης δεν θα απαντήσει στο μήνυμα και ο εκδότης δεν θα αποθηκεύσει και δεν θα ξαναστείλει το μήνυμα. Τα μηνύματα έχουν την υψηλότερη αποδοτικότητα μετάδοσης σε αυτό το επίπεδο, αλλά μπορεί να μην παραδοθούν μία φορά.

QoS 0:At most once(deliver and forgot)

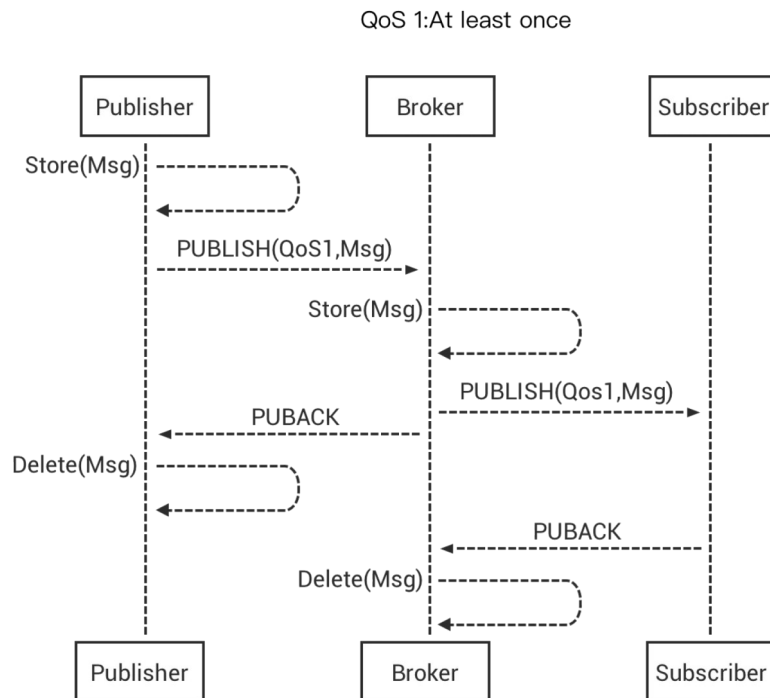


Σχήμα 2.45: MQTT QoS 0

Πηγή: <https://www.emqx.com/en/blog/introduction-to-mqtt-qos>

QoS 1 - Δημοσίευση τουλάχιστον μία φορά

Όταν το QoS είναι 1, το μήνυμα μπορεί να εγγραφεί ότι θα δημοσιευτεί τουλάχιστον μία φορά. Το MQTT εγγυάται το QoS 1 μέσω ενός απλού μηχανισμού ACK. Ο εκδότης θα δημοσιεύσει το μήνυμα και θα περιμένει την απάντηση του πακέτου PUBACK του παραλήπτη. Εάν η απάντηση PUBACK δεν ληφθεί εντός του καθορισμένου χρόνου, ο εκδότης θα θέσει το DUP του μηνύματος σε 1 και θα ξαναστείλει το μήνυμα. Ο παραλήπτης θα πρέπει να απαντήσει στο μήνυμα PUBACK όταν λαμβάνει ένα μήνυμα με QoS 1. Ο παραλήπτης μπορεί να δεχτεί το ίδιο μήνυμα πολλές φορές. Ανεξάρτητα από τη σημαία DUP, ο παραλήπτης θα αντιμετωπίσει το ληφθέν μήνυμα ως νέο μήνυμα και θα στείλει ένα πακέτο PUBACK ως απάντηση.



Σχήμα 2.46: MQTT QoS 1

Πηγή: <https://www.emqx.com/en/blog/introduction-to-mqtt-qos>

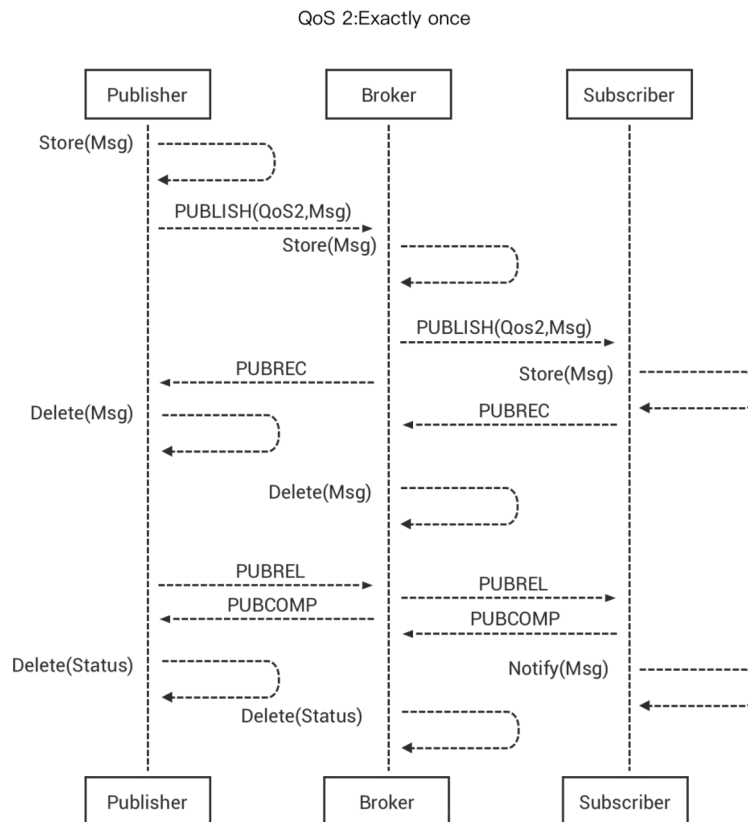
QoS 2 - Δημοσίευση μόνο μία φορά

Όταν το QoS είναι 2, οι εκδότες και οι συνδρομητές διασφαλίζουν ότι τα μηνύματα δημοσιεύονται μόνο μία φορά μέσω δύο συνόδων. Αυτό είναι το υψηλότερο επίπεδο ποιότητας υπηρεσίας και η απώλεια και η επανάληψη μηνυμάτων είναι απαράδεκτες. Η χρήση αυτού του επιπέδου ποιότητας υπηρεσίας έχει ένα πρόσθετο κόστος.

Αφού ο εκδότης δημοσιεύσει ένα μήνυμα με QoS 2, θα αποθηκεύσει το δημοσιευμένο μήνυμα και θα περιμένει τον παραλήπτη να απαντήσει με το μήνυμα PUBREC. Αφού ο εκδότης λάβει το μήνυμα PUBREC, μπορεί να απορρίψει με ασφάλεια το προηγουμένως δημοσιευμένο μήνυμα, επειδή γνωρίζει ήδη ότι ο παραλήπτης έλαβε επιτυχώς το μήνυμα. Ο εκδότης αποθηκεύει το μήνυμα PUBREC και απαντά με ένα PUBREL, περιμένοντας να απαντήσει ο παραλήπτης με το μήνυμα PUBCOMP. Όταν ο αποστολέας λάβει το μήνυμα PUBCOMP, θα διαγράψει την προηγουμένως αποθηκευμένη κατάσταση.

Όταν ο παραλήπτης λάβει ένα μήνυμα PUBLISH με QoS 2, επεξεργάζεται το μήνυμα και επιστρέφει ως απάντηση ένα PUBREC. Όταν ο παραλήπτης λάβει το μήνυμα PUBREL, απορρίπτει όλες τις αποθηκευμένες καταστάσεις και απαντά με PUBCOMP.

Κάθε φορά που συμβαίνει απώλεια πακέτων κατά τη μετάδοση, ο αποστολέας είναι υπεύθυνος για την επαναποστολή του προηγούμενου μηνύματος. Αυτό ισχύει ανεξάρτητα από το αν ο αποστολέας είναι εκδότης ή μεσίτης. Επομένως, ο παραλήπτης πρέπει επίσης να απαντά σε κάθε μήνυμα εντολής.



Σχήμα 2.47: MQTT QoS 2

Πηγή: <https://www.emqx.com/en/blog/introduction-to-mqtt-qos>

Η διαφορά του QoS στη δημοσίευση και τη συνδρομή

Η QoS των μηνυμάτων δημοσίευσης MQTT δεν είναι από άκρο σε άκρο, αλλά μεταξύ του πελάτη και του διακομιστή. Το επίπεδο QoS στο οποίο οι συνδρομητές λαμβάνουν τα μηνύματα MQTT εξαρτάται τελικά από το QoS του δημοσιευμένου μηνύματος και το QoS της συνδρομής του θέματος.

Όταν το QoS που χρησιμοποιείται από τον πελάτη A για τη δημοσίευση είναι μεγαλύτερο από το QoS που χρησιμοποιείται από τη συνδρομή του πελάτη B, το QoS του διακομιστή που προωθεί τα μηνύματα στον πελάτη B είναι το QoS που χρησιμοποιείται από τη συνδρομή του πελάτη B.

Όταν η QoS που χρησιμοποιείται από τον πελάτη A publish είναι μικρότερη από την QoS που χρησιμοποιείται από τη συνδρομή του πελάτη B, η QoS του διακομιστή που προωθεί μηνύματα στον πελάτη B είναι η QoS που χρησιμοποιείται από τη δημοσίευση του πελάτη A.

Μπορείτε να ανατρέξετε στον ακόλουθο πίνακα για το QoS των μηνυμάτων που έλαβε ο πελάτης σε διάφορες καταστάσεις:

QoS of publish	QoS of subscribe	QoS of received message
0	0	0
0	1	0
0	2	0
1	0	0
1	1	1
1	2	1
2	0	0
2	1	1
2	2	2

2.2.4.4 Η βασική διαφορά μεταξύ AMQP και MQTT

Τόσο το AMQP όσο και το MQTT χρησιμοποιούνται στο Διαδίκτυο των πραγμάτων. Ωστόσο, ας συζητήσουμε τις βασικές διαφορές:

- Το MQTT έχει αρχιτεκτονική πελάτη/μεσίτη, ενώ το AMQP έχει αρχιτεκτονική πελάτη ή μεσίτη και πελάτη ή διακομιστή.

- Το MQTT ακολουθεί την αφαίρεση της δημοσίευσης και της συνδρομής ενώ το AMQP ακολουθεί τις μεθόδους απόκρισης ή αίτησης και δημοσίευσης ή συνδρομής.

- Το μέγεθος της επικεφαλίδας του AMQP είναι 8bytes και του MQTT είναι 2bytes. Το μέγεθος του μηνύματος του MQTT είναι μικρό και καθορισμένο ενώ το AMQP έχει διαπραγματεύσιμο και απροσδιόριστο [26].

- Η μέθοδος του MQTT είναι connected, publish, close, subscribe και disconnect. Το AMQP ακολουθεί το Consume, deliver, publish, get, select, acknowledge, delete, recover, reject, open και close.

- Το MQTT έχει μερική υποστήριξη για cache και proxy ενώ το AMQP προσφέρεται πλήρης υποστήριξη.

- Τόσο το AMQP όσο και το MQTT ακολουθούν το πρωτόκολλο TCP, το δυαδικό πρότυπο και το σύστημα ουρών αναμονής ανοικτού κώδικα.

- Η ασφάλεια που προσφέρεται στο AMQP είναι IPSec, SASL, TLS ή SSL, ενώ το MQTT παρέχει μόνο τα πρότυπα ασφαλείας TLS ή SSL[27]. Το AMQP μαζί με το TCP χρησιμοποιεί το SCTP για σκοπούς μετάδοσης. Το OASIS υποστηρίζει τόσο το AMQP όσο και το MQTT.

2.2.5 Message Brokers

Μερικοί διαθέσιμοι message brokers είναι οι:

2.2.5.1 RabbitMQ



Σχήμα 2.48: RabbitMQ logo

Πηγή: <https://www.drouiz.com/blog/2018/10/19/rabbitmq-gestor-de-cola/>

Το RabbitMQ είναι ένας από τους πιο διαδεδομένους διαμεσολαβητές μηνυμάτων ανοικτού κώδικα. Αρχικά βασίστηκε στο Advanced Message Queuing Protocol (AMQP). Αργότερα, τροποποιήθηκε για να υποστηρίζει το Message Queuing Telemetry Transport (MQTT), το Streaming Text Oriented Messaging Protocol (STOMP) και πολλά άλλα κοινά πρωτόκολλα. Το RabbitMQ έχει σχεδιαστεί για να χρησιμοποιείται για μια ποικιλία σεναρίων ανταλλαγής μηνυμάτων που έχουν αναπτυχθεί τα τελευταία 25 χρόνια. Είναι γνωστό για την αξιοπιστία και τη διαλειτουργικότητα [28]. Έχει υλοποιηθεί σε πολλές γλώσσες- ως εκ τούτου, αποτελεί μία από τις καλύτερες επιλογές για τη δημιουργία αξιόπιστων, μεγάλης κλίμακας, ομαδοποιημένων, ανθεκτικών υποδομών ανταλλαγής μηνυμάτων.

Το RabbitMQ είναι ένας παραδοσιακός διαμεσολαβητής μηνυμάτων που υποστηρίζει πολλά πρωτόκολλα μηνυμάτων, όπως AMQP, MQTT και STOMP και ονομάζεται επίσης υβριδικός διαμεσολαβητής. Υποστηρίζει διάφορες παραλλαγές των τεχνικών ανταλλαγής μηνυμάτων pub-sub, points to point, request-reply. Χρησιμοποιεί ένα μοντέλο smart broker/dumb consumer και επικεντρώνεται στην παράδοση μηνυμάτων στους καταναλωτές με συνέπεια. Προσφέρει μεγάλες επιδόσεις αν ρυθμιστεί σωστά, υποστηρίζει βιβλιοθήκες πελατών σε Java, Ruby, .NET, PHP και πολλές άλλες γλώσσες και προσφέρει διάφορα πρόσθετα που μπορούν να προστεθούν για την επέκταση περιπτώσεων χρήσης και σεναρίων ολοκλήρωσης. Προσφέρει τόσο σύγχρονους όσο και ασύγχρονους τρόπους επικοινωνίας.

Το RabbitMQ υποστηρίζει μια ποικιλία επιβεβαιώσεων για διαφορετικές περιπτώσεις χρήσης και υποστηρίζει συναλλαγές σε ουρές ανταλλαγής μηνυμάτων. Επιτρέπει επίσης το διαχωρισμό των διαφόρων σημασιολογικών συναλλαγών (εάν απαιτείται). Επιτρέπει επίσης κατανεμημένες συναλλαγές όπως οι συναλλαγές X/Open XA. Υποστηρίζει επίσης την ομαδοποίηση μηνυμάτων και τα idempotent μηνύματα. Υποστηρίζει πολύ λεπτομερή έλεγχο όσον αφορά την πρόσβαση στις ουρές αναμονής. Μπορεί κανείς να περιορίσει την πρόσβαση σε ορισμένες ουρές, να διαχειριστεί το βάθος και πολλά άλλα.

2.2.5.2 HiveMQ



Σχήμα 2.49: HiveMQ logo

Πηγή: <https://www.codecentric.de/aktuelles/news/mqtt-iot-neue-partnerschaft-hivemq>

Το HiveMQ είναι ένας διαμεσολαβητής MQTT ειδικά προσαρμοσμένος για επιχειρήσεις, οι οποίες βρίσκονται στην αναδυόμενη εποχή της επικοινωνίας μεταξύ μηχανών (M2M) και του Διαδικτύου των πραγμάτων. Κατασκευάστηκε από την αρχή με γνώμονα τη μέγιστη επεκτασιμότητα και τις έννοιες ασφάλειας που είναι έτοιμες για επιχειρήσεις. Υλοποιεί το πρωτόκολλο MQTT, το de-facto πρότυπο ανταλλαγής μηνυμάτων M2M, και μέσω της 100% συμμόρφωσης με τις προδιαγραφές, κατέχει ηγετική θέση όσον αφορά την επαγγελματική υιοθέτηση όλων των δυνατοτήτων του Διαδικτύου των Πραγμάτων για τις επιχειρήσεις [29].

