



Πανεπιστήμιο Πειραιώς

**Τμήμα Δικτυοκεντρικών Πληροφοριακών
Συστημάτων**

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

«Ψηφιακά Συστήματα και Υπηρεσίες»

Διπλωματική Εργασία

Θέμα Εργασίας:

**Καινοτόμος προσέγγιση δυναμικής ανάλυσης
δεδομένων, καθορισμού πόρων επεξεργασίας και
παραγωγής περιεχομένου σε περιβάλλον Edge / Fog
Computing και Διαδικτύου των Πραγμάτων**

**An innovative approach for dynamic data analysis,
processing nodes identification and content
generation in Edge / Fog Computing and Internet of
things environment**

Μεταπτυχιακός Φοιτητής:

Γεώργιος Γέγος – ΜΕ14022

Επιβλέπων: Δημοσθένης Κυριαζής, Επ. Καθηγητής

Πειραιάς 2017

Copyright © Γέγος Γεώργιος, 2017 Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Ευχαριστίες

Η διπλωματική αυτή εκπονήθηκε στο **Εργαστήριο Δικτυοκεντρικών Συστημάτων και Υπηρεσιών (ΕΔΣΥ)** το οποίο ανήκει στο **Τμήμα Ψηφιακών Συστημάτων** του Πανεπιστημίου Πειραιώς, υπό την επίβλεψη του Καθηγητή Δημοσθένη Κυριαζή.

Θα ήθελα να ευχαριστήσω θερμά τον κ. Δημοσθένη Κυριαζή για την υποστήριξη και καθοδήγηση που μου προσέφερε, καθώς και για την ευκαιρία που μου έδωσε να ασχοληθώ με την ερευνητική περιοχή του Fog Computing και Internet of things.

Θα ήθελα επίσης να ευχαριστήσω ολόψυχα την οικογένειά μου για την συμπαράσταση και την πολύτιμη στήριξη που μου παρείχαν σε όλα τα χρόνια της φοιτητικής μου ζωής.

Περίληψη

Τα τελευταία χρόνια η κυρίαρχη τάση στο χώρο της πληροφορικής ήταν τα αποκαλούμενο Cloud Computing, το οποίο αποτελεί πλέον τη βασική προσέγγιση που ακολουθείται στα πληροφοριακά συστήματα σε ολόκληρο τον κόσμο.

Η παρούσα διπλωματική εργασία μελετά το Fog Computing ή αλλιώς Fogging το οποίο είναι μια αρχιτεκτονική που επεκτείνει το μοντέλο του Cloud Computing στα άκρα του δικτύου και αναμένεται να αποτελέσει ιδανική πλατφόρμα για το Internet of things (Διαδίκτυο των Πραγμάτων) και υπηρεσίες όπως είναι το διασυνδεδεμένο όχημα (connected vehicle), τις έξυπνες πόλεις, τα δίκτυα που χρησιμοποιούν αισθητήρες.

Ο στόχος της είναι να αναδείξει την χρησιμότητα του Fogging σε συνδυασμό με την αρχιτεκτονική του Cloud λόγω των συνεχώς αυξανόμενων αναγκών όσον αφορά στη διακίνηση δεδομένων. Αρχικά, παρουσιάζεται η θεωρία γύρω από τις αρχιτεκτονικές που προαναφέρθηκαν καθώς και οι τεχνολογίες που χρησιμοποιούνται. Με σκοπό να γίνουν καλύτερα κατανοητά αναλύεται η αρχιτεκτονική του Fog, τα πλεονεκτήματα και τα μειονεκτήματα του σε σχέση με του Cloud.

Στη συνέχεια, μελετάμε την μεθοδολογία και την σχεδίαση της εφαρμογής που έχουμε σκοπό να αναπτύξουμε όπως επίσης και διάφορα προγραμματιστικά εργαλεία που μπορούν να μας βοηθήσουν στην υλοποίηση του συστήματος.

Τέλος, συνδυάζοντας τις παραπάνω γνώσεις, υλοποιείται μια διαδικτυακή εφαρμογή με θέμα την εύρεση αποτελεσμάτων των δημοτικών εκλογών ανάλογα με την τοποθεσία του χρήστη και παρουσιάζεται η λειτουργία του.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<i>Ευχαριστίες</i>	3
Περίληψη	4
Πίνακας Περιεχομένων	5
Λιστα σχηματων	7
ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ ΕΙΣΑΓΩΓΗ	9
1.1 Ορισμος του προβληματος	9
1.2 Δομη της Διπλωματικης Εργασιας	12
1.3 Συνεισφορα της Διπλωματικης Εργασιας	12
ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	13
<i>Εισαγωγη</i>	13
2. <i>Cloud Computing</i>	13
2.1 Η εξέλιξη του Cloud Computing	14
2.2 Θεωρία	20
2.3 Το μοντέλο Software-As-a-Service (SaaS)	25
2.4 Το Platform-As-a-Service μοντέλο	28
2.5 Το Infrastructure-As-a-Service μοντέλο	31
3. <i>Cloud μοντελα αναπτυξης</i>	32
3.1 Δημόσια clouds (public clouds)	33
3.2 Ιδιωτικά clouds (private clouds)	35
3.3 Υβριδικά clouds (Hybrid)	36
4. <i>Cloud Computing Platforms</i>	38
4.1 Google Compute Engine (GCE)	39
4.2 Amazon EC2	40
4.3 Windows Azure	43
4.4 Yahoo open-source cloud-serving engine	46
4.5 Eucalyptus software	48
5. <i>Internet of Things (IoT)</i>	49
5.1 Ορισμός του IoT	49
5.2 Το IoT στη ζωή μας	51
5.3 Κυρια χαρακτηριστικα του IoT	52
5.4 Εφαρμογές	55
ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ	68
6.1 <i>Fog Computing</i>	68
6.2 <i>Fog Computing Μεγαλης Κλιμακας και Πραγματικου χρονου</i>	71
6.3 <i>Χαρακτηριστικα Ασφαλειας Fog Computing</i>	75

6.4 Εφαρμογες <i>Fog Computing</i>	80
6.4.1 Υπολογιστική ομίχλη στο Έξυπνο Δίκτυο	80
6.4.2 Υπολογιστική ομίχλη σε έξυπνα φανάρια και συνδεδεμένα οχήματα	80
6.4.3 Ασύρματος Αισθητήρας και Δίκτυα Ενεργοποίησης	81
6.4.4 Αποκεντρωμένος Έλεγχος Έξυπνου Κτιρίου	81
6.4.5 IoT και Κυβερνο-φυσικά συστήματα (CPS)	82
6.4.6 Software Defined Networks (SDN)	83
ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ ΥΛΟΠΟΙΗΣΗ	84
7.1 <i>Javascript</i>	84
7.1.1 Ιστορία	86
7.1.2 Τυποποίηση	87
7.1.3 Μετέπειτα εξελίξεις	87
7.1.4 Χαρακτηριστικά	88
7.2 <i>NodeJS</i>	93
7.2.1 Ιστορία	93
7.2.2 Επισκόπηση	94
7.2.3 Τεχνικές Λεπτομέρειες	95
7.3 <i>MongoDB</i>	97
7.4 Αρχιτεκτονική Υλοποίησης	102
7.5 Πληροφορίες υλοποίησης	106
ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ ΣΕΝΑΡΙΑ - ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ	111
8.1 ΕΙΣΑΓΩΓΗ	111
8.2 ΠΡΩΤΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ	113
8.3 ΔΕΥΤΕΡΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ	116
8.4 ΤΡΙΤΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ	117
ΚΕΦΑΛΑΙΟ ΕΚΤΟ ΣΥΜΠΕΡΑΣΜΑΤΑ	118
ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ ΒΙΒΛΙΟΓΡΑΦΙΑ	120

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

ΕΙΚΟΝΑ 1 : ΣΥΝΝΕΦΑ ΤΟΥ CLOUD COMPUTING	16
ΕΙΚΟΝΑ 2 : ΜΙΚΡΟΤΕΡΑ ΚΎΜΑΤΑ ΜΕΣΑ ΣΤΗΝ ΕΠΟΧΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ.	17
ΕΙΚΟΝΑ 3 : ΕΞΕΛΙΞΗ ΤΟΥ CLOUD COMPUTING	18
ΕΙΚΟΝΑ 4 : ΑΥΞΟΜΕΙΩΣΗ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΠΟΡΩΝ CLOUD COMPUTING.....	24
ΕΙΚΟΝΑ 5 : ΔΗΜΟΣΙΑ CLOUDS	35
ΕΙΚΟΝΑ 6 : ΥΒΡΙΔΙΚΟ (HYBRID) CLOUD.....	37
ΕΙΚΟΝΑ 7 : ΠΑΡΑΔΕΙΓΜΑΤΑ CSP ΚΑΙ ΟΙ ΑΝΤΙΣΤΟΙΧΕΣ ΠΡΟΣΦΟΡΕΣ ΤΟΥΣ.....	38
ΕΙΚΟΝΑ 8 : ΠΛΑΤΦΟΡΜΕΣ CLOUD COMPUTING	39
ΕΙΚΟΝΑ 9 : GOOGLE COMPUTE ENGINE	40
ΕΙΚΟΝΑ 10 : AMAZON EC2	41
ΕΙΚΟΝΑ 11 : ΠΛΑΤΦΟΡΜΑ WINDOWS AZURE	45
ΕΙΚΟΝΑ 12 : ΥΑΗΟΟ CLOUD COMPUTING	48
ΕΙΚΟΝΑ 13 : ΑΡΧΙΤΕΚΤΟΝΙΚΗ EUCALYPTUS	49
ΕΙΚΟΝΑ 14 : ΑΡΧΙΤΕΚΤΟΝΙΚΗ FOG.....	69
ΕΙΚΟΝΑ 15 : ΠΛΕΟΝΕΚΤΗΜΑΤΑ FOG COMPUTING.....	72
ΕΙΚΟΝΑ 16 : ΛΟΓΟΤΥΠΟ JAVASCRIPT	86
ΕΙΚΟΝΑ 17 : ΛΟΓΟΤΥΠΟ NODEJS.....	94
ΕΙΚΟΝΑ 18 : Η ΔΟΜΗ ΤΗΣ MONGODB.....	100
ΕΙΚΟΝΑ 19 : ΠΑΡΑΔΕΙΓΜΑ ΤΗΣ ΔΟΜΗΣ ΕΝΟΣ ΕΓΓΡΑΦΟΥ ΣΤΗΝ MONGODB.....	100
ΕΙΚΟΝΑ 20 : ΕΓΓΡΑΦΟ ΜΕ ΠΟΙΚΙΛΟΥΣ ΤΥΠΟΥΣ ΔΕΔΟΜΕΝΩΝ	101
ΕΙΚΟΝΑ 21 : ΕΓΓΡΑΦΟ ΜΕ ΕΜΦΟΛΕΥΜΕΝΟ ΈΓΓΡΑΦΟ	102
ΕΙΚΟΝΑ 22 : ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ	104
ΕΙΚΟΝΑ 23 : ΔΙΑΦΟΡΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ CLOUD ΜΕ FOG COMPUTING	107
ΕΙΚΟΝΑ 24 : ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΕΦΑΡΜΟΓΗΣ ΠΕΛΑΤΗ.....	114

ΕΙΚΟΝΑ 25 : ΛΕΙΤΟΥΡΓΙΕΣ ΕΦΑΡΜΟΓΗΣ ΠΕΛΑΤΗ	115
ΕΙΚΟΝΑ 26 : ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΟΥ ΠΕΙΡΑΙΑ ΣΕ ΠΙΝΑΚΑ ΚΑΙ ΔΙΑΓΡΑΜΜΑ	116
ΕΙΚΟΝΑ 27 : ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΟΥ ΖΩΓΡΑΦΟΥ ΣΕ ΠΙΝΑΚΑ ΚΑΙ ΔΙΑΓΡΑΜΜΑ	117
ΕΙΚΟΝΑ 28 : ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΟΥ ΝΙΚΑΙΑΣ ΣΕ ΠΙΝΑΚΑ ΚΑΙ ΔΙΑΓΡΑΜΜΑ	118
ΕΙΚΟΝΑ 29 : ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΟΥ ΝΙΚΑΙΑΣ ΣΕ ΠΙΝΑΚΑ ΜΕ ΑΔΥΝΑΜΙΑ ΦΟΡΤΩΣΗΣ ΔΙΑΓΡΑΜΜΑΤΟΣ	119

ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

ΕΙΣΑΓΩΓΗ

1.1 ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Σήμερα, υπάρχουν δισεκατομμύρια συσκευές που συνδέονται με το Διαδίκτυο, και αυτό έχει οδηγήσει σε ορισμένες προόδους στις τεχνολογικές εξελίξεις των Ηλεκτρονικών και Τηλεπικοινωνιών κατά τα τελευταία χρόνια που οδήγησε σε διάφορα είδη πολύ ισχυρών συσκευών με δυνατότητες επικοινωνίας και δικτύωσης που έχουν προσελκύσει τις βιομηχανίες να υιοθετήσουν αυτή την τεχνολογία στην καθημερινή επιχείρησή τους για να αυξήσουν την αποτελεσματικότητά τους. Εκτός από τον βιομηχανικό τομέα, υπάρχουν και άλλοι τομείς όπως η βοήθεια των υπηρεσιών διαβίωσης, τις δημόσιες υπηρεσίες, κλπ, τα οποία έχουν μεγάλη ζήτηση για τις εξελίξεις της Τεχνολογίας Πληροφοριών και Επικοινωνιών. Ως εκ τούτου, υπάρχει η ανάγκη για ένα νέο πρότυπο στην επικοινωνία Machine2Machine που επιτρέπει τη συνδεσιμότητα των "Πραγμάτων" στο Παγκόσμιο Διαδίκτυο. Αυτό το παράδειγμα είναι γνωστό με τον όρο IoT.

Το IoT είναι το δίκτυο των φυσικών αντικειμένων ή "πράγματα" ενσωματωμένα με ηλεκτρονικά, αισθητήρες και συνδεσιμότητα για να μπορέσει να επιτύχει την αξία και την εξυπηρέτηση με την ανταλλαγή δεδομένων με τον κατασκευαστή, χειριστή ή / και σε άλλες συνδεδεμένες συσκευές μέσω προηγμένων πρωτοκόλλων επικοινωνίας χωρίς ανθρώπινη λειτουργία. Η τεχνολογία του Διαδικτύου των πραγμάτων έχει εξελιχθεί γύρω από το περιβάλλον που βασίζεται στην τεχνολογία πληροφοριών και επικοινωνίας και κοινωνικής υποδομής, και πρέπει να γνωρίζουμε την τεχνολογική εξέλιξη του στο μέλλον.

Συνδέοντας δισεκατομμύρια ή ακόμη και τρισεκατομμύρια συσκευές στο Internet, συνειδητοποιούμε ότι υπάρχουν πολλές εφαρμογές που χρησιμοποιούνται από τις βιομηχανίες, η κυβέρνηση, το κοινό, κ.λπ. Για παράδειγμα, το σύστημα ευφυών μεταφορών (ITS), εφαρμογή η οποία παρακολουθεί την κίνηση σε μια πόλη με ασύρματους αισθητήρες ή μέσω παρακολούθησης βίντεο, και στέλνει τις πληροφορίες στους χρήστες σχετικά με τις κινητές συσκευές τους με τη βοήθεια του Global Positioning System (GPS) πομποδέκτη, με σκοπό να ενημερώσει τους χρήστες να αποφύγουν μποτιλιάρισμα και την πρόληψη των ατυχημάτων. Αυτό είναι μόνο ένα παράδειγμα εφαρμογής από πολλά περισσότερα παραδείγματα όπως το έξυπνο σπίτι και e-Health εφαρμογές. Τεράστιες ποσότητες δεδομένων παράγονται από τα δισεκατομμύρια των συνδεδεμένων συσκευών και μεταφέρονται σε όλο το δίκτυο με το Internet.

Κάπου εδώ ξεκινούν τα οφέλη της ενσωμάτωσης IoT με το σύννεφο. Ένα προφανές πλεονέκτημα αυτής της ολοκλήρωσης, είναι η ευελιξία την οποία ο χρήστης παίρνει με την πρόσβαση στις υπηρεσίες που προσφέρονται από τον πάροχο cloud μέσω ενός web interface. Αυτό δίνει επίσης την ευελιξία στον πάροχο υπηρεσιών M2M να προσφέρει τις υπηρεσίες του σε περισσότερους πελάτες.

Το cloud computing είναι συνήθως ένα μοντέλο μιας πιο ευέλικτης, on-demand δικτυακής πρόσβασης σε ένα κοινόχρηστο σύνολο παραμετροποιήσιμων υπολογιστικών πόρων (π.χ., δίκτυα, servers, storage, εφαρμογές και υπηρεσίες) που μπορεί να τροφοδοτηθεί γρήγορα και διανέμεται με την ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδραση του πάροχου της υπηρεσίας. Πρόκειται για μια νέα τεχνική υπολογιστών που προσφέρει επιλογές για την ενοικίαση της υποδομής αποθήκευσης και υπηρεσίες πληροφορικής, την ενοικίαση των επιχειρηματικών διαδικασιών και ενός συνόλου εφαρμογών. Αυτή η νέα τεχνική απλοποιεί τις υπολογιστικές διαδικασίες του πελάτη με την ενοικίαση πόρων και υπηρεσιών.

Τα συστήματα Cloud βρίσκονται μέσα στο Διαδίκτυο, το οποίο είναι ένα μεγάλο ετερογενές δίκτυο με πολλές ταχύτητες, τεχνολογίες, τοπολογίες και τύπους χωρίς κεντρικό έλεγχο. Λόγω της μη ομογενούς και χαλαρής ελεγχόμενης φύσης του Διαδικτύου, υπάρχουν πολλά θέματα, ιδίως αυτά που σχετίζονται με την ποιότητα υπηρεσίας και παραμένουν άλυτα. Ένα τέτοιο ζήτημα που επηρεάζει σοβαρά την ποιότητα της υπηρεσίας είναι η καθυστέρηση του δικτύου. Οι εφαρμογές πραγματικού χρόνου με το οποίο οι χρήστες αλληλεπιδρούν άμεσα, επηρεάζονται δυσμενώς από την καθυστέρηση και το jitter που προκαλείται από καθυστέρηση στο δίκτυο.

Το άλλο μεγάλο πρόβλημα που αντιμετωπίζει το cloud computing είναι η ασφάλεια και η προστασία. Δεδομένου ότι τα συστήματα του cloud εντοπίζονται στο Διαδίκτυο, τα αιτήματα των χρηστών, οι αποκρίσεις των συστημάτων και η μετάδοση των δεδομένων πρέπει να διασχίσουν ένα μεγάλο αριθμό ενδιάμεσων δικτύων μεταξύ των χρηστών και των συστημάτων. Όταν τα δεδομένα των πελατών βρίσκονται σε ένα δημόσιο cloud, υπάρχει κίνδυνος ακεραιότητας και εμπιστευτικότητας τους. Όσο βαθύτερα τα δεδομένα μέσα στο Διαδίκτυο, τόσο μεγαλύτερος ο κίνδυνος των δεδομένων να ταξιδέψουν μια μεγάλη απόσταση προς και από τον υπολογιστή του χρήστη στο σύστημα του cloud, ακόμη και αν τα δεδομένα είναι κρυπτογραφημένα. Παρομοίως, η διαθεσιμότητα των συστημάτων cloud μπορεί επίσης να υποστεί επιθέσεις χρησιμοποιώντας διάφορες μεθόδους. Έτσι, μπορεί κανείς να δει ότι τα συστήματα cloud προς το παρόν αντιμετωπίζουν διάφορες απειλές για την ασφάλεια λόγω της ίδιας της φύσης της εφαρμογής τους στο πλαίσιο του Διαδικτύου σε συνδυασμό με την τοπολογική ανεξαρτησία.

Το cloud computing είναι μια αποτελεσματική μέθοδος για τη διατήρηση και τη διαχείριση ιδιωτικών κέντρων δεδομένων η οποία απελευθερώνει την επιχείρηση και τους πελάτες από τις προδιαγραφές τόσων πολλών λεπτομερειών που μπορούν να δημιουργήσουν πρόβλημα στις ευαίσθητες σε καθυστέρηση εφαρμογές και απαιτούν μεγάλο αριθμό κόμβων, προκειμένου να ανταποκριθεί στις απαιτήσεις καθυστέρησης. Το IoT όμως

απαιτεί την υποστήριξη της κινητικότητας και μια μεγάλη κλίμακα Γεω-κατανομής εκτός από την επίγνωση της τοποθεσίας και των δυνατοτήτων χαμηλής καθυστέρησης. Ως εκ τούτου, χρειαζόμαστε μια νέα πλατφόρμα για να καλύψει όλες αυτές τις απαιτήσεις. Υπάρχει μια νέα πλατφόρμα που προσφέρει ένα νέο σύνολο διαδικτυακών εφαρμογών και υπηρεσιών προς τους τελικούς χρήστες, με την επέκταση της πλατφόρμας cloud. Αυτή η νέα πλατφόρμα ονομάζεται Fog Computing και είναι επίσης γνωστό ως Fogging.

1.2 ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Στο δεύτερο κεφάλαιο παρουσιάζονται το Cloud Computing και το Διαδίκτυο των πραγμάτων (IoT) και μελετούνται οι δομές και οι αρχιτεκτονικές τους.

Στο τρίτο κεφάλαιο παρουσιάζεται η προτεινόμενη λύση στο πρόβλημα που ορίσαμε σε αυτό το κεφάλαιο που προκύπτει ότι είναι μια νέα αρχιτεκτονική το Fog Computing.

Στο τέταρτο κεφάλαιο παρουσιάζεται η υλοποίηση μια εφαρμογής που λύνει το πρόβλημα του πρώτου κεφαλαίου χρησιμοποιώντας την αρχιτεκτονική της υπολογιστικής ομίχλης.

Στο πέμπτο κεφάλαιο η εφαρμογή αναλύεται με την χρήση σεναρίων με σκοπό την κατανόηση της λειτουργίας του συστήματος μέσω οθονών.

Στο έκτο κεφάλαιο παρουσιάζονται τα συμπεράσματα της διπλωματικής εργασίας και τα αποτελέσματα της προτεινόμενης λύσης που παρουσιάστηκε.

1.3 ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η διπλωματική αυτή συγκεντρώνει (στο θεωρητικό μέρος) τις απαραίτητες γνώσεις που πρέπει να έχει κάποιος για να κατανοήσει το Fogging. Στο δεύτερο μέρος παρουσιάζεται η υλοποίηση μιας εφαρμογής με την αρχιτεκτονική της υπολογιστικής ομίχλης έχοντας ως στόχο να αναλύσει

τον τρόπο χρήσης της καθώς και να αναδείξει τα νέα χαρακτηριστικά που εισάγει σε σχέση με το cloud computing.

ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

ΕΙΣΑΓΩΓΗ

Πριν προχωρήσουμε στην υλοποίηση της διπλωματικής εργασίας, πρέπει να κατανοήσουμε καλύτερα μια σειρά από θεωρητικά θέματα. Για το λόγο αυτό ανατρέξαμε στη βιβλιογραφία (ελληνική και ξένη) που υπάρχει κυρίως στο διαδίκτυο. Τα αποτελέσματα αυτής της έρευνας τα παρουσιάζουμε στο κεφάλαιο αυτό.

2. CLOUD COMPUTING

Η ταχεία εξέλιξη των δυνατοτήτων του hardware σε συνδυασμό με την γρήγορη wide area επικοινωνία και διαθεσιμότητα των virtualization solutions επιτρέπουν νέα operating μοντέλα για το IT. Το Cloud Computing [2] είναι ένα από αυτά. Το Cloud Computing σίγουρα δεν είναι μια επανάσταση στο τομέα της πληροφορικής αλλά σίγουρα είναι μια εξέλιξη της πληροφορικής.

Χαρακτηριστικά του Cloud Computing:

- ✓ Virtual
- ✓ Λογισμικό, βάσεις δεδομένων, web servers, λειτουργικά συστημάτων, αποθήκευση και δικτύωση ως virtual servers.
- ✓ On demand
- ✓ πρόσθεση και αφαίρεση επεξεργαστών, μνήμη, network bandwidth, αποθηκευτικός χώρος.

Σήμερα το Cloud Computing υπόσχεται αξιόπιστες υπηρεσίες που παρέχονται μέσω των data centers επόμενης γενιάς τα οποία είναι χτισμένα πάνω σε virtualized compute και αποθηκευτικές τεχνολογίες. Οι καταναλωτές θα είναι σε θέση να αποκτήσουν πρόσβαση σε εφαρμογές και δεδομένα on demand από ένα “Cloud” από οποιοδήποτε μέρος στο κόσμο.

Οι φυσικοί πόροι ανήκουν στην υπηρεσία πάροχο. Το κόστος μίσθωσης είναι μεταβλητό ανάλογα κάθε φορά από την κατανάλωση του πόρου. Η συνολική επιβάρυνση ορίζεται σύμφωνα με τη χρήση της κάθε υπηρεσίας.

Το Cloud Computing μειώνει σημαντικά το κόστος γιατί η οργάνωση που το θέτει σε εφαρμογή:

- I. δεν χρειάζεται να αγοράσει servers για να φιλοξενήσει τις εφαρμογές,
- II. δεν έχει έξοδα φυσικής υποδομής, όπως κατανάλωση ενέργειας και κόστος εξυπηρέτησης και
- III. λιγότερο προσωπικό είναι απαραίτητο για λόγους συντήρησης.

2.1 Η εξέλιξη του Cloud Computing

Για να καταλάβουμε τι είναι και τι δεν είναι το Cloud Computing [2], είναι σημαντικό να καταλάβουμε πώς έχει εξελιχθεί αυτό το υπολογιστικό μοντέλο. Όπως έχει σημειώσει ο Alvin Toffler στο διάσημο βιβλίο του, *The third wave* (Bantam, 1980), ο πολιτισμός έχει εξελιχθεί σε κύματα (τρία από αυτά μέχρι σήμερα: το πρώτο κύμα ήταν οι αγροτικές κοινωνίες, το δεύτερο ήταν η βιομηχανική εποχή, και το τρίτο είναι η εποχή της πληροφορίας). Μέσα στο κάθε κύμα έχουν υπάρξει μικρότερα κύματα. Σε αυτήν την μεταβιομηχανική εποχή της πληροφορίας, βρισκόμαστε τώρα στην αρχή αυτής της εποχής που πολλοί νοιώθουν ότι θα είναι η εποχή του Cloud Computing.

Όπως ανέφερα νωρίτερα, μέσα σε κάθε κύμα υπάρχουν και τα μικρότερα κύματα, και υπάρχουν ήδη αρκετά μέσα στην εποχή της πληροφορίας, όπως δείχνει και το Σχήμα 2. Ξεκινήσαμε με mainframe υπολογιστές και προχωρήσαμε στους mini υπολογιστές, στους προσωπικούς

υπολογιστές, και ούτω καθεξής, και τώρα εισερχόμαστε στο Cloud Computing.



Εικόνα 1 : Σύννεφα του Cloud Computing

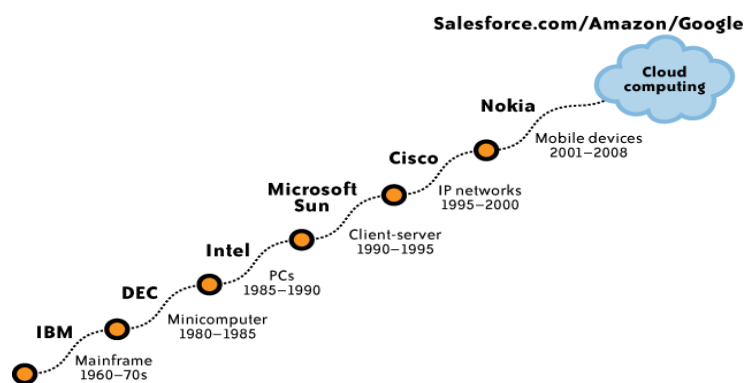
Μια άλλη άποψη δείχνει ότι το Cloud Computing το ίδιο είναι μια λογική εξέλιξη της πληροφορικής. Η εικόνα 3 δείχνει το Cloud Computing και τους παρόχους υπηρεσιών Cloud (CSP) ως επεκτάσεις του μοντέλου Internet service provider (ISP).

Στην αρχή (ISP 1.0), ISP's γρήγορα πολλαπλασιάστηκαν για να παρέχουν πρόσβαση στο διαδίκτυο σε οργανισμούς και ιδιώτες. Αυτές οι πρώιμες ISP's απλώς προβλέπανε τη σύνδεση στο Internet για χρήστες και μικρές επιχειρήσεις, τις πιο πολλές φορές με dial-up τηλεφωνική υπηρεσία. Καθώς η πρόσβαση στο Διαδίκτυο έγινε ένα εμπόρευμα, οι ISP's ενωθήκανε και ψάξανε για άλλες υπηρεσίες προστιθέμενης αξίας, όπως η παροχή πρόσβασης στο ηλεκτρονικό ταχυδρομείο και σε servers στις εγκαταστάσεις τους (ISP 2.0). Αυτή η έκδοση οδήγησε γρήγορα σε εξειδικευμένες εγκαταστάσεις για το server hosting οργανισμών, μαζί με την υποδομή για την υποστήριξη των server και των εφαρμογών που τρέχουν σε αυτούς. Αυτές οι εξειδικευμένες εγκαταστάσεις είναι γνωστές ως collocation facilities (ISP 3.0). Αυτές οι εγκαταστάσεις είναι ένα είδος data center, όπου πολλοί πελάτες εντοπίζουν δίκτυα, server, και εργαλεία αποθήκευσης και διασυνδέονται σε

μια ποικιλία τηλεπικοινωνιών και άλλων παροχών υπηρεσιών δικτύων με ελάχιστο κόστος και πολυπλοκότητα. Όσο οι collocation εγκαταστάσεις εξαπλώνονταν και εμπορευματοποιήθηκαν, το επόμενο βήμα στην εξέλιξη ήταν η δημιουργία των Application service providers (ASPs), οι οποίοι επικεντρώθηκαν σε μια υψηλότερης

* Wikipedia definition of collocation facility: http://en.wikipedia.org/wiki/Collocation_facility.

προστιθέμενης αξίας παροχή εξειδικευμένων εφαρμογών για οργανισμούς, και όχι μόνο στην υπολογιστική υποδομή (ISP 4.0). Στο ASPs τυπικά ανήκει και λειτουργεί το λογισμικό των εφαρμογών που παρέχετε, καθώς και η απαραίτητη υποδομή.

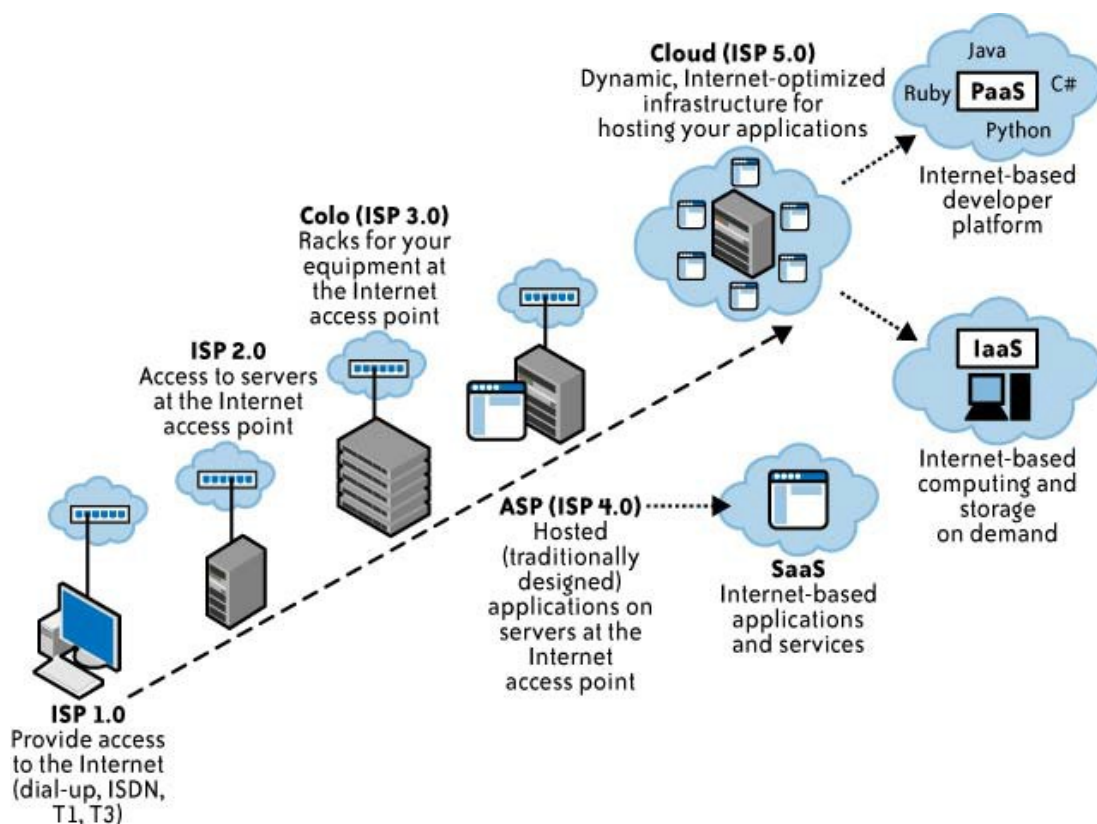


Εικόνα 2 : Μικρότερα κύματα μέσα στην εποχή της πληροφορίας.

Αν και το ASPs μπορεί να φαίνεται παρόμοιο με ένα μοντέλο παροχής υπηρεσιών του Cloud Computing το οποίο αναφέρεται ως Software-as-a-service (Seas), υπάρχει μια σημαντική διαφορά στο πώς παρέχονται αυτές οι υπηρεσίες. Από το ASPs πααρέχονται υπηρεσίες σε πολλαπλούς πελάτες, αυτό το έκαναν μέσω ειδικών υποδομών. Δηλαδή κάθε πελάτης έχει τη δική του ιδιαίτερη περίπτωση της αίτησης, αυτή η περίπτωση συνήθως τρέχει σε ειδικό host ή server. Η σημαντική διαφορά μεταξύ Seas παρόχων και ASPs παρόχων είναι ότι οι Seas πάροχοι προσφέρουν πρόσβαση σε εφαρμογές σε μια κοινόχρηστη υποδομή .

Το Cloud Computing (ISP 5.0) ορίζει το μοντέλο SPI, το οποίο είναι γενικά αποδεκτό για παροχή SaaS, platform-as-a-service (PaaS), και για infrastructure-as-a-service (IaaS).

Με την προσοχή ολόενα να μεγαλώνει, με αυτή τη διαφημιστική εκστρατεία, που καταβάλλεται για το Cloud Computing, όλο και περισσότερες εταιρείες ισχυρίζονται ότι είναι “cloudy”. Ξαφνικά, πολλές εταιρείες διεκδικούν να λειτουργούν σε cloud. Ομοίως, ένας αριθμός από computer groups έχουν ανακοινώσει τις προσπάθειές τους για την



Εικόνα 3 : Εξέλιξη του Cloud Computing

προώθηση κάποιων πτυχών του Cloud Computing. Μερικές από αυτές τις ομάδες είναι εγκατεστημένες (π.χ. National Institute of Standards and Technology προσπαθούν να προωθήσουν την τυποποίηση στο Cloud Computing), και μερικές από αυτές είναι καινούργιες, αφού προκύψανε με την εμφάνιση αυτού του νέου μοντέλου υπολογισμού (π.χ. η προώθηση του Cloud Security Alliance’s για την ασφάλεια στο Cloud Computing, ή η

προώθηση του Open Cloud Manifesto's της διαλειτουργικότητας του cloud). Πολλές άλλες ομάδες έχουν ανακοινώσει επίσης προσπάθειες αφιερωμένες στο Cloud Computing, όπως η Distributed Management Task Force (DMTF), την Technology Association of America, μια βιομηχανία υψηλής τεχνολογίας, και το Jericho Forum, μια ένωση διεθνής ασφάλειας πληροφοριών, μεταξύ πολλών άλλων.

