



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΛΕΤΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ ΑΛΓΟΡΙΘΜΩΝ ΚΑΤΑΝΟΜΗΣ  
ΠΟΡΩΝ ΣΕ ΔΙΚΤΥΑ ΙΟΤ ΜΕ ΤΕΧΝΟΛΟΓΙΕΣ ΥΠΟΛΟΓΙΣΤΙΚΗΣ  
ΟΜΙΧΛΗΣ

ΠΑΥΛΙΔΗΣ ΔΗΜΗΤΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος  
Επίκουρος Καθηγητής

Λαμία ..... έτος .....





ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΛΕΤΗ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ  
ΑΛΓΟΡΙΘΜΩΝ ΚΑΤΑΝΟΜΗΣ ΠΟΡΩΝ ΣΕ  
ΔΙΚΤΥΑ ΙΟΤ ΜΕ ΤΕΧΝΟΛΟΓΙΕΣ  
ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΟΜΙΧΛΗΣ

ΠΑΥΛΙΔΗΣ ΔΗΜΗΤΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος  
Επίκουρος Καθηγητής

Λαμία ..... έτος .....





UNIVERSITY OF  
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

STUDY AND SIMULATION OF RESOURCE  
ALLOCATION ALGORITHMS IN IOT  
NETWORKS WITH FOG COMPUTING  
TECHNOLOGIES

PAVLIDIS DIMITRIOS

FINAL THESIS

ADVISOR

Tziritas Nikolaos  
Assistant Professor

Lamia ..... year .....



«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις <sup>(1)</sup>, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ.), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ...../...../20.....

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»







## ΠΕΡΙΛΗΨΗ

---

Το Internet of Things (IoT) αποτελεί ένα σύστημα αλληλένδετων υπολογιστικών συσκευών, μηχανών, αντικειμένων και αισθητήρων που επιτρέπει την επεξεργασία και μεταφορά δεδομένων από και προς συσκευές μέσω του διαδικτύου ή άλλων δικτύων επικοινωνίας χωρίς να απαιτείται ανθρώπινη παρέμβαση, οπότε είναι αναμφίβολα ακμάζων και ολοένα και περισσότερο χρησιμοποιείται σε ποικιλία εφαρμογών.

Το cloud computing, λόγω της παροχής πληθώρας υπηρεσιών μέσω του διαδικτύου όπως αποθήκευση δεδομένων, δικτύωση, λογισμικό αλλά και εργαλείων όπως βάσεις δεδομένων και διακομιστές, αποτελεί μια από τις επικρατέστερες τάσεις στο χώρο της πληροφορικής.

Η παρούσα διπλωματική εργασία μελετά την τεχνολογία υπολογιστικής ομίχλης η οποία επεκτείνει το cloud computing στα άκρα των εκάστοτε δικτύων, πιο κοντά στους τελικούς χρήστες. Αποτελεί προς το παρόν την ιδανικότερη πλατφόρμα για το Internet of Things αλλά και πληθώρας υπηρεσιών καθώς η ιδέα είναι ότι η επεξεργασία των δεδομένων που λαμβάνονται από το περιβάλλον μέσω των διάφορων αισθητήρων συμβαίνει τοπολογικά πολύ πιο κοντά σε αυτούς και όχι σε cloud servers.

Στόχος της είναι η καλύτερη κατανόηση των διαφόρων εφαρμογών της τεχνολογίας υπολογιστικής ομίχλης σε δίκτυα IoT και αυτό επιτυγχάνεται με την περαιτέρω ανάλυση της αρχιτεκτονικής υπολογιστικής ομίχλης, την παράθεση σημειωσών διαφορών με το cloud computing και την αναφορά τόσο των πλεονεκτημάτων και των μειονεκτημάτων.

Μελετώνται οι εφαρμογές της αρχιτεκτονικής και έτσι γίνεται καλύτερα κατανοητή η πραγματική χρησιμότητα καθώς και τα οφέλη των υλοποιήσεων του Fog Computing.

Τέλος γίνεται μια προσομοίωση διαφόρων αλγορίθμων κατανομής πόρων με τη χρήση του εργαλείου iFogSim και παρατίθενται τα αποτελέσματα των προσομοιώσεων αυτών.

## ABSTRACT

---

The Internet of Things (IoT) is a system of interconnected computing devices, machines, objects, and sensors that allows the processing and transfer of data to and from devices via the Internet or other communication networks without the need for human intervention, so it is undoubtedly prosperous and increasingly mostly used in a variety of applications.

Cloud computing, due to providing a variety of services through the Internet such as data storage, networking, software, and tools such as databases and servers, is one of the prevailing trends in the field of information technology.

This thesis studies the fog computing technology which extends the cloud computing to the ends of the respective networks, closer to the end users. It is currently the ideal platform for the Internet of Things and a variety of services as the idea is that the processing of data received from the environment through various sensors occurs topologically much closer to them and not to cloud servers.

Its goal is to better understand the various applications of cloud computing technology in IoT networks, and this is achieved by further analyzing the cloud computing architecture, listing significant differences with cloud computing and reporting both advantages and disadvantages.

The applications of the architecture are studied and thus the real usability as well as the benefits of the Fog Computing implementations are better understood.

Finally, various resource allocation algorithms are simulated using the iFogSim tool and the results of these simulations are presented.



## Table of Contents

---

ΠΕΡΙΛΗΨΗ .....	I
ABSTRACT .....	III
<b><u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ .....</u></b>	<b><u>2</u></b>
1.1 ΤΟ ΟΡΑΜΑ ΤΟΥ ΙΟΤ .....	4
1.2 Η ΕΝΝΟΙΑ ΤΟΥ CLOUD COMPUTING.....	5
1.3 Η ΕΝΝΟΙΑ ΤΟΥ FOG COMPUTING .....	6
<b><u>ΚΕΦΑΛΑΙΟ 2 ΕΙΣΑΓΩΓΙΚΟΙ ΟΡΙΣΜΟΙ ΚΑΙ ΕΝΝΟΙΕΣ .....</u></b>	<b><u>7</u></b>
2.1 CLOUD COMPUTING .....	7
2.1.Α ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ CLOUD COMPUTING.....	7
2.1.Β ΜΟΝΤΕΛΑ ΥΠΗΡΕΣΙΩΝ ΤΟΥ CLOUD COMPUTING.....	8
2.1.Γ ΤΥΠΟΙ CLOUD COMPUTING .....	9
2.2 FOG COMPUTING .....	10
2.2.Α ΠΕΡΙΓΡΑΦΗ ΤΟΥ FOG COMPUTING .....	10
2.2.Β ΑΡΧΙΤΕΚΤΟΝΙΚΗ OPEN FOG CONSORTIUM .....	12
2.2.Γ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ FOG COMPUTING .....	14
2.2.Δ ΟΙ ΠΡΟΚΛΗΣΕΙΣ ΓΙΑ ΤΟ FOG COMPUTING.....	15
2.2.Ε ΕΦΑΡΜΟΓΕΣ ΤΟΥ FOG COMPUTING .....	16
2.3 INTERNET OF THINGS (ΙΟΤ) .....	19
2.3.Α ΟΡΙΣΜΟΣ ΤΟΥ ΙΟΤ .....	19
2.3.Β ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΙΟΤ .....	19
2.3.Γ ΕΦΑΡΜΟΓΕΣ ΤΟΥ ΙΟΤ.....	20
<b><u>ΚΕΦΑΛΑΙΟ 3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΟΜΙΧΛΗΣ.....</u></b>	<b><u>22</u></b>
3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ FOG N-ΚΟΜΒΩΝ.....	22
3.2 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ OPENFOG.....	25
3.2.Α ΟΠΤΙΚΗ ΚΟΜΒΟΥ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΑΝΑΦΟΡΑΣ OPENFOG .....	27
3.2.Β ΟΠΤΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΑΝΑΦΟΡΑΣ OPENFOG .....	29
3.2.Γ ΟΠΤΙΚΗ ΛΟΓΙΣΜΙΚΟΥ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΑΝΑΦΟΡΑΣ OPENFOG.....	30
<b><u>ΚΕΦΑΛΑΙΟ 4 RELATED WORK.....</u></b>	<b><u>34</u></b>
4.1 ΚΑΤΑΝΟΜΗ ΠΟΡΩΝ .....	34
4.2 DATA PLACEMENT ΑΛΓΟΡΙΘΜΟΙ.....	35
4.2.Α EDGE-WARD ΑΛΓΟΡΙΘΜΟΣ .....	36
4.2.Β CLOUD-ONLY ΑΛΓΟΡΙΘΜΟΣ .....	38
4.3 LOAD BALANCING ΑΛΓΟΡΙΘΜΟΙ .....	39
4.3.Α ΑΛΓΟΡΙΘΜΟΣ MINIMUM LOAD.....	42
4.3.Β ΑΛΓΟΡΙΘΜΟΣ MINIMUM DISTANCE .....	43

4.3.Γ ΑΛΓΟΡΙΘΜΟΣ MINIMUM HOP DISTANCE AND LOAD .....	44
4.3.Δ EFFICIENT RESOURCE ALLOCATION ΑΛΓΟΡΙΘΜΟΣ .....	45
<b><u>ΚΕΦΑΛΑΙΟ 5: ΕΡΓΑΛΕΙΟ IFOGSIM.....</u></b>	<b>47</b>
5.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΣΟΜΟΙΩΤΗ.....	47
5.2 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ .....	51
5.3 ΟΜΟΙΟΤΗΤΕΣ IFOGSIM ΚΑΙ CLOUDSIM .....	53
<b><u>ΚΕΦΑΛΑΙΟ 6: ΠΡΟΣΟΜΟΙΩΣΗ.....</u></b>	<b>54</b>
6.1 ΠΕΙΡΑΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	55
6.1.Α ΠΕΙΡΑΜΑ 1 <sup>ο</sup> .....	55
6.1.2 ΠΕΙΡΑΜΑ 2 <sup>ο</sup> .....	56
6.1.3 ΠΑΡΑΤΗΡΗΣΕΙΣ ΕΠΙ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ .....	56
6.2 ΠΑΡΑΜΕΤΡΟΙ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΣΥΣΚΕΥΩΝ .....	57
6.3 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	59
6.3.Α ΚΑΤΑΝΑΛΩΣΗ ΕΝΕΡΓΕΙΑΣ.....	59
6.3.Β ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ.....	62
6.3.Γ ΧΡΗΣΗ ΔΙΚΤΥΟΥ.....	64
<b><u>ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u></b>	<b>66</b>
<b><u>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u></b>	<b>67</b>
<b><u>ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ.....</u></b>	<b>68</b>



## ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

---

Το διαδίκτυο είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων υπολογιστών και συσκευών, το πλήθος των οποίων είναι τεράστιο. Τα προβλήματα που δημιουργούνται από την προσπάθεια επικοινωνίας, ανταλλαγής δεδομένων αλλά και παροχής υπηρεσιών από υπολογιστή σε υπολογιστή, από συσκευή σε συσκευή, από τελικό χρήστη σε τελικό χρήστη και εν γένει από έναν απομακρυσμένο κόμβο σε έναν άλλο ενδιαμέσο ή τελικό κόμβο, πληθαίνουν και καθιστούν τα παραπάνω εξαιρετικά χρονοβόρα, ολοένα και πιο δαπανηρά και ενέχουν κινδύνους ασφαλείας και ιδιωτικότητας.

Όταν αναφερόμαστε σε διαδίκτυο των πραγμάτων (Internet of Things), εννοούμε δίκτυα τέτοια τα οποία αποτελούνται από αντικείμενα στα οποία είναι ενσωματωμένα αισθητήρες και ηλεκτρονικά κυκλώματα. Αυτά συλλέγουν γνωρίσματα, επεξεργάζονται και αποστέλλουν δεδομένα προκειμένου να παρέχουν απαραίτητες απαιτούμενες πληροφορίες σε έναν ή παραπάνω χρήστες είτε συσκευές. Είναι το παρόν και μέλλον στην παρακολούθηση τοπικού είτε ευρέως περιβάλλοντος πληροφοριών προκειμένου να καθίσταται δυνατή η πρόληψη συμβάντων, η επαρκής και έγκαιρη ενημέρωση για τα εκάστοτε χαρακτηριστικά του περιβάλλοντος καθώς και η ανθρώπινη είτε αυτοματοποιημένη παρέμβαση μέσω διασυνδεδεμένων συσκευών για την πρόληψη ασφαλείας είτε τη διόρθωση ή παραμετροποίηση των γνωρισμάτων του. Το IoT περιστρέφεται γύρω από την τεχνολογία πληροφοριών (information technology) και επικοινωνίας και ως εκ τούτου είναι ευνόητοι οι λόγοι για τους οποίους ενσωματώνεται με το cloud computing.

Το πλήθος των εφαρμογών του IoT είναι ολοένα αυξανόμενο και εκτείνονται σε πολλούς κλάδους όπως των τηλεπικοινωνιών, της ενέργειας και των αυτοκινήτων. Στον τομέα των οικιακών, τα έξυπνα σπίτια είναι εξοπλισμένα με έξυπνους θερμοστάτες, έξυπνες συσκευές θέρμανσης, φωτισμού και ηλεκτρικές συσκευές που μπορούν να ελέγχονται απομακρυσμένα μέσω τοπικού δικτύου είτε του διαδικτύου (cloud) είτε αυτοματοποιημένα βάσει των δεδομένων που συλλέγουν αισθητήρες από το περιβάλλον και να μεταβάλλουν μεταβλητές όπως θερμοκρασία, φωτεινότητα στο χώρο, υγρασία και άλλα. Στη γεωργία τα έξυπνα συστήματα που βασίζονται στο IoT βοηθούν στην παρακολούθηση της θερμοκρασίας, της υγρασίας χώρου και εδάφους και της ηλιακής ακτινοβολίας χρησιμοποιώντας διασυνδεδεμένους αισθητήρες καθώς και στην αυτοματοποιημένη είτε κατά απαίτηση λειτουργία συστημάτων άρδευσης. Στα οδικά δίκτυα η παρακολούθηση μέσω ασύρματων αισθητήρων και καταγραφής οπτικοακουστικού υλικού συλλέγει πληροφορίες και ενημερώνει τους οδηγούς μέσω εφαρμογών πλοήγησης για τυχών ατυχήματα, κυκλοφοριακές συμφορήσεις, οδικά έργα και προτείνει εναλλακτικές διαδρομές που οδηγούν στην ομαλή, ασφαλή και συνεχή κίνηση των οχημάτων και μέσων μεταφορών.

Η ευελιξία σύνδεσης χρηστών εξ αποστάσεως σε εφαρμογές αλλά και η πρόσβαση σε υπηρεσίες μέσω του διαδικτύου αλλά και του cloud computing είναι ένα από τα κυριότερα πλεονεκτήματα του διαδικτύου των πραγμάτων.

Οι υπηρεσίες αυτές είναι προσιτές, εύκολα προσβάσιμες και απευθύνονται στον ευρύτερο πληθυσμό αλλά και σε εξειδικευμένες ομάδες ατόμων και ως αποτέλεσμα είναι τεράστιο το πλήθος τελικών χρηστών που τις χρησιμοποιούν. Οι παραπάνω συνθήκες παρέχουν επίσης και την ευελιξία στον εκάστοτε πάροχο υπηρεσιών να προσφέρει τις υπηρεσίες του σε περισσότερους τελικούς χρήστες, παραμετροποιημένες κατάλληλα έτσι ώστε να καλύπτουν τις ανάγκες του εκάστοτε χρήστη.

Με απλά λόγια, το cloud computing είναι η παροχή παραμετροποιημένων υπολογιστικών υπηρεσιών, όπως διακομιστών (servers), αποθήκευσης δεδομένων, δικτύωσης, λογισμικού, ανάλυσης δεδομένων καθώς και τεχνητής νοημοσύνης, μέσω του διαδικτύου. Το αποτέλεσμα είναι η ευρεία δυνατότητα προσφοράς ευέλικτων πόρων, ενός τεράστιου συνόλου εφαρμογών, υποδομών αποθήκευσης και διαχείρισης δεδομένων και επικοινωνίας σε ένα τεράστιο αριθμό τελικών χρηστών αλλά και συσκευών.

Λόγω της φύσης του διαδικτύου, στο οποίο βασίζεται το cloud computing, προκύπτουν θέματα που αφορούν την αλλοίωση της ποιότητας των υπηρεσιών αλλά και την προστασία των δεδομένων και την ασφάλεια των χρηστών. Τα cloud είναι κυρίως προσβάσιμα με τη χρήση διαδικτύου και αυτό σημαίνει ότι αιτήματα χρηστών, αποκρίσεις συστημάτων καθώς και η μετάδοση δεδομένων από και προς τον τελικό χρήστη περνούν από πληθώρα κόμβων και ως συνέπεια και τοπικών δικτύων. Ως αποτέλεσμα δημιουργούνται χρονικές καθυστερήσεις, αλλοίωση των δεδομένων και ανάγκη επαναμετάδοσης τους καθώς και θέματα που αφορούν την ακεραιότητα αλλά και ιδιωτικότητα των δεδομένων αυτών.

Σε αυτό το σημείο είναι που το fog computing αναλαμβάνει την ταχεία και αξιόπιστη εξυπηρέτηση των επιθυμητών διαδικασιών των εκάστοτε τελικών χρηστών. Η αποκεντρωμένη υπολογιστική υποδομή της υπολογιστικής ομίχλης επιτρέπει σε δεδομένα, διαδικασίες αποθήκευσης, εφαρμογές αλλά ακόμη και κάποιους πολύπλοκους υπολογισμούς να λαμβάνουν χώρα μεταξύ ενός οποιουδήποτε cloud και της εκάστοτε πηγής παραγωγής δεδομένων. Τα πλεονεκτήματα και η πληθώρα των υπολογιστικών και όχι μόνο δυνατοτήτων των cloud έρχεται πιο κοντά στον τελικό χρήστη με ειδοποιό διαφορά την αποφυγή υποβάθμισης της αποτελεσματικότητας της υπολογιστικής διαδικασίας ή την αύξηση του χρόνου εξυπηρέτησης ενός αιτήματος καθώς και την αποφυγή δημιουργίας κενών ασφαλείας ή παραβίασης της ιδιωτικότητας των δεδομένων.

Αξίζει να σημειωθεί ότι το fog computing δεν αντικαθιστά το cloud ή τις cloud-based υπηρεσίες, αλλά συμπληρώνει το cloud computing. Η υπολογιστική ομίχλη επιτρέπει τη βραχυπρόθεσμη ανάλυση δεδομένων στα άκρα του δικτύου, ενώ το cloud εκτελεί διαδικασίες μακροπρόθεσμης ανάλυσης δεδομένων. Παρά το γεγονός ότι στα δίκτυα IoT οι αισθητήρες αλλά και οι συσκευές ακμής ή αλλιώς edge devices είναι αυτά που συλλέγουν ερεθίσματα και παράγουν δεδομένα, δεν διαθέτουν πολλές φορές το υλικό είτε λογισμικό προκειμένου να αποθηκεύσουν μεγάλο όγκο δεδομένων ή να εκτελέσουν προηγμένους υπολογισμούς. Οι διακομιστές cloud στον αντίποδα αυτού, έχουν τη δυνατότητα εκπλήρωσης των παραπάνω αλλά είναι συχνά απομακρυσμένοι, τόσο ώστε να μην μπορούν να ανταποκρίνονται έγκαιρα.

## 1.1 Το όραμα του IoT

---

Το Internet of Things αποτελεί τοπικό ή ευρύτερο δίκτυο συσκευών που συλλέγουν ποικιλία περιβαλλοντικών γνωρισμάτων και μεταβολών μέσω αισθητήρων, μεταφράζουν και επεξεργάζονται τα δεδομένα που προκύπτουν και μέσω τεχνολογιών λογισμικού επικοινωνούν και αποστέλλουν δεδομένα μεταξύ συσκευών εντός τοπικού δικτύου είτε σε απομακρυσμένες συσκευές μέσω του διαδικτύου και άλλων ενδιάμεσων δικτύων.

Η τεχνολογία γύρω από το IoT έχει γνωρίσει καταγιστική πρόοδο λόγω της πληθώρας διαφόρων συγκλινουσών τεχνολογιών που μπορούν να έχουν εφαρμογές σε τέτοια δίκτυα. Από τα πιο απλά υπολογιστικά συστήματα, αισθητήρες που μπορούν να διαβάσουν με μεγάλη ακρίβεια τις παραμέτρους του περιβάλλοντος, τα ολοένα και πιο ισχυρά ενσωματωμένα συστήματα αλλά και η επιστήμη και εφαρμογή μεθόδων μηχανικής μάθησης (machine learning), το IoT δηλώνει παρόν σε συστήματα ελέγχου και αυτοματισμού και είναι πολλά υποσχόμενο για το μέλλον τέτοιων συστημάτων.

Υπόσχεται την ενσωμάτωση και ως εκ τούτου τη μετατροπή των εμπορικών ηλεκτρικών και οικιακών συσκευών σε λειτουργικά μέρη ενός δικτύου IoT. Δεδομένου αυτού, ανοίγονται πόρτες σε νέες τεχνολογικές καινοτομίες που θα επιτρέψουν την περεταίρω επικοινωνία μεταξύ συσκευών και θα δημιουργήσουν νέους τύπους αλληλεπίδρασης μεταξύ συσκευών και ανθρώπων, προς όφελος των τελευταίων.

Εφαρμογές όπως έξυπνες πόλεις, υποδομές και υπηρεσίες για τη βελτίωση της ποιότητας ζωής καθώς και υπηρεσίες βιώσιμης κατανομής και χρήσης περιβαλλοντικών πόρων αλλά και πόρων συστημάτων. Η συνεχής ανάλυση και παραμετροποίηση χαρακτηριστικών του περιβάλλοντος αποτελεί για το IoT το δρόμο για τη βελτίωση ποιότητας ζωής, της αποφυγή βραχυπρόθεσμων είτε μακροπρόθεσμων δυσμενών καταστάσεων, τον βέλτιστο σχεδιασμό για αντιμετώπιση εγγενών προβλημάτων που απειλούν μακροπρόθεσμα το περιβάλλον και τις ζωές μας.

Η επικρατέστερη τάση που χαρακτηρίζει το IoT είναι η ραγδαία ανάπτυξη αλλά και μαζική παραγωγή ολοένα και περισσότερων τελικών συσκευών που μπορούν να συνδεθούν στο διαδίκτυο και να ελεγχθούν μέσω αυτού. Το ευρύ φάσμα καθημερινών δραστηριοτήτων που μπορούν να μετατραπούν σε λιγότερο χρονοβόρες αλλά εξίσου αποτελεσματικές με τη χρήση τέτοιων συσκευών είναι τόσο μεγάλο που είναι εύκολα αντιληπτό ότι υπάρχει πλήθος ιδιαιτεροτήτων που διαφέρουν από συσκευή σε συσκευή. Εάν συλλογιστούμε ότι μεταξύ αυτών υπάρχουν βασικά χαρακτηριστικά που μοιράζονται οι εν λόγω συσκευές, καταλήγουμε σε ευκαιρίες άμεσης ενσωμάτωσης του εκάστοτε τοπικού περιβάλλοντος σε υπολογιστικά συστήματα IoT με αποτελέσματα να καταλήγουμε σε εμφανώς αυξημένη αποδοτικότητα, οικονομικά οφέλη, στη μείωση ανθρωπογενών παρεμβολών στο περιβάλλον αλλά και στην ελάττωση καταπόνησης και καταβολής ανθρώπινης προσπάθειας.

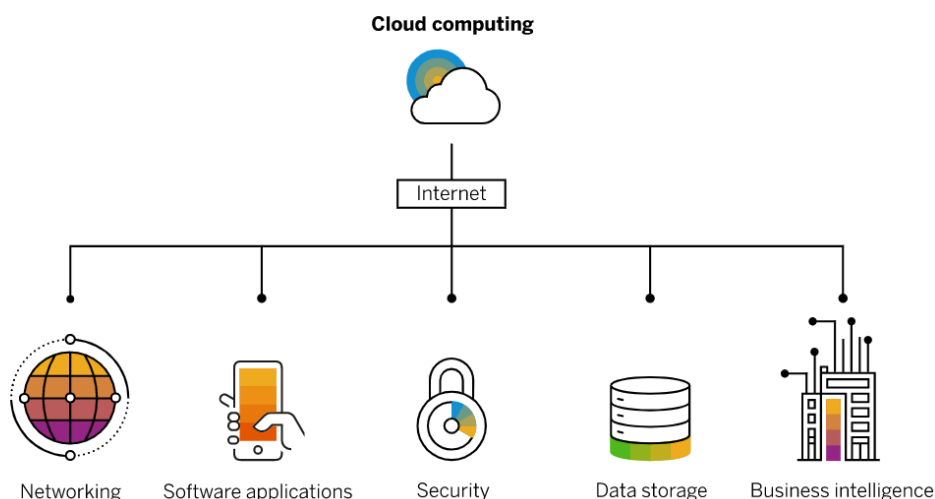
## 1.2 Η έννοια του cloud computing

Το cloud computing είναι η παραμετροποιημένη και on demand διάθεση υπολογιστικών πόρων, όπως η αποθήκευση δεδομένων και η παροχή υπηρεσιών υπολογιστικής ισχύος, χωρίς να απαιτείται η άμεση ενεργή διαχείριση από τον τελικό χρήστη.

Προκειμένου να μην επιβαρύνεται το τελικό υπολογιστικό σύστημα και να αποφεύγεται η αποθήκευση δεδομένων σε σκληρούς δίσκους, μέσω του cloud computing καθίσταται δυνατή η αποθήκευση σε απομακρυσμένες βάσεις δεδομένων. Απαιτείται η συνεχής πρόσβαση στο διαδίκτυο προκειμένου να είναι προσβάσιμα τα πρότερα αποθηκευμένα δεδομένα καθώς και τα διάφορα προγράμματα λογισμικού για την διαχείρισή τους.

Το cloud computing είναι μια δημοφιλής επιλογή για ανθρώπους και επιχειρήσεις. Η ονομασία του προλογίζει το γεγονός ότι πρόκειται για έναν εικονικό χώρο στον οποίο βρίσκονται πληροφορίες και πληθώρα υπηρεσιών οι οποίες μπορούν να προσπελαύνονται εξ αποστάσεως. Οι εταιρείες που παρέχουν cloud υπηρεσίες, επιτρέπουν στους χρήστες να αποθηκεύουν δεδομένα σε απομακρυσμένους διακομιστές και στη συνέχεια να έχουν πρόσβαση στα δεδομένα αυτά μέσω του διαδικτύου. Λόγω της φύσης του, προάγει την εξ αποστάσεως εργασία, καθώς επιτρέπει στον εκάστοτε τελικό χρήστη να έχει πρόσβαση τόσο σε προσωπικά και εργασιακά δεδομένα όσο και σε εξειδικευμένο λογισμικό και εφαρμογές.

Το cloud computing αποσυμφορεί την εκάστοτε τελική συσκευή από την συσσώρευση και επεξεργασία δεδομένων και μεταφέρει τον όγκο δεδομένων αλλά και τη διαδικασία της επεξεργασίας σε τεράστια απομακρυσμένα υπολογιστικά συμπλέγματα με αποτέλεσμα δεδομένα, εφαρμογές και υπηρεσίες να είναι διαθέσιμα μέσω του διαδικτύου οπουδήποτε στον κόσμο.



Εικόνα 1.1: Cloud computing technology (<https://www.sap.com/>)

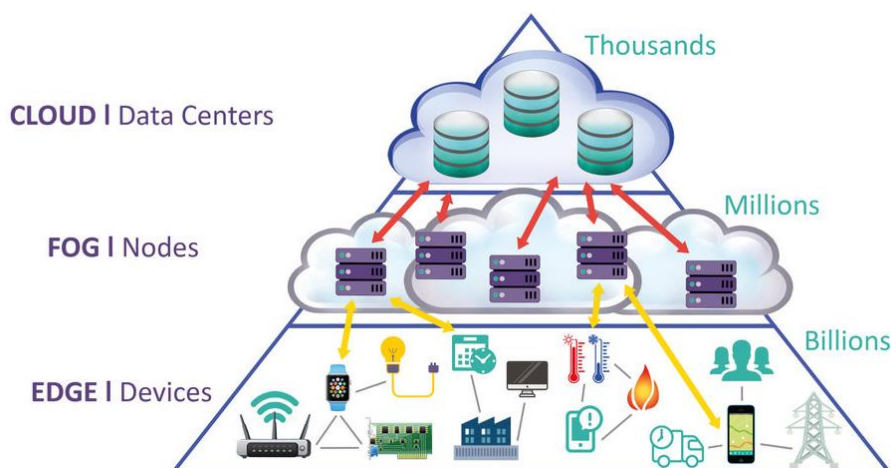
### 1.3 Η έννοια του fog computing

Το fog computing αποτελεί μια εναλλακτική λύση έναντι του cloud computing όσον αφορά την προσφορά υπηρεσιών σε τελικούς χρήστες. Η προσέγγιση αυτή πραγματεύεται το πρόβλημα της αύξησης των τελικών συσκευών και εκμεταλλεύεται την ευκαιρία που προκύπτει από το γεγονός ότι αυτές οι τελικές συσκευές παράγουν δεδομένα και πραγματοποιούν υπολογισμούς και μεταξύ τους μεταφορές δεδομένων και πληροφορίας στην άκρη του δικτύου.

Από τη στιγμή που τοπολογικά οι παραπάνω διαδικασίες μπορούν να πραγματοποιούνται κοντά στις τελικές συσκευές, αντί να δημιουργούνται δίαυλοι επικοινωνίας με clouds για την αποθήκευση και μεταφορά δεδομένων, το εύρος ζώνης περιορίζεται σε εύρος ζώνης σημείων πρόσβασης, όπως δρομολογητών. Ως συνέπεια αυτού, μειώνεται η καθολική ανάγκη για εύρος ζώνης, αφού η ανάγκη για μετάδοση δεδομένων μακριά από τελικές συσκευές σε απομακρυσμένα clouds και data centers μειώνεται δραματικά.

Στον τομέα της αποθήκευσης δεδομένων, υπάρχει μια σημαντική διαφορά μεταξύ cloud και fog computing. Στην αρχιτεκτονική υπολογιστικής ομίχλης, δεν απαιτείται η άμεση αποθήκευση ολόκληρου του όγκου δεδομένων στο cloud αλλά μόνο δεδομένων που δεν είναι βραχυπρόθεσμα αναγκαία, με αποτέλεσμα να υπάρχει η ευελιξία στρατηγικής διαχείρισης δεδομένων και αποδοτικότερης κατανομής πόρων συστήματος. Με αυτό τον σχεδιασμό, αυξάνεται η απόδοση και η αποτελεσματικότητα και μειώνεται σημαντικά το κόστος.

Μεταφέροντας τοπολογικά κομμάτι της υπολογιστικής επεξεργασίας και ανάλυσης δεδομένων πιο κοντά στον τελικό χρήστη και εντός ενός πλαισίου αρχιτεκτονικής υπολογιστικής ομίχλης, δίνεται η δυνατότητα στον τελικό χρήστη να αξιοποιήσει αποδοτικότερα την υπολογιστική ισχύ της εκάστοτε τελικής συσκευής, πράγμα το οποίο βελτιώνει την εμπειρία χρήσης και μειώνει τις επιβαρύνσεις στο cloud. Το fog computing είναι ιδιαίτερα σημαντικό για IoT συσκευές.



Εικόνα 1.2: Τα επίπεδα συσκευών σε υλοποιήσεις ομίχλης (leanBI: IoT and Predictive Analytics)

# ΚΕΦΑΛΑΙΟ 2 Εισαγωγικοί ορισμοί και έννοιες

## 2.1 Cloud computing

Με απλά λόγια, το cloud computing είναι η παροχή υπολογιστικών υπηρεσιών μέσω του διαδικτύου. Η χρήση υπηρεσιών όπως cloud servers, εφαρμογές αποθήκευσης, βάσεις δεδομένων, εφαρμογές δικτύωσης, λογισμικά, εφαρμογές analytics και υπηρεσίες τεχνητής νοημοσύνης προσφέρει δυνατότητες καινοτομιών και πρόσβαση σε ευέλικτους υπολογιστικούς πόρους που δεν μπορεί να υποστηρίξει το hardware πολλών τελικών συσκευών.

Σύμφωνα με το US National Institute for Standards and Technology, υπολογιστικό νέφος ορίζεται ως «ένα μοντέλο που δίνει τη δυνατότητα της συνεχούς, εύκολης και on demand υψηλών απαιτήσεων πρόσβαση σε ένα κοινόχρηστο σύνολο παραμετροποιήσιμων υπολογιστικών πόρων που μπορεί να εφοδιάσει και να διατεθεί γρήγορα, με ελάχιστη καταβολή προσπάθειας διαχείρισης ή αλληλεπίδραση με τον πάροχο υπηρεσιών. Αυτό το μοντέλο αποτελείται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα υπηρεσιών και τέσσερα μοντέλα ανάπτυξης». Παρακάτω θα αναλύσουμε τα βασικά χαρακτηριστικά του cloud computing, τα μοντέλα υπηρεσιών που είναι τέσσερα σε πλήθος και τα μοντέλα ανάπτυξης.

### 2.1.α Χαρακτηριστικά του cloud computing

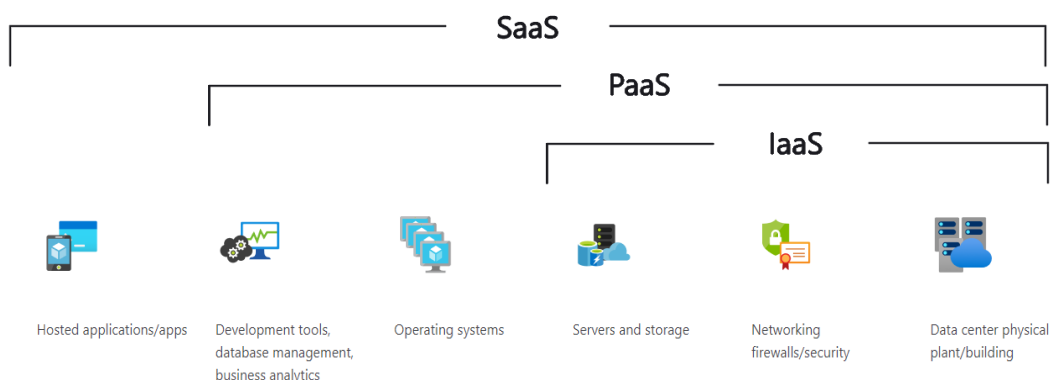
Το cloud computing αφορά τόσο τις υπηρεσίες, τις εφαρμογές και το λογισμικό που παρέχουμε όσο και στο hardware των απομακρυσμένων data centers και των παρόχων αυτών των υπηρεσιών. Βάσει ορισμού του cloud computing, τα βασικά χαρακτηριστικά είναι πέντε σε αριθμό και είναι τα εξής:

- **Κατ' απαίτηση self-service εξυπηρέτηση:** Ένας χρήστης υπηρεσιών cloud μπορεί να χαιρεί μονομερούς παροχής υπολογιστικών δυνατοτήτων όπου και όποτε απαιτείται χωρίς να είναι απαραίτητη η αλληλεπίδραση τελικού χρήστη και παρόχου υπηρεσιών
- **Ευρεία πρόσβαση στο διαδίκτυο:** Οι υπηρεσίες και υπολογιστικοί πόροι είναι διαθέσιμοι μέσω οποιασδήποτε συσκευής με δυνατότητα σύνδεσης στο διαδίκτυο και προσβάσιμοι μέσω τυπικών μηχανισμών που παραμετροποιούν τις υπηρεσίες και προωθούν τη χρήση αυτών από ετερογενείς πλατφόρμες πελατών
- **Συγκέντρωση πόρων (resource pooling):** Οι υπολογιστικοί πόροι του παρόχου υπηρεσιών συγκεντρώνονται για να εξυπηρετούν τους καταναλωτές χρησιμοποιώντας ένα multi-tenant μοντέλο, με φυσικούς και εικονικούς πόρους που εκχωρούνται δυναμικά ανάλογα με τη ζήτηση των καταναλωτών. Υπάρχει μια αίσθηση γεωγραφικής ανεξαρτησίας καθώς ο τελικός χρήστης δεν γνωρίζει την τοποθεσία των πόρων. Παράδειγμα τέτοιων πόρων είναι η αποθήκευση, επεξεργασία, μνήμη και bandwidth.
- **Άμεση και ταχεία ελαστικότητα:** Οι δυνατότητες παρέχονται και απελευθερώνονται ελαστικά, ακόμη και αυτοματοποιημένα, προκειμένου να δημιουργείται ταχεία κλιμάκωση και αποκλιμάκωση των υπολογιστικών αναλογιών ανάλογα με τη ζήτηση. Για τον καταναλωτή οι δυνατότητες φαντάζουν απεριόριστες και διαθέσιμες σε οποιαδήποτε ποσότητα ανά πάσα στιγμή, πράγμα το οποίο δεν συμβαίνει πάντα.
- **Μετρούμενη εξυπηρέτηση:** Τα cloud συστήματα ελέγχουν και βελτιστοποιούν αυτόματα τη χρήση των πόρων με ρυθμούς κατάλληλους για τον κάθε τύπο υπηρεσίας. Η χρήση των πόρων παρακολουθείται, παραμετροποιείται και παρατίθεται έτσι ώστε να υπάρχει διαφάνεια τόσο για τον τελικό χρήστη, όσο και για τον πάροχο υπηρεσίας.