Ως μέλη της επιτροπής OASIS, η ομάδα της HiveMQ συμμετείχε άμεσα στη δημιουργία και την κυκλοφορία του MQTT 5.0, της νεότερης έκδοσης του ελαφρού πρωτοκόλλου. Είναι 100% συμβατό με το MQTT και υλοποιεί όλα τα χαρακτηριστικά του MQTT σε μεμονωμένες και ομαδοποιημένες εγκαταστάσεις.

2.2.5.3 Mosquitto MQTT



Σχήμα 2.50: Mosquitto MQTT logo

Πηγή: <https://idroot.us/install-mosquitto-mqtt-ubuntu-20-04/>

Το Eclipse Mosquitto είναι ένας διαμεσολαβητής μηνυμάτων ανοικτού κώδικα (με άδεια EPL/EDL) που υλοποιεί τις εκδόσεις 5.0, 3.1.1 και 3.1 του πρωτοκόλλου MQTT. Το Mosquitto είναι ελαφρύ και είναι κατάλληλο για χρήση σε όλες τις συσκευές, από υπολογιστές χαμηλής ισχύος με μία πλακέτα μέχρι πλήρεις διακομιστές.

Το πρωτόκολλο MQTT παρέχει μια ελαφριά μέθοδο για την εκτέλεση της ανταλλαγής μηνυμάτων με τη χρήση ενός μοντέλου δημοσίευσης/εγγραφής (publish/subscribe). Αυτό το καθιστά κατάλληλο για την ανταλλαγή μηνυμάτων στο Διαδίκτυο των πραγμάτων, όπως με αισθητήρες χαμηλής ισχύος ή κινητές συσκευές όπως τηλέφωνα, ενσωματωμένους υπολογιστές ή μικροελεγκτές [30].

Το έργο Mosquitto παρέχει επίσης μια βιβλιοθήκη C για την υλοποίηση πελατών MQTT, καθώς και τους πολύ δημοφιλείς πελάτες MQTT της γραμμής εντολών `mosquitto_pub` και `mosquitto_sub`.

Το Eclipse Mosquitto παρέχει μια ελαφριά υλοποίηση διακομιστή του πρωτοκόλλου MQTT που είναι κατάλληλη για όλες τις περιπτώσεις, από μηχανήματα πλήρους ισχύος έως ενσωματωμένα και μηχανήματα χαμηλής ισχύος. Οι αισθητήρες και οι ενεργοποιητές, οι οποίοι αποτελούν συχνά τις πηγές και τους προορισμούς των μηνυμάτων MQTT, μπορεί να είναι πολύ μικροί και να στερούνται ισχύος. Αυτό ισχύει και για τις ενσωματωμένες μηχανές στις οποίες συνδέονται, όπου θα μπορούσε να εκτελεστεί το Mosquitto.

Τυπικά, η τρέχουσα υλοποίηση του Mosquitto έχει ένα εκτελέσιμο αρχείο της τάξης των 120kB που καταναλώνει περίπου 3MB RAM με 1000 συνδεδεμένους πελάτες. Έχουν αναφερθεί επιτυχείς δοκιμές με 100.000 συνδεδεμένους πελάτες σε μέτριους ρυθμούς μηνυμάτων [31].

Εκτός από το να δέχεται συνδέσεις από εφαρμογές-πελάτες MQTT, το Mosquitto διαθέτει μια γέφυρα που του επιτρέπει να συνδέεται με άλλους διακομιστές MQTT, συμπεριλαμβανομένων άλλων περιπτώσεων του Mosquitto. Αυτό επιτρέπει την κατασκευή δικτύων διακομιστών MQTT, τα οποία περνούν μηνύματα MQTT από οποιαδήποτε θέση του δικτύου σε οποιαδήποτε άλλη, ανάλογα με τη διαμόρφωση των γεφυρών.

Το Mosquitto αποτελεί βασικό συστατικό του έργου Eclipse StreamSheets, το οποίο παρέχει μια εύχρηστη διεπαφή που μοιάζει με λογιστικό φύλλο πραγματικού χρόνου και επιτρέπει την επεξεργασία των εισερχόμενων δεδομένων από MQTT, OPC-UA, REST και άλλα πρωτόκολλα από μη προγραμματιστές για την παραγωγή πινάκων οργάνων, διαδικασιών ελέγχου ή σχεδόν οτιδήποτε απαιτεί είσοδο και έξοδο πολλαπλών πρωτοκόλλων.

2.2.6 ΕΦΑΡΜΟΓΗ ΔΙΑΔΡΑΣΤΙΚΗΣ ΟΠΤΙΚΟΠΟΙΗΣΗΣ

Στη συνέχεια θα αναλυθούν μερικές γνωστές εφαρμογές διαδραστικής οπτικοποίησης.

2.2.6.1 Grafana



Σχήμα 2.51: Grafana logo

Πηγή: https://chrisreeves.co.nz/2020/06/12/grafana_alerting_telegram_webhook/

Η Grafana είναι μια λύση ανάλυσης και παρακολούθησης ανοιχτού κώδικα, η οποία αναπτύσσεται και υποστηρίζεται από την Grafana Labs. Επιτρέπει την πραγματοποίηση ερωτημάτων, γραφικών

παραστάσεων και ειδοποιήσεων σχετικά με τις μετρήσεις χρονοσειρών, ανεξάρτητα από το πού είναι αποθηκευμένα τα δεδομένα. Το Grafana υποστηρίζεται από μια ζωντανή κοινότητα προγραμματιστών και χρησιμοποιείται σήμερα από χιλιάδες εταιρείες για την παρακολούθηση των πάντων.

Χαρακτηριστικά της Grafana:

- υποστηρίζει πάνω από περισσότερες από 30 πηγές δεδομένων ανοικτού κώδικα και εμπορικές πηγές δεδομένων με ελάχιστη υποδομή.

- μπορεί να ενσωματωθεί απρόσκοπτα στη ροή εργασίας για να βοηθήσει την εξερεύνηση μετρήσεων, την απεικόνιση δεδομένων, την εξερεύνηση αρχείων καταγραφής, την αξιολόγηση σημείων δεδομένων και την ειδοποίηση θεμάτων χρησιμοποιώντας διαφορετικά κανάλια - και μπορεί να τα κάνει όλα αυτά σε πραγματικό χρόνο.

- παρέχει ενσωματωμένη υποστήριξη για εξέχουσες βάσεις δεδομένων χρονοσειρών, όπως Graphite, InfluxDB, Prometheus, Elasticsearch και πολλές άλλες.

- έρχεται με μια τεράστια συλλογή διαδραστικών plugins που ξεκινούν από απλά διαγράμματα και γραφικές παραστάσεις μέχρι heatmaps, γεωχάρτες, διαγράμματα ροής και πολλά άλλα.

Ορισμένα πάνελ στο Grafana επιτρέπουν τον ορισμό κανόνων ειδοποίησης και την συνεχή αξιολόγηση των μετρήσεων σε σχέση με προκαθορισμένα κατώτατα όρια. Στη συνέχεια, μπορούν να στέλνονται ειδοποιήσεις μέσω διάφορων καναλιών, όπως email, Slack, PagerDuty, VictorOps κ.λπ.

Η Grafana επιτρέπει τη δημιουργία δυναμικών ταμπλό με τη βοήθεια μεταβλητών προτύπων. Αυτά τα ταμπλό εμφανίζουν τόνους δεδομένων χρησιμοποιώντας το ίδιο πρότυπο για να αυξήσουν την επαναχρησιμοποίηση.

Η Grafana επιτρέπει επίσης την υποβολή ερωτημάτων και την ανάμειξη διαφορετικών πηγών δεδομένων στο ίδιο γράφημα, κάτι που αποτελεί σπάνιο χαρακτηριστικό. Επίσης, άλλα χαρακτηριστικά όπως η χρήση ad hoc φίλτρων και η προσθήκη σχολίων το καθιστούν σκληρό ανταγωνιστή στην αγορά των εργαλείων απεικόνισης και ανάλυσης δεδομένων.

Για την υποβολή ερωτημάτων σε δεδομένα στο Grafana, πρέπει να χρησιμοποιηθεί μια γλώσσα ερωτημάτων. Πρόκειται για μια βασική μαθηματική γλώσσα για την αλληλεπίδραση με τη βάση δεδομένων.

Η Grafana είναι ανοικτού κώδικα, πράγμα που σημαίνει ότι είναι δωρεάν. Οποιοσδήποτε μπορεί να εγκαταστήσει το λογισμικό από το έργο ανοικτού κώδικα στο GitHub και να αρχίσει να το χρησιμοποιεί σήμερα. Για να παρακάμψουν τη δύσκολη διαδικασία εγκατάστασης και συντήρησης, οι χρήστες μπορούν να συνεργαστούν με το Hosted Grafana της MetricFire.

2.2.6.2 Power BI



Σχήμα 2.52: Power BI logo

Πηγή: <https://santebf.net/formations/power-bi/>

Το Power BI είναι μια λύση επιχειρηματικής ανάλυσης που αναπτύχθηκε από τη Microsoft Corporation. Επικεντρώνεται στην παροχή διαδραστικής απεικόνισης και δυνατοτήτων επιχειρηματικής ευφυΐας στους τελικούς χρήστες με ένα εύχρηστο περιβάλλον εργασίας, όπως κάθε άλλο προϊόν της Microsoft.

Χαρακτηριστικά του Power BI

- Το Power BI παρέχει στους μη τεχνικούς επιχειρηματικούς χρήστες μια ευέλικτη πλατφόρμα για την ανάλυση και την οπτικοποίηση των δεδομένων τους με ζωντανά dashboards και αναφορές.
- Το προϊόν είναι μια συλλογή διαφορετικών τμημάτων που συνεργάζονται για να αναδείξουν σημαντικές πληροφορίες που θα μπορούσαν να διαμοιραστούν σε όλους τους οργανισμούς ή να ενσωματωθούν στην εφαρμογή ή τον ιστότοπο.
- Το Power BI επιτρέπει την εισαγωγή δεδομένων από αρχεία, βάσεις δεδομένων, αποθήκες Azure, αναφορές Salesforce και πολλά άλλα. Οι χρήστες μπορούν να επιλέξουν να λάβουν δεδομένα από οποιαδήποτε από τις μορφές Excel, CSV, JSON, PDF κ.λπ.
- Υποστηρίζουν επίσης βάσεις δεδομένων όπως Oracle, MySQL, SQL Server, IBM DB2, SAP Hana και πολλές άλλες, συμπεριλαμβανομένων διαδικτυακών υπηρεσιών όπως Salesforce Reports, Google Analytics και Smartsheets. Υπάρχει ένας μακρύς κατάλογος πηγών δεδομένων που θα μπορούσαν να χρησιμοποιηθούν με το Power BI και υποστηρίζονται από την εφαρμογή.

- Οι εκθέσεις μπορούν να οργανωθούν σε σελίδες όπως ακριβώς και μια παρουσίαση.
- Το Power BI επιτρέπει στους χρήστες να αποκτήσουν πρόσθετη εικόνα των δεδομένων φιλτράροντας κάθε μεμονωμένο πίνακα ή διάγραμμα.
- Οι χρήστες μπορούν να προσθέσουν σελιδοδείκτες ή να τεμαχίσουν τα δεδομένα για να εμβαθύνουν περισσότερο. Μπορούν επίσης να αφήνουν σχόλια για να τα βλέπουν και άλλοι.
- Το Power BI επιτρέπει επίσης στους χρήστες να υποβάλλουν ερωτήσεις σχετικά με τα δεδομένα τους σε απλά αγγλικά. Υπάρχει ένα βασικό αγγλικό UI για την αναζήτηση δεδομένων.

Το Power BI χωρίζεται σε τρία διαφορετικά μέρη και κάθε μέρος έχει ελαφρώς διαφορετικό ρόλο:

- Power BI Desktop - Μια εφαρμογή των Windows που χρησιμοποιείται για τη δημιουργία μοντέλων δεδομένων, την ανάλυση των δεδομένων και τη δημιουργία αναφορών.
- Power BI service - Μια υπηρεσία SaaS που χρησιμοποιείται κυρίως για την κοινή χρήση των αναφορών. Η υπηρεσία παρέχει περιορισμένες δυνατότητες επεξεργασίας εκθέσεων και μοντελοποίησης δεδομένων.
- Power BI App - Μια εφαρμογή για κινητά για συσκευές Windows, iOS και Android που χρησιμοποιείται από τους τελικούς χρήστες για την προβολή της έκθεσης.

2.2.6.3 Zabbix



Σχήμα 2.53: Zabbix logo

Πηγή: <https://www.tienle.com/2017/12-25/server-monitoring-with-zabbix.html>

Το Zabbix είναι ένα εργαλείο παρακολούθησης ανοικτού κώδικα για συστήματα πληροφορικής, όπως δίκτυα, διακομιστές, εικονικές μηχανές και υπηρεσίες cloud. Το Zabbix μπορεί να συλλέγει μετρήσεις, να ανιχνεύει προβλήματα, να οπτικοποιεί, να ειδοποιεί και να στέλνει ειδοποιήσεις. Υπάρχει επίσης λειτουργικότητα για την παρακολούθηση εφαρμογών. Το Zabbix έχει σχεδιαστεί για περιβάλλοντα μικρής κλίμακας, από οπουδήποτε έως λίγες συσκευές έως χίλιες. Το Zabbix επιτρέπει επίσης την εύκολη καταναμημένη παρακολούθηση, όπου μπορείτε να διατηρήσετε τον αριθμό των συνδέσεων και των ανοικτών θυρών σε χαμηλά επίπεδα. Έχουν γίνει κάποιες εγκαταστάσεις μεγάλης κλίμακας, αλλά απαιτούν πολλή βελτιστοποίηση. Το Zabbix δεν αποθηκεύει δεδομένα το ίδιο, αλλά μπορεί να χρησιμοποιήσει ένα ευρύ φάσμα βάσεων δεδομένων. Το backend του Zabbix είναι γραμμένο σε C και το web frontend είναι σε PHP. Το Zabbix ως υπηρεσία είναι διαθέσιμο μέσω του ιστότοπου του Zabbix και υπάρχουν 5 στάδια τιμολόγησης.

2.2.6.4 Tableau

Το Tableau είναι ένα εργαλείο που χρησιμοποιείται για την οπτικοποίηση δεδομένων και την ανάλυση που πραγματοποιείται στον τομέα της επιχειρηματικής ανάλυσης. Μπορεί όμως να χρησιμοποιηθεί για περισσότερους από αυτόν τον έναν σκοπό - το Tableau διαθέτει άλλα πρόσθετα για τη διαχείριση δεδομένων και τη διαχείριση διακομιστών και οι προγραμματιστές μπορούν να προσαρμόσουν τα δικά τους πρόσθετα.

Το Tableau μπορεί να αντλήσει δεδομένα από οποιαδήποτε πηγή, να τα συνδυάσει και να επιτρέψει περαιτέρω μετασχηματισμούς προκειμένου να προετοιμάσει τα δεδομένα για οπτικοποίηση. Δημοφιλείς πηγές δεδομένων για το Tableau είναι το Excel, το PDF, η Oracle, οι υπηρεσίες Amazon Web Services.

Το οικοσύστημα του Tableau αποτελείται από τα ακόλουθα εργαλεία:

Tableau Public

Επιτρέπει τη δημιουργία βιβλίων εργασίας που θα είναι διαθέσιμα σε όλους από τον ιστότοπο του Tableau. Εδώ δεν παρέχεται καμία προστασία της ιδιωτικότητας. Είναι μια καλή επιλογή για εκπαιδευτικούς σκοπούς.

Tableau Desktop

Πρόκειται για ένα λογισμικό για τη δημιουργία βιβλίων εργασίας σε τοπική συσκευή και τη χρήση τους για οποιαδήποτε ανάγκη. Σε αυτή την περίπτωση, ένα βιβλίο εργασίας μπορεί να είναι ιδιωτικό. Το Tableau Desktop υπάρχει σε δύο εκδόσεις: Tableau Desktop Personal και Tableau Desktop Professional.

