



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΩΝ**

**ΑΣΦΑΛΕΙΣ ΔΟΜΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ
ΜΙΚΡΟΥΠΗΡΕΣΙΩΝ ΓΙΑ ΜΟΝΤΕΛΑ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΒΡΕΤΤΟΣ Π. ΜΟΥΛΟΣ

Διπλωματούχος Μηχανικός Πληροφοριακών & Επικοινωνιακών Συστημάτων

Πανεπιστημίου Αιγαίου

ΕΠΙΒΛΕΠΟΥΣΑ:

ΘΕΟΔΩΡΑ. ΒΑΡΒΑΡΙΓΟΥ

Καθηγήτρια Ε.Μ.Π.

ΑΘΗΝΑ, ΝΟΕΜΒΡΙΟΣ 2019



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΩΝ

ΑΣΦΑΛΕΙΣ ΔΟΜΕΣ ΑΡΧΙΤΕΚΤΟΝΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ
ΜΙΚΡΟΥΠΗΡΕΣΙΩΝ ΓΙΑ ΜΟΝΤΕΛΑ ΠΟΛΛΩΝ ΔΕΔΟΜΕΝΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΒΡΕΤΤΟΣ Π. ΜΟΥΛΟΣ

Διπλωματούχος Μηχανικός Πληροφοριακών & Επικοινωνιακών Συστημάτων
Πανεπιστημίου Αιγαίου

Συμβουλευτική Επιτροπή:

Θ. ΒΑΡΒΑΡΙΓΟΥ, Καθ. Ε.Μ.Π. (Επιβλέπουσα)

Π. ΤΣΑΝΑΚΑΣ, Καθ. Ε.Μ.Π.

Σ. ΠΑΠΑΒΑΣΙΛΕΙΟΥ, Καθ. Ε.Μ.Π.

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή

.....
Θ. Βαρβαρίγου
Καθ. ΕΜΠ

.....
Π. Τσανάκας
Καθ. ΕΜΠ

.....
Σ. Παπαβασιλείου
Καθ. ΕΜΠ

.....
Ε. Βαρβαρίγος
Καθ. ΕΜΠ

.....
Α. Δουλάμης
Καθ. ΕΜΠ

.....
Δ. Ασκούνης
Καθ. ΕΜΠ

.....
Ι. Ψαρράς
Καθ. ΕΜΠ

ΑΘΗΝΑ, ΝΟΕΜΒΡΙΟΣ 2019

.....
Βρεττός Π. Μουλός

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

Copyright © Βρεττός Π. Μουλός, 2019.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Η έγκριση της διδακτορικής διατριβής από την Ανώτατη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ. Πολυτεχνείου δεν υποδηλώνει αποδοχή των γνώμων του συγγραφέα (Ν. 5343/1932, Άρθρο 202).

Περίληψη

Η παρούσα διδακτορική διατριβή ασχολείται με την ανάπτυξη τεχνολογιών συγκέντρωσης, επεξεργασίας και ανάλυσης δεδομένων από ροές είτε από μεγάλα αποθετήρια δεδομένων ακολουθώντας το μοντέλο των μικροπηρεσιών σε δομικό στοιχείο. Βασικό αντικείμενο της διατριβής είναι η παρουσίαση και η ανάλυση του πλαισίου εκείνου που μπορεί να δώσει στον αρχιτέκτονα των ψηφιακών συστημάτων τη δυνατότητα να μετατρέψει είτε να δομήσει το υπό υλοποίηση σύστημα με τέτοιο τρόπο ώστε να μπορεί να ανταπεξέλθει στις προκλήσεις των πολλών δεδομένων.

Η είσοδος στην ανάπτυξη εφαρμογών με τις αρχές των μικροπηρεσιών έχει προσφέρει πλεονεκτήματα τόσο όσο αναφορά στην καλύτερη λογική διαχείριση των υπηρεσιών όσο και στη συντήρηση των εν λόγω συστημάτων. Η όλο και μεγαλύτερη χρήση του cloud computing προσέφερε τα απαραίτητα εχέγγυα για να χτιστεί πάνω σε αυτό μια νέα αρχιτεκτονική που συνδυάζει την υπηρεσιοστρέφια με την ευελιξία διαχείρισης των υπηρεσιών. Ο νέος τρόπος δομής βασίζεται όλο και περισσότερο στο συνδυασμό έτοιμων ‘από το ντουλάπι’ υπηρεσιών που χτίζουν τη νέα υπηρεσία που καλείται να υλοποιηθεί.

Μοιραία όμως αυτή η προσέγγιση δημιουργεί νέα ερωτήματα που καλούμαστε να απαντήσουμε. Είναι αναγκαίο να παρουσιαστούν συστήματα που θα αναλαμβάνουν την ενορχήστρωση όλων αυτών των μικροπηρεσιών καθώς και την εισαγωγή προτύπων που δομούν αυστηρά αυτές τις υπηρεσίες ώστε να μπορούν να επικοινωνήσουν ορθά μεταξύ τους. Τέλος αυτά τα θέματα συμβατότητας στο κομμάτι του ενιαίου καναλιού επικοινωνίας όσο και στο τρόπο που αλληλοεπιδρούν πρέπει να καλύπτονται από το μανδύα της ασφάλειας. Παράγοντας ιδιαίτερα σημαντικός μιας και κάθε πλατφόρμα προσφοράς υπηρεσιών προσφέρει συνήθως δικό της τρόπο προσέγγισης στο κομμάτι της υποδομής.

Στην παρούσα διατριβή, εισάγεται μια μεθοδολογία που βασίζεται σε πιστοποιημένες μεθοδολογίες οργανισμών ασφάλειας, εμπλουτισμένη και τροποποιημένη ώστε να μπορεί να εφαρμοστεί στο περιβάλλον της πληροφορικής. Η μεθοδολογία πρέπει να ακολουθεί τη ροή της πληροφορίας ώστε να μπορεί να προσφέρει το καλύτερο έλεγχο των δεδομένων αλλά και για να καταφέρει να ανταποκριθεί στις αυξημένες απαιτήσεις των υποσυστημάτων. Ταυτόχρονα αναλύεται το μοντέλο που πρέπει να ακολουθηθεί για την εύρεσης μικροπηρεσιών μέσα από ενιαίες αγορές (marketplaces) όπου θα γίνεται η δημοσίευση των προϊόντων-υπηρεσιών. Ο συνδυασμός αυτών των υπηρεσιών θα πρέπει να έχει μια στρατηγική ώστε να μπορεί να ανταπεξέλθει στις προκλήσεις του φέρνουν τα πολλά δεδομένα. Μια τέτοια ανάλυση παρουσιάζεται και αναλύεται διεξοδικά ώστε να γίνουν αντιληπτές οι νέες τεχνικές που πρέπει να ακολουθηθούν. Τέλος παρουσιάζεται ένα σχεδιάγραμμα (Blueprint) που αποτελεί το δομικό κομμάτι για το κάθε προσφερόμενο

προϊόν ώστε να μπορεί παρουσιάσει καλύτερα τις υπηρεσίες που προσφέρει με ένα συντακτικά και δομικά ενιαίο τρόπο. Η ανάπτυξη αυτού του σχεδιαγράμματος εκτός από θέμα διαλειτουργικότητας των υπηρεσιών που μπορεί να προσφέρει, είναι απαραίτητο για τις νέες προκλήσεις ασφάλειας που δημιουργούνται τόσο όσο αναφορά τα δεδομένα όσο και στις υπηρεσίες αυτές καθ' αυτές.

Λέξεις Κλειδιά: ασφάλεια μικροπηρεσιών, υπηρεσιοστρεφής αρχιτεκτονικές, βελτιστοποίηση αλγορίθμων διαχείρισης πολλών δεδομένων, πολλά δεδομένα.

Abstract

This doctoral thesis deals with the development of technologies for the collection, processing and analysis of data from streams or from large data repositories following the small-service model as a building block. The main purpose of this thesis is to present and analyze the framework that can give the digital systems architect the ability to transform or structure the system that has been implemented so that it can meet the challenges of multiple data sets.

The introduction of application development with mini-services has offered advantages both in terms of better rational management of services and maintenance of these systems. The increasing use of cloud computing has provided the necessary credentials to build on it a new architecture that combines service-oriented with service management flexibility. The new way of building is increasingly based on a combination of ready-to-use 'closet' services that build the new service called for.

But this approach raises new questions that we are called upon to answer. They need to have systems in place to orchestrate all of these mini-services as well as introduce standards that strictly structure these services so they can communicate properly with each other. Finally, these issues of compatibility in the single channel of communication and in the way they interact must be covered by the mantle of security. An important factor as each service platform usually offers its own way of approaching the infrastructure.

In this thesis, we introduce a methodology based on accredited security organization methodologies, enriched and modified so that it can be applied in the IT environment. The methodology must follow the flow of information in order to be able to provide better control over the data but also to meet the increased demands of the subsystems. At the same time, we analyze the model to be followed for finding micro services through marketplaces where service products will be published. Combining these services should have a strategy to meet the challenges of bringing in a lot of data. Such an analysis is presented and analyzed in detail to understand the new techniques to be followed. Finally, a blueprint is presented that is the building block for each product offered so that it can better present its services in a structured way. The development of this layout, in addition to the interoperability of the services it can provide, is necessary for the new security challenges that arise both in reference to the data and in the services themselves.

Keywords: Microservices security, Service Oriented Architectures, Data design optimization algorithms, Big Data.

Πρόλογος

Η διδακτορική διατριβή που παρουσιάζεται στις επόμενες σελίδες εκπονήθηκε από τον Οκτώβριο του 2009 μέχρι τον Νοέμβριο του 2019, στο εργαστήριο Τηλεπικοινωνιών του τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής, στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου υπό την επίβλεψη της κ. Θεοδώρας Βαρβαρίγου.

Θα ήθελα να ευχαριστήσω ιδιαίτερα την καθηγήτριά μου κ. Θεοδώρα Βαρβαρίγου για την υποστήριξη, και την καθοδήγηση που μου παρείχε από την αρχή ως το τέλος της προσπάθειάς μου, καθώς επίσης τους καθηγητές της τριμελούς συμβουλευτικής επιτροπής κ. κ. Παναγιώτη Τσανάκα και Συμεών Παπαβασιλείου.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους συναδέλφους με τους οποίους συνεργάστηκα κατά την διάρκεια της εκπόνησης της διατριβής μου. Ιδιαίτερες ευχαριστίες ωστόσο θα ήθελα να απευθύνω στους συναδέλφους και φίλους Μάνια Καρδαρά και Παύλο Κρανά.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου τους φίλους μου και τη πρώην γυναίκα μου για την στήριξη τους όλα αυτά τα χρόνια.

Βρεττός Μουλός

Νοέμβριος 2019

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΤΙΤΛΟΣ	ii
ΣΧΗΜΑΤΑ – ΕΙΚΟΝΕΣ	XVII
ΠΙΝΑΚΕΣ.....	XX
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	1
1.1 Οργάνωση του εγγράφου.....	6
ΚΕΦΑΛΑΙΟ 2 ΑΠΟΘΕΤΗΡΙΑ ΔΕΔΟΜΕΝΩΝ.....	9
2.1 Μοντέλα αρχιτεκτονικών δομών διαχείρισης δεδομένων.....	12
2.2 Συστήματα βάσεων δεδομένων.....	14
2.2.1 Εξωτερικές πηγές δεδομένων.....	17
2.3 Μεθοδολογία κύκλου ζωής πληροφορίας	18
ΚΕΦΑΛΑΙΟ 3 ΑΓΟΡΕΣ ΥΠΗΡΕΣΙΟΣΤΡΕΦΩΝ ΕΦΑΡΜΟΓΩΝ	33
3.1 Αγορές με υπηρεσίες εγκατεστημένες στο νέφος.....	34
3.2 Αρχές πρότυπου σχεδιασμού	47
3.3 Μοντέλα κέρδους και διαχείρισης τιμών	50

3.3.1	Μοντελοποίηση οικονομικού μοντέλου	51
3.4	Εύρεση υπηρεσιών, επιλογή και επίλυση τους.....	52
3.5	Αρχές σχεδίασης πλατφόρμας	62
3.6	Αρχιτεκτονική ενδιάμεσου λογισμικού προσανατολισμένη σε υπηρεσιοστρεφείς αρχιτεκτονικές.....	67
3.7	Αποτελέσματα ανάλυσης.....	74
3.8	Σύνοψη.....	77
ΚΕΦΑΛΑΙΟ 4 ΣΥΝΘΕΤΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΥΜΒΑΝΤΩΝ		81
4.1	Εισαγωγή	82
4.2	Σύνθετη Επεξεργασία Συμβάντων	83
4.3	Αρχιτεκτονική	86
4.4	Παραλληλοποίηση του Complex Event Processing.....	89
4.5	Ροές δεδομένων.....	91
4.6	Τελεστές Γεγονότων.....	93
4.6.1	Τελεστής Map.....	93
4.6.2	Τελεστής Filter	94

4.6.3	Τελεστής Demux	95
4.6.4	Τελεστής Union	96
4.7	Τελεστές προσαρμοσμένοι με χρήση παραθύρου.....	97
4.7.1	Τελεστής Aggregate.....	98
4.7.2	Τελεστής Join	100
4.8	Τελεστές διαχείρισης συστημάτων αποθήκευσης.....	102
4.8.1	Τελεστής Insert.....	102
4.8.2	Τελεστής Update.....	103
4.8.3	Τελεστής Delete.....	104
4.8.4	Τελεστής Select	104
4.9	Τελεστής προσαρμοσμένοι από τους χρήστες.....	105
4.10	Στρατηγικές Παραλληλοποίησης	105
4.11	Ελαστικότητα	109
4.12	Διασυνδεδεμένο σύστημα ανάλυσης πολλών δεδομένων	112
4.13	Συγκριτική μελέτη	126

4.14	Σύνοψη	133
ΚΕΦΑΛΑΙΟ 5 ΜΙΚΡΟΥΠΗΡΕΣΙΕΣ		135
5.1	Εισαγωγή	136
5.2	Αρχιτεκτονική Προσανατολισμένη στην Υπηρεσία (SOA).....	136
5.3	Εταιρικό τοπίο διακυβέρνησης	138
5.4	Χαρακτηριστικά της SOA	139
5.4.1	Περιορισμοί στη SOA.....	140
5.4.2	Ελλείψεις στη SOA.....	144
5.4.3	Πως οι μικρουπηρεσίες επεκτείνουν τη SOA	145
5.4.4	Πλεονεκτήματα των μικρουπηρεσιών	147
5.4.5	Προκλήσεις.....	150
5.5	Blueprint	153
5.5.1	Ενορχήστρωση υπηρεσιών.....	157
5.5.2	Ενοποιημένο πρότυπο μικρουπηρεσιών	161
5.6	Συνοψη.....	186

ΚΕΦΑΛΑΙΟ 6 ΕΠΙΛΟΓΟΣ.....	190
ΚΕΦΑΛΑΙΟ 7 ΒΙΒΛΙΟΓΡΑΦΙΑ.....	196

ΣΧΗΜΑΤΑ – ΕΙΚΟΝΕΣ

Εικόνα 1 Κύκλου ζωής πληροφοριών	25
Εικόνα 2 Τύποι πληροφοριών που δημιουργούν τον Ωκεανό των δεδομένων.....	29
Εικόνα 3 Διαφορετικές μορφές αποθετηρίων δεδομένων	31
Εικόνα 4 Κύκλος ζωής υπηρεσιών νέφους.....	44
Εικόνα 5 Κύριες φάσεις και διεργασίες των ηλεκτρονικών αγορών	48
Εικόνα 6 Έξυπνες δομές υπολογιστικού νέφους από διαφορετικούς πάροχους υπηρεσιών	64
Εικόνα 7 Κατανομή υπηρεσιών στα επίπεδα του Υπολογιστικού νέφους.....	65
Εικόνα 8 Αρχιτεκτονική παρουσίαση επιπέδων	69
Εικόνα 9 Διαχείριση τρόπου εγκατάστασης cloud εφαρμογών.....	71
Εικόνα 10 Πρότυπη αναγγελία της Request Transformer	75
Εικόνα 11 Δενδροειδής υπολογισμός πόρων.....	77
Εικόνα 12 Complex event processing	87
Εικόνα 13 Αρχιτεκτονική για παράλληλη διαχείριση του CEP.....	90
Εικόνα 14 Παράδειγμα του Τελεστή Map.....	94

Εικόνα 15 Παράδειγμα του τελεστή Filter	95
Εικόνα 16 Παράδειγμα του τελεστή Demux	96
Εικόνα 17 Παράδειγμα του τελεστή Union	97
Εικόνα 18 Παράδειγμα του τελεστή Aggregate.....	99
Εικόνα 19 Παράδειγμα του τελεστή Join	102
Εικόνα 20 Στρατηγική Παραλληλοποίησης	107
Εικόνα 21 Σύγκριση και ανάλυση γράφων με βάση δείκτες ομοιότητας.....	116
Εικόνα 22 Γενική αρχιτεκτονική συστήματος διαχείρισης πολλών δεδομένων για ανάλυση μικρών κειμένων.....	122
Εικόνα 23 Σύγκριση και ανάλυση γράφων βάση της λίστας των χαρακτηριστικών	125
Εικόνα 24 Ακρίβεια αποτελεσμάτων ανάλογα με το μέγεθος των δεδομένων εκπαίδευσης	128
Εικόνα 25 Απόδοση συστήματος με χρήση πολλών IM	130
Εικόνα 26 Συνολικός όγκος επεξεργαζόμενων tweets ανά μονάδα χρόνου.....	131
Εικόνα 27 Χρόνος επεξεργασίας tweets ανάλογα με τον αριθμό των IM.....	132
Εικόνα 28 Διάγραμμα υπηρεσιών	159

Εικόνα 29 Πρότυπο Blueprint.....	165
Εικόνα 30 Δομή JSON σχήματος του πρώτου τμήματος.....	169
Εικόνα 31 Γενική δομή JSON σχήματος του δεύτερου τμήματος.....	173
Εικόνα 32 Εσωτερική δομή του σχήματος σχετικά με τα μοντέλα κόστους.....	174
Εικόνα 33 Δομή JSON σχήματος του τρίτου τμήματος.....	180
Εικόνα 34 Δομή JSON σχήματος του τέταρτου τμήματος.....	185

ΠΙΝΑΚΕΣ

Πίνακας 1 Σύγκριση ηλεκτρονικών αγορών.....	40
Πίνακας 2 Περιοχές τομέα για προκλήσεις της SOA.....	141
Πίνακας 3 Επιχειρηματικά πλεονεκτήματα	148
Πίνακας 4 Πρώτο τμήμα του Blueprint.....	167
Πίνακας 5 Δεύτερο τμήμα του Blueprint.....	172
Πίνακας 6 Τρίτο τμήμα του Blueprint.....	176
Πίνακας 7 Τέταρτο τμήμα του Blueprint.....	184

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Συντηρητικές μελέτες εκτιμούν ότι το 2013 επεξεργαζόμασταν 4.4 zetabytes δεδομένων [1], αριθμός που όμως δε συγκρίνεται με την εκθετική αύξηση που παρουσιάζεται από τότε μέχρι τώρα. Η ίδια μελέτη εκτιμά ότι το 2020 θα φτάσει στα 44 zetabytes και μάλιστα ερευνητές τονίζουν ότι το 90% των δεδομένων δημιουργείται τα δύο τελευταία χρόνια [2]. Εκτός από τα δεδομένα που προέρχονται από τον άνθρωπο -που υπολογίζονται στα 1.7 megabytes ανά δευτερόλεπτο [3]- η είσοδος του IoT επέφερε εκτόξευση των κατηγοριών, των τύπων και του είδους των δεδομένων που επεξεργαζόμαστε και δημιουργούνται. Για παράδειγμα, οι διακομιστές του Walmart χειρίζονται περισσότερες από 1.000.000 συναλλαγές πελατών κάθε ώρα, και αυτές οι πληροφορίες εισάγονται σε βάσεις δεδομένων που αποθηκεύουν περισσότερα από 2,5 Petabytes δεδομένων — το ισοδύναμο του είναι 167 φορές ο αριθμός των βιβλίων στη βιβλιοθήκη του Κογκρέσου [4]. Ένα ακόμα παράδειγμα είναι ο μεγάλος επιταχυντής αδρονίων που παράγει περίπου 15 Petabytes δεδομένων ετησίως – αρκετά για να γεμίσει τα περισσότερα datacenter στην Ελλάδα [5]. Από την άλλη, κοινωνικά δίκτυα κάθε μέρα παράγουν μεγάλο όγκο δεδομένων, για παράδειγμα το Facebook λειτουργεί έχοντας

περίπου 500 Terabytes δεδομένων από τις περιγραφές χρηστών και αρκετών εκατοντάδων Terabyte δεδομένων από εικόνες. Παρόμοιοι ιλιγγιώδεις αριθμοί ισχύουν και για το Youtube [6], τα MMS που αποστέλλονται [7], τα tweets που παρουσιάζονται [8] και, τέλος, τα δεδομένα που επεξεργάζεται για την επιλογή καλύτερων προϊόντων το eBay [9]. Σε αυτά τα παραδείγματα παρουσιάζονται τα δεδομένα που παράγονται μόνο από τον άνθρωπο, εάν σε αυτό συνυπολογιστούν και τα δεδομένα – μεταδεδομένα που δημιουργούνται από μηχανές προς μηχανές ο αριθμός ξεφεύγει από κάθε φαντασία. Για παράδειγμα η αεροδιαστημική εταιρία Boeing για τους Jet κινητήρες μπορεί να παράγει 10 TeraByte επιχειρησιακών πληροφοριών για κάθε 30 λεπτά λειτουργίας. Εάν σε αυτό υπολογιστεί ότι ένα αεροπλάνο για να πετάξει από την Ελλάδα στην Αμερική θέλει πάνω από 10 ώρες αυτό αντιστοιχεί σε μερικές εκατοντάδες Terabyte δεδομένων. Γίνεται σαφές ότι για όλες τις πτήσεις που πραγματοποιούνται καθημερινά, εάν η Boeing ήθελε να αποθηκεύσει όλα τα δεδομένα, κανένα αποθηκευτικό σύστημα δεν θα μπορούσε να τα εξυπηρετήσει. Εμφανίζεται λοιπόν το εύλογο ερώτημα πως μπορούμε να διαχειριστούμε τέτοιο πλήθος δεδομένων, σε τόσο μικρό χρονικό διάστημα, με τέτοια ανομοιομορφία και τέτοια μεγάλη ταχύτητα δημιουργίας.

Η ανάλυση του Doug Laney στο [10] παρουσιάζει τους τρεις παράγοντες που μπορεί να οδηγήσουν και να μετατρέψουν ένα πρόβλημα σε μη διαχειρίσιμο λόγω των πολλών δεδομένων που καταλήγουν να πρέπει να επεξεργαστούν. Από τότε οι παράγοντες

έχουν αυξηθεί κυρίως λόγω της πολυπλοκότητας και της εξάρτησης των εφαρμογών από διαφορετικές πηγές δεδομένων αλλά και της ραγδαίας αύξησης των δεδομένων που παράγονται [11] [12]. Αρκετές δουλειές [13] [14] [15] [16] [17] [18] παρουσιάζουν επιπλέον “Vs” πέρα από τα volume, velocity και variety που παρουσιάστηκαν από τον Laney. Το πρόβλημα, ότι όλο και περισσότερες κατηγορίες προβλημάτων συμπεριλαμβάνουν τέτοια ζητήματα, γεγονός που αναγνωρίστηκε λίγο αργότερα κάνοντας το θέμα αυτό να βρίσκεται τόσο στις δέκα τεχνολογικές τάσης στρατηγικού χαρακτήρα [19] όσο και στις δέκα πιο κρίσιμες τεχνολογικές τάσεις για τα επόμενα πέντε χρόνια [20]. Το 2010, εκδόθηκε από το Λευκό Οίκο, και συγκεκριμένα από το γραφείο διαχείρισης ισολογισμού και το γραφείο επιστήμης και τεχνολογίας, η διακήρυξη ότι τα μεγάλα δεδομένα αποτελούν Εθνική πρόκληση και προτεραιότητα μαζί με την υγειονομική περίθαλψη και την εθνική ασφάλεια [21]. Ακολούθησε μετά από δυο χρόνια η υπηρεσία προηγμένων ερευνητικών έργων για την άμυνα που ανακοίνωσε επενδύσεις για εργαλεία διαχείρισης πολλών δεδομένων ώστε να αντιμετωπίσουν τους εθνικούς στόχους στους τομείς της ενεργειακής άμυνας, της εκπαίδευσης και της υγείας [22].

Μια πιο κριτική ματιά θα μπορούσε όμως να ανάγει την ανάγκη τέτοιων αναλύσεων στην όλο και μεγαλύτερη κατεύθυνση εταιριών και κρατών να μοντελοποιούν άτομα, χρήστες και πολίτες για λόγους μάρκετινγκ και ελέγχου. Η πρόκληση της προστασίας των προσωπικών δεδομένων έχει αναχθεί σε ανάγκη και χαρακτηριστικό που

πρέπει οι εφαρμογές και τα συστήματα να συμπεριλάβουν από τη φάση της μοντελοποίησης τους. Αρκετές μελέτες [23] [24] [25] [26] [27] τονίζουν το ευαίσθητο κομμάτι της διασφάλισης του ελέγχου των δεδομένων στην νέα αυτή εποχή των πολλών δεδομένων όπου πια τα δεδομένα θεωρούνται το νέο πετρέλαιο.

Το τεράστιο κενό της προστασίας των δεδομένων από τις εφαρμογές και τις υπηρεσίες τείνει να ξεφύγει από τα στενά όρια του προγραμματιστή είτε ακόμα και ενός οργανισμού ή εταιρίας. Ο κύριος λόγος είναι ότι οι εφαρμογές αρχίζουν να εξαρτώνται από μεθόδους και υπηρεσίες που αναπτύσσονται εκτός του πλαισίου ελέγχου της εταιρίας. Ο τρόπος επικοινωνίας είναι σε αυτές τις περιπτώσεις το διαδίκτυο όπου είτε μέσα από συγκεκριμένο κανάλι και πρωτόκολλο είτε χρησιμοποιώντας το HTTP επικοινωνούν για την ανταλλαγή πληροφοριών. Οι λόγοι που επιλέγεται αυτή η *outsource* στρατηγική είναι α) οι υπηρεσίες είναι εξεζητημένες και αναπτύσσονται από συγκεκριμένο πάροχο β) είναι ακριβό το κόστος δημιουργίας είτε συντήρησης της υπηρεσίας αυτής στη περίπτωση που επιλεγεί να αναπτυχθεί εντός των πλαισίων της εταιρίας γ) είναι φθηνότερο το κόστος λειτουργίας δ) επιλέγεται συνειδητά για λόγους μεταφοράς της ευθύνης σε περίπτωση σφάλματος είτε κακού QoS και ε) υπάρχει η ανάγκη για πολύ γρήγορη διαθεσιμότητα της υπηρεσίας οπότε έτοιμες υπηρεσίες είναι προτιμητέες για να μειώσουν τον χρόνο προγραμματισμού.

