



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΑΠΟΦΑΣΕΩΝ**

**Υλοποίηση Lambda Architecture για Δεδομένα Μεγάλης
Κλίμακας**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΕΤΣΑ ΙΩΑΝΝΗ

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2021



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Υλοποίηση Lambda Architecture για Δεδομένα Μεγάλης Κλίμακας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΕΤΣΑ ΙΩΑΝΝΗ

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή τη 18^η Μαρτίου 2021.

.....
Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

.....
Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2021

.....

ΠΕΤΣΑΣ ΙΩΑΝΝΗΣ

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

Copyright © Πέτσας Ιωάννης, 2021

Με επιφύλαξη παντός δικαιώματος. All rights reserved

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Εισαγωγή: Ο όρος «Μεγάλα Δεδομένα» χαρακτηρίζεται από τεράστιο όγκο, ποικιλία και συνεχή συσσώρευση νέων δεδομένων, από πολλές πηγές και σε διαφορετικές μορφές, καθιστώντας δυσχερή την αξιοποίησή τους.

Σκοπός: Σκοπός της παρούσας διπλωματικής εργασίας είναι η αναπαραγωγή μίας Περίπτωσης Χρήσης και η εφαρμογή λογισμικού που θα υποστηρίζει την προγραμματιστική υλοποίηση της Αρχιτεκτονικής Lambda.

Υλικά – Μέθοδος: Η βιβλιογραφική ανασκόπηση της σχετικής ελληνικής και διεθνούς βιβλιογραφίας, μέσα από διαδικτυακές βάσεις δεδομένων, διενεργήθηκε στο χρονικό διάστημα από τον Οκτώβριο του 2020 έως και το Φεβρουάριο του 2021. Η Περίπτωση Χρήσης αφορά την ανάλυση πραγματικών δεδομένων από δύο Datasets: το πρώτο με δεδομένα τροχαίων ατυχημάτων στις Ηνωμένες Πολιτείες της Αμερικής (Η.Π.Α.) και το δεύτερο με καιρικά φαινόμενα στις Η.Π.Α., για το χρονικό διάστημα 2016 – 2019.

Αποτελέσματα: Τα αποτελέσματα της έρευνας οδήγησαν σε συσχετίσεις μεταξύ των τροχαίων ατυχημάτων και των επικρατούμενων καιρικών συνθηκών, τη δεδομένη χρονική στιγμή, αναδεικνύοντας το σκοπό της διπλωματικής εργασίας, δηλαδή να δοθεί αξία σε δεδομένα ευρείας κλίμακας, που, υπό άλλες συνθήκες, θα παρέμεναν ανεκμετάλλευτα.

Συμπεράσματα: Η τεχνολογία έχει δημιουργηθεί για να υπηρετεί τις ανάγκες του ανθρώπου και καλείται να επιτύχει τον παραπάνω σκοπό. Οι λύσεις λογισμικού και η ανάπτυξη νέων μοντέλων αρχιτεκτονικής μπορούν να εξυπηρετήσουν στην επωφελή αξιοποίηση των δεδομένων. Μοντέλα αρχιτεκτονικής όπως η Lambda, που επιλέχθηκε στα πλαίσια της παρούσας διπλωματικής, βοηθούν σημαντικά στη μείωση της ψαλίδας μεταξύ της παραγωγής δεδομένων και της τελικής αξιοποίησής τους.

Λέξεις Κλειδιά: Μεγάλα Δεδομένα, Αρχιτεκτονική Lambda, Διαδίκτυο των Πραγμάτων, Descriptive Analytics, Boxplot, Αλγόριθμος k-means, Clustering Γεωχωρικών Δεδομένων, Τροχαία ατυχήματα, Καιρικά Φαινόμενα, Ηνωμένες Πολιτείες Αμερικής (Η.Π.Α.).

Abstract

Introduction: "Big Data" are characterized by huge volume, variety, and constant accumulation of new data, from many sources and in different formats, making them difficult to be used.

Purpose: The purpose of this study is to reproduce a Use Case and apply a software that will support the programming implementation of Lambda Architecture.

Materials – Method: A review of relevant Greek and international literature, through online databases, was carried out from October 2020 to February 2021. The Use Case analyzes real data from two Datasets: the first with traffic accident data in the United States of America (U.S.A.), and the second with weather conditions in the U.S.A., from 2016 to 2019.

Results: The results of the survey led to correlations between road accidents and prevailing weather conditions, at a given time, highlighting the purpose of this study, which is to give value to large-scale data that, otherwise, would remain unexploited.

Conclusions: Technology has been created to serve human needs and is called to achieve this purpose. Software solutions and development of new architectural models can serve to benefit most of the data. Architectural models such as the Lambda, selected in the context of this research, significantly help to reduce the gap between data production and their final utilization.

Keywords: Big Data, Lambda Architecture, Internet of Things, Descriptive Analytics, Boxplot, k-means Algorithm, Geospatial Data Clustering, Traffic Accidents, Weather Conditions, United States of America (U.S.A.).

Πίνακας περιεχομένων

Κατάλογος Πινάκων	13
Κατάλογος εικόνων	14
1 Εισαγωγή	17
1.1 Το πρόβλημα και η ιδέα.....	17
1.2 Σκοπός της διπλωματικής.....	18
1.3 Οργάνωση κειμένου.....	19
2 Ορισμοί	21
2.1 Διαδίκτυο των Πραγμάτων (Internet of Things)	21
2.1.1 Ορισμός.....	21
2.1.2 Τρόπος λειτουργίας	22
2.1.3 Εφαρμογές.....	23
2.1.4 Οφέλη και πλεονεκτήματα.....	25
2.1.5 Επικοινωνία συσκευών	26
2.1.6 Προκλήσεις	30
2.2 Δεδομένα Μεγάλης Κλίμακας (Big Data)	38
2.2.1 Ορισμός.....	38
2.2.2 Εξέλιξη.....	38
2.2.3 Πεδίο Εφαρμογής.....	40
2.2.4 Χαρακτηριστικά των Δεδομένων Μεγάλης Κλίμακας.....	42
2.2.5 Αξία Δεδομένων Μεγάλης Κλίμακας.....	43
2.2.6 Πεδία εφαρμογής Δεδομένων Μεγάλης Κλίμακας.....	44
2.3 Lambda architecture	44
2.3.1 Δομή lambda architecture	45
2.3.2 Τρόπος λειτουργίας lambda architecture.....	49
2.3.3 Πλεονεκτήματα και Οφέλη lambda architecture	50
2.3.4 Προκλήσεις της αρχιτεκτονικής lambda.....	52
2.3.5 Karra architecture.....	52
2.3.6 Διαφορές μεταξύ αρχιτεκτονικών karra και lambda.....	53

2.4	Descriptive Analytics.....	53
2.4.1	Ορισμός.....	53
2.4.2	Boxplot.....	54
2.5	Clustering Spatial Data.....	56
2.5.1	Ορισμός Geospatial Data.....	56
2.5.2	Clustering Γεωχωρικών Δεδομένων.....	57
2.6	Αλγόριθμος k-means.....	60
2.6.1	Ορισμός αλγορίθμου k-means.....	60
2.6.2	Μειονεκτήματα – Αδυναμίες του αλγορίθμου k-means.....	61
3	Τεχνολογίες.....	63
3.1	FIWARE.....	63
3.1.1	Ιστορία.....	63
3.1.2	Ορισμός.....	64
3.1.3	Context Broker.....	64
3.1.4	IoT Agents.....	67
3.2	MongoDB.....	73
3.2.1	Ορισμός.....	73
3.2.2	Χαρακτηριστικά.....	74
3.2.3	Πλεονεκτήματα.....	74
3.2.4	Ροές Αλλαγών – Change Streams.....	74
3.3	SQL Server.....	75
3.3.1	Ορισμός SQL.....	75
3.3.2	Σχεσιακές βάσεις Δεδομένων.....	75
3.3.3	Microsoft SQL Server.....	77
3.4	Apache Superset.....	77
3.4.1	Προέλευση.....	77
3.4.2	Ορισμός.....	78
3.5	Docker.....	79
3.5.1	Containers.....	79
3.5.2	Ορισμός.....	80
3.5.3	Αρχείο docker-compose.yml.....	81

3.6	Προγραμματιστικές αρχές	81
3.6.1	Γλώσσα προγραμματισμού - C#	81
3.6.2	ASP.NET Core Framework.....	82
3.6.3	Ασύγχρονος προγραμματισμός	82
3.6.4	API.....	83
4	Αρχιτεκτονική – Υλοποίηση	85
4.1	Εγκατάσταση	85
4.2	Σχεδιάγραμμα αρχιτεκτονικής.....	86
4.2.1	Επίπεδο Προσομοίωσης (Simulation Layer)	87
4.2.2	Συγχρονισμός Δεδομένων	90
4.2.3	K-Means Algorithm.....	96
4.2.4	Απεικόνιση και χρήση δεδομένων	97
4.3	Σημαντική σημείωση	100
5	Περίπτωση Χρήσης (Use Case).....	101
5.1	Δεδομένα Dataset.....	101
5.1.1	Dataset Τροχαίων Ατυχημάτων	101
5.1.2	Dataset Καιρικών Συνθηκών.....	102
5.1.3	Συνδυασμός των δύο Dataset.....	102
5.2	Αξιοποίηση Δεδομένων	104
5.2.1	Πίνακας οργάνων Τριμήνου.....	104
5.2.2	Πίνακας οργάνων Ιστορικών Δεδομένων.....	111
5.3	Προεπισκόπηση Λειτουργιών (Video)	114
5.4	Σύγκριση – Συμπεράσματα.....	115
5.5	Οφέλη – Παρατηρήσεις	121
6	Επίλογος – Συμπεράσματα	122
7	Βιβλιογραφία	125
Παράρτημα	132	
	Κατάλογος συντομογραφιών	132
	Κατάλογος τμημάτων κώδικα	136

Κατάλογος Πινάκων

Πίνακας 1. Σύγκριση πρωτοκόλλων επικοινωνίας.....	30
Πίνακας 2. Πεδία εφαρμογής των Δεδομένων Μεγάλης Κλίμακας.....	44
Πίνακας 3. Αριθμητικά δεδομένα ενός boxplot	55
Πίνακας 4. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και του 2018, για την πολιτεία της Καλιφόρνια.....	115
Πίνακας 5. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις της πολιτείας της Καλιφόρνια.....	116
Πίνακας 6. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους της πολιτείας της Καλιφόρνια .	116
Πίνακας 7. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και του 2018, για την πολιτεία του Τέξας	117
Πίνακας 8. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις της πολιτείας του Τέξας	118
Πίνακας 9. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους της πολιτείας του Τέξας	118
Πίνακας 10. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πολιτείες της Καλιφόρνια και του Τέξας.....	119
Πίνακας 11. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις στις πολιτείες της Καλιφόρνια και του Τέξας ...	119
Πίνακας 12. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους στις πολιτείες της Καλιφόρνια και του Τέξας	120

Κατάλογος εικόνων

Εικόνα 1. Τοπολογία IoT	21
Εικόνα 2. Κύρια συστατικά στοιχεία του IoT	23
Εικόνα 3. Εφαρμογές του IoT	24
Εικόνα 4. Απεικόνιση κοινών προτύπων επικοινωνίας IoT	26
Εικόνα 5. Τεχνολογικές προκλήσεις του IOT	31
Εικόνα 6. Κίνδυνοι στην ασφάλεια των συσκευών IoT	31
Εικόνα 7. Κατηγορίες συνδεσιμότητας	32
Εικόνα 8. Ποσοστιαία αύξηση δεδομένων μεγάλης κλίμακας, ανά έτος.....	38
Εικόνα 9. Απεικόνιση της αποθήκευσης, ανάλυσης και επεξεργασίας της πληροφορίας ...	39
Εικόνα 10. Οι κύριες πηγές δεδομένων μεγάλης κλίμακας και η ταχεία ανάπτυξή τους	40
Εικόνα 11. Η εξέλιξη των big data	42
Εικόνα 12. Τα τέσσερα V των Δεδομένων Μεγάλης Κλίμακας	43
Εικόνα 13. Αξία της οικονομίας δεδομένων στην Ε.Ε.....	43
Εικόνα 14. Πεδία εφαρμογής των Δεδομένων Μεγάλης Κλίμακας.....	44
Εικόνα 15. Παράδειγμα συστήματος που χρησιμοποιεί Lambda Architecture	45
Εικόνα 16. Σχηματική απεικόνιση της αρχιτεκτονικής Lambda.....	46
Εικόνα 17. Τα επίπεδα της αρχιτεκτονικής lambda	48
Εικόνα 18. Τα δεδομένα ταξινομούνται σε ευρετήρια ταυτόχρονα τόσο από το serving επίπεδο, όσο και από το speed επίπεδο	49
Εικόνα 19. Όταν το επίπεδο εξυπηρέτησης ολοκληρώσει μία εργασία, μετακινείται στο επόμενο επίπεδο και το επίπεδο ταχύτητας διαγράφει το αντίγραφο των δεδομένων που μόλις προστέθηκαν στο επίπεδο εξυπηρέτησης.....	50
Εικόνα 20. Δομή της αρχιτεκτονικής kappa.....	52
Εικόνα 21. Κατηγορίες Big Data Analytics (Περιγραφική, Διαγνωστική, Προγνωστική & Καθοδηγητική Αναλυτική).....	54
Εικόνα 22. Σχηματική απεικόνιση ενός boxplot	55
Εικόνα 23. Ασυμμετρία σε boxplot.....	56
Εικόνα 24. Βήματα αλγορίθμου k – means.....	61
Εικόνα 25. Συνάρτηση τετραγωνικού λάθους.....	61
Εικόνα 26. Απεικόνιση των components FIWARE	64

Εικόνα 27. Τα μέλη σε ένα σενάριο επικοινωνίας για ανάκτηση δεδομένων κατάστασης .	65
Εικόνα 28. Παράδειγμα επικοινωνίας μέσω της βόρειας θύρας του IoT Agent	69
Εικόνα 29. Παράδειγμα επικοινωνίας μέσω της νότιας θύρας του IoT Agent	70
Εικόνα 30. Οπτική απεικόνιση ενός πλήρους οικοσυστήματος βασισμένο σε FIWARE....	72
Εικόνα 31. Παράδειγμα BSON αντικειμένου στη βάση MongoDB.....	73
Εικόνα 32. Απεικόνιση μίας σχέσης σε σχεσιακή βάση δεδομένων.....	76
Εικόνα 33. Απεικόνιση τύπων ενώσεων (JOIN) σε σχεσιακή βάση δεδομένων	76
Εικόνα 34. Απεικόνιση διαφορών Virtualization vs Containerization.....	79
Εικόνα 35. Παράδειγμα επανάκλησης.....	83
Εικόνα 36. Αρχιτεκτονική της εφαρμογής.....	86
Εικόνα 37. Απεικόνιση δεδομένων οντότητας συλλογής IoT Agent	91
Εικόνα 38. Απεικόνιση δεδομένων οντότητας συλλογής Orion Context Broker.....	91
Εικόνα 39. Προσθήκη βάσης δεδομένων στο Superset.....	98
Εικόνα 40. Προσθήκη πίνακα για ανάλυση στο Superset.....	99
Εικόνα 41. Αποτύπωση Πίνακα Οργάνων για την Καλιφόρνια, το πρώτο τρίμηνο του 2017	104
Εικόνα 42. Φίλτρο για περαιτέρω εκλέπτυνση των δεδομένων	105
Εικόνα 43. Ατυχήματα της πολιτείας της Καλιφόρνια, το 2017	105
Εικόνα 44. Διάγραμμα Area για αποτύπωση ημερησίων ατυχημάτων	106
Εικόνα 45. Διάγραμμα Nightingale Rose για οπτικοποίηση εβδομαδιαίων ατυχημάτων..	106
Εικόνα 46. Πόλεις με το μεγαλύτερο αριθμό ατυχημάτων	107
Εικόνα 47. Δρόμοι με τα περισσότερα ατυχήματα.....	107
Εικόνα 48. Διάγραμμα Boxplot για αποτύπωση ατυχημάτων.....	108
Εικόνα 49. Αριθμητικά δεδομένα boxplot.....	108
Εικόνα 50. Χάρτης ατυχημάτων.....	109
Εικόνα 51. Από πάνω αριστερά προς κάτω δεξιά, 10, 15, 20, 30 centroids αντίστοιχα	110
Εικόνα 52. Ενδεικτική τοποθέτηση ασθενοφόρων.....	111
Εικόνα 53. Αναζήτηση ιστορικών δεδομένων	112
Εικόνα 54. Αποτελέσματα ιστορικής αναζήτησης.....	113
Εικόνα 55. Αναζήτηση ασθενοφόρου με τοποθεσία ατυχήματος.....	114
Εικόνα 56. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017, για την πολιτεία της Καλιφόρνια, σε εβδομαδιαία βάση	115

Εικόνα 57. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και 2018 για την πολιτεία του Τέξας, σε εβδομαδιαία βάση..... 117

1

Εισαγωγή

1.1 Το πρόβλημα και η ιδέα

Το Διαδίκτυο αποτελεί, εδώ και δεκαετίες, τη μεγαλύτερη πηγή δεδομένων στον κόσμο. Η αυξανόμενη υιοθέτησή του, καθώς και η ευκολότερη προσβασιμότητα σε ανθρώπους, μέσω προσιτών τεχνολογιών (έξυπνα τηλέφωνα, tablet), δίνει νέα όρια στα πλαίσια λειτουργίας του. Οι άνθρωποι χρησιμοποιούν το διαδίκτυο, πλέον, ως αναπόσπαστο κομμάτι της καθημερινότητάς τους: από αναζήτηση πληροφοριών, ανταλλαγή και επεξεργασία δεδομένων, μέχρι επικοινωνία μεταξύ ανθρώπων, ακόμα και στις πιο αντίξοες συνθήκες ή και στις πιο απομακρυσμένες περιοχές. Αυτό έχει οδηγήσει σε μία ραγδαία αύξηση όγκου δεδομένων και ο ρυθμός παραγωγής πληροφορίας έχει ξεφύγει από τα όρια του διαχειρίσιμου.

Σε αυτό το δυσμενές – από πλευράς όγκου δεδομένων – περιβάλλον, έρχεται να συνεισφέρει αρνητικά η άνοδος της τεχνολογίας. Η αρχή έγινε με τη δημιουργία μίας νέας τάξης πραγμάτων με την έλευση των έξυπνων τηλεφώνων. Δόθηκε η δυνατότητα σε ανθρώπους να έχουν πρόσβαση στο διαδίκτυο, καθώς και σε σεβαστά – για εκείνη την εποχή – επίπεδα υπολογιστικής ισχύος. Σήμερα, τα έξυπνα τηλέφωνα έχουν πολλαπλάσια υπολογιστική ισχύ, συγκριτικά με εκείνη την εποχή, η οποία, μάλιστα, είναι συγκρίσιμη με αυτή των σύγχρονων υπολογιστών, και έχουν βρει το δρόμο για τα χέρια πολλών ανθρώπων, από άτομα πολύ μικρής έως πολύ μεγάλης ηλικίας.

Σε αυτό το εύφορο έδαφος, γεννήθηκε η ιδέα του όρου «Διαδίκτυο των Πραγμάτων», καθώς και η εξάπλωσή του. Ξεκίνησε από μία πρωτοπόρα ιδέα και έχει καταλήξει σε εφαρμόσιμη αρχιτεκτονική συστημάτων. Η ιδέα, ουσιαστικά, ήταν η δυνατότητα επικοινωνίας μεταξύ ετερογενών πόρων, οι οποίοι, φέροντας ενσωματωμένες αισθητήριες συσκευές ικανοποιητικής επεξεργαστικής δύναμης, έχουν τη δυνατότητα να ανταλλάσσουν και να αξιοποιούν πληροφορίες που συλλέγονται μέσα από διαδικασίες παρατήρησης. Στόχος αυτής της

επικοινωνίας είναι να δοθεί και να εξαπλωθεί, σε όλες τις συσκευές, η ψευδαίσθηση της ευφυΐας, μέσω της αυτόνομης κατασκευής και χρήσης πληροφοριακών δικτύων από «έξυπνες» συσκευές, για τη δημιουργία και λήψη ανεξάρτητων αποφάσεων.

Όπως είναι εύκολα αντιληπτό, η απότομη ανάπτυξη και υιοθέτηση των παραπάνω τεχνολογιών έχει οδηγήσει σε μία άναρχη και ανεκμετάλλευτη παραγωγή δεδομένων. Το πλήθος των έξυπνων συσκευών, ο όγκος και η συχνότητα των δεδομένων τα κατατάσσει, πλέον, ως «Δεδομένα Μεγάλης Κλίμακας». Ο συνήθης τρόπος επεξεργασίας τόσο μεγάλων δεδομένων είναι μέσω ανάλυσής τους σε μεταγενέστερο χρόνο και ανά συγκεκριμένες χρονικές περιόδους. Αυτό γίνεται για δύο κυρίως λόγους: πρώτον, επειδή απαιτείται αρκετή επεξεργαστική ισχύς για την ανάλυσή τους, και δεύτερον, διότι η μορφή των δεδομένων δεν είναι σταθερή μεταξύ όλων των συσκευών, συνήθως απαιτείται μετασχηματισμός των δεδομένων σε μία κοινή βάση. Λύση στο παραπάνω πρόβλημα έρχεται να δώσει η Αρχιτεκτονική Lambda. Ο συγκεκριμένος τρόπος ανάπτυξης συστημάτων λογισμικού δίνει τη δυνατότητα για απρόσκοπτη λειτουργία και ενσωμάτωση νέων δεδομένων. Τα επίπεδα (layers) που την απαρτίζουν επιτρέπουν σε όλα τα στάδια παραγωγικής αξιοποίησης να λειτουργούν κανονικά. Τα νέα δεδομένα αποθηκεύονται, τα παλιά δεδομένα δύνανται να υποστούν επεξεργασία και ανάλυση, καθώς επίσης υποστηρίζεται και η αναζήτηση ιστορικών δεδομένων του συστήματος.

1.2 Σκοπός της διπλωματικής

Έχοντας αναφέρει συνοπτικά το πρόβλημα που παρουσιάζεται από το πλήθος και την ποικιλομορφία των δεδομένων μέσω του Διαδικτύου των Πραγμάτων, σκοπός της παρούσας διπλωματικής εργασίας είναι η αναπαραγωγή μίας Περίπτωσης Χρήσης και η υλοποίηση λογισμικού που θα υποστηρίζει την προγραμματιστική υλοποίηση της Αρχιτεκτονικής Lambda.

Θα δημιουργηθεί ένα επίπεδο προσομοίωσης (Simulation Layer), όπου πραγματικά δεδομένα – που έχουν αντληθεί από επιστημονικά Datasets – θα τροφοδοτούνται σε εικονικές έξυπνες συσκευές που είναι συνδεδεμένες στο Διαδίκτυο των Πραγμάτων (IoT Agents). Αυτές, με τη σειρά τους, θα αποθηκεύουν την πληροφορία που τους αποστέλλεται σε μία κεντρική βάση δεδομένων. Το πλήθος των δεδομένων που αποστέλλονται είναι πολύ υψηλό, καθιστώντας τα δεδομένα που συλλέγονται στη Βάση Δεδομένων ως Δεδομένα Μεγάλης Κλίμακας.

Στη συνέχεια, αυτά τα δεδομένα μετασχηματίζονται σε μία περισσότερο φιλική μορφή, για ανάλυση δεδομένων, και ενσωματώνονται σε σύστημα Επιχειρηματικής Ευφυΐας (Business Intelligence ή απλά BI). Αυτό θα δώσει τη δυνατότητα στους χρήστες για απρόσκοπτη ανάλυση, αναζήτηση και επεξεργασία δεδομένων.

Επιγραμματικά, θα αναφερθεί η Περίπτωση Χρήσης που έχει υλοποιηθεί στα πλαίσια της παρούσας διπλωματικής εργασίας. Το Κεφάλαιο 5 παρέχει περισσότερο λεπτομερή ανάλυση. Μέσω δύο Datasets, ένα Dataset με δεδομένα καιρικών συνθηκών και άλλο ένα με δεδομένα τροχαίων ατυχημάτων στις Ηνωμένες Πολιτείες της Αμερικής (Η.Π.Α.), θα τροφοδοτηθούν σε 40 εικονικούς αισθητήρες. Αυτοί θα αποθηκεύσουν την πληροφορία σε μία κεντρική βάση δεδομένων. Με προγραμματιστικά components που υλοποιήθηκαν, αυτά τα δεδομένα συγχρονίζονται ασύγχρονα με ένα σύστημα Business Intelligence. Εκεί, βάσει ημερολογιακών κριτηρίων καθώς και γεωγραφικής εγγύτητας, συσχετίζεται ένα δεδομένο ατυχήματος με ένα δεδομένο καιρού, εμπλουτίζοντας έτσι την πληροφορία που υπάρχει για το κάθε ατύχημα.

Στο σύστημα Business Intelligence, έχουν υλοποιηθεί διεπαφές για τους χρήστες που παρουσιάζουν την πληροφορία με ένα απλό και κατανοητό τρόπο. Παρουσιάζονται χάρτες, heatmaps, αναλύσεις, διαγράμματα και ιστορικά δεδομένα. Δίδεται, ωστόσο, η δυνατότητα σε οποιονδήποτε να χρησιμοποιήσει queries για να αναζητήσει οποιαδήποτε ιστορικά δεδομένα, καθώς και να τα παρουσιάσει με όποιον τρόπο επιθυμεί.

Σημαντικό να αναφερθεί είναι πως, μέσω του ασύγχρονου συγχρονισμού δεδομένων, το σύστημα τροφοδοτείται συνεχόμενα με δεδομένα και πάντα θα προσφέρει την πλήρη εικόνα τους, και όχι ένα μέρος του, αποτελώντας χαρακτηριστικό παράδειγμα ορθής χρήσης της Αρχιτεκτονικής Lambda.

1.3 Οργάνωση κειμένου

Η εργασία έχει χωριστεί σε 7 κεφάλαια.

Το τρέχον κεφάλαιο (Κεφάλαιο 1) είναι εισαγωγικό. Εξηγεί το πρόβλημα που εξετάζει η διπλωματική και τους στόχους στους οποίους αποσκοπεί.

Στο Κεφάλαιο 2, δίνονται οι ορισμοί των τριών σημαντικότερων εννοιών της εργασίας: το Διαδίκτυο των Πραγμάτων (Internet of Things), των Δεδομένων Μεγάλης Κλίμακας (Big Data), καθώς και της Αρχιτεκτονικής Lambda (Lambda Architecture). Γίνεται ιστορική αναδρομή της πορείας του Διαδικτύου των Πραγμάτων, καθώς και των Δεδομένων Μεγάλης κλίμακας, εξετάζονται τα χαρακτηριστικά τους και το πεδίο εφαρμογής τους. Όσον αφορά την Αρχιτεκτονική Lambda, περιγράφονται ο τρόπος λειτουργίας της, τα πλεονεκτήματα και τα οφέλη της.

Στο Κεφάλαιο 3, δίδονται οι ορισμοί και η χρησιμότητα όλων των τεχνολογιών που αξιοποιούνται από την παρούσα εργασία.

Το Κεφάλαιο 4 αναφέρεται στην την αρχιτεκτονική δομή της υλοποίησης: αρχίζει περιγράφοντας τον τρόπο εγκατάστασης της εφαρμογής, προχωρά σε ανάλυση των μοντέλων

δεδομένων – παρέχοντας διάγραμμα της αρχιτεκτονικής, και επεξηγεί όλα τα components που αναπτύχθηκαν για να εξυπηρετήσουν τις ανάγκες.

Στο Κεφάλαιο 5, όπως προαναφέρθηκε, παρουσιάζεται η Περίπτωση Χρήσης: ένα σενάριο με πραγματικά δεδομένα που προβάλλει, με τον καλύτερο τρόπο, το σκοπό της εργασίας. Περιγράφει τις λειτουργικές απαιτήσεις του συστήματος καθώς και τη χρησιμότητά του, και παρουσιάζει τα αποτελέσματα της έρευνας στο συγκεκριμένο σενάριο.

Στο Κεφάλαιο 6, παρουσιάζονται τα Συμπεράσματα που μπορέσαμε να εξάγουμε από την παρούσα εργασία.

Τέλος, το Κεφάλαιο 7 περιλαμβάνει τις βιβλιογραφικές πηγές που αξιοποιήθηκαν κατά τη συγγραφή της εργασίας.

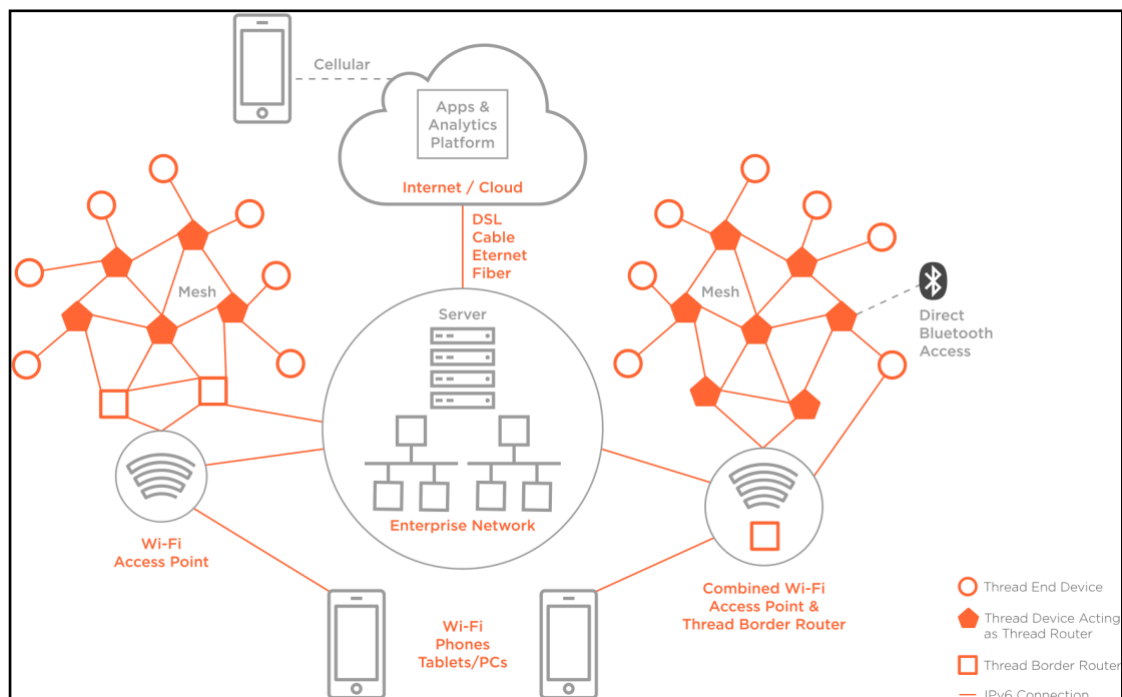
2

Ορισμοί

2.1 Διαδίκτυο των Πραγμάτων (Internet of Things)

2.1.1 Ορισμός

Το Διαδίκτυο των Πραγμάτων (Internet of Things – IoT) είναι μία νέα διασύνδεση της τεχνολογίας και αποτελεί μία επέκταση των συνδέσεων που ήδη υπάρχουν μεταξύ ατόμων και υπολογιστών, ώστε να περιλαμβάνει ψηφιακά συνδεδεμένα «πράγματα», τα οποία μετρούν και αναφέρουν δεδομένα. Αυτά τα δεδομένα μπορεί να είναι απλοί αριθμοί από ένα σταθερό ή κινητό αισθητήρα ή πιο περίπλοκα ευρήματα από συσκευές που μετρούν και αναφέρουν πολλές ροές δεδομένων ταυτόχρονα.



Εικόνα 1. Τοπολογία ΙοΤ. [1]

Με άλλα λόγια, το Διαδίκτυο των Πραγμάτων (ΔτΠ) είναι ένα ευρύ δίκτυο επικοινωνίας που περιλαμβάνει μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα στο διαδίκτυο. Πρωταρχικός του στόχος είναι η σύνδεση και η ανταλλαγή δεδομένων μεταξύ όλων των προαναφερθέντων μελών. Υπάρχουν πολλοί διαθέσιμοι ορισμοί για το IoT, ωστόσο το κοινό χαρακτηριστικό τους αποτελεί η διασύνδεση καθημερινών συσκευών, στις οποίες προστίθεται κατάλληλα διαμορφωμένο υλικό, με σκοπό να μπορούν να συλλέγουν πληροφορίες, να τις ανταλλάσσουν και να τις επεξεργάζονται. [2]

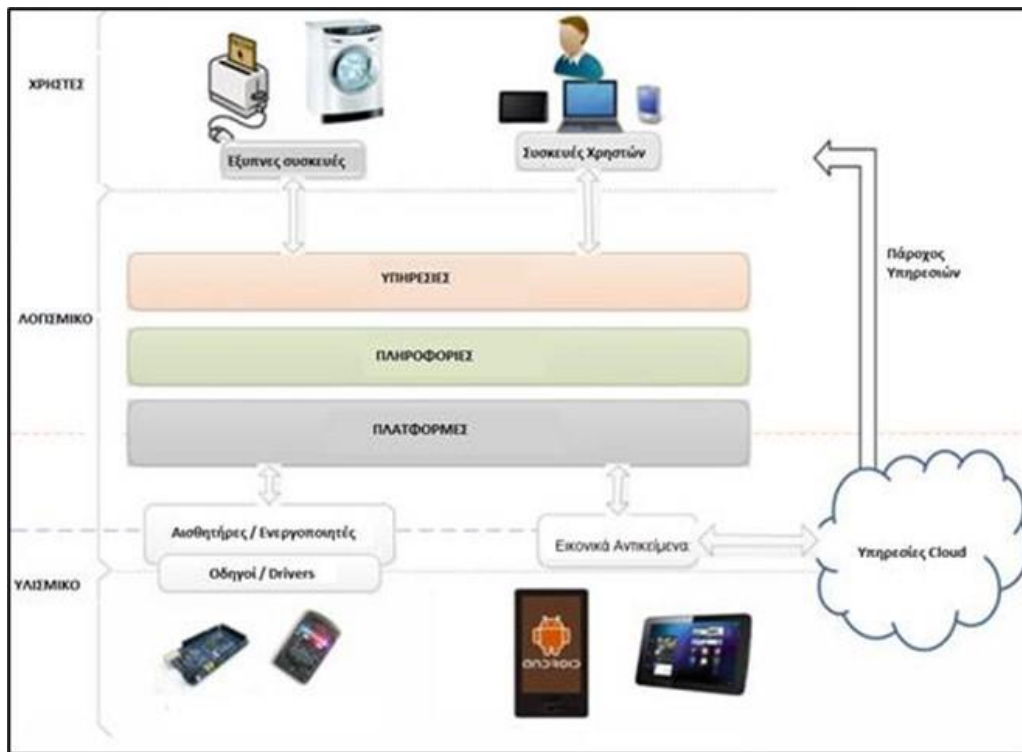
Με απλά λόγια, το IoT αναφέρεται σε μία εικονική σύνδεση στο Διαδίκτυο από πράγματα, διαδικασίες, ανθρώπους, ζώα και σχεδόν όλα όσα βλέπουμε, τα οποία μπορούν να επηρεάσουν το φυσικό περιβάλλον. Περιγράφει μία κατάσταση όπου τα πάντα στο περιβάλλον μας είναι ικανά να επικοινωνούν αυτόματα μεταξύ τους, χωρίς να χρειάζεται αλληλεπίδραση μεταξύ ανθρώπων ή ανθρώπου με μηχανή. Δηλαδή δεν απαιτείται διαμεσολάβηση του ανθρώπινου παράγοντα. Εκτός από το γεγονός ότι είναι μία πρωτοποριακή ανακάλυψη, μπορεί επίσης να αποδειχτεί εξαιρετικά ωφέλιμη για τη διευκόλυνση της ζωής μας σε πολλαπλές εφαρμογές. [3]

Χαρακτηριστικά παραδείγματα εφαρμογών του IoT, που διευκολύνουν την καθημερινότητα, αποτελούν οι αυτοματισμοί σπιτιών, καθώς και διάφορες πλατφόρμες που επιτρέπουν λειτουργίες όπως η πρόσβαση στις συσκευές, η διασφάλιση της σωστής εγκατάστασης και λειτουργίας της συσκευής, η διασύνδεση διάφορων συσκευών σε τοπικά δίκτυα, το cloud ή με άλλες συσκευές. [4]

2.1.2 Τρόπος λειτουργίας

Οι δυνατότητες του Διαδικτύου των πραγμάτων είναι απεριόριστες και τα βασικά στοιχεία που απαιτούνται για ένα λειτουργικό σύστημα IoT αναφέρονται ακολούθως. (*Εικόνα 2*)

1. **Αισθητήρες και συσκευές** με δυνατότητες για συλλογή δεδομένων από το περιβάλλον, αποθήκευση, λήψη, μεταβίβαση και μετατροπή των δεδομένων σε ηλεκτρικά σήματα. Ορισμένα παραδείγματα αισθητήρων είναι οι αισθητήρες υγρασίας, θερμοκρασίας αέρα, ραδιενέργειας και κίνησης.
2. **Συνδεσιμότητα**, καθώς τα δεδομένα που συλλέγονται κοινοποιούνται και σε άλλες συσκευές με ευκολία μέσω του Διαδικτύου. Υπό φυσιολογικές συνθήκες, οι αισθητήρες και οι συσκευές IoT επικοινωνούν με εφαρμογές και υπηρεσίες που εκτελούνται μέσω κάποιου δικτύου στο cloud.
3. **Επεξεργασία δεδομένων**. Για να είναι πραγματικά αποτελεσματικό και πολύτιμο το σύστημα Internet of Things, οι τεράστιοι όγκοι δεδομένων που δημιουργούνται απαιτούν εξελιγμένο λογισμικό ανάλυσης, ώστε μόλις τα δεδομένα μεταβιβαστούν από τη συσκευή στο cloud, το λογισμικό να μπορεί να ξεκινήσει την ανάλυσή τους. [4]



Εικόνα 2. Κύρια συστατικά στοιχεία του IoT. [4]

Μία επιπλέον συσκευή που είναι απαραίτητη για την υλοποίηση του IoT είναι ο ενεργοποιητής (actuator). Αποτελεί έναν κινητήρα, ο οποίος χρησιμοποιείται για να μετακινήσει ή να ελέγξει έναν άλλο μηχανισμό ή σύστημα, βάσει ορισμένων οδηγιών. Υπάρχουν τρεις τύποι ενεργοποιητών στο IoT:

1. Υδραυλικός: χρησιμοποιεί την πίεση υγρών για να εκτελέσει μηχανική κίνηση
2. Pneumatic: χρησιμοποιεί αέρα σε υψηλή πίεση για να επιτρέψει τη μηχανική λειτουργία και
3. Ηλεκτρικός: τροφοδοτείται από ένα μοτέρ και μετατρέπει την ηλεκτρική ενέργεια σε μηχανική λειτουργία. [4]

2.1.3 Εφαρμογές

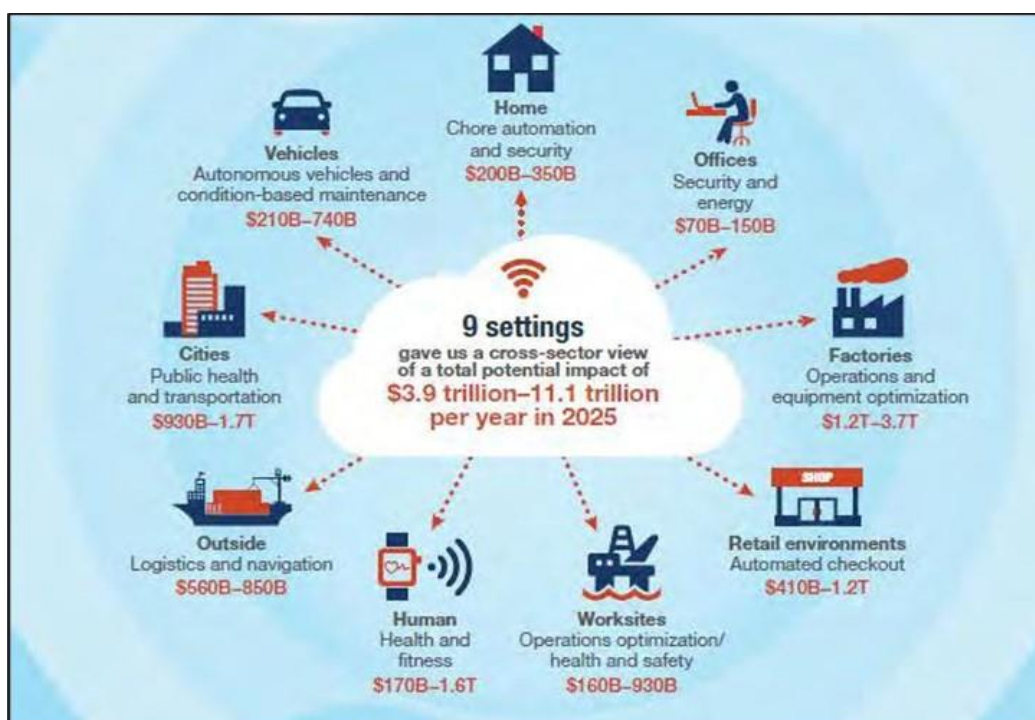
Πλέον, οι εφαρμογές του IoT είναι πάρα πολλές, με την κυριότερη να αφορά οικιακούς αυτοματισμούς. Δίνεται η δυνατότητα στο χρήστη να ελέγχει το φωτισμό, τη θερμοκρασία, τα πολυμέσα και τα συστήματα ασφαλείας ενός σπιτιού.

Ωστόσο, παρατηρείται και μεγάλο εύρος εφαρμογής στον τομέα της υγείας. Προσφέρεται η δυνατότητα στο ιατρονοσηλευτικό προσωπικό, μέσω διασύνδεσης πολλών συσκευών, να έχει πρόσβαση στο ιατρικό ιστορικό και τις εργαστηριακές και απεικονιστικές εξετάσεις των νοσηλευομένων. Μία δεύτερη, και σημαντική, εφαρμογή αποτελούν οι βηματοδότες. Δίνεται

η δυνατότητα στο ιατρικό προσωπικό να έχει άμεση πρόσβαση σε ζωτικές πληροφορίες του ασθενούς. Με αυτόν τον τρόπο, διευκολύνεται η παρακολούθηση των ασθενών, χωρίς να είναι απαραίτητη η συνεχής ανθρώπινη παρουσία, γεγονός ιδιαίτερα σημαντικό για ασθενείς που κατοικούν σε απομακρυσμένες περιοχές χωρίς δυνατότητα εύκολης και συχνής πρόσβασης σε υπηρεσίες υγείας. Παράλληλα, μπορούν να εφαρμοστούν απομακρυσμένα θεραπευτικές παρεμβάσεις, μέσω τροποποίησης των ρυθμίσεων και ενεργοποίησης των συστημάτων ασφαλείας.

Η τεχνολογία του IoT συναντάται ευρέως και στον τομέα των μεταφορών και της αυτοκίνησης. Τα σύγχρονα αυτοκίνητα διαθέτουν αισθητήρες GPS, υγρασίας, θερμοκρασίας, ταχύτητας. Με αυτόν τον τρόπο, οι οδηγοί μπορούν να ενημερώνονται για τις καιρικές συνθήκες, την κυκλοφορία στους δρόμους και την κατάσταση του οχήματός τους. Επίσης, στα μέσα μαζικής μεταφοράς (M.M.M.), εφαρμόζονται συστήματα τηλεματικής. Τα συστήματα αυτά παρέχουν στους χρήστες πληροφορίες για τα ακριβή δρομολόγια, την τοποθεσία των M.M.M., καθώς και τον εκτιμώμενο χρόνο άφιξης του μέσου στη στάση. [2]

Τέλος, σημαντικό ρόλο διαδραματίζει και στον τεχνολογικό τομέα και τον τομέα της μαζικής παραγωγής. Η διαρκής συλλογή πληροφοριών από πολλαπλούς αισθητήρες, η επεξεργασία και διανομή αυτών των δεδομένων στο διαδίκτυο, σε συνδυασμό με τη δυνατότητα απομακρυσμένου ελέγχου, δίνει τη δυνατότητα για έλεγχο πάρκων εναλλακτικής ενέργειας (αιολικών, ηλιακών), επίβλεψη αγροτικών καλλιεργειών, βελτιστοποίηση της ενεργειακής απόδοσης εργοστασίων και επιχειρήσεων, καθώς και αυτοματοποίηση στην παρακολούθηση συνθηκών και αποθεμάτων. [4] (Εικόνα 3)



Εικόνα 3. Εφαρμογές του IoT. [5]

2.1.4 Οφέλη και πλεονεκτήματα

Το Διαδίκτυο των Πραγμάτων (IoT) αποτελεί νέα, συναρπαστική τεχνολογία και εκπροσωπεί την τέταρτη γενιά του διαδικτύου. Τα κυριότερα πλεονεκτήματα της είναι τα εξής:

1. Αποτελεσματικότητα και εξοικονόμηση χρόνου

Το IoT επιτρέπει την επικοινωνία μεταξύ συσκευών η οποία ονομάζεται Machine-to-Machine (M2M). Οι συσκευές είναι σε θέση να επικοινωνούν με τους ανθρώπους, ενημερώνοντάς τους για την κατάσταση και τη θέση τους. Η αλληλεπίδραση μεταξύ μηχανών και ανθρώπου δίνει ακριβή αποτελέσματα, εξασφαλίζει καλύτερη απόδοση και εξοικονόμηση πολύτιμου χρόνου.

2. Αυτοματισμός και έλεγχος

Λόγω των φυσικών αντικειμένων που συνδέονται και ελέγχονται ψηφιακά και κεντρικά με ασύρματη υποδομή, υπάρχει μεγάλος αριθμός αυτοματισμών και ελέγχου στη λειτουργία. Χωρίς ανθρώπινη παρέμβαση, οι μηχανές είναι σε θέση να επικοινωνούν μεταξύ τους, ταχύτερα και αποτελεσματικότερα.

3. Υποβοήθηση στη λήψη αποφάσεων

Η έγκαιρη και γρήγορη συγκέντρωση μεγάλου όγκου πληροφοριών βοηθά στη λήψη καλύτερων αποφάσεων.

4. Παρακολούθηση

Ένα από τα προφανέστερα πλεονεκτήματα του IoT είναι η παρακολούθηση. Καίριας σημασίας καθημερινές εργασίες απλουστεύονται μέσω της παρακολούθησης. Η πρόβλεψη για έγκαιρη προμήθεια αγαθών, η ενοποίηση πολλαπλών αγορών, η παρακολούθηση της ημερομηνίας λήξης τους είναι απόρροια της ενσωμάτωσης της τεχνολογίας στην καθημερινότητα. Παράλληλα, όταν το IoT εφαρμοστεί σε υπηρεσίες, η αυτοματοποίηση επιτρέπει ευκολότερα και αποτελεσματικότερα τον έλεγχο καθημερινών εργασιών, τη διατήρηση της διαφάνειας, την ομοιομορφία στα καθήκοντα και τη διατήρηση της ποιότητας των υπηρεσιών.

5. Εξοικονόμηση πόρων

Το μεγαλύτερο πλεονέκτημα του IoT είναι η εξοικονόμηση χρημάτων και πόρων. Το IoT αποδεικνύεται πολύ χρήσιμο στην καθημερινότητα, κάνοντας τις συσκευές να επικοινωνούν μεταξύ τους, με αποτελεσματικό τρόπο, εξοικονομώντας ενέργεια και κόστος.

6. Καλύτερη ποιότητα ζωής

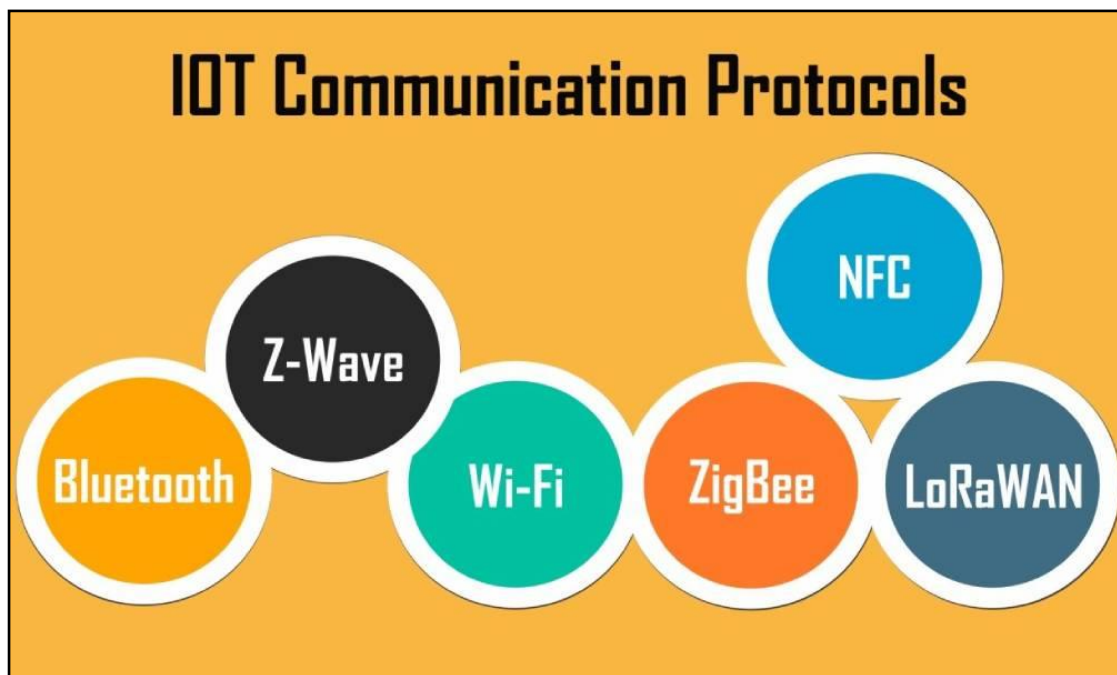
Όλες οι εφαρμογές αυτής της τεχνολογίας καταλήγουν σε αυξημένη άνεση, ευκολία και καλύτερη διαχείριση, βελτιώνοντας ουσιαστικά την ποιότητα ζωής. [5]

2.1.5 Επικοινωνία συσκευών

Για να καταστεί δυνατή η επικοινωνία μεταξύ των συσκευών IoT, αλλά και η αλληλεπίδρασή τους με τον έξω κόσμο, χρησιμοποιούνται δύο βάσεις επικοινωνίας. Το πρώτο επίπεδο αφορά τη χρήση ενός προτύπου συνδεσιμότητας με το δίκτυο και, σε δεύτερη φάση, η επικοινωνία λαμβάνει χώρα με τη χρήση ενός πρωτοκόλλου.

