



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ:

ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ ΣΕ ΔΙΚΤΥΑ ΝΕΑΣ ΓΕΝΙΑΣ

**Συλλογή IoT δεδομένων και σημασιολογικός εμπλουτισμός
τους με εφαρμογή στην παρακολούθηση ποιότητας αέρα σε
αστικό περιβάλλον**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Δημήτρη Λυμπέρη

Επιβλέπων : Χρήστος Γκουμόπουλος, Αναπληρωτής Καθηγητής

Μέλη εξεταστικής επιτροπής: Κωστούλας Θεόδωρος, Αναπληρωτής Καθηγητής
Σκούτας Δημήτριος, Επίκουρος Καθηγητής

Σάμος, Φεβρουάριος 2023

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε το ακαδημαϊκό έτος 2022-2023 στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών «Διαδίκτυο των Πραγμάτων: Ευφυή Περιβάλλοντα σε Δίκτυα Νέας Γενιάς» της Πολυτεχνικής Σχολής του Πανεπιστημίου Αιγαίου, του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων. Η εργασία πραγματεύεται την ανάπτυξη ενός ολοκληρωμένου πλαισίου συλλογής και σημασιολογικής επισημείωσης IoT δεδομένων από ετερογενείς συσκευές, μελετώντας την εφαρμογή του στην παρακολούθηση της ποιότητας αέρα σε αστικό περιβάλλον.

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Γκουμόπουλο Χρήστο για την εμπιστοσύνη που έδειξε στο πρόσωπό μου και για την δυνατότητα που μου έδωσε να ασχοληθώ με ένα πολυδιάστατο θέμα, εφαρμόζοντας ένα μεγάλο μέρος των γνώσεων που απέκτησα κατά την διάρκεια των σπουδών μου στο μεταπτυχιακό πρόγραμμα. Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα τους φίλους μου Ζαχαρία και Μανώλη για την στήριξη και την συμπαράστασή τους σε αυτή μου την προσπάθεια.

Τέλος, θα ήθελα να εκφράσω την απέραντη ευγνωμοσύνη μου στην οικογένειά μου την Αντιγόνη, τον Γιάννη και τον Μιχάλη για την κατανόηση, τη συμπαράσταση και την αμέριστη υποστήριξή τους, υπήρξαν ανεκτίμητο στήριγμα καθ' όλη τη διάρκεια των σπουδών μου.

© 2023

του

ΔΗΜΗΤΡΗ ΛΥΜΠΙΕΡΗ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Εφαρμογές έξυπνων πόλεων	1
1.2	Αντικείμενο διπλωματικής.....	2
1.2.1	Κίνητρο για τη διεξαγωγή της εργασίας	2
1.2.2	Στόχοι και οφέλη εφαρμογής	3
1.2.3	Ευρωπαϊκός δείκτης ποιότητας αέρα (AQI).....	4
1.3	Δομή της διπλωματικής	6
2	Βιβλιογραφική Επισκόπηση	8
3	Αρχιτεκτονική	16
3.1	Γενική αρχιτεκτονική μεθοδολογίας.....	16
3.2	Αρχιτεκτονική σε επίπεδο υλοποίησης.....	18
4	ΙοΤ Πλατφόρμες Υλικού	20
4.1	Arduino UNO WIFI REV2	20
4.2	Arduino MEGA 2560 REV3.....	22
4.3	TTGO LoRa32 v1.0	24
4.4	LoPy4.....	26
4.5	Σύγκριση Μικροελεγκτών	28
5	Δίκτυα Επικοινωνίας	30
5.1	Wi-Fi	30
5.2	Ethernet.....	31
5.3	LoRa/LoRaWAN	32
5.3.1	Φυσικό επίπεδο LoRa.....	32
5.3.2	LoRaWAN επίπεδο MAC.....	34
5.3.3	Τοπολογία LoRaWAN.....	36
5.3.4	The Things Network.....	37
5.4	Πρωτόκολλα Επιπέδου Εφαρμογής.....	39
5.4.1	HTTP.....	39
5.4.2	MQTT.....	40
6	ΙοΤ Πλατφόρμες Λογισμικού	45
6.1	Thingspeak.....	45
6.2	Ubidots.....	47

6.3	Σύγκριση Thingspeak – Ubidots	48
7	Σημασιολογικός Ιστός	50
7.1	Εισαγωγή.....	50
7.2	Οντολογίες	52
7.3	Διαδίκτυο των Πραγμάτων και σημασιολογία.....	53
7.4	Neo4j.....	55
8	Περιβαλλοντικοί και Γεωχωρικοί Αισθητήρες.....	56
8.1	Αισθητήρας αιωρούμενων σωματιδίων PMS5003	56
8.2	Αισθητήρες αερίων MQ.....	58
8.2.1	Μέτρηση O ₃ και NO ₂ - MQ131.....	59
8.2.2	Μέτρηση SO ₂ - MQ136.....	60
8.3	Αισθητήρας GPS.....	61
9	Μελέτη περίπτωσης «Πράσινη Διαδρομή» βασισμένη στην γενική αρχιτεκτονική.....	65
9.1	Επίπεδο πηγής IoT δεδομένων.....	65
9.1.1	<i>IoT Κόμβοι</i>	66
9.1.2	<i>Βαθμονόμηση αισθητήρων αερίων MQ</i>	68
9.1.3	<i>Περιγραφή λειτουργίας IoT κόμβων</i>	72
9.2	Επίπεδο πηγής ανοικτών δεδομένων	76
9.3	Επίπεδο δικτύου	82
9.3.1	<i>Πύλη δικτύωσης Lorix One WiFX</i>	82
9.3.2	<i>Καταχώρηση Lorix One WiFX στο TTS</i>	83
9.3.3	<i>Ρύθμιση Lorix One WiFX</i>	86
9.3.4	<i>Σύνδεση IoT κόμβων στο The Things Stack</i>	90
9.3.5	<i>Μορφοποίηση ωφέλιμης πληροφορίας στο TTS</i>	93
9.3.6	<i>Διασύνδεση TTS με τις IoT πλατφόρμες</i>	95
9.4	Επίπεδο IoT platforms	100
9.5	Επίπεδο υπηρεσιών	104
9.5.1	<i>Περιγραφή λειτουργίας υπηρεσίας συλλογής IoT δεδομένων</i>	104
9.5.2	<i>Περιγραφή λειτουργίας υπηρεσίας συλλογής OpenData δεδομένων</i>	109
9.6	Επίπεδο Σημασιολογίας.....	112
9.6.1	<i>Οντολογία εφαρμογής</i>	112
9.6.2	<i>Περιγραφή λειτουργίας υπηρεσίας σημασιολογικής επισημείωσης δεδομένων</i>	116
9.7	Επίπεδο Εφαρμογής.....	120
9.7.1	<i>Περιγραφή εφαρμογής «Πράσινης Διαδρομής»</i>	120
9.7.2	<i>GraphQL</i>	122

9.7.3	<i>GraphQL Schema</i>	123
9.7.4	<i>Apollo Server - Client</i>	125
9.7.5	<i>Queries-Mutations εφαρμογής</i>	126
9.7.6	<i>FrontEnd</i>	132
10	Συμπεράσματα	137
	Βιβλιογραφία	139

Λίστα Σχημάτων

Σχήμα 1: Ευρωπαϊκός δείκτης ποιότητας αέρα.	5
Σχήμα 2: Ζώνες δεικτών και μηνύματα ενημέρωσης πληθυσμού.	6
Σχήμα 3: Προτεινόμενη οντολογία μελέτης [6].	9
Σχήμα 4: Αρχιτεκτονική συστήματος [8] [9].	10
Σχήμα 5: Προτεινόμενη οντολογία μελέτης [10].	11
Σχήμα 6: Προτεινόμενη οντολογία μελέτης [11].	12
Σχήμα 7: Οντολογία περιβαλλοντικής δομής έξυπνης πόλης [12].	13
Σχήμα 8: Προτεινόμενο μοντέλο μελέτης [13].	14
Σχήμα 9: Γενική αρχιτεκτονική ανάπτυξης εφαρμογών στηριζόμενη στη συλλογή ετερογενών δεδομένων και στον σημασιολογικό εμπλουτισμό τους.	18
Σχήμα 10: Αρχιτεκτονική σε επίπεδο εφαρμογής.	19
Σχήμα 11: Arduino UNO WIFI REV2.	20
Σχήμα 12: Διάγραμμα ακροδεκτών Arduino UNO WiFi Rev.2	21
Σχήμα 13: Arduino MEGA 2560 REV3.	22
Σχήμα 14: Διάγραμμα ακροδεκτών Arduino MEGA 2560 REV3.	23
Σχήμα 15: TTGO LoRa32 v1.0	24
Σχήμα 16: Διάγραμμα ακροδεκτών TTGO LoRa32.	25
Σχήμα 17: Lopy4.	26
Σχήμα 18: Διάγραμμα ακροδεκτών Lopy4.	27
Σχήμα 19: Στοιβά πρωτοκόλλου LoRa και LoRaWAN.	34
Σχήμα 20: Παράθυρα λήψης κλάσης A.	35
Σχήμα 21: Παράθυρα λήψης κλάσης B.	35
Σχήμα 22: Παράθυρα λήψης κλάσης C.	36
Σχήμα 23: Τυπική υλοποίηση δικτύου LoRaWAN.	37
Σχήμα 24: Αρχιτεκτονική TTN.	38
Σχήμα 25: Μοντέλο Publish/Subscribe - MQTT.	41
Σχήμα 26: Αντικείμενο JSON.	42
Σχήμα 27: MQTT (Επίπεδο Εφαρμογής) στη στοιβά TCP/IP.	42
Σχήμα 28: MQTT/TLS (Επίπεδο Εφαρμογής) στη στοιβά TCP/IP.	43
Σχήμα 29: Κανάλι συλλογής δεδομένων 7 μεταβλητών.	46
Σχήμα 30: Χρονοσειρές δεδομένων στο Thingspeak.	46
Σχήμα 31: Κανάλι συσκευής aegean-lopy4-node1 στο Ubidots.	47
Σχήμα 32: Εξατομικευμένη σελίδα παρακολούθησης δεδομένων στο Ubidots.	48
Σχήμα 33: Οπτικοποιήσεις δεδομένων στο Thingspeak μέσω MATLAB.	49
Σχήμα 34: Παραδείγματα απλής τριπλέτας και συνδυασμού τριπλετών.	51
Σχήμα 35: Παράδειγμα απεικόνισης γραφοδεδομένων στη Neo4j.	55
Σχήμα 36: Αισθητήρας αιωρούμενων σωματιδίων PMS5003.	57
Σχήμα 37: Διάγραμμα λειτουργίας αισθητήρα αιωρούμενων σωματιδίων.	57
Σχήμα 38: Αισθητήρες αερίων MQ.	58
Σχήμα 39: Ηλεκτρονικό διάγραμμα αισθητήρων MQ.	58
Σχήμα 40: Εσωτερική δομή αισθητήρα MQ.	59

Σχήμα 41: Αισθητήρας όζοντος MQ131	60
Σχήμα 42: Αισθητήρας μέτρησης SO ₂ , NO ₂ - MQ136	61
Σχήμα 43: Μονάδα GPS VK2828U8G5LF	62
Σχήμα 44: Διάγραμμα διασύνδεσης VK2828U8G5LF	63
Σχήμα 45: Σχηματικό διάγραμμα συσκευής (Τροφοδοτικό, αναπτυξιακή πλακέτα).....	67
Σχήμα 46: Σχηματικό διάγραμμα αισθητήρων.....	67
Σχήμα 47: Πρωτότυπο IoT κόμβου.....	68
Σχήμα 48: Διάγραμμα κυκλώματος προσαρμογής τάσης για τον αισθητήρα MQ.....	69
Σχήμα 49: Διάγραμμα ευαισθησίας αισθητήρα MQ131.	70
Σχήμα 50: Χαρακτηριστικές συγκέντρωσης αερίων με power regression.	71
Σχήμα 51: Διάγραμμα ροής λειτουργίας IoT κόμβων.....	73
Σχήμα 52: Εφαρμογή median φίλτρου σε μετρήσεις αισθητήρα MQ.....	74
Σχήμα 53: Υποβολή δεδομένων στο Thingspeak με HTTP POST αίτημα.	75
Σχήμα 54: MQTT payload.....	75
Σχήμα 55: Παράδειγμα κωδικοποίησης μετρούμενων μεγεθών.	76
Σχήμα 56: Αίτημα κλήσης στο Weatherbit.	78
Σχήμα 57: Αποτέλεσμα αιτήματος κλήσης στο Weatherbit.....	78
Σχήμα 58: Αίτημα κλήσης στο open_weather.....	79
Σχήμα 59: Αποτέλεσμα αιτήματος κλήσης στο open_weather.	79
Σχήμα 60: Αίτημα κλήσης στο ninjas_airq.	80
Σχήμα 61: Αποτέλεσμα αιτήματος κλήσης στο ninjas_airq.....	80
Σχήμα 62: Αίτημα κλήσης στο iqair.....	81
Σχήμα 63: Αποτέλεσμα αιτήματος κλήσης στο iqair.	81
Σχήμα 64: Lorix One Gateway.....	82
Σχήμα 65: Lorix One PoE.	82
Σχήμα 66: TTN - Περιβάλλον πλατφόρμας.	83
Σχήμα 67: TTN – Επιλογή Cluster.....	84
Σχήμα 68: Περιβάλλον πλατφόρμας The Thing Stack.....	84
Σχήμα 69: Καταχώρηση EUI δικτυακής πύλης.....	85
Σχήμα 70: Καταχώρηση ονόματος συσκευής και πλάνου συχνότητας.....	85
Σχήμα 71: Δημιουργία κλειδιού API για την αυθεντικοποίηση της δικτυακής πύλης.....	86
Σχήμα 72: Πίνακας ελέγχου Lorix One.....	87
Σχήμα 73: Lorix One - Ρύθμιση ζώνης ώρας και NTP.	88
Σχήμα 74: Lorix One - Ρύθμιση συχνότητας LoRa και κέρδους κεραίας.....	88
Σχήμα 75: Lorix One - Ρύθμιση προγράμματος προώθησης πακέτων.....	89
Σχήμα 76: Lorix One - Ρυθμίσεις διακομιστή LoRaWAN.	89
Σχήμα 77: Δημιουργία application στο The Things Stack.	90
Σχήμα 78: Σελίδα επισκόπησης εφαρμογής.....	90
Σχήμα 79: Σύνολο υλοποιημένων εφαρμογών στο TTS.	91
Σχήμα 80: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 1).....	91
Σχήμα 81: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 2).....	92
Σχήμα 82: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 3).....	93
Σχήμα 83: Payload formatter για την IoT πλατφόρμα Ubidots.....	94

Σχήμα 84: Payload formatter για την IoT πλατφόρμα Thingspeak.	94
Σχήμα 85: Παράδειγμα αποκωδικοποίησης μηνύματος.	95
Σχήμα 86: Σελίδα επιλογής Webhook.	96
Σχήμα 87: Σελίδα API Keys καναλιού στο Thingspeak.	96
Σχήμα 88: Καρτέλα ρύθμισης Webhook Thingspeak.	97
Σχήμα 89: Επιλογή The Things Stack plugin στην πλατφόρμα Ubidots.	98
Σχήμα 90: Ρύθμιση The Thing Stack plugin στο Ubidots.	98
Σχήμα 91: Αναγνωριστικό του plugin The Things Stack.	99
Σχήμα 92: Καρτέλα ρύθμισης Webhook Ubidots.	99
Σχήμα 93: Διάγραμμα διασυνδεδεμένων συσκευών με την πλατφόρμα Thingspeak.	101
Σχήμα 94: Διάγραμμα διασυνδεδεμένων συσκευών με την πλατφόρμα Ubidots.	101
Σχήμα 95: Δημιουργία συσκευής MQTT στην πλατφόρμα Thingspeak.	102
Σχήμα 96: Διαπιστευτήρια συσκευής MQTT στην πλατφόρμα Thingspeak.	103
Σχήμα 97: Διαπιστευτήριο MQTT στην πλατφόρμα Ubidots.	104
Σχήμα 98: Απόσπασμα αρχείου ρυθμίσεων υπηρεσίας συλλογής IoT δεδομένων.	105
Σχήμα 99: Συνάρτηση υποβολής αιτημάτων ασύγχρονης σύνδεσης με τους διακομιστές MQTT.	106
Σχήμα 100: Μέθοδος επανάκλησης on_connect().	107
Σχήμα 101: Μέθοδος επανάκλησης on_message().	108
Σχήμα 102: Παραδείγματα επεξεργασμένων μηνυμάτων.	108
Σχήμα 103: Απόσπασμα αρχείου καταγραφής app.log.	108
Σχήμα 104: Απόσπασμα αρχείου ρυθμίσεων υπηρεσίας συλλογής OpenData δεδομένων.	110
Σχήμα 105: Μέθοδος δημιουργίας URL αιτημάτων.	110
Σχήμα 106: Συνάρτηση κύριου προγράμματος υπηρεσίας.	110
Σχήμα 107: Μέθοδος parse_api().	111
Σχήμα 108: Παραδείγματα επεξεργασμένων μηνυμάτων υπηρεσίας <i>async_api_request</i>	112
Σχήμα 109: Κώδικας προγραμματισμένης εκτέλεσης της εφαρμογής <i>async_api_request</i>	112
Σχήμα 110: Οντολογία Εφαρμογής.	113
Σχήμα 111: Ιεραρχία κλάσεων και ιδιότητες οντολογίας.	114
Σχήμα 112: Μέθοδοι διαχείρισης συνδέσεων με τη βάση Neo4j.	117
Σχήμα 113: Μέθοδος μοντελοποίησης δεδομένων από την πλατφόρμα Thingspeak.	117
Σχήμα 114: Μέθοδος μοντελοποίησης δεδομένων από την πλατφόρμα Ubidots.	118
Σχήμα 115: Μέθοδος μοντελοποίησης δεδομένων από τις ανοικτές πλατφόρμες.	118
Σχήμα 116: Μέθοδος εκτέλεσης συναλλαγής εγγραφής IoT δεδομένων.	119
Σχήμα 117: Μέθοδος εκτέλεσης συναλλαγής εγγραφής δεδομένων από ανοικτές πλατφόρμες.	120
Σχήμα 118: Αρχιτεκτονική εφαρμογής πλήρους στοίβας (Full Stack).	121
Σχήμα 119: Απόσπασμα από το GraphQL Schema της εφαρμογής.	124
Σχήμα 120: Στιγμιότυπο του Apollo Studio Explorer.	125
Σχήμα 121: Συναρτήσεις αντιστοίχισης της μετρούμενης συγκέντρωσης ρύπων με δείκτη τιμών 0-5.	127
Σχήμα 122: Συνάρτηση υπολογισμού του AQI για κάθε περιβαλλοντικό ρύπο.	127
Σχήμα 123: Ερώτημα GraphQL τύπου Query για την ανάκτηση περιβαλλοντικών και γεωχωρικών δεδομένων.	128
Σχήμα 124: Παράμετροι ερωτήματος (Σχήμα 123).	128

Σχήμα 125: Αίτημα ανάκτησης περιβαλλοντικών δεδομένων, μέσω της εφαρμογής GraphQL πλήρους στοίβας.	129
Σχήμα 126: Ερώτημα GraphQL τύπου Mutation για τον υπολογισμό την "Πράσινης Διαδρομής".	130
Σχήμα 127: Παράμετροι ερωτήματος (Σχήμα 126).....	130
Σχήμα 128: Γράφος διασυνδεδεμένων κόμβων με σχέσεις hasNeighbor.	131
Σχήμα 129: Απάντηση ερωτήματος υπολογισμού διαδρομής.....	132
Σχήμα 130: Αρχική οθόνη εφαρμογής.	134
Σχήμα 131: Παραδείγματα αναδύμενου παραθύρου απεικόνισης περιβαλλοντικών δεδομένων σταθμού.	135
Σχήμα 132: Επιλογή σημείων έναρξης και λήξης διαδρομής στο χάρτη.	135
Σχήμα 133: Απεικόνιση προτεινόμενης «Πράσινης Διαδρομής» στο χάρτη.	136

Λίστα Πινάκων

Πίνακας 1: Ατμοσφαιρικοί ρύποι και επιπτώσεις στην υγεία.	3
Πίνακας 2: Περίληψη σχετικών ερευνών.	15
Πίνακας 3: Τεχνικά Χαρακτηριστικά Arduino UNO WiFi Rev.2	21
Πίνακας 4: Τεχνικά Χαρακτηριστικά Arduino MEGA 2560 REV3.	23
Πίνακας 5: Τεχνικά χαρακτηριστικά TTGO LoRa32.	25
Πίνακας 6: Συγκριτικός πίνακας μικροελεγκτών εφαρμογής.	28
Πίνακας 7: Τεχνικά χαρακτηριστικά MQ131.....	60
Πίνακας 8: Τεχνικά χαρακτηριστικά MQ136.....	61
Πίνακας 9: Προτάσεις πρωτοκόλλου NMEA 0183.....	62
Πίνακας 10: Πίνακας αντιστοίχισης ακίδων VK2828U8G5LF.	63
Πίνακας 11: Τεχνικά Χαρακτηριστικά VK2828U8G5LF.....	64
Πίνακας 12: IoT Κόμβοι.....	65
Πίνακας 13 : Περιγραφή στοιχείων κυκλώματος προσαρμογής.	69
Πίνακας 14: Συντελεστές χαρακτηριστικών συγκέντρωσης αερίων.....	72
Πίνακας 15: Πηγές ανοικτών περιβαλλοντικών δεδομένων.	77
Πίνακας 16: Πίνακας τελικών σημείων MQTT στις IoT πλατφόρμες.	101
Πίνακας 17: Κλάσεις οντολογίας.	114
Πίνακας 18: Ιδιότητες αντικειμένων οντολογίας.	115
Πίνακας 19: Πίνακας Ιδιοτήτων Δεδομένων.....	115

Ακρωνύμια

<i>Συντ/φία</i>	<i>Επεξήγηση</i>
API	Application Programming Interface
AQI	Air Quality Index
ppm	parts per million
VOC	Volatile Organic Compounds
GPS	Global Positioning System
TTL	Transistor-Transistor Logic
GMT	Greenwich Mean Time
NMEA	National Marine Electronics Association
ASCII	American Standard Code for Information Interchange
IoT	Internet of Things
WiFi	Wireless Fidelity
EEPROM	Electrically Erasable Programmable Read-Only Memory
SoC	System on a Chip
ISM	Industrial, Scientific and Medical
IMU	Inertial Measurement Unit
BLE	Bluetooth Low Energy
UART	Universal Asynchronous Receiver and Transmitter
OSI	Open Systems Interconnection
AP	Access Point
PtP	Point to Point
URI	Unified Resource Identifier
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
TLS	Transport Layer Security
LPWA	Low-Power Wide-Area
ISM	Industrial, Scientific and Medical
CSS	Chirp Spread Spectrum
ADR	Adaptive Data Rate
MAC	Media Access Control
CSMA	Carrier-sense multiple access
QoS	Quality of Service
PoE	Power over Ethernet
EUI	Extended Unique Identifier

JSON	JavaScript Object Notation
OTAA	Over the Air Activation
TTN	The Things Network
TTS	The Things Stack
OWL	Ontology Web Language
RDF	Resource Description Framework
SOS	Sensor Observation Service
OGC	Open Geospatial Consortium
EISCO	Environment Indicators Smart City Ontology
ENVO	Environment Ontology
GCIO	Global City Indicator Ontology
SSN	Semantic Sensor Network
SCEO	Smart City Environmental Ontology
SDL	Schema Definition Language

Περίληψη

Η παρούσα μεταπτυχιακή εργασία εστιάζει στη βελτίωση της παρακολούθησης της ποιότητας του αέρα σε αστικά περιβάλλοντα, χρησιμοποιώντας δεδομένα που συλλέγονται από συσκευές IoT και εφαρμόζοντας τεχνικές σημασιολογικού εμπλουτισμού. Η εργασία χρησιμοποιεί τον Ευρωπαϊκό Δείκτη Ποιότητας Αέρα (EAQI) ως αναφορά, για την κατανόηση της σοβαρότητας της ατμοσφαιρικής ρύπανσης σε διάφορες περιοχές και την αναγνώριση των αιτιών που την προκαλούν.

Για την επίτευξη του σκοπού αυτού, συλλέγονται ετερογενή δεδομένα από IoT συσκευές και ανοικτές πλατφόρμες περιβαλλοντικών δεδομένων. Οι συσκευές IoT τοποθετούνται στρατηγικά στο αστικό περιβάλλον για τη συλλογή δεδομένων σχετικά με τις διάφορες μετρήσεις ποιότητας του αέρα, όπως η συγκέντρωση βλαβερών αέριων ρύπων (SO₂, NO₂, O₃) και τα επίπεδα των αιωρούμενων σωματιδίων (PM_{2.5}, PM₁₀). Αρχικά, τα δεδομένα αποστέλλονται σε IoT πλατφόρμες και στη συνέχεια μεταδίδονται μέσω του πρωτοκόλλου MQTT σε έναν κεντρικό διακομιστή. Στο σημείο αυτό εξατομικευμένες υπηρεσίες συλλέγουν γεωχωρικού τύπου περιβαλλοντικά δεδομένα από ανοικτές πλατφόρμες. Στο σύνολό τους τα συλλεχθέντα δεδομένα υποβάλλονται σε επεξεργασία και σημασιολογικό σχολιασμό εκχωρώντας κλάσεις και ιδιότητες από μια οντολογία.

Η χρήση τεχνικών σημασιολογικού εμπλουτισμού επιτρέπει μια πιο ολοκληρωμένη κατανόηση των δεδομένων, καθώς προσθέτει πλαίσιο και νόημα στα ακατέργαστα δεδομένα που συλλέγονται από διαφορετικές πηγές. Επιπλέον, αποθηκεύοντας τα σημασιολογικά δεδομένα σε μια βάση γραφοδεδομένων Neo4j, διευκολύνεται η αναγνώριση προτύπων και σχέσεων μέσα στα δεδομένα, καθιστώντας ευκολότερο τον εντοπισμό περιοχών ανησυχίας και πιθανών αιτιών ατμοσφαιρικής ρύπανσης.

Επίσης, η εργασία περιλαμβάνει την υλοποίηση διαδικτυακής εφαρμογής για την οπτικοποίηση των περιβαλλοντικών δεδομένων των σταθμών, υποβάλλοντας ερωτήματα GraphQL προς τη βάση. Αυτό επιτρέπει τον εύκολο εντοπισμό περιοχών με υψηλά επίπεδα ρύπανσης. Επιπρόσθετα, ο χρήστης μπορεί να αναζητήσει μια διαδρομή που θα ακολουθήσει, βάσει των προσφάτων περιβαλλοντικών δεδομένων ποιότητας αέρα, η οποία είναι απαλλαγμένη από ατμοσφαιρική μόλυνση, ορίζοντας τα σημεία έναρξης και λήξης της διαδρομής στο χάρτη.

Λέξεις Κλειδιά: Διαδίκτυο των Πραγμάτων, Σημασιολογικός εμπλουτισμός, Οντολογίες, Δείκτης ποιότητας αέρα (AQI), Έξυπνες πόλεις, MQTT, Neo4j, GraphQL.

Abstract

This master's thesis focuses on enhancing urban air quality monitoring through the utilization of data obtained from IoT devices and the application of semantic enrichment techniques. The European Air Quality Index (EAQI) is used as a reference throughout the study to determine the severity of air pollution in different places and to identify the causes of it.

To achieve this purpose, heterogeneous data is collected from IoT devices and open environmental data platforms. IoT devices are strategically placed in the urban environment to collect data on various air quality metrics, such as the concentration of harmful air pollutants (SO₂, NO₂, O₃) and levels of particulate matter (PM_{2.5}, PM₁₀). First, data is sent to IoT platforms and then transmitted via the MQTT protocol to a central server. At this point custom services collect geospatial environmental data from open platforms. The acquired data is processed and semantically annotated by assigning ontology classes and properties.

The use of semantic enrichment techniques enables a more complete understanding of the data as it adds context and meaning to the raw data collected from different sources. Additionally, by storing the semantic data in a Neo4j graph database, it facilitates the recognition of patterns and relationships within the data, making it easier to identify areas of concern and potential causes of air pollution.

The task also includes the implementation of a web application to visualize the stations' environmental data by submitting GraphQL queries to the database. This allows easy detection of areas with high levels of pollution. Additionally, the user can search for a route to follow based on the latest environmental air quality data that is free of air pollution, by setting the start and end points of the route on the map.

Keywords: *Internet of Things (IoT), Semantic enrichment, Ontologies, Air quality index, Smart cities, MQTT, Neo4j, GraphQL.*

1

Εισαγωγή

1.1 Εφαρμογές έξυπνων πόλεων

Η τεχνολογική ενδυνάμωση των πόλεων ως τρόπος αντιμετώπισης των προκλήσεων για τη βιωσιμότητά τους και την ποιότητα ζωής του πληθυσμού τους, έχει οδηγήσει στην υλοποίηση της έννοιας των έξυπνων πόλεων. Σε μια έξυπνη πόλη, οι τεχνολογίες πληροφοριών και επικοινωνιών χρησιμοποιούνται ως βασικά στοιχεία για τη συλλογή και ανάλυση μεγάλων ποσοτήτων πληροφοριών που παράγονται από διάφορες πηγές, όπως δίκτυα αισθητήρων, συστήματα κυκλοφορίας και συσκευές πολιτών. Τέτοια δεδομένα μπορούν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών ικανών να επηρεάσουν την καθημερινή ζωή των ανθρώπων βελτιώνοντας τις υπηρεσίες της πόλης και συμβάλλοντας στην οικονομική ανάπτυξη, την περιβαλλοντική βιωσιμότητα και την ποιότητα ζωής των ατόμων και της κοινωνίας. Ταυτόχρονα, είναι δυνατό να προωθηθούν διαδικασίες λήψης αποφάσεων που βασίζονται σε αξιόπιστες πληροφορίες παρέχοντας παράλληλα καινοτόμες και καλύτερες υπηρεσίες, εξοικονομώντας χρόνο και με σύνεση χρησιμοποιώντας φυσικούς, οικονομικούς, υλικούς και ανθρώπινους πόρους.

Η ανάπτυξη εφαρμογών έξυπνων πόλεων συνήθως αντιμετωπίζει προκλήσεις, όπως η ικανοποίηση πολλών πολύπλοκων απαιτήσεων, η ενσωμάτωση ετερογενών πηγών δεδομένων και η εξέταση γεωγραφικών πληροφοριών που αντιπροσωπεύουν τον πραγματικό αστικό χώρο. Οι πλατφόρμες έξυπνων πόλεων διαδραματίζουν βασικό ρόλο στην υλοποίηση αυτών των εφαρμογών καθώς παρέχουν υπηρεσίες υψηλού επιπέδου που μπορούν να επαναχρησιμοποιηθούν από προγραμματιστές. Παρά την ύπαρξη πολλών πλατφορμών για την υποστήριξη της ανάπτυξης εφαρμογών έξυπνων πόλεων, οι περισσότερες από αυτές δεν συσχετίζουν τις προσφερόμενες υπηρεσίες με γεωγραφικές πληροφορίες, δεν υποστηρίζουν την εκτέλεση σημασιολογικών

ερωτημάτων στα διαθέσιμα δεδομένα και έχουν περιορισμούς που μπορεί να επιβαρύνουν τις εργασίες ανάπτυξης.

1.2 Αντικείμενο διπλωματικής

Το αντικείμενο της παρούσας εργασίας είναι η καταγραφή υπηρεσιών και μεθόδων για τη συλλογή μεγάλου όγκου γεωχωρικών και περιβαλλοντικών δεδομένων χρησιμοποιώντας αισθητήρες ενσωματωμένους σε ετερογενή συστήματα IoT. Τα δεδομένα αυτά έχουν πολυτροπική φύση και αφορούν αντικείμενα, δραστηριότητες ή φαινόμενα, καταγράφοντας ταυτόχρονα τη θέση στην οποία αυτά εντοπίζονται. Η εξασφάλιση της διαλειτουργικότητας και ο χειρισμός ετερογενών δεδομένων μεγάλης κλίμακας αποτελεί βασικό παράγοντα για την ανάπτυξη εφαρμογών στις έξυπνες πόλεις αλλά και σε άλλους τομείς όπως το έξυπνο σπίτι, η έξυπνη υγεία, η γεωργία ακριβείας κ.α., συμβάλλοντας στην οικονομική εξέλιξη των παραγωγικών φορέων.

Τα γεωχωρικά δεδομένα συλλέγονται από ετερογενείς κόμβους οι οποίοι έχουν ενσωματωμένους μικροελεγκτές διαφορετικής τεχνολογίας και κατασκευαστή και χρησιμοποιούν διαφορετικές τεχνολογίες ασύρματης/ενσύρματης επικοινωνίας και δικτύωσης. Τα δεδομένα αυτά εμπλουτίζονται και από τρίτες πηγές μέσω διασυνδέσεων προγραμματισμού εφαρμογών (APIs) με σκοπό τη μελέτη των μετρούμενων μεγεθών στο σύνολο της γεωγραφικής έκτασης της περιοχής ενδιαφέροντος, σύμφωνα με το σενάριο λειτουργίας. Η εργασία αυτή στοχεύει στην υλοποίηση της κατάλληλης υποδομής ενδιάμεσου λογισμικού για τη συλλογή και διαχείριση γεωχωρικών δεδομένων και εφαρμόζεται για την ανεύρεση της καταλληλότερης διαδρομής στο χάρτη, με τους λιγότερους ατμοσφαιρικούς ρύπους, που μπορεί να διανύσει ένας χρήστης.

1.2.1 Κίνητρο για τη διεξαγωγή της εργασίας

Η ραγδαία ανάπτυξη της βιομηχανίας και των μεταφορών, όπως έχει παρατηρηθεί στις αναπτυσσόμενες χώρες, έχει οδηγήσει στην αύξηση της ατμοσφαιρικής ρύπανσης με σοβαρές επιπτώσεις για τις εκάστοτε κυβερνήσεις και τους πολίτες. Έχει παρατηρηθεί ότι η μακροχρόνια έκθεση των ανθρώπων σε αιωρούμενα σωματίδια εσωτερικού - εξωτερικού χώρου, οδηγεί στην επιβάρυνση της υγείας των ανθρώπων, επιφέροντας ακόμη και το θάνατο.

Οι ανθρώπινες δραστηριότητες που αποτελούν σημαντικές πηγές ατμοσφαιρικής ρύπανσης είναι:

- Η χρήση υδρογονανθράκων από οχήματα με κινητήρες εσωτερικής καύσης (π.χ., λεωφορεία και βαρέα οχήματα)
- Η παραγωγή θερμότητας και ηλεκτρικής ενέργειας (π.χ. βιομηχανικές μονάδες παραγωγής ενέργειας, λέβητες πετρελαίου και άνθρακα).
- Οι βιομηχανικές εγκαταστάσεις (π.χ. εργοστάσια κατασκευής, ορυχεία, διυλιστήρια πετρελαίου).
- Η αποτέφρωση/ καύση αποβλήτων.

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας η ατμοσφαιρική ρύπανση αποτελεί την πέμπτη κατά σειρά αιτία με τις περισσότερες επιπτώσεις στην υγεία [1]. Η ατμοσφαιρική ρύπανση σε συνδυασμό με τα χρόνια προβλήματα υγείας που αντιμετωπίζουν οι ευπαθείς ομάδες, μπορεί να επιδεινώσει δραματικά την κατάσταση της υγείας τους. Το 2016 καταγράφηκαν παγκοσμίως περισσότεροι από 4.2 εκατομμύρια πρόωφοι θάνατοι ανθρώπων, που ζούσαν σε αστικές και

αγροτικές περιοχές. Αυτή η θνησιμότητα οφείλεται στην έκθεση σε μικρά σωματίδια διαμέτρου ίσης ή μικρότερης των 2,5 μικρών (PM_{2.5}), τα οποία προκαλούν καρκίνους, καρδιαγγειακές και αναπνευστικές παθήσεις [2].

Η υποβάθμιση της ποιότητας του αέρα, σύμφωνα με τον ΠΟΥ οφείλεται σε ρύπους που έχουν χαρακτηριστεί παθογόνοι και για τους οποίους έχουν τεθεί όρια επιτρεπόμενων συγκεντρώσεων [3], [4]. Οι ρύποι με τις ισχυρότερες ενδείξεις για τις επιπτώσεις τους στην υγεία είναι τα σωματίδια PM, το όζον O₃, το διοξείδιο του αζώτου NO₂ και το διοξείδιο του θείου SO₂ (Πίνακας 1).

Πίνακας 1: Ατμοσφαιρικοί ρύποι και επιπτώσεις στην υγεία.

A/A	Περιγραφή	Ρύποι	Επιπτώσεις στην υγεία
1	Αιωρούμενα σωματίδια	PM	Αναπνευστικές παθήσεις, καρδιαγγειακές παθήσεις, καρκίνος πνευμόνων
2	Όζον	O ₃	Αναπνευστικά προβλήματα, άσθμα, πνευμονοπάθειες, μείωση λειτουργίας των πνευμόνων
3	Διοξείδιο του αζώτου	NO ₂	Μείωση λειτουργίας των πνευμόνων, βρογχίτιδα
4	Διοξείδιο του θείου	SO ₂	Ερεθισμό ματιών, βήχα, άσθμα, χρόνια βρογχίτιδα

1.2.2 Στόχοι και οφέλη εφαρμογής

Η συνεχής παρακολούθηση της ποιότητας του αέρα σε πραγματικό χρόνο στα μεγάλα αστικά κέντρα, είναι επιβεβλημένη για την πρόβλεψη κρίσιμων καταστάσεων και την διαφύλαξη της υγείας, της ασφάλειας και της άνεσης των πολιτών. Οι περισσότερες χώρες έχουν υλοποιήσει τις δικές τους υποδομές έχοντας υιοθετήσει πολιτικές και πρότυπα παρακολούθησης της ατμοσφαιρικής ρύπανσης, για την λήψη μέτρων που αποσκοπούν στην προστασία του περιβάλλοντος και την έγκαιρη προειδοποίηση των πολιτών.

Με βάση την τεχνολογική απελευθέρωση που προσφέρει το Διαδίκτυο των Πραγμάτων, έχουν υλοποιηθεί συστήματα παρακολούθησης της ποιότητας του αέρα στο πλαίσιο της «έξυπνης πόλης». Με την τεχνολογική εξέλιξη των δικτύων αισθητήρων και τη διαρκή σμίκρυνση των ηλεκτρονικών συσκευών, δίνεται η δυνατότητα δημιουργίας συστημάτων παρακολούθησης της ποιότητας του αέρα, χαμηλού κόστους, που βρίσκονται εγκατεστημένα στους καθημερινούς χώρους διαβίωσης.

Σκοπός της παρούσας εφαρμογής είναι η υλοποίηση ενός συστήματος παρακολούθησης της ποιότητας του αέρα, κάνοντας χρήση δεδομένων που προέρχονται από ετερογενείς πηγές όπως

αισθητήρες περιβάλλοντος και πλατφόρμες ανοικτών δεδομένων. Απώτερος στόχος είναι η υποστήριξη της κινητικότητας των πολιτών σε γεωγραφικές περιοχές σε συνάρτηση με τα επίπεδα ατμοσφαιρικής ρύπανσης που επικρατούν σε αυτές, αποβλέποντας στη βελτίωση της ποιότητας ζωής τους και την προώθηση φιλικών προς το περιβάλλον συμπεριφορών από τους πολίτες. Επιπρόσθετα, η εφαρμογή υιοθετεί έναν συμβουλευτικό και παροτρυντικό χαρακτήρα σχετικά με τις ημερήσιες δραστηριότητες των χρηστών όπως περίπατος, τρέξιμο, ποδηλασία, ενημερώνοντάς τους άμεσα για τα επίπεδα επικινδυνότητας των συγκεντρώσεων των επιβλαβών ρύπων.

Τα οφέλη από την υιοθέτηση τέτοιου είδους εφαρμογών είναι αρκετά, επιβεβαιώνοντας τον ανθρωποκεντρικό χαρακτήρα της τεχνολογίας. Ενδεικτικά:

- Τα δεδομένα που συλλέγονται από την παρακολούθηση της ποιότητας του αέρα, συμμετέχουν σε μεγάλο ποσοστό στην εκτίμηση των επιπτώσεων που προκαλεί η ατμοσφαιρική ρύπανση στη δημόσια υγεία.
- Τα δεδομένα ποιότητας αέρα συμβάλλουν στον έλεγχο εφαρμογής των προτύπων ποιότητας αέρα που έχει θέσει ο Παγκόσμιος Οργανισμός Υγείας, σε μια γεωγραφική περιοχή.
- Η εφαρμογή αυτή συνδράμει στην αξιολόγηση των προγραμμάτων ελέγχου της ατμοσφαιρικής ρύπανσης που εφαρμόζονται σε μια περιοχή, για την εξαγωγή έγκυρων συμπερασμάτων σχετικά με την αποτελεσματικότητά τους.
- Με βάση τα δεδομένα που συλλέγονται, μπορούν να ληφθούν νέα μέτρα ελέγχου για την προστασία του περιβάλλοντος και της υγείας όλων των ζωντανών οργανισμών.
- Τα δεδομένα ποιότητας του αέρα βοηθούν στην εξαγωγή συμπερασμάτων για το ποσοστό θνησιμότητας οποιασδήποτε γεωγραφικής τοποθεσίας λόγω της ατμοσφαιρικής ρύπανσης. Συμβάλει στην αξιολόγηση και την σύγκριση βραχυπρόθεσμων και μακροπρόθεσμων ασθνεσιών / διαταραχών της υγείας ως απόρροια της ατμοσφαιρικής ρύπανσης.

1.2.3 Ευρωπαϊκός δείκτης ποιότητας αέρα (AQI)

Ο δείκτης ποιότητας αέρα αποτελεί ένα περιγραφικό σύστημα ειδοποίησης του πληθυσμού, σχετικά με το επίπεδο της ποιότητας του ατμοσφαιρικού αέρα και τους κινδύνους υγείας που πιθανόν να ελλοχεύονται. Οι ενημερώσεις απευθύνονται στο ευρύ κοινό και στις ευπαθείς ομάδες όπως είναι τα παιδιά, οι ηλικιωμένοι και τα άτομα με καρδιαγγειακά και αναπνευστικά νοσήματα. Ο δείκτης AQI χρησιμοποιείται για τη λήψη αποφάσεων σχετικά με την εκτέλεση δραστηριοτήτων σε εξωτερικούς χώρους. Για παράδειγμα λαμβάνοντας υπόψιν το δείκτη, θα μπορούσε να επιτραπεί ή όχι η διεξαγωγή αθλητικών εκδηλώσεων μια συγκεκριμένη ημέρα.

Ο Ευρωπαϊκός Οργανισμός Περιβάλλοντος χρησιμοποιεί μετρήσεις έως και πέντε διαφορετικών ρύπων για την εξαγωγή συμπερασμάτων, ενώ οι δείκτες ποιότητας αέρα στηρίζονται στις συγκεντρώσεις των ρυπογόνων αερίων ή σωματιδίων με μονάδες $\mu\text{g}/\text{m}^3$ [5]. Ο συνολικός δείκτης αέριας ρύπανσης μιας περιοχής, υπολογίζεται από την υψηλότερη συγκέντρωση των ρύπων: αιωρούμενα σωματίδια (PM10), λεπτά αιωρούμενα σωματίδια (PM2.5), όζον (O3), διοξείδιο του Αζώτου (NO2) και διοξείδιο του Θείου (SO2).

Ο δείκτης αντικατοπτρίζει τον πιθανό αντίκτυπο της ποιότητας του αέρα στην υγεία των ανθρώπων, σύμφωνα με τα επίπεδα συγκεντρώσεων των ρύπων, όταν αυτά υπερβαίνουν κάποια προκαθορισμένα όρια. Η νομοθεσία της Ευρωπαϊκής Ένωσης καθορίζει πρότυπα ποιότητας αέρα,

για τα επιτρεπόμενα όρια, τόσο σε βραχυπρόθεσμα (ωριαία ή ημερήσια) όσο και σε μακροπρόθεσμα (ετήσια) χρονικά διαστήματα. Τα πρότυπα μακροχρόνιων επιπέδων είναι αυστηρότερα έναντι των βραχυπρόθεσμων, καθώς μπορεί να προκύψουν σοβαρές επιπτώσεις στην υγεία από την μακροχρόνια έκθεση σε ρύπους. Ο δείκτης δείχνει την βραχυπρόθεσμη κατάσταση της ποιότητας του αέρα η οποία διαφέρει σημαντικά από την μακροπρόθεσμη.

Οι τιμές συγκέντρωσης έως και πέντε βασικών ρύπων, καθορίζουν το επίπεδο του δείκτη που αντικατοπτρίζει την ποιότητα αέρα στον αντίστοιχο σταθμό παρακολούθησης. Ο δείκτης αντιστοιχεί στο φτωχότερο ποιοτικά επίπεδο, για οποιονδήποτε από τους πέντε ρύπους (Σχήμα 1). Ο υπολογισμός του δείκτη, για σταθμούς κυκλοφορίας οχημάτων, χρησιμοποιεί μόνο δεδομένα για τους ρύπους NO₂ και PM (PM_{2.5} ή PM₁₀ ή και τα δύο). Αυτό συμβαίνει επειδή οι συγκεντρώσεις SO₂ μπορεί να είναι υψηλές σε τοπικές περιοχές και να αλλοιώσουν την εικόνα της ποιότητας του αέρα. Σε αντίθεση με το διοξείδιο του θείου, τα επίπεδα του όζοντος είναι συνήθως πολύ χαμηλά στους σταθμούς κυκλοφορίας. Σε βιομηχανικούς σταθμούς, ο δείκτης αέρας ρύπανσης υπολογίζεται με δεδομένα τουλάχιστον τριών ρύπων NO₂, O₃ και PM (PM_{2.5} ή PM₁₀ ή και τα δύο). Οι μετρήσεις ποιότητας αέρα που υπερβαίνουν τις μέγιστες τιμές στην κατηγορία “Extremely poor”, δεν λαμβάνονται υπόψη για τον υπολογισμό του δείκτη, δεδομένου ότι αυτές οι τιμές συνήθως είναι λανθασμένες. Οι ζώνες του πίνακα δεικτών συμπληρώνονται από σχετικά μηνύματα που σχετίζονται με την υγεία των ανθρώπων και παρέχουν συστάσεις τόσο για τον γενικό όσο και για τον ευαίσθητο πληθυσμό (Σχήμα 2).

Pollutant	Index level (based on pollutant concentrations in µg/m ³)					
	Good	Fair	Moderate	Poor	Very poor	Extremely poor
Particles less than 2.5 µm (PM _{2.5})	0-10	10-20	20-25	25-50	50-75	75-800
Particles less than 10 µm (PM ₁₀)	0-20	20-40	40-50	50-100	100-150	150-1200
Nitrogen dioxide (NO ₂)	0-40	40-90	90-120	120-230	230-340	340-1000
Ozone (O ₃)	0-50	50-100	100-130	130-240	240-380	380-800
Sulphur dioxide (SO ₂)	0-100	100-200	200-350	350-500	500-750	750-1250

Σχήμα 1: Ευρωπαϊκός δείκτης ποιότητας αέρα.

AQ index	General population	Sensitive populations
Good	The air quality is good. Enjoy your usual outdoor activities.	The air quality is good. Enjoy your usual outdoor activities.
Fair	Enjoy your usual outdoor activities	Enjoy your usual outdoor activities
Moderate	Enjoy your usual outdoor activities	Consider reducing intense outdoor activities, if you experience symptoms.
Poor	Consider reducing intense activities outdoors, if you experience symptoms such as sore eyes, a cough or sore throat	Consider reducing physical activities, particularly outdoors, especially if you experience symptoms.
Very poor	Consider reducing intense activities outdoors, if you experience symptoms such as sore eyes, a cough or sore throat	Reduce physical activities, particularly outdoors, especially if you experience symptoms.
Extremely poor	Reduce physical activities outdoors.	Avoid physical activities outdoors.

Σχήμα 2: Ζώνες δεικτών και μηνύματα ενημέρωσης πληθυσμού.

1.3 Δομή της διπλωματικής

Η παρούσα εργασία απαρτίζεται από δέκα κεφάλαια ακολουθούμενα από τη βιβλιογραφία. Το παρόν κεφάλαιο αποτελεί την εισαγωγή της εργασίας, στην οποία αναφέρονται το αντικείμενο της μελέτης, η συνεισφορά και η δομή της. Επιπρόσθετα, παρουσιάζεται το κίνητρο για την υιοθέτηση της περίπτωσης εφαρμογής με σκοπό την ανεύρεση κατάλληλης διαδρομής στο χάρτη, που μπορεί να διανύσει ένας χρήστης, αξιοποιώντας δεδομένα που σχετίζονται με ατμοσφαιρικούς ρύπους.

Στο δεύτερο κεφάλαιο πραγματοποιείται βιβλιογραφική επισκόπηση συναφών μελετών, που εστιάζουν στη συλλογή και στη σημασιολογική επισημείωση περιβαλλοντικών δεδομένων.

Στο τρίτο κεφάλαιο αναλύεται η γενική αρχιτεκτονική της μεθοδολογίας συλλογής ετερογενών δεδομένων, από το στάδιο της συλλογής τους μέχρι και το τελικό επίπεδο εφαρμογής. Επιπρόσθετα, περιγράφεται η αρχιτεκτονική σε επίπεδο υλοποίησης, παρέχοντας αναλυτικότερες πληροφορίες για τις τεχνολογίες των IoT συσκευών, τη δικτύωσή τους και τα πρωτόκολλα για την αποστολή των δεδομένων στις IoT πλατφόρμες.

Στο τέταρτο κεφάλαιο γίνεται αναφορά στην κατασκευαστική ετερογένεια των πιο κοινών προγραμματιζόμενων IoT συσκευών της αγοράς, που χρησιμοποιήθηκαν στην περίπτωση εφαρμογής.

Στο πέμπτο κεφάλαιο παρουσιάζονται οι τεχνολογίες δικτύωσης και τα δικτυακά πρωτόκολλα που χρησιμοποιήθηκαν κατά την φάση της υλοποίησης, καταγράφοντας τα κύρια χαρακτηριστικά τους.

Στο έκτο κεφάλαιο παρουσιάζονται οι IoT πλατφόρμες Thingspeak και Ubidots, μελετώντας τα προσφερόμενα χαρακτηριστικά τους στην περίπτωση των δωρεάν αδειών χρήσης.

Το έβδομο κεφάλαιο περιγράφει τις έννοιες του σημασιολογικού ιστού και των οντολογιών, καθώς και τη συμβολή της σημασιολογίας στο Διαδίκτυο των Πραγμάτων.

Το όγδοο κεφάλαιο παρουσιάζει τους αισθητήρες αερίων, αιωρούμενων σωματιδίων και προσδιορισμού γεωγραφικής θέσης που χρησιμοποιήθηκαν στην περίπτωση εφαρμογής.

Στο ένατο κεφάλαιο περιγράφεται αναλυτικά η υλοποίηση όλων των επιπέδων της αρχιτεκτονικής που παρουσιάστηκε στο τρίτο κεφάλαιο, στα πλαίσια της μελέτης εφαρμογής για την αναζήτηση της καταλληλότερης διαδρομής στο χάρτη με το ελάχιστο δυνατό φορτίο ατμοσφαιρικών ρύπων.

Η εργασία ολοκληρώνεται με τα συμπεράσματα που προκύπτουν από την εφαρμογή της μεθοδολογίας.

2

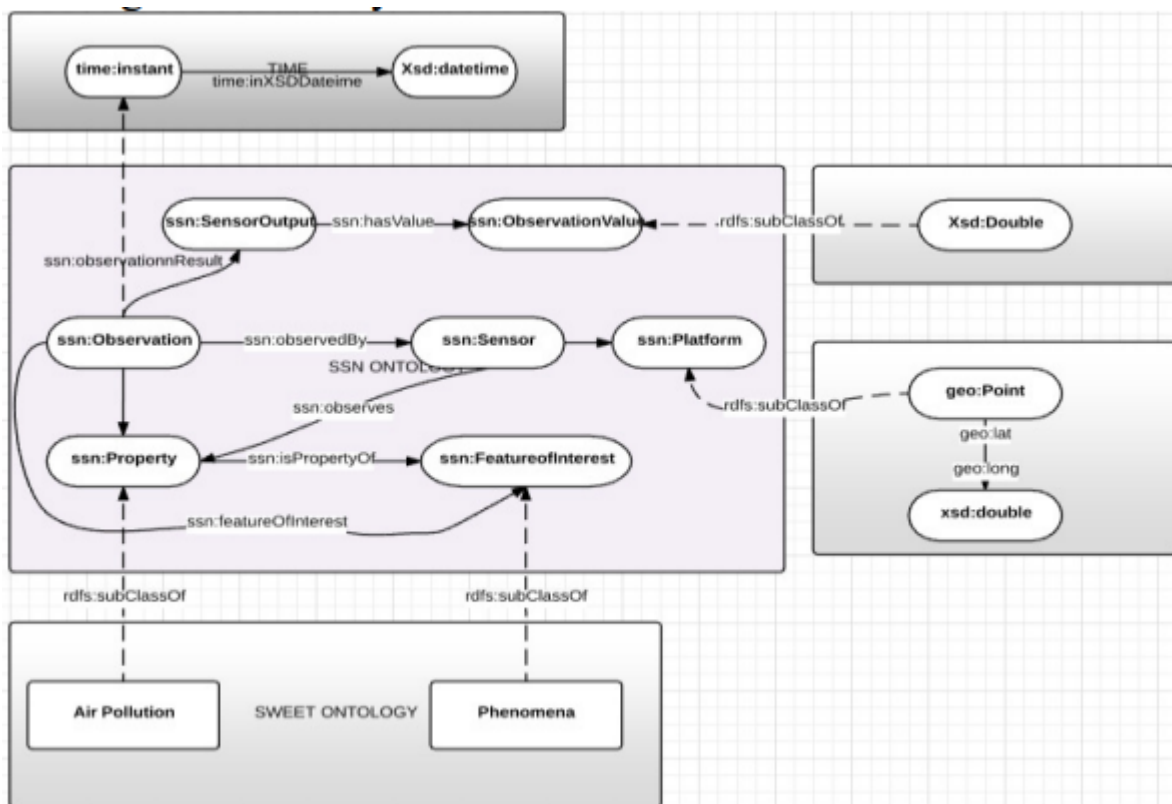
Βιβλιογραφική Επισκόπηση

Η ετερογένεια των δεδομένων αποτελεί θέμα καθοριστικής σημασίας για τις εφαρμογές του Διαδικτύου των Πραγμάτων. Έχουν καταγραφεί αρκετές προσπάθειες ανάπτυξης δικτύου αισθητήρων για τη συλλογή και την αναπαράσταση δεδομένων ποιότητας αέρα. Η ετερογένεια των μοντέλων και της δομής που παρουσιάζουν τα δεδομένα όπως ταχύτητα ανέμου, ατμοσφαιρική ρύπανση, καιρικές συνθήκες, συγκέντρωση ατμοσφαιρικών ρύπων καθώς και η πολυπλοκότητα των συσκευών, δυσχεραίνουν την απρόσκοπτη ενοποίηση δεδομένων και υπηρεσιών. Στις εφαρμογές και στις υπηρεσίες υψηλότερου επιπέδου, τα δεδομένα πρέπει να αρχειοθετούνται και να επισημαίνονται σημασιολογικά χρησιμοποιώντας μια κοινή διεπαφή που αντιπροσωπεύει συσκευές και δεδομένα IoT, προσανατολισμένη προς στις υπηρεσίες.

Οι συγγραφείς της μελέτης [6] διαβλέποντας την πολυπλοκότητα και την αδυναμία των υφιστάμενων σημασιολογικών μοντέλων, μέχρι εκείνη τη στιγμή, να επεξεργαστούν δεδομένα μεγάλης κλίμακας, οδηγήθηκαν στον σχεδιασμό ενός μικρού και ευέλικτου μοντέλου για την αντιμετώπιση της ετερογένειας, τον σχολιασμό και την παρουσίαση δεδομένων αισθητήρων. Για την υλοποίησή του χρησιμοποίησαν την οντολογία SSN [7], η οποία παρέχει μια βασική σημασιολογική περιγραφή των αισθητήρων δικτύωσης την οποία επέκτειναν με οντολογίες από άλλους τομείς, σχεδιάζοντας μια απλή λύση οντολογίας. Η πρόταση μπορεί να συνδυάσει τις μετρήσεις των αισθητήρων, με πληροφορίες της γεωγραφικής θέσης του σταθμού. Το προτεινόμενο μοντέλο εφαρμόστηκε σε περιβαλλοντικά δεδομένα για τη διαχείριση συμβάντων και την υποβολή ερωτημάτων (Σχήμα 3).

Ο κύριος στόχος της συγκεκριμένης υλοποίησης είναι η εξάλειψη της ετερογένειας των δεδομένων από τους αισθητήρες, χρησιμοποιώντας ένα κοινό μοντέλο που παρέχει ευκολία στην υποβολή ερωτημάτων και στο συλλογισμό που εφαρμόζεται στα δεδομένα. Αυτή η αρχιτεκτονική προτείνεται για να επιδείξει πώς τα δεδομένα μετατρέπονται από ακατέργαστη μορφή, σε πλούσια

σημασιολογική περιγραφή. Όλα τα μεταδεδομένα αισθητήρων και η μέτρησή τους μπορούν να δημοσιευτούν, να ευρετηριαστούν, σε αυτά να υποβληθούν ερωτήματα και να ανακαλυφθούν, να αιτιολογηθούν και να επαναχρησιμοποιηθούν για άλλες εφαρμογές και υπηρεσίες.

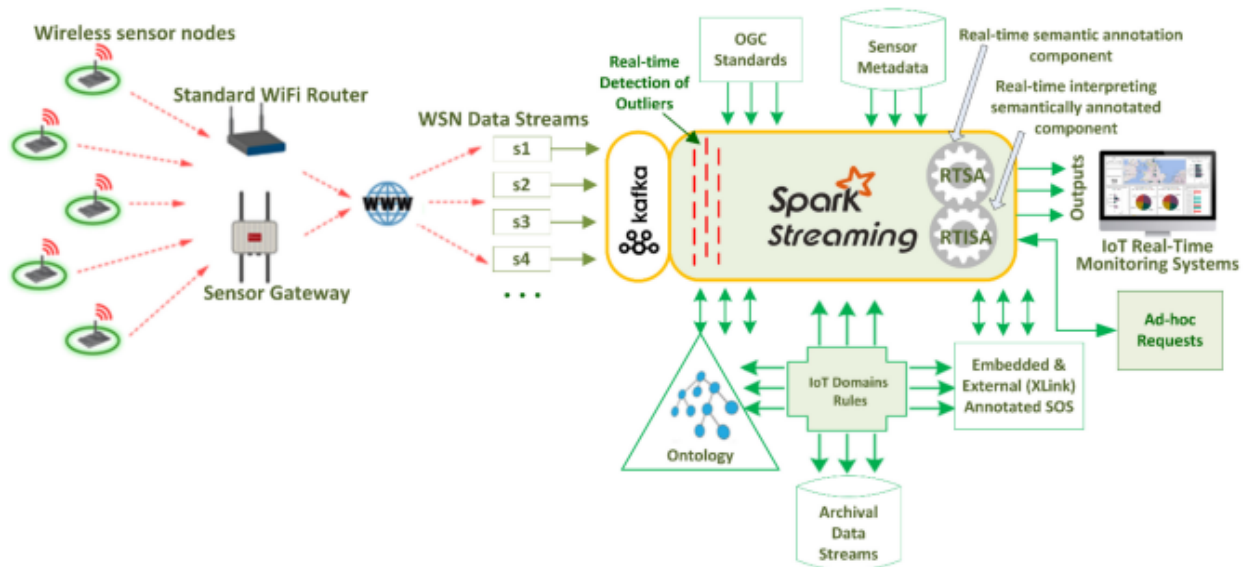


Σχήμα 3: Προτεινόμενη οντολογία μελέτης [6].

Οι συσκευές IoT επιτρέπουν την παρακολούθηση της ποιότητας του αέρα σε πραγματικό χρόνο, οι οποίες παράγουν ροές δεδομένων ανίχνευσης συνεχώς, μεταδίδοντας τα δεδομένα αυτά σε έναν κεντρικό διακομιστή. Τα ακατέργαστα δεδομένα ροής αισθητήρων παρέχουν ελάχιστη χρησιμότητα, εκτός εάν σχολιάζονται σωστά. Ως εκ τούτου, οι ερευνητές [8], [9] υλοποίησαν το ολοκληρωμένο σύστημα IoTSAS για την σημασιολογική επισημείωση ροών δεδομένων σε πραγματικό χρόνο. Παράλληλα, πρότειναν το μοντέλο Semantic Sensor Web (SSW), το οποίο είναι ένας συνδυασμός του Sensor Web και των τεχνολογιών του Semantic Web. Οι ερευνητικές εργασίες εστιάζουν στην ενσωμάτωση της σημασιολογίας σε πραγματικό χρόνο σε ετερογενή δεδομένα ροής αισθητήρων με πλαίσιο στο IoT. Σε αυτό το πλαίσιο, αναπτύχθηκαν διαφορετικοί σημασιολογικοί σχολιασμοί για δεδομένα ροής αισθητήρων με τη μορφή Υπηρεσίας Παρατήρησης Αισθητήρων (SOS), επεκτείνοντας το πρότυπο OGC και εφαρμόστηκαν σε ένα σύστημα παρακολούθησης της ποιότητας του αέρα σε πραγματικό χρόνο.

Η πολυπλοκότητα, η δυναμικότητα, η τυποποίηση, η γενίκευση και ο μεγάλος όγκος των μη δομημένων ροών δεδομένων από τους αισθητήρες είναι οι κύριες προκλήσεις για την επεξεργασία σε πραγματικό χρόνο. Η επιλογή τεχνολογιών και προτύπων για την ανάπτυξη τεχνικής ενσωμάτωσης σημασιολογίας σε πραγματικό χρόνο σε ετερογενή δεδομένα παρατήρησης αισθητήρων και μεταδεδομένα αισθητήρων με πλαίσιο στο IoT είναι εξαιρετικά σημαντική. Το

προτεινόμενο σύστημα σημασιολογικού σχολιασμού σε πραγματικό χρόνο (Σχήμα 4) χρησιμοποιεί το Apache Kafka για τη μετατροπή της διαφορετικής μορφής δεδομένων των αισθητήρων σε μία ενιαία, το Spark Streaming για τη σημασιολογική επισημείωση των δεδομένων και το Apache Cassandra database για την αποθήκευσή τους.

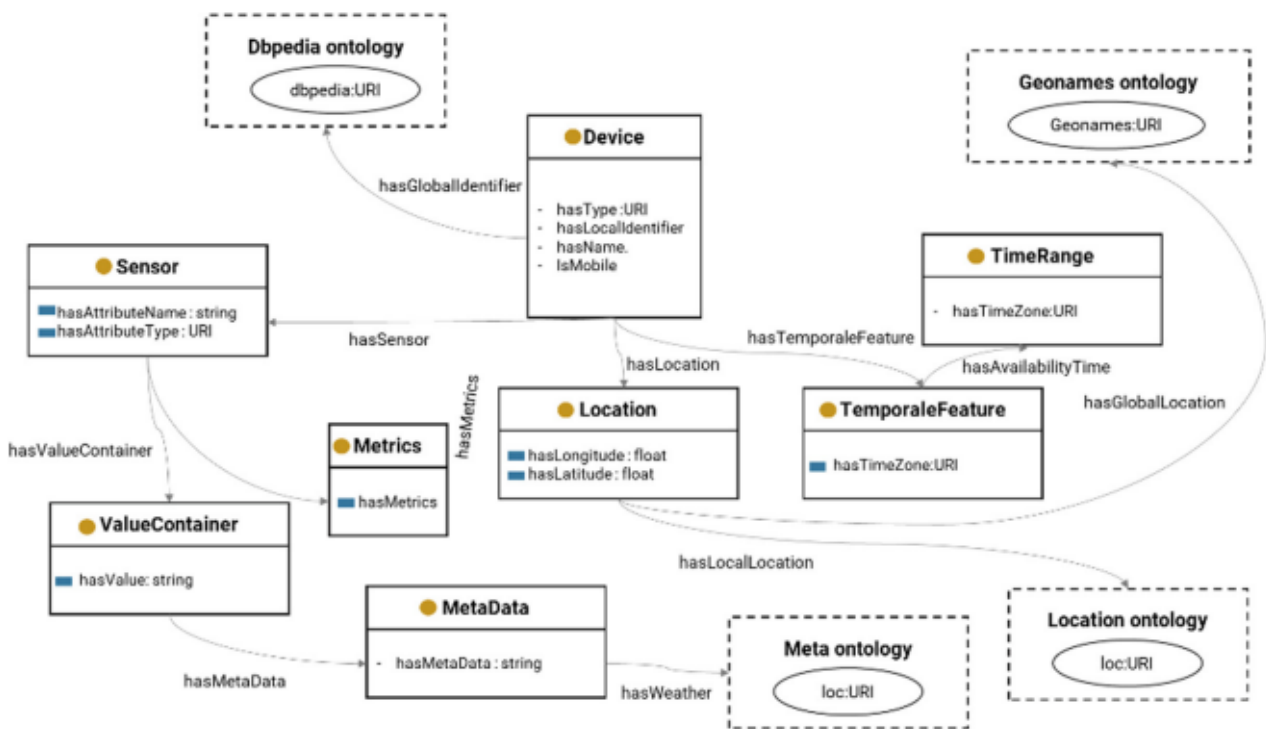


Σχήμα 4: Αρχιτεκτονική συστήματος [8] [9].

Επίσης, έχουν καταγραφεί μελέτες που επικεντρώνονται στη σημασιολογική επισημάνση δεδομένων έξυπνων πόλεων για την παρακολούθηση της ποιότητας αέρα και στην εκπαίδευση αλγόριθμων μηχανικής μάθησης με την γνώση που κρύβεται στα γραφοδεδομένα. Συγκεκριμένα, στην εργασία [10], παρουσιάστηκε μια προσέγγιση που επιτρέπει την κατανοητή ανακάλυψη γνώσης στο χώρο του IoT, χρησιμοποιώντας οντολογίες και μηχανική μάθηση. Προτάθηκε ένα σύνολο οντολογιών για την τυποποίηση της διαλειτουργικότητας και της ανταλλαγής πληροφοριών μεταξύ των συσκευών (Σχήμα 5). Η τελική οντολογία επιτρέπει την κατανόηση των δεδομένων που μεταδίδονται από τη συσκευή, συμπεριλαμβανομένων των πληροφοριών του δείκτη ποιότητας αέρα, καθώς και την παροχή άλλων πληροφοριών σχετικά με την τοποθεσία, τα χαρακτηριστικά, τις μετρήσεις πρόβλεψης. Οι έννοιες που σχετίζονται με άλλους σχετικούς τομείς, όπως αισθητήρες, παρατήρηση, μέτρηση και τοποθεσία, συμπεριλαμβάνονται από άλλες εξωτερικές οντολογίες. Για παράδειγμα, η οντολογία Air Quality Index παρέχει το νόημα της παρατήρησης που ανακτάται από τους αισθητήρες. Επίσης, η οντολογία Geonames παρέχει περισσότερες πληροφορίες σχετικά με τη θέση της συσκευής με βάση τις συντεταγμένες GPS.

Ο κύριος λόγος πίσω από τη χρήση οντολογιών και κατανοητής μηχανικής μάθησης είναι η επαναχρησιμοποίηση των παραμέτρων του μοντέλου μηχανικής εκμάθησης και η εξαγωγή περισσότερων πληροφοριών από τα δεδομένα που παρέχονται από μία μόνο συσκευή. Το κύριο μοντέλο πρόβλεψης του δείκτη ποιότητας αέρα που χρησιμοποιήθηκε σε αυτή τη μελέτη, είναι ο αλγόριθμος Random Forests. Τα χαρακτηριστικά που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου είναι οι συγκεντρώσεις των ρύπων NO₂, SO₂, O₃, PM₁₀ που παρέχει κάθε σταθμός. Τα

αποτελέσματα έδειξαν ότι ο αλγόριθμος Random Forests είχε καλύτερη απόδοση στο σύνολο δεδομένων 2 ετών σε σύγκριση με άλλα μοντέλα πρόβλεψης που εφαρμόστηκαν.

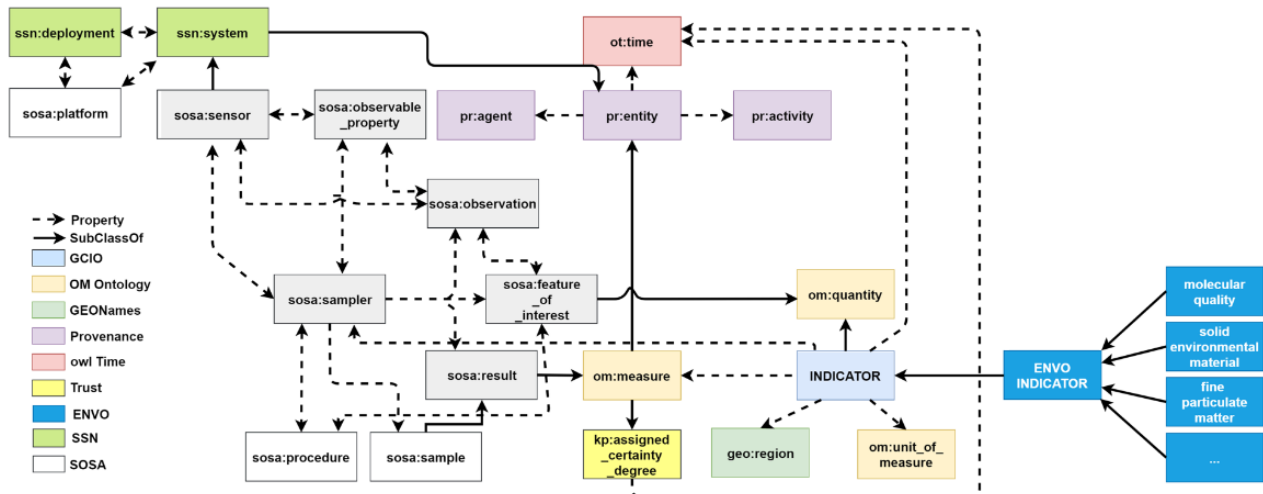


Σχήμα 5: Προτεινόμενη οντολογία μελέτης [10].

Η τεχνολογική ανάπτυξη και η διάδοση του εξοπλισμού IoT οδήγησε στην παραγωγή μεγάλων όγκων περιβαλλοντικών δεδομένων οι οποίοι, σε ορισμένες περιπτώσεις, είναι ελλιπείς, ακολουθούν διαφορετικές μορφές αναπαράστασης και έχουν διαφορετικές σημασιολογικές προσεγγίσεις. Με την ετερογένεια των διαφορετικών στοιχείων IoT γείρονται ζητήματα διαλειτουργικότητας που ενδέχεται να παρεμποδίσουν την αποτελεσματικότητα των συστημάτων λήψης αποφάσεων υποστήριξης για έξυπνες πόλεις, όπου έχει εξεταστεί η χρήση τεχνικών big data και μηχανικής μάθησης πέραν της εξερεύνησης δεδομένων έξυπνων πόλεων. Οι ερευνητές στην εργασία [11] προτείνουν τη χρήση μιας πλατφόρμας που βασίζεται σε περιβάλλον IoT για έξυπνες πόλεις παρέχοντας διαλειτουργικότητα από την συλλογή δεδομένων μέχρι την εξαγωγή και οπτικοποίηση γνώσης κάνοντας χρήση των τεχνολογιών του σημασιολογικού ιστού καθώς και του ορισμού μιας οντολογίας για δείκτες περιβάλλοντος. Η περίπτωση χρήσης παρακολούθησης ποιότητας αέρα εξετάστηκε για την αξιολόγηση της πλατφόρμας. Τα κίνητρα που οδήγησαν τους ερευνητές στο σχεδιασμό της πλατφόρμας είναι:

- Η έλλειψη κοινής μορφής δεδομένων και προτύπων κοινής χρήσης.
- Η ετερογένεια στις τεχνολογίες δικτύωσης και αισθητήρων.
- Η έλλειψη τυποποιημένου ορισμού περιβαλλοντικών δεικτών.
- Η σημασιολογική διαλειτουργικότητα μεταξύ λύσεων IoT για τον τομέα του περιβάλλοντος.