Η εύρεση και η καταλληλότητα μιας υπηρεσίας παραμένει όμως ένα ζητούμενο όπου καλείται ο προγραμματιστής είτε ο αρχιτέκτονας των υπηρεσιών να αντιμετωπίσει. Η αναζήτηση, ο εντοπισμός και η ανάλυση εάν κάποια υπηρεσία είναι κατάλληλη για το σκοπό και τη λειτουργικότητα που έχει στο μυαλό του ο προγραμματιστής παραμένει εμπόδιο. Ο κύριος λόγος είναι ότι σε αντίθεση με τον παραδοσιακό τρόπο προγραμματισμού όπου οι βιβλιοθήκες είναι μαζεμένες σε κεντρικά αποθετήρια που μπορεί να ανασυρθούν με εργαλεία αυτόματα είτε να αναζητηθούν σε αυτά, στις υπηρεσίες κάτι αντίστοιχο κεντροποιημένο δεν έχει επικρατήσει. Η λογική βέβαια και οι απαιτήσεις δεν είναι οι ίδιες μιας και ένα αποθετήριο υπηρεσιών είναι αρκετά πιο πολύπλοκο αφού δεν περιέχει μόνο το κώδικα και την άδεια χρήσης αλλά είναι μια πιο ολοκληρωμένη πλατφόρμα που για να είναι λειτουργική θα πρέπει να έχει παραμέτρους όπως το κόστος της εκάστοτε υπηρεσίας, τα ποιοτικά χαρακτηριστικά των αποτελεσμάτων που προσφέρει, λειτουργικές και μη λειτουργικές απαιτήσεις που έχει αλλά και αναλυτική περιγραφή του παρόχου της υπηρεσίας με στόχο να γίνει αντιληπτή η αξιοπιστία του.

Τα χαρακτηριστικά μιας εκάστοτε υπηρεσίας μπορούν και πρέπει για λόγους διαλειτουργικότητας, κατανόησης, σαφήνειας αλλά και ασφάλειας να οριστούν ώστε να μπορούν να δώσουν ρητές πληροφορίες για τη λειτουργία τους και το τρόπο εκτέλεσης τους. Αυτό το χαρακτηριστικό αρχείο που ουσιαστικά ορίζει εκτός των άλλων και το τρόπο εγκατάστασης της υπηρεσίας, μπορεί να παραλληλιστεί με το σχέδιο ενός προσχεδίου

(blueprint) που περιγράφει με λεπτομέρεια κάθε πτυχή αυτού. Έχοντας μια τέτοια περιγραφή για κάθε υπηρεσία θα μπορούσε σαν πρώτη ύλη να χρησιμοποιηθεί αυτή η συλλογή για να δομήσει μια αγορά (marketplace) υπηρεσιών ώστε να προσφέρει τη μέγιστη δυνατή πληροφορία στο προγραμματιστή.

1.1 Οργάνωση του εγγράφου

Η παρούσα διατριβή αποτελείται από έξι (6) κεφάλαια. Στις ενότητες των κεφαλαίων αυτών παρουσιάζεται με αναλυτικό τρόπο το αντικείμενο της διδακτορικής διατριβής.

Το υπόλοιπο έγγραφο δομείται ως εξής: στην Ενότητα 2 αναλύονται τα αποθετήρια δεδομένων καθώς και οι τρόποι βέλτιστης διαχείρισης των δεδομένων. Ακόμα παρουσιάζεται η μεθοδολογία διαχείρισης νέων δεδομένων από το χρονικό σημείο της εμφάνισης μέχρι το σημείο της μετατροπής τους σε πληροφορία και την μετέπειτα διαχείριση τους. Στην Ενότητα 3 εξετάζουμε τα βασικά ερευνητικά θέματα σχετικά με τη μελέτη της διάχυσης των υπηρεσιοστρεφών προγραμμάτων. Παρουσιάζεται πως μπορεί να δομηθεί μια πλατφόρμα αγορών που να μπορεί να εκθέτει υπηρεσίες ώστε αυτές να ενοικιαστούν είτε να πουληθούν. Ακόμα παρουσιάζεται μια πρότυπη υλοποίηση που διαχειρίζεται τα θέματα που επιφέρει το

Cloud Computing και ο κόσμος των πολλών δεδομένων. Η Ενότητα 4 μοντελοποιεί τις έννοιες που διέπουν οι επεξεργαστές συμβάντων και παρουσιάζει μια μελέτη σχετικά με τις αποδόσεις που έχουν τέτοιες αρχιτεκτονικές συγκρίνοντας μονολιθικά συστήματα με συστήματα που βασίζονται στα microservices. Στην Ενότητα 5 εξετάζονται οι ιδιαιτερότητες και τα θέματα ασφάλειας που αντιμετωπίζουν προσεγγίσεις που μετατρέπουν είτε ακολουθούν το μοντέλο των μικροπηρεσιών. Σε αυτή την ενότητα υπάρχει αναφορά στις ελλείψεις των έως τώρα αρχιτεκτονικών μοντέλων που οδηγούν στην ελλιπή απόδοση και αξιοπιστία. Σε συνέχεια των ευρημάτων και της ανάγκης για ενιαίο λεπτομερές αρχείο περιγραφής των υπηρεσιών που αναφέρεται στο προηγούμενο κεφάλαιο παρουσιάζεται μια πρότυπη υλοποίηση ενός Blueprint που καλείται να διαχειριστεί θέματα τόσο ασφάλειας όσο και συμβατότητας και κατανόησης των υπηρεσιών. Τέλος η Ενότητα 8 συνοψίζει την εργασία και προτείνει κατευθύνσεις για μελλοντική έρευνα. Η διατριβή ολοκληρώνεται με βιβλιογραφικές αναφορές.

ΚΕΦΑΛΑΙΟ 2

Αποθετήρια δεδομένων

Η απότομη είσοδος των μεγάλων δεδομένων (Big Data) στη πληροφορική είχε ως αποτέλεσμα να υπάρξουν μεταλλάξεις σε πολλούς από τους τομείς της. Ο κύριος λόγος για αυτές τις μεταλλάξεις ήταν η αλλαγή στη φιλοσοφία και την αξία των δεδομένων καθώς και η δυνατότητα να μπορούμε να αποθηκεύουμε αλλά και να τα επεξεργαζόμαστε, σε χρόνο που δεν είναι αποτρεπτικό για την αξία της πληροφορίας που θα εξαχθεί αυτά. Η εξέλιξη μπορεί να παραλληλιστεί με την εξέλιξη της βιομηχανίας του ψαρέματος όπου παρόλο που το ψάρεμα ξεκίνησε την προϊστορική εποχή, μόνο μετά από αρκετούς αιώνες (16ος αιώνας) δόθηκε η δυνατότητα στους ψαράδες να έχουν σαν εργαλείο δουλειάς τις βάρκες και να μπορούν να έχουν πρόσβαση σε μια μεγαλύτερη επιφάνια περιοχών ώστε να μπορούν να ψαρέψουν. Αυτή η εφεύρεση μετέτρεψε το ψάρεμα σε μια κερδοφόρο επιχειρηματική επιλογή με πολλές προοπτικές σε αντίθεση με τις μέχρι τότε επιλογές που υπήρχαν. Αυτό ήταν η “κλωτσιά γνώσης” που χρειαζόταν ώστε να ξεκινήσει μια σειρά από εφευρέσεις στο νέο κόσμο που ανοίχτηκε μπροστά. Τα πλοία μπορούσαν να ταξιδεύουν σε περισσότερες περιοχές και να βρίσκουν με ακρίβεια με χρήση της πυξίδας σημεία που ήξεραν ότι υπήρχαν ψάρια. Ξεκίνησε μετά η χρήση συστημάτων χαρτογράφησης και ναυσιπλοΐας με οδηγό τα αστέρια που με τα χρόνια εξελίχθηκε με τη χρήση GPS και συστημάτων παρακολούθησης του βυθού. Στη σύγχρονη εποχή υπάρχουν sonar, πληροφορίες σχετικά με τον καιρό σε πραγματικό χρόνο, αισθητήρες για τη

θερμοκρασία της θάλασσάς και των ρευμάτων και δορυφορικά συστήματα που δίνουν πληροφορίες για τις κινήσεις των κοπαδιών.

Αυτή η έξυπνη επανάσταση συσκευών, sensors και τεχνολογιών είναι αυτή που τροφοδοτεί την ανάπτυξη της Big Data τεχνολογίας. Θεωρώντας ότι αυτές οι συσκευές πραγματοποιούν μετρήσεις είτε εκτελούν ενέργειες, τα δεδομένα που παράγουν μπορούν να χρησιμοποιηθούν και κατά συνέπεια να αποθηκευτούν ώστε στο μέλλον να αναλυθούν για να παράγουν γνώση είτε από την εμπειρία που αποκομίσαμε από τα αποτελέσματα των ενεργειών είτε από τα ίδια τα δεδομένα χρησιμοποιώντας τα με διαφορετικό τρόπο. Αυτά τα ψηφιακά αποτυπώματα δημιουργούν τεράστιες ποσότητες δεδομένων που με τη σειρά τους “αναγκάζουν” τη τεχνολογικά να βρει λύσεις για να μπορέσει να τα επεξεργαστεί και να εξορύξει γνώση.

Μεγάλοι εταιρίες συμπεριλαμβανομένων των Amazon, Facebook, Walmart, Google και eBay έχουν συνειδητοποιήσει το πρόβλημα της ανάλυσης των πολλών δεδομένων καθώς και ότι με τις όλο και περισσότερες ανάγκες που δημιουργούνται τόσο περισσότερα δεδομένα πρέπει να κρατούν. Για παράδειγμα η Walmart που διαχειρίζεται πάνω από ένα εκατομμύριο συναλλαγές κάθε ώρα εισάγει όλα αυτά τα στοιχεία σε μια βάση δεδομένων που το μέγεθος της ξεπερνά τα 2.5 petabytes [28]. Σε μια τέτοια βάση υπάρχουν δεδομένα που προέρχονται από πολλές και διαφορετικές πηγές μέσα από το κύκλο εργασιών της εταιρίας. Αποθηκεύονται δεδομένα σχετικά με τις πωλήσεις των προϊόντων και τις τιμές τους, δεδομένα σχετικά με τη συχνότητα επισκέψεων των πελατών και τους τύπους των αγαθών που αγοράζουν, είτε δεδομένα σχετικά με τις αποδώσεις των

ημερών σε κύκλο εργασιών και πως αλλάζουν οι πωλήσεις σε περίοδο εκπτώσεων είτε εορτών. Μπορεί όμως μια τέτοια βάση να έχει δεδομένα και από εξωτερικές πηγές όπως από κοινωνικά δίκτυα βλέποντας πως κάποιο προϊόν και η τιμή του επηρεάζει θετικά είτε αρνητικά έναν αγοραστή, καθώς και τη δημοφιλία του. Επιπρόσθετα θα μπορούσε να περιέχει δεδομένα που σε μια πρώτη ματιά δεν έχουν καμία σχέση με τα προϊόντα είτε με τον αγοραστή αλλά με το περιβάλλον όπου δρα. Τέτοια δεδομένα θα μπορούσαν να ήταν τα δεδομένα καιρού, όπου μεγάλοι κολοσσοί μπορούν για παράδειγμα έχοντας όλα αυτά τα δεδομένα να προσφέρουν νέες κατευθυνόμενες και εξυπνότερες υπηρεσίες καθώς και προσφορές. Μια τέτοια περίπτωση είναι όταν ο πελάτης έχει στη λίστα με τα προϊόντα που έχει αγοράσει χαλιά και μοκέτες, ξέροντας λοιπόν ότι υπάρχει συνεργαζόμενο μαγαζί φύλαξης χαλιών σε ακτίνα 10 χιλιομέτρων μπορείς όταν ο καιρός αρχίζει να καλοκαιρεύει, ξεπερνώντας τους 28 βαθμούς Κελσίου, να του αποστείλεις κουπόνι προσφοράς για το συγκεκριμένο μαγαζί.

Σενάρια όπως το παραπάνω είναι εργαλεία απαραίτητα όχι μόνο για την ανάπτυξη μιας εταιρίας αλλά και για την επιβίωση της. Η ταχύτητα που αλλάζουν οι ανάγκες των ατόμων έχει αλλάξει δραματικά κάνοντας δύσκολη τη πρόγνωση τους καθώς δεν μπορούμε πια να ελέγχουμε την πληροφορία που δέχονται. Η εποχή που τα μέσα επικοινωνίας για τον άνθρωπο ήταν περιορισμένα στα δάχτυλα του ενός χεριού - εφημερίδες, τηλεόραση, ραδιόφωνο, περιοδικά- έχει περάσει και πια έχουμε φτάσει στην εποχή που το internet έχει εδραιωθεί και μονοπωλήσει. Αυτή η χωρίς φραγμούς και έλεγχο ανακάλυψη κάνει εξαιρετικά δύσκολο να προβλεφθεί από που επηρεάζεται ο

καταναλωτής. Ο συνδυασμός της χρήσης των εξωτερικών πηγών μαζί με τη δυσκολία να υπολογίσεις τις ανάγκες και να αναλύσεις τη διάθεση του καταναλωτή αναγκάζουν τα data warehouses να μην είναι όπως παλαιά αποδοτικά και ικανά ώστε να προσφέρουν τις υπηρεσίες που μια εταιρία θέλει να έχει. Ακόμα λιγότερα αποδοτικές βέβαια είναι οι περιπτώσεις χρήσης σχεσιακής βάσης δεδομένων για τη αποθήκευση αυτών των πληροφοριών.

Το μέλλον οποιασδήποτε επιτυχούς επιχείρησης εξαρτάται από την ικανότητά της να αξιοποιεί και να χρησιμοποιεί τεράστιες ποσότητες δεδομένων που είναι διαθέσιμες, προσφέροντας έτσι βελτίωση στη παραγωγικότητα, μείωση κόστους σε τομείς που μπορούν να βελτιστοποιηθούν και αύξηση κερδών. Ο τρόπος που αποθηκεύουν οι οργανισμοί τα δεδομένα όμως, είναι εξίσου σημαντικό στοιχείο και ίσως ο πυλώνας όλου αυτού του οικοδομήματος. Η επιλογή και η αναγνώριση της καλύτερης αποθηκευτικής λύσης περνά πρώτα από την ανάλυση των ιδιαιτεροτήτων, των μειονεκτημάτων και της αρχιτεκτονικής των υπάρχοντων λύσεων. Μέσα από αυτή την ανάλυση και τη σύγκριση μπορεί να φανούν οι προτεραιότητες που ένα υγιές και αποδοτικό σύστημα πρέπει να έχει.

2.1 Μοντέλα αρχιτεκτονικών δομών διαχείρισης δεδομένων

Η σωστή επιλογή του τρόπου αλλά και του τύπου των δομών δεδομένων που θα χρησιμοποιηθεί για την αποθήκευση και την επεξεργασία των δεδομένων μιας εφαρμογής

είναι απαραίτητος παράγοντας για την εύρυθμη και βέλτιστη λειτουργία της εφαρμογής [29]. Σε αυτό συνεπικουρεί και το γεγονός ότι όλο και περισσότερα δεδομένα είναι απαραίτητο να επεξεργαστούν στις σύγχρονες εφαρμογές.

Οι περιπτώσεις του WallMart, eBay, Amazon, Twitter, WhatsApp στην επεξεργασία, αποθήκευση και ανάλυση δεδομένων παρέχουν μια μικρή ματιά στο ταχέως διευρυμένο οικοσύστημα πηγών μαζικών συνόλων δεδομένων που υπάρχουν σήμερα. Τα δεδομένα μπορούν να διαρθρωθούν (π.χ. οικονομικά, στατιστικά, ηλεκτρονικά ιατρικά αρχεία εικόνων, στρατιωτικά αρχεία και πρωτόκολλα), ημι-δομημένα (π.χ., άρθρα, tweets, SMS), μη δομημένα (π.χ. ήχος και βίντεο), όπως ακόμα και σε σχέση με το χρόνο (stream δεδομένων) π.χ., stream tweets, γενικά αρχεία καταγραφής παρακολούθησης.

Για τη σωστή ανάλυση αλλά και εύρεση του καλύτερου μοντέλου θα πρέπει πρώτα να γίνει μια ολοκληρωμένη μελέτη των φόρτων εργασίας των δεδομένων ώστε να μπορεί να βοηθήσει στην κατανόηση των επιπτώσεών τους στο σχεδιασμό λογισμικού. Ακολουθώντας τις μεθόδους του αριθμητικού υπολογισμού [30] ο Mehul Shah et al [31] προσπάθησε να ορίσει ένα σύνολο "νάνων δεδομένων" — που σημαίνει βασικούς πυρήνες επεξεργασίας δεδομένων — που παρέχουν τρέχουσα και μελλοντική κάλυψη των φόρτων εργασίας με επίκεντρο τα δεδομένα. Σχεδιαζόμενα από ένα εκτεταμένο σύνολο φόρτων εργασίας, καθορίζουν ένα σύνολο διαστάσεων ταξινόμησης (χρόνος απόκρισης, μοτίβο πρόσβασης, σύνολο εργασίας, τύπος δεδομένων, ανάγνωση έναντι εγγραφής, πολυπλοκότητα επεξεργασίας) και παράγεται το συμπέρασμα ότι πέντε μοντέλα φόρτου εργασίας θα μπορούσαν να καλύψουν ικανοποιητικά φόρτους εργασίας με επίκεντρο τα

δεδομένα. Πρώτο μοντέλο είναι η κατανεμημένη ταξινόμηση σε κλίμακα, ακολουθεί η αναζήτηση ευρετηρίου στη μνήμη, το σύστημα σύστασης και οι πολυκριτριακοί παράγοντες που μπορούν να διέπουν αυτό, η κατάργηση διπλοτύπων δεδομένων με διαδοχική πρόσβαση και τέλος το ανέβασμα βίντεο μαζί με το αντίστοιχο σύστημα διαχείρισης ροής με τέτοια απόκριση που δεν επηρεάζει είτε ενοχλεί το χρήστη. Ενώ οι φόρτοι εργασίας ηλεκτρονικής αναλυτικής επεξεργασίας (OLAP) μπορούν εύκολα να εκφραστούν ως συνδυασμός του πρώτου, του τρίτου και του πέμπτου παράγοντα, οι φόρτοι εργασίας ηλεκτρονικής επεξεργασίας συναλλαγών (OLTP) μπορούν να συλληφθούν μόνο μερικώς. Αυτό μπορεί να γίνει αντιληπτό από μεγάλα σύνολα εργασιών που μπορεί να μην χωρέσουν στη μνήμη.

2.2 Συστήματα βάσεων δεδομένων

Επικρατέστερο σύστημα για τη δομή μιας βάσης δεδομένων είναι το σχεσιακό μοντέλο αποθήκευσης όπου είναι δημοφιλές μιας και περιέχει πληροφορίες που είναι οργανωμένες σε στήλες, σειρές και πίνακες, φιλοσοφία που είναι αρκετά κοντά στη λογική των προβλημάτων που καλείται να αντιμετωπίσει. Ταυτόχρονα με μηχανισμούς ευρετηρίασης οι οποίοι περιοδικά μπορεί να διασχίζουν τα δεδομένα της βάσης, είτε συγκεκριμένων στηλών και πεδίων, καθιστούν πιο γρήγορη την πρόσβαση σε σχετικές πληροφορίες. Αυτή η τεχνική είναι αποδοτική σε ερωτήματα που ο τύπος τους είναι γνωστός από πριν καθώς και είναι ξεκάθαρο ότι το ερώτημα θα βασίζεται σε αυτό το πεδίο για το -τουλάχιστον αρχικό- διαχωρισμό του. Παραδοσιακά παραδείγματα είναι βάσεις

δεδομένων που παρέχουν συγκεντρωτικές πληροφορίες για συγκεκριμένες λογικές καταστάσεις είτε οντότητες που χαρακτηρίζουν αυτές. Όλες οι ενημερώσεις και οι διαγραφές εμφανίζονται όταν φορτώνονται νέα δεδομένα επιτρέποντας στο συστήματα ευρετηρίασης να μπορεί να παρέχει ορθά και πλήρη αποτελέσματα.

Στη κατηγορία των βάσεων δεδομένων υπάρχει και άλλη μια φιλοσοφία που ακολουθεί διαφορετική τεχνική στο τρόπο που αποθηκεύει και διαχειρίζεται τα δεδομένα. Η κατηγορία αυτή έχει επικρατήσει να λέγεται NoSQL βάση δεδομένων. Σε αυτή τη κατηγορία η διαφοροποίηση βρίσκεται στο γεγονός ότι υπάρχει θεμελιακή διαφορά στο τρόπο δημιουργίας, στο τρόπο αποθήκευσης αλλά και στον τύπο της πληροφορίας που αποθηκεύεται. Ενώ οι σχεσιακές βάσεις έχουν δεδομένα που δομούνται με συγκεκριμένη μορφή και τρόπο, όπως για παράδειγμα το μητρώο ΑΦΜ του κάθε πολίτη, οι μη σχεσιακές μπορεί να περιλαμβάνουν δεδομένα σχετικά με τη διεύθυνση του ατόμου μαζί με τις προτιμήσεις του για τα πράγματα που αγοράζει από το internet και τους τόπους που επισκέπτεται.

Ενώ οι περισσότερες αναγνώσεις, εγγραφές και αναφορές διαχειρίζονται συνήθως από έναν Διαχειριστή Βάσεων Δεδομένων, ορισμένες βάσεις δεδομένων συναλλαγών παρέχουν συμμόρφωση με την ακεραιότητα, συνέπεια, απομόνωση και ανθεκτικότητα (ACID) για να διασφαλίζεται ότι οι πληροφορίες που περιέχονται στη βάση δεδομένων είναι συνεπείς και τυχόν συναλλαγές που πραγματοποιούνται , είναι πλήρεις.

Τα σύνθετα συστήματα σχεσιακών βάσεων δεδομένων μπορούν να οδηγήσουν τις βάσεις δεδομένων να μετατραπούν σε «νησιά πληροφοριών» όπου οι πληροφορίες δεν

μπορούν να διαμοιραστούν εύκολα από το ένα σύστημα βάσεων δεδομένων σε ένα άλλο. Συχνά σε μεγάλες επιχειρήσεις εντοπίζεται αυτό το πρόβλημα μιας και οι σχεσιακές τους βάσεις δεδομένων αναπτύχθηκαν σε χωριστά τμήματα με διαφορετικές προτεραιότητες και διαφορετικούς στόχους. Παράδειγμα είναι σχεσιακές βάσεις δεδομένων από διαφορετικές λειτουργίες της επιχείρησης ή του ομίλου που έχουν τους πελάτες καθώς και τα αντίστοιχα προϊόντα της κάθε ανεξάρτητης οντότητας. Αυτή η περίπτωση μπορεί να προσφέρει ευελιξία στη κάθε οντότητα να λειτουργήσει καθώς ορίζει μόνη της το σχήμα καθώς και το τρόπο που διαχειρίζεται αυτά τα δεδομένα αλλά δυστυχώς δημιουργείται το θέμα ότι δεν μπορεί να ενοποιηθεί όλο το πελατολόγιο μιας και όπως είναι λογικό όλοι οι πελάτες θα έπρεπε να διαχειρίζονται με τον ίδιο τρόπο και να υπάρχει πρόσβαση στα δεδομένα αυτών από όλα τα τμήματα της εταιρίας ώστε να προσφέρει ακόμα καλύτερη εξυπηρέτηση σε αυτούς μιας και αυτός είναι ο βασικός στόχος της κάθε εταιρίας.

Μια διαφορετική προσέγγιση είναι αυτή που εισήγαγε η Google με την ανάπτυξη εφαρμογών που είχαν ως βάση το διαδίκτυο και τα συστήματα διαχείρισης δεδομένων [32]. Η ιδέα αυτή εμπλουτίστηκε μεγαλώνοντας το οικοσύστημα με άλλα συστήματα που είχαν να κάνουν με βέλτιστους τρόπους επεξεργασίας και αποθήκευσης όπως το MapReduce [33], Chubby [34] και το Bigtable [35]. Την προσέγγιση αυτή ενστερνίστηκαν και άλλοι μεγάλοι πάροχοι (Yahoo, Amazon, Netflix, EBay) ακολουθώντας την τάση που ήθελε όλο και περισσότερες εφαρμογές να δομούνται και να εξαρτώνται από μη κεντρικοποιημένα συστήματα που χρησιμοποιούν ως κανάλι επικοινωνίας το διαδίκτυο. Αυτές οι εφαρμογές δεν έχουν μόνο διαφορετική λειτουργία αλλά διαφοροποιούνται

αρκετά και στην ίδια τη φιλοσοφία της επεξεργασίας και ανάκτησης των δεδομένων. Το αποτέλεσμα είναι η δημιουργία πολλών διαφορετικών πλατφόρμων αποθήκευσης που ακολουθούν τη NoSQL λογική. Οι βασικοί τύποι διαφοροποιήσεων είναι α) η ανάγκη για αποθήκευση και επεξεργασία τεράστιων όγκων δεδομένων όπως αυτή των φωτογραφιών, των βίντεο, των γεωγραφικών δεδομένων κλπ, β) η ανάγκη για πρόσβαση σε πραγματικό χρόνο αποθηκευμένων τεράστιων δομών δεδομένων που πρέπει να είναι διαθέσιμοι από οπουδήποτε στον κόσμο γ) η ανάγκη για ευελιξία: δυνατότητα εύκολης και γρήγορης αλλαγής της δομής των δεδομένων αυτών δ) η ανάγκη αποθήκευσης μη δομημένων δεδομένων και ε) η ανάγκη για δυνατότητα κλιμάκωσης. Αυτές οι απαιτήσεις έστρεψαν το ενδιαφέρον σε νέες υλοποιήσεις που μπόρεσαν να ανταπεξέλθουν αν όχι ταυτόχρονα σε όλα, τουλάχιστον στα περισσότερα από αυτά. Απαιτήσεις που δεν μπορούσε το παραδοσιακό μοντέλο των σχεσιακών βάσεων να ακολουθήσει.

2.2.1 Εξωτερικές πηγές δεδομένων

Εκτός από τη διαχείριση και την επεξεργασία των δεδομένων που παράγονται από τα συστήματα και τις εφαρμογές μέσα στο στενό πλαίσιο της υποδομής, ένα σημαντικό στοιχείο που θα πρέπει να εστιαστεί είναι ο προσανατολισμός προς την αξιολόγηση και τη διαχείριση των διαύλων δεδομένων που προέρχονται από εξωτερικές πληροφορίες. Ο έλεγχος και η διαχείριση των κρίσιμων υποδομών δεν μπορούν να μείνουν ανεπηρέαστες από το εξωτερικό περιβάλλον τους όσον αφορά την αρχιτεκτονική ή την ασφάλειά τους. Τα δεδομένα από τον έξω κόσμο τείνουν να αυξηθούν εκθετικά σε μέγεθος, και αυτό είναι

κάτι που πιέζει τη διαδικασία της προστασίας, της διατήρησης και της αναβάθμισης μιας υπάρχουσας υποδομής, πόσο μάλλον εάν αυτή είναι κρίσιμη. Από το 2013, το ανθρώπινο δημιουργηθέν περιεχόμενο εκτιμήθηκε σε 2,5 exabytes ($2,5 \times 10^{18}$), το 90% των οποίων δημιουργήθηκε τα τελευταία δύο έτη [36]. Το 2025, η Διεθνής εταιρεία δεδομένων (IDC) προβλέπει ότι θα υπάρχουν συνολικά 163 zettabytes (163×10^{21}) δεδομένων [37]. Αυτή η τεράστια ποσότητα δεδομένων εγείρει την ανάγκη για ένα ποιοτικό πλαίσιο που θα λαμβάνει υπόψη τις διάφορες πηγές του και θα οδηγεί σε ποιοτικά δεδομένα.