2.1.5.1 Πρότυπα συνδεσιμότητας

Η συνδεσιμότητα είναι το κυριότερο χαρακτηριστικό μίας συσκευής του Διαδικτύου των Πραγμάτων. Κρίνεται απαραίτητη για να καταστεί δυνατή η επικοινωνία μεταξύ των υπόλοιπων συσκευών, αλλά και η λειτουργικότητα της συσκευής.



Εικόνα 4. Απεικόνιση κοινών προτύπων επικοινωνίας IoT. [6]

Υπάρχει πληθώρα επιλογών για τη συνδεσιμότητα των συσκευών με το δίκτυο (*Εικόνα 4*), με τα κυριότερα από αυτά να αναλύονται παρακάτω.

2.1.5.1.1 Bluetooth

Το **Bluetooth** είναι ένα πρότυπο μικρής εμβέλειας, το οποίο έχει αποδεικτεί πολύ σημαντικό στην πληροφορική, αλλά και σε άλλες κατηγορίες καταναλωτικών προϊόντων. Αναμένεται να διαδραματίσει ακόμα σημαντικότερο ρόλο σε προϊόντα που φοριούνται (wearables), όπως τα έξυπνα ρολόγια. Το νέο πρότυπο Bluetooth Low – Energy (χαμηλής ενέργειας) – το οποίο ονομάζεται και BLE ή Bluetooth Smart – είναι μία σημαντική προσθήκη, γιατί προσφέρει την ίδια εμβέλεια με το απλό Bluetooth, αλλά με σημαντικά χαμηλότερη κατανάλωση ενέργειας.

2.1.5.1.2 Zigbee

Το **ZigBee** είναι παρόμοιο με το Bluetooth και χρησιμοποιείται κυρίως σε βιομηχανικές εγκαταστάσεις. Έχει ορισμένα σημαντικά πλεονεκτήματα σε πολύπλοκα συστήματα, καθώς προσφέρει λειτουργία χαμηλής ισχύος, υψηλή ασφάλεια, υψηλή αντοχή και είναι σε θέση να εκμεταλλευτεί τα ασύρματα δίκτυα ελέγχου και αισθητήρων σε εφαρμογές IoT. Η τελευταία έκδοση του ZigBee, που ονομάζεται 3.0, κυκλοφόρησε πρόσφατα και ενοποιεί, ουσιαστικά, τα διάφορα ασύρματα πρότυπα ZigBee που υπήρχαν, σε ένα ενιαίο πρότυπο.

2.1.5.1.3 Z – Wave

Το **Z – Wave** είναι ένα πρότυπο IoT επικοινωνιών ραδιοσυχνότητας (Radio Frequency ή RF) χαμηλής ισχύος που σχεδιάστηκε κυρίως για οικιακό αυτοματισμό, δηλαδή για προϊόντα όπως λαμπτήρες και αισθητήρες. Το Z – Wave χρησιμοποιεί ένα απλούστερο πρωτόκολλο από τα υπόλοιπα πρότυπα, το οποίο προσφέρει ταχύτερη και απλούστερη ανάπτυξη. Ωστόσο, το μεγαλύτερο μειονέκτημα του προτύπου είναι πως αποτελεί ιδιοκτησία της Sigma Designs, η οποία διατηρεί και τα δικαιώματα για την παραγωγή ολοκληρωμένων κυκλωμάτων (integrated chip), σε αντίθεση με άλλα πρότυπα που είναι ευρέως διαθέσιμα για παραγωγή.

2.1.5.1.4 WiFi

Η συνδεσιμότητα μέσω **WiFi** (Wireless fidelity) είναι ένα από τα περισσότερο δημοφιλή πρότυπα επικοινωνίας IoT και συχνά αποτελεί την προφανέστερη επιλογή, δεδομένης της διαθεσιμότητας ενός οικιακού περιβάλλοντος LAN. Χάρη στην ευρεία υπάρχουσα υποδομή, προσφέρεται για γρήγορη μεταφορά δεδομένων και δυνατότητα χειρισμού μεγάλων ποσοτήτων δεδομένων. Προς το παρόν, το πιο κοινό πρότυπο WiFi, που χρησιμοποιείται σε οικείες και πολλές επιχειρήσεις, είναι το 802.11n, το οποίο προσφέρει εύρος εκατοντάδων megabit ανά δευτερόλεπτο, το οποίο είναι ικανοποιητικό για τη μεταφορά αρχείων, αλλά μπορεί να είναι πολύ απαιτητικό, από πλευράς κατανάλωσης ενέργειας, για πολλές συσκευές IoT.

2.1.5.1.5 Δίκτυα κινητής τηλεφωνίας (Cellular)

Οποιαδήποτε συσκευή IoT, που απαιτεί λειτουργία σε μεγαλύτερες αποστάσεις, μπορεί να εκμεταλλευτεί τις δυνατότητες **κινητής επικοινωνίας** GSM / 3G / 4G. Ενώ το cellular είναι σαφώς ικανό να στέλνει μεγάλες ποσότητες δεδομένων, ειδικά μέσω του 4G, το κόστος και η κατανάλωση ενέργειας είναι πολύ υψηλά για πολλές εφαρμογές. Μπορεί, ωστόσο, να είναι ιδανικό για έργα χαμηλών απαιτήσεων σε bandwidth, όπως για παράδειγμα μία εγκατάσταση που βασίζεται σε αισθητήρες που στέλνουν πολύ λίγα δεδομένα μέσω του Διαδικτύου.

2.1.5.1.6 NFC

Το **NFC** (Near Field Communication) ή επικοινωνία κοντινού επιπέδου είναι πρότυπο που επιτρέπει απλές και ασφαλείς επικοινωνίες μεταξύ ηλεκτρονικών συσκευών. Είναι ιδιαίτερα συχνή η χρήση του σε έξυπνα τηλέφωνα (smartphone), επιτρέποντας στους καταναλωτές να πραγματοποιούν ανέπαφες τραπεζικές συναλλαγές. Ουσιαστικά, επεκτείνει την ικανότητα της τεχνολογίας καρτών χωρίς επαφή και επιτρέπει στις συσκευές να μοιράζονται πληροφορίες. Ωστόσο, η χρήση του είναι πολύ συγκεκριμένη, δεδομένου ότι λειτουργεί σε απόσταση μικρότερη από 4 εκατοστά.

2.1.5.1.7 LoRaWan

Το **LoRaWAN** (Long Range Wide Area Network) είναι από τα δημοφιλή πρότυπα συνδεσιμότητας IoT, καθώς στοχεύει σε εγκαταστάσεις δικτύου ευρείας περιοχής (WAN). Σχεδιάστηκε για την παροχή WAN χαμηλής ισχύος, το οποίο απαιτείται για την υποστήριξη χαμηλού κόστους και ασφαλούς επικοινωνίας σε συσκευές IoT που χρησιμοποιούνται σε έξυπνες πόλεις και βιομηχανικές εφαρμογές. Συγκεκριμένα, έχει τη δυνατότητα να υποστηρίξει μεγάλα δίκτυα με εκατομμύρια συσκευές, με τους ρυθμούς δεδομένων να κυμαίνονται από 0,3kbps έως 50kbps. [6]

2.1.5.2 Πρωτόκολλα επικοινωνίας

Εφόσον ολοκληρωθεί η συνδεσιμότητα με το δίκτυο, οι συσκευές επικοινωνούν, είτε μεταξύ τους, είτε με κάποια άλλη συσκευή – όπως με την εφαρμογή ενός κινητού τηλεφώνου – μέσω ενός πρωτοκόλλου επικοινωνίας. Το κάθε πρωτόκολλο έχει τα δικά του χαρακτηριστικά που καθορίζουν τον τρόπο λειτουργίας του. Τα κυριότερα πρωτόκολλα επικοινωνίας στο Διαδίκτυο των Πραγμάτων, τα οποία είναι και ευρέως χρησιμοποιούμενα, αναλύονται παρακάτω.

2.1.5.2.1 AMQP

Το πρωτόκολλο **AMQP** (Advanced Message Queuing Protocol) βασίζεται στο TCP/IP. Είναι σχεδιασμένο για την επικοινωνία μεταξύ των εξυπηρετητών, με πρωταρχικό σκοπό τη μικρότερη δυνατή απώλεια μηνυμάτων. Είναι ιδιαίτερα αξιόπιστο στην επεξεργασία πολλαπλών συναλλαγών, λαμβάνοντας υπόψη τη σειρά προτεραιότητας της κάθε συναλλαγής. Αποτελεί το συνηθέστερο πρωτόκολλο επικοινωνίας σε τραπεζικές συναλλαγές.

2.1.5.2.2 CoAP

Το πρωτόκολλο **CoAP** (Constrained Application Protocol) θεωρείται απλούστερη εκδοχή του HTTP πρωτοκόλλου (HyperText Transfer Protocol). Η λειτουργία του βασίζεται στο

πρωτόκολλο UDP (Universal Datagram Protocol) και, χάρη σε αυτό, εφαρμόζεται σε συσκευές περιορισμένων πόρων που συνδέονται στο ίδιο δίκτυο ή σε συσκευές διαφορετικών δικτύων που επικοινωνούν μεταξύ τους μέσω κοινού δικτύου.

2.1.5.2.3 XMPP

Το **XMPP** (Extensible Messaging and Presence Protocol, αρχική ονομασία Jabber) είναι ένα πρωτόκολλο επικοινωνίας σχεδιασμένο για ανταλλαγή άμεσων μηνυμάτων (IM), πληροφορίες παρουσίας και κατάστασης. Βασίζεται στο μορφότυπο XML (Extensible Markup Language) και επιτρέπει την ανταλλαγή δομημένων δεδομένων σχεδόν σε πραγματικό χρόνο μεταξύ δύο ή περισσότερων οντοτήτων δικτύου. Χάρη στον επεκτάσιμο σχεδιασμό του, το πρωτόκολλο προσφέρει μία πληθώρα εφαρμογών, πέρα από το παραδοσιακό IM, στον ευρύτερο χώρο του message-oriented middleware, όπως για παράδειγμα τη σηματοδότηση για VoIP (Voice over Internet Protocol), βίντεο, μεταφορά αρχείων και παιχνίδια.

Η αρχιτεκτονική του δικτύου XMPP είναι παρόμοια με αυτή των email. Ο καθένας μπορεί να εκτελέσει το δικό του διακομιστή, χωρίς την ύπαρξη κεντρικού διακομιστή. Αυτή η προσέγγιση επιτρέπει στους χρήστες να συνεργάζονται με άλλους σε οποιονδήποτε διακομιστή χρησιμοποιώντας ένα λογαριασμό χρήστη, παρόμοιο με μία διεύθυνση email.

2.1.5.2.4 DDS

Το πρωτόκολλο **DDS** (Data – Distribution Service) στοχεύει στις συσκευές που χρησιμοποιούν άμεσα δεδομένα και τα διαμοιράζει σε άλλες, χωρίς να είναι απαραίτητη η παρεμβολή δομής εξυπηρέτησης. Πρακτικά, δηλαδή, πραγματοποιεί απευθείας σύνδεση των συσκευών μεταξύ τους. Ενσωματώνει τα στοιχεία ενός πλήρους συστήματος, παρέχοντας άμεση (low – latency) συνδεσιμότητα δεδομένων, εξαιρετική αξιοπιστία και επεκτάσιμη.

2.1.5.2.5 MQTT

Το **MQTT** (Message Queueing Telemetry Transport) είναι ένα ελαφρύ πρωτόκολλο M2M (Lightweight Machine – to – Machine). Βασίζεται στο TCP/IP και έχει ως πολύ σημαντικό πλεονέκτημά του το γεγονός πως δύναται να μεταφέρει κάθε τύπο πληροφορίας. Σχεδόν όλες οι cloud πλατφόρμες IoT υποστηρίζουν το πρωτόκολλο MQTT για την αποστολή και λήψη δεδομένων από έξυπνα αντικείμενα. Προσφέρεται ως ένας απλός τρόπος διανομής πληροφοριών τηλεμετρίας που χρησιμοποιεί το μοτίβο επικοινωνίας δημοσίευσης – εγγραφής (publish/subscribe). Με αυτόν τον τρόπο, επιτρέπει στις συσκευές να επικοινωνούν ασύγχρονα και ιδιαίτερα αποδοτικά, ακόμα και σε απομακρυσμένα δίκτυα μικρής αξιοπιστίας. [2]

2.1.5.2.6 RESTful HTTP

Το Representational State Transfer (REST) δεν είναι, στην πραγματικότητα, ένα πρωτόκολλο επικοινωνίας, αλλά μία αρχιτεκτονική προσέγγιση για τη διευθέτηση των επικοινωνιών μεταξύ αυτόνομων συστημάτων. Το REST χρησιμοποιεί τις μεθόδους HTTP GET, POST, PUT και DELETE για να παρέχει ένα σύστημα ανταλλαγής μηνυμάτων, όπου όλες οι ενέργειες μπορούν να εκτελεστούν απλά χρησιμοποιώντας τις σύγχρονες εντολές HTTP αιτήματος/απόκρισης (request/response). Ωστόσο, αυτή η απλότητα και ευελιξία του το καθιστούν πολύ χρήσιμο σε εφαρμογές IoT. [7]

Πρωτόκολλο	Transport	Μηνύματα	Τύπος επικοινωνίας	QoS	Ασφάλεια
AMQP	TCP	Publish/Subscribe	Middleware	Ναι	TLS/SSL
CoAP	UDP	Request/Response	Web Transfer	Ναι	DTLS
XMPP	TCP	Request/Response, Publish/Subscribe	Client/Server, Server/Server	Όχι	TLS/SASL
DDS	TCP, UDP	Publish/Subscribe	M2M, Device to Device	Ναι	TLS/SSL
MQTT	TCP	Publish/Subscribe	Lightweight M2M	Ναι	TLS/SSL
RESTful	HTTP	Request/Response	Client/Server	Όχι	HTTPS

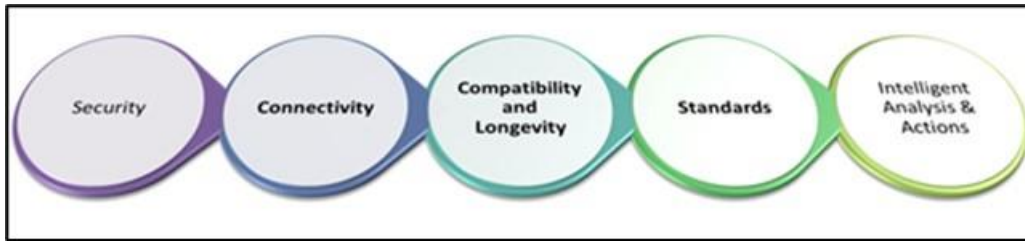
Πίνακας 1. Σύγκριση πρωτοκόλλων επικοινωνίας. [7]

2.1.6 Προκλήσεις

Το Διαδίκτυο των Πραγμάτων μπορεί να προσελκύει το ενδιαφέρον σε πολλούς τομείς της ανθρώπινης δραστηριότητας, όπως η βιομηχανία, οι μεταφορές και η υγειονομική περίθαλψη, αλλά, όπως κάθε νέα τεχνολογία, για να ευδοκιμήσει, χρειάζεται να αντιμετωπιστούν ορισμένες προκλήσεις.

2.1.6.1 Τεχνολογικές προκλήσεις

Προκειμένου να εξασφαλιστεί ομαλή και αποδοτική λειτουργία του IoT, είτε ως αυτόνομη λύση, είτε ως μέρος των υπάρχοντων συστημάτων, θα πρέπει να αντιμετωπιστούν αρκετές τεχνολογικές προκλήσεις. (Εικόνα 5) [8], [9], [10], [11], [12]

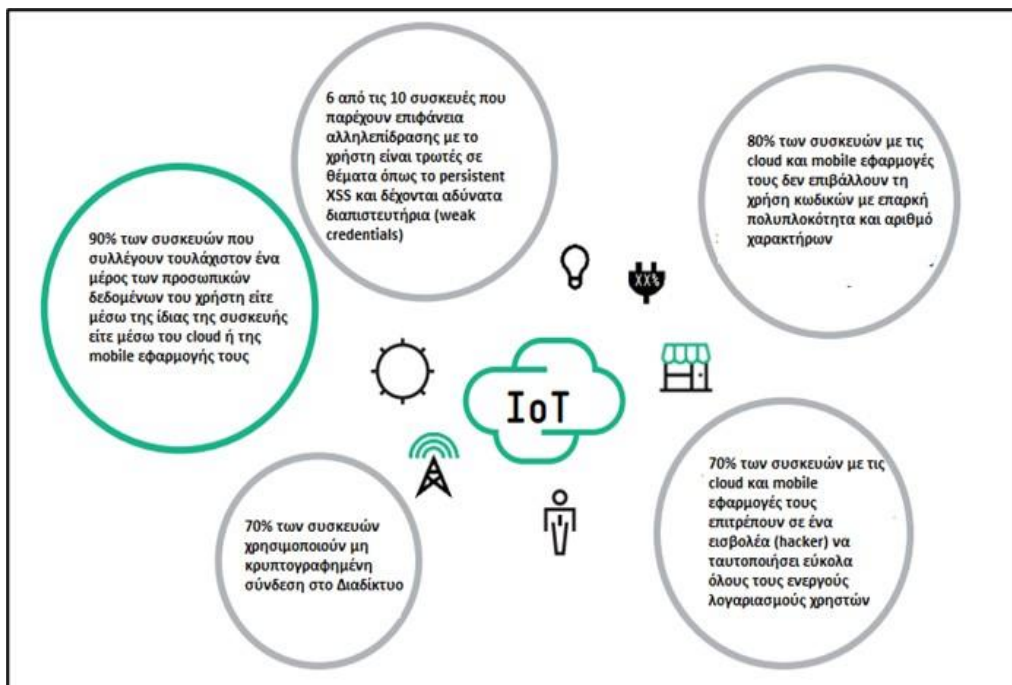


Εικόνα 5. Τεχνολογικές προκλήσεις του IOT. [8]

2.1.6.1.1 Ασφάλεια

Από τη γέννηση της ιδέας του Διαδικτύου των Πραγμάτων μέχρι σήμερα, παρατηρούνται σοβαρές έλλειψεις όσον αφορά την **ασφάλεια** που παρέχουν οι συσκευές. Επειδή η ιδέα της δικτύωσης συσκευών και άλλων αντικειμένων είναι σχετικά νέα, η ασφάλεια δε θεωρείται πάντα πρώτη προτεραιότητα κατά τη φάση σχεδιασμού ενός προϊόντος. Επιπλέον, επειδή το IoT αποτελεί μία νέα αγορά, πολλοί κατασκευαστές ενδιαφέρονται περισσότερο να διαθέσουν τα προϊόντα τους γρήγορα, παρά να λάβουν τα απαραίτητα μέτρα για να δημιουργήσουν ασφάλεια εξ αρχής.

Την κατάσταση έρχεται να επιδεινώσει η χρήση κωδικών ή προεπιλεγμένων κωδικών πρόσβασης, οι οποίοι μπορούν να οδηγήσουν σε παραβιάσεις. Ακόμα και σε περίπτωση που οι κωδικοί πρόσβασης αλλάξουν, συχνά δεν είναι αρκετά ισχυροί για να αποτρέψουν τη διείσδυση. (*Εικόνα 6*)



Εικόνα 6. Κίνδυνοι στην ασφάλεια των συσκευών IoT. [4]

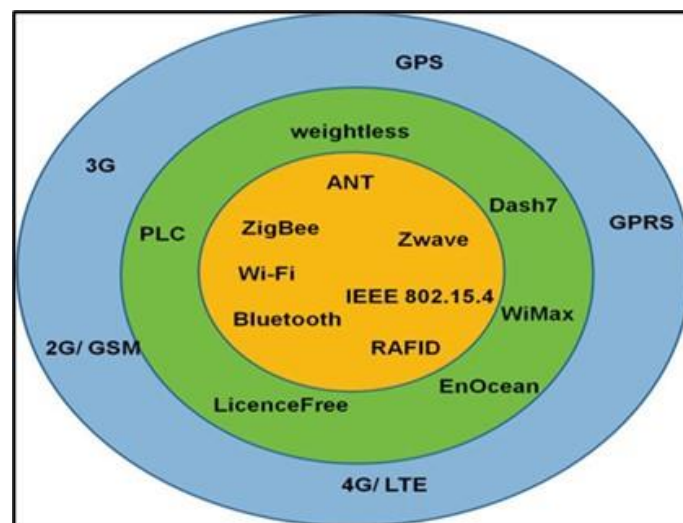
Ένα άλλο κοινό πρόβλημα που αντιμετωπίζουν οι συσκευές IoT είναι ότι συχνά, λόγω περιορισμών, δε διαθέτουν τους υπολογιστικούς πόρους που είναι απαραίτητοι για την

εφαρμογή ισχυρής ασφάλειας. Ως εκ τούτου, πολλές συσκευές δεν προσφέρουν ή δε μπορούν να προσφέρουν προηγμένες δυνατότητες ασφαλείας. Για παράδειγμα, οι αισθητήρες υγρασίας ή θερμοκρασίας δε μπορούν να χειριστούν προηγμένη κρυπτογράφηση ή άλλα μέτρα ασφαλείας. Επιπλέον, καθώς πολλές συσκευές IoT βασίζονται στη λογική του «ρυθμίζω και ξεχνώ», τίθενται σε λειτουργία και, μέχρι το τέλος της ζωής τους, σχεδόν ποτέ δε λαμβάνουν ενημερώσεις ασφαλείας ή λογισμικού. Όσον αφορά τις ενημερώσεις, πολλά συστήματα περιλαμβάνουν υποστήριξη μόνο για ένα καθορισμένο χρονικό πλαίσιο. Συνεπώς, η δυνατότητα επιπλέον ασφάλειας μπορεί να λήξει εάν δεν προστεθεί επιπλέον υπηρεσία υποστήριξης. Καθώς πολλές συσκευές IoT παραμένουν στο δίκτυο για πολλά έτη, η έννοια της επικαιροποιημένης ασφάλειας αποτελεί μία από τις δυσκολότερες προκλήσεις.

Η σύνδεση παλαιών στοιχείων, που δεν είναι εγγενώς σχεδιασμένα για συνδεσιμότητα IoT, αποτελεί μία ακόμα πρόκληση ασφαλείας. Η αντικατάσταση της παλαιάς υποδομής με νέα, συνδεδεμένη τεχνολογία δεν είναι – πάντα – συμφέρουσα, οπότε πολλές συσκευές θα μετασκευαστούν με έξυπνους αισθητήρες. Ωστόσο, καθώς τα παλαιά στοιχεία, που πιθανότατα δεν έχουν ενημερωθεί ή δεν είχαν ποτέ ασφάλεια έναντι σύγχρονων απειλών, παραμένουν σε λειτουργία, η πιθανότητα της επίθεσης εντείνεται. [13]

2.1.6.1.2 Συνδεσιμότητα

Μία από τις σοβαρότερες προκλήσεις που αντιμετωπίζει το IoT είναι η αποτελεσματική **συνδεσιμότητα** των συσκευών. Καθένα από τα πρότυπα που χρησιμοποιούνται σήμερα καλύπτουν συγκεκριμένες ανάγκες. Ωστόσο, δεν υπάρχει μία ολοκληρωμένη λύση που να καλύπτει τα πάντα. Οι επιλογές συνδεσιμότητας ταξινομούνται σε τρεις μεγάλες κατηγορίες λαμβάνοντας υπόψη την κάλυψη της γεωγραφικής περιοχής τους, δηλαδή προσωπικό δίκτυο (PAN), τοπικό δίκτυο (LAN) και δίκτυο ευρείας περιοχής (WAN). (*Εικόνα 7*)



Εικόνα 7. Κατηγορίες συνδεσιμότητας. [14]

Οι λύσεις που προτείνονται για να αντιμετωπίσουν τα ζητήματα αυτά περιλαμβάνουν την τροποποίηση των υπάρχοντων δικτύων Wi-Fi, δεδομένου ότι αποτελεί το συνηθέστερο τρόπο σύνδεσης στο IoT, με σκοπό να επιτευχθεί ευρύτερη κάλυψη και να υποστηριχθούν δίκτυα πλέγματος (mesh networking) που βοηθούν σημαντικά στην εξοικονόμηση ενέργειας και τη βελτίωση της εμβέλειας. Κρίνεται απαραίτητο να υποστηριχτούν οι επικοινωνίες από συσκευή σε συσκευή (D2D), από συσκευή σε διακομιστή (D2S) και από διακομιστή σε διακομιστή (S2S), προκειμένου να διευκολυνθεί η ανταλλαγή πληροφοριών εντός του IoT. [14]

2.1.6.1.3 Συμβατότητα

Η **συμβατότητα** είναι σημαντική πρόκληση για το οικοσύστημα του IoT, καθώς διάφορα προϊόντα συνδέονται μεταξύ τους. Οι νέες τεχνολογίες συχνά χαρακτηρίζονται από ένα μεγάλο ανταγωνισμό. Το IoT αναπτύσσεται σε πολλές διαφορετικές κατευθύνσεις, με πολλές διαφορετικές τεχνολογίες να ανταγωνίζονται για να γίνουν το πρότυπο.

Ο ανταγωνισμός, με τη σειρά του, δημιουργεί αυξημένες επιλογές στους καταναλωτές, αλλά μπορεί να δημιουργήσει και απογοήτευση από την έλλειψη συμβατότητας. Αυτή, συνήθως, προέρχεται από μη ενοποιημένες υπηρεσίες cloud, από έλλειψη τυποποιημένων πρωτοκόλλων και διαφορές στο υλικό, το λογισμικό και τα λειτουργικά συστήματα μεταξύ συσκευών IoT. Επειδή δεν υπάρχει μία ενιαία γλώσσα επικοινωνίας, τα περισσότερα προϊόντα αδυνατούν να συνδεθούν μεταξύ τους και οδηγούν σε ζητήματα συμβατότητας. Για τη σύνδεση συσκευών μεταξύ τους, απαιτείται συνεργασία μεταξύ επιχειρήσεων για την απόκτηση πληροφοριών υποδομής για κάθε προϊόν και το σχεδιασμό μίας καθολικής γλώσσας κωδικοποίησης από τους προγραμματιστές. Για να διασφαλιστεί η επιτυχία του IoT, απαιτείται λύση σε αυτά τα ζητήματα.

Το πρότυπο επικοινωνίας αποτελεί το συνηθέστερο πρόβλημα συμβατότητας. Το Bluetooth υπήρξε από καιρό το πρότυπο επικοινωνίας για συσκευές IoT, όσον αφορά τον οικιακό αυτοματισμό. Ωστόσο, αρκετοί ανταγωνιστές άρχισαν να αμφισβητούν τις δυνατότητές του, οδηγώντας την άνθιση άλλων πρωτοκόλλων, δυσχεραίνοντας ακόμα περισσότερο την ποικιλομορφία του χώρου.

Η συμβατότητα για συσκευές IoT εξαρτάται, επίσης, από την ενασχόληση των χρηστών με ενημερώσεις που βελτιώνουν και διορθώνουν τις συσκευές τους. Όταν οι συσκευές IoT φέρουν διαφορετικές εκδόσεις λογισμικού, μπορεί να προκαλέσουν πολλά είδη προβλημάτων απόδοσης και έλλειψης ασφαλείας. Συνεπώς, είναι πολύ σημαντικό οι καταναλωτές IoT να διατηρούν τις συσκευές τους αναβαθμισμένες και ενημερωμένες. [8], [14], [15]

2.1.6.1.4 Προτυποποίηση

Το Διαδίκτυο των Πραγμάτων (IoT) είναι ένας καινούργιος αναδυόμενος τομέας της σύγχρονης τεχνολογίας που χρησιμοποιείται σε ένα ευρύ φάσμα της ανθρώπινης δραστηριότητας. Τα περισσότερα πρότυπα Διαδικτύου που υπάρχουν σήμερα δεν είχαν την πρόβλεψη να συμπεριλάβουν το IoT στις προδιαγραφές τους και, επομένως, το πεδίο εφαρμογής τους δεν επαρκεί για να το υποστηρίξει τεχνικά και οικονομικά. Τα τεχνολογικά πρότυπα απαρτίζονται από πρωτόκολλα δικτύου, πρωτόκολλα επικοινωνίας, πρότυπα συγκέντρωσης δεδομένων, καθώς και πρωτόκολλα που αναφέρονται στο χειρισμό, την επεξεργασία και την αποθήκευση των δεδομένων που συλλέγονται από τους αισθητήρες. Σήμερα, πολλές συσκευές IoT έχουν αναπτυχθεί με ιδιόκτητα (proprietary) πρωτόκολλα και αυτό καθιστά δύσκολη την επικοινωνία μεταξύ πολλών συσκευών IoT. [8], [16]

Με την εκθετική αύξηση του αριθμού και των τύπων των έξυπνων συσκευών κατά τα επόμενα έτη, το IoT αποτελεί μία μεγάλη πρόκληση και ειδικότερα για τις ρυθμιστικές αρχές. Μία από τις μεγαλύτερες προκλήσεις, από τις οποίες εξαρτάται η τελική επιτυχία του IoT, είναι η ανάπτυξη παγκόσμιων διαλειτουργικών προτύπων. Ωστόσο, τα πρότυπα IoT σήμερα είναι ακόμη ανοικτά, σε επίπεδο συσκευής, πρωτοκόλλου και λογισμικού, καθώς δεν υπάρχουν παγκόσμια επικυρωμένα πλαίσια τυποποίησης. Χωρίς επιβολή προτύπων, η αξία και η εμπορική βιωσιμότητα του IoT θα δυσκολευτούν να αξιοποιήσουν πλήρως τις δυνατότητές του. [16]

Η αποτυχία της έλλειψης παγκόσμιων προτύπων δεν οφείλεται στην έλλειψη προσπάθειας. Τα τελευταία χρόνια, αρκετές βιομηχανίες και κυβερνητικοί οργανισμοί δημιούργησαν πρότυπα για τη βελτίωση της διαλειτουργικότητας και της ασφάλειας των συσκευών IoT, όπως:

- Cloud Security Alliance (CSA)
- Groupe Spéciale Mobile Association's (GSMA)
- IEEE Standards Association
- ioXt Alliance
- IoT Security Foundation
- Open Web Application Security Project (OWASP)
- U.S. Department of Homeland Security. [17]

Η τρέχουσα έλλειψη παγκόσμιων προτύπων, που σχετίζονται με την αναφορά και το χειρισμό τρωτών σημείων του IoT, σημαίνει ότι οι προμηθευτές δεν έχουν καμία υποχρέωση να αποκαλύψουν ή να αποκαταστήσουν τρωτά σημεία, αφήνοντας εκατομμύρια ευάλωτων συσκευών σε κίνδυνο εκμετάλλευσης. [17]

Για την αντιμετώπιση αυτών των δυσχερειών, χρειάζεται να συμφωνηθούν κοινά πρότυπα. Αυτά θα πρέπει να αναφέρονται όχι μόνο στο υλικό, αλλά και στο λογισμικό, ώστε οι

μελλοντικές εφαρμογές να μπορούν να αναπτυχθούν σε ένα ομοιογενές περιβάλλον, το οποίο θα επιτρέπει την εύκολη μετεγκατάσταση σε όλα τα συστήματα. [16]

2.1.6.1.5 Ευφυής Ανάλυση & Ενέργειες

Το τελευταίο στάδιο της υλοποίησης του IoT είναι η εξαγωγή πληροφοριών από την ανάλυση των δεδομένων που συλλέγονται. Αυτό μπορεί να επιτευχθεί με τη βελτίωση των μοντέλων τεχνικής νοημοσύνης και του λογισμικού ποιοτικής ανάλυσης, καθώς και με την επεξεργασία και ανάλυση δεδομένων σε πραγματικό χρόνο. [8]

Οι προκλήσεις που ανακύπτουν από την ανάγκη υιοθέτησης έξυπνων αναλυτικών στοιχείων στο IoT είναι η ανακριβής ανάλυση, λόγω ελαττωμάτων στα δεδομένα ή/και στο μοντέλο, η ικανότητα συστημάτων παλαιού τύπου να αναλύουν μη δομημένα δεδομένα και η διαχείρισή τους σε πραγματικό χρόνο. [8]

Οι ενέργειες, που μπορούν να εκφραστούν ως διασυνδέσεις M2M και M2H και αποτελούν σημαντικοί παράγοντες που ευνοούν την ταχεία επέκταση του IoT, είναι:

- Μείωση κόστους αισθητήρων
- Μείωση κόστους συλλογής και αποθήκευσης δεδομένων μέσω λύσεων cloud
- Διεύρυνση της συνδεσιμότητας στο Διαδίκτυο
- Αύξηση υπολογιστικής ισχύος [8], [18]

2.1.6.2 Προκλήσεις των επιχειρήσεων

Ένα σοβαρό κίνητρο για την επένδυση και τη λειτουργία οποιασδήποτε επιχείρησης είναι ένα υγιές και σταθερό επιχειρηματικό μοντέλο για το IoT, το οποίο πρέπει να ικανοποιεί όλες τις απαιτήσεις για όλα τα είδη εμπορίου: κάθετες, οριζόντιες, και καταναλωτικές αγορές. [8]

Με τόσο έντονο ανταγωνισμό στην αγορά του IoT, οι επιχειρήσεις που επιθυμούν να εισέλθουν σε αυτόν τον καινοτόμο τομέα πρέπει να είναι προετοιμασμένες για μία αγορά που συνεχώς μεταβάλλεται, με τους πελάτες της να θέλουν συνεχώς μία ομαλότερη και πιο προηγμένη εμπειρία. [15]

Όπως προαναφέρθηκε, παρά τις προσπάθειες για δημιουργία παγκόσμιων προτύπων, δεν υπήρξε μέχρι τώρα επαρκές κίνητρο ώστε η βιομηχανία να ευθυγραμμιστεί με ένα μόνο σύνολο προτύπων. Το αποτέλεσμα είναι ένα συνονθύλευμα οδηγιών που αφορούν μόνο ορισμένες πτυχές για την ασφάλεια των συσκευών IoT. [17]

Όσον αφορά την ανάπτυξη βιομηχανικού περιβάλλοντος IoT, το οποίο απαιτεί μαζική επένδυση, οι επιχειρήσεις δυσκολεύονται να το υιοθετήσουν, καθώς το υλικό και το λειτουργικό δεν είναι ανοικτά και διαλειτουργικά. Συνεπώς, παρατηρείται η αναγκαιότητα να δημιουργηθούν ανοικτές και ολοκληρωμένες λύσεις IoT που να εξυπηρετούν συγκεκριμένα τις

απαιτήσεις της βιομηχανίας. Σημαντικό να αναφερθεί είναι πως αυτές οι λύσεις, αντί να αντικαταστήσουν τις υπάρχουσες εφαρμογές με νέα συστήματα, θα πρέπει να είναι αρκετά ευέλικτες, ώστε να επιτρέπουν στις βιομηχανίες να εξελίσσονται και να προσαρμόζονται σταδιακά. Αυτό απαιτεί εμπειρογνομοσύνη και επενδύσεις για τη δημιουργία καινοτομίας στις υπάρχουσες αρχιτεκτονικές υλικού και λογισμικού. [14]

2.1.6.3 Προκλήσεις της κοινωνίας και διασφάλιση της ιδιωτικότητας

Η ενσωμάτωση του IoT σε πληθώρα εφαρμογών, όπως τις μεταφορές, την υγεία, τον οικιακό αυτοματισμό και την ασφάλεια, έχει επιπτώσεις στην καθημερινότητα των ανθρώπων. Ο άνθρωπος, είτε εν γνώσει του, είτε εν αγνοία του, χρησιμοποιεί, συνεχόμενα και με αυξανόμενο ρυθμό, σημαντικό αριθμό προϊόντων και υπηρεσιών με διασύνδεση στο IoT, με σκοπό να βελτιώσει την ποιότητα της ζωής του και να εξοικονομήσει χρόνο και χρήμα. [19]

Η κατανόηση του IoT από τους υποψήφιους πελάτες δεν είναι εύκολη υπόθεση, καθώς εμφανίζονται νέες χρήσεις για τις συσκευές, ενώ ταυτόχρονα ο ρυθμός ανάπτυξης νέων συσκευών είναι ραγδαίος. Η έλλειψη κατανόησης ή εκπαίδευσης των καταναλωτών, κυρίως όσον αφορά τις βέλτιστες πρακτικές για την ασφάλεια των συσκευών IoT, μπορεί να κλονίσει την εμπιστοσύνη τους, αλλά, πιο σημαντικό, τους αφήνει ευάλωτους σε μία εισβολή στην ιδιωτική τους ζωή. [8]

Η **προστασία της ιδιωτικής ζωής** αποτελεί βασική απαίτηση των περισσότερων εφαρμογών και έχει μετατραπεί σε σοβαρή ανησυχία, η οποία προσέλκυσε την προσοχή διακεκριμένων τεχνολογικών εταιρειών και κυβερνητικών υπηρεσιών παγκοσμίως.

Είναι σύνηθες φαινόμενο οι συσκευές IoT, χρησιμοποιώντας ως αφορμή την περιορισμένη χωρητικότητα αποθήκευσης, να αποθηκεύουν ή να διαμοιράζονται δεδομένα σε απομακρυσμένες μονάδες αποθήκευσης (storage accounts). Οι χρήστες, λόγω αμέλειας ή άγνοιας, εναποθέτουν πληροφορίες, ακόμα και στιγμιότυπα, της ιδιωτικής τους ζωής, σε περιβάλλοντα που, τις περισσότερες φορές, δεν πληρούν βασικές αρχές ασφάλειας, χωρίς να έχουν συμφωνήσει **ρητά** σε αυτό.

Αυτή την έλλειψη, δηλαδή ενός πλαισίου νομικών κανόνων, ήρθε να καλύψει τα τελευταία χρόνια ο **Γενικός Κανονισμός Προστασίας Δεδομένων Προσωπικού Χαρακτήρα** (General Data Protection Regulation) ή κοινώς **GDPR**. Δημιουργήθηκε το Μάιο του 2016 και τέθηκε σε πλήρη νομική ισχύ, σε όλες τις χώρες της Ευρωπαϊκής Ένωσης, στις 25 Μαΐου 2018.

Οι νέοι κανονισμοί άλλαξαν το επιχειρησιακό και νομικό τοπίο για τις εταιρείες πληροφορικής, και, κατά επέκταση, και εκείνων που δραστηριοποιούνται με το IoT. Οι κατασκευαστές συσκευών οφείλουν να συμμορφώνονται με διαφορετικό κανονιστικό κώδικα, ενώ οι πλατφόρμες cloud IoT έχουν ένα εντελώς νέο σύνολο νομικών υποχρεώσεων που πρέπει να τηρούν όταν ορίζουν τους όρους χρήσης τους.

Οι κυριότεροι άξονες επιρροής του IoT από την εφαρμογή του GDPR απαριθμούνται παρακάτω.

2.1.6.3.1 Παραβιάσεις Ασφαλείας

Για λόγους διαφάνειας, ο κανονισμός GDPR αναγκάζει γενικές υποχρεωτικές ειδοποιήσεις για παραβιάσεις δεδομένων κάθε είδους. Από το Μάιο του 2018, οι υπεύθυνοι συντήρησης δεδομένων υποχρεούνται να ανακοινώνουν τυχόν παραβιάσεις δεδομένων και, σε ορισμένες ακραίες περιπτώσεις, οφείλουν να ειδοποιήσουν απευθείας τους τελικούς χρήστες ότι τα δεδομένα τους έχουν παραβιαστεί προσωρινά. Οι ειδοποιήσεις πρέπει να γίνουν εντός 72 ωρών από το συμβάν, βάσει του νόμου. Ο στόχος αυτών των νέων κανονισμών είναι να ενισχυθεί η ασφάλεια στο διαδίκτυο (που παραμένει σταθερά ένα σοβαρό ζήτημα) και να προσφέρει περισσότερες πληροφορίες σχετικά με το ιστορικό δεδομένων των χρηστών.

2.1.6.3.2 Συγκατάθεση

Το ζήτημα της συγκατάθεσης σχετικά με την ανταλλαγή πληροφοριών είναι ένα μείζον ζήτημα τα τελευταία έτη. Πολλοί εκπρόσωποι της κοινότητα του IoT ζητούσαν περισσότερη συζήτηση συναίνεσης μεταξύ των πελατών και των παρόχων IoT εδώ και χρόνια. Ο κανονισμός GDPR αποσκοπεί να αντιμετωπίσει αυτές τις ανησυχίες, αναγκάζοντας τις εταιρείες να προσφέρουν περισσότερες ρήτρες opt-in ή opt-out. Δε μπορεί πλέον η αδράνεια ενός χρήστη να θεωρηθεί συγκατάθεση. Απαιτείται από τους υπευθύνους επεξεργασίας δεδομένων να λαμβάνουν ρητή συγκατάθεση από όλους τους πελάτες πριν εξαγάγουν τα προσωπικά τους δεδομένα.

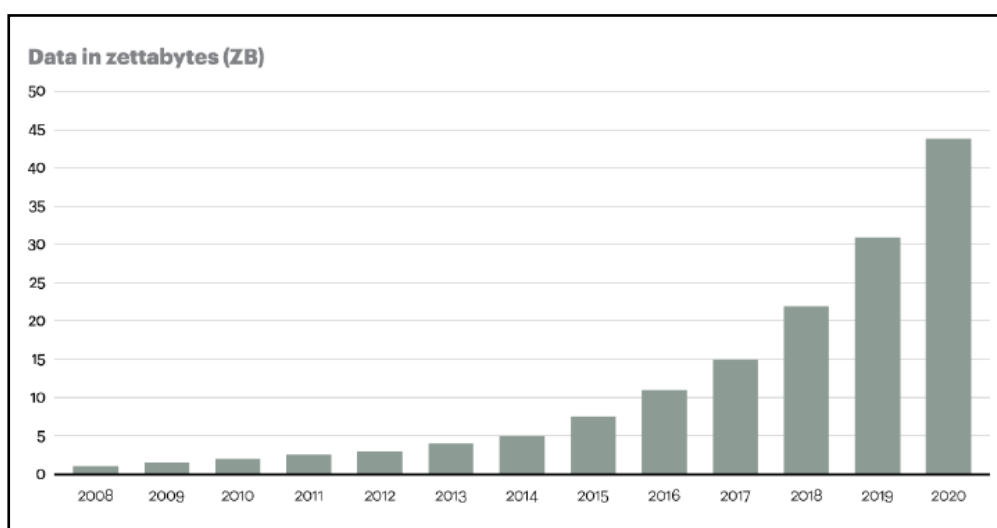
2.1.6.3.3 Δικαιώματα υποκειμένων δεδομένων

Πριν από τον κανονισμό GDPR, οι ελεγκτές δεδομένων είχαν πολύ μεγαλύτερη ελευθερία, σε σύγκριση με τους χρήστες, όσον αφορά το ποια δεδομένα μπορούσαν να καταγραφούν ή όχι, καθώς και τον τρόπο καταγραφής τους. Μετά τις νέες αλλαγές, δόθηκε στους τελικούς χρήστες το δικαίωμα για πολύ περισσότερο εκτεταμένο έλεγχο στα δεδομένα τους. Μπορούν, για παράδειγμα, να αιτηθούν αφαίρεσης όλων των δεδομένων τους από το σύστημα καταγραφής. Επίσης, ένα σημαντικό πλεονέκτημα αποτελεί το γεγονός πως δόθηκε η δυνατότητα φορητότητας των δεδομένων, επιτρέποντας, ουσιαστικά, στους χρήστες να έχουν πρόσβαση και να χρησιμοποιούν τα δικά τους δεδομένα σε άλλες πλατφόρμες ή υπηρεσίες.

2.2 Δεδομένα Μεγάλης Κλίμακας (Big Data)

2.2.1 Ορισμός

Ο όρος «Δεδομένα Μεγάλης Κλίμακας» (Big Data) αναφέρεται σε σύνολα σύνθετων, δομημένων, ημιδομημένων και μη δομημένων δεδομένων μεγάλου όγκου. Η αποθήκευση, η ανάκτηση, η επεξεργασία και η ανάλυση αυτών των δεδομένων απαιτούν τη χρήση νέων τεχνολογιών και έχουν ως αποτέλεσμα την άντληση πληροφοριών και τη χρησιμοποίησή τους. Απώτερο στόχο αποτελεί η βελτιστοποίηση των διαδικασιών που χρησιμοποιούνται σε διάφορους τομείς (όπως την υγεία, τη βιομηχανία και τις μεταφορές), η πρόβλεψη πιθανών προκλήσεων και η διευκόλυνση της λήψης αποφάσεων. [20], [21], [22] (*Εικόνα 8*)



Εικόνα 8. Ποσοστιαία αύξηση δεδομένων μεγάλης κλίμακας, ανά έτος. [23]

2.2.2 Εξέλιξη

Παρόλο που η έννοια των Δεδομένων Μεγάλης Κλίμακας εμφανίστηκε πρόσφατα, τα θεμέλιά της είχαν τεθεί πάρα πολλά χρόνια πριν. Ακόμα και πριν την εμφάνιση των ηλεκτρονικών υπολογιστών (H/Y), υπήρχε πάντα η ανάγκη για αποθήκευση, ανάλυση και αναπαραγωγή πληροφοριών.

Χαρακτηριστικά παραδείγματα του παρελθόντος αποτελούν η βιβλιοθήκη της Βαβυλώνας (2400 π.Χ.) και η βιβλιοθήκη της Αλεξάνδρειας (300 π.Χ.). Το 100 μ.Χ. ανακαλύπτεται ο μηχανισμός των Αντικυθήρων. Θεωρείται ένας αρχαίος αναλογικός, μηχανικός υπολογιστής και όργανο αστρονομικών παρατηρήσεων.

Το 1663, ο John Graunt διεξάγει το πρώτο καταγεγραμμένο πείραμα στατιστικής ανάλυσης, που αποσκοπούσε στον περιορισμό της εξάπλωσης της επιδημίας της πανούκλας στην Ευρώπη.

Το 1881, ο Herman Hollerith δημιουργεί μία μηχανή πινακοποίησης, που χρησιμοποιήθηκε για να μειώσει το φόρτο εργασίας των απογραφών των Η.Π.Α.

Το 1965, η κυβέρνηση των Η.Π.Α. σχεδιάζει την κατασκευή του μεγαλύτερου κέντρου αποθήκευσης δεδομένων, εντός του οποίου καταγράφονται 742 εκατομμύρια φορολογικά στοιχεία και 175 εκατομμύρια αποτυπώματα πολιτών σε μαγνητική ταινία. Ενώ, το 1975, η IBM εισάγει το σχεσιακό μοντέλο βάσης δεδομένων. Με αυτόν τον τρόπο, έδωσε τη δυνατότητα στον κάθε άνθρωπο να χειρίζεται μία βάση δεδομένων, χωρίς να απαιτούνται εξειδικευμένες γνώσεις πληροφορικής. Το 1976 εφαρμόζεται ευρέως η χρήση Συστημάτων Σχεδιασμού Απαιτήσεων Υλικών (Material Requirements Planning Systems – MRP) στις επιχειρήσεις, καθιστώντας τον Η/Υ μέρος της καθημερινότητας για την καταγραφή δεδομένων.

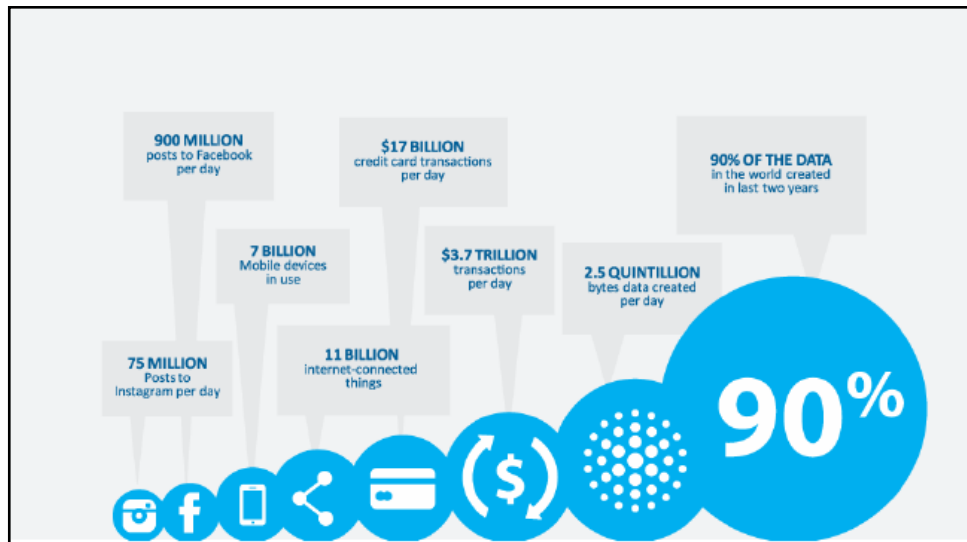
Ο όρος “Big Data” εμφανίζεται για πρώτη φορά το 1989. Σημαντικό ρόλο στην ευρεία εξάπλωση της ψηφιοποίησης δεδομένων διαδραματίζει η μείωση του κόστους ψηφιοποίησης το 1996. Το 1997, η Google παρουσιάζει τη δική της μηχανή αναζήτησης, η οποία είναι και η δημοφιλέστερη παγκοσμίως. Το 1999, ο όρος “Big Data” συναντάται πλέον και σε επιστημονικά άρθρα.