Το cloud computing αναφέρεται τόσο στις εφαρμογές που παρέχονται ως υπηρεσίες μέσω του Διαδικτύου όσο και στο υλικό και τα συστήματα λογισμικού στα data centers που παρέχουν αυτές τις υπηρεσίες. Οι υπηρεσίες αυτές εδώ και καιρό αναφέρονται ως Software-as-a-Service (SaaS). Το software και το hardware του data center είναι αυτό που θα ονομάζουμε cloud. Όταν ένα cloud γίνεται διαθέσιμο στο ευρύ κοινό με pay-as-you-go τρόπο, τότε λέμε ότι το cloud είναι Public Cloud. Η υπηρεσία από την οποία πωλείται είναι το Utility Computing. Χρησιμοποιούμε τον όρο Private Cloud όταν αναφερόμαστε στα εσωτερικά data centers μιας επιχείρησης ή ενός οργανισμού που δεν διατίθενται στο ευρύ κοινό. Οπότε, Cloud Computing είναι το άθροισμα των SaaS και Utility Computing, δεν περιέχει όμως τα Private Clouds. Οι άνθρωποι μπορούν να είναι χρήστες ή παροχείς SaaS, ή χρήστες ή παροχείς Utility Computing. Εμείς επικεντρωνώμαστε στους SaaS παρόχους (χρήστες του cloud) και Cloud παροχείς, οι οποίοι έχουν λάβει λιγότερη προσοχή από ότι οι SaaS χρήστες. Από την άποψη του hardware, τρεις πτυχές είναι νέες στο Cloud Computing.

- 1) Η ψευδαίσθηση του άπειρου των υπολογιστικών πόρων που είναι διαθέσιμοι για demand, εξαλείφοντας έτσι την ανάγκη για το Cloud Computing.
- 2) Η κατάργηση της up-front δέσμευσης από τους χρήστες του Cloud, επιτρέποντας έτσι στις εταιρείες να ξεκινήσουν μικρούς και αυξανόμενους hardware πόρους μόνο όταν υπάρχει αύξηση των αναγκών τους.
- 3) Η δυνατότητα πληρωμής για τη χρήση υπολογιστικών πόρων που απαιτούνται σε βραχυπρόθεσμη βάση (π.χ. επεξεργαστές με την ώρα και

αποθήκευση με την μέρα), και η απελευθέρωσή τους όταν απαιτείται, έτσι έχουμε κερδοφόρα διατήρηση, αφήνοντας τις μηχανές και την αποθήκη ανενεργές όταν δεν χρειάζονται πλέον.

Κάθε εφαρμογή χρειάζεται ένα υπολογιστικό μοντέλο, ένα μοντέλο αποθήκευσης και ένα μοντέλο επικοινωνίας. Η στατιστική πολυπλεξίας που είναι απαραίτητη για την επίτευξη της ελαστικότητας και της ψευδαίσθησης της άπειρης χωρητικότητας που απαιτεί κάθε ένας από αυτούς τους πόρους για να μπορεί εικονικά να κρύψει το π'σ εφαρμόζεται η πολυπλεξία και η κοινοποίηση των εφαρμογών. Η αποψη μου είναι ότι διαφορετικές utility computing προσφορές μπορούν να διακριθούν με βάση το αφαιρετικό επίπεδο που παρουσιάζουν στο προγραμματιστή και το επίπεδο διαχείρισης των πόρων.

Πότε είναι προτιμότερο να τρέχει ένα Utility Computing σε Private Cloud; Η πρώτη περίπτωση είναι όταν η ζήτηση για την υπηρεσία ποικίλλει με τον χρόνο. Η τροφοδότηση ενός Data center για το φορτίο αιχμής θα πρέπει να διατηρείται λίγες μέρες κάθε μήνα οδηγώντας έτσι σε επαρκή αξιοποίηση σε άλλους χρόνους, για παράδειγμα. Αντ'αυτού, το Cloud Computing επιτρέπει σε έναν οργανισμό να πληρώνει με την ώρα για τους υπολογιστικούς πόρους, αυτο δυνητικά μπορεί να οδηγήσει σε εξοικονόμηση κόστους, ακόμη και αν το ωρομίσθια χρέωση μιας μηχανής από ένα cloud πάροχο είναι μεγαλύτερη από το ποσοστό του να κατέχεις μια δικιά σου. Μια δεύτερη περίπτωση είναι όταν η ζήτηση είναι γνωστή εκ των προτέρων. Για παράδειγμα, μια νεοσύστατη εταιρεία στο διαδίκτυο θα πρέπει να υποστηρίζει ένα spike που είναι σε ζήτηση όταν γίνεται δημοφιλής, ακολουθούμενο ενδεχομένως από μια μείωση που δημιουργείται μόλις ένας επισκέπτης φύγει. Τέλος, οι οργανισμοί που εκτελούν batch analytics μπορούν να χρησιμοποιούν το "associativity κόστος" του Cloud Computing έτσι ώστε να ολοκληρώνουν τους υπολογισμούς τους γρηγορότερα από: το να χρησιμοποιούν 1000 EC2 μηχανές για 1 ώρα που κοστίζει το ίδιο με το να χρησιμοποιούν 1 μηχανή για 1000 ώρες. Για την πρώτη περίπτωση μιας επιχείρησης στο διαδίκτυο με ποικίλη ζήτηση στην πάροδο του χρόνου και

έσοδα ανάλογα με τις ώρες χρήσης, έχουμε συγκεντρώσει το tradeoff στην παρακάτω εξίσωση:

$$\text{User hours}(\text{cloud}) * (\text{revenue} - \text{cost}(\text{cloud})) > \text{User hours}(\text{datacenter}) * (\text{revenue} - \text{cost}(\text{datacenter}) / \text{Utilization})$$

Η αριστερή πλευρά πολλαπλασιάζει τα καθαρά έσοδα ανά χρήστη την ώρα με τον αριθμό των χρηστών ανά ώρα, δίνοντας το αναμενόμενο κέρδος από τη χρήση του Cloud Computing. Η δεξιά πλευρά εκτελεί τον ίδιο υπολογισμό για ένα datacenter σταθερού μεγέθους από factoring στη μέση τιμή του utilization, συμπεριλαμβανομένου του nonpeak φόρτου εργασίας του datacenter. Όποια πλευρά είναι μεγαλύτερη αποτελεί την καλύτερη ευκαιρία για μεγαλύτερο κέρδος. [2]

2.2 Θεωρία

Το Cloud Computing αναφέρεται τόσο στις εφαρμογές που παρέχονται ως υπηρεσίες μέσω του διαδικτύου όπως και στο υλικό και τα συστήματα λογισμικού στα datacenters που παρέχουν αυτές τις υπηρεσίες. Ο ορισμός του Cloud Computing βασίζεται σε πέντε χαρακτηριστικά: multitenancy (κοινόχρηστοι πόροι), τεράστιες δυνατότητες κλιμάκωσης, ελαστικότητα, pay as you go, και self-provisioning των πόρων.

Multitenancy (κοινόχρηστοι πόροι)

Σε αντίθεση με προηγούμενα μοντέλα υπολογιστών, τα οποία ανέλαβαν αποκλειστικούς πόρους (οι υπολογιστικές εγκαταστάσεις που προορίζονται για ένα μόνο χρήστη ή ιδιοκτήτη), το Cloud Computing βασίζεται σε ένα επιχειρηματικό μοντέλο στο οποίο οι πόροι διαμοιράζονται (δηλαδή, πολλοί χρήστες χρησιμοποιούν τον ίδιο πόρο), στα επίπεδα δικτύου, host και στο επίπεδο εφαρμογής.

Massive scalability

Παρά το γεγονός ότι οι οργανισμοί μπορεί να έχουν εκατοντάδες ή χιλιάδες συστήματα, το Cloud Computing παρέχει τη δυνατότητα να κλιμακωθούν σε δεκάδες χιλιάδες συστήματα, καθώς επίσης και την ικανότητα να κλιμακώνει μαζικά το εύρος ζώνης και το χώρο αποθήκευσης.

Ελαστικότητα

Οι χρήστες μπορούν γρήγορα να αυξάνουν και να μειώνουν τους υπολογιστικούς τους πόρους, ανάλογα με τις απαιτήσεις τους, καθώς και να απελευθερώνουν πόρους για άλλες χρήσεις, όταν δεν είναι πλέον απαραίτητοι.

Pay as you go

Οι χρήστες πληρώνουν μόνο για τους πόρους που έχουνε πράγματι χρησιμοποιήσει και μόνο για το χρόνο που τα χρειάζονται.

Self-provisioning των πόρων

Οι χρήστες αυτοεξυπηρετούνται με τους πόρους, όπως πρόσθετα συστήματα (δυνατότητα επεξεργασίας λογισμικού και αποθήκευσης) και τους πόρους δικτύου. Ένα από τα χαρακτηριστικά του Cloud Computing είναι η ελαστικότητα των πόρων. Αυτή η δυνατότητα του cloud επιτρέπει στους χρήστες να αυξήσουν ή να μειώσουν τους υπολογιστικούς τους πόρους, όπως απαιτείται, όπως δείχνει η παρακάτω εικόνα.

Υπάρχει πάντα μια επίγνωση της αναφοράς των υπολογιστικών πόρων, αλλά η πρόβλεψη μελλοντικών αναγκών είναι δύσκολη, ειδικά όταν οι απαιτήσεις αλλάζουν συνεχώς. Το Cloud Computing αποτελεί ένα μέσο το οποίο προσφέρει πόρους on demand στο IT και address spikes σε χρήση.

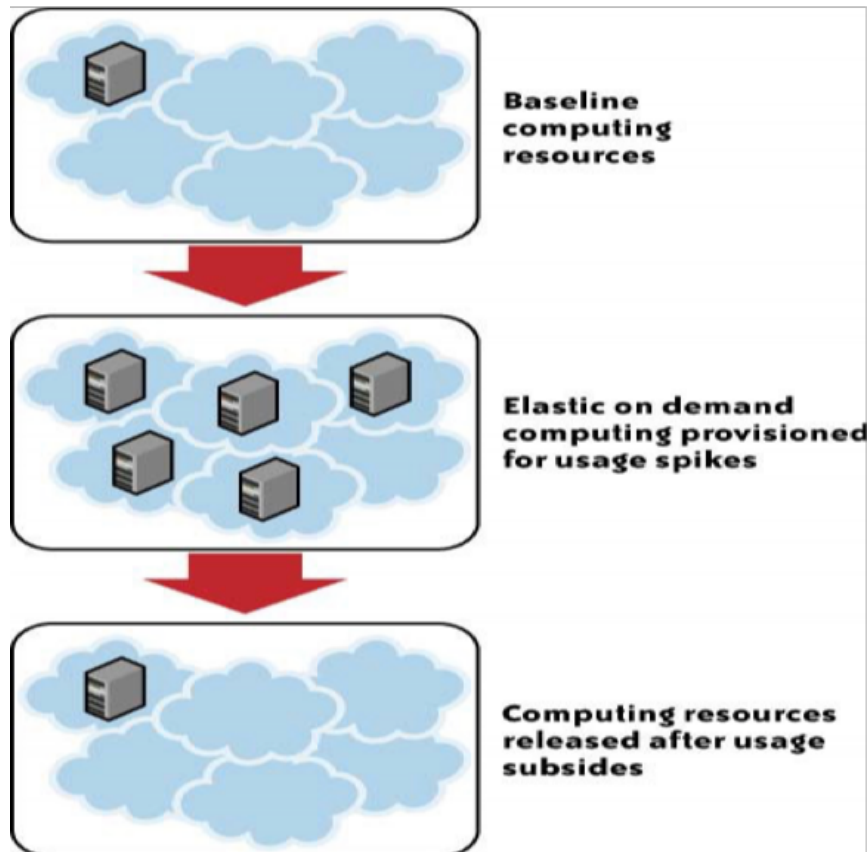
Το ενδιαφέρον στο cloud αυξάνεται γιατί οι λύσεις cloud παρέχουν στους χρήστες πρόσβαση σε δυνατότητες supercomputer σε ένα μικρό μέρος του κόστους για την αγορά μια τέτοιας ολοκληρωτικής λύσης. Το πιο

σημαντικό είναι ότι αυτές οι λύσεις μπορούν να αποκτηθούν on demand. Το δίκτυο γίνεται ο supercomputer στο cloud όπου οι χρήστες μπορούν να αγοράσουνε ότι χρειάζονται, όταν το χρειάζονται. Το Cloud Computing αναγνωρίζει πού παρέχονται οι κλιμακούμενες IT-enabled δυνατότητες σαν υπηρεσία στους πελάτες που χρησιμοποιούν τεχνολογίες διαδικτύου.

Το Cloud Computing έχει προκαλέσει σημαντικό ενδιαφέρον στην αγορά και προβλέπεται να αναπτυχθεί πολύ, όπως φαίνεται και στο σχήμα 3, το οποίο υπογραμμίζει το πρόσφατο αξιοσημείωτο cloud που λανσάρεται και τα τρέχοντα και προβλεπόμενα έσοδα για τις cloud-based υπηρεσίες.

Το Cloud Computing αναμένεται να γίνει μια σημαντική κινητήρια δύναμη στην ανάπτυξη των παγκόσμιων IT δαπανών. Στην πραγματικότητα, οι υπηρεσίες cloud αναμένεται να αυξηθούν με ένα σύνθετο ρυθμό ετήσιας ανάπτυξης (CAGR) 27% και να φτάσουν τα 42 δισεκατομύρια δολάρια μέχρι το 2012. Οι δαπάνες για μη cloud IT υπηρεσίες αναμένεται να αυξηθούν σε ένα μέσο ετήσιο ρυθμό αύξησης 5%, σύμφωνα με την IDC.

Το φάσμα των συσκευών για την πρόσβαση στο cloud έχει επεκταθεί τα τελευταία χρόνια. Υπολογιστές στο σπίτι, υπολογιστές επιχειρήσεων, υπολογιστές δικτύου, συσκευές κινητής τηλεφωνίας, προσαρμοσμένες συσκευές χειρός, και στατικές συσκευές



Εικόνα 4 : Αυξομείωση υπολογιστικών πόρων cloud computing

(συμπεριλαμβανομένων των ψυγείων) είναι όλα σε απευθείας σύνδεση. Είναι ενδιαφέρον το πως, η ανάπτυξη του iPhone και ο πολλαπλασιασμός των διαθέσιμων εφαρμογών από το App Store της Apple, δείχνει μια βελτίωση στην άποψη πρόσβασης στο cloud. Για παράδειγμα μπορούμε τώρα πια να χρησιμοποιήσουμε το Skype μέσω του iPhone, φέρνοντας έτσι το peer-to-peer δίκτυο πολύ πιο κοντά στους χρήστες, και η Salesforce.com εισήγαγε μια εφαρμογή που επιτρέπει στους χρήστες να έχουν πρόσβαση στις υπηρεσίες της από το iPhone, καθώς και από smartphone άλλων εταιρειών πέραν της Apple.

Browsers και thin clients

Οι χρήστες πολλαπλών τύπων συσκευών μπορούν να έχουν πρόσβαση τώρα σε εφαρμογές και πληροφορίες από οπουδήποτε μπορούν να φορτώσουν έναν browser. Πράγματι, οι browsers (προγράμματα

περιήγησης) γίνονται όλο και πιο εξελιγμένα. Σε εφαρμογές επιχειρήσεων, όπως η SAP και η Oracle, μπορούμε να έχουμε πρόσβαση μέσω ενός browser interface. Ο γενικός πληθυσμός έχει εξοικειωθεί με τη λειτουργία του προγράμματος περιήγησης και χρησιμοποιεί μια διακριτή εφαρμογή, όπου το πλαίσιο είναι έξυπνο, χωρίς να απαιτείται κατάρτιση ή άλλοι οδηγοί χρήσης.

High-speed broadband access

Ένα κρίσιμο συστατικό του cloud είναι το ευρυζωνικό δίκτυο, το οποίο προσφέρει τα μέσα για να συνδεθούν εξαρτήματα και παρέχει μία από τις πιο ουσιαστικές διαφορές στην έννοια του Cloud Computing 30 χρόνια πριν. Η ευρυζωνική πρόσβαση είναι πλέον ευρέως διαθέσιμη παγκόσμια, ιδιαίτερα σε μητροπόλεις. Υπάρχει ασύρματη πρόσβαση (π.χ. WiFi, κινητά, αναδυόμενες WiMAX), η οποία έχει συστήσει κινητές συσκευές ως σημεία εισόδου σε IT πόρους της επιχείρησης και του cloud.

Data centers and server farms

Τα data centers και τα server farms απαιτούν Cloud-based υπηρεσίες και μεγάλη υπολογιστική ικανότητα και φιλοξενούνται σε data centers και server farms. Αυτά τα κατακεκομμένα data centers και τα server farms εκτείνονται σε πολλές τοποθεσίες και μπορεί να συνδεθούν μέσω συνδέσεις Internet παρέχοντας κατακεκομμένα υπολογιστικά συστήματα και δυνατότητες παροχής υπηρεσιών. Μια σειρά από παραδείγματα σήμερα αποδεικνύουν την ευελιξία και την επεκτασιμότητα της ισχύος του Cloud Computing. Για παράδειγμα η Google έχει συνδεθεί με έναν πολύ μεγάλο αριθμό από ακριβούς servers για να παρέχει τεράστια ευελιξία και ισχύ. Το Amazon's Elastic Compute Cloud (EC2), παρέχει virtualization στο data center για τη δημιουργία τεράστιου αριθμού από εικονικές περιπτώσεις για υπηρεσίες που έχουν ζήτηθεί. Η Salesforce.com παρέχει SaaS στην ευρεία πελατειακή βάση της, με την ομαδοποίηση των πελατών της σε ομάδες για να ενεργοποιήσει την επεκτασιμότητα και την ευελιξία.

Storage Devices

Μειώνοντας το κόστος αποθήκευσης και την ευελιξία με την οποία μπορεί να αναπτυχθεί η αποθήκευση, έχει αλλάξει όλο το τοπίο της αποθήκευσης. Η σταθερή άμεση συσκευή αποθήκευσης πρόσβασης (DASD – direct access storage device) έχει αντικατασταθεί με δίκτυα περιοχής αποθήκευσης (SAN), τα οποία έχουν μειώσει το κόστος και επιτρέπουν πολύ περισσότερο ευελιξία στην αποθήκευση επιχειρήσεων. Το λογισμικό SAN διαχειρίζεται την ενσωμάτωση των συσκευών αποθήκευσης και μπορεί να εκχωρεί ανεξάρτητα χώρο αποθήκευσης on demand σε μια σειρά απο συσκευές.

2.3 Το μοντέλο Software-As-a-Service (SaaS)

Οι παραδοσιακές μέθοδοι αγοράς λογισμικού που συμμετέχει ο πελάτης στη φόρτωση του λογισμικού στο δικό του hardware με αντάλλαγμα τα τέλη αδείας (κεφαλαιουχική δαπάνη, που είναι γνωστή ως κόστος επένδυσης). Ο πελάτης θα μπορούσε επίσης να αγοράσει ένα συμβόλαιο συντήρησης για να λαμβάνει patches για το λογισμικό ή άλλες υπηρεσίες υποστήριξης. Ο πελάτης ασχολείται με τη συμβατότητα των λειτουργικών συστημάτων, την εγκατάσταση των patch, και τη τήρηση των συμβάσεων άδειας χρήσης.

Σε ένα μοντέλο SaaS, ο πελάτης δεν αγοράζει λογισμικό, αλλά το νοικιάζει για χρήση σε ένα συνδρομητικό ή pay-per-use μοντέλο (μια λειτουργική δαπάνη, που είναι γνωστή ως OpEx). Σε ορισμένες περιπτώσεις, η υπηρεσία είναι δωρεάν για περιορισμένη χρήση. Τυπικά, η αγορασμένη υπηρεσία είναι πλήρης από hardware, software, και προοπτική υποστήριξης. ο χρήστης έχει πρόσβαση στην υπηρεσία μέσω οποιασδήποτε εξουσιοδοτημένης συσκευής. Σε ορισμένες περιπτώσεις, προπαρασκευαστικές εργασίες που απαιτούνται για τη δημιουργία της εταιρείας – συγκεκριμένα στοιχεία για την υπηρεσία να χρησιμοποιηθούν πλήρως και ενδεχομένως να ενσωματωθούν με άλλες εφαρμογές που δεν βρίσκονται στη SaaS πλατφόρμα.

Βασικά πλεονεκτήματα του μοντέλου SaaS:

- Το SaaS δίνει τη δυνατότητα στον οργανισμό να αναθέτει τη φιλοξενία και τη διαχείριση των εφαρμογών σε ένα τρίτο μέρος (πωλητής λογισμικού και παροχής υπηρεσιών) ως μέσο για τη μείωση του κόστους των αδειών λογισμικού, servers, και άλλες υποδομές και προσωπικές απαιτήσεις που απαιτούνται για την φιλοξενία της εφαρμογής εσωτερικά.
- Το SaaS επιτρέπει στους προμηθευτές λογισμικού να ελέγχουν και να περιορίζουν τη χρήση, απαγορεύοντας την αντιγραφή και τη διανομή, και διευκολύνοντας τον έλεγχο όλων των παράγωγων εκδόσεων του λογισμικού τους. Ο συγκεντρωτικός έλεγχος του SaaS επιτρέπει συχνά τον πωλητή ή τον προμηθευτή να δημιουργήσει μια συνεχή ροή εσόδων με πολλαπλές επιχειρήσεις, και τους χρήστες να μπορούν να κάνουν preloading του λογισμικού σε κάθε συσκευή από οποιονδήποτε κατασκευαστή.
- Οι αιτήσεις παροχής που χρησιμοποιούν το μοντέλο SaaS χρησιμοποιούν συνήθως προσέγγιση παράδοσης one-to-many, με το Web ως υποδομή. Ο τελικός χρήστης μπορεί να έχει πρόσβαση σε μια εφαρμογή SaaS μέσω ενός web browser.

Ορισμένοι πωλητές SaaS παρέχουν τη δική τους διασύνδεση που έχει σχεδιαστεί για να υποστηρίζει χαρακτηριστικά που είναι μοναδικά για τις εφαρμογές τους.

- Μια τυπική εγκατάσταση SaaS δεν απαιτεί κανένα υλικό και μπορεί να τρέξει πάνω από το υπάρχον internet υποδομής πρόσβασης. Μερικές φορές μπορεί να απαιτούνται αλλαγές στους κανόνες και τις ρυθμίσεις του τείχους προστασίας για να καταστεί δυνατή η ομαλή λειτουργία της εφαρμογής SaaS.
- Η διαχείριση μιας εφαρμογής SaaS υποστηρίζεται από τον προμηθευτή και την προοπτική του τελικού χρήστη, σύμφωνα με την οποία μια

εφαρμογή SaaS μπορεί να ρυθμιστεί χρησιμοποιώντας ένα API, αλλά οι εφαρμογές SaaS δεν μπορούν να εφαρμοστούν πλήρως. □Μια τυπική υπηρεσία SaaS είναι SaaS μέσω ενός δημόσιου δικτύου, στην οποία μια SaaS-based εφαρμογή είναι να παραδίδονται μέσω του Διαδικτύου στο firewall του εκάστοτε οργανισμού. Η πιο σημαντική αρχιτεκτονική διαφορά μεταξύ του παραδοσιακού μοντέλου του λογισμικού και του μοντέλου SaaS είναι ο αριθμός των ενοικιαστών που η εφαρμογή υποστηρίζει. Το παραδοσιακό λογισμικό μοντέλο είναι ένα απομονωμένο, single-tenant μοντέλο, το οποίο σημαίνει ότι ένας πελάτης αγοράζει μια εφαρμογή λογισμικού και το εγκαθιστά σε έναν διακομιστή. Ο διακομιστής τρέχει μόνο τη συγκεκριμένη εφαρμογή και μόνο για του αυτού του ενός πελάτη end user group. Το μοντέλο SaaS είναι ένα multitenant αρχιτεκτονικό μοντέλο. □Οι λύσεις που προσφέρουν τα SaaS είναι πολύ διαφορετικές από το φορέα φορέα παροχής υπηρεσιών της εφαρμογής (ASP). Δύο είναι οι κυριότερες αιτίες για αυτό:

- Οι εφαρμογές ASP είναι παραδοσιακά, single-tenant εφαρμογές, αλλά φιλοξενούνται αλλά φιλοξενούνται από τρίτους. Είναι client/server εφαρμογές στις οποίες προσθέτονται HTML frontends για να επιτρέπεται η απομακρυσμένη πρόσβαση στην αίτηση.
- Οι εφαρμογές ASP δεν είναι γραμμένες ως Net-native εφαρμογές. Ως αποτέλεσμα, οι επιδόσεις τους μπορεί να είναι κακές, και οι ενημερώσεις των εφαρμογών δεν είναι καλύτερες από τις self-managed premise-based εφαρμογές. Συγκριτικά, οι SaaS εφαρμογές είναι multitenant εφαρμογές οι οποίες φιλοξενούνται από έναν προμηθευτή με τεχνογνωσία στις εφαρμογές που έχουν σχεδιαστεί ως Net-native εφαρμογές και ενημερώνονται σε συνεχή βάση.

2.4 Το Platform-As-a-Service μοντέλο

Σε ένα PaaS μοντέλο, ο προμηθευτής προσφέρει ένα περιβάλλον ανάπτυξης για προγραμματιστές εφαρμογών, οι οποίοι αναπτύσσουν εφαρμογές και προσφέρουν τις υπηρεσίες αυτές μέσω της πλατφόρμας του παρόχου. Ο πάροχος συνήθως αναπτύσσει εργαλεία και πρότυπα για την ανάπτυξη, και κανάλια για τη διανομή και τη πληρωμή. Ο πάροχος λαμβάνει συνήθως μια πληρωμή για να παρέχει τη πλατφόρμα, τις πωλήσεις και τις υπηρεσίες διανομής. Αυτό επιτρέπει την ταχεία διάδοση εφαρμογών λογισμικού, δεδομένου του χαμηλού κόστους εισόδου και της αξιοποίησης των υφιστάμενων διαύλων για την απόκτηση πελατών.

Το PaaS είναι μια παραλλαγή του SaaS σύμφωνα με την οποία το περιβάλλον ανάπτυξης προσφέρεται ως υπηρεσία. Οι προγραμματιστές χρησιμοποιούν τα δομικά στοιχεία (π.χ. προκαθορισμένα τμήματα του κώδικα) του προμηθευτή δηλαδή το περιβάλλον ανάπτυξης, για να δημιουργήσουν τις δικές τους εφαρμογές.

Λύσεις για το PaaS είναι οι πλατφόρμες ανάπτυξης για τις οποίες το εργαλείο ανάπτυξης φιλοξενείται το ίδιο στο cloud και η πρόσβαση σε αυτό γίνεται μέσω ενός browser. Με PaaS, οι προγραμματιστές μπορούν να χτίζουν web εφαρμογές χωρίς να χρειάζεται η εγκατάσταση κανενός εργαλείου στον υπολογιστή τους, και στη συνέχεια μπορούν να αναπτύξουν εκείνες τις εφαρμογές που δεν χρειάζονται εξειδικευμένες δεξιότητες για τη διαχείριση του συστήματος.

Τα συστήματα PaaS είναι χρήσιμα επειδή επιτρέπουν ανεξάρτητους προγραμματιστές και νεοσύστατες επιχειρήσεις να αναπτύξουν web-based εφαρμογές χωρίς το κόστος και την πολυπλοκότητα της αγοράς server και της ρύθμισής τους. Τα οφέλη των PaaS εφαρμογών έγκεινται στην αύξηση του αριθμού των ανθρώπων που μπορούν να προγραμματίσουν, διατηρήσουν και αναπτύξουν web εφαρμογές. Με λίγα λόγια, το PaaS προσφέρεται για να εκδημοκρατίζει την ανάπτυξη των web εφαρμογών με τον ίδιο τρόπο που η Microsoft Access εκδημοκρατίζει την ανάπτυξη της εφαρμογής client/server.

Σήμερα, η δημιουργία web εφαρμογών απαιτεί έμπειρους προγραμματιστές με τρεις απαραίτητες εξειδικεύσεις:

- ❖ Backend ανάπτυξη διακομιστή (π.χ. Java/J2EE)
- ❖ Ανάπτυξη Frontend πελάτη (π.χ. Javascript/Dojo)
- ❖ Website administration

Το PaaS προσφέρει τη δυνατότητα για τους γενικούς προγραμματιστές να δημιουργήσουν web εφαρμογές χωρίς να χρειάζεται εξειδικευμένη τεχνογνωσία, η οποία επιτρέπει μια ολόκληρη γενιά της Microsoft Access, Lotus Notes, και PowerBuilder προγραμματιστές να δημιουργήσουν web εφαρμογές, χωρίς πάρα πολύ απότομη καμπύλη εκμάθησης. □ Η εναλλακτική λύση για PaaS είναι η ανάπτυξη web εφαρμογών χρησιμοποιώντας εργαλεία ανάπτυξης desktop, όπως τα Eclipse ή Microsoft Access, και στη συνέχεια να ανπτύξει χειροκίνητα αυτές τις εφαρμογές σε έναν cloud-hosting πάροχο, όπως είναι ο Amazon Web Services (AWS). □ Μία PaaS λύση θα πρέπει να περιλαμβάνει τα ακόλουθα στοιχεία:

- ✓ Μία PaaS development λύση θα πρέπει να είναι browser-based.
- ✓ Ένα end-to-end λύση PaaS θα πρέπει να παρέχει να παρέχουν μια υψηλής παραγωγικότητας ολοκληρωμένη ανάπτυξη.
- ✓ Μια λύση PaaS θα πρέπει να παρέχει ενσωμάτωση με εξωτερικές διαδικτυακές υπηρεσίες και βάσεις δεδομένων.
- ✓ Μια λύση PaaS πρέπει να παρέχει ολοκληρωμένη παρακολούθηση των δραστηριοτήτων της εφαρμογής και των χρηστών, για να βοηθήσει τους προγραμματιστές να καταλάβουν τις εφαρμογές τους και τις βελτιώσεις που χρειάζεται να γίνουν σε αυτές.
- ✓ Επεκτασιμότητα, αξιοπιστία και ασφάλεια θα πρέπει να βασίζονται σε ένα PaaS χωρίς να απαιτείται πρόσθετη ανάπτυξη, διαμόρφωση ή άλλα έξοδα. Multitenancy (η δυνατότητα μιας εφαρμογής να διαμερίζει

αυτόματα την κατάσταση και τα δεδομένα για την εξυπηρέτηση ενός αυθαίρετου αριθμού χρηστών) πρέπει να γίνεται δεκτή χωρίς ζρόσθητη εργασία οποιουδήποτε είδους.

- ✓ Μια λύση PaaS πρέπει να υποστηρίζει τόσο την επίσημη όσο και την on-demand συνεργασία καθ'όλη την διάρκεια ζωής του λογισμικού (την ανάπτυξη, τον έλεγχο, την τεκμηρίωση και τις λειτουργίες), όσο θα διατηρεί την ασφάλεια του πηγαίου κώδικα και των σχετικών δικαιωμάτων πνευματικής ιδιοκτησίας.
- ✓ Μια λύση PaaS πρέπει να στηρίζει pay-as-you-go μετρητές χρέωσης. Οι πλατφόρμες PaaS έχουν επίσης λειτουργικές διαφορές από τις παραδοσιακές πλατφόρμες ανάπτυξης, όπως οι εξής:
 - Multitenant development tools Τα παραδοσιακά εργαλεία ανάπτυξης που προορίζονται για ένα μόνο χρήστη. Ένα cloud-based studio πρέπει να υποστηρίζει πολλαπλούς χρήστες, ο καθένας με πολλαπλά ενεργά έργα.
 - Multitenant deployment architecture Η επεκτασιμότητα δεν είναι συχνά κάτι που ανησυχεί την αρχική προσπάθεια για την ανάπτυξη και αφήνεται στους διαχειριστές του συστήματος να το χειριστούνε όταν το έργο αναπτυχθεί. Στην PaaS, η επεκτασιμότητα της εφαρμογής και των δεδομένων πρέπει να είναι ενσωματωμένη (π.χ., η εξισορρόπηση φορτίου και το failover πρέπει να είναι βασικά στοιχεία της αναπτυσσόμενης πλατφόρμας).
 - Integrated management Οι παραδοσιακές λύσεις ανάπτυξης (συνήθως) δεν συνδέονται με την παρακολούθηση χρόνου, αλλά σε PaaS η δυνατότητα παρακολούθησης θα πρέπει να βασίζεται στην πλατφόρμα ανάπτυξης.

Integrated billing

Οι PaaS προσφορές απαιτούν μηχανισμούς για την τιμολόγηση με βάση τη χρήση που είναι μοναδική για το κόσμο του SaaS.

2.5 Το Infrastructure-As-a-Service μοντέλο

Στο παραδοσιακό hosted application μοντέλο, ο πάροχος παρέχει το σύνολο της υποδομής για τον πελάτη για την εκτέλεση των εφαρμογών του. Συχνά, αυτό συνεπάγεται τη στέγαση ειδικού υλικού που αγοράζεται ή μισθώνεται για τη συγκεκριμένη εφαρμογή. Το μοντέλο IaaS παρέχει επίσης την υποδομή για να τρέξει τις εφαρμογές, αλλά η προσέγγιση του Cloud Computing δίνει τη δυνατότητα να προσφέρει ένα pay-per-use μοντέλο και να αναβαθμίζει την υπηρεσία ανάλογα με τη ζήτηση. Από την πλευρά του παρόχου IaaS, μπορεί να δημιουργήσει μια υποδομή που χειρίζεται τα peaks και troughs των πελατών του και προσθέτει νέες ικανότητες, όπως τις συνολικές αυξήσεις της ζήτησης. Ομοίως, σε ένα hosted μοντέλο εφαρμογής, ο πάροχος IaaS μπορεί να καλύψει μόνο το hosting της εφαρμογής, ή μπορεί να επεκταθεί και σε άλλες υπηρεσίες (όπως υποστήριξη εφαρμογής, ανάπτυξη εφαρμογής και βελτιώσεις) και μπορεί να στηρίξει την πιο ολοκληρωμένη outsourcing του IT.

Το μοντέλο IaaS είναι παρόμοιο με το utility computing, στο οποίο η βασική ιδέα είναι να προσφέρει computing υπηρεσίες με τον ίδιο τρόπο όπως και utilities. Δηλαδή, θα πληρώσει για την ποσότητα της επεξεργαστικής ισχύς, το χώρο στο δίσκο και άλλα που πραγματικά καταναλώνουν. Το IaaS τυπικά είναι μια υπηρεσία που συνδέεται με το cloud computing και αναφέρεται σε online υπηρεσίες που δεν απασχολούν το χρήστη για τις λεπτομέρειες των υποδομών, συμπεριλαμβανομένων και των φυσικών πόρων πληροφορικής, την τοποθεσία, data partitioning, scaling, ασφάλεια, backup και ούτω καθεξής. Στο Cloud Computing, ο πάροχος έχει τον απόλυτο έλεγχο της υποδομής. Οι χρήστες του utility computing, αντιθέτως, αναζητούν μια υπηρεσία που τους εμποτρέπει να αναπτύξουν, να διαχειριστούν, και να κλιμακώσουν τις online υπηρεσίες χρησιμοποιώντας τους πόρους του παρόχου και να πληρώνουν συνδρομητικά για τους πόρους που καταναλώνουν. Ωστόσο, ο πελάτης θέλει να έχει τον έλεγχο της γεωγραφικής θέσης των υποδομών και του τι τρέχει στο κάθε server.

