



**Πανεπιστήμιο Πελοποννήσου**  
**Σχολή Θετικών Επιστημών και Τεχνολογίας**  
**Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών**

## **Μεταπτυχιακή Διπλωματική Εργασία**

**Ανάπτυξη και ποιοτικά χαρακτηριστικά εφαρμογών στο  
υπολογιστικό νέφος**

**ΓΑΣΠΑΡΗ ΜΙΧΑΕΛΑ-ΧΡΙΣΤΙΝΑ**  
**ΑΜ 2010004**

**ΛΑΖΑΡΗ ΕΥΓΕΝΙΑ**  
**Α.Μ. 2010014**

**Επιβλέπων Καθηγητής: Βασιλάκης Κωνσταντίνος**

**Τρίπολη, Νοέμβριος 2012**

Copyright © 2012

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών της Σχολής Θετικών Επιστημών και Τεχνολογίας του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

...αφιερώνεται ...

## Ευχαριστίες

## Περίληψη

Στη διπλωματική αυτή εργασία γίνεται μία επισκόπηση ζητημάτων που αφορούν το υπολογιστικό νέφος. Αρχικά εξετάζονται οι βασικές έννοιες και δομές, δηλαδή η αρχιτεκτονική του, τα μοντέλα εφαρμογών και υπηρεσιών και παρουσιάζονται και παραδείγματα πλατφορμών. Στη συνέχεια, εμβαθύνουμε στα ζητήματα της ανάπτυξης εφαρμογών στο υπολογιστικό νέφος, σε θέματα ποιότητας υπηρεσίας καθώς και στην ασφάλεια και ιδιωτικότητα δεδομένων. Κλείνοντας, εξετάζουμε τις διαφαινόμενες προοπτικές και τάσεις που αφορούν το υπολογιστικό νέφος.

## **Abstract**

The present thesis overviews aspects and issues related to the computational cloud. Firstly, we examine the cloud's basic concepts and structures, i.e. its architecture, the application and services models and we present examples of cloud platforms. Afterwards, we study in more detail the aspects of application development, quality of service and security & privacy within the cloud. In the last section, we discuss the cloud's prospects and trends.

## Περιεχόμενα

<i>Ευχαριστίες</i>	4
<i>Περίληψη</i>	5
<i>Abstract</i>	6
<i>Περιεχόμενα</i>	7
<i>Ευρετήριο σχημάτων</i>	10
<b>Κεφάλαιο 1: Εισαγωγή. Ορισμός του υπολογιστικού νέφους και τα πλεονεκτήματά του</b>	<b>11</b>
<b>1.1 Ορισμός του υπολογιστικού νέφους</b>	<b>11</b>
<b>1.2 Τύποι υπολογιστικών νεφών</b>	<b>12</b>
1.2.1 Μοντέλο NIST	12
1.2.2 Μοντέλο κύβου για το νέφος	13
1.2.3 Μοντέλα Ανάπτυξης	14
1.2.4 Μοντέλα Υπηρεσιών	15
<b>1.3 Παραδείγματα υπολογιστικού νέφους</b>	<b>17</b>
<b>1.4 Πλεονεκτήματα υπολογιστικού νέφους</b>	<b>19</b>
<b>1.5 Μειονεκτήματα υπολογιστικού νέφους</b>	<b>20</b>
<b>Κεφάλαιο 2: Η αρχιτεκτονική του υπολογιστικού νέφους</b>	<b>22</b>
<b>2.1 Εξερεύνηση της αρχιτεκτονικής στοίβας της υπολογιστικής νέφους</b>	<b>22</b>
<b>2.2 Συνθεσιμότητα</b>	<b>24</b>
<b>2.3 Υποδομή</b>	<b>26</b>
<b>2.4 Υπολογιστικό πλέγμα</b>	<b>28</b>
2.4.1 Παράδειγμα Υπολογιστικού πλέγματος - SETI	29
<b>2.5 Υπολογιστική συναλλαγών</b>	<b>31</b>
<b>Κεφάλαιο 3: Μοντέλα εφαρμογών και υπηρεσιών στο υπολογιστικό νέφος</b>	<b>34</b>
<b>3.1 Υποδομή ως υπηρεσία (ΥωΥ)</b>	<b>35</b>
3.1.1 Φόρτος εργασίας του ΥωΥ	36
<b>3.2 Πλατφόρμα ως υπηρεσία (ΠωΥ)</b>	<b>38</b>
<b>3.3 Λογισμικό ως υπηρεσία (ΛωΥ)</b>	<b>39</b>
<b>3.4 Ταυτότητα ως υπηρεσία (ΤωΥ)</b>	<b>41</b>
<b>3.5 Συμμόρφωση ως υπηρεσία (ΣωΥ)</b>	<b>42</b>
<b>3.6 Υλικό ως υπηρεσία (HaaS)</b>	<b>44</b>
<b>Κεφάλαιο 4: Πλατφόρμες για το υπολογιστικό νέφος</b>	<b>45</b>
<b>4.1 Amazon EC<sup>2</sup></b>	<b>48</b>
<b>4.2 Google App Engine</b>	<b>53</b>
4.2.1 Google Datastore	54

4.3	Microsoft Azure	57
<b>Κεφάλαιο 5: Ανάπτυξη εφαρμογών στο υπολογιστικό νέφος</b>		<b>61</b>
5.1	Μεταφορά εφαρμογών στο νέφος	62
5.2	Τρόποι σύνδεσης χρηστών με το νέφος	62
5.3	Εφαρμογές στα νέφη	63
5.3.1	Ιδιότητες αξιόπιστων δοσοληψιών	64
5.3.2	Χαρτογράφηση λειτουργικότητας	65
5.3.3	Χαρακτηριστικά γνωρίσματα εφαρμογής	67
5.3.4	Χαρακτηριστικά γνωρίσματα υπηρεσιών νέφους	67
5.3.5	Αφαιρετικότητα συστήματος	68
5.3.6	Έκρηξη νέφους (cloud bursting)	70
5.4	Λειτουργικό σύστημα Jolicloud (για Netbook)	72
5.5	Λειτουργικό σύστημα Chromium (το πρόγραμμα πλοήγησης ως λειτουργικό σύστημα)	74
5.6	Λογισμικό Gbridge	76
<b>Κεφάλαιο 6: Θέματα ποιότητας υπηρεσίας στο υπολογιστικό νέφος</b>		<b>78</b>
6.1	Πρότυπα υψηλής κλιμάκωσης της εφαρμογής που ταιριάζουν καλύτερα στο νέφος	78
6.1.1	Δυνατότητα Μεταφοράς	78
6.1.2	Κλιμάκωση σε Επίπεδο Διαδικτύου	79
6.1.3	Έκρηξη υπολογισμού	79
6.1.4	Ελαστική αποθήκευση	79
6.2	Αξιοπιστία νεφών	80
6.2.1	Πλεονασμός (Redundancy)	80
6.2.2	MapReduce	82
6.2.2.1	Το βήμα step	83
6.2.2.2	Το βήμα reduce	84
6.2.2.3	Λειτουργία του MapReduce	85
6.2.3	Hadoop: το ανοικτού κώδικα MapReduce	87
6.3	SLAs και δημόσιοι πάροχοι νεφών	88
6.3.1	SLA του Amazon AWS	88
6.3.2	SLA του Microsoft Azure	89
6.3.3	SLA του νέφους Rackspace	90
<b>Κεφάλαιο 7: Ασφάλεια και ιδιωτικότητα στο υπολογιστικό νέφος</b>		<b>92</b>
7.1	Διαχείριση ασφάλειας στο νέφος	92
7.1.1	Πρότυπα διαχείρισης Ασφαλείας	94
7.1.1.1	ITIL	94
7.1.1.2	ISO 27001/27002	95
7.1.2	Διαχείριση ασφαλείας στο νέφος	96
7.2	Διαχείριση ασφαλείας ευπάθειας, επιδιορθώσεων και ρύθμισης παραμέτρων	96
7.2.1	Διαχείριση ευπαθειών	97
7.2.2	Διαχείριση επιδιορθώσεων ασφαλείας	98
7.2.3	Διαχείριση διαμόρφωσης ασφαλείας	98
7.3	Απόρρητο – προστασία προσωπικών δεδομένων	99
7.3.1	Έννοια απορρήτου	99
7.3.2	Κύκλος ζωής δεδομένων	99



7.3.2.1	Παραγωγή της πληροφορίας (Generation of the information)	100
7.3.2.2	Χρήση (Use)	100
7.3.2.3	Μεταφορά (Transfer)	101
7.3.2.4	Μετασχηματισμός (Transformation)	101
7.3.2.5	Αποθήκευση (Storage)	101
7.3.2.6	Αρχειοθέτηση (Archival)	102
7.3.2.7	Καταστροφή (Destruction)	102
<b>7.4</b>	<b>Προβληματισμοί που αφορούν την ασφάλεια προσωπικών δεδομένων στο νέφος</b>	<b>102</b>
7.4.1	Πρόσβαση (Access)	102
7.4.2	Συμμόρφωση (Compliance)	102
7.4.3	Αποθήκευση (Storage)	103
7.4.4	Διατήρηση (Retention)	103
7.4.5	Καταστροφή (Destruction)	103
7.4.6	Έλεγχος και παρακολούθηση (Audit and monitoring)	103
7.4.7	Παραβιάσεις απορρήτου (Privacy breaches)	104
<b>7.5</b>	<b>Αλλαγές στη διαχείριση και τη συμμόρφωση κινδύνου προστασίας προσωπικών δεδομένων σε σχέση με το υπολογιστικό νέφος</b>	<b>104</b>
7.5.1	Αρχή περιορισμού συλλογής (Collection Limitation Principle)	104
7.5.2	Αρχή περιορισμού χρήσης (Use Limitation Principle)	104
7.5.3	Αρχή ασφάλειας (Security Principle)	105
7.5.4	Αρχή διατήρησης και καταστροφής (Retention and Destruction Principle)	105
7.5.5	Αρχή μεταφοράς (Transfer Principle)	105
7.5.6	Αρχή υπευθυνότητας (Accountability Principle)	106
<b>Κεφάλαιο 8:</b>	<b>Το μέλλον του υπολογιστικού νέφους</b>	<b>107</b>
<b>8.1</b>	<b>Προβλέψεις για το μέλλον του νέφους</b>	<b>108</b>
8.1.1	Εργαλείο μεγέθυνσης για τους νεοεισερχόμενους	108
8.1.2	Αξιοπιστία – οικονομία – ασφάλεια – ευκολία	108
8.1.3	Αύξηση των εξυπηρετών στα κέντρα δεδομένων νέφους	109
8.1.4	Μειωμένο κόστος εν συγκρίσει με τα εταιρικά κέντρα δεδομένων	109
8.1.5	Μείωση της αναλογίας διαχειριστών προς εξυπηρετές	109
8.1.6	Η εξάπλωση του ανοικτού κώδικα	110
8.1.7	Πραγματικά πρότυπα μέσω των APIs της Amazon	110
8.1.8	Η αύξηση της χρήσης ΛωΥ από τα βασικά πρότυπα του Ιστού	111
8.1.9	Το πρότυπο νέφους ISO	111
8.1.10	Η κυβερνητική πρωτοβουλία στην υιοθέτηση νεφών	113
<b>8.2</b>	<b>Προβλέψεις για το μέλλον των εφαρμογών που εκτελούνται σε νέφη</b>	<b>114</b>
8.2.1	Μειωμένο κόστος ανάπτυξης εφαρμογών και ταχεία εξέλιξη στους μηχανισμούς αποθήκευσης	114
8.2.2	Ο ρόλος των πλαισίων ανάπτυξης εφαρμογών	114
8.2.3	Πολύ-επίπεδο νέφος	115
8.2.4	Μεγαλύτερη προστασία των ευαίσθητων δεδομένων	116
8.2.5	Υψηλότερου επιπέδου υπηρεσίες με μοναδικά APIs	117
8.2.6	Υιοθέτηση και αύξηση των mashups	118
8.2.7	Εξέλιξη των εργαλείων ανάπτυξης για τη κατασκευή mashups	119
8.2.8	Η χρήση των ΠωΥ και FaaS ως κυρίαρχα εργαλεία	121
<b>Αναφορές</b>		<b>122</b>

## Ευρετήριο σχημάτων

Εικόνα 1 - Οι ορισμοί υπολογιστικού νέφους του NIST	12
Εικόνα 2 - Το μοντέλο νέφους κύβου	14
Εικόνα 3 - Θέσεις επέκτασης για διαφορετικούς τύπους νεφών	15
Εικόνα 4 - Μοντέλο αναφοράς νέφους	17
Εικόνα 5 - Google AppS	18
Εικόνα 6 - Microsoft Windows Azure	18
Εικόνα 7 - Amazon Web Services	18
Εικόνα 8 - Επίπεδα της στοίβας υπολογισμού στην αρχιτεκτονική του νέφους	24
Εικόνα 9 - Συνθεσιμότητα μεταξύ διαφόρων επιπέδων νέφους	25
Εικόνα 10 - Μέρος του σωρού του υπολογισμού νεφών που έχει οριστεί ως εξυπηρέτης	27
Εικόνα 11 - Αρχιτεκτονική Υπολογιστικού Πλέγματος (Grid Computing)	29
Εικόνα 12 - Η αρχιτεκτονική του υπολογιστικού πλέγματος (grid computing) διαχωρίζει την εφαρμογή πυρήνα από τους κόμβους επεξεργασίας των δεδομένων της	30
Εικόνα 13 - Αρχιτεκτονική Υπολογιστικών Συναλλαγών (Transactional Computing)	31
Εικόνα 14 - Μια εφαρμογή συναλλαγών χωρίζει μια εφαρμογή σε παρουσίαση, επιχειρηματική λογική και αποθήκευση δεδομένων	33
Εικόνα 15 - Οικοσύστημα Υπολογιστικού Νέφους (Cloud Computing Ecosystem)	34
Εικόνα 16 - Υποδομή ως υπηρεσία (YωY)	36
Εικόνα 17 - Ένα εικονικό ιδιωτικό τμήμα εξυπηρέτη σ' ένα νέφος YωY	37
Εικόνα 18 - Πλατφόρμα ως υπηρεσία (ΠωY)	39
Εικόνα 19 - Λογισμικό ως υπηρεσία (ΛωY)	41
Εικόνα 20 - Συμμόρφωση ως υπηρεσία (ΣωY)	44
Εικόνα 21 - Μία εικονική συσκευή είναι ένα λογισμικό που εγκαθίσταται ως ενδιάμεσο λογισμικό σε μια εικονική μηχανή	46
Εικόνα 22 - Η κορυφή της διεπαφής της υπολογιστικής νεφών περιλαμβάνει τη διεπαφή του χρήστη και το API για το επίπεδο της εφαρμογής	47
Εικόνα 23 - Υποδομή νέφους Amazon	52
Εικόνα 24 - Google App Engine	54
Εικόνα 25 - Google Datastore	55
Εικόνα 26 - Microsoft Azure	60
Εικόνα 27 - Δημιουργία χάρτη χαρακτηριστικών γνωρισμάτων για την παρουσίαση της λειτουργικότητας	66
Εικόνα 28 - Εφαρμογή που αναπτύσσεται εξ ολοκλήρου στο νέφος	69
Εικόνα 29 - Εφαρμογή που διαχειρίζεται την υπερχειλίση δοσοληψιών σ' ένα σύστημα κρατήσεων (παράδειγμα έκρηξης νέφους)	71
Εικόνα 30 - Παράδειγμα χρηστών σε μεγάλους οργανισμούς	72
Εικόνα 31 - Εικόνα από το λειτουργικό σύστημα Jolicloud	74
Εικόνα 32 - Λειτουργικό σύστημα Chrome	76
Εικόνα 33 - Χαρακτηριστικό γνώρισμα SecureShares του Gbridge για ασφαλή σύνδεση ενός υπολογιστή με άλλον με χρήση του Gtalk	77
Εικόνα 26 - Ρόλοι master και worker	83
Εικόνα 27 - Λειτουργίες map και reduce σ' ένα μοντέλο MapReduce	84
Εικόνα 28 - Λειτουργία MapReduce	87
Εικόνα 29 - Υπηρεσίες ταυτότητας	93
Εικόνα 30 - Κύκλος ζωής ITIL σε μια επιχείρηση	95
Εικόνα 31 - Κύκλος ζωής δεδομένων	100
Εικόνα 34 - Μέλλον του υπολογιστικού νέφους	107
Εικόνα 35 - Μέλλον του Διαδικτύου συναρτήσει του υπολογιστικού νέφους	112
Εικόνα 36 - Υπηρεσίες πολυ-επίπεδου νέφους	116
Εικόνα 37 - Google Maps	119
Εικόνα 38 - Mashup σύστημα	121

## Κεφάλαιο 1: Εισαγωγή. Ορισμός του υπολογιστικού νέφους και τα πλεονεκτήματά του

Η υπολογιστική νέφος (cloud computing) είναι η παροχή υπολογιστικών και αποθηκευτικών πόρων μέσω ενός δικτύου υπολογιστών στους τελικούς χρήστες, έπειτα από σχετικό αίτημα. Το υπολογιστικό νέφος συνδέεται άμεσα με διαφόρων ειδών εφαρμογές και υπηρεσίες, όπου αυτές εκτελούνται σε ένα κατακευματισμένο δίκτυο κάνοντας χρήση εικονικών (virtual) και απεριόριστων (unlimited) πόρων.

Τα συστήματα υπολογιστικού νέφους έχουν ευρέως διαδοθεί στις μέρες μας και επεκτείνονται καθημερινά ολοένα και περισσότερο, καθώς το νέφος χρησιμοποιείται πλέον και από ένα μεγάλο πλήθος χρηστών. Σε αυτό προφανώς συνέβαλλε και η ραγδαία εξέλιξη της τεχνολογίας, η ανάπτυξη της πληροφορικής και της χρήσης του Διαδικτύου και των δυνατοτήτων που παρέχει αυτό παγκοσμίως. Το υπολογιστικό νέφος αποτελεί μια σημαντική εξέλιξη για τον κόσμο του Διαδικτύου.

Επίσης χρήσιμο είναι να αναφερθεί πως δεν επωφελούνται όλες οι απόψεις των εφαρμογών από τη χρήση του υπολογιστικού νέφους: για παράδειγμα, ο έλεγχος των δοσοληψιών, η ασφάλεια και η τήρηση των κανόνων είναι πιθανό να υποβαθμιστούν αν πραγματοποιηθεί μετάβαση από ένα παραδοσιακό πλαίσιο υλοποίησης των εφαρμογών στο πλαίσιο του υπολογιστικού νέφους.

Για να χαρακτηρίζεται μία υπηρεσία ως *υπηρεσία νέφους*, θα πρέπει η υπηρεσία να είναι εν δυνάμει προσιτή μέσω οποιουδήποτε υπολογιστή στο διαδίκτυο. Το νέφος μειώνει το κόστος λειτουργίας των υπηρεσιών, καθώς ελαχιστοποιεί το κόστος κτήσης υπολογιστικής υποδομής και βασίζεται στο μοντέλο πληρωμής ανάλογα με τη χρήση.

### 1.1 Ορισμός του υπολογιστικού νέφους

Η έννοια του όρου “νέφος” δεν έχει αποσαφηνιστεί πλήρως. Με τη χρήση της έννοιας του “νέφους” κυρίως αναφερόμαστε σε δύο έννοιες:

- **Αφαίρεση (Abstraction):** Το υπολογιστικό νέφος αποκρύπτει τις λεπτομέρειες της εφαρμογής και των συστημάτων από τους χρήστες και τους υπεύθυνους ανάπτυξης. Οι εφαρμογές εκτελούνται σε φυσικά συστήματα που δεν είναι προσδιορισμένα, τα δεδομένα αποθηκεύονται σε τοποθεσίες που είναι άγνωστες, η διαχείριση των συστημάτων εκτελείται από άλλους, και η πρόσβαση από τους χρήστες γίνεται από παντού.
- **Εικονικοποίηση (Virtualization):** Το υπολογιστικό νέφος εικονικοποιεί τα συστήματα με τη συγκέντρωση και τη διανομή των πόρων. Τα συστήματα και η αποθήκευση μπορούν να υλοποιούνται μέσω μίας συγκεντρωτικής υποδομής, τα κόστη αξιολογούνται με συγκεκριμένες μετρικές, επιτρέπεται η πολυ-μίσθωση (δηλ. η μίσθωση του ίδιου εξοπλισμού σε πολλαπλούς χρήστες του ταυτόχρονα), και οι πόροι είναι επεκτάσιμοι με ευελιξία.

Η υπολογιστική κοινή ωφέλεια (utility computing)<sup>1</sup>, είναι ένα όνειρο που χρονολογείται από την αρχή της ίδιας της υπολογιστικής βιομηχανίας. Αξίζει να αναφερθεί πως η υπολογιστική νέφος, πέρα από την παροχή για αποδοτικότερων και πιο προσιτών υπολογισμών, έχει επιτρέψει και την ανάπτυξη και χρήση εξειδικευμένων συστημάτων.

## 1.2 Τύποι υπολογιστικών νεφών

Σε σχέση με τους τύπους των υπολογιστικών νεφών εξετάζονται συνήθως δύο διακριτά σύνολα μοντέλων:

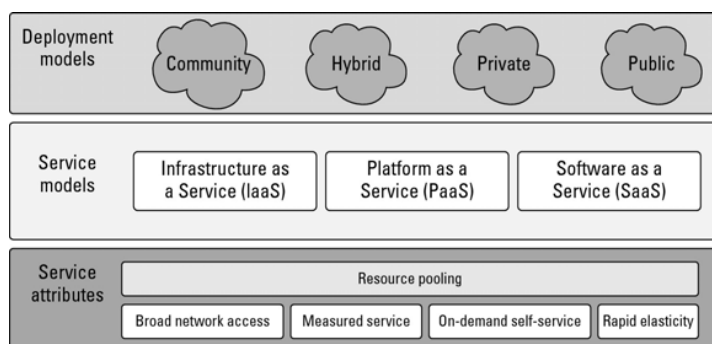
- **Μοντέλα διάθεσης (deployment models):** Τα μοντέλα αυτά αναφέρονται στη θέση και στη διαχείριση της υποδομής του νέφους.
- **Μοντέλα υπηρεσιών (service models):** Τα μοντέλα αυτά αναφέρονται στους ιδιαίτερους τύπους υπηρεσιών που μπορεί κάποιος να έχει πρόσβαση σε μια πλατφόρμα υπολογιστικού νέφους.

### 1.2.1 Μοντέλο NIST

Το Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας των Ηνωμένων Πολιτειών (NIST), διαθέτει ένα σύνολο από λειτουργικούς ορισμούς (<http://www.nist.gov/itl/cloud/>) που διαχωρίζουν το υπολογιστικό νέφος στα μοντέλα υπηρεσιών και στα μοντέλα διάθεσης.

Βάσει του μοντέλου του NIST, αρχικά δεν απαιτήθηκε για ένα νέφος να χρησιμοποιεί εικονικοποίηση για να αντλεί πόρους, ούτε και υποστηρίζει μοντέλα πολλαπλής μίσθωσης (υπενθυμίζεται ότι η πολλαπλή μίσθωση αφορά στη διανομή των πόρων μεταξύ δύο ή περισσότερων πελατών). Ο ανανεωμένος ορισμός του NIST θέτει ως απαίτηση για τα δίκτυα υπολογιστικού νέφους να χρησιμοποιούν εικονικοποίηση και να υποστηρίζουν το μοντέλο πολλαπλής μίσθωσης.

Το μοντέλο του NIST, περιλαμβάνοντας τις διάφορες κατατάξεις που ακολουθεί και τη σχέση τους με τα ιδιαίτερα χαρακτηριστικά του υπολογιστικού νέφους, παρουσιάζεται στην Εικόνα 1.



**Εικόνα 1 - Οι ορισμοί υπολογιστικού νέφους του NIST**

<sup>1</sup> Ως υπολογιστική κοινή ωφέλεια αναφέρεται το μοντέλο όπου υπάρχει απεριόριστη ποσότητα υπολογιστικών πόρων (επεξεργασία, αποθήκευση κ.λπ.) και ο κάθε ενδιαφερόμενος χρησιμοποιεί όσο θέλει, πληρώνοντας για την κατανάλωση. Ο όρος είναι δανεισμένος από τα δίκτυα κοινής ωφέλειας, όπως το δίκτυο ύδρευσης και το δίκτυο ηλεκτρισμού.

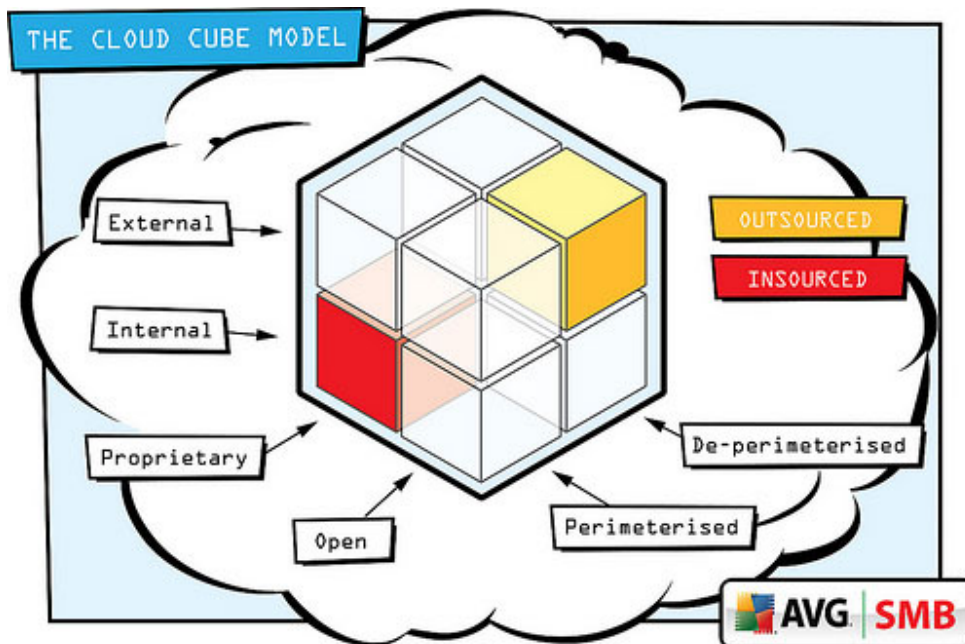
Λόγω του ότι το υπολογιστικό νέφος περιλαμβάνει ένα σύνολο συστατικών που αλληλεπιδρούν και τα οποία είναι βασισμένα σε πρότυπα όπως η Προσανατολισμένη στις Υπηρεσίες Αρχιτεκτονική (Service Oriented Architecture), είναι αναμενόμενο ότι οι μελλοντικές εκδόσεις του NIST ενδέχεται να συμπεριλάβουν πρόσθετα χαρακτηριστικά που αφορούν τα συστατικά και τις μεταξύ τους αλληλεπιδράσεις. Ακόμη το μοντέλο νέφους NIST δεν καλύπτει ενδιάμεσες υπηρεσίες όπως π.χ. οι μεσίτες δοσοληψιών ή υπηρεσιών.

### 1.2.2 Μοντέλο κύβου για το νέφος

Στα πλαίσια του Open Group λειτουργεί μια ένωση αποκαλούμενη *φόρουμ του Jericho* (<https://collaboration.opengroup.org/jericho/>) που κυρίως εστιάζει στην ασφάλεια και την ασφαλή συνεργασία στα πλαίσια του νέφους. Η ομάδα έχει αναπτύξει ένα μοντέλο που προσπαθεί να κατηγοριοποιήσει ένα δίκτυο νέφους, βασισμένο σε τέσσερις παράγοντες. Όπως περιγράφεται στην εργασία με τίτλο "Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration" [2], ο κάθε τύπος νέφους αλλάζει κατά πολύ το πού τοποθετείται το όριο μεταξύ του δικτύου του πελάτη και του νέφους.

Οι τέσσερις διαστάσεις του μοντέλου νέφους κύβου παρουσιάζονται στην παρακάτω εικόνα και απαριθμούνται ακολούθως:

- **Φυσική θέση των δεδομένων (Physical location of the data):** Εσωτερική (Internal: I, δηλαδή εντός του οργανισμού) / Εξωτερική (External: E, δηλαδή οπουδήποτε στο διαδίκτυο).
- **Ιδιοκτησία (Ownership):** Το Ιδιόκτητο (Proprietary: P, δηλαδή η ιδιοκτησία των υποδομών ή/και των υπηρεσιών ή/και των διεπαφών κ.τ.λ. ανήκει στον οργανισμό) / Ανοικτό (Open: O, δηλαδή η ιδιοκτησία των ανωτέρω στοιχείων ανήκει σε εξωτερικό πάροχο). Ο τύπος ιδιοκτησίας είναι ένα μέτρο όχι μόνο της ιδιοκτησίας της τεχνολογίας, αλλά της διαλειτουργικότητας, της ευκολίας της μεταφοράς στοιχείων, και του βαθμού που "κλειδώνει" η εφαρμογή των προμηθευτών.
- **Όριο ασφάλειας (Security boundary):** Η Περιμετροποίηση (Perimeterised: Per) / αποπεριμετροποίηση (De-perimeterised: D-p) είναι ένα μέτρο σχετικά με το εάν η λειτουργία είναι μέσα ή έξω από τα όρια ασφάλειας του τοίχου προστασίας. Ένα περιμετροποιημένο νέφος τυπικά βασίζεται στην έννοια του παραδοσιακού τείχους προστασίας (firewall), ενώ ένα αποπεριμετροποιημένο νέφος βασίζεται σε άλλες προσεγγίσεις για την επίτευξη της ασφάλειας (όπως είναι για παράδειγμα στις *εντολές του φόρουμ Jericho* - [https://collaboration.opengroup.org/jericho/commandments\\_v1.2.pdf](https://collaboration.opengroup.org/jericho/commandments_v1.2.pdf)), οι οποίες επί παραδείγματι περιλαμβάνουν την συνύφανση δεδομένων με μετα-δεδομένα και μηχανισμούς που να αποτρέπουν τη μη παραδεκτή και μη εξουσιοδοτημένη χρήση τους.
- **Προμήθεια (Sourcing):** Η Υλοποίηση με ίδιους πόρους (Insourced) ή η προμήθεια από εξωτερικό πάροχο (Outsourced), αναφέρεται στο πότε η υπηρεσία παρέχεται από τον πελάτη ή το φορέα παροχής υπηρεσιών.



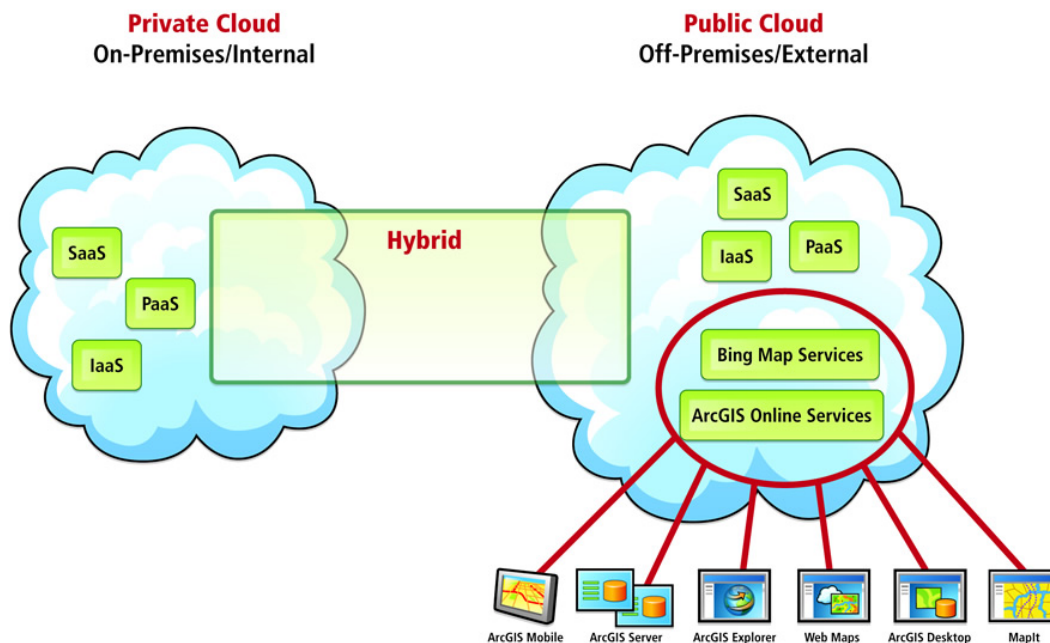
Εικόνα 2 - Το μοντέλο νέφους κύβου

Συνολικά, η τέταρτη διάσταση αντιστοιχεί σε δύο διαφορετικές καταστάσεις στις οκτώ πιθανές μορφές νέφους: Per (IP, IO, EP, EO) και D-p (IP, IO, EP, EO).

### 1.2.3 Μοντέλα Ανάπτυξης

Ένα μοντέλο διάθεσης καθορίζει το σκοπό του νέφους και τη φύση της τοποθεσίας αυτού. Οι ορισμοί του NIST για τα τέσσερα μοντέλα διάθεσης (κατά μία έννοια αντιστοιχεί στη διάσταση της ιδιοκτησίας στο μοντέλο κύβου) είναι οι ακόλουθοι:

- **Δημόσιο νέφος (Public cloud):** Η υποδομή δημοσίου νέφους είναι διαθέσιμη για δημόσια χρήση, και ανήκει σε έναν οργανισμό που πωλεί υπηρεσίες νέφους.
- **Ιδιωτικό νέφος (Private cloud):** Η υποδομή ιδιωτικού νέφους χρησιμοποιείται για την αποκλειστική χρήση από έναν οργανισμό. Το νέφος μπορεί να το διαχειρίζεται είτε ο ίδιος ο οργανισμός είτε κάποιος τρίτος.
- **Νέφος Κοινότητας (Community cloud):** Σε αυτό το μοντέλο η δομή του νέφους έχει οργανωθεί για να εξυπηρετεί μια κοινή λειτουργία ή ένα κοινό σκοπό. Μπορεί να χρησιμοποιείται από έναν οργανισμό ή από πολλούς οργανισμούς που ασχολούνται με κοινά ζητήματα, τα οποία μπορεί να περιλαμβάνουν την αποστολή, την πολιτική, την ασφάλεια, τις ανάγκες συμμόρφωσης σε κανονισμούς κτλ. Το νέφος κοινότητας μπορεί να το διαχειρίζονται οι οργανισμοί που το χρησιμοποιούν ή ένας τρίτος.
- **Υβριδικό νέφος (Hybrid cloud):** Η δομή ενός υβριδικού νέφους συνδυάζει πολλά μοντέλα νεφών (ιδιωτικό, κοινοτικό ή δημόσιο). Τα νέφη αυτά διατηρούν τα ιδιαίτερα χαρακτηριστικά τους, αλλά είναι συνδεδεμένα μαζί σαν μια μονάδα.



Εικόνα 3 - Θέσεις επέκτασης για διαφορετικούς τύπους νεφών

### 1.2.4 Μοντέλα Υπηρεσιών

Στο μοντέλο διάθεσης, οι διαφορετικοί τύποι νεφών αφορούν την τοποθεσία και την ιδιοκτησία των στοιχείων του νέφους. Στη συνέχεια, όταν το υπολογιστικό νέφος έχει αναπτυχθεί, διαφορετικοί προμηθευτές προσφέρουν νέφη όπου διαθέτουν διαφορετικές υπηρεσίες. Το χαρτοφυλάκιο των προσφερόμενων υπηρεσιών προσθέτει άλλο ένα σύνολο ορισμών, το αποκαλούμενο μοντέλο υπηρεσιών. Οι τρεις τύποι υπηρεσιών οι οποίοι έχουν γίνει αποδεκτοί παγκοσμίως είναι οι εξής [1]:

- Λογισμικό ως υπηρεσία - ΛωΥ (Software as a Service - SaaS):** Το μοντέλο ΛωΥ είναι ένα πλήρες λειτουργικό περιβάλλον με εφαρμογές λογισμικού που παρέχονται στο χρήστη, χωρίς ο ίδιος να χρειάζεται να αναβαθμίσει η να εγκαταστήσει κάποια εφαρμογή. Η ευθύνη του χρήστη αρχίζει και τελειώνει με την εισαγωγή και τη διαχείριση των δεδομένων του. Σε αντιδιαστολή, ο πάροχος της εφαρμογής είναι υπεύθυνος για ένα ευρύ κατάλογο θεμάτων που ξεκινά από την ίδια την εφαρμογή και φτάνει μέχρι και την υποδομή της υπηρεσίας. Σε αυτό το είδος υπηρεσίας, η εφαρμογή παρέχεται στον χρήστη μέσω μιας διεπαφής η μιας εφαρμογής πλοήγησης.
- Πλατφόρμα ως υπηρεσία - ΠωΥ (Platform as a Service - PaaS):** Το μοντέλο ΠωΥ παρέχει στους χρήστες λειτουργικά συστήματα, υπηρεσίες, πλαίσια ανάπτυξης, εφαρμογές, εικονικές μηχανές και δομές έλεγχου. Ο χρήστης σε αυτό το είδος υπηρεσίας έχει τη δυνατότητα να αναπτύξει εφαρμογές χρησιμοποιώντας γλώσσες και εργαλεία προγραμματισμού που υποστηρίζονται από τον πάροχο, ή ακόμα και να κάνει επέκταση των εφαρμογών του στην υποδομή νέφους. Ο χρήστης δηλαδή είναι υπεύθυνος για την εγκατάσταση και τη διαχείριση των εφαρμογών που αναπτύσσει στο εκάστοτε περιβάλλον, ενώ ο πάροχος υπηρεσιών διαχειρίζεται την υποδομή

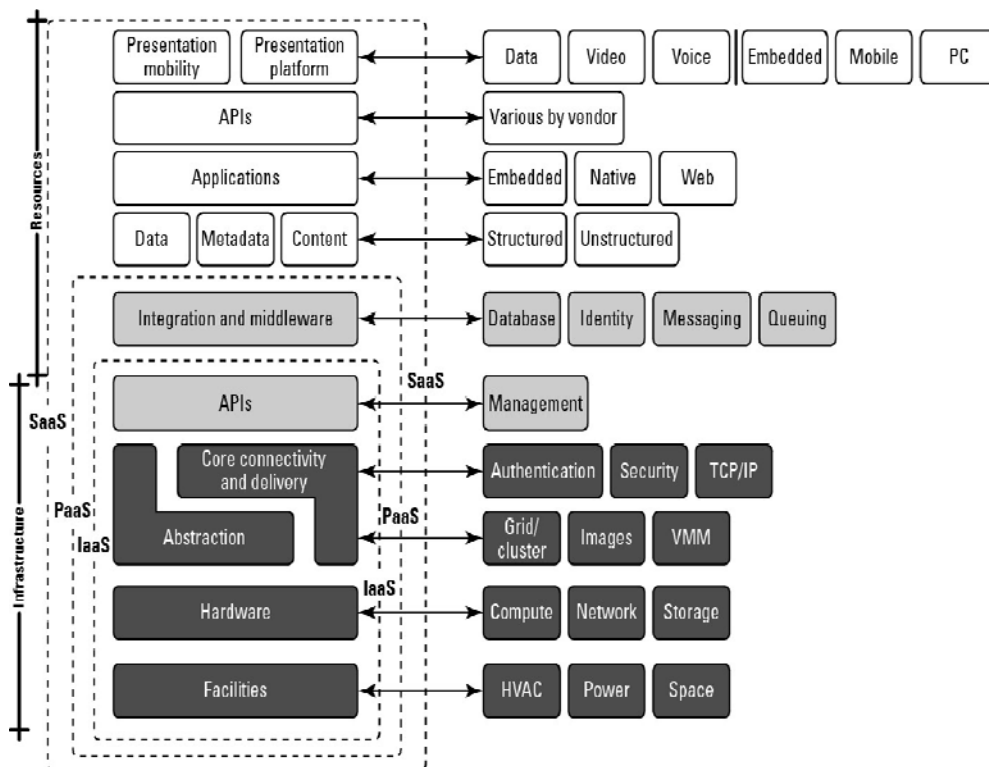
νέφους όπου ο χρήστης φιλοξένει τις δίκες του εφαρμογές, τα λειτουργικά συστήματα, τα πλαίσια ανάπτυξης και φυσικά την υπολογιστική υποδομή.

- **Υποδομή ως υπηρεσία - ΥωΥ (Infrastructure as a Service - IaaS):** Το μοντέλο ΥωΥ είναι ένα είδος υπηρεσίας που παρέχει στους χρήστες εικονικές μηχανές, εικονική αποθήκευση, εικονική υποδομή, καθώς και άλλους υπολογιστικούς πόρους, τους οποίους ο κάθε χρήστης μπορεί να εκμεταλλευθεί κατά το δοκούν. Ο φορέας παροχής υπηρεσιών ΥωΥ διαχειρίζεται όλη την υποδομή, ενώ ο χρήστης είναι αρμόδιος για το λειτουργικό σύστημα και τις εφαρμογές.

Τα τρία διαφορετικά μοντέλα υπηρεσιών είναι γνωστά ως το μοντέλο ΛΠΥ (ΛωΥ – ΠωΥ, ΥωΥ) [SPI (SaaS - PaaS - IaaS)] του υπολογιστικού νέφους. Πολλά άλλα μοντέλα υπηρεσιών έχουν αναφερθεί: ΑωΥ (αποθήκευση ως υπηρεσία - StaaS - Storage as a Service), ΤωΥ (ταυτότητα ως υπηρεσία - IaaS - Identity as a Service), ΣωΥ (συμμόρφωση ως υπηρεσία - CaaS - Compliance as a Service) και ούτω καθ' εξής. Εντούτοις, οι υπηρεσίες SPI καλύπτουν όλες τις άλλες δυνατότητες.

Είναι χρήσιμο να λογίζονται τα μοντέλα υπηρεσιών υπολογιστικού νέφους σε όρους στοιβάς υλικού/λογισμικού. Μια τέτοια αναπαράσταση αποκαλείται *μοντέλο αναφοράς νέφους* (Cloud Reference Model) και παρουσιάζεται στην Εικόνα 4. Στο κατώτατο σημείο της στοιβάς είναι το υλικό ή η υποδομή που περιλαμβάνει το δίκτυο. Καθώς κινούμαστε προς τα πάνω στη στοιβά, κάθε μοντέλο υπηρεσιών κληρονομεί τις ιδιότητες του μοντέλου υπηρεσιών που βρίσκεται από κάτω του. Το μοντέλο ΥωΥ έχει τα μικρότερα επίπεδα ενσωματωμένης λειτουργίας και το χαμηλότερο επίπεδο ολοκλήρωσης, ενώ το ΛωΥ έχει το πιο αυξημένο επίπεδο ολοκλήρωσης.





Εικόνα 4 - Μοντέλο αναφοράς νέφους

### 1.3 Παραδείγματα υπολογιστικού νέφους

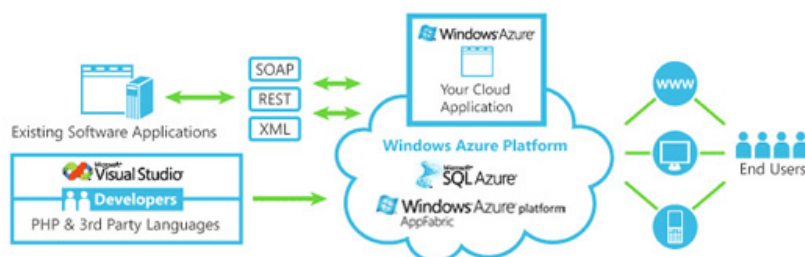
Για να γίνει περισσότερο κατανοητός ο τρόπος με τον οποίο η υπολογιστική νέφους έχει αλλάξει τη φύση της ανάπτυξης των εμπορικών συστημάτων, εξετάζονται τα ακόλουθα τρία παραδείγματα, τα οποία θα αναλυθούν και στο κεφάλαιο των πλατφορμών του υπολογιστικού νέφους:

**Google AppEngine:** Ο χρήστης αρχικά δημιουργεί ένα λογαριασμό, κι εν συνεχεία έχει πρόσβαση σε πληθώρα υπηρεσιών όπως είναι το ηλεκτρονικό ταχυδρομείο, χάρτες, επεξεργασία και αποθήκευση εικόνων, ατζέντα, σουίτα επεξεργασίας εγγράφων, καθώς και προσωπικό χώρο για προβολή στο διαδίκτυο. Όλα αυτά είναι προσβάσιμα μέσω μιας σύνδεσης στο Internet. Αυτό είναι ένα κλασικό παράδειγμα του μοντέλου *λογισμικό ως υπηρεσία*.

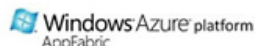


**Εικόνα 5 - Google Apps**

**Microsoft Azure:** Η συγκεκριμένη πλατφόρμα παρέχει επεξεργαστική ισχύ, αποθήκευση δεδομένων και διάφορα εργαλεία ανάπτυξης εφαρμογών, όπου όμως υπάρχει η υποστήριξη και συμβατότητα μεταξύ διαφόρων προϊόντων και τεχνολογιών της Microsoft.

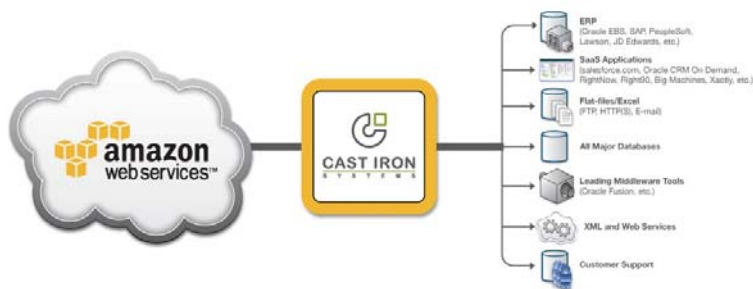


#### The Platform Products



**Εικόνα 6 - Microsoft Windows Azure**

**Amazon Web Services:** Μια από τις επιτυχημένες επιχειρήσεις βασισμένες σε νέφος, είναι οι υπηρεσίες Ιστού του Amazon, οι οποίες είναι της μορφής ΥωΥ και προσφέρουν την ενοικίαση εικονικών υπολογιστών στην υποδομή της Amazon.



**Εικόνα 7 - Amazon Web Services**

Αυτές οι νέες δυνατότητες επιτρέπουν στις εφαρμογές να προγραμματιστούν μειωμένη δαπάνη, να επεκταθούν σε σύντομο χρόνο και να γίνουν διαθέσιμες στο διαδίκτυο με κάποιο μοντέλο άδειας χρήσης.

#### 1.4 Πλεονεκτήματα υπολογιστικού νέφους

Οι Peter Mell και Tim Grance [4] ταξινομούν το υπολογιστικό νέφος στα τρία μοντέλα SPI υπηρεσιών (SaaS, PaaS και IaaS) και σε τέσσερις τύπους νεφών (public, private, community και hybrid), ενώ επίσης ορίζουν πέντε ουσιαστικά χαρακτηριστικά που τα συστήματα υπολογιστικών νεφών πρέπει να προσφέρουν:

- **Αυτοεξυπηρέτηση κατόπιν απαίτησης (On-demand self-service):** Ένας καταναλωτής μπορεί να αξιοποιήσει τους πόρους των υπολογιστών όταν και όποτε τους χρειάζεται χωρίς να είναι αναγκαίο να αλληλεπιδράσει με τον πάροχο της κάθε υπηρεσίας νέφους μεμονωμένα.
- **Ευρεία πρόσβαση στο δίκτυο (Broad network access):** Ένας χρήστης μπορεί να έχει πρόσβαση στους υπολογιστικούς πόρους του νέφους, που είναι διαθέσιμοι στο δίκτυο, μέσω τυποποιημένων μεθόδων. Οι μέθοδοι αυτοί προωθούν τη χρήση ετερογενών λειτουργικών συστημάτων από την πλευρά του χρήστη όπως τα laptop, tablets, smartphones, Macs και PDA.
- **Μετρήσιμη ποσότητα υπηρεσίας (Measured service):** Η χρήση των πόρων υπολογιστικού νέφους μετριέται ποσοτικά, ελέγχεται, και αναφέρεται στον χρήστη, βάση ενός συστήματος μέτρησης. Ένας χρήστης χρεώνεται βάση του επίπεδου των υπηρεσιών που του παρέχονται και μέσω μιας γνωστής μετρικής, όπως π.χ. η χρησιμοποιούμενη ποσότητα αποθήκευσης, ο αριθμός των δοσοληψιών, η είσοδος/έξοδος, το εύρος ζώνης, η χρησιμοποιούμενη ποσότητα επεξεργαστικής ισχύος και ούτω καθεξής.
- **Ταχεία ελαστικότητα (Rapid elasticity):** Οι δυνατότητες του νέφους μπορούν να παρέχονται στους αιτούμενους και να απελευθερώνονται από αυτούς με ελαστικό τρόπο, σε αρκετές περιπτώσεις με αυτόματο τρόπο. Από την πλευρά του αιτουμένου, οι πόροι υπολογιστικού νέφους πρέπει να φαίνονται απεριόριστοι και να μπορούν να αποκτηθούν οποιαδήποτε στιγμή και σε οποιοδήποτε ποσότητα.
- **Συγκέντρωση πόρων με το μοντέλο δεξαμενής (Resource pooling):** Ένας πάροχος υπηρεσιών νέφους δημιουργεί πόρους που συγκεντρώνονται μαζί σε ένα σύστημα που υποστηρίζει το μοντέλο “πολλαπλής μίσθωσης”, δηλαδή την ταυτόχρονη εξυπηρέτηση πολλών καταναλωτών. Επίσης φυσικά και εικονικά συστήματα διανέμονται η αναδιανέμονται δυναμικά όπως απαιτείται από τις εκάστοτε ανάγκες του καταναλωτή, ο οποίος όμως δεν έχει γνώση της ακριβούς θέσης των πόρων αυτών. Τέτοιοι πόροι είναι οι εικονικές μηχανές, η μνήμη και ο αποθηκευτικός χώρος.

Ενώ αυτά τα πέντε χαρακτηριστικά γνωρίσματα του υπολογιστικού νέφους είναι στην ουσία γενικώς αποδεκτά, πρέπει επίσης να εξεταστούν τα ακόλουθα πρόσθετα πλεονεκτήματα:

- **Χαμηλότερο κόστος (Lower costs):** Επειδή τα δίκτυα νέφους λειτουργούν με υψηλότερη αποδοτικότητα και με μεγαλύτερο βαθμό αξιοποίησης, είναι σύνηθες το φαινόμενο της σημαντικής μείωσης των δαπανών.
- **Αξιοπιστία (Reliability):** Η κλίμακα των δικτύων υπολογιστικού νέφους και η δυνατότητα τους να υλοποιούν εξισορρόπηση φόρτου και εφεδρική λειτουργία τα καθιστά ιδιαίτερα αξιόπιστα, συχνά πιο αξιόπιστα από αυτό που μπορεί να επιτευχθεί σε έναν απλό οργανισμό.
- **Ποιότητα υπηρεσίας (Quality of Service):** Η ποιότητα της υπηρεσίας (QoS) είναι κάτι το οποίο μπορεί να συμφωνηθεί στα πλαίσια της σύμβασης με τον προμηθευτή.
- **Εξωτερική ανάθεση της διαχείρισης τεχνολογιών πληροφοριών (Outsourced IT management):** Η προσέγγιση του υπολογιστικού νέφους επιτρέπει στους πελάτες να αναθέτουν την διαχείριση των υπολογιστικών πόρων σε τρίτους, εντός ή εκτός της επιχείρησης. Με αυτό τον τρόπο επιτυγχάνεται σημαντική μείωση των δαπανών στελέχωσης σε προσωπικό πληροφορικής.
- **Ευκολία χρησιμοποίησης (Ease of utilization):** Ανάλογα με τον τύπο υπηρεσίας που προσφέρεται, μπορεί να μην απαιτούνται οι άδειες χρήσης υλικού ή λογισμικού για τη χρήση των υπηρεσιών.
- **Απλουστευμένη συντήρηση και αναβάθμιση (Simplified maintenance and upgrade):** Στο υπολογιστικό νέφος, μπορούν να εφαρμόζονται αναβαθμίσεις λογισμικού καθώς και να δημιουργούνται αντίγραφα ασφάλειας. Αυτό σημαίνει πως οι χρήστες έχουν πάντα πρόσβαση στις πιο πρόσφατες εκδόσεις λογισμικού.
- **Χαμηλό κόστος αρχικής επένδυσης (Low Barrier to Entry):** Ειδικότερα, οι αρχικές ανάγκες επένδυσης κεφαλαίου μειώνονται σημαντικά.

### 1.5 Μειονεκτήματα υπολογιστικού νέφους

Παρ' όλο που τα οφέλη του υπολογιστικού νέφους όπως προαναφέρθηκαν είναι σημαντικά, υπάρχουν και σημαντικά μειονεκτήματα τα οποία περιγράφονται ακολούθως:

- Το κυριότερο ίσως μειονέκτημα του υπολογιστικού νέφους είναι αναμφισβήτητα η ιδιωτικότητα και η ασφάλεια. Αυτό συμβαίνει ότι τα δεδομένα του οργανισμού μεταφέρονται σε συστήματα που δεν είναι πλέον υπό τον έλεγχό του. Έτσι διατρέχεται επιπρόσθετος κίνδυνος διάπραξης αδικήματος από άλλους. Δεν μπορούμε επομένως να βασιστούμε απόλυτα σε έναν πάροχο νέφους να διατηρήσει την ιδιωτικότητα των δεδομένων ενός οργανισμού, μας παρά την οποιαδήποτε αυστηρότητα του νομοθετικού και κανονιστικού πλαισίου.
- Κατά γενικό κανόνα, τα πλεονεκτήματα του υπολογιστικού νέφους παρουσιάζονται ως επιτακτική ανάγκη, περισσότερο για τους μικρούς οργανισμούς σε σχέση με τους πιο μεγάλους. Οι μεγάλοι οργανισμοί μπορούν

να υποστηρίξουν προσωπικό πληροφορικής και ενέργειες σχεδίασης κατάλληλου λογισμικού για τις ιδιαίτερες ανάγκες τους.

- Όταν χρησιμοποιείται μια εφαρμογή ή μια υπηρεσία στο νέφος, χρησιμοποιείται κάτι που δεν είναι απαραίτητως τόσο εξατομικευμένο (τουλάχιστον στο βαθμό που ενδεχομένως κάποιος επιθυμεί). Επιπλέον, αν και πολλές εφαρμογές υπολογιστικού νέφους είναι πολύ εξελιγμένες, οι «παραδοσιακές» εφαρμογές που αναπτύσσονται από διάφορες εταιρείες έχουν πολλές λειτουργίες που δεν περιλαμβάνονται σε εκείνες που είναι διαθέσιμες εντός του υπολογιστικού νέφους.
- Το υπολογιστικό νέφος είναι ένα άναρχο σύστημα, όπως είναι το Διαδίκτυο σε γενικές γραμμές. Όλα τα αιτήματα που υποστηρίζονται από το πρωτόκολλο HTTP (PUT, GET, κ.ά.) είναι αιτήματα σε έναν πάροχο υπηρεσιών, ο οποίος τα επεξεργάζεται και στη συνέχεια στέλνει μια απάντηση. Κατόπιν ο εξυπηρετούμενος μπορεί να αποστείλει νέο αίτημα και να λάβει νέα απάντηση κ.ο.κ. Αν και μπορεί να μοιάζει ότι συνεχίζεται μια συνομιλία μεταξύ του πελάτη και του παρόχου, στην ουσία υπάρχει μια αρχιτεκτονική αποσύνδεση μεταξύ διαφορετικών αιτημάτων, η οποία εισάγει μία *έλλειψη κατάστασης*, που έχει με τη σειρά της ως αποτέλεσμα τα μηνύματα να ταξιδεύουν σε διαφορετικές διαδρομές και τα δεδομένα να φθάνουν χωρίς αλληλουχία. Επομένως, για να επιβληθεί η συνοχή των δοσοληψιών στο σύστημα, χρειάζεται να προστεθούν ορισμένες τεχνικές, κυρίως με τη μορφή μεσιτών υπηρεσιών, διαχειριστών δοσοληψιών κ.λπ. Αυτό στη συνέχεια μπορεί να επηρεάσει κατά πολύ την απόδοση ορισμένων εφαρμογών.
- Στις μέρες μας, οι περισσότεροι οργανισμοί βρίσκονται αντιμέτωποι με νομοθετικά και κανονιστικά ζητήματα που αφορούν τη λειτουργία τους. Στην Ευρώπη, η Ευρωπαϊκή Οικονομική Κοινότητα (Ε.Ο.Κ.) έχει μια σειρά από δικές της νομοθεσίες σχετικά με τις επιχειρήσεις. Η επέκταση ενός υπολογιστικού νέφους μεταξύ διαφόρων χωρών μπορεί να έχει το μειονέκτημα της συμμόρφωσης με μια σειρά διαφορετικών νόμων (ανάλογα με το τι ισχύει στη κάθε χώρα) και κατά συνέπεια τα ένδικα μέσα (π.χ. η προσφυγή κατά του παρόχου νέφους για ένα θέμα παραβίασης ιδιωτικότητας) δεν είναι απόλυτα σαφές με ποιά νομοθεσία θα κριθεί. Οι νόμοι και οι ρυθμιστικές διατάξεις των περισσότερων χωρών τοποθετούν ολόκληρο το φορτίο ευθύνης επάνω στον πελάτη.