Το 2005, εμφανίζεται το WEB 2.0 και αναπτύσσεται από τον Apache το Hadoop, ένα ανοικτού κώδικα λογισμικό επεξεργασίας Δεδομένων Μεγάλης Κλίμακας. Το 2007, η Apple κυκλοφορεί το iPhone και δημιουργεί μία ισχυρή καταναλωτική αγορά για έξυπνα κινητά τηλέφωνα, ενώ, το 2008, ο αριθμός των συνδεδεμένων συσκευών στο διαδίκτυο ξεπερνά τον παγκόσμιο πληθυσμό. Το 2012, η κυβέρνηση Ομπάμα ανακοινώνει την πρωτοβουλία έρευνας και ανάπτυξης μεγάλων δεδομένων που αποτελείται από 84 προγράμματα σε έξι τομείς. Το 2020, ο ψηφιακός κόσμος κατέχει 40 zettabytes, 57 φορές των συνολικό αριθμό των κόκκων άμμου από όλες τις παραλίες παγκοσμίως. **(Εικόνα 9)**



Εικόνα 9. Απεικόνιση της αποθήκευσης, ανάλυσης και επεξεργασίας της πληροφορίας. [23]

Τα τελευταία 20 έτη, έχει αλλάξει δραματικά ο τρόπος διαχείρισης των δεδομένων μεγάλης κλίμακας. Τα Big Data δημιουργούνται ουσιαστικά από τα πάντα γύρω μας ανά πάσα στιγμή. Κάθε ψηφιακή αλληλεπίδραση με μέσα μαζικής δικτύωσης παράγει δεδομένα, τα οποία συγκεντρώνονται από πολλαπλές πηγές, με τρομακτική ταχύτητα, όγκο και ποικιλία. Το εύρος των δεδομένων που συλλέγονται από επιχειρήσεις σήμερα είναι σχεδόν εξωπραγματικό. Σύμφωνα με την IBM, δημιουργούνται πάνω από 2,5 τετράκις εκατομμύρια bytes δεδομένων ετησίως, ενώ το 90% από αυτά έχει δημιουργηθεί μέσα στα τελευταία δύο έτη. [23] (**Εικόνα 10**)



Εικόνα 10. Οι κύριες πηγές δεδομένων μεγάλης κλίμακας και η ταχεία ανάπτυξή τους. [23]

2.2.3 Πεδίο Εφαρμογής

Η αξιοποίηση των Δεδομένων Μεγάλης Κλίμακας, με τα κατάλληλα μέσα, στην ουσία προσδίδει αξία σε πληροφορίες που υπάρχουν αποθηκευμένες αλλά δεν έχουν ομαδοποιηθεί και δεν έχει κατανοηθεί η αξία τους. Η κατανόηση και ομαδοποίηση του καταναλωτικού κοινού είναι μία από τις κυριότερες χρήσεις των Big Data. Γίνεται ανάλυσή τους, με απώτερο στόχο την καλύτερη κατανόηση των καταναλωτών και των προτιμήσεών τους. Οι περισσότερες επιχειρήσεις επιδιώκουν να εμπλουτίσουν τα δεδομένα τους, αντλώντας πρόσθετες πληροφορίες ακόμα και από τα μέσα κοινωνικής δικτύωσης, ώστε να αποκτήσουν πλήρη εικόνα για κάθε πελάτη. Η διαδικασία αυτή επιτρέπει τη δημιουργία μοντέλων πρόβλεψης της ζήτησης.

Παράλληλα, τα Big Data χρησιμοποιούνται όλο και περισσότερο για τη βελτίωση των επιχειρηματικών διαδικασιών. Ενδεικτικά, μπορεί να χρησιμοποιηθούν για να βελτιστοποιηθεί η διαχείριση των αποθεμάτων, με βάση τις προβλέψεις από τα μέσα κοινωνικής δικτύωσης, τις αναζητήσεις στο διαδίκτυο ή την πρόβλεψη του καιρού. Επίσης, χρησιμοποιούνται κατά κόρον για να καταστήσουν αποδοτικότερη την αλυσίδα ανεφοδιασμού και να βελτιστοποιήσουν τη

διαδρομή παράδοσης, αλλά και σε τμήματα ανθρώπινου δυναμικού, προκειμένου να επισπεύσουν τη διαδικασία των προσλήψεων.

Η υπολογιστική δύναμη των Big Data επέτρεψε την αποκωδικοποίηση του DNA και υπάρχει πρόβλεψη ότι, μελλοντικά, ίσως επιτρέψει την καλύτερη κατανόηση της εξέλιξης των νοσημάτων, βελτιώνοντας την ποιότητα των παρεχόμενων υπηρεσιών υγείας. Καίρια είναι και η συνεισφορά των μεγάλων δεδομένων στην παρακολούθηση και πρόβλεψη των επιδημιών ή πανδημιών. Δεδομένα από ιατρικά αρχεία, σε συνδυασμό με τα μέσα κοινωνικής δικτύωσης, παρέχουν τη δυνατότητα παρακολούθησης της πορείας των κρουσμάτων και του τρόπου διασποράς. Επίσης, μπορούν να συλλεγούν πληροφορίες σχετικά με τη συμπτωματολογία, το χρόνο εκδήλωσης των συμπτωμάτων και της ανταπόκρισης σε φαρμακευτικές αγωγές, χωρίς να εκτίθεται το υγειονομικό προσωπικό σε κίνδυνο. Με αυτόν τον τρόπο, μπορεί, παράλληλα, να κριθεί – έως ένα βαθμό – και η αποτελεσματικότητα μίας θεραπευτικής αντιμετώπισης, αλλά και να δοθούν οδηγίες στους ασθενείς.

Ένας άλλος τομέας, όπου χρησιμοποιούνται ευρέως τα δεδομένα μεγάλης κλίμακας, είναι ο αθλητισμός. Συμβάλλουν στην καταγραφή στατιστικών, όπως οι επιδόσεις των αθλητών, ενώ υπάρχουν ειδικοί μηχανισμοί που ενσωματώνονται στον αθλητικό εξοπλισμό και βοηθούν τον αθλητή να ενισχύσει και να παρακολουθεί τις επιδόσεις του.

Η επιστήμη και η έρευνα επηρεάζονται από τις νέες δυνατότητες που φέρνουν τα Big Data. Για παράδειγμα, το κέντρο δεδομένων του CERN, που είναι το εργαστήριο πυρηνικής φυσικής με το μεγαλύτερο και ισχυρότερο επιταχυντή σωματιδίων παγκοσμίως, έχει 65.000 επεξεργαστές και αναλύει 30 petabytes των δεδομένων.

Τα μεγάλα δεδομένα συμβάλλουν, σε μεγάλο βαθμό, στη βελτίωση της ασφάλειας και τη δυνατότητα επιβολής του νόμου. Διάφορες κυβερνητικές οργανώσεις τα χρησιμοποιούν για να αποτρέψουν τρομοκρατικές ενέργειες ή για την ανίχνευση και την πρόληψη των επιθέσεων στον κυβερνοχώρο. Επίσης, η αστυνομία τα χρησιμοποιεί για να προχωρήσει σε συλλήψεις ή για να προβλέψει κάποια εγκληματική δραστηριότητα. Τέλος, οι τράπεζες και οι εταιρείες πιστωτικών καρτών έχουν τη δυνατότητα να ανιχνεύσουν τις παράνομες συναλλαγές και να τις εμποδίσουν, ειδικά σε περίπτωση κλοπής ή απώλειας των καρτών.

Επιπρόσθετα, χρησιμοποιούνται για τη βελτίωση και την αποδοτικότερη λειτουργία των πόλεων. Με αυτόν τον τρόπο, επιτρέπουν και διευκολύνουν τη ρύθμιση της κυκλοφορίας, με βάση πληροφορίες που αντλούν σε πραγματικό χρόνο σχετικά με την κυκλοφοριακή συμφόρηση ή και τροχαία ατυχήματα. Επισημαίνεται ότι, σήμερα, αρκετές πόλεις καλούνται «έξυπνες» πόλεις.

Τέλος, στις συναλλαγές υψηλής συχνότητας (HFT), τα μεγάλα δεδομένα χρησιμοποιούνται ευρέως. Υπάρχουν ειδικοί αλγόριθμοι που χρησιμοποιούνται για τη λήψη επενδυτικών αποφάσεων. Σήμερα, η πλειοψηφία των συναλλαγών στο χρηματιστήριο πραγματοποιείται

μέσω αλγορίθμων που λαμβάνουν υπόψη όλο και περισσότερο τα δεδομένα των μέσων κοινωνικής δικτύωσης και των ειδησεογραφικών ιστοσελίδων.

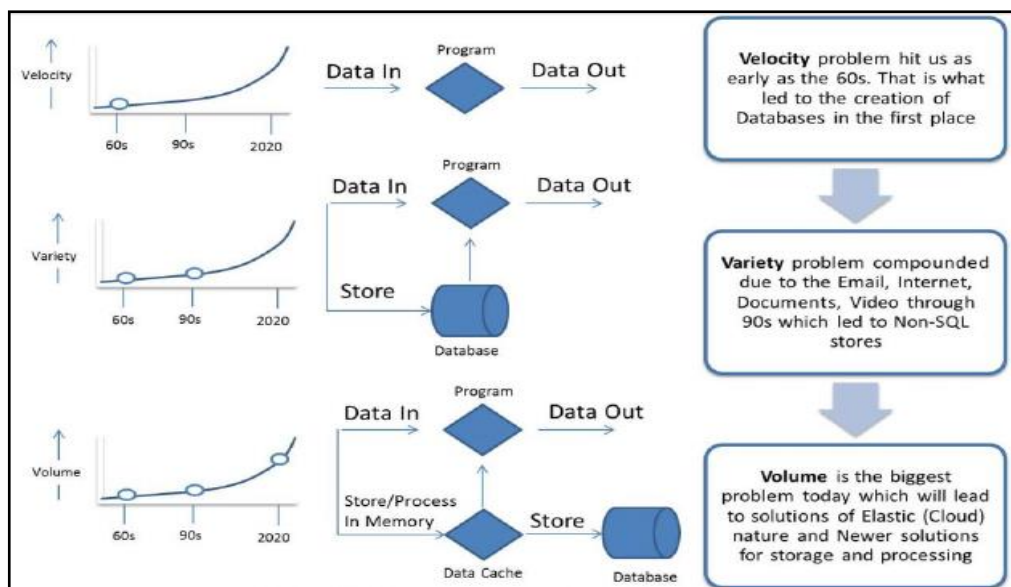
Ωστόσο, τα μεγάλα δεδομένα δεν αφορούν μόνο την αγορά και τις επιχειρήσεις. Αντίθετα, όλοι οι άνθρωποι μπορούν να επωφεληθούν για να βελτιώσουν την καθημερινότητά τους, μέσα από έξυπνες συσκευές.

2.2.4 Χαρακτηριστικά των Δεδομένων Μεγάλης Κλίμακας

Τα Δεδομένα Μεγάλης Κλίμακας χαρακτηρίζονται από τα τέσσερα V, που αναλύονται ακολούθως:

1. Όγκος (Volume)

Αφορά τον όγκο των δεδομένων προς διαχείριση και ανάλυση. Κυμαίνεται σε επίπεδα petabytes (PB) ή και zettabytes (ZB). Αυτό οφείλεται κυρίως στην αύξηση της χρήσης των κινητών συσκευών και των κοινωνικών δικτύων. (Εικόνα 11)



Εικόνα 11. Η εξέλιξη των big data. [23]

2. Ταχύτητα (Velocity)

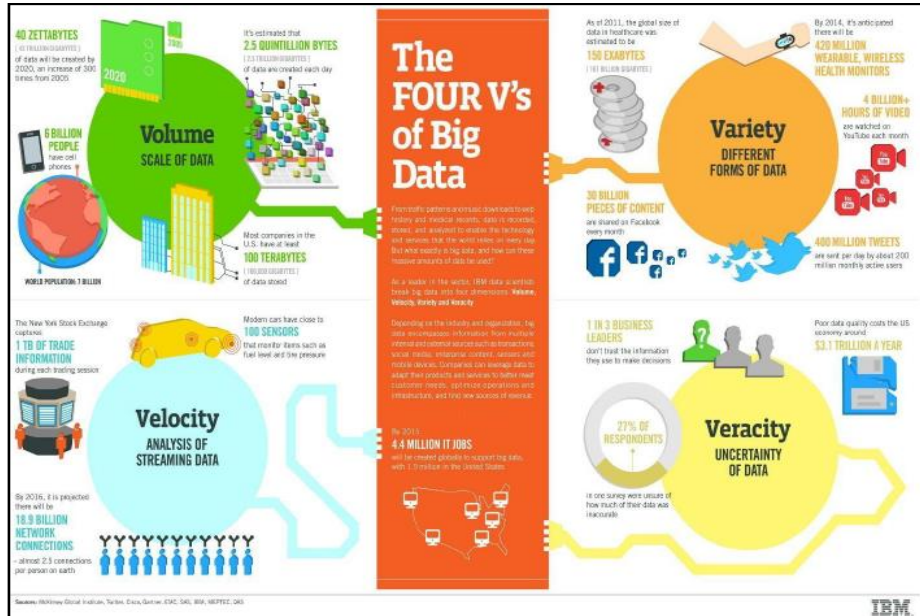
Αφορά τον ταχύτατο ρυθμό εισαγωγής, ανανέωσης, επεξεργασίας και ανάλυσης των δεδομένων. Βασικό χαρακτηριστικό τους αποτελεί η γρήγορη ταχύτητα παραγωγής μεγάλου όγκου πληροφοριών σε πραγματικό χρόνο. Ανάλογα με τη φύση των δεδομένων, οι τυπικές απαιτήσεις επεξεργασίας κυμαίνονται από kilobytes/sec έως terabytes/sec.

3. Ποικιλομορφία (Variety)

Αναφέρεται στην ετερογενή φύση των δεδομένων, επειδή συλλέγονται από διάφορες πηγές και δεν εμπίπτουν στην ίδια κατηγοριοποίηση. Απαιτεί νέες συνδυαστικές μεθόδους επεξεργασίας.

4. Αξιοπιστία (Veracity)

Αναφέρεται στην ποιότητα των δεδομένων. Αυτό οφείλεται στο γεγονός ότι υπάρχει μεγάλος αριθμός πηγών παραγωγής και συλλογής δεδομένων, οπότε τίθεται ζήτημα προέλευσης, ακρίβειας και πληρότητας των δεδομένων. [20] (*Εικόνα 12*)



Εικόνα 12. Τα τέσσερα V των Δεδομένων Μεγάλης Κλίμακας. [23]

2.2.5 Αξία Δεδομένων Μεγάλης Κλίμακας

Τα ψηφιακά δεδομένα αποτελούν σημαντική πηγή εσόδων για την οικονομική ανάπτυξη, την ανταγωνιστικότητα, την καινοτομία, τη δημιουργία θέσεων απασχόλησης και την κοινωνική πρόοδο. [20], [24], [25], [26] Η αξία της οικονομίας δεδομένων της Ευρωπαϊκής Ένωσης (Ε.Ε.) ήταν μεγαλύτερη από 285 δισ. ευρώ το 2015, αντιπροσωπεύοντας πάνω από το 1,94% του Α.Ε.Π. (Ακαθάριστου Εγχώριου Προϊόντος) της Ε.Ε., ενώ, το 2020, 739 δισ. ευρώ, αντιπροσωπεύοντας το 4% του συνολικού Α.Ε.Π. της Ε.Ε. [20], [27] (*Εικόνα 13*)



Εικόνα 13. Αξία της οικονομίας δεδομένων στην Ε.Ε. [20]

2.2.6 Πεδία εφαρμογής Δεδομένων Μεγάλης Κλίμακας

Τα πεδία εφαρμογής των Δεδομένων Μεγάλης Κλίμακας παρουσιάζονται συνοπτικά στον **Πίνακα 2** και την **Εικόνα 14**.

Αποτελεσματικότητα και διοικητική μεταρρύθμιση
Ασφάλεια και καταπολέμηση του εγκλήματος
Υποδομές
Οικονομία και εργασία
Εκσυγχρονισμός της νομοθεσίας
Υπηρεσίες πολιτών και επιχειρήσεων

Πίνακας 2. Πεδία εφαρμογής των Δεδομένων Μεγάλης Κλίμακας. [20]



Εικόνα 14. Πεδία εφαρμογής των Δεδομένων Μεγάλης Κλίμακας. [23]

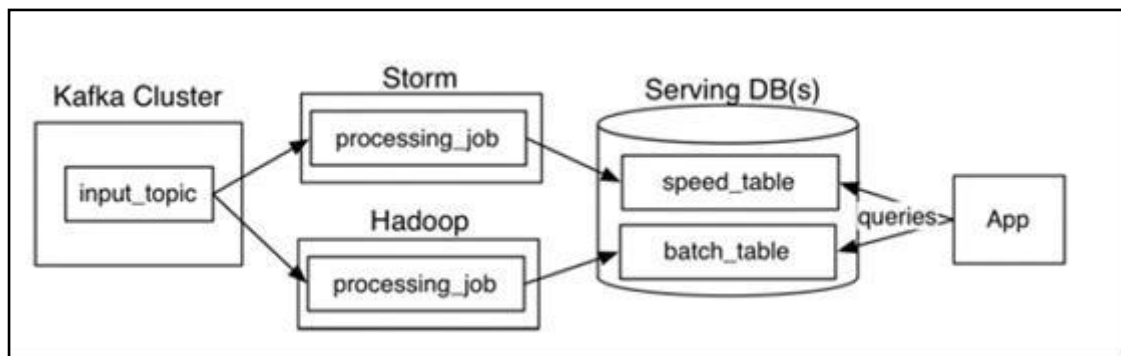
2.3 Lambda architecture

Ο ρυθμός αύξησης των δεδομένων είναι πολύ μεγάλος. Πέρα από την ανάπτυξή τους σε όγκο, τα Big Data εξελίσσονται και στον τρόπο επεξεργασίας τους. Για αυτόν το λόγο, τα πληροφοριακά συστήματα δεδομένων, που αποθηκεύουν και επεξεργάζονται την πληροφορία σε μεγάλες ποσότητες, το γνωστό batch processing, δεν επαρκεί πλέον. Για την επίλυση του προβλήματος αυτού, εισήχθη η έννοια της ανάλυσης σε πραγματικό χρόνο, δηλαδή να γίνονται ενέργειες για την άμεση εξόρυξη της πληροφορίας. Για την ανάλυση αυτών των «ροών πληροφορίας» χρησιμοποιείται ευρύτατα ο όρος stream processing.

Παρόλο που το batch processing έχει ξεπεραστεί, χαρακτηρίζει τις περισσότερες Big Data αρχιτεκτονικές και εφαρμόζεται εκτενέστατα από πολλά συστήματα. Άρα, ήταν πλέον επιτακτική η ανάγκη να δημιουργηθεί μία αποδοτική αρχιτεκτονική που θα χρησιμοποιεί και τις δύο μεθόδους επεξεργασίας, batch και streaming. Για το «πάντρεμα» του batch με το stream processing, μίλησε για πρώτη φορά ο Nathan Marz, το 2011, μέσω της δημιουργίας του Lambda Architecture. [28], [29]

2.3.1 Δομή lambda architecture

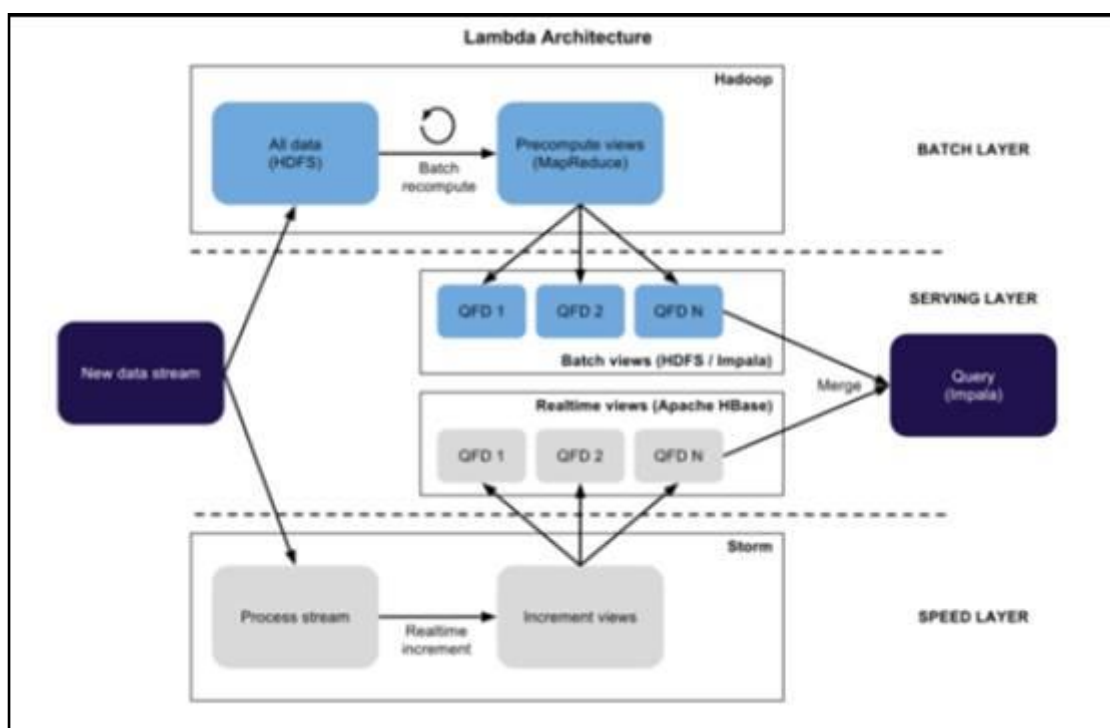
Αυτή η αρχιτεκτονική αποτελεί υβριδικό μοντέλο, το οποίο αποτελείται από ένα τμήμα που κάνει αναλύσεις τύπου batch (batch layer) πάνω σε όλα τα διαθέσιμα δεδομένα, ένα υποσύστημα επεξεργασίας δεδομένων που μόλις έχουν φτάσει μέσα στο σύστημα (speed layer) και ένα τελευταίο τμήμα, το service layer, που πρακτικά ενώνει τα δύο προηγούμενα τμήματα, προσφέροντας δυνατότητες αναζήτησης και συνδυασμού των αποτελεσμάτων των batch και speed layers. [28] (Εικόνα 15)



Εικόνα 15. Παράδειγμα συστήματος που χρησιμοποιεί Lambda Architecture. [28]

Η αρχιτεκτονική Lambda αποτελεί μία αρχιτεκτονική επεξεργασίας δεδομένων που έχει σχεδιαστεί για να αντιμετωπίζει τεράστιες ποσότητες δομημένων ή μη δομημένων δεδομένων, εκμεταλλευόμενη και τις δύο μεθόδους επεξεργασίας, batch και streaming. Αυτή η προσέγγιση στην αρχιτεκτονική προσπαθεί να εξισορροπήσει την καθυστέρηση, την απόδοση και την ανοχή στα σφάλματα, χρησιμοποιώντας τη μαζική επεξεργασία για την παροχή ολοκληρωμένης και ακριβούς εικόνας των δεδομένων. Παράλληλα, πραγματοποιεί ανάλυση ροής σε πραγματικό χρόνο, για να παρέχει εικόνες των διαδικτυακών δεδομένων. Οι δύο μέθοδοι επεξεργασίας ενώνονται πριν από την παρουσίαση, παρέχοντας στο χρήστη μία ολοκληρωμένη εικόνα των δεδομένων. Η άνοδος της χρήσης της αρχιτεκτονικής Lambda συσχετίζεται με την εμφάνιση και την εξέλιξη των μεγάλων δεδομένων, την ανάλυση των

δεδομένων σε πραγματικό χρόνο και την προσπάθεια μετριασμού των καθυστερήσεων του MapReduce. [30] (*Εικόνα 16*)



Εικόνα 16. Σχηματική απεικόνιση της αρχιτεκτονικής Lambda. [30]

Επισημαίνεται ότι, στην *Εικόνα 16*, απεικονίζεται σχηματικά ένα σύστημα που βασίζεται στην αρχιτεκτονική Lambda, καθώς και ορισμένες τεχνολογίες που μπορούν να χρησιμοποιηθούν κατά την εφαρμογή του. Σε γενικές γραμμές, η αρχιτεκτονική Lambda αποτελείται από τρία επίπεδα: το επίπεδο παραγωγής (batch layer), το επίπεδο εξυπηρέτησης (serving layer) και το επίπεδο ταχύτητας (speed layer). [30]

2.3.1.1 Πηγές δεδομένων

Τα δεδομένα μπορούν να ληφθούν από μία ποικιλία πηγών, οι οποίες μπορούν, ακολούθως, να συμπεριληφθούν στην αρχιτεκτονική Lambda για ανάλυση. Πολλές φορές, αποτελούν πηγή ροής, όπως το Apache Kafka, η οποία δεν είναι η αρχική πηγή δεδομένων καθατή, αλλά είναι ένα ενδιάμεσο αρχείο δεδομένων, το οποίο μπορεί να κρατήσει δεδομένα για να εξυπηρετήσει τόσο το επίπεδο παραγωγής, όσο και το επίπεδο ταχύτητας της Lambda Architecture. Τα δεδομένα παραδίδονται ταυτόχρονα τόσο στο batch επίπεδο, όσο και στο speed επίπεδο, προκειμένου να καταστεί δυνατή μία παράλληλη αναζήτηση. [31]

2.3.1.2 *Batch layer*

Το batch layer περιέχει το αμετάβλητο, συνεχώς αναπτυσσόμενο, κύριο σύνολο δεδομένων, αποθηκευμένο και κατανεμημένο σε ένα σύστημα αρχείων, όπως το HDFS (Hadoop Distributed File System). [30], [32] Με τέτοια επεξεργασία (MapReduce), οι προβολές υπολογίζονται με βάση αυτό το ακατέργαστο σύνολο δεδομένων. Μία τεχνολογία, όπως το Apache Hadoop, είναι ιδανική για να εφαρμοστεί ως batch layer, με οικονομικά αποδοτικό τρόπο. Αυτό το επίπεδο αποθηκεύει όλα τα δεδομένα που εισέρχονται στο σύστημα ως προβολές και τα προετοιμάζει για αναζήτηση. Τα δεδομένα εισόδου αποθηκεύονται σε ένα μοντέλο που μοιάζει με μία σειρά μεταβολών – ενημερώσεων που έγιναν σε ένα σύστημα εγγραφής, παρόμοιο με την έξοδο ενός συστήματος αλλαγής δεδομένων (Continuous Damping Control – CDC). Συχνά, πρόκειται απλώς για ένα αρχείο με μορφή τιμών διαχωρισμένων με κόμμα (CSV – Comma-Separated Values). Τα δεδομένα αντιμετωπίζονται ως αμετάβλητα και προσαρτώνται μόνο για να διασφαλιστεί μία αξιόπιστη ιστορική εγγραφή όλων των εισερχόμενων δεδομένων. [31]

2.3.1.3 *Serving layer*

Η κύρια λειτουργία του επιπέδου εξυπηρέτησης (serving layer) είναι να φορτώσει και να εκθέσει τις προβολές, ώστε αυτές να μπορούν, στη συνέχεια, να διερευνηθούν. Αυτό το αρχείο δεδομένων του επιπέδου εξυπηρέτησης δεν απαιτεί τυχαίες εγγραφές. Αντίθετα, πρέπει να υποστηρίζει μαζικές ενημερώσεις και τυχαίες αναγνώσεις. Ιδανικοί υποψήφιοι για αυτό το επίπεδο, δεδομένου ότι απαιτούνται εξαιρετικά απλές εφαρμογές, είναι οι ElephantDB, Impala ή Voldemort. [30], [33] Πρακτικά, αυτό το επίπεδο τοποθετεί σε ευρετήριο σταδιακά τις τελευταίες προβολές. Επίσης, μπορεί να επαναπροσδιορίσει όλα τα δεδομένα για να διορθώσει ένα σφάλμα κωδικοποίησης ή να δημιουργήσει διαφορετικά ευρετήρια για διαφορετικές περιπτώσεις χρήσης. Η βασική απαίτηση σε αυτό το επίπεδο είναι ότι η επεξεργασία γίνεται με εξαιρετικά παραλληλισμένο τρόπο, προκειμένου να ελαχιστοποιηθεί ο χρόνος για την αναζήτηση του συνόλου δεδομένων. Ενώ εκτελείται μία αναζήτηση, τα δεδομένα που εισάγονται πρόσφατα θα βρίσκονται στην ουρά για αναζήτηση στην επόμενη διαδικασία αναζήτησης. [31]

2.3.1.4 *Speed layer*

Το επίπεδο ταχύτητας (speed layer) ασχολείται μόνο με νέα δεδομένα, τα οποία δεν ελήφθησαν υπόψη στο επίπεδο της εξυπηρέτησης. Πρακτικά, αντισταθμίζει τις ενημερώσεις του επιπέδου εξυπηρέτησης, λόγω του υψηλού λανθάνοντος χρόνου που παρουσιάζουν, με την υπέρθεσή του πάνω από συστήματα επεξεργασίας ροής (Storm, S4, Spark) και τυχαία αρχεία δεδομένων ανάγνωσης/εγγραφής για τον υπολογισμό των προβολών σε πραγματικό χρόνο. [30], [34], [35],

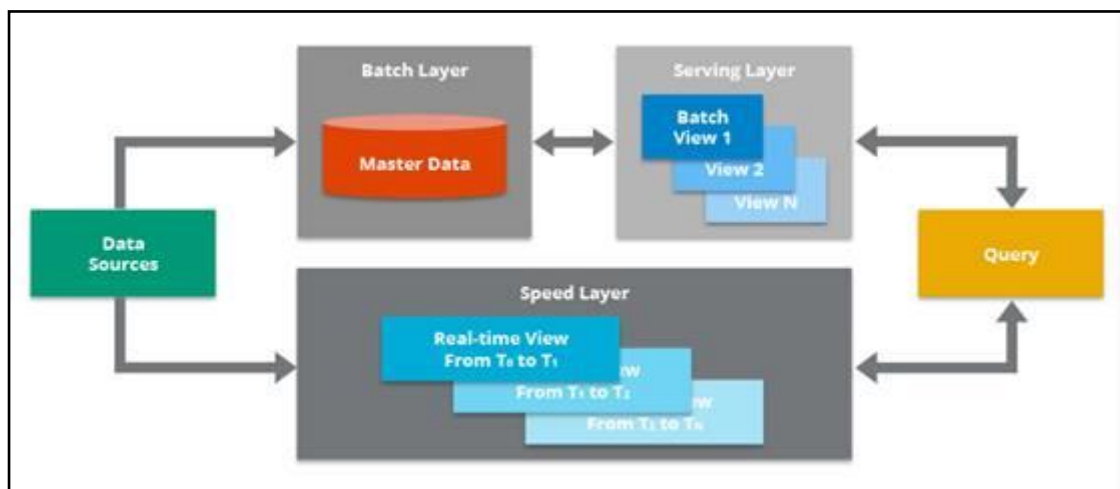
[36] Αυτές οι προβολές παραμένουν έγκυρες, ώσπου τα δεδομένα να παραδοθούν και να υποβληθούν για επεξεργασία από τα επίπεδα batch και serving.

Αυτό το επίπεδο συμπληρώνει το επίπεδο εξυπηρέτησης με δημιουργία ευρετηρίου των πιο πρόσφατα προστιθέμενων δεδομένων, που δεν έχουν μπει σε ευρετήρια από το επίπεδο εξυπηρέτησης. Πρακτικά, περιλαμβάνει τα δεδομένα που το επίπεδο εξυπηρέτησης θέτει σε ευρετήριο για αναζήτηση, καθώς και νέα δεδομένα που έφτασαν μετά την έναρξη της τρέχουσας αναζήτησης. Επισημαίνεται ότι υπάρχει καθυστέρηση μεταξύ του χρόνου προσθήκης των τελευταίων δεδομένων στο σύστημα και του χρόνου που τα τελευταία δεδομένα θα είναι διαθέσιμα για αναζήτηση. Όταν πρωτοεμφανίστηκε η αρχιτεκτονική Lambda, το Apache Storm ήταν ένας κορυφαίος κινητήρας επεξεργασίας ροής που χρησιμοποιήθηκε σε εφαρμογές, αλλά άλλες τεχνολογίες έχουν αποκτήσει έκτοτε μεγαλύτερη δημοτικότητα ως υποψήφιοι για αυτό το στοιχείο (όπως το Hazelcast Jet, το Apache Flink και το Apache Spark Streaming). [31]

Προκειμένου να ληφθεί ένα ολοκληρωμένο αποτέλεσμα, πρέπει οι προβολές batch αλλά και οι προβολές σε πραγματικό χρόνο να διερευνηθούν και τα αποτελέσματά τους να συγχρονιστούν. [30]

2.3.1.5 Αναζήτηση (query)

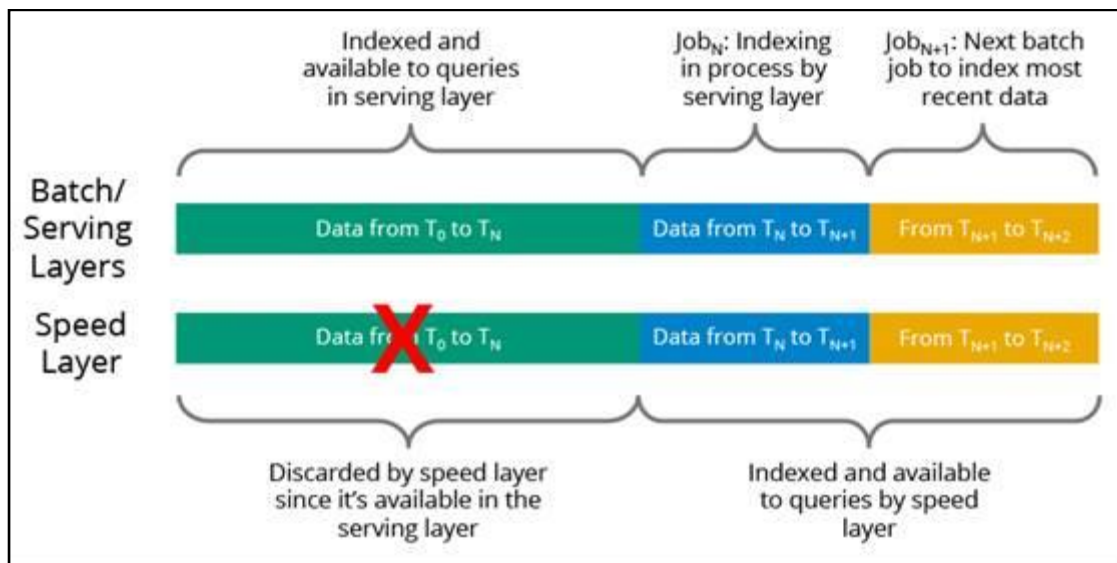
Αυτό το στοιχείο είναι υπεύθυνο για την υποβολή ερωτημάτων τελικού χρήστη τόσο σε επίπεδο εξυπηρέτησης, όσο και σε επίπεδο ταχύτητας και στο συγχρονισμό των αποτελεσμάτων. Αυτό δίνει στους τελικούς χρήστες μία πλήρη αναζήτηση για όλα τα δεδομένα, συμπεριλαμβανομένων των δεδομένων που προστέθηκαν πρόσφατα, για να παρέχουν ένα σύστημα ανάλυσης σχεδόν σε πραγματικό χρόνο. [31] (*Εικόνα 17*)



Εικόνα 17. Τα επίπεδα της αρχιτεκτονικής lambda. [31]

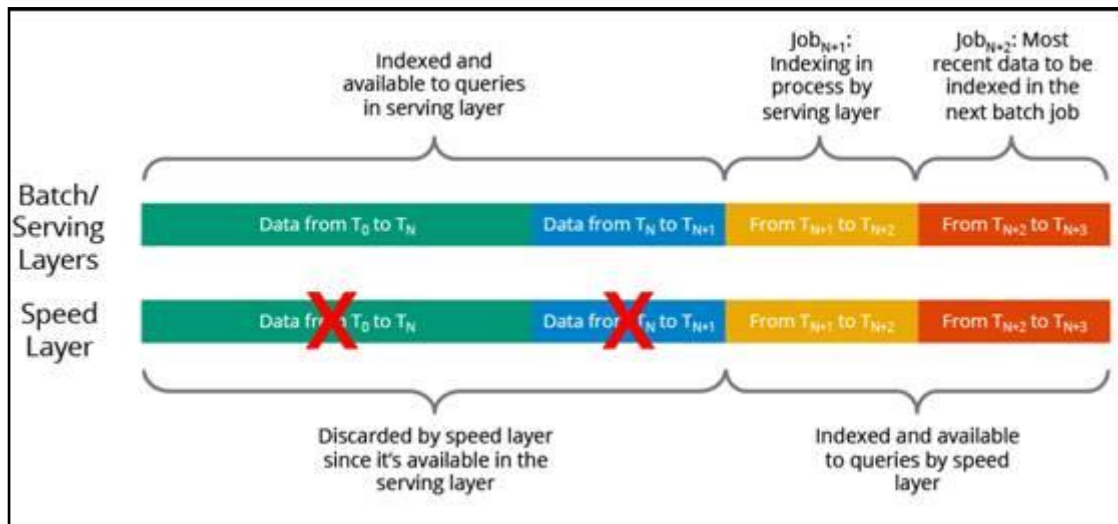
2.3.2 Τρόπος λειτουργίας lambda architecture

Τα επίπεδα batch/serving συνεχίζουν να τοποθετούν σε ευρετήρια όλα τα εισερχόμενα δεδομένα. Δεδομένου ότι το επίπεδο batch απαιτεί χρόνο, το επίπεδο speed συμπληρώνει τα επίπεδα batch/serving, με δημιουργία ευρετηρίου όλων των νέων, μη ταξινομημένων δεδομένων σε σχεδόν πραγματικό χρόνο. Αυτό παρέχει μία μεγάλη και συνεπή βάση δεδομένων στα επίπεδα batch/serving, που μπορεί να αναπαρασταθεί ανά πάσα στιγμή, μαζί με ένα μικρότερο ευρετήριο που περιέχει τα περισσότερα πρόσφατα δεδομένα. [31] (Εικόνα 18)



Εικόνα 18. Τα δεδομένα ταξινομούνται σε ευρετήρια ταυτόχρονα τόσο από το serving επίπεδο, όσο και από το speed επίπεδο. [31]

Μόλις ολοκληρωθεί μία εργασία δημιουργίας ευρετηρίου, τα πρόσφατα δεδομένα είναι διαθέσιμα για αναζήτηση. Συνεπώς, δεν κρίνεται πλέον αναγκαίο το αντίγραφο του επιπέδου ταχύτητας και, για αυτό το λόγο, διαγράφεται. Στη συνέχεια, το επίπεδο εξυπηρέτησης αρχίζει να οργανώνει σε ευρετήρια τα πιο πρόσφατα δεδομένα στο σύστημα που δεν είχαν ακόμα οργανωθεί από αυτό το επίπεδο. Αυτή η συνεχιζόμενη παράδοση μεταξύ των επιπέδων speed και batch/serving διασφαλίζει ότι όλα τα δεδομένα είναι έτοιμα για αναζήτηση και ότι ο λανθάνων χρόνος για τη διαθεσιμότητα των δεδομένων είναι χαμηλός. [31] (Εικόνα 19)



Εικόνα 19. Όταν το επίπεδο εξυπηρέτησης ολοκληρώσει μία εργασία, μετακινείται στο επόμενο επίπεδο και το επίπεδο ταχύτητας διαγράφει το αντίγραφο των δεδομένων που μόλις προστέθηκαν στο επίπεδο εξυπηρέτησης. [31]

2.3.3 Πλεονεκτήματα και Οφέλη *lambda architecture*

Η αρχιτεκτονική *lambda* προσπαθεί να εξισορροπήσει τις ανησυχίες γύρω από το λανθάνοντα χρόνο, τη συνέπεια των δεδομένων, την επεκτασιμότητα, την ανοχή σφαλμάτων και την ανοχή σε ανθρώπινα σφάλματα.

1. Λανθάνων χρόνος

Τα ακατέργαστα δεδομένα ταξινομούνται σε ευρετήρια στο επίπεδο εξυπηρέτησης, έτσι ώστε οι τελικοί χρήστες να μπορούν να πραγματοποιούν αναζητήσεις και να αναλύουν όλα τα δεδομένα. Δεδομένου ότι η μαζική δημιουργία ευρετηρίου απαιτεί σημαντικό χρονικό διάστημα, τείνει να υπάρχουν, για ένα σχετικά μεγάλο χρονικό παράθυρο, δεδομένα που δεν είναι προσωρινά διαθέσιμα στους τελικούς χρήστες για ανάλυση. Το επίπεδο ταχύτητας χρησιμοποιεί τεχνολογίες επεξεργασίας ροής για άμεση καταχώρηση των πρόσφατων δεδομένων σε ευρετήρια, μειώνοντας το χρονικό παράθυρο δεδομένων που δεν είναι διαθέσιμα για αναζήτηση και ανάλυση. Πρακτικά, συμβάλλει στη μείωση του λανθάνοντος χρόνου (δηλαδή του χρόνου αναμονής για τη διάθεση δεδομένων για ανάλυση), ο οποίος είναι εγγενές χαρακτηριστικό των επιπέδων *batch/serving*. [31]

2. Συνοχή δεδομένων

Μία βασική ιδέα πίσω από την αρχιτεκτονική *lambda* είναι ότι εξαλείφει τον κίνδυνο ασυνέπειας δεδομένων που παρατηρείται συχνά σε δομημένα συστήματα. Σε μία δομημένη βάση δεδομένων, ένα αντίγραφο των δεδομένων ενδέχεται να αντικατοπτρίζει την ενημερωμένη τιμή, αλλά ένα άλλο αντίγραφο ενδέχεται να διατηρεί την προηγούμενη τιμή.

Στην αρχιτεκτονική lambda, η διαδικασία δημιουργίας ευρετηρίου μπορεί να διασφαλίσει ότι τα δεδομένα αντικατοπτρίζουν την τελευταία κατάσταση τόσο σε batch επίπεδο, όσο και σε speed επίπεδο. [31]

3. Επεκτασιμότητα

Η Αρχιτεκτονική lambda δεν καθορίζει τις ακριβείς τεχνολογίες που θα χρησιμοποιηθούν, αλλά βασίζεται σε δομημένες τεχνολογίες επέκτασης που μπορούν να επεκταθούν με την απλή προσθήκη περισσότερων κόμβων. Αυτό της επιτρέπει να χρησιμοποιηθεί, ανεξάρτητα από το πόσα δεδομένα χρειάζονται για επεξεργασία. [31]

4. Ανοχή σε σφάλματα

Η αρχιτεκτονική lambda βασίζεται σε δομημένα συστήματα που υποστηρίζουν ανοχή σφαλμάτων. Σε περίπτωση σφάλματος υλικού, υπάρχουν άλλοι κόμβοι για τη συνέχιση του φορτίου εργασίας. [31]

5. Ανοχή σε ανθρώπινα λάθη

Δεδομένου ότι τα μη επεξεργασμένα δεδομένα αποθηκεύονται, λειτουργεί ως σύστημα εγγραφής για τα δεδομένα προς ανάλυση και όλα τα ευρετήρια μπορούν να δημιουργηθούν ξανά από αυτό το σύνολο δεδομένων. Αυτό σημαίνει ότι εάν υπάρχουν σφάλματα ή παραλείψεις στον κώδικα για τη δημιουργία του ευρετηρίου, ο κώδικας μπορεί να ενημερωθεί και, στη συνέχεια, να εκτελεστεί ξανά για να επανασυνδεθούν όλα τα δεδομένα. [30], [31]

Τα κύρια οφέλη που προκύπτουν από τη χρήση της αρχιτεκτονικής lambda ότι δεν απαιτείται η εγκατάσταση, η συντήρηση ή η διαχείριση οποιουδήποτε λογισμικού. Η εφαρμογή της μπορεί να κλιμακωθεί αυτόματα ή να κλιμακωθεί με την προσαρμογή της χωρητικότητάς της. Επίσης, διαθέτει αυτοματοποιημένη υψηλή διαθεσιμότητα, η οποία αναφέρεται στο γεγονός ότι οι εφαρμογές χωρίς διακομιστές έχουν ήδη ενσωματωμένη διαθεσιμότητα και ανοχή σφαλμάτων. Το συγκεκριμένο χαρακτηριστικό αποτελεί εγγύηση ότι όλα τα αιτήματα θα λάβουν απάντηση σχετικά με το κατά πόσο ήταν επιτυχημένα ή όχι. Τέλος, προσφέρει, επιχειρηματική ευελιξία, καθώς βοηθάει τις επιχειρήσεις να ανταποκριθούν, σε πραγματικό χρόνο, στις μεταβαλλόμενες ανάγκες των επιχειρήσεων και της αγοράς. [37]

Το μόνο μειονέκτημα που έχει αποδοθεί στην αρχιτεκτονική lambda είναι ότι έχει επικριθεί ως υπερβολικά περίπλοκη. Κάθε επίπεδο απαιτεί τη δική του βάση δεδομένων και οι βάσεις πρέπει να διατηρούνται συγχρονισμένες, προκειμένου να διασφαλίζουν συνεπή και ακριβή αποτελέσματα. [31] Όταν δεν υπάρχει κοινό εργαλείο για την πραγματοποίηση εφαρμογών τύπου batch και stream, η ιδέα πρέπει να υλοποιηθεί δύο φορές, μία με τις τεχνολογίες του batch layer και μία με αυτές του speed layer. Επίσης, το serving layer πρέπει να ρυθμιστεί σωστά, με τρόπο που να δέχεται και να επεξεργάζεται ορθά τύπους και δομές δεδομένων από

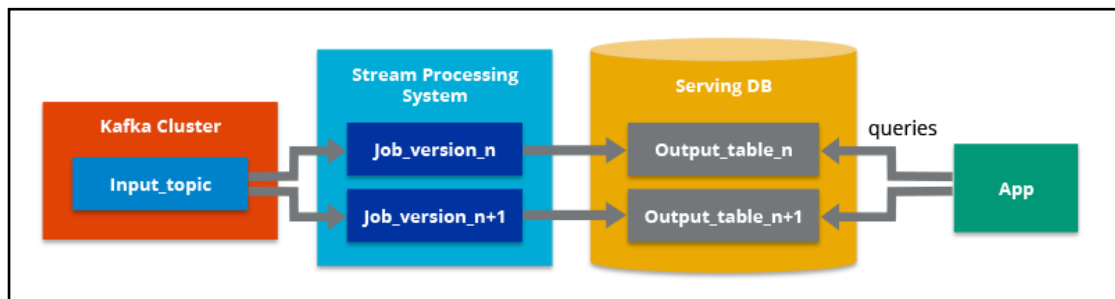
διαφορετικές πηγές. Όλα αυτά προκαλούν μεγάλο προγραμματιστικό φόρτο, τόσο όσον αφορά τη δημιουργία, όσο και τη συντήρηση. [28]

2.3.4 Προκλήσεις της αρχιτεκτονικής lambda

Η αρχιτεκτονική lambda μπορεί να θεωρηθεί πολύ περίπλοκη. Οι διαχειριστές πρέπει συνήθως να διατηρούν δύο ξεχωριστές βάσεις δεδομένων για επίπεδα batch και streaming. Μία τέτοια αναγκαιότητα μπορεί να δυσκολέψει τον εντοπισμό σφαλμάτων. [37]

2.3.5 Kappa architecture

Η αρχιτεκτονική Kappa αποτελεί μία αρχιτεκτονική λογισμικού που χρησιμοποιείται για την επεξεργασία δεδομένων ροής. Η βασική προϋπόθεση, πίσω από την Kappa Architecture, είναι ότι μπορεί να εκτελεστεί επεξεργασία των δεδομένων σε πραγματικό χρόνο. Βασίζεται σε μία αρχιτεκτονική ροής, στην οποία μία εισερχόμενη σειρά δεδομένων αποθηκεύεται, για πρώτη φορά, σε μία μηχανή ανταλλαγής μηνυμάτων, όπως η Apache Kafka. Από εκεί, μία μηχανή επεξεργασίας ροής θα προσπελάσει τα δεδομένα και θα τα μετατρέψει σε αναλύσιμη μορφή και, στη συνέχεια, θα τα αποθηκεύσει σε μία βάση δεδομένων για τους τελικούς χρήστες. (Εικόνα 20)



Εικόνα 20. Δομή της αρχιτεκτονικής kappa. [38]

Η αρχιτεκτονική Kappa υποστηρίζει ανάλυση των δεδομένων σε πραγματικό χρόνο, όταν τα δεδομένα διαβάζονται και μετασχηματίζονται αμέσως μετά την εισαγωγή τους στη μηχανή ανταλλαγής μηνυμάτων. Αυτό καθιστά τα πρόσφατα δεδομένα γρήγορα διαθέσιμα για ερωτήματα τελικών χρηστών. Υποστηρίζει, επίσης, ανάλυση των δεδομένων από το ιστορικό, διαβάζοντας τα αποθηκευμένα δεδομένα ροής από τη μηχανή ανταλλαγής μηνυμάτων, σε μεταγενέστερο χρόνο, με μαζικό τρόπο, προκειμένου να δημιουργήσει περαιτέρω αποτελέσματα για περισσότερους τύπους ανάλυσης.