Τα διαθέσιμα χαρακτηριστικά για ένα τυπικό σύστημα IaaS περιλαμβάνουν: Scalability (επεκτασιμότητα)

Η δυνατότητα να αναβαθμίζονται οι απαιτήσεις υποδομής, όπως οι υπολογιστικοί πόροι, η μνήμη και η αποθήκευση (σε near-real-time ταχύτητες) με βάση τις απαιτήσεις του χρήστη.

Pay-as-you-go

Η δυνατότητα να αγοράσετε το ακριβές ποσό της υποδομής που απαιτείται σε κάθε συγκεκριμένη χρονική στιγμή.

Best-of-breed technology and resources

Πρόσβαση σε best-of-breed τεχνολογικές λύσεις και ανώτερο IT ταλέντο για ένα μέρος του κόστους.

📖 Βιβλιογραφία: Cloud security and privacy, Above the clouds(A Berkeley View Of Cloud Computing) Electrical engineering and Computer Sciences University of California at Berkeley.

3. CLOUD ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ

Ο όρος cloud είναι μια μεταφορά για το Διαδίκτυο και είναι μια απλοποιημένη αναπαράσταση του συμπλέγματος, internetworked συσκευές και των συνδέσεων που αποτελούν το Διαδίκτυο. Ιδιωτικά (private) και δημόσια (public) clouds είναι τα υποσύνολα του Διαδικτύου και ορίζονται με βάση τη σχέση τους με την επιχείρηση. Ιδιωτικά και δημόσια clouds μπορεί επίσης να αναφέρονται ως εσωτερικά ή εξωτερικά clouds. Η διαφοροποίηση γίνεται με βάση τη σχέση του cloud με την επιχείρηση.

Οι έννοιες για τα δημόσια και ιδιωτικά cloud είναι σημαντικές γιατί υποστηρίζουν το cloud computing, το οποίο επιτρέπει τη δημιουργία προβλέψεων, την επεκτασιμότητα, και τους virtualized πόρους μέσω των συνδέσεων δικτύου από ένα προμηθευτή ή μιας IT επιχείρησης στους

πελάτες έναντι αμοιβής. Οι τελικοί χρήστες που χρησιμοποιούν τις υπηρεσίες που προσφέρονται μέσω του cloud computing μπορεί αν μην γνωρίζουν για αυτό, να μην έχουν εμπειρία ή να μην μπορούν να ελέγξουν την τεχνολογική υποδομή που το υποστηρίζει.

Η πλειοψηφία των υποδομών cloud computing αποτελείται από αξιόπιστες υπηρεσίες που παρέχονται μέσω των data centers και έχουν ενσωματωθεί σε διακομιστές (servers) με διαφορετικά επίπεδα των τεχνολογιών virtualization. Οι υπηρεσίες είναι προσβάσιμες οπουδήποτε υπάρχει διαθέσιμη πρόσβαση στο Διαδίκτυο. Το cloud εμφανίζεται ως ένα εννιαίο σημείο πρόσβασης για όλες τις πληροφοριακές ανάγκες των καταναλωτών. Εμπορικές προσφορές θα πρέπει να πληρούν τις απαιτήσεις ποιότητας της εξυπηρέτησης των πελατών και συνήθως προσφέρουν υπηρεσίες επιπέδου συμφωνιών (service level agreements SLAs). Τα open standards είναι κρίσιμα για την ανάπτυξη του cloud computing, και το open source λογισμικό παρέχει τα θεμέλια για πολλές εφαρμογές του cloud computing (π.χ. η χρήση του Xen στο AWS).

3.1 Δημόσια clouds (public clouds)

Τα δημόσια clouds ή εξωτερικά clouds περιγράφουν το cloud computing με την παραδοσιακή mainstream έννοια, σύμφωνα με την οποία οι πόροι τροφοδοτούνται δυναμικά σε ένα fine-grained, self-service βάση πάνω από το internet, μέσα από web εφαρμογές ή υπηρεσίες web, από ένα off-site, τρίτο πάροχο που μοιράζεται τους πόρους και τους λογαριασμούς σε fine-grained, utility-computing βάση.

Ένα δημόσιο cloud φιλοξενείται, λειτουργεί και διοικείται από ένα τρίτο προμηθευτή από ένα ή περισσότερα data centers. Η υπηρεσία προσφέρεται σε πολλαπλούς πελάτες (το cloud προσφέρεται σε πολλούς ενοικιαστές) κατά τη διάρκεια μιας κοινής υποδομής. Βλέπε Εικόνα 5.



Εικόνα 5 : Δημόσια Clouds

Σε ένα δημόσιο cloud, η διαχείριση της ασφάλειας και day-to-day εργασίες έχουν υποβιβαστεί σε άλλους κατασκευαστές, που είναι υπεύθυνοι για την προσφορά υπηρεσιών του δημόσιου cloud. Ως εκ τούτου, ο πελάτης της υπηρεσίας που προσφέρει το δημόσιο cloud έχει χαμηλό επίπεδο του ελέγχου και της εποπτείας των φυσικών και λογικών πτυχών της ασφάλειας ενός ιδιωτικού cloud.

3.2 Ιδιωτικά clouds (private clouds)

Ιδιωτικά clouds και εσωτερικά clouds είναι όροι που χρησιμοποιούνται για να περιγράψουν τις προσφορές που προσομοιώνουν το cloud computing με τα ιδιωτικά δίκτυα. Αυτά τα (τυπικά αυτοματοποιημένα virtualization) προϊόντα που ισχυρίζονται ότι προσφέρουν κάποια οφέλη του cloud computing, χωρίς τους κρυφούς κινδύνους, αξιοποιώντας την ασφάλεια των δεδομένων, την εταιρική διακυβέρνηση, και αφορούν την αξιοπιστία. Οι οργανισμοί πρέπει να τα αγοράσουν, να τα κατασκευάσουν, και να τα διαχειριστούν και, ως εκ τούτου, να μην επωφελούνται από το χαμηλό κόστος κεφαλαίου και την less hands-on διαχείριση. Η οργάνωση των πελατών ενός ιδιωτικού cloud είναι υπεύθυνη για τη λειτουργία του ιδιωτικού αυτού cloud.

Τα ιδιωτικά cloud διαφέρουν από τα δημόσια cloud στο ότι το δίκτυο, η πληροφορική και η υποδομές αποθήκευσης που σχετίζονται με private clouds είναι αφιερωμένα σε έναν ενιαίο οργανισμό και δεν μοιράζονται με άλλους οργανισμούς (π.χ. το cloud είναι αφιερωμένο σε ένα μεμονωμένο ενοικιαστή). Ως εκ τούτου, έχει προκύψει μια ποικιλία από σχέδια με private clouds.

Dedicated

Private clouds που φιλοξενούνται σε customer-owned data centers ή σε collocation εγκαταστάσεις, και λειτουργούνε από τα εσωτερικά τμήματα του IT.

Community (κοινότητα)

Τα ιδιωτικά clouds που βρίσκονται σε εγκαταστάσεις ενός τρίτου, ανήκουν, διαχειρίζονται και λειτουργούνε από έναν πάροχο ο οποίος δεσμεύεται από τα προσαρμοσμένα SLAs και από συμβατικές ρήτρες με τις απαιτήσεις για ασφάλεια και συμμόρφωση.

Managed

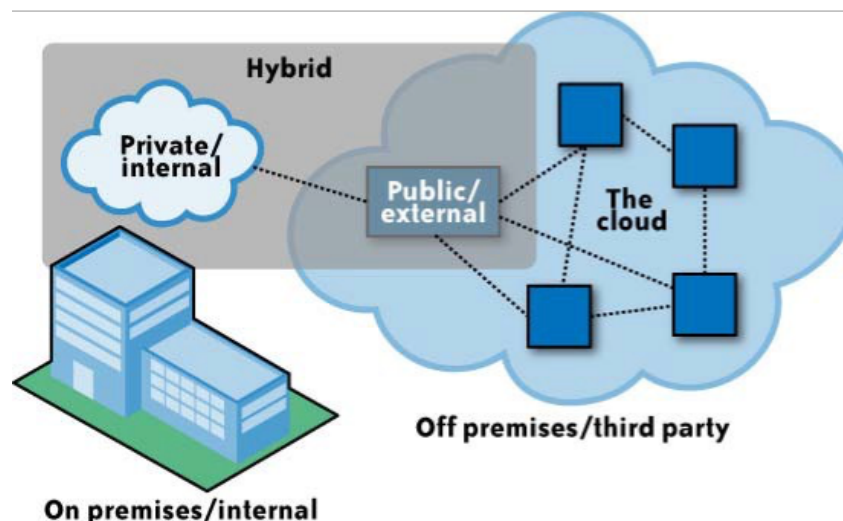
Οι υποδομές των ιδιωτικών cloud ανήκουν στο πελάτη και τις διαχειρίζεται ο

πάροχος.

Σε γενικές γραμμές, σε ένα ιδιωτικό cloud μοντέλο λειτουργίας, η διαχείριση της ασφάλειας και η λειτουργία των day-to-day hosts έχουν υποβιβαστεί στο εσωτερικό (internal) IT ή σε ένα τρίτο μέλος με συμβατικά SLAs. Σύμφωνα με το παρόν μοντέλο διακυβέρνησης, ένας πελάτης του ιδιωτικού cloud θα πρέπει να έχει υψηλό βαθμό του ελέγχου και της εποπτείας των φυσικών και λογικών πτυχών της ασφάλειας των υποδομών του ιδιωτικού cloud τόσο του hypervisor όσο και για τα εικονικά hosts OSs. Με αυτόν το υψηλό βαθμό ελέγχου και διαφάνειας, είναι πιο εύκολο για έναν πελάτη να συμμορφωθεί με τα καθιερωμένα εταιρικά πρότυπα ασφαλείας, τις πολιτικές και τους κανόνες συμμόρφωσης.

3.3 Υβριδικά clouds (Hybrid)

Ένα περιβάλλον υβριδικού cloud αποτελείται από πολλαπλούς εσωτερικούς και εξωτερικούς παρόχους και είναι μια πιθανή ανάπτυξη για τους οργανισμούς. Με ένα υβριδικό cloud, οι οργανισμοί μπορούν να τρέξουν μη βασικές εφαρμογές σε ένα δημόσιο cloud, διατηρώντας βασικές εφαρμογές και ευαίσθητα δεδομένα in-house σε ένα ιδιωτικό cloud. (βλ. Σχήμα 6).



Εικόνα 6 : Υβριδικό (Hybrid) cloud

Το Σχήμα 7 παραθέτει μερικά παραδείγματα των CSPs.

	Cloud providers	What they offer	Target cloud product segment	
	Amazon AWS	Cloud-based infrastructure hosting including storage, Virtual Private Clouds (VPC)	Infrastructure-as-a-service	<i>Service-centric</i>
	Salesforce AppExchange	Cloud-based application hosting	Platform-as-a-service	
Established organizations	IBM	Cloud infrastructure hosting and related value-added services	Cloud infrastructure	<i>Products and services</i>
	Microsoft	Cloud-based software platform	Application development platform	
	Sun	Cloud infrastructure hosting and related value-added services	Cloud infrastructure	
New entrants	Engine Yard	Platform to run Ruby on Rails applications	Platform-as-a-service	<i>Niche services</i>
	FlexiScale	Cloud hosting platform similar to Amazon's EC2 platform – aimed towards start-ups	Infrastructure-as-a-service	
	CohesiveFT	Offers a cloud-based VPN security solution	Cloud security management service	<i>Niche management services</i>
	RightScale	Cloud management platform; capable of managing cloud infrastructure from multiple providers	Cloud infrastructure management service	

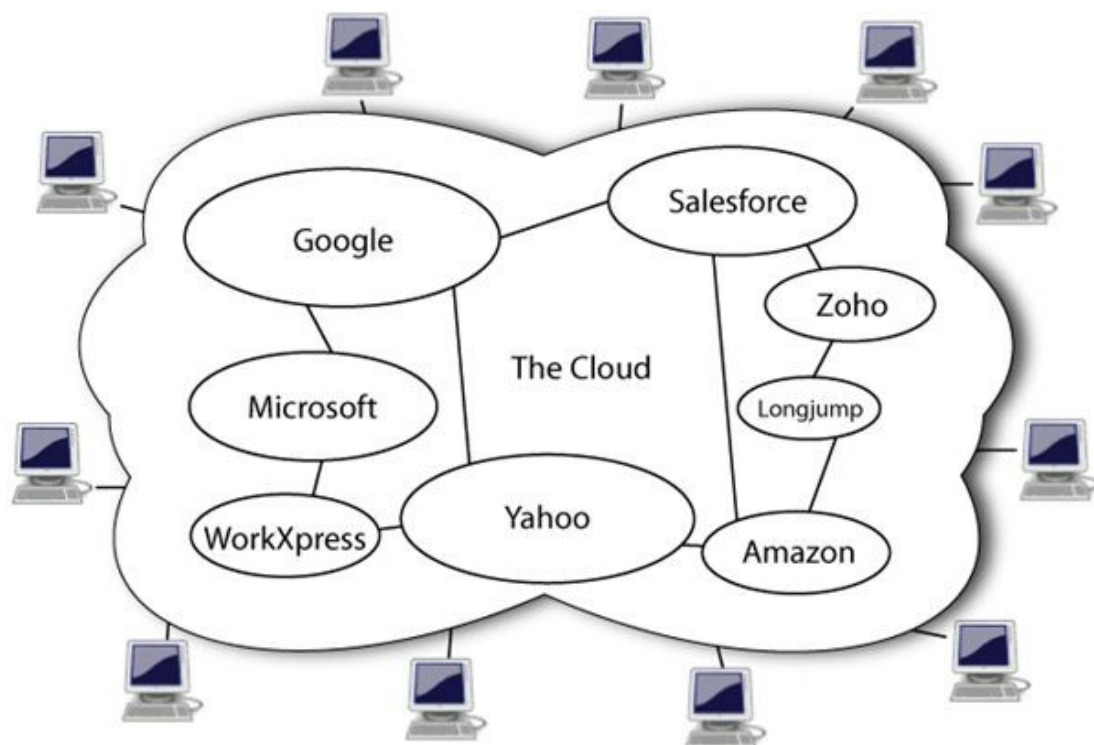
Εικόνα 7 : Παραδείγματα CSP και οι αντίστοιχες προσφορές τους

Οι υπηρεσίες που παρέχονται μέσω της ενσωμάτωσης των στοιχείων cloud εξελίσσονται, τα εμπόδια ξεπερνιούνται, και οι προϋποθέσεις αναπτύσσονται. Μια σημαντική ανησυχία είναι η εμπιστοσύνη που θα πρέπει να αναπτυχθεί στο ότι μια εταιρεία ή ενός μεμονωμένου ατόμου οι πληροφορίες είναι το ίδιο ασφαλείς και ιδιωτικές. Το να αναπτυχθεί αυτή η εμπιστοσύνη είναι ένα πολύ σημαντικό ορόσημο στην έκδοση του πλήρους φάσματος του cloud computing.

📖 Βιβλιογραφία: Cloud security and privacy, Above the clouds(A Berkeley

View Of Cloud Computing) Electrical engineering and Computer Sciences
University of California at Berkeley

4. Cloud Computing Platforms

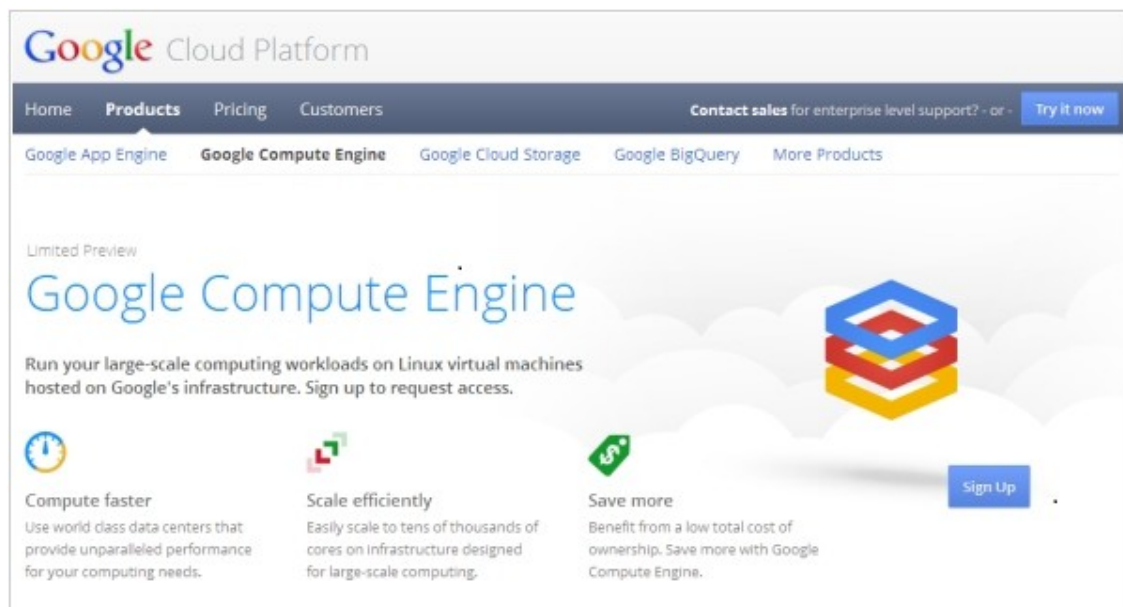


Εικόνα 8 : Πλατφόρμες cloud computing

Κάθε εφαρμογή χρειάζεται ένα υπολογιστικό μοντέλο για αποθήκευση και, αν υποθεθεί ότι η εφαρμογή είναι ακόμα trivially distributed, ένα μοντέλο

επικοινωνίας. Η στατιστική πολυπλεξία που είναι αναγκαία για να επιτευχθεί η ελαστικότητα και η ψευδαίσθηση της άπειρης χωρητικότητας απαιτεί πόρους για να είναι εικονική, έτσι ώστε η εφαρμογή του πως θα πρέπει να πολυπλέκονται και να κοινοποιούνται, να μπορούν να κρύβονται απο τον προγραμματιστή. Διάφορες utility computing προσφορές θα πρέπει να διακρίνονται με βάση το επίπεδο αφαίρεσης που παρουσιάζεται στον προγραμματιστή και το επίπεδο διαχείρισης των πόρων.

4.1 Google Compute Engine (GCE)

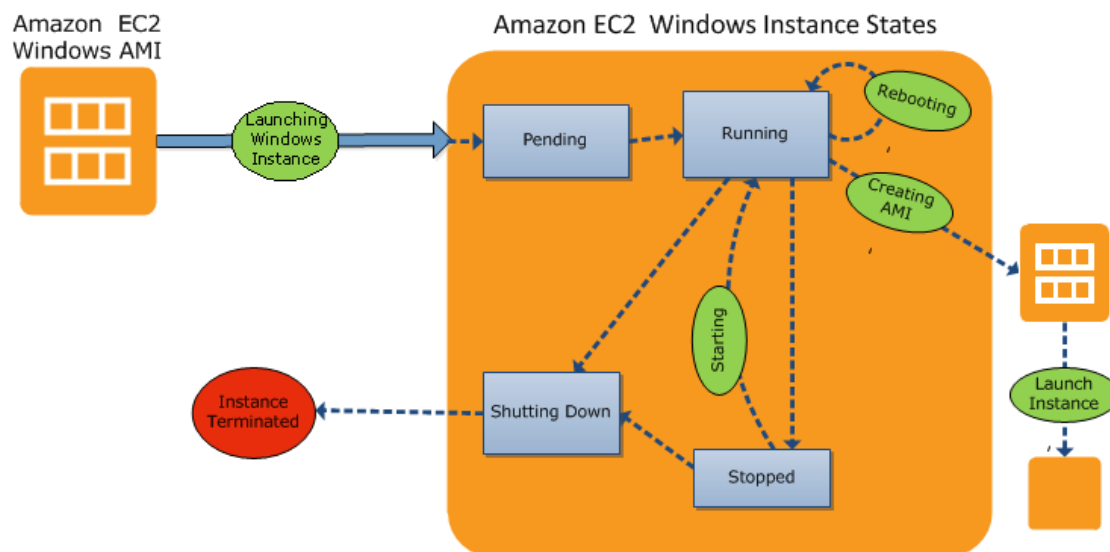


Εικόνα 9 : Google Compute Engine

Το Google Compute Engine (GCE) είναι ένα IaaS προϊόν που ανακοίνωσε η Google στο Google I/O τον Ιούνιο του 2012. Χρησιμοποιεί το KVM, όπως το hypervisor, και υποστηρίζει μόνο guest εικόνες που τρέχουν σε λειτουργικό σύστημα Linux. Το Google Compute Engine προσφέρει μια ξεκούραστη API για τη διαχείριση των πόρων, όπως τον δίσκο, τις εικόνες και τα instances. Το προεπιλεγμένο λειτουργικό σύστημα που παρέχεται από την Google είναι το Debian 6.0 και το 7.0, εκτός απο το Debian το μόνο λειτουργικό σύστημα που υποστηρίζεται είναι το CentOS 6.2.

Κάθε Google Compute Engine instance ξεκινά με ένα πόρο δίσκου. Ανάλογα με τον επιλεγμένο τύπο μηχανής, το instance μπορεί να ξεκινήσει με μηδενικό χώρο στο δίσκο(scratch), ή και τα δύο. Scratch χώρος στο δίσκο είναι ο χώρος που συνδέεται με τη ζωή ενός instance. Αν το instance τεραμιστεί για οποιονδήποτε λόγο, όλα τα δεδομένα στο scratch δίσκο χάνονται. Αντίθετα, οι persistent (επίμονοι) δίσκοι ζούνε πέρα απο τη ζωή ενός instance. Μπορούν να συνδέονται κατά την εκκίνηση, ή να επισυνάπτονται ή να αποκόβονται από την εκτέλεση ενός instance on demand. Οι Persistent δίσκοι μπορούν επίσης να συνδεθούν σε read-only λειτουργία για να πολλαπλασιάσει πολλά instance σε ένα. Οι Persistent δίσκοι προσφέρουν επιπλέον χαρακτηριστικά απο τους scratch δίσκους, όπως persistent instances στο δίσκο, booting από έναν persistent δίσκο, και μετάβαση των persistent δίσκων σε διαφορετικές ζώνες.

4.2 Amazon EC2



Εικόνα 10 : Amazon EC2

Το Amazon EC2 [4] είναι στο ένα άκρο του φάσματος. Ένα παράδειγμα EC2 μοιάζει πολύ με φυσικό υλικό, και οι χρήστες μπορούν να ελέγχουν σχεδόν ολόκληρη τη στοίβα λογισμικού, απο τον πυρήνα και προς τα πάνω. Η

εκτεθειμένη API είναι “thin”, μερικές δεκάδες APIκαλούνται να ζητήσουνε και να ρυθμίσουνε το εικονικό υλικό. Δεν υπάρχει όριο στα είδη των εφαρμογών που μπορούν να φιλοξενηθούν. Το χαμηλότερο επίπεδο virtualization-CPU, block-συσκευή αποθήκευσης, σύνδεση σε επίπεδο IP- επιτρέπει στους προγραμματιστές να γράψουνε ότι κώδικα θέλουνε. Από την άλλη πλευρά, αυτό είναι δύσκολο για το Amazon, να προσφέρει αυτόματη επεκτασιμότητα και failover, γιατί η σημασιολογία που συνδέεται με την αντιγραφή και άλλα διαχειριστικά του state management είναι άκρως εξαρτημένη από την εφαρμογή.

Το AWS προσφέρει μια σειρά από υψηλού επιπέδου υπηρεσίες διαχείρισης, συμπεριλαμβανομένων πολλών διαφορετικών συσκευών αποθήκευσης σε χρήση σε συνδυασμό με τον EC2, όπως είναι η SimpleDB. Ωστόσο, αυτές οι προσφορές έχουνε υψηλότερη λανθάνουσα κατάσταση και μη συνηθισμένα API, και αυτό που καταλαβαίνουμε είναι ότι δεν χρησιμοποιούνται τόσο ευρέως ως μέρη ενός AWS. Στο άλλο άκρο του φάσματος είναι οι domain-specific εφαρμογές όπως το Google, το AppEngine, το Force.com και το Salesforce που είναι μία πλατφόρμα επιχειρηματικής ανάπτυξης λογισμικού. Το AppEngine απευθύνεται αποκλειστικά σε παραδοσιακές εφαρμογές Web, επιβάλλοντας την υποδομή μιας εφαρμογής να διαχωρίζεται μεταξύ stateless βαθμίδα αποθήκευσης και stateful βαθμίδα αποθήκευσης. Επιπλέον, οι AppEngine εφαρμογές αναμένεται να είναι request-reply based, και ως τέτοια διαχωρίζεται αυστηρά ανάλογα με πόσο χρόνο χρειάζεται να λειτουργήσει η CPU για την εξυπηρέτηση ενός συγκεκριμένου αιτήματος. Οι εντυπωσιακοί αυτόματοι μηχανισμοί κλίμακας και υψηλής διαθεσιμότητας του AppEngine καθώς και το ιδιόκτητο Megastore data storage που είναι διαθέσιμο στις εφαρμογές του AppEngine, στηρίζονται σε αυτούς τους περιορισμούς. Έτσι, το AppEngine δεν είναι κατάλληλο για γενικού σκοπού computing. Ομοίως, το Force.com έχει σχεδιαστεί για να υποστηρίζει τις επιχειρηματικές εφαρμογές που τρέχουν ενάντια στην Salesforce.com βάση δεδομένων, και τίποτα άλλο.

Το Azure της Microsoft είναι ένα ενδιάμεσο σημείο σε αυτό το φάσμα

της ευελιξίας ενάντια στην άνεση του προγραμματιστή. Οι εφαρμογές Azure είναι γραμμένες με .NET βιβλιοθήκες, και μετατρέπονται σε Common Language Runtime, με ένα language independent διαχειριζόμενο περιβάλλον. Το σύστημα υποστηρίζει γενικής χρήσης computing και όχι μόνο μία κατηγορία της εφαρμογής. Οι χρήστες έχουν τη δυνατότητα επιλογής της γλώσσας, αλλά δεν μπορεί να ελέγξει το υποκείμενο λειτουργικό σύστημα ή την εκτέλεσή του. Οι βιβλιοθήκες παρέχουν ένα βαθμό αυτόματης ρύθμισης παραμέτρων του διαδικτύου και failover/επεκτασιμότητα, αλλά απαιτούνε τον προγραμματιστή να προσδιορίζει ορισμένες ιδιότητες της εφαρμογής, προκειμένου να το πράξουνε οι βιβλιοθήκες. Έτσι, το Azure είναι ενδιάμεσο μεταξύ ολοκληρωμένων frameworks των εφαρμογών όπως το AppEngine από την μία πλευρά, και hardware εικονικές μηχανές όπως είναι το EC2 και άλλες.

Amazon Elastic Compute Cloud (Amazon EC2)

Το Amazon Elastic Compute Cloud είναι μια διαδικτυακή υπηρεσία που παρέχει δυνατότητα αλλαγής μεγέθους της υπολογιστικής δυναμικότητας στο cloud. Έχει σχεδιαστεί για να κάνουν οι προγραμματιστές ευκολότερο web computing. Η απλή υπηρεσία για interface της Amazon EC2 μας δίνει τη δυνατότητα να αποκτήσουμε και να διαμορφώσουμε χωρητικότητα με την ελάχιστη τριβή. Μας παρέχει πλήρη έλεγχο των πόρων του υπολογιστή μας και μας επιτρέπει να εκτελέσουμε το αποδεδειγμένο computing περιβάλλον του Amazon. Το Amazon EC2 μειώνει το χρόνο που απαιτείται για την απόκτηση και την εκκίνηση νέων instances του server σε λεπτά, επιτρέποντάς μας να αναβαθμίσουμε γρήγορα τη χωρητικότητα, πάνω και κάτω, καθώς αλλάζουνε οι πληροφοριακές μας απαιτήσεις. Το Amazon EC2 αλλάζει και τα οικονομικά των servers του cloud computing, επιτρέποντας στο πελάτη να πληρώσει μόνο για τη χωρητικότητα που πραγματικά χρησιμοποιούνε. Παρέχει επίσης στους προγραμματιστές τα εργαλεία για την δημιουργία ανθεκτικών εφαρμογών και να απομονώσουν τον εαυτό τους από σενάρια αποτυχίας.

Η λειτουργικότητα του Amazon EC2

Το Amazon EC2 παρουσιάζει μια πραγματική cloud computing υπηρεσία, η οποία μας επιτρέπει να χρησιμοποιούμε διασυνδέσεις web υπηρεσιών για να ξεκινήσουμε instances με μια ποικιλία λειτουργικών συστημάτων.

Για να χρησιμοποιήσουμε το Amazon EC2 μπορούμε απλά:

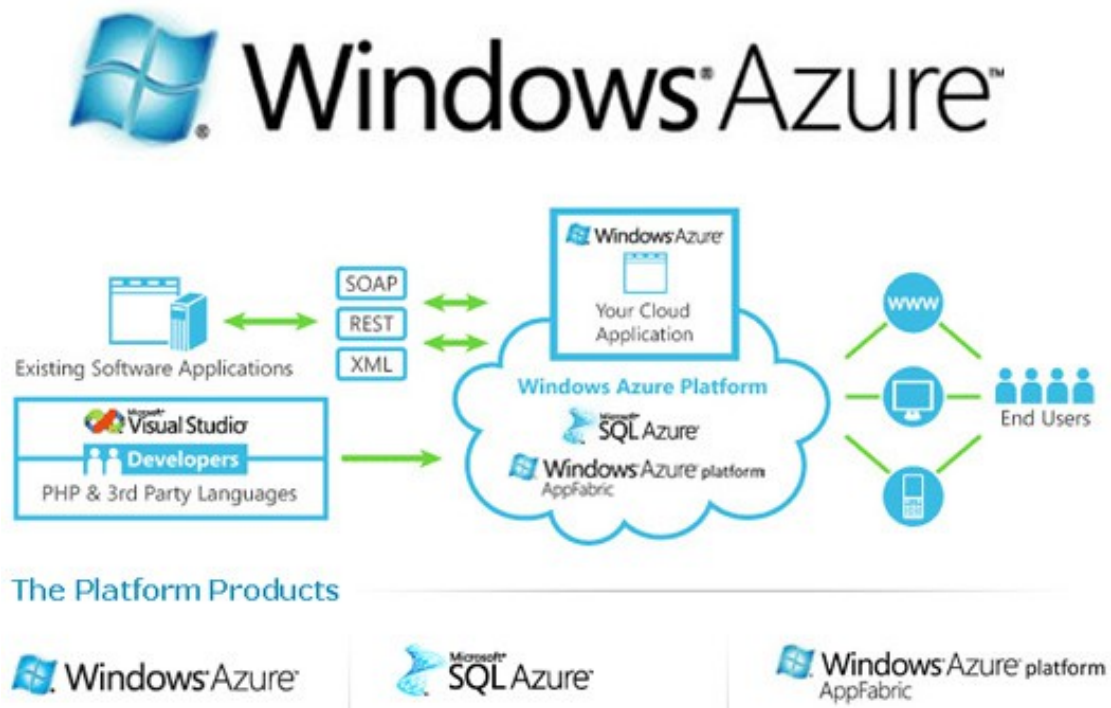
- ❖ Επιλέγουμε ένα pre-configured, template Image Machine Amazon (AMI) για να ανέβει και να λειτουργήσει αμέσως. Ή μπορούμε να δημιουργήσουμε ένα AMI που να περιέχει τις εφαρμογές μας, τις βιβλιοθήκες, τα δεδομένα και τις συναφείς ρυθμίσεις.
- ❖ Διαμορφώνουμε ασφαλή πρόσβαση στο δίκτυο για το Amazon EC2 instance.
- ❖ Επιλέγουμε ποια instances θέλουμε, στη συνέχεια ξεκινάμε, καταγράφουμε και τερματίζουμε όσα πιο πολλά instances από το AMI χρειάζονται, χρησιμοποιώντας τη διαδικτυακή υπηρεσία APIs ή την ποικιλία των εργαλείων διαχείριση που παρέχονται.
- ❖ Προσδιορίζουμε αν θέλουμε να τρέξουμε σε πολλαπλές τοποθεσίες, χρησιμοποιώντας στατικές IP παραμέτρους, ή επισυνάπτουμε persistent block αποθηκευτικό χώρο στα instances μας.
- ❖ Πληρώνουμε μόνο για τους πόρους που πραγματικά καταναλώνουμε, όπως τις ώρες για παράδειγμα ή τη μεταφορά δεδομένων.

4.3 Windows Azure

Το Windows Azure [5] είναι μια cloud computing πλατφόρμα, που δημιουργήθηκε από την Microsoft, για τη δημιουργία, την ανάπτυξη και τη διαχείριση εφαρμογών και υπηρεσιών μέσω ενός παγκόσμιου δικτύου των Microsoft-managed datacenters. Παρέχει τόσο την πλατφόρμα ως υπηρεσία (PaaS) και τις υποδομές ως υπηρεσία (IaaS) των υπηρεσιών του και υποστηρίζει πολλές διαφορετικές γλώσσες προγραμματισμού, εργαλεία και frameworks, συμπεριλαμβανομένων τόσο της Microsoft ειδικά για third-party λογισμικά ή συστήματα. Το Windows Azure κυκλοφόρησε στις 1 Φεβρουαρίου

του 2010.

Το Windows Azure είναι η πλατφόρμα εφαρμογής cloud της Microsoft. Μπορεί να χρησιμοποιηθεί για την κατασκευή μιας web εφαρμογής που τρέχει και αποθηκεύει τα



Εικόνα 11 : Πλατφόρμα Windows Azure

δεδομένα του σε datacenters της Microsoft. Μπορεί να συνδέσει on-premises εφαρμογές μεταξύ τους ή να συνδέσει μεταξύ τους διάφορα σύνολα των πληροφοριών ταυτότητας.