2.3 Μεθοδολογία κύκλου ζωής πληροφορίας

Τα δεδομένα έχουν μοναδικό και εξέχοντα ρόλο τόσο στον εταιρικό κόσμο όσο και στο χώρο των υποδομών ασφάλειας που διαχειρίζονται νευραλγικές και εμπιστευτικές πληροφορίες. Η δεύτερη κατηγορία εμπεριέχει αρκετές τεχνικές προκλήσεις όσο αναφορά τη διαχείριση μιας και δεν υπάρχει μια ολιστική λύση που να προσεγγίζει τις ανοιχτές προκλήσεις της αξιοπιστίας και της διασφάλισης των δεδομένων. Συγκεκριμένα για τη διασφάλιση των κρίσιμων υποδομών με αποτελεσματικό και έγκαιρο τρόπο, είναι πολύ σημαντικό να εξεταστεί και να κατανοηθεί κάθε πτυχή του κύκλου ζωής και της διαχείρισης των πληροφοριών.

Σε ένα γενικό πλαίσιο, ο κύκλος ζωής των πληροφοριών μπορεί να θεωρηθεί ότι έχει μια τυποποιημένη δομή που αποτελείται από τα ακόλουθα βασικά στάδια: συλλογή, επικύρωση, αποθήκευση, ανάλυση, χρήση και αξιολόγηση. Ωστόσο, όσον αφορά τα συστήματα ασφαλείας, χρειάζεται μια πιο αναλυτική και ολοκληρωμένη δομή, που θα ενσαρκώνει νέα στάδια δίνοντας έτσι την ευελιξία που πρέπει να ενσωματώνει ένα

σύγχρονο σύστημα διαχείρισης. Ο προτεινόμενος κύκλος ζωής των πληροφοριών είναι ένα υβριδικό μοντέλο βασισμένο στα πρωτόκολλα του Οργανισμού Βορειοατλαντικού Συμφώνου (NATO), του Οργανισμού Ευρωπαϊκής Συνοριοφυλακής και Ακτοφυλακής — Frontex [38] και του Ελληνικού στρατού [39], που είναι προσαρμοσμένοι για να αντιμετωπίσουν τις προκλήσεις στη διαχείριση της πληροφορίας. Σε αντίθεση με τα προαναφερθείσα συστήματα που έχουν ως επίκεντρο του ανθρώπινου κεφάλαιο, η εφαρμογή τους σε πληροφοριακά συστήματα δεν είναι τετριμμένη και δεν αντικατοπτρίζει τις προκλήσεις που εμφανίζονται. Τα πληροφοριακά συστήματα έχουν διαφορετικές ιδιαιτερότητες που θα πρέπει να λογιστούν ώστε να θεωρηθεί πλήρες ένα σύστημα. Οπότε η μίξη και η δημιουργία ενός υβριδίου που χρησιμοποιεί τη γνώση από τα αυστηρά συστήματα των πιο πάνω οργανισμών και τη χρήση των νέων τεχνικών και αρχιτεκτονικών που προσφέρει η ασφάλεια στο τομέα της πληροφορικής είναι επιβεβλημένη. Αυτή η προσέγγιση πρόκειται για μια νέα καινοτόμο προσέγγιση που μπορεί να διαχειριστεί ένα ευρύ φάσμα πηγών δεδομένων.

Ο προτεινόμενος κύκλος ζωής των πληροφοριών αποτελείται από τα ακόλουθα στάδια όπως παρουσιάζεται και στην Εικόνα 1:

Απόδοση εργασιών (Tasking): είναι η πρώτη φάση του κύκλου που καθορίζει όλους τους στόχους ασφάλειας και τις αποφάσεις που πρέπει να ληφθούν για τον καθορισμό των απαιτήσεων πληροφόρησης και, συνεπώς, των πηγών δεδομένων που θα χρησιμοποιηθούν καθώς και των δεδομένων που θα συγκεντρωθούν. Κάθε εργασία ασφαλείας είναι μια ανεξάρτητη οντότητα που έχει τις δικές της απαιτήσεις δεδομένων. Οι

εργασίες οργανώνονται σε κατηγορίες που σχετίζονται με το τρέχον σενάριο ασφαλείας. Επιπλέον, λόγω του διακριτού χαρακτήρα τους, τα καθήκοντα δεν έχουν την ίδια σημασία και, κατά συνέπεια, μπορεί να έχουν διαφορετικές προτεραιότητες.

Πηγές δεδομένων (Data Sources): Ανεξάρτητα από το πόσο καλά έχει συσταθεί και οργανωθεί το στάδιο της εκτέλεσης των εργασιών, είναι ζωτικής σημασίας να προσδιοριστούν οι κατάλληλες πηγές δεδομένων που ανταποκρίνονται στις απαιτήσεις των εργασιών. Για να καθορίσουμε τις πηγές από τις οποίες μπορούμε να αντλήσουμε πληροφορίες, πρέπει πρώτα να τις κατηγοριοποιήσουμε. Για το σκοπό αυτό, χρησιμοποιούμε τους κλάδους συλλογής πληροφοριών, οι οποίοι περιγράφουν αυτού του είδους τις πληροφορίες, καθώς και την πηγή και το σχήμα τους. Εν συντομία, οι πιο σημαντικοί από αυτούς τους κλάδους είναι οι εξής: HUMINT (ανθρώπινη νοημοσύνη) [40], GEOINT (γεωχωρική νοημοσύνη), OSINT (υπηρεσία πληροφοριών ανοικτού κώδικα), SIGINT (νοημοσύνη σημάτων) και COMINT (πληροφορίες επικοινωνίας) [41], TECHINT (τεχνολογία πληροφοριών), CYBINT/DNINT (πληροφορίες για τον κυβερνοχώρο/ψηφιακό δίκτυο πληροφοριών) και IMINT (νοημοσύνη) [41]. Επιπλέον, τα δεδομένα από όλες αυτές τις πηγές μπορούν να συνδυαστούν με υπάρχοντα δεδομένα στις βάσεις δεδομένων της υποδομής.

Συλλογή δεδομένων (Data Collection): Αυτό είναι το στάδιο στο οποίο συγκεντρώνονται τα δεδομένα από τις καθορισμένες ροές προέλευσης δεδομένων. Περιέχει όλες τις μεθόδους και τις διεπαφές προγραμματισμού εφαρμογών (API) για τη σύνδεση με τις πηγές, προκειμένου να σχεδιαστούν τα ζητούμενα δεδομένα. Είναι

σημαντικό το στάδιο αυτό να είναι πάντα ενημερωμένο και να ακολουθεί όλες τις αλλαγές που μπορεί να προκύψουν στις πηγές ή στα δεδομένα (π.χ. αλλαγή της μορφής των δεδομένων ή αλλαγή στον τρόπο πρόσβασης στις πηγές κ.λπ.). Είναι επίσης ζωτικής σημασίας να διαθέτουμε τα κατάλληλα εργαλεία για τη συλλογή αυτών των δεδομένων, αφού δεν παρέχονται όλα μέσω υπηρεσιών ή API. Για το σκοπό αυτό, μπορεί να είναι απαραίτητο να χρησιμοποιούνται μηχανές ανίχνευσης για να καταγραφούν και να συλλεχθούν (δυναμικές) ιστοσελίδες, εικόνες και βίντεο από τοποθεσίες του διαδικτύου, αγορές, φόρουμ, κοινωνικά μέσα και πηγές σκοτεινής ιστού.

Επιλογή δεδομένων (Data Cleaning): Αυτό το στάδιο εξασφαλίζει την ποιότητα των δεδομένων επί των οποίων ενεργούμε. Ακόμα ενσωματώνει τους δείκτες ποιότητας των πολλών δεδομένων [42], καθιστώντας βέβαιο ότι ενεργούμε σε καθαρά, ορθά, ακριβή και ολοκληρωμένα δεδομένα. Με βάση αυτούς τους δείκτες, το στάδιο καθαρισμού δεδομένων εντοπίζει όλες τις ανωμαλίες και τις καταστροφές που προκύπτουν από τα δεδομένα που προέρχονται από ένα σύνολο εγγραφών, έναν πίνακα ή μια βάση δεδομένων. Στη συνέχεια, προχωρά μόνο με τις κατάλληλες προδιαγραφές, προκειμένου να ανταποκριθεί στην επιθυμητή ποιότητα που απαιτεί.

Μετασχηματισμός δεδομένων (Data Transformation): Ανεξάρτητα από τη διαδικασία καθαρισμού που εφαρμόζεται στα δεδομένα, αυτά παραμένουν μη δομημένα και χωρίς σχήμα. Για να είναι σε θέση να αποθηκεύονται και να αναλυθούν, πρέπει να έχουν κάποια δομή. Για το λόγο αυτό, εφαρμόζονται οι σωστοί μετασχηματισμοί οπότε σε αυτό το στάδιο μετατρέπονται ένα σύνολο τιμών δεδομένων από τη μορφή δεδομένων του

συστήματος προέλευσης σε μορφή δεδομένων ενός συστήματος δεδομένων προορισμού [43].

Ευρετηριοποίηση και αποθήκευση (Index and Storage): Σε αυτό το στάδιο, τα συλλεγόμενα δεδομένα αποθηκεύονται για μελλοντική χρήση. Λόγω του μεγάλου ποσού των δεδομένων, καταφεύγουμε σε έναν πιο οργανωμένο και δομημένο τρόπο αποθήκευσης. Επιπλέον, με τη χρήση αλγορίθμων ευρετηρίου ή κατακερματισμού, αποδοτικών βάσεων δεδομένων, η ανάκτηση είναι εφικτή. Υπάρχουν πολλές επιλογές ευρετηριοποίησης, από τη δημιουργία ευρετηρίων μη τεχνητής νοημοσύνης, μέχρι την τεχνητή νοημοσύνη και τις τεχνικές συνεργατικής τεχνητής νοημοσύνης [43].

Συρραφή δεδομένων (Data Collation): όπως αναφέρεται, κάθε εργασία απαιτεί ένα διαφορετικό σύνολο δεδομένων, το οποίο οδηγεί στην πρόσβαση σε πολλές προελεύσεις δεδομένων. Όλες οι εργασίες θα διαμορφωθούν τελικά μαζί, εξυπηρετώντας τον αρχικό στόχο ασφαλείας. Το ίδιο πρέπει να γίνει με τα δεδομένα. Αυτό είναι το στάδιο όπου τα δεδομένα συγκεντρώνονται και συνδυάζονται σωστά, προκειμένου να τροφοδοτηθούν στο στάδιο της ανάλυσης όπου θα γίνουν οι σχέσεις.

Ανάλυση δεδομένων (Data Analysis): Τα προ-επεξεργασμένα δεδομένα σε αυτό το στάδιο αναλύονται καταλλήλως με στόχο την ανακάλυψη χρήσιμων πληροφοριών για τη στήριξη της λήψης αποφάσεων. Ανάλογα με τα δεδομένα που παρέχονται, τεχνικές όπως η εξόρυξη δεδομένων, οι περιγραφικές στατιστικές ή η ανάλυση διερευνητικών δεδομένων και η απεικόνιση δεδομένων μπορούν να χρησιμοποιηθούν για τη μοντελοποίηση και την ανακάλυψη γνώσεων. Επιπλέον, οι τεχνικές βαθιάς μάθησης που

μπορούν να επιτύχουν αφηρημένες αναπαραστάσεις καθώς και οι σημασιολογικές ετικέτες μπορούν να αποκτήσουν σχεσιακές και σημασιολογικές γνώσεις [44], Εξαγάγοντας τις μέγιστες πληροφορίες από τα δεδομένα που δόθηκαν. Η διαδικασία ανάλυσης δεδομένων μπορεί να οδηγήσει σε πρόσθετες αιτήσεις για δεδομένα ή πληροφορίες, επομένως είναι απαραίτητο να συσχετίσουμε αυτό με ιστορικά δεδομένα και πληροφορίες μοτίβου από προηγούμενη ανάλυση.

Εξαγωγή πληροφοριών (Information Extraction): Αυτό είναι το στάδιο στο οποίο η γνώση που είχε παραχθεί προηγουμένως μεταφράζεται σε συμπεράσματα, προβλέψεις και υπολογισμούς. Αυτό το βήμα αποτελεί αναπόσπαστο μέρος της ανάλυσης δεδομένων, καθώς ολοκληρώνει ολόκληρη τη φάση επεξεργασίας δεδομένων. Τα επακόλουθα αποτελέσματα και οι γνώσεις που προκύπτουν από το στάδιο της ανάλυσης δεδομένων είναι ενοποιημένα υπό μία πλήρη πληροφορία.

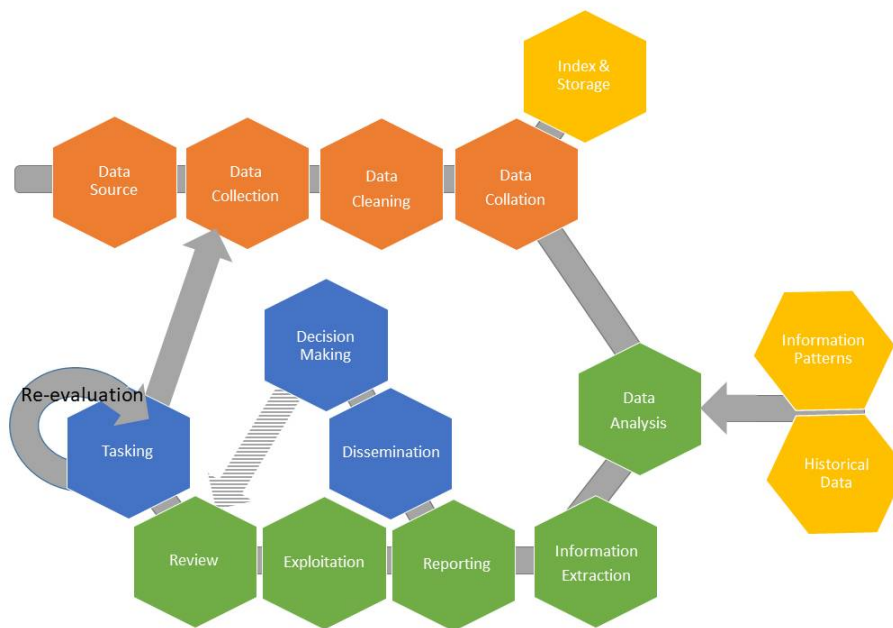
Αναφορά (Reporting): η φάση αυτή είναι η σύνοψη των αποτελεσμάτων που διαπιστώθηκαν κατά τη διάρκεια των δραστηριοτήτων ανάλυσης, συμπεριλαμβανομένων των τυχόν μέτρων βεβαιότητας που εμπλέκονται.

Διάδοση (Dissemination): Τα αποτελέσματα και η γνώση διανέμονται σε αυτό το στάδιο. Οι πληροφορίες που παράγονται διαβιβάζονται στο εξωτερικό περιβάλλον της κρίσιμης υποδομής, για παράδειγμα σε άλλους οργανισμούς και υπηρεσίες, οι οποίοι θα λαμβάνουν τις κατάλληλες αποφάσεις και θα προχωρούν σε δράσεις.

Εκμετάλλευση (Exploitation): Σε αυτό το στάδιο, η γνώση που παράγεται προωθείται και αξιοποιείται εσωτερικά μεταξύ της διοίκησης της υποδομής. Ο κύριος σκοπός είναι να χρησιμοποιηθούν τα αποτελέσματα για τα δικά τους συμφέροντα και να τροφοδοτήσουν τα συμπεράσματά τους και να έχουν κατάλληλα αποτελέσματα στη διαδικασία αναθεώρησης.

Λήψη αποφάσεων (Decision Making): Σε αυτό το στάδιο, με βάση τις πληροφορίες που λαμβάνουν οι σχετικοί αποδέκτες, λαμβάνονται οι κατάλληλες αποφάσεις. Τα αποτελέσματα των δράσεων που ακολουθούν τις αποφάσεις έχουν μεγάλο αντίκτυπο στην ανατροφοδότηση του κύκλου πληροφοριών, δεδομένου ότι αντικατοπτρίζουν τη λειτουργικότητα και την ποιότητα του ίδιου του κύκλου.

Επανεξέταση (Review): Τέλος, τα αποτελέσματα των αποφάσεων και, ως εκ τούτου, η ποιότητα των παραγόμενων δεδομένων επανεξετάζονται προκειμένου να δημιουργηθούν τα κατάλληλα σχόλια για την επαναξιολόγηση του κύκλου ζωής των πληροφοριών. Κάθε μέρος της φάσης της εφαρμογής θα πρέπει να εξετασθεί, να συγκρίνεται και να αξιολογείται έναντι της αρχικής πρόθεσης και του σκοπού της όσον αφορά το τι επιτεύχθηκε τελικά. Η διαμόρφωση των ίδιων ερωτήσεων, η αξιολόγηση της ποιότητας ολόκληρης της διαδικασίας, η σύνοψη των αποτελεσμάτων και η ερμηνεία της αποτελεσματικότητάς τους και του αντικτύπου τους είναι ζωτικά βήματα για μια υγιή ανασκόπηση που θα αναβαθμίσει και θα ενισχύσει τη λειτουργία του κύκλου ζωής των πληροφοριών.



Εικόνα 1 Κύκλου ζωής πληροφοριών

Για την αύξηση της παραγωγικότητας, απαιτείται ένας μηχανισμός για την αποθήκευση πληροφοριών και την αποτελεσματική ανάκτηση πληροφοριών. Τα δεδομένα θα έχουν συλλεχθεί προσεκτικά μέσω εκτενούς έρευνας και ανάλυσης της προέλευσης τους. Θα συνίστανται όχι μόνο σε πληροφορίες που είναι χρήσιμες, αλλά και σε δεδομένα που θα μπορούσαν να ικανοποιήσουν τις μελλοντικές ανάγκες. Επιπλέον, αυτό επιλύει πιθανά ζητήματα που έχουν επισημανθεί από εξειδικευμένους ειδικούς είτε από άλλους φορείς που μπορεί να είχαν αντιμετωπίσει ίδια προβλήματα είτε χρησιμοποίησαν τα εν λόγω δεδομένα για σκοπό που λογικά θα φανεί χρήσιμο στο μέλλον. Αυτή η αξιολόγηση των δεδομένων και των πηγών, η οποία διεξάγεται στο πλαίσιο του κύκλου ζωής των

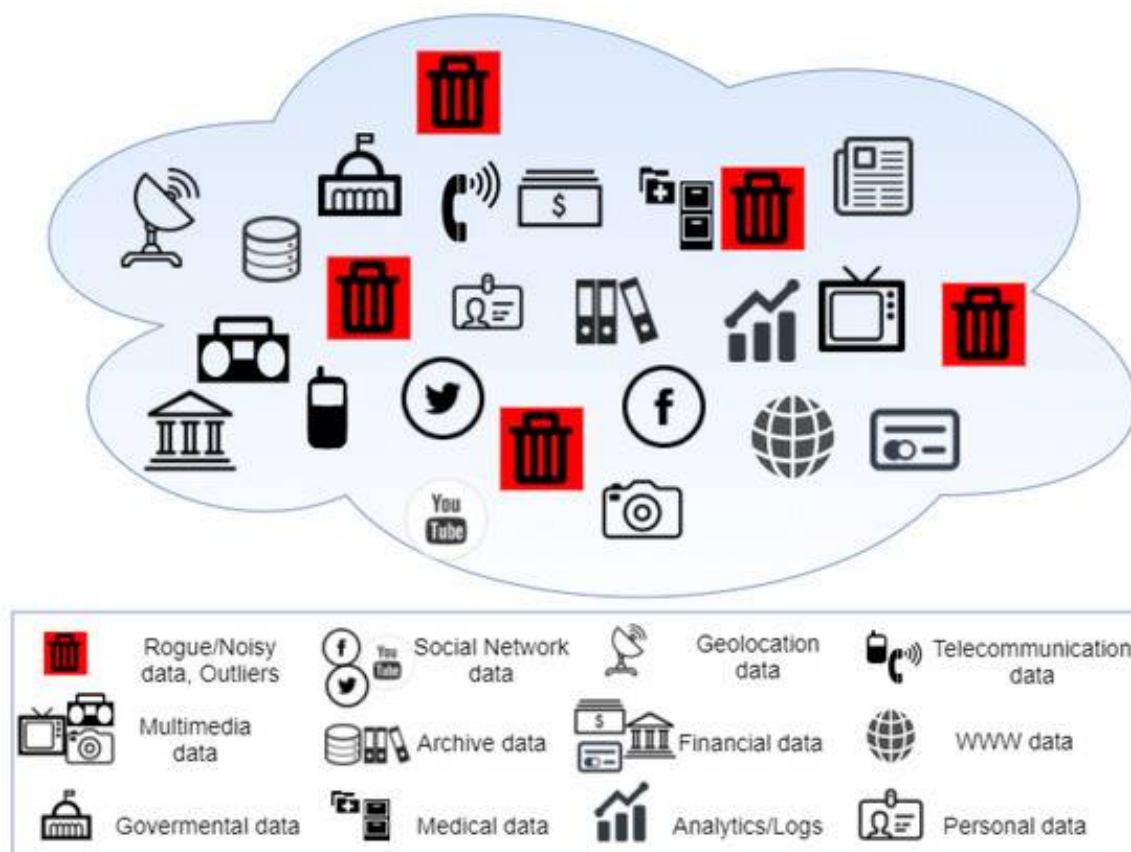
πληροφοριών, δεν θα πρέπει να είναι μονοδιάστατη και αυστηρή ως προς το τρόπο υλοποίησης της, επειδή ο τύπος δεδομένων και η τεχνολογία που συνοδεύει συνεχώς εξελίσσεται.

Ένα ζήτημα που είναι πολύ πιο εμφανές στην εποχή των μεγάλων δεδομένων είναι η έλλειψη διακυβέρνησης και η απουσία οποιασδήποτε δομής, καθιστώντας την ανάκτηση και τη χρήση των δεδομένων εξαιρετικά περίπλοκη. Επιπλέον, η κατανάλωση περιττών ή εκτός περιεχομένου δεδομένων θα μπορούσε να προκαλέσει αρνητικές παρενέργειες στους διάφορους τύπους πόρων και υποδομών. Αυτός ο "ωκεανός δεδομένων" δεν μπορεί να παρέχει κατάλληλα αποτελέσματα λόγω του υπερβολικού ποσού και της ποικιλίας των πληροφοριών που περιέχει. Τα δεδομένα που περιέχονται θα μπορούσαν να προέρχονται από πληθώρα πηγών, όπως: (α) κοινωνικά δίκτυα όπως το Facebook και το Twitter, (β) κυβερνητικά και χρηματοπιστωτικά ιδρύματα, όπως εμπορικές αγορές και τράπεζες, (γ) νοσοκομεία και κλινικές υγείας, (δ) αποθήκες δεδομένων ή/και κάθε άλλη πηγή που θα μπορούσε να κατέχει χρήσιμες πληροφορίες. Ωστόσο, ταυτόχρονα, υπάρχουν στοιχεία που θα μπορούσαν τελικά να μειώνουν την απόδοση του συστήματος όπως απεικονίζεται στην Εικόνα 2 (που απεικονίζεται ως κόκκινος κάδος απόρριψης), δεδομένου ότι δεν υπάρχει μηχανισμός για να το φιλτράρουν. Χωρίς την ύπαρξη του εν λόγω μηχανισμού, ορισμένα από τα δεδομένα που συγκεντρώνονται μπορεί να είναι θορυβώδη, μη διαχειριζόμενα ή ακόμη και σπασμένα. Αυτή η έλλειψη πληροφοριών από πολλαπλές πηγές και η ύπαρξη "παραπλανητικών δεδομένων" μπορεί να μεταμορφώσει αυτόν τον ωκεανό δεδομένων σε έναν τεράστιο βάλτο.

Λαμβάνοντας υπόψη τις προκλήσεις που αναφέρθηκαν παραπάνω και έχοντας θεωρητικά περιγράψει τον κύκλο ζωής των πληροφοριών, θα πρέπει να καθορίσουμε τον τρόπο εφαρμογής των πρώτων σταδίων του. Όπως ορίζεται στους όρους μεγάλων δεδομένων, οι λίμνες δεδομένων [45] μοιάζουν με τη σωστή προσέγγιση μοντελοποίησης. Πρόκειται για ένα αποθετήριο δεδομένων αποθήκευσης με δυνατότητα κλιμάκωσης, το οποίο κατέχει ένα τεράστιο όγκο στοιχείων στην εγγενή του μορφή (δομημένο, ημιδομημένο ή ακόμη και μη δομημένο) και ως εκ τούτου είναι ο ιδανικός υποψήφιος για το στάδιο της συλλογής δεδομένων. Η δομή και οι απαιτήσεις των δεδομένων δεν καθορίζονται μέχρι να χρειαστούν τα δεδομένα, τα οποία παρέχουν μεγάλη ευελιξία δίνοντας στους προγραμματιστές και τους επιστήμονες δεδομένων τη δυνατότητα να ρυθμίσουν εύκολα και να ρυθμίσουν ξανά τα μοντέλα, τα ερωτήματα και τις εφαρμογές τους. Αυτός είναι ένας αποτελεσματικός τρόπος για τις επιχειρήσεις και τις πλατφόρμες να συλλέγουν και να αποθηκεύουν πρωτογενείς πληροφορίες από μια ποικιλία εισροών, από δεδομένα διαχείρισης πελατειακών σχέσεων (CRM) σε δημοσιεύσεις μέσω κοινωνικής δικτύωσης.

Επιπλέον, θα χρειαστούν λιγότερο χρόνο για την πρόσβαση και τον εντοπισμό δεδομένων, ενισχύοντας έτσι την προετοιμασία των δεδομένων και την επαναχρησιμοποίηση των αποκτημάτων. Οι λίμνες δεδομένων έρχονται με πρόσθετα προνόμια που φαίνεται να είναι πολύ χρήσιμα στη χρήση και την ανάλυση των δεδομένων. Δεδομένου του ρυθμού αλλαγής στην επιστήμη των δεδομένων, η δυνατότητα αποθήκευσης αυτών των πληροφοριών στην εγγενή μορφή τους θα ενισχύσει σίγουρα τη

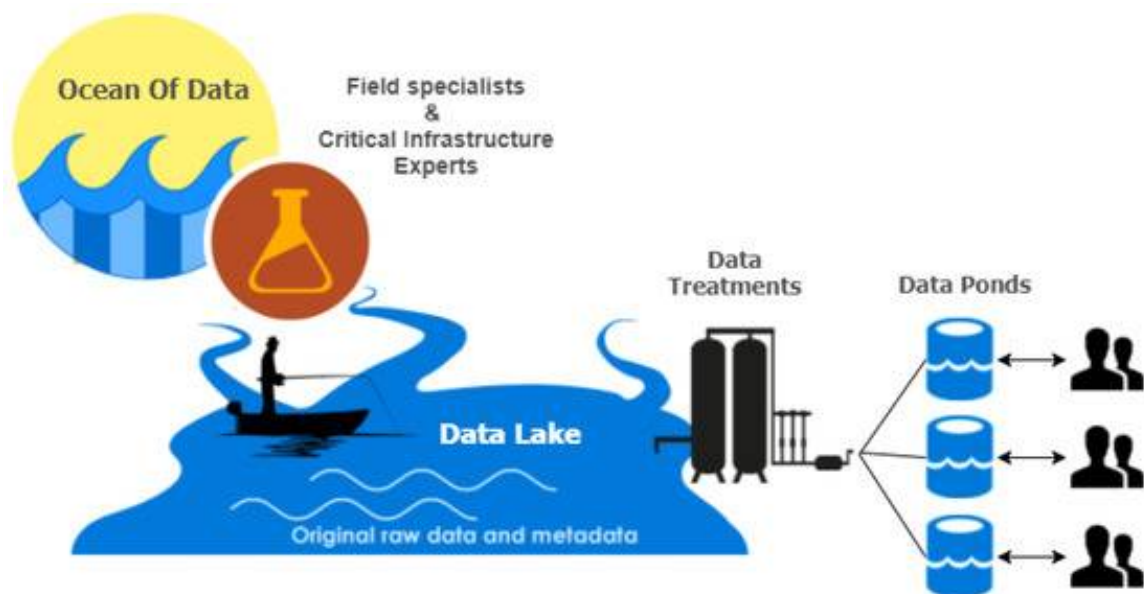
διαλειτουργικότητα, όπως θα είναι ευκολότερη στη χρήση σε μελλοντικά συστήματα. Το κόστος αποθήκευσης είναι επίσης ένα από τα κύρια ζητήματα που πρέπει να εξεταστούν και, ως εκ τούτου, δεδομένης της συγκριτικής μη δομημένης φύσης και της ποικιλίας των δεδομένων που μπορούν να αποθηκευτούν χωρίς προηγούμενη επεξεργασία, οι λίμνες δεδομένων προσφέρουν υγιή οικονομική αξία [45]. Τα ανεπαρκώς σχεδιασμένα και παραμελημένα συστήματα αντιμετωπίζουν τον κίνδυνο μετασχηματισμού σε βάλτους, που περιέχουν αχρησιμοποίητη πληροφορία που όχι μόνο θα αυξηθεί στο συνολικό μέγεθος, αλλά μπορεί επίσης να δημιουργήσει ανακρίβειες κατά την εκτέλεση ορισμένων αλγορίθμων (π.χ. μηχανική μάθηση) [46]. Αυτός είναι ο λόγος για τον οποίο οι λίμνες δεδομένων ουσιαστικά μετατρέπονται σε άχρηστες και αρχίζουν να έχουν κοινά χαρακτηριστικά με τον αχανή ωκεανό των δεδομένων.