Η Kappa Architecture θεωρείται απλούστερη εναλλακτική λύση έναντι της Lambda Architecture, καθώς χρησιμοποιεί την ίδια τεχνολογία για να χειριστεί τόσο την επεξεργασία

των δεδομένων σε πραγματικό χρόνο, όσο και την επεξεργασία του ιστορικού των δεδομένων. Και οι δύο αρχιτεκτονικές συνεπάγονται την αποθήκευση δεδομένων για την ενεργοποίηση της ανάλυσης δεδομένων μεγάλης κλίμακας. Και οι δύο αρχιτεκτονικές είναι επίσης χρήσιμες για την αντιμετώπιση της «ανοχής ανθρώπινων σφαλμάτων». [38]

2.3.6 Διαφορές μεταξύ αρχιτεκτονικών *kappa* και *lambda*

Η αρχιτεκτονική *kappa* είναι ένα άλλο μοντέλο σχεδιασμού που μπορεί να συναντήσει κανείς στην εξερεύνηση της αρχιτεκτονικής *lambda*. Η αρχιτεκτονική *kappa* είναι παρόμοια με την αρχιτεκτονική *lambda* χωρίς να υπάρχουν ξεχωριστές τεχνολογίες για το batch επίπεδο. Αντίθετα, όλα τα δεδομένα δρομολογούνται απλά μέσω ενός δικτύου επεξεργασίας ροής. Όλα τα δεδομένα αποθηκεύονται σε ένα κατάλογο δεδομένων (όπως το Apache Kafka) και, όταν απαιτείται επανασύνδεση, τα δεδομένα διαβάζονται εκ νέου από αυτήν την πηγή.

Αφορά μία απλοποιημένη προσέγγιση, καθώς απαιτεί μόνο μία βάση δεδομένων. Ωστόσο, οι βάσεις δεδομένων της αρχιτεκτονικής *kappa* δε μπορούν να αποθηκεύσουν ολόκληρο το ιστορικό των δεδομένων. Οι τεχνολογικές καινοτομίες καταρρίπτουν αυτόν τον περιορισμό, έτσι ώστε πολύ μεγαλύτερα σύνολα δεδομένων να μπορούν να αποθηκευτούν και μέσω της αρχιτεκτονικής *kappa*. Με αυτόν τον τρόπο, προσφέρεται η δυνατότητα επεξεργασίας εξαιρετικά μεγάλων συνόλων δεδομένων με οικονομικά αποδοτικό τρόπο, ενώ, ταυτόχρονα, διατηρείται η απλότητα της χρήσης μόνο μίας μηχανής επεξεργασίας. [31]

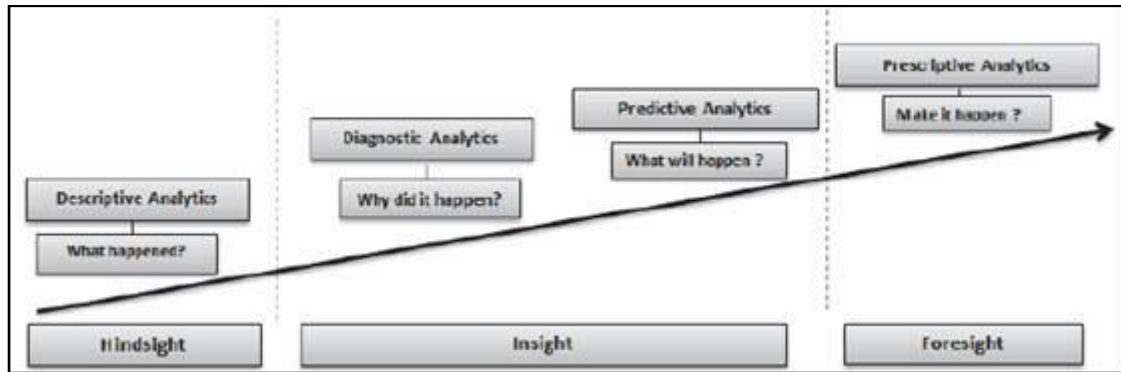
2.4 Descriptive Analytics

2.4.1 Ορισμός

Ο όρος “Big Data Analytics” ή «Αναλυτική Μεγάλων Δεδομένων» περιγράφει μία διαδικασία μετασηματισμού ακατέργαστων δεδομένων σε πληροφορία, δηλαδή συνοψίζει την ανάγκη να εξαχθούν πολύτιμα συμπεράσματα και προβλέψεις από τα δεδομένα, στοχεύοντας στην υποβοήθηση της λήψης αποφάσεων και τη βελτίωση των διαδικασιών των επιμέρους οργανισμών. [39], [40] Η λειτουργία τους βασίζεται στο να αποκαλύψουν κρυφά μοτίβα, να εντοπίσουν άγνωστες συσχετίσεις, να κατανοήσουν τις τάσεις της αγοράς, τις προτιμήσεις των πελατών και να αντλήσουν άλλες χρήσιμες επιχειρηματικές πληροφορίες. [39], [41]

Τα Big Data Analytics περιλαμβάνουν την ενσωμάτωση ετερογενών δεδομένων, τον έλεγχο της ποιότητας των δεδομένων, την ανάλυση, τη μοντελοποίηση, την ερμηνεία και την εξακρίβωση. [39], [42], [43] Η χρησιμοποίησή τους επιτρέπει την ανάλυση μεγάλων συνόλων δεδομένων, την ταυτοποίηση συστάδων, τη συσχέτιση μεταξύ συνόλων δεδομένων και την ανάπτυξη προγνωστικών μοντέλων με τη χρήση τεχνικών εξόρυξης δεδομένων. [39], [44]

Τα Big Data Analytics μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες: Descriptive (Περιγραφική Αναλυτική), Diagnostic (Διαγνωστική Αναλυτική), Predictive (Προγνωστική Αναλυτική) και Prescriptive (Καθοδηγητική Αναλυτική. [39], [40], [41], [45] (*Εικόνα 21*)



Εικόνα 21. Κατηγορίες Big Data Analytics (Περιγραφική, Διαγνωστική, Προγνωστική & Καθοδηγητική Αναλυτική). [39]

Η περιγραφική ανάλυση (Descriptive Analytics) αποτελεί στατιστική μέθοδο, η οποία χρησιμοποιείται για την αναζήτηση και τη σύνοψη δεδομένων, προκειμένου να προσδιοριστούν μοτίβα ή έννοιες. [46] Δηλαδή συνίσταται στην περιγραφή της υφιστάμενης κατάστασης και συμβάλλει στη σκιαγράφηση της εικόνας σχετικά με τις προηγούμενες επιδόσεις, αξιοποιώντας παλαιότερα δεδομένα. Η χρησιμοποίηση αυτής της ανάλυσης αποτελεί τη βάση, επί της οποίας θα δομηθεί η ενδεδειγμένη προσέγγιση για την επίτευξη του στόχου. [39], [40], [41] Η περιγραφική αναλυτική είναι επίσης γνωστή και ως μάθηση χωρίς επίβλεψη. [39], [47]

Ειδικότερα, συνοψίζει τι συνέβη και ποιος είναι ο αντίκτυπος μίας παραμέτρου ή ενός γεγονότος. Αποτελεί το απλούστερο επίπεδο κατανόησης και χρήσης. Ουσιαστικά, περιγράφει τα δεδομένα χωρίς να συνοδεύονται από κάποια ανάλυση, διερεύνηση, διευκρίνιση ή συσχέτισμό μεταξύ τους. Οι περιγραφικές αναλύσεις συγκεντρώνουν, κατηγοριοποιούν και ταξινομούν τα δεδομένα, τα οποία μετατρέπονται σε πολύτιμες πληροφορίες, προκειμένου να βοηθήσουν στην κατανόηση και ανάλυση των αποφάσεων, των επιδόσεων και των αποτελεσμάτων. [39]

2.4.2 Boxplot

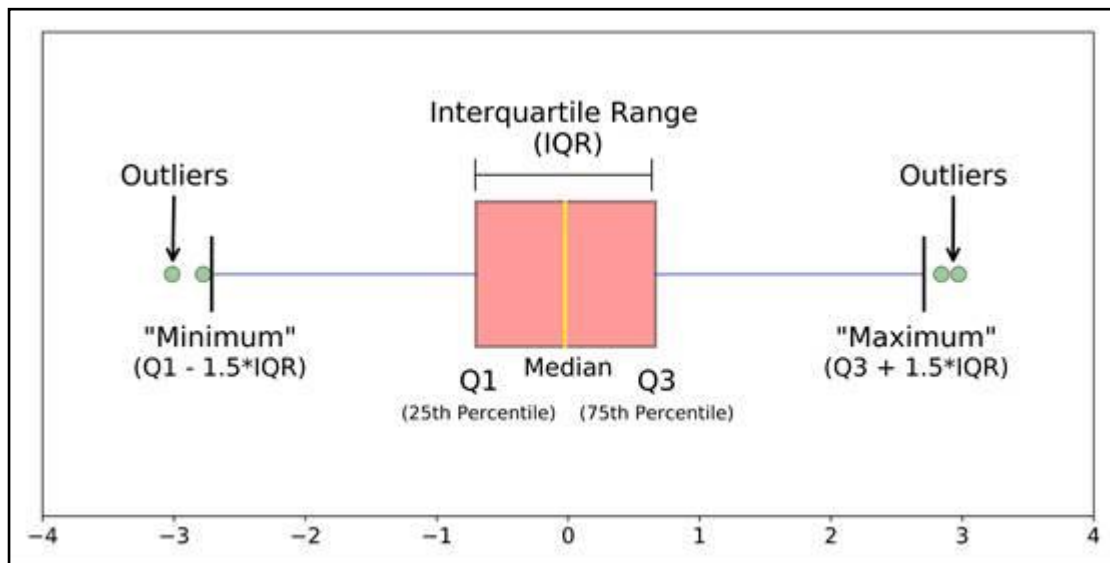
Στην περιγραφική στατιστική, το θηκόγραμμα (boxplot, επίσης γνωστό και ως box and whisker plot) αποτελεί ένα βολικό τρόπο γραφικής απεικόνισης πέντε αριθμητικών δεδομένων μίας σειράς παρατηρήσεων, ο οποίος χρησιμοποιείται συχνά στην επεξηγηματική ανάλυση δεδομένων. (*Πίνακας 3*) [48]

Μικρότερη παρατήρηση
Πρώτο τεταρτημόριο (Q1)
Διάμεσος (δ)
Τρίτο τεταρτημόριο (Q3)
Μεγαλύτερη παρατήρηση

Πίνακας 3. Αριθμητικά δεδομένα ενός boxplot. [48]

Το boxplot καλείται επίσης και διάγραμμα των πέντε αριθμών. Πρόκειται για ένα ορθογώνιο με δύο κεραίες (whiskers) το οποίο κατασκευάζεται με τον ακόλουθο τρόπο: η κάτω βάση του ορθογωνίου βρίσκεται στο Q1 και η πάνω στο Q3 . Η διάμεσος δ αναπαριστάται με ένα οριζόντιο ευθύγραμμο τμήμα μέσα στο ορθογώνιο. Το μήκος των βάσεων του ορθογωνίου λαμβάνεται αυθαίρετα. Η πάνω και η κάτω κεραία έχουν τη μορφή T και ανεστραμμένου T αντίστοιχα και εκτείνονται μέχρι τις οριακές τιμές. [49]

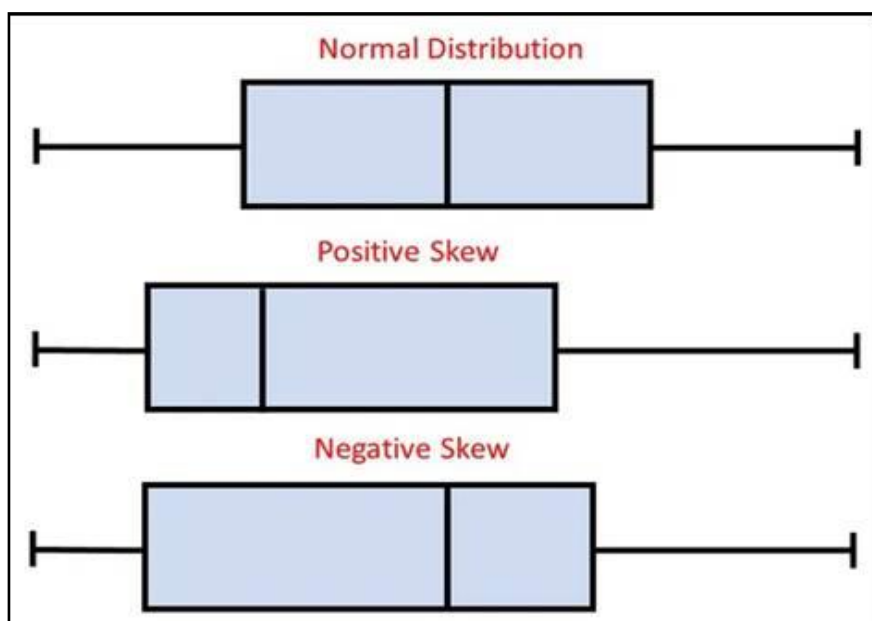
Τα boxplot δείχνουν οπτικά την κατανομή των αριθμητικών δεδομένων και της ασυμμετρίας, μέσω της εμφάνισης των τεταρτημορίων δεδομένων (ή εκατοστημορίων) και των μέσων όρων. Ένα boxplot χωρίζει τα δεδομένα σε ενότητες, καθεμία από τις οποίες περιέχει περίπου το 25% των δεδομένων. (**Εικόνα 22**)



Εικόνα 22. Σχηματική απεικόνιση ενός boxplot. [50]

Τα boxplot είναι χρήσιμα καθώς παρέχουν μία οπτική περίληψη των δεδομένων. Με αυτόν τον τρόπο, επιτρέπουν στους ερευνητές να προσδιορίσουν γρήγορα τις διαμέσους, τη διασπορά του συνόλου δεδομένων και την ασυμμετρία. Σημειώνεται ότι, σε κάθε boxplot, η κάθετη γραμμή αναπαριστά τη διάμεσο των δεδομένων. [51]

Επίσης, τα boxplot είναι χρήσιμα γιατί δείχνουν την ασυμμετρία ενός συνόλου δεδομένων, δηλαδή εάν ένα σύνολο στατιστικών δεδομένων είναι κανονικά κατανομημένο ή παρουσιάζει ασυμμετρία. Εάν το ευθύγραμμο τμήμα που αναπαριστά τη διάμεσο βρίσκεται στο μέσο του ορθογώνιου, η κατανομή είναι συμμετρική. Εάν βρίσκεται πλησιέστερα προς την κάτω πλευρά του ορθογώνιου, η κατανομή παρουσιάζει θετική ασυμμετρία. Τέλος, εάν βρίσκεται πλησιέστερα προς την άνω πλευρά του ορθογώνιου, η κατανομή παρουσιάζει αρνητική ασυμμετρία. Επίσης, τα θηκογράμματα είναι εξαιρετικά χρήσιμα για τη σύγκριση των κατανομών δύο ή περισσότερων δειγμάτων. Σημειώνεται ότι το θηκόγραμμα μπορεί να σχεδιαστεί είτε οριζόντια, είτε κατακόρυφα. [49], [51] (*Εικόνα 23*)



Εικόνα 23. Ασυμμετρία σε boxplot. [51]

Το boxplot δείχνει διαφορές μεταξύ των πληθυσμών, δειγμάτων ή ομάδων. Οι αποστάσεις μεταξύ των διαφόρων τμημάτων του boxplot αναδεικνύουν το μέγεθος της διασποράς και την ασυμμετρία των δεδομένων και, παράλληλα, παρέχουν πολλές στατιστικές πληροφορίες. [48]

2.5 Clustering Spatial Data

2.5.1 Ορισμός Geospatial Data

Τα χωρικά δεδομένα, επίσης γνωστά και ως γεωχωρικά δεδομένα (Geospatial Data), είναι ένας όρος που χρησιμοποιείται για την περιγραφή δεδομένων που σχετίζονται ή περιέχουν πληροφορίες σχετικά με μία συγκεκριμένη τοποθεσία στην επιφάνεια της Γης. Τα γεωχωρικά δεδομένα υπάρχουν σε ποικίλες μορφές και περιέχουν κάτι περισσότερο από απλές

πληροφορίες συγκεκριμένης τοποθεσίας. [52], [53] Τα δεδομένα αυτά έχουν ορισμένα ιδιαίτερα χαρακτηριστικά, όπως θέση, απόσταση και γειτνίαση, η ανάλυση των οποίων τους προσδίδει προστιθέμενη αξία. [54]

Η θέση πάνω στην επιφάνεια της γης προσδιορίζεται επακριβώς με βάση τις γεωγραφικές ή καρτεσιανές συντεταγμένες, οι οποίες αποτελούν δύο διαφορετικούς τρόπους μέτρησης του γεωγραφικού μήκους – δηλαδή την απόσταση από τον πρώτο μεσημβρινό που διέρχεται από το αστεροσκοπείο του Greenwich στην Αγγλία – και του γεωγραφικού πλάτους, την απόσταση από τον ισημερινό. Σήμερα ωστόσο, ο ακριβής προσδιορισμός των γεωγραφικών συντεταγμένων ενός σημείου γίνεται με τη βοήθεια ειδικών συσκευών εφοδιασμένων με τεχνολογία εντοπισμού θέσης (GPS – Global Position System) και τη βοήθεια δορυφόρων.

Τα δεδομένα συνήθως χωρίζονται σε δύο ομάδες, με βάση την πηγή από όπου προέρχονται, δηλαδή σε πρωτογενή και δευτερογενή δεδομένα. Τα πρωτογενή δεδομένα είναι αυτά που συλλέγει ο ερευνητής ακολουθώντας κάποια μεθοδολογία και ορισμένα εργαλεία. Δευτερογενή θεωρούνται τα δεδομένα που έχουν συλλεχθεί και είναι διαθέσιμα είτε δωρεάν, είτε με ορισμένο κόστος. Και στις δύο περιπτώσεις, είναι απαραίτητη η συλλογή δεδομένων που, ειδικά στην περίπτωση των γεωχωρικών δεδομένων, θα πρέπει να περιλαμβάνει κάποιου είδους χωρική πληροφορία, άμεση (γεωγραφικές συντεταγμένες) ή έμμεση (τιμή σε ένα σύστημα αναφοράς, όπως το όνομα του Δήμου).

2.5.2 Clustering Γεωχωρικών Δεδομένων

Clustering γεωχωρικών δεδομένων ονομάζεται η μέθοδος ομαδοποίησης χωρικών δεδομένων σε ομάδες που ονομάζονται clusters (συστάδες). Τα δεδομένα που ομαδοποιούνται σε ένα cluster φέρουν μεγάλο βαθμό ομοιότητας, ενώ, παράλληλα, τα clusters πρέπει να είναι όσο πιο ανόμοια γίνεται μεταξύ τους. Ο σκοπός είναι να γίνει μία γενίκευση και να αναγνωριστούν συσχετίσεις μεταξύ των δεδομένων.

Οι ομαδοποιήσεις που γίνονται, συνήθως, βασίζονται σε κριτήρια, τα οποία αναλύονται παρακάτω.

2.5.2.1 Ομαδοποίηση Κατατμήσεων (Partition Clustering)

Μία ομαδοποίηση κατατμήσεων είναι ένας διαχωρισμός των σημείων δεδομένων σε μη επικαλυπτόμενα υποσύνολα, έτσι ώστε κάθε σημείο δεδομένων να βρίσκεται ακριβώς σε ένα υποσύνολο. Βασικά, ταξινομεί τα δεδομένα σε ομάδες, ικανοποιώντας αυτές τις δύο απαιτήσεις:

1. Κάθε σημείο δεδομένων ανήκει μόνο σε ένα cluster
2. Κάθε cluster έχει τουλάχιστον ένα σημείο δεδομένων.

Αυτή η ομαδοποίηση δύναται να εκτελεστεί μέσω τριών αλγορίθμων: μέσω του K-Means, του K-medoids clustering/PAM (Partitioning Around Medoids) και του CLARA (Clustering LARge Applications).

Αυτή η μέθοδος χρησιμοποιείται για δημιουργία clusters σε δεδομένα που δεν έχουν επισημανθεί ρητά (unlabeled data) και οι πιο συνηθισμένες χρήσεις της είναι για ομαδοποίηση αποθεμάτων ανά δραστηριότητα πωλήσεων, προσδιορισμό ομάδων στην παρακολούθηση της υγείας και άλλα.

2.5.2.2 *Ιεραρχική Ομαδοποίηση*

Η ιεραρχική ομαδοποίηση είναι μία μέθοδος ομαδοποίησης παρόμοια με αυτή των καταταμίσεων, αλλά ο τρόπος ταξινόμησης των σημείων δεδομένων είναι διαφορετικός. Αρχικά, θεωρεί πως κάθε σημείο δεδομένων είναι ένα ξεχωριστό cluster. Στη συνέχεια, συγχωνεύει το πιο παρόμοιο cluster που γειτνιάζει. Αυτό επαναλαμβάνεται έως ότου συγχωνευθούν όλες οι συστάδες.

Αν και η ιεραρχική ομαδοποίηση είναι υπολογιστικά ακριβής, παράγει διαισθητικά αποτελέσματα και χρησιμοποιείται όταν δεν υπάρχουν πολλές πληροφορίες σχετικά με τα δεδομένα. Ένα πραγματικό σενάριο, όπου η ιεραρχική ομαδοποίηση χρησιμοποιείται εκτεταμένα, είναι η χαρτογράφηση ιών κατά τη διάρκεια της εξάπλωσης μίας επιδημίας.

2.5.2.3 *Ασαφής Ομαδοποίηση (Fuzzy Clustering)*

Στην ασαφή ομαδοποίηση, μέσω του αλγορίθμου Fuzzy C-Means, ανακαλύπτεται το κεντροειδές των σημείων δεδομένων και, στη συνέχεια, υπολογίζεται η απόσταση κάθε σημείου δεδομένων από τα γνωστά κεντροειδή έως ότου τα σχηματισμένα cluster σταθεροποιηθούν. Διαφέρει πολύ σαν φιλοσοφία από την ομαδοποίηση καταταμίσεων, καθώς επιτρέπει στα σημεία δεδομένων να ταξινομούνται μερικώς σε περισσότερα από ένα cluster.

Κάθε σημείο δεδομένων μπορεί, θεωρητικά, να ανήκει σε όλες τις ομάδες με συνάρτηση μέλους που κυμαίνεται μεταξύ 0 και 1. Με 0 ορίζεται το σημείο δεδομένων που βρίσκεται στο περισσότερο απομακρυσμένο δυνατό σημείο από το κέντρο ενός cluster και το 1 όπου το σημείο δεδομένων είναι πλησιέστερα προς το κέντρο.

Η ασαφής ομαδοποίηση είναι χρήσιμη όταν πρέπει να ομαδοποιηθούν εικόνες ή να καταταμιστούν περιοχές νερού, βλάστησης και βράχων σε δορυφορικές εικόνες. Χρησιμοποιείται πολύ συχνά σε περιπτώσεις όπου ο αριθμός των cluster δε μπορεί να αποφασιστεί εκ των προτέρων και επιτρέπει τη συγχώνευση συστάδων με αδύναμα όρια.

2.5.2.4 Ομαδοποίηση βάσει Πυκνότητας

Η ομαδοποίηση, με βάση την πυκνότητα, λειτουργεί ομαδοποιώντας περιοχές υψηλής πυκνότητας και ξεχωρίζοντάς τις από περιοχές χαμηλής πυκνότητας. Ο πεισσότερο γνωστός αλγόριθμος ομαδοποίησης είναι ο αλγόριθμος DBSCAN (Density-based spatial clustering of applications with noise – μέσω εφαρμογής θορύβου).

Η πυκνότητα υπολογίζεται χρησιμοποιώντας δύο παραμέτρους:

1. EPS: Καθορίζει τη γειτνίαση γύρω από το σημείο δεδομένων, δηλαδή εάν η απόσταση μεταξύ δύο σημείων είναι ίση ή μικρότερη από την τιμή του EPS, τότε τα σημεία θεωρούνται γειτονικά
2. MinPts: Ορίζει τον ελάχιστο αριθμό σημείων δεδομένων που σχηματίζουν μία γειτονιά. Το μέγεθος του συνόλου δεδομένων και η τιμή των MinPts είναι άμεσα αναλογικά.

Ο αλγόριθμος DBSCAN εξετάζει κάθε σημείο και εάν αυτό εμπεριέχει την τιμή MinPts εντός EPS, τότε ξεκινά η δημιουργία cluster. Οποιοδήποτε άλλο σημείο ορίζεται ως θόρυβος. Αυτή η διαδικασία συνεχίζεται έως ότου σχηματιστεί ένα σύμπλεγμα συνδεδεμένο με πυκνότητα και, στη συνέχεια, επανεκκινείται με ένα νέο σημείο.

Η μέθοδος χρησιμοποιείται κυρίως για ομαδοποίηση σε επίπεδο χώρο. Μπορούν να επιτευχθούν καλά αποτελέσματα, εάν χρησιμοποιείται για τη χαρτογράφηση των επιπτώσεων των φυσικών καταστροφών ή για τη σχεδίαση της θέσης των μετεωρολογικών σταθμών σε μία πόλη.

2.5.2.5 Ομαδοποίηση βάσει μοντέλου

Αυτή η μέθοδος ομαδοποίησης χρησιμοποιεί ορισμένα μοντέλα για συστάδες και προσπαθεί να βελτιστοποιήσει την προσαρμογή μεταξύ των δεδομένων και των μοντέλων. Σε αυτήν την προσέγγιση, τα δεδομένα θεωρούνται ότι προέρχονται από ένα μείγμα κατανομών πιθανότητας, καθεμία από τις οποίες αντιπροσωπεύει ένα διαφορετικό cluster. Κάθε στοιχείο διαμορφώνεται είτε από μία κανονική κατανομή, είτε από κατανομή Gauss.

Ο αλγόριθμος μεγιστοποίηση προσδοκίας (expectation-maximization) είναι ο πλέον γνωστός αλγόριθμος ομαδοποίησης με βάση το μοντέλο. Στην προκειμένη περίπτωση, ένας clustering αλγόριθμος θεωρείται πως λειτουργεί ιδανικά όταν τα δεδομένα συμμορφώνονται με το μοντέλο.

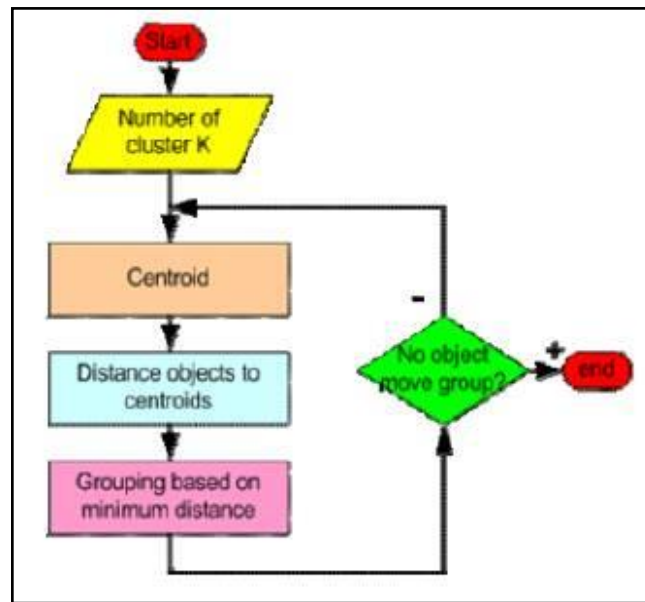
Χρησιμοποιείται σε περιπτώσεις όπου οι συστάδες έχουν αδύναμα όρια και τα σημεία δεδομένων έχουν πολλά σημεία γειτνίασης. Τέλος, είναι πολύ περισσότερο ευέλικτο όσον αφορά τη συνδιακύμανση (covariance) συστάδας. [55]

2.6 Αλγόριθμος *k-means*

2.6.1 Ορισμός αλγορίθμου *k-means*

Ο διαμεριστικός αλγόριθμος *k-means* αποτελεί έναν από τους απλούστερους και δημοφιλέστερους αλγορίθμους ομαδοποίησης, οι οποίοι ανήκουν στην ευρύτερη κατηγορία των τεχνικών εκμάθησης χωρίς επίβλεψη. Ο συγκεκριμένος αλγόριθμος είναι δημοφιλής εξαιτίας της απλότητας της υλοποίησής του και της γραμμικής πολυπλοκότητάς του.

Η πολυπλοκότητα του συγκεκριμένου αλγορίθμου είναι της τάξης n ($O(n)$), όπου n το σύνολο των στοιχείων. Η διαδικασία της ομαδοποίησης ενός συνόλου δεδομένων, με βάση τον αλγόριθμο *k-means*, είναι εύκολη. Η μοναδική προϋπόθεση για την εφαρμογή του είναι ο αριθμός (k) των clusters (ομάδων) που θα προκύψουν να είναι καθορισμένος εκ των προτέρων. Η κύρια ιδέα είναι να προσδιοριστούν αρχικά k centroids (κεντροειδή), ένα για κάθε cluster. Αυτά τα αρχικά centroids πρέπει να επιλεγούν με επιδέξιο και σωστό τρόπο, δεδομένου ότι οι διαφορετικές αρχικές θέσεις για τα centroids δίνουν κάθε φορά διαφορετικά αποτελέσματα. Συνεπώς, η αρχική θέση των centroids είναι καθοριστικής σημασίας για τα αποτελέσματα που θα προκύψουν από το συγκεκριμένο αλγόριθμο. Για αυτόν το λόγο, θεωρείται καλύτερη η επιλογή εκείνων των centroids που να απέχουν μεταξύ τους όσο περισσότερο γίνεται. Το επόμενο βήμα είναι η επιλογή κάθε στοιχείου από το σύνολο των δεδομένων και η συσχέτισή του με το κοντινότερο σε αυτό centroid. Όταν αυτό πραγματοποιηθεί για όλα τα στοιχεία του συνόλου δεδομένων, το πρώτο βήμα έχει ολοκληρωθεί και προκύπτει η πρώτη ομαδοποίηση. Στη συνέχεια, απαιτείται να υπολογιστούν ξανά k νέα centroids, που θα είναι το κέντρο βάρους του κάθε cluster του προηγούμενου βήματος. Ορίζονται, λοιπόν, νέα k centroids και επαναλαμβάνεται, εκ νέου, η διαδικασία ανάθεσης καθενός από τα στοιχεία του συνόλου δεδομένων στο κοντινότερο με αυτό, νέο πλέον, centroid. Το αποτέλεσμα αυτής της επανάληψης είναι ότι, σε κάθε βήμα, τα centroids αλλάζουν θέση (ορίζονται νέα) και τα στοιχεία ανατίθενται στο κατάλληλο cluster κάθε φορά, με βάση το κοντινότερο centroid. Όταν σε κάποια επανάληψη δε σημειωθούν αντιμεταθέσεις στοιχείων, τότε τερματίζει η εκτέλεση του αλγορίθμου. Το αποτέλεσμα που προκύπτει είναι η ομαδοποίηση του συνόλου δεδομένων σε k clusters. [56] (*Εικόνα 24*)



Εικόνα 24. Βήματα αλγορίθμου k-means. [56]

Απώτερος στόχος του αλγορίθμου είναι να ελαχιστοποιήσει μία αντικειμενική συνάρτηση, τη λεγόμενη συνάρτηση τετραγωνικού λάθους, η οποία ορίζεται από τον ακόλουθο τύπο. (Εικόνα 25)

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

Εικόνα 25. Συνάρτηση τετραγωνικού λάθους. [56]

2.6.2 Μειονεκτήματα – Αδυναμίες του αλγορίθμου k-means

Στη συνέχεια, παρατίθενται οι αδυναμίες που παρουσιάζει ο αλγόριθμος k-means:

- Ο αλγόριθμος συγκλίνει σε τοπικό βέλτιστο και όχι σε καθολικό βέλτιστο.
- Ο τρόπος ορισμού των αρχικών centroids δεν είναι σαφώς καθορισμένος. Συνήθως επιλέγονται τυχαία. Συνεπώς, το αποτέλεσμα εξαρτάται από τα αρχικά centroids. Για αυτόν το λόγο, ενδέχεται οι λύσεις που θα προκύψουν από την εφαρμογή του να μην είναι οι βέλτιστες, ακριβώς λόγω «κακής» αρχικής επιλογής των centroids. Προκειμένου, να ξεπεραστεί αυτό το πρόβλημα, συνιστώνται πολλαπλές δοκιμές, με διαφορετικά αρχικά centroids.
- Επιπλέον, υπάρχει περίπτωση ένα cluster να μείνει χωρίς μέλη και να μην ανανεωθεί κάποιο centroid. Το πρόβλημα αυτό καλείται πρόβλημα των απόμακρων στοιχείων.
- Επίσης, τα αποτελέσματα εξαρτώνται και από το μέτρο απόστασης που χρησιμοποιείται. Συχνά, χρειάζεται να γίνει κανονικοποίηση των στοιχείων του συνόλου δεδομένων, προκειμένου να εφαρμοστεί κάποιο μέτρο απόστασης. Για

παράδειγμα, όταν ο αλγόριθμος εκτελεί clustering, με βάση την Ευκλείδεια απόσταση, προϋποθέτει ότι τα δεδομένα των clusters έχουν όλα σφαιρικό σχηματισμό.

- Ο αλγόριθμος δυσκολεύεται να αναγνωρίσει ομάδες με διαφορετικό σχηματισμό και μέγεθος, κυρίως σε πολύ μεγάλα σύνολα δεδομένων. Το επίπεδο δυσκολίας σχετίζεται με την πυκνότητα των στοιχείων και ποικίλλει.
- Τέλος, τα αποτελέσματα εξαρτώνται από την τιμή του k , η οποία αποτελεί στοιχείο εισόδου για τον αλγόριθμο. Παρόλο που υπάρχουν πολλοί τρόποι εκτίμησης του k , ο αλγόριθμος δεν καταφέρνει να βρει ένα ευρέως αποδεκτό βέλτιστο k . Ως βέλτιστο k ορίζεται εκείνο που αποδίδει με τον καλύτερο τρόπο το διαχωρισμό του εκάστοτε συνόλου δεδομένων, με τρόπο που οι ομάδες να έχουν νόημα. [56]

3

Τεχνολογίες

Στο Κεφάλαιο 3, αναφέρονται οι τεχνολογικές λύσεις που αποτελούν τα εργαλεία υλοποίησης για το λογισμικό που αναπτύχθηκε στην παρούσα εργασία. Αρχικά, δίδεται ο ορισμός τους και ορισμένα χαρακτηριστικά τους. Ακολούθως, γίνεται ειδική αναφορά σε συγκεκριμένες τεχνολογικές πτυχές που χρησιμοποιήθηκαν στο κομμάτι της υλοποίησης.

3.1 FIWARE

Το FIWARE αποτελεί μία πλατφόρμα, που προωθείται από την Ευρωπαϊκή Ένωση, για την ανάπτυξη εφαρμογών Διαδικτύου του Πραγμάτων (IoT) και Διαδικτύου του Μέλλοντος (Future Internet). Παρέχει μία εντελώς ανοικτή, δημόσια και δωρεάν αρχιτεκτονική, καθώς και ένα σύνολο προδιαγραφών που επιτρέπει στους προγραμματιστές, τους παρόχους υπηρεσιών, τις εταιρείες και άλλους οργανισμούς να αναπτύξουν προϊόντα που ανταποκρίνονται στις ανάγκες τους, ενώ παραμένουν ανοικτά και καινοτόμα.

3.1.1 Ιστορία

Το FIWARE χρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση στο πλαίσιο του έργου συνεργασίας δημόσιου-ιδιωτικού τομέα για το Διαδίκτυο του Μέλλοντος (FI-PPP – Future Internet Public - Private Partnership).

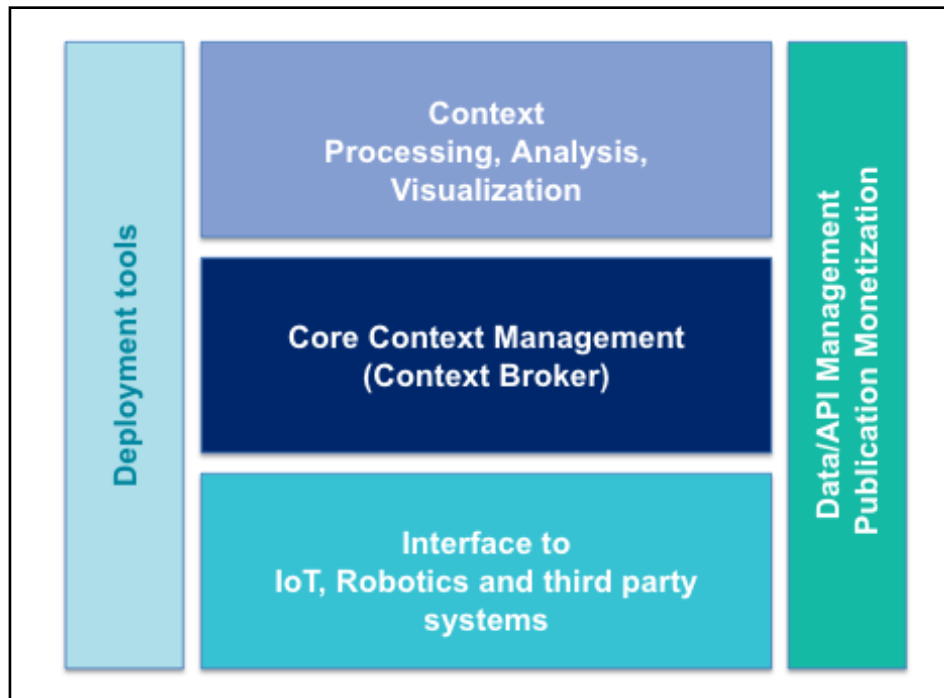
Στις αρχές του 2012, η ευρωπαϊκή κοινοπραξία, που συγκροτήθηκε από την Telefónica και την Orange, κολοσσούς των τηλεπικοινωνιών, καθώς και τις συμβουλευτικές εταιρείες Engineering και Atos, ανακοίνωσαν ένα έργο για την ανάπτυξη προτύπων FIWARE για έξυπνες πόλεις.

Μετέπειτα, δημιουργήθηκε το ίδρυμα FIWARE (FIWARE Foundation), προκειμένου να διατηρήσει ενεργή την πλατφόρμα και να εξασφαλίσει και άλλες πηγές χρηματοδότησης, εκτός της Ευρωπαϊκής Ένωσης. Αρκετές εταιρείες, σε διεθνές επίπεδο, την υποστηρίζουν μέσω αυτού του συνεταιρισμού. Πρόσφατα, το Μάιο του 2020, η Red Hat έγινε μέλος του ιδρύματος.
[57]

3.1.2 Ορισμός

Το FIWARE, ουσιαστικά, είναι μία ολοκληρωμένη λύση εφαρμογών λογισμικού (components) ανοικτού κώδικα, που μπορούν να χρησιμοποιηθούν είτε εξ' ολοκλήρου, είτε σε συνδυασμό με components τρίτων, για την επιτάχυνση ανάπτυξης Έξυπνων Λύσεων (Smart Solutions).

Αποτελείται από διάφορα εργαλεία, όπως ο Context Broker, components διασύνδεσης με συσκευές IoT, σε πολύ δημοφιλή πρωτόκολλα επικοινωνίας, και τέλος με components για επεξεργασία, ανάλυση και οπτικοποίηση δεδομένων. [58] (*Εικόνα 26*)



Εικόνα 26. Απεικόνιση των components FIWARE. [58]

Το μόνο **απαραίτητο** component, για να θεωρηθεί μία λύση πως αξιοποιεί τις δυνατότητες του FIWARE, είναι η ύπαρξη του FIWARE Context Broker.

3.1.3 Context Broker

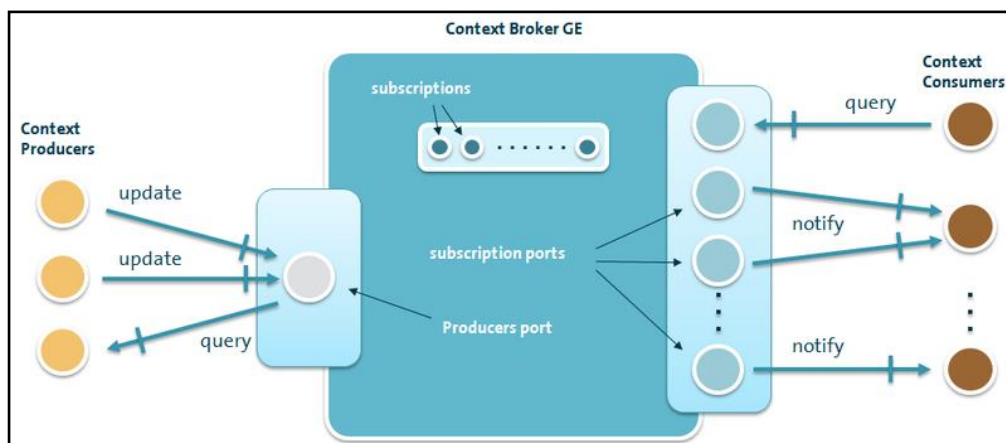
Ο Context Broker περιέχει όλη τη λογική της πλατφόρμας FIWARE και η ύπαρξή του αποτελεί προϋπόθεση για το χαρακτηρισμό μίας λύσης ως «Powered by FIWARE».

Δεν υπάρχει δόκιμος όρος στα Ελληνικά. Ωστόσο, όπως υπονοεί η Αγγλική του ονομασία, πρόκειται για μεσίτη (ή μεσάζοντα) που παρέχει πληροφορίες για την κατάσταση (context) των συσκευών που είναι συνδεδεμένες με αυτόν. Ουσιαστικά, αποτελεί μία γέφυρα μεταξύ του έξω κόσμου και των συσκευών. Επίσης, ευθύνεται για την αποθήκευση των καταστάσεων των συσκευών (context) σε βάση δεδομένων (MongoDb).

Παρέχει τη δυνατότητα Ενημέρωσης/Λήψης (Publish/Subscribe) δεδομένων μέσω των διεπαφών NGS19 και NGS110. Χρησιμοποιώντας αυτές τις διεπαφές, οι client μπορούν να έχουν τις κάτωθι λειτουργικότητες:

1. Καταχώρηση (register) εφαρμογών παραγωγής καταστάσεων (context producer), όπως, για παράδειγμα, η αρχικοποίηση ενός αισθητήρα ανίχνευσης κίνησης σε ένα δωμάτιο.
2. Ανανέωση δεδομένων κατάστασης (context information), για παράδειγμα ανίχνευση κίνησης
3. Ειδοποιήσεις για αλλαγές κατάστασης, είτε τη στιγμή που συμβαίνουν, είτε ανά τακτικά χρονικά διαστήματα
4. Αναζήτηση δεδομένων (querying)

Όλες οι παραπάνω είναι αναγκαίες σε περιβάλλοντα που απαιτούν επικοινωνία μεταξύ των συσκευών παραγωγής δεδομένων (context producers), όπως οι αισθητήρες, και των συσκευών ανάλυσης δεδομένων (context consumer applications), όπως μία εφαρμογή σε ένα έξυπνο τηλέφωνο. [59]



Εικόνα 27. Τα μέλη σε ένα σενάριο επικοινωνίας για ανάκτηση δεδομένων κατάστασης. [59]

Υπάρχουν αρκετοί Context Broker για την πλατφόρμα του FIWARE. Ωστόσο, ο πλέον ευρέως διαδομένος είναι ο Orion Context Broker, που υλοποιεί το FIWARE NGSI v2 API, και χρησιμοποιήθηκε στα πλαίσια αυτής της υλοποίησης.

3.1.3.1 Παράδειγμα Λειτουργικότητας

Προκειμένου να μπορέσει να μελετηθεί και να κατανοηθεί η λειτουργικότητα του Orion Context Broker, χρειάστηκε να εγκατασταθεί, ώστε να δοκιμαστεί απευθείας.

Για λόγους ευκολίας, συγκεντρώθηκαν όλοι οι διαθέσιμοι τύποι ενεργειών της προγραμματιστικής διεπαφής (Restful API) του Orion Context Broker σε μία συλλογή του

εργαλείου Postman¹. Ωστόσο, θα παρουσιαστούν ενδεικτικά ορισμένες σε ένα σενάριο που αφορά τη λειτουργία μίας έξυπνης λάμπας (smart lamp).

3.1.3.1.1 Καταχώρηση εντολών για την Έξυπνη Λάμπα

Η λειτουργικότητα των έξυπνων λαμπών ποικίλει. Οι απλούστερες λάμπες μπορεί να δέχονται μόνο εντολές για ενεργοποίηση και απενεργοποίηση, ενώ οι πιο προηγμένες δίνουν τη δυνατότητα ρύθμισης της φωτεινότητας ή μπορούν να συνδυάζονται με λειτουργίες άλλων έξυπνων συσκευών. Άρα, η πρώτη ενέργεια αφορά την καταχώρηση, για τη συγκεκριμένη έξυπνη λάμπα, όλων των λειτουργικοτήτων που μπορεί να προσφέρει. Αυτό γίνεται ως εξής.

```
1. curl --location --request POST 'http://192.168.10.254:1026/v2/registrations' \  
2. --header 'Content-Type: application/json' \  
3. --header 'fiware-service: openiot' \  
4. --header 'fiware-servicepath: /' \  
5. --data-raw '{  
6.     "description": "Lamp Commands",  
7.     "dataProvided": {  
8.         "entities": [  
9.             {  
10.                "id": "urn:ngsi-ld:Lamp:001",  
11.                "type": "Lamp"  
12.            }  
13.        ],  
14.        "attrs": [  
15.            "on",  
16.            "off"  
17.        ]  
18.    },  
19.    "provider": {  
20.        "http": {  
21.            "url": "http://thesis-iota-json:4041"  
22.        },  
23.        "legacyForwarding": true  
24.    }  
}
```

Τμήμα Κώδικα 1. cURL κλήση για καταχώρηση εντολών συσκευής σε IoT Agent.

¹ <https://www.getpostman.com/collections/0366d7edf1b7f748dcd0>

3.1.3.1.2 Έλεγχος κατάστασης Λάμπας

```
1. curl --location --request GET 'http://192.168.10.254:1026/v2/entities/urn:ngsi-  
id:Lamp:001?options=keyValues' \  
2. --header 'fiware-service: openiot' \  
3. --header 'fiware-servicepath: /'
```

Τμήμα Κώδικα 2. cURL κλήση για αναζήτηση Context συσκευής.

Το αποτέλεσμα αυτής της κλήσης ενημερώνει εάν η λάμπα βρίσκεται σε λειτουργία ή όχι.

3.1.3.1.3 Εντολή να τεθεί σε λειτουργία η Λάμπα

```
1. curl --location --request PATCH 'http://192.168.10.254:1026/v2/entities/urn:ngsi-  
id:Lamp:001/attrs' \  
2. --header 'fiware-service: openiot' \  
3. --header 'fiware-servicepath: /' \  
4. --header 'Content-Type: application/json' \  
5. --data-raw '{  
6.   "on": {  
7.     "type" : "command",  
8.     "value" : "on"  
9.   }  
10. }'
```

Τμήμα Κώδικα 3. cURL κλήση για αλλαγή context συσκευής.

Σαν αποτέλεσμα αυτής της κλήσης, η έξυπνη λάμπα, εάν δε βρίσκεται σε λειτουργία και παραμένει συνδεδεμένη στο δίκτυο του Orion Context Broker, θα λάβει εντολή να ενεργοποιηθεί. Όταν ολοκληρωθεί ή αποτύχει η διαδικασία, επιστρέφεται μήνυμα επιτυχίας ή σφάλματος.

3.1.4 IoT Agents

Η εφαρμογή λογισμικού (component) IoT Agent παρέχει τη δυνατότητα σε μία ή πολλαπλές συσκευές IoT να επικοινωνήσουν με τον Context Broker του FIWARE, χρησιμοποιώντας τα δικά τους, εγγενή, πρωτόκολλα επικοινωνίας και πρότυπα μεταφοράς δομημένων δεδομένων. Πρόκειται, ουσιαστικά, για ένα μεταγλωττιστή του πρωτοκόλλου επικοινωνίας της συσκευής IoT σε γλώσσα που καταλαβαίνει ο Context Broker.

Όλοι οι IoT Agent βασίζονται σε μία κοινή λογική, που θεσπίζει το FIWARE IoT Agent Framework. Η επιβαλλομένη αυτή κοινή λογική, σε όλες τις εκφάνσεις του IoT Agent, είναι η παρακάτω:

- Για όλες τις southbound επικοινωνίες, δηλαδή για επικοινωνίες από τον Context Broker προς τη συσκευή IoT, ο agent παρακολουθεί για μεταβολές σε οντότητες καταστάσεων (context entities) και δημιουργεί επανακλήσεις (callbacks) στη συσκευή
- Για όλες τις βόρειες (northbound) επικοινωνίες, δηλαδή από την IoT συσκευή προς τον Context Broker, ο agent παρέχει τη δυνατότητα αποστολής δομημένης πληροφορίας
- Αυθεντικοποίηση βασισμένη στο πρωτόκολλο OAuth2. [60]

Οι IoT Agents έχουν πρόσβαση σε βάση δεδομένων, η οποία δεν είναι απαραίτητο να είναι κοινή με αυτή του Context Broker, αλλά τα μόνα δεδομένα που αποθηκεύουν (persist) είναι το σχήμα των οντοτήτων κάθε συσκευής.