## 2.1.β Μοντέλα υπηρεσιών του cloud computing

Οι περισσότερες cloud υπηρεσίες εμπίπτουν σε τέσσερις μεγάλες κατηγορίες, οι οποίες ονομάζονται και στοίβα του υπολογιστικού νέφους επειδή τοποθετούνται το ένα πάνω στο άλλο. Τα τέσσερα διαφορετικά μοντέλα είναι τα εξής:

- **Infrastructure as a Service (IaaS):** Το μοντέλο IaaS είναι ένας τύπος υπηρεσίας υπολογιστικού νέφους που προσφέρει on-demand βασικούς υπολογιστικούς, αποθηκευτικούς και δικτυακούς πόρους, σε μια pay-as-you-go βάση. Η αλλαγή μιας υποδομής σε μια λύση IaaS βοηθά στη μείωση κόστους συντήρησης των datacenter, στην μείωση εξόδων που αφορούν το hardware και στην απόκτηση επιχειρηματικών πληροφοριών σε πραγματικό χρόνο. Οι λύσεις IaaS παρέχουν την ευελιξία κλιμάκωσης των υπολογιστικών πόρων σε αναλογία με την ζήτηση, την δυνατότητα ταχείας παροχής υπηρεσιών και αυξημένη αξιοπιστία της συνολικής υποδομής.
- **Platform as a Service (PaaS):** Το μοντέλο PaaS αναφέρεται σε υπηρεσίες cloud που παρέχουν μια on-demand πλατφόρμα για την ανάπτυξη, τη δοκιμή, την παράδοση και τη διαχείριση εφαρμογών λογισμικού. Δίνει τη δυνατότητα αγοράς πόρων από παρόχους cloud υπηρεσιών, σε μια pay-as-you-go βάση και πρόσβασης σε αυτούς μέσω διαδικτύου. Το συγκεκριμένο μοντέλο έχει σχεδιαστεί για να διευκολύνει τη δημιουργία web ή mobile εφαρμογών, χωρίς φόβο για ρύθμιση ή διαχείριση της υποδομής διακομιστών, αποθήκευσης, δικτύου και βάσεων δεδομένων.
- **Software as a Service (SaaS):** Το μοντέλο SaaS είναι μια on-demand μέθοδος για την παράδοση εφαρμογών λογισμικού μέσω διαδικτύου και παρέχεται με συνδρομή. Παράδειγμα είναι τα εργαλεία ηλεκτρονικού ταχυδρομείου, ημερολογίου και γραφείου. Η υποδομή, το ενδιάμεσο λογισμικό, το λογισμικό εφαρμογών και τα δεδομένα εφαρμογών βρίσκονται στο datacenter του παρόχου υπηρεσιών και διαθέσιμες εφαρμογές ενοικιάζονται από χρήστες ή οργανισμούς και είναι προσβάσιμες μέσω προγραμμάτων περιήγησης ιστού.
- **Serverless computing:** Το μοντέλο Serverless computing επικεντρώνεται στη εξέλιξη της λειτουργικότητας εφαρμογών χωρίς να σπαταλάται χρόνος για διαχείριση υποδομής. Για εφαρμογές χωρίς server, ο πάροχος υπηρεσιών παραμετροποιεί και διαχειρίζεται αυτόματα την υποδομή που απαιτείται για την εκτέλεση αυτών. Στην πραγματικότητα υπάρχουν servers οι οποίοι εκτελούν εντολές, τρέχουν εφαρμογές και διαχειρίζονται υποδομή και πόρους οι οποίοι όμως είναι αόρατοι. Το μοντέλο είναι επεκτάσιμο και λειτουργεί βάσει συμβάντων, χρησιμοποιώντας πόρους και παραμετροποιώντας το περιβάλλον χρήστη ανάλογα με τη δραστηριότητα και τις εντολές.



Εικόνα 2.1: Μοντέλα υπηρεσιών cloud computing (<https://azure.microsoft.com/>)

## 2.1.γ Τύποι cloud computing

---

Υπάρχουν διαφορετικοί τύποι cloud με αξιοσημείωτες μεταξύ τους διαφορές και ως συνέπεια δεν είναι κάθε τύπος υπηρεσιών cloud κατάλληλος για όλους. Κάθε τύπος cloud-based υπηρεσιών, κάθε μοντέλο εξυπηρέτησης έχει εξελιχθεί και διαφοροποιηθεί κατάλληλα προκειμένου να προσφέρει αποδοτικότερες λύσεις για τις ανάγκες του εκάστοτε τελικού χρήστη.

Πρώτα από όλα ο τελικός χρήστης πρέπει να επιλέγει τον τύπο ή διαφορετικά την αρχιτεκτονική cloud στην οποία θα λάβουν χώρα οι υπηρεσίες cloud που καλύπτουν τις ανάγκες του. Υπάρχουν τέσσερις διαφορετικοί τύποι ανάπτυξης υπηρεσιών cloud:

- **Δημόσιο νέφος (public cloud):** Σε ένα δημόσιο cloud οι υπηρεσίες προσφέρονται από παρόχους μέσω του δημόσιου διαδικτύου, καθιστώντας τις υπηρεσίες διαθέσιμες σε όποιον θέλει να τις χρησιμοποιήσει ή να τις αγοράσει. Ενδέχεται να είναι διαθέσιμες δωρεάν είτε να πωλούνται on-demand, δίνοντας τη δυνατότητα στους τελικούς χρήστες να χρεώνονται μόνο για τη διάρκεια χρήσης ή το εύρος ζώνης που καταναλώνουν. Τα δημόσια cloud μπορούν να αναπτυχθούν ταχύτερα από ότι η υλοποίηση μιας τοπικής εσωτερικής υποδομής cloud για μια επιχείρηση και προσφέρουν μια σχεδόν απείρωσ επεκτάσιμη πλατφόρμα υπηρεσιών. Όσον αφορά ανησυχίες περί ιδιωτικότητας και ασφάλειας των χρηστών, σε περιπτώσεις όπου ο πάροχος χρησιμοποιεί τις κατάλληλες μεθόδους ασφαλείας, το δημόσιο νέφος είναι εξίσου ασφαλές με το ιδιωτικό.
- **Ιδιωτικό νέφος (private cloud):** Ένα ιδιωτικό cloud αναφέρεται σε ένα σύνολο υπολογιστικών πόρων και υπηρεσιών που χρησιμοποιούνται αποκλειστικά από μια και μόνο επιχείρηση ή οργανισμό. Η υλοποίηση ενός ιδιωτικού cloud μπορεί να βρίσκεται εντός του datacenter μιας επιχείρησης/οργανισμού είτε να προσφέρεται on-demand και επί πληρωμή από τους παρόχους υπηρεσιών. Ιδιωτικό είναι ένα τέτοιο νέφος του οποίου η υποδομή και υπηρεσίες τοποθετούνται σε ένα ιδιωτικό δίκτυο. Προσφέρει πολλά από τα πλεονεκτήματα ενός δημόσιου cloud όπως είναι η αυτοεξυπηρέτηση, η επεκτασιμότητα και ελαστικότητα με τη διαφορά ότι δίνεται η δυνατότητα επιπλέον ελέγχου και παραμετροποίησης αποκλειστικών υπολογιστικών πόρων και υπηρεσιών μιας ιδιωτικής υπολογιστικής υποδομής και μέσω ενός ιδιωτικού δικτύου.
- **Υβριδικό νέφος (Hybrid cloud):** Τα υβριδικά νέφη είναι ένας συνδυασμός δημόσιων και ιδιωτικών νεφών τα οποία είναι συνδεδεμένα μεταξύ τους με τεχνολογία που επιτρέπει την κοινή χρήση δεδομένων και εφαρμογών μεταξύ τους. Επιτρέποντας τη δυνατότητα ανταλλαγής και μεταφοράς δεδομένων και εφαρμογών μεταξύ ιδιωτικών και δημοσίων cloud, προσφέρεται μεγαλύτερη ευελιξία, περισσότερες επιλογές και δίνεται η δυνατότητα βελτιστοποίησης της υπάρχουσας υποδομής και ασφάλειας.
- **Multi-cloud:** Τα multi-clouds είναι υλοποίηση cloud που αποτελείται από περισσότερες από μια cloud-based υπηρεσίες που είναι διαθέσιμες από περισσότερους από έναν παρόχους cloud υλοποιήσεων, ιδιωτικών είτε δημόσιων. Όλα τα υβριδικά νέφη είναι multi-cloud αλλά δεν ισχύει το αντίστροφο. Ένα multi-cloud περιβάλλον μπορεί να δημιουργηθεί προκειμένου να καταστεί δυνατός ο καλύτερος έλεγχος των δεδομένων είτε ως επιπλέον χώρος αποθήκευσης για την βέλτιστη ανάκτηση δεδομένων σε περίπτωση καταστροφής. Διαφορετικά δημιουργείται κατά λάθος ως αποτέλεσμα σφαλμάτων του τμήματος IT μιας επιχείρησης.



## 2.2 Fog Computing

---

Η συνεχής πρόοδος επιστήμης και τεχνολογίας οδήγησε σε καινοτομίες που άλλαξαν τον τρόπο ζωής και την καθημερινότητά μας. Μεταξύ άλλων, μια από τις τεχνολογίες που υπόσχεται ένα ευκολότερο, ασφαλέστερο, πρακτικότερο και καλύτερα προετοιμασμένο αύριο είναι η αρχιτεκτονική και οι τεχνολογίες του IoT. Αυτοοδηγούμενα αυτοκίνητα, αυτόνομα συστήματα τροφοδοσίας, μονάδες παραγωγής ενέργειας, γεωργικές εκτάσεις, σούπερ μάρκετ, υγειονομική περίθαλψη, σχολεία καθώς και πολλές ακόμη και εφαρμογές εκμεταλλεύονται συσκευές και έξυπνα αντικείμενα που αποτελούν αναπόσπαστο κομμάτι του διαδικτύου και της ύπαρξης μας χωρίς να το γνωρίζουμε.

Το IoT κερδίζει συνεχώς έδαφος καθώς προσφέρει ένα τεράστιο εύρος υπηρεσιών και εφαρμογών. Παρά το γεγονός ότι ήδη αποτελεί έναν κλάδο στον τομέα της επιστήμης της πληροφορικής, οι τεχνολογίες IoT δεν μπορούν να λειτουργήσουν χωρίς την ενσωμάτωση τεχνολογιών που συλλέγουν και επεξεργάζονται δεδομένα για την βελτιστοποίηση των λειτουργιών τους. Αναφορικά με τέτοιες τεχνολογίες, μια αρχιτεκτονική που υλοποιεί όλα τα παραπάνω είναι αυτή του fog computing.

Τεράστιες ποσότητες δεδομένων, συλλέγονται από αισθητήρες για να αποθηκευτούν σε cloud datacenters, όπου αναλύονται για την εξαγωγή αποτελεσμάτων και την λήψη αυτοματοποιημένων μέτρων. Παρά το γεγονός ότι οι ταχύτητες επεξεργασίας δεδομένων έχουν αυξηθεί σημαντικά και ταχέως, το εύρος ζώνης για τη μεταφορά δεδομένων από και προς τα datacenters δεν έχει δει παρόμοια εξέλιξη. Από τη μια, η υποστήριξη της μεταφοράς δεδομένων από και προς δισεκατομμύρια IoT συσκευές γίνεται ολοένα και δυσκολότερο να επιτευχθεί λόγω του τεράστιου όγκου και της γεωγραφικής κατανομής αυτών. Από την άλλη υπάρχει ανάγκη μείωσης λανθάνοντος χρόνου, εκμηδενισμός απαιτήσεων συνδεσιμότητας και υποστήριξη επεξεργασίας και αποθήκευσης πιο κοντά στο σημείο όπου παράγονται τα δεδομένα.

Με την επέκταση του cloud computing και των cloud-based υπηρεσιών στα άκρα των IoT δικτύων, δίνεται η δυνατότητα να εικονικοποιηθεί και να εκμεταλλευτεί μια νέα ιεραρχία πόρων από τον πυρήνα προς την άκρη του δικτύου, όπου τα δεδομένα μπορούν να χρησιμοποιηθούν για την έγκαιρη λήψη αποφάσεων και την αμεσότερη υποστήριξη.

### 2.2.α Περιγραφή του fog computing

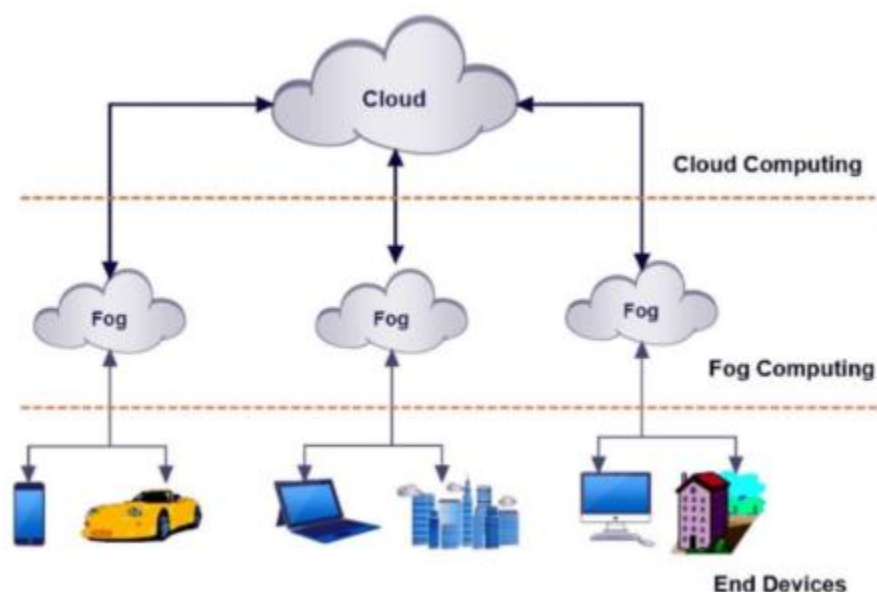
---

Το fog computing προτείνει την εισαγωγή υπολογιστικών κόμβων στα όρια των IoT δικτύων. Οι κόμβοι λειτουργούν ως φίλτρο, μειώνοντας την ποσότητα των δεδομένων που αποστέλλονται στο cloud αλλά και ως συσκευές που παρέχουν επεξεργαστική ισχύ πιο κοντά στο σημείο ανίχνευσης μεταβολών και συλλογής δεδομένων. Η αρχιτεκτονική ενός τέτοιου δικτύου αποτελείται συνήθως από τρία επίπεδα:

- **Επίπεδο τελικών συσκευών και αντικειμένων:** Αυτό το επίπεδο είναι το σημείο στη δομή του IoT όπου ξεκινά η συλλογή και δημιουργία μιας συνεχούς δεδομένων. Διαθέτει συσκευές δικτύου και συσκευές αισθητήρων που διαθέτουν πρωτόκολλα επικοινωνίας τα οποία καθιστούν δυνατή τη μετάδοση των παραγόμενων δεδομένων μέσω των ενδιάμεσων κόμβων στο επόμενο επίπεδο, αυτό της υπολογιστικής ομίχλης.
- **Επίπεδο υπολογιστικής ομίχλης:** Αυτό το επίπεδο αναμένεται να διαθέτει πληθώρα αποκεντρωμένων κόμβων κατανεμημένων στα όρια των IoT δικτύων. Η κύρια λειτουργία του επιπέδου αυτού είναι ο χειρισμός των δικτύων καθώς και η αποτελεσματική διαχείριση των δεδομένων που συνεχώς λαμβάνονται από αυτά.

Η κύρια υπολογιστική διαδικασία και επεξεργασία των δεδομένων συμβαίνει σε αυτό το επίπεδο με αποτέλεσμα οι IoT εφαρμογές να βελτιώνονται καθώς δημιουργείται καθεστώς ελέγχου μετάδοσης δεδομένων στο επίπεδο cloud και μειώνεται ο χρόνος απόκρισης αιτημάτων των εφαρμογών σε επίπεδο IoT.

- **Επίπεδο cloud:** Το επίπεδο αυτό είναι επίσης γνωστό και ως datacenter επίπεδο και θεωρείται το ανώτερο επίπεδο της αρχιτεκτονικής IoT-cloud. Σε αυτό το επίπεδο δίνεται η δυνατότητα πρόσβασης στο δίκτυο όλων των κοινόχρηστων πόρων, συσκευών και αντικειμένων των IoT δικτύων του πρώτου επιπέδου. Η αποθήκευση καθώς και άλλες cloud-based υπηρεσιών απαιτούν ισχυρά επεξεργαστικά συστήματα που μπορούν να ανταπεξέλθουν στο φόρτο εργασίας και εκτελούνται σε αυτό το επίπεδο.



Εικόνα 2.2: Επίπεδα αρχιτεκτονικής δικτύου fog [2]

Η υπολογιστική ομίχλη στοχεύει στην εκμετάλλευση μεγάλου αριθμού κόμβων ακμής, όπως φορητές συσκευές, κινητά τηλέφωνα, δρομολογητές κ.α., για να υποστηρίξει επιλεκτικά ευαίσθητες στο χρόνο, γεωγραφικά καταναμημένες εφαρμογές ή εφαρμογές κινητών συσκευών, μέσω των οποίων αισθητήρες και ενεργοποιητές χρησιμοποιούνται σε διαφορετικά πλαίσια και υπηρεσίες επεξεργασίας. Το fog computing έρχεται ως μια προσθήκη στα σενάρια IoT-Cloud, με την εισαγωγή ενός νέου επιπέδου συσκευών που μπορούν να τρέχουν υπηρεσίες και να φέρουν εις πέρας συγκεκριμένες ατομικές είτε επιχειρηματικές αποστολές, τόσο ανεξάρτητα όσο και παράλληλα με τις ήδη υπάρχουσες υποδομές και υπηρεσίες cloud. Ενώ το ίδιο το υπολογιστικό νέφος είναι η κεντρική ιδέα γύρω από την ανάπτυξη πολλών υλοποιήσεων δικτύων, η υπολογιστική ομίχλη αντιπροσωπεύει μια πρόταση αποστασιοποίησης από κλειστά συστήματα που εστιάζουν στο υπολογιστικό νέφος και εξαρτώνται από αυτό.

Ένα από τα προβλήματα που εγείρονται από το προαναφερθέν σενάριο είναι ο τρόπος με τον οποίο μπορεί να ελεγχθεί η πολυπλοκότητα της ανάπτυξης εφαρμογών υπολογιστικής ομίχλης, κυρίως λόγω της κλίμακας και της ετερογένειας της υποδομής του. Ορισμένες λειτουργίες είναι κατάλληλες για το cloud επίπεδο και άλλες ταιριάζουν περισσότερο στο επίπεδο της υπολογιστικής ομίχλης, ενώ άλλες μπορούν καλύτερα να εκχωρούνται δυναμικά σε διαφορετικούς κόμβους ανάλογα με τα χαρακτηριστικά της υποδομής.

## 2.2.β Αρχιτεκτονική Open Fog Consortium

Το 2011 η ανάγκη ανάπτυξης μιας νέας τεχνολογίας για την αντιμετώπιση διαχείρισης του τεράστιου σε αριθμό πλήθους των συσκευών IoT καθώς και του τεράστιου όγκου δεδομένων που αυτές παράγουν και επικοινωνούν, ώθησε την επιστημονική κοινότητα στη δημιουργία της αρχιτεκτονικής υπολογιστικής ομίχλης η οποία θα αποτελούσε προέκταση του υπολογιστικού νέφους και θα λειτουργούσε ως βοηθητικός μηχανισμός στη διανομή των δεδομένων, στην αποτελεσματική κατανομή και διαχείριση υπολογιστικών πόρων και στην εξάλειψη καθυστερήσεων απόκτησης δεδομένων ή απόκρισης εφαρμογών.

Λόγω της έλλειψης επαρκούς προτυποποίησης και απαιτήσεων υλικού και λογισμικού για την υλοποίηση συστημάτων υπολογιστικής ομίχλης, δεν ήταν δυνατή η ευρεία αναγνώριση, υιοθέτηση και χρήση τεχνολογιών υπολογιστικής ομίχλης σε τοπικά συστήματα αλλά και σε ευρύτερα συστήματα μεγάλης κλίμακας. Στα τέλη του 2015 ιδρύθηκε μια κοινοπραξία από εταιρείες κολοσσούς για τη δημιουργία μιας λύσης η οποία αποτελεί μέχρι και σήμερα πρότυπο στο οποίο βασίζεται η ανάπτυξη, εγκατάσταση και λειτουργία της αρχιτεκτονικής υπολογιστικής ομίχλης.

Οι απεριόριστες προσεγγίσεις υλοποιήσεων cloud-based συστημάτων αδυνατούν να προσεγγίσουν τις απαιτήσεις ταχύτητας και διαχείρισης του όγκου δεδομένων που απορρέει συνεχώς από δίκτυα IoT. Cisco, Intel, Microsoft, Dell, ARM και το πανεπιστήμιο του Princeton, ίδρυσαν το Open Fog Consortium το οποίο όρισε την αρχιτεκτονική υπολογιστικής ομίχλης, μιας αρχιτεκτονικής για την αντιμετώπιση των προκλήσεων υποδομής και συνδεσιμότητας με έμφαση στην επεξεργασία δεδομένων και την επέκταση ευφυών συστημάτων πιο κοντά στις τελικές συσκευές. Μεταξύ άλλων, όρισε τα πρότυπα ενσωμάτωσης τεχνολογιών υπολογιστικής ομίχλης σε μια ευρεία σειρά περιπτώσεων δικτύων και συστημάτων επικοινωνίας.

Στην τεχνική αναφορά που δημοσιεύτηκε το 2017 με τίτλο «OpenFog Reference Architecture for Fog Computing» ορίζεται ένα σύνολο βασικών αρχών της αρχιτεκτονικής υπολογιστικής ομίχλης που αναφέρονται και ως πυλώνες της αρχιτεκτονικής. Αυτές οι βασικές αρχές αντιπροσωπεύουν τα χαρακτηριστικά που πρέπει να διαθέτει και τις προϋποθέσεις που πρέπει να πληροί ένα σύστημα προκειμένου να μπορεί να ενσωματώσει την οριζόντια αρχιτεκτονική της υπολογιστικής ομίχλης σε επίπεδο συστήματος.



Εικόνα 2.3: Οι βασικοί πυλώνες της αρχιτεκτονικής αναφοράς του OpenFog (<https://www.iiconsortium.org>)

Οι οκτώ βασικοί πυλώνες της αρχιτεκτονικής υπολογιστικής ομίχλης είναι οι παρακάτω:

- **Ασφάλεια (Security):** Η ανομοιογένεια των δικτύων, η ετερογένεια του υλικού του εκάστοτε κόμβου και η ανομοιομορφία του λογισμικού που διαθέτει, σε συνδυασμό με την πληθώρα διαφορετικών υλοποιήσεων ασφαλείας, απορρήτου, ανωνυμίας, ιδιωτικότητας και αξιοπιστίας είναι μια σειρά προκλήσεων που δύσκολα μπορεί κανείς να ξεπεράσει. Γι' αυτό η βασική αρχή της ασφάλειας περιγράφει τους μηχανισμούς ασφαλείας που μπορούν να εφαρμοστούν σε έναν κόμβο προκειμένου να καταστεί ασφαλής. Η παραμετροποίηση των μηχανισμών για να ταιριάζουν και να καλύπτουν τις απαιτήσεις ασφαλείας έγκειται στις δυνατότητες υλικού και λογισμικού.
- **Επεκτασιμότητα (Scalability):** Η βασική αρχή της επεκτασιμότητας αναφέρεται στις δυναμικές τεχνικές και την ελαστικότητα που παρέχεται από τις εφαρμογές της αρχιτεκτονικής ομίχλης. Με την ενσωμάτωση τεχνολογιών ομίχλης, συστήματα και μεμονωμένοι κόμβοι μπορούν να κλιμακωθούν μέσω προσθηκών λογισμικού και υλικού προκειμένου να μπορούν να ακολουθήσουν τη απαιτήσεις της ζήτησης. Η ευελιξία και η συνεχής δυνατότητα επεκτασιμότητας είναι καίριας σημασίας για τις εφαρμογές ομίχλης και αφορά πολλές διαφορετικές πτυχές της αρχιτεκτονικής όπως την απόδοση, την δυνατότητα κλιμακούμενης χωρητικότητας, την αξιοπιστία, την ασφάλεια τελικών συσκευών και κόμβων, την επεκτασιμότητα λογισμικού και υλικού.
- **Ανοιχτότητα (Openness):** Η ανοιχτότητα είναι απαραίτητη για την επιτυχία ενός συστήματος υπολογιστικής ομίχλης λόγω της επιθυμίας των τελικών χρηστών για πλήρως διαλειτουργικά συστήματα που υποστηρίζονται από ένα ενεργό οικοσύστημα προμηθευτών. Η ανοιχτότητα δίνει τη δυνατότητα στους κόμβους υπολογιστικής ομίχλης να μπορούν να τοποθετούνται οπουδήποτε εντός του δικτύου.
- **Αυτονομία (Autonomy):** Η δυνατότητα υποστήριξης αυτονομίας σε όλο το εύρος δικτύου, επιτρέπει στους κόμβους ομίχλης να εξακολουθούν να παρέχουν την προκαθορισμένη υπηρεσία και διαλειτουργικότητα ακόμη και σε περιπτώσεις αστοχιών και σφαλμάτων της εξωτερικής υπηρεσίας. Η δυνατότητα αυτονομίας εντός των ορίων του δικτύου σημαίνει ότι πλέον η λήψη αποφάσεων δεν είναι αποκλειστική ευθύνη του υπολογιστικού νέφους αλλά δημιουργεί προϋποθέσεις προκειμένου να είναι επαρκής η επεξεργαστική ισχύς των τελικών συσκευών για την επεξεργασία των δεδομένων και την εκπλήρωση της αρχικής αποστολής του δικτύου.
- **Δυνατότητα προγραμματισμού (Programmability) :** Μέσω της δυνατότητας προγραμματισμού υλικού και λογισμικού των κόμβων υπολογιστικής ομίχλης δημιουργείται ένα προσαρμοστικό δίκτυο που μπορεί να ενσωματώσει δυναμικές αναπτύξεις και να δρα αυτοματοποιημένα. Λόγω της αυξημένης λειτουργικότητας των κόμβων ομίχλης η δυνατότητα αυτή είναι αναγκαία για την εξέλιξη της αρχιτεκτονικής.
- **Αξιοπιστία, διαθεσιμότητα και επισκευασιμότητα (Reliability, Availability and Serviceability):** Η επιτυχία μιας υλοποίησης είναι η παροχή της σχεδιασμένης λειτουργικότητας ακόμη και υπό συνθήκες στρες του συστήματος. Η διαθεσιμότητα έγκειται στην άμεση και ανεξάρτητη από το χρόνο και τη γεωγραφική κατανομή παροχή υπηρεσιών. Η δυνατότητα έγκαιρου εντοπισμού σφαλμάτων λογισμικού και αστοχιών υλικού διασφαλίζει την άμεση αποκατάσταση βλαβών και στο σύνολό της σημαίνει τη σωστή λειτουργία και την συνεχή παροχή υπηρεσιών.
- **Ευελιξία (Agility Pillar):** Λόγω της κλίμακας και του όγκου δεδομένων που παράγονται από τις τελικές συσκευές, η ανάλυση των δεδομένων και η εξαγωγή χρήσιμων αναλυτικών στοιχείων, συμπερασμάτων και λύσεων δεν είναι δυνατή μόνο με την

αξιοποίηση της ανθρώπινης αντίληψης και ικανότητας. Μέσω της ευελιξίας που παρέχουν τα συστήματα υπολογιστικής ομίχλης, παράγονται σχεδόν άμεσα χρήσιμες πληροφορίες και αναλυτικά στοιχεία μαζί με όλες τους τις προεκτάσεις. Έτσι μειώνεται ο φόρτος εργασίας δικτύου και δίνεται η δυνατότητα έγκαιρης λήψης αποφάσεων και προσαρμογής σε αναμενόμενες ή απρόβλεπτες μεταβολές περιβάλλοντος.

- **Ιεραρχία (Hierarchy Pillar):** Αν και η υπολογιστική ιεραρχία και ιεραρχία συστήματος δεν είναι απαραίτητες σε όλες τις υλοποιήσεις συστημάτων υπολογιστικής ομίχλης, μπορούν και εφαρμόζονται στις περισσότερες υλοποιήσεις. Όπως προαναφέρθηκε, η υπολογιστική ομίχλη είναι μια υλοποίηση που συμπληρώνει την αρχιτεκτονική υπολογιστικού νέφους και ως συνέπεια αυτού επιτρέπει πέρα από την μεμονωμένη λειτουργία υπολογιστικού νέφους και υπολογιστικής ομίχλης, την από κοινού λειτουργία αυτών. Βάσει των λειτουργικών απαιτήσεων των δικτύων IoT, συντελείται μια λογική ιεραρχία πόρων συστήματος υπολογιστικής ομίχλης από τα αντικείμενα έως και το ανώτερο επίπεδο παραγωγής αναλυτικών στοιχείων και λήψης αποφάσεων. Ανάλογα με την κλίμακα και τη φύση του σεναρίου που αντιμετωπίζεται, η ιεραρχία μπορεί να είναι ένα δίκτυο διαμερισμένων έξυπνων και συνδεδεμένων συστημάτων διατεταγμένων σε φυσικά ή λογικά επίπεδα είτε σε ένα ενιαίο φυσικό σύστημα.

### 2.2.γ Χαρακτηριστικά του fog computing

Οι τεχνολογίες υπολογιστικής ομίχλης έχουν ένα μεγάλο εύρος γνωρισμάτων που τις χαρακτηρίζουν. Μερικά από αυτά παρατίθενται παρακάτω:

- **Ετερογένεια:** Η υπολογιστική ομίχλη αποτελεί μια εικονικοποιημένη πλατφόρμα που παρέχει υπηρεσίες υπολογισμού, αποθήκευσης και δικτύωσης μεταξύ τελικών συσκευών και κέντρα δεδομένων cloud. Οι υπολογιστικοί πόροι, οι πόροι αποθήκευσης και δικτύωσης είναι τόσο δομικά στοιχεία της υπολογιστικής ομίχλης, όσο και του υπολογιστικού νέφους.
- **Γεωγραφική κατανομή:** Τα δίκτυα ομίχλης κατανέμουν γεωγραφικά κόμβους και αισθητήρες βάσει του εκάστοτε σεναρίου με αποτέλεσμα να δίνεται η δυνατότητα υποστήριξης ταχύτερης ανάλυσης δεδομένων, δυνατότητες άμεσης και αυτοματοποιημένης λήψης αποφάσεων και καλύτερη ποιότητα υπηρεσιών ανεξαρτήτως της τοποθεσίας του τελικού χρήστη.
- **Υποστήριξη δικτύων αισθητήρων μεγάλης κλίμακας:** Η ομίχλη διαθέτει ένα χαρακτηριστικό που εφαρμόζεται όταν το σύστημα παρακολούθησης του περιβάλλοντος επεκτείνει εγγενώς τα συστήματα παρακολούθησης που προκαλούνται από τις ιεραρχικές απαιτήσεις των πόρων υπολογισμού και αποθήκευσης.
- **Υποστήριξη κινητικότητας:** Είναι σημαντικό για πολλές εφαρμογές ομίχλης να έχουν τη δυνατότητα επικοινωνίας με κινητές ή φορητές συσκευές και ως συνέπεια αυτού υποστηρίζουν πρωτόκολλα και τεχνικές υποστήριξης φορητότητας.
- **Μείωση χρόνων απόκρισης και καθυστέρησης:** Οι περισσότερες υπηρεσίες έξυπνων εφαρμογών είναι ανεπαρκείς λόγω της έλλειψης υποστήριξης υπηρεσιών στην άκρη του δικτύου. Η ομίχλη υποστηρίζει την παροχή υπηρεσιών στα όρια του δικτύου συμπεριλαμβανομένων και εφαρμογών με απαιτήσεις χαμηλού λανθάνοντος χρόνου.
- **Real-time αλληλεπίδραση:** Η δυνατότητα αλληλεπίδρασης σε πραγματικό χρόνο αφορά μια σειρά από συνδυασμένες δράσεις και είναι ένα από τα χαρακτηριστικά των τεχνολογιών ομίχλης. Δίνεται η δυνατότητα παρακολούθησης και επεξεργασίας του περιβάλλοντος σε πραγματικό χρόνο αντί για τμηματική επεξεργασία των δεδομένων.

- **Διαλειτουργικότητα:** Η αδιάκοπη υποστήριξη ποικιλίας υπηρεσιών απαιτεί τη συνεργασία διαφορετικών εφαρμογών και υπηρεσιών που ενδέχεται να διατίθενται από διαφορετικούς παρόχους. Σε δίκτυα ομίχλης η διαλειτουργικότητα που τα χαρακτηρίζει δίνει την ευελιξία σε υπηρεσίες και εφαρμογές να μπορούν να υποστηρίζονται και να εκτελούνται απρόσκοπτα προκειμένου να προκύπτουν με βέλτιστο τρόπο αναλυτικά στοιχεία και προληπτικές αποφάσεις.
- **Δυνατότητες μάθησης και ανταπόκριση σε πελατοκεντρικές απαιτήσεις:** Η πρόσβαση σε δεδομένα και αναλυτικά στοιχεία παρέχει καλύτερη οπτική σχετικά με τις απαιτήσεις των χρηστών, καλύτερη τοποθέτηση για τη διάθεση λειτουργιών μετάδοσης, αποθήκευσης και ελέγχου από το δίκτυο IoT στο cloud και αντίστροφα. Λόγω της εγγύτητας των κόμβων στους τελικούς χρήστες, οι εφαρμογές μπορούν να παρέχουν αποδοτικότερη και ευκολότερη διαχείριση των ήδη υπάρχοντων αλλά των πιθανά επερχόμενων απαιτήσεων των χρηστών.

### 2.2.6 Οι προκλήσεις για το fog computing

Οι προκλήσεις και τα εμπόδια σε δίκτυα ομίχλης μπορούν να διαχωριστούν σε εγγενή εμπόδια που βασίζονται και περιορίζονται από την διαθέσιμη τεχνολογία και σε εξωτερικά εμπόδια που εμπíπτουν σε νομοθεσίες και θέματα περί απορρήτου και προσωπικών δεδομένων. Τα ενυπάρχοντα εμπόδια αφορούν περιορισμούς που συναντώνται κατά την χρήση και αξιοποίηση των υπηρεσιών ομίχλης όπως:

- **Έλλειψη υπηρεσιών για τελικές συσκευές:** Το νέφος παρέχει πλατφόρμες και υπηρεσίες που η ομίχλη δεν είναι σε θέση να παρέχει ακόμη σε αυτό το βαθμό. Δεν υπάρχουν υπηρεσίες οργάνωσης, on-demand αποθήκευσης και υπηρεσίες διαχείρισης δεδομένων στο επίπεδο ομίχλης για τις τελικές συσκευές.
- **Έλλειψη τυποποιημένου υλικού:** Υπάρχει μια μικρή ομάδα συσκευών που μπορούν να χρησιμοποιηθούν για υλοποιήσεις συστημάτων IoT που αξιοποιούν στο έπακρο την αρχιτεκτονική, τις υπηρεσίες και τις δυνατότητες της ομίχλης. Το γεγονός ότι υπάρχει ποικιλία στις δυνατότητες και στο λογισμικό της εκάστοτε ενδιάμεσης συσκευής που εκτελεί χρέη κόμβου σημαίνει ότι η αρχιτεκτονική ομίχλης θα πρέπει να είναι αρκετά ευέλικτη ώστε να μπορεί να υποστηρίξει ποικιλία υλικού και λογισμικού ετερογενών συσκευών και θα έχει να αντιμετωπίσει διαφορετικές χωρητικότητες πόρων.
- **Διαχείριση:** Η ομίχλη αποσκοπεί στο να καταστήσει την πληροφορία και τις υπηρεσίες διαχείρισης δεδομένων ως έννοιες πιο οικείες και προσιτές στον τελικό χρήστη. Πράγμα το οποίο σημαίνει ότι είναι επιτακτική η δημιουργία μεγάλου αριθμού κόμβων ομίχλης μεταξύ τελικών συσκευών και cloud datacenter. Η αποτελεσματική διαχείριση και η βέλτιστη τοποθέτηση των κόμβων στην τοπολογία δικτύου και η παραμετροποιημένη συνεργασία τους με συσκευές και υπηρεσίες νέφους είναι μια διαδικασία που δεν έχει οριστικοποιηθεί και δεν προσεγγίζει ακόμη την βέλτιστη έκδοση της. Η διαχείριση της ροής δεδομένων και η συνεργασία των cloud-based υπηρεσιών με το υλικό και λογισμικό του επιπέδου ομίχλης αποτελεί ακόμη πρόβλημα των παρόχων υπηρεσιών.
- **Διαχείριση QoS:** Όσο το πλήθος των κόμβων αυξάνεται, η κατανομή τους καθίσταται ολοένα και πιο δύσκολη και η υλοποίηση της αρχιτεκτονικής γίνεται δυσκολότερη. Τα προβλήματα διαχείρισης οδηγούν σε εκπτώσεις που αφορούν την ποιότητα υπηρεσιών (λανθάνων χρόνος, καθυστερήσεις-απώλειες πακέτων δεδομένων, κατατμήσεις, αποτυχία δικτύου). Σε εφαρμογές μεγάλης κλίμακας, η εξέλιξη υλικού και λογισμικού στην αρχιτεκτονική ομίχλης επιλύει κάποια από τα παραπάνω προβλήματα, αλλά όχι όλα τα προβλήματα (σφάλματα κατανομής και συνεργασίας των κόμβων).

Όλα τα παραπάνω επηρεάζουν την υποδομή, τις εφαρμογές και τις υπηρεσίες της ομίχλης που αξιοποιούνται από δίκτυα IoT τα οποία προϋποθέτουν ποιότητα στις υπηρεσίες που προσφέρουν. Ένα ακόμη πρόβλημα που προκύπτει είναι το εξής:

- **Έλλειψη διαφάνειας δικτύου:** Εφαρμογές η λειτουργία των οποίων συνδέεται με καταναμημένα συστήματα, αποκρύπτουν συνεχώς πληροφορίες που αφορούν την κίνηση δεδομένων εντός του δικτύου, πράγμα το οποίο σημαίνει ότι οι εφαρμογές θα εκτελούνται όπως θα εκτελούνταν σε μια μόνο συσκευή και όχι με παραμετροποιημένες προσθήκες προκειμένου να ανταποκρίνονται σε κάθε συσκευή.

Τα εξωτερικά εμπόδια αφορούν περιορισμούς από εξωτερικούς παράγοντες όπως επίδοξους online επιδρομείς που αποσκοπούν πρόσβαση σε προσωπικά δεδομένα είτε περιορισμούς κυβερνητικών υπηρεσιών και ανεπάρκειας υπηρεσιών παρόχων.