## Κεφάλαιο 2: Η αρχιτεκτονική του υπολογιστικού νέφους

Όσον αφορά τα συστήματα διαχείρισης πληροφοριών, αυτά επιβάλλεται να παρακολουθούν τις πληροφορίες που είναι ζωτικής σημασίας γενικά για την ομαλή λειτουργία μιας επιχείρησης. Ωστόσο, τα ίδια αυτά συστήματα διαχειρίζονται επίσης τις επιχειρησιακές διαδικασίες μιας επιχείρησης. Το πρώτο βήμα είναι η αναγνώριση και η ονομασία κάθε διαδικασίας, αλλά κι ο προσδιορισμός των επιχειρησιακών γεγονότων που χαρακτηρίζουν την έναρξη και το τέλος της. Η βαθύτερη ανάλυση περιλαμβάνει την αναγνώριση των ενδιάμεσων επιχειρησιακών γεγονότων και των ανθρώπινων αποφάσεων που λαμβάνονται κατά τη διάρκεια της διαδικασίας.

Οι επιχειρηματικές διαδικασίες μπορούν συχνά να διαχωρίζονται σε κατακόρυφες ή οριζόντιες. Οι κατακόρυφες διαδικασίες λειτουργούν μέσα σε μια ενιαία οργανωτική μονάδα (όπως οι πωλήσεις) και διαχειρίζονται ένα συνεκτικό σύνολο πληροφοριών σχετικά με εκείνη τη μονάδα, ενώ επίσης υποστηρίζονται από πακέτα λογισμικού ή συστημάτων. Από την άλλη πλευρά, οι οριζόντιες διαδικασίες τέμνουν τις οργανωτικές μονάδες [5].

Στη συνέχεια παρουσιάζονται εισαγωγικές έννοιες που αφορούν την αρχιτεκτονική του υπολογιστικού νέφους, όπως η αρχιτεκτονική στοίβα της υπολογιστικής νέφους (cloud stack), η συνθεσιμότητα, καθώς και η υποδομή. Έπειτα αναφέρονται οι 2 κυριότερες αρχιτεκτονικές του υπολογιστικού νέφους και γίνεται πιο περιγραφική η ανάλυσή τους μέσω των συνοδευτικών εικόνων.

### 2.1 Εξερεύνηση της αρχιτεκτονικής στοίβας της υπολογιστικής νέφους

Ξεκινώντας την εξερεύνηση της στοίβας των επιπέδων που απαντώνται στην υπολογιστική νέφους, χρήσιμο είναι αρχικά να σημειωθεί πως η υπολογιστική νέφους (cloud computing), στηρίζεται στην αρχιτεκτονική για την ανάπτυξη εφαρμογών ευρείας κλίμακας στο Διαδίκτυο κατά τη διάρκεια των τελευταίων 20 χρόνων. Σε αυτά τα τυποποιημένα πρωτόκολλα δικτύωσης, η υπολογιστική νέφους προσθέτει τις προόδους στην όψη και στην εικόνα των συστημάτων που έγινε διαθέσιμη την τελευταία δεκαετία. Η υπολογιστική νέφους δημιουργεί ένα σύστημα όπου οι πόροι μπορούν να συγκεντρωθούν και να διαχωριστούν όπως απαιτείται.

Η αρχιτεκτονική νέφους διαθέτει τη δυνατότητα του να μπορεί να συνδέσει το λογισμικό που εκτελείται σε υλικό πολλαπλών τοποθεσιών, για την παροχή μιας υπηρεσίας κατόπιν σχετικού αιτήματος. Αυτός ο συνδυασμός αφαίρεσης και μετρώμενης υπηρεσίας διαχωρίζει τις αρχιτεκτονικές απαιτήσεις των συστημάτων υπολογιστικής νέφους από τη γενική περιγραφή που δίνεται για μια εφαρμογή Διαδικτύου.

Πολλές είναι οι περιγραφές εκείνες οι οποίες περιγράφουν την υπολογιστική νέφους με όρους δύο αρχιτεκτονικών στρωμάτων:

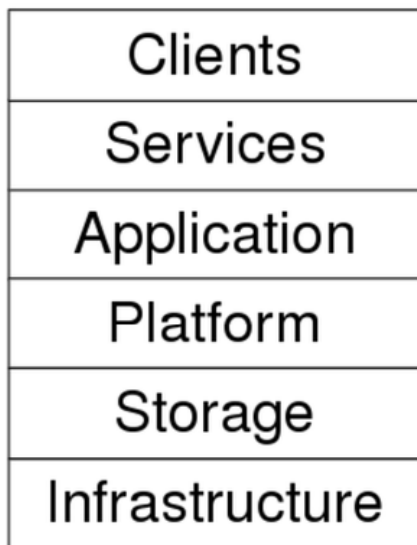
1. Ένας εξυπηρετούμενος ως εμπρόσθιο μέρος.
2. Το “νέφος” ως οπίσθιο μέρος.

Αυτή είναι μια εξαιρετικά απλοϊκή περιγραφή, δεδομένου ότι κάθε ένα από τα δύο στοιχεία αποτελούνται από διάφορα στρώματα συστατικών, συμπληρωματικές λειτουργίες και από ένα συνδυασμό από τυποποιημένα καθώς και ιδιόκτητα (proprietary) πρωτόκολλα. Η υπολογιστική νέφος μπορεί να διαφοροποιηθεί από τα παλαιότερα μοντέλα με την περιγραφή μιας ενθυλακωμένης υπηρεσίας τεχνολογίας πληροφοριών, που ελέγχεται συχνά μέσω μιας καθορισμένης προγραμματιστικής διασύνδεσης εφαρμογών (Application Programming Interface, API), ορίζοντας έτσι τις υπηρεσίες που παραδίδονται προς το δίκτυο.

Ένα νέφος μπορεί να δημιουργηθεί μέσα στην υποδομή του ίδιου του οργανισμού ή να μεταφερθεί σ' ένα άλλο κέντρο δεδομένων (data center). Έτσι, ενώ οι πόροι σ' ένα νέφος μπορούν να είναι πραγματικοί φυσικοί πόροι συχνότερα είναι εικονικοί πόροι, διότι οι εικονικοί πόροι είναι πιο εύκολο να τροποποιηθούν (π.χ. να αντικατασταθούν ή να αναβαθμιστούν) και να βελτιστοποιηθούν. Ένα υπολογιστικό νέφος απαιτεί εικονική αποθήκευση για να υποστηρίξει την οργάνωση και την αποθήκευση των δεδομένων. Από την πλευρά ενός χρήστη, είναι σημαντικό οι πόροι να εμφανίζονται ως απεριόριστου μεγέθους, η υπηρεσία να είναι μετρήσιμη και η τιμολόγηση να μπορεί να μετρηθεί [6].

Προκειμένου να είναι εφικτή η επίτευξη των στόχων της υπολογιστικής νέφους, η αρχιτεκτονική του νέφους διαμορφώνεται σε μία στοίβα που περιλαμβάνει έξι επίπεδα. Τα επίπεδα αυτά, σε καθοδική παράθεση, έχουν ως εξής:

- Οι εξυπηρετούμενοι, που αποτελούνται από το υλικό και λογισμικό του υπολογιστή του χρήστη, και οι οποίοι προορίζονται αποκλειστικά για την πρόσβαση στις υπηρεσίες του νέφους.
- Οι υπηρεσίες, που είναι σχεδιασμένες με σκοπό την επικοινωνία δύο ή περισσότερων υπολογιστών μέσα σ' ένα δίκτυο και το οποίο με τη σειρά του μπορεί να προσπελαστεί από άλλα στοιχεία υπολογιστικής νέφους (όπως είναι άλλοι χρήστες ή λογισμικό).
- Η εφαρμογή, η οποία ευρισκόμενη εντός του νέφους εξαλείφει την ανάγκη για εγκατάσταση κι εκτέλεση μιας εφαρμογής στον εξυπηρετούμενο. Στο σημείο αυτό υποστηρίζεται η συντήρηση και η λειτουργία του εκάστοτε λογισμικού.
- Η πλατφόρμα, που παρέχει μια σχετική ευκολία στην ανάπτυξη των εφαρμογών, χωρίς το κόστος και την πολυπλοκότητα της αγοράς και διαχείρισης των επιπέδων του υλικού, αλλά και του λογισμικού.
- Ο αποθηκευτικός χώρος, που παρέχει υπηρεσίες αποθήκευσης, οι οποίες συχνά κοστίζουν ανάλογα με κάποια τιμολογιακή πολιτική (για παράδειγμα, gigabytes ανά μήνα).
- Η υποδομή, που σχετίζεται με την εικονικότητα μιας πλατφόρμας ως υπηρεσία.



*Εικόνα 8 - Επίπεδα της στοίβας υπολογισμού στην αρχιτεκτονική του νέφος*

## **2.2 Συνθεσιμότητα**

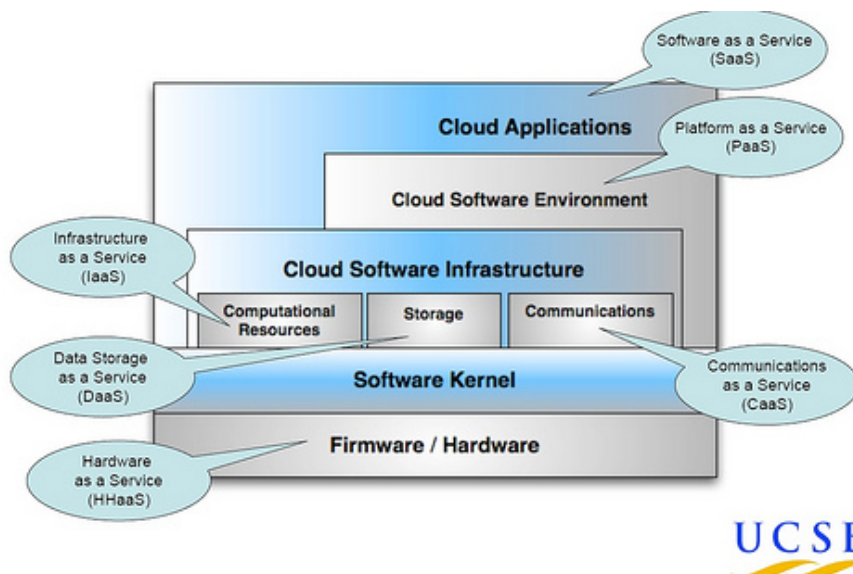
Οι εφαρμογές που δημιουργήθηκαν και δημιουργούνται στο νέφος είναι συχνά χτισμένες από μια συλλογή συστατικών, τα οποία αλληλεπιδρούν μεταξύ τους για να παρέχουν την ολοκληρωμένη λειτουργία. Αυτό το χαρακτηριστικό γνώρισμα καλείται *συνθεσιμότητα* (composability): ένα σύνθετο σύστημα χρησιμοποιεί συστατικά για να «συναρμολογήσει» υπηρεσίες που μπορούν να προσαρμοστούν για ένα συγκεκριμένο σκοπό που χρησιμοποιεί συγκεκριμένα μέρη.

Για να είναι εφικτή η συνθεσιμότητα, τα συστατικά είναι απαραίτητο να διαθέτουν τα ακόλουθα χαρακτηριστικά:

- Να είναι *τμηματικά* (modular), να συνιστούν δηλαδή ανεξάρτητες μονάδες που είναι συνεργατικές, επαναχρησιμοποιήσιμες και αντικαταστάσιμες.
- Να είναι *άνευ καταστάσεων* (stateless): Μια δοσοληψία εκτελείται αδιαφορώντας για άλλες δοσοληψίες ή αιτήματα.

Στην Εικόνα 9 παρουσιάζεται μία διαμόρφωση επιπέδων που αφορά την αλληλεξάρτηση και τη συνθεσιμότητα μεταξύ των διαφόρων επιπέδων στο νέφος [7].





UCSB

**Εικόνα 9 - Συνθεσιμότητα μεταξύ διαφόρων επιπέδων νέφους**

Ωστόσο, δεν είναι απόλυτη απαίτηση να είναι οι συναλλαγές άνευ καταστάσεων: κάποιες εφαρμογές υπολογισμού νέφους συμπεριλαμβάνουν μετάπτωση μεταξύ καταστάσεων, κάτι που κυρίως επιτυγχάνεται με την ενσωμάτωση οντοτήτων όπως *επόπτες δοσοληψιών* (transaction monitors). Σε σπανιότερες περιπτώσεις, πλήρη συστήματα δοσοληψιών αναπτύσσονται σε νέφη, τα οποία όμως είναι δυσκολότερο να σχεδιαστούν σε μια κατακεκολλημένη αρχιτεκτονική.

Αν κι ο υπολογισμός νεφών δεν απαιτεί να διαθέτει το υλικό και το λογισμικό το χαρακτηριστικό της συνθεσιμότητας, το χαρακτηριστικό αυτό είναι ιδιαίτερα επιθυμητό από τη μεριά του υπεύθυνου για την ανάπτυξη εφαρμογών. Αυτό συμβαίνει διότι καθιστά τη σχεδίαση του συστήματος ευκολότερη και τις λύσεις πιο φορητές και διαλειτουργικές.

Υπάρχει μια τάση να διαθέτουν τα συστήματα υπολογισμού νέφους πιο ελαττωμένο το χαρακτηριστικό της συνθεσιμότητας στο επίπεδο των χρηστών, καθώς οι υπηρεσίες ενσωματώνουν περισσότερα επίπεδα της στοίβας της υπολογιστικής νέφους. Από τη σκοπιά ενός ΥωΥ (υποδομή ως υπηρεσία, Infrastructure as a Service), όπως οι υπηρεσίες Διαδικτύου του Amazon (Amazon Web Services), το GoGrid ή το Rackspace, δεν υπάρχει κανένας λόγος να προσφέρει κάποιος μη-τυποποιημένα στιγμιότυπα μηχανών στους πελάτες, διότι οι πελάτες αυτοί είναι σχεδόν βέβαιο ότι αναπτύσσουν εφαρμογές που στηρίζονται σε τυποποιημένα λειτουργικά συστήματα, όπως τα Linux, Windows, Solaris κ.λπ.

Στο επόμενο επίπεδο της στοίβας υπολογιστικής νέφους, μία ΠωΥ (πλατφόρμα ως υπηρεσία, Platform as a Service), όπως το Windows Azure ή το Google AppEngine μπορούν να περιορίσουν τον αριθμό των τυποποιημένων που διατίθενται, προσφέροντας μόνο τυποποιημένα μέρη που λειτουργούν με τις δικές τους πλατφόρμες. Από την πλευρά του εκάστοτε παρόχου ΠωΥ, η πρόθεση είναι να διατίθεται μία πλατφόρμα με τμήματα που μπορούν να συντεθούν μεταξύ τους.

Θεωρώντας, τον υψηλότερο βαθμό ολοκλήρωσης της υπολογιστικής νέφους, δηλ. το ΛωΥ (λογισμικό ως υπηρεσία, Software as a Service), η έννοια της συνθεσιμότητας για τους χρήστες μπορεί να εξαφανιστεί παντελώς. Ένας πάροχος ΛωΥ, όπως το Quicken.com ή το Salesforce.com παραδίδει μια εφαρμογή ως υπηρεσία σε έναν πελάτη, κι εκεί δεν υπάρχει κανένα ιδιαίτερο όφελος από τη σκοπιά του παρόχου να είναι σε θέση ο πελάτης να συνθέσει τις δικές του εφαρμογές. Ένας πάροχος υπηρεσιών που μεταπωλεί ένα ΛωΥ μπορεί να έχει τις ακόλουθες επιλογές:

- Να προσφέρει μία από εναλλακτικές μονάδες.
- Να προσαρμόσει τις πληροφορίες που περιλαμβάνονται σε μία μονάδα για έναν πελάτη.
- Να πωλήσει την υπηρεσία κάτω από το εμπορικό σήμα του.
- Να εκτελέσει κάποιο άλλο περιορισμένο είδος προσαρμογής.

Βέβαια οι τροποποιήσεις αυτές είναι γενικά περιορισμένες.

Αυτή η εικόνα, δηλαδή ότι η συνθεσιμότητα περιορίζεται καθώς μετακινούμαστε σε υψηλότερα επίπεδα στη στοιβία της υπολογιστικής νέφους, αφορά τη σκοπιά του χρήστη. Για τους φορείς παροχής υπηρεσιών ΠωΥ ή ΛωΥ, οι οποίοι έχουν ως στόχο τη δημιουργία μιας πλατφόρμας ή μιας υπηρεσίας που θα δοθεί σε στελέχη ανάπτυξης, μεταπωλητές ή χρήστες, το να έχουν στη διάθεσή τους ένα σύστημα με υψηλή συνθεσιμότητα είναι εξαιρετικά σημαντικό καθώς επωφελούνται τουλάχιστον στις εξής διαστάσεις:

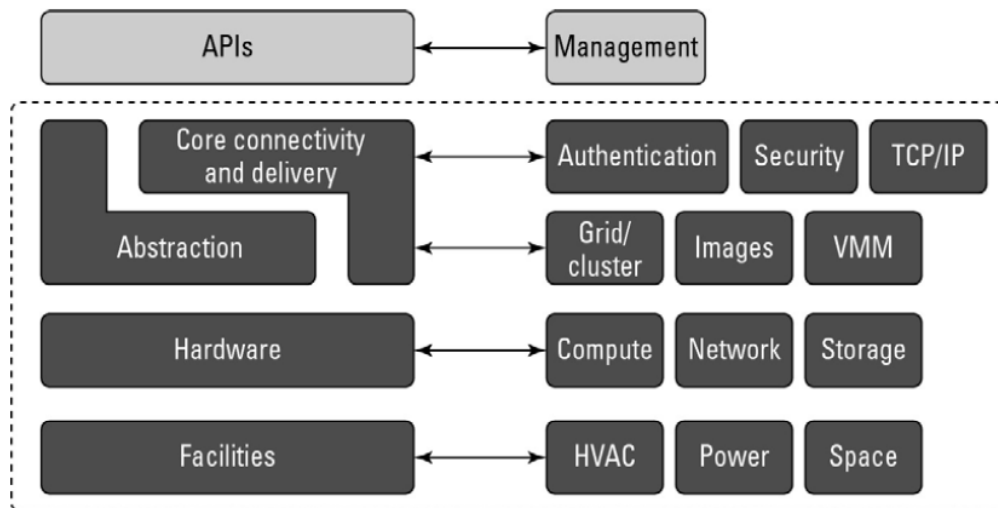
- Πιο εύκολη συναρμολόγηση συστημάτων.
- Φτηνότερη ανάπτυξη συστημάτων.
- Περισσότερο αξιόπιστη λειτουργία.
- Μεγαλύτερη ομάδα στελεχών ανάπτυξης.
- Διευκόλυνση χρήσης μεθοδολογιών λογικής σχεδίασης συστημάτων.

### 2.3 Υποδομή

Οι μεγαλύτεροι φορείς παροχής υπηρεσιών ΥωΥ στηρίζονται στην τεχνολογία εικονικών μηχανών για να διαμορφώσουν εξυπηρετές που μπορούν να εκτελέσουν εφαρμογές. Οι εικονικοί εξυπηρετές που περιγράφονται από την άποψη μιας εικόνας ή ενός στιγμιότυπου μηχανής, έχουν χαρακτηριστικά που χρησιμοποιούνται για την περιγραφή πραγματικών εξυπηρετών, οι οποίοι διαθέτουν ένα συγκεκριμένο αριθμό κύκλων του μικροεπεξεργαστή (ΚΜΕ – κεντρικής μονάδας επεξεργασίας - CPU), κάποια δεδομένη ταχύτητα πρόσβασης στη μνήμη και συγκεκριμένο εύρος ζώνης δικτύου. Στις εικονικές μηχανές ανατίθενται συγκεκριμένοι πόροι. Το λογισμικό που εκτελείται στις εικονικές μηχανές είναι αυτό που καθορίζει τη χρησιμότητα του συστήματος της υπολογιστικής νέφους.

Στην Εικόνα 10 παρουσιάζεται το μέρος της στοιβάς της υπολογιστικής νέφους που ορίζεται ως “εξυπηρετής”. Στο διάγραμμα, η προγραμματιστική διασύνδεση εφαρμογών (το API- Application Programming Interface) παρουσιάζεται με γκριζό χρώμα διότι είναι

ένα προαιρετικό συστατικό που δεν παραδίδεται πάντα με τον εξυπηρέτη. Το συστατικό *Επόπτης Εικονικών Μηχανών* (Virtual Machine Monitor – VMM) παρακολουθεί τις εικονικές μηχανές και καλείται επίσης και *hypervisor*. Πρόκειται για λογισμικό χαμηλού επιπέδου, το οποίο επιτρέπει σε διαφορετικά λειτουργικά συστήματα να εκτελεστούν σε διακριτές περιοχές μνήμης (memory spaces) και διαχειρίζεται την είσοδο - έξοδο (I/O) για τις εικονικές μηχανές.



**Εικόνα 10 - Μέρος του σωρού του υπολογισμού νεφών που έχει οριστεί ως εξυπηρέτης**

Η έννοια του εικονικού εξυπηρέτη παρέχει σε ένα στέλεχος ανάπτυξης εφαρμογών έναν νέο τρόπο σκέψης για τον προγραμματισμό εφαρμογών. Για παράδειγμα, όταν ένας προγραμματιστής δημιουργεί λογισμικό που απαιτεί την παράλληλη εκτέλεση διαφορετικών εργασιών, μπορεί να γράψει μια εφαρμογή η οποία δημιουργεί επιπρόσθετα νήματα εκτέλεσης, τα οποία πρέπει να διαχειρίζεται η εφαρμογή. Όταν ένα στέλεχος ανάπτυξης εφαρμογών δημιουργεί μια εφαρμογή που χρησιμοποιεί μια υπηρεσία νέφους, το στέλεχος αυτό μπορεί να συνδέσει την εφαρμογή με την κατάλληλη υπηρεσία και να δώσει έτσι τη δυνατότητα στην ίδια την εφαρμογή να φροντίσει για την ικανότητα κλιμάκωσής της. Για παράδειγμα, μια εφαρμογή τρισδιάστατης απεικόνισης, η οποία μπορεί να χρειάζεται πολύ χρόνο για να ολοκληρωθεί από έναν εξυπηρέτη, να μπορεί να κλιμακωθεί, συνδεδεμένη με πολλαπλούς εξυπηρέτες στο νέφος για μια μικρή χρονική περίοδο, ώστε να ολοκληρώσει την εργασία σε μια παρόμοια ή χαμηλότερη τιμή αλλά ταυτόχρονα πολύ πιο γρήγορα.

Στις μελλοντικές εφαρμογές, τα στελέχη ανάπτυξης εφαρμογών θα πρέπει να εξισορροπήσουν τις αρχιτεκτονικές ανάγκες των προγραμμάτων τους, έτσι ώστε οι εφαρμογές τους είτε να δημιουργούν νέα νήματα όταν μπορούν είτε να δημιουργούν νέες εικονικές μηχανές. Οι εφαρμογές θα πρέπει επίσης να εξετάζουν προσεκτικά το πως χρησιμοποιούν τους πόρους του νέφους, το πότε είναι σκόπιμο να κλιμακωθεί η εκτέλεση στο νέφος, το πως θα ελέγχονται τα στιγμιότυπα που εκτελούνται και πότε δεν χρειάζεται να επεκτείνεται η χρήση της εφαρμογής του νέφους. Αυτό θα απαιτήσει ένα νέο τρόπο σκέψης στην ανάπτυξη εφαρμογών και η δυνατότητα κατάλληλης κλιμάκωσης είναι κάτι που πρέπει να ενσωματωθεί στην αρχιτεκτονική των εφαρμογών.

## 2.4 Υπολογιστικό πλέγμα

Το *Υπολογιστικό Πλέγμα* (Grid Computing) είναι η ευκολότερη αρχιτεκτονική εφαρμογής για την ενσωμάτωση στο νέφος. Είναι γεγονός ότι μια εφαρμογή υπολογιστικού πλέγματος (Grid Computing) είναι ένα λογισμικό που χωρίζει την επεξεργασία του σε μικρά κομμάτια που μπορούν έπειτα να υποβληθούν μεμονωμένα σε επεξεργασία.

Στη συνέχεια, ως χρήστες του Υπολογιστικού Πλέγματος, αναφέρονται οι ακόλουθοι:

- Ερευνητές και μηχανικοί που:
  - Ελέγχουν απομακρυσμένα εφαρμογές και εργαλεία λογισμικού.
  - Προσπελούν αποθήκες δεδομένων και (υπερ)υπολογιστές.
  - Αναλύουν δεδομένα και εν συνεχεία τα εμφανίζουν σε παγκόσμιο επίπεδο.
- Βιομηχανίες που:
  - Συνδέουν ανθρώπους (μέσω video conferencing και συστημάτων απομακρυσμένης πρόσβασης).
  - Διαμοιράζουν πόρους.
  - Επιλύουν προβλήματα.
- Κυβερνήσεις που:
  - Εφαρμόζουν παγκόσμιες πολιτικές βασισμένες στο διαμοιρασμό συνεργατικών πληροφοριών.
  - Αντιμετωπίζουν καταστροφές με χρήση συστημάτων λήψης αποφάσεων.
- Λειτουργοί της παιδείας μέσω:
  - Δημιουργίας και λειτουργίας ψηφιακών αιθουσών (video conferencing).
  - παροχής συστημάτων απομακρυσμένης πρόσβασης σε βιβλιοθήκες, συλλογές, κ.τ.λ.



**Εικόνα 11 - Αρχιτεκτονική Υπολογιστικού Πλέγματος (Grid Computing)**

### 2.4.1 Παράδειγμα Υπολογιστικού πλέγματος - SETI

Σε αυτό το σημείο, ένα παράδειγμα το οποίο προσδίδει ιδιαίτερη έμφαση στην αρχιτεκτονική του υπολογιστικού πλέγματος (Grid Computing), αποτελεί το λεγόμενο SETI (the Search for Extra-Terrestrial Intelligence, η αναζήτηση της εξωγήινης νοημοσύνης).

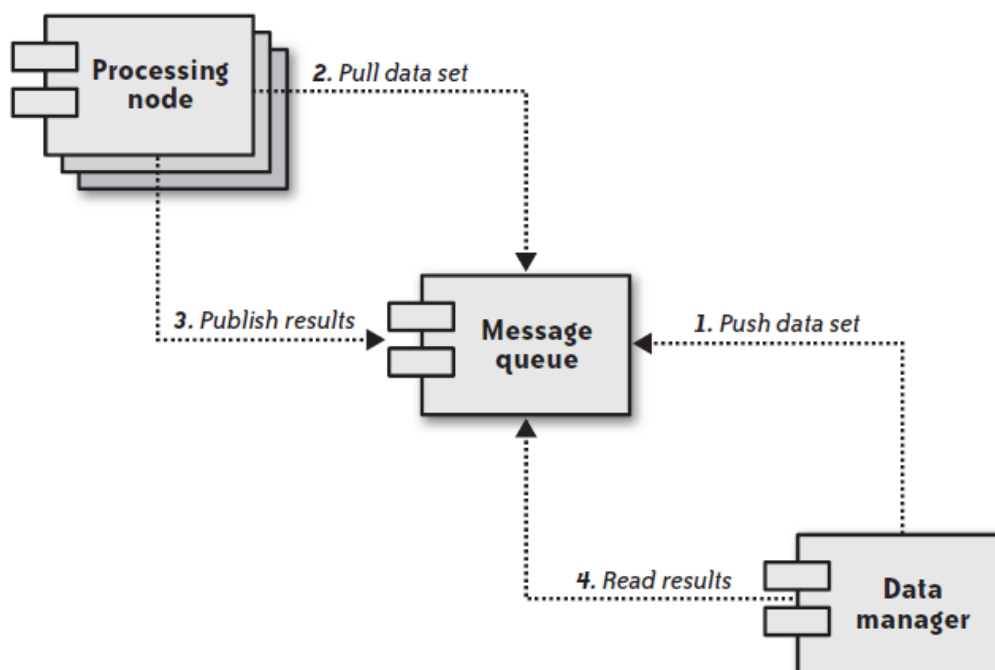
Το SETI διαθέτει ραδιο-τηλεσκόπια τα οποία είναι σε συνεχή δραστηριότητα με σκοπό τον εντοπισμό σημάτων από το διάστημα. Μέσω λοιπόν των ραδιοσυχνοτήτων, συλλέγονται στοιχεία τα οποία στη συνέχεια υπόκεινται σε επεξεργασία με σκοπό την εύρεση ενός μη φυσικού σήματος που μπορεί να αντιπροσωπεύει μια προσπάθεια επικοινωνίας μεταξύ άλλων πολιτισμών. Βέβαια πρέπει να σημειωθεί πως υπάρχει περίπτωση να χρειαστεί ένα μεγάλο χρονικό διάστημα για έναν υπολογιστή προκειμένου να επεξεργαστεί όλα εκείνα τα δεδομένα. Όμως, πολλοί υπολογιστές που χρησιμοποιούν μόνο τους εφεδρικούς κύκλους του επεξεργαστή τους, μπορούν να αντιμετωπίσουν το πρόβλημα αυτό εξαιρετικά γρήγορα

Γύρω στο 1999, το SETI αποφάσισε να χρησιμοποιήσει τους εφεδρικούς κύκλους των υπολογιστών γραφείου των καταναλωτών, για την επεξεργασία των δεδομένων του. Να αναφερθεί πως τα εμπορικά και κυβερνητικά συστήματα χρησιμοποιούνταν προκειμένου να δημιουργήσουν ένα δίκτυο υπερ-υπολογιστών και να εκτελέσουν εκ των υστέρων, τους ίδιους υπολογισμούς. Κατ' αυτόν τον τρόπο δημιουργήθηκε το λογισμικό SETI@home, όπου πρόκειται για ένα λογισμικό το οποίο υπάρχει ελεύθερο στο διαδίκτυο και κάθε χρήστης μπορεί να το εγκαταστήσει όποτε επιθυμεί στον υπολογιστή του. Γενικότερα το υπολογιστικό πλέγμα συγκροτούν, όλοι οι υπολογιστές οι οποίοι εκτελούν το λογισμικό SETI@home, συμπεριλαμβανομένων και των υπολογιστών γραφείου που χρησιμοποιούν στην καθημερινότητά τους απλοί χρήστες. Το SETI@home ενεργοποιείται όταν ένας υπολογιστής παραμένει σε αδράνεια και επεξεργάζεται τα

σύνολα των στοιχείων από την τελευταία φορά που ο υπολογιστής ήταν συνδεδεμένος στο διαδίκτυο. Στη συνέχεια, τα δεδομένα αυτά υποβάλλονται στο SETI, το οποίο ελέγχει δύο φορές οτιδήποτε αφορά την επεξεργασία από άλλους συμμετέχοντες (πέραν δηλαδή αυτών που θεωρούνται *έμπιστοι*) και σε περίπτωση που βρεθεί κάποιο ενδιαφέρον στοιχείο εξετάζεται επιπλέον.

Προσφάτως, οι φάρμες εξυπηρετών (server farms) δημιουργήθηκαν για εργασίες υπολογιστικού πλέγματος (Grid Computing), όπως είναι η απόδοση βίντεο (video rendering). Τόσο οι υπερ-υπολογιστές όσο και οι φάρμες εξυπηρετών, είναι πολύ δαπανηρές επένδυσης. Όσον αφορά το νέφος, αυτό αποτελεί μια φθηνή και εύκολη λύση για την εκτέλεση μιας εφαρμογής υπολογιστικού πλέγματος: Όταν υπάρχουν δεδομένα προς επεξεργασία, απλά καλείται ένας εξυπηρετής για την επεξεργασία των δεδομένων αυτών. Κατόπιν, εκείνος ο εξυπηρετής μπορεί είτε να “κλείσει” είτε να “αντλήσει” ένα άλλο σύνολο δεδομένων προς επεξεργασία.

Στη συνέχεια, στην Εικόνα 12 επεξηγείται η ροή διαδικασίας μιας εφαρμογής υπολογισμού πλέγματος. Αρχικά, ένας εξυπηρετής ή μια συστάδα εξυπηρετών λαμβάνει τα δεδομένα που απαιτούν επεξεργασία. Υποβάλλει έπειτα εκείνη την εργασία σε μια ουρά μηνυμάτων (μέσω της ροής με αριθμό 1). Επίσης άλλοι εξυπηρετές, που συχνά αποκαλούνται *εργάτες* (workers) (ή στην περίπτωση των SETI@home, υπολογιστές γραφείου), εξετάζουν την ουρά μηνυμάτων (μέσω της ροής με αριθμό 2) με σκοπό την εύρεση των νέων συνόλων δεδομένων. Κατά την εμφάνιση ενός νέου συνόλου δεδομένων, ο πρώτος υπολογιστής που τοχνεύει αν, το επεξεργάζεται κι έπειτα στέλνει τα αποτελέσματα πίσω στην ουρά μηνυμάτων (μέσω της ροής με αριθμό 3). Τα δύο στοιχεία (ο πρώτος υπολογιστής και η ουρά) μπορούν να λειτουργήσουν ανεξάρτητα το ένα από το άλλο [6].



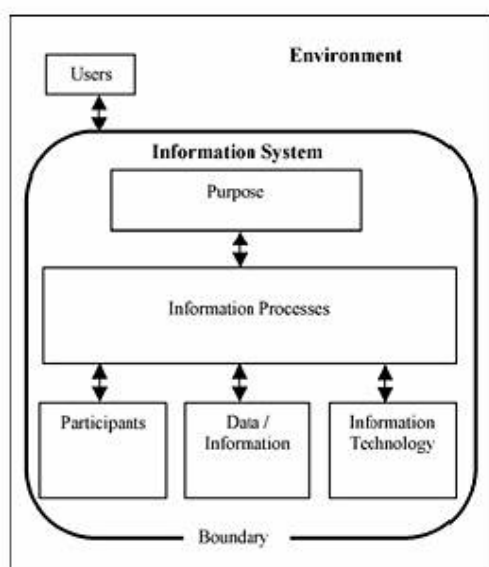
**Εικόνα 12 - Η αρχιτεκτονική του υπολογιστικού πλέγματος (grid computing) διαχωρίζει την εφαρμογή πυρήνα από τους κόμβους επεξεργασίας των δεδομένων της**

Επιπροσθέτως, η έννοια του υπολογιστικού νέφους (cloud computing) βοηθάει στη συγκεκριμένη περίπτωση, εξαιτίας του ότι δεν υπάρχει η ανάγκη κατοχής κάποιων από τους εξυπηρετές, όταν δεν υπάρχουν κάποια δεδομένα προς επεξεργασία. Έπειτα, ο αριθμός των εξυπηρετών μπορεί να αυξομειωθεί για την υποστήριξη των απαιτούμενων συνόλων δεδομένων που χρειάζονται από την εφαρμογή. Με άλλα λόγια, αντί να υπάρχουν υπολογιστές που να βρίσκονται εκτός λειτουργίας και να περιμένουν νέα δεδομένα να φτάσουν για να απασχοληθούν, υπάρχει η δυνατότητα ύπαρξης αυτών των εξυπηρετών, οι οποίοι “ανοίγουν” καθώς αυξάνει το ποσοστό εισερχόμενων δεδομένων, και “κλείνουν” καθώς μειώνεται το ποσοστό εισερχόμενων δεδομένων.

Επιπλέον, για το λόγο ότι ο υπολογισμός πλέγματος περιορίζεται σε μια μικρή αγορά (στην επιστημονική, στην οικονομική και σε άλλα δεδομένα μεγάλης κλίμακας), στη συνέχεια θα αναλυθεί ένας πιο διαδεδομένος υπολογισμός, η *υπολογιστική συναλλαγών* (Transactional Computing).

## 2.5 Υπολογιστική συναλλαγών

Σύστημα συναλλαγών ορίζεται το σύστημα στο οποίο ένα ή περισσότερα στοιχεία εισερχόμενων δεδομένων υποβάλλονται σε από κοινού επεξεργασία ως μια ενιαία συναλλαγή και καθιερώνουν σχέσεις με άλλα στοιχεία που ήδη υπάρχουν στο σύστημα. Ως συναλλαγή, θεωρείται η δραστηριότητα εκείνη η οποία αλλάζει τα αποθηκευμένα δεδομένα και παραδείγματα μιας τέτοιας δραστηριότητας είναι η χρήση της πιστωτικής κάρτας, η πραγματοποίηση μιας κράτησης αλλά και η πραγματοποίηση ανάληψης μετρητών από ένα ATM. Ο πυρήνας ενός συστήματος συναλλαγών είναι γενικά μια σχεσιακή βάση δεδομένων, η οποία διαχειρίζεται τις σχέσεις μεταξύ όλων των δεδομένων που αποτελούν το σύστημα. Επιπλέον, ένα σύστημα επεξεργασίας συναλλαγών συλλέγει και αποθηκεύει δεδομένα σχετικά με συναλλαγές και μερικές φορές ελέγχει αποφάσεις που λαμβάνονται ως μέρος μιας συναλλαγής.



**Εικόνα 13 - Αρχιτεκτονική Υπολογιστικών Συναλλαγών (Transactional Computing)**

Διακρίνονται οι εξής δύο τύποι επεξεργασίας συναλλαγών:

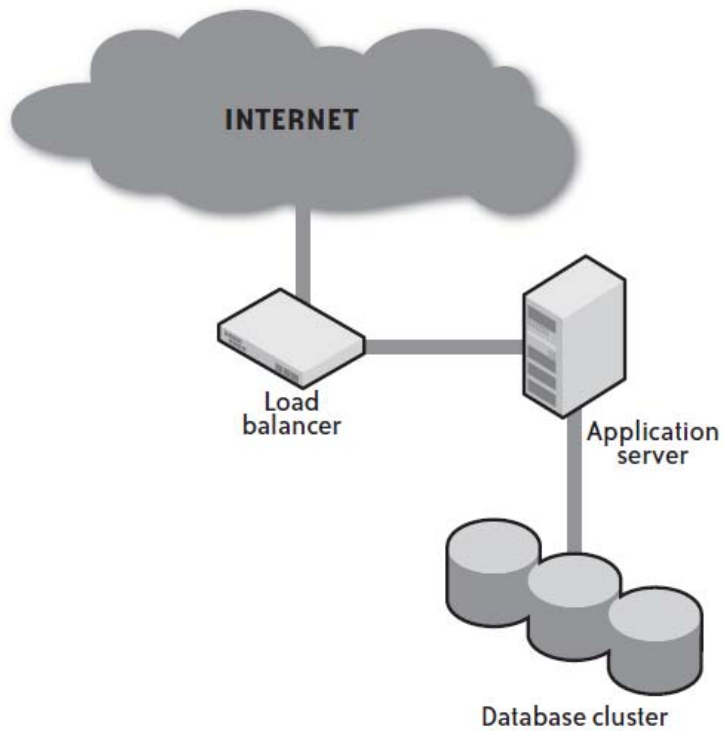
- Μαζική επεξεργασία συναλλαγών (Batch transaction processing): οι πληροφορίες συλλέγονται και αποθηκεύονται, αλλά δεν υπόκεινται προς επεξεργασία άμεσα. Η επεξεργασία των πληροφοριών γίνεται μαζικά, σε μεταγενέστερο της συλλογής χρόνο. Παραδείγματα της επεξεργασίας αυτής αποτελούν, η επεξεργασία παραγγελιών σε μία αποθήκη ή οι φορολογικοί έλεγχοι.
- Επεξεργασία συναλλαγών πραγματικού χρόνου (Real time transaction processing). Αποτελείται από μια πράξη, η οποία υποβάλλεται αμέσως σε επεξεργασία και ο χειριστής έχει πρόσβαση σε μια online βάση δεδομένων. Παραδείγματα συναλλαγών πραγματικού χρόνου είναι η ανάληψη χρημάτων από έναν τραπεζικό λογαριασμό και οι δανεισμοί βιβλίων σε μια βιβλιοθήκη.

Στην Εικόνα 14 παρουσιάζεται το λογικό σχεδιάγραμμα ενός συστήματος συναλλαγών με μεγάλη διαθεσιμότητα. Στα πλαίσια αυτού του είδους αρχιτεκτονικής, ένας εξυπηρετής εφαρμογής (application servers) διαχειρίζονται τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων και τα παρουσιάζουν στους χρήστες μέσω μιας διαδικτυακής διεπαφής. Ο εξυπηρετής εφαρμογής υλοποιεί την *επιχειρηματική λογική*, τους κανόνες δηλαδή του οργανισμού σχετικά με το πώς γίνεται η επεξεργασία των δεδομένων. Οι χρήστες αλληλεπιδρούν με τα δεδομένα μέσω της συγκεκριμένης διεπαφής. Οι περισσότεροι ιστόχωροι και εφαρμογές του Διαδικτύου που χρησιμοποιούνται καθημερινά, είναι κάποιας μορφής του συστήματος συναλλαγών. Για την υψηλή διαθεσιμότητα και αυξημένες επιδόσεις, είναι δυνατόν να χρησιμοποιηθούν πολλαπλοί εξυπηρετές εφαρμογής, και μεταξύ των εξυπηρετών και των χρηστών τοποθετείται ένα σύστημα *εξισορρόπησης φόρτου*, το οποίο διαμοιράζει τα εισερχόμενα αιτήματα μεταξύ των εξυπηρετών, μεριμνώντας να μην προωθούνται αιτήματα στους εξυπηρετές που βρίσκονται εκτός λειτουργίας.

Σε ό, τι αφορά την ανάπτυξη ενός συστήματος συναλλαγών στο νέφος, να σημειωθεί ότι είναι πιο σύνθετη και λιγότερο προφανής από την ανάπτυξη ενός συστήματος πλέγματος. Μία αιτία που αυξάνει την πολυπλοκότητα είναι ότι σε ένα σύστημα πλέγματος οι κόμβοι είναι βραχύβιοι, ενώ σε ένα σύστημα συναλλαγών οι κόμβοι να είναι μακρόβιοι.

Ακόμη, μια βασική πρόκληση για οποιοδήποτε σύστημα που απαιτεί μακρόβιους κόμβους σε μια υποδομή νεφών, αποτελεί το γεγονός ότι ο μέσος χρόνος μεταξύ των αποτυχιών (mean time between failures, MTBF) ενός εικονικού εξυπηρετή είναι απαραίτητως μικρότερος από τον αντίστοιχο του υλικού. Για παράδειγμα, αν υπάρχουν δύο φυσικοί εξυπηρετές με μέσο χρόνο μεταξύ των αποτυχιών ίσο με τρία χρόνια, όπου ο ένας μπορεί να αναλάβει όλη την εργασία σε περίπτωση αποτυχίας του άλλου, είναι λιγότερο πιθανό να έχουμε μια διακοπή λειτουργίας ολόκληρου του συστήματος, απ' ό,τι θα ήταν αν είχαμε έναν μόνο φυσικό εξυπηρετή με δύο εικονικούς κόμβους. Ο αριθμός των φυσικών κόμβων είναι αυτός που καθορίζει το MTBF και από τη στιγμή που υπάρχουν λιγότεροι φυσικοί κόμβοι. Το MTBF είναι χαμηλότερο για οποιοδήποτε δοσμένο κόμβο στο σύστημα συναλλαγών που βασίζεται σε νέφος, έναντι του αντίστοιχου συστήματος που βασίζεται σε φυσικούς επεξεργαστές.



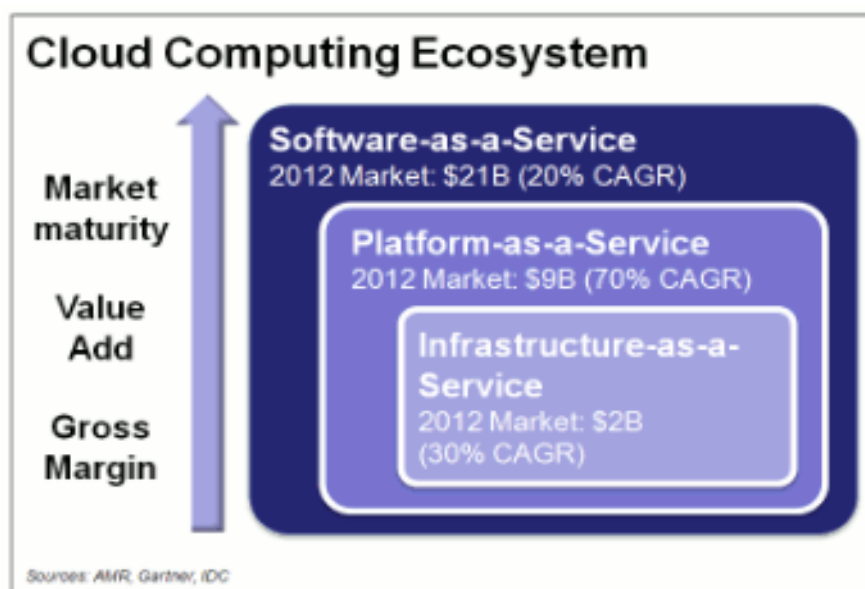


*Εικόνα 14 - Μια εφαρμογή συναλλαγών χωρίζει μια εφαρμογή σε παρουσίαση, επιχειρηματική λογική και αποθήκευση δεδομένων*

### Κεφάλαιο 3: Μοντέλα εφαρμογών και υπηρεσιών στο υπολογιστικό νέφος

Σε αυτό το κεφάλαιο περιγράφονται κάποιοι από τους διαφορετικούς τύπους μοντέλων του υπολογιστικού νέφους, οι οποίοι κατηγοριοποιούνται ως ένα σύνολο μοντέλων υπηρεσιών. Οι εφαρμογές του νέφους υποστηρίζονται από ένα σύνολο στρωμάτων, επάνω στο οποίο οι καταναμημένες εφαρμογές μπορούν να χτιστούν είτε να φιλοξενηθούν. Αυτά τα στρώματα περιλαμβάνουν την υποδομή, την πλατφόρμα και το λογισμικό. Ακόμη, ένας χρήστης μπορεί να στηριχτεί στα συγκεκριμένα στρώματα για να δημιουργήσει εφαρμογές βασισμένες στο νέφος, ανάλογα με τον τύπο και το επίπεδο της υπηρεσίας που του προσφέρεται.

Χρήσιμο είναι να αναφερθεί πως το υπολογιστικό νέφος μπορεί να διαχωριστεί σε τρία κύρια στρώματα, τα οποία διαμορφώνουν ένα οικοσύστημα υπολογιστικού νέφους, όπως φαίνεται στην Εικόνα 15. Επίσης, το στρώμα εφαρμογής αποτελεί τη βάση για το λογισμικό ως υπηρεσία (ΛωΥ), ενώ το στρώμα πλατφόρμας αποτελεί τη βάση για τα μοντέλα πλατφόρμας ως υπηρεσία (ΠωΥ) που αναλύονται αργότερα.



**Εικόνα 15 - Οικοσύστημα Υπολογιστικού Νέφους (Cloud Computing Ecosystem)**

Τα μοντέλα υπηρεσιών που περιγράφονται σε αυτό το σημείο, δηλαδή η υποδομή ως υπηρεσία (ΥωΥ), το λογισμικό ως υπηρεσία (ΛωΥ) και η πλατφόρμα ως υπηρεσία (ΠωΥ), είναι χρήσιμα για την ταξινόμηση των δυνατοτήτων του υπολογιστικού νέφους, αλλά και σε συγκεκριμένες ιδιότητες των προμηθευτών, των προϊόντων και υπηρεσιών. Ακόμη η υποδομή ως υπηρεσία επιτρέπει τη δημιουργία εικονικών υπολογιστικών συστημάτων ή δικτύων.

Το λογισμικό ως υπηρεσία αντιπροσωπεύει μια φιλοξενούμενη εφαρμογή που είναι παγκοσμίως διαθέσιμη μέσω του Διαδικτύου, συνήθως μέσω μιας εφαρμογής πλοήγησης. Με το λογισμικό ως υπηρεσία, ο χρήστης αλληλεπιδρά άμεσα με το φιλοξενούμενο λογισμικό. Το ΛωΥ μπορεί να θεωρηθεί ως ένα εναλλακτικό μοντέλο του

λογισμικού που πωλείται στα καταστήματα (Commercial off-the-shelf, COTS) και μπορεί να αντικαταστήσει ένα μεγάλο μέρος του λογισμικού που αγοράζεται σήμερα.

Η πλατφόρμα ως υπηρεσία είναι μια υποδομή υπολογιστικού νέφους, η οποία παρέχει ένα περιβάλλον ανάπτυξης επάνω στο οποίο μπορούν να κατασκευαστούν οι εφαρμογές. Το ΠωΥ παρέχει ένα μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να δημιουργήσει ή να επεκτείνει σύνθετες εφαρμογές, όπως τα συστήματα διαχείρισης σχέσης πελατών (Customer relationship management, CRM) ή τα συστήματα προγραμματισμού επιχειρηματικών πόρων (Enterprise resource planning, ERP). Επίσης, το ΠωΥ προσφέρει διάφορα οφέλη που έχει το υπολογιστικό νέφος και συχνά βασίζεται στο μοντέλο της υπηρεσιοστρεφούς αρχιτεκτονικής.

Καθώς το υπολογιστικό νέφος ωριμάζει, διάφοροι τύποι υπηρεσιών εισάγονται και ταιριάζουν με αυτές τις αρχιτεκτονικές. Από αυτούς τους τύπους υπηρεσιών, η πιο ολοκληρωμένη είναι η ΤωΥ (ταυτότητα ως υπηρεσία - IaaS - Identity as a Service). Αυτού του είδους η ταυτότητα διαθέτει τη δυνατότητα να παρέχει υπηρεσίες πιστοποίησης κι εξουσιοδότησης σε καταναμημένα δίκτυα.

Τα πρωτόκολλα υποδομής και υποστήριξης για ΤωΥ περιγράφονται αργότερα σε αυτό το κεφάλαιο. Άλλοι τύποι υπηρεσιών όπως η *συμμόρφωση ως υπηρεσία* (ΣωΥ), ο έλεγχος, οι επικοινωνίες, και πολλές κάθετες υπηρεσίες που είναι υπό ανάπτυξη αναφέρονται στο παρόν κεφάλαιο [1].

### 3.1 Υποδομή ως υπηρεσία (ΥωΥ)

Σε αυτή την ενότητα αναλύεται η προσέγγιση *υποδομή ως υπηρεσία* ΥωΥ (Infrastructure as a Service), η οποία αποτελεί ένα είδος υπηρεσίας υπολογιστικής νέφους, όπου το υλικό το οποίο που διατίθεται στους χρήστες εικονικοποιείται στο νέφος.

Αναλυτικότερα, η υποδομή αυτή δημιουργεί ένα μοντέλο υπολογιστικής χρησιμότητας, το οποίο μπορεί να εκμεταλλευτεί ανάλογα με τις ανάγκες του ο κάθε χρήστης, χωρίς σημαντικά όρια στη δυνατότητα κλιμάκωσης.

Συγκεκριμένα, ο κάθε χρήστης ανάλογα με τις απαιτήσεις του, μπορεί να χρησιμοποιήσει τους υπολογιστικούς πόρους που του παρέχονται και να πληρώσει σύμφωνα με το μοντέλο πληρωμής, ανάλογα με τη χρήση. Αυτό σημαίνει πως μια επιχείρηση ή ακόμα ένας ιδιώτης δε χρειάζεται να προβεί σε περαιτέρω έξοδα υπολογιστικής υποδομής, με αποτέλεσμα να μειώνεται αισθητά το κόστος λειτουργίας των υπηρεσιών των εκάστοτε επιχειρήσεων.

Γενικά, το ΥωΥ μπορεί να παρουσιάζεται ως μια άριστα εξελιγμένη τεχνολογία, που διαθέτει την ικανότητα να βοηθήσει την όποια μικρή επιχείρηση να εξελιχθεί μέσα σ' ένα σχετικά γρήγορο χρονικό διάστημα. Αυτό συμβαίνει μιας και διαθέτει πρόσβαση σε μεγαλύτερου και ανώτερου επιπέδου τεχνολογίες. Ακόμη, τα στελέχη ανάπτυξης εφαρμογών αλληλεπιδρούν με το μοντέλο ΥωΥ και δημιουργούν εικονικό υλικό στο οποίο αναπτύσσονται εφαρμογές και υπηρεσίες. Συγκεκριμένα, ένας πάροχος ΥωΥ έχει δημιουργήσει μια υπηρεσία χρησιμότητας υλικού και ο χρήστης προμηθεύεται εικονικούς πόρους, όπως απαιτείται προκειμένου να ολοκληρωθεί μια υπηρεσία.



**Εικόνα 16 - Υποδομή ως υπηρεσία (YωY)**

Στο YωY, οι εικονικοί πόροι αντιστοιχίζονται στα πραγματικά συστήματα. Όταν ο εξυπηρετούμενος αλληλεπιδρά με μια υπηρεσία YωY και αιτείται τους πόρους από τα εικονικά συστήματα, τα αιτήματα κατευθύνονται τελικά στους πραγματικούς εξυπηρετές που πραγματοποιούν την πραγματική εργασία.

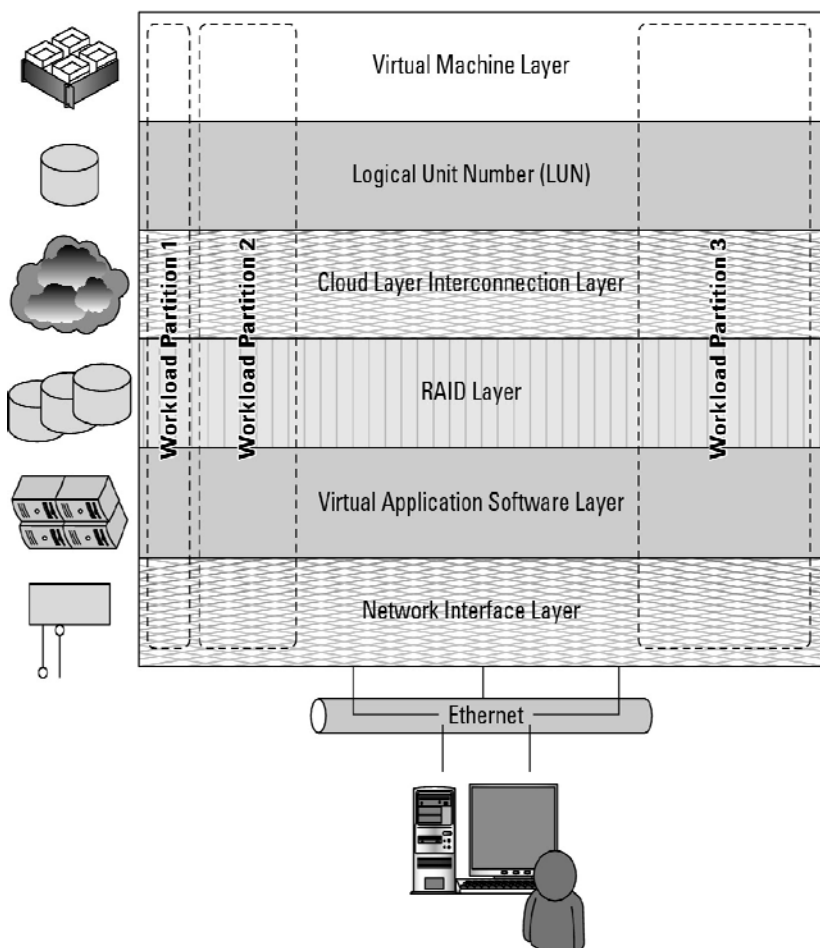
### 3.1.1 Φόρτος εργασίας του YωY

Η θεμελιώδης μονάδα του εικονικού πελάτη σε μια διεργασία YωY καλείται *φόρτος εργασίας*. Ένας φόρτος εργασίας προσομοιώνει τη δυνατότητα ενός συγκεκριμένου τύπου πραγματικού εξυπηρετή να υλοποιήσει ένα σύνολο εργασιών. Η εργασία που γίνεται μπορεί να μετρηθεί από τον αριθμό συναλλαγών ανά λεπτό (TPM) ή μια παρόμοια μετρική. Επιπροσθέτως προς τη μετρική της απόδοσης αυτή, ο φόρτος εργασίας έχει ορισμένες παραμέτρους όπως οι είσοδοι/έξοδοι δίσκου (Disk I/Os) που μετρώνται σε Input/Output Per Second (IOPS), η ποσότητα σε μνήμη τυχαίας προσπέλασης (Random Access Memory – RAM) που καταναλώνεται σε Mbytes, λήψεις/αποστολές δεδομένων από το δίκτυο, κ.τ.λ. Η Εικόνα 17 απεικονίζει τον τρόπο που τρεις περιπτώσεις εικονικών ιδιωτικών εξυπηρετών, διαχωρίζονται σε μια στοίβα YωY. Τα τρία φορτία εργασιών απαιτούν τρία διαφορετικά μεγέθη υπολογιστών (μικρό, μεσαίο και μεγάλο).