3.1.4.1 TCP Θύρες

Όλοι οι IoT Agents, δεδομένου ότι βασίζονται στο κοινό Agent Framework, είναι εφοδιασμένοι με δύο TCP (Transmission Control Protocol) θύρες, μία βόρεια (north) και μία νότια (south). Όπως προαναφέρθηκε, η βόρεια θύρα εξυπηρετεί, αποκλειστικά, την επικοινωνία με τον Context Broker, ενώ η νότια θύρα, την άμεση επικοινωνία με την IoT συσκευή.

3.1.4.1.1 Παράδειγμα Northbound επικοινωνίας

Μέσω της επικοινωνίας με βόρεια κατεύθυνση, επιτυγχάνεται η επικοινωνία με τον Context Broker.

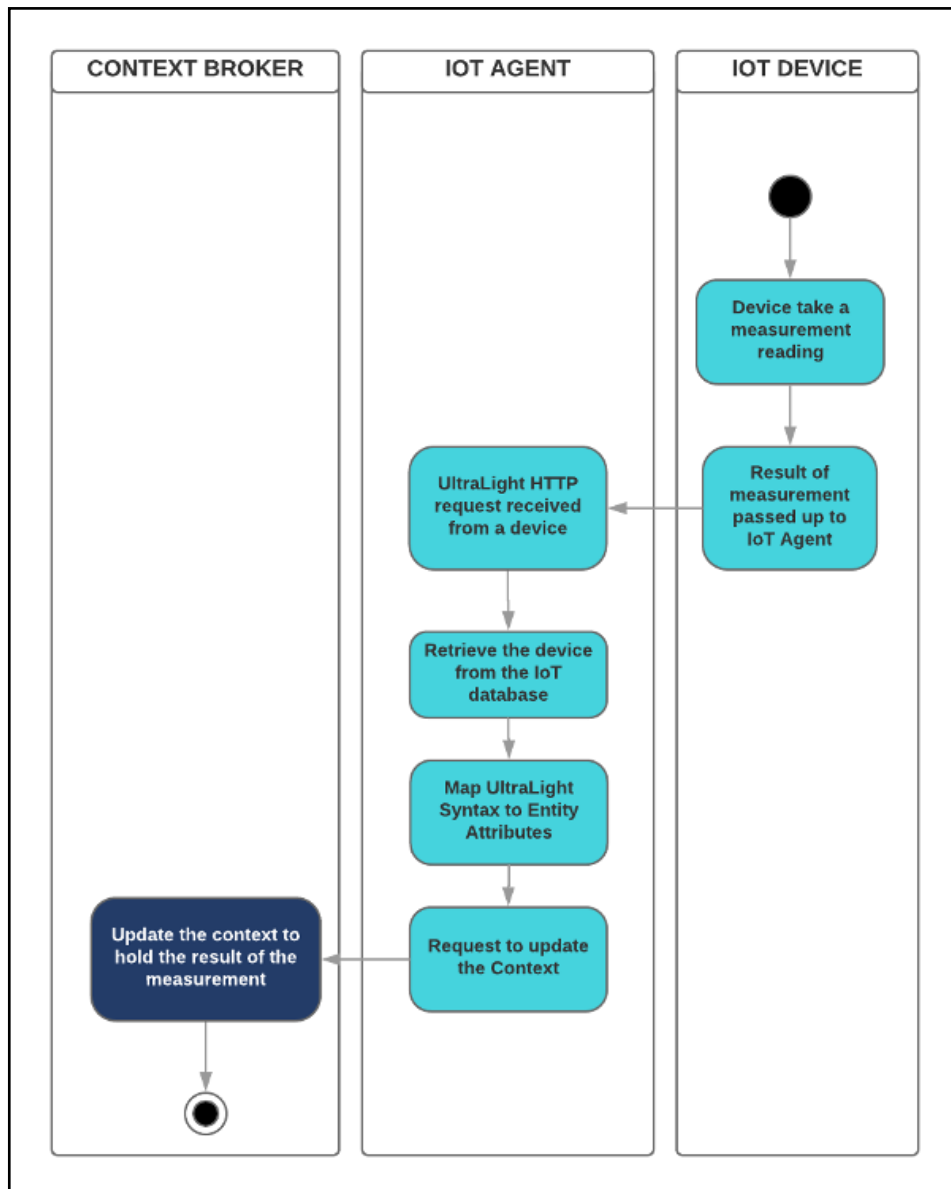
Για να κατανοηθεί καλύτερα η έννοια της northbound επικοινωνίας, θα εξηγηθεί με ένα απλό παράδειγμα.

Όταν ένας αισθητήρας θερμοκρασίας λαμβάνει μία νέα μέτρηση, πρέπει να ενημερώσει το component εκείνο που, εάν ερωτηθεί από κάποιο χρήστη, θα φροντίσει να του παρουσιάσει την τελευταία μέτρηση.

Συνεπώς, η διαδρομή της πληροφορίας ξεκινά από την IoT συσκευή (αισθητήρας), μεταφέρεται στον IoT Agent, ο οποίος ενημερώνει τον Context Broker, ο οποίος θα φροντίσει να αποθηκεύσει την αλλαγή (persist) σε βάση δεδομένων.

Εάν ερωτηθεί, με τη σειρά του, ο Context Broker, ως πηγή της κοινής αλήθειας, θα φροντίσει να αναζητήσει τη μέτρηση μίας συσκευής από τη βάση δεδομένων, για να παρουσιάσει τα αποτελέσματα στο χρήστη.

Στο σχεδιάγραμμα που ακολουθεί, παρουσιάζεται και οπτικά η προαναφερθείσα διαδικασία. (*Εικόνα 28*)



Εικόνα 28. Παράδειγμα επικοινωνίας μέσω της βόρειας θύρας του IoT Agent. [61]

3.1.4.1.2 Παράδειγμα Southbound επικοινωνίας

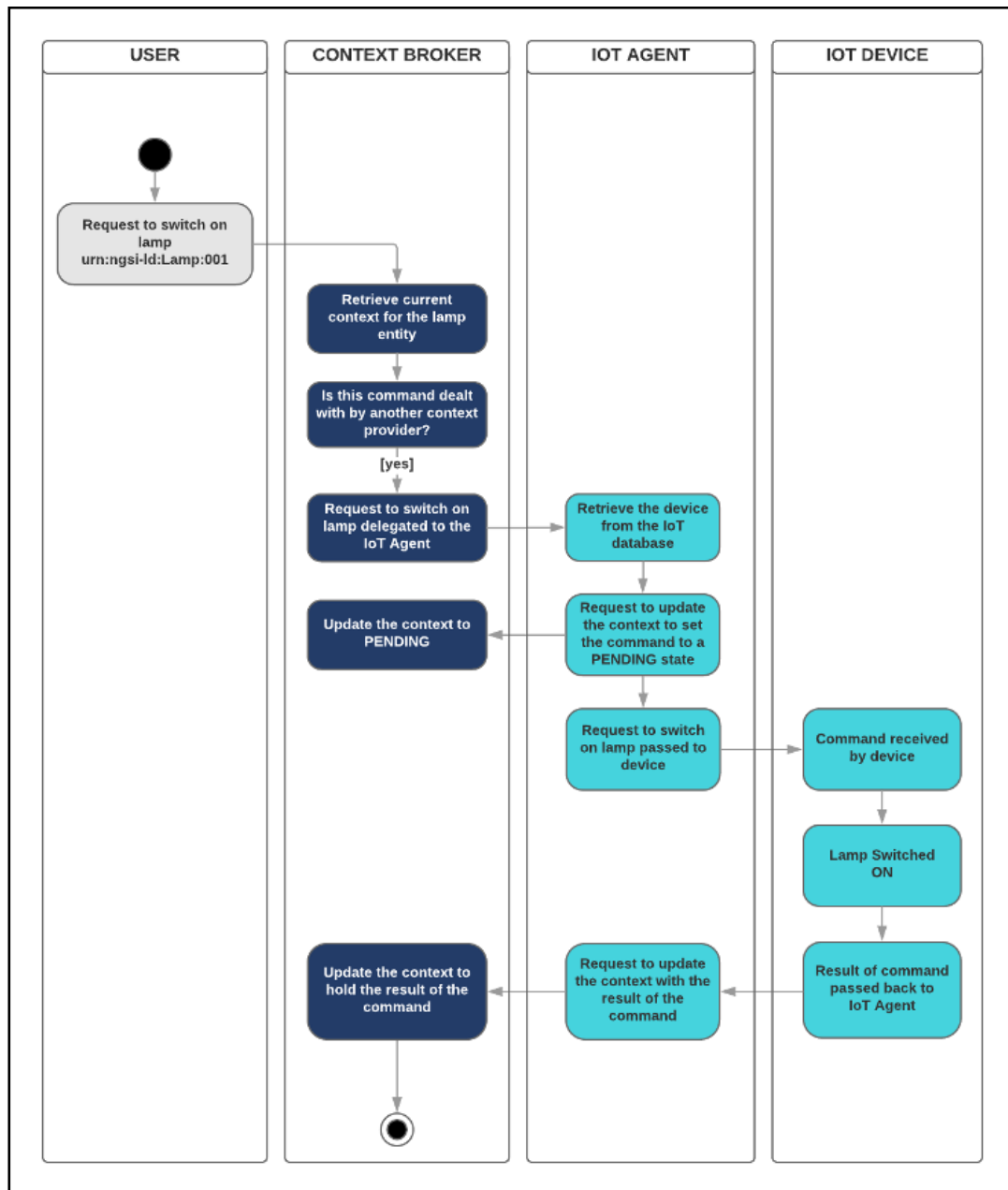
Η επικοινωνία από τη νότια θύρα είναι, ουσιαστικά, η αντίστροφη διαδικασία από την επικοινωνία με βόρεια κατεύθυνση (northbound).

Όταν ο Context Broker λαμβάνει μία εντολή για αλλαγή κατάστασης (context change) για μία έξυπνη συσκευή, επικοινωνεί με τον IoT Agent, ο οποίος μεταφέρει την εντολή στην εκάστοτε συσκευή.

Η μόνη διαφορά, σε αυτήν την περίπτωση, είναι πως ο Context Broker περιμένει να ενημερωθεί εάν η εντολή εκτελέστηκε επιτυχώς. Η συσκευή, όταν ολοκληρώσει επιτυχώς ή αποτυχημένα την εντολή, ενημερώνει τον IoT Agent, ο οποίος, με τη σειρά του, ενημερώνει τον Context

Broker για να μπορέσει να προχωρήσει σε αποθήκευση της κατάστασης (persist) στη βάση δεδομένων.

Ακολουθεί σχεδιάγραμμα που παρουσιάζει μία southbound επικοινωνία. (Εικόνα 29)



Εικόνα 29. Παράδειγμα επικοινωνίας μέσω της νότιας θύρας του IoT Agent. [61]

3.1.4.2 Πρωτόκολλα επικοινωνίας

Οι IoT Agent υποστηρίζουν τα πιο κοινά πρωτόκολλα επικοινωνίας για το Διαδίκτυο των Πραγμάτων (IoT), όπως HTTP (Hypertext Transfer Protocol), MQTT (Message Queuing Telemetry Transport) και AMQP (Advanced Message Queuing Protocol).

3.1.4.3 Πρότυπα μεταφοράς δεδομένων

Σύμφωνα με όσα έχουν προαναφερθεί, υπάρχει πληθώρα επιλογών όσον αφορά τα πρότυπα μεταφοράς δομημένων δεδομένων. Στα πλαίσια αυτής της διπλωματικής εργασίας, χρησιμοποιήθηκαν τα πρότυπα JSON και Ultralight.

3.1.4.3.1 IoT Agent JSON

Ο JSON IoT Agent, όπως υπονοεί και το όνομά του, αναλαμβάνει να μεταφέρει δεδομένα (payload) με τη μορφή JSON (JavaScript Object Notation). Πρόκειται για ένα πολύ κοινό μορφότυπο δεδομένων που χρησιμοποιεί κείμενο, αναγνώσιμο και από άνθρωπο, για τη μετάδοση πληροφοριών που αποτελούνται από ζεύγη κλειδιών-τιμών (key-value pairs). [62]

Ακολουθεί παράδειγμα επικοινωνίας συσκευής με τον IoT Agent μέσω πρωτοκόλλου HTTP για ενημέρωση μίας τιμής.

```
1. curl --location --request POST
   'http://192.168.10.254:7896/iot/json?k=jsonkey&i=motion001' \
2. --header 'Content-Type: application/json' \
3. --data-raw '{"corridor":"12", "reportDate":"14/2/2020", "state":"online"}'
```

Τμήμα Κώδικα 4. cURL κλήση ενημέρωσης Context σε IoT Agent JSON.

3.1.4.3.2 IoT Agent Ultralight (UL)

Ο Ultralight IoT Agent μεταφέρει τα δομημένα δεδομένα του με ένα πρότυπο κειμένου που ονομάζεται Ultralight. Πρόκειται για ένα πολύ συνοπτικό μορφότυπο κειμένου, που βασίζεται σε ζεύγη κλειδιών-τιμών (key-value pairs) διαχωρισμένα με το χαρακτήρα “|”. Πρόκειται για ένα πολύ οικονομικό – από άποψη μεταφοράς δεδομένων – πρότυπο, αλλά είναι αρκετά ασαφές, γιατί δε μπορεί να γίνει έλεγχος τύπου (type check) στα δεδομένα, καθώς όλες οι τιμές θεωρούνται ότι αποτελούν κείμενο (string).

Ακολουθεί το ίδιο παράδειγμα επικοινωνίας με IoT Agent που χρησιμοποιήθηκε παραπάνω, με αποτύπωση μέσω του προτύπου Ultralight

```
1. curl --location --request POST
   'http://192.168.10.254:7897/iot/d?k=ulkey&i=motion001' \
2. --header 'Content-Type: text/plain' \
3. --data-raw 'c|12|rd|14/2/2020|s|online'
```

Τμήμα Κώδικα 5. cURL κλήση ενημέρωσης Context σε IoT Agent Ultralight.

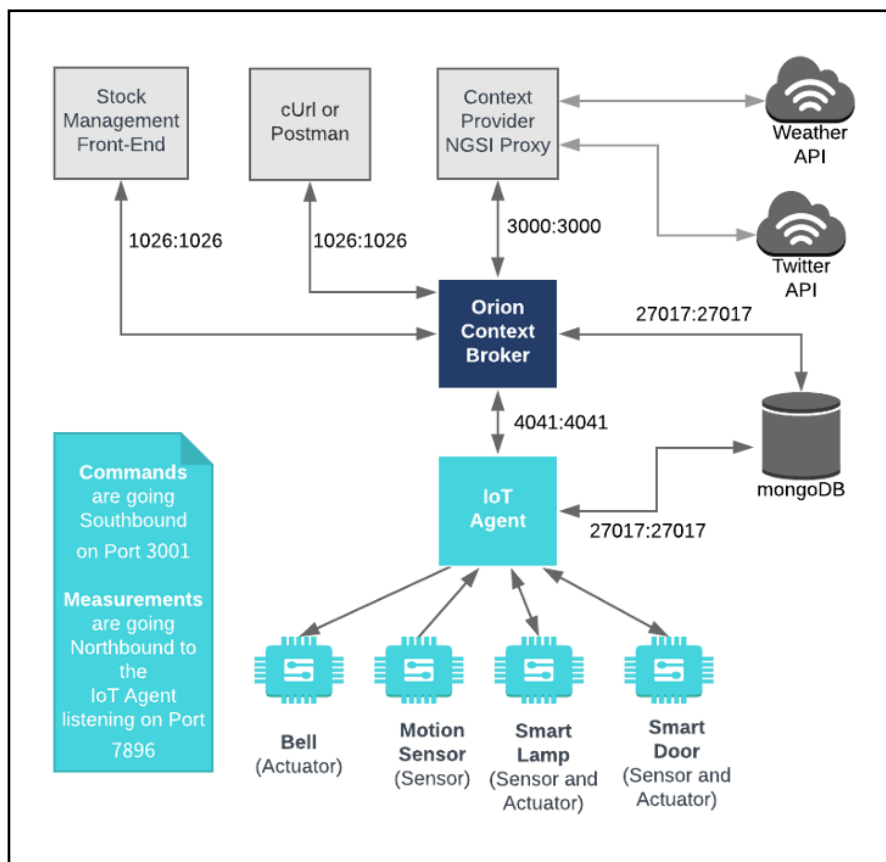
3.1.4.3.3 Διαφορές μεταξύ IoT Agent JSON και Ultralight

Όπως αναφέρθηκε και παραπάνω ο Ultralight IoT Agent επικοινωνεί χρησιμοποιώντας μία απλή γραμμή “|” (separator) για να ξεχωρίζει μία λίστα ζευγών κλειδιών – τιμών. Το payload του είναι πολύ απλό και σύντομο, αλλά αποτελεί σχετικά ασαφή μηχανισμό επικοινωνίας. Η κυριότερη χρήση του συγκεκριμένου προτύπου είναι σε περιπτώσεις όπου η οικονομία στα μηνύματα επικοινωνίας είναι καίριας σημασίας.

Το πιο κοινό πρότυπο μηνυμάτων που χρησιμοποιείται στο Διαδίκτυο είναι αυτό του JSON, που είναι οικείο σε οποιονδήποτε προγραμματιστή λογισμικού. Το JSON είναι αρκετά πιο γλαφυρό από το Ultralight, αλλά το κόστος αποστολής μεγαλύτερων μηνυμάτων αντισταθμίζεται από την εξοικείωση της σύνταξης.

Δεν υπάρχει πρακτική διαφορά μεταξύ της επικοινωνίας με το πρότυπο JSON και της επικοινωνίας με πρότυπο απλού κειμένου Ultralight, υπό την προϋπόθεση ότι η βάση αυτής της επικοινωνίας – με άλλα λόγια το πρωτόκολλο που καθορίζει τον τρόπο μετάδοσης των μηνυμάτων μεταξύ των στοιχείων – παραμένει το ίδιο.

Είναι προφανές πως η μετατροπή του payload των μηνυμάτων JSON εντός του IoT Agent σε μηνύματα NGSI και το αντίστροφο είναι διαδικασία που αφορά αποκλειστικά τον JSON IoT Agent. Το αντίστοιχο συμβαίνει και για μηνύματα Ultralight και τον Ultralight IoT Agent.



Εικόνα 30. Οπτική απεικόνιση ενός πλήρους οικοσυστήματος βασισμένο σε FIWARE. [61]

3.2 MongoDB

Ως βάση δεδομένων, για την επικοινωνία μεταξύ των IoT συσκευών και του Context Broker, χρησιμοποιείται η MongoDB. Πρόκειται για μία NoSQL (not only SQL) βάση δεδομένων, η οποία έχει ως κύριο χαρακτηριστικό τη δυνατότητα ανομοιογένειας των δεδομένων καθώς και την έλλειψη ανάγκης για διασύνδεση των επιμέρους συλλογών (πινάκων).

3.2.1 Ορισμός

Η MongoDB είναι μία δωρεάν και ανοικτού λογισμικού διαπλατφορμική, εγγραφοκεντρική βάση δεδομένων. Ανήκει στην κατηγορία των NoSQL βάσεων δεδομένων, δηλαδή τα δεδομένα δεν αποθηκεύονται σε πίνακες, όπως στις σχεσιακές βάσεις δεδομένων, αλλά σε σχήματα (schemas) με μορφή BSON (Binary JSON), η οποία είναι παρόμοια με αυτή του προτύπου JSON. [63]

Χάρη στη μεγάλη απήχηση και την ευρεία χρήση της MongoDB, υπάρχουν πολλά υποστηρικτικά εργαλεία για τη διαχείρισή της, τα οποία είναι έργα Ανοικτού Κώδικα (Open Source) και υποστηρίζονται από μεγάλη κοινότητα προγραμματιστών.

Ουσιαστικά, η MongoDB αποτελεί βάση δεδομένων που αποτελείται από μία συλλογή εγγράφων, στην οποία αποθηκεύονται οι απαιτούμενες πληροφορίες. Αναλυτικότερα, σε αντιστοιχία με τις παραδοσιακές σχεσιακές βάσεις δεδομένων, η εγγραφή (record), που αποθηκεύεται σε έναν πίνακα (table), ονομάζεται έγγραφο (document) στη MongoDB και αποθηκεύεται σε μία συλλογή (collection). Το έγγραφο (document) περιέχει όλα τα πεδία που απαρτίζουν το αντικείμενο και αποτελείται από ζεύγη κλειδιών – τιμών (keyvalue pairs). [64]

(*Εικόνα 31*)



Εικόνα 31. Παράδειγμα BSON αντικειμένου στη βάση MongoDB. [65]

3.2.2 Χαρακτηριστικά

Η MongoDB παρέχει υψηλή διαθεσιμότητα και απόδοση, καθώς και αυτόματη κλιμάκωση. Τα κυριότερα χαρακτηριστικά της βάσης είναι:

- Ενσωμάτωση εγγενών αντικειμένων (nested documents) στο μοντέλο δεδομένων, η οποία μειώνει δραστικά την ανάγκη για πολλαπλές αναγνώσεις ή εγγραφές στα αποθηκευτικά μέσα
- Δυνατότητα οριζόντιας κλιμάκωσης και κατακερματισμού των δεδομένων (sharding) σε ένα σύνολο (cluster) υπολογιστών
- Αυτόματη ανάκαμψη από βλάβες (automatic fail over) και πλεονασμό δεδομένων (data redundancy) χάρη στην υπηρεσία λειτουργίας αντιγράφων (replication)
- Παροχή δυνατότητας δεικτοδότησης (indexing) εγγράφων, καθώς και στα ενσωματωμένα έγγραφα (nested documents), γεγονός που επιτρέπει ταχύτερη αναζήτηση και μικρότερο αντίκτυπο στους πόρους ενός συστήματος.

3.2.3 Πλεονεκτήματα

Η χρήση εγγράφων, έναντι των παραδοσιακών οντοτήτων των πινάκων των σχεσιακών βάσεων, προσφέρει τα παρακάτω πλεονεκτήματα:

- Δυνατότητα χρήσης αδόμητου ή ημιδομημένου σχήματος (schema) που υποστηρίζει την εύκολη επέκταση των εφαρμογών.
- Δυνατότητα για αποθήκευση ενσωματωμένων εγγράφων, που εξαλείφει την ανάγκη για πράξεις τύπου join σε παραδοσιακές σχεσιακές βάσεις δεδομένων.
- Η μορφή τους διευκολύνει πολύ την ανάπτυξη εφαρμογών γιατί μοιάζει σε τύπους δεδομένων πολλών μοντέρνων γλωσσών προγραμματισμού (όπως η javascript). [64]

3.2.4 Ροές Αλλαγών – Change Streams

Υπάρχει μεγάλη πίεση για τις σύγχρονες εφαρμογές να αντιδρούν άμεσα στις αλλαγές που εμφανίζονται. Χάρη στις ροές αλλαγών (Change Streams), υπάρχει αυτή η δυνατότητα που επιτρέπει στις εφαρμογές να μεταδίδουν ροές δεδομένων σε πραγματικό χρόνο, αξιοποιώντας τις υποκείμενες δυνατότητες αναπαραγωγής της MongoDB.

Αποτελεί πολύ χρήσιμη και καιρίας σημασίας δυνατότητα που επιτρέπει, σε ευρύ φάσμα εφαρμογών, να αντιδράσει σε αλλαγή δεδομένων. Για παράδειγμα, δυναμικές εφαρμογές συναλλαγών που πρέπει να ενημερωθούν σε πραγματικό χρόνο καθώς οι τιμές των μετοχών αλλάζουν ή η δημιουργία ενός IoT pipeline που δημιουργεί συναγερμούς κάθε φορά που ένα συνδεδεμένο όχημα κινείται έξω από μία περιφραγμένη περιοχή.

Η ροή αλλαγών (change stream) ειδοποιεί την εφαρμογή, μέσω προγραμματιστικού callback, για όλες τις αλλαγές σε έγγραφα (συμπεριλαμβανομένων των διαγραφών) και παρέχει πρόσβαση σε όλες τις διαθέσιμες πληροφορίες καθώς πραγματοποιούνται αλλαγές. Αυτό συμβαίνει χωρίς την εκ νέου αναζήτηση (querying) δεδομένων που – συνήθως – προκαλεί καθυστερήσεις, καθώς η βάση δεδομένων ελέγχεται τακτικά, ακόμα και εάν δεν έχει μεταβληθεί τίποτα.

Τα κυριότερα χαρακτηριστικά των ροών αλλαγών (Change Streams) είναι τα ακόλουθα:

- Στοχευμένες αλλαγές. Οι ροές μπορούν να φιλτραριστούν για να παρέχουν πληροφορίες μόνο για τις επιθυμητές αλλαγές σε συγκεκριμένα έγγραφα. Για παράδειγμα, τα φίλτρα μπορούν να είναι σε τύπο λειτουργίας ή να αποτελούν πεδία εντός του εγγράφου.
- Ταξινόμηση αλλαγών. Η MongoDB διαθέτει λογικό ρολόι που επιτρέπει στις εφαρμογές να λαμβάνουν πάντα τις αλλαγές, με τη σειρά που εφαρμόστηκαν στη βάση δεδομένων.
- Ασφάλεια. Οι ροές αλλαγών είναι ασφαλείς. Οι χρήστες μπορούν να δημιουργήσουν ροές αλλαγών μόνο σε συλλογές στις οποίες τους έχει παραχωρηθεί πρόσβαση.
- Ευκολία στη χρήση. [66]

3.3 *SQL Server*

3.3.1 *Ορισμός SQL*

Η SQL (Structured Query Language) είναι μία γλώσσα βάσεων δεδομένων, σχεδιασμένη για την ανάκτηση και διαχείριση δεδομένων σε συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων. [67] Οι λειτουργίες της περιλαμβάνουν την εισαγωγή, ανάκτηση, ανανέωση και διαγραφή (Create, Read, Update, Delete ή κοινώς CRUD).

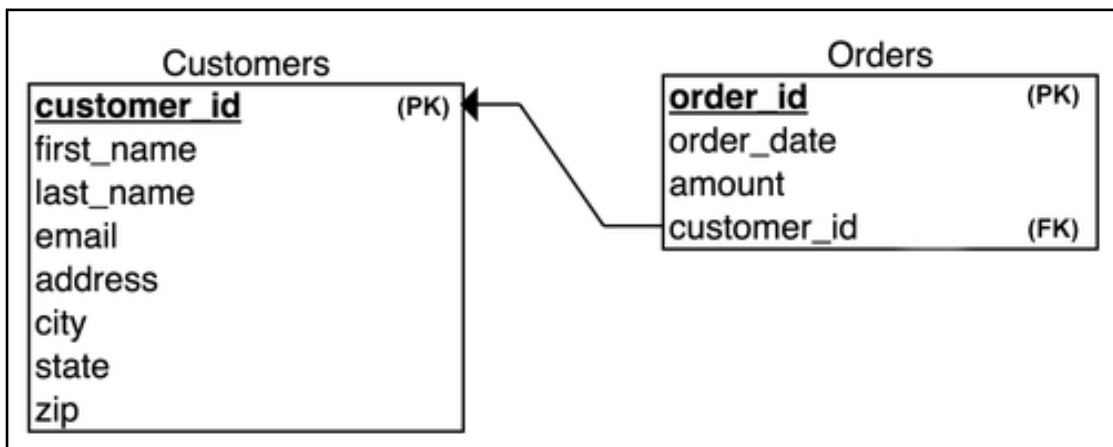
Δημιουργήθηκε, στις αρχές τις δεκαετίας του 1970, στα εργαστήρια της IBM από τους Ντόναλντ Τσάμπερλεϊν και Ρέιμοντ Μπους και, αρχικά, ονομαζόταν SEQUEL (Structured English Query Language). [67]

3.3.2 *Σχεσιακές βάσεις Δεδομένων*

Ο όρος «σχεσιακή» βάση δεδομένων αναφέρεται στην ενδεχόμενη «συγγένεια» δεδομένων από διαφορετικούς πίνακες αποθήκευσης αυτών των δεδομένων ή εγγραφών (records). Τα έγγραφα (records), σε μία βάση δεδομένων, αποθηκεύονται σε πίνακες, υπό μορφή μίας γραμμής, με τις στήλες της να είναι οι ιδιότητες της οντότητας.

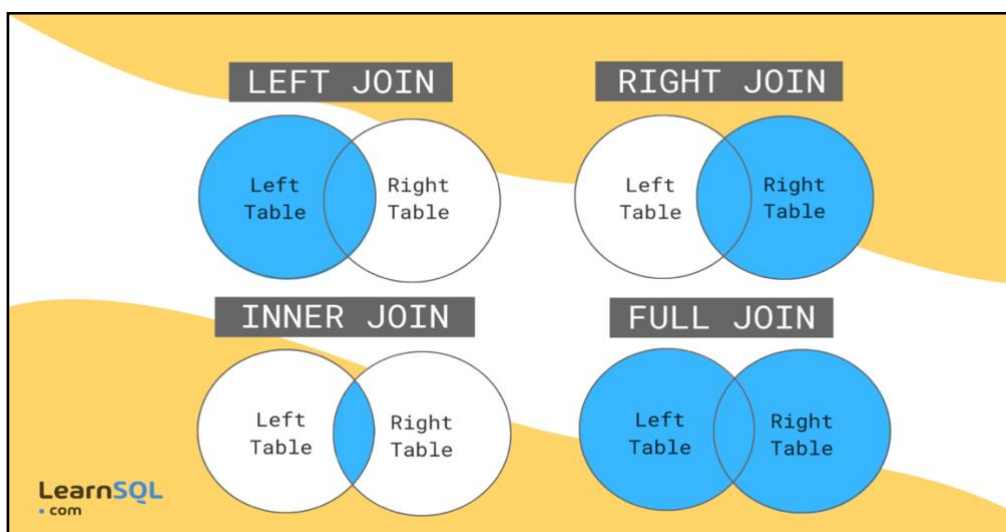
Κάθε γραμμή –υποχρεωτικά, εκτός από περιπτώσεις προηγμένου ελέγχου από το χρήστη – φέρει ένα μοναδικό αριθμό ταυτοποίησης, το οποίο ονομάζεται συνήθως ID. Αυτός ο αριθμός είναι μοναδικός και αυξάνεται για κάθε νέα εγγραφή στον πίνακα (auto-increment) και ονομάζεται Πρωτεύον Κλειδί (Primary Key ή PK). Πέρα από το Primary Key, ένας πίνακας δύναται να έχει και ξένα κλειδιά (Foreign Keys ή FK).

Όταν δύο οντότητες είναι συνδεδεμένες ή «συγγενικές», τότε σημαίνει πως η μία οντότητα έχει σαν μία στήλη της, ως Foreign Key, το Primary Key της άλλης οντότητας. Με την εντολή σύνδεσης JOIN, μπορεί να δημιουργηθεί μία ενιαία εγγραφή η οποία είναι το υπερσύνολο των δύο οντοτήτων. (Εικόνα 32)



Εικόνα 32. Απεικόνιση μίας σχέσης σε σχεσιακή βάση δεδομένων. [68]

Η γλώσσα SQL παρέχει πολλές λειτουργίες και δυνατότητες, όσον αφορά την αναζήτηση και την ένωση των δεδομένων, με το JOIN να είναι μία από τις πιο κοινές. Υπάρχουν πολλοί τύποι JOIN, με τους κυριότερους να είναι το INNER, OUTER, RIGHT και LEFT, ο καθένας με τη δική του λειτουργικότητα. (Εικόνα 33)



Εικόνα 33. Απεικόνιση τύπων ενώσεων (JOIN) σε σχεσιακή βάση δεδομένων. [69]

3.3.3 *Microsoft SQL Server*

Ο Microsoft SQL Server αποτελεί σύστημα διαχείρισης βάσεων δεδομένων (DataBase Management System – DBMS). Έχει αναπτυχθεί και προωθείται από τη Microsoft και, ενώ μέχρι πρόσφατα υπήρχε αποκλειστικά στα λειτουργικά συστήματά της, τα τελευταία έτη, χάρη στη στροφή της εταιρείας προς το λογισμικό ανοικτού κώδικα, έγινε διαθέσιμο και σε λογισμικά συστήματα Linux, αλλά και μέσω Docker.

Δεδομένου ότι αποτελεί σύστημα βάσης δεδομένων, διαθέτει τα ακόλουθα χαρακτηριστικά:

- Ποικιλία διασυνδέσεων με το χρήστη
- Ανεξαρτησία φυσικών και λογικών δεδομένων
- Βελτιστοποίηση ερωτημάτων
- Ακεραιότητα δεδομένων
- Έλεγχος ταυτόχρονης λειτουργίας
- Δημιουργία αντιγράφων ασφαλείας – για επαναφορά και ασφάλεια. [70]

Ο SQL Server κυκλοφόρησε για πρώτη φορά το 1989. Η σχεσιακή γλώσσα που χρησιμοποιεί ο SQL Server για την επικοινωνία των εφαρμογών με το DBMS ονομάζεται Transact – SQL (T – SQL) και αποτελεί διάλεκτο της σημαντικότερης γλώσσας βάσεων δεδομένων: της Δομημένης Γλώσσας Ερωτημάτων SQL (Structured Query Language). [70]

Η ανάκτηση των δεδομένων από τη βάση δεδομένων πραγματοποιείται με τη μορφή ερωτήσεων (queries) που υποβάλλονται από το χρήστη. [71]

Σε γλώσσα T – SQL, γράφονται οι αποθηκευμένες διαδικασίες (stored procedures), οι οποίες πρακτικά αφορούν μία αλληλουχία προτάσεων και επεκτάσεων SQL που αποθηκεύονται στο διακομιστή της βάσης δεδομένων και εκτελούνται αδιαίρετα. [70]

3.4 *Apache Superset*

3.4.1 *Προέλευση*

Το Apache Superset αναπτύχθηκε αρχικά από την Airbnb, έως ότου εντάχθηκε στο Apache Software Foundation, μία Μ.Κ.Ο. (Μη Κυβερνητική Οργάνωση) που ανήκει στην εταιρεία Apache, γνωστή και από τον Apache HTTP Server. [72]

Το Apache Software Foundation είναι μία κοινότητα προγραμματιστών ανοικτού κώδικα. Το λογισμικό που παράγεται διανέμεται, σύμφωνα με τους όρους της Άδειας Apache, και είναι δωρεάν και ανοικτού κώδικα (Free and Open-Source Software – FOSS). Τα έργα της Apache

χαρακτηρίζονται από μία συνεργατική διαδικασία ανάπτυξης και μία ανοιχτή και ρεαλιστική άδεια λογισμικού. [73]

3.4.2 Ορισμός

Το Apache Superset είναι ένα λογισμικό ανοιχτού κώδικα για ανάλυση Δεδομένων Μεγάλης Κλίμακας (Big Data) και ανήκει στην κατηγορία λογισμικού Business Intelligence (BI). Μέσω dashboard που παρέχει, μπορούν να απεικονιστούν πολλά δεδομένα, με τρόπο που να δίνει αξία στα δεδομένα, και παρέχουν, στο χρήστη, κατανόηση αυτών των δεδομένων, δυνατότητα παρατήρησης ανωμαλιών και γενικότερη επίβλεψη.

Χάρη στην προσέγγιση που έχει, χωρίς κωδικοποίηση στην πλοήγηση δεδομένων, επιτρέπει σε όλους τους χρήστες ευκολία στην πρόσβαση δεδομένων. Από τη μία πλευρά, οι χρήστες που είναι λιγότερο τεχνικοί βρίσκουν μία εύκολη διεπαφή για την αναζήτηση και την απεικόνιση δεδομένων. Από την άλλη πλευρά, οι προηγμένοι χρήστες απολαμβάνουν την ταχύτητα και την ευκολία κοινοποίησης του περιεχομένου που δημιουργούν. [74]

Τα κυριότερα χαρακτηριστικά και η κύρια λειτουργικότητα του Apache Superset είναι:

- Ένα σύνολο οπτικοποιήσεων των δεδομένων που περιλαμβάνει χάρτες, διαγράμματα, απεικόνιση γεωγραφικών συντεταγμένων, pivot πίνακες, heatmaps, boxplots και πολλά άλλα διαγράμματα που ανήκουν στην κατηγορία των Descriptive Analytics
- Ευέλικτος τρόπος ταυτοποίησης των χρηστών, με υποστήριξη για LDAP (Lightweight Directory Access Protocol), OpenID, OAuth. Τροποποίηση ρόλων και αδειών για κάθε χρήστη
- Παροχή συνδεσιμότητας με τις περισσότερες βάσεις δεδομένων που χρησιμοποιούν SQL
- Caching (προσωρινή αποθήκευση εντός μνήμης) δεδομένων, με διαδοχικές παραμέτρους χρονικού ορίου ανά αναφορά, πίνακα και βάση δεδομένων. Με αυτόν τον τρόπο, οι βάσεις δεδομένων αποδεσμεύονται από βαρύ φορτίο αναζήτησης δεδομένων και, παράλληλα, επιταχύνει τη φόρτωση σημαντικών dashboard
- Δυνατότητα τροποποίησης της εμφάνισης με χρήση CSS (Cascading Style Sheets) και Bootstrap. [74]

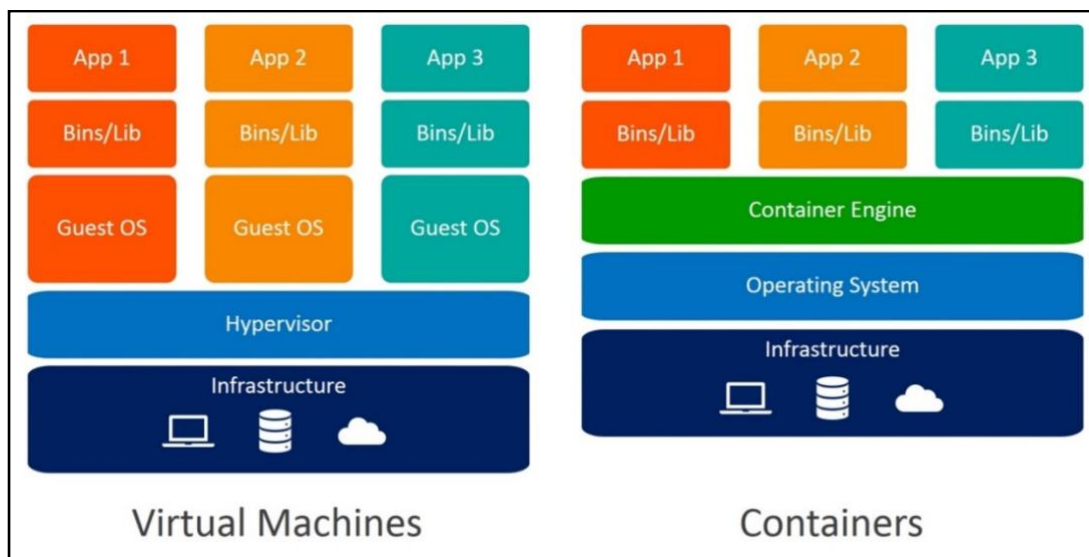
Στα πλαίσια της παρούσας διπλωματικής εργασίας, έγινε εκτεταμένη χρήση των Descriptive Analytics που παρέχει το Apache Superset, καθώς και χαρτών, heatmaps και boxplots.

3.5 Docker

3.5.1 Containers

Τα container είναι μία μορφή εικονικοποίησης (virtualization) του λειτουργικού συστήματος. Ένα μόνο container μπορεί να χρησιμοποιηθεί για την εκτέλεση οποιασδήποτε λογισμικής λύσης, από μία μικρή υπηρεσία μικροεφαρμογών έως μία μεγαλύτερη εφαρμογή. Μέσα σε ένα container, υπάρχουν όλα τα απαραίτητα εκτελέσιμα, binary (δυαδικός) κώδικας, βιβλιοθήκες και αρχεία διαμόρφωσης, προκειμένου να μπορέσει να λειτουργήσει αυτούσια.

Σε σύγκριση με τις παραδοσιακές προσεγγίσεις εικονοποίησης υπολογιστικών συστημάτων (virtualization), τα container δεν περιέχουν εικόνες (images) λειτουργικού συστήματος. Αυτό τα καθιστά περισσότερο ελαφριά και φορητά, με σημαντικά λιγότερα έξοδα. (*Εικόνα 34*)



Εικόνα 34. Απεικόνιση διαφορών Virtualization vs Containerization. [75]

Σε μεγαλύτερες αναπτύξεις εφαρμογών, πολλαπλά container μπορούν να αναπτυχθούν ως μία ή περισσότερες συστάδες container (stack). Η διαχείριση αυτών των συστάδων (stack) μπορεί να γίνεται από έναν εννορηστροτή, όπως το Kubernetes ή το Docker.

Τα container είναι ένας απλοποιημένος τρόπος για τη δημιουργία, τη δοκιμή, την ανάπτυξη και την αναδιάταξη εφαρμογών σε πολλά περιβάλλοντα, από τον τοπικό φορητό υπολογιστή ενός προγραμματιστή έως ένα κέντρο δεδομένων εσωτερικής εγκατάστασης, ακόμα και στο cloud.

Σε ένα Λειτουργικό Σύστημα (ΛΣ), είναι δυνατόν ο πυρήνας (kernel) να επιτρέπει την συνύπαρξη πολλαπλών απομονωμένων χώρων χρήστη (user-space instances), διαδικασία γνωστή και ως εικονικοποίηση σε επίπεδο ΛΣ (Operating-system-level virtualization ή containerization). Ένα τέτοιο απομονωμένο στιγμιότυπο ή container λειτουργεί ως κανονικός

υπολογιστής, με τους δικούς του πόρους, χωρίς να έχει επίγνωση των υπόλοιπων στιγμιότυπων και χωρίς να μοιράζεται τους πόρους του μαζί τους.

Τα σημαντικότερα πλεονεκτήματα των container είναι τα ακόλουθα:

1. Απομόνωση: μία εφαρμογή που λειτουργεί ως container είναι απομονωμένη από το περιβάλλον εκτέλεσης και το πραγματικό ΛΣ του τελικού χρήστη
2. Ελαφρά εκτέλεση: χρησιμοποιεί απευθείας τον πυρήνα του ΛΣ, εξοικονομώντας πόρους
3. Εύκολη εγκατάσταση, μεταφορά στην παραγωγή και migration
4. Συνεπέστερη λειτουργία: οι χρήστες γνωρίζουν ότι οι εφαρμογές σε container θα εκτελούνται το ίδιο, ανεξάρτητα από το πού αναπτύσσονται.
5. Καλύτερη και ευκολότερη ανάπτυξη εφαρμογών, επειδή συνδράμουν θετικά στην επιτάχυνση των κύκλων ανάπτυξης, δοκιμών και παραγωγής. [76]

Αντίθετα, τα σημαντικότερα μειονεκτήματά του είναι τα εξής:

1. Σε εξαιρετικά απαιτητικές εφαρμογές, η τεχνολογία των Containers υστερεί σε σχέση με την παραδοσιακή εγκατάστασή τους σε εξυπηρετητές (servers)
2. Τα διάφορα containers μοιράζονται το ίδιο Host Operating System και αυτό τα περιορίζει στην ίδια αρχιτεκτονική. [77]

3.5.2 Ορισμός

Το Docker αποτελεί μία πλατφόρμα διαχείρισης και εκτέλεσης containers ανοικτού κώδικα, η οποία αναπτύσσεται ενεργά και υποστηρίζεται από πολλούς cloud providers και πλατφόρμες cloud ή πλατφόρμες υποδομής ως υπηρεσία (Infrastructure as a service – IaaS). Η πρώτη της έκδοση ήταν στις 13 Μαρτίου του 2013. [78] Αναπτύχθηκε αρχικά σαν εσωτερικό project της εταιρείας DotCloud, η οποία μεταγενέστερα μετονομάστηκε σε Docker. Το Docker χρησιμοποιούσε LXC (Linux Containers) στα αρχικά στάδια και, αργότερα, αντικατέστησε το LXC με δική του βιβλιοθήκη, τη libcontainer. [79] Παρόλο που πρόκειται για μία σχετικά νέα πλατφόρμα, είναι η πλέον χρησιμοποιούμενη, για προσωπική χρήση, αλλά και για παραγωγικά συστήματα. [78]

Το Docker εισήγαγε ένα ολόκληρο οικοσύστημα για τη διαχείριση containers. Διαθέτει ένα repository από έτοιμα containers ή εικόνες (images), εντός του οποίου ανευρίσκονται πολλές εφαρμογές έτοιμες προς εκτέλεση, όπως βάσεις δεδομένων, web servers ή κάποιο έτοιμο λειτουργικό. [79] Λόγω της συνδεσιμότητας με έξυπνες συσκευές ή με το ευρύτερο Διαδίκτυο, καθώς και των δυνατοτήτων για τοπική αποθήκευση και επεξεργασία δεδομένων, είναι ιδιαίτερα χρήσιμο στην υποστήριξη εφαρμογών για το Διαδίκτυο των Πραγμάτων.

Το λογισμικό Docker είναι διαθέσιμο σε περιβάλλοντα Linux, Windows ή Mac.

3.5.3 Αρχείο *docker-compose.yml*

Το *docker-compose* είναι ένα χρήσιμο εργαλείο για εντοπισμό εφαρμογών Docker και χρησιμοποιείται για την εκτέλεση πολλαπλών container ως μία υπηρεσία. Κάθε ένα από τα container λειτουργεί μεμονωμένα, αλλά μπορεί να αλληλεπιδρά μεταξύ των υπολοίπων container της συστάδας (stack) όταν απαιτείται.

Το αρχείο *docker-compose* είναι πολύ εύκολο να γραφτεί σε μία scripting γλώσσα που ονομάζεται YAML (Yet Another Markup Language, και βασίζεται στο πρότυπο XML (Extensible Markup Language). Ένα άλλο σπουδαίο προτέρημα για το *docker-compose* είναι ότι οι χρήστες μπορούν να ενεργοποιήσουν όλες τις υπηρεσίες (container), χρησιμοποιώντας μία μόνο εντολή.

Τα κυριότερα οφέλη του *docker-compose* είναι τα παρακάτω:

- Δυνατότητα ανάπτυξης σύνθετων λύσεων σε έναν υπολογιστή
- Γρήγορη και εύκολη διαμόρφωση, χάρη στην ευκολία της γλώσσας YAML
- Υψηλή παραγωγικότητα, καθώς μειώνει το χρόνο που απαιτείται για την εκτέλεση εργασιών
- Ασφάλεια, επειδή όλα τα container είναι απομονωμένα το ένα από το άλλο. [80]

3.6 Προγραμματιστικές αρχές

3.6.1 Γλώσσα προγραμματισμού - C#

Η γλώσσα που επιλέχθηκε για την υλοποίηση της εργασίας είναι η C# (C-sharp). Είναι μία από τις υποστηριζόμενες γλώσσες του .NET (dotnet) framework και αναπτύχθηκε από τη Microsoft, ως μέρος του ολοκληρωμένου περιβάλλοντος ανάπτυξης (Integrated Development Environment ή IDE) του Visual Studio. [81]

Πρόκειται για μία διαδικαστική, αντικειμενοστραφή γλώσσα προγραμματισμού, η οποία βασίζεται στη C++, αλλά υιοθετεί πτυχές από Java και Delphi. Το συντακτικό της είναι πολύ μοντέρνο και θεωρείται γλώσσα γενικού σκοπού.

Για πολλά χρόνια, ήταν αποδέκτης αρνητικών σχολίων, λόγω της στενής σχέσης και εξάρτησης που είχε από το Λειτουργικό Σύστημα Microsoft Windows. Ωστόσο, από το 2016 και μετά, χάρη στην ανάπτυξη και την ενσωμάτωση του dotnet Core Framework, έχει δοθεί η δυνατότητα για διαπλατφορμική ανάπτυξη εφαρμογών και γνωρίζει μεγάλη άνθιση και αποδοχή.

Χρησιμοποιείται ευρέως για ανάπτυξη εφαρμογών Διαδικτύου (Web Application), εφαρμογών Windows και παιχνιδιών μέσω της μηχανής Unity.

3.6.2 ASP.NET Core Framework

Οι διαδικτυακές υποδομές, που υλοποιήθηκαν, στα πλαίσια της λύσης λογισμικού, αναπτύχθηκαν σε Microsoft ASP.NET Core.

Το ASP.NET Core είναι ένα framework ανοικτού κώδικα και εκτελέσιμο σε πολλαπλές πλατφόρμες (cross-platform), για τη δημιουργία σύγχρονων cloud-based εφαρμογών συνδεδεμένων στο Διαδίκτυο, όπως διαδικτυακές εφαρμογές, εφαρμογές IoT αλλά και για απλή χρήση σε υπολογιστές.

Οι εφαρμογές ASP.NET Core εκτελούνται σε .NET Core (dotnet core), ένα runtime εκτέλεσης εφαρμογών. Είχε σχεδιαστεί για να παρέχει ένα βελτιστοποιημένο πλαίσιο ανάπτυξης για εφαρμογές που αναπτύσσονται στο cloud ή εκτελούνται εσωτερικά. Αποτελείται από modular components με ελάχιστη επιβάρυνση, ώστε να διατηρεί την ευελιξία κατά την κατασκευή εφαρμογών. Όλες οι εφαρμογές που αναπτύσσονται είναι δυνατόν να αναπτυχθούν, αλλά και να εκτελεστούν σε Windows, Mac και Linux.