Τον Ιούνιο του 2012, το Windows Azure κυκλοφόρησε τα εξής νέα χαρακτηριστικά:

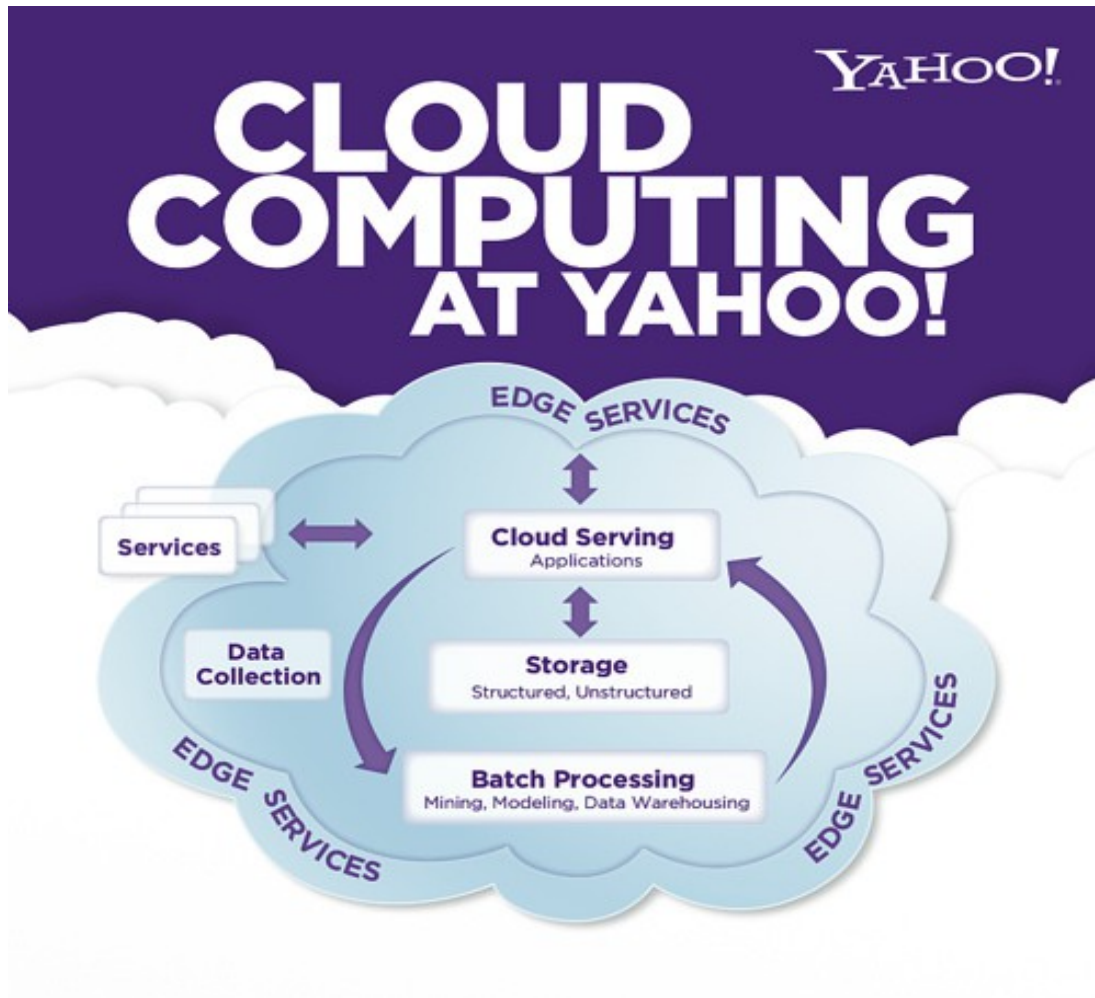
- Οι ιστοσελίδες επιτρέπουν στους προγραμματιστές να δημιουργήσουν δικτυακούς χρησιμοποιώντας ASP.NET, PHP, ή Node.js και μπορούν να αναπτυχθούν με τη χρήση FTP, Git ή Team Foundation Server.

- Εικονικές μηχανές επιτρέπουν στους προγραμματιστές να μετεγκαταστήσουν τις εφαρμογές και τις υποδομές τους, χωρίς αλλαγή του υφιστάμενου κώδικα, και μπορούν να τρέξουν τα Windows Server και Linux οι εικονικές μηχανές.
- Cloud Services, η πλατφόρμα της Microsoft ως υπηρεσία (PaaS) περιβάλλοντος που χρησιμοποιείται για να δημιουργήσει ολοκληρωμένες εφαρμογές και υπηρεσίες.
- Υποστηρίζει σενάρια multi-tier και αυτοματοποιημένες υλοποιήσεις.
- Η διαχείριση των δεδομένων, με βάση δεδομένων SQL, παλαιότερα γνωστή ως SQL Azure Database, εργάζεται για να δημιουργήσει την κλίμακα και να επεκτείνει τις εφαρμογές στο σύννεφο χρησιμοποιώντας την τεχνολογία Microsoft SQL Server. Ενσωματώνεται με το Active Directory και το Microsoft System Center και Hadoop.
- Υπηρεσίες οπτικοακουστικών μέσων επικοινωνίας, μια προσφορά PaaS που μπορεί να χρησιμοποιηθεί για την κωδικοποίηση, την προστασία του περιεχομένου, streaming και analytics. □ Η Azure Platform των Windows παρέχει ένα API χτισμένο σε REST, HTTP και XML που επιτρέπει στους προγραμματιστές να αλληλεπιδρούν με τις υπηρεσίες που παρέχονται από το Windows Azure. Η Microsoft παρέχει επίσης ένα client-side διαχειρίζεται βιβλιοθήκη κατηγορίας που ενσωματώνει τις λειτουργίες αλληλεπίδρασης με τις υπηρεσίες. Ενσωματώνει επίσης με το Microsoft Visual Studio, Git και Eclipse. Τον Οκτώβριο του 2012, η Microsoft κυκλοφόρησε μια download PDF αφίσα που συνοψίζει τα χαρακτηριστικά του Windows Azure σε αυτό. Το PDF περιέχει clickable περιοχές που δείχνοντας την πρόσθετη τεκμηρίωση. □ Το Windows Azure χρησιμοποιεί ένα εξειδικευμένο λειτουργικό σύστημα, που ονομάζεται Windows Azure για να τρέξει το “fabric layer” του – ένα σύμπλεγμα που φιλοξενείται σε datacenters της Microsoft που διαχειρίζεται computing πόρους και αποθήκευσης των ηλεκτρονικών υπολογιστών και των διατάξεων των πόρων (ή ένα

υποσύνολό τους) σε εφαρμογές που εκτελούνται πάνω από το Windows Azure. Έχει περιγραφεί ως ένα cloud layer στην κορυφή ενός αριθμού συστημάτων του Windows Server, που χρησιμοποιούν Windows Server 2008 και μια προσαρμοσμένη έκδοση του Hyper-V, που είναι γνωστή ως Windows Azure Hypervisor για την παροχή των υπηρεσιών virtualization. Η κλιμάκωση και η αξιοπιστία ελέγχονται από το Windows Azure Fabric ελεγκτή, ώστε οι υπηρεσίες και το περιβάλλον να μην κολλάνε, εάν ένας από τους εξυπηρετητές κολλάει μέσα στο datacenter της Microsoft και παρέχει τη διαχείριση της εφαρμογής περιήγησης του χρήστη, όπως πόρους μνήμης και εξισορρόπησης φορτίου.

4.4 Yahoo open-source cloud-serving engine

Το Network World – Yahoo αναπτύσσει μια εσωτερική μηχανή cloud-serving για να ενισχύσει την παραγωγικότητά της, και σκοπεύει να απελευθερώσει τον κώδικά της ως ένα open source φέτος.



Εικόνα 12 : Yahoo cloud computing

Παρόλα αυτά, το εσωτερικό project της Yahoo θα μπορούσε να κάνει την ανάπτυξη των δικών του Web ιδιοτήτων για να γίνει πιο αποτελεσματικό και να δώσει στις επιχειρήσεις και στους προγραμματιστές μια πλατφόρμα ελεύθερου λογισμικού για να δημιουργήσουν τον δικό τους cloud.

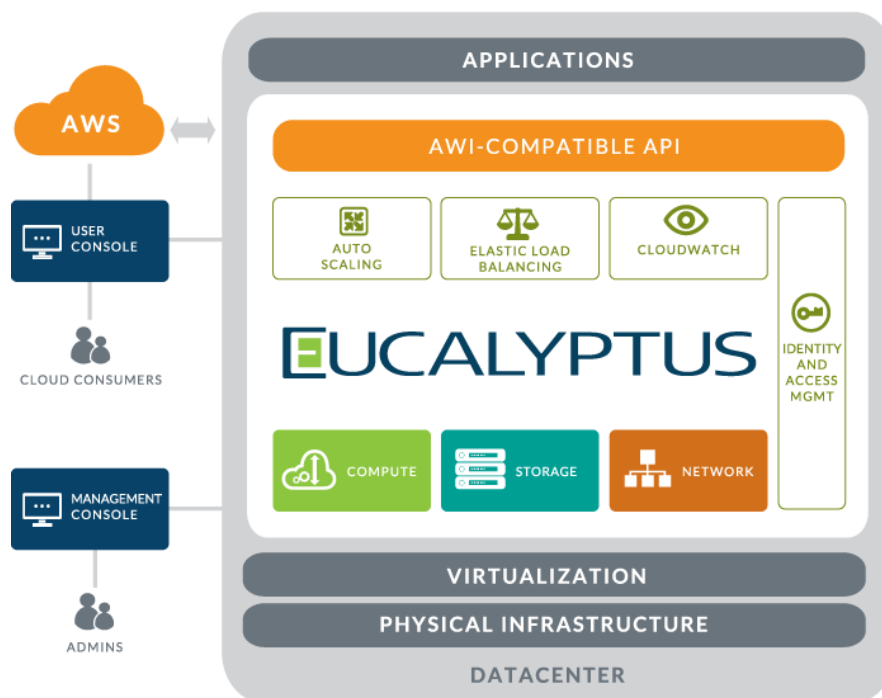
Η μηχανή του cloud-serving επιτρέπει στους προγραμματιστές να οικοδομήσουν υπηρεσίες σε containers που βρίσκονται πάνω στο virtual-machine layer, αφήνοντας προγραμματιστές να αναπτύσσουν γρήγορα εφαρμογές που έχουν ανεβεί και να τρέχουν με μια σειρά από κοινές υπηρεσίες. Είναι κάπου μεταξύ as-a-service υποδομής και της πλατφόρμας platform- as-a-service offering, για παράδειγμα, ένα χαμηλότερο επίπεδο

αφαίρεσης από αυτό που προβλέπεται από το Google App Engine, ένα γνήσιο PaaS παιχνίδι.

Ο μηχανισμός cloud-serving είναι γραμμένος σε Java και C++ και υποστηρίζει PHP και Javascript. Μόλις οι χρήστες ανοιχτού πηγαίου κώδικα θα μπορούν να υιοθετήσουν και άλλες γλώσσες, όπως οι Python και η Net Framework της Microsoft παρόλο που θα είναι λίγο δύσκολο, ή Java stack. Πριν από την απελευθέρωσή του σαν open-source το Yahoo θα αφαιρέσει μερικά στοιχεία που είναι ειδικά για το Yahoo και ως εκ τούτου δεν θα βοηθήσει τους εξωτερικούς χρήστες.

4.5 Eucalyptus software

Το Eucalyptus είναι ένα ελεύθερο λογισμικό και open-source υπολογιστών για την κατασκευή Amazon Web Services (AWS) συμβατό με ιδιωτικά και υβριδικά περιβάλλοντα



Εικόνα 13 : Αρχιτεκτονική Eucalyptus

cloud computing επιδοτούμενα από την εταιρεία Eucalyptus Systems. Το Eucalyptus επιτρέπει τη συγκέντρωση υπολογισμού, την αποθήκευση και τους πόρους του δικτύου που μπορούν να αναβαθμιστούν δυναμικά προς τα πάνω ή προς τα κάτω, όπως ο φόρτος εργασίας αίτησης αλλαγής. Το Eucalyptus Systems ανακοίνωσε μια επίσημη συμφωνία με την AWS τον Μάρτιο του 2012 για να διατηρηθεί η συμβατότητα.

Οι εντολές του Eucalyptus μπορούν να διαχειριστούν είτε Amazon ή Eucalyptus instances. Οι χρήστες μπορούν επίσης να μετακινήσουν instances μεταξύ του Eucalyptus private cloud και του Amazon Elastic Compute Cloud για να δημιουργήσουν ένα υβριδικό cloud. Το Virtualization hardware απομονώνει τις εφαρμογές από τις λεπτομέρειες του hardware του υπολογιστή.

📖 Βιβλιογραφία: Cloud Security Priva, AWS | Amazon Elastic Compute Cloud (EC2) – Scalable Cloud Servers, Windows Azure - Wikipedia, the free encyclopedia, Yahoo to open-source cloud-serving engine

5. INTERNET OF THINGS (IOT)

5.1 Ορισμός του IoT

Το Internet of Things (IoT) [6] είναι ένα δίκτυο φυσικών αντικειμένων, συσκευών, οχημάτων, κτιρίων αλλά και άλλων αντικειμένων τα οποία περιέχουν ενσωματωμένα ηλεκτρονικά συστήματα, λογισμικά, αισθητήρες και διαδικτυακή δυνατότητα σύνδεσης – κάτι που επιτρέπει σε αυτά τα αντικείμενα να συλλέγουν και να ανταλλάσσουν δεδομένα. Το IoT δίνει την δυνατότητα στα αντικείμενα αυτά να ελέγχονται απομακρυσμένα μέσω της

υπάρχουσας δικτυακής υποδομής δημιουργώντας ευκαιρίες άμεσης ενσωμάτωσης του φυσικού κόσμου με τα υπολογιστικά συστήματα έχοντας ως αποτέλεσμα τη βελτίωση της αποτελεσματικότητας και της ακρίβειας αλλά και τη μείωση του κόστους. Από την στιγμή μάλιστα που το IoT εξοπλίζεται με αισθητήρες και ενεργοποιητές αποτελεί μέρος έξυπνων συστημάτων της καθημερινότητας όπως είναι τα έξυπνα σπίτια, οχήματα και πόλεις. Κάθε αντικείμενο αναγνωρίζεται μοναδικά από το ενσωματωμένο υπολογιστικό σύστημα και μπορεί να λειτουργεί τόσο αυτόνομα όσο και σε συνεργασία με την υπόλοιπη διαδικτυακή υποδομή.

Ο όρος Internet of Things [8] προτάθηκε από τον Kevin Ashton το 1999 αν και ήταν υπό συζήτηση τουλάχιστον από το 1991. Η ιδέα του Internet of Things αρχικά έγινε δημοφιλής μέσω του Auto-ID Center στο MIT αλλά και μέσω σχετικών δημοσιεύσεων ανάλυσης της αγοράς. Εκείνες τις ημέρες η ταυτοποίηση ραδιοσυχνοτήτων (RFID) θεωρήθηκε ως προϋπόθεση για το Internet of Things, αφού αν όλα τα αντικείμενα και οι άνθρωποι ήταν εξοπλισμένοι στην καθημερινότητα με αναγνωριστικά, θα μπορούσαν να διαχειρίζονται και να απογράφονται από υπολογιστές. Εκτός από την χρήση RFID, η σήμανση των πραγμάτων μπορεί να επιτευχθεί μέσα από τεχνολογίες όπως η κοντινού τύπου επικοινωνία, οι κώδικες QR, τα barcodes και η ψηφιακή υδατογράφηση.

Πριν αρχίσουμε να κατανοούμε τη σημασία του IoT είναι πρώτα απαραίτητο να καταλάβουμε τις διαφορές μεταξύ του Διαδικτύου (Internet) και του Παγκόσμιου Ιστού (World Wide Web) όροι που συχνά χρησιμοποιούνται εναλλακτικά.

Το Διαδίκτυο είναι το φυσικό επίπεδο που αποτελείται από μεταγωγείς, δρομολογητές και άλλες συσκευές. Η κύρια λειτουργία του είναι να μεταφέρει πληροφορίες από το ένα σημείο στο άλλο γρήγορα, αξιόπιστα και με ασφάλεια. Ο Παγκόσμιος Ιστός από την άλλη πλευρά, είναι ένα επίπεδο εφαρμογών που λειτουργεί πάνω από το Διαδίκτυο. Ο κύριος ρόλος του είναι

να παρέχει μία διεπαφή που καθιστά τις πληροφορίες που ρέουν σε όλο το Διαδίκτυο χρησιμοποιήσιμες.

5.2 Το IoT στη ζωή μας

Σήμερα ο όρος Internet of Things (το Διαδίκτυο των Πραγμάτων) χρησιμοποιείται για να υποδηλώσει την προηγμένη συνδεσιμότητα συσκευών, συστημάτων και υπηρεσιών πέρα από τη μηχανή με μηχανή επικοινωνία αφού καλύπτει μία ποικιλία από πρωτόκολλα, τομείς και εφαρμογές. Σύμφωνα με το Gartner, θα υπάρχουν σχεδόν 26 δισεκατομμύρια συσκευές στο Internet of Things μέχρι το 2020. Σύμφωνα με το ABI Research, πάνω από 30 δισεκατομμύρια συσκευές θα συνδέονται ασύρματα από το Διαδίκτυο των πραγμάτων (Διαδίκτυο των πάντων) μέχρι το 2020.

Η Cisco δημιούργησε έναν δυναμικό "μετρητή συνδέσεων" προκειμένου να παρακολουθεί τον εκτιμώμενο αριθμό συνδεδεμένων αντικειμένων από τον Ιούλιο του 2013 μέχρι τον Ιούλιο του 2020. Η ιδέα αυτή, όπου οι συσκευές συνδέονται στο Διαδίκτυο/Παγκόσμιο Ιστό μέσω χαμηλής ισχύος ραδιοσήματα, είναι η πιο ενεργή περιοχή έρευνας στο IoT. Τα ραδιοσήματα χαμηλής ισχύος δεν χρειάζεται να χρησιμοποιούν Wi-Fi ή Bluetooth. Χαμηλότερης ενέργειας και χαμηλότερου κόστους εναλλακτικές διερευνώνται υπό την κατηγορία του Chirp Networks. Βρίσκεται υπό σκέψη μια εκδοχή που θέλει όλες τις συσκευές να χρησιμοποιούν μία διεύθυνση IP ως ένα μοναδικό αναγνωριστικό. Για κόμβους, γέφυρες και άλλες συσκευές τύπου δρομολόγησης αυτό φαίνεται να είναι περιττό.

Σύμφωνα με μία έρευνα και μελέτη που διεξήχθη από το Pew Research Internet Project, μία μεγάλη πλειοψηφία των ειδικών της τεχνολογίας και των χρηστών του Internet που εμπλέκονταν οι οποίοι ανταποκρίθηκαν (περίπου το 83%) συμφώνησαν με την ιδέα ότι το Διαδίκτυο των πραγμάτων και των ενσωματωμένων υπολογιστών θα έχει διαδεδομένα και ευεργετικά αποτελέσματα έως το 2025. Φαίνεται ότι η επόμενη εξέλιξη του

Διαδικτύου αλλάζει τα πάντα. Το Internet of Things (IoT), κάποιες φορές αναφέρεται ως το Διαδίκτυο των αντικειμένων και υπόσχεται να αλλάξει τα πάντα συμπεριλαμβανομένων και εμάς τους ίδιους. Αυτό μπορεί να μοιάζει σαν μία τολμηρή δήλωση αλλά λαμβάνοντας υπόψη τον αντίκτυπο που έχει στην εκπαίδευση, την επικοινωνία, τις επιχειρήσεις, την επιστήμη αλλά και την ανθρωπότητα, αναμφισβήτητα το Διαδίκτυο είναι ένα από τα πιο σημαντικά και ισχυρά δημιουργήματα σε όλη την ανθρώπινη ιστορία. Έτσι, θεωρώντας ότι το IoT αντιπροσωπεύει την επόμενη εξέλιξη του Διαδικτύου, κάνει ένα τεράστιο άλμα στην ικανότητά του να συγκεντρώνει, να αναλύει και να διανέμει δεδομένα που μπορούμε να μετατρέψουμε σε πληροφορίες, γνώσεις και τελικά σοφία. Στο πλαίσιο αυτό το IoT γίνεται ιδιαίτερα σημαντικό και χρήσιμο.

Ήδη, οι εργασίες του IoT βρίσκονται σε εξέλιξη και υπόσχονται να κλείσουν το χάσμα μεταξύ φτώχειας και πλούτου, βελτιώνοντας την κατανομή των πόρων του κόσμου σε αυτούς που τα χρειάζονται περισσότερο, βοηθώντας μας έτσι να κατανοήσουμε τον πλανήτη μας έτσι ώστε να μπορούμε να είμαστε πιο παραγωγικοί και λιγότερο αντιδραστικοί. Παρόλα αυτά υπάρχουν πολλά εμπόδια που απειλούν να επιβραδύνουν την ανάπτυξη του IoT, συμπεριλαμβανομένης της μετάβασης στο IPv6, έχοντας ένα κοινό σύνολο προτύπων και ανάπτυξης των πηγών ενέργειας για εκατομμύρια, ακόμη και δισεκατομμύρια μικροσκοπικούς αισθητήρες. Ωστόσο καθώς οι επιχειρήσεις, οι κυβερνήσεις, οι οργανισμοί τυποποίησης και η ακαδημαϊκή κοινότητα λύνουν τις προκλήσεις αυτές, το IoT θα συνεχίσει να προοδεύει.

5.3 Κυρια χαρακτηριστικά του IoT

[10] Από τεχνικής άποψης, το Internet of Things δεν είναι το αποτέλεσμα μιας μόνο πρωτότυπης τεχνολογίας. Σε αντίθεση, διάφορες συμπληρωματικές τεχνικές εξέλιξης παρέχουν δυνατότητες που συνεργάζονται για να βοηθήσουν να γεφυρωθεί το χάσμα μεταξύ του εικονικού και του φυσικού κόσμου. Οι δυνατότητες αυτές περιλαμβάνουν:

- ❖ **Επικοινωνία και συνεργασία:** Τα αντικείμενα έχουν την δυνατότητα να δικτυώνονται με τους πόρους του Διαδικτύου ή ακόμη και το ένα με το άλλο, να κάνουν χρήση των δεδομένων και των υπηρεσιών και να ενημερώνουν την κατάστασή τους. Οι ασύρματες τεχνολογίες, όπως το GSM και UMTS, Wi-Fi, Bluetooth, ZigBee και διάφορα άλλα ασύρματα πρότυπα δικτύωσης που είναι αυτή τη στιγμή υπό ανάπτυξη, ιδιαίτερα εκείνων που σχετίζονται με τα προσωπικά ασύρματα δίκτυα (WPANs), έχουν πρωταρχική σημασία εδώ.
- ❖ **Διευθυνσιοδότηση:** Σε ένα Internet of Things, τα αντικείμενα μπορούν να τοποθετούνται και να διευθυνσιοδοτούνται μέσω της ανεύρεσης ή το όνομα των υπηρεσιών κι έτσι έχουν την δυνατότητα να επιβεβαιώνονται ή να ρυθμίζονται εξ αποστάσεως.
- ❖ **Ταυτοποίηση:** Τα αντικείμενα είναι μοναδικά αναγνωρίσιμα. Οι τεχνολογίες RFID, NFC (Near Field Communication) και οι οπτικά αναγνώσιμοι κώδικες (bar codes) είναι παραδείγματα τεχνολογιών με τις οποίες μπορούν να εντοπιστούν ακόμη και παθητικά αντικείμενα που δεν έχουν ενσωματωμένους ενεργειακούς πόρους και να αναγνωριστούν με τη βοήθεια ενός 'διαμεσολαβητή' όπως μία συσκευή αναγνώρισης RFID ή ένα κινητό τηλέφωνο. Η ταυτοποίηση επιτρέπει στα αντικείμενα να συνδέονται με πληροφορίες που σχετίζονται με το συγκεκριμένο αντικείμενο και μπορούν να ανακτηθούν από έναν διακομιστή, υπό τον όρο ο μεσολαβητής να είναι συνδεδεμένος στο δίκτυο.
- ❖ **Ανίχνευση:** Τα αντικείμενα διαθέτουν αισθητήριες με την βοήθεια των οποίων συλλέγουν πληροφορίες σχετικά με το περιβάλλον τους, καταγράφοντας, διαβιβάζοντας ή αντιδρώντας άμεσα σε αυτό.
- ❖ **Ενεργοποίηση:** Τα αντικείμενα περιέχουν ενεργοποιητές προκειμένου να χειριστούν το περιβάλλον τους. Για παράδειγμα μετατρέποντας τα ηλεκτρικά σήματα σε μηχανική κίνηση. Τέτοιοι ενεργοποιητές μπορούν

να χρησιμοποιηθούν για να ελέγξουν εξ αποστάσεως διεργασίες στον πραγματικό κόσμο μέσω του Διαδικτύου.

- ❖ Ενσωματωμένη επεξεργασία πληροφοριών: Έξυπνα αντικείμενα διαθέτουν έναν επεξεργαστή ή μικροελεγκτή, καθώς και την ικανότητα αποθήκευσης. Αυτοί οι πόροι μπορούν να χρησιμοποιηθούν για παράδειγμα, για να επεξεργάζονται και να ερμηνεύουν πληροφορίες των αισθητήρων ή ακόμη και να παρέχουν στα προϊόντα μία μνήμη για το πως έχουν χρησιμοποιηθεί.
- ❖ Εντοπισμός: Τα έξυπνα αντικείμενα έχουν επίγνωση της φυσικής τους θέσης ή μπορούν να εντοπίζονται. Το GPS ή το δίκτυο κινητής τηλεφωνίας είναι κατάλληλες τεχνολογίες για την επίτευξη αυτού του στόχου, καθώς και οι ραδιοφάροι (π.χ. γειτονικοί WLAN σταθμοί βάσεις ή αναγνώστες RFID με γνωστές συντεταγμένες) όπως επίσης και οι οπτικές ίνες.
- ❖ Διεπαφές χρήστη: Τα έξυπνα αντικείμενα μπορούν να επικοινωνούν με τους ανθρώπους με κατάλληλο τρόπο (είτε άμεσα είτε έμμεσα, για παράδειγμα μέσω ενός smart phone). Καινοτόμα παραδείγματα αλληλεπίδρασης είναι: χειροπιαστές διεπαφές χρήστη, ευέλικτες πολυμερείς βάσεις εικόνας και φωνής ή μέθοδοι αναγνώρισης χειρονομιών.

Οι περισσότερες ειδικές εφαρμογές χρειάζονται μόνο ένα υποσύνολο αυτών των δυνατοτήτων, ιδιαίτερα μετά την εφαρμογή όλων αυτών είναι συχνά δαπανηρό και απαιτεί σημαντική τεχνική προσπάθεια. Οι εφαρμογές Logistics για παράδειγμα, αυτή τη στιγμή επικεντρώνονται στην προσέγγιση εντοπισμού (δηλ. τη θέση του τελευταίου σημείου ανάγνωσης) και στην σχετικά χαμηλού κόστους αναγνώριση των αντικειμένων που χρησιμοποιούν RFID ή bar codes. Αισθητήρες δεδομένων ή ενσωματωμένοι επεξεργαστές είναι περιορισμένοι σε τέτοιες εφαρμογές εφοδιασμού όπου οι πληροφορίες

είναι απαραίτητες και ουσιαστικής σημασίας, όπως η θερμοκρασία που πρέπει να ελέγχεται κατά τη μεταφορά εμβολίων. Οι προάγγελοι της επικοινωνίας σε αντικείμενα καθημερινής χρήσης έχουν κάνει ήδη την εμφάνισή τους, ιδιαίτερα σε σχέση με το RFID, για παράδειγμα, η επικοινωνία μικρής εμβέλειας όπως η χρήση έξυπνων κλειδιών-καρτών από τις πόρτες των δωματίων ενός ξενοδοχείου ή αντίστοιχα εισιτήρια για σκι που επικοινωνούν με τα lift. Περισσότερο φουτουριστικά σενάρια περιλαμβάνουν ένα έξυπνο τραπέζι παιχνιδιόχαρτων, όπου η πορεία του παιχνιδιού βιντεοσκοπείται χρησιμοποιώντας τραπουλόχαρτα εξοπλισμένα με RFID. Παρόλα αυτά όλες αυτές οι εφαρμογές εξακολουθούν να περιλαμβάνουν ειδικά συστήματα σε τοπική ανάπτυξη, επομένως δεν μιλάμε για ένα 'Διαδίκτυο' με την έννοια ενός ανοικτού, επεκτάσιμου και τυποποιημένου συστήματος.

5.4 Εφαρμογές

Παρακάτω θα ρίξουμε μια ματιά σε μερικές κατηγορίες σχετικές με IoT εφαρμογές. Ενώ υπάρχουν κυριολεκτικά εκατοντάδες εφαρμογές που αναγνωρίζονται από διάφορες βιομηχανίες, είναι εφικτό να ταξινομηθούν με έναν απλό, λογικό τρόπο. [11]

5.4.1 Πρώτη κατηγορία εφαρμογών

Η πρώτη κατηγορία περικλείει την ιδέα εκατομμυρίων ετερογενών έξυπνων και διασυνδεδεμένων συσκευών με μοναδικά αναγνωριστικά ID τα οποία αλληλεπιδρούν με άλλα μηχανήματα/αντικείμενα, υποδομές και το φυσικό περιβάλλον. Στην κατηγορία αυτή, το IoT παίζει σε μεγάλο βαθμό τον ρόλο της διοίκησης, του ελέγχου και της διαδρομής. Όπως συμβαίνει με όλες τις πτυχές του IoT, η ασφάλεια και η προστασία είναι υψίστης σημασίας.

Οι εφαρμογές αυτές δεν αποσκοπούν στην εξόρυξη των δεδομένων από τις συμπεριφορές των ανθρώπων αλλά κυρίως στην επέκταση του

αυτοματισμού και της μηχανής προς μηχανή, μηχανής προς υποδομή, μηχανής προς φύση και γενικότερα επικοινωνιών που μπορούν να συνεισφέρουν και να βοηθήσουν στην απλοποίηση της ζωής των ανθρώπων.

5.4.2 Δεύτερη κατηγορία εφαρμογών

Η δεύτερη κατηγορία σχετίζεται με την επεξεργασία των δεδομένων που συλλέγονται από τους τελικούς κόμβους (έξυπνες συσκευές με αισθητήρες και δυνατότητα σύνδεσης) και την εύρεση των δεδομένων για τις τάσεις και τις συμπεριφορές που μπορούν να παράγουν χρήσιμες πληροφορίες σχετικές με το marketing προκειμένου να δημιουργηθεί επιπλέον εμπόριο.

Οι εταιρείες πιστωτικών καρτών αλλά και οι κάρτες των μελών των καταστημάτων ήδη παρακολουθούν και χρησιμοποιούν την συμπεριφορά των ανθρώπων προκειμένου να καταλήξουν σε προσφορές που μπορούν να προωθήσουν αυξημένες πωλήσεις. Τώρα, το ερώτημα είναι πόσο μακριά θα καταφέρει να πάει αυτή η εκμετάλλευση των δεδομένων. Περιπτώσεις χρήσης θα μπορούσαν να περιλαμβάνουν την παρακολούθηση των καταστημάτων που έχουμε επισκεφθεί, τους διαδρόμους που περιηγηθήκαμε και τον χρόνο που ξοδέψαμε κατά την διάρκεια των αγορών μας, ακόμα και το είδος των αντικειμένων που κρατήσαμε και πλοηγηθήκαμε. Τέτοια σενάρια είναι εύκολα εφικτά με τη χρήση ενός κινητού τηλεφώνου με GPS ,RFID και έξυπνες/ασύρματες ετικέτες στα καταστήματα. Το αποτέλεσμα θα μπορούσε να είναι τόσο απλό όσο η παροχή e-mail προσφορών ή οι υπηρεσίες push στα σημεία πώλησης.

Έτσι, μπορούμε να δούμε πως σε αυτή την κατηγορία το IoT μπορεί να επιτρέψει τη συλλογή δεδομένων σε κάθε πτυχή της καθημερινής ζωής του ανθρώπου με ευχάριστες ή δυσάρεστες συνέπειες. Αυτή η δεύτερη κατηγορία, κυρίως οι συζητήσεις γύρω από την ιδιωτική ζωή, την ασφάλεια και την κοινωνική ευθύνη που πάει μαζί με την αυτοεπίγνωση, συνέδεσε τον κόσμο.

5.4.3 Παραδείγματα εφαρμογών

Όταν περάσαμε το κατώφλι του να συνδέουμε τελικά περισσότερα αντικείμενα απ' ότι ανθρώπους στο Διαδίκτυο, ένα τεράστιο παράθυρο άνοιξε που μας έδωσε την ευκαιρία να δημιουργήσουμε εφαρμογές σε τομείς όπως ο αυτοματισμός και η μηχανή προς μηχανή επικοινωνία. Στην πραγματικότητα οι δυνατότητες είναι σχεδόν ατελείωτες.

Τομείς των εφαρμογών περιλαμβάνουν: την πολεοδομία, τη διαχείριση αποβλήτων, το περιβάλλον, την κοινωνική αλληλεπίδραση, την αντιμετώπιση των καταστάσεων έκτακτης ανάγκης, τις έξυπνες αγορές, τις συσκευές οικιακού αυτοματισμού, τους έξυπνους μετρητές και πολλά άλλα. Τα ακόλουθα παραδείγματα τονίζουν μερικούς από τους τρόπους που το IoT αλλάζει τη ζωή των ανθρώπων προς το καλύτερο.

5.4.4 Το έξυπνο σπίτι

Τα μελλοντικά έξυπνα σπίτια θα έχουν “συνείδηση” για το τι συμβαίνει μέσα σε ένα κτίριο, κυρίως επηρεαζόμενα από τρεις πτυχές: τη χρήση των πόρων, την ασφάλεια και την άνεση. Στόχος είναι να επιτευχθούν καλύτερα επίπεδα άνεσης όπως επίσης και μείωση των συνολικών δαπανών.

Επιπλέον, τα έξυπνα σπίτια θα είναι σε θέση να αντιμετωπίσουν επαρκώς τα θέματα ασφαλείας μέσω πολύπλοκων συστημάτων ασφαλείας για την ανίχνευση πυρκαγιάς, κλοπής ή παράνομης εισόδου. Οι φορείς που εμπλέκονται σ' αυτό το σενάριο αποτελούν μία πολύ ετερογενή ομάδα.

Διάφοροι φορείς θα συνεργάζονται στο σπίτι του κάθε χρήστη, όπως εταιρίες του Διαδικτύου, κατασκευαστές συσκευών, φορείς εκμετάλλευσης τηλεπικοινωνιών, πάροχοι υπηρεσιών οπτικοακουστικών μέσων, εταιρείες προστασίας, εταιρείες κοινής ωφελείας ηλεκτρικής ενέργειας κ.α.

- **Χρήση ενέργειας και νερού.** Η κατανάλωση ενέργειας και νερού παρακολουθείται ώστε να ληφθούν οι κατάλληλες συμβουλές για τη μείωση των πόρων και του κόστους.

- **Απομακρυσμένες συσκευές ελέγχου.** Ενεργοποίηση και απενεργοποίηση συσκευών εξ' αποστάσεως για την αποφυγή ατυχημάτων και εξοικονόμηση της ενέργειας.
- **Συστήματα ανίχνευσης εισβολής.** Ανίχνευση των παραθύρων και των θυρών ώστε να γίνονται αντιληπτές οι παραβιάσεις και να αποτρέπουν τους εισβολείς.
- **Περιμετρικός έλεγχος πρόσβασης.** Έλεγχος πρόσβασης σε ζώνες περιορισμένης πρόσβασης και εντοπισμός των ατόμων σε μη εγκεκριμένες περιοχές.

5.4.5 Έξυπνες πόλεις

Παρά το γεγονός ότι ο όρος έξυπνη πόλη εξακολουθεί να είναι μία ασαφής έννοια, υπάρχει μία γενική συμφωνία ότι είναι μία αστική περιοχή που δημιουργεί βιώσιμη ανάπτυξη και υψηλή ποιότητα ζωής. Το μοντέλο αποσαφηνίζει τα χαρακτηριστικά μιας έξυπνης πόλης, περιλαμβάνοντας την οικονομία, τους ανθρώπους, τη διακυβέρνηση, την κινητικότητα, το περιβάλλον και τη διαβίωση. Ξεπερνώντας αυτούς τους βασικούς τομείς η υλοποίηση μπορεί να επιτευχθεί μέσω της ισχυρής δύναμης του ανθρώπου ή του κοινωνικού κεφαλαίου και των υποδομών. Για το τελευταίο μία πρώτη ανάλυση των επιχειρήσεων καταλήγει στο συμπέρασμα ότι πολλοί τομείς και κλάδοι θα επωφεληθούν από πιο ψηφιοποιημένες και έξυπνες πόλεις.

- **Μελέτη κατάστασης.** Παρακολούθηση των δονήσεων και των υλικών συνθηκών σε κτίρια, γέφυρες και ιστορικά μνημεία.
- **Χάρτες αστικού θορύβου.** Παρακολούθηση ήχου σε περιοχές που υπάρχουν μπαρ και σε κεντρικές ζώνες σε πραγματικό χρόνο.

