ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΕΠΙΧΕΙΡΗΣΙΑΚΗΣ ΕΡΕΥΝΑΣ

ΕΡΓΑΣΤΗΡΙΟ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ

Διπλωματική Εργασία:

«Τεχνολογίες Ανάπτυξης Λογισμικού για την Υλοποίηση Συστήματος Εμπορευματικών Μεταφορών Αστικών Περιοχών»

Ον/μο: Δεληγιάννη Αγγελική

Αρ. μητρώου: 02114350

Εποπτεία : κ. Τατσιόπουλος Ηλίας Καθηγητής Ε.Μ.Π.

Επίβλεψη: κ. Παπαδόπουλος Γεώργιος Ε.ΔΙ.Π. Ε.Μ.Π.

Κ. Γκαγιαλής Σωτήριος Ε.ΔΙ.Π. Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

Ευχαριστίες

Αρχικά θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επόπτη καθηγητή, κ. Ηλία Τατσιόπουλο, για την ανάθεση της διπλωματικής εργασίας στον ιδιαίτερα αγαπημένο μου τομέα των πληροφοριακών συστημάτων στην διοίκηση της παραγωγής

Επίσης θα ήθελα να ευχαριστήσω τους επιβλέποντες κ. Γεώργιο Παπαδόπουλο και κ. Γκαγιαλή Σωτήριο, μέλη Ε.ΔΙ.Π. στο Εργαστήριο Οργάνωσης Παραγωγής του Τομέα Βιομηχανικής Διοίκησης και Επιχειρησιακής Έρευνας της σχολής Μηχανολόγων Μηχανικών ΕΜΠ, για τις ευκαιρίες, την καθοδήγηση και τις γνώσεις πάνω στο θέμα της διπλωματικής εργασίας μου.

Ακόμα, θα ήθελα να ευχαριστήσω τον κ. Ευριπίδη Κεχαγιά, διδακτορικό φοιτητή του τομέα Τομέας Βιομηχανικής Διοίκησης και Επιχειρησιακής Έρευνας της σχολής Μηχανολόγων Μηχανικών ΕΜΠ, για την αμέριστη βοήθεια του και την καθοδήγηση του στην εκπόνηση της διπλωματικής εργασίας

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τον σύντροφό μου, για την ανιδιοτελή αγάπη τους, και την αμέριστη υποστήριξή τους στην επίτευξη των στόχων και των ονείρων μου.

Περίληψη

Η παρούσα εργασία αποτελεί μια ερευνητική προσέγγιση των τεχνολογιών ανάπτυξης ενός συστήματος δρομολόγησης και προγραμματισμού, που λειτουργεί σε περιβάλλον cloud, και λαμβάνει υπόψιν τις παραμέτρους που έχουν σχέση με την δρομολόγηση και με την ανταλλαγή δεδομένων μεταξύ των συμπλεκόμενων μερών.

Προτείνεται μια αρχιτεκτονική συστήματος σύμφωνα με τα στοιχεία και τα δεδομένα που χρειάζονται για την λειτουργία του, με βάση την οποία αναλύονται και προτείνονται κατάλληλες τεχνολογίες για την ανάπτυξη και την διασύνδεση των επιμέρους υπό-συστημάτων. Αξιολογούνται οι πλατφόρμες ανάπτυξης εφαρμογών στον Ιστό, οι υπηρεσίες για τις εφαρμογές που δημιουργούνται στις πλατφόρμες, τα πρωτόκολλα επικοινωνίας μεταξύ των εφαρμογών έως τις βάσεις δεδομένων, καταλήγοντας στους λόγους που το Λογισμικό ως Υπηρεσία αποτελεί κατάλληλη επιλογή για το πληροφοριακό σύστημα που πραγματεύεται η παρούσα διπλωματική εργασία.

Βασικός σκοπός της εργασίας είναι η αναλυτική διερεύνηση και αξιολόγηση των διαθέσιμων τεχνολογιών υλοποίησης λογισμικού για την ανάπτυξη ενός τέτοιου πληροφοριακού συστήματος.

Η διπλωματική εργασία μπορεί να διακριθεί σε τρία μέρη :

Στην πρόταση λειτουργίας μιας αρχιτεκτονικής του πληροφοριακού συστήματος της παρούσας εργασίας, στο οποίο υπεισέρχονται όλα τα εμπλεκόμενα μέρη, από τις εταιρίες logistics, στους παρόχους γεωγραφικών στοιχείων (χαρτών) μέχρι και στους παρόχους κυκλοφοριακών συνθηκών, και τα αντίστοιχα στατικά και δυναμικά δεδομένα τους.

Το δεύτερο μέρος αφορά όλες εκείνες τις τεχνολογίες που είναι αναγκαίες για την λειτουργία, ανάπτυξη και διασύνδεση όλων των στοιχείων του προτεινόμενου συστήματος. Με βασικό σκοπό την έρευνα, αξιολόγηση και τελικά την πρόταση των καταλληλότερων για το σύστημα.

Το τρίτο μέρος αφορά την λειτουργία του συστήματος στο διαδίκτυο, δηλαδή τους λόγους που στην αρχιτεκτονική το λογισμικό έχει προταθεί να παρέχεται μέσω του cloud ως υπηρεσία, και την αξία που προσδίδει μια τέτοια τεχνολογική πρόταση.

Η παρούσα διπλωματική εργασία δημιουργήθηκε και αξιοποιήθηκε στο σχεδιασμό συστήματος στα πλαίσια του ερευνητικού έργου «SMARTRANS» του Εθνικού Μετσόβιου Πολυτεχνείου.

Abstract

This paper is a research approach to the cloud-based routing and programming technology development technologies and takes into consideration the routing and data sharing parameters between the parties involved. A system architecture is proposed according to the data and data needed for its operation, based on which are analyzed and proposed suitable technologies for the development and interconnection of the individual subsystems. Web application development platforms, application-based platform applications, application-to-database communication protocols are evaluated, ending the reasons that Software as a Service is an appropriate choice for the information system that is dealt with in this diploma thesis.

The main aim of the thesis is the detailed investigation and evaluation of the available software technologies for the development of such an information system.

The dissertation can be distinguished in three parts:

In the proposal of an information system architecture of the present work, involving all the parties involved, from the logistics companies, geographic data providers (maps) to the providers of traffic conditions, and their respective static and dynamic data.

The second part concerns all those technologies that are necessary for the operation, development and interconnection of all elements of the proposed system. With the main purpose of researching, evaluating and finally proposing the most suitable for the system.

The third part is about the operation of the system on the internet as a cloud system, that is, the reasons why software has been proposed to be provided through the cloud as a service, and the value that such a technology proposes.

This thesis was created and utilized in system design in the framework of the "SMARTRANS" research project of the National Technical University of Athens.

ΠΕΡΙΕΧΟΜΕΝΑ

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

1. Εισαγωγή

1.1. Σκοπός και Στόχοι Εργασίας

Η σημερινή εποχή χαρακτηρίζεται ως ψηφιακή εποχή. Η χρήση του διαδικτύου είναι τόσο εκτενής που την συναντούμε καθημερινά χωρίς πολλές φορές να την αναγνωρίζουμε. Επομένως, η χρήση του διαδικτύου δεν θα μπορούσε να λείπει από τις διαδικασίες διαχείρισης της εφοδιαστικής αλυσίδας, και συγκεκριμένα στον τομέα του προγραμματισμού και της διανομής προϊόντων.

Με αυτή την σκέψη η έρευνα πάνω σε έναν τέτοιο τομέα αποκτά ιδιαίτερο ενδιαφέρον. Αν και η δημιουργία ενός πληροφοριακού συστήματος βασισμένο στην παροχή Λογισμικού ως Υπηρεσίας μέσω ενός υπολογιστικού νέφους, που έχει στόχο την δρομολόγηση και τον προγραμματισμό διανομής προϊόντων στις αστικές περιοχές λαμβάνοντας υπόψιν δυναμικά και στατικά δεδομένα, βρίσκεται ακόμα σε πρώιμο στάδιο, στην παρούσα διπλωματική εργασία θα ερευνηθούν οι τεχνολογίες που απαιτούνται για την υλοποίηση μιας τέτοιας ιδέας. Το σύστημα θα είναι σε θέση να δημιουργεί ένα αρχικό πλάνο δρομολόγησης μεταφορών σε αστικές περιοχές, να παρακολουθεί τον στόλο διανομής, να ανιχνεύσει αποκλίσεις από το αρχικό πλάνο και ανάλογα με το χρονοδιάγραμμα, να υποδεικνύει βέλτιστες ανά δρομολογήσεις. Χρησιμοποιώντας ένα τέτοιο σύστημα, δεδομένα σχετικά με τα απρόβλεπτα γεγονότα η διανομή γίνεται ταχύτερη και οικονομικότερη, εξασφαλίζοντας ταυτόχρονα καλύτερη εξυπηρέτηση πελατών.

Βασικός σκοπός της εργασίας είναι η αναλυτική διερεύνηση και αξιολόγηση των τεχνολογιών ενός τέτοιου πληροφοριακού συστήματος. Προτείνεται μια αρχιτεκτονική συστήματος σύμφωνα με τα στοιχεία και τα δεδομένα που χρειάζονται για την λειτουργία του, με βάση την οποία αναλύονται και προτείνονται κατάλληλες τεχνολογίες για την ανάπτυξη και την διασύνδεση των επιμέρους υπό-συστημάτων. Αξιολογούνται οι πλατφόρμες ανάπτυξης εφαρμογών στον Ιστό, οι υπηρεσίες για τις εφαρμογές που δημιουργούνται στις πλατφόρμες, τα πρωτόκολλα επικοινωνίας μεταξύ των εφαρμογών έως τις βάσεις δεδομένων, καταλήγοντας στους λόγους που το Λογισμικό ως Υπηρεσία αποτελεί κατάλληλη επιλογή για το πληροφοριακό σύστημα που πραγματεύεται η παρούσα διπλωματική εργασία.

1.2. Δομή Εργασίας

Κεφάλαιο 2 : Αρχιτεκτονική Συστήματος

Στο κεφάλαιο αυτό αναφέρονται αναλυτικά η αρχιτεκτονική του συστήματος που πραγματεύεται η διπλωματική και ο τρόπος λειτουργίας του συστήματος συνολικά, καθώς και οι διάφορες λειτουργίες των επιμέρους υποσυστημάτων και διαπλεκόμενων. Με σκοπό να αποκτηθεί μια εικόνα των λειτουργιών και των αντίστοιχων τεχνολογιών που χρειάζεται ένα τέτοιο σύστημα

Κεφάλαιο 3 : Τεχνολογίες ανάπτυξης και διασύνδεσης

Ερευνώνται και αξιολογούνται αναλυτικά οι τεχνολογίες ανάπτυξης και διασύνδεσης του πληροφοριακού συστήματος, η πλατφόρμα ανάπτυξης, οι διαδικτυακές υπηρεσίες και τα πρωτόκολλα επικοινωνίας/ ανταλλαγής δεδομένων μεταξύ των εφαρμογών. Καταλήγοντας σε αντίστοιχες προτάσεις των επιμέρους στοιχείων για το σύστημα. Το συγκεκριμένο κεφάλαιο στοχεύει στην εύρεση των τεχνολογιών που θα μπορούσαν να λειτουργήσουν σωστά και ομαλά το σύστημα.

Κεφάλαιο 4 : Βάσεις δεδομένων

Παρουσιάζεται μια βασική τεχνολογία για την αποθήκευση και χρήση των δεδομένων που εισέρχονται και εξέρχονται στο σύστημα. Μελετώνται τα διάφορα είδη βάσεων δεδομένων που υπάρχουν στο εμπόριο, καταλήγοντας στην βάση που θα ήταν χρησιμότερη για τον πάροχο του Λογισμικού ως Υπηρεσία. Με σκοπό την εμφάνιση εκείνης της βάσης δεδομένων που θα εξυπηρετούσε καταλληλότερα το σύστημα.

Κεφάλαιο 5 : Μοντέλα παροχής λογισμικού

Στα μοντέλα παροχής λογισμικού μελετάται η εξέλιξη του τρόπου παροχής λογισμικού στις εταιρίες, από την απλή τοπική εγκατάσταση του λογισμικού στους διακομιστές της εταιρίας, στην χρήση του φιλοξενούμενου λογισμικού, έως την χρήση του λογισμικού μέσω του διαδικτύου ως υπηρεσία. Στοχεύοντας στην εμφάνιση του μοντέλου που εξυπηρετεί κατάλληλα το σκοπό του συστήματος.

Κεφάλαιο 6 : Μοντέλα ανάπτυξης λογισμικού στο Cloud

Σε αυτό το κεφάλαιο παρουσιάζεται η χρήση του υπολογιστικού νέφους (Cloud) ως μέσο ανάπτυξης λογισμικού και οι άλλες μορφές που μπορούν οι εταιρίες να χρησιμοποιούν το υπολογιστικό νέφος για διάφορες υπηρεσίες. Επίσης, στο συγκεκριμένο κεφάλαιο γίνεται

εκτενής αναφορά του Λογισμικό ως Υπηρεσία (SaaS) ως επιχειρηματικό μοντέλο για τα διάφορα είδη επιχειρήσεων, και η μελέτη των διαφόρων πλεονεκτημάτων του στις επιχειρήσεις και στους πελάτες τους. Επιπρόσθετα, αναλύονται δύο βασικά μοντέλα υπηρεσιών του εμπορίου και προτείνεται ένας συνδυασμός χρήσης τους στο σύστημα της εργασίας. Στόχος αυτού του κεφαλαίου είναι η παρουσίαση των εναλλακτικών επιλογών που υπάρχουν στο εμπόριο για την χρήση του υπολογιστικού νέφους και η πρόταση του συνδυασμού που εξυπηρετεί το σύστημα και τις τεχνολογίες του.

Κεφάλαιο 7 : Συμπεράσματα

Στο τελευταίο αυτό κεφάλαιο παραθέτουμε τα συμπεράσματα από τη έρευνα και αξιολόγηση των τεχνολογιών που αναπτύχθηκαν στα προηγούμενα κεφάλαια της διπλωματικής εργασίας, και η τελική συνολική πρόταση για το πληροφοριακό σύστημα.

1.3. Διαχείριση Διαδικασιών Εφοδιαστικής Αλυσίδας

Η διαχείριση των διαδικασιών της εφοδιαστικής αλυσίδας αποτελεί έναν από τους κορυφαίους τομείς ενδιαφέροντος των επιχειρήσεων, καθώς ασκεί μεγάλη επιρροή στην απόδοση των επιχειρήσεων στην σημερινή εποχή και στην διασφάλιση της ποιότητας των υπηρεσιών/ προϊόντων τους. Στο σύγχρονο ανταγωνιστικό περιβάλλον ο σωστός έλεγχος των διαδικασιών της εφοδιαστικής αλυσίδας είναι ανέφικτος δίχως την χρήση των πληροφοριακών συστημάτων, από τον σχεδιασμό της παραγωγής έως μεταφορά και την τελική παράδοση του προϊόντος στον καταναλωτή.

Με τον όρο διαχείριση της εφοδιαστικής αλυσίδας εννοούμε διαχείριση της ροής υλικών, πληροφοριών και υπηρεσιών από τους προμηθευτές πρώτων υλών έως και τους τελικούς πελάτες. Αυτή η διαχείριση μπορεί να χωριστεί σε δύο κατηγορίες, στον προγραμματισμό της εφοδιαστικής αλυσίδας και στην εκτέλεση της εφοδιαστικής αλυσίδας.

Ο προγραμματισμός της εφοδιαστικής αλυσίδας είναι ένας τομέας που έχει μελετηθεί κατά κόρων και έχουν αναπτυχθεί πολλά τεχνολογικά συστήματα που βοηθούν στις διαδικασίες προγραμματισμού εφοδιασμού (MRP I, MRP II) καθώς και ολοκληρωμένα συστήματα SCP.

Από την άλλη πλευρά, υπάρχει και το εκτελεστικό κομμάτι της εφοδιαστικής αλυσίδας. Σε αυτό μπορεί να αναφερθεί ότι συγκεκριμένοι τομείς έχουν λάβει μεγαλύτερη μελέτη, όπως είναι η διαχείριση των αποθηκών με χαρακτηριστικό παράδειγμα τα πληροφοριακά συστήματα διαχείρισης Αποθηκών (WMS). Υπάρχει, όμως, και μια ενότητα εκτελεστικής διαδικασίας που αφορά την διανομή των προϊόντων.

Στον τομέα αυτόν οι περισσότερες μελέτες έχουν επικεντρωθεί στον βέλτιστο προγραμματισμό και στην βέλτιστη δρομολόγηση της διανομής κάτω από γνωστές και σταθερές μεταβλητές. Αυτές οι μελέτες αφορούν το ευρέως γνωστό πρόβλημα της δρομολόγησης των οχημάτων (VRP). Ωστόσο, η παραδοχή ότι αυτές οι παράμετροι είναι στατικές είναι λανθασμένη από την στιγμή που το περιβάλλον είναι μεταβλητό. Μία παραλλαγή του VRP είναι το πρόβλημα δρομολόγησης με χρονικά παράθυρα (VRPTW) εντός των οποίων πρέπει να παραδοθούν οι διάφορες απαιτήσεις των πελατών, καθώς και οι τύποι οχημάτων του στόλου (VRP με ετερογενές στόλο - VRPHF), για τους οποίους ποικίλλουν τα χαρακτηριστικά, η ικανότητα και τα κόστη μεταφοράς.

Με βάση την ανάγκη για έρευνα όσον αφορά την εκτέλεση των διανομών σε αστικές περιοχές στις οποίες υπεισέρχονται και δυναμικά δεδομένα, όπως είναι η κίνηση στους δρόμους, οι καιρικές συνθήκες, προβλήματα του στόλου διανομής, ο κύριος στόχος αυτής της εργασίας είναι να προτείνει ένα σύστημα διαχείρισης της διανομής προϊόντων με δυναμική αντιμετώπιση γεγονότων σε πραγματικό χρόνο.

1.4. Πληροφοριακά Συστήματα στην Εκτέλεση της Διανομής

Μέχρι τώρα τα πληροφοριακά συστήματα για τον προγραμματισμό και την δρομολόγηση των διανομών βρίσκονταν τοπικά εγκατεστημένα στους διακομιστές της εκάστοτε εταιρίας. Ωστόσο, η ανάγκη για προγραμματισμό σε επίπεδο πραγματικού χρόνου θα καθιστούσε ανέφικτό και οικονομικά απρόσιτο την σύνδεση κάθε μιας εταιρίας με του παρόχους των δυναμικών δεδομένων και συγκεκριμένα με τους παρόχους των γεωγραφικών στοιχείων και συνθηκών.

Η λύση για αυτό το πρόβλημα είναι η χρήση ενός διαδικτυακού λογισμικού που έχει στόχο τον προγραμματισμό και την δρομολόγηση των διανομών κάθε εταιρίας, δηλαδή την χρήση του Λογισμικού ως Υπηρεσίας, που αποτελεί μια νέα τεχνολογική εξέλιξη στον τομέα της πληροφορικής. Με αυτόν τον τρόπο οι εταιρίες Logistics δεν είναι υποχρεωμένες να επικοινωνούν με τους παρόχους των δυναμικών δεδομένων που δεν τις αφορούν, παρά μόνο πληρώνουν τις άδειες χρήσης στον πάροχο του Λογισμικού. Έτσι, ο πάροχος του Λογισμικού ως Υπηρεσία είναι εκείνος που επιβαρύνεται με διασυνδέσεις με τους υπόλοιπους παρόχους και με τις εταιρίες Logistics.

2. Αρχιτεκτονική Συστήματος

2.1. Εισαγωγή

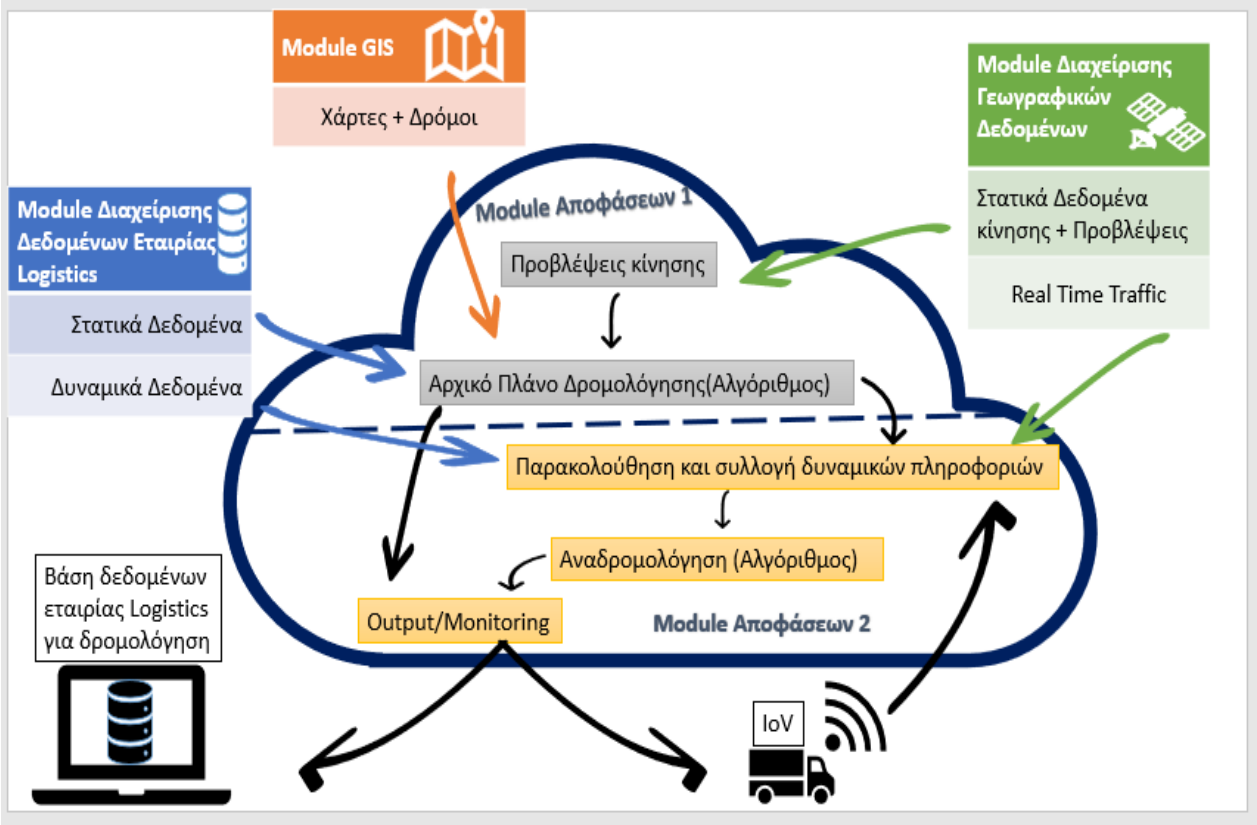
Στο συγκεκριμένο κεφάλαιο παρουσιάζεται η αρχιτεκτονική ενός Online συστήματος δρομολόγησης και προγραμματισμού διανομής, το οποίο αντιμετωπίζει στατικά δεδομένα και δυναμικές αποκλίσεις από το αρχικό πλάνο δρομολόγησης, και παρέχει ανά δρομολογήσεις για τη βελτίωση της απόδοσης των αστικών μεταφορών. Αρχικά, παρουσιάζεται ο τρόπος λειτουργίας και η σύνδεση των συμβαλλόμενων του συστήματος, δίνοντας μια γενικότερη αίσθηση της αρχιτεκτονικής. Στη συνέχεια, αναλύονται τα υποσύνολα της αρχιτεκτονικής (υποσυστήματα), εμβαθύνοντας στην λειτουργία του υποσυστήματος ως σύνολο. Και τέλος, γίνεται εστίαση των επιμέρους modules, όπου γίνεται εκτενής αναφορά της λειτουργικότητας, της χρησιμότητας, των παρεχόμενων δεδομένων και των τεχνολογιών που χρησιμοποιούν για το σύστημα.

2.2. Αρχιτεκτονική

2.2.1. Γενική Αρχιτεκτονική και Λειτουργία Του Συστήματος

Το σύστημα διαχειρίζεται η εταιρία λογισμικού το οποίο παρέχεται στις εταιρίες logistics χρησιμοποιώντας το μοντέλο λογισμικό ως υπηρεσία (SaaS). Έχοντας πρόσβαση στις υπηρεσίες του συστήματος, η εταιρία logistics αποστέλλει στοιχεία που αφορούν τις παραγγελίες (σημεία παράδοσης, χρόνους παράδοσης, παράθυρα χρόνου) και στοιχεία για τον στόλο της (οδηγούς, οχήματα, χωρητικότητα). Στη συνέχεια, το σύστημα συνδυάζει τα παραπάνω μαζί με ιστορικά στοιχεία κίνησης καταλήγοντας σε πρώτη φάση σε μια στατιστική εκτίμηση της κίνησης, δημιουργώντας τελικά ένα αρχικό πλάνο δρομολόγησης. Αυτό επιστρέφει στις εταιρίες Logistics και καταλήγει στις συσκευές πλοήγησης μέσα στα οχήματα της εταιρίας. Έτσι, ξεκινάει η μεταφορά των προϊόντων από τα οχήματα στους πελάτες. Ωστόσο, κατά τη διάρκεια της παράδοσης, στον διακομιστή (server) της εταιρίας logistics καταφτάνουν στοιχεία που μπορεί να απαιτούν τροποποίηση της διαδρομής των οχημάτων. Τέτοια μπορεί να είναι είτε ακύρωση παραγγελιών, αλλαγές στα παράθυρα χρόνου, μηχανικές βλάβες του οχήματος αλλά και απρόοπτα έξω-εταιρικά γεγονότα όπως για παράδειγμα κυκλοφοριακά προβλήματα πραγματικού χρόνου, άσχημες καιρικές συνθήκες που μπορούν να συμβούν κατά την εκτέλεση ενός δρομολογίου. Σε αυτή τη περίπτωση αν και εφόσον χρειαστεί το σύστημα υλοποιεί ανά δρομολόγηση και ενημερώνει τις συσκευές πλοήγησης μέσα στα οχήματα της εταιρίας, ώστε ο εκάστοτε οδηγός να προβεί σε αλλαγή είτε της πορείας του είτε και του πλάνου παραδόσεων του.

Επομένως, με βάση τα παραπάνω φαίνεται ότι η αρχιτεκτονική του συστήματος αποτελείται από 3 βασικά υποσυστήματα. Το υποσύστημα διαχείρισης δεδομένων που περιλαμβάνει την πλατφόρμα για την δρομολόγηση των οχημάτων βασισμένο στα στατικά και στα δυναμικά στοιχεία, το υποσύστημα επεξεργασίας δεδομένων ανάμεσα στο υποσύστημα διαχείρισης δεδομένων και στα οχήματα και τέλος το υποσύστημα που βρίσκεται στο όχημα που επιτρέπει την αλληλεπίδραση ανάμεσα στην πλατφόρμα του λογισμικού στις συσκευές πλοήγησης των οχημάτων και στο υποσύστημα διαχείρισης δεδομένων της εταιρίας. Όπως φαίνεται στο Σχήμα 2.1



Σχήμα 2.1: Έξυπνη Αρχιτεκτονική Συστήματος Μεταφορών

2.2.2. Υποσύστημα Διαχείρισης Δεδομένων

Στο υποσύστημα διαχείρισης δεδομένων περιλαμβάνονται οι βάσεις δεδομένων της εταιρίας logistics, με διάφορα στοιχεία και πληροφορίες της εταιρίας και της διανομής. Επιπλέον, σε αυτό το υποσύστημα συναντάται το Γεωγραφικό Σύστημα Πληροφοριών (ΓΣΠ), γνωστό ευρέως και ως G.I.S. Geographic Information Systems, το οποίο αποτυπώνει τον πραγματικό κόσμο σε δια δραστική μορφή, επιτρέπει στις εταιρίες να αναλύουν τα χωρικά δεδομένα (Spatial Data), δηλαδή γεωγραφικά δεδομένα, σημεία, γραμμές, πολύγωνα κ.α. που παρέχουν πληροφορίες σχετικά με γεωγραφικές ή χωρικές τοποθεσίες, και να τα

χρησιμοποιούν ως ένα σύστημα επικοινωνίας για τον εντοπισμό τερματικών και άλλων τοποθεσιών (Bueno, 2011).Τέλος, η διαχείριση των γεωγραφικών δεδομένων κίνησης στοχεύει στην βελτιστοποίηση της διανομής σε πραγματικό χρόνο υποδεικνύοντας τόσο στατιστικά στοιχεία κίνησης στους δρόμους, όσο και δυναμικά στοιχεία που αποσκοπούν στην ευέλικτη αποφυγή ή λύση παραγόντων του εξωτερικού περιβάλλοντος.

2.2.3. Υποσύστημα Επεξεργασίας Δεδομένων

Το υποσύστημα επεξεργασίας των δεδομένων ασχολείται με τα στοιχεία που προκύπτουν από το υποσύστημα διαχείρισης δεδομένων, και είναι χωρισμένο σε δύο modules αποφάσεων. Πιο συγκεκριμένα, τα στατικά στοιχεία της εταιρίας Logistics, των γεωγραφικών δεδομένων κίνησης και πληροφοριών καταφτάνουν στο υποσύστημα επεξεργασίας και δημιουργούν ένα αρχικό πλάνο δρομολόγησης των οχημάτων. Αυτό με την σειρά του αποστέλλεται στην διεπαφή του οχήματος για να ξεκινήσει η διανομή. από την άλλη πλευρά, τα διάφορα δυναμικά στοιχεία που καταγράφονται κατά τη διάρκεια της εκτέλεσης της διανομής, συγκεντρώνονται στο δεύτερο module αποφάσεων του υποσυστήματος επεξεργασίας δεδομένων. Σε αυτό το σημείο εάν οι δυναμικές πληροφορίες απαιτούν τροποποίηση της διανομής, δημιουργείται ένα διαφορετικό πλάνο δρομολόγησης ή/και ανά δρομολόγηση, εάν δεν απαιτείται τότε δεν αλλάζει το αρχικό πλάνο. Και σε αυτή την περίπτωση, οι τελικές αποφάσεις αποστέλλονται στον οδηγό του οχήματος. Σημαντική λεπτομέρεια σε όλη αυτή τη διαδικασία είναι η χρήση του μοντέλου Λογισμικό ως Υπηρεσία (SaaS) για την περάτωση της επεξεργασίας και της επικοινωνίας των τερματικών.

2.2.4. Υποσύστημα Διεπαφών Χρήστη

Το συγκεκριμένο υποσύστημα απαρτίζεται από δύο μέρη, την διεπαφή της συσκευής του οχήματος και την διεπαφή - βάση δεδομένων της εταιρίας logistics που αφορά τα δρομολόγια των οχημάτων. Η διεπαφή της συσκευής του οχήματος αποτελείται πρώτον από τον εξοπλισμό που εξασφαλίζει την επικοινωνία με το υποσύστημα επεξεργασίας δεδομένων, το οποίον όπως αναφέρθηκε ανωτέρω κάνει χρήση του μοντέλου παροχής υπηρεσιών SaaS, και δεύτερον από το τερματικό που εκπληρώνει όλες τις προδιαγραφές και απαιτήσεις της χρήσης του, δηλαδή οθόνη υψηλής ανάλυσης, αρκετή υπολογιστική ισχύς, ανθεκτικότητα και δυνατότητες ενσωμάτωσης διαφόρων λειτουργιών του λογισμικού.

2.3. Ανάλυση Modules Συστήματος

Το υποσύστημα διαχείρισης δεδομένων απαρτίζεται από τρείς βασικές μονάδες α) τη μονάδα γεωγραφικών πληροφοριών, β) τη μονάδα διαχείρισης δεδομένων της εταιρείας logistics, και γ) την μονάδα διαχείρισης γεωγραφικών δεδομένων κίνησης. Το υποσύστημα

επεξεργασίας δεδομένων περιλαμβάνει, την μονάδα αποφάσεων (Στατικών δεδομένων) και την μονάδα αποφάσεων (Δυναμικών δεδομένων). Η λειτουργικότητα κάθε δομοστοιχείου αναλύεται παρακάτω.

2.3.1. Μονάδα Γεωγραφικών Πληροφοριών

Τα συστήματα γεωγραφικών πληροφοριών είναι ένα εργαλείο ανάλυσης χωρικών δεδομένων. Χρησιμοποιούνται διότι, οι περισσότερες, περίπου το 80%, ανθρώπινες δραστηριότητες είναι άμεσα ή έμμεσα συνδεδεμένες με τοποθεσίες που βρίσκονται στο έδαφος. Πρέπει να σημειωθεί ότι δίνεται η δυνατότητα συλλογής δεδομένων στο πεδίο σε διάφορες θέσεις, τα οποία στη συνέχεια αναλύονται και εμφανίζονται.

Ένας χρήστης του συστήματος GIS μπορεί να βλέπει τον πραγματικό κόσμο σε θεματικά επίπεδα Σχήμα 2.2 (οδικό δίκτυο, χρήσεις γης, διοικητικές περιοχές, υδρογραφία, ανάγλυφο και δορυφορικές εικόνες), καθώς επίσης επισημαίνονται μοτίβα και σχέσεις μεταξύ γεωγραφικών δεδομένων με σκοπό την επίλυση προβλημάτων. (Esri, 2011)

Είναι λάθος να συγχέεται ο όρος GIS με τον όρο GPS. Το GIS είναι ένα πρόγραμμα υπολογιστή που έχει σχεδιαστεί για να συλλαμβάνει, να αναλύει, να ερμηνεύει και να αποθηκεύει δεδομένα που έχουν μεταδοθεί από συστήματα πλοήγησης όπως το GPS και να κάνουν τις πληροφορίες διαθέσιμες για χρήση.

Επομένως, τα τέσσερα βασικά πλεονεκτήματα των GIS χαρτών είναι

- Ο δυναμικός συμβολισμός
- Εύκολος και ακριβής εντοπισμός χωρικών στοιχείων (συντεταγμένες)
- Συνοδευτική περιγραφική πληροφορία
- Εύκολη αποθήκευση και επαναφορά χαρτών

Σχήμα 2.2 : Θεματικά Επίπεδα GIS

2.3.2. Μονάδα Διαχείρισης Δεδομένων της Εταιρείας Logistics

Η μονάδα αυτή περιέχει όλη τη βάση δεδομένων της εταιρείας logistics. Από την μία πλευρά συμπεριλαμβάνονται τα στατικά στοιχεία της εταιρείας, δηλαδή τα δεδομένα παραγγελιών, τα δεδομένα στόλου, περιορισμοί δρομολογήσεων, προβλέψεις χρόνων διανομής και time windows. Από την άλλη πλευρά τα δυναμικά στοιχεία της εταιρείας, όπως τροποποιήσεις παραγγελιών, τροποποιήσεις time windows και προβλήματα του στόλου.

Δεδομένα Παραγγελιών: Είναι στοιχεία που αφορούν την αλληλεπίδραση της εταιρείας και του πελάτη η της εταιρείας εσωτερικά. Η ονομασία ,η ποσότητα παραγγελίας, η διεύθυνση αποστολής, το κόστος, οι ημερομηνίες και οι τρόποι πληρωμής είναι μερικές λεπτομέρειες που βρίσκονται στην βάση δεδομένων των παραγγελιών.

Δεδομένα Στόλου: Βασικό χαρακτηριστικό μιας εταιρείας logistics είναι ο στόλος της. Αναλυτικότερα η ποσότητα και η διαθεσιμότητα των οδηγών, προσωπικά στοιχεία των οδηγών, η ποσότητα και η χωρητικότητα είτε των ιδιωτικών της οχημάτων είτε των ενοικιαζόμενων και διάφορα άλλα στοιχεία για την βέλτιστη λειτουργία της εταιρείας στο κομμάτι της διανομής.

Περιορισμοί Δρομολογήσεων: Οι περιορισμοί προέρχονται από την ανάλυση STEEPLE, ακρωνύμιο των λέξεων **S**ocietal**T**echnical**E**nvironmental**E**thical**P**olitical**L**egal**E**conomical. Κοινωνικά (Societal), οι περιορισμοί δημιουργούνται από τον αντίκτυπο ενός καλού συστήματος μεταφοράς στους πελάτες και στο ανθρώπινο δυναμικό της εταιρίας. Τεχνικά (technical), το μέσο και το είδος μεταφοράς καθώς και ο αλγόριθμος δρομολόγησης συντάσσουν περιοριστικούς όρους. Περιβαλλοντικά (Environmental), οι κλιματικές αλλαγές και η ασταμάτητη ρύπανση του περιβάλλοντος απαιτούν ελαχιστοποίηση της ατμοσφαιρικής ρύπανσης και της ηχορύπανσης. Ηθικά (Ethical), με βάση την υπουργική απόφαση Υ.Α. Γ438/οικ.28317/2481/09/2009, το δίκτυο μεταφορών πρέπει να εξασφαλίζει την ασφάλεια των επιβατών των οχημάτων. Πολιτικά (political) και νομικά (legal), το δίκτυο διανομής πρέπει να σέβεται διάφορους νομοθετικούς όρους και οδικούς κανονισμούς στο κομμάτι της κυκλοφορίας. Τέλος, το οικονομικό (economical) κομμάτι, δηλαδή η μείωση του κόστους μεταφοράς αποτελεί κύριο στόχο κάθε εταιρίας logistics. (Sabiri, 2017)

Προβλέψεις Χρόνων Διανομής: Οι προβλέψεις χρόνων διανομής είναι δεδομένα που προκύπτουν από την ανάλυση διαφόρων παραγόντων. Αρχικά, στοιχεία από προηγούμενες διανομές ταυτοποιούνται από τις τιμές των τότε προκαθορισμένων παραγόντων και με άλλα πιθανά ιστορικά δεδομένα. Και στη συνέχεια λαμβάνοντας υπόψιν μελλοντικούς προκαθορισμένους παράγοντες (εποχή, μέρα της εβδομάδας, περιοχή παράδοσης κ.α.) και

παράγοντες καιρικών συνθηκών , δίνει μια εκτίμηση του χρόνου του ταξιδιού. Όλα αυτά τα στοιχεία αντλούνται από βάσεις δεδομένων οδικών σταθμών, μετεωρολογικών σταθμών, οδικής κυκλοφορίας κλπ.