Οι συγγραφείς παρατήρησαν ότι η αποκλειστική χρήση οντολογιών όπως η ENVO, η GCIO ή η SSN δεν αρκεί για να καλύψει όλες τις απαιτήσεις για την σημασιολογική επισήμειωση των δεικτών περιβάλλοντος σε μια έξυπνη πόλη. Η προτεινόμενη οντολογία EISCO είναι το αποτέλεσμα συνδυασμού των παραπάνω οντολογιών και ορίστηκε για να καλύψει τους κύριους ορισμούς που απαιτούνται για την σημασιολογική αναπαράσταση δεδομένων που emπίπτουν στην παρακολούθηση του περιβάλλοντος σε μια έξυπνη πόλη.



Σχήμα 6: Προτεινόμενη οντολογία μελέτης [11].

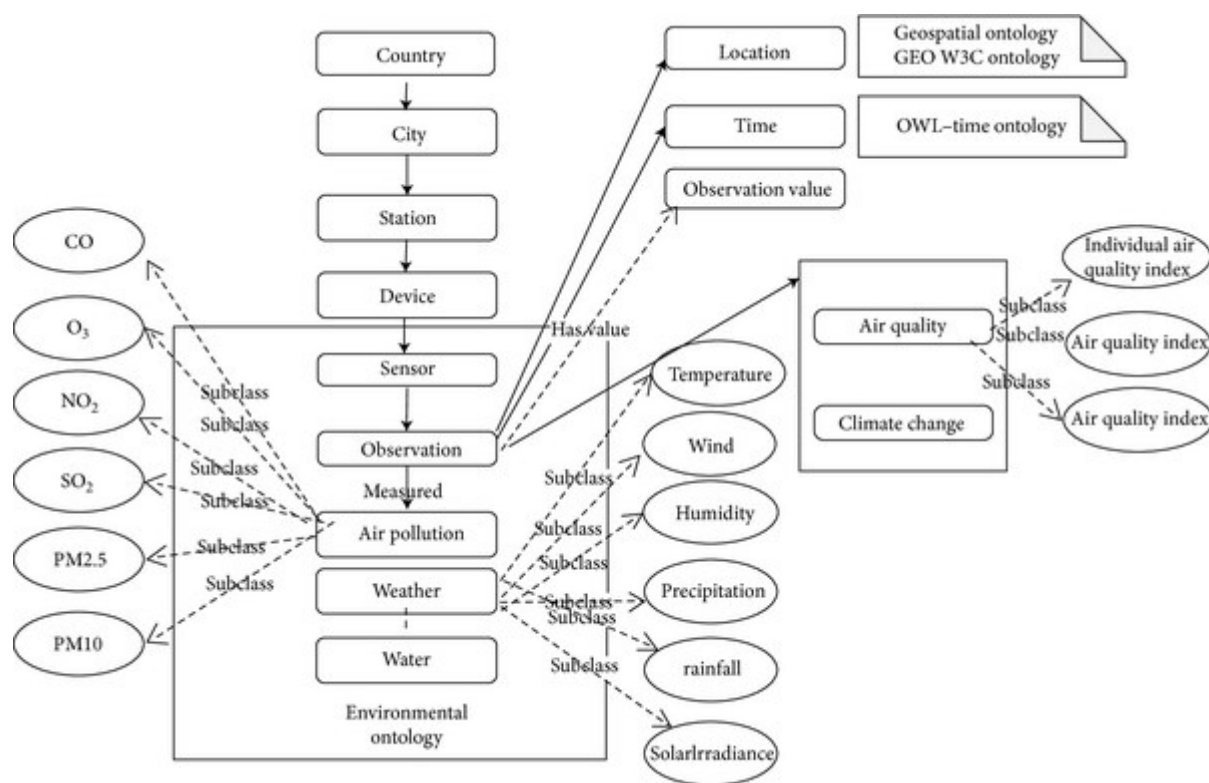
Άλλη μια ερευνητική εργασία [12] που επικεντρώνεται στα κύρια χαρακτηριστικά των μεγάλων δεδομένων (όγκος, ταχύτητα, ποικιλία) αναφέρει ότι η υποδομή των έξυπνων πόλεων δημιουργεί ετερογενή μεγάλα δεδομένα. Η κύρια πρόκληση για τους μελετητές είναι η ικανότητα συλλογής και προώθησης έγκαιρων δεδομένων της πόλης από έναν τεράστιο αριθμό ετερογενών πηγών, όπως αισθητήρες, διακομιστές, συσκευές, οχήματα, κτίρια και ανθρώπινες δραστηριότητες, καθώς και την αντιμετώπιση τόσο ιστορικών όσο και μεγάλων δεδομένων σε πραγματικό χρόνο. Οι περισσότερες δραστηριότητες που σχετίζονται με δεδομένα, απαιτούν τεχνολογίες όπως αποθήκευση, επεξεργασία, ανάλυση και κοινή χρήση, ενώ για την απόδοση νοήματος στα δεδομένα με σκοπό την ανίχνευση συμβάντων, ανάπτυξη συλλογιστικής και υποστήριξη αποφάσεων, απαιτούνται οι σημασιολογικές τεχνολογίες. Λαμβάνοντας υπόψη τα παραπάνω, η έρευνα στοχεύει στη δημιουργία και ανάπτυξη ενός γενικού πλαισίου για έξυπνες πόλεις (SSWF) που χρησιμοποιεί έναν συνδυασμό δραστηριοτήτων που σχετίζονται με δεδομένα και σημασιολογία.

Το προτεινόμενο πλαίσιο παρέχει μια καθολική βάση γνώσεων για δεδομένα που παράγονται από διαφορετικές πηγές δεδομένων παρέχοντας κατανόηση της σημασίας τους, του πλαισίου και των σχέσεων μεταξύ δεδομένων και γεγονότων. Το SSWF έχει προσθέσει εκατομμύρια εγγραφές σε τριπλέτες RDF χρησιμοποιώντας δεδομένα παρακολούθησης ποιότητας αέρα και καιρού από 40 ευρωπαϊκές χώρες. Τα κύρια χαρακτηριστικά αυτού του πλαισίου είναι:

- Η κατασκευή ενός καθολικού μοντέλου σημασιολογικών δεδομένων,
- Ο ορισμός ένα συνόλου κανόνων συσχέτισης για πρόβλεψη, προτάσεις και αφαιρέσεις.
- Η χρήση σημασιολογικών προτύπων RDF για να γίνουν τα δεδομένα «αυτοπεριγραφικά».

- Η διαχείριση μεγάλων δεδομένων.

Οι μελετητές δημιούργησαν μια κοινή βάση γνώσης (SCEO) που μπορεί να χειριστεί στατικά, ημι-στατικά και δεδομένα σε πραγματικό χρόνο. Η οντολογία SCEO είναι μια καθολική οντολογία, η οποία συγχωνεύει περισσότερες από μία οντολογίες, επεκτείνοντας τη βάση γνώσεων. Αυτή η συγχώνευση είναι απαραίτητη για τη μεταφορά γνώσης μεταξύ διαφορετικών βάσεων γνώσης. Το κοινό χαρακτηριστικό μεταξύ των οντολογιών είναι τα στοιχεία τοποθεσίας και χρόνου. Τα στοιχεία αυτά χρησιμοποιούν την τυπική οντολογία GEO W3C. Το Σχήμα 7 δείχνει μια γενική εικόνα της οντολογίας της περιβαλλοντικής δομής μιας έξυπνης πόλης. Η οντολογία περιέχει ένα σύνολο κατηγοριών όπως (County, City, Station, Device, Sensor, Air pollution, Weather, Water, Quality Air, Climate Change). Περιέχει επίσης ένα σύνολο υποκατηγοριών όπως η ατμοσφαιρική ρύπανση (CO, O₃, NO₂, SO₂, PM_{2.5}, PM₁₀) και ο καιρός (θερμοκρασία, άνεμος, υγρασία, βροχοπτώσεις κ.λπ.).

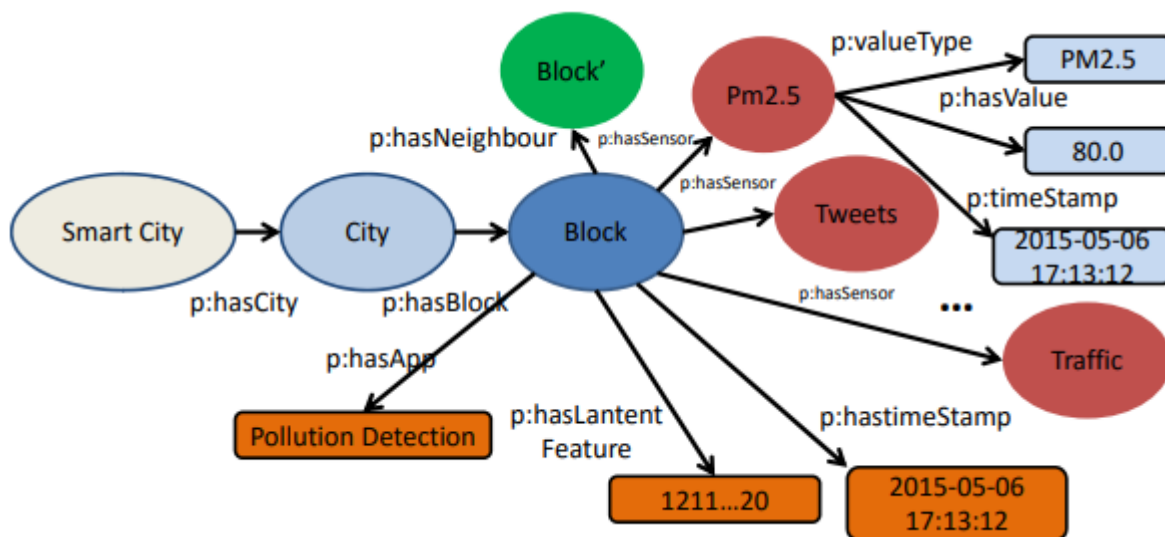


Σχήμα 7: Οντολογία περιβαλλοντικής δομής έξυπνης πόλης [12].

Οι συγγραφείς της μελέτης [13] αναφέρουν ότι η ποικιλία, ο όγκος, η ετερογένεια και η φύση σε πραγματικό χρόνο των δεδομένων που λαμβάνονται από τις έξυπνες πόλεις, θέτουν σημαντικές προκλήσεις. Συχνά συναντώνται διαφορετικοί τύποι δεδομένων με διαφορετικές μορφές αναπαράστασης, κατανομής, κλίμακες και πυκνότητες, με αποτέλεσμα την ύπαρξη μεμονωμένης πληροφορίας. Για την αντιμετώπιση των παραπάνω οι μελετητές πρότειναν ένα σημασιολογικό πλαίσιο που ενσωματώνει το IoT με τη μηχανική μάθηση και εφαρμόζεται στις έξυπνες πόλεις. Το προτεινόμενο πλαίσιο ανακτά και μοντελοποιεί αστικά δεδομένα για ορισμένα είδη εφαρμογών IoT που βασίζεται σε τεχνολογίες σημασιολογίας και μηχανικής μάθησης. Επιπλέον, προτείνονται δύο

περιπτωσιολογικές μελέτες: ανίχνευση ρύπανσης από κινούμενα οχήματα και ανίχνευση προτύπων κυκλοφορίας. Τα πειραματικά αποτελέσματα έδειξαν ότι το προτεινόμενο σύστημά είναι επεκτάσιμο και ικανό να φιλοξενήσει μεγάλο αριθμό αστικών περιοχών με διαφορετικούς τύπους εφαρμογών IoT.

Το προτεινόμενο σύστημα χρησιμοποιεί σημασιολογικές τεχνολογίες για τη μοντελοποίηση και τον σχολιασμό όλων των ειδών δεδομένων για τη βελτίωση του νοήματός τους και την απόκρυψη της πολυπλοκότητας των πηγών δεδομένων, παρέχοντας μια τυπική μορφή αναπαράστασης. Επιπλέον, υιοθετήθηκε ένα επίπεδο για την ενσωμάτωση μεθόδων μηχανικής μάθησης, όπως είναι η μεταφορά μάθησης για την επίλυση της αραιότητας δεδομένων. Το έργο της μεταφοράς μάθησης περιλαμβάνει τη μεταφορά γνώσης από περιοχές με πλούσια δεδομένα σε περιοχές με αραιά στοιχεία, διαπιστώνοντας ότι τα δεδομένα σε διαφορετικές περιοχές έχουν διαφορετικές κατανομές ως προς τα χαρακτηριστικά τους. Υιοθετήθηκαν προεκπαιδευμένα auto-encoders νευρωνικά δίκτυα για τη δημιουργία αντιστοίχισης χαρακτηριστικών από ένα αρχικό στιγμιότυπο σε μια κρυφή αναπαράσταση και χρησιμοποιήθηκαν τα δεδομένα προέλευσης για την εκπαίδευση έναν ταξινομητή για προβλέψεις σε νέους στόχους.



Σχήμα 8: Προτεινόμενο μοντέλο μελέτης [13].

Μια ακόμη ερευνητική εργασία που εστιάζει στην αναγκαιότητα της αξιολόγησης και της παρακολούθησης της ποιότητας του αέρα για την κατανόηση των επιπτώσεων της ρύπανσης στην υγεία του πληθυσμού είναι η [14]. Οι ερευνητές διαπίστωσαν ότι για να κλιμακωθεί ο έλεγχος σε επίπεδο πόλης, οι σταθεροί σταθμοί ανίχνευσης δεν επαρκούν λόγω της έλλειψης κάλυψης, του κόστους εγκατάστασης και συντήρησης, καθώς και της επίλυσης των παρατηρούμενων φαινομένων. Οι συγγραφείς μέσω της πλατφόρμας OpenSense αποσκοπούν στην ανάπτυξη ενός ολοκληρωμένου πολυεπίπεδου συστήματος ανίχνευσης, σε κλίμακα πόλης, που συνδυάζει διαφορετικά επίπεδα ανίχνευσης ποιότητας αέρα όπως σταθμούς αναφοράς, κινητή ανίχνευση στα μέσα μαζικής μεταφοράς και συμμετοχική ανίχνευση πλήθους. Με την εφαρμογή του προτεινόμενου συστήματος μπορούν να κατανοηθούν οι επιπτώσεις στην υγεία από την έκθεση στην αστική ατμοσφαιρική ρύπανση και να εξαχθούν χάρτες ποιότητας του αέρα υψηλής ανάλυσης.

Επιπλέον, προτείνεται μια σημασιολογική προσέγγιση για τον χαρακτηρισμό και την έκθεση των δεδομένων ποιότητας του αέρα, ώστε να είναι διαθέσιμα στους πολίτες και τους προγραμματιστές εφαρμογών με τρόπο ώστε να μπορούν να χρησιμοποιηθούν και να κατανοηθούν αποτελεσματικά. Το προτεινόμενο μοντέλο απαρτίζεται από επιμέρους επίπεδα, όπου σε καθένα από αυτά παρέχεται πρόσθετη αξία στα δεδομένα, όπως σημασιολογικοί σχολιασμοί που περιγράφουν τον καθαρισμό και την προεπεξεργασία δεδομένων, τις χρονικές και χωρικές συναθροίσεις και τους σχολιασμούς συμβάντων. Αυτή η προσέγγιση επιτρέπει στους χρήστες να επιλέξουν το επίπεδο δεδομένων που ταιριάζει καλύτερα στον προορισμό τους, δηλαδή ανεπεξέργαστα δεδομένα, χρονικές και χωρικές συναθροίσεις, αναπαραστάσεις χαρτών ρύπανσης κ.λπ..

Στον ακόλουθο πίνακα (Πίνακας 2) παρουσιάζονται τα κύρια χαρακτηριστικά των επιλεγμένων μελετών, για συστήματα σημασιολογικής επισημείωσης δεδομένων με σκοπό την παρακολούθηση της ποιότητας του αέρα σε έξυπνες πόλεις.

Πίνακας 2: Περίληψη σχετικών ερευνών.

Αναφορά	Year	Framework	Οντολογία	Επέκταση οντολογιών	Semantic Annotation	Semantic Reasoning	Big Data	Μηχανική Μάθηση	Τομέας Εφαρμογής
[6]	2017	Duy et al.	✓	SSN, TIME, SWEET, GEO	✓	✓	✗	✗	Παρακολούθηση ποιότητας αέρα
[8]	2021	Sejdiu et al.	onto-core.owl	SensorML, O&M, TansduserML	✓	✗	✓	✗	Παρακολούθηση ποιότητας αέρα
[9]	2021	IoTSAS	onto-core.owl	SensorML, O&M, TansduserML	✓	✗	✓	✗	Παρακολούθηση ποιότητας αέρα
[10]	2019	✗	DeviceModel.owl	Air Quality Index, Geonames	✗	✗	✗	Random Forests	Παρακολούθηση ποιότητας αέρα
[11]	2021	Rubi et al.	EISCO	ENVO, GCIO, SSN	✗	✗	✓	✗	Παρακολούθηση ποιότητας αέρα
[12]	2020	SSFW	SCEO	GEO W3C	✓	✓	✓	✗	Παρακολούθηση ποιότητας αέρα και καιρού
[13]	2016	Zhang et al.	✓	✗	✗	✗	✗	Transfer Learning	Ανίχνευση ρύπανσης, Ανίχνευση προτύπων κυκλοφορίας
[14]	2015	OpenSense	✓	SSN	✓	✗	✗	✗	Παρακολούθηση ποιότητας αέρα

3

Αρχιτεκτονική

3.1 Γενική αρχιτεκτονική μεθοδολογίας

Σε αυτή την εργασία, προτείνεται μια γενική μεθοδολογία συλλογής ετερογενών γεωχωρικών δεδομένων που μπορεί να υιοθετηθεί από διάφορους τομείς εφαρμογών. Η αρχιτεκτονική του συστήματος επικεντρώνεται στη συγκέντρωση δεδομένων από ετερογενείς κόμβους που αναπτύσσονται σε διαφορετικές γεωγραφικές θέσεις και ο εμπλουτισμός τους από ανεξάρτητες πηγές ανοικτών δεδομένων. Τα κύρια χαρακτηριστικά της μεθοδολογίας που υιοθετήθηκε είναι η ενσωμάτωση διάφορων τεχνολογιών με διαθέσιμη επικοινωνιακή υποδομή για τη συλλογή, αποθήκευση και επεξεργασία των δεδομένων. Το τελευταίο επίπεδο της συλλογής περιλαμβάνει τη συγκέντρωση, τη διαμόρφωση και την προετοιμασία των δεδομένων για την σημασιολογική τους επισήμανση. Η ανάλυση των δεδομένων πραγματοποιείται σε ανώτερο επίπεδο κάνοντας χρήση σημασιολογικών εργαλείων και τεχνικών, με σκοπό την εξαγωγή γνώσης για την παροχή νέων λειτουργιών και υπηρεσιών που εφαρμόζονται στο επίπεδο εφαρμογής.

Στο Σχήμα 3 απεικονίζεται η γενικευμένη αρχιτεκτονική της μεθοδολογίας με την διαστρωμάτωση των λειτουργιών από άκρο σε άκρο. Τα επιμέρους επίπεδα είναι το στρώμα πηγής δεδομένων, το επίπεδο δικτύου (πρωτόκολλα επικοινωνίας και δικτύωση), το επίπεδο IoT πλατφόρμας (αποθήκευση, επεξεργασία και ανάλυση δεδομένων), το επίπεδο εξατομικευμένων υπηρεσιών, το επίπεδο σημασιολογίας και το επίπεδο εφαρμογής. Αναλυτικότερα:

Επίπεδο πηγής δεδομένων: Είναι το πρώτο στρώμα της αρχιτεκτονικής και περιλαμβάνει αισθητήρες, έξυπνους μετρητές, ενεργοποιητές που παρακολουθούν υπηρεσίες, δραστηριότητες ή εξοπλισμό σε λειτουργία. Επιπρόσθετα, στο στρώμα αυτό ανήκουν σύνολα δεδομένων που μοιράζονται οι κυβερνητικές αρχές ή πηγές ανοικτών δεδομένων με δυνατότητα διασύνδεσης προγραμματισμού εφαρμογών (APIs). Ανάλογα με το είδος των πηγών και την περίπτωση εφαρμογής, τα δεδομένα μπορεί να αφορούν θερμοκρασία, υγρασία, τοποθεσία, ποιότητα αέρα κ.λπ.. Η κύρια λειτουργία αυτού του επιπέδου είναι να λαμβάνει πληροφορίες από το περιβάλλον ή από άλλες πηγές και να μεταβιβάζει τα δεδομένα σε ανώτερο επίπεδο, έτσι ώστε να μπορούν να πραγματοποιηθούν ορισμένες ενέργειες με βάση αυτές τις πληροφορίες.

Επίπεδο δικτύου: Αποτελεί το συνδετικό επίπεδο μεταξύ του επιπέδου πηγών δεδομένων και του πρώτου επιπέδου ενδιάμεσου λογισμικού (Επίπεδο IoT πλατφόρμας). Λαμβάνει δεδομένα από

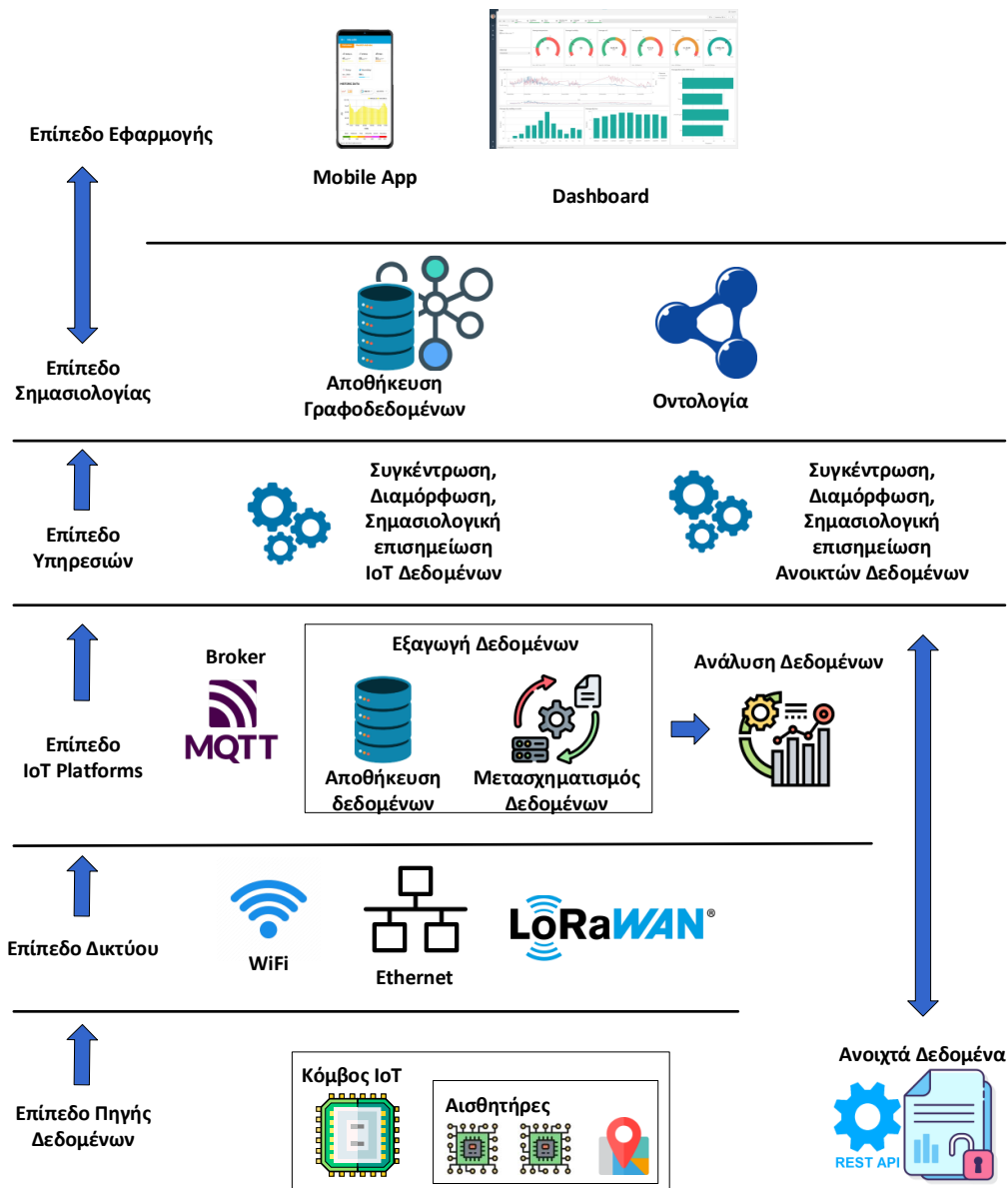
το αρχικό επίπεδο και τα προωθεί χρησιμοποιώντας ενσύρματες ή ασύρματες τεχνολογίες δικτύωσης όπως 3G, 4G, WiFi, Bluetooth, Ethernet, LoRa, υπέρυθρες κ.λπ., ανάλογα με το είδος της πηγής. Το επίπεδο δικτύωσης συσχετίζει όλες τις πηγές δεδομένων και τους επιτρέπει να μοιράζονται τις πληροφορίες με άλλα συνδεδεμένα συστήματα. Όλη η μεταφορά των δεδομένων μπορεί να πραγματοποιηθεί με ασφάλεια, ανάλογα με το είδος του πρωτοκόλλου που χρησιμοποιείται, διατηρώντας τα δεδομένα που λαμβάνονται εμπιστευτικά. Στο επίπεδο αυτό ανήκουν και οι δικτυακές πύλες οι οποίες προσφέρουν τη δυνατότητα έμμεσης διασύνδεσης των συσκευών του προηγούμενου επιπέδου, στο διαδίκτυο.

Επίπεδο IoT Platform: Οι κύριες αρμοδιότητες αυτού του επιπέδου είναι: α) Η ενσωμάτωση δεδομένων που λαμβάνονται από διαφορετικά είδη συνδεδεμένων συσκευών, β) Η επεξεργασία των ληφθέντων δεδομένων, γ) Η παροχή των ληφθέντων δεδομένων σε διάφορες εφαρμογές ή υπηρεσίες. Οι IoT πλατφόρμες μπορούν να επικοινωνήσουν απευθείας με τη συσκευή εάν και οι δύο χρησιμοποιούν συμβατές τεχνολογίες και πρωτόκολλα. Το επίπεδο πλατφόρμας IoT είναι επίσης υπεύθυνο για την παροχή λειτουργιών όπως η αποθήκευση χρονοσειρών σε βάση δεδομένων, ο μετασχηματισμός, η ανάλυση και η διαχείρισή των δεδομένων που ελήφθησαν από τις απομακρυσμένες συσκευές. Επιπρόσθετα στο επίπεδο αυτό περιλαμβάνονται εργαλεία οπτικοποίησης των επεξεργασμένων δεδομένων σε γραφικούς πίνακες. Οι υπηρεσίες του επιπέδου αυτού είναι συνήθως προσβάσιμες μέσω RESTAPI που βασίζονται σε HTTP, ενώ ταυτόχρονα υλοποιούν τις κατάλληλες υποδομές του πρωτοκόλλου MQTT για την προώθηση των δεδομένων σε υπηρεσίες ή εφαρμογές που ανήκουν σε ανώτερα επίπεδα.

Επίπεδο Υπηρεσιών: Είναι το τελευταίο επίπεδο της μεθόδου συλλογής δεδομένων και σε αυτό υλοποιούνται εξατομικευμένες υπηρεσίες για την σημασιολογική επισήμανση των δεδομένων σε πραγματικό χρόνο. Σε αυτό το στάδιο τα δεδομένα συλλέγονται από το επίπεδο IoT Platform, μετασχηματίζονται σε μια ενιαία μορφή και προετοιμάζονται για τη σημασιολογική τους επισήμανση όπως ορίζεται από την οντολογία που συναντάμε στο επόμενο στάδιο. Επιπρόσθετα, συλλέγονται πληροφορίες από ανοικτές πηγές δεδομένων τα οποία υπόκεινται σε αντίστοιχες διεργασίες με αυτές των IoT δεδομένων.

Επίπεδο Σημασιολογίας: Στο επίπεδο αυτό τα συλλεγμένα και διαμορφωμένα δεδομένα των αισθητήρων, μοντελοποιούνται χρησιμοποιώντας τεχνικές σημασιολογικού ιστού όπως είναι οι γλώσσες Μηχανικής Οντολογίας OWL και RDF. Τα δεδομένα αντιστοιχίζονται στη σημασιολογική τους μορφή, ταξινομούνται στις αντίστοιχες κατηγορίες σύμφωνα με την οντολογία και αποθηκεύονται σε βάσεις γραφοδεδομένων. Τα αποθηκευμένα δεδομένα αναλύονται με την εκτέλεση ερωτημάτων με σκοπό την αναγνώριση μοτίβων και την εξαγωγή γνώσης όπως είναι η κατανόηση ανωμαλιών ή κρίσιμων γεγονότων.

Επίπεδο Εφαρμογής: Το επίπεδο αυτό είναι υπεύθυνο για την παράδοση συγκεκριμένων υπηρεσιών στο χρήστη. Διαχειρίζεται όλη τη διαδικασία εφαρμογής σύμφωνα με τις πληροφορίες που λαμβάνονται από το επίπεδο σημασιολογίας. Ορίζει διάφορες εφαρμογές στις οποίες το Διαδίκτυο των Πραγμάτων μπορεί να αναπτυχθεί, όπως έξυπνα σπίτια, έξυπνα πόλεις, έξυπνη υγεία και γεωργία ακριβείας.

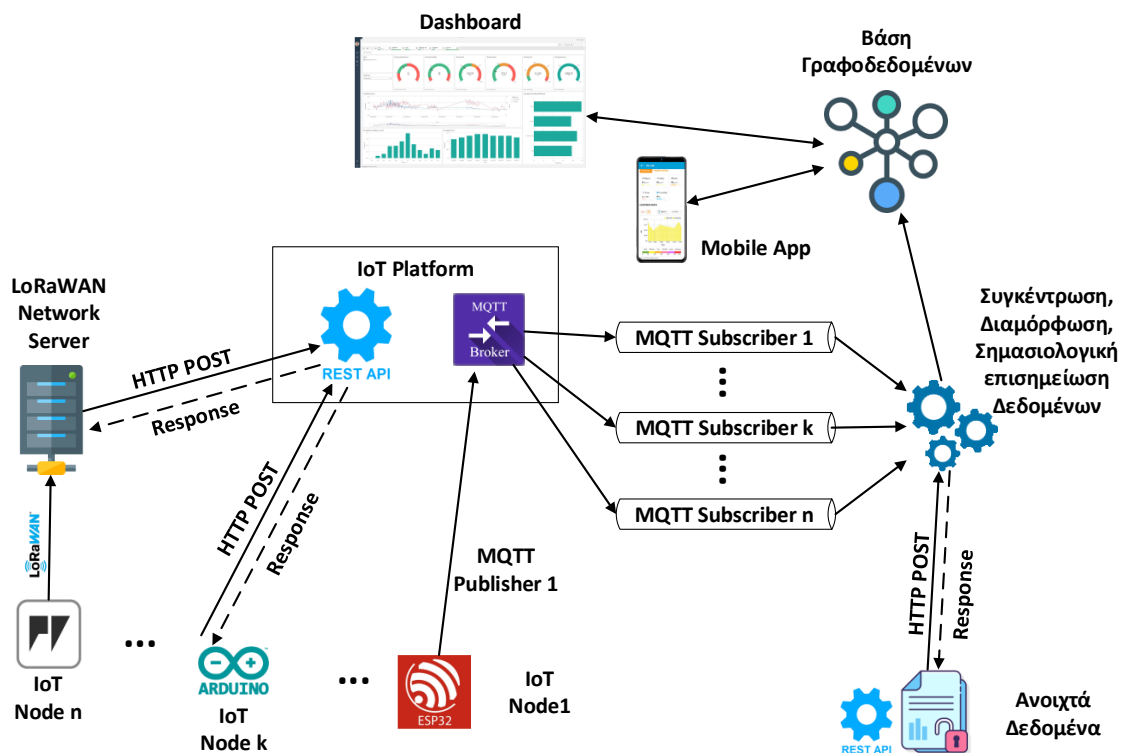


Σχήμα 9: Γενική αρχιτεκτονική ανάπτυξης εφαρμογών στηριζόμενη στη συλλογή ετερογενών δεδομένων και στον σημασιολογικό εμπλουτισμό τους.

3.2 Αρχιτεκτονική σε επίπεδο υλοποίησης

Στην ενότητα αυτή παρουσιάζεται η αρχιτεκτονική της μεθοδολογίας συλλογής ετερογενών γεωχωρικών δεδομένων, σε επίπεδο εφαρμογής (Σχήμα 10). Ετερογενείς συσκευές με διαφορετική τεχνολογία κατασκευής συνδέονται σε αντίστοιχο δίκτυο (WiFi, Ethernet, LoRaWAN) ανάλογα με τις δυνατότητες που προσφέρει το εκάστοτε υλικό. Για την επικοινωνία των συσκευών με τις IoT πλατφόρμες και την μεταφορά των δεδομένων, χρησιμοποιούνται τα δικτυακά πρωτόκολλα HTTP και MQTT, αξιοποιώντας RESTfull HTTP APIs και τον MQTT Broker των πλατφορμών. Στην περίπτωση των συσκευών που χρησιμοποιούν τη δικτύωση LoRaWAN, παρεμβάλλεται μια δικτυακή πύλη κατάλληλης τεχνολογίας, μέσω της οποίας η συσκευή αποκτά πρόσβαση στο

διαδίκτυο. Η επικοινωνία μεταξύ της δικτυακής πύλης και της πλατφόρμας IoT γίνεται μέσω του πρωτοκόλλου HTTP.



Σχήμα 10: Αρχιτεκτονική σε επίπεδο εφαρμογής.

Μια εξατομικευμένη υπηρεσία πραγματοποιεί εγγραφές στα κανάλια της IoT πλατφόρμας τα οποία συλλέγουν τα δεδομένα από τις αντίστοιχες συσκευές, κάνοντας χρήση του πρωτοκόλλου MQTT. Κάθε νέα πληροφορία που παραλαμβάνεται από την IoT πλατφόρμα προωθείται στην υπηρεσία μέσω του MQTT Broker. Σε αυτό το στάδιο η υπηρεσία παραλαμβάνει ένα JSON αντικείμενο, το οποίο κατακερματίζεται, φιλτράρεται η ωφέλιμη πληροφορία και μετασχηματίζεται σε μια ενιαία μορφή. Η πληροφορία που προέκυψε από την παραπάνω διαδικασία προετοιμάζεται κατάλληλα για την σημασιολογική της επισημείωση και την εγγραφή της στη βάση γραφοδεδομένων. Επιπρόσθετα μια νέα εξατομικευμένη υπηρεσία αναλαμβάνει τη συλλογή ανοιχτών δεδομένων μέσω RESTfull HTTP API αιτημάτων. Οι απαντήσεις των αιτημάτων είναι σε μορφή JSON αντικειμένων τα οποία υπόκεινται στην ίδια διεργασία που ακολουθήθηκε στα IoT δεδομένα.

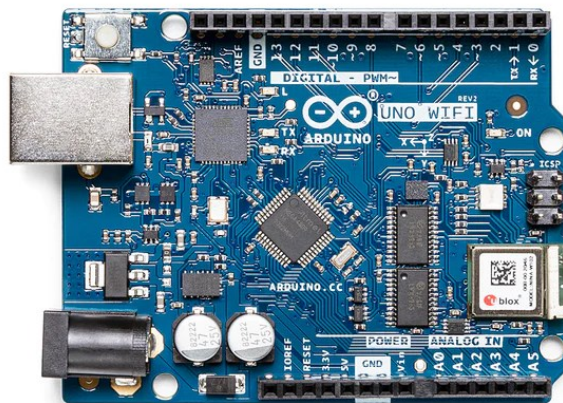
Στη συνέχεια τα διαμορφωμένα δεδομένα κατακερματίζονται και μοντελοποιούνται σύμφωνα με την οντολογία που έχει οριστεί στα πλαίσια της εφαρμογής. Η οντολογία περιλαμβάνει τις οντότητες που περιγράφουν τους IoT κόμβους, τους κόμβους των πηγών ανοιχτών δεδομένων, τη γεωγραφική τους θέση, τα μετρούμενα μεγέθη, τις τιμές τους στο πέρασμα του χρόνου καθώς και τις σχέσεις μεταξύ τους. Η σημασιολογική επισημείωση των δεδομένων πραγματοποιείται από μια υπηρεσία που εκτελεί τη μοντελοποίηση και την εγγραφή τους στη βάση γραφοδεδομένων. Τα αποθηκευμένα δεδομένα υπόκεινται σε ανάλυση με την εκτέλεση κατάλληλων ερωτημάτων, για την εξαγωγή γνώσης και της απεικόνισής της στο χρήστη μέσω της τελικής εφαρμογής.

4

IoT Πλατφόρμες Υλικού

4.1 Arduino UNO WIFI REV2

Το Arduino UNO WiFi Rev.2 (Σχήμα 11) αποτελεί μια από τις πιο διαδεδομένες αναπτυξιακές πλακέτες συσκευών IoT, παρέχοντας ευελιξία στην υλοποίηση βασικών σεναρίων εφαρμογών του Διαδικτύου των Πραγμάτων [15].



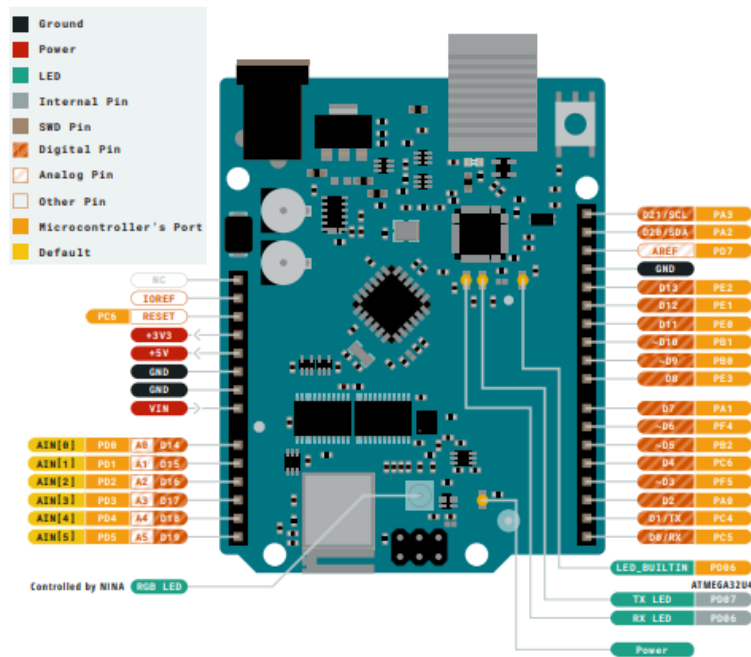
Σχήμα 11: Arduino UNO WIFI REV2.

Η συσκευή ενσωματώνει τον μικροελεγκτή 8-bit ATmega4809 της Microchip με ταχύτητα χρονισμού έως 20 MHz, 48 KB Flash, 6 KB SRAM και 256 byte EEPROM. Το Arduino UNO WiFi Rev.2 διαθέτει 14 ψηφιακές εισόδους/εξόδους, εκ των οποίων 5 από αυτές μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, 6 αναλογικές εισόδους, σύνδεση USB, υποδοχή τροφοδοσίας, σύνδεση προγραμματισμού ICSP και κουμπί επαναφοράς (Σχήμα 12).

Η πλακέτα διαθέτει το αυτόνομο SoC u-blox NINA-W102 με ενσωματωμένη στοίβα πρωτοκόλλου TCP/IP που μπορεί να παρέχει πρόσβαση σε ένα δίκτυο Wi-Fi ή να λειτουργεί ως σημείο πρόσβασης. Η συσκευή παρέχει υποστήριξη για Wi-Fi 802.11b/g/n στη ζώνη ISM 2,4 GHz και επικοινωνίες Bluetooth v4.2 (Bluetooth BR/EDR και Bluetooth χαμηλής ενέργειας). Το chipset επικοινωνίας στο Arduino UNO WiFi Rev.2 μπορεί να λαμβάνει ρόλο τόσο Bluetooth® όσο και Bluetooth® Low Energy-πελάτης και συσκευή υποδοχής.

Επιπρόσθετα, η πλακέτα διαθέτει έναν επιταχυντή κρυπτογράφησης ATECC608A, διασφαλίζοντας ότι τα δεδομένα παραμένουν ασφαλή και ιδιωτικά αποθηκεύοντας έως και 16 κλειδιά στη μνήμη EEPROM. Το ολοκληρωμένο κύκλωμα ATECC608A είναι κατάλληλο για διαχείριση και ανταλλαγή κλειδίων τερματικού σε δίκτυο IoT, για κρυπτογράφηση δεδομένων και μικρών μηνυμάτων, για ασφαλή εκκίνηση συστήματος και προστασία κλωνοποίησης συσκευής. Τέλος, το Arduino UNO WiFi Rev.2 ενσωματώνει μια περιφερειακή μονάδα IMU LSM6DS3TR η οποία περιλαμβάνει 3D ψηφιακό επιταχυνσιόμετρο και γυροσκόπιο.

Ακολουθεί πίνακας με τα τεχνικά χαρακτηριστικά του Arduino UNO WiFi Rev.2 (Πίνακας 3).



Σχήμα 12: Διάγραμμα ακροδεκτών Arduino UNO WiFi Rev.2

Πίνακας 3: Τεχνικά Χαρακτηριστικά Arduino UNO WiFi Rev.2

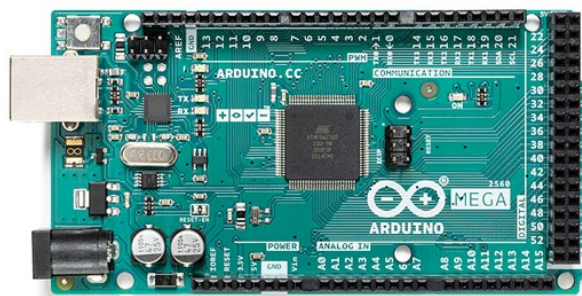
A/A	Χαρακτηριστικό	Περιγραφή	
1	Τύπος Σύνδεσης	USB-B	
2	Μικροεπεξεργαστής	ATmega4809	
3	Συχνότητα Χρονισμού	≤ 20 MHz	
	Μνήμη	FLASH	48KB

4		SRAM	6144 Bytes
		EEPROM	256 Bytes
5	Συνδεσιμότητα	Bluetooth	u-blox NINA-W102
		Wi-Fi	u-blox NINA-W102
		Κρυπτογράφηση	ATECC608A
6	Ψηφιακές I/O		14
7	Αναλογικές εισοδοι		6
8	Αναλογικές έξοδοι PWM		5
9	Επικοινωνία	UART	Ναι
		I2C	Ναι
		SPI	Ναι
10	Αισθητήρες		LSM6DS3TR
11	Τάση εισόδου (ονομαστική)		6-20V
12	Τάση I/O		5V
13	Ρεύμα ανά I/O		20 mA

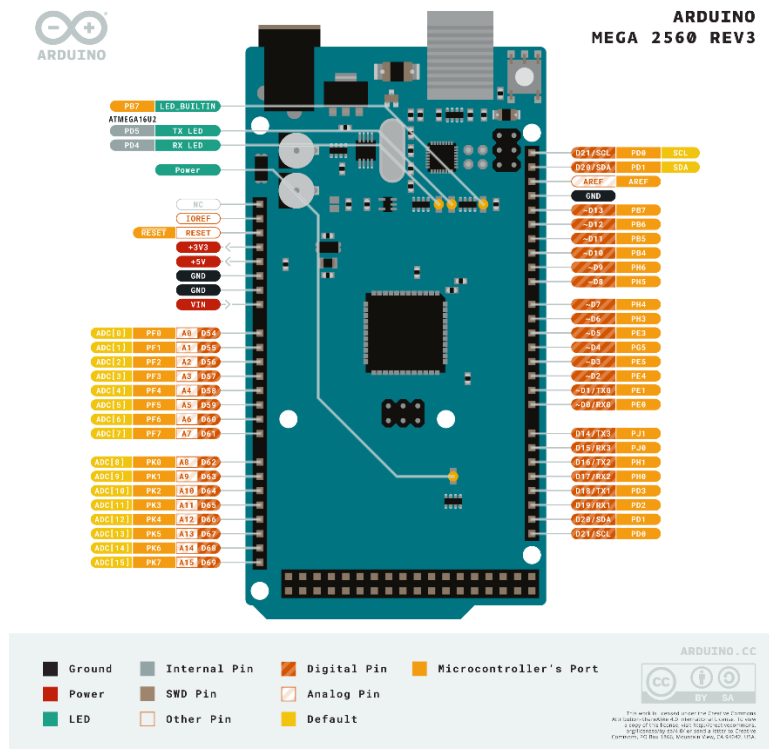
4.2 Arduino MEGA 2560 REV3

Το Arduino Mega 2560 είναι μια πλακέτα που βασίζεται στον 8-bit επεξεργαστή ATmega2560 με ταχύτητα χρονισμού 16 MHz, 256 KB Flash, 8 KB SRAM και 4KB (Σχήμα 13). Διαθέτει 54 ψηφιακές εισόδους/εξόδους (από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν ως αναλογικές έξοδοι PWM), 16 αναλογικές εισόδους, 4 UARTs, σύνδεση USB, υποδοχή τροφοδοσίας, σύνδεση προγραμματισμού ICSP, και κουμπί επαναφοράς (Σχήμα 14) [16].

Η πλακέτα δεν διαθέτει αυτοτελώς κατάλληλη υποδομή για ασύρματη ή ενσύρματη δικτύωση. Παρέχεται από τον κατασκευαστή δυνατότητα διασύνδεσης με επιπρόσθετες πλακέτες που διαθέτουν χαρακτηριστικά κατάλληλα για οποιαδήποτε διασύνδεση.



Σχήμα 13: Arduino MEGA 2560 REV3



Σχήμα 14: Διάγραμμα ακροδεκτών Arduino MEGA 2560 REV3.

Ακολουθεί πίνακας με τα τεχνικά χαρακτηριστικά του Arduino MEGA (Πίνακας 4).

Πίνακας 4: Τεχνικά Χαρακτηριστικά Arduino MEGA 2560 REV3.

A/A	Χαρακτηριστικό	Περιγραφή	
1	Τύπος Σύνδεσης	USB-B	
	Μικροεπεξεργαστής	ATmega2560	
2	Συχνότητα Χρονισμού	16 MHz	
3	Μνήμη	FLASH	256KB
		SRAM	8KB
		EEPROM	4KB
4	Ψηφιακές I/O	54	
5	Αναλογικές εισοδοι	16	
6	Αναλογικές έξοδοι PWM	15	
7	Επικοινωνία	UART	Ναι, 4
		I2C	Ναι
		SPI	Ναι

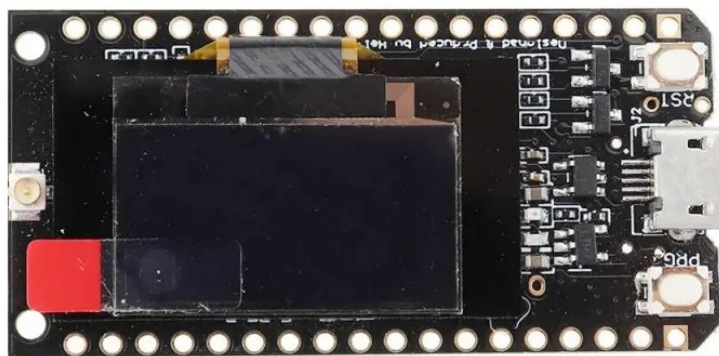
8	Τάση εισόδου (ονομαστική)	7-12V
9	Τάση I/O	5V
10	Ρεύμα ανά I/O	20 mA

4.3 TTGO LoRa32 v1.0

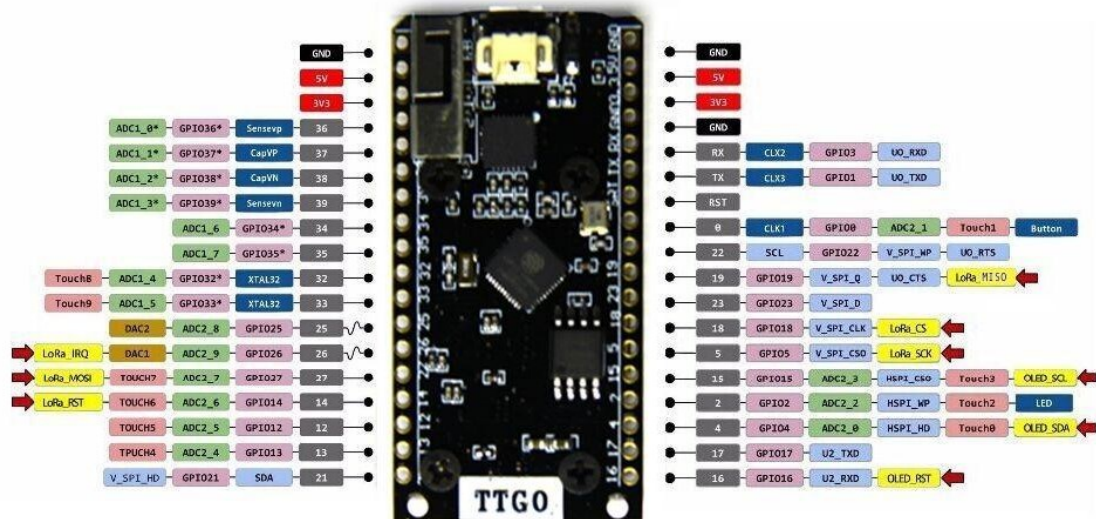
Το TTGO LoRa32 (Σχήμα 15) είναι μια συσκευή που παρουσιάστηκε από τη Semtech και αποτελεί μια από τις πρώτες που εμφανίστηκαν στην αγορά, προσφέροντας τη δυνατότητα επικοινωνίας μέσω της τεχνολογίας LoRa [17]. Λειτουργεί με μια ζώνη ραδιοσυχνοτήτων μεγάλης εμβέλειας για τη μετάδοση ή λήψη μικρών πακέτων δεδομένων. Ανάλογα με τη γεωγραφική θέση χρήσης της τεχνολογίας, αντιστοιχούν συγκεκριμένες ζώνες συχνοτήτων LoRa. Για το λόγο αυτό διατίθενται στην αγορά διαφορετικές εκδόσεις και η επιλογή από τον χρήστη πρέπει να γίνει με ιδιαίτερη προσοχή.

Η μονάδα αποτελείται από το Espressif ESP32, έναν 32-bit επεξεργαστή διπλού πυρήνα Tensilica LX6 με συχνότητα χρονισμού στα 240 MHz, QSPI FLASH 4 MB, 540 KB SRAM και 4096 Bytes EEPROM. Το TTGO LoRa32 συνοδεύεται από το Semtech SX1276 για την επικοινωνία LoRa και από μια οθόνη OLED 0.96". Η συσκευή διαθέτει 28 ακροδέκτες GPIO πολλαπλών χρήσεων και μπορούν να χρησιμοποιηθούν ως ψηφιακές εισοδοί/έξοδοι εάν δεν χρησιμοποιούνται οι άλλες αντίστοιχες λειτουργίες τους. Το ESP32 διαθέτει επίσης έναν ενσωματωμένο αισθητήρα hall-effect που ανιχνεύει αλλαγές στο μαγνητικό πεδίο του περιβάλλοντός του [18], [19].

Η συσκευή παρέχει υποστήριξη για επικοινωνία μέσω των πρωτοκόλλων Wi-Fi 802.11b/g/n στη ζώνη ISM 2,4 GHz και Bluetooth v4.2 (Bluetooth BR/EDR και Bluetooth χαμηλής ενέργειας).



Σχήμα 15: TTGO LoRa32 v1.0



Σχήμα 16: Διάγραμμα ακροδεκτών TTGO LoRa32.

Ακολουθεί πίνακας με τα τεχνικά χαρακτηριστικά του Arduino UNO WiFi Rev.2 (Πίνακας 5).

Πίνακας 5: Τεχνικά χαρακτηριστικά TTGO LoRa32.

A/A	Χαρακτηριστικό	Περιγραφή	
1	Τύπος Σύνδεσης	Micro-USB	
2	Μικροεπεξεργαστής	Tensilica LX6	
3	Συχνότητα Χρονισμού	≤ 240 MHz	
4	Μνήμη	FLASH	4MB
		SRAM	540KB
		EEPROM	4096 Bytes
5	Συνδεσιμότητα	Bluetooth	Ναι
		Wi-Fi	Ναι
		LoRa	Ναι
6	Ψηφιακές I/O	≤ 18	
7	Αναλογικές εισοδοι	≤ 12	
8	Αναλογικές έξοδοι PWM	≤ 1	
9	Επικοινωνία	UART	Ναι
		I2C	Ναι
		SPI	Ναι
		I2S	Ναι

10	Αισθητήρες	Hall Effect
11	Τάση εισόδου (ονομαστική)	2.7V-3.3V
12	Τάση I/O	3.3V
13	Ρεύμα ανά I/O	40 mA

4.4 LoPy4

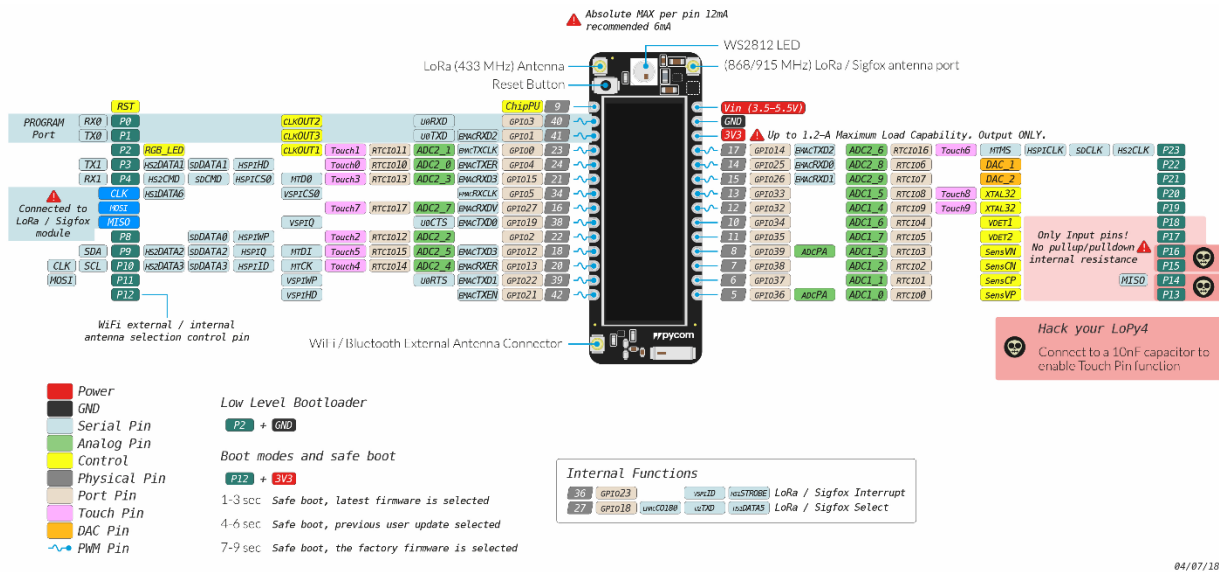
Το Lopy4 είναι μια αναπτυξιακή πλακέτα της εταιρίας Pycom που προσφέρει πολλές επιλογές διασύνδεσης μέσω ραδιοσυχνοτήτων (RF) όπως είναι οι τεχνολογίες LoRa, SIGFOX, Bluetooth (Classic and Low Energy, BLE) και Wi-Fi. Σε αντίθεση με τις περισσότερες πλακέτες μικροελεγκτών, το LoPy4 (Σχήμα 17) διαθέτει ένα ενσωματωμένο λειτουργικό σύστημα που υποστηρίζει τη γλώσσα προγραμματισμού MicroPython, η οποία αποτελεί ένα ειδικό υποσύνολο της γλώσσας προγραμματισμού Python 3 και των βιβλιοθηκών για μικροελεγκτές.

Η μονάδα διαθέτει το Espressif ESP32, το οποίο περιέχει έναν επεξεργαστή διπλού πυρήνα Xtensa 32 bit LX6 που λειτουργεί με συχνότητα χρονισμού έως και 240 MHz, 8 MB μνήμης flash (για την αποθήκευση του κώδικα προγράμματος και ορισμένων αρχείων σε ένα σύστημα αρχείων) και 520 KB RAM (για αποθήκευση μεταβλητών, πληροφοριών κατάστασης και buffer). Η μονάδα ESP32 διαθέτει επίσης ενσωματωμένη συνδεσιμότητα WiFi και BLE. Επιπλέον, το LoPy4 διαθέτει 4 MB PSRAM (ψευδοστατική RAM) που χρησιμοποιείται ως επέκταση μνήμης για το ESP32 [20].



Σχήμα 17: Lopy4.

Η πλακέτα προσφέρει 18 ακροδέκτες εισόδου/εξόδου γενικής χρήσης (18 GPIO), από τους οποίους έως και 12 μπορούν να χρησιμοποιηθούν ως ακροδέκτες αναλογικής εισόδου (με αναλογικούς ψηφιακούς μετατροπείς 12 bit (ADC)) και δύο ως ακροδέκτες αναλογικής εξόδου (ψηφιακός αναλογικός μετατροπέας 8 bit (DAC)). Οι περισσότερες ακίδες GPIO μπορούν να ρυθμιστούν για συγκεκριμένα πρωτόκολλα υλικού. Μπορούν να χρησιμοποιηθούν συνολικά 3 σειριακές θύρες UART, 2 θύρες I2C, 3 θύρες SPI, 1 διάυλος CAN, 1 κανάλι PWM και μια θύρα I2S (Σχήμα 18).



Σχήμα 18: Διάγραμμα ακροδεκτών LoPy4.

Η πλακέτα μπορεί να προγραμματιστεί μέσω της σειριακής διεπαφής ή μέσω WiFi χρησιμοποιώντας σύνδεση telnet ή FTP. Από προεπιλογή, το LoPy4 λειτουργεί ως σημείο πρόσβασης Wi-Fi (SSID: lopy4-wlan-XXXX, Κωδικός πρόσβασης: www.pycom.io) και ένας χρήστης μπορεί να συνδεθεί στη μονάδα αφού εγγραφεί στο δίκτυο WiFi για να ανεβάσει προγράμματα και αρχεία δεδομένων.

Το Lopy4 διαθέτει ενσωματωμένη κεραμική κεραία για την επικοινωνία μέσω WiFi και Bluetooth. Επίσης παρέχεται η δυνατότητα σύνδεσης και εξωτερικής κεραίας που μπορεί να συνδεθεί μέσω μιας υποδοχής τύπου U.FL. Διατίθενται ενσωματωμένες αντίστοιχες υποδοχές για την προσθήκη κεραίας LoRa ή SIGFOX. Η συσκευή χρησιμοποιεί το ολοκληρωμένο κύκλωμα LoRa SX1276 της εταιρείας Semtech και μπορεί να ρυθμιστεί ώστε να λειτουργεί είτε στη ζώνη συχνοτήτων των 433 MHz, 868 MHz ή 915 MHz. Η μονάδα LoRa συνδέεται μέσω διασύνδεσης SPI στον μικροελεγκτή και όλες οι απαιτούμενες συνδέσεις των ακίδων πομποδέκτη LoRa με τον μικροελεγκτή είναι ήδη ενσωματωμένες στην πλακέτα LoPy4. Εφόσον η μονάδα υλοποιεί μόνο το φυσικό επίπεδο LoRa, η στοίβα πρωτοκόλλου LoRaWAN υλοποιείται σε λογισμικό στον μικροελεγκτή. Η εφαρμοζόμενη λειτουργικότητα LoRaWAN είναι συμβατή με LoRaWAN Class A/C.

4.5 Σύγκριση Μικροελεγκτών

Πίνακας 6: Συγκριτικός πίνακας μικροελεγκτών εφαρμογής.

A/A	Χαρακτηριστικό	Arduino UNO WiFi Rev.2	Arduino MEGA 2560 REV3	TTGO Lora32 v1.0	Lopy4	
1	Τύπος Σύνδεσης	USB-B	USB-B	Micro USB	Όχι	
2	Μικροεπεξεργαστής	ATmega4809	ATmega2560	Tensilica LX6	Tensilica LX6	
3	Bit	8	8	32	32	
4	Πυρήνες	1	1	2	2	
5	Συχνότητα Χρονισμού	≤ 20 MHz	16 MHz	≤ 240 MHz	≤ 240 MHz	
6	Μνήμη	FLASH	48KB	256KB	4MB	8MB
		SRAM	6144 Bytes	8KB	540KB	520KB
		EEPROM	256 Bytes	4KB	4096 Bytes	×
7	Συνδεσιμότητα	Bluetooth	✓	×	✓	✓
		Wi-Fi	✓	×	✓	✓
		LoRa	×	×	✓	✓
		Sigfox	×	×	×	✓
		Κρυπτογράφηση	✓	×	×	×
8	Ψηφιακές I/O GPIO	14	54	≤ 18	≤ 18	
9	Αναλογικές είσοδοι ADC	6	16	≤ 12	≤ 12	
10	Αναλογικές έξοδοι DAC	×	×	1	2	

11	Έξοδοι PWM		5	15	1	1
12	Επικοινωνία	UART	1	4	1	2
		I2C	1	1	1	1
		SPI	1	1	×	2
		I2S	×	×	✓	✓
		CAN	×	×	×	✓
13	Αισθητήρες		LSM6DS3TR	×	Hall Effect	×
14	Τάση εισόδου (ονομαστική)		6-20V	7-12V	2.7V-3.3V	3,3V-5.5V
15	Τάση I/O		5V	5V	3.3V	3.3V
16	Ρεύμα ανά I/O		20 mA	20 mA	40 mA	40 mA
17	Οθόνη		×	×	✓	×
18	Προγραμματισμός		Arduino IDE, Platformio	Arduino IDE, Platformio	Arduino IDE, Platformio	Micropython Pymark
19	Ενδεικτικό Κόστος		46,70 €	42,00 €	33,15 €	38,45 €

5

Δίκτυα Επικοινωνίας

5.1 Wi-Fi

Το Wi-Fi αποτελεί την πιο κοινή τεχνολογία ασύρματου δικτύου που χρησιμοποιείται στον τομέα των ηλεκτρονικών ειδών ευρείας κατανάλωσης αλλά και στις επιχειρήσεις. Το Wi-Fi συναντάται σε οικιακούς και επαγγελματικούς χώρους, σε μέσα μεταφοράς όπως τρένα και αεροπλάνα, σε εμπορικά κέντρα και σε πολλά άλλα μέρη. Το Wi-Fi έχει διαδραματίσει θεμελιώδη ρόλο στην παροχή καινοτομίας στο Διαδίκτυο των Πραγμάτων, παρέχοντας διάχυτη συνδεσιμότητα για τη σύνδεση μιας μεγάλης ποικιλίας «αντικειμένων» μεταξύ τους, στο διαδίκτυο. Οι οικονομικές δυνατότητες του Διαδικτύου των Πραγμάτων είναι απεριόριστες και το Wi-Fi προσφέρει ένα ευρύ φάσμα ευκαιριών σε διάφορους τομείς, όπως έξυπνα σπίτια, έξυπνες πόλεις, αυτοκίνητα, υγειονομική περίθαλψη, επιχειρήσεις, κυβερνήσεις και βιομηχανικά περιβάλλοντα IoT.

Το Wi-Fi βασίζεται στο πρότυπο IEEE 802.11, το οποίο αποτελεί μέρος των προτύπων IEEE 802 (πρότυπα LAN). Τα πρότυπα IEEE 802.11 έχουν υποστεί πολλαπλές διαμορφώσεις ή τροποποιήσεις (IEEE 802.11b, IEEE 802.11g, IEEE 802.11n), με κάθε τροποποίηση να υποστηρίζει διαφορετικό ρυθμό μετάδοσης δεδομένων ή το εύρος επικοινωνίας (εσωτερικών και εξωτερικών χώρων). Σύμφωνα με το μοντέλο OSI, τα πρότυπα IEEE 802.11 καλύπτουν το φυσικό επίπεδο και το επίπεδο σύνδεσης δεδομένων, επομένως ορίζουν τα πρότυπα και τα πρωτόκολλα που εφαρμόζουν οι προμηθευτές τεχνολογίας στο επίπεδο 1 και το επίπεδο 2 του ασύρματου τοπικού δικτύου.

Τα δίκτυα Wi-Fi είναι ασφαλή, αξιόπιστα, γρήγορα και έχουν παρόμοιο επίπεδο απόδοσης με τα ενσύρματα δίκτυα Ethernet. Τα χαρακτηριστικά αυτά σε συνδυασμό με τη φορητότητα των διασυνδεδεμένων συσκευών, τα καθιστούν ευρέως διαδεδομένα. Τα δίκτυα Wi-Fi λειτουργούν στις ζώνες ραδιοσυχνοτήτων 2.4 GHz και 5 GHz, ενώ μπορούν να χρησιμοποιηθούν και κάποιες άλλες

ζώνες συχνοτήτων, όπως 6 GHz και 60 GHz. Οι ζώνες ραδιοσυχνοτήτων 2.4GHz και 5 GHz και άλλες ζώνες ραδιοσυχνοτήτων Wi-Fi βρίσκονται στο μη αδειοδοτημένο ραδιοφάσμα.

Στα δίκτυα Wi-Fi μπορεί να παρατηρηθεί παρεμβολή σήματος από άλλες ασύρματες συσκευές που χρησιμοποιούν ή λειτουργούν στην ίδια ζώνη συχνοτήτων. Επίσης, οι υψηλές συχνότητες απαιτούν περισσότερη ισχύ, γεγονός που το καθιστά ενεργοβόρο για τις συσκευές/μικροελεγκτές IoT.

Το Wi-Fi χρησιμοποιεί συνήθως την τοπολογία δικτύου αστέρα, όπου το AP λειτουργεί ως μεταγωγέας, ή δρομολογητής. Όλες οι συσκευές Wi-Fi συνδέονται σε αυτό αντί να συνδέονται απευθείας μεταξύ τους. Υπάρχει επίσης άλλη μια τοπολογία δικτύου που χρησιμοποιείται στα δίκτυα Wi-Fi, η οποία επιτρέπει τη διασύνδεση μεταξύ σημείων (PtP) όπου οι συσκευές επικοινωνούν απευθείας μεταξύ τους. Στην τοπολογία αστέρα, ένας δρομολογητής λειτουργεί ως σταθμός βάσης ή δέκτης στο ένα άκρο και συνδέεται στο Διαδίκτυο στο άλλο άκρο. Η σύνδεση στο Διαδίκτυο στην περίπτωση τοπολογίας σημείου προς σημείο έχει μεγαλύτερο βαθμό δυσκολίας δύσκολη ή ακριβή, καθώς κάθε συσκευή πρέπει να έχει συνδεσιμότητα στο Διαδίκτυο χωρίς να περνάει από κάποια κεντρική πύλη.

Μία από τις μεγαλύτερες απειλές σε οποιοδήποτε ασύρματο δίκτυο είναι η ασφάλεια, ειδικά με τεχνολογίες φάσματος χωρίς αδειδότηση, όπως είναι το Wi-Fi. Τα δίκτυα Wi-Fi ασφαλίζουν ή επαληθεύουν ποιος μπορεί να έχει πρόσβαση στο δίκτυο και να κρυπτογραφήσει δεδομένα που μεταδίδονται ασύρματα στο δίκτυο. Η ρύθμιση των κανόνων ασφαλείας και ο έλεγχος ταυτότητας για έναν μεγάλο αριθμό συσκευών IoT που χρησιμοποιούν Wi-Fi ως επιλογή συνδεσιμότητας αποτελεί δύσκολη διαδικασία.

5.2 Ethernet

Το πρωτόκολλο Ethernet είναι ένα πολύ γνωστό και ευρέως προσαρμοσμένο πρωτόκολλο Διαδικτύου που χρησιμοποιείται για τη μετάδοση δεδομένων σε υψηλή ταχύτητα στο LAN. Αυτό το πρωτόκολλο καλύπτει το φυσικό επίπεδο και το επίπεδο σύνδεσης δεδομένων του μοντέλου OSI. Υπάρχουν διάφορα είδη προτύπων Ethernet και τα νέα πρότυπα παρέχουν μεταφορά δεδομένων με ρυθμό 10 Gbps. Το φυσικό επίπεδο αποτελείται από καλώδια και συσκευές. Υπάρχουν διαφορετικά καλώδια Ethernet όπως ομοαξονικά καλώδια (παλαιότερη τεχνολογία), καλώδια συνεστραμμένου ζεύγους και καλώδια οπτικών ινών. Τα καλώδια που χρησιμοποιούνται περισσότερο είναι τα καλώδια συνεστραμμένου ζεύγους. Αυτά έχουν διαφορετικές κατηγορίες, όπως CAT 5/5e (ρυθμός μεταφοράς δεδομένων 100 Mbps), CAT 6 (έως 1 Gbps), CAT 6a (έως 10 Gbps) και CAT 7 (έως 10 Gbps). Σε κάθε άκρο των στριμμένων καλωδίων, υπάρχουν υποδοχές Rj-45 8 ακίδων που είναι υπεύθυνες για την αποστολή ή τη λήψη δεδομένων είτε σε λειτουργία ημιαμφίδρομης είτε σε πλήρη αμφίδρομη λειτουργία.

Μερικά από τα οφέλη χρήσης ενός ενσύρματου δικτύου Ethernet είναι:

- Είναι προσαρμοσμένο παγκοσμίως και είναι ένα εντελώς ανοιχτό και δωρεάν πρότυπο.
- Παρέχει πολύ μεγαλύτερη ταχύτητα σε σύγκριση με τα ασύρματα δίκτυα.
- Δεν υπάρχουν παρεμβολές προσφέροντας μεγαλύτερη σε σύγκριση με ένα ασύρματο δίκτυο.

- Δεν απαιτείται όνομα χρήστη/κωδικός πρόσβασης σε αντίθεση με το Wi-Fi ή τη σύζευξη στην περίπτωση του Bluetooth. Με το Ethernet, μετά την σύνδεση η συσκευή είναι σε θέση να αποστείλει άμεσα δεδομένα.
- Δεν απαιτεί ακριβό εξοπλισμό.

Το κύριο μειονέκτημα ενός ενσύρματου δικτύου έναντι ενός ασύρματου, είναι η κινητικότητα των χρηστών και η δυσκολία στην αλλαγή θέσης δικτυακών συσκευών. Στις IoT εφαρμογές συνήθως χρησιμοποιείται η ασύρματες τεχνολογίες δικτύωσης, αλλά αυτό δεν σημαίνει ότι το Ethernet δεν αποτελεί κατάλληλη επιλογή όταν οι συνθήκες υλοποίησης το επιτρέπουν.

5.3 LoRa/LoRaWAN

Το LoRa είναι μια τεχνολογία διαμόρφωσης ραδιοσυχνοτήτων για δίκτυα χαμηλής κατανάλωσης, ευρείας περιοχής (LPWAN). Δημιουργήθηκε από τη Semtech για την τυποποίηση των LPWAN και αναφέρεται στις ασύρματες ζεύξεις εξαιρετικά μεγάλης εμβέλειας που επιτρέπει αυτή η τεχνολογία. Η τεχνολογία αυτή παρέχει επικοινωνίες μεγάλης εμβέλειας, μια πύλη που βασίζεται στο LoRa μπορεί να λαμβάνει και να μεταδίδει σήματα σε απόσταση μεγαλύτερη από 15 χιλιόμετρα σε αγροτικές περιοχές. Ακόμη και σε πυκνά αστικά περιβάλλοντα, τα μηνύματα μπορούν να ταξιδέψουν έως και 5 χιλιόμετρα, ανάλογα με το πόσο βαθιά βρίσκονται σε εσωτερικούς χώρους οι τελικές συσκευές (τελικοί κόμβοι).