- **Ασφάλεια:** Τα cloud datacenter μπορούν να επιβάλλουν μια σειρά πολύπλοκων μέτρων για τον έλεγχο πρόσβασης σε υπηρεσίες και προσωπικά δεδομένα πράγμα το οποίο δεν είναι εφικτό σε υλοποιήσεις ομίχλης λόγω κλίμακας δικτύου και ευρείας κατανομής των κόμβων με αποτέλεσμα να υπάρχουν κενά ασφαλείας. Υπάρχει μια μικρή σειρά από αντίμετρα που μπορούν να ληφθούν για να εξαλείψουν μερικά από τα υπάρχοντα κενά αλλά συνεχίζουν να υπάρχουν κενά ασφαλείας που αφορούν χαρακτηριστικά και ιδιότητες συστήματος που δεν μπορούν ακόμη να αντιμετωπιστούν αποτελεσματικά.
- **Νομικές και ρυθμιστικές απαιτήσεις:** Για ένα μέρος εφαρμογών και υπηρεσιών ενέχουν νομικοί περιορισμοί που απαιτούν την διατήρηση των δεδομένων σε μια συγκεκριμένη τοποθεσία καθώς και την συγκεκριμενοποίηση των υπηρεσιών αποθήκευσης και επεξεργασίας αυτών. Τα cloud datacenters μπορούν να αποτελούν μέρη τέτοιων συστημάτων μόνο εάν η τοποθεσία και ο τύπος του cloud είναι συγκεκριμένα, αλλά για συστήματα ομίχλης αυτό δεν είναι δυνατό. Λόγω της φύσης της αρχιτεκτονικής και του λογισμικού του νέφους, υπάρχει διαφάνεια ροής και διασποράς των δεδομένων ενώ σε υλοποιήσεις ομίχλης λόγω της εικονικοποίησης πόρων, οι εφαρμογές δεν γνωρίζουν την πραγματική τοποθεσία των δεδομένων.

### 2.2.ε Εφαρμογές του fog computing

Μεταξύ cloud και fog computing υπάρχει μια αμοιβαία αλληλεξάρτηση. Ορισμένες λειτουργίες προορίζονται για εκτέλεση και επεξεργάζονται αποτελεσματικότερα σε κόμβους ομίχλης, ενώ άλλες ταιριάζουν καλύτερα στο νέφος. Παρά το γεγονός ότι η ομίχλη κερδίζει συνεχώς έδαφος, το νέφος παραμένει και θα συνεχίσει να αποτελεί ένα μεγάλο και σημαντικό μέρος των υπολογιστικών συστημάτων.

Η αρχιτεκτονική αναφοράς του OpenFog επιτρέπει την επικοινωνία μεταξύ fog και cloud καθώς και μεταξύ κόμβων ομίχλης και προσφέρει πολλά πλεονεκτήματα:

- **Ασφάλεια:** Πρόσθετοι παράμετροι ασφαλείας για την επίτευξη αξιόπιστων και ακέραιων συναλλαγών.
- **Γνώση:** Επίγνωση των πελατοκεντρικών στόχων προκειμένου να επιτυγχάνεται αυτονομία.
- **Ευελιξία:** Δυνατότητα γρήγορης και προσιτής κλιμάκωσης εντός των πλαισίων μιας δεδομένης και κοινής υποδομής.
- **Λανθάνον χρόνος:** Επεξεργασία δεδομένων σε πραγματικό χρόνο και εξάλειψη χρονικών καθυστερήσεων και σφαλμάτων.
- **Αποδοτικότητα:** Δυναμική διαχείριση χρησιμοποιούμενων και μη πόρων από τις τελικές συσκευές.

Τα δεδομένα που παράγονται από ένα τοπικό σύστημα αισθητήρων θα μπορούσαν να μεταφερθούν σε ένα cloud για επεξεργασία δεδομένων, εύρεση και ανάλυση σφαλμάτων και ανεπιθύμητων μεταβολών και στη συνέχεια αποστολή των δεδομένων στο τοπικό δίκτυο προκειμένου να γίνει καταγραφή, παραμετροποίηση και μεταβολές στο υλικό είτε λογισμικό της αρχικής υλοποίησης. Σε αυτό το σενάριο υπάρχουν παρόλα αυτά πολλά προβλήματα.

Η συνεχής διατήρηση μεγάλου εύρους ζώνης για την συνεχή μεταφορά δεδομένων ενέχει μεγάλο κόστος ενώ οι απομακρυσμένες συνδέσεις για τη μεταφορά δεδομένων θα μπορούσαν να αποτελέσουν στόχο για hacker. Μια αστοχία σύνδεσης, μια απρόσμενη διακοπή λειτουργίας ή ακόμη και υπερφόρτωσης του νέφους σημαίνει ότι δεδομένα αποστέλλονται στο κενό και ο έλεγχος των ενεργειών του τοπικού συστήματος χάνεται. Τέλος η απόσταση μεταξύ τοπικού συστήματος και νέφους σημαίνει ότι σε περιπτώσεις όπου η άμεση ενημέρωση, πρόβλεψη και επίλυση προβλημάτων είναι ζωτικής σημασίας, η καθυστέρηση που μεσολαβεί κατά την μεταφορά δεδομένων και αναλυτικών στοιχείων μπορεί να έχει ως αποτέλεσμα αστοχίες και ατυχήματα χρηματικού και ηθικού κόστους.

Με στρατηγική και δυναμική τοποθέτηση κόμβων στα όρια του τοπικού δικτύου, ένα μεγάλο μέρος προβλημάτων και επικείμενων κινδύνων μπορούν να αποφευχθούν, ενώ το κόστος της τεχνολογίας συστημάτων αισθητήρων και συσκευών μετρήσεων έχει μειωθεί με αποτέλεσμα να είναι προσβάσιμες αλλά και προσιτές. Η ενσωμάτωση κόμβων ομίχλης στο τοπικό δίκτυο σημαίνει ότι υπάρχει τοπικότητα στην επεξεργασία με αποτέλεσμα να μειώνεται ο κίνδυνος επίθεσης. Σε περιπτώσεις σφαλμάτων είτε επικείμενων κινδύνων, υπάρχει άμεση ενημέρωση και παροχή λύσεων με αποτέλεσμα να αποφεύγονται ανεπιθύμητα αποτελέσματα. Με την μεταφορά της πλειοψηφίας των λειτουργιών λήψης αποφάσεων στα όρια του τοπικού δικτύου, μόνο η περιστασιακή δημιουργία διεπαφών με το νέφος δημιουργεί μια ιεραρχία και ένα ανώτερο σύστημα ελέγχου.

Η αρχιτεκτονική αναφοράς του OpenFog περιγράφει μια γενικευμένη υλοποίηση ομίχλης ευέλικτη και ικανή να παραμετροποιηθεί σε ανάγκες εφαρμογών, παρέχοντας επιχειρηματική αξία για εφαρμογές IoT που απαιτούν επεξεργασία δεδομένων και λήψη αποφάσεων σε πραγματικό χρόνο, χαμηλό λανθάνων χρόνο, προηγμένες παραμέτρους ασφάλειας που εκτείνεται μόνο στα όρια του τοπικού δικτύου. Παρακάτω παρατίθενται μερικές περιπτώσεις ενσωμάτωσης ομίχλης σε εφαρμογές της καθημερινότητας.

- **Μεταφορές, έξυπνα οχήματα και έλεγχος κυκλοφορίας:** Πέρα από τα επίγεια μέσα μεταφοράς υπάρχουν θαλάσσια και εναέρια μέσα μεταφοράς ανθρώπινου δυναμικού και εμπορευμάτων. Με την ενσωμάτωση αισθητήρων, έξυπνων υποδομών και τεχνολογιών ομίχλης για αυτόνομες μεταφορές, τα οχήματα θα μπορούσαν να αποτελούν κόμβους ομίχλης εντός ενός ευρύτερου τοπικού δικτύου όπου δεδομένα συλλέγονται και επεξεργάζονται από τα οχήματα προτού αποσταλούν στα υπόλοιπα οχήματα εντός δικτύου. Με συνεχή ενημέρωση για παραμέτρους περιβάλλοντος καθίσταται δυνατή η παροχή υπηρεσιών όπως αυτόνομη οδήγηση και μεταφορές, υπηρεσίες ασφαλείας και αποφυγής ατυχημάτων, υπηρεσίες για την αποφυγή συσσώρευσης οχημάτων και ως συνέπεια τις αποτελεσματικότερες μεταφορές και μετακινήσεις. Ένα άλλο σενάριο προτείνει την τοποθέτηση κόμβων ομίχλης στρατηγικά τοποθετημένους στις προκαθορισμένες αρτηρίες μεταφορών έτσι ώστε να συλλέγουν πληροφορίες που αφορούν καιρικές συνθήκες, πιθανές δυσμενών συνθηκών σε σημεία της διαδρομής, ενημέρωση για ατυχήματα και πιθανές εναλλακτικές διαδρομές και πολλές ακόμη υπηρεσίες.



- **Έξυπνες πόλεις:** Η ενσωμάτωση τεχνολογιών ομίχλης σε σενάρια έξυπνης πόλης καλούνται να αντιμετωπίσουν προκλήσεις όπως η κυκλοφοριακή συμφόρηση, η δημόσια ασφάλεια, η αποδοτικότερη διαχείριση ενέργειας με στόχο την ελαχιστοποίηση κατανάλωσης, την επίτευξη στόχων υγιεινής και διατήρηση υγιούς περιβάλλοντος για τους κατοίκους, την ενσωμάτωση ευφυών νοσοκομείων για την διασφάλιση ενός συστήματος υγείας προσανατολισμένο στον άνθρωπο και την ελεύθερη, δημόσια και ασφαλή σύνδεση στο διαδίκτυο και στις υπηρεσίες αυτού. Επίσης η υποδομή ομίχλης φιλοδοξεί να αναδιαμορφώσει τη βιομηχανία με ευφυή εργοστάσια που εκμεταλλεύονται λογισμικό και έξυπνες συσκευές για την πρόληψη ελλείψεων πρώτων υλών, αποφυγή ατυχημάτων εργατικής φύσης και την αποδοτικότερη παραγωγή με την ελαχιστοποίηση του περιβαλλοντικού κόστους.
- **Συστήματα παρακολούθησης υγείας:** Με την ανάπτυξη του κατάλληλου πλαισίου για την επεξεργασία και μετάδοση δεδομένων σε πραγματικό χρόνο μέσω κόμβων ομίχλης, καθίσταται δυνατή η συνεχής παρακολούθηση της δημόσιας υγείας. Η συνεχής παρακολούθηση των παραμέτρων υγείας είναι ο εξέχων στόχος στην επιστήμη της ανάλυσης βιοϊατρικών δεδομένων προκειμένου να λαμβάνονται απαραίτητα προληπτικά μέτρα και να γίνονται σωστές προβλέψεις για πτυχές που αφορούν την υγειονομική περίθαλψη είτε επερχόμενες ενδημικές είτε πανδημικές νοσήσεις.
- **Συστήματα ασφαλείας και παρακολούθησης:** Τα συστήματα παρακολούθησης παράγουν τεράστιο όγκο δεδομένων και χρησιμοποιούνται για την παρακολούθηση υλικών αγαθών και χώρων αποθήκευσης καθώς και για την ανθρώπινη ασφάλεια. Πολλές καθημερινές δραστηριότητες βασίζονται ολοένα σε αισθητήρες, κάμερες και συστήματα παρακολούθησης για την ασφάλεια του ανθρώπου, τον εντοπισμό μη εξουσιοδοτημένης πρόσβασης και την εν γένει αύξηση των μέτρων ασφαλείας, την αύξηση αξιοπιστίας των συστημάτων παρακολούθησης και την αποτελεσματικότητα καταστολής συμβάντων παράκαμψης μέτρων ασφαλείας. Ο τεράστιος όγκος δεδομένων που θα έπρεπε να αποσταλεί στο νέφος θα σήμαινε αδυναμία άμεσης ενημέρωσης για κενά ασφαλείας και λήψης αποφάσεων για την καταστολή αυτών σε πραγματικό χρόνο. Η παρακολούθηση και ο εντοπισμός γεγονότων παράκαμψης μέτρων ασφαλείας σε πραγματικό χρόνο απαιτούν αυστηρά χαμηλούς χρόνους απόκρισης και σχεδόν αμελητέες καθυστερήσεις πράγμα το οποίο σημαίνει ότι η έννοια της επικαιρότητας είναι κρίσιμης σημασίας τόσο για το κομμάτι της ανίχνευσης κενών ασφαλείας όσο και για το κομμάτι της άμεσης απόκρισης και καταστολής με την ομίχλη να δίνει λύσεις στα παραπάνω προβλήματα.
- **Συστήματα ασύρματων αισθητήρων και δίκτυα ενεργοποιητών:** Τα δίκτυα αισθητήρων ενσωματώνονται παραδοσιακά σε εφαρμογές για την παρακολούθηση του περιβάλλοντος και τον εντοπισμό σφαλμάτων είτε ανεπιθύμητων συμβάντων. Σε συστήματα που ενσωματώνουν δίκτυα ασύρματων αισθητήρων, τα δίκτυα ενεργοποιητών μπορούν να αποτελούν κόμβους ομίχλης για την επεξεργασία και ανάλυση των δεδομένων που συλλέγουν οι αισθητήρες του τοπικού συστήματος και σε συνδυασμό με τις δυνατότητες των συσκευών ενεργοποίησης είναι δυνατή η αυτοματοποιημένη ενημέρωση και αντίδραση του εκάστοτε συστήματος με μια σειρά ενεργειών, μηχανισμών και εφαρμογών που θα τεθούν σε λειτουργία για την αποφυγή δυσμενών καταστάσεων.

## 2.3 Internet of Things (IoT)

---

### 2.3.α Ορισμός του IoT

---

Το IoT είναι ένα δίκτυο φυσικών αντικειμένων, όπως έξυπνες συσκευές, οχήματα με δυνατότητα σύνδεσης στο διαδίκτυο και ενσωματωμένα με πληθώρα αισθητήρων, έξυπνων κτιρίων και άλλων αντικειμένων τα οποία περιέχουν ενσωματωμένα ηλεκτρονικά συστήματα, λογισμικό, αισθητήρες και δυνατότητα σύνδεσης στο διαδίκτυο. Τα αντικείμενα αυτά, είτε βρίσκονται στο ίδιο τοπικό δίκτυο, είτε σε απομακρυσμένα δίκτυα, έχουν τη δυνατότητα συλλογής πληροφοριών, μετάφραση αυτών σε δεδομένα και ανταλλαγή δεδομένων μεταξύ συσκευών.

Με την αξιοποίηση υπηρεσιών διαδικτύου, cloud-based εφαρμογών και λογισμικού, τα αντικείμενα αυτά και τα δεδομένα που συλλέγουν μπορούν να παρακολουθούνται σε πραγματικό χρόνο απομακρυσμένα, πράγμα το οποίο ανοίγει το δρόμο για την ενσωμάτωση του φυσικού κόσμου με υπολογιστικά συστήματα πράγμα το οποίο έχει ως αποτέλεσμα την αποτελεσματικότερη παρακολούθηση, την αυτοματοποιημένη ή on demand επέμβαση για την παραμετροποίηση συστήματος, την μείωση του κόστους διαχείρισης πόρων καθώς και την ακρίβεια στη καταγραφή και ανάλυση δεδομένων. Λόγω της φύσης των συσκευών, κάθε μια μπορεί να λειτουργήσει αυτόνομα και κατά μονάδα είτε σε συνεργασία με τις υπόλοιπες της τοπικής ή ευρύτερης διαδικτυακής υποδομής.

Προκειμένου να γίνει καλύτερα κατανοητή η έννοια του IoT είναι απαραίτητο να γίνει διαχωρισμός μεταξύ του διαδικτύου (internet) και του παγκόσμιου ιστού (world wide web), δύο εννοιών που συχνά συγχέονται αλλά εσωκλείουν διαφορετικό περιεχόμενο, εφαρμογές και υποδομή. Ως διαδίκτυο ορίζεται ένα σύστημα διασυνδεδεμένων υπολογιστικών δικτύων που χρησιμοποιεί το πρωτόκολλο ελέγχου μετάδοσης TCP/IP προκειμένου να επιτρέπει την επικοινωνία μεταξύ δικτύων και συσκευών. Πρόκειται για ένα δίκτυο ιδιωτικών, δημοσίων, ακαδημαϊκών, επιχειρηματικών και κυβερνητικών δικτύων τοπικής είτε παγκόσμιας εμβέλειας που συνδέονται μέσω ενός ευρέως φάσματος ηλεκτρονικών, ασύρματων και οπτικών τεχνολογιών δικτύωσης. Η κύρια λειτουργία του είναι η γρήγορη και αξιόπιστη μεταφορά δεδομένων μεταξύ συσκευών με ασφαλή τρόπο. Μεταξύ του ευρέως φάσματος πόρων και υπηρεσιών πληροφοριών που αναλαμβάνει να μεταφέρει, είναι και οι εφαρμογές του παγκόσμιου ιστού (www). Ο παγκόσμιος ιστός είναι ένα επίπεδο εφαρμογών που λειτουργεί επάνω στο διαδίκτυο. Είναι ένα σύστημα πληροφοριών, όπου έγγραφα και πόροι ιστού καθίστανται εύκολα προσβάσιμα μέσω web διευθύνσεων URL και είναι προσβάσιμα μόνο με τη χρήση διαδικτύου.

### 2.3.β Χαρακτηριστικά του IoT

---

Παρά το γεγονός ότι η έννοια του IoT υφίσταται εδώ και πολύ καιρό, το γεγονός ότι δύναται να καθίσταται ως πρακτική σήμερα οφείλεται σε μια σειρά διαφορετικών τεχνολογιών. Μερικές εξ αυτών είναι:

- **τεχνολογίες αισθητήρων χαμηλού κόστους και χαμηλής ισχύος:** οι οικονομικά προσιτοί και αξιόπιστοι αισθητήρες καθιστούν τις τεχνολογίες IoT πιο προσιτές
- **συνδεσιμότητα:** πληθώρα πρωτοκόλλων δικτύου για το διαδίκτυο έχουν μετατρέψει την σύνδεση αισθητήρων σε cloud αλλά και την ενσωμάτωση τους σε αντικείμενα και συσκευές πολύ ευκολότερη με αποτέλεσμα να επιτυγχάνεται αποτελεσματικότερη και ασφαλέστερη μεταφορά δεδομένων

- **πλατφόρμες cloud computing:** η διαθεσιμότητα και προσβασιμότητα σε πλατφόρμες cloud, έχει δώσει σε επιχειρήσεις και στο καταναλωτικό κοινό τη δυνατότητα να έχουν πρόσβαση σε cloud-based υπηρεσίες, εφαρμογές και λογισμικό που μπορούν να αξιοποιήσουν για να διαχειριστούν αποτελεσματικά δραστηριότητες, για τις οποίες ενδεχομένως να μην έχουν οι ίδιοι κατάλληλους πόρους, υποδομή ή τεχνολογία
- **machine learning:** με την αλματώδη πρόοδο στη μηχανική μάθηση και τα analytics καθώς και την ελευθερία πρόσβασης σε τεράστιες ποσότητες δεδομένων, οι επιχειρήσεις μπορούν να συλλέγουν πληροφορίες ευκολότερα και πολύ πιο αποτελεσματικά· η έγερση αυτών των όμορων τεχνολογιών επεκτείνει συνεχώς τα όρια και τις δυνατότητες του IoT και ως συνέπεια αυτού τα δεδομένα που παράγονται από αυτό τροφοδοτούν τις τεχνολογίες αυτές και τις ωθούν να πάνε ένα βήμα παραπέρα
- **τεχνητή νοημοσύνη (AI):** η πρόοδος στα νευρωνικά δίκτυα (neural networks) έφερε την τεχνολογία NLP (natural-language processing) σε IoT συσκευές· οι NLP τεχνολογίες είναι ένας επιστημονικός τομέας που απασχολεί τη γλωσσολογία, την επιστήμη των υπολογιστών και την τεχνητή νοημοσύνη και ασχολείται με τις αλληλεπιδράσεις μεταξύ των υπολογιστικών συστημάτων και της ανθρώπινης γλώσσας

Το γεγονός ότι το IoT δεν είναι αποτέλεσμα μιας πρωτότυπης τεχνολογίας αλλά προσεγγίζεται διεπιστημονικά, αφορά και απασχολεί πληθώρα κλάδων της επιστήμης, σημαίνει ότι πληθώρα εφαρμογών και τεχνικών συμπληρώνουν την υποδομή του IoT προκειμένου να επιτευχθεί η ζεύξη μεταξύ φυσικού και ψηφιακού κόσμου.

### 2.3.γ Εφαρμογές του IoT

Το IoT μετατρέπει τις συμβατικές συσκευές σε τέτοιες που μπορούν να συλλέξουν και να ανταλλάξουν δεδομένα, να συνδεθούν με το διαδίκτυο και να αποστείλουν δεδομένα σε ανθρώπους ή cloud προκειμένου να ενημερώσουν ή να συμβάλλουν στην συλλογή πληροφορίας, να επικοινωνήσουν με άλλες συσκευές είτε με ανθρώπους και τέλος να ενεργοποιηθούν αυτοματοποιημένα βάσει ερεθισμάτων είτε ανθρωπίνων εντολών.

Περιλαμβάνει δισεκατομμύρια ετερογενών έξυπνων συσκευών με μοναδική ταυτότητα τα οποία συλλέγουν δεδομένα, τα επεξεργάζονται και τα επικοινωνούν σε άλλες συσκευές είτε “αντικείμενα” (objects). Εφαρμογές του IoT είτε αποσκοπούν στην συλλογή δεδομένων από δραστηριότητες της καθημερινότητας και την εξέλιξη της μηχανής και του αυτοματισμού είτε στην επεξεργασία των δεδομένων προκειμένου να προκύψουν συμπεράσματα που άλλοτε θα ενεργοποιήσουν τα αντανάκλαστικά του εμπορικού marketing και άλλοτε θα λειτουργήσουν προς όφελος επίτευξης ενός υψηλότερου βιοτικού επιπέδου.

Μερικές από τις πιο σημαντικές εφαρμογές του IoT παρατίθενται παρακάτω:

- **Έξυπνο σπίτι:** Η αποδοτικότερη διαχείριση των πόρων προκειμένου να καλύπτονται οι ανάγκες των παραμενόντων, η ασφαλέστερη διαμονή αυτών καθώς και η όσο το δυνατό πιο ομαλή και ευχάριστη διαβίωση είναι πράγματα τα οποία το έξυπνο σπίτι φροντίζει. Η κατανάλωση ενέργειας περιορίζεται και το σπίτι προσαρμόζεται στις συνήθειες και την καθημερινότητα των ανθρώπων, ο on-demand και εξ αποστάσεως χειρισμός συσκευών για τη μείωση του κόστους και την αποφυγή ατυχημάτων, η ανίχνευση παραβιάσεων ασφαλείας και ο περιορισμός παραβίασης της ιδιωτικότητας είναι μερικές από τις πολλές ιδιότητες που χαρακτηρίζουν το έξυπνο σπίτι.

- **Έξυπνες πόλεις:** Οι πόλεις σε όλο τον κόσμο γίνονται ολοένα και πιο έξυπνες και αποκτούν “συνείδηση”. Από τη διαχείριση των φωτεινών σηματοδοτών έως τους αισθητήρες συλλογής δεδομένων περιβάλλοντος και τη διαχείριση των απορριμμάτων, από την παρακολούθηση της κατάστασης και στατικότητας κτιρίων, δημοσίων έργων και μνημείων έως τον παραμετροποιημένο φωτισμό, από την παρακολούθηση των αστικών οδικών δικτύων και τον εντοπισμό των smartphone προκειμένου να αποφεύγονται κυκλοφοριακές συμφορήσεις, η έξυπνη πόλη προτρέπει όλες τις πόλεις ανά τον κόσμο να προβούν σε αλλαγές και να μετατραπούν σε τέτοιες που άνθρωποι ανά τον κόσμο να επιδιώκουν μια ευκαιρία να γίνουν κάτοικοι αυτής.
- **Έξυπνο περιβάλλον:** Η έγκαιρη ανίχνευση πυρκαγιάς σε δασικές περιοχές μπορεί να σώσει ζωές και τοπικά οικοσυστήματα. Η παρακολούθηση της ποιότητας του αέρα μπορεί να προσδιορίσει περιοχές με μεγαλύτερες συγκεντρώσεις ρυπογόνων αερίων και να οδηγήσει σε μέτρα καταπολέμησης του φαινομένου. Η παρακολούθηση καιρικών φαινομένων οδηγεί στην αποφυγή μοτίβων ανθρώπινης δραστηριότητας που τα εντείνουν. Η έγκαιρη ενημέρωση για ξεσπάσματα της φύσης μπορεί να οδηγήσει σε έγκαιρη λήψη μέτρων και μείωση δυσμενών καταστάσεων. Το περιβάλλον αποτελεί ένα από τα σημαντικότερα στρατηγήματα της ανθρωπότητας που θα κληροδοτηθεί στις επόμενες γενιές και το IoT αποτελεί αρωγό στη διατήρηση της ακεραιότητάς του.
- **Μεταφορές και εφοδιασμός:** Οι έξυπνες μεταφορές χρησιμοποιούν αναδυόμενες τεχνολογίες για να κάνουν τις μετακινήσεις πιο οικονομικές, ασφαλείς και ευχάριστες. Ταυτόχρονα, η αυτόνομη μεταφορά αγαθών βελτιώνει την αποτελεσματικότητα της εφοδιαστικής αλυσίδας. Επίσης δίνεται η δυνατότητα της συνεχούς παρακολούθησης των εμπορευμάτων καθώς και της ποιότητας αυτών. Τα προϊόντα παρακολουθούνται και στην περίπτωση επικίνδυνων φορτίων υπάρχει συνεχής έλεγχος για την διασφάλιση αυτών καθώς και προειδοποιητικές παρεμβάσεις σε περίπτωση που κάποια εκ των παραμέτρων του περιβάλλοντος απειλεί το εμπόρευμα ή το καθιστά επικίνδυνο για το περιβάλλον γύρω του. Καθολικά, το IoT καθιστά τις μεταφορές ασφαλέστερες και καλύτερα διαχειρίσιμες, πολύ πιο αποδοτικές και οικονομικά συμφέρουσες.
- **IoT στην ιατρική επιστήμη:** Η συνεχής παρακολούθηση, η πρόληψη, ο τακτικός έλεγχος και η άμεση επέμβαση είναι οι βασικοί παράγοντες για την διατήρηση μιας καλής υγείας. Από τη συνεχή συλλογή δεδομένων εν δυνάμει ασθενών και παθόντων έως την αποθήκευση, επεξεργασία και ανάλυση αυτών μέσω cloud-based υπηρεσιών και τη δυνατότητα άμεσης ενημέρωσης για ταχεία επέμβαση και αποφυγή επιδείνωσης ατυχών περιστατικών, το IoT έχει μερικώς συμβάλει στην αποτελεσματικότερη παρακολούθηση και καταπολέμηση περιστατικών που μπορούν να διορθωθούν όταν η επέμβαση είναι άμεση και υπόσχεται να φέρει την επανάσταση στην ιατρική παρακολούθηση. Η ανθρώπινη αλληλεπίδραση καθίσταται πιο αποτελεσματική αφού επιτρέπεται ο συνεχής εντοπισμός και παρακολούθηση των ασθενών. Το IoT προσφέρει υπηρεσίες για την βοήθεια ηλικιωμένων και ατόμων με ειδικές ανάγκες, υπηρεσίες για τη συνεχή επιτήρηση ασθενών και αθλητών και συνεχή παρακολούθηση των συνθηκών αποθήκευσης φαρμάκων και εμβολίων για αποφυγή αλλοίωσης ή καταστροφής τους.

Στην πραγματικότητα οι δυνατότητες είναι πραγματικά σχεδόν απεριόριστες και περιορίζονται μόνο από την περιστασιακή ματαιοδοξία του ανθρώπινου νου.

## ΚΕΦΑΛΑΙΟ 3 Αρχιτεκτονική υπολογιστικής ομίχλης

---

Στην αναφορά της αρχιτεκτονικής OpenFog γίνεται αναφορά στους 8 βασικούς πυλώνες που ορίζουν την υπολογιστική ομίχλη. Μεταξύ άλλων, προκειμένου να υπάρχει μια ομαλότητα και ομοιομορφία όσον αφορά το λογισμικό, την τεχνολογία και την ορολογία γύρω από την υπολογιστική ομίχλη, έχουν οριστεί συγκεκριμένα πρότυπα που προορίζονται ως αρωγοί στην υποστήριξη της συνεργασίας μεταξύ διαφορετικών παρόχων καθώς και μεταξύ των ανομοιογενών τεχνολογιών και λογισμικών που παρέχουν. Μερικές από τις κοινόχρηστες αναφορές που ορίζουν τα προαναφερθέντα πρότυπα είναι οι εξής:

- **περιγραφή της αρχιτεκτονικής OpenFog:** πρόκειται για μια συγκεκριμενοποιημένη και ταυτόχρονα αφηρημένη αναπαράσταση στιγμιότυπων ενός ή περισσότερων κόμβων υπολογιστικής ομίχλης
- **οπτική της αρχιτεκτονικής OpenFog:** η οπτική αναφέρεται σε τρόπους εξέτασης μιας ή περισσότερων διαφορετικών απόψεων παρακολούθησης ενός συστήματος υπολογιστικής ομίχλης
- **άποψη της αρχιτεκτονικής OpenFog:** ως άποψη ορίζεται η αναπαράσταση μιας ή περισσότερων δομικών πτυχών ενός συστήματος αρχιτεκτονικής υπολογιστικής ομίχλης
- **προοπτική της αρχιτεκτονικής OpenFog:** ως προοπτική ορίζεται μια κάθετη εξέταση της αρχιτεκτονικής

Η εφαρμογή των στοιχείων της αρχιτεκτονικής και των διάφορων οπτικών προσέγγισης του εκάστοτε σεναρίου προκειμένου αυτό να ανταπεξέρχεται δυναμικά στις ήδη υπάρχουσες ανάγκες και απαιτήσεις αλλά και σε αυτές που ενδέχεται να προκύψουν είναι κομμάτι της λειτουργικής οπτικής συστημάτων υπολογιστικής ομίχλης όπως ορίζεται από την αρχιτεκτονική αναφοράς OpenFog.

Στον αντίποδα αυτού, ο τρόπος με τον οποίο αναπτύσσεται το εκάστοτε σύστημα υπολογιστικής ομίχλης καθώς και η ευελιξία του λογισμικού προκειμένου να επιτρέπει πληθώρα παραμετροποιήσεων είναι εξέχουσας σημασίας και αφορά το κομμάτι της υλοποίησης. Υπάρχει ένας μεγάλος βαθμός ευελιξίας τόσο στην τεχνολογία όσο και στο λογισμικό κατά την εξέλιξη της εκάστοτε υλοποίησης χωρίς παρόλα αυτά να υπάρχει ασάφεια όσον αφορά τη διαφάνεια συγκεκριμένων προκαθορισμένων αρχιτεκτονικών στοιχείων υλικού και λογισμικού. Παρά το γεγονός ότι μερικά στοιχεία θα είναι άλλοτε ελάσσονος και άλλοτε εξέχουσας σημασίας βάσει των αναγκών του συστήματος καθώς και της τεχνολογίας που χρησιμοποιείται στην εκάστοτε υλοποίηση, τα αρχιτεκτονικά στοιχεία παραμένουν ίδια.

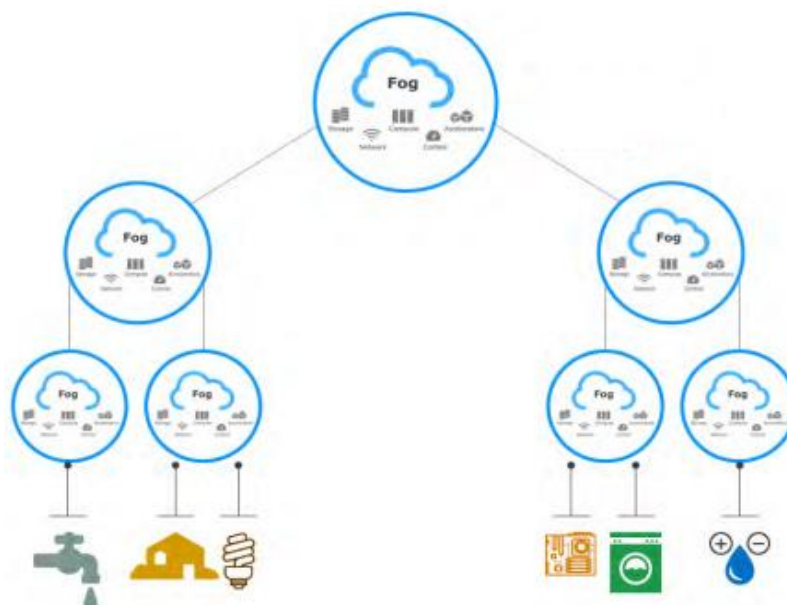
### 3.1 Αρχιτεκτονική Fog N-κόμβων

---

Στις περισσότερες υλοποιήσεις συστημάτων υπολογιστικής ομίχλης υπάρχουν ένα ή παραπάνω επίπεδα κόμβων. Το σύνολο των επιπέδων διαφέρει από σύστημα σε σύστημα βάσει των αναγκών του και υπάρχει μια σειρά από παράγοντες για τον καθορισμό του πλήθους των επιπέδων κόμβων υπολογιστικής ομίχλης. Συγκεκριμένα, υπάρχει ένας διαχωρισμός μεταξύ κόμβων υπολογιστικής ομίχλης βασιζόμενος στην λειτουργία που έχουν οριστεί να εκτελούν.

Για να γίνει πιο κατανοητό, θα αναλύσουμε αναφορικά ένα τυχαίο σενάριο που ενσωματώνει τεχνολογίες υπολογιστικής ομίχλης και θα δούμε πως θα μπορούσε να γίνει ο διαχωρισμός μεταξύ διαφορετικών επιπέδων κόμβων υπολογιστικής ομίχλης. Έστω λοιπόν ότι:

1. Στο κατώτερο επίπεδο υπάρχουν κόμβοι υπολογιστικής ομίχλης και πιο συγκεκριμένα συσκευές που είναι υπεύθυνες για τη συλλογή δεδομένων από τους στρατηγικά τοποθετημένους αισθητήρες στο περιβάλλον υλοποίησης, την κανονικοποίηση των δεδομένων και στην αποστολή εντολών στις έξυπνες συσκευές, στα αντικείμενα και στους ενεργοποιητές συστήματος προκειμένου να διατηρείται μια ομαλή και αποδοτική συνέχεια στην υλοποίηση του συστήματος.
2. Στο επίπεδο που βρίσκεται ακριβώς πάνω από το κατώτερο επίπεδο κόμβων υπολογιστικής ομίχλης βρίσκονται κόμβοι οι οποίοι είναι υπεύθυνοι για την διήθηση των δεδομένων, τη συμπίεση των σημαντικών πληροφοριών σε συστοιχίες δεδομένων τα οποία έπεται να μεταφερθούν σε ένα είτε παραπάνω από ένα επίπεδα πάνω από το παρόν επίπεδο κόμβων. Σε ορισμένες περιπτώσεις θα μπορούσαν να παράγουν και στατιστικά είτε αναλυτικά στοιχεία για την εν δυνάμει επεξεργασία σε πραγματικό χρόνο ή το λιγότερο σε χρόνο τέτοιο που είναι πολύ μικρότερος από ότι θα ήταν στην περίπτωση επεξεργασίας σε ένα κάποιο απομακρυσμένο υπολογιστικό νέφος.
3. Στο επίπεδο ή στα επίπεδα που βρίσκονται στα όρια του τοπικού δικτύου και πιο κοντά στο υπολογιστικό νέφος συμβαίνει ως επί το πλείστον η συγκέντρωση σημαντικής πληροφορίας και η επεξεργασία αυτής για την μετατροπή αυτής σε αναλυτικά στοιχεία. Σε επίπεδα που βρίσκονται πιο μακριά από τις τελικές συσκευές είναι δυνατή η αποδοτικότερη παραγωγή σημαίνουσας πληροφορίας καθώς ο όγκος δεδομένων αφορά περισσότερα κομμάτια της υλοποίησης και η γνώση που παράγεται αφορά το σύνολο των ενεργειών της εκάστοτε υλοποίησης. Όσο πιο απομακρυσμένο είναι το επίπεδο και οι κόμβοι που το συντελούν από την πηγή της πληροφορίας, τόσο πιο σημαίνουσα και ασφαλής είναι η πληροφορία που παράγεται.



Εικόνα 3.1: Αρχιτεκτονική Fog N-κόμβων [1]

Στο παραπάνω παράδειγμα έγινε αναφορά στο κατώτατο επίπεδο κόμβων υπολογιστικής ομίχλης το οποίο είναι το πλησιέστερο στις τελικές συσκευές, αισθητήρες και συστήματα ενεργοποιητών, χωρίς αυτό να σημαίνει παρόλα αυτά ότι δεν υπάρχει πιθανότητα ενσωμάτωσης ενός ανώτερου συστήματος αισθητήρων είτε ενεργοποιητών σε ένα η περισσότερα ανώτερα επίπεδα, όπου η καθαρή γνώση από τα κατώτερα επίπεδα θα συνδυάζεται με τα δεδομένα που συλλέγονται στο εκάστοτε ανώτερο επίπεδο. Έπειτα ο αριθμός των επιπέδων που ακολουθούν αυτό το κατώτερο επίπεδο είναι ασαφής και εξαρτάται από τις απαιτήσεις και το εύρος του τοπικού δικτύου. Η κλιμάκωση των επιπέδων έγκειται στο σενάριο που αντιμετωπίζει η εκάστοτε υλοποίηση υπολογιστικής ομίχλης. Μερικές από τις απαιτήσεις του σεναρίου που καθορίζουν αφανώς το πλήθος των επιπέδων υπολογιστικών κόμβων είναι:

- η ποσότητα και το είδος εργασίας που απαιτείται για το κάθε επίπεδο
- ο αριθμός των αισθητήρων και το εύρος του δικτύου ενεργοποιητών
- οι δυνατότητες βάσει υλικού και λογισμικού των κόμβων υπολογιστικής ομίχλης στο κάθε επίπεδο
- ο λανθάνων χρόνος κατά την επικοινωνία μεταξύ διαφορετικών κόμβων στο ίδιο είτε σε όμορα επίπεδα καθώς και ο λανθάνων χρόνος κατά την επικοινωνία μεταξύ των αισθητήρων και του ή των ενεργοποιητών στο κάθε επίπεδο
- η αξιοπιστία και η συνέχεια είτε περιοδικότητα διαθεσιμότητας των κόμβων υπολογιστικής ομίχλης βάσει του φόρτου εργασίας είτε της ίδιας της εργασίας που εκτελούν

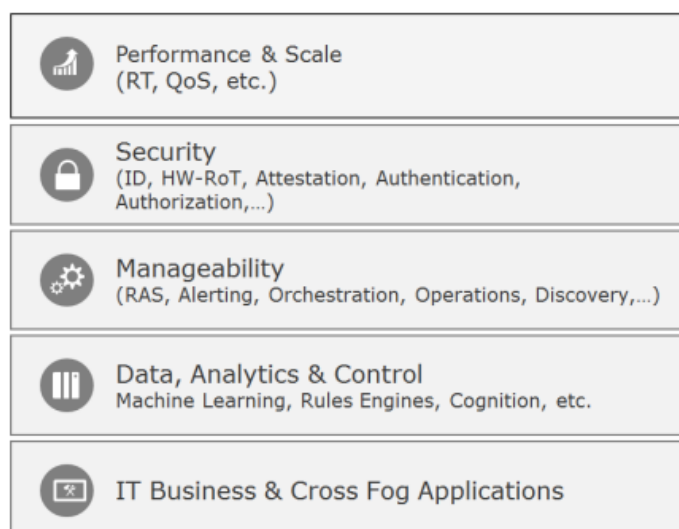
Η ιδέα γύρω από την αρχιτεκτονική Fog N-κόμβων είναι ότι πέραν του κατώτερου επιπέδου όπου και αρχίζει να σχηματίζεται η πληροφορία, κάθε επόμενο επίπεδο κόμβων εξάγει σημαίνοντα δεδομένα και δημιουργεί μια μορφή νοημοσύνης στην εξαγωγή αναλυτικών στοιχείων, προβλέψεων και βέλτιστων λύσεων, η οποία οξύνεται σε κάθε επόμενο επίπεδο. Τα διάφορα επίπεδα κόμβων εξυπηρετούν τον αποτελεσματικότερο διαχωρισμό και την αποτελεσματικότερη διαχείριση της πληροφορίας προκειμένου να παρέχεται η καλύτερη δυνατή λειτουργική διαχείριση και η αποτελεσματικότερη διαχείριση συστήματος.