3.6.3 Ασύγχρονος προγραμματισμός

Παραδοσιακά, ο προγραμματισμός είναι μία διαδοχική εκτέλεση εντολών με σύγχρονες κλήσεις συστήματος, που αποκλείουν εντελώς το νήμα εκτέλεσης (Main Thread) μέχρι να ολοκληρωθεί μία λειτουργία, όπως η εγγραφή ενός αρχείου στο δίσκο. Όσο το νήμα εκτέλεσης ενός προγράμματος παραμένει αποκλεισμένο, όλες οι ενέργειες και η επαφή με το πρόγραμμα, όπως το πάτημα ενός κουμπιού, δεν ανταποκρίνονται.

Σε συστήματα με βεβαρμένο φόρτο εργασίας, το πρόγραμμα περιμένει σχεδόν όλη την ώρα για την εκτέλεση μίας ενέργειας: είτε το δίσκο, είτε τη βάση δεδομένων, το δίκτυο ή τη διεπαφή με το χρήστη. Σε συστήματα με ελαφρύ φόρτο εργασίας, αυτό μπορεί να επιλυθεί δημιουργώντας ένα νέο νήμα (thread) για κάθε δράση αποκλεισμού (thread-block). Ενώ το ένα νήμα «κοιμάται», το άλλο τρέχει. Όμως σε μεγάλα συστήματα, κάτι τέτοιο θα ήταν ανέφικτο.

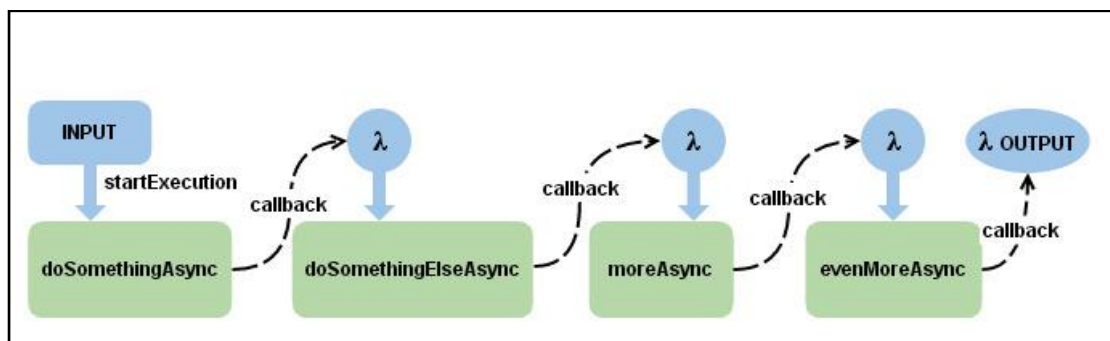
Τη λύση στο παραπάνω πρόβλημα έρχεται να δώσει ο ασύγχρονος προγραμματισμός. Μέσω έξυπνων μεθόδων και αλγορίθμων, επιτυγχάνεται η εκτέλεση μίας λειτουργίας σε κατάσταση μη αποκλεισμού του νήματος εκτέλεσης. Οι δύο τρόποι, που εξετάστηκαν και χρησιμοποιήθηκαν ευρέως στην υλοποίηση, αναλύονται παρακάτω.

3.6.3.1 Επανάκληση (Callback)

Μία κοινή λύση στο πρόβλημα του ασύγχρονου προγραμματισμού είναι η χρήση των προγραμματιστικών callback ή επανακλήσεων.

Με αυτήν τη μέθοδο, μία εντολή ξεκινά να εκτελείται σε νέο νήμα (Thread), ενώ το κύριο νήμα εκτέλεσης (Main Thread) παραμένει μη αποκλεισμένο. Ο καλών δεν περιμένει την

ολοκλήρωση της διαδικασίας εκείνη τη στιγμή, αλλά ενημερώνεται για την ολοκλήρωσή της σε δεύτερο χρόνο. Το νέο νήμα, που δημιουργήθηκε από το Main Thread, αναλαμβάνει να ολοκληρώσει τη διαδικασία και, με την περάτωσή της, να ειδοποιήσει τον αρχικό καλούντα. (Εικόνα 35)



Εικόνα 35. Παράδειγμα επανάκλησης. [82]

3.6.3.2 Async/Await

Με τη χρήση του Task-Based Asynchronous Pattern (TAP) προγραμματιστικού μοντέλου, μία πολύ χρονοβόρα διαδικασία μπορεί να χωριστεί σε μικρότερα Tasks, με τη λογική του «διαίρει και βασίλευε» (divide-and-conquer algorithm).

Με τις δεσμευμένες λέξεις (keywords) `async` και `await`, υποδεικνύεται στο πρόγραμμα ποια Tasks είναι χρονοβόρα (long-running). Το πρόγραμμα ξεκινά την εκτέλεσή τους, σε νέο νήμα (thread), και προχωρά στην υπόλοιπη εκτέλεση του κώδικα. Όταν πλέον το αποτέλεσμα του Task κρίνεται αναγκαίο, τότε χρησιμοποιείται η δεσμευμένη λέξη `await`, που υποδεικνύει πως πρέπει να περιμένει ο υπολογιστής για την εκτέλεση του κώδικα.

Είναι, ουσιαστικά, ένας ενορχηστρωτής εντολών: όλα τα long running tasks ξεκινούν να εκτελούνται παράλληλα και, με την υπόδειξη της λέξης `await`, η εκτέλεση θα περιμένει για το τελικό αποτέλεσμα.

3.6.4 API

Διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface, API) ονομάζεται η διεπαφή που ένα υπολογιστικό σύστημα ή εφαρμογή παρέχει, προκειμένου να επιτρέψει να γίνουν προς αυτό αιτήσεις από άλλα προγράμματα, για εκτέλεση εργασιών ή ανταλλαγή δεδομένων. [83]

Το API είναι ένα σύνολο παραδοχών και ορισμών, που καθορίζουν τον τρόπο με τον οποίο μία υπηρεσία «καλείται» μέσα από ένα προγραμματιστικό περιβάλλον. Το API επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στις λειτουργικότητες ενός προγράμματος μέσα από

καλά καθορισμένες δομές δεδομένων. Ουσιαστικά, το API περιγράφει τη λειτουργία και τη δομή ενός μέρους ή ολόκληρου προγράμματος και ορίζει τη διεπαφή του με τον έξω κόσμο. Έχοντας την τεκμηρίωση ενός API, είναι δυνατός ο προγραμματισμός ενός συστήματος χωρίς να διατίθενται άλλου είδους πληροφορίες. [83]

4

Αρχιτεκτονική – Υλοποίηση

Στο Κεφάλαιο 3, παρουσιάστηκαν τα τεχνολογικά μέσα που χρησιμοποιήθηκαν για την περάτωση της υλοποίησης. Στο Κεφάλαιο 4, θα γίνει μία λεπτομερής ανάλυση της γενικότερης αρχιτεκτονικής που χρησιμοποιήθηκε και του τρόπου με τον οποίο συνδυάζονται μεταξύ τους οι τεχνολογικές λύσεις που αναφέρθηκαν.

4.1 Εγκατάσταση

Η λογισμική λύση που έχει αναπτυχθεί υλοποιήθηκε με τη λογική των image containers και της ευκολίας που παρέχει το Docker. Τα κυριότερα πλεονεκτήματα έχουν ήδη αναφερθεί. Ωστόσο, αξίζει να επισημανθεί πως η φορητότητα, η ευκολία στη μεταφορά και η αναπαραγωγή (replication) αυτούσιου του συστήματος αποτέλεσαν τους σημαντικότερους παράγοντες για αυτήν την επιλογή.

Το αρχείο **docker-compose.yml** φροντίζει να κατεβάσει όλα τα προαπαιτούμενα container images για να μπορέσει να λειτουργήσει η εφαρμογή και δε χρειάζεται καμία τροποποίηση. Το αρχείο **.env**, που εμπεριέχεται στον κώδικα της εφαρμογής, περιέχει συγκεντρωμένες όλες τις παραμετροποιήσεις που δύναται να κάνει ο χρήστης. Οι αλλαγές αυτές αφορούν κυρίως κωδικούς και ονοματολογίες βάσεων δεδομένων.

Συνεπώς, το μόνο προαπαιτούμενο, για να μπορέσει κάποιος να εγκαταστήσει την εφαρμογή σε νέα υποδομή, είναι η ύπαρξη του Docker Engine. Στη συνέχεια, τα απαραίτητα βήματα είναι τα παρακάτω:

- Λήψη του project από το GitHub (source control)²

```
1. git clone https://github.com/petsasj/FIWAREHub.git
```

Τμήμα Κώδικα 6. Εντολή αντιγραφής Git Repository.

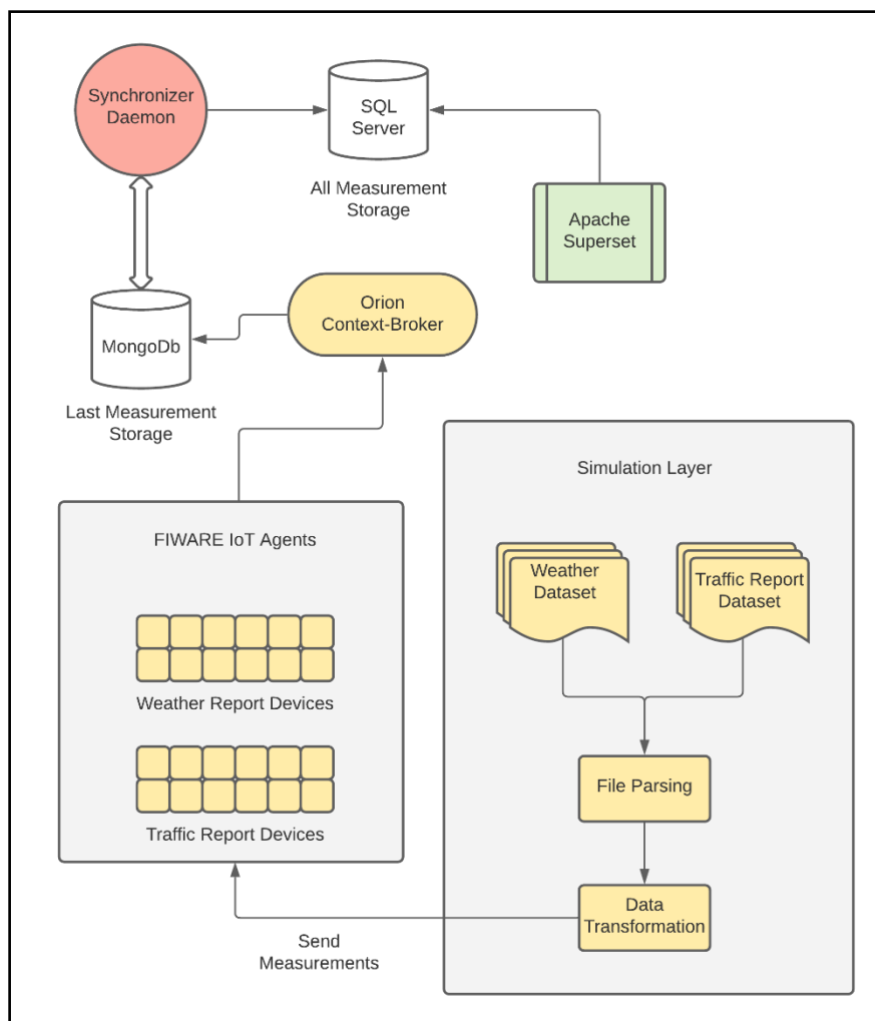
² <https://github.com/petsasj/FIWAREHub>

- Λήψη αρχείων zip που περιέχουν τα δύο Dataset³⁴
- Τοποθέτηση των δυο zip αρχείων στο φάκελο **FIWAREHub.Datasets**
- Εκτέλεση δύο εντολών μέσω του CLI του Λειτουργικού Συστήματος, μέσα στο φάκελο του project

```
1. docker-compose build
2. docker-compose up -d
```

Τμήμα Κώδικα 7. Εντολή ενεργοποίησης Docker συστάδας (Docker stack).

4.2 Σχεδιάγραμμα αρχιτεκτονικής



Εικόνα 36. Αρχιτεκτονική της εφαρμογής.

³ <https://www.kaggle.com/sobhanmoosavi/us-accidents>

⁴ <https://www.kaggle.com/sobhanmoosavi/us-weather-events>

Στην *Εικόνα 36*, παρουσιάζεται, υπό μορφή διαγράμματος, η γενικότερη αρχιτεκτονική της εφαρμογής, το οποίο είναι χωρισμένο σε περαιτέρω τμήματα.

4.2.1 Επίπεδο Προσομοίωσης (Simulation Layer)

Η λογική της υλοποίησης της εφαρμογής στηρίζεται στην ύπαρξη μίας αδιάκοπης ροής δεδομένων, προερχόμενη από πολλές συσκευές IoT. Επειδή αυτό είναι ανέφικτο να αναπαραχθεί σε πειραματικά πλαίσια, δημιουργήθηκε ένα εικονικό επίπεδο προσομοίωσης.

Το επίπεδο αυτό, όπως υπονοεί και το όνομά του, προσομοιώνει την ύπαρξη – από πολύ λίγων μέχρι, ουσιαστικά, άπειρων – συσκευών. Στα πλαίσια της συγκεκριμένης υλοποίησης, αυτό επιτεύχθηκε μέσω άντλησης δεδομένων από δύο πραγματικά Dataset και τροφοδότησή τους στις εικονικές συσκευές. Η ανάλυση των δεδομένων που χρησιμοποιήθηκαν και ο τρόπος ανάλυσής τους θα γίνουν στο Κεφάλαιο 5, όπου παρουσιάζεται η Περίπτωση Χρήσης (Use Case). Ωστόσο, για λόγους εννοιολογικής συνάφειας, αναφέρεται πως τα δύο Dataset πραγματεύονται δεδομένα καιρικών συνθηκών και δεδομένα τροχαίων ατυχημάτων αντίστοιχα.

4.2.1.1 Ανάλυση Dataset (Dataset Parsing)

Το πρώτο βήμα στην υλοποίηση είναι η επεξεργασία των δύο Dataset.

Η εφαρμογή φροντίζει να αποσυμπιέσει τα δυο zip αρχεία που τοποθετήθηκαν στο φάκελο FIWAREHub.Datasets.

Ακολουθεί μία διαδικασία που ονομάζεται προγραμματιστικά **data parsing**. Τα δύο αρχεία είναι σε μορφή Τιμών Διαχωρισμένων με Κόμμα ή κοινώς Comma Separated Values (CSV), μία μορφή αρχείου κειμένου που επιτρέπει την ομοειδή απεικόνιση πολλών δεδομένων σε συνοπτική και συνεκτική μορφή.

Στον αντικειμενοστραφή προγραμματισμό, κάθε οντότητα αποτυπώνεται σαν μία προγραμματιστική κλάση (class) ή μοντέλο (model), με τις ιδιότητές της (class properties) να συγκεντρώνουν την πληροφορία. Άρα, για να αξιοποιηθούν τα δεδομένα των Datasets, πρέπει να αποτυπωθούν σε προγραμματιστικές κλάσεις (class).

Τα παραπάνω φροντίζει να διεκπεραιώσει η εφαρμογή FIWAREHub.Parsers⁵.

Μετατρέποντας λοιπόν το CSV αρχείο, έχουμε μία λίστα από οντότητες που έχουν τις πληροφορίες που αποτελούν το αντικείμενο του ενδιαφέροντος. Οι οντότητες, μία για το αρχείο καιρικών συνθηκών και μία για τα τροχαία ατυχήματα, ονομάζονται **DatasetAccidentReport.cs** και **DatasetWeatherEvent.cs** αντίστοιχα.

⁵ <https://github.com/petsasj/FIWAREHub/blob/master/FIWAREHub.Parsers>

Οι παραπάνω οντότητες βρίσκονται στο project `FIWAREHub.Models\ParserModels`⁶. Το συγκεκριμένο project λειτουργεί ως κοινή πηγή όλων των κλάσεων, το οποίο είναι συνήθης πρακτική στον αντικειμενοστραφή προγραμματισμό.

Έχοντας, πλέον, την πληροφορία σε μορφή κλάσεων/μοντέλων, μπορεί να ξεκινήσει ο μετασχηματισμός των δεδομένων. Στη συγκεκριμένη περίπτωση, η πληροφορία παρέμεινε αυτούσια, αλλά έγιναν κάποιες μετατροπές μονάδων. Επειδή τα Dataset που χρησιμοποιήθηκαν προέρχονται από τις Η.Π.Α., τα δεδομένα αποτυπώνονται σε Imperial μονάδα μέτρησης (μίλια αντί χιλιομέτρων, Fahrenheit αντί βαθμών Κελσίου και άλλα). Αυτά μετατράπηκαν σε μονάδα μέτρησης SI (Système international).

4.2.1.2 Αποστολή δεδομένων από συσκευές

Όπως αναφέρθηκε και στο Κεφάλαιο 4, ο Fiware IoT Agent, που χρησιμοποιήθηκε για την υλοποίηση της εργασίας, είναι ένας μεταγλωττιστής επικοινωνιών μεταξύ μίας IoT συσκευής και του Context Broker. Οι συσκευές επικοινωνούν, με το δικό τους πρότυπο και πρωτόκολλο με τον IoT Agent, και αυτός φροντίζει να μεταφέρει την πληροφορία στον Context Broker για αποθήκευση. Συνεπώς, οι IoT Agent που επιλέχθηκαν – σύμφωνα με το πρότυπο μορφοτύπου που υποστηρίζουν, θα πρέπει να συνδεθούν με εικονικές συσκευές, το ρόλο των οποίων θα φροντίσει να αναλάβει μία cURL κλήση με νέα δεδομένα. Από εκεί προκύπτει και ο χαρακτηρισμός «προσομοίωση».

Πρώτο βήμα είναι η αρχικοποίηση (provision) των συσκευών IoT, το οποίο φροντίζει να κάνει η εφαρμογή αυτόματα στην κλάση **ProvisionController.cs**⁷, μέσω αποστολής cURL (Client URL) κλήσης στον IoT Agent. Ενδεικτικά, παρέχεται παράδειγμα του τρόπου εφαρμογής της αρχικοποίησης (provision) των συσκευών που βασίζονται σε πρότυπο επικοινωνίας τύπου JSON.

```
1. // POST Model for Provision of multiple devices
2. var json = new ProvisionDevicesModel
3. {
4.     Devices = FIWAREUrls.WeatherDeviceIds.Select(id => new Device
5.     {
6.         DeviceId = id,
7.         EntityName = $"urn:ngsi-ld:{id}",
```

⁶ <https://github.com/petsasj/FIWAREHub/blob/master/FIWAREHub.Models>

⁷ <https://github.com/petsasj/FIWAREHub/blob/master/FIWAREHub.Web/Controllers/Api/ProvisionController.cs>


```

8.     EntityType = "weatherReport",
9.     Attributes = entityAttributes.Select(ea => new Attribute
10.    {
11.        Name = ea.name,
12.        ObjectId = ea.objectId,
13.        Type = ea.type
14.    }).ToList()
15.    }).ToList()
16. };
17.
18. using var fiwareClient = new FIWAREClient();
19. var response = await fiwareClient.SendJson(HttpMethod.Post,
    $"{FIWAREUrls.IoTJsonNorth}{FIWAREUrls.DeviceProvisionPath}", json);

```

Τμήμα Κώδικα 8. Αρχικοποίηση, μέσω διεπαφής API, για συσκευές JSON.

Με παρόμοιο τρόπο, αρχικοποιούνται και οι συσκευές που βασίζονται στο πρότυπο Ultralight. Συνολικά, στην εφαρμογή, έχει επιλεγεί να υπάρχουν 40 συσκευές IoT, 20 που μεταφέρουν δεδομένα καιρού, χρησιμοποιώντας το πρότυπο Ultralight, και άλλες 20 που μεταφέρουν δεδομένα τροχαίων ατυχημάτων, χρησιμοποιώντας το πρότυπο JSON.

Με τη χρήση της North TCP θύρας του IoT Agent, αποστέλλονται νέα δεδομένα από την εικονική συσκευή (μέσω HttpRequest), τα οποία καταλήγουν στον Context Broker. Η αποστολή τόσο μεγάλου αριθμού δεδομένων, τα οποία μάλιστα δύναται να ανανεώνονται και συνεχόμενα, απαιτεί ασύγχρονη προγραμματιστική προσέγγιση για αποφυγή απώλειας δεδομένων, αλλά και απόδοσης. Το έργο αυτό καλείται να εκπονήσει η κλάση **FIWAREMeasurementsSubmitter.cs**⁸. Για κάθε νέο δεδομένο που προστίθεται στη λίστα των οντοτήτων καιρού ή τροχαίων ατυχημάτων, φροντίζει να το αποστείλει, ως μία εικονική συσκευή, στον εκάστοτε IoT Agent. Ανά τακτά χρονικά διαστήματα (σε περιόδους των 5 λεπτών), ενημερώνεται και μία οντότητα στη βάση δεδομένων, που περιγράφει την πρόοδο της διαδικασίας.

Τέλος, στο project FIWAREHub.Web⁹, πέραν των ιστοσελίδων που εξυπηρετούν την απεικόνιση των δεδομένων, έχουν δημιουργηθεί και διεπαφές εφαρμογών προγραμματισμού (API) που εξυπηρετούν την εκκίνηση επιμέρους τμημάτων της εφαρμογής.

⁸ <https://github.com/petsasj/FIWAREHub/blob/master/FIWAREHub.Web/Services/FIWAREMeasurementsSubmitter.cs>

⁹ <https://github.com/petsasj/FIWAREHub/tree/master/FIWAREHub.Web>

Η παρακάτω διεπαφή φροντίζει να εκτελεστεί ο κώδικας της διαδικασίας προσομοίωσης και όλα τα επιμέρους τμήματα του κώδικα που αναφέρθηκαν παραπάνω.

```
1. curl --location -g --request GET  
   'https://192.168.10.254:8090/iotmeasurement/SynchronizeReadings'
```

**Τμήμα Κώδικα 9. Διεπαφή API για εκκίνηση συγχρονισμού
προσομοίωσης και αποστολής δεδομένων.**

4.2.2 Συγχρονισμός Δεδομένων

Τόσο οι FIWARE IoT Agents, όσο και ο Orion Context Broker χρησιμοποιούν ως μέσο αποθήκευσης δεδομένων τη MongoDB.

Με την αρχικοποίηση των εικονικών συσκευών, δημιουργούνται 2 συλλογές (collections) στη βάση δεδομένων. Η μία αφορά τις συσκευές με ίδιο πρότυπο μορφοτύπου επικοινωνίας – είτε JSON, είτε Ultralight – και η άλλη συλλογή που δημιουργείται από τον Orion Context Broker. Στη διαμόρφωση (configuration) της παρούσας εργασίας, που έγινε μέσω του docker-compose.yml και του αρχείου .env, επιλέχθηκε οι συσκευές Ultralight και JSON να επικοινωνούν με την ίδια συλλογή. Στη συλλογή που δημιουργεί ο Orion Context Broker, παράγεται μία οντότητα για κάθε συσκευή που έχει αρχικοποιηθεί (provision) και αποθηκεύει δεδομένα που έχει αποστείλει η συσκευή.

Οι συσκευές IoT, μόλις αποστείλουν δεδομένα, είτε μέσω της προσομοίωσης που παρουσιάστηκε παραπάνω, είτε μέσω πραγματικών συσκευών και δεδομένων, επικοινωνούν με τον Orion Context Broker, μέσω της North TCP θύρας του IoT Agent. Αυτός, με τη σειρά του, αναλαμβάνει να ενημερώσει τα δεδομένα στη MongoDB.

4.2.2.1 Το πρόβλημα

Όταν αρχικοποιείται μία συσκευή, δημιουργεί μία μοναδική εγγραφή στις δύο συλλογές στη MongoDB.

Η μία οντότητα, στη συλλογή των IoT Agents, περιγράφει τη μοναδική ταυτότητα (ID) της συσκευής, τις ιδιότητες (properties) και τον τύπο των δεδομένων που αναμένει να λάβει ο Agent. **(Εικόνα 37)** Σε αυτήν τη συλλογή, οι οντότητες δε φέρουν στοιχεία. Ο μόνος λόγος ύπαρξης του schema της οντότητας κάθε συσκευής είναι για έλεγχο ορθότητας αποστολής δεδομένων, όσον αφορά τον τύπο τους (type checking). Εάν, για παράδειγμα, σε ένα αριθμητικό πεδίο σταλεί δεδομένο κείμενο, τότε θα παρουσιαστεί μήνυμα λάθους.

Document	Data	Type
[1] (id="5fa9093c9ee384f8346e591...")		Document
_id	5fa9093c9ee384f8346e591e	ObjectId
active		Array
[0]		Document
object_id	rt	String
name	ReportTime	String
type	string	String
[1]		Document
object_id	we	String
name	WeatherEvent	String
type	string	String
[2]		Document
object_id	sv	String
name	Severity	String
type	string	String
[3]		Document
object_id	owcd	String
name	OriginalWeatherConditionDescription	String
type	string	String
[4]		Document
object_id	tmp	String
name	Temperature	String
type	double	String

Εικόνα 37. Απεικόνιση δεδομένων οντότητας συλλογής IoT Agent.

Η οντότητα που δημιουργείται στον Orion Context Broker επεκτείνει, ουσιαστικά, την οντότητα που αναφέρθηκε παραπάνω, αυτή δηλαδή των συσκευών που είναι συνδεδεμένες με τους IoT Agents, καθώς και σε αυτή εμπεριέχονται οι τύποι των πληροφοριών που απαρτίζουν τα δεδομένα που αποστέλλει μία συσκευή. Επιπρόσθετα, όμως, εμπεριέχει και τις τιμές των δεδομένων για κάθε ιδιότητα. (*Εικόνα 38*)

Document	Data	Type
[1] (id="{ "id" : "urn:ngsi-Id:Weat...")		Document
_id		Document
id	urn:ngsi-Id:Weather05	String
type	weatherReport	String
servicePath	/	String
attrNames		Array
attrs		Document
ReportTime		Document
WeatherEvent		Document
value	Clear	String
type	string	String
md		Document
mdNames		Array
creDate	1604913468.6128368	Double
modDate	1604921539.1917822	Double
Severity		Document
OriginalWeatherConditionDes...		Document
Temperature		Document
WindChill		Document
Humidity		Document
value	74.0	Double
type	double	String
md		Document

Εικόνα 38. Απεικόνιση δεδομένων οντότητας συλλογής Orion Context Broker.

Δεδομένων των προαναφερθέντων, μπορεί να επισημανθεί μία σοβαρή έλλειψη για την υλοποίηση της συγκεκριμένης εργασίας: ο Orion Context Broker κρατάει **μόνο** την τελευταία τιμή των δεδομένων. Επειδή κρατάει μία μοναδική εγγραφή για κάθε συσκευή, αυτή

αναεώνεται με τις τιμές που έχει στείλει η συσκευή την τελευταία φορά που επικοινωνήσε με τον Orion Context Broker. Δεν παρέχεται καθόλου ιστορικότητα των δεδομένων.

4.2.2.2 Η λύση – Synchronizer Daemon

Το παραπάνω πρόβλημα κλήθηκε να λύσει η εφαρμογή **FIWAREHub.SynchronizerDaemon**¹⁰. Όπως παρατηρείται από το όνομα, πρόκειται για ένα δαίμονα. Στον προγραμματισμό, ο δαίμονας είναι ένα υπολογιστικό πρόγραμμα που αναλαμβάνει να περατώνει εργασίες στο παρασκήνιο, χωρίς να απαιτείται συναναστροφή με το χρήστη.

Συνεπώς, αυτό που έχει αναλάβει η συγκεκριμένη εφαρμογή είναι να συγχρονίσει τα δεδομένα και να διατηρήσει ιστορικό για όλες τις συσκευές.

Η συγκεκριμένη εφαρμογή βασίζεται στο **Change Stream API**, που παρέχει η MongoDB. Μόλις εκτελεστεί, ανοίγει μία σύνδεση με τη MongoDB, δημιουργεί ένα κέρσορα (cursor) και παρακολουθεί για αλλαγές στις οντότητες.

```
1. // Helper class for connection to MongoDB
2. var orionContext = new OrionContext();
3.
4. // Change Stream API of MongoDB (based on replica set oplog)
5. // This is an async task and creates a thread for each change
6. using (var cursor = await orionContext.Entities.WatchAsync())
7. {
8. }
```

Τμήμα Κώδικα 10. Αρχικοποίηση κέρσορα (cursor) για παρακολούθηση αλλαγών σε MongoDB.

Κάθε αλλαγή, που πραγματοποιείται στη μοναδική οντότητα της συσκευής στη συλλογή του Orion Context Broker, παρακολουθείται από το συγκεκριμένο κέρσορα και, με τη μορφή προγραμματιστικού callback, ειδοποιείται η εφαρμογή για αλλαγές και τις τιμές τους. Ακολούθως, αποθηκεύει αυτές τις μεταβολές σε μία βάση δεδομένων που επιλέγεται. Στην προκειμένη περίπτωση, επιλέχθηκε ο Microsoft SQL Server.

Ενδεχομένως, η παραπάνω υλοποίηση να ακούγεται απλή. Ωστόσο, για να επιτευχθεί η απρόσκοπτη λειτουργία των συστημάτων, ο ακριβής συγχρονισμός – δίχως απώλειες δεδομένων – όταν ένα σύστημα δέχεται πολύ μεγάλο αριθμό αλλαγών δεδομένων, απαιτεί ιδιαίτερη προσέγγιση.

¹⁰ <https://github.com/petsasj/FIWAREHub/tree/master/FIWAREHub.SynchronizerDaemon>

Ενδεικτικά, αναφέρεται πως, στο επίπεδο προσομοίωσης που αναφέρθηκε παραπάνω, μέσα σε λίγες ώρες, έπρεπε αρχικά να αποσταλούν στους IoT Agents και μετέπειτα να συγχρονιστούν, μέσω του δαίμονα, παραπάνω από 1 εκατομμύριο εγγραφές. Με τόσα πολλά δεδομένα και με τη συχνότητα των αλλαγών να ξεπερνούν τις 70 προσθήκες και συγχρονισμούς το δευτερόλεπτο, η αποδοτικότητα του προγράμματος έπρεπε να είναι υψηλή.

Για την επίτευξη αυτού, έπρεπε όλες οι ενέργειες να είναι ασύγχρονες. Επιτρέποντας στο νήμα εκτέλεσης (Main Thread) της εφαρμογής να αναλάβει περισσότερο επικοινωνικές λειτουργίες, όπως:

- Ενημέρωση για λήψη δεδομένων μέσω μηνυμάτων
- Εκτέλεση προγραμματιστικού χρονομέτρου για περιοδικό συγχρονισμό με τον SQL Server
- Διαχείριση των Concurrent λιστών, που χρησιμοποιούνται για thread-safe αποθήκευση δεδομένων
- Πρόβλεψη για αποτροπή σφαλμάτων
- Συχνός έλεγχος ορθότητας και ομαλής λειτουργίας των υποσυστημάτων

Τις ασύγχρονες λειτουργίες αναλαμβάνει η κλάση **Daemon.cs**¹¹. Μόλις λάβει ειδοποίηση από τον κέρσορα πως νέα δεδομένα εισήχθησαν στη βάση, αναλαμβάνει να διεκπεραιώσει κάποιες εργασίες:

1. Η πληροφορία που λαμβάνει είναι στη μορφή BSON. Όπως αναφέρθηκε και στο Κεφάλαιο 3, το BSON είναι μία μορφή, η οποία βασίζεται στο JSON, αλλά με ελαφρώς διαφοροποιημένο συντακτικό. Επομένως, το πρώτο βήμα είναι να μετατραπεί η πληροφορία που λαμβάνεται μέσω του BSON σε προγραμματιστικό μοντέλο (model)
2. Το μοντέλο πρέπει να αναγνωρίσει εάν η πληροφορία που έχει λάβει αφορά δεδομένα καιρού ή δεδομένα τροχαίου ατυχήματος
3. Μετατροπή δεδομένων σε οντότητα προς εισαγωγή στον SQL Server
4. Προσθήκη στην Concurrent λίστα οντοτήτων.

```
1. // creates instance of DTO Object and adds to ConcurrentQueue
2. if (entityTypeEnum == EntityTypeEnum.WeatherReport)
3. {
4.     Report($"Received Weather Report from device {deviceId}");
5. }
```

¹¹ <https://github.com/petsasj/FIWAREHub/blob/master/FIWAREHub.SynchronizerDaemon/Daemon.cs>

```

6.     // Deserialize to Typed object
7.     var weatherUpdate =
    JsonSerializer.Deserialize<WeatherReportUpdate>(change.UpdateDescription.Update
    dFields);
8.
9.     if (weatherUpdate == null)
10.        continue;
11.
12.    var uow = await getUnitOfWorkAsync();
13.
14.    // Delay to ensure nothing gets added to lists while SQL Save Operation is
    happening
15.    while (_unitOfWorkLock || _unitOfWork?.IsObjectsSaving == true ||
    uow.IsObjectsSaving)
16.    {
17.        Report("Lists are locked, waiting");
18.        await Task.Delay(50);
19.    }
20.
21.    var weatherReport = new WeatherReport(uow, weatherUpdate) { DeviceId =
    deviceId?.ToString() };
22.
23.    WeatherReports.Enqueue(weatherReport);
24. }

```

Τμήμα Κώδικα 11. Διαχείριση αλλαγών από MongoDB.

Οι Concurrent λίστες οντοτήτων, ανά τακτικά χρονικά διαστήματα, πρέπει να καθαρίζουν (clear) για να μπορέσει το σύστημα να παραμείνει λειτουργικό και να μη χρησιμοποιεί πόρους που δε χρειάζεται. Για να διασφαλιστεί ότι η πληροφορία που λαμβάνεται θα καταγραφεί και δε θα χαθεί μέσω της διαδικασίας καθαρισμού, έχει προβλεφθεί μηχανισμός κλειδώματος του Main Thread, που αναλαμβάνει να κάνει την εγγραφή στον SQL Server. Όσες πληροφορίες λαμβάνονται στο μεσοδιάστημα, όσο δηλαδή το σύστημα είναι σε κατάσταση εγγραφής – κλειδώματος, παραμένουν σε νέο ενεργό thread, μέχρι να αποδεσμευτεί το κλείδωμα και να καθαριστεί η λίστα από ήδη εγγεγραμμένες οντότητες.

```

1.     _unitOfWorkLock = true;
2.
3.     Report("SQL Syncing Started");
4.

```

```

5.     if (WeatherReports.Any() || RoadTrafficReports.Any())
6.     {
7.         // Diagnostic stopwatch to inform elapsed time for update
8.         var stopwatch = new System.Diagnostics.Stopwatch();
9.         stopwatch.Start();
10.
11.        var uow = await getUnitOfWorkAsync(true);
12.
13.        // SQL Save
14.        Report("Saving items in SQL");
15.        await uow.CommitChangesAsync();
16.
17.        stopwatch.Stop();
18.        var elapsedTime = stopwatch.ElapsedMilliseconds;
19.        var totalItemCount = WeatherReports.Count + RoadTrafficReports.Count;
20.
21.        Report($"Database update took {elapsedTime}ms for {totalItemCount}
22.        items.");
23.
24.        // Renews lists
25.        Report("Clearing Lists");
26.        WeatherReports.Clear();
27.        RoadTrafficReports.Clear();

```

***Τμήμα Κώδικα 12. Κλείδωμα νήματος εκτέλεσης για
εγγραφή και καθάρισμα λιστών.***

Μόλις ολοκληρωθεί το ξεκλείδωμα της λίστας, οι οντότητες που παραμένουν ενεργές προστίθενται στην – καθαρή – πλέον λίστα και ο κύκλος συνεχίζει.

Ο συγχρονισμός με τη βάση δεδομένων έχει οριστεί κάθε ένα λεπτό. Ωστόσο, μπορεί να παραμετροποιηθεί για συχνότερους ή αραιότερους συγχρονισμούς.

Αποτέλεσμα των παραπάνω ήταν πως, στα δεδομένα που παράχθηκαν μέσω του επιπέδου προσομοίωσης, τα οποία ξεπερνούσαν το 1 εκατομμύριο εγγραφές, χρειάστηκαν περίπου 3 ώρες προκειμένου να συγχρονιστούν.

Για μεγαλύτερες ανάγκες και απαιτήσεις, το γεγονός πως η υλοποίηση είναι βασισμένη στο Docker δίνει τη δυνατότητα για χρήση περισσότερων από ένα instance της εφαρμογής, σε πολλαπλές υπολογιστικές μονάδες, ώστε να περατώνεται ο συγχρονισμός ταχύτερα.

4.2.3 K-Means Algorithm

Στα πλαίσια της διπλωματικής εργασίας, χρησιμοποιήθηκε μία βιβλιοθήκη, η οποία μπορεί να εφαρμόσει Μηχανική Μάθηση (Machine Learning) για την αποτύπωση αλγορίθμων. Αυτή η βιβλιοθήκη ονομάζεται Accord και περιέχει ευρεία γκάμα χρησιμότητας. [84] Στην προκειμένη περίπτωση, χρησιμοποιήθηκε για Clustering Μονάδων ο αλγόριθμος K-Means, ο οποίος αναλύθηκε προηγουμένως.

Τα γεωχωρικά δεδομένα, που έχουν αντληθεί από τα δεδομένα των συσκευών IoT, αποτυπώνονται σε μορφή double[] (double array), τα οποία τροφοδοτούνται επαναληπτικά στη βιβλιοθήκη. Με τη σειρά της, η βιβλιοθήκη εκτελεί τον αλγόριθμο και αυτό έχει ως αποτέλεσμα τη δημιουργία των clusters και των κεντροειδών (centroids) τους.

Στο παρακάτω χωρίο αποτυπώνεται η χρήση της βιβλιοθήκης. Η βιβλιοθήκη απαιτεί τα δεδομένα να είναι στη σωστή μορφή, αλλά και τον αριθμό των clusters που είναι επιθυμητό να δημιουργηθούν. Το τελικό αποτέλεσμα της εκτέλεσης είναι ένας πίνακας γεωχωρικών συντεταγμένων, το πλήθος του οποίου καθορίστηκε από τον αριθμό των κεντροειδών που ορίστηκε στη βιβλιοθήκη, όπου κάθε εγγραφή του είναι το κέντρο ενός cluster.

```
1. foreach (var group in groups)
2. {
3.     // Group by quarter
4.     var quarters = group.GroupBy(g => new
5.     {
6.         Quarter = g.Quarter,
7.         Year = g.StartTime.GetValueOrDefault().Year
8.     }).ToList();
9.
10.    foreach (var yearlyQuarter in quarters)
11.    {
12.        // Select Coordinates for K-Means algorithm
13.        var coordinates = yearlyQuarter.Select(rtr => new double[]
14.        {rtr.Latitude.GetValueOrDefault(),
15.        rtr.Longitude.GetValueOrDefault()}).ToArray();
16.
17.        var kmeans = new KMeans(clustersNumber);
18.        var clusterCollection = kmeans.Learn(coordinates);
19.
20.        // Create DTO
21.        var quarterPeriod = new QuarterlyPeriod(_unitOfWork)
22.        {
```



```

21.         DateFrom = lowestTime,
22.         DateTo = largestTime,
23.         State = group.Key,
24.         Name = $"{yearlyQuarter.Key.Year} Q{yearlyQuarter.Key.Quarter}",
25.         Quarter = yearlyQuarter.Key.Quarter,
26.         Year = yearlyQuarter.Key.Year
27.     };
28.     clusterCollection.ToList().ForEach(cc =>
29.     {
30.         quarterPeriod.ClusterCentroids.Add(new ClusterCentroid(_unitOfWork)
31.         {
32.             Latitude = cc.Centroid[0],
33.             Longitude = cc.Centroid[1]
34.         });
35.     });
36.

```

Τμήμα Κώδικα 13. Η χρήση της βιβλιοθήκης Accord για αποτύπωση του αλγορίθμου K-Means.

4.2.4 Απεικόνιση και χρήση δεδομένων

Τα δεδομένα, που εισάγονται συνεχόμενα στη βάση δεδομένων Microsoft SQL Server, είναι πολυπληθή και μεγάλης κλίμακας. Αυτό επιτρέπει την επίτευξη του στόχου της παρούσας εργασίας, όσον αφορά το κομμάτι εισαγωγής των δεδομένων, αλλά χρειάζεται να επιλυθεί και το ζήτημα της ορθής αξιοποίησής τους.

Αυτό επετεύχθη με τη χρήση ενός λογισμικού Business Intelligence (BI). Όπως αναφέρθηκε στο Κεφάλαιο 3, επιλέχθηκε το Apache Superset.

4.2.4.1 Εγκατάσταση

Χάρη στο Docker, η εγκατάσταση του Apache Superset, η οποία απαιτεί πάρα πολλά components, γίνεται πολύ εύκολα. Αυτό διευκολύνεται ακόμα περισσότερο μέσω του αρχείου docker-config.yml, που περιέχει και τα προαπαιτούμενα components αλλά και όλες τις ρυθμίσεις που είναι απαραίτητες για την ομαλή λειτουργία.

Το μόνο που χρειάζεται από το χρήστη είναι η δημιουργία ενός λογαριασμού με administrative δικαιώματα, που είναι ουσιαστικά ο πρώτος χρήστης. Αυτό επιτυγχάνεται μέσω εκτέλεσης της παρακάτω γραμμής κώδικα στην κονσόλα του λογισμικού συστήματος.

```
1. docker-compose exec superset superset-init
```

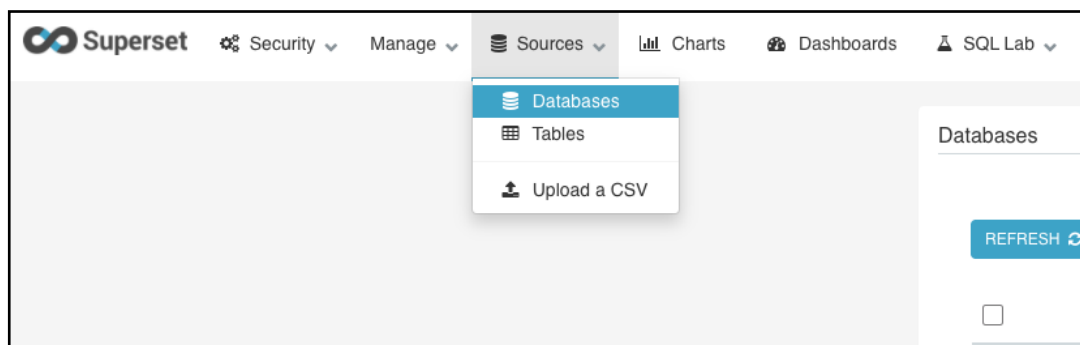
Τμήμα Κώδικα 14. Δημιουργία αρχικού χρήστη σε Apache Superset.

Στη συνέχεια, ζητείται από το χρήστη η εισαγωγή username και κωδικού για εισαγωγή στο σύστημα.

4.2.4.2 Προσθήκη Βάσης Δεδομένων

Για να μπορέσει να αξιοποιήσει τα δεδομένα ο χρήστης, πρέπει να προστεθεί η βάση δεδομένων που περιέχει τις πληροφορίες σαν πηγή δεδομένων στο Superset. (**Εικόνα 39**)

Αυτό γίνεται μέσω της επιλογής του μενού, Sources => Databases => + (Προσθήκη)



Εικόνα 39. Προσθήκη βάσης δεδομένων στο Superset.

Στο πεδίο SQL Alchemy URI απαιτείται το connection string της βάσης δεδομένων SQL.

```
1. mssql+pymssql://sa:XXXXXXXXXX@thesis_sqlsrv:1433/FiwareHub
```

Τμήμα Κώδικα 15. Μορφή Connection String για βάση δεδομένων στο Superset.

4.2.4.3 Ανάλυση – Απεικόνιση δεδομένων



Για να μπορέσει να γίνει ανάλυση των δεδομένων, θα προστεθούν και οι πίνακες που θέλουμε να αναλύσουμε μέσα στο Superset.

Αυτό επιτυγχάνεται μέσω της επιλογής στο μενού Sources => Tables => + (Προσθήκη).

Το μόνο που απαιτείται είναι η επιλογή της βάσης δεδομένων – εφόσον το Superset δίνει τη δυνατότητα επιλογής δεδομένων από πολλαπλές βάσεις δεδομένων – και η εισαγωγή του ονόματος του πίνακα στον SQL Server. (**Εικόνα 40**)

Import a table definition

Database *	FiwareHub
Schema	Schema Schema, as used only in some databases like Postgres, Redshift and DB2
Table Name *	WeatherReports Name of the table that exists in the source database

SAVE  

Εικόνα 40. Προσθήκη πίνακα για ανάλυση στο Superset.

Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να χρησιμοποιήσει πληθώρα μεθόδων απεικόνισης (visualizations), ιστορικής αναζήτησης και πολλών άλλων, με την προσθήκη Charts.

Η προσθήκη Chart επιτυγχάνεται μέσω της επιλογής του μενού Charts => +.

4.3 Σημαντική σημείωση

Η αρχιτεκτονική που χρησιμοποιήθηκε, όπως περιγράφηκε παραπάνω, περιέχει λύσεις που εξυπηρετούν συγκεκριμένα την εκπόνηση της παρούσας εργασίας.

Το επίπεδο προσομοίωσης μπορεί πολύ εύκολα να αντικατασταθεί με πραγματικές συσκευές Διαδικτύου των Πραγμάτων που βασίζονται στο πρωτόκολλο FIWARE JSON και FIWARE Ultralight αντίστοιχα. Η υλοποίηση της εφαρμογής δε στηρίζεται σε αυτό, παρά μόνο λειτουργεί συμπληρωματικά για την πληρέστερη απεικόνιση της Περίπτωσης Χρήσης (Use Case).

Το κομμάτι του συγχρονισμού είναι πλήρως λειτουργικό, σε περίπτωση αντικατάστασης με πραγματικές FIWARE συσκευές. Το μόνο που θα χρειαστεί είναι η δημιουργία νέων μοντέλων, τα οποία περιγράφουν τις μετρήσεις που παρέχει μία συσκευή, εάν αυτές διαφοροποιούνται από τα ήδη υπάρχοντα μοντέλα.

Τέλος, δίδεται στον ενδιαφερόμενο η δυνατότητα επιλογής για την αξιοποίηση των δεδομένων του. Κάλλιστα μπορεί να επιλέξει κάποιο άλλο Business Intelligence πρόγραμμα, εάν δεν καλύπτει τις ανάγκες του το Apache Superset ή να χρησιμοποιήσει τα δεδομένα σε αναλύσεις μέσω γλωσσών Statistical Computing (όπως για παράδειγμα η R).

5

Περίπτωση Χρήσης (Use Case)

Στο Κεφάλαιο 5, θα παρουσιαστεί η Περίπτωση Χρήσης (Use Case).

Με τον όρο «Περίπτωση Χρήσης» γίνεται αναφορά σε ένα σύνολο σεναρίων που συνδέονται με ένα συγκεκριμένο σκοπό του χρήστη. Χρησιμεύει στο να καθοριστούν και να αναλυθούν αποδοτικά οι λειτουργικές απαιτήσεις του συστήματος, παρέχοντας μία σαφή περιγραφή για τις ενέργειες στις οποίες πρέπει να προβεί το σύστημα. Τέλος, παρέχει την κατάλληλη βάση για την πραγματοποίηση ελέγχων, προκειμένου να επαληθεύεται η ορθή λειτουργία του συστήματος.

Στην παρούσα μελέτη, η Περίπτωση Χρήσης αφορά την ανάλυση δεδομένων τροχαίων ατυχημάτων, σε συνδυασμό με τα καιρικά φαινόμενα που λάμβαναν χώρα τη δεδομένη χρονική στιγμή στο σημείο που διαδραματίστηκε το ατύχημα.

5.1 Δεδομένα Dataset

Όπως προαναφέρθηκε στο Κεφάλαιο 4, για να μπορέσει να υλοποιηθεί το σενάριο της Περίπτωσης Χρήσης, έπρεπε να χρησιμοποιηθεί μεγάλος αριθμός συσκευών Διαδικτύου των Πραγμάτων.

Αυτό επιτεύχθηκε με τη χρήση του Επιπέδου Προσομοίωσης. Χρησιμοποιήθηκαν πραγματικά δεδομένα από 2 Datasets, τα οποία τροφοδοτήθηκαν σε εικονικές συσκευές. Το πρώτο Dataset αφορούσε δεδομένα τροχαίων ατυχημάτων στις Ηνωμένες Πολιτείες της Αμερικής (Η.Π.Α.) και το δεύτερο αφορούσε καιρικά φαινόμενα στις Η.Π.Α.

5.1.1 Dataset Τροχαίων Ατυχημάτων

Το πρώτο Dataset¹² που χρησιμοποιήθηκε αφορά δεδομένα ατυχημάτων στις Η.Π.Α. για το χρονικό διάστημα 2016 – 2019.

¹² <https://www.kaggle.com/sobhanmoosavi/us-accidents>

Στο σενάριο, τα συγκεκριμένα δεδομένα υποκαθιστούν το ρόλο μίας έξυπνης συσκευής που θα ήταν τοποθετημένη σε ένα αυτοκίνητο. Αυτή η συσκευή αποθηκεύει δεδομένα τοποθεσίας (γεωγραφικό πλάτος και μήκος) ανά τακτά χρονικά διαστήματα. Αυτά τα γεωγραφικά δεδομένα μπορούν πολύ εύκολα να μεταφραστούν σε μία διεύθυνση, σε αναγνώσιμη, από άνθρωπο, μορφή.