Εικόνα 2 Τύποι πληροφοριών που δημιουργούν τον Ωκεανό των δεδομένων

Δημιουργώντας ένα αποδοτικό και ευέλικτο επίπεδο αποθήκευσης, το επόμενο βήμα είναι να ανακτήσετε τις απαιτούμενες πληροφορίες και να εκτελέσετε αυστηρή ανάλυση σε αυτό. Η εκτέλεση αυτού του είδους ενεργειών στα δεδομένα που περιέχει μια λίμνη δεδομένων δεν εξυπηρετεί κανένα σημαντικό σκοπό και δεν θα αποφέρει τα επιθυμητά αποτελέσματα. Αυτός είναι ο λόγος για τον οποίο είναι προτιμότερο να επικεντρωθούμε σε μια περιοχή ενδιαφέροντος ή σε ένα είδος περιεχομένου, όπως οι

πωλήσεις, το μάρκετινγκ, οι αναρτήσεις μέσω κοινωνικής δικτύωσης ή οι δορυφορικές και εναέριες φωτογραφίες. Η «λίμνη» μπορεί να είναι προτιμότερη, επειδή η τοποθεσία και η πρόσβαση των δεδομένων σχετικά με μια ορισμένη κατηγορία μπορεί να πραγματοποιηθεί πολύ αποτελεσματικά. Ωστόσο, όταν απαιτούνται τα δεδομένα, πρέπει να διενεργούνται ορισμένες διαδικασίες επεξεργασίας. Μεταξύ της αποθήκευσης δεδομένων και της επεξεργασίας δεδομένων βρίσκεται ένα άλλο επίπεδο αφαίρεσης δεδομένων. Αυτή η στρώση μπορεί να συγκριθεί με το Data Marts συστήματα [47], τα οποία είναι μια περιοχή ανασυγκρότησης για δεδομένα που εξυπηρετούν τις ανάγκες ενός συγκεκριμένου τμήματος ή επιχειρηματικής μονάδας. Αυτή η "πηγή δεδομένων" (data pond) είναι ένα υποσύνολο της λίμνης δεδομένων που επικεντρώνεται σε ένα συγκεκριμένο θέμα ή επιχειρησιακό τμήμα, τοποθετημένο στο σκοπό της εργασίας χωρίς πλεονασμό και επομένως εφαρμόζοντας τα στάδια μετασχηματισμού δεδομένων και συρραφής δεδομένων. Η επιλογή και οργάνωση των δεδομένων για κάθε ανάγκη μπορεί να απλοποιήσει την ανάλυση δεδομένων με μεγάλο περιθώριο, ενώ πληροί μια μικρότερη ή πιο συγκεκριμένη απαίτηση. Μια λίμνη δεδομένων μπορεί να απεικονιστεί ως ένα μέρος καθαρού νερού, και μέρη αυτού μπορεί να χαρακτηριστούν ως data pond, όπου ο χρήστης "βουτάει" για να επικεντρωθεί η ανάλυσή του (Εικόνα 3).



Εικόνα 3 Διαφορετικές μορφές αποθετηρίων δεδομένων

Μέσω της συνεχούς βελτίωσης των δεδομένων και των λιμνών δεδομένων και του data pond, ένα σύστημα μπορεί να αξιοποιήσει βέλτιστα και αποτελεσματικά τις παρεχόμενες εισροές από εξωτερικά ή εσωτερικά περιβάλλοντα. Από τη μία πλευρά, αυτή η έννοια οδηγεί τις επιχειρηματικές διαδικασίες να παράγουν ταχύτερα αποτελέσματα. Από την άλλη πλευρά, μπορούν να συναχθούν ολοκληρωμένα και ακριβή συμπεράσματα, δεδομένου ότι ο "θόρυβος δεδομένων" μειώνεται στο ελάχιστο.

ΚΕΦΑΛΑΙΟ 3

Αγορές υπηρεσιοστρεφών εφαρμογών

Σε ένα όλο και αυξανόμενο περιβάλλον τεχνολογικών εξελίξεων στον τομέα του υπολογιστικού νέφους, η έννοια των XaaS (Τα Πάντα Ως Υπηρεσία) [48] [49] είναι σήμερα όλο και πιο συχνή στους παρόχους υπολογιστικού νέφους. Μέσα από τον αφηρημένο ορισμό του XaaS χαρακτηρίζουν και μετατρέπουν την εμπορία και την προμήθεια οποιουδήποτε τύπου ICT σε προϊόντα και υπηρεσίες που παρέχονται μέσα από το υπολογιστικό νέφος. Προς αυτή την κατεύθυνση, μπορούν να αναγνωριστούν πολλοί διαφορετικοί τύποι παρόχων, οι οποίοι προσφέρουν μια ποικιλία υπηρεσιών που βασίζονται στην πλήρη στοίβα του υπολογιστικού νέφους αλλά και σε συγκεκριμένα επίπεδα της στοίβας αυτής. Η διαφοροποίηση γίνεται σύμφωνα με το μοντέλο SPI [50], όπου τα πιο ευρέως αποδεκτά είναι αυτά στο λογισμικό (SaaS), την πλατφόρμα (PaaS) και την υποδομή (IaaS) ως υπηρεσία. Διάφορα χαρακτηριστικά, όπως η εικονική διαμόρφωση του υλικού, η ταχεία εξυπηρέτηση, η δυνατότητα κλιμάκωσης, η ελαστικότητα, η λογιστική κατανομή και τα μοντέλα κατανομής κόστους επιτρέπουν στο υπολογιστικό νέφος να προσαρμόζει αποτελεσματικά τους πόρους για να προσαρμόζονται στις δυναμικές απαιτήσεις των χρηστών του διαδικτύου.

3.1 Αγορές με υπηρεσίες εγκατεστημένες στο νέφος

Στην νέα αυτή εποχή, οι απαιτήσεις των χρηστών και των παρόχων τείνουν να ξεπεράσουν τα προαναφερθέντα τεχνικά χαρακτηριστικά και επιχειρούν να εισέλθουν στον επιχειρηματικό χώρο [51]. Το ενδιαφέρον ερώτημα είναι 'εάν και με ποιον τρόπο οι υπηρεσίες θα αποτελέσουν αντικείμενο διαπραγμάτευσης στο μέλλον' [52]. Στο κόσμο των πληροφοριών και υπηρεσιών στον οποίο πηγάζουν αλλά και δρουν διαφορετικού τύπου χρήστες, απαιτούνται περιβάλλοντα που θα έχουν χαμηλό επίπεδο γνώσης για την απρόσκοπτη ενσωμάτωση των επιχειρησιακών και τεχνικών πτυχών στη διαδικασία παροχής υπηρεσιών. Προς αυτή την κατεύθυνση, οι αναδυόμενες αγορές στοχεύουν στο να επιτρέψουν την παροχή «προϊόντων» ως δυνητικά σύνθετων προσφερόμενων υπηρεσιών που μπορεί να ενσωματώνουν υπηρεσίες από διαφορετικούς παρόχους εντός και μεταξύ των στρωμάτων στοίβας των υπηρεσιών υπολογιστικού νέφους [53]. Οι αγορές αποτελούν ένα κομβικό σημείο για διάφορους παράγοντες που στηρίζουν τον πλήρη κύκλο ζωής των υπηρεσιών (χρονο-σχεδιασμός, ανάλυση και σχεδιασμός συστήματος, ανάπτυξη και δοκιμές αυτού, προμήθεια, ανάπτυξη, ανακάλυψη, σύνθεση, εκτέλεση και παρακολούθηση [54]). Επιπλέον, οι αγορές (marketplaces) υποστηρίζουν προηγμένες δυνατότητες τιμολόγησης και χρέωσης [55], εξετάζοντας και διαχειρίζοντας διάφορους και ενδεχομένως ποικίλους επιχειρηματικούς όρους και προϋποθέσεις (δηλ. τιμή, κατανομή εσόδων, προώθηση). Παρόλο που ο όρος "Marketplace" είναι επιχειρηματικός όρος, τα περιβάλλοντα των αγορών υπολογιστικού νέφους αντιμετωπίζουν αρκετές τεχνικές προκλήσεις εκτός από τις επιχειρηματικές.

Η υποστήριξη της ετερογένειας των υπηρεσιών, έχει επισημανθεί ως μία από τις τρέχουσες αρχές σχεδιασμού του διαδικτύου που θα πρέπει να παραμείνει στη μελλοντική αρχιτεκτονική του [56]. Προς αυτή την κατεύθυνση, οι αγορές επιτρέπουν την παροχή προϊόντων που αποτελούνται από δυναμικά ετερογενείς υπηρεσίες και από διαφορετικούς παρόχους. Επιπλέον, μια θεμελιώδης αρχή του νέου σχεδιασμού της αρχιτεκτονικής του μέλλοντος στο διαδίκτυο αναφέρεται στη διάδοση και την ανταλλαγή πληροφοριών που οδηγεί στη λεγόμενη κατάσταση "All-Win" [57]. Όπως περιγράφεται στην αντίστοιχη έκθεση, το μελλοντικό διαδίκτυο θα πρέπει να επιτρέπει «την εκ των υστέρων ανταλλαγή των πληροφοριών μεταξύ επιπέδων και παραγόντων, γεγονός που υποδηλώνει ότι οι διάφοροι ενδιαφερόμενοι θα πρέπει να μπορούν να παρέχουν σε άλλους πληροφορίες σχετικά με πιθανές επιλογές και τις προτιμήσεις τους». Η διάδοση των πληροφοριών που ενεργοποιείται μέσω των αγορών αφορά όχι μόνο στη συγκέντρωση, τη συλλογή και την αξιολόγηση, αλλά και στη λήψη αποφάσεων σε ένα αρκετά ετερογενές από μεριάς συμφερόντων περιβάλλον. Οι μηχανισμοί της αγοράς επιτρέπουν τη λήψη αποφάσεων μέσω προσεγγίσεων που ερμηνεύουν τις επιχειρηματικές και τεχνικές απαιτήσεις υψηλού επιπέδου και παρέχουν προτάσεις επιλογής και συστάσεις για συνθέσεις προϊόντων. Επιπροσθέτως, οι αγορές υπερβαίνουν την επιλογή των υπηρεσιών που βασίζονται στο SLA [58] ή την ανακάλυψη υπηρεσιών με βάση την πρόβλεψη και την ανάλυση των επιδόσεων. Οι αγορές χρησιμοποιούν τεχνικές για τα μοντέλα τιμολόγησης και την κοινή χρήση εσόδων για την εξυγίανση είτε την οικονομική αναδιάρθρωση των επιχειρήσεων μέσω της ανάλυσης των υφιστάμενων εξαρτήσεων μεταξύ των παρόχων των αντίστοιχων προϊόντων.

Οι αγορές υπηρεσιών βασισμένες στο υπολογιστικό νέφος (Cloud Marketplaces), ως περιβάλλοντα όπου όλοι οι ενδιαφερόμενοι που εμπλέκονται στον κύκλο ζωής των υπηρεσιών συμμετέχουν για να ικανοποιήσουν τους επιχειρηματικούς τους στόχους, δεν είναι πολύ συνηθισμένα, ενώ οι λίγες υπάρχουσες προσεγγίσεις στερούνται βασικής λειτουργικότητας, όπως η υποστήριξη σύνθετων προσφορών ή η αυτοματοποιημένη ανάλυση και διασταύρωση των απαιτήσεων των πελατών. Ως εκ τούτου, η ανάγκη για νέου τύπου αγορές για υπηρεσίες cloud και πόρους που ακολουθούν την τεχνική πρόοδο των τεχνολογιών υπολογιστικού νέφους είναι αισθητή. Αυτά τα “one-stop-shop” των προσφορών σύννεφου θα βελτιώσουν την αποτελεσματικότητα των υπηρεσιών ως πραγματικά επιχειρηματικά οικοσυστήματα και θα αξιοποιήσουν επίσης την ευρεία υιοθέτηση των τεχνολογιών Cloud Computing.

Οι πάροχοι χρησιμοποιούν έτοιμα προϊόντα υπηρεσιών είτε από άλλους χρήστες είτε από εταιρίες (Off-the-shelf), για να καθορίσουν νέες παροχές και πακέτα προϊόντων καθώς και πληροφορίες σχετικά με τις επιχειρήσεις και τους στόχους σε επίπεδο υπηρεσιών. Ταυτόχρονα οι χρήστες ενδέχεται να θέσουν συγκεκριμένα αιτήματα υπηρεσιών και, συγκεκριμένα, συμβόλαια/συμβάσεις παροχής υπηρεσιών στους παρόχους. Ωστόσο, για να είναι ακόμα πιο εκτενές το σύνολο παρεχόμενων υπηρεσιών και ιδιοτήτων, οι παρεχόμενες υπηρεσίες δεν πρέπει να περιορίζονται μόνο στη διαπαφή μεταξύ του προαναφερόμενου πάροχου και του χρήστη -που το μέγιστο που μπορεί να προσφέρει είναι μια συνάθροιση και σύνθεση προϊόντων- αλλά να επικεντρώνονται στην αλληλεπίδραση μεταξύ τεχνικών και επιχειρηματικών πτυχών για την παροχή υπηρεσιών

που αντιμετωπίζουν τις προκλήσεις που σχετίζονται με τα μοντέλα τιμολόγησης σε καταναεμημένα περιβάλλοντα [59] [60]. Ένα πρότυπο marketplace θα έπρεπε να ενσωματώνει έναν προσομοιωτή τιμολόγησης για τα προσφερόμενα προϊόντα, επιτρέποντας στους παρόχους να αξιολογούν το μοντέλο τιμολόγησης συνοδευόμενο από το προϊόν και να το συνδέει με τα (τεχνικά και επιχειρηματικά) χαρακτηριστικά του. Σύμφωνα με τη διαδικασία προσομοίωσης, το marketplace θα πρέπει να προτείνει συγκεκριμένα μοντέλα τιμολόγησης και να υλοποιεί το κάθε προϊόν με δυναμικό τρόπο. Το τελευταίο επιτρέπει την προσαρμογή του προϊόντος, δεδομένου ότι η συνάθροιση υπηρεσιών και η ανάπτυξη τους μπορούν να ρυθμιστούν με βάση τις απαιτήσεις των χρηστών στα πλαίσια του σημερινού περιβάλλοντος της αγοράς (δηλαδή δυναμικές προσφορές από παρόχους). Επιπλέον, ένα marketplace επιτρέπει την αυτοματοποιημένη κατανομή των εσόδων στους διάφορους παρόχους με βάση τις συνεισφορές τους στο προσφερόμενο προϊόν, παρέχοντας παράλληλα στους τελικούς χρήστες μια λύση τελικού αποτελέσματος για την επιλογή προϊόντων με βάση τις τεχνικές και τα επιχειρηματικά κριτήρια, τα οποία βρίσκονται στη συνέχεια και εξετάζονται από τις λειτουργίες της αγοράς κατά τη διάρκεια της διαδικασίας σύναψης συμβάσεων και προβλέψεων.

Οι αλγόριθμοι και οι λειτουργίες πρέπει να ακολουθούν τη νέα φιλοσοφία που επιτάσσει η νέα επιχειρηματική φιλοσοφία δομής υπηρεσιών. Αυτό πρέπει να γίνει μέσα από ένα αρχιτεκτονικό σχέδιο προσανατολισμένο στην υπηρεσία αυτή καθαυτή που επιτρέπει την ενσωμάτωσή του σε οποιοδήποτε περιβάλλον νέφους. Ταυτόχρονα θα πρέπει να προνοήσει ο δημιουργός αυτής της αρχιτεκτονικής ότι ίσως αυτό το πακέτο υπηρεσιών

μπορεί να χρησιμοποιηθεί σαν υπο-κομμάτι άλλου προγράμματος είτε υπηρεσίας. Αυτή η λογική της ενθυλάκωσης σε επίπεδο component είναι κάτι που εμφανίζεται σαν αποτέλεσμα της όλο και πιο συχνής χρήσης υπηρεσιοστρεφών υπηρεσιών στο Cloud Computing.

Ένας τελευταίος παράγοντας που θα πρέπει να λογιστεί είναι αυτός της απόδοσης της υπηρεσίας/component αφού μπορεί να έχει διαφορετούμενο ρόλο ανάλογα με τη χρήση της. Είναι δυνατό δηλαδή να δημιουργείται η ανάγκη ελέγχου απόδοσης ώστε να είναι ξεκάθαρος ο παράγοντας απόδοσης αλλά, συνάμα, μπορεί και ο παράγοντας της καθυστέρησης να είναι σημαντικός. Αυτός είναι και ο μοναδικός τρόπος να γίνει αμοιβαία αποδεκτό και ελέγξιμο το κατά πόσο είναι υπαίτιο το σύστημα για τη μη παροχή υπηρεσιών (DoS) είτε για τη καθυστέρηση απόκρισης σε ένα περιβάλλον πολλαπλών υπηρεσιών. Έτσι μπορεί να χρησιμοποιηθεί αυτή η ανάλυση ως αντικείμενο για επιβολή είτε αποτροπή κάποιου προστίμου.

Υπάρχουν διάφορες προσεγγίσεις που δείχνουν τη σημασία των διαμεσολαβητών στις ηλεκτρονικές αγορές [61] [62]. Ακόμη και από τα παλαιότερα υπολογιστικά περιβάλλοντα του δικτύου και από τις πρώιμες εφαρμογές του cloud, τα marketplaces διαδραματίζουν σημαντικό ρόλο, ειδικά στη μετάβαση από το ερευνητικό στο καθαρά επιχειρηματικό οικοσύστημα [63] [64] [65]. Οι αγορές αυτές (marketplaces) αποσκοπούν στο να επιτρέπουν την παροχή υπηρεσιών με τη πρόσβαση και τη παρουσίαση των προϊόντων τους με δομημένο τρόπο ως προς τη διαπραγμάτευση των τιμών, τον ορισμό των συμβάσεων (SLAs) και την προσαρμογή που μπορεί να γίνει σε αυτά, καθώς επίσης

και τη τιμολογιακή πολιτική τους [66] [67]. Αυτά τα περιβάλλοντα είναι συνήθως εμπλουτισμένα με μηχανισμό συστάσεων καλύτερων επιλογών για την αποτελεσματικότερη επιλογή υπηρεσιών [68]. Επίσης μια ακόμα κατηγορία χαρακτηριστικών είναι αυτή της διαχείρισης των διαπραγματευτικών συμβάσεων SLA καθώς και των παραμέτρων που έχουν να κάνουν με τη ποιότητα της υπηρεσίας QoS, προκειμένου να διασφαλιστεί ότι οι πόροι διατίθενται εντός των περιορισμών ποιότητας, χρόνου και προϋπολογισμού που θέτουν οι πελάτες [69] [70].

Με βάση τα ανωτέρω, οι ηλεκτρονικές αγορές υποστηρίζουν πέντε βασικές φάσεις (δηλαδή αυτές της παροχής πληροφοριών, διαπραγμάτευσης και ρύθμισης των τιμών, σύναψης συμβάσεων, ανάλυσης και τέλος του διακανονισμού με το πελάτη) και εξυπηρετούν τις ανάγκες τριών κύριων ρόλων, αυτού του παρόχου υπηρεσιών, των τελικών χρηστών και των παρόχων αγορών. Ένα νευραλγικό κομμάτι από αυτό το μοντέλο είναι ο προσανατολισμός του στις ηλεκτρονικές αγορές του τομέα των υπηρεσιών. Για το λόγο αυτό θα πρέπει να ακολουθείται η φιλοσοφία της υπηρεσιοστρέφειας η οποία παρέχει τη δυνατότητα να μην υπάρχουν περιορισμοί στην τοποθεσία εγκατάστασής του, όπως συμβαίνει και με κάθε υπηρεσία που βασίζεται στο σύννεφο [71].

Ο Πίνακας 1 συνοψίζει την αρχική αξιολόγηση των πιο εξελιγμένων, υφιστάμενων ηλεκτρονικών αγορών για τις υπηρεσίες που βασίζονται σε αρχιτεκτονικές cloud. Πρέπει να σημειωθεί ότι οι λειτουργίες κάθε φάσης (που παρατίθενται στον πίνακα) θα χρησιμεύσουν επίσης ως χαρακτηριστικά και κριτήρια αξιολόγησης των ηλεκτρονικών αγορών.

Marketplace		Windows Azure Marketplace	Google Apps Marketplace	AppExchange on Force.com	Google Play Store	SuiteApp.com	Zoho
		Functionality					
1. Information phase							
1.1	Product Definition	+	+	+	+	+	+
1.2	Product Search	+	+	+	+	+	+
1.3	User Bid for Missing Applications	-	+	+	-	-	+
1.4	Related Products	+	+	+	+	+	+
1.5	Recommendation	-	-	+	+	-	-
1.6	Advertisement	-	-	+	+	-	-
1.7	Community Rating & Comments	+	+	+	+	+	+
1.8	Social Analysis	-	-	-	+	-	-
2. Negotiation phase							
2.1	Bid to Product	-	-	+	-	-	+
2.2	Product Resolution	+	+	+	-	+	-
2.3	Product Specification	+	+	+	+	+	+
2.4	Product Customizing	-	+	+	-	-	-
2.5	Composite Resolution	-	+	+	-	-	-
2.6	Real-time Resolution	-	-	+	-	-	-
2.7	Profile-based Resolution	-	-	-	+	-	-
2.8	Basket Management	-	+	+	+	-	-
3. Contracting phase							
3.1	Contract Management	-	+	+	+	-	-
4. Settlement phase							
4.1	Delivery Support	+	+	+	+	+	+
4.2	Payment Support	+	+	+	+	+	+
4.3	Rating & Charring	-	+	-	-	-	-
4.4	Settlement & Revenue Sharing	-	-	-	-	-	-
5. Analytics							
5.1	Reporting on Products	-	+	+	+	-	-
5.2	Competitive Products Rankings	-	-	+	-	-	-
5.3	Business Model Simulation	-	-	-	-	-	-

Πίνακας 1 Σύγκριση ηλεκτρονικών αγορών

Όσον αφορά την πρώτη ενημερωτική φάση γίνεται σαφές ότι όλες οι βασικές λειτουργίες περιήγησης υποστηρίζονται από τον εκάστοτε πάροχο που διαθέτει πλατφόρμα αγορών. Για παράδειγμα οι λειτουργίες που επιτρέπουν στους χρήστες να αναζητούν τον κατάλογο υπηρεσιών για προϊόντα που ταιριάζουν στις ανάγκες τους. Τα ακόλουθα χαρακτηριστικά είναι αυτά που προσφέρουν την ανάλυση που είναι αναγκαία για αυτό το σκοπό, Ορισμός Προϊόντος (1.1), Αναζήτηση Προϊόντος (1.2) και Συναφή Προϊόντα (1.4). Ωστόσο, όσον αφορά τη διαφήμιση (1.6), τα κλασικά banner που εμφανίζονται στο εσωτερικό της διεπαφής του marketplace για να προωθήσουν μια συγκεκριμένη υπηρεσία εφαρμόζονται μόνο από το AppExchange αλλά ακόμη και σε αυτή τη συγκεκριμένη περίπτωση, οι πάροχοι υπηρεσιών μπορούν να διαφημίζονται μόνο στη σελίδα της αγοράς όπου περιγράφεται το δικό τους προϊόν.

Συνεχίζοντας στη δεύτερη φάση τα χαρακτηριστικά μιας τυπικής αγοράς συναλλαγών που υποστηρίζει τις διαπραγματεύσεις μεταξύ πελατών και παρόχων, υλοποιείται από τους περισσότερους παρόχους. Ωστόσο, αυτό δεν συμβαίνει στο marketplace της Microsoft.

Χαρακτηριστικά που σχετίζονται με τη σύναψη συμβάσεων (χαρακτηριστικό γνώρισμα 3.1) καθώς και τα βασικά βήματα της φάσης διευθέτησης (χαρακτηριστικά 4.1 και 4.2) είναι οι κύριες ενέργειες που απαιτούνται σε πολύ απλές περιπτώσεις, όπως η διαδικασία παραγγελίας κλασικής ιστοσελίδας. Όσον αφορά όμως αυτές τις βασικές ανάγκες, η εμπορία των υπηρεσιών αντί των αγαθών δεν έχει μεγάλη διαφορά. Δηλαδή, πρώτα ο πελάτης επιλέγει την ατομική υπηρεσία ή μια σύνθεση υπηρεσιών στο καλάθι

αγορών του (χαρακτηριστικό 2.5), και μετά αναθεωρούνται οι λεπτομέρειες και οι όροι της σύμβασης (χαρακτηριστικά 3.1). Μετά από αυτό το βήμα η παραγγελία μπορεί να πραγματοποιηθεί.

Καθεμία από τις αγορές που υποστηρίζουν αυτά τα χαρακτηριστικά μπορεί να ενεργοποιήσει την παράδοση του προϊόντος (χαρακτηριστικό 4.1) καθώς και να εξασφαλίσει την ορθή πληρωμή των υπηρεσιών για τον πελάτη. Τα αναλυτικά στοιχεία του marketplace, όπως το προϊόν αναφοράς (χαρακτηριστικό 5.1), διατίθενται σε τρεις από τις πλατφόρμες σύγκρισης. Αυτές υπολογίζουν ένα είδος εισοδηματικής έκθεσης και παραδίδουν τα στοιχεία στους Παρόχους Υπηρεσιών. Στην περίπτωση των Google Apps, του Google Play Store και του AppExchange, αυτά τα στατιστικά στοιχεία βασίζονται κυρίως στον αριθμό των στιγμιότυπων που πωλήθηκαν. Αυτό σημαίνει ότι τα marketplaces παρακολουθούν τις συναλλαγές που πραγματοποιούνται. Αναλύοντας λίγο περισσότερο αυτό το κομμάτι, η κατάταξη των ανταγωνιστικών προϊόντων (5.2), υλοποιούνται μόνο από το Google Play Store, το οποίο καθορίζει μια κατάταξη που συγκρίνει παρόμοιες εφαρμογές και προβάλλει τους δυνητικούς πελάτες που θα μπορούσε να έχει.