Ένας πελάτης, αν δεν χρησιμοποιούσε το πλέγμα, θα έπρεπε να διατηρεί την αντίστοιχη μηχανή που απαιτείται για να εκτελεστεί κάθε ένας από αυτούς τους φόρτους εργασίας. Η υποδομή YωY εκτελεί αυτά τα στιγμιότυπα εξυπηρετών στο κέντρο δεδομένων που προσφέρει η υπηρεσία, παίρνοντάς τα από μια ομάδα εικονικών μηχανών. Υπάρχουν τρία στρώματα που αποτελούν εκφράσεις των φυσικών συστημάτων και διαχωρίζονται ως λογικές μονάδες (Logical UNits). Τα LUNs που αφορούν το *στρώμα υπερσύνδεσης νέφους* και το *στρώμα εικονικής εφαρμογής λογισμικού* είναι λογικές δομές. Τα LUNs είναι περιέκτες (containers) λογικής αποθήκευσης. Το στρώμα υπερσύνδεσης νέφους είναι ένα εικονικό δίκτυο στο οποίο ανατίθενται IP διευθύνσεις από το δίκτυο YωY και το στρώμα λογισμικού εικονικής εφαρμογής περιέχει το λογισμικό που εκτελείται στις

εικονικές μηχανές (virtual machine - VM) που έχουν χωριστεί από τα φυσικά στοιχεία στο εικονικό ιδιωτικό νέφος του ΥωΥ. Το LUN RAID παρέχει την αποθήκευση σε φυσικούς δίσκους.

Από τη σκοπιά της αρχιτεκτονικής, στον πελάτη μιας υποδομής ΥωΥ, ανατίθεται το δικό του ιδιωτικό δίκτυο. Το Amazon Elastic Computer Cloud (EC2), συμπεριφέρεται όπως κάθε εξυπηρετής που διαθέτει το δικό του ξεχωριστό εσωτερικό δίκτυο, εκτός κι αν ο πελάτης δημιουργήσει το δικό του εικονικό ιδιωτικό νέφος (μέσω μιας πρόσθετης λειτουργίας του EC2), το οποίο παρέχει ένα εναλλακτικό τρόπο επίλυσης αυτού του προβλήματος. Καθώς επεκτείνεται η διεργασία στο EC2, νέα δίκτυα προστίθενται στην υποδομή, κάτι το οποίο καθιστά εύκολη τη λογική κλιμάκωση μιας επέκτασης EC2, αλλά επιβάλλει τα πρόσθετα γενικά έξοδα δικτύων επειδή η κυκλοφορία πρέπει να καθοδηγηθεί μεταξύ των λογικών δικτύων. Τα όρια δρομολόγησης των υπηρεσιών Ιστού της Amazon, μεταδίδουν και διανέμουν σε πολλαπλούς αποδέκτες την κυκλοφορία, όπου αυτό απαιτείται, διότι η δικτύωση Layer-2 (Σύνδεση δεδομένων) δεν υποστηρίζεται. Το Rackspace cloud (<http://www.rackspacecloud.com>) ακολουθεί το μοντέλο ανάθεσης Υπηρεσιών Διαδικτύου της Amazon (AWS) IP.



**Εικόνα 17 - Ένα εικονικό ιδιωτικό τμήμα εξυπηρετή σ' ένα νέφος ΥωΥ**

Σύνηθες παράδειγμα ενός μοντέλου υπηρεσιών ΥωΥ είναι οι Υπηρεσίες Διαδικτύου της Amazon (AWS) στο Amazon.com. Οι υπηρεσίες Διαδικτύου της Amazon (AWS)

διαθέτουν διάφορα κέντρα δεδομένων στα οποία οι εξυπηρετές εκτελούνται πάνω σε μια εικονική πλατφόρμα (Xen) και χωρίζονται σε λογικές υπολογιστικές μονάδες διάφορων μεγεθών.

Επίσης, τα στελέχη ανάπτυξης εφαρμογών έχουν τη δυνατότητα είτε να εφαρμόσουν αντίγραφα (images) συστημάτων τα οποία περιέχουν διαφορετικά λειτουργικά συστήματα και εφαρμογές είτε να δημιουργούν τα δικά τους αντίγραφα συστημάτων. Η αποθήκευση μπορεί να γίνεται σε διαφορετικά τμήματα, οι βάσεις δεδομένων μπορούν να δημιουργηθούν και μια σειρά από υπηρεσίες όπως τα μηνύματα και οι ειδοποιήσεις μπορούν να καλούνται ώστε η κατανεμημένη εφαρμογή να δουλεύει σωστά.

### **3.2 Πλατφόρμα ως υπηρεσία (ΠωΥ)**

Το μοντέλο του υπολογιστικού νέφους ως υπηρεσία (ΠωΥ) είναι ένα περιβάλλον λογισμικού το οποίο παρέχει στον χρήστη την κατάλληλη υπολογιστική υποδομή, προκειμένου να λειτουργούν οι εφαρμογές του μέσω του διαδικτύου με τη μορφή υπηρεσίας. Η πλατφόρμα υπηρεσίας προσφέρει στον χρήστη τα κατάλληλα εργαλεία και το κατάλληλο περιβάλλον, για να του δοθεί η δυνατότητα να δημιουργήσει τις εφαρμογές σύμφωνα με τις απαιτήσεις του.

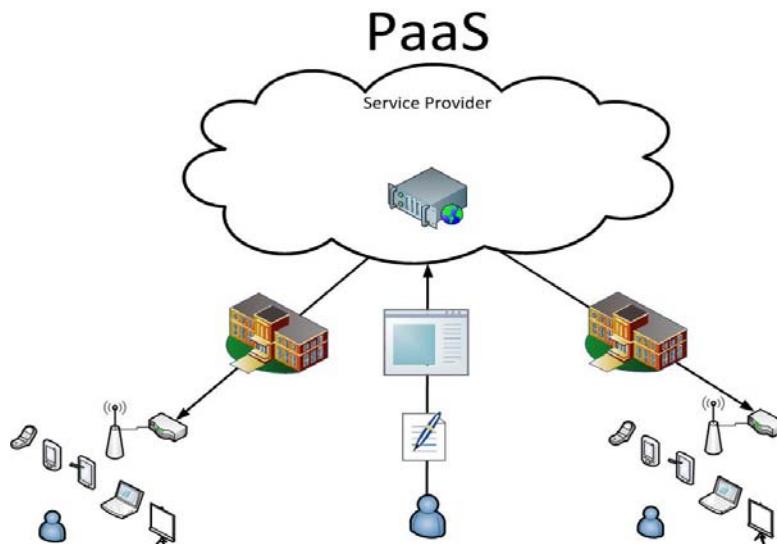
Δίνεται ακόμη η ευκαιρία στον κάθε χρήστη σε αυτό το είδος υπηρεσίας να μπορεί να αλληλεπιδράσει με το λογισμικό, δηλαδή μπορεί να εισάγει, αλλά και να ανακτήσει δεδομένα, να εκτελέσει ενέργειες της πλατφόρμας, καθώς και να ανακτήσει/αντλήσει αποτελέσματα. Έχει επίσης τη δυνατότητα να δημιουργήσει από την αρχή μια δική του εφαρμογή (π.χ. κάποια επιχειρηματική εφαρμογή ή μια εφαρμογή για μέσα κοινωνικής δικτύωσης (Social Network), ή ακόμα και να δοκιμάσει να επεκτείνει κάποια ήδη υπάρχουσα εφαρμογή στο νέφος. Όλα αυτά γίνονται βέβαια με τη χρήση διαφόρων γλωσσών προγραμματισμού που υποστηρίζονται από τον πάροχο υπηρεσιών.

Συνεπώς ο χρήστης είναι υπεύθυνος για την εγκατάσταση και τη διαχείριση των εφαρμογών που αναπτύσσει στο εκάστοτε περιβάλλον, ενώ ο πάροχος υπηρεσιών είναι αρμόδιος για όλα τα λειτουργικά θέματα της υπηρεσίας, όπως η διατήρηση και η διαχείριση του κύκλου ζωής των προσφερόμενων υπηρεσιών του, τα πλαίσια ανάπτυξης, καθώς και την υπολογιστική υποδομή. Ακόμη αξίζει να σημειωθεί πως η παροχή αυτών των υπηρεσιών σημαίνει χαμηλότερο κόστος για τους χρήστες, καθ' ότι δεν χρειάζεται να δαπανούν επιπλέον κεφάλαια για την εγκατάσταση υλικού και λογισμικού στον υπολογιστή τους.

Ένα παράδειγμα πλατφόρμας υπηρεσίας είναι η πλατφόρμα της Google App Engine όπου τα εργαλεία, αλλά και η δομή του συστήματος καθορίζονται από την Google. Ένα στέλεχος ανάπτυξης μπορεί να δημιουργήσει μια εφαρμογή σε κάποια γλώσσα προγραμματισμού, όπως την Python ή τη Java, χρησιμοποιώντας την καθορισμένη προγραμματιστική διασύνδεση εφαρμογών της Google (Google API).

Επίσης, η Google λειτουργεί ως προμηθευτής ΠωΥ μέσα στο σύστημα αυτό, δεδομένου ότι παρέχει αρκετές εφαρμογές υπηρεσιών Ιστού στους πελάτες της ως ένα τμήμα αυτού του μοντέλου υπηρεσιών. Παραδείγματα αυτού του τύπου αποτελούν το Google Maps, το Google Earth, το Gmail κ.ά.

Ένα σημείο στο οποίο το ΠωΥ μειονεκτεί είναι η έλλειψη μεταφερσιμότητας: εάν κάποιος προγραμματιστής-χρήστης δημιουργήσει μια εφαρμογή που γράφεται σε μια συγκεκριμένη γλώσσα προγραμματισμού, όπως π.χ. στην γλώσσα Python χρησιμοποιώντας το Google App Engine, είναι πολύ πιθανό η εφαρμογή αυτή να μη λειτουργεί σε κάποιο άλλο περιβάλλον.



*Εικόνα 18 - Πλατφόρμα ως υπηρεσία (ΠωΥ)*

### **3.3 Λογισμικό ως υπηρεσία (ΛωΥ)**

Το λογισμικό ως υπηρεσία (ΛωΥ) θεωρείται ότι αποτελεί το πληρέστερο μοντέλο υπηρεσιών υπολογιστικού νέφους. Αυτό συμβαίνει διότι το υπολογιστικό υλικό, καθώς επίσης και το λογισμικό παρέχονται από έναν προμηθευτή με τη μορφή προσφοράς πλήρους υπηρεσίας.

Γενικά το λογισμικό ως υπηρεσία (ΛωΥ) ως η παροχή στον χρήστη μιας πλήρους υπολογιστικής υποδομής καθώς και λειτουργικότητα λογισμικού που υλοποιεί μία συγκεκριμένη υπηρεσία, μέσω είτε μιας διεπαφής είτε μιας εφαρμογής πλοήγησης. Για τον λόγο αυτό, ο χρήστης μπορεί να έχει πρόσβαση στην υπηρεσία από όπου και αν βρίσκεται, οποιαδήποτε στιγμή το θελήσει. Εν ολίγοις, το λογισμικό προσεγγίζεται διεθνώς μέσω του Διαδικτύου.

Σήμερα, πολλοί χρήστες αποκτούν εξοικείωση με τα συστήματα ΛωΥ, τα οποία είναι είτε αντικαταστάσεις είτε υποκατάστατα του τοπικά εγκατεστημένου λογισμικού. Κάποια γνωστά παραδείγματα του λογισμικού ΛωΥ για τους τελικούς χρήστες αποτελούν το Gmail, το ημερολόγιο της Google, το Facebook, το twitter κ.ά.. Πιο αναλυτικά, π.χ. το twitter είναι διαθέσιμο μέσω του παγκόσμιου ιστού, επομένως παρέχεται ευρεία πρόσβαση σε όλους τους χρήστες. Επίσης δεν απαιτεί κάποια εγκατάσταση λογισμικού και δεν αποθηκεύει δεδομένα στον υπολογιστή του εκάστοτε χρήστη.

Ωστόσο οι εφαρμογές του λογισμικού ΛωΥ έρχονται να συγχωνευτούν και να βοηθήσουν και σε ό,τι αφορά τον επιχειρηματικό τομέα, διότι παρέχονται σε διάφορες μορφές και μεγέθη και περιλαμβάνουν εξαιρετικά προσαρμοσμένο λογισμικό, όπως

δηλαδή τις εφαρμογές διαχείρισης σχέσεων πελατών (CRM), εφαρμογές Help Desk, λύσεις διαχείρισης ανθρώπινου δυναμικού (HR), καθώς επίσης και πάρα πολλές online εκδόσεις γνωστών εφαρμογών.

Στο μοντέλο ΛωΥ η υποδομή της υπηρεσίας, οι εφαρμογές καθώς και το λογισμικό ελέγχονται από τον πάροχο υπηρεσιών, ανεξάρτητα από το που εκτελούνται τα διαφορετικά τμήματα του λογισμικού. Ως συνέπεια αυτού οι αναβαθμίσεις και ενημερώσεις της εφαρμογής να γίνονται αυτόματα και οι όποιες αλλαγές να πραγματοποιούνται πολύ πιο γρήγορα.

Συμπερασματικά, το μοντέλο ΛωΥ έχει ως αποτέλεσμα να μειώνονται οι δαπάνες διανομής και συντήρησης και ταυτόχρονα να ελαχιστοποιούνται τα κόστη του συστήματος για τον τελικό χρήστη.

Σημειώνεται ότι με τη χρήση των υπηρεσιών ΛωΥ ο πελάτης δεν χρειάζεται να δημιουργεί αντίγραφα ασφάλειας καθώς και αυτό το αναλαμβάνουν οι υπεύθυνοι στα κέντρα δεδομένων. Επίσης, ο χρήστης δεν αγοράζει το λογισμικό και την άδεια χρήσης αυτού, όπως γίνεται με την παραδοσιακή μέθοδο διάθεσης λογισμικού, αλλά πληρώνει μέσω μιας συνδρομής, με βάση τη χρήση που κάνει. Σ' έναν μικρό αριθμό περιπτώσεων, μπορεί να καταβάλλεται μια σταθερή αμοιβή (πάγια χρέωση), και συχνά συνδέεται με μια αμοιβή συντήρησης.

Οι εφαρμογές ΛωΥ έχουν χαμηλό κόστος αρχικής επένδυσης, συγκριτικά με τις τοπικά εγκαθιστώμενες «παραδοσιακές» εφαρμογές. Αθροιστικά, το επαναλαμβανόμενο κόστος για τη χρήση του λογισμικού είναι μικρότερο σε σχέση με την αγορά αδειών λογισμικού. Όλοι οι χρήστες εκτελούν την ίδια έκδοση λογισμικού, με αποτέλεσμα το λογισμικό του καθενός χρήστη να είναι συμβατό με του άλλου. Το ΛωΥ ακόμη έχει τη δυνατότητα να υποστηρίζει πολλαπλούς χρήστες και να παρέχει ένα κοινό μοντέλο δεδομένων μέσω ενός μοναδικού μοντέλου - στιγμιότυπου. Υπάρχει βέβαια η εναλλακτική λύση εικονικού λογισμικού ξεχωριστών στιγμιότυπων, αλλά δεν χρησιμοποιείται συχνά.





*Εικόνα 19 - Λογισμικό ως υπηρεσία (ΛωΥ)*

### **3.4 Ταυτότητα ως υπηρεσία (ΤωΥ)**

Η αναγνώριση και η διακρίβωση μιας ταυτότητας είναι μια κεντρική λειτουργία των δικτύων. Μία υπηρεσία ταυτότητας αποθηκεύει όλες εκείνες τις πληροφορίες που σχετίζονται με μια ψηφιακή οντότητα, μπορεί να απαντά σε ερωτήματα σε σχέση με αυτές και επίσης διαχειρίζεται τις ηλεκτρονικές συναλλαγές.

Οι υπηρεσίες ταυτότητας έχουν ως κεντρικές λειτουργίες τις εξής παρακάτω:

1. Την αποθήκευση δεδομένων,
2. Μια μηχανή ερώτησης και
3. Μια μηχανή εφαρμογής πολιτικής, η οποία έχει ως στόχο να διατηρεί την ακεραιότητα των δεδομένων.

Σημειώνεται ότι τα κατακεμημένα συστήματα συναλλαγών, όπως τα εταιρικά δίκτυα ή τα συστήματα υπολογιστικού νέφους, αυξάνουν τις δυσκολίες που αντιμετωπίζουν τα συστήματα διαχείρισης ταυτότητας με την έκθεση μιας μεγαλύτερης επιφάνειας σε έναν εισβολέα σε σχέση με ένα ιδιωτικό δίκτυο.

Κεντρική αρχή του σχεδιασμού των ασφαλών δικτύων είναι η υποστήριξη της απόρρητης πρόσβασης σε πόρους, η εξουσιοδοτημένη χρήση καθορισμένων δικαιωμάτων και προνομίων και η επικυρωμένη έγκριση από ένα αντικείμενο βασισμένο στην ταυτότητά του. Ως αντιμετώπιση αυτού του προβλήματος, η εγκατάσταση ενός μηχανισμού διακρίβωσης ταυτότητας μπορεί να θεωρηθεί το κλειδί για την ασφαλή λήψη αποφάσεων οποτεδήποτε χρειάζεται ένα αντικείμενο ή μια οντότητα να αποδείξει την ιδιοκτησία σε πόρους ή το δικαίωμα χρήσης τους.

Οι υπηρεσίες που παρέχουν διαχείριση ψηφιακής ταυτότητας ως υπηρεσία, αποτελούσαν μέρος των συστημάτων υπερ-δικτύων. Η υπηρεσία περιοχών ονοματολογίας (Domain Name Service - DNS) μπορεί να εκτελείται σε ένα ιδιωτικό δίκτυο, αλλά βρίσκεται στη καρδιά του Διαδικτύου ως υπηρεσία που παρέχει διακρίβωση ταυτότητας. Οι εξυπηρετές ονοματολογίας που καλύπτουν τα διάφορα domains του Διαδικτύου (.COM, .ORG, .EDU, .MIL, .TV, .RU, κλπ) είναι εξυπηρετές ΤωΥ. Το DNS καθιερώνει την ταυτότητα ενός Domain καθώς εντάσσει σε αυτό ένα σύνολο καθορισμένων διευθύνσεων, που συνδέονται με έναν ιδιοκτήτη. Εάν ο προσδιορισμός είναι ο καθορισμένος αριθμός IP διευθύνσεων (IP address - Internet Protocol address), οι υπόλοιπες ιδιότητες είναι τα μεταδεδομένα της.

Πολλές από τις υπηρεσίες οι οποίες εκτελούνται σ' ένα νέφος μπορούν να κατηγοριοποιηθούν ως ΤωΥ. Εντούτοις, οι περισσότεροι εμπειρογνώμονες στον τομέα ΤωΥ, όταν καθορίζουν μια υπηρεσία ταυτότητας χρησιμοποιούν πιο αυστηρούς ορισμούς. Έτσι έχουμε ως αποτέλεσμα, η υπηρεσία να πρέπει να λειτουργήσει ως συστατικό σύμφωνα με τους κανόνες της υπηρεσιοστρεφούς αρχιτεκτονικής. Αυτός ο πιο αυστηρός ορισμός, περιορίζει το ΤωΥ στις νεώτερες υπηρεσίες λογισμικού.

Γενικά μια ταυτότητα ορίζεται ως ένα σύνολο χαρακτηριστικών - γνωρισμάτων που καθιστούν κάτι αναγνωρίσιμο ή γνωστό. Στα συστήματα δικτύων υπολογιστών, αυτό το σύνολο αποτελεί η ψηφιακή ταυτότητα κάποιου ατόμου.

Η ψηφιακή ταυτότητα είναι οι ιδιότητες και τα μεταδεδομένα ενός αντικείμενου μαζί με ένα σύνολο σχέσεων με άλλα αντικείμενα που το καθιστούν αναγνωρίσιμο. Αυτό δεν σημαίνει ότι όλα τα αντικείμενα είναι μοναδικά, αλλά εξ ορισμού μια ψηφιακή ταυτότητα πρέπει να είναι μοναδική, μέσω της ανάθεσης ενός μοναδικού γνωρίσματος.

Οι βάσεις δεδομένων αποθηκεύουν τις πληροφορίες και τις σχέσεις σε πίνακες, γραμμές και στήλες. Η ταυτότητα των πληροφοριών που αποθηκεύεται κατ' αυτό τον τρόπο είναι σύμφωνη με την έννοια μιας οντότητας και οι αρχιτέκτονες των βάσεων δεδομένων προσπαθούν πάντα να βρουν τον βέλτιστο τρόπο για τη μείωση του συνόλου των δεδομένων τους σε ένα βασικό σύνολο από ταυτότητες.

Όταν μία ταυτότητα ανήκει σε ένα πρόσωπο, μπορεί να περιλαμβάνει τα εξής:

- **Πράγματα που είμαστε:** Βιολογικά χαρακτηριστικά όπως η ηλικία, το φύλο, το γένος, η εμφάνιση, κ.τ.λ.
- **Πράγματα που γνωρίζουμε:** Βιογραφία, προσωπικά στοιχεία όπως οι αριθμοί ασφάλειας, τους προσωπικούς κωδικούς ασφάλειας - PINs, το μέρος που πήγαμε σχολείο, κ.τ.λ.
- **Πράγματα που έχουμε:** Τα δακτυλικά αποτυπώματα, ένα τραπεζικός λογαριασμός στον οποίο μπορούμε να έχουμε πρόσβαση, ένα κλειδί ασφάλειας που μας δόθηκε, αντικείμενα που κατέχουμε, κ.τ.λ.
- **Πράγματα που σχετιζόμαστε:** Η οικογένεια και οι φίλοι μας, μια άδεια λογισμικού, πεποιθήσεις και αξίες, δραστηριότητες, προσωπικές πεποιθήσεις και επιλογές, ένα λογαριασμό iGoogle, κ.τ.λ.

### 3.5 Συμμόρφωση ως υπηρεσία (ΣωΥ)

Σε ό,τι αφορά το υπολογιστικό νέφος, αυτό διαθέτει από τη φύση του διάφορες αρμοδιότητες. Οι νόμοι της χώρας στην οποία γίνεται η αίτηση μπορεί να διαφέρουν με

τους νόμους της χώρας όπου το αίτημα υποβάλλεται σε επεξεργασία και είναι πιθανό να διαφέρουν και με τους νόμους της χώρας όπου η υπηρεσία παρέχεται.

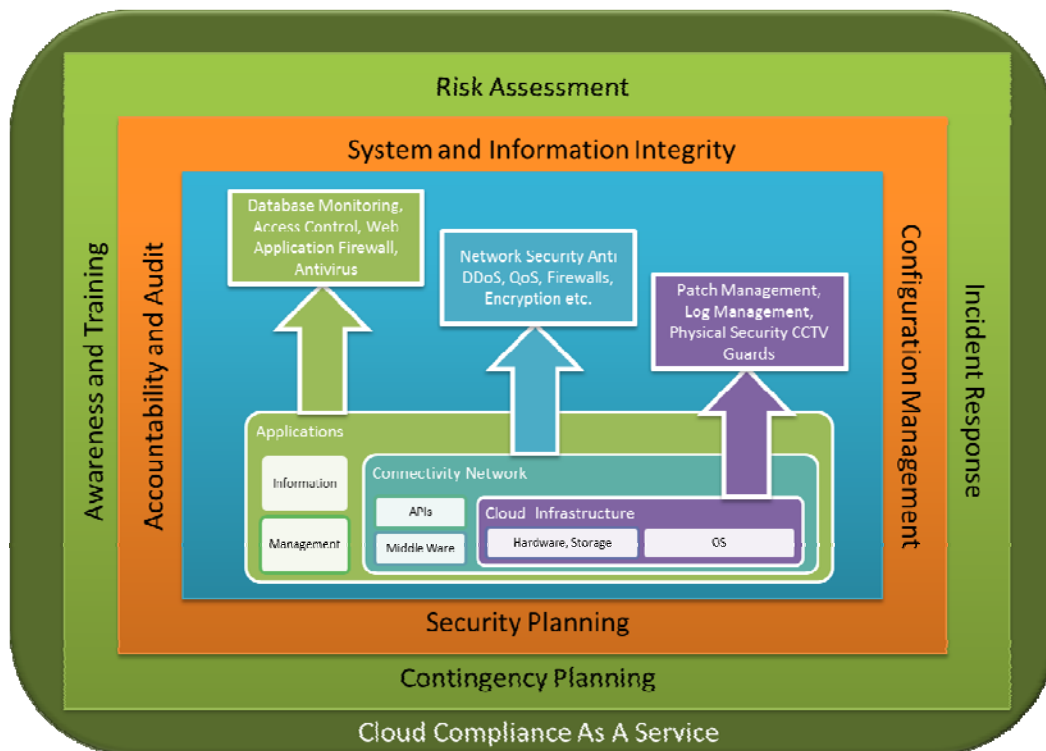
Η συμμόρφωση είναι κάτι πολύ περισσότερο από απλά μια παρεχόμενη υπηρεσία, έτσι ώστε κάποιος να μπορεί να έχει πρόσβαση σε έναν πόρο. Η συμμόρφωση είναι ένα σύνθετο ζήτημα το οποίο απαιτεί ιδιαίτερη πείρα.

Μια εφαρμογή *Συμμόρφωσης ως Υπηρεσία* (Compliance as a Service - CaaS) χρειάζεται για να διαδραματίσει τον ρόλο της *τρίτης έμπιστης οντότητας*. Τα ΣωΥ πρέπει να δομηθούν σε μια Προσανατολισμένη στις Υπηρεσίες Αρχιτεκτονική (Service Oriented Architecture-SOA) προκειμένου να είναι έμπιστα.

Ένα ΣωΥ πρέπει να παρέχει τα ακόλουθα:

- Να διαχειριστεί τις σχέσεις νέφους,
- Να καταλαβαίνει τις πολιτικές ασφάλειας και τις διαδικασίες,
- Να γνωρίζει πως να χειριστεί τις πληροφορίες και τη διαχείριση της ιδιωτικότητας,
- Να τηρεί αρχειοθέτηση,
- Να επιτρέπει στο σύστημα να υποβάλλονται ερωτήματα,
- Να μπορούν όλα τα επίπεδα να συμπεριληφθούν σε μία συμφωνία επιπέδου υπηρεσίας (Service Level Agreement).

Το ΣωΥ μπορεί να αποτελέσει μια υπηρεσία μεγάλης προστιθέμενης αξίας για έναν οργανισμό,. Προκειμένου να εφαρμοστεί το ΣωΥ, μερικές επιχειρήσεις οργανώνονται σε μια δομή που συνήθως αναφέρεται ως "κάθετα νέφη", τα οποία είναι νέφη που ειδικεύονται στη κάθετη αγορά.



Εικόνα 20 - Συμμόρφωση ως υπηρεσία (ΣωΥ)

### 3.6 Υλικό ως υπηρεσία (HaaS)

Όσον αφορά το υλικό ως υπηρεσία (HaaS - Hardware as a Service), αποτελεί μια διαφορετική μορφή υπηρεσίας η οποία είναι διαθέσιμη στους χρήστες. Η συγκεκριμένη υπηρεσία εν αντιθέσει με τις υπόλοιπες προσφέρει στον χρήστη το υλικό εκείνο, προκειμένου να έχει την δυνατότητα αποθήκευσης δεδομένων σε αυτό.

Επιπλέον, γίνεται γνωστό το ότι οι πόροι της συγκεκριμένης υπηρεσίας χρησιμοποιούν και το μοντέλο πολλαπλής μίσθωσης. Ακόμη, στην υπηρεσία αυτή ο χρήστης χρεώνεται ανάλογα με τους υπολογιστικούς πόρους που καταναλώνει. Οι πόροι αυτοί μπορεί να είναι ο εξοπλισμός του δικτύου, η μνήμη, ο χώρος αποθήκευσης, η ΚΜΕ κ.ά. Τέλος, παρέχεται στους χρήστες η συνδεσιμότητα με το διαδίκτυο, όπου τους επιτρέπεται να αποκτούν πρόσβαση στο υλικό από τις επιχειρήσεις τους, καθώς επίσης και ένα περιβάλλον εικονικών πλατφορμών.

## Κεφάλαιο 4: Πλατφόρμες για το υπολογιστικό νέφος

Σε αυτό το κεφάλαιο, θα περιγραφούν οι σημαντικότερες δημόσιες πλατφόρμες νέφους από το Amazon, τη Google και τη Microsoft, καθώς και οι υπηρεσίες που παρέχονται από τη σκοπιά του τελικού χρήστη [5].

Μια πλατφόρμα στο νέφος, είναι ένα στρώμα λογισμικού που χρησιμοποιείται για να δημιουργηθούν πάνω από αυτό υπηρεσίες υψηλού επιπέδου. Διαφορετικοί πάροχοι υπηρεσιών ΠωΥ προσφέρουν υπηρεσίες με διαφορετικές δυνατότητες, που παρέχονται από τα στελέχη ανάπτυξης.

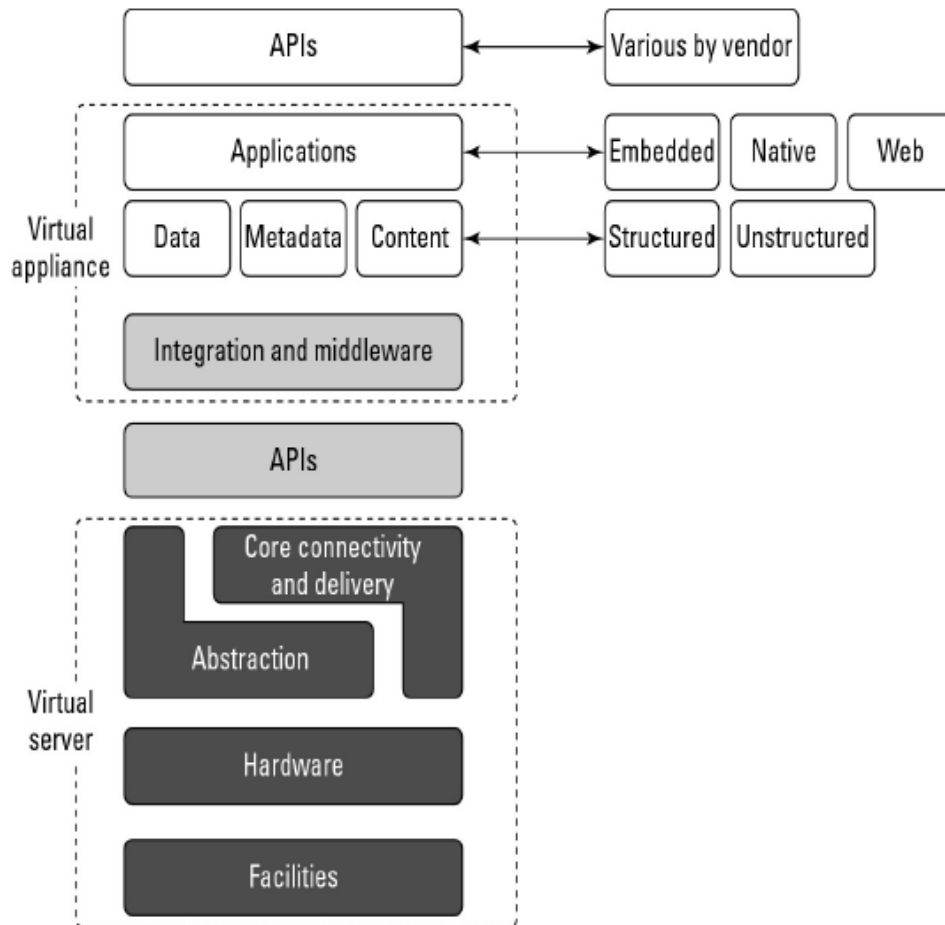
Τρία από τα σημαντικότερα παραδείγματα του ΠωΥ είναι τα ακόλουθα:

- Salesforce.com's Force.com Platform
- Windows Azure Platform
- Google Apps και Google App Engine

Αυτές οι τρεις υπηρεσίες προσφέρουν το υλικό και το λογισμικό που απαιτείται για να κατασκευαστούν και να αναπτυχθούν εφαρμογές Ιστού ή υπηρεσίες οι οποίες έχουν δημιουργηθεί από έναν υπεύθυνο μέσα στο πλαίσιο και το εύρος των ικανοτήτων που επιτρέπει η πλατφόρμα.

Γενικά, οι πλατφόρμες ενσωματώνουν όλη τη στοίβα λογισμικού νεφών, εκτός από το επίπεδο παρουσίασης που αναπαριστά τη διεπαφή του χρήστη. Αυτό που διαχωρίζει μια πλατφόρμα από μια εικονική συσκευή (βλ. Εικόνα 21) είναι ότι το λογισμικό που είναι εγκατεστημένο, κατασκευάζεται από συστατικά και υπηρεσίες κι ελέγχεται μέσω της προγραμματιστικής διασύνδεσης εφαρμογών API (Application Programming Interface), που δημοσιεύει ο φορέας παροχής πλατφορμών.

Ακόμη έχει σημασία για τους προμηθευτές λειτουργικών συστημάτων να ωθήσουν τα περιβάλλοντα ανάπτυξης τους στο νέφος, με τις ίδιες τεχνολογίες που έχουν χρησιμοποιηθεί επιτυχώς για τη δημιουργία εφαρμογών Διαδικτύου.



**Εικόνα 21 - Μία εικονική συσκευή είναι ένα λογισμικό που εγκαθίσταται ως ενδιάμεσο λογισμικό σε μια εικονική μηχανή**

Κατά συνέπεια, μπορεί να βρεθεί μια πλατφόρμα βασισμένη σε μια εικονική μηχανή Sun XVM, που περιλαμβάνει ένα περιβάλλον Net Beans IDE (ολοκληρωμένο περιβάλλον ανάπτυξης - Integrated Development Environment) και που υποστηρίζει τον εξυπηρέτη εφαρμογών ιστού Sun Glassfish Web Server με χρήση Perl ή Ruby.

Στη συνέχεια όσον αφορά τα Windows, η Microsoft ενδιαφέρεται ομοίως για την παροχή μιας πλατφόρμας η οποία θα επιτρέπει στους χρήστες της, στην προσπάθειά τους να θέσουν σε λειτουργία πλατφόρμες με Windows, να τα εκτελέσουν σε μια Hyper-V VM εικονική μηχανή (Virtual Machine) με χρήση του πλαισίου εργασίας ASP.NET.

Οι πλατφόρμες συνοδεύονται συχνά από εργαλεία με σκοπό την ενίσχυση του σχεδίου και της ανάπτυξης της εφαρμογής. Ανάλογα με τον πάροχο, υπάρχουν συγκεκριμένα εργαλεία, τα οποία διαχωρίζονται στα ακόλουθα:

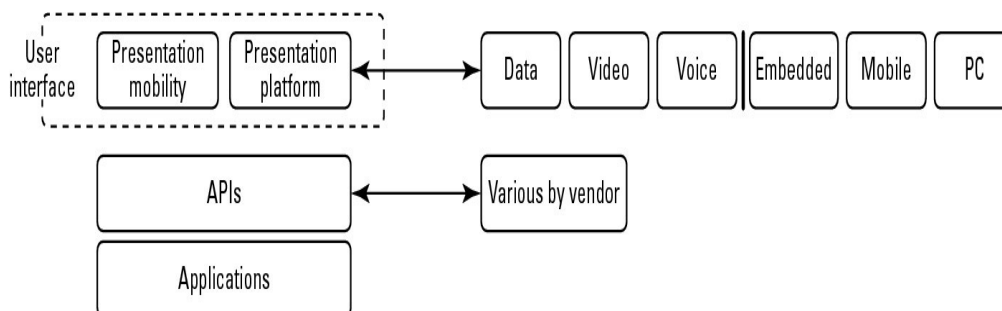
- για τη συνεργασία ομάδων,
- εξεταστικά,
- για την οργάνωση και μέτρηση της απόδοσης του προγράμματος και των ιδιοτήτων,

- για τις βάσεις δεδομένων και της ολοκλήρωσης υπηρεσιών Ιστού, καθώς και
- αποθήκευσης

Ακριβώς όπως μια εικονική συσκευή μπορεί να είναι προσπελάσιμη από τον μέσω της προγραμματιστικής διασύνδεσης εφαρμογών API (Application Programming Interface), έτσι και μια εφαρμογή που είναι ενσωματωμένη στο νέφος, με χρήση μιας υπηρεσίας πλατφόρμας θα μπορεί να εκθέσει/παρουσιάσει την υπηρεσία αυτή, μέσω της προγραμματιστικής διασύνδεσης εφαρμογών της, εν ολίγοις μέσω του API της. Οι χρήστες θα αλληλεπιδρούν έπειτα με την πλατφόρμα, καταναλώνοντας υπηρεσίες μέσω αυτής της προγραμματιστικής διασύνδεσης εφαρμογών, αφήνοντας στην πλατφόρμα την κατάλληλη διαχείριση της υπηρεσίας.

Επιπλέον, αρκετές πλατφόρμες παρέχουν εργαλεία ανάπτυξης της διεπαφής του χρήστη βασισμένα στη γλώσσα μορφοποίησης υπερκειμένου (Hypertext Markup Language-HTML), JavaScript, ή κάποια άλλη τεχνολογία. Δεδομένου ότι ο Ιστός γίνεται περισσότερο προσανατολισμένος στα πολυμέσα (media), αρκετά στελέχη ανάπτυξης εφαρμογών έχουν επιλέξει να εργαστούν με πλουσιότερα περιβάλλοντα διαδικτύου όπως είναι τα: Adobe Flash, Flex, Air ή ακόμη και με άλλες εναλλακτικές λύσεις όπως το Windows Silverlight.

Η Εικόνα 22 παρουσιάζει τη κορυφή της στοίβας της υπολογιστικής νεφών, η οποία περιλαμβάνει την προγραμματιστική διασύνδεση εφαρμογών (API) και τη λειτουργία παρουσίασης.



**Εικόνα 22 - Η κορυφή της διεπαφής της υπολογιστικής νεφών περιλαμβάνει τη διεπαφή του χρήστη και το API για το επίπεδο της εφαρμογής**

Η προγραμματιστική διασύνδεση εφαρμογών (API), είναι μια από τις βασικές διαφορές που διαχωρίζουν την υπολογιστική νέφος από τα παραδοσιακά μοντέλα των εφαρμογών Διαδικτύου, διότι είναι το μέσο για τους πόρους που απαιτούνται για την υποστήριξη των εφαρμογών.

Μία προγραμματιστική διασύνδεση εφαρμογών (ένα API) μπορεί να ελέγξει τη ροή δεδομένων, τις επικοινωνίες και άλλες σημαντικές πτυχές της εφαρμογής νέφος. Δυστυχώς, κάθε πάροχος νέφος έχει τη δική του προγραμματιστική διασύνδεση εφαρμογών νέφος (το δικό του API νέφος), συνεπώς δεν υπάρχει καμία τυποποίηση.

## 4.1 Amazon EC2

Το νέφος της Amazon, παρέχει την υποδομή ως υπηρεσία (infrastructure as a service, IaaS), με την οποία η υπολογιστική υποδομή (των εξυπηρετών, του δικτύου κ.ά.) ρυθμίζεται μέσω μιας αυτοματοποιημένης κονσόλας διαχείρισης βασισμένης στο Διαδίκτυο. Αυτή η υπηρεσία ΥωΥ, που καλείται Elastic Compute Cloud ή EC2, είναι μόνο μια από ένα μεγάλο σύνολο υπηρεσιών που αποτελούν τη πλατφόρμα νέφους του Amazon. Επίσης, ο όρος EC2 χρησιμοποιείται συχνά για να περιγράψει ολόκληρη τη προσφορά νεφών.

Κάποια από τα σημαντικά χαρακτηριστικά της Amazon EC2 είναι τα εξής:

- **Amazon Elastic Compute Cloud (EC2):** Παροχή πρόσβασης των χρηστών σε εικονικές μηχανές με την επιθυμητή χωρητικότητα.
- **Amazon Simple DB:** Μη σχεσιακή βάση δεδομένων.
- **Amazon Elastic Block Storage (EBS):** Διατήρηση δεδομένων σε ένα σύνολο φυσικών εξυπηρετών.
- **Amazon Simple Queue Service (SQS):** Αξιόπιστη και επεκτάσιμη ουρά μηνυμάτων.
- **Amazon Simple Storage Service (S<sup>3</sup>):** Αποθηκευτικός χώρος δεδομένων σε μορφή αρχείων.
- **Elastic Load balancing:** Υπηρεσία ελαστικής εξισορρόπησης φορτίου.
- **Amazon Cloud Watch:** Παροχή παρακολούθησης πόρων του νέφους.
- **Amazon Auto Scale:** Επέκταση δυναμικότητας του Amazon EC2.
- **Amazon Virtual Private cloud (VPC):** Οι εικονικοί εξυπηρετές μπορούν να συνδεθούν στο εσωτερικό δίκτυο μιας επιχείρησης.

Σε αυτό το σημείο επεξηγώντας την υπηρεσία Elastic Compute Cloud, είναι αναγκαίο να σημειωθεί το ότι παρέχει στους χρήστες πρόσβαση σε εικονικές μηχανές και διάφορους υπολογιστικούς πόρους, χάρη στα οποία έχουν τη δυνατότητα να προσαρμόζουν δυναμικά τη χωρητικότητά τους. Οι μηχανές αυτές βρίσκονται σε φυσικούς εξυπηρετές, με τις λεπτομέρειες του πραγματικού φυσικού εξυπηρετή, όπως η θέση του, η χωρητικότητά του κ.λπ. να αποκρύπτονται από τον τελικό χρήστη.

Επιπροσθέτως, οι χρήστες στην υπηρεσία αυτή όταν «προμηθεύονται» έναν εξυπηρετή, μπορούν να διαλέξουν κάποιους ήδη διαμορφωμένους εικονικούς εξυπηρετές ή ακόμη και να δημιουργήσουν τις δικές τους Εικόνες Μηχανής της Amazon (Amazon Machine Images-AMIs), όπου περιέχουν διάφορες εφαρμογές, λειτουργικά συστήματα και δεδομένα. Οι Εικόνες Μηχανής της Amazon (AMIs), εξαιτίας του ότι περιέχουν διάφορες επιλογές ρυθμίσεων, χρησιμοποιούνται από τους χρήστες με σκοπό να προσδιοριστεί η επιθυμητή χρήση του εικονικού εξυπηρετή και αποθηκεύονται σε ένα κοινό αποθηκευτικό χώρο της λειτουργίας αποθήκευσης Amazon S<sup>3</sup>.

Επίσης, χάρη στη χρήση και λειτουργία του Amazon EC2, ο κάθε εξυπηρετούμενος διαθέτει τη δυνατότητα αύξησης ή μείωσης των προσφερόμενων υπολογιστικών του πόρων μέσα σε ένα μικρό χρονικό διάστημα λίγων λεπτών. Στην υπηρεσία αυτή ο



λογαριασμός του χρήστη χρεώνεται σε ωριαία βάση σύμφωνα με το χρόνο που χρησιμοποιήσει τον εξυπηρέτη και ανάλογα με τη χωρητικότητα αυτού. Αξίζει να σημειωθεί πως ένας μικρός εξυπηρέτης Linux κοστίζει μερικά σεντς ανά ΚΜΕ-ώρα (CPU, Central Processing Unit), δηλαδή ανά ώρα που επεξεργάζεται η κεντρική μονάδα επεξεργασίας, ενώ ένας μεγαλύτερος εξυπηρέτης που έχει φορτωθεί εκ των προτέρων με λογισμικό που έχει άδεια (license) όπως τα Windows και με άλλα προϊόντα βάσεων δεδομένων ή λογισμικά, θα μπορούσε να κοστίζει περίπου ένα δολάριο ανά ώρα.

Ένα ακόμη χαρακτηριστικό της υπηρεσίας αυτής, είναι ότι οι χρήστες έχουν πρόσβαση στους εξυπηρέτες, με το ρόλο του διαχειριστή. Έχοντας σαν αποτέλεσμα με αυτό τον τρόπο να τους ελέγχουν καθολικά, να μπορούν να κάνουν επέκταση των εφαρμογών, καθώς επίσης και να τις καταστήσουν δημόσια προσιτές μέσω του Διαδικτύου, χρησιμοποιώντας στατικές διευθύνσεις IP, οι οποίες χρεώνονται σε μηνιαία βάση.

Σε αυτό το σημείο πρέπει να αναφερθεί πως η μεταφορά δεδομένων μέσω του δικτύου σε οποιοδήποτε εξυπηρέτη, με ή χωρίς μια στατική IP διεύθυνση, χρεώνεται με βάση τη χρήση, σε ποσοστά μερικών σεντς ανά gigabyte που μεταφέρονται.

Επιπλέον, αξίζει να αναφερθεί το ότι η Amazon προσφέρει επίσης στους χρήστες την υπηρεσία Simple DB. Η υπηρεσία αυτή παρέχει έναν αποθηκευτικό χώρο όπου μπορούν να αποθηκευτούν ακόμη και να ανακτηθούν ζευγάρια κλειδιών. Επίσης μέσω αυτής προσφέρει έτοιμες συναρτήσεις στους χρήστες, αλλά και υποδομή. Η Simple DB αποτελεί μια μη σχεσιακή βάση δεδομένων. Αναλυτικότερα, στη βάση αυτή κάθε στοιχείο μπορεί να αποτελείται από διάφορα ζευγάρια χαρακτηριστικών γνωρισμάτων. Επίσης, διαφορετικά στοιχεία σε μια περιοχή μπορούν να έχουν διαφορετικά σύνολα ιδιοτήτων. Ακόμη, τα ερωτήματα στην υπηρεσία αυτή μπορούν να συμπεριλάβουν σε οποιοδήποτε αριθμό συνθηκών πάνω σε ιδιότητες, συμπεριλαμβανομένων των συνθηκών ανισότητας. Ένα ακόμα χαρακτηριστικό της Simple DB είναι πως δεν υποστηρίζει σχέσεις αντικειμένων καθώς και συναλλαγές, όπως και δεν επιτρέπει τα joins (συνδέσεις) μεταξύ των στοιχείων της βάσης. Η υπηρεσία Simple DB, δίνει τη δυνατότητα στα δεδομένα να διαθέτουν αντίγραφα. Για τον λόγο αυτό χαρακτηρίζει ένα μοντέλο *ενδεχόμενης συνέπειας* (eventual consistency), όπου οι ενημερώσεις διαδίδονται σε τουλάχιστον ένα αντίγραφο και τελικά φτάνουν όλα τα αντίγραφα, αν και με κάποια καθυστέρηση. Υπάρχει περίπτωση όμως αυτό να οδηγήσει σε αντιληπτή ασυνέπεια του μοντέλου αυτού, μιας κι ένα read που ακολουθεί ένα write μπορεί να μην παράγει πάντα το σωστό αποτέλεσμα.

Σημειώνεται εδώ ότι όταν διακόπτεται η επικοινωνία με τον εξυπηρέτη, οποιαδήποτε δεδομένα χρηστών υπάρχουν στο σύστημα αρχείων, δηλαδή ότι δεν είναι μέρος των Εικόνων Μηχανής της Amazon (AMIs), αυτομάτως χάνεται. Αυτό συμβαίνει γιατί οι εικονικοί εξυπηρέτες δεν έχουν μόνιμο χώρο αποθήκευσης. Προκειμένου λοιπόν τα γεγονότα να αποθηκεύονται συνεχώς, όπως σε μια σχεσιακή βάση δεδομένων, θα πρέπει να γίνει τοποθέτηση σε έναν εικονικό εξυπηρέτη της υπηρεσία Elastic Block Storage (EBS). Η υπηρεσία αυτή προσφέρει έναν σταθερό χώρο αποθήκευσης και διατήρησης των δεδομένων των χρηστών σε ένα μεγάλο σύνολο φυσικών εξυπηρετών. Αφού γίνει η εκκίνηση ενός εικονικού εξυπηρέτη, τα δεδομένα των χρηστών πρέπει να συνδεθούν από την υπηρεσία EBS ως λογικός χώρος αποθήκευσης που τοποθετείται ως εξωτερική συσκευή. Ο χώρος αποθήκευσης της υπηρεσίας EBS μπορεί να προσεγγιστεί μόνο εάν είναι συνδεδεμένη μ' έναν εικονικό εξυπηρέτη που εκτελείται. Περαιτέρω, οποιοδήποτε

άλλοι εξυπηρέτες μπορούν να έχουν πρόσβαση σε αυτά τα δεδομένα μόνο μέσω του εξυπηρέτη όπου βρίσκεται η EBS.

Η Amazon Simple Queue Service (SQS) αποτελεί μια αξιόπιστη και επεκτάσιμη ουρά μηνυμάτων, που προσφέρεται με σκοπό την προσωρινή αποθήκευση μηνυμάτων και δεδομένων που μεταφέρονται μεταξύ των εξυπηρετών. Βέβαια, στην υπηρεσία αυτή δε χρειάζεται ο κάθε εξυπηρέτης να είναι συνεχώς διαθέσιμος. Άρα, κάθε εξυπηρέτης μπορεί να τεθεί σε λειτουργία, μόνο όταν πρέπει να επεξεργαστεί τα μηνύματα που βρίσκονται στην υπηρεσία SQS και εν συνεχεία να τα καταχωρήσει στη δική του βάση δεδομένων.

Η Amazon προσφέρει μια ακόμα διαφορετική υπηρεσία αποθήκευσης, την Amazon Simple Storage Service (S3). Πρόκειται για μια δικτυακή υπηρεσία αποθήκευσης. Η Amazon S3 δίνει τη δυνατότητα αποθήκευσης δεδομένων σε μορφή αρχείων οποιουδήποτε τύπου και αντικείμενου. Επίσης, είναι ιδιαίτερα χρήσιμη για την αποθήκευση μεγάλων συλλογών με μη δομημένα δεδομένα, τα οποία πρέπει να προσεγγιστούν από πολλές εφαρμογές εξυπηρευόμενων. Έπειτα, στην υπηρεσία αυτή οι χρήστες μπορούν να έχουν πρόσβαση στα αντικείμενα της υπηρεσίας S3, οποιαδήποτε στιγμή και από οπουδήποτε στο Διαδίκτυο.

Ένας ακόμη τρόπος προσπέλασης των αντικειμένων, εφαρμόζεται μέσω των εικονικών εξυπηρετών που βρίσκονται στο νέφος της Amazon. Το μοντέλο αποθήκευσης S3 παρέχει επίσης συνέπεια στα αντίγραφα, μιας και σε περίπτωση που κάποια δεδομένα δεν έχουν διαδοθεί σε όλα τα αντίγραφα, τότε κάποιοι πελάτες θα μπορούν να διαβάσουν ακόμα τα παλιά δεδομένα.

Η υπηρεσία αυτή χρησιμοποιείται επίσης με σκοπό την αποθήκευση των εικόνων μηχανής της Amazon (AMIs). Οι εικόνες αυτού του τύπου, καθορίζονται από μέρος κάθε χρήστη με τους ακόλουθους τρόπους:

- Είτε από την αρχή με το λειτουργικό σύστημα, καθώς και τα αρχεία της εφαρμογής από τους δικούς τους εξυπηρέτες.
- Είτε από ένα ήδη διαθέσιμο AMI (Amazon).

Επίσης, τέτοιου είδους εικόνες μπορούν να τεθούν στη διάθεση άλλων χρηστών, και γενικότερα στο κοινό. Περαιτέρω, ένας τέτοιος διαμοιρασμός μπορεί να συνδυαστεί με τη πύλη πληρωμών του Amazon μέσω μιας συμφωνίας DevPay, με το οποίο ένα μέρος των δαπανών που πληρώνεται από τους χρήστες τέτοιων εικόνων μηχανής της Amazon (AMIs) πιστώνεται στο λογαριασμό του δημιουργού αυτών. Κατά συνέπεια ο διαμοιρασμός των εικόνων μηχανής της Amazon (AMIs) με βάση το DevPay, στο S3, έχει δημιουργήσει ένα νέο δίαυλο διανομής λογισμικού, και πολλές βάσεις δεδομένων αλλά και πακέτα λογισμικού, όπως από την Oracle ή την IBM, είναι διαθέσιμα με αυτό τον τρόπο.

Όσον αφορά την υποδομή νέφους Amazon EC2, ο χρήστης χρειάζεται να καθορίσει ρητά μια αρχιτεκτονική η οποία να επιτρέπει την εξελισιμότητα, χρησιμοποιώντας εργαλεία, που παρέχονται από το νέφος Amazon για να διαχειριστεί την ελαστικότητα. Έτσι προκύπτει η υπηρεσία Amazon Cloud Watch η οποία χρησιμοποιείται για την παρακολούθηση της κατάστασης όλων των υπηρεσιών της Amazon. Επίσης, η υπηρεσία αυτή είναι αρμόδια για την παρακολούθηση της κατάστασης όλων των εικονικών

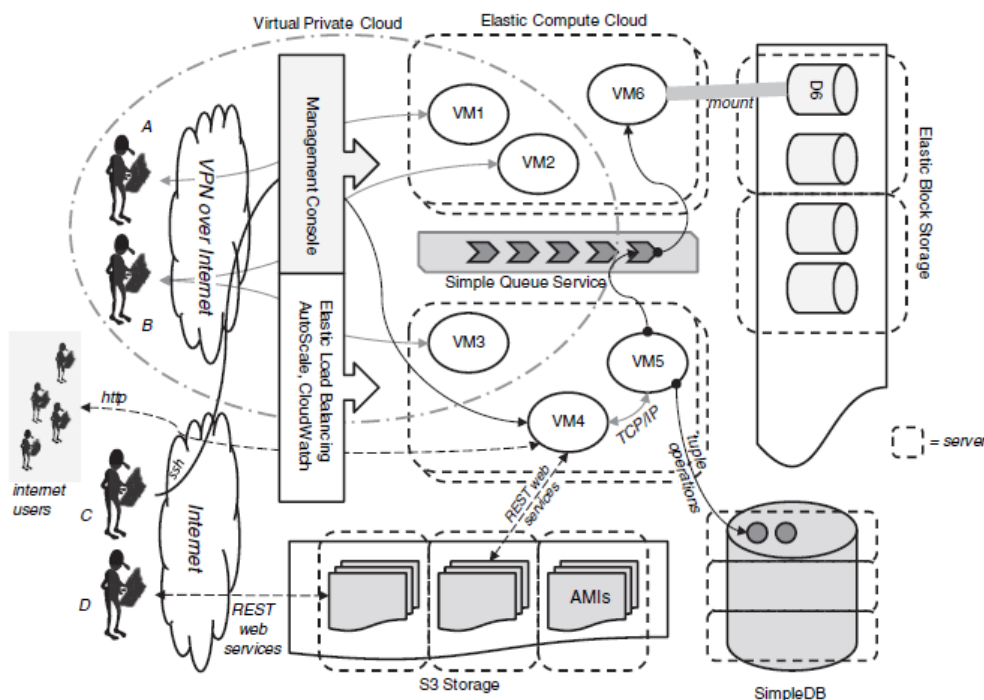
εξυπηρετών ενός χρήστη, όπως η κεντρική μονάδα επεξεργασίας (ΚΜΕ) και η χρησιμοποίηση συσκευών εισόδου/εξόδου (Input/ Output-I/O). Ωστόσο, τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν και από μια άλλη υπηρεσία, την Amazon Auto Scale. Η συγκεκριμένη υπηρεσία μπορεί να προσθέσει είτε να αφαιρέσει, πάντα ανάλογα με τη ζήτηση των χρηστών, εικονικούς εξυπηρετές, έχοντας σκοπό την εξισορρόπηση του φόρτου εργασίας αλλά και την περαιτέρω μείωση του κόστους.

Μια ακόμη υπηρεσία είναι η Elastic Load Balancing, η οποία επιτρέπει σε μια ομάδα εικονικών εξυπηρετών να διαρθρωθούν σ' ένα σύνολο όπου τα εισερχόμενα αιτήματα (π.χ. συνδέσεις HTTP) να κατανέμονται έτσι ώστε να εξισοροπείται ο φόρτος εργασίας. Στη συνέχεια, τα αιτήματα αυτά μπορούν να ελεγχθούν και να χρησιμοποιηθούν από τις δυο προαναφερθείσες υπηρεσίες για τον λόγο που προαναφέρθηκε.