Ένα βασικό χαρακτηριστικό του LoRa είναι οι εξαιρετικά χαμηλές απαιτήσεις ενέργειας, οι οποίες επιτρέπουν τη δημιουργία συσκευών που λειτουργούν με μπαταρία και μπορούν να διαρκέσουν έως και 10 χρόνια[21]. Όσον αφορά τη διάρκεια ζωής της μπαταρίας, η ενέργεια που απαιτείται για τη μετάδοση ενός πακέτου δεδομένων είναι ελάχιστη, δεδομένου ότι τα πακέτα είναι πολύ μικρά και μεταδίδονται μόνο μερικές φορές την ημέρα. Επιπλέον, όταν οι τελικές συσκευές μεταβαίνουν σε κατάσταση αναστολής λειτουργίας, η κατανάλωση ενέργειας είναι της τάξης των milliwatts (mW), επιτρέποντας στην μπαταρία μιας συσκευής να διαρκέσει για πολλά, πολλά χρόνια.

Όσον αφορά τη χωρητικότητα, ένα δίκτυο LoRaWAN μπορεί να υποστηρίξει εκατομμύρια μηνύματα. Ο αριθμός των μηνυμάτων που υποστηρίζονται, εξαρτάται από τον αριθμό των πυλών (gateways) που είναι εγκατεστημένες. Μια πύλη οκτώ καναλιών μπορεί να υποστηρίξει μερικές εκατοντάδες χιλιάδες μηνύματα κατά τη διάρκεια μιας 24ωρης περιόδου. Εάν κάθε τελική συσκευή στέλνει 10 μηνύματα την ημέρα, μια τέτοια πύλη μπορεί να υποστηρίξει περίπου 10.000 συσκευές. Εάν το δίκτυο περιλαμβάνει 10 τέτοιες πύλες, το δίκτυο μπορεί να υποστηρίξει περίπου 100.000 συσκευές και ένα εκατομμύριο μηνύματα. Εάν κατά τη φάση της υλοποίησης ενός έργου απαιτείται περισσότερη χωρητικότητα, το μόνο που χρειάζεται είναι να προστεθούν επιπλέον πύλες στο δίκτυο. Δεν υπάρχει σχέση ένας προς έναν μεταξύ IoT συσκευών και πυλών που βασίζονται στο LoRa σε ένα δίκτυο LoRaWAN. Τα μηνύματα που αποστέλλονται από και προς τις τελικές συσκευές, διασχίζουν όλες τις πύλες που βρίσκονται εντός εμβέλειας. Η κατάργηση διπλότυπων χειρίζεται σε επόμενο στάδιο από το διακομιστή δικτύου.

5.3.1 Φυσικό επίπεδο LoRa

Το LoRa αντιπροσωπεύει το φυσικό επίπεδο ενός δικτύου LoRaWAN. Διαχειρίζεται τη διαμόρφωση, την τροφοδοσία, τους πομποδέκτες και το αναλογικό σήμα με τέτοιο τρόπο ώστε να

ικανοποιεί τις απαιτήσεις του επόμενου σταδίου επεξεργασίας. Η αρχιτεκτονική βασίζεται στις ακόλουθες ζώνες συχνοτήτων που δεν απαιτούν αδειδότηση (ISM):

- 915 MHz: Στις ΗΠΑ με όρια ισχύος αλλά χωρίς όριο κύκλου λειτουργίας.
- 868 MHz: Στην Ευρώπη με κύκλο λειτουργίας 1% και 10%.
- 433 MHz: Στην Ασία.

Ένα παράγωγο του φάσματος διασποράς chirp (CSS) είναι η τεχνική διαμόρφωσης που χρησιμοποιείται στο LoRa. Το CSS εξισορροπεί τον ρυθμό δεδομένων με την ευαισθησία σε ένα σταθερό εύρος ζώνης καναλιού. Το CSS χρησιμοποιήθηκε για πρώτη φορά τη δεκαετία του 1940 για στρατιωτική επικοινωνία μεγάλης εμβέλειας (radar) χρησιμοποιώντας διαμορφωμένους παλμούς chirp για την κωδικοποίηση δεδομένων και βρέθηκε ότι είναι ιδιαίτερα ανθεκτικός σε παρεμβολές, φαινόμενα Doppler και πολλαπλές διαδρομές. Τα chirps είναι ημιτονοειδή κύματα που αυξάνονται ή μειώνονται με την πάροδο του χρόνου. Δεδομένου ότι χρησιμοποιούν ολόκληρο το κανάλι για επικοινωνία, είναι σχετικά ισχυρά όσον αφορά τις παρεμβολές. Ο ρυθμός bit που χρησιμοποιεί το LoRa είναι συνάρτηση του ρυθμού chirp και του ρυθμού συμβόλων. Το bitrate αντιπροσωπεύεται από το R_b , ο συντελεστής διασποράς με S , το εύρος ζώνης με B όπως παρουσιάζονται στη σχέση (1). Επομένως, το bitrate (bps) μπορεί να κυμαίνεται από 0,3 kbps έως 5 kbps.

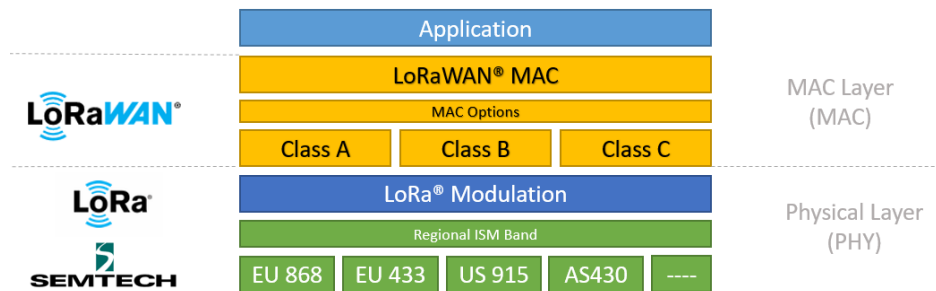
$$R_b = S \times \frac{1}{\left\lfloor \frac{2S}{B} \right\rfloor} \quad (1)$$

Αυτή η μορφή διαμόρφωσης επιτρέπει χαμηλή ισχύ σε μεγάλες αποστάσεις, όπως διαπίστωσε και σε στρατιωτικές εφαρμογές. Τα δεδομένα κωδικοποιούνται χρησιμοποιώντας τον αυξανόμενο ή μειούμενο ρυθμό συχνότητας και μπορούν να αποσταλούν πολλαπλές μεταδόσεις με διαφορετικούς ρυθμούς δεδομένων στην ίδια συχνότητα. Το CSS επιτρέπει τη λήψη σημάτων στα 19,4 dB κάτω από το επίπεδο θορύβου. Το εύρος ζώνης υποδιαιρείται σε πολλαπλές υποζώνες. Το LoRa χρησιμοποιεί κανάλια των 125 kHz και αφιερώνει έξι κανάλια 125 kHz και ψευδοτυχαία μεταπήδηση καναλιών. Ένα πλαίσιο θα μεταδοθεί με συγκεκριμένο συντελεστή διασποράς. Όσο υψηλότερος είναι ο συντελεστής διασποράς, τόσο πιο αργή είναι η μετάδοση αλλά τόσο μεγαλύτερη η εμβέλεια μετάδοσης. Τα πλαίσια στο LoRa είναι ορθογώνια, με αποτέλεσμα να μπορούν να αποσταλούν πολλαπλά ταυτόχρονα, εφόσον το καθένα αποστέλλεται με διαφορετικό συντελεστή διασποράς. Συνολικά υπάρχουν έξι διαφορετικοί συντελεστές διασποράς ($SF = 7$ έως $SF = 12$). Ένα τυπικό πακέτο LoRa απαρτίζεται από τον πρόλογο (preamble), την κεφαλίδα και ωφέλιμο φορτίο μεγέθους 51 έως 222 byte.

Τα δίκτυα LoRa διαθέτουν ένα ισχυρό χαρακτηριστικό που ονομάζεται Προσαρμοστικός Ρυθμός Δεδομένων (ADR) που επιτρέπει τη δυναμικά κλιμακούμενη χωρητικότητα με βάση την πυκνότητα των κόμβων και την υποδομή. Το ADR ελέγχεται από τη διαχείριση του δικτύου στο νέφος. Οι κόμβοι που βρίσκονται κοντά σε ένα σταθμό βάσης μπορούν να ρυθμιστούν σε υψηλότερο ρυθμό δεδομένων λόγω της ποιότητας του σήματος. Αυτοί οι κόμβοι που βρίσκονται σε κοντινή απόσταση μπορούν να μεταδώσουν δεδομένα, να απελευθερώσουν το εύρος ζώνης τους και να εισέλθουν σε κατάσταση αδράνειας γρήγορα έναντι των απομακρυσμένων κόμβων που εκπέμπουν με χαμηλότερους ρυθμούς.

5.3.2 LoRaWAN επίπεδο MAC

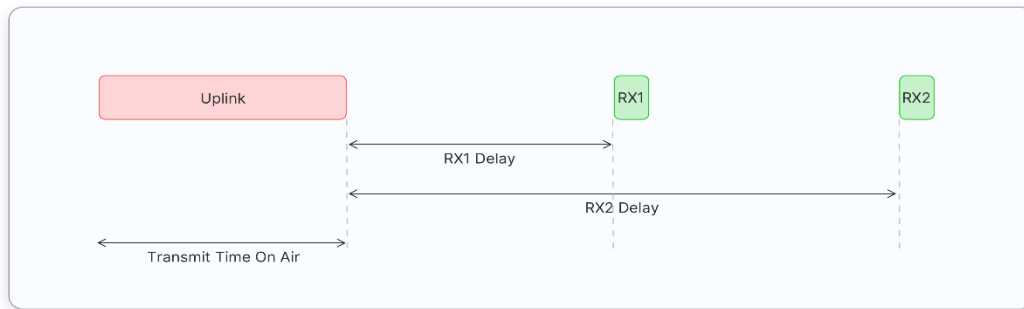
Το LoRaWAN είναι ένα πρωτόκολλο επιπέδου ελέγχου πρόσβασης μέσων (MAC) χτισμένο πάνω από τη διαμόρφωση LoRa PHY. Είναι ένα επίπεδο λογισμικού που καθορίζει τον τρόπο με τον οποίο οι συσκευές χρησιμοποιούν το υλικό LoRa, για παράδειγμα όταν μεταδίδουν, και τη μορφή των μηνυμάτων (Σχήμα 13).



Σχήμα 19: Στοιβά πρωτοκόλλου LoRa και LoRaWAN.

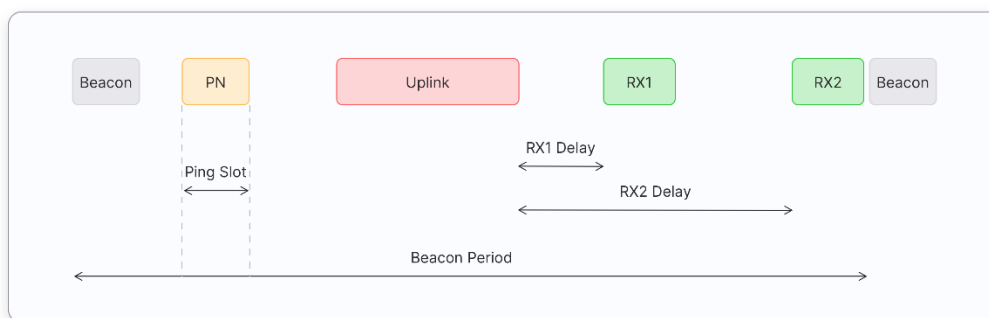
Το LoRaWAN αντιπροσωπεύει το MAC που βρίσκεται πάνω από ένα LoRa PHY. Το LoRaWAN MAC είναι ένα ανοιχτό πρωτόκολλο, ενώ το PHY είναι κλειστό. Η τεχνολογία LoRaWAN MAC ελέγχει τους πόρους μετάδοσης προκειμένου να ανταποκρίνεται στα χαρακτηριστικά βελτιστοποίησης του ρυθμού μετάδοσης δεδομένων, της χρήσης πόρων μετάδοσης και ενέργειας. Αυτοί οι πόροι είναι το εύρος ζώνης που πρέπει να καταληφθεί, ο παράγοντας φάσματος διασποράς που θα χρησιμοποιηθεί και η ισχύς μετάδοσης για κάθε κόμβο [22]. Υπάρχουν τρία πρωτόκολλα MAC που αποτελούν μέρος του επιπέδου σύνδεσης δεδομένων. Αυτά τα τρία πρωτόκολλα εξισορροπούν την καθυστέρηση με τη χρήση ενέργειας. Η κλάση-A είναι η καλύτερη για τον μετριασμό της ενέργειας ενώ έχει την υψηλότερη καθυστέρηση. Η κλάση-B είναι μεταξύ κλάσης-A και κλάσης-C. Η κλάση-C έχει την ελάχιστη καθυστέρηση αλλά την υψηλότερη κατανάλωση ενέργειας.

Οι συσκευές που ανήκουν στην κλάση-A είναι αισθητήρες και τερματικοί σταθμοί που τροφοδοτούνται από μπαταρία. Όλες οι τελικές συσκευές LoRaWAN πρέπει να υποστηρίζουν την εφαρμογή κλάσης A. Η επικοινωνία κλάσης A ξεκινά πάντα από την τελική συσκευή. Μια συσκευή μπορεί να στείλει ένα μήνυμα ανερχόμενης ζεύξης ανά πάσα στιγμή. Μόλις ολοκληρωθεί η μετάδοση ανοδικής ζεύξης, η συσκευή ανοίγει δύο σύντομα παράθυρα λήψης κατερχόμενης ζεύξης. Υπάρχει μια καθυστέρηση μεταξύ του τέλους της μετάδοσης ανερχόμενης ζεύξης και της έναρξης των παραθύρων λήψης RX1 και RX2 αντίστοιχα. Εάν ο διακομιστής δικτύου δεν ανταποκρίνεται κατά τη διάρκεια αυτών των δύο παραθύρων λήψης, η επόμενη κατερχόμενη ζεύξη θα είναι μετά την επόμενη μετάδοση ανοδικής ζεύξης (Σχήμα 20). Ο διακομιστής μπορεί να ανταποκριθεί κατά τη διάρκεια του πρώτου παραθύρου λήψης (RX1) ή κατά τη διάρκεια του δεύτερου παραθύρου λήψης (RX2), αλλά δεν χρησιμοποιεί και τα δύο παράθυρα.



Σχήμα 20: Παράθυρα λήψης κλάσης A.

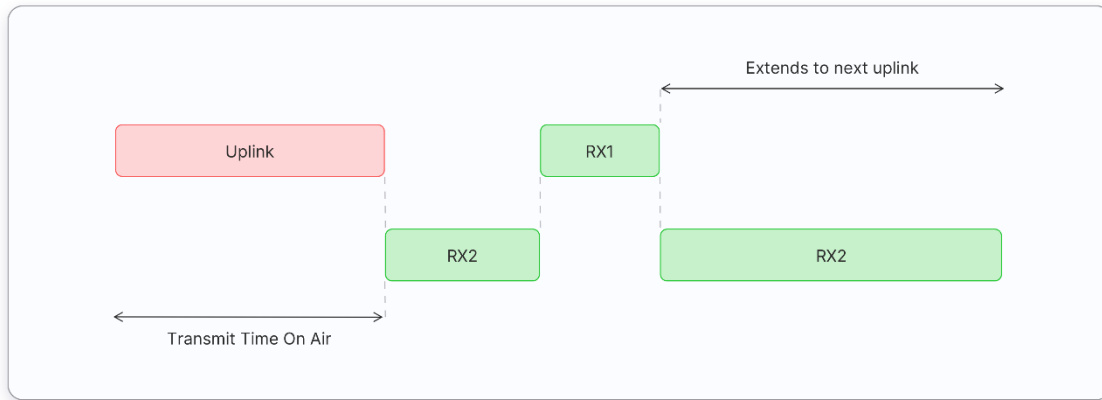
Εκτός από τα παράθυρα λήψης που εκκινήθηκαν από την κλάση A, οι συσκευές της κλάσης B ανοίγουν παράθυρα προγραμματισμένης λήψης για τη λήψη μηνυμάτων κατερχόμενης ζεύξης από τον διακομιστή δικτύου. Χρησιμοποιώντας χρονικά συγχρονισμένα beacon που μεταδίδονται από την πύλη, οι συσκευές ανοίγουν περιοδικά παράθυρα λήψης. Ο χρόνος μεταξύ δύο beacon είναι γνωστός ως περίοδος. Η συσκευή ανοίγει «υποδοχές ring» κατερχόμενης ζεύξης σε προγραμματισμένες ώρες για τη λήψη μηνυμάτων από τον διακομιστή δικτύου. Οι συσκευές της κλάσης B ανοίγουν επίσης παράθυρα λήψης μετά την αποστολή μιας ανερχόμενης ζεύξης (Σχήμα 21).



Σχήμα 21: Παράθυρα λήψης κλάσης B.

Οι τελικές συσκευές της κλάσης B έχουν μικρότερο χρόνο καθυστέρησης από τις τελικές συσκευές της κλάσης A, επειδή είναι προσβάσιμες σε προ ρυθμισμένους χρόνους και δεν χρειάζεται να στείλουν ανερχόμενη ζεύξη για να λάβουν μια κατερχόμενη ζεύξη. Η διάρκεια ζωής της μπαταρίας είναι μικρότερη στην κλάση B από την κλάση A, επειδή η συσκευή ξοδεύει περισσότερο χρόνο σε ενεργή λειτουργία, κατά τη διάρκεια των beacons και των υποδοχών ring.

Οι συσκευές κλάσης C επεκτείνουν την κλάση A και δεν μπορούν να εφαρμόσουν την κλάση B. Διατηρούν τα παράθυρα λήψης ανοιχτά εκτός εάν εκπέμπουν, όπως φαίνεται στο Σχήμα 6. Αυτό επιτρέπει επικοινωνία χαμηλής καθυστέρησης, αλλά πολλές φορές οι συσκευές αυτής της κλάσης είναι περισσότερο ενεργοβόρες από αυτές της κλάσης A. Οι συσκευές κλάσης C μπορεί να είναι ενεργοποιητές ή συσκευές plug-in.



Σχήμα 22: Παράθυρα λήψης κλάσης C.

Κάθε τερματική συσκευή πρέπει να είναι εγγεγραμμένη σε ένα δίκτυο πριν από την αποστολή και λήψη μηνυμάτων. Αυτή η διαδικασία είναι γνωστή ως ενεργοποίηση. Υπάρχουν δύο διαθέσιμες μέθοδοι ενεργοποίησης:

- **Over-The-Air-Activation (OTAA):** Το OTAA αποτελεί την πιο ασφαλής και συνιστώμενη μέθοδος ενεργοποίησης για τελικές συσκευές. Οι συσκευές εκτελούν μια διαδικασία σύνδεσης με το δίκτυο, κατά την οποία εκχωρείται μια δυναμική διεύθυνση συσκευής και διαπραγματεύονται τα κλειδιά ασφαλείας με τη συσκευή.
- **Ενεργοποίηση μέσω εξατομίκευσης (ABP):** Το ABP απαιτεί ενσωματωμένη κωδικοποίηση της διεύθυνσης της συσκευής καθώς και των κλειδιών ασφαλείας στη συσκευή. Το ABP είναι λιγότερο ασφαλές από το OTAA και έχει επίσης το μειονέκτημα ότι οι συσκευές δεν μπορούν να αλλάξουν παρόχους δικτύου χωρίς χειροκίνητη αλλαγή των κλειδιών στη συσκευή.

Το LoRaWAN είναι ένα ασύγχρονο πρωτόκολλο που βασίζεται στο ALOHA. Το πρωτόκολλο ALOHA επινοήθηκε αρχικά στο Πανεπιστήμιο της Χαβάης το 1968 ως μια μορφή επικοινωνίας πολλαπλής πρόσβασης προτού υπάρξουν τεχνολογίες όπως η CSMA. Στο ALOHA, οι πελάτες μπορούν να μεταδίδουν μηνύματα χωρίς να γνωρίζουν εάν άλλοι πελάτες βρίσκονται σε διαδικασία μετάδοσης ταυτόχρονα. Δεν υπάρχουν κρατήσεις ή τεχνικές πολυπλεξίας. Η βασική αρχή είναι ότι ο διανομέας (ή η πύλη στην περίπτωση του LoRaWAN) αναμεταδίδει αμέσως τα πακέτα που έχει λάβει. Εάν ένα τελικό σημείο παρατηρήσει ότι ένα από τα πακέτα του δεν επιβεβαιώθηκε, θα περιμένει και στη συνέχεια θα επαναμεταδώσει το πακέτο. Στο LoRaWAN, οι συγκρούσεις συμβαίνουν μόνο εάν οι εκπομπές χρησιμοποιούν τα ίδια κανάλια και τον ίδιο παράγοντα διασποράς.

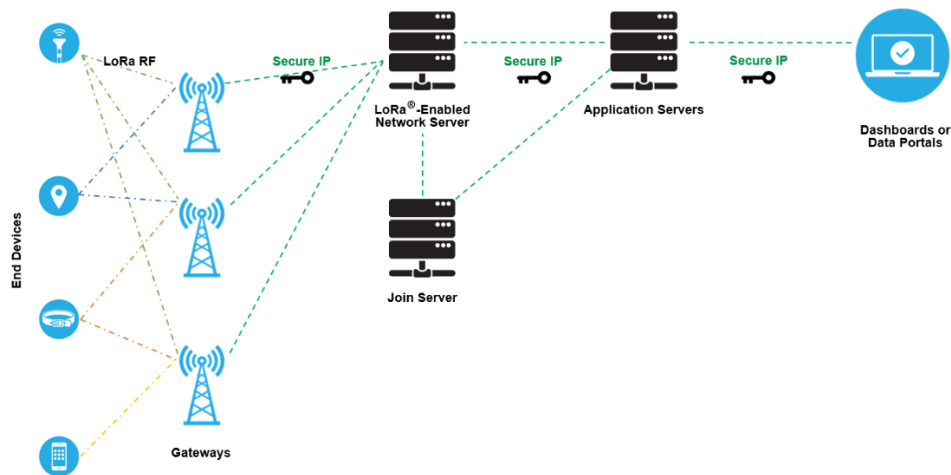
5.3.3 Τοπολογία LoRaWAN

Το LoRaWAN βασίζεται σε μια τοπολογία δικτύου αστέρα (Σχήμα 23). Η επικοινωνία LoRa βασίζεται στη μετάδοση και όχι στη δημιουργία μιας σχέσης μεταξύ ομότιμων τερματικών συσκευών. Το LoRaWAN δίκτυο μπορεί να διαθέτει πολλαπλές πύλες, προσφέροντας μεγάλη εμβέλεια δικτύωσης. Ένας πάροχος υπηρεσιών νέφους μπορεί να λάβει διπλά μηνύματα από πολλαπλές πύλες και είναι υπεύθυνος για τη διαχείριση και την αντιμετώπιση των διπλών εκπομπών. Ένα χαρακτηριστικό στοιχείο του LoRaWAN είναι ότι τα δεδομένα των χρηστών μεταφέρονται από έναν τελικό κόμβο στην πύλη μέσω του πρωτοκόλλου LoRaWAN. Σε εκείνο το σημείο, η πύλη LoRaWAN θα προωθήσει το πακέτο μέσω οποιουδήποτε backhaul (όπως 4G LTE,

Ethernet ή Wi-Fi) σε μια αποκλειστική υπηρεσία δικτύου LoRaWAN στο νέφος. Σε αντίθεση με το LoRaWAN, οι περισσότερες αρχιτεκτονικές WAN αποδεσμεύουν οποιονδήποτε έλεγχο των δεδομένων τη στιγμή που αφήνουν το δίκτυό τους σε έναν προορισμό στο Διαδίκτυο.

Η υπηρεσία δικτύου έχει τους κανόνες και τη λογική για να εκτελέσει τα απαραίτητα ανώτερα επίπεδα της στοίβας δικτύου. Μια παρενέργεια αυτής της αρχιτεκτονικής είναι ότι η μεταβίβαση από τη μια πύλη στην άλλη δεν είναι απαραίτητη, όπως είναι στην επικοινωνία LTE. Εάν ένας κόμβος είναι κινητός και μετακινείται από κεραία σε κεραία, οι υπηρεσίες δικτύου θα συλλάβουν πολλαπλά πανομοιότυπα πακέτα από διαφορετικές διαδρομές. Αυτές οι υπηρεσίες δικτύου επιτρέπουν στα συστήματα LoRaWAN να επιλέγουν την καλύτερη διαδρομή και πηγή πληροφοριών όταν ένας τερματικός κόμβος συσχετίζεται με περισσότερες από μία πύλες. Οι αρμοδιότητες των υπηρεσιών δικτύου περιλαμβάνουν:

- Υπηρεσίες ασφαλείας διπλότυπης αναγνώρισης και τερματισμού πακέτων.
- Δρομολόγηση κατερχόμενης ζεύξης.
- Μηνύματα επιβεβαίωσης.



Σχήμα 23: Τυπική υλοποίηση δικτύου LoRaWAN.

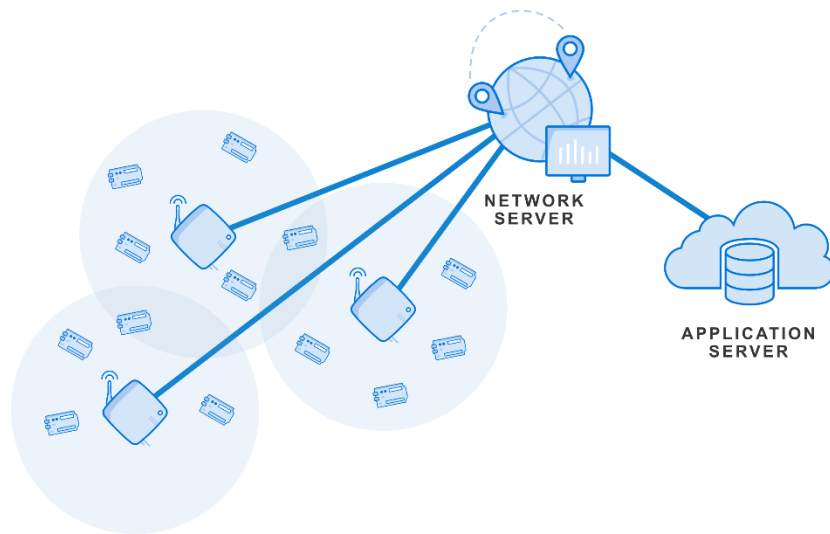
5.3.4 The Things Network

Το The Things Network, γνωστό και ως TTN, αποτελεί μια υποδομή ανοιχτού κώδικα που στοχεύει στην παροχή δωρεάν κάλυψης δικτύου LoRaWAN. Το TTN υλοποιείται από μια αναπτυσσόμενη κοινότητα σε όλο τον κόσμο και βασίζεται σε εθελοντικές συνεισφορές των μελών της. Στον ιστότοπο του TTN παρουσιάζεται πλήθος διαφορετικών οδηγών, που επιτρέπουν στους χρήστες να εγκαταστήσουν δικτυακές πύλες μεγάλης εμβέλειας με LoRa στην πόλη τους, για την ανάπτυξη του δικτύου. Χάρη στην ανάπτυξη του πηγαίου κώδικα και στην υποδομή, η κάλυψή τους είναι ήδη αρκετά καλή στις μεγάλες πόλεις και εξαπλώνεται σε μικρότερες. Στην Ελλάδα τα τελευταία χρόνια αναπτύχθηκαν αρκετά δίκτυα LoRaWAN σε διάφορες περιοχές όπως στην Κομοτηνή, Καβάλα, Θεσσαλονίκη, Κοζάνη, Βόλος, Πάτρα, Αθήνα, Σύρος, Χανιά, Ηράκλειο [23].

Για να χρησιμοποιήσει κάποιος χρήστης το δίκτυο TTN, πρέπει πρώτα να δημιουργήσει έναν λογαριασμό. Η εγγραφή είναι δωρεάν καθώς υιοθετείται πλήρως η φιλοσοφία της κοινότητας να

παρέχει ένα εντελώς δωρεάν και ανοιχτό δίκτυο. Με τη δημιουργία του λογαριασμού, ο χρήστης μπορεί να δημιουργήσει διάφορες εφαρμογές σύμφωνα με της ανάγκες του. Μια εφαρμογή TTN δίνει πρόσβαση στις ακόλουθες παραμέτρους:

- Το αναγνωριστικό της εφαρμογής (Επιτρέπει την αναγνώριση των κόμβων-συσκευών στο δίκτυο).
- Τις καταχωρημένες συσκευές (τους αισθητήρες, την κατάστασή τους και τα δεδομένα που αποστέλλουν).
- Αποκωδικοποιητής ωφέλιμου φορτίου (Σύνταξη κώδικα που χρησιμοποιείται για την αποκωδικοποίηση των εισερχόμενων μηνυμάτων).
- Γενικές ρυθμίσεις της εφαρμογής (π.χ. ασφάλεια, κλειδιά δικτύου, μέθοδος αυθεντικοποίησης).



Σχήμα 24: Αρχιτεκτονική TTN.

5.4 Πρωτόκολλα Επιπέδου Εφαρμογής

Υπάρχουν πολλά πρωτόκολλα επικοινωνίας που μπορούν να χρησιμοποιηθούν για την αποστολή δεδομένων από τις συσκευές IoT. Αυτές οι συσκευές στέλνουν δεδομένα στο cloud χρησιμοποιώντας διαφορετικούς τύπους τεχνολογιών επικοινωνίας. Ορισμένες από αυτές τις τεχνολογίες βασίζονται στην IP διευθυνσιοδότηση των συσκευών, όπως το Ethernet ή το Wi-Fi, ενώ άλλες όχι, όπως το BLE ή το LoRa. Με τη χρήση IP διευθυνσιοδότησης τα δεδομένα μπορούν να αποσταλούν απευθείας στο διαδίκτυο ενώ σε μερικές περιπτώσεις συνίσταται η χρήσης κάποιας ενδιάμεσης συσκευής η οποία θα έχει το ρόλο της πύλης. Η τελευταία διαχειρίζεται όλα τα δεδομένα που αποστέλλονται από τις τοπικές συσκευές και αποστέλλει μόνο σχετικές πληροφορίες στο νέφος.

Στις περιπτώσεις που χρησιμοποιούνται τεχνολογίες που δεν υποστηρίζουν εγγενώς την IP διευθυνσιοδότηση, η χρήση της πύλης είναι επιβεβλημένη όπως για παράδειγμα στο LoRaWAN. Η πύλη θα προωθήσει όλα τα πλαίσια που λαμβάνονται από τους κόμβους LoRaWAN στον διακομιστή δικτύου και στη συνέχεια, ο διακομιστής δικτύου θα επεξεργαστεί τα πακέτα, δρομολογώντας ή απορρίπτοντάς τα.

Το τελευταίο βήμα της διαδρομής των δεδομένων IoT είναι η πλατφόρμα IoT. Υπάρχουν διάφοροι τύποι πλατφορμών IoT, ανάλογα με τις λειτουργίες τους. Ορισμένες πλατφόρμες προσφέρουν επικοινωνία και διαχείριση συσκευών, ενώ άλλες χρησιμοποιούν δεδομένα που συλλέγονται για την εκτέλεση αναλύσεων. Στο κεφάλαιο αυτό αναλύονται τα πρωτόκολλα που χρησιμοποιήθηκαν κατά την φάση της υλοποίησης.

5.4.1 HTTP

Το πρωτόκολλο HTTP είναι ένα από τα βασικά πρωτόκολλα του Διαδικτύου και ένα από τα βασικά πρωτόκολλα που χρησιμοποιούνται για την επικοινωνία των συσκευών με τις πλατφόρμες IoT. Το HTTP λειτουργεί σε μια αρχιτεκτονική πελάτη-διακομιστή μέσω συνδέσεων TCP. Από προεπιλογή, το HTTP χρησιμοποιεί τη θύρα 80 για την ακρόαση των αιτημάτων, αλλά μπορούν να διαμορφωθούν και άλλες θύρες TCP για την εκτέλεση του διακομιστή ιστού.

Τα τρία βασικά χαρακτηριστικά του πρωτοκόλλου HTTP είναι:

- Όταν ένα πρόγραμμα περιήγησης ξεκινά ένα αίτημα HTTP, αναμένεται μια απάντηση από τον διακομιστή HTTP. Αφού ο διακομιστής επεξεργαστεί το αίτημα, αποστέλλει μια απάντηση στον πελάτη. Στη συνέχεια, η σύνδεση τερματίζεται και όλα τα δεδομένα περιόδου λειτουργίας εξαφανίζονται. Την επόμενη φορά που θα συνδεθούν ο πελάτης και ο διακομιστής, η διαδικασία θα ξεκινήσει από την αρχή.
- Μετά το τέλος μιας σύνδεσης, ο διακομιστής και ο πελάτης απορρίπτουν όλα τα σχετικά δεδομένα. Έτσι, κανένας από αυτούς δεν κρατά πληροφορίες από προηγούμενες συνδέσεις.
- Διαφορετικοί τύποι δεδομένων μπορούν να μεταφερθούν μεταξύ του υπολογιστή-πελάτη και του διακομιστή, από κείμενο μέχρι ψηφιακά μέσα ή ακόμα και δυαδικά αρχεία.

Οι μέθοδοι που εφαρμόζονται στο HTTP είναι οι ακόλουθοι:

- **GET:** Ζητάει να λάβει ένα πόρο από ένα καθορισμένο URI.

- **POST:** Στέλνει τα δεδομένα του σώματος στον διακομιστή. Μπορεί να είναι κείμενο, αρχείο κ.α..
- **HEAD:** Είναι η ίδια μέθοδος με την GET, αλλά μεταφέρει μόνο το τμήμα της κεφαλίδας και την κατάσταση.
- **PUT:** Αυτή η μέθοδος αντικαθιστά τα δεδομένα στο διακομιστή (στο συγκεκριμένο URI) με τα δεδομένα που αποστέλλονται στο σώμα.
- **DELETE:** Καταργεί όλα τα τρέχοντα δεδομένα στον πόρο που προσδιορίζεται από το URI.
- **CONNECT:** Δημιουργεί μια σύνδεση με το διακομιστή για ένα συγκεκριμένο URI.
- **OPTIONS:** Περιγράφει τις επιλογές σύνδεσης για το URI.
- **TRACE:** Ζητάει το loopback του αιτήματος, για δοκιμή.

Το HTTPS προσθέτει ένα επίπεδο ασφάλειας στο HTTPS χρησιμοποιώντας το πρωτόκολλο TLS για την κρυπτογράφηση των δεδομένων. Τα κύρια χαρακτηριστικά του για τη διαφύλαξη της ασφάλειας των επικοινωνιών είναι ο έλεγχος ταυτότητας, το απόρρητο και η ακεραιότητα. Από προεπιλογή το πρωτόκολλο HTTPS χρησιμοποιεί την TCP θύρα 443, αλλά μπορεί να καθοριστεί μια άλλη θύρα στον διακομιστή ιστού.

Η ασφάλεια του HTTPS εξαρτάται εξ' ολοκλήρου από το πρωτόκολλο TLS, το οποίο χρησιμοποιεί δημόσια και ιδιωτικά κλειδιά για τη δημιουργία βραχυπρόθεσμων κλειδιών συνεδρίων επικοινωνίας. Χρησιμοποιεί επίσης πιστοποιητικά που μπορούν να ελέγξουν την ταυτότητα τόσο του διακομιστή όσο και του πελάτη. Τα κλειδιά και τα πιστοποιητικά θα πρέπει να είναι διαφορετικά για κάθε συσκευή IoT για την ύπαρξη ενός ικανοποιητικού επιπέδου ασφαλείας.

5.4.2 MQTT

Το MQTT είναι ένα πρότυπο πρωτόκολλο ανταλλαγής μηνυμάτων, δημοσίευσης/εγγραφής (publish/subscribe), εξαιρετικά απλό και ελαφρύ, που έχει σχεδιαστεί για συσκευές χαμηλής υπολογιστικής ισχύος, για δίκτυα με χαμηλό εύρος ζώνης (bandwidth) ή υψηλής καθυστέρησης. Οι αρχές σχεδιασμού αποσκοπούν στο να ελαχιστοποιήσουν το εύρος ζώνης του δικτύου και τις απαιτήσεις πόρων των συσκευών, ενώ προσπαθούν επίσης να διασφαλίσουν την αξιοπιστία και την διασφάλιση της παράδοσης των δεδομένων, κάνοντας χρήση υπηρεσίας ποιότητας QoS. Τα παραπάνω καθιστούν το πρωτόκολλο ιδανικό για υλοποιήσεις στο «Διαδίκτυο των Πραγμάτων» όπου το εύρος ζώνης και η κατανάλωση ενέργειας αποτελούν προτεραιότητα.

Το MQTT εφευρέθηκε από τους Dr.Andy Stanford-Clark της IBM και Arlen Nipper της Arcom το 1999. Ο στόχος τους ήταν να δημιουργηθεί ένα πρωτόκολλο για παρακολούθηση σωληνώσεων πετρελαίου μέσω δορυφορικής σύνδεσης το οποίο θα είχε ελάχιστη ενεργειακή απαίτηση και μικρό εύρος ζώνης (bandwidth) του δικτύου. Από το 2010 η έκδοση 3.1 διατέθηκε ελεύθερα στο κοινό ενώ τον Μάρτιο του 2019 η έκδοση 5.0 έγινε πρότυπο από οργανισμό OASIS. Από τότε αρκετές εταιρίες και οργανισμοί έχουν φτιάξει ποικίλες εφαρμογές στηριζόμενες σε αυτό, μια από τις μεγαλύτερες υλοποιήσεις αποτελεί και η εφαρμογή Messenger του Facebook [24].

Τα βασικά χαρακτηριστικά του πρωτοκόλλου είναι:

- Εύκολη υλοποίηση.
- Παροχή επιπέδων ποιότητας υπηρεσίας στην παράδοση των μηνυμάτων (QoS).
- Χαμηλές απαιτήσεις σε υπολογιστική ισχύ και εύρος ζώνης (bandwidth).
- Ανεξαρτησία του είδους των δεδομένων που αποστέλλονται.

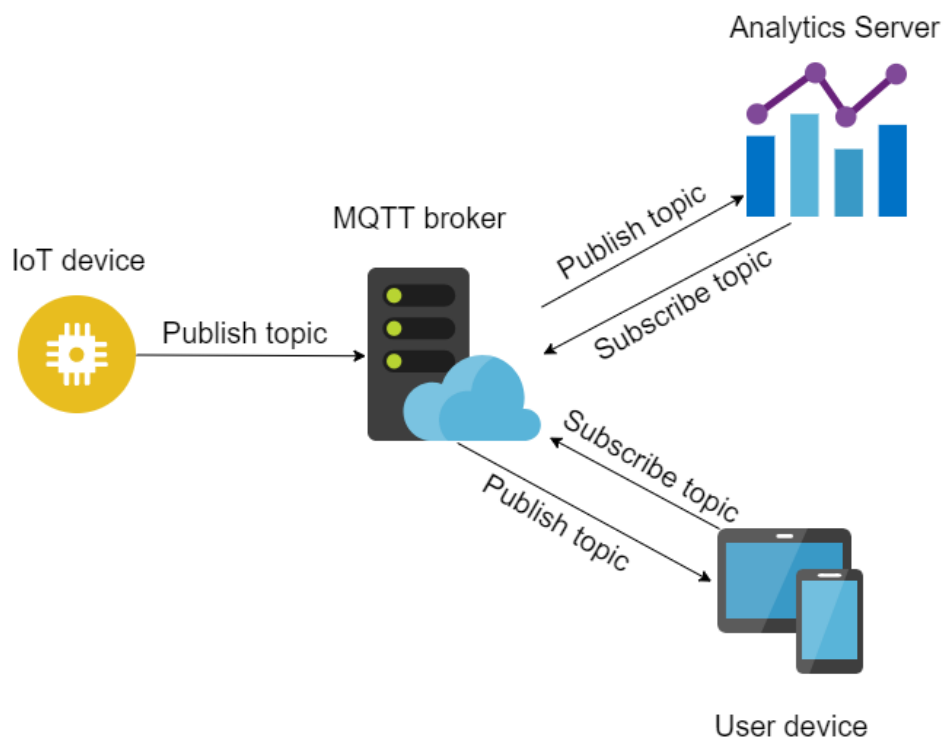
- Συνεχής επίγνωση της κατάστασης της σύνδεσης.

Στο πρωτόκολλο MQTT, χρησιμοποιείται το μοντέλο εκδότη/συνδρομητή (publish/subscribe) (Σχήμα 25). Ο εκδότης (publisher) δημοσιεύει μηνύματα σε συγκεκριμένα θέματα (Topics) τα οποία λαμβάνουν οι συνδρομητές που είναι εγγεγραμμένοι σε αυτά. Κάθε συνδρομητής εγγράφεται σε ένα ή περισσότερα θέματα και λαμβάνει τα μηνύματα μόνο αυτών. Ένα θέμα (Topic) μπορεί να αναφέρεται σε μια μεμονωμένη τιμή (συμβολοσειρά, αριθμό κ.λπ.) ή σε ένα αντικείμενο. Η γενική προσέγγιση χρησιμοποιεί αντικείμενα JSON για τη σύνθεση πληροφοριών σε ένα μόνο θέμα, όπως φαίνεται στο ακόλουθο παράδειγμα (Σχήμα 26):

Μεταξύ των εκδοτών και συνδρομητών υπάρχει μια ενδιάμεση οντότητα που ονομάζεται *Broker*. Η οντότητα αυτή εκτελεί τις ακόλουθες εργασίες:

- Λήψη όλων των μηνυμάτων και φιλτράρισμα αυτών.
- Επίγνωση όλων των συνδρομητών του συστήματος.
- Λήψη των μηνυμάτων από τους εκδότες και προώθησή τους στους ενδιαφερόμενους συνδρομητές.

Αυτή η δομή παρέχει ένα σημαντικό πλεονέκτημα στην υλοποίηση μιας εφαρμογής. Συγκεκριμένα, οι εκδότες και οι συνδρομητές απαιτείται να γνωρίζουν μόνο την ύπαρξη του Broker και όχι την μεταξύ τους «διεύθυνση» για να επιτευχθεί η επικοινωνία. Ο τύπος της επικοινωνία είναι ασύγχρονος, οι εκδότες και οι συνδρομητές δεν απαιτείται να είναι σε λειτουργία ταυτόχρονα ενώ δεν διακόπτουν την λειτουργία τους κατά την αποστολή ή παραλαβή των μηνυμάτων.

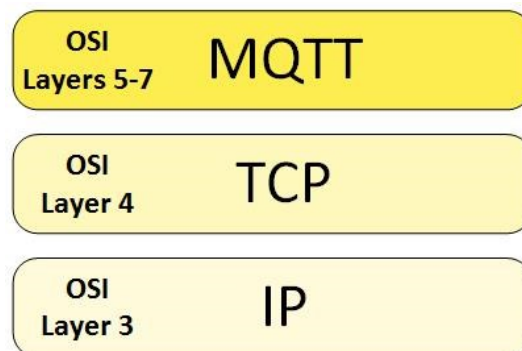


Σχήμα 25: Μοντέλο Publish/Subscribe - MQTT.

```
{ "sensorType": "Thermometer",  
  "sensorModel": "AM2302",  
  "temp": 25,  
  "hum": 40 }
```

Σχήμα 26: Αντικείμενο JSON.

Στο πρωτόκολλο MQTT με τον όρο client εννοούμε τόσο τους εκδότες (publishers) όσο και τους συνδρομητές (subscribers). Ένας client μπορεί να είναι ταυτόχρονα και εκδότης και συνδρομητής. Οι clients μπορεί να είναι οποιοσδήποτε τύπος συσκευής (microcontroller, smartphone, server κτλ.) οι οποίοι φέρουν μια υλοποίηση της βιβλιοθήκης του MQTT για να συνδεθούν με τον Broker [25]. Το MQTT σε επίπεδο επικοινωνίας διαδικτύου κάνει χρήση της στοίβας TCP/IP, ενώ υλοποιείται στο επίπεδο της εφαρμογής (Σχήμα 27). Η σύνδεση μεταξύ client και Broker γίνεται με την αποστολή ενός μηνύματος *CONNECT*. Όταν πραγματοποιηθεί η σύνδεση ο Broker απαντάει με ένα μήνυμα επιβεβαίωσης σύνδεσης *CONNACK*.



Σχήμα 27: MQTT (Επίπεδο Εφαρμογής) στη στοίβα TCP/IP.

5.4.2.1 Ποιότητα υπηρεσίας (QoS)

Στο πρωτόκολλο MQTT η ποιότητα υπηρεσίας (Quality of Service) αποτελείται από τρία επίπεδα τα οποία δηλώνουν το πόσο εγγυημένη είναι η παράδοση ενός μηνύματος μεταξύ αποστολέα και παραλήπτη. Το QoS ορίζεται κάθε φορά από τον εκάστοτε publisher ή subscriber. Κατά συνέπεια το QoS μπορεί να υποβιβαστεί για κάποιο παραλήπτη ο οποίος έκανε subscribe με μικρότερο επίπεδο QoS. Αναλυτικότερα, τα επίπεδα QoS του πρωτοκόλλου MQTT είναι:

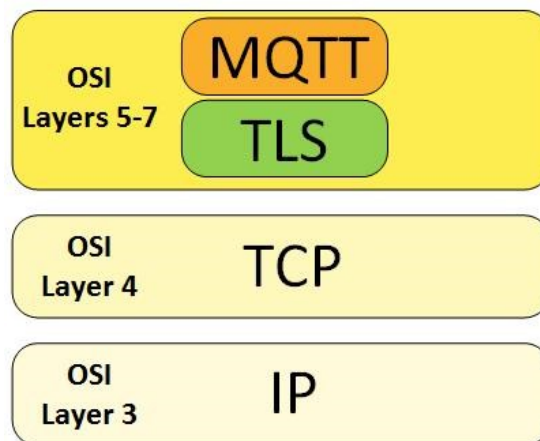
- **QoS 0:** Το μήνυμα αποστέλλεται το πολύ μια φορά, δεν λαμβάνεται επιβεβαίωση παράδοσης και δεν αποθηκεύεται. Το επίπεδο αυτό χρησιμοποιείται σε αξιόπιστες συνδέσεις μεταξύ clients και Broker ή σε περιπτώσεις που η διατήρηση της πληροφορίας δεν αποτελεί προτεραιότητα.
- **QoS 1:** Το μήνυμα αποστέλλεται τουλάχιστον μια φορά. Ο αποστολέας αναμένει απάντηση PUBACK από τον Broker, αν δεν την λάβει σε συγκεκριμένο χρονικό διάστημα προσπαθεί να αποστείλει το μήνυμα ξανά. Με αυτόν τον τρόπο μπορεί το μήνυμα να φτάσει στον αποδέκτη

περισσότερες φορές. Χρησιμοποιείται σε εφαρμογές όπου η παράδοση της πληροφορίας αποτελεί προτεραιότητα.

- **QoS 2:** Το μήνυμα θα φτάσει στον παραλήπτη μια μόνο φορά. Αν πραγματοποιηθούν περισσότερες αποστολές του μηνύματος από τον Client, ο Broker θα το προωθήσει μια μόνο φορά στον παραλήπτη. Χρησιμοποιείται σε εφαρμογές που η πληροφορία πρέπει να ληφθεί από τον παραλήπτη μια μόνο φορά.

Σε πολλές εφαρμογές του Διαδικτύου των Πραγμάτων, η ιδιωτικότητα και η διαφύλαξη του απορρήτου των δεδομένων αποτελεί ύψιστη προτεραιότητα. Συνήθως, τα πακέτα TCP που περιλαμβάνουν την πληροφορία, αποστέλλονται προς το νέφος μέσω της προώθησής τους από ενδιάμεσες υποδομές όπως Routers, Firewalls κτλ. Αυτό εγκυμονεί κινδύνους λόγω της πιθανής ανάγνωσης ή επεξεργασίας των δεδομένων από τα ενδιάμεσα σημεία.

Για τον λόγο αυτό χρησιμοποιείται το πρωτόκολλο κρυπτογραφίας TLS. Το πρωτόκολλο αυτό χρησιμοποιεί μηχανισμούς «χειραψίας» για να δημιουργηθεί μια ασφαλής σύνδεση μεταξύ Client και Broker. Η κρυπτογράφηση TLS πραγματοποιείται μεταξύ του επιπέδου εφαρμογής MQTT και του επιπέδου TCP, ενώ υλοποιείται στο επίπεδο της εφαρμογής (Σχήμα 28). Κάνοντας χρήση του πρωτοκόλλου TLS οι συνδέσεις στον Broker γίνονται στην θύρα 8883 και όχι στην 1883 όπως γίνεται με την χρήση μόνο του TCP/IP.



Σχήμα 28: MQTT/TLS (Επίπεδο Εφαρμογής) στη στοιβή TCP/IP.

5.4.2.2 MQTT 3.1.1

Η έκδοση MQTT 3.1.1 συναντάται και υλοποιείται σχεδόν από το σύνολο των IoT πλατφορμών. Τα κύρια χαρακτηριστικά του πρωτοκόλλου της έκδοσης 3.1.1 [26] είναι:

- Υποστηρίζεται η αποστολή και λήψη μηνυμάτων μέσω WebSocket.
- Τα αναγνωριστικά πελάτη επεκτάθηκαν με μέγεθος 65.535 bytes.
- Δεν χρειάζεται η αναμονή για λήψη απάντησης CONNACK από τον Broker. Οι συσκευές μπορούν να συνδέονται, να δημοσιεύουν και να αποσυνδέονται.
- Το αναγνωριστικό πελάτη μπορεί να οριστεί σε μήκος 0 byte. Αυτό χρησιμοποιείται σε ανώνυμες υλοποιήσεις πελατών.

- Αναφορά σφάλματος στο MQTT SUBACK για την ειδοποίηση των πελατών σχετικά με τις απαγορευμένες συνδρομές.
- Όταν ένας πελάτης συνδέεται με έναν Broker με μια μόνιμη περίοδο λειτουργίας, ορίζεται μια επισήμανση στο μήνυμα CONNACK για να ενημερώσει τον πελάτη ότι ο Broker έχει ήδη πληροφορίες σχετικά με αυτόν, γεγονός που οδηγεί σε πιο αποτελεσματική επικοινωνία.

5.4.2.3 MQTT 5

Το MQTT 5 κυκλοφόρησε το 2019 και αντιμετωπίζει δύο ζητήματα που προκύπταν με το ευρέως χρησιμοποιούμενο πρωτόκολλο MQTT 3.1.1. Το MQTT 3.1.1 αντιμετωπίζει προβλήματα προσαρμογής ή προσθήκης μεταδεδομένων στο πρωτόκολλο, κάτι που συνηθίζεται στα δεδομένα HTTP. Επίσης, αντιμετωπίζει δυσκολίες με τη διαλειτουργικότητα κατά την επικοινωνία μεταξύ διαφορετικών πλατφορμών προμηθευτών, βιβλιοθηκών και διαδρομών δεδομένων. Οι κύριες βελτιώσεις που έρχονται με το πρωτόκολλο MQTT 5 είναι οι εξής:

- Καλύτερη υποστήριξη επεκτασιμότητας.
- Βελτιωμένη αναφορά σφαλμάτων.
- Περισσότερη αποτελεσματικότητα σε μικρούς πελάτες.
- Η προσθήκη ιδιοτήτων χρήστη.

Ένα νέο χαρακτηριστικό στο MQTT 5 είναι οι ιδιότητες στην κεφαλίδα MQTT. Μπορούν να εκχωρηθούν προσαρμοσμένες κεφαλίδες για τη μεταφορά μεταδεδομένων. Μια άλλη λειτουργικότητα που προστέθηκε, είναι οι κωδικοί αιτίας. Αυτοί οι κωδικοί βοηθούν στον προσδιορισμό του είδους του σφάλματος που παρουσιάζεται σε μια επικοινωνία MQTT. Αυτή η νέα έκδοση προσφέρει επίσης κωδικούς επιστροφής. Με αυτούς, ο Broker μπορεί να ενημερώσει τους πελάτες ποιες συγκεκριμένες δυνατότητες υποστηρίζει. Το MQTT 5 παρουσίασε το πακέτο AUTH. Αυτό το πακέτο προσφέρει προηγμένες τεχνικές ελέγχου ταυτότητας, συμπεριλαμβανομένου του OAuth. Επίσης, παρέχεται η δυνατότητα επανελέγχου της ταυτότητα ενός πελάτη χωρίς να κλείσει η τρέχουσα σύνδεση.

Στο MQTT 5, είτε ο πελάτης είτε ο Broker μπορούν να τερματίσουν τη σύνδεση. Στην έκδοση 3.1.1, μόνο ο πελάτης μπορούσε να ολοκληρώσει την μεταξύ τους επικοινωνία. Είναι προτιμότερο στις αξιόπιστες συνδέσεις να παραλείπεται η επαναποστολή μηνυμάτων. Στη νέα έκδοση οι Broker και οι πελάτες δεν αποστέλλουν αναμεταδόσεις μηνυμάτων εκτός και αν προκύψουν TCP σφάλματα, Επιτρέποντας την καλύτερη χρήση της ποιότητας υπηρεσίας (QoS). Τέλος, ένα ακόμη ενδιαφέρον χαρακτηριστικό του MQTT 5 είναι η ένδειξη της μορφής του ωφέλιμου φορτίου (payload), η οποία καθορίζει εάν είναι δυαδικής μορφής ή κείμενο.

6

IoT Πλατφόρμες Λογισμικού

6.1 Thingspeak

Μια από τις IoT πλατφόρμες που χρησιμοποιήθηκαν για την αποστολή και την προσωρινή αποθήκευση των δεδομένων από τους αισθητήρες είναι το ThingSpeak. Μέσα από τη συγκεκριμένη πλατφόρμα γίνεται δίνεται η δυνατότητα για ανάλυση των δεδομένων που σχετίζονται με τον δείκτη ποιότητας αέρα με απώτερο στόχο τον προσδιορισμό του και την ενημέρωση του χρήστη μέσω μηνύματος ηλεκτρονικής αλληλογραφίας για την κατάσταση της ποιότητας του αέρα.

Το Thingspeak είναι μια πλατφόρμα για την υλοποίηση εφαρμογών που σχετίζονται με το διαδίκτυο των πραγμάτων IoT [27]. Παρέχει μια σειρά υπηρεσιών που επιτρέπουν στον χρήστη να συγκεντρώσει, να απεικονίσει και να αναλύσει ροές δεδομένων πραγματικού χρόνου στο cloud. Επίσης δίνεται η δυνατότητα σύνταξης εκτελέσιμου κώδικα σε γλώσσα MATLAB, ο οποίος μπορεί να χρησιμοποιηθεί για ανάλυση και επεξεργασία των δεδομένων [28]. Τα βασικότερα χαρακτηριστικά του Thingspeak είναι:

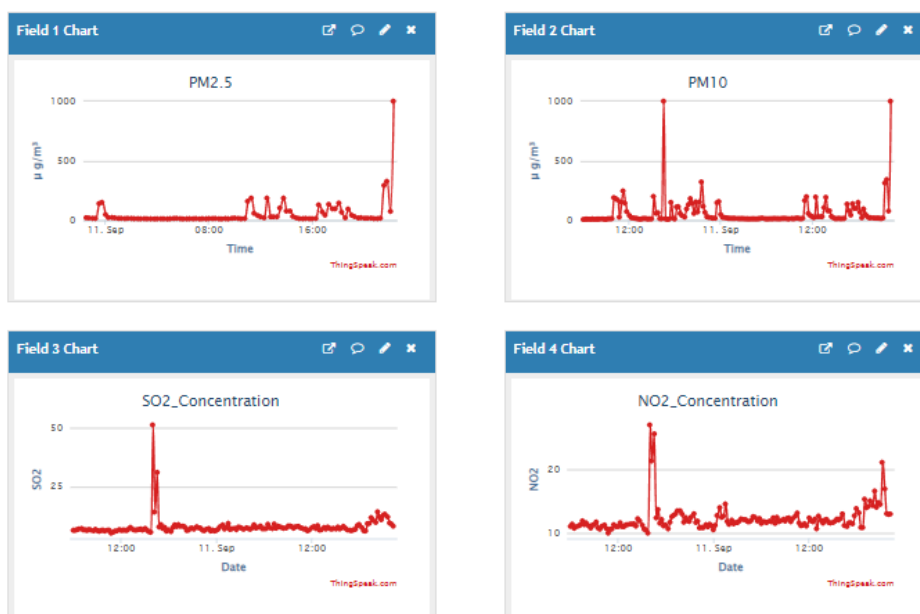
- Οπτικοποίηση των δεδομένων των αισθητήρων σε πραγματικό χρόνο.
- Παροχή δεδομένων κατ' απαίτηση από τρίτες πηγές (API).
- Εύκολη διαμόρφωση των συσκευών για την αποστολή δεδομένων στο Thingspeak χρησιμοποιώντας δημοφιλή πρωτόκολλα (HTTP, HTTPS, MQTT).
- Αυτόματη εκτέλεση αναλύσεων του IoT βάσει χρονοπρογραμμάτων ή συμβάντων.
- Δημιουργία συστημάτων IoT χωρίς την ρύθμιση διακομιστών ή την ανάπτυξη web λογισμικού.

Στην δωρεάν έκδοση των παρεχόμενων υπηρεσιών της πλατφόρμας, ο χρήστης έχει την δυνατότητα να εργαστεί με τέσσερα διαφορετικά κανάλια συλλογής δεδομένων. Σε κάθε κανάλι συνδέεται μια συσκευή IoT και σε αυτό μπορούν να συλλεχθούν δεδομένα από οκτώ διαφορετικές μεταβλητές (Σχήμα 29). Επίσης ο ρυθμός αποστολής δεδομένων προς το Thingspeak, για την

δωρεάν άδεια χρήσης, περιορίζεται στο ένα μήνυμα ανά δεκαπέντε δευτερόλεπτα, ενώ παρέχεται η δυνατότητα να εγγραφούν πολλά δεδομένα ταυτόχρονα στο ίδιο κανάλι, από την ίδια συσκευή. Τέλος, υπάρχει περιορισμός στον ρυθμό ανανέωσης των διαγραμμάτων απεικόνισης δεδομένων που έχουν προκύψει από προγράμματα MATLAB, που έχει συντάξει ο ίδιος ο χρήστης. Συγκεκριμένα, οι απεικονίσεις αυτές εμφανίζονται κατά είσοδο του χρήστη στην σελίδα του καναλιού και δεν ανανεώνονται δυναμικά. Για την ανανέωσή τους ο χρήστης πρέπει να φορτώσει ξανά την σελίδα στην οποία εμφανίζονται. Στην σελίδα κάθε καναλιού ο χρήστης μπορεί να παρακολουθήσει τις χρονοσειρές των δεδομένων που συλλέγονται και καταγράφονται σε κάθε πεδίο ξεχωριστά, οι οποίες ανανεώνονται σε πραγματικό χρόνο (Σχήμα 30).

Name	<input type="text" value="Node_1_Air_Quality"/>
Description	<input type="text"/>
Field 1	<input type="text" value="PM2.5"/> <input checked="" type="checkbox"/>
Field 2	<input type="text" value="PM10"/> <input checked="" type="checkbox"/>
Field 3	<input type="text" value="SO2"/> <input checked="" type="checkbox"/>
Field 4	<input type="text" value="NO2"/> <input checked="" type="checkbox"/>
Field 5	<input type="text" value="O3"/> <input checked="" type="checkbox"/>
Field 6	<input type="text" value="Lat"/> <input checked="" type="checkbox"/>
Field 7	<input type="text" value="Lng"/> <input checked="" type="checkbox"/>
Field 8	<input type="text"/> <input type="checkbox"/>

Σχήμα 29: Κανάλι συλλογής δεδομένων 7 μεταβλητών.

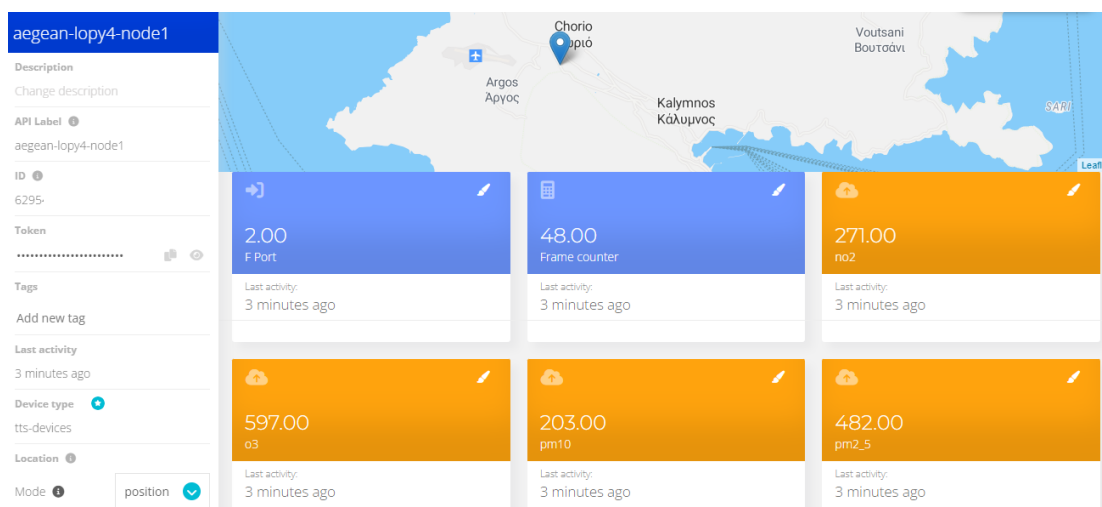


Σχήμα 30: Χρονοσειρές δεδομένων στο Thingspeak.

6.2 Ubidots

Το Ubidots είναι μια πλατφόρμα ανάλυσης και οπτικοποίησης δεδομένων για εφαρμογές του διαδικτύου των πραγμάτων. Μετατρέπει τα δεδομένα αισθητήρων σε πληροφορίες που έχουν σημασία για επιχειρηματικές αποφάσεις, αλληλεπιδράσεις μεταξύ μηχανών, εκπαιδευτικών ερευνών και αυξάνει την εξοικονόμηση παγκόσμιων πόρων.

Η πλατφόρμα Ubidots χρησιμοποιείται για την αποστολή των δεδομένων στο νέφος από οποιαδήποτε συσκευή που διαθέτει δυνατότητα διασύνδεσης στο διαδίκτυο (Σχήμα 31). Προσφέρει την επιλογή διαμόρφωσης προκαθορισμένων ενεργειών και προγραμματισμό ειδοποιήσεων βάσει των δεδομένων, σε πραγματικό.

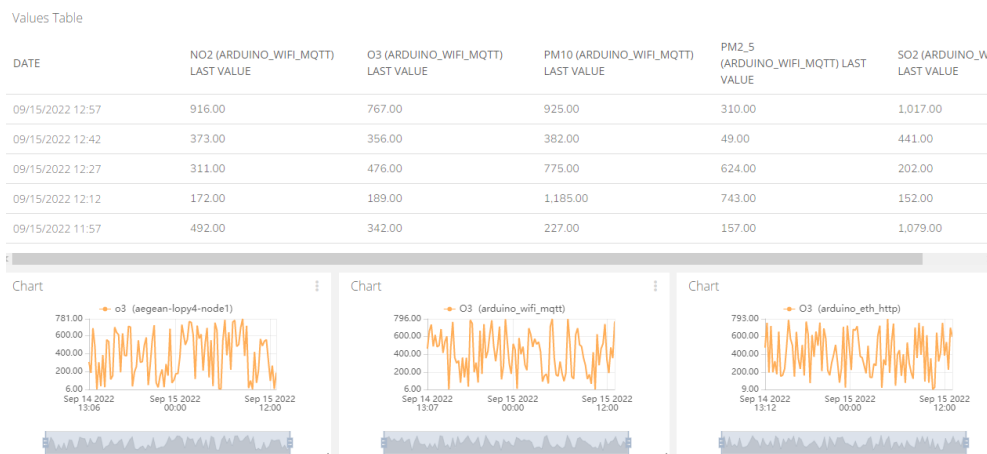


Σχήμα 31: Κανάλι συσκευής aegean-lopy4-node1 στο Ubidots.

Μερικά από τα χαρακτηριστικά της πλατφόρμας είναι [29]:

- Δημιουργία εξατομικευμένων APIs.
- Ανάλυση δεδομένων με χρήση node.js cloud ενισχύοντας τη συνδεσιμότητα των δεδομένων πέρα από τα όρια της αρχιτεκτονικής και των δυνατοτήτων που προσφέρει η πλατφόρμα.
- Επέκταση παρακολούθησης και ανάλυσης δεδομένων των εφαρμογών, με την ενσωμάτωση API όπως το Weather Underground, το Watson της IBM, το Google Locations, το Zapier και άλλα.
- Υποστήριξη περισσότερων από 200 βιβλιοθηκών και SDKs για την εύκολη διασύνδεση των συσκευών με την πλατφόρμα μέσω HTTP, MQTT, TCP, UDP ή με ανάλυση προσαρμοσμένων/βιομηχανικών πρωτοκόλλων.
- Υποστήριξη εργαλείου μαθηματικών υπολογισμών. Προσφέρει τη δυνατότητα δημιουργίας σύνθετων μεταβλητών που προκύπτουν από τη μαθηματική συσχέτιση ακατέργαστων δεδομένων που προέρχονται από τις συσκευές. Η υπολογιστική απόδοση των σύνθετων μεταβλητών εξαρτάται από την πολυπλοκότητα της συσχέτισης.
- Βελτιστοποιημένη υποδομή για λήψη, υπολογισμό και επεξεργασία εκατομμυρίων δεδομένων το δευτερόλεπτο.

- Διατήρηση των δεδομένων για 2 χρόνια, προσφέροντας χώρους αποθήκευσης και εξόρυξης δεδομένων για την εξαγωγή πρόσθετων πληροφοριών, ανίχνευση ανωμαλιών ή τη δημιουργία προγνωστικής συντήρησης.
- Δημιουργία εξατομικευμένης σελίδας παρακολούθησης των δεδομένων με δυνατότητα αναπαράστασής τους σε γραφήματα, πίνακες, χάρτες σε πραγματικό χρόνο (Σχήμα 32). Οι σελίδες μπορεί να είναι δημόσιες ή ιδιωτικές.
- Δημιουργία αναφορών σχετικά με τους σταθμούς ενδιαφέροντος και τα δεδομένα που σχετίζονται με αυτούς. Οι αναφορές μπορούν να αποστέλλονται ηλεκτρονικά στους αποδέκτες κατόπιν χρονοπρογραμματισμού.
- Διαχείριση συμβάντων και αποστολή ενημερωτικών μηνυμάτων μέσω e-mail, sms, telegram, slack, ηχητικής κλίσης, κατά την ενεργοποίηση προγραμματισμένων συνθηκών. Παράδειγμα ενεργοποίησης συμβάντος είναι όταν η τιμή του μετρούμενου μεγέθους ενός αισθητήρα υπερβεί μια ορισμένη τιμή ή ένα χρονικό διάστημα ή όταν η γεωγραφική θέση του σταθμού αλλάξει γεωγραφική περιοχή.



Σχήμα 32: Εξατομικευμένη σελίδα παρακολούθησης δεδομένων στο Ubidots.

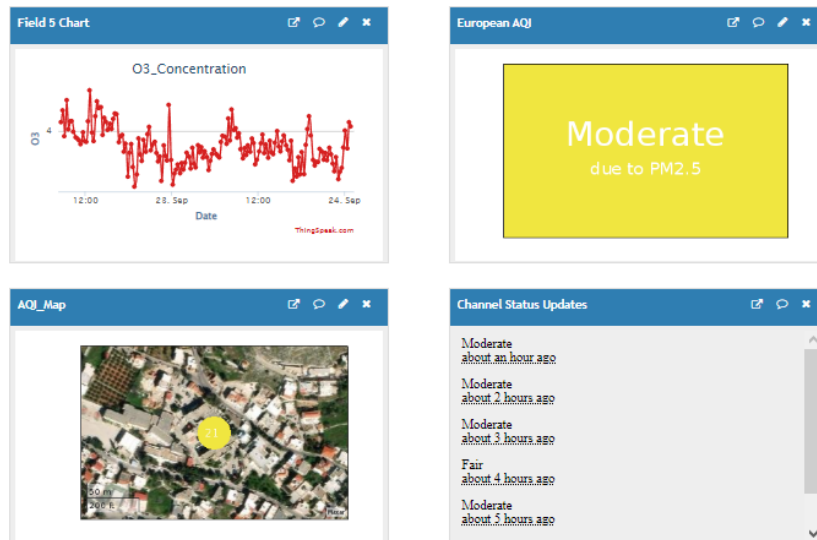
6.3 Σύγκριση Thingspeak – Ubidots

Οι πλατφόρμες διαθέτουν διαφορετικά πλάνα τιμολόγησης με δωρεάν και εμπορικές άδειες χρήσης, ανάλογα με το είδος και τις ανάγκες της εφαρμογής. Στο πλαίσιο της εργασίας χρησιμοποιήθηκαν οι δωρεάν άδειες που προσφέρονται και οι ακόλουθες συγκρίσεις αφορούν το συγκεκριμένο προφίλ χρήσης.

Για το Thingspeak:

- Διατίθενται τέσσερα (4) κανάλια επικοινωνίας ανά λογαριασμό χρήστη. Κάθε κανάλι αντιστοιχεί σε μία IoT συσκευή.
- Σε κάθε κανάλι μπορούν να καταγραφούν μέχρι οκτώ διαφορετικά μετρούμενα μεγέθη της συσκευής.
- Η ελάχιστη περίοδος μεταξύ δυο διαδοχικών μηνυμάτων που αποστέλλει μια συσκευή στην πλατφόρμα είναι δεκαπέντε (15) δευτερόλεπτα. Στις εμπορικές άδειες χρήσης ο χρόνος αυτός φτάνει μέχρι το ένα (1) δευτερόλεπτο [30].

- Κάθε κανάλι προσφέρει ένα στατικό πίνακα αναπαράστασης χρονοσειρών των δεδομένων. Ο πίνακας αυτός μπορεί να εμπλουτιστεί με νέες οπτικοποιήσεις, έπειτα από επεξεργασία των δεδομένων μέσω της γλώσσας προγραμματισμού MATLAB (Σχήμα 33). Για τις οπτικοποιήσεις αυτές ο χρήστης δεν μπορεί να ορίσει ρυθμό ανανέωσης ενώ ενεργοποιούνται μέσω της εκτέλεσης του προγράμματος MATLAB κατά την είσοδο στο κανάλι της συσκευής.



Σχήμα 33: Οπτικοποιήσεις δεδομένων στο Thingspeak μέσω MATLAB.

Για το Ubidots:

- Διατίθενται τρία (3) κανάλια επικοινωνίας ανά λογαριασμό χρήστη. Κάθε κανάλι αντιστοιχεί σε μία IoT συσκευή.
- Δεν υπάρχει αναφορά για το μέγιστο πλήθος των μεταβλητών που μπορούν να καταγραφούν σε ένα κανάλι. Στα πλαίσια της εργασίας ο μέγιστος αριθμός μεταβλητών που χρησιμοποιήθηκε ήταν δέκα (10).
- Η μέγιστη συχνότητα αποστολής μηνυμάτων από μια συσκευή προς την πλατφόρμα κυμαίνεται από ένα (1) έως τέσσερα (4) μηνύματα το δευτερόλεπτο, με μέγιστο μέγεθος μηνύματος 10Kbytes [31].
- Η απεικόνιση των δεδομένων γίνεται μέσω ενός δυναμικού πίνακα που μπορεί να περιλαμβάνει διαφορετικούς τύπους αναπαράστασης όπως γραφήματα, δείκτες, πίνακες, χάρτες κ.α.. Ο δυναμικός πίνακας μπορεί να αναπαριστά δεδομένα από διαφορετικά κανάλια συσκευών, προσφέροντας μια συγκεντρωτική εικόνα της εφαρμογής στο χρήστη.
- Δεν προσφέρεται η δυνατότητα σύνταξης συναρτήσεων με κάποια γλώσσα προγραμματισμού για την επεξεργασία και την ανάλυση των δεδομένων.