Tableau Online

Επιτρέπει την παράλληλη επεξεργασία του βιβλίου εργασίας φορτώνοντάς το στο νέφος του ομίλου Tableau.

Διακομιστής Tableau

Αν κάποιο άτομο δε θέλει να φορτώνει τα βιβλία εργασίας οπουδήποτε αλλού εκτός από τα προσωπικά μηχανήματα, μπορείτε να μοιραστεί με τον Tableau Server. Αφού δημιουργηθεί ένα βιβλίο εργασίας στο Tableau Desktop θα πρέπει να το δημοσιευθεί στον Server και να δοθούν διαπιστευτήρια πρόσβασης στους υπόλοιπους συναδέλφους.

2.2.6.5 Splunk



Σχήμα 2.54: Splunk logo

Πηγή: <https://sweetcode.io/splunk-conf18-to-feature-industry-visionaries-and-product-innovations/>

Το Splunk είναι ένα προϊόν λογισμικού που βοηθά την αναζήτηση, την ανάλυση, την απεικόνιση, την αναφορά και την ειδοποίηση σχετικά με τα δεδομένα μηχανής που συλλέγει από οποιοδήποτε σύστημα ή στοιχείο. Χρησιμοποιείται κυρίως ως συσσωρευτής αρχείων καταγραφής που επιτρέπει στους χρήστες να αναζητούν διορατικά δεδομένα από τεράστιους όγκους ευρετηριασμένων αρχείων καταγραφής.

Η Splunk Enterprise Solution είναι ικανή να εισάγει δεδομένα από σχεδόν οπουδήποτε, όπως αισθητήρες, συσκευές, εφαρμογές και ιστότοπους. Αυτά τα δεδομένα μπορούν στη συνέχεια να φιλτραριστούν και να εισαχθούν σε ευρετηριαστές, οι οποίοι με τη σειρά τους τα μετατρέπουν σε μεμονωμένα συμβάντα, έτοιμα για αναζήτηση και φιλτράρισμα. Το προϊόν μπορεί να φιλοξενηθεί και να υποστηριχθεί στο σπίτι ή να εγγραφεί σε συνδρομή μέσω cloud. Το Splunk εστιάζει ολόενα και περισσότερο στη μηχανική μάθηση και την τεχνητή νοημοσύνη, παρέχοντας στους χρήστες χαρακτηριστικά όπως η προγνωστική ανάλυση, η ανίχνευση ακραίων τιμών και η πρόβλεψη χρονοσειρών, τα οποία τους βοηθούν να εντοπίζουν αποκλίσεις προτύπων και να ειδοποιούν για προβλήματα ακόμη και πριν αυτά εμφανιστούν.

2.2.6.6 Datadog



Σχήμα 2.55: Datadog logo

Πηγή: <https://mattturck.com/datadog/>

Το Datadog επιτρέπει την προβολή μετρήσεων υποδομή και αρχείων καταγραφής, όλα στον ίδιο πίνακα ελέγχου. Το Datadog υποστηρίζει την οπτικοποίηση, την αντιμετώπιση προβλημάτων, την παρακολούθηση και την ειδοποίηση. Το Datadog ενσωματώνεται με εκατοντάδες διαφορετικές εφαρμογές ή υπηρεσίες και μπορεί να επικοινωνεί με οποιοδήποτε περιβάλλον, όπως διακομιστές, κοντέινερ, κινητά, προγράμματα περιήγησης ιστού και υπηρεσίες cloud. Ο πρωταρχικός στόχος του Datadog είναι να βλέπει μέσα σε οποιαδήποτε στοίβα και οποιαδήποτε εφαρμογή. Το Datadog υποστηρίζει επίσης ειδοποιήσεις, συνεργασία και σας επιτρέπει να συνδυάζετε δεδομένα από διάφορες πηγές σε ένα οπτικό υλικό.

3. Σχεδιασμός

3.1 Στόχοι της εφαρμογής

Οι βασικοί στόχοι της εφαρμογής είναι οι παρακάτω:

- Ο πρώτος στόχος της εφαρμογής του IoT είναι η εμφάνιση των δεδομένων της στην οθόνη του χρήστη. Συγκεκριμένα, τα δεδομένα αυτά θα πρέπει να είναι διαθέσιμα μέσω οπτικοποίησης διεπαφής γραφικών χρήστη (Graphical User Interface – GUI), στην οποία θα εμφανίζονται γραφήματα αναπαράστασης χρήσιμων δεδομένων και η μεταβολή τους με την πάροδο του χρόνου, καθώς και κουμπιά ενεργειών από την πλευρά του χρήστη προς το IoT.

- Η εφαρμογή αυτή προορίζεται για χρήση από άτομα διαφορετικών επιπέδων γνώσεων και ικανοτήτων σε ό,τι αφορά τον τομέα της τεχνολογίας. Ως εκ τούτου, είναι απαραίτητο να δημιουργηθεί μια απλή στην κατανόηση επιφάνεια αλληλεπίδρασης, φιλική προς οποιοδήποτε χρήστη. Σε καμία περίπτωση δε θα πρέπει να απαιτείται από την εφαρμογή η γνώση των τεχνολογιών που έχουν επιλεγεί για τη δημιουργία της, ούτε προφανώς γνώσεις προγραμματισμού.

- Παράλληλα, είναι πιθανό κάποιοι χρήστες να γνωρίζουν περισσότερα για την αρχιτεκτονική ενός IoT από το μέσο χρήστη και να επιθυμούν να έχουν πρόσβαση στα δεδομένα, όπως αυτά αποθηκεύονται στη βάση (raw data). Για αυτό θα πρέπει να υποστηρίζεται μια διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface - API) που συμβαδίζει με τους περιορισμούς του αρχιτεκτονικού στύλ μεταφοράς αντιπροσωπευτικών καταστάσεων (REpresentational State Transfer – REST), γνωστό και ως RESTful web API, για λήψη δεδομένων μέσω URI. Κάθε διαφορετική διεύθυνση αντιστοιχεί σε ένα σύνολο δεδομένων σε μορφή BSON.

- Είναι σημαντικό να υπάρχει ασφάλεια στην ανταλλαγή δεδομένων μεταξύ πελάτη και εξυπηρετητή. Χωρίς αυτή θα ήταν δυνατό να σταλούν δεδομένα στο server από τρίτο, κακόβουλο χρήστη ή το αντίστροφο.

- Καθώς η εφαρμογή θα λαμβάνει και θα στέλνει δεδομένα με μεγάλη ταχύτητα, θα πρέπει να υπάρχει γρήγορη απόκριση της εφαρμογής τόσο στη λήψη, όσο και στην αποστολή οδηγιών. Σε πολλές εφαρμογές, η απόδοση μειώνεται όσο αυξάνεται το πλήθος των αποθηκευμένων δεδομένων στη βάση, το οποίο θα πρέπει να αποφευχθεί σε αυτήν την περίπτωση.

- Για τις ανάγκες της διπλωματικής, η εφαρμογή καλύπτει τις ανάγκες των αισθητήρων ενός σπιτιού. Παράλληλα, η βασική δομή της αρχιτεκτονικής πρέπει να είναι τέτοια, ώστε να είναι δυνατή η επέκταση στο επίπεδο μιας εταιρίας που υποστηρίζει μεγάλο πλήθος εγκαταστάσεων ταυτοχρόνως ως ένα σύμπλεγμα (cluster), χωρίς φυσικά να έχει αντίκτυπο στην απόδοση της εφαρμογής.

3.2 Τύπος δεδομένων IoT

Τα δεδομένα που θα χρησιμοποιηθούν είναι εικονικά και δεν αντιστοιχούν σε πραγματικές μετρήσεις αισθητήρων. Ωστόσο, για να εξομοιωθεί η αρχιτεκτονική ενός IoT σε ένα σπίτι, θα θεωρηθεί ότι δίνονται περιβαλλοντικές τιμές (όπως θερμοκρασία ή υγρασία) με σκοπό τη διατήρησή τους σε συγκεκριμένα επίπεδα στο χώρο του σπιτιού. Παράλληλα, θα δίνεται η δυνατότητα αποστολής εντολών σε συσκευές, οι οποίες θα επηρεάζουν τις παραπάνω τιμές. Μερικά παραδείγματα τέτοιων συσκευών είναι:

- το κλιματιστικό
- ο μηχανισμός που ανοιγοκλείνει τα στόρια
- ο μηχανισμός που ανεβάζει/κατεβάζει τις τέντες
- ο μηχανισμός που ανοίγει/κλείνει τον εξαερισμό
- ο μηχανισμός που ανοίγει/κλείνει τους φεγγίτες

3.3 Επιλογές IoT

Για την υλοποίηση της συγκεκριμένης εργασίας έγιναν οι εξής επιλογές προγραμμάτων και πρωτοκόλλων:

3.3.1 MONGODB

Η βάση δεδομένων MongoDB έχει σχεδιαστεί από την αρχή για να είναι μια καταναμημένη βάση δεδομένων, οπότε διατηρεί την απόδοση και την οριζόντια κλιμάκωση. Επίσης, το μοντέλο εγγράφων της MongoDB επιτρέπει την εύκολη μοντελοποίηση και επεξεργασία σχεδόν κάθε είδους δομής δεδομένων. Η μορφή δεδομένων BSON της MongoDB, εμπνευσμένη από το JSON, επιτρέπει την ύπαρξη αντικειμένων σε μια συλλογή με διαφορετικά σύνολα πεδίων (π.χ. το μεσαίο όνομα ενός χρήστη μόνο όταν ισχύει, ή πληροφορίες για συγκεκριμένη περιοχή που ισχύουν μόνο για ορισμένες εγγραφές).

Η MongoDB υποστηρίζει τη δημιουργία ρητών σχημάτων και την επικύρωση των δεδομένων, ώστε να μην ξεφεύγουν από τον έλεγχο, αλλά αυτή η ευελιξία είναι ένα πλεονέκτημα σε δεδομένα του

πραγματικού κόσμου, ιδιαίτερα ενός IoT, που συχνά απαιτούν αλλαγές στις απαιτήσεις ή στο περιβάλλον.

Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν wrappers, όπως οι ORM (Object Relational Mappers), για να μεταφέρουν τα δεδομένα σε μορφή αντικειμένου για χρήση σε προγράμματα. Η απόφαση της MongoDB να αποθηκεύει και να αναπαριστά τα δεδομένα σε μορφή εγγράφου επιτρέπει την πρόσβαση σε αυτά από οποιαδήποτε γλώσσα, σε δομές δεδομένων που είναι εγγενείς σε αυτή τη γλώσσα (π.χ. λεξικά στην Python, συσχετιστικοί πίνακες στην JavaScript, χάρτες στη Java κ.λπ.)

3.3.2 MQTT

Οι λόγοι είναι:

- Οι πελάτες MQTT είναι ελαφροί και απαιτούν ελάχιστους πόρους, ώστε να μπορούν να χρησιμοποιηθούν σε μικρούς μικροελεγκτές. Μπορούν επίσης να χρησιμοποιηθούν για τη βελτιστοποίηση του εύρους ζώνης του δικτύου.

- Το πρωτόκολλο MQTT μπορεί να κλιμακωθεί για τη σύνδεση εκατομμυρίων συσκευών IoT.

- Το MQTT επιτρέπει τη ροή μηνυμάτων από το σύννεφο προς τη συσκευή και από τη συσκευή προς το σύννεφο. Αυτό απλοποιεί τη μετάδοση μηνυμάτων σε μια ομάδα πραγμάτων.

- Το MQTT είναι γνωστό για την αξιόπιστη παράδοση μηνυμάτων, καθιστώντας το ιδανική επιλογή για πολλές περιπτώσεις χρήσης IoT. Διαθέτει καθορισμένα επίπεδα ποιότητας υπηρεσιών.

- Το MQTT απλοποίησε τη διαδικασία κρυπτογράφησης μηνυμάτων με τη χρήση TLS και την αυθεντικοποίηση των πελατών με τη χρήση σύγχρονων πρωτοκόλλων όπως το OAuth.

- Το MQTT επιλύει το κοινό πρόβλημα των αναξιόπιστων κυψελοειδών δικτύων μέσω των οποίων συνδέονται πολλές συσκευές IoT. Το πρωτόκολλο υποστηρίζει μόνιμες συνεδρίες, μειώνοντας έτσι τον χρόνο επανασύνδεσης του πελάτη με τον διαμεσολαβητή.

3.3.2 Mosquitto MQTT

Το Mosquitto MQTT είναι σχεδιασμένο για συσκευές IoT. Είναι ιδανικό για απομακρυσμένες συσκευές χαμηλής ισχύος που στέλνουν μηνύματα σε δίκτυο με περιορισμένο εύρος ζώνης. Είναι αποδοτικό ως προς την καλωδίωση και οι προσπάθειες για την υλοποίηση σε έναν πελάτη είναι μικρότερες. Επομένως, είναι πιο κατάλληλο για το είδος της εφαρμογής που υλοποιείται από ότι είναι το RabbitMQ ή το HiveMQ.

3.3.3 Node-RED

Το Node-RED είναι σχεδιασμένο για IoT projects, καθώς απαιτεί πολύ λιγότερους πόρους από το Apache Nifi. Το βασικό πλεονέκτημα του Apache Nifi έναντι του Node-RED είναι εκείνο της κλιμάκωσης του project, το οποίο δεν αποτελεί πρόβλημα για την εμβέλεια της συγκεκριμένης εφαρμογής.

3.3.4 Grafana

Τα πλεονεκτήματα του Grafana έναντι των υπόλοιπων εφαρμογών διαδραστικής οπτικοποίησης είναι τα εξής:

- Το Grafana είναι ένα σύστημα απεικόνισης δεδομένων όπου ενσωματώνονται διαφορετικές αποθήκες δεδομένων, έτσι ώστε να μπορούν να διαμορφωθούν κατάλληλες μετρήσεις των χρονικά δεσμευμένων δεδομένων. Αυτό βοηθά στην παρακολούθηση των δεδομένων και τη χρήση τους για την ανάλυση παρελθοντικών και τρεχόντων δεδομένων που μπορούν να χρησιμοποιηθούν στην επιχείρηση. Το Tableau είναι εργαλείο επιχειρηματικής ευφυΐας που βοηθά στη δημιουργία ταμπλό και στην αποθήκευση δεδομένων εκεί, ώστε οι χρήστες να μπορούν να απεικονίσουν την παρουσία δεδομένων σε οποιαδήποτε μορφή.
- Στο Grafana τα γραφήματα σχηματίζονται με τη βοήθεια των αρχείων καταγραφής ή των μετρήσεων από οποιαδήποτε πηγή, ενώ το Tableau δίνει την οπτικοποίηση των γραφικών παραστάσεων μόνο όταν τα δεδομένα είναι αποθηκευμένα σε μορφή excel.
- Οι ειδοποιήσεις μπορούν να δημιουργηθούν εύκολα στη Grafana με τη βοήθεια των μετρήσεων και των διαγραμμάτων που σχηματίζονται. Αυτές οι ειδοποιήσεις βοηθούν τους χρήστες να γνωρίζουν τα δεδομένα εάν υπάρχουν απρόβλεπτα γεγονότα ή μετασχηματισμοί. Αυτό βοηθά στη διόρθωσή τους και στην πρόβλεψη των ελαττωμάτων του ίδιου. Αντίθετα, στο tableau οι ειδοποιήσεις δεν είναι διαθέσιμες και οι χρήστες πρέπει να ελέγχουν τη διεπαφή χρήστη για να γνωρίζουν τη ροή των δεδομένων.
- Η Grafana βοηθά τις εκροές δεδομένων να σχηματίσουν οργανωμένα δεδομένα, ώστε οι χρήστες να μπορούν να κατανοήσουν τα δεδομένα εύκολα. Αυτό βοηθά τα δεδομένα να χρησιμοποιηθούν σε συγκεκριμένες ανάγκες ή για την επικοινωνία μεταξύ ομάδων. Στο tableau, η ίδια η είσοδος είναι σε μορφή excel, το οποίο περιορίζει τις δυνατότητες οργάνωσης.
- Τα ερωτήματα μπορούν να δημιουργηθούν εύκολα με το Grafana και αυτό βοηθά τους χρήστες να εντοπίσουν τη ροή των δεδομένων στο σύστημα και να τα οργανώσουν. Τα dashboards βοηθούν στη δημιουργία των γραφημάτων, ώστε τα δεδομένα να μπορούν να απεικονιστούν

εύκολα. Ωστόσο, τα ερωτήματα δεν μπορούν να δημιουργηθούν στο tableau και απαιτείται μια εξωτερική πηγή εάν πρέπει να χρησιμοποιηθούν ερωτήματα στην εφαρμογή [32].