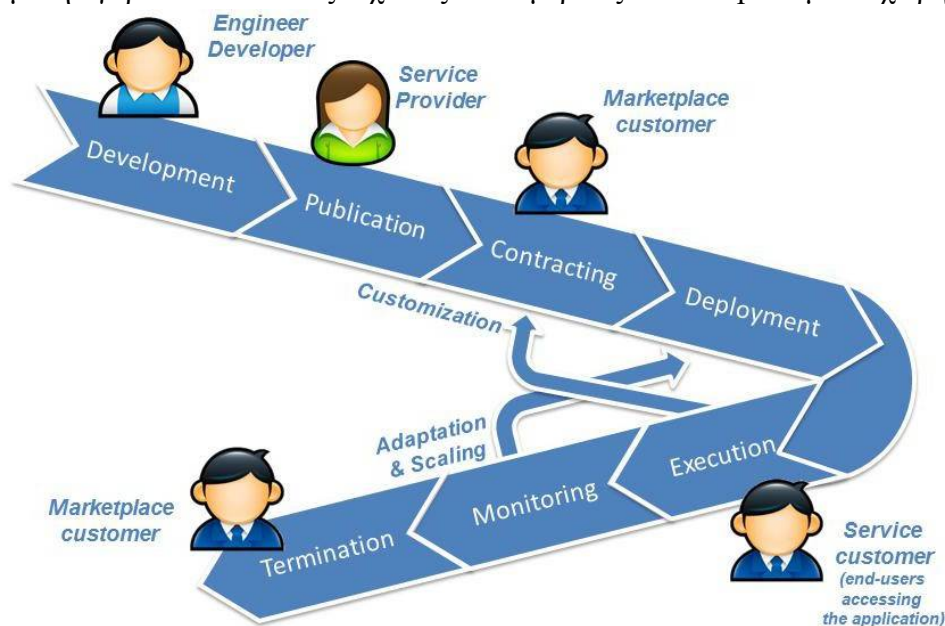
Οι προηγμένοι ηλεκτρονικοί κατάλογοι υπηρεσιών και εφαρμογών, που επιτρέπουν τη δομημένη καταχώρηση των προϊόντων, θεωρούνται ως μια άλλη ιδέα εφαρμογής της αγοράς - που υιοθετήθηκε από τα Windows Azure Marketplace, το SuiteApp.com και το Zoho. Ενώ αυτοί οι κατάλογοι καλύπτουν τη φάση πληροφόρησης των αγορών, δεν προσφέρουν λειτουργίες που απαιτούνται από τις άλλες φάσεις (δηλαδή διαπραγμάτευση, σύναψη συμβάσεων και διακανονισμός), οι οποίες αφέθηκαν στους

παρόχους υπηρεσιών. Παρότι το Google Play Store υποστηρίζει όλες τις φάσεις, πρέπει να σημειωθεί ότι οι υπηρεσίες που φιλοξενούνται και διαφημίζονται δεν είναι υπηρεσίες που βασίζονται στο υπολογιστικό σύννεφο, δεδομένου ότι εγκαθίστανται τοπικά, ενώ δεν μπορούν να ομαδοποιηθούν και να συμπεριληφθούν σε νέες υπηρεσίες. Μια παρόμοια λύση παρέχεται από το Google Apps Marketplace που εκμεταλλεύεται διάφορα εργαλεία Google (π.χ. Checkout, Analytics) για να καλύψει τις διάφορες φάσεις μιας συναλλαγής αγοράς. Το ίδιο το Google Apps Marketplace δίδει έμφαση σε βελτιωμένες λειτουργίες αναζήτησης για τους καταναλωτές και αναθέτει σε άλλα στάδια της συναλλαγής στην αγορά συμπληρωματικά προϊόντα της Google. Το Salesforce.com έχει αναπτύξει το AppExchange, το οποίο είναι το πιο προηγμένο marketplace σε σύγκριση με τα προηγούμενα, καλύπτοντας επίσης υπηρεσίες που βασίζονται σε Cloud Computing.

Οπότε μια πρότυπη πλατφόρμα για την αναζήτηση υπηρεσιών θα πρέπει να αντιμετωπίζει αυτά τα κενά, και ιδιαίτερα την προσομοίωση μοντέλων τιμών, τον δυναμικό προσδιορισμό των τιμών, τα προηγμένα χαρακτηριστικά επιλογής και κατανομής εσόδων, τα οποία δεν προσφέρονται από οποιαδήποτε αγορά με τεχνολογίες Cloud Computing. Μια προσέγγιση που θα μπορούσε να λύσει τα προβλήματα που παρουσιάζονται καθώς και να είναι εξατομικευμένη στις απαιτήσεις που υπάρχουν θα πρέπει να έχει σχεδιαστεί με τέτοιο τρόπο ώστε να λαμβάνει υπόψιν όλους τους χρήστες που αλληλοεπιδρούν με αυτή. Οι κύριοι stakeholders που αναγνωρίστηκαν είναι οι ιδιώτες, τα ΜΜΕ και οι επιχειρήσεις. Ωστόσο, ακολουθώντας τις αρχές σχεδιασμού με γνώμονα την εξυπηρέτηση και καθορίζοντας επίπεδα αφαίρεσης για την επικοινωνία με τα υπόλοιπα

στοιχεία της πλατφόρμας, το marketplace μπορεί να συσχετιστεί με οποιαδήποτε σύγχρονη πλατφόρμα cloud ή πάροχο.

Για να είναι πλήρες το μοντέλο ανάλυσης και σχεδιασμού του marketplace θα πρέπει να περιλαμβάνει ολόκληρο τον κύκλο ζωής μιας εφαρμογής σε ένα Cloud Computing περιβάλλον, από την ανάπτυξή της έως την εμπορευματοποίηση και την παρακολούθηση της επίδοσης της, όπως παρουσιάζεται και στην Εικόνα 4. Μετά την ανάπτυξη των κομματιών που αποτελούν τη δομή του λογισμικού που δημιουργούν μια εφαρμογή, θα πρέπει να παρέχει μια γλώσσα και πλαίσιο περιγραφής που ονομάζεται blueprint [72], [73] που επιτρέπει στον προγραμματιστή να περιγράψει τις εξαρτήσεις ή τις απαιτήσεις που έχει η εφαρμογή για να αναπτυχθεί στο Cloud Computing. Η προτεινόμενη αγορά επεκτείνει τις τεχνικές λεπτομέρειες του blueprint με επιχειρηματικές



Εικόνα 4 Κύκλος ζωής υπηρεσιών νέφους

παραμέτρους (όπως τα μοντέλα τιμολόγησης), ακολουθώντας τις προδιαγραφές του

προϊόντος για να εμπορευματοποιήσει την εφαρμογή/υπηρεσία και για να δημιουργήσει προϊόντα υπηρεσιών που θα διαπραγματεύονται και θα συνδυάζονται μέσω του marketplace. Η περιγραφή του προϊόντος αποτελεί αναπόσπαστο μέρος της σύμβασης που υπογράφεται από τον πελάτη, καθώς υπάρχει και σαν αναφορά στο blueprint, προκειμένου να παρέχει στην πλατφόρμα το πλήρες σύνολο των πληροφοριών που απαιτούνται για την αποτελεσματική επιλογή και διαμόρφωση των πόρων του νέφους που θα αναπτυχθούν και θα παραχθούν από την αίτηση. Η αίτηση σε τεχνικό επίπεδο επιτυγχάνεται με τη μετατροπή του blueprint και της συμφωνηθείσας προσφοράς προϊόντος σε ένα πρότυπο που περιγράφει με ακρίβεια είτε το υλικό είτε τις πλατφόρμες που θα χρησιμοποιηθούν (PaaS, IaaS). Αυτό μπορεί να είναι σε μορφή OVF++ [74], μια επέκταση του 4CAAST έργου [75] στο OVF [76] είτε σε οποιαδήποτε άλλη μορφή απαιτεί είτε διευκολύνει τον πάροχο.

Με βάση την ανάλυση των ηλεκτρονικών αγορών και τις δυνατότητες των πλατφόρμων Cloud Computing, εντοπίστηκαν οι βασικοί ρόλοι για μια αγορά Cloud Computing, καθώς και τα συναφή μέρη. Από την *πλευρά των καταναλωτών*, οι ρόλοι που μπορούν να διακριθούν συνιστώνται α) στους τελικούς καταναλωτές, οι οποίοι απλώς καταναλώνουν μια ορισμένη υπηρεσία, η οποία συνήθως βασίζεται σε SaaS, β) τους επιχειρηματικούς πελάτες, οι οποίοι εκμεταλλεύονται τις υπηρεσίες cloud που διατίθενται στην αγορά, όπως για παράδειγμα να αναπτύξουν τις δικές τους παρουσίες εφαρμογών στο cloud, και γ) στις επιχειρήσεις που εκμεταλλεύονται τους πόρους της πλατφόρμας ως ειδική περίπτωση συγκερασμού ετερογενών σύννεφων.

Από την πλευρά του *παρόχου*, μπορούν να διακριθούν μεταξύ α) υποδομής ως παρόχων υπηρεσιών, συνήθως ενισχυμένοι με τις δυνατότητες του δικτύου ως υπηρεσίας, β) ως υπηρεσία (PaaS) και γ) το λογισμικό ως υπηρεσία (SaaS) [77]. Άλλες σημαντικές οντότητες σε αυτό το δίκτυο αξιών είναι οι *πάροχοι λογισμικού (ή προμηθευτές)*, οι οποίοι προσφέρουν το λογισμικό που θα αναπτυχθεί στους πόρους του σύννεφου (εφαρμογές και ενδιάμεσο λογισμικό είτε πλατφόρμα), καθώς και οι *συναθροιστές υπηρεσιών* με το ρόλο τόσο του καταναλωτή όσο και του παρόχου, συνδυάζοντας προσφορές διαφορετικών τύπων σε νέα σύνθετα προϊόντα που θα αποτελέσουν το εμπόρευμα στη συνέχεια, στο marketplace.

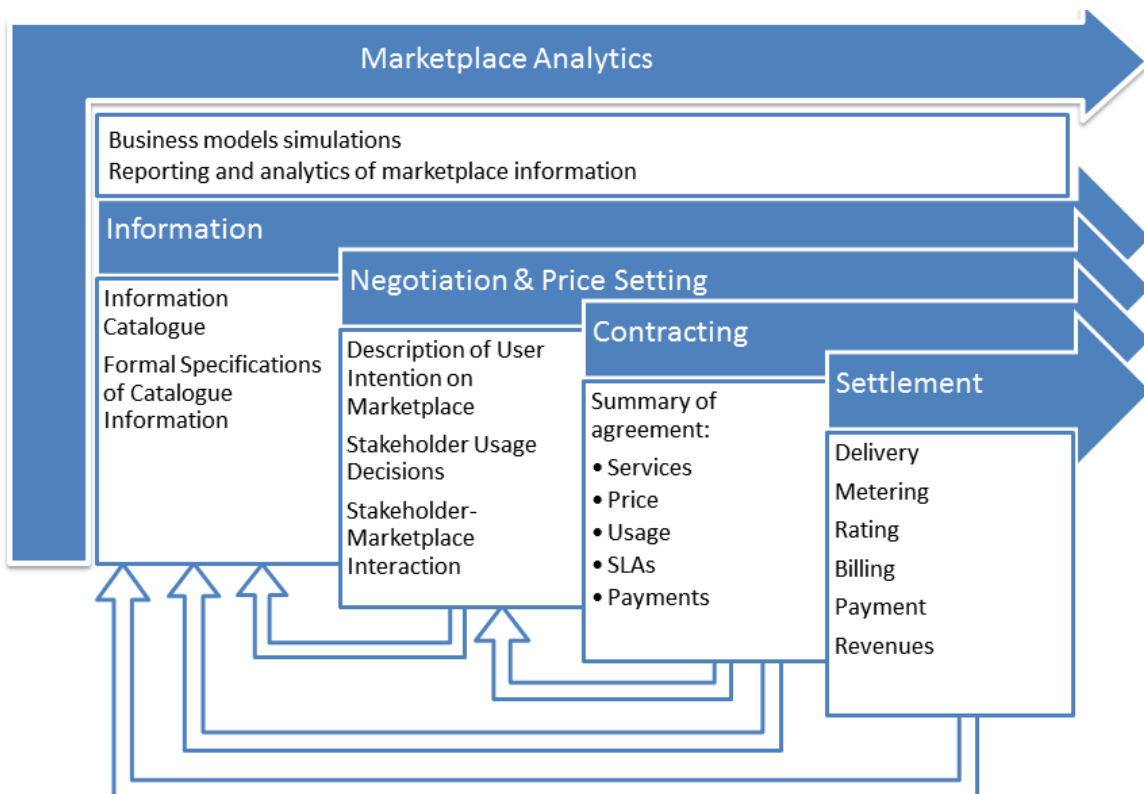
Ακόμη και αν ένα marketplace για αγορά υπηρεσιών cloud computing μοιράζεται πολλά χαρακτηριστικά με τις γενικές ηλεκτρονικές αγορές, περιλαμβάνει ορισμένα μοναδικά γνωρίσματα τα οποία είναι μέρος ενός σύγχρονου cloud περιβάλλοντος. Αυτά τα οποία ήταν στην πραγματικότητα και τα κίνητρα για την ανάλυση και διαφοροποίηση από τα κλασικά μοντέλα που υπήρχαν. Στις ακόλουθες ενότητες θα αναλυθεί ο τρόπος με τον οποίο το marketplace αντιμετωπίζει αυτές τις προκλήσεις και οι οποίες είναι στην πραγματικότητα οι καινοτομίες που εισάγονται με την υλοποίηση αυτών των χαρακτηριστικών. Πρέπει να σημειωθεί ότι η κύρια προσπάθεια σε αυτό το marketplace, επικεντρώνεται στις πτυχές των δυναμικών μοντέλων τιμολόγησης, της προσομοίωσης τιμών και της αποτελεσματικής επίλυσης των αιτήσεων. Ωστόσο, ως ενσωματωμένη αγορά cloud, οι τυπικές διεργασίες όπως η σύναψη συμβάσεων και οι πληρωμές

υποστηρίζονται επίσης με τη χρήση ή την επέκταση των λύσεων από έτοιμες επιλογές λογισμικού (off the shelf).

3.2 Αρχές πρότυπου σχεδιασμού

Η προτεινόμενη λύση θα πρέπει να προάγει τη δημιουργία δυναμικών επιχειρηματικών οικοσυστημάτων στα οποία πολλοί πάροχοι υπηρεσιών μπορούν να βρουν ένα δίκαιο περιβάλλον για την ανάπτυξη και την παροχή νέων υπηρεσιών [78]. Η υποστήριξη ευέλικτων μοντέλων εσόδων θα βοηθήσει τους παρόχους υπολογιστικού νέφους να δημιουργήσουν έσοδα από την προσπάθεια δημιουργίας και διατήρησης της πλατφόρμας, ενώ οι πάροχοι υπηρεσιών μπορούν να λειτουργούν και να προσφέρουν τις υπηρεσίες τους με μηδενική αρχική κεφαλαιακή δαπάνη (CAPEX) [79].

Το cloud marketplace θα πρέπει να υποστηρίζει τη διαπραγμάτευση όλων των διαφορετικών τύπων υπηρεσιών με ενιαίο τρόπο · ένα «κατάστημα με μία στάση» για όλες τις παροχές υπηρεσιών: SaaS, PaaS, IaaS, NaaS ή γενικά XaaS. Μια λεπτομερής διασύνδεση του επιπέδου δόμησης των υπηρεσιών που υποστηρίζει την προδιαγραφή των εμπορικών προσφορών για οποιονδήποτε τύπο υπηρεσίας, χρησιμοποιώντας τα πιο κατάλληλα μοντέλα εσόδων για κάθε υπηρεσία, απλοποιεί τη διαδικασία δημιουργίας των εφαρμογών. Το marketplace μπορεί επίσης να εφαρμόσει συγκεκριμένες πολιτικές και περιορισμούς για να αποφασίσει ποιες πλατφόρμες και υπηρεσίες θα χρησιμοποιηθούν στην τελική ανάπτυξη υπηρεσιών, με βάση τις επιχειρηματικές απαιτήσεις τόσο του παρόχου υπηρεσιών όσο και του πελάτη.



Εικόνα 5 Κύριες φάσεις και διεργασίες των ηλεκτρονικών αγορών

Οι κύριες διεργασίες του marketplace χωρίζονται σε τέσσερις διαφορετικές φάσεις όπως παρουσιάζονται στην Εικόνα 5 και όπως περιγράφεται στο [80]. Στην πρώτη φάση, την λεγόμενη φάση των πληροφοριών, οι παράγοντες της αγοράς πρέπει να ανταλλάξουν πληροφορίες. Κατά τη διάρκεια της δεύτερης φάσης, την λεγόμενη φάση διαπραγμάτευσης και καθορισμού των τιμών, αφού ο καταναλωτής έχει επιλέξει ορισμένες υπηρεσίες, λαμβάνει χώρα η εξομάλυνση και η δόμηση των τιμών. Ο καταναλωτής

επιτρέπεται να ορίσει συγκεκριμένους περιορισμούς για το αίτημά του ή να επιλέξει από ένα σύνολο επιλογών προσαρμογής, προκαθορισμένες από τον πάροχο υπηρεσιών, για ένα συγκεκριμένο προϊόν. Το marketplace θα προσδιορίσει οποιοσδήποτε επιχειρηματικές εξαρτήσεις και θα προτείνει μια επιλογή και συνδυασμό υπηρεσιών και πόρων που ικανοποιούν τους περιορισμούς που καθορίζονται από τον καταναλωτή. Στην τρίτη φάση (σύμβαση), όταν ο πάροχος και ο καταναλωτής έχουν συμφωνήσει για ένα συγκεκριμένο σύνολο όρων προϊόντος και επιχείρησης, πρέπει να οριστεί μια νομικά δεσμευτική σύμβαση πριν από την παράδοση των υπηρεσιών. Στη φάση του διακανονισμού, οι υπηρεσίες που εκτελούνται στο cloud παρέχονται, παρακολουθούνται και λογιστικοποιούνται σύμφωνα με τις υπογεγραμμένες συμβάσεις. Δεδομένου ότι οι υπηρεσίες μπορεί να είναι το αποτέλεσμα μιας διαδικασίας ανάλυσης και σύνθεσης, ο μηχανισμός διαμοιρασμού εσόδων της αγοράς απλοποιεί τη λογιστική διαδικασία που επιτρέπει στους παρόχους υπηρεσιών να προωθήσουν τη χρήση της προσφοράς τους στις τελικές εφαρμογές και υπηρεσίες. Οι τέσσερις κύριες φάσεις του marketplace βελτιώνονται επίσης με τη λειτουργική ανάλυση των κατά μέρους ιστορικών δεδομένων και καταγραφών(logs) του marketplace. Η ανάλυση είναι μια οριζόντια διαδικασία που συναθροίζει πληροφορίες από διάφορες πηγές (λειτουργικές, επιχειρηματικές, κοινωνικές, σχόλια χρηστών), προκειμένου να αποκτήσει γνώσεις για τη βελτίωση των επιχειρηματικών όρων και προϋποθέσεων για τα υπάρχοντα ή νέα προϊόντα και επιχειρηματικά μοντέλα.

3.3 Μοντέλα κέρδους και διαχείρισης τιμών

Ανάλογα με το επιχειρηματικό μοντέλο, ένας πάροχος υπηρεσιών μπορεί να επιλέξει ένα ή περισσότερα μοντέλα τιμών για καθεμία από τις προσφορές που παρέχονται. Στη γενική περίπτωση, η τιμή είναι μια συνάρτηση που εξαρτάται από μία ή περισσότερες παραμέτρους. Για παράδειγμα, μια τιμή μπορεί να είναι ο συνδυασμός μιας σταθερής χρέωσης συνδρομής s συν μια χρέωση συναλλαγής Tf , τότε η τιμή $p(x) = s + x * Tf$, όπου x μπορεί να είναι ο αριθμός των κλήσεων εξυπηρέτησης, είναι ο χρόνος που πέρασε κατά την αλληλεπίδραση με την υπηρεσία ή η ποσότητα των πόρων που καταναλώνονται. Ένας πάροχος μπορεί επίσης να επιλέξει να ορίσει διάφορες εναλλακτικές επιλογές μοντέλων τιμών για την ίδια υπηρεσία, προκειμένου να ανταποκριθεί στις διαφορετικές απαιτήσεις των πελατών. Επίσης, ένα μοντέλο τιμών μπορεί να προσαρμοστεί κατά τη φάση των διαπραγματεύσεων μεταξύ παρόχου υπηρεσιών και καταναλωτή υπηρεσιών. Μια βασική λειτουργικότητα του marketplace είναι ότι επιτρέπει στον πάροχο υπηρεσιών να καθορίζει κατάλληλα μοντέλα τιμών και ότι παρέχει υποστήριξη σε αυτή την εργασία, π.χ. με προκαθορισμένα μοντέλα τιμών βέλτιστης πρακτικής ή με προσομοίωση και ανάλυση. Το μοντέλο τιμής μορφοποιείται και μπορεί να αξιοποιηθεί, π.χ., να επιλέγει αυτόματα τις υπηρεσίες με βάση τις προτιμήσεις τιμολόγησης του χρήστη ή να υπολογίζει τις πληρωμές που πρέπει να πραγματοποιούνται από τους καταναλωτές υπηρεσιών με βάση την πραγματική κατανάλωση των υπηρεσιών τους.

Το προτεινόμενο πλαίσιο τιμολόγησης, η εφαρμογή του οποίου έχει περιγραφεί προηγουμένως [78], προβλέπει επίσης ότι αρκετές υπάρχουσες προσφορές υπηρεσιών

μπορούν να συνδυαστούν σε νέες, σύνθετες, είτε από έναν από τους αρχικούς παρόχους υπηρεσιών είτε από έναν άλλο (ο οποίος θα είναι στη συνέχεια είτε ένας απλός μεταπωλητής είτε θα παρέχει μια υπηρεσία προστιθέμενης αξίας). Στη γενική περίπτωση, κάθε υπηρεσία μπορεί να βασίζεται σε άλλες υπηρεσίες που αποτελούν δίκτυο υπηρεσιών. Ταυτόχρονα, κάθε υπηρεσία προσφέρεται από έναν ορισμένο πάροχο που μπορεί να είναι ο καταναλωτής ενός άλλου παρόχου. Οι διαδικασίες της αγοράς επιτρέπουν στους παρόχους υπηρεσιών (και στους καταναλωτές) να αναλύουν αυτό το επιχειρηματικό δίκτυο και να υπολογίζουν το συνολικό μοντέλο τιμών για τον καταναλωτή ενός δικτύου υπηρεσιών. Αντιστρόφως, κατά τη διανομή του παραγόμενου εισοδήματος για μια συγκεκριμένη συναλλαγή αγοράς από τον τελικό καταναλωτή σε όλους τους εμπλεκόμενους παρόχους σύμφωνα με το επιχειρηματικό δίκτυο, το προκύπτον σύνθετο μοντέλο επιμερισμού των εσόδων ορίζεται και διαχειρίζεται επίσης από το marketplace.

3.3.1 Μοντελοποίηση οικονομικού μοντέλου

Ένα εργαλείο προσομοίωσης είναι ένα βασικό συστατικό του marketplace, ούτως ώστε οι προαναφερθείσες πτυχές να μπορούν να αξιολογηθούν αποτελεσματικά από τους παρόχους υπηρεσιών ελαχιστοποιώντας τον κίνδυνο λανθασμένων εκτιμήσεων σχετικά με το κόστος και αυξάνοντας την απόδοση της επένδυσής τους σε ένα νέο ή προσαρμοσμένο μοντέλο προϊόντος ή τιμής [81]. Ένας πάροχος μπορεί να επιλέξει και να ορίσει διάφορες εναλλακτικές επιλογές μοντέλων και τιμών για την ίδια υπηρεσία, προκειμένου να ανταποκριθεί στις διαφορετικές απαιτήσεις των πελατών. Το μοντέλο τιμών έχει μεγάλο

αντίκτυπο στο δυνητικό μερίδιο αγοράς, τα έσοδα και το κέρδος μιας προσφοράς υπηρεσιών και η επιλογή του σωστού μοντέλου είναι ένα ζωτικό, αν και δύσκολο, έργο για τους παρόχους υπηρεσιών. Το εργαλείο προσομοίωσης προσφέρει διαφορετικές επιλογές οπτικοποίησης, προσομοίωσης και βελτιστοποίησης. Για σύνθετες υπηρεσίες που απαιτούν τη λειτουργία υπηρεσιών τρίτων, ένα γράφημα εξάρτησης θα πρέπει να εμφανίζεται πλήρες με τα μοντέλα τιμών για κάθε υπηρεσία στο γράφημα, μαζί με την προβλεπόμενη χρήση μιας υπηρεσίας, τα αναμενόμενα έσοδα, το αναμενόμενο κόστος της που απευθύνεται σε τρίτους παρόχους υπηρεσιών και βέβαια το κέρδος. Μία από τις καινοτομίες που εισάγονται σε αυτό το εργαλείο [82] είναι ότι, ως συστατικό του marketplace, η χρήση μιας υπηρεσίας μπορεί είτε να προβλεφθεί βάσει μιας μαθηματικής συνάρτησης χρήσης (που θα καθοριστεί από τον χρήστη) είτε με βάση ιστορικά δεδομένα χρήσης που παρατηρούνται από το marketplace, απλοποιώντας έτσι τη διαδικασία ορισμού του μοντέλου τιμών και που επιτρέπουν την προσαρμογή του δυναμικού μοντέλου σύμφωνα με τις τρέχουσες συνθήκες και τάσεις του marketplace.

3.4 Εύρεση υπηρεσιών, επιλογή και επίλυση τους

Ένα από τα σημαντικότερα χαρακτηριστικά του marketplace είναι η επίλυση των απαιτήσεων των πελατών για κατάλληλα προϊόντα είτε συνθέσεις προϊόντων που θα ενωθούν ώστε να προσφέρουν μια ενοποιημένη νέα υπηρεσία(υψηλού επιπέδου) [+++3]. Οι απαιτήσεις συνδέονται συνήθως με πτυχές του προϊόντος υψηλού επιπέδου, ειδικές για την εφαρμογή και υπό αυτή την έννοια, ένα μεγάλο πρόκριμα της αγοράς θα είναι η

δυνατότητα ανάλυσης και μετάφρασης των απαιτήσεων των πελατών σε ένα σύνολο τεχνικών και επιχειρηματικών παραμέτρων για υπηρεσίες, τεχνολογίες και πόρους. Κατά τη διάρκεια αυτής της απόφασης (τόσο τεχνικής όσο και επιχειρηματικής), προσδιορίζονται κατάλληλοι συμβατικοί όροι που περιλαμβάνουν υποχρεώσεις και πολιτικές για την προμήθεια ενός προϊόντος στο απαιτούμενο επίπεδο ποιότητας υπηρεσίας (QoS).