Παραπάνω αναφερθήκαμε στο γεγονός ότι πρέπει να υπάρχει ευελιξία στο λογισμικό που εκτελείται στο κάθε επίπεδο και είναι σημαντική η διαπίστωση ότι λόγω διαφορετικού υλικού στο κάθε επίπεδο θα υπάρχει και ανομοιογένεια λογισμικού σε όλο το εύρος του συστήματος υπολογιστικής ομίχλης. Είναι σημαντικό να υπάρχει η δυνατότητα επιτυχούς μετάφρασης και μεταφοράς των δεδομένων και ενδεχόμενων λειτουργιών που είναι απαραίτητες για την αποσυμπίεση και εκ νέου διαμόρφωση της πληροφορίας στο ανώτερο επίπεδο κόμβων όπως και εξίσου σημαντική είναι η δυνατότητα του λογισμικού να παραμετροποιείται και να αλλάζει προκειμένου να ανταποκρίνεται στις παρούσες ανάγκες αλλά και στις πιθανώς επερχόμενες. Προκειμένου όμως να καθίσταται δυνατή η συνεχής και απρόσκοπτη μετανάστευση λογισμικού από και προς κόμβους του ίδιου επιπέδου είτε όμορων επιπέδων χωρίς να επιβαρύνεται η φυσική συνέχεια ροής πληροφορίας και δημιουργίας λειτουργικής ευφυΐας, πρέπει να υπάρχει και αντίστοιχη ευελιξία αλλά και δόση διαχρονικότητας στο υλικό που φιλοξενεί το εκάστοτε λογισμικό προκειμένου να μπορεί να υποστηρίξει αλλαγές και προσθήκες.

## 3.2 Περιγραφή της αρχιτεκτονικής OpenFog

Παραπάνω αναλύσαμε μερικά από τα χαρακτηριστικά της υπολογιστικής ομίχλης, με βάση τα οποία δίνεται η ευκαιρία για την επίλυση πολλών προβλημάτων σε σενάρια της καθημερινότητας καθώς και επισημαίνεται η χρηστικότητα εφαρμογών υπολογιστικής ομίχλης σε ήδη υπάρχοντα συστήματα. Ο παράγοντας της καθυστέρησης εκπαραθυρώνεται και η αξιοπιστία των συστημάτων είναι συγκριτικά σαφώς μεγαλύτερη από ότι αυτή που προσφέρουν υπολογιστικά νέφη, από τα οποία πλέον δεν εξαρτώνται τα εν λόγω συστήματα. Στον αντίποδα παράθεσης των χαρακτηριστικών που προσφέρει η ενσωμάτωση τεχνολογιών υπολογιστικής ομίχλης έγινε μια περιγραφή των απαιτήσεων για την αρχική υλοποίηση του συστήματος καθώς και για τη σημαντικότητα της επικαιρότητας των αρχιτεκτονικών στοιχείων προκειμένου να είναι σε θέση να ανταπεξέλθουν στις ανάγκες του συστήματος αλλά και να είναι παραπάνω από ικανά και ευέλικτα προκειμένου να καθίστανται ανθεκτικά στο χρόνο. Τέλος αναφερθήκαμε στο γεγονός ότι οι υλοποιήσεις υπολογιστικής ομίχλης ενέχουν τη βελτίωση διάφορων πτυχών του συστήματος και πέρα από τη δημιουργία της, την δυναμικά αυξανόμενη ευφυία συστήματος από την πηγή παραγωγής των ακατέργαστων δεδομένων προς τα ανώτερα επίπεδα κόμβων υπολογιστικής ομίχλης.

Ήρθε η ώρα να προχωρήσουμε στην περιγραφή των απαιτήσεων των εμπλεκόμενων μερών στην ανάπτυξη των αρχιτεκτονικών στοιχείων αλλά και στοιχείων συστήματος, όπως στην κατασκευή κυκλωμάτων επεξεργασίας δεδομένων, στη ανάπτυξη και ενσωμάτωση του συστήματος, στην ανάπτυξη λογισμικού και στον προγραμματισμό των εφαρμογών. Στόχος είναι η ανάπτυξη μιας αρχιτεκτονικής που θα παρέχει δυνατότητα ανάπτυξης διαλειτουργικών οικοσυστημάτων υπολογιστικής ομίχλης μέσω της ευθυγράμμισης των ετερογενών και ανόμοιων χαρακτηριστικών και αρχιτεκτονικών στοιχείων της υπολογιστικής ομίχλης και ταυτόχρονα μέσω της εξομίωσης εν δυνάμει αποκλινουσών εργασιών σε μια κοινή καθομιλουμένη ορολογία. Παρακάτω και προτού προχωρήσουμε στην σε λεπτομέρειες κατώτερου επιπέδου, θα δούμε μια περιγραφή της σύνθετης αρχιτεκτονικής OpenFog:



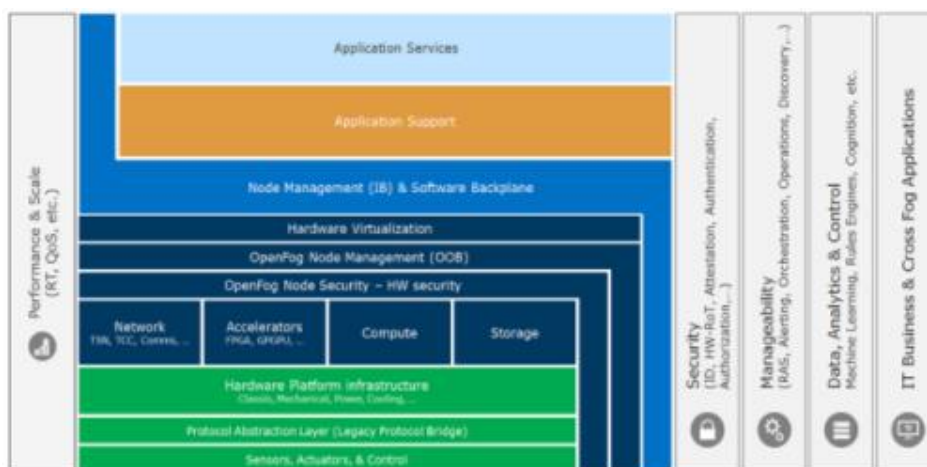
Εικόνα 3.2: Περιγραφή αρχιτεκτονικής Fog computing [1]



Η αφηρημένη αρχιτεκτονική του OpenFog περιλαμβάνει προοπτικές όπως:

- **Απόδοση (Performance):** Υπάρχει μια ευρεία σειρά από διαφορετικούς λόγους για τους οποίους η υπολογιστική ομίχλη αποτελεί τη σύγχρονη λύση στην απελευθέρωση τοπικών δικτύων IoT από αποκεντρωμένα υπολογιστικά νέφη και στην και είναι απόλυτα λογικό να υπάρχει πληθώρα απαιτήσεων και πολλά ζητήματα σχεδιασμού τα οποία φροντίζουν να διασφαλιστεί η εκπλήρωση αυτών και η διασφάλιση ικανοποίησης του ή των τελικών χρηστών.
- **Ασφάλεια (Security):** Προκειμένου να διασφαλίζεται η επιτυχία υλοποιήσεων υπολογιστικής ομίχλης, η ασφάλεια από και προς μια οποιαδήποτε συσκευή είτε κόμβο αποτελεί έναν καθοριστικό παράγοντα και ως εκ τούτου είναι επιτακτική από το κατώτερο επίπεδο του υλικού έως και το λογισμικό.
- **Διαχειρισιμότητα (Manageability):** Η αποδοτική διαχείριση των δεδομένων σε σενάρια υλοποιήσεων υπολογιστικής ομίχλης είναι μια κρίσιμη πτυχή που κλιμακώνεται σε όλα τα επίπεδα μιας ιεραρχίας υπολογιστικής ομίχλης προκειμένου να δύναται να αποδίδει απρόσκοπτα και άνευ καθυστερήσεων σαν σύνολο.
- **Ανάλυση δεδομένων και έλεγχος (Data Analytics and Control):** Οι κόμβοι υπολογιστικής ομίχλης, ανεξαρτήτως επιπέδου ιεραρχίας πρέπει να χαρακτηρίζονται από αυτονομία στην αξιόπιστη εξαγωγή αναλυτικών στοιχείων και καθαρών δεδομένων και αυτό απαιτεί την τοπική επεξεργασία και ανάλυση δεδομένων σε συνδυασμό με συνεχή έλεγχο του συνόλου ενεργοποιητών που τους έχει ανατεθεί βάσει ιεραρχίας και απαιτήσεων του εκάστοτε σεναρίου.
- **Πληροφοριακά Συστήματα Επιχειρήσεων και platform-independent εφαρμογές υπολογιστικής ομίχλης (IT Business and Cross Fog Applications):** Σε ένα οικοσύστημα υπολογιστικής ομίχλης όπου υπάρχουν πολλαπλοί προμηθευτές ετερογενούς υλικού και ανομοιογενούς λογισμικού χρήσης, επιτακτική είναι η δυνατότητα παραμετροποιημένης μετεγκατάστασης και σωστής λειτουργίας ανεξαρτήτως συσκευής και ιεραρχίας αυτής σε περίπτωση όπου πρόκειται για κόμβο υπολογιστικής ομίχλης.

Η αρχιτεκτονική αναφοράς OpenFog αποτελεί έναν συνδυασμό προοπτικών όπως τις αναφέραμε παραπάνω και μιας πληθώρας ανομοιογενών οπτικών των ενδιαφερόμενων μερών οι οποίες χρησιμοποιούνται δυναμικά και ανεξάρτητα για την κάλυψη των αναγκών ενός σεναρίου ομίχλης. Οι τρεις οπτικές (views) που ορίζει η αρχιτεκτονική είναι οι εξής:

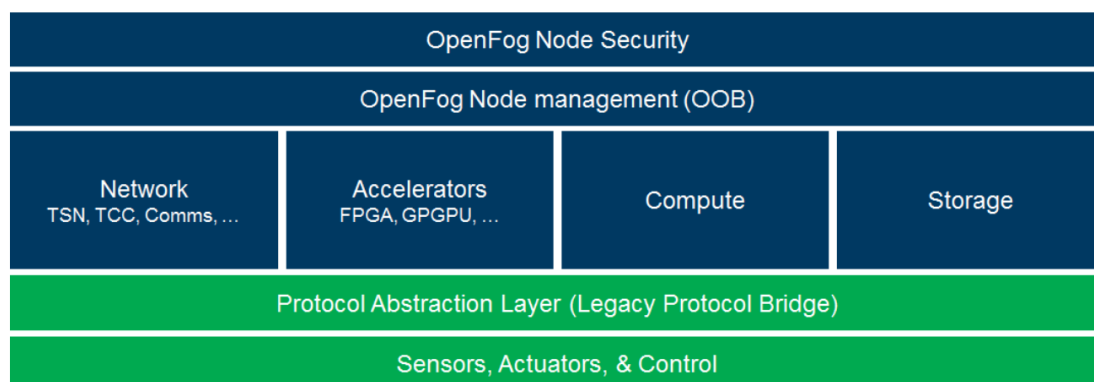


Εικόνα 3.3: Οπτικές αρχιτεκτονικής ομίχλης [1]

- **Λογισμικό (Software view):** Η οπτική λογισμικού αναπαρίσταται στα τρία κορυφαία επίπεδα της αρχιτεκτονικής OpenFog του παραπάνω σχήματος και περιλαμβάνει τις υπηρεσίες εφαρμογών (Application Services), την υποστήριξη εφαρμογών (Application Support) και τη διαχείριση κόμβων και **παράλληλου λογισμικού (Node Management (NB) & Software Backplane)**. **Κατά την ανασκόπηση του paper να διευκρινιστεί εάν αυτή είναι η σωστή μετάφραση-ορισμός του Software Backplane**
- **Σύστημα (System view):** Η οπτική συστήματος αναπαρίσταται στα μεσαία στρώματα της αρχιτεκτονικής OpenFog του παραπάνω σχήματος και περιλαμβάνει την εικονικοποίηση του υλικού (Hardware Virtualization), τη διαχείριση κόμβων OpenFog (OpenFog Node Management), την ασφάλεια κόμβων-υλικού OpenFog (OpenFog Node Security – Hardware Security), τις διάφορες λειτουργίες που αναλαμβάνουν να φέρουν εις πέρας οι κόμβοι υπολογιστικής ομίχλης όπως την δικτύωση και επικοινωνία μεταξύ δικτύων με ενσωμάτωση πρωτοκόλλων δικτύου (Network), η ενσωμάτωση υλικού υπολογιστικής φύσεως για την αυξημένη απόδοση ορισμένων λειτουργιών σε αντίθεση με την ανάθεση όλου του εύρους λειτουργιών σε μια κεντρική μονάδα επεξεργασίας γενικής χρήσης (Accelerators), η επεξεργασία των δεδομένων (Compute) και η αποθήκευση δεδομένων (Storage), και τέλος την υποδομή της πλατφόρμας υλικού (Hardware Platform Infrastructure).
- **Κόμβος (Node view):** Η οπτική του κόμβου αναπαρίσταται στα δύο κατώτερα στρώματα της αρχιτεκτονικής OpenFog του παραπάνω σχήματος και περιλαμβάνει το επίπεδο διαχείρισης πρωτοκόλλων (Protocol Abstraction Layer) και το σύνολο των αισθητήρων, ενεργοποιητών και τον έλεγχο αυτών (Sensors, Actuators & Control).

### 3.2.α Οπτική κόμβου αρχιτεκτονικής αναφοράς OpenFog

Στην περιγραφή της αρχιτεκτονικής OpenFog, η βασική μονάδα της υπολογιστικής ομίχλης είναι ο κόμβος και αποτελεί την οπτική χαμηλότερου επιπέδου. Τα ενδιαφερόμενα μέρη που εμπλέκονται στη διαμόρφωση της οπτικής κόμβου είναι μεταξύ άλλων οι κατασκευαστές αρχιτεκτονικών υλικών όπως τσιπ, οι υπεύθυνοι για την αρχιτεκτονική υλικού και λογισμικού και οι υπεύθυνοι για την αρχιτεκτονική και τις δυνατότητες του συστήματος. Προτού αναλύσουμε τα δύο κατώτερα στρώματα της αρχιτεκτονικής OpenFog που αποτελούν την οπτική του κόμβου, θα αναλύσουμε τις διάφορες λειτουργίες που συμβαίνουν στον κόμβο καθώς και θέματα διαχείρισης και ασφάλειας σε επίπεδο κόμβου:



Εικόνα 3.4: Οπτική κόμβου και λειτουργιών που απασχολούν το επίπεδο κόμβου [1]

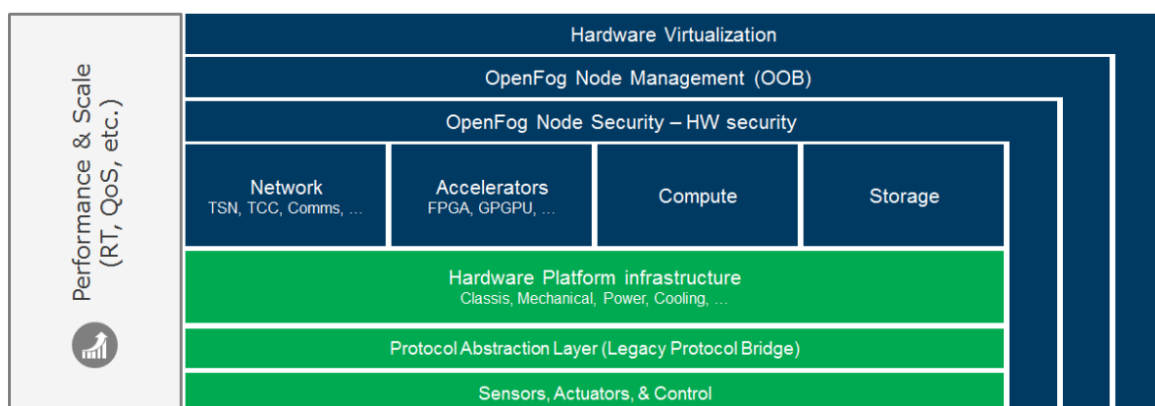
- **Ασφάλεια κόμβου:** Η καθολική ασφάλεια ενός συστήματος που ενσωματώνει τεχνολογίες υπολογιστικής ομίχλης εξαρτάται από την ασφάλεια σε κάθε επίπεδο ιεραρχίας των κόμβων. Η σύνθετη λειτουργία που αναλαμβάνει να φέρει εις πέρας ένας κόμβος υπολογιστικής ομίχλης σημαίνει ότι υπάρχουν διαφορετικές λειτουργίες και εφαρμογές στις οποίες πρέπει να υπάρχει προστασία αλλά και διαφορετικό υλικό πέρα από το λογισμικό το οποίο πρέπει αντίστοιχα να προστατεύεται κατάλληλα προκειμένου να επιτυγχάνεται η συνολική ασφάλεια του συστήματος. Η ασφάλεια είναι επιτακτικό να λαμβάνεται υπόψιν και να ενσωματώνεται τόσο σε υλικό και λογισμικό, όσο και σε όλα τα ενδιάμεσα επίπεδα μεταξύ αυτών
- **Διαχείριση κόμβων:** Παρά το γεγονός ότι σε υλοποιήσεις συστημάτων υπολογιστικής ομίχλης υπάρχει ένα τοπικό δίκτυο το οποίο ενσωματώνει κόμβους υπολογιστικής ομίχλης, η φύση υλοποίησης του εκάστοτε σεναρίου κλιμακώνεται βάσει των αναγκών αυτού, το πλήθος των κόμβων σε κάθε επίπεδο ιεραρχία κόμβων μπορεί να διαφέρει και η απόσταση του κάθε κόμβου υπολογιστικής ομίχλης διαφέρει βάσει επιπέδου ιεραρχίας. Κάθε κόμβος πρέπει να υποστηρίζει διεπαφές διαχείρισης προκειμένου οι υπεύθυνοι διαχείρισης συστήματος υψηλότερου επιπέδου ιεραρχίας να μπορούν να παρακολουθούν, να παραμετροποιούν και να ελέγχουν τους κόμβους χαμηλότερου επιπέδου.
- **Δίκτυο:** Μια από τις κύριες λειτουργίες ενός κόμβου υπολογιστικής ομίχλης είναι η διύλιση πληροφοριών για την εξαγωγή καθαρής πληροφορίας και αναλυτικών στοιχείων. Για την επίτευξη ομαλούς λειτουργίας του συστήματος αυτές οι πληροφορίες πρέπει να μπορούν να προωθούνται από και προς κόμβους ίδιου είτε διαφορετικού επιπέδου ιεραρχία κόμβων. Μέσω του δικτύου επιτυγχάνεται η συνεχής επικοινωνία μεταξύ κόμβων και λόγω της ιδιότητας μερικών εφαρμογών να είναι ευαίσθητες στο χρόνο, είναι επιτακτικής σημασίας η ελάττωση της καθυστέρησης και σε ορισμένες περιπτώσεις ενδέχεται να χρειαστεί να υποστηρίζεται και το σύνολο προτύπων Time-Sensitive Networking της IEEE.
- **Επιταχυντές:** Η φύση των κόμβων υπολογιστικής ομίχλης επιτάσσει την χρήση αποκλειστικού υλικού για να ικανοποιήσει περιορισμούς καθυστέρησης και ισχύος.
- **Υπολογιστική επεξεργασία:** Η λειτουργία που ανατίθεται σε κάθε κόμβο υπολογιστικής ομίχλης, το αρχιτεκτονικό υλικό που τον συντελεί και το λογισμικό που υποστηρίζει διαφέρουν από τους υπόλοιπους. Προκειμένου να μπορεί να εκτελέσει μια σειρά ανομοιογενών λειτουργιών πρέπει να διαθέτει υπολογιστικές δυνατότητες γενικού σκοπού καθώς και λογισμικό το οποίο να επιτρέπει τη διαλειτουργικότητα μεταξύ κόμβων υπολογιστικής ομίχλης του ίδιου είτε διαφορετικού επιπέδου.
- **Αποθήκευση:** Η διεξαγωγή έρευνας για την δημιουργία πληροφορίας και αναλυτικών στοιχείων αποτελεί εργασία που απαιτεί τη συλλογή ενός σεβαστού πλήθους πληροφοριών οι οποίες πρέπει πρώτα να αποθηκευτούν προκειμένου να επεξεργαστούν. Έτσι, στους κόμβους υπολογιστικής ομίχλης ενσωματώνονται συσκευές αποθήκευσης οι οποίες με τη σειρά τους πρέπει να πληρούν τις αντίστοιχες απαιτήσεις απόδοσης, αξιοπιστίας και ακεραιότητας δεδομένων προκειμένου να καθίσταται δυνατή η ομαλή και ακέραια λειτουργία του συνολικού συστήματος. Πέρα από την πληροφορία που θα προωθηθεί σε υψηλότερα επίπεδα ιεραρχίας για δημιουργία ευφυίας συστήματος, υπάρχει πλήθος πληροφοριών που πρέπει να αποθηκεύεται τοπικά πράγμα το οποίο σημαίνει ότι υπάρχει ανάγκη τοπικής και αυτόνομης αποθήκευσης για την εξυπηρέτηση εφαρμογών που εκτελούνται στον κόμβο. Πρέπει να σημειωθεί ότι υπάρχει ανομοιογένεια υλικού και στις συσκευές αποθήκευσης.

Αφού εξαντλήσαμε μερικά από τα μεσαία στρώματα της αρχιτεκτονικής OpenFog τα οποία είναι άμεσα προσκείμενα στον κόμβο υπολογιστικής ομίχλης και αφορούν λιγότερο τις λειτουργίες συστήματος, ήρθε η ώρα να αναλύσουμε τα δύο κατώτερα επίπεδα της αρχιτεκτονικής που ανήκουν στη οπτική του κόμβου:

- **Αισθητήρες, ενεργοποιητές και έλεγχος:** Στο IoT οι συσκευές αυτές θεωρούνται κατώτερα στοιχεία και ενδέχεται να υπάρχουν εκατοντάδες τέτοιες συσκευές οι οποίες να σχετίζονται με έναν μόνο κόμβο υπολογιστικής ομίχλης και αυτό πάντα έγκειται στην υλοποίηση και τις ανάγκες του εκάστοτε σεναρίου. Αυτές οι συσκευές-αντικείμενα ενδέχεται να είναι έξυπνα και να μπορούν να εκτελέσουν βασικές λειτουργίες υπολογιστικής ομίχλης και σε άλλες περιπτώσεις μπορεί να μη διαθέτουν την ικανότητα εκτέλεσης βασικών λειτουργιών και επεξεργασίας αλλά στο σύνολό τους έχουν οι περισσότερες από αυτές κάποιο βαθμό συνδεσιμότητας και ενσωματώνουν ενσύρματα είτε ασύρματα πρωτόκολλα επικοινωνίας.
- **Επίπεδο Πρωτοκόλλου:** Η ανομοιογένεια των παραπάνω έξυπνων είτε χαζών αντικειμένων σημαίνει ότι οι υπηρεσίες που εκτελούν και τα δεδομένα που παράγουν διαφέρουν. Σε συνάρτηση με το γεγονός ότι πολλά από αυτά δεν μπορούν να διασυνδεθούν με τον κόμβο υπολογιστικής ομίχλης, δημιουργήθηκαν πρωτόκολλα που δίνουν τη δυνατότητα στα αντικείμενα να μπορούν να επικοινωνήσουν με τους κόμβους υπολογιστικής ομίχλης προκειμένου οι πληροφορίες που συλλέγονται και στη συνέχεια προωθούνται στον κόμβο να μπορούν να μεταφραστούν και να γίνουν κατανοητά από τον κόμβο. Από τη στιγμή που υπάρχει ομοιογένεια στα δεδομένα που συλλέγονται στον κόμβο υπολογιστικής ομίχλης, αυτά με τη σειρά τους μπορούν να χρησιμοποιηθούν για ανάλυση και λειτουργίες συστήματος και λογισμικού υψηλότερου επιπέδου.

### 3.2.β Οπτική συστήματος αρχιτεκτονικής αναφοράς OpenFog

Η οπτική συστήματος αποτελείται από έναν ή περισσότερους κόμβους σε συνδυασμό με άλλα αρχιτεκτονικά στοιχεία και στοιχεία λογισμικού για τη υλοποίηση ενός συστήματος υπολογιστικής ομίχλης και αφορά τις ανησυχίες των ενδιαφερόμενων μερών για τη δημιουργία συστημάτων προκειμένου να καθίσταται ευκολότερη η ανάπτυξη συστημάτων υπολογιστικής ομίχλης. Παρακάτω παρατίθεται η οπτική συστήματος ενός κόμβου χωρίς αυτό να σημαίνει ότι δεν υποστηρίζεται η έννοια πολλαπλών κόμβων για την κάλυψη πλεοναζόντων αναγκών συστήματος είτε για την ικανοποίηση απαιτήσεων ανάπτυξης.



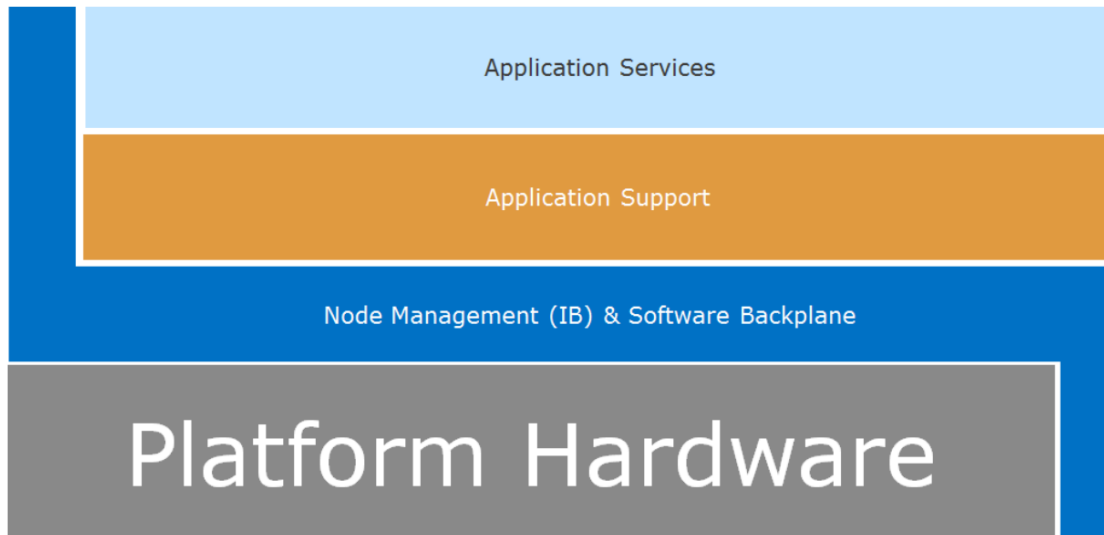
Εικόνα 3.5: Σύνθετη αναπαράσταση της οπτικής συστήματος ενός κόμβου που είναι ενσωματωμένος σε ένα σύστημα υπολογιστικής ομίχλης [1]

Τα στρώματα της οπτικής συστήματος που δεν αναλύσαμε κατά την ανάπτυξη της οπτικής κόμβου είναι:

- **Υποδομή πλατφόρμας υλικού (Hardware Platform Infrastructure):** Η δυνατότητα υλοποίησης συστημάτων τα οποία παρέχουν αξιόπιστα και ανθεκτικά μηχανικά μέρη καθώς και προστασία αρχιτεκτονικών στοιχείων και στοιχείων λογισμικού ξεκινά από τα στοιχεία υλικού που θα επιλεγούν για την υλοποίηση στο επίπεδο κόμβου και επεκτείνεται στα αρχιτεκτονικά στοιχεία της προβολής συστήματος. Μερικά σενάρια εφαρμογών της υπολογιστικής ομίχλης αναφέρθηκαν σε προηγούμενο κεφάλαιο και έχουν ορίσει πληθώρα συσκευών οι οποίες είναι προσκείμενες στους κόμβους υπολογιστικής ομίχλης είτε εαυτούς του κόμβους ως μέρη ενός συστήματος το οποίο είναι ευθέως εκτεθειμένο σε περιβάλλοντα και στις συνθήκες που επικρατούν σε αυτά. Οι παράμετροι του περιβάλλοντος που καλούνται να παρακολουθούν και από τις οποίες καλούνται να εξάγουν δεδομένα μπορεί ενίοτε να αποτελέσουν πρόβλημα είτε να δημιουργήσουν προβλήματα σε στοιχεία υλικού υλοποιήσεων υπολογιστικής ομίχλης. Καθώς το περιβάλλον στο οποίο εκτείνεται το εκάστοτε σενάριο υπολογιστικής ομίχλης και οι συνθήκες αυτού διαφέρουν, υπάρχουν συγκεκριμένες απαιτήσεις για τη δημιουργία μιας προστατευμένης υλοποίησης. Μερικές εξ αυτών πέρα από την συμμόρφωση με τους τοπικούς κανονισμούς και πρακτικές είναι η προστασία από περιβαλλοντικούς παράγοντες, η προστασία από πράξεις βανδαλισμού, η ενσωμάτωση στοιχείων λειτουργικής ασφάλειας για την προστασία του ανθρώπινου δυναμικού καθώς και του υλικού υλοποίησης, η αποτελεσματική διαχείριση κατανάλωσης ενέργειας και η διασφάλιση βέλτιστων συνθηκών λειτουργίας συστήματος, η υποστήριξη διαλειτουργικότητας και η υιοθέτηση αποδεκτής αισθητικής και διαφόρων παραγόντων.
- **Εικονικοποίηση υλικού:** Η εικονικοποίηση υλικού πραγματοποιείται από διαφορετικούς μηχανισμούς εικονικοποίησης βασιζόμενους στο υλικό οι οποίοι είναι διαθέσιμοι σχεδόν στα περισσότερα στοιχεία υλικού μιας υλοποίησης. Η διαδικασία της εικονικοποίησης παίζει σημαντικό ρόλο στην ασφάλεια του συστήματος και επιτρέπει σε πολλές οντότητες-αντικείμενα να μπορούν να μοιράζονται και να δρουν πάνω σε ένα ενιαίο φυσικό σύστημα. Η εικονικοποίηση επίσης αποτρέπει εικονικές μηχανές από την προσπάθεια εφαρμογής εξωτερικών οδηγιών είτε χρήσης διαφορετικών στοιχείων συστήματος από αυτά που έχουν προκαθοριστεί από τους σχεδιαστές και το υλικό.

### 3.2.γ Οπτική λογισμικού αρχιτεκτονικής αναφοράς OpenFog

Η οπτική λογισμικού αποτελείται από το λογισμικό που εκτελείται σε μια πλατφόρμα υπολογιστικής ομίχλης που αποτελείται από έναν ή περισσότερους κόμβους σε συνδυασμό με άλλα αρχιτεκτονικά στοιχεία τα οποία συνεργάζονται αρμονικά για τη δημιουργία ενός συστήματος που ανταποκρίνεται στις ανάγκες του εκάστοτε σεναρίου. Το λογισμικό που χρησιμοποιείται και εκτελείται από πλατφόρμες υπολογιστικής ομίχλης είναι κατάλληλα παραμετροποιημένο προκειμένου να ικανοποιεί παρούσες καθώς και επικείμενες ανάγκες της υλοποίησης. Προκειμένου να καθίσταται δυνατή η αξιόπιστη λειτουργία της πλατφόρμας και να διατηρείται υψηλή η αποδοτικότητα στην εξαγωγή αναλυτικών στοιχείων, δημιουργία υψηλότερου επιπέδου ευφυΐας, παραγωγή καθαρής πληροφορίας και παροχής λύσεων είναι επιτακτική η συνεχής συνεργασία μεταξύ κόμβου, πλατφόρμας και λογισμικού.



Εικόνα 3.6: Αναπαράσταση οπτικής λογισμικού αρχιτεκτονικής OpenFog.

Όπως διακρίνεται στο παραπάνω σχήμα, υπάρχουν τρία στρώματα που τοποθετούνται πάνω από το στρώμα του υλικού πλατφόρμας και αποτελούν την οπτική συστήματος της αρχιτεκτονικής OpenFog:

**1. Διαχείριση κόμβων και παράλληλου λογισμικού:** Το συγκεκριμένο στρώμα της οπτικής λογισμικού είναι υπεύθυνο για τη γενικότερη λειτουργία του κόμβου, τη διαχείριση λειτουργιών του κόμβου και η διαχείριση των διαδικασιών επικοινωνίας αυτού τόσο με άλλους κόμβους όσο και με συστήματα. Η παραλληλότητα λογισμικού που υποστηρίζεται και καθίσταται δυνατή μέσω αυτού του στρώματος της αρχιτεκτονικής είναι ζωτικής σημασίας για την δημιουργία καναλιών ευκολότερης, αποδοτικότερης και ταχύτερης επικοινωνίας μεταξύ κόμβων.

**1.1.** Το κομμάτι του παράλληλου λογισμικού περιλαμβάνει:

- 1.1.1.** Λειτουργικά σύστημα με πολλαπλές μονοπύρηνες μικροεπεξεργαστικές μονάδες που λειτουργούν πάνω από ένα επίπεδο εικονικοποίησης και εκτείνονται έως και το επίπεδο μικροϋπηρεσιών εφαρμογών
- 1.1.2.** Drivers λογισμικού και υλικού που επιτρέπουν τη δημιουργία διεπαφών μεταξύ τους
- 1.1.3.** Υπηρεσίες επικοινωνίας που επιτρέπουν την επικοινωνία μεταξύ κόμβων που ανήκουν στο ίδιο είτε διαφορετικά επίπεδα ιεραρχίας και ενδέχεται να βοηθήσουν στον καθορισμό δικτύων και στον καθορισμό στοιβών πρωτοκόλλων
- 1.1.4.** Λογισμικό συστήματος διαχείρισης αρχείων
- 1.1.5.** Παροχή υπηρεσιών υποστήριξης εικονικοποίησης βάσει υλικού για την εκτέλεση μικροϋπηρεσιών λογισμικού και εφαρμογών
- 1.1.6.** Παροχή υπηρεσιών απομόνωσης (containerization) βάσει λειτουργικού συστήματος για την εκτέλεση μικροϋπηρεσιών λογισμικού και εφαρμογών

\* Τα κιβώτια απομόνωσης λογισμικού παρέχουν μηχανισμούς απομόνωσης πόρων της βασικής μονάδας επεξεργασίας (CPU), μνήμης, διεπαφών εισόδου-εξόδου, δικτύου και προσφέρουν ξεχωριστά namespaces τα οποία λόγω της φύσης των κιβωτίων (containers), τα οποία δεν απαιτούν λειτουργικό σύστημα είτε ενσωματώνουν ξεχωριστό λειτουργικό σύστημα, απομονώνουν την οπτική εφαρμογή του λειτουργικού συστήματος.