- Η τεκμηρίωση του Graphite δεν είναι τόσο καλή όσο της Grafana. Στα έγγραφα του Grafana εξηγούνται οι βασικές έννοιες και όλα τα σεμινάρια και τα plugins, ώστε ο χρήστης να εξοικειωθεί με τα χαρακτηριστικά της εφαρμογής.
- Διαφορετικά εργαλεία προσφέρονται στο Graphite για να προσφέρουν τα plugins αν και η ίδια η εφαρμογή δεν υποστηρίζει άμεσα τα plugins. Ταυτόχρονα, ένα μεγάλο σύνολο βιβλιοθήκης plugins παρέχεται στο Grafana, έτσι ώστε οι χρήστες να μπορούν να προσαρμόσουν τις γραφικές παραστάσεις για οπτικοποίηση και να είναι πιο καινοτόμες για τις γραφικές παραστάσεις. Επίσης, αυτά τα πρόσθετα μπορούν να χρησιμοποιηθούν για την επεξεργασία του πίνακα οργάνων.
- Οι επιλογές οπτικοποίησης παρέχονται και στις δύο εφαρμογές, αλλά οι προσαρμογές και η επεξεργασία ταμπλό δεν είναι διαθέσιμες στο graphite. Στο Grafana και στην οπτικοποίηση, η επεξεργασία περιλαμβάνεται επίσης με ευέλικτα και πλούσια χαρακτηριστικά για να ικανοποιήσει τους πελάτες. Το ταμπλό είναι εύκολο στη χρήση, ώστε να αρέσει στους πελάτες η Grafana για την οπτικοποίηση που παρέχει.

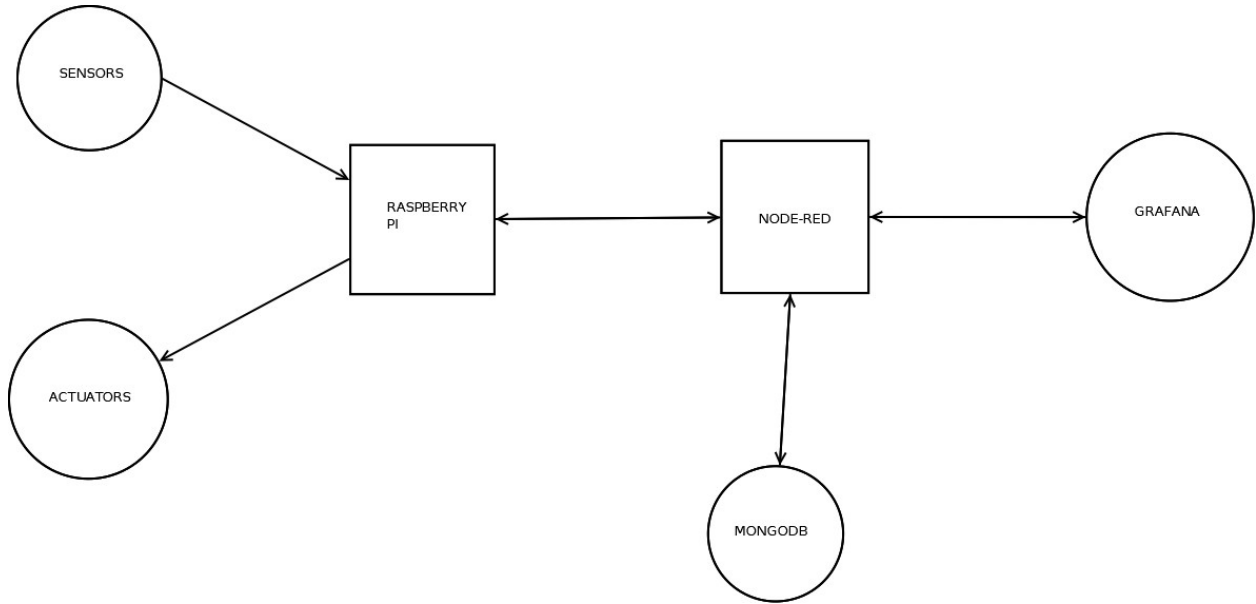
3.4 Παραδοχές

Θεωρούμε ότι όλα τα δεδομένα έρχονται από το SBC και συγκεκριμένα από τους αισθητήρες του, όπως επίσης και ότι στέλνονται στο SBC και συγκεκριμένα στους actuators, μέσω της λειτουργίας του MQTT. Η πλακέτα έχει προσομοιωθεί από τα δεδομένα που εισάγουμε στη βάση και από εντολές που στέλνει ο χρήστης και δεν έχει πραγματική υλοποίηση, ούτε σε επίπεδο αισθητήρων που στέλνουν υπαρκτές μετρήσεις τιμών, ούτε σε επίπεδο εντολών προς το SBC.

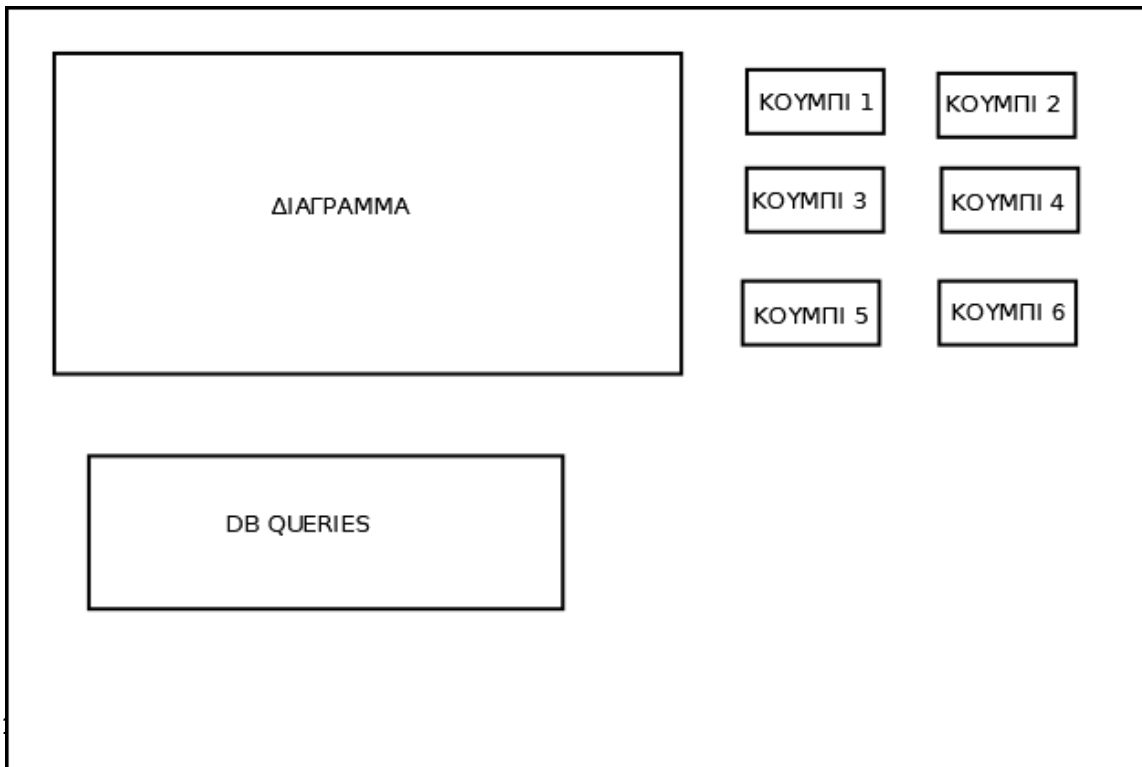
3.5 Αρχιτεκτονική

Έχοντας υπόψιν τις τεχνολογίες που επιλέξαμε στο προηγούμενο κεφάλαιο, πρέπει να βρούμε έναν τρόπο να τις συνδυάσουμε, ώστε να δημιουργήσουμε την υποδομή του IoT. Για να επιτευχθεί αυτό, χρειάζεται σύνδεση των αισθητήρων που δίνουν τα εικονικά δεδομένα με την πλακέτα που θα συλλέγει και θα οργανώνει τις πληροφορίες, στη συνέχεια με το message broker, ο οποίος θα δρομολογεί τις πληροφορίες στην κατάλληλη ροή, ώστε να αποθηκευτούν στη βάση δεδομένων και να

εμφανιστούν στην εφαρμογή οπτικοποίησης. Με βάση τα παραπάνω, το σχήμα που περιγράφει τη βασική σύνδεση των εφαρμογών είναι το εξής:



Η τελική εικόνα του Grafana θα πρέπει να είναι κάπως έτσι:



Η επιφάνεια θα περιέχει τα εξής:

- Στο πάνω αριστερό μέρος θα βρίσκεται το διάγραμμα που απεικονίζεται μετά την εκτέλεση του ερωτήματος που θέτουμε στη βάση. Συνήθως, πρόκειται για μια συνάρτηση μιας περιβαλλοντικής τιμής ως προς το χρόνο.

- Στο πάνω δεξιό μέρος θα βρίσκονται κουμπιά, τα οποία στέλνουν εικονικές εντολές που προσομοιώνουν την εκτέλεση εντολών από το χρήστη. Στο αριστερό μέρος θα βρίσκεται μια εντολή ενεργοποίησης, ενώ στο δεξιό θα είναι η αντίστροφη εντολή για την απενεργοποίηση μιας λειτουργίας (για παράδειγμα, ανέβασμα τέντας / κατέβασμα τέντας). Να σημειωθεί ότι στην εικόνα απεικονίζονται συνολικά έξι κουμπιά, αλλά στην πράξη θα μπορούσαμε να έχουμε περισσότερα ή λιγότερα, ανάλογα με το πλήθος των εντολών που υποστηρίζονται από την εφαρμογή.

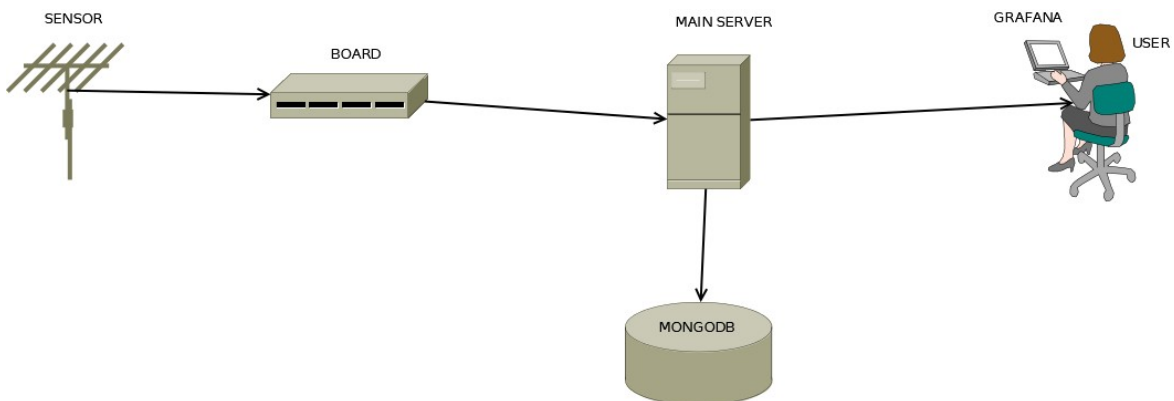
- Στο κάτω μέρος θα βρίσκεται ένα πεδίο κειμένου, στο οποίο ο χρήστης εισάγει το ερώτημα στη βάση, προκειμένου να εμφανιστεί το κατάλληλο διάγραμμα. Τα ερωτήματα αυτά θα μπορούσαν να έχουν δοθεί εκ των προτέρων από το σχεδιαστή της εφαρμογής στο χρήστη, ώστε να μπορεί να ελέγχει όποια τιμή επιλέξει. Αυτό σημαίνει ότι δεν απαιτείται από το χρήστη να δημιουργήσει δικά του ερωτήματα, ούτε να κατανοεί το πως λειτουργούν τα υπάρχοντα, απλά να έχει αποθηκευμένα τα ερωτήματα όπως είναι και να τα αντιγράφει στο πεδίο για να τα εκτελέσει.

3.6 Σενάρια χρήσης

Μερικά πιθανά σενάρια χρήσης είναι τα εξής:

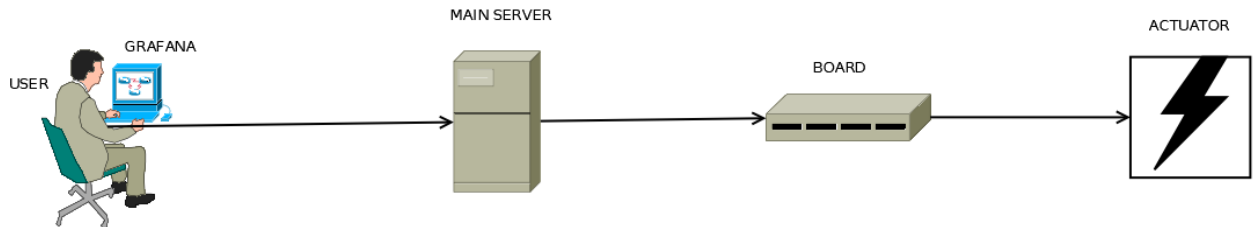
3.6.1 Εισαγωγή δεδομένων στο σύστημα μέσω αισθητήρων

Ο αισθητήρας αντιλαμβάνεται σημαντική μεταβολή σε κάποια τιμή, τη μεταδίδει στην πλακέτα και από εκεί στον κεντρικό server που ελέγχεται από το Node-RED. Το Node-RED ενημερώνει τη βάση και ταυτόχρονα και το χρήστη μέσω της οπτικοποίησης του Grafana.