Time Windows: Τα παράθυρα χρόνου είναι καθορισμένες ημερήσιες περίοδοι κατά τις οποίες γίνεται/επιτρέπεται η διανομή. Εξαρτώνται τόσο από την πολιτεία όσο και από την ίδια την εταιρεία. Για παράδειγμα, σύμφωνα με τις διατάξεις της υπ' αριθμό Δ.Ο.Υ/ οικ-4836/Φ.911 από 17/21.09.2018 (ΦΕΚ 4179, τ. Β΄) Κοινής Υπουργικής Απόφασης, για την εύρυθμη μετακίνηση των πολιτών στο κέντρο της Αθήνας και την αποφυγή κυκλοφοριακής συμφόρησης, δεν επιτρέπεται η κυκλοφορία οχημάτων τροφοδοσίας (παράδοση – παραλαβή) κατά τις ώρες από 10:30 μέχρι 14:30. (Clausen, Friedrich, Thaller, & Geiger, 2016). Από την πλευρά της εταιρίας προς τους πελάτες της, οι παραδόσεις γίνονται σε τακτά χρονικά διαστήματα και πραγματοποιούνται εντός ενός προγραμματισμένου χρονικού πλαισίου. Το παράθυρο χρόνου επιλέγεται από τον ίδιο τον πελάτη, ώστε να μπορεί να γίνει η παραλαβή. Συνηθίζεται, αυτό είτε να συμπληρώνεται κατά την δημιουργία της παραγγελίας ηλεκτρονικά ή μη, είτε να είναι καθορισμένο από τον πελάτη στην περίπτωση που υπάρχει συχνή αλληλεπίδραση.

Δυναμικά δεδομένα : Ο πραγματικός κόσμος χαρακτηρίζεται από μεγάλη αβεβαιότητα, επομένως όλα τα παραπάνω στοιχεία δεν μπορούν να βελτιστοποιήσουν το μοντέλο λειτουργίας μιας εταιρείας Logistics. Καθημερινά υπεισέρχονται αβέβαιοι ή πιο σωστά δυναμικοί παράγοντες όπως τροποποιήσεις παραγγελιών, τροποποιήσεις time windows και προβλήματα του στόλου. Οι τροποποιήσεις των παραγγελιών και των Time windows αφορούν την πελατειακή αλληλεπίδραση της εταιρείας, ενώ τα προβλήματα στόλου (ατύχημα στον δρόμο, μηχανική βλάβη, ασθένεια οδηγού) αφορούν το κομμάτι της διανομής της εταιρείας. Και στις τρεις περιπτώσεις συμβάντων, η εταιρεία αποφασίζει για αλλαγή πλάνου ή/και ανά δρομολόγηση, όταν αυτά τα δεδομένα εμφανιστούν στην μονάδα επεξεργασίας δεδομένων, και ειδικότερα στην μονάδα αποφάσεων δυναμικών δεδομένων.

2.3.3. Μονάδα Διαχείρισης Γεωγραφικών Δεδομένων Κίνησης

Τα δεδομένα κίνησης χωρίζονται και αυτά σε στατικά και δυναμικά. Τα στατικά δεδομένα κίνησης δίνονται από τους παρόχους χαρτών, ενώ η πρόβλεψη κίνησης μπορεί να γίνει είτε από τους παρόχους χαρτών είτε από το ίδιο το σύστημα, με βάση τα ιστορικά στοιχεία κίνησης των εταιριών των χαρτών, καταλήγοντας σε ένα χρονοδιάγραμμα κίνησης κάθε δρόμου. Με αυτόν τον τρόπο μπορεί να γίνει και η πρόβλεψη της κίνησης στους δρόμους, καθώς και πρόβλεψη του χρόνου μεταβίβασης από το ένα σημείο στο άλλο. Επιπρόσθετα,

πλέον ο πάροχος δίνει και την δυνατότητα για δυναμικά (πραγματικού χρόνου) δεδομένα κίνησης όπου ουσιαστικά χρησιμοποιεί δεδομένα από το GPS των συσκευών των χρηστών που το έχουν ενεργοποιημένο σε συνδυασμό με την εφαρμογή του παρόχου και στην συνέχεια αναλύει (π.χ. συσχετίζοντας δεδομένα που προέρχονται από κινητά που βρίσκονται στον ίδιο δρόμο) και υπολογίζει την ταχύτητα με την οποία κινούνται στην συγκεκριμένη περιοχή. Οι πιο γνωστοί πάροχοι είναι το «Google Maps» και το «HERE WeGo – Offline Maps & GPS».

2.3.4. Μονάδα Αποφάσεων (Στατικών Δεδομένων)

Όλα τα στατικά στοιχεία που αναλύθηκαν ανωτέρω, αποστέλλονται στο λογισμικό που παρέχεται από τον πάροχο λογισμικού μέσω του μοντέλου Λογισμικό ως Υπηρεσία. Αρχικά, γίνεται μια στατιστική εκτίμηση της κίνησης και στη συνέχεια προκύπτει ένα αρχικό πλάνο δρομολόγησης. Η δημιουργία αυτού του πλάνου δρομολόγησης είναι αποτέλεσμα της χρήσης αλγορίθμων για την επίλυση του προβλήματος δρομολόγησης οχημάτων με περιορισμούς παραθύρων χρόνου –Vehicle Routing Problems with Time Windows (VRPTW). Στόχος της αντικειμενικής συνάρτηση είναι η ελαχιστοποίηση του συνολικού κόστους της διαδρομής σύμφωνα με τους διάφορους περιορισμούς. Το πλάνος δρομολόγησης εμφανίζεται στην διεπαφή της συσκευής του οχήματος και ο οδηγός μπορεί να ξεκινήσει την διανομή.

2.3.5. Μονάδα Αποφάσεων (Δυναμικών Δεδομένων)

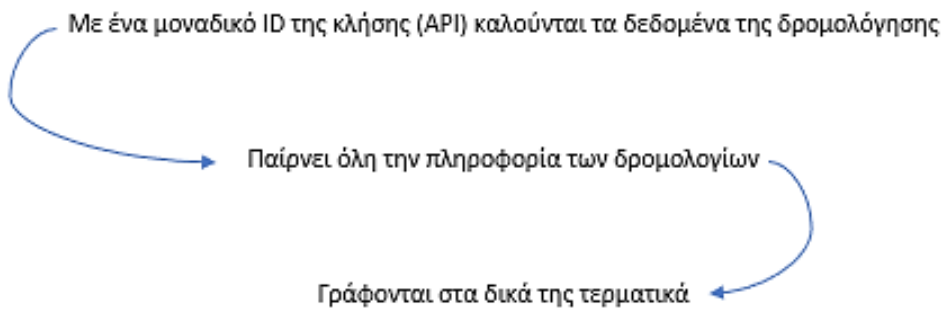
Στην μονάδα επεξεργασίας δεδομένων υπάρχει και η υπό μονάδα αποφάσεων που σχετίζεται με τα δυναμικά δεδομένα. Σε αυτή γίνεται συνεχής παρακολούθηση και ανίχνευση νέων ή/και τροποποιημένων πληροφοριών. Ανάλογα με τα δεδομένα που καταφτάνουν το λογισμικό με χρήση αλγορίθμου δυναμικής δρομολόγησης οχημάτων με περιορισμούς παραθύρων χρόνου – Dynamic Vehicle Routing Problems with Time Windows (DVRPTW), υλοποιεί είτε ανά δρομολόγηση, είτε αλλαγή του πλάνου δρομολόγησης, είτε καμία αλλαγή στο ήδη υπάρχον πλάνο. Τελικά, οποιαδήποτε απόφαση εμφανίζεται στην διεπαφή της συσκευής του οχήματος και ο οδηγός ακολουθεί τις οδηγίες της συσκευής, και στην διεπαφή της εταιρίας logistics.

2.3.6. Εξερχόμενα Δεδομένα και Παρακολούθηση

Η εταιρία logistics πρέπει να πάρει πίσω το πλάνο των παραδόσεων, το δρομολόγιο, το φορτηγό που θα κάνει την διαδρομή και τις ώρες παράδοσης, ώστε να βγάλει την σειρά φόρτωσης (ακολουθούν την μέθοδο Last In, First Out – LIFO).

Εάν τρέχει γρήγορα ο αλγόριθμος (< 1min), μπορούν άμεσα να σταλούν στην εταιρία και στο όχημα τα δρομολόγια.

Εάν η αποστολή των δεδομένων είναι ασύγχρονη, τότε μπορεί να γίνεται ένα αίτημα της απάντησης, δηλαδή να καλεί αργότερα και να του στέλνει πίσω τις διαδρομές. Αυτή η διαδικασία πραγματοποιείται με τον εξής τρόπο Σχήμα 2.3.



Σχήμα 2.3: Ασύγχρονή Διαδικασία Αποστολής Δεδομένων

Η οπτικοποίηση του αποτελέσματος, δηλαδή η εμφάνιση των τελικών καταστάσεων σε ένα τερματικό, είναι σχετικά εύκολη στην υλοποίηση και δωρεάν. Αντίθετα η παρακολούθηση, δηλαδή η εποπτεία σε πραγματικό χρόνο των καταστάσεων των παραδόσεων, των οχημάτων κ.λπ., πιθανότατα να συναντήσει προβλήματα με τον κανονισμό GDPR (Γενικό Κανονισμό Προστασία Δεδομένων). Η παρακολούθηση χρειάζεται συγκεκριμένη τεχνολογία (REST API) για να στέλνει στον διακομιστή τα δεδομένα. Επίσης, χρειάζεται την ίδια τεχνολογία στις συσκευές των μεταφορικών μέσων, για να αποστέλλονται οι διαδρομές και να τις βλέπουν. Η παρακολούθηση έχει σχέση με την τεχνολογία της διεπαφής του οχήματος, και είναι ένας τρόπος ώστε στην διεπαφή της εταιρίας logistics να εμφανίζονται όλα τα δεδομένα της δρομολόγησης που αφορούν την κατάσταση της διανομής των εμπορευμάτων.

2.3.7. Διεπαφή Οχήματος

Τα οχήματα κάνουν χρήση της νέας τεχνολογίας IoV, δηλαδή Internet of Vehicles. Με την ταχεία ανάπτυξη των υπολογιστικών τεχνολογιών και των επικοινωνιών, η νέα εποχή οδηγεί στην εξέλιξη των συμβατικών δικτύων Ad-hoc για οχήματα (VANET) στο διαδίκτυο των οχημάτων (IoV). Τα δύο κύρια χαρακτηριστικά του IoV είναι

Η δικτύωση των οχημάτων

Η ευφυία των οχημάτων

Η δικτύωση των οχημάτων αποτελείται από τη διασύνδεση VANET (καλούμενη επίσης διασύνδεση οχημάτων), την τηλεματική οχημάτων (επίσης καλούμενα συνδεδεμένα οχήματα) και το κινητό διαδίκτυο (το όχημα είναι ως τροχοφόρος κινητός τερματικός σταθμός). Η νοημοσύνη των οχημάτων θεωρεί ότι η ενσωμάτωση του οδηγού και του οχήματος ως ενότητας είναι πιο έξυπνη με τη χρήση τεχνολογιών δικτύων, η οποία αναφέρεται στη βαθιά μάθηση, τον υπολογιστικό υπολογισμό, τον υπολογισμό σμήνους, την αβεβαιότητα, την τεχνητή νοημοσύνη κλπ. (Fangchun, Shangguang, Jinglin, Zhihan, & Qibo, 2014)

Με αυτόν τον τρόπο τα οχήματα στην συγκεκριμένη αρχιτεκτονική δύναται να παρέχουν πληροφορίες για την κατάσταση τους, για τα γεωγραφικά δεδομένα κίνησης και για οποιαδήποτε άλλη πληροφορία έχει να κάνει με την διανομή των παραγγελιών.

3. Τεχνολογίες Ανάπτυξης και Διασύνδεσης

3.1. Εισαγωγή στις Τεχνολογίες Ανάπτυξης και Διασύνδεσης

Το κεφάλαιο 3 παρουσιάζει συγκεκριμένες τεχνολογίες για την ανάπτυξη και διασύνδεση των στοιχείων του συστήματος, με σκοπό σε κάθε ενότητα να προτείνεται η καταλληλότερη τεχνολογία για την αρχιτεκτονική του συστήματος δρομολόγησης. Όπως έγινε φανερό στο Κεφάλαιο 2, τα διάφορα μέρη του συστήματος επικοινωνούν μέσω του διαδικτύου χρησιμοποιώντας λογισμικά, εφαρμογές και υπηρεσίες Ιστού. Η λειτουργία του συστήματος επιτυγχάνεται όταν το Λογισμικού ως Υπηρεσία έχει οικοδομηθεί πάνω σε κατάλληλες τεχνολογίες ανάπτυξης και έπειτα στις τεχνολογίες διασύνδεσης αυτών. Συγκεκριμένα, το Back-end σύστημα υλοποιείται σε μια πλατφόρμα ανάπτυξης, όπου γίνεται χρήση συγκεκριμένης τεχνολογίας ανάπτυξης διαδικτυακών υπηρεσιών, που με την σειρά τους συλλέγουν τα δεδομένα από τις βάσεις δεδομένων. Η κλήση των υπηρεσιών γίνεται μέσω κάποιου πρωτοκόλλου, τα οποία δείχνουν με ποιο τρόπο κωδικοποιείται η επικοινωνία των συστημάτων. Η πρώτη ενότητα του κεφαλαίου επικεντρώνεται στην πλατφόρμα ανάπτυξης των εφαρμογών και υπηρεσιών του Λογισμικού, και εκτιμά τις γνωστότερες πλατφόρμες ανάπτυξης. Η δεύτερη ενότητα εξετάζει τα είδη των υπηρεσιών που αναπτύσσονται στις πλατφόρμες ανάπτυξης για την αλληλεπίδραση των εφαρμογών, και διαχωρίζει και αξιολογεί τις έννοιες των διαδικτυακών υπηρεσιών και των API's υπηρεσιών. Η τρίτη, και τελευταία ενότητα του κεφαλαίου, αφορά τα πρωτόκολλα επικοινωνίας και ανταλλαγής δεδομένων μεταξύ των εφαρμογών (SOAP, REST WSDL). Συγκεκριμένα, εξετάζεται η διαφοροποίηση των πρωτοκόλλων επικοινωνίας με το πρωτόκολλο εφαρμογών (HTTP) και αναλύονται τα στοιχεία κάθε πρωτοκόλλου επικοινωνίας ξεχωριστά.

Πίνακας 3.1: Τεχνολογίες Ανάπτυξης και Διασύνδεσης και οι Αντίστοιχες Προτάσεις για το Σύστημα

Είδος Τεχνολογίας	Τεχνολογίες
Πλατφόρμα ανάπτυξης	Microsoft.net J2EE
Διαδικτυακές υπηρεσίες	Web Services API's services
Πρωτόκολλα ανταλλαγής δεδομένων	SOAP WSDL REST

3.2. Πλατφόρμα Ανάπτυξης

3.2.1. Εισαγωγή στις Πλατφόρμες Ανάπτυξης

Σε αυτό το κεφάλαιο θα γίνει μια σύγκριση μεταξύ των δυο επιλογών που έχουν οι επιχειρήσεις για δημιουργία δικτυακών υπηρεσιών βασισμένων σε XML: η πλατφόρμα Java 2, Enterprise Edition (J2EE) που κατασκευάστηκε από την Sun Microsystems και άλλους φορείς της βιομηχανίας, και το Microsoft.NET, που κατασκευάστηκε από τη Microsoft Corporation. Για να ξεκινήσει η ανάλυση των επιμέρους τεχνολογιών αποτελεί επιτακτική ανάγκη να οριστεί το πλαίσιο στο οποίο δημιουργούνται οι εφαρμογές και ο λόγος για τον οποίο επιλέγονται κάθε φορά, οι διαφορετικές πλατφόρμες ανάπτυξης εφαρμογών. Οι περισσότεροι εγκαθιστούν χωρίς να γνωρίζουν επακριβώς τι είναι οι πλατφόρμες ανάπτυξης ώστε να χρησιμοποιήσουν εφαρμογές ενώ κάποιοι από αυτούς τις αναφέρουν ως framework. Αλλά τι είναι μια πλατφόρμα ανάπτυξης και ποια η διαφορά με ένα απλό framework ;

Ετυμολογικά η λέξη framework αποτελείται από τις λέξεις frame και work, δηλαδή πλαίσιο και εργασία. Οι έννοιες θα οριστούν χρησιμοποιώντας γνωστά παραδείγματα που εμφανίζονται καθημερινά. Αν σκεφτεί κανείς ένα πλαίσιο μιας φωτογραφίας, τι θα έπρεπε να συμβαίνει ώστε η φωτογραφία να τοποθετηθεί μέσα σε αυτό ; Το πλαίσιο θα πρέπει να έχει καθορισμένο σχήμα και μέγεθος, διαφορετικά δεν θα ταιριάζει στην φωτογραφία. Ή όταν χτίζεται ένα σπίτι είναι αναγκαίο αυτό να βρίσκεται μέσα σε ένα πλαίσιο, ώστε να γίνονται όλες οι εργασίες οικοδόμησης του μέσα σε αυτό. Με τον ίδιο τρόπο, καθορίζεται και το πλαίσιο ιστού, είναι δηλαδή μια συμφωνία όπου το λογισμικό προσφέρει μεγαλύτερη λειτουργικότητα, η οποία με την σειρά της μπορεί να επεκταθεί από επιπρόσθετους γραπτούς κώδικες του εκάστοτε χρήστη. Δίνει μια βασική δομή (λήψη πηγαίου κώδικα) γύρω από την οποία προσθέτοντας κώδικες και δημιουργώντας εφαρμογές, δίνει την δυνατότητα για μεγαλύτερη λειτουργικότητα του συστήματος. Βασικό χαρακτηριστικό ενός πλαισίου είναι η επαναχρησιμοποίηση των λειτουργιών, όταν ένας χρήστης επιθυμεί να χρησιμοποιήσει την ίδια λειτουργικότητα σε πολλά σημεία μέσα στο έργο του. Με αυτόν τον τρόπο απλοποιείται η διαδικασία δημιουργίας εφαρμογών ιστού και εξοικονομείται χρόνος, καθώς μειώνονται τα πράγματα στο οποία πρέπει να εστιάσει.

Από την άλλη πλευρά μια πλατφόρμα παρέχει τόσο το υλικό όσο και τα εργαλεία λογισμικού που απαιτούνται για την εκτέλεση μιας εφαρμογής, είτε πρόκειται για ένα αυτόνομο πρόγραμμα είτε για ένα πρόγραμμα που έχει χτιστεί πάνω σε ένα πλαίσιο. Κυρίως, δίνεται ως **Platform as a Service (6.7.2)** δηλαδή η βάση του κώδικα δεν δίνεται, αλλά αποτελεί μέρος

μια φιλοξενούμενης λύσης που εκτελείται σε ένα νέφος -cloud- (**6.6),** το οποίο προσεγγίζεται μέσω API ή GUI, έννοιες που θα αναλυθούν στο επόμενο κεφάλαιο. Η χρήση μιας πλατφόρμας συνήθως αποτελείται από μια διμερής δομή. Από την μία πλευρά υπάρχει η συγκεκριμένη φιλοξενούμενη εφαρμογή με τον κώδικα και από την άλλη υπάρχει η πλατφόρμα που εκτελείται στο νέφος , που λαμβάνει, υπολογίζει και επιστρέφει δεδομένα στην εφαρμογή. Βασικό στοιχείο και των δύο είναι ότι μέσω αυτών δίνεται η δυνατότητα ανάπτυξης διαδικτυακών εφαρμογών και υπηρεσιών ιστού (web services). Οι υπηρεσίες στο διαδίκτυο πρέπει να είναι αξιόπιστες, κλιμακούμενες και πρέπει να εκτελούνται σε αποδεκτά επίπεδα. Οι πιο γνωστές πλατφόρμες ανάπτυξης είναι η Microsoft.net platform και η Java 2 Platform, Enterprise Edition (J2EE), οι οποίες είναι εξελίξεις της υπάρχουσας τεχνολογίας Server εφαρμογών. (Vawter & Roman, 2001)

Η net. Πλατφόρμα είναι μια προσέγγιση της Microsoft, εξού και το όνομα της, για τις υπηρεσίες web. Εκτελείται σε μια ενιαία πλατφόρμα, συγκεκριμένα στα windows, αλλά υποστηρίζει εγγενώς πολλές γλώσσες. από την άλλη πλευρά πλατφόρμα της java, γνωστή και ως Java 2 Platform, Enterprise Edition (J2EE), είναι μια ανεξάρτητη του λειτουργικού συστήματος που αναπτύσσεται σε μια γλώσσα προγραμματισμού, αυτής της Java. Και οι δυο .Net και J2EE παρουσιάζουν διάφορες προκλήσεις και ευκαιρίες στο περιβάλλον ανάπτυξης εφαρμογών ιστού. Γι' αυτό το λόγο η πλατφόρμα που υιοθετείται, εξαρτάται από διάφορους παράγοντες, συμπεριλαμβανομένου του περιβάλλοντος ανάπτυξης, τις ανάγκες και του στόχους της επιχείρησης. (Miller, 2003)

3.2.2. Java 2 Platform, Enterprise Edition (J2EE)

Η πλατφόρμα Java 2 Platform, Enterprise Edition (J2EE) σχεδιάστηκε για να απλοποιήσει πολύπλοκα προβλήματα με την ανάπτυξη και τη διαχείριση πολύ-επίπεδων επιχειρησιακών λύσεων. Το J2EE είναι ένα πρότυπο και όχι ένα προϊόν. Για παράδειγμα Εάν υπάρχουν δυο εφαρμογές, για να μπορέσουν να τρέχουν θα πρέπει να υπακούν σε μια σύμβαση J2EE. Αποτελεί ιστορικά μια αρχιτεκτονική για την ανάπτυξη εφαρμογών διακομιστή και μπορεί να χρησιμοποιηθεί για την κατασκευή ιστοσελίδων, στοιχείων λογισμικού ή και πακέτων εφαρμογών. Πλέον έχει επεκταθεί και υποστηρίζει την κατασκευή υπηρεσιών ιστού που βασίζονται σε XML, με χαρακτηριστική την αλληλεπίδραση με άλλες υπηρεσίες που πιθανότατα να μην είναι γραμμένες με βάση το πρότυπο J2EE. (Vawter & Roman, 2001)

Η βάση της αρχιτεκτονικής για το J2EE είναι η γλώσσα προγραμματισμού Java , η οποία επιτρέπει στις επιχειρήσεις να γράψουν τον κώδικα τους σε μια γλώσσα μια φόρα και στη

συνέχεια να τον αναπτύξουν σε οποιαδήποτε πλατφόρμα. Τα βήματα για κάτι τέτοιο είναι τρία:

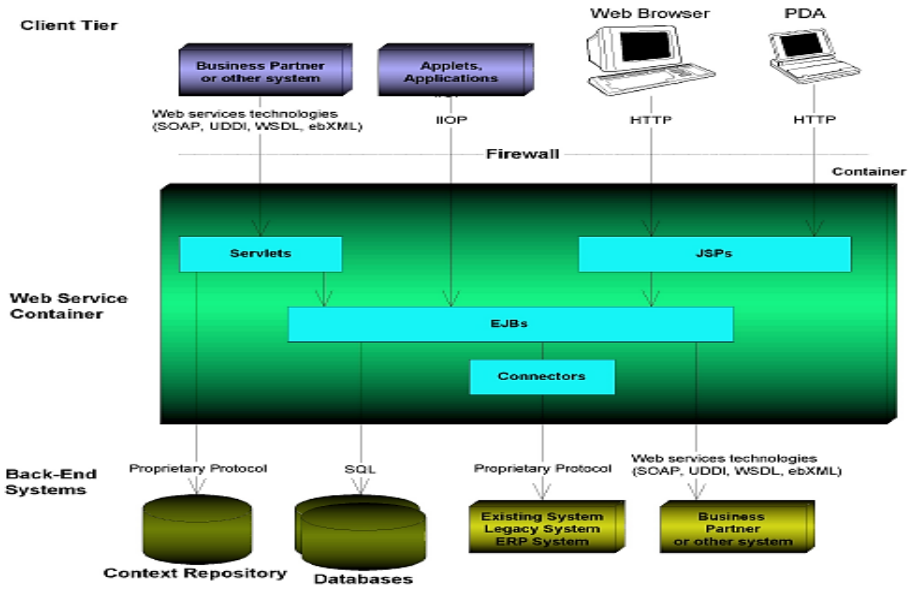
Α. Οι προγραμματιστές γράφουν τον κώδικα τους στην γλώσσα Java

Β. Ο κώδικας καταρτίζεται σε bytecode

Γ. Όταν ο κώδικα είναι έτοιμος να εκτελεστεί τότε το Java runtime Environment ερμηνεύει αυτό το bytecode και το εκτελεί. (Vawter & Roman, 2001)

Το Java Runtime Environment (JRE) είναι ένα σύνολο εργαλείων λογισμικού για την ανάπτυξη εφαρμογών Java. Συνδυάζει την Java Virtual Machine (JVM), τον πυρήνα πλατφόρμας και τις υποστηρικτικές βιβλιοθήκες. (Technopedia, Java Runtime Environment (JRE), 2019)

Το μοντέλο ανάπτυξης υπηρεσιών ιστού με το J2EE φαίνεται Σχήμα 3.1



Σχήμα 3.1:Το Μοντέλο Ανάπτυξης Υπηρεσιών Ιστού με το J2EE

Η εφαρμογή J2EE παρέχει τις αναγκαίες υπηρεσίες για την ανάπτυξη επιχειρηματικών υπηρεσιών, όπως είναι οι συναλλαγές, υπηρεσίες ασφάλειας και ειδικές υπηρεσίες. Το επιχειρηματικό επίπεδο εκτελεί την συλλογή και επεξεργασία δεδομένων, το οποίο συνδέεται με βάσεις δεδομένων ή συστήματα που προϋπάρχουν και χρησιμοποιούν την αρχιτεκτονική σύνδεσης της Java. Μπορεί επίσης να συνδεθεί με επιχειρήσεις που συνεργάζεται και χρησιμοποιούν τεχνολογίες υπηρεσιών ιστού (SOAP, UDDI, WSDL, ebXML) μέσω των API Java για XML. Τέλος, τα προγράμματα περιήγησης στο διαδίκτυο και οι ασύρματες συσκευές συνδέονται με σελίδες JavaServer (JSPs), οι οποίες αποδίδουν διεπαφές χρήστη. (Vawter & Roman, 2001)

Πρέπει να προστεθεί ότι οι προμηθευτές που προσφέρουν πλατφόρμες J2EE δίνουν κάποιες επιπλέον λειτουργίες που κανονικά δεν υπάρχουν στο πρότυπο J2EE και που ενδεχομένως επηρεάζουν τη φορητότητα, όπως είναι τα στοιχεία ηλεκτρονικού εμπορίου ή και η ενσωμάτωση B2B. Ωστόσο, υπάρχουν χαρακτηριστικά που δεν επηρεάζουν την φορητότητα του κώδικα εφαρμογής καθώς βρίσκονται στο Back end σύστημα. Τέτοιες είναι η εξισορρόπηση του φορτίου και η προσωρινή αποθήκευση δεδομένων. (Vawter & Roman, 2001)

3.2.3. Microsoft.NET Platform

Η Microsoft.NET πλατφόρμα είναι μια οικογένεια προϊόντων που επιτρέπει στους οργανισμούς να δημιουργούν έξυπνες υπηρεσίες ιστού επιχειρηματικής κλάσης. Για χάριν ευκολίας θα αναφέρεται ως .NET, όπως είναι και ευρέως γνωστό στην καθομιλουμένη ως dot net. Η σημαντική διαφορά ανάμεσα στην .NET και στην J2EE είναι το γεγονός ότι το ένα όπως αναφέρθηκε και παραπάνω είναι ένα πρότυπο στο οποίο τα προϊόντα είναι γραμμένα, ενώ η .NET είναι ένα πακέτο προϊόντων.

Αν κάποιος παρατηρούσε το DNA της .NET θα έβλεπε ότι είναι σε μεγάλο βαθμό μια επανεγγραφή των Windows, το οποίο ήταν ο πρόγονος της πλατφόρμας της Microsoft για ανάπτυξη εφαρμογών για επιχειρήσεις. Αυτό το DNA περιλάμβανε τις εξής τεχνολογίες.

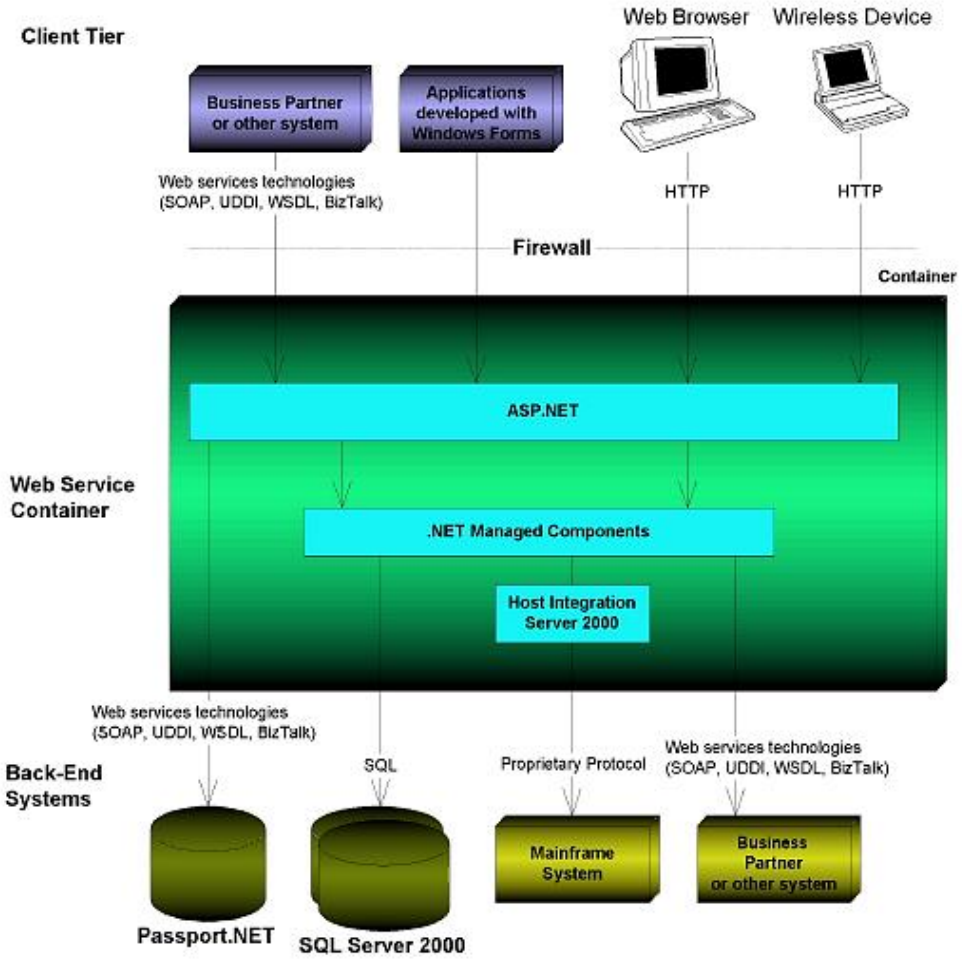
Microsoft Transaction Server (MTS) και του COM +,

Microsoft Message Queue (MSMQ)

Βάση δεδομένων Microsoft SQL Server

Πλέον, όμως, το .NET Framework αντικαθιστά αυτές τις τεχνολογίες και περιλαμβάνει ένα στρώμα υπηρεσιών ιστού καθώς και βελτιωμένη υποστήριξη γλώσσας.

Το μοντέλο ανάπτυξης υπηρεσιών ιστού με την .NET φαίνεται στο Σχήμα 3.2



Σχήμα 3.2:Το Μοντέλο Ανάπτυξης Υπηρεσιών Ιστού με την .NET

Η εφαρμογή .NET παρέχει τις απαραίτητες υπηρεσίες για τις επιχειρησιακές εφαρμογές, όπως υπηρεσίες συναλλαγών, ασφάλειας και μηνυμάτων. Το επιχειρηματικό στρώμα της εφαρμογής .NET κατασκευάζεται με τη χρήση διαχειριστικών στοιχείων .NET, με το οποίο εκτελούνται η συλλογή και επεξεργασία των δεδομένων. Αυτό το επίπεδο συνδέεται με βάσεις δεδομένων χρησιμοποιώντας Active Data Objects (ADO.NET) και τα υπάρχοντα συστήματα που χρησιμοποιούν υπηρεσίες που παρέχονται από το Microsoft Host Integration Server 2000. Οι συνεργάτες της επιχείρησης μπορούν να συνδεθούν είτε με το επιχειρηματικό επίπεδο είτε με την εφαρμογή .NET μέσω τεχνολογιών υπηρεσιών ιστού. Οι ασύρματες συσκευές συνδέονται με Active Server Pages (ASP.NET), οι οποίες αποδίδουν διεπαφές χρήστη σε HTML, XHTML ή WML. (Vawter & Roman, 2001)

Το βασικότερο χαρακτηριστικό της πλατφόρμας .NET και μια από τις πιο θεμελιώδεις πτυχές της είναι ότι προσφέρει ανεξαρτησία και δια λειτουργικότητα γλώσσας. ένα παράδειγμα για να γίνει πιο κατανοητό αυτό που λέγεται ως ανεξαρτησία γλώσσας είναι το παρακάτω.

Ένα στοιχείο .NET μπορεί να γραφτεί εν μέρει στην VB.NET, στην έκδοση .NET της Visual Basic (γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (EVENT driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM (Visual Basic, 2019)) και στην συνέχεια σε C#, στην νέα γλώσσα προγραμματισμού που καθορίζεται από αντικειμενικά στοιχεία της Microsoft.

Τα βήματα για κάτι τέτοιο είναι τα εξής :

1. Ο πηγαίος κώδικας μεταφράζεται στην ενδιάμεση γλώσσα MSIL ή IL, έτσι ο κώδικας πλέον είναι ουδέτερος σε γλώσσα και είναι αντίστοιχος του bytecode της Java.
2. Ο κώδικας MSIL ερμηνεύεται και μεταφράζεται σε ένα εγγενές εκτελέσιμο αρχείο. Το .NET πλαίσιο περιλαμβάνει το Common Language Runtime (CLR), αντίστοιχο του JRE. Το CLR είναι ο διαμεσολαβητής της Microsoft.
3. Ο .NET κώδικας τελικά τρέχει μέσα στο CLR.

3.2.4. ASP.NET

Υπάρχει μία παρεξήγηση ότι το ASP.NET και το .NET είναι το ίδιο. Το .NET όπως αναφέρθηκε είναι ένα λογισμικό πλαίσιο ή η υποδομή που αναπτύχθηκε από τη Microsoft. Το ASP.NET από την άλλη πλευρά είναι μία εφαρμογή που χρησιμοποιείται για την κατασκευή διαφόρων άλλων εφαρμογών.

Βασική διαφορά και ιστορία

Το ASP.NET ένα πλαίσιο εφαρμογών ιστού ανοιχτού κώδικα που βοηθά στην δημιουργία δυναμικών ιστοσελίδων. Αναπτύχθηκε από την Microsoft ούτως ώστε οι εφαρμογές με δυναμικές ιστοσελίδες να μπορούν να δημιουργηθούν εύκολα. Βοηθάει. Επίσης. στην οικοδόμηση καλύτερων εφαρμογών ιστού και υπηρεσιών διαδίκτυο. Δημιουργήθηκε το 2002, είναι βασισμένο στην Common Language Runtime (CLR) και υποστηρίζει οποιαδήποτε γλώσσα .NET.

Το .NET πρόκειται για ένα πλαίσιο λογισμικού που αναπτύχθηκε από τη Microsoft. Υποστηρίζει διάφορες γλώσσες και παρέχει περιβάλλον για διάφορες γλώσσες.(3.2.3)

Αρχιτεκτονική και μοντέλα προγραμματισμού

ASP.NET είναι μία εφαρμογή που παρέχει μία σειρά μοντέλων προγραμματισμού που διευκολύνουν τη διαδικασία δημιουργίας μιας εφαρμογής του.

Κάποια από αυτά τα μοντέλα είναι:

- ASP.NET Web Forms

- ASP.NET MVC
- SP.NET Web pages
- ASP.NET Web API
- ASP.NET Webhooks
- SignalR

Το .NET έχει μια καθορισμένη αρχιτεκτονική με στοιχεία όπως Common Language Infrastructure (CLI), Common Language Runtime (CLR), συναρμολογήσεις, βιβλιοθήκη κλάσεων, μοντέλα εφαρμογών.

Το CLI παρέχει μια πλατφόρμα για την ανάπτυξη εφαρμογών και όλες τις λειτουργίες που απαιτούνται για την ίδια. Το CLR είναι μια μηχανή εκτέλεσης με υπηρεσίες που προσφέρουν υπηρεσίες χειρισμού εξαιρέσεων, συλλογής κλπ. Η βιβλιοθήκη παρέχει χώρους ονομάτων και πολλά API με λειτουργίες όπως ανάγνωση, γραφή, χειρισμός εγγράφων XML κλπ. Επίσης παρέχει C++, που είναι μια γλώσσα που χρησιμοποιείται για την κατάρτιση προγραμμάτων και την εκτέλεση αυτών μέσα στο δεδομένο πλαίσιο. (EDUBCA, 2019)

3.2.5. Ομοιότητες και Διαφορές των J2EE και .NET

Για να γίνουν πιο κατανοητά τα δύο μοντέλο στον Πίνακας 3.2 παρουσιάζονται οι ομοιότητες και αναλογίες τους.