- **Ανίχνευση Smartphone.** Εντοπισμός συσκευών iPhone και Android και γενικά οποιασδήποτε συσκευής λειτουργεί με WiFi ή Bluetooth διεπαφές.
- **Έξυπνος φωτισμός.** Ευφυής και προσαρμόσιμος φωτισμός με βάση τις καιρικές συνθήκες στα φώτα του δρόμου.
- **Διαχείριση απορριμμάτων.** Ανίχνευση των επιπέδων των σκουπιδιών για τη βελτιστοποίηση των δρομολογίων συλλογής απορριμμάτων.
- **Έξυπνοι δρόμοι.** Ευφυείς αυτοκινητόδρομοι με προειδοποιητικά μηνύματα και εκτροπές ανάλογα με τις κλιματικές συνθήκες και απροσδόκητα γεγονότα όπως ατυχήματα ή κυκλοφοριακή συμφόρηση.

5.4.6 Έξυπνη βιομηχανία

Σε μία παγκόσμια αλυσίδα εφοδιασμού, οι εταιρείες θα είναι σε θέση να παρακολουθούν όλα τα προϊόντα τους μέσω ετικετών αναγνώρισης ραδιοσυχνότητας. Ως εκ τούτου, οι εταιρείες θα μειώσουν τα λειτουργικά τους έξοδα και θα βελτιώσουν την παραγωγικότητά τους. Επίσης, η συντήρηση των μηχανημάτων θα διευκολυνθεί από τους συνδεδεμένους αισθητήρες, επιτρέποντας την παρακολούθηση σε πραγματικό χρόνο, της καλής λειτουργίας και της απόδοσης του εξοπλισμού του εργοστασίου.

Σε γενικές γραμμές, το IoT θα παρέχει αυτόματες διαδικασίες που συνεπάγονται τη δραστική μείωση του αριθμού των εργαζομένων που χρειάζονται. Οι εργαζόμενοι θα αντικατασταθούν με bar code scanners, αναγνώστες, αισθητήρες και ενεργοποιητές, και τελικά από πολύπλοκα ρομπότ τόσο αποτελεσματικά όσο ένα ανθρώπινο ον.

Χωρίς καμία αμφιβολία, οι τεχνολογίες αυτές θα φέρουν ευκαιρίες για τους εργαζόμενους σε νέες υπαλληλικές θέσεις κι ένας μεγάλος αριθμός τεχνικών θα είναι απαραίτητος για τον προγραμματισμό και την επισκευή

αυτών των μηχανημάτων. Αυτό είναι συνώνυμο με την ανάγκη δημιουργίας νέων θέσεων εργασίας κυρίως στον τομέα της συντήρησης αλλά αποτελεί επίσης μία νέα πρόκληση προκειμένου να προχωρήσουμε και να εξελίξουμε τέτοια είδη εργασίας για την αποφυγή της ανεργίας.

- **Ποιότητα εσωτερικού αέρα.** Παρακολούθηση τοξικών επιπέδων φυσικού αερίου και οξυγόνου μέσα στους χώρους εργασίας για την εξασφάλιση της ασφάλειας των εργαζομένων και των εμπορευμάτων.
- **Παρακολούθηση θερμοκρασίας.** Έλεγχος της θερμοκρασίας στο εσωτερικό των βιομηχανιών και σε ψυγεία που περιέχουν ευαίσθητα εμπορεύματα.
- **Παρουσία του όζοντος.** Παρακολούθηση των επιπέδων του όζοντος κατά τη διάρκεια της διαδικασία ξήρανσης του κρέατος σε εργοστάσια τροφίμων.
- **Παρακολούθηση εσωτερικού χώρου.** Αξιολόγηση εσωτερικού χώρου με χρήση ενεργού ZigBee και παθητικών ετικετών (RFID/NFC).

5.4.7 Έξυπνο περιβάλλον

Η έννοια του έξυπνου περιβάλλοντος προωθεί την ιδέα ενός φυσικού κόσμου που είναι πλούσιος και ορατά συνυφασμένος με αισθητήρες, ενεργοποιητές, οθόνες και υπολογιστικά στοιχεία, που είναι ενταγμένα αρμονικά στα καθημερινά αντικείμενα της ζωής μας και συνδέονται μέσω ενός συνεχούς δικτύου. Έχουν περιγραφεί ως το υποπρόϊόν των υπολογιστικών συστημάτων καθιστώντας την ανθρώπινη αλληλεπίδραση με το σύστημα μία ευχάριστη εμπειρία.

- **Πυρανίχνευση δασικών περιοχών.** Παρακολούθηση των αερίων καύσης και των συνθηκών που ευνοούν μία ενδεχόμενη πυρκαγιά προκειμένου να οριστούν οι επικίνδυνες ζώνες.

- **Ατμοσφαιρική ρύπανση.** Έλεγχος των εκπομπών CO₂ της ρύπανσης των εργοστασίων που εκπέμπονται από τα αυτοκίνητα και των τοξικών αερίων που παράγονται σε αγροκτήματα.
- **Παρακολούθηση επιπέδων χιονιού.** Μέτρηση της στάθμης του χιονιού και ενημέρωση σε πραγματικό χρόνο για την ποιότητα του χιονιού στις πίστες του σκι και τον κίνδυνο χιονοστιβάδων.
- **Πρόληψη κατολίσθησης και χιονοστιβάδων.** Παρακολούθηση της υγρασίας του εδάφους, των δονήσεων και της πυκνότητας της γης για την ανίχνευση επικίνδυνων φαινομένων.
- **Πρώιμη ανίχνευση σεισμών.** Κατανεμημένος έλεγχος σε συγκεκριμένες περιοχές δονήσεων.

5.4.8 Μεταφορές/Εφοδιασμός

Στις μεταφορές του εφοδιασμού, το IoT δεν βελτιώνει μόνο τα συστήματα ροής υλικών αλλά και το παγκόσμιο σύστημα εντοπισμού στίγματος και αυτόματης αναγνώρισης των εμπορευμάτων. Επίσης, αυξάνει την ενεργειακή απόδοση και κατά συνέπεια μειώνει την κατανάλωση ενέργειας.

Έτσι, το IoT αναμένεται να επιφέρει βαθιές αλλαγές στην παγκόσμια αλυσίδα εφοδιασμού, μέσω της 'έξυπνης' μετακίνησης φορτίου. Αυτό θα επιτευχθεί μέσω του συνεχούς συγχρονισμού των πληροφοριών της εφοδιαστικής αλυσίδας και της απρόσκοπτης παρακολούθησης σε πραγματικό χρόνο. Θα καταστήσει την αλυσίδα εφοδιασμού διαφανή, ορατή και ελεγχόμενη, επιτρέποντας την έξυπνη επικοινωνία μεταξύ των ανθρώπων και των φορτίων/εμπορευμάτων.

- ❖ **Κυκλοφοριακή Συμφόρηση.** Παρακολούθηση των οχημάτων αλλά και των πεζών για την βελτιστοποίηση της οδήγησης και της κίνησης των φαναριών.

- ❖ **Έξυπνη Στάθμευση.** Παρακολούθηση των χώρων στάθμευσης για την διαθεσιμότητα των ελεύθερων θέσεων στην πόλη.
- ❖ **Ποιότητα των συνθηκών φόρτωσης.** Παρακολούθηση των δονήσεων, χτυπημάτων και έλεγχος των δεμάτων προκειμένου να εξασφαλιστεί ότι παραμένουν κλειστά καθ' όλη τη διάρκεια του ταξιδιού.
- ❖ **Θέση αντικειμένων.** Αναζήτηση μεμονωμένων αντικειμένων σε αποθήκες ή λιμάνια.
- ❖ **Αποθήκευση/Ασυμβατότητα/Ανίχνευση.** Προειδοποίηση εκπομπών σε μεταφορές εύφλεκτων εμπορευμάτων και απομόνωση αυτών που περιέχουν εκρηκτικά υλικά.
- ❖ **Παρακολούθηση Προϊόντων.** Έλεγχος των διαδρομών που ακολουθούν ευαίσθητα προϊόντα όπως φάρμακα, κοσμήματα ή επικίνδυνα εμπορεύματα.

5.4.9 Λιανική Πώληση

Το IoT αντιλαμβάνεται τόσο τις ανάγκες των πελατών όσο και τις ανάγκες των επιχειρήσεων: συγκρίνει την τιμή ενός προϊόντος σε σχέση με άλλα προϊόντα της ίδιας ποιότητας με χαμηλότερη τιμή και δίνει πληροφορίες όχι μόνο στους πελάτες αλλά και στα καταστήματα και επιχειρήσεις. Έχοντας αυτές τις πληροφορίες σε πραγματικό χρόνο βοηθά τις επιχειρήσεις να βελτιώσουν τις αγορές τους και να ικανοποιήσουν τις ανάγκες των πελατών.

Προφανώς, μεγάλες αλυσίδες λιανικής πώλησης θα εκμεταλλευτούν τη δεσπόζουσα θέση τους για να ενδυναμώσουν και να επιβάλλουν τη μελλοντική IoT αγορά λιανικής πώλησης. Ειδικότερα, οι εταιρείες με θέσεις ελέγχου θα είναι ικανές να ωθήσουν την υιοθέτηση της τεχνολογίας IoT λόγω των τεράστιων μεριδίων των αγορών τους.

- **Έλεγχος Προμηθειών.** Παρακολούθηση των συνθηκών αποθήκευσης και παρακολούθηση προϊόντων.
- **Πληρωμή NFC.** Επεξεργασία των πληρωμών με βάση την περιοχή ή την διάρκεια της δραστηριότητας για τα μέσα μαζικής μεταφοράς, τα γυμναστήρια, τα θεματικά πάρκα κ.α.
- **Εφαρμογές Έξυπνης Αγοράς.** Παροχή συμβουλών στο σημείο πώλησης σύμφωνα με τις συνήθειες των πελατών, τις προτιμήσεις και ενημέρωση για αλλεργικά συστατικά ή ημερομηνίες λήξης.
- **Έξυπνη Διαχείριση Προϊόντων.** Έλεγχος των προϊόντων στα ράφια και στις αποθήκες για την αυτοματοποίηση των διαδικασιών εμπλουτισμού του αποθέματος.

5.4.10 Ιατρική Παρακολούθηση Εξ' αποστάσεως

Ο έλεγχος και η πρόληψη είναι δύο από τους βασικούς στόχους της διατήρησης της υγείας. Ήδη σήμερα, οι άνθρωποι έχουν την δυνατότητα να παρακολουθούνται και να ελέγχονται από τους ειδικούς ακόμα και στην περίπτωση που βρίσκονται σε διαφορετικό μέρος. Η παρακολούθηση του ιστορικού της υγείας των ανθρώπων είναι μία άλλη πτυχή του IoT στην οποία δείχνει μεγάλη ευελιξία. Επιχειρηματικές εφαρμογές θα μπορούσαν να προσφέρουν την δυνατότητα ιατρικών υπηρεσιών όχι μόνο στους ασθενείς αλλά και στους ειδικούς, οι οποίοι χρειάζονται πληροφορίες για να προχωρήσουν στην ιατρική αξιολόγησή τους. Στον τομέα αυτό, το IoT καθιστά την ανθρώπινη αλληλεπίδραση πολύ πιο αποτελεσματική αφού επιτρέπει όχι μόνο τον εντοπισμό αλλά και την παρακολούθηση των ασθενών. Η παροχή πληροφοριών σχετικά με την κατάσταση του ασθενή καθιστά την όλη διαδικασία πιο αποτελεσματική και επίσης αφήνει τους ανθρώπους πολύ πιο ικανοποιημένους. Οι πιο σημαντικοί παράγοντες σε αυτό το σενάριο θα είναι τα δημόσια και ιδιωτικά νοσοκομεία και τα ιδρύματα. Αξίζει να σημειωθεί ότι οι φορείς εκμετάλλευσης τηλεπικοινωνιών είναι αρκετά ενεργοί στην ηλεκτρονική υγεία.

- ❖ **Ανίχνευση Πτώσης.** Βοήθεια για ηλικιωμένους και άτομα με ειδικές ανάγκες που ζουν ανεξάρτητοι.
- ❖ **Ιατρικά Ψυγεία.** Έλεγχος των συνθηκών μέσα σε καταψύκτες αποθήκευσης εμβολίων, φαρμάκων και οργανικών στοιχείων.
- ❖ **Φροντίδα Αθλητών.** Σημεία ελέγχου σε κέντρα υψηλής απόδοσης.
- ❖ **Επιτήρηση Ασθενών.** Παρακολούθηση των ασθενών μέσα στα νοσοκομεία αλλά και τα γηροκομεία.
- ❖ **Υπεριώδης Ακτινοβολία.** Μέτρηση της υπεριώδους ακτινοβολίας του ήλιου UV προκειμένου να προειδοποιηθεί ο κόσμος ώστε να αποφεύγεται η έκθεση στον ήλιο συγκεκριμένες ώρες.
- ❖ **Καλύτερη ποιότητα ζωής για τους ηλικιωμένους.** Ο παγκόσμιος πληθυσμός αυξάνεται. Στην πραγματικότητα, περίπου 1 δισεκατομμύριο άτομα ηλικίας 65 ετών και άνω θα πρέπει να ταξινομούνται ενώ έχουν φθάσει σε «μη εργάσιμη ηλικία» από τα μέσα του αιώνα. Το IoT μπορεί να βελτιώσει σημαντικά την ποιότητα ζωής για τον αυξανόμενο αριθμό των ηλικιωμένων. Για παράδειγμα, φανταστείτε μία μικρή φορητή συσκευή προσαρμοσμένη πάνω μας που μπορεί να ανιχνεύσει τα ζωτικά σημεία ενός ατόμου και να στείλει μια προειδοποίηση σε έναν επαγγελματία του τομέα της υγείας, όταν έχει επιτευχθεί ένα ορισμένο όριο, ή την αίσθηση όταν ένα άτομο έχει πέσει κάτω και δεν μπορεί να σηκωθεί.

5.4.11 Έξυπνη Ενέργεια

Το πεδίο αυτό έχει πολλές συσχετίσεις με άλλα σενάρια, όπως το έξυπνο σπίτι και την έξυπνη πόλη. Το βασικό ζήτημα σε αυτά τα σενάρια είναι να εντοπίσει τρόπους για την εξοικονόμηση ενέργειας. Ουσιαστικά αναφερόμαστε σε αυτό που είναι γνωστό ως ένα έξυπνο δίκτυο. Σε αυτόν τον τομέα εφαρμογής, τις πρωτοβουλίες που συνεπάγεται μια πιο κατανομημένη παραγωγή ενέργειας θα πρέπει να επισημανθεί, όπως πολλά σπίτια σήμερα

έχουν ένα ηλιακό πάνελ, για παράδειγμα. Ως ζωτικής σημασίας συστατικό, οι έξυπνοι μετρητές θεωρούνται απαραίτητη προϋπόθεση για την ενεργοποίηση της έξυπνης παρακολούθησης, τον έλεγχο και την επικοινωνία σε εφαρμογές πλέγματος.

Έτσι, συνδυάζοντας την ανάλυση της προσφοράς και της ζήτησης, οι επιχειρήσεις ενέργειας θα είναι σε θέση να παρέχουν πιο αποτελεσματική διαμόρφωση της ζήτησης. Δεν θα παρέχουν μόνο κίνητρα για τους καταναλωτές, αλλά στην πραγματικότητα θα απενεργοποιήσουν συσκευές που δεν χρειάζονται. Επιπλέον, οι ενέργειες αυτές πρέπει να πραγματοποιηθούν αυτόματα. Εδώ, αντιμετωπίζουμε και πάλι ένα ετερογενές σενάριο με τη συμμετοχή διάφορων ενδιαφερόμενων. Οι κύριοι παράγοντες είναι φυσικά οι εταιρείες κοινής ωφέλειας της ενέργειας, αλλά και οι δημόσιοι φορείς θα είναι επίσης σημαντικοί παράγοντες.

- Έξυπνοι Θερμοστάτες
- Ελεγκτές φόρτωσης
- Plug-in ηλεκτρικά οχήματα
- Έξυπνες συσκευές
- Επεκτάσεις εύρους
- Ενεργειακές Υπηρεσίες Διασύνδεσης
- Μετατροπείς

5.4.12 Έξυπνη Κτηνοτροφία

Στον κόσμο του Ίντερνετ των πραγμάτων, ακόμα και οι αγελάδες θα πρέπει να είναι συνδεδεμένες. Μια ειδική έκθεση του Economist με τίτλο «επαυξημένη Επιχειρηματικότητα» περιγράφει το πώς θα πρέπει να παρακολουθούνται οι αγελάδες. Έτσι, μια ολλανδική νεοσύστατη εταιρεία τοποθέτησε εμφυτεύματα αισθητήρες στα αυτιά των βοοειδών. Αυτό επιτρέπει

στους αγρότες να παρακολουθούν την υγεία των αγελάδων και τις κινήσεις τους, εξασφαλίζοντας μία υγιή, πιο άφθονη προμήθεια κρέατος για τους ανθρώπους να καταναλώνουν. Κατά μέσο όρο, κάθε αγελάδα παράγει περίπου 200 MB ενημέρωσης κατ' έτος.

- **Υδροπονία.** Έλεγχος των συνθηκών των φυτών που αναπτύσσονται στο νερό προκειμένου να παρθεί η μέγιστη απόδοση των καλλιεργειών.
- **Φροντίδα απογόνων.** Έλεγχος των συνθηκών καλλιέργειας των απογόνων σε φάρμες ζώων για να διασφαλιστεί η επιβίωση και η υγεία τους.
- **Παρακολούθηση ζώων.** Εντοπισμός και αναγνώριση των ζώων που βόσκουν στα λιβάδια ή σε μεγάλους στάβλους.
- **Μελέτη επιπέδου τοξικού αερίου.** Μελέτη αερισμού και ποιότητα του αέρα σε αγροκτήματα για την ανίχνευση των επιβλαβών αερίων από περιπτώματα.

5.4.13 Έξυπνη Μέτρηση

Έξυπνες εφαρμογές μέτρησης προσφέρουν τα οφέλη της βελτιωμένης αξιοπιστίας και ακρίβειας, την ευκολία της βαθμονόμησης, της ασφάλειας και προηγμένα χαρακτηριστικά χρέωσης όπως επίσης την ευκαιρία για καλύτερη διαχείριση της χρήσης της ενέργειας.

- ❖ **Έξυπνη Παρακολούθηση.** Παρακολούθησης της κατανάλωσης ενέργειας και της διαχείρισης.
- ❖ **Παρακολούθηση επιπέδου στάθμης.** Παρακολούθηση της στάθμης του νερού, πετρελαίου και φυσικού αερίου σε δεξαμενές αποθήκευσης και βυτία.
- ❖ **Φωτοβολταϊκές εγκαταστάσεις.** Παρακολούθηση και βελτιστοποίηση της απόδοσης στον τομέα της ηλιακής ενέργειας.

- ❖ **Ροή νερού.** Μέτρηση της πίεσης του νερού σε συστήματα μεταφοράς νερού.
- ❖ **Μέτρηση του υλικού.** Μέτρηση του επιπέδου κενού και του βάρους του εμπορεύματος.

5.4.14 Έξυπνη Γεωργία

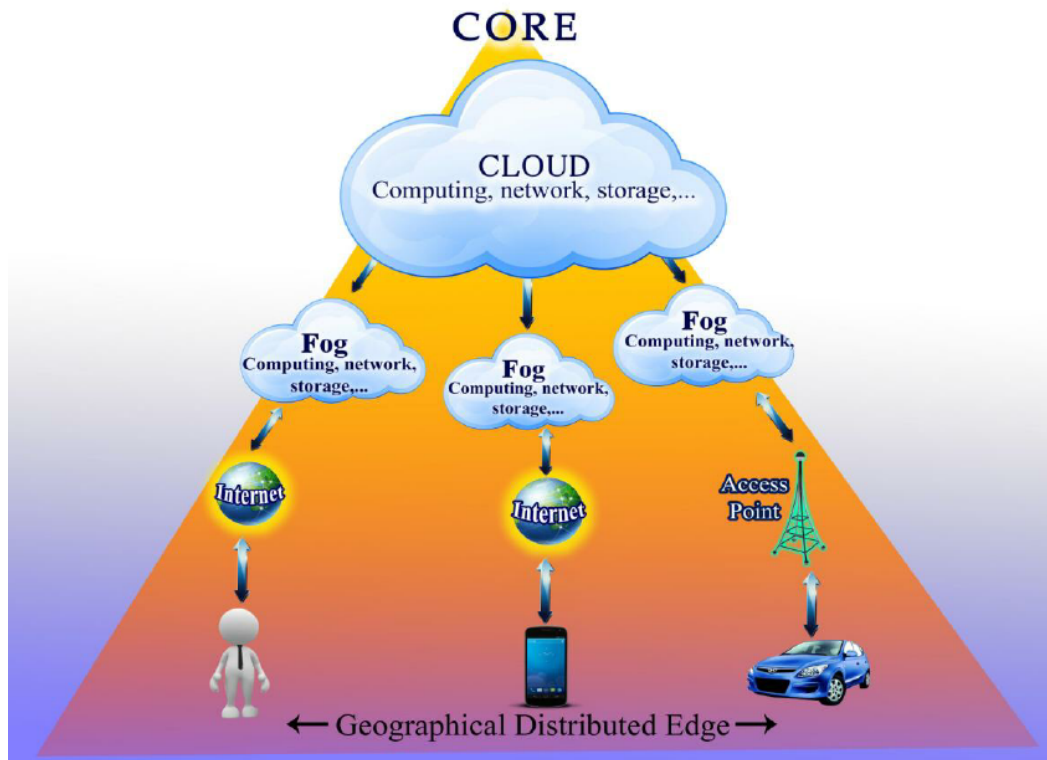
Υπάρχει ένα ευρύ φάσμα της γεωργίας που βασίζεται σε διάφορες πρακτικές και εφαρμογές που έχουν τη δυνατότητα να αυξήσουν την παραγωγή τροφίμων και την ικανότητα προσαρμογής του συστήματος παραγωγής τροφίμων, καθώς και τη μείωση των εκπομπών όπως επίσης και να ενισχύσουν την αποθήκευση του άνθρακα στα γεωργικά εδάφη και τη βιομάζα.

- **Βελτίωση της ποιότητας κρασιού.** Παρακολούθηση της υγρασίας του εδάφους και της διαμέτρου του κορμού σε αμπελώνες προκειμένου να πραγματοποιηθεί έλεγχος στην ποσότητα της ζάχαρης στα σταφύλια και της υγείας του αμπελιού.
- **Θερμοκήπια.** Έλεγχος συνθηκών μικροκλίματος για τη μεγιστοποίηση της παραγωγής φρούτων και λαχανικών και την ποιότητά τους.
- **Γήπεδα του γκολφ.** Επιλεκτική άρδευση στις ξηρές ζώνες για τη μείωση των υδάτινων πόρων που απαιτούνται στο πράσινο.
- **Μετεωρολογικοί σταθμοί.** Μελέτη των καιρικών συνθηκών σε τομείς προκειμένου να προβλεφθούν οι αλλαγές στον σχηματισμό πάγου, στη βροχή, στην ξηρασία, στο χιόνι ή στον άνεμο.
- **Κοπρόχωμα.** Έλεγχος της υγρασίας και της θερμοκρασίας στα επίπεδα των αλφάλα (superfood σαν το τριφύλλι), σανό, άχυρο, κλπ. για την πρόληψη μυκήτων και άλλων μικροβιακών ρύπων.

ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ

6.1 FOG COMPUTING

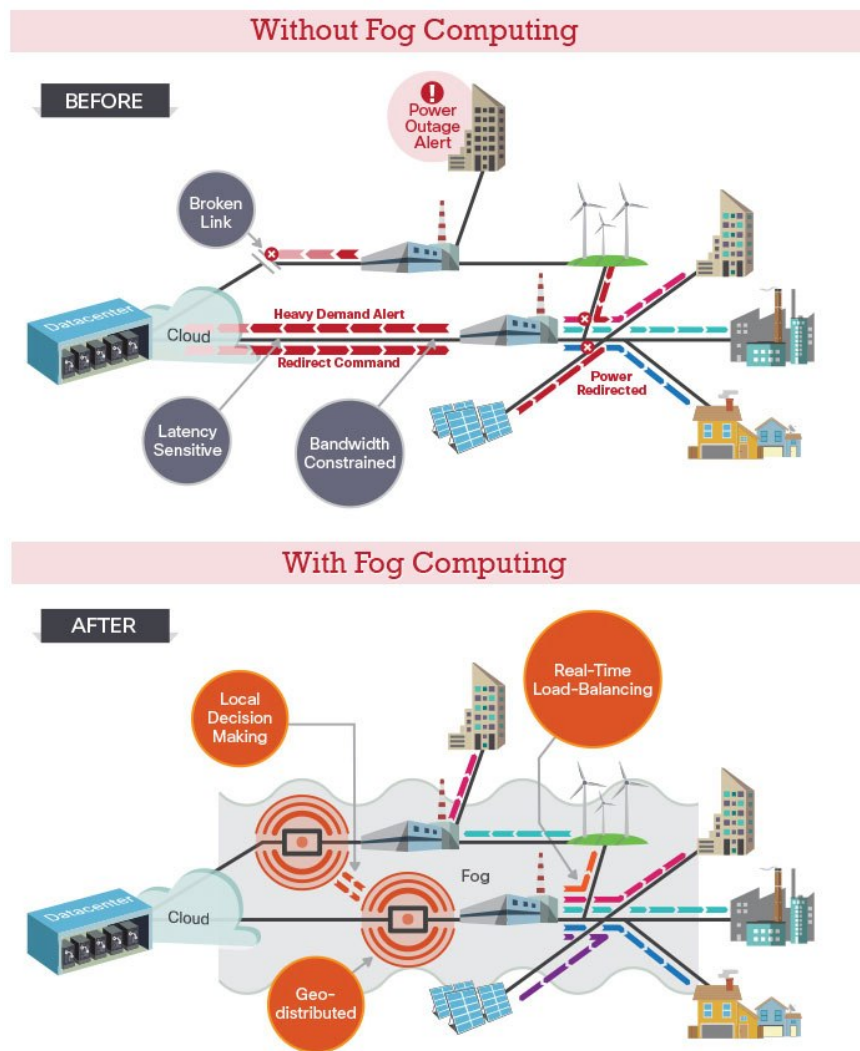
Στην παρακάτω εικόνα απεικονίζεται η αρχιτεκτονική ενός υπολογιστικού περιβάλλοντος ομίχλης σε μια επιχείρηση. Παρατηρείται ότι το δίκτυο ομίχλης είναι πιο κοντά στις έξυπνες συσκευές, η επεξεργασία των δεδομένων συμβαίνει πιο κοντά στις συσκευές και η επεξεργασμένη πληροφορία διοχετεύεται στο περιβάλλον cloud.



Εικόνα 14 : Αρχιτεκτονική Fog

Το Cloud computing [12] [13] ανήκει στο παρελθόν και έχει πλέον ξεπεραστεί από μια νέα έννοια που ονομάζεται υπολογιστική ομίχλη η οποία είναι σίγουρα πολύ καλύτερη και μεγαλύτερη από το σύννεφο. Το fog computing είναι αρκετά παρόμοιο με το cloud computing και ακριβώς όπως το cloud παρέχει επίσης στους χρήστες του δεδομένα, αποθήκευση, υπολογιστική και υπηρεσίες εφαρμογών. Τα χαρακτηριστικά που διακρίνουν την ομίχλη από το σύννεφο είναι η υποστήριξη της κινητικότητας, η εγγύτητά της προς τους τελικούς χρήστες και η πυκνή γεωγραφική κατανομή του. Οι υπηρεσίες της φιλοξενούνται στην άκρη του δικτύου ή ακόμα και σε συσκευές όπως set-top-box ή τα σημεία πρόσβασης. Με τον τρόπο αυτό, η υπολογιστική ομίχλη βοηθά στη μείωση του χρόνου της καθυστέρησης υπηρεσίας και ακόμη βελτιώνει το QoS, το οποίο έχει ως αποτέλεσμα μια ανώτερη εμπειρία χρήστη. Η υπολογιστική ομίχλη υποστηρίζει ακόμη τις εφαρμογές του Ίντερνετ των πραγμάτων (IoT) που απαιτούν προβλεπόμενη καθυστέρηση ή καθυστέρηση σε πραγματικό χρόνο. Ένα πράγμα στο Ίντερνετ των πραγμάτων αναφέρεται ως οποιοδήποτε φυσικό αντικείμενο ή αντικείμενο που φτιάχτηκε από ανθρώπινο χέρι που μπορεί να εκχωρηθεί μια IP Πρωτοκόλλου Διαδικτύου και παρέχεται με την ικανότητα να μεταφέρει δεδομένα μέσω δικτύου. Μερικά από αυτά μπορεί να καταλήξουν να δημιουργούν πολλά δεδομένα. Η Cisco μας δίνει ένα παράδειγμα ενός κινητήρα τζετ, η οποία είναι ικανή να δημιουργήσει 10 terabytes δεδομένων σχετικά με την κατάσταση και τις επιδόσεις του το πολύ σε μισή ώρα. Μεταδίδοντας όλα αυτά τα δεδομένα στο cloud και στη συνέχεια μεταδίδοντας απάντηση πίσω, καταλήγουμε να δημιουργήσουμε μια τεράστια ζήτηση σε εύρος ζώνης. Αυτή η διαδικασία απαιτεί περαιτέρω ένα σημαντικό χρονικό διάστημα να λάβει χώρα και μπορεί να υποφέρει από μεγάλη καθυστέρηση. Στο fog computing, ένα μεγάλο μέρος της επεξεργασίας λαμβάνει χώρα σε ένα router. Αυτό το είδος των υπολογισμών δημιουργεί μια εικονική πλατφόρμα που παρέχει υπηρεσίες δικτύωσης, υπολογιστικής και αποθήκευσης μεταξύ των παραδοσιακών κέντρων δεδομένων cloud computing και τις τελικές συσκευές. Οι υπηρεσίες αυτές είναι κεντρικής σημασίας και για την ομίχλη και το cloud computing. Είναι επίσης σημαντικό

για τη στήριξη των αναδυόμενων εφαρμογών του Internet. Το fog computing έχει επίσης την ικανότητα να ενεργοποιήσει μια νέα κατηγορία συγκεντρωτικών υπηρεσιών και εφαρμογών, όπως η έξυπνη κατανομή ενέργειας. Στην έξυπνη διανομή ενέργειας, όλες οι εφαρμογές εξισορρόπησης του φορτίου της ενέργειας θα τρέχουν σε συσκευές στην άκρη του δικτύου, που θα μεταβαίνουν αυτόματα σε εναλλακτικές πηγές ενέργειας, όπως η αιολική και η ηλιακή, κλπ, με βάση τη διαθεσιμότητα, τη ζήτηση και τις χαμηλότερες τιμές. Η χρήση της υπολογιστικής ομίχλης μπορεί να επιταχύνει τη διαδικασία της καινοτομίας, με τρόπους που δεν έχουμε δει ποτέ πριν. Αυτό περιλαμβάνει εφαρμογές αυτο-ίασης, αυτο-οργάνωσης και αυτο-μόρφωσης για βιομηχανικά δικτυακά προϊόντα.



Εικόνα 15 : Πλεονεκτήματα Fog Computing

6.2 FOG COMPUTING ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ ΚΑΙ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ

Η Υπολογιστική Ομίχλη είναι ένα ιδιαίτερα κατακεντρωμένο ευρέως αποκεντρωμένο "σύννεφο" που λειτουργεί κοντά στο επιχειρησιακό επίπεδο, όπου δημιουργούνται τα δεδομένα και χρησιμοποιούνται πιο συχνά. Η ομίχλη

στο επίπεδο του εδάφους είναι μια εξαιρετική επιλογή για εφαρμογές που απαιτούν υπολογιστική δύναμη κοντά τους, όπου υπάρχει μεγάλος όγκος σε πραγματικού χρόνου ή / και χρονικά κρίσιμων τοπικών δεδομένων, τα οποία έχουν τη μεγαλύτερη σημασία στο πλαίσιο τους. Όπου ο γρήγορος εντοπισμός αποτελεσμάτων είναι σημαντικός και καθιστά την αποστολή πολλών μη επεξεργασμένων δεδομένων στο "σύννεφο" περιττή, ανεπιθύμητη ή το εύρος ζώνης είναι ακριβό ή περιορισμένο.

Παραδείγματα εφαρμογών της υπολογιστικής ομίχλης μέσα σε ένα βιομηχανικό περιβάλλον είναι η ανάλυση, η βελτιστοποίηση και ο προηγμένος έλεγχος σε ένα κέντρο εργασίας κατασκευής, μονάδα λειτουργίας, εντός και μεταξύ των μονάδων-επιχειρήσεις, όπου οι αισθητήρες, ελεγκτές, ιστορικές, αναλυτικές μηχανές μοιράζονται δεδομένα διαδραστικά σε πραγματικό χρόνο. Στα άνω άκρα της "ομίχλης" είναι η τοπική υπολογιστική δύναμη, όπως τα συστήματα παραγωγής που εκτείνονται σε κέντρα εργασίας και τις λειτουργίες της μονάδας, υψηλότερα ακόμα θα είναι περιφερειακά σύννεφα και, τέλος, το σύννεφο σε επίπεδο επιχείρησης. Η υπολογιστική ομίχλη δεν είναι ανεξάρτητη του cloud computing, αλλά συνδέεται με την αποστολή επεξεργασμένων συνοπτικών πληροφοριών και σε αντάλλαγμα λαμβάνει πληροφορίες που απαιτούνται σε τοπικό επίπεδο.

Το fog computing τοποθετεί την διαχείριση δεδομένων, την υπολογιστική δύναμη, την απόδοση, την αξιοπιστία και την ανάκτηση στα χέρια των ανθρώπων που κατανοούν τις ανάγκες? οι επιχειρηματίες, οι μηχανικοί και το προσωπικό IT για τη λειτουργία της μονάδας, μια πλατφόρμα πετρελαίου και φυσικού αερίου, ή άλλη τοπική λειτουργία, έτσι ώστε να μπορεί να προσαρμοστεί με σκοπό την υψηλή ταχύτητα σε περιβάλλον πραγματικού χρόνου. [15] [16]

Επίσης, μειώνει τις ανάγκες του εύρους ζώνης, όπως απαιτείται το 80% του συνόλου των δεδομένων στο τοπικό πλαίσιο, όπως οι πιέσεις, οι θερμοκρασίες, τα υλικά, τα ποσοστά ροής. Για να σταλούν αυτές οι πληροφορίες σε πραγματικό χρόνο μέσα στο νέφος της επιχείρησης θα είναι

μεγάλο βάρος σε εύρος ζώνης και αποθήκευση. Μπορεί να παρουσιαστεί υπερχειλίση δεδομένων σε βάσεις, για πληροφορίες που σπάνια χρησιμοποιούνται σε αυτό το επίπεδο. Με αυτόν τον τρόπο μια περιορισμένη ποσότητα επεξεργασμένων πληροφοριών μπορεί να μεταδοθεί μέχρι το σύννεφο, αλλά και από το σύννεφο στην τοπική λειτουργία, όπως η ανατροφοδότηση της απόδοσης των προϊόντων από τους πελάτες πίσω στην πηγή των προϊόντων.

Τοποθετείται υπολογιστική δύναμη όπου αυτό είναι αναγκαίο, και αποδοτικό, κατάλληλο για τον σκοπό, να βρίσκεται όπου πρέπει να είναι, σε ένα κέντρο εργασίας, μέσα σε ένα πίνακα ελέγχου, σε ένα γραφείο, σε ένα εργαστήριο, σε ένα ράφι σε ένα κέντρο δεδομένων, οπουδήποτε και παντού, όλα τα δεδομένα που σχετίζονται με τον διαμοιρασμό να κατανοηθούν και να βελτιωθεί η απόδοση. Ενώ βρίσκεται παντού σε έναν οργανισμό, η υπολογιστική ομίχλη λειτουργεί ως ένας ενιαίος πόρος, σαν ένα κατανεμημένο cloud χαμηλού επιπέδου που ενσωματώνεται με τα κεντρικά σύννεφα για την απόκτηση της ανατροφοδότησης της αγοράς και των πελατών, τις επιθυμίες και τη συμπεριφορά του που αντικατοπτρίζουν την απόδοση του προϊόντος στα μάτια του πελάτη.