Σε περίπτωση ατυχήματος, δηλαδή όταν η συσκευή ανιχνεύσει μεγάλο κραδασμό στο σώμα του αυτοκινήτου, θεωρεί πως έγινε ατύχημα. Ανάλογα με τη σοβαρότητα του ατυχήματος, η οποία – μάλιστα – καταγράφεται, μπορεί να ειδοποιήσει και τις αρμόδιες αρχές για βοήθεια ή αναφορά του συμβάντος.

Τέτοιου είδους έξυπνες συσκευές έχουν κάνει ήδη την εμφάνισή τους στα νέα αυτοκίνητα και αρχίζουν να γίνονται ολοένα και πιο κοινά, ακόμα και σε αυτοκίνητα χαμηλότερης αξίας.

5.1.2 Dataset Καιρικών Συνθηκών

Το δεύτερο Dataset¹³ που χρησιμοποιήθηκε αφορά δεδομένα καιρικών φαινομένων και συνθηκών στις Η.Π.Α. για το χρονικό διάστημα 2016 – 2019.

Αυτά τα δεδομένα ενέχουν το ρόλο έξυπνων συσκευών αναγνώρισης καιρικών φαινομένων, οι οποίες τοποθετούνται διάσπαρτες σε διάφορα γεωγραφικά σημεία. Λόγω του χαμηλού κόστους τους, μπορεί να χρησιμοποιηθεί πολύ μεγάλος αριθμός συσκευών. Οι συσκευές έχουν τη δυνατότητα να αναγνωρίζουν βασικά καιρικά δεδομένα, όπως το καιρικό φαινόμενο (ήλιος, βροχή, χιόνι), τη δριμύτητά του, την κατεύθυνση του ανέμου και άλλα. Χάρη στη δυνατότητα της επικοινωνίας τους με εξειδικευμένες συσκευές καιρικών φαινομένων, έχουν πρόσβαση σε ακόμα πιο σύνθετη πληροφορία, όπως η ταχύτητα του ανέμου και οι μετρήσεις ύψους βροχής. Φυσικά γνωρίζοντας την τοποθεσία όπου ανευρίσκεται η κάθε συσκευή, είναι γνωστές – ανά πάσα στιγμή – οι καιρικές συνθήκες που επικρατούν σε κάθε γεωγραφικό σημείο όπου υπάρχει μία τέτοια συσκευή.

Παρόλο που οι προαναφερθείσες συσκευές αφορούν, κυρίως, εξειδικευμένες περιπτώσεις χρήσης, έχουν αρχίσει να γίνονται προσιτές για όσους παρατηρητές καιρού ενδιαφέρονται να αναγνωρίζουν, να αποθηκεύουν και να αναλύουν καιρικά φαινόμενα.

5.1.3 Συνδυασμός των δύο Dataset

Έχοντας, ουσιαστικά, δύο τμήματα πληροφοριών που μας ενδιαφέρουν, από τη μία γεωγραφικά δεδομένα ενός ατυχήματος, και από την άλλη καιρικές συνθήκες, χρειάζεται να

¹³ <https://www.kaggle.com/sobhanmoosavi/us-weather-events>

πραγματοποιηθεί συνδυασμός των δύο αυτών τμημάτων πληροφοριών, προκειμένου να δημιουργηθεί μία εμπλουτισμένη πληροφορία.

Σε αυτό το σημείο, έρχεται να βοηθήσει η ύπαρξη της σχεσιακής βάσης δεδομένων (SQL Server), μέσω της χρήσης της εντολής JOIN και της χρήση ενός εικονικού πίνακα (View).

```
1. CREATE VIEW FullReport AS
2. SELECT RTR.StartTime, WR.ReportTime, *
3. FROM RoadTrafficReport RTR
4.     INNER JOIN WeatherReport WR
5.         ON RTR.City = WR.City
6.         AND RTR.County = WR.County
7.         AND RTR.State = WR.State
8.         AND RTR.ZipCode = WR.ZipCode
9.         AND RTR.Country = WR.Country
10.        AND (WR.ReportTime BETWEEN
11.                (SELECT DATEADD(HOUR, -2, RTR.StartTime))
12.                AND
13.                (SELECT DATEADD(HOUR, 2, RTR.StartTime))
14.        )
15. ORDER BY RTR.StartTime
```

Τμήμα Κώδικα 16. Δημιουργία Εικονικού Πίνακα εμπλουτισμένης πληροφορίας.

Μέσω του **Τμήματος Κώδικα 16**, παράγεται ένας εικονικός πίνακας με όνομα FullReport, ο οποίος αντιπροσωπεύει μία πλήρη πληροφορία για την Περίπτωση Χρήσης.

Οι δύο πίνακες, ο ένας με τα καιρικά φαινόμενα και ο άλλος με τα τροχαία ατυχήματα, συνδέθηκαν μέσω INNER JOIN, όχι μέσω ενός Ξένου κλειδιού (Foreign Key), αλλά βάσει συνθηκών. Κάθε οντότητα από τη βάση των τροχαίων ατυχημάτων φέρει πληροφορίες για τη γεωγραφική τοποθεσία (Πόλη, Περιφέρεια, Πολιτεία, Ώρα ατυχήματος). Αντίστοιχη πληροφορία φέρουν και οι οντότητες καιρικών συνθηκών, οπότε, με βάση αυτά τα στοιχεία, πραγματοποιείται η ένωση των δύο.

Είναι σημαντικό να αναφερθεί πως είναι αδύνατον να υπάρξει μέτρηση καιρικών συνθηκών την ακριβή χρονική στιγμή που έγινε ένα ατύχημα, άρα στο πεδίο της Ώρας δόθηκε ένα εύρος ± 2 ωρών και επιλέγεται η μέτρηση που είναι πλησιέστερη στην ώρα του ατυχήματος.

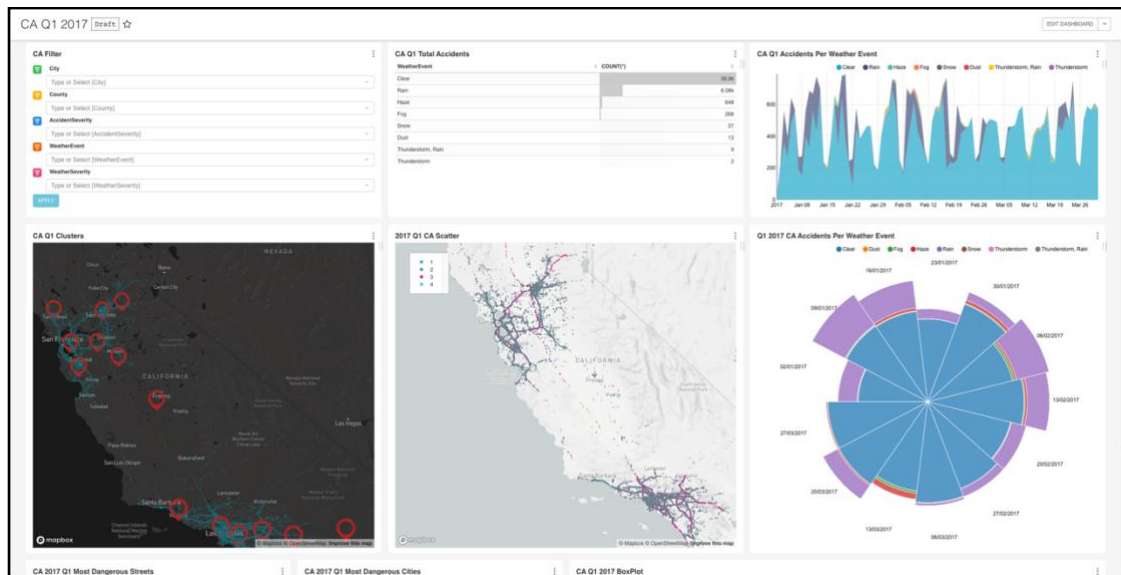
5.2 Αξιοποίηση Δεδομένων

Έχοντας, πλέον, μία εμπλουτισμένη πληροφορία σε μία μορφή που προσφέρεται για ανάλυση, δημιουργήθηκαν Dashboards (Πίνακες Οργάνων) στο λογισμικό Business Intelligence (BI) Apache Superset.

Λόγω μεγάλου όγκου δεδομένων, στα πλαίσια της παρούσας Περίπτωσης Χρήσης, επιλέχθηκαν να αναλυθούν η Καλιφόρνια και το Τέξας, δηλαδή οι δύο πολιτείες με το μεγαλύτερο αριθμό ατυχημάτων για τα έτη 2017 και 2018.

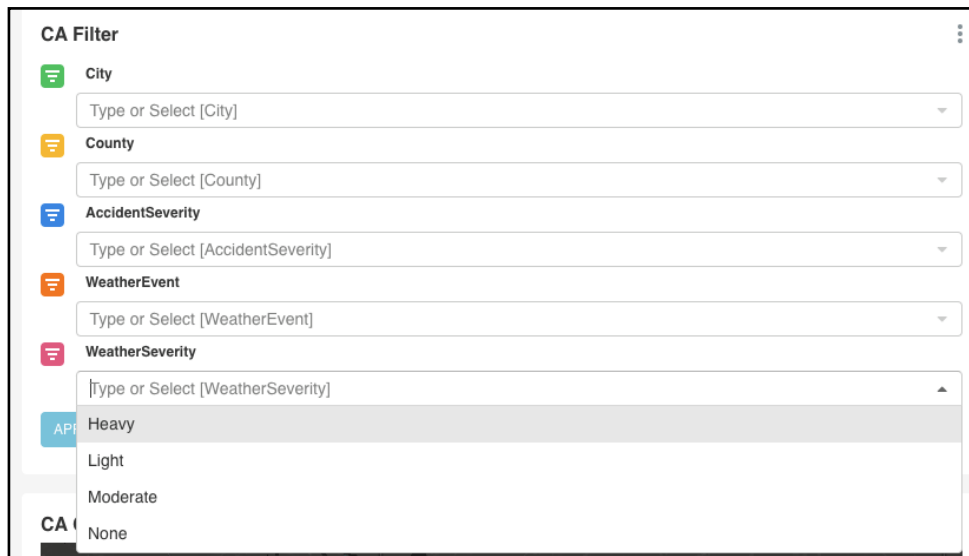
5.2.1 Πίνακας οργάνων Τριμήνου

Κάθε τριμηνιαίο Dashboard απαρτίζεται από ένα σύνολο πινάκων, διαγραμμάτων και χαρτών που αποτυπώνουν την πληροφορία με κατανοητό τρόπο.



Εικόνα 41. Αποτύπωση Πίνακα Οργάνων για την Καλιφόρνια, το πρώτο τρίμηνο του 2017.

Όλες οι πληροφορίες, που αφορούν αυτό το τρίμηνο, μπορούν να φιλτραριστούν περαιτέρω με τη χρήση φίλτρων που βρίσκεται στο πάνω αριστερό μέρος του πίνακα οργάνων (**Εικόνα 42**)



Εικόνα 42. Φίλτρο για περαιτέρω εκλέπτυνση των δεδομένων.

5.2.1.1 Συνολικά ατυχήματα ανά καιρικό φαινόμενο

Στον πίνακα της **Εικόνας 43**, παρουσιάζονται το καιρικό φαινόμενο (αριστερά) και ο αριθμός των τροχαίων ατυχημάτων που αντιστοιχούν σε αυτό (δεξιά).

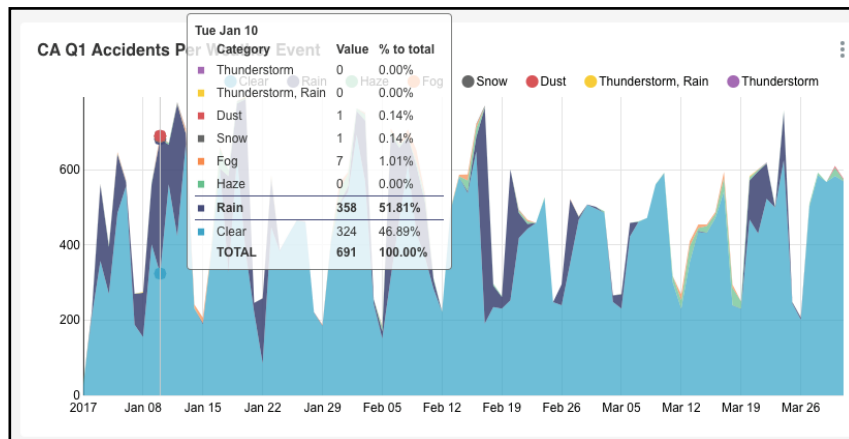
CA Q1 Total Accidents	
WeatherEvent	COUNT(*)
Clear	35.9k
Rain	6.08k
Haze	648
Fog	268
Snow	37
Dust	13
Thunderstorm, Rain	9
Thunderstorm	2

Εικόνα 43. Ατυχήματα της πολιτείας της Καλιφόρνια, το 2017.

5.2.1.2 Ημερήσια ανάλυση ατυχημάτων

Μέσω του διαγράμματος Area, μπορεί να αποτυπωθεί, σε ημερήσια βάση, ο αριθμός των ατυχημάτων ανά καιρικό φαινόμενο.

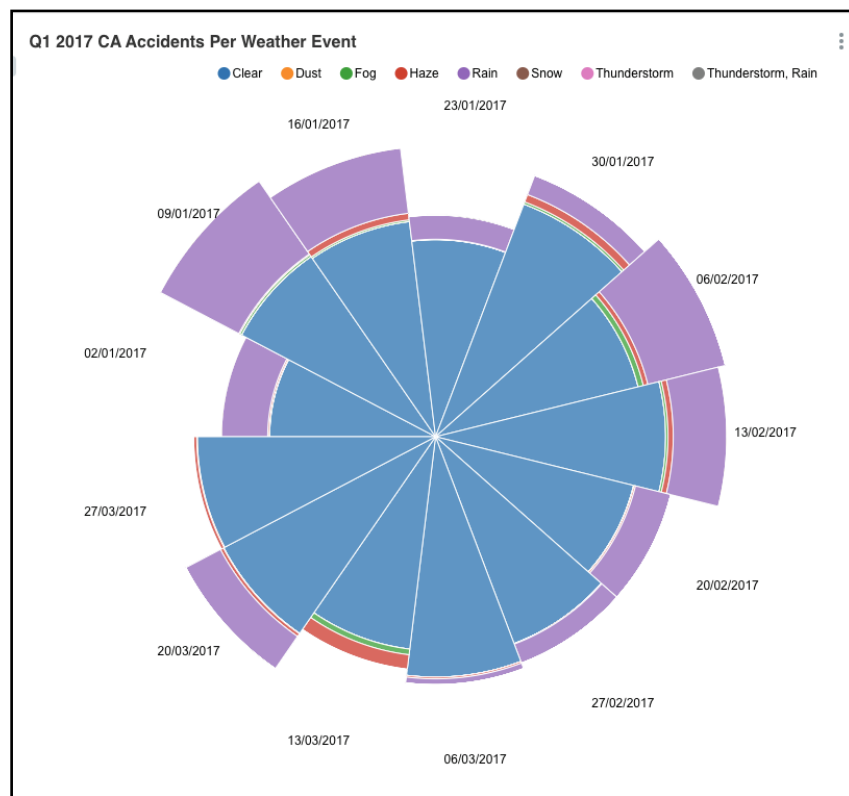
Κάνοντας mouse over με το ποντίκι, παρουσιάζεται ένα αναδυόμενο παράθυρο που φέρει πληροφορίες ατυχημάτων για τη συγκεκριμένη ημέρα, όπως τον αριθμό και το ποσοστό τους.



Εικόνα 44. Διάγραμμα Area για αποτύπωση ημερησίων ατυχημάτων.

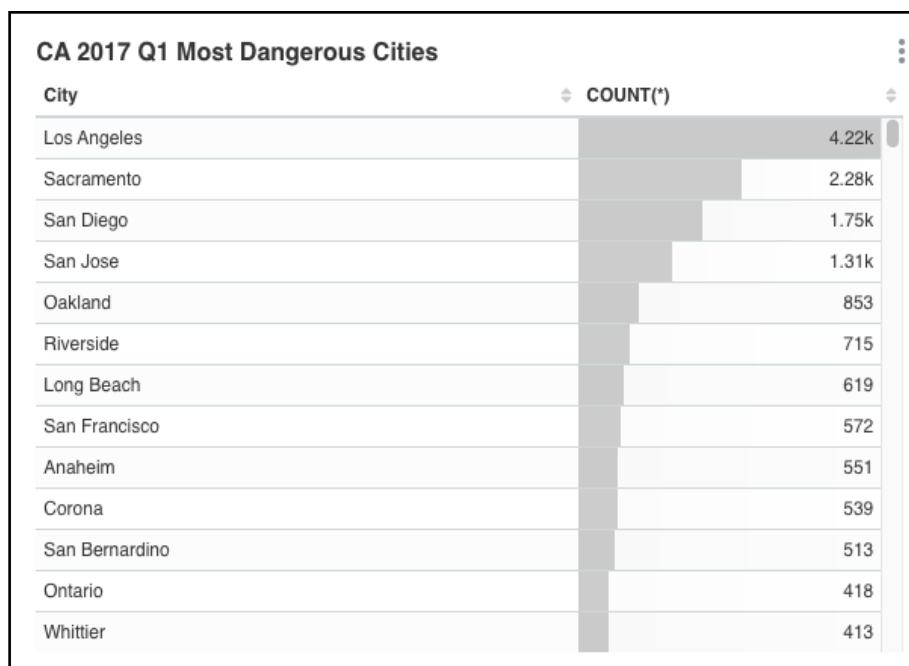
5.2.1.3 Εβδομαδιαία ανάλυση ατυχημάτων

Μέσω του διαγράμματος Nightingale Rose, η περίοδος του τριμήνου αναλύεται σε εβδομαδιαία βάση. Πατώντας πάνω σε μία εβδομαδιαία αναπαράσταση, το διάγραμμα μεταμορφώνεται σε ένα pie chart για την ανάλυση της επιλεγμένης περιόδου.



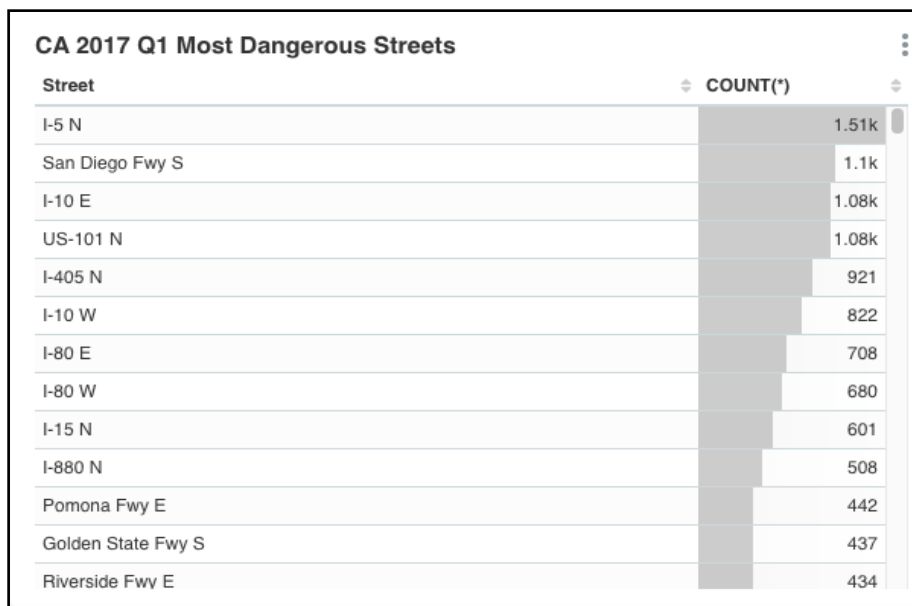
Εικόνα 45. Διάγραμμα Nightingale Rose για οπτικοποίηση εβδομαδιαίων ατυχημάτων.

5.2.1.4 Πίνακας πόλεων



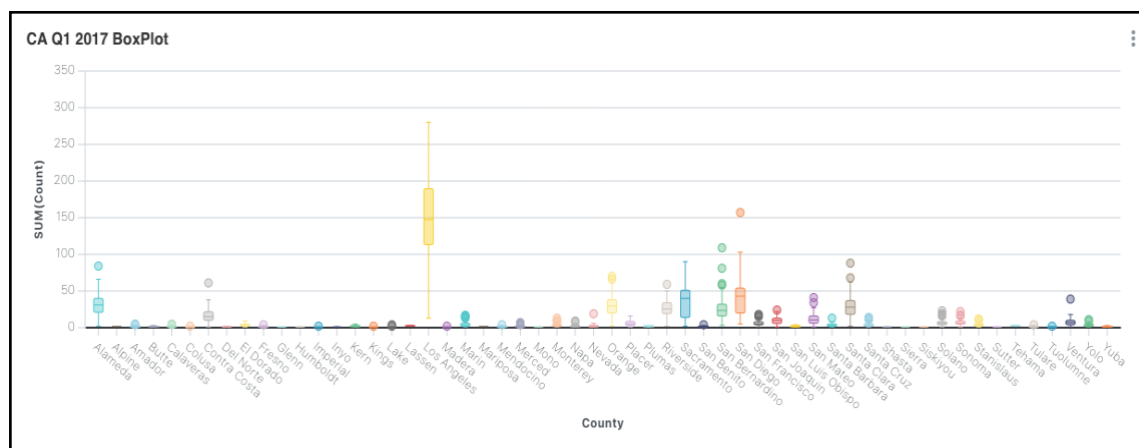
Εικόνα 46. Πόλεις με το μεγαλύτερο αριθμό ατυχημάτων.

5.2.1.5 Πίνακας δρόμων με τα περισσότερα ατυχήματα



Εικόνα 47. Δρόμοι με τα περισσότερα ατυχήματα.

5.2.1.6 Boxplot



Εικόνα 48. Διάγραμμα Boxplot για αποτύπωση ατυχημάτων.

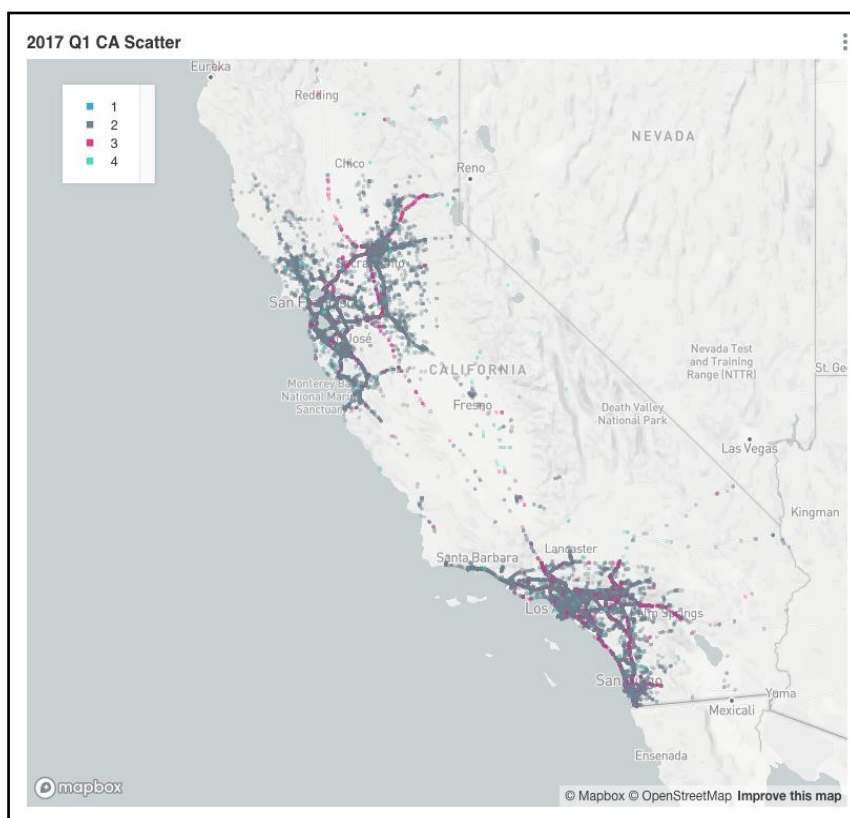
Στο διάγραμμα της **Εικόνας 48**, παρουσιάζεται το διάγραμμα Boxplot για τα ατυχήματα ανά περιφέρεια της πολιτείας.

Κάνοντας mouse-over με το ποντίκι, εμφανίζονται λεπτομέρειες για τη μικρότερη παρατήρηση (Min), μεγαλύτερη παρατήρηση (Max), διάμεσο (Median), πρώτο τεταρτημόριο (1st Quartile), και τρίτο τεταρτημόριο (3rd Quartile). (**Εικόνα 49**)

Los Angeles	
Min	13
Max	280
Median	147.5
1st Quartile	113.25
3rd Quartile	189.5

Εικόνα 49. Αριθμητικά δεδομένα boxplot.

5.2.1.7 Χάρτης συνολικών ατυχημάτων



Εικόνα 50. Χάρτης ατυχημάτων.

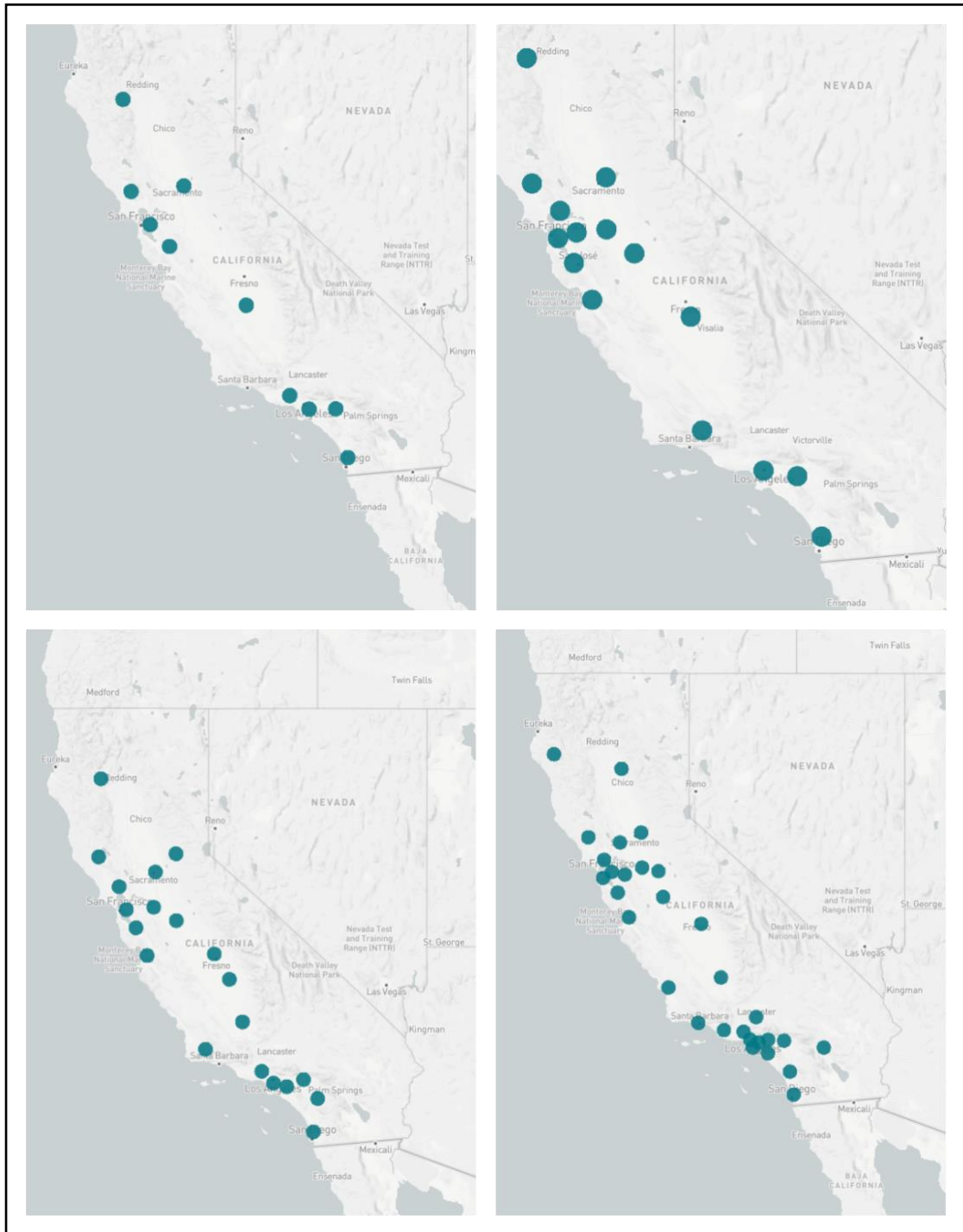
Στο χάρτη της **Εικόνας 50**, αποτυπώνονται, γεωγραφικά, όλα τα ατυχήματα του τριμήνου. Χρησιμοποιείται διαφορετική χρωματική απόχρωση για την αποτύπωση της σοβαρότητας του ατυχήματος, θεωρώντας ως νούμερο 1 το λιγότερο σοβαρό και ως 4 το περισσότερο σοβαρό.

5.2.1.8 Χάρτης με προτεινόμενες τοποθεσίες ασθενοφόρων

Στα πλαίσια της αξιοποίησης των πληροφοριών που έχουν συλλεχθεί, δημιουργήθηκε ένας χάρτης με τα προτεινόμενα σημεία όπου θα μπορούσαν να τοποθετηθούν ασθενοφόρα για την αντιμετώπιση των τροχαίων ατυχημάτων.

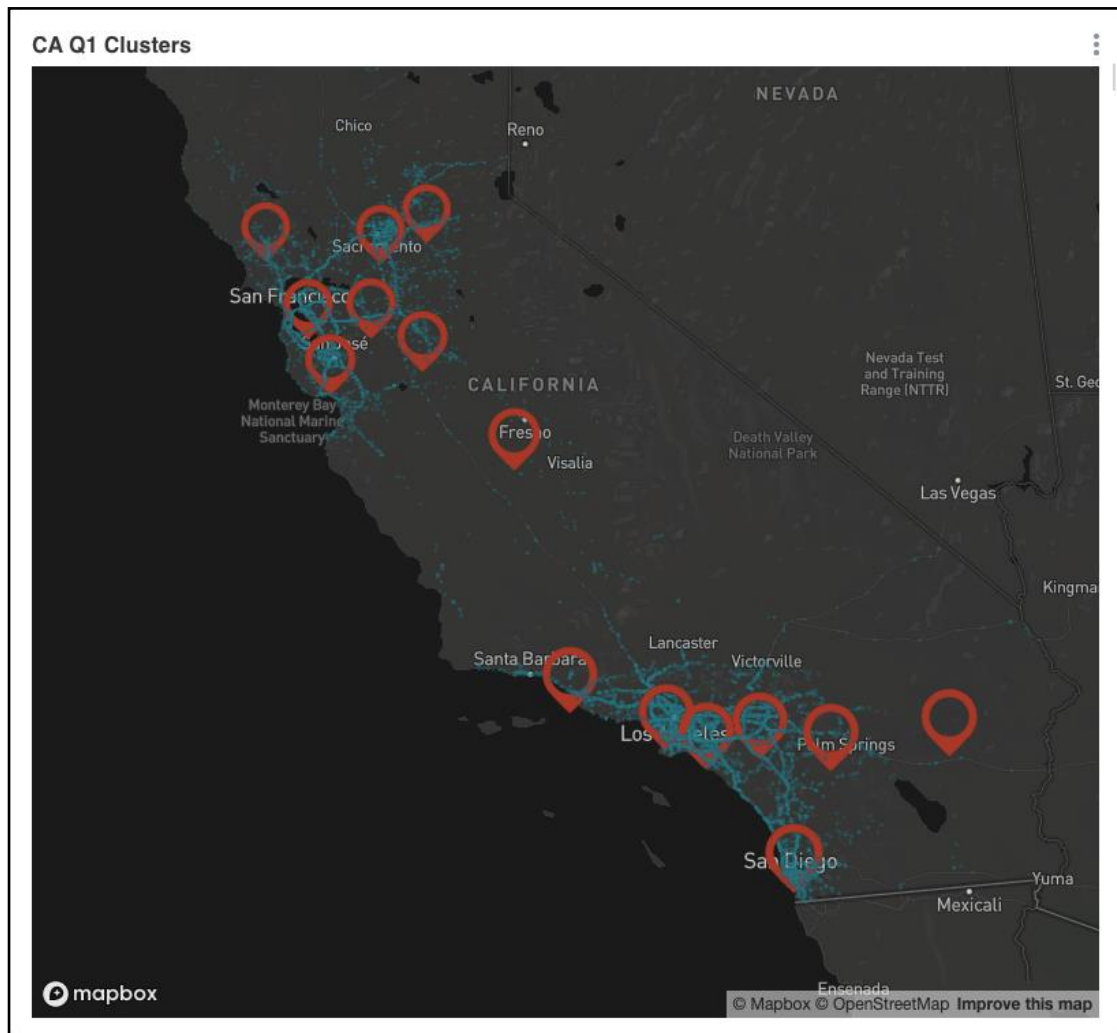
Τα γεωγραφικά δεδομένα (μήκος και πλάτος) των ατυχημάτων τροφοδοτήθηκαν στον αλγόριθμο K-Means και παράχθηκαν, μέσω αυτού, centroids (κεντροειδή) με τοποθεσίες όπου θα εξυπηρετούσε η παρουσία ασθενοφόρων για την έγκαιρη αντιμετώπιση των τροχαίων ατυχημάτων.

Δοκιμάστηκαν σενάρια με 10, 15, 20 και 30 centroids, τα οποία παρουσιάζονται στην **Εικόνα 51**.



Εικόνα 51. Από πάνω αριστερά προς κάτω δεξιά, 10, 15, 20, 30 centroids αντίστοιχα.

Επιλέχθηκε να εμφανίζονται 15 centroids για να περιοριστούν, όσο το δυνατόν περισσότερο, οι επικαλύψεις τοποθεσιών. Τα παραπάνω δεδομένα εμφανίζονται σε πιο κατανοητή μορφή μέσω νέου χάρτη, που αναπτύχθηκε εκτός του προγράμματος BI και ενσωματώθηκε με τη χρήση Iframe. Με κόκκινο χρώμα, εντοπίζονται οι προτεινόμενες τοποθεσίες των ασθενοφόρων. (**Εικόνα 52**)



Εικόνα 52. Ενδεικτική τοποθέτηση ασθενοφόρων.

5.2.2 Πίνακας οργάνων Ιστορικών Δεδομένων

Για την πληρέστερη οπτικοποίηση της πληροφορίας, αλλά και για να δοθεί η δυνατότητα ιστορικής αναζήτησης και στατιστικών, δημιουργήθηκε ένας πίνακας οργάνων (Dashboard) που αφορά Ιστορικά Δεδομένα. Σε αυτόν τον πίνακα, συγκεντρώνονται συνολικά δεδομένα ατυχημάτων, αλλά και δυνατότητες αναζήτησης δεδομένων.

5.2.2.1 Αναζήτηση Ιστορικών Δεδομένων

Δημιουργήθηκε σελίδα, η οποία εντάχθηκε στη λύση του Superset μέσω Iframe, που δίνει τη δυνατότητα στο χρήστη να αναζητήσει, μέσω φίλτρων, στατιστικά δεδομένα ατυχημάτων, κατηγοριοποιημένα ανά δρόμο.

Ο χρήστης μπορεί να αναζητήσει βάσει Πολιτείας, πόλης, βαρύτητας ατυχήματος, τριμήνου, καιρικού φαινομένου και έντασης του καιρικού φαινομένου.

Historical Data Percentage

State

Cities

Streets

Accident Severity

Quarters

Weather Event

Weather Severity

Match Weather Conditions in Total Count

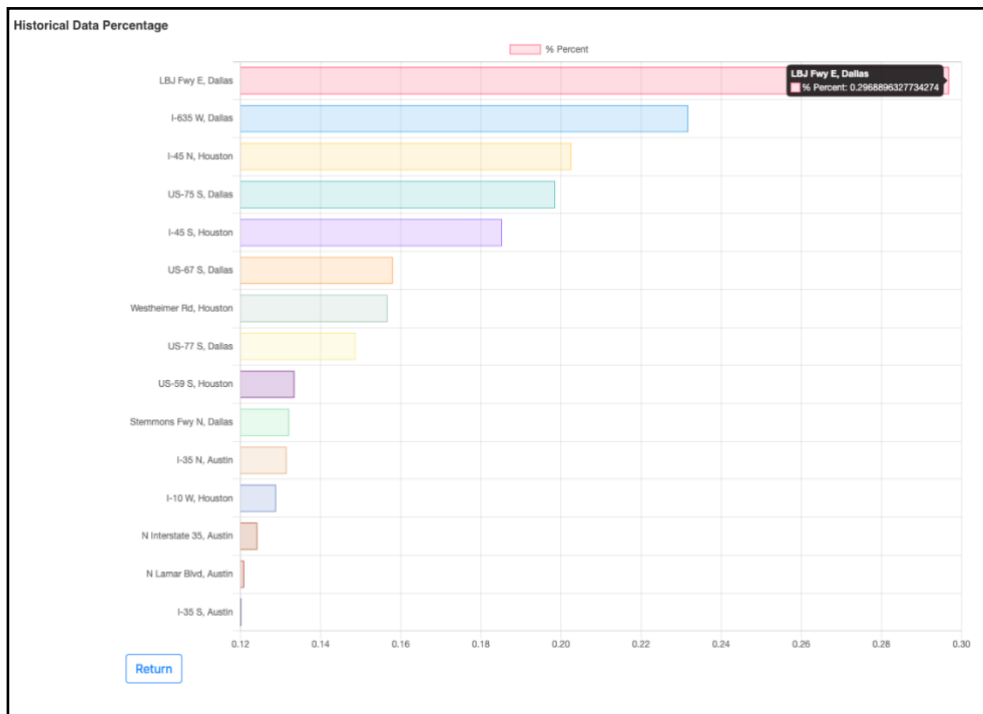
***Εικόνα 53.** Αναζήτηση ιστορικών δεδομένων.*

Με τη χρήση των dropdown menu για επιλογή κριτηρίων, σχηματίζεται ένα ερώτημα (query), το οποίο έχει ως αποτέλεσμα τα ποσοστά ατυχημάτων ανά δρόμο.

Συγκεκριμένα στην **Εικόνα 53**, σχηματίζεται η ερώτηση: «Ποιοι δρόμοι στο Τέξας, έχουν σοβαρότητα ατυχήματος Υψηλό (μεγαλύτερο του 3), κατά τη διάρκεια του δεύτερου τριμήνου, με καθαρό καιρό;».

Η τελευταία επιλογή, με όνομα «Match Weather Conditions in Total Count», αφορά τον αριθμό του **παρονομαστή**: εάν δηλαδή επιθυμεί ο χρήστης ο αριθμός των ατυχημάτων που καταμετρήθηκαν στον αριθμητή, να διαιρεθεί με το σύνολο των ατυχημάτων της περιόδου που έχουν **τα ίδια καιρικά κριτήρια ή όχι**.

Τα αποτελέσματα αυτής της ερώτησης παρουσιάζονται, υπό μορφή bar charts, και αντιπροσωπεύουν το ποσοστό ατυχημάτων που σημειώθηκαν στο συγκεκριμένο δρόμο με τα κριτήρια που αναζήτησε ο χρήστης. (**Εικόνα 54**)



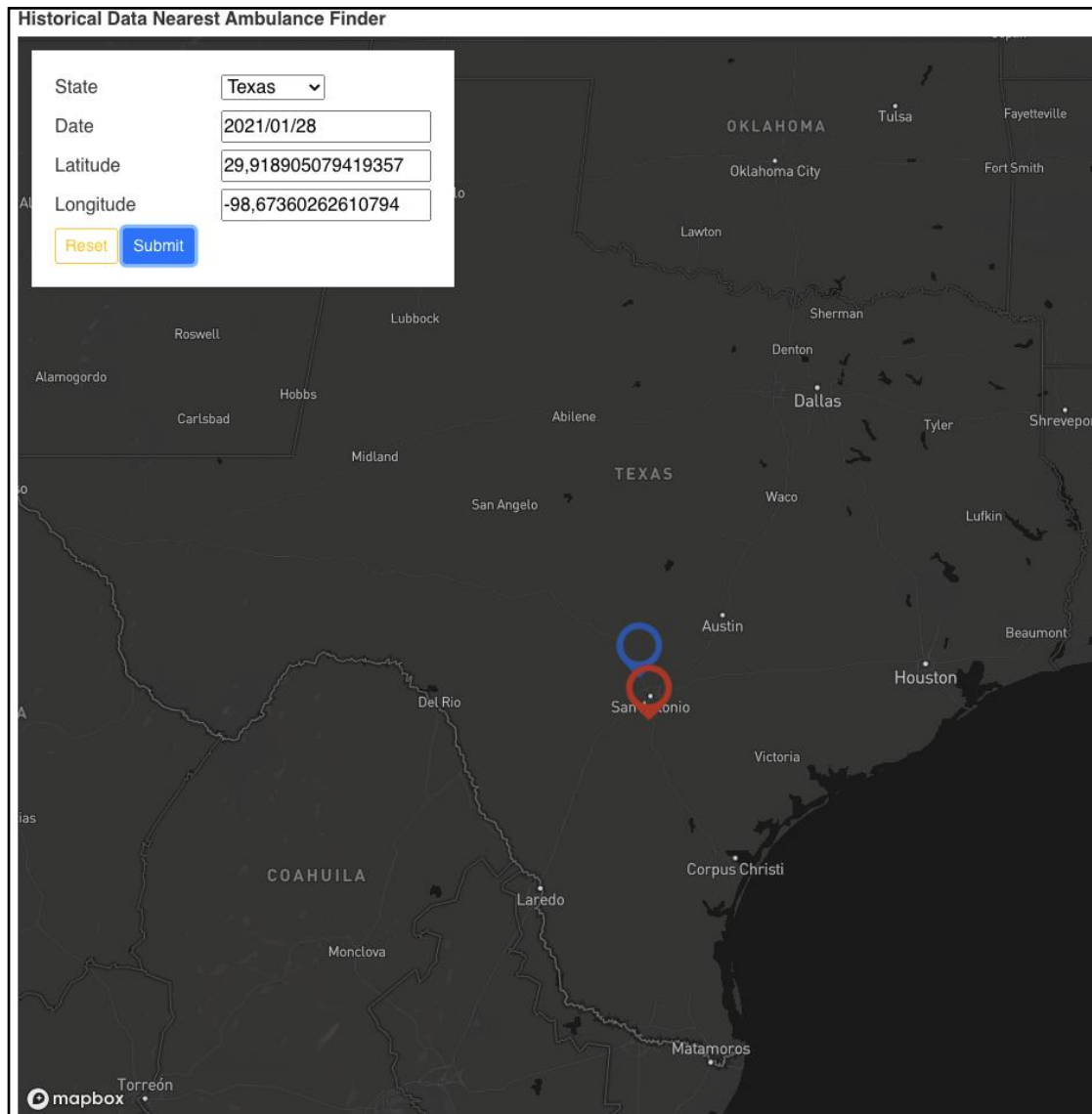
Εικόνα 54. Αποτελέσματα ιστορικής αναζήτησης.

5.2.2.2 Αναζήτηση πλησιέστερου ασθενοφόρου

Με βάση τη μελέτη που έγινε στην τριμηνιαία ανάλυση των πολιτειών για τη δημιουργία προτεινόμενων θέσεων για τοποθέτηση ασθενοφόρων, δίνεται η δυνατότητα στο χρήστη να αναζητήσει την πλησιέστερη θέση ασθενοφόρου με ένα σημείο ατυχήματος.

Για να φέρει αποτελέσματα, χρειάζεται να γίνει επιλογή του σημείου ατυχήματος στο χάρτη ή χειρωνακτική εισαγωγή του γεωγραφικού πλάτους και μήκους στον ειδικά διαμορφωμένο χώρο της φόρμας και επιλογή της ημερομηνίας.

Στην **Εικόνα 55**, παρουσιάζεται, με μπλε χρώμα, το επιλεγμένο σημείο ατυχήματος καθώς και η ημερομηνία που έχει εισαχθεί. Το κόκκινο χρώμα αντιπροσωπεύει την τοποθεσία του πλησιέστερου ασθενοφόρου.



Εικόνα 55. Αναζήτηση ασθενοφόρου με τοποθεσία ατυχήματος.

5.3 Προεπισκόπηση Λειτουργιών (Video)

Για την καλύτερη οπτικοποίηση της πλατφόρμας για την ανάλυση των δεδομένων, καθώς και επειδή τα πλαίσια συγγραφής ενός κειμένου δεν επιτρέπουν τη σφαιρική απεικόνισή της, παρατίθεται ένα βίντεο που αναδεικνύει τη λειτουργικότητα του Business Intelligence Software.¹⁴

¹⁴ <https://youtu.be/GEg0QWeQNxI>

5.4 Σύγκριση – Συμπεράσματα

Σε αυτήν την ενότητα, θα πραγματοποιηθεί σύγκριση των δύο datasets της παρούσας έρευνας. Στην **Εικόνα 56**, απεικονίζονται σχηματικά ο αριθμός των ατυχημάτων ανά καιρικό φαινόμενο σε εβδομαδιαία βάση, για το πρώτο τρίμηνο του 2017 και του 2018 στην πολιτεία της Καλιφόρνια.



Εικόνα 56. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017, για την πολιτεία της Καλιφόρνια, σε εβδομαδιαία βάση.

Στον **Πίνακα 4**, παρουσιάζονται, για το πρώτο τρίμηνο του 2017 και το αντίστοιχο του 2018, η καιρική συνθήκη, ο απόλυτος αριθμός των ατυχημάτων και το ποσοστό των ατυχημάτων ανάλογα με την καιρική συνθήκη, ως προς το σύνολο των ατυχημάτων, στην πολιτεία της Καλιφόρνια.

Καλιφόρνια				
Καιρικό φαινόμενο	1ο τριμηνο 2017		1ο τριμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
Καθαρός καιρός	35,938	83.59	36,232	86.34
Βροχή	6,079	14.14	4,116	9.81
Λοιπές δυσμενείς καιρικές συνθήκες	977	2.27	1,614	3.85
Σύνολο	42,994	100.00	41,962	100.00

Πίνακας 4. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και του 2018, για την πολιτεία της Καλιφόρνια.

Τα περισσότερα τροχαία ατυχήματα συμβαίνουν με καλές καιρικές συνθήκες σε ποσοστά 83,59% για το 2017 και 86,34% για το 2018. Τα αντίστοιχα ποσοστά με βροχή είναι 14,14% και 9,81% και με λοιπές δυσμενείς καιρικές συνθήκες 2,68% και 3,68%.

Στον **Πίνακα 5**, αναφέρονται οι πέντε πόλεις με τα περισσότερα ατυχήματα στην πολιτεία της Καλιφόρνια. Παρατηρείται μία σοβαρή μείωση κατά 5 ποσοστιαίες μονάδες το 2018, η οποία οφείλεται στη σημαντική μείωση των ατυχημάτων στις πόλεις Σακραμέντο, Σαν Ντιέγκο και Σαν Χοσέ. Πιθανώς να ορίστηκαν νέα ανώτατα όρια εντός αυτών των πόλεων ή να βελτιώθηκαν σημαντικά οι οδικές υποδομές (καλύτερη σηματοδότηση, βελτιώσεις διασταυρώσεων και άλλα). Τα ατυχήματα σε αυτές τις πόλεις αντιπροσωπεύουν το 24,22% για το 2017 και 19,05% για το 2018 του συνόλου των ατυχημάτων της πολιτείας.

5 πόλεις της Καλιφόρνιας με τα περισσότερα ατυχήματα				
Πόλεις	1ο τρίμηνο 2017		1ο τρίμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
Λος Άντζελες	4,220	9.82	3,819	9.10
Σακραμέντο	2,280	5.30	1,190	2.84
Σαν Ντιέγκο	1,750	4.07	1,220	2.91
Σαν Χοσέ	1,310	3.05	969	2.31
Όκλαντ	853	1.98	798	1.90
Σύνολο των 5 πολεων	10,413	24.22	7,996	19.05
Γενικό σύνολο της πολιτείας	42,994		41,965	

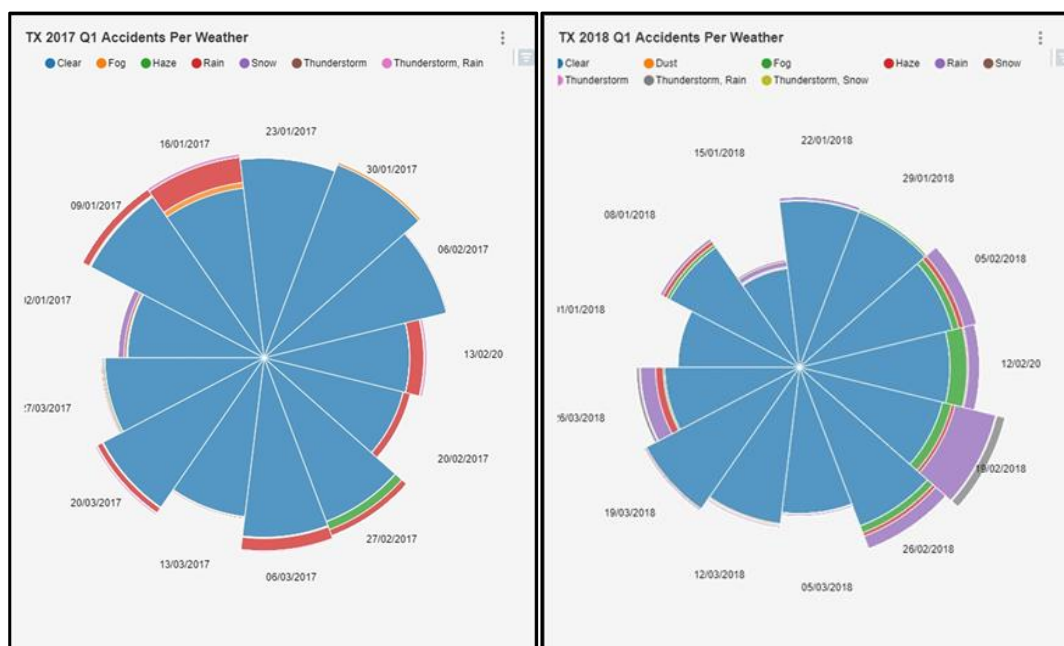
Πίνακας 5. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις της πολιτείας της Καλιφόρνια.