Πίνακας 3.2: Ομοιότητες και Αναλογίες J2EE και .NET

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	J2EE	.NET
Είδος τεχνολογίας	Πρότυπο	Προϊόν
Ενδιάμεσοι Προμηθευτές	30+	Microsoft
Διερμηνέας	JRE	CLR
Δυναμικές Ιστοσελίδες	JSP	ASP.NET
Πρόσβαση σε βάσεις δεδομένων	JDBC, SQL/J	ADO.NET
SOAP, WSDL, UDDI	Υποστηρίζει	Υποστηρίζει

Ωστόσο δεν λείπουν και οι διαφορές που υπάρχουν ανάμεσα σε αυτές τις δύο πλατφόρμες. Παρακάτω παρατίθενται τα επιχειρήματα υπέρ της .NET και της J2EE, και η επιλογή για την συγκεκριμένη αρχιτεκτονική του συστήματος.

Επιχειρήματα υπέρ του .NET

1. Το εμπορεύεται αποκλειστικά η Microsoft
2. Είναι η πρώτη που κυκλοφόρησε στοιχεία για της διαδικτυακές υπηρεσίες
3. Έχει πολύ καλύτερα εργαλεία (Visual Studio.NET)
4. Έχει ένα απλούστερο μοντέλο προγραμματισμού, που επιτρέπει στους προγραμματιστές που μπορεί να είναι και απλοί εργαζόμενοι σε μια εταιρία, να είναι παραγωγικοί χωρίς να καταστρέψουν ό,τι έχουν φτιάξει.
5. Παρέχει γλωσσική ουδετερότητα κατά την ανάπτυξη νέων εφαρμογών eBusiness, ενώ το J2EE αντιμετωπίζει τις άλλες γλώσσες ως ξεχωριστές εφαρμογές
6. Το .NET επωφελείται από την ισχυρή αλληλεπίδρασή του με το υποκείμενο λειτουργικό σύστημα
7. Παρέχει σημαντική ασφάλεια δεδομένων σε σχέση με το J2EE

Επιχειρήματα υπέρ του J2EE

1. Διατίθεται στο εμπόριο από ολόκληρο τον κλάδο
2. Αποτελεί μία αποδεδειγμένη πλατφόρμα, με μερικά νέα API υπηρεσιών Web. Το .NET είναι μια επανεγγραφή και εισάγει τον κίνδυνο όπως και με οποιαδήποτε άλλη τεχνολογία πρώτης γενιάς
3. Δίνει ουδετερότητα πλατφόρμας, επιτρέπει την χρήση οποιουδήποτε λειτουργικού συστήματος, όπως τα Windows, το UNIX ή το mainframe. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το περιβάλλον στο οποίο είναι πιο παραγωγικοί. (Vawter & Roman, 2001)

Για το υπάρχον σύστημα, λαμβάνοντας υπόψην όλα τα παραπάνω η καταλληλότερη πλατφόρμα ανάπτυξης είναι η .NET, η οποία δίνει την δυνατότητα στους προγραμματιστές να αναπτύσσουν με μεγαλύτερη ευκολία τις εφαρμογές τους, είτε λόγω της γλωσσικής ουδετερότητας είτε λόγω του απλούστερου μοντέλου προγραμματισμού. Επίσης, με τη χρήση της .NET προσδίδεται μεγαλύτερη αξιοπιστία και ασφάλεια ανάμεσα στους πελάτες και στις επιχειρήσεις, σημαντικό γεγονός δεδομένου ότι βρίσκεται σε εφαρμογή ο κανονισμός (ΕΕ) 2016/679 του Ευρωπαϊκού Κοινοβουλίου και του Συμβουλίου της 27ης Απριλίου 2016 για την προστασία των φυσικών προσώπων έναντι της επεξεργασίας των δεδομένων προσωπικού χαρακτήρα και για την ελεύθερη κυκλοφορία των δεδομένων αυτών και την κατάργηση της οδηγίας 95/46/ΕΚ.

3.3. Διαδικτυακές Υπηρεσίες (Web Services)

3.3.1. Εισαγωγή στις Διαδικτυακές Υπηρεσίες

Στην προηγούμενη ενότητα αναλύθηκαν τα διάφορα είδη πλατφόρμας που συναντά κανείς για την ανάπτυξη διαδικτυακών υπηρεσιών, στα αγγλικά web services. Περιληπτικά, οι διαδικτυακές υπηρεσίες είναι ένα νέο υποσύνολο εφαρμογών του παγκόσμιου Ιστού, αποτελούν ανεξάρτητα στοιχεία οποιασδήποτε εφαρμογής και δημοσιεύονται στον Διαδίκτυο με τέτοιο τρόπο ώστε άλλες διαδικτυακές υπηρεσίες να τα βρίσκουν και να τα χρησιμοποιούν προς εξυπηρέτηση του δικού τους στόχου. Οι διαδικτυακές υπηρεσίες χρησιμοποιούνται καθημερινά από τον άνθρωπο, χαρακτηριστικά παραδείγματα αποτελούν, η υπηρεσία πιστωτικών καρτών που επεξεργάζεται συναλλαγές με πιστωτικές κάρτες για έναν συγκεκριμένο αριθμό λογαριασμού, μια αεροπορική υπηρεσία που παρέχει το πρόγραμμα πτήσεων, τη διαθεσιμότητα και τις λειτουργίες κράτησης, και μια υπηρεσία δεδομένων αγοράς των χρηματιστηρίων. Όπως είναι εμφανές, από τις πιο απλές έως και τις πιο πολύπλοκες, οι υπηρεσίες Ιστού υπόσχονται ευελιξία και διευκόλυνση στην χρήση του Διαδικτύου για περαιτέρω σκοπούς, γι' αυτό το λόγο μεγάλες εταιρίες (IBM, Microsoft, Sun κ.α.) έχουν επενδύσει σε μεγάλο βαθμό (Roy & Ramanujan, 2001). Σκοπός του παρόντος κεφαλαίου είναι η ανάλυση και κατανόηση της έννοιας των διαδικτυακών υπηρεσιών.

3.3.2. Προέλευση Διαδικτυακών Υπηρεσιών

Οι διαδικτυακές υπηρεσίες αποτελούν συνέπεια της φυσικής εξέλιξης της τεχνολογίας της πληροφορικής και ειδικότερα του παγκόσμιου Ιστού, κρατώντας ωστόσο τον θεμελιώδη χαρακτήρα του Ιστού ως μέσο που επιτρέπει την αλληλεπίδραση του ανθρώπου με τις εφαρμογές. Στην αρχή ο Ιστός υπήρξε ο τρόπος διαμοιρασμού και διανομής των πληροφοριών σε παγκόσμια κλίμακα, στη συνέχεια εξελίχθηκε σε μια γιγαντιαίων διαστάσεων βιβλιοθήκη κατανεμημένων υπολογιστικών συστημάτων, κατάφερε να διευρύνει την χρήση του , επιτρέποντας ακόμα πιο εξελιγμένες μορφές αλληλεπίδρασης ανάμεσα στους διακομιστές και στα προγράμματα περιήγησης. Φτάνοντας στο σήμερα με τις διαδικτυακές εφαρμογές έχει αποκτήσει την ικανότητα να αλληλοεπιδρούν και οι εφαρμογές μεταξύ τους, βασισμένες πάντα στο Διαδίκτυο, έτσι αναπτύσσονται οι ηλεκτρονικές αγορές, η αυτοματοποίηση των συναλλαγών και η κοινή χρήση των πόρων σε μεγάλη κλίμακα. Κάποιος θα μπορούσε να αναρωτηθεί για το πως γίνεται όλα τα παραπάνω να υλοποιηθούν καθώς στο διαδίκτυο από την φύση του συνυπάρχουν πολλές πλατφόρμες και μοντέλα προγραμματισμού. Η επίτευξη αυτών γίνεται με την υποστήριξη ενός μοντέλου διαδικτυακών υπηρεσιών, που προσθέτει ευελιξία χωρίς την χρήση των υπαρχόντων μοντέλων. (Curbera, Nagy, & Weerawarana, 2001)

Από την μια πλευρά, οι διαδικτυακές υπηρεσίες χρειάζονται ελάχιστο σύνολο προτύπων για τη διασφάλιση της βασικής δια λειτουργικότητας, και από την άλλη, παρέχουν μια ενιαία άποψη των εφαρμογών σε ετερογενή συστήματα, δίνοντας με αυτόν το τρόπο την πρόσβαση στην εκμετάλλευση εναλλακτικών μοντέλων αλληλεπίδρασης όταν αυτές είναι διαθέσιμες. Επομένως, οι διαδικτυακές υπηρεσίες έχουν διττή μορφή όσον αφορά το πρόβλημα της ετερογένειας. (Curbera, Nagy, & Weerawarana, 2001)

3.3.3. Προσδιορισμός Διαδικτυακών Υπηρεσιών

Στη διεθνή βιβλιογραφία αναφέρονται διάφοροι ορισμοί για τις διαδικτυακές υπηρεσίες, ωστόσο παρατηρείται μεγάλη ομοιότητα μεταξύ αυτών. Ο πρώτος ορισμός που συναντά κανείς είναι ο εξής :

«Μια υπηρεσία Web είναι μια υπηρεσία λογισμικού που χρησιμοποιείται για την επικοινωνία μεταξύ δύο συσκευών σε ένα δίκτυο. Πιο συγκεκριμένα, μια υπηρεσία Web είναι μια εφαρμογή λογισμικού με τυποποιημένο τρόπο παροχής δια λειτουργικότητας μεταξύ διαφορετικών εφαρμογών. Αυτό γίνεται μέσω HTTP χρησιμοποιώντας τεχνολογίες όπως XML, SOAP, WSDL και UDDI» (Technopedia, Web Service, 2019)

Ένας άλλος είναι

«Μια υπηρεσία Web είναι μια δικτυωμένη εφαρμογή που είναι σε θέση να αλληλοεπιδράσει χρησιμοποιώντας τυπικά πρωτόκολλα Web εφαρμογής για εφαρμογές σε καλά καθορισμένες διεπαφές και η οποία περιγράφεται χρησιμοποιώντας μια τυπική γλώσσα περιγραφής λειτουργιών.» (Curbera, Nagy, & Weerawarana, 2001)

Διακρίνονται και στους δύο ορισμούς τα βασικά στοιχεία των διαδικτυακών υπηρεσιών, δηλαδή:

- Δια λειτουργικότητα
- Ισχυρή εξάρτηση από τα πρότυπα
- Κοινή παρουσίαση

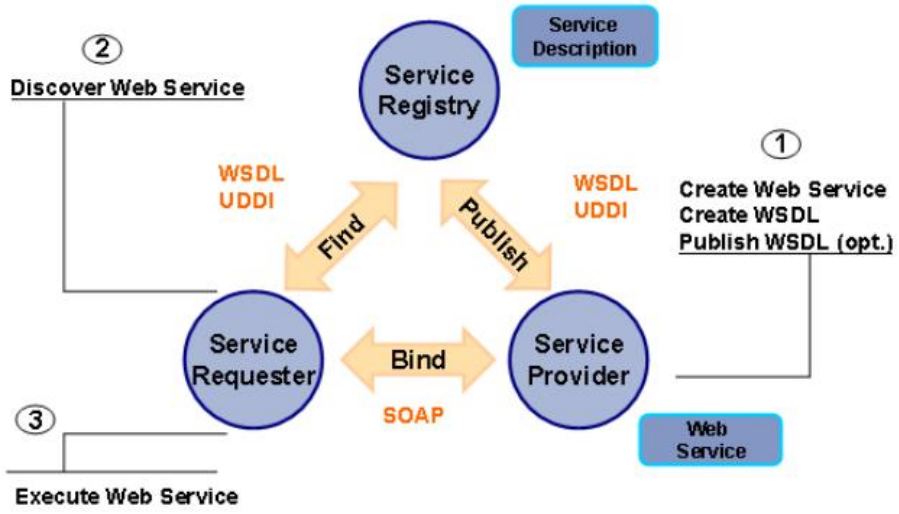
Βασικός στόχος των υπηρεσιών Ιστού είναι να παρέχουν μια κοινή παρουσίαση των εφαρμογών, που ενδεχομένων να χρησιμοποιούν διάφορα πρωτόκολλα επικοινωνίας και μοντέλα αλληλεπίδρασης, επιτρέποντας τες συγχρόνως να επωφελούνται από πιο αποτελεσματικά πρωτόκολλα ανταλλαγής δεδομένων όταν είναι διαθέσιμα και στα δύο αλληλοεπιδρώμενα μέρη.

Οι διαδικτυακές υπηρεσίες απαρτίζονται από δύο βασικά στοιχεία

1. Πρωτόκολλό ανταλλαγής δεδομένων (όπως θα αναφερθούν στο επόμενο κεφάλαιο)
2. Σύνολο μετά δεδομένων για την λειτουργική αναπαράσταση των διαδικτυακών εφαρμογών.

3.3.4. Αρχιτεκτονική Διαδικτυακών Υπηρεσιών

Μια τυπική αρχιτεκτονική διαδικτυακών υπηρεσιών αποτελείται από τρείς οντότητες: από τους παρόχους υπηρεσιών που τις δημιουργούν και τις δημοσιεύουν στο Διαδίκτυο, από τους μεσάζοντες που διατηρούν μητρώο δημοσιευμένων υπηρεσιών και από τους αιτούντες των υπηρεσιών, οι οποίοι βρίσκουν τις υπηρεσίες αναζητώντας το μητρώο του μεσάζοντα και δεσμεύουν τις εφαρμογές τους στον πάροχο υπηρεσιών ώστε να χρησιμοποιούν συγκεκριμένες υπηρεσίες (Roy & Ramanujan, 2001). Όπως φαίνεται στο Σχήμα 3.3



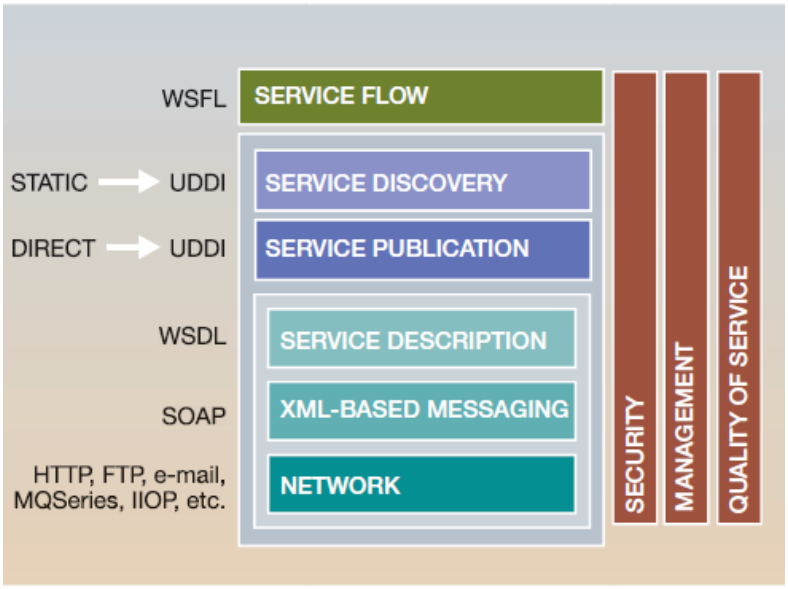
Σχήμα 3.3: Αρχιτεκτονική Διαδικτυακών Υπηρεσιών

Η υπηρεσία δημοσιεύεται με ένα μητρώο υπηρεσιών που βασίζεται σε ένα πρότυπο που ονομάζεται UDDI, ακρωνύμιο των λέξεων “Universal Description, Discovery, and Integration”. Το μητρώο UDDI παρέχει στον αιτούντα της υπηρεσίας μια περιγραφή υπηρεσίας WSDL και μια διεύθυνση URL (ομοιόμορφος ανιχνευτής πόρων) δείχνοντας την ίδια την υπηρεσία. Ορίζει ένα ηλεκτρονικό μητρώο επιχειρήσεων όπου οι επιχειρήσεις μπορούν να περιγράψουν την επιχείρησή τους και να καταχωρήσουν τις υπηρεσίες τους στο Web καθώς και να ανακαλύψουν και να ενσωματωθούν με άλλες επιχειρήσεις που προσφέρουν υπηρεσίες Web. Στη συνέχεια, ο ίδιος μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για να συνδέσει άμεσα με την υπηρεσία και να την επικαλεστεί. (Kreger, 2002) (Roy & Ramanujan, 2001). Οι λειτουργίες του UDDI συνοψίζονται στον Πίνακας 3.3

Πίνακας 3.3: Λειτουργίες του UDDI

Πληροφορία	Λειτουργίες
White Pages: Πληροφορίες όπως το όνομα, η διεύθυνση, το τηλέφωνο και άλλα στοιχεία επικοινωνίας για μια επιχείρηση.	Publish: Πώς ο προμηθευτής μιας WS καταχωρεί τον εαυτό του.
Yellow Pages: Πληροφορίες που κατηγοριοποιούν επιχειρήσεις, με βάση υπάρχοντα πρότυπα (μη ηλεκτρονικά).	Find: Πώς μια εφαρμογή βρίσκει συγκεκριμένη Υπηρεσία Ιστού.
Green Pages: Τεχνικές πληροφορίες που παρέχονται από μια επιχείρηση για τις Υπηρεσίες Ιστού	Bind: Πώς μια εφαρμογή συνδέεται και αλληλοεπιδρά με μια Υπηρεσία Ιστού, αφότου αυτή βρεθεί

Για να υλοποιηθούν τα άνωθεν, αποτελεί επιτακτική ανάγκη η ύπαρξη των λεγόμενων στοιβών προγραμματισμού υπηρεσιών Web ή αλλιώς Web Services programming stacks. Συγκεκριμένα είναι μια συλλογή τυποποιημένων πρωτοκόλλων και διεπαφών προγραμματισμού εφαρμογών (API) που επιτρέπει σε άτομα και εφαρμογές να εντοπίζουν και να χρησιμοποιούν υπηρεσίες Web. (Kreger, 2002). Τα σημαντικότερα οφέλη που προσφέρουν, είναι ότι επιτρέπουν την δημιουργία ενός περιβάλλοντος τοπικής ανάπτυξης. Με αυτόν τον τρόπο κάθε υπολογιστής μπορεί να λειτουργεί ως διακομιστής, όπου αναπτύσσονται τοπικές εφαρμογές και μετέπειτα τις εκτελούν και παραχωρούν πρόσβαση σε άτομα εκτός του δικτύου. Επιπλέον, απλοποιούν την διαδικασία εγκατάστασης, εξοικονομώντας χρόνο και διευκολύνοντας τον προγραμματιστή. (Litwin, 2017) Η στοίβα αποτελείται από διάφορα στρώματα που εξυπηρετούν τον σκοπό της (Σχήμα 3.4). Το σημαντικό σε κάθε στρώμα της στοίβας προγραμματισμού υπηρεσιών Web είναι η τυποποίηση απλών, ανοικτών πρωτοκόλλων και API. (Kreger, 2002)



Σχήμα 3.4: Βαθμίδες Στοιβάδας Προγραμματισμού Υπηρεσιών Web

Η θεμελιώδης βαθμίδα (από κάτω προς τα πάνω) είναι το διαδίκτυο (Network), καθώς όλες οι διαδικτυακές υπηρεσίες, όπως λέει και το όνομα τους, πρέπει να είναι διαθέσιμες σε κάποιο δίκτυο. Συχνά το διαδίκτυο βασίζεται στο πρωτόκολλο HTML, παρ' όλα αυτά υπάρχουν και άλλα είδη πρωτοκόλλων που χρησιμοποιούνται, όπως το Inter-ORB**Protocol (IIOP**) ή το IBM MQSeries*. (Kreger, 2002)

Το αμέσως επόμενο στρώμα είναι ένα στρώμα μηνυμάτων XML, που έχει ως σκοπό την διευκόλυνση της επικοινωνίας ανάμεσα στις διαδικτυακές υπηρεσίες και στους πελάτες τους. Βασίζεται στο SOAP, δηλαδή σε ένα πρωτόκολλο που διευκολύνει την δημοσίευση, την εύρεση, την σύνδεση και την κλήση των λειτουργιών. (Kreger, 2002)

Στη συνέχεια βρίσκεται το WSDL, μια προδιαγραφή που περιγράφει την διαδικτυακή υπηρεσία. Έχουν την μορφή εγγράφων XML.

Τα δύο επόμενα επίπεδα αφορούν την δημοσίευση και την εύρεση της υπηρεσίας. από την μία η δημοσίευση αποτελεί ενέργεια του παρόχου, όπου καθιστά το έγγραφο WSDL διαθέσιμο. Κάτι τέτοιο πραγματοποιείται με την αποστολή του WSDL (ή ενός δείκτη URL στο WSDL) ως μήνυμα ηλεκτρονικού ταχυδρομείου σε έναν προγραμματιστή. Η δημοσίευση διαφημίζει επίσης το WSDL σε ένα μητρώο UDDI για πολλούς προγραμματιστές ή για την εκτέλεση υπηρεσιών προς εύρεση. από την άλλη, η εύρεση μιας υπηρεσίας γίνεται από τον αιτούντα που έχει πρόσβαση στο WSDL αυτής. Η ενέργεια μπορεί να είναι τόσο απλή όσο η πρόσβαση σε ένα αρχείο ή μια διεύθυνση URL που περιέχει το WSDL ή είναι τόσο περίπλοκη όσο η αναζήτηση ενός μητρώου UDDI και η χρήση των αρχείων WSDL ώστε να επιλέξει μία από πολλές πιθανές υπηρεσίες.

Στην κορυφή βρίσκεται το επίπεδο ροής υπηρεσιών, όπου η συνάθροιση των διαδικτυακών υπηρεσιών διαμορφώνεται σε ροές εργασίας και η αναπαράσταση αυτής της συνάθροισης εμφανίζεται ως διαδικτυακή υπηρεσία ανώτερου επιπέδου. Η δραστηριότητα τυποποίησης σε αυτό το επίπεδο βρίσκεται σε εξέλιξη, αλλά η IBM έχει δημιουργήσει τη Γλώσσα ροής υπηρεσιών ιστού (WSFL) ως συνεισφορά της στην τυποποίηση. (Kreger, 2002).

Αυτά αποτελούν τα οριζόντια επίπεδα της στοιβάδας, ωστόσο όπως φαίνεται στο Σχήμα 3.4 υπάρχουν και άλλα τρία κάθετα επίπεδα, που πρέπει να αντιμετωπίζονται, αν και ανεξάρτητα κάθε φορά, σε κάθε στρώμα της στοίβας. Αυτά έχουν σχέση με τις αυστηρές απαιτήσεις των σημερινών επιχειρήσεων, για τις οποίες αποτελεί επιτακτική ανάγκη η παροχή μιας υποδομής επιχειρησιακού επιπέδου, δηλαδή ασφάλειας, διαχείρισης και ποιότητα εξυπηρέτησης της διαχείρισης.

3.3.5. Πλεονεκτήματα Διαδικτυακών Υπηρεσιών

Υπάρχουν πολλά πλεονεκτήματα χρήσης των διαδικτυακών υπηρεσιών

- Δια λειτουργικότητα: Ένα από τα πλεονεκτήματα της διαδικτυακής υπηρεσίας είναι η δια λειτουργικότητα. Οι υπηρεσίες Web επιτρέπουν σε εφαρμογές να επικοινωνούν, να ανταλλάσσουν δεδομένα και να ανταλλάσσουν υπηρεσίες μεταξύ τους. Οι κοινές μέθοδοι επικοινωνίας βασισμένες σε πρότυπα που έχουν αναπτυχθεί, καθιστούν δυνατή την ανεξαρτησία της διαδικτυακής υπηρεσίας από την πλατφόρμα.
- Ευχρηστία: Οι υπηρεσίες Web έχουν σχεδιαστεί για δύο σκοπούς, για αποστολή αιτημάτων ιστότοπου και για λήψη δεδομένων. Οι υπηρεσίες Web είναι οι ίδιες. Η δυνατότητα των υπηρεσιών διαδικτύου ποικίλλει από την απλή αναζήτηση πληροφοριών σε περίπλοκους αλγοριθμικούς υπολογισμούς. Γι' αυτό μπορεί να χρησιμοποιηθεί εύκολα.
- Επαναχρησιμοποίηση: Οι Υπηρεσίες Web έχουν σχεδιαστεί ώστε να συνδυάζονται για να προσφέρουν υπηρεσίες προστιθέμενης αξίας. Χρησιμεύουν ως δομικές μονάδες για την εύκολη επαναχρησιμοποίηση των στοιχείων της υπηρεσίας Web σε άλλες υπηρεσίες. Επίσης, οι εφαρμογές παλαιού τύπου μπορούν να ενσωματωθούν σε υπηρεσίες ιστού που θα χρησιμοποιηθούν από άλλους
- Ανάπτυξη: Οι υπηρεσίες Web αναπτύσσονται μέσω προτύπων Διαδικτύου όπως το Standard Apache, Axis2 για την παροχή υπηρεσιών HTTP, WSDL. Αυτό καθιστά απλή την ανάπτυξη.

- Κόστος: Το κόστος μειώνεται λόγω των νέων συστημάτων που συναρμολογούνται από τα πακέτα διαδικτυακών υπηρεσιών. Το αποθηκευμένο κόστος μπορεί να αποτελέσει όφελος τόσο για τον πάροχο όσο και για τον πελάτη. Επιπλέον, επιτυγχάνεται καλύτερη απόδοση. (Curbera, Nagy, & Weerawarana, 2001) (Kreger, 2002) (Resources, 2018)

3.3.6. Υπηρεσίες API

Στην συγκεκριμένη παράγραφο θα αναφερθούν οι διαφορές μεταξύ API's και Web Services, δύο έννοιες που συχνά συγχέονται, ωστόσο διαφέρουν αρκετά. Επομένως, θα απαντηθούν οι ερωτήσεις σχετικά με το τι είναι ένα API και ποια η αναλογία μεταξύ API και Web Services.

Η ανάλυση θα ξεκινήσει με μια πολύ απλή αναλογία, όλες οι τίγρεις είναι γάτες αλλά όλες οι γάτες δεν είναι τίγρεις. Με παρόμοιο τρόπο, όλες οι υπηρεσίες ιστού είναι API, αλλά δεν είναι όλα τα API υπηρεσίες ιστού. Τα API ιστού και οι υπηρεσίες ιστού συχνά συγχέονται μεταξύ τους, ωστόσο, τα API ιστού αποτελούν εξέλιξη των υπηρεσιών διαδικτύου. Και οι δύο διευκολύνουν τη μεταφορά πληροφοριών, αλλά τα διαδικτυακά API είναι πιο δυναμικά από τις υπηρεσίες ιστού.

API σημαίνει Application Programming Interface, δηλαδή Διασύνδεση Προγραμματισμού Εφαρμογών και ορίζεται από το Ινστιτούτο Τεχνολογίας Λογισμικού ως: *«Η διεπαφή προγραμματισμού εφαρμογών (API) είναι η παλαιότερη τεχνολογία που διευκολύνει την ανταλλαγή μηνυμάτων ή δεδομένων μεταξύ δύο ή περισσότερων διαφορετικών εφαρμογών λογισμικού. Το API είναι η εικονική διασύνδεση μεταξύ δύο λειτουργιών αλληλεπίδρασης λογισμικού, όπως ο επεξεργαστής κειμένου, το υπολογιστικό φύλλο. Ένα API είναι το λογισμικό που χρησιμοποιείται για την υποστήριξη ολοκλήρωσης σε επίπεδο συστήματος πολλαπλών προϊόντων λογισμικού (COTS) ή λογισμικού που αναπτύχθηκε πρόσφατα σε υπάρχουσες ή νέες εφαρμογές.»* (Souza, Redmiles, Cheng, & Millen, 2004)

Η λέξη διασύνδεση χρησιμοποιείται για να δηλώσει ρητά ότι τα API είναι κατασκευάσματα που υπάρχουν στα όρια τουλάχιστον δύο διαφορετικών εφαρμογών ή συστατικών λογισμικού. Αυτές οι δύο (ή περισσότερες) εφαρμογές αναπτύσσονται συχνά από διαφορετικές ομάδες. Ένα χαρακτηριστικό παράδειγμα API αποτελεί το API των Microsoft Windows που επιτρέπει σε ένα πρόγραμμα να έχει πρόσβαση και να χρησιμοποιεί πόρους (συστήματα αρχείων) του υποκείμενου λειτουργικού συστήματος. (Souza, Redmiles, Cheng, & Millen, 2004)

Τα API υιοθετούνται σε μεγάλο βαθμό από τη βιομηχανία, διότι υποστηρίζουν το διαχωρισμό της διεπαφής από την εφαρμογή, που σήμερα αποτελεί μια αυξανόμενη τάση στον σχεδιασμό λογισμικού. Το κύριο πλεονέκτημα αυτής της προσέγγισης είναι η δυνατότητα διαχωρισμού των modules σε δημόσια (το ίδιο το API) και ιδιωτικά (η εφαρμογή των API), έτσι ώστε οι αλλαγές στο ιδιωτικό μέρος να μπορούν να πραγματοποιηθούν χωρίς να επηρεαστούν οι δημόσιες. (Souza, Redmiles, Cheng, & Millen, 2004)

3.3.7. Παραδείγματα Τεχνολογίας API

Ένα σημαντικό παράδειγμα που μπορεί να αναφερθεί προέρχεται από τα συστήματα κράτησης ταξιδιών. Για παράδειγμα, όταν ένας χρήστης θέλει να ταξιδέψει με οποιαδήποτε αεροπορική εταιρία, πηγαίνει σε ένα σύστημα κράτησης ταξιδιών (παράδειγμα το viva.gr). Εισάγει τον προορισμό, αφού επιλέξει ημερομηνίες και κάνει κλικ στην αναζήτηση, αυτό που λαμβάνει είναι τα αποτελέσματα από διαφορετικές αεροπορικές εταιρείες. Πώς επικοινωνεί το οποιοδήποτε σύστημα κράτησης ταξιδιών με όλες αυτές τις αεροπορικές εταιρείες; Πρέπει να υπάρχουν ορισμένοι τρόποι με τους οποίους αυτές οι αεροπορικές εταιρείες εκθέτουν κάποιο επίπεδο πληροφοριών στο σύστημα. Αυτό γίνεται μέσω API.

Ένα άλλο παράδειγμα είναι όταν ένας χρήστης έχει εγκαταστήσει μια εφαρμογή στο κινητό του και την τρέχει, τότε κάποιες φορές δίνεται η δυνατότητα ο χρήστης να συνδεθεί ή να συνεχίσει με το Facebook και το Google. Σε αυτή την περίπτωση, το Google και το Facebook εκθέτουν επίσης κάποιο επίπεδο πληροφοριών χρηστών. Υπάρχει ήδη μια συμφωνία μεταξύ της εφαρμογής και του Google / Facebook, μέσω της οποίας χρησιμοποιείται η τεχνολογία API.

3.3.8. Διαφορές Μεταξύ Υπηρεσιών και Πρόταση για το Σύστημα

Τα API και οι υπηρεσίες Web είναι αμφότερα μέσα επικοινωνίας μεταξύ παρόχων υπηρεσιών και καταναλωτών υπηρεσιών. Στον Πίνακας 3.4 παρουσιάζεται η αναλογία μεταξύ αυτών των δύο.

Πίνακας 3.4: Διαφορές Ανάμεσα στο API και στις Διαδικτυακές Υπηρεσίες

Διαδικτυακές Υπηρεσίες	API
Όλες οι διαδικτυακές υπηρεσίες είναι API	Όλα τα API δεν είναι διαδικτυακές υπηρεσίες
Μπορεί να φιλοξενηθεί μόνο από το IIS (Internet Information Services)	Μπορεί να φιλοξενηθεί από οποιαδήποτε εφαρμογή ή και από το IIS

Δεν είναι ανοικτή πηγή αλλά μπορεί να χρησιμοποιηθεί από οποιονδήποτε πελάτη που κατανοεί την XML	Είναι ανοικτού κώδικα και μπορεί να χρησιμοποιηθεί από οποιονδήποτε πελάτη που κατανοεί το JSON ή το XML
Απαιτεί ένα πρωτόκολλο SOAP για να λαμβάνει και να στέλνει δεδομένα μέσω του δικτύου (δεν είναι μια «ελαφριά» αρχιτεκτονική).	Είναι «ελαφριά» αρχιτεκτονική και είναι καλό για συσκευές με περιορισμένο εύρος ζώνης, όπως οι κινητές συσκευές.
Μια υπηρεσία Web χρησιμοποιεί μόνο τρία πρωτόκολλα (SOAP, REST και XML-RPC) για επικοινωνία	Το API μπορεί να χρησιμοποιεί οποιοδήποτε πρωτόκολλο επικοινωνίας.
Υποστηρίζει μόνο το πρωτόκολλο HTTP	Υποστηρίζει το πρωτόκολλο HTTP: URL, κεφαλίδες αιτήματος / απόκρισης, προσωρινή αποθήκευση, εκδόσεις, μορφές περιεχομένου.

Ένα API είναι μια μέθοδος με την οποία οι προμηθευτές τρίτων μπορούν να γράψουν προγράμματα που διασυνδέονται εύκολα με άλλα προγράμματα. Μια υπηρεσία Web έχει σχεδιαστεί έτσι ώστε να διαθέτει μια διεπαφή που απεικονίζεται σε μια μορφή επεξεργαστή με μηχανή που συνήθως καθορίζεται στη Γλώσσα περιγραφής της υπηρεσίας Web (WSDL). Συνήθως, το "HTTP" είναι το πιο συχνά χρησιμοποιούμενο πρωτόκολλο επικοινωνίας. Η υπηρεσία Web χρησιμοποιεί επίσης SOAP, REST και XML-RPC ως μέσο επικοινωνίας. Το API μπορεί να χρησιμοποιήσει οποιοδήποτε μέσο επικοινωνίας για να ξεκινήσει την αλληλεπίδραση μεταξύ εφαρμογών.

Ένα API ορίζει ακριβώς τις μεθόδους για ένα πρόγραμμα λογισμικού που αλληλοεπιδρά με το άλλο. Όταν η ενέργεια αυτή περιλαμβάνει την αποστολή δεδομένων μέσω δικτύου, οι υπηρεσίες Web εισέρχονται. Ένα API περιλαμβάνει γενικά κλήσεις λειτουργιών μέσα από ένα πρόγραμμα λογισμικού.

Στην περίπτωση εφαρμογών Web, το API που χρησιμοποιείται είναι διαδικτυακό. Οι εφαρμογές της επιφάνειας εργασίας όπως τα υπολογιστικά φύλλα (για παράδειγμα Excel) και τα έγγραφα (για παράδειγμα Word) χρησιμοποιούν API βασισμένα σε VBA και COM που δεν περιλαμβάνουν υπηρεσία Web. Μια εφαρμογή διακομιστή (Server), μπορεί να χρησιμοποιεί ένα API μέσα στον διακομιστή(Server) και να μην απαιτεί κάποια υπηρεσία Web.

Μια υπηρεσία Web είναι ένα API σε HTTP, ενώ ένα API δεν χρειάζεται πάντα να βασίζεται στο διαδίκτυο. Το API αποτελείται από ένα πλήρες σύνολο κανόνων και προδιαγραφών για ένα πρόγραμμα λογισμικού που θα ακολουθήσει για να διευκολύνει την αλληλεπίδραση. Μια υπηρεσία Web ενδέχεται να μην περιέχει ένα πλήρες σύνολο προδιαγραφών και μερικές φορές ενδέχεται να μην είναι σε θέση να εκτελέσει όλες τις εργασίες που είναι δυνατό από ένα πλήρες API.

Επομένως, γίνεται πασιφανές ότι η χρήση των υπηρεσιών API είναι σαφώς καλύτερη από την χρήση οποιασδήποτε άλλης διαδικτυακής υπηρεσίας. Ωστόσο, πρέπει να επισημανθεί ότι για να μπορέσει κάποιος πελάτης να χρησιμοποιεί σύστημα με API τεχνολογία, θα πρέπει να έχει υλοποιήσει μια επέκταση στο δικό του σύστημα, δηλαδή καθίσταται αναγκαίο ο software provider της εκάστοτε επιχείρησης να έχει προσαρμόσει τα API's.

Για το προς υλοποίηση σύστημα, προτείνεται η διασύνδεση μεταξύ των διαφόρων modules να γίνεται με την χρήση των API's υπηρεσιών, καθώς αποτελούν μια ευρύτερη έννοια των διαδικτυακών υπηρεσιών, δίνοντας με αυτόν τον τρόπο μεγαλύτερη ευελιξία στα εμπλεκόμενα μέρη. Σημαντική επισήμανση είναι το γεγονός ότι όλοι θα πρέπει να έχουν πραγματοποιήσει επέκταση στο πληροφοριακό τους σύστημα, ώστε αυτό να μπορεί να προσαρμόσει και να χρησιμοποιήσει την τεχνολογία των API's.

Στην περίπτωση που κάποιος πελάτης logistic δεν μπορεί να επεκτείνει το πληροφοριακό του σύστημα, τότε η διασύνδεση με τις εφαρμογές του Λογισμικού γίνεται μέσω της τεχνολογίας ASPx.net που αναφέρθηκε στην ενότητα 3.2.4. Οι υπόλοιπο εμπλεκόμενοι, δηλαδή ο πάροχος των δεδομένων κίνησης, χαρτών κ.λπ.. είναι αναγκαίο να είναι συμβατοί με την τεχνολογία των API's services.