- 1.1.7. Δυνατότητες ανακάλυψης υπηρεσιών και λειτουργιών ετερογενών συστημάτων υπολογιστικής ομίχλης καθώς και δημιουργίας πολλαπλών ορίων εμπιστοσύνης μεταξύ υλοποιήσεων ομίχλης που πρέπει να συνεργαστούν
  - 1.1.8. Δυνατότητες ανακάλυψης κόμβων υπολογιστικής ομίχλης είτε υλοποιήσεων υπολογιστικής ομίχλης που ενσωματώνονται σε ένα ευρύτερο σύστημα υπολογιστικής ομίχλης και πλέον αποτελούν ένα σύμπλεγμα κόμβων από και προς τους οποίους πρέπει να γίνει μεταφορά δεδομένων και πρέπει να επιτευχθεί επικοινωνία
  - 1.1.9. Δυνατότητα υποστήριξης και διαχείρισης της κατάστασης stateful και stateless υπολογιστικών μοντέλων
  - 1.1.10. Δυνατότητες υποστήριξης και διαχείρισης υποδομών για τη δημοσίευση συμβάντων, την ειδοποίηση για αλλαγές κατάστασης και τη μετάδοση μηνυμάτων.
- 1.2. Το κομμάτι της διαχείρισης κόμβων υπεύθυνο για τη διατήρηση αρχιτεκτονικών στοιχείων και στοιχείων λογισμικού των κόμβων υπολογιστικής ομίχλης στην προκαθορισμένη επιθυμητή κατάσταση καθώς και για τη διατήρηση της λειτουργίας του του κόμβου σε επίπεδα τέτοια που να μην δημιουργούνται θέματα μη διαθεσιμότητας υπηρεσιών, θέματα ανθεκτικότητας τόσο στο χρόνο όσο και στις ολοένα μεταβαλλόμενες και πιο σύνθετες απαιτήσεις της υλοποίησης καθώς και θέματα μειωμένης απόδοσης. Παρακάτω παρατίθεται το σύνολο των δυνατοτήτων των κόμβων υπολογιστικής ομίχλης:
- 1.2.1. Διατήρηση τη επιθυμητής κατάστασης του λειτουργικού συστήματος καθώς και του χρόνου εκτέλεσης της εφαρμογής μέσω ενός παράγοντα λογισμικού που επιτρέπει τη διαχείριση των λειτουργικών παραμέτρων του κόμβου καθώς και την υποστήριξη εφαρμογών
  - 1.2.2. Δυνατότητες καταγραφής, αποθήκευσης και εξαγωγής αναλυτικών στοιχείων και συμπερασμάτων για το προσωπικό διαχείρισης συστήματος ή αυτοματοποιημένα συστήματα παρακολούθησης και διαχείρισης της υποδομής προκειμένου να ληφθεί παραμετροποιημένη και αυτοματοποιημένη ή μη λήψη μέτρων για την επίλυση προβλημάτων, διόρθωση λαθών και αποκατάσταση λειτουργίας του συστήματος
  - 1.2.3. Δυνατότητες διαχείρισης παραμέτρων ασφαλείας, κλειδιών, κρυπτογράφησης και επιβεβαίωσης ταυτότητας
  - 1.2.4. Δυνατότητες συνεχούς παρακολούθησης της χωρητικότητας και δυνατότητες οργάνωσης πρόσθετων υπολογιστικών πόρων, πόρων δικτύωσης και αποθήκευσης
  - 1.2.5. Δυνατότητες παροχής συνεχούς διαθεσιμότητας παροχής υπηρεσιών ακόμη και σε περιπτώσεις αστοχίας υλικού είτε δυσλειτουργίας λογισμικού σε έναν ή παραπάνω κόμβους, πράγμα το οποίο σημαίνει τη μεταφορά του φόρτου εργασίας σε διαθέσιμο και ικανό για την εξυπηρέτηση του αιτήματος κόμβο υπολογιστικής ομίχλης. Απαραίτητη είναι η διατήρηση επαρκούς χωρητικότητας συστήματος σε κατάσταση ετοιμότητας προκειμένου να μην διακυβεύονται τυχόν προσυμφωνημένες δεσμεύσεις μεταξύ του παρόχου υπηρεσίας και του τελικού χρήστη της εκάστοτε υλοποίησης υπολογιστικής ομίχλης

- 2. Υποστήριξη στο επίπεδο εφαρμογών:** Αυτό το στρώμα της οπτικής λογισμικού της αρχιτεκτονικής OpenFog περιλαμβάνει ένα ευρύ φάσμα λογισμικών που χρησιμοποιούν και μοιράζονται πληθώρα εφαρμογών και εξαρτάται από τα υποκείμενα επίπεδα της αρχιτεκτονικής. Σε ορισμένες περιπτώσεις ενδέχεται να συμβεί απομόνωση στοιχείων του εν λόγω επιπέδου σε κιβώτια απομόνωσης μέσω κάποιας μορφής εικονικοποίησης προκειμένου να επιτευχθεί ευκολότερη ζεύξη, αυξημένη ασφάλεια και ενίοτε ταχύτερη είτε ευκολότερη κλιμάκωση του παρόντος επιπέδου. Το επίπεδο υποστήριξης εφαρμογών περιλαμβάνει:
- 2.1.** Συνεχή ενημέρωση και διαχείριση του λογισμικού και παραμέτρων εφαρμογών
  - 2.2.** Παροχή ενός συνόλου από παραμετροποιημένα, απομονωμένα και ασφαλή περιβάλλοντα εκτέλεσης εφαρμογών και υπηρεσιών
  - 2.3.** Παροχή διακομιστών εφαρμογών είτε διακομιστών ιστού που δύνανται να φιλοξενήσουν και να παρέχουν μικροϋπηρεσίες υποστήριξης κόμβων
  - 2.4.** Υποστήριξη εφαρμογών ανταλλαγής μηνυμάτων, εφαρμογών οι οποίες είναι εξαρτώμενες από συμβάντα καθώς και υποστήριξη μικροϋπηρεσιών επικοινωνίας
  - 2.5.** Παροχή υποστήριξης υπηρεσιών ασφαλείας στο επίπεδο υποστήριξης εφαρμογών
  - 2.6.** Δυνατότητα επεξεργασίας των δεδομένων εφαρμογών και αποθήκευση αυτών
  - 2.7.** Παροχή εργαλείων και εφαρμογών για την αξιόπιστη εξαγωγή αναλυτικών στοιχείων
- 3. Επίπεδο υπηρεσιών εφαρμογής:** Λόγω της ανομοιογένειας των υλοποιήσεων συστημάτων υπολογιστικής ομίχλης βάσει του σεναρίου που καλούνται να αντιμετωπίσουν, της μεταβαλλόμενης κλίμακα της εκάστοτε υλοποίησης η οποία συνεπάγεται τη χρήση διαφορετικών αρχιτεκτονικών στοιχείων και στοιχείων λογισμικού και της απόκλισης διαθέσιμων πόρων από ένα σενάριο σε ένα άλλο, το παρόν επίπεδο παρέχει μια ευέλικτη συλλογή παραμετροποιήσιμων μικροϋπηρεσιών για την κάλυψη των αναγκών των υπηρεσιών υπολογιστικής ομίχλης. Οι εφαρμογές ενδέχεται να εκτελούνται και σε περιβάλλοντα απομονωμένα και ασφαλή όπως εικονικές μηχανές είτε container και να επωφελούνται από τις μικροϋπηρεσίες αυτού του επιπέδου. Το σύνολο των υπηρεσιών αυτού του επιπέδου εκτελούνται πλησίον των αντικειμένων IoT και περιέχουν ένα σύνολο διεπαφών που καθιστά δυνατή την επικοινωνία των υπηρεσιών υπολογιστικής ομίχλης ανώτερου επιπέδου με τα εν λόγω αντικείμενα. Οι βασικές υπηρεσίες του επιπέδου κάνουν ευδιάκριτο τον διαχωρισμό της κύριας υλοποίησης από τις συσκευές και τα αντικείμενα στα άκρα του συστήματος ενώ οι δευτερεύουσες μικροϋπηρεσίες περιλαμβάνουν ένα ευρύ φάσμα υπηρεσιών καταγραφής και διαχείρισης δεδομένων. Οι υπηρεσίες εξαγωγής αναλυτικών στοιχείων προσδίδουν στους κόμβους που βρίσκονται στα άκρα του συστήματος υπηρεσίες που είναι πιο αντιδραστικές στη φύση μέσω της τροφοδότησης με πληροφορίες για το περιβάλλον ενώ στους κόμβους που είναι απομακρυσμένοι από τα τοπικά δίκτυα παραγωγής δεδομένων, τα αναλυτικά στοιχεία προσδίδουν αυξημένες ικανότητες πρόβλεψης βάσει μηχανικής μάθησης και ανάπτυξη τεχνητής μηχανικής ευφυΐας.



## ΚΕΦΑΛΑΙΟ 4 RELATED WORK

---

### 4.1 Κατανομή πόρων

---

Η κατανομή πόρων είναι η συστηματική προσέγγιση της κατανομής των διαθέσιμων πόρων μεταξύ των κόμβων υπολογιστικής ομίχλης και του υπολογιστικού νέφους μέσω του τοπικού δικτύου και του διαδικτύου. Στα υπολογιστικά συστήματα και πιο συγκεκριμένα στις αρχιτεκτονικές υπολογιστικού νέφους και υπολογιστικής ομίχλης, τα σενάρια και οι υλοποιήσεις είναι δυναμικά και πολύπλοκα με αποτέλεσμα η κατανομή πόρων να αποτελεί σημαντική πρόκληση για τους παρόχους εφαρμογών, υπηρεσιών και των εν λόγω συστημάτων.

Οι πόροι συστήματος πρέπει να κατανέμονται βέλτιστα στις εφαρμογές που εκτελούνται σε επίπεδο ιεραρχίας κόμβων είτε στα datacenter των υπολογιστικών νεφών. Η σειρά και η χρονική κατανομή των πόρων είναι ένας σημαντικός παράγοντας για την επίτευξη μιας διαδικασίας βέλτιστης κατανομής πόρων. Ως προέκταση αυτού, έχει γίνει έρευνα ενός σεναρίου το οποίο βασίζεται στην διορατικότητα και στη δυνατότητα πρόβλεψης προκειμένου να επιτευχθεί η ανάλυση των απαιτούμενων πόρων. Πιο συγκεκριμένα έχει προταθεί ότι η πιθανότητα διατήρησης πόρων από διαφορετικούς τελικούς χρήστες μπορεί να χρησιμοποιηθεί για την εξαγωγή προβλέψεων που αφορούν ενδεχόμενες μελλοντικές συνθήκες φόρτωσης. Μηχανισμοί εικονικοποίησης διακομιστών χρησιμοποιούνται προκειμένου να επιτευχθεί η ικανοποίηση των τελικών χρηστών καθώς και η αναλογία κατανομής και χρήσης πόρων τόσο σε κόμβους υπολογιστικής ομίχλης όσο και στο υπολογιστικό νέφος.

Ο τεράστιος όγκος δεδομένων που παράγεται από τις τελικές συσκευές συστημάτων IoT και η δυνατότητα που παρέχουν οι τεχνολογίες υπολογιστικής ομίχλης που ενσωματώνονται σε δίκτυα Cloud-IoT για λήψη αποφάσεων σε πραγματικό χρόνο σημαίνει την ταυτόχρονη αποστολή δεδομένων από και προς κόμβους υπολογιστικής ομίχλης με αποδέκτες το υπολογιστικό νέφος είτε τις τελικές συσκευές. Η δημιουργία μεγάλου όγκου δεδομένων (Big Data) σημαίνει ότι σε περιπτώσεις που η κατανομή των πόρων δεν είναι αποδοτική, η συνεχής ανταλλαγή πληροφορίας και οι διαδρομές που διανύουν τα μπλοκ δεδομένων θα έχουν τεράστιο κόστος και θα απαιτούν αρχιτεκτονικά στοιχεία με πολύ μεγαλύτερη επεξεργαστική ισχύ με αποτέλεσμα το κόστος να αυξάνεται σημαντικά σε όλα τα επίπεδα ιεραρχίας κόμβων και σε περιπτώσεις όπου τα δεδομένα θα πρέπει να ταξιδέψουν από και προς μια σειρά από απομακρυσμένα υπολογιστικά νέφη με σημαντικά μεγαλύτερες χωρητικότητες και επεξεργαστική ισχύ, τότε τα συστήματα και οι υλοποιήσεις καταλήγουν να είναι ασύμφορα και μη αποδοτικά.

Στη μια όψη του νομίσματος βρίσκονται οι προμηθευτές, οι οποίοι έχουν ως βασικό στόχο τη δυναμική διάθεση των πόρων προκειμένου να πετύχουν το υψηλότερο δυνατό οικονομικό κέρδος ενώ στην άλλη όψη του νομίσματος βρίσκονται οι καταναλωτές οι οποίοι βασίζονται στην αποδοτική κατανομή πόρων σε υλοποιήσεις Cloud-Fog-IoT προκειμένου να φέρουν εις πέρας απαιτητικές εργασίες και να καλύψουν ατομικές ανάγκες, εργασιακές ανάγκες, ανάγκες που αφορούν βιομηχανία, επιχειρήσεις, εμπορεύματα, υπηρεσίες υγείας, υπηρεσίες μεταφορών και πολλές ακόμη υπηρεσίες και εφαρμογές της καθημερινότητας.

Υπάρχουν πολλές τεχνικές κατανομής πόρων οι οποίοι είναι προσκείμενοι τόσο στο υπολογιστικό νέφος όσο και στην υπολογιστικής ομίχλη.

Μεταξύ άλλων, τα σημαντικότερα ζητήματα που αφορούν την κατανομή δεδομένων είναι τα εξής:

- **Επεκτασιμότητα:** Απαιτείται η χρήση αλγορίθμων κατανομής πόρων οι οποίοι πρέπει να είναι επεκτάσιμοι προκειμένου να μπορούν να χρησιμοποιούνται για την αποτελεσματικά δυναμική κατανομή πόρων
- **Ενεργειακή απόδοση:** Η επίτευξη μειωμένης κατανάλωσης ενέργειας κατά τις περιόδους λειτουργίας των εν λόγω συστημάτων θα βελτιώσει την απόδοση του συστήματος
- **Προγραμματισμός εργασιών:** Η αποτελεσματική διαχείριση των εργασιών που εκτελούνται από κόμβους συστήματος σημαίνει την ελαχιστοποίηση του χρόνου εκτέλεσης και την επίτευξη ενός αποδοτικού συστήματος
- **Εξισορρόπηση φορτίου:** Όταν εφαρμόζονται αποτελεσματικοί μηχανισμοί εξισορρόπησης εισερχόμενων και εξερχόμενων φορτίων, τυχών έξοδα και επιβαρύνσεις που σχετίζονται με τους κόμβους συστήματος ελαχιστοποιούνται

Εξέχουσας σημασίας είναι και οι προδιαγραφές λειτουργίας υλοποιήσεων συστημάτων υπολογιστικής ομίχλης.

## 4.2 Data Placement Αλγόριθμοι

---

Οι υλοποιήσεις διαφόρων σεναρίων αρχιτεκτονικής υπολογιστικής ομίχλης έχει ως σκοπό την αποτελεσματικότερη διαχείριση του εφαρμογών και διαδικασιών δικτύου και αποθήκευσης δεδομένων καθώς και την αποδοτικότερη επεξεργασία των δεδομένων από τις εφαρμογές. Προκειμένου να επιτευχθεί το παραπάνω εγχείρημα, υπάρχουν υλοποιήσεις αρχιτεκτονικών ομίχλης οι οποίοι ενσωματώνουν μηχανισμούς για την καλύτερη διανομή των πόρων.

Ένα παράδειγμα αρχιτεκτονικής δικτύωσης είναι αυτό που αποτελείται από τρία επίπεδα αρχιτεκτονικών στοιχείων και τεχνολογιών. Υπάρχει το κατώτερο επίπεδο συσκευών IoT το οποίο περιλαμβάνει μεταξύ άλλων αισθητήρες και ενεργοποιητές. Το δεύτερο επίπεδο ενσωματώνει συσκευές οι οποίες αποτελούν κόμβους υπολογιστικής ομίχλης οι οποίοι συλλέγουν δεδομένα και έπειτα από επεξεργασία εξάγουν πληροφορία υπό τη μορφή δεδομένων και λύσεις υπό τη μορφή εντολών προς τις συσκευές του κατώτερου επιπέδου. Τέλος υπάρχει το ανώτερο επίπεδο του υπολογιστικού νέφους όπου αποτελείται από συσκευές με σημαντικά αυξημένη επεξεργαστική ισχύ και δυνατότητες αποθήκευσης και παροχής υπηρεσιών που δεν μπορούν να παρέχονται σε κανένα από τα κατώτερα επίπεδα του συγκεκριμένου σεναρίου αρχιτεκτονικής υπολογιστικής ομίχλης.

Επεξεργασία δεδομένων συμβαίνει σε όλα τα επίπεδα της παραπάνω αρχιτεκτονικής βάσει αναγκών επεξεργαστικής ισχύς, δυνατοτήτων αποθήκευσης και πόρων δικτύου.

Υπάρχουν αλγόριθμοι τοποθέτησης δεδομένων (data placement) οι οποίοι είναι υπεύθυνοι για την διανομή υπηρεσιών, δεδομένων και εφαρμογών σε συγκεκριμένες συσκευές και επίπεδα της αρχιτεκτονικής. Βάσει κάθε αλγορίθμου υπάρχει διαφοροποίηση όσον αφορά την τοποθέτηση επεξεργασίας των δεδομένων καθώς και την κατανομή πόρων δικτύου. Θα αναλύσουμε τρεις αλγόριθμους data placement οι οποίοι προσφέρουν διαφορετικές στρατηγικές τοποθέτησης δεδομένων.

#### 4.2.α Edge-ward αλγόριθμος

---

Ο αλγόριθμος edge-ward [6] θέλει την επεξεργασία δεδομένων αλλά και τη λήψη αποφάσεων αν συμβαίνει όσο το δυνατό πλησιέστερα στα άκρα του δικτύου. Βασίζεται στη στρατηγική First Come-First Serve και αντιπροσωπεύει μια στρατηγική data placement η οποία συμβαίνει στις συσκευές-κόμβους υπολογιστικής ομίχλης.

Οι απαιτήσεις μιας εφαρμογής εξυπηρετούνται από έναν ή περισσότερους κόμβους υπολογιστικής ομίχλης. Στην περίπτωση που ένας κόμβος υπολογιστικής ομίχλης δεν δύναται να φέρει εις πέρας την εκτέλεση όλων των τμημάτων μιας εφαρμογής, τότε επιστρατεύονται επιπλέον συσκευές υπολογιστικής ομίχλης οι οποίες μοιράζονται το υπολογιστικό βάρος εκτέλεσης μιας εφαρμογής και αντιπροσωπεύουν τις διαδρομές μέσω των οποίων εκτελούνται τα επιμέρους τμήματα της εφαρμογής.

Τα αιτήματα εφαρμογών απαντώνται με βάση τη σειρά άφιξης στον κόμβο ομίχλης και σε περίπτωση που στον κόμβο έχει ανατεθεί η περάτωση μιας άλλης εφαρμογής με αποτέλεσμα να μην υπάρχουν διαθέσιμοι πόροι, τότε τα αιτήματα μεταβαίνουν σε μια κατάσταση αναμονής μέχρι να απελευθερωθούν οι απαιτούμενοι υπολογιστικοί πόροι. Σε περιπτώσεις όπου δεν είναι δυνατή η εκτέλεση μιας εφαρμογής είτε τμημάτων αυτής λόγω ελλιπούς υπολογιστικής χωρητικότητας, τότε μέσω του αλγορίθμου αναζητείται συσκευή υπολογιστικής ομίχλης σε ανώτερα επίπεδα ιεραρχίας τοπολογίας δικτύου και πάντα μέσα στα πλαίσια του επιπέδου υπολογιστικής ομίχλης η οποία να διαθέτει την απαραίτητη χωρητικότητα και επεξεργαστική δυνατότητα προκειμένου να εξυπηρετήσει τις απαιτήσεις του εκάστοτε τμήματος εφαρμογής.

Ο αλγόριθμος edge-ward παρουσιάζει καλύτερη απόδοση σε περιπτώσεις όπου ο σχεδιασμός και κατανομή πόρων των συσκευών υπολογιστικής ομίχλης υλοποιείται με ιεραρχικό τρόπο (αρχιτεκτονική Fog N-κόμβων) και σκοπός του είναι η εξυπηρέτηση αιτημάτων εφαρμογών αποκλειστικά από κόμβους υπολογιστικής ομίχλης ανεξαρτήτως επιπέδου ιεραρχίας κόμβου και με τρόπο τέτοιο που τα αιτήματα επεξεργάζονται και απαντώνται βάσει σειράς άφιξης στον κόμβο προς επεξεργασία. Επιλέγει όλες τις διαδρομές από τη συσκευή στο νοτιότερο άκρο προς τη βορειότερη συσκευή στην τοπολογία και τοποθετεί ενότητες εφαρμογών σε κάθε διαδρομή. Για κάθε διαδρομή, ο αλγόριθμος επαναλαμβάνεται και σταδιακά τοποθετεί όλες τις ενότητες εφαρμογών στις εν λόγω διαδρομές.

Στην περίπτωση μόνο όπου δεν είναι δυνατή η επεξεργασία κάποιου αιτήματος από κανέναν κόμβο επιπέδου ιεραρχίας και υπάρχουν απαιτήσεις εφαρμογής ή τμήματος αυτής στις οποίες δεν μπορούν να ανταποκριθούν οι κόμβοι υπολογιστικής ομίχλης, το αίτημα εφαρμογής προωθείται για επεξεργασία και προγραμματισμό στο νέφος.

**Algorithm 1:** Edge-ward module placement

```

for  $p \in \text{PATHS}$  do Across all paths
   $\text{placedModules} \leftarrow \{\}$ ;
  for Fog device  $d \in p$  do ▷ leaf-to-root traversal
     $\text{modulesToPlace} \leftarrow \{\}$ ;
    for module  $w \in \text{app}$  do ▷ find modules ready for placement on device  $d$ 
      if all predecessors of  $w$  are in  $\text{placedModules}$  then ▷ if all predecessors are placed
        add  $w$  to  $\text{modulesToPlace}$ ;
      end
    end
    for module  $\theta \in \text{modulesToPlace}$  do
      if  $d$  already has instance of  $\theta$  as  $\theta'$  then
        if  $\text{CPU}_\theta^{\text{req}} \geq \text{CPU}_d^{\text{avail}}$  then ▷ device  $d$  does not have CPU capacity to host  $\theta$ 
           $\tilde{\theta} \leftarrow \text{merge}(\theta, \theta')$ ;
           $f \leftarrow \text{parent}(d)$ ;
          while  $\text{CPU}_{\tilde{\theta}}^{\text{req}} \geq \text{CPU}_f^{\text{avail}}$  do ▷ find device north of  $d$  for hosting  $\theta$ 
             $f \leftarrow \text{parent}(f)$ ;
          end
          Place  $\tilde{\theta}$  on device  $f$ ;
          add  $\theta$  to  $\text{placedModules}$ ;
        end
      else ▷ device  $d$  can host  $\theta$ 
        Place  $\theta$  on device  $d$ ;
        add  $\theta$  to  $\text{placedModules}$ ;
      end
    end
    else if no device north of  $d$  has an instance of  $\theta$  then ▷ if not, will be handled by subsequent iterations
      if  $\text{CPU}_\theta^{\text{req}} \leq \text{CPU}_d^{\text{avail}}$  then
        Place  $\theta$  on device  $d$ ;
        add  $\theta$  to  $\text{placedModules}$ ;
      end
    end
  end
end

```

**Εικόνα 4.1:** Ψευδοκώδικας του data placement αλγόριθμου edge-ward [6]

Έννοιες που χρησιμοποιούνται στον ψευδοκώδικα αντιπροσωπεύουν:

$\vartheta$ : επιλεγμένο τμήμα εφαρμογής (selected module)

$p$ : μονοπάτι-διαδρομή (path)

$d$ : συσκευή (device)

$w$ : τμήματα εφαρμογής (modules)

Όταν αφιχθούν τα τμήματα εφαρμογής προς εξυπηρέτηση, ο αλγόριθμος προσπαθεί να τα τοποθετήσει βάσει σειράς άφιξης ένα προς ένα στη συσκευή. Σε περίπτωση που ένα επιλεγμένο τμήμα εφαρμογής ( $\vartheta$ ) τοποθετηθεί στη συσκευή ( $d$ ), ο αλγόριθμος ελέγχει εάν το κάποιο μέρος του συγκεκριμένου τμήματος έχει ήδη τοποθετηθεί στη συσκευή σε προηγούμενο χρόνο ως μέρος κάποιας άλλης διαδρομής. Εάν συμβαίνει κάτι τέτοιο τότε τα μέρη του τμήματος της εφαρμογής συγχωνεύονται και τοποθετούνται στη συσκευή εάν αυτή μπορεί να τα φιλοξενήσει, διαφορετικά αναζητείται συσκευή βόρεια της παρούσας ( $d$ ) για την εκτέλεση των συγχωνευμένων μερών τμήματος εφαρμογής. Σε περίπτωση που δεν υπάρχει μέρος ενός τμήματος εφαρμογής ( $\vartheta$ ) στην παρούσα συσκευή ( $d$ ) αλλά και σε καμία συσκευή βορειότερα αυτής, τότε το παρόν επιλεγμένο τμήμα εφαρμογής ( $\vartheta$ ) τοποθετείται στη συσκευή ( $d$ ). Διαφορετικά, τμήμα της εφαρμογής που είχε ανατεθεί σε μια οποιαδήποτε συσκευή βορειότερα της παρούσας, θα διαμορφωθεί από τα πρώιμα βήματα εκτέλεσης του αλγορίθμου σε μελλοντική εκτέλεση του βρόχου επανάληψης.

## 4.2.β Cloud-only αλγόριθμος

Ο αλγόριθμος cloud-only [6] χρησιμοποιεί την στρατηγική προτεραιότητας καθυστέρησης (delay priority strategy) και βασίζεται σε υλοποίηση η οποία θέλει τα δεδομένα και η επεξεργασία αυτών να συμβαίνει σε κέντρα δεδομένων απομακρυσμένων υπολογιστικών νεφών. Οι λειτουργικές μονάδες εφαρμογών εκτελούνται στο νέφος. Οι αισθητήρες καταγράφουν πληροφορία τοπικά υπό τη μορφή δεδομένων για το περιβάλλον γύρω τους και τα δεδομένα μέσω των κόμβων υπολογιστικής ομίχλης προωθούνται στο νέφος προκειμένου μετά από επεξεργασία να προκύψουν αναλυτικά στοιχεία και συγκεκριμενοποιημένη γνώση. Έπειτα οι αποφάσεις για τις λειτουργίες των αρχιτεκτονικών στοιχείων του συστήματος λαμβάνονται στο νέφος και αποστέλλονται εάν αυτό είναι απαραίτητα μέσω των κόμβων ομίχλης σε τελικές συσκευές και ενεργοποιητές προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα.

Όμοια, έννοιες που χρησιμοποιούνται στον ψευδοκώδικα αντιπροσωπεύουν:

$\theta$ : επιλεγμένο τμήμα εφαρμογής (selected module)

$p$ : μονοπάτι-διαδρομή (path)

$d$ : συσκευή (device)

$w$ : τμήματα εφαρμογής (modules)

---

### Algorithm 2 – Cloud-only

---

```
1: while  $p \in PATHS$  do ▷ Across all paths
2:    $placeList := \{\}$  ▷ device list
3:   while Fog device  $d \in p$  do ▷ way
4:     while module  $w \in app$  do
5:       if all predec. of  $w$  are placed then
6:         add  $w$  to  $placeList$ 
7:       end if
8:     end while
9:     while module  $\theta \in placeList$  do
10:      Place  $\theta$  on device  $CLOUD$ 
11:    end while
12:   end while
13: end while
```

---

Εικόνα 4.2: Ψευδοκώδικας του data placement αλγόριθμου cloud-only [6]

Όταν δημιουργούνται αιτήματα εφαρμογών προς εξυπηρέτηση, τότε εντός βρόγχου αρχίζει η διαδικασία ελέγχου όλων των συσκευών ( $d$ ) εντός μιας διαδρομής ( $p$ ) προκειμένου να διαπιστωθεί εάν υπάρχει σε κάποιον από τους κόμβους αυτούς κάποιο τμήμα της εφαρμογής ( $w$ ) το οποίο προηγείται του παρόντος. Σε περίπτωση που όλα τα τμήματα εφαρμογής που προηγούνται του παρόντος έχουν τοποθετηθεί στη λίστα συσκευών-διαδρομή ( $p$ ) και συνεχίζουν να υπάρχουν τμήματα εφαρμογής που δεν έχουν τοποθετηθεί σε αυτή ( $p$ ), τότε τοποθετούνται και αυτά αντίστοιχα στο τέλος της λίστας. Όταν όλα τα τμήματα της εφαρμογής έχουν τοποθετηθεί στη λίστα συσκευών-διαδρομή ( $p$ ), η εφαρμογή είναι έτοιμη να προωθηθεί στο επίπεδο του νέφους προκειμένου συγκεκριμένα τμήματα εφαρμογής ( $\theta$ ) που ανήκουν στη λίστα συσκευών-διαδρομή ( $p$ ) να είναι διαθέσιμα προς επεξεργασία και εξαγωγή αναλυτικών στοιχείων.

## 4.3 Load Balancing Αλγόριθμοι

Κατά την υλοποίηση σεναρίων που ενσωματώνουν υπολογιστική ομίχλη, στόχος είναι η δυνατότητα επεξεργασίας δεδομένων πιο κοντά στον τελικό χρήστη προκειμένου να μειωθεί ο λανθάνον χρόνος, να αυξηθεί η ασφάλεια των δεδομένων και να μην υπάρχει άμεση εξάρτηση από απομακρυσμένα υπολογιστικά νέφη. Η εκτέλεση εφαρμογών οι οποίες εμφανίζουν μια ευαισθησία σε χρονικές καθυστερήσεις είναι σχεδόν αδύνατο να εκτελεστούν στο νέφος λόγω γεωγραφικής κατανομής η οποία σημαίνει ότι στην επικοινωνία μεταξύ τελικού χρήστη και cloud datacenter υπάρχει μεγάλη καθυστέρηση.

Οι συσκευές ομίχλης που αποτελούν κόμβους στο επίπεδο ιεραρχίας αρχιτεκτονικής υπολογιστικής ομίχλης θεωρούνται κινητά datacenter τα οποία βρίσκονται εντός του τοπικού δικτύου και η επικοινωνία με αυτούς δεν ενέχει χρονικές καθυστερήσεις. Η έλλειψη καθυστέρησης σημαίνει ότι οι κόμβοι είναι ικανοί να επεξεργαστούν αιτήματα εφαρμογών και να εκτελέσουν τμήματα αυτών στην άκρη του δικτύου ακόμη και όταν πρόκειται για εφαρμογές ευαίσθητες σε χρονικές καθυστερήσεις. Πιο συγκεκριμένα, τμήματα εφαρμογών και ροές δεδομένων που απαιτούν χαμηλό λανθάνοντα χρόνο μπορούν να εκτελούνται από κινητά datacenter ενώ εφαρμογές με μεγαλύτερες απαιτήσεις επεξεργαστικής ισχύος και διεργασίες ανθεκτικές σε κατάσταση λανθάνουσα προωθούνται σε cloud datacenter για επεξεργασία.

Η επεξεργαστική ισχύς που δαπανάται από τον κάθε κόμβο ομίχλης διαφέρει βάσει του φόρτου δεδομένων προς επεξεργασία είτε βάσει εφαρμογών και αιτημάτων που καλείται να εκτελέσει. Προκειμένου να γίνεται αποδοτικότερη διαχείριση του φορτίου εργασίας και να προκύπτουν συστήματα με αυξημένη αξιοπιστία και αποδοτικότητα, χαμηλό κόστος και μειωμένες χρονικές καθυστερήσεις, είναι απαραίτητη η ενσωμάτωση μηχανισμών εξισορρόπησης φορτίου σε όλο το εύρος των κινητών datacenter. Παρακάτω παραθέτουμε αλγόριθμους εξισορρόπησης φορτίου που εμφανίζονται στο paper “ Load balancing aware scheduling algorithms for fog networks ” των Anil Singh και Nitin Auluck. Πριν συνεχίσουμε στην παράθεση των αλγορίθμων, ορίζουμε μεταβλητές και τις έννοιες που αντιπροσωπεύουν προκειμένου να είναι πιο κατανοητή η μελέτη των αλγορίθμων.

$C$	Σύνολο των data center υπολογιστικού νέφους
$M$	Σύνολο των microdata center (κόμβοι ομίχλης)
$d$	Σύνολο των data center (υπολογιστικού νέφους και microdata)
$m_i, m_f$	Τοπικά και απομακρυσμένα data center
$n$	Συνολικός αριθμός εργασιών
$SJ$	Σύνολο όλων των εργασιών
$FD$	Σύνολο όλων των συσκευών ομίχλης
$L_j$	Σύνολο των ετικετών που έχουν ανατεθεί σε εργασίες
$L_r$	Σύνολο των ετικετών που έχουν ανατεθεί σε πόρους
$l_p, l_{sp}, l_{pu}$	Ετικέτα για ιδιωτικές, ημι-ιδιωτικές και δημόσιες εργασίες αντίστοιχα
$l_{ht}, l_{st}, l_{ut}$	Ετικέτα για αξιόπιστους, ημι-αξιόπιστους και αναξιόπιστους πόρους αντίστοιχα
$j_i$	Ξεχωριστή εργασία η οποία ανήκει στο σύνολο εργασιών ( $j_i \in SJ$ )
$cp(m_k)$	Χωρητικότητα ενός microdata center που ανήκει στο σύνολο αυτών ( $m_k \in M$ )
$et(j_i)$	Κόστος εκτέλεσης μιας εργασίας $j_i$
$st(j_i)$	Χρόνος έναρξης εκτέλεσης μιας εργασίας $j_i$
$ft(j_i)$	Χρόνος περάτωσης μιας εργασίας $j_i$

$CL(m_k)$	Παρόν φορτίο στο microdata center $m_k \in M$
Threshold	Μέγιστο φορτίο σε ένα microdata center χωρίς να υπάρξει υποβάθμιση απόδοσης
$D(j_i, m_k)$	Απόσταση από μια εργασία $j_i$ σε ένα microdata center $m_k$
cd	Καθυστέρηση επικοινωνίας
wt	Χρόνος αναμονής
$DC_{system}$	Κόστος υλοποίησης
$UT_{system}$	Αξιοποίηση συστήματος
$RT_{av}$	Μέσος χρόνος απόκρισης εργασίας
$FT_{av}$	Μέσος χρόνος περάτωσης εργασίας
$TD_{av}$	Μέσος χρόνος καθυστέρησης εργασίας

Πίνακας συμβόλων κλειδιά για την κατανόηση ψευδοκώδικα των αλγορίθμων [7]

	$I_{ht}$	$I_{st}$	$I_{lt}$
$I_p$	Y	N	N
$I_{sp}$	Y	Y	N
$I_{pu}$	Y	Y	Y

Πίνακας χαρτογράφησης με βάση ετικέτες ασφαλείας και πόρων [7]

Στον παραπάνω πίνακα παρουσιάζονται οι αντιστοιχίσεις μεταξύ ειδών εργασιών και ειδών πόρων. Η τιμή “Y” υποδηλώνει εγκυρότητα χαρτογράφησης και έγκυρη αντιστοίχιση μεταξύ ενός συγκεκριμένου πόρου και μιας συγκεκριμένης εργασίας ενώ η τιμή “N” υποδηλώνει μη έγκυρη αντιστοίχιση και αδυναμία χαρτογράφησης μεταξύ των παραπάνω.

Το κόστος μεταφοράς της κατάστασης της εργασίας  $j_i$  συμβολίζεται με  $sc(j_i, m_i)$ . Το κόστος επικοινωνίας μεταξύ μιας εργασίας  $j_i$  και του τοπικού datacenter  $m_i$  διαμορφώνεται ως:  $cd(j_i, m_k) = t + sc(j_i, m_k) + s(j_i) / bw$ , όπου  $bw$  το εύρος ζώνης συνδέσμου επικοινωνίας μεταξύ χρήστη και microdata center και  $t$  ο χρόνος για την προετοιμασία της σύνδεσης.

Οι εργασίες μπορούν να αποσταλούν πέραν των microdata center και σε cloud datacenter. Στην περίπτωση αυτή  $c$  είναι το cloud και  $cd(j_i, c)$  η καθυστέρηση επικοινωνίας μεταξύ μιας εργασίας  $j_i$  και του νέφους  $c$ . Η καθυστέρηση μπορεί να υπολογιστεί ως εξής:  $cd(j_i, c) = t + sc(j_i, c) + s(j_i) / bw$ , όπου  $bw$  το εύρος ζώνης συνδέσμου επικοινωνίας μεταξύ χρήστη και microdata center και  $t$  ο χρόνος για την προετοιμασία της σύνδεσης.

Οι εργασίες μπορούν να εκτελεστούν σε οποιοδήποτε microdata center αλλά συνήθως ανατίθενται στο πλησιέστερο διαθέσιμο. Πριν ανατεθεί μια εργασία σε ένα microdata center, γίνεται έλεγχος που αφορά το μέγιστο φορτίο που μπορεί να επεξεργαστεί χωρίς να υποβαθμιστεί η επεξεργαστική του απόδοση έτσι ώστε το εκάστοτε φορτίο που πρόκειται να του ανατεθεί να μην υπερβαίνει το μέγιστο φορτίο που αυτό μπορεί να υποστηρίξει.

Η ελαχιστοποίηση του μέσου χρόνου απόκρισης και του χρόνου ολοκλήρωσης καθώς και η ελαχιστοποίηση της καθυστέρησης είναι το πρόβλημα βελτιστοποίησης για το οποίο γίνονται προσπάθειες επίλυσης εντός του paper. Συγκεκριμένα προκύπτει ότι ο μέσος χρόνος απόκρισης μιας εργασίας ( $RT_{av}$ ) είναι το άθροισμα του χρόνου απόκρισης κάθε εργασίας προς το συνολικό αριθμό αυτών, μέσος χρόνος καθυστέρησης μιας εργασίας ( $TD_{av}$ ) είναι το άθροισμα του χρόνου καθυστέρησης κάθε εργασίας προς το συνολικό αριθμό αυτών και μέσος χρόνος ολοκλήρωσης εκτέλεσης μιας εργασίας ( $FT_{av}$ ) είναι το άθροισμα του χρόνου ολοκλήρωσης εκτέλεσης κάθε εργασίας προς το συνολικό αριθμό αυτών αντίστοιχα.

Από τα παραπάνω προκύπτει ότι για κάθε microdata center το κόστος εκτέλεσης μιας εργασίας  $j_i$  το πολύ ίσο με την χωρητικότητα του microdata center εάν από αυτή αφαιρέσουμε το άθροισμα του κόστους εκτέλεσης κάθε εργασίας που έχει ανατεθεί σε αυτό, ενώ αντίστοιχα για κάθε νέφος το κόστος εκτέλεσης μια εργασίας  $j_i$  το πολύ ίσο με την χωρητικότητα του cloud datacenter εάν από αυτή αφαιρέσουμε το άθροισμα του κόστους εκτέλεσης κάθε εργασίας που έχει ανατεθεί σε αυτό. Οι παραπάνω συνθήκες αποτελούν συνθήκες διαθέσιμης χωρητικότητας προκειμένου να υπάρχει επαρκής χωρητικότητα σε microdata centers και cloud datacenters για την εκτέλεση μιας εργασίας  $j_i$ .

Προτού γίνει ανάλυση των προτεινόμενων αλγορίθμων, παρατίθεται ο κύριος αλγόριθμος load balancing scheduling [7] ο οποίος χρησιμοποιεί βάσει προτίμησης συστήματος, για εξισορρόπηση φορτίου, τους αλγόριθμους που θα αναλυθούν αργότερα. Ο αλγόριθμος ξεκινά με τον υπολογισμό του χρόνου έναρξης της εργασίας ( $st(j_i)$ ), του χρόνου καθυστέρησης επικοινωνίας ( $cd(j_i)$ ) και το κόστος εκτέλεσης ( $et(j_i)$ ) της εργασίας. Στην εικόνα που ακολουθεί και στη γραμμή 2, αντί της μεταβλητής που ορίζει το κόστος εκτέλεσης μια εργασίας ( $et(j_i)$ ) έχει γραφτεί η μεταβλητή  $ct$  η οποία δεν έχει οριστεί στον πίνακα μεταβλητών. Έπειτα γίνεται μια κατηγοριοποίηση των εργασιών σε τρεις κατηγορίες βάσει της ετικέτας ασφαλείας αυτών ( $l_p, l_{sp}, l_{pu}$ ).