7

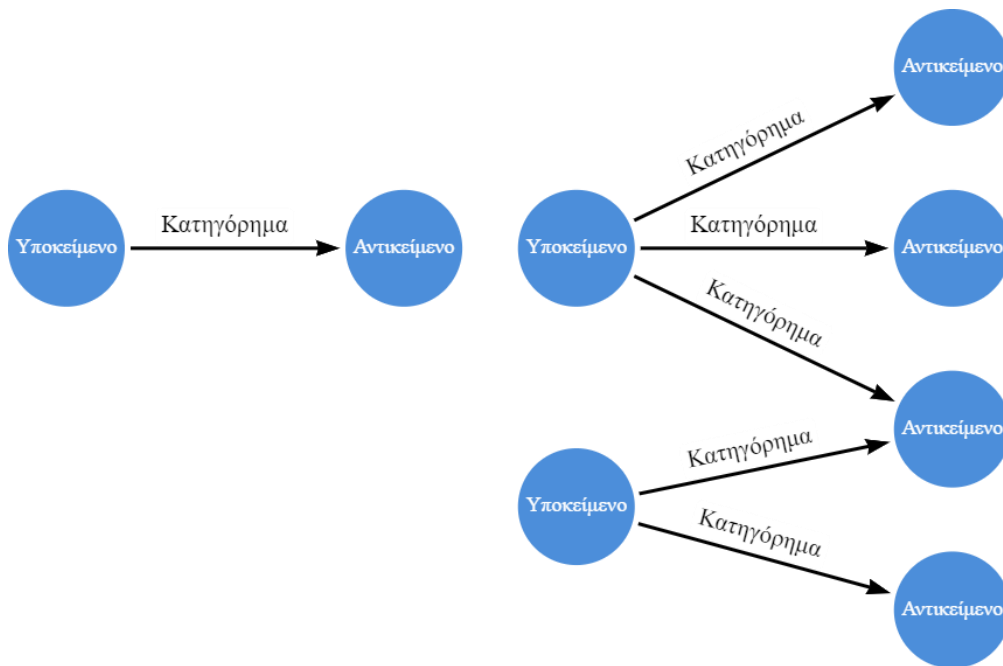
Σημασιολογικός Ιστός

7.1 Εισαγωγή

Ο σημασιολογικός ιστός είναι ένα σύνολο τεχνολογιών και προτύπων που έχουν σχεδιαστεί για να διευκολύνουν τους υπολογιστές και άλλες συσκευές να κατανοούν και να επεξεργάζονται το νόημα των πληροφοριών στο Διαδίκτυο. Ένας από τους βασικούς στόχους του σημασιολογικού ιστού είναι να επιτρέψει στις μηχανές να κατανοήσουν τις σχέσεις μεταξύ διαφορετικών κομματιών δεδομένων και να χρησιμοποιήσουν αυτή την κατανόηση για την εκτέλεση εργασιών και τη λήψη αποφάσεων. Βασίζεται στην ιδέα της παροχής ενός κοινού πλαισίου για την αναπαράσταση και την οργάνωση των δεδομένων στο διαδίκτυο, επιτρέποντας σε ετερογενή συστήματα και εφαρμογές να επικοινωνούν και να ανταλλάσσουν πληροφορίες με αποτελεσματικό τρόπο.

Ένα από τα βασικά στοιχεία του σημασιολογικού ιστού είναι η χρήση μεταδεδομένων, τα οποία είναι δεδομένα για δεδομένα. Τα μεταδεδομένα περιγράφουν το περιεχόμενο και το πλαίσιο των πόρων Ιστού και βοηθούν στον καθορισμό των σχέσεων μεταξύ τους. Συνήθως αναπαρίσταται χρησιμοποιώντας μια επίσημη γλώσσα, όπως το Πλαίσιο Περιγραφής Πόρων (RDF), το οποίο είναι ένα πρότυπο που αναπτύχθηκε από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C). Το RDF παρέχει ένα σύνολο δομών για την αναπαράσταση πόρων, ιδιοτήτων και σχέσεων μεταξύ τους. Οι πόροι αντιπροσωπεύουν οντότητες όπως για παράδειγμα άτομα, τοποθεσίες ή συμβάντα, ενώ οι ιδιότητες είναι χαρακτηριστικά ή ιδιότητες τέτοιων οντοτήτων όπως ονόματα ή διευθύνσεις. Οι σχέσεις μεταξύ πόρων και ιδιοτήτων αναπαρίστανται χρησιμοποιώντας τριπλέτες (Σχήμα 34), που αποτελούνται από ένα υποκείμενο (subject), ένα κατηγορημα (predicate) και ένα αντικείμενο (object). Ένα ακόμη βασικό χαρακτηριστικό του σημασιολογικού ιστού είναι η χρήση οντολογιών, οι οποίες είναι επίσημες αναπαραστάσεις ενός συνόλου εννοιών και των σχέσεων μεταξύ αυτών, εντός ενός τομέα. Οι οντολογίες παρέχουν μια κοινή δομή για την περιγραφή και την

αναπαράσταση των δεδομένων, επιτρέποντας σε διαφορετικά συστήματα και εφαρμογές να επικοινωνούν και να ανταλλάσσουν πληροφορίες.



Σχήμα 34: Παραδείγματα απλής τριπλέτας και συνδυασμού τριπλετών.

Επιπρόσθετα, ο σημασιολογικός ιστός περιλαμβάνει μια ποικιλία εργαλείων και τεχνολογιών για πρόσβαση, διαχείριση και χειρισμό δεδομένων στον ιστό. Ένα τέτοιο εργαλείο είναι η γλώσσα ερωτημάτων SPARQL, η οποία είναι ένα πρότυπο που αναπτύχθηκε από την W3C για την αναζήτηση δεδομένων RDF. Ένα ακόμη εργαλείο είναι το Semantic Web Services Framework (SWSF), το οποίο είναι ένα σύνολο προτύπων και τεχνολογιών που επιτρέπουν την ενοποίηση και τη διαλειτουργικότητα διαφορετικών συστημάτων και εφαρμογών στον Ιστό.

Ο σημασιολογικός ιστός έχει τη δυνατότητα να βελτιώσει σημαντικά την αποδοτικότητα και την αποτελεσματικότητα των εργασιών που περιλαμβάνουν την επεξεργασία και την ανάλυση δεδομένων στο διαδίκτυο. Μπορεί επίσης να συμβάλει στη βελτίωση της ακρίβειας και της αξιοπιστίας των πληροφοριών και να διευκολύνει την καλύτερη επικοινωνία μεταξύ των μηχανών. Ορισμένες εφαρμογές του σημασιολογικού ιστού περιλαμβάνουν τη βελτίωση των μηχανών αναζήτησης, την ενσωμάτωση και τη διαλειτουργικότητα διαφορετικών συστημάτων και εφαρμογών και την υποστήριξη εργασιών συλλογισμού και λήψης αποφάσεων. Η τεχνολογία αυτή αποτελεί ένα σημαντικό βήμα για την εξέλιξη του παγκόσμιου ιστού, επιτρέποντας στις μηχανές να ερμηνεύουν την έννοια του περιεχομένου του, διευκολύνοντας ταυτόχρονα την ανταλλαγή πληροφοριών με αποδοτικό τρόπο. Τέλος, έχει τη δυνατότητα να μεταμορφώσει τον τρόπο που χρησιμοποιούμε το διαδίκτυο και να ενεργοποιήσει ένα ευρύ φάσμα νέων εφαρμογών και υπηρεσιών.

7.2 Οντολογίες

Οι οντολογίες αποτελούν ένα βασικό συστατικό του σημασιολογικού ιστού, μια συλλογή τεχνολογιών και προτύπων που στοχεύουν να κάνουν τις πληροφορίες στον Παγκόσμιο Ιστό πιο εύκολα κατανοητές από τις μηχανές. Μια οντολογία είναι μια επίσημη αναπαράσταση ενός συνόλου εννοιών ενός τομέα και των σχέσεων μεταξύ των εννοιών αυτών. Παρέχει ένα κοινό λεξιλόγιο για την περιγραφή και την αναπαράσταση δεδομένων, επιτρέποντας σε διαφορετικά συστήματα και εφαρμογές να επικοινωνούν και να ανταλλάσσουν πληροφορίες αποτελεσματικά.

Οι οντολογίες τυπικά ορίζονται χρησιμοποιώντας μια επίσημη γλώσσα, όπως η Γλώσσα Οντολογίας Ιστού (OWL), η οποία είναι ένα πρότυπο που αναπτύχθηκε από την Κοινοπραξία του Παγκόσμιου Ιστού (W3C). Η OWL παρέχει ένα σύνολο δομών για τον ορισμό κλάσεων, ιδιοτήτων και σχέσεων μεταξύ τους. Οι κλάσεις αντιπροσωπεύουν τύπους οντοτήτων, όπως άτομα, μέρη ή γεγονότα, ενώ οι ιδιότητες αντιπροσωπεύουν χαρακτηριστικά ή χαρακτηριστικά αυτών των οντοτήτων, όπως ονόματα ή διευθύνσεις. Οι σχέσεις μεταξύ κλάσεων και ιδιοτήτων ορίζονται χρησιμοποιώντας αξιώματα, τα οποία είναι λογικές δηλώσεις που περιγράφουν τις συνθήκες υπό τις οποίες ισχύει μια σχέση. Οι οντολογίες χρησιμοποιούνται συχνά για τον ορισμό της γνώσης για έναν συγκεκριμένο τομέα, όπως οι έννοιες και οι σχέσεις σε ένα συγκεκριμένο πεδίο ή κλάδο. Για παράδειγμα, μια οντολογία για τον τομέα της υγειονομικής περίθαλψης μπορεί να περιλαμβάνει κλάσεις για ασθένειες, θεραπείες και φάρμακα, καθώς και ιδιότητες για συμπτώματα και παρενέργειες, καθώς και σχέσεις μεταξύ αυτών των εννοιών.

Ένα από τα κύρια οφέλη των οντολογιών είναι ότι παρέχουν ένα κοινό λεξιλόγιο για την περιγραφή και την αναπαράσταση δεδομένων. Αυτό επιτρέπει σε διαφορετικά συστήματα και εφαρμογές να επικοινωνούν και να ανταλλάσσουν πληροφορίες αποτελεσματικά, καθώς μπορούν να χρησιμοποιούν τους ίδιους όρους και έννοιες για να αναπαραστήσουν τα δεδομένα. Για παράδειγμα, ένας πάροχος υγειονομικής περίθαλψης προσπαθεί να μοιραστεί πληροφορίες ασθενών με ένα ερευνητικό ινστιτούτο. Χωρίς την οντολογία, ο πάροχος και το ινστιτούτο ενδέχεται να χρησιμοποιούν διαφορετικούς όρους και έννοιες για να αναπαραστήσουν τα δεδομένα, καθιστώντας δύσκολη την κατανόηση και την ερμηνεία των δεδομένων. Με μια οντολογία, μπορούν να χρησιμοποιήσουν ένα κοινό λεξιλόγιο για να αναπαραστήσουν τα δεδομένα, διευκολύνοντας την κατανόηση και την ερμηνεία τους.

Οι οντολογίες μπορούν επίσης να χρησιμοποιηθούν για τη βελτίωση της ακρίβειας και της αξιοπιστίας των πληροφοριών στον ιστό. Ορίζοντας σαφείς και ακριβείς ορισμούς για έννοιες και σχέσεις, οι οντολογίες μπορούν να συντελέσουν στη μείωση της ασάφειας και της παρερμηνείας. Για παράδειγμα, ένας χρήστης ψάχνει στον ιστό πληροφορίες σχετικά με μια συγκεκριμένη ασθένεια. Χωρίς την οντολογία, ο χρήστης μπορεί να συναντήσει μια ποικιλία διαφορετικών ορισμών και περιγραφών της νόσου, που θα μπορούσαν να οδηγήσουν σε σύγχυση και παραπληροφόρηση. Με την οντολογία, ο χρήστης μπορεί να είναι σίγουρος ότι οι πληροφορίες στις οποίες έχει πρόσβαση χρησιμοποιούν ένα συνεπές και καλά καθορισμένο λεξιλόγιο.

Ένα ακόμη πλεονέκτημα των οντολογιών είναι ότι επιτρέπουν την ενοποίηση και τη διαλειτουργικότητα διαφορετικών συστημάτων και εφαρμογών. Παρέχοντας ένα κοινό λεξιλόγιο, οι οντολογίες επιτρέπουν σε διαφορετικά συστήματα να κατανοούν και να ερμηνεύουν τα δεδομένα που ανταλλάσσονται, ανεξάρτητα από τις συγκεκριμένες τεχνικές λεπτομέρειες των συστημάτων.

Αυτό καθιστά δυνατή τη συνεργασία διαφορετικών συστημάτων και την αποτελεσματική ανταλλαγή δεδομένων, βελτιώνοντας την αποδοτικότητα και την αποτελεσματικότητα των εργασιών που περιλαμβάνουν επεξεργασία και ανάλυση δεδομένων από πολλαπλές πηγές.

Οι οντολογίες μπορούν επίσης να χρησιμοποιηθούν για την υποστήριξη εργασιών συλλογισμού και λήψης αποφάσεων. Με την αναπαράσταση των σχέσεων μεταξύ διαφορετικών εννοιών και ιδιοτήτων, οι οντολογίες μπορούν να χρησιμοποιηθούν για την εξαγωγή νέων πληροφοριών και για τη λήψη αποφάσεων με βάση αυτές τις πληροφορίες. Για παράδειγμα, ένας πάροχος υγειονομικής περίθαλψης προσπαθεί να καθορίσει την καταλληλότερη θεραπεία για έναν ασθενή. Χρησιμοποιώντας μια οντολογία που αντιπροσωπεύει τις σχέσεις μεταξύ ασθενειών, θεραπειών και παρενεργειών, ο πάροχος μπορεί να χρησιμοποιήσει αλγόριθμους συλλογισμού για να συμπεράνει την πιθανή αποτελεσματικότητα διαφορετικών θεραπειών με βάση τα συμπτώματα και το ιατρικό ιστορικό του ασθενούς.

Εν κατακλείδι, οι οντολογίες αποτελούν ένα ισχυρό εργαλείο για την αναπαράσταση και την οργάνωση της γνώσης στον ιστό. Παρέχοντας ένα κοινό λεξιλόγιο και μια επίσημη αναπαράσταση εννοιών και σχέσεων, οι οντολογίες επιτρέπουν την ενοποίηση και διαλειτουργικότητα διαφορετικών συστημάτων και εφαρμογών, βελτιώνουν την ακρίβεια και την αξιοπιστία των πληροφοριών και υποστηρίζουν εργασίες συλλογισμού και λήψης αποφάσεων.

7.3 Διαδίκτυο των Πραγμάτων και σημασιολογία

Στο πλαίσιο του Διαδικτύου των Πραγμάτων (IoT), ο σημασιολογικός ιστός μπορεί να είναι ιδιαίτερα χρήσιμος καθότι συντελεί στη διευκόλυνση της επικοινωνίας και της διαλειτουργικότητας μεταξύ διαφορετικών συσκευών και συστημάτων. Για παράδειγμα, σε μια έξυπνη πόλη που χρησιμοποιεί αισθητήρες IoT για τη μέτρηση του δείκτη ποιότητας αέρα (AQI), ο σημασιολογικός ιστός μπορεί να βοηθήσει να διασφαλιστεί ότι τα δεδομένα που συλλέγονται από τους αισθητήρες ερμηνεύονται με ακρίβεια και κατανοούνται από άλλες συσκευές και συστήματα που το χρησιμοποιούν. Επίσης, εάν τα δεδομένα που συλλέγονται από τους αισθητήρες περιλαμβάνουν μεταδεδομένα που υποδεικνύουν την τοποθεσία και την ώρα που έγιναν οι μετρήσεις, άλλες συσκευές και συστήματα θα μπορούσαν να χρησιμοποιήσουν αυτές τις πληροφορίες για να εμφανίσουν χάρτες σε πραγματικό χρόνο των επιπέδων AQI σε όλη την πόλη. Αυτό θα μπορούσε να είναι χρήσιμο σε κατοίκους της πόλης, οι οποίοι θα μπορούσαν να χρησιμοποιήσουν αυτές τις πληροφορίες για να λάβουν τεκμηριωμένες αποφάσεις σχετικά με την ποιότητα του αέρα και τις κατάλληλες προφυλάξεις όταν κριθεί απαραίτητο.

Η τελευταία λέξη της τεχνολογίας στα σημασιολογικά δεδομένα IoT (Internet of Things) αναφέρεται στις τρέχουσες βέλτιστες πρακτικές και τεχνολογίες για την αναπαράσταση, την αποθήκευση και την επεξεργασία δεδομένων που παράγονται από συσκευές και συστήματα IoT. Συνολικά, περιλαμβάνεται η χρήση ενός συνδυασμού τεχνολογιών και προσεγγίσεων για να καταστεί δυνατή η αποτελεσματική αναπαράσταση, αποθήκευση και επεξεργασία δεδομένων που παράγονται από συσκευές και συστήματα IoT. Μερικές από τις βασικές τάσεις και εξελίξεις σε αυτόν τον τομέα περιλαμβάνουν:

- Τη χρήση σημασιολογικών μεταδεδομένων για τον σχολιασμό και την περιγραφή δεδομένων IoT: Τα σημασιολογικά μεταδεδομένα είναι δομημένα δεδομένα που χρησιμοποιούνται για να

περιγράφουν το νόημα και το πλαίσιο άλλων δεδομένων. Με τον σχολιασμό των δεδομένων IoT με σημασιολογικά μεταδεδομένα, μπορεί να είναι ευκολότερο για άλλες συσκευές και συστήματα να κατανοήσουν και να κάνουν χρήση των δεδομένων.

- Την ανάπτυξη μοντέλων και οντολογιών σημασιολογικών δεδομένων: Οι οντολογίες είναι επίσημες αναπαραστάσεις των εννοιών και των σχέσεων μέσα σε έναν τομέα γνώσης. Με τη δημιουργία οντολογιών για δεδομένα IoT για συγκεκριμένο τομέα, μπορεί να είναι ευκολότερο για τις συσκευές και τα συστήματα να κατανοήσουν και να χρησιμοποιήσουν τα δεδομένα.
- Τη χρήση αλγόριθμων μηχανικής μάθησης και της τεχνητής νοημοσύνης για την επεξεργασία και ανάλυση δεδομένων IoT: Οι τεχνικές μηχανικής μάθησης και τεχνητής νοημοσύνης μπορούν να χρησιμοποιηθούν για την εξαγωγή πληροφοριών και μοτίβων από μεγάλες ποσότητες δεδομένων IoT. Αυτό μπορεί να είναι ιδιαίτερα χρήσιμο για εργασίες όπως η ανίχνευση ανωμαλιών και η προγνωστική συντήρηση.
- Την υιοθέτηση προτύπων ανοιχτών δεδομένων και πλαισίων διαλειτουργικότητας: Για να καταστεί δυνατή η απρόσκοπτη ανταλλαγή δεδομένων μεταξύ διαφορετικών συσκευών και συστημάτων IoT, είναι σημαντικό να υιοθετηθούν πρότυπα ανοιχτών δεδομένων και πλαίσια διαλειτουργικότητας. Αυτό μπορεί να βοηθήσει στη διασφάλιση ότι τα δεδομένα αποθηκεύονται και μεταδίδονται με τρόπο συνεπή και συμβατό με ένα ευρύ φάσμα συσκευών και συστημάτων.

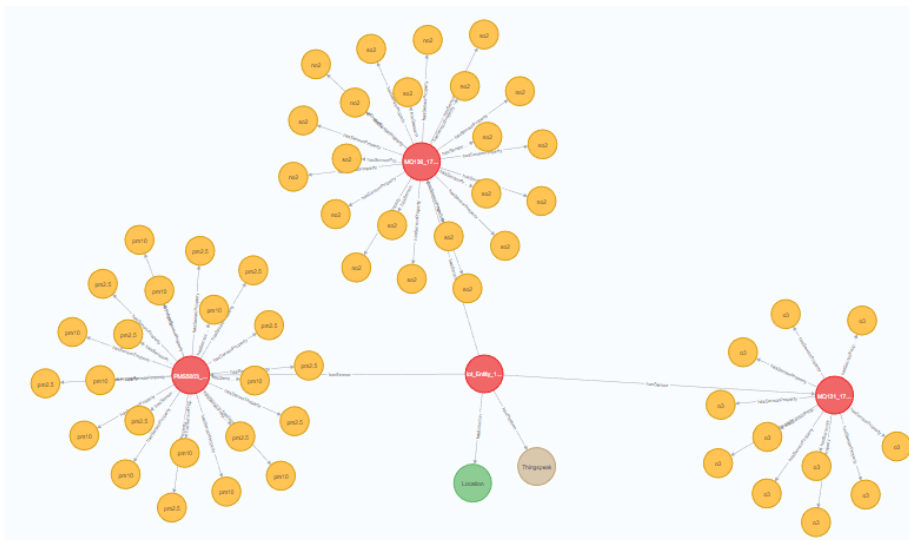
Ο σημασιολογικός ιστός μπορεί να αποτελέσει ένα ισχυρό εργαλείο επιτρέποντας στο IoT να λειτουργεί πιο αποτελεσματικά και αποδοτικά, και μπορεί να βοηθήσει στη διευκόλυνση των συσκευών και των συστημάτων να επικοινωνούν και να συνεργάζονται. Τα οφέλη από τη χρήση του σημασιολογικού ιστού στο Διαδίκτυο των Πραγμάτων (IoT) είναι:

- Βελτιωμένη επικοινωνία και διαλειτουργικότητα μεταξύ συσκευών: Ο σημασιολογικός ιστός μπορεί να επιτρέψει στις συσκευές να επικοινωνούν μεταξύ τους με πιο ουσιαστικό τρόπο, χρησιμοποιώντας μεταδεδομένα αναγνώσιμα από μηχανή για να περιγράψουν το νόημα και το πλαίσιο των δεδομένων που συλλέγουν. Αυτό μπορεί να βελτιώσει τη διαλειτουργικότητα διαφορετικών συσκευών και συστημάτων, καθιστώντας ευκολότερη την ενοποίηση και την ανταλλαγή δεδομένων μεταξύ τους.
- Βελτιωμένες δυνατότητες λήψης αποφάσεων και επίλυσης προβλημάτων: Ο σημασιολογικός ιστός μπορεί να επιτρέψει την αποτελεσματική ανάλυση και επεξεργασία μεγάλων ποσοτήτων δεδομένων, επιτρέποντας σε συσκευές και συστήματα να εντοπίζουν τάσεις και πρότυπα και να λαμβάνουν πιο τεκμηριωμένες αποφάσεις.
- Αυξημένη απόδοση και παραγωγικότητα: Επιτρέποντας την ενοποίηση και τη διαλειτουργικότητα διαφορετικών συσκευών και συστημάτων, ο σημασιολογικός ιστός μπορεί να βελτιώσει την αποδοτικότητα και την αποτελεσματικότητα των εργασιών που περιλαμβάνουν επεξεργασία και ανάλυση δεδομένων από πολλαπλές πηγές.
- Βελτιωμένη ακρίβεια και αξιοπιστία των πληροφοριών: Η χρήση μεταδεδομένων και οντολογιών αναγνώσιμων από μηχανή μπορεί να βοηθήσει στη μείωση της ασάφειας και της παρανόησης, βελτιώνοντας την ακρίβεια και την αξιοπιστία των πληροφοριών στον ιστό.
- Βελτιωμένη ασφάλεια και απόρρητο: Ο σημασιολογικός ιστός μπορεί να επιτρέψει τη χρήση προηγμένων πρωτοκόλλων ασφαλείας και ελέγχου απορρήτου, όπως κρυπτογραφημένη επικοινωνία και έλεγχος πρόσβασης, για την προστασία ευαίσθητων δεδομένων.

7.4 Neo4j

Το Neo4j είναι ένα σύστημα διαχείρισης βάσεων γραφοδεδομένων ανοιχτού κώδικα που αναπτύχθηκε από τη Neo4j, Inc. Έχει σχεδιαστεί για την αποθήκευση, διαχείριση και αναζήτηση δεδομένων γράφων, τα οποία αποτελούνται από κόμβους (που αντιπροσωπεύουν οντότητες ή αντικείμενα) και ακμές (που αντιπροσωπεύουν σχέσεις ή συνδέσεις μεταξύ κόμβων). Το Neo4j βασίζεται στην ιδιότητα μοντελοποίησης γράφων, το οποίο αναπαριστά τα δεδομένα ως ένα σύνολο κόμβων και σχέσεων που έχουν ιδιότητες (ζεύγη κλειδιού-τιμής) συνδεδεμένες σε αυτούς (Σχήμα 35). Αυτό επιτρέπει την αναπαράσταση πολύπλοκων, αλληλένδετων δεδομένων με ευέλικτο και επεκτάσιμο τρόπο.

Το Neo4j χρησιμοποιείται συνήθως για εργασίες όπως η αποθήκευση και η αναζήτηση δεδομένων κοινωνικού δικτύου, οι μηχανές συστάσεων, ο εντοπισμός απάτης και η ανάλυση δικτύου. Χρησιμοποιείται επίσης συχνά σε συνδυασμό με άλλες τεχνολογίες, όπως η μηχανική μάθηση και η επεξεργασία φυσικής γλώσσας, για την εκτέλεση εργασιών όπως η προγνωστική ανάλυση και η ανάλυση συναισθήματος. Ένα από τα κύρια χαρακτηριστικά της βάσης Neo4j είναι η ικανότητά της να αποθηκεύει αποτελεσματικά και να αναζητά μεγάλα και πολύπλοκα σύνολα γραφοδεδομένων. Παρέχει έναν αριθμό ισχυρών δυνατοτήτων αναζήτησης και δημιουργίας ευρετηρίου, όπως η γλώσσα υποβολής ερωτημάτων Cypher, η οποία είναι μια δηλωτική γλώσσα ειδικά σχεδιασμένη για την αναζήτηση δεδομένων σε γράφους. Παρουσιάζει υψηλή επεκτασιμότητα και μπορεί να χειριστεί μεγάλες ποσότητες δεδομένων και κίνησης. Μπορεί να αναπτυχθεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των εγκαταστάσεων, στο cloud και ως υπηρεσία. Το μοντέλο γραφήματος ιδιοτήτων που χρησιμοποιείται από τη βάση, επιτρέπει την αναπαράσταση πολύπλοκων, αλληλένδετων δεδομένων με ευέλικτο τρόπο. Ένα από τα χαρακτηριστικά της Neo4j είναι ότι διαθέτει μια ισχυρή και ενεργή κοινότητα χρηστών και προγραμματιστών, παρέχοντας πληθώρα πόρων και υποστήριξη σε όσους χρησιμοποιούν τη βάση δεδομένων. Τέλος, το γεγονός ότι είναι ανοιχτού κώδικα σημαίνει ότι είναι ελεύθερα διαθέσιμη για χρήση και τροποποίηση. Αυτό μπορεί να είναι επωφελές για όσους θέλουν να προσαρμόσουν τη βάση δεδομένων ή να την ενσωματώσουν με άλλα συστήματα.



Σχήμα 35: Παράδειγμα απεικόνισης γραφοδεδομένων στη Neo4j.

8

Περιβαλλοντικοί και Γεωχωρικοί Αισθητήρες

8.1 Αισθητήρας αιωρούμενων σωματιδίων PMS5003

Για τον υπολογισμό της συγκέντρωσης των αιωρούμενων σωματιδίων στην ατμόσφαιρα, χρησιμοποιούνται συνήθως αισθητήρες οπτικής ανίχνευσης. Ο αισθητήρας PMS5003 (Σχήμα 36) που χρησιμοποιήθηκε στην εφαρμογή, υπολογίζει τον αριθμό των αιωρούμενων σωματιδίων διαφορετικού μεγέθους που βρίσκονται στον αέρα και αναπαράγει τις μετρήσεις που συνέλεξε μέσω της σειριακής διασύνδεσής του με τον επεξεργαστή.

Αυτός ο αισθητήρας μπορεί να εισαχθεί σε μεταβλητά όργανα που σχετίζονται με τη συγκέντρωση αιωρούμενων σωματιδίων στον αέρα ή σε άλλους εξοπλισμούς περιβαλλοντικής βελτίωσης για να παρέχουν έγκαιρα σωστά δεδομένα συγκέντρωσης. Παρέχει την δυνατότητα μέτρησης αιρούμενων σωματιδίων μεγέθους PM_1 , $PM_{2.5}$ και PM_{10} .

Τα κύρια χαρακτηριστικά του αισθητήρα PMS5003 είναι:

- Η μηδενική τιμή ψευδούς συναγερμού.
- Η απόκρισή του σε πραγματικό χρόνο.
- Η αξιοπιστία των δεδομένων των μετρήσεων.
- Η ελάχιστη διακριτή διάμετρος σωματιδίων που μπορεί να μετρηθεί είναι της τάξης των 0.3 μικρομέτρων.
- Η υψηλή απόδοση που διαθέτει έναντι των ηλεκτρομαγνητικών παρεμβολών, λόγω της μεταλλικής θωράκισης που διαθέτει.
- Η προαιρετική κατεύθυνση εισόδου και εξόδου αέρα για προσαρμογή διαφορετικού σχεδιασμού.

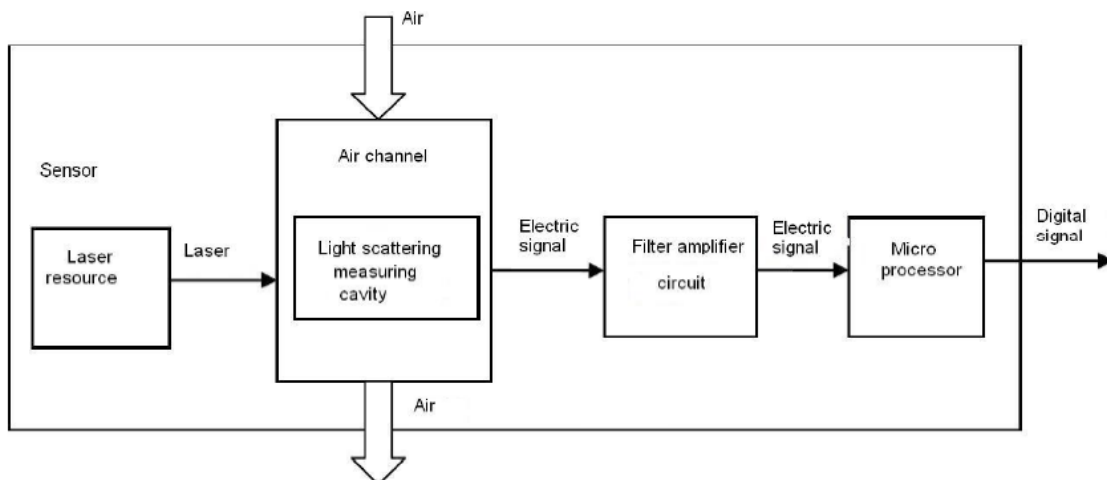


Pin	Purpose
1	VCC (5V)
2	GND
3	SET (3.3V, active low)
4	RX data in (3.3V)
5	TX data out (3.3V)
6	RESET (3.3V, active low)
7	nc
8	nc

Power is 5V. All logic is 3.3V!

Σχήμα 36: Αισθητήρας αιωρούμενων σωματιδίων PMS5003.

Η αρχή λειτουργίας του αισθητήρα βασίζεται στην σκέδαση του φωτός και αποτελεί την πιο συνήθη μέθοδο για τέτοιου είδους μετρήσεις. Ένας ανεμιστήρας αναρροφά ατμοσφαιρικό αέρα δημιουργώντας μια ροή που κατευθύνεται μπροστά από μια δέσμη laser (Σχήμα 37). Με βάση την σκέδαση που προκαλεί η δέσμη από την πρόσπτωση της στα αιωρούμενα σωματίδια, υπολογίζεται το μέγεθός τους και το πλήθος τους ανά μονάδα όγκου.



Σχήμα 37: Διάγραμμα λειτουργίας αισθητήρα αιωρούμενων σωματιδίων.

Η ακρίβεια του αισθητήρα μπορεί να επηρεαστεί σημαντικά από την θερμοκρασία και την υγρασία του περιβάλλοντος, οδηγώντας σε λανθασμένους υπολογισμούς των συγκεντρώσεων των αιωρούμενων σωματιδίων. Η αλληλεπίδραση της υγρασίας με τα σωματίδια, υποβαθμίζει τις οπτικές τους ιδιότητες με αποτέλεσμα τα σήματα που δέχεται ο ανιχνευτής λόγω της σκέδασης, να μην είναι αντιπροσωπευτικά. Επιπλέον λόγω της υγρασίας τα οπτικά μέρη του αισθητήρα, μπορούν να θαμπώσουν και να οδηγήσουν σε αδυναμία μέτρησης των σωματιδίων [32].

8.2 Αισθητήρες αερίων MQ

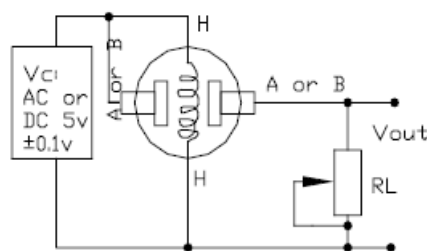
Οι αισθητήρες αερίων MQ χρησιμοποιούνται ευρέως για την ανίχνευση της συγκέντρωσης των αερίων στην ατμόσφαιρα, λόγω του χαμηλού τους κόστους και της μεγάλης διάρκειας ζωής τους. Στην αγορά είναι διαθέσιμο μεγάλο εύρος τέτοιων αισθητήρων, προσφέροντας τη δυνατότητα ανίχνευσης πλήθους διαφορετικών αερίων (Σχήμα 38).



Σχήμα 38: Αισθητήρες αερίων MQ.

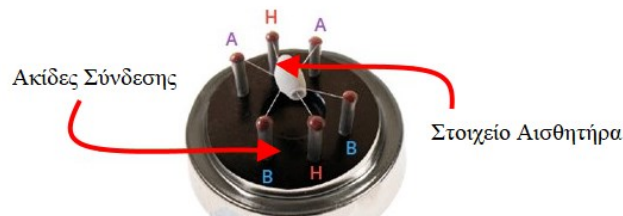
Ο όρος συγκέντρωση αερίων χρησιμοποιείται για να περιγράψει την ποσότητα του αερίου κατ' όγκο στον αέρα. Οι δυο ευρέως χρησιμοποιούμενες μονάδες μέτρησης είναι το ποσοστό συγκέντρωσης και η αναλογία της τάξης του εκατομμυριοστού ppm. Το ppm είναι η αναλογία ενός αερίου προς ένα άλλο. Για παράδειγμα, 1000 ppm CO σημαίνει ότι αν μπορούσαν να μετρηθούν ένα εκατομμύριο μόρια αερίου, τα 1000 από αυτά θα ήταν μονοξειδίου του άνθρακα και τα υπόλοιπα θα ανήκαν σε άλλα αέρια.

Οι αισθητήρες MQ διαθέτουν ενσωματωμένη μεταβλητή αντίσταση όπου η τιμή της μεταβάλλεται ανάλογα με την συγκέντρωση του αερίου. Αν η συγκέντρωση είναι υψηλή, η αντίσταση λαμβάνει μικρές τιμές, σε αντίθετη περίπτωση η τιμή της αντίστασης αυξάνεται. Εκτός από την ενσωματωμένη αντίσταση, είναι απαραίτητη η χρήση μιας εξωτερικής αντίστασης φορτίου. Η τελευταία, χρησιμεύει για την ρύθμιση της ευαισθησίας και της ακρίβειας του αισθητήρα. Όσο μεγαλύτερη είναι η τιμή της αντίστασης φορτίου, τόσο αυξάνεται η ευαισθησία του αισθητήρα. Η υψηλή τιμή της αντίστασης φορτίου επιτρέπει την μέτρηση υψηλής συγκέντρωσης αερίων με μικρότερη ακρίβεια. Για τον λόγο αυτό, η τιμή της αντίστασης αυτής απαιτεί τον πειραματικό της προσδιορισμό. Εκτός από τις αντιστάσεις που αναφέρθηκαν, ο αισθητήρας διαθέτει μια επιπλέον αντίσταση που χρησιμοποιείται ως θερμαντήρας, παρέχοντας την κατάλληλη θερμοκρασία για την εύρυθμη λειτουργία του (Σχήμα 39).



Σχήμα 39: Ηλεκτρονικό διάγραμμα αισθητήρων MQ.

Όπου A, B, H είναι οι ακίδες του αισθητήρα. Μεταξύ των ακίδων A και B βρίσκεται η αντίσταση που μεταβάλλεται ανάλογα με την συγκέντρωση του αερίου, ενώ μεταξύ των ακίδων H βρίσκεται η αντίσταση θέρμανσης (Σχήμα 40). Οι αισθητήρες της σειράς MQ διαθέτουν αναλογική έξοδο, όπου το σήμα εξόδου είναι ανάλογο της συγκέντρωσης του μετρούμενου αερίου στο χώρο που βρίσκεται εγκατεστημένος. Η τάση εξόδου του αισθητήρα κυμαίνεται από 0-5V, με τάση λειτουργίας του αισθητήρα τα 5V.



Σχήμα 40: Εσωτερική δομή αισθητήρα MQ.

8.2.1 Μέτρηση O_3 και NO_2 - MQ131

Το όζον κοντά στην επιφάνεια του εδάφους δημιουργείται από χημικές αντιδράσεις μεταξύ οξειδίων του αζώτου (NO_x) και πτητικών οργανικών ενώσεων (VOC) παρουσία ηλιακού φωτός. Η εισπνοή όζοντος μπορεί να προκαλέσει διάφορα προβλήματα υγείας, ειδικά για όσους πάσχουν από πνευμονικές ασθένειες όπως το άσθμα. Το όζον στο επίπεδο του εδάφους μπορεί επίσης να έχει επιβλαβείς επιπτώσεις στην ευαίσθητη βλάστηση και στα οικοσυστήματα.

Το διοξείδιο του αζώτου (NO_2) έχει έντονη οσμή και ένα κόκκινο, κίτρινο και καστανό χρώμα. Η παρουσία του στην ατμόσφαιρα έχει επιπλέον αποτέλεσμα τη μείωση της φωτεινότητας και συντελεί στη δημιουργία της φωτοχημικής αιθαλομίχλης. Οι κυριότερες πηγές μονοξειδίου του αζώτου (NO) στην ατμόσφαιρα είναι οι μηχανές εσωτερικής καύσης των οχημάτων, οι βιομηχανικοί καυστήρες, οι καυστήρες κεντρικής θέρμανσης, οι καυστήρες που χρησιμοποιούνται για την παραγωγή ηλεκτρικής ενέργειας και γενικά οι κλίβανοι που χρησιμοποιούν ορυκτά καύσιμα. Το NO με διάφορες χημικές αντιδράσεις που συντελούνται στην ατμόσφαιρα και ενισχύονται με την παρουσία της ηλιακής ακτινοβολίας, μετατρέπεται σε διοξείδιο του αζώτου (NO_2).

Η έκθεση μικρής χρονικής διάρκειας στο NO_2 μπορεί να οδηγήσει σε αναπνευστικές δυσλειτουργίες, ενώ η μακροχρόνια έκθεση προκαλεί ευαισθησία του αναπνευστικού συστήματος, μόνιμες αλλοιώσεις των πνευμόνων και μείωση του προσδόκιμου ζωής [33].

Ο πιο συνηθισμένος και επίσημος τρόπος μέτρησης του όζοντος της στάθμης του εδάφους είναι η φωτομετρία απορρόφησης υπεριώδους ακτινοβολίας και οι «χαμηλού κόστους» αισθητήρες όζοντος, που βασίζονται σε ηλεκτροχημική ανάλυση. Ίδιου τύπου αισθητήρες χρησιμοποιούνται για τη μέτρηση του διοξειδίου του θείου. Για την μέτρηση της συγκέντρωσης του O_3 και του NO_2 χρησιμοποιήθηκε ο αισθητήρας MQ131 (Σχήμα 41).

Το ευαίσθητο υλικό του αισθητήρα αερίων MQ131 είναι το WO_3 , το οποίο παρουσιάζει χαμηλότερη αγωγιμότητα στον καθαρό αέρα. Όταν υπάρχει παρουσία του μετρούμενου αερίου, η αγωγιμότητα του αισθητήρα μειώνεται όσο αυξάνεται η συγκέντρωση του αερίου. Ο αισθητήρας

αερίων MQ131 έχει υψηλή ευαισθησία στο όζον και παρουσιάζει επίσης ευαισθησία σε ισχυρά οξείδια όπως Cl_2 , NO_2 κ.λπ. [34]. Με τη χρήση ενός κατάλληλου κυκλώματος μπορεί να πραγματοποιηθεί η μετατροπή της συγκέντρωσης του αερίου σε αναλογική τάση και στη συνέχεια με τη χρήση ενός επεξεργαστή να αποδοθεί ως ψηφιακό δεδομένο για περαιτέρω επεξεργασία, ανάλογα με το είδος της εφαρμογής. Η λήψη μετρήσεων από τον αισθητήρα είναι σχετικά απλή διαδικασία αλλά απαιτείται βαθμονόμηση του αισθητήρα.



Σχήμα 41: Αισθητήρας όζοντος MQ131

Πίνακας 7: Τεχνικά χαρακτηριστικά MQ131

Αέριο στόχος	Όζον (O_3) Διοξείδιο του αζώτου (NO_2)
Εύρος μετρήσεων	10...1000ppm
Τάση τροφοδοσίας	$\leq 24Vdc$
Τάση θερμαντήρα	$5.0 \pm 0.2 Vdc$ ή Vac
Κατανάλωση θερμαντήρα	$\leq 900mW$
Χρόνος προθέρμανσης	Περισσότερες από 48 ώρες

8.2.2 Μέτρηση SO_2 - MQ136

Το διοξείδιο του θείου SO_2 είναι ένα αέριο άχρωμο και άοσμο σε μικρές συγκεντρώσεις αποτελώντας έναν από τους πιο χαρακτηριστικούς ρύπους που εμφανίζονται σε αστικές περιοχές. Προέρχεται κυρίως από τις καύσεις στερεών ή υγρών καυσίμων που περιέχουν θείο. Σημαντικές ανθρωπογενείς πηγές διοξειδίου του θείου στην ατμόσφαιρα είναι οι διάφορες βιομηχανικές δραστηριότητες, η παραγωγή τσιμέντου, η παραγωγή γύψου, χυτήρια μεταλλεύματος, οχήματα, αγροτικές δραστηριότητες, η δύλιση πετρελαίου και γενικά κάθε βιομηχανική κατεργασία θειούχων ενώσεων. Υπάρχουν και φυσικές πηγές παραγωγής διοξειδίου του θείου, όπως οι πυρκαγιές και η σκόνη από απογυμνωμένο έδαφος.

Η εισπνοή διοξειδίου του θείου προσβάλλει κυρίως το ανώτερο αναπνευστικό σύστημα και ένα ποσοστό του το κατώτερο. Άτομα με καρδιοαγγειακές παθήσεις ή χρόνιες παθήσεις των πνευμόνων, παιδιά και ηλικιωμένοι πρέπει να αποφεύγουν την έκθεση τους στο SO_2 [33].

Ο πιο συνηθισμένος τρόπος μέτρησης του διοξειδίου του θείου και του αζώτου είναι οι αισθητήρες που βασίζονται σε ηλεκτροχημική ανάλυση. Για την μέτρηση της συγκέντρωσης του παραπάνω αερίου χρησιμοποιήθηκε ο αισθητήρας MQ136 (Σχήμα 42).

Το ευαίσθητο υλικό του αισθητήρα αερίων MQ136 είναι το SnO₂, το οποίο παρουσιάζει χαμηλότερη αγωγιμότητα στον καθαρό αέρα. Όταν υπάρχει παρουσία του SO₂, η αγωγιμότητα του αισθητήρα είναι ανάλογη της συγκέντρωσης του αερίου. Ο αισθητήρας MQ136 έχει υψηλή ευαισθησία στο SO₂ και μπορεί να χρησιμοποιηθεί για την ανίχνευση άλλων ατμών που περιέχουν θείο [35]. Με τη χρήση ενός κατάλληλου κυκλώματος μπορεί να πραγματοποιηθεί η μετατροπή της συγκέντρωσης του αερίου σε αναλογική τάση και στη συνέχεια με τη χρήση ενός επεξεργαστή να αποδοθεί ως ψηφιακό δεδομένο για περαιτέρω επεξεργασία, ανάλογα με το είδος της εφαρμογής. Για τη λήψη μετρήσεων απαιτείται βαθμονόμηση του αισθητήρα.



Σχήμα 42: Αισθητήρας μέτρησης SO₂, NO₂ - MQ136

Πίνακας 8: Τεχνικά χαρακτηριστικά MQ136

Αέριο στόχος	Διοξείδιο του θείου (SO ₂)
Εύρος μετρήσεων	10...1000ppm
Τάση τροφοδοσίας	5.0±0.1 Vdc ή Vac
Τάση θερμαντήρα	5.0±0.1 Vdc ή Vac
Κατανάλωση θερμαντήρα	≤800mW
Χρόνος προθέρμανσης	Περισσότερες από 24 ώρες

8.3 Αισθητήρας GPS

Στην παρούσα εργασία χρησιμοποιήθηκαν αισθητήρες GPS για τον προσδιορισμό της γεωγραφικής θέσης των IoT συσκευών. Οι αισθητήρες αυτοί διασυνδέθηκαν σειριακά με τον αντίστοιχο επεξεργαστή ως μια περιφερειακή συσκευή ή χρησιμοποιήθηκαν σε συνδυασμό με άλλους αισθητήρες σε αναπτυξιακές πλακέτες που παρέχουν οι κατασκευάστριες εταιρίες, για την υποστήριξη των διαθέσιμων επεξεργαστών.

Στην ενότητα αυτή παρουσιάζεται η μονάδα GPS V.KEL VK2828U8G5LF (Σχήμα 43). Η συσκευή είναι μικρή σε μέγεθος 25 x 25 x 4 mm και διαθέτει ενσωματωμένη κεραία, καθιστώντας

εύκολη τη δημιουργία συνδέσεων με τους δορυφόρους. Σε ιδανικές συνθήκες μπορεί να συνδεθεί σε χρονικό διάστημα 45 δευτερολέπτων. Η διασύνδεση της σε επίπεδο υλικού με έναν επεξεργαστή, πραγματοποιείται με τη χρήση TTL σειριακής διεπαφής. Η σειριακή επικοινωνία σε επίπεδο TTL παραμένει πάντα μεταξύ των ορίων των 0V και Vcc, που συχνά είναι 5V ή 3,3V. Ένα λογικό υψηλό ('1') αντιπροσωπεύεται από την τάση Vcc, ενώ ένα λογικό χαμηλό ('0') είναι 0V. Από τη συσκευή μπορούν να ανακτηθούν πληροφορίες όπως γεωγραφικό μήκος και πλάτος, υψόμετρο και ώρα GMT [36].



Σχήμα 43: Μονάδα GPS VK2828U8G5LF

Η μονάδα GPS χρησιμοποιεί το σειριακό πρωτόκολλο ανταλλαγής μηνυμάτων NMEA 0183. Τα δεδομένα σε αυτό το πρωτόκολλο μεταδίδονται σε συμβολοσειρές ή ακολουθία χαρακτήρων ASCII από τον «ομιλητή» προς τους «ακροατές». Ένα άλλο χαρακτηριστικό του πρωτοκόλλου NMEA 0183 είναι ότι χρησιμοποιεί τα ηλεκτρικά πρότυπα RS-422 και RS232 [37]

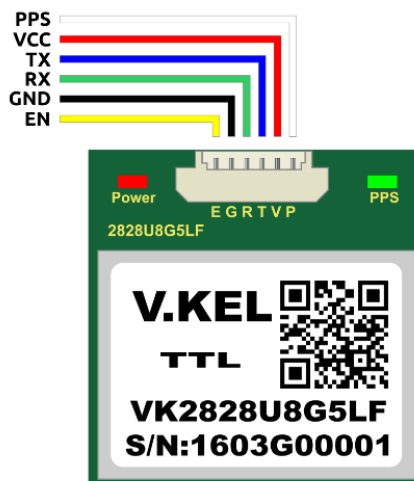
Μερικές φορές, το πρωτόκολλο ανταλλαγής μηνυμάτων NMEA 0183 προκαλεί σύγχυση επειδή δεν υπάρχει μόνο μία «πρόταση». Υπάρχουν διαφορετικές προτάσεις NMEA 0183 με διαφορετικές δυνατότητες και σκοπούς, συνήθως αυτό που αλλάζει στην πρόταση είναι οι πληροφορίες που μπορούν να παρέχουν. Για την κατανόηση των διαφορετικών δυνατοτήτων, τους σκοπούς και τις παρεχόμενες πληροφορίες αυτών των προτάσεων παρατίθεται συγκεντρωτικός πίνακας (Πίνακας 9).

Πίνακας 9: Προτάσεις πρωτοκόλλου NMEA 0183.

A/A	Πρόταση	Περιγραφή
1	\$GPGGA	Δεδομένα χρόνου, θέσης.
2	\$GPGLL	Γεωγραφικό πλάτος, γεωγραφικό μήκος, ώρα UTC καθορισμού θέσης και κατάστασης.
3	\$GPGSA	Τρόπος λειτουργίας δέκτη GPS, δορυφόροι που χρησιμοποιούνται στη θέση, τιμές DOP.

4	\$GPGSV	Αριθμός δορυφόρων που προβάλλονται, αριθμοί ταυτότητας δορυφόρου, υψόμετρο, αζιμούθιο, τιμές SNR.
5	\$GPRMC	Δεδομένα ώρας, ημερομηνίας, θέσης, πορείας, ταχύτητας.
6	\$GPVTG	Πορεία, πληροφορίες ταχύτητας σε σχέση με το έδαφος.

Η ηλεκτρονική διασύνδεση της συσκευής VK2828U8G5LF παρουσιάζεται στο (Σχήμα 44) και η αντιστοίχιση των ακίδων στον Πίνακα 10.



Σχήμα 44: Διάγραμμα διασύνδεσης VK2828U8G5LF.

Πίνακας 10: Πίνακας αντιστοίχισης ακίδων VK2828U8G5LF.

A/A	Ακίδα	Χρώμα	Περιγραφή
1	PSS	Λευκό	Τυπική χρονική έξοδος παλμού
2	VCC	Κόκκινο	Είσοδος τροφοδοσίας 3.3V-5V
3	TX	Μπλε	UART/TTL
4	RX	Πράσινο	UART/TTL
5	GND	Μαύρο	Γείωση
6	EN	Κίτρινο	Ακίδα ενεργοποίησης. Η συσκευή βρίσκεται σε κατάσταση τερματισμού λειτουργίας όταν η τάση στην ακίδα είναι σε χαμηλό δυναμικό και ενεργοποιείται όταν είναι σε υψηλό ή ασύνδετη.

Ακολουθεί πίνακας με τα τεχνικά χαρακτηριστικά της συσκευής (Πίνακας 11).

Πίνακας 11: Τεχνικά Χαρακτηριστικά VK2828U8G5LF.

A/A	Χαρακτηριστικό	Περιγραφή
1	Ολοκληρωμένο κύκλωμα	UBX-M8030-KT
2	Κανάλια λήψης	56
3	Συχνότητα λήψης	L 1 [1575.42MHz]
4	Ταχύτητα	<0.1 m/s
5	Διεύθυνση	< 0.5 μοίρες
6	Ακρίβεια χρονισμού	30ns
7	Μέγιστο ύψος	50 Km
8	Μέγιστη ταχύτητα	500 m/s
9	Επιτάχυνση	≤4G
10	Σύστημα συντεταγμένων	WGS-84
11	Πρωτόκολλο εξόδου	NMEA 0183 V3.0(GGA, GSA, GSV, RMC, VTG, GLL)
12	Ρυθμός μετάδοσης	9600bps(προεπιλογή) [1200,2400,4800, 19200, 38400, 57600, 115200, 230400,460800,921600]
13	Συχνότητα ενημέρωσης	1-10 Hz
14	UART/TTL	NAI
15	RS232	Προαιρετικό
16	Τάση εισόδου	3.3V – 5V
17	Κατανάλωση	30 mA @ 5V

9

Μελέτη περίπτωσης «Πράσινη Διαδρομή» βασισμένη στην γενική αρχιτεκτονική

9.1 Επίπεδο πηγής IoT δεδομένων

Στα πλαίσια υλοποίησης της εφαρμογής «Ανίχνευση Πράσινης Διαδρομής», χρησιμοποιήθηκαν 9 IoT κόμβοι οι οποίοι χαρακτηρίζονται από ετερογένεια σε επίπεδο υλικού, δικτύωσης και πρωτοκόλλου επικοινωνίας. Όλοι οι IoT σταθμοί περιλαμβάνουν τους ίδιους περιβαλλοντικούς και γεωχωρικούς αισθητήρες που αναφέρθηκαν στην ενότητα 8. Οι σταθμοί διαχωρίστηκαν σε δυο ομάδες οι οποίες αποστέλλουν τα συλλεχθέντα δεδομένα σε δυο διαφορετικές πλατφόρμες όπως παρουσιάζονται στον Πίνακα 12.

Πίνακας 12: IoT Κόμβοι.

A/A	Όνομα	Υλικό	Δίκτυο	Πρωτόκολλο	IoT Πλατφόρμα
1	AQI_Ethernet_Node1	Arduino Mega 2560	Ethernet	HTTP	Thingspeak
2	Node_1_Air_Quality	Arduino UNO WIFI	WiFi	HTTP	Thingspeak

3	Node_2_AQI_MQTT	Arduino UNO WIFI	WiFi	MQTT	Thingspeak
4	Node_1_Lorawan_AQI	TTGO LoRa32 v1.0	LoRa	LoRaWAN	Thingspeak
5	Node_2_Lorawan_AQI	TTGO LoRa32 v1.0	LoRa	LoRaWAN	Thingspeak
6	arduino_wifi_mqtt	Arduino UNO WIFI	WiFi	MQTT	Ubidots
7	arduino_eth_http	Arduino Mega 2560	Ethernet	HTTP	Ubidots
8	aegean-lopy4-node1	Lopy4	LoRa	LoRaWAN	Ubidots
9	aegean-lopy4-node2	Lopy4	LoRa	LoRaWAN	Ubidots

9.1.1 IoT Κόμβοι

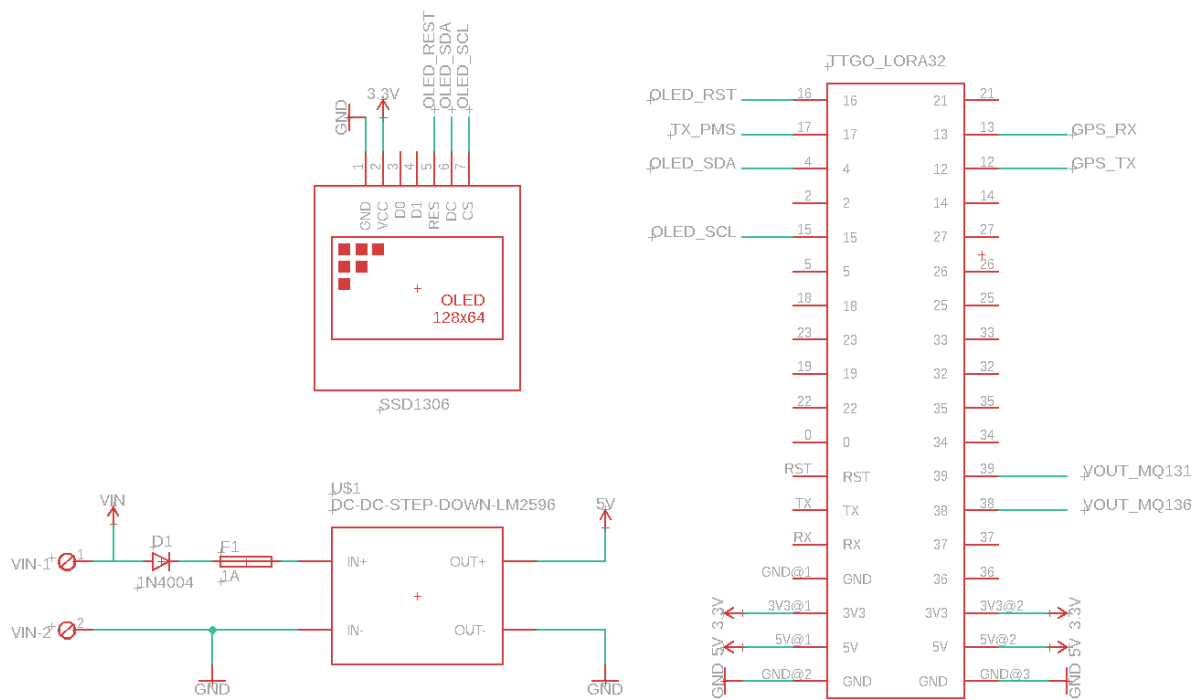
Για την υλοποίηση των IoT κόμβων χρησιμοποιήθηκαν τα ακόλουθα υλικά:

- Αναπτυξιακή πλακέτα μικροελεγκτή (Ενότητα 4)
- Αισθητήρες αερίων: MQ136, MQ131.
- Αισθητήρας αιωρούμενων σωματιδίων PMS5003.
- Διάτρητη πλακέτα πρωτοτύπου.
- Κλέμα σύνδεσης τροφοδοσίας.
- Ασφάλεια TR5/1A.
- Δίοδος 1N4007.
- Πλακέτα υποβάθμισης συνεχούς τάσης LM2596.

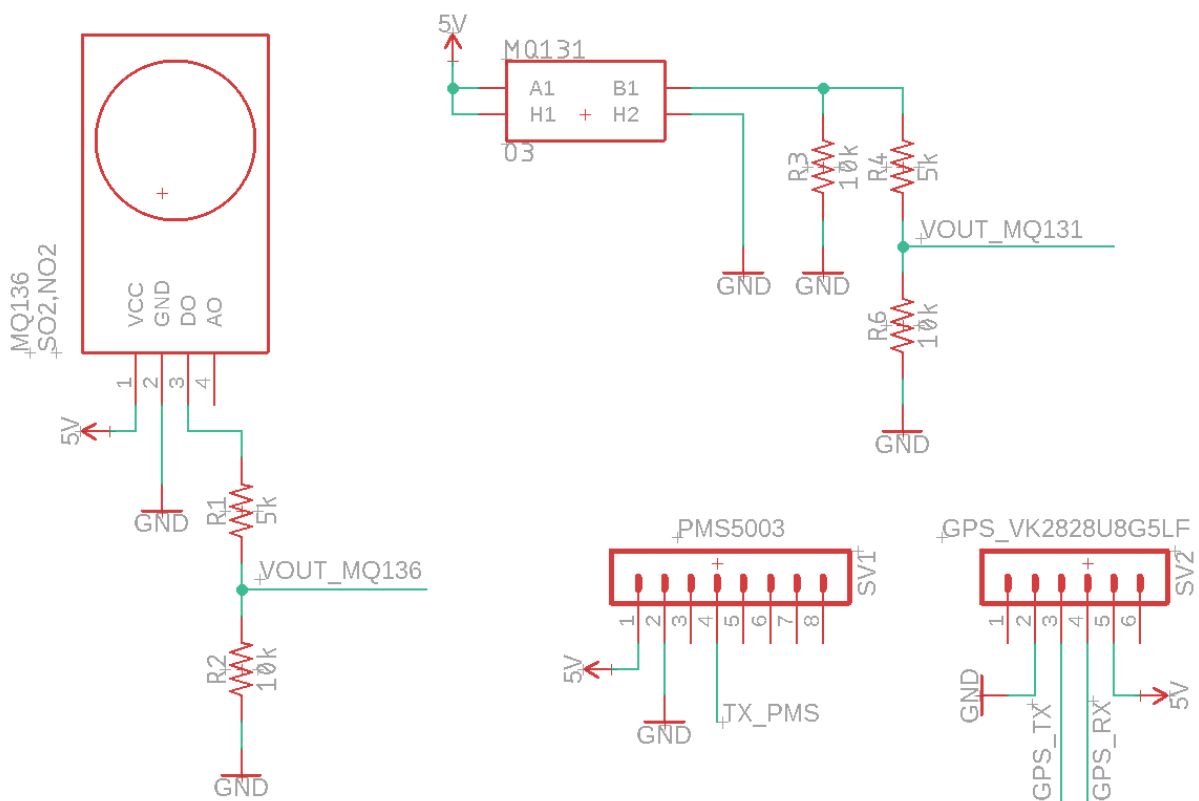
Όλα τα εξαρτήματα του υλικού που αναφέρθηκαν, τοποθετήθηκαν σε μια διάτρητη πλακέτα βακελίτη για την ανάπτυξη του πρωτότυπου IoT κόμβου, σύμφωνα με τα ακόλουθα σχηματικά διαγράμματα (Σχήμα 45, Σχήμα 46). Τα σχηματικά διαγράμματα που παρουσιάζονται αποτελούν τυπικό παράδειγμα διασύνδεσης της αναπτυξιακής πλακέτας TTGO LoRa32 v1.0.

Οι αναπτυξιακές πλακέτες TTGO LoRa32 v1.0 και Lopy4 απαιτούν δυο διαφορετικές τάσεις, 3.3V για την λειτουργία του επεξεργαστή και της OLED οθόνης (στο TTGO LoRa32 v1.0), ενώ τα 5V χρειάζονται για την λειτουργία όλων των αισθητήρων. Οι συσκευές τροφοδοτούνται από εξωτερική πηγή, με τάση 12V/2A. Η τάση εισόδου οδηγείται σε έναν υποβαθμιστή για την δημιουργία των 5V, διαμέσου μιας διόδου και μιας ασφάλειας 1A. Η δίοδος χρησιμοποιείται για την αποτροπή λειτουργίας της συσκευής με ανάστροφη πολικότητα στην είσοδο του κυκλώματος, γεγονός που θα επέφερε δυσάρεστα αποτελέσματα για τα εξαρτήματα που απαρτίζουν την συσκευή. Η τάση 3.3V παράγεται από έναν ρυθμιστή τάσης που βρίσκεται ενσωματωμένος στην αναπτυξιακή

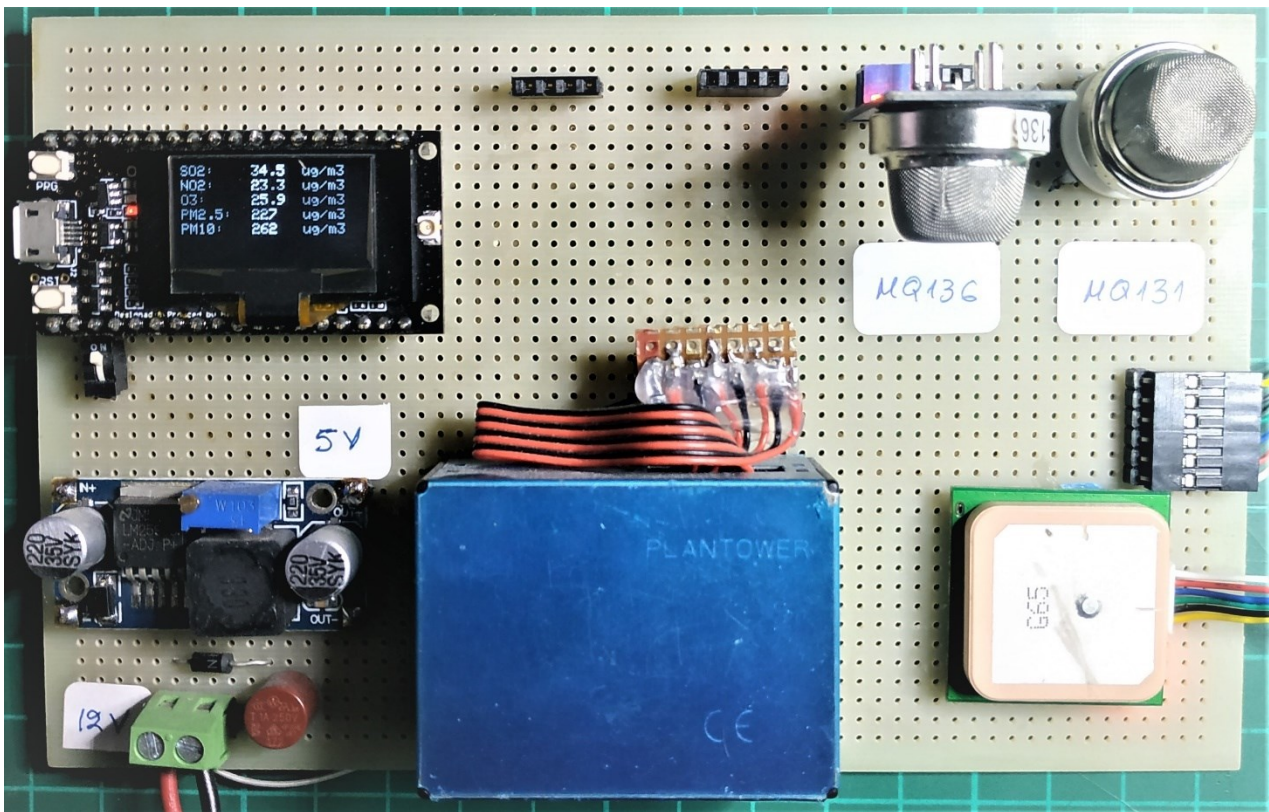
πλακέτα του επεξεργαστή, λαμβάνοντας ως τάση εισόδου 5V. Η τελική μορφή ενός ενδεικτικού IoT κόμβου παρουσιάζεται στο Σχήμα 47.



Σχήμα 45: Σχηματικό διάγραμμα συσκευής (Τροφοδοτικό, αναπτυξιακή πλακέτα).



Σχήμα 46: Σχηματικό διάγραμμα αισθητήρων.



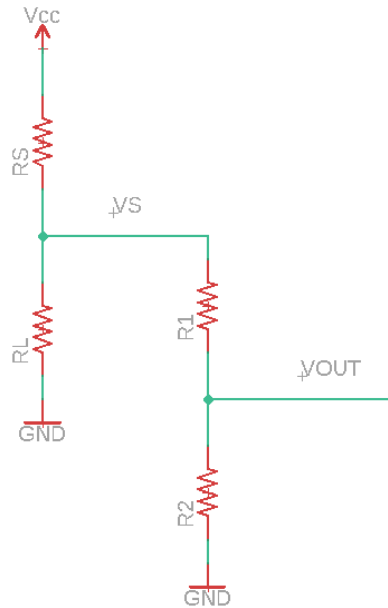
Σχήμα 47: Πρωτότυπο IoT κόμβου.

9.1.2 Βαθμονόμηση αισθητήρων αερίων MQ

Οι αισθητήρες αερίων MQ χρειάζονται βαθμονόμηση για τον ακριβή υπολογισμό της συγκέντρωσης των αερίων στην ατμόσφαιρα. Η διαδικασία αυτή έχει αναλυθεί σε σχετικές έρευνες χωρίς όμως να γίνεται λεπτομερής περιγραφή των βημάτων που θα πρέπει να ακολουθηθούν [38]–[41]. Στην συνέχεια, παρουσιάζεται ένας αναλυτικός οδηγός βαθμονόμησης όπως προέκυψε από τη μελέτη υφιστάμενων εργασιών και προσωπικής προσέγγισης της διαδικασίας.

Η αναλογική έξοδος του αισθητήρα έχει εύρος 0-5V, ενώ οι περισσότεροι μικροελεγκτές, δέχονται αναλογικό σήμα εισόδου 0-3.3V. Η διαφορά τάσης μεταξύ των δυο συσκευών καθιστά αδύνατη την διασύνδεσή τους. Για το λόγο αυτό απαιτείται η χρήση ενός κυκλώματος προσαρμογής που θα επιτρέπει την δειγματοληψία του σήματος από τον επεξεργαστή.

Το κύκλωμα προσαρμογής (Σχήμα 48) που σχεδιάστηκε, είναι ένας διαιρέτης τάσης όπου τοποθετήθηκε στην έξοδο κάθε αισθητήρα αερίου MQ που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής. Το εν λόγω κύκλωμα μετατρέπει το εύρος τάσης 0-5V του αισθητήρα, σε 0-3.3V. Τα υλικά που απαρτίζουν το κύκλωμα επεξηγούνται αναλυτικά στον πίνακα 3.



Σχήμα 48: Διάγραμμα κυκλώματος προσαρμογής τάσης για τον αισθητήρα MQ.

Πίνακας 13 : Περιγραφή στοιχείων κυκλώματος προσαρμογής.

A/A	Όνομα Στοιχείου	Περιγραφή
1	RS	Αντίσταση Αισθητήρα. Μεταβάλλεται ανάλογα με την συγκέντρωση του αερίου που εξετάζουμε.
2	RL	Αντίσταση φορτίου. Ρυθμίζει την ευαισθησία και την ακρίβεια του αισθητήρα. Στο datasheet αναφέρεται το εύρος τιμών της.
3	R1	Η τιμή της αντίστασης είναι 5KΩ.
4	R2	Η τιμή της αντίστασης είναι 10KΩ.
5	V _{cc}	Τάση λειτουργίας αισθητήρα, 5V.
6	V _s	Αναλογική τάση εξόδου αισθητήρα. Εύρος τιμών 0-5V.
7	V _{out}	Τάση εξόδου κυκλώματος προσαρμογής. Αναλογική Είσοδος επεξεργαστή.

Οι αναλογικές είσοδοι των ευρέως διαδεδομένων επεξεργαστών π.χ. ESP32, έχουν ανάλυση 12 bit. Αυτό σημαίνει ότι η τιμή που μετράει ο επεξεργαστής (*Raw Value*), έχει εύρος τιμών 0-4095. Το εύρος τιμών μπορεί να διαφέρει ανά επεξεργαστή, γι' αυτό προτείνεται η αναζήτησή τους από το αντίστοιχο τεχνικό φυλλάδιο που προσφέρει ο κατασκευαστής. Από τα παραπάνω προκύπτει η

σχέση (1), που συνδέει την τάση εξόδου του κυκλώματος προσαρμογής με την μετρούμενη τιμή του επεξεργαστή.

$$V_{out} = \frac{3.3 * Raw_Value}{4095} \quad (1)$$

Από τον διαιρέτη τάσης των αντιστάσεων R_1 και R_2 προκύπτει η σχέση (2). Γνωρίζοντας τις τιμές τους και από την σχέση (1), μπορεί να υπολογιστεί η τάση εξόδου του αισθητήρα V_s .

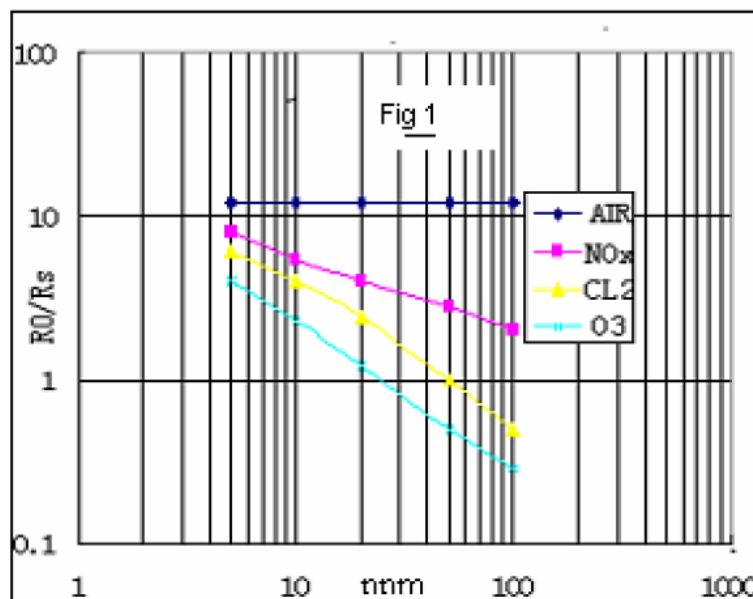
$$V_s = \frac{R_1 + R_2}{R_2} * V_{out} \quad (2)$$

Από την σχέση (2) και την τάση εξόδου του αισθητήρα, όπως προκύπτει από τον διαιρέτη τάσης των αντιστάσεων R_s και R_L , μπορεί να υπολογιστεί η αντίσταση του αισθητήρα R_s (Σχέση 3).

$$R_s = \frac{(V_{CC} - V_s) * R_L}{V_s} \quad (3)$$

Από τις σχέσεις (1), (2), (3) μπορεί να γίνει γνωστή, οποιαδήποτε στιγμή η τρέχουσα τιμή της αντίστασης του αισθητήρα.

Στα εγχειρίδια των αισθητήρων αερίων MQ περιλαμβάνεται το διάγραμμα ευαισθησίας τους (Σχήμα 49), το οποίο απεικονίζει τη γραφική συσχέτιση του λόγου των αντιστάσεων R_s/R_o και της συγκέντρωσης των μετρούμενων αερίων του αισθητήρα σε ppm.



Σχήμα 49: Διάγραμμα ευαισθησίας αισθητήρα MQ131.