Ακόμη, οι επιχειρήσεις που επιδιώκουν να υιοθετήσουν την υπολογιστική νέφος, χρειάζεται να υπολογίσουν και την ασφάλεια του εταιρικού τους IT προφίλ (Information technology). Η ασφάλεια δικτύων είναι ένα σημαντικό στοιχείο που πρέπει να ληφθεί ιδιαίτερα υπόψη εξ αρχής. Οι υπολογιστικοί πόροι μιας επιχείρησης προστατεύονται συνήθως από παραδοσιακά τείχη προστασίας όπως τα firewalls, από εξυπηρετές αντιπροσώπευσης (proxy servers), συστήματα ανίχνευσης εισβολών κ.ά. Βέβαια, η ασφάλεια μιας επιχείρησης προϋποθέτει να είναι προστατευμένοι με τον ίδιο τρόπο και οι εικονικοί εξυπηρετές που εκτελούνται στο νέφος. Αυτό είναι δυνατόν να επιτευχθεί, κάνοντας χρήση των ίδιων πολιτικών και μέτρων προστασίας που εφαρμόζονται σε οποιουδήποτε πόρους στα κέντρα δεδομένων.

Τέλος, το νέφος Amazon παρέχει μια ακόμα υπηρεσία, την Amazon Virtual Private cloud - (VPC). Μέσω αυτής της συγκεκριμένης υπηρεσίας, οι εικονικοί εξυπηρετές μπορούν να συνδεθούν στο εσωτερικό δίκτυο μιας επιχείρησης χρησιμοποιώντας ένα εικονικό ιδιωτικό δίκτυο (VPN - Virtual Private Network). Οι εξυπηρετές αυτοί έχουν ιδιωτικές διευθύνσεις IP, που ελέγχονται από το κέντρο δικτύων της επιχείρησης. Ωστόσο, η πρόσβαση σε αυτό το δίκτυο, αλλά και από αυτό το δίκτυο προς τον εξωτερικό κόσμο μπορεί να ελέγχεται από το ίδιο σύνολο πολιτικών, παραδοσιακών τειχών προστασίας firewalls και εξυπηρετών αντιπροσώπευσης. Όσον αφορά την αποδοτικότητα, αυτά τα firewalls και οι εξυπηρετές αντιπροσώπευσης, καλό είναι να αναπτύσσονται στους εικονικούς εξυπηρετητές στο νέφος ακόμη κι ενώ διατηρούνται οι ίδιες πολιτικές, μιας και τα τείχη προστασίας καθώς και οι εξυπηρετές διαμεσολάβησης αναπτύσσονται συχνά σε ένα ευρύ δίκτυο της επιχείρησης που κατανέμεται σε πολλές τοποθεσίες.

Στην Εικόνα 23, απεικονίζεται ένα παράδειγμα, στο οποίο φαίνεται η λειτουργία των υπηρεσιών που παρέχονται στην υποδομή νέφος του Amazon EC2 από την πλευρά του χρήστη.



**Εικόνα 23 - Υποδομή νέφους Amazon**

Συγκεκριμένα, οι υπηρεσίες νέφους του Amazon παρουσιάζονται στο σχήμα με διακεκομμένα πλαίσια. Μπορούμε να παρατηρήσουμε ότι ο χρήστης C εφοδιάζει με δεδομένα τον εικονικό εξυπηρέτη VM4, μέσω της κονσόλας διαχείρισης (Management Console). Η συγκεκριμένη εφαρμογή δίνει τη δυνατότητα στους χρήστες να παράγουν ζευγάρια κλειδιών με την υποδομή δημόσιου κλειδιού PKI.

Η πρόσβαση των χρηστών στην κονσόλα διαχείρισης γίνεται:

- Είτε μέσω του Secure Shell (SSH) για έναν εξυπηρέτη Linux
- Είτε μέσω απομακρυσμένης σύνδεσης επιφάνειας εργασίας για ένα εξυπηρέτη Windows

Επίσης, ο εικονικός εξυπηρέτης VM4 είναι προσιτός στους χρήστες του διαδικτύου μέσω του πρωτοκόλλου HTTP (Hypertext Transfer Protocol Secure). Ακόμη, ο χρήστης C εκτός από τον VM4 εφοδιάζει και τους VM5 και VM6. Όμως, εάν ο VM4 είναι ένας εξυπηρέτης δικτύου, τότε ο VM5 μπορεί να είναι ένας εξυπηρέτης βάσης δεδομένων. Άρα γίνεται σαφές, ότι αυτοί οι δύο μπορούν να επικοινωνούν μέσω του εσωτερικού δικτύου του νέφους με χρήση του πρωτοκόλλου επικοινωνίας TCP/IP. Δεδομένου ότι ο VM5 είναι ένας εξυπηρέτης βάσης δεδομένων, πρέπει να είναι σε θέση να αποθηκεύει και να ανακτά δεδομένα μέσω της Simple DB.

Στο ίδιο σχήμα, ο VM6 έχει τοποθετήσει ένα λογικό χώρο αποθήκευσης D6, όπου ενδεχομένως διατηρεί τα αρχειοθετημένα δεδομένα. Επίσης βλέπουμε ότι ο VM5 στέλνει δεδομένα στον VM6 χωρίς χρήση του πρωτοκόλλου επικοινωνίας TCP/IP, αλλά με χρήση της υπηρεσίας Amazon Simple Queue Service (SQS). Κατά συνέπεια, ο VM5 δεν σχετίζεται με την αρχειοθέτηση εκτός από τα logs δεδομένα της SQS κι ο VM6 πρέπει να είναι ανοιχτός μόνο σε περίπτωση ανάγκης. Επιπρόσθετα, οι χρήστες A και B της

παραπάνω εικόνας, προσπελαύνουν τους εικονικούς εξυπηρέτες VM1, VM2 και VM3 μέσω ενός Εικονικού Ιδιωτικού Δικτύου που εκτελείται μέσω του Διαδικτύου.

## 4.2 Google App Engine

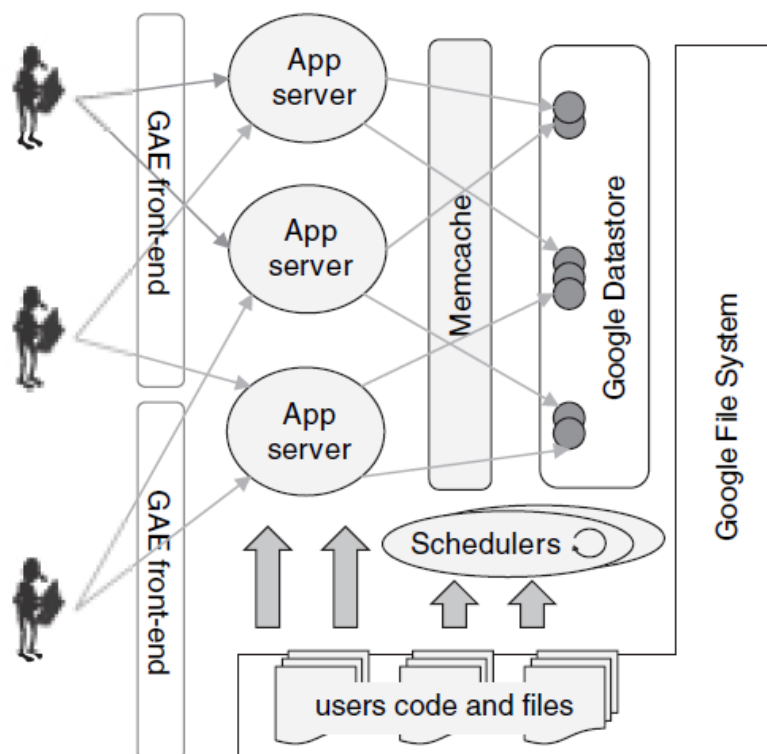
Το νέφος της Google, που αποκαλείται και Google App Engine, είναι μια πλατφόρμα ως υπηρεσία (platform as a service, PaaS). Σε αντίθεση με την υποδομή του Amazon ως νέφος υπηρεσιών, όπου διατίθενται στους χρήστες εικονικές μηχανές τις οποίες ελέγχουν πλήρως, συμπεριλαμβανομένης της εγκατάστασης και της εκτέλεσης του λογισμικού σ' αυτές, μια υπηρεσία ΠωΥ κρύβει το πραγματικό περιβάλλον εκτέλεσης από τους χρήστες. Αντ' αυτού, μια πλατφόρμα λογισμικού παρέχεται μαζί μ' ένα ολοκληρωμένο περιβάλλον (SDK- Software Development Kit), με χρήση της οποίας οι χρήστες αναπτύσσουν εφαρμογές κι ακολουθώντας τις θέτουν σε λειτουργία στο νέφος.

Επίσης, η πλατφόρμα ΠωΥ είναι αρμόδια για την εκτέλεση των εφαρμογών, συμπεριλαμβανομένης της συντήρησης εξωτερικών αιτημάτων υπηρεσιών, καθώς και της εκτέλεσης σχεδιασμένων εργασιών που περιλαμβάνονται στην εφαρμογή. Με το να καταστήσει διαφανείς στο χρήστη τους εξυπηρέτες εκτέλεσης, μια πλατφόρμα ΠωΥ μπορεί να διαμοιράσει τους εξυπηρέτες της εφαρμογής στους χρήστες που χρειάζονται χαμηλότερες χωρητικότητες, καθώς επίσης και να ρυθμίσει αυτόματα τους πόρους που διατίθενται σε εφαρμογές που αντιμετωπίζουν βαρύ φόρτο εργασίας.

Στην Εικόνα 24, απεικονίζεται από τη σκοπιά των χρηστών, η Google App Engine για την οποία γίνεται και λόγος. Οι χρήστες αναπτύσσουν τον κώδικα είτε σε γλώσσα Java είτε σε Python, μαζί με τα σχετικά αρχεία, τα οποία είναι αποθηκευμένα στο Google File System, το οποίο είναι ένα σύστημα αποθήκευσης μεγάλης κλίμακας ανθεκτικό σε σφάλματα. Είναι σημαντικό να σημειωθεί ότι μια εφαρμογή είναι άμεσα διαθέσιμη στο Διαδίκτυο μόλις εισαχθεί επιτυχώς σε αυτό.

Η χρήση των υπολογιστικών πόρων για μια εφαρμογή υπολογίζεται όσον αφορά τις αιτήσεις στο Διαδίκτυο και τις ΚΜΕ-ώρες που ξοδεύονται για την εκτέλεση των αιτήσεων αυτών. Η διαφορά όμως με το πρότυπο ΥωΥ έγκειται στο ότι μια εφαρμογή ΠωΥ μπορεί να επεκταθεί και να γίνει δημόσια διαθέσιμη όλες τις ημέρες και ώρες τις εβδομάδας, αλλά να υπάρχει χρέωση μόνο όταν προσπελαύνεται.

Αντίθετα, στο πρότυπο ΥωΥ επειδή καθίσταται μια εφαρμογή συνεχώς διαθέσιμη, υπάρχει πλήρες κόστος για τουλάχιστον μερικούς από τους εξυπηρέτες που εκτελούνται συνεχώς. Περαιτέρω, η ανάπτυξη των εφαρμογών στη Google App Engine είναι δωρεάν μέσα στα όρια χρήσης της εταιρείας και κατά συνέπεια οι εφαρμογές μπορούν να αναπτυχθούν και να δοκιμαστούν και αυτές δωρεάν. Εν συνεχεία, μπορούν να αρχίσουν να αναλαμβάνουν το κόστος μόνο όταν προσπελαύνονται από έναν επαρκή όγκο αιτήσεων. Το πρότυπο ΠωΥ επιτρέπει στη Google να παρέχει μια τέτοια δωρεάν υπηρεσία επειδή οι εφαρμογές δεν εκτελούνται σε εικονικές μηχανές. Μια εφαρμογή που έχει αναπτυχθεί και δεν προσπελαύνεται απλά, καταναλώνει τον αποθηκευτικό χώρο για τον κώδικα και τα δεδομένα της και δεν χρησιμοποιεί κανέναν κύκλο ΚΜΕ.



**Εικόνα 24 - Google App Engine**

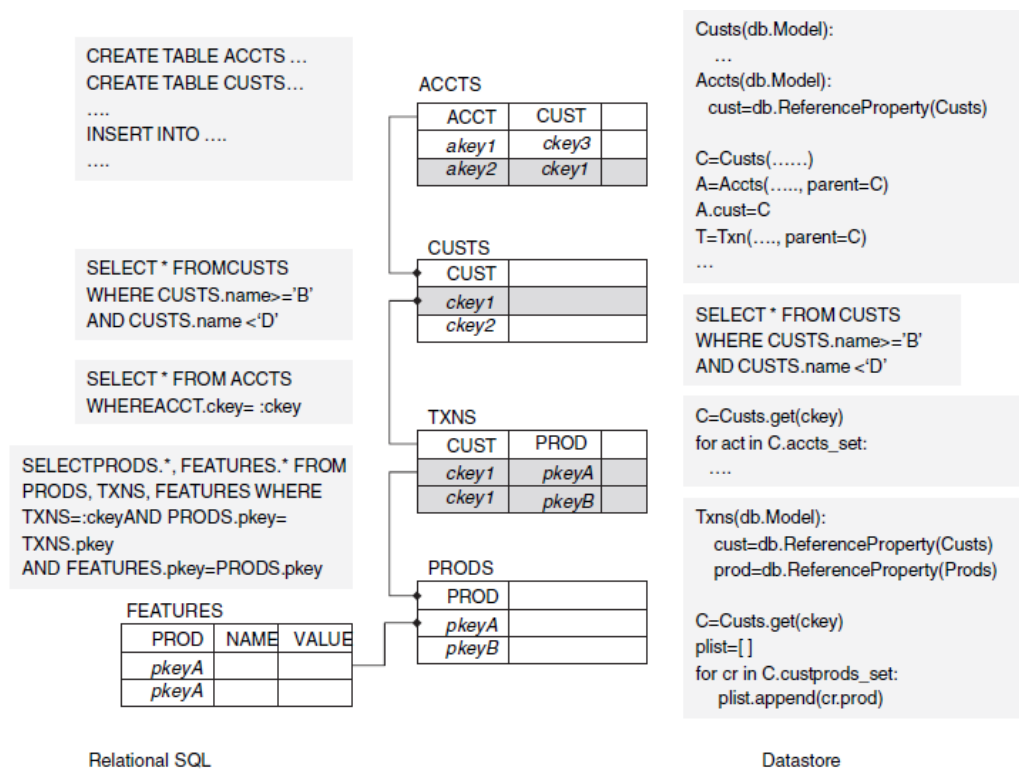
Όσον αφορά τις εφαρμογές της Google App Engine, εξυπηρετούνται από έναν μεγάλο αριθμό εξυπηρετών Διαδικτύου, οι οποίοι βρίσκονται στα κέντρα δεδομένων της Google. Οι εξυπηρετές αυτοί φορτώνουν τον κώδικα από το Google File System στη μνήμη και εξυπηρετούν τις αιτήσεις από τους τελικούς χρήστες. Βέβαια δεν υπάρχει καμία εγγύηση ότι ο ίδιος εξυπηρετής θα υποστηρίξει οποιοσδήποτε αιτήσεις, ακόμη και στην περίπτωση δύο αιτήσεων από την ίδια συνεδρία HTTP. Ακόμα οι εφαρμογές μπορούν επίσης να καθορίσουν μερικές λειτουργίες για να εκτελεστούν από έναν χρονοπρογραμματιστή.

Έτσι ενώ αυτή η αρχιτεκτονική είναι σε θέση να εξασφαλίσει ότι οι εφαρμογές κλιμακώνονται καθώς αυξάνεται ο φόρτος εργασίας, σε αντιδιαστολή ο κώδικας της εφαρμογής δεν μπορεί εύκολα να βασιστεί στα δεδομένα της μνήμης. Συνεπώς, για να αντιμετωπιστεί μερικώς αυτό το ζήτημα, παρέχεται μια καταναμημένη μνήμη που αποκαλείται και Memcache. Πιο συγκεκριμένα, οι συνεδρίες HTTP υλοποιούνται με χρήση της Memcache, έτσι ώστε ακόμα κι αν οι αιτήσεις από την ίδια συνεδρία πηγαίνουν σε διαφορετικούς εξυπηρετές, οι τελευταίες να είναι σε θέση να ανακτήσουν τα δεδομένα συνεδρίας τους για περισσότερο χρονικό διάστημα (δεδομένου ότι η Memcache δεν εγγυάται ότι διατηρεί πάντα τα αποθηκευμένα στη μνήμη, δεδομένα).

#### 4.2.1 Google Datastore

Το Google Datastore είναι μία μη σχεσιακή βάση δεδομένων -όπως η Simple DB του Amazon- στην οποία οι εφαρμογές εναποθέτουν δεδομένα. Το Datastore επιτρέπει στις εφαρμογές να καθορίζουν δομημένους τύπους οι οποίοι καλούνται “kinds”. Επίσης επιτρέπει στις τελευταίες να αποθηκεύουν τα στιγμιότυπά τους τα οποία καλούνται

“entities”, δηλαδή οντότητες, μ’ ένα κατανομημένο τρόπο στο σύστημα αρχείων Google File System. Υπάρχει η πιθανότητα κάποιος να θεωρήσει τα “kinds” του Datastore ως δομές πίνακα και τα “entities” ως εγγραφές, εντούτοις υπάρχουν σημαντικές διαφορές μεταξύ ενός σχεσιακού μοντέλου και του Datastore, μερικές από τις οποίες φαίνονται στην Εικόνα 25.



**Εικόνα 25 - Google Datastore**

Σε ένα σχεσιακό μοντέλο όλες οι γραμμές ενός πίνακα έχουν το ίδιο σύνολο στηλών. Σε αντιδιαστολή όλες οι οντότητες ενός “kind” δεν χρειάζεται να έχουν τις ίδιες ιδιότητες. Αντ’ αυτού, πρόσθετες ιδιότητες μπορούν να προστεθούν σε οποιαδήποτε οντότητα. Το χαρακτηριστικό αυτό γνώρισμα είναι ιδιαίτερος χρήσιμο σε καταστάσεις όπου δεν μπορούν να προβλεφθούν όλες οι πιθανές ιδιότητες ενός μοντέλου, ειδικά στις καταστάσεις που εμφανίζονται περιστασιακά για ένα μόνο μικρό υποσύνολο των εγγραφών.

Για παράδειγμα, ένα μοντέλο που αποθηκεύει προϊόντα διαφορετικών τύπων (π.χ. βιβλία, κλπ) θα πρέπει να επιτρέψει σε κάθε προϊόν να έχει ένα διαφορετικό σύνολο χαρακτηριστικών γνωρισμάτων. Σε ένα σχεσιακό μοντέλο, αυτό πολύ πιθανόν να εφαρμοζόταν με χρήση ενός πίνακα χαρακτηριστικών γνωρισμάτων, όπως παρουσιάζεται στο κάτω αριστερά σημείο της παραπάνω εικόνας. Για χρήση του Datastore, δεν απαιτείται αυτός ο πίνακας (“kind”). Αντίθετα όμως, σε κάθε οντότητα προϊόντος μπορεί να ανατεθεί ένα διαφορετικό σύνολο ιδιοτήτων στο χρόνο εκτέλεσης.

Επίσης, το Datastore επιτρέπει απλά ερωτήματα με συγκεκριμένους όρους, όπως για παράδειγμα το πρώτο ερώτημα που παρουσιάζεται στην Εικόνα 25 για την ανάκτηση όλων των πελατών που έχουν ως ονόματα, κάποιο συγκεκριμένο εύρος γραμμάτων. Η

σύνταξη του ερωτήματος (που ονομάζεται GQL) είναι ουσιαστικά ίδια με την SQL, αλλά με μερικούς περιορισμούς. Παραδείγματος χάριν, όλοι οι όροι ανισότητας σ' ένα ερώτημα πρέπει να αφορούν μία συγκεκριμένη ιδιότητα.

Επίσης οι σχέσεις μεταξύ των πινάκων σ' ένα σχεσιακό μοντέλο, διαμορφώνονται με τη χρήση ξένων κλειδιών. Κατά συνέπεια, κάθε λογαριασμός στον πίνακα ACCTS έχει έναν δείκτη με όνομα ckey στον πελάτη του πίνακα CUSTS όπου αυτό ανήκει. Οι σχέσεις διασχίζονται μέσω ερωτημάτων με τη χρήση ξένων κλειδιών, όπως είναι η ανάκτηση όλων των λογαριασμών ενός συγκεκριμένου πελάτη. Γενικότερα, επικρατεί η γνώμη ότι το Datastore παρέχει μια περισσότερο αντικειμενοστραφή προσέγγιση στις σχέσεις των δεδομένων. Οι ορισμοί μοντέλων μπορούν να περιλάβουν αναφορές σε άλλα μοντέλα και κατά συνέπεια κάθε οντότητα του ACCTS “kind” περιλαμβάνει μια αναφορά στον πελάτη του, ο οποίος είναι μια οντότητα του CUSTS “kind”. Περαιτέρω, οι σχέσεις που ορίζονται από τέτοιες αναφορές μπορούν να διασταυρωθούν κι από τις δύο κατευθύνσεις, έτσι ώστε όχι μόνο η μια να μπορεί να προσπελάσει άμεσα τον πελάτη ενός λογαριασμού, αλλά κι όλους τους λογαριασμούς ενός συγκεκριμένου πελάτη, χωρίς να χρειαστεί να εκτελεστεί η οποιαδήποτε λειτουργία ερωτήματος.

Σχετικά με τα ερωτήματα GQL, δεν μπορούν να εκτελέσουν συνενώσεις (joins) μεταξύ μοντέλων. Οι συνενώσεις είναι πολύ σημαντικές τη χρήση της SQL για την ανάκτηση των δεδομένων από πολλαπλούς πίνακες. Για παράδειγμα, το ερώτημα που παρουσιάζεται στην εικόνα ανακτά λεπτομέρειες όλων των προϊόντων που έχουν αγοραστεί από έναν συγκεκριμένο πελάτη. Το ερώτημα αυτό χρειάζεται να ενώσει τα στοιχεία από τους πίνακες των συναλλαγών (TXNS), των προϊόντων (PRODS) και των χαρακτηριστικών γνωρισμάτων των προϊόντων (FEATURES). Ακόμα κι αν η GQL δεν επιτρέπει ενώσεις, η ικανότητά της να διασχίζει συσχετίσεις μεταξύ των οντοτήτων, συχνά επιτρέπει την αποφυγή συνενώσεων, όπως φαίνεται στο παράδειγμα της παραπάνω εικόνας, με την αποθήκευση των αναφορών των πελατών και των προϊόντων στο μοντέλο Txns, είναι δυνατή η ανάκτηση όλων των συναλλαγών για ένα συγκεκριμένο πελάτη μέσω μιας αντίστροφης διάσχισης της αναφοράς του πελάτη.

Το προϊόν αναφέρεται σε κάθε συναλλαγή κι έπειτα εμφανίζεται το σύνολο των προϊόντων αλλά και τα χαρακτηριστικά γνωρίσματά τους (αξίζει να σημειωθεί ότι δεν απαιτείται ένα χωριστό μοντέλο χαρακτηριστικών γνωρισμάτων εξαιτίας της ευελιξίας του μοντέλου της Εικόνα 25). Είναι σημαντικό να αναφερθεί ότι, ενώ η διάσχιση της σχέσης των αντικειμένων μπορεί να χρησιμοποιηθεί ως εναλλακτική λύση, αυτό δε μπορεί να είναι πάντα εφικτό διότι υπάρχει η πιθανότητα κάποια στιγμή να χρειαστεί να εκτελεστούν συνενώσεις (joins) από τον κώδικα της εφαρμογής.

Αξίζει να σημειωθεί ακόμη ότι στο σχήμα που παρακολουθήσαμε προηγούμενα, οι οντότητες των ειδών Accts και Txns “kinds” αποκτούν στιγμιότυπα με μια παράμετρο “parent” που καθορίζει μια συγκεκριμένη οντότητα πελατών κι έτσι συνδέονται αυτές οι τρεις οντότητες μέσα μια ομάδα οντοτήτων. Το Datastore εξασφαλίζει ότι όλες οι οντότητες που ανήκουν σε μια συγκεκριμένη ομάδα, αποθηκεύονται μαζί στο κατανεμημένο σύστημα αρχείων. Επιτρέπει βέβαια να ομαδοποιηθούν τα βήματα επεξεργασίας του Datastore, σε συναλλαγές, όπου οι ενημερώσεις των δεδομένων είναι ατομικές. Εντούτοις, αυτό απαιτεί ότι κάθε συναλλαγή χειρίζεται μόνο τις οντότητες που ανήκουν στην ίδια ομάδα οντοτήτων. Ωστόσο, ενώ αυτό το μοντέλο συναλλαγής επαρκεί για τις περισσότερες από τις online εφαρμογές, οι σύνθετες ενημερώσεις που



ενημερώνουν πολλές ανεξάρτητες οντότητες δεν μπορούν εκτελεστούν ατομικά, σε αντιδιαστολή με μια σχεσιακή βάση δεδομένων όπου δεν υπάρχει κανένας τέτοιος περιορισμός.

Συνοψίζοντας,, το Google Datastore είναι ένα κατανεμημένο κατάστημα αντικειμένων όπου οι οντότητες όλων των Google App Engine εφαρμογών διατηρούνται χρησιμοποιώντας ένα μεγάλο αριθμό εξυπηρετών καθώς και το κατανεμημένο σύστημα αρχείων Google File System. Επίσης από τη σκοπιά των χρηστών, είναι σημαντικό να εξασφαλιστεί ότι παρά τον διαμοιρασμό ενός κατανεμημένου σχήματος αποθήκευσης με πολλούς χρήστες, τα δεδομένα της εφαρμογής (α) ανακτώνται επαρκώς κι (β) ενημερώνονται ατομικά. Το Datastore παρέχει ένα μηχανισμό για τη συγκρότηση οντοτήτων από διαφορετικά “kinds” σε μια ιεραρχία.

### 4.3 Microsoft Azure

Η εταιρεία Microsoft διαθέτει στο κοινό την υπηρεσία νέφους που καλείται Azure. Γενικά, η Microsoft Azure είναι μια υπηρεσία ΠωΥ όπως και η Google App Engine που αναφέρθηκε στην ενότητα 4.2. Είναι μια πλατφόρμα επέκτασης και προσφέρει αρκετές λειτουργίες, καθώς και εργαλεία στους χρήστες. Στην υπηρεσία αυτή τα στελέχη ανάπτυξης εφαρμογών έχουν τη δυνατότητα, χρησιμοποιώντας τα εργαλεία ανάπτυξης της Microsoft, να δημιουργούν εφαρμογές. Ένα παράδειγμα των εργαλείων αυτών είναι το Microsoft Visual Studio μαζί με τις γλώσσες προγραμματισμού που υποστηρίζει, C#, Visual Basic, ASPs, Java, PHP κ.λπ. Επίσης προσφέρονται υπηρεσίες σχεσιακής αποθήκευσης, στις οποίες όμως υπάρχουν κάποιοι περιορισμοί οι όποιοι αναφέρονται εκτενεστέρα στην επομένη ενότητα.

Μεταξύ της υπηρεσίας νέφους Azure και των υπηρεσιών νέφους της Amazon υπάρχουν κάποιες ομοιότητες, οι οποίες είναι:

- Η χρήση της εικονικοποίησης (virtualization).
- Η παροχή στους χρήστες της δυνατότητας ελέγχου, όσον αφορά την ελαστικότητα.
- Η παροχή στους χρήστες της δυνατότητας ελέγχου σε σχέση με τον αριθμό των εικονικών εξυπηρετών που υπάρχουν σε μια εφαρμογή.
- Η αποθήκευση αρχείων οποιουδήποτε τύπου και αντικειμένων.
- Μια υπηρεσία ουράς μηνυμάτων όπως η Amazon SQS.

Στην υπηρεσία Microsoft Azure ο κώδικας εφαρμογής εκτελείται σ' έναν αριθμό εικονικών μηχανών, τα λεγόμενα και ως στιγμιότυπα (instances). Κάθε εικονική μηχανή διαθέτει λειτουργικό σύστημα Windows Server. Οι χρήστες στην υπηρεσία αυτή έχουν τη δυνατότητα να διευκρινίζουν τον αριθμό των εκάστοτε στιγμιότυπων που απαιτούνται από μια συγκεκριμένη εφαρμογή. Παρ' όλα αυτά, δεν έχουν κανένα έλεγχο κατά την εκκίνηση των στιγμιότυπων αλλά και του χρονικού διαστήματος που παραμένουν προσβάσιμα.

Ο κώδικας εφαρμογής μπορεί να αναπτυχθεί στο Azure με δυο τρόπους :

- Ως ρόλος διαδικτύου (web roles).

- Ως ρόλος εργαζομένου (worker roles).

Στα στιγμιότυπα των ρόλων διαδικτύου, ο κώδικας εκτελείται μέσω ενός εξυπηρετή δικτύου (Microsoft Internet Information Service - IIS). Ο εξυπηρετής αυτός συμπεριλαμβάνεται σε κάθε στιγμιότυπο. Επίσης οι ρόλοι Διαδικτύου ανταποκρίνονται στις HTTP αιτήσεις που είναι για την εξισορρόπηση φόρτου εργασίας στα στιγμιότυπα της εφαρμογής.

Στα στιγμιότυπα των ρόλων εργαζομένου, ο κώδικας μπορεί να εκτελεστεί ως μια μέθοδος επεξεργασίας ενός συνόλου δεδομένων (batch). Επίσης μπορεί να επικοινωνήσει μ' ένα ρόλο Διαδικτύου μέσω του αποθηκευτικού χώρου των δεδομένων του Azure, όπως είναι οι ουρές αναμονής ή οι πίνακες. Επίσης οι εφαρμογές του ρόλου εργαζομένου δεν μπορούν να προσπελαστούν από ένα εξωτερικό δίκτυο. Βεβαία μπορούν να κάνουν εξωτερικές HTTP αιτήσεις, είτε στους ρόλους εργαζομένου είτε μέσω του Διαδικτύου.

Στη συνέχεια αναφέρεται ο τρόπος αποθήκευσης της υπηρεσίας Azure. Ο αποθηκευτικός χώρος δεδομένων του Azure επιτρέπει την αποθήκευση μη σχεσιακών πινάκων και ουρών. Εντούτοις, από παρέχεται ένα εναλλακτικό σχήμα αποθήκευσης βάσεων δεδομένων από τις υπηρεσίες SQL Data Services. Οι υπηρεσίες δεδομένων παρέχονται από έναν αριθμό συστάδων (clusters) που εκτελούν τη βάση δεδομένων του Microsoft SQL Server. Ο Microsoft SQL Server επεκτείνεται σε διάφορα κέντρα δεδομένων της Microsoft τα οποία ονομάζονται *αρχές* (authorities). Κάθε αρχή μπορεί να φιλοξενήσει έναν αριθμό από *περιέκτες* (containers), οι οποίοι είναι παρόμοιοι με τους πίνακες βάσεων δεδομένων. Επίσης, κάθε περιέκτης περιέχει οντότητες που έχουν ιδιότητες, όπου όμως οι ιδιότητες αυτές για κάθε οντότητα μπορούν να διαφέρουν στον τύπο, αλλά και τον αριθμό. Ακόμα οι εφαρμογές προσπελαίνουν τις υπηρεσίες SQL Data Services στο επίπεδο των περιεκτών, μέσω ενός γενικού σχήματος διευθύνσεων. Εν ολίγοις δηλαδή, η αρχιτεκτονική των αρχών, των περιεκτών αλλά και των πινάκων έχει αντικατασταθεί από ένα παραδοσιακό σχεσιακό μοντέλο που υποστηρίζεται από τον Microsoft SQL Server. Ειδικότερα, εκεί περιλαμβάνεται η υποστήριξη για συνενώσεις (joins) μεταξύ των πινάκων, των συναλλαγών και άλλων χαρακτηριστικών γνωρισμάτων της σχεσιακής βάσης δεδομένων.

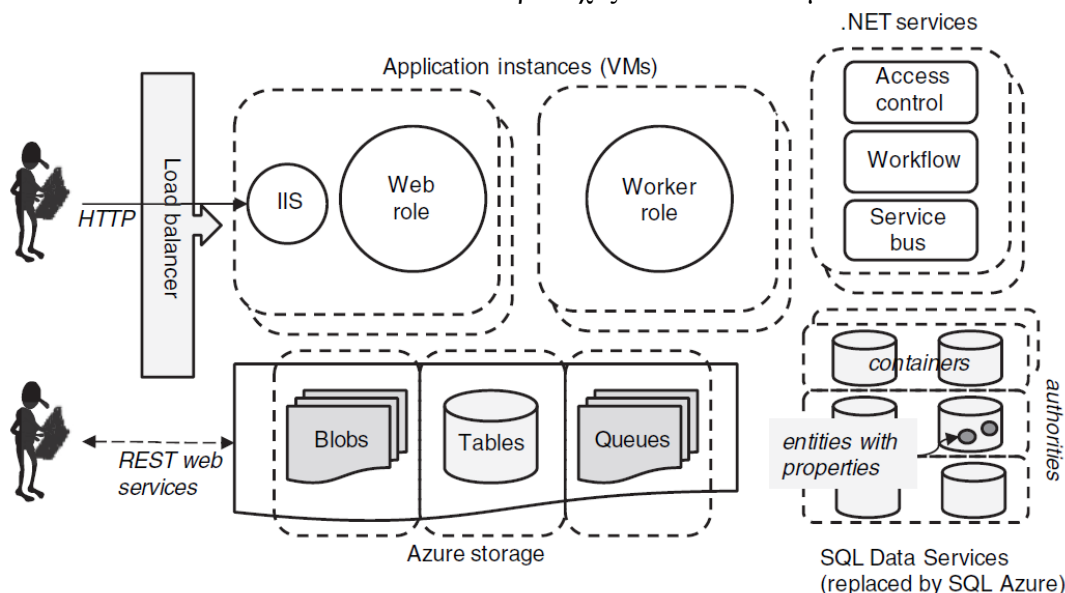
Εντούτοις, κάθε βάση δεδομένων SQL Azure περιορίζεται σε μέγεθος λιγότερο από 10 GB. Βέβαια, εάν υπάρξει περίπτωση που χρειαστεί να αποθηκευτούν μεγαλύτερες ποσότητες δεδομένων, τότε θα πρέπει να τροφοδοτηθούν πολλαπλά στιγμιότυπα εικονικών βάσεων δεδομένων. Με δεδομένο όμως πως σε βάσεις δεδομένων δεν επιτρέπονται ερωτήματα που προσπελαίνουν πολλαπλά στιγμιότυπα, τότε θα πρέπει μέσα στον κώδικα της εφαρμογής να εφαρμοστούν ερωτήματα σε μεγαλύτερα σύνολα δεδομένων, συμπεριλαμβανομένων των συνενώσεων. Έτσι προκύπτει το μοντέλο SQL Azure, το οποίο αντιπροσωπεύει ένα πρακτικό συμβιβασμό μεταξύ των περιορισμών μη σχεσιακών μοντέλων εναντίον μιας παραδοσιακής σχεσιακής βάσης δεδομένων. Συγκεκριμένα, τα πλήρως σχεσιακά ερωτήματα γίνονται επιτρεπτά, αλλά μόνο σε μικρά σύνολα δεδομένων, τα οποία με τη σειρά τους είναι πιθανό να επαρκούν για πολλές υπηρεσίες νεφών.

Το Microsoft Azure, εκτός από τις υπηρεσίες υπολογισμού και αποθήκευσης, παρέχει και τις καλούμενες υπηρεσίες δικτύου (δηλαδή .NET), οι οποίες είναι:

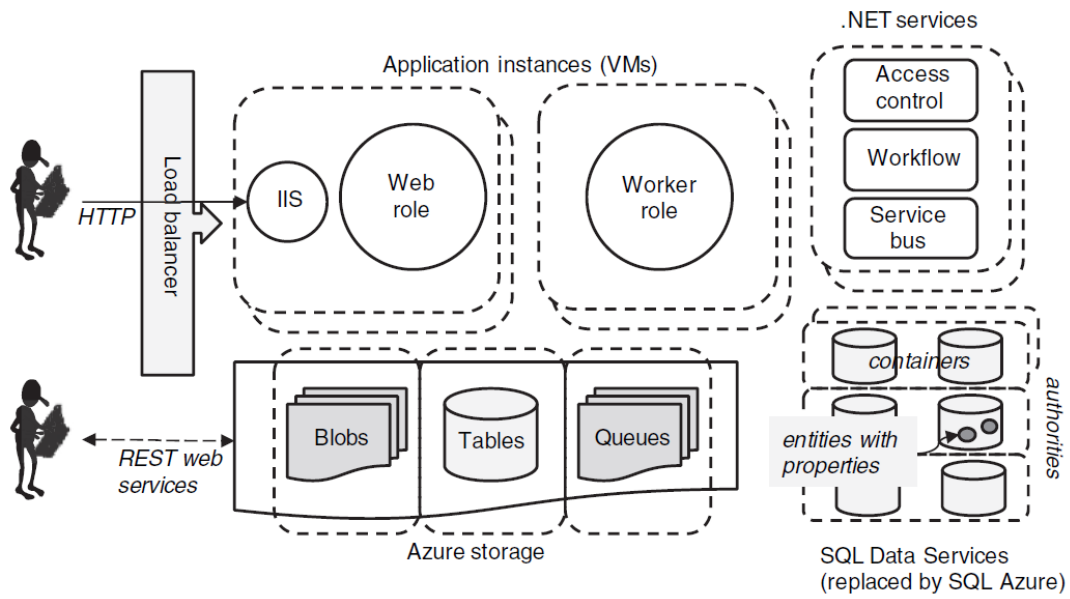
- Υπηρεσίες ελέγχου πρόσβασης που παρέχουν διακριτικά (tokens) με σκοπό την ασφάλεια της πρόσβασης.
- Ένα δίκτυο υπηρεσιών που ενεργοποιεί άκρα (end points) για δημοσιευμένες υπηρεσίες.
- Μια υπηρεσία δικτύου που βασίζεται στη ρύθμιση της ροής εργασίας.

Όλες αυτές οι υπηρεσίες βασίζονται στα προϊόντα επιχειρήσεων της Microsoft, για τη διαχείριση της ταυτότητας και των υπηρεσιών Ιστού, τα οποία επεκτείνονται στο νέφος σε μια καταναμημένη και γενικά προσπελάσιμη πλατφόρμα. Σε σύγκριση με την υπηρεσία ΠωΥ της Google, το Microsoft Azure έχει το πλεονέκτημα μιας μεγάλης βάσης εφαρμογών της Microsoft, η οποία ήδη χρησιμοποιείται από μεγάλο όγκο επιχειρήσεων.

Επιπλέον, χάρη στην ταχεία εξέλιξη της εμπορικής έκδοσης, είναι πολύ πιθανό να γίνει ακόμα ευκολότερη η ενσωμάτωση των υπάρχουσών εφαρμογών της Microsoft στο Azure στο προσεχές μέλλον. Η



Εικόνα 26 απεικονίζει την αρχιτεκτονική του Microsoft Azure, περιλαμβάνοντας τους ρόλους των στιγμιότυπων και τις λοιπές έννοιες που περιγράφηκαν ανωτέρω.



**Εικόνα 26 - Microsoft Azure**

## Κεφάλαιο 5: Ανάπτυξη εφαρμογών στο υπολογιστικό νέφος

Ο τρόπος με τον οποίο σχεδιάστηκε το Διαδίκτυο γενικά προβλέπει τη μεταχείριση της κάθε μιας αίτησης που πραγματοποιείται σ' έναν εξυπηρετή ως μια ανεξάρτητη δοσοληψία. Για το λόγο αυτό, όλες οι βασικές HTTP (Hypertext Transfer Protocol Secure) εντολές είναι ατομικής φύσεως - όπως για παράδειγμα:

1. Η GET για την ανάγνωση δεδομένων,
2. Η PUT για την εγγραφή δεδομένων, κ.ά.

Επιπλέον, η σχεδίαση των διαδικτυακών πρωτοκόλλων αλλά και των κατανεμημένων εφαρμογών του Διαδικτύου, έχει γίνει θεωρώντας τα ως υπηρεσίες χωρίς καταστάσεις (stateless).

Ωστόσο, ενώ οι εξυπηρετές χωρίς καταστάσεις είναι πιο εύκολο να σχεδιαστούν, πολλές λειτουργίες απαιτούν τη διατήρηση καταστάσεων. Ως παράδειγμα, ας θεωρήσουμε ένα σύστημα κρατήσεων, το οποίο χρησιμοποιεί κάποιος πελάτης με σκοπό να προμηθευτεί κάποιο προϊόν. Οι ενέργειες που ακολουθεί ο πελάτης είναι οι ακόλουθες:

1. Αρχικά εξετάζει τον κατάλογο.
2. Ύστερα κάνει κράτηση στο προϊόν που τον ενδιαφέρει.
3. Στο τέλος πληρώνει γι' αυτό το προϊόν.

Σε ένα σύστημα με πολλούς χρήστες, εάν δεν υποστηρίζονται καταστάσεις, ο εκάστοτε ενδιαφερόμενος δεν θα έχει τη δυνατότητα να γνωρίζει εάν το προϊόν που επιθυμεί είναι ήδη κρατημένο από κάποιον άλλο χρήστη. Αυτό θα του γνωστοποιηθεί τη στιγμή που θα ετοιμαστεί να το πληρώσει. Επίσης, στο σύστημα που υποστηρίζει καταστάσεις, είναι πολύ πιο εύκολο να επιστραφεί ένα προϊόν στην αποθήκη και εν συνεχεία να επιστραφούν τα χρήματα που πληρώθηκαν, εάν είναι εφικτό να επαναληφθούν τα βήματα που έγιναν για την αγορά του προϊόντος, από την αντίστροφη όμως κατεύθυνση (στο πλαίσιο των υπηρεσιών διαδικτύου, η επανάληψη αυτή αναφέρεται με τον όρο *δοσοληψίες αποζημίωσης* – compensating transactions).

Έχει γίνει εκτενής έρευνα σε σχέση με το πώς είναι δυνατόν να υποστηριχθούν πλήρως τα σημασιολογικά χαρακτηριστικά των δοσοληψιών που παρέχονται από τους εξυπηρετές με διατήρηση καταστάσεων, πάνω σε εξυπηρετές που δεν διατηρούν καταστάσεις. Στο πλαίσιο αυτό έχουν προταθεί τα ακόλουθα:

- Ανάπτυξη εποπτών δοσοληψιών.
- Ανάπτυξη εξυπηρετών ουράς μηνυμάτων.
- Ανάπτυξη άλλου λογισμικού.

Η υπολογιστική νέφους προσφέρει δυνατότητες υλοποίησης δοσοληψιών, κυρίως μέσω των ακόλουθων λύσεων:

- της έννοιας της «νορχήστρωσης», όπου η ροή διεργασιών μπορεί να θεωρηθεί ως μια υπηρεσία.

- της έννοιας του διαύλου υπηρεσιών (service bus) που ελέγχει τα στοιχεία ενός νέφους.

### **5.1 Μεταφορά εφαρμογών στο νέφος**

Προκειμένου να προσδιοριστεί ο χρόνος που πρέπει να μεταφερθούν κάποιες εφαρμογές στο νέφος, δηλαδή να προσφέρονται πλέον μέσω του νέφους και όχι με τον «παραδοσιακό» τρόπο, θα πρέπει να αναλυθεί το κατά πόσο χρειάζονται να υποστηριχθούν κάποια κρίσιμα χαρακτηριστικά γνωρίσματα στις εφαρμογές αυτές. Αφού κατανοηθούν αυτά τα χαρακτηριστικά γνωρίσματα, τότε μπορεί να αποφασιστεί κατά πόσο το νέφος είναι σε θέση να τα υποστηρίζει.

Τέτοια χαρακτηριστικά γνωρίσματα είναι τα ακόλουθα τρία:

1. Η πρόσβαση σε δεδομένα.
2. Οι καθυστερήσεις.
3. Η ασφάλεια δεδομένων.

Στο σημείο αυτό πρέπει να αναφερθεί πως κατά τη μετακίνηση μιας εφαρμογής στο νέφος, πρέπει να χρησιμοποιηθούν οι προγραμματιστικές διασυνδέσεις εφαρμογών APIs (Application programming interface) του συγκεκριμένου παρόχου υπηρεσιών του νέφους. Ειδικότερα, υπάρχουν προγραμματιστικές διασυνδέσεις εφαρμογών (APIs) για κάθε ένα τύπο υπηρεσίας του νέφους:

- Υποδομής,
- Υπηρεσιών λογισμικού, καθώς και
- Εφαρμογές παρόχων πλατφορμών.

Αυτές οι προγραμματιστικές διασυνδέσεις εφαρμογών δεν είναι διαλειτουργικές, κάτι που σημαίνει ότι παρόλο που μπορεί να αλλάξει η κατάσταση στο μέλλον, εντούτοις, τα στελέχη ανάπτυξης εφαρμογών πρέπει να κάνουν τη καλύτερη επιλογή όσον αφορά τον πάροχο ώστε να τους επιτρέπει την όσο το δυνατό μεγαλύτερη ευελιξία.

### **5.2 Τρόποι σύνδεσης χρηστών με το νέφος**

Οι τρόποι με τους οποίους οι εξυπηρετούμενοι μπορούν να συνδεθούν με το νέφος ποικίλουν. Οι πιο δημοφιλείς τρόποι με τους οποίους επιτυγχάνεται η σύνδεση αυτή, είναι οι ακόλουθοι:

- Μέσω ενός προγράμματος πλοήγησης, όπως ο Internet Explorer, ο Mozilla Firefox, κ.ά., ή
- Μέσω μιας εξειδικευμένης εφαρμογής.

Αυτές οι δύο εφαρμογές μπορούν να εκτελούνται σε έναν εξυπηρετή, έναν υπολογιστή γραφείου, σ' ένα κινητό τηλέφωνο ή σ' ένα tablet. Ωστόσο, όλες οι συσκευές αυτές που αναφέρθηκαν, έχουν κοινό το ότι ανταλλάσσουν δεδομένα μέσω ενός μη ασφαλούς μέσου. Για μια ασφαλή μετάδοση δεδομένων υπάρχουν οι εξής τρεις τρόποι:

- Χρήση ενός ασφαλούς πρωτοκόλλου για τη μεταφορά δεδομένων, όπως το (Secure Sockets Layer-SSL) (HTTPS), το FTPS, το IPsec, ή η ασφαλής σύνδεση με χρήση του (secure shell- SSH) για τη σύνδεση ενός πελάτη με το νέφος.
- Δημιουργία μιας εικονικής σύνδεσης με χρήση ενός εικονικού ιδιωτικού δικτύου (Virtual Private Network, VPN), ή με χρήση ενός απομακρυσμένου πρωτοκόλλου μεταφοράς δεδομένων όπως είναι το Microsoft RDP ή το Citrix ICA, όπου τα δεδομένα προστατεύονται από ένα μηχανισμό σήραγγας (tunneling).
- Κρυπτογράφηση των δεδομένων έτσι ώστε ακόμα κι αν η ροή των δεδομένων διακοπεί για κάποιο λόγο ή παρεμποδιστεί, να μη τίθεται σημαντικό πρόβλημα διαρροής των αποκρυπτογραφημένων δεδομένων.

Σε πολλές περιπτώσεις, οι συνδέσεις από την πλευρά των εξυπηρετούμενων, χρησιμοποιούν δύο ή περισσότερους από τους παραπάνω τρόπους για την επικοινωνία τους με το νέφος. Ας θεωρήσουμε ένα παράδειγμα χρήσης μη ασφαλούς επικοινωνίας για την επικοινωνία με το νέφος: Έστω η υποδομή ενός ξενοδοχείου που συνδέεται στο Διαδίκτυο ώστε να μπορεί κάποιος χρήστης να συνδεθεί με τις υπηρεσίες του ξενοδοχείου. Επιπρόσθετα, το δίκτυο του ξενοδοχείου δεν προστατεύεται από κάποιο τείχος προστασίας (firewall). Έτσι οι χρήστες επιλέγουν να φέρουν μαζί τους φορητούς δρομολογητές, οι οποίοι τους παρέχουν ένα προσωπικό τείχος προστασίας (firewall) σε μορφή υλικού, και οι φορητοί αυτοί δρομολογητές δημιουργούν ένα εικονικό ιδιωτικό δίκτυο (VPN), όπως αναφέρθηκε και προηγουμένως.

### 5.3 Εφαρμογές στα νέφη

Κατά την ανάπτυξη μιας εφαρμογής στο νέφος, θα πρέπει αναλυθούν τα πλεονεκτήματα και τα μειονεκτήματα του καταναμημένου συστήματος (δηλαδή του Διαδικτύου στη περίπτωση μας) αλλά και των θεμελιωδών χαρακτηριστικών που προσφέρουν τα νέφη. Κάθε εφαρμογή μπορεί να εκτελείται πλήρως ή μερικώς στο νέφος. Τα θεμελιώδη χαρακτηριστικά που πρέπει να προσφέρει ένα νέφος είναι τα ακόλουθα:

- Αφαιρετικότητα συστήματος και επαναπροσανατολισμός.
- Εξελιξιμότητα.
- Σύνολο προγραμματιστικών διασυνδέσεων εφαρμογών(APIs) της εφαρμογής και του συστήματος
- Μικρές καθυστερήσεις στα τοπικά δίκτυα υπολογιστών (Local Area network-LAN) και στα δίκτυα ευρείας περιοχής (Wide Area network-WAN).

Αρχικά κατά τη μεταφορά της εφαρμογής στο νέφος, το στέλεχος ανάπτυξης θα πρέπει να εξετάσει εάν οι λειτουργίες της εφαρμογής εξυπηρετούνται καλύτερα από το νέφος ή την συνήθη τοπική ανάπτυξη. Η απάντηση εξαρτάται σ' ένα πολύ μεγάλο ποσοστό από τα χαρακτηριστικά γνωρίσματα της εφαρμογής που προσπαθεί να συντηρήσει ή να ενισχύσει.

Επίσης η θέση μιας εφαρμογής ή μιας υπηρεσίας παίζει πολύ σημαντικό ρόλο στο τρόπο με τον οποίο πρέπει να κατασκευαστεί και να αναπτυχθεί μια εφαρμογή. Ειδικότερα, μια

εφαρμογή ή μια διαδικασία που εκτελείται σ' έναν υπολογιστή γραφείου ή σε έναν εξυπηρετή, εκτελείται συνεκτικά και ενιαία ως μια μονάδα, κάτω από τον έλεγχο ενός προγράμματος. Με λίγα λόγια μια ενέργεια προξενεί τα εξής:

1. Αρχικά, προκαλεί μια κλήση προγράμματος,
2. Εν συνεχεία, εκτελείται ο κώδικας και
3. Τελικά, επιστρέφεται ένα αποτέλεσμα.

Στη συνέχεια ας θεωρήσουμε τις ατομικές δοσοληψίες. Ως ατομική δοσοληψία θεωρούμε μία ακολουθία συμβάντων της μορφής: *αίτηση* → *διαδικασία* → *απόκριση*. Όμως καθώς η δοσοληψία εκτελείται τοπικά μέσα σε μια εφαρμογή, η διαδικασία έχει καταστάσεις και κατά συνέπεια η δοσοληψία είναι συνεπής και η κατάστασή της είναι πάντα γνωστή. Μια συνεπής δοσοληψία είτε πετυχαίνει, ολοκληρώνεται και μονιμοποιείται είτε αποτυγχάνει και επιστρέφει στην προηγούμενη κατάσταση. Όταν δεν μπορεί να γίνει η επιστροφή στη προηγούμενη κατάσταση (φαινόμενο που ονομάζεται και rollback) λόγω πιθανής δέσμευσης της δοσοληψίας σε μια εφαρμογή πολλών χρηστών, τότε απαιτείται η διόρθωση της κατάστασης ή η εκτέλεση κάποιας αντισταθμιστικής ενέργειας σε μεταγενέστερο χρόνο.

Μια εφαρμογή που εκτελείται ως υπηρεσία στο Διαδίκτυο, αποτελείται από δυο μέρη:

- Αυτό του εξυπηρετούμενου που υποβάλλει μια αίτηση και
- Αυτό του εξυπηρετή που αποκρίνεται στην αίτηση.

Η αίτηση είναι αποσυνδεδεμένη από την απόκριση, υπό την έννοια ότι η δοσοληψία εκτελείται σε δύο ή περισσότερες τοποθεσίες. Επίσης σε ένα καταναμημένο σύστημα, η δοσοληψία είναι άνευ καταστάσεων. Προκειμένου λοιπόν σε μια καταναμημένη αρχιτεκτονική να δημιουργηθεί ένα σύστημα με καταστάσεις, πρέπει να προστεθεί μία οντότητα με τον ρόλο *υπεύθυνος δοσοληψιών*. Η ενδιάμεση αυτή οντότητα θα μπορεί να χρησιμοποιηθεί για τις δοσοληψίες και να αντιδρά αναλόγως όταν οι τελευταίες επιτυγχάνουν ή αποτυγχάνουν.

Τέλος κατά τη μεταφορά των εφαρμογών στο νέφος διατηρούνται τα χαρακτηριστικά γνωρίσματα μιας αρχιτεκτονικής τριών επιπέδων. Επίσης τα φυσικά συστήματα γίνονται εικονικά. Οι εικονικές μηχανές δεν είναι μόνο άνευ καταστάσεων, αλλά και η τοποθεσία όπου πραγματοποιείται η εκτέλεση του προγράμματος μπορεί να είναι διαφορετική κάθε φορά που εκτελείται η διαδικασία.

### 5.3.1 Ιδιότητες αξιόπιστων δοσοληψιών

Οι ιδιότητες ACID των αξιόπιστων δοσοληψιών διατυπώθηκαν από τον Jim Gray και εφαρμόστηκαν για πρώτη φορά στην τεχνολογία των βάσεων δεδομένων προς το τέλος της δεκαετίας του '70. Στις μέρες μας χρησιμοποιείται από οποιαδήποτε εφαρμογή που διαβάζει και γράφει σε ένα σύνολο αποθηκευμένων δεδομένων, το οποίο περιλαμβάνει όλους τους τύπους εφαρμογής. Συγκεκριμένα, τα γράμματα της λέξης ACID προέρχονται από τις ακόλουθες τέσσερις λέξεις:

- **Ατομικότητα (Atomicity):** Όσον αφορά την ιδιότητα της ατομικότητας πρέπει να ειπωθεί πως καθορίζει μια δοσοληψία ως κάτι που δεν μπορεί να



διαιρεθεί αλλά πρέπει να ολοκληρωθεί ή αντιστρόφως να εγκαταλειφθεί, ως μια ενιαία μονάδα.

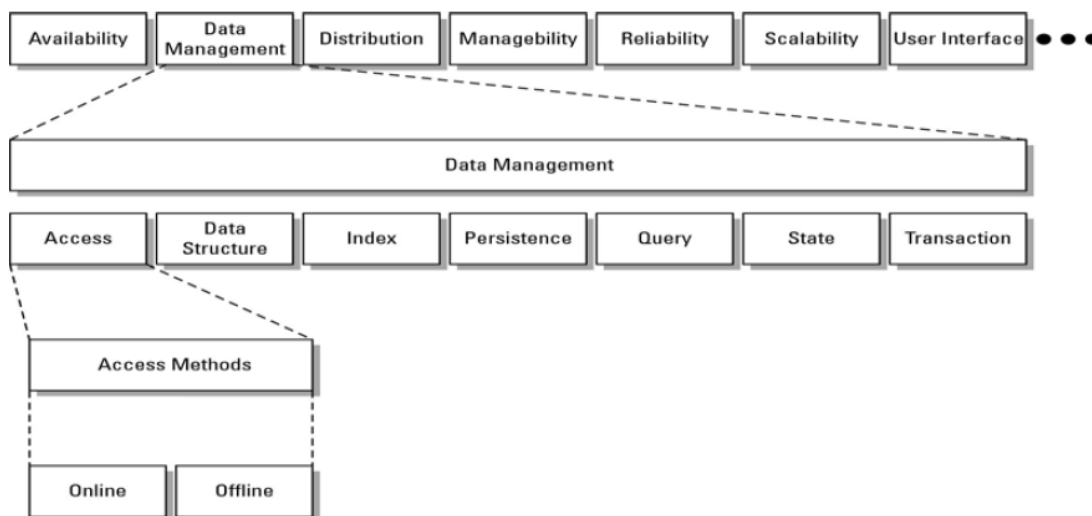
- **Συνέπεια (Consistency):** Η ιδιότητα της συνέπειας δηλώνει ότι το σύστημα στο τέλος της δοσοληψίας θα πρέπει να μεταφερθεί από μια γνωστή-συνεπή κατάσταση σε μια άλλη. Επίσης η ιδιότητα αυτή δηλώνει ότι πρέπει να διατηρηθεί η ακεραιότητα του συστήματος.
- **Απομόνωση (Isolation):** Η ιδιότητα της απομόνωσης δηλώνει ότι τα αποτελέσματα μιας δοσοληψίας δεν πρέπει να επηρεάζονται από άλλες δοσοληψίες που τυχόν εκτελούνται την ίδια στιγμή.
- **Διάρκεια (Durability):** Η ιδιότητα της διάρκειας δηλώνει ότι το σύστημα πρέπει να έχει ένα μηχανισμό ο οποίος μετά την επιτυχή πραγματοποίηση μιας δοσοληψίας τα αποτελέσματά της μένουν μόνιμα στο σύστημα.