---

**Algorithm 1** Main load balancing scheduling algorithm

---

```

1: procedure LOAD-BALANCING
2:   Calculate  $st, ct, cd, \forall j_i \in J$ .
3:   Combine all jobs based on their labels:  $l_p, l_{sp}$  &  $l_{pu}$ .
4:   for  $i=1$  to  $n$  do
5:     if ( $j_i \rightarrow l_p$ ) then
6:       Assign  $j_i$  to its nearest mdc  $m_k$  with label  $l_{ft}$ .
7:       if ( $CL(m_k) > \text{Threshold}$ ) then
8:         Loadbalance( $j_i, l_p$ );
9:       end if
10:    end if
11:    if ( $j_i \rightarrow l_{sp}$ ) then
12:      Assign  $j_i$  to its nearest mdc  $m_k$  with label  $l_{ft}$  or  $l_{st}$ .
13:      if ( $CL(m_k) > \text{Threshold}$ ) then
14:        Loadbalance( $j_i, l_{sp}$ );
15:      end if
16:    end if
17:    if ( $j_i \rightarrow l_{pu}$ ) then
18:      Assign  $j_i$  to its nearest mdc  $m_k$  with label  $l_{ft}, l_{st}$  or  $l_{ut}$ .
19:      if ( $CL(m_k) > \text{Threshold}$ ) then
20:        Loadbalance( $j_i, l_{pu}$ );
21:      end if
22:    end if
23:  end for
24:  Calculate  $DC, UT, \forall c, m \in C, M$ .
25:  Calculate  $RT, FT, TD \forall j \in J$ .
26: end procedure

```

---

Εικόνα 4.3: Ψευδοκώδικας κύριου αλγόριθμου για load balancing [7].



Στη συνέχεια ξεκινά ένας βρόγχος που επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλες οι εργασίες. Σε περίπτωση εργασίας με ετικέτα ασφαλείας private ( $l_p$ ), η εργασία ανατίθεται στο πλησιέστερο microdata center με ετικέτα πόρων trusted ( $l_{ht}$ ). Στις γραμμές 6, 12 & 18, υπάρχει αναφορά σε μεταβλητή  $l_{ht}$  η οποία δεν έχει οριστεί και πρόκειται για τη μεταβλητή  $l_{ht}$ . Ομοίως γίνεται έλεγχος ετικέτας ασφαλείας εργασίας προκειμένου να διαπιστωθεί εάν πρόκειται για semi-private είτε public. Στις γραμμές 7, 13 & 19 γίνεται έλεγχος για να διαπιστωθεί εάν το παρόν φορτίο του microdata center στο οποίο έχει αρχικά αφιχθεί η εργασία είναι μεγαλύτερο από αυτό που μπορεί να υποστηρίξει χωρίς να μειωθεί η απόδοση.

Σε περιπτώσεις που το φορτίο του κόμβου είναι μεγαλύτερο από αυτό που μπορεί να υποστηρίξει, εκτελείται η διαδικασία Loadbalance η οποία δέχεται ως ορίσματα μια εργασία και την ετικέτα ασφαλείας αυτής και ενσωματώνει έναν από τους αλγόριθμους που θα αναλύσουμε παρακάτω για την ανάθεση αυτής στον καταλληλότερο κόμβο. Τελικά η εργασία έχει ανατεθεί στο καταλληλότερο microdata center είτε έχει προωθηθεί στο cloud και σειρά παίρνει η επόμενη αφιχθείσα εργασία. Όταν έχουν εξυπηρετηθεί όλες οι εργασίες, γίνεται υπολογισμός του κόστους υλοποίησης (DC) και της αξιοποίησης συστήματος (UT) για κάθε τοπικό είτε απομακρυσμένο microdata center, καθώς και υπολογισμός του μέσου χρόνου απόκρισης (RT) και ολοκλήρωσης (FT) της κάθε εργασίας.

Εργασίες με ετικέτα ασφαλείας private μπορούν να ανατεθούν μόνο στον αρχικό κόμβο που προωθούνται, αυτές με ετικέτα semi-private μπορούν να εκτελεστούν σε τοπικά και απομακρυσμένα microdata center και οι αυτές με ετικέτα public μπορούν να εκτελεστούν σε οποιαδήποτε συσκευή. Σε περιπτώσεις μη ιδιωτικών εργασιών, όταν δεν υπάρχει τοπικό microdata center στο οποίο να μπορούν να ανατεθούν, προωθούνται σε απομακρυσμένα microdata center είτε σε cloud datacenters. Όταν οι εργασίες είναι ιδιωτικές και το αρχικό microdata center δεν είναι διαθέσιμο, μπαίνουν σε ουρά αναμονής και προγραμματίζονται για εκτέλεση εκ νέου όταν θα είναι διαθέσιμο το τοπικό microdata center.

#### 4.3.α Αλγόριθμος Minimum Load

Σε αυτό τον αλγόριθμο [7] εάν μια εργασία έχει αναθέσει σε ένα συγκεκριμένο microdata center που δεν διαθέτει την απαιτούμενη χωρητικότητα για να την εκτελέσει, ο αλγόριθμος αναζητά το microdata center με τη μέγιστη διαθέσιμη χωρητικότητα και αναθέτει την εργασία σε αυτό. Αυτός ο αλγόριθμος δεν ελέγχει την απόσταση του καταλληλότερου κόμβου από κόμβο στον οποίο φτάνει η εργασία, όπως επίσης δεν ελέγχει την περίπτωση ύπαρξης κόμβου πλησιέστερου στον αρχικό με χωρητικότητα επαρκή για εκτέλεση της εργασίας αλλά όχι τη μέγιστη διαθέσιμη. Ο αλγόριθμος ενδιαφέρεται μόνο για το microdata center με τη μέγιστη δυνατή χωρητικότητα τη δεδομένη στιγμή.

---

##### Algorithm 2 ML (Minimum Load)

---

```

1: procedure LOADBALANCE( $j_i$ , SECURITY LABEL OF  $j_i$ )
2:    $\forall k$ , find  $m_k$  with least load while considering security label of job  $j_i$ .
3:   if (CL( $m_k$ ) > Threshold &  $!l_p$ ) then
4:     Migrate  $j_i$  to  $c$ .
5:   else
6:     Migrate  $j_i$  to  $m_k$ .
7:   end if
8: end procedure

```

---

Εικόνα 4.4: Ψευδοκώδικας του load balancing αλγορίθμου minimum load [7].

Ο αλγόριθμος λαμβάνει ως παραμέτρους μια εργασία και το security label αυτής και αναζητά το microdata center με τη μεγαλύτερη διαθέσιμη χωρητικότητα ενώ ταυτόχρονα ελέγχει και την ιδιωτικότητα ή μη της εργασίας που έχει αναλάβει να αναθέσει. Όταν βρει το κατάλληλο microdata center, ελέγχει το παρόν φορτίο του και στην περίπτωση που το φορτίο που του έχει ανατεθεί είναι μεγαλύτερο από το μέγιστο φορτίο που μπορεί να υποστηρίξει τη δεδομένη στιγμή, τότε η εργασία αποστέλλεται στο cloud datacenter. Στην περίπτωση που το παρόν φορτίο του δεδομένου κόμβου είναι μικρότερο από το μέγιστο δυνατό που μπορεί να υποστηρίξει, τότε έχει σημασία και ο παράγοντας της ετικέτας ασφαλείας γιατί στην περίπτωση που η εργασία είναι private, δεν μπορεί να αποσταλεί στο νέφος. Πρέπει η χωρητικότητα του microdata center να είναι μεγαλύτερη από τη μέγιστη δυνατή που μπορεί να υποστηρίξει αλλά και η ετικέτα ασφαλείας της εργασίας να είναι semiprivate είτε public, προκειμένου να μπορεί να προωθηθεί στο νέφος. Στην περίπτωση που η ετικέτα εργασίας είναι private, ανεξαρτήτως χωρητικότητας του αρχικού microdata center, η εργασία μπαίνει σε ουρά αναμονής και προγραμματίζεται για εκτέλεση όταν θα απελευθερωθούν οι απαιτούμενοι πόροι.

### 4.3.8 Αλγόριθμος Minimum Distance

Ο αλγόριθμος minimum distance [7] αναθέτει μια εργασία σε ένα συγκεκριμένο microdata center-κόμβο ομίχλης και συγκεκριμένα στο πλησιέστερο τοπολογικά κόμβο από το σημείο δημιουργίας ενός αιτήματος εφαρμογής για την εκτέλεση μιας ή περισσότερων εργασιών. Στην περίπτωση αυτή όπου ο κόμβος στον οποίο έχει ανατεθεί η εκτέλεση της εργασίας δεν έχει επαρκή χωρητικότητα διαθέσιμη προκειμένου να εκτελέσει την εργασία, η εργασία μπαίνει στην ουρά αναμονής και αναμένει την απελευθέρωση επαρκούς χωρητικότητας του microdata center στο οποίο έχει ανατεθεί. Όταν η εργασία που καταλαμβάνει τη διαθέσιμη χωρητικότητα του κόμβου απελευθερώσει τη χωρητικότητα αυτή μετά την εκτέλεσή της, ξεκινά η εκτέλεση εργασιών εκ νέου με σειρά προτεραιότητας άφιξης στον κόμβο των εργασιών που βρίσκονται στην ουρά αναμονής.

---

#### Algorithm 3 MD (Minimum Distance)

---

- 1: **procedure** LOADBALANCE( $j_i$ , SECURITY LABEL OF  $j_i$ )
  - 2:    $\forall k$ , find set of  $m_k$ 's with min.  $D_{ik}$  while considering security label of job  $j_i$ .
  - 3:   Select an mdc  $m_k$  from the set.
  - 4:   Migrate  $j_i$  to  $m_k$ .
  - 5: **end procedure**
- 

Εικόνα 4.5: Ψευδοκώδικας του load balancing αλγόριθμου minimum distance [7]

Ο αλγόριθμος λαμβάνει ως παραμέτρους την εργασία προς εκτέλεση και την ετικέτα ασφαλείας της εργασίας αυτής. Στη συνέχεια αναζητά το πλησιέστερο microdata center τοπολογικά στη τοποθεσία δημιουργίας της εργασίας εντός της υλοποίησης και αποστέλλει την εργασία στον πλησιέστερο. Στην προκειμένη περίπτωση παρά το γεγονός ότι λαμβάνεται υπόψιν η παράμετρος της ετικέτας ασφαλείας, δεν επηρεάζεται η διαδικασία αναζήτησης και ανάθεσης της εργασίας. Επίσης, δεν λαμβάνεται υπόψιν και η αναγκαιότητα ύπαρξης διαθέσιμης χωρητικότητας του microdata center ικανής, να καλύψει το φορτίο της εκάστοτε εργασίας. Η μόνη πορεία που ακολουθείται κατά την εκτέλεση του αλγόριθμου είναι η εύρεση του κοντινότερου τοπολογικά κόμβου ομίχλης για την ανάθεση της εργασίας.

### 4.3.γ Αλγόριθμος Minimum Hop Distance and Load

Ο αλγόριθμος [7] δέχεται ως εισόδους μια εργασία και την ετικέτα ασφαλείας αυτής. Ορίζει τον αριθμό αλμάτων ίσο με 1, καθώς ο αλγόριθμος ξεκινά αναζήτηση για microdata centers τα οποία βρίσκονται σε απόσταση ενός άλματος από το microdata center στο οποίο έχει αφιχθεί η εργασία και δεν μπορεί να εξυπηρετηθεί. Ορίζεται ένα flag ως αληθές. Ξεκινά η αναζήτηση microdata center σε απόσταση ενός άλματος ενώ ταυτόχρονα λαμβάνεται υπόψιν και η ετικέτα ασφαλείας της εργασίας.

---

**Algorithm 4** MHDL (Minimum Hop Distance and Load)

---

```
1: procedure LOADBALANCE( $j_i$ , SECURITY LABEL OF  $j_i$ )
2:   Number of hops  $D = 1$ .
3:   Flag=true;
4:   while (Flag) do
5:     Find mdcs that are at distance  $D$  from  $j_i$  while considering security label of job  $j_i$ .
6:     if (There exist some mdcs) then
7:       Select  $m_k$  with least load.
8:       if ( $CL(m_k) > \text{Threshold}$ ) then
9:          $D++$ ;
10:      else Migrate  $j_i$  to  $m_k$ .
11:        Flag=False;
12:      end if
13:    else break;
14:    end if
15:  end while
16:  if (Flag==true & ! $l_p$ ) then
17:    Migrate  $j_i$  to  $c$ .
18:  end if
19: end procedure
```

---

Εικόνα 4.6: Ψευδοκώδικας του load balancing αλγόριθμου minimum hop distance and load [7].

Εάν υπάρχει ένα ή παραπάνω microdata center σε απόσταση ενός άλματος από το αρχικό τότε αναζητείται αυτό με τη μεγαλύτερη χωρητικότητα. Εάν το φορτίο του είναι μεγαλύτερο από το μέγιστο φορτίο που μπορεί να του ανατεθεί χωρίς να υποβαθμιστεί η απόδοση, τότε ο αριθμός των αλμάτων αυξάνεται κατά 1 και ο αλγόριθμος πλέον ξεκινά την αναζήτηση για κόμβους ομίχλης οι οποίοι βρίσκονται σε απόσταση δύο αλμάτων έως ότου να βρεθεί ο κατάλληλος.

Στον αντίποδα αυτού, εάν το microdata center μπορεί να εξυπηρετήσει την εργασία, τότε αυτή ανατίθεται σε αυτό και το flag αλλάζει σε false προκειμένου να γνωστοποιηθεί ότι η εργασία έχει ανατεθεί και δεν υπάρχει ανάγκη περαιτέρω έρευνας. Στην περίπτωση που δεν υπάρχει κανένα microdata center που να μπορεί να εκτελέσει το αίτημα εφαρμογής και ο αριθμός των αλμάτων αυξηθεί εκ νέου κατά ένα αλλά πλέον σε απόσταση  $x$  αλμάτων ( $D=x$ ) δεν υπάρχει κανένα microdata center, τότε έχει λαμβάνεται η απόφαση ότι η εργασία δεν πρόκειται να ανατεθεί σε κάποιο microdata center και προωθείται για επεξεργασία και εκτέλεση στο cloud. Η ετικέτα Flag η οποία στην περίπτωση που δεν υπάρχει microdata center ικανό να εκτελέσει την εργασία παραμένει True και σε νέα συνθήκη if γίνεται έλεγχος επίσης και για την τιμή της ετικέτας ασφαλείας της εργασίας προς εκτέλεση. Στην περίπτωση που η εργασία δεν έχει ετικέτα ασφαλείας private, προωθείται και μπορεί να εκτελεστεί από το νέφος.

#### 4.3.6 Efficient Resource Allocation Αλγόριθμος

Σε υλοποιήσεις νέφους-ομίχλης, η επίτευξη της βέλτιστης κατανομής πόρων είναι ένας από τους βασικούς στόχους για την επιτυχία του συστήματος, καθώς η αποδοτική κατανομή πόρων σημαίνει βελτιωμένη απόδοση συστήματος αλλά και ικανοποίηση των τελικών χρηστών. Στο paper “An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing” [8] των Swati Agarwal, Shashank Yadav και Arun Kumar Yadav, προτείνεται μια αρχιτεκτονική για την επίλυση του προβλήματος που σχετίζεται με την ανοχή σφαλμάτων, υπερχειλίση πόρων και την μειωμένη ροή φορτίου εντός του δικτύου.

Το μοντέλο που προτείνεται στο paper έχει σχεδιαστεί σε περιβάλλον νέφους-ομίχλης τριών επιπέδων (πελάτης-ομίχλη-νέφος). Ο αλγόριθμος που θα προταθεί αργότερα εφαρμόζεται στο επίπεδο ομίχλης για την εξυπηρέτηση αιτημάτων των τελικών χρηστών και στην περίπτωση αυτή μόνο που δεν υπάρχουν διαθέσιμοι πόροι για την εξυπηρέτηση κάποιου αιτήματος το αίτημα προωθείται για εξυπηρέτηση στο επίπεδο του νέφους.

1. Το σύνολο των datacenter τοποθετείται στα επίπεδα ομίχλης και νέφους και περιέχει ένα πλήθος από fog data servers και cloud data centers αντίστοιχα.
2. Κάθε fog data server περιέχει έναν data server manager ο οποίος ελέγχει συνεχώς την διαθεσιμότητα του επεξεργαστή και είναι υπεύθυνος για τη διαχείριση των VMs.
3. Οι τελικοί χρήστες μπορούν να αποστείλουν αιτήματα προς εξυπηρέτηση σε οποιονδήποτε fog data server του επιπέδου ομίχλης και αυτός με τη σειρά του θα εκχωρήσει το αίτημα προς εξυπηρέτηση στον fog server manager.
4. Ο fog server manager επεξεργάζεται το αίτημα υπό τις ακόλουθες συνθήκες:
  - i. Στην περίπτωση που οι απαιτούμενες επεξεργαστικές μονάδες είναι διαθέσιμες στον fog data server που εκχωρείται αρχικά το αίτημα, το αποτέλεσμα αποστέλλεται στον πελάτη ο οποίος απαντά με μια επιβεβαίωση στον fog server manager προκειμένου να ενημερωθεί η κατάσταση της λίστας αιτημάτων.
  - ii. Εάν δεν είναι διαθέσιμος ο απαιτούμενος αριθμός επεξεργαστικών μονάδων του fog data server, τότε το αίτημα εφαρμογής χωρίζεται σε επιμέρους υποδεέστερες εργασίες προκειμένου να ανατεθεί προς επεξεργασία βάσει διαθεσιμότητας.
  - iii. Εάν ο fog data server έχει διαθέσει το σύνολο των μονάδων επεξεργασίας για την εξυπηρέτηση ενός αιτήματος είτε υποδεέστερων εργασιών αλλά βρίσκεται σε κατάσταση επικείμενης απελευθέρωσης, το αίτημα μπαίνει σε κατάσταση αναμονής για ένα χρονικό διάστημα και έπειτα ανατίθεται ξανά στον fog data server και η διαδικασία ελέγχου διαθεσιμότητας των μονάδων επεξεργασίας επαναλαμβάνεται.
  - iv. Εάν το σύνολο των επεξεργαστικών μονάδων ενός fog data server έχει ανατεθεί στην εξυπηρέτηση ενός αιτήματος αλλά η εξυπηρέτηση μιας υποδεέστερης εργασίας αυτού αποτύχει, τότε δεν γίνεται εκ νέου επεξεργασία ολόκληρου του αιτήματος, αλλά μόνο των υποεργασιών αυτού που απέτυχαν να εξυπηρετηθούν.
  - v. Εάν δεν υπάρχει fog data server με διαθέσιμη επεξεργαστική χωρητικότητα στο επίπεδο ομίχλης, τότε το αίτημα προωθείται για εξυπηρέτηση στον cloud data server του επιπέδου υπολογιστικού νέφους.
5. Εάν ο χρήστης που υπέβαλε το αίτημα προς εξυπηρέτηση δεν έχει λάβει αποτέλεσμα εντός του μέγιστου προκαθορισμένου χρόνου, τότε περιμένει για την ολοκλήρωση επεξεργασίας του αιτήματος.
6. Εάν δεν έχει λάβει αποτέλεσμα εντός προκαθορισμένου χρόνου, τότε το αίτημα έχει προωθηθεί στον cloud data server για επεξεργασία και εξαγωγή αποτελεσμάτων.
7. Ο cloud data server παρέχει τις απαιτούμενες επεξεργαστικές μονάδες στον τελικό χρήστη άμεσα προκειμένου να μειωθεί ο χρόνος απόκρισης και στέλνει επιβεβαίωση επιτυχούς ανάθεσης και επεξεργασίας αιτήματος στον fog data server.

Στο παραπάνω μοντέλο υλοποίησης cloud-fog, ο ρόλος του fog server manager είναι η συγκέντρωση διαθέσιμων επεξεργαστικών μονάδων των fog data server για εξυπηρέτηση αιτημάτων, ο ρόλος των VMs είναι ο χειρισμός αιτημάτων που προορίζονται για τον fog data server και η εκχώρηση του αποτελέσματος επεξεργασίας των αιτημάτων στον fog server manager και τέλος ο ρόλος του fog server είναι η ενσωμάτωση ενός fog server manager και ενός αριθμού VM προκειμένου να καθίσταται δυνατή η διαχείριση αιτημάτων με τη χρήση της τεχνικής εικονικοποίησης διακομιστή (server virtualization).

Ο αλγόριθμος που προτείνουν οι Swati Agarwal, Shashank Yadav και Arun Kumar Yadav είναι ο Efficient Resource Allocation και στόχος του είναι η αποδοτική διαχείριση των πόρων και η αποσυμφόρηση του δικτύου.

Request $i_f$	Ένα αίτημα από έναν τελικό χρήστη προς τον fog data server
Request $i_c$	Ένα αίτημα από τον fog data server στον cloud data server
CS <sub>i</sub>	Ο cloud server που επεξεργάζεται ένα αίτημα $i$
FS <sub>i</sub>	Ο fog server που επεξεργάζεται ένα αίτημα $i$
FSM	Ο fog server manager του κάθε fog data server
VM <sub>i</sub>	Το σύνολο των VM ενός fog data server
M <sub>st</sub>	Ο ελάχιστος χρονικός περιορισμός για την απελευθέρωση των επεξεργαστικών μονάδων
Max_time	Ο μέγιστος χρόνος για την απελευθέρωση μιας επεξεργαστικής μονάδας
T <sub>i</sub>	Οι οριακές τιμές κάθε αιτήματος $i_f$
T	Το σύνολο των αιτημάτων-εργασιών προς εκτέλεση
$t_1, t_2, t_3, \dots$	Οι υποδεέστερες εργασίες ενός αιτήματος

Πίνακας συμβόλων κλειδιά για την κατανόηση του ψευδοκώδικα [8]

Ο ψευδοκώδικας του αλγόριθμου Efficient Resource Allocation παρατίθεται παρακάτω ελαφρώς διαφοροποιημένος για λόγους καλύτερης κατανόησης των λειτουργιών:

---

**ERA Algorithm:** Efficient Resource Allocation

---

```

1:   for each request  $i_f$  do
2:       each request  $i_f$  is sent to the nearest fog data server based on user's location
3:       each fog server FS will process the user's request
4:       fog server manager FSM will process the request based on the following conditions
5:       if all requesting processors of the first fog server FS found are available then
6:           fog server loads the client's results and sends acknowledgement to its fog server manager FSM
7:       end if
8:       if only some of the requesting processors are available on the fog server FS, then
9:           the number of tasks T is divided into a number of sub-tasks as per availability
10:           $T = t_1 * t_2 * t_3 * \dots * t_n$ 
11:       end if
12:       if all processors are being used but some of them fail while processing then
13:           goto step 10
14:       if no processor is available on the fog server FS within the current fog cluster, then
15:           request  $i_f$  is propagated to the cloud server CS through network
16:           the cloud server CS will process the service request
17:           cloud server CS forwards the processing results to the client directly
18:           cloud server CS sends an acknowledgement to the respective fog server manager FSM
19:       end if
20:       calculate the threshold Ti
21:       if  $T_i \leq \text{Max\_time}$ , then
22:           client receives a message to wait for processing to end
23:       end if
24:   end for

```

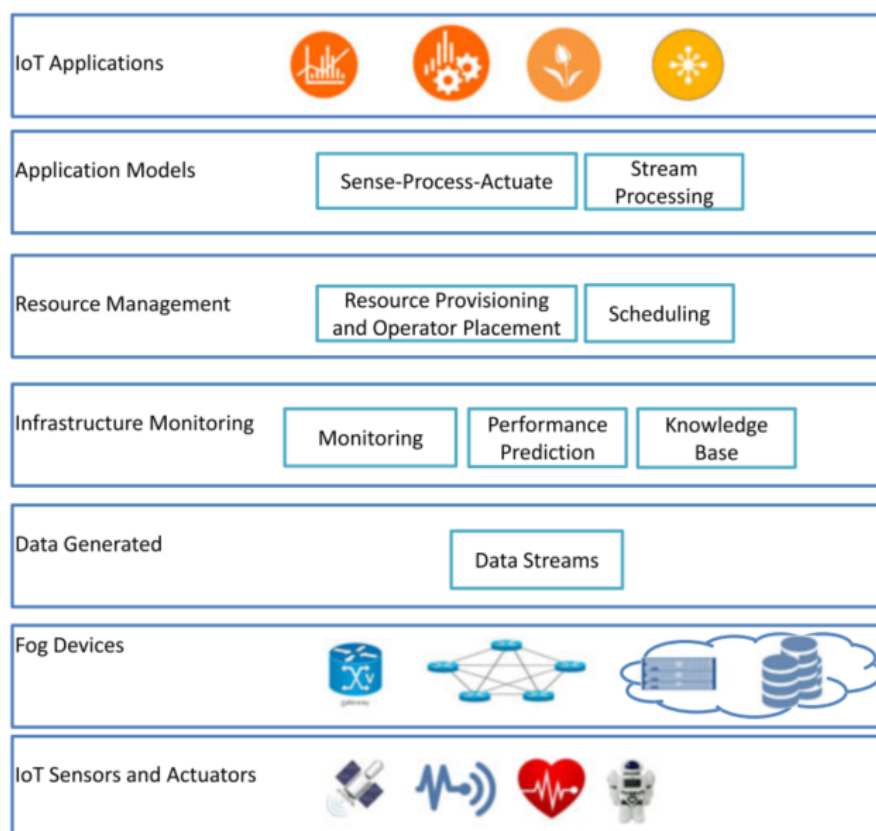
---

Εικόνα 4.8: Ψευδοκώδικας του Efficient Resource Allocation αλγόριθμου [8]

## ΚΕΦΑΛΑΙΟ 5: Εργαλείο iFogSim

### 5.1 Αρχιτεκτονική προσομοιωτή

Η αρχιτεκτονική του περιβάλλοντος διάφορων σεναρίων υλοποίησης συστημάτων υπολογιστικής ομίχλης αποτελείται από μια σειρά επιπέδων, εκ των οποίων κάθε ένα είναι υπεύθυνο για μια σειρά από διαδικασίες και διεργασίες προκειμένου να επιτυγχάνεται ομαλή λειτουργία σε κάθε επίπεδο αλλά και μέσω αυτής να επιτυγχάνεται επίσης ευκολότερη, ταχύτερη και αποτελεσματικότερη λειτουργία σε κάθε ένα από τα ανώτερα επίπεδα.



Εικόνα 5.1: Αρχιτεκτονική υπολογιστικής ομίχλης στο εργαλείο iFogSim (iFogSim [6])

**IoT Sensors and Actuators:** Στο κατώτατο επίπεδο ενός συστήματος υπολογιστικής ομίχλης στο εργαλείο προσομοίωσης iFogSim τοποθετούνται οι συσκευές-«αντικείμενα» IoT, όπως αισθητήρες IoT οι οποίοι είναι στρατηγικά κατανομημένοι προκειμένου να ανιχνεύουν μεταβολές στο περιβάλλον και να παράγουν πληροφορία η οποία προωθείται στα ανώτερα επίπεδα για επεξεργασία, είτε ενεργοποιητές οι οποίοι βάσει των πληροφοριών που λαμβάνονται από το περιβάλλον, εφόσον αυτές έχουν επεξεργαστεί, ανταποκρίνονται σε μεταβολές με τρόπο ο οποίος έχει προκαθοριστεί, αποφασίζεται δυναμικά από το σύστημα έπειτα από επεξεργασία της πληροφορίας και ανάλυση των δεδομένων που εκρέουν από αυτή ή ορίζεται χειροκίνητα από τη μονάδα είτε ομάδα διαχείρισης των συσκευών κατώτερου επιπέδου.

Στο εργαλείο προσομοίωσης iFogSim ως αισθητήρας ορίζεται η φυσική συσκευή η οποία χαρακτηρίζεται από συγκεκριμένες ιδιότητες εκπομπής δεδομένων, οι οποίες ενέχουν μια ευελιξία βάσει της οποίας μπορούν να προσαρμοστούν σε διαφορετικά σενάρια υπολογιστικής ομίχλης και να συμμετέχουν στην προσομοίωση συσκευών IoT, οι οποίες χαρακτηρίζονται από ετερογένεια τόσο σε υλικό όσο και σε λογισμικό, που μπορούν να συλλέξουν πληροφορία υπό τη μορφή δεδομένων και να την επικοινωνήσουν. Παρομοίως, ως ενεργοποιητής ορίζεται η φυσική συσκευή η οποία χαρακτηρίζεται από συγκεκριμένες ιδιότητες και μπορεί να χρησιμοποιηθεί για την προσομοίωση της διάδρασης μεταξύ του συστήματος και των τελικών συσκευών βάσει των αποτελεσμάτων επεξεργασίας της πληροφορίας στα ανώτερα επίπεδα της αρχιτεκτονικής.

Το εργαλείο iFogSim κατά τη διάρκεια της προσομοίωσης δεν επηρεάζεται από ζητήματα δικτύου που υπάρχουν σε φυσικές υλοποιήσεις και δε δίνει τη δυνατότητα παραμετροποίησης των παραγόντων που σε μια φυσική υλοποίηση θα κάνανε πιο αποδοτική και αξιόπιστη τη λειτουργία του συστήματος. Ως εκ τούτου κατά τη διάρκεια της προσομοίωσης, τα ζητήματα αυτά ανάγονται ως χαρακτηριστικά στα ανώτερα επίπεδα της αρχιτεκτονικής τα οποία θα αναλύσουμε παρακάτω. Μέσω της δημιουργίας ευέλικτων προφίλ των χαρακτηριστικών, ασύρματων και μη, συσκευών IoT, δίνεται η δυνατότητα δημιουργίας ενός συμπεριφορικού μοντέλου φυσικού επιπέδου τέτοιων συσκευών το οποίο μπορεί να ενσωματώνεται κατάλληλα σε διαφορετικά σενάρια Cloud-Fog-IoT στο iFogSim και να δώσει αποτελέσματα μέσω προσομοίωσης τα οποία αντικατοπτρίζουν αυτά των αντίστοιχων φυσικών συσκευών σε μια υλοποίηση.

**Fog Devices:** Ως συσκευές υπολογιστικής ομίχλης ορίζονται οι φυσικές συσκευές εντός του δικτύου οι οποίες μπορούν να υποστηρίξουν μια σειρά από εφαρμογές. Οι συσκευές αυτές που αποτελούν μεσάζοντες μεταξύ αισθητήρων και του δικτύου ονομάζονται πύλες και αποτελούν το φυσικό μέσο τροποποίησης των δεδομένων και μεταμόρφωσης αυτών σε πληροφορία η οποία μπορεί να γίνει κατανοητή και να χρησιμοποιηθεί από τα ανώτερα επίπεδα της αρχιτεκτονικής. Οι συσκευές υπολογιστικής ομίχλης περιλαμβάνουν πόρους υπολογιστικών νεφών οι οποία παρέχονται κατ' απαίτηση από γεωγραφικά κατανομημένα κέντρα δεδομένων. Στο συγκεκριμένο επίπεδο της αρχιτεκτονικής περιλαμβάνεται το σύνολο των πόρων που παρεμβάλλεται μεταξύ τελικών συσκευών και υπολογιστικού νέφους. Οι συσκευές είναι διατεταγμένες σε μια ιεραρχική τοπολογία με δυνατότητα άμεσης επικοινωνίας μεταξύ ενός ζεύγους γονέα-παιδιού στην ιεραρχία. Οι εφαρμογές που εκτελούνται σε μια συσκευή ομίχλης είναι υπεύθυνες για την επεξεργασία των ροών δεδομένων που παράγονται από συσκευές-«αντικείμενα» τα οποία βρίσκονται στο ιεραρχικά κατώτερο επίπεδο.

Το εργαλείο iFogSim βασίζεται στον ορισμό που θέλει το fog computing να είναι μια υποδομή με χαρακτηριστικά παρόμοια με αυτά του υπολογιστικού νέφους με την ειδοποιό διαφορά ότι τοποθετείται και μεταφέρει σημαίνουσες επεξεργαστικές και όχι μόνο δυνατότητες πλησίον των τελικών συσκευών. Λόγω έλλειψης ιεραρχικής οργάνωσης των συσκευών ομίχλης δεν υποστηρίζεται η επικοινωνία μεταξύ συσκευών, η τοποθέτηση των εφαρμογών πραγματοποιείται μόνο με κατεύθυνση από τα κατώτερα προς τα ανώτερα επίπεδα μόνο και δεν είναι δυνατή ακόμη η μεταφόρτωση τμημάτων δεδομένων είτε εφαρμογών μεταξύ συσκευών στο ίδιο επίπεδο ιεραρχίας.

**Data Generated:** Το ανώτερο επίπεδο από αυτό των συσκευών ομίχλης είναι το επίπεδο ροής δεδομένων και αποτελείται από τα δεδομένα που παράγονται από τις συσκευές των κατώτερων επιπέδων. Τα μπλοκ δεδομένων ενδέχεται να παράγονται από αισθητήρες, στην οποία περίπτωση πρόκειται για ακατέργαστα δεδομένα που καθρεφτίζουν τις μεταβολές στο περιβάλλον, να μεταδίδονται από και προς συγκεκριμένες μονάδες εφαρμογών ή να μεταδίδονται από μια μονάδα εφαρμογής σε ενεργοποιητές. Οι ροές δεδομένων μπορεί επίσης να δημιουργούνται από συσκευές ομίχλης υπό τη μορφή λεπτομερειών που αφορούν την χρήση πόρων, και προωθούνται ως ροές δεδομένων στο αμέσως επόμενο στην αρχιτεκτονική επίπεδο της παρακολούθησης προκειμένου να προκύψει μια γενική εικόνα για την κατάσταση των συσκευών των κατώτερων επιπέδων.

**Infrastructure Monitoring:** Το επίπεδο παρακολούθησης είναι υπεύθυνο για τον συνεχή έλεγχο της χρήσης πόρων, την παρακολούθηση κατανάλωσης ενέργειας κατά τις περιόδους λειτουργίας του συστήματος και τον έλεγχο διαθεσιμότητας των αντικειμένων και συσκευών που ανήκουν στα κατώτερα επίπεδα της αρχιτεκτονικής. Για λόγους που εξυπηρετούν την απλότητα του σχεδιασμού σεναρίων προς προσομοίωση σε περιβάλλον iFogSim, υπάρχουν κάποια στοιχεία του παρόντος επιπέδου σε φυσικό επίπεδο που δεν έχουν συμπεριληφθεί στον προσομοιωτή. Προκειμένου να υπάρχει ρεαλισμός κατά την προσομοίωση, τα στοιχεία που δεν έχουν συμπεριληφθεί από το εργαλείο μπορούν να ενσωματωθούν στα σενάρια προσομοίωσης δημιουργώντας οντότητες οι οποίες επεξεργάζονται τα στατιστικά στοιχεία χρήσης πόρων που παράγονται και εκπέμπονται από τις συσκευές ομίχλης καθώς και παραμετροποιημένα μηνύματα τα οποία εκπέμπονται από όλες τις οντότητες υπολογιστικής ομίχλης και δηλώνουν την διαθεσιμότητα ή μη της εκάστοτε οντότητας.

**Resource Management:** Η διαχείριση πόρων είναι το βασικότερο συστατικό της αρχιτεκτονικής και το επίπεδο διαχείρισης πόρων αποτελείται από τα στοιχεία που διαχειρίζονται τους πόρους του κατώτερου επιπέδου των συσκευών ομίχλης. Το επίπεδο αυτό είναι υπεύθυνο για τη διαχείριση των πόρων με τρόπο τέτοιο που να μην γίνονται εκπτώσεις σε θέματα που αφορούν τα υψηλά στάνταρ της ποιότητας υπηρεσιών αλλά ταυτόχρονα μεριμνά και για την ελαχιστοποίηση άσκοπης σπατάλης πόρων. Οι πληροφορίες που λαμβάνονται από τις συσκευές και τα αντικείμενα των κατώτερων επιπέδων λαμβάνονται από το παρόν επίπεδο μέσω του επιπέδου παρακολούθησης και μέσω διαφόρων στοιχείων τοποθέτησης και προγραμματισμού παρακολουθείται η κατάσταση των διαθέσιμων πόρων προκειμένου να εντοπίζονται οι οντότητες που μπορούν να φιλοξενήσουν μια συγκεκριμένη ενότητα εφαρμογής και να κατανέμουν τους διαθέσιμους πόρους της συσκευής στις εν λόγω ενότητες εφαρμογών αποτελεσματικά. Το συγκεκριμένο επίπεδο λειτουργεί βάσει των απαιτήσεων ποιότητας υπηρεσιών, που διαφέρουν από μια υλοποίηση σε μια άλλη και ορίζονται από το επίπεδο εφαρμογής, με τρόπο τέτοιο ώστε κάθε μια από τις συνιστώσες της εκάστοτε εφαρμογής να μπορεί να ανέλθει στο επίπεδο της ποιότητας υπηρεσιών που απαιτείται. Η πολιτική διαχείρισης πόρων μπορεί να είναι αρκετά περίπλοκη ώστε να επιτρέπει την μετεγκατάσταση στοιχείων και δυναμικών αλλαγών στην κατανομή των πόρων συσκευής σε στοιχεία ή τόσο απλή ώστε να επιτρέπει τη στατική παροχή στοιχείων σε μια συσκευή ομίχλης. Οι υλοποιήσεις του επιπέδου διαχείρισης πόρων μπορεί να επιτρέπουν τη διαχείριση των πόρων της εκάστοτε συσκευής από τη ίδια χωρίς να υπάρχει γνώση πέραν αυτής για τον τρόπο αυτό, τη συγκέντρωση σε μια κεντρική οντότητα των πληροφοριών που αφορούν τη διαχείριση πόρων της εκάστοτε συσκευής, είτε την εφαρμογή μιας ενδιάμεσης λύσης κατά



την οποία συνυπάρχουν συσκευές που διαχειρίζονται αυτόνομα τους πόρους αυτών καθώς και συσκευές οι οποίες ενημερώνουν έναν ή περισσότερους κόμβους διαχείρισης πόρων.

Στην παρούσα έκδοση του εργαλείου, παρέχεται μια πολιτική στατικής τοποθέτησης των εφαρμογών με τις μονάδες εφαρμογών να εκχωρούνται στατικά σε συσκευές ομίχλης και η πολιτική αυτή μπορεί να αντικατασταθεί από δυναμικές πολιτικές που μπορούν να μετεγκαταστήσουν μονάδες εφαρμογών σε άλλες συσκευές ομίχλης δυναμικά με βάση κριτήρια όπως η κατανάλωση ενέργειας και ο λανθάνων χρόνος.