Όπου R_s είναι η τιμή της αντίστασης του αισθητήρα που μεταβάλλεται ανάλογα με την συγκέντρωση του προς εξέταση αερίου και R_o η αντίσταση του αισθητήρα στον καθαρό αέρα. Το διάγραμμα απεικονίζει την γραμμική μεταβολή της ευαισθησίας του αισθητήρα σε κάθε αέριο με τους άξονες R_s/R_o και ppm να είναι σε λογαριθμική κλίμακα. Συμπερασματικά προκύπτει ότι η

γραφική παράσταση της ευαισθησίας του αισθητήρα για κάθε αέριο είναι μια εκθετική συνάρτηση της μορφής (Σχέση 4).

$$y = ax^b, \text{ με } b < 0 \quad (4)$$

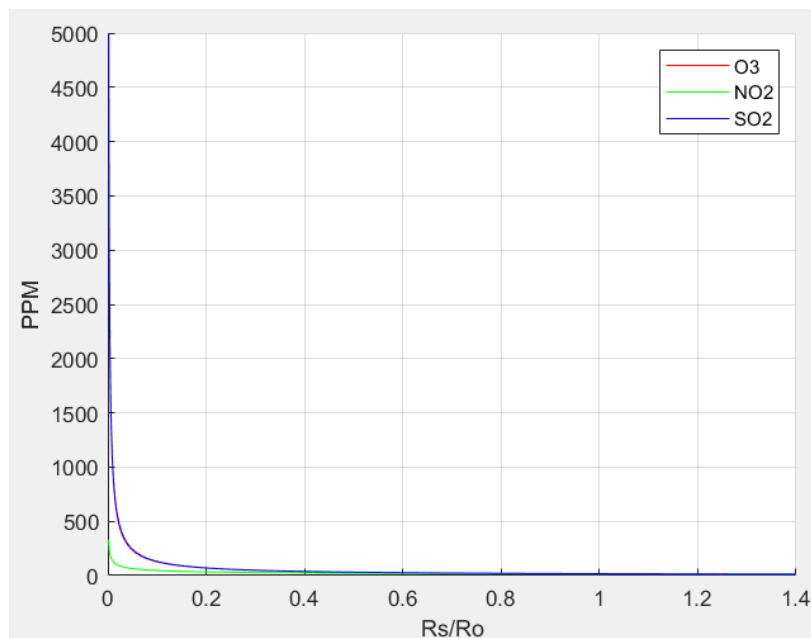
Σύμφωνα με τα μεγέθη του γραφήματος, η σχέση (4) γίνεται:

$$ppm = a \left(\frac{R_s}{R_o} \right)^b \quad (5)$$

Λύνοντας την σχέση (5) ως προς R_o (Σχέση 6), μπορούμε να υπολογίσουμε την αντίσταση που έχει ο αισθητήρας στον καθαρό αέρα, γνωρίζοντας την συγκέντρωση του αερίου σε ppm την στιγμή της μέτρησης.

$$R_o = R_s \left(\frac{a}{ppm} \right)^{1/b} \quad (6)$$

Κάνοντας χρήση του λογισμικού *WebplotDigitizer* [42], εξήχθησαν διαδοχικά σημεία τιμών (ppm, R_s/R_o) από το διάγραμμα ευαισθησίας, που συνθέτουν την γραφική παράσταση της συγκέντρωσης του αερίου που θα μετρηθεί. Η διαδικασία αυτή επαναλήφθηκε για όλα τα αέρια που επιλέχθηκαν, από όλους τους αισθητήρες. Στη συνέχεια οι συντεταγμένες που προέκυψαν από το λογισμικό εισήχθησαν στο MATLAB και μέσω ενός προγράμματος που συντάχθηκε, υπολογίστηκε με power regression η γραφική παράσταση που ταιριάζει στα δεδομένα αυτά (Σχήμα 50). Παρατηρείται ότι η γραφική παράσταση του όζοντος και του διοξειδίου του θείου είναι σχεδόν όμοιες. Με αυτόν τον τρόπο υπολογίστηκαν οι συντελεστές a , b της σχέσης (5), για κάθε αέριο (Πίνακας 4).



Σχήμα 50: Χαρακτηριστικές συγκέντρωσης αερίων με power regression.

Πίνακας 14: Συντελεστές χαρακτηριστικών συγκέντρωσης αερίων.

A/A	Αέριο	Συντελεστής α	Συντελεστής b
1	O ₃	16.69	-0.8811
2	NO ₂	16.21	-0.4586
3	SO ₂	17.02	-0.8847

Ο δόκιμος τρόπος βαθμονόμησης των αισθητήρων είναι η τοποθέτηση τους σε ελεγχόμενο περιβάλλον, γνωρίζοντας εκ των προτέρων το είδος και τη συγκέντρωση του ελεγχόμενου αερίου (ppm) που εμπεριέχεται σε αυτό [43], [44]. Υπολογίζοντας τον μέσο όρο διαδοχικών μετρήσεων της αντίστασης R_s , μπορεί να υπολογιστεί μέσω της σχέσης 6, η αντίσταση βαθμονόμησης R_o . Η διαδικασία αυτή απαιτεί εξοπλισμού υψηλού κόστους που δεν ήταν διαθέσιμος για την υλοποίηση αυτής της μελέτης.

Εναλλακτικά, η αντίσταση μπορεί να υπολογιστεί έμμεσα από το διάγραμμα ευαισθησίας του αισθητήρα, από τον λόγο R_s/R_o στον καθαρό αέρα. Αρχικά υπολογίζεται ο μέσος όρος διαδοχικών μετρήσεων της αντίστασης R_s και στην συνέχεια από την τιμή του λόγου R_s/R_o , υπολογίζεται η τιμή της αντίστασης βαθμονόμησης R_o (Σχέση 7).

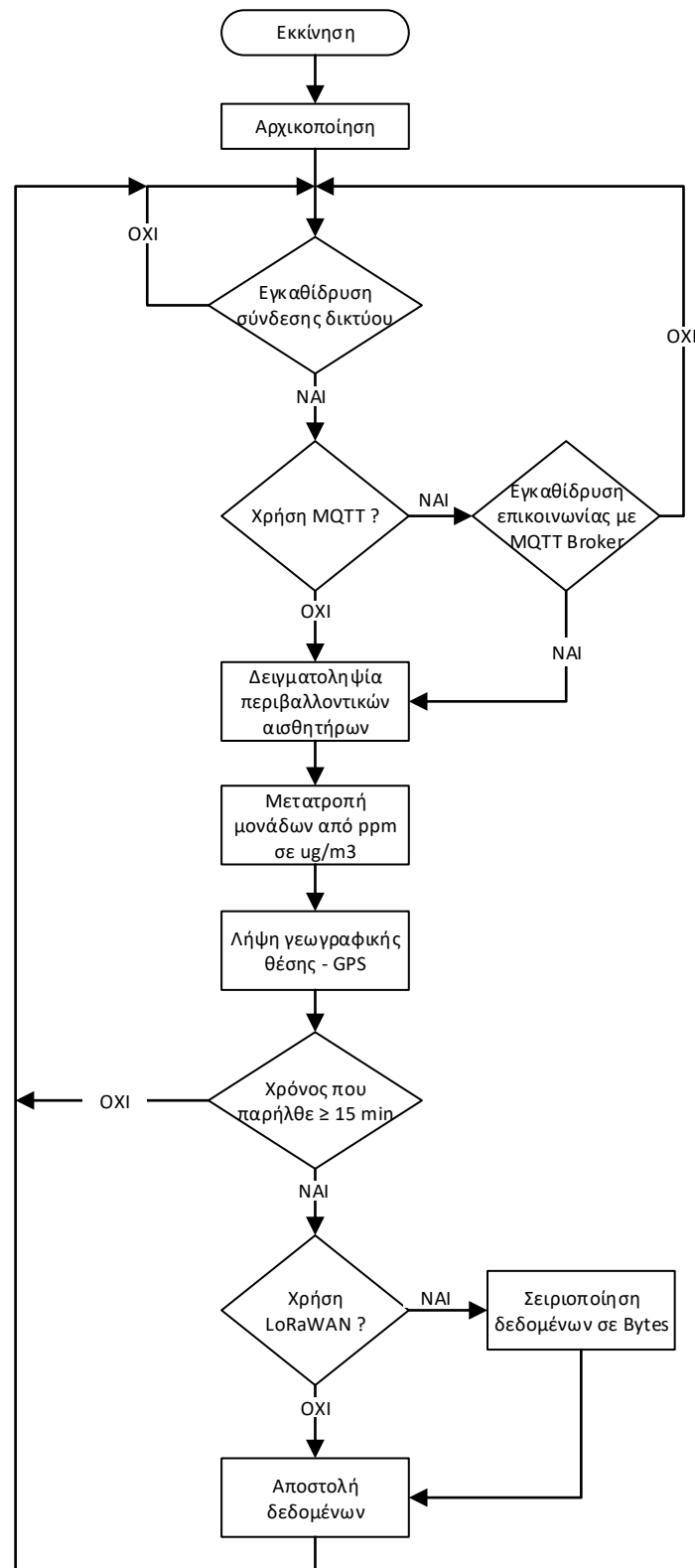
$$R_o = R_{S (air)} / Ratio_{(clean air)} \quad (7)$$

9.1.3 Περιγραφή λειτουργίας IoT κόμβων

Το πρόγραμμα το οποίο εκτελείται σε κάθε μια IoT συσκευή του συστήματος (Πίνακας 12), πραγματοποιεί τρεις διαδοχικές λειτουργίες οι οποίες απεικονίζονται στο ακόλουθο διάγραμμα ροής (Σχήμα 51). Η πρώτη, αφορά την εγκαθίδρυση σύνδεσης στο εκάστοτε δίκτυο WiFi, Ethernet, LoRaWAN και τη διατήρηση της συνδεσιμότητας σε αυτό. Στη λειτουργία αυτή προστίθεται και η εγκαθίδρυση σύνδεσης με τον MQTT Broker των IoT πλατφορμών, για όσους κόμβους χρησιμοποιούν το πρωτόκολλο MQTT. Η δεύτερη λειτουργία αφορά τη δειγματοληψία των περιβαλλοντικών αισθητήρων για τον προσδιορισμό της τρέχουσας συγκέντρωσης των αερίων O₃, NO₂, SO₂ και των αιωρούμενων σωματιδίων μεγέθους PM2.5 και PM10 στον ατμοσφαιρικό αέρα. Οι συγκεντρώσεις των ρύπων μετατρέπονται στις κατάλληλες μονάδες $\mu\text{g}/\text{m}^3$, σύμφωνα με τις υποδείξεις για τον υπολογισμό του Ευρωπαϊκού δείκτη ποιότητας αέρα. Στην λειτουργία αυτή περιλαμβάνεται και η επικοινωνία με τον αισθητήρα GPS, για την ανάκτηση του γεωγραφικού μήκους και πλάτους της τρέχουσας θέσης του IoT κόμβου. Η τρίτη λειτουργία αφορά τη συγκέντρωση των συλλεχθέντων δεδομένων, την διαμόρφωσή τους στην κατάλληλη μορφή ανάλογα με το είδος του πρωτοκόλλου και την αποστολή τους στις IoT πλατφόρμες ή στο διακομιστή του δικτύου.

Αναλυτικότερα, το πρόγραμμα αρχικοποιεί τις εκάστοτε εισόδους/εξόδους του επεξεργαστή, ορίζοντας τα pins των αναλογικών εισόδων στις οποίες είναι συνδεδεμένοι οι αισθητήρες αερίων και των ψηφιακών για τη σειριακή επικοινωνία με τον αισθητήρα αιωρούμενων σωματιδίων και το GPS. Ορίζονται οι ρυθμοί μετάδοσης της σειριακής επικοινωνίας του επεξεργαστή για την απεικόνιση των μηνυμάτων στην κονσόλα του χρήστη και την επικοινωνία με τον αισθητήρα

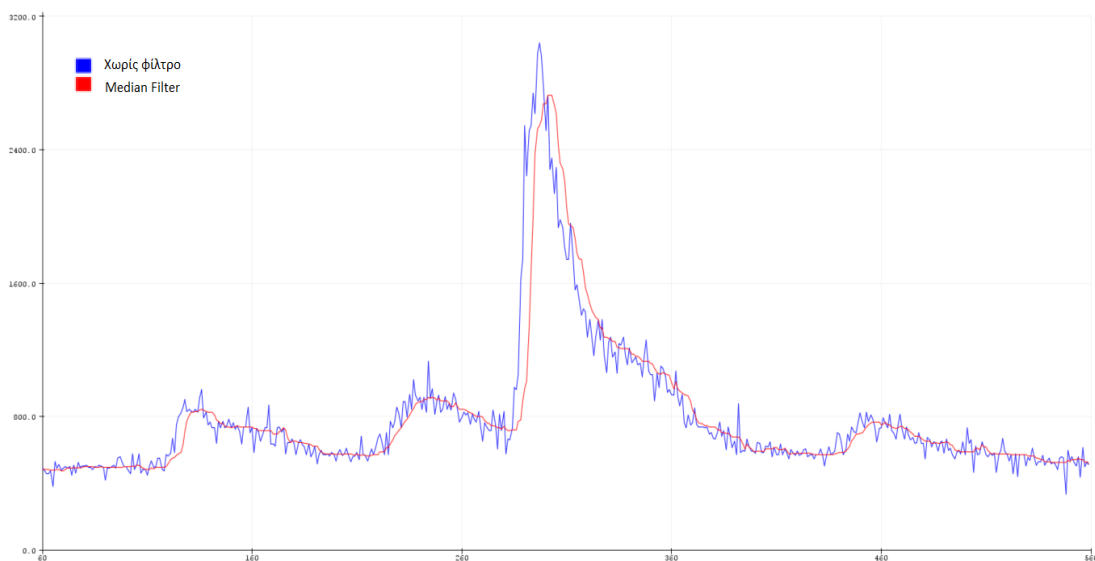
αιωρούμενων σωματιδίων PMS5003. Στην περίπτωση της αναπτυξιακής πλακέτας TTGO LoRa32 v1.0, το στάδιο αυτό περιλαμβάνει και την αρχικοποίηση του πρωτοκόλλου I²C για την επικοινωνία με την οθόνη OLED.



Σχήμα 51: Διάγραμμα ροής λειτουργίας IoT κόμβων.

Κατά την εκκίνηση του κύκλου λειτουργίας του κυρίου προγράμματος ελέγχεται εάν η σύνδεση με το δίκτυο είναι ενεργή, διαφορετικά πραγματοποιείται προσπάθεια επανασύνδεσης μέχρι αυτή να αποκατασταθεί. Εάν ο κόμβος χρησιμοποιεί το πρωτόκολλο επικοινωνίας MQTT, πραγματοποιείται αίτηση σύνδεσης με τον MQTT Broker. Αν η σύνδεση με τον Broker αποκατασταθεί, το πρόγραμμα συνεχίζει στο βήμα της δειγματοληψίας των αισθητήρων, διαφορετικά επιστρέφει στον έλεγχο εγκαθίδρυσης σύνδεσης στο δίκτυο.

Στη συνέχεια, υπολογίζονται οι συγκεντρώσεις των αερίων (ενότητα 9.1.2) και των αιωρούμενων σωματιδίων που υπάρχουν στην ατμόσφαιρα. Στο σημείο αυτό παρατηρήθηκε ότι οι μετρήσεις των αισθητήρων MQ ήταν εξαιρετικά θορυβώδεις με πολύ απότομες μεταβολές στην τιμή τους. Επιλέχθηκε η χρήση ενός median φίλτρου για την αντιμετώπιση αυτού του προβλήματος (Σχήμα 52). Κάθε νέα μέτρηση προστίθεται σε έναν μονοδιάστατο πίνακα μήκους, όσο και το παράθυρο του median φίλτρου που ορίστηκε στην αρχή του κώδικα. Αν ο πίνακας εμπεριέχει παλαιότερες μετρήσεις, τότε αντικαθίσταται η παλαιότερη μέτρηση με την νεότερη. Στην συνέχεια, τα στοιχεία ταξινομούνται κατά αύξουσα τιμή και υπολογίζεται η μεσαία τιμή από τον σύνολο των τιμών που περιέχονται στον πίνακα. Επειδή ο δείκτης ποιότητας αέρα που θέλουμε να μετρήσουμε λαμβάνει υπόψη τις συγκεντρώσεις των ρύπων σε μονάδες $\mu\text{g}/\text{m}^3$, οι συγκεντρώσεις των αερίων μετατρέπονται από ppm σε $\mu\text{g}/\text{m}^3$ [45], [46]. Το στάδιο αυτό ολοκληρώνεται με την ανάκτηση του γεωγραφικού μήκους και πλάτους της τρέχουσας θέσης του IoT κόμβου, από τον αισθητήρα GPS.



Σχήμα 52: Εφαρμογή median φίλτρου σε μετρήσεις αισθητήρα MQ.

Σε κάθε κύκλο του προγράμματος υπολογίζονται τα msec που παρήλθαν από την στιγμή εκκίνησης της συσκευής. Η μέτρηση αυτή συγκρίνεται με την προκαθορισμένη περίοδο αποστολής των δεδομένων στις IoT πλατφόρμες ή στον διακομιστή δικτύου στην περίπτωση του LoRaWAN. Αν ο χρόνος που παρήλθε είναι μεγαλύτερος ή ίσος της προκαθορισμένης περιόδου τότε η συσκευή αποστέλλει τα δεδομένα που συνέλλεξε. Διαφορετικά εκτελείται ένας νέος κύκλος του προγράμματος μέχρι να ικανοποιηθεί η συνθήκη. Στα πλαίσια της εφαρμογής «Πράσινη

Διαδρομή», η συχνότητα αποστολής των δεδομένων ορίστηκε ως ένα μήνυμα ανά δεκαπέντε (15) λεπτά.

Στο τελευταίο στάδιο και μετά τον έλεγχο της χρονικής διάρκειας των δεκαπέντε λεπτών, τα δεδομένα συγκεντρώνονται, διαμορφώνονται κατάλληλα, όποτε αυτό απαιτείται και αποστέλλονται είτε στις IoT πλατφόρμες, είτε στον διακομιστή δικτύου. Στην περίπτωση χρήσης του πρωτοκόλλου HTTP οι τιμές των μετρούμενων μεγεθών συλλέγονται και διαμορφώνονται σε ένα POST αίτημα (Σχήμα 53).

```
POST /update HTTP/1.1
api_key=api&field1=pm2_5&field2=pm10&field3=ug_m3_SO2&field4=ug_m3_NO2&
field5=ug_m3_O3&field6=lat&field7=lng
Content-Type: application/x-www-form-urlencoded
```

Σχήμα 53: Υποβολή δεδομένων στο Thingspeak με HTTP POST αίτημα.

Στην περίπτωση χρήσης του πρωτοκόλλου MQTT, τα μετρούμενα μεγέθη διαμορφώνονται ως ένα JSON (Σχήμα 54) αντικείμενο που αποτελεί την ωφέλιμη πληροφορία (payload) του μηνύματος που θα αποσταλεί στην αντίστοιχη πλατφόρμα. Προϋπόθεση στη χρήση των δυο παραπάνω πρωτοκόλλων αποτελεί η εκ των προτέρων γνώση της αντιστοιχίας των μετρούμενων μεγεθών με τα πεδία ή τις μεταβλητές αποθήκευσής τους στην πλατφόρμα.

```
{
  "field1": "496.00",
  "field2": "1061.00",
  "field3": "783.00",
  "field4": "843.00",
  "field5": "428.00",
  "field6": "36.960762",
  "field7": "26.963938"
}
```

Σχήμα 54: MQTT payload.

Για τις συσκευές IoT που χρησιμοποιούν το πρωτόκολλο LoRaWAN και η δικτύωση υλοποιείται μέσω του The Things Network, οι τιμές των μετρούμενων μεγεθών δεν μπορούν να αποσταλούν με τη μορφή κειμένου αλλά απαιτείται η κωδικοποίησή τους σε bytes [47]. Η διαδικασία κωδικοποίησης περιλαμβάνει τα ακόλουθα στάδια:

1. Ελέγχεται ο τύπος της μεταβλητής. Εάν ο τύπος είναι FLOAT τότε η τιμή της μεταβλητής πολλαπλασιάζεται με τον αριθμό 10000000 για την απαλοιφή του δεκαδικού μέρους και τη διατήρηση της ακρίβειας του αριθμού.
2. Υπολογίζεται το μέγεθος σε bytes της μεταβλητής.
3. Η τιμή μετατρέπεται σε δυαδική μορφή και υπόκεινται σε δυαδική πράξη AND με τον αριθμό 0xFF. Στη συνέχεια ο αρχικός αριθμός ολισθαίνει προς τα δεξιά κατά 8 bits, για την επόμενη

δυναδική πράξη AND. Η διαδικασία επαναλαμβάνεται τόσες φορές όσες είναι και το μέγεθος της μεταβλητής.

Το αποτέλεσμα της κωδικοποίησης είναι η μετατροπή της τρέχουσας τιμής της μεταβλητής σε bytes με δεκαεξαδική αναπαράσταση, από το ελάχιστο σημαντικό byte στο πιο σημαντικό. Η διαδικασία αυτή ακολουθείται για κάθε μετρούμενο μέγεθος και τα παραγόμενα bytes διατάσσονται διαδοχικά όπως παρουσιάζονται στο ακόλουθο παράδειγμα (Σχήμα 55). Η σειρά διάταξης των μεταβλητών είναι σημαντική και πρέπει να καταγράφεται για την υλοποίηση της αντίστροφης διαδικασίας. Το τελευταίο στάδιο του κυρίως προγράμματος ολοκληρώνεται με την αποστολή των δεδομένων και την αρχικοποίηση της μέτρησης χρόνου για την καταμέτρηση της νέας περιόδου των 15 λεπτών.

```
PM2.5: 176 --> Encoded:bytes:0 B0
PM10: 735 --> Encoded:bytes:2 DF
ug_m3_SO2: 168 --> Encoded:bytes:0 A8
ug_m3_NO2: 901 --> Encoded:bytes:3 85
ug_m3_O3: 621 --> Encoded:bytes:2 6D
Lat: 36.957470 --> Encoded:bytes:16 7 43 20
Long: 26.968935 --> Encoded:bytes:10 13 22 0
-----Combine-----
bytes:10 13 22 0 16 7 43 20 2 6D 3 85 0 A8 2 DF 0 B0
```

Σχήμα 55: Παράδειγμα κωδικοποίησης μετρούμενων μεγεθών.

9.2 Επίπεδο πηγής ανοικτών δεδομένων

Στο επίπεδο πηγής δεδομένων εκτός από τα δεδομένα των κόμβων συνυπάρχουν και περιβαλλοντικά δεδομένα που προέρχονται από ανοικτές πηγές δεδομένων. Τα ανοιχτά δεδομένα μπορούν να χαρακτηριστούν ως δεδομένα που είναι διαθέσιμα στο σύνολό τους, προσβάσιμα στο διαδίκτυο και μπορούν να επαναχρησιμοποιηθούν και να αναδιανεμηθούν. Η χρήση τους προωθεί την οικονομική πρόοδο ενισχύοντας την επιχειρηματική δραστηριότητα για καινοτόμα προϊόντα και υπηρεσίες. Η πλειονότητα των πρωτοβουλιών ανοικτών δεδομένων είναι μέρος των προσπαθειών των κυβερνήσεων να ανοίξουν δεδομένα για να ενισχύσουν πλήρως τη διαφάνεια, να βελτιώσουν τη λογοδοσία, να δημιουργήσουν οικονομική ανάπτυξη, να προωθήσουν την καινοτομία, να ενδυναμώσουν τους πολίτες, να καταπολεμήσουν τη διαφθορά, να επιτύχουν περιβαλλοντικούς στόχους και παρέχουν καλύτερες δημόσιες υπηρεσίες.

Μια έξυπνη πόλη είναι ένα φυσικό περιβάλλον για την παραγωγή ανοικτών δεδομένων λόγω των εγγενών δυνατοτήτων της για μαζική παραγωγή δεδομένων. Η πληθώρα των δεδομένων που συλλέγονται μέσω των ενσωματωμένων αισθητήρων σε αστικές υποδομές, εγκαταστάσεις, περιβάλλοντα, σε συνδυασμό με τη χρήση αναλυτικών στοιχείων, βοηθούν στην πρόβλεψη και στην παροχή χρήσιμων γνώσεων για την ανάπτυξη της πόλης. Η αξιοποίηση του τεράστιου πλούτου δεδομένων για τη δημιουργία νέων προϊόντων, καινοτόμων υπηρεσιών και τη δημιουργία νέων έξυπνων οικοσυστημάτων πραγματοποιείται μέσω των διεπαφών προγραμματισμού εφαρμογών APIs. Τα APIs χρησιμοποιούνται ευρέως στη δημιουργία λύσεων IoT παρέχοντας απρόσκοπτη ροή

δεδομένων, μέσω του πρωτοκόλλου HTTP. Η χρήση των APIs επιτρέπει στους προγραμματιστές να σχεδιάζουν προηγμένες εφαρμογές που είναι εύκολο να ενσωματωθούν με άλλες υπηρεσίες Ιστού.

Κατά την υλοποίηση της εφαρμογής «Πράσινη Διαδρομή» αναζητήθηκαν πηγές ανοικτών περιβαλλοντικών δεδομένων με δυνατότητα διεπαφής προγραμματισμού εφαρμογών API. Οι περισσότερες πηγές που συναντήθηκαν στο διαδίκτυο παρείχαν υπηρεσίες Web API κυμαινόμενου κόστους, ανάλογα με τον αριθμό των αιτημάτων που μπορεί να υποβάλει ο χρήστης. Από την έρευνα προέκυψαν τέσσερις διαφορετικές πηγές με δωρεάν υπηρεσίες API [48]–[51], εκ των οποίων η μια θέτει ημερήσιο περιοριστικό όριο στα υποβληθέντα αιτήματα για την ανάκτηση πληροφορίας (Πίνακας 15).

Πίνακας 15: Πηγές ανοικτών περιβαλλοντικών δεδομένων.

A/A	Όνομα	Ιστότοπος	Δεδομένα	Ημερήσια Αιτήματα
1	iqair	https://www.iqair.com/	AQI, CO, NO ₂ , O ₃ , SO ₂ , PM2.5, PM10	Χωρίς περιορισμό
2	ninjas_airq	https://api-ninjas.com	AQI, CO, NO ₂ , O ₃ , SO ₂ , PM2.5, PM10	Χωρίς περιορισμό
3	open_weather	https://openweathermap.org	AQI, CO, NO, NO ₂ , O ₃ , SO ₂ , NH ₃ , PM2.5, PM10	Χωρίς περιορισμό
4	weatherbit	https://www.weatherbit.io	AQI, CO, NO ₂ , O ₃ , SO ₂ , PM2.5, PM10, επίπεδα γύρης	50

Οι ατμοσφαιρικές μετρήσεις οι οποίες αφορούν καιρικά δεδομένα και ατμοσφαιρικούς ρύπους παρέχονται από τα Web API των ιστότοπων του παραπάνω πίνακα. Οι ιστότοποι αυτοί υποστηρίζουν ατμοσφαιρικά δεδομένα για εκατοντάδες πόλεις ανά τον κόσμο, τα οποία ανανεώνονται σε τακτά χρονικά διαστήματα και μπορούν να αντληθούν τοπικά για οποιαδήποτε εφαρμογή. Κατά την εγγραφή στις παραπάνω υπηρεσίες, παρέχεται ένα μοναδικό κλειδί (API key) στο χρήστη με το οποίο μπορούν να πραγματοποιηθούν τα απομακρυσμένα αιτήματα κλήσεων στις υπηρεσίες αυτές. Οι μορφές των κλήσεων για κάθε υπηρεσία και τα αποτελέσματα των αιτημάτων παρουσιάζονται στα ακόλουθα σχήματα.

https://air-quality.p.rapidapi.com/current/airquality?lat=36.960876&lon=26.971832&hours=1

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	X-RapidAPI-Host	air-quality.p.rapidapi.com			
<input checked="" type="checkbox"/>	X-RapidAPI-Key	f2f1f98281msh84645eabe6802...			
Key	Value	Description			

Σχήμα 56: Αίτημα κλήσης στο Weatherbit.

```
1 {
2   "city_name": "Kálymnos",
3   "country_code": "GR",
4   "data": [
5     {
6       "aqi": 37,
7       "co": 323.77243,
8       "mold_level": 1,
9       "no2": 11.94,
10      "o3": 77.607,
11      "pm10": 40.54,
12      "pm25": 4.374072,
13      "pollen_level_grass": 1,
14      "pollen_level_tree": 1,
15      "pollen_level_weed": 1,
16      "predominant_pollen_type": "Molds",
17      "so2": 3.54
18    }
19  ],
20  "lat": 36.9609,
21  "lon": 26.9718,
22  "state_code": "ESYE42",
23  "timezone": "Europe/Athens"
24 }
```

Σχήμα 57: Αποτέλεσμα αιτήματος κλήσης στο Weatherbit.

http://api.openweathermap.org/data/2.5/air_pollution?lat=36.960876&lon=26.971832&appid=740af68f7f2cda5d66

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> lat	36.960876			
<input checked="" type="checkbox"/> lon	26.971832			
<input checked="" type="checkbox"/> appid	740af68f7f2cda5d66			
Key	Value	Description		

Σχήμα 58: Αίτημα κλήσης στο open_weather.

```
1  {
2      "coord": {
3          "lon": 26.9718,
4          "lat": 36.9609
5      },
6      "list": [
7          {
8              "main": {
9                  "aqi": 3
10             },
11             "components": {
12                 "co": 196.93,
13                 "no": 0,
14                 "no2": 2.38,
15                 "o3": 77.25,
16                 "so2": 3.34,
17                 "pm2_5": 20.98,
18                 "pm10": 40.61,
19                 "nh3": 0
20             },
21             "dt": 1667520769
22         }
23     ]
24 }
```

Σχήμα 59: Αποτέλεσμα αιτήματος κλήσης στο open_weather.

https://api.api-ninjas.com/v1/airquality?lat=36.960876&lon=26.971832&X-Api-Key=SVVP4kvv853P

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	lat	36.960876		
<input checked="" type="checkbox"/>	lon	26.971832		
<input checked="" type="checkbox"/>	X-Api-Key	SVVP4kvv853P		

Σχήμα 60: Αίτημα κλήσης στο ninjas_airq.

```
1 {
2   "CO": {
3     "concentration": 196.93,
4     "aqi": 2
5   },
6   "NO2": {
7     "concentration": 2.38,
8     "aqi": 2
9   },
10  "O3": {
11    "concentration": 77.25,
12    "aqi": 105
13  },
14  "SO2": {
15    "concentration": 3.34,
16    "aqi": 4
17  },
18  "PM2.5": {
19    "concentration": 20.98,
20    "aqi": 61
21  },
22  "PM10": {
23    "concentration": 40.61,
24    "aqi": 37
25  },
26  "overall_aqi": 105
27 }
```

Σχήμα 61: Αποτέλεσμα αιτήματος κλήσης στο ninjas_airq.

http://api.airvisual.com/v2/nearest_city?lat=36.960876&lon=26.971832&key=f55e45be-30ef-

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	lat	36.960876		
<input checked="" type="checkbox"/>	lon	26.971832		
<input checked="" type="checkbox"/>	key	f55e45be-30ef-		

Σχήμα 62: Αίτημα κλήσης στο iqair.

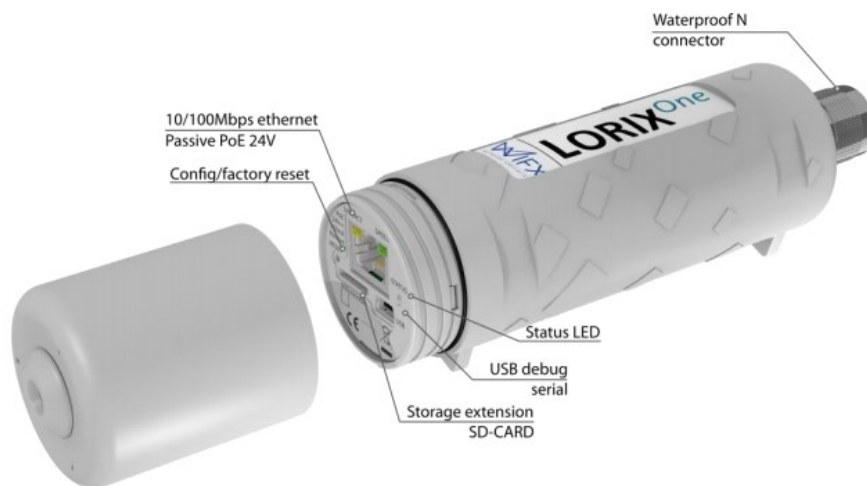
```
2   "status": "success",
3   "data": {
4     "city": "Kalymnos",
5     "state": "South Aegean",
6     "country": "Greece",
7     "location": {
8       "type": "Point",
9       "coordinates": [
10      26.9769,
11      36.95447
12    ]
13   },
14   "current": {
15     "pollution": {
16       "ts": "2022-11-04T00:00:00.000Z",
17       "aqius": 57,
18       "mainus": "p2",
19       "aqicn": 22,
20       "maincn": "p2"
21     },
22     "weather": {
23       "ts": "2022-11-04T00:00:00.000Z",
24       "tp": 20,
25       "pr": 1016,
26       "hu": 66,
27       "ws": 1.6,
28       "wd": 75,
29       "ic": "01n"
```

Σχήμα 63: Αποτέλεσμα αιτήματος κλήσης στο iqair.

9.3 Επίπεδο δικτύου

9.3.1 Πύλη δικτύωσης Lorix One WiFX

Για τη λήψη των δεδομένων από τις συσκευές που διαθέτουν τεχνολογία επικοινωνίας LoRa, απαιτείται η ύπαρξη μιας συσκευής που έχει το ρόλο της δικτυακής πύλης και προωθεί τα δεδομένα που λαμβάνει ασύρματα μέσω του πρωτοκόλλου LoRaWAN, στο διαδίκτυο. Για τις ανάγκες του έργου επιλέχθηκε το gateway Lorix One WiFX (Σχήμα 64), το οποίο είναι κατάλληλο για λειτουργία σε εξωτερικό περιβάλλον [52]. Η συσκευή διαθέτει μια θύρα PoE 24V για την τροφοδοσία και τη σύνδεσή της στο διαδίκτυο (Σχήμα 65). Επίσης, διατίθεται υποδοχή για microSD κάρτα μνήμης είτε για να εκκινήσει η συσκευή από αυτή, είτε για την επέκταση του αποθηκευτικού της χώρου. Η κατανάλωση της συσκευής κυμαίνεται μεταξύ 1W και 2.8W, καθιστώντας την ικανή να τροφοδοτηθεί από ηλεκτρικές διατάξεις παραγωγής ηλεκτρικής ενέργειας με φωτοβολταϊκά πάνελ.



Σχήμα 64: Lorix One Gateway.

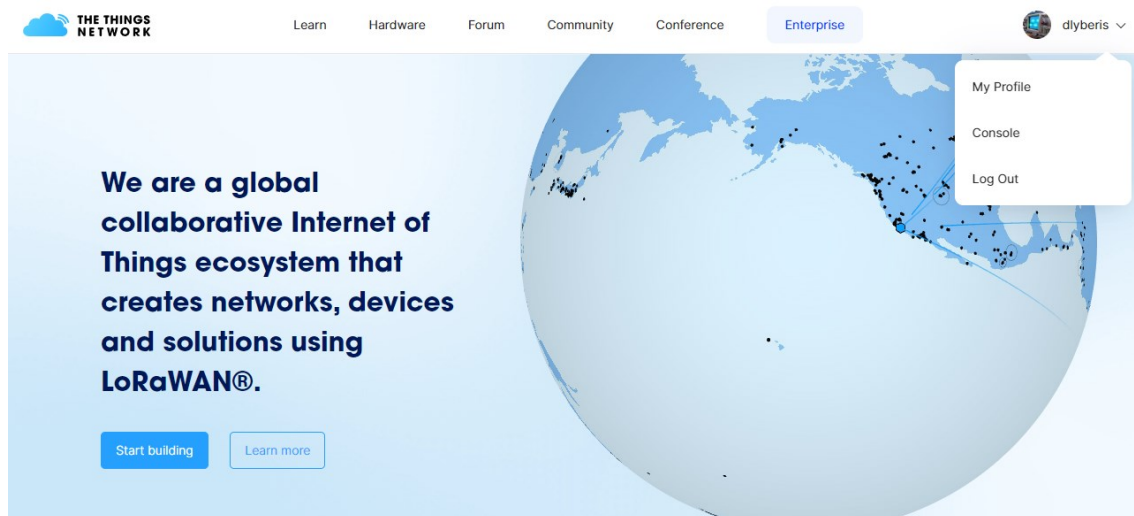
Σε περίπτωση που η συσκευή εγκατασταθεί σε δυσπρόσιτη περιοχή όπου η ενσύρματη σύνδεση δεν είναι εφικτή, μπορεί να συνδεθεί μέσω δρομολογητή κινητής τηλεφωνίας 3G/4G LTE και να αποκτήσει πρόσβαση στο διαδίκτυο. Τα πακέτα πληροφορίας που αποστέλλονται από τις απομακρυσμένες συσκευές, λαμβάνονται από το Lorix One και προωθούνται στο LoRaWAN Server. Ο τελευταίος, αποκωδικοποιεί τα πακέτα και αναλαμβάνει την προώθησή τους στον αντίστοιχο αποδέκτη ανάλογα με το είδος της εφαρμογής.



Σχήμα 65: Lorix One PoE.

9.3.2 Καταχώρηση Lorix One WiFX στο TTS

Η πλατφόρμα που χρησιμοποιήθηκε για τη χρήση του διακομιστή δικτύου LoRaWAN είναι το The Things Stack. Η αρχική επιλογή ήταν η πλατφόρμα The Things Network, η οποία δεν υποστηρίζεται πια από τα τέλη του 2021, για όλες τις εφαρμογές. Το The Things Stack αποτελεί τη μετεγκατάσταση του The Things Network σε μια νεότερη υποδομή υλικού και πλατφόρμα λογισμικού με σκοπό να συμβαδίσει με τις τρέχουσες τεχνολογίες. Για τη χρήση της πλατφόρμας απαιτείται η δημιουργία λογαριασμού με email. Η εγγραφή είναι δωρεάν, καθώς υιοθετείται πλήρως η φιλοσοφία της κοινότητας, να παρέχει ένα εντελώς δωρεάν και ανοιχτό δίκτυο (Σχήμα 66).

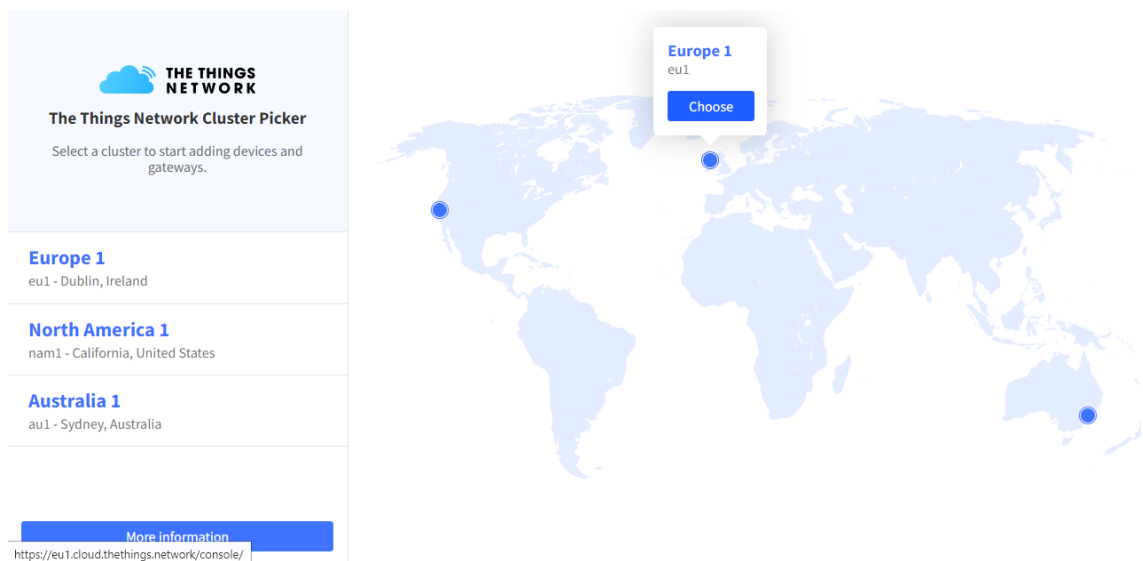


Σχήμα 66: TTN - Περιβάλλον πλατφόρμας.

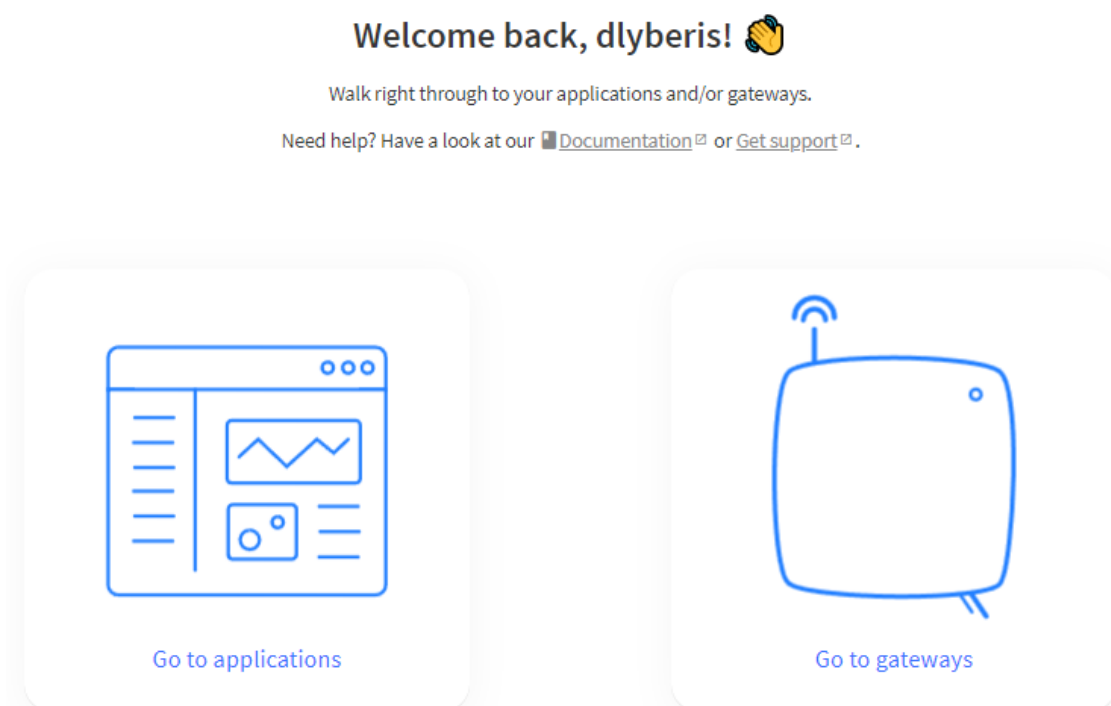
Με την επιλογή *Console* ο χρήστης μπορεί να επιλέξει το cluster στο οποίο θα μπορεί να και να προσθέσει συσκευές, δικτυακές πύλες και να δημιουργήσει εφαρμογές. Οι διαθέσιμες επιλογές είναι τα cluster Europe 1, North America 1, Australia 1 και υποδηλώνουν την τοποθεσία του κάθε cluster (Σχήμα 67). Η επιλογή του καταλληλότερου cluster γίνεται σύμφωνα με τη γεωγραφική θέση των δικτυακών πυλών που προστίθενται στο δίκτυο. Η πλησιέστερη γεωγραφική θέση του cluster οδηγεί σε ελαχιστοποίηση του χρόνου καθυστέρησης του δικτύου. Στα πλαίσια του έργου επιλέχθηκε το cluster Europe 1.

Μετά την επιλογή cluster, ο χρήστης μπορεί να προσθέσει, να αφαιρέσει και να διαχειριστεί εφαρμογές και πύλες δικτύωσης του LoRaWAN (Σχήμα 68). Με την επιλογή *Go to gateways* και στη συνέχεια *Register gateway* μπορεί να καταχωρηθεί και να ενεργοποιηθεί μια νέα πύλη δικτύωσης.

Το πρώτο βήμα της διαδικασίας καταχώρησης μιας δικτυακής πύλης είναι η εισαγωγή του EUI (Σχήμα 69). Το EUI είναι ένα μοναδικό αναγνωριστικό μεγέθους 64bit, που ενσωματώνεται σε κάθε συσκευή από τον εκάστοτε κατασκευαστή και μπορεί να χρησιμοποιηθεί για την καταχώριση της πύλης στο The Things Network. Σε περιπτώσεις που η συσκευή δεν διαθέτει ενσωματωμένο EUI, το βήμα αυτό μπορεί να παραληφθεί. Για τη δικτυακή πύλη Lorix One, το αναγνωριστικό EUI είναι η διεύθυνση MAC που αναγράφεται στη πίσω όψη του πλαστικού περιβλήματός της.



Σχήμα 67: TTN – Επιλογή Cluster.



Σχήμα 68: Περιβάλλον πλατφόρμας The Thing Stack.

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
Learn more in our [Gateway Guide](#).

Gateway EUI

FC C2 3D FF FE Confirm

To continue, please confirm the Gateway EUI so we can determine onboarding options

Σχήμα 69: Καταχώρηση EUI δικτυακής πύλης.

Τα επόμενα βήματα αφορούν την καταχώρηση ενός μοναδικού ονόματος της συσκευής και η επιλογή του πλάνου συχνότητας, σύμφωνα με τη νομοθεσία στην οποία υπάγεται η γεωγραφική περιοχή στην οποία θα εγκατασταθεί η συσκευή (Σχήμα 70).

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.
Learn more in our [Gateway Guide](#).

Gateway EUI

FC C2 3D FF FE Reset

Gateway ID *

aegean-uni-gtw-lrv

Gateway name

My new gateway

Frequency plan *

Europe 863-870 MHz (SF9 for RX2 - recommended) | v

Require authenticated connection

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

Select which information can be seen by other network participants, including [Packet Broker](#)

Share status within network

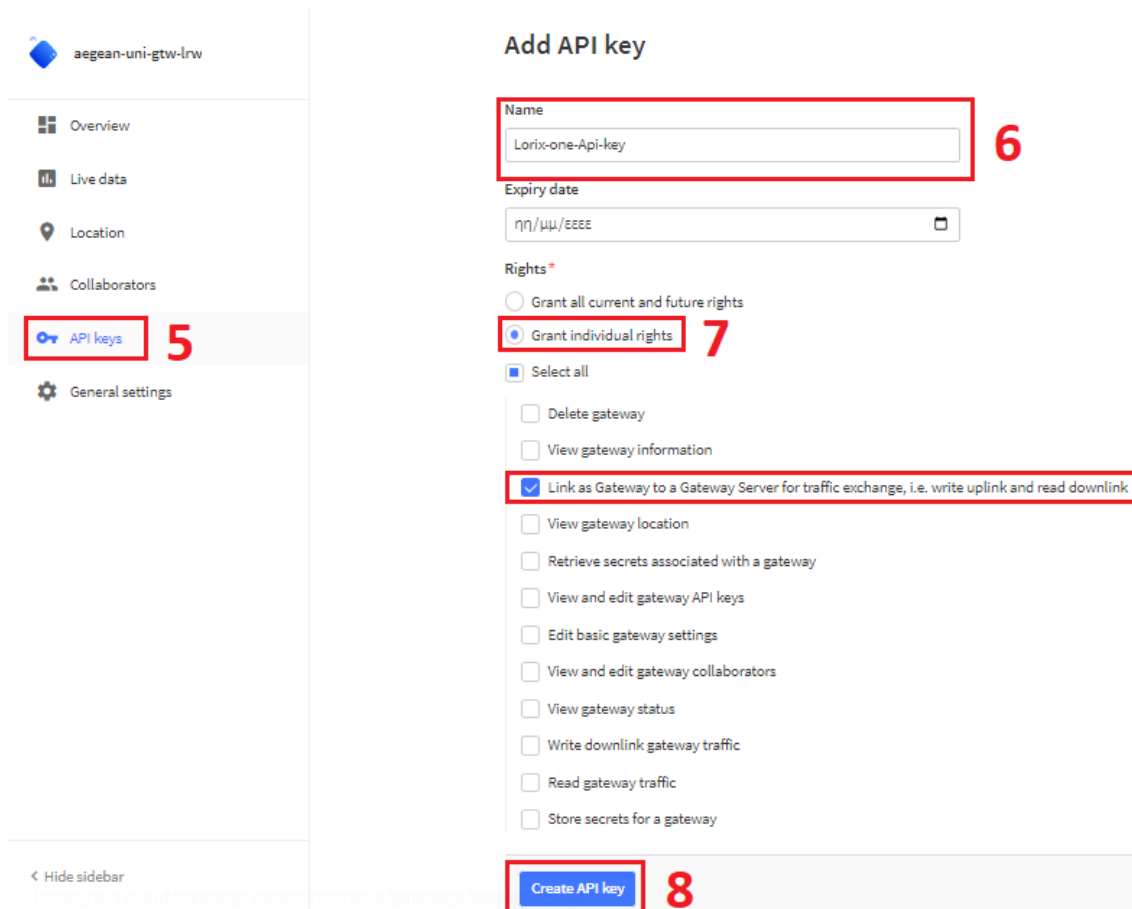
Share location within network

Register gateway

Σχήμα 70: Καταχώρηση ονόματος συσκευής και πλάνου συχνότητας.

Ορισμένες δικτυακές πύλες απαιτούν ένα κλειδί API για να τους εκχωρηθούν κατάλληλα δικαιωμάτων και να τους επιτραπεί η σύνδεση στο The Things Stack. Το Lorix One χρειάζεται ένα τέτοιο κλειδί για την αυθεντικοποίηση του στο δίκτυο. Για την έκδοση API κλειδιού, επιλέγεται στο μενού *API keys* της δικτυακής πύλης που καταχωρείται και στη συνέχεια *Προσθήκη κλειδιού*

API. Στο νέο παράθυρο εισάγεται ένα όνομα για το κλειδί και επιλέγεται η σύνδεση ως πύλη σε έναν διακομιστή πύλης για την ανταλλαγή κίνησης πακέτων. Το κλειδί που δημιουργήθηκε θα χρησιμοποιηθεί στη διαδικασία ρύθμισης της συσκευής όπως αναφέρεται στην επόμενη ενότητα.



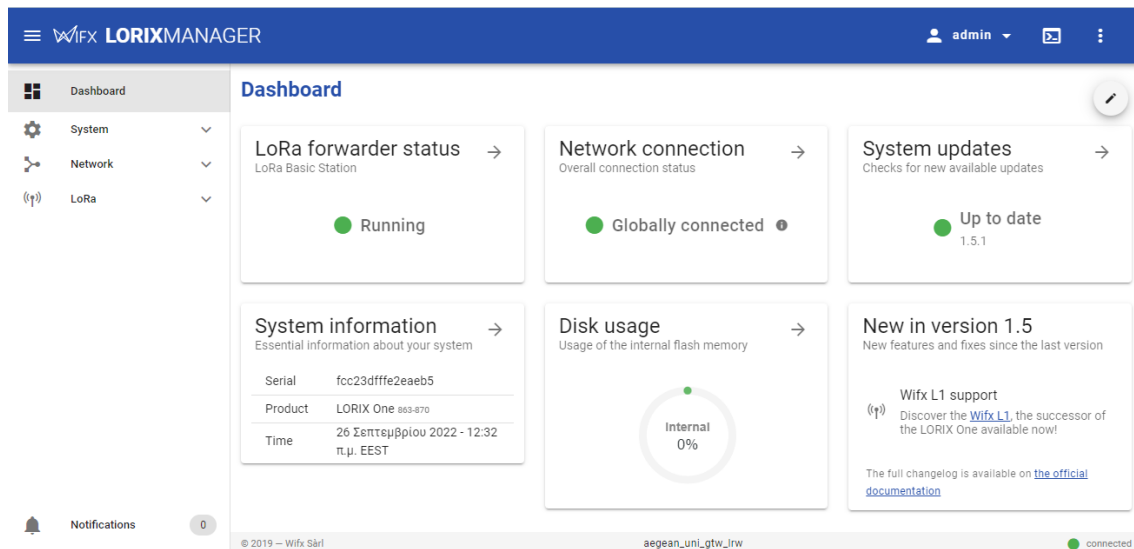
Σχήμα 71: Δημιουργία κλειδιού API για την αυθεντικοποίηση της δικτυακής πύλης.

9.3.3 Ρύθμιση Lorix One WiFX

Η συσκευή Lorix One προσφέρει δυο διαφορετικούς τρόπους προσβασιμότητας για το χρήστη, απομακρυσμένη πρόσβαση μέσω του δικτύου ή τοπική μέσω ενός καλωδίου mini USB. Στα πλαίσια του έργου επιλέχθηκε ο πρώτος τρόπος για την ρύθμισή της.

Με τη σύνδεση της συσκευής στο τοπικό δίκτυο, θα της αποδοθεί αυτόματα μια διεύθυνση IP από τον DHCP Server του δικτύου. Εάν δεν είναι δυνατή η πρόσβαση σε διακομιστή DHCP, η δικτυακή πύλη δεν θα λάβει καμία διεύθυνση και θα πρέπει να ρυθμιστεί μέσω του USB καλωδίου. Η συσκευή χρησιμοποιεί το πρωτόκολλο mDNS (multicast DNS), ώστε να είναι προσβάσιμη από το τοπικό δίκτυο με το ακόλουθο όνομα DNS: <https://lorix-one-xxxxxx.local>. Όπου xxxxxx είναι τα τελευταία 6 ψηφία της διεύθυνσης MAC, που αναγράφεται στην πίσω όψη του πλαστικού περιβλήματος. Η πρόσβαση στον πίνακα ελέγχου της συσκευής πραγματοποιείται από οποιοδήποτε ενημερωμένο πρόγραμμα περιήγησης (Edge, Safari, Firefox, Chrome, Opera), πληκτρολογώντας τη διεύθυνση IP ή το όνομα DNS (Σχήμα 72).

Το μόνο που απαιτείται για τη σύνδεση είναι το όνομα χρήστη και ο κωδικός πρόσβασης. Τα προκαθορισμένα στοιχεία πρόσβασης είναι admin/lorix4u, σύμφωνα με τις ρυθμίσεις του κατασκευαστή. Συνιστάται η αλλαγή του κωδικού πρόσβασης, για μεγαλύτερη ασφάλεια. Εφόσον έχει γίνει πλέον η σύνδεση στις ρυθμίσεις της συσκευής, είναι εφικτός ο προγραμματισμός της για τις ανάγκες του δικτύου.



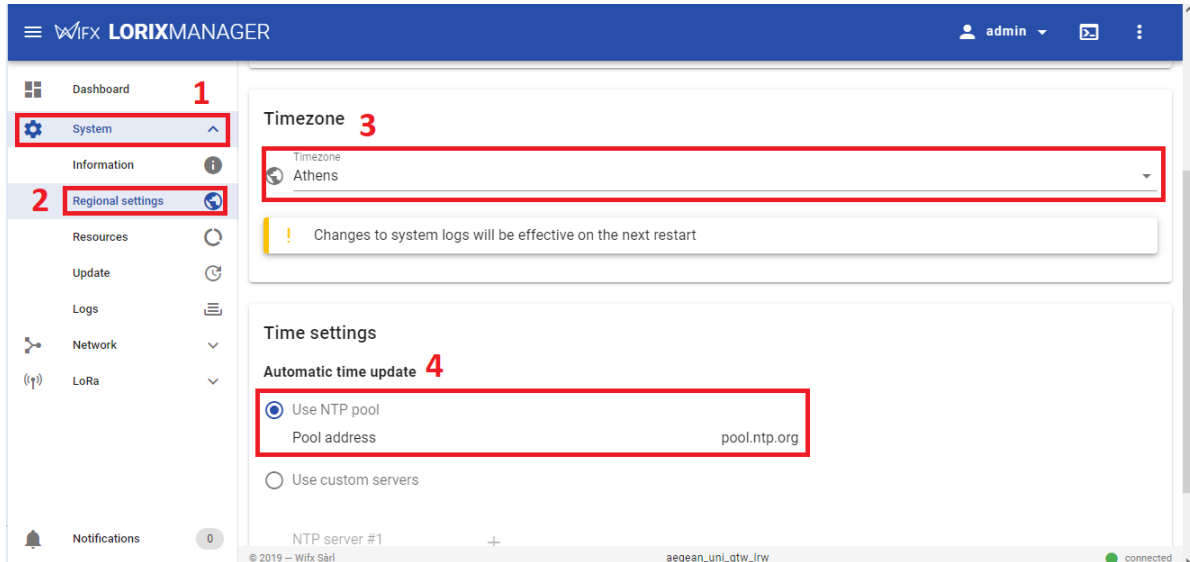
Σχήμα 72: Πίνακας ελέγχου Lorix One.

Αρχικά, απαιτείται η μετάβαση στις ρυθμίσεις της συσκευής για τον ορισμό της ζώνης ώρας και του εξυπηρετητή του πρωτοκόλλου δικτυακού χρόνου για το συγχρονισμό της ώρας της συσκευής. Η ρύθμιση αυτή είναι απαραίτητη για την καταγραφή των συμβάντων της συσκευής με ορθή χρονοσήμανση (Σχήμα 73). Στη συνέχεια, ορίζεται η συχνότητα λειτουργίας LoRa σύμφωνα με την γεωγραφική περιοχή χρήσης της συσκευής καθώς και το κέρδους της κεραίας που χρησιμοποιείται (Σχήμα 74). Το βήμα αυτό πρέπει υποχρεωτικά να προηγηθεί πριν ξεκινήσει η διαδικασία προώθησης πακέτων.

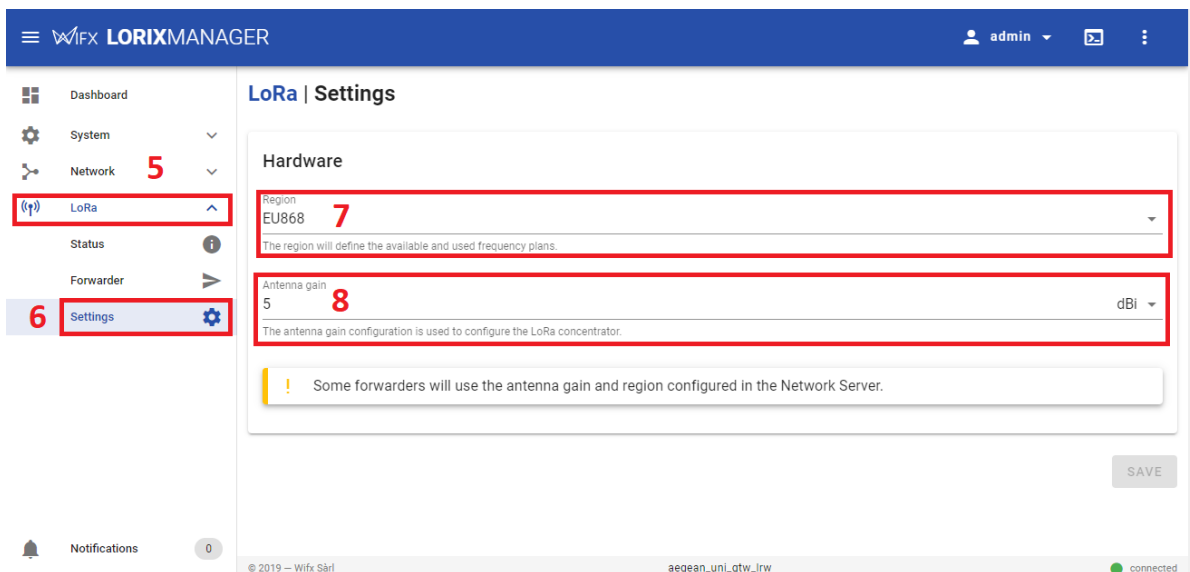
Η πύλη λαμβάνει πακέτα LoRaWAN από το περιβάλλον. Στη γενική αρχιτεκτονική LoRaWAN, αυτά τα πακέτα προωθούνται σε έναν διακομιστή δικτύου ή μια υπηρεσία νέφους, τα οποία θα δρομολογήσουν αυτά τα πακέτα περαιτέρω στον διακομιστή εφαρμογών. Η πιο γνωστή υπηρεσία είναι το είναι το The Things Network. Το επόμενο βήμα αφορά τη ρύθμιση του Packet Forwarder για την επιλογή του σωστού προγράμματος προώθησης πακέτων. Το The Things Network χρησιμοποιείται το LoRa Basic Station (Σχήμα 75). Η διαμόρφωση του Basic Station ορίζεται αυτόματα σχετικά με την περιοχή και τον τύπο της κεραίας όπως πραγματοποιήθηκε στο προηγούμενο βήμα.

Το τελευταίο μέρος της ρύθμισης του Lorix One αποτελεί η ενεργοποίηση του διακομιστή δικτύου LoRaWAN (Σχήμα 76). Εισάγεται η διεύθυνση και η θύρα του διακομιστή που χρησιμοποιήθηκε κατά την εγγραφή της δικτυακής πύλης στο TTN. Η τυπική μορφή της διεύθυνσης του διακομιστή είναι region-server.cloud.thethings.network ενώ η θύρα είναι η 8887. Ενεργοποιείται η ασφαλής διασύνδεση με τον διακομιστή μέσω του πρωτοκόλλου TLS και επιλέγεται η μεταφόρτωση πιστοποιητικού αυθεντικοποίησης. Το πιστοποιητικό που επιλέχθηκε

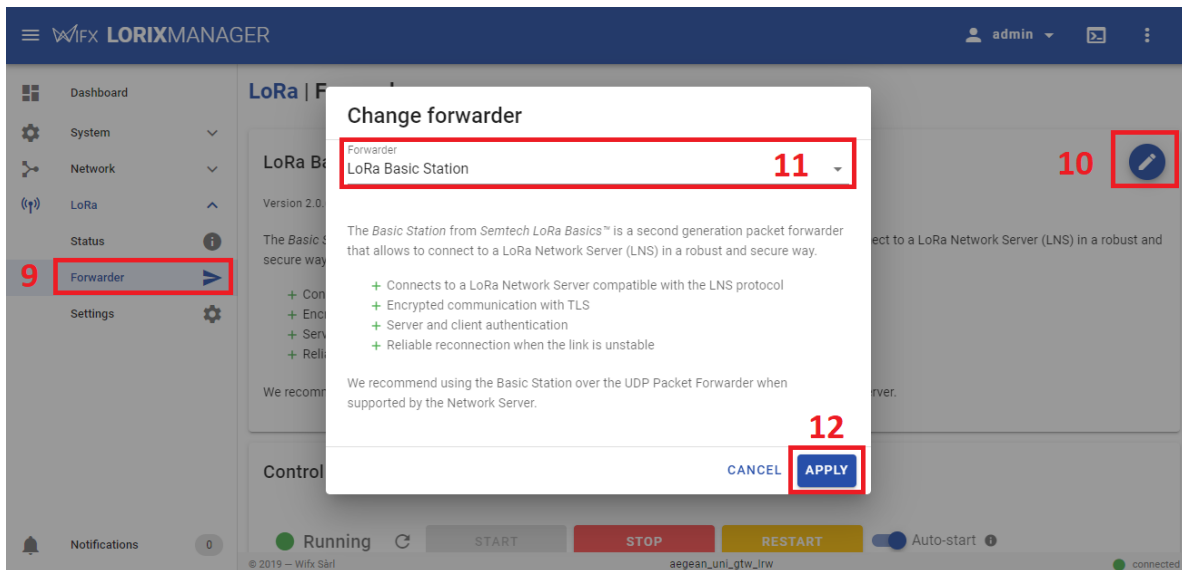
είναι το ISRG Root X1 [53] Ενεργοποιείται ο έλεγχος ταυτότητας πελάτη και επιλέγεται η χρήση διακριτικού. Στο πεδίο αυτό εισάγεται το κλειδί API που δημιουργήθηκε κατά την εγγραφή της συσκευής στο The Things Network. Σε περίπτωση που δεν συγκρατήθηκε το κλειδί στην φάση της εγγραφής, απαιτείται η διαγραφή του από το TTN και η έκδοση νέου.



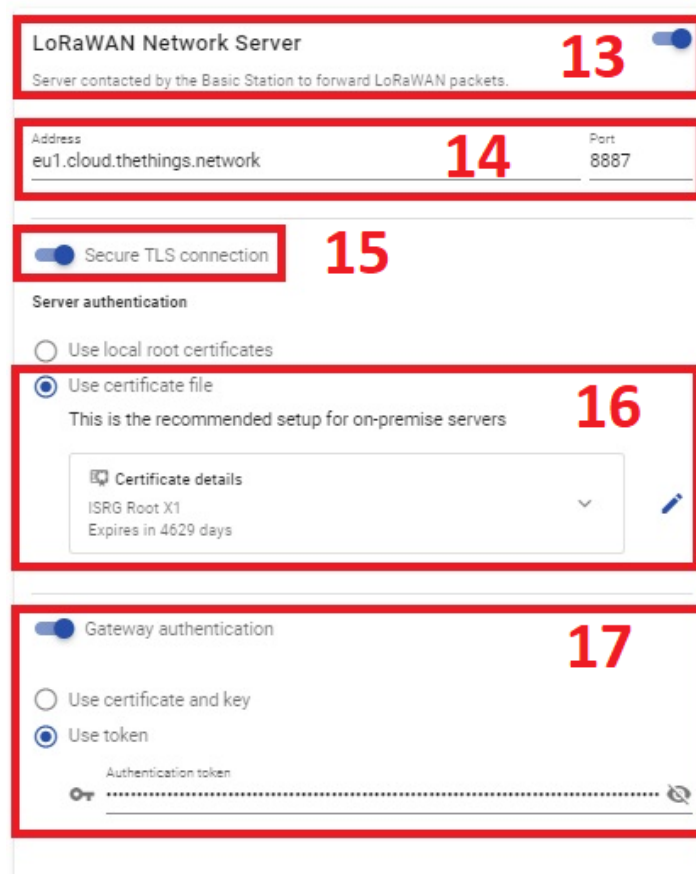
Σχήμα 73: Lorix One - Ρύθμιση ζώνης ώρας και NTP.



Σχήμα 74: Lorix One - Ρύθμιση συχνότητας LoRa και κέρδους κεραίας.



Σχήμα 75: Lorix One - Ρύθμιση προγράμματος προώθησης πακέτων.

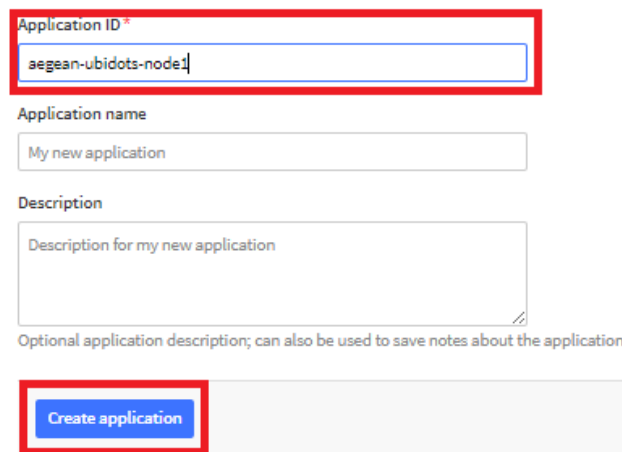


Σχήμα 76: Lorix One - Ρυθμίσεις διακομιστή LoRaWAN.

9.3.4 Σύνδεση IoT κόμβων στο The Things Stack

Για την αναγνώριση των LoRaWAN κόμβων από τον διακομιστή δικτύου, απαιτείται η προσθήκη των συσκευών σε αυτόν και η ρύθμισή τους με τα αντίστοιχα κλειδιά τα οποία χρησιμοποιούνται για την ταυτοποίησή τους. Τα πακέτα που λαμβάνονται από τον διακομιστή και δεν συνοδεύονται από έγκυρα κλειδιά απορρίπτονται. Οι συσκευές ομαδοποιούνται στο διακομιστή σε μια οντότητα που ονομάζεται «application», η οποία περιλαμβάνει τις ρυθμίσεις που αφορούν το σύνολο των συσκευών που περιλαμβάνονται σε αυτή. Αρχικά, ο χρήστης μεταβαίνει στο μενού *Add application* για να δημιουργήσει την εφαρμογή, συμπληρώνει το αναγνωριστικό της αίτησης και επιλέγει το *Create Application* για να δημιουργήσει την εφαρμογή (Σχήμα 77). Η εφαρμογή δημιουργείται και ο χρήστης ανακατευθύνεται στη σελίδα επισκόπησης εφαρμογής (Σχήμα 78).

Add application



Application ID*

aegean-ubidots-node

Application name

My new application

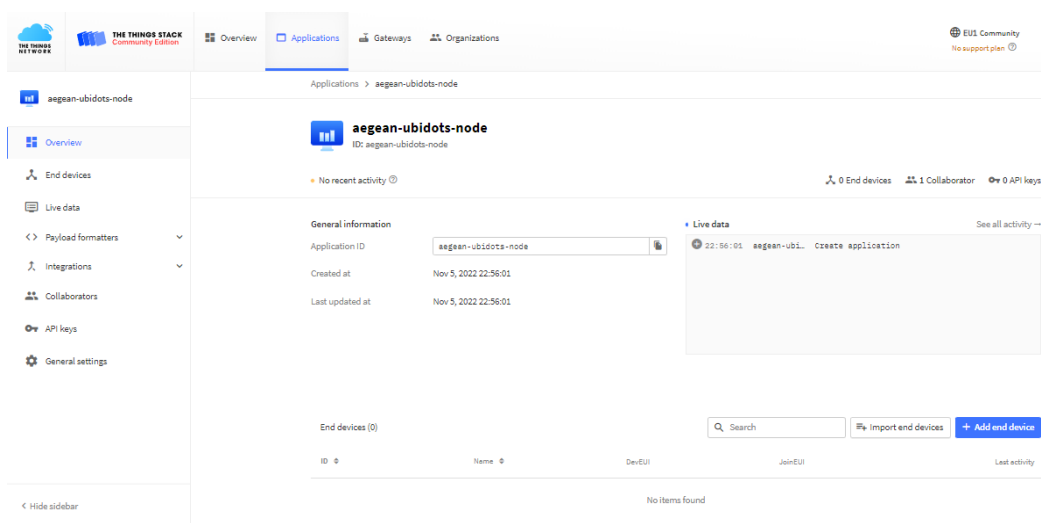
Description

Description for my new application

Optional application description; can also be used to save notes about the application

Create application

Σχήμα 77: Δημιουργία application στο The Things Stack.



aegean-ubidots-node

Applications > aegean-ubidots-node

aegean-ubidots-node
ID: aegean-ubidots-node

No recent activity

0 End devices 1 Collaborator 0 API keys

General information

Application ID aegean-ubidots-node

Created at Nov 5, 2022 22:58:01

Last updated at Nov 5, 2022 22:58:01

Live data

22:56:01 aegean-ubi... Create application

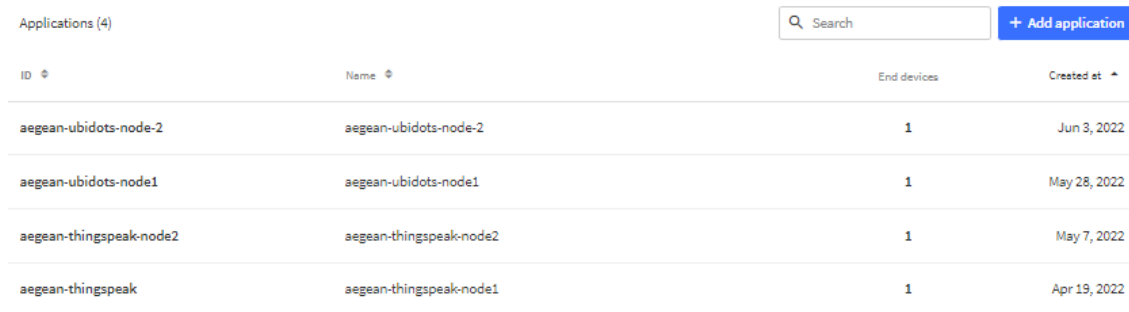
End devices (0)

Search Import end devices Add end device

ID	Name	DevEUI	JoinEUI	Last activity
No items found				

Σχήμα 78: Σελίδα επισκόπησης εφαρμογής.