### 5.3.2 Χαρτογράφηση λειτουργικότητας

Για να αποφασίσει κάποιος υπεύθυνος να μεταφέρει την εφαρμογή του στο νέφος έτσι ώστε να επωφεληθεί από την ανάπτυξη του νέφους, θα χρειαστεί να αποδομήσει τη λειτουργικότητα της συγκεκριμένης εφαρμογής στα βασικά στοιχεία της και εν συνεχεία να αναγνωρίσει τις λειτουργίες που είναι θεμελιώδεις και που μπορούν να υποστηριχθούν από το νέφος.

Επιπλέον, τα συστήματα δοσοληψιών απαιτούν τα δεδομένα μιας βάσης δεδομένων να διατηρούν την ακεραιότητα των δοσοληψιών του μοντέλου ACID. Για πολλά μη σχεσιακά συστήματα αποθήκευσης στο νέφος, (όπως η υπηρεσία Amazon Simple Storage Service (S3), η υπηρεσία Google Storage, καθώς και η υπηρεσία Windows Azure), η δυνατότητα του συστήματος να διατηρεί την ακεραιότητα των δοσοληψιών μέσω του κλειδώματος των αρχείων, δεν υποστηρίζεται άμεσα. Αυτοί οι τύποι των συστημάτων αποθήκευσης είναι ασφαλείς και αποθηκεύουν μεγάλες ποσότητες δεδομένων. Διαθέτουν όμως πολύ αργή πρόσβαση στα συγκεκριμένα δεδομένα κι επίσης δεν υποστηρίζουν επερωτήσεις και ανάκτηση πληροφορίας.

Στην Εικόνα 27, κατασκευάζεται ένα δέντρο ιδιοτήτων για ένα σύστημα δοσοληψιών όπου η λειτουργικότητα αποσυντίθεται σε διαφορετικές περιοχές λειτουργίας. Παρατηρούμε πως στο ανώτερο επίπεδο βρίσκονται τα χαρακτηριστικά γνώρισμα υψηλού επιπέδου, όπου κάποια από αυτά αφορούν τη λειτουργία της εφαρμογής ενώ κάποια άλλα όχι. Στη συνέχεια, σε καθοδική παράθεση προς το χαρακτηριστικό γνώρισμα της διαχείρισης δεδομένων, το δεύτερο επίπεδο σχετίζεται αρχικά με την προσπέλαση δεδομένων κι έπειτα με την προσπέλαση μεθόδων. Ένα κρίσιμο χαρακτηριστικό γνώρισμα της εφαρμογής αυτής είναι η ανάγκη προσπέλασης δεδομένων όταν ο εξυπηρετούμενος είναι τόσο online στο νέφος όσο και offline.



**Εικόνα 27 - Δημιουργία χάρτη χαρακτηριστικών γνωρισμάτων για την παρουσίαση της λειτουργικότητας**

Η αλληλεπίδραση της εφαρμογής τόσο με το υπολογιστικό νέφος όσο και με τα δεδομένα καθορίζεται από το αν επιτρέπεται τόσο η προσπέλαση των δεδομένων online όσο και offline. Εάν η εφαρμογή μπορεί να προσπελάσει τα δεδομένα μόνο όταν ο εξυπηρετούμενος είναι online, τότε η τελευταία θα χρειάζεται ως μοναδική αποθήκη δεδομένων (data store), την πρόσβαση στον αποθηκευτικό χώρο που βασίζεται στο νέφος. Ίσως η εφαρμογή θα ήταν εξ ολοκλήρου στο νέφος και θα βασιζόταν στην εφαρμογή πλοήγησης

Με σκοπό να επιτραπεί τόσο η προσπέλαση των δεδομένων του νέφους όσο και η τοπική προσπέλαση δεδομένων, θα πρέπει να δημιουργηθεί μια υβριδική εφαρμογή αποτελούμενη από τα εξής τμήματα:

- Ένα τμήμα νέφους,
- Ένα τοπικό τμήμα.

Ακόμα και αν η πρόσβαση στα δεδομένα, στο τοπικό σύστημα είναι ένα απλό σύστημα αποθήκευσης, εντούτοις απαιτείται υποστήριξη στην πλευρά του εξυπηρετούμενου. Έτσι, για την υποστήριξη της πρόσβασης στα δεδομένα της εφαρμογής, μπορεί επίσης να χρειαστεί η κατασκευή ενός χαρακτηριστικού γνωρίσματος συγχρονισμού, το οποίο προσθέτει μεγαλύτερη επιβάρυνση στην εφαρμογή.

Αυτού του είδους η χαρτογράφηση μας οδηγεί στα ακόλουθα συμπεράσματα όσον αφορά το υπολογιστικό νέφος:

- Μια εφαρμογή που χρησιμοποιεί αποθήκευση δεδομένων στο νέφος, ωφελείται περισσότερο από την ανάπτυξη της σε αυτό, σε σχέση με μια εφαρμογή που χρησιμοποιεί τοπική (offline) αποθήκευση.
- Στην περίπτωση μιας υβριδικής εφαρμογής, υπάρχουν κάποιοι παράγοντες που μπορούν να αντισταθμίσουν το κόστος της τοπικής (offline) αποθήκευσης και να κάνουν πιο λειτουργικό το νέφος. Τέτοιοι είναι η κλιμάκωση (scalability), τα έξοδα καθώς και η διάχυτη, δηλαδή ευρεία, πρόσβαση.

### 5.3.3 Χαρακτηριστικά γνωρίσματα εφαρμογής

Η αποδόμηση της λειτουργικότητας μιας συγκεκριμένης εφαρμογής στα βασικά στοιχεία της, αποτελεί μόνο ένα τμήμα της διαδικασίας σε ό,τι αφορά τη μεταφορά αυτής στο νέφος. Αυτό συμβαίνει διότι κάθε πλατφόρμα νέφους διαθέτει το δικό της σύνολο χαρακτηριστικών γνωρισμάτων που χρειάζεται να χαρτογραφηθούν.

Τα κύρια κριτήρια για να εξαχθεί το συμπέρασμα αν ωφελούνται οι εφαρμογές από την ανάπτυξή τους στο νέφος είναι τα ακόλουθα:

- Δεν υλοποιούν βασικές επιχειρησιακές λειτουργίες.
- Δεν υπάρχουν ευαίσθητα δεδομένα να προστατεύσει.
- Γίνονται ανεκτές υψηλές καθυστερήσεις δικτύου ή χαμηλό εύρος ζώνης δικτύου.
- Είναι εφαρμογές που δεν παρέχουν συγκεκριμένο ανταγωνιστικό πλεονέκτημα.
- Βασίζονται σε τυποποιημένες τεχνολογίες βιομηχανίας.
- Δεν χρειάζεται να είναι προσαρμοσμένες σε κάποιο πρότυπο.
- Είναι αρκετά ώριμες και κατανοητές ώστε να μπορούν να συνδεθούν επιτυχώς στο νέφος.

### 5.3.4 Χαρακτηριστικά γνωρίσματα υπηρεσιών νέφους

Εφόσον μέχρι αυτό το σημείο ανάπτυξης, αναφέρθηκαν τα χαρακτηριστικά γνωρίσματα μιας πλατφόρμας, ώστε να ωφελείται μια εφαρμογή από την ανάπτυξή της στο νέφος, ακολουθεί η αντιστοίχιση αυτών των χαρακτηριστικών γνωρισμάτων με τα παρακάτω βασικά χαρακτηριστικά γνωρίσματα των υπηρεσιών νέφους:

- Εφαρμογές.
- Βασικές υπηρεσίες.
- Υποδομή.
- Χαρακτηριστικά γνωρίσματα πλατφορμών.
- Αποθήκευση.

Στο τρέχον στάδιο της ανάπτυξης μιας εφαρμογής, δεν είναι εφικτή η αντιστοίχιση των αναγκών της εφαρμογής μ' ένα σύνολο παρόχων υπηρεσιών νέφους. Αυτό συμβαίνει για τους λόγους ότι ο κάθε πάροχος:

- Έχει μια μοναδική λύση.
- Χρησιμοποιεί τις δικές του προγραμματιστικές διασυνδέσεις εφαρμογών (APIs).
- Παρέχει μοναδικές υπηρεσίες.

Επομένως, κάθε πάροχος υπηρεσιών νέφους απαιτείται να έχει δεξιότητες ανάπτυξης εφαρμογών, καθώς και να παρέχει ολοκλήρωση μεταξύ των νεφών. Ίσως αυτό το

φαινόμενο να αλλάξει στο μέλλον καθώς αναπτύσσονται περισσότερα πρότυπα. Όμως προς το παρόν, τα στελέχη ανάπτυξης εφαρμογών πρέπει να αντιστοιχίσουν την εφαρμογή τους με τον καλύτερο πάροχο.

### 5.3.5 Αφαιρετικότητα συστήματος

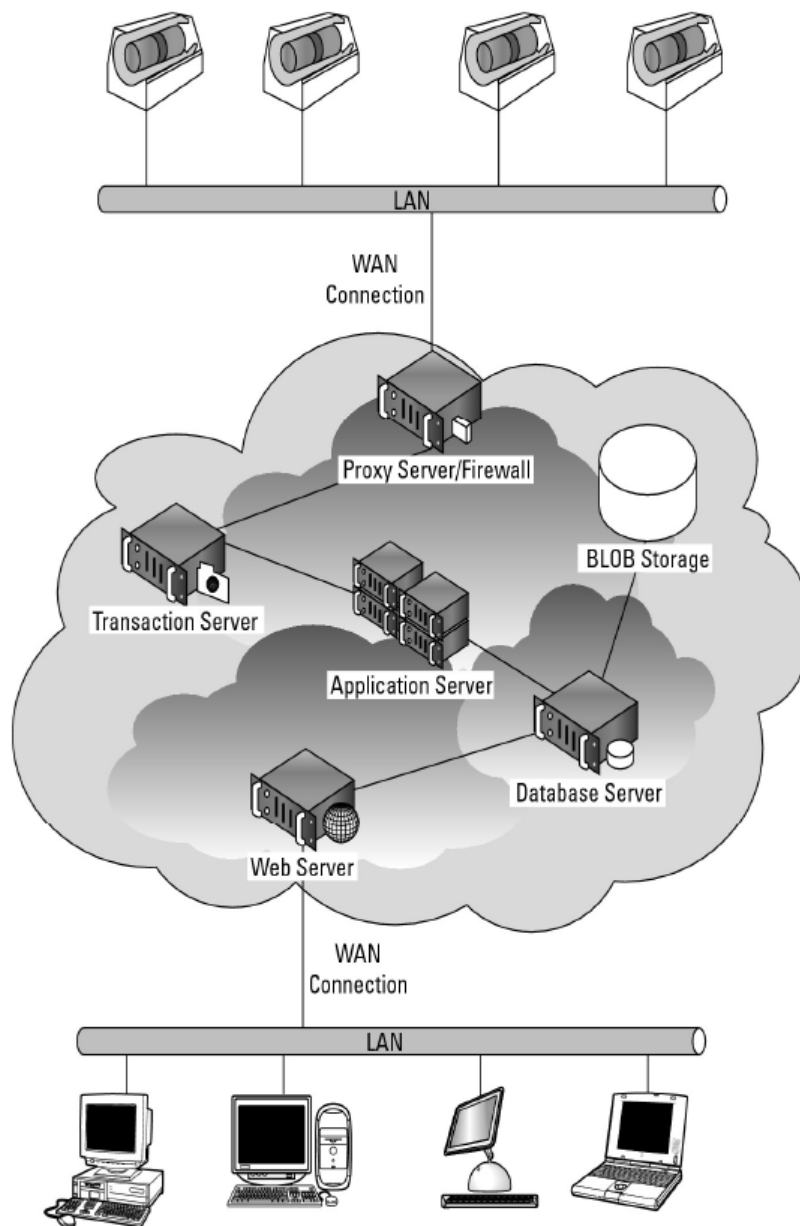
Όπως έχει αναφερθεί και προηγουμένα, το νέφος μετατρέπει τα φυσικά συστήματα σε εικονικά. Ειδικότερα, οι οργανισμοί επιλέγουν να αναπτύξουν εξ ολοκλήρου τα συστήματα τους στο νέφος, έχοντας σαν σκοπό να μπορούν να αναδιαμορφώσουν το βασικό μέρος της διαδικασίας και να εξαλείψουν την έννοια της υποδομής.

Ένα παράδειγμα του φαινομένου αυτού, αποτελεί μια υπηρεσία ιατρικής απεικόνισης. Στο παρελθόν, όπως ήταν σύνηθες, με την υπηρεσία αυτή, δημιουργούνταν απεικονίσεις ασθενών (ακτινογραφίες, τομογραφίες κ.λπ.) κι έπειτα αποθηκεύονταν σε έναν τοπικό υπολογιστή. Εν συνεχεία, αφ' ότου κάποιος αποθήκευε την εικόνα στον υπολογιστή, γινόταν αυτομάτως διαθέσιμη στο τοπικό δίκτυο του νοσοκομείου. Με τον τρόπο αυτό μπορούσε κάποιος γιατρός να τη δει και να την επεξηγήσει. Επιπλέον, όταν οι γιατροί, ήταν εκτός του χώρου του νοσοκομείου, τότε θα χρειαζόταν να εισέλθουν μέσω ενός εικονικού ιδιωτικού δικτύου (Virtual Private Network-VPN) στον εξυπηρετή του νοσοκομείου προκειμένου να προβάλουν το αρχείο στον δικό τους υπολογιστή.

Η παραπάνω υπηρεσία απεικόνισης μπορεί να μεταβληθεί κάνοντας χρήση των δύο ακόλουθων τρόπων:

1. Αρχικά, εξαλείφεται η έννοια της υποδομής. Αυτό συμβαίνει μέσω της επανανάπτυξης της εφαρμογής μετακινώντας τις αποθηκευμένες εικόνες από το τοπικό δίκτυο LAN του νοσοκομείου στον κοινό αποθηκευτικό χώρο του νέφους. Αυτό το χαρακτηριστικό γνώρισμα εξαλείφει άμεσα την ανάγκη διατήρησης μεγάλης ικανότητας αποθήκευσης τοπικά. Έπειτα, καθώς οι χρήστες της υπηρεσίας προσπελαίνουν τις εικόνες, ενεργοποιείται ένα χαρακτηριστικό γνώρισμα που απαντάται στα δίκτυα παράδοσης περιεχομένου (Content Delivery Networks) και το οποίο τυπικά ενσωματώνει και ο πάροχος της υπηρεσίας νέφους. Βάσει του χαρακτηριστικού αυτού, το δίκτυο τοποθετεί τα αντίγραφα εικόνων που χρησιμοποιήθηκαν πρόσφατα σε τοποθεσίες που είναι πιο κοντά στους αναγνώστες κι έτσι το σύστημα γίνεται αρκετά γρηγορότερο.
2. Στο δεύτερο στάδιο, εξαλείφεται η τοπική επεξεργασία που σχετίζεται με τις μηχανές απεικόνισης, δηλαδή με τη συλλογή των στοιχείων και των δεδομένων.

Επιπροσθέτως, στο νέο σύστημα όπως απεικονίζεται και στην Εικόνα 28, τα αρχεία δημιουργούνται τοπικά και μεταφέρονται στο νέφος. Επίσης οι εικονικές μηχανές επεξεργάζονται τις απεικονίσεις. Ακόμα το σύστημα ενσωματώνει λειτουργικότητα ουράς μηνυμάτων, έτσι ώστε να παρέχεται μία ομαλή ροή αιτήσεων προς τον εξυπηρετή. Όσον αφορά τις χρονικές περιόδους που διακινείται το μέγιστο φορτίο, το σύστημα δημιουργεί αυτόματα νέα στιγμιότυπα για να χειριστεί το φόρτο εργασίας.



**Εικόνα 28 - Εφαρμογή που αναπτύσσεται εξ ολοκλήρου στο νέφος**

Τέλος, όταν ο εξυπηρετής της εφαρμογής ολοκληρώσει την επεξεργασία απεικόνισης ακολουθεί την εξής διαδικασία:

- Αρχικά, ενημερώνει την ουρά αναμονής μηνυμάτων,
- Στη συνέχεια, καταγράφει το αποτέλεσμα σε μια βάση δεδομένων, και
- Τέλος, παρουσιάζει το αποτέλεσμα σε μια ιστοσελίδα, η οποία κι αυτή με τη σειρά της διατίθεται μέσω του νέφους.

Το σύστημα που αναπτύσσεται στο νέφος είναι πιο αποδοτικό, διότι το σύστημα εκτελεί τις επεξεργασίες πάντα στο βέλτιστο φόρτο εργασίας του. Επίσης, η υποδομή, ο

αποθηκευτικός χώρος και το σύστημα ουράς εξαλείφουν μεγάλο μέρος του κόστους και της λειτουργικής πολυπλοκότητας.

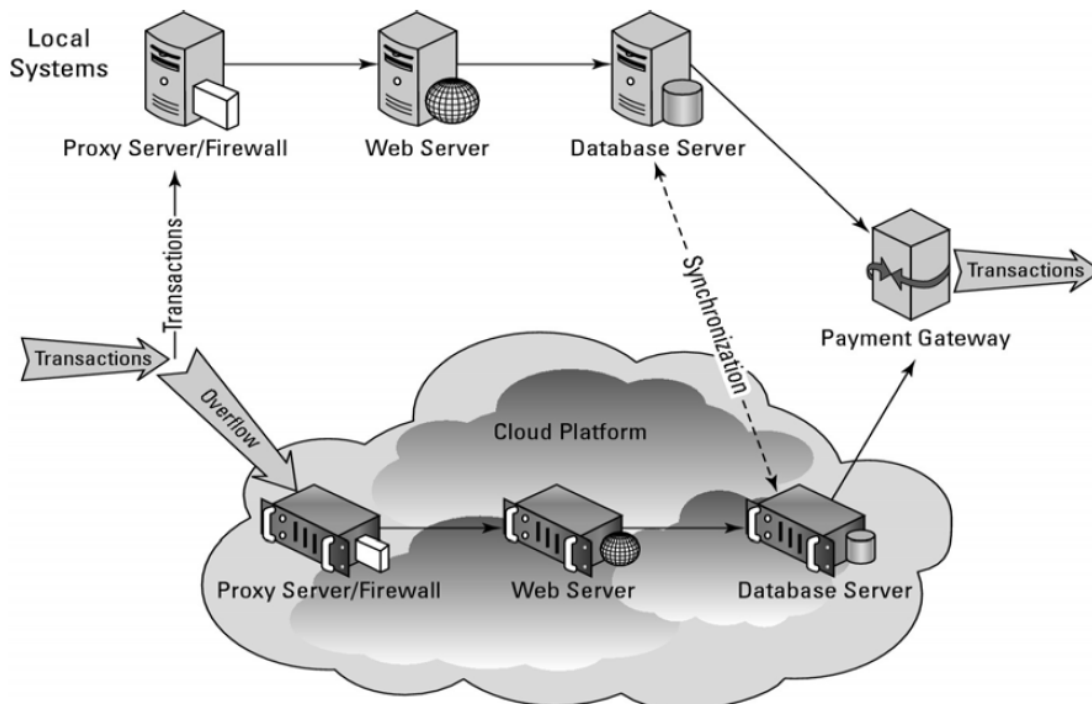
Στη συνέχεια, οι απεικονίσεις είναι διαθέσιμες, πάντα μέσω ενός προγράμματος πλοήγησης και επειδή το σύστημα έχει ικανότητα κλιμάκωσης, η υπηρεσία απεικόνισης μπορεί να επεκταθεί και σε άλλες ιστοσελίδες. Τέλος, ένα πολύ σημαντικό χαρακτηριστικό έχει να κάνει με την εξοικονόμηση χρόνου. Δηλαδή όταν αποφασιστεί να γίνει μετασχηματισμός των εικόνων σε διαφορετικές μορφοποιήσεις (formats), η διαδικασία αυτή μπορεί να πραγματοποιηθεί στον κεντρικό εξυπηρέτη και όχι στα διαφορετικά συστήματα απεικόνισης.

### 5.3.6 Έκρηξη νέφους (cloud bursting)

Οι περισσότερες εφαρμογές νέφους είναι υβριδικές, μιας και ένα μέρος τους βρίσκεται σ' ένα τοπικό σύστημα, ενώ ένα άλλο μέρος τους βρίσκεται στο νέφος. Ένας από τους πιο σημαντικούς λόγους για τον οποίο είναι επιθυμητό αυτό το γεγονός (το οποίο γεγονός ονομάζεται *έκρηξη νέφους*), έχει να κάνει με το ότι το νέφος μπορεί να εξυπηρετήσει ως επιπρόσθετη χωρητικότητα σε περιόδους μεγάλου φόρτου εργασίας.

Παραδείγματα τέτοιων συστημάτων (με μεγάλο όγκο εργασίας σε μικρές χρονικές περιόδους) αποτελούν τα συστήματα επεξεργασίας δοσοληψιών και πιο συγκεκριμένα τα συστήματα κρατήσεων. Σε ένα σύστημα κρατήσεων, υπάρχει ένα συγκεκριμένο επίπεδο, το οποίο γενικά είναι χαμηλό, όπου πραγματοποιούνται δοσοληψίες όλες τις χρονικές στιγμές. Σε συγκεκριμένες χρονικές περιόδους όμως (όπως για παράδειγμα στις περιόδους αργιών, διακοπών, κ.λπ.), η ζήτηση αυξάνεται σημαντικά. Έτσι, αν το σύστημα δημιουργούσε μια υποδομή με σκοπό το χειρισμό της μέγιστης αυτής ζήτησης, τότε η υποδομή αυτή θα υποχρησιμοποιείτο το μεγαλύτερο χρονικό διάστημα.

Επίσης, τα περισσότερα υβριδικά συστήματα σχεδιάζονται με απώτερο σκοπό την κλωνοποίηση του τοπικού συστήματος στο νέφος. Συχνά, υπάρχει ένα μικρό μέρος δραστηριότητας που εξελίσσεται στο μέρος του συστήματος, το οποίο βρίσκεται στο νέφος, αλλά όσο αυξάνεται η ζήτηση, τόσο το συγκεκριμένο μέρος του συστήματος παίρνει επιπρόσθετους πόρους από το νέφος. Η Εικόνα 29 παρουσιάζει ένα απλό σύστημα κρατήσεων που έχει στηθεί για έκρηξη νέφους.



**Εικόνα 29 - Εφαρμογή που διαχειρίζεται την υπερχείλιση δοσοληψιών σ' ένα σύστημα κρατήσεων (παράδειγμα έκρηξης νέφους)**

Συχνά τα συστήματα κρατήσεων απαιτούν οι δοσοληψίες όχι μόνο να έχουν την ιδιότητα της ατομικότητας, αλλά και το σύστημα να είναι συνεπές κατά την εκτέλεση παραλλήλων δοσοληψιών. Για να ικανοποιηθεί αυτή η ανάγκη, πρέπει να δημιουργηθεί ένας επόπτης δοσοληψιών. Στην εικόνα παρουσιάζεται ως διακεκομμένη γραμμή μεταξύ των δύο εξυπηρετών βάσεων δεδομένων, με όνομα “Synchronization”. Ο μηχανισμός αυτός έχει σκοπό την εκτέλεση κλειδώματος εγγραφών σε μία βάση δεδομένων.

Στα περισσότερα συστήματα κρατήσεων, το μεγαλύτερο μέρος της συμφόρησης παράγεται στην ιστοσελίδα του Διαδικτύου καθώς οι χρήστες περιηγούνται στο περιεχόμενό της. Για τον λόγο αυτό, προκειμένου να αντιμετωπιστεί η συμφόρηση αυτή προκύπτουν οι εξής λύσεις:

- Αναδημιουργία της ιστοσελίδας προκειμένου να δημιουργηθούν επιπρόσθετα στιγμιότυπα εξυπηρετών φόρτου εργασίας.
- βελτιστοποίηση στην ιστοσελίδα μέσω της δημιουργίας κάποιου στατικού περιεχομένου ώστε να μην χρειάζονται συνεχώς άσκοπες δοσοληψίες με δυναμικό περιεχόμενο.
- Τέλος, ένα πολύ σημαντικό βήμα αποτελεί ο συγχρονισμός των αλλαγών μεταξύ των τοπικών υπολογιστών και των εξυπηρετών του νέφους προκειμένου να διατηρούνται οι τρέχουσες πληροφορίες.

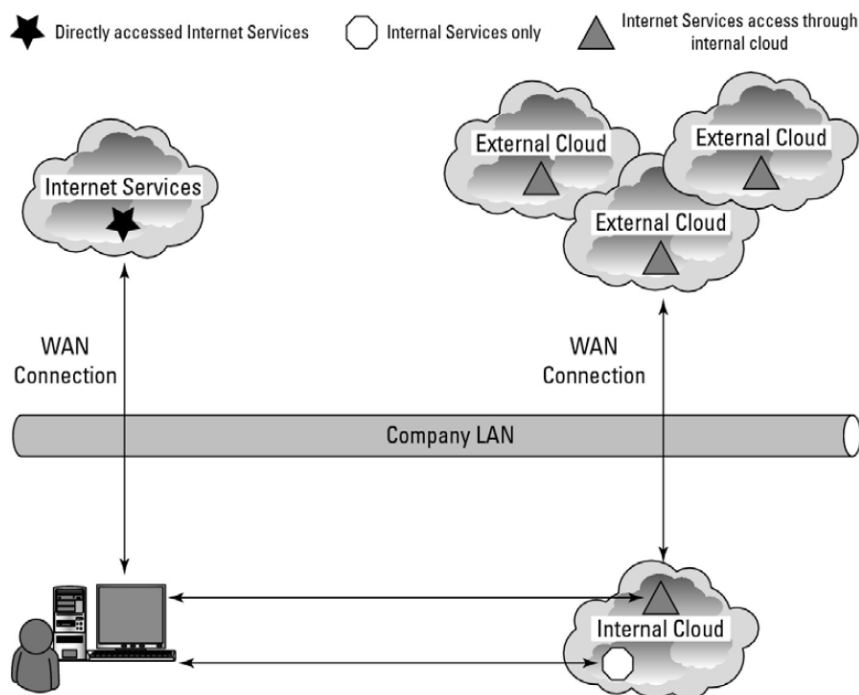
Άλλο ένα μεγάλο πρόβλημα που αντιμετωπίζουν τα συστήματα κρατήσεων, είναι ο συγχρονισμός των πληρωμών και η επικοινωνία με τις επιχειρήσεις πιστωτικών καρτών καθώς και τους χρηματοπιστωτικούς οργανισμούς. Μια πιθανή λύση είναι να μετακινηθεί ολόκληρος ο συγχρονισμός στο νέφος, έτσι ώστε η επεξεργασία των

πληρωμών να μην μπορεί να επηρεάσει τα υπόλοιπα μέρη του συστήματος. Σημειώνεται επίσης ότι δεν παίζει κανένα ρόλο το γεγονός ότι ένας εικονικός εξυπηρέτης εκτελεί τις διαδικασίες πληρωμών, καθώς και το ότι η διαδικασία αυτή δεν έχει καταστάσεις.

Τέλος άξιο λόγου είναι πως οι αρχιτεκτονικές νέφους προσφέρουν αρκετά πλεονεκτήματα, που εκτιμάται ότι προοδευτικά θα υιοθετηθούν από μεγάλους οργανισμούς, ως βασικά μέρη της σχεδίασής τους. Για παράδειγμα, στην Εικόνα 30 παρατηρούνται τα ακόλουθα:

- Ένα εσωτερικό νέφος παρέχει υπηρεσίες δοσοληψιών μεγάλης ταχύτητας στο τοπικό δίκτυο LAN.
- Ένα εξωτερικό νέφος παρέχει υπηρεσίες όσον αφορά άλλες ανάγκες των χρηστών.
- Το νέφος της επιχείρησης αναπαράγεται σε πολλαπλές ιστοσελίδες.

Επιπλέον, οι υπηρεσίες αυτές εξαρτώνται από διάφορους παράγοντες, όπως είναι το κόστος, οι καθυστερήσεις αλλά και η ευκολία.



Εικόνα 30 - Παράδειγμα χρηστών σε μεγάλους οργανισμούς

#### 5.4 Λειτουργικό σύστημα Jolicloud (για Netbook)

Σήμερα η δημοτικότητα των netbooks και των κινητών τηλεφώνων έχει αυξήσει πάρα πολύ τους πιθανούς χρήστες των συσκευών υπολογιστικού νέφους. Μέχρι σήμερα ωστόσο, όλες αυτές οι συσκευές, χρησιμοποιούν τα κλασσικά λειτουργικά συστήματα, όπως είναι τα Windows, Linux και Macintosh όσον αφορά τους υπολογιστές, καθώς και τα Android, το IOS και τα Windows Mobile όσον αφορά τα κινητά τηλέφωνα. Η κύρια



διαφορά που έχουν αυτές οι συσκευές μεταξύ τους συνίσταται στο κατά πόσο είναι σε θέση να εκτελέσουν βίντεο και animation (ειδικότερα το Adobe Flash).

Όσον αφορά την ασφαλή σύνδεση μεταξύ εξυπηρετούμενων και νέφους, μπορούμε να αναφέρουμε ότι καμία από τις προηγούμενες φορητές συσκευές δε μπορεί να συνδεθεί με ασφαλή τρόπο στο νέφος.

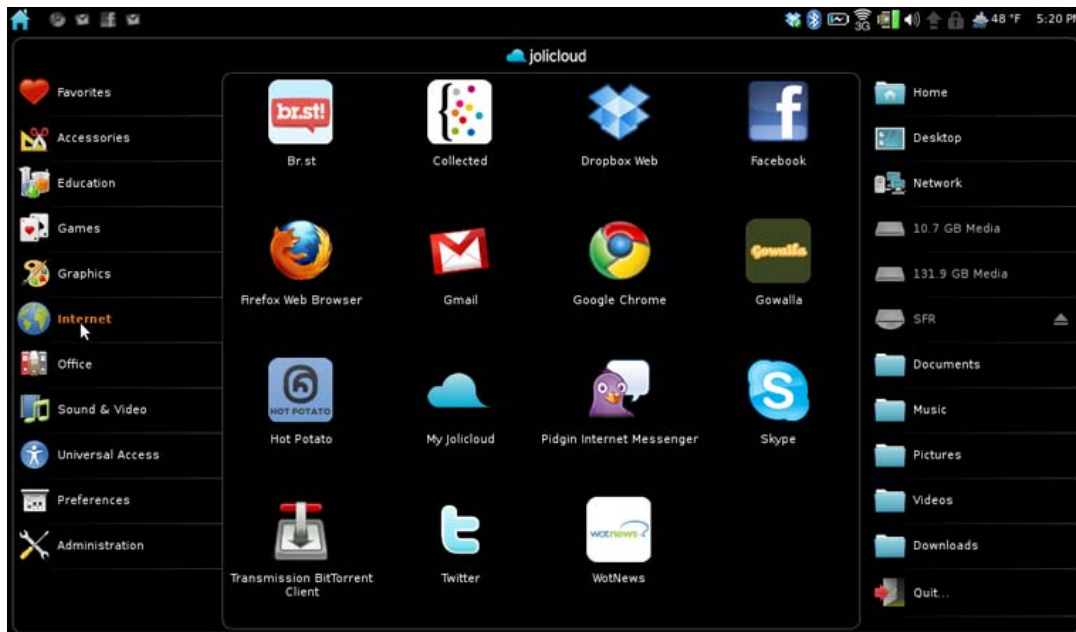
Από το 2008 κιόλας, η εταιρεία Netvides κυκλοφόρησε ένα ελαφρύ λειτουργικό βασισμένο σε Linux. Σκοπός αυτού ήταν οι χρήστες να συνδέονται εύκολα σε εφαρμογές του νέφους. Το όνομα που αυτό έλαβε ήταν το Jolicloud. Είναι ένα ελεύθερο λογισμικό, το οποίο οι χρήστες μπορούν να εγκαταστήσουν εύκολα είτε στο netbook είτε στον υπολογιστή τους καθώς και στο tablet.

Επίσης να αναφερθεί πως το Jolicloud έχει μικρές απαιτήσεις χωρητικότητας και επιτρέπει την αποθήκευση αρχείων των χρηστών και πρόσβαση σε αυτά από οποιαδήποτε συσκευή οπουδήποτε βρίσκεται ο εκάστοτε χρήστης. Το Jolicloud μπορεί να χρησιμοποιηθεί ως μοναδικό λειτουργικό σύστημα σε ένα netbook ή ακόμα και να χρησιμοποιηθεί μαζί με τα Windows. Κατά την τελευταία έκδοση, το Jolicloud μετονομάστηκε σε JoliOS και υποστηρίζει την έκδοση flash 10.2

Το Jolicloud επιπρόσθετα επικεντρώνεται στη κατασκευή μιας κοινωνικής πλατφόρμας με αυτόματες ενημερώσεις και εγκαταστάσεις λογισμικού. Η εφαρμογή είναι κατασκευασμένη με χρήση της HTML 5 και έχει ήδη προ-εγκατεστημένα τα ακόλουθα προγράμματα και εφαρμογές:

- Gmail.
- Skype.
- Twitter.
- Firefox.
- Dropbox.
- VLC.
- Flash.
- Picasa.

Οποιοδήποτε πρόγραμμα πλοήγησης υποστηρίζει την HTML 5 μπορεί να χρησιμοποιηθεί για τη διεπαφή με το Jolicloud. Επίσης, Το Jolicloud διατηρεί μια βιβλιοθήκη ή αλλιώς έναν κατάλογο εφαρμογών με περισσότερες από 700 εφαρμογές (ως μέρος ενός καταστήματος εφαρμογών, app store).



*Εικόνα 31 - Εικόνα από το λειτουργικό σύστημα Jolicloud*

## **5.5 Λειτουργικό σύστημα Chromium (το πρόγραμμα πλοήγησης ως λειτουργικό σύστημα)**

Το Chrome OS είναι ένα λειτουργικό σύστημα της Google το οποίο είναι βασισμένο σε Linux. Εν αντιθέσει με άλλες διανομές Linux, το Chrome OS δεν είναι ένα ελεύθερο λογισμικό αλλά παρέχεται ήδη εγκατεστημένο σε συγκεκριμένο υλικό. Το υλικό αυτό είναι προϊόν κατασκευαστών πρωτότυπου εξοπλισμού (OEMs) εγκεκριμένους από τη Google. Ένας κατασκευαστής OEM δημιουργεί συστήματα από διαφορετικά στοιχεία και τα διαθέτει στην αγορά κάτω από ένα εμπορικό όνομα. Με τον ίδιο τρόπο είναι κατασκευασμένο και το λειτουργικό σύστημα Android που είναι εγκατεστημένο σε πληθώρα κινητών τηλεφώνων.

Το Chrome OS λοιπόν, παρέχεται μέσω ενός φορητού υπολογιστή που καλείται Chromebook. Η έμπνευση για τη σχεδίαση και τη δημιουργία του λειτουργικού αυτού συστήματος προήλθε μετά τη δημιουργία του προγράμματος πλοήγησης Google Chrome και την μεγάλη απήχηση αυτού στους χρήστες. Έτσι η Google θέλησε να παρέχει ένα λειτουργικό που να διαθέτει παρόμοια χαρακτηριστικά γνωρίσματα με το Chrome και να είναι ιδιαίτερα αποδοτικό. Εκτός αυτού, πρόθεση των σχεδιαστών είναι να το κάνουν άμεσο ανταγωνιστή των πολύ διάσημων στην αγορά, iPad της Apple.

Όσον αφορά την ομοιότητα με το πρόγραμμα πλοήγησης Google Chrome, η διεπαφή χρήστη είναι παρόμοια με την εφαρμογή πλοήγησης ιστού Chrome και περιλαμβάνει ένα σύστημα αναπαραγωγής πολυμέσων που αναπαράγει ήχους κωδικοποιημένους σε MP3 και εικόνες σε μορφή JPEG τόσο online όσο και offline. Επιπρόσθετα, είναι ενσωματωμένο το Adobe Flash με τον ίδιο ακριβώς τρόπο που είναι ενσωματωμένο και στο πρόγραμμα πλοήγησης. Όταν κάποιος χρήστης εκκινεί τον Chrome, έχει τη δυνατότητα να δει συνδέσεις στις σημαντικότερες εφαρμογές νέφους της Google, όπως είναι το Gmail, τα Google Apps, το YouTube, καθώς και ιστοσελίδες όπως το Facebook,

το Hulu, το Pandora, το Twitter, κ.λπ. Επίσης, μέσω αυτού του λειτουργικού παρέχεται στους χρήστες γρήγορη και εύκολη πρόσβαση στο νέφος καθώς και ασφάλεια των δεδομένων τους που αποθηκεύονται σε αυτό. Οι προδιαγραφές υλικού του λειτουργικού συστήματος Chrome περιλαμβάνουν μια μονάδα (module) εμπιστευτικής πλατφόρμας (Trusted Platform Module), η οποία μαζί με ένα διακόπτη (switch), παρέχει ένα όσο το δυνατόν περισσότερο έμπιστο μονοπάτι το οποίο μπορεί να χρησιμοποιείται για την εκκίνηση του συστήματος σ' ένα μοντέλο ανάπτυξης. Θα μπορούσαμε να πούμε πως είναι ένα αρκετά ασφαλές λειτουργικό και αυτό γιατί συμπεριλαμβάνεται μια έκδοση λογισμικού απομακρυσμένης σύνδεσης, που δημιουργεί μια κρυπτογραφημένη σύνδεση όπως κάνουν το RDP της Microsoft, το ICA του Citrix ή ένας εξυπηρετούμενος VNC.

Κάτι το οποίο χαρακτηρίζει το λειτουργικό Chrome OS, είναι η γρήγορη εκκίνηση του συστήματος, η οποία επιτυγχάνεται σε ελάχιστα δευτερόλεπτα. Αυτό συμβαίνει διότι η συγκεκριμένη συσκευή είναι απαλλαγμένη από τις περισσότερες συσκευές των σύγχρονων συστημάτων υπολογιστών.

Βέβαια επειδή όπως αναφέρθηκε νωρίτερα η πρόσβαση των χρηστών στο Chrome OS γίνεται μόνο μέσω των Chromebook, αποφασίστηκε από την Google να αναπτύξει ένα ελεύθερο λειτουργικό σύστημα βασισμένο στο Chrome OS. Έτσι δημιουργήθηκε το Chromium OS το οποίο παρέχει σε κάθε χρήστη τη δυνατότητα να το κατεβάσει και να το εγκαταστήσει στον υπολογιστή του και να βιώσει την εμπειρία της αυτόματης σύνδεσης με τις εφαρμογές νέφους. Η αρχιτεκτονική του Chromium OS βασίζεται σ' ένα σύστημα τριών επιπέδων:

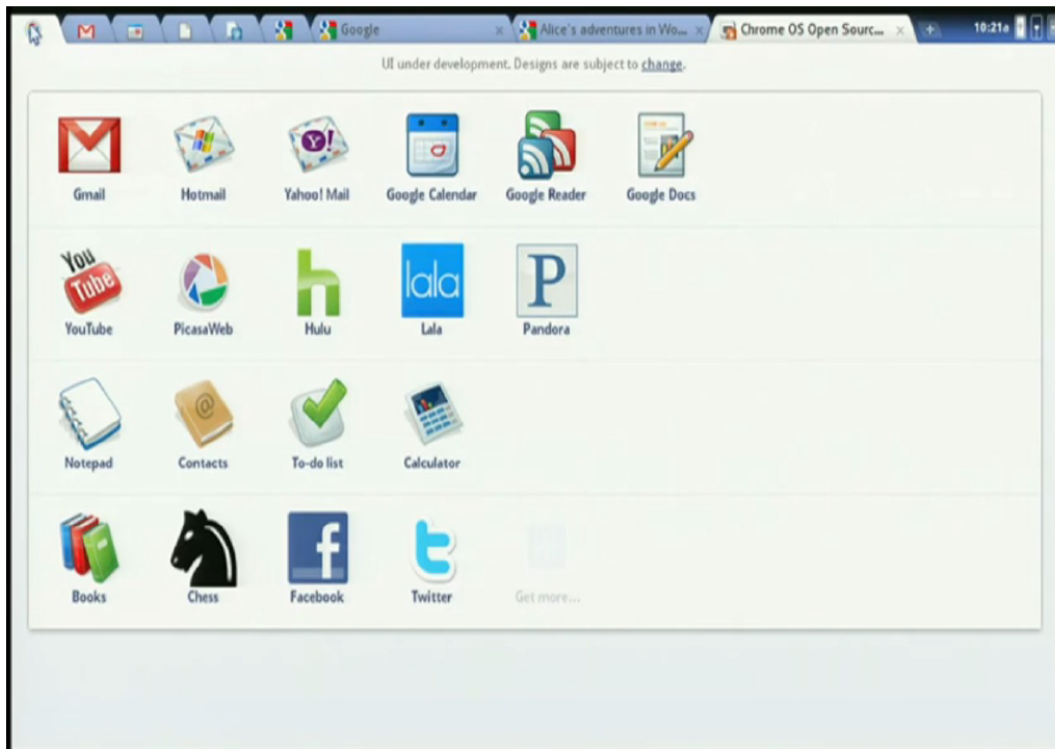
- Το επίπεδο υλικού,
- Την εφαρμογή πλοήγησης και
- Ένα σύνολο λογισμικού και βοηθητικών προγραμμάτων του συστήματος.

Το Chromium OS έχει υιοθετήσει ένα σύνολο ρουτινών ασφάλειας στο firmware του, οι οποίες εκτελούνται κατά τη διάρκεια της εκκίνησης και αποθηκεύουν τις πληροφορίες που είναι απαραίτητες για την εκτέλεση της αποκατάστασης του συστήματος.

Στη φάση αυτή να αναφερθεί πως και στο Chrome OS καθώς και στο Chromium OS περιλαμβάνεται η υπηρεσία Cloud Print της Google, που επιτρέπει σε μια εφαρμογή να εκτυπώσει σε οποιοδήποτε συνδεδεμένο εκτυπωτή χωρίς να χρειάζεται να κάνει χρήση κάποιου οδηγού εκτυπωτών (printer driver). Έτσι δεν χρειάζεται να αναπτυχθούν υποσυστήματα υλικού και λογισμικού σχετικά με εκτυπώσεις. Αντ' αυτού, εγκαθίσταται ένας εξυπηρετής διαμεσολάβησης (proxy server) που καταχωρεί σ' έναν εκτυπωτή την υπηρεσία και διαχειρίζεται τις εκτυπώσεις των χρηστών.

Τελειώνοντας θα μπορούσε να αναφερθεί πως μεταξύ των δυο αυτών λειτουργικών υπάρχει μια διάφορα. Στο Chrome OS ο χρήστης δεν θα χρειαστεί να μπει σε διαδικασία αναβάθμισης αλλά και συντήρησης αυτού. Εν αντιθέσει στο Chromium OS ο χρήστης είναι αυτός ο οποίος θα πρέπει να εγκαταστήσει τις αναβαθμίσεις και να συντηρεί το λογισμικό.

Στην Εικόνα 32 που ακολουθεί, απεικονίζεται το λειτουργικό σύστημα Chrome με τις πολλαπλές ετικέτες.



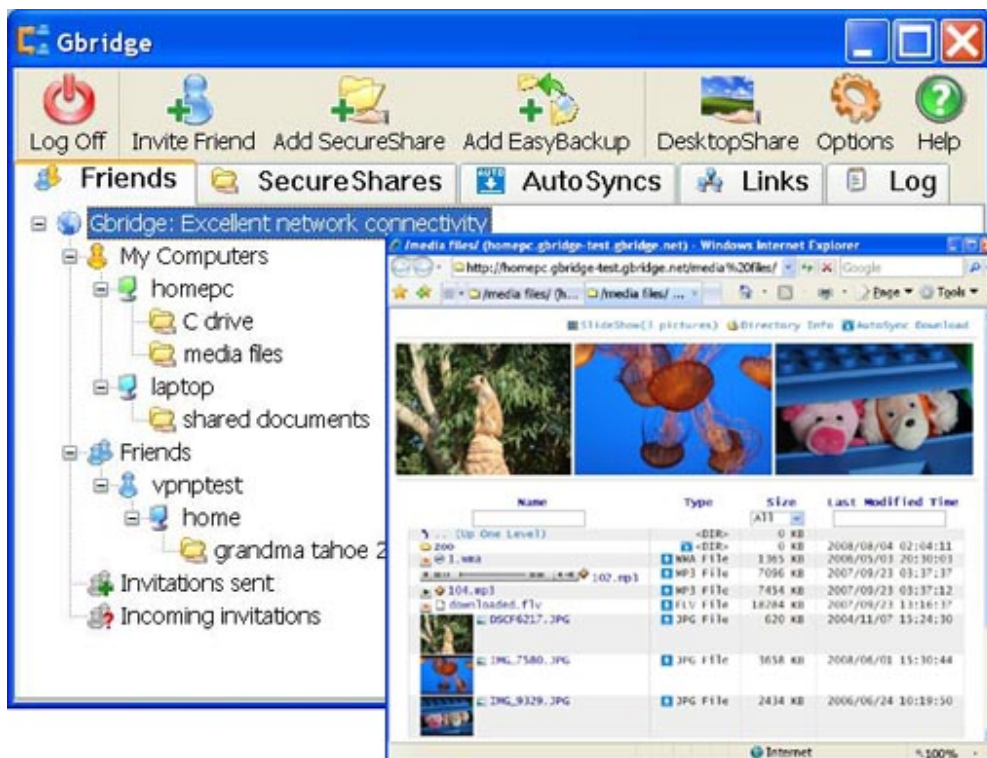
*Εικόνα 32 - Λειτουργικό σύστημα Chrome*

## 5.6 Λογισμικό Gbridge

Το λογισμικό Gbridge, είναι μια ενδιαφέρουσα λύση η οποία απεικονίζει τη χρήση του εικονικού ιδιωτικού δικτύου (Virtual Private Network, VPN) σε μια σύνδεση νέφους. Το Gbridge λειτουργεί με εφαρμογές που αναπτύσσονται με χρήση των Google Apps. Έτσι καθίσταται εφικτή η ασφαλής σύνδεση με τις εφαρμογές αυτές με τη χρήση ενός εικονικού ιδιωτικού δικτύου.

Αρχικά για να χρησιμοποιήσει κάποιος το λογισμικό αυτό, χρειάζεται να διαθέτει έναν λογαριασμό στο Gmail. Επιπλέον χρειάζεται η εγγραφή στο δίκτυο GoogleTalk και η σύνδεση μ' ένα άλλο υπολογιστή με χρήση του λογαριασμού Google. Το Gbridge επιτρέπει επίσης σε επιπρόσθετους χρήστες να συνδεθούν όταν τους προσκαλέσει κάποιος άλλος και υποστηρίζει συνεργατικά χαρακτηριστικά γνωρίσματα, όπως το διαμοιρασμό οθονών με χρήση του λογισμικού Virtual Network Computing (VNC), τη συνομιλία, τη «ζωντανή» περιήγηση σε φακέλους υπολογιστή, το συγχρονισμό φακέλων, καθώς και την αυτοματοποιημένη λήψη αντιγράφων ασφαλείας.

Η Εικόνα 33 παρουσιάζει την περιήγηση ενός φακέλου με χρήση μιας σύνδεσης VPN που χρησιμοποιεί το χαρακτηριστικό γνώρισμα Secure Shares του Gbridge.



*Εικόνα 33 - Χαρακτηριστικό γνώρισμα SecureShares του Gbridge για ασφαλή σύνδεση ενός υπολογιστή με άλλον με χρήση του Gtalk*

## Κεφάλαιο 6: Θέματα ποιότητας υπηρεσίας στο υπολογιστικό νέφος

### 6.1 Πρότυπα υψηλής κλιμάκωσης της εφαρμογής που ταιριάζουν καλύτερα στο νέφος

Τα πρότυπα τα οποία θα αναλυθούν σ' αυτό το κεφάλαιο, σχετίζονται άμεσα με την δημιουργία υψηλής κλιμάκωσης σε εφαρμογές. Αυτά λοιπόν είναι τα ακόλουθα τέσσερα:

- Δυνατότητα Μεταφοράς (Transference)
- Κλιμάκωση σε Επίπεδο Διαδικτύου (Internet scale)
- Έκρηξη υπολογισμού (Burst compute)
- Ελαστική αποθήκευση (Elastic storage)

Αυτά τα πρότυπα αποτελούν τα πιο κοινά πρότυπα μετακίνησης μιας εφαρμογής στο νέφος και παρέχουν συγκεκριμένα πολύ σημαντικά πλεονεκτήματα, όπως είναι η υποστήριξη πληθώρας χρηστών, δυνατότητα εκτέλεσης μεγαλύτερου αριθμού υπολογισμών, καθώς και ακόμα μεγαλύτερη ποσότητα δεδομένων.

#### 6.1.1 Δυνατότητα Μεταφοράς

Η έννοια της μεταφοράς, έχει να κάνει με τη διαδικασία η οποία μετακινεί αυτούσια στο νέφος μία υπάρχουσα εφαρμογή. Αυτό πολύ πιθανόν να είναι επιθυμητό για οικονομικούς λόγους, μιας και είναι πιο χαμηλό το κόστος χρήσης των πόρων ενός νέφους, απ' ότι είναι η αντίστοιχη εκτέλεση της συγκεκριμένης εφαρμογής σ' ένα εξυπηρετή, εφόσον βέβαια δεν υιοθετούνται πλήρως εικονικές υπηρεσίες στο τοπικό κέντρο δεδομένων.

Θα πρέπει να δοθεί ιδιαίτερη σημασία στις προσαρμογές που σχετίζονται με το τοπικό κέντρο δεδομένων και που δεν υποστηρίζει το νέφος. Για παράδειγμα, εάν η εφαρμογή εξαρτάται από έναν οδηγό συσκευών για Linux, δε θα μπορεί να πραγματοποιηθεί η μεταφορά της εφαρμογής στο νέφος. Αυτό συμβαίνει μιας και οι πάροχοι δε θα επιτρέψουν να γίνει επαναδημιουργία και τροποποίηση του οδηγού συσκευών.

Κάποια παραδείγματα εφαρμογών για το πρότυπο μεταφοράς αποτελούν το ηλεκτρονικό ταχυδρομείο, τα συστήματα διαχείρισης σχέσης πελατών CRM (Customer Relationship Management), κ.λπ.

Ένα παράδειγμα μετακίνησης μιας εφαρμογής στο νέφος αποτελεί μια μικρή επιχείρηση, το Zillow.com. Αυτή είναι μια ιστοσελίδα, που παρέχει σε καταναλωτές πληροφορίες για αγορές ακινήτων και σπιτιών σε οποιαδήποτε διεύθυνση των Η.Π.Α. Η συγκεκριμένη επιχείρηση, δε μπορούσε να πληρώσει όλους τους εξυπηρετές που χρειάζονταν για την επεξεργασία των αλλαγών στις τιμές των ακινήτων, λόγω του ότι οι τιμές αλλάζουν συνεχώς και συνεπώς ο φόρτος επεξεργασίας είναι μεγάλος. Επομένως, για να επιτύχουν μεγάλη ανταγωνιστικότητα και ταυτόχρονα να μην «εκτοξευθούν» τα έξοδα της επιχείρησης, έλαβαν την απόφαση να μετακινήσουν ολόκληρη την εφαρμογή στο νέφος.

### 6.1.2 Κλιμάκωση σε Επίπεδο Διαδικτύου

Το πρότυπο αυτό περιλαμβάνει τη δημιουργία μιας εφαρμογής νέφους, η οποία διαθέτει τη δυνατότητα να χειριστεί ένα τεράστιο αριθμό χρηστών. Παραδείγματα αυτού του προτύπου, αποτελούν οι εφαρμογές YouTube, Flickr, Facebook και Twitter. Το πρότυπο αυτό είναι πολύ κοινό για νέες εφαρμογές, για το λόγο του ότι δεν απαιτείται μεγάλο κόστος έναρξης, μια και δε χρειάζεται να αγοραστεί κανένας εξυπηρέτης. Επίσης, δίνει τη δυνατότητα επέκτασης όταν απαιτηθεί από τη μεριά των χρηστών.

Ένα παράδειγμα το οποίο αναλύεται είναι το ευρέως γνωστό και για τη χρήση του Facebook, όπου αρχικά εκτελούνταν σε έναν εξυπηρέτη και εξυπηρετούσε μόνο φοιτητές του Χάρβαρντ. Βέβαια στη συνέχεια, δημιουργήθηκε το δικό του κέντρο δεδομένων με σκοπό την ταυτόχρονη εξυπηρέτηση και υποστήριξη 400 εκατομμυρίων χρηστών. Επίσης, μια άλλη παρεμφερής εφαρμογή αποτελεί το εξίσου ευρέως γνωστό Twitter, το οποίο επίσης αρχικά εκτελούνταν σε έναν εξυπηρέτη και επειδή διαδόθηκε παγκοσμίως, οι χρήστες του αυξήθηκαν με έναν πρωτοφανή και παράλληλα γοργό ρυθμό. Βέβαια, αντιμετώπισε πολλές διακοπές λειτουργίας που οφείλονταν σε κάποια προβλήματα κλιμάκωσης.

Επομένως, σε περίπτωση που κάνει έναρξη των δραστηριοτήτων της μια μικρή επιχείρηση (με μεγάλο όμως όραμα), θα πρέπει να σκεφτούν οι δημιουργοί της να την εντάξουν από την αρχή στο νέφος κάνοντας χρήση μιας ισχυρής σχεδίασης, αλλά και προτύπων κλιμάκωσης. Επίσης, η σχεδίαση μιας εφαρμογής για την ένταξή της από την αρχή στο νέφος, ενέχει ρίσκο σε περιπτώσεις εφαρμογών με απρόβλεπτη αύξηση των χρηστών της. Μια από τις μεγαλύτερες προκλήσεις σχεδίασης γι' αυτό το πρότυπο κλιμάκωσης, σχετίζεται με τη δομή των βάσεων δεδομένων. Πιο συγκεκριμένα, χωρίς την απαραίτητη προσοχή, μια βάση δεδομένων μπορεί γρήγορα να αποτρέψει την κλιμάκωση.

### 6.1.3 Έκρηξη υπολογισμού

Σε ό,τι αφορά τις εφαρμογές που εφαρμόζουν την έκρηξη υπολογισμού, αυτές έχουν τη δυνατότητα χειρισμού δυναμικής προσθήκης πόρων, οποτεδήποτε αυτοί απαιτούνται. Το σημαντικό είναι ότι οι πόροι διαστασιοποιούνται σωστά, χωρίς να υποχρησιμοποιούνται αλλά ούτε και να υπερχρησιμοποιούνται. Ένα αντιπροσωπευτικό παράδειγμα αυτού, αποτελεί η ιστοσελίδα <http://eventseer.net/>. Συνήθως χρησιμοποιεί το νέφος του Amazon, για παραγωγή στατικής μορφής των ιστοσελίδων που έχουν τροποποιηθεί (περιέχει 600.000 ιστοσελίδες), επειδή δεν έχει την εσωτερική ικανότητα να επιτελέσει αυτή τη λειτουργία στους δικούς της εξυπηρέτες αρκετά γρήγορα.

Το πρότυπο της έκρηξης υπολογισμού, σχετίζεται άμεσα με οικονομικούς παράγοντες που αφορούν το νέφος. Το κόστος της επιπρόσθετης ικανότητας σε υλικό, που απαιτείται για να υποστηρίξει αυτό το σχέδιο, είναι απαγορευτικό. Εντούτοις, οι εκρήξεις του φόρτου εργασίας μπορούν να αντιμετωπιστούν επαρκώς με χρήση του μοντέλου νέφους “πληρωμή για τη διάρκεια χρήσης” (pay only for what you use).

### 6.1.4 Ελαστική αποθήκευση

Οι εφαρμογές έχουν τη δυνατότητα να αυξηθούν εκθετικά, από τη σκοπιά του αποθηκευτικού χώρου. Αν και η τοπική αποθήκευση είναι σχετικά φθηνή, δε παύει η

διαχείρισή της να είναι αρκετά δαπανηρή. Κατά συνέπεια, η χρησιμοποίηση μιας πλατφόρμας νέφους μπορεί να είναι μια φθηνότερη εναλλακτική λύση, όσον αφορά τη διαχείριση της τοπικής αποθήκευσης.

Ωστόσο, η χρήση αυτού του προτύπου απαιτεί προσεκτική σχεδίαση για την πρόσβαση στα δεδομένα. Για παράδειγμα, εάν κάποιος χρησιμοποιεί το νέφος μόνο για την αποθήκευση δεδομένων και ταυτόχρονα τα επεξεργάζεται τοπικά, η απόδοση μπορεί να γίνει απαράδεκτα χαμηλή. Έτσι το πρότυπο αυτό θα γινόταν πρακτικά μη εφαρμόσιμο.