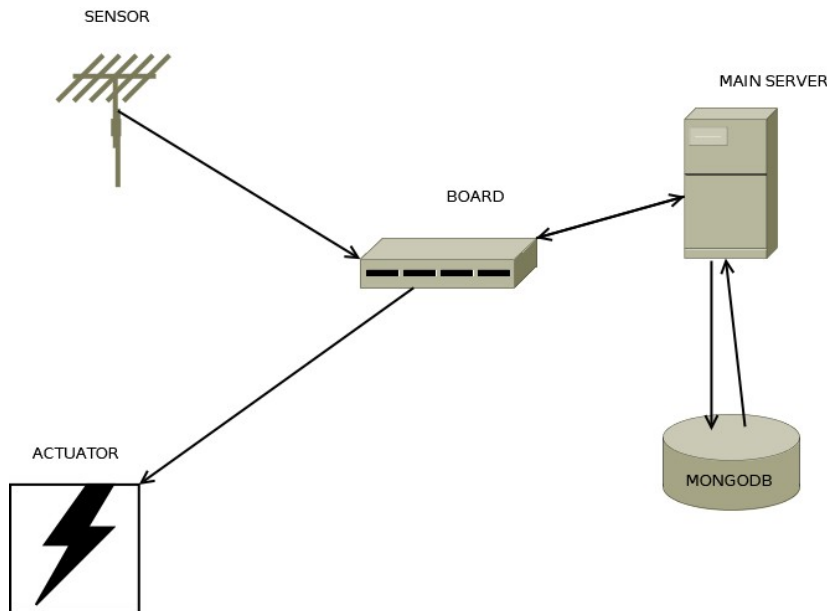
3.6.2 Αποστολή εντολών του χρήστη

Ο χρήστης αποφασίζει να στείλει εντολή στο σύστημα μέσω του Grafana. Αυτή μεταδίδεται στον κεντρικό server στο Node-RED, το οποίο με βάση την επιλογή της κατάλληλης ροής στέλνει το αντίστοιχο μήνυμα στην πλακέτα, ώστε να ενεργοποιήσει τον κατάλληλο actuator, ο οποίος εκτελεί τη συγκεκριμένη εντολή (π.χ. να ανοίξει μια πόρτα)

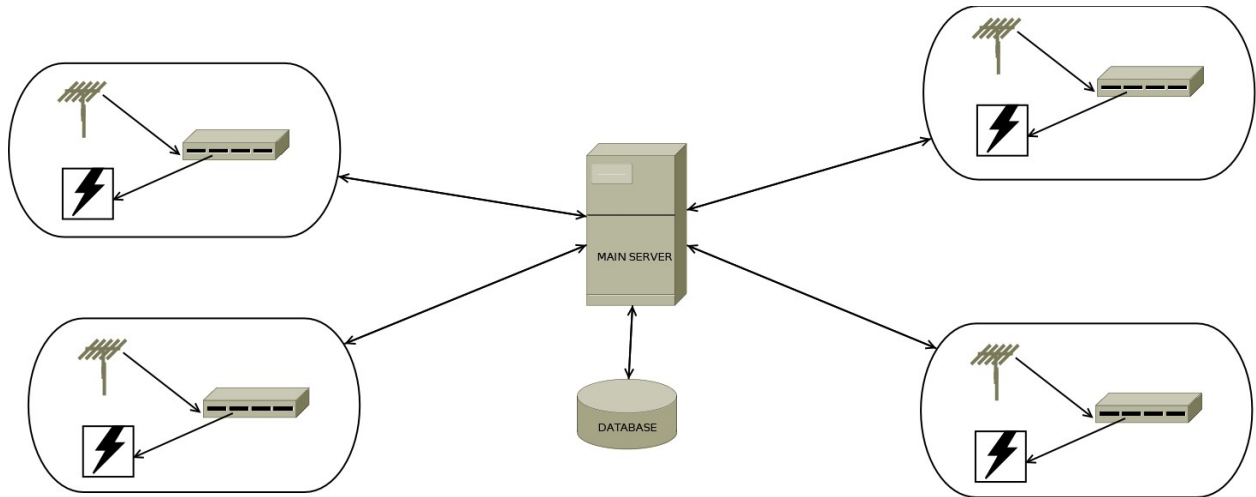


3.6.3 Ανανέωση δεδομένων του συστήματος μέσω κανόνων

Το κεντρικό σύστημα του Node-RED, με βάση τους δοσμένους κανόνες, υπολογίζει ότι με βάση τις πιο πρόσφατες μετρήσεις κάποιας τιμής, πρέπει να ενεργοποιήσει από μόνο του κάποια ροή (π.χ. αν η θερμοκρασία πέσει κάτω από μια δοσμένη τιμή, ενεργοποιεί αυτόματα την ηλεκτρική σόμπα)



Μια επέκταση της συγκεκριμένης εφαρμογής θα είχε τη δυνατότητα να συνδέσει τον κεντρικό server (που θα αντιστοιχούσε στον πάροχο της υπηρεσίας) με πολλά διαφορετικά σπίτια, έχοντας τη δυνατότητα να επικοινωνήσει με όλους τους sensors και actuators και να εμφανίζει τις πληροφορίες του σπιτιού στον αντίστοιχο πελάτη μέσω της εφαρμογής οπτικοποίησης.



Η τελική μορφή της εργασίας θα είναι η εξής:

- Ο χρήστης θα εισάγει δεδομένα είτε μέσω αρχείου είτε μέσω της κονσόλας, τα οποία θα μεταδίδονται μέσω του Mosquitto MQTT στο Node-RED (τα δεδομένα αυτά εξομοιώνουν τα δεδομένα των αισθητήρων που θα παρείχε το Raspberry Pi)
- Το Node-RED θα διοχετεύει τα δεδομένα αυτά στην κατάλληλη ροή, ώστε να αποθηκευτούν στη βάση.
- Ταυτόχρονα, θα στέλνονται δεδομένα προς το Grafana ώστε να τα οπτικοποιήσει με τη χρήση γραφημάτων.
- Παράλληλα, ο χρήστης θα έχει τη δυνατότητα να ζητήσει raw data από τη βάση μέσω κάποιων URI, ώστε να λάβει πληροφορίες σε μορφή BSON.
- Τέλος, ο χρήστης θα έχει τη δυνατότητα να πατάει κουμπιά στη διεπαφή του Grafana, με τα οποία θα στέλνει εντολές στο Node-RED και αυτό με τη σειρά του στο Mosquitto MQTT (οι εντολές αυτές προσομοιώνουν τις εντολές που θα στέλνονταν στους actuators του Raspberry Pi)

4. Λειτουργίες και Αρχιτεκτονική

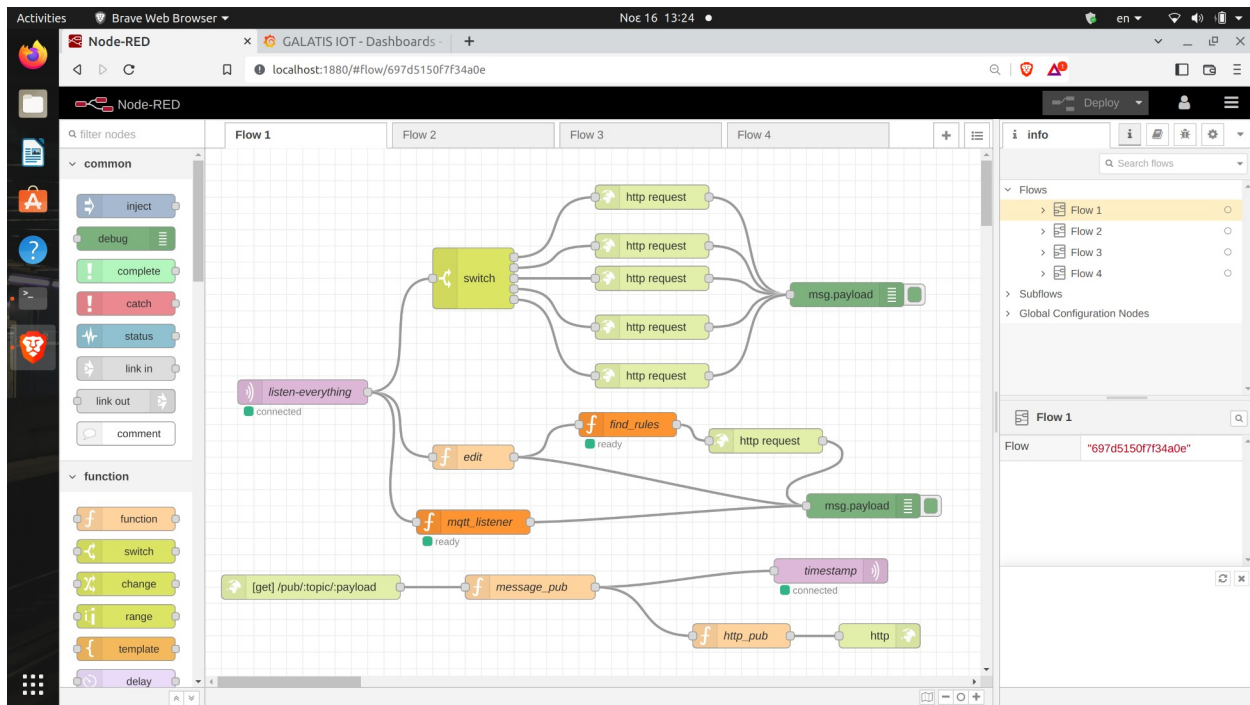
4.1 Ανάπτυξη Σεναρίων Χρήσης

Για την εκτέλεση των σεναρίων χρησιμοποιήθηκε μια βάση δεδομένων, η οποία περιείχε διάφορες μετρήσεις αεροπλάνων κατά τη διάρκεια των πτήσεων. Η βάση αυτή είναι open source και βρίσκεται στην εξής σελίδα: (βάλτε τη σελίδα)

Για τη μαζική εισαγωγή δεδομένων στη βάση χρησιμοποιήθηκε ένα script σε Python, το οποίο διαβάζει τις υπάρχουσες εγγραφές της βάσης data και τις μετατρέπει, ώστε να αποθηκευτούν με βάση το format της βάσης weather.

4.1.1 Εισαγωγή δεδομένων στο σύστημα μέσω αισθητήρων

Για την προσομοίωση της εισαγωγής δεδομένων χρησιμοποιούμε της κατάλληλες ροές (flows) στο Node-RED, ώστε να λαμβάνουμε τα δεδομένα των μετρήσεων και να εμφανίζουμε όσες πληρούν ορισμένα κριτήρια (ενδεχομένως μόνο μέρη των μετρήσεων).



Με το node listen-everything έχουμε ένα listener όλων των μηνυμάτων μέσω του MQTT, το οποίο προσομοιώνει την αποστολή πακέτων από την πλακέτα. Όταν σταλεί ένα μήνυμα, υπάρχουν τρεις δυνατές εκφάνσεις:

- η ροή το μεταβιβάζει στο mqtt_listener , που αντλεί μέσα από το payload του μηνύματος τα δεδομένα της εγγραφής και την αποθηκεύει στη βάση δεδομένων weather.

- η ροή το μεταβιβάζει στον κόμβο edit, ο οποίος το επεξεργάζεται κατάλληλα ώστε να δοθεί στον κόμβο find_rule , ο οποίος βρίσκει τον κανόνα που εισήγαγε ο χρήστης μέσω του μηνύματος και τον προσθέτει στο σύστημα.

- η ροή το μεταβιβάζει στο switch, το οποίο, ανάλογα με την τιμή του payload του μηνύματος εκτελεί το αντίστοιχο http request.

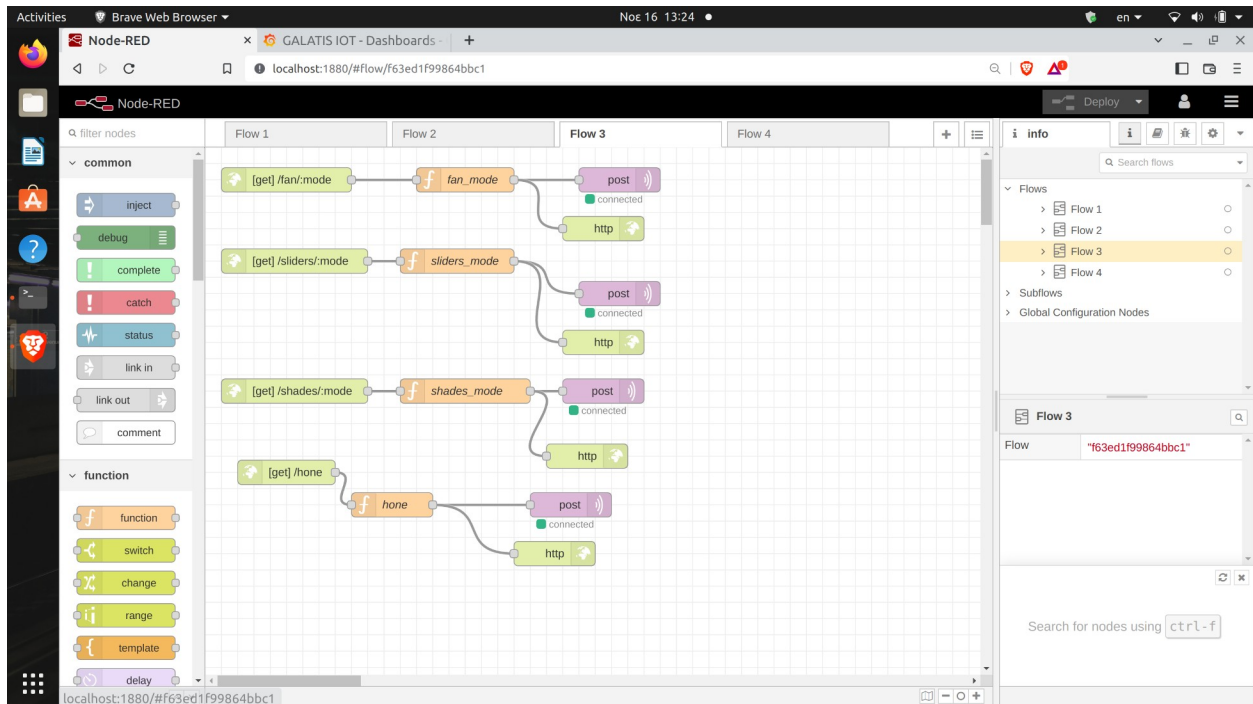
Πρέπει να σημειωθεί ότι:

- Ο κόμβος msg.payload βρίσκεται ώστε να επιβεβαιώσουμε στο Node-RED ότι η διαδικασία ολοκληρώθηκε σωστά, σε όλες τις ροές στις οποίες υπάρχει. Πρόκειται για ένα node το οποίο απλά τυπώνει το απεσταλμένο μήνυμα από τον τελευταίο κόμβο πριν αυτόν στην καρτέλα παρουσίασης του Node-RED.

- Οι συναρτήσεις είναι υλοποιημένες με τέτοιο τρόπο, ώστε να ελέγχουν πάντα εάν το απεσταλμένο μήνυμα είναι στη μορφή που πρέπει, ώστε να το λάβουν υπόψιν. Για παράδειγμα, εάν ο χρήστης στείλει μήνυμα εισαγωγής στοιχείου στη βάση, η ροή διαχείρισης κανόνων θα αναγνωρίσει το γεγονός ότι το μήνυμα είναι σε format το οποίο είναι διαφορετικό από εκείνο που αναμένει, οπότε απλώς αγνοεί το μήνυμα αυτό.

4.1.2 Αποστολή εντολών του χρήστη

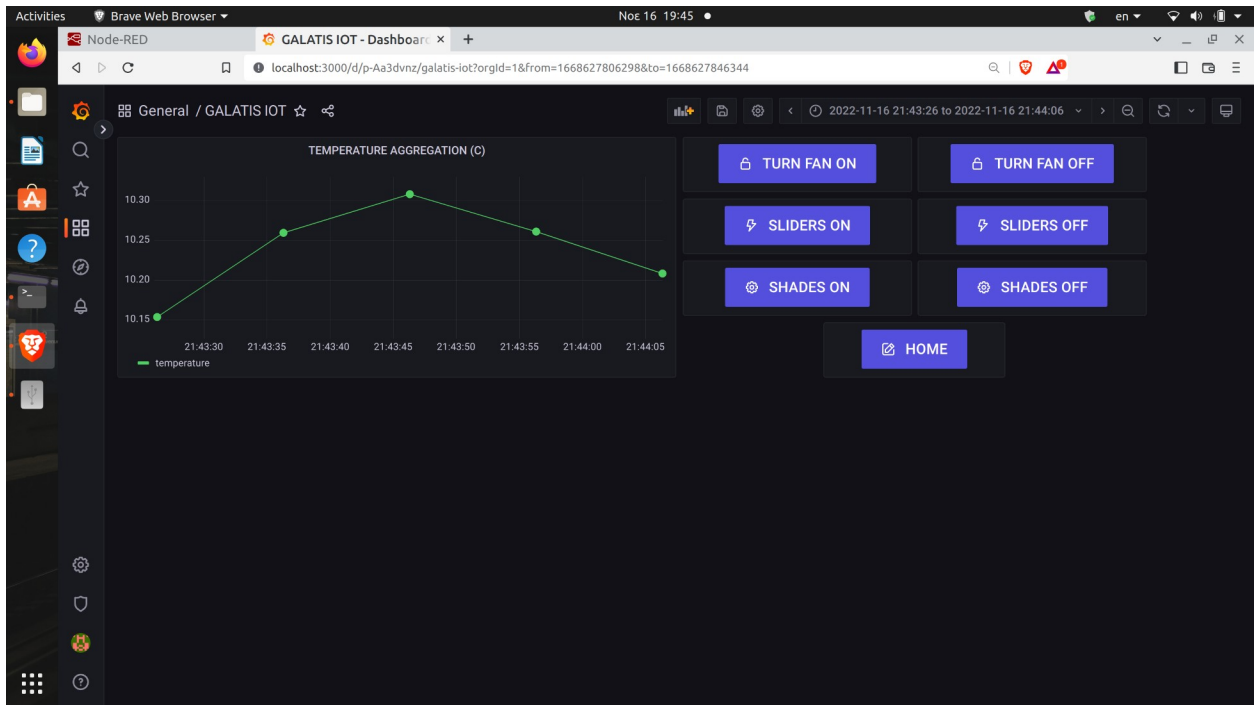
Για την προσομοίωση της αποστολής εντολών του χρήστη χρησιμοποιούμε τις παρακάτω ροές, οι οποίες δέχονται σήμα μέσω http get και στέλνουν αντίστοιχο μήνυμα post στο MQTT. Το μήνυμα αυτό εμφανίζεται σε οποιονδήποτε χρήστη είναι εγγεγραμμένος στο συγκεκριμένο topic, οπότε θα ήταν δυνατό να θέσουμε συγκεκριμένα topics για κάθε actuator και να στέλνουμε μήνυμα σε καθένα από αυτά όποτε χρειαστεί να σταλεί εντολή σε εκείνον τον actuator.