Τα χαρακτηριστικά ενός συστήματος ομίχλης computing είναι:

- Ένα ιδιαίτερα κατανεμημένο υπολογιστικό σύστημα (HDCC).
- Ένα πλέγμα peer-to-peer των υπολογιστικών κόμβων σε μια εικονικά ιεραρχική δομή που να ταιριάζει με τον οργανισμό
- Να επικοινωνεί με έξυπνους αισθητήρες, ελεγκτές, ιστορικούς, συστήματα ποιότητας και ελέγχου υλικών και άλλα όπως οι ομότιμοι
- Να τρέχει σε προσιτές τιμές
- Να υποστηρίζει πολλαπλές πλατφόρμες λειτουργικών Unix, Windows, Mac

- Να χρησιμοποιεί απλή, γρήγορη και τυποποιημένα πρωτόκολλα IoT internet (TCP / IP, πρίζες, κλπ)
- Εμπειρία χρήστη σε φυλλομετρητή, μετά από όλα, είναι η βασική πτυχή της «Βιομηχανικής Ίντερνετ των πραγμάτων»
- Χτισμένο πάνω σε υψηλής απόδοσης κατανεμημένη υπολογιστική τεχνολογία

Η Σύλληψη, η αποθήκευση, η επικύρωση, ο καθαρισμός και το φιλτράρισμα, η ενσωμάτωση, η ανάλυση, η πρόβλεψη, η προσαρμογή και η βελτιστοποίηση των επιδόσεων σε χαμηλότερα επίπεδα σε ολόκληρη την επιχείρηση σε πραγματικό χρόνο απαιτεί (HPC) υπολογιστική ενέργεια υψηλής απόδοσης. Αυτό δεν σημαίνει απαραίτητα υψηλό κόστος, καθώς εμπορικά PC με μέτριες αποδόσεις με τη δύναμη ενός τυπικού φορητού υπολογιστή θα είναι αρκετά και το λογισμικό που τρέχει το σύστημα δεν χρειάζεται να είναι ακριβό.

Για την σχεδίαση ενός τέτοιου συστήματος, αντλούνται στοιχεία από την εμπειρία, τις αρχιτεκτονικές, τα εργαλεία και τις επιτυχίες των γιγάντων της πληροφορικής όπως η Google, Amazon, το YouTube, το Facebook, το Twitter και άλλα. Έχουν δημιουργήσει ισχυρές υπολογιστικές αρχιτεκτονικές υψηλής απόδοσης που εκτείνονται σε παγκόσμια κέντρα δεδομένων. Έχουν παράσχει εργαλεία ανάπτυξης και γλώσσες, όπως την Google GO (golang) που είναι κατάλληλες για υψηλής ταχύτητας παράλληλη κατανεμημένη επεξεργασία και ισχυρές υπηρεσίες δικτύωσης και web services. Έχοντας μια παρόμοια ανάγκη, αλλά πιο καλά κατανεμημένη, μπορούν διάφορες επιχειρήσεις να υιοθετήσουν παρόμοιες αρχιτεκτονικές υπολογιστών υψηλών επιδόσεων για να παραδώσουν και να μοιραστούν τα αποτελέσματα εκεί που χρειάζονται σε πραγματικό χρόνο.

Υπάρχουν διάφοροι τρόποι για να χρησιμοποιηθούν οι υπηρεσίες cloud για αποθήκευση αρχείων, εγγράφων και πολυμέσων σε

απομακρυσμένες υπηρεσίες στις οποίες μπορεί να υπάρχει πρόσβαση όποτε γίνεται σύνδεση στο Internet. Το κύριο πρόβλημα σε ένα σύννεφο είναι να διατηρήσει την ασφάλεια για τα δεδομένα του χρήστη με τρόπο που να εγγυάται μόνο πιστοποιημένους χρήστες και κανέναν άλλο να αποκτήσει πρόσβαση σε αυτά τα δεδομένα. Το ζήτημα της παροχής ασφάλειας σε εμπιστευτικές πληροφορίες είναι βασικό πρόβλημα της ασφάλειας, ότι δεν παρέχει το επίπεδο βεβαιότητας που επιθυμούν οι περισσότεροι άνθρωποι. Υπάρχουν διάφοροι μέθοδοι για την προστασία απομακρυσμένων δεδομένων στο σύννεφο με τη χρήση τυποποιημένων μεθόδων ελέγχου πρόσβασης και κρυπτογράφησης. Είναι καλό να πούμε ότι όλες οι τυπικές προσεγγίσεις που χρησιμοποιούνται για την παροχή ασφάλειας έχει αποδειχθεί ότι αποτυγχάνουν από καιρό σε καιρό για διάφορους λόγους, συμπεριλαμβανομένων ελαττωματικών εφαρμογών, λάθη κώδικα, επιθέσεις εμπιστευτικών πληροφοριών, λανθασμένες υπηρεσίες, και τη δημιουργική κατασκευή αποτελεσματικών και εξελιγμένων επιθέσεων που δεν προβλέφθηκαν από τους φορείς υλοποίησης των διαδικασιών ασφαλείας. Η οικοδόμηση ενός ασφαλούς και αξιόπιστου cloud computing περιβάλλοντος δεν είναι αρκετή, γιατί οι επιθέσεις σε δεδομένα συνεχίζουν να συμβαίνουν, και όταν το κάνουν, οι πληροφορίες χάνονται και δεν υπάρχει τρόπος να επαναφερθούν. Χρειάζονται λύσεις σε τέτοια ατυχήματα. Η ασφάλεια στις υπηρεσίες Cloud μπορεί να διασφαλιστεί με την δημιουργία πρόσθετων χαρακτηριστικών ασφαλείας. Η βασική ιδέα είναι ότι μπορούμε να περιορίσουμε τη ζημιά των κλεμμένων δεδομένων, αν μειωθεί η αξία των κλεμμένων δεδομένων στον εισβολέα. Μπορούμε να το επιτύχουμε αυτό μέσω δολωμάτων-επιθέσεων (παραπληροφόρηση) προληπτικά σε συνδυασμό και με άλλα χαρακτηριστικά ασφαλείας τα οποία προσφέρει η πλατφόρμα του Fog computing.

6.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΑΣΦΑΛΕΙΑΣ FOG COMPUTING

Το Fog Computing [17] είναι ένα σύστημα σε επίπεδο οριζόντιας αρχιτεκτονικής που διανέμει τους πόρους και τις υπηρεσίες πληροφορικής και ελέγχου, αποθήκευσης και δικτύωσης και επικοινωνιών πιο κοντά στις πηγές δεδομένων. Μέσω της συνεχούς επικοινωνίας ομίχλης-πραγμάτων, η υπολογιστική ομίχλη μπορεί να λύσει τις προκλήσεις του εύρους ζώνης, της καθυστέρησης και των επικοινωνιών που συνδέονται με τα δίκτυα επόμενης γενιάς που θα χρησιμοποιούν IoT, 5G και την τεχνητή νοημοσύνη. Μέσα από αυτή την κατακεκομμένη προσέγγιση της αρχιτεκτονικής, είναι επίσης μια ιδανική αρχιτεκτονική ασφάλειας του κυβερνοχώρου. Το fog computing επιτρέπει σε βιομηχανικές επιχειρήσεις την τυποποίηση αρχιτεκτονικών ασφαλείας σε IoT πλατφόρμες, τους προμηθευτές και τους πελάτες. Από τη σχεδίαση, η ομίχλη φιλοξενεί τις μοναδικές αρχιτεκτονικές και τα τρωτά σημεία του IoT. Λειτουργεί στην περιοχή μεταξύ σύννεφου και πραγμάτων, παρά στην περίμετρο, για να προσφέρει σαφή πλεονεκτήματα.

Απαριθμούνται πέντε θεμελιώδεις τρόποι ότι το fog computing κάνει το Ίντερνετ των πραγμάτων πιο ασφαλή: [14]

- ✓ Το fog computing είναι χτισμένο από το μηδέν για την cloud-προς-πράγματα ασφάλεια IoT, προσφέροντας ένα νέο επίπεδο προστασίας πάνω και πέρα από την IT ασφάλεια. Ο κοινός παρονομαστής είναι ότι η υπολογιστική ομίχλη επιτρέπει σε ένα συνεχές υπηρεσιών, γεφυρώνοντας το χάσμα μεταξύ νέφους και των πραγμάτων. Γεμίζει το κενό αυτό, επιτρέποντας τη διανομή των υπολογιστών και τον έλεγχο, την αποθήκευση και τις δικτυακές λειτουργίες πιο κοντά στους τελικούς χρήστες συσκευών (ή «πράγματα»). Αυτό αποτελεί μια εντελώς διαφορετική υπόθεση από την παραδοσιακή IT περίμετρο ασφαλείας.
- ✓ Υπολογιστική & Έλεγχος: Με την ομίχλη, η υπολογιστική και ο έλεγχος διενεργείται σε ή κοντά στη συσκευή του τελικού χρήστη, σε αντίθεση με το να αποθηκευτούν σε απομακρυσμένα κέντρα δεδομένων ή κυψελοειδή δίκτυα. Σε αυτό το κατακεκομμένο

περιβάλλον, οι απειλές ή επιθέσεις θα χρειαστεί να περάσουν τους κόμβους της ομίχλης, οποία μπορεί να εντοπίσει γρήγορα ασυνήθιστη δραστηριότητα και να μετριαστεί πριν περάσει από το σύστημα.

- ✓ Αποθήκευση δεδομένων: Το ίδιο ισχύει και για την αποθήκευση δεδομένων στην αρχιτεκτονική ομίχλης. Τα δεδομένα που συλλέγονται από ή διασκορπισμένα σε τελικούς χρήστες οι συσκευές, διαχειρίζονται και προστατεύονται από την κατακευματισμένη υποδομή ασφαλών κόμβων ομίχλης. Ως εκ τούτου, τα δεδομένα θα πρέπει να προστατεύονται καλύτερα από ό, τι εάν αποθηκεύονται στις συσκευές του χρήστη και πιο διαθέσιμη από ότι αν διατηρηθεί σε απομακρυσμένα κέντρα δεδομένων
- ✓ Επικοινωνία & Δικτύωση: Με την ομίχλη, η επικοινωνία και η δικτύωση πραγματοποιείται σε (ή κοντά) στη συσκευή του τελικού χρήστη και δεν δρομολογείται όλη την κίνηση μέσω των δικτύων κορμού. Αυτό παρέχει επίσης ένα πλεονέκτημα προστασίας της ιδιωτικής ζωής. Σε ορισμένες περιπτώσεις, όπως αναπτύξεις που εφαρμόζουν το D2D πρότυπο ασύρματης συσκευής, η ομίχλη μειώνει τις πιθανότητες των υποκλοπών λόγω του ότι κρατούν την επικοινωνία με τα συστήματα σε κοντινή απόσταση.
- ✓ Παράλληλα, το fog computing μπορεί να προστατεύσει ακόμα και τις μικρότερες συσκευές με περιορισμένους πόρους. Οι περισσότερες IoT συσκευές είναι μικρές και έχουν σχεδιαστεί για να λειτουργούν σε περιβάλλοντα με ελάχιστους πόρους. Αυτές οι συσκευές με περιορισμένους πόρους, οι οποίες έχουν μικρή ή καθόλου ικανότητα να υπερασπιστούν τους εαυτούς τους ενάντια σε εκλεπτυσμένες επιθέσεις στον κυβερνοχώρο (όπως η επίθεση Dyn), μπορούν να συνεργάζονται με ένα κατακευματισμένο πολυεπίπεδο δίκτυο κόμβων ομίχλης και διακομιστές του cloud με

σκοπό να επιτευχθούν τα απαραίτητα επίπεδα προστασίας, σύμφωνα με το δόγμα "άμυνας εις βάθος".

Ένα λιτό λογισμικό client να υπάρχει στη συσκευή είναι το μόνο που χρειάζεται για τον εντοπισμό ύποπτων στοιχείων σε χρήση ή σε διαμετακόμιση. Αν το πρόγραμμα αυτό ή ο κόμβος ομίχλης εντοπίσει ασυνήθιστη δραστηριότητα, τα δεδομένα κατά τη χρήση ή κατά τη μεταφορά θα επισημανθούν και θα κρατηθούν σε καραντίνα. Τα ύποπτα στοιχεία και οι ασυνήθιστες δραστηριότητες μπορούν στη συνέχεια να αναλυθούν για παραβιάσεις της ασφάλειας, προστασίας της ιδιωτικής ζωής και τη διαθεσιμότητας. Επιπλέον, ένας μικρός αισθητήρας μπορεί να αναμένεται να λειτουργήσει πέντε έτη ή περισσότερο με την εσωτερική μπαταρία του, και αυτό μπορεί να μην παρέχει αρκετή ενέργεια για ισχυρή κρυπτογράφηση. Ένας κοντινός κόμβος ομίχλης μπορεί να εκτελέσει τις πιο εξελιγμένες λειτουργίες ασφαλείας που είναι απαραίτητες για την προστασία του.

Όπως αναφέρθηκε παραπάνω, μια κατανεμημένη σειρά κόμβων ομίχλης μπορεί να προσφέρει ένα πιο ασφαλές περιβάλλον από μεμονωμένες συσκευές με περιορισμένους πόρους για την αποθήκευση ευαίσθητων δεδομένων ζωτικής σημασίας. Για παράδειγμα, αν ένα βίντεο ασφαλείας που είναι αποθηκευμένο μέσα σε μια έξυπνη κάμερα και η κάμερα έχει κλαπεί, τα δεδομένα εξαφανίζονται το ίδιο. Αν το βίντεο είναι αποθηκευμένο σε ένα ασφαλή κόμβο ομίχλης ο οποίος παρακολουθείται, το βίντεο είναι ασφαλές ακόμη και αν η κάμερα κλαπεί.

Επιπροσθέτως, η υπολογιστική ομίχλη μπορεί να βοηθήσει να διατηρηθούν τα διαπιστευτήρια ασφαλείας και το λογισμικό στις νεότερες εκδόσεις τους σε ένα μεγάλο αριθμό συσκευών στην κλίμακα ενός παγκόσμιου IoT περιβάλλοντος. Απαιτώντας κάθε συσκευή να συνδεθεί με το νέφος για την ενημέρωση των διαπιστευτηρίων και το λογισμικό της, αρκετές φορές την ημέρα, είναι πρακτικά αδύνατο. Αλλά οι κόμβοι ομίχλης έχουν σχεδιαστεί για να είναι μια κατανεμημένη υποδομή για να ελέγχει τα διαπιστευτήρια ασφαλείας σε τεράστιους αριθμούς συσκευών που βασίζονται

σε εφαρμογές ή / και την ιδιοκτησία ταυτόχρονα χωρίς πέφτει η επικοινωνία τους.

Επίσης , η υπολογιστική ομίχλη μπορεί να παρακολουθεί την κατάσταση ασφαλείας των κατανεμημένων συστημάτων με επεκτάσιμο και αξιόπιστο τρόπο . Στον κόσμο του IoT, είναι απαραίτητο να είναι σε θέση να διατυπωθεί, με ένα αξιόπιστο τρόπο, αν ένας μεγάλος αριθμός των κατανεμημένων συσκευών και των συστημάτων λειτουργούν με ασφάλεια και εμπιστοσύνη. Πολλές από τις σημερινές παραβιάσεις έχουν σχεδιαστεί για να στέλνουν μηνύματα κατάστασης που κάνουν τις διεργασίες να φαίνονται φυσιολογικές. Η Ομίχλη έχει την υποδομή για να ανιχνεύσει αυτά τα είδη των επιθέσεων.

Τέλος , το fog computing μπορεί να παρέχει υπηρεσίες αντιμετώπισης περιστατικών σε πραγματικό χρόνο που επιτρέπουν σε συστήματα IoT να ανταποκριθούν σε κινδύνους / παραβιάσεις χωρίς διακοπή των υπηρεσιών τους .Αυτό είναι μια ιδιαίτερα κρίσιμη λειτουργία σε βιομηχανίες όπου τα συστήματα και οι διεργασίες IoT παρέχουν στις επιχειρήσεις μεγάλη παραγωγή εσόδων. Κλείνοντας τις διεργασίες δεν λογίζεται ως πρακτική επιλογή. Εδώ είναι τρία παραδείγματα:

- ❖ Μια γεννήτρια ρεύματος μολύνεται από κακόβουλο λογισμικό. Κλείνοντας τις γεννήτριες ρεύματος μπορεί να προκαλέσει διαταραχές στο δίκτυο και διακοπές ρεύματος. Η Ομίχλη μπορεί να βοηθήσει και να ανταποκριθεί με ασφάλεια, διατηρώντας παράλληλα το σύστημα χωρίς αποσυνδέσεις κατά τις εργασίες συντήρησης.
- ❖ Αποσυνδέσεις των βιομηχανικών συστημάτων ελέγχου (ICS) σε κατασκευαστικές λειτουργίες. Οι κατασκευαστές απαιτούν αδιάλειπτες διαδικασίες, όχι μόνο για τα έσοδα, αλλά και για λόγους ασφαλείας. Η Ομίχλη επιτρέπει την αντιμετώπιση των περιστατικών χωρίς διακοπές.

- ❖ Ένα συνδεδεμένο αυτοκίνητο μολύνεται με κακόβουλο λογισμικό, ενώ βρίσκεται σε κίνηση. Όπως γίνεται αντιληπτό, το κλείσιμο του κινητήρα δεν είναι μια επιλογή, όταν το αυτοκίνητο είναι στην εθνική οδό. Η Ομίχλη επιτρέπει την ανίχνευση του κακόβουλου προγράμματος και την επίλυση προβλημάτων κατά τη μεταφορά.

6.4 ΕΦΑΡΜΟΓΕΣ FOG COMPUTING

Ο ρόλος της υπολογιστικής ομίχλης έχει επεξεργαστεί σχετικά με τα επόμενα έξι σενάρια. Τα πλεονεκτήματα του fog computing πληρούν τις απαιτήσεις των εφαρμογών σε αυτά τα σενάρια.

6.4.1 Υπολογιστική ομίχλη στο Έξυπνο Δίκτυο:

Οι εφαρμογές εξισορρόπησης του φορτίου της ενέργειας μπορεί να τρέξουν σε συσκευές στην άκρη του δικτύου, όπως οι έξυπνοι μετρητές και τα μικρο-δίκτυα. Με βάση την ενεργειακή ζήτηση, τη διαθεσιμότητα και τη χαμηλότερη τιμή, αυτές οι συσκευές έχουν την δυνατότητα να αλλάζουν αυτόματα σε εναλλακτικές πηγές ενέργειας, όπως η ηλιακή και η αιολική.

6.4.2 Υπολογιστική ομίχλη σε έξυπνα φανάρια και συνδεδεμένα οχήματα:

Μια βιντεοκάμερα που ανιχνεύει ένα ασθενοφόρο και τα φώτα που αναβοσβήνουν μπορεί να αλλάξει αυτόματα τα φανάρια του δρόμου για να ανοίξει διαδρόμους για το όχημα και να περάσει ανάμεσα στη κίνηση που υπάρχει εκείνη την ώρα. Τα έξυπνα φανάρια του δρόμου αλληλεπιδρούν τοπικά με αισθητήρες, ανιχνεύουν την παρουσία των πεζών και ποδηλατών, και μετράει την απόσταση και την ταχύτητα προσέγγισης οχημάτων.

6.4.3 Ασύρματος Αισθητήρας και Δίκτυα Ενεργοποίησης:

Τα παραδοσιακά ασύρματα δίκτυα αισθητήρων υπολείπονται σε εφαρμογές που να υπερβαίνουν σε ανίχνευση και εντοπισμό, καθώς απαιτούν ενεργοποιητές να ασκήσουν φυσικές ενέργειες, όπως το άνοιγμα, το κλείσιμο ή ακόμα και να μεταφέρουν αισθητήρες. Σε αυτό το σενάριο, ενεργοποιητές χρησιμεύουν ως συσκευές ομίχλης που μπορούν να ελέγξουν τη διαδικασία μέτρησης από μόνες τους, τη σταθερότητα και τις παλινδρομικές συμπεριφορές με τη δημιουργία ενός συστήματος κλειστού βρόχου. Για παράδειγμα, στο σενάριο της αυτο-διατήρησης των τρένων, η παρακολούθηση του αισθητήρα στο ρουλεμάν της αμαξοστοιχίας μπορεί να ανιχνεύσει επίπεδα θερμότητας, επιτρέποντας στις εφαρμογές να στείλουν μια αυτόματη ειδοποίηση προς τον χειριστή του τρένου για να σταματήσει το τρένο στο επόμενο σταθμό για τη συντήρηση έκτακτης ανάγκης και την αποφυγή πιθανού εκτροχιασμού. Σε σενάρια σωτήριων αεραγωγών, οι αισθητήρες πάνω στους αεραγωγούς παρακολουθούν τις συνθήκες του αέρα που ρέει μέσα και έξω από τα ορυχεία και αυτόματα αλλάζει τη ροή του αέρα αν οι συνθήκες γίνουν επικίνδυνες για τους ανθρακωρύχους.

6.4.4 Αποκεντρωμένος Έλεγχος Έξυπνου Κτιρίου:

Οι εφαρμογές αυτού του σεναρίου διευκολύνονται από ασύρματους αισθητήρες οι οποίοι τοποθετήθηκαν για τη μέτρηση της θερμοκρασίας, της υγρασίας, ή τα επίπεδα των διαφόρων αερίων στην ατμόσφαιρα του κτιρίου. Σε αυτή την περίπτωση, οι πληροφορίες μπορούν να ανταλλάσσονται μεταξύ όλων των αισθητήρων σε έναν όροφο, και οι αναγνώσεις τους μπορούν να συνδυαστούν για να σχηματίσουν αξιόπιστες μετρήσεις. Τα εξαρτήματα του συστήματος μπορεί στη συνέχεια να εργαστούν μαζί για να μειώσουν τη θερμοκρασία, να εισάγουν φρέσκο αέρα ή να ανοίξουν τα παράθυρα. Τα κλιματιστικά μπορεί να αφαιρέσουν την υγρασία από τον αέρα ή την αύξηση της υγρασίας. Οι αισθητήρες μπορούν επίσης να εντοπίζουν και να αντιδρούν

σε κινήσεις (π.χ., ενεργοποιώντας ή απενεργοποιώντας τα φώτα). Κάπως έτσι, συσκευές ομίχλης θα μπορούσαν να ανατεθούν σε κάθε όροφο και να συνεργαστούν σε υψηλότερο επίπεδο ενεργοποίησης. Με τη βοήθεια του fog computing να εφαρμόζεται σε αυτό το σενάριο, έξυπνα κτίρια θα μπορούσαν να διατηρήσουν την κατασκευή τους, τα εξωτερικά και εσωτερικά περιβάλλοντα τους με σκοπό την εξοικονόμηση ενέργειας, νερού και άλλων πόρων.

6.4.5 IoT και Κυβερνο-φυσικά συστήματα (CPS):

Τα συστήματα που βασίζονται στην υπολογιστική ομίχλη γίνονται μια σημαντική κατηγορία του IoT και των CPS. Με βάση τους παραδοσιακούς φορείς πληροφορίας, συμπεριλαμβανομένου του Διαδικτύου και του δικτύου τηλεπικοινωνιών, το IoT είναι ένα δίκτυο που μπορεί να διασυνδέσει συνηθισμένα φυσικά αντικείμενα με διευθύνσεις δικτύου. Τα CPS διαθέτουν ένα σφιχτό συνδυασμό μεταξύ υπολογιστικής και φυσικών στοιχείων του συστήματος. Τα CPS συντονίζουν επίσης την ενσωμάτωση των ηλεκτρονικών υπολογιστών με τη υλική και μηχανική διάσταση των συστημάτων που έχουν επίκεντρο τις πληροφορίες. Το IoT και τα CPS υπόσχονται να μεταμορφώσουν τον κόσμο μας με νέες σχέσεις μεταξύ του ελέγχου που βασίζονται σε υπολογιστές και συστήματα επικοινωνίας, μηχανικής συστημάτων και της φυσικής πραγματικότητας. Η υπολογιστική ομίχλη σε αυτό το σενάριο είναι χτισμένη στην έννοια των ενσωματωμένων συστημάτων στην οποία τα προγράμματα λογισμικού και υπολογιστές είναι ενσωματωμένα σε συσκευές για λόγους που αφορούν όχι μόνο τον υπολογισμό. Παραδείγματα των συσκευών περιλαμβάνουν παιχνίδια, αυτοκίνητα, ιατρικές συσκευές και μηχανήματα. Ο στόχος είναι να ενταχθούν η αφαιρετικότητα και η ακρίβεια του λογισμικού και της δικτύωσης με τη δυναμική, την αβεβαιότητα και τον θόρυβο στο φυσικό περιβάλλον. Χρησιμοποιώντας τις αναδυόμενες γνώσεις, αρχές και μεθόδους των CPS, θα καταστεί δυνατό να αναπτυχθούν

νέες γενιές ευφυών ιατρικών συσκευών και συστημάτων, «έξυπνοι» αυτοκινητόδρομοι, κτίρια, εργοστάσια, αγροτικά και ρομποτικά συστήματα.

6.4.6 Software Defined Networks (SDN):

Το SDN είναι ένα αναδυόμενο σχήμα πληροφορικής και δικτύωσης, και έγινε ένα από τα πιο δημοφιλή θέματα στη βιομηχανία της πληροφορικής. Χωρίζει στρώματα ελέγχου και επικοινωνίας δεδομένων. Ο έλεγχος γίνεται σε κεντρικό server, και οι κόμβοι ακολουθούν διαδρομή επικοινωνίας που αποφασίζεται από τον διακομιστή. Ο κεντρικός διακομιστής μπορεί να χρειαστεί καταναμημένο σχεδιασμό. Η έννοια SDN μελετήθηκε σε WLAN, ασύρματα δίκτυα αισθητήρων, αλλά δεν διαθέτουν ασύρματη επικοινωνία πολλαπλών ανακλάσεων (multihop) και δρομολόγηση πολλαπλών ανακλάσεων. Επιπλέον, δεν υπάρχει καμία επικοινωνία μεταξύ ομότιμων σε αυτό το σενάριο. Η SDN έννοια μαζί με το fog computing θα επιλύσει τα κύρια θέματα στα δίκτυα οχημάτων, την διαλείπουσα συνδεσιμότητα, τις συγκρούσεις και τον υψηλό ρυθμό απώλειας πακέτων, με αύξηση επικοινωνιών από όχημα σε όχημα και από όχημα στην υποδομή επικοινωνιών καθώς και με κεντρικό έλεγχο. Η SDN έννοια για τα δίκτυα οχημάτων προτείνεται για πρώτη φορά.

ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ ΥΛΟΠΟΙΗΣΗ

7. ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτό το κεφάλαιο παρουσιάζονται οι τεχνολογίες υλοποίησης της εφαρμογής που υλοποιείται με σκοπό να αναδείξει τα πλεονεκτήματα του fog computing και να λύσει το πρόβλημα που ορίσαμε στο πρώτο κεφάλαιο σχετικά με το cloud computing και το Internet of things. Έπειτα, αναλύεται η αρχιτεκτονική της εφαρμογής καθώς και η λειτουργικότητα της και τέλος, περιγράφονται μερικά βασικά σημεία κώδικα.

7.1 JAVASCRIPT



Εικόνα 16 : Λογότυπο JavaScript

Η JavaScript [19] είναι μία επίπεδη, δυναμική, τυποποιημένη και ερμηνευτική γλώσσα προγραμματισμού. Έχει τυποποιηθεί βάσει της γλωσσικής προδιαγραφής ECMAScript. Μαζί με την HTML και την CSS, είναι μία από τις τρεις βασικές τεχνολογίες παραγωγής του Παγκόσμιου Ιστού· χρησιμοποιείται από την πλειοψηφία των δικτυακών τόπων και υποστηρίζεται από όλα τα σύγχρονα προγράμματα περιήγησης δίχως τη χρήση πρόσθετων εξαρτημάτων λογισμικού. Η JavaScript είναι πρωτότυπη και βασισμένη σε πρώτης τάξης λειτουργίες, κάτι που την καθιστά ως μια παραδειγματική πολύ-γλώσσα, που υποστηρίζει αντικειμενοστραφή, επιτακτικής ανάγκης και λειτουργικά στυλ προγραμματισμού. Διαθέτει μία Διεπαφή Προγραμματισμού Εφαρμογών (API-Application Programming Interface) για την εργασία με κείμενο, πίνακες, ημερομηνίες και κανονικές εκφράσεις, αλλά δεν περιλαμβάνει καμία Είσοδο/Έξοδο (I/O-Input/Output), όπως είναι η δικτύωση, η αποθήκευση ή οι γραφικές εγκαταστάσεις, διότι βασίζεται σε αυτά που διαθέτει το περιβάλλον υποδοχής στο οποίο είναι ενσωματωμένη.

Παρά κάποιες ονοματικές, συντακτικές και προδιαγραφικές ομοιότητες, η JavaScript και η Java δεν σχετίζονται μεταξύ τους και έχουν πολύ διαφορετικές σημασιολογίες. Το συντακτικό της JavaScript προέρχεται στην πραγματικότητα από την C, ενώ η σημασιολογία και ο σχεδιασμός της επηρεάζονται από τις γλώσσες προγραμματισμού Self και Scheme. Η JavaScript χρησιμοποιείται επίσης σε περιβάλλοντα που δεν βασίζονται στο δίκτυο, όπως τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές και οι μικρές εφαρμογές της επιφάνειας εργασίας. Νεότερες και γρηγορότερες εικονικές μηχανές JavaScript και πλατφόρμες βασισμένες πάνω τους έχουν αυξήσει επίσης τη δημοτικότητα της JavaScript για διαδικτυακές εφαρμογές από την πλευρά του διακομιστή. Από την πλευρά του πελάτη, η JavaScript έχει εφαρμοστεί παραδοσιακά ως μία ερμηνευτική γλώσσα, αλλά πιο πρόσφατα προγράμματα περιήγησης μόλις που προλαβαίνουν να εκτελέσουν

μεταγλώττιση. Χρησιμοποιείται επίσης στην ανάπτυξη παιχνιδιών, στη δημιουργία εφαρμογών επιφάνειας εργασίας και φορητών εφαρμογών και στον προγραμματισμό δικτύων από πλευράς του διακομιστή(server-side) με νεότερες εικονικές μηχανές όπως το Node.js.

7.1.1 Ιστορία

Η JavaScript αναπτύχθηκε αρχικά μέσα σε 10 ημέρες, τον Μάιο του 1995 από τον Brendan Eich, καθώς αυτός εργαζόταν για την Εταιρεία Επικοινωνιών Netscape (Netscape Communications Corporation). Πράγματι, ενώ ανταγωνίζονταν τη Microsoft για την υιοθέτηση της χρήσης διαδικτυακών τεχνολογιών και πλατφόρμων, η Netscape έλαβε υπόψιν της το διακομιστή της από την πλευρά του πελάτη, προσφέροντας ένα καταμεμημένο λειτουργικό σύστημα με μια φορητή έκδοση της Sun Microsystems της Java, παρέχοντας ένα περιβάλλον στο οποίο θα μπορούσαν να «τρέξουν» μικροεφαρμογές. Επειδή η Java ήταν ανταγωνιστής της C++ και στόχο είχε τους επαγγελματίες προγραμματιστές, η Netscape ήθελε επίσης μία ελαφριά ερμηνευτική γλώσσα που θα συμπλήρωνε την Java με την προσφυγή σε μη επαγγελματίες προγραμματιστές, όπως η Visual Basic της Microsoft .

Αν και αναπτύχθηκε με το όνομα Mocha, η γλώσσα επίσημα ονομάστηκε LiveScript όταν διατέθηκε για πρώτη στην αγορά σε δοκιμαστικές εκδόσεις της Netscape Navigator 2.0 τον Σεπτέμβριο του 1995, αλλά μετονομάστηκε σε JavaScript όταν χρησιμοποιήθηκε στην έκδοση 2.0B3 του προγράμματος περιήγησης της Netscape.

Η αλλαγή της επωνυμίας από LiveScript σε JavaScript συνέπεσε στο περίπου με τη Netscape, προσθέτοντας υποστήριξη στη τεχνολογία της Java για το διαδικτυακό πρόγραμμα περιήγησης Navigator της Netscape. Η τελική επιλογή του ονόματος προκάλεσε σύγχυση, δίνοντας την εντύπωση ότι η γλώσσα ήταν ένα παρακλάδι της γλώσσας προγραμματισμού Java και η

επιλογή έχει χαρακτηριστεί ως ένα τέχνασμα προώθησης από τη Netscape.

7.1.2 Τυποποίηση

Το Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλλει την JavaScript στο Διεθνή Οργανισμό Ecma (Ecma International) προκειμένου να εξεταστεί ως ένα βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη έκδοση με το όνομα ECMAScript [14]. Τον Ιούνιο του 1997, ο Διεθνής Οργανισμός Ecma δημοσίευσε την πρώτη έκδοση της προδιαγραφής ECMA-262. Τον Ιούνιο του 1998, έγιναν ορισμένες τροποποιήσεις για να προσαρμοστεί στο πρότυπο ISO/IEC-16262, και δημοσιεύτηκε η δεύτερη έκδοση. Η τρίτη έκδοση της ECMA-262 δημοσιεύτηκε το Δεκέμβριο του 1999.

Η ανάπτυξη της τέταρτης έκδοσης του προτύπου ECMAScript δεν ολοκληρώθηκε ποτέ. Η πέμπτη έκδοση δημοσιεύτηκε το Δεκέμβριο του 2009. Η τρέχουσα έκδοση του προτύπου ECMAScript είναι η έκτη, η οποία κυκλοφόρησε τον Ιούνιο του 2015.

7.1.3 Μετέπειτα εξελίξεις

Η JavaScript έχει γίνει μία από τις δημοφιλέστερες γλώσσες προγραμματισμού στο Παγκόσμιο Ιστό. Η έλευση της τεχνολογίας Ajax επέστρεψε την JavaScript στο προσκήνιο και τράβηξε περισσότερο την προσοχή των επαγγελματιών προγραμματιστών. Το αποτέλεσμα ήταν μία γρήγορη εξάπλωση των συνολικών πλαισίων και βιβλιοθηκών, βελτιωμένες πρακτικές προγραμματισμού της JavaScript και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Παγκόσμιο Ιστό.

Τον Ιανουάριο του 2009, ιδρύθηκε το CommonJS με στόχο τον

καθορισμό ενός κοινού προτύπου βιβλιοθήκης, κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης.

7.1.4 Χαρακτηριστικά

Τα παρακάτω χαρακτηριστικά είναι κοινά σε όλες τις προσαρμοσμένες εφαρμογές της ECMAScript, εκτός εάν ορίζεται ρητά κάτι διαφορετικό.