## 6.2 Αξιοπιστία νεφών

Σε όλους τους κλάδους της μηχανικής, η αξιοπιστία αποτελεί την ικανότητα ενός συστήματος να εκτελέσει τις λειτουργίες που απαιτούνται κάτω υπό συγκεκριμένες συνθήκες, για μια προκαθορισμένη χρονική περίοδο. Στο λογισμικό, η αξιοπιστία αφορά τη δυνατότητα μιας εφαρμογής να εκτελεί απρόσκοπτα και σωστά όλες τις λειτουργίες μέχρι την ολοκλήρωσή της συγκεκριμένης εφαρμογής. Για να είναι αυτό εφικτό, θα πρέπει και τα στοιχεία από τα οποία εξαρτάται (εξυπηρετές, σκληροί δίσκοι, ρεύμα κ.λπ.) να διαθέτουν και αυτά αντίστοιχο επίπεδο αξιοπιστίας.

Ωστόσο, η ολοκλήρωση καθορίζεται από τον σχεδιαστή της εφαρμογής. Πιο συγκεκριμένα, ακόμη και με τη χρήση του τέλει λογισμικού και χωρίς να υφίσταται το παραμικρό σφάλμα (bug) στα συστήματα λογισμικού, οι εφαρμογές που χρησιμοποιούν χιλιάδες εξυπηρετές θα οδηγηθούν σε αποτυχίες σε κάποιο κομμάτι του υλικού και ένας αριθμός στιγμιότυπων θα αποτύχει. Κατά συνέπεια, θα αποτύχει και η εφαρμογή που βασίζεται σε αυτά τα στιγμιότυπα.

### 6.2.1 Πλεονασμός (Redundancy)

Ο πλεονασμός αποτελεί τον σημαντικότερο επιπρόσθετο παράγοντα σχεδίασης μιας αξιόπιστης εφαρμογής στο νέφος. Πολλές τεχνικές σχεδίασης για να επιτύχουν υψηλή αξιοπιστία, εξετάζουν τα εξής:

- Το λογισμικό,
- Τα δεδομένα, καθώς και
- Το υλικό πλεονασμού

Ένα τέτοιο παράδειγμα αποτελεί η NASA, η οποία χτίζει για δεκαετίες συστήματα που αποκαλούνται “triple modular redundancy with spares”. Η έννοια αυτή συμβολίζει, ότι τρία αντίγραφα ενός κρίσιμου συστήματος είναι ενεργά και παράγουν απαντήσεις που λαμβάνονται από ένα άτομο ή σύστημα, το οποίο έχει τη δυνατότητα να αποφασίζει την τελική απάντηση. Επίσης, αν αποτύχει ένα ή περισσότερα από τα ενεργά συστήματα, μπορεί να αντικατασταθεί από κάποιο εφεδρικό και να τεθεί άμεσα σε λειτουργία.

Τέτοιου είδους πλεονασμός, εξασφαλίζει την άμεση αποκατάσταση από ανιχνευμένες συνθήκες αποτυχίας. Επιπλέον, αυτές οι τεχνικές μπορεί να είναι ακριβές και σύνθετες, όμως μπορεί να μειωθεί το κόστος τους εφόσον αναπτυχθούν στο νέφος, εξαιτίας του ότι πληθώρα λειτουργιών και πόρων, παρέχονται από τους ίδιους τους παρόχους νεφών.

Για τα πλεονάζοντα στοιχεία λογισμικού, το φαινόμενο μπορεί να αποτελείται από διπλά ή τριπλά πλεονάζοντα τμήματα λογισμικού (όπου είναι τμήματα της εφαρμογής) και τα



οποία εκτελούνται παράλληλα με κοινούς έλεγχους επικύρωσης. Μια ιδέα αποτελεί η ύπαρξη των στοιχείων που αναπτύσσονται από διαφορετικές ομάδες με βάση τις ίδιες προδιαγραφές. Αυτή η προσέγγιση μπορεί μεν να κοστίζει περισσότερο, αλλά μπορεί να απαιτηθεί από την ίδια την εφαρμογή για λόγους αξιοπιστίας. Επίσης, επειδή κάθε στοιχείο σχεδιάζεται για να εκτελέσει την ίδια λειτουργία, οι αποτυχίες των ταυτόχρονων και ίδιων στοιχείων ανακαλύπτονται εύκολα και διορθώνονται κατά τη διάρκεια της δοκιμής της ποιότητας.

Παρ' όλο που τα στοιχεία λογισμικού, παρέχουν μια διαδικασία εγγύησης της ποιότητας, με έναν τρόπο αρκετά έξυπνο για την επικύρωση της ακρίβειας υπηρεσιών, εντούτοις ορισμένες εφαρμογές, μπορούν να θελήσουν να αναπτύξουν τον πλεονασμό των στοιχείων στο περιβάλλον της παραγωγής. Τέτοιου είδους πλεονασμός στοιχείων στον χρόνο εκτέλεσης, μπορεί να είναι χρήσιμος για καταστάσεις τέτοιες, όπου το περιβάλλον είναι κατά μια έννοια “επικίνδυνο” και “επιβλαβές” και δε μπορεί να ελεγχθεί αυστηρά. Σε “επικίνδυνες” καταστάσεις, οι πολλαπλές παράλληλες διαδικασίες μιας εφαρμογής μπορούν να παρέχουν ελέγχους εγκυρότητας μεταξύ τους και να επιτρέψουν στην πλειοψηφία να αποφασίσει. Επιπλέον, είναι θετικό το γεγονός ότι αν και είναι αλήθεια ότι τα στοιχεία λογισμικού προκαλούν επιπρόσθετη κατανάλωση πόρων, το αντιστάθμισμα (tradeoff) μεταξύ της αξιοπιστίας και του κόστους του επιπρόσθετου υλικού είναι πολύ καλύτερο και βοηθάει τους χρήστες.

Όπως έχει ήδη αναφερθεί σε προηγούμενα κεφάλαια, η χρήση παρόχου νέφους παρουσιάζει αρκετά μικρότερη πολυπλοκότητα σε σχέση με την τοπική εγκατάσταση, διότι ο πάροχος νέφους διαθέτει ενσωματωμένη υποδομή. Μια άλλη τεχνική σχεδίασης η οποία είναι βασισμένη εξ ολοκλήρου στον πλεονασμό, περιλαμβάνει τη χρήση των κάτωθι υπηρεσιών:

- Συσταδοποίηση (clustering), όπου με αυτή την υπηρεσία συνδέονται πολλοί υπολογιστές μεταξύ τους με σκοπό να ενεργήσουν ως ένας ενιαίος ταχύτερος υπολογιστής.
- Εξισορρόπηση φόρτου εργασίας, ο φόρτος εργασίας διαμοιράζεται μεταξύ πολλαπλών υπολογιστών, κατά το δυνατόν σε μερίδια ανάλογα προς τη δυναμικότητα του κάθε μεμονωμένου υπολογιστή.
- Αντιγραφή δεδομένων με σκοπό την ανεξάρτητη, αλλά και παράλληλη επεξεργασία πολλαπλών και ίδιων αντιγράφων δεδομένων.
- Προστασία σύνθετων λειτουργιών με δοσοληψίες, για την εξασφάλιση της ακεραιότητας των διαδικασιών και της συνέπειας των δεδομένων.

Το υλικό πλεονασμού αποτελεί μια από τις πιο δημοφιλείς στρατηγικές για την παροχή αξιόπιστων συστημάτων. Το υλικό αυτό μπορεί να περιλαμβάνει πλεοναστικές υποδομές για:

- την αποθήκευση δεδομένων (συστοιχίες δίσκων RAID – redundant array of inexpensive disks),
- τις διεπαφές δικτύου, καθώς και
- τις παροχές ηλεκτρικού ρεύματος.

Επίσης, με αυτού του είδους την υποδομή υλικού, μπορούν να εμφανιστούν μεμονωμένες αποτυχίες στοιχείων χωρίς να μπορούν να επηρεάσουν τη συνολική αξιοπιστία της εφαρμογής. Εν κατακλείδι, είναι σημαντικό να χρησιμοποιηθούν τυποποιημένα στοιχεία υλικού ώστε να επιτραπεί η εύκολη εγκατάσταση καθώς και αντικατάσταση.

## 6.2.2 MapReduce

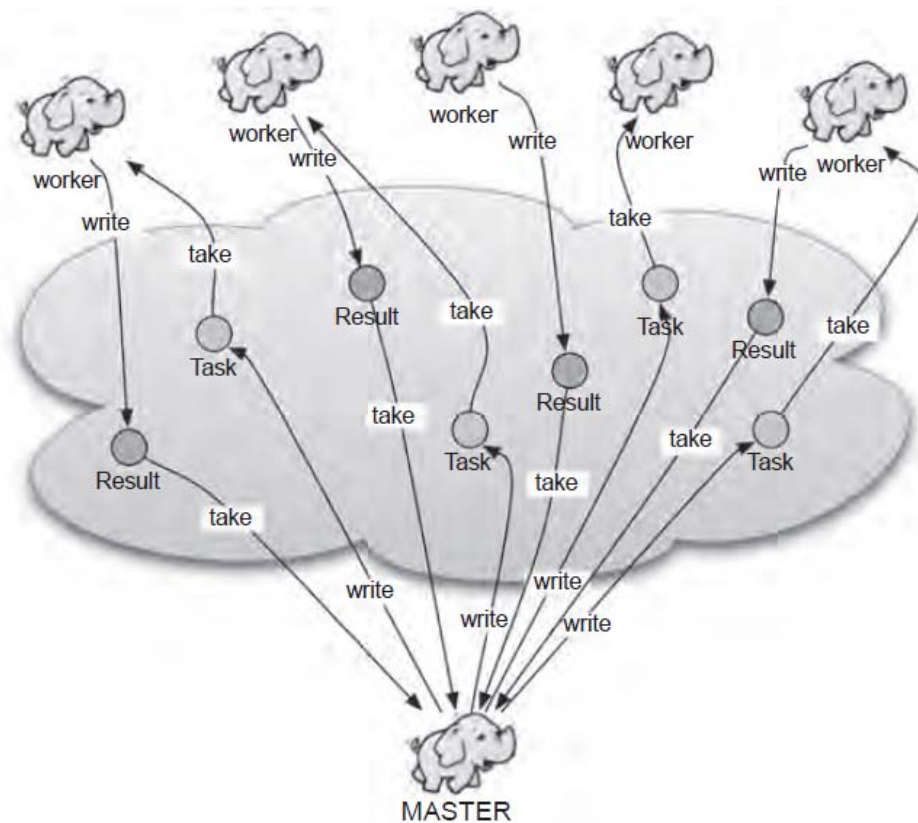
Το MapReduce είναι ένα πλαίσιο εργασίας λογισμικού, το οποίο αναπτύχθηκε από την Google, έχοντας σαν σκοπό την επίλυση του προβλήματος αναζήτησης σε όλο το περιεχόμενο του Παγκόσμιου Ιστού (ο οποίος Ιστός στο τέλος του 2008 είχε περισσότερα από ένα τρισεκατομμύριο μοναδικά URLs). Το MapReduce αποτελεί ένα χαλαρά συνδεδεμένο και κατανεμημένο υπολογισμό μεγάλης κλίμακας, όπου λειτουργεί σε μεγάλα σύνολα δεδομένων και τα οποία χρησιμοποιούνται με τη σειρά τους από συστάδες παραδοσιακών (δηλαδή φθηνών κι απλών) υπολογιστών. Ο λόγος που χρησιμοποιούνται συστάδες υπολογιστών, είναι επειδή ο αριθμός των URLs (Uniform Resource Locator) αυξήθηκε αλματωδώς, και η Google στόχευε να σταματήσει να απασχολείται με θέματα αξιοπιστίας υλικού και να ασχοληθεί περισσότερο με την αξιοπιστία που παρέχεται από το λογισμικό.

Ας σημειωθεί εδώ, πως ακόμη και αν γινόταν χρήση δεκάδων χιλιάδων εξυπηρετών με άριστα στατιστικά αξιοπιστίας, η Google δε θα επιτύγχανε τον επιθυμητό στόχο. Αυτό θα συνέβαινε επειδή, ακόμη και αν καθένας από τους μεμονωμένους εξυπηρετές διέθετε άριστα στατιστικά αξιοπιστίας, αποτέλεσμα αυτού θα ήταν να υπάρξουν σίγουρα πολλαπλές αποτυχίες ανά ημέρα, καθώς μια μηχανή θα βρίσκεται στο μέσο χρόνο αποτυχίας της (mean-time-to-failure) (η πιθανότητα να μην αποτύχει κανείς από  $n$  υπολογιστές με πιθανότητα αποτυχίας  $p$  είναι  $(1-p)^n$ : οσοδήποτε μικρό και αν είναι το  $p$ , η πιθανότητα μειώνεται σημαντικά καθώς το  $n$  αυξάνεται). Επομένως, γίνεται προφανές ότι ο μόνος τρόπος κατασκευής ενός αξιόπιστου συστήματος, είναι η ύπαρξη ενός λογισμικού προετοιμασμένου να αντιμετωπίσει αυτές τις αποτυχίες.

Το όνομα MapReduce έχει τις ρίζες του στον συναρτησιακό προγραμματισμό, που περιλαμβάνει τις λειτουργίες map και reduce, οι οποίες με τη σειρά τους πρωτοεμφανίστηκαν στη γλώσσα προγραμματισμού Lisp. Στη Lisp, μία πράξη map δέχεται ως είσοδο μια λειτουργία και μια ακολουθία τιμών και ύστερα εφαρμόζει τη λειτουργία σε κάθε τιμή της ακολουθίας. Ακόμη, μία πράξη reduce συνδυάζει όλα τα στοιχεία μιας ακολουθίας, κάνοντας χρήση μιας δυαδικής λειτουργίας. Για παράδειγμα, μπορεί να χρησιμοποιήσει τον τελεστή “+” για να προσθέσει όλα τα στοιχεία μιας ακολουθίας.

Το MapReduce επιτυγχάνει αξιοπιστία, για το λόγο ότι διαμοιράζει λειτουργίες του συνόλου των δεδομένων σε κάθε ένα κόμβο ενός δικτύου. Κάθε κόμβος, υποβάλλει περιοδικά μια έκθεση με τα αποτελέσματα και τις ενημερώσεις της κατάστασής του. Όπως προβλέπεται, ένας κόμβος που έχει αποτύχει, παραμένει σιωπηλός και δεν υποβάλλει τίποτα. Ο υπεύθυνος εκείνου του κόμβου, σημειώνει τον εργαζόμενο (worker) του κόμβου που έχει αποτύχει και τον ξαναστέλνει για να δουλέψει. Αυτό το φαινόμενο απεικονίζεται στην Εικόνα 34. Εδώ έχουμε τους ρόλους του υπεύθυνου (master) και του εργαζομένου (worker), όπου συμβολίζονται με το ελεφαντάκι (το ελεφαντάκι αποτελεί την εικόνα της υλοποίησης Hadoop, που είναι η ανοιχτού κώδικα έκδοση του

MapReduce). Ο υπεύθυνος (master) διαιρεί το πρόβλημα σε εργασίες και τις αναθέτει στους εργαζόμενους (workers). Έπειτα, κάθε εργαζόμενος (worker) εκτελεί μια εργασία και υποβάλλει ένα αποτέλεσμα που λαμβάνει ο υπεύθυνος (master) και το χρησιμοποιεί στο τελικό αποτέλεσμα.



**Εικόνα 34 - Ρόλοι master και worker**

Ο υπεύθυνος (master) επιτελεί τις ακόλουθες εργασίες:

- Αρχικοποιεί τη σειρά και την χωρίζει σε εργασίες σύμφωνα με τον αριθμό των διαθέσιμων εργαζομένων (workers).
- Στέλνει σε κάθε εργαζόμενο (worker) την υπο-εργασία του.
- Λαμβάνει το αποτέλεσμα από κάθε εργαζόμενο (worker).

Ο εργαζόμενος (worker) επιτελεί τις ακόλουθες εργασίες:

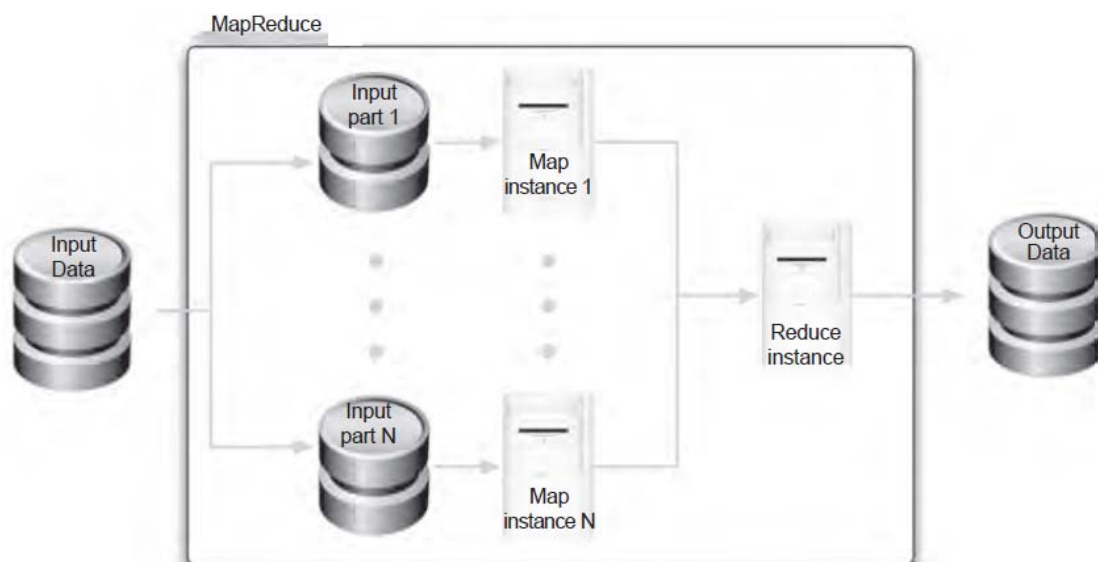
- Λαμβάνει την υπο-εργασία του από τον υπεύθυνο (master).
- Επεξεργάζεται την υπο-εργασία του.
- Επιστρέφει το αποτέλεσμα στον υπεύθυνο (master).

### 6.2.2.1 Το βήμα step

Η λειτουργία map, η οποία περιλαμβάνεται στη βιβλιοθήκη MapReduce, λαμβάνει ως είσοδο ένα ζευγάρι και παράγει ένα σύνολο ενδιάμεσων ζευγαριών κλειδιών. Η

βιβλιοθήκη MapReduce, ομαδοποιεί όλες τις ενδιάμεσες τιμές που σχετίζονται με το ίδιο ενδιάμεσο κλειδί I (όπου αυτό το κλειδί είναι αυθαίρετο και καθορισμένο από το χρήστη για την οργάνωση των ενδιάμεσων αποτελεσμάτων) και τις μεταφέρει στη λειτουργία reduce.

Ένας κόμβος υπευθύνου (master) δέχεται την είσοδο, την χωρίζει σε μικρότερα υπο-προβλήματα και κατανέμει τα υπο-προβλήματα αυτά στους κόμβους των workers. Οι κόμβοι των εργαζομένων (workers) μπορούν να επαναλάβουν τη διαδικασία για όσα επίπεδα απαιτούνται, μέχρις ότου να φθάσουν στο επιθυμητό μέγεθος του προβλήματος. Ένας κόμβος εργαζόμενου (worker) που λαμβάνει κατάλληλου μεγέθους προβλήματα, επεξεργάζεται το φόρτο εργασίας και υποβάλλει τα αποτελέσματα στον κόμβο-γονέα του. Η Εικόνα 35 παρουσιάζει τη λειτουργία map στα πλαίσια ολόκληρου του μοντέλου MapReduce. Τα δεδομένα εισόδου χωρίζονται σε υποσύνολα, όπου κάθε ένα από αυτά ανατίθεται από μια λειτουργία map. Έπειτα, μεταφέρει την έξοδο όλων των λειτουργιών map σε μια λειτουργία reduce, η οποία με τη σειρά της ετοιμάζει την τελική έξοδο.



**Εικόνα 35 - Λειτουργίες map και reduce σ' ένα μοντέλο MapReduce**

### 6.2.2.2 Το βήμα reduce

Η λειτουργία reduce, η οποία περιλαμβάνεται επίσης στη βιβλιοθήκη MapReduce, δέχεται ως είσοδο ένα ενδιάμεσο κλειδί I και ένα σύνολο τιμών για εκείνο το κλειδί. Επίσης, συγχωνεύει αυτές τις τιμές για να διαμορφώσει ένα ενδεχομένως μικρότερο σύνολο τιμών. Ο κόμβος υπευθύνου (master) λαμβάνει τις λύσεις/απαντήσεις σε όλα τα υπο-προβλήματα και τα συνδυάζει για να παράγει μια συνολική λύση, η οποία είναι και η τελικά ζητούμενη.

Ένα παράδειγμα της βιβλιοθήκης MapReduce αποτελεί το παρακάτω πρόγραμμα, όπου θεωρούμε το πρόβλημα της μέτρησης του αριθμού των εμφανίσεων κάθε λέξης σε μια μεγάλη συλλογή εγγράφων. Πιο συγκεκριμένα, έχουμε τον κώδικα για την επίλυση αυτού του προβλήματος:

```

void map(String name, String document):
    for each word w in document:
        EmitIntermediate(w, 1);
void reduce(String word, Iterator partialCounts):
    int result = 0;
    for each pc in partialCounts:
        result += ParseInt(pc);
    Emit(result);

```

Αναλυτικά, όσον αφορά την ερμηνεία του κώδικα:

- Για τη λειτουργία *map*, με τη μεταβλητή *name* συμβολίζουμε το όνομα του εγγράφου και με τη μεταβλητή *document* το περιεχόμενο του εγγράφου.
- Για τη λειτουργία *reduce*, με τη μεταβλητή *word* συμβολίζουμε μια λέξη και με τη μεταβλητή *partialCounts* μια λίστα με το άθροισμα κάποιων αριθμών.
- Όσον αφορά το στόχο της λειτουργίας *map*, αυτός είναι η εκπομπή κάθε λέξης με την προσθήκη ενός σχετικού αριθμού εμφανίσεων (στο παράδειγμα που βρισκόμαστε είναι 1).
- Όσον αφορά το στόχο της λειτουργίας *reduce*, σχετίζεται με την πρόσθεση όλων των αριθμών για μια συγκεκριμένη λέξη.

### 6.2.2.3 Λειτουργία του MapReduce

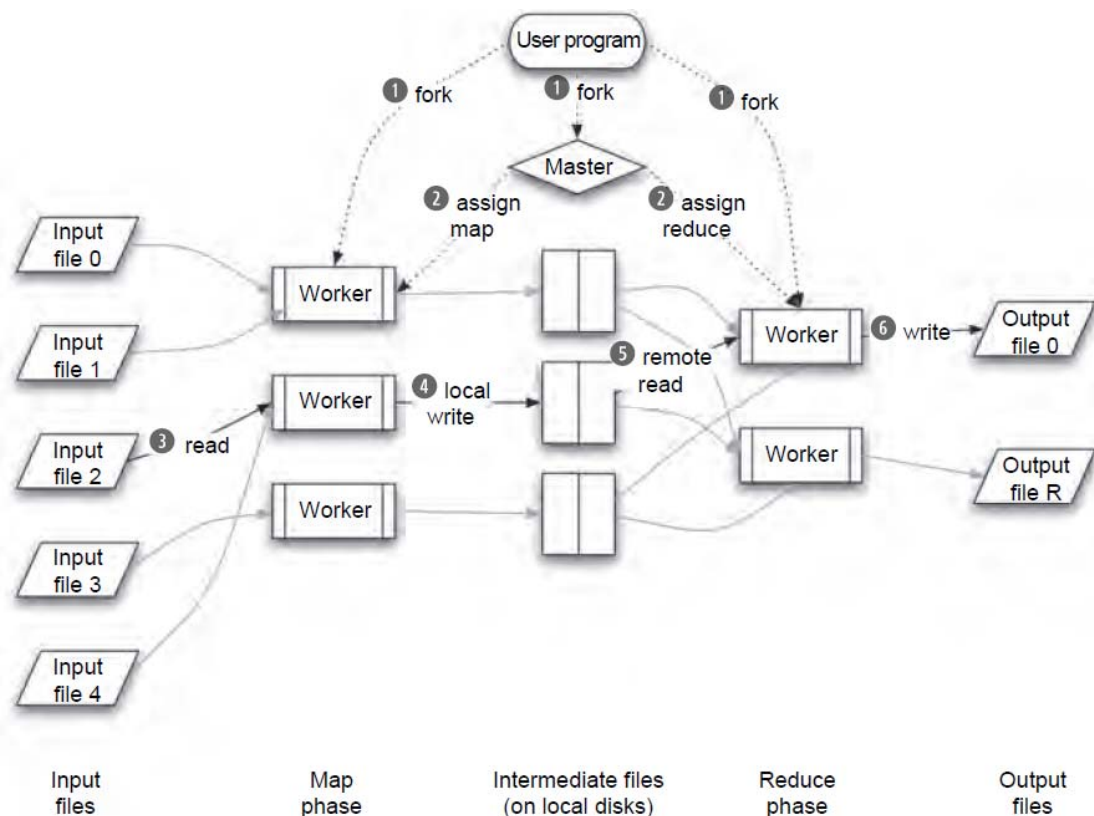
Οι εφαρμογές του MapReduce, αποτελούν περίπλοκα πλαίσια εργασίας για αξιόπιστη και παράλληλη επεξεργασία σε κατανεμημένα συστήματα. Ακόμη, επιτρέπουν κατανεμημένη επεξεργασία των λειτουργιών *map* και *reduce*. Αυτό συμβαίνει λόγω του ότι όλες οι λειτουργίες *map* είναι ανεξάρτητες μεταξύ τους, και αυτό έχει ως αποτέλεσμα να μπορούν να εκτελεστούν παράλληλα. Επιπλέον, με την έννοια του κλειδιού ορίζονται οι οποιοσδήποτε εξαρτήσεις στο σύνολο των δεδομένων.

Το MapReduce αρκετές φορές μπορεί να κριθεί ως ανεπαρκές σε σχέση με πιο βελτιωμένους ακολουθιακούς αλγορίθμους. Όμως, διαθέτει το πλεονέκτημα του ότι μπορεί να εφαρμοστεί σε τεράστια σύνολα δεδομένων, τα οποία είναι πολύ μεγαλύτερα από αυτά που μπορεί να χειριστεί ένας οποιοσδήποτε εξυπηρετής. Για παράδειγμα, μια μεγάλη συστοιχία εξυπηρετών μπορεί να χρησιμοποιήσει το MapReduce για τη ταξινόμηση ενός petabyte (1 εκατομμύριο GB δηλαδή) δεδομένων σε μερικές μόνο ώρες. Πιο σημαντικό είναι το γεγονός ότι, αυτός ο ογκώδης παραλληλισμός επιτρέπει την ομαλή ανάκαμψη από αποτυχίες των εξυπηρετών ή των συσκευών αποθήκευσης, κατά τη διάρκεια της επεξεργασίας. Αυτό συμβαίνει μιας και είναι εφικτός ο επαναπρογραμματισμός των λειτουργιών *map* και *reduce*, εφ' όσον τα δεδομένα είναι ακόμα διαθέσιμα.

Στην Εικόνα 36 παρουσιάζεται μια λεπτομερής εξέταση της λειτουργίας του MapReduce. Ειδικότερα, η βιβλιοθήκη MapReduce πρώτα διαχωρίζει τα αρχεία εισόδου σε *M* κομμάτια, όπου το κάθε κομμάτι είναι της τάξης των 16 MB ως 64 MB. Κατόπιν,

δημιουργεί πολλά αντίγραφα του προγράμματος και τα αναθέτει σε μια συστάδα μηχανών. Ακολούθως, λειτουργεί με τον ακόλουθο τρόπο:

1. Ένα από τα αντίγραφα είναι σημαντικότερο από τα άλλα: ο υπεύθυνος (master). Τα υπόλοιπα είναι εργαζόμενοι (workers) που τους αναθέτει εργασίες ο υπεύθυνος (master). Ο υπεύθυνος (master) επιλέγει τους εργαζομένους (workers) που δεν απασχολούνται με κάποια εργασία και αναθέτει σε κάθε έναν είτε μια εργασία *map* ή μια εργασία *reduce*.
2. Ένας εργαζόμενος (worker) που του έχει ανατεθεί μια εργασία *map*, διαβάζει το περιεχόμενο της αντίστοιχης εισόδου. Αναλύει τα ζεύγη κλειδιών-δεδομένων εισόδου και μεταφέρει κάθε ζεύγος στη λειτουργία *map* που είναι καθορισμένη από το χρήστη. Τα ενδιαμέσα ζεύγη κλειδιών που παράγονται από τη λειτουργία *map* και δεν χρειάζονται στην έξοδο του προγράμματος, αποθηκεύονται στη μνήμη.
3. Όσον αφορά τα ενδιαμέσα ζεύγη, είναι αποθηκευμένα στον τοπικό σκληρό δίσκο, ο οποίος είναι χωρισμένος σε R περιοχές με χρήση της λειτουργίας διαχωρισμού (partitioning). Οι τοποθεσίες αυτών των αποθηκευμένων ζευγών στον τοπικό δίσκο μεταφέρονται στον υπεύθυνο (master) που είναι αρμόδιος για την αποστολή αυτών των τοποθεσιών στους εργαζομένους (workers) της λειτουργίας *reduce*.
4. Όταν ένας εργαζόμενος (worker) της λειτουργίας *reduce* ειδοποιείται από τον υπεύθυνο (master) για τις τοποθεσίες, τότε χρησιμοποιεί απομακρυσμένες κλήσεις για να διαβάσει τα αποθηκευμένα δεδομένα από τους τοπικούς δίσκους των εργαζομένων (workers) της λειτουργίας *map*. Εφόσον εν συνεχεία, ο εργαζόμενος (worker) της λειτουργίας *reduce* έχει διαβάσει όλα τα ενδιαμέσα δεδομένα, τα ταξινομεί με βάση τα ενδιαμέσα κλειδιά κατά τέτοιο τρόπο ώστε όλα τα στιγμιότυπα του ίδιου κλειδιού να είναι ομαδοποιημένα μαζί. Εάν η ποσότητα των ενδιαμέσων δεδομένων είναι πάρα πολύ μεγάλη για να χωρέσει στη μνήμη, τότε χρησιμοποιείται ένας εξωτερικός δίσκος.
5. Ο εργαζόμενος (worker) της λειτουργίας *reduce* διατρέχει επανειλημμένως τα ταξινομημένα ενδιαμέσα δεδομένα και για κάθε μοναδικό ενδιαμέσο κλειδί που ανακαλύπτει, μεταφέρει το κλειδί και το αντίστοιχο σύνολο από ενδιαμέσες τιμές, στη λειτουργία *reduce* του χρήστη. Η έξοδος της λειτουργίας *reduce* επισυνάπτεται σε ένα τελικό αρχείο εξόδου.
6. Όταν ολοκληρωθούν όλες οι εργασίες *map* και *reduce*, ο υπεύθυνος (master) επαναφέρει το πρόγραμμα του χρήστη. Η κλήση MapReduce επιστρέφεται και η έξοδος της βρίσκεται στα R αρχεία εξόδου.



**Εικόνα 36 - Λειτουργία MapReduce**

### 6.2.3 Hadoop: το ανοικτού κώδικα MapReduce

Όπως αναφέρθηκε και προηγουμένως, το MapReduce αναπτύχθηκε από τη Google, δεν παρέμεινε ωστόσο μόνο στα όρια της Google, αλλά τα ξεπέρασε με τη δημιουργία του Hadoop. Αναλυτικότερα, το Hadoop αναπτύχθηκε ως ένα έργο ανοικτού κώδικα του οργανισμού Apache, με στόχο να παρέχει τη λειτουργικότητα του MapReduce. Το Hadoop χρησιμοποιείται ευρέως, από πολλές σημαντικές ομάδες, μερικές εκ των οποίων, απαριθμούνται παρακάτω:

1. Amazon, A9 αναζήτηση προϊόντων, καθώς και Elastic MapReduce ως τμήμα των υπηρεσιών Ιστού του Amazon.
2. Adobe, κοινωνικές υπηρεσίες.
3. Baidu, μηχανή αναζήτησης κινέζικης γλώσσας.
4. Facebook, συστάδα 4.800 υπολογιστών που χρησιμοποιείται για την αποθήκευση αντιγράφων από πηγές δεδομένων.
5. Hulu, αποθήκευση και ανάλυση logs.
6. IBM, μπλε νέφος (Blue Cloud) που βασίζεται στο Hadoop.
7. NetSeer, 1.000 EC2 στιγμιότυπα που εκτελούν το Hadoop για την επεξεργασία και ανάλυση εξυπηρετητών και logs.

8. New York Times, μετατροπές εικόνας μεγάλης κλίμακας που εκτελούνται στο EC2.
9. Rackspace, συστάδα 30 κόμβων που αναλύει και ταξινομεί logs από το σύστημα ηλεκτρονικού ταχυδρομείου.
10. Yahoo!, περισσότεροι από 100.000 υπολογιστές σε περισσότερους από 25.000 εξυπηρετητές που εκτελούν το Hadoop.

### 6.3 SLAs και δημόσιοι πάροχοι νεφών

Στη φάση αυτή, θα γίνει αναφορά στα καλούμενα συμφωνητικά παροχής υπηρεσιών (SLAs- Service Level Agreement) τα οποία αποτελούν ένα σημαντικό τρόπο για τους πάροχους νεφών, έτσι ώστε να είναι σε θέση να μπορούν να αντιμετωπίσουν τις όποιες δυσκολίες, όπως στη συνέχεια θα αναφερθούν. Η μεταφορά των εφαρμογών στο νέφος σίγουρα δεν αποτελεί μια τόσο απλή υπόθεση, λόγω της ανησυχίας των ανθρώπων στην εκχώρηση του ελέγχου των αρμοδιοτήτων και της διαχείρισης με τις οποίες είναι εξοικειωμένοι.

Ειδικότερα, ένα συμφωνητικό παροχής υπηρεσιών (SLA) ορίζει ειδικά κριτήρια ή επίπεδα υπηρεσιών, τα οποία δεσμεύεται να παρέχει ένας πάροχος σε μια επιχείρηση μέσω μιας συμβατικής υποχρέωσης. Επίσης, ένα συμφωνητικό παροχής υπηρεσιών (SLA), κατοχυρώνει την ποιότητα των υπηρεσιών που παρέχονται στις εκάστοτε επιχείρησης σε ένα υψηλό επίπεδο.

Επιπροσθέτως, τα συμφωνητικά παροχής υπηρεσιών SLAs μπορούν να αναφερθούν γενικότερα και ως *ποσοτικά προσδιορισίμες μετρικές*, οι οποίες περιγράφουν την ικανοποιητική λειτουργία μιας υπηρεσίας. Οι μετρικές αυτές, πρέπει να διατηρούνται σε μια συνεχή βάση. Ωστόσο, μια τυχόν αδυναμία να διατηρηθούν, αποτελεί τη λεγόμενη και *παραβίαση επιπέδου ποιότητας υπηρεσίας* (SLA violation). Επιπλέον, ένα συμφωνητικό παροχής υπηρεσιών SLA αποτελεί ένα τρόπο προσφοράς των υπηρεσιών τους για τους παρόχους νέφους. Ο λόγος για τον οποίο συμβαίνει αυτό, είναι διότι μπορούν να αξιώσουν ό,τι χρηματικό ποσό θελήσουν ανάλογα με την ποιότητα της προσφοράς τους. Βέβαια, απ' την άλλη πλευρά, ο εξυπηρετούμενος διαθέτει το δικαίωμα αποζημίωσης συνήθως υπό μορφή πίστωσης, σε περίπτωση που ο πάροχος δεν είναι σε θέση να εκπληρώσει τους στόχους SLA που είχε εκ των προτέρων υποσχεθεί.

Επίσης, τα συμφωνητικά παροχής υπηρεσιών προσφέρονται συνήθως από παρόχους υπηρεσιών για τις προσφορές των κέντρων δεδομένων τους. Ακόμη, τα συμφωνητικά παροχής υπηρεσιών νέφους είναι ομοίως δομημένα με εκείνα που προσφέρονται για παραδοσιακές υπηρεσίες φιλοξενίας (hosting), αλλά μπορεί να διαφέρουν σε συγκεκριμένες μετρικές που είναι ενσωματωμένες στο συμφωνητικό παροχής υπηρεσιών SLA. Στη συνέχεια του κεφαλαίου αυτού, θα γίνει μια αναφορά στα συμφωνητικά παροχής υπηρεσιών SLAs τριών σημαντικών παρόχων νέφους: Amazon EC2, Microsoft Azure και Rackspace Cloud.

#### 6.3.1 SLA του Amazon AWS

Το Amazon AWS προσφέρει ένα συμφωνητικό παροχής υπηρεσιών SLA για τις υπηρεσίες υπολογισμού EC2, καθώς και υπηρεσίες αποθηκευτικού χώρου S3. Το



επίπεδο της υπηρεσίας EC2 υπολογίζεται σε ετήσια βάση, ενώ το επίπεδο της υπηρεσίας S3 σε μηνιαία διαστήματα. Το συμφωνητικό παροχής υπηρεσιών SLA της EC2 υπηρεσίας εγγυάται δύο παραμέτρους:

- Τη δυνατότητα να προσπελαστούν στιγμιότυπα και
- Τη δυνατότητα να ξεκινήσουν νέα στιγμιότυπα.

Το Amazon εγγυάται πρόσβαση σε στιγμιότυπα που βρίσκονται στο δίκτυό του και για παράγοντες που βρίσκονται υπό τον έλεγχό του, εξαιρεί δηλαδή αποτυχίες στιγμιότυπων, οι οποίες δεν οφείλονται συγκεκριμένα στη μη διαθεσιμότητα στοιχείων που δεν βρίσκονται υπό τον έλεγχό του. Σε περίπτωση παραβίασης του SLA, τότε ο πελάτης του EC2 έχει δικαίωμα σε μια επιστροφή πίστωσης 10% για τη μηνιαία χρέωση υπηρεσιών του συγκεκριμένου μήνα, αποκλείοντας όμως οποιεσδήποτε αμοιβές που χρεώθηκαν για πρώτη φορά κατά τη διάρκεια αυτού του μήνα.

Σε σχέση με τα επίπεδα μηνιαίων υπηρεσιών του Amazon S3, αυτά καθορίζονται βάσει μιας ποσότητας που καθορίζεται ως ποσοστό λάθους (Error Rate), το οποίο ισοδυναμεί με τον αριθμό των αιτήσεων υπηρεσιών που οδηγούν σε εσωτερικό λάθος (Internal Error) ή σφάλμα διαθεσιμότητας υπηρεσίας (Service Unavailable) διά του συνολικού αριθμού αιτήσεων, που πραγματοποιούνται σ' ένα διάστημα πέντε λεπτών. Το μηνιαίο ποσοστό νέων υπηρεσιών (Monthly Uptime Percentage) καθορίζεται ως το 100%, μείον το μέσο ποσοστό σφαλμάτων όλων των διαστημάτων πέντε λεπτών του μήνα. Εν συνεχεία, ο εξυπηρετούμενος λαμβάνει μια πίστωση υπηρεσιών 10% του μηνιαίου λογαριασμού, εάν το μηνιαίο ποσοστό νέων υπηρεσιών είναι μικρότερο από 99,9% και μια πίστωση 25% εάν είναι μικρότερο κι από 99%.

Τέλος, για οτιδήποτε σχετίζεται με την απόκτηση πίστωσης για παραβιάσεις επιπέδου υπηρεσιών είτε για το EC2 είτε για το S3, ένας εξυπηρετούμενος υποχρεούται να αποστείλει μια ειδοποίηση στο Amazon μέσα σε 30 ημέρες, από τη στιγμή εκείνη της παραβίασης και να παράσχει μια λεπτομερή περίληψη της περίπτωσης του.

### 6.3.2 SLA του Microsoft Azure

Τα συμφωνητικά παροχής υπηρεσιών (SLAs) του Microsoft Azure χρησιμοποιούνται για τις υπηρεσίες Azure Compute, Azure Storage, AppFabric Access Control και SQL Server. Το συμφωνητικό παροχής υπηρεσιών (SLA) του Azure Compute, είναι παρόμοιο με αυτό του Amazon για το EC2. Ειδικότερα, όπως και στο EC2, τίθεται σε επίπεδο 99,95% και περιέχει δύο στοιχεία:

- Τη διαχείριση της συνδεσιμότητας των ρόλων του διαδικτύου και
- Τη διαχείριση του χρόνου λειτουργίας (uptime) όλων των στιγμιότυπων των ρόλων των πελατών.

Ο χρόνος διακοπής της συνδεσιμότητας (connectivity downtime) ισούται με το σύνολο των διαστημάτων πέντε λεπτών για τα οποία δύο ή περισσότεροι ρόλοι του διαδικτύου σε διαφορετικές περιοχές δεν είναι διαθέσιμοι. Συνεπώς, το μηνιαίο ποσοστό uptime συνδεσιμότητας (Monthly Connectivity Uptime Percentage) που καθορίζει SLA Compute υπολογίζεται ως:

Μηνιαίο ποσοστό uptime συνδεσιμότητας = (Μέγιστος χρόνος συνδεσιμότητας – Διακοπή συνδεσιμότητας) / Μέγιστος χρόνος συνδεσιμότητας

Το συμφωνητικό παροχής υπηρεσιών (SLA) του Azure Storage, μοιάζει αρκετά με το συμφωνητικό παροχής υπηρεσιών (SLA) του Amazon S3. Αυτό συμβαίνει για το λόγο ότι καθορίζει ένα ποσοστό λάθους μ' ένα τρόπο παρόμοιο με του S3, έχοντας την εξής επουσιώδη διαφορά: το συνολικό διάστημα για μέτρηση είναι εξ ορισμού 1 ώρα για το Azure, έναντι του διαστήματος πέντε λεπτών του S3. Επίσης, και τα δύο νέφη καθορίζουν τα συμφωνητικά παροχής υπηρεσιών (SLAs) σε μηνιαία βάση και έχουν πιστώσεις υπηρεσιών 10% και 25%, όπου και οφείλονται σε αστοχίες του συμφωνητικού παροχής υπηρεσιών (SLA) στα επίπεδα 99,9% και 99%, αντίστοιχα.

Τα συμφωνητικά παροχής υπηρεσιών (SLAs) του Azure AppFabric Access Control και του SQL Server αντιθέτως, δεν έχουν καμία ομοιότητα με κάποια αντίστοιχα του Amazon. Τόσο τα SLA του Azure AppFabric Access Control όσο και του SQL Server καθορίζονται σε μηνιαία βάση, με τις πιστώσεις υπηρεσιών 10% και 25% να αντιστοιχίζονται στα επίπεδα 99,9% και 99%. Όσον αφορά το AppFabric, τα συμφωνητικά παροχής υπηρεσιών (SLAs) βασίζονται στα ποσοστά συνδεσιμότητας και uptime επεξεργασίας των υπηρεσιών ελέγχου προσπέλασης (Access Control Services), ενώ, όσον αφορά τον SQL Server, τα συμφωνητικά παροχής υπηρεσιών (SLAs) βασίζονται στη συνδεσιμότητα, αλλά και βασικές διαδικασίες ανάγνωσης κι εγγραφής (read, write). Επιπλέον, αυτό το συμφωνητικό παροχής υπηρεσιών (SLA) επιτρέπει αποκλεισμό για προγραμματισμένη συντήρηση μέχρι και 10 ώρες το χρόνο, που όμως δεν παραβιάζει το συμφωνητικό παροχής υπηρεσιών (SLA). Για να απαιτήσει ένας πελάτης πίστωση για παραβίαση του Azure SLA, πρέπει να επικοινωνήσει με τη Microsoft εντός πέντε ημερών, από τη στιγμή της παραβίασης. Ύστερα η επικοινωνία αυτή, αναμφίβολα πρέπει να επαναληφθεί πριν το τέλος του επόμενου μηνιαίου κύκλου.

### 6.3.3 SLA του νέφους Rackspace

Τα τρία συμφωνητικά παροχής υπηρεσιών (SLAs) του Rackspace, ονομάζονται Cloud, Sites και Files. Τα συμφωνητικά αυτά δομούνται διαφορετικά από τα συμφωνητικά παροχής υπηρεσιών (SLAs) του Amazon και της Microsoft. Επίσης, σε αντίθεση με τα άλλα συμφωνητικά παροχής υπηρεσιών, το συμφωνητικό παροχής υπηρεσιών του Rackspace μπορεί να εξουσιοδοτήσει τους εξυπηρετούμενους να λάβουν πιστώσεις μέχρι το ποσό ολόκληρης της μηνιαίας αμοιβής. Αυτό βέβαια θα συμβεί, όπως και στις άλλες δυο περιπτώσεις, με την ειδοποίηση των πελατών στο Rackspace για την οποιαδήποτε παραβίαση του συμφωνητικού παροχής υπηρεσιών.

Το συμφωνητικό παροχής υπηρεσιών του νέφους Rackspace διαχειρίζεται τη λειτουργία της υποδομής του νέφους Rackspace, συμπεριλαμβανομένης της τεχνολογίας HVAC (heating, ventilation and air conditioning) στο κέντρο δεδομένων, στη συνδεσιμότητα δικτύου και στον εικονικό εξυπηρέτη, αλλά και συμπεριλαμβανομένων του υπολογισμού και της αποθήκευσης. Όσον αφορά την περίπτωση της ισχύς του κέντρου δεδομένων και της συνδεσιμότητας του δικτύου, το Rackspace εγγυάται 100% uptime και προσφέρει 5% πίστωση για κάθε ώρα μη διαθεσιμότητας, μέχρι και το 100% του συνολικού μηνιαίου λογαριασμού. Σε περίπτωση αποτυχίας ενός εικονικού εξυπηρέτη, το Cloud SLA εγγυάται μια αντικατάσταση σε διάστημα έως μίας ώρας, αφότου ανακαλυφθεί το

πρόβλημα. Στη συνέχεια, για κάθε επιπρόσθετη ώρα, μετά από την αρχική αντικατάσταση, παρέχεται μια 5% πίστωση στον πελάτη έως ότου επιλυθεί η κατάσταση.

Τέλος, το SLA Sites διαχειρίζεται τη διαθεσιμότητα των ιστοσελίδων, το ηλεκτρονικό ταχυδρομείο, καθώς και τις βάσεις δεδομένων. Παρέχει επίσης, μια πίστωση υπηρεσιών μιας ημέρας για κάθε μια ώρα χρόνου διακοπής. Επιπλέον, αυτό το συμφωνητικό παροχής υπηρεσιών διαχειρίζεται τη διαθεσιμότητα των υπηρεσιών επικύρωσης, αποθήκευσης, καθώς και του δικτύου παράδοσης περιεχομένου (CDN, content delivery network). Η ελάχιστη αποδεκτή διαθεσιμότητα για SLA Sites καθορίζεται στο 99,9%. Για επίπεδα διαθεσιμότητας μικρότερα απ' αυτό το επίπεδο, η πίστωση υπολογίζεται σε μια κλίμακα που φτάνει μέχρι το 96,5%.

## Κεφάλαιο 7: Ασφάλεια και ιδιωτικότητα στο υπολογιστικό νέφος

### 7.1 Διαχείριση ασφάλειας στο νέφος

Κατά την υιοθέτηση των υπηρεσιών δημόσιου νέφους, ένα μεγάλο μέρος του δικτύου μας, όπως το σύστημα, οι εφαρμογές και τα δεδομένα θα τεθούν υπό τον έλεγχο τρίτων. Επίσης, το μοντέλο παροχής υπηρεσιών νέφους θα δημιουργήσει νέφη εικονικών παραμέτρων, καθώς και ένα μοντέλο ασφάλειας με κοινές ευθύνες μεταξύ του εξυπηρετούμενου και του παρόχου υπηρεσιών νέφους (CSP – cloud service provider). Αυτό το διαμοιρασμένο μοντέλο ευθύνης θα φέρει νέες προκλήσεις διαχείρισης ασφάλειας στο προσωπικό πληροφορικής.

Έχοντας αυτό κατά νου, το πρώτο ερώτημα που πρέπει να απαντηθεί από τον ανώτερο υπάλληλο ασφάλειας πληροφοριών (CISO – chief information security officer), είναι εάν οι υπηρεσίες νέφους διαθέτουν επαρκή διαφάνεια, προκειμένου να διαχειριστεί και να εφαρμόσει τις διαδικασίες διαχείρισης ασφάλειας (προληπτικοί έλεγχοι), ώστε να βεβαιώσει την επιχείρηση ότι τα δεδομένα της στο νέφος είναι επαρκώς ασφαλή. Για να είναι δυνατό να απαντηθεί το ερώτημα αυτό, πρέπει να εξετασθούν τα ακόλουθα:

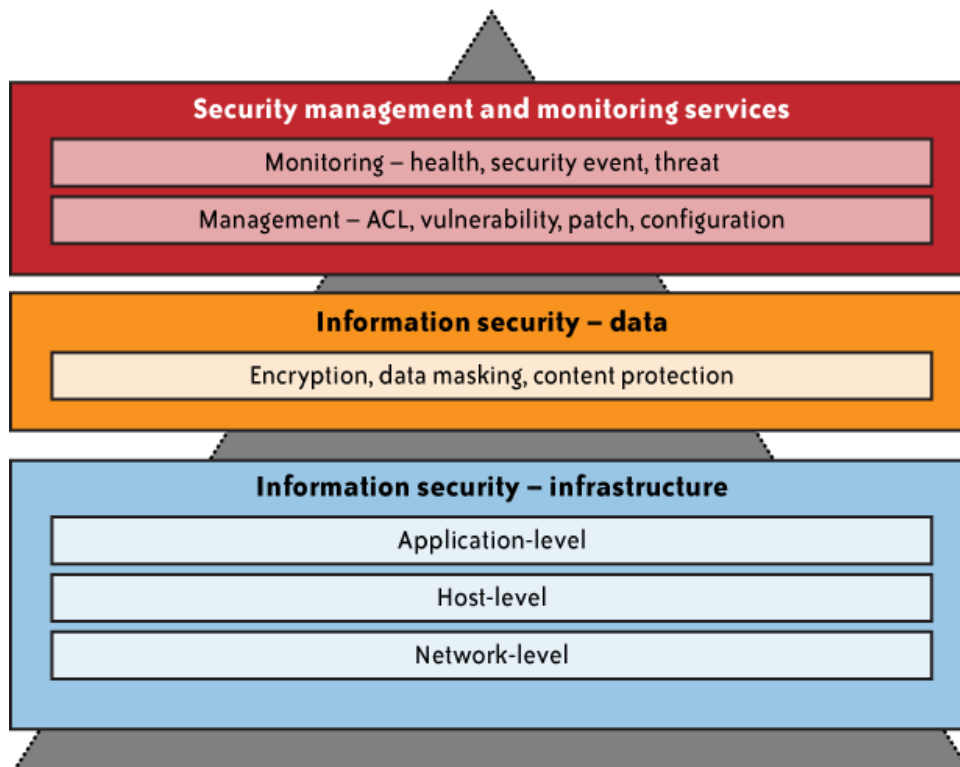
1. ποιούς ελέγχους ασφάλειας πρέπει να διαθέτει επιπλέον ο εξυπηρετούμενος πέραν των ενσωματωμένων ελέγχων που διαθέτει η πλατφόρμα νέφους,
2. με ποιόν τρόπο θα προσαρμοστούν τα εργαλεία και οι διαδικασίες διαχείρισης ασφαλείας μιας επιχείρησης προκειμένου να διαχειριστούν την ασφάλεια στο νέφος.

Και τα δύο ζητήματα πρέπει συνεχώς να επαναξιολογούνται βάση της ευαισθησίας των δεδομένων και των αλλαγών του επιπέδου υπηρεσιών κατά τη διάρκεια του χρόνου.

Εν συνεχεία, ο εξυπηρετούμενος, πρέπει να εξασκηθεί στην κατανόηση των ορίων εμπιστοσύνης των υπηρεσιών του στο νέφος. Θα πρέπει λοιπόν να κατανοήσει όλα τα επίπεδα που υπάρχουν στην δικτυακή υπηρεσία νέφους όπως:

- Την εφαρμογή,
- Τη βάση δεδομένων,
- Την αποθήκευση, καθώς και
- Τις διαδικτυακές υπηρεσίες συμπεριλαμβανομένων των υπηρεσιών ταυτότητας (Εικόνα 37).

Πρέπει επίσης να κατανοήσει το σκοπό και την έκταση της διαχείρισης του πληροφοριακού συστήματος και των ευθυνών ελέγχου που βαρύνουν τον ίδιο, συμπεριλαμβανομένων της πρόσβασης, της αλλαγής, της διαμόρφωσης και της διαχείρισης ευπάθειας.



**Εικόνα 37 - Υπηρεσίες ταυτότητας**

Το επίπεδο των λειτουργικών ευθυνών, που μπορούν να μεταβιβαστούν από τον χρήστη στον προμηθευτή, μπορεί να ποικίλει και θα εξαρτηθεί από διάφορους παράγοντες συμπεριλαμβανομένων:

- Της διεπαφής παρόχου υπηρεσίας (SPI – service provider interface),
- Της συμφωνίας για το επίπεδο υπηρεσιών που παρέχει ο προμηθευτής (SLA), καθώς και
- Τις συγκεκριμένες ικανότητες του προμηθευτή για την υποστήριξη της επέκτασης των εσωτερικών διαδικασιών και εργαλείων διαχείρισης ασφαλείας.

Οι πιο ώριμοι οργανισμοί τεχνολογίας πληροφορικής χρησιμοποιούν πλαίσια διαχείρισης ασφάλειας, όπως το ISO/IEC 27000 και το πλαίσιο διαχείρισης υπηρεσιών: Information Technology Infrastructure Library (ITIL). Αυτά τα τυποποιημένα πλαίσια διαχείρισης παρέχουν τις οδηγίες για τον προγραμματισμό και την εκτέλεση ενός προγράμματος διακυβέρνησης με την υποστήριξη των διαδικασιών διαχείρισης που προστατεύουν τις πληροφορίες.

Τα πλαίσια διαχείρισης όπως το ITIL, το οποίο θα αναλυθεί παρακάτω, θα βοηθήσουν με τη συνεχή βελτίωση των υπηρεσιών που είναι απαραίτητη ώστε να ευθυγραμμίσει εκ νέου τις υπηρεσίες τεχνολογιών πληροφορίας IT στις μεταβαλλόμενες επιχειρησιακές ανάγκες. Συνεχής βελτίωση υπηρεσιών συνεπάγεται και βελτιώσεις στις υπηρεσίες τεχνολογιών πληροφορικής που υποστηρίζουν τις επιχειρησιακές διαδικασίες, όπως η αυτοματοποίηση των πωλήσεων χρησιμοποιώντας έναν πάροχο υπηρεσιών νέφους.

Λαμβάνοντας υπόψη τα δυναμικά χαρακτηριστικά των υπηρεσιών υπολογιστικού νέφους, οι δραστηριότητες που λαμβάνουν μέρος μέσα στα πλαίσια της διαδικασίας διαχείρισης ασφαλείας πρέπει να αναθεωρούνται συνεχώς για να παραμείνουν ενημερωμένες και αποτελεσματικές.

### 7.1.1 Πρότυπα διαχείρισης Ασφαλείας

Τα πρότυπα τα οποία είναι σχετικά με τη διαχείριση ασφάλειας στο νέφος είναι το ITIL και το ISO/IEC 27001 και 27002.

#### 7.1.1.1 ITIL

Η βιβλιοθήκη υποδομής τεχνολογίας πληροφοριών (Information Technology Infrastructure Library -ITIL) είναι ένα σύνολο βέλτιστων πρακτικών και οδηγιών που καθορίζουν μια ολοκληρωμένη διαδικασία, βασισμένη στην προσέγγιση για τη διαχείριση υπηρεσιών πληροφοριακής τεχνολογίας. Το ITIL μπορεί να εφαρμοστεί σχεδόν κάθε τύπο περιβάλλοντος τεχνολογιών πληροφορίας IT συμπεριλαμβανομένου του λειτουργικού περιβάλλοντος νεφών. Το ITIL επιδιώκει να εξασφαλίσει ότι αποτελεσματικά μέτρα ασφαλείας πληροφοριών πρέπει να λαμβάνονται σε στρατηγικό, τακτικό, και λειτουργικό επίπεδο.