Ο κόμβος fan/:mode συνδέεται με τη συνάρτηση fan_mode , η οποία χειρίζεται τη λειτουργία του ανεμιστήρα. Συγκεκριμένα, αν η παράμετρος mode έχει την τιμή on, τότε ενεργοποιεί τον ανεμιστήρα, αλλιώς τον απενεργοποιεί. Το μήνυμα στέλνεται στο θέμα post του MQTT, αλλά και σαν απάντηση http, σε περίπτωση που κάποιος θέλει να το επιβεβαιώσει με αυτόν τον τρόπο.

Με αντίστοιχο τρόπο λειτουργούν και οι κόμβοι shades/:mode και sliders/:mode.

Εξετάζουμε μια απλή χρήση των ροών σε συνεργασία με το Grafana:



The screenshot shows a terminal window with a dark background and light text. The terminal is running a series of commands to update the system and set up an MQTT subscription. The commands and their outputs are as follows:

```

594 sudo apt-get update
595 sudo apt-get upgrade
596 history
597 which steam
598 ls -la /usr/games/steam
599 which steam
600 ls
601 cd /usr/share/applications
602 ls
603 gedit steam.desktop
604 sudo gedit steam.desktop
605 df -h
606 pydf
607 sudo apt install pydf
608 pydf
609 sudo apt install discus
610 discus
611 sudo -i
612 gedit insert.py
613 history
galatis@galatis-Inspiron-3585:~/scripts$ mosquitto sub -u guest -P 123456 -t test
^Cgalatis@galatis-Inspiron-3585:~/scripts$ mosquitto_sub -u guest -P 123456 -t post
fan.deactivate()

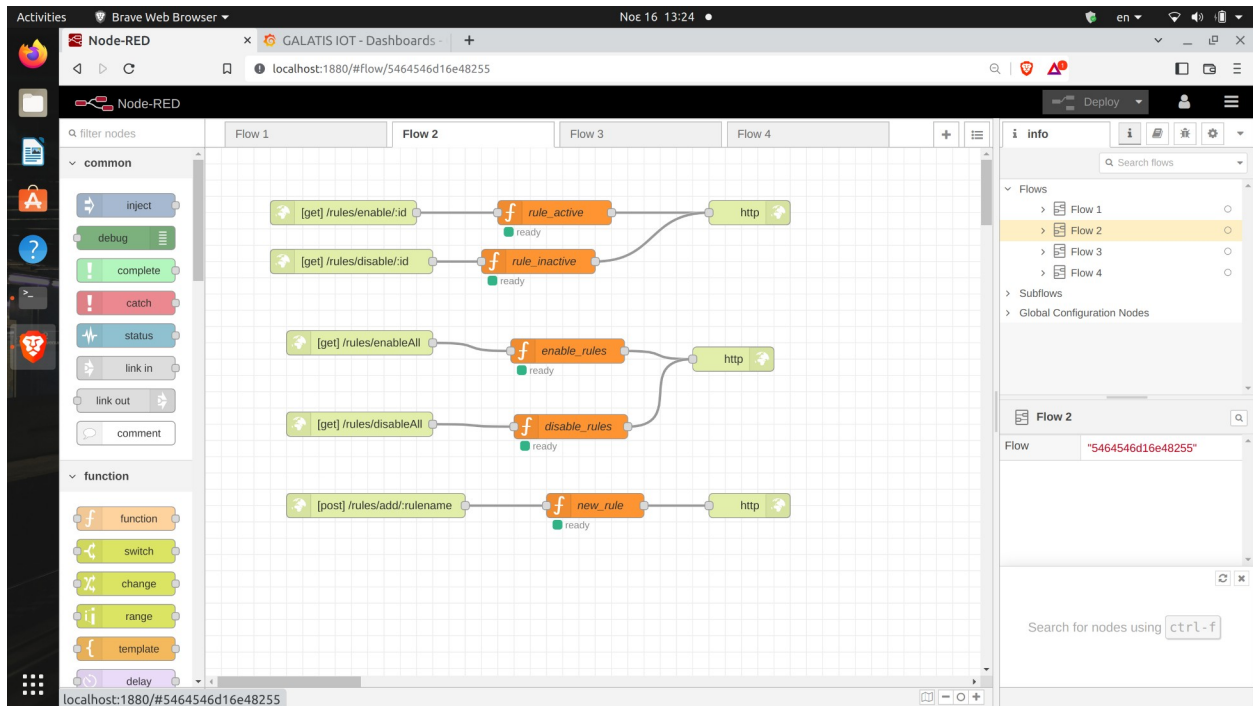
```

The terminal window is titled 'galatis@galatis-Inspiron-3585: ~/scripts'. The browser window in the background shows the same dashboard as the first screenshot, but with the 'TURN FAN OFF' button highlighted with a checkmark.

- Παρατηρούμε ότι η τιμή της θερμοκρασίας είναι χαμηλή μέσω του διαγράμματος του Grafana.
- Πατούμε το κουμπί που σβήνει τον ανεμιστήρα (επιβεβαιώνουμε ότι πατήθηκε από το σύμβολο tick που εμφανίζεται δίπλα στο κείμενο).
- Το κουμπί στέλνει http request στη διεύθυνση που αντιστοιχεί στη ροή του ανεμιστήρα με τιμή της παραμέτρου mode στο off.
- Το αποτέλεσμα είναι η αποστολή του μηνύματος fan.deactivate() στο θέμα post του MQTT. Προφανώς, σε πραγματικές συνθήκες θα είχε σταλεί εντολή απενεργοποίησης του actuator του ανεμιστήρα.
- Για να δούμε αυτό το μήνυμα κάνουμε subscribe στο mosquitto στο θέμα test, χρησιμοποιώντας όνομα χρήστη guest και κωδικό 123456. Στις ρυθμίσεις του Mosquitto MQTT έχουν εισαχθεί ονόματα χρήστη με αντίστοιχους κωδικούς τα οποία έχουν δικαιώματα σύνδεσης στο MQTT. Αν δε χρησιμοποιηθεί σωστό όνομα και κωδικός, η σύνδεση στο topic απορρίπτεται από το Mosquitto.

4.1.3 Ανανέωση δεδομένων του συστήματος μέσω κανόνων

Για την ανανέωση των δεδομένων μέσω κανόνων χρησιμοποιούνται οι παρακάτω ροές, οι οποίες τίθενται σε λειτουργία κάθε φορά που εισάγονται καινούρια δεδομένα στη βάση. Το REST API μπορεί να χρησιμοποιηθεί για ενεργοποίηση/απενεργοποίηση των κανόνων.



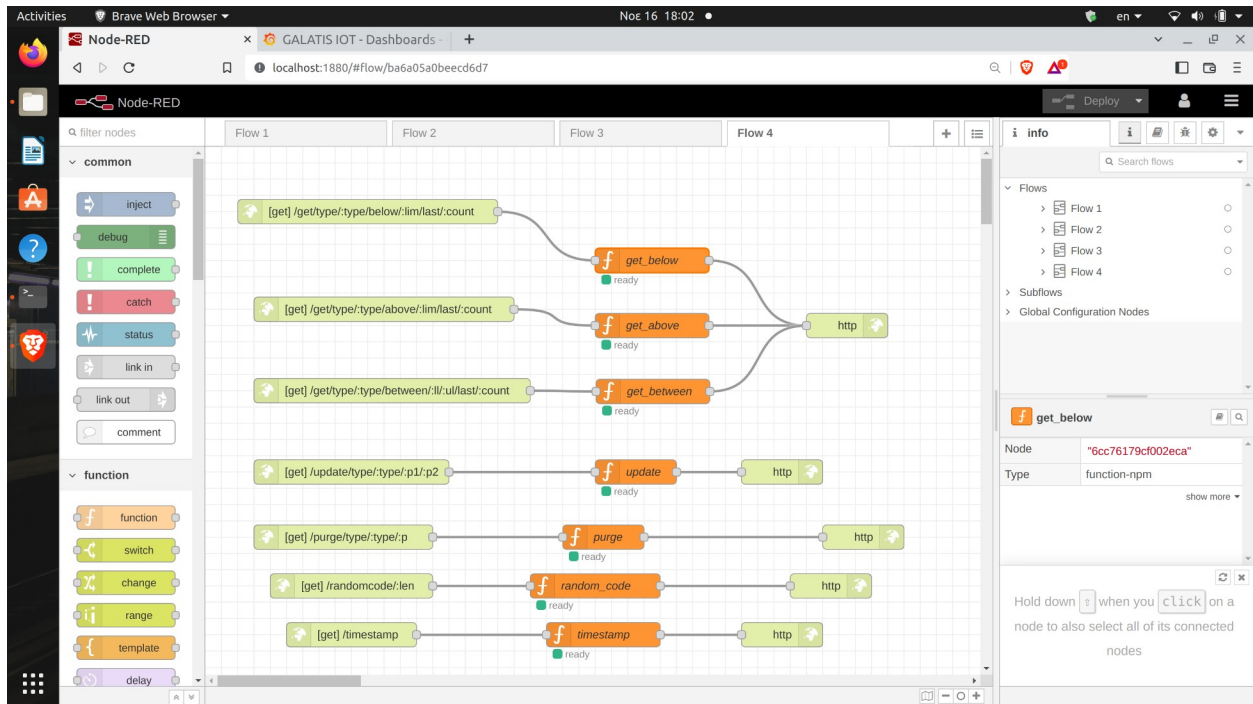
Οι ροές είναι οι εξής:

- `get/rules/enable/:id` , η οποία συνδέεται με τη συνάρτηση `rule_active` (η υλοποίηση της οποίας βρίσκεται στο παράρτημα), με σκοπό να ενεργοποιήσει τον κανόνα που αντιστοιχεί στο `id` που δόθηκε μέσω του URI
- `get/rules/disable/:id` , η οποία συνδέεται με τη συνάρτηση `rule_inactive` (η υλοποίηση της οποίας βρίσκεται στο παράρτημα), με σκοπό να απενεργοποιήσει τον κανόνα που αντιστοιχεί στο `id` που δόθηκε μέσω του URI
- `get/rules/enableAll` , η οποία συνδέεται με τη συνάρτηση `enable_rules` (η υλοποίηση της οποίας βρίσκεται στο παράρτημα), με σκοπό να ενεργοποιήσει όλους τους κανόνες
- `get/rules/disableAll` , η οποία συνδέεται με τη συνάρτηση `disable_rules` (η υλοποίηση της οποίας βρίσκεται στο παράρτημα), με σκοπό να απενεργοποιήσει όλους τους κανόνες

- `post/rules/add/:rulename` , η οποία συνδέεται με τη συνάρτηση `new_rule` (η υλοποίηση της οποίας βρίσκεται στο παράρτημα), με σκοπό να εισάγει νέο κανόνα στο σύστημα με τίτλο `rulename`

4.2 Ανάπτυξη REST API

Για την υποστήριξη του REST API έχουν δημιουργηθεί οι εξής ροές:



Οι ροές είναι οι εξής:

- `get/type/:type/below/:lim/last/:count`, η οποία εμφανίζει τις `:count` πιο πρόσφατες εγγραφές τύπου `:type` με έτος εγγραφής πριν το `:lim`

- `get/type/:type/above/:lim/last/:count`, η οποία εμφανίζει τις `:count` πιο πρόσφατες εγγραφές τύπου `:type` με έτος εγγραφής μετά το `:lim`

- `get/type/:type/between/:ll/:ul/last/:count`, η οποία εμφανίζει τις `:count` πιο πρόσφατες εγγραφές τύπου `:type` με έτος εγγραφής μεταξύ `:ll` και `:ul`

- `update/type/:type/:p1/:p2`, η οποία βρίσκει όλες τις εγγραφές οι οποίες έχουν τύπο `:type` και τιμή ατμοσφαιρικής πίεσης μεγαλύτερη από `:p1` και τη θέτει ίση με `:p2`

- `purge/type/:type/:limit`, η οποία βρίσκει όλες τις εγγραφές οι οποίες έχουν τύπο `:type` και τιμή ατμοσφαιρικής πίεσης μικρότερη από `:p1` και τις διαγράφει από τη βάση

- `randomcode/:len`, η οποία επιστρέφει μια τυχαία συμβολοσειρά από γράμματα και αριθμούς μήκους `:len`

- `timestamp`, η οποία επιστρέφει την ώρα τη στιγμή που καλείται σε μορφή `timestamp`

Πρέπει να τονιστεί ότι οι δύο τελευταίες ροές δεν έχουν καμία σχέση με τη βάση και υπάρχουν μόνο για να ελέγχει ο χρήστης ότι η σύνδεση με το server του Node-RED ήταν επιτυχής και ότι η επικοινωνία λειτουργεί όπως θα έπρεπε.

Επίσης, οι ροές που κάνουν `update / delete` σε εγγραφές στη βάση δε θα έπρεπε να χρησιμοποιούνται, τόσο γιατί οι εγγραφές της βάσης μας είναι χρονικές μετρήσεις και δεν υπάρχει λόγος να αφαιρεθούν, όσο και γιατί η επεξεργασία της βάσης μέσω του REST API θα πρέπει να αποφεύγεται. Ο μόνος λόγος για τον οποίο αναπτύχθηκαν είναι για να παρουσιαστεί ο τρόπος με τον οποίο θα λειτουργούσαν αυτές οι ροές, εάν υπήρχε λόγος χρήσης τους.

Στη συνέχεια, θα ακολουθήσουν εικόνες εκτέλεσης κάθε μίας από αυτές της ροές.