Μια εφαρμογή μπορεί να περιέχει απεριόριστο αριθμό συσκευών, αλλά είναι προτιμότερο να ταξινομούνται σε εφαρμογές ανά λειτουργία ή γεωγραφική περιοχή. Στην περίπτωση της εφαρμογής «Πράσινη Διαδρομή» κάτι τέτοιο δεν ήταν εφικτό, εξαιτίας της ιδιαιτερότητας των IoT πλατφορμών να αντιστοιχούν ένα κανάλι συλλογής δεδομένων με μια μόνο συσκευή. Για το λόγο αυτό δημιουργήθηκαν τόσες εφαρμογές όσες είναι και οι LoRaWAN συσκευές (Σχήμα 79).



ID	Name	End devices	Created at
aegean-ubidots-node-2	aegean-ubidots-node-2	1	Jun 3, 2022
aegean-ubidots-node1	aegean-ubidots-node1	1	May 28, 2022
aegean-thingspeak-node2	aegean-thingspeak-node2	1	May 7, 2022
aegean-thingspeak	aegean-thingspeak-node1	1	Apr 19, 2022

Σχήμα 79: Σύνολο υλοποιημένων εφαρμογών στο TTS.

Για την προσθήκη μιας συσκευής, αρχικά ο χρήστης ανοίγει την εφαρμογή στην οποία θέλει να προσθέσει τη συσκευή. Μεταβαίνει στην επιλογή *End devices* που βρίσκεται στο αριστερό μενού και επιλέγει *Add end device* για να ανακατευθυνθεί στη σελίδα εγγραφής της τελικής συσκευής. Επιλέγεται η μη αυτόματη καταχώρηση συσκευής και υποβάλλονται οι κατάλληλες πληροφορίες σύμφωνα με το εγχειρίδιο του κατασκευαστή (Σχήμα 80). Επιλέγεται το πλάνο συχνότητας, σύμφωνα με τη νομοθεσία στην οποία υπάγεται η γεωγραφική περιοχή στην οποία θα εγκατασταθεί η συσκευή και εισάγονται η έκδοση LoRaWAN και η έκδοση περιφερειακών παραμέτρων. Η δικτυακή πύλη και η συσκευή πρέπει να χρησιμοποιούν το ίδιο πλάνο συχνότητας για την εγκαθίδρυση επικοινωνίας μεταξύ τους.



End device type

Input Method

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan *

Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version *

LoRaWAN Specification 1.0.2

Regional Parameters version *

RP001 Regional Parameters 1.0.2 revision B

Σχήμα 80: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 1).

Στη συνέχεια επιλέγεται η τροποποίηση των σύνθετων ρυθμίσεων της συσκευής, για τον ορισμό της μεθόδου ενεργοποίησης και της κλάσης της (Σχήμα 81). Για τις συσκευές που χρησιμοποιήθηκαν στην εφαρμογή, ορίστηκε ως μέθοδος ενεργοποίησης OTAA και η κλάση A. Η μέθοδος OTAA αποτελεί την πιο ασφαλή και επεκτάσιμη μέθοδο ενεργοποίησης των συσκευών LoRaWAN και έχει καθιερωθεί ως προεπιλογή των συσκευών που διατίθενται στην αγορά.

Show advanced activation, LoRaWAN class and cluster settings ^

Activation mode ⓘ *

Over the air activation (OTAA)

Activation by personalization (ABP)

Define multicast group (ABP & Multicast)

Additional LoRaWAN class capabilities ⓘ

None (class A only) | v

Network defaults ⓘ

Use network's default MAC settings

Cluster settings ⓘ

Skip registration on Join Server

Σχήμα 81: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 2).

Στο τελευταίο στάδιο της ρύθμισης εισάγεται το μοναδικό αναγνωριστικό JoinEUI, εάν αυτό παρέχεται από τον κατασκευαστή της συσκευής. Σε αντίθετη περίπτωση το πεδίο αυτό μπορεί να συμπληρωθεί με μηδενικά με την προϋπόθεση να προγραμματιστεί η συσκευή με την ίδια τιμή. Το JoinEUI είναι ένα αναγνωριστικό μεγέθους 64bit που χρησιμοποιείται για την αναγνώριση του Join Server κατά την ενεργοποίηση. Ο ρόλος του Join Server είναι να αποθηκεύει τα Root Keys, να δημιουργεί κλειδιά περιόδου λειτουργίας και να τα στέλνει με ασφάλεια στον διακομιστή δικτύου και στον διακομιστή εφαρμογών. Ακολουθεί η εισαγωγή του DevEUI το οποίο έχει εκχωρηθεί από τον κατασκευαστή, για τις εμπορικές συσκευές. Εάν η συσκευή είναι προγραμματιζόμενη, μπορεί να δημιουργηθεί ένα τυχαίο αναγνωριστικό χρησιμοποιώντας το κουμπί *Generate* με την προϋπόθεση να προγραμματιστεί η συσκευή με την ίδια τιμή. Το DevEUI είναι ένα αναγνωριστικό μεγέθους 64bit που χρησιμοποιείται για την αναγνώριση της συσκευής. Επίσης, εισάγεται το κλειδί κρυπτογράφησης AppKEY που χρησιμοποιείται από τη μέθοδο ενεργοποίησης OTAA, εάν αυτό παρέχεται από τον κατασκευαστή. Διαφορετικά μπορεί να δημιουργηθεί ένα χρησιμοποιώντας το κουμπί *Generate* και να προγραμματιστεί στη συσκευή. Τέλος, εισάγεται το μοναδικό αναγνωριστικό της συσκευής και επιλέγεται το κουμπί *Register end device* (Σχήμα 82).

Provisioning information

JoinEUI

This end device can be registered on the network

DevEUI 7/50 used

AppKey

End device ID

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

Σχήμα 82: Ρυθμίσεις LoRaWAN συσκευής στο TTS (Μέρος 3).

9.3.5 Μορφοποίηση ωφέλιμης πληροφορίας στο TTS

Όπως αναφέρθηκε και στην ενότητα «Περιγραφή λειτουργίας IoT κόμβων», οι συσκευές IoT που χρησιμοποιούν το πρωτόκολλο LoRaWAN κωδικοποιούν τις τιμές των μετρούμενων μεγεθών (payload) σε bytes για την αποστολή των δεδομένων. Από την πλευρά της διαχείρισης του δικτύου, κάθε νέο μήνυμα που καταφτάνει στον εξυπηρετητή εφαρμογών, υπόκεινται στην αντίστροφη διαδικασία για την αποκωδικοποίηση των bytes και την εξαγωγή της ωφέλιμης πληροφορίας και την προώθησή της στο επόμενο επίπεδο της αρχιτεκτονικής.

Οι μορφοποιητές δεδομένων (Payload formatters) επιτρέπουν τη επεξεργασία δεδομένων από και προς τελικές συσκευές. Αυτό είναι χρήσιμο για τη μετατροπή των κωδικοποιημένων δεδομένων σε διακριτή πληροφορία που είναι αναγνώσιμη και διαχειρίσιμη. Στο The Things Stack ο χρήστης έχει τη δυνατότητα να δημιουργήσει εξατομικευμένους αποκωδικοποιητές δεδομένων για κάθε συσκευή ξεχωριστά ή να τον υλοποιήσει σε κάθε εφαρμογή. Με τη δεύτερη μέθοδο δίνεται η δυνατότητα της αποκωδικοποίησης των μηνυμάτων που προέρχονται από όλες τις συσκευές που ανήκουν σε μία εφαρμογή και χρησιμοποιούν την ίδια μορφή κωδικοποίησης των δεδομένων.

Για την προσθήκη της συνάρτησης αποκωδικοποίησης, ο χρήστης ανοίγει την εκάστοτε εφαρμογή, μεταβαίνει στην επιλογή *Payload formatters* που βρίσκεται στο αριστερό μενού και επιλέγει *Uplink* για να ανακατευθυνθεί στη σελίδα σύνταξης της συνάρτησης. Η συνάρτηση `JavaScript Decoder()` καλείται όταν λαμβάνεται ένα μήνυμα ανερχόμενης ζεύξης δεδομένων από μια συσκευή. Αυτή η συνάρτηση αποκωδικοποιεί το ωφέλιμο φορτίο που λαμβάνεται από την τελική συσκευή και την μετατρέπει σε ένα αναγνώσιμο αντικείμενο JSON που αποστέλλεται στις IoT πλατφόρμες. Στα πλαίσια υλοποίησης της «Πράσινης Διαδρομής» συντάχθηκαν δυο προσαρμοσμένοι μορφοποιητές payload με τη γλώσσα προγραμματισμού JavaScript και εφαρμόστηκαν στις αντίστοιχες εφαρμογές (Σχήμα 83, Σχήμα 84). Η κύρια διαφορά μεταξύ των

δύο συναρτήσεων είναι ο τρόπος αναπαράστασης της αποκωδικοποιημένης πληροφορίας και αυτό εξαρτάται από τη μορφή των δεδομένων που μπορούν δεχτούν οι IoT πλατφόρμες. Στο Σχήμα 85 απεικονίζεται ένα παράδειγμα αποκωδικοποίησης μηνύματος, κάνοντας χρήση της δοκιμαστική σελίδας ελέγχου που προσφέρεται στο μενού της καταχωρημένης συσκευής στο TTS.

Setup

Formatter type *

Custom Javascript formatter

Formatter code *

```
1 function Decoder(bytes, port) {
2   var PM2_5 = (bytes[3] << 24) | (bytes[2]<< 16) | (bytes[1]<< 8) | (bytes[0]);
3   var PM10 = (bytes[7] << 24) | (bytes[6]<< 16) | (bytes[5]<< 8) | (bytes[4]);
4   var S02 = (bytes[11] << 24) | (bytes[10]<< 16) | (bytes[9]<< 8) | (bytes[8]);
5   var N02 = (bytes[15] << 24) | (bytes[14]<< 16) | (bytes[13]<< 8) | (bytes[12]);
6   var O3 = (bytes[19] << 24) | (bytes[18]<< 16) | (bytes[17]<< 8) | (bytes[16]);
7   var lat = ((bytes[23] << 24) | (bytes[22]<< 16) | (bytes[21]<< 8) | (bytes[20]))/1000000;
8   var lng = ((bytes[27] << 24) | (bytes[26]<< 16) | (bytes[25]<< 8) | (bytes[24]))/1000000;
9   var position = {"value": 1, context:{"lat": lat, "lng": lng}};
10
11  return {
12    position:position,
13    O3:O3,
14    N02:N02,
15    S02:S02,
16    PM10:PM10,
17    PM2_5:PM2_5
18  };
19 }
```

Σχήμα 83: Payload formatter για την IoT πλατφόρμα Ubidots.

Setup

Formatter type *

Custom Javascript formatter

Formatter code *

```
1 function decodeUplink(input) {
2   var b=input.bytes;
3   var pm2_5 = (b[1]<< 8)|b[0];
4   var pm10 = (b[3]<< 8)|b[2];
5   var S02 = (b[5]<< 8)|b[4];
6   var N02 = (b[7]<< 8)|b[6];
7   var O3 = (b[9]<< 8)|b[8];
8   var lat = ((b[13]<< 24)|(b[12]<< 16)|(b[11]<< 8)|b[10])/1000000;
9   var long = ((b[17]<< 24)|(b[16]<< 16)|(b[15]<< 8)|b[14])/1000000;
10  return {
11    data: {
12      field1: pm2_5,
13      field2: pm10,
14      field3: S02,
15      field4: N02,
16      field5: O3,
17      field6: lat,
18      field7: long
19    },
20    warnings: [],
21    errors: []
22  };
23 }
```

Σχήμα 84: Payload formatter για την IoT πλατφόρμα Thingspeak.

Byte payload: 5E 00 00 00 C5 02 00 00 56 03 00 00 89 03 00 00 07 02

FPort: 1

Test decoder

Decoded test payload

```
{
  "NO2": 905,
  "O3": 519,
  "PM10": 709,
  "PM2_5": 94,
  "SO2": 854,
  "position": {
    "context": {
      "lat": 36.950396,
      "lng": 26.977652
    }
  },
  "value": 1
}
```

Complete uplink data

```
{
  "f_port": 1,
  "frm_payload": "XgAAAMUCAABWAwAAiQMAAAcCAAB80TMCdKkwbAQ==",
  "decoded_payload": {
    "NO2": 905,
    "O3": 519,
    "PM10": 709,
    "PM2_5": 94,
    "SO2": 854,
    "position": {
      "context": {
        "lat": 36.950396
      }
    }
  }
}
```

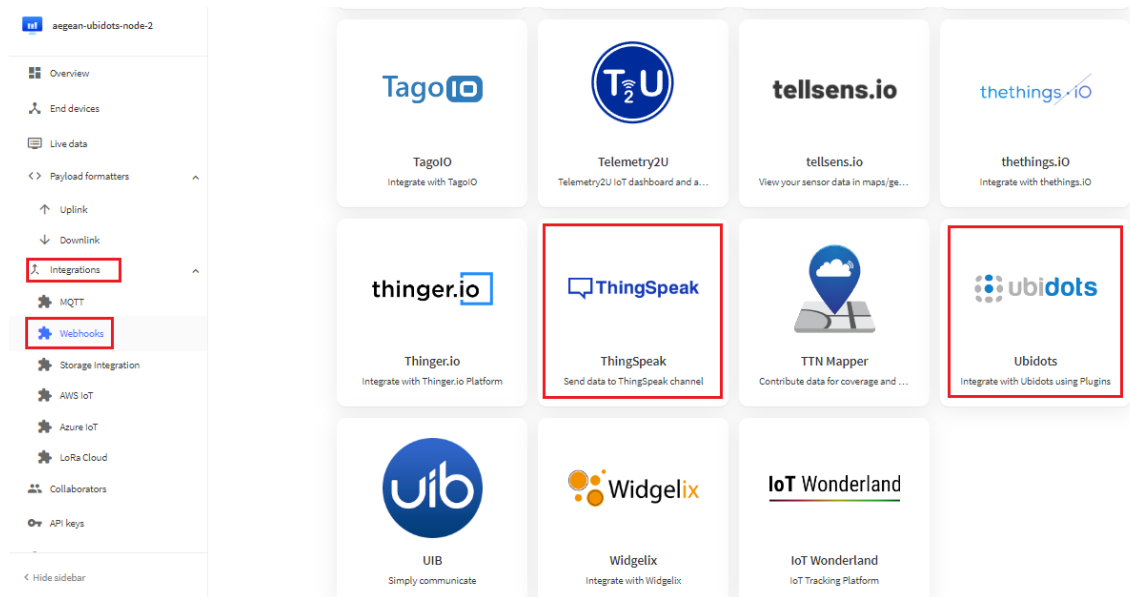
✔ Payload is valid

Σχήμα 85: Παράδειγμα αποκωδικοποίησης μηνύματος.

9.3.6 Διασύνδεση TTS με τις ΙοΤ πλατφόρμες

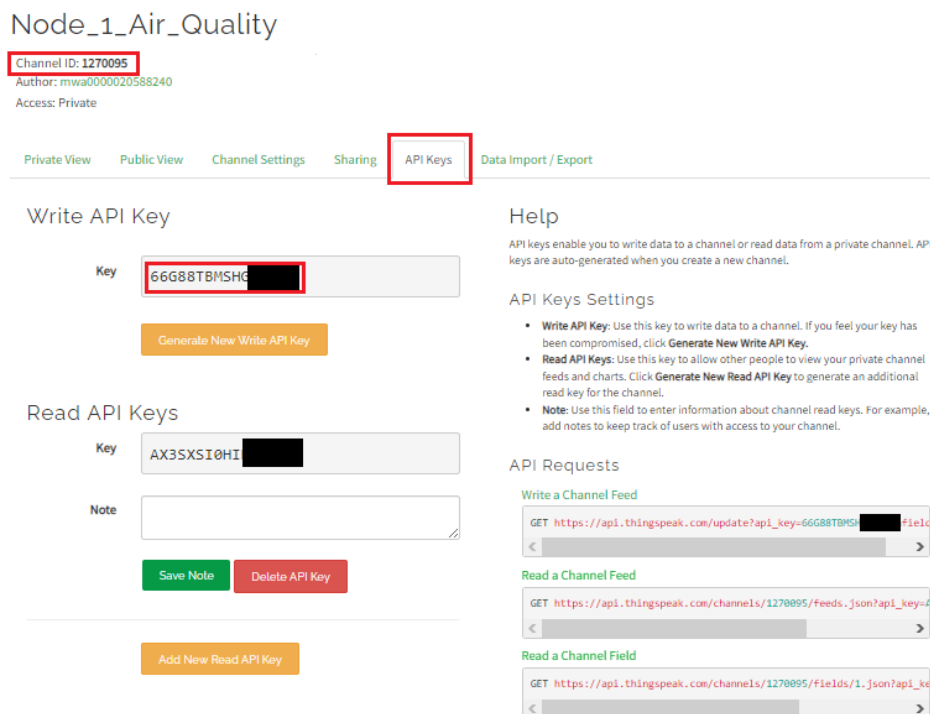
Το περιβάλλον του TTS προσφέρει τη δυνατότητα διασύνδεσης με εφαρμογές υψηλού επιπέδου, που ονομάζονται *integrations* και συντελούν στην αποθήκευση, οπτικοποίηση και ανάλυση των δεδομένων που αποστέλλονται από τις συσκευές, μέσω διεπαφών προγραμματισμού εφαρμογών APIs. Τέτοιες εφαρμογές είναι οι ΙοΤ πλατφόρμες Thingspeak και Ubidots, που χρησιμοποιούνται στην εφαρμογή «Πράσινη Διαδρομή». Η διαχείριση των *integrations* γίνεται εντός των εφαρμογών που υλοποιήθηκαν σε προηγούμενη ενότητα. Η διασύνδεση του The Things Stack με τις ΙοΤ πλατφόρμες πραγματοποιείται μέσω των Webhook τα οποία επιτρέπουν την αποστολή μηνυμάτων σε συγκεκριμένα τελικά σημεία HTTP(S), μεταβιβάζοντας τα δεδομένα ως ένα αντικείμενο JSON. Τα τελικά σημεία ορίζονται από τις ΙοΤ πλατφόρμες, για την παραλαβή των δεδομένων.

Για την προσθήκη ενός Webhook, ο χρήστης μεταβαίνει στην κονσόλα ανοίγει την εκάστοτε εφαρμογή και επιλέγει την καρτέλα *Webhooks* στο μενού *Integrations*. Πατώντας το κουμπί *Add webhook* πραγματοποιείται ανακατεύθυνση στη σελίδα επιλογής ή δημιουργίας νέου Webhook. Στα πλαίσια υλοποίησης της εφαρμογής επιλέχθηκαν τα Webhooks Thingspeak και Ubidots κατά αντιστοιχία με τις ρυθμισμένες εφαρμογές (Σχήμα 79).



Σχήμα 86: Σελίδα επιλογής Webhook.

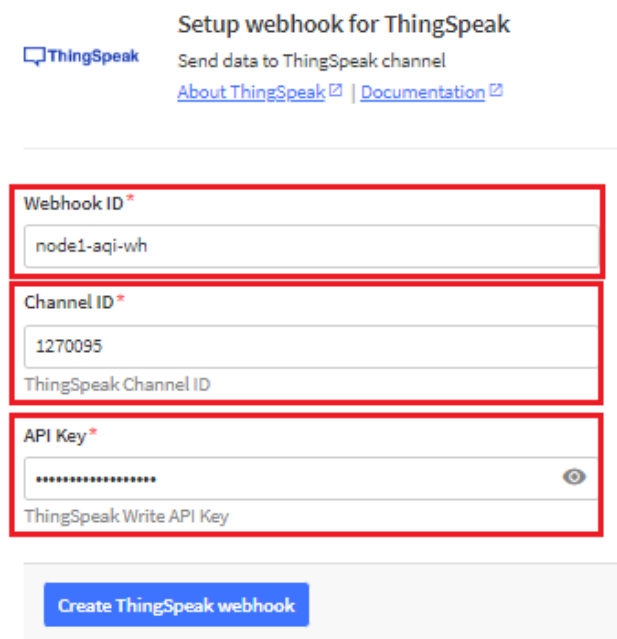
Για τη διασύνδεση του TTS με τις ΙοΤ πλατφόρμες, απαιτείται η δημιουργία καναλιού συλλογής δεδομένων για κάθε συνδεδεμένη συσκευή, σε κάθε μια από αυτές. Για την υλοποίηση του Webhook με το Thingspeak απαιτούνται το αναγνωριστικό καναλιού και το Write API Key που βρίσκονται στην καρτέλα *API Keys* (Σχήμα 87).



Σχήμα 87: Σελίδα API Keys καναλιού στο Thingspeak.

Είναι σημαντικό να αναφερθεί ότι πριν από την υλοποίηση της ενσωμάτωσης του Webhook, πρέπει να συνταχθεί ο κώδικας της συνάρτησης για την αποκωδικοποίηση του ωφέλιμου φορτίου ανερχόμενης σύνδεσης, όπως αναφέρθηκε στην προηγούμενη ενότητα.

Μετά τη δημιουργία του καναλιού στο Thingspeak, ο χρήστης πρέπει να επιστρέψει στην κονσόλα του TTS και να δημιουργήσει μια ενοποίηση Webhook και να συμπληρώσει τα στοιχεία του καναλιού (Σχήμα 88). Στο πεδίο *Webhook ID* εισάγεται ένα περιγραφικό όνομα για την ενσωμάτωση, στο πεδίο *Channel ID* εισάγεται το αναγνωριστικό του καναλιού στην πλατφόρμα Thingspeak και στο πεδίο *API Key*, το κλειδί εγγραφής *Write API Key*. Η διαδικασία ολοκληρώνεται πιέζοντας το κουμπί *Create Thingspeak webhook*.



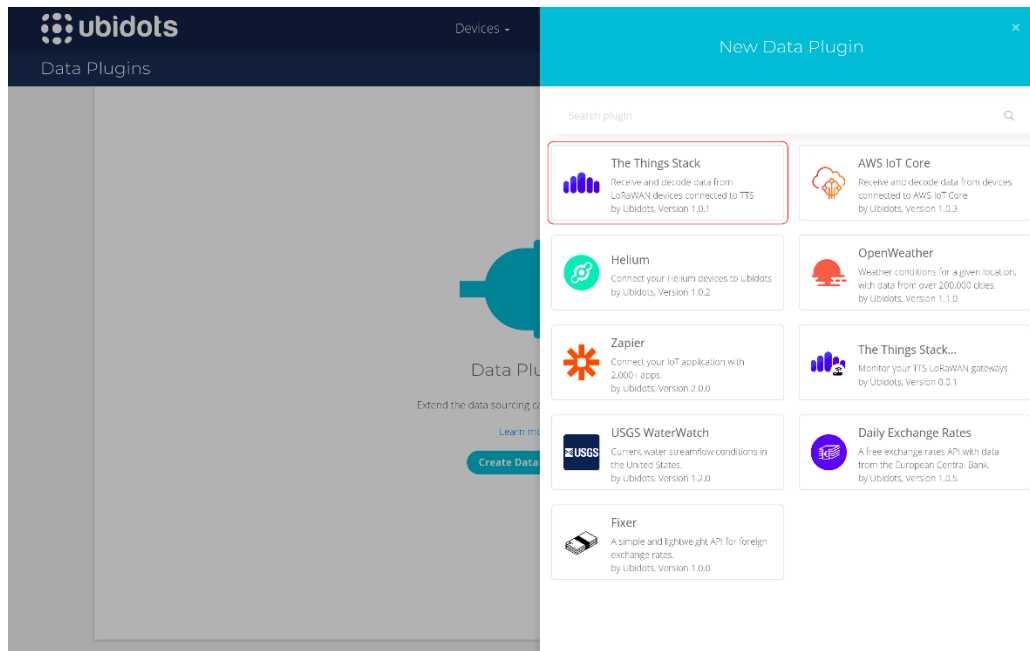
The screenshot shows a web interface titled "Setup webhook for ThingSpeak". At the top, there is a ThingsSpeak logo and the text "Send data to ThingSpeak channel". Below this are two links: "About ThingSpeak" and "Documentation". The main form consists of three input fields, each highlighted with a red border:

- The first field is labeled "Webhook ID*" and contains the text "node1-aqi-wh".
- The second field is labeled "Channel ID*" and contains the number "1270095". Below this field is the text "ThingSpeak Channel ID".
- The third field is labeled "API Key*" and contains a series of dots, indicating a masked password. To the right of this field is an eye icon for toggling visibility. Below this field is the text "ThingSpeak Write API Key".

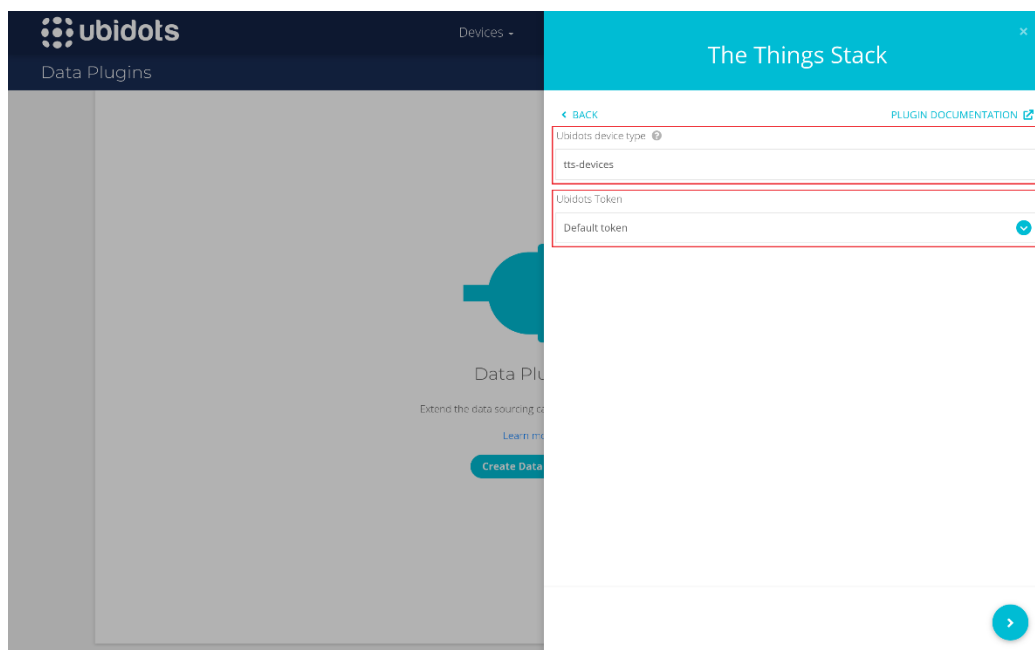
At the bottom of the form is a blue button with the text "Create ThingSpeak webhook".

Σχήμα 88: Καρτέλα ρύθμισης Webhook Thingspeak.

Για την υλοποίηση του Webhook με το Ubidots απαιτείται η δημιουργία μιας ιδιωτικής διεύθυνσης URL τελικού σημείου για την παραλαβή του ωφέλιμου φορτίου (JSON) από το The Things Stack. Η διαδικασία αυτή υλοποιείται με την ενεργοποίηση του plugin *The Things Stack* στην πλατφόρμα Ubidots. Ο χρήστης συνδέεται στο λογαριασμό του στην πλατφόρμα Ubidots και στην καρτέλα *Devices* στο επάνω μέρος του πίνακα ελέγχου, επιλέγει *Plugins*. Πατώντας το κουμπί για την προσθήκη νέου plugin προκύπτει ένα αναδυόμενο παράθυρο με τα διαθέσιμα plugins. Όταν παρουσιαστούν οι διαθέσιμες προσθήκες, επιλέγει το *The Things Stack* (Σχήμα 89). Στη συνέχεια, εισάγεται ένα περιγραφικό όνομα στο πεδίο Ubidots device type για έναν τύπο συσκευής. Αυτός ο τύπος συσκευής επιτρέπει να πραγματοποιηθούν αλλαγές για όλες τις συσκευές που λαμβάνουν δεδομένα μέσω αυτού του plugin. Στο δεύτερο πεδίο επιλέγεται το Ubidots Token, το οποίο μπορεί να είναι το προεπιλεγμένο διακριτικό του λογαριασμού του χρήστη ή να δημιουργηθεί ένα νέο διακριτικό αποκλειστικά για αυτό το plugin (Σχήμα 90). Μετά την ολοκλήρωση ρύθμισης του plugin και την επιτυχή εκτέλεσή του, το Ubidots θα μπορεί να δημιουργεί αυτόματα μια συσκευή κάθε φορά που λαμβάνεται ένα ωφέλιμο φορτίο από μια νέα συσκευή.

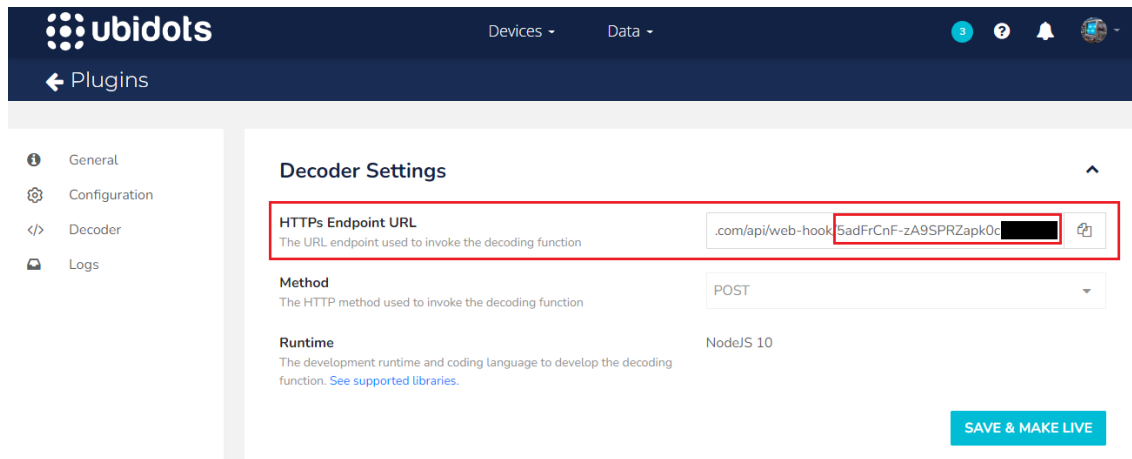


Σχήμα 89: Επιλογή The Things Stack plugin στην πλατφόρμα Ubidots.



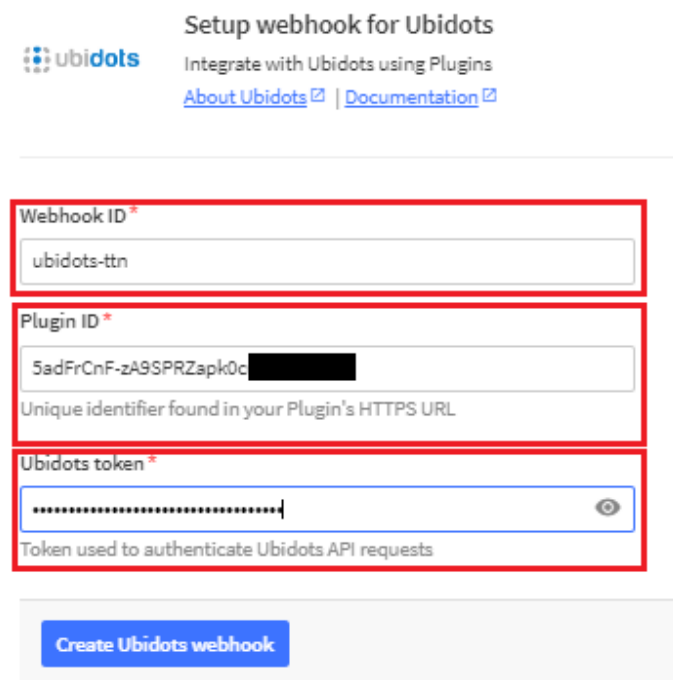
Σχήμα 90: Ρύθμιση The Thing Stack plugin στο Ubidots.

Για την ολοκλήρωση του *Integration* με το The Things Stack, είναι απαραίτητη η καταχώρηση του αναγνωριστικού του plugin και του προαναφερθέντος διακριτικού. Για την ανεύρεση του αναγνωριστικού, ο χρήστης επιλέγει το plugin που δημιουργήθηκε και μεταβαίνει στην καρτέλα *Decoder* από το αριστερό μενού. Το αναγνωριστικό του plugin είναι διαθέσιμο ως μέρος της διεύθυνσης URL τελικού σημείου HTTP (Σχήμα 91).



Σχήμα 91: Αναγνωριστικό του plugin The Things Stack.

Μετά την ολοκλήρωση της ρύθμισης του plugin στο Ubidots, ο χρήστης πρέπει να επιστρέψει στην κονσόλα του TTS και να δημιουργήσει μια ενοποίηση Webhook και να συμπληρώσει τα στοιχεία του plugin (). Στο πεδίο *Webhook ID* εισάγεται ένα περιγραφικό όνομα για την ενσωμάτωση, στο πεδίο *Plugin ID* εισάγεται το αναγνωριστικό και στο πεδίο *Ubidots token*, το προεπιλεγμένο διακριτικό του λογαριασμού του χρήστη. Η διαδικασία ολοκληρώνεται πιέζοντας το κουμπί *Create ubidots webhook*. Μετά την ολοκλήρωση των ρυθμίσεων, ο χρήστης μπορεί να δει τη συσκευή να εμφανίζεται στη λίστα τελικών συσκευών της πλατφόρμας Ubidots, μόλις αυτή αποστείλει ένα μήνυμα ανερχόμενης σύνδεσης.



Σχήμα 92: Καρτέλα ρύθμισης Webhook Ubidots.

9.4 Επίπεδο IoT platforms

Όπως αναφέρθηκε και στην ενότητα «Γενική αρχιτεκτονική μεθοδολογίας», οι κύριες αρμοδιότητες αυτού του επιπέδου είναι η αποθήκευση των δεδομένων που λαμβάνονται από διαφορετικά είδη συνδεδεμένων συσκευών, η επεξεργασία και η ανάλυσή τους. Τα δεδομένα αυτά είναι συνήθως είναι προσβάσιμα, μπορούν να ανακτηθούν μέσω RESTfull HTTP APIs ή μέσω του πρωτοκόλλου MQTT και να προωθηθούν σε υπηρεσίες ή εφαρμογές που ανήκουν σε ανώτερα επίπεδα. Οι IoT πλατφόρμες Thingspeak και Ubidots υλοποιούν το πρωτόκολλο MQTT, διαθέτοντας την κατάλληλη υποδομή για την λήψη και την αποστολή των δεδομένων.

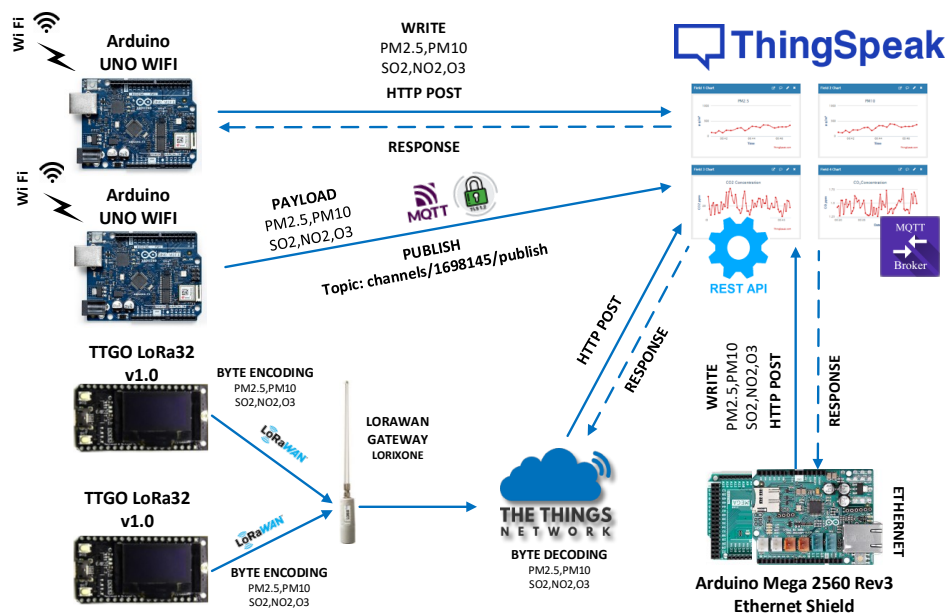
Ο MQTT Broker είναι το κεντρικό σημείο επικοινωνίας και είναι υπεύθυνος για την αποστολή όλων των μηνυμάτων μεταξύ των αποστολέων και των νόμιμων παραληπτών. Ως πελάτες χαρακτηρίζονται οι συσκευές, οι υπηρεσίες και οι εφαρμογές που συνδέονται με τον Broker και μπορούν να δημοσιεύουν ή να εγγραφούν σε Topics για να αποκτήσουν πρόσβαση στις πληροφορίες. Ένα Topic περιέχει τις πληροφορίες δρομολόγησης για τον Broker. Κάθε πελάτης που θέλει να αποστείλει μηνύματα, τα δημοσιεύει σε ένα συγκεκριμένο θέμα και κάθε πελάτης που θέλει να λαμβάνει μηνύματα εγγράφεται σε ένα συγκεκριμένο θέμα. Ο Broker παραδίδει όλα τα μηνύματα με το αντίστοιχο Topic στους κατάλληλους πελάτες. Με αυτόν τον τρόπο, η σύνδεση μεταξύ του πελάτη και του Broker παραμένει ανοιχτή, αλλά τα δεδομένα αποστέλλονται μόνο όταν είναι απαραίτητο, εξοικονομώντας μπαταρία, εύρος ζώνης δικτύου και βελτιώνοντας την εμπειρία σε πραγματικό χρόνο.

Στο πλαίσιο υλοποίησης της εργασίας χρησιμοποιήθηκε η δυνατότητα επικοινωνίας μέσω του πρωτοκόλλου MQTT που προσφέρουν οι IoT πλατφόρμες. Με τον τρόπο αυτό αντιμετωπίστηκε η ετερογένεια της δικτύωσης και της χρήσης διαφορετικών πρωτοκόλλων για την αποστολή της πληροφορίας από τις IoT συσκευές. Οι πληροφορίες συλλέγονται και προωθούνται στους αποδέκτες έχοντας συγκεκριμένο μοντέλο διαμόρφωσης της πληροφορίας (JSON αντικείμενο), ενισχύοντας με αυτό τον τρόπο την διαλειτουργικότητα των συστημάτων. Η έμφαση στη διαλειτουργικότητα των πληροφοριών και η προσαρμοστικότητα των IoT πλατφορμών, καθιστούν το πρωτόκολλο MQTT την καλύτερη λύση ως ένα μέσο διασύνδεσης με τα ανώτερα επίπεδα της αρχιτεκτονικής.

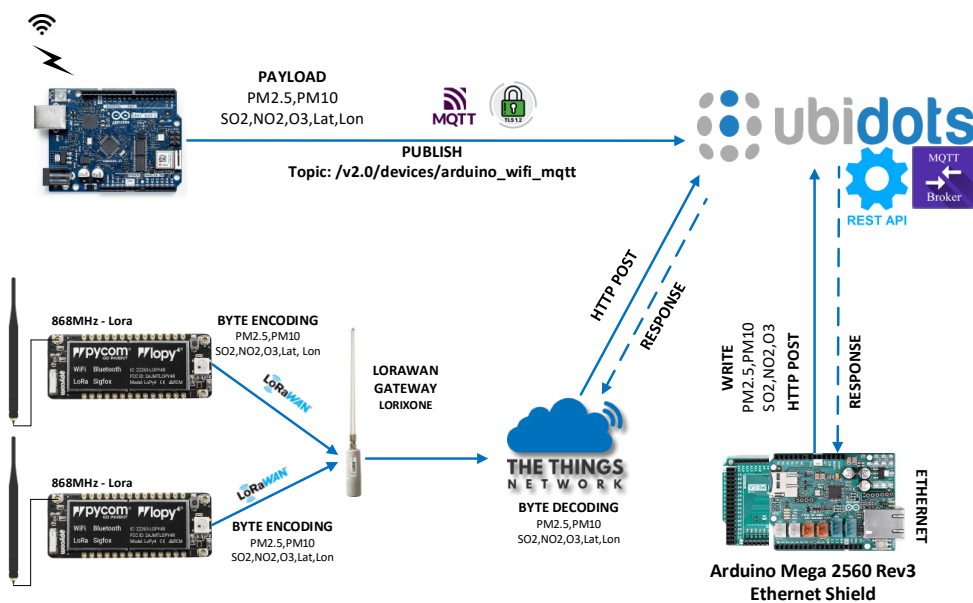
Κατά την υλοποίηση της εφαρμογής «Πράσινη Διαδρομή» χρησιμοποιήθηκαν εννέα (9) διαφορετικές IoT συσκευές. Οι πέντε (5) από αυτές αποστέλλουν τα περιβαλλοντικά και γεωχωρικά τους δεδομένα στην πλατφόρμα Thingspeak (Σχήμα 93), ενώ οι υπόλοιπες τέσσερις (4) στην πλατφόρμα Ubidots (Σχήμα 94). Η πλατφόρμα Thingspeak υλοποιεί τη λειτουργικότητα δημοσίευσης και εγγραφής με επίπεδο ποιότητας υπηρεσίας QoS=0, σε αντίθεση με το Thingspeak, η πλατφόρμα Ubidots υποστηρίζει επίπεδο ποιότητας υπηρεσίας έως και 1. Το QoS ορίστηκε στο μηδέν για τις δυο πλατφόρμες. Οι πλατφόρμες παρέχουν τη δυνατότητα απόκτησης πρόσβασης στο API μέσω απλού ή ασφαλούς MQTT με TLS, χρησιμοποιώντας τα ακόλουθα τελικά σημεία με βάση τον τύπο του λογαριασμού που επιλέγεται κάθε φορά (Πίνακας 16).

Πίνακας 16: Πίνακας τελικών σημείων MQTT στις IoT πλατφόρμες.

Πλατφόρμα	Τύπος Σύνδεσης	Διεύθυνση Broker	Πόρτα	Κρυπτογράφηση
Thingspeak	TCP	mqtt.thingspeak.com	1883	Όχι
Thingspeak	TCP	mqtt.thingspeak.com	8883	TLS
Ubidots	TCP	industrial.api.ubidots.com	1883	Όχι
Ubidots	TCP	industrial.api.ubidots.com	8883	TLS



Σχήμα 93: Διάγραμμα διασυνδεδεμένων συσκευών με την πλατφόρμα Thingspeak.



Σχήμα 94: Διάγραμμα διασυνδεδεμένων συσκευών με την πλατφόρμα Ubidots.

Κατά την υλοποίηση της διασύνδεσης των Publishers και των Subscribers με τους αντίστοιχους Brokers των IoT πλατφορμών, υιοθετήθηκε η χρήση του MQTT πρωτοκόλλου με TLS, εξασφαλίζοντας την κρυπτογράφηση των δεδομένων και την αποφυγή της έκθεσης του διακριτικού API και των δεδομένων των αισθητήρων σε τρίτους. Τα πιστοποιητικά μπορούν να εγκατασταθούν στις προγραμματιζόμενες συσκευές IoT και βρίσκονται αναρτημένα στις ιστοσελίδες των πλατφορμών.

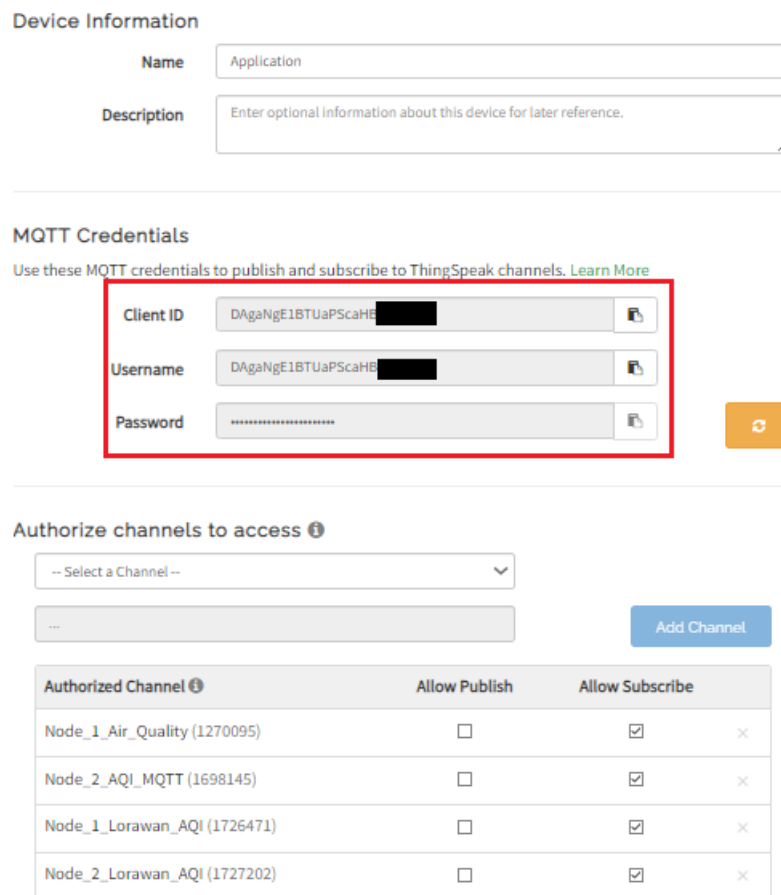
Για την πρόσβαση στα κανάλια των συσκευών της πλατφόρμας Thingspeak, συμπεριλαμβανομένων των διαπιστευτηρίων, ο χρήστης πρέπει να μεταβεί στην πλατφόρμα και να ορίσει την πρόσβαση σε αυτά από την επιλογή *Devices MQTT*. Κάθε συσκευή IoT ή εφαρμογή διαμορφώνεται με τα αντίστοιχα διαπιστευτήρια που είναι απαραίτητα για την επικοινωνία MQTT με την πλατφόρμα ThingSpeak και την εξουσιοδοτημένη πρόσβαση συγκεκριμένων καναλιών. Ο χρήστης από το μενού *Devices MQTT* επιλέγει την προσθήκη νέας συσκευής πιέζοντας το πλήκτρο *Add a new device* και συμπληρώνει τα απαραίτητα στοιχεία στο αναδυόμενο παράθυρο (Σχήμα 95). Στο πεδίο *Name* εισάγεται το περιγραφικό όνομα της συσκευής ή της εφαρμογής που θα συνδεθεί. Στην ενότητα *Εξουσιοδότηση καναλιών*, επιλέγεται ένα κανάλι από το μενού λίστας και στη συνέχεια προστίθεται με την πίεση του κουμπιού *Add Channel*. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να συνδεθούν όλα τα επιθυμητά κανάλια στη λίστα εξουσιοδοτημένων καναλιών. Για κάθε ένα από τα κανάλια που επελέγησαν ορίζεται η δυνατότητα δημοσίευσης δεδομένων ή εγγραφής σε αυτό.

The screenshot shows the 'Add a new device' form in the Thingspeak platform. The form is titled 'Add a new device' and contains several sections. The 'Device Information' section has a 'Name*' field with 'Application' entered and a 'Description' field with placeholder text. The 'Authorize channels to access' section shows a dropdown menu with 'Node_1_Lorawan_AQI (1726471)' selected. Below this is a table of channels with checkboxes for 'Allow Subscribe'. The 'Add Device' button is highlighted in green.

Channel Name	Allow Subscribe
Node_1_Lorawan_AQI (1726471)	<input checked="" type="checkbox"/>
Node_1_Air_Quality (1270095)	<input checked="" type="checkbox"/>
Node_2_AQI_MQTT (1698145)	<input checked="" type="checkbox"/>

Σχήμα 95: Δημιουργία συσκευής MQTT στην πλατφόρμα Thingspeak.

Σε αυτό το σημείο, η πλατφόρμα ThingSpeak δημιουργεί μια λίστα διαπιστευτηρίων για τη συσκευή που περιλαμβάνει το αναγνωριστικό πελάτη, το όνομα χρήστη και τον κωδικό πρόσβασης (Σχήμα 96). Ο χρήστης μπορεί να προβάλλει και να αντιγράψει αυτά τα στοιχεία ή επιλέγοντας *Λήψη διαπιστευτηρίων* να τα αποθηκεύσει σε ένα τοπικό αρχείο.



Device Information

Name: Application

Description: Enter optional information about this device for later reference.

MQTT Credentials

Use these MQTT credentials to publish and subscribe to ThingSpeak channels. [Learn More](#)

Client ID: DAgaNgE1BTUaPScAHE [redacted]

Username: DAgaNgE1BTUaPScAHE [redacted]

Password: [redacted]

Authorize channels to access

-- Select a Channel --

...

Add Channel

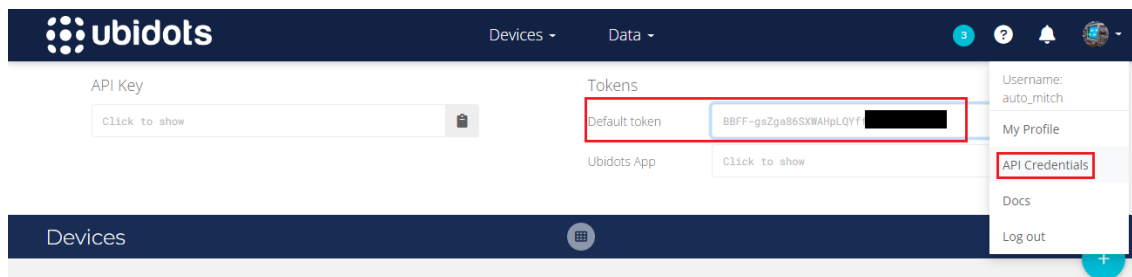
Authorized Channel	Allow Publish	Allow Subscribe	
Node_1_Air_Quality (1270095)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
Node_2_AQI_MQTT (1698145)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
Node_1_Lorawan_AQI (1726471)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×
Node_2_Lorawan_AQI (1727202)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	×

Σχήμα 96: Διαπιστευτήρια συσκευής MQTT στην πλατφόρμα Thingspeak.

Με την ολοκλήρωση της παραπάνω διαδικασίας η νέα συσκευή εμφανίζεται στη σελίδα *Devices MQTT*. Ο χρήστης μπορεί να αλλάξει τις ρυθμίσεις της συσκευής και να προσθέσει κανάλια σε αυτήν επιλέγοντας το κουμπί *Επεξεργασία*. Επίσης μπορεί να χρησιμοποιηθεί το αναγνωριστικό πελάτη, το όνομα χρήστη και ο κωδικός πρόσβασης στο πρόγραμμα-πελάτη MQTT, για την απόκτηση πρόσβασης στη συσκευή ThingSpeak MQTT για δημοσίευση και εγγραφή στα συνδεδεμένα κανάλια της.

Για την πλατφόρμα Ubidots, η διασύνδεση μέσω του πρωτοκόλλου MQTT για δημοσίευση ή εγγραφή είναι ευκολότερη διαδικασία. Για την αλληλεπίδραση με τον Broker Ubidots MQTT, απαιτείται η χρήση ενός μόνο διακριτικού. Το διακριτικό είναι ένα μοναδικό κλειδί που εξουσιοδοτεί μια συσκευή ή μια εφαρμογή να αλληλεπιδρά με το Ubidots API (Σχήμα 97). Για τη δημιουργία σύνδεσης με τον Broker MQTT, πρέπει να εκχωρηθεί ως όνομα χρήστη το διαπιστευτήριο ενώ δεν απαιτείται κάποιος κωδικός πρόσβασης. Εάν η σύνδεση είναι επιτυχής,

προσφέρεται η δυνατότητα δημοσίευσης και εγγραφής στα αντίστοιχα *Topics*, διαφορετικά ο Broker επιστρέφει ένα σφάλμα και αποσυνδέει τον πελάτη.



Σχήμα 97: Διαπιστευτήριο MQTT στην πλατφόρμα Ubidots.

9.5 Επίπεδο υπηρεσιών

9.5.1 Περιγραφή λειτουργίας υπηρεσίας συλλογής IoT δεδομένων

Το πρόγραμμα `Multi_Client_MQTT` αποτελεί μια εξατομικευμένη υπηρεσία που εκτελείται αδιάληπτα σε έναν διακομιστή του Πανεπιστημίου Αιγαίου και έχει συνταχθεί σε γλώσσα προγραμματισμού Python 3.9.1. Οι κύριες λειτουργίες της υπηρεσίας είναι η συλλογή των δεδομένων που αποστέλλουν οι IoT συσκευές στις πλατφόρμες Thingspeak και Ubidots, το φιλτράρισμα της ωφέλιμης πληροφορίας και η μορφοποίησή της σε μια ενιαία δομή. Η διαμόρφωση των δεδομένων εξυπηρετεί την προώθησή και την ευκολότερη διαχείρισή τους από τα ανώτερα επίπεδα της αρχιτεκτονικής. Οι λειτουργίες αυτές εκτελούνται σε πραγματικό χρόνο καθώς η συλλογή των δεδομένων από τις IoT πλατφόρμες πραγματοποιείται με την εγγραφή του προγράμματος πελάτη σε όλα τα *Topics* των IoT συσκευών, κάνοντας χρήση του πρωτοκόλλου MQTT. Η υπηρεσία συλλογής IoT δεδομένων μπορεί να πραγματοποιήσει τις προαναφερθείσες λειτουργίες εξυπηρετώντας πολλαπλούς χρήστες με ταυτόχρονες εγγραφές σε *Topics* διαφορετικών καναλιών επικοινωνίας.

Για την υλοποίηση του πρωτοκόλλου χρησιμοποιήθηκε το *Eclipse Paho*, ένα έργο ανοιχτού κώδικα με σκοπό την παροχή υψηλής ποιότητας υλοποιήσεις εργαλείων και βιβλιοθηκών για επικοινωνίες M2M και IoT. Το *Eclipse Paho* καλύπτει υλοποιήσεις πελατών MQTT για διάφορες πλατφόρμες και γλώσσες προγραμματισμού όπως Java, Python, Go κ.λ.π.. Το *Paho Python Client* παρέχει μια κλάση πελάτη με υποστήριξη τόσο για MQTT v3.1 όσο και για v3.1.1 σε Python 2.7 ή 3.x [54]. Επίσης παρέχει βοηθητικές λειτουργίες για την εύκολη δημοσίευση μεμονωμένων μηνυμάτων σε έναν διακομιστή MQTT.

Το πρόγραμμα της υπηρεσίας απαρτίζεται από επιμέρους αρχεία όπου υλοποιούν συναρτήσεις και με συνδυαστικό τρόπο επιτελούν τις παραπάνω λειτουργίες. Το αρχείο `multi_client_config.py` περιλαμβάνει όλες τις απαραίτητες πληροφορίες των IoT πλατφορμών, των ετικετών των δεδομένων και το σύνολο των διαπιστευτηρίων που απαιτούνται για την εγκαθίδρυση συνδέσεων με τους αντίστοιχους Brokers. Στα πλαίσια της εφαρμογής «Πράσινη Διαδρομή» χρησιμοποιήθηκαν εννιά (9) συσκευές IoT. Οι πέντε (5) από αυτές αποστέλλουν τα δεδομένα τους στην πλατφόρμα Thingspeak και οι υπόλοιπες τέσσερις (4) στο Ubidots. Εξαιτίας των περιορισμών

που θέτουν οι IoT πλατφόρμες στο πλήθος των καναλιών επικοινωνίας ανά λογαριασμό χρήστη, δημιουργήθηκαν τέσσερις (4) διαφορετικοί λογαριασμοί, δύο (2) στο Thingspeak και δυο (2) στο Ubidots. Στο αρχείο ρυθμίσεων περιλαμβάνονται οι διευθύνσεις των Brokers, οι θύρες επικοινωνίας, τα ονόματα των χρηστών και τα κανάλια επικοινωνίας που αντιστοιχούν σε αυτά (Σχήμα 98).

```
1 import os.path
2 import enum
3 # Using enum class create enumerations
4 class Platforms(enum.Enum):
5     Ubidots = "Ubidots"
6     Thingspeak = "Thingspeak"
7
8 class Platforms_Variables(enum.Enum):
9     Ubidots=["no2", "o3", "pm10", "pm2_5", "so2", "position"]
10    Thingspeak = {'field1': 'pm2.5', 'field2': 'pm10', 'field3': 'so2', 'field4': 'no2', 'field5': 'o3',
11                'field6': 'lat', 'field7': 'lng', 'status': 'AQI', 'created_at': 'timestamp'}
12
13 mqtt_keepalive = 60
14 # Ubidots: Maximum devices per account = 3
15 # Thingspeak: Maximum devices per account = 4
16 clients=[
17     {"broker_type":Platforms.Ubidots.value,"broker":"industrial.api.ubidots.com","port":8883,"username":"BBFF-gsZga86SXWAhPLQYff8S",
18     "pass":"","clientId":"Ubidots_1","devices":["aegean-lop4-node1","arduino_wifi_mqtt","arduino_eth_http"]},
19
20     {"broker_type":Platforms.Ubidots.value,"broker":"industrial.api.ubidots.com","port":8883,"username":"BBFF-PAjkojxVEYkeS1GX6gey",
21     "pass":"","clientId":"Ubidots_2","devices":["aegean-lop4-node2"]},
22
23 # DL Account
24 {"broker_type":Platforms.Thingspeak.value,"broker":"mqtt3.thingspeak.com","port":8883,"username":"DAgaNgE1BTUaPSc",
25 "pass":"gHzZYq6FG6Sng3R6", "clientId":"DAgaNgE1BTUaPSc", "channels":["1270095","1698145","1726471","1727202"]},
26
27 # icSD Account
28 {"broker_type":Platforms.Thingspeak.value,"broker":"mqtt3.thingspeak.com","port":8883,"username":"HSQvGg1IJzMQLxo",
29 "pass":"ehsENVZ/0ucYx9f", "clientId":"HSQvGg1IJzMQLxo", "channels":["1727524"]},
30 ]
```

Σχήμα 98: Απόσπασμα αρχείου ρυθμίσεων υπηρεσίας συλλογής IoT δεδομένων.

Κατά την διάρκεια εκκίνησης του κυρίου προγράμματος της υπηρεσίας (αρχείο *multi_client_MQTT.py*), πραγματοποιείται ανάγνωση της λίστας των πελατών που βρίσκονται στο αρχείο *multi_client_config.py*. Στη συνέχεια πραγματοποιούνται αιτήματα ασύγχρονης σύνδεσης με τους διακομιστές MQTT, σύμφωνα με τα στοιχεία επικοινωνίας των Brokers και των αντίστοιχων διαπιστευτηρίων κάθε χρήστη (Σχήμα 99). Στην υπηρεσία χρησιμοποιήθηκε η έκδοση του πρωτοκόλλου MQTT v3.1.1 και για την ασφαλή αποστολή και παραλαβή μηνυμάτων το πρωτόκολλο SSL/TLS.

Κάθε νέο μήνυμα που καταφτάνει στον υπολογιστή-πελάτη MQTT, τοποθετείται σε έναν buffer λήψης. Τα εισερχόμενα μηνύματα βρίσκονται στον buffer λήψης και παραμένουν εκεί μέχρι να διαβαστούν από το πρόγραμμα-πελάτη. Τα εξερχόμενα μηνύματα και οι επιβεβαιώσεις μηνυμάτων τοποθετούνται αντίστοιχα σε ξεχωριστούς buffers αποστολής. Για την αυτόματη ανάγνωση των buffers χρησιμοποιείται η ενσωματωμένη συνάρτηση *loop_forever()*, η οποία διαβάζει τους buffers λήψης και αποστολής και επεξεργάζεται τυχόν μηνύματα που προκύπτουν. Η μέθοδος *loop_forever()* μπλοκάρει το κύριο νήμα του προγράμματος αποτρέποντάς το να τερματιστεί, εκτελείται επ' αόριστο και χειρίζεται τις αυτόματες επανασυνδέσεις. Επειδή η υπηρεσία καλείται να διαχειριστεί περισσότερες από μια συνδέσεις πελάτη με τους Brokers, είναι απαραίτητη η κλήση και η εκκίνηση της συνάρτησης *loop_forever()* για κάθε σύνδεση πελάτη ξεχωριστά. Για να υλοποιηθεί η απαίτηση αυτή χρησιμοποιήθηκε μια δεξαμενή νημάτων (Thread pool Executor), για την αυτόματη δημιουργία τους, τη στιγμή που αυτά είναι απαραίτητα. Η

δεξαμενή είναι υπεύθυνη για τη διαχείριση των νημάτων, τα ελέγχει ακόμη και όταν δεν χρησιμοποιούνται, θέτοντάς τα σε αναμονή. Με αυτό τον τρόπο αποτρέπεται η άσκοπη κατανάλωση υπολογιστικών πόρων του διακομιστή στον οποίο εκτελείται η υπηρεσία. Τα νήματα της δεξαμενής επαναχρησιμοποιούνται μόλις ολοκληρωθεί η εργασία που εκτελούν, παρέχοντας προστασία έναντι μιας απροσδόκητης αποτυχίας της, όπως είναι η δημιουργία εξάιρεσης, χωρίς να επηρεάζεται το ίδιο το νήμα εργασίας. Η δεξαμενή νημάτων παρέχεται στην Python με την κλάση `ThreadPoolExecutor`.

```
72 def Create_connections():
73     logging.info("Start Create_connections")
74     for i in range(len(config.clients)):
75         # Extract Data from each client to prepare connections
76         mqtt_server_host = config.clients[i]["broker"]
77         mqtt_server_port = config.clients[i]["port"]
78         userName = config.clients[i]["username"]
79         password = config.clients[i]["pass"]
80         clientId = config.clients[i]["clientId"]
81         client = mqtt.Client(clientId, mqtt.MQTTv311)
82         config.clients[i]["client"] = client
83         try:
84             client.connect_async(host=mqtt_server_host, port=mqtt_server_port,
85                                 keepalive=config.mqtt_keepalive)
86             logging.info("{} is connected on {}:{}".format(config.clients[i]["clientId"],
87                                                            config.clients[i]["broker"], config.clients[i]["port"]))
88         except:
89             logging.error("Connection Fialed to broker ", mqtt_server_host)
90             continue
91         client.on_connect = on_connect
92         client.on_subscribe = on_subscribe
93         client.on_message = on_message
94         client.username_pw_set(userName, password)
95         client.tls_set(tls_version=mqtt.ssl.PROTOCOL_TLS)
96         logging.info("End Create_connections")
```

Σχήμα 99: Συνάρτηση υποβολής αιτημάτων ασύγχρονης σύνδεσης με τους διακομιστές MQTT.

Επιπρόσθετα, η μέθοδος `loop_forever()` ελέγχει τους buffers λήψης, εξετάζει τα μηνύματα και ανάλογα με τον τύπο τους, ενεργοποιεί την κατάλληλη μέθοδο επανάκλησης (callback functions). Στην περίπτωση, που λάβει ένα μήνυμα CONNACK, ενεργοποιεί την μέθοδο `on_connect()`. Αυτή η συνάρτηση καλείται μετά τη σύνδεση του πελάτη με τον αντίστοιχο Broker και πραγματοποιεί εγγραφές στα αντίστοιχα Topics για την παραλαβή των δεδομένων (Σχήμα 100). Για την πλατφόρμα Thingspeak, η δομή του Topic έχει τη μορφή `channels/<channelID>/subscribe` και η εγγραφή αφορά το σύνολο των ενημερώσεων της ροής του καναλιού. Το `channelID` είναι το αναγνωριστικό του καναλιού στην πλατφόρμα. Για την πλατφόρμα Ubidots η μορφή του Topic είναι `/v1.6/devices/{DEVICE_LABEL}/+`, με το `DEVICE_LABEL` να αποτελεί το μοναδικό αναγνωριστικό της συσκευής. Το σύμβολο `+` είναι χαρακτήρας μπαλαντέρ πολλαπλών επιπέδων του MQTT και επιτρέπει την εγγραφή σε πολλά Topics ταυτόχρονα. Το Ubidots αποστέλλει ξεχωριστά μηνύματα για κάθε μια από της μεταβλητές του καναλιού.

Όταν ένας πελάτης λάβει κάποιο μήνυμα από ένα Topic στο οποίο είχε κάνει εγγραφή, καλείται η συνάρτηση επανάκλησης `on_message()` (Σχήμα 101). Η μέθοδος αυτή επεξεργάζεται το

λαμβανόμενο μήνυμα και το μετατρέπει σε μια δομημένη μορφή ανεξάρτητα από ποια πλατφόρμα προέρχεται. Αρχικά, από το μήνυμα εξάγεται το Topic και η ωφέλιμη πληροφορία payload. Από το Topic αναγνωρίζεται η πλατφόρμα, το κανάλι επικοινωνίας και το όνομα της συσκευής από την οποία προήλθε το μήνυμα. Ανάλογα την πλατφόρμα καλείται η αντίστοιχη μέθοδος επεξεργασίας του payload. Η διαδικασία της επεξεργασίας περιλαμβάνει την εξαγωγή τιμών των περιβαλλοντικών και γεωχωρικών μεγεθών και την χρονοσήμανση εγγραφής τους στην πλατφόρμα. Η χρονοσήμανση υπόκειται σε επεξεργασία ώστε να μετατραπεί από ένα δυσανάγνωστο για το χρήστη timestamp, σε μια εύκολα αναγνώσιμη ημερομηνία και ώρα τοπικής ζώνης. Μερικά από τα παραδείγματα επεξεργασμένων μηνυμάτων παρουσιάζονται στο Σχήμα 102.

```
18 def on_connect(client, userdata, flags, rc):
19     logging.info("Result from connect: {}".format(mqtt.connack_string(rc)))
20     if rc == 0:
21         logging.info("[INFO] Connected to broker")
22         for i in range(len(config.clients)):
23             if config.clients[i]["client"] == client:
24                 if config.clients[i]["broker_type"] == config.Platforms.Ubidots.value:
25                     # Get devices' names in order to construct subscription topics
26                     devices = config.clients[i]["devices"]
27                     # Iterate devices' names and subscribe to the corresponding topic
28                     for dev in devices:
29                         client.subscribe(
30                             "/v1.6/devices/{}/+".format(dev), qos=0)
31                         logging.info(
32                             "Subscribed to topic: \"/v1.6/devices/{}/+\"".format(dev))
33                     break
34                 elif config.clients[i]["broker_type"] == config.Platforms.Thingspeak.value:
35                     # Get channels' id in order to construct subscription topics
36                     channels = config.clients[i]["channels"]
37                     # Iterate channels and subscribe to the corresponding topic
38                     for ch in channels:
39                         client.subscribe(
40                             "channels/{}/subscribe".format(ch), qos=0)
41                         logging.info(
42                             "Subscribed to topic: \"channels/{}/subscribe\"".format(ch))
43                     break
44             else:
45                 logging.error("[ERROR] Error, connection failed")
```

Σχήμα 100: Μέθοδος επανάκλησης on_connect().

Τέλος το πρόγραμμα της υπηρεσίας περιλαμβάνει σύστημα καταγραφής συμβάντων για την κατανόηση της ροής του και την αποσφαλμάτωση δυσλειτουργιών που ενδέχεται να προκύψουν κατά την παραγωγική λειτουργία. Η καταγραφή χρήσιμων δεδομένων από τις σωστές θέσεις, συμβάλλει στην εύκολη διόρθωση των σφαλμάτων, αλλά και στην χρησιμοποίηση των δεδομένων για την ανάλυση της απόδοσης της εφαρμογής. Η εφαρμογή δημιουργεί ένα το αρχείο καταγραφής *app.log* στο οποίο αποθηκεύει πληροφορίες για τις συνδέσεις στους Brokers, τις εγγραφές στα Topics των συσκευών στις IoT πλατφόρμες, τα εισερχόμενα μηνύματα πριν και μετά την επεξεργασία τους (Σχήμα 103).

```
52 def on_message(client, userdata, msg):
53     logging.info("Message received. Topic: {}. Payload: {}".format(msg.topic,
54 str(msg.payload.decode("utf-8")))
55     global msg_counter
56     msg_counter = msg_counter + 1
57     parsed_data = prd.get_data(msg.topic, str(msg.payload.decode("utf-8")))
58     if parsed_data is not None:
59         logging.info("Parsed_Message: {}".format(parsed_data))
60         if 'AQI' not in parsed_data:
61             logging.info("Ready to write Parsed_Message")
62             neo4j.write_msg(parsed_data)
63             logging.info("Parsed_Message was written")
64         global msg_parsed_counter
65         msg_parsed_counter = msg_parsed_counter + 1
66     logging.info("Total_msgs:{0}, Total_parsed_msgs:{1}".format(
67 msg_counter, msg_parsed_counter))
```

Σχήμα 101: Μέθοδος επανάκλησης on_message().

```
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'no2', 'value': '349.0', 'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'no2', 'value': '45.0', 'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'o3', 'value': '87.0', 'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'o3', 'value': '599.0', 'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'pm10', 'value': '117.0', 'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'pm10', 'value': '871.0', 'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'pm2.5', 'value': '741.0', 'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'pm2.5', 'value': '279.0', 'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'so2', 'value': '819.0', 'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'so2', 'value': '411.0', 'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'position', 'value': {'lat': '36.950396', 'lng': '26.977408'},
'timestamp': '2022-08-23T17:36:04'}
{'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'position', 'value': {'lat': '36.961808', 'lng': '26.956288'},
'timestamp': '2022-08-23T17:35:14'}
{'platform': 'Thingspeak', 'channel': '1698145', 'timestamp': '2022-08-23T17:48:53', 'pm2.5': '765.00 ', 'pm10': '124.00 ',
'so2': '756.00 ', 'no2': '680.00 ', 'o3': '636.00 ', 'lat': '36.960762 ', 'lng': '26.963938'}
{'platform': 'Thingspeak', 'channel': '1726471', 'timestamp': '2022-08-23T17:49:58', 'pm2.5': '113', 'pm10': '971', 'so2': '638',
'no2': '796', 'o3': '578', 'lat': '36.9544896', 'lng': '26.9836992'}
{'platform': 'Thingspeak', 'channel': '1727202', 'timestamp': '2022-08-23T17:50:15', 'pm2.5': '489', 'pm10': '540', 'so2': '387',
'no2': '119', 'o3': '695', 'lat': '36.9523008', 'lng': '26.9811712'}
```

Σχήμα 102: Παραδείγματα επεξεργασμένων μηνυμάτων.

```
2022-08-23 17:43:37,917 [MainThread - 21492] [INFO ] Start Create_connections
2022-08-23 17:43:37,917 [MainThread - 21492] [INFO ] Ubidots_1 is connected on industrial.api.ubidots.com:8883.
2022-08-23 17:43:37,931 [MainThread - 21492] [INFO ] Ubidots_2 is connected on industrial.api.ubidots.com:8883.
2022-08-23 17:43:37,942 [MainThread - 21492] [INFO ] DAGaNgE1BTUaPScAH8zCB4 is connected on mqtt3.thingspeak.com:8883.
2022-08-23 17:43:37,955 [MainThread - 21492] [INFO ] HSQvGgI1JzMQLxozEAQmD5g is connected on mqtt3.thingspeak.com:8883.
2022-08-23 17:43:37,967 [MainThread - 21492] [INFO ] End Create_connections
2022-08-23 17:43:39,035 [ThreadPool_0 - 20384] [INFO ] Result from connect: Connection Accepted.
2022-08-23 17:43:39,035 [ThreadPool_0 - 20384] [INFO ] [INFO] Connected to broker
2022-08-23 17:43:39,036 [ThreadPool_0 - 20384] [INFO ] Subscribed to topic: "/v1.6/devices/aegean-lop4-node1/f_cnt"
2022-08-23 17:43:39,036 [ThreadPool_0 - 20384] [INFO ] Subscribed to topic: "/v1.6/devices/arduino_wifi_mqtt/"
2022-08-23 17:43:39,037 [ThreadPool_0 - 20384] [INFO ] Subscribed to topic: "/v1.6/devices/arduino_eth_http/"
2022-08-23 17:43:39,039 [ThreadPool_1 - 4480] [INFO ] Result from connect: Connection Accepted.
2022-08-23 17:43:39,039 [ThreadPool_1 - 4480] [INFO ] [INFO] Connected to broker
2022-08-23 17:43:39,040 [ThreadPool_1 - 4480] [INFO ] Subscribed to topic: "/v1.6/devices/aegean-lop4-node2/"
2022-08-23 17:43:39,246 [ThreadPool_0 - 20384] [INFO ] I've subscribed with QoS: 0
2022-08-23 17:43:39,264 [ThreadPool_0 - 20384] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node1/f_cnt. Payload: {"value": 366.0,
2022-08-23 17:43:39,265 [ThreadPool_0 - 20384] [INFO ] Total_msgs:1, Total_parsed_msgs:0
2022-08-23 17:43:39,265 [ThreadPool_1 - 4480] [INFO ] I've subscribed with QoS: 0
2022-08-23 17:43:39,266 [ThreadPool_0 - 20384] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node1/f_port. Payload: {"value": 2.0,
2022-08-23 17:43:39,266 [ThreadPool_0 - 20384] [INFO ] Total_msgs:2, Total_parsed_msgs:0
2022-08-23 17:43:39,267 [ThreadPool_0 - 20384] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node1/no2. Payload: {"value": 349.0, "t
2022-08-23 17:43:39,267 [ThreadPool_0 - 20384] [INFO ] Parsed_Message: {'platform': 'Ubidots', 'device': 'aegean-lop4-node1', 'var': 'no2', 'valu
2022-08-23 17:43:39,268 [ThreadPool_0 - 20384] [INFO ] Ready to write Parsed_Message
2022-08-23 17:43:39,268 [ThreadPool_0 - 20384] [INFO ] creating session
2022-08-23 17:43:39,284 [ThreadPool_1 - 4480] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node2/f_cnt. Payload: {"value": 365.0,
2022-08-23 17:43:39,285 [ThreadPool_1 - 4480] [INFO ] Total_msgs:4, Total_parsed_msgs:0
2022-08-23 17:43:39,285 [ThreadPool_1 - 4480] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node2/f_port. Payload: {"value": 2.0, "t
2022-08-23 17:43:39,286 [ThreadPool_1 - 4480] [INFO ] Total_msgs:5, Total_parsed_msgs:0
2022-08-23 17:43:39,286 [ThreadPool_1 - 4480] [INFO ] Message received. Topic: /v1.6/devices/aegean-lop4-node2/no2. Payload: {"value": 45.0, "tim
2022-08-23 17:43:39,287 [ThreadPool_1 - 4480] [INFO ] Parsed_Message: {'platform': 'Ubidots', 'device': 'aegean-lop4-node2', 'var': 'no2', 'value
2022-08-23 17:43:39,287 [ThreadPool_1 - 4480] [INFO ] Ready to write Parsed_Message
2022-08-23 17:43:39,288 [ThreadPool_1 - 4480] [INFO ] creating session
2022-08-23 17:43:40,186 [ThreadPool_2 - 26684] [INFO ] Result from connect: Connection Accepted.
2022-08-23 17:43:40,187 [ThreadPool_2 - 26684] [INFO ] [INFO] Connected to broker
2022-08-23 17:43:40,188 [ThreadPool_2 - 26684] [INFO ] Subscribed to topic: "channels/1270095/subscribe"
2022-08-23 17:43:40,188 [ThreadPool_2 - 26684] [INFO ] Subscribed to topic: "channels/1698145/subscribe"
```

Σχήμα 103: Απόσπασμα αρχείου καταγραφής app.log.