- **Επιτακτική και δομημένη** Η JavaScript υποστηρίζει ένα μεγάλο μέρος του δομημένου προγραμματισμού σύνταξης της C (π.χ. εντολές if, while loops, εντολές switch, do while loops κ.τ.λ.). Μία μερική εξαίρεση είναι η οριοθέτηση πεδίου: η JavaScript αρχικά είχε μόνο τη λειτουργική οριοθέτηση πεδίου με var. Η ECMAScript 2015 προσθέτει μία λέξη κλειδί let για block scoping, που σημαίνει ότι τώρα η JavaScript έχει τόσο τη λειτουργική οριοθέτηση πεδίου όσο και την οριοθέτηση πεδίου block. Όπως και η C, η JavaScript κάνει διάκριση μεταξύ των εκφράσεων και των εντολών. Μία συντακτική διάφορα σε σχέση με τη C είναι η αυτόματη εισαγωγή ερωτηματικού, η οποία επιτρέπει στα ερωτηματικά, που κανονικά θα τερμάτιζαν οι εντολές, να παραλειφθούν.
- **Δυναμική**
 - ✚ **Πληκτρολόγηση** Όπως στις περισσότερες γλώσσες προγραμματισμού, οι τύποι σχετίζονται με τιμές, όχι με μεταβλητές. Για παράδειγμα, μία μεταβλητή x θα μπορούσε να συνδεθεί με έναν αριθμό και αργότερα να επανασυνδεθεί με μια συμβολοσειρά. Η JavaScript υποστηρίζει διάφορους τρόπους για τον έλεγχο του τύπου ενός αντικειμένου, συμπεριλαμβανομένης και της πληκτρολόγησης duck (duck

typing).

- ✚ **Αντικειμενοστραφής** Η JavaScript είναι σχεδόν εξ' ολοκλήρου αντικειμενοστραφής. Τα αντικείμενα της JavaScript είναι συνειρμικοί πίνακες, εμπλουτισμένοι με πρωτότυπα. Οι ονομασίες των αντικειμένων είναι λέξεις κλειδιά. Αυτές υποστηρίζουν δύο ισοδύναμες συντάξεις: τη σημείωση κουκίδας (dot notation) (`obj.x = 10`) και τη σημείωση παρένθεσης (bracket notation) (`obj['x'] = 10`). Οι ιδιότητες και οι τιμές τους μπορούν να προστεθούν, να αλλαχθούν ή να διαγραφούν κατά τον χρόνο εκτέλεσης. Οι περισσότερες ιδιότητες ενός αντικειμένου μπορούν να απαριθμηθούν χρησιμοποιώντας ένα βρόγχο `for ...in`. Η JavaScript έχει ένα μικρό αριθμό ενσωματωμένων αντικειμένων όπως είναι η `Function` και η `Date`.
- ✚ **Αξιολόγηση χρόνου εκτέλεσης** Η JavaScript περιλαμβάνει μία λειτουργία `eval` η οποία μπορεί να εκτελέσει εντολές που δίνονται ως συμβολοσειρές κατά τον χρόνο εκτέλεσης.

➤ **Αντικειμενοστραφής προγραμματισμός με βάση τα πρωτότυπα**

- ✚ **Πρωτότυπα** Η JavaScript χρησιμοποιεί πρωτότυπα καθώς πολλές άλλες αντικειμενοστραφείς γλώσσες χρησιμοποιούν κλάσεις για κληρονομικότητα. Είναι δυνατό να προσομοιώσουμε πολλά χαρακτηριστικά βασισμένα στις κλάσεις με πρωτότυπα στη JavaScript.
- ✚ **Οι λειτουργίες ως κατασκευαστές αντικειμένων** Οι λειτουργίες διπλασιάζονται ως κατασκευαστές αντικειμένων μαζί με τον τυπικό τους ρόλο. Η πρόσθεση μίας νέας λειτουργίας σε μία λειτουργία `call`, θα δημιουργήσει ένα υπόδειγμα ενός πρωτοτύπου, το οποίο θα κληρονομήσει ιδιότητες και μεθόδους από τον κατασκευαστή (συμπεριλαμβανομένων και ιδιοτήτων από το πρωτότυπο `Object` (Αντικείμενο)). Η ECMAScript 5 προσφέρει τη μέθοδο δημιουργίας `Object`, επιτρέποντας τη ρητή

δημιουργία ενός υποδείγματος χωρίς να κληρονομήσει αυτόματα από το πρωτότυπο Object. Η ιδιότητα του κατασκευαστή του πρωτοτύπου ορίζει το αντικείμενο που θα χρησιμοποιηθεί για το καινούριο εσωτερικό πρωτότυπο του αντικειμένου. Νέες μέθοδοι μπορούν να προστεθούν με την τροποποίηση της λειτουργίας του πρωτοτύπου ως ένας κατασκευαστής. Οι ενσωματωμένοι κατασκευαστές της JavaScript, όπως είναι ο Array(Πίνακας) και το Object, έχουν επίσης πρωτότυπα που μπορούν να τροποποιηθούν. Αν και είναι εφικτό να τροποποιηθεί το πρωτότυπο Object, θεωρείται γενικά μια κακή πρακτική επειδή τα περισσότερα αντικείμενα στη JavaScript θα κληρονομήσουν μεθόδους και ιδιότητες από το πρωτότυπο Object και ίσως να μην αναμένουν την τροποποίηση του πρωτοτύπου.

✚ **Οι συναρτήσεις ως μέθοδοι** Σε αντίθεση με πολλές αντικειμενοστραφείς γλώσσες, δεν υπάρχει διάκριση μεταξύ του ορισμού της συνάρτησης και του ορισμού της μεθόδου. Αντίθετα, η διάκριση γίνεται κατά τη διάρκεια που καλείται μία συνάρτηση· όταν κληθεί μία συνάρτηση ως μία μέθοδος ενός αντικειμένου, η λέξη κλειδί της συνάρτησης συνδέεται με εκείνο το αντικείμενο για αυτή την επίκληση.

➤ **Έμμεση και άμεση ανάθεση** Η JavaScript είναι μία γλώσσα ανάθεσης.

✚ **Οι συναρτήσεις ως Ρόλοι (Traits και Mixins)** Η JavaScript υποστηρίζει εγγενώς διάφορες εφαρμογές με βάση λειτουργίες ως πρότυπα Ρόλων (Role patterns) όπως είναι τα Traits και Mixins. Μία τέτοια συνάρτηση καθορίζει επιπλέον συμπεριφορά με τουλάχιστον μία λειτουργία συνδεδεμένη με τη λέξη κλειδί this μέσα στο σώμα της συνάρτησης. Ένας Role μετά πρέπει να ανατεθεί άμεσα μέσω του call ή να εφαρμοστεί σε αντικείμενα

που χρειάζονται να παρουσιάσουν επιπλέον συμπεριφορά η οποία δεν διανέμεται μέσω της αλυσίδας του πρωτοτύπου.

- ✚ **Σύνθεση Αντικειμένου και Κληρονομικότητα** Ενώ η άμεση ανάθεση με βάση λειτουργίες δεν καλύπτει τη σύνθεση στη JavaScript, η έμμεση ανάθεση ήδη συμβαίνει κάθε φορά που η πρωτότυπη αλυσίδα προχωράει προκειμένου, π.χ., να βρει μία μέθοδο με την οποία ίσως να σχετίζεται αλλά και η οποία μέθοδος δεν θα ανήκει άμεσα σε κάποιο αντικείμενο. Μόλις η μέθοδος βρεθεί, λαμβάνει το όνομά της μέσα στο πλαίσιο του αντικειμένου. Έτσι, η κληρονομικότητα στη JavaScript καλύπτεται από μία αυτόματη ανάθεση η οποία συνδέεται με την ιδιότητα του πρωτοτύπου των κατασκευαστικών λειτουργιών.

➤ Ποικίλης

- ✚ **Περιβάλλον χρόνου εκτέλεσης** Η JavaScript βασίζεται συνήθως σε ένα περιβάλλον χρόνου εκτέλεσης (π.χ., σε ένα πρόγραμμα περιήγησης του Ιστού) προκειμένου να παρέχει αντικείμενα και μεθόδους βάσει των οποίων σενάρια μπορούν να αλληλοεπιδρούν με το περιβάλλον (π.χ., μία ιστοσελίδα DOM). Βασίζεται επίσης στο περιβάλλον χρόνου εκτέλεσης προκειμένου να παρέχει την ικανότητα να συμπεριληφθούν εισαγωγικά σενάρια (`import scripts`) (π.χ., HTML `<script>` στοιχεία). Αυτό δεν είναι ένα καθαυτό χαρακτηριστικό της γλώσσας, αλλά είναι συνηθισμένο στις περισσότερες εφαρμογές της JavaScript. Η JavaScript επεξεργάζεται ένα-ένα τα μηνύματα από μία σειρά. Κατά τη φόρτωση ενός μηνύματος, η JavaScript καλεί μία λειτουργία που είναι συνδεδεμένη με αυτό το μήνυμα, η οποία δημιουργεί ένα πλαίσιο στοίβας κλήσεων (`call stack frame`) (τα `arguments` των συναρτήσεων και τις τοπικές μεταβλητές). Η στοίβα των κλήσεων μειώνεται και αυξάνεται βάσει των αναγκών των συναρτήσεων.

Μετά την ολοκλήρωση της λειτουργίας, όταν η στοίβα είναι άδεια, η JavaScript προχωράει στο επόμενο μήνυμα της σειράς. Αυτό ονομάζεται `event loop`, και περιγράφεται ως «`run to completion`» επειδή κάθε μήνυμα δέχεται πλήρη επεξεργασία πριν το επόμενο μήνυμα ληφθεί υπόψη. Ωστόσο, το μοντέλο συγχρονισμού της γλώσσας περιγράφει το `event loop` ως `non-blocking`: η είσοδος/έξοδος (`input/output`) του προγράμματος γίνεται με τη χρήση των συναρτήσεων `event` και `callback`. Αυτό σημαίνει, για παράδειγμα, ότι η JavaScript μπορεί να επεξεργαστεί ένα κλικ του ποντικιού καθώς περιμένει να της επιστρέψει πληροφορίες ένα ερώτημα βάσης δεδομένων.

✚ **Variadic συναρτήσεις** Ένας απροσδιόριστος αριθμός παραμέτρων μπορεί να μεταβιβαστεί σε μία λειτουργία. Η λειτουργία μπορεί να αποκτήσει πρόσβαση σε αυτές μέσω επίσημων παραμέτρων και μέσω, επίσης, `local arguments` αντικειμένων. Οι `Variadic` συναρτήσεις μπορούν ακόμα να δημιουργηθούν μέσω της χρήσης της μεθόδου δέσμευσης (`bind method`).

✚ **Πίνακες και αντικείμενα** Όπως σε πολλές γλώσσες υλοποίησης σεναρίων, οι πίνακες και τα αντικείμενα (συνειρμικοί πίνακες σε άλλες γλώσσες) μπορούν να δημιουργηθούν με μία σαφή σύντομη σύνταξη. Στην πραγματικότητα, αυτοί οι τύποι αποτελούν τη βάση των δεδομένων διαμόρφωσης της JSON.

✚ **Κανονικές εκφράσεις (Regular expressions)** Η JavaScript υποστηρίζει επίσης κανονικές εκφράσεις με τρόπο παρόμοιο όπως η Perl, ο οποίος παρέχει ένα συνοπτικό και ισχυρό συντακτικό για

χειραγώγηση κειμένου, το οποίο είναι πιο εκλεπτυσμένο από τις λειτουργίες των ενσωματωμένων συμβολοσειρών.

7.2 NODEJS



Εικόνα 17 : Λογότυπο NodeJS

Το Node.js [21] είναι ένα ανοιχτού κώδικα, διασταύρωσης πλατφόρμων, περιβάλλον χρόνου εκτέλεσης για τον προγραμματισμό εφαρμογών από την πλευρά του διακομιστή και είναι χτισμένη σε περιβάλλον Javascript.

Το Node.js παρέχει μία αρχιτεκτονική βάσει συμβάντων και μία non-blocking I/O API σχεδιασμένη να βελτιστοποιεί την απόδοση και κλιμάκωση μιας εφαρμογής. Σε αντίθεση με τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων μία διεργασία node στηρίζεται σε ένα μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου και περιέχει μία ενσωματωμένη βιβλιοθήκη για να επιτρέπει στις εφαρμογές να λειτουργούν ως ένας αυτόνομος διαδικτυακός διακομιστής.

Κάποιοι από τους χρήστες του λογισμικού NodeJS είναι το LinkedIn , Microsoft, Netflix, Paypal και Yahoo!.

7.2.1 Ιστορία

Το Node.js εφευρέθηκε το 2009 από τον Ryan Dahl και άλλους προγραμματιστές που εργάζονταν στη Joyent. Το Node.js δημιουργήθηκε και εκδόθηκε πρώτη φορά για χρήση των Linux το 2009. Η ανάπτυξη και διατήρησή του καθοδηγήθηκαν από τον Dahl και χρηματοδοτήθηκαν από την Joyent, την εταιρεία που εργαζόταν ο Dahl. Ο Dahl εμπνεύστηκε τη δημιουργία του node από την ανάγκη του να ενημερώσει τον χρήστη σε πραγματικό χρόνο για την κατάσταση ενός αρχείου που ανεβάζει στο διαδίκτυο. Μετά από αρκετές αποτυχημένες προσπάθειες σε C, Lua και Haskell, η κυκλοφορία της V8 Javascript Engine της Google τον παρέτρεψε να ασχοληθεί με την Javascript.

Το 2011, ένας διαχειριστής πακέτων εισήχθη στη βιβλιοθήκη Node.js, με το όνομα npm. Ο διαχειριστής πακέτων επιτρέπει τη δημοσίευση και τον κοινοπρασιακό των ανοιχτού κώδικα βιβλιοθηκών Node.js από την κοινότητα, και είναι σχεδιασμένος να απλοποιεί την εγκατάσταση, αναβάθμιση και απεγκατάσταση των βιβλιοθηκών. Το πρώτο Node.js που κατασκευάστηκε για να υποστηρίξει τα Windows κυκλοφόρησε τον Ιούλιο του 2011.

7.2.2 Επισκόπηση

Το Node.js επιτρέπει τη δημιουργία διαδικτυακών διακομιστών και εργαλείων δικτύωσης που χρησιμοποιούν την JavaScript και μία συλλογή από modules που χειρίζονται διάφορες βασικές λειτουργίες. Τα modules χειρίζονται το σύστημα αρχείων εισόδου/εξόδου, τη δικτύωση (HTTP, TCP, UDP, DNS, ή TLS/SSL), δυαδικά δεδομένα (ενδιάμεσες μνήμες), λειτουργίες κρυπτογράφησης, ροές δεδομένων και άλλες βασικές λειτουργίες. Τα modules του Node χρησιμοποιούν ένα API που είναι σχεδιασμένο να μειώνει την πολυπλοκότητα της συγγραφής εφαρμογών διακομιστών. Τα πλαίσια μπορούν να χρησιμοποιηθούν για να επιταχύνουν την ανάπτυξη εφαρμογών και κοινών πλαισίων όπως τα Express.js, Socket.IO και Connect.

Το Node.js αρχικά χρησιμοποιήθηκε για την κατασκευή προγραμμάτων δικτύου, όπως διαδικτυακών διακομιστών, κάτι που το καθιστούσε παρόμοιο με την PHP και την Python. Η μεγαλύτερη διαφορά μεταξύ της PHP και του Node.js είναι ότι η PHP είναι μια blocking γλώσσα, όπου οι εντολές εκτελούνται αφού πρώτα έχει ολοκληρωθεί η προηγούμενη εντολή, ενώ το Node.js είναι μια non-blocking γλώσσα όπου οι εντολές εκτελούνται παράλληλα, και χρησιμοποιεί ανακλήσεις για να σηματοδοτήσει κάποια ολοκλήρωση.

7.2.3 Τεχνικές Λεπτομέρειες

Το NodeJS [20] είναι ένα ένα Javascript περιβάλλον χρόνου εκτέλεσης που επεξεργάζεται τις εισερχόμενες αιτήσεις σε έναν βρόγχο, ο οποίος ονομάζεται βρόγχος συμβάντων(event loop).

☞ Threading

Το Node.js λειτουργεί σε ένα μονό νήμα (thread), χρησιμοποιώντας non-blocking κλήσεις εισόδου/εξόδου, κάτι που του επιτρέπει να υποστηρίζει δεκάδες χιλιάδες ταυτόχρονες συνδέσεις χωρίς να αναλαμβάνει το βάρος μεταγωγής του κόστους του περιεχομένου του νήματος. Ο σχεδιασμός του διαμοιρασμού ενός μονού νήματος μεταξύ όλων των αιτημάτων προορίζεται για την κατασκευή εξαιρετικά ταυτόχρονων εφαρμογών, όπου οποιαδήποτε λειτουργία που εκτελεί είσοδο/έξοδο πρέπει να χρησιμοποιεί μία ανάκληση. Για να φιλοξενηθεί ο βρόγχος συμβάντων μονού σήματος, το Node.js χρησιμοποιεί την βιβλιοθήκη libuv η οποία με την σειρά της χρησιμοποιεί μία σταθερών διαστάσεων πισίνα (pool) νήματος που είναι υπεύθυνη για όλες τις non-blocking ασύγχρονες εισόδου/εξόδου λειτουργίες.

☰ V8

Η V8 είναι η μηχανή εκτέλεσης της JavaScript που είναι κατασκευασμένη για το Google Chrome και ανοιχτού κώδικα από την Google το 2008. Γραμμένη σε C++, η V8 μεταγλωττίζει τον πηγαίο κώδικα JavaScript σε εγγενή κώδικα μηχανής αντί να τον ερμηνεύσει σε πραγματικό χρόνο.

Το Node.js περιέχει libuv για να χειρίζεται ασύγχρονα συμβάντα. Το libuv είναι επίπεδο αφαίρεσης για δικτυακή και συστημική λειτουργικότητα αρχείων, τόσο στα Windows όσο και στα συστήματα βασισμένα σε POSIX όπως τα Linux, Mac OS X, OSS στο NonStop και Unix.

Η κύρια λειτουργικότητα του Node.js βρίσκεται σε μία JavaScript βιβλιοθήκη. Οι συνδέσεις Node.js γραμμένες σε C++, συνδέουν αυτές τις τεχνολογίες μεταξύ τους και με το λειτουργικό σύστημα.

☰ Διαχείριση πακέτων

Το npm είναι ο προ-εγκατεστημένος διαχειριστής πακέτων για την πλατφόρμα διακομιστή Node.js. Χρησιμοποιείται για να εγκαθιστά Node.js προγράμματα από το μητρώο npm, οργανώνοντας την εγκατάσταση και τη διαχείριση των τρίτων προγραμμάτων Node.js. Το npm δεν πρέπει να συγχέεται με την δήλωση CommonJS require(). Δεν χρησιμοποιείται για την φόρτωση κώδικα: αντ' αυτού, χρησιμοποιείται για την εγκατάσταση κώδικα και τον χειρισμό εξαρτήσεων του κώδικα από τη γραμμή εντολών. Τα πακέτα που βρίσκονται στο μητρώο του npm μπορούν να κυμαίνονται από απλές βοηθητικές βιβλιοθήκες, όπως η Underscore.js, μέχρι λειτουργείς έργων όπως το Grant.

☰ Unified API

Το Node.js μπορεί να συνδυαστεί με ένα πρόγραμμα περιήγησης, μία

βάση δεδομένων (όπως η MongoDB ή η CouchDB) και μία JSON για μία ενοποιημένη προγραμματιστική JavaScript στοίβα (stack). Με την προσαρμογή του για το τι ήταν ουσιαστικά τα προγραμματιστικά μοτίβα από την πλευρά του διακομιστή όπως τα MVC, MVP, MVVM κ.τ.λ., το Node.js επιτρέπει την επαναχρησιμοποίηση του ίδιου μοντέλου και της ίδιας υπηρεσιακής διεπαφής μεταξύ της πλευράς του πελάτη και της πλευράς του διακομιστή.

☞ Βρόγχος Συμβάντων

Το Node.js καταγράφει τον εαυτό του στο λειτουργικό σύστημα έτσι ώστε να ειδοποιείται όταν γίνεται μία σύνδεση, και το λειτουργικό σύστημα θα εκδώσει ένα callback, το οποίο είναι ένα από τα κύρια χαρακτηριστικά του node. Μέσα στον χρόνο εκτέλεσης του Node.js κάθε σύνδεση είναι μία μικρή κατανομή στοίβας, χρησιμοποιώντας έναν βρόγχο συμβάντων για επεκτασιμότητα. Σε αντίθεση με άλλους διακομιστές βασισμένους σε συμβάντα, ο βρόγχος συμβάντων του Node.js δεν χρειάζεται να καλείται ρητά. Αντ' αυτού ορίζονται οι ανακλήσεις, και ο διακομιστής εισέρχεται αυτόματα στον βρόγχο στο τέλος ορισμού της ανάκλησης. Το Node.js εξέρχεται του βρόγχου συμβάντων όταν δεν υπάρχουν περαιτέρω ανακλήσεις που πρέπει να εκτελεστούν.

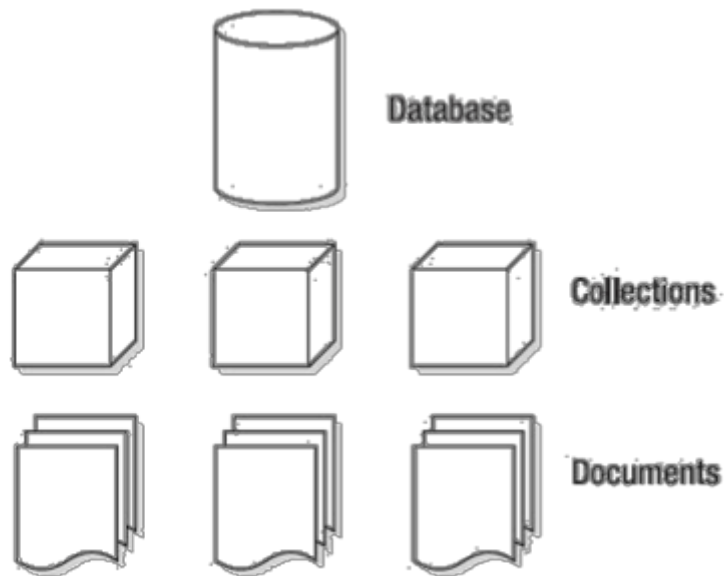
7.3 MongoDB

Η ονομασία της βάσης δεδομένων MongoDB [22] προήλθε από τη λέξη humongous που στα ελληνικά σημαίνει τεράστιος και είναι μια σχετικά νέα βάση δεδομένων, που δεν έχει την έννοια των πινάκων, των σχημάτων, της

γλώσσας SQL και των συσχετίσεων (joins). Με λίγα λόγια, είναι μια πολύ διαφορετική βάση από τις συνηθισμένες σχεσιακές βάσεις δεδομένων.

Η ομάδα που σχεδίασε τη βάση αυτή, αποφάσισε να μη δημιουργήσει άλλη μια βάση δεδομένων, που θα έκανε τα πάντα. Αντίθετα ήθελε να δημιουργήσει μια βάση που θα έχει να κάνει με έγγραφα (documents), και όχι με γραμμές (rows), και θα ήταν πολύ γρήγορη, επεκτάσιμη, και εύκολη στη χρήση. Για να γίνει αυτό, η ομάδα έπρεπε να εγκαταλείψει ορισμένες λειτουργίες, πράγμα που σημαίνει ότι η MongoDB δεν είναι μια ιδανική υποψήφια βάση για όλες τις καταστάσεις. Για παράδειγμα, η απουσία της υποστήριξης συναλλαγών (transactions) σημαίνει ότι δεν θα χρησιμοποιήσουμε τη βάση MongoDB για λογιστικές εφαρμογές. Όμως η MongoDB, είναι πολύ καλή στην αποθήκευση σύνθετων δεδομένων. Η MongoDB, παρέχει μια πλούσια, προσανατολισμένη σε έγγραφα βάση, η οποία είναι σχεδιασμένη για να προσφέρει ταχύτητα και επεκτασιμότητα. Μπορεί να εκτελεστεί σχεδόν σε οποιοδήποτε λειτουργικό σύστημα χρειαστεί, όπως Windows, Linux και Mac.

Η εγκατάσταση της MongoDB φιλοξενεί έναν αριθμό βάσεων δεδομένων. Όπως φαίνεται και στην παρακάτω εικόνα, κάθε βάση δεδομένων από αυτές περιέχει ένα σύνολο από συλλογές (collections). Κάθε συλλογή περιέχει ένα σύνολο από έγγραφα. Τέλος, το κάθε έγγραφο είναι ένα σύνολο από ζεύγη κλειδιού-τιμής.



Εικόνα 18 : Η δομή της MongoDB

Τα έγγραφα έχουν δυναμικό σχήμα (dynamic schema). Αυτό σημαίνει ότι τα έγγραφα της ίδιας συλλογής δε χρειάζεται να έχουν το ίδιο σύνολο πεδίων ή την ίδια δομή, ενώ ακόμη, τα κοινά πεδία μίας συλλογής εγγράφων μπορούν να περιέχουν διαφορετικούς τύπους δεδομένων.

```

{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}

```

← field: value
← field: value
← field: value
← field: value

Εικόνα 19 : Παράδειγμα της δομής ενός εγγράφου στην MongoDB

Η MongoDB [23] αποθηκεύει τα έγγραφα στο δίσκο σε φόρμα σειριοποίησης BSON. Το BSON είναι μία δυαδική αναπαράσταση των JSON εγγράφων, αλλά περιέχει πολλούς περισσότερους τύπους δεδομένων από το JSON. Το κέλυφος JavaScript mongo και οι οδηγοί γλώσσας MongoDB

(MongoDB language drivers) κάνουν τη μετάφραση μεταξύ BSON και της αναπαράστασης των εγγράφων με βάση τη γλώσσα προγραμματισμού. Στη δική μας περίπτωση η γλώσσα προγραμματισμού είναι η Java.

Η τιμή ενός πεδίου μπορεί να είναι οποιοδήποτε τύπου δεδομένων BSON, συμπεριλαμβανομένων άλλων εγγράφων, πινάκων και πινάκων από έγγραφα. Για παράδειγμα, το έγγραφο που φαίνεται στην εικόνα περιέχει τιμές από ποικίλους τύπους.

```
var mydoc = {
  _id: ObjectId("5099803df3f4948bd2f98391"),
  name: { first: "Alan", last: "Turing" },
  birth: new Date('Jun 23, 1912'),
  death: new Date('Jun 07, 1954'),
  contribs: [ "Turing machine", "Turing test", "Turingery" ],
  views : NumberLong(1250000)
}
```

Εικόνα 20 : Έγγραφο με ποικίλους τύπους δεδομένων

Εδώ βλέπουμε πως:

- Το πεδίο `_id` περιέχει `ObjectId`
- Το πεδίο `name` περιέχει ένα ενσωματωμένο έγγραφο που περιέχει τα πεδία `first` και `last`

- Τα πεδία birth και death περιέχουν τιμές τύπου Date
- Το contribs περιέχει έναν πίνακα από συμβολοσειρές (array of strings)
- Το views περιέχει μία τιμή τύπου NumberLong

Όσον αφορά το πεδίο `_id`, εξ' ορισμού η MongoDB δημιουργεί ένα μοναδικό ευρετήριο γι' αυτό κατά τη δημιουργία της συλλογής. Επίσης είναι πάντα το πρώτο πεδίο στα έγγραφα και μπορεί να περιέχει κάθε τύπο BSON πλην των πινάκων.

Ο χρήστης της MongoDB χρησιμοποιεί την τελεία για να έχει πρόσβαση στα στοιχεία ενός πίνακα και στα πεδία ενός ενσωματωμένου εγγράφου. Δηλαδή θα πρέπει να χρησιμοποιήσει κάτι τέτοιο:

'<array>.<index>' ή '<embedded document>.<field>'

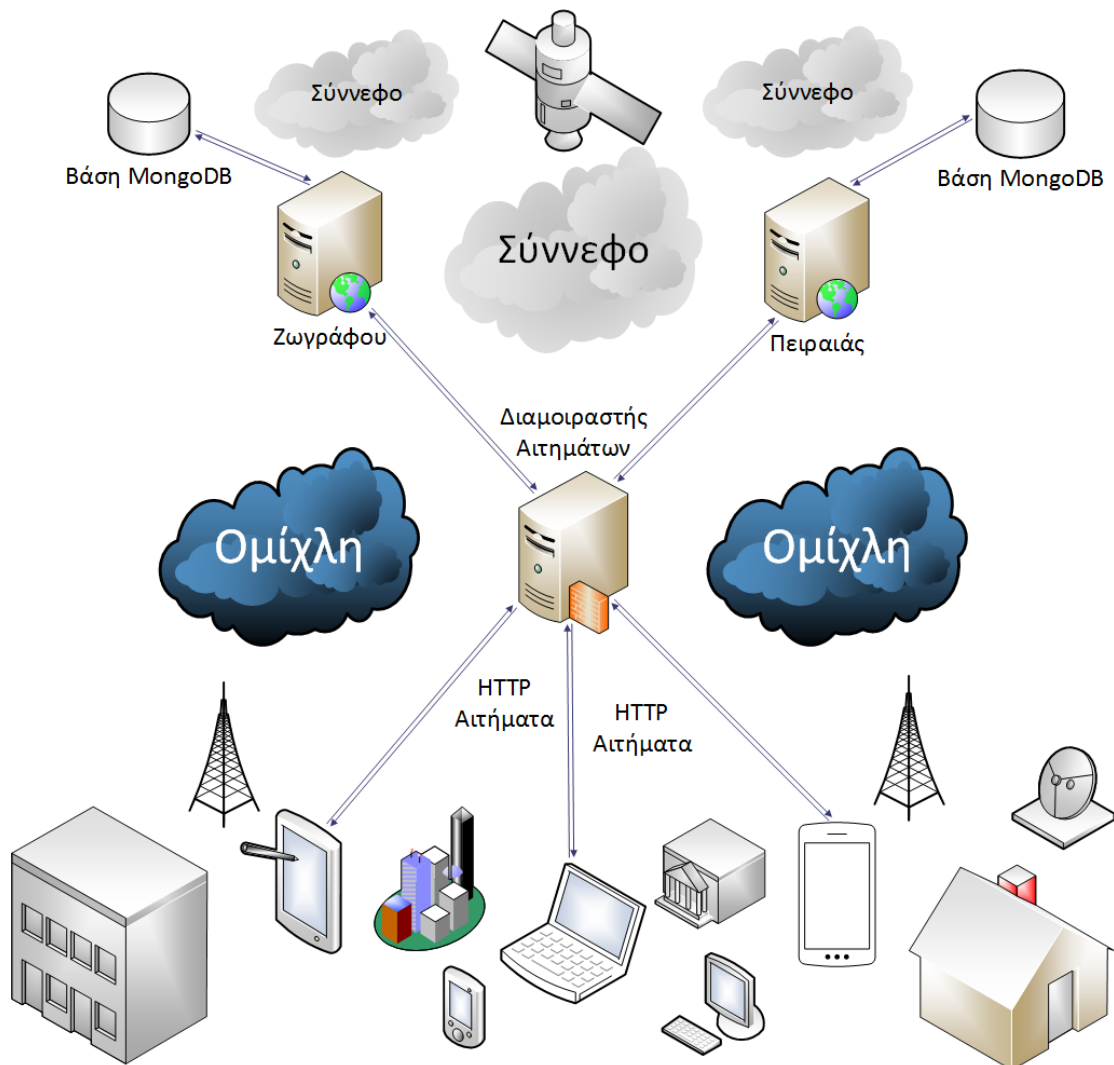
για πίνακα και για ενσωματωμένο έγγραφο, αντίστοιχα. Συγκεκριμένα, στο παρακάτω παράδειγμα, για να έχουμε πρόσβαση στο πεδίο phone που περιέχει το τηλέφωνο του χρήστη '123xyz', θα γράφαμε `contact.phone`



Εικόνα 21 : Έγγραφο με εμφολευμένο έγγραφο

7.4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΛΟΠΟΙΗΣΗΣ

Η παρακάτω αρχιτεκτονική χρησιμοποιείται στην εφαρμογή που υλοποιήθηκε με θέμα τις “Δημοτικές Εκλογές 2014” και τα αποτελέσματά τους τα οποία ανευρέθηκαν στην ιστοσελίδα του Υπουργείου Εσωτερικών[1]. Σκοπός της δημιουργίας της είναι να αναδείξει την εισαγωγή της υπολογιστικής ομίχλης ως ενδιάμεσο του cloud computing με τον τεράστιο αριθμό συσκευών



Εικόνα 22 : Αρχιτεκτονική Συστήματος Προτεινόμενης Λύσης

που συνδέονται στο διαδίκτυο και αποκαλείται το Ίντερνετ των πραγμάτων (IoT).

Η εν λόγω εφαρμογή δομείται από 3 τεχνολογίες από τις οποίες η ραχοκοκκαλιά της είναι η γλώσσα Java που περιγράψαμε παραπάνω και η βάση δεδομένων της υλοποιείται με MongoDB. Τα αποτελέσματα των εκλογών που πάρθηκαν από την επίσημη ιστοσελίδα του υπουργείου διαμορφώθηκαν κατάλληλα σε έγγραφα και εισάχθηκαν στην βάση. Στην συνέχεια, έπρεπε να δημιουργηθεί μια αρχιτεκτονική ώστε να ακολουθεί τα πρότυπα της αρχιτεκτονικής της ομίχλης με αποτέλεσμα να εκμεταλλευτούμε την γεωλογική κατανομή της τεχνολογίας και να δημιουργήσουμε δύο server όπου ο καθένας θα στέλνει και θα απαντάει στα ερωτήματα που του κάνουμε ως clients ανάλογα όμως με την περιοχή που βρίσκεται ο χρήστης. Με αυτό το τρόπο επιτυγχάνουμε να τοποθετήσουμε την επεξεργασία και την όλη αρχιτεκτονική πιο κοντά στις συσκευές και στον τελικό χρήστη-client από το να στέλνουμε τα ερωτήματα σε κέντρα δεδομένων απομακρυσμένα κατά πολύ από το χρήστη και αποφεύγουμε την μεγάλη διαδρομή που θα είχαν να ακολουθήσουν τα πακέτα δεδομένων μέσω ατελείωτων κόμβων κάτι που θα δημιουργούσε κινδύνους εμπιστευτικότητας και ακεραιότητας τους.

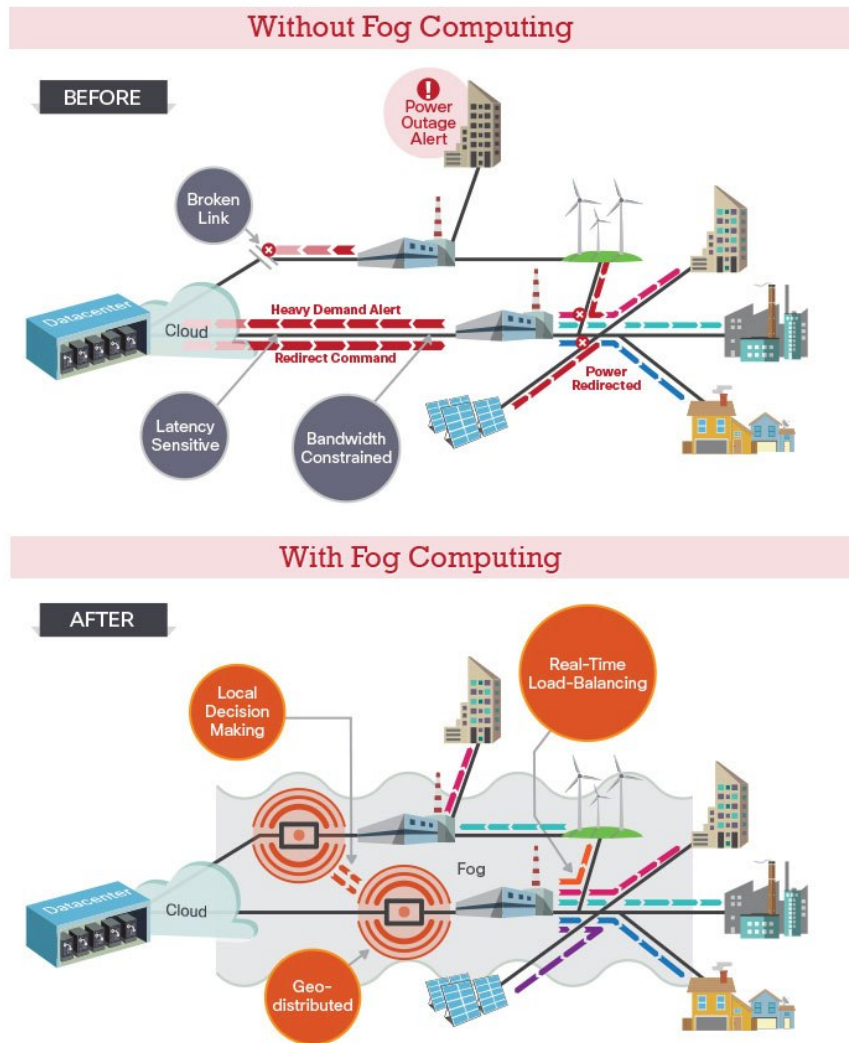
Ενδιάμεσα στους διακομιστές, τοποθετείται ένας load balancer ο οποίος θα διενεργεί σε πραγματικό χρόνο την προ-επεξεργασία των αιτημάτων http που θα αποστέλουν οι clients και θα στέλνει τα αιτήματα στον κατάλληλο διακομιστή ανάλογα με την τοποθεσία του χρήστη για να λάβει τα κατάλληλα αποτελέσματα των εκλογών της περιοχής του και του δήμου του οποίου βρίσκεται.