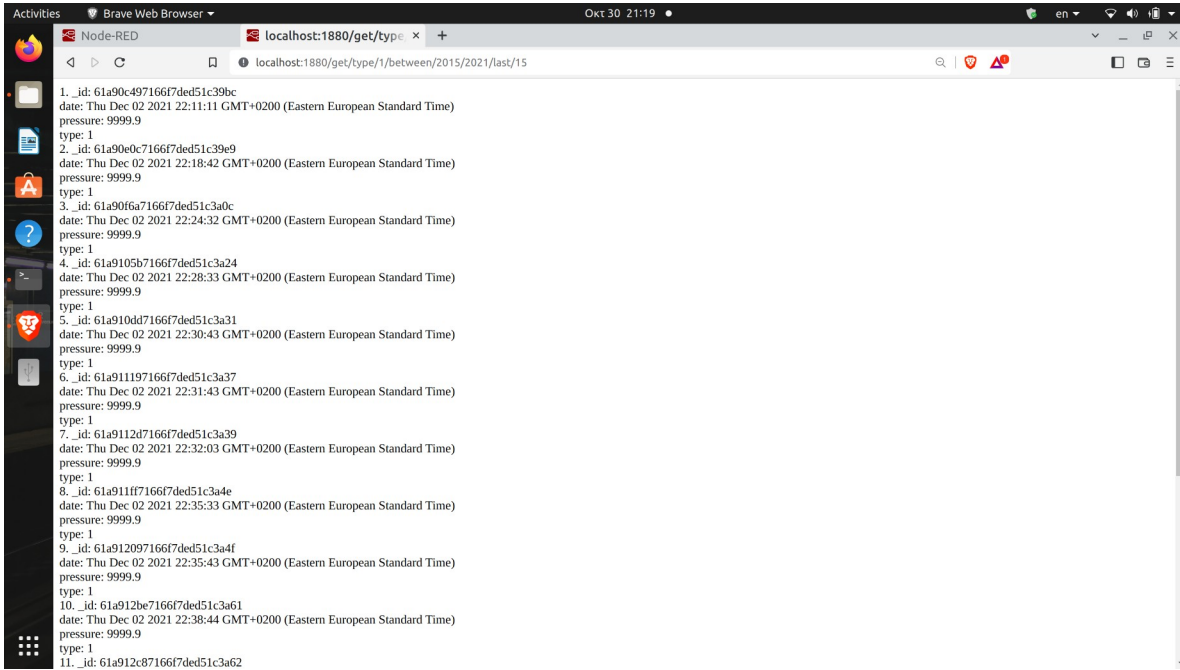
- Ζητούμε τις 15 πιο πρόσφατες εγγραφές τύπου 1 με έτος εγγραφής μετά το 2015


```
Node-RED localhost:1880/get/type/1/above/2015/last/15
localhost:1880/get/type/1/above/2015/last/15
1. _id: 61a90c497166f7ded51c39bc
date: Thu Dec 02 2021 22:11:11 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
2. _id: 61a90e0c7166f7ded51c39e9
date: Thu Dec 02 2021 22:18:42 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
3. _id: 61a90f6a7166f7ded51c3a0c
date: Thu Dec 02 2021 22:24:32 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
4. _id: 61a9105b7166f7ded51c3a24
date: Thu Dec 02 2021 22:28:33 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
5. _id: 61a910dd7166f7ded51c3a31
date: Thu Dec 02 2021 22:30:43 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
6. _id: 61a911197166f7ded51c3a37
date: Thu Dec 02 2021 22:31:43 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
7. _id: 61a9112d7166f7ded51c3a39
date: Thu Dec 02 2021 22:32:03 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
8. _id: 61a911ff7166f7ded51c3a4e
date: Thu Dec 02 2021 22:35:33 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
9. _id: 61a912097166f7ded51c3a4f
date: Thu Dec 02 2021 22:35:43 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
10. _id: 61a912be7166f7ded51c3a61
date: Thu Dec 02 2021 22:38:44 GMT+0200 (Eastern European Standard Time)
pressure: 9999.9
type: 1
11. _id: 61a912c87166f7ded51c3a62
```

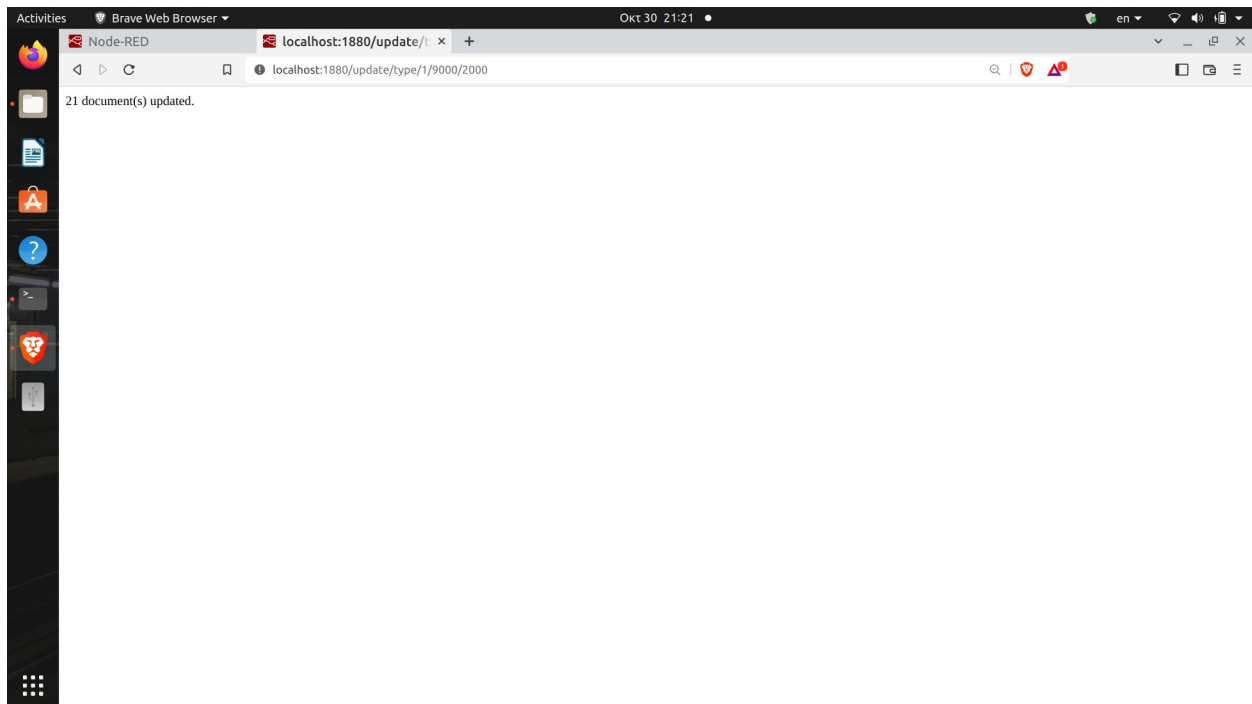
- Ζητούμε τις 15 πιο πρόσφατες εγγραφές τύπου 1 με έτος εγγραφής πριν το 2022

```
Node-RED localhost:1880/get/type/1/below/2015/last/15
localhost:1880/get/type/1/below/2015/last/15
1. _id: 61954b9842d734d300312d9c
date: Wed Nov 17 2021 22:35:58 GMT+0200 (Eastern European Standard Time)
pressure: 1015.3
type: 1
2. _id: 61954ba242d734d300312d9d
date: Wed Nov 17 2021 22:36:08 GMT+0200 (Eastern European Standard Time)
pressure: 1025.9
type: 1
3. _id: 61954bac42d734d300312d9e
date: Wed Nov 17 2021 22:36:18 GMT+0200 (Eastern European Standard Time)
pressure: 1030.8
type: 1
4. _id: 61954bb642d734d300312d9f
date: Wed Nov 17 2021 22:36:28 GMT+0200 (Eastern European Standard Time)
pressure: 1026.1
type: 1
5. _id: 61954bc142d734d300312da0
date: Wed Nov 17 2021 22:36:38 GMT+0200 (Eastern European Standard Time)
pressure: 1020.8
type: 1
6. _id: 61954bc42d734d300312da1
date: Wed Nov 17 2021 22:36:49 GMT+0200 (Eastern European Standard Time)
pressure: 1033.4
type: 1
7. _id: 61954bd542d734d300312da2
date: Wed Nov 17 2021 22:36:59 GMT+0200 (Eastern European Standard Time)
pressure: 1003.6
type: 1
8. _id: 61954be942d734d300312da4
date: Wed Nov 17 2021 22:37:19 GMT+0200 (Eastern European Standard Time)
pressure: 1028.2
type: 1
9. _id: 61954bf342d734d300312da5
date: Wed Nov 17 2021 22:37:29 GMT+0200 (Eastern European Standard Time)
pressure: 1018.5
type: 1
10. _id: 61954bfd42d734d300312da6
date: Wed Nov 17 2021 22:37:39 GMT+0200 (Eastern European Standard Time)
pressure: 1032.6
type: 1
11. _id: 61954c0742d734d300312da7
```

- Ζητούμε τις 15 πιο πρόσφατες εγγραφές τύπου 1 με έτος εγγραφής μεταξύ του 2015 και του 2021

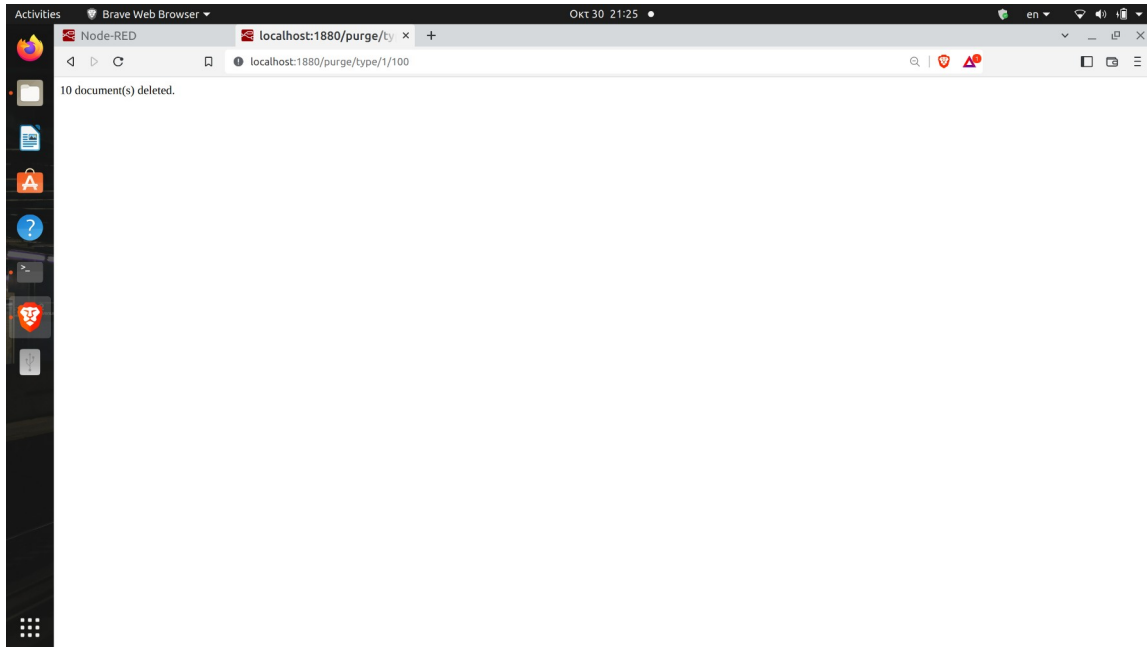


Ζητούμε την αλλαγή όλων των εγγραφών τύπου 1 και ατμοσφαιρικής τιμής μεγαλύτερης του 9000 σε τιμή ίση με 2000



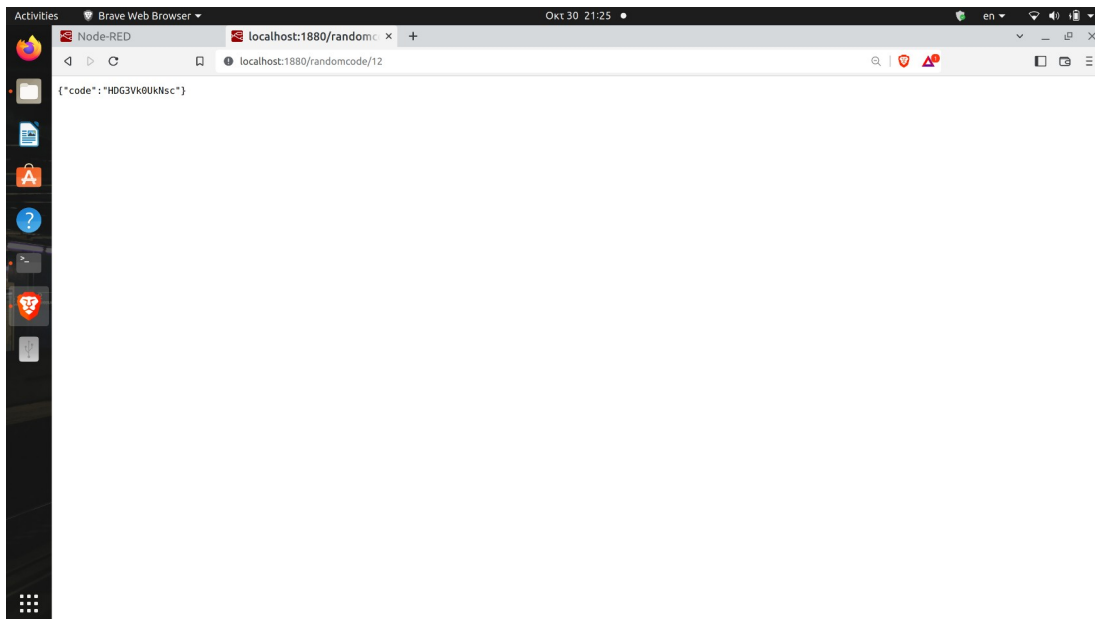
Παρατηρούμε ότι το REST API δεν εμφανίζει τις εγγραφές που επηρεάστηκαν από την εκτέλεση του ερωτήματος, παρά μόνο το πλήθος των εγγραφών αυτών.

- Ζητούμε την διαγραφή όλων των εγγραφών τύπου 1 και ατμοσφαιρικής τιμής μικρότερης του 100

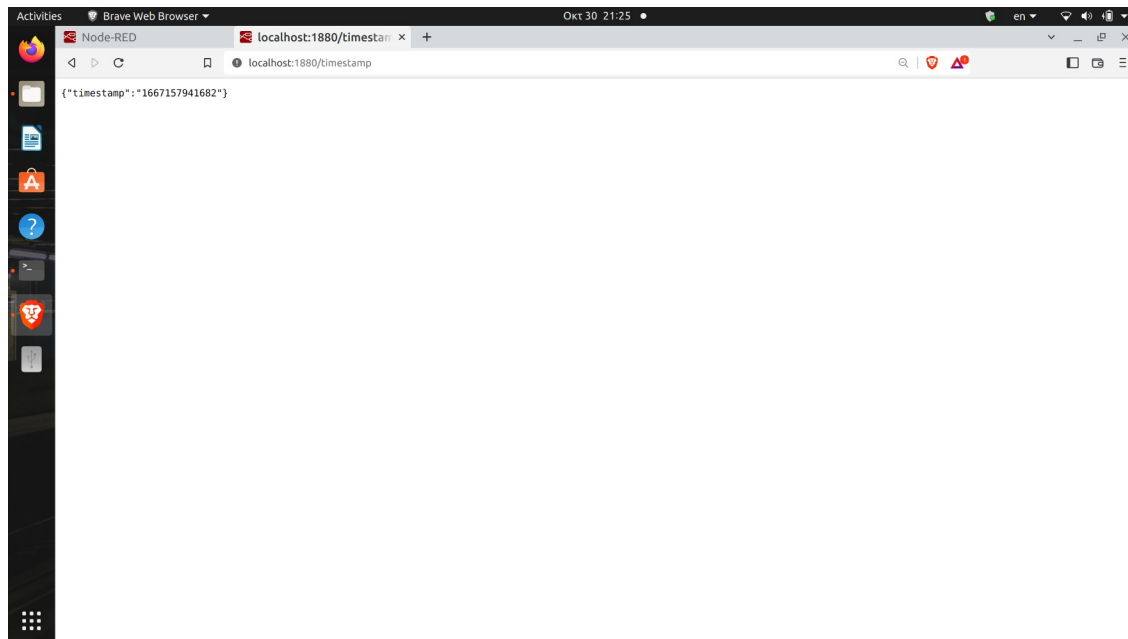


Όπως και πριν, εμφανίζεται μόνο το πλήθος των εγγραφών που αφαιρέθηκαν από τη βάση και όχι κάθε μια εγγραφή ξεχωριστά.