Ο στόχος του πλαισίου διαχείρισης ασφάλειας ITIL διαιρείται σε δύο μέρη:

- Πραγματοποίηση των απαιτήσεων ασφάλειας: Οι απαιτήσεις ασφάλειας καθορίζονται συνήθως στο συμφωνητικό παροχής υπηρεσιών (SLA), οι οποίες διευκρινίζονται στις συμβάσεις, στη νομοθεσία και στις εσωτερικά ή εξωτερικά επιβληθείσες πολιτικές.
- Πραγματοποίηση ενός βασικού επιπέδου ασφάλειας: Είναι απαραίτητο για να εγγυηθεί την ασφάλεια και τη συνοχή της οργάνωσης και προκειμένου να επιτευχθεί μια απλουστευμένη υπηρεσία για τη διαχείριση ασφάλειας των πληροφοριών.

Οι καθιερωμένες διαδικασίες διαχείρισης ασφάλειας ευθυγραμμίζονται επίσης με τις πολιτικές και τα πρότυπα των τεχνολογιών πληροφορικής ενός οργανισμού, που ως στόχο έχουν:

- Την εμπιστευτικότητα,
- Την ακεραιότητα, και
- Τη διαθεσιμότητα των πληροφοριών.

Στην Εικόνα 38 απεικονίζεται ο κύκλος ζωής σε μια επιχείρηση βάσει της βιβλιοθήκης ITIL. Οι διάφορες ειδικότητες διαχείρισης ασφάλειας, αντιπροσωπεύονται από τις σχετικές λειτουργίες ISO και ITIL.

Η ασφάλεια πληροφοριών θεωρείται μια επαναληπτική διαδικασία που πρέπει:

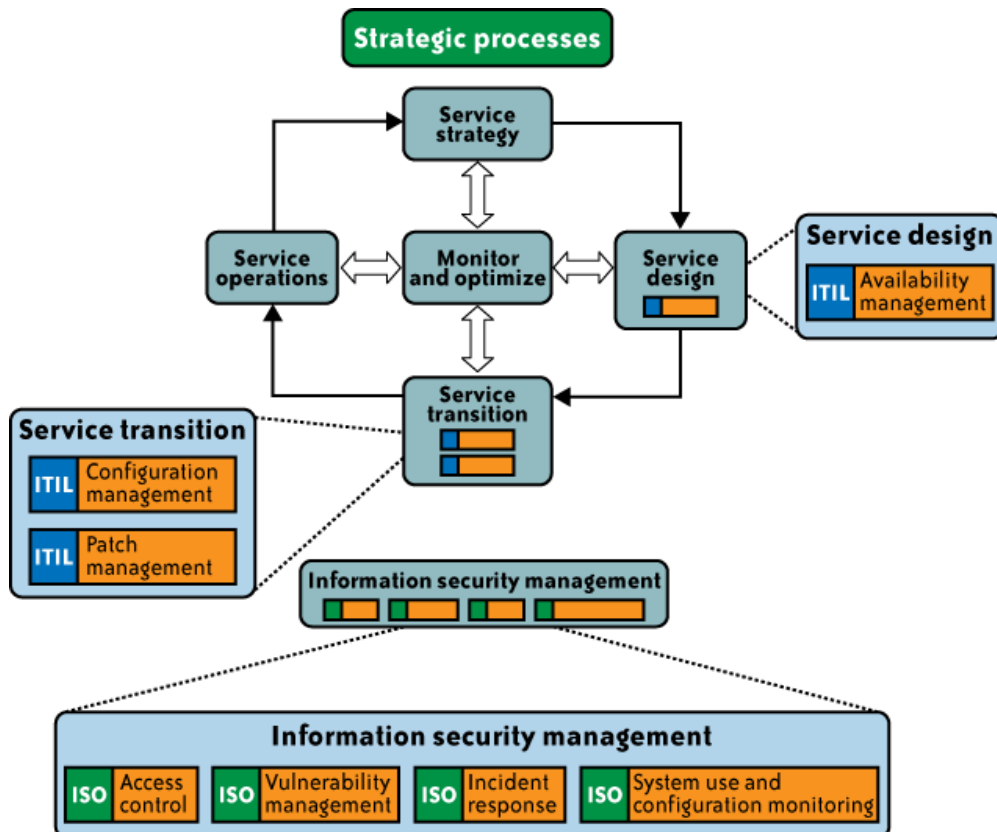
- Να ελεγχθεί,
- Να προγραμματιστεί,
- Να εφαρμοστεί,

- Να αξιολογηθεί και
- Να διατηρηθεί.

Το ITIL διαχωρίζει την ασφάλεια πληροφοριών σε:

- Πολιτικές (Policies): Οι γενικοί στόχοι που μια οργάνωση προσπαθεί να επιτύχει.
- Διεργασίες (Processes): Αυτό που πρέπει να συμβεί ώστε να επιτευχθούν οι στόχοι.
- Διαδικασίες (Procedures): Τι και πότε κάνει κάτι κάποιος για να επιτευχθούν οι στόχοι.
- Οδηγίες εργασίας (Work instructions): Οδηγίες για τη λήψη συγκεκριμένων μέτρων.

Η διεργασία διαχείρισης ασφάλειας ITIL είναι βασισμένη στον κώδικα συμπεριφοράς για τη διαχείριση ασφάλειας πληροφοριών που είναι γνωστή ως ISO / IEC 1779 9:2005.



Εικόνα 38 - Κύκλος ζωής ITIL σε μια επιχείρηση

### 7.1.1.2 ISO 27001/27002

Το ISO/IEC 27001 τυπικά καθορίζει τις υποχρεωτικές απαιτήσεις για ένα σύστημα διαχείρισης ασφάλειας πληροφοριών (Information Security Management System – ISMS). Είναι επίσης ένα πρότυπο πιστοποίησης και χρησιμοποιεί το ISO/IEC 27002 για

την παροχή των κατάλληλων ελέγχων ασφάλειας πληροφοριών μέσα στο ISMS. Εντούτοις, από τη στιγμή που το ISO/IEC 27002 είναι περισσότερο ένας κώδικας συμπεριφοράς/οδηγίας παρά πρότυπο πιστοποίησης, οι οργανισμοί είναι ελεύθεροι να επιλέξουν και να εφαρμόσουν τους ελέγχους που τους ταιριάζουν.

Λαμβάνοντας υπόψη τη σύγχρονη τάση των οργανισμών που κινούνται προς το ISO/IEC 27001 για τη διαχείριση ασφάλειας των πληροφοριών, υπάρχει μια γενική συναίνεση μεταξύ των επαγγελματιών ασφάλειας πληροφοριών για την αναθεώρηση των καλύτερων πρακτικών διαχείρισης ασφάλειας ITIL με το στόχο την ενδυνάμωση της εφαρμογής και λογικής ασφάλειας στην υποδομή τεχνολογίας πληροφοριών και επικοινωνιών (ICT).

### **7.1.2 Διαχείριση ασφαλείας στο νέφος**

Μετά από την ανάλυση των διεργασιών διαχείρισης στα πλαίσια του ITIL και του ISO, προσδιορίζονται οι ακόλουθες σχετικές διαδικασίες ως οι προτεινόμενες περιοχές διαχείρισης ασφαλείας για την εξασφάλιση των υπηρεσιών στο νέφος:

- Διαχείριση διαθεσιμότητας (Availability management – ITIL)
- Έλεγχος πρόσβασης (Access control – ISO/IEC 27002, ITIL)
- Διαχείριση ευπάθειας (Vulnerability management – ISO/IEC 27002)
- Διαχείριση επιδιορθώσεων (Patch management – ITIL)
- Διαχείριση διαμόρφωσης (Configuration management – ITIL)
- Αντίδραση σε περιστατικά ασφάλειας (Incident response – ISO/IEC 27002)
- Χρήση συστήματος και παρακολούθηση προσβάσεων (System use and access monitoring – ISO/IEC 27002)

Οι παραπάνω διαδικασίες διαχείρισης ασφάλειας επιλέχθηκαν κυρίως βάση του αντίκτυπου που θα έχουν στην ελαχιστοποίηση του γενικού κινδύνου για τον οργανισμό.

## **7.2 Διαχείριση ασφαλείας ευπάθειας, επιδιορθώσεων και ρύθμισης παραμέτρων**

Μια σημαντική απειλή στις υπηρεσίες νέφους αποτελεί η δυνατότητα του malware (malicious software) ή ενός cracker να εκμεταλλεύεται με απομακρυσμένη πρόσβαση την ευπάθεια των στοιχείων υποδομής, των υπηρεσιών δικτύων και των εφαρμογών. Η απειλές αυτές συνιστούν ιδιαίτερα μεγάλο κίνδυνο για ένα δημόσιο μοντέλο παράδοσης ΠωΥ και ΥωΥ, όπου η διαχείριση της ευπάθειας, των επιδιορθώσεων και της ρύθμισης παραμέτρων παραμένουν στον χρήστη. Όσον αφορά τους εξυπηρετούμενους, πρέπει να έχουν κατά νου ότι στα περιβάλλοντα υπολογιστικού νέφους, ο χαμηλότερος ή υψηλότερος κοινός παρονομαστής ασφάλειας μοιράζεται από όλους τους μισθωτές σε ένα εικονικό περιβάλλον πολλαπλής μίσθωσης. Ως εκ τούτου, το βάρος πέφτει επάνω τους με σκοπό να γίνει ιδιαίτερα κατανοητή η έκταση των ευθυνών τους προς τη διαχείριση της ασφάλειας. Γενικότερα, οι χρήστες πρέπει να απαιτήσουν να γίνουν τα CSPs διαφανέστερα για την ασφάλεια των λειτουργιών του νέφους τους, έτσι ώστε να



βοηθηθούν οι ίδιοι να κατανοήσουν αλλά και να προγραμματίσουν συμπληρωματικές λειτουργίες διαχείρισης ασφάλειας.

Ωστόσο, σε γενικές γραμμές, οι πάροχοι υπηρεσιών είναι αρμόδιοι για τα ακόλουθα:

- τη διαχείριση της ευπάθειας,
- την εγκατάσταση επιδιορθώσεων και τη διαμόρφωση παραμέτρων (VPC) της υποδομής (δίκτυα, φιλοξενία, εφαρμογές και αποθήκευση) που είναι διαχειριζόμενα CSPs,
- τις υπηρεσίες τρίτων στις οποίες μπορούν να βασιστούν.

Εντούτοις, οι χρήστες δε μπορούν και δε δύναται να αποφύγουν τα καθήκοντα VPC που τους αντιστοιχούν. Επίσης, πρέπει από μέρους τους να κατανοήσουν στο μέγιστο βαθμό τις διάφορες πτυχές των VPC για τις οποίες είναι αρμόδιοι. Ένα πεδίο διαχείρισης VPC πρέπει να εξετάσει την ασφάλεια, καθώς και να συμπεριλάβει συστήματα και εφαρμογές διαχειριζόμενα από τον χρήστη που αλληλεπιδρούν με τις υπηρεσίες νέφους.

Σαν μια δεδομένη πρακτική, οι CSPs μπορεί να έχουν καθιερώσει αυτά τα προγράμματα μέσα στη περιοχή διαχείρισης ασφάλειάς τους, αλλά τυπικά αυτή η διαδικασία γίνεται εσωτερικά στο CSP και δεν εμφανίζεται στους χρήστες. Οι CSPs πρέπει να βεβαιώσουν τους χρήστες τους, για το ότι χρησιμοποιούν τύπο πλαισίου ελέγχου και διασφάλισης ISO/IEC 27002.

### 7.2.1 Διαχείριση ευπαθειών

Η διαχείριση ευπαθειών (security vulnerability management) δεν παύει να είναι ένα απαραίτητο στοιχείο απειλής που πρέπει να διαχειριστούν οι χρήστες. Σκοπό έχει να βοηθήσει τη προστασία της φιλοξενίας, των συσκευών δικτύου και των εφαρμογών από τις επιθέσεις ενάντια σε γνωστές ευπάθειες. Οι ώριμοι οργανισμοί έχουν καθιερώσει μια διαδικασία διαχείρισης ευπαθειών, όπου περιλαμβάνει τα εξής:

- Μια ρουτίνα εξέτασης των συστημάτων τα οποία είναι συνδεδεμένα στο δίκτυό τους, αξιολογώντας τους κινδύνους ευπάθειας στην οργάνωση, καθώς και
- Μια διαδικασία επανόρθωσης (συνήθως που τροφοδοτεί ένα πρόγραμμα διαχείρισης επιδιορθώσεων) για να αντιμετωπίσουν τους κινδύνους.

Οι οργανισμοί που χρησιμοποιούν το ISO/IEC 27002, γνωρίζουν το τρόπο να κατευθύνουν αυτό το πρόγραμμα, χρησιμοποιώντας έναν στόχο διαχείρισης ελέγχου τεχνικής ευπάθειας που αποσκοπεί στο να μειώσει τα αποτελέσματα κινδύνων που προκύπτουν από τις δημοσιευμένες τεχνικές αξιοποίησης ευπαθειών. Η διαχείριση τεχνικών ευπαθειών (Technical vulnerability management) πρέπει να εφαρμοστεί αποτελεσματικά, συστηματικά και με επαναλαμβανόμενο τρόπο, έτσι ώστε οι μετρήσεις που λαμβάνονται να επιβεβαιώνουν την αποτελεσματικότητά της διαχείρισης. Αυτές οι εκτιμήσεις πρέπει να περιλαμβάνουν τα λειτουργικά συστήματα και οποιεσδήποτε άλλες εφαρμογές σε χρήση.

Τόσο ο εξυπηρετούμενος όσο και το CSP είναι αρμόδιοι για τη διαχείριση ευπάθειας της υποδομής του νέφους, ανάλογα με την υπηρεσία SPI στο πλαίσιο εφαρμογής.

### 7.2.2 Διαχείριση επιδιορθώσεων ασφαλείας

Η διαχείριση επιδιορθώσεων ασφαλείας (security patch management), είναι παρόμοια με τη διαχείριση ευπαθειών. Ειδικότερα, η διαχείριση επιδιορθώσεων ασφαλείας είναι ένα ζωτικής σημασίας στοιχείο διαχείρισης απειλής, για τη προστασία των φιλοξενούμενων συσκευών δικτύων και των εφαρμογών από την εκμετάλλευση μιας γνωστής ευπάθειας από τους μη εξουσιοδοτημένους χρήστες. Οι διεργασίες διαχείρισης επιδιορθώσεων ακολουθούν ένα πλαίσιο διαχείρισης αλλαγής και τροφοδοτούνται άμεσα από τις ενέργειες που κατευθύνονται από το πρόγραμμα διαχείρισης ευπάθειας. Η διαχείριση επιδιορθώσεων ασφαλείας μετριάζει τον κίνδυνο για έναν οργανισμό τόσο από εσωτερικές όσο και από εξωτερικές απειλές. Ως εκ τούτου, οι πάροχοι ΛωΥ πρέπει συχνά να αξιολογούν τις νέες ευπάθειες και τις επιδιορθώσεις του firmware και του λογισμικού σε όλα τα συστήματα που συμμετέχουν στην παράδοση της υπηρεσίας \*ωΥ στους χρήστες.

Το πεδίο της ευθύνης της διαχείρισης ευπαθειών για τους χρήστες, θα έχει μια διαρκώς αυξανόμενη έκταση όσο μετακινούμαστε από τις υπηρεσίες τύπου ΛωΥ σε αυτές τύπου ΠωΥ και εν συνεχεία στις τύπου ΥωΥ. Πιο συγκεκριμένα, οι εξυπηρετούμενοι είναι απαλλαγμένοι από τα καθήκοντα διαχείρισης επιδιορθώσεων σε ένα περιβάλλον ΛωΥ, ενώ είναι αρμόδιοι για τη διαχείριση επιδιορθώσεων για ολόκληρη τη στοίβα του λογισμικού (λειτουργικό σύστημα, εφαρμογές, και βάση δεδομένων) που είναι εγκατεστημένο και λειτουργεί στην πλατφόρμα ΥωΥ. Οι πελάτες είναι επίσης αρμόδιοι για την επιδιόρθωση των εφαρμογών τους που εκτελούνται σε πλατφόρμες ΠωΥ.

### 7.2.3 Διαχείριση διαμόρφωσης ασφαλείας

Η διαχείριση διαμόρφωσης ασφαλείας είναι μια άλλη σημαντική πρακτική διαχείρισης απειλών για την προστασία των δικτυακών συσκευών δικτύων και των εφαρμογών που φιλοξενούνται από τους μη εξουσιοδοτημένους χρήστες που εκμεταλλεύονται οποιαδήποτε αδυναμία στις ρυθμίσεις παραμέτρων. Η διαχείριση διαμόρφωσης ασφαλείας συσχετίζεται πολύ με το πρόγραμμα διαχείρισης ευπάθειας και είναι ένα υποσύνολο της γενικής διαχείρισης ρύθμισης παραμέτρων.

Η διαχείριση διαμόρφωσης ασφαλείας συνίσταται στον ορισμό παραμέτρων που αφορούν την επίβλεψη και τον έλεγχο πρόσβασης σε κρίσιμα συστήματα και αρχεία ρυθμίσεων βάσεων δεδομένων, συμπεριλαμβανομένων:

- της ρύθμισης παραμέτρων του λειτουργικού συστήματος,
- τις πολιτικές του τείχους προστασίας,
- τη ρύθμιση παραμέτρων του δικτύου,
- Τη τοπική και απομακρυσμένη αποθήκευση και
- Τη βάση δεδομένων διαχείρισης του έλεγχου πρόσβασης.

Στο μοντέλο παροχής υπηρεσιών SPI, ο φόρτος της διαχείρισης διαμόρφωσης ασφαλείας από την οπτική της ευθύνης του χρήστη, βαίνει αυξανόμενος καθώς μετακινούμαστε από υπηρεσίες τύπου ΛωΥ, σε αυτές τύπου ΠωΥ και έτι περαιτέρω σε υπηρεσίες τύπου ΥωΥ. Δηλαδή, οι πάροχοι υπηρεσιών ΛωΥ και ΠωΥ είναι αρμόδιοι για τη διαχείριση των ρυθμίσεων παραμέτρων των πλατφορμών τους, ενώ οι χρήστες ΥωΥ είναι αρμόδιοι

για τη διαχείριση των ρυθμίσεων του λειτουργικού συστήματος, της εφαρμογής και της βάσης δεδομένων που φιλοξενείται στην πλατφόρμα ΥωΥ. Οι χρήστες είναι επίσης αρμόδιοι για τη διαχείριση των ρυθμίσεων παραμέτρων των εφαρμογών τους που δραστηριοποιούνται στη πλατφόρμα ΠωΥ.

### **7.3 Απόρρητο – προστασία προσωπικών δεδομένων**

Σε σχέση με την προστασία των προσωπικών δεδομένων, θα πρέπει να αποσαφηνιστεί ότι δεν είναι υποσύνολο της ασφάλειας πληροφοριών. Όπως αναφέρει και ο Tim Mather “κάποιος μπορεί να έχει ασφάλεια αλλά να μην διαφυλάσσεται το απόρρητό του, αλλά κανείς δεν μπορεί να διαφυλάξει το απόρρητό του χωρίς την ύπαρξη ασφάλειας”. Επιπρόσθετα, σημειώνεται ότι η ασφάλεια υποδομής και δεδομένων στο δημόσιο υπολογιστικό νέφος είναι πιθανό να είναι υποβαθμισμένη, σε σχέση με αυτή που υπάρχει σε συγκεκριμένους οργανισμούς. Με βάση αυτή τη πιθανότητα της μικρότερης ασφάλειας, συνεπάγεται ότι ο κίνδυνος παραβίασης του απορρήτου αυξάνεται επίσης. Πρέπει εντούτοις να σημειωθεί ότι πολλές μικρές και μεσαίου μεγέθους επιχειρήσεις (SMBs) έχουν περιορίσει τους πόρους που διαθέτουν στην ασφάλεια πληροφοριών και κατά συνέπεια δεν εστιάζουν ιδιαίτερα σε αυτήν την περιοχή. Για αυτούς τους οργανισμούς, η ασφάλεια που διατίθεται από έναν δημόσιο πάροχο υπηρεσιών νέφους (CSP) μπορεί να είναι μεγαλύτερη.

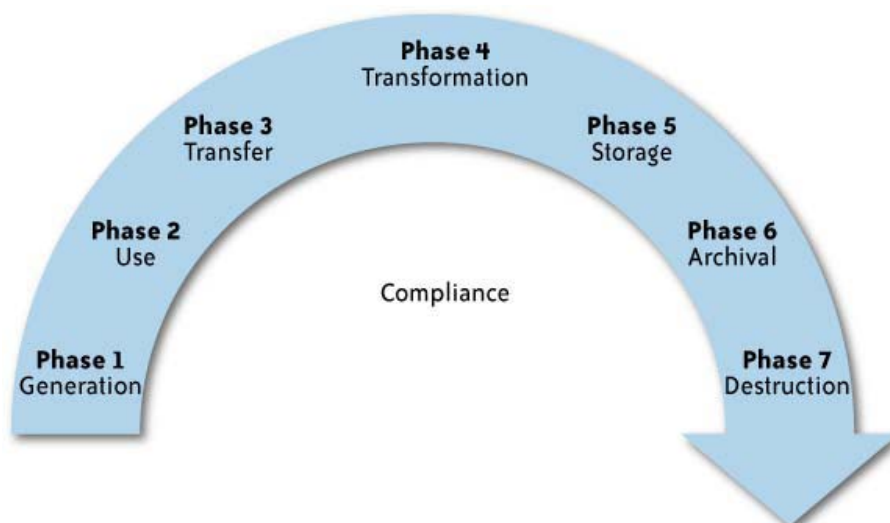
Επίσης θα πρέπει να λάβουμε υπ’ όψιν ότι και μια φαινομενικά μικρή παραβίαση δεδομένων μπορεί να έχει σοβαρές επιπτώσεις, καθώς επίσης και μακροπρόθεσμες συνέπειες, όπως η αρνητική δημοσιότητα και η απώλεια της εμπιστοσύνης των χρηστών.

#### **7.3.1 Έννοια απορρήτου**

Η αντίληψη για την έννοια του απορρήτου ποικίλλει ευρέως μεταξύ των χωρών. Επίσης, διαμορφώνεται από τις δημόσιες προσδοκίες και τις νομικές ερμηνείες, και με αυτά τα δεδομένα ένας συνοπτικός ορισμός είναι σχεδόν αδύνατος. Τα δικαιώματα ή οι υποχρεώσεις απορρήτου συσχετίζονται με τη συλλογή, χρήση, κοινοποίηση, αποθήκευση και καταστροφή των προσωπικών δεδομένων (ή των προσωπικά αναγνωρίσιμων πληροφοριών – personally identifiable information – PII). Συνολικά, το απόρρητο αφορά την υπευθυνότητα των οργανισμών σε θέματα δεδομένων, καθώς επίσης και τη διαφάνεια των πρακτικών ενός οργανισμού σχετικά με τις προσωπικές πληροφορίες.

#### **7.3.2 Κύκλος ζωής δεδομένων**

Οι πληροφορίες προσωπικού περιεχομένου, χρειάζεται να υπόκεινται σε διαχείριση ως τμήμα των δεδομένων τα οποία χρησιμοποιούνται εκ μέρους του οργανισμού. Επίσης, η πληροφορία πρέπει να υπόκειται σε διαχείριση από τη χρονική στιγμή την οποία λαμβάνεται, μέχρι και την τελική της διάθεση. Οτιδήποτε αφορά την προστασία των προσωπικών δεδομένων-πληροφοριών, χρειάζεται να εξετάσει τον αντίκτυπο του νέφους σε κάθε μια από τις ακόλουθες φάσεις, όπως αυτές παρουσιάζονται στην Εικόνα 39.



**Εικόνα 39 - Κύκλος ζωής δεδομένων**

Τα στοιχεία σε κάθε μια από αυτές τις φάσεις είναι τα ακόλουθα.

#### 7.3.2.1 Παραγωγή της πληροφορίας (Generation of the information)

- **Ιδιοκτησία (Ownership)**: Ποιός στον οργανισμό κατέχει ΡΠΙ και με ποιό τρόπο διατηρείται η ιδιοκτησία, στην περίπτωση εκείνη που ο οργανισμός χρησιμοποιεί το υπολογιστικό νέφος;
- **Ταξινόμηση (Classification)**: Με ποιόν τρόπο και πότε είναι οι ΡΠΙ ταξινομημένες; Ενδέχεται να υπάρχουν περιορισμοί στη χρήση του υπολογιστικού νέφους για τις συγκεκριμένες κατηγορίες δεδομένων;
- **Διακυβέρνηση (Governance)**: Μήπως υπάρχει κάποια δομή διακυβέρνησης, η οποία να εξασφαλίσει ότι οι ΡΠΙ υπόκεινται σε διαχείριση και ταυτοχρόνως προστατεύονται κατά τη διάρκεια του κύκλου ζωής τους, ακόμη και όταν αποθηκεύονται ή υποβάλλονται σε επεξεργασία σε ένα περιβάλλον υπολογιστικού νέφους;

#### 7.3.2.2 Χρήση (Use)

- **Σύγκριση εσωτερικού – εξωτερικού (Internal versus external)**: Οι ΡΠΙ χρησιμοποιούνται μόνο μέσα στον οργανισμό ή μήπως χρησιμοποιούνται έξω από αυτόν, όπως παραδείγματος χάριν σε ένα δημόσιο νέφος;
- **Τρίτο μέρος (Third party)**: Οι πληροφορίες διαμοιράζονται με τρίτους, όπως παραδείγματος χάριν οι υπεργολάβοι ή CSPs;
- **Καταλληλότητα (Appropriateness)**: Είναι η χρήση των πληροφοριών σύμφωνη, με τον πρωταρχικό στόχο για τον οποίο αυτές συλλέχθηκαν; Είναι η χρήση μέσα στο νέφος κατάλληλη, βάση των δεσμεύσεων του οργανισμού σε θέματα που αφορούν προσωπικά δεδομένα;
- **Ανακάλυψη/κλήτευση (Discovery/subpoena)**: Ο τρόπος διαχείρισης των πληροφοριών στο νέφος είναι τέτοιος που θα δίνει την ευχέρεια στον

οργανισμό, να ‘συμβαδίσει’ με τις νομικές απαιτήσεις σε περίπτωση νομικών διαδικασιών;

### 7.3.2.3 Μεταφορά (Transfer)

- Σύγκριση δημοσίων – ιδιωτικών δικτύων: Στην περίπτωση εκείνη την οποία οι πληροφορίες μεταφέρονται σε ένα νέφος, ο οργανισμός χρησιμοποιεί δημόσια δίκτυα και –αν ναι– αυτά προστατεύονται όπως επιβάλλεται; (Οι PII επιβεβλημένη ανάγκη είναι να προστατεύονται απόλυτα και πάντα, προκειμένου να αντιμετωπίσουν τα επίπεδα κινδύνου και τις νομικές απαιτήσεις.)
- Απαιτήσεις κρυπτογράφησης (Encryption requirements): Είναι εφικτό να κρυπτογραφούνται οι PII; Μερικοί νόμοι απαιτούν οι PII να είναι κρυπτογραφημένες όταν διαβιβάζονται μέσω ενός δημόσιου δικτύου (και αυτό θα συμβεί όταν ο οργανισμός χρησιμοποιεί ένα δημόσιο νέφος).
- Έλεγχος πρόσβασης (Access control): Γίνονται οι απαιτούμενοι έλεγχοι πρόσβασης στις PII, όταν είναι στο νέφος;

### 7.3.2.4 Μετασχηματισμός (Transformation)

- Παραγωγή (Derivation): Είναι η αρχική προστασία και οι περιορισμοί χρήσης που ελέγχουν πότε το δεδομένο είναι μετασχηματισμένο και δεν είναι στην αρχική του μορφή είτε πρόκειται να επεξεργαστεί περαιτέρω στο νέφος;
- Συγκέντρωση (Aggregation): Είναι τα δεδομένα συγκεντρωμένα στο νέφος έτσι ώστε να μην συσχετίζονται πλέον με ένα ευπροσδιόριστο άτομο (και ως εκ τούτου δεν θεωρούνται πλέον PII);
- Ακεραιότητα (Integrity): Η ακεραιότητα των PII μπορεί να διατηρηθεί όταν είναι στο νέφος;

### 7.3.2.5 Αποθήκευση (Storage)

- Έλεγχος πρόσβασης: Γίνονται οι απαραίτητοι έλεγχοι πρόσβασης στα PII όταν αποθηκεύονται στο νέφος, έτσι ώστε μόνο τα άτομα που είναι εξουσιοδοτημένα να έχουν πρόσβαση σε αυτά;
- Σύγκριση δομημένου – μη δομημένου: Πώς είναι τα δεδομένα αποθηκευμένα για να επιτρέψουν στον οργανισμό να έχει πρόσβαση και διαχείριση των δεδομένων στο μέλλον;
- Ακεραιότητα-διαθεσιμότητα-Εμπιστευτικότητα (Integrity-availability-confidentiality): Πώς διατηρείται η ακεραιότητα, διαθεσιμότητα και εμπιστευτικότητα των δεδομένων στο νέφος;
- Κρυπτογράφηση (Encryption): Διάφοροι νόμοι και κανονισμοί απαιτούν ότι ορισμένοι τύποι PII πρέπει να αποθηκεύονται μόνο όταν είναι κρυπτογραφημένοι. Αυτή η απαίτηση υποστηρίζεται από το CSP;

### 7.3.2.6 Αρχαιοθέτηση (Archival)

- **Νόμιμο και συμμορφούμενο (Legal and compliany):** Τα ΡΙΙ μπορεί να έχουν συγκεκριμένες απαιτήσεις, οι οποίες και αναφέρονται στο πόσο καιρό αυτά θα πρέπει να αποθηκεύονται και να αρχειοθετούνται. Αυτές οι απαιτήσεις υποστηρίζονται από το CSP;
- **Ζητήματα αποθήκευσης εκτός εγκαταστάσεων (Off-site considerations):** Το CSP παρέχει τη δυνατότητα για μακροπρόθεσμη αποθήκευση εκτός των εγκαταστάσεων που υποστηρίζουν απαιτήσεις αρχειοθέτησης;
- **Ζητήματα μέσων (Media concerns):** Οι πληροφορίες αποθηκεύονται σε μέσα που θα είναι προσιτά στο μέλλον; Οι πληροφορίες αποθηκεύονται σε φορητά μέσα που μπορεί να είναι πιο ευαίσθητα στην απώλεια; Ποιος ελέγχει τα μέσα και ποια είναι η δυνατότητα του οργανισμού να ανακτήσει τέτοια μέσα από το CSP εάν κριθεί απαραίτητο;
- **Διατήρηση (Retention):** Πόσο καιρό τα δεδομένα θα διατηρηθούν από το CSP; Είναι η περίοδος διατήρησης σύμφωνη με τη περίοδο διατήρησης του οργανισμού;

### 7.3.2.7 Καταστροφή (Destruction)

- **Ασφάλεια (Secure):** Το CSP καταστρέφει τα ΡΙΙ που λαμβάνονται από τους χρήστες κατά τρόπο ασφαλή ώστε να αποφευχθεί πιθανή παραβίαση των πληροφοριών;
- **Ολοκλήρωση (Complete):** Οι διάφορες πληροφορίες καταστρέφονται παντελώς; Η καταστροφή σβήνει εντελώς τα δεδομένα ή υπάρχει η πιθανότητα αυτά να ανακτηθούν κάποια στιγμή;

## 7.4 Προβληματισμοί που αφορούν την ασφάλεια προσωπικών δεδομένων στο νέφος

Οι κυριότεροι προβληματισμοί που σχετίζονται με το απόρρητο – ασφάλεια προσωπικών δεδομένων στο νέφος, αφορούν τα ακόλουθα.

### 7.4.1 Πρόσβαση (Access)

Η ανησυχία που κυριαρχεί σε σχέση με την ανάπτυξη υπηρεσιών στο νέφος, είναι η δυνατότητα του οργανισμού να παρέχει στα άτομα πρόσβαση σε όλες τις προσωπικές πληροφορίες και να συμμορφώνεται με τα δηλωμένα αιτήματα. Ένα κρίσιμο θέμα το οποίο προκύπτει είναι το κατά πόσο σίγουρο είναι, ότι εάν ένας χρήστης ο οποίος θα ζητήσει από τον οργανισμό να διαγραφούν τα στοιχεία του, ο οργανισμός αυτός θα μπορεί να εξασφαλίσει παράλληλα το γεγονός ότι όλες οι πληροφορίες του χρήστη αυτού, θα διαγραφούν στο νέφος.

### 7.4.2 Συμμόρφωση (Compliance)

Το θέμα στο οποίο δίνεται έμφαση στο σημείο αυτό, αφορά στο ποιές είναι οι απαιτήσεις συμμόρφωσης απορρήτου στο νέφος. Με άλλα λόγια, το πρόβλημα που γεννάται είναι

ποιοί είναι οι εφαρμοστέοι νόμοι, οι κανονισμοί, τα πρότυπα και οι συμβατικές υποχρεώσεις, που διέπουν αυτές τις πληροφορίες, καθώς και ποιος είναι αρμόδιος για τη διατήρηση της συμμόρφωσης. Τα νέφη μπορούν να εμπίπτουν στην αρμοδιότητα πολλών φορέων, για παράδειγμα τα δεδομένα στα νέφη μπορεί να είναι αποθηκευμένα σε πολλές χώρες. Άρα, το πρόβλημα που ανακύπτει σχετίζεται με το ποιος έχει την αρμοδιότητα ελέγχου των δεδομένων μιας οντότητας στο νέφος και πώς αυτό καθορίζεται;

### **7.4.3 Αποθήκευση (Storage)**

Ο προβληματισμός που κυριαρχεί αναφορικά με το θέμα αυτό της αποθήκευσης, σχετίζεται με το πού ακριβώς αποθηκεύονται τα δεδομένα στο νέφος. Οι νόμοι απορρήτου σε διάφορες χώρες, θέτουν βέβαια και διάφορους περιορισμούς. Ένας από αυτούς τους περιορισμούς, αφορά τη μετακίνηση διάφορων τύπων προσωπικών δεδομένων σε άλλες χώρες, από έναν οργανισμό. Γενικά, μια μεταφορά που μπορεί να προκύψει χωρίς τη γνώση του οργανισμού, όταν δηλαδή ένα δεδομένο αποθηκεύεται στο νέφος, ως συνέπεια αυτού είναι επομένως να υπάρξει μια πιθανή παραβίαση του τοπικού νόμου.

### **7.4.4 Διατήρηση (Retention)**

Τα προβλήματα που αφορούν τη διατήρηση δεδομένων στο νέφος, αναφέρονται στο χρονικό όριο διατήρησης των προσωπικών πληροφοριών που έχουν μεταφερθεί στο νέφος. Επίσης, μεταξύ των προβλημάτων συμπεριλαμβάνεται και η πολιτική διατήρησης που ελέγχει τα δεδομένα και την ιδιοκτησία των δεδομένων (αν δηλαδή ανήκουν στον οργανισμό ή στο CSP).

### **7.4.5 Καταστροφή (Destruction)**

Όταν παρέλθει η περίοδος της διατήρησης, τα δεδομένα επιβάλλεται να καταστραφούν. Βέβαια, το θέμα το οποίο ανακύπτει είναι, στο πώς ο πάροχος νέφους καταστρέφει τα PII στο τέλος της περιόδου διατήρησης και στο πώς οι οργανισμοί εξασφαλίζουν ότι τα PII τους καταστρέφονται από τον CSP στο σωστό χρονικό σημείο. Ακόμη, ένα άλλο πρόβλημα έγκειται στο γεγονός ότι τα παραπάνω, δεν είναι διαθέσιμα σε άλλους χρήστες νεφών.

Όσον αφορά τους παρόχους νέφους αποθήκευσης, αυτοί αναπαράγουν συνήθως τα δεδομένα στα πολλαπλά συστήματα. Με τον τρόπο αυτό αυξάνεται η διαθεσιμότητα, η οποία δε παύει να αποτελεί ένα από τα σημαντικά οφέλη. Βέβαια, το συγκεκριμένο όφελος μετατρέπεται σε μειονέκτημα, τη στιγμή εκείνη την οποία ο οργανισμός προσπαθεί να καταστρέψει τα δεδομένα, καθ' ότι δεν είναι σίγουρο ότι μπορεί να τα καταστρέψει όλα με μια μόνο ενέργεια.

### **7.4.6 Έλεγχος και παρακολούθηση (Audit and monitoring)**

Ο έλεγχος και η παρακολούθηση αφορούν τον τρόπο εκείνο χάρη στον οποίο οι οργανισμοί, μπορούν να ελέγξουν τους CSP τους, αλλά και να διαβεβαιώνουν τους ενδιαφερόμενους, ότι οι απαιτήσεις απορρήτου καλύπτονται όταν τα PII τους βρίσκονται στο νέφος.

### 7.4.7 Παραβιάσεις απορρήτου (Privacy breaches)

Ο προβληματισμός που αφορά τις παραβιάσεις απορρήτου, σχετίζεται με το πώς γνωστοποιείται ότι μια παραβίαση έχει εμφανιστεί. Επίσης, με το πώς εξασφαλίζεται ότι το CSP ειδοποιεί όταν μια παραβίαση εμφανίζεται, καθώς και με το ποιος είναι αρμόδιος για τη διαχείριση της διαδικασίας ανακοίνωσης παραβιάσεων.

## 7.5 Αλλαγές στη διαχείριση και τη συμμόρφωση κινδύνου προστασίας προσωπικών δεδομένων σε σχέση με το υπολογιστικό νέφος

Στις υπο-ενότητες που ακολουθούν, περιγράφεται ο αντίκτυπος του υπολογιστικού νέφους σε βασικά θέματα αρχών απορρήτου.

### 7.5.1 Αρχή περιορισμού συλλογής (Collection Limitation Principle)

Η εν λόγω αρχή περιορισμού συλλογής, διευκρινίζει ότι η συλλογή των προσωπικών δεδομένων, πρέπει να περιοριστεί στην ελάχιστη ποσότητα δεδομένων που απαιτείται για το σκοπό για τον οποίο συλλέγονται. Οποιαδήποτε τέτοια δεδομένα, πρέπει να αποκτηθούν με νόμιμα και δίκαια μέσα και μόνο όπου απαιτείται.

Η μη επαρκής πληροφόρηση των προμηθευτών δεδομένων σε σχέση με τα δεδομένα που συλλέγονται και την προτιθέμενη χρήση τους δημιουργεί παρανοήσεις. Υπάρχουν περιεκτικά πλαίσια ασφάλειας, καθώς και προτύπων όπως το ISO 27000, οι οδηγίες του NIS και διάφορα άλλα, όπου οι οργανισμοί οφείλουν να ξέρουν πώς να τα εφαρμόσουν. Δεν υπάρχει κανένα παγκοσμίως υιοθετημένο πρότυπο απορρήτου, αντ' αυτού υπάρχουν συγκρουόμενοι νόμοι, κανονισμοί και απόψεις για το τι είναι το απόρρητο και τι απαιτείται από τους οργανισμούς για να το προστατεύσουν.

Επιπλέον, είναι σημαντικό, οι συμφωνίες παροχής υπηρεσιών (SLAs) να καθορίζονται πριν από την παροχή ή τον διαμοιρασμό οποιασδήποτε πληροφορίας, για το λόγο ότι είναι δύσκολο να συζητηθούν αργότερα.

### 7.5.2 Αρχή περιορισμού χρήσης (Use Limitation Principle)

Η αρχή περιορισμού χρήσης, διευκρινίζει ότι τα προσωπικά δεδομένα δε πρέπει να αποκαλυφθούν, αλλά ούτε να χρησιμοποιηθούν για άλλους λόγους εκτός από εκείνους τους οποίους αφορούν και που αρχικά έχουν συμφωνηθεί.

Το υπολογιστικό νέφος τοποθετεί μια διαφορετική συλλογή από πληροφορίες χρηστών και επιχειρήσεων σε μια ενιαία θέση. Όπως τα δεδομένα ρέουν μέσα στο νέφος, απαιτείται μια ισχυρή πολιτική διαχείρισης δεδομένων για να εξασφαλίσει ότι ο αρχικός σκοπός της συλλογής και του περιορισμού στη χρήση είναι συνδεδεμένος με τα δεδομένα. Αυτό είναι κρίσιμο όταν οι οργανισμοί δημιουργούν μια κεντρική βάση δεδομένων, επειδή οι μελλοντικές εφαρμογές μπορούν εύκολα να συνδυάσουν τα δεδομένα για νέους σκοπούς.

Εν συνεχεία, η δυνατότητα να συνδυαστούν τα δεδομένα από πολλαπλές πηγές, έχει σαν αποτέλεσμα να αυξάνει τον κίνδυνο απροσδόκητων χρήσεων εκ μέρους των κυβερνήσεων. Οι κυβερνήσεις διαφόρων χωρών, μπορούν να ζητήσουν από τους CSPs να υποβάλουν έκθεση, σχετικά με τους ιδιαίτερους τύπους συμπεριφορών, ακόμη και να



ελέγχουν τις δραστηριότητες ιδιαίτερων τύπων ή κατηγοριών χρηστών. Η δυνατότητα ότι ένα CSP θα μπορούσε να υποχρεωθεί να ενημερώνει μια κυβέρνηση, ακόμη και έναν τρίτο για τις δραστηριότητες των χρηστών του, μπορεί να δημιουργήσει προβλήματα και στον πάροχο και στους χρήστες του.

### 7.5.3 Αρχή ασφάλειας (Security Principle)

Η ασφάλεια αποτελεί μια από τις βασικές απαιτήσεις του απορρήτου. Η εν λόγω αρχή καθιστά σαφές, ότι τα προσωπικά δεδομένα πρέπει να προστατευθούν μέσω λογικών εγγυήσεων ασφάλειας από κινδύνους όπως π.χ.:

- Η απώλεια,
- Η αναρμόδια πρόσβαση,
- Η καταστροφή,
- Η χρήση,
- Η τροποποίηση και
- Η κοινοποίηση των στοιχείων.

### 7.5.4 Αρχή διατήρησης και καταστροφής (Retention and Destruction Principle)

Η αρχή που αφορά τη διατήρηση και την καταστροφή, δίνει έμφαση στο ότι τα απόρρητα προσωπικά δεδομένα, επιβάλλεται να μη διατηρούνται παραπάνω χρονικό διάστημα, από αυτό που απαιτείται από τη νομοθεσία ή τους περιορισμούς, για να εκτελεστεί η εργασία για την οποία χρησιμοποιήθηκαν. Στη συνέχεια, τα δεδομένα χρήσιμο είναι να καταστραφούν με ιδιαίτερη ασφάλεια, όταν η περίοδος διατήρησης φτάσει στο πέρας της. Για τους οργανισμούς και τις επιχειρήσεις, διαδραματίζει ιδιαίτερα σημαντικό ρόλο, το χρονικό διάστημα εκείνο το οποίο τα δεδομένα πρέπει να διατηρηθούν ακέραια και ο τρόπος που αυτά θα προβούν σε ολική καταστροφή.

Γενικά, η πραγματική διαδικασία διαγραφής μερικές φορές καθορίζεται αόριστα. Το γεγονός αυτό δημιουργεί αμφιβολίες για το κατά πόσο τα δεδομένα που έχουν αναπαραχθεί ή που έχουν αντίγραφα ασφαλείας, όντως διαγράφονται. Η διαγραφή ενός αρχείου πολλές φορές αντιστοιχεί στη σημείωση ότι τα μπλοκ που καταλάμβανε είναι πλέον ελεύθερα. Έως ότου επικαλυφθούν πραγματικά τα blocks, τα δεδομένα είναι ακόμη εκεί και μπορούν να ανακτηθούν.

Η κρυπτογράφηση των δεδομένων διευκολύνει σημαντικά την καταστροφή τους. Τα κρυπτογραφημένα δεδομένα μπορούν να καταστραφούν, ακόμη και όταν οι οργανισμοί δεν έχουν πλέον πρόσβαση στα δεδομένα τους, καταστρέφοντας το κλειδί κρυπτογράφησης.

### 7.5.5 Αρχή μεταφοράς (Transfer Principle)

Σύμφωνα με την αρχή μεταφοράς, τα δεδομένα δε πρέπει να μεταφέρονται σε χώρες οι οποίες παρέχουν το ίδιο επίπεδο προστασίας απορρήτου με τον οργανισμό που συνέλεξε τις πληροφορίες. Σε ένα περιβάλλον υπολογιστικού νέφους, η υποδομή μοιράζεται

μεταξύ των οργανισμών, επομένως, υπάρχουν απειλές που συνδέονται με το γεγονός ότι τα δεδομένα αποθηκεύονται και εν συνεχεία υποβάλλονται σε επεξεργασία, έχοντας απομακρυσμένη πρόσβαση. Αυξάνεται έτσι η ανάγκη να προστατευτεί το απόρρητο των δεδομένων, τα οποία αποθηκεύονται στο νέφος.

Η μεταφορά δεδομένων γίνεται ιδιαίτερα περίπλοκη, διότι τα δεδομένα μπορούν και ενδέχεται να είναι οπουδήποτε στον κόσμο. Αντί τα δεδομένα να αποθηκεύονται στους εξυπηρετές της επιχείρησης, αποθηκεύονται στους εξυπηρετές του παρόχου υπηρεσιών, οι οποίοι μπορεί να είναι στην Ευρώπη, στη Κίνα ή οπουδήποτε αλλού. Το γεγονός αυτό μπορεί να φέρνει το υπολογιστικό νέφος σε σύγκρουση με διάφορες νομικές απαιτήσεις, όπως οι ευρωπαϊκοί νόμοι που απαιτούν από την επιχείρηση να γνωρίζει που βρίσκονται τα προσωπικά δεδομένα που έχει στη κατοχή της ανά πάσα στιγμή.

Ένας πάροχος υπηρεσιών εφαρμογών υπολογιστικού νέφους (ASP), προσφέρει στους εξυπηρετούμενους του την επιλογή να αποθηκεύσουν τα δεδομένα τους μόνο σε ευρωπαϊκούς εξυπηρετές (έναντι υψηλότερης αμοιβής). Εντούτοις, είναι μια μη πρακτική λύση επειδή περιορίζει την ευελιξία και την αποδοτικότητα που παρέχουν τα υπολογιστικά νέφη.

### **7.5.6 Αρχή υπευθυνότητας (Accountability Principle)**

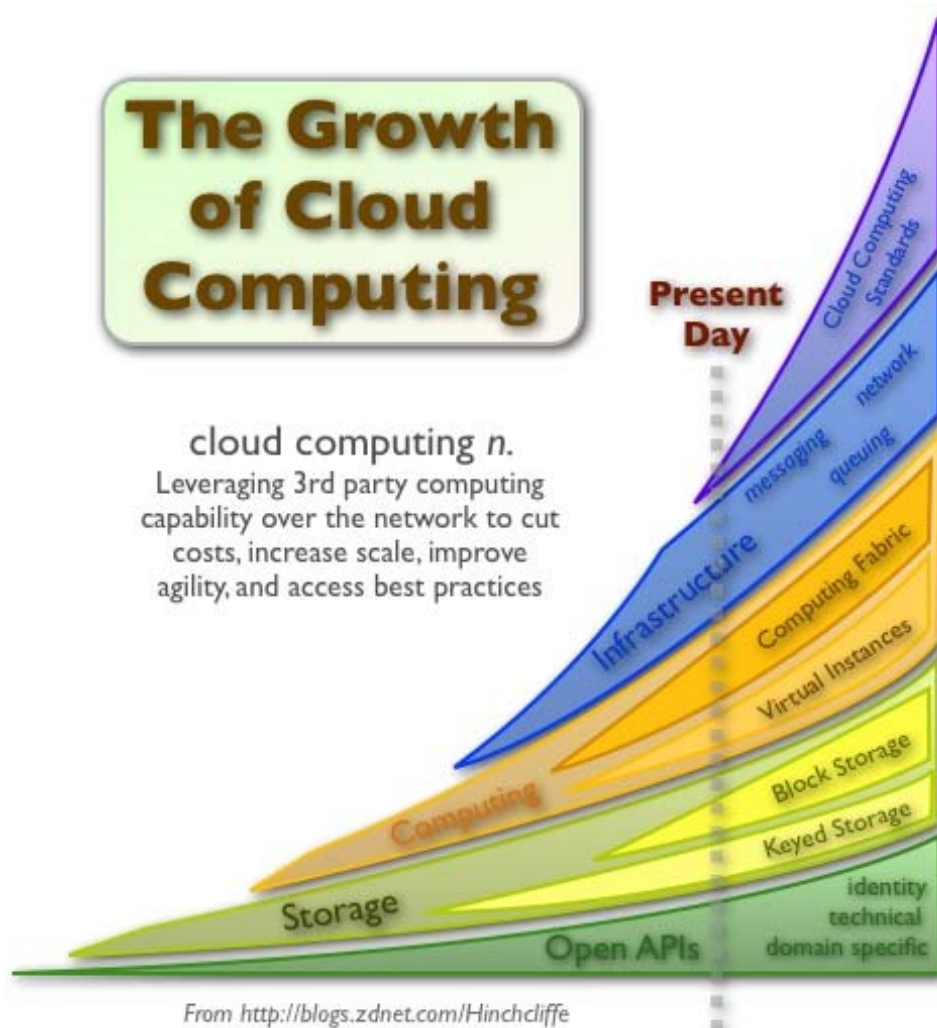
Τελευταία αρχή αποτελεί αυτή της υπευθυνότητας, η οποία υποδηλώνει ότι ένας οργανισμός είναι υπεύθυνος για τις προσωπικές πληροφορίες οι οποίες βρίσκονται υπό από τον έλεγχο του. Επιβάλλεται να υποδείξει κάποια άτομα, τα οποία να διακρίνονται για την υπευθυνότητά τους στη συμμόρφωση του οργανισμού με τις υπόλοιπες αρχές.

Η υπευθυνότητα μέσα στο υπολογιστικό νέφος, μπορεί να επιτευχθεί με την ενσωμάτωση πολιτικών στα δεδομένα και με τη χρήση μηχανισμών. Είναι χρήσιμο να συμβεί αυτό, με σκοπό να εξασφαλιστεί πως οι πολιτικές για τις οποίες γίνεται λόγος, υιοθετούνται από τα συμβαλλόμενα μέρη. Τα μέρη αυτά είτε χρησιμοποιούν είτε αποθηκεύουν είτε μοιράζονται εκείνα τα δεδομένα, ανεξάρτητα από την αρμοδιότητα στην οποία οι πληροφορίες υποβάλλονται σε επεξεργασία.

## Κεφάλαιο 8: Το μέλλον του υπολογιστικού νέφους

Στο τελευταίο αυτό κεφάλαιο παρατίθεται μια ανάλυση σχετικά με το μέλλον του υπολογιστικού νέφους [10]. Σαφέστατα δεν υπάρχει κάποιος κανόνας ή κάποιος αλγόριθμος σύμφωνα με τον οποίο μπορούμε να υπολογίσουμε με ασφάλεια πως θα είναι το νέφος στο μέλλον και ποιες υπηρεσίες θα προσφέρει. Ωστόσο με βάση τα δεδομένα που ισχύουν μέχρι σήμερα, άλλα και τη δεδομένη εξέλιξη της τεχνολογίας θα παρουσιάσουμε κάποιες προβλέψεις για το πώς πιθανολογείται ότι θα εξελιχθεί ο τομέας αυτός.

Οι προβλέψεις που θα ακολουθήσουν, χωρίζονται σε δυο ενότητες. Η πρώτη αφορά την εξέλιξη του νέφους γενικά και η δεύτερη την εξέλιξη των εφαρμογών που θα εκτελούνται ή θα λαμβάνουν χώρα στο νέφος [9].



Εικόνα 40 - Μέλλον του υπολογιστικού νέφους

## **8.1 Προβλέψεις για το μέλλον του νέφους**

### **8.1.1 Εργαλείο μεγέθυνσης για τους νεοεισερχόμενους**

Ένα ουσιαστικό θέμα που τίθεται είναι πως το υπολογιστικό νέφος θα μετατραπεί σε μια πραγματική μηχανή μεγέθυνσης για τις πρώτες επιχειρήσεις που θα το υιοθετήσουν. Στη συνέχεια, οι επιχειρήσεις αυτές θα είναι σε θέση να φτάσουν αρκετά γρήγορα τις ανταγωνίστριες τους, καθώς και να αποκτήσουν πλεονέκτημα έναντι αυτών των επιχειρήσεων που δε χρησιμοποιούν καθόλου τις υπηρεσίες του υπολογιστικού νέφους.

Με βάση το γεγονός ότι ήδη πεντακόσιες χιλιάδες επιχειρήσεις χρησιμοποιούν τις υπηρεσίες δικτύου του Amazon (Amazon Web Services - AWS), πλέον πολλές νέες επιχειρήσεις δεν αγοράζουν εξυπηρετές. Αντ' αυτού χρησιμοποιούν το νέφος, καθώς με τον τρόπο αυτό εξοικονομούν σημαντικά χρηματικά ποσά από τα κεφάλαιά τους. Δίνεται η δυνατότητα έτσι στις επιχειρήσεις να επενδύουν τα χρήματά τους σε άλλες λειτουργίες, έτσι ώστε να μπορέσουν να επεκταθούν στην αγορά γρηγορότερα, έναντι των ανταγωνιστών τους.

### **8.1.2 Αξιοπιστία – οικονομία – ασφάλεια – ευκολία**

Σύμφωνα με τη λογική εξέλιξη των πραγμάτων, το νέφος θα είναι φθηνότερο, περισσότερο αξιόπιστο, ασφαλέστερο, αλλά και ευκολότερο στη χρήση. Οι επιχειρήσεις όπως αναφέρθηκε και προηγουμένως θα στρέφονται σε λύσεις υπολογιστικού νέφους, καθώς το κόστος είναι χαμηλότερο από αυτό της αγοράς και της συντήρησης ιδιόκτητων εφαρμογών και λειτουργιών. Επίσης, οι οικονομίες κλίμακας οδηγούν προς τα κάτω τις δαπάνες των παρόχων νεφών. Δεδομένου ότι οι πάροχοι νέφους χτίζουν ολοένα και μεγαλύτερα κέντρα δεδομένων, απαιτούν καλύτερες τιμές από τις εταιρίες ενέργειας. Κυρίως βέβαια συνεχίζουν να χρησιμοποιούν όσο το δυνατόν περισσότερο λογισμικό ανοικτού-κώδικα και ο σκληρός ανταγωνισμός θα αναγκάσει να διατηρηθούν οι τιμές σε χαμηλά επίπεδα.

Ο ανταγωνισμός και οι αυξημένες απαιτήσεις των εξυπηρετούμενων, θα αναγκάσουν τους παρόχους νεφών να υποσχεθούν ισχυρότερα συμφωνητικά παροχής υπηρεσιών (Service Level Agreement - SLAs) στα οποία θα πρέπει να εμμείνουν. Οι περισσότεροι πάροχοι νέφους θα διατηρούν πολλές εγκαταστάσεις, κάθε μια με την πλεονάζουσα ικανότητα να αντισταθμίσει τις παραβιάσεις υλικού ή δικτύου μετακινώντας το φόρτο εργασίας προς διαφορετικούς εξυπηρετές ή σε εξ ολοκλήρου διαφορετικές εγκαταστάσεις.

Αναφορικά με την ασφάλεια, το νέφος διαθέτει μεγαλύτερη εξειδίκευση στον τομέα αυτό της ασφάλειας ότι οι παραδοσιακές μέθοδοι (μεμονωμένη ανάπτυξη από εταιρείες). Έτσι, με την πάροδο του χρόνου θα επεκταθούν ακόμη περισσότερο οι προσπάθειες που γίνονται για να καταστούν τα νέφη ασφαλέστερα. Μέχρι και τα δημόσια νέφη θα παρέχουν καλύτερη ασφάλεια δεδομένων. Καθώς η μετάδοση των δεδομένων και η αποθήκευσή τους στις εγκαταστάσεις αποθήκευσης νέφους αναμένεται να παραμένουν κρυπτογραφημένες σε όλους τους τύπους νέφους, η ασφάλεια των δεδομένων στα κέντρα δεδομένων μη-νεφών θα βελτιωθεί δεδομένου ότι θα υιοθετούν οι καλύτερες πρακτικές που αρχικά δοκιμάστηκαν στα δημόσια και στα ιδιωτικά νέφη.

Τέλος, η ευκολία χρήσης του υπολογιστικού νέφους θα συνεχίσει να βελτιώνεται. Δεδομένου ότι όλα τα ανταγωνιστικά συστήματα λογισμικού τείνουν προς αυτή την κατεύθυνση, καθώς επίσης ότι είναι υπαρκτή η ψηφοφορία των χρηστών για τους καλύτερους παρόχους υπηρεσιών, είναι σχεδόν βέβαιο ότι η χρήση του νέφους θα γίνει πολύ πιο φιλική προς τον χρήστη

### **8.1.3 Αύξηση των εξυπηρετών στα κέντρα δεδομένων νέφους**

Τα μεγάλα κέντρα δεδομένων νέφους προβλέπεται ότι θα περιλαμβάνουν 500.000 εξυπηρετές, που θα κοστίζουν \$1 δισεκατομμύριο μέχρι το 2020. Ένας μικρός κατάλογος από τους μεγάλους παρόχους νέφους, συμπεριλαμβανομένων των Amazon, Apple, Google, Microsoft, Oracle, IBM, ακόμα και το Facebook, αποτελούν μια κατηγορία από μόνοι τους όταν πρόκειται για το μέγεθος του κέντρου δεδομένων τους.