9.5.2 Περιγραφή λειτουργίας υπηρεσίας συλλογής *OpenData* δεδομένων

Το πρόγραμμα *async_api_request* αποτελεί τη δεύτερη εξατομικευμένη υπηρεσία που εκτελείται αδιάληπτα στον διακομιστή του πανεπιστημίου Αιγαίου. Όπως και η προηγούμενη υπηρεσία, έχει συνταχθεί σε γλώσσα προγραμματισμού Python 3.9.1. Οι κύριες λειτουργίες της είναι η υποβολή HTTP αιτημάτων διεπαφής προγραμματισμού εφαρμογών API σε ανοιχτές πηγές περιβαλλοντικών δεδομένων, το φιλτράρισμα των απαντήσεων και η μορφοποίησή των δεδομένων σε μια ενιαία δομή. Η διαμόρφωση των δεδομένων εξυπηρετεί την ευκολότερη διαχείριση της πληροφορίας από τα ανώτερα επίπεδα της αρχιτεκτονικής. Οι λειτουργίες αυτές εκτελούνται σε προκαθορισμένο χρόνο (15 λεπτά), όμοιο με αυτόν που αποστέλλουν οι IoT συσκευές στις πλατφόρμες. Η υπηρεσία συλλογής ανοιχτών δεδομένων με ασύγχρονο τρόπο μπορεί να υποβάλλει ταυτόχρονα πολλαπλά αιτήματα σε διαφορετικούς διακομιστές υπηρεσιών και να διαβάσει τις αποκρίσεις. Με αυτόν το τρόπο εξασφαλίζεται η εύρυθμη λειτουργία του προγράμματος, αποφεύγοντας την καθυστέρηση υποβολής νέων αιτημάτων μέχρι να ληφθεί η απάντηση από το τελευταίο ερώτημα. Οι διεργασίες αυτές τύπου I/O Bound των οποίων η ταχύτητα ολοκλήρωσης εξαρτάται από το χρόνο που αφιερώνεται σε αναμονή για την ολοκλήρωση των λειτουργιών εισόδου/εξόδου. Η ταχύτητα ενός αιτήματος HTTP εξαρτάται από το δίκτυο, τη μνήμη, την ταχύτητα διακομιστή κ.λπ.

Ο ταυτόχρονος προγραμματισμός είναι μια μορφή υπολογισμού, όπου πολλαπλές εργασίες θα εκτελούνται ταυτόχρονα. Η Python έχει τρεις κύριες βιβλιοθήκες για ταυτόχρονο προγραμματισμό την *Multiprocessing*, την *Threading* και την *Asyncio*. Για την υλοποίηση της υπηρεσίας χρησιμοποιήθηκε η *Asyncio*. Η βιβλιοθήκη αυτή χρησιμοποιεί μόνο ένα νήμα και ένα μόνο πυρήνα του επεξεργαστή για την εκτέλεση πολλών εργασιών ταυτόχρονα (σε αυτήν την περίπτωση πολλαπλά αιτήματα HTML). Το *Asyncio* απαιτεί μεγάλες περιόδους αναμονής στις οποίες επιτρέπει σε άλλες λειτουργίες να εκτελούνται κατά τη διάρκεια αυτού του χρόνου διακοπής λειτουργίας. Οι ασύγχρονες ρουτίνες μπορούν να κάνουν «παύση» ενώ περιμένουν το τελικό τους αποτέλεσμα και να αφήνουν άλλες ρουτίνες να εκτελούνται εν τω μεταξύ. Ο ασύγχρονος κώδικας, μέσω του παραπάνω μηχανισμού, δίνει την αίσθηση της ταυτόχρονης χρήσης.

Το πρόγραμμα της υπηρεσίας απαρτίζεται από επιμέρους αρχεία όπου υλοποιούν συναρτήσεις και με συνδυαστικό τρόπο επιτελούν τις παραπάνω λειτουργίες. Το αρχείο *api_config.py* περιλαμβάνει όλες τις απαραίτητες πληροφορίες των απομακρυσμένων πηγών ανοικτών δεδομένων, για την υποβολή των ερωτημάτων. Οι πληροφορίες αυτές απαρτίζονται από το όνομα της υπηρεσίας, τη διεύθυνση URL του απομακρυσμένου διακομιστή, τα διαπιστευτήρια σύνδεσης και τις κεφαλίδες (Headers). Επιπρόσθετα, στο αρχείο αυτό αποθηκεύονται οι γεωγραφικές συντεταγμένες των σημείων που θα ανακριθούν τα περιβαλλοντικά δεδομένα. Τα στοιχεία αυτά περιλαμβάνουν το όνομα της περιοχής του γεωγραφικού σημείου, το γεωγραφικό μήκος και πλάτος, καθώς και το όνομα της υπηρεσίας από την οποία θα ανακτηθούν τα δεδομένα (Σχήμα 104).

Κατά την διάρκεια εκκίνησης του κυρίου προγράμματος της υπηρεσίας (αρχείο *async_api_request.py*), πραγματοποιείται μια HTTP συνεδρία πελάτη με την οποία διαχειρίζονται όλα τα αιτήματα. Στη συνέχεια για κάθε ένα γεωγραφικό σημείο καλείται η μέθοδος *create_req_url*, για τη δημιουργία του πλήρους URL του αιτήματος, σύμφωνα με τα στοιχεία που είναι αποθηκευμένα στο αρχείο των ρυθμίσεων (Σχήμα 105). Κάθε αίτημα τοποθετείται σε μια ουρά εργασιών για ασύγχρονη διεκπεραίωση, λαμβάνοντας τα αποτελέσματα μόλις ολοκληρωθούν (Σχήμα 106).

```
4 Loc = {'Agios_Savvas': {'lat': 36.94479897403746, 'lon': 26.98052565369983, 'api_name': 'open_weather'},
5 'Airport': {'lat': 36.96411328843375, 'lon': 26.940153270090374, 'api_name': 'ninjas_airq'},
6 'Myrties': {'lat': 36.990931578588416, 'lon': 26.93151331395161, 'api_name': 'open_weather'},
7 'Melitsaxas': {'lat': 36.986894208621614, 'lon': 26.928929475287934, 'api_name': 'ninjas_airq'},
8 'Pothia_01': {'lat': 36.94906497634183, 'lon': 26.982050079436924, 'api_name': 'open_weather'},
9 'Pothia_02': {'lat': 36.95297945129611, 'lon': 26.975620533540777, 'api_name': 'ninjas_airq'},
10 'Pothia_03': {'lat': 36.960876, 'lon': 26.971832, 'api_name': 'open_weather'},
11 'Agios_Athanasios': {'lat': 36.95664154400455, 'lon': 26.965767803949493, 'api_name': 'ninjas_airq'},
12 'Chora_01': {'lat': 36.9636846089928, 'lon': 26.960327032094618, 'api_name': 'open_weather'},
13 'Taxiarchis': {'lat': 36.96502031893271, 'lon': 26.95289955137949, 'api_name': 'ninjas_airq'},
14 'Port_Authority': {'lat': 36.94657487279934, 'lon': 26.984530254135713, 'api_name': 'open_weather'}
15 }
16 # Using enum class create enumerations
17
18 class SupportedApis(enum.Enum):
19
20     def __init__(self, api_name, url, key, headers):
21         self.api_name = api_name
22         self.url = url
23         self.key = key
24         self.headers = headers
25
26 WEATHERBIT = ('weatherbit', 'https://air-quality.p.rapidapi.com/current/airquality', 'f2f1f98281msh84645eabe6802dep112',
27 {'X-RapidAPI-Host': 'air-quality.p.rapidapi.com', 'X-RapidAPI-Key': 'f2f1f98281msh84645eabe6802dep112', 'c'})
28 IQAIR = ('iqair', 'http://api.airvisual.com/v2/nearest_city', 'f55e45be-30ef-416d-b31e', {})
29 NINJAS_AIRQ = ('ninjas_airq', 'https://api.api-ninjas.com/v1/airquality', 'SWVP4kvv853PkuYseZE',
30 {'X-API-Key': 'SWVP4kvv853PkuYseZE'})
31 OPEN_WEATHER = ('open_weather', 'http://api.openweathermap.org/data/2.5/air_pollution', '740af68f7f2cda5d', {})
```

Σχήμα 104: Απόσπασμα αρχείου ρυθμίσεων υπηρεσίας συλλογής OpenData δεδομένων.

```
3 def create_req_url(lat, lon, api_name):
4     url = ""
5     if api_name == apic.SupportedApis.WEATHERBIT.api_name:
6         url = '{0}?lat={1}&lon={2}&hours=1'.format(apic.SupportedApis.WEATHERBIT.url, lat, lon)
7         headers = apic.SupportedApis.WEATHERBIT.headers
8
9     elif api_name == apic.SupportedApis.IQAIR.api_name:
10        url = '{0}?lat={1}&lon={2}&key={3}'.format(apic.SupportedApis.IQAIR.url, lat, lon, apic.SupportedApis.IQAIR.key)
11        headers = apic.SupportedApis.IQAIR.headers
12
13    elif api_name == apic.SupportedApis.NINJAS_AIRQ.api_name:
14        url = '{0}?lat={1}&lon={2}&X-API-Key={3}'.format(apic.SupportedApis.NINJAS_AIRQ.url, lat, lon, apic.SupportedApis.NINJAS_AIRQ.key)
15        headers = apic.SupportedApis.NINJAS_AIRQ.headers
16
17    elif api_name == apic.SupportedApis.OPEN_WEATHER.api_name:
18        url = '{0}?lat={1}&lon={2}&appid={3}'.format(apic.SupportedApis.OPEN_WEATHER.url, lat, lon, apic.SupportedApis.OPEN_WEATHER.key)
19        headers = apic.SupportedApis.OPEN_WEATHER.headers
20
21    return [url, headers]
```

Σχήμα 105: Μέθοδος δημιουργίας URL αιτημάτων.

```
26 async def main():
27     async with aiohttp.ClientSession() as session:
28         tasks = []
29
30         for val in apic.Loc.items():
31             [url, headers] = apim.create_req_url(val[1]['lat'], val[1]['lon'], val[1]['api_name'])
32             tasks.append(asyncio.ensure_future(get_data(session, url, headers, val[0], val[1]['api_name'],
33 | val[1]['lat'], val[1]['lon'])))
34         original_data = await asyncio.gather(*tasks)
35
36     for data in original_data:
37         logging.info("Ready to write API Request Data")
38         neo4j.write_api_data(data)
39         logging.info("API Request Data was written")
```

Σχήμα 106: Συνάρτηση κύριου προγράμματος υπηρεσίας.

Οι ουρά εργασιών εξυπηρετείται από το βρόγχο συμβάντων που είναι ο πυρήνας για κάθε asyncio εφαρμογή. Οι βρόχοι συμβάντων εκτελούν ασύγχρονες εργασίες, επανακλήσεις, λειτουργίες I/O δικτύου και υποδιεργασίες. Οι απαντήσεις των ερωτημάτων λαμβάνονται με τη

μορφή JSON αντικειμένων τα οποία επεξεργάζονται μέσω της μεθόδου `parse_api()`, για την εξαγωγή και διατήρηση μόνο των ωφέλιμων πληροφοριών (Σχήμα 107). Η μέθοδος ανάλογα με το όνομα της απομακρυσμένης υπηρεσίας αποκλείει περιττές πληροφορίες και διατηρεί της συγκεντρώσεις των προτιμητέων αερίων και αιωρούμενων σωματιδίων για τον προσδιορισμό του δείκτη ποιότητας αέρα. Στα στοιχεία αυτά προστίθενται το όνομα του γεωγραφικού σημείου, οι συντεταγμένες του, το όνομα της υπηρεσίας και η χρονοσήμανση. Η τελευταία υπόκειται σε επεξεργασία ώστε να μετατραπεί από ένα δυσανάγνωστο για το χρήστη timestamp, σε μια εύκολα αναγνώσιμη ημερομηνία και ώρα τοπικής ζώνης.

```
24 async def parse_api(data,name,api_name,lat,long,ts):
25
26     new_data={}
27     if api_name=='ninjas_airq':
28         exclude_data=['overall_aqi','CO']
29         for key,values in data.items():
30             if key not in exclude_data:
31                 new_data.update({key.casefold():values['concentration']})
32     if api_name=='open_weather':
33         exclude_data=['no','co','nh3']
34         for key,values in data['list'][0].items():
35             if (key=='components'):
36                 for key2,value2 in data['list'][0][key].items():
37                     if key2 not in exclude_data:
38                         if key2=="pm2_5":
39                             new_data.update({"pm2.5":value2})
40                         else:
41                             new_data.update({key2:value2})
42     if api_name=='weatherbit':
43         exclude_data=['aqi','co','mold_level','pollen_level_grass','pollen_level_tree',
44                     'pollen_level_weed','predominant_pollen_type']
45         for key,values in data['data'][0].items():
46             if key not in exclude_data:
47                 new_data.update({key:values})
48     new_data.update({'name':name})
49     new_data.update({'api_name':api_name})
50     new_data.update({'lat':lat})
51     new_data.update({'long':long})
52     new_data.update({'timestamp':ts})
53     return new_data
```

Σχήμα 107: Μέθοδος `parse_api()`.

Μερικά από τα παραδείγματα επεξεργασμένων μηνυμάτων παρουσιάζονται στο Σχήμα 108. Τέλος το πρόγραμμα της υπηρεσίας περιλαμβάνει σύστημα καταγραφής συμβάντων για την αποσφαλμάτωση δυσλειτουργιών που ενδέχεται να προκύψουν κατά την παραγωγική λειτουργία. Η μορφή των καταγραφών περιλαμβάνει την ακριβή ημερομηνία και ώρα του συμβάντος, το επίπεδο καταγραφής που υποδηλώνει τη σοβαρότητά του, καθώς και κάποια επιπλέον στοιχεία που βοηθούν στην περιγραφή του. Η εφαρμογή δημιουργεί ένα το αρχείο καταγραφής `apis_req.log` στο οποίο αποθηκεύει τις πληροφορίες των συμβάντων.

```
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Port_Authority', 'api_name': 'open_weather', 'lat': 36.94657487279934, 'long': 26.984530254135713, 'timestamp': '2022-11-29T00:04:21'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Pothia_03', 'api_name': 'open_weather', 'lat': 36.960876, 'long': 26.971832, 'timestamp': '2022-11-29T00:04:21'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Melitsaxas', 'api_name': 'ninjas_airq', 'lat': 36.986894208621614, 'long': 26.928929475287934, 'timestamp': '2022-11-29T00:04:22'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Pothia_02', 'api_name': 'ninjas_airq', 'lat': 36.95297945129611, 'long': 26.975620533540777, 'timestamp': '2022-11-29T00:04:22'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Taxiarchis', 'api_name': 'ninjas_airq', 'lat': 36.96502031893271, 'long': 26.95289955137949, 'timestamp': '2022-11-29T00:04:22'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Agios_Athanasios', 'api_name': 'ninjas_airq', 'lat': 36.95664154400455, 'long': 26.965767803949493, 'timestamp': '2022-11-29T00:04:22'}
{'no2': 1.41, 'o3': 67.95, 'so2': 1.88, 'pm2.5': 3.09, 'pm10': 3.8, 'name': 'Airport', 'api_name': 'ninjas_airq', 'lat': 36.96411328843375, 'long': 26.940153270090374, 'timestamp': '2022-11-29T00:04:22'}
```

Σχήμα 108: Παραδείγματα επεξεργασμένων μηνυμάτων υπηρεσίας `async_api_request`.

Το κύριο πρόγραμμα εκτελείται κάθε δεκαπέντε (15) λεπτά από τον `AsyncIOScheduler`, ο οποίος είναι κατάλληλος για περιπτώσεις που η εφαρμογή εκτελείται σε ένα βρόγχο `asyncio` καθώς εξοικονομεί πόρους για την εκτέλεση μια νέας διαδικασίας ή ενός νήματος (Σχήμα 109).

```
11 if __name__ == '__main__':
12
13     logFormatter = logging.Formatter("%(asctime)s [%(levelname)-5.5s] %(message)s")
14     rootLogger = logging.getLogger('apscheduler')
15     fileHandler = logging.FileHandler('scheduler.log')
16     fileHandler.setFormatter(logFormatter)
17     rootLogger.addHandler(fileHandler)
18     consoleHandler = logging.StreamHandler()
19     consoleHandler.setFormatter(logFormatter)
20     rootLogger.addHandler(consoleHandler)
21     rootLogger.setLevel(logging.DEBUG)
22
23     scheduler = AsyncIOScheduler()
24     scheduler.add_job(apireq.main, 'interval', minutes=15)
25     scheduler.start()
26
27     try:
28         asyncio.get_event_loop().run_forever()
29     except (KeyboardInterrupt, SystemExit) as err:
30         logging.exception(err)
31         scheduler.shutdown()
```

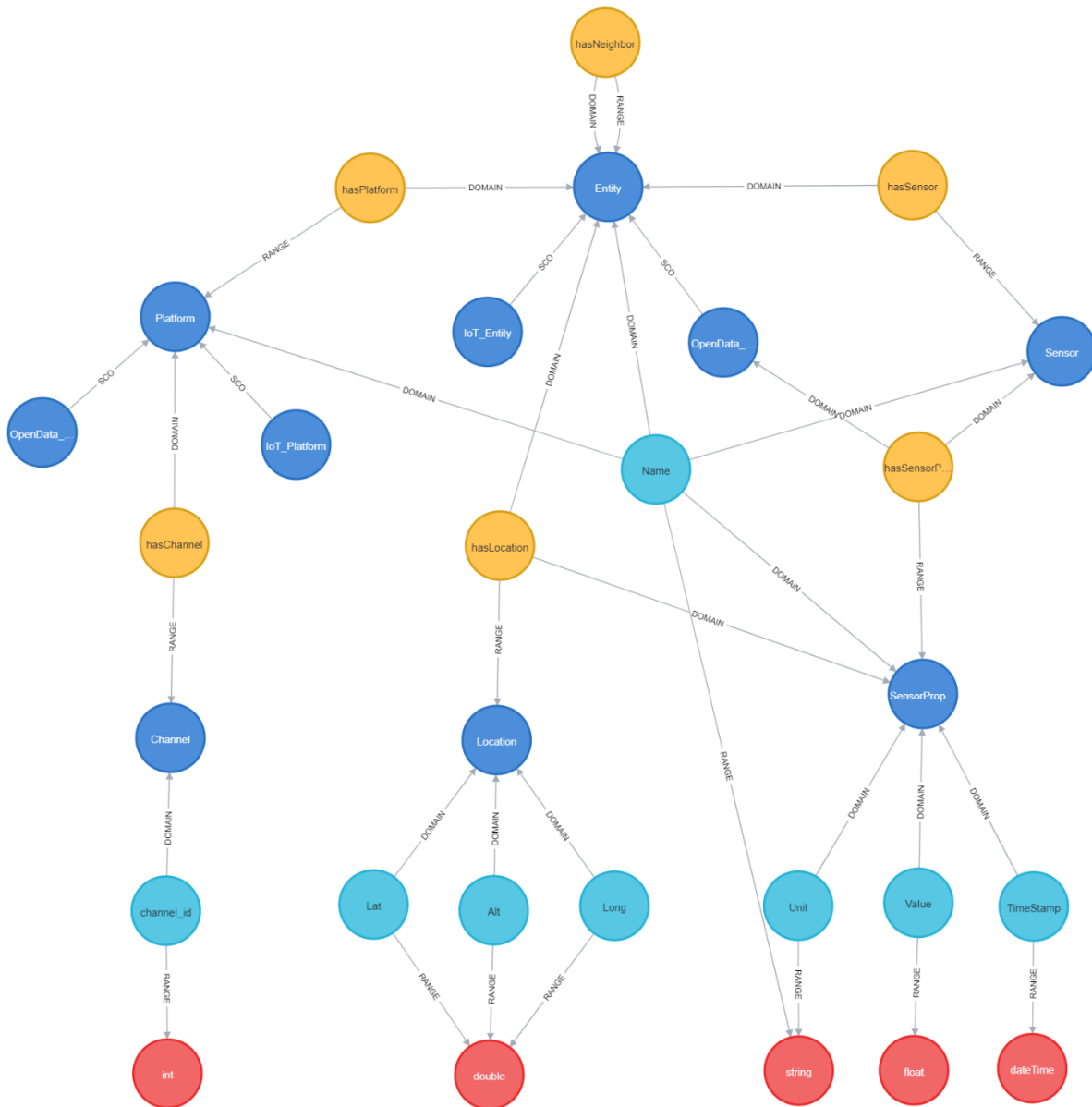
Σχήμα 109: Κώδικας προγραμματισμένης εκτέλεσης της εφαρμογής `async_api_request`.

9.6 Επίπεδο Σημασιολογίας

9.6.1 Οντολογία εφαρμογής

Στην ενότητα αυτή παρουσιάζεται η οντολογία που χρησιμοποιήθηκε για την μοντελοποίηση των δεδομένων που σχετίζονται με τον περιβαλλοντικό τομέα εφαρμογής της εργασίας. Η οντολογία (Σχήμα 110) μοντελοποιεί τα δεδομένα των μετρούμενων μεγεθών που παράγονται από τις IoT συσκευές και τις πλατφόρμες ανοικτών δεδομένων. Επιπρόσθετα, μοντελοποιούνται και οι γεωχωρικές πληροφορίες είτε των σταθερών σταθμών, είτε των μετρούμενων μεγεθών κινητών κόμβων ως μελλοντική επέκταση του

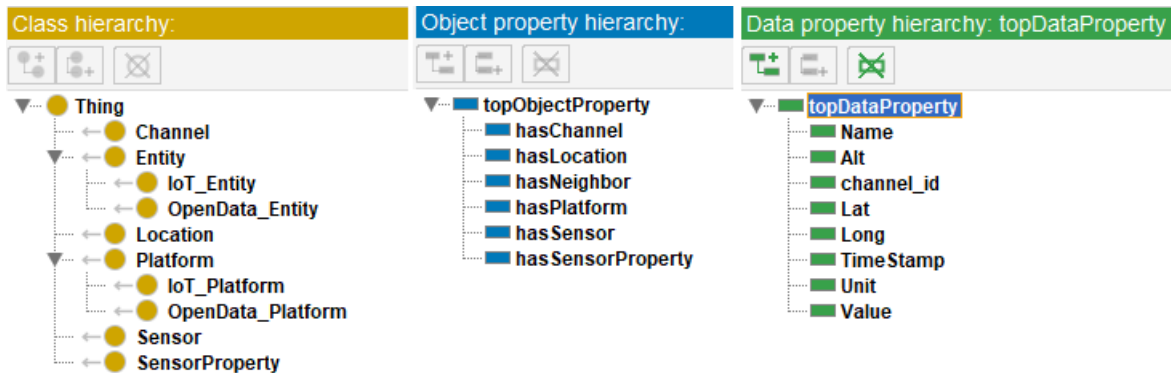
συστήματος. Η προτεινόμενη οντολογία αποτελεί μια «ελαφριά» έκδοση της περιγραφής των πόρων και των οντοτήτων, συμβάλλοντας στην μείωση του υπολογιστικού και επεξεργαστικού χρόνου, κατά τη φάση της αναζήτησης. Η απλότητά της την καθιστά κατάλληλη, να χρησιμοποιηθεί ως έχει ή ακόμη και να επεκταθεί για την επισημείωση δεδομένων διαφορετικών τομέων εφαρμογής.



Σχήμα 110: Οντολογία Εφαρμογής.

Η κύρια κλάση της οντολογίας είναι η *Entity*, η οποία αναπαριστά οποιαδήποτε ετερογενή πηγή δεδομένων. Από την κλάση αυτή υλοποιούνται δυο υποκλάσεις, η *IoT_Entity* και η *OpenData_Entity*. Η πρώτη αναπαριστά πραγματικούς IoT κόμβους, ενώ η δεύτερη αναπαριστά εικονικούς κόμβους με δεδομένα από ανοικτές πλατφόρμες, όπως αναφέρθηκαν στην ενότητα « Επίπεδο πηγής ανοικτών δεδομένων». Επιπρόσθετα, η οντολογία επιτρέπει την παρουσίαση πλατφορμών μέσω της κλάσης *Platform* για την περιγραφή της προέλευσης της πληροφορίας. Η

κλάση *Platform* υλοποιεί δύο υποκλάσεις την *IoT_Platform* για τις IoT πλατφόρμες (π.χ. Thingspeak, Ubidots) και την *OpenData_Platform* για τις πλατφόρμες ανοικτών δεδομένων (π.χ. iqair, open_weather, weatherbit). Στο Σχήμα 111 παρουσιάζεται η ιεραρχία των κλάσεων, των σχέσεων και των ιδιοτήτων των δεδομένων, όπως δημιουργήθηκαν στο ανοικτό λογισμικό κατασκευής οντολογιών Protégé [55]. Ο Πίνακας 17 παρουσιάζει αναλυτικά τις κλάσεις που περιλαμβάνονται στην οντολογία, τις κλάσεις από τις οποίες κληρονομούν τις ιδιότητές τους και σε ποια οντότητα αντιστοιχούν.



Σχήμα 111: Ιεραρχία κλάσεων και ιδιότητες οντολογίας.

Πίνακας 17: Κλάσεις οντολογίας.

Κλάση	Υπερκλάση	Περιγραφή
Entity	-	Γενική μορφή οντότητας συστήματος
IoT_Entity	Entity	Οντότητες πραγματικών IoT κόμβων
OpenData_Entity	Entity	Οντότητες εικονικών IoT κόμβων από δεδομένα ανοικτών πλατφορμών
Platform	-	Πλατφόρμα
IoT_Platform	Platform	Πλατφόρμα IoT.
OpenData_Platform	Platform	Ανοικτές Πλατφόρμες IoT δεδομένων
Channel	-	Κανάλι πλατφόρμας
Location	-	Γεωχωρική θέση κόμβου
Sensor	-	Αισθητήρας κόμβου
SensorProperty	-	Μετρούμενο μέγεθος

Οι ιδιότητες των αντικειμένων (object properties) αποτελούν τις σχέσεις που επεκτείνουν το πλαίσιο ιεραρχίας κλάσεων μιας οντολογίας. Οι ιδιότητες ενός αντικειμένου συνδέουν ένα υποκείμενο (subject) και ένα αντικείμενο (object) με ένα κατηγορημα (predicate) σχηματίζοντας μια σημασιολογική τριπλέτα. Ένα σημαντικό χαρακτηριστικό τους είναι η ικανότητα να συνδέουν

στιγμιότυπα οντοτήτων που ανήκουν σε διαφορετικές κλάσεις βασιζόμενες σε σημασιολογικούς συσχετισμούς μεταξύ των αντικειμένων. Στον ακόλουθο πίνακα (Πίνακας 18) παρουσιάζονται οι ιδιότητες αντικειμένων της οντολογίας καθώς τις κλάσεις τις οποίες συνδέουν. Ιδιαίτερη αναφορά χρίζει η σχέση *hasNeighbor* η οποία προσάπτει τη γειτονική σχέση λόγω απόστασης μεταξύ δύο οντοτήτων που ανήκουν στην κλάση *Entities*. Η σχέση αυτή χρησιμοποιείται σε επίπεδο συλλογισμού στην εφαρμογή για την ανεύρεση ενός περιπατητικού μονοπατιού απαλλαγμένο από ατμοσφαιρικούς ρύπους.

Πίνακας 18: Ιδιότητες αντικειμένων οντολογίας.

Ιδιότητα	Υποκείμενο (Subject)	Αντικείμενο (Predicate)	Περιγραφή
hasChannel	Platform	Channel	Platform hasChannel Channel
hasLocation	Entity SensorProperty	Location	Entity ή SensorProperty hasLocation Location
hasPlatform	Entity	Platform	Entity hasPlatform Platform
hasSensor	Entity	Sensor	Entity hasSensor Sensor
hasSensorProperty	OpenData_Entity Sensor	SensorProperty	OpenData_Entity or Sensor hasSensorProperty SensorProperty
hasNeighbor	Entity	Entity	Entity hasNeighbor Entity

Οι ιδιότητες δεδομένων αφορούν συγκεκριμένες ιδιότητες κάθε κλάσης και περιγράφουν τα ακατέργαστα δεδομένα που λαμβάνονται από τις αντίστοιχες πηγές. Εκφράζονται με τη μορφή διαφορετικών τύπων δεδομένων όπως string, int, float κτλ. Ο πίνακας που ακολουθεί (Πίνακας 19) παρουσιάζει τις ιδιότητες δεδομένων της οντολογίας, την κλάση στην οποία ανήκουν και τον τύπο των δεδομένων που περιγράφουν.

Πίνακας 19: Πίνακας Ιδιοτήτων Δεδομένων.

Ιδιότητα	Κλάση	Τύπος Δεδομένων	Περιγραφή
channel_id	Channel	Integer	Κανάλι πλατφόρμας
Alt	Location	Double	Υψόμετρο
Lat	Location	Double	Γεωγραφικό πλάτος
Long	Location	Double	Γεωγραφικό μήκος
Unit	SensorProperty	String	Μονάδα Μέτρησης
TimeStamp	SensorProperty	dateTime	Χρονοσήμανση
Value	SensorProperty	float	Τιμή μετρούμενου μεγέθους

Name	Entity, Platform, Sensor, SensorProperty	String	Όνομα οντότητας
------	--	--------	-----------------

9.6.2 Περιγραφή λειτουργίας υπηρεσίας σημασιολογικής επισημείωσης δεδομένων

Η υπηρεσία σημασιολογικής επισημείωσης των δεδομένων που λαμβάνονται από τις υπηρεσίες του προηγούμενου επιπέδου, είναι μια εξατομικευμένη υπηρεσία με σκοπό την σημασιολογική περιγραφή των δεδομένων και την εγγραφή τους στη βάση Neo4j. Η σημασιολογική επισημείωση πραγματοποιείται σύμφωνα με την οντολογία που αποτελεί το μοντέλο περιγραφής της δομής των δεδομένων όπως αναφέρθηκε στην προηγούμενη ενότητα. Όπως οι προηγούμενες υπηρεσίες, έχει συνταχθεί σε γλώσσα προγραμματισμού Python 3.9.1. Οι κύριες λειτουργίες της είναι η λήψη των φιλτραρισμένων δεδομένων από το επίπεδο των υπηρεσιών, η μοντελοποίηση τους και η εγγραφή τους με τη μορφή κόμβων και σχέσεων στη βάση γραφοδεδομένων Neo4j. Η διασύνδεση της Python με τη βάση γραφοδεδομένων Neo4j πραγματοποιείται μέσω του επίσημου προγράμματος οδήγησης *Neo4j Python Driver* το οποίο επιτρέπει τη δημιουργία, ανάγνωση, ενημέρωση και αναζήτηση δεδομένων από τους γράφους.

Το πρόγραμμα της υπηρεσίας απαρτίζεται από δυο επιμέρους αρχεία όπου υλοποιούν τις συναρτήσεις εγκαθίδρυσης σύνδεσης με τη βάση Neo4j και τις συναρτήσεις σημασιολογικής επισημείωσης. Το αρχείο *neo4j_config.py* αποτελεί το αρχείο ρυθμίσεων για τη σύνδεση με τη βάση Neo4j και περιλαμβάνει το uri της βάσης, το συνθηματικό, και το μέγιστο συνολικό αριθμό των επιτρεπόμενων συνδέσεων, ανά κεντρικό υπολογιστή που πρέπει να διαχειρίζονται από τη δεξαμενή συνδέσεων. Το αρχείο *Neo4j_transactions.py* περιλαμβάνει τη συνάρτηση *neo4j_init()* για την εγκαθίδρυση μιας σύνδεσης με τη βάση, σύμφωνα με τα στοιχεία σύνδεσης που εμπεριέχονται στο αρχείο ρυθμίσεων. Οι μεταβλητές *driver* και *logger* είναι καθολικές ώστε να είναι προσβάσιμες από άλλα μέρη του προγράμματος. Επιπρόσθετα, περιλαμβάνονται δυο επιπλέον συναρτήσεις, η *neo4j_destroy()* χρησιμοποιείται για τον τερματισμό της σύνδεσης με τη βάση δεδομένων, ελέγχοντας πρώτα αν έχει δημιουργηθεί κάποια ενεργή σύνδεση και καταγράφεται ένα μήνυμα υπόδειξης όταν η σύνδεση τερματιστεί. Η συνάρτηση *get_driver()* χρησιμοποιείται για τον έλεγχο ύπαρξης υφιστάμενης σύνδεσης και την εγκαθίδρυση μιας νέας σε περίπτωση απουσίας της. Οι παραπάνω συναρτήσεις παρουσιάζονται στο Σχήμα 112.

Στη συνέχεια ορίζονται άλλες τρεις συναρτήσεις που μοντελοποιούν τα δεδομένα των IoT κόμβων και των ανοικτών πλατφορμών σύμφωνα με την ορισμένη οντολογία. Η συνάρτηση *add_th_iot_data(query,params)* (Σχήμα 113) μοντελοποιεί τα δεδομένα που συλλέχθηκαν από την πλατφόρμα Thingspeak και επεξεργάστηκαν από την εξατομικευμένη υπηρεσία της ενότητας *Περιγραφή λειτουργίας υπηρεσίας συλλογής IoT δεδομένων*. Τα ορίσματα της συνάρτησης είναι το ερώτημα στη γλώσσα Cypher της Neo4j, για την εγγραφή των δεδομένων στη βάση και οι παράμετροι που αποτελούν τις ιδιότητες των δεδομένων των οντοτήτων για κάθε IoT κόμβο που συνδέεται στην πλατφόρμα Thingspeak. Οι ιδιότητες των δεδομένων περιλαμβάνουν το κανάλι της πλατφόρμας, το όνομα της πλατφόρμας, τις τιμές των μετρούμενων μεγεθών των περιβαλλοντικών ρύπων, τις γεωγραφικές συντεταγμένες του σταθμού και τη χρονοσήμανση της μέτρησης.


```
27 def neo4j_init(logging):
28     global driver
29     global logger
30     logger = logging
31     driver = GraphDatabase.driver(decode(config.uri), auth=(decode(config.user), decode(config.
32     password)), max_connection_pool_size=config.max_connection_pool_size)
33     print("Connected on Neo4j Database")
34
35 def neo4j_destroy():
36     global driver
37     if driver is not None:
38         driver.close()
39         logger.info("NEO4J IS CLOSED")
40
41 def get_driver():
42     global driver
43     if driver is None:
44         neo4j_init()
45     return driver
```

Σχήμα 112: Μέθοδοι διαχείρισης συνδέσεων με τη βάση Neo4j.

```
46 def add_th_iot_data(tx, channel, platform, pm2_5, pm10, so2, no2, o3, lng, lat, timestamp):
47     tx.run("MERGE (a:IoT_Entity{label: \"IoT_Entity\", name: \"IoT_Entity_\"+$channel})-[:hasPlatform]
48     ->(b:IoT_Platform{label: \"IoT_Platform\", name: $platform, channel: $channel})"
49     "MERGE (a)-[:hasSensor]->(c:Sensor{label: \"Sensor\", name: \"PMS5003_\"+$channel})"
50     "MERGE (c)-[:hasSensorProperty]->(d:SensorProperty{label: \"SensorProperty\", name: \"pm2.
51     5\", value: $pm2_5, unit: \"ug/m3\", timestamp: $timestamp})"
52     "MERGE (c)-[:hasSensorProperty]->(e:SensorProperty{label: \"SensorProperty\", name:
53     \"pm10\", value: $pm10, unit: \"ug/m3\", timestamp: $timestamp})"
54     "MERGE (a)-[:hasSensor]->(f:Sensor{label: \"Sensor\", name: \"MQ131_\"+$channel})"
55     "MERGE (f)-[:hasSensorProperty]->(g:SensorProperty{label: \"SensorProperty\", name: \"o3\",
56     value: $o3, unit: \"ug/m3\", timestamp: $timestamp})"
57     "MERGE (a)-[:hasSensor]->(h:Sensor{label: \"Sensor\", name: \"MQ136_\"+$channel})"
58     "MERGE (h)-[:hasSensorProperty]->(i:SensorProperty{label: \"SensorProperty\", name:
59     \"no2\", value: $no2, unit: \"ug/m3\", timestamp: $timestamp})"
60     "MERGE (h)-[:hasSensorProperty]->(j:SensorProperty{label: \"SensorProperty\", name:
61     \"so2\", value: $so2, unit: \"ug/m3\", timestamp: $timestamp})"
62     "MERGE (a)-[:hasLocation]->(k:Location{label: \"Location\", long: $lng, lat: $lat})",
63     channel=channel, platform=platform, pm2_5=float(pm2_5), pm10=float(pm10), so2=float(so2),
64     no2=float(no2), o3=float(o3), lng=float(lng), lat=float(lat), timestamp=timestamp)
```

Σχήμα 113: Μέθοδος μοντελοποίησης δεδομένων από την πλατφόρμα Thingspeak.

Η μέθοδος `add_ubi_iot_data(query, params)` (Σχήμα 114) μοντελοποιεί τα δεδομένα που συλλέχθηκαν από την πλατφόρμα Ubidots και επεξεργάστηκαν από την εξατομικευμένη υπηρεσία της ενότητας *Περιγραφή λειτουργίας υπηρεσίας συλλογής IoT δεδομένων*. Τα ορίσματα της συνάρτησης, όμοια με την προηγούμενη, είναι το ερώτημα Cypher και οι ιδιότητες των δεδομένων σύμφωνα με την οντολογία. Οι ιδιότητες των δεδομένων περιλαμβάνουν την πλατφόρμα, το όνομα της συσκευής που συνδέεται στην πλατφόρμα Ubidots, το όνομα της μεταβλητής, η τιμή της και η χρονοσήμανση της μέτρησης. Η συνάρτηση αυτή πραγματοποιεί έλεγχο αν το όνομα της μεταβλητής αφορά γεωχωρικά δεδομένα ή τιμές μετρούμενων περιβαλλοντικών μεγεθών, για την εκτέλεση του αντίστοιχου κώδικα Cypher. Η ανάγκη αυτή προκύπτει την ιδιαιτερότητα της

πλατφόρμας Ubidots να αποστέλλει ξεχωριστά μηνύματα για κάθε μια από τις μεταβλητές του καναλιού.

Η τελευταία μέθοδος `add_api_data(query, params)` (Σχήμα 115) μοντελοποιεί τα δεδομένα που συλλέγονται από τις ανοικτές πλατφόρμες περιβαλλοντικών δεδομένων. Όμοια με τις προαναφερθείσες συναρτήσεις τα ορίσματα της συνάρτησης είναι το ερώτημα Cypher και οι ιδιότητες των δεδομένων. Οι ιδιότητες των δεδομένων περιλαμβάνουν το όνομα του γεωχωρικού σημείου για το οποίο ανακτώνται τα δεδομένα (Σχήμα 104), το όνομα της πλατφόρμας, τις τιμές των μετρούμενων μεγεθών των περιβαλλοντικών ρύπων, τις γεωγραφικές συντεταγμένες του σταθμού και τη χρονοσήμανση της μέτρησης.

```
59 def add_ubi_iot_data(tx, platform, device, var, value, timestamp):
60     if var=='position':
61         logger.info("UBIDOTS GEO COORDS")
62         tx.run("MERGE (a:IoT_Entity{label: \"IoT_Entity\", name: \"IoT_Entity_\"+$device})-
63             [:hasPlatform]->(b:IoT_Platform{label: \"IoT_Platform\", name: $platform, channel: $device})"
64             "MERGE (a)-[:hasLocation]->(k:Location{label: \"Location\", long: $lng, lat: $lat})",
65             platform=platform, device=device, var=var, value=value, timestamp=timestamp, lat = float
66             (value['lat']),lng = float(value['lng']))
67         logger.info("UBIDOTS GEO COORDS - DONE")
68     else:
69         logger.info("UBIDOTS SENSOR DATA")
70         sensors = {"pm2.5": "PM5003", "pm10": "PM5003", "o3": "MQ131", "so2": "MQ136", "no2": "MQ136"}
71         tx.run("MERGE (a:IoT_Entity{label: \"IoT_Entity\", name: \"IoT_Entity_\"+$device})-
72             [:hasPlatform]->(b:IoT_Platform{label: \"IoT_Platform\", name: $platform, channel: $device})"
73             "MERGE (a)-[:hasSensor]->(c:Sensor{label: \"Sensor\", name: $sensor+\"_\"+$device})"
74             "MERGE (c)-[:hasSensorProperty]->(d:SensorProperty{label: \"SensorProperty\", name: $var,
75             value: $value, unit: \"ug/m3\", timestamp: $timestamp})",
76             platform=platform, device=device, var=var, value=float(value), timestamp=timestamp,
77             sensor=sensors[var])
78         logger.info("UBIDOTS SENSOR DATA - DONE")
```

Σχήμα 114: Μέθοδος μοντελοποίησης δεδομένων από την πλατφόρμα Ubidots.

```
75 def add_api_data(tx, name, api_name, pm2_5, pm10, so2, no2, o3, long, lat, timestamp):
76     logger.info("API DATA")
77     tx.run("MERGE (a:Open_Data_Entity{label: \"Open_Data_Entity\", name: \"Open_Data_Entity_\"+$name})-
78         [:hasPlatform]->(b:Open_Data_Platform{label: \"Open_Data_Platform\", name: $api_name})"
79         "MERGE (a)-[:hasSensorProperty]->(d:SensorProperty{label: \"SensorProperty\", name: \"pm2.
80         5\", value: $pm2_5, unit: \"ug/m3\", timestamp: $timestamp})"
81         "MERGE (a)-[:hasSensorProperty]->(e:SensorProperty{label: \"SensorProperty\", name:
82         \"pm10\", value: $pm10, unit: \"ug/m3\", timestamp: $timestamp})"
83         "MERGE (a)-[:hasSensorProperty]->(g:SensorProperty{label: \"SensorProperty\", name: \"o3\",
84         value: $o3, unit: \"ug/m3\", timestamp: $timestamp})"
85         "MERGE (a)-[:hasSensorProperty]->(i:SensorProperty{label: \"SensorProperty\", name:
86         \"no2\", value: $no2, unit: \"ug/m3\", timestamp: $timestamp})"
87         "MERGE (a)-[:hasSensorProperty]->(j:SensorProperty{label: \"SensorProperty\", name:
88         \"so2\", value: $so2, unit: \"ug/m3\", timestamp: $timestamp})"
89         "MERGE (a)-[:hasLocation]->(k:Location{label: \"Location\", long: $long, lat: $lat})",
90         name=name, api_name=api_name, pm2_5=float(pm2_5), pm10=float(pm10), so2=float(so2),
91         no2=float(no2), o3=float(o3), long=float(long), lat=float(lat), timestamp=timestamp)
92     logger.info("API DATA - DONE")
```

Σχήμα 115: Μέθοδος μοντελοποίησης δεδομένων από τις ανοικτές πλατφόρμες.

Στη Neo4j, μια συναλλαγή είναι μια σειρά πράξεων που εκτελούνται μαζί ως μια ενιαία, ατομική μονάδα εργασίας. Το πρόγραμμα οδήγησης Python για το Neo4j παρέχει έναν τρόπο εργασίας με συναλλαγές χρησιμοποιώντας το αντικείμενο συνεδρίας. Μια συναλλαγή ξεκινά δημιουργώντας μια νέα συνεδρία προετοιμάζοντας την διαδικασία συναλλαγών. Μόλις ξεκινήσει μια συναλλαγή, τυχόν ερωτήματα ή λειτουργίες Cypher που εκτελούνται σε αυτήν τη συνεδρία θα αποτελούν μέρος της συναλλαγής. Είναι σημαντικό να αναφερθεί ότι όταν χρησιμοποιούνται αυτές οι λειτουργίες, δεν απαιτείται ιδιαίτερος χειρισμός για τη δέσμευση ή την επαναφορά της συναλλαγής, καθώς το αντικείμενο συνεδρίας θα εκτελέσει όλο τον χειρισμό.

Η `session.write_transaction()` είναι μια συνάρτηση που παρέχεται από το πρόγραμμα οδήγησης Neo4j Python που απλοποιεί τη διαδικασία εκτέλεσης μιας δήλωσης Cypher μέσα σε μια συναλλαγή εγγραφής στη βάση. Αυτή η συνάρτηση λαμβάνει δύο ορίσματα, μια συνάρτηση επανάκλησης και τυχόν πρόσθετα ορίσματα που πρέπει να περάσουν σε αυτή. Η συνάρτηση επανάκλησης εκτελείται στο πλαίσιο μιας συναλλαγής εγγραφής και μεταβιβάζεται ένα στιγμιότυπο της συναλλαγής ως πρώτο όρισμα, επιτρέποντας την εκτέλεση δηλώσεων Cypher και άλλων λειτουργιών της βάσης δεδομένων εντός της συναλλαγής. Η συνάρτηση `session.write_transaction()` χειρίζεται την έναρξη της συναλλαγής, τη δέσμευση των αλλαγών και τον χειρισμό τυχόν σφαλμάτων που ενδέχεται να προκύψουν. Μπορεί να χρησιμοποιηθεί για την εκτέλεση μιας μεμονωμένης δήλωσης ή ενός μικρού αριθμού Cypher σε μια συναλλαγή παρέχοντας ευκολία στη διαχείριση των σφαλμάτων που ενδέχεται να προκύψουν.

Λαμβάνοντας υπόψιν τον τρόπο λειτουργίας της παραπάνω συνάρτησης του προγράμματος οδήγησης, δημιουργήθηκαν δυο επιπρόσθετες συναρτήσεις για τον χειρισμό των συναλλαγών εγγραφής τόσο των IoT δεδομένων όσο και αυτών που προέρχονται από τις ανοικτές πλατφόρμες. Η συνάρτηση `write_msg()` (Σχήμα 116) είναι υπεύθυνη για την εκτέλεση της εγγραφής των IoT δεδομένων και λαμβάνει ως όρισμα το επεξεργασμένο μήνυμα από τις IoT πλατφόρμες. Το μήνυμα εξετάζεται ως προς το περιεχόμενό του και αναγνωρίζεται η πλατφόρμα από την οποία προήλθε η πρωτογενής πληροφορία. Στη συνέχεια εγκαθιδρύεται μια συνεδρία και εκτελείται η συναλλαγή εγγραφής σύμφωνα με τις συναρτήσεις `add_th_iot_data()` και `add_ubi_iot_data()` αντίστοιχα.

```
88 def write_msg(msg):
89     logger.info("creating session")
90     try:
91         if msg['platform'] == 'Thingspeak':
92             with get_driver().session() as session:
93                 session.write_transaction(add_th_iot_data, msg['channel'], msg['platform'], msg['pm2.
94                     5'],
95                     msg['pm10'], msg['so2'], msg['no2'], msg['o3'], msg['lng'], msg['lat'], msg
96                     ['timestamp'])
97         elif msg['platform'] == 'Ubidots':
98             with get_driver().session() as session:
99                 session.write_transaction(add_ubi_iot_data, msg['platform'], msg['device'], msg
100                     ['var'], msg['value'], msg['timestamp'])
101     except RuntimeError as err:
102         logger.info(err)
```

Σχήμα 116: Μέθοδος εκτέλεσης συναλλαγής εγγραφής IoT δεδομένων.

Ομοίως, η συνάρτηση `write_api_data()` (Σχήμα 117) είναι υπεύθυνη για την εκτέλεση της εγγραφής των δεδομένων από τις ανοικτές πλατφόρμες και λαμβάνει ως όρισμα το επεξεργασμένο μήνυμα από τις απαντήσεις των αιτημάτων ανάκτησης δεδομένων από τις IoT πλατφόρμες. Στη συνέχεια εγκαθιδρύεται μια συνεδρία και εκτελείται η συναλλαγή εγγραφής σύμφωνα με τη συνάρτηση `add_api_data()`.

Οι κλήσεις των συναρτήσεων αυτών πραγματοποιούνται απευθείας από τις υπηρεσίες συλλογής δεδομένων της προηγούμενης ενότητας και προσαρμόζονται στον κώδικα των κυρίων προγραμμάτων τους. Και στις δυο περιπτώσεις, μετά την ολοκλήρωση της διαμόρφωσης της ωφέλιμης πληροφορίας εγκαθιδρύεται μια συνεδρία και εκτελούνται οι συναλλαγές εγγραφής των δεδομένων.

```
101 def write_api_data(resp):
102     logger.info("creating session")
103     try:
104         with get_driver().session() as session:
105             session.write_transaction(add_api_data, resp['name'], resp['api_name'], resp['pm2.5'], resp
106                                     ['pm10'], resp['so2'], resp['no2'], resp['o3'], resp['long'], resp['lat'], resp
107                                     ['timestamp'])
106     except RuntimeError as err:
107         logger.info(err)
```

Σχήμα 117: Μέθοδος εκτέλεσης συναλλαγής εγγραφής δεδομένων από ανοικτές πλατφόρμες.

9.7 Επίπεδο Εφαρμογής

9.7.1 Περιγραφή εφαρμογής «Πράσινης Διαδρομής»

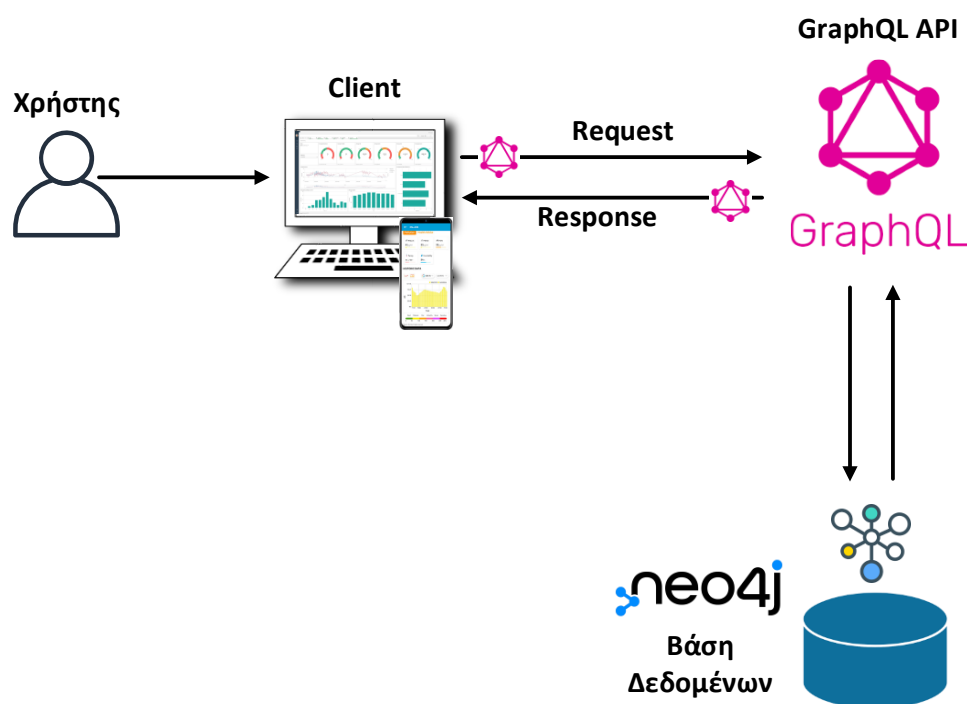
Το επίπεδο εφαρμογής στην αρχιτεκτονική του προτεινόμενου πλαισίου αναφέρεται στο λογισμικό και τις υπηρεσίες που εκτελούνται πάνω από το υποκείμενο υλικό και την υποδομή επικοινωνίας, για να παρέχουν συγκεκριμένες λειτουργίες και υπηρεσίες στους χρήστες. Είναι υπεύθυνο για την κατανόηση των δεδομένων που συλλέχθηκαν από τις επιμέρους πηγές για την παροχή μιας χρήσιμης διεπαφής για τους χρήστες και την αλληλεπίδραση με το σύστημα. Διαδραματίζει κρίσιμο ρόλο στη συνολική αρχιτεκτονική του συστήματος, καθώς παρέχει την απαραίτητη λειτουργικότητα μετατρέποντας το σύστημα σε χρήσιμο και πολύτιμο εργαλείο για τους χρήστες.

Στα πλαίσια της παρούσας εργασίας το επίπεδο εφαρμογής έχει ως βασικό σκοπό τη συγκέντρωση σημασιολογικών δεδομένων για τον προσδιορισμό του δείκτη ποιότητας αέρα σε μια συγκεκριμένη γεωγραφική περιοχή ενδιαφέροντος για το χρήστη. Απώτερος στόχος είναι ο προσδιορισμός μιας διαδρομής στο χάρτη που θα προταθεί στον χρήστη, η οποία κατά μήκος της διαθέτει τους λιγότερους ατμοσφαιρικούς ρύπους, λαμβάνοντας υπόψιν τους υπολογισμένους δείκτες ποιότητας αέρα ανά περιοχή.

Αναλυτικότερα, έστω ένας χρήστης της εφαρμογής «Πράσινη Διαδρομή» επιθυμεί να κάνει περίπατο από το σπίτι του στο πλησιέστερο πάρκο που απέχει δυο χιλιόμετρα με τα πόδια. Μέσω της εφαρμογής αναζητά μια διαδρομή που θα ακολουθήσει, βάσει των προσφάτων περιβαλλοντικών δεδομένων ποιότητας αέρα, η οποία είναι απαλλαγμένη από ατμοσφαιρική

μόλυνση καθώς αντιμετωπίζει χρόνια αναπνευστικά προβλήματα. Αρχικά, ο χρήστης εισάγει τη γεωγραφική θέση της αφετηρίας και του προορισμού του στην εφαρμογή και στη συνέχεια υπολογίζεται και απεικονίζεται σε χάρτη, η προτεινόμενη διαδρομή. Για τον υπολογισμό της διαδρομής υπολογίζεται ο δείκτης ποιότητας αέρα όλων των σταθμών που βρίσκονται στην ευρύτερη γεωγραφική περιοχή των σημείων ενδιαφέροντος και επιλέγονται αυτοί που ο δείκτης τους βρίσκεται σε φυσιολογικά επίπεδα (Good, Fair, Moderate, Poor). Στη συνέχεια συνάπτονται σημασιολογικές σχέσεις γειτονίας *hasNeighbor* βάσει της χιλιομετρικής απόστασης μεταξύ των κόμβων, προσδιορίζοντας τα σημεία από τα οποία θα περάσει ο χρήστης έως ότου φτάσει στον προορισμό του.

Η εφαρμογή που αναπτύχθηκε είναι πλήρους στοίβας (Full Stack) και περιλαμβάνει υλοποίηση με αρχιτεκτονική πολλαπλών επιπέδων. Η αρχιτεκτονική είναι ευρέως γνωστή ως εφαρμογή τριών επιπέδων (Σχήμα 118), η οποία απαρτίζεται από μια διαδικτυακή εφαρμογή (frontend) με την οποία αλληλεπιδρά ο χρήστης, το επίπεδο API (GraphQL) και μια βάση δεδομένων (backend). Η εφαρμογή πλήρους στοίβας με GraphQL αναφέρεται σε μια εφαρμογή που χρησιμοποιεί το επίπεδο διεπαφής τόσο στον πελάτη όσο και στον διακομιστή. Το frontend της εφαρμογής, είναι υπεύθυνο για το χειρισμό της διεπαφής χρήστη και την υποβολή ερωτημάτων GraphQL στην πλευρά του διακομιστή. Το backend, είναι υπεύθυνο για το χειρισμό των ερωτημάτων GraphQL, την εκτέλεση οποιωνδήποτε απαραίτητων λειτουργιών της βάσης δεδομένων και την επιστροφή των ζητούμενων δεδομένων στον πελάτη.



Σχήμα 118: Αρχιτεκτονική εφαρμογής πλήρους στοίβας (Full Stack).

Σε μια εφαρμογή πλήρους στοίβας με GraphQL, ο πελάτης και ο διακομιστής επικοινωνούν χρησιμοποιώντας τη διεπαφή GraphQL, η οποία επιτρέπει έναν ευέλικτο και αποτελεσματικό τρόπο ανάκτησης και τροποποίησης δεδομένων. Ο πελάτης μπορεί να καθορίσει ακριβώς ποια δεδομένα

χρειάζεται και ο διακομιστής θα επιστρέψει μόνο αυτά, μειώνοντας τον όγκο των δεδομένων που πρέπει να σταλούν μέσω του δικτύου. Αυτό μπορεί να βελτιώσει την απόδοση και την επεκτασιμότητα της εφαρμογής. Επιπλέον, η χρήση του GraphQL τόσο στον πελάτη όσο και στον διακομιστή επιτρέπει έναν συνεπή και ενοποιημένο τρόπο αλληλεπίδρασης με δεδομένα, ανεξάρτητα από το πού προέρχονται.

9.7.2 GraphQL

Οι περισσότερες εφαρμογές σήμερα έχουν την ανάγκη ανάκτησης δεδομένων από έναν διακομιστή όπου αυτά αποθηκεύονται σε μία βάση δεδομένων. Το API έχει την ευθύνη να παρέχει μια διεπαφή στα αποθηκευμένα δεδομένα που ταιριάζει στις ανάγκες της εφαρμογής. Το GraphQL είναι μια προδιαγραφή για τη δημιουργία API που παρέχει μια αποτελεσματική και ευέλικτη λύση έναντι του REST. Αναπτύχθηκε σαν κλειστή εφαρμογή το 2012 από το Facebook, ενώ διανεμήθηκε σαν ανοιχτό λογισμικό το 2015 [56]. Η προδιαγραφή GraphQL περιγράφει μια γλώσσα ερωτημάτων API και έναν τρόπο εκπλήρωσης αυτών των αιτημάτων, που έχει σχεδιαστεί για να αναπαραστήσει περίπλοκες εμφωλευμένες εξαρτήσεις δεδομένων σε σύγχρονες εφαρμογές [57].

Κατά τη δημιουργία ενός GraphQL API, περιγράφονται τα διαθέσιμα δεδομένα χρησιμοποιώντας ένα σύστημα αυστηρού τύπου. Αυτοί οι ορισμοί τύπων γίνονται οι προδιαγραφές για το API και η εφαρμογή πελάτη είναι ελεύθερη να ζητήσει τα δεδομένα που απαιτεί με βάση αυτούς τους ορισμούς τύπων, οι οποίοι ορίζουν επίσης τα σημεία εισόδου για το API. Έναντι πολλών τελικών σημείων που επιστρέφουν σταθερές δομές δεδομένων, ένας διακομιστής GraphQL παρέχει μόνο ένα τελικό σημείο και ανταποκρίνεται ακριβώς με τα δεδομένα που ζήτησε ο πελάτης.

Με το GraphQL επιτρέπεται στον χρήστη να έχει λεπτομερείς πληροφορίες σχετικά με τα δεδομένα που ζητούνται στο backend. Επίσης, μπορεί να πραγματοποιηθεί παρακολούθηση επιδόσεων χαμηλού επιπέδου των αιτημάτων, που επιβάλλονται σε επεξεργασία από τον διακομιστή. Το GraphQL χρησιμοποιεί την έννοια των λειτουργιών επίλυσης για την συλλογή των δεδομένων που ζητούνται από έναν πελάτη. Η οργάνωση και η μέτρηση της απόδοσης αυτών των λύσεων παρέχει σημαντικές πληροφορίες σχετικά με τα σημεία συμφόρησης στο σύστημα.

Συνοπτικά, το GraphQL προσφέρει πολλά πλεονεκτήματα σε σχέση με τα RESTful API, όπως ευελιξία, βελτιωμένη εμπειρία προγραμματιστή, καλύτερη τεκμηρίωση και δυνατότητες σε πραγματικό χρόνο. Το GraphQL μπορεί επίσης να βοηθήσει στην ενοποίηση δεδομένων από διαφορετικά συστήματα. Προσφέρεται η δυνατότητα να δημιουργηθούν GraphQL APIs που ενσωματώνουν δεδομένα από πολλές υπηρεσίες μαζί και παρέχουν έναν σαφή τρόπο για την ενοποίηση δεδομένων από αυτά τα διαφορετικά συστήματα σε ένα ενιαίο σχήμα GraphQL. Το GraphQL μπορεί να χρησιμοποιηθεί για την τμηματοποίηση της ανάκτησης δεδομένων στην εφαρμογή σε ένα μοτίβο αλληλεπίδρασης δεδομένων βάσει στοιχείων. Δεδομένου ότι κάθε ερώτημα GraphQL μπορεί να περιγράψει ακριβώς τη διέλευση γραφήματος και τα πεδία που πρέπει να επιστραφούν, η ενθυλάκωση αυτών των ερωτημάτων με στοιχεία εφαρμογής μπορεί να βοηθήσει στην απλοποίηση της ανάπτυξης και της δοκιμής εφαρμογών. Τέλος, η ενδοσκόπηση είναι ένα ισχυρό χαρακτηριστικό του GraphQL που μας επιτρέπει να ρωτάμε ένα GraphQL API για τους τύπους και τα ερωτήματα που υποστηρίζει. Η ενδοσκόπηση γίνεται ένας τρόπος αυτο-τεκμηρίωσης του API. Τα εργαλεία που κάνουν χρήση της ενδοσκόπησης μπορούν να παρέχουν

τεκμηρίωση API αναγνώσιμη από τον άνθρωπο, καθώς και εργαλεία οπτικοποίησης και να αξιοποιήσουν την παραγωγή κώδικα για τη δημιουργία πελατών API [58].

Ωστόσο, το GraphQL έχει ορισμένα μειονεκτήματα όπως η προσωρινή αποθήκευση, η ασφάλεια και η πολυπλοκότητα. Αναλυτικότερα, μία από τις πιο αξιοσημείωτες προκλήσεις της υιοθέτησης του GraphQL είναι ότι ορισμένες πρακτικές από το REST δεν ισχύουν κατά τη χρήση του GraphQL. Για παράδειγμα, οι κωδικοί κατάστασης HTTP χρησιμοποιούνται συνήθως για να υποδηλώσουν επιτυχία, αποτυχία ή άλλες καταστάσεις για ένα αίτημα REST. Στο GraphQL, κάθε αίτημα επιστρέφει «200 OK», ανεξάρτητα από το αν ήταν πλήρως επιτυχές, αυξάνοντας την πολυπλοκότητα στον χειρισμό σφαλμάτων. Αντί για έναν κωδικό κατάστασης που περιγράφει το αποτέλεσμα του αιτήματός, τα σφάλματα GraphQL επιστρέφονται συνήθως σε επίπεδο πεδίου. Αυτό σημαίνει ότι ένα μέρος του ερωτήματός GraphQL μπορεί να ανακτηθεί με επιτυχία, ενώ άλλα πεδία να επιστρέψουν σφάλματα και θα πρέπει να αντιμετωπιστούν κατάλληλα.

Η προσωρινή αποθήκευση είναι άλλος ένας τομέας που αντιμετωπίζεται διαφορετικά στο GraphQL σε σχέση με το REST. Στο GraphQL κάθε αίτημα μπορεί να περιέχει ένα διαφορετικό σύνολο επιλογών και γι' αυτό δεν μπορεί να επιστραφεί ένα αποθηκευμένο αποτέλεσμα για ολόκληρο το αίτημα. Το μειονέκτημα αυτό μετριάζεται από τους περισσότερους πελάτες GraphQL που κάνουν χρήση κρυφής μνήμης σε επίπεδο εφαρμογής αν και τις περισσότερες φορές, τα αιτήματά GraphQL βρίσκονται σε περιβάλλον ελέγχου ταυτότητας, όπου η προσωρινή αποθήκευση δεν εφαρμόζεται.

Τέλος, το πρόβλημα ερωτήματος $n + 1$ είναι ένα κοινό πρόβλημα στις υλοποιήσεις ανάκτησης δεδομένων με GraphQL που μπορεί να οδηγήσει σε πολλαπλές διαδρομές μετ' επιστροφής στο επίπεδο δεδομένων και μπορεί να επηρεάσει αρνητικά την απόδοση. Το πρόβλημα εμφανίζεται όταν ένας πελάτης ζητά μια λίστα στοιχείων, όπου κάθε στοιχείο έχει μια σχέση με έναν άλλο τύπο στοιχείου και ο διακομιστής εκτελεί ένα ξεχωριστό ερώτημα βάσης δεδομένων για κάθε στοιχείο για να ανακτήσει τα σχετικά δεδομένα. Από τα παραπάνω προκύπτει ότι καθώς αυξάνεται ο αριθμός των στοιχείων, αυξάνεται και ο αριθμός των ερωτημάτων, γεγονός που μπορεί να μειώσει σημαντικά την απόδοση. Για την αντιμετώπιση του προβλήματος $n + 1$ γίνεται χρήση ενσωματωμένων εφαρμογών GraphQL στις βάσεις δεδομένων, όπως είναι η βιβλιοθήκη Neo4j GraphQL και το PostGraphile. Οι ενσωματώσεις αυτές επιτρέπουν τη δημιουργία ενός ερωτήματος βάσης δεδομένων από ένα αυθαίρετο αίτημα GraphQL, διασφαλίζοντας μόνο μία διαδρομή μετ' επιστροφής στη βάση δεδομένων.

9.7.3 GraphQL Schema

Το GraphQL χρησιμοποιεί ένα σύστημα ισχυρού τύπου για να καθορίσει τις δυνατότητες ενός API. Όλοι οι τύποι που εκθέτονται σε ένα API καταγράφονται στο Schema χρησιμοποιώντας την γλώσσα ορισμού σχήματος GraphQL (SDL). Το Schema GraphQL χρησιμοποιείται για τον καθορισμό του τρόπου με τον οποίο ένας πελάτης μπορεί να έχει πρόσβαση στα δεδομένα και καθορίζει τους τύπους και τα πεδία στα οποία μπορούν να υποβληθούν ερωτήματα ή να τροποποιηθούν. Ο καθορισμός του σχήματος δίνει την δυνατότητα στις ομάδες που υλοποιούν το backend και το frontend να εργάζονται χωρίς περαιτέρω επικοινωνία, καθώς όλοι γνωρίζουν την καθορισμένη δομή των δεδομένων που αποστέλλονται μέσω διαδικτύου.

Το Schema σχεδιάζεται με τέτοιο τρόπο, ώστε να ταιριάζει όσο το δυνατόν περισσότερο με τη δομή των δεδομένων στη βάση Neo4j (Σχήμα 119). Οι οντότητες που προσδιορίζονται στη βάση γραφοδεδομένων, στο GraphQL γίνονται τύποι, οι ιδιότητες των οντοτήτων γίνονται πεδία στους τύπους και οι συνδέσεις ή οι σχέσεις που συνδέουν τους τύπους ορίζονται ως πεδία που αναφέρονται σε άλλους τύπους. Κάθε τύπος περιέχει πεδία, τα οποία μπορεί να είναι βαθμωτοί τύποι, αντικείμενα ή ακόμη και λίστες.

Στο GraphQL υπάρχουν τρεις τρόποι για ανάκτηση, εισαγωγή, ενημέρωση ή διαγραφή δεδομένων.

- Queries: Ανάκτηση δεδομένων προς ανάγνωση.
- Mutations: Δημιουργία νέων δεδομένων, ενημέρωση υφιστάμενων δεδομένων, διαγραφή υφιστάμενων δεδομένων.
- Subscriptions: Συνδέσεις πραγματικού χρόνου με τον διακομιστή με σκοπό την άμεση ενημέρωση για σημαντικά γεγονότα.

```
161 input user_point {
162   latitude: Float!
163   longitude: Float!
164 }
165
166 type IoT_Platform {
167   channel: String!
168   iotEntitieshasPlatform: [IoT_Entity!]! @relationship(type: "hasPlatform", direction: IN)
169   label: String!
170   name: String!
171 }
172
173 type Location {
174   iotEntitieshasLocation: [IoT_Entity!]! @relationship(type: "hasLocation", direction: IN)
175   label: String!
176   lat: Float!
177   long: Float!
178   openDataEntitieshasLocation: [OpenData_Entity!]! @relationship(type: "hasLocation", direction: IN)
179 }
180
181 type OpenData_Entity implements Entity{
182   coLocationIoTEntities: [IoT_Entity!]! @relationship(type: "hasNeighbor", direction: OUT)
183   coLocationOpenDataEntities: [OpenData_Entity!]! @relationship(type: "hasNeighbor", direction: OUT)
184   hasLocationLocations: [Location!]! @relationship(type: "hasLocation", direction: OUT)
185   hasPlatformOpenDataPlatforms: [OpenData_Platform!]! @relationship(type: "hasPlatform", direction: OUT)
186   iotEntitiescoLocation: [IoT_Entity!]! @relationship(type: "hasNeighbor", direction: IN)
187   label: String!
188   name: String!
189   openDataEntitiescoLocation: [OpenData_Entity!]! @relationship(type: "hasNeighbor", direction: IN)
190 }
```

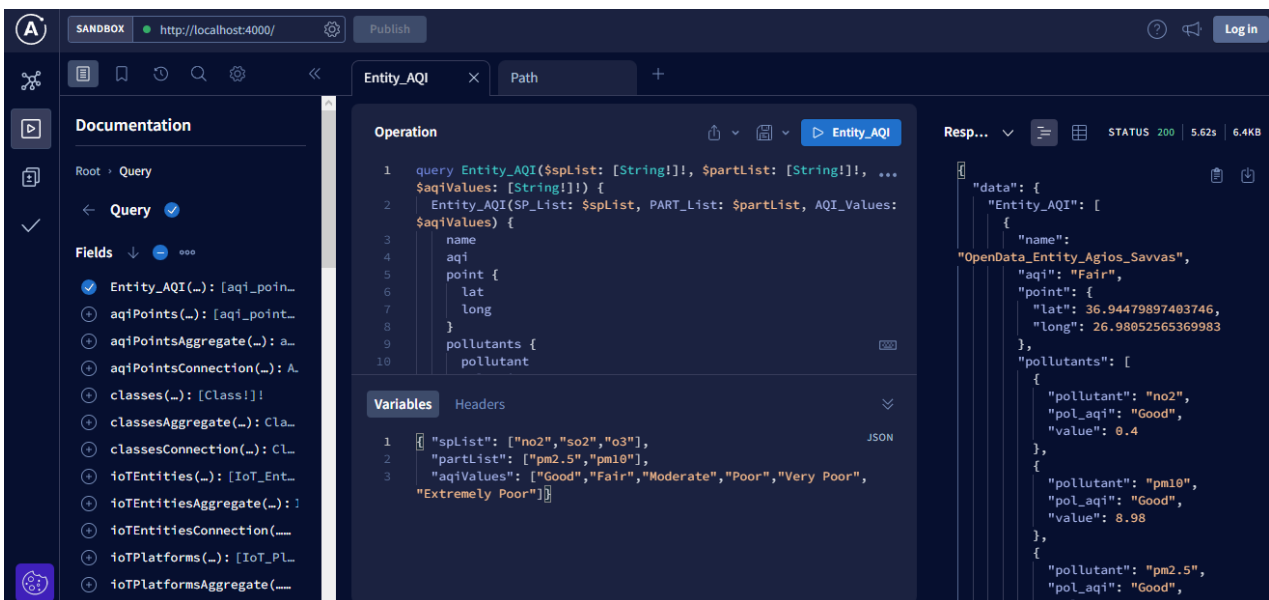
Σχήμα 119: Απόσπασμα από το GraphQL Schema της εφαρμογής.