Μέρος αυτής της διαδικασίας είναι επίσης η επιλογή κατάλληλων μοντέλων τιμών ικανών να υποστηρίζουν αποτελεσματικά τις επιχειρηματικές πτυχές ενός προϊόντος με βάση τους πελάτες, QoS ή/και επιχειρηματικούς περιορισμούς. Η ανάλυση και η επιλογή των προϊόντων και των μοντέλων τιμών βασίζεται σε ένα δυναμικό σύνολο παραμέτρων, λαμβάνοντας υπόψη όχι μόνο το αίτημα του πελάτη αλλά και πληροφορίες όπως το προφίλ χρήστη (συναφείς πληροφορίες που σχετίζονται με έναν χρήστη), την αγορά και την κατάσταση της υποδομής και την εμπειρία σχετικά με την αποτελεσματικότητα των προηγούμενων ψηφισμάτων. Για να υποστηριχθεί αυτό, ένας μηχανισμός επίλυσης επιχειρηματικών δραστηριοτήτων εφαρμόστηκε μετά από ένα αρχιτεκτονικό σχέδιο, ώστε να επιτρέπει στους προγραμματιστές, τους παρόχους και τους φορείς αγοράς να παράγουν προχωρημένους και σύνθετους αλγόριθμους για την επίλυση συγκεκριμένων τύπων αιτημάτων ή προϊόντα. Η δυναμικότητα της διαδικασίας επίλυσης επιτρέπει την περαιτέρω βελτιστοποίηση των τοπικών και εξωγενών προϊόντων και των πόρων της πλατφόρμας, τόσο σε τεχνικό όσο και σε επιχειρηματικό επίπεδο, καθώς και στην ανάπτυξη νέων, εξατομικευμένων προσφορών προϊόντων, συνθέσεις προϊόντων και μοντέλα τιμών τα

οποία μπορούν να αντιμετωπιστούν ομαλά από την αγορά και την υποκείμενη (-ές) πλατφόρμα (-ες) υπολογιστικού νέφους.

Η επιτυχής υλοποίηση του οράματος του υπολογιστικού νέφους και της εύκολης και γρήγορης εύρεσης των κατάλληλων υπηρεσιών που είναι αναγκαίες για την εκάστοτε περίπτωση εξαρτάται σε μεγάλο βαθμό από την εφαρμογή μηχανισμών που προσαρμόζουν την κατανάλωση στους πόρους και αντίστροφα. Σε τελική ανάλυση, αυτή η δυνατότητα θα μπορούσε να διατηρήσει μια βιώσιμη αγορά για το υπολογιστικό νέφος. Η μείωση των τιμών δεν αποδίδεται μόνο στις χαμηλές τιμές των πόρων αλλά στην ορθή χρήση αυτών. Για το λόγο αυτό, η αύξηση της προμήθειας υποτίθεται ότι είναι η εξαίρεση και όχι ο κανόνας σε περιβάλλοντα που δομούνται από χρήση υπηρεσιών από υπολογιστικά νέφη. Καθοριστική για την προσαρμογή της δέσμευσης πόρων στην κατανάλωση είναι η δυνατότητα των εφαρμογών που αναπτύσσονται από τα τμήματα πληροφορικής στις εγκαταστάσεις του υπολογιστικού νέφους να κλιμακώνονται (scalable systems/software) έτσι ώστε να προσφέρουν κατανεμημένους πόρους στο απαιτούμενο επίπεδο QoS. Για να γίνει αυτό το σημείο πιο σαφές, είναι σημαντικό να συνειδητοποιήσουμε ότι η υποδομή νέφους δεν είναι μια μεμονωμένη οντότητα, αλλά διανέμεται σε διαφορετικούς παρόχους υπηρεσιών. Παρά τις προσπάθειες των παρόχων να παρουσιάσουν μια ενοποιημένη υποδομή, αποκρύπτοντας τις πολυπλοκότητες της διαλειτουργικότητας κάτω από τα λιγότερο κοινές Σουίτες Εργαλείων Λογισμικού (API), όπως το Attic [83] και το libcloud [84], είναι αναπόφευκτο ότι οι πόροι είναι κατακερματισμένοι και κατανεμημένοι σε διαφορετικές φυσικές εγκαταστάσεις. Αυτή η διαφορετικότητα μπορεί να σημαίνει για

κάποιους παρόχους διαφορετικό φυσικό server για κάποιους άλλους διαφορετικό δίκτυο και διαφορετικό χώρο η κτήριο εγκατάστασης. Σε περιπτώσεις μεγάλων οργανισμών και πολυεθνικών εταιριών μπορεί αυτή η γεωγραφική κατανομή να φτάσει μέχρι και σε διαφορετική ήπειρο.

Μια εφαρμογή που εκτελείται στο cloud αποτελείται από πολλά instances που εκτελούνται σε διαφορετικές εικονικές μηχανές διασυνδεδεμένες με τα παραδοσιακά πρωτόκολλα ως μέσα επικοινωνίας. Στο υπολογιστικό νέφος, η χρήση των κατάλληλων μεθοδολογιών και API που επιτρέπουν την εύκολη δημιουργία εφαρμογών για cloud είναι απαραίτητη. Τέτοια λύση στο υπολογιστικό νέφος μπορεί να υλοποιηθεί με τροποποίηση και χρήση λογισμικού που θα 'ζει' στο επίπεδο της Πλατφόρμας (PaaS). Αυτό το ενδιάμεσο επίπεδο παρέχει ένα χρήσιμο και πλούσιο interface για τον προγραμματιστή εφαρμογών του cloud ώστε να μπορεί να προσφέρει τον χειρισμό και τη τιμολόγηση των πόρων. Η εφαρμογή του θα πρέπει να εστιάζει στην επιχειρηματική λογική και όχι στις αποκλίσεις των πλατφορμών ή στην εφαρμογή των κανόνων ελαστικότητας και ελαχιστοποίησης κόστους. Η αρχιτεκτονική διαχείριση του ενδιάμεσου επιπέδου θα πρέπει στα πλαίσια της ανάπτυξης των εφαρμογών να κάνει χρήση τεχνικών βελτιστοποίησης πόρων [85] με στόχο την υλοποίηση του προαναφερθέντος οράματος. Την τελευταία δεκαετία, ένας αυξανόμενος αριθμός προϊόντων ενδιάμεσου λογισμικού επέτρεψε τον πολλαπλασιασμό των εταιρικών εφαρμογών που λειτουργούν με την επικοινωνία τους να είναι και εκτός εταιρικού δικτύου και μια αγορά που οι πλατφόρμες ενδιάμεσου λογισμικού του υπολογιστικού νέφους αυξάνεται σταθερά.

Παρά το γεγονός ότι το υπολογιστικό νέφος προσπαθεί να παρουσιαστεί σαν την ιδεατή λύση στο κόσμο της πληροφορικής ως προς τη μεριά της χρησιμότητας, απομένει να βελτιωθεί από την άποψη της ευελιξίας, της ευκολίας χρήσης και της ευελιξίας σε επίπεδο επιχειρήσεων. Οι μεγαλύτερες απαιτήσεις για την υλοποίηση αυτού του οράματος είναι η γρήγορη κάθετη κλιμάκωση των εφαρμογών στους διαθέσιμους πόρους, η οποία είναι εφικτή με τη χρήση "του ραφιού" για τεχνολογίες και λογισμικά πακέτα εικονικής διαμόρφωσης (Commercial off-the-shelf είτε commercially available off-the-shelf (COTS) [86]) καθώς και την παροχή της απαραίτητης λειτουργικότητας για την αγορά αυτών των αιτήσεων για χρήση από τρίτα μέρη. Αν και η εικονική διαμόρφωση επιτρέπει την προώθηση των πόρων του υπολογιστή ως βοηθητικό πρόγραμμα, το υπολογιστικό νέφος από την άλλη παρουσιάζεται σαν ενδιάμεσο λογισμικό που ενεργοποιεί την πληροφορική ως επιχειρηματικό μοντέλο υπηρεσίας. Σε αυτό το επιχειρηματικό μοντέλο, οι εφαρμογές που βασίζονται σε εικονικά διαμορφωμένα μέσα μπορούν να πωλούνται με οικονομικά αποδοτικό χαρακτήρα, να είναι εύκολα προσβάσιμες από οπουδήποτε και από κάθε ενδιαφερόμενο πελάτη μειώνοντας τα τεχνολογικά εμπόδια για την έγκρισή τους. Το PaaS, ως τοπικό ενδιάμεσο λογισμικό σύννεφο, επιτρέπει τη δημιουργία τέτοιων εφαρμογών σε Υποδομή-Ως-Υπηρεσία που καταναλώνουν εικονικά διαμορφωμένους πόρους.

Υπάρχουν διάφορα ζητήματα που διαφοροποιούν την εφαρμογή υπολογιστικού νέφους σε αντίθεση με την εφαρμογή ιδιωτικών εταιρικών κέντρων δεδομένων όπως:

Σωστή χρήση των διαμορφωμένων πόρων: το κόστος είναι μία από τις πιο εξέχουσες κινητήριες δυνάμεις πίσω από το υπολογιστικό νέφος, καθώς διανέμει το

κόστος χρήσης των πόρων σε μια μεγάλη πελατειακή βάση. Ωστόσο, η υπερβολική χρήση των πόρων νέφους μπορεί να επιφέρει πολύ μεγάλα προκαταβολικά κόστη που εξαλείφουν τα οφέλη του υπολογιστικού νέφους. Σε άλλες περιπτώσεις, οι οργανισμοί που υπόκεινται κυρίως σε θέματα ασφαλείας ή πειραματίζονται, δεν είναι έτοιμοι να μετακινήσουν ολόκληρη την υποδομή τους στο cloud, αλλά αντί αυτού επιλέγουν να μετακινήσουν μέρος των λειτουργιών τους σε έναν πάροχο cloud, προκειμένου να απορροφήσουν τις απότομες διακυμάνσεις της ζήτησης ή για να μειώσουν τις διοικητικές-γραφειοκρατικές-διαχειριστικές δαπάνες. Σε τέτοιες περιπτώσεις, η πραγματική δύναμη του υπολογιστικού νέφους είναι ότι έχει δομηθεί με τέτοιο τρόπο ώστε να μπορεί να εκμεταλλευτεί στο έπακρο το ενδιάμεσο επίπεδο δομής και η δυνατότητα να δημιουργεί ελαστικές εφαρμογές με οριζόντιο εύρος σε περιπτώσεις που οργανώνονται οι εφαρμογές υπό διοικητική διαχείριση και κάθετο στους πόρους που προσφέρει ο πάροχος. Ένας ενιαίος μηχανισμός για την αυτόματη κλιμάκωση είναι απαραίτητος αποκρύπτοντας την παραμετροποίηση των πόρων από την εφαρμογή. Ένα παράδειγμα εμφανίζεται στο [87] για την εξωτερική ανάθεση τμήματος μιας εφαρμογής για κινητές συσκευές.

Διαθεσιμότητα: η διαθεσιμότητα έχει κεντρικό ρόλο στη διαχείριση των Service Level Agreements (SLAs). Για το λόγο αυτό οι πάροχοι υπηρεσιών κάνουν ό,τι καλύτερο για να αυξήσουν τη διαθεσιμότητα των εφαρμογών υπηρεσιών νέφους που φιλοξενούνται. Ελαττωματικά στοιχεία υλικού μπορούν να ρίξουν τμήματα της υποδομής. Σε αυτές τις περιπτώσεις η διαθεσιμότητα είναι δυνατή όχι μόνο από τη μετεγκατάσταση της εικονικής μηχανής, αλλά με τους ίδιους μηχανισμούς που μπορούν να επιτύχουν τη κλιμάκωση της

συμπεριφοράς. Για παράδειγμα, λόγω της συντήρησης υλικού στις εγκαταστάσεις μιας υπηρεσίας παροχής, αυτή μπορεί να επιλέξει να αναθέσει σε εξωτερικούς υπολογιστές τις εικονικές μηχανές (VM) ή τις φιλοξενούμενες εφαρμογές με στόχο την επίτευξη αδιάλειπτης παροχής υπηρεσιών. Η όλη διαδικασία πρέπει να είναι διαφανής για την πλατφόρμα που διαχειρίζεται το σύννεφο και το ενδιάμεσο λογισμικό θα πρέπει να επιτρέπει και να διευκολύνει τέτοιου είδους ενέργειες διαφυλάσσοντας ταυτόχρονα εκτός από την αδιάληπτη λειτουργία όλων των εφαρμογών, την ασφάλεια των δεδομένων τόσο κατά τη μεταφορά όσο και κατά τη λειτουργία.

Χρέωση: σε μια υποδομή υπολογιστικού νέφους, η ανάπτυξη εικονικών μηχανών είτε containers σε διαφορετικούς προμηθευτές έχει ένα μεγάλο πλεονέκτημα στο τομέα της τιμολόγησης. Τα instances της εφαρμογής στο νέφος μπορεί να μετεγκατασταθούν σε διάφορους παρόχους υπηρεσιών που προσφέρουν ένα ανταγωνιστικότερο πλεονέκτημα στο τομέα της τιμολόγησης ή της διαθεσιμότητας τους. Αυτό συνάδει με μια ευρύτερη άποψη της υπολογιστικής ενδιάμεσης πλατφόρμας στην ανταγωνιστική αγορά όπου οι υποδομές-ως-υπηρεσία διασχίζουν διοικητικούς τομείς για να εκτελέσουν όσο δυνατόν πιο αποδοτικά τις εφαρμογές. Το πρότυπο για την ανάλυση της χρησιμότητας των διαθέσιμων πόρων θα πρέπει να επεκταθεί και στις εφαρμογές που είναι προσανατολισμένες για να τρέχουν στο υπολογιστικό νέφος. Τα εγγενή ενδιάμεσα προγράμματα θα πρέπει να επιτρέπουν την τιμολόγηση και την παρακολούθηση της χρήσης της εφαρμογής του υπολογιστικού νέφους, καθώς επίσης και την παρακολούθηση του κόστους της, από την άποψη των πόρων. Με αυτόν τον τρόπο ένας ευφυής μηχανισμός κλιμάκωσης θα

μπορούσε να αποφέρει άμεσα οφέλη στον τελικό χρήστη επιτρέποντας την ορθή πρόβλεψη χρήσης και κατά συνέπεια την τιμολόγηση ανάλογα με τη χρήση, ενώ ταυτόχρονα μεγιστοποιεί το κέρδος για τον ιδιοκτήτη της αίτησης.

Λειτουργικότητα της υπηρεσιογενής πλατφόρμας εύρεσης αναζήτησης και απόκτησης εφαρμογών που είναι προσανατολισμένες στο υπολογιστικό νέφος (cloud marketplaces): υπήρξαν διάφορες προσπάθειες για την παροχή αγορών [88], [89]. Η λειτουργικότητα της αγοράς θεωρείται θεμελιώδης σε ένα τοπικό ενδιάμεσο πλαίσιο υπολογιστικού νέφους, καθώς συνδυάζεται στενά με τις επιχειρηματικές πτυχές του, όπως η λογιστική ανάλυση και η τιμολόγηση. Οι πόροι, τα προϊόντα και οι υπηρεσίες θα διαφημιστούν στις αγορές για όλα τα ενδιαφερόμενα και καινοτόμα μέρη που χρησιμοποιούν. Σε αυτό το πλαίσιο, μια ενδιάμεση πλατφόρμα υπολογιστικού νέφους θα πρέπει να είναι σε θέση να παρέχει στους πράκτορες/πελάτες πιθανές φθηνότερες εναλλακτικές λύσεις που μπορούν να υποστηρίξουν τα SLAs, ενώ ταυτόχρονα να μειώνει το κόστος και να μεγιστοποιεί το κέρδος. Τα παζάρια πόρων πολλαπλών στρωμάτων φαίνεται να είναι ένα αναπόφευκτο μέρος ενός ενδιάμεσου πόρου υπολογιστικού νέφους μαζί με μεθόδους για να τους κρατήσει ενημερωμένους σε πραγματικό χρόνο. Σε μια πιο αφηρημένη μορφή, μια αγορά θα μπορούσε να δημιουργηθεί ως εφαρμογή υπολογιστικού νέφους. Ωστόσο, η διατήρηση αυτής της λειτουργικότητας στο ενδιάμεσο λειτουργικό επίπεδο παρέχει γρήγορη και οικονομικά αποδοτική κλιμάκωση χωρίς να εκθέτει τις λεπτομέρειες κλιμάκωσης στις άλλες εφαρμογές του υπολογιστικού νέφους ή σε άλλους ενδιαφερόμενους και ανταγωνιστές.

Σε όλες τις παραπάνω περιπτώσεις είναι προφανές ότι οι απαιτήσεις για ένα ενδιάμεσο πλαίσιο υπολογιστικού νέφους είναι πολύ δύσκολες για να εκπληρωθούν σε ένα καταναμημένο υπόστρωμα εικονικής διαμόρφωσης. Σε συνδυασμό με τη δυναμική φύση της τιμολόγησης των πόρων και την αυξανόμενη απελευθέρωση από τον προμηθευτή των προτύπων διασύνδεσης, η ορθή χρήση των πόρων του υπολογιστικού νέφους θα διαδραματίσει καίριο ρόλο στο μέλλον. Προς αυτή την κατεύθυνση το περιβάλλον υπολογιστικού νέφους θα προσαρμοστεί σε μια αγορά πόρων και υπηρεσιών με πρωτοφανή ευελιξία.

Για να ικανοποιηθούν οι παραπάνω απαιτήσεις, θα πρέπει να επανακαθοριστεί μια νέα αρχιτεκτονική ενδιάμεσου συστήματος. Στην προσπάθειά όμως αυτή θα πρέπει να εγκαταλειφθεί η απαίτηση συμβατότητας για τις παλαιότερες εφαρμογές, επειδή δεν ταιριάζουν με τις παραπάνω απαιτήσεις. Αυτές μπορεί να οδηγήσουν σε μια πληθώρα προβλημάτων είτε εννοιολογικής φύσης είτε θεμάτων συμβατότητας. Μοιραία η αρχιτεκτονική θα πρέπει να είναι προσανατολισμένη σε σύγχρονες κλιμακούμενες εφαρμογές που πηγάζουν από τη φιλοσοφία των πλατφόρμων του υπολογιστικού νέφους. Σε αυτά τα περιβάλλοντα, οι φιλικά προς το υπολογιστικό νέφος εφαρμογές μπορεί να θεωρηθούν ως πράκτορες σμήνη (agent swarms) που κλιμακώνονται όταν είναι απαραίτητο, αλλά μπορούν την ίδια στιγμή να κρατήσουν την κατάστασή τους και να συγχρονίζονται αυτόματα με ελάχιστες απαιτήσεις για τη μετεγκατάσταση δεδομένων που πραγματοποιείται από το σύστημα εξισορρόπησης φορτίου που συνήθως είναι ενταγμένο στην ενδιάμεση πλατφόρμα διαχείρισης. Η [90] πάνω στη Cassandra προσφέρει τη

δημιουργία και συγχρονισμό νέων κόμβων παρέχοντας σε αυτήν μόνο τις θέσεις των υφιστάμενων κόμβων. Αυτό είναι το ίδιο του -ενδιάμεσου- λογισμικού που μπορεί να βγάλει από τη δύσκολη θέση της επιλογής τον χρήστη και να του προτείνει να επιλέξει τις καλύτερες εναλλακτικές λύσεις κόστους ώστε να αναπτύξει την εφαρμογή απρόσκοπτα αλλά και ταυτόχρονα να παρέχει ταχύτερη κάλυψη του αρχικού κεφαλαίου που είχε επενδυθεί (Return of Investment). Σε όλες τις περιπτώσεις ο αρχικός και κύριος στόχος είναι η απόδοση της επένδυσης οπότε η προτεινόμενη επιλογή και στόχευση είναι κάτι παραπάνω από επιτακτική. Ταυτόχρονα τα συστήματα αυτά είναι ευκολότερα και φθηνότερα στη διαχείριση, στη μοντελοποίηση, την προσομοίωση και στον υπολογισμό της τελικής τιμής. Τέτοια συστήματα υπάρχουν ή μπορούν να παρασχεθούν με ελάχιστες αλλαγές σε υπάρχοντες υλοποιήσεις.

Στόχος είναι η δημιουργία μιας προηγμένης πλατφόρμας PaaS υπολογιστικού νέφους, η οποία να υποστηρίζει τη βελτιστοποιημένη και ελαστική φιλοξενία εφαρμογών πολλαπλών βαθμίδων στο διαδίκτυο. Θα ενσωματώνει όλα τα απαραίτητα χαρακτηριστικά, διευκολύνοντας τον προγραμματισμό των εφαρμογών και επιτρέποντας ταυτόχρονα τη δημιουργία ενός αληθινού επιχειρηματικού οικοσυστήματος, όπου οι εφαρμογές που προέρχονται από διαφορετικούς παρόχους μπορούν να προσαρμόζονται σε διαφορετικούς χρήστες, να συγκεντρώνονται και να διαπραγματεύονται μαζί. Υπάρχουν πολλές άλλες πλατφόρμες PaaS που στοχεύουν σε ένα περιορισμένο ρεπερτόριο των πτυχών υπολογιστικού νέφους, ο στόχος όμως είναι να αντιμετωπίζουν το πρόβλημα με έναν τυποποιημένο, ανεξάρτητο από πωλητή και προσανατολισμένο προς τις επιχειρήσεις

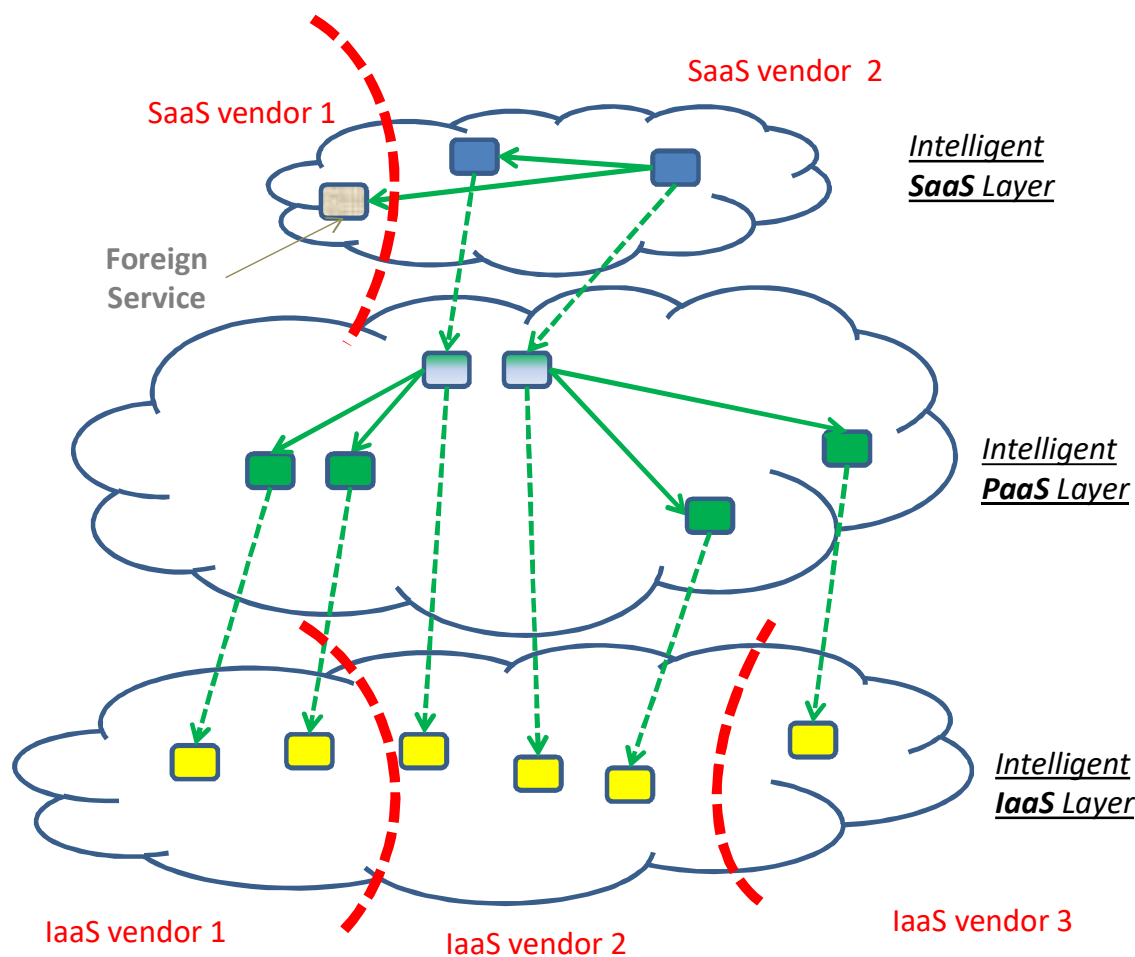
τρόπο. Ταυτόχρονα στόχος είναι να πρέπει να προσβλέπει σε μεγαλύτερη αυτοματοποίηση και αυξημένη νοημοσύνη, δημιουργώντας ένα ενοποιημένο ενδιάμεσο λογισμικό.

3.5 Αρχές σχεδίασης πλατφόρμας

Το πρώτο βήμα στο σχεδιασμό μιας ευέλικτης και έξυπνης αρχιτεκτονικής ενδιάμεσου προγράμματος για μια ευέλικτη πλατφόρμα υπολογιστικού νέφους είναι ο ορισμός μιας αφηρημένης αποσύνθεσης στρώσεων για την απομόνωση και αποδόμηση των επιχειρηματικών πτυχών της προτεινόμενης λύσης. Ακολουθώντας την παραδοσιακή διάκριση μιας πλατφόρμας υπολογιστικού νέφους όσον αφορά τα στρώματα [91], υπάρχει το επίπεδο SaaS όπου είναι το επίπεδο όπου οι εφαρμογές υπολογιστικού νέφους εξάγονται ως υπηρεσίες. Το στρώμα PaaS όπου είναι το περιβάλλον εκτέλεσης των εφαρμογών με δυνατότητα υπολογιστικού νέφους και τέλος το στρώμα IaaS όπου είναι το περιβάλλον των πόρων που χρησιμοποιείται από εφαρμογές στο σύννεφο.

Στην αρχιτεκτονική, ακολουθώντας την υπόθεση ότι ένα σύννεφο μπορεί να θεωρηθεί ως μία οντότητα από την άποψη των IaaS, όλα τα ζητήματα διαλειτουργικότητας μεταξύ των API που καλύπτουν τον κύκλο ζωής των εικονικών μηχανών και τις παραμέτρους του αποκρύπτονται πίσω από ένα μεμονωμένο κοινό API συνάθροισης, όπως το libcloud ή το Attic. Αυτά τα API παρέχουν ένα στρώμα προσαρμογής που είναι ο συνδεδεμένος κρίκος για την ευκολότερη και πιο ευέλικτη επικοινωνία.