Ένα χαρακτηριστικό παράδειγμα του κόστους αυτών των κέντρων, αποτελούν οι εγκαταστάσεις που διατηρεί στο Σικάγο η Microsoft, έχοντας μέγεθος 700.000 τετραγωνικά πόδια και κόστος \$500 εκατομμύρια. Σε ανάλογα επίπεδα κυμαίνονται και οι εγκαταστάσεις των υπόλοιπων εταιρειών που αναφέρθηκαν.

Όλοι αυτοί οι πάροχοι νέφους χτίζουν κέντρα δεδομένων που έχουν 50.000–100.000 εξυπηρετές. Ένα τέτοιο κέντρο κοστίζει περίπου \$53 εκατομμύρια ετησίως. Αυτό αποτελεί περίπου το 45% του συνολικού κόστους του κέντρου δεδομένων. Το 25% πηγαίνει στη διανομή ενέργειας και την ψύξη (\$18 εκατομμύρια/έτος), ένα 15% είναι για το κόστος ηλεκτρικής ενέργειας (\$9 εκατομμύρια/έτος) και τέλος ένα 15% είναι δαπάνες δικτύου.

### **8.1.4 Μειωμένο κόστος εν συγκρίσει με τα εταιρικά κέντρα δεδομένων**

Το ανηγμένο κόστος των κέντρων δεδομένων των παρόχων του υπολογιστικού νέφους είναι χαμηλότερο από το κόστος των εταιρικών κέντρων δεδομένων. Αυτό συμβαίνει διότι τα κέντρα δεδομένων των παρόχων του νέφους έχουν τεράστιες οικονομίες κλίμακας, καθώς και σημαντικές φοροαπαλλαγές και συνεχίζουν να μειώνουν τις δαπάνες τους κάθε έτος.

Επίσης, τα μεγάλα κέντρα δεδομένων νέφους υλοποιούν πολύπλοκα λογισμικά, τα οποία χρησιμεύουν ώστε να μετακινούν τον φόρτο εργασίας από μια εγκατάσταση σε κάποια άλλη σε περίπτωση αστοχίας. Προσπαθούν με αυτό τον τρόπο, καθώς και με άλλες τεχνικές να μειώσουν το επιπλέον κόστος. Τα επόμενα χρόνια το χάσμα μεταξύ των εταιρικών κέντρων αποθήκευσης δεδομένων και των παρόχων κέντρων αποθήκευσης δεδομένων σε νέφη θα συνεχίσει να διευρύνεται. Υπολογίζεται ότι μέχρι το 2020, οι δαπάνες παρόχων νέφους θα είναι λιγότερο από τα 25% σε σχέση με τα εταιρικά κέντρα δεδομένων.

### **8.1.5 Μείωση της αναλογίας διαχειριστών προς εξυπηρετές**

Έως το 2020 η αναλογία των διαχειριστών προς τους εξυπηρετές, θα ανέλθει από 1:1,000 σε 1:10,000. Αυτό διότι τα μεγάλα κέντρα δεδομένων επενδύουν ιδιαίτερα σε λογισμικό που αυτοματοποιεί τις διάφορες λειτουργίες. Ένα τυπικό εταιρικό κέντρο στοιχείων αναπτύσσει δραστηριότητες σε μια αναλογία περίπου 1 μέλους προσωπικού για κάθε 100

εξυπηρετές, ενώ οι καλύτεροι πάροχοι νέφους αναπτύσσουν δραστηριότητες με αναλογία 1:1,000.

Τα κέντρα δεδομένων νέφους δουλεύουν τόσο αποτελεσματικά, για το λόγο ότι έπρεπε να χτίσουν υψηλά επίπεδα αυτοματισμού στη λειτουργία τους για να προσφέρουν ένα βασικό πακέτο νέφους. Ένας τρόπος που μπορούν να επιτύχουν υψηλά επίπεδα αυτοματοποίησης, είναι με την τυποποίηση των πλατφορμών τους. Βέβαια μόνο μερικές διαφορετικές πλατφόρμες μπορούν να προμηθευτούν από την Amazon. Αλλά ένα τυπικό εταιρικό κέντρο δεδομένων, έχει έναν τεράστιο αριθμό εφαρμογών (συχνά χιλιάδες) με κυριολεκτικά εκατοντάδες διαφορετικές πιθανές ρυθμίσεις του υλικού και του λογισμικού. Η επιδείνωση αυτής της πολυπλοκότητας έχει σαν αποτέλεσμα, οι προβλέψεις για τη χρήση μοντέλων και οι κλιμακούμενες απαιτήσεις να είναι συνήθως αβέβαιες. Το τελικό αποτέλεσμα αυτής της σταθερής κίνησης για περισσότερη αυτοματοποίηση, που μισθώνει τους καλύτερους ανθρώπους και δημιουργεί μεγαλύτερα κέντρα δεδομένων, θα οδηγήσει τους παρόχους νέφους σε μια αναλογία ενός διαχειριστή για κάθε δέκα χιλιάδες εξυπηρετές στην επόμενη δεκαετία.

### 8.1.6 Η εξάπλωση του ανοικτού κώδικα

Το νέφος του μέλλοντος, αναμφίβολα θα κυριαρχείται από συστήματα ανοικτού λογισμικού, έτσι όπως γίνεται και σε διάφορους τομείς της επιστήμης των υπολογιστών. Τα συστήματα αυτά έχουν οδηγήσει την μέχρι τώρα εξέλιξη του υπολογιστικού νέφους. Βέβαια αυτό λειτουργεί αμφίδρομα καθώς η γρήγορη πρόοδος του νέφους τροφοδοτεί και την εξέλιξη του ανοικτού λογισμικού.

Το Linux είναι το λειτουργικό σύστημα που επιλέγεται στο νέφος. Κάποιοι πάροχοι νέφους εκτελούν το ανοιχτό λογισμικό RedHat Linux. Επίσης μια άλλη επικρατούσα επιλογή ανοικτού λογισμικού είναι το Xen το οποίο χρησιμοποιείται από χιλιάδες εξυπηρετές της Amazon που υποστηρίζουν το EC2. Κάποιες άλλες εναλλακτικές επιλογές είναι η Microsoft και το VMware, το κόστος των οποίων δεν είναι αμελητέο.

Το ανοιχτό λογισμικό είναι μια ισχυρή δύναμη. Αυτό γίνεται για το λόγο ότι οι δαπάνες απόκτησης και χορήγησης αδειών του λογισμικού ανοικτού κώδικα, είναι 80% χαμηλότερες από τις συγκρίσιμες ιδιόκτητες προσφορές. Τέλος, το νέφος είναι ιδανικό για τους προγραμματιστές ανοικτού κώδικα, λόγω του ότι έχουν τη δυνατότητα να το χρησιμοποιήσουν για ανάπτυξη, δοκιμή και αποθήκευση με χαμηλό κόστος.

### 8.1.7 Πραγματικά πρότυπα μέσω των APIs της Amazon

Τα APIs της Amazon προβλέπεται να ηγούνται μέσα στην αγορά αυτή. Ωστόσο, είναι δυνατόν να μετρηθεί η ωριμότητα μιας τεχνολογίας πληροφοριών IT με βάση τον αριθμό, το ποσοστό υιοθέτησης και το βάθος του προτύπου της. Το νέφος (τουλάχιστον το YωY μέρος του νέφους) είναι σήμερα στην πρόωρη ακμή του σε αυτή τη κλίμακα, αλλά αυτό θα αλλάξει γρήγορα.

Όπως αναφέρθηκε και νωρίτερα, πλήθος του συνόλου των προτύπων έχει διαμορφώσει διάφορες πτυχές της τυποποίησης νέφους και προγραμματιστικών διασυνδέσεων εφαρμογών APIs. Αλλά η ανάγκη θα οδηγήσει μερικές ομάδες στη δημιουργία των δικών τους προτύπων χωρίς να περιμένουν τη παροχή τους από τους οργανισμούς προτύπων. Παραδείγματος χάριν, οι Zend, Microsoft, IBM, Rackspace, Nirvanix, και

GoGrid ανέπτυξαν το Simple Cloud API ανοικτού κώδικα για να αλληλεπιδρά με διαφορετικούς παρόχους νέφους. Αρχικά, αυτό ήταν μια προσπάθεια να καταστεί εύκολο ώστε οι προγραμματιστές PHP να έχουν πρόσβαση σε υπηρεσίες πολλών νεφών.

Τη τελευταία δεκαετία, η κοινότητα της Τεχνολογίας της Πληροφορίας έχει ανακαλύψει ότι λειτουργεί καλύτερα στο να πάρει κάτι που ένας πάροχος (ο κύριος πάροχος σε μια κατηγορία) έχει αναπτύξει και να το μετατρέψει σε πρότυπα. Αυτό έρχεται σε αντιδιαστολή με τη δημιουργία μιας ομάδας μελέτης και μιας επιτροπής προτύπων, η οποία περνά από μια μακροχρόνια διαδικασία προκειμένου να δημοσιεύσει ένα διεθνές πρότυπο.

### 8.1.8 Η αύξηση της χρήσης ΛωΥ από τα βασικά πρότυπα του Ιστού

Τα επόμενα χρόνια αναμένεται να αυξηθεί η χρήση του λογισμικού ως υπηρεσία (ΛωΥ) μέσω των βασικών προτύπων του ιστού. Οι επιχειρήσεις είναι πιο εύκολο να υιοθετηθούν πρότυπα ΛωΥ, καθ' όσον το μόνο που απαιτείται για την χρήση τους είναι ένα πρόγραμμα πλοήγησης. Ακόμη, για το λογισμικό ως υπηρεσία δεν απαιτούνται πρόσθετα πρότυπα καθώς χρησιμοποιούνται τα ίδια πρότυπα πάνω στα οποία δημιουργούνται οι βασικοί δικτυακοί τόποι και οι εφαρμογές ιστού.

Δεδομένου ότι η HTML 5 υποστηρίζεται ευρέως από τα προγράμματα πλοήγησης, οι εφαρμογές ΛωΥ θα εκμεταλλευθούν τα χαρακτηριστικά γνωρίσματά της. Έτσι, θα προσφέρουν στους χρήστες μια πιο διαδραστική εμπειρία, εξαλείφοντας τα όποια παράπονα αυτών κατά τη μετάβαση τους στις εφαρμογές ΛωΥ.

Επίσης, σύμφωνα με μελέτη της Gartner (<http://www.gartner.com/technology/home.jsp>) το μέγεθος της αγοράς λογισμικού ως υπηρεσία (ΛωΥ), θα ανέρχεται στα \$16 δισεκατομμύρια μέχρι το 2013. Η IDC (<http://www.idc.com/>) συνεχίζει στην έκθεσή της ότι το ετήσιο ποσοστό αύξησης για το ΛωΥ είναι στο 40%. Περισσότερες από 76% των αμερικανικών εταιριών χρησιμοποιούν τουλάχιστον μια εφαρμογή ΛωΥ. Περίπου το 45% των επιχειρήσεων ξοδεύουν τουλάχιστον 25% των IT προϋπολογισμών σε ΛωΥ. Οι επιχειρήσεις αυτές μετατοπίζουν ήδη τους χρήστες τους στο νέφος, ακόμη και αν δεν είναι ακόμα πλήρως έτοιμες να μετατοπίσουν την υποδομή τους στο νέφος

### 8.1.9 Το πρότυπο νέφους ISO

Ως γνωστόν, το υπολογιστικό νέφος έχει ευρέως διαδοθεί στις μέρες μας και επεκτείνεται καθημερινά ολοένα και περισσότερο. Λόγω όμως της διαφημιστικής εκστρατείας, είναι εύκολο να ξεχαστεί ότι το νέφος είναι κάτι νέο. Τα νέα πράγματα ως εκ τούτου δεν έχουν πρότυπα, διότι είναι νωρίς για να εξαχθούν σωστά συμπεράσματα με σκοπό τη δημιουργία προτύπων. Προβλέπεται όμως ότι στα επόμενα χρόνια ένα πρότυπο νέφους ISO θα προκύψει. Τα πρότυπα οδηγούν την επιλογή και η επιλογή οδηγεί την επέκταση της αγοράς.

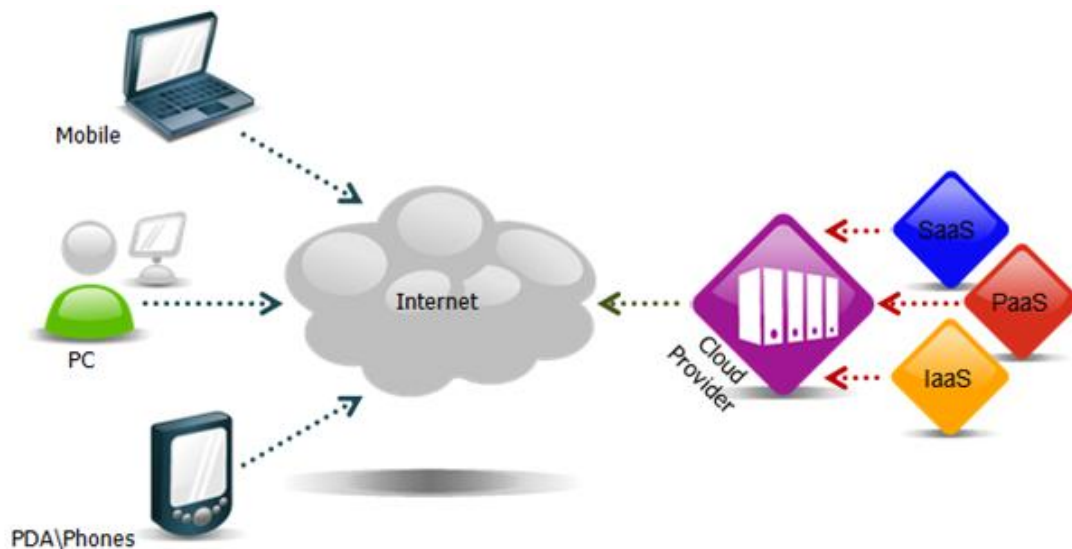
Εν συνεχεία, η κοινοπραξία ανοικτού νέφους (Open Cloud Consortium - OCC) απασχολείται ήδη με τη δια-λειτουργικότητα των διαφόρων νεφών. Η συμμαχία ασφάλειας νέφους (Cloud Security Alliance - CSA) ωθεί ενεργά τις κοινά καλύτερες πρακτικές. Το (αμφισβητούμενο) μανιφέστο ανοικτού νέφους (Open Cloud Manifesto) ωθεί τους προμηθευτές να συμφωνήσουν για τις βασικές αρχές και τη δια-λειτουργικότητα. Οι λίγοι δυνατοί υπέρ-προμηθευτές είναι οι AMD, IBM, Rackspace,

Oracle, και VMware. Αλλά οι σημαντικότεροι φορείς είναι ενάντια σε αυτή την ιδέα, συμπεριλαμβανομένων της Microsoft, της Google και της Amazon, οι οποίες σκέφτονται τη δια-λειτουργικότητα σαν το πρόωρο στάδιο που θα τους βλάψει πιο πολύ.

Επίσης, η ομάδα προτύπων οργανισμών αποκαλούμενη ως “Cloud Standards Coordination Working Group”, περιλαμβάνει διάφορους φορείς όπως τους OASIS, OMG, DMTF, SNIA, OGF, CSA, OCC, CCIF και TM Forum.

Η ομάδα αυτή εξετάζει την τυποποίηση σε διάφορους τομείς, ορισμένοι εκ των οποίων αναγράφονται παρακάτω:

- Στον τομέα της ασφάλειας,
- Στις διεπαφές στο ΥωΥ,
- Στις πληροφορίες σχετικά με την επεξεργασία όπως οι πόροι και η περιγραφή των στοιχείων,
- Στα διοικητικά πλαίσια,
- Στα πρότυπα ανταλλαγής δεδομένων,
- Στις ταξινομήσεις νέφους και
- Στα πρότυπα αναφοράς.



**Εικόνα 41 - Μέλλον του Διαδικτύου συναρτήσει του υπολογιστικού νέφους**

Ο οργανισμός ISO διαμόρφωσε μια επιτροπή για τις υπηρεσίες υπηρεσιοστρεφούς αρχιτεκτονικής (Service Oriented Architecture - SOA) και Ιστού, καθώς και μια ομάδα μελέτης για το υπολογιστικό νέφος αποκαλούμενη ως SC38, "κατανεμημένες πλατφόρμες εφαρμογής και υπηρεσίες (DAPS)" ([www.iso.org/iso/standards\\_development/technical\\_committees/other\\_bodies/iso\\_technical\\_committee.htm?commid=601355](http://www.iso.org/iso/standards_development/technical_committees/other_bodies/iso_technical_committee.htm?commid=601355)). Οι ορισμοί και τα θέματα που η επιτροπή αυτή εξετάζει έχουν ως εξής:



1. Ο τελικός χρήστης είναι ο βασικός ενδιαφερόμενος στο υπολογιστικό νέφος. Τα κεντρικά συστήματα χρηστών εμπλουτίζουν τις ζωές των ατόμων, την εκπαίδευση, την επικοινωνία, τη συνεργασία, την επιχείρηση, τη ψυχαγωγία και τη κοινωνία ως σύνολο.
2. Οι πρωτοβουλίες προώθησης της φιλικότητας προς τον χρήστη μπορούν να αυξήσουν κατά πολύ την ευημερία της ανθρωπότητας, ενεργοποιώντας ή ενισχύοντας το υπολογιστικό νέφος όπου είναι δυνατόν.
3. Το «άνοιγμα» των προτύπων, των συστημάτων και του λογισμικού ενδυναμώνει και προστατεύει τους χρήστες. Η υιοθέτηση των υπαρχόντων προτύπων όπου είναι δυνατόν, είναι προς όφελος όλων των ενδιαφερόμενων.
4. Η διαφάνεια ενθαρρύνει την εμπιστοσύνη και την υπευθυνότητα. Καθώς οι αποφάσεις πρέπει να είναι ανοικτές στο κοινό, δεν πρέπει να λαμβάνονται ποτέ κλεισμένων των θυρών.
5. Η δια-λειτουργικότητα εξασφαλίζει την αποτελεσματικότητα του υπολογιστικού νέφους ως δημόσιο πόρο. Τα συστήματα πρέπει να είναι δια-λειτουργικά πέρα από ένα ελάχιστο σύνολο καθορισμένων προτύπων και να αποφεύγεται το κλείδωμα από τον πάροχο.
6. Η αναπαράσταση όλων των ενδιαφερόμενων-χρηστών, είναι απαραίτητη. Ο πάροχος/οι δεν διαθέτει την ικανότητα να διαχειριστεί τη δια-λειτουργικότητα και τις προσπάθειες για τη δημιουργία προτύπων.
7. Η διάκριση ενάντια σε οποιοδήποτε συμβαλλόμενο μέρος για οποιοδήποτε λόγο είναι απαράδεκτη – ελαχιστοποίηση των εμποδίων εισόδου στην αγορά.
8. Η εξέλιξη είναι μια τρέχουσα διαδικασία σε μια ανώριμη αγορά. Τα πρότυπα μπορεί να πάρουν κάποιο χρόνο για να αναπτυχθούν και να συγχωνευτούν, αλλά οι δραστηριότητες πρέπει να συντονιστούν και να είναι σε συνεργασία.
9. Η ισορροπία εμπορικών και καταναλωτικών ενδιαφερόντων είναι υψίστης σημασίας. Σε περίπτωση που υπάρχει αμφιβολία, επικρατούν τα ενδιαφέροντα του καταναλωτή.
10. Η ασφάλεια είναι θεμελιώδης, και υποχρεωτικό να ενσωματώνεται. Η έλλειψη ασφάλειας δεν θα οδηγούσε στην πιθανή υιοθέτηση του νέφους.

Στα πλαίσια του οργανισμού ISO, απαιτείται πολύς χρόνος ώστε να προκύψουν τα νέα διεθνή πρότυπα. Η συμμετοχή του όμως συμβάλλει στο γεγονός ότι το νέφος είναι ένα σοβαρό και μόνιμο μέρος στο σχηματισμό Τεχνολογίας της Πληροφορίας (IT). Το πρότυπο ISO που θα προκύψει θα έχει διαδεδομένη και κυρίαρχη επιρροή στη βιομηχανία.

### **8.1.10 Η κυβερνητική πρωτοβουλία στην υιοθέτηση νεφών**

Μια άλλη παράμετρος που εξετάζεται, είναι ο ρόλος που θα διαδραματίσουν οι κυβερνήσεις στην υιοθέτηση νέφους. Το υπολογιστικό νέφος αναπτύσσεται αυτήν την περίοδο μεγάλο ρόλο στην ομοσπονδιακή κυβέρνηση των ΗΠΑ, αυτό άλλωστε φανερώνει το γεγονός ότι ο προϋπολογισμός Τεχνολογίας της Πληροφορίας (IT) της Αμερικάνικης κυβέρνησης είναι \$76 δισεκατομμύρια. Η NASA χρησιμοποιεί το

υπολογιστικό νέφος και τα κοινωνικά μέσα για να γίνει αποδοτικότερη και να μειώσει τα κόστη. Άλλες σημαντικά κυβερνητικά έργα νέφους περιλαμβάνουν τον στρατό της Αμερικής, το υπουργείο οικονομικών και πολλά άλλα.

Επιπλέον, λόγω του ότι ιστορικά η κυβέρνηση δεν υπήρξε ποτέ ηγέτης στη χρήση της τεχνολογίας πληροφοριών, αυτή τη φορά τα πράγματα δείχνουν διαφορετικά. Οι επιχειρήσεις θα βρεθούν στην ασυνήθιστη θέση να προσπαθούν να προλάβουν τις εξελίξεις που παρουσιάζουν οι κυβερνητικοί οργανισμοί στο τομέα αυτό.

## **8.2 Προβλέψεις για το μέλλον των εφαρμογών που εκτελούνται σε νέφη**

### **8.2.1 Μειωμένο κόστος ανάπτυξης εφαρμογών και ταχεία εξέλιξη στους μηχανισμούς αποθήκευσης**

Σύμφωνα λοιπόν, με τον τρόπο και τους γρήγορους ρυθμούς που εξελίσσεται το υπολογιστικό νέφος, στο εγγύς μέλλον όλοι θα είναι σε θέση να πραγματοποιούν όλες τις υπολογιστικές τους εργασίες μέσα στο νέφος. Διαφαίνεται πως το κόστος ανάπτυξης των εφαρμογών θα γίνει τόσο χαμηλό, που θα φτάσει σε σημείο να πάψει να αποτελεί εμπόδιο.

Ένα ακόμη χαρακτηριστικό το οποίο θα αλλάξει είναι αυτό του εξοπλισμού. Εν αντιθέσει με τις παραδοσιακές μεθόδους, πλέον δε θα χρειάζεται να αγοράζεται ο εξοπλισμός για την ανάπτυξη εφαρμογών. Καθώς επίσης, δε θα χρειάζεται η μίσθωση ειδικών με σκοπό την εγκατάσταση, αλλά και συντήρηση του εξοπλισμού καθ' ότι οι λειτουργίες αυτές παρέχονται από το ίδιο το νέφος. Θα αλλάξει ακόμη και η φύση της ανάπτυξης από εξωτερικούς συνεργάτες, όπως είναι για παράδειγμα οι προμηθευτές.

Επίσης, στα επόμενα χρόνια θα παρουσιαστεί μια γρήγορη εξέλιξη σε διάφορες υπηρεσίες, όπως στους μηχανισμούς αποθήκευσης, ιδιαίτερα για τα μη δομημένα δεδομένα. Ακόμη, εξέλιξη θα παρουσιαστεί και στις στρατηγικές διαβάθμισης, αλλά και στην υποστήριξη της αρχιτεκτονικής των βάσεων δεδομένων (sharding).

Με αυξανόμενο ρυθμό, θα προκύψει ένας μεγάλος αριθμός υπηρεσιών οι οποίες θα δημιουργούνται από συστήματα ΥωΥ, όπως τα Ruby on Rails, Hadoop, αλλά και από συστήματα ΠωΥ. Ένα παράδειγμα αυτών είναι η Amazon, η οποία ανέπτυξε το (Elastic Blog Storage-EBS), το εικονικό ιδιωτικό νέφος (Virtual Private cloud-VPC), μια υπηρεσία φιλοξενίας MySQL, καθώς και διάφορες προσθήκες στις ήδη υπάρχουσες υπηρεσίες της μέσα σε ένα διάστημα 12 μηνών.

### **8.2.2 Ο ρόλος των πλαισίων ανάπτυξης εφαρμογών**

Τα πλαίσια ανάπτυξης εφαρμογών διαφαίνεται ότι θα έχουν έναν σημαντικό ρόλο στην αύξηση και την εξέλιξη του υπολογιστικού νέφους. Ένας μεγάλος και αναπτυσσόμενος κατάλογος από πλαίσια ανάπτυξης εφαρμογών ανοικτού κώδικα, θα διατίθενται στο νέφος. Αυτά τα πλαίσια εφαρμογής χρησιμοποιούνται ευρέως για την οικοδόμηση μιας ευρείας γκάμας εφαρμογών Ιστού. Υπό αυτή την έννοια, διαδραματίζουν σημαντικό ρόλο στην αύξηση και την εξέλιξη του υπολογιστικού νέφους.

Τα πλαίσια ανάπτυξης εφαρμογών είναι πάντα ένας σημαντικός επιταχυντής για την ανάπτυξη, επειδή παρέχουν στα στελέχη ανάπτυξης εφαρμογών ένα μεγάλο μέρος του θεμελιώδους υλικού μιας εφαρμογής. Στις εφαρμογές που είναι βασισμένες στο νέφος, το γεγονός αυτό θα είναι πιο σημαντικό από ποτέ.

### 8.2.3 Πολύ-επίπεδο νέφος

Μια επιπρόσθετη αλλαγή που επήλθε από το υπολογιστικό νέφος, αφορά τις λειτουργίες των διαφορετικών επιπέδων. Στο υπολογιστικό νέφος συναντώνται τρία επίπεδα υπηρεσιών, αυτά της παρουσίασης, της λογικής της εφαρμογής και της αποθήκευσης.

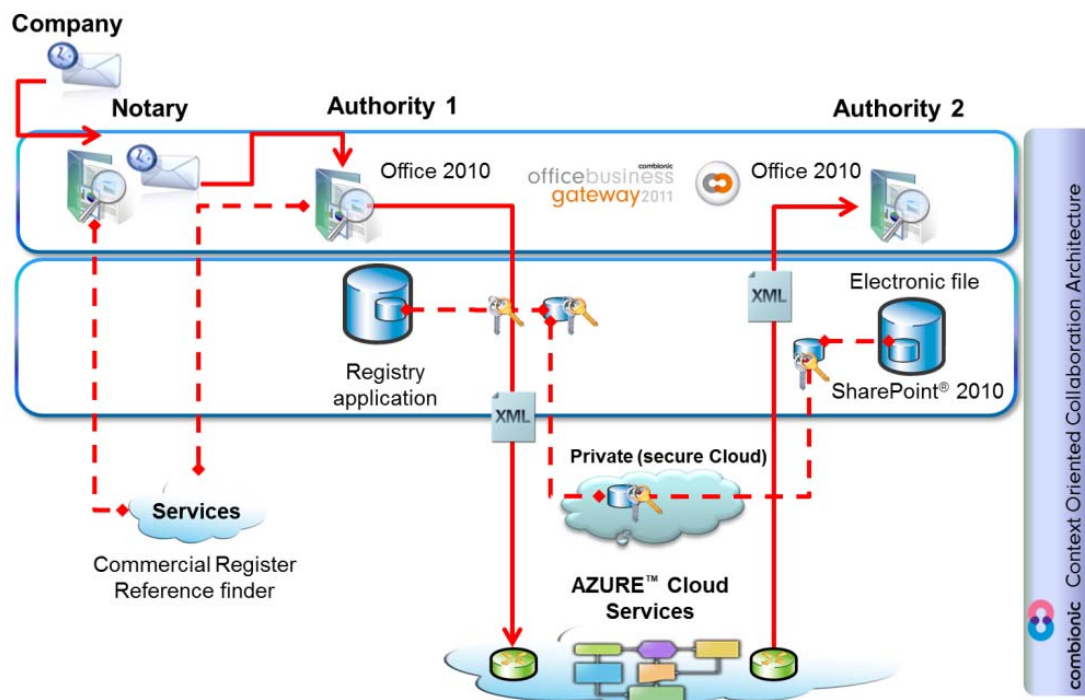
Θεωρώντας διαθέσιμο ένα παραθυρικό περιβάλλον, το πρώτο επίπεδο (παρουσίασης) είναι μια εφαρμογή πλοήγησης. Το δεύτερο, ή αλλιώς μέσο επίπεδο (λογική της εφαρμογής) εκτελείται μια μηχανή που χρησιμοποιεί κάποια τεχνολογία δυναμικού περιεχομένου Ιστού (όπως ASP, ASP.NET, CGI, ColdFusion, JSP/Java, PHP, Perl, Python, Ruby σε Rails ή Struts 2). Τέλος, το τρίτο επίπεδο (αποθήκευση) είναι μια βάση δεδομένων. Πάντως το δεύτερο καθώς και το τρίτο επίπεδο, είναι αυτά όπου θα εκτελούνται συχνότερα στο υπολογιστικό νέφος.

Οι πολύ-επίπεδες εφαρμογές υπήρξαν ως πρότυπα για πολλά έτη και διαφαίνεται ότι αυτό δεν θα αλλάξει. Αυτό γίνεται διότι, αυτός ο τύπος αρχιτεκτονικής εφαρμογής παρέχει ένα πρότυπο χαλαρής σύνδεσης για τα στελέχη ανάπτυξης εφαρμογών, ώστε να έχουν τη δυνατότητα να δημιουργούν ευέλικτες και επαναχρησιμοποιήσιμες εφαρμογές. Επίσης, διασπώντας μια εφαρμογή σε επίπεδα, τα στελέχη ανάπτυξης χρειάζεται μόνο να τροποποιήσουν ή να προσθέσουν έναν συγκεκριμένο επίπεδο, αντί να ξαναγράψουν ολόκληρη την εφαρμογή.

Εδώ πρέπει να σημειωθεί, πως οι εφαρμογές ιστού είναι εκ φύσεως πολύ-επίπεδες. Μια εφαρμογή Ιστού, είναι μια εφαρμογή που γίνεται προσβάσιμη μέσω ενός προγράμματος πλοήγησης σε ένα δίκτυο, όπως το Διαδίκτυο ή ένα ενδοδίκτυο. Ο όρος αυτός μπορεί επίσης να σημαίνει μια εφαρμογή λογισμικού υπολογιστών που φιλοξενείται σε μία εφαρμογή πλοήγησης με ελεγχόμενο τρόπο (όπως ένα applet της Java), ή ακόμη να είναι το περιβάλλον κωδικοποιημένο μέσα σε μια γλώσσα που υποστηρίζεται από την εφαρμογή πλοήγησης (όπως η JavaScript, που συνδυάζεται με μια γλώσσα σήμανσης, όπως η HTML).

Το 1995 η Netscape, ήταν αυτή που εισήγαγε τη JavaScript η οποία επέτρεπε στους προγραμματιστές να προσθέτουν δυναμικά στοιχεία στη διεπαφή χρήστη, όπου εκτελούνταν από τη πλευρά του εξυπηρετούμενου. Νωρίτερα, έπρεπε να σταλθούν όλα τα δεδομένα στον εξυπηρετή για επεξεργασία και τα αποτελέσματα παραδίδονταν μέσω των σελίδων HTML πίσω στον εξυπηρετούμενο.

Η Macromedia (που είναι πλέον μέρος του Adobe) εισήγαγε το Flash το 1996. Με τη χρήση του Flash είναι εφικτή η ανάπτυξη προηγμένων εφαρμογών ιστού που περιέχουν χαρακτηριστικά όπως ο άμεσος χειρισμός, τα κινητά μενού κ.ά..



Εικόνα 42 – Υπηρεσίες πολυ-επίπεδου νέφους

### 8.2.4 Μεγαλύτερη προστασία των ευαίσθητων δεδομένων

Η ασφάλεια στο υπολογιστικό νέφος αποτελεί ένα μεγίστης σημασίας θέμα για τους χρήστες. Τα επόμενα χρόνια, προβλέπεται πως οι υπηρεσίες ασφάλειας θα περιλαμβάνουν περισσότερες και πιο ισχυρές επιλογές με σκοπό πάντα την προστασία των ευαίσθητων δεδομένων.

Επιπροσθέτως, οι κύριοι πάροχοι υπολογιστικού νέφους, θα συνεχίσουν να ωθούν τη φυσική ασφάλεια σε πολύ υψηλά επίπεδα. Στον τομέα της ασφάλειας θα προσφέρονται από τους παρόχους τα εξής:

- παγίδες εισβολέων
- Πολλαπλά συστήματα αναγνώρισης
- Βιομετρικοί ανιχνευτές
- Υψηλή επιτήρηση για την αντιμετώπιση της κατασκοπίας.
- Razor wire (περιτύλιγμα για προστασία υλικού, συνήθως με κατοχυρωμένα πνευματικά δικαιώματα).

Ωστόσο, η ασφάλεια του δικτύου και της υποδομής, που ήδη είναι αρκετά ισχυρή, θα αυξηθεί ακόμη περισσότερο και θα επεκταθεί ώστε να ανιχνεύεται συνεχώς κάθε θύρα. Επίσης, η ανίχνευση παρείσφρησης βασισμένη στο σύστημα υπογραφής (εντοπισμός

γνωστών μοτίβων που ακολουθούν οι εισβολείς) θα εντοπίζει άμεσα οποιοσδήποτε γνωστές επιθέσεις. Ύστερα, η ανίχνευση ανωμαλίας θα εντοπίζει τις επιθέσεις σε πραγματικό χρόνο και το υπερβολικό εύρος ζώνης που δε μπορεί ποτέ να προσεγγιστεί από τις εφαρμογές, θα ανατρέψει τις επιθέσεις (Denial Of Service - DDoS).

Όσον αφορά τους παρόχους, θα κατακερματίζουν και θα κρυπτογραφούν κάθε χρονική στιγμή όλα τα αρχεία που αποθηκεύονται στο νέφος. Επίσης, θα κρατούν τα δεδομένα κρυπτογραφημένα μέχρι τη στιγμή που θα υποβάλλονται προς επεξεργασία σε μια κεντρική μονάδα επεξεργασίας (Central Processing Unit - CPU).

Αξιοσημείωτο είναι ότι τα δεδομένα αυτά, δε θα είναι σε θέση να προβεί σε αποκρυπτογράφησή τους, ούτε ο ίδιος ο πάροχος, πόσο μάλλον δε θα μπορεί και απ' την πλευρά του ένας hacker να κάνει κάτι παρεμφερές. Ακόμη, τα αποκαλούμενα «σινικά τείχη» (πρόκειται για μοντέλο ασφάλειας όπου το δικαίωμα ανάγνωσης και εγγραφής σε αρχεία καθορίζεται από τη συμμετοχή των δεδομένων σε κλάσεις και σύνολα που δομικώς είναι ελεύθερα από συγκρούσεις συμφερόντων) θα κρατούν τα δεδομένα ενός εξυπηρετούμενου του νέφους κρυφά από τους υπόλοιπους εξυπηρετούμενους. Συνεπώς, όλοι οι χρήστες πρέπει να χρησιμοποιούν επικύρωση πολλαπλών συστημάτων για τον έλεγχο πρόσβασης και επίσης να έχουν καλή βασική διαχείριση που θα εξασφαλίζει την επικύρωση και την κρυπτογράφηση όλων των προγραμματιστικών διασυνδέσεων εφαρμογών APIs (Application Programming Interface).

Επιπροσθέτως, το υλικό, τα hypervisors, καθώς και τα λειτουργικά συστήματα θα κινηθούν προς ένα σύστημα το οποίο θα παρέχει μεγάλη ασφάλεια σε όλα τα επίπεδα. Ως αποτέλεσμα αυτού θα είναι ουσιαστικά αδύνατο για ενέργειες, όπως οι επιθέσεις με υπερχειλίση ενδιάμεσης μνήμης (buffer overflow), να αποκτήσουν τον έλεγχο μιας κεντρικής μονάδας επεξεργασίας (CPU).

Τέλος, λόγω του μεγάλου αριθμού διαθέσιμων εικονικών μηχανών που προσφέρονται σε όλους τους χρήστες νέφους κάθε χρονική στιγμή, θα γίνει ρουτίνα η χρησιμοποίηση μερικών από εκείνες τις μηχανές, με σκοπό να γίνεται συχνά έλεγχος διεύθυνσης και ευπάθειας του νέφους σε επίπεδο χρηστών, όπως επίσης και σε επίπεδο παρόχου – διαχείρισης (backplane).

Η ασφάλεια λοιπόν στο νέφος, αποτελεί ένα μείζονος σημασίας θέμα, το οποίο πλέον παίρνει τέτοια μεγάλη διάσταση, όπου μόνο σε μερικά έτη θα καθορίσει τα πρότυπα για την εξασφάλιση των πληροφοριών. Όλων των ειδών τα δεδομένα, θα είναι ιδιωτικά και εξασφαλισμένα εκτός από την ακριβή στιγμή λειτουργίας της κεντρικής μονάδας επεξεργασίας (CPU), καθιστώντας έτσι τα νέφη τις ασφαλέστερες θέσεις για να εργαστεί κανείς.

### **8.2.5 Υψηλότερου επιπέδου υπηρεσίες με μοναδικά APIs**

Κατά τη διάρκεια της δεκαετίας που έπεται, οι επιχειρήσεις με τις πολύτιμες αποθήκες δεδομένων θα προσφέρουν υπηρεσίες υψηλότερου επιπέδου στα υπάρχοντα νέφη, κάθε ένα με μια μοναδική προγραμματιστική διασύνδεση εφαρμογών. Όσον αφορά τις βάσεις δεδομένων αδειών οδήγησης, τις ποικίλες βάσεις δεδομένων που επικυρώνουν τη ταυτότητα των καταναλωτών, την ακίνητη περιουσία, τη χαρτογράφηση, τις βάσεις δεδομένων προϊόντων, όπως και πολλούς άλλους τύπους δεδομένων, θα έχουν σύντομα ισχυρές προγραμματιστικές διασυνδέσεις εφαρμογών σε μια υπηρεσία φιλοξενίας

νέφους. Όλες αυτές οι πολύτιμες αποθήκες δεδομένων, θα είναι εύκολα προσβάσιμες στα στελέχη ανάπτυξης εφαρμογών και στους καταναλωτές.

Αυτό το πρότυπο θα λειτουργήσει επίσης εξαιρετικά για τις απαιτητικές υπολογιστικές εργασίες, όπως για παράδειγμα η χαρτογράφηση, οι μετασχηματισμοί, η επεξεργασία βίντεο και φωτογραφιών, καθώς και για πολλούς ακόμη επιστημονικούς υπολογισμούς.

Όταν φιλοξενείται μια ικανότητα/δυνατότητα στο νέφος με μια προγραμματιστική διασύνδεση εφαρμογών, προκειμένου να γίνεται προσβάσιμη από τους χρήστες, μετατρέπεται σε ένα συστατικό που μπορεί να «καταναλωθεί», με την ενσωμάτωσή του σε έναν νέο τύπο εφαρμογής. Αυτό έχει σαν αποτέλεσμα, να αλλάζει τον τρόπο με τον οποίο όλοι σκέπτονται για τις εφαρμογές. Αυτή η προσέγγιση καλείται “mashup” ή ακόμη και σύνθετη εφαρμογή.

### 8.2.6 Υιοθέτηση και αύξηση των mashups

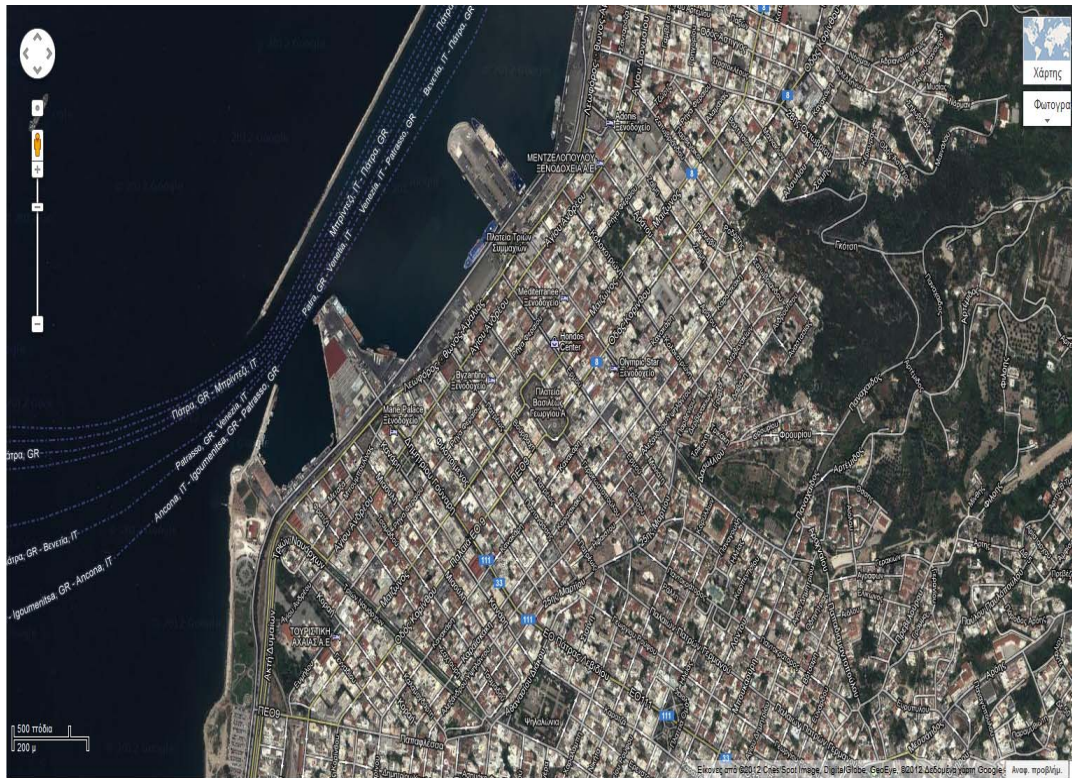
Αναφορικά με την υιοθέτηση και αύξηση των mashups, οι αναλυτές προβλέπουν ότι η χρήση τους θα αυξηθεί ιδιαίτερα τα επόμενα πέντε χρόνια. Όπως το υπολογιστικό νέφος αντιπροσωπεύει έναν μετασχηματισμό πλατφορμών, οι οποίες θα προκαλέσουν θεμελιώδεις αλλαγές στην οικοδόμηση της υποδομής τεχνολογίας πληροφοριών, έτσι και τα mashups αντιπροσωπεύουν μια ριζική αλλαγή στη κατασκευή της εφαρμογής.

Στα πλαίσια της ανάπτυξης εφαρμογών ιστού, ένα mashup είναι μια εφαρμογή ιστοσελίδας ή μια λειτουργία η οποία συνδυάζει δεδομένα από δύο ή περισσότερες εξωτερικές πηγές, με σκοπό τη δημιουργία μιας νέας υπηρεσίας. Επίσης, ένα ακόμη χαρακτηριστικό του mashup είναι πως βασίζεται στις τεχνολογίες του web 2.0, όπως και τα κοινωνικά δίκτυα. Ιδιαίτερα σημαντικό ρόλο κατά τη δημιουργία διάφορων mashups, έχουν οι προγραμματιστές διασυνδέσεις εφαρμογών (Application Programming Interface, API). Χρησιμοποιώντας τις προγραμματιστικές διασυνδέσεις εφαρμογών, επιτυγχάνεται γρήγορη και εύκολη πρόσβαση σε διάφορες πηγές δεδομένων αλλά και υπηρεσιών, με σκοπό τη δημιουργία νέων λειτουργιών.

Αρκετοί άνθρωποι θα συνειδητοποιήσουν, ότι διαθέτουν τη δυνατότητα να έχουν πρόσβαση σε μεγάλες ποσότητες δεδομένων, καθώς και σε υψηλού επιπέδου τεχνολογίες, όπου ποτέ πριν δεν μπορούσαν. Επίσης, εύκολα θα μπορούν να αναπτύξουν καινοτόμες εφαρμογές και να τις φιλοξενήσουν στο νέφος. Η ταχεία υιοθέτηση και ανάπτυξη των mashups, θα τροφοδοτήσει την περαιτέρω ανάπτυξη του νέφους, καθώς γίνεται εφικτή η χρήση τους μόνο λόγω αυτού και το αντίστροφο.

Ένα παράδειγμα mashup, αποτελεί η χρήση των χαρτογραφικών δεδομένων για τη προσθήκη πληροφοριών θέσης στα δεδομένα ακινήτων, όπου προβαίνει στη δημιουργία ενός νέου και ευδιάκριτου API ιστού, όπου δεν παρέχεται αρχικά από καμία άλλη πηγή. Οι χάρτες της Google (Google Maps) είναι πιθανόν το πιο κοινό στοιχείο που χρησιμοποιείται στα mashups, με όλα τα είδη διαφορετικών δεδομένων που συγχωνεύονται για να δημιουργήσουν ένα απλό αλλά χρήσιμο είδος mashup, όπως απεικονίζονται στην Εικόνα 43.





**Εικόνα 43 - Google Maps**

Βέβαια για τα mashups, δεν υπάρχει ακόμα κάποιο επίσημο πρότυπο. Τυπικά, βασίζονται στις αρχές αρχιτεκτονικής του REST για τη δημιουργία, την ανάγνωση, την αναβάθμιση και τη διαγραφή λειτουργιών σε κάποια αποθετήρια (repositories). Τα βασικά πρότυπα στα οποία βασίζονται τα mashups περιλαμβάνουν την γλώσσα XML (Extensible Markup Language), η οποία εναλλάσσεται όπως το REST ή οι υπηρεσίες Ιστού.

Τέλος, διάφορες ομάδες έχουν διαμορφώσει ήδη κάποια πρότυπα mashup. Η ομάδα Open Mashup Alliance (OMA), έχει προτείνει την επιχειρηματική γλώσσα σήμανσης Mashup (EMML). Μια προτεινόμενη γλώσσα διατύπωσης ερωτημάτων αποκαλούμενη ως MashQL, είναι για τα ερωτήματα έναντι των μοντέλων μεταδεδομένων.

### **8.2.7 Εξέλιξη των εργαλείων ανάπτυξης για τη κατασκευή mashups**

Προβλέπεται μελλοντικά, ότι τα ισχυρότερα εργαλεία ανάπτυξης θα εξελιχτούν αρκετά γρήγορα έχοντας σαν σκοπό να καταστήσουν εύκολη την κατασκευή mashups. Ειδικότερα, λόγω των πολλών απαιτήσεων, που οι τελικοί χρήστες έχουν προκειμένου να κατασκευάσουν τις δικές τους εφαρμογές, θα εξελιχθεί μια κατηγορία εργαλείων η οποία θα επιτρέπει κάτι τέτοιο.

Γενικότερα, ένας «παραδοσιακός» προγραμματιστής θα δημιουργήσει ένα εργαλείο προσανατολισμένο σε ένα τομέα μιας συγκεκριμένης εφαρμογής. Ο τομέας μπορεί να είναι κάποια πτυχή μιας επιχειρησιακής διαδικασίας ή ενός κοινωνικού σχεδιασμού ή κάτι το ενδιάμεσο. Κατόπιν, ένας μη-προγραμματιστής (π.χ. ο τελικός χρήστης) θα είναι σε θέση να χρησιμοποιήσει αποτελεσματικά αυτό το εργαλείο. Με βάση το τι παρέχεται από τον προγραμματιστή, ο τελικός χρήστης θα μπορεί να εισάγει μια νέα εφαρμογή με

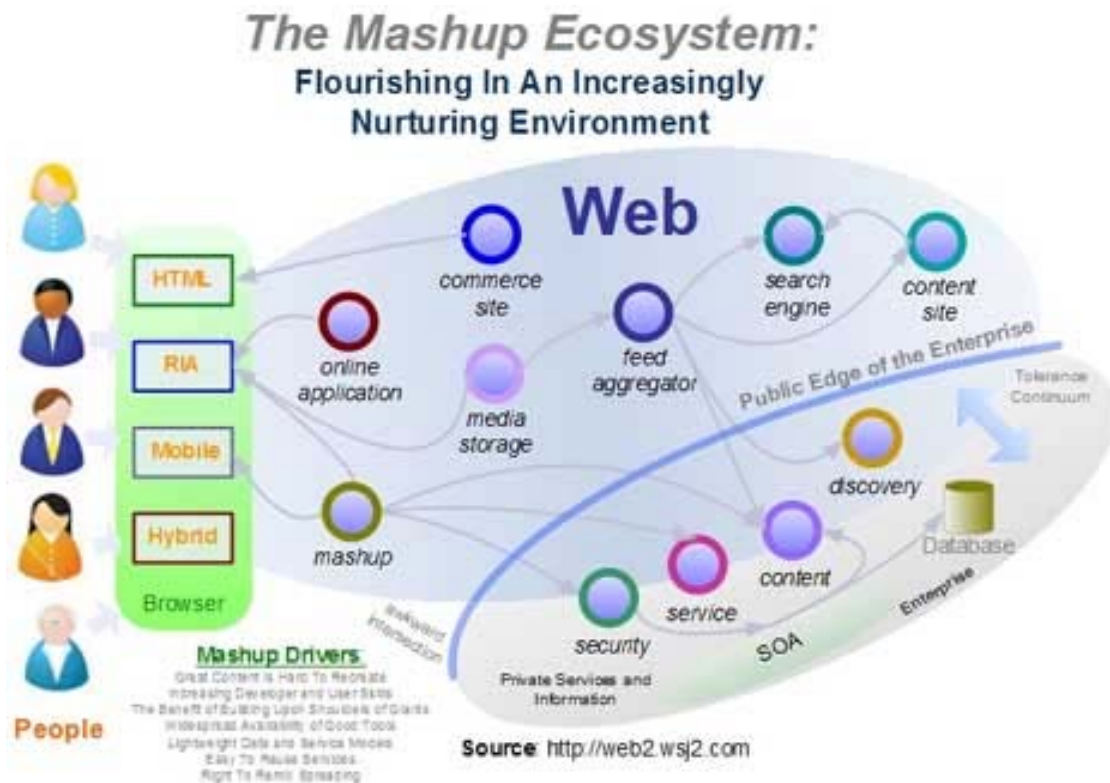
σκοπό τη δημιουργία μιας εφαρμογής όπου ο ίδιος και οι συνεργάτες/φίλοι του επιθυμούν να χρησιμοποιήσουν. Επειδή αυτά είναι καλά κατασκευασμένα στοιχεία εφαρμογής, και επειδή το εργαλείο για τη δημιουργία μιας τελικής εφαρμογής θα είναι προσεκτικό σχετικά με το τι επιτρέπει στους χρήστες να κάνουν, η κατάληξη θα είναι μια ισχυρή εφαρμογή.

Για το λόγο ότι το εργαλείο το οποίο χρησιμοποιείται για τη δημιουργία μιας τελικής εφαρμογής, θα είναι προσεκτικό σχετικά με το ποιες ενέργειες επιτρέπει στους χρήστες να κάνουν, κατάληξη αυτού θα είναι μια ισχυρή εφαρμογή. Η εφαρμογή αυτή, καθώς και τα επιμέρους τμήματά της, θα εκτελούνται στο υπολογιστικό νέφος. Αυτό θα επιτρέψει σε εργαλεία, εφαρμογές αλλά και σε στοιχεία, να χρησιμοποιούνται σε συγκεκριμένες εφαρμογές, οι οποίες χρησιμοποιούν την επίδραση του δικτύου για να διαδοθεί τι είναι καλό και τι δεν είναι. Με τον τρόπο αυτό θα δημιουργηθούν χιλιάδες άπλες εφαρμογές. Κάτι τέτοιο άλλωστε γίνεται εύκολα αντιληπτό, παρατηρώντας τις εφαρμογές στο iPhone και στο Facebook. Αυτό που στην ουσία αλλάζει είναι η ατομική φύση δημιουργίας της εφαρμογής.

Αυτή τη στιγμή υπάρχουν περισσότερα από 2.000 στοιχεία mashup στο ProgrammableWeb.com (<http://www.programmableweb.com/>), τα οποία χωρίζονται σε διάφορες κατηγορίες. Ο αριθμός των κατηγοριών αλλά και τα mashup API που ανήκουν στη κάθε κατηγορία, συνεχώς αυξάνεται. Το γεγονός αυτό, σε συνδυασμό με την ανάπτυξη ισχυρών εργαλείων τα οποία είναι απλά στη χρήση για τη κατασκευή των Mashups, αλλάζει ολόκληρη τη φιλοσοφία της ανάπτυξης εφαρμογών.

Τέλος, αξίζει να αναφερθεί σαν γεγονός ότι τα στελέχη ανάπτυξης εκτός των ΗΠΑ και της Ευρώπης, θα ξεπεράσουν τη δύση επειδή δεν επιβαρύνονται με τη κληρονομιά των υποδομών τεχνολογίας πληροφοριών. Θα κατασκευάσουν και θα πωλήσουν στοιχεία mashup, καθώς και εργαλεία για την ανάπτυξή τους, ή θα κατασκευάσουν περιπλοκότερες εφαρμογές οι οποίες θα εκτελούνται στο νέφος. Αυτό μπορούν να το επιτύχουν με μικρότερο κόστος, επειδή δεν είναι απαραίτητο να αγοράσουν την ακριβή υποδομή τεχνολογίας πληροφοριών.





**Εικόνα 44 – Mashup σύστημα**

### 8.2.8 Η χρήση των ΠωΥ και FaaS ως κυρίαρχα εργαλεία

Στα χρόνια που έπονται, φαίνεται ότι θα κυριαρχήσει ένας νέος τρόπος για την κατασκευή εφαρμογών. Ο τρόπος αυτός είναι το πλαίσιο ως υπηρεσία (Framework as a Service - FaaS), στο οποίο θα εξελιχθούν τόσο το AppEngine όσο και το Force.com και η πλατφόρμα ως υπηρεσία (ΠωΥ) με την προσθήκη των εξής χαρακτηριστικών:

- Ισχυρά εργαλεία κατασκευής mashup,
- Προγραμματιστές διασυνδέσεις εφαρμογών (APIs),
- Χορήγηση αδειών, συσκευασίας, τιμολογιακών και βοηθητικών εργαλείων καθώς και
- Υπηρεσίες με στόχο στην οικοδόμηση mashups.

Το σύστημα που θα εξελιχθεί για να υποστηρίξει τα mashups, θα περιλαμβάνει ένα στοιχείο ανταλλαγής mashup. Αυτό θα είναι μια θέση στο διαδίκτυο προσβάσιμη ως εφαρμογή ΛωΥ που θα επιτρέπει στα στελέχη ανάπτυξης εφαρμογών να βρουν παρόμοια στοιχεία με εκείνα των εφαρμογών iPhone. Αυτά τα στοιχεία δεν θα είναι αυτόνομες εφαρμογές: αντ' αυτού θα είναι συνιστώσες που με την προσθήκη επιπλέον στοιχείων θα μπορούν να δημιουργούν mashups. Η ανταλλαγή αυτή θα επιτρέπει όχι μόνο στα στοιχεία να επικοινωνήσουν αλλά θα παρέχει επίσης και τα συστατικά χορήγησης αδειών. Επιπρόσθετα, τα στοιχεία αυτά θα εισάγουν ένα στρώμα τιμολόγησης έτσι ώστε οι πάροχοι υπηρεσιών να έχουν έναν τρόπο να κερδίζουν από τη χρήση τους.

## Αναφορές

1. Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, 2011.
2. Jericho Forum - Position Paper, Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration, April 2009.
3. U.S. Federal Cloud Computing Initiative, July 2009 (<http://www.scribd.com/doc/17914883/US-Federal-Cloud-Computing-Initiative-RFQ-GSA>)
4. Peter Mell και Tim Grance, The NIST Definition of Cloud Computing, NIST, September 2011.
5. Gautam Shroff, Enterprise Cloud Computing: Technology, Architecture, Applications, Cambridge University Press, 2010.
6. George Reese, Cloud Application Architectures, O'Reilly, 2009.
7. Lamia Youseff, Maria Butrico, Dilma Da Silva, Toward a Unified Ontology of Cloud Computing, GCE, 2008.
8. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "Above the Clouds: A Berkley View on Cloud Computing", Electrical Engineering and Computer Sciences University of California at Berkeley, 2009
9. Jothy Rosenberg and Arthur Mateos, The Cloud at your Service – The when, how, and why of enterprise cloud computing, Manning Publications Co., 2011.
10. Ashish Sarin, Portlets in Action, Manning Publications Co., pp. 640, September, 2011.