3.4. Πρωτόκολλα Ανταλλαγής Δεδομένων

3.4.1. Εισαγωγή στα Πρωτόκολλα Ανταλλαγής Δεδομένων

Στο προηγούμενο κεφάλαιο έγινε εκτενής ανάλυση για τις υπηρεσίες, API & Υπηρεσίες Web, που συναντά κανείς με σκοπό την αλληλεπίδραση διαφόρων εφαρμογών. Επίσης, πολύ συχνά αναφερόταν η έννοια του πρωτοκόλλου, και συγκεκριμένα το πρωτόκολλο SOAP και REST. Το πρωτόκολλο είναι ένα σύνολο κανόνων που χρησιμοποιείται για συγκεκριμένους σκοπούς. Όταν γίνεται αναφορά στα πρωτόκολλα, πρόκειται για την επικοινωνία. Για παράδειγμα, η επικοινωνία μεταξύ δύο ανθρώπων στα ελληνικά, γίνεται γιατί αυτά τα άτομα, μπορούν να καταλάβουν τα ελληνικά. Εδώ τα ελληνικά είναι το πρωτόκολλο. Τη στιγμή που αρχίσουν να μιλάνε σε άλλη γλώσσα που δεν καταλαβαίνουν και οι δύο, τότε

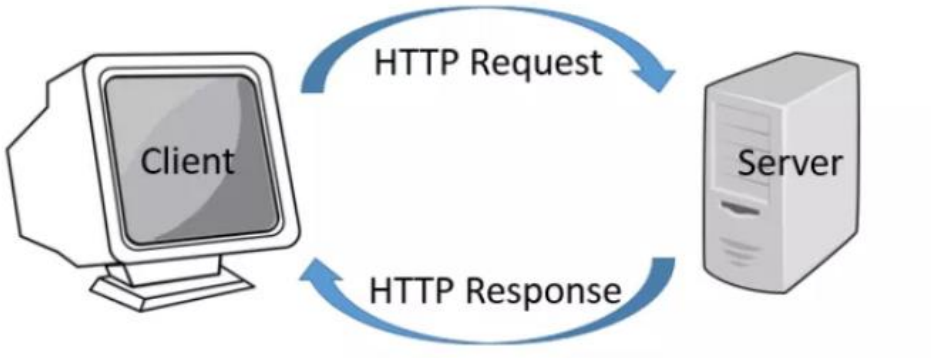
δημιουργείται πρόβλημα με το πρωτόκολλο. Έτσι, χρειάζεται και τα δύο μέρη να συμφωνήσουν σε ένα σύνολο κανόνων για την πραγματοποίηση της επικοινωνίας.

Όμως, για να γίνει καλύτερα κατανοητή η έννοια των πρωτοκόλλων, θα πρέπει να εισχωρήσουμε πιο βαθιά, και να αναφερθούμε για άλλη μια φορά στον παγκόσμιο ιστό, δίνοντας ένα χαρακτηριστικό παράδειγμα της καθημερινής χρήσης.

Όπως έχει ήδη αναφερθεί μέσω του Παγκόσμιου Ιστού ο χρήστης έχει την δυνατότητα να βρίσκει διάφορους πόρους, αυτή είναι άλλωστε η ουσία ύπαρξής του. Μέσω των πόρων καταλήγει σε έναν τελικό σκοπό, όπως για παράδειγμα η παραγγελία ενός προϊόντος ή η προβολή ενός βίντεο. Οι πόροι του διαδικτύου βρίσκονται σε διακομιστές. Για παράδειγμα η προβολή ενός βίντεο, δηλαδή η προβολή ενός πόρου, στον διαδικτυακό ιστότοπο YouTube, είναι μια υπηρεσία του διαδικτύου. Το συγκεκριμένο βίντεο βρίσκεται στην βάση δεδομένων του YouTube, ωστόσο ο χρήστης μέσω του Παγκόσμιου Ιστού έχει πρόσβαση στην αναπαράσταση του βίντεο και όχι άμεσα στον ίδιο τον πόρο. Ο πόρος που είναι αποθηκευμένος στην βάση δεδομένων και η αναπαράσταση του είναι δύο εντελώς διαφορετικά στοιχεία, καθώς ένα βίντεο μπορεί ο χρήστης να το δει μέσω flash player και μέσω HTML (Κκάθε φορά θα εμφανίζεται με διαφορετικό τρόπο).

Ο Παγκόσμιο Ιστός, όπως έχει αναφερθεί, αποτελεί μια πλατφόρμα εφαρμογών με σκοπό την συμβατότητα τους μέσω των διαφορετικών αναπαραστάσεων. Παρόλα αυτά, αν δεν υπήρχαν κανόνες που να δεσμεύουν τους προγραμματιστές να δημιουργούν συγκεκριμένες αναπαραστάσεις, τότε η επικοινωνία μεταξύ των εφαρμογών θα ήταν ιδιαίτερα περίπλοκη και δύσκολη. Επομένως, ήταν και συνεχίζει να είναι αναγκαία η καθιέρωση κάποιων συγκεκριμένων αναπαραστάσεων, για παράδειγμα JSON & XML. (Δημανίδης, 2017)

Με ποιον τρόπο, όμως, οι εφαρμογές επικοινωνούν μεταξύ τους στον Παγκόσμιο Ιστό ; Η απάντηση σε αυτό το ερώτημα δόθηκε με το πρωτόκολλο HTTP. Ο σκοπός του HTTP είναι να μπορούν να επικοινωνούν εφαρμογές μεταξύ τους, όχι υπολογιστές. Την επικοινωνία των υπολογιστών την αναλαμβάνει το Διαδίκτυο. Κάθε φορά που ένας χρήστης πληκτρολογεί μια διεύθυνση URL, σε ένα πρόγραμμα περιήγησης, τότε ουσιαστικά αποστέλλει ένα αίτημα στον διακομιστή του Ιστού και αυτός ανταποκρίνεται με την αναπαράσταση ενός πόρου, του πόρου που αντιστοιχεί στην συγκεκριμένη διεύθυνση.



Σχήμα 3.5: Πρωτόκολλο HTTP

Το πρωτόκολλο HTTP είναι ένα πρωτόκολλο εφαρμογών και όχι ανταλλαγής μηνυμάτων. Αυτό που πραγματικά συμβαίνει είναι το εξής: Οι προγραμματιστές χρησιμοποιούν τον παγκόσμιο Ιστό με σκοπό την επικοινωνία ανάμεσα στις εφαρμογές και όχι ως πλατφόρμα εφαρμογών. Καταφέρνουν να εισχωρούν την λογική των εφαρμογών τους μέσα στα μηνύματα HTTP, υλοποιώντας κατά αυτόν τον τρόπο πρωτόκολλα εφαρμογών για την συγκεκριμένη εφαρμογή. Πραγματοποιούν το παράδοξο να αναπτύσσουν πρωτόκολλα εφαρμογών πάνω σε υπάρχων πρωτόκολλο εφαρμογών. Επομένως, δεν είναι το HTTP το πρωτόκολλο επικοινωνίας των εφαρμογών, αλλά υπάρχει ένα σύνολο πρωτοκόλλων για την επικοινωνία εφαρμογών στον Παγκόσμιο Ιστό, σε αυτά συγκαταλέγονται τα πρωτόκολλα SOAP, WSDL, REST. (Δημανίδης, 2017)

3.4.2. Πρωτόκολλο SOAP

Το SOAP (Simple Object Access Protocol) είναι μια γλώσσα XML που ορίζει την αρχιτεκτονική και τις μορφές μηνυμάτων, παρέχοντας έτσι μια στοιχειώδη μορφή πρωτοκόλλου επεξεργασίας. Το έγγραφο SOAP ορίζει ένα στοιχείο XML, που ονομάζεται φάκελος, το οποίο περιέχει μια κεφαλίδα και ένα κυρίως σώμα.

- Η κεφαλίδα του SOAP είναι ένα εκτάσιμο στοιχείο για πληροφορίες σχετικά με τις υποδομές μηνυμάτων που μπορούν να χρησιμοποιηθούν για σκοπούς δρομολόγησης (π.χ. διεύθυνση) και διαμόρφωση ποιότητας υπηρεσίας (QoS) (π.χ. συναλλαγές, ασφάλεια, αξιοπιστία).
- Το σώμα περιέχει την ωφέλιμη πληροφορία του μηνύματος. Είναι το στρώμα που περιλαμβάνει τις υποχρεωτικές πληροφορίες, οι οποίες προορίζονται για το δέκτη των μηνυμάτων.

Το σχήμα του XML χρησιμοποιείται για να περιγράψει τη δομή του μηνύματος SOAP, έτσι ώστε οι μηχανές SOAP στα δύο τελικά σημεία να μπορούν α ταξινομούν και να καταργούν το

περιεχόμενο του μηνύματος και να το κατευθύνουν στην κατάλληλη υλοποίηση. (Pautasso, Zimmermann, & Leymann, 2008)

Ενώ η κεφαλίδα και το σώμα ορίζονται ως ανεξάρτητα στοιχεία, στην πραγματικότητα συνδέονται. Η σχέση μεταξύ μιας εγγραφής σώματος και μιας καταχώρησης κεφαλίδας έχει ως εξής: Μια καταχώρηση σώματος είναι σημασιολογικά ισοδύναμη με μια καταχώρηση κεφαλίδας προοριζόμενη για τον προεπιλεγμένο ρόλο και με χαρακτηριστικό SOAP mustUnderstand με τιμή "1". Ο προεπιλεγμένος ρόλος υποδεικνύεται χρησιμοποιώντας το χαρακτηριστικό του ρόλου. (Box, και συν., 2000)

Ένα μήνυμα SOAP ταξιδεύει από τον εντολέα στον τελικό προορισμό, ενδεχομένως περνώντας μέσω ενός συνόλου διαμεσολαβητών SOAP κατά μήκος της διαδρομής του μηνύματος. Ένας διαμεσολαβητής SOAP είναι μια εφαρμογή που είναι ικανή να λαμβάνει και να προωθεί μηνύματα SOAP. Και οι δύο διαμεσολαβητές καθώς και ο τελικός προορισμός αναγνωρίζονται από ένα URI. Δεν είναι δυνατόν όλα τα μέρη ενός μηνύματος SOAP να προορίζονται για τον τελικό προορισμό του μηνύματος SOAP αλλά, αντίθετα, μπορεί να προορίζονται για έναν ή περισσότερους από τους διαμεσολαβητές στη διαδρομή του μηνύματος. (Box, et al., 2000)

Το αν θα κάποια μέρη του μηνύματος προορίζονται για κάποιον ενδιάμεσο διαμεσολαβητή το καθορίζει ο ρόλος (actor) της κεφαλίδας. Η ιδιότητα αυτή χρησιμοποιείται ώστε η επικεφαλίδα του μηνύματος να απευθυνθεί σε έναν τέτοιο κόμβο. Επίσης, ένα άλλο στοιχείο της κεφαλίδας είναι το MustUnderstand το οποίο καθορίζει αν πρέπει ή όχι ο παραλήπτης να προχωρήσει στην επεξεργασία κάποιου στοιχείου της Κεφαλίδας.

Σχήμα 3.6: Επικοινωνία Μέσω του Πρωτοκόλλου SOAP

Ο ρόλος του παραλήπτη ενός στοιχείου κεφαλίδας είναι παρόμοιος με αυτόν της αποδοχής μιας σύμβασης, δεδομένου ότι δεν μπορεί να επεκταθεί πέραν του παραλήπτη. Δηλαδή, ένας παραλήπτης που λαμβάνει ένα στοιχείο κεφαλίδας δεν πρέπει να προωθεί αυτό το στοιχείο κεφαλίδας στην επόμενη εφαρμογή κατά τη διαδρομή μηνύματος SOAP. Ο παραλήπτης ωστόσο μπορεί να εισάγει ένα παρόμοιο στοιχείο κεφαλίδας, αλλά στην περίπτωση αυτή, η σύμβαση είναι μεταξύ αυτής της εφαρμογής και του παραλήπτη αυτού του στοιχείου κεφαλίδας. (Box, et al., 2000)

Το SOAP είναι ένα σχετικά απλό πρωτόκολλο χρήσης. Τα πακέτα εργαλείων SOAP για τους διακομιστές Apache και Microsoft βασίζονται σε δωρεάν λήψεις. (Roy & Ramanujan, 2001)

Σχήμα 3.7: Κλήση και Απάντηση Πρωτοκόλλου SOAP

Ένα παράδειγμα μιας υπηρεσίας Web, που χρησιμοποιεί SOAP, φαίνεται στο Σχήμα 3.7. Τι τμήμα (α) είναι ένα αίτημα SOAP που αποστέλλεται μέσω HTTP σε μια υπηρεσία Web για τη θερμοκρασία στην Νέα Υόρκη. Το τμήμα (b) δείχνει την απόκριση HTTP SOAP για αυτό το αίτημα της θερμοκρασίας καιρού. Το στοιχείο XML GetCurrentTempOutput στο μήνυμα απόκρισης SOAP περιέχει τη σημερινή θερμοκρασία της Νέας Υόρκης, η οποία φαίνεται ότι είναι 20 βαθμοί της κλίμακας Κελσίου.

3.4.3. Γλώσσα WSDL (Web Services Description Language)

Όπως αναφέρθηκε στην προηγούμενη ενότητα, η περιγραφή του πρωτοκόλλου SOAP παρέχεται σε ένα αρχείο XML, το οποίο ονομάζεται Web Services Description Language (WSDL). Ένα αρχείο WSDL καθορίζει δύο πράγματα

- Που βρίσκεται διαθέσιμη η υπηρεσία
- Ποιο πρωτόκολλο επικοινωνίας χρησιμοποιείται

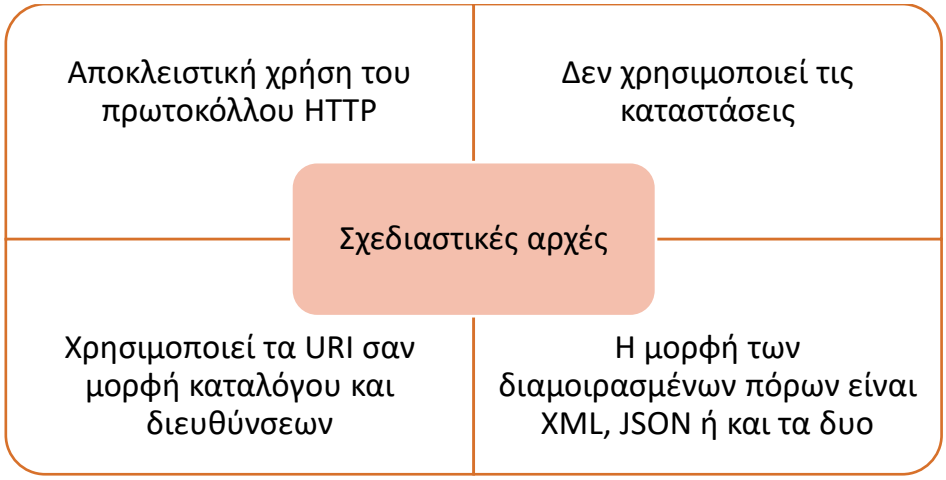
Όπως αναφέρεται και στο κεφάλαιο 9 του βιβλίου των (Δουληγέρης, 2015) *το έγγραφο WSDL είναι ένα σχήμα XML για την περιγραφή Υπηρεσιών Ιστού, ως ένα σύνολο από τελικά σημεία, τα οποία λειτουργούν με ανταλλαγή μηνυμάτων, που περιέχουν πληροφορίες προσανατολισμένες είτε στα έγγραφα είτε στις διαδικασίες. Ένα έγγραφο WSDL χρησιμοποιείται για να περιγράψει τι μπορεί να κάνει μια WS, πού βρίσκεται και πώς μπορεί κάποιος να το καλέσει.*

3.4.4. Πρωτόκολλο REST

Πως ο Παγκόσμιος Ιστός μπορεί να βελτιωθεί, ώστε να είναι πιο φιλικός στην ανάπτυξη εφαρμογών;. Η αποδοχή του Παγκόσμιου Ιστού ανέδειξε την ανάγκη για ένα κοινό σχεδιαστικό σημείο, για όλους εκείνους τους προγραμματιστές που ανέπτυσσαν λογισμικό στον Παγκόσμιο Ιστό και άρα χρησιμοποιούσαν το πρωτόκολλα HTTP. Εδώ έρχεται η έννοια του REST, Representational State Transfer, ενός συνόλου κανόνων. Έξι χρόνια μετά την ίδρυση του Παγκόσμιου Ιστού ο Fielding, κατάφερε να βρει αυτό το κοινό σημείο, χρησιμοποιώντας τους κανόνες REST. Έτσι, σχεδίασε την νέα έκδοση HTTP/1.1 βασισμένος στη φιλοσοφία REST. Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι παρά την σύνδεση που προσδίδει η ιστορία του REST με το πρωτόκολλο HTTP, δεν αφορά μόνο συστήματα Παγκόσμιου Ιστού αλλά οποιοδήποτε σύστημα το οποίο ανταλλάσσει πληροφορίες μέσω ενός δικτύου. (Δημανίδης, 2017)

Το πρωτόκολλο REST λόγω του ότι είναι απλό στη χρήση του έχει καταφέρει να εδραιωθεί σαν μοντέλο σχεδιασμού υπηρεσιών μέσω διαδικτύου, δηλαδή υπηρεσιών ιστού. Συγκεκριμένα, οι υπηρεσίες ιστού εστιάζουν στον τρόπο με τον οποίο οι πόροι του συστήματος χαρακτηρίζονται και μεταφέρονται διαμέσου του πρωτοκόλλου HTTP από έναν μεγάλο αριθμό χρηστών. Το REST είναι μια αρχιτεκτονική ιδανική για την ανάπτυξη εφαρμογών διαδικτύου. (Rodriguez, 2008)

Το πρωτόκολλο REST οφείλει την συμπαγή εφαρμογή του σε τέσσερις σχεδιαστικές αρχές:



Σχήμα 3.8: Σχεδιαστικές Αρχές Πρωτοκόλλου REST

Είναι φανερό ότι προσπαθεί να μοιάσει σε αρχιτεκτονικές που χρησιμοποιούν το πρωτόκολλο HTTP, με την διαφορά ότι περιορίζει την συνήθη διασύνδεση σε τέσσερις γνωστές λειτουργίες του HTTP. Σκοπός του είναι η αλληλεπίδραση με τους πόρους του συστήματος και όχι με τα μηνύματα, για αυτό το λόγο το πρωτόκολλό REST χρησιμοποιεί το HTTP για την μεταφορά δεδομένων, και όχι να λειτουργεί πάνω στο πρωτόκολλο HTTP για την μεταφορά μηνυμάτων, όπως κάνει το SOAP.

Οι λειτουργίες του REST, από την πλευρά του διακομιστή είναι οι εξής

- GET: Όλες οι μέθοδοι με τις οποίες αντλούνται δεδομένα από τον διακομιστή
- POST: Όλες οι μέθοδοι με τις οποίες προστίθενται δεδομένα από τον διακομιστή
- PUT: Όλες οι μέθοδοι με τις οποίες ενημερώνονται δεδομένα από τον διακομιστή
- DELETE: Όλες οι μέθοδοι με τις οποίες διαγράφονται δεδομένα από τον διακομιστή (Rodriguez, 2008)

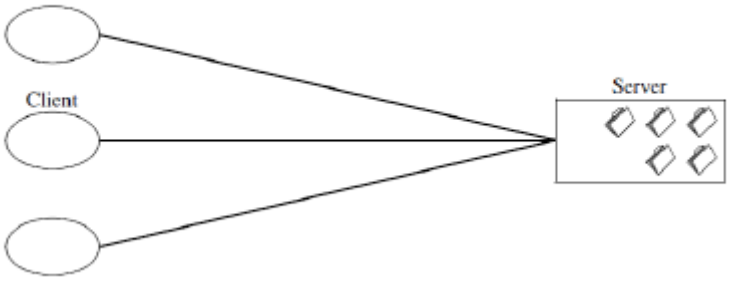
Οι περιορισμοί που ορίζει το REST σε ένα σύστημα είναι οι εξής :

Εξυπηρετητής - Πελάτης (Client - Server): Σε ένα σύστημα REST υπάρχει το μοντέλο πελάτη-εξυπηρετητή. Ο διαχωρισμός του πελάτη από τον εξυπηρετητή σημαίνει διαχωρισμός της διεπαφής χρήστη από την αποθήκευση των δεδομένων

.

Σχήμα 3.9: Εξυπηρετητής - Πελάτης στο Σύστημα REST

Αν-ιθαγένεια: Η κατάσταση του πελάτη παραμένει ανεξάρτητη από την κατάσταση του εξυπηρετητή, με αυτόν τον τρόπο ελαφρύνεται το φορτίο του εξυπηρετητή και εκείνος δεν ελέγχει τον πελάτη. Ο πελάτης οφείλει να στέλνει όλη την πληροφορία που απαιτείται ώστε να γίνει το αίτημά του αποδεκτό ή όχι.



Σχήμα 3.10: Αν-ιθαγένεια Εξυπηρετητή – Πελάτη

Κρυφή μνήμη: Πολλοί προγραμματιστές πιστεύουν ότι μια στοίβα 3.3.4 εμποδίζει την εφαρμογή Ιστού ή το API. Αλλά στην πραγματικότητα, ο λόγος είναι η αρχιτεκτονική τους. Η βάση δεδομένων μπορεί να είναι ένα πιθανό κομμάτι συντονισμού σε μια εφαρμογή ιστού. Για να κλιμακωθεί μια εφαρμογή καλά, πρέπει να αποθηκεύεται το περιεχόμενο στην κρυφή μνήμη και στη συνέχεια να το παραδίδεται ως απάντηση. Οι υπηρεσίες REST θα πρέπει να αποθηκεύονται σωστά στην κρυφή μνήμη για κλιμάκωση.

Ομοιόμορφη Διεπαφή: Βασικό στοιχείο του REST είναι η ομοιόμορφη διεπαφή, δηλαδή πρέπει όλα τα συστήματα στο δίκτυο να επικοινωνούν και να συνεργάζονται με τον ίδιο τρόπο. (Δημανίδης, 2017)

Σύστημα Πολλών Επιπέδων (Layered Systems): Το σύστημα χωρίζεται σε επίπεδα, τα οποία αφορούν την πληροφορία που θα διατηρείται. (Δημανίδης, 2017)

Ελευθερία υλοποίησης: Το REST είναι απλά ένας μηχανισμός για τον ορισμό των υπηρεσιών στο διαδίκτυο. Είναι ένα αρχιτεκτονικό στυλ που μπορεί να υλοποιηθεί με πολλούς τρόπους. Λόγω αυτής της ευελιξίας, οι υπηρεσίες REST δημιουργούνται με τον τρόπο που επιθυμεί ο εκάστοτε προγραμματιστής. Εάν ακολουθήσει τις αρχές του REST, ο διακομιστής έχει την ελευθερία να επιλέξει την πλατφόρμα ή την τεχνολογία. Η προσωρινή αποθήκευση είναι απαραίτητη για την κλιμάκωση των υπηρεσιών REST.

3.4.5. Σύγκριση Πρωτοκόλλων REST και SOAP και Πρόταση για το Σύστημα

Οι δύο πιο δημοφιλείς τεχνολογίες για την δημιουργία μιας αρχιτεκτονικής συστήματος με υψηλότερη απόδοση και μεγαλύτερη διατηρησιμότητα είναι το SOAP και το REST. Για την

σύγκριση των τεχνολογιών αυτών έχουν προταθεί κάποιες μελέτες, με κάθε μια να ξεχωρίζει διαφορετικές προσεγγίσεις.

Προσέγγιση των Muehlen, Nickerson & Swenson (2005)

Σε αυτή την προσέγγιση η σύγκριση γίνεται μέσω ενός γενικού μοντέλου που αντιπροσωπεύει τον τυπικό σχεδιασμό μιας εφαρμογής.

Σχήμα 3.11:Τυπικος Σχεδιασμός Εφαρμογής

Η προσέγγιση του πρωτοκόλλου SOAP βασίζεται στην γλώσσα WSDL για την επικοινωνία και στο SOAP ως πρότυπο μηνυμάτων. Με βασικά πλεονεκτήματα το σφιχτό ζευγάρωμα των στοιχείων. από την άλλη, με το REST το σύστημα διέπεται από την αρχή της επεκτασιμότητας του Παγκόσμιου Ιστού, με βασικά πλεονεκτήματα την ευκολία των λίγων λειτουργιών. Κατηγοριοποιημένα τα εκάστοτε χαρακτηριστικά παρουσιάζονται στον παρακάτω Πίνακας 3.5

Πίνακας 3.5: Σύγκριση SOAP και REST

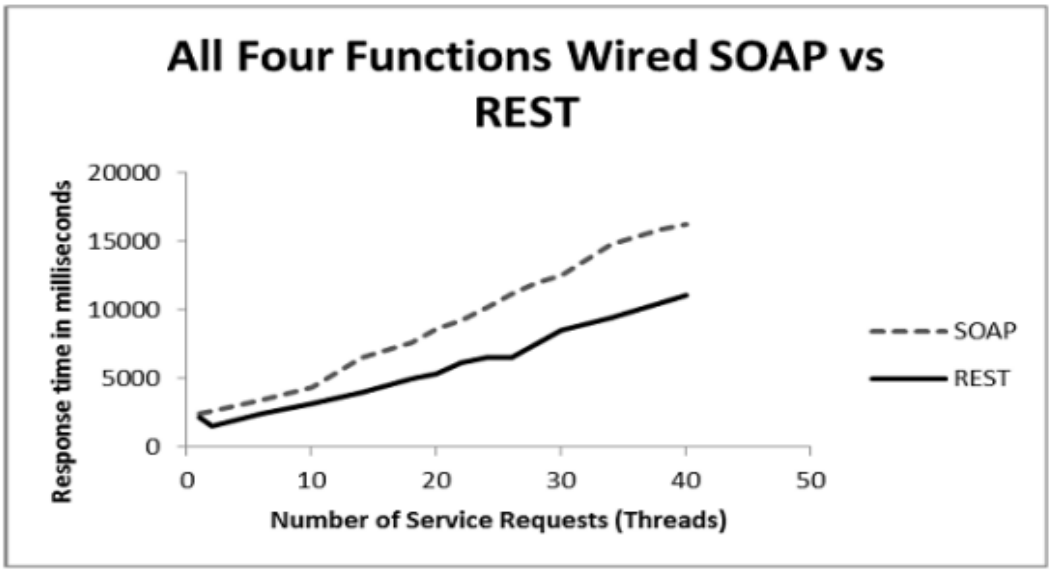
		SOAP	REST
Χαρακτηριστικά	*Λειτουργίες*	Ορίζονται από το WSDL	Ορίζονται μέσα στο μήνυμα
		Η ίδια λειτουργία για πολλά στιγμιότυπα	Κάθε αντικείμενο τις δικές του λειτουργίες
	Διεύθυνση	Μοναδική για κάθε λειτουργία	Μοναδική για κάθε στιγμιότυπο διαδικασίας
	Σύνδεση στοιχείων	Στενή	Χαλαρή
Πλεονεκτήματα	*Πελάτης*	Άμεση ενσωμάτωση υπαρχόντων API's	Δεν χρειάζεται πληροφορίες δρομολόγησης
			Γενικότερη διεπαφή για τις ειδοποιήσεις
	Σύνδεση	Πιθανός εντοπισμός σφαλμάτων	Πιθανή καθυστέρηση

Μειονεκτήματα		Γνώση λειτουργιών εκ των προτέρων	Μεγάλος όγκος αντικειμένων
		Διαφορετικές θύρες για τις διάφορες ειδοποιήσεις	Πιθανή δυσκολία διαχείρισης χώρου URI
		Έμμεση δημιουργία στιγμιότυπων	

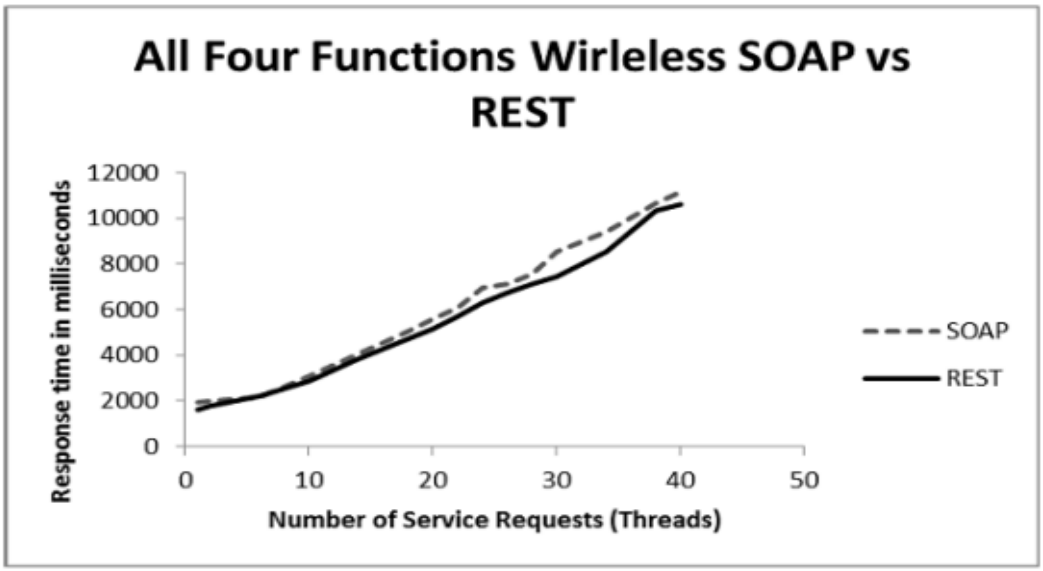
Προσέγγιση Potti

Για την συγκεκριμένη προσέγγιση, αναπτύχθηκαν δύο υπηρεσίες ιστού, η μια βασίστηκε στο SOAP και η άλλη στο REST, και είχαν ως βασικές λειτουργίες τις GET, POST, PUT και DELETE. Έτσι, συγκρίθηκαν οι επιδόσεις των REST και SOAP πρωτοκόλλων. Πρέπει, επίσης, να επισημανθεί ότι η μελέτη έγινε και για τους δύο τύπους συνδέσεων, ενσύρματης και ασύρματης, και ότι η βασική διαδικασία ήταν ότι ένας πελάτης έκανε αίτηση για ένα αρχείο στον διακομιστή και ο διακομιστής με την σειρά του έστελνε το αρχείο πίσω. Τα κριτήρια αυτής της μελέτης ήταν α) ο χρόνος απόκρισης β) τα δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput) με βάση το μέγεθος των αρχείων και γ) τα δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput) με βάση τον αριθμό των πελατών.

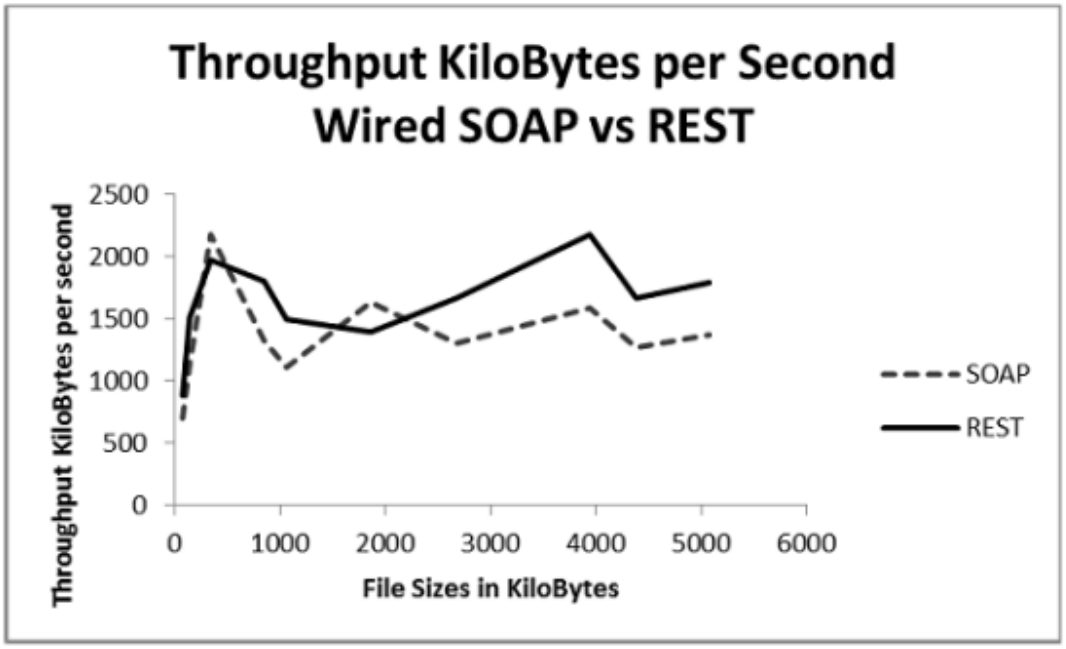
Η ανάλυση κατέδειξε τα εξής διαγράμματα



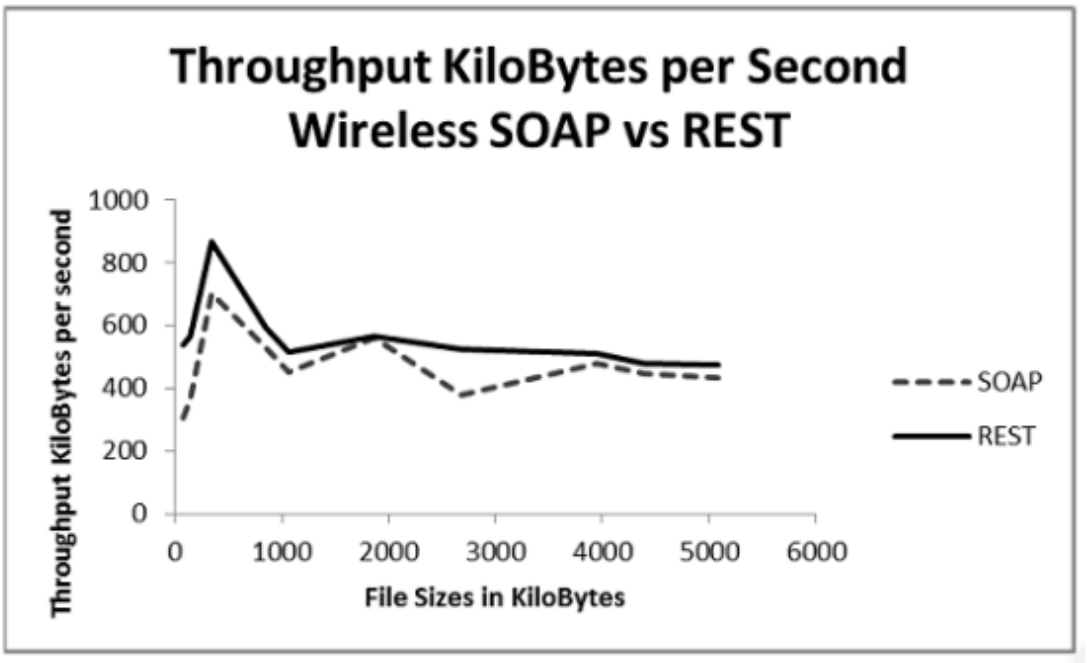
Σχήμα 3.12: Χρόνος Απόκρισης REST και SOAP σε Ενσύρματο



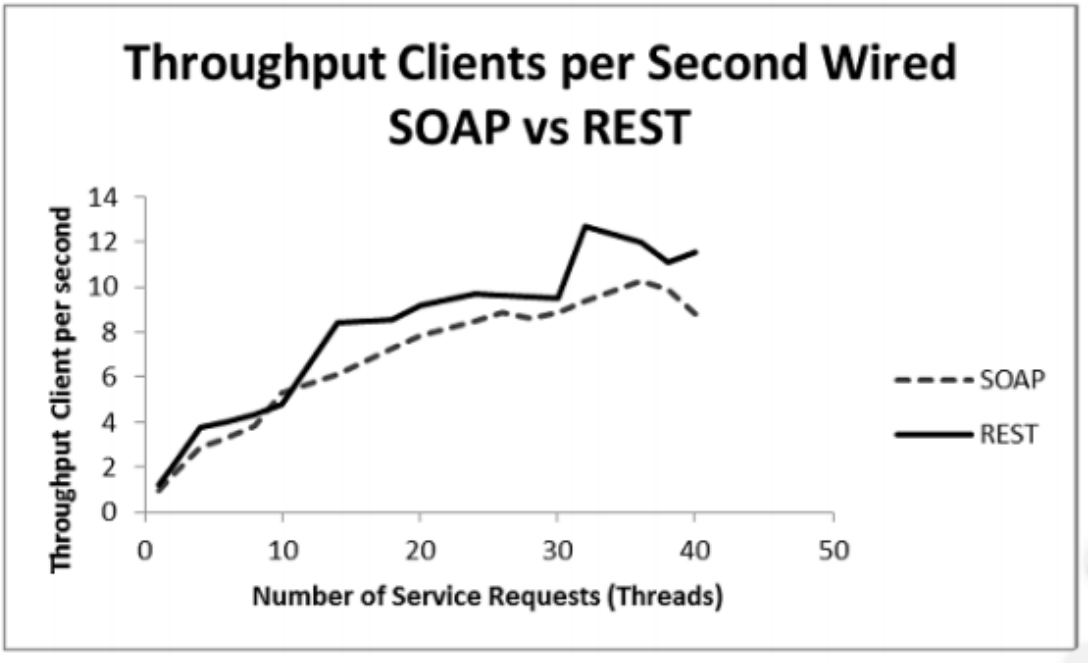
Σχήμα 3.13: Χρόνος Απόκρισης REST και SOAP σε Ασύρματο



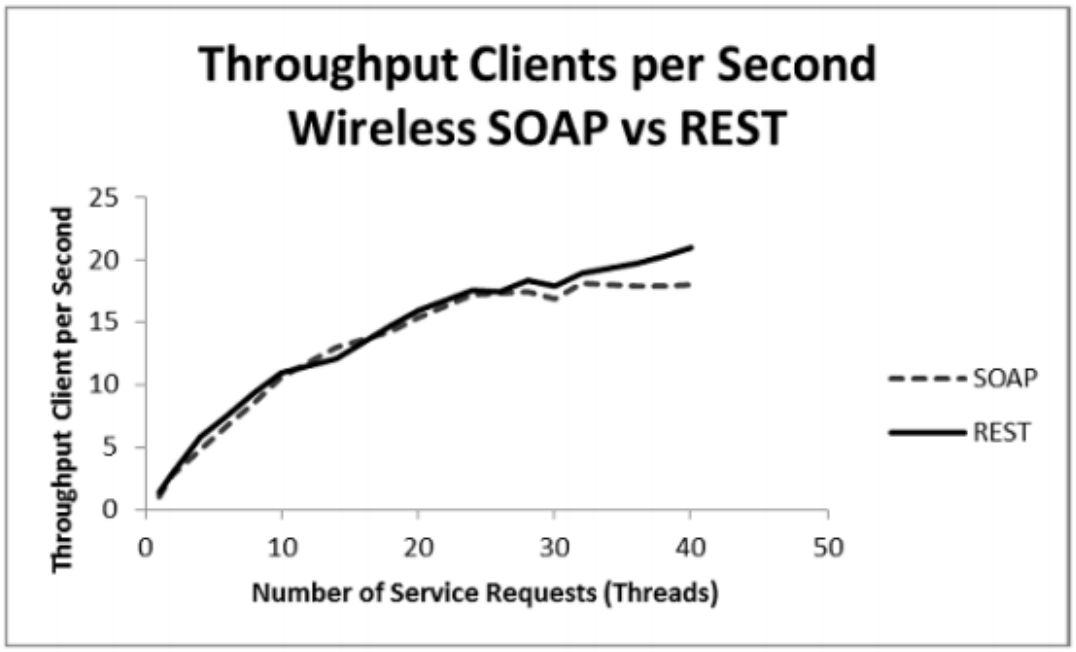
Σχήμα 3.14: Δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput)με βάση το μέγεθος των αρχείων, στο REST και στο SOAP σε ενσύρματο δίκτυο



Σχήμα 3.15: Δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput)με βάση το μέγεθος των αρχείων, στο REST και στο SOAP σε ασύρματο δίκτυο



Σχήμα 3.16:Δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput) με βάση τον αριθμό των πελατών, στο REST και στο SOAP σε ενσύρματο δίκτυο



Σχήμα 3.17: Δεδομένα που υποβάλλονται σε επεξεργασία σε μια μονάδα του χρόνου (throughput) με βάση τον αριθμό των πελατών, στο REST και στο SOAP σε ασύρματο δίκτυο

Στα τέσσερα πρώτα διαγράμματα φαίνεται πως το πρωτόκολλό REST έχει καλύτερη απόδοση για όλες τις διαδικασίες. Στα δύο τελευταία, επίσης παρουσιάζεται το γεγονός ότι το πρωτόκολλο REST έχει συγκριτικά καλύτερη απόδοση με την αύξηση του αριθμού των αιτήσεων για εξυπηρέτηση. Επομένως, κατέληξαν στο συμπέρασμα ότι οι χρόνοι απόκρισης του REST ήταν πολύ καλύτεροι σε σχέση με του SOAP, και για τα δύο είδη δικτύου. Ωστόσο, για εφαρμογές που απαιτούν μεγάλο εύρος ζώνης και πολύπλοκη επεξεργασία το SOAP είναι καλύτερη τεχνολογία. (Πλασταράς, 2016)

Επειδή το σύστημα που μελετάται είναι πραγματικού χρόνου, οι χρόνοι απόκρισης του παίζουν καθοριστικό ρόλο για την επιλογή του πρωτοκόλλου ανταλλαγής δεδομένων. Επομένως, η σύγκριση των δύο τεχνολογιών οδήγησε στο συμπέρασμα πως η καταλληλότερη τεχνολογία είναι το πρωτόκολλο επικοινωνίας REST. Ακόμα, το REST φαίνεται ότι έχει καλύτερη απόδοση, επεκτασιμότητα και δια λειτουργικότητα σε σχέση με το SOAP.