Επιπρόσθετα, στο Schema ορίζονται και οι συναρτήσεις επίλυσης (Resolvers) που χειρίζονται την εκτέλεση ενός ερωτήματος ή μιας τροποποίησης δεδομένων GraphQL στην πλευρά του διακομιστή. Κάθε πεδίο σε ένα ερώτημα GraphQL ή μια ενημέρωση αντιστοιχίζεται σε μια συνάρτηση επίλυσης, η οποία είναι υπεύθυνη για την ανάκτηση των δεδομένων από τη βάση Neo4j και την επιστροφή τους στον πελάτη. Η βιβλιοθήκη Neo4j GraphQL προσφέρει τη δυνατότητα της προσθήκης προσαρμοσμένης λογικής στο API της εφαρμογής, χρησιμοποιώντας συναρτήσεις οι

οποίες περιλαμβάνουν ερωτήματα που έχουν συνταχθεί σε γλώσσα Cypher. Η ενσωμάτωση του Cypher ερωτήματος στο Schema γίνεται με τη δήλωση `@cypher`, για την αντιστοίχιση των αποτελεσμάτων του ερωτήματος με το δηλωμένο πεδίο GraphQL. Για τη χρήση των δηλώσεων `@cypher` απαιτείται η εγκατάσταση της πρόσθετης βιβλιοθήκης APOC. Στα πλαίσια της εργασίας, υιοθετήθηκε η δυνατότητα συμπερίληψης κώδικα Cypher, συντάσσοντας ένα ερώτημα για τον υπολογισμό του δείκτη ποιότητας αέρα των σταθμών και ένα ερώτημα ενημέρωσης δεδομένων για τον υπολογισμό της «Πράσινης Διαδρομής».

9.7.4 Apollo Server - Client

Το Apollo είναι μια συλλογή εργαλείων που διευκολύνουν τη χρήση του GraphQL, στον διακομιστή, στην εφαρμογή πελάτη ή ακόμη και στο νέφος. Στο πλαίσιο υλοποίησης της εφαρμογής, χρησιμοποιήθηκε ο διακομιστής Apollo για τη δημιουργία του GraphQL API και το Apollo Client, μια βιβλιοθήκη JavaScript από την πλευρά του πελάτη, για την αναζήτηση του GraphQL API από την εφαρμογή. Επίσης, χρησιμοποιήθηκε ο Explorer του Apollo Studio το οποίο αποτελεί εργαλείο για τη δημιουργία και εκτέλεση ερωτημάτων GraphQL (Σχήμα 120).



Σχήμα 120: Στιγμιότυπο του Apollo Studio Explorer.

Ο διακομιστής Apollo είναι ένας δημοφιλής διακομιστής GraphQL ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί για τη δημιουργία backends GraphQL. Παρέχει ένα σύνολο εργαλείων και βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για τη δημιουργία και εκτέλεση ενός διακομιστή GraphQL σε περιβάλλον Node.js. Αναλυτικότερα, είναι υπεύθυνος για τον ορισμό του Schema, τον χειρισμό ερωτημάτων και τροποποιήσεων δεδομένων με συναρτήσεις επίλυσης, σύνδεση με ποικίλες πηγές δεδομένων, τη διαχείριση απόδοσης ερωτημάτων με προσωρινή αποθήκευση και ομαδοποίηση, την ασφάλεια (έλεγχος ταυτότητας, εξουσιοδότηση και επικύρωση του ερωτήματος του πελάτη) και τις ενημερώσεις των δεδομένων σε πραγματικό χρόνο. Μπορεί να ενσωματωθεί με άλλες βιβλιοθήκες και πλαίσια Node.js για τη δημιουργία εφαρμογών πλήρους στοίβας.

Για την εγκατάσταση και εκτέλεση του Apollo Server [59] σε τοπικό υπολογιστή για την εκτέλεση των πειραμάτων, προηγήθηκε η εγκατάσταση του Node.js, δημιουργήθηκε ένας νέος κατάλογος έργου και αρχικοποιήθηκε ως έργο Node.js, εγκαταστάθηκε η βιβλιοθήκη του Apollo Server και δημιουργήθηκε ένα νέο αρχείο server.js για τον διακομιστή. Στη συνέχεια, στο αρχείο server.js εισήχθη το αντικείμενο του Apollo Server, ορίστηκε το Schema GraphQL και οι συναρτήσεις επίλυσης. Τέλος, δημιουργήθηκε ένα νέο στιγμιότυπο του Apollo Server και εκκινήθηκε ο διακομιστής. Με την ολοκλήρωση της παραπάνω διαδικασίας, το Apollo Studio Explorer του διακομιστή γίνεται προσβάσιμο από ένα πρόγραμμα περιήγησης, μεταβαίνοντας στη διεύθυνση <http://localhost:4000>.

Το Apollo Client είναι μια βιβλιοθήκη ανοιχτού κώδικα για την διασύνδεση εφαρμογών Javascript με GraphQL API. Παρέχει ένα σύνολο εργαλείων και βιβλιοθηκών που μπορούν να χρησιμοποιηθούν για την αλληλεπίδραση με ένα σύστημα υποστήριξης GraphQL και τη διαχείριση δεδομένων στην πλευρά του πελάτη. Διαθέτει ενσωματώσεις με πολλά πλαίσια frontend, συμπεριλαμβανομένων των React και Vue.js, καθώς και εγγενών εκδόσεων για κινητά για iOS και Android. Για την υλοποίηση της εφαρμογής της εργασίας, χρησιμοποιήθηκε η ενσωμάτωση του Vue Apollo Client [60] για την ανάκτηση δεδομένων μέσω GraphQL στα δομικά στοιχεία Vue. Το Apollo Client χειρίζεται την προσωρινή αποθήκευση δεδομένων πελάτη και μπορεί επίσης να χρησιμοποιηθεί για τη διαχείριση τοπικών δεδομένων κατάστασης.

9.7.5 *Queries-Mutations εφαρμογής*

Κατά την είσοδο του χρήστη στην ιστοσελίδα της εφαρμογής, υποβάλλεται ένα GraphQL ερώτημα στη βάση για ανάκτηση των περιβαλλοντικών δεδομένων των σταθμών παρατήρησης που καλύπτουν τη γεωγραφική περιοχή που απεικονίζεται στο χάρτη εκείνη τη χρονική στιγμή. Το ερώτημα ανάκτησης δεδομένων αιτείται από τη βάση όλες τις τιμές των μετρούμενων μεγεθών (SO₂, NO₂, O₃, PM₁₀, PM_{2.5}) και τον δείκτη ποιότητας αέρα (AQI) των σταθμών της περιοχής, καθώς και τα γεωχωρικά τους δεδομένα για την απεικόνισή τους στο χάρτη.

Σύμφωνα με την Ευρωπαϊκό Οργανισμό Ατμοσφαιρικής Ρύπανσης, όσο υψηλότερη είναι η τιμή του AQI, τόσο μεγαλύτερο είναι το επίπεδο της ατμοσφαιρικής ρύπανσης. Για το λόγο αυτό ο δείκτης ποιότητας αέρα υπολογίζεται λαμβάνοντας υπόψη το μέγιστο όλων των επιμέρους δεικτών AQI των περιβαλλοντικών ρύπων (Σχέση 8).

$$AQI = \max(AQI_{SO_2}, AQI_{NO_2}, AQI_{O_3}, AQI_{PM_{10}}, AQI_{PM_{2.5}}) \quad (8)$$

Ο υπολογισμός του ευρωπαϊκού δείκτη ποιότητας αέρα δεν προκύπτει άμεσα από την εξέταση των στιγμιότυπων του γραφήματος και η σύνταξη της γλώσσας Cypher δεν επιτρέπει τη χρήση άμεσων συνδυαστικών υπολογισμών των δεδομένων. Για το λόγο αυτό απαιτήθηκε η συγγραφή βοηθητικών συναρτήσεων εντός της βάσης Neo4j, κάνοντας χρήση της πρόσθετης βιβλιοθήκης APOC. Οι συναρτήσεις αυτές λαμβάνουν ως όρισμα την μετρούμενη τιμή της συγκέντρωσης του ρύπου και επιστρέφουν έναν ακέραιο δείκτη με εύρος τιμών 0-5 (Σχήμα 121). Η αριθμητική τιμή του δείκτη αντιστοιχεί στην λεκτική περιγραφή του επιπέδου μόλυνσης του αέρα, σύμφωνα με την ακόλουθη λίστα (0:Good, 1:Fair, 2:Moderate, 3:Poor, 4: Very poor, 5: Extremely poor) (Σχήμα 1).

Συνολικά συντάχθηκαν πέντε συναρτήσεις αντιστοίχισης, όσοι είναι και οι ρύποι που παρακολουθούνται.

```
63 CALL apoc.custom.declareFunction(  
64   'aqi_pm10(value::NUMBER) :: INT',  
65   '  
66   RETURN CASE  
67     WHEN ($value >= 0 AND $value < 20) THEN 0  
68     WHEN ($value >= 20 AND $value < 40) THEN 1  
69     WHEN ($value >= 40 AND $value < 50) THEN 2  
70     WHEN ($value >= 50 AND $value < 100) THEN 3  
71     WHEN ($value >= 100 AND $value < 150) THEN 4  
72     WHEN ($value >= 150 AND $value < 1200) THEN 5  
73     ELSE -1  
74   END  
75   '  
76 );  
77  
78 CALL apoc.custom.declareFunction(  
79   'aqi_so2(value::NUMBER) :: INT',  
80   '  
81   RETURN CASE  
82     WHEN ($value >= 0 AND $value < 100) THEN 0  
83     WHEN ($value >= 100 AND $value < 200) THEN 1  
84     WHEN ($value >= 200 AND $value < 350) THEN 2  
85     WHEN ($value >= 350 AND $value < 500) THEN 3  
86     WHEN ($value >= 500 AND $value < 750) THEN 4  
87     WHEN ($value >= 750 AND $value < 1250) THEN 5  
88     ELSE -1  
89   END  
90   '  
91 );
```

Σχήμα 121: Συναρτήσεις αντιστοίχισης της μετρούμενης συγκέντρωσης ρύπων με δείκτη τιμών 0-5.

Οι κλίσεις όλων των προαναφερθέντων συναρτήσεων ομαδοποιήθηκαν και συγχωνεύτηκαν σε μια νέα (Σχήμα 122), που λαμβάνει ως όρισμα το όνομα του περιβαλλοντικού ρύπου και επιστρέφει την ακέραια τιμή του δείκτη αντιστοίχισης (0-5). Με αυτό τον τρόπο επιτεύχθηκε η ενσωμάτωση των υπολογισμών του AQI για κάθε ρύπο, εντός του ερωτήματος Cypher.

```
123 CALL apoc.custom.declareFunction(  
124   'AQI(pollutant::STRING, value::NUMBER) :: STRING',  
125   '  
126   RETURN CASE  
127     WHEN $pollutant = "pm2.5" THEN custom.aqi_pm2_5($value)  
128     WHEN $pollutant = "pm10" THEN custom.aqi_pm10($value)  
129     WHEN $pollutant = "so2" THEN custom.aqi_so2($value)  
130     WHEN $pollutant = "no2" THEN custom.aqi_no2($value)  
131     WHEN $pollutant = "o3" THEN custom.aqi_o3($value)  
132   END  
133   '  
134 );
```

Σχήμα 122: Συνάρτηση υπολογισμού του AQI για κάθε περιβαλλοντικό ρύπο.

Αξιοποιώντας τη δυνατότητα που προσφέρει η βιβλιοθήκη Neo4j GraphQL για την προσθήκη προσαρμοσμένης λογικής στο API της εφαρμογής, χρησιμοποιήθηκε ένα ερώτημα GraphQL τύπου *Query* το οποίο περιλαμβάνει εμφωλευμένο, τον κώδικα του ερωτήματος σε γλώσσα Cypher (Σχήμα 123). Η ενσωμάτωση του Cypher ερωτήματος στο Schema πραγματοποιήθηκε με τη δήλωση `@cypher`, για την αντιστοίχιση των αποτελεσμάτων του ερωτήματος με το δηλωμένο πεδίο GraphQL. Το ερώτημα λαμβάνει σαν παραμέτρους τρεις λίστες (Σχήμα 124) με τα ονόματα των αέριων ρύπων, των αιωρούμενων σωματιδίων και των λεκτικών δεικτών ποιότητας αέρα επιστρέφοντας σε λίστα τα περιβαλλοντικά και γεωχωρικά δεδομένα των σταθμών.

```
41 type Query{
42   Entity_AQI(SP_List: [String!]!, PART_List: [String!]!, AQI_Values: [String!]!): [aqi_points]
43   @cypher(
44     statement: """
45     call n10s.inference.nodesLabelled('Entity', {catNameProp: 'Label', catLabel: 'Resource',
46     subCatRel: 'SCO' }) YIELD node
47     match (node)-[:hasLocation]-(l:Location)
48     with node,avg(l.Lat) as lat, avg(l.Long) as long
49     with node, lat,long,localdatetime({timezone: 'Europe/Athens'}) as now
50     match (node)-[:hasSensor*0..1]-()-[:hasSensorProperty]->(b:SensorProperty)
51     where (((b.name in $SP_List) AND (LocalDateTime(b.timestamp) >= now-duration({hours:1})))
52     or ((b.name in $PART_List) AND (LocalDateTime(b.timestamp) >= now-duration({hours:24}))))
53     with lat,long, id(node) as id, node.name as Entity, b.name as pollutant, avg(b.value) as
54     value
55     with lat,long, id, Entity, pollutant, value, toInteger(custom.AQI(pollutant, value)) as
56     aqi_idx
57     with lat,long, id, Entity,aqi_idx, collect({pollutant:pollutant, pol_aqi:$AQI_Values
58     [aqi_idx], value:round(value, 2, 'CEILING')}) as pol
59     unwind pol as pollutants
60     return {name:Entity, point:{lat:lat, long:long},aqi:$AQI_Values[max(aqi_idx)],
61     pollutants:collect(pollutants)}
62     """
63   )
64 }
```

Σχήμα 123: Ερώτημα GraphQL τύπου Query για την ανάκτηση περιβαλλοντικών και γεωχωρικών δεδομένων.

```
{
  "SP_List": ["no2", "so2", "o3"],
  "PART_List": ["pm2.5", "pm10"],
  "AQI_Values": ["Good", "Fair", "Moderate", "Poor", "Very Poor", "Extremely Poor"]
}
```

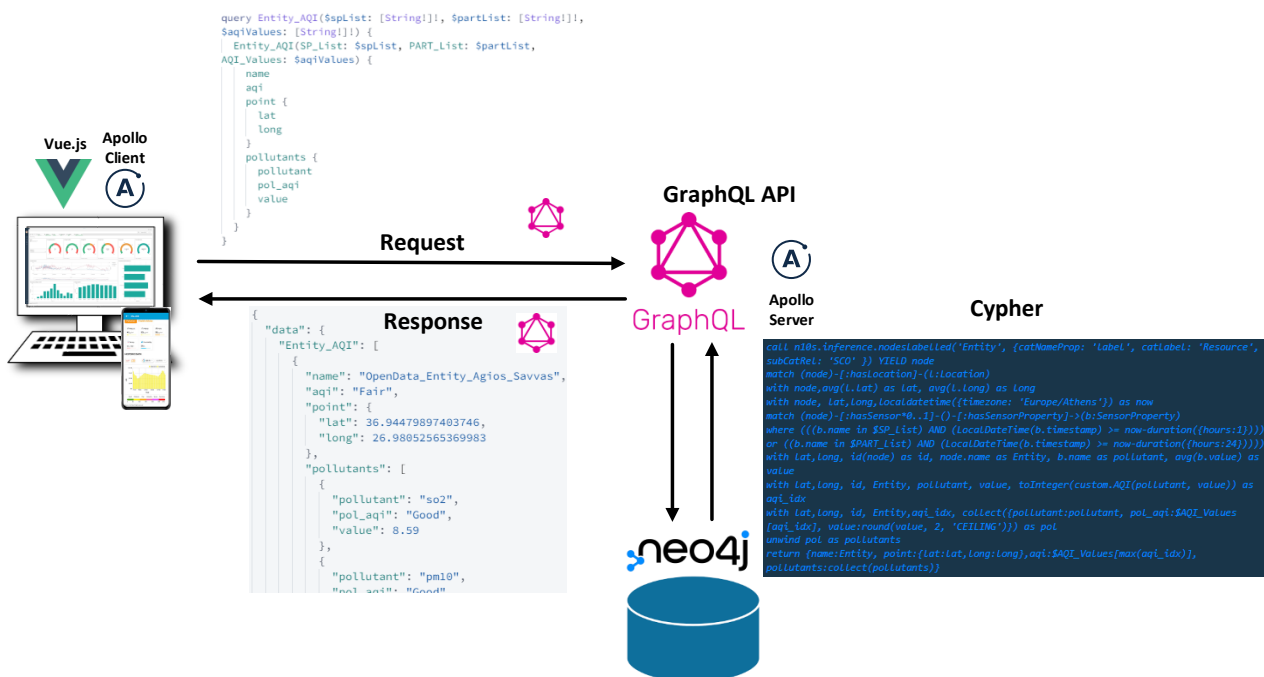
Σχήμα 124: Παράμετροι ερωτήματος (Σχήμα 123).

Ο κώδικας Cypher του ερωτήματος, χρησιμοποιώντας την πρόσθετη βιβλιοθήκη Neosemantics (ns10), ξεκινάει με τη συλλογιστική εργασία αναζήτησης των οντοτήτων με ετικέτα *Entity* καθώς και όλων εκείνων που έχουν χαρακτηριστεί ως υποκατηγορίες του με τη σχέση *SCO*. Για κάθε κόμβο σταθμό που επιστράφηκε αναζητούνται τα μονοπάτια που έχουν σχέση *hasLocation* και καταλήγουν σε κόμβο *Location*, υπολογίζοντας το μέσο γεωγραφικό μήκος και πλάτος των τοποθεσιών που σχετίζονται με κάθε σταθμό.

Στη συνέχεια, το ερώτημα χρησιμοποιεί τη συνάρτηση *localdatetime* για να λάβει την τρέχουσα ώρα στη ζώνη ώρας Ευρώπη/Αθήνα και αναζητά το μονοπάτι που ξεκινάει από κόμβο σταθμού (IoT_Entity ή OpenData_Entity) και καταλήγει σε κόμβο με ετικέτα *SensorProperty*, παρεμβάλλοντας τις σχέσεις *hasSensor* και *hasSensorProperty*. Αυτή η αντιστοίχιση πραγματοποιείται με μια σχέση μεταβλητού μήκους με μέγιστο βάθος 1, που σημαίνει ότι θα ταιριάζει με οποιονδήποτε αριθμό σχέσεων *hasSensor* μεταξύ 0 και 1.

Κατόπιν, οι κόμβοι *SensorProperty* που προέκυψαν φιλτράρονται ελέγχοντας εάν η ιδιότητα *name* του κόμβου βρίσκεται σε μια λίστα συγκεκριμένων ρύπων και εάν η διάρκεια της ιδιότητας *timestamp* από την τρέχουσα ώρα δεν υπερβαίνει τη μια ή τις 24 ώρες ανάλογα με τον ρύπο. Συγκεκριμένα, τα δεδομένα των αέριων ρύπων φιλτράρονται για το χρονικό διάστημα της τελευταίας ώρα από την στιγμή που εκτελείται το ερώτημα, ενώ τα δεδομένα των ρύπων αιωρούμενων σωματιδίων φιλτράρονται για το χρονικό διάστημα του τελευταίου εικοσιτετράωρου. Στη συνέχεια υπολογίζεται η μέση τιμή κάθε ρύπου και χρησιμοποιείται η συνάρτηση *custom.AQI* για τον υπολογισμό του δείκτη ποιότητας αέρα κάθε ρύπου. Τέλος, τα συλλεχθέντα και υπολογισμένα δεδομένα ομαδοποιούνται και επιστρέφονται με τη μορφή λίστας. Το τελικό αποτέλεσμα του ερωτήματος περιλαμβάνει το όνομα του σταθμού, τις γεωγραφικές του συντεταγμένες, το τελικό AQI και τη συλλογή των ρύπων (όνομα ρύπου, AQI, μέση τιμή).

Στο ακόλουθο σχήμα (Σχήμα 125) συγκεντρώνονται όλα τα τμήματα της διαδικασίας υποβολής του ερωτήματος προς τη βάση, που αναφέρθηκαν παραπάνω. Το διάγραμμα απεικονίζει πώς τα επιμέρους στοιχεία αλληλοεπιδρούν μεταξύ τους, ακολουθώντας τη ροή ενός αιτήματος από την εφαρμογή πελάτη, την αναζήτηση περιβαλλοντικών δεδομένων των σταθμών, την υποβολή στο GraphQL API, στη συνέχεια την επίλυση δεδομένων από τη βάση Neo4j και την επιστροφή τους στον πελάτη για την απόδοση των αποτελεσμάτων σε μια ενημερωμένη προβολή της διεπαφής του χρήστη.



Σχήμα 125: Αίτημα ανάκτησης περιβαλλοντικών δεδομένων, μέσω της εφαρμογής GraphQL πλήρους στοιβάς.

Μετά την ανάκτηση των περιβαλλοντικών δεδομένων των σταθμών, ο χρήστης καλείται να εισάγει τη γεωγραφική θέση της αφετηρίας και του προορισμού του στην εφαρμογή για τον υπολογισμό της προτεινόμενης διαδρομής. Για την υλοποίηση αυτού του τμήματος της εφαρμογής, χρησιμοποιήθηκε ένα ερώτημα GraphQL τύπου *Mutation*, καθώς το ερώτημα εγγράφει νέα δεδομένα στην βάση (Σχήμα 126). Το ερώτημα λαμβάνει σαν παραμέτρους έξι λίστες (Σχήμα 127) με τα ονόματα των αέριων ρύπων, των αιωρούμενων σωματιδίων, των λεκτικών δεικτών ποιότητας αέρα, των επιτρεπτών δεικτών ποιότητας αέρα καθώς και των γεωγραφικών σημείων έναρξης και λήξης της διαδρομής. Η απάντηση του ερωτήματος επιστρέφει μια λίστα με τα διαδοχικά γεωχωρικά δεδομένα των σταθμών, που σχηματίζουν την προτεινόμενη διαδρομή.

```
type Mutation{
  Path(SP_List: [String!]!, PART_List: [String!]!, AQI_Values: [String!]!, proper_aqi: [String!]!, start_point: user_point!, end_point: user_point!)
  : [path_points]
  @cypher(
    statement: """
optional match ()-[r:hasNeighbor]-()
delete r
with *
call n10s.inference.nodesLabelled('Entity', {catNameProp: 'Label', catLabel: 'Resource', subCatRel: 'SCD' }) YIELD node
match (node)-[:hasLocation]-(l:Location)
with node,avg(l.Lat) as lat, avg(l.Long) as Long
with node, lat,Long,LocalDateTime({timezone: 'Europe/Athens'}) as now
match (node)-[:hasSensor*0..1]-()-[:hasSensorProperty]->(b:SensorProperty)
where (((b.name in $SP_List) AND (LocalDateTime(b.timestamp) >= now-duration({hours:1}))) or ((b.name in $PART_List) AND (LocalDateTime(b.timestamp) >= now-duration({hours:24}))))
with lat,Long, id(node) as id, node.name as Entity, b.name as pollutant, avg(b.value) as value
with lat,Long, id, Entity, pollutant, value, toInteger(custom.AQI(pollutant, value)) as aqi_idx
with lat,Long, id, Entity, aqi_idx, collect({pollutant:pollutant, pol_aqi:$AQI_Values[aqi_idx], value:round(value, 2, 'CEILING')}) as pol
unwind pol as pollutants
with {id:id, name:Entity, point:point({latitude:lat,longitude:Long}),aqi:$AQI_Values[max(aqi_idx)]} as aqi_points
where aqi_points.aqi in $proper_aqi
with collect({id:aqi_points.id,name:aqi_points.name, point: aqi_points.point}) as points
unwind points as c
unwind points as d
with c, d where c.name <> d.name
with c, d, distance(c.point, d.point) as distance order by distance
with c, collect({id:d.id,name:d.name,weight:distance})[..4] as otherNodes
unwind otherNodes as on
with c, on
match (m),(n) where id(m)=c.id and id(n)=on.id and m<>n and id(m)<id(n)
create (m)-[:hasNeighbor{weight:on.weight}]->(n)
with apoc.coll.toSet(collect(id(m))+collect(id(n))) as id_list
match(z)-[:hasLocation]-(x:Location) where id(z) in id_list
with z,point({latitude: avg(x.Lat), longitude: avg(x.Long)}) as geo_points
with z,point($start_point) as start_point, point($end_point) as end_point, geo_points
with z,geo_points,[start_point,end_point] as se
unwind se as endpoints
with z,enápoints,distance(geo_points,endpoints) as dist order by dist
with endpoints,collect(z){@} as edge_nodes
with collect(edge_nodes) as edge
with edge[0] as start_node, edge[1] as end_node
CALL apoc.algo.dijkstra(start_node, end_node, 'hasNeighbor', 'weight') YIELD path, weight
unwind nodes(path) as path_entities
with collect(path_entities.name) as path_nodes
match (o)-[:hasLocation]-(p:Location) where o.name in path_nodes
with o,avg(p.Lat) as latitude, avg(p.Long) as longitude, apoc.coll.indexOf(path_nodes,o.name) as idx
return {idx:idx+1,name:o.name,lat:latitude,Long:longitude} as sequence order by idx
""")
}
```

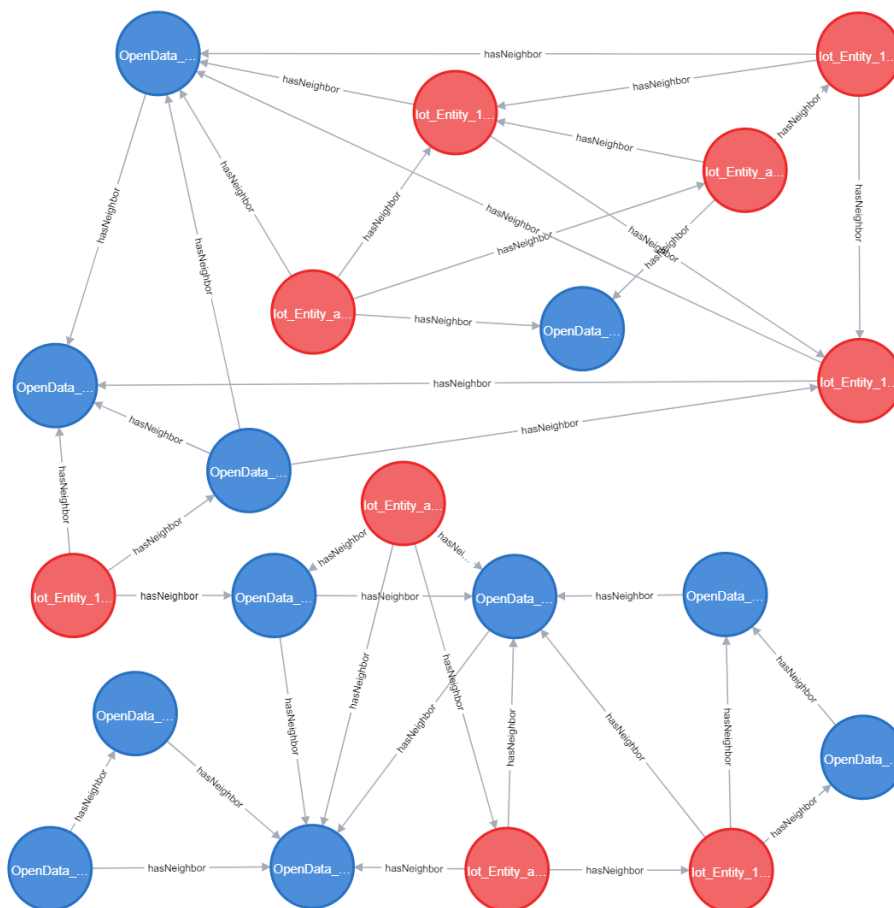
Σχήμα 126: Ερώτημα GraphQL τύπου *Mutation* για τον υπολογισμό την "Πράσινης Διαδρομής".

```
{
  "SP_List": ["no2", "so2", "o3"],
  "PART_List": ["pm2.5", "pm10"],
  "AQI_Values": ["Good", "Fair", "Moderate", "Poor", "Very Poor", "Extremely Poor"],
  "proper_aqi": ["Good", "Fair", "Moderate", "Poor"],
  "start_point": {latitude: 36.96413219670586, longitude: 26.952653220606674},
  "end_point": {latitude: 36.95530873325873, longitude: 26.976775119232496}
}
```

Σχήμα 127: Παράμετροι ερωτήματος (Σχήμα 126).

Για τον υπολογισμό της διαδρομής υπολογίζεται ο δείκτης ποιότητας αέρα όλων των σταθμών που βρίσκονται στην ευρύτερη γεωγραφική περιοχή των σημείων ενδιαφέροντος, όπως έγινε στο προηγούμενο ερώτημα. Τα αποτελέσματα φιλτράρονται διατηρώντας μόνο τους σταθμούς που ο δείκτης τους ανήκει στα ακόλουθα επίπεδα (Good, Fair, Moderate, Poor), όπως δηλώνεται από το όρισμα *\$proper_aqi*. Στη συνέχεια, το ερώτημα υπολογίζει τη χιλιομετρική απόσταση μεταξύ των σταθμών, για όλους τους δυνατούς συνδυασμούς, ταξινομεί τα αποτελέσματα κατά αύξουσα σειρά και επιστρέφει μια συλλογή δεδομένων που περιλαμβάνει τους τέσσερις πλησιέστερους κόμβους για κάθε ένα σταθμό ξεχωριστά.

Έπειτα, το ερώτημα δημιουργεί στη βάση νέες σταθμισμένες σημασιολογικές σχέσεις γειτονίας *hasNeighbor*, μεταξύ των φιλτραρισμένων κόμβων, σύμφωνα με την υπολογισμένη απόσταση. Το αποτέλεσμα είναι να δημιουργηθεί στη βάση ένας γράφος διασυνδεδεμένων σταθμών που θα χρησιμοποιηθεί αργότερα για τον υπολογισμό της συντομότερης πράσινης διαδρομής, προσδιορίζοντας τα σημεία από τα οποία θα περάσει ο χρήστης έως ότου φτάσει στον προορισμό του (Σχήμα 128). Για τις σταθμισμένες σχέσεις ως βάρος ορίστηκε η υπολογισμένη απόσταση μεταξύ των κόμβων.



Σχήμα 128: Γράφος διασυνδεδεμένων κόμβων με σχέσεις *hasNeighbor*.

Αξίζει να αναφερθεί ότι για τη σύναψη καινούργιων σχέσεων τύπου *hasNeighbor*, κάθε φορά που εκτελείται το ερώτημα, προστέθηκε στην αρχή του κώδικα Cypher η δυνατότητα διαγραφής

τυχόν υπάρχουσών σχέσεων *hasNeighbor* μεταξύ των οντοτήτων. Η ενέργεια αυτή διασφαλίζει τη δημιουργία νέων γράφων αναζήτησης της «Πράσινης διαδρομής» σύμφωνα με τα πιο πρόσφατα περιβαλλοντικά δεδομένα των σταθμών.

Στη συνέχεια, το ερώτημα υπολογίζει τους πλησιέστερους κόμβους (*start_node*, *end_node*) του παραπάνω γραφήματος, σε σχέση με τα σημεία έναρξης και λήξης της διαδρομής όπως προσδιορίστηκαν από το χρήστη και εισήχθησαν σαν όρισμα στο ερώτημα. Κατόπιν, χρησιμοποιείται ο αλγόριθμος Dijkstra μέσω της συνάρτησης *shortestPath*, για την αναζήτηση του συντομότερου μονοπατιού μεταξύ των κόμβων *start_node* και του *end_node* που συνδέονται με τη σχέση *hasNeighbor*. Η μέγιστη απόσταση μεταξύ των δύο κόμβων μπορεί να είναι 15 αναπηδήσεις. Στη συνέχεια, για κάθε ένα κόμβο του μονοπατιού υπολογίζεται ένας δείκτης διαδοχής ανάλογα με τη θέση του στο μονοπάτι και ξεκινώντας από τον κόμβο *start_node*. Τέλος, το ερώτημα επιστρέφει μια λίστα με την τελική ακολουθία των κόμβων στο μονοπάτι, συμπεριλαμβανομένου του ονόματος, του γεωγραφικού πλάτους, του γεωγραφικού μήκους και του δείκτη στη διαδρομή, ταξινομημένα με βάση τον δείκτη του κόμβου (Σχήμα 129).

```
{
  "data": {
    "Path": [
      {
        "idx": 1,
        "lat": 36.95664154400455,
        "long": 26.965767803949493,
        "name": "OpenData_Entity_Agios_Athanasios"
      },
      {
        "idx": 2,
        "lat": 36.964352463962754,
        "long": 26.956613291737053,
        "name": "OpenData_Entity_Taxiarchis"
      }
    ]
  }
}
```

Σχήμα 129: Απάντηση ερωτήματος υπολογισμού διαδρομής.

9.7.6 FrontEnd

Το frontend αποτελεί ένα Single Page Application, μια διαδικτυακή εφαρμογή που αλληλεπιδρά με τον χρήστη κάνοντας δυναμικές εγγραφές των δεδομένων στην τρέχουσα σελίδα, αντί της προεπιλεγμένης μεθόδου στην οποία γίνεται η φόρτωση ολόκληρων σελίδων. Η συγκεκριμένη εφαρμογή αποτελείται από ένα αρχείο HTML το οποίο είναι και ο κεντρικός κορμός της εφαρμογής ενώ το περιεχόμενο της αλλάζει δυναμικά μεταβάλλοντας το Document Object Model (DOM).

Για το απεικονιστικό τμήμα της εφαρμογής (frontend) χρησιμοποιήθηκε το framework Vue.js [61]. Στο Vue.js η δομή υλοποιείται με components όπου το καθένα από αυτά είναι ένα αρχείο με κατάληξη *.vue που χωρίζεται σε 3 διαφορετικά τμήματα.

- **Template:** Το τμήμα του κώδικα που χρησιμοποιείται για την απεικόνιση της εφαρμογής (HTML).
- **Style:** Το τμήμα του κώδικα που καθορίζει την μορφή του template (CSS).
- **Script:** Ο κώδικας της JavaScript που εκτελείται κάθε φορά που φορτώνετε το συγκεκριμένο component.

Σε συνδυασμό με το Vue.js, χρησιμοποιήθηκε το Leaflet [62] μια βιβλιοθήκη JavaScript ανοικτού κώδικα για τη δημιουργία διαδραστικών χαρτών σε διαδικτυακές εφαρμογές, το οποίο επιτρέπει στους προγραμματιστές να δημιουργούν χάρτες με προσαρμοσμένους δείκτες, αναδυόμενα παράθυρα και διάφορα άλλα χαρακτηριστικά. Έχει σχεδιαστεί για να είναι ελαφρύ και εύκολο στη χρήση και λειτουργεί με μια μεγάλη ποικιλία παρόχων χαρτογράφησης, συμπεριλαμβανομένων των OpenStreetMap, Google Maps και Mapbox.

Επιπλέον, χρησιμοποιήθηκε το Leaflet-Routing-Machine (LRM) [63], ένα πρόσθετο για τη βιβλιοθήκη Leaflet, που παρέχει λειτουργικότητα δρομολόγησης για χάρτες. Επιτρέπει στους προγραμματιστές να προσθέτουν οδηγίες στροφή προς στροφή στους χάρτες τους, συμπεριλαμβανομένων επιλογών για διαφορετικούς τρόπους μεταφοράς, σημεία διαδρομής και προσαρμοσμένους δείκτες. Το πρόσθετο αυτό χρησιμοποιήθηκε για τον προσδιορισμό και την απεικόνιση της προτεινόμενης «Πράσινης διαδρομής» στο χάρτη.

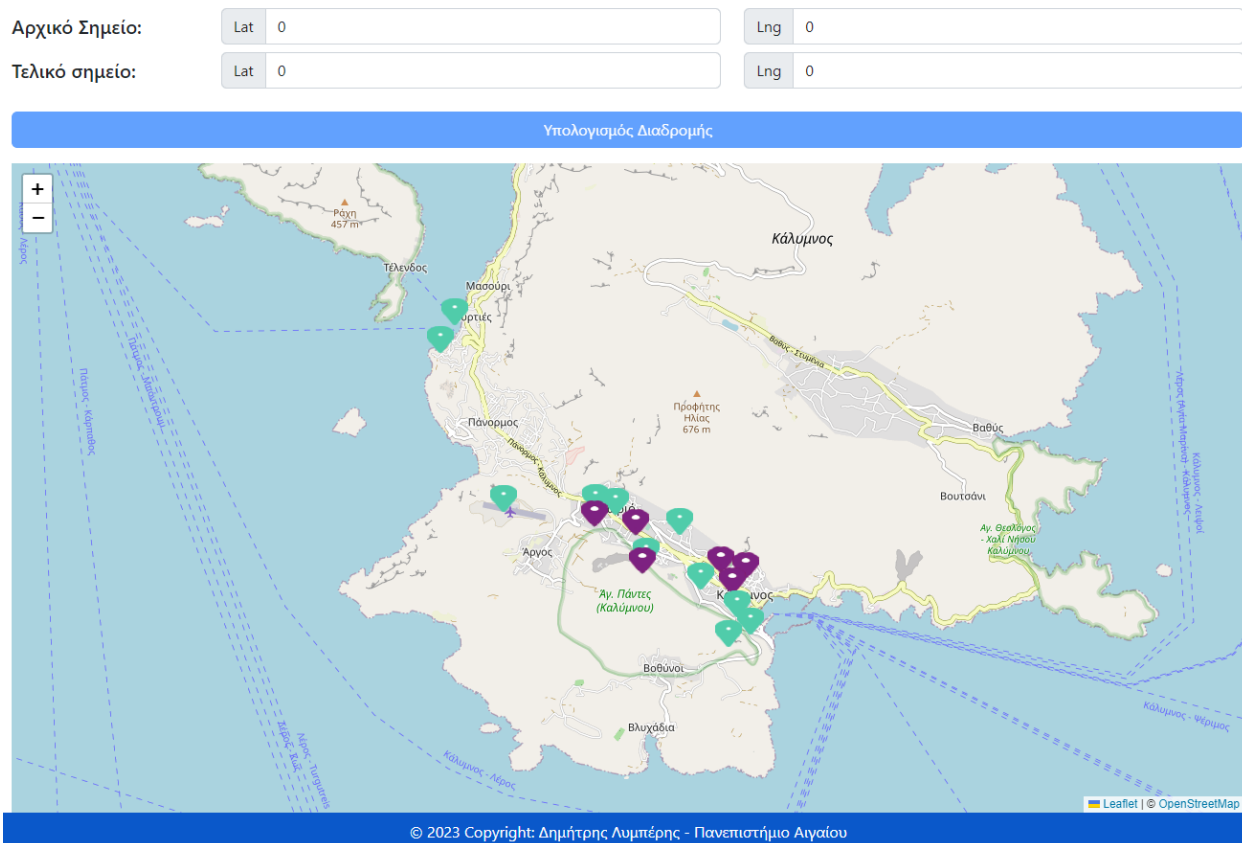
Ένα ακόμη εργαλείο το οποίο χρησιμοποιήθηκε για την δημιουργία της διαδικτυακής εφαρμογής είναι το Mapbox [64], μια βιβλιοθήκη JavaScript για τη δημιουργία διανυσματικών χαρτών. Είναι χτισμένο πάνω στο WebGL, ένα Javascript API που χρησιμοποιείται για την αναπαράσταση διαδραστικών 3D και 2D γραφικών και παρέχει μια εξαιρετικά προσαρμόσιμη, υψηλής απόδοσης εμπειρία απόδοσης γεωγραφικών χαρτών. Το Mapbox GL μπορεί να χρησιμοποιηθεί με το Leaflet, αλλά μπορεί επίσης να χρησιμοποιηθεί και ως αυτόνομη βιβλιοθήκη.

Τέλος για τη σχεδίαση της ιστοσελίδας χρησιμοποιήθηκε το Bootstrap [65], το οποίο αποτελεί ένα από τα πιο δημοφιλή πλαίσια ανάπτυξης front-end. Παρέχει ένα σύνολο στοιχείων CSS και JavaScript, όπως κουμπιά, φόρμες και γραμμές πλοήγησης, που μπορούν να χρησιμοποιηθούν για τη δημιουργία της διάταξης του περιεχομένου της ιστοσελίδας καθώς και της δυναμικής απόκρισής του σε οποιοδήποτε μέγεθος οθόνης.

Οι παραπάνω τεχνολογίες συνδυάστηκαν για τη δημιουργία μιας διαδικτυακής εφαρμογής που εμφανίζει έναν διαδραστικό χάρτη με λειτουργικότητα δρομολόγησης. Η βιβλιοθήκη Leaflet χρησιμοποιήθηκε για τη δημιουργία του χάρτη και την παροχή λειτουργικότητας δρομολόγησης μέσω του πρόσθετου leaflet-routing-machine. Το Mapbox GL χρησιμοποιήθηκε για την προσθήκη πρόσθετων χαρακτηριστικών και προσαρμοσμένης μορφής στο χάρτη. Τέλος, το Bootstrap χρησιμοποιήθηκε για τη σχεδίαση και τη δημιουργία της διάταξης της εφαρμογής, διασφαλίζοντας δυναμική απόκριση του περιεχομένου ακόμη και σε κινητές συσκευές.

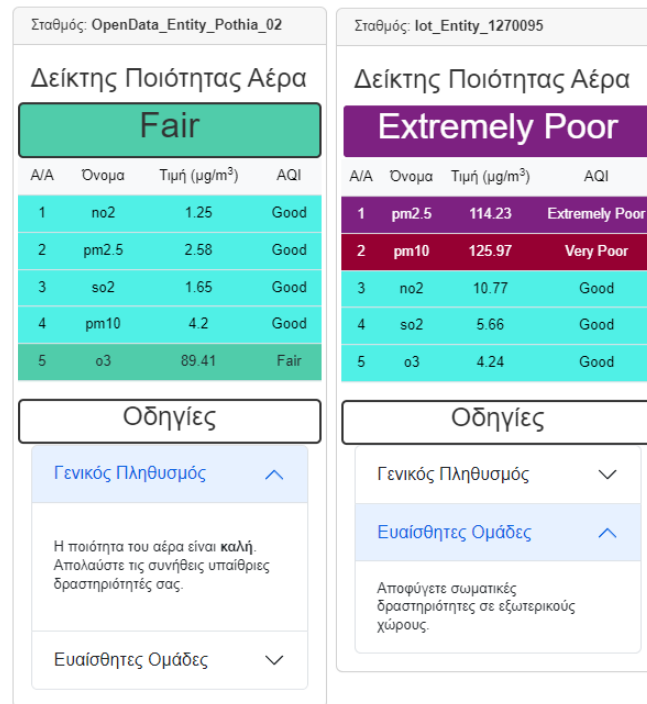
Όπως αναφέρθηκε και σε προηγούμενη ενότητα, κατά το άνοιγμα της εφαρμογής από τον χρήστη, υποβάλλεται ένα GraphQL ερώτημα τύπου *query* στη βάση, για την ανάκτηση των περιβαλλοντικών δεδομένων των σταθμών παρατήρησης που βρίσκονται στη γεωγραφική περιοχή

που απεικονίζεται στο χάρτη. Το ερώτημα επιστρέφει όλες τις τιμές των συγκεντρώσεων και τους επιμέρους δείκτες ποιότητας αέρα όλων των ρύπων παρακολούθησης για κάθε σταθμό. Επίσης επιστρέφεται ο γενικός δείκτης AQI κάθε σταθμού της περιοχής, καθώς και τα γεωγραφικά του δεδομένα για την απεικόνισή του στο χάρτη (Σχήμα 130). Οι σταθμοί εμφανίζονται με γεωγραφικούς δείκτες πάνω στο χάρτη που το χρώμα τους προκύπτει από το επίπεδο του δείκτη AQI (Σχήμα 1).



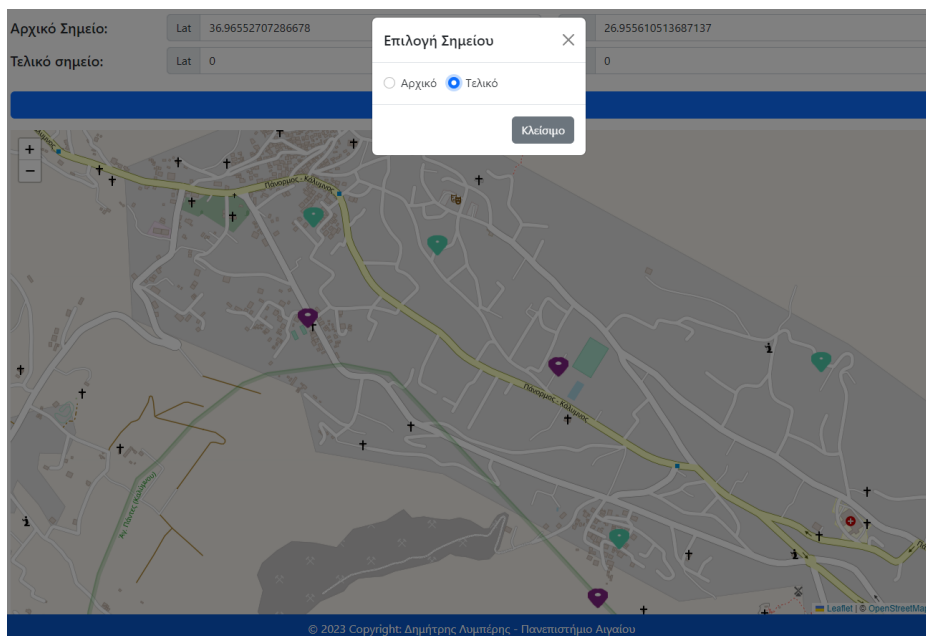
Σχήμα 130: Αρχική οθόνη εφαρμογής.

Επιλέγοντας τον γεωγραφικό δείκτη ενός σταθμού, στην οθόνη εμφανίζεται ένα αναδυόμενο παράθυρο με τα περιβαλλοντικά του δεδομένα (Σχήμα 131). Στο επάνω τμήμα του παραθύρου απεικονίζεται ο γενικός δείκτης ποιότητας αέρα του σταθμού με τη χρωματική του αντιστοίχιση. Ακολουθεί ένας πίνακας με τις υπολογισμένες συγκεντρώσεις των ρύπων σε $\mu\text{g}/\text{m}^3$ και τον αντίστοιχο δείκτη AQI για κάθε έναν από αυτούς. Με τον διαφορετικό χρωματισμό των επιμέρους ρύπων, ο χρήστης είναι σε θέση να κατανοήσει ποιοι από αυτούς είναι υπεύθυνοι για τη διαμόρφωση του γενικού δείκτη AQI. Στο τελευταίο τμήμα εμφανίζονται οδηγίες που σχετίζονται με την υγεία των ανθρώπων και παρέχουν συστάσεις τόσο για τον γενικό όσο και για τον ευαίσθητο πληθυσμό. Τα μηνύματα έχουν συμβουλευτικό χαρακτήρα και αφορούν τις δραστηριότητες που μπορούν να εκτελεστούν, δεδομένου του AQI.



Σχήμα 131: Παραδείγματα αναδυόμενου παραθύρου απεικόνισης περιβαλλοντικών δεδομένων σταθμού.

Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να εισάγει το σημείο έναρξης και τερματισμού μιας περιπατητικής διαδρομής που επιθυμεί να διανύσει. Ο ορισμός των άκρων της διαδρομής πραγματοποιείται με την επιλογή του αντίστοιχου σημείου πάνω στο χάρτη. Σε κάθε επιλογή εμφανίζεται ένα αναδυόμενο παράθυρο και ο χρήστης καλείται να προσδιορίσει αν το σημείο αυτό είναι το αρχικό ή το τελικό (Σχήμα 132). Στο επάνω μέρος της εφαρμογής, μετά από κάθε επιλογή, εμφανίζονται οι γεωγραφικές συντεταγμένες του σημείου.



Σχήμα 132: Επιλογή σημείων έναρξης και λήξης διαδρομής στο χάρτη.

Πιέζοντας το κουμπί «Υπολογισμός Διαδρομής», ένα ερώτημα GraphQL τύπου *Mutation* υποβάλλεται στη βάση. Η απάντηση του ερωτήματος επιστρέφει μια λίστα με τα διαδοχικά γεωγραφικά δεδομένα των σταθμών, που θα σχηματίζουν την προτεινόμενη διαδρομή. Στο σημείο αυτό, το πρόσθετο Leaflet-Routing-Machine (LRM) παραλαμβάνει το σημείο αφετηρίας, τους ενδιάμεσους σταθμούς και το σημείο προορισμού, και υπολογίζει το μονοπάτι. Τέλος, η εφαρμογή απεικονίζει τη διαδρομή στο χάρτη με πράσινο χρώμα, στα σημεία έναρξης και λήξης τοποθετεί γεωγραφικούς δείκτες με μπλε χρώμα και εμφανίζει ένα παράθυρο με τις οδηγίες πλοήγησης της διαδρομής.

Υπολογισμός Διαδρομής

Αρχικό Σημείο: Lat 36.96514113445453 Lng 26.952698707646054

Τελικό σημείο: Lat 36.94686895760853 Lng 26.983632087740265

Πάνορμος - Κάλυμνος, Πάνορμος - Κάλυμνος, Πάνορμος - Κάλυμνος, .
4.7 km, 52 min

- ➡ Head east on Πάνορμος - Κάλυμνος 450 m
- ➡ Turn right 45 m
- ⬅ Turn left 60 m
- ✓ You have arrived at your 1st destination, on the right 0 m
- ➡ Head northeast 60 m
- ➡ Turn right 45 m
- ⬅ Turn left onto Πάνορμος - Κάλυμνος 9 m
- ➡ Turn right 20 m
- ➡ Turn right 80 m
- ➡ Turn right 100 m
- ⬅ Turn left 70 m

© 2023 Copyright: Δημήτρης Λυμπέρης - Πανεπιστήμιο Αιγαίου

Σχήμα 133: Απεικόνιση προτεινόμενης «Πράσινης Διαδρομής» στο χάρτη.

10

Συμπεράσματα

Σε αυτή την εργασία μελετήθηκαν και καταγράφηκαν υπηρεσίες και μέθοδοι για τη συλλογή μεγάλου όγκου γεωχωρικών δεδομένων από ετερογενή IoT συστήματα. Τα γεωχωρικά δεδομένα συλλέγονται από κόμβους οι οποίοι διαθέτουν ενσωματωμένους μικροελεγκτές διαφορετικού κατασκευαστή και χρησιμοποιούν διαφορετικές τεχνολογίες ασύρματης/ενσύρματης επικοινωνίας και δικτύωσης. Τα δεδομένα αυτά εμπλουτίζονται και από τρίτες πηγές μέσω διασυνδέσεων προγραμματισμού εφαρμογών (APIs) με σκοπό τη μελέτη των μετρούμενων μεγεθών σε γεωγραφικές περιοχές ενδιαφέροντος. Η εργασία αυτή υλοποίησε την κατάλληλη υποδομή ενδιάμεσου λογισμικού για τη συλλογή και διαχείριση περιβαλλοντικών και γεωχωρικών δεδομένων. Τα δεδομένα αυτά προετοιμάζονται κατάλληλα και τροφοδοτούν την εφαρμογή ανεύρεσης της καταλληλότερης διαδρομής στο χάρτη, με τους λιγότερους ατμοσφαιρικούς ρύπους, που μπορεί να διανύσει ένας χρήστης.

Για την ανίχνευση και παρακολούθηση της συγκέντρωσης των ρύπων, χρησιμοποιήθηκαν αισθητήρες αερίων και αιωρούμενων σωματιδίων χαμηλού κόστους. Οι αισθητήρες αερίων αποτέλεσαν το δυσκολότερο σημείο της έρευνας, εξαιτίας της έλλειψης πληροφοριών σχετικά με την βαθμονόμησή τους. Δοκιμάστηκε πληθώρα βιβλιοθηκών ανοικτού κώδικα χωρίς ενθαρρυντικά αποτελέσματα, αφήνοντας αμφιβολίες για την αξιοπιστία των μετρήσεων. Συμπληρωματικά, να αναφερθεί ότι σε αντίστοιχες έρευνες, υπήρξαν περιγραφές βαθμονόμησης τέτοιων αισθητήρων με ελλιπή στοιχεία για την αναπαραγωγή της διαδικασίας από τρίτους. Από την έρευνα που πραγματοποιήθηκε προέκυψε μια αναλυτική τεκμηρίωση της διαδικασίας βαθμονόμησης, η οποία μπορεί να χρησιμοποιηθεί στο μέλλον σε παρόμοιες μελέτες. Παρόλα αυτά, η χρήση κατάλληλης συσκευής βαθμονόμησης, με έκθεση των αισθητήρων σε αέρια γνωστής συγκέντρωσης, είναι επιβεβλημένη.

Η τεχνολογία LoRa και το πρωτόκολλο LoRaWAN αποτέλεσαν πολύ καλή επιλογή στην υλοποίηση των δικτύων των IoT συσκευών. Η μεγάλη εμβέλεια, η ανοχή στο θόρυβο και στις παρεμβολές, η υψηλή ενεργειακή απόδοση και η κλιμακωσιμότητα αποτελούν αξιόλογα χαρακτηριστικά για την αύξηση της γεωγραφικής κάλυψης στο πλαίσιο της εφαρμογής. Η χρήση του The Things Stack ως ενδιάμεσης υποδομής παρείχε ευκολία στη διαχείριση και τον έλεγχο των απομακρυσμένων συσκευών. Προσέφερε τη δυνατότητα εύκολης διασύνδεσης με εφαρμογές υψηλού επιπέδου (integrations) όπως είναι οι IoT πλατφόρμες Thingspeak και Ubidots, μέσω διεπαφών προγραμματισμού εφαρμογών APIs. Επίσης, η χρήση των ενσωματωμένων μορφοποιητών ωφέλιμης πληροφορίας του The Things Stack, επέτρεψαν την γρήγορη αποκωδικοποίηση των δεδομένων που παρήγαγαν οι τελικές συσκευές και την αποστολή τους στις IoT πλατφόρμες.

Η εξασφάλιση της διαλειτουργικότητας και ο χειρισμός ετερογενών δεδομένων μεγάλης κλίμακας αποτέλεσε βασικό παράγοντα για την ανάπτυξη της εφαρμογής «Πράσινη Διαδρομή». Σημαντική συνεισφορά στο στάδιο υλοποίησης αποτέλεσε η δυνατότητα επικοινωνίας μέσω του πρωτοκόλλου MQTT που προσέφεραν οι IoT πλατφόρμες. Η αξιοποίηση της δωρεάν υποδομής που παρέχεται από αυτές (MQTT Broker), ακόμη και στη δωρεάν έκδοση, συνέβαλε στην αντιμετώπιση της ετερογένειας στη δικτύωση και στα δικτυακά πρωτόκολλα που χρησιμοποιούν οι IoT συσκευές για την αποστολή της πληροφορίας. Επιπρόσθετα, οι IoT πλατφόρμες με την αποθήκευση και την εξερεύνηση της ιστορικότητας των δεδομένων των τελικών συσκευών, παρέχουν επιπρόσθετη ασφάλεια στη διαφύλαξη της πρωτογενούς πληροφορίας. Αν και οι υπηρεσίες που παρέχουν θα μπορούσαν να υλοποιηθούν από εξατομικευμένες εφαρμογές, η χρήση τους αποδεικνύεται πολύτιμη επιταχύνοντας τις διαδικασίες ανάπτυξης των τελικών εφαρμογών και μειώνοντας ταυτόχρονα το κόστος υλοποίησής τους σε ανθρωποημέρες.

Συμπερασματικά, η χρήση τεχνικών σημασιολογικού ιστού στην παρακολούθηση του δείκτη ποιότητας αέρα IoT σε αστικά περιβάλλοντα, αποδεικνύεται πολύτιμο εργαλείο για τη συλλογή και ανάλυση ετερογενών δεδομένων. Μέσω της χρήσης της οντολογίας και των προσαρμοσμένων εργαλείων σχολιασμού δεδομένων, τα δεδομένα μπορούν να εμπλουτιστούν σημασιολογικά και να αποκτηθεί μια βαθύτερη κατανόηση των προτύπων και των τάσεων της ατμοσφαιρικής ρύπανσης στα αστικά περιβάλλοντα. Επιπλέον, η χρήση βάσεων δεδομένων γραφημάτων όπως το Neo4j και γλωσσών ερωτημάτων όπως η GraphQL διευκολύνει την αποθήκευση, την ανάκτηση και την ανάλυση εμπλουτισμένων δεδομένων. Ο συνδυασμός αυτών των τεχνολογιών, μαζί με εργαλεία χαρτογράφησης όπως το Leaflet, παρέχει ένα ολοκληρωμένο και αποτελεσματικό πλαίσιο για την παρακολούθηση της ποιότητας του αέρα σε έξυπνες πόλεις. Τα αποτελέσματα αυτής της μελέτης δείχνουν τη δυνατότητα των τεχνολογιών σημασιολογικού Ιστού να διαδραματίσουν κρίσιμο ρόλο στις έξυπνες πόλεις και στη βελτίωση της ποιότητας του αέρα στις αστικές περιοχές. Αξιοποιώντας τη δύναμη του σημασιολογικού ιστού και των τεχνολογιών IoT, μπορεί να αποκτηθεί μια ολοκληρωμένη εικόνα της ατμοσφαιρικής ρύπανσης σε πραγματικό χρόνο, παρέχοντας τα απαραίτητα δεδομένα για τη λήψη τεκμηριωμένων αποφάσεων για ένα ποιοτικότερο και πιο υγιές αστικό περιβάλλον.

Βιβλιογραφία

- [1] “7 million premature deaths annually linked to air pollution.” <https://www.who.int/news/item/25-03-2014-7-million-premature-deaths-annually-linked-to-air-pollution> (accessed Oct. 10, 2022).
- [2] “Ambient (outdoor) air pollution.” [https://www.who.int/en/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/en/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health) (accessed Oct. 10, 2022).
- [3] “World Health Organization Regional Office for Europe,” 2006, Accessed: Oct. 10, 2022. [Online]. Available: www.euro.who.int
- [4] “WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide.” <https://apps.who.int/iris/handle/10665/345329> (accessed Oct. 10, 2022).
- [5] European Environment Agency, “European Air Quality Index.” <https://airindex.eea.europa.eu/Map/AQI/Viewer/> (accessed Jul. 29, 2022).
- [6] T. K. Duy, G. Quirchmayr, A. Tjoa, and H. H. Hanh, “A semantic data model for the interpretation of environmental streaming data,” *7th International Conference on Information Science and Technology, ICIST 2017 - Proceedings*, pp. 376–380, May 2017, doi: 10.1109/ICIST.2017.7926788.
- [7] M. Compton *et al.*, “The SSN ontology of the W3C semantic sensor network incubator group,” *Journal of Web Semantics*, vol. 17, pp. 25–32, Dec. 2012, doi: 10.1016/J.WEBSEM.2012.05.003.
- [8] B. Sejdiu, F. Ismaili, and L. Ahmedi, “A Real-Time Integration of Semantic Annotations into Air Quality Monitoring Sensor Data,” *Communications in Computer and Information Science*, vol. 1447, pp. 98–113, 2021, doi: 10.1007/978-3-030-83007-6_5/FIGURES/7.
- [9] B. Sejdiu, F. Ismaili, and L. Ahmedi, “IoTSAS: An Integrated System for Real-Time Semantic Annotation and Interpretation of IoT Sensor Stream Data,” *Computers 2021, Vol. 10, Page 127*, vol. 10, no. 10, p. 127, Oct. 2021, doi: 10.3390/COMPUTERS10100127.
- [10] L. Noussair, J. T. Fernández Breis, J. Zahir, and H. Mousannif, “Towards distributed learning in internet of things. air quality monitoring use case,” *Communications in Computer and Information Science*, vol. 1085, pp. 154–159, 2019, doi: 10.1007/978-3-030-32213-7_12/FIGURES/3.
- [11] J. N. S. Rubí and P. R. de Lira Gondim, “IoT-based platform for environment data sharing in smart cities,” *International Journal of Communication Systems*, vol. 34, no. 2, p. e4515, Jan. 2021, doi: 10.1002/DAC.4515.
- [12] K. Eldahshan, E. K. Elsayed, and H. Mancy, “Semantic Smart World Framework,” *Applied Computational Intelligence and Soft Computing*, vol. 2020, 2020, doi: 10.1155/2020/8081578.
- [13] N. Zhang, H. Chen, X. Chen, and J. Chen, “Semantic Framework of Internet of Things for Smart Cities: Case Studies,” *Sensors 2016, Vol. 16, Page 1501*, vol. 16, no. 9, p. 1501, Sep. 2016, doi: 10.3390/S16091501.
- [14] J.-P. Calbimonte, J. Eberle, and K. Aberer, “Semantic Data Layers in Air Quality Monitoring for Smarter Cities,” *Proceedings of the Sixth Workshop on Semantics for Smarter Cities*, pp. 3–19, 2015, Accessed: Dec. 28, 2022. [Online]. Available: <https://infoscience.epfl.ch/record/212731>

- [15] “UNO WiFi Rev2 | Arduino Documentation | Arduino Documentation.” <https://docs.arduino.cc/hardware/uno-wifi-rev2> (accessed Jul. 31, 2022).
- [16] “Mega 2560 Rev3 | Arduino Documentation | Arduino Documentation.” <https://docs.arduino.cc/hardware/mega-2560?queryID=69056c2bd9c0bd7272883fd0ef67d5b7> (accessed Aug. 01, 2022).
- [17] “LILYGO® TTGO LORA32 868/915Mhz ESP32 LoRa OLED 0.96”.” http://www.lilygo.cn/prod_view.aspx?TypeId=50060&Id=1326&FId=t3:50060:3 (accessed Aug. 02, 2022).
- [18] “TTGO LoRa32 SX1276 OLED Board Pinout and Use with Arduino IDE.” <https://microcontrollerslab.com/ttgo-lora32-sx1276-oled-board-pinout-getting-started-with-arduino-ide/> (accessed Aug. 02, 2022).
- [19] “TTGO LoRa32 SX1276 OLED with Arduino IDE | Random Nerd Tutorials.” <https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide/> (accessed Aug. 02, 2022).
- [20] Pycom, “Lopy4 Datasheet Version 1.1,” Mar. 2020. https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_LoPy4_v2.pdf (accessed Aug. 05, 2022).
- [21] Semtech Corporation, “LoRa and LoRaWAN: A Technical Overview LoRa® and LoRaWAN®: A Technical Overview,” 2020, Accessed: Sep. 07, 2022. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf
- [22] L. Oliveira, J. J. P. C. Rodrigues, S. A. Kozlov, R. A. L. Rabêlo, and V. H. C. de Albuquerque, “MAC layer protocols for internet of things: A survey,” *Future Internet*, vol. 11, no. 1, Jan. 2019, doi: 10.3390/FI11010016.
- [23] “The Things Network - Greece.” <https://www.thethingsnetwork.org/country/greece/> (accessed Sep. 24, 2022).
- [24] “Facebook using MQTT | Eclipse Mosquitto.” <https://mosquitto.org/blog/2011/08/facebook-using-mqtt/> (accessed Sep. 10, 2022).
- [25] D. Soni and A. Makwana, “A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT) MP-Index View project Analysis and Survey on String Matching Algorithms for Ontology Matching View project A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT),” 2017, Accessed: Sep. 10, 2022. [Online]. Available: <https://www.researchgate.net/publication/316018571>
- [26] “mqtt-v3.1.1-errata01-os-complete Specification URIs,” 2015, Accessed: Sep. 11, 2022. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.doc>
- [27] “Learn More - ThingSpeak IoT.” https://thingspeak.com/pages/learn_more (accessed Sep. 11, 2022).
- [28] S. Raj and D. Nettikadan, “Smart Community Monitoring System using Thingspeak IoT Platform,” *Article in International Journal of Applied Engineering Research*, vol. 13, pp. 13402–13408, 2018, Accessed: Sep. 11, 2022. [Online]. Available: <http://www.ripublication.com>
- [29] “IoT platform features.” <https://ubidots.com/platform/> (accessed Sep. 15, 2022).

- [30] “How to Buy - ThingSpeak IoT.” <https://thingspeak.com/prices> (accessed Sep. 24, 2022).
- [31] “How many ‘dots’ can I send to Ubidots? | Ubidots Help Center.” <https://help.ubidots.com/en/articles/2093902-how-many-dots-can-i-send-to-ubidots> (accessed Sep. 24, 2022).
- [32] Zhou Yong, “PMS5003 Series Manual Datasheet,” Jun. 01, 2016. <https://www.digikey.gr/htmldatasheets/production/2903006/0/0/1/pms5003-series-manual.html> (accessed Jul. 29, 2022).
- [33] Adrián Fernandez *et al.*, “World Health Organization Regional Office for Europe,” 2006, Accessed: Jul. 29, 2022. [Online]. Available: www.euro.who.int
- [34] Henan Hanwei Electronics Co, “MQ131 Datasheet.” <https://www.digikey.in/htmldatasheets/production/2483660/0/0/1/mq131.html> (accessed Jul. 29, 2022).
- [35] Zhengzhou Winsen Electronics Technology Co., “Hydrogen Sulfide Gas Sensor, Model MQ136.” [https://www.winsen-sensor.com/d/files/mq136-\(ver1_6\)---manual.pdf](https://www.winsen-sensor.com/d/files/mq136-(ver1_6)---manual.pdf) (accessed Jul. 29, 2022).
- [36] V.KEL Communications Equipment (SHENZHEN) Co., “GPS Module VK2828U7G5LF.” <https://www.openhacks.com/uploads/productos/vk2828u7g5lf.pdf> (accessed Jul. 29, 2022).
- [37] “National Marine Electronics Association - NMEA.” https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard (accessed Jul. 30, 2022).
- [38] R. Senthilkumar, P. Venkatakrishnan, and N. Balaji, “Intelligent based novel embedded system based IoT enabled air pollution monitoring system,” *Microprocess Microsyst*, vol. 77, p. 103172, Sep. 2020, doi: 10.1016/J.MICPRO.2020.103172.
- [39] Z. Idrees, Z. Zou, and L. Zheng, “Edge Computing Based IoT Architecture for Low Cost Air Pollution Monitoring Systems: A Comprehensive System Analysis, Design Considerations & Development,” *Sensors 2018, Vol. 18, Page 3021*, vol. 18, no. 9, p. 3021, Sep. 2018, doi: 10.3390/S18093021.
- [40] K. K. Sai and S. Mukherjee, “Low Cost IoT Based Air Quality Monitoring Setup Using Arduino and MQ Series Sensors With Dataset Analysis,” *Procedia Comput Sci*, vol. 165, pp. 322–327, 2019, doi: 10.1016/j.procs.2020.01.043.
- [41] M. Hnatiuc, D. Dorcea, and I. Lazar, “Acquisition and Calibration Interface for Gas Sensors Bioengineering View project underwater system View project Acquisition and Calibration Interface for Gas Sensors,” 2018, doi: 10.1109/SIITME.2018.8599253.
- [42] Ankit Rohatgi, “WebPlotDigitizer - Extract data from plots, images, and maps.” <https://automeris.io/WebPlotDigitizer/> (accessed Jul. 29, 2022).
- [43] K Senthil Babu and C Nagaraja, “Calibration of MQ-7 and Detection of Hazardous Carbon Monoxide Concentration in Test Canister Related papers,” *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 1, pp. 18–24, 2018.
- [44] K. Visvam Devadoss Ambeth, “Human security from death defying gases using an intelligent sensor system,” *Sens Biosensing Res*, vol. 7, pp. 107–114, Mar. 2016, doi: 10.1016/J.SBSR.2016.01.006.
- [45] T. K. Boguski, “Environmental Science and Technology Briefs for Citizens Understanding Units of Measurement,” 2006, Accessed: Oct. 25, 2022. [Online]. Available: www.engg.ksu.edu/CHSR/

- [46] “PPM mg/m3 converter for gases | Teesing.” <https://teesing.com/en/library/tools/ppm-mg3-converter> (accessed Oct. 25, 2022).
- [47] “Working with Bytes | The Things Network.” <https://www.thethingsnetwork.org/docs/devices-and-gateways/bytes/> (accessed Nov. 07, 2022).
- [48] “AirVisual API.” <https://api-docs.iqair.com/> (accessed Nov. 04, 2022).
- [49] “Air Quality API - API Ninjas.” <https://api-ninjas.com/api/airquality> (accessed Nov. 04, 2022).
- [50] “Air Quality API | Current Air Quality - Weatherbit.” <https://www.weatherbit.io/api/airquality-current> (accessed Nov. 04, 2022).
- [51] “Air Pollution - OpenWeatherMap.” <https://openweathermap.org/api/air-pollution> (accessed Nov. 04, 2022).
- [52] “LORIX One LoRaWAN gateway - Wifx IoT.” <https://iot.wifx.net/en/products/lorix-one/#documentation> (accessed Sep. 25, 2022).
- [53] “Root Certificates | The Things Stack for LoRaWAN.” <https://www.thethingsindustries.com/docs/reference/root-certificates/> (accessed Oct. 01, 2022).
- [54] “Eclipse Paho | The Eclipse Foundation.” <https://www.eclipse.org/paho/index.php?page=clients/python/index.php> (accessed Nov. 26, 2022).
- [55] “protégé.” <https://protege.stanford.edu/> (accessed Jan. 12, 2023).
- [56] “GraphQL: A data query language - Engineering at Meta.” <https://engineering.fb.com/2015/09/14/core-data/graphql-a-data-query-language/> (accessed Jan. 21, 2023).
- [57] “Introduction to GraphQL | GraphQL.” <https://graphql.org/learn/> (accessed Jan. 21, 2023).
- [58] W. Lyon, “Fullstack GraphQL Applications with GRANDstack Essential Excerpts Neo4j,” pp. 11–13, 2022.
- [59] “Introduction to Apollo Server - Apollo GraphQL Docs.” <https://www.apollographql.com/docs/apollo-server/> (accessed Jan. 24, 2023).
- [60] “Vue Apollo.” <https://apollo.vuejs.org/> (accessed Jan. 24, 2023).
- [61] “Introduction | Vue.js.” <https://vuejs.org/guide/introduction.html> (accessed Feb. 02, 2023).
- [62] “Documentation - Leaflet - a JavaScript library for interactive maps.” <https://leafletjs.com/reference.html> (accessed Feb. 02, 2023).
- [63] “Leaflet Routing Machine.” <https://www.liedman.net/leaflet-routing-machine/> (accessed Feb. 02, 2023).
- [64] “API Reference | Mapbox GL JS | Mapbox.” <https://docs.mapbox.com/mapbox-gl-js/api/> (accessed Feb. 02, 2023).
- [65] “Get started with Bootstrap · Bootstrap v5.2.” <https://getbootstrap.com/docs/5.2/getting-started/introduction/> (accessed Feb. 02, 2023).