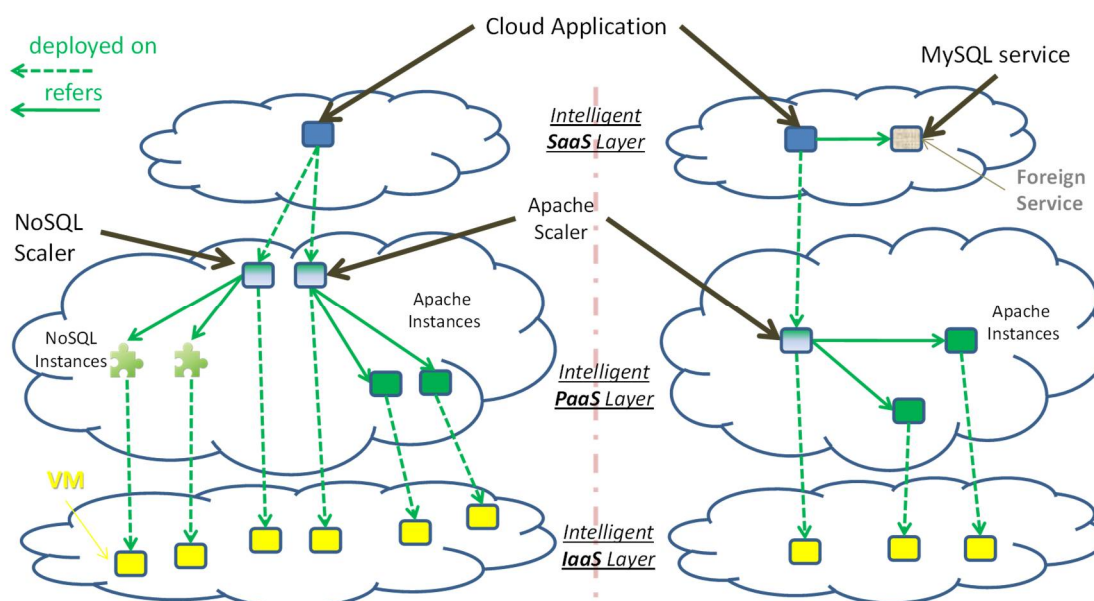
Το επίπεδο SaaS μπορεί να αντιπροσωπεύεται από την SOA αρχιτεκτονική σε μια πιο αφηρημένη δομή [92]. Σε ορισμένες προσεγγίσεις το στρώμα PaaS μπορεί να χρησιμοποιήσει εξαγόμενες υπηρεσίες και για το λόγο αυτό μπορεί επίσης να θεωρηθεί ως ένα πιο πρωτόγονο επίπεδο υπηρεσιών. Το πιο συνηθισμένο παράδειγμα είναι η χρήση μιας βάσης δεδομένων που μπορεί να αποδίδεται στο επίπεδο PaaS από μια εφαρμογή cloud. Παρά το γεγονός ότι αυτό είναι δυνατό σε συγκεκριμένους παρόχους υπηρεσιών υπολογιστικού νέφους, δεν είναι κατάλληλο για ευφυή σύννεφα που εκτείνονται σε διάφορους πωλητές, δεδομένου ότι έρχονται σε αντίθεση με τη φύση των PaaS ως ένα περιβάλλον εκτέλεσης που παρουσιάζει εννοιολογικές προκλήσεις όπως παρουσιάζεται στην Εικόνα 6.



Εικόνα 6 Έξυπνες δομές υπολογιστικού νέφους από διαφορετικούς πάροχους υπηρεσιών

Σε μια SOA τύπου λογική θα γινόταν η επιλογή η βάση δεδομένων να τρέχει σαν εφαρμογή στο PaaS και να εξαγάγει τις υπηρεσίες της στο επίπεδο SaaS. Από αυτή την άποψη, είτε η βάση δεδομένων μπορεί να εξαχθεί ως υπηρεσία ή μπορεί να εγκατασταθεί ως ένα άλλο προϊόν που εξαρτάται από την εφαρμογή cloud, αλλά δεν εξάγεται ως υπηρεσία cloud. Σε κάθε περίπτωση αυτό είναι εσωτερικό για την εφαρμογή που έχει

αναπτυχθεί για το υπολογιστικό νέφος. Στην Εικόνα 7 το αριστερό κομμάτι είναι χωρίς τη χρήση έξυπνης διαχείρισης σε αντίθεση με το δεξί που ακολουθεί το νέο μοντέλο. Η διάκριση μεταξύ εφαρμογών cloud και υπηρεσιών είναι άσχετη με την έννοια του δικτύου των υπηρεσιών, δεδομένου ότι οποιαδήποτε εφαρμογή μπορεί να καταναλώσει ή να εξαγάγει μια διασύνδεση υπηρεσιών. Ωστόσο, κάθε εφαρμογή χρειάζεται διάφορα τμήματα του λογισμικού που είναι σε θέση να υποστηρίξουν τις δραστηριότητές της.



Εικόνα 7 Κατανομή υπηρεσιών στα επίπεδα του Υπολογιστικού νέφους

Αναλύοντας αυτό με ένα απλό παράδειγμα, μια ιστοσελίδα γραμμένη σε ένα συνδυασμό γλωσσών PHP και HTML που εξυπηρετείται από μια βάση δεδομένων και ένα διακομιστή web θα μπορούσε ακολουθώντας μια απλουστευμένη ανάπτυξη σε ένα περιβάλλον

βασισμένο σε υπολογιστικό νέφος να ισοδυναμεί με την τοποθέτηση κάθε στοιχείου της εφαρμογής σε ένα εικονικό μηχάνημα και να έχει κατακόρυφη κλιμάκωση. Δηλαδή ανάλογα με το φόρτο υπηρεσιών στη μονάδα της ώρας ο πάροχος υπηρεσιών δημιουργεί είτε σταματά εικονικές μηχανές για να καλύψει το υπολογιστικό φορτίο. Αυτή η προσέγγιση συνεπάγεται διάφορα προβλήματα. Το πρόβλημα είναι ο τρόπος αντιμετώπισης της κλιμάκωσης τόσο για τη βάση δεδομένων όσο και για το διακομιστή web όταν η χρήση της εφαρμογής αυξάνεται πέραν της χωρητικότητας της εικονικής μηχανής. Τα κύρια σημεία ανησυχίας μπορούν να συνοψιστούν στον ορισμό των εναυσμάτων που ενεργοποιούν την κλιμάκωση της βάσης δεδομένων καθώς και στον ορισμό των εναυσμάτων που ενεργοποιούν την κλιμάκωση του διακομιστή web.

Εννοιολογικά μια εφαρμογή γίνεται πιο περίπλοκη όταν υλοποιεί την ελαστικότητα απευθείας (αν και η δυνατότητα κλιμάκωσης είναι μέρος της εφαρμογής). Για τον ίδιο λόγο, η τιμολόγηση των εξαγόμενων υπηρεσιών δεν εξαρτάται μόνο από την εφαρμογή, αλλά συμπεριλαμβάνει και τη χρήση των πόρων και τον επικουρικών υπηρεσιών/προϊόντων που χρησιμοποιεί. Ακόμα εξαρτάται από τις αναφορές σε άλλες υπηρεσίες και το περιθώριο κέρδους του προμηθευτή. Η προσδοκία των εφαρμογών cloud είναι η εκμετάλλευση του νέφους με ομοιόμορφο τρόπο με βάση την γεγονός ότι μια εφαρμογή κάνει μόνο ένα πράγμα, και υλοποιεί μια επιχειρηματική λογική. Μια εφαρμογή που μπορεί να ακολουθήσει το προηγούμενο παράδειγμα εμφανίζεται στην Εικόνα 7. Αυτός ο τύπος του σχεδιασμού μιας εφαρμογής προσανατολισμένης στη φιλοσοφία του Cloud Computing μπορεί εύκολα να σχεδιαστεί με ένα γραφικό εργαλείο. Αυτό είναι ένα

σημαντικό πλεονέκτημα μιας και τα σύγχρονα συστήματα είναι αρκετά πολύπλοκα και ο κώδικας τείνει να μεγαλώνει και να γίνεται όλο και πιο πολύπλοκος. Με αυτό το τρόπο ο προγραμματιστής είτε ο αρχιτέκτονας των εφαρμογών διευκολύνεται τόσο κατά τη διάρκεια της δημιουργίας όσο και κατά τη διάρκεια του ελέγχου.

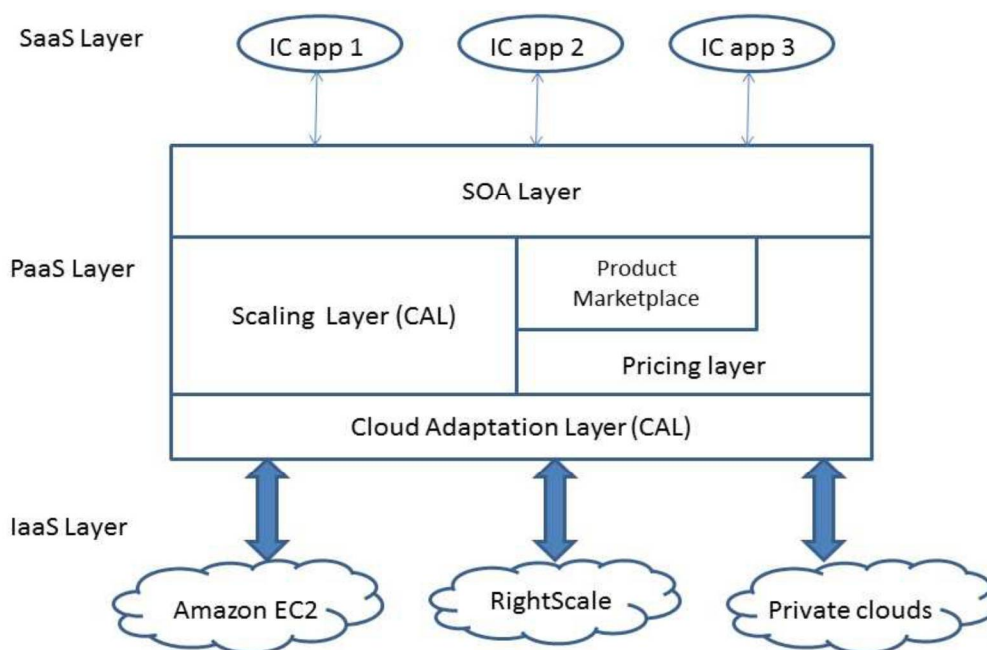
3.6 Αρχιτεκτονική ενδιάμεσου λογισμικού προσανατολισμένη σε υπηρεσιοστρεφείς αρχιτεκτονικές

Όπως παρουσιάστηκε στο παραπάνω παράδειγμα, θα πρέπει να διερευνηθούν πολλές πληροφορίες και πτυχές για να σχεδιαστεί ένα ευφύες ενδιάμεσο επίπεδο. Ο συνδυασμός ενός προϊόντος με τη μονάδα εξισορρόπησης φόρτου προϊόντος επιτρέπει την ελαστικότητα μιας εφαρμογής νέφους οριζόντια. Αλλά αυτό δεν είναι αρκετό. Για να βελτιωθούν οι δυνατότητες κλιμάκωσης της πλατφόρμας, πρέπει να εισαχθεί ένας συγκεκριμένος παράγοντας ως θεμελιώδες στοιχείο της αρχιτεκτονικής. Αυτός ο παράγοντας, που έχει ονομαστεί *product scaler*, αποκρύπτει την οριζόντια ή κατακόρυφη κλιμάκωση ενός προϊόντος, παρέχοντας μια ανεξάρτητη διασύνδεση. Το προϊόν/υπηρεσία περιγράφει τους κάθετους κανόνες κλιμάκωσης σύμφωνα με τη χρήση του, οι οποίοι, με άλλα λόγια, περιγράφουν μεθόδους για τη μείωση ή την αύξηση της δέσμευσης πόρων με βάση την τρέχουσα ζήτηση. Η μονάδα εξισορρόπησης φορτίου καταχωρεί νέα instances ή σταματά και αποδεσμεύει αυτά τα instances που δεν έχουν ρόλο για τη συγκεκριμένη υπηρεσία. Από την άλλη και ο *product scaler* του προϊόντος είναι υπεύθυνος για τη λήψη απόφασης κλιμάκωσης για την αύξηση των instances των προϊόντων, τον

επαναυπολογισμό των ρυθμίσεων παραμέτρων του προϊόντος και τις τροποποιήσεις πόρων που έχουν αντιστοιχιστεί με βάση την συνολική ζήτηση. Αυτό είναι δυνατό επειδή - δεδομένων των κατακόρυφων κανόνων του προϊόντος και της δυνατότητας υπολογισμού της κατανομής της ζήτησης- ο product scaler μπορεί α) να παρουσιάσει μια πρόταση κλιμάκωσης, β) να κάνει μια εκχώρηση τιμής στην πρόταση και γ) να πραγματοποιήσει μια ρύθμιση παραμέτρων εξισορρόπησης φορτίου. Η οριζόντια κλιμάκωση και η εξισορρόπηση φορτίου είναι δύο διαδικασίες που σχετίζονται με την εννοιολογική διαδικασία και, ως εκ τούτου, πρέπει να ελέγχονται από τον product scaler για τους προαναφερόμενους λόγους. Η μονάδα εξισορρόπησης φόρτου προϊόντος μπορεί να είναι ένα προϊόν με δική του διαφορετική στρατηγική που παρέχεται από διαφορετικό προμηθευτή και οι πολιτικές για την εξισορρόπηση φόρτου θα μπορούσαν να υλοποιηθούν σε αυτό το προϊόν. Αυτά θα παρέχονται στον product scaler ως υποδείξεις. Ωστόσο, η πραγματική λογική ελαστικότητας θα πρέπει να βρίσκεται στο ενδιάμεσο επίπεδο, δεδομένου ότι συνδέεται με εσωτερικά στοιχεία και η μονάδα εξισορρόπησης φορτίου θα πρέπει να είναι όσο το δυνατόν απλούστερη.

Αυτές οι παρατηρήσεις αποτυπώνουν την κύρια ουσία της ελαστικότητας του υπολογιστικού νέφους στην αρχιτεκτονική. Ένα υπηρεσιοστρεφές marketplace έρχεται ως μέσο για την ανάκτηση των τιμών των προϊόντων και των πόρων για την τιμολόγηση της κατανάλωσης. Στην ανάλυσή θεωρείται η παραδοχή του κενού εικονικού μηχανήματος, που σημαίνει ότι ένα εικονικό μηχάνημα που είναι κενό στη δημιουργία μπορεί να έχει διάφορα προϊόντα λογισμικού που είναι απαραίτητα για τις υπηρεσίες των πελατών όπου

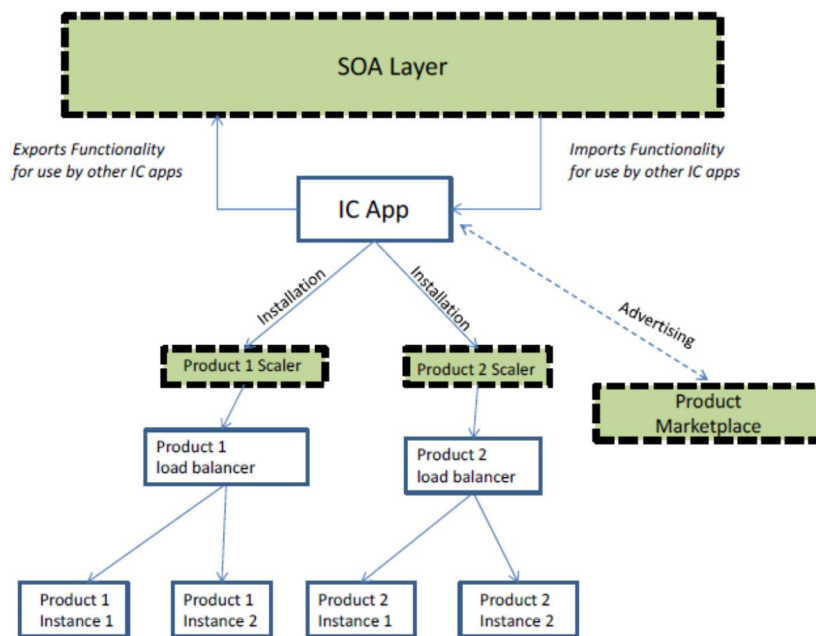
μπορεί να εγκατασταθούν ή να απεγκατασταθούν κατ' απαίτηση. Μια τέτοια λειτουργικότητα μπορεί να την πραγματοποιήσουν προϊόντα και πλατφόρμες όπως το Puppet [93] και το Chef [94]. Η αγορά υπηρεσιών μπορεί να ενημερωθεί από προϊόντα λογισμικού που παρέχονται από προμηθευτές, υπηρεσίες που εξάγονται από το επίπεδο και τους πόρους των παρόχων cloud που ανακτώνται από το Cloud Adaptation Layer (CAL) όπως παρουσιάζεται και στην Εικόνα 9. Υπό την έννοια αυτή, δεν υπάρχει ανάγκη για μια μοναδική διαχείριση πόρων ή μια βελτιστοποίηση πόρων όπως σε περιβάλλοντα με περιορισμούς παραδοσιακών πόρων.



Εικόνα 8 Αρχιτεκτονική παρουσίαση επιπέδων

Σε αντίθεση με αυτά τα περιβάλλοντα, η παρούσα αρχιτεκτονική εκμεταλλεύεται το γεγονός ότι οι πλατφόρμες cloud επιτρέπουν την ψευδαίσθηση ενός απεριόριστου αριθμού πόρων. Η έλλειψη πόρων μπορεί να αντικατοπτρίζεται στις υψηλές τιμές αυτών των πόρων στην αγορά προϊόντων και ο μόνος περιορισμός στην κατανάλωση τους είναι το πορτοφόλι του χρήστη. Ως εκ τούτου, η έμφαση στην τιμολόγηση είναι ένα άλλο χαρακτηριστικό γνώρισμα των πλατφόρμων cloud που επηρεάζουν τον σχεδιασμό της αρχιτεκτονικής. Για τον λόγο αυτόν, δόθηκε ιδιαίτερη προσοχή στο να συμπεριληφθεί μια υγιής αποσύνθεση των υποδομών και να ληφθεί υπόψη η συγκεκριμένη λεπτομέρεια για τις αποφάσεις. Η προτεινόμενη αρχιτεκτονική μας απεικονίζεται στην Εικόνα 8, ενώ μια τυπική εφαρμογή της αρχιτεκτονικής και η εξάρτησή της από προϊόντα εμφανίζονται στην Εικόνα 9.

Στην Εικόνα 8 το επίπεδο CAL είναι το θεμελιώδες στρώμα που επιτρέπει τη δημιουργία, την καταστροφή, τη διαμόρφωση, τη διαχείριση και την παρακολούθηση μιας εικονική μηχανής. Επίσης, επιτρέπει τη δυνατότητα υποβολής ερωτημάτων σε διαφορετικές υπηρεσίες παροχής cloud για πόρους και την τιμολόγηση τους. Αυτή η λίστα προωθείται στην υποενότητα του pricing layer ενώ συνδέεται με το επίπεδο κλιμάκωσης, προκειμένου να κλιμακωθούν τα προϊόντα που είναι απαραίτητα για τη λειτουργία των εφαρμογών που βρίσκονται στο επίπεδο του SOA.



Εικόνα 9 Διαχείριση τρόπου εγκατάστασης cloud εφαρμογών

Το επίπεδο τιμολόγησης είναι υπεύθυνο για τη βέλτιστη επιλογή της καλύτερης τιμής των υπάρχουσών προτάσεων κλιμάκωσης, της τοπολογίας από το επίπεδο κλιμάκωσης που θα παραχθεί αυτό ή ακόμη και για να προτείνει μια οικονομικά αποτελεσματική εναλλακτική λύση. Επίσης, υπολογίζει τις τιμές της χρήσης του στρώματος εφαρμογής και τη χρήση των πόρων. Για να επιτευχθεί αυτό, κάνει χρήση και συνεχή ερωτήματα σχετικά με τα προϊόντα που βρίσκονται στο marketplace. Το marketplace είναι ο κεντρικός κατάλογος τιμών όπου μια εφαρμογή που ακολουθεί την προτεινόμενη αρχιτεκτονική μπορεί να αναζητήσει προϊόντα που ικανοποιούν τις

εξαρτήσεις των υπηρεσιών της. Αυτές οι εξαρτήσεις μπορούν να κωδικοποιηθούν ως ένα συγκεκριμένο προϊόν (είτε συγκεκριμένη έκδοση αυτού του προϊόντος) ή προϊόντα που συμμορφώνονται με μια συγκεκριμένη έκδοση ενός προτύπου, για παράδειγμα ANSI SQL 98. Το επίπεδο SOA εξάγει ένα κατάλληλο API για την κατασκευή agents στους οποίους θα αναπτυχθούν οι υπηρεσίες SaaS. Ενώ ο scale manager διαχειρίζεται τους πόρους σύμφωνα με τη χρήση του SaaS, το επίπεδο SOA που διαχειρίζεται τη χρήση του SaaS είτε χρησιμοποιείται από άλλα SaaS είτε απευθείας από έναν πελάτη. Ο παράγοντας που εκπροσωπεί το SaaS που αναπτύσσεται στην πλατφόρμα εξάγει υπηρεσίες για κατανάλωση από άλλες SaaS υποδομές, παρακολουθεί την τιμολόγηση των πόρων της ανάπτυξης, παρακολουθεί την υγεία της υπηρεσίας, συμμετέχει στη διαμόρφωση ή την επαναδιαμόρφωση του σύμφωνα με τις αποφάσεις τιμολόγησης και την τιμολόγηση της SaaS κατανάλωσης. Κατ' αρχήν, η υλοποίηση του SaaS κατανέμεται μεταξύ των scaler και των αντικειμένων που εγκαθίστανται στα προϊόντα, αλλά η οντότητα αντιπροσωπεύεται από τους agents του SOA.

Οι πράκτορες(agent) είναι επίσης υπεύθυνοι για τη διαφήμιση της αίτησης στην αγορά προϊόντων. Στην Εικόνα 9, η εφαρμογή IC είναι στην πραγματικότητα ο agent της SOA που είναι χτισμένος πάνω στο SOA API επίπεδο. Η αποσύνθεση που περιγράφεται ακολουθεί και βασίζεται στις αρχές ενός κατανεμημένου συστήματος βάσεων δεδομένων [95] .

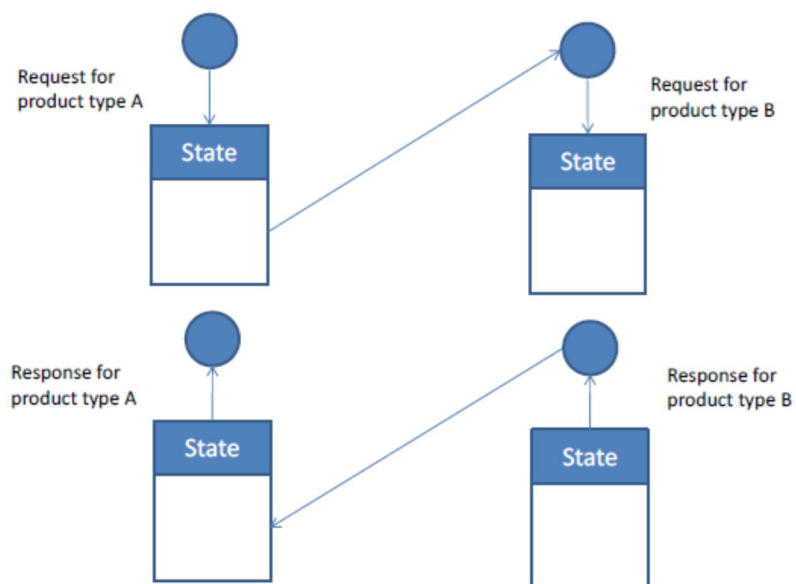
Ένα μεγάλο πλεονέκτημα της αρχιτεκτονικής που προκύπτει από την Εικόνα 9 είναι ότι η πολλαπλή μίσθωση έρχεται δωρεάν στο επίπεδο SaaS, αν και δεν απαιτείται

στο επίπεδο του PaaS. Αυτό το πλεονέκτημα μπορεί να φανεί καλύτερα από το γεγονός ότι τα στοιχεία PaaS βλέπουν μόνο έναν μισθωτή, την SaaS επιπέδου εφαρμογή. Η εφαρμογή επιπέδου SaaS, χρησιμοποιώντας τη φιλοσοφία του SOA, μπορεί να δημιουργήσει διαφορετική διαχείριση ανά μισθωτή. Αυτό δεν ανατίθεται σε μηχανισμούς στρώσεων PaaS (αν και μπορούν να χρησιμοποιηθούν για την αποδοτική πολλαπλή μίσθωση της εφαρμογής) και για το λόγο αυτό ο έλεγχος πρόσβασης με βάση τον ρόλο στους πόρους που προορίζονται για την εφαρμογή είναι δυνατός. Έχοντας αυτό σαν δεδομένο, μπορεί να γίνει κατανοητό ότι η έννοια του χρήστη είναι ενσωματωμένη και ξεκάθαρη στην αίτηση. Αυτό είναι ένα κρίσιμο χαρακτηριστικό της προτεινόμενης πλατφόρμας, επειδή αποσυνδέει τις απαιτήσεις πολλαπλής μίσθωσης από μια προσέγγιση μεικτού στρώματος, όπως αυτή που εφαρμόστηκε από την Salesforge που βασιζόταν σε ένα μόνο επίπεδο και πιο συγκεκριμένα στην cloud εφαρμογή. Αυτό είναι το μόνο μέρος της πλατφόρμας που θα πρέπει να είναι γνωστό στους πελάτες, επειδή αυτό το κομμάτι επιτρέπει και διαχειρίζεται την αλληλεπίδραση μεταξύ τους. Ωστόσο, η ευελιξία του σχεδιασμού επιτρέπει με μικρές τροποποιήσεις τη συμπερίληψη των πληροφοριών που σχετίζονται με τον μισθωτή στο scaler όταν η μισθωτή προσανατολισμένη τιμολόγηση πόρων και η κλιμάκωση είναι απαραίτητη. Σε αυτή την περίπτωση ο scaler διατηρεί συγκεκριμένο VM για το μισθωτή και συγκεκριμένες ενέργειες ελαστικότητας. Ωστόσο, όπως και στην περίπτωση του backbone internet, οι ενοικιαστές θα πρέπει να είναι οι πελάτες του επιπέδου SaaS που συγκεντρώνουν τη μικροκατανάλωση από την άποψη της ομαδικής κατανάλωσης [96].

3.7 Αποτελέσματα ανάλυσης

Σε αυτή την ενότητα, αξιολογείται ο προτεινόμενος σχεδιασμός, τονίζοντας τα πλεονεκτήματα και τα μειονεκτήματά του ως μια μελλοντική αρχιτεκτονική ενδιάμεσου σχεδιασμού. Τα σενάρια στα οποία δίνονται βάση είναι κυρίως εκείνα όπου το κύριο συστατικό (component) της εφαρμογής που χειρίζεται τις αιτήσεις υπηρεσιών βρίσκεται σε ένα διακομιστή web και χρησιμοποιεί άλλα προϊόντα, μετατρέποντας τις αιτήσεις υπηρεσιών σε άλλες αιτήσεις προϊόντων. Αναμένεται ότι η πλειονότητα των εφαρμογών θα ακολουθήσουν αυτό το σχεδιασμό, όπως παρουσιάζεται και στην Εικόνα 10. Εφαρμογές όπως τα αυτοματοποιημένα αντίγραφα ασφαλείας δεν συμμορφώνονται με αυτό το μοτίβο σχεδίασης, αλλά αναμένεται ότι αυτός ο περιορισμός δεν είναι έγκυρος όταν εκτίθενται ως μόνιμες αποθήκες αντικειμένων SaaS για σκοπούς αρχειοθέτησης.

Στη παρούσα αρχιτεκτονική γίνεται η παραδοχή ότι τα προϊόντα μπορούν να διαμορφωθούν με συγκεκριμένο τρόπο. Για μια σωστή περιγραφή ενός ελαστικού κομματιού πρέπει να περιγράψουμε τόσο τους κανόνες κλιμάκωσης του προϊόντος όσο και το ίδιο το προϊόν. Κεντρική σε αυτή την περιγραφή είναι η έννοια του usage profile vector U . Το διάνυσμα αυτό είναι μια ποσότητα που σχετίζεται με τη χρήση του προϊόντος που μπορεί να μετρηθεί, και κατά συνέπεια δείχνει την ποσοτικοποίηση του φόρτου.