4. Βάσεις Δεδομένων

4.1. Εισαγωγή

Στο Κεφάλαιο 4 παρουσιάζεται μια από τις σημαντικότερες τεχνολογίες για κάθε σύστημα το οποίο διαχειρίζεται δεδομένα. Στο προς ανάλυση σύστημα παρατηρείται πως τα δεδομένα των χαρτών και των δρόμων, τα δεδομένα των στοιχείων κίνησης των προβλέψεων των πραγματικών δεδομένων κίνησης που καλούνται για την δημιουργία είτε του αρχικού πλάνου δρομολόγησης είτε της ανά δρομολόγησης βρίσκονται σε βάσεις δεδομένων. Τα δεδομένα (στατικά και δυναμικά) της εταιρείας logistics που εισέρχονται στο cloud βρίσκονται σε μία βάση δεδομένων, καθώς και αυτά που εξέρχονται μετά την χρήση του αλγορίθμου οδηγούνται σε μια άλλη βάση. Εδώ πρέπει να επισημανθεί ότι ακόμα και το cloud έχει μία βάση δεδομένων, στην οποία υπάρχουν όλα τα δεδομένα που αντλεί ο αλγόριθμος για την λειτουργία του. Επομένως είναι φανερό ότι κάθε module του συστήματος μας εξαρτάται από μία βάση δεδομένων, για αυτό το λόγο σε αυτό το κεφάλαιο γίνεται μία εκτενής ανάλυση του τι είναι μία βάση δεδομένων, ποια είδη υπάρχουν και εν τέλει οδηγείται σε μία πρόταση για το υπάρχον σύστημα. Οι βάσεις δεδομένων που προτείνονται έχουν σχέση με το τον πάροχο του λογισμικού ως υπηρεσία και όχι με τους άλλους παρόχους.

4.2. Γενικά για τις Βάσεις δεδομένων

Οι βάσεις δεδομένων είναι παντού αλλά δεν είναι φανερές, είναι κρυμμένες πίσω από τα εργαλεία και τις υπηρεσίες που χρησιμοποιούνται καθημερινά. Μπορεί κανείς να αναρωτηθεί που αποθηκεύουν τα δεδομένα τους δηλαδή τις φωτογραφίες, τα status, τα blogs διάφορες εφαρμογές όπως το Facebook το Twitter κλπ.. Που αποθηκεύει η Google τις λεπτομέρειες των σελίδων στο διαδίκτυο; Που αποθηκεύονται οι επαφές του κινητό τηλεφώνου; Όλα αποθηκεύονται σε μία βάση δεδομένων. Οι βάσεις δεδομένων αποτελούν το μεγαλύτερο μέρος των εργασιών στα πληροφοριακά συστήματα που χρησιμοποιούνται καθημερινά. Τι είναι λοιπόν μία βάση δεδομένων? Είναι απλά πράγματα που συμπιέζονται μαζί; Όχι, οι βάσεις δεδομένων δεν είναι τόσο τυχαίες, είναι οργανωμένες έχουν δομή και όλα τα δεδομένα που αποθηκεύονται σε αυτές ταιριάζουν σε αυτή τη δομή. Η οργανωμένη συλλογή δεδομένων καλείται βάση δεδομένων.

Ένα απλό παράδειγμα είναι μία βάση ενός ατόμου, το οποίο την χρησιμοποιεί για να αποθηκεύσει το βιβλίο διευθύνσεων του. Αυτή η βάση δεδομένων περιέχει τα ονόματα, τις διευθύνσεις, τα γενέθλιά, τους αριθμούς τηλεφώνων, ίσως και κάποιο σχόλιο, των διαπροσωπικών του σχέσεων. Όταν αυτός θέλει να προσθέσει στο βιβλίο διευθύνσεων του

μία νέα επαφή, τότε γράφει όλα τα στοιχεία του νέου προσώπου. Οι λεπτομέρειες που γράφει ο χρήστης είναι αποθηκευμένες σε πεδία στη βάση δεδομένων του βιβλίου διευθύνσεων του, και οι οποίες τοποθετούνται με τη σειρά στην κάθε μία στήλη της βάσης δεδομένων. Οι πληροφορίες για ένα άτομο αποτελούν μία ολόκληρη σειρά στο αρχείο της βάσης δεδομένων. Κάθε σειρά ονομάζεται εγγραφή και περιέχει πληροφορίες για ένα διαφορετικό άτομο στο βιβλίο διευθύνσεων του. Σε αντίθεση με ένα χαρτόδετο βιβλίο διευθύνσεων, σε μία ψηφιακή βάση δεδομένων ο χρήστης μπορεί να πραγματοποιήσει διάφορες εργασίες. Όπως είναι για παράδειγμα, η εύρεση ενός συγκεκριμένου ατόμου. Ο χρήστης μπορεί να κάνει την αναζήτηση ακριβέστερη στο πεδίο που θέλει, δηλαδή την στήλη στην οποία θέλει να γίνει η αναζήτηση. Επιπρόσθετα, του δίνεται η δυνατότητα να ταξινομεί στη βάση δεδομένων ούτως ώστε να εμφανίζει τα στοιχεία με διαφορετική σειρά (αύξουσα ή φθίνουσα). Αναφέρθηκε ανωτέρω ότι ο κάτοχος της βάσης δεδομένων του βιβλίου διευθύνσεων μπορεί να αποθηκεύει και το όνομα και την ημερομηνία γέννησης του ατόμου, αυτό οδηγεί στο συμπέρασμα ότι στήλες των πεδίων μπορούν να αποθηκεύουν διάφορους τύπους δεδομένων, αλλά μόνο έναν τύπο δεδομένων κάθε φορά. Τα πεδία όνομα, επώνυμο, διεύθυνση, αποθηκεύουν όλα τα δεδομένα τους με τον τύπο κειμένου (μπορεί να περιέχουν δηλαδή ένα μείγμα γραμμάτων αριθμών και συμβόλων), το πεδίο ημερομηνία γέννησης έχει τύπο ημερομηνίας, επομένως αποθηκεύει ημερομηνίες από το ημερολόγιο (dd/mm/yy).

Με αυτό το παράδειγμα παρουσιάστηκε ο τρόπος με τον οποίο χρησιμοποιείται μία απλή βάση δεδομένων στην καθημερινή ζωή, ωστόσο πρέπει να επισημανθεί το γεγονός ότι αυτή ήταν σχετικά απλή και κατανοητή σχεσιακή βάση. Τα συστήματα διαχείρισης βάσεων δεδομένων και βάσεων δεδομένων έχουν γίνει αναπόσπαστο μέρος κάθε είδους εργασίας, είτε στη διαχείριση επιχειρηματικών δεδομένων είτε στη διαχείριση των λογαριασμών των νοικοκυριών. Οι διάφορες εφαρμογές όπως το Facebook και οι υπηρεσίες ιστού χρησιμοποιούν αρκετά πολύπλοκες βάσεις δεδομένων, που αυτές μπορεί να είναι σχεσιακές ή μη σχεσιακές. Η αποθήκευση και διαχείριση δεδομένων σε ένα υπολογιστικό σύστημα είναι μια από τις πιο σημαντικές εφαρμογές των υπολογιστών. Ειδικότερα, στην επιστήμη της πληροφορικής με τον όρο βάσεις δεδομένων αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων αποθηκευμένων δεδομένων.

4.3. Ιστορική Αναδρομή

Μόλις η ανθρώπινη φυλή άρχισε να μετράει και να γράφει, για τη μέτρηση και την αποθήκευση των εμπορευμάτων δημιουργήθηκε η ανάγκη για την συλλογή πληροφοριών των εμπορευμάτων. Για αρχή καθόρισαν τι είναι τα δεδομένα (data). Η Λέξη δεδομένα προέρχεται από τον πληθυντικό αριθμό της λατινικής λέξης datum, που σημαίνει το

δεδομένο. Δεδομένα για ένα γεγονός ή μία πληροφορία μπορεί να είναι μία αναπαράσταση ενός γεγονότος, ενός αριθμού, μιας ιδέας. Έτσι, τα δεδομένα αναφέρονται σε ποιοτικά ή ποσοτικά χαρακτηριστικά μιας μεταβλητής ή ενός συνόλου μεταβλητών. Τα δεδομένα από μόνα τους δεν σημαίνουν κάτι, αφού μπορεί να είναι απλά ένα σύνολο λέξεων. Για παράδειγμα, κάποιοι αριθμοί μπορεί να σημαίνουν είτε μία ηλικία είτε μία ημερομηνία είτε ένα μήνα. Ωστόσο, εάν προσδιοριστεί τι είναι αυτοί οι αριθμοί, τότε υποδηλώνουν μία πληροφορία. Αυτός ο προσδιορισμός ονομάζεται μετά δεδομένα, δηλαδή δεδομένα σχετικά με τα δεδομένα. Περαιτέρω ο άνθρωπος δημιούργησε τη βάση δεδομένων που είναι μία συλλογή δεδομένων ή μία συλλογή οργανωμένων πληροφοριών. (Berg, Seymour, & Goel, 2012)

Η προέλευση των βάσεων δεδομένων πηγάζει από τη βιβλιοθήκες, τα ιατρικά αρχεία, τα κυβερνητικά αρχεία, τα επιχειρηματικά αρχεία πριν από την εφεύρεση των υπολογιστών. Μόλις οι άνθρωποι συνειδητοποίησαν ότι έπρεπε να ανακτούν διαφορά και συγκεκριμένα δεδομένα προσπάθησαν να βρουν τρόπους για την καλύτερη και ευκολότερη αποθήκευση και αναζήτηση δεδομένων. Με την εξέλιξη της επιστήμης της πληροφορικής, και συγκεκριμένα με την εμφάνιση των υπολογιστών, οι βάσεις δεδομένων άλλαξαν πολύ γρήγορα αφού η συλλογή και η συντήρηση των δεδομένων μετατράπηκε σε μία πολύ εύκολη υπόθεση. (Berg, Seymour, & Goel, 2012)

Το 1960, ο Charles W. Bachman σχεδίασε το Ολοκληρωμένο Σύστημα Διαχείρισης Βάσης Δεδομένων, το "πρώτο" ΣΔΒΔ. Παράλληλα, η IBM δημιούργησε ένα δικό της σύστημα βάσης δεδομένων, γνωστό ως IMS. Και τα δύο ΣΔΒΔ περιγράφονται ως οι πρόδρομοι των βάσεων δεδομένων, που σκοπό είχαν την πλοήγηση. Η πρόσβαση στα δεδομένα γινόταν πηγαίνοντας από τη μια εγγραφή στην άλλη. Οι λεπτομέρειες αποθήκευσης εξαρτιούνταν από τον τύπο των δεδομένων. Έτσι, η προσθήκη ενός επιπλέον πεδίου στη βάση δεδομένων απαιτούσε την επανεγγραφή του υποκείμενου σχεδίου πρόσβασης / τροποποίησης. Έμφαση δόθηκε στην επεξεργασία των εγγραφών, όχι στη συνολική δομή του συστήματος. Το πρόβλημα με αυτού του είδους βάσεων δεδομένων ήταν το γεγονός ότι ένας χρήστης θα έπρεπε να γνωρίζει τη φυσική δομή της βάσης δεδομένων προκειμένου να ζητήσει πληροφορίες. (Berg, Seymour, & Goel, 2012) (Foote, 2017)

Μέχρι τα μέσα της δεκαετίας του 1960, έγιναν διαθέσιμα πολλά είδη συστημάτων βάσεων δεδομένων γενικής χρήσης. Δημιουργώντας την ανάγκη ενός προτύπου. Η ομάδα του Bachman ανέλαβε την ευθύνη για το σχεδιασμό και την τυποποίηση μιας γλώσσας που ονομάζεται Κοινή Επιχειρηματική Προσανατολισμένη Γλώσσα (COBOL) και το 1971

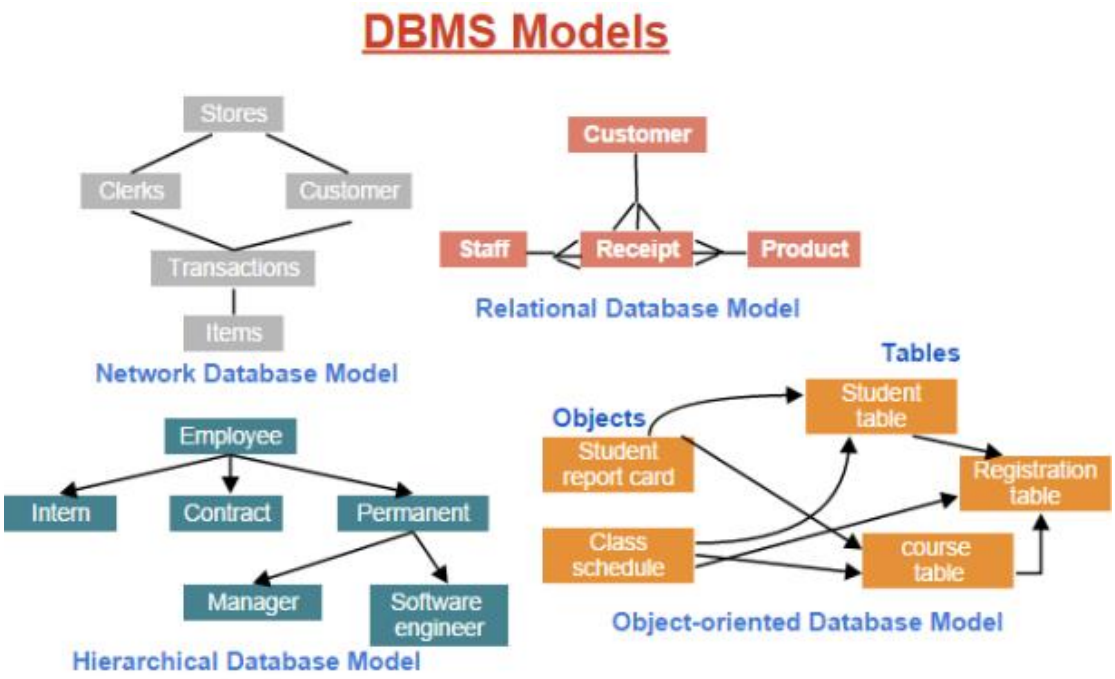
παρουσίασε το πρότυπο (προσέγγιση CODASYL). Βέβαια, αποδείχθηκε τελικά πως η συγκεκριμένη προσέγγιση ήταν περίπλοκη και απαιτούσε σημαντική εκπαίδευση για την χρήση του. (Berg, Seymour, & Goel, 2012) (Foote, 2017)

Παράλληλα, ο Edgar Codd, υπάλληλος της IBM στην ανάπτυξη συστημάτων σκληρού δίσκου το 1970 έγραψε μια σειρά εγγράφων, περιγράφοντας νέους τρόπους για την κατασκευή βάσεων δεδομένων. Οι ιδέες του τελικά εξελίχθηκαν σε ένα έγγραφο με τίτλο, *Σχεσιακό Μοντέλο Δεδομένων για Μεγάλες Κοινές Τράπεζες Δεδομένων*, το οποίο περιγράφει τη νέα μέθοδο αποθήκευσης δεδομένων και επεξεργασίας μεγάλων βάσεων δεδομένων. Τα αρχεία δεν αποθηκεύονταν σε μια λίστα ελεύθερων μορφών συνδεδεμένων αρχείων, όπως στο μοντέλο CODASYL, αλλά χρησιμοποιούσαν έναν "πίνακα με εγγραφές σταθερού μήκους", όπως χαρακτηριστικά αναφέρει. Αυτό είναι και το μοντέλο σχεσιακής βάσης δεδομένων. (Berg, Seymour, & Goel, 2012) (Foote, 2017)

Ένα νέο μοντέλο βάσης δεδομένων που ονομάζεται Entity-Relationship ή ER, προτάθηκε από τον P. Chen το 1976. Αυτό το μοντέλο επέτρεψε στους σχεδιαστές να επικεντρωθούν στην εφαρμογή δεδομένων αντί της λογικής δομής του πίνακα Ο όρος Relational Database Management System (RDBMS) σχεδιάστηκε κατά τη διάρκεια αυτής της περιόδου. (Berg, Seymour, & Goel, 2012)

Φτάνοντας στο 1980, έχουν ήδη αναπτυχθεί τρία μοντέλα οργάνωσης δεδομένων, το μοντέλο δικτύου (Network Database Model), το ιεραρχικό μοντέλο (Hieratical Database Model) και το σχεσιακό μοντέλο (Relational Database Model), όπως φαίνεται στο Σχήμα 4.1. Η SQL (Structured Query Language - δομημένη γλώσσα ερωτημάτων) έγινε το πρότυπο για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα.

Το 1990, εμφανίστηκε το ORDBMS (Object Relational Database Management System). Ο βασικός στόχος της βάσης δεδομένων σχεσιακών αντικειμένων ήταν να γεφυρωθεί το χάσμα μεταξύ των σχεσιακών βάσεων δεδομένων και των τεχνικών μοντελοποίησης αντικειμένων που χρησιμοποιούνται στις γλώσσες προγραμματισμού, όπως Java, C ++, Visual Basic .NET ή C #. Έτσι, μια εναλλακτική λύση για την επίτευξη μιας τέτοιας γέφυρας είναι η χρήση τυποποιημένων συστημάτων σχεσιακών βάσεων δεδομένων με κάποια μορφή λογισμικού αντικειμενοστραφούς χαρτογράφησης (ORM). (Berg, Seymour, & Goel, 2012)



Σχήμα 4.1: Μοντέλα Οργάνωσης Βάσεων δεδομένων

Φτάνοντας στο 2000, οι εταιρείες που κυριάρχησαν στο μεγαλύτερο μερίδιο αγοράς των βάσεων δεδομένων ήταν η IBM, Microsoft και Oracle.

4.4. Βάσεις Δεδομένων ή Υπολογιστικά φύλλα

Η βάση δεδομένων και τα υπολογιστικά φύλλα (excel) είναι διαφορετικά. Τι είναι ένα υπολογιστικό φύλλο; Είναι ένα ηλεκτρονικό βιβλίο ή μία ηλεκτρονική έκδοση των φύλλων εργασιών λογιστικής. Δημιουργήθηκε για να διευκολύνει τους ανθρώπους που χρειάζονται να αποθηκεύσουν τις λογιστικές τους πληροφορίες σε ψηφιακή μορφή πινάκων . Έτσι, είναι δυνατή η δημιουργία πινάκων σε ένα υπολογιστικό φύλλο. Αυτός είναι και ο λόγος που κάποιοι πιστεύουν ότι τα υπολογιστικά φύλλα και οι βάσεις δεδομένων είναι εναλλάξιμα, στην πραγματικότητα δεν είναι.

Έχουν τα εξής τρία όμοια χαρακτηριστικά

1. Μπορούν να περιέχουν μία μεγάλη ποσότητα πινάκων δεδομένων
2. Μπορούν να χρησιμοποιήσουν τα υπάρχοντα δεδομένα για να κάνουν υπολογισμούς.
3. Ούτε τα υπολογιστικά φύλλα ούτε η βάση δεδομένων χρησιμοποιούνται συνήθως από ένα μόνο άτομο, αντίθετα πολλοί χρήστες δουλεύουν με τα δεδομένα των πινάκων.

Οι διαφορές μεταξύ των δύο μορφών αποθήκευσης δεδομένων βρίσκονται στον τρόπο με τον οποίο αυτά τα τρία χαρακτηριστικά εφαρμόζονται.

Πίνακας 4.1: Διαφορές Υπολογιστικών Φύλλων και Βάσεων Δεδομένων

Υπολογιστικά Φύλλα	Βάσεις Δεδομένων
Κάθε κελί αντιμετωπίζεται ως μοναδική οντότητα. Μπορεί να αποθηκεύσει κάθε τύπου πληροφορίας, η οποία μπορεί να επεξεργαστεί	Μόνο ανεπεξέργαστα δεδομένα. Κάθε κελί περιέχει την μικρότερη δυνατή πληροφορία
Οποιοσδήποτε τύπος σε οποιοδήποτε κελί	Και κάθε πεδίο χαρακτηρίζεται από τον τύπο των δεδομένων. Με αυτή τη λειτουργία αποτρέπονται τυχαία λάθη (εάν ο χρήστης προσπαθήσει να εισάγει ένα δεδομένο άλλου τύπου τότε το λογισμικό εμφανίζει σφάλμα)
Τα δεδομένα αποθηκεύονται σε ένα κελί	Τα δεδομένα αποθηκεύονται σε μία καταγραφή ενός πίνακα
Οι πληροφορίες επιδέχονται μορφοποίηση	Οι πληροφορίες δεν επιδέχονται μορφοποίηση
Τα κελιά μπορούν να περιέχουν υπολογισμούς (συναρτήσεις και τύπος)	Όλοι οι υπολογισμοί και οι συναντήσεις βασίζονται στα υπάρχοντα δεδομένα
Έως 1000 γραμμές	>1000 γραμμές
Δεν υποστηρίζεται έλεγχος	Έλεγχο των δικαιωμάτων πρόσβασης και περιορισμός των χρηστών

Τα στοιχεία της βάσης δεδομένων που αναφέρθηκαν βελτιώνουν την ακεραιότητα των δεδομένων και αυτό είναι σημαντικό ειδικά σε μεγάλα σύνολα δεδομένων.

4.5. Διαφορές SQL και NoSQL

Η SQL (προφέρεται "ess-que-el") αντιπροσωπεύει τη δομημένη γλώσσα ερωτημάτων και χρησιμοποιείται για την επικοινωνία με μια βάση δεδομένων. Σύμφωνα με το ANSI (Αμερικανικό Εθνικό Ινστιτούτο Προτύπων), είναι η τυπική γλώσσα για τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων. Οι εντολές SQL χρησιμοποιούνται για την εκτέλεση εργασιών όπως η ενημέρωση ή η ανάκτηση δεδομένων από μια βάση δεδομένων. Το NoSQL βασικά αναφέρεται ως "μη-SQL" ή αλλιώς "μη σχεσιακή βάση δεδομένων. Παρέχει

στοιχεία για την αποθήκευση και βελτίωση των δεδομένων που διαμορφώνονται, εκτός από τις σχέσεις των πινάκων που εμπλέκονται στις σχεσιακές βάσεις δεδομένων.

Στον παρακάτω πίνακα παρατίθενται οι διαφορές ανάμεσα στην δημοφιλή NoSQL βάση δεδομένων και στις σχεσιακές βάσεις δεδομένων.

Πίνακας 4.2: Διαφορές Μεταξύ Σχεσιακών Βάσεων Δεδομένων και της MongoDB

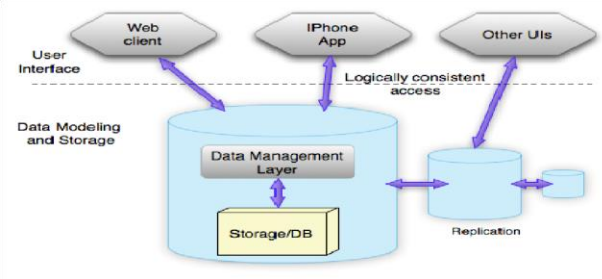
Σχεσιακή βάση δεδομένων	MongoDB	Διαφορές
Πίνακας	*Συλλογή*	Στη Σχεσιακή βάση δεδομένων, ο πίνακας περιέχει τις στήλες και τις σειρές που χρησιμοποιούνται για την αποθήκευση των δεδομένων, ενώ, στο MongoDB, αυτή η ίδια δομή είναι γνωστή ως συλλογή. Η συλλογή περιέχει έγγραφα τα οποία με τη σειρά τους περιέχουν Πεδία, τα οποία με τη σειρά τους είναι ζεύγη κλειδιών-τιμών.
Σειρά	*Έγγραφο*	Στη Σχεσιακή βάση δεδομένων, η σειρά αντιπροσωπεύει ένα απλό, δομημένο στοιχείο δεδομένων σε έναν πίνακα. Στο MongoDB, τα δεδομένα αποθηκεύονται σε έγγραφα.
Στήλη	*Πεδίο*	Στη Σχεσιακή βάση δεδομένων η στήλη υποδηλώνει ένα σύνολο τιμών δεδομένων. Αυτά στο MongoDB είναι γνωστά ως Πεδία.
Ενώσεις	*Ενσωματωμένα έγγραφα*	Στη Σχεσιακή βάση δεδομένων τα δεδομένα κατανέμονται σε διάφορους πίνακες και για να παρουσιάσουν μια πλήρη εικόνα όλων των δεδομένων, σχηματίζεται μερικές φορές ένωση σε πίνακες.. Στο MongoDB, τα δεδομένα αποθηκεύονται κανονικά σε μία μόνο συλλογή, αλλά διαχωρίζονται με τη χρήση ενσωματωμένων εγγράφων.

4.6. NoSQL

4.6.1. Εισαγωγή στις NoSQL Βάσεις Δεδομένων

Ο Carlo Strozzi χρησιμοποίησε τον όρο NoSQL το 1998 για να ονομάσει την χαλαρή σχεσιακή βάση δεδομένων ανοιχτού κώδικα που δεν εκθέτει την τυπική διασύνδεση SQL. Ο Eric Evans, ένας υπάλληλος του Rackspace, επανάφερε τον όρο NoSQL όταν ο Johan Oskarsson του Last.fm ήθελε να οργανώσει μια εκδήλωση για να συζητήσει κατανεμημένες βάσεις δεδομένων ανοιχτού κώδικα. (Berg, Seymour, & Goel, 2012)

Η αρχιτεκτονική των NoSQL βάσεων δεδομένων βασίζεται στον διαχωρισμό μεταξύ της διαχείρισης δεδομένων και της αποθήκευσης δεδομένων. Η προηγούμενη εποχή των βάσεων δεδομένων που βασίζονται σε SQL προσπάθησε να ικανοποιήσει αμφότερες τις ανησυχίες με τις βάσεις δεδομένων. Αυτό είναι πολύ δύσκολο και αναπόφευκτα οι εφαρμογές αναλαμβάνουν μέρος του καθήκοντος της διαχείρισης δεδομένων, παρέχοντας ορισμένα καθήκοντα επικύρωσης και προσθέτοντας λογική μοντελοποίησης. Μία από τις βασικές έννοιες του κινήματος NoSQL είναι να επικεντρωθεί η βάση δεδομένων στις εργασίες αποθήκευσης δεδομένων υψηλής απόδοσης και να παρέχει πρόσβαση χαμηλού επιπέδου σε ένα επίπεδο διαχείρισης δεδομένων με τρόπο που να επιτρέπει την εύκολη καταγραφή των καθηκόντων διαχείρισης δεδομένων στη γλώσσα προγραμματισμού.



Σχήμα 4.2: Αρχιτεκτονική NoSQL Βάσης Δεδομένων

Η πιο γνωστή NoSQL βάση δεδομένων είναι η MongoDB

4.6.2. MongoDB

Η MongoDB είναι μια NoSQL βάση δεδομένων προσανατολισμένη σε έγγραφα, που χρησιμοποιείται για αποθήκευση δεδομένων μεγάλου όγκου. Αναπτύχθηκε από την MongoDB Inc και η αρχική έκδοση ήταν το 2009. Το MongoDB γράφτηκε στις γλώσσες προγραμματισμού C, C ++, JavaScript και μπορεί να υποστηρίξει τα λειτουργικά συστήματα Windows Vista και αργότερα, Linux, OS X 10.7.

Τα βασικά χαρακτηριστικά της συγκεκριμένης βάσης δεδομένων είναι τα εξής:

- Κάθε βάση δεδομένων περιέχει συλλογές οι οποίες με τη σειρά τους περιέχουν έγγραφα. Κάθε έγγραφο μπορεί να διαφέρει με διαφορετικό αριθμό πεδίων. Το μέγεθος και το περιεχόμενο κάθε εγγράφου μπορεί να διαφέρουν μεταξύ τους.
- Η δομή του εγγράφου είναι περισσότερο σύμφωνη με τον τρόπο με τον οποίο οι προγραμματιστές κατασκευάζουν τις κλάσεις και τα αντικείμενα τους στις αντίστοιχες γλώσσες προγραμματισμού τους. Για τους προγραμματιστές οι τάξεις τους δεν είναι σειρές και στήλες, αλλά έχουν σαφή δομή με ζεύγη βασικής αξίας.
- Όπως φαίνεται στην εισαγωγή με τις βάσεις δεδομένων NoSQL, οι σειρές (ή τα έγγραφα που ονομάζονται στο MongoDB) δεν χρειάζεται να έχουν προκαθορισμένη διάταξη. Αντ' αυτού, τα πεδία μπορούν να δημιουργηθούν εν πτήση.
- Το μοντέλο δεδομένων που είναι διαθέσιμο στο MongoDB επιτρέπει την πιο εύκολη αναπαράσταση των ιεραρχικών σχέσεων, την αποθήκευση σε συστοιχίες και άλλες πιο σύνθετες δομές.
- Τα περιβάλλοντα MongoDB είναι πολύ κλιμακούμενα. Οι εταιρείες σε όλο τον κόσμο έχουν ορίσει συμπλέγματα με μερικά από αυτά να τρέχουν 100+ κόμβους με περίπου εκατομμύρια έγγραφα μέσα στη βάση δεδομένων (Vardhan & Rahamathulla, 2018)

4.7. SQL Βάσεις Δεδομένων

4.7.1. Εισαγωγή για τις SQL Βάσεις Δεδομένων

Σε αυτό το κεφάλαιο θα αναλυθούν οι διάφορες Σχεσιακές Βάσεις Δεδομένων, δηλαδή Βάσεις που βασίζονται στην γλώσσα SQL. Ορισμένα κοινά συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων που χρησιμοποιούν SQL είναι: Oracle, Sybase, Microsoft SQL Server, Access, Ingres κλπ. Παρόλο που τα περισσότερα συστήματα βάσεων δεδομένων χρησιμοποιούν SQL, οι περισσότεροι από αυτούς έχουν επίσης τις δικές τους πρόσθετες επεκτάσεις που συνήθως χρησιμοποιούνται μόνο στο σύστημά τους . Ωστόσο, οι τυποποιημένες εντολές SQL όπως "Επιλογή", "Εισαγωγή", "Ενημέρωση", "Διαγραφή", "Δημιουργία" και "Πτώση" μπορούν να χρησιμοποιηθούν για την επίτευξη σχεδόν όλων των σκοπών με μια βάση δεδομένων. Τα βασικά χαρακτηριστικά των κλασσικών βάσεων δεδομένων, παρέχουν τις εγγυήσεις ACID (Ατομικότητα, Συνέπεια, Απομόνωση, Ανθεκτικότητα).

4.7.2. MS Access

Η Microsoft Access είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) από τη Microsoft. Προσφέρει ένα συνδυασμό γραφικών διεπαφών χρήστη (GUI) και εργαλείων

ανάπτυξης εφαρμογών με τη σχεσιακή μηχανή βάσης δεδομένων Microsoft Jet. Η MS Access αναπτύχθηκε από τη Microsoft και κυκλοφόρησε τον Νοέμβριο του 1992. Υποστηρίζει το λειτουργικό σύστημα των Windows.

Η Microsoft Access διαθέτει άδεια χρήσης ως μέρος της σουίτας εφαρμογών του Microsoft Office. Με την Access, οι χρήστες μπορούν να σχεδιάζουν πίνακες και φόρμες και να δημιουργούν σύνθετα ερωτήματα, χωρίς να έχουν εμπειρία στον προγραμματισμό. Οι προγραμματιστές λογισμικού ή οι αναλυτές δεδομένων μπορούν να δημιουργήσουν εφαρμογές λογισμικού χρησιμοποιώντας την Access ως βάση δεδομένων back-end. Τυπικά, οι εφαρμογές της Access αναπτύσσονται χρησιμοποιώντας Visual Basic for Applications (VBA) και άλλες εφαρμογές της Microsoft.

Μερικά αξιοσημείωτα χαρακτηριστικά της Access:

- Χρησιμοποιεί το δικό της αρχείο αποθήκευσης βάσεων δεδομένων.
- Μπορεί να χρησιμοποιεί πολλαπλές πηγές back-end.
- Η Access είναι μια εφαρμογή που μπορεί να χρησιμοποιηθεί για τη δημιουργία και διαχείριση βάσεων δεδομένων σε προσωπικούς υπολογιστές. Μπορεί να χρησιμοποιηθεί ως εργαλείο απεικόνισης, δημιουργίας φορμών και αναφορών χρησιμοποιώντας τα υποκείμενα δεδομένα.
- Μπορεί επίσης να χρησιμοποιηθεί ως το «front-end» ενός προγράμματος, ενώ άλλα προϊόντα λειτουργούν ως πίνακες "back-end", όπως Microsoft SQL Server και προϊόντα εκτός της Microsoft, όπως το Oracle και το Sybase. Ομοίως, ορισμένες εφαρμογές όπως η Visual Basic, το ASP.NET ή το Visual Studio .NET μπορούν να χρησιμοποιήσουν τη βάση δεδομένων της Access για τους πίνακες και τα queries της βάσης δεδομένων.
- Μπορεί επίσης να συμπεριληφθεί ως μέρος μιας σύνθετης λύσης, όπου διασυνδέεται με άλλες εφαρμογές και τεχνολογίες όπως τα Microsoft Excel, Microsoft Outlook, Microsoft Word, Microsoft PowerPoint και ActiveX. (Vardhan & Rahamathulla, 2018)

Η MS Access χρησιμοποιείται από μικρές επιχειρήσεις, σε τμήματα μεγάλων εταιρειών, και από προγραμματιστές για χόμπι για τη δημιουργία ad hoc προσαρμοσμένων desktop συστημάτων για το χειρισμό των δεδομένων. Μερικοί επαγγελματίες προγραμματιστές εφαρμογών χρησιμοποιούν την Access για γρήγορη ανάπτυξη εφαρμογών, ειδικά για τη δημιουργία πρωτοτύπων και αυτόνομων εφαρμογών που χρησιμεύουν ως εργαλεία για πωλητές on-the-road. (Bassil, 2011)

4.7.3. SQLite

Το SQLite είναι ένα ανοικτού κώδικα, ενσωματωμένο, σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων, σχεδιασμένο γύρω στο 2000. Είναι μια “ελαφριά βάση δεδομένων”, εξού και το όνομα της lite, με μηδενική διαμόρφωση, χωρίς απαιτήσεις ενός διακομιστή ή εγκατάστασης. Παρά την απλότητα της, ενσωματώνει δημοφιλή χαρακτηριστικά των συστημάτων διαχείρισης βάσεων δεδομένων.