Στον **Πίνακα 6**, αναφέρονται οι πέντε αυτοκινητόδρομοι με τα περισσότερα ατυχήματα. Η επικινδυνότητα των αυτοκινητοδρόμων παρέμεινε σχεδόν ίδια. Ενδεχομένως να υπήρξαν μικρές βελτιώσεις στον αυτοκινητόδρομο I-10 E, καθώς το 2018 εμφανίζεται μία μείωση κατά 242 ατυχήματα, που αντιστοιχεί σε ποσοστό της τάξης του 22,42%, σε σχέση με το 2017. Τα ατυχήματα σε αυτούς τους αυτοκινητοδρόμους αντιπροσωπεύουν το 13,24% για το 2017 και 12,17% για το 2018 του συνόλου των ατυχημάτων της πολιτείας.

5 αυτοκινητόδρομοι της Καλιφόρνιας με τα περισσότερα ατυχήματα				
Αυτοκινητόδρομοι	1ο τρίμηνο 2017		1ο τρίμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
1-5 N	1,510	3.51	1,320	3.15
San Diego Fwy S	1,100	2.56	1,000	2.38
I-10 E	1,080	2.51	838	2.00
US - 101 N	1,080	2.51	973	2.32
I - 405 N	921	2.14	977	2.33
Σύνολο 5 αυτοκινητοδρόμων	5,691	13.24	5,108	12.17
Γενικό σύνολο της πολιτείας	42,994		41,965	

Πίνακας 6. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους της πολιτείας της Καλιφόρνια.

Στην **Εικόνα 57**, απεικονίζονται σχηματικά ο αριθμός των ατυχημάτων ανά καιρικό φαινόμενο σε εβδομαδιαία βάση, για το 2017 και 2018 αντίστοιχα στην πολιτεία του Τέξας.



Εικόνα 57. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και 2018 για την πολιτεία του Τέξας, σε εβδομαδιαία βάση.

Στον **Πίνακα 7**, παρουσιάζονται, για το πρώτο τρίμηνο του 2017 και το αντίστοιχο του 2018, η καιρική συνθήκη, ο απόλυτος αριθμός των ατυχημάτων και το ποσοστό των ατυχημάτων ανάλογα με την καιρική συνθήκη, ως προς το σύνολο των ατυχημάτων, στην πολιτεία του Τέξας. Από τα στοιχεία του **Πίνακα 7**, προκύπτει ότι τα περισσότερα ατυχήματα συμβαίνουν με καλές καιρικές συνθήκες, σε ποσοστά 94,46% για το 2017 και 90,15% για το 2018.

Καιρικό φαινόμενο	Τέξας			
	1ο τρίμηνο 2017		1ο τρίμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
Καθαρός καιρός	22,621	94.46	19,138	90.15
Βροχή	870	3.63	1,114	5.25
Λοιπές δυσμενείς συνθήκες	457	1.91	977	4.60
	23,948	100.00	21,229	100.00

Πίνακας 7. Αριθμός τροχαίων ατυχημάτων, ανά καιρικό φαινόμενο, το πρώτο τρίμηνο του 2017 και του 2018, για την πολιτεία του Τέξας.

Στον **Πίνακα 8**, αναφέρονται οι πέντε περισσότερο επιρρεπείς σε τροχαία ατυχήματα πόλεις της πολιτείας του Τέξας. Τα ατυχήματα στις πέντε αυτές πόλεις αντιπροσωπεύουν το 86,42% το 2017 και το 77,36% το 2018 ως προς το σύνολο των ατυχημάτων της πολιτείας.

Παρατηρείται μία μείωση κατά 5 ποσοστιαίες μονάδες το 2018, η οποία οφείλεται στη σημαντική μείωση των ατυχημάτων στις πόλεις Χιούστον και Ντάλας. Πιθανώς να ορίστηκαν νέα ανώτατα όρια εντός αυτών των πόλεων.

5 Πόλεις του Τέξας με τα περισσότερα ατυχήματα				
Πόλεις	1ο τρίμηνο 2017		1ο τρίμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
Χιούστον	9,170	38.29	7,120	33.54
Νταλλας	5,320	22.21	3,450	16.25
Ωστιν	4,490	18.75	3,790	17.85
Σαν Αντώνιο	1,140	4.76	1,690	7.96
Φορθ Γουορθ	575	2.40	373	1.76
Σύνολο των 5 πόλεων	20,695	86.42	16,423	77.36
Γενικό σύνολο της πολιτείας	23,948		21,229	

Πίνακας 8. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις της πολιτείας του Τέξας.

Στον **Πίνακα 9**, αναγράφονται οι πέντε αυτοκινητόδρομοι με τα περισσότερα ατυχήματα στην πολιτεία του Τέξας. Η επικινδυνότητα των αυτοκινητοδρόμων αυτών παρέμεινε σχεδόν ίδια. Το ποσοστό των ατυχημάτων στους πέντε αυτοκινητοδρόμους του Τέξας είναι της τάξεως του 6-7% του συνόλου των ατυχημάτων της πολιτείας.

5 αυτοκινητόδρομοι του Τέξας με τα περισσότερα ατυχήματα				
Αυτοκινητόδρομοι	1ο τρίμηνο 2017		1ο τρίμηνο 2018	
	Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
1-45 N	457	1.91	372	1.75
1-45 S	351	1.47	321	1.51
LBJ Fwy E	300	1.25	287	1.35
1-35 N	233	0.97	268	1.26
US -59 S	223	0.93	182	0.86
Σύνολο των 5 αυτοκινητοδρόμων	1,564	6.53	1,430	6.74
Γενικό σύνολο της πολιτείας	23,948		21,229	

Πίνακας 9. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους της πολιτείας του Τέξας.

Στον **Πίνακα 10**, παρουσιάζονται, για το πρώτο τρίμηνο του 2017 και το αντίστοιχο του 2018, ο αριθμός των τροχαίων ατυχημάτων στις δύο πολιτείες ανά καιρικό φαινόμενο. Από τη συγκριτική ανάλυση, φαίνεται ότι η πολιτεία της Καλιφόρνια επηρεάζεται περισσότερο από τις καιρικές συνθήκες, σε σχέση με την πολιτεία του Τέξας. Ο συνολικός αριθμός των τροχαίων ατυχημάτων είναι σχεδόν διπλάσιος στην Καλιφόρνια, σε σύγκριση με το Τέξας. Πιο

συγκεκριμένα, για το 2017, ο λόγος των ατυχημάτων της Καλιφόρνια προς τα ατυχήματα του Τέξας είναι 1,80 και ο αντίστοιχος λόγος για το 2018 είναι 1,98. Με βάση τον πληθυσμό των δύο πολιτειών, προκύπτει ότι ο αριθμός των ατυχημάτων ανά 100.000 κατοίκους, για την πολιτεία της Καλιφόρνια, είναι 107,65 το 2017 και 105,08 το 2018. Τα αντίστοιχα στοιχεία, για την πολιτεία του Τέξας, είναι 72,38 και 80,55 ατυχήματα ανά 100.000 κατοίκους.

	Καιρικό φαινόμενο	Καλιφόρνια		Τέξας	
		Αριθμός ατυχημάτων	Ποσοστό	Αριθμός ατυχημάτων	Ποσοστό
1ο τρίμηνο 2017	Καθαρός καιρός	35,938	83.59	22,621	94.46
	Βροχή	6079	14.14	870	3.63
	Λοιπές δυσμενείς συνθήκες	977	2.27	457	1.91
	Σύνολο της πολιτείας	42,994	100.00	23,948	100.00
1ο τρίμηνο 2018	Καθαρός καιρός	36,232	86.34	19,138	90.15
	Βροχή	4,116	9.81	1,114	5.25
	Λοιπές δυσμενείς συνθήκες	1,614	3.85	977	4.60
	Σύνολο της πολιτείας	41,962	100.00	21,229	100.00

Πίνακας 10. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πολιτείες της Καλιφόρνια και του Τέξας.

Στον *Πίνακα 11*, αναφέρονται στοιχεία ατυχημάτων για τις πέντε μεγαλύτερες πόλεις κάθε πολιτείας, στα οποία εμφανίζονται τεράστιες διαφορές. Για τις πόλεις της Καλιφόρνια, το ποσοστό των ατυχημάτων στις πέντε αυτές πόλεις είναι 24,22% για το 2017 και 19,05% για το 2018, ενώ τα αντίστοιχα ποσοστά για τις πόλεις του Τέξας είναι 86,42% και 77,36%.

Ατυχήματα 1ου τριμήνου του 2017 και του 2018 των 5 μεγαλύτερων πόλεων από κάθε									
Πόλεις	Καλιφόρνια				Τέξας				
	2017		2018		Πόλεις	2017		2018	
	Ατυχήματα	Ποσοστό	Ατυχήματα	Ποσοστό		Ατυχήματα	Ποσοστό	Ατυχήματα	Ποσοστό
Λος Άντζελες	4,220	9.82	3,819	9.10	Χιούστον	9,170	38.29	7,120	33.54
Σακραμέντο	2,280	5.30	1,190	2.84	Νταλλας	5,320	22.21	3,450	16.25
Σαν Ντιέγκο	1,750	4.07	1,220	2.91	Ωστιν	4,490	18.75	3,790	17.85
Σαν Χοσέ	1,310	3.05	969	2.31	Σαν Αντώνιο	1,140	4.76	1,690	7.96
Όκλαντ	853	1.98	798	1.90	Φορθ Γουορθ	575	2.40	373	1.76
Σύνολο των 5 πόλεων	10,413	24.22	7,996	19.05		20,695	86.42	16,423	77.36
Γενικό σύνολο της πολιτείας	42,994		41,965			23,948		21,229	

Πίνακας 11. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στις πέντε επικίνδυνες πόλεις στις πολιτείες της Καλιφόρνια και του Τέξας.

Στον **Πίνακα 12**, παρουσιάζονται οι πέντε αυτοκινητόδρομοι με τα περισσότερα ατυχήματα από κάθε πολιτεία. Για τους αυτοκινητόδρομους της Καλιφόρνια, το ποσοστό των ατυχημάτων στις πέντε αυτούς αυτοκινητόδρομους είναι 13,24% για το 2017 και 12,17% για το 2018, ενώ τα αντίστοιχα ποσοστά για τις πόλεις του Τέξας είναι 6,53% και 6,74%.

Σύνολο 5 αυτοκινητοδρόμων									
Καλιφόρνια					Τέξας				
	2017		2018			2017		2018	
Αυτοκινητόδρομοι	Ατυχήματα	Ποσοστό	Ατυχήματα	Ποσοστό	Αυτοκινητόδρομοι	Ατυχήματα	Ποσοστό	Ατυχήματα	Ποσοστό
1-5 N	1,510	3.51	1,320	3.15	1-45 N	457	1.91	372	1.75
San Diego Fwy S	1,100	2.56	1,000	2.38	1-45 S	351	1.47	321	1.51
I-10 E	1,080	2.51	838	2.00	LBJ Fwy E	300	1.25	287	1.35
US - 101 N	1,080	2.51	973	2.32	1-35 N	233	0.97	268	1.26
I - 405 N	921	2.14	977	2.33	US -59 S	223	0.93	182	0.86
Σύνολο 5 αυτοκινητοδρόμων	5,691	13.24	5,108	12.17	Σύνολο 5 αυτοκινητοδρόμων	1,564	6.53	1,430	6.74
Γενικό σύνολο της πολιτείας	42,994		41,965		Γενικό σύνολο της πολιτείας	23,948		21,229	

Πίνακας 12. Αριθμός και ποσοστό τροχαίων ατυχημάτων, το πρώτο τρίμηνο του 2017 και του 2018, στους πέντε επικίνδυνους αυτοκινητοδρόμους στις πολιτείες της Καλιφόρνια και του Τέξας.

Συμπερασματικά, στην Καλιφόρνια, τα ατυχήματα είναι περισσότερα τόσο σε απόλυτους αριθμούς, όσο σε αναγωγή ανά 100.000 κατοίκους, σε σύγκριση με το Τέξας. Αυτή η υπεροχή της πολιτείας της Καλιφόρνια μπορεί, εν μέρει, να αποδοθεί στο γεγονός ότι η Καλιφόρνια είναι η μεγαλύτερη, σε πληθυσμό, πολιτεία των Ηνωμένων Πολιτειών της Αμερικής. Συγκεκριμένα, η Καλιφόρνια έρχεται τρίτη σε έκταση (423.967 τ.χλμ.) και πρώτη σε πληθυσμό (39.937.500). [85] Αντίστοιχα, η πολιτεία του Τέξας έχει επιφάνεια 695.662 τ.χλμ., είναι η δεύτερη μεγαλύτερη – σε έκταση – πολιτεία των Η.Π.Α. μετά την Αλάσκα, και δεύτερη – σε πληθυσμό – μετά την Καλιφόρνια, με 29.730.300 κατοίκους. [86]

Ωστόσο, τα ατυχήματα στην Καλιφόρνια είναι διεσπαρμένα σε όλη τη πολιτεία, δεδομένου ότι ο αριθμός τους στις πέντε μεγαλύτερες πόλεις και τους πέντε περισσότερο επιρρεπείς σε ατυχήματα αυτοκινητοδρόμους αντιπροσωπεύουν το 37,46% για το 2017 και το 31,22% για το 2018 του συνόλου των ατυχημάτων της πολιτείας. Αντίθετα, στο Τέξας, παρατηρείται ο όγκος των ατυχημάτων να εμφανίζεται στις πέντε πόλεις και τους πέντε αυτοκινητόδρομους σε ποσοστά 92,95% για το 2017 και 84,1% για το 2018 του συνόλου των ατυχημάτων της πολιτείας. Τέλος, το ποσοστό των ατυχημάτων στο Λος Άντζελες, που αποτελεί τη μεγαλύτερη σε πληθυσμό πόλη της Καλιφόρνιας με 3,967 εκατομμύρια κατοίκους (σύμφωνα με στοιχεία

του 2019), είναι 9,82%, ενώ το αντίστοιχο ποσοστό για την ίδια χρονική στο Χιούστον, που αποτελεί τη μεγαλύτερη σε πληθυσμό πόλη του Τέξας με 2,31 εκατομμύρια κατοίκους (σύμφωνα με στοιχεία του 2019), είναι 38,29%.

5.5 Οφέλη – Παρατηρήσεις

Χάρη στην κατανοητή απεικόνιση των δεδομένων, που δεν περιορίζεται από τη συνεχή ένταξη μεταγενέστερων, δίνεται η δυνατότητα, ανά πάσα στιγμή, να είναι γνωστή με σαφήνεια η κατάσταση που επικρατεί στα τροχαία ατυχήματα, καθώς και η συσχέτισή τους με τα καιρικά φαινόμενα.

Στην προκειμένη περίπτωση, στις νότιες πολιτείες της Αμερικής, όπως αναλύθηκαν στα πλαίσια της Περίπτωσης Χρήσης, δεν παρατηρούνται ακραία καιρικά φαινόμενα τακτικά. Εντοπίστηκε μία αύξηση ατυχημάτων σε περιόδους έντονων βροχοπτώσεων, στα πλαίσια του φυσιολογικού.

Ωστόσο, η παραπάνω ανάλυση παρέχει πληροφορίες που είναι αξιοποιήσιμες σε πολλαπλά επίπεδα. Εντοπίζεται η ανάγκη για βελτίωση του οδικού δικτύου σε δρόμους που συσσωρεύουν μεγάλο αριθμό ατυχημάτων. Περιφέρειες πολιτειών που εμφανίζουν σημαντικό αριθμό τροχαίων ατυχημάτων, ενδεχομένως, να χρήζουν μεγαλύτερης χρηματοδότησης για επίβλεψη και διόρθωση τυχόν ελλείψεων. Αναδεικνύονται περίοδοι ή καιρικά φαινόμενα, κατά τη διάρκεια των οποίων μπορεί να χρειαστεί να μειωθεί το όριο ταχύτητας κυκλοφορίας στους δρόμους, προς αποφυγή τροχαίων ατυχημάτων. Τέλος, μέσω της μελέτης που πραγματοποιήθηκε αναφορικά με τις προτεινόμενες θέσεις των ασθενοφόρων, μπορεί να επιτευχθεί μείωση των θανάσιμων ατυχημάτων, χάρη στην έγκαιρη αντιμετώπισή τους.

Γενικά, οι εξαγόμενες πληροφορίες μπορούν να αναλυθούν από αρμόδιους ειδικούς και συγκοινωνιολόγους και να οδηγήσουν σε αποφάσεις που χρησιμεύουν στο γενικότερο κοινωνικό σύνολο.

Αυτός ήταν και ο σκοπός αυτής της διπλωματικής, δηλαδή να δοθεί **αξία** σε δεδομένα ευρείας κλίμακας, που, υπό άλλες συνθήκες, θα παρέμεναν ανεκμετάλλευτα.

6

Επίλογος –

Συμπεράσματα

Οι ηλεκτρονικές συσκευές αποτελούν μεγάλο μέρος της ανθρώπινης δραστηριότητας και η συνεχόμενη αυξανόμενη ενσωμάτωσή τους προμηνύεται βέβαιη. Η βελτίωση της απόδοσης αλλά και του μεγέθους ηλεκτρονικών μερών, για λόγους ευκολότερης ενσωμάτωσης, έχει προκαλέσει μία ταχεία εξάπλωση μίας γενιάς συσκευών, η οποία έχει δημιουργήσει μία νέα τάξη πραγμάτων, το Διαδίκτυο των Πραγμάτων. Οι προκλήσεις που αναδεικνύονται είναι μεγάλες, ενώ οι λύσεις για αυτές δεν είναι πάντα γνωστές. Ωστόσο, η διαχείριση των παραγόμενων δεδομένων αποτελεί τη σημαντικότερη πτυχή στις νέες ευκαιρίες που παρουσιάζονται σε όλους τους τομείς εφαρμογής.

Μέχρι σήμερα, η γνώση και η πληροφορία δημιουργούνται και αναλώνονται στα πλαίσια κλειστών στεγανών. Ο κάτοχος της πληροφορίας ήταν συχνά και ο αναλυτής της, αφήνοντας μικρό περιθώριο για έρευνα και διαδικασίες παρατήρησης. Ωστόσο, χάρη στην ιδέα και την εξάπλωση του IoT, καθώς και χάρη στη δημιουργία των οικοσυστημάτων ανοικτών προτύπων που ευαγγελίζεται, η πληροφορία, πλέον, είναι ευρέως διαθέσιμη.

Μέσω της ποικιλομορφίας και της χρησιμότητας των συσκευών, η ιδέα του Διαδικτύου των Πραγμάτων έχει γίνει ευρέως αποδεκτή, με προοπτικές εξάπλωσης από ένα περιορισμένο φάσμα σε εκτεταμένη παγκόσμια κάλυψη. Η τόσο μεγάλη κλιμάκωση, όμως, γεννά ένα νέο πρόβλημα, τη δημιουργία τεράστιας ποσότητας δεδομένων που – κατά κόρον – παραμένουν ανεκμετάλλευτα.

Ο όρος «Μεγάλα Δεδομένα» χαρακτηρίζεται από τεράστιο όγκο, ποικιλία και συνεχή συσσώρευση νέων δεδομένων, τα οποία προέρχονται από πολλές πηγές και παρέχονται σε διαφορετικές μορφές. Ενδεικτικά, αναφέρεται πως, κάθε δύο ημέρες, παράγονται δεδομένα και πληροφορίες ίσες με αυτές που είχαν παραχθεί από την αρχή της ανθρωπότητας έως το 2003.

Τα Δεδομένα Μεγάλης Κλίμακας παρουσιάζουν αυξημένη δυσκολία ως προς την καταγραφή, τη διαχείριση και την επεξεργασία τους. Χρησιμοποιούνται διάφορες τεχνικές και τεχνολογίες, με στόχο την εξαγωγή της χρήσιμης πληροφορίας μέσα από αυτόν τον τεράστιο όγκο. Ωστόσο, η αποτελεσματική αξιοποίηση αυτών των δεδομένων καθίσταται δύσκολο έργο, λόγω του αυξανόμενου αριθμού νέων πηγών δεδομένων, των απαιτήσεων για νέα δεδομένα, καθώς και

της ανάγκης για αυξημένη ταχύτητα επεξεργασίας. Η συνεχής ροή δεδομένων, που συσσωρεύονται με πρωτοφανή ρυθμό, παρουσιάζει νέες προκλήσεις και αναδεικνύει ένα πολύ σημαντικό πρόβλημα, καθώς τα δεδομένα δεν είναι χρήσιμα από μόνα τους και αποκτούν αξία μόνο όταν μπορούν να αξιοποιηθούν για λήψη αποφάσεων.

Το ένα πρόβλημα, λοιπόν, αφορά το γεγονός ότι πρόκειται για ετερογενή και, ενδεχομένως, ημιτελή δεδομένα. Ο ανθρώπινος νους έχει τη δυνατότητα να κατανοήσει κάποια πληροφορία, ανεξάρτητα από τη μορφή στην οποία ανευρίσκεται. Αντίθετα, οι ηλεκτρονικοί υπολογιστές λειτουργούν πολύ αποτελεσματικότερα εάν τα στοιχεία που αποθηκεύουν έχουν όλα ίδιο μέγεθος και ίδια δομή. Συνεπώς, αποτελεί μεγάλη πρόκληση η αποδοτική μετατροπή των μεγάλων δεδομένων, ώστε να έχουν κοινή δομή.

Μία δεύτερη πρόκληση αποτελεί η ικανότητα διαχείρισης μεγάλου και ταχέως αυξανόμενου όγκου δεδομένων. Πέρα από την αύξηση του όγκου των δεδομένων, πολλές σύγχρονες εφαρμογές είναι αναγκαίο να παράγουν αποτέλεσμα πολύ γρήγορα. Αυτό σημαίνει ότι τα αποτελέσματα πρέπει να αναλυθούν άμεσα, σχεδόν σε πραγματικό χρόνο, ώστε να υλοποιηθούν κάποιες ενέργειες ή να ληφθούν αποφάσεις.

Η τεχνολογία έχει δημιουργηθεί για να υπηρετεί τις ανάγκες του ανθρώπου και, σε αυτό το πλαίσιο, επιστρατεύεται για να δώσει λύση στο παραπάνω πρόβλημα. Οι λύσεις λογισμικού και η ανάπτυξη νέων μοντέλων αρχιτεκτονικής μπορούν να εξυπηρετήσουν στην ωφέλιμη εκμετάλλευση των δεδομένων. Μοντέλα αρχιτεκτονικής όπως η Lambda, που επιλέχθηκε στα πλαίσια της παρούσας διπλωματικής, βοηθούν σημαντικά στη μείωση της ψαλίδας μεταξύ της παραγωγής δεδομένων και της τελικής αξιοποίησής τους.

Η πληροφορία ακολουθεί ένα νέο κύκλο ζωής και αξιοποίησης. Περνά σε πρώτη φάση από ένα επίπεδο ταχείας αποθήκευσης μέχρι να ενταχθεί στο ευρετήριο (index) της βάσης δεδομένων και, από εκεί, περνά στο επίπεδο εξυπηρέτησης, επιτρέποντάς της πλέον να είναι διαθέσιμη για αναζήτηση. Ουσιαστικά, αυτή η αρχιτεκτονική επιτρέπει σε μία πληροφορία, από τη στιγμή της δημιουργίας της, να είναι σχεδόν άμεσα διαθέσιμη προς εκμετάλλευση.

Όπως είναι εύκολο να αντιληφθεί κανείς, τα δεδομένα ευρείας κλίμακας είναι κρίσιμα για την υποστήριξη της λήψης αποφάσεων και θα διαδραματίσουν καθοριστικό ρόλο στην πρόοδο της κοινωνίας. Σημαντικό βήμα προς τη σωστή χρησιμοποίησή τους αποτελεί η ορθή οργάνωση και απεικόνιση όλων αυτών των δεδομένων, η οποία θα καθοδηγήσει την ταχύτερη και σωστότερη ανάλυσή τους. Είναι σημαντικό να βελτιωθεί η ποιότητα και η μορφή των νεοσχηματισμένων δεδομένων, επιτρέποντας εξαγωγή συμπερασμάτων που καθιστούν τη λήψη αποφάσεων αρκετά ευκολότερη.

Αυτό το ρόλο έχουν τα λογισμικά Επιχειρηματικής Ευφυΐας (Business Intelligence), τα οποία προσφέρουν μία από τις αποδοτικότερες λύσεις όσον αφορά τη δυνατότητα οπτικοποίησης δεδομένων μεγάλης κλίμακας. Χάρη στη δυνατότητα απρόσκοπτης ενσωμάτωσης νέων

πληροφοριών, και μέσω χρήσης αποτελεσματικών απεικονίσεων, προσφέρονται για ταχεία ανάλυση δεδομένων. Η δυνατότητα συνδυασμού ετερογενών δεδομένων, για τη δημιουργία μίας εμπλουτισμένης πληροφορίας, βοηθά στην εξαγωγή ολοκληρωμένων συμπερασμάτων.

Ο κόσμος εξελίσσεται με ταχύτατους ρυθμούς. Για να είναι χρήσιμη μία πληροφορία, χρειάζεται να αναλυθεί άμεσα και όχι με την πάροδο του χρόνου ή – λόγω περιορισμών – για συγκεκριμένο ημερολογιακό εύρος. Βρισκόμαστε σε ένα κομβικό σημείο της ανθρώπινης εξέλιξης, όπου έχουμε τη δυνατότητα να συλλέγουμε πληροφορίες με ρυθμούς που δεν ήταν ποτέ εφικτό στην ιστορία της ανθρωπότητας. Συνεπώς, είναι χρέος μας να ανακαλύψουμε, να εφαρμόσουμε ή να εφευρίσκουμε λύσεις για να αξιοποιήσουμε τα δεδομένα που έχουμε στη διάθεσή μας.

Ο κόσμος του αύριο **οφείλει** να είναι καλύτερος γιατί χτίζεται στην πληροφορία του σήμερα.

7

Βιβλιογραφία

1. Thread Group. Thread Network Topology, 2021. Διαθέσιμο στο: <https://www.threadgroup.org/BUILT-FOR-IOT/Commercial>.
2. Πλασκασοβίτη Αθηνά. Ανάπτυξη Υποδομής MQTT για IoT Εφαρμογές. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Συστημάτων Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής. Διπλωματική Εργασία, Αθήνα, 2020.
3. Quek Tommy. The advantages and disadvantages of Internet of Things (IoT), 2017. Διαθέσιμο στο: <https://www.linkedin.com/pulse/advantages-disadvantages-internet-things-iot-tommy-quek>.
4. Λοΐζος Χρίστος. Μελέτη Υποδομών Έξυπνων Σπιτιών και Εξερεύνηση Συσχέτισης Αυτών με τα Τελευταία Πρωτόκολλα και Πρότυπα του Σημασιολογικού Ιστού. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων. Διπλωματική Εργασία, Αθήνα, 2016.
5. Ελευθερίου Ηλίας. Διαδίκτυο των Πάντων: Τεχνολογία και Εφαρμογές. Πανεπιστήμιο Θεσσαλίας, Πολυτεχνική Σχολή, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. Διπλωματική Εργασία, Βόλος, 2016.
6. DataFlair. IoT Technology & Protocols – 7 Important IoT Communication Protocols, 2021. Διαθέσιμο στο: <https://data-flair.training/blogs/iot-technology/>.
7. Singh Pradeep. IoT Bytes, Application Protocols for IoT, 2021. Διαθέσιμο στο: <https://iotbytes.wordpress.com/application-protocols-for-iot/>.
8. Banafa Ahmed. Three Major Challenges Facing IoT, 2017. Διαθέσιμο στο: <https://iot.ieee.org/newsletter/march-2017/three-major-challenges-facing-iot.html/>.
9. Wong Hua. The Top Five Challenges of IoT By Special Guest Sook Hua Wong, Industry segment manager, Keysight Technologies June 30, 2020, Edited by Ken Briodagh. Διαθέσιμο στο: <https://www.iotevolutionworld.com/iot/articles/445866-top-five-challenges-iot.htm>.

10. What Is The Internet Of Things And What Challenges Does It Pose? Διαθέσιμο στο: <https://www.chakray.com/what-is-internet-of-things-and-what-challenges-does-it-pose/>.
11. Walters Nyambi. The Iot Revolution: Challenges And Opportunities, 2016. Διαθέσιμο στο: <https://www.gbnews.ch/the-iot-revolution/>.
12. IBM Social Media. The Internet of Things, 2010. Διαθέσιμο στο: <https://sites.google.com/a/cortland.edu/the-internet-of-things/>.
13. Shea Sharon, Wigmore Ivy. Techtarget, IoT security (internet of things security), 2021. Διαθέσιμο στο: <https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security>.
14. Asaad Ahmed Gad-Elrab Ahmed. Benefits and Challenges of Internet of Things for Telecommunication Networks, 2019. Διαθέσιμο στο: <https://www.intechopen.com/books/telecommunication-networks-trends-and-developments/benefits-and-challenges-of-internet-of-things-for-telecommunication-networks>.
15. Began Kate. 5 challenges still facing the Internet of Things. Διαθέσιμο στο: <https://www.iot-now.com/2020/06/03/103228-5-challenges-still-facing-the-internet-of-things/>.
16. Pal Arpan, Hemant Kumar Rath, Samar Shailendra and Abhijan Bhattacharyya. IoT Standardization: The Road Ahead, 2018. Διαθέσιμο στο: <https://www.intechopen.com/books/internet-of-things-technology-applications-and-standardization/iot-standardization-the-road-ahead>.
17. Bedwell Patrick. The U.S. Government is Creating Security Standards for IoT Devices, 2020. Διαθέσιμο στο: <https://securityboulevard.com/2020/12/the-u-s-government-is-creating-security-standards-for-iot-devices/>.
18. Khvoynitskaya Sandra. The history and future of the internet of things, 2019. Διαθέσιμο στο: <https://www.itransition.com/blog/iot-history>.
19. Kaushikm Neeraj, Bagga Teena. Internet of Things (IOT): Implications in Society, 2020. Διαθέσιμο στο: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3563104#:~:text=Adding%20technology%20to%20things%20creates,Security%20%26%20safety%3B%20Big%20Data.
20. Χατζής Κωνσταντίνος. Αξιολόγηση εργαλείων διαχείρισης Δεδομένων Μεγάλης Κλίμακας για αξιοποίησή τους από το Δημόσιο Τομέα, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων. Διπλωματική Εργασία, Αθήνα, 2018.
21. IBM Big Data & Analytics Hub. The Four V's of Big Data, 2014. Διαθέσιμο στο: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>.

22. Rouse Margaret. Big data, 2019. Διαθέσιμο στο: <https://searchdatamanagement.techtarget.com/definition/big-data>.
23. Προδρομίτη Γερασμούλα. Μεγάλα δεδομένα: Η εξόρυξή τους και η συμβολή τους στην επιχειρηματική ευφυΐα, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Φυσικής. Πτυχιακή Εργασία, Αθήνα, 2017.
24. ESPC Strategic Notes. Enter the Data Economy: EU Policies for a Thriving Data Ecosystem' Issue 21, 2017. Διαθέσιμο στο: https://ec.europa.eu/epsc/publications/strategic-notes/enter-dataeconomy_en#h-2.
25. OECD. 'Data-driven Innovation: Big Data for Growth and Well-being' OECD Publishing, 2015. Διαθέσιμο στο: https://read.oecd-ilibrary.org/science-and-technology/data-driveninnovation_9789264229358-en#page1.
26. Brynjolfsson, Hitt, Kim. 'Strength in numbers: How does data-driven decision making affect firm performance?', 2011. Διαθέσιμο στο: http://ebusiness.mit.edu/research/papers/2011.12_Brynjolfsson_Hitt_Kim_Strength%20in%20Numbers_302.pdf.
27. EU Data economy study, 2017. Διαθέσιμο στο: <https://ec.europa.eu/digital-singlemarket/en/news/final-results-european-data-market-study-measuring-size-and-trends-eu-dataeconomy>.
28. Χατζηκυριάκος Γιώργος. Υλοποίηση Αρχιτεκτονικής Ανάλυσης Ροών Δεδομένων σε πραγματικό χρόνο με υποστήριξη μεθόδων Αποθήκευσης Στοιχείων και Εξόρυξης Πληροφορίας, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής. Διπλωματική Εργασία, Αθήνα, 2017.
29. Hausenblas Michael, Bijmens Nathan. Lambda Architecture. Διαθέσιμο στο: <http://lambdaarchitecture.net/>.
30. Χαλβαντζής Νικόλαος. Σχεδιασμός και Υλοποίηση Πλατφόρμας Ελαστικής Διαχείρισης Εργαλείων Επεξεργασίας Μεγάλων Δεδομένων σε Υπολογιστικά Νέφη, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών. Διπλωματική Εργασία, Αθήνα, 2015.
31. Hazelcast. What Is Lambda Architecture? 2020. Διαθέσιμο στο: <https://hazelcast.com/glossary/lambda-architecture/>.
32. Borthakur D. "HDFS architecture guide," Hadoop Apache Proj., p. 53, 2008.
33. "Voldemort." Διαθέσιμο στο: <http://www.project-voldemort.com/voldemort/>.
34. Apache. "Storm, distributed and fault-tolerant realtime computation.", 2019. Διαθέσιμο στο: <https://storm.apache.org/>.

35. Apache. “S4: Distributed Stream Computing Platform.”, 2009. Διαθέσιμο στο: <http://incubator.apache.org/s4/>.
36. Apache. “Apache Spark™ - Lightning-Fast Cluster Computing.”, 2018. Διαθέσιμο στο: <https://spark.apache.org/>.
37. Databricks. Lambda Architecture, 2020. Διαθέσιμο στο: <https://databricks.com/glossary/lambda-architecture>.
38. Hazelcast. What Is the Kappa Architecture? 2021. Διαθέσιμο στο: <https://hazelcast.com/glossary/kappa-architecture/#:~:text=What%20is%20the%20Kappa%20Architecture,with%20a%20single%20technology%20stack>.
39. Δερβένης Κωνσταντίνος. Χρήση Big Data στον τομέα της Υγείας, Πανεπιστήμιο Πειραιώς, Τμήμα Πληροφορικής, Πρόγραμμα Μεταπτυχιακών Σπουδών «Προηγμένα Συστήματα Πληροφορικής». Μεταπτυχιακή Εργασία, Αθήνα, 2019.
40. Senthilkumar SA, Bharatendara K Rai, Amruta A Meshram, Angappa Gunasekaran, Chandrakumarmangalam S. Big Data in Healthcare Management: A Review of Literature. *American Journal of Theoretical and Applied Business* 2018; 4(2):57-69.
41. Sonnati R. Improving Healthcare Using Big Data Analytics. *International Journal of Scientific & Technology Research* 2017; 6(03):142-146.
42. Ristevski B, Chen M. Big Data Analytics in Medicine and Healthcare. *J Integr Bioinform.* 2018; 15(3):20170030.
43. Wu PY, Cheng CW, Kaddi CD, Venugopalan J, Hoffman R, Wang MD. Omic and Electronic Health Record Big Data Analytics for Precision Medicine. *IEEE Trans Biomed Eng* 2017; 64:263–73.
44. Viceconti M, Hunter P, Hose R. Big data, big knowledge: big data for personalized healthcare. *IEEE J Biomed Health Inform* 2015; 19:1209–1215.
45. Khalifa, M. Health Analytics Types, Functions and Levels: A Review of Literature. In *Data, Informatics and Technology: An Inspiration for Improved Healthcare* A. Hasman et al. (Eds.) IOS Press 2018; 137-140.
46. Valamis. What is Descriptive Analytics? 2021. Διαθέσιμο στο: <https://www.valamis.com/hub/descriptive-analytics#descriptive-analytics-definition>.
47. Auffray C, Balling R, Barroso I, Bencze L, Benson M, Bergeron J, et al. Making sense of big data in health research: towards an EU action plan, *Genome Med.* 2016; 8. Διαθέσιμο στο: <https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-016-0323-y>.

48. Θηκόγραμμα – Boxplot. Διαθέσιμο στο: <https://eclass.uoa.gr/modules/document/file.php/PPP356/%CE%98%CE%97%CE%9A%CE%9F%CE%93%CE%A1%CE%91%CE%9C%CE%9C%CE%91.pdf>.
49. Παπαδόπουλος Γ. Περιγραφική Στατιστική. Εργαστήριο Μαθηματικών & Στατιστικής. Διαθέσιμο στο: <https://www.aua.gr/gpapadopoulos/files/perigrafiki11.pdf>.
50. Galarnyk Michael. Understanding Boxplots, 2018. Διαθέσιμο στο: <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51>.
51. McLeod Saul. What does a box plot tell you? 2019. Διαθέσιμο στο: <https://www.simplypsychology.org/boxplots.html>.
52. Safe Software Inc. What is Geospatial Data? 2021. Διαθέσιμο στο: <https://www.safe.com/what-is/geospatial-data/>.
53. Safe Software Inc. What is Spatial Data? 2021. Διαθέσιμο στο: <https://www.safe.com/what-is/spatial-data/>.
54. Χωρικά Δεδομένα, 2010. Διαθέσιμο στο: https://repository.kallipos.gr/bitstream/11419/5031/1/02_chapter_2.pdf.
55. Pattnaik Anubhav. Towards Data Science, Geospatial Clustering: Kinds and Uses, 2020. Διαθέσιμο στο: <https://towardsdatascience.com/geospatial-clustering-kinds-and-uses-9aef7601f386>.
56. Ακακιάδου Γεωργία. Μελέτη του αλγορίθμου ομαδοποίησης k-means σε δεδομένα του παγκόσμιου ιστού, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Πληροφορικής. Διπλωματική εργασία, Θεσσαλονίκη, 2007.
57. Wikipedia, 2020. FIWARE. Διαθέσιμο στο: <https://es.wikipedia.org/wiki/FIWARE>.
58. FIWARE Foundation, 2021. FIWARE catalogue. Διαθέσιμο στο: <https://www.fiware.org/developers/catalogue/>.
59. Fimac, 2021. Publish/Subscribe Context Broker – Orion. Διαθέσιμο στο: <https://fimac.m-iti.org/3b.php>.
60. FIWARE Foundation, 2021. Welcome to the Fiware Iot Agent Framework. Διαθέσιμο στο: <https://iotagent-node-lib.readthedocs.io/en/latest/>.
61. FIWARE Foundation, 2020. FIWARE/tutorials.IoT-Agent-JSON. Διαθέσιμο στο: <https://github.com/FIWARE/tutorials.IoT-Agent-JSON>.
62. Wikipedia, 2020. JSON. Διαθέσιμο στο: <https://el.wikipedia.org/wiki/JSON>.

63. Καραβέλλας Σοφοκλής. Σύστημα οπτικοακουστικής επικοινωνίας και καταγραφής με το πρωτόκολλο WebRTC. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών, Εργαστήριο Υπολογιστικών Συστημάτων. Διπλωματική Εργασία, Αθήνα, 2018.
64. Καλλιαντζής Ιωάννης. Εφαρμογή συστάσεων για ταξιδιωτικές επιλογές με ενοποίηση πληροφορίας κοινωνικών δικτύων. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής. Πτυχιακή Εργασία, Θεσσαλονίκη, 2016.
65. MongoDB, Inc 2008. Data Model Design. Διαθέσιμο στο: <https://docs.mongodb.com/manual/core/data-model-design/>.
66. Londner Raphael, Cabral Alyson. An Introduction to Change Streams. Διαθέσιμο στο: <https://www.mongodb.com/blog/post/an-introduction-to-change-streams>.
67. Κολοβός Κωνσταντίνος. Μετατροπή Βάσης Δεδομένων Σε Σχήμα XML. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής. Διπλωματική Εργασία, Αθήνα, 2009.
68. SQL Joins Explained, 2020. What is a SQL join? Διαθέσιμο στο: <http://www.sql-join.com/>.
69. Vertabelo SA, 2020. How to Learn SQL JOINS. Διαθέσιμο στο: <https://learnsql.com/blog/learn-and-practice-sql-joins/>.
70. Λαμπαθάκη Φεναρέτη. Ολοκλήρωση Συστημάτων και Εφαρμογών με χρήση Web Services «Διαλειτουργικότητα Συστημάτων μεταξύ δημόσιων φορέων». Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών. Διπλωματική Εργασία, Αθήνα, 2005.
71. Τσακαλάκης Ιωάννης. Έξυπνο Σπίτι: Έλεγχος και Επιτήρηση Ζεστού Νερού Χρήσης με τον Arduino. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Ηλεκτρικής Ισχύος. Διπλωματική Εργασία, Αθήνα, 2017.
72. Wikipedia, 2020. Apache Superset. Διαθέσιμο στο: https://en.wikipedia.org/wiki/Apache_Superset.
73. Wikipedia, 2020. The Apache Software Foundation. Διαθέσιμο στο: https://en.wikipedia.org/wiki/The_Apache_Software_Foundation.
74. Beauchemin Maxime, 2016. Superset: Airbnb's data exploration platform. Διαθέσιμο στο: <https://medium.com/airbnb-engineering/caravel-airbnb-s-data-exploration-platform-15a72aa610e5#.8r1qsvxvb>.

75. Weaveworks, 2020. A Practical Guide to Choosing between Docker Containers and VMs. Διαθέσιμο στο: <https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms>.
76. NetApp, 2021. Containers and persistent storage. What are containers? Διαθέσιμο στο: <https://www.netapp.com/devops-solutions/what-are-containers/#:~:text=Containers%20are%20a%20form%20of,%2C%20libraries%2C%20and%20configuration%20files>.
77. Χουσιαδάς Δημήτριος. Τεχνολογίες Διαδικτύου για την Συλλογή και Ανάλυση Προσωπικών Δεδομένων Υγείας & Ευεξίας. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών. Διπλωματική Εργασία, Αθήνα, 2017.
78. Λουκίδης-Ανδρέου Φώτιος. Αυτόματη ενίσχυση ασφαλείας σε περιβάλλον Docker μέσω Συστήματος Υποχρεωτικού Ελέγχου (MAC). Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας, Πληροφορικής και Υπολογιστών. Διπλωματική Εργασία, Αθήνα, 2017.
79. Σωτηρόπουλος Ιωάννης Νεκτάριος. Κλιμακώσιμο Σύστημα για κατ' απαίτηση Συλλογή και Επεξεργασία Δεδομένων Δικτυακής Κίνησης. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής. Διπλωματική Εργασία, Αθήνα, 2017.
80. Simplilearn, 2020. What is Docker Compose: Benefits and Basic Commands. Διαθέσιμο στο: <https://www.simplilearn.com/tutorials/docker-tutorial/docker-compose>.
81. Wikipedia, 2021. C Sharp (programming language). Διαθέσιμο στο: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)).
82. Pineapple Team, 2013. The Callback Syndrome In Node.js. Διαθέσιμο στο: <https://blog.4psa.com/the-callback-syndrome-in-node-js/>.
83. Wikipedia, 2021. API. Διαθέσιμο στο: <https://en.wikipedia.org/wiki/API>.
84. Accord.NET Framework, 2017. Διαθέσιμο στο: <http://accord-framework.net/>.
85. Wikipedia, 2021. Καλιφόρνια. Διαθέσιμο στο: <https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CE%BB%CE%B9%CF%86%CF%8C%CF%81%CE%BD%CE%B9%CE%B1>.
86. Wikipedia, 2021. Τέξας. Διαθέσιμο στο: <https://el.wikipedia.org/wiki/%CE%A4%CE%AD%CE%BE%CE%B1%CF%82>.

Παράρτημα

Κατάλογος συντομογραφιών

Διαδίκτυο των Πραγμάτων.....	ΔτΠ
Internet of Things	IoT
Επιχειρηματική Ευφυΐα – Business Intelligence	BI
Ηνωμένες Πολιτείες της Αμερικής.....	Η.Π.Α.
Global Positioning System	GPS
Μέσα Μαζικής Μεταφοράς.....	M.M.M.
Machine-to-Machine.....	M2M
Bluetooth Low Energy (χαμηλής ενέργειας).....	BLE
Radio-frequency	RF
Wireless fidelity.....	WiFi
Τοπικό Δίκτυο	LAN
Global System for Mobile Communications	GSM
Επικοινωνία Κοντινού Πεδίου – Near Field Communication.....	NFC
Long Range Wide Area Network	LoRaWAN
Δίκτυο Ευρείας Περιοχής.....	WAN
Advanced Message Queuing Protocol.....	AMQP
Πρωτόκολλο Ελέγχου Μετάδοσης/Πρωτόκολλο Διαδικτύου (Transmission-Control Protocol/Internet Protocol.....	TCP/IP
Constrained Application Protocol.....	CoAP
Πρωτόκολλο Μεταφοράς Υπερκειμένου ή HyperText Transfer Protocol.....	HTTP
User Datagram Protocol ή Universal Datagram Protocol	UDP
Extensible Messaging and Presence Protocol.....	XMPP
Instant Messenger (άμεσα μηνύματα)	IM
Extensible Markup Language	XML
Voice over Internet Protocol.....	VoIP
Data-Distribution Service	DDS

Message Queuing Telemetry Transport.....	MQTT
Representational State Transfer.....	REST
Quality of Service.....	QoS
Transport Layer Security.....	TLS
Secure Sockets Layer.....	SSL
Datagram Transport Layer Security.....	DTLS
Simple Authentication and Security Layer.....	SASL
Hypertext Transfer Protocol Secure.....	HTTPS
Προσωπικό Δίκτυο.....	PAN
Συσκευή σε Συσκευή.....	D2D
Συσκευή σε Διακομιστή.....	D2S
Διακομιστή σε Διακομιστή.....	S2S
Cloud Security Alliance.....	CSA
Groupe Spéciale Mobile Association's.....	GSMA
Open Web Application Security Project.....	OWASP
Machine-to-Human.....	M2H
General Data Protection Regulation – Γενικός Κανονισμός Προστασίας Δεδομένων Προσωπικού Χαρακτήρα.....	GDPR
Ευρωπαϊκή Ένωση.....	E.E.
Ηλεκτρονικοί Υπολογιστές.....	H/Y
Προ Χριστού.....	π.Χ.
Μετά Χριστού.....	μ.Χ.
International Business Machines Corporation.....	IBM
Σύστημα Σχεδιασμού Απαιτήσεων Υλικών – Material Requirements Planning Systems.....	MRP
Συναλλαγές υψηλής συχνότητας.....	HFT
Ακαθάριστο Εγχώριο Προϊόν.....	A.Ε.Π.
Hadoop Distributed File System.....	HDFS
Σύστημα Αλλαγής Δεδομένων – Continuous Damping Control.....	CDC
Τιμές διαχωρισμένες με κόμμα – Comma-Separated Values.....	CSV
Clustering LARge Applications.....	CLARA
Partitioning Around Medoids.....	PAM

Density-based spatial clustering of applications with noise	DBSCAN
Συνεργασία δημόσιου-ιδιωτικού τομέα για το Διαδίκτυο του Μέλλοντος – Future Internet Public-Private Partnership	FI-PPP
Mongo database	MongoDB
Application Programming Interface	API
Client URL	cURL
Transmission Control Protocol	TCP
JavaScript Object Notation	JSON
Ultralight.....	UL
Database.....	Db
Not Only SQL.....	NoSQL
Binary JSON.....	BSON
Structured Query Language.....	SQL
Create, Read, Update, Delete.....	CRUD
Structured English Query Language.....	SEQUEL
Αριθμός Ταυτοποίησης.....	ID
Κύριο Κλειδί – Primary Key	PK
Εξωτερικό Κλειδί – Foreign Keys.....	FK
DataBase Management System	DBMS
Transact – SQL.....	T – SQL
Μη Κυβερνητική Οργάνωση.....	M.K.O.
Free and Open-Source Software.....	FOSS
Lightweight Directory Access Protocol.....	LDAP
Cascading Style Sheets	CSS
Λειτουργικό Σύστημα.....	ΛΣ
Infrastructure as a service	IaaS
Linux Containers	LXC
Yet Another Markup Language	YAML
Extensible Markup Language	XML
Integrated Development Environment.....	IDE
Task-Based Asynchronous Pattern.....	TAP
Système international.....	SI

Τετραγωνικά Χιλιόμετρα.....τ.χλμ

Κατάλογος τμημάτων κώδικα

Τμήμα Κώδικα 1.....	66
Τμήμα Κώδικα 2.....	67
Τμήμα Κώδικα 3.....	67
Τμήμα Κώδικα 4.....	71
Τμήμα Κώδικα 5.....	71
Τμήμα Κώδικα 6.....	85
Τμήμα Κώδικα 7.....	86
Τμήμα Κώδικα 8.....	88
Τμήμα Κώδικα 9.....	90
Τμήμα Κώδικα 10.....	92
Τμήμα Κώδικα 11.....	93
Τμήμα Κώδικα 12.....	94
Τμήμα Κώδικα 13.....	96
Τμήμα Κώδικα 14.....	98
Τμήμα Κώδικα 15.....	98
Τμήμα Κώδικα 16.....	103