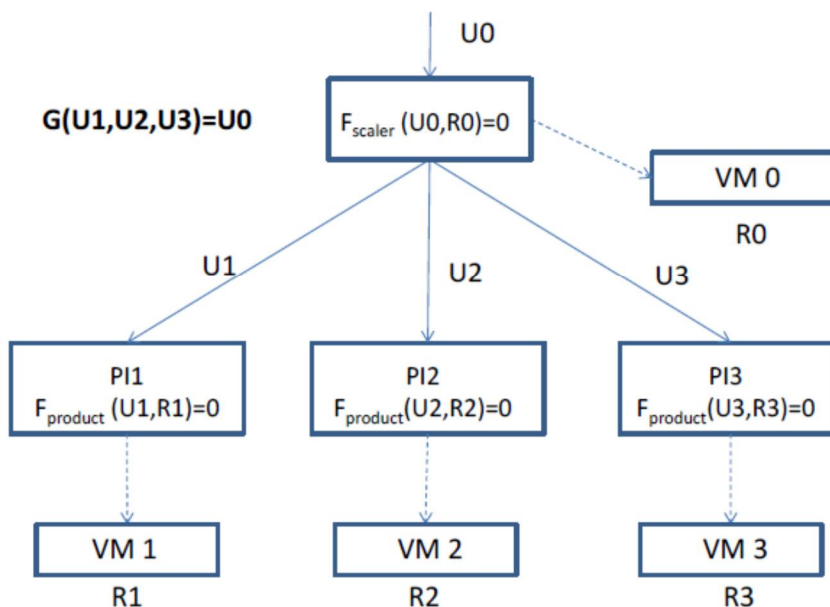


Εικόνα 10 Πρότυπη αναγγελία της Request Transformer

Κάθε χρήση ικανοποιείται από ένα διάνυσμα πόρων R που αντιστοιχίζεται σε ένα VM, που ποσοτικοποιεί τους πόρους που είναι απαραίτητοι για την ικανοποίηση ενός φορτίου U . Για παράδειγμα, τα διανυσματικά στοιχεία πόρων μπορεί να είναι υλικά όπως η CPU, η μνήμη RAM, η τοπική αποθήκευση, η αποθήκευση στο cloud, η ταχύτητα bitrate σύνδεσης δικτύου, ενώ τα στοιχεία προφίλ χρήσης θα μπορούσαν να είναι ποσοστό αίτησης, ρυθμός φόρτωσης ή λήψης και αδράνεια συναλλαγής. Πολλοί συνδυασμοί πόρων μπορούν να ικανοποιήσουν ένα συγκεκριμένο προφίλ και πολλά προφίλ μπορούν να ικανοποιηθούν από ένα συγκεκριμένο συνδυασμό πόρων. Αυτό το αποτέλεσμα που είναι το καρτεσιανό αποτέλεσμα του υποσυνόλου του προφίλ πόρου/χρήσης έχει μια χαρακτηριστική συνάρτηση που είναι ένας σχεδιαστικός περιορισμός και αντιστοιχεί στη

συνάρτηση F στην Εικόνα 11. Μια άλλη απαραίτητη λειτουργία που εμφανίζεται επίσης στην Εικόνα 11 είναι η συνάρτηση G που χαρακτηρίζει το scaler. Η συνάρτηση αυτή ποσοτικοποιεί τον τρόπο με τον οποίο ένα φορτίο χωρίζεται σε περιπτώσεις. Στην απλούστερη μορφή του για ένα προφίλ χρήσης ο ρυθμός αιτήσεων σε stateless διακομιστές web είναι μια απλή άθροιση. Ανάλογα με το εκάστοτε προφίλ χρήσης θα υπάρχουν διαφορετικές συναρτήσεις. Αυτές οι λειτουργίες και οι περιγραφές είναι απαραίτητες για τη σχεδίαση του ενδιάμεσου προϊόντος, επειδή παρέχουν αφαιρετικότητα στα προϊόντα. Κάτω από αυτό το αφαιρετικό προϊόν μπορεί να θεωρηθεί ότι λογίζονται ως μαύρα κουτιά που μπορούν να τιμολογούνται εύκολα. Αν και υπάρχει σημαντικό έργο προς αυτή την κατεύθυνση, τα αποτελέσματα δεν είναι έτοιμα για χρήση [97] και οι τοπικές προσεγγίσεις βελτιστοποίησης που λαμβάνουν τη μορφή “trigger” και “scaling action” όπως στο [98] είναι συνηθισμένες. Στον αντίποδα η συγκεκριμένη αρχιτεκτονική μπορεί να χειριστεί αυτού του είδους τους κανόνες στο scaler.

Για λόγους καθαρά οικονομίας των πόρων, πολλά προϊόντα από πολλαπλούς ενοίκους αναμένεται να φιλοξενούνται σε ένα μόνο Virtual Machine. Η πρακτική αυτή θέτει σοβαρές δυσκολίες στη μέτρηση και τη λογιστική των πόρων και οδηγεί στον ανταγωνισμό για πόρους μέσα σε μια εικονική μηχανή λόγω έλλειψης απομόνωσης. Επιπλέον, το πρόβλημα γίνεται εξαιρετικά περίπλοκο όσον αφορά τη διαθεσιμότητα. Στην προσέγγισή μας, υποθέτουμε ότι ένα προϊόν για έναν συγκεκριμένο μισθωτή είναι εγκατεστημένο σε ένα μόνο Virtual Machine.



Εικόνα 11 Δενδροειδής υπολογισμός πόρων

Ένα προϊόν μπορεί να πάρει τη μορφή μιας στοίβας προϊόντων, όπως ένα Apache Httpd + Debian ή Postges + FreeBSD. Όμως τα εσωτερικά προϊόντα και λογισμικά που τρέχουν μέσα σε ένα Virtual Machine λογίζονται ως ένα μοναδικό προϊόν ή υπηρεσία από την εφαρμογή που τρέχει στο cloud.

3.8 Σύνοψη

Το υπολογιστικό νέφος είναι ένα μοντέλο πληροφορικής που στοχεύει να μετατρέψει το IT σε μια δυναμική, οικονομικά αποδοτική, πάντα διαθέσιμη οντότητα με

μια ευρύτερη υιοθέτηση του ως πραγματική επιχειρηματική λύση. Με την υιοθέτηση της φιλοσοφίας «ως Υπηρεσία» και ενός μοντέλου τιμολόγησης που πληρώνει pay as you go (ή άλλα δυναμικά μοντέλα τιμολόγησης), προκύπτει η νέα έννοια που μπορεί να καταστήσει τον υπολογιστή πιο προσιτό και πιο ευέλικτο στις μικρές και μεσαίες επιχειρήσεις. Αυτό μπορεί να επιτευχθεί με τη χρήση προτύπων και προηγμένων αρχιτεκτονικών σχεδίων, όπως ένα smart cloud middleware, το οποίο διευκολύνει την ανάπτυξη και την προμήθεια των μελλοντικών προϊόντων με δυνατότητα να υποστηρίζονται από τη cloud φιλοσοφία. Αυτή η αρχιτεκτονική παρέχει μηχανισμούς που επιτρέπουν στους προγραμματιστές να αντιμετωπίσουν το μείζον πρόβλημα της ελαστικότητας σε εικονικά διαμορφωμένους πόρους, μιας και είναι μια απαραίτητη μετρική για να συνειδητοποιήσουν οι εμπλεκόμενοι την αποδοτικότητα του κόστους. Στο παρόν κεφάλαιο περιεγράφηκαν οι διαφοροποιήσεις των marketplace που υπάρχουν καθώς και οι ιδιότητες που θα πρέπει να έχουν για να είναι κατάλληλες να υποστηρίξουν τις απαιτήσεις των cloud εφαρμογών. Ταυτόχρονα παρουσιάστηκαν τα βασικά στοιχεία μιας κατάλληλης αρχιτεκτονικής ενδιάμεσου λογισμικού με ιδιαίτερη έμφαση στα απαραίτητα στοιχεία για την υποστήριξη της ελαστικότητας και της τιμολόγησης. Επιπλέον, δείχτηκε ότι η λογική οριζόντιας και κάθετης ελαστικότητας μπορεί να κωδικοποιηθεί εννοιολογικά ως ένας παράγοντας που ελέγχει τη μονάδα εξισορρόπησης φορτίου των προϊόντων με ελάχιστες προσθήκες σε αυτό. Οι κύριες συνεισφορές είναι η περιγραφή μιας γενικής μορφής κανόνων ελαστικότητας, η έννοια του προϊόντος και, τέλος, η αρχιτεκτονική του τοπικού ενδιάμεσου αρχιτεκτονικού μοντέλου στο νέφος. Επιπλέον, αναλύθηκε ο μηχανισμός και οι διαδικασίες για την πραγματοποίηση

αποφάσεων ελαστικότητας χρησιμοποιώντας την αρχιτεκτονική, λαμβάνοντας υπόψη το κόστος. Η σχεδίαση επικεντρώνεται στην ελαστικότητα και τα ζητήματα τιμολόγησης των πόρων σε μια προσπάθεια παροχής μιας λύσης υψηλού επιπέδου για τη διαχείριση των πόρων στις εφαρμογές που τρέχουν στο νέφος. Το κύριο όφελος αυτής της προσέγγισης είναι ότι η δυνατότητα κλιμάκωσης και η τιμολόγηση μιας εφαρμογής μπορούν να επανεκτιμηθούν με διαφάνεια σε ένα γενικό και, ταυτόχρονα, αποτελεσματικό τρόπο για το νέφος.

ΚΕΦΑΛΑΙΟ 4

Σύνθετη επεξεργασία συμβάντων

Ένα από τα σημαντικότερα προβλήματα στη διαχείριση εταιρικών δεδομένων είναι ο διαχωρισμός των βάσεων δεδομένων μεταξύ των επιχειρησιακών βάσεων δεδομένων και δεδομένων αποθήκευσης. Αυτός ο διαχωρισμός έχει πραγματοποιηθεί γιατί κάθε κατηγορία έχει διαφορετικές δυνατότητες σε OLTP και OLAP συστήματα διαχείρισης. Λόγω αυτού του διαχωρισμού, τα αντίγραφα από τις λειτουργικές βάσεις δεδομένων στις αποθήκες δεδομένων θα πρέπει να εκτελούνται περιοδικά. Αυτά τα αντίγραφα έχουν πραγματοποιηθεί από μια διαδικασία Extract-Transform-Load (ETL) που αναλογεί σε ποσό στο 80% του προϋπολογισμού της εκτέλεσης επιχειρηματικών αναλύσεων. Ο επιθυμητός στόχος είναι η αντιμετώπιση αυτού του μεγάλου προβλήματος με την παροχή μιας πλατφόρμας μεγάλων δεδομένων σε πραγματικό χρόνο, η οποία έχει ως κύριο στόχο τις λειτουργίες, την OLTP και την OLAP, σε μία ενιαία λύση διαχείρισης δεδομένων. Για να πραγματοποιηθεί ένα τέτοιο ενδεχόμενο όμως θα πρέπει πριν να έχουν διαχειριστεί τα δεδομένα με τέτοιο τρόπο που να μπορούν να επεξεργαστούν και αποθηκευτούν. Μια παράλληλη και συνάμα κατανεμημένη μηχανή επεξεργασίας δεδομένων που θα μπορεί να κλιμακώνεται ανάλογα με την ανάγκη επεξεργασίας των δεδομένων ροής και που να μπορεί να συνδυαστεί με την επεξεργασία των δεδομένων που είναι αποθηκευμένα για καιρό σε αποθετήρια (data at rest) είναι η πύλη για την επιτυχία ενός τέτοιου εγχειρήματος.

4.1 Εισαγωγή

Το σύγχρονο εταιρικό σύστημα διαχείρισης χρησιμοποιεί δύο κύριες τεχνολογίες για τη διαχείριση δεδομένων, τις βάσεις δεδομένων για πραγματικού χρόνου αναζητήσεις που έχουν να κάνουν με τις τρέχουσες επιχειρησιακές ανάγκες (operational databases) και τις αποθήκες/αποθετήρια δεδομένων (data warehouses). Οι λειτουργικές βάσεις δεδομένων παρέχουν συνέπεια για δεδομένα που ενημερώνονται συχνά. Αυτές οι εγγυήσεις συνέπειας είναι γνωστές ως ιδιότητες ACID ή εγγυήσεις συναλλαγών. Οι επιχειρησιακές βάσεις είναι συστήματα επεξεργασίας συναλλαγών (OLTP) που είναι σε θέση να επεξεργάζονται σύντομες συναλλαγές με ένα μεγάλο τμήμα των ανανεωμένων συναλλαγών. Τα αποθετήρια δεδομένων χρησιμοποιούνται κυρίως για ερωτήματα που έχουν να κάνουν με επιχειρηματική ανάλυση της εταιρίας ή του οργανισμού. Αυτά είναι σε θέση να απαντήσουν σε μεγάλα αναλυτικά ερωτήματα σε χρόνο που πλησιάζει το άμεσο και γι' αυτό είναι συνήθως συστήματα αναλυτικής επεξεργασίας (OLAP).

Δυστυχώς, και τα δύο συστήματα δεν είναι αποδοτικά στην επεξεργασία των αντίθετων διαδικασιών. Γι' αυτό το λόγο οι επιχειρήσεις πρέπει να έχουν και τα δύο είδη βάσεων δεδομένων. Αυτό έχει ως αποτέλεσμα να χρειάζεται περιοδικά αντιγραφή των δεδομένων από τις λειτουργικές βάσεις δεδομένων στα αποθετήρια δεδομένων. Η σωστή διοίκηση των δεδομένων είχε ως κύριο στόχο την επίλυση αυτού του προβλήματος στη διαχείριση εταιρικών δεδομένων με τη σύλληψη ενός νέου συστήματος διαχείρισης

δεδομένων που είναι σε θέση να λύσει και τα δύο είδη φόρτου εργασίας, αποφεύγοντας έτσι την ανάγκη για ETLs.

Παρόλο που μπορούν να επιλυθούν τέτοιου είδους προβλήματα με συγκεκριμένου είδους βάσεις δεδομένων, η διαχείριση των ερωτημάτων και των δεδομένων καθώς και ο ρυθμός και ο τρόπος που αποστέλλονται παραμένει ένα μεγάλο θέμα ειδικά εάν είναι πολλά τα δεδομένα που καλείται να δεχτεί, είτε εάν είναι απροσδιόριστος ο ρυθμός δεδομένων που μπορεί να λάβει.

4.2 Σύνθετη Επεξεργασία Συμβάντων

Η σύνθετη επεξεργασία συμβάντων (CEP) είναι ένα νέο παράδειγμα για την ανάλυση σε δεδομένα πραγματικού χρόνου που καταγράφονται από ετερογενείς προελεύσεις δεδομένων. Αντί να αποθηκεύει τα δεδομένα και στη συνέχεια να τα επεξεργάζεται, τα δεδομένα υποβάλλονται σε επεξεργασία μόλις παραληφθούν, είτε στη χειρότερη περίπτωση βάση ενός παραθύρου δεδομένων που αποθηκεύεται στη μνήμη. Τα CEP ερωτημάτων είναι συνεχή ερωτήματα που τρέχουν σε μία ροή γεγονότων. Τα συνεχόμενα ερωτήματα μοντελοποιούνται ως γραφήματα όπου οι κόμβοι είναι τελεστές CEP και τα βέλη είναι ροή συμβάντων. Οι χειριστές CEP (CEP operators) είναι υπολογιστικά κουτιά που επεξεργάζονται τα συμβάντα που έχουν περάσει πάνω από την εισερχόμενη ροή και παράγουν συμβάντα εξόδου στις εξερχόμενες ροές. Οι χειριστές CEP μπορούν να είναι είτε stateless είτε stateful, ανάλογα με το αν λειτουργούν στην τρέχουσα κατάσταση (πλειάδα) ή σε ένα σύνολο συμβάντων (παράθυρο). Τέτοιου είδους υλοποιήσεις έχουν παρουσιαστεί [99] [100] και [101].

Αργότερα παρουσιάστηκαν λύσεις σαν το Storm [102] και το S4 [103] που ακολούθησαν μια παρόμοια προσέγγιση με αυτή του StreamCloud [104] για την επίλυση προβλημάτων κλιμάκωσης των συστημάτων CEP, προκειμένου να είναι σε θέση να επεξεργάζονται σε πραγματικό χρόνο την αύξηση του ποσού των δεδομένων που παράγονται καθημερινά. Με αυτά τα συστήματα, ένα CEP μπορεί να τρέξει σειρά ερωτημάτων με κατανεμημένο και παράλληλο τρόπο πάνω από αρκετές μηχανές, που με αυτό το τρόπο μπορεί να αυξήσει τον αριθμό των δεδομένων που μπορεί να επεξεργαστεί ανά δευτερόλεπτο. Ταυτόχρονα με τη χρήση παράλληλης επεξεργασίας μπορεί να προσεγγίσει υψηλότερη ταχύτητα μετάδοσης χρησιμοποιώντας λιγότερους πόρους. Το κύριο μειονέκτημα αυτών των συστημάτων έγκειται στις τακτικές προσεγγίσεις δικτύωσης που χρησιμοποιούνται για την επικοινωνία (περίπου το 50% της CPU χρησιμοποιείται για δικτύωση).

Η προτεινόμενη προσέγγιση που επιλέχθηκε (LeanBigData CEP Engine [105]) έχει ως στόχο να ξεπεράσει το υψηλό θέμα της υπερ-επιβάρυνσης της επικοινωνίας που υπέστησαν οι τρέχουσες λύσεις, παρέχοντας υψηλή δυνατότητα κλιμάκωσης χρησιμοποιώντας ένα προσαρμοζόμενο επίπεδο επικοινωνίας. Δεδομένου ενός παράλληλου κατανεμημένου ερωτήματος, το CEP θα διασυνδέσει τα δευτερεύοντα ερωτήματα με το κανάλι επικοινωνίας που ταιριάζει καλύτερα στο συγκεκριμένο σενάριο. Ξεκινώντας από τις κλήσεις διαδικασιών για δευτερεύοντα ερωτήματα που εκτελούνται στο ίδιο νήμα και φτάνοντας μέχρι Remote Direct Memory Access που έχει σαν δίκτυο InfiniBand για δευτερεύοντα ερωτήματα που εκτελούνται σε διαφορετικούς υπολογιστές.

Οι κύριοι στόχοι του σχεδιασμού είναι η δυνατότητα κλιμάκωσης και η ευελιξία. Για το σκοπό αυτό, ο σχεδιασμός της αρχιτεκτονικής στοχεύει στην υλοποίηση ενός μηχανισμού CEP που μπορεί να κλιμακώνονται σε τρεις διαστάσεις:

- **Όγκος ροής:** το CEP πρέπει να είναι σε θέση να διαχειρίζεται αυτά τα σενάρια όπου τεράστιο μέρος των μεγάλων ποσοτήτων που μεταφέρονται σε ένα συγκεκριμένο ρεύμα δεδομένων πρέπει να υποβληθούν σε επεξεργασία με πολύ χαμηλό latency.

- **Αριθμός ερωτημάτων:** το CEP πρέπει να αξιοποιήσει τους πόρους του συστήματος στο περιβάλλον ανάπτυξης, προκειμένου να εκτελέσει όσο το δυνατόν μεγαλύτερο αριθμό ταυτόχρονων ερωτημάτων.

- **Πολυπλοκότητα των ερωτημάτων:** το σύστημα βελτιστοποίησης του CEP των πόρων του συστήματος θα μπορεί να εκτελέσει ερωτήματα που αποτελούνται από έναν τελεστή, καθώς και από ερωτήματα που αποτελούνται από αρκετούς τελεστές.

Νέες αρχιτεκτονικές που μπορούν να εκμεταλλευτούν πολλούς πυρήνες αλλά και μη ομοιόμορφη πρόσβαση στη μνήμη μας επιτρέπουν να μεγιστοποιήσουμε την αξιοποίηση των πόρων στους υπολογιστές όπου αναπτύσσεται. Το CEP θα ξεκινήσει ένα νήμα/διεργασία ανά πυρήνα για να εκμεταλλευτεί την τοπική μνήμη της CPU και, ταυτόχρονα, να ελαχιστοποιήσει το context switch μεταξύ νημάτων/διεργασιών.

Η ελαστικότητα και η εξισορρόπηση φόρτου είναι χαρακτηριστικά που απαιτούνται για να μπορεί το CEP να επαναπαραμετροποιηθεί. Για παράδειγμα, δεδομένου ενός cluster CEP που αποτελείται από "n" μηχανήματα που τρέχουν "m"

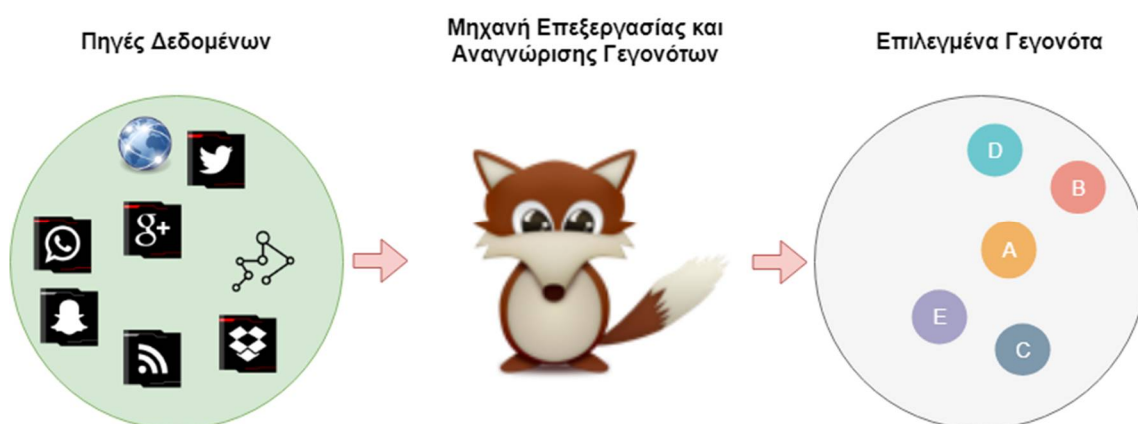
ερωτήματα, ο στόχος της ελαστικότητας είναι να εκτελεστεί κάθε ερώτημα στον κατάλληλο αριθμό των μηχανημάτων που απαιτούνται για την εκπλήρωση της ποιότητας της υπηρεσίας που ορίζεται από το χρήστη. Η εξισορρόπηση φόρτου χρησιμοποιείται όταν το σύστημα εντοπίζει ότι η εκ νέου ρύθμιση των εσωτερικών επικοινωνιών των ερωτημάτων ενός παράλληλου κατανεμημένου ερωτήματος μπορεί να βελτιώσει την απόδοση του ερωτήματος που εκτελείται χωρίς να προστεθεί κάποιο νέο μηχάνημα. Τόσο η ελαστικότητα όσο και η εξισορρόπηση φορτίου έχουν την απαίτηση, ότι πρέπει να ελαχιστοποιηθεί ο αντίκτυπος των ενεργειών επαναδιαμόρφωσης σε κανονική επεξεργασία. Το CEP ακόμα μπορεί να βελτιώσει τη διαχείριση της μνήμης και να μειώσει την αναποτελεσματικότητα του garbage collection με την εφαρμογή τεχνικών όπως η επαναχρησιμοποίηση αντικειμένων. Τέλος, η εφαρμογή των τυποποιημένων διαχειριστών CEP που ορίζονται στο [106], [107] θα περιλαμβάνει επίσης τον σχεδιασμό προσαρμοσμένων operators, όπως οι τελεστές αποθήκευσης δεδομένων (data store operators). Οι τελεστές αποθήκευσης δεδομένων επιτρέπουν στο CEP να συσχετίζει συμβάντα ροής με αποθηκευμένα δεδομένα, για παράδειγμα, σε χώρους όπως key-value data stores.

4.3 Αρχιτεκτονική

Σκοπός είναι να σχεδιάσει και να αναπτυχθεί μια υποδομή λογισμικού ικανή να ανιχνεύει και να εντοπίζει σημαντικά γεγονότα που συνέβησαν εξ ολοκλήρου σε ένα δεδομένο σύστημα ή σε όλα τα συστήματα σε πραγματικό χρόνο. Αυτή η υποδομή πραγματοποιείται από ένα μηχανισμό επεξεργασίας ροής (SPE) για κατανεμημένο

υπολογισμό υψηλής απόδοσης και έναν μηχανισμό σύνθετης επεξεργασίας συμβάντων (CEP) που μπορεί να εκτελέσει ερωτήματα που μοιάζουν με SQL στο SPE. Από εδώ και πέρα θα αναφέρονται οι δύο αυτοί μηχανισμοί ως CEP.

Κοιτάζοντας το CEP ως ένα μαύρο κουτί, λαμβάνει συνεχώς ακατέργαστα δεδομένα από εξωτερικές πηγές δεδομένων και αυτό αντιλαμβάνεται τα γεγονότα όπου και τα κατηγοριοποιεί. Η διαδικασία αυτή γίνεται συνήθως με τη χρήση αλγορίθμων pattern matching στα πρωτογενή δεδομένα (Εικόνα 12).



Εικόνα 12 Complex event processing

Οι προελεύσεις δεδομένων τροφοδοτούν συνεχώς το CEP με συμβάντα που περιέχουν ακατέργαστα δεδομένα που εξάγονται από πηγές πληροφοριών. Όταν απαιτείται, αυτά τα συμβάντα προ-υποβάλλονται σε επεξεργασία, ώστε να μπορούν να χρησιμοποιηθούν από το CEP. Μια συνηθισμένη ενέργεια προ-επεξεργασίας είναι ο μετασχηματισμός της μορφής συμβάντος. Τα γεγονότα που έχουν μετατραπεί σε μια

μορφή που είναι έτοιμη για CEP ονομάζονται “πλειάδα”(tuple). Τα δομικά στοιχεία των αλγορίθμων που ταιριάζουν με τα μοτίβα που εφαρμόζονται από το CEP είναι απλές λειτουργίες όπως η προβολή, η συνάθροιση και η συσχέτιση.

Οι χρήστες του CEP μπορούν να εκφράσουν τα αιτήματά τους χρησιμοποιώντας μια γλώσσα που μοιάζει με SQL, η οποία θα μεταφράσει αυτόματα αυτά σε συνεχόμενα ερωτήματα CEP. Τα ερωτήματα CEP είναι κυκλικά γραφήματα των τελεστών ροής που μπορούν να εντοπίσουν το μοτίβο ενδιαφέροντος συνδυάζοντας διάφορες λειτουργίες, όπως το φιλτράρισμα, η συσσώρευση, η ομαδοποίηση και η συσχέτιση. Οι τελεστές ροής ταξινομούνται σε τρεις κύριες ομάδες σύμφωνα με τη λειτουργία που εκτελούν σε διάφορα σύνολα, όπως σε:

- **Τελεστές προσανατολισμένους σε γεγονότα:** η έξοδος του υπολογισμού εξαρτάται μόνο από την πραγματική πλειάδα. Αυτή η κατηγορία παρέχει λειτουργίες όπως η προβολή και το φιλτράρισμα. Η προβολή χρησιμοποιείται όταν μια πλειάδα πρέπει να μετατραπεί σε πλειάδα με διαφορετικό αριθμό/τύπο πεδίων. Το φιλτράρισμα χρησιμοποιείται συνήθως για την επιλογή των τιμών από συνεχείς ροές σύμφωνα με τις τιμές που είναι αποθηκευμένες στα πεδία τους.

- **Τελεστές προσανατολισμένους στα παράθυρα:** η έξοδος του υπολογισμού εξαρτάται μόνο από το σύνολο των πλειάδων που συνθέτουν την τρέχουσα κατάσταση. Η κατάσταση του διατηρείται είτε σε χρονικά παράθυρα είτε σε χωρικά παράθυρα. Προηγείται η αποθήκευση και το ξεσκαρτάρισμα και στη συνέχεια σύμφωνα

με τη χρονοσφραγίδα τους, λαμβάνει αποφάσεις παρατηρώντας τη σειρά άφιξης των πλειάδων.