**Application Models:** Οι εφαρμογές που δημιουργήθηκαν για υλοποίηση σεναρίων υπολογιστικής ομίχλης βασίζονται στο μοντέλο της κατανομημένης ροής δεδομένων (Distributed Data Flows - DDF). Μια εφαρμογή μοντελοποιείται ως μια συλλογή μονάδων εφαρμογής, οι οποίες αποτελούν τα στοιχεία επεξεργασίας δεδομένων. Τα δεδομένα που παράγονται ως έξοδοι από μια μονάδα εφαρμογής μπορούν να χρησιμοποιηθούν ως είσοδοι από μια άλλη μονάδα εφαρμογής, προκαλώντας εξάρτηση δεδομένων μεταξύ των δύο μονάδων εφαρμογής.

Αυτό το μοντέλο εφαρμογής μας επιτρέπει την αναπαράσταση μιας εφαρμογής με τη μορφή κατευθυνόμενου γραφήματος, όπου οι κορυφές αντιπροσωπεύουν μονάδες εφαρμογής και οι κατευθυνόμενες ακμές δείχνουν τη ροή δεδομένων μεταξύ των μονάδων.

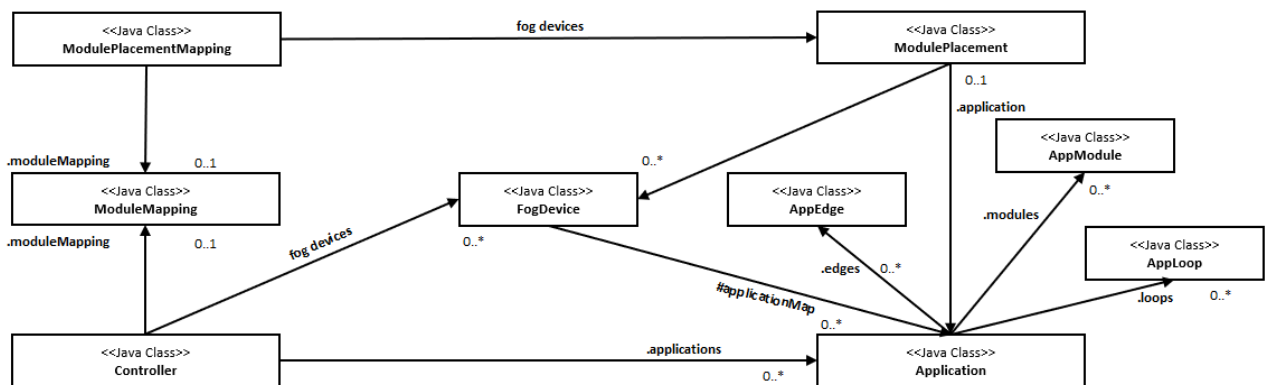
**IoT Applications:** Μια από τις κυριότερες ανάγκες που οδήγησαν στη δημιουργία της υπολογιστικής ομίχλης ήταν η ανάγκη για επεξεργασία δεδομένων και απόκριση συστήματος βάσει αναλυτικών στοιχείων σε πραγματικό χρόνο σε συνάρτηση με την κλιμάκωση της επέκτασης του IoT. Οι υλοποιήσεις συστημάτων IoT χρησιμοποιούν αισθητήρες ως μέσο παρακολούθησης του περιβάλλοντος και παραγωγής πληροφορίας που το αφορούν και στην πλειοψηφία τους ενσωματώνονται σε πλειάδες. Η αρχιτεκτονική του iFogSim υποστηρίζει για εφαρμογές IoT τα παρακάτω μοντέλα:

- **Μοντέλο Sense-Process-Actuate:** Πληροφορίες για μεταβολές στο περιβάλλον συλλέγονται από τους αισθητήρες και μεταδίδονται ως ροές δεδομένων για επεξεργασία και εξαγωγή αναλυτικών στοιχείων από εφαρμογές που εκτελούνται στις συσκευές ομίχλης. Μετά την εξαγωγή αποφάσεων για το περιβάλλον του συστήματος, οι εντολές για την επίτευξή τους αποστέλλονται στους ενεργοποιητές.
- **Μοντέλο Stream-Processing:** Στο μοντέλο υπάρχει ένα δίκτυο λειτουργικών μονάδων εφαρμογών που εκτελούνται στις συσκευές υπολογιστικής ομίχλης και επεξεργάζονται συνεχώς πληροφορία υπό τη μορφή ροών δεδομένων που παράγονται και εκπέμπονται από τους αισθητήρες. Οι πληροφορίες που εξάγονται μετά την επεξεργασία των δεδομένων αποθηκεύονται σε datacenter προκειμένου να χρησιμοποιηθούν μακροπρόθεσμα για μεγάλης κλίμακας αναλύσεις.

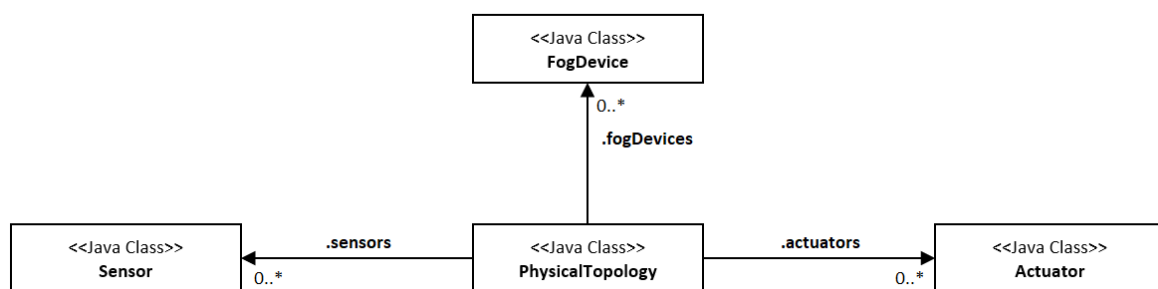
Παρά το γεγονός ότι θεωρούμε το δεύτερο μοντέλο ως υποκατηγορία του αρχικού, αμφότερα μπορούν να χρησιμοποιηθούν προκειμένου να καλύψουν μια πληθώρα διαφορετικών περιπτώσεων και σεναρίων υλοποίησης συστημάτων που χρησιμοποιούν εφαρμογές IoT.

## 5.2 Σχεδιασμός και υλοποίηση

Ο iFogSim αξιοποιεί βασικές λειτουργίες προσομοίωσης συμβάντων που βρίσκονται στο CloudSim. Στις εικόνες που ακολουθούν φαίνονται οι στοιχειώδεις κλάσεις του iFogSim, οι οντότητες καθώς και η φυσική τοπολογία των κλάσεων αυτών.



Εικόνα 5.2: Στοιχειώδεις κλάσεις του iFogSim [6]



Εικόνα 5.3: Φυσική τοπολογία των κλάσεων του iFogSim [6]

Οι κλάσεις του iFogSim είναι γραμμένες με java και είναι οι εξής:

**FogDevice (συσσκευή υπολογιστικής ομίχλης):** Στην κλάση αυτή ορίζονται τα χαρακτηριστικά του hardware της συσκευής καθώς και οι πιθανές σχέσεις και εξαρτήσεις που ενδέχεται να υπάρχουν μεταξύ αυτών και άλλων συσκευών, αισθητήρων είτε ενεργοποιητών. Οι μέθοδοι αυτής της κλάσης ορίζουν τον τρόπο με τον οποίο προγραμματίζονται οι πόροι μιας συσκευής ομίχλης μεταξύ των λειτουργικών μονάδων εφαρμογών που εκτελούνται σε αυτήν και πώς οι μονάδες αναπτύσσονται και ολοκληρώνουν την εκτέλεσή τους σε αυτές.

**Sensor (Αισθητήρας):** Η κλάση περιέχει χαρακτηριστικά που αντιπροσωπεύουν αυτά ενός αισθητήρα και ως οντότητα ο Sensor και η κλάση που τον αναπαριστά συνδέεται με συσκευή ομίχλης. Η κλάση ορίζει τα χαρακτηριστικά εξόδου του αισθητήρα και την κατανομή των ροών tuples.

**Actuator (Ενεργοποιητής):** Αυτή η κλάση μοντελοποιεί έναν ενεργοποιητή ορίζοντας τις ιδιότητες σύνδεσης δικτύου και την εκτέλεση μιας ενέργειας κατά την άφιξη μιας ροής tuple από μια λειτουργική μονάδα κάποιας εφαρμογής.

**Tuple (ακολουθία πληροφοριών):** Τα tuples αποτελούν μια θεμελιώδεις μονάδες επικοινωνίας μεταξύ των fog οντοτήτων και καθιστούν δυνατή τη ροή δεδομένων και πληροφορίας εντός της αρχιτεκτονικής.

**Application (Εφαρμογή):** Ο σχεδιασμός μιας εφαρμογής στον iFogSim ακολουθεί το μοντέλο DDF (Directed Data Flow), κατά το οποίο η εφαρμογή μοντελοποιείται ως κατευθυνόμενος γράφος, με τις κορυφές του ακυκλικού αυτού γράφου (Directed Acyclic Graph) να αντιπροσωπεύουν μονάδες επεξεργασίας εισερχόμενων δεδομένων και ακμές ορισμού εξαρτήσεων δεδομένων μεταξύ των μονάδων επεξεργασίας. Οι οντότητες που εκτελούν τις παραπάνω λειτουργίες αναπαρίστανται από τις παρακάτω κλάσεις:

- **AppModule:** Η κλάση είναι υπεύθυνη για την υλοποίηση των κορυφών του παραπάνω γράφου και αντιπροσωπεύει τα στοιχεία επεξεργασίας των εφαρμογών. Η άφιξη ενός εισερχόμενου tuple εκκινεί την επεξεργασία και τη δημιουργία tuple εξόδου που αποστέλλονται στον γράφο. Ο αριθμός των tuple εξόδου για κάθε ένα που καταφθάνει προς επεξεργασία είναι μεταβλητός και μπορεί να αποφασιστεί μέσω ενός μοντέλου επιλεκτικότητας.
- **AppEdge:** Μέσω της κλάσης υποδηλώνεται η εξάρτηση δεδομένων μεταξύ των λειτουργικών μονάδων και ορίζεται ένα κατευθυνόμενο άκρο στο μοντέλο DDF. Κάθε κορυφή χαρακτηρίζεται από τον τύπο tuple που μεταφέρει. Ο iFogSim υποστηρίζει δύο τύπους ακμών εφαρμογών: περιοδικές ακμές και ακμές γεγονότων. Τα tuples εκπέμπονται σε τακτικά διαστήματα και αυτά που βασίζονται σε ακμές γεγονότων και συγκεκριμένα σε συμβάν  $e=(u,v)$  αποστέλλονται όταν το module της πηγής (u) λάβει ένα tuple προς επεξεργασία και βάσει μοντέλου επιλεκτικότητας επιτρέπεται η εκπομπή tuple που εκπέμπονται από την κορυφή e.
- **AppLoop:** Μια πρόσθετη κλάση που διυλίζει τους βρόγχους ελέγχου διεργασίας και επιστρέφει αυτούς που ενδιαφέρουν τον χρήστη. Αναπαριστά μια λίστα από modules εφαρμογών η προέλευση των οποίων έρχεται από την αρχή εκτέλεσης του βρόγχου μέχρι και την ολοκλήρωσή του.

Πέραν των λειτουργιών στις οποίες αναφερθήκαμε παραπάνω, οι οποίες αφορούν την επεξεργασία των ρών tuple, ο iFogSim παρέχει την προσομοίωση επιπλέον υπηρεσιών. Οι υπηρεσίες αυτές είναι:

- **Monitoring service (υπηρεσία παρακολούθησης):** Μέσω της μεθόδου executeTuple(), η οποία εμφανίζεται στην κλάση FogDevice, δίνεται η δυνατότητα επεξεργασίας ακολουθιών πληροφορίας ενώ παράλληλα η συσκευή στην οποία συμβαίνει η διαδικασία επεξεργασίας των δεδομένων παρέχει συνεχή ενημέρωση όσον αφορά τους διαθέσιμους και κατειλημμένους πόρους ανά πάσα στιγμή. Τα στατιστικά στοιχεία που προκύπτουν από την επεξεργασία των δεδομένων μπορούν να ενθυλακωθούν εντός ενός tuple και να αποσταλούν στο επίπεδο Resource Management που αναφέρθηκε παραπάνω προκειμένου να γίνει επιλογή της πολιτικής διαχείρισης που εξυπηρετεί καλύτερα την κάθε υλοποίηση. Οι παραπάνω ακολουθίες πληροφορίας μπορούν να αξιοποιηθούν από το χρήστη ως αντικείμενο μελέτης της επίδοσης εκτέλεσης των εφαρμογών και μπορούν να αποθηκευτούν τοπικά για να μελετηθούν ακόμη και όταν δεν υπάρχει διαθέσιμο δίκτυο. Τα δεδομένα που παράγονται και αφορούν τη χρήση πόρων αποτελούν παράγοντες στον υπολογισμό της κατανάλωσης ενέργειας της κάθε συσκευής.

- **Resource Management Service (υπηρεσία διαχείρισης πόρων):** Το εργαλείο iFogSim διαθέτει δύο επίπεδα διαχείρισης πόρων για την εκτέλεση αιτημάτων εφαρμογών, οι οποίοι διαχωρίζονται ως δύο ανεξάρτητες πολιτικές διαχείρισης που καθιστούν το fog computing ευέλικτο, ικανό να επεκτείνεται και να τροποποιείται βάσει αναγκών των χρηστών και του συστήματος. Οι δύο πολιτικές είναι:
  - 1 . **Application Placement:** Η πολιτική ορίζει τον τρόπο τοποθέτησης αιτημάτων εφαρμογής (modules) στις καταλληλότερες συσκευές ομίχλης, τη χρονική στιγμή υποβολής του αιτήματος προς επεξεργασία.
  - 2 . **Application Scheduling:** Ο scheduler των πόρων κατανέμει ανάλογα και παράλληλα με τον καταλληλότερο τρόπο τους πόρους που διαθέτει μια συσκευή σε ενεργά και προς εξυπηρέτηση module εφαρμογών.

### 5.3 Ομοιότητες iFogSim και CloudSim

---

Το εργαλείο iFogSim αποτελεί μια υλοποίηση η οποία είναι επέκταση του CloudSim και κληρονομεί πληθώρα χαρακτηριστικών από αυτό. Τα στοιχεία στα οποία εμφανίζεται ομοιότητα είναι:

- **Ταυτόχρονη εκτέλεση πολλαπλών εφαρμογών:** Στον iFogSim είναι δυνατή η ταυτόχρονη εκτέλεση εφαρμογών στην υποδομή. Μέσω των διαθέσιμων κλάσεων δίνεται η δυνατότητα για δημιουργία συσκευών με ετερογενή αρχιτεκτονικά χαρακτηριστικά ακόμη και στο ίδιο επίπεδο ιεραρχίας (UPPER-cloud, MIDDLE-fog, LOWER-edge) και κώδικας που μπορεί να παραμετροποιηθεί και να διαμορφώσει διαφοροποιημένες συμπεριφορές εφαρμογών και πολιτικών τοποθέτησης.
- **Δυνατότητα μετεγκατάστασης module εφαρμογών:** Όπως και η κλάση AppModule του CloudSim, έτσι και στον iFogSim υποστηρίζεται μέσω αυτής η μετεγκατάσταση των module από μια συσκευή ομίχλης σε μια άλλη. Οι συσκευές που εμπλέκονται στη διαδικασία διατηρούν πληροφορίες που την αφορούν και κατευθύνουν την κίνηση στη συσκευή μετεγκατάστασης.
- **Διασυνδεδεμένες πολιτικές διαχείρισης πόρων:** Οι διάφορες πολιτικές διαχείρισης πόρων στον iFogSim μπορούν να ενσωματωθούν ως προαιρετικές συνιστώσες που μπορούν να μην χρησιμοποιηθούν κατά την επεξεργασία αιτημάτων, πράγμα το οποίο επιτρέπει την επανάληψη εκτέλεσης του εκάστοτε σεναρίου υλοποίησης, κάθε φορά με χρήση διαφορετικής πολιτικής διαχείρισης πόρων.

## ΚΕΦΑΛΑΙΟ 6: Προσομοίωση

Η προσομοίωση αφορά μια υλοποίηση τεσσάρων διαφορετικών σεναρίων τοποθεσία συσκευών cloud, κόμβων υπολογιστικής ομίχλης και τελικών συσκευών υπό τη μορφή αισθητήρων και actuators. Οι συσκευές που θα δημιουργηθούν στο κομμάτι του κώδικα, έχουν χαρακτηριστικά κόμβου υπολογιστικής ομίχλης, καθώς είναι η κύρια οντότητα του προσομοιωτή και παραμετροποιώντας τα χαρακτηριστικά αυτής μπορούμε να δημιουργήσουμε συσκευές με αρχιτεκτονικά χαρακτηριστικά όμοια ενός νέφους, ενός κόμβου ομίχλης είτε μιας τελικής συσκευής. Στην περίπτωση της τελικής συσκευής, θα μπορούσαμε να πούμε ότι πρόκειται για μια edge συσκευή η οποία διαθέτει επεξεργαστική ικανότητα σε κάποιο βαθμό και μπορεί να εξυπηρετήσει τμηματικά είτε εξ ολοκλήρου ένα αίτημα εφαρμογής (app module).

Σκοπός της προσομοίωσης είναι να εκτελεστεί για δεδομένο κύκλο εκτέλεσης διαμοιρασμός των αιτημάτων εφαρμογών, ανάλογα πάντα με τους αλγόριθμους προς προσομοίωση, στις κατάλληλες συσκευές προκειμένου να επιτευχθεί βέλτιστη κατανομή μεταξύ τελικών συσκευών, κόμβων και στην περίπτωση αυξημένων αναγκών είτε υπερφόρτωσης τελικών συσκευών και κόμβων ιεραρχίας ομίχλης, υπολογιστικού νέφους.

Στα σενάρια που θα υλοποιήσουμε δημιουργούμε μια συσκευή με αυξημένη επεξεργαστική ισχύ και χαρακτηριστικά σε συνάρτηση και με αυτές που βρίσκονται σε νοτιότερα επίπεδα από αυτή, το νέφος, στο επίπεδο 0. Έπειτα δημιουργούμε μια σειρά από συσκευές-κόμβους ομίχλης στο αμέσως επόμενο επίπεδο, το επίπεδο 1, οι οποίες έχουν επεξεργαστική ισχύ σημαντικά μειωμένη σε συνάρτηση με το νέφος και παρέχουν δυνατότητα επεξεργασίας πιο κοντά στις τελικές συσκευές. Στο τελευταίο επίπεδο, το επίπεδο 2, δημιουργούμε τελικές συσκευές οι οποίες παρόλα αυτά έχουν χαρακτηριστικά κόμβων υπολογιστικής ομίχλης και μπορούν να εξυπηρετήσουν τμηματικά αιτήματα εφαρμογών. Σε κάθε συσκευή αντιστοιχείται ένας αισθητήρας (sensor) και μια συσκευή ενεργοποιητής (actuator), από τις οποίες παράγονται random ροές δεδομένων και στις οποίες αποστέλλονται εντολές για δράση στο τοπικό περιβάλλον αντίστοιχα. Οι κόμβοι του πρώτου επιπέδου αποτελούν και gateway τελικών συσκευών προς τις υπηρεσίες και εφαρμογές του νέφους.

Οι αλγόριθμοι που θα προσομοιώσουμε είναι οι data placement αλγόριθμοι **Cloud-Only** και ο **Edge-wards** [6]. Ο πρώτος προωθεί τα αιτήματα εφαρμογής προς εξυπηρέτηση στο νέφος όπου υπάρχει πλασματικά απεριόριστη χωρητικότητα και επεξεργαστική ισχύς από την οπτική του τελικού χρήστη και μετά από την εξυπηρέτηση των αιτημάτων, δεδομένα και αποτελέσματα προωθούνται στην πηγή παραγωγής για εξυπηρέτηση τοπικών αναγκών και παραμετροποίηση τοπικού περιβάλλοντος. Ο δεύτερος ακολουθεί την οπτική που θέλει την επεξεργασία αιτημάτων και εξαγωγή δεδομένων να συμβαίνει πάντα όσο το δυνατό πιο κοντά στον τελικό χρήστη και από συσκευές που ανήκουν στα χαμηλότερα επίπεδα ιεραρχίας συσκευών.

Για τα σενάρια μας ορίζουμε ως GW τις συσκευές υπολογιστικής ομίχλης και ως E τις τελικές συσκευές, των οποίων λειτουργίες και ιδιότητες αναφέραμε παραπάνω.

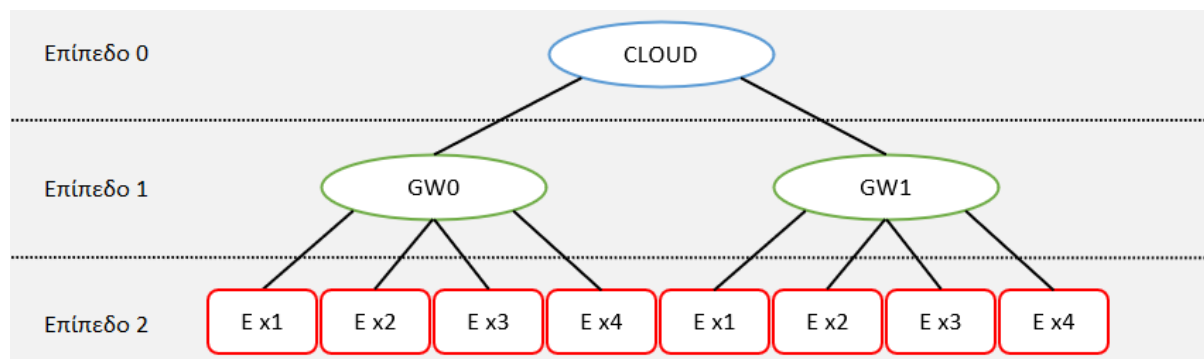
## 6.1 Πειράματα προσομοίωσης

Για να είναι ευκόλως εννοούμενη η συσκευή στην οποία αναφερόμαστε κάθε φορά, αυτές χαρακτηρίζονται από τη γραφή E-γ-z, όπου γ ο αριθμός της συσκευής ομίχλης βόρεια αυτής και z ο αριθμός της συσκευής. Εάν θέλουμε να κάνουμε μια αναφορά στη συσκευή E1 που βρίσκεται υπό τη συσκευή GW0, αναφερόμαστε στη συσκευή E-0-1.

Προκειμένου να επιτευχθεί ένα ευρύτερο σύνολο αποτελεσμάτων για γενικευμένες παρατηρήσεις, τα αποτελέσματα που θα παραχθούν θα έχουν ως βάση δύο κεντρικές ιδέες. Στις υλοποιήσεις που θα παραθέσουμε, θα διατηρήσουμε δύο διαφορετικές βάσεις τοπολογίας και θα δούμε αποτελέσματα για μεταβολές συγκεκριμένων στοιχείων εντός της κάθε υλοποίησης και συγκεκριμένα μεταβολές που αφορούν τον αριθμό των κόμβων ομίχλης είτε τον αριθμό τελικών συσκευών ανά κόμβο ομίχλης. Στο πρώτο πείραμα θα εξετάσουμε τοπολογίες με μοναδικό cloud, μεταβλητό αριθμό τελικών συσκευών ανά κόμβο ομίχλης αλλά σταθερό αριθμό κόμβων ομίχλης. Στο δεύτερο πείραμα θα εξετάσουμε τοπολογίες στις οποίες ενσωματώνουμε μοναδικό cloud, διατηρούμε σταθερό αριθμό τελικών συσκευών ανά κόμβο και μεταβάλλουμε τον αριθμό των κόμβων ομίχλης.

### 6.1.α Πείραμα 1<sup>ο</sup>

Στο πρώτο πείραμα θα δημιουργήσουμε ένα σύνολο τεσσάρων τοπολογιών δικτύου στις οποίες το πλήθος των κόμβων ομίχλης στο ενδιαμέσο επίπεδο μεταξύ cloud και edge παραμένει σταθερό και ίσο με δύο.



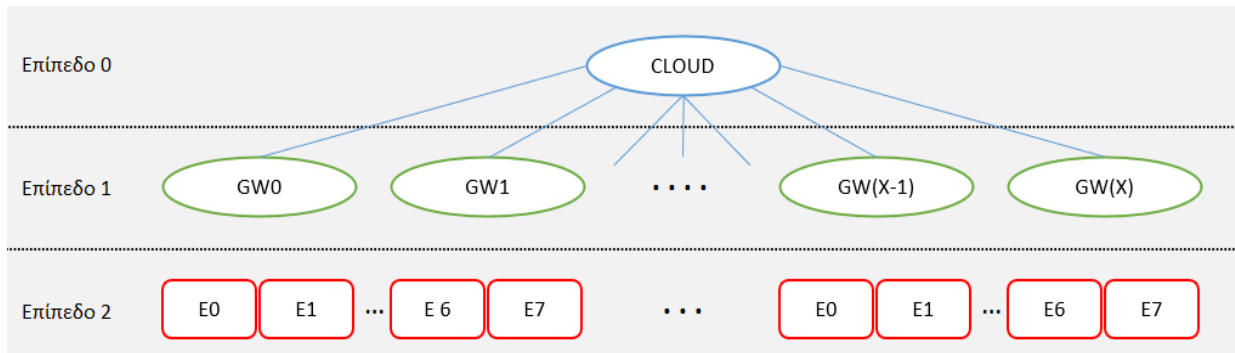
Όπως βλέπουμε παραπάνω υπάρχει μοναδικό cloud και δύο gateways (GW0, GW1) τα οποία αποτελούν κόμβους ομίχλης στο επίπεδο 1 της υλοποίησης. Για κάθε τοπολογία που παράγουμε μεταβάλλουμε τον αριθμό των τελικών συσκευών και διατηρούμε το πλήθος των συσκευών των επιπέδων 0 και 1 σταθερό. Δημιουργούμε και εξετάζουμε τέσσερις διαφορετικές τοπολογίες οι οποίες ενσωματώνουν για κάθε κόμβο ομίχλης από τους δύο, πλήθος τελικών συσκευών ίσο με 4, 8, 16 και 32 αντίστοιχα.

Πιο συγκεκριμένα έχουμε τις εξής τοπολογίες:

- **Τοπολογία 1<sup>η</sup>**: 1 Cloud (CLOUD), 2 Gateways (GW0, GW1) και 8 end devices (E0, E1, ..., E6, E7) ανά κόμβο ομίχλης
- **Τοπολογία 2<sup>η</sup>**: 1 Cloud (CLOUD), 2 Gateways (GW0, GW1) και 16 end devices (E0, E1, ..., E14, E15) ανά κόμβο ομίχλης
- **Τοπολογία 3<sup>η</sup>**: 1 Cloud (CLOUD), 2 Gateways (GW0, GW1) και 32 end devices (E0, E1, ..., E30, E31) ανά κόμβο ομίχλης
- **Τοπολογία 4<sup>η</sup>**: 1 Cloud (CLOUD), 2 Gateways (GW0, GW1) και 64 end devices (E0, E1, ..., E62, E63) ανά κόμβο ομίχλης

### 6.1.2 Πείραμα 2<sup>ο</sup>

Στο δεύτερο πείραμα θα δημιουργήσουμε ένα σύνολο τεσσάρων τοπολογιών δικτύου στις οποίες το πλήθος των τελικών συσκευών ανά κόμβο στο ακραίο επίπεδο edge παραμένει σταθερό και ίσο με 8.



Όπως βλέπουμε παραπάνω υπάρχει μοναδικό cloud και σταθερό πλήθος end devices (E0, E1, ..., E6, E7) στο επίπεδο 2 της υλοποίησης. Για κάθε τοπολογία που παράγουμε μεταβάλλουμε τον αριθμό των gateways (GW0, GW1, ..., GW(X-1), GW(X)) και διατηρούμε το πλήθος των συσκευών των επιπέδων 0 και 2 σταθερό. Δημιουργούμε και εξετάζουμε τέσσερις διαφορετικές τοπολογίες οι οποίες ενσωματώνουν πλειάδες σταθερού πλήθους και συγκεκριμένα ίσου με 8 τελικές συσκευές και μεταβάλλουμε τον αριθμό των κόμβων ομίχλης του επιπέδου 1 σε 2, 4, 8 και 16.

Πιο συγκεκριμένα έχουμε τις εξής τοπολογίες:

- **Τοπολογία 1<sup>η</sup>:** 1 Cloud (CLOUD), 2 Gateways (GW0, GW1) και 8 end devices (E0, E1, ..., E6, E7) ανά κόμβο ομίχλης
- **Τοπολογία 2<sup>η</sup>:** 1 Cloud (CLOUD), 4 Gateways (GW0, GW1, GW2, GW3) και 8 end devices (E0, E1, ..., E6, E7) ανά κόμβο ομίχλης
- **Τοπολογία 3<sup>η</sup>:** 1 Cloud (CLOUD), 8 Gateways (GW0, GW1, ..., GW6, GW7) και 8 end devices (E0, E1, ..., E6, E7) ανά κόμβο ομίχλης
- **Τοπολογία 4<sup>η</sup>:** 1 Cloud (CLOUD), 16 Gateways (GW0, GW1, ..., GW6, GW7) και 8 end devices (E0, E1, ..., E6, E7) ανά κόμβο ομίχλης

### 6.1.3 Παρατηρήσεις επί των πειραμάτων

Σημαντικό είναι να αναφέρουμε ότι για τα παραπάνω πειράματα υπάρχει αντιστοίχιση μεταξύ τοπολογιών του πρώτου πειράματος και του δεύτερου με τρόπο τέτοιο που ανεξαρτήτως πλήθους κόμβων και τελικών συσκευών ανά κόμβο, καταλήγουμε σε ίδιο αριθμό τελικών κόμβων. Για παράδειγμα η δεύτερη τοπολογία του πρώτου πειράματος αποτελείται από 2 κόμβους ομίχλης και 16 τελικές συσκευές ανά κόμβο ενώ η αντίστοιχη τοπολογία του δεύτερου πειράματος αποτελείται από 4 κόμβους και 8 τελικές συσκευές. Και οι δύο τοπολογίες έχουν ίδιο αριθμό τελικών συσκευών αλλά ενσωματώνουν διαφορετικό αριθμό κόμβων ομίχλης. Τη συγκεκριμένη παράμετρο του αριθμού των κόμβων ομίχλης μπορούμε να τη χρησιμοποιήσουμε προκειμένου να συγκρίνουμε αποτελέσματα που αφορούν ενέργεια, χρόνους εκτέλεσης και χρήση δικτύου και να εξετάσουμε τα οφέλη στρατηγικής ενσωμάτωσης (στην προκειμένη περίπτωση στατικά) κόμβων ομίχλης σε διαφορετικά πειράματα και υλοποιήσεις.

## 6.2 Παράμετροι συστήματος και συσκευών

Όπως προαναφέρθηκε, στις συσκευές που δημιουργούνται υπάρχει ετερογένεια όσον αφορά τα χαρακτηριστικά και τις αρχιτεκτονικές παραμέτρους. Παρόλα αυτά, όσες συσκευές παράγονται και συμμετέχουν στα παραπάνω σενάρια δημιουργούνται μέσω μιας κλάσης η οποία παράγει κόμβους ομίχλης. Δεδομένου αυτού, μοιράζονται αν και διαφορετικά παραμετροποιημένα, τα χαρακτηριστικά μιας συσκευής ομίχλης. Τα χαρακτηριστικά αυτά είναι το **επίπεδο** της κάθε συσκευής, το οποίο έχουμε υποδείξει στο προηγούμενο υποκεφάλαιο, η παράμετρος **MIPS** (million instructions per second) η οποία αναφέρεται στην επεξεργαστική ισχύ της εκάστοτε συσκευής, η **RAM**, το **εύρος ζώνης δικτύου** για κίνηση ροών δεδομένων upwards (βόρεια) και downwards (νότια) (**UpBw** και **DownBw** αντίστοιχα), η καθυστέρηση προώθησης δεδομένων upwards (**UpLatency**), η παράμετρος **RATE/MIPS** και η ενέργεια που καταναλώνεται από την εκάστοτε συσκευή. Στην τελευταία παράμετρο, υπάρχει μια διαφοροποίηση όσον αφορά την καταναλωθείσα ενέργεια, καθώς όταν η εκάστοτε συσκευή, ανεξαρτήτως επιπέδου, αναλαμβάνει την εξυπηρέτηση ενός ή περισσότερων αιτημάτων εφαρμογών, χρησιμοποιεί υπολογιστικούς και επεξεργαστικούς πόρους και ως συνέπεια αυτού η ενέργεια που καταναλώνει (**Power while Busy**) είναι εμφανώς μεγαλύτερη από αυτή που καταναλώνει σε περίπτωση που απλά βρίσκεται σε κατάσταση αναμονής (**Power while Idle**). Για την υλοποίηση και προσομοίωση των παραπάνω σεναρίων, οι τιμές που θα αναθέσουμε είναι ενδεικτικές και μπορούν να παραμετροποιηθούν. Υποθέτουμε ότι οι συσκευές έχουν επεξεργαστική δυνατότητα βάσει επιπέδου στο οποίο ανήκουν σαφώς μεγαλύτερη όταν μεταβαίνουμε από ένα κατώτερο επίπεδο σε κάποιο βόρεια αυτού. Ως συνέπεια αυτού αυξημένα θα είναι και τα επίπεδα των υπολοίπων χαρακτηριστικών όπως RAM, ενέργεια και άλλα. Έτσι δημιουργούμε τον παρακάτω πίνακα με τις παραμέτρους που αναφέραμε προηγουμένως και ορίζουμε τις τιμές αυτών για την κάθε συσκευή.

Η δημιουργία των συσκευών προκύπτει από την εξής γραμμή κώδικα:

```
createFodDevice(String nodeName, long mips, int ram, long upBw, long downBw, double ratePerMips, double busyPower, double idlePower);
```

Συσκευή	Cloud	Fog node (GW X)	Edge device (E X)
Επίπεδο/Layer	0 / UPPER	1 / MIDDLE	2 / LOWER
MIPS	1000000	50000	5000
RAM	100000	10000	1000
UpBw	100	10000	10000
DownBw	10000	10000	100
UpLatency	1	5	2
RATE/MIPS	0.01	0.0	0.0
Power while Busy	16*103	107.34	87.53
Power while Idle	16*83.25	83.43	82.44

Στη παραπάνω γραμμή κώδικα διακρίνουμε τις παραμέτρους mips, ram, upBw, downBw, ratePerMips, busyPower και idlePower οι οποίες είναι και αυτές που ορίζουμε στον παραπάνω πίνακα.



Για τις συσκευές **sensor** και **actuator**, ορίζεται ως καθυστέρηση επικοινωνίας (**latency**) των δύο συσκευών ίση με 6ms και 1ms αντίστοιχα και ο τύπος των συσκευών είναι deterministic.

Οι τιμές κόστους όσον αφορά την εκτέλεση εργασιών που απαιτούν την προσπέλαση μνήμης (Memory Cost), τη χρήση αποθηκευτικού χώρου (Storage Cost), το κόστος χρήσης του εύρους ζώνης (Bandwidth cost) καθώς και το καθολικό κόστος απασχόλησης πόρων (Global cost) δίνονται στον παρακάτω πίνακα. Όταν κάνουμε αναφορά στο κόστος εννοούμε το κόστος με το οποίο επιβαρύνεται το σύστημα είτε η εκάστοτε συσκευή σε περιπτώσεις που είναι απαραίτητη για την εξυπηρέτηση ενός αιτήματος η προσπέλαση μνήμης, η χρήση αποθηκευτικού χώρου καθώς και η χρήση του εύρους δικτύου. Το κόστος του εύρους δικτύου παρόλα αυτά ορίζεται ίσο με 0 γιατί θεωρούμε ότι πρόκειται για χαρακτηριστικό ιδιόκτητο για τις υλοποιήσεις μας.

Global Cost	Memory Cost	Storage Cost	Bandwidth Cost
3.0	0.05	0.001	0.0

Η εφαρμογή, αιτήματα της οποίας εξυπηρετούν οι συσκευές της εκάστοτε τοπολογίας μας αποτελείται από τρία modules τα οποία χαρακτηρίζονται από διαφορετικές απαιτήσεις συστήματος προκειμένου να μπορούν να φιλοξενηθούν και να εξυπηρετηθούν από τις συσκευές μας. Τα χαρακτηριστικά που ορίζουν τη βαρύτητα του κάθε module είναι η μνήμη RAM, η επεξεργαστική ισχύς (MIPS), το μέγεθος των modules (size) και τέλος το εύρος ζώνης (Bandwidth) που καταλαμβάνουν. Δεδομένου ότι διαμορφώνουμε ελεύθερα τις παραπάνω μεταβλητές, καταλήγουμε στον παρακάτω πίνακα.

Module	RAM	MIPS	Size	Bandwidth
<b>Lesser Module</b>	50	250	500	100
<b>Main Module</b>	200	1000	2000	800
<b>Storage Module</b>	100	250	12000	100

Τέλος ορίζουμε τις παραμέτρους των ροών tuples. Αυτό γίνεται με την συνάρτηση:

**addAppEdge(String source, String destination, double tupleCpuLength, double tupleNwLength, String tupleType, int direction, int edgeType);**

Συγκεκριμένα ορίζουμε την πηγή (source) καθώς και τον προορισμό (destination) της εργασίας προς εκτέλεση, το πλήθος των διεργασιών που απαιτούνται (σε MIPS) για την εξυπηρέτηση της ροής (tupleCpuLength), το μήκος δικτύου (σε bytes) των tuples που μεταφέρονται (tupleNwLength), τον τύπο του tuple (tupleType), τη διεύθυνση (direction) και τέλος τον τύπο της ακμής (edgeType). Συνοπτικά δημιουργούμε ακμές που συνδέουν πηγή και προορισμό ροών tuples και ορίζουμε τα χαρακτηριστικά των ροών αυτών.

Ακμές γράφου DAG	Διεύθυνση ροών Tuples	CPU length	Network length
IoTSensor → lesserModule	TUPLE UP	100	200
lesserModule → mainModule	TUPLE UP	1000	600
mainModule → storageModule	TUPLE UP	1000	300
mainModule → lesserModule	TUPLE DOWN	100	50
lesserModule → IoTActuator	TUPLE DOWN	100	50

## 6.3 Αποτελέσματα

### 6.3.α Κατανάλωση ενέργειας

Η κατανάλωση ενέργειας που αφορά τις τελικές συσκευές παραμένει ίδια καθ' όλη τη διάρκεια των προσομοιώσεων τόσο στον Edgewards, όσο και στο CloudOnly αλγόριθμο ανεξαρτήτως πειράματος και τοπολογίας.