Το SQLite είναι πολύ ελαφρύ (είναι μικρότερο από 500Kb) σε σύγκριση με άλλα συστήματα διαχείρισης βάσεων δεδομένων όπως το SQL Server ή το Oracle. Δεν είναι σύστημα διαχείρισης βάσεων δεδομένων πελάτη-διακομιστή. Αποτελεί μια βάση δεδομένων εντός μνήμης (in memory database), γι' αυτό το λόγο δεν απαιτείται ούτε εγκατάσταση ούτε διαμόρφωση. Μία βάση δεδομένων εντός μνήμης είναι ένα σύστημα διαχείρισης βάσεων δεδομένων που βασίζεται κυρίως στην κύρια μνήμη για αποθήκευση δεδομένων υπολογιστή. Μια τυπική βάση δεδομένων SQLite περιέχεται σε ένα μόνο αρχείο στο χώρο αποθήκευσης του υπολογιστή με όλα τα αντικείμενα βάσης δεδομένων (πίνακες, προβολές, ενεργοποιητές κ.λπ.) που περιλαμβάνονται σε αυτό το αρχείο. Δεν απαιτείται ειδικός διακομιστής.

Η SQLite είναι ιδανική βάση δεδομένων σε έξι περιπτώσεις, των οποίων χαρακτηριστικό γνώρισμα είναι η ελαφρότητα.

- Όταν γίνεται ανάπτυξη ενσωματωμένου λογισμικού για συσκευές όπως τηλεοράσεις, κινητά τηλέφωνα, κάμερες, οικιακές ηλεκτρονικές συσκευές κ.λπ.
- Το SQLite μπορεί να χειριστεί αιτήματα HTTP χαμηλής έως μέσης επισκεψιμότητας και να διαχειριστεί πολύπλοκες πληροφορίες
- Στην αποθήκευση ενός αρχείου αρχείων, το SQLite μπορεί να παράγει αρχεία μικρότερου μεγέθους και με λιγότερα μετά-δεδομένα που περιλαμβάνονται από τα κανονικά αρχεία ZIP.
- Στην επεξεργασία ορισμένων δεδομένων μέσα σε μια εφαρμογή, μπορεί να χρησιμοποιηθεί το SQLite ως προσωρινό σύνολο δεδομένων. Φορτώνοντας τα δεδομένα σε μια βάση δεδομένων SQLite in-memory, εκτελώντας τα queries και εξάγοντας τα δεδομένα με τη επιθυμητή μορφή εμφάνισης στην εφαρμογή.
- Παρέχει έναν εύκολο και αποτελεσματικό τρόπο επεξεργασίας χρησιμοποιώντας μεταβλητές εντός μνήμης. Για παράδειγμα, όταν αναπτύσσεται ένα πρόγραμμα όπου πρέπει να γίνουν κάποιοι υπολογισμοί σε ορισμένα αρχεία. Δημιουργώντας μια

βάση δεδομένων SQLite, εισάγοντας τις εγγραφές εκεί, και με μόνο ένα query να επιλεχθούν οι εγγραφές και να εκτελεστούν οι υπολογισμοί.

Τέλος, για σκοπούς μάθησης και κατάρτισης, το SQLite είναι μια καλή εφαρμογή, αφού δεν απαιτείται εγκατάσταση και μορφοποίηση, απλά μια αντιγραφή των βιβλιοθηκών SQLite.

Πλεονεκτήματα και Μειονεκτήματα SQLite

Το SQLite είναι μια ανοιχτή πηγή και δεν απαιτείται εμπορική άδεια (είναι δωρεάν). Είναι σύστημα διαχείρισης βάσεων δεδομένων μεταξύ των πλατφορμών. Μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα λειτουργικών συστημάτων όπως Windows, Mac OS, Linux και Unix. Το SQLite προσφέρει έναν αποτελεσματικό τρόπο αποθήκευσης δεδομένων, το μήκος των στηλών είναι μεταβλητό και δεν είναι σταθερό. Έτσι το SQLite θα διαθέσει μόνο χώρο που χρειάζεται ένα πεδίο. Για παράδειγμα, εάν μια στήλη με τύπο δεδομένων varchar (200) διαθέτει 200 χαρακτήρες και η εγγραφή είναι 10 χαρακτήρων, τότε το SQLite θα διαθέσει μόνο 10 χαρακτήρες για αυτήν την τιμή και όχι για ολόκληρο το 200 χώρο. Τέλος, όσον αφορά τις διαδικτυακές υπηρεσίες, το SQLite παρέχει API για ένα ευρύ φάσμα γλώσσας προγραμματισμού.

Το SQLite δεν υποστηρίζει πολλές λειτουργίες και εντολές άλλων βάσεων δεδομένων, αυτό την καθιστά περιορισμένης χρήσης. Ενώ για μεγάλα δεδομένα πρέπει να η εσωτερική μνήμη να είναι πολύ μεγάλη. Για επιχειρηματική χρήση είναι ακατάλληλη.

4.7.4. My SQL

Το MySQL είναι ένα δωρεάν σύστημα διαχείρισης βάσεων δεδομένων SQL, με ανοικτή πηγή, πολλαπλών χρήσεων και πολλαπλών χρηστών, το οποίο διαθέτει περισσότερες από 10 εκατομμύρια εγκαταστάσεις. Το βασικό πρόγραμμα λειτουργεί ως διακομιστής που παρέχει πρόσβαση σε πολλούς χρήστες σε διάφορες βάσεις δεδομένων.

Η MySQL περιλαμβάνει ένα ευρύ υποσύνολο του ANSI SQL 99, καθώς και επεκτάσεις, υποστήριξης μέσω πλατφόρμας, αποθηκευμένες διαδικασίες, ενεργοποιητές, δρομείς, ενημερώσιμες προβολές και υποστήριξη επεξεργασίας κατανεμημένων συναλλαγών X / Open XA. Επιπλέον, υποστηρίζει μηχανισμούς ανεξάρτητης αποθήκευσης, υποστήριξη SSL, προσωρινή αποθήκευση queries, ενσωματωμένη βιβλιοθήκη βάσεων δεδομένων και συμβατότητα με το ACID χρησιμοποιώντας τους μηχανισμούς InnoDB συμπλεγμάτων.

4.7.5. Oracle

Η Oracle αναφέρεται ως RDBMS και πρόκειται για ένα πολύ-μοντέλο σύστημα διαχείρισης βάσεων δεδομένων που διαχειρίζεται η Oracle Corporation. Η Oracle αναπτύχθηκε από τον

Larry Ellison και τους Bob Miner και Edoates στα εργαστήρια ανάπτυξης λογισμικού (SDL) το 1977.

Η Oracle έχει κυκλοφορήσει πολλές εκδόσεις όπως το Oracle 10g, το Oracle 11g, το Oracle 11i, όπου το g σημαίνει "πλέγμα" και το Internet. Η τελευταία έκδοση είναι 12c, το c σημαίνει "σύννεφο". Το Oracle 12c έρχεται σε 3 εκδόσεις για την εγκατάσταση όπως

1. Oracle Database Enterprise Edition 2 (EE): Η Oracle είναι η κλιμάκωση της βιομηχανικής δράσης τόσο σε σύμπλεγμα όσο και σε ενιαία σύνθεση συστημάτων, χωρίς περιορισμούς στους πόρους του διακομιστή.
2. Oracle DB πρότυπο έκδοσης 2 (SE2) που προορίζεται για εφαρμογές μικρού έως μεσαίου μεγέθους περιλαμβάνει επίσης πραγματικές συστάδες εφαρμογών.
3. Oracle DB προσωπική έκδοση (PE) είναι συμβατή με τα SE2 και EE

Η Oracle Corporation παρέχει ενημερώσεις έκδοσης κώδικα ή ενημερώσεις κώδικα ασφαλείας που εμποδίζουν την κλοπή δεδομένων. Η Oracle είναι η πιο δημοφιλή βάση δεδομένων στον κόσμο για τη διεξαγωγή επεξεργασίας ηλεκτρονικών συναλλαγών και αποθήκευσης δεδομένων. Η Oracle χρησιμοποιείται από σχεδόν όλες τις μεγάλες εφαρμογές και κυρίως από τον Τραπεζικό τομέα. Προσφέρει έναν ισχυρό συνδυασμό τεχνολογίας και ολοκληρωμένων, προ συσκευασμένων επιχειρηματικών εφαρμογών, συμπεριλαμβανομένων βασικών λειτουργιών. Δεν έχει βρόχους, συνθήκες, συστοιχίες, δρομείς και πίνακες temp που είναι πολύ αργές λειτουργίες και καταναλώνουν πόρους. (Vardhan & Rahamathulla, 2018) Ένα στιγμιότυπο περιλαμβάνει ένα σύνολο διαδικασιών λειτουργικού συστήματος και δομών μνήμης που αλληλοεπιδρούν με την αποθήκευση. Εκτός από την αποθήκευση, η βάση δεδομένων αποτελείται από ηλεκτρονικά αρχεία καταγραφής και επαναφοράς που διατηρούν το ιστορικό συναλλαγών. Οι διαδικασίες μπορούν με τη σειρά τους να αρχειοθετήσουν τα αρχεία καταγραφής Online redo σε αρχεία καταγραφής αρχείων, τα οποία παρέχουν τη βάση για ανάκτηση δεδομένων και για ορισμένες μορφές αναπαραγωγής δεδομένων. (Bassil, 2011)

Το Oracle RDBMS αποθηκεύει δεδομένα λογικά με τη μορφή χώρων πίνακα και φυσικά με τη μορφή αρχείων δεδομένων. Σε φυσικό επίπεδο, τα αρχεία δεδομένων περιλαμβάνουν ένα ή περισσότερα μπλοκ δεδομένων, όπου το μέγεθος του μπλοκ μπορεί να διαφέρει μεταξύ των αρχείων δεδομένων. Το Oracle διαθέτει λεξικό δεδομένων, ευρετήρια και συμπλέγματα. (Bassil, 2011) (Vardhan & Rahamathulla, 2018)

Οι μετέπειτα εκδόσεις του 10g, έδωσαν την δυνατότητα υπολογιστικής πλέγματος, στις οποίες μια εφαρμογή Instance μπορεί να χρησιμοποιήσει πόρους CPU από άλλο κόμβο στο δίκτυο. (Bassil, 2011)

4.8. Πρόταση για το σύστημα

Είναι φανερό ότι υπάρχει μία πληθώρα βάσεων δεδομένων τις οποίες από τις οποίες μπορεί να γίνει η επιλογή για το υπάρχον σύστημα. Αρχικά, επιλέγεται η χρήση μιας σχεσιακής βάσης δεδομένων και όχι μιας NoSQL, διότι η χρήση των σχεσιακών βάσεων είναι ευρύτερη και πιο οικεία με τους περισσότερους χρήστες. Επίσης, η MongoDB, η γνωστότερη NoSQL βάση δεδομένων, είναι προσανατολισμένη σε έγγραφα, και χρησιμοποιείται για αποθήκευση δεδομένων μεγάλου όγκου. Το υπο ανάλυση σύστημα δεν έχει βασικό στόχο μόνο την αποθήκευση δεδομένων μεγάλου όγκου, αλλά και την διαρκή μεταξύ τους σχέση, επομένως μια NoSQL βάση δεν μπορεί να εξυπηρετήσει αυτόν τον σκοπό.

Όσον αφορά τις SQL βάσεις δεδομένων, η MS Access χρησιμοποιείται από μικρές επιχειρήσεις ή απο τμήματα μεγάλων επιχειρήσεων, επομένως δεν θεωρείται κατάλληλη για το υπο ανάλυση σύστημα που χρειάζεται μια στοιβαρή και αρκετά δυναμική βάση. H SQLite είναι πολύ ελαφριά (είναι μικρότερη από 500Kb) σε σύγκριση με άλλα συστήματα διαχείρισης βάσεων δεδομένων όπως το SQL Server ή το Oracle, και δεν αντέχει τον όγκο πληροφοριών και δεδομένων που υπεισέρχονται στο σύστημα. Ακόμα, το MySQL είναι ένα δωρεάν ανοιχτό σύστημα διαχείρισης βάσεων δεδομένων SQL, με πολλαπλές χρήσεις και πολλαπλούς χρήστες, το οποίο δεν μπορεί να διαθέσει την ασφάλεια και την κλιμάκωση που χρειάζεται το σύστημα. Επομένως, προτείνεται ως βάση δεδομένων για το λογισμικό ως υπηρεσία η Oracle, η οποία λύνει απαιτητικά επιχειρηματικά προβλήματα και παρέχει μεγάλη αξιοπιστία, κλιμάκωση και απόδοση, για βάσεις δεδομένων και εφαρμογές. Όπως θα παρουσιαστεί και στο κεφάλαιο 6.8.2 (Oracle Cloud), η σύνδεση με το Cloud είναι άμεση αν χρησιμοποιείται ήδη η Oracle.

5. Μοντέλα Παροχής Λογισμικού

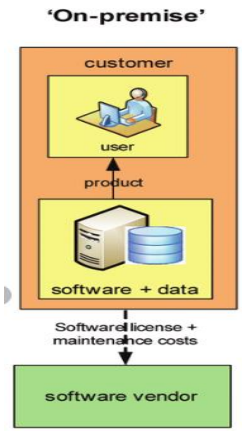
5.1. Εισαγωγή

Τα τελευταία χρόνια έχει αλλάξει πολύ ο τρόπος με τον οποίο παρέχονται το λογισμικό και οι εφαρμογές στους χρήστες και τις εταιρείες. Το παραδοσιακό μοντέλο απόκτησης του λογισμικού, γνωστό και ως on-premises Software, απαρτίζεται από την λήψη και εγκατάσταση του λογισμικού σε έναν ηλεκτρονικό υπολογιστή ή σε ένα ν διακομιστή. Την διαδοχή πήρε το off-site hosting Software, το οποίο αποτελεί μια εναλλακτική λύση όπου το «εξειδικευμένο» υλικό βρίσκεται σε άλλο κτίριο από τους τελικούς χρήστες. Πριν από το 2005, τα μοντέλα αυτά ήταν η πιο κοινή μορφή παροχής λογισμικού σε άτομα και επιχειρήσεις. Ωστόσο, μια άλλη μέθοδος παροχής λογισμικού πρότασσε αρκετά πλεονεκτήματα σε σχέση με τη συμβατική μέθοδο, αυτή είναι το Λογισμικό ως Υπηρεσία (SaaS) που αποτελεί έναν ταχέως αναπτυσσόμενο κλάδο στην βιομηχανία της πληροφορικής. Συγκεκριμένα, το λογισμικό παραδίδεται στον τελικό χρήστη μέσω του Διαδικτύου, χωρίς εγκατάσταση του λογισμικού στον υπολογιστή ή στον διακομιστή. Το κεφάλαιο 5, πραγματεύεται αυτά τα τρία μοντέλα παροχής λογισμικού. Συγκεκριμένα, η πρώτη ενότητα αφορά την τοπική εγκατάσταση του λογισμικού στην εταιρεία, στη συνέχεια η επόμενη ενότητα αφορά το φιλοξενούμενο λογισμικό δηλαδή την εναλλακτική λύση της τοπικής εγκατάστασης και στην τελευταία ενότητα αναφέρεται η νεότερη μέθοδο διανομής του λογισμικού, το μοντέλο του λογισμικού ως υπηρεσίας μέσω του διαδικτύου. Σε κάθε ενότητα αξιολογούνται τα πλεονεκτήματα και τα μειονεκτήματα κάθε μεθόδου και στο τέλος επιλέγεται η καλύτερη εξ αυτών, που θα χρησιμοποιηθεί στο σύστημα.

5.2. Τοπική Εγκατάσταση Λογισμικού στην Εταιρία

5.2.1. Γενικά Στοιχεία Μοντέλου Τοπικής Εγκατάστασης

Πριν από την τεχνολογική ανάπτυξη των δικτύων κυρίως του διαδικτύου, η μόνη λύση που μπορούσαν να υιοθετήσουν οι εταιρίες ήταν η εγκατάσταση του λογισμικού στην εσωτερική υποδομή της εταιρίας, δηλαδή στους διακομιστές ή/και στους υπολογιστές της. Η τοπική εγκατάσταση χρησιμοποιεί τους εγγενείς πόρους υπολογιστών της εταιρίας και απαιτεί ένα αντίγραφο λογισμικού που έχει λάβει άδεια ή έχει αγοραστεί εξολοκλήρου από έναν ανεξάρτητο πωλητή λογισμικού.



Σχήμα 5.1: Τοπική Εγκατάσταση Λογισμικού

Συνήθως για μια τέτοια λύση απαιτούνται μεγάλες προκαταβολικές αγορές, με αποτέλεσμα οι εταιρίες να καταλήγουν σε μεγάλες κεφαλαιουχικές δαπάνες. Ωστόσο, πληρώνουν μόνο μία φορά για τις άδειες χρήσης, οπότε το Συνολικό Κόστος Ιδιοκτησίας της λύσης μπορεί να είναι χαμηλότερο από την εφαρμογή σε υπολογιστικό νέφος. Βέβαια, αυτό το γεγονός δεν σημαίνει ότι η τοπική εγκατάσταση λογισμικού δεν έχει κανένα λειτουργικό κόστος, αν αναλογιστεί κανείς το γεγονός ότι πρέπει να αντιμετωπίζουν με κάποιο τρόπο τη συντήρησή του. Το λογισμικό εγκαθίσταται και λειτουργεί σε υπολογιστές στις εγκαταστάσεις της εταιρίας, πράγμα που σημαίνει ότι η εταιρία έχει καλύτερο έλεγχο για το πώς και πότε το σύστημα είναι διαθέσιμο. Επίσης, ένας σημαντικός λόγος που πολλές επιχειρήσεις προτιμούν την τοπική εγκατάσταση είναι η ασφάλεια των δεδομένων τους, η οποία εξαρτάται αποκλειστικά από τις συνθήκες της.

5.2.2. Πλεονεκτήματα Τοπικής Εγκατάστασης Λογισμικού

- Μεγαλύτερος έλεγχος του λογισμικού
- Διατήρηση της διεύθυνσης IP - διεύθυνση διαδικτυακού πρωτοκόλλου- μέσα στην εταιρία
- Υλοποίηση σημαντική προσαρμογής του λογισμικού στις απαιτήσεις της εταιρίας
- Ασφάλεια δεδομένων

5.2.3. Μειονεκτήματα Τοπικής Εγκατάστασης Λογισμικού

- Υψηλό κόστος
- Απαιτεί την δημιουργία και συντήρηση μιας υποδομής πληροφορικής εντός της εταιρίας
- Απαιτούνται τεχνική υποστήριξη, αναβαθμίσεις και έλεγχος εκδόσεων από την εταιρία, με έλεγχο των αναβαθμίσεων για τυχόν επιπλοκές με τις προηγούμενες προσαρμογές

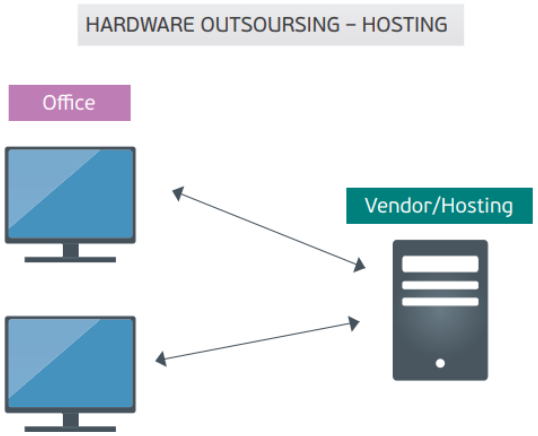
- Δύσκολη και δαπανηρή η ενσωμάτωση με εφαρμογές τρίτων
- Απαιτεί περισσότερο χρόνο ανάπτυξης

5.3. «Φιλοξενούμενο» Λογισμικό

5.3.1. Γενικά Στοιχεία Μοντέλου «Φιλοξενούμενου» Λογισμικού

Όπως αναφέρθηκε και ανωτέρω μια εναλλακτική λύση της τοπικής εγκατάστασης, αναπτύχθηκε την δεκαετία του 90’ με την διείσδυση του Διαδικτύου στις επιχειρήσεις. Πιο συγκεκριμένα, το off-hosting software, όπως ονομάζεται στα αγγλικά αυτός ο τρόπος, είναι ένα μοντέλο παροχής λογισμικού όπου αυτό βρίσκεται στο σύστημα του πωλητή -παρόχου και προσπελάζεται από τους χρήστες μέσω ενός προγράμματος περιήγησης ιστού. Έτσι, οι προμηθευτές λογισμικών φιλοξενούν πολλούς διαφορετικούς πελάτες δίνοντας τους μόνο την άδεια χρήσης του λογισμικού. Ένα φιλοξενούμενο λογισμικό βρίσκεται σε ένα τρίτο κέντρο δεδομένων, την τοποθεσία φιλοξενίας, την οποία διαχειρίζεται επιχειρησιακά ένας φορέας παροχής υπηρεσιών εξ ονόματος του δικαιούχου. Μια εξειδικευμένη υπηρεσία που προσφέρεται από το συγκεκριμένο μοντέλο είναι το μοντέλο παροχής υπηρεσιών εφαρμογών (ASP), όπου ο προμηθευτής παρέχει πρόσθετη εμπειρία στον τομέα.

Η πρόταση αξίας του «φιλοξενούμενου» λογισμικού είναι ότι το κόστος για την ιδιοκτησία και τη λειτουργία του λογισμικού είναι πολύ μικρότερο αν ανατεθεί σε κάποιον τρίτο με την υποδομή και την γνώση που έχει, έναντι της τοπικής εγκατάστασης του. Μεγάλο μέρος του γενικού κόστους του τομέα της πληροφορικής μεταφέρεται στον φορέα φιλοξενίας, αφού δεν χρειάζεται πλέον η δημιουργία, η στελέχωσή και η διατήρηση ενός κέντρου δεδομένων ή/και γραφείου υποστήριξης πληροφορικής.



Σχήμα 5.2: Διακομιστής και Τερματικά του «Φιλοξενούμενου» Λογισμικού

5.3.2. Πλεονεκτήματα «Φιλοξενούμενου» Λογισμικού

- Ο πάροχος επιβαρύνεται με το κόστος υποδομής
- Δυνητικά χαμηλότερο κόστος από την τοπική εγκατάσταση

5.3.3. Μειονεκτήματα «Φιλοξενούμενου» Λογισμικού

- Πολύ περιορισμένη προσαρμογή στις ανάγκες της εταιρίας
- Οι αναβαθμίσεις ενδέχεται να σπάσουν τυχόν προηγούμενες προσαρμογές
- Η υποστήριξη και η εξυπηρέτηση είναι συνήθως μια επιπλέον χρέωση
- Η ενσωμάτωση με εφαρμογές τρίτων μπορεί να είναι δύσκολη / δαπανηρή
- Η ενιαία μίσθωση περιορίζει τις ευκαιρίες μείωσης του κόστους
- Γενικά χαμηλή χρηστικότητα
- Προβλήματα ασφάλειας δεδομένων

5.4. Μοντέλο Παροχής Λογισμικού ως Υπηρεσία

5.4.1. Γενικά Στοιχεία Μοντέλου Λογισμικό ως Υπηρεσία

Το μοντέλο παροχής του Λογισμικού με χρήση του Λογισμικού ως Υπηρεσία αναφέρεται σε μια μέθοδο παροχής φιλοξενούμενων εφαρμογών λογισμικού σε πελάτες σε ένα περιβάλλον υπολογιστικού νέφους. Στην πλειονότητα των περιπτώσεων, οι πελάτες υποχρεούνται να καταβάλλουν μηνιαία ή ετήσια συνδρομή στον πάροχο υπηρεσιών υπολογιστικού νέφους για να συνεχίσουν να χρησιμοποιούν μια συγκεκριμένη υπηρεσία λογισμικού. Η παράδοση του Saas αποτελεί τη ραχοκοκαλιά ενός αναδυόμενου μοντέλου λογισμικού που εξαλείφει την ανάγκη των πελατών να επιβαρύνονται με έξοδα αγοράς ή συντήρησης ακριβών διακομιστών, εφαρμογών και λογισμικού. Το μοντέλο παροχής του λογισμικού ως Υπηρεσία επιτρέπει στις επιχειρήσεις να μειώνουν σημαντικά το κόστος χρήσης και συντήρησης λογισμικού, καθώς πλέον ο πάροχος υπηρεσιών του cloud είναι αυτός επιβαρύνεται την ανάπτυξη και συντήρηση των εφαρμογών και του υλικού στο οποίο εκτελούνται οι εφαρμογές. Με το μοντέλο παροχής του λογισμικού ως Υπηρεσία, τα κεφάλαια και τα λειτουργικά έξοδα, που συνήθως συνδέονται με τη φιλοξενία εφαρμογών λογισμικού επί τόπου, μετατοπίζονται από τον πελάτη στον πάροχο υπηρεσιών λογισμικού cloud. Ο προμηθευτής του Λογισμικού είναι σε θέση να κατανείμει τα κεφάλαια και τα λειτουργικά του έξοδα σε μια μεγάλη βάση πελατών.

5.4.2. Ορισμοί Λογισμικού ως Υπηρεσία

Στη διεθνή βιβλιογραφία και αρθρογραφία, συναντά κανείς ποικίλους ορισμούς του μοντέλου SaaS. Το ακρωνύμιο SaaS πρωτοεμφανίστηκε το 2001 στον ορισμό του από το Software & Information Industry’s (SIIA) eBusiness Division σύμφωνα με τον οποίο

" Στο μοντέλο του Λογισμικού ως Υπηρεσία, η εφαρμογή ή η Υπηρεσία, έχει αναπτυχθεί από ένα συγκεντρωτικό κέντρο δεδομένων δια μέσου ενός δικτύου - Internet, Intranet, LAN, or VPN – χορηγώντας πρόσβαση και χρήση σε επαναλαμβανόμενη επί πληρωμή βάση."

Ένας άλλος ορισμός που δίνεται από τους John Rittinghouse και James Ransome είναι

"Το λογισμικό ως υπηρεσία (SaaS) είναι ένα μοντέλο διανομής λογισμικού στο οποίο οι εφαρμογές φιλοξενούνται από έναν προμηθευτή ή πάροχο υπηρεσιών και τίθενται στη διάθεση των πελατών μέσω ενός δικτύου, συνήθως του Διαδικτύου" (Rittinghouse & Ransome, August 17, 2009)

Επίσης, από τον Ho-fung Leung δίνεται ένας άλλος ορισμός :

"Το λογισμικό ως υπηρεσία είναι ένα μοντέλο ανάπτυξης λογισμικού, με το οποίο ένας πάροχος επιτρέπει μια εφαρμογή στους πελάτες για χρήση ως υπηρεσία κατόπιν αιτήσεως" (Ho-fung, September 30, 2010)

Τέλος, σύμφωνα με τους Wei-Tek Tsai, Xin Sun, Janaka Balasooriya

"Στο SaaS, το Λογισμικό εμφανίζεται στους τελικούς χρήστες ως υπηρεσία κατόπιν ζήτησης, συνήθως σε ένα πρόγραμμα περιήγησης. Διευκολύνει τους χρήστες από τα προβλήματα ανάπτυξης και συντήρησης λογισμικού. Το λογισμικό μοιράζεται συχνά από πολλούς ενοικιαστές, ενημερώνεται αυτόματα από τα «νέφη» και δεν χρειάζεται να αγοραστεί επιπλέον άδεια. Διάφορα χαρακτηριστικά μπορούν να ζητηθούν κατόπιν παραγγελίας και η αναβάθμισή τους είναι αρκετά συχνή." (Tsai, Sun, & Balasooriya, 2010)

Όπως είναι εμφανές οι παραπάνω ορισμοί έχουν τέσσερα βασικά χαρακτηριστικά. Το SaaS είναι ένα μοντέλο υπηρεσιών μέσω διαδικτύου, που παρέχεται από τον πάροχο στον πελάτη μέσω δικτύου. Η υπηρεσία εξαρτάται από τις χρεώσεις με βάση τις οποίες χορηγούνται οι αντίστοιχες άδειες χρήσης. Με βάση τους ορισμούς που αναπτύχθηκαν, το Λογισμικό ως Υπηρεσία μπορεί θεωρηθεί ως επιχειρηματικό μοντέλο. Όμως, τι είναι ένα επιχειρηματικό μοντέλο και ποια είναι τα στοιχεία του Λογισμικού ως Υπηρεσία που το καθιστούν ως ένα από αυτά.

5.4.3. Πλεονεκτήματα Παροχής Λογισμικού ως Υπηρεσία

- Ευελιξία έτσι ώστε να προσαρμόζεται η λύση στις διαφοροποιούμενες ανάγκες
- Εύκολα προσβάσιμο, οπουδήποτε, οποτεδήποτε, σε οποιαδήποτε συσκευή
- Μειωμένο κόστος για το υλικό και το λογισμικό, άρα και μικρότερο συνολικό κόστος ιδιοκτησίας

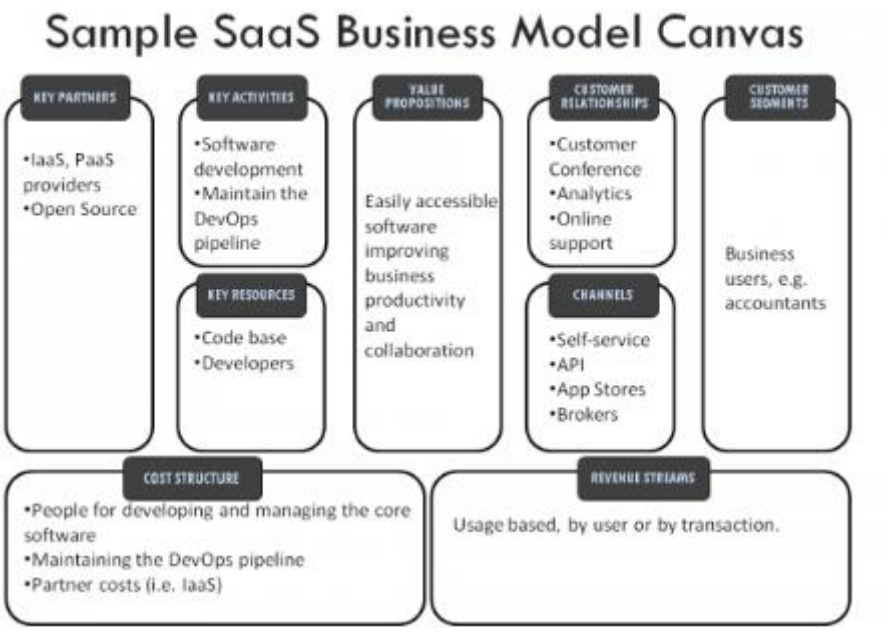
- Οι περισσότεροι προμηθευτές μέσω σύννεφων υλοποιούν μια μεγάλη επένδυση σε υποδομή ασφαλείας και δημιουργίας αντιγράφων ασφαλείας. Εάν ένας διακομιστής αποτύχει, αυτόματα λαμβάνεται σε άλλο διακομιστή.
- Το λογισμικό βρίσκεται στην πιο πρόσφατη έκδοση.
- Η γρήγορη υλοποίηση ως λύση είναι ταχύτερη και ευκολότερη στην εφαρμογή
- Ο προμηθευτής σύννεφων παρέχει τεχνικούς πόρους
- Οι περισσότερες επιχειρήσεις χρησιμοποιούν μόνο το 5-10% των πόρων διακομιστή τους, με μια λύση σε ένα σύννεφο για την οποία πληρώνουν σε μηνιαία βάση μόνο το ποσό πόρων διακομιστή που χρειάζονται.

5.4.4. Μειονεκτήματα Παροχής Λογισμικού ως Υπηρεσία

- Δεν παρέχεται από όλους τους προμηθευτές η απαιτούμενη προσαρμογή
- Απόλυτη εξάρτηση από το Διαδίκτυο και τον πάροχο σας. Αν το Διαδίκτυο τεθεί εκτός λειτουργίας, ολόκληρη η επιχείρησή είναι εκτός σύνδεσης και δεν έχει πρόσβαση στον διακομιστή.
- Οι επιχειρήσεις που βρίσκονται σε μια περιοχή με περιορισμένη πρόσβαση στο Διαδίκτυο, δεν μπορούν να χρησιμοποιήσουν ένα μοντέλο βασισμένο στο Υπολογιστικό Νέφος
- Η ασφάλεια των δεδομένων ποικίλλει ανάλογα με τον προμηθευτή και είναι εκτός του ελέγχου της εταιρίας.
- Εάν δεν διατηρηθούν οι τρέχουσες πληρωμές προμηθευτών, ενδέχεται να χαθεί η πρόσβαση στο σύστημα.
- Δυσκολία μεταφοράς δεδομένων σε ένα νέο σύστημα.
- Ανάλογα με το μέγεθος της επιχείρησής, εκεί μπορεί να μην είναι μεγάλο το πλεονέκτημα κόστους.

5.4.5. Επιχειρηματικό Μοντέλο Λογισμικού ως Υπηρεσία

Ένα επιχειρηματικό μοντέλο απεικονίζει τον τρόπο με τον οποίο μια επιχείρηση, ή γενικότερα ένας οργανισμός, κερδίζει έσοδα δημιουργώντας και προσφέροντας αξία σε κάποιο προϊόν. Εργαλείο του επιχειρηματικού σχεδίου είναι ο καμβάς του επιχειρηματικού μοντέλου (Business model canvas). Όπως φαίνεται στο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, το επιχειρηματικό μοντέλο μπορεί να παρουσιαστεί μέσω εννέα δομικών στοιχείων που ικανοποιούν τους τέσσερις κύριους τομείς δραστηριότητας (υποδομή, αξία, πελάτες και οικονομικά στοιχεία).



Σχήμα 5.3: Στοιχεία Καμβά Επιχειρηματικού Σχεδίου του Λογισμικού ως Υπηρεσία

- **Υποδομή**

Βασικοί συνεργάτες (Key Partners) είναι το δίκτυο των συνεργατών και των προμηθευτών. Βελτιστοποιούν το επιχειρηματικό μοντέλο, μειώνουν τον κίνδυνο και συμμετέχουν στην απόκτηση συγκεκριμένων πόρων και δραστηριοτήτων. (Osterwalder & Pigneur, 2010)

Βασικές δραστηριότητες (Key Activities) είναι οι ενέργειες που πρέπει να λάβει μια επιχείρηση για να λειτουργήσει με επιτυχία. Εξαρτώνται από τον τύπο της επιχείρησης και τον τρόπο εσωτερική οργάνωσης της. (Osterwalder & Pigneur, 2010)

Βασικοί Πόροι (Key Resources) είναι τα υλικά, τα άυλα περιουσιακά στοιχεία, οι ανθρώπινοι πόροι και οι χρηματικοί πόροι, που επιτρέπουν σε μια επιχείρηση να υλοποιήσει το επιχειρηματικό της μοντέλο. (Osterwalder & Pigneur, 2010)

- **Αξία**

Πρόταση Αξίας (Value Proposition) είναι ο λόγος για τον οποίο ο πελάτης θα επιλέξει να αγοράσει το προϊόν/υπηρεσία από την συγκεκριμένη επιχείρηση. Ικανοποιεί ένα πρόβλημα ή μια ανάγκη της αγοράς στόχου,, προσφέροντας ένα πακέτο ωφελειών είτε ποσοτικών είτε ποιοτικών. Η ικανοποίηση μιας ανάγκης, η απόδοση, η παραμετροποίηση, η ασφάλεια, η αισθητική, το Brand name/Status, η βελτιωμένη τιμή, η μείωση κόστους χρήσης, η μείωση του ρίσκου, η προσβασιμότητα και η χρηστικότητα είναι τα στοιχεία που διαμορφώνουν μια πρόταση αξίας. (Osterwalder & Pigneur, 2010)

- **Πελάτες**

Αγορές-στόχοι (Customer Segments) είναι ίσως το σημαντικότερο στοιχείο, καθώς αποτελεί την καρδιά ενός επιχειρηματικού πλάνου. Χωρίς πελάτες δεν υπάρχουν και έσοδα. Οπότε σε αυτό το σημείο κάθε οργανισμός καθορίζει ποιοι είναι αυτοί οι πελάτες που σκοπεύει να εξυπηρετήσει και ποιες είναι οι ανάγκες που είναι σε θέση να εξυπηρετήσει. Ενδεικτικά τύποι πελατών στους οποίους θα μπορούσε να απευθυνθεί μια επιχείρηση είναι η μαζική αγορά (mass market), εξειδικευμένη αγορά (niche market), τμηματοποιημένη αγορά (Segmented), διαφοροποιημένη αγορά (Diversified), πλατφόρμες πολλών μερών (multi-sided platforms or multi-sided markets). (Osterwalder & Pigneur, 2010)

Σχέσεις με πελάτες (Customer Relations). Κάθε επιχείρηση πρέπει να αποσαφηνίζει το είδος της σχέσης που θέλει να υπάρχει ανάμεσα σε εκείνη και στην αγορά στόχο. Το εύρος των σχέσεων κυμαίνεται από σχέση προσωπικού επιπέδου έως τυπική αυτοματοποιημένη σχέση, και εξαρτώνται από τα κίνητρα (απόκτηση νέων πελατών, διατήρηση πελατών, αύξηση των πωλήσεων) που έχει η επιχείρηση. (Osterwalder & Pigneur, 2010)

Δίκτυα (Channels) ή αλλιώς κανάλια επικοινωνίας είναι ο τρόπος με τον οποίο ο οργανισμός προσεγγίζει τους πελάτες της για να παραδώσει την πρόταση αξίας της. Στα δίκτυα συμπεριλαμβάνονται η επικοινωνία η διανομή και οι πωλήσεις, αποτελούν σημεία επαφής με τους πελάτες και διαδραματίζουν σημαντικό ρόλο εξυπηρετώντας διάφορες λειτουργίες. Διαδοχικά οι λειτουργίες είναι η ενημέρωση ,η αξιολόγηση της πρότασης αξίας, η αγορά, η παράδοση και η εξυπηρέτηση μετά την πώληση. (Osterwalder & Pigneur, 2010)

- **Οικονομικά στοιχεία**

Δομή κόστους (Cost Structure) περιγράφει όλες τις δαπάνες (κόστη για απόκτηση παγίων και κόστος λειτουργίας της επιχείρησης) που θα υποστεί η επιχείρηση. Τέτοιες δαπάνες μπορούν να υπολογιστούν σχετικά εύκολα μετά τον ορισμό των βασικών πόρων, των βασικών δραστηριοτήτων και των βασικών εταιρικών σχέσεων. (Osterwalder & Pigneur, 2010)

Ροές εσόδων (Revenue Streams) αφορούν τα έσοδα που παράγονται από την προσφορά της πρότασης αξίας στην αγορά στόχος. Όπως αναφέρθηκε παραπάνω αν η αγορά στόχος είναι η καρδιά του επιχειρηματικού πλάνου τότε η ροές εσόδων είναι οι αρτηρίες του. Διάφορα είδη ροών εσόδων που μπορεί να αναπτύξει μία επιχείρηση είναι η πώληση παγίων, το τέλος χρήσης, οι συνδρομές, ενοικίαση ή μίσθωση, η άδεια χρήσης, η μεσιτική αμοιβή και η διαφήμιση. (Osterwalder & Pigneur, 2010)

Στην παρούσα εργασία αποσαφηνίζεται η χρησιμότητα του SaaS σε ένα ERP σύστημα μιας επιχείρησης.