- Ζητούμε μια τυχαία γραμματοσειρά μήκους 12



- Ζητούμε τη χρονική στιγμή σε μορφή timestamp



5. Συμπεράσματα και Επεκτάσεις

5.1 Επίτευξη στόχων

Για να αξιολογήσουμε την επιτυχία του συγκεκριμένου project, θα εξετάσουμε κατά πόσο επιτεύχθηκαν οι αρχικοί στόχοι της διπλωματικής.

- Ο πρώτος στόχος της εφαρμογής του IoT ήταν η εμφάνιση των δεδομένων της στην οθόνη του χρήστη. Ο στόχος αυτός επιτεύχθηκε, όπως φαίνεται από τις εικόνες της οθόνης, στις οποίες απεικονίζονται γραφήματα αναπαράστασης της θερμοκρασίας και διάφορα κουμπιά ενεργειών.

- Ο δεύτερος στόχος ήταν η κατασκευή της εφαρμογής με τέτοιο τρόπο, ώστε να είναι εύχρηστη και φιλική προς το χρήστη. Η εφαρμογή αυτή παρέχει κουμπιά και γραφήματα εύκολα στην κατανόηση, ενώ δεν απαιτείται γνώση των τεχνολογιών που έχουν επιλεγεί για τη δημιουργία της, οπότε αυτός ο στόχος ήταν επιτυχής.

- Ο τρίτος στόχος σχετιζόταν με τη δυνατότητα της εφαρμογής να υποστηρίξει την εξόρυξη δεδομένων σε μορφή BSON μέσω ενός RESTful API. Προφανώς, αυτός ο στόχος επιτεύχθηκε, αφού, όπως είναι ξεκάθαρο από τις εικόνες από τις παραπάνω εικόνες, η εφαρμογή δίνει αυτή τη δυνατότητα στο χρήστη.

- Στον τέταρτο στόχο είχε τονιστεί η σημασία της ασφάλειας στην ανταλλαγή δεδομένων μεταξύ πελάτη και εξυπηρετητή. Παρόλο που έχουν υλοποιηθεί ζευγάρια ονομάτων και κωδικών για τη χρήση του Mosquitto MQTT και του Node-Red, οι κωδικοί αυτοί δεν είναι αρκετά προστατευμένοι, ώστε να μπορούν να χρησιμοποιηθούν σε επαγγελματική εφαρμογή. Επομένως, ο συγκεκριμένος στόχος δεν έχει επιτευχθεί σε ικανοποιητικό βαθμό.

- Ο πέμπτος στόχος σχετιζόταν με την απαίτηση των χρηστών από την εφαρμογή να διατηρεί την ταχύτητα απόκρισης, ακόμη και αν αυξηθεί το πλήθος των δεδομένων στη βάση. Καθώς για τις ανάγκες της εφαρμογής χρησιμοποιήθηκε η βάση MongoDB, η οποία χαρακτηρίζεται για την αποδοτική διάσχιση δεδομένων, διαπιστώνουμε ότι ο συγκεκριμένος στόχος ήταν επιτυχής και το αποτέλεσμα θα ήταν το ίδιο ακόμη και αν είχε αυξηθεί ο όγκος των δεδομένων της βάσης.

- Τέλος, ο έκτος στόχος ήταν η δυνατότητα επέκτασης της εφαρμογής, ώστε να προσαρμόζεται στις ανάγκες εταιρίας που υποστηρίζει πλήθος εγκαταστάσεων ως cluster. Αν και η δομή της εφαρμογής είναι εφαρμόσιμη και σε μεγαλύτερο επίπεδο, το πρόγραμμα εισροής δεδομένων Node-Red δεν είναι βέλτιστο για χρήση σε clusters από πολλά IoT, καθώς το Apache NiFi είναι πιο ικανό να διαχειριστεί ροές από πολλές διαφορετικές πηγές. Ως εκ τούτου, ο στόχος αυτός έχει υλοποιηθεί εν μέρει.

5.2 Επεκτάσεις

Ως πιθανές επεκτάσεις της εφαρμογής μπορούμε να προτείνουμε τα επόμενα:

5.2.1 Συστήματα ασφαλείας

Μια επέκταση θα ήταν η σχεδίαση συστημάτων ασφαλείας, με χρήση κατάλληλων πρωτοκόλλων κωδικοποίησης, ώστε να είναι ανθεκτικά στις επιθέσεις κακόβουλων χρηστών. Η αλλαγή αυτή θα καθιστούσε την εφαρμογή ικανή να υποστηρίξει τις ανάγκες των χρηστών εφαρμογών IoT.

5.2.2 Χρήση Apache NiFi αντί για Node-RED

Μια άλλη επέκταση θα ήταν η αλλαγή του προγράμματος εισροής δεδομένων από το Node-Red στο Apache NiFi, το οποίο έχει περισσότερες δυνατότητες διαχείρισης εφαρμογών μεγαλύτερης κλίμακας.

Για την επίτευξη της διαχείρισης δεδομένων μεγαλύτερης κλίμακας θα μπορούσαν να υλοποιηθούν επίσης οι επόμενες τεχνικές:

5.2.3 Ισοζύγιση φορτίου (load balancing)

Καθώς αυξάνεται η καταπόνηση ενός ιστότοπου ή μιας επιχειρηματικής εφαρμογής, τελικά, ένας μόνο διακομιστής δεν μπορεί να υποστηρίξει τον πλήρη φόρτο εργασίας. Για να ανταποκριθούν στη ζήτηση, οι οργανισμοί κατανέμουν το φόρτο εργασίας σε πολλούς διακομιστές. Η πρακτική αυτή, που ονομάζεται "εξισορρόπηση φορτίου", αποτρέπει την υπερφόρτωση ενός μόνο διακομιστή, η οποία θα μπορούσε να τον κάνει να επιβραδύνει, να απορρίπτει αιτήσεις και ακόμη και να καταρρεύσει.

Από τη σκοπιά του χρήστη, η εξισορρόπηση φορτίου λειτουργεί ως ένας αόρατος διαμεσολαβητής που βρίσκεται μεταξύ ενός πελάτη και μιας ομάδας διακομιστών, διασφαλίζοντας ότι τα αιτήματα σύνδεσης δεν χάνονται. Χωρίς την εξισορρόπηση φορτίου, οι εφαρμογές, οι ιστότοποι, οι βάσεις δεδομένων και οι διαδικτυακές υπηρεσίες πιθανότατα θα αποτύγχαναν όταν η ζήτηση γινόταν πολύ υψηλή [33]. Ένας μοναδικός ιστότοπος υψηλής επισκεψιμότητας μπορεί να δέχεται εκατοντάδες ή χιλιάδες αιτήματα χρηστών ταυτόχρονα. Χρειάζεται πολλαπλούς διακομιστές για να γεμίσουν με ακρίβεια τις ιστοσελίδες με τις ζητούμενες πληροφορίες, συμπεριλαμβανομένου του κειμένου, των φωτογραφιών, του βίντεο και της ροής ήχου.

Η εξισορρόπηση φορτίου εκτελεί αυτές τις κρίσιμες εργασίες:

- Διαχειρίζεται τις αιχμές κυκλοφορίας και αποτρέπει τις αιχμές σε έναν μόνο διακομιστή

- Ελαχιστοποιεί τον χρόνο απόκρισης των αιτημάτων των χρηστών
- Εξασφαλίζει την απόδοση και την αξιοπιστία των υπολογιστικών πόρων, τόσο των φυσικών όσο και των εικονικών
- Προσθέτει πλεονασμό και ανθεκτικότητα σε υπολογιστικά περιβάλλοντα

5.2.4 Χρήση Apache Kafka αντί για MQTT

Η Kafka συνδυάζει τρεις βασικές δυνατότητες, ώστε να μπορούν να υλοποιηθούν οι περιπτώσεις χρήσης για ροή συμβάντων από άκρη σε άκρη με μια ενιαία δοκιμασμένη λύση:

Δημοσίευση και να εγγραφή σε ροές συμβάντων, συμπεριλαμβανομένης της συνεχούς εισαγωγής/εξαγωγής των δεδομένων από άλλα συστήματα.

Αποθήκευση ροών συμβάντων με διάρκεια και αξιοπιστία.

Επεξεργασία ροών γεγονότων καθώς συμβαίνουν ή εκ των υστέρων.

Το Kafka μπορεί να αναπτυχθεί σε υλικό bare-metal, σε εικονικές μηχανές και σε κοντέινερ, και σε εγκαταστάσεις, καθώς και στο cloud. Υπάρχει δυνατότητα επιλογής μεταξύ της αυτοδιαχείρισης των περιβαλλόντων Kafka και της χρήσης πλήρως διαχειριζόμενων υπηρεσιών που προσφέρονται από διάφορους προμηθευτές.

Εξυπηρετητές: Το Kafka εκτελείται ως συστάδα ενός ή περισσότερων διακομιστών που μπορεί να καλύπτει πολλαπλά κέντρα δεδομένων ή περιοχές cloud. Ορισμένοι από αυτούς τους διακομιστές αποτελούν το επίπεδο αποθήκευσης, που ονομάζεται brokers. Άλλοι διακομιστές εκτελούν το Kafka Connect για τη συνεχή εισαγωγή και εξαγωγή δεδομένων ως ροές συμβάντων για την ενσωμάτωση του Kafka με τα υπάρχοντα συστήματα, όπως οι σχεσιακές βάσεις δεδομένων, καθώς και με άλλες συστάδες Kafka. Ένα σύμπλεγμα Kafka είναι εξαιρετικά κλιμακούμενο και ανεκτικό σε σφάλματα: αν κάποιος από τους διακομιστές του αποτύχει, οι άλλοι διακομιστές θα αναλάβουν το έργο τους για να διασφαλίσουν συνεχή λειτουργία χωρίς απώλεια δεδομένων.

Πελάτες: Επιτρέπουν την υλοποίηση κατανεμημένων εφαρμογών και μικρουπηρεσιών που διαβάζουν, γράφουν και επεξεργάζονται ροές συμβάντων παράλληλα, σε κλίμακα και με ανοχή σε σφάλματα, ακόμη και σε περίπτωση προβλημάτων δικτύου ή βλαβών μηχανών. Η Kafka συνοδεύεται από μερικούς τέτοιους πελάτες που περιλαμβάνονται, οι οποίοι συμπληρώνονται από δεκάδες πελάτες που παρέχονται από την κοινότητα της Kafka: υπάρχουν πελάτες για Java και Scala, συμπεριλαμβανομένης της βιβλιοθήκης Kafka Streams υψηλότερου επιπέδου, για Go, Python, C/C++ και πολλές άλλες γλώσσες προγραμματισμού, καθώς και REST APIs [34].

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Hendrickson, J. (2019). What is a Smart Home? Ανάκτηση από How-To-Geek:

<https://www.howtogeek.com/405693/what-is-a-smarthome/>

[2] Internet of Things in Smart Home (n.d.). Ανάκτηση από SCAND:

<https://scand.com/company/blog/internet-of-things-in-smart-home/>

[3] What is IoT? (n.d.). Ανάκτηση από Oracle:

<https://www.oracle.com/internet-of-things/what-is-iot/>

[4] 20 Smart Home Ideas to Use for Your Build (n.d.). Ανάκτηση από Madison Homebuilders:

<https://madisonhomebuilders.com/blog/smart-home-ideas-build/>

[5] Single-board computer (n.d.). Ανάκτηση από Wikipedia:

https://en.wikipedia.org/wiki/Single-board_computer

[6] What is a Database? (n.d.). Ανάκτηση από Oracle:

<https://www.oracle.com/database/what-is-database/>

[7] Message broker (n.d.). Ανάκτηση από Wikipedia:

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

[8] Odunlade, E. (2018). Getting Started with the Arduino IoT Cloud Ανάκτηση από Electronics-lab:

<https://www.electronics-lab.com/project/getting-started-arduino-iot-cloud/>

[9] Fromaget, P. (2017). Raspberry Pi vs Arduino: 7 Differences you should know Ανάκτηση από Raspberry Tips:

<https://raspberrytips.com/raspberry-pi-vs-arduino/>

[10] Sathyanarayanan, A. (2019). Raspberry Pi vs Arduino Ανάκτηση από EDUCBA:

<https://www.educba.com/raspberry-pi-vs-arduino/>

[11] NoSQL (n.d.). Ανάκτηση από Wikipedia:

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

[12] Time series database (n.d.). Ανάκτηση από Wikipedia:

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

[13] Dix, P. (2019). Time series database (TSDB) explained Ανάκτηση από influxdata:

<https://www.influxdata.com/time-series-database/>

[14] Smith, M. , Gargiulo, M. (2021). MongoDB's New Time Series Collections Ανάκτηση από MongoDB:

<https://www.mongodb.com/developer/products/mongodb/new-time-series-collections/>

[15] NoSQL vs. SQL databases (n.d.). Ανάκτηση από MongoDB:

<https://www.mongodb.com/nosql-explained/nosql-vs-sql>

[16] Advantages of MongoDB (n.d.). Ανάκτηση από MongoDB:

<https://www.mongodb.com/advantages-of-mongodb>

[17] Node-RED (n.d.). Ανάκτηση από Unipi:

<https://www.unipi.technology/products/node-red-66>

[18] Carder, J. (2021). Apache NiFi Overview: Key Benefits and Features Ανάκτηση από OpenLogic:

<https://www.openlogic.com/blog/apache-nifi-overview>

[19] Basu, K. (2015). What is StreamSets? Ανάκτηση από StreamSets:

<https://streamsets.com/blog/what-is-streamsets/>

[20] Node-RED vs Apache NiFi (2018). Ανάκτηση από Reddit:

https://www.reddit.com/r/bigdata/comments/8zeh2w/node_red_vs_apache_nifi/

[21] Zigbee (n.d.). Ανάκτηση από Wikipedia:

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

[22] Advanced Message Queuing Protocol (n.d.). Ανάκτηση από Wikipedia:

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

[23] AMQP: Introducing the Advanced Message Queuing Protocol (2019). Ανάκτηση από IONOS:

<https://www.ionos.com/digitalguide/websites/web-development/advanced-message-queueing-protocol-amqp/>

[24] MQTT (n.d.). Ανάκτηση από Wikipedia:

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

[25] What is MQTT? Definition and Details (n.d.). Ανάκτηση από Paessler:

<https://www.paessler.com/it-explained/mqtt>

[26] Pedamkar, P. (2018). AMQP vs MQTT Ανάκτηση από EDUCBA:

<https://www.educba.com/amqp-vs-mqtt/>

[27] Kamal, R. (2021). AMQP vs MQTT: IoT Protocols You Must Know About Ανάκτηση από Intuz:

<https://www.intuz.com/blog/mqtt-vs-amqp-iot-protocols-you-must-know>

[28] Pedamkar, P. (2019). What is RabbitMQ? Ανάκτηση από EDUCBA:

<https://www.educba.com/what-is-rabbitmq/>

[29] Mishra, H. (2019). HiveMQ Broker Overview Ανάκτηση από IoTBYHVM:

<https://iotbyhvm.ooo/hivemq-mqtt-broker/>

[30] Eclipse Mosquitto (n.d.). Ανάκτηση από Mosquitto:

<https://mosquitto.org/>

[31] Eclipse Mosquitto (n.d.). Ανάκτηση από Eclipse Foundation:

<https://projects.eclipse.org/projects/iot.mosquitto>

[32] Grafana vs Tableau (n.d.). Ανάκτηση από EDUCBA:

<https://www.educba.com/grafana-vs-tableau/>

[33] Load Balancing (n.d.). Ανάκτηση από IBM:

<https://www.ibm.com/topics/load-balancing>

[34] Apache Kafka (n.d.). Ανάκτηση από kafka:

<https://kafka.apache.org/intro>