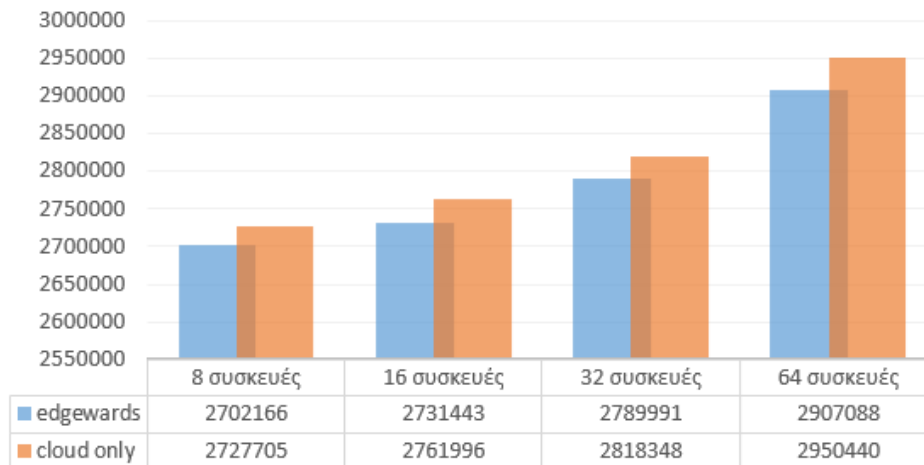
Στον πρώτο αλγόριθμο τα αιτήματα εφαρμογών που εξυπηρετούνται στις τελικές συσκευές (lesserModule) έχουν συγκεκριμένες και μη μεταβλητές απαιτήσεις συστήματος καθώς είναι στατικά στοιχεία τις παραμέτρους των οποίων έχουμε ορίσει στον κώδικά μας και έχουμε αναφέρει σε παραπάνω κεφάλαιο.. Έτσι λόγω της ιδιαιτερότητας του αλγόριθμου να εξυπηρετεί το σύνολο των υποδεέστερων εργασιών ενός αιτήματος εφαρμογής στη συσκευή που θα εκχωρηθεί η πρώτη παραχθείσα υποδεέστερη διεργασία του εκάστοτε αιτήματος εφαρμογής, υπάρχει συγκεκριμένος αριθμός αιτημάτων που εκτελούνται σε κάθε τελική συσκευή και παραμένει ίδιος για κάθε συσκευή ανεξαρτήτως πλήθους αυτών και τοπολογίας δικτύου. Το ίδιο συμβαίνει και με τις συσκευές-κόμβους ομίχλης στις οποίες μεταβαίνουν οι πρώτες παραχθείσες υποδεέστερες διεργασίες αιτημάτων που δεν μπορούν να εξυπηρετηθούν μια δεδομένη στιγμή στην τελική συσκευή λόγω έλλειψης αρχιτεκτονικών πόρων. Έτσι ο αριθμός αιτημάτων εργασιών και υποδεέστερων διεργασιών αυτών που εκτελούνται στους κόμβους ομίχλης παραμένει ίδιος. Συνεπώς τόσο στις τελικές συσκευές όσο και στις συσκευές-κόμβους ομίχλης, η κατανάλωση ενέργειας παραμένει ίδια μεταξύ τελικών συσκευών καθώς και μεταξύ κόμβων ομίχλης.

Στον CloudOnly αλγόριθμο, τόσο οι τελικές συσκευές όσο και οι συσκευές-κόμβοι ομίχλης δεν χρησιμοποιούνται για επεξεργασία αιτημάτων εφαρμογής, αλλά μόνο για την προώθηση αιτημάτων από το κατώτερο επίπεδο υλοποίησης προς το cloud για επεξεργασία και εξαγωγή αποτελεσμάτων και από το cloud προς το κατώτερο επίπεδο υλοποίησης προκειμένου να ληφθούν ενέργειες μέσω των συσκευών ενεργοποιητών. Έτσι η ενέργεια που καταναλώνεται τόσο από τις τελικές συσκευές όσο και από τους κόμβους ομίχλης είναι το σύνολο κατανάλωσης ενέργειας της εκάστοτε συσκευής κατά τη διάρκεια εκτέλεσης, με την στιγμιαία ενέργεια που καταναλώνεται κάθε δεδομένη στιγμή να είναι ίση με την ενέργεια που καταναλώνεται όταν η συσκευή βρίσκεται σε κατάσταση idle.

Όσον αφορά την κατανάλωση ενέργειας στο cloud, αυτή μεταβάλλεται καθώς το storageModule ή αλλιώς διαδικασία αποθήκευσης δημιουργείται σε αυτό με αποτέλεσμα τα παραγόμενα δεδομένα να αποστέλλονται στο cloud για αποθήκευση. Έτσι είναι μόνο λογικό να αυξάνεται η κατανάλωση ενέργειας αναλογικά με την αύξηση των τελικών συσκευών των παραπάνω τοπολογιών.

Παρατηρούμε λοιπόν ότι στο **πρώτο πείραμα** η αύξηση ενέργειας στο cloud τόσο για τον Edgewards όσο και για τον CloudOnly είναι προσεγγιστικά ανάλογη του συνόλου τελικών συσκευών και η διαφορά κατανάλωσης ενέργειας του cloud μεταξύ των αλγορίθμων παράγεται λόγω της ανάθεσης εκτέλεσης μόνο του storageModule στο cloud στην περίπτωση του Edgewards ενώ στην περίπτωση του CloudOnly έχουμε την εκτέλεση όλων των modules (lesser-main-storage Module) στο cloud.

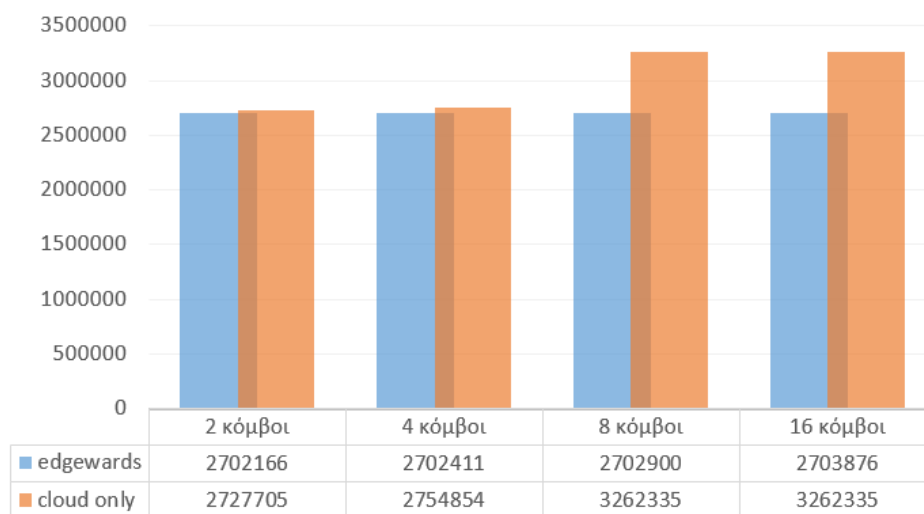
## Κατανάλωση ενέργειας στο cloud (Watt)



Εικόνα 6.1: Πείραμα 1<sup>ο</sup> όπου έχουμε σταθερό πλήθος κόμβων ίσο με 2 και μεταβλητό πλήθος τελικών συσκευών.

Στην περίπτωση του **δεύτερου πειράματος** παρατηρούμε ότι έχουμε για τον αλγόριθμο Edgewards ελάχιστες αποκλίσεις κατανάλωσης ενέργειας στο cloud και αυτό προκύπτει λόγω της εκτέλεσης της διαδικασίας storageModule για μεγαλύτερο αριθμό τελικών συσκευών για κάθε επόμενη τοπολογία με διαφοροποιημένο αριθμό κόμβων. Για τον αλγόριθμο CloudOnly παρατηρούμε ότι στις τοπολογίες 1 και 2 όπου έχουμε πλήθος κόμβων 2 και 4 αντίστοιχα και πλήθος τελικών συσκευών 16 και 32 αντίστοιχα, η διαφορά στην κατανάλωση ενέργειας στο cloud είναι διαφοροποιημένη ελαφρώς καθώς τα αρχιτεκτονικά χαρακτηριστικά του cloud είναι επαρκή για την ταυτόχρονη εκτέλεση όλων των module (lesser-main-cloud Module), ενώ στις περιπτώσεις των σεναρίων 3 και 4 με πλήθος κόμβων 8 και 16 αντίστοιχα και πλήθος τελικών συσκευών ίσο με 64 και 128 αντίστοιχα τα αρχιτεκτονικά χαρακτηριστικά του cloud δεν είναι επαρκή με αποτέλεσμα να απασχολείται πλήρως και για περισσότερο χρόνο όπως θα δούμε παρακάτω για την εκτέλεση όλων των αιτημάτων εφαρμογών.

## Κατανάλωση ενέργειας στο cloud (Watt)



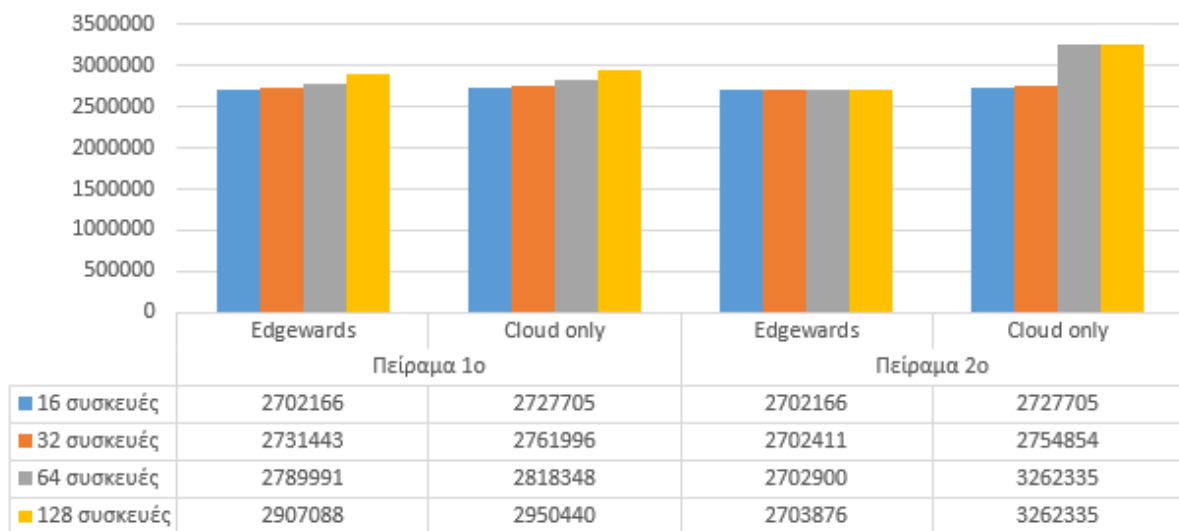
Εικόνα 6.2: Πείραμα 2<sup>ο</sup> όπου έχουμε σταθερό πλήθος τελικών συσκευών ανά κόμβο και μεταβλητό πλήθος κόμβων.

Για κάθε τοπολογία του 1<sup>ου</sup> πειράματος υπάρχει μια τοπολογία του 2<sup>ου</sup> που αποτελείται από ίδιο αριθμό τελικών συσκευών. Δεδομένου ότι στους κόμβους ομίχλης εκτελείται το mainModule, καταλαβαίνουμε ότι όσο μεγαλύτερος είναι ο αριθμός των κόμβων, τόσες περισσότερες mainModule διαδικασίες καλούνται να εξυπηρετηθούν. Δεν θα μπορούμε στη διαδικασία να ελέγξουμε τις περιπτώσεις πειραμάτων όπου δεν εκτελείται τίποτα στους κόμβους ομίχλης ενώ οι ίδιοι αποτελούν μόνο μέσο εκτέλεσης εργασιών που δεν μπορούν να εκτελεστούν στις τελικές συσκευές είτε μέσο προώθησης δεδομένων στο cloud για επεξεργασία.

Λαμβάνοντας αυτό υπόψιν, είναι κατανοητή η αυξημένη κατανάλωση ενέργειας του cloud στις περιπτώσεις του 2<sup>ου</sup> πειράματος έναντι του 1<sup>ου</sup> για τον αλγόριθμο CloudOnly καθώς πέραν των lesserModule των τελικών συσκευών καλείται να εκτελέσει ένα σημαντικά μεγαλύτερο αριθμό από σαφώς πιο απαιτητικά modules (mainModule) που προωθούνται πλέον από τους κόμβους ομίχλης στο cloud.

Για τον Edgewards αλγόριθμο, παρατηρούμε ότι η κατανάλωση ενέργειας στο cloud αυξάνεται σημαντικά λιγότερο στις περιπτώσεις του 2<sup>ου</sup> πειράματος καθώς ο αριθμός τελικών συσκευών ανά κόμβο ομίχλης είναι σημαντικά μικρότερος από ότι στις περιπτώσεις του 1<sup>ου</sup> πειράματος. Έτσι, εάν υπάρχουν modules που δεν μπορούν να εκτελεστούν από τις τελικές συσκευές, προωθούνται στους κόμβους ομίχλης. Κομβικό είναι το γεγονός ότι στο 1<sup>ο</sup> πείραμα, οι κόμβοι είναι 2 σε πλήθος πράγμα το οποίο σημαίνει ότι φτάνουν ταυτόχρονα πολύ περισσότερα αιτήματα σε κάθε έναν από αυτούς σε αντίθεση με τις περιπτώσεις του 2<sup>ου</sup> πειράματος. Το αποτέλεσμα είναι για το 1<sup>ο</sup> πείραμα ότι οι κόμβοι δεν μπορούν να εξυπηρετήσουν ταυτόχρονα όλα τα αιτήματα εφαρμογών λόγω έλλειψης πόρων, με αποτέλεσμα να προωθείται μεγαλύτερος αριθμός αιτημάτων εφαρμογών στο cloud. Ως συνέπεια αυτού, η κατανάλωση ενέργειας του cloud στο 1<sup>ο</sup> πείραμα έναντι του 2<sup>ου</sup> για την περίπτωση του αλγόριθμου Edgewards αυξάνεται με σημαντικά μεγαλύτερο ρυθμό.

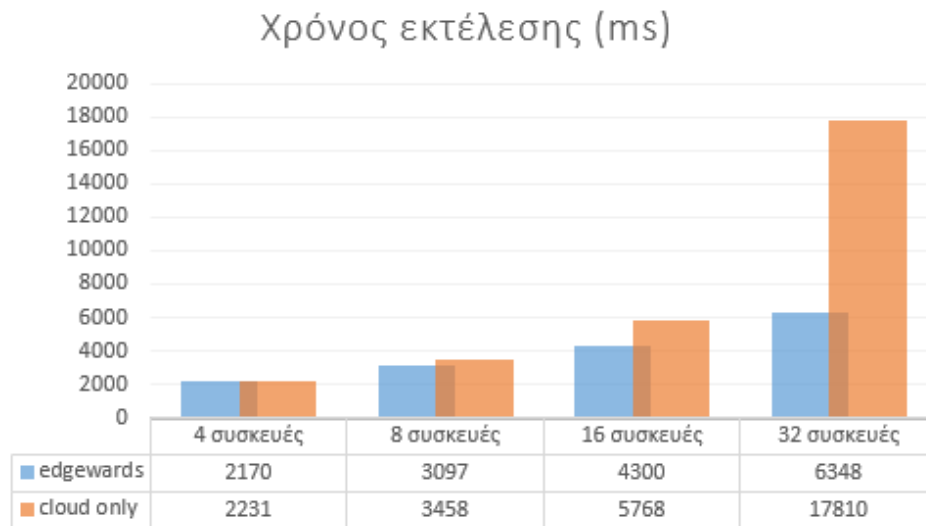
### Συγκριτικά σύνολα κατανάλωσης ενέργειας στο cloud μεταξύ πειραμάτων (Watt)



**Εικόνα 6.3:** Συγκεντρωτικό διάγραμμα κατανάλωσης ενέργειας στο cloud για τις όλες τις περιπτώσεις των πειραμάτων 1 και 2 που αποτελούνται από συνολικά ίσο αριθμό τελικών συσκευών αλλά ενσωματώνουν διαφορετικό αριθμό κόμβων και αριθμό τελικών συσκευών ανά κόμβο.

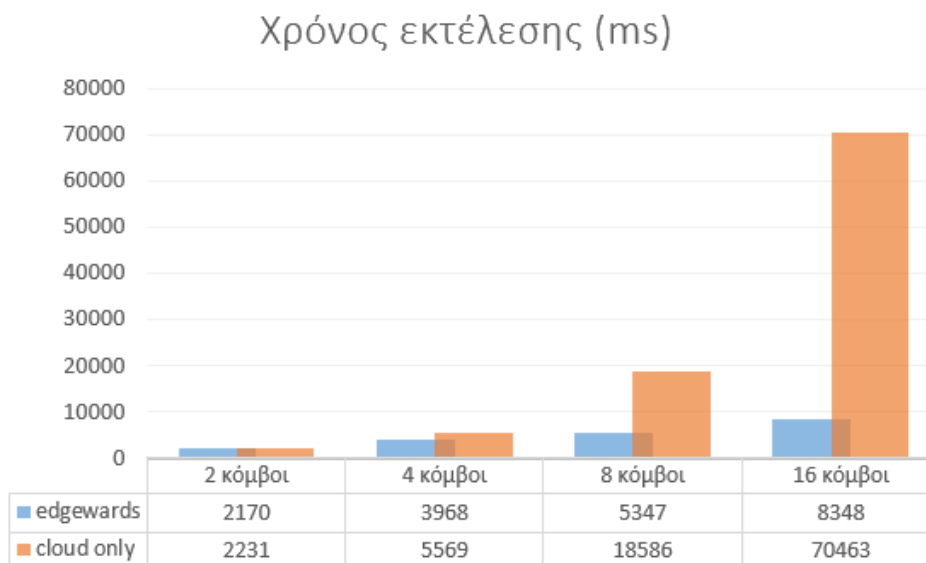
### 6.3.β Χρόνος εκτέλεσης

Στο 1<sup>ο</sup> πείραμα, ο χρόνος εκτέλεσης αυξάνεται σχεδόν γραμμικά για τον αλγόριθμο Edgewards και αυτό επειδή υπάρχει μια ομοιομορφία όσον αφορά την κατανομή αιτημάτων εφαρμογής από τις συσκευές κατώτερου επιπέδου προς το cloud. Σε αντίθεση ο αλγόριθμος CloudOnly προωθεί το σύνολο λιγότερο απαιτητικών module (lesserModule) και πιο απαιτητικών module (mainModule) στο cloud με αποτέλεσμα όσο αυξάνεται ο αριθμός των συσκευών που αποστέλλουν δεδομένα στο cloud να αυξάνεται με σημαντικά μεγαλύτερο ρυθμό ο χρόνος εκτέλεσης του συνόλου αιτημάτων.



Εικόνα 6.4: Πείραμα 1<sup>ο</sup> όπου έχουμε σταθερό πλήθος κόμβων ίσο με 2 και μεταβλητό πλήθος τελικών συσκευών.

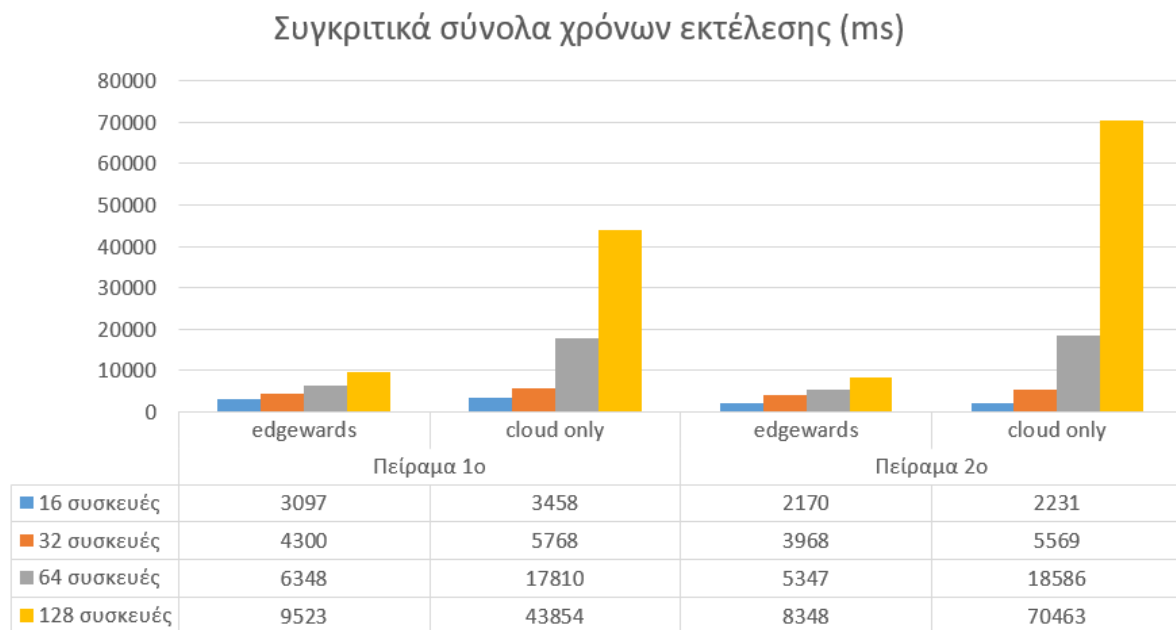
Ομοίως στο 2<sup>ο</sup> πείραμα, ο χρόνος εκτέλεσης αυξάνεται σχεδόν γραμμικά για τον αλγόριθμο Edgewards. Σε αντίθεση με τον Edgewards αλγόριθμο, στην περίπτωση του CloudOnly, λόγω και της ταυτόχρονης εξυπηρέτησης περισσότερων mainModule διαδικασιών τα οποία είναι πολύ πιο απαιτητικά όσον αφορά τη χρήση πόρων, ο χρόνος εκτέλεσης αυξάνεται με πολύ μεγαλύτερο ρυθμό.



Εικόνα 6.5: Πείραμα 2<sup>ο</sup> όπου έχουμε σταθερό πλήθος τελικών συσκευών ανά κόμβο και μεταβλητό πλήθος κόμβων.

Εάν παρατηρήσουμε το συγκεντρωτικό διάγραμμα χρόνων εκτέλεσης των διαφορετικών περιπτώσεων των πειραμάτων 1 και 2, καταλήγουμε ομοίως στο ίδιο συμπέρασμα με αυτό που εξαγάγαμε παραπάνω. Οι χρόνοι εκτέλεσης για τον αλγόριθμο Edgewards είναι μικρότεροι στις διαφορετικές περιπτώσεις του 2<sup>ου</sup> πειράματος έναντι των περιπτώσεων του 1<sup>ου</sup> πειράματος και αυτό επειδή πλήθος lesserModule που δεν μπορεί να εκτελεστεί από τις τελικές συσκευές προωθείται σε έναν πολύ μεγαλύτερο αριθμό κόμβων. Όπως αναφέραμε για το 2<sup>ο</sup> πείραμα, σε κάθε κόμβο ενσωματώνονται πολύ λιγότερες συσκευές έναντι αντίστοιχων περιπτώσεων του πειράματος 1, με αποτέλεσμα να υπάρχει συνολικά μεγαλύτερο εύρος πόρων συστήματος στο επίπεδο 1, δηλαδή το επίπεδο που έχουμε ορίσει ως επίπεδο κόμβων ομίχλης στην υλοποίηση. Ως αποτέλεσμα αυτού, υπάρχει σημαντικά μειωμένη ανάγκη προώθησης πόρων στο cloud και επίσης σημαντικά μειωμένος χρόνος εκτέλεσης για κάθε περίπτωση του 2<sup>ου</sup> πειράματος έναντι του 1<sup>ου</sup> για τον αλγόριθμο Edgewards.

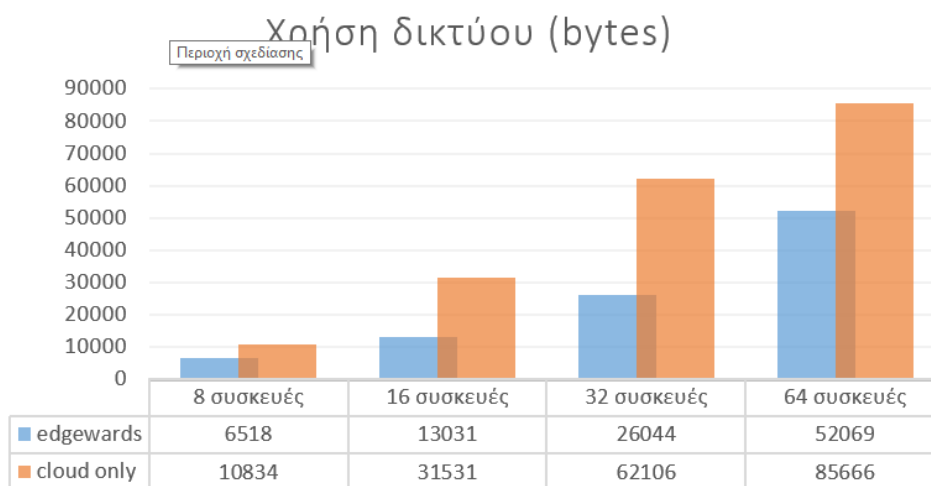
Για τον CloudOnly αλγόριθμό παρατηρούμε, όμοια με το συγκεντρωτικό διάγραμμα κατανάλωσης ενέργειας και το συμπέρασμα παραπάνω, ότι η εξάρτηση της επεξεργασίας δεδομένων από το cloud σημαίνει την περαιτέρω επιβάρυνση της υλοποίησης χρονικά με μεγαλύτερες περιόδους εργασίας και εξυπηρέτησης του συνόλου αιτημάτων που παράγονται από τελικές συσκευές και κόμβους ομίχλης.



**Εικόνα 6.6:** Συγκεντρωτικό διάγραμμα χρόνου εκτέλεσης του συνόλου αιτημάτων εφαρμογών για τις όλες τις περιπτώσεις των πειραμάτων 1 και 2 που αποτελούνται από συνολικά ίσο αριθμό τελικών συσκευών αλλά ενσωματώνουν διαφορετικό αριθμό κόμβων και αριθμό τελικών συσκευών ανά κόμβο.

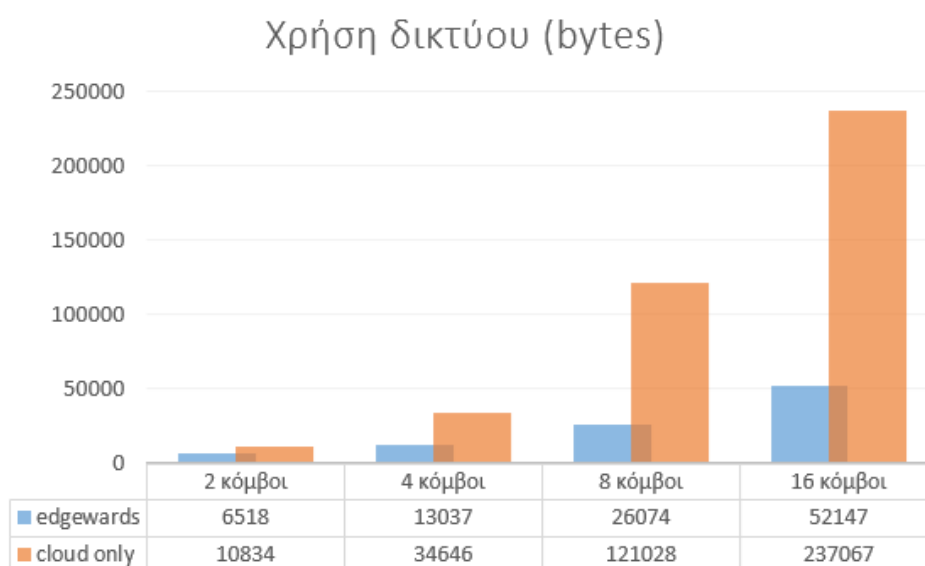
### 6.3.γ Χρήση δικτύου

Στην περίπτωση του αλγόριθμου Edgewards, ανεξαρτήτως πειράματος παρατηρούμε σημαντικά χαμηλότερες τιμές χρήσης δικτύου. Στο 1<sup>ο</sup> πείραμα δεδομένου ότι οι κόμβοι ομίχλης είναι ίσοι με 2 σε πλήθος, υπάρχει περιορισμός εύρους δικτύου για upload και download από και προς cloud και τελικές συσκευές. Ως συνέπεια, έχουμε χαμηλότερες τιμές χρήσης δικτύου από τον CloudOnly καθώς στην περίπτωση του Edgewards είναι περιορισμένος ο αριθμός αιτημάτων εφαρμογών που εκτελούνται στο cloud και το cloud χρησιμοποιείται κυρίως για τη διαδικασία αποθήκευσης δεδομένων (storageModule).



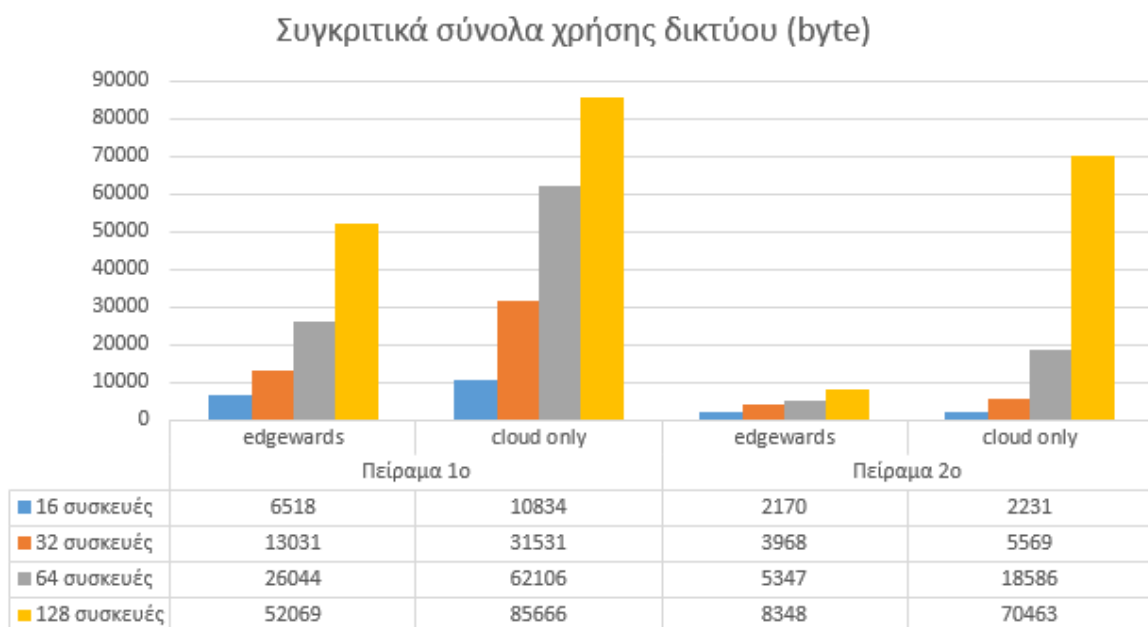
Εικόνα 6.7: Πείραμα 1<sup>ο</sup> όπου έχουμε σταθερό πλήθος κόμβων ίσο με 2 και μεταβλητό πλήθος τελικών συσκευών.

Στην περίπτωση του 2<sup>ου</sup> πειράματος, το αυξημένο πλήθος κόμβων στην υλοποίηση σημαίνει ότι η ίδια πληροφορία μοιράζεται μεταξύ περισσότερων κόμβων και ως συνέπεια υπάρχει περισσότερο συνολικά εύρος δικτύου για μεταφορά δεδομένων. Ως αποτέλεσμα αυτού υπάρχει ελάττωση του εύρους δικτύου που απασχολείται καθώς μεταφέρονται ίδια ποσά πληροφορίας που απασχολούν το δίκτυο για πολύ μικρότερες χρονικές περιόδους λόγω του αυξημένου αριθμού καναλιών μετάδοσης. Έτσι παρατηρούνται πολύ μικρότερες μετρήσεις για τον Edgewards ενώ και στην περίπτωση του CloudOnly παρατηρούμε ελαφρώς χαμηλότερη χρήση εύρους δικτύου συγκριτικά με τις τιμές του 1<sup>ου</sup> πειράματος.



Εικόνα 6.8: Πείραμα 2<sup>ο</sup> όπου έχουμε σταθερό πλήθος τελικών συσκευών ανά κόμβο και μεταβλητό πλήθος κόμβων.

Στο συγκεντρωτικό διάγραμμα χρήσης δικτύου παρατηρούμε ότι η ενσωμάτωση κόμβων ομίχλης στην υλοποίηση σημαίνει την ελάττωση χρήσης εύρους δικτύου τόσο για τον Edgewards όσο και για τον CloudOnly αλγόριθμο. Πιο συγκεκριμένα στην περίπτωση του αλγόριθμου Edgewards παρατηρούμε ότι κατά τη μετάδοση αιτημάτων εφαρμογής που δεν μπορούν να εξυπηρετηθούν από την τελική συσκευή είτε από τον κόμβο ομίχλης, λόγω του αυξημένου πλήθους καναλιών μετάδοσης, απασχολείται σημαντικά λιγότερο εύρος δικτύου. Στην περίπτωση του CloudOnly αλγόριθμου παρατηρείται ότι υπάρχουν μεταβολές που αφορούν τη συνολική χρήση δικτύου αλλά δεν πρόκειται για τόσο εμφανή διαφορά καθώς εξακολουθεί να υπάρχει μεγάλο πλήθος δεδομένων που πρέπει να μεταγκατασταθούν στο cloud για εξυπηρέτηση και στη συνέχεια να προωθηθούν αποτελέσματα της επεξεργασίας στους αντίστοιχους κόμβους και τελικές συσκευές που τα παρήγαγαν. Το αυξημένο πλήθος καναλιών σημαίνει μόνο ότι η πληροφορία θα τμηθεί και μεταδοθεί μέσω περισσότερων καναλιών, πράγμα το οποίο σημαίνει αθροιστικά μεγαλύτερο εύρος δικτύου, άρα και χρονικά μικρότερες περιόδους απασχόλησης του δικτύου.



**Εικόνα 6.9:** Συγκεντρωτικό διάγραμμα χρήσης δικτύου για τις όλες τις περιπτώσεις των πειραμάτων 1 και 2 που αποτελούνται από συνολικά ίσο αριθμό τελικών συσκευών αλλά ενσωματώνουν διαφορετικό αριθμό κόμβων και αριθμό τελικών συσκευών ανά κόμβο.



## ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα

---

Λαμβάνοντας υπόψιν τα πειράματα που διεξήχθησαν καθώς και τα αποτελέσματα που εξαγάγαμε στα παραπάνω κεφάλαια, καταλήγουμε στη διαπίστωση ότι η ενσωμάτωση τεχνολογιών υπολογιστικής ομίχλης καθώς και η προσέγγιση εξυπηρέτησης των αιτημάτων πλησίον του τοπικού δικτύου και όσο το δυνατό νοτιότερα στα επίπεδα δικτύου θέτει τις προϋποθέσεις για δίκτυα έξυπνων συσκευών τα οποία μπορούν να χαρακτηρίζονται από ένα βαθμό αυτονομίας αναφορικά με το cloud.

Επιπλέον, η ενσωμάτωση κόμβων ομίχλης σε σενάρια cloud-IoT δημιουργεί υλοποιήσεις που μπορούν να διαχειριστούν αποδοτικότερα τους πόρους συστήματος και να μειώσουν λειτουργικά κόστη και χρόνους εξυπηρέτησης αιτημάτων τελικών χρηστών. Η στρατηγική ενσωμάτωση κόμβων καθώς και η επιλογή επαρκούς πλήθους αριθμού αυτών για την όσο το δυνατό αποδοτικότερη και ευρύτερη κατανομή των δεδομένων που παράγονται από όλα τα επίπεδα της υλοποίησης είναι χαρακτηριστικά και ιδιότητες που ορίζονται από τον διαχειριστή της υποδομής και περιορίζονται από τις δυνατότητες υλικού, λειτουργικού συστήματος, ετερογένειας αρχιτεκτονικών στοιχείων των συσκευών όλων των επιπέδων της υλοποίησης και τέλος από τον διαθέσιμο χρηματικό προϋπολογισμό που μπορεί να διαθέσει ο εκάστοτε φορέας που επιθυμεί να εκμεταλλευτεί την υλοποίηση.

Το μόνο σίγουρο είναι ότι οι σημαντικά ελαττωμένες τιμές που παρατηρήθηκαν σε χρόνους εκτέλεσης, χρήσης πόρων του cloud και των datacenter αυτού και χρήσης δικτύου καθιστούν την υπολογιστική ομίχλη την κατευθυντήρια οδό για δίκτυα ανεξάρτητα από cloud και αποδοτικότερα από τις μέχρι τώρα υλοποιήσεις.

1. Produced by the OpenFog Consortium Architecture Working Group [www.OpenFogConsortium.org](http://www.OpenFogConsortium.org), “**OpenFog Reference Architecture for Fog Computing**”, February 2017
2. Hany F. Atlam, Robert J. Walters and Gary B. Wills “**Fog Computing and the Internet of Things: A Review**”, Published: 8 April 2018
3. “**A Framework of Fog Computing: Architecture, Challenges and Optimization**”, YANG LIU, JONATHAN E. FIELDSEND, and GEYONG MIN
4. “**Modeling and Simulation of Fog and Edge Computing Environments Using iFogSim Toolkit**”, Rajkumar Buyya and Satish Narayana Srirama, Wiley Telecom, Page(s): 433 – 465, 2019
5. “**Fog Computing: Characteristics and Challenges**”, Shabnam Kumari, Surender Singh and Radha, International Journal of Emerging Trends & Technology in Computer Science, Volume 6, Issue 2, March - April 2017
6. Special Issue Paper “**iFogSim: A toolkit for modeling and simulation of resource management techniques on the Internet of Things, Edge and Fog computing environments**”, Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh and Rajkumar Buyya
7. “**Load balancing aware scheduling algorithms for fog networks**”, Anil Singh and Nitin Auluck
8. “**An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing**”, Swati Agarwal, Shashank Yadav and Arun Kumar Yadav

## ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

---

- [1]. <https://www.oracle.com/internet-of-things/what-is-iot/>
- [2]. <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#benefits>
- [3]. <https://www.geeksforgeeks.org/types-of-cloud/>
- [4]. <https://www.sap.com/poland/insights/what-is-cloud-computing.html>