5.4.6. Πλεονεκτήματα SaaS στο ERP Σύστημα

Τα τελευταία χρόνια η ζήτηση για λογισμικό ως υπηρεσία (SaaS) με σκοπό την διαχείριση πόρων αυξάνεται. Οι οργανισμοί έχουν παρατηρήσει πλεονεκτήματα στη χρήση του λογισμικού σε ένα υπολογιστικό νέφος για την επίλυση διαφόρων λειτουργικών διαδικασιών. Για τις μικρομεσαίες επιχειρήσεις (Small and Medium Enterprises – SME) θεωρείται ως η καλύτερη ευκαιρία, αφού δεν χρειάζεται η επένδυση χρημάτων στο λογισμικό και στην διαχείριση /συντήρηση του, ενώ οι μεγάλες επιχειρήσεις (Large Enterprises – LE) το θεωρούν ως συμπλήρωμα του ήδη υπάρχοντος λογισμικού. (Churakova & Mikhramova, June 2010). Και στις δύο περιπτώσεις μειώνεται η υπολογιστική ισχύς που απαιτείται να έχει μια εταιρία ώστε να φιλοξενήσει το λογισμικό που χρειάζεται.

Οφέλη για τις μικρές και μεσαίες επιχειρήσεις (SME)

Το υπολογιστικό νέφος επιτρέπει στις μικρομεσαίες επιχειρήσεις να αντιμετωπίσουν τις προκλήσεις του προϋπολογισμού της κλιμάκωσης των βασικών δραστηριοτήτων τους, έχοντας πρόσβαση στο υλικό και στο λογισμικό, που προσφέρουν ικανοποιητική ποιότητα εξυπηρέτησης και χαμηλό κόστος ιδιοκτησίας. Το SaaS είναι το μοντέλο για το οποίο οι περισσότερες μικρομεσαίες επιχειρήσεις χρησιμοποιούν το υπολογιστικό νέφος, περιλαμβάνοντας την πρόσβαση στο λογισμικό μέσω του νέφους, έναντι της συνήθης ανάπτυξης του λογισμικού στις υπολογιστικές μονάδες της εταιρίας. Παρακάτω παρατίθενται έξι πλεονεκτήματα για της χρήσης του SaaS από τις μικρομεσαίες επιχειρήσεις

Χαμηλό κόστος

Από οικονομικής πλευράς, η υιοθέτηση ενός SaaS μοντέλου αποτρέπει μια αρχική επένδυση σε μια υποδομή λογισμικού, μειώνει την ανάγκη για τεχνικές και πόρους πληροφορικής. Αυτός ο νέος τρόπος χρήσης της πληροφορικής, βοηθάει στην καλύτερη διαχείριση των δαπανών και αποτελεί πραγματικά οικονομικό όφελος για τις μικρομεσαίες επιχειρήσεις των οποίων οι οικονομικοί πόροι είναι αρκετά περιορισμένοι. (Birger, 2018), (Johanssona & Ruivob, 2013)

Γρήγορη, εύκολη διαθεσιμότητα και κινητικότητα

Δυο στοιχεία χρειάζονται μόνο για την χρήση του μοντέλου SaaS, ένα πρόγραμμα περιήγησης και σύνδεση στο Internet. Δεν απαιτείται η εγκατάσταση λογισμικού και των ενημερώσεων στους υπολογιστές της εταιρίας. Επίσης, οι εφαρμογές SaaS επιτρέπουν την

πρόσβαση από οπουδήποτε και δεν εξαρτώνται από την ποιότητα του υλικού ή τις συσκευές αποθήκευσης που είναι εγκατεστημένες σε κάθε συσκευή, με αποτέλεσμα να αυξάνεται η παραγωγικότητα (Birger, 2018), (Johanssona & Ruivob, 2013)

Ευχέρεια διαχείρισης

Η επιχείρηση δεν μεριμνά για αναβαθμίσεις και επιδιορθώσεις του λογισμικό, καθώς ο πάροχος βελτιώνει καθημερινά το λογισμικό και οι αναβαθμίσεις είναι διαθέσιμες σε όλους τους χρήστες ανά πάσα στιγμή. Επομένως, ο χρόνος κατά τον οποίο οι υπολογιστές διακόπτουν την λειτουργία τους μηδενίζεται. (Birger, 2018), (Johanssona & Ruivob, 2013)

Πληρωμή ανά χρήση

Το χρηματικό ποσό που καταβάλλει η επιχείρηση συνδέεται άμεσα με την χρήση της, δηλαδή τον αριθμό των χρηστών, τον αποθηκευτικό χώρο που χρησιμοποιεί και τις διάφορες παραμετροποιήσεις που έχουν ζητηθεί στον πάροχο. Αυτό το χαρακτηριστικό είναι πολύ σημαντικό αν αναλογιστεί κανείς επιχειρήσεις που έχουν εποχική δραστηριότητα ή μικρομεσαίες επιχειρήσεις που χρησιμοποιούν το ίδιο λογισμικό με μεγάλες επιχειρήσεις. (Birger, 2018), (Johanssona & Ruivob, 2013)

Πολύ-επίπεδο λογισμικό

Οι πάροχοι του Λογισμικού ως Υπηρεσία συνηθίζουν να δημιουργούν διάφορα επίπεδα στο λογισμικό ώστε να δίνεται η δυνατότητα στους χρήστες να επιλέγουν τις υπηρεσίες που θέλουν να χρησιμοποιήσουν. Οπότε, Εάν μια δραστηριότητα εξελίσσεται, τότε μπορούν να επιλεχθούν πιο προηγμένες υπηρεσίες του SaaS. Κάτι τέτοιο εξυπηρετεί άμεσα μια νεοσύστατη επιχείρηση, η οποία στοχεύει στην ανάπτυξη αλλά δεν θέλει από την αρχή να επενδύσει χρήματα σε προηγμένες λειτουργίες του SaaS. (Birger, 2018), (Johanssona & Ruivob, 2013)

Φιλικό προς το περιβάλλον

Οι διακομιστές του ηλεκτρονικού υπολογιστή καταναλώνουν μεγάλη ποσότητα ενέργειας και απαιτούν δροσερό περιβάλλον ώστε να δουλεύουν σωστά. Επομένως, η χρήση ενός μοντέλου SaaS μειώνει την συνολική κατανάλωση ηλεκτρικής ενέργειας από τις επιχειρήσεις, αντιμετωπίζοντας διάφορες προκλήσεις αειφόρου ανάπτυξης. (Birger, 2018)

Οφέλη για τις μεγάλες επιχειρήσεις (LE)

Είναι λογικό μια μεγάλη επιχείρηση να έχει διαφορετικές ανάγκες από μια μικρή επιχείρηση, επομένως οι πάροχοι του SaaS θα έπρεπε να επικεντρωθούν και στις απαιτήσεις αυτών.

Παρατηρούμε όμως ότι η ανάγκη των μεγάλων επιχειρήσεων δεν έχει σχέση με βασικές λειτουργίες της αλλά με δευτερεύουσες. Πιο συγκεκριμένα, η ανάθεση υποστηρικτικών εφαρμογών σε παρόχους υπηρεσιών SaaS, όπως είναι καθημερινές λειτουργίες (εφαρμογές ηλεκτρονικού ταχυδρομείου) που δεν αποτελούν ανταγωνιστικό πλεονέκτημα, δίνει την δυνατότητα στις επιχειρήσεις να επικεντρώνονται σε υψηλής προτεραιότητας λειτουργίες του επιχειρηματικού μοντέλου τους. (Churakova & Mikhramova, June 2010)

Σχέσεις με πελάτες

Με την εμφάνιση του SaaS, το μοντέλο push μετασχηματίστηκε σε μοντέλο pull. ενώ παλιότερα, ο πωλητής έπρεπε να πείσει τον αγοραστή-πελάτη να αγοράσει το προϊόν του, πλέον ο πωλητής μετατρέπεται σε πάροχο υπηρεσίας που έλκει τους πελάτες του με την λύση ενός προβλήματός τους. Ο πάροχος πραγματοποιεί μια σημαντική επένδυση στη δημιουργία, στη διατήρηση και στη βελτίωση αυτής της υπηρεσίας, ώστε ο πελάτης που έχει την ελευθερία να αλλάξει πάροχο αν δεν μείνει ικανοποιημένος, να μην διακόψει την πελατειακή τους σχέση. (Churakova & Mikhramova, June 2010).

Επομένως, η σχέση με τους πελάτες εξαρτάται και από την πορεία της διαδικασίας των πωλήσεων και από την ωριμότητά της. Οι πάροχοι των υπηρεσιών SaaS πρέπει να εξασφαλίζουν μακροπρόθεσμο βιώσιμο ρεύμα μικρότερων συνδρομών, άρα η πελατειακή βάση πολυάριθμη. Σκοπός των παρόχων είναι η συνεχής απόκτηση νέων πελατών προκειμένου να παραμείνουν στην επιχείρηση ακόμη και με το μικρό ποσοστό χρέωσης. Για να διατηρήσουν την πελατεία τους, οι πάροχοι είναι αναγκαίο να εστιάζουν στο γεγονός ότι αυτό που προσφέρουν δεν είναι ένα λογισμικό αλλά μια Υπηρεσία. Τέλος, κατά τη διάρκεια της υιοθέτησης του SaaS από μια επιχείρηση, η καθημερινή προοπτική επεξεργασίας είναι το στοιχείο που συνοδεύεται από βραχυπρόθεσμη και μακροπρόθεσμη αποδοτικότητα. Ο πάροχος πρέπει οποιαδήποτε στιγμή να εξασφαλίζει την διαφάνεια στην αξιολόγηση της ποιότητας της υπηρεσίας από τους χρήστες, χτίζοντας με αυτόν τον τρόπο μια σχέση εμπιστοσύνης. (Churakova & Mikhramova, June 2010)

Δίκτυα επικοινωνίας

Στη διεθνή βιβλιογραφία κάποιοι θεωρούν ότι το διαδίκτυο από μόνο του μπορεί να αποτελέσει δίαυλο επικοινωνίας μεταξύ των πελατών και της πρότασης αξίας, δηλαδή ότι η υπηρεσία μπορεί και πουλά τον εαυτό της. Από την άλλη πλευρά υπάρχουν υποστηρικτές που προωθούν το μοντέλο στο οποίο ο κάθε συνεργάτης κατέχει σημαντικό ρόλο στην πρόταση αξίας και στον τύπο του πελάτη που εξυπηρετούν. Σύμφωνα με το βιβλίο *Channel*

Excellence του Axel Schultze για το SaaS υπάρχουν ειδικοί «καταλύτες» που δεν παράγουν ούτε προμηθεύουν την επιχείρηση με κάποιον τρόπο αλλά βοηθούν τους πελάτες να επιλέξουν το κατάλληλο μοντέλο SaaS ή/και να το προσαρμόσουν στο επιχειρησιακό τους μοντέλο. (Schultze, December 3, 2007)

Αγορά Στόχος

Αν και στα προηγούμενα κεφάλαια αναλύθηκαν τα οφέλη για τις μικρομεσαίες και μεγάλες επιχειρήσεις, πρέπει να αναφερθεί πως αυτές δεν καθορίζουν την αγορά στόχος αλλά αποτελούν ένα μεγάλο υποσύνολο της. Συγκεκριμένα, υπάρχουν τέσσερα βασικά χαρακτηριστικά της αγοράς στόχου

- Σύνδεση στο διαδίκτυο και χρήση περιηγητή ιστού
- Το τμήμα πληροφορικής να μην είναι δυνατό όπως οι μικρομεσαίες ή να επικεντρώνεται σε άλλες βασικές λειτουργίες (μεγάλες επιχειρήσεις)
- Να είναι οικονομικά αδύναμες ως προς την αγορά του λογισμικού
- Η βάση των χρηστών να είναι σχετικά μικρή

5.5. Πρόταση για το σύστημα

Απο την παραπάνω ανάλυση φαίνεται ότι η τοπική εγκατάσταση λογισμικού στους διακομιστές των εταιριών δεν αποτελεί την κατάλληλη λύση για ένα τέτοιο σύστημα, καθώς αυτό θα σήμαινε υψηλό κόστος τόσο για την εγκατάσταση του (αγορά αντίστοιχων αδειών) όσο και για την συντήρηση του. Επίσης, η ενσωμάτωση με εφαρμογές τρίτων, όπως είναι στην περίπτωση του συστήματος οι εφαρμογές που παρέχουν τα δεδομένα κίνησης, είναι αρκετά δαπανηρή και πολύπλοκη. Ενώ τελικά ο χρόνος ανάπτυξης του λογισμικού θα ήθελε περισσότερο χρόνο και μεγαλύτερη υποδομή του πληροφοριακού συστήματος, που εν τέλει σηματοδοτεί μεγαλύτερο κόστος. Απο την άλλη πλευρά, η δεύτερη λύση του «Φιλοξενούμενου» Λογισμικού, φαίνεται ότι μειονεκτεί και αυτή στο κόστος διότι γίνεται ενιαία μίσθωση, που περιορίζει τις ευκαιρίες μείωσης του κόστους, χρεώνεται επίσης η εξυπηρέτηση και η υποστήριξη καθώς. Όμως, πέρα απο το κόστος, η προσαρμογή στις ανάγκες της εταιρίας είναι περιορισμένη, και εμφανίζεται και σε αυτή την περίπτωση η δαπανηρή και δύσκολη ενσωμάτωση με εφαρμογές τρίτων.

Στην παρούσα εργασία η μέθοδος διανομής του λογισμικού επιλέγετα να γίνεται σε ένα περιβάλλον υπολογιστικού νέφους δηλαδή επιλέγεται το μοντέλο παροχής λογισμικού ως υπηρεσία, καθώς φαίνεται να πλεονεκτεί σε σχέση με τα άλλα είδη.

6. Μοντέλα Ανάπτυξης Λογισμικού Στο Cloud

6.1. Εισαγωγή

Το κεφάλαιο 6 αναφέρεται στο υπολογιστικό νέφος (Cloud), σε έναν από τους πιο διαδεδομένους τεχνολογικούς όρους τη δεκαετία που διανύουμε και συνάμα, μία τεχνολογία ικανή να αλλάξει τον τρόπο με τον οποίο χρησιμοποιείται σε καθημερινή βάση η Τεχνολογία. Είναι το κεφάλαιο στο οποίο γίνεται εισαγωγή του όρου Λογισμικό ως Υπηρεσία, που προτάθηκε στο κεφάλαιο 5 ως μοντέλο παροχής λογισμικού στις εταιρίες. Συγκεκριμένα, έπειτα από μια ιστορική αναδρομή των τεχνολογιών και της εξέλιξης την πληροφορικής, που κατέληξαν να δημιουργήσουν το υπολογιστικό νέφος, παρουσιάζονται τα πλεονεκτήματα και τα τρία μοντέλα υπηρεσιών αυτού. Τέλος, γίνεται αναφορά και αξιολόγηση δύο βασικών μοντέλων, του Microsoft Azure και του Oracle Cloud, για τα οποία προτείνεται συγκεκριμένος τρόπος χρήσης τους για το σύστημα που πραγματεύεται η παρούσα εργασία.

6.2. Ιστορική Αναδρομή

Η σύγχρονη εποχή χαρακτηρίζεται από ραγδαία ανάπτυξη της τεχνολογίας και της πληροφορικής. Ειδικότερα, η βιομηχανία της πληροφορικής (IT industry), η οποία έχει ως ιδιαίτερα γνωρίσματα την υψηλή ταχύτητα εξέλιξης και την καινοτομία, έχει αλλάξει ριζικά την καθημερινότητα του ανθρώπου, τόσο σε προσωπικό επίπεδο όσο και σε επαγγελματικό, τα τελευταία εκτατό χρόνια. Χαρακτηριστικά παραδείγματα αυτής της επιρροής αποτελούν η χρήση του ηλεκτρονικού υπολογιστή, η ενσωμάτωση του Διαδικτύου και το Υπολογιστικό Νέφος (Cloud Computing) που με τη σειρά του δημιούργησε το τρίπτυχο SPI δηλαδή, το Λογισμικό ως υπηρεσία (SaaS), την Πλατφόρμα ως Υπηρεσία (PaaS) και την Υποδομή ως Υπηρεσία (IaaS). (Churakova & Mikhramova, June 2010), (Tsai, Sun, & Balasooriya, 2010)

Συγκεκριμένα, με δεδομένο το γεγονός ότι η υπολογιστική ισχύς είναι μια μη ντετερμινιστική έννοια, αναφέρονται οι απαιτήσεις ενός λογισμικού που κατά προσέγγιση είναι η ποσοτική έννοια της υπολογιστικής ισχύος. Όπως έγινε φανερό και στο κεφάλαιο 6.2 για την καλύτερη και γρηγορότερη επίλυση προβλημάτων λογισμικού οι επιχειρήσεις είτε αναγκάζονταν να διαμοιράσουν το πρόβλημα σε πολλούς υπολογιστές είτε να επινοικιάσουν το λογισμικό από άλλες εταιρίες, αυτό σημαίνει ότι τα λογισμικά με τα οποία μια εταιρία διαχειρίζεται τις λειτουργίες και τις διαδικασίες της είναι ένα αρκετά «βαρύ πρόγραμμα», όπως θα λεγόταν στην γλώσσα της πληροφορικής. Εμβαθύνοντας περισσότερο στην φράση βαρύ πρόγραμμα, παρατηρείται ότι ένα λογισμικό μπορεί να λειτουργήσει σε συγκεκριμένο λειτουργικό σύστημα, απαιτεί συγκεκριμένη ταχύτητα επεξεργαστή όπως και αναγκαίο χώρο στη μνήμη

RAM και αποθηκευτικό χώρο στον σκληρό δίσκο, ενώ κάποιες φορές έχουν σημασία η κάρτα γραφικών και οι θύρες εσόδου και εξόδου. Όλα τα παραπάνω δημιουργούν ανάγκες οι οποίες λόγω κόστους, είτε είναι αδύνατον να υλοποιηθούν από τις μικρές ή νέες επιχειρήσεις είτε επιβαρύνουν κατά πολύ τις μεγαλύτερες επιχειρήσεις.

Το υπολογιστικό νέφος δημιουργήθηκε την δεκαετία του 1950 κατά την οποία χρειάστηκε η αύξηση της αποδοτικότητας των κεντρικών υπολογιστών διαφόρων επιχειρήσεων και ιδρυμάτων. Πιο συγκεκριμένα, οι χρήστες του υπολογιστικού νέφους δύναται να έχουν ταυτόχρονη πρόσβαση σε δυναμικές και επεκτάσιμες βάσεις δεδομένων με την χρήση του διαδικτύου από πολλαπλά τερματικά, χωρίς να γνωστοποιείται η θέση των φυσικών πόρων και των συσκευών (Almishal & Youssef, May 2014), (Kulkarni & Gambhir, Jan-Feb 2012). Η χρήση των υπολογιστικών νεφών δίνει επίσης την δυνατότητα στους χρήστες να αναπτύξουν, να διαχειρίζονται και να παρέχουν εφαρμογές μέσω αυτού.

Σύμφωνα με το NIST (National Institute of Standards and Technology, NIST) η τεχνολογία του υπολογιστικού νέφους ορίζεται ως εξής: *"Το Υπολογιστικό νέφος είναι ένα μοντέλο που επιτρέπει την εύκολη πρόσβαση σε δίκτυο σε μια κοινόχρηστη ομάδα από διαμορφωμένους υπολογιστικούς πόρους (π.χ. δίκτυα, διακομιστές, εφαρμογές αποθήκευσης και υπηρεσίες) που μπορεί να παρέχεται γρήγορα και να απελευθερώνεται με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης του παρόχου υπηρεσιών"*

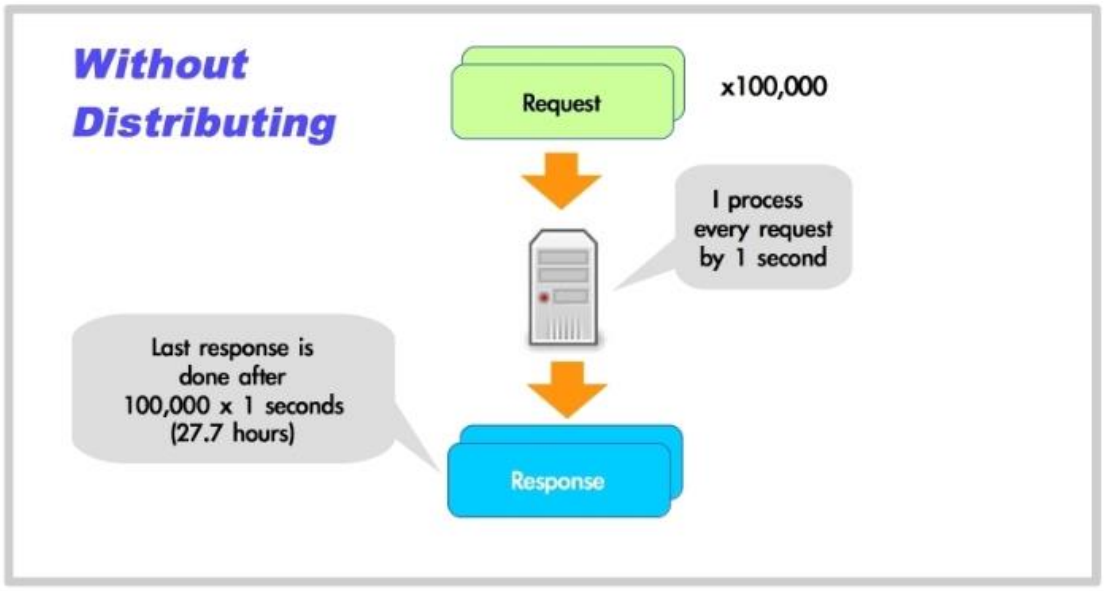
Η αρχή της επαναχρησιμοποίησης των υπηρεσιών (Service reusability principle) είναι η τεχνική βάση δημιουργίας του υπολογιστικού νέφους. Οι επαναχρησιμοποιούμενες υπηρεσίες σχεδιάζονται πάνω στην αρχιτεκτονική της προσανατολισμένης υπηρεσίας (SOA), οι οποίες μπορούν να επαναχρησιμοποιηθούν ανεξάρτητα από τις επιχειρηματικές διαδικασίες ή τεχνολογίες της επιχείρησης, με σκοπό την αύξηση της αποδοτικότητας. Συγκεκριμένα, από την πλευρά της επιχείρησης, η επαναχρησιμοποίηση έχει άμεσο αντίκτυπο στον δείκτη «επιστροφή στην επένδυση- ROI» (Δείκτης αναλογίας ποσού που καταναώθηκε προς τα μεικτά κέρδη της επένδυσης), αφού μειώνει τα κόστη συντήρησης, σχεδιασμού, ανάπτυξης και διαφόρων δοκιμών. Χαρακτηριστικό παράδειγμα αποτελεί ένας οργανισμός ασφάλισης στο Ηνωμένο Βασίλειο, όπου εξοικονομήθηκαν 2.000.000 λίρες από την επαναχρησιμοποίηση των υπηρεσιών κατά 51%. (Feuerlicht & Lozina, 2007)

Ωστόσο, πέρα από το υπολογιστικό νέφος υπήρχαν και άλλα ήδη μοντέλων όπως είναι το υπολογιστικό πλέγμα (grid computing), κατανεμημένο υπολογιστικό σύστημα (distributed computing), υπολογιστική χρησιμότητα (utility computing) και η αυτόνομη υπολογιστική

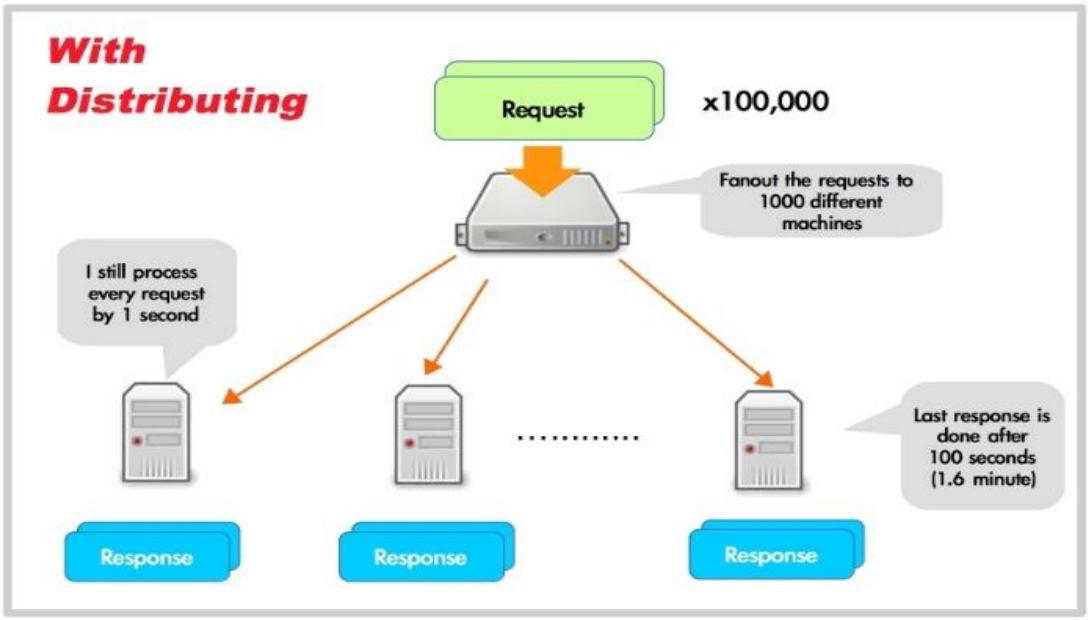
(autonomic computing). Παρακάτω αναλύονται αυτά τα μοντέλα ώστε να φανερωθεί η βασική διαφορά ανάμεσα σε αυτά και στο υπολογιστικό νέφος.

6.3. Κατανεμημένο Υπολογιστικό Σύστημα (Distributed Computing)

Το **κατανεμημένο υπολογιστικό σύστημα (distributed computing)** αποτελείται από επιμέρους ανεξάρτητους υπολογιστές συνδεδεμένους με τέτοιον τρόπο που ο χρήστης θεωρεί ότι χρησιμοποιεί μόνο έναν υπολογιστή. Στο βιβλίο DISTRIBUTED SYSTEMS – Principles and Paradigms ορίζεται ως *"Ένα κατανεμημένο σύστημα είναι μια συλλογή από αυτόνομα υπολογιστικά στοιχεία που εμφανίζεται στους χρήστες του ως ένα ενιαίο συνεκτικό σύστημα"* (Tanenbaum & Steen, February 26, 2016)



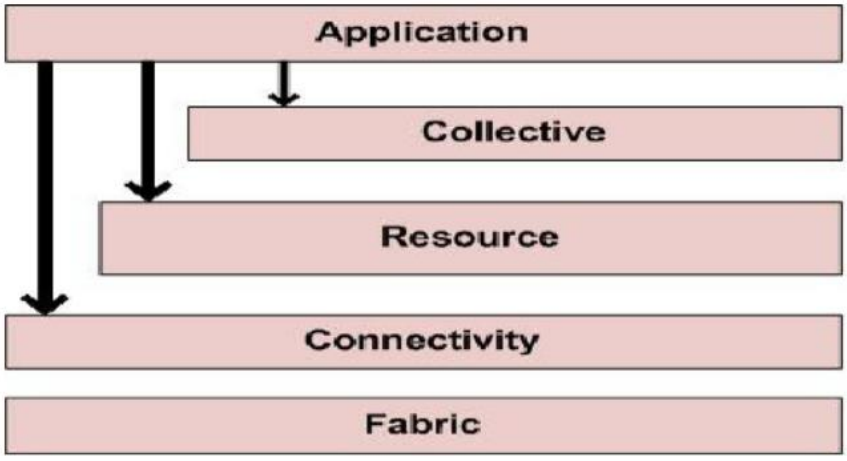
Σχήμα 6.1: Μη Κατανεμημένο Υπολογιστικό Σύστημα



Σχήμα 6.2: Κατανεμημένο Υπολογιστικό Σύστημα

6.4. Grid Computing

Το Grid computing είναι ένα υπολογιστικό μοντέλο που βασίζεται σε ένα δίκτυο υπολογιστών με σκοπό την επίλυση μεγάλων υπολογιστικών προβλημάτων. Προσφέρει υψηλά επίπεδα υπολογιστικής, αποθήκευσης και χωρητικότητας δικτύου. (Gerami, 2010). Η βασική διαφορά από τα κατανεμημένα υπολογιστικά συστήματα είναι το γεγονός ότι οι υπολογιστικοί πόροι μπορούν να βρίσκονται σε μεγάλη γεωγραφικά απόσταση μεταξύ τους. Έτσι, όταν ένας υπολογιστής δεν έχει περαιτέρω υπολογιστική ισχύ (υπερφόρτωση) ώστε να τρέξει μια συγκεκριμένη εφαρμογή, μπορεί μέσω του grid computing να την τρέξει σε κάποιον άλλον υπολογιστή στον οποίο υπάρχουν ανεκμετάλλευτοι υπολογιστικοί πόροι.



Σχήμα 6.3: Επίπεδα Grid Computing

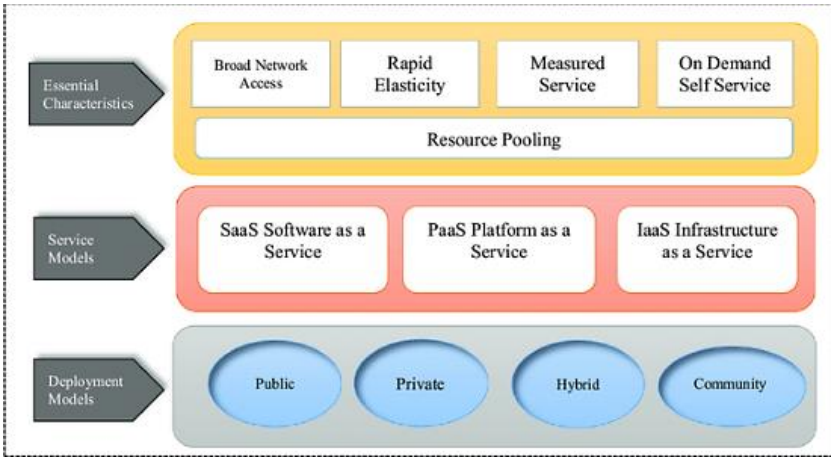
Ένα από τα πιο διάσημα παραδείγματα του υπολογιστικού πλέγματος είναι η αναζήτηση εξωτερικής ευφυίας που διευθύνεται από το SETI, ένα πρόγραμμα δηλαδή που σκοπεύει στην απόδειξη της ύπαρξης εξωγήινης νοημοσύνης συλλέγοντας σήματα ή διάφορες άλλες μορφές επικοινωνίας στο διάστημα. Αυτή η εφαρμογή όπως είναι φανερό χρήζει τεράστιας υπολογιστικούς πόρους για την σάρωση του ουρανού από πολλούς υπολογιστές. (Dar & Riyaz, 2017) (Patel, Seyfi, Tew, & Jaradat, 2011)

6.5. Υπολογιστική Χρησιμότητα (Utility Computer)

Υπολογιστική χρησιμότητα (Utility Computer) είναι μια αρχή στην οποία μια εταιρία μισθώνει μια άλλη εταιρία για υπηρεσίες πληροφορικής, δηλαδή μίσθωση λογισμικού, χώρου αποθήκευσης δεδομένων κλπ.. Ουσιαστικά σε αυτό το μοντέλο οι πόροι ενοικιάζονται, μειώνοντας με αυτόν τον τρόπο όπως και στο μοντέλο του υπολογιστικού πλέγματος, το κόστος των υπολογιστικών πόρων. Αυτό το μοντέλο υπηρεσιών είχαν υιοθετήσει μεγάλες εταιρίες όπως η IMB, σε τράπεζες και σε μεγάλους οργανισμούς. Συγκεκριμένα, η IBM είναι γνωστή για την παροχή της υπηρεσίας χρονομεριστικής μίσθωσης (Time sharing), δηλαδή ένας υπολογιστής για πολλούς χρήστες. (Tiwari, Shrivastava, Pandey, & Tripathi, 2011) (Patel, Seyfi, Tew, & Jaradat, 2011)

6.6. Υπολογιστικό Νέφος

Είναι φανερό ότι το υπολογιστικό νέφος δεν αποτελεί μια επανάσταση στον τομέα της πληροφορικής, είναι μια εξέλιξη, που βασίζεται στον συνδυασμό των τεχνολογικών μοντέλων των κατανεμημένων υπολογιστικών συστημάτων (distributed computing), του υπολογιστικού πλέγματος (Grid computing) και της υπολογιστικής χρησιμότητας (Utility computing). Το εννοιολογικό πλαίσιο του υπολογιστικού νέφους είναι αυτό μέσω του οποίου αναγνωρίζονται με ευκολία τα πέντε θεμελιώδη χαρακτηριστικά που ορίζουν την αρχιτεκτονική του και εξηγούν τις διαφοροποιήσεις που υφίσταται σε σχέση με τους προγόνους του. (Patel, Seyfi, Tew, & Jaradat, 2011)



Σχήμα 6.4: Conceptual View of Cloud Computing

Αναλυτικά τα στοιχεία του Σχήμα 6.4

- **Ευρεία πρόσβαση στο δίκτυο (Broad network Access) :**

Η ευρεία πρόσβαση στο δίκτυο αποτελεί μια σημαντική ιδιότητα του υπολογιστικού νέφους καθώς η διαθεσιμότητα των υπηρεσιών του γίνεται μέσω οποιασδήποτε συσκευής και πλατφόρμας, με ή χωρίς διαδίκτυο ανάλογα τον τύπο του υπολογιστικού νέφους. (Mell & Grance, 2011)

- **Ταχεία ελαστικότητα (Rapid Elasticity)**

Οι πόροι στο υπολογιστικό νέφος μπορούν να δεσμεύονται και να αποδεσμεύονται γρήγορα. Όταν δεσμεύονται από κάποιον χαρακτηρίζονται ως μη διαθέσιμοι για τους υπόλοιπους. Επίσης, αυτοί οι πόροι διατίθενται σε οποιαδήποτε ποσότητα και οποιαδήποτε στιγμή στον τελικό χρήστη. Παράδειγμα τέτοιων πόρων είναι ο χρόνος χρήσης του διακομιστή ή το μέγεθος του αποθηκευτικού χώρου. (Mell & Grance, 2011)

- **Αυτό-εξυπηρέτηση κατά απαίτηση (On Demand Self Service)**

Ο τελικός χρήστης μπορεί να χρησιμοποιεί τους πόρους που θέλει χωρίς την διαμεσολάβηση κάποιου τρίτου προσώπου ή την επικοινωνία με τον πάροχο της υπηρεσίας. (Mell & Grance, 2011)

- **Μετρούμενη υπηρεσία (Measured Service)**

Σε ένα υπολογιστικό νέφος, η χρήση των πόρων ελέγχεται και βελτιστοποιείται αυτόματα. Η παρακολούθηση από το νέφος, παρουσιάζεται στον τελικό χρήστη με την μορφή αναφορών, έτσι γίνεται φανερός ο τρόπος διαχείρισης τόσο στον χρήστη όσο και στον πάροχο. (Mell & Grance, 2011)

- **Διάθεση πόρων (Resource Pooling)**

Οι υπολογιστικοί πόροι του παρόχου χρησιμοποιούνται για να εξυπηρετήσουν ποικίλους και διάφορους καταναλωτές με τη χρήση του μοντέλου πολλαπλών μισθωτών (multitenant), με τους ποικίλους φυσικούς και εικονικούς πόρους να θέτονται δυναμικά και εκ νέου σύμφωνα με τη ζήτηση των καταναλωτών. (Mell & Grance, 2011)

Θα ήταν σκόπιμο να αναφέρουμε και άλλα διάφορα στοιχεία του σε σχέση με συγκεκριμένα χαρακτηριστικά.

Πίνακας 6.1: Χαρακτηριστικά Cloud

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ
Κοινή χρήση πόρων **(Resource Sharing)**	**Οι εκχωρημένοι πόροι δεν μοιράζονται.** (Assigned resources are not shared)
Ετερογένεια πόρων **(Resource Heterogeneity)**	**Πρόσμιξη ετερογενών πόρων** (Aggregation of heterogeneous resources.)
Οπτικοποίηση **(Virtualization)**	**Οπτικοποίηση πλατφόρμας υλικού και λογισμικού** (Virtualization of hardware and software platforms)
Ασφάλεια **(Security)**	**Ασφάλεια λόγω απομόνωσης** (Security through isolation.)
Υπηρεσίες υψηλού επιπέδου **(High level services)**	**Δεν έχουν καθοριστεί ακόμα υπηρεσίες υψηλού επιπέδου** (No high-level services defined yet.)
Αρχιτεκτονική **(Architecture)**	**Ο χρήστης επιλέγει την αρχιτεκτονική** (User chosen architecture.)
Εξάρτηση Λογισμικού **(Software Dependencies)**	**Η εφαρμογή παραμένει ανεξάρτητη του λογισμικού** (Application domain independent software)
Ανεξαρτησία πλατφόρμας **(Platform Awareness)**	**Το λογισμικό SP λειτουργεί σε ένα προσαρμοσμένο περιβάλλον** (The SP software works on a customized environment)
Ροή εργασιών λογισμικού **(Software Workflow)**	**Η ροή εργασίας δεν είναι απαραίτητη για τις περισσότερες εφαρμογές** (Workflow is not essential for most applications.)
Ευελιξία **(Scalability)**	**Ευελιξία στους κόμβους στους ιστοτόπου και στο υλικό** (Nodes, sites, and hardware scalability)

Αυτοδιαχείριση **(Self-Management)**	**Επαναπροσδιορισμός, αυτό-θεραπεία** (Reconfigurability, self-healing.)
Βαθμός κεντρικοποίησης **(Centralization Degree)**	**Κεντρικός έλεγχος (μέχρι τώρα)** (Centralized control (until now))
Ευχρηστία **(Usability)**	**Φιλικότητα προς τον χρήστη** (User friendliness.)
Τυποποίηση **(Standardization)**	**Έλλειψη προτύπων για τη δια λειτουργικότητα των νεφών** (Lack of standards for Clouds interoperability)
Πρόσβαση χρηστών **(User Access)**	**Διαφάνεια ως προς την πρόσβαση** (Access transparency for the end user.)
Μοντέλο πληρωμής **(Payment Model)**	**Ευέλικτο** (Flexible)
Εγγυήσεις QoS **(QoS Guarantees)**	**Περιορισμένη υποστήριξη, επικεντρωμένη στη διαθεσιμότητα και το χρόνο λειτουργίας** (Limited support, focused on availability and uptime.)

Επιπρόσθετα τα τέσσερα μοντέλα εφαρμογής νέφους δίνουν την δυνατότητα στις επιχειρήσεις να επιλέξουν σε ποιον τύπο θα αναπτύξουν τις εφαρμογές τους.

- **Δημόσιο Νέφος (Public Cloud)**

Η χρήση ενός δημόσιου νέφους προβλέπεται για ένα ευρύ κοινό. Όλοι οι χρήστες βρίσκονται στην ίδια υποδομή στην οποία περιορίζεται η διαμόρφωση, εμφανίζονται θέματα προστασίας των δεδομένων και της διαθεσιμότητας, ωστόσο είναι μεγαλύτερο από το ιδιωτικό νέφος και μπορεί να επεκταθεί. (Harris) (Mell & Grance, 2011)

- **Ιδιωτικό Νέφος (Private Cloud)**

Ένα ιδιωτικό νέφος είναι δημιουργημένο για επιχειρήσεις ούτως ώστε να υπάρχει μεγαλύτερος έλεγχος και ασφάλεια δεδομένων. Η διαχείρισή του μπορεί να γίνεται είτε από την ίδια την επιχείρηση είτε από κάποιο τρίτο πρόσωπο. (Harris)

- **Κοινοτικό Νέφος (Community cloud)**

Η υποδομή ενός κοινοτικού νέφους είναι σχεδιασμένη με τέτοιο τρόπο που εξυπηρετεί συγκεκριμένους κοινούς σκοπούς των οργανισμών ή επιχειρήσεων που ανήκουν σε αυτό.

- **Υβριδικό Νέφος (Hydridic Cloud)**

Το υβριδικό νέφος όπως λέει και η ονομασία του είναι μια σύνθεση των παραπάνω υποδομών, έχοντας δύο χαρακτηριστικά. Αρχικά, οι επιμέρους υποδομές παραμένουν ξεχωριστές οντότητες και δεύτερον, συνδέονται μεταξύ τους επιτρέποντας την μεταφορά δεδομένων. (Mell & Grance, 2011)

6.7. Μοντέλα Υπηρεσιών του Υπολογιστικού Νέφους

6.7.1. Λογισμικό ως Υπηρεσία (Software as a Service – SaaS)

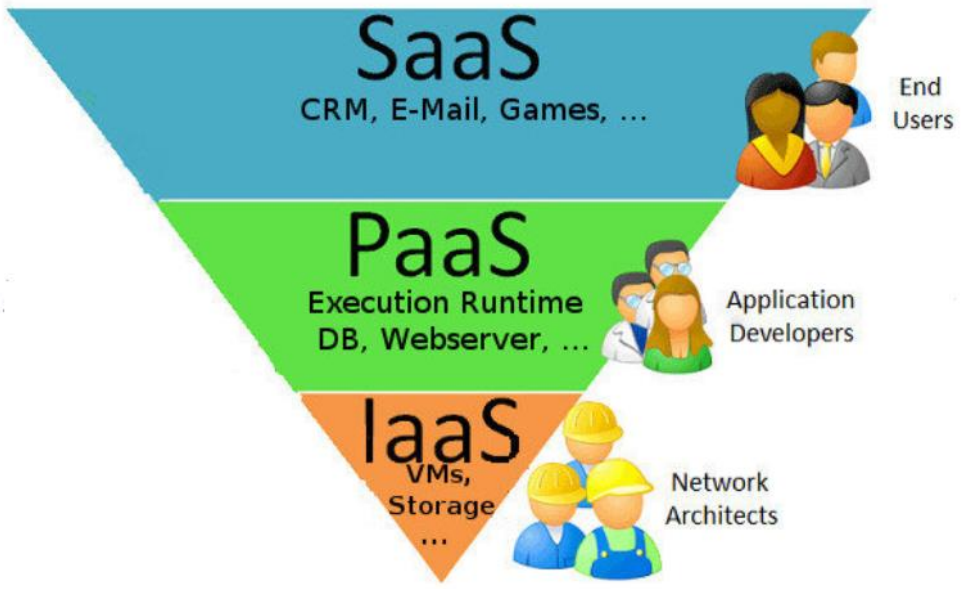
Στο SaaS, ο χρήστης δεν αγοράζει το λογισμικό, αλλά το υπενοικιάζει από έναν πάροχο υπηρεσιών. Το λογισμικό υπάρχει σε έναν κεντρικό διακομιστή και διατίθεται ως υπηρεσία. Ο χρήστης δεν έχει πρόσβαση στην υποδομή της εφαρμογής, παρά μόνο μπορεί να ζητήσει την τροποποίηση κάποιων χαρακτηριστικών την εφαρμογής από τον πάροχο. (Harris) (Mell & Grance, 2011)

6.7.2. Πλατφόρμα ως Υπηρεσία (Platform as a Service – PaaS)

Σε αυτό το μοντέλο ο χρήστης έχει την δυνατότητα να αναπτύξει τις δικές του εφαρμογές πάνω σε μια υπάρχουσα βάση λογισμικού ή σε ένα περιβάλλον ανάπτυξης. Παρέχεται ως υπηρεσία στην οποία κατασκευάζονται υψηλότερα επίπεδα εξυπηρέτησης. (Harris)

6.7.3. Υποδομή ως Υπηρεσία (Infrastructure as a Service –IaaS)

Η υποδομή ως υπηρεσία παρέχει την δυνατότητα να χρησιμοποιούνται τυποποιημένες υπηρεσίες και πόροι του υπολογιστικού νέφους, με σκοπό την αυθαίρετη ανάπτυξη και εκτέλεση λογισμικού.



Σχήμα 6.5: Μοντέλα Υπηρεσιών Υπολογιστικού Νέφους

6.8. Εμπορικές Λύσεις Cloud Computing

6.8.1. Microsoft Azure (WAS)

Το Microsoft Azure είναι μια πλατφόρμα δια λειτουργικών υπηρεσιών cloud computing, συμπεριλαμβανομένων τεχνολογιών ανοιχτού κώδικα, βασισμένων σε πρότυπα και λύσεις από τη Microsoft. Αντί να κατασκευάσει μια εγκατάσταση διακομιστή on-premise ή να νοικιάσει φυσικούς διακομιστές από τα παραδοσιακά κέντρα δεδομένων, η δομή χρέωσης της Azure βασίζεται στην κατανάλωση πόρων και όχι στην εφεδρική χωρητικότητα. Το WAS παρέχει αποθηκευτικό χώρο σε νέφη με τη μορφή Blobs (αρχεία χρήστη), Πίνακες (δομημένη αποθήκευση) και ουρές (παράδοση μηνυμάτων).

Το Microsoft Azure χρησιμοποιεί την τεχνολογία γνωστή ως οπτικοποίηση (virtualization), η οποία διαχωρίζει τη στενή σύζευξη μεταξύ της CPU ενός υπολογιστή και του λειτουργικού συστήματος της κοντά σε ένα στρώμα αφαίρεσης που ονομάζεται hypervisor. Αυτό εξομοιώνει όλες οι λειτουργίες ενός πραγματικού υπολογιστή και της cpu σε μία εικονική μηχανή. Έχει τη δυνατότητα να τρέχει πολλαπλές εικονικές μηχανές ταυτόχρονα, και κάθε εικονική μηχανή μπορεί να τρέξει οποιοδήποτε λειτουργικό σύστημα όπως τα Windows ή τα Linux. Το Microsoft Azure λαμβάνει αυτή την τεχνολογία και την επαναλαμβάνει σε μαζική κλίμακα.

Κάθε κέντρο δεδομένων της Microsoft σε όλο τον κόσμο έχει πολλά επίπεδα γεμάτα με διακομιστές και κάθε διακομιστής περιλαμβάνει ένα hypervisor για την εκτέλεση πολλαπλών εικονικών μηχανών. Ένας διακόπτης δικτύου παρέχει συνδεσιμότητα στους διακομιστές όπου καθένας τρέχει ένα ειδικό κομμάτι λογισμικού, αυτός ονομάζεται Fabric Controller. Ο Fabric Controller είναι συνδεδεμένος με ένα άλλο ειδικό λογισμικό γνωστό ως orchestrator που εκτελεί τις διαδικασίες συμπεριλαμβανομένου τις απαντήσεις στα αιτήματα των χρηστών. Τα αιτήματα πραγματοποιούνται χρησιμοποιώντας το API ιστού του orchestrator.

Όταν ένας χρήστης ζητήσει να δημιουργήσει μία εικονική μηχανή, ο orchestrator πακετάρει όλα όσα χρειάζονται, επιλέγει το καλύτερο επίπεδο με τους διακομιστές, στη συνέχεια στέλνει το πακέτο και δημιουργεί αίτημα στον Fabric Controller. Μόλις αυτός δημιουργήσει την εικονική μηχανή ο χρήστης μπορεί να συνδεθεί εκεί.

Με αυτό τον τρόπο το Microsoft Azure διευκολύνει τους προγραμματιστές να είναι ευέλικτοι στο να χτίζουν, να αναπτύσσουν και να διαχειρίζονται τις εφαρμογές και τις υπηρεσίες τους στο διαδίκτυο. Ωστόσο αυτή η ευελιξία μπορεί να έχει και απρόβλεπτες συνέπειες αν δημιουργηθούν μη εξουσιοδοτημένοι πόροι ή εάν παραμείνουν κάποιοι πόροι ενώ δεν χρειάζονται πλέον. (Calder, Wang, Ogus, Nilakantan, & Skjolsvold, 2011)

Η επιλογή του Microsoft Azure γίνεται αφού

1. Δεν διαθέτει προκαθορισμένο κόστος
2. Δεν υπάρχει αξιόλογη χρονική καθυστέρηση όσον αφορά την παροχή πόρων.

Με μια φόρμουλα χρέωσης που βασίζεται στη χρήση, το Azure είναι μια καλή επιλογή για τις επιχειρήσεις που μεταβαίνουν από τους διακομιστές Windows που είναι επιτρεπτοί στο cloud.

6.8.2. Oracle Cloud

Το Oracle Cloud είναι η δημόσια υπηρεσία cloud της εταιρείας βάσεων δεδομένων Oracle, η οποία έχει τοποθετήσει και προσαρμόσει τις υπηρεσίες του cloud ως την καλύτερη επιλογή για την εκτέλεση βάσεων δεδομένων Oracle.

Η Oracle διαθέτει ένα παγκόσμιο δίκτυο διαχειριζόμενων κέντρων δεδομένων που θέτουν σε λειτουργία τις υπηρεσίες του νέφους. Χρησιμοποιεί υπηρεσίες Oracle Cloud για την παροχή εφαρμογών cloud, διακομιστών, αποθήκευσης και επεξεργασίας σε αυτό το παγκόσμιο δίκτυο. Το ίδιο το δίκτυο είναι χτισμένο σε 25 GB Ethernet, κανένας κόμβος δεν διαχωρίζεται από άλλο κόμβο από περισσότερα από ένα hop στην επίπεδη τοπολογία του δικτύου. Αυτό το δίκτυο προσφέρει μια ισχυρή συμφωνία επιπέδου υπηρεσιών (SLA).

Οι υποδομές, οι πλατφόρμες, οι υπηρεσίες λογισμικού και δεδομένων (IaaS, PaaS, SaaS, DaaS) μπορούν να χρησιμοποιηθούν για την επέκταση ή την ενσωμάτωση των εφαρμογών μιας εταιρείας στο cloud της Oracle ή για την κατασκευή και ανάπτυξη νέων. Και οι δυο υπηρεσίες virtualized multi-tenant deployment και bare metal μπορούν να συνδεθούν με μια ενιαία διεπαφή προγράμματος εφαρμογής (API).

6.9. Πρόταση για το σύστημα

Λαμβάνοντας όλα τα παραπάνω υπόψιν η χρήση του λογισμικού ως υπηρεσία είναι κατάλληλη τόσο για μικρομεσαίες όσο και για μεγάλες επιχειρήσεις, που θέλουν να μειώσουν τα κόστη τους ενώ παράλληλα θέλουν να διασφαλίσουν και να αυξήσουν την ποιότητα των υπηρεσιών τους προς τους πελάτες.

Στην παρούσα διπλωματική εργασία υπάρχουν δύο εναλλακτικές επιλογές για την χρήση των παραπάνω παρόχων νέφους.

Platform as a Service = Azure (web services)
Infrastructure as a Service = δημιουργία της Oracle Database σε IaaS

Platform as a Service = Oracle Database
Infrastructure as a Service = Azure (web services)

Η διαφορά αυτών των δύο έχει να κάνει κυρίως με το ποια θα είναι η PaaS. Εάν επιλεχτεί η Azure ως Platform as a Service, τότε οι Υπηρεσίες συνδέονται ευκολότερα και καλύτερα κάνοντας απλά εγκατάσταση των υπηρεσιών χωρίς να απαιτείται η γνώση του στησίματος του διακομιστή. (Υπενθυμίζεται ότι τα Services και το Azure είναι της Microsoft). Στην περίπτωση που επιλεχτεί η Oracle ως Platform as a Service, η διαμόρφωση που μπορεί να γίνει είναι πολύ μικρή λόγω του ότι είναι στην γλώσσα προγραμματισμού java. Επομένως, η καλύτερη λύση είναι πρώτη.

7. Συμπεράσματα

Η παρούσα διπλωματική εργασία διερευνά τις τεχνολογίες υλοποίησης ενός Online συστήματος δρομολόγησης και προγραμματισμού διανομής, το οποίο αντιμετωπίζει στατικά δεδομένα και δυναμικές αποκλίσεις από το αρχικό πλάνο δρομολόγησης, και παρέχει ανά δρομολογήσεις για τη βελτίωση της απόδοσης των αστικών μεταφορών. Στην αρχή προτάθηκε μια συγκεκριμένη αρχιτεκτονική αυτού του συστήματος, η οποία θα μπορούσε να λειτουργεί στον πραγματικό κόσμο και να παρέχει τα επιθυμητά αποτελέσματα. Στη συνέχεια παρουσιάστηκαν οι τεχνολογίες (λογισμικά, εφαρμογές και υπηρεσίες Ιστού) που απαιτούνται για την ανάπτυξη και διασύνδεση των στοιχείων του συστήματος. Πιο αναλυτικά, το Back-end σύστημα υλοποιείται σε μια πλατφόρμα ανάπτυξης. Για το σύστημα προτάθηκε ως καλύτερη η χρήση της .NET πλατφόρμας ανάπτυξης, η οποία είναι ευκολότερη και πιο ασφαλής από την ανταγωνιστική πλατφόρμα J2EE. Στην συγκεκριμένη πλατφόρμα χρησιμοποιείται η τεχνολογία ανάπτυξης διαδικτυακών υπηρεσιών, δηλαδή ανεξάρτητα στοιχεία οποιασδήποτε εφαρμογής, που δημοσιεύονται στο Διαδίκτυο με τέτοιο τρόπο ώστε άλλες διαδικτυακές υπηρεσίες να τα βρίσκουν και να τα χρησιμοποιούν προς εξυπηρέτηση του δικού τους στόχου. Ωστόσο, επειδή οι διαδικτυακές υπηρεσίες αποτελούν κομμάτια του διαδικτύου, προτάθηκε η χρήση των API's υπηρεσιών, δηλαδή ένα υπερσύνολο διαδικτυακών υπηρεσιών που δεν απαιτούν την χρήση του διαδικτύου. Είναι «ελαφριά» αρχιτεκτονική και είναι καλό για συσκευές με περιορισμένο εύρος ζώνης, όπως οι κινητές συσκευές. Μπορούν να φιλοξενηθεί από οποιαδήποτε εφαρμογή ή και από το IIS, είναι ανοικτού κώδικα, μπορούν να χρησιμοποιηθούν από οποιονδήποτε πελάτη που κατανοεί το JSON ή το XML και υποστηρίζουν το πρωτόκολλο HTTP: URL, κεφαλίδες αιτήματος / απόκρισης, προσωρινή αποθήκευση, εκδόσεις, μορφές περιεχομένου. Τέλος, το API μπορεί να χρησιμοποιεί οποιοδήποτε πρωτόκολλο επικοινωνίας/ ανταλλαγής δεδομένων.

Για την επικοινωνία και την ανταλλαγή δεδομένων των εφαρμογών και του λογισμικού του υπό ανάλυση συστήματος, θεωρήθηκε ότι καταλληλότερο πρωτόκολλο είναι το πρωτόκολλο REST έναντι του πρωτοκόλλου SOAP, λόγω του ότι είναι απλό στη χρήση του έχει καταφέρει να εδραιωθεί σαν μοντέλο σχεδιασμού υπηρεσιών μέσω διαδικτύου, λόγω της ιδανικής αρχιτεκτονικής για την ανάπτυξη εφαρμογών διαδικτύου. Επίσης, το σύστημα που μελετάται είναι πραγματικού χρόνου, οι χρόνοι απόκρισης του παίζουν καθοριστικό ρόλο για την επιλογή του πρωτοκόλλου ανταλλαγής δεδομένων. Επομένως, η σύγκριση των δύο τεχνολογιών οδήγησε στο συμπέρασμα πως η καταλληλότερη τεχνολογία είναι το

πρωτόκολλο επικοινωνίας REST. Ακόμα, το REST φαίνεται ότι έχει καλύτερη απόδοση, επεκτασιμότητα και δια λειτουργικότητα σε σχέση με το SOAP.

Μια ακόμα τεχνολογία που παίζει σημαντικό ρόλο στο σύστημα της εργασίας είναι η βάση δεδομένων που χρησιμοποιείται. Από την διερεύνηση παρατηρήθηκε ότι υπάρχουν πολλές και διαφορετικές βάσεις δεδομένων, ο βασικότερος διαχωρισμός τους αποτελεί η χρήση ή μη της γλώσσας SQL, δηλαδή οι βάσεις διαχωρίζονται σε SQL ή σε NoSQL βάσεις δεδομένων. Αρχικά προτείνεται η χρήση των σχεσιακών βάσεων καθώς είναι ευρύτερη και πιο οικεία με τους περισσότερους χρήστες. Το υπό ανάλυση σύστημα δεν έχει βασικό στόχο μόνο την αποθήκευση δεδομένων μεγάλου όγκου όπως κάνουν οι NoSQL βάσεις, αλλά και την διαρκή μεταξύ τους σχέση.

Όσον αφορά τις SQL βάσεις δεδομένων, η MS Access χρησιμοποιείται από μικρές επιχειρήσεις ή από τμήματα μεγάλων επιχειρήσεων, επομένως δεν θεωρείται κατάλληλη για το υπό ανάλυση σύστημα που χρειάζεται μια στοιβαρή και αρκετά δυναμική βάση. Η SQLite είναι πολύ ελαφριά (είναι μικρότερη από 500Kb) σε σύγκριση με άλλα συστήματα διαχείρισης βάσεων δεδομένων όπως το SQL Server ή το Oracle, και δεν αντέχει τον όγκο πληροφοριών και δεδομένων που υπεισέρχονται στο σύστημα. Ακόμα, το MySQL είναι ένα δωρεάν ανοιχτό σύστημα διαχείρισης βάσεων δεδομένων SQL, με πολλαπλές χρήσεις και πολλαπλούς χρήστες, το οποίο δεν μπορεί να διαθέσει την ασφάλεια και την κλιμάκωση που χρειάζεται το σύστημα. Στη συνέχεια προτείνεται ως βάση δεδομένων για το λογισμικό ως υπηρεσία η Oracle, η οποία λύνει απαιτητικά επιχειρηματικά προβλήματα και παρέχει μεγάλη αξιοπιστία, κλιμάκωση και απόδοση, για βάσεις δεδομένων και εφαρμογές. Βασικό πλεονέκτημά της είναι το γεγονός ότι είναι άμεση η σύνδεση της συγκεκριμένης βάσης με την αντίστοιχη υποδομή ως υπηρεσία στο υπολογιστικό νέφος.

Παρουσιάστηκαν οι τρείς τρόποι με τους οποίους παρέχεται ένα λογισμικό σε μια εταιρίας. Πιο συγκεκριμένα, η ανάλυση ξεκίνησε από την τοπική εγκατάσταση του λογισμικού στους διακομιστές της εκάστοτε εταιρίας, συνεχίστηκε στην χρήση του «φιλοξενούμενου λογισμικού» δηλαδή λογισμικού που βρίσκεται σε απομακρυσμένους διακομιστές όχι σε αυτούς της εταιρίας, και κατέληξε στην νεότερη λύση αυτής του Λογισμικού ως Υπηρεσίας, δηλαδή λογισμικού που βρίσκεται σε ένα υπολογιστικό νέφος και η χρήση του γίνεται μέσω διαδικτύου με τις κατάλληλες άδειες. Ήταν αναγκαίο να παρουσιαστούν και οι παλιότερες μέθοδοι, ούτως ώστε να υπάρχει μια ολοκληρωμένη άποψη στους λόγους που στην αρχιτεκτονική του συστήματος έχει προταθεί το Λογισμικό να παρέχεται ως υπηρεσία στο Cloud.

Παρουσιάζονται αναλυτικά τα στοιχεία που το Λογισμικό ως Υπηρεσία αποτελεί μια κατάλληλη λύση για το σύστημα, τόσο από την οπτική γωνία της εταιρίας logistics όσο και από την οπτική γωνία του παρόχου του λογισμικού μέσω του συστήματος. Όπως έγινε φανερό τα τελευταία χρόνια έχει παρατηρηθεί ότι τα πλεονεκτήματα στη χρήση του λογισμικού σε ένα υπολογιστικό νέφος για την επίλυση διαφόρων λειτουργικών διαδικασιών είναι μια συμφέρουσα λύση για τις εταιρίες. Για τις μικρομεσαίες επιχειρήσεις (Small and Medium Enterprises – SME) θεωρείται ως η καλύτερη ευκαιρία, αφού δεν χρειάζεται η επένδυση χρημάτων στο λογισμικό και στην διαχείριση /συντήρηση του, ενώ οι μεγάλες επιχειρήσεις (Large Enterprises – LE) το θεωρούν ως συμπλήρωμα του ήδη υπάρχοντος λογισμικού. Και στις δύο περιπτώσεις μειώνεται η υπολογιστική ισχύς που απαιτείται να έχει μια εταιρία ώστε να φιλοξενήσει το λογισμικό που χρειάζεται.

Στο τελευταίο κεφάλαιο έγινε μια εκτενής αναφορά στο υπολογιστικό νέφος, στην εξέλιξη του, στα μοντέλα υπηρεσιών του, και στις εμπορικές λύσεις του νέφους. Προτάθηκε για το σύστημα ότι η πλατφόρμα ως υπηρεσία θα είναι η Microsoft Azure (WAS) και υποδομή ως υπηρεσία Oracle Cloud, διότι οι υπηρεσίες API's συνδέονται ευκολότερα και καλύτερα με την συγκεκριμένη πλατφόρμα, καθώς αποτελούν προϊόντα του ίδιου παρόχου (Microsoft), και από την άλλη πλευρά η σύνδεση της βάσης δεδομένων της oracle με την υποδομή ως υπηρεσία της ίδιας εταιρίας είναι αρκετά εύκολη καθώς και αυτές αποτελούν προϊόντα της ίδιας εταιρίας.

Συνοψίζεται ότι με βάση την αρχική αρχιτεκτονική του συστήματος οι τεχνολογίες που προτείνονται για την υλοποίηση του Λογισμικού ως Υπηρεσία είναι οι εξής:

Τεχνολογίες	Προτάσεις
Πλατφόρμα ανάπτυξης	Microsoft.net
Διαδικτυακές υπηρεσίες	API's services
Πρωτόκολλα ανταλλαγής δεδομένων	REST
Βάσεις δεδομένων	Oracle database
Μοντέλο Παροχής Λογισμικού	Λογισμικό ως Υπηρεσία (SaaS)
Πλατφόρμα ως Υπηρεσία	Microsoft Azure (WAS)
Υποδομή ως Υπηρεσία	Oracle Cloud

Στην παρούσα διπλωματική εργασία αξιολογήθηκαν όλες οι τεχνολογίες που είναι αναγκαίες για την υλοποίηση της προτεινόμενης αρχιτεκτονικής του συστήματος, που έχει ως στόχο την δρομολόγηση και τον προγραμματισμού αστικών διανομών, που λειτουργεί σε περιβάλλον

cloud, και λαμβάνει υπόψιν του παραμέτρους που έχουν σχέση με την δρομολόγηση και με την ανταλλαγή δεδομένων μεταξύ των συμπλεκόμενων μερών. Μέσα από την αξιολόγηση των τεχνολογιών, έγινε φανερό ότι με την κατάλληλη χρήση αυτών στο υπό ανάλυση σύστημα, μπορούν οι εταιρίες Logistics να χρησιμοποιούν το Λογισμικό ως Υπηρεσία μέσω του υπολογιστικού νέφους με ευκολία, αξιοπιστία και ασφάλεια, έχοντας σαν αποτέλεσμα καλύτερη εποπτεία των διαδικασιών τους, αυξημένη απόδοση, διασφάλιση της ποιότητας των υπηρεσιών/ προϊόντων τους, καθοριστικά για εταιρίες που λειτουργούν στο σύγχρονο ανταγωνιστικό περιβάλλον.

Βιβλιογραφία

Alassafi, M. O., Alharthi, A., Walters, R. J., & Wills, G. (2019, April 10). *A Framework for Critical Security Factors that Influence the Decision of Cloud Adoption by Saudi Government Agencies*. Ανάκτηση από Researchgate.net: https://www.researchgate.net/figure/Conceptual-view-of-cloud-computing_fig1_316595509

Alcon, D. J. (2016). *FROM SOFTWARE AS A GOOD TO SOFTWARE AS A SERVICE (SAAS): A METHOLOGY TO DEFINE THE TRANSFORMATION TOWARDS THE SAAS BUSINESS MODEL.* PhD Thesis, Universitat Abat Oliba CEU , Departamento de Empresa y Economia.

Almishal, A., & Youssef, A. E. (May 2014). Cloud Service Providers: A Comparative Study. *International Journal of Computer Applications & Information Technology, 5*(2), 7.

Bassil, Y. (2011). A Comparative Study on the Performance of the Top DBMS Systems. *Journal of Computer Science & Research (JCSCR)*, 20-31.

Berg, K., Seymour, D. T., & Goel, R. (2012). History Of Databases. *International Journal of Management & Information Systems*, 29-36.

Birger. (2018, January 10). *Birger*. Ανάκτηση από https://www.birger.technology/blog/10-benefits-of-saas-for-an-sme/

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., . . . Winer, D. (2000, August 22). *W3C.* Ανάκτηση από W3C Web site: www.w3c.org

Bueno, M. d. (2011). *Spatial Data, Use and Dissemination.*

Calder, B., Wang, J., Ogus, A., Nilakantan, N., & Skjolsvold, A. (2011). Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency. *SOSP'11*, 143-157.

Churakova, I., & Mikhramova, R. (June 2010). *Software as a Service: Study and Analysis of SaaS Business Model and Innovation Ecosystems.* Master Thesis, UNIVERSITEIT GENT, FACULTEIT ECONOMIE EN BEDRIJFSKUNDE, Belgium.

Clausen, U., Friedrich, H., Thaller, C., & Geiger, C. (2016). Commercial Transport. *2nd Interdisciplinary Conference on Production, Logistics and Traffic (ICPLT) 2015* (σσ. XV, 440). Switzerland: Springer International Publishing.

Curbera, F., Nagy, W. A., & Weerawarana, S. (2001). *Web Services: Why and How.*

Dar, R. M., & Riyaz, R. (2017). Grid Computing: An Insight into Heterogeneous Computing Environment. *International Journal of Advanced Research in Computer Science and Software Engineering*, 57-61.

EDUBCA. (2019). *ASP.NET vs .NET*. Ανάκτηση από EDUBCA: https://www.educba.com/asp-net-vs-net/

Esri. (2011). *GIS Γενική παρουσίαση.* Ανάκτηση από Docplayer: https://docplayer.gr/5785928-Gis-geniki-paroysiasi.html

Fangchun, Y., Shangguang, W., Jinglin, L., Zhihan, L., & Qibo, S. (2014). *An Overview of Internet of Vehicles.* Beijing: China Communications.

Feuerlicht, G., & Lozina, J. (2007). *Understanding Service Reusability.* University of Technology, Faculty of Information Technology, Sydney, Australia.

Foote, K. D. (2017, March 23). *A Brief History of Database Management*. Ανάκτηση από Dataversity : https://www.dataversity.net/brief-history-database-management/#

Gerami, M. (2010, AUGUST). Some Basic Concepts of Grid Computing. *JOURNAL OF TELECOMMUNICATIONS, 4*(1), 14-17.

Harris, T. (χ.χ.). *CLOUD COMPUTING – An Overview.*

Ho-fung, L. (September 30, 2010). *Service Intelligence and Service Science: Evolutionary Technologies and Challenges.* (D. K. Chiu, & P. C. Hung, Επιμ.) IGI Global.

Johanssona, B., & Ruivob, P. (2013). Exploring Factors for Adopting ERP as SaaS. *CENTERIS 2013 - Conference on ENTERprise Information Systems / ProjMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies* (σσ. 94 – 99). Elsevier Ltd.

Kreger, H. (2002). Introduction to Web services architecture. *IBM SYSTEMS JOURNAL*, 170-177.

Kulkarni, G. A., & Gambhir, J. (Jan-Feb 2012). Cloud Computing-Software as Service. *International Journal of Computer Science & Information Technology Research Excellence, 2*(1), 90-93.

Litwin, B. (2017). *An Introduction to Web Service Stacks And 3 Options to Consider*. Ανάκτηση από A2hosting: https://www.a2hosting.com/blog/web-service-stack/

Mell, P., & Grance, T. (2011, September). *The NIST Definition of Cloud Computing.* Special Publication 800-145.

Miller, G. (2003, June). The Web services debate: .NET vs. J2EE. *COMMUNICATIONS OF THE ACM*, σσ. 58-63.

Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation.* Canada: John Wiley & Sons, Inc., Hoboken, New Jersey.

Patel, A., Seyfi, A., Tew, Y., & Jaradat, A. (2011). Comparative study and review of grid, cloud, utility computing and software as a service for use by libraries. *Library Hi Tech News* , 25-32 .

Pautasso, C., Zimmermann, O., & Leymann, F. (2008). RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision. *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, (σσ. 805-814). Beijing.

Qian, L., Luo, Z., Du, Y., & Guo, L. (January 2007). Cloud Computing: An Overview. Στο M. G. Jaatun, G. Zhao, & C. Rong, *Cloud Computing* (σ. 707). Berlin, Germany: Springer-Verlag Berlin and Heidelberg GmbH & Co. KG.

Resources, T. A. (2018, May). *6 Differences between Web Services vs API*. Ανάκτηση από Test Automation Resources: https://testautomationresources.com/api-testing/differences-web-services-api/

Rittinghouse, J., & Ransome, J. (August 17, 2009). *Cloud Computing: Implementation, Management, and Security.* CRC Press.

Rodriguez, A. (2008). *© Copyright IBM Corporation 2008.* IBM.

Roy, J., & Ramanujan, A. (2001, Nov/Dec). Understanding Web Services. *IT Professional* , σσ. 69 - 73.

Sabiri, C. (2017). *Capacitated Vehicle Routing Problem with Fixed Fleet of Delivery Vehicles of Uniform capacity.* Capstone, AL AKHAWAYN UNIVERSITY, SCHOOL OF SCIENCE AND ENGINEERING .

Schultze, A. (December 3, 2007). *Channel Excellence* (1st εκδ.).

Souza, C. D., Redmiles, D., Cheng, L.-T., & Millen, D. (2004). How a good software practice thwarts collaboration the multiple roles of APIs in software development. *12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, (σσ. ΄221-230).

Spruit, M. (2019, April 10). *Software as a service and the pricing strategy for vendors*. Ανάκτηση από Researchgate: https://www.researchgate.net/figure/On-premise-vs-software-as-a-service_fig2_289223600

Tanenbaum, A. S., & Steen, M. v. (February 26, 2016). *Distributed Systems: Principles and Paradigms* (2nd Edition εκδ.). CreateSpace Independent Publishing Platform.

Technopedia. (2019, April). *Java Runtime Environment (JRE)*. Ανάκτηση από Technopedia: https://www.techopedia.com/definition/5442/java-runtime-environment-jre

Technopedia. (2019, April). *Web Service*. Ανάκτηση από Technopedia web site : https://www.techopedia.com/definition/25301/web-service

Tiwari, P., Shrivastava, R. K., Pandey, S., & Tripathi, P. K. (2011). Cloud Computing Utility and Applications. *International Journal of Advanced Computer Research*, 54-58.

Tsai, W.-T., Sun, X., & Balasooriya, J. (2010). Service-Oriented Cloud Computing Architecture. *2010 Seventh International Conference on Information Technology*, (σ. 6). Arizona.

Uriarteq, R. B. (2019, April 10). *Supporting Autonomic Management of Clouds: Service-Level-Agreement, Cloud Monitoring and Similarity Learning*. Ανάκτηση από Researchgate.net: https://www.researchgate.net/figure/Cloud-delivery-models-examples-of-applications-and-their-target-consumers_fig6_276062277

Vardhan, G. V., & Rahamathulla, V. (2018). SURVEY OF VARIOUS DATABASES AND ITS APPLICATIONS IN THE REAL WORLD . *Internation Journal of Advanced Research in Computer Science*, 237-239.

Vawter, C., & Roman, E. (2001). *J2EE vs. Microsoft.NET - A comparison of building XML-based web services.* The Middleware Company.

Visual Basic. (2019). Ανάκτηση από Wikipedia : https://el.wikipedia.org/wiki/Visual_Basic

Δημανίδης, Α. (2017). *Ανάπτυξη RESTful APIs με τη χρήση της γλώσσας Gherkin και του OpenAPI Specification.* Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

Δουληγέρης, Χ., & Δουληγέρης, Χ. (2015). ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ. Στο Χ. Δουληγέρης, & Χ. Δουληγέρης, *Πληροφοριακά συστήματα στο διαδίκτυο.* Αθήνα: Εκδόσεις Κάλλιπος.

ΕΙΕ, Ε. Κ. (χ.χ.). *SaaS Βασικές έννοιες Πνευματικής Ιδιοκτησίας και Προσωπικών Δεδομένων.* Αθήνα: Νομικές Εκδόσεις.

Πλασταράς, Ε. (2016). *ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΤΕΧΝΟΛΟΓΙΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ REST.* Θεσσαλονίκη.