Οι Clients ή αλλιώς οι χρήστες θα έχουν στην διαθεσή τους μια εφαρμογή γραμμένη σε Javascript η οποία αυτόματα θα ελέγχει το εύρος ζώνης, την ταχύτητα του δικτύου και την τοποθεσία τους και θα αποστέλλει τις κατάλληλες πληροφορίες στον load balancer για να συνεχίσει με τη σειρά του την προ-επεξεργασία των πληροφοριών.

Στην συνέχεια , εφόσον ο ενδιάμεσος load balancer επεξεργαστεί τα αιτήματα και τα στείλει στον κατάλληλο διακομιστή της περιοχής όπου βρίσκεται ο χρήστης , τότε διενεργείται σε πραγματικό χρόνο η κεντρική επεξεργασία της αρχιτεκτονικής που δομήθηκε. Ο διακομιστής αυτός επεξεργάζεται τις πληροφορίες του αιτήματος , επικοινωνεί με την βάση και εκτελεί ερωτήματα για να λάβει τα αποτελέσματα των εκλογων της περιοχής που ζητείται με τη βοήθεια της NodeJS γλώσσας που υλοποιεί το server side κομμάτι του συστήματος. Μόλις λάβει τα κατάλληλα αποτελέσματα εκλογών τότε ελέγχει μια δεύτερη παράμετρο πριν στείλει απάντηση πίσω στον balancer , τον αριθμό των συνδέσεων που υπάρχουν εκείνη την ώρα στον διακομιστή κάτι που επηρεάζει την τελική απάντηση για το αν θα στείλει απλά τα δεδομένα σε text μορφή ή και ένα γράφημα των αποτελεσμάτων των εκλογών μαζί. Η τρίτη και τελευταία παράμετρος που λαμβάνεται υπόψη στην απάντηση του διακομιστή είναι η ταχύτητα σύνδεσης η οποία έχει υπολογιστεί ήδη μεταξύ του client και του διακομιστή και αποφασίζει το αν θα υπάρχει το κατάλληλο bandwidth για να σταλεί και το γράφημα στον χρήστη ή απλά τα αποτελέσματα σε text μορφή.

Ουσιαστικά , όλη αυτή η επεξεργασία που πραγματοποιείται στον διακομιστή προσφέρει αποφάσεις σε τοπικό επίπεδο, σε πραγματικό χρόνο και απορροφά μεγάλο φόρτο αιτημάτων τα οποία θα έφταναν σε ένα κέντρο δεδομένων κάπου μακριά σε επίπεδο σύννεφου. Αυτό αποτρέπει την ασφυξία που θα προκαλούσε ο μεγάλος φόρτος και την καθυστέρηση που θα υπήρχε λόγω του αυξανόμενου αριθμού των συσκευών. Παράλληλα , η επιπλέον επεξεργασία παραμέτρων προσφέρει μεγαλύτερο έλεγχο δεδομένων των οποίων αποστέλλονται μέσα στο δίκτυο ανάλογα με τις υπάρχουσες συνθήκες ταχύτητας και συνδέσεων αλλά και πιο φιλτραρισμένων αποτελεσμάτων με αποτέλεσμα την αποφυγή του γνωστού jitter.

Στο παρακάτω σχήμα 23, παρατηρούμε όλες τις διαφορές που υπάρχουν στην περίπτωση που δεν χρησιμοποιούμε fog computing αρχιτεκτονική και στην άλλη περίπτωση στην οποία δομείται με αυτήν.



Εικόνα 23 : Διαφορές Αρχιτεκτονικής Cloud με Fog Computing

7.5 ΠΛΗΡΟΦΟΡΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτό το σημείο θα περιγράψουμε τον κώδικα της εφαρμογής που αναπτύχθηκε για να καταλάβουμε και να αναλύσουμε τις παραμέτρους που χρησιμοποιήθηκαν με σκοπό να επιτευχθεί η επεξεργασία στους διακομιστές που δημιουργήσαμε.

Αρχικά, το πρώτο βήμα είναι να κατασκευάσουμε το διακομιστή ο οποίος θα δέχεται τα αιτήματα και θα αποκρίνεται στους πελάτες. Στην παρακάτω εικόνα, παρατηρούμε την δήλωση πολλών βιβλιοθηκών οι οποίες χρειάζονται για να λειτουργήσει σωστά ο διακομιστής και η εντολή `http.createServer(app).listen(5252)` είναι ουσιαστικά και η σημαντικότερη όπου δημιουργεί έναν `http server` ο οποίος ακούει τα αιτήματα που αποστέλλονται στην πόρτα 5252. Η επόμενη σημαντική εντολή είναι ο ορισμός της διεύθυνσης της `mongodb (url1)` της βάσης δεδομένων που περιέχει αποθηκευμένα σε έγγραφα τα αποτελέσματα των εκλογών της περιοχής. Οι τελευταίες εντολές είναι χρήσιμες για να έχει την δυνατότητα ο διακομιστής να αναγνωρίζει και να υποστηρίζει τα δεδομένα σε `json` μορφή αλλά και τα `urls`.

```
var http = require('http');
var fs = require('fs');
var url = require('url');
var express = require("express");
var mongoose = require("mongoose");
var MongoClient = require('mongodb').MongoClient, format = require('util').format;
var path = require("path");
var app = express();
var bodyParser = require('body-parser');
var os = require("os");
var plotly = require('plotly')("geo7geg", "tqoodU7A1pKEEE0sJffU");
var mysql = require('mysql');

var percentRam = 0;
var percentCpu = 0;
var imagesize = 5000;
// Http & Https Server
var server = http.createServer(app).listen(5252);

console.log('Server running at http://localhost:5252/');

// Connection URL
var url1 = 'mongodb://Georges-MacBook-Pro.local:27017/test';

// JSON & URL
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
```

Η πρώτη παράμετρος που έπρεπε να υπολογιστεί με σκοπό την επεξεργασία των αιτημάτων που αποστέλλονται στον διακομιστή, είναι να βρεθούν οι ανοιχτές συνδέσεις του διακομιστή με άλλους πελάτες οι οποίοι βρίσκονται κοντά στην ίδια περιοχή και στέλνουν αιτήματα για να λάβουν τα αποτελέσματα των εκλογών της περιοχής τους.

```
console.log("Connected correctly to server");
server.getConnections(function(error, count) {
    server.maxConnections = 2;
    console.log("Connection number : " + count);
    count_connection = count;
});
```

Η πρώτη γραμμή απλά είναι μια εντολή εμφάνισης ότι ο πελάτης συνδέθηκε με επιτυχία με τον διακομιστή και στην συνέχεια από τη δεύτερη γραμμή και έπειτα η συνάρτηση `getConnections` μας επιστρέφει τον αριθμό των ανοιχτών συνδέσεων εκείνη την ώρα και την εμφανίζουμε στην κονσόλα. Η συνάρτηση αυτή δίνεται από την βιβλιοθήκη της `http` και έχουμε το δικαίωμα να της ορίσουμε και τον μέγιστο αριθμό ανοιχτών συνδέσεων που μπορεί να δεχτεί ο διακομιστής που στην προκειμένη περίπτωση για λόγους δοκιμών έχει οριστεί μόνο σε δύο ως το άνω όριο.

Στην συνέχεια , η δεύτερη παράμετρος που λαμβάνεται υπόψη στην απόκριση του διακομιστή είναι οι αποδόσεις του υπολογιστή που λειτουργεί ως `server` και συγκεκριμένα οι αποδόσεις του επεξεργαστή και της μνήμης RAM. Με την συνάρτηση `cpuAverage` υπολογίζουμε το ποσοστό που χρησιμοποιεί ο επεξεργαστής κάθε 2 δευτερόλεπτα καθώς μετράμε το σύνολο του χρόνου που μένει `idle` και το σύνολο του χρόνου των `ticks` από όλους τους πυρήνες του και με τη βοήθεια πράξεων βρίσκουμε το τελικό ποσοστό χρήσης του επεξεργαστή εκείνη την στιγμή στον διακομιστή. Στην επόμενη φάση , στην συνάρτηση `setInterval` δηλώνουμε ότι κάθε δύο δευτερόλεπτα θα

υπολογίζει και θα εμφανίζει τα ποσοστά της χρήσης επεξεργαστή και μνήμης RAM. Μέσα στην συνάρτηση αυτή μετρίεται σε ποσοστό η ποσότητα της ελεύθερης μνήμης που υπάρχει εκείνη τη στιγμή. Με την `os.totalmem()` που

```
function cpuAverage() {
    //Initialise sum of idle and time of cores and fetch CPU info
    var totalIdle = 0, totalTick = 0;
    var cpus = os.cpus();

    //Loop through CPU cores
    for(var i = 0, len = cpus.length; i < len; i++) {

        //Select CPU core
        var cpu = cpus[i];

        //Total up the time in the cores tick
        for(type in cpu.times) {
            totalTick += cpu.times[type];
        }

        //Total up the idle time of the core
        totalIdle += cpu.times.idle;
    }

    //Return the average Idle and Tick times
    return {idle: totalIdle / cpus.length, total: totalTick / cpus.length};
}

//Grab first CPU Measure
var startMeasure = cpuAverage();

setInterval(function() {

    //Grab second Measure
    var endMeasure = cpuAverage();

    //Calculate the difference in idle and total time between the measures
    var idleDifference = endMeasure.idle - startMeasure.idle;
    var totalDifference = endMeasure.total - startMeasure.total;

    //Calculate the average percentage CPU usage
    var percentageCPU = 100 - ~~(100 * idleDifference / totalDifference);

    startMeasure = cpuAverage();
    //Output result to console
    //console.log(percentageCPU + "% CPU Usage.");

    var usedRAM = os.totalmem()-os.freemem();
    var percentageRAM = 100 - ~~(((usedRAM / 1000000) / (os.totalmem() / 1000000)) * 100);
    //console.log(os.freemem());
    console.log(percentageCPU + "% CPU Usage ---- " + percentageRAM + "% RAM Usage ");
    percentRam = percentageRAM;
    percentCpu = percentageCPU;
}, 2000);
```

αντιπροσωπεύει το μέγεθος όλης της μνήμης και την `os.freemem()` που μας δίνει το μέγεθος της ελεύθερης μνήμης υπολογίζουμε το ποσοστό της ελεύθερης μνήμης RAM.

Η τρίτη και τελευταία παράμετρος που λαμβάνεται υπόψη είναι η ταχύτητα σύνδεσης μεταξύ διακομιστή και εφαρμογής πελάτη. Ο υπολογισμός της πραγματοποιείται με το αίτημα του πελάτη για μια εικόνα μεγέθους 5 mb η οποία μόλις ολοκληρωθεί η αποστολή και λήψη της από την εφαρμογή πελάτη, αυτή υπολογίζει την ταχύτητα σε kbps και στέλνει πίσω στο διακομιστή το αποτέλεσμα. Ο διακομιστής με την σειρά του λαμβάνει την ταχύτητα μέσω του req.query.speed την οποία διαιρεί με το μέγεθος του διαγράμματος imagesize που πρόκειται να σταλεί στον πελάτη και εμφανίζει το αποτέλεσμα στην κονσόλα. Το αποτέλεσμα είναι σε δευτερόλεπτα και αντικατοπτρίζει τον χρόνο που θα περιμένει ο πελάτης για να λάβει το

```
var speedtime = imagesize / req.query.speed ;
console.log("Speed is (kbps):");
console.log(req.query.speed);
console.log("Speed time is :");
console.log(speedtime);
```

σχετικό διάγραμμα με τα αποτελέσματα των εκλογών της περιοχής όπου βρίσκεται.

Έχοντας υπολογίσει όλες τις παραμέτρους που χρειάζονται για την απόκριση του διακομιστή στην εφαρμογή πελάτη όπως παρατηρούμε στην

```
if(count_connection <= 2 && percentRam > 20 && percentCpu < 20 && speedtime < 12){
  res.setHeader('Content-Type', 'application/json');
  res.jsonp([
    { streamstatus: undefined,
      url: 'https://plot.ly/~geo7geg/2',
      message: '',
      warning: '',
      filename: 'get-requests-example',
      error: 'errorimage' }
  ]);
  return console.log (error);
}else{
  res.setHeader('Content-Type', 'application/json');
  res.jsonp([
    { streamstatus: undefined,
      url: 'https://plot.ly/~geo7geg/2',
      message: '',
      warning: '',
      filename: 'get-requests-example',
      error: 'nosend' }
  ]);
  return console.log (error);
}
```

πρώτη γραμμή της εικόνας συγκρίνουμε τα αποτελέσματα των υπολογισμών με κάποιες σταθερές παραμέτρους που θα πρέπει να ικανοποιούνται όλες ειδάλλως η απόκριση του διακόμιστη δεν θα περιέχει το διάγραμμα των αποτελεσμάτων. Ο αριθμός των παράλληλων συνδέσεων θα πρέπει να είναι τουλάχιστον δύο , η χρήση του επεξεργαστή να έχει ποσοστό λιγότερο του 20% όπως και το ποσοστό της ελεύθερης μνήμης RAM να είναι μεγαλύτερο του 20% και τέλος ο χρόνος απόκρισης να είναι λιγότερος από τα 12 δευτερόλεπτα που θεωρούνται ο μέσος όρος αναμονής ενός χρήστη πελάτη. Αφού ικανοποιηθούν όλες οι συγκρίσεις των αποτελεσμάτων με τις σταθερές παραμέτρους τότε ο διακομιστής αποκρίνεται στην εφαρμογή πελάτη και στέλνει απάντηση σε μορφή json όπου ορίζει το url του διαγράμματος που πρέπει να φορτώσει και το error σε περίπτωση που δεν ικανοποιηθούν οι συγκρίσεις με αποτέλεσμα να αποκριθεί με nosend. Η εφαρμογή πελάτη δέχεται την απάντηση και εμφανίζει τα σχετικά αποτελέσματα και σε περίπτωση που δεν σταλεί το διάγραμμα, ενημερώνει με μήνυμα ότι δεν μπορεί να φορτωθεί.

ΚΕΦΑΛΑΙΟ ΠΕΜΠΤΟ ΣΕΝΑΡΙΑ - ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ

8.1 ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο περιγράφονται οι περιπτώσεις χρήσης της εφαρμογής πελάτη που αναπτύχθηκε με γνώμονα την αρχιτεκτονική του fog computing, υλοποιημένη σε javascript και με θέμα την παρουσίαση των αποτελεσμάτων των δημοτικών εκλογών 2014.



Εικόνα 24 : Αρχική σελίδα εφαρμογής πελάτη

Στην παραπάνω εικόνα παρατηρούμε την αρχική σελίδα της εφαρμογής πελάτη όπου ο χρήστης αλληλεπιδρά με σκοπό να λάβει τα αποτελέσματα που επιθυμεί. Υπάρχουν τρεις διαφορετικές περιπτώσεις χρήσης. Η πρώτη βασίζεται στην τοποθεσία του χρήστη καθώς ο διακομιστής θα αποκριθεί με αποτελέσματα σχετικά με την περιοχή του.

Η δεύτερη περίπτωση έχει να κάνει με τις παραμέτρους της server side εφαρμογής οι οποίες αναφέρθηκαν στην υλοποίηση και εάν επαληθευτούν, επιστρέφουν στην εφαρμογή πελάτη τα αποτελέσματα σε μορφή κειμένου στοιχισμένα σε ειδικά μορφοποιημένο πίνακα και ένα διάγραμμα αλληλεπίδρασης με τα σχετικά αποτελέσματα των εκλογών της περιοχής.

Η τρίτη και τελευταία περίπτωση χρήσης αποτελεί συνέχεια της δεύτερης και παρουσιάζεται το πως αποκρίνεται ο διακομιστής όταν υπολογίζει ότι τα δεδομένα που επιθυμεί να στείλει, καθυστερούν να φτάσουν στην εφαρμογή πελάτη. Αυτό έχει ως αποτέλεσμα να στείλει μόνο τα δεδομένα σε μορφή

κειμένου , ακυρώνοντας την αποστολή του διαγράμματος λόγω αργής ταχύτητας σύνδεσης και καθυστερήσεων του δικτύου.

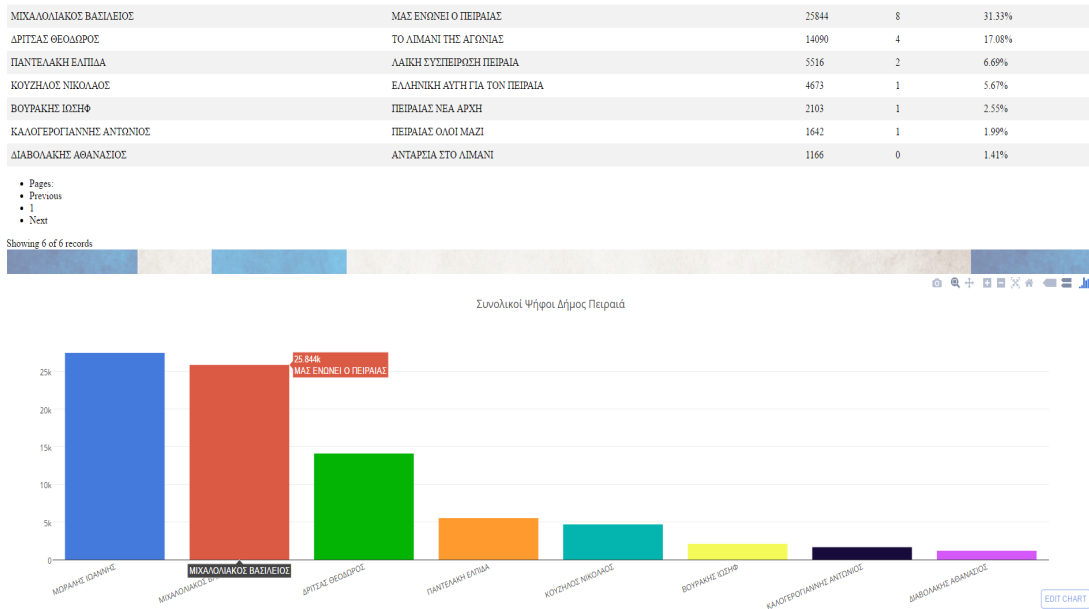
8.2 ΠΡΩΤΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ

Αρχικά, στο πρώτο σενάριο χρήσης υπάρχουν δύο εφαρμογές πελάτη από τις οποίες η πρώτη τοποθετείται σε περιοχή του Πειραιά και η δεύτερη στην περιοχή της Ζωγράφου.



Εικόνα 25 : Λειτουργίες εφαρμογής πελάτη

Με την επιλογή της Ευρεσης Αποτελεσμάτων , η εφαρμογή πελάτη ανιχνεύει αυτόματα την τοποθεσία την οποία βρίσκεται και στέλνει αίτημα Http, που περιέχει την περιοχή του χρήστη, στον διακομιστή που λειτουργεί ως load balancer όπου λαμβάνει χώρα η προ επεξεργασία του συστήματος. Στην συνέχεια, ο load balancer προωθεί το αίτημα σε τοπικό διακομιστή της περιοχής του χρήστη και αποκρίνεται αποτελέσματα σχετικά με την περιοχή που έλαβε. Στην πρώτη περίπτωση, η εφαρμογή πελάτη θα λάβει αποτελέσματα εκλογών του Δήμου Πειραιά.



Εικόνα 26 : Αποτελέσματα Δήμου Πειραιά σε πίνακα και διάγραμμα

Στην δεύτερη περίπτωση, θα λάβει αποτελέσματα που αφορούν τον Δήμο Ζωγράφου.



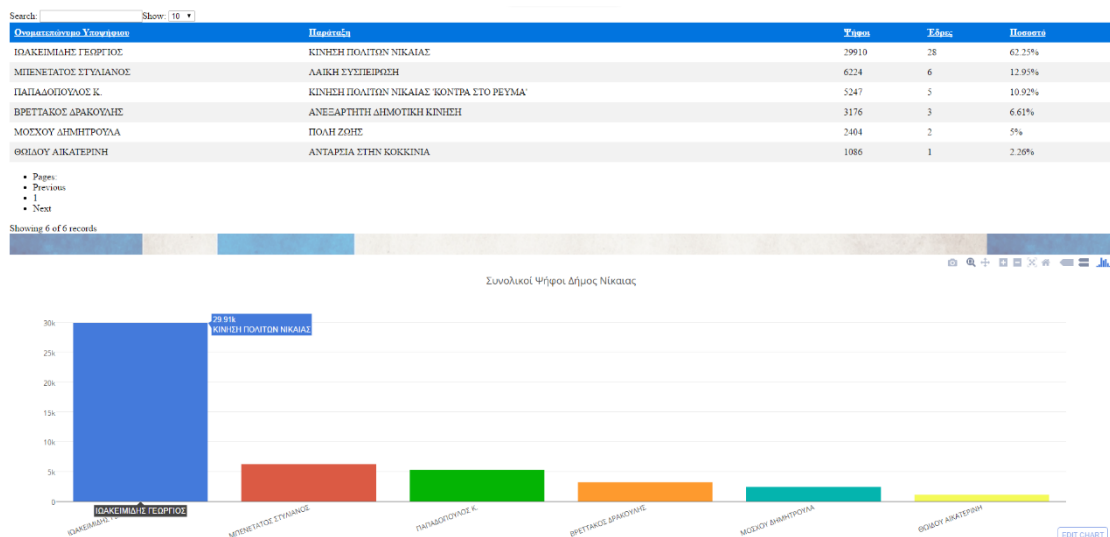
Εικόνα 27 : Αποτελέσματα Δήμου Ζωγράφου σε πίνακα και διάγραμμα

Και στις δύο περιπτώσεις λαμβάνουμε αποτελέσματα σε μορφή κειμένου και διαγραμμάτων τα οποία είναι σχετικά με την περιοχή της εφαρμογής χρήστη. Ο load balancer παίζει ρόλο προωθητή των αιτημάτων Http και προ επεξεργάζεται το πού θα στείλει το κάθε αίτημα με σκοπό ο διακομιστής να είναι τοπικός και όσο το δυνατόν κοντινότερος. Με αυτόν τον τρόπο πετυχαίνει να ταξιδέψει το αίτημα όσο το δυνατόν λιγότερους κόμβους έχοντας ως αποτέλεσμα την ισχυρότερη ασφάλεια και ακεραιότητα των δεδομένων. Επίσης, μοιράζει τον φόρτο του δικτύου σε διαφορετικούς διακομιστές έτσι ώστε να μην εστιάζονται τα αιτήματα σε ένα μεγάλο κέντρο δεδομένων.

8.3 ΔΕΥΤΕΡΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ

Στο δεύτερο σκέλος του συστήματος , η εφαρμογή χρήστη υπολογίζει την ταχύτητα μεταξύ αυτής και του τοπικού διακομιστή εκτός της τοποθεσίας της και στέλνει αίτημα το οποίο φτάνει στον τελικό διακομιστή έτσι ώστε να υπολογίσει με τη σειρά του μέσα από πολλές παραμέτρους τα αποτελέσματα του και την αποκρίσή του.

Οπότε, επιλέγοντας πάλι την Εύρεση Αποτελεσμάτων , ο διαμοιραστής αιτημάτων θα προωθήσει το αίτημα που περιέχει την τοποθεσία και την ταχύτητα σε τοπικό διακομιστή όπου θα πραγματοποιηθεί η κεντρική επεξεργασία του συστήματος. Ο τοπικός διακομιστής συγκρίνει τα δεδομένα που συλλέγει με τις παραμέτρους και στην προκειμένη περίπτωση που επαληθεύονται όλες τότε αποκρίνεται με αποτελέσματα σε μορφή κειμένου και διαγράμματος όπως στην παρακάτω εικόνα.



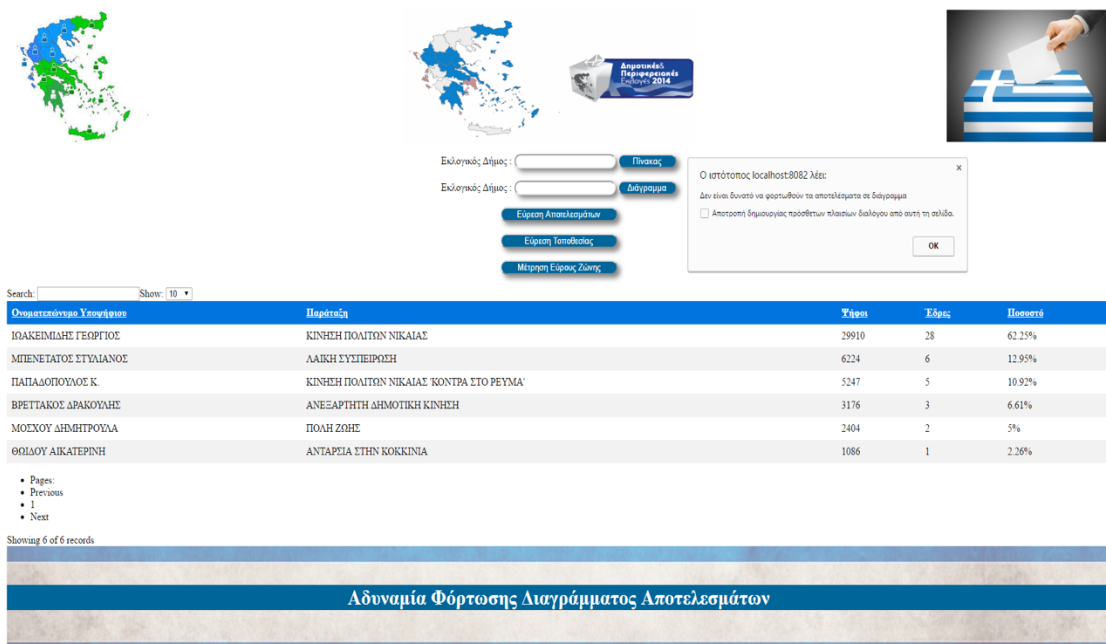
Εικόνα 28 : Αποτελέσματα Δήμου Νίκαιας σε πίνακα και διάγραμμα

Στο συγκεκριμένο σενάριο ο αριθμός των συνδέσεων με τον διακομιστή ήταν λιγότερες από όσες ορίζει το σύστημα , η ταχύτητα του επεξεργαστή του

και της μνήμης RAM κρίνονται επαρκείς και η ταχύτητας σύνδεσης μεταξύ του διακομιστή και του χρήστη ήταν αρκετή ώστε να αποκριθεί με όλες τις μορφές αποτελεσμάτων.

8.4 ΤΡΙΤΗ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ

Το τελευταίο σενάριο χρήσης αναφέρεται στην περίπτωση όπου ο τοπικός διακομιστής αποκρίνεται μόνο με αποτελέσματα εκλογών της περιοχής του χρήστη σε μορφή κειμένου καθώς είτε η ταχύτητα σύνδεσης μεταξύ της εφαρμογής χρήστη και διακομιστή κρίνεται ελλιπής είτε ο αριθμός των συνδέσεων του τοπικού διακομιστή είναι μεγάλος είτε ο φόρτος εργασίας του τοπικού διακομιστή σε επεξεργαστή και μνήμη RAM καθιστά αδύνατη την απόκριση και με διάγραμμα αποτελεσμάτων.



Εικόνα 29 : Αποτελέσματα Δήμου Νίκαιας σε πίνακα με αδυναμία φόρτωσης διαγράμματος

Αυτό έχει ως αποτέλεσμα η εφαρμογή πελάτη να δεχτεί μόνο τα αποτελέσματα σε μορφή κειμένου όπως στην παραπάνω εικόνα και να εμφανίσει πως η φόρτωση διαγράμματος αποτελεσμάτων κρίνεται αδύνατη καθώς ο τοπικός διακομιστής απέρριψε την αποστολή του.

Η μη αποστολή των παραπάνω δεδομένων λόγω της επεξεργασίας παραμέτρων του τοπικού διακομιστή αποτελεί ένα επιπλέον πλεονέκτημα της αρχιτεκτονικής του fog computing αφού για να επιτύχουμε αυτήν την απόφαση έπρεπε να χρησιμοποιήσουμε τοπική υπολογιστική δύναμη η οποία διαμοιράζεται σε πολλούς διακομιστές ενώ με την περίπτωση του cloud computing θα έπρεπε αυτός ο υπολογισμός να γίνει σε ένα κεντρο δεδομένων όπου ο φόρτος εργασίας θα ήταν τεράστιος. Τέλος, με τον συγκεκριμένο τρόπο γλυτώνουμε την αποστολή δεδομένων που θα καθυστερούσαν την ροή του δικτύου και βελτιώνει την ταχύτητα και την αξιοπιστία του.

ΚΕΦΑΛΑΙΟ ΕΚΤΟ ΣΥΜΠΕΡΑΣΜΑΤΑ

Εν κατακλείδι, το fog computing είναι ιδανική πλατφόρμα για το Διαδίκτυο των Πραγμάτων (Internet of Things) και υπηρεσίες όπως είναι το διασυνδεδεμένο όχημα (connected vehicle), τις έξυπνες πόλεις, τα δίκτυα που χρησιμοποιούν αισθητήρες και έχει τη δυνατότητα να επεκτείνει το «μοντέλο» του cloud computing στα «άκρα» του δικτύου δηλαδή στα συστήματα που χρησιμοποιούμε προκειμένου να αποκτήσουμε πρόσβαση στο Διαδίκτυο όπως είναι, για παράδειγμα, ένα router. Συμπερασματικά, αυτό οδηγεί σε

μεταφορά της υπολογιστικής ισχύος στα «άκρα» του δικτύου, μειώνοντας την ανάγκη για επεξεργασία στα κέντρα δεδομένων.

Αυτός είναι και ο λόγος που τους τελευταίους μήνες όλο και περισσότεροι από τους μεγάλες κατασκευαστές εξοπλισμού για το χώρο των τηλεπικοινωνιών και της πληροφορικής έχουν αρχίσει να ασχολούνται με το fog computing. Η Cisco ήταν μεν η πρώτη αλλά τώρα στη σχετική λίστα περιλαμβάνονται εταιρείες όπως η Intel, η IBM και η Samsung, οι οποίες προσπαθούν να ενσωματώσουν το fogging που αναμένεται να μπορέσει να τους δώσει λύσεις στα ζητήματα που δημιουργεί το cloud.

Καταλήγοντας, αναλύσαμε το Fog Computing και τις εφαρμογές πραγματικού χρόνου του και παρατηρήσαμε ότι μπορεί να διαχειριστεί τον τεράστιο όγκο δεδομένων που δημιουργεί το Διαδίκτυο των Πραγμάτων. Η υλοποίηση της καινοτόμου προσέγγισης που παρουσιάσαμε με βάση την αρχιτεκτονική του fog computing, αποδεικνύει την βελτίωση της καθυστέρησης του δικτύου με την απορρόφηση του φόρτου από τα κέντρα δεδομένων και της τοπικής επεξεργασίας σε συνδυασμό με την παραγωγή αποτελεσμάτων.

ΚΕΦΑΛΑΙΟ ΕΒΔΟΜΟ ΒΙΒΛΙΟΓΡΑΦΙΑ

- ✚ [1] Υπουργείο Εσωτερικών, Δεδομένα Εφαρμογής, <http://www.ypes.gr>
- ✚ [2] Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., ... & Stoica, I. (2009). Above the clouds: A Berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28(13)*, 2009.
- ✚ [3] Krutz, R. L., & Vines, R. D. (2010). *Cloud security: A comprehensive guide to secure cloud computing*. Wiley Publishing.
- ✚ [4] Wikipedia, AWS | Amazon Elastic Compute Cloud (EC2) – Scalable Cloud Servers, https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud
- ✚ [5] Wikipedia, Windows Azure, https://en.wikipedia.org/wiki/Microsoft_Azure
- ✚ [6] F. Bonomi, “Connected vehicles, the internet of things, and fog computing,” in The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA, 2011
- ✚ [7] Open fog consortium, Top 5 ways Fog Computing can make IoT more secure, <https://www.openfogconsortium.org/top-5-ways-fog-computing-can-make-iot-more-secure/>
- ✚ [8] Wikipedia 2015 , IoT, https://en.wikipedia.org/wiki/Internet_of_Things
- ✚ [9] Bassi, A., Bauer, M., Fiedler, M., Kramp, T., Van Kranenburg, R., Lange, S., & Meissner, S. (2013). Enabling things to talk. *Designing IoT solutions with the IoT architectural reference model*, 163-211.
- ✚ [10] Mattern, F., & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. In *From active data management to*

event-based systems and more (pp. 242-259). Springer Berlin Heidelberg.
<http://vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf>

- ✚ [11] Vermesan, O., & Friess, P. (Eds.). (2014). *Internet of things-from research and innovation to market deployment* (pp. 74-75). Aalborg: River Publishers.
- ✚ [12] Cloud Security Alliance, “Top Threat to Cloud Computing V1.0,” March 2010.[Online].Available: <https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>
- ✚ [13] Prevention Of Malicious Insider In The Cloud Using Decoy Documents by S. Muqtyar Ahmed, P. Namratha, C. Nagesh
- ✚ [14] Cloud Security: Attacks and Current Defenses Gehana Booth, Andrew Soknacki, and Anil Somayaji.
- ✚ [15] Overview of Attacks on Cloud Computing by Ajey Singh, Dr. Maneesh Shrivastava.
- ✚ [16] D.Jamil and H. Zaki, “Security Issues in Cloud Computing and Countermeasures,” *International Journal of Engineering Science and Technology*, Vol. 3 No. 4, pp. 2672-2676, April 2011.
- ✚ [17] K. Zunnurhain and S. Vrbsky, “Security Attacks and Solutions in Clouds,” 2nd IEEE InternationalConference on Cloud Computing Technology and Science, Indianapolis, December 2010.
- ✚ [18] W. A. Jansen, “Cloud Hooks: Security and Privacy Issues in Cloud Computing,” 44th Hawaii International Conference on System Sciences, pp. 1–10, Koloa, Hawaii, January 2011.
- ✚ [19] Wikipedia, Javascript, <https://en.wikipedia.org/wiki/JavaScript>

- ✚ [20] Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83.
- ✚ [21] Wikipedia, Nodejs, <https://en.wikipedia.org/wiki/Node.js>
- ✚ [22] MongoDB site, <https://www.mongodb.org/>
- ✚ [23] Banker, K. (2011). *MongoDB in action*. Manning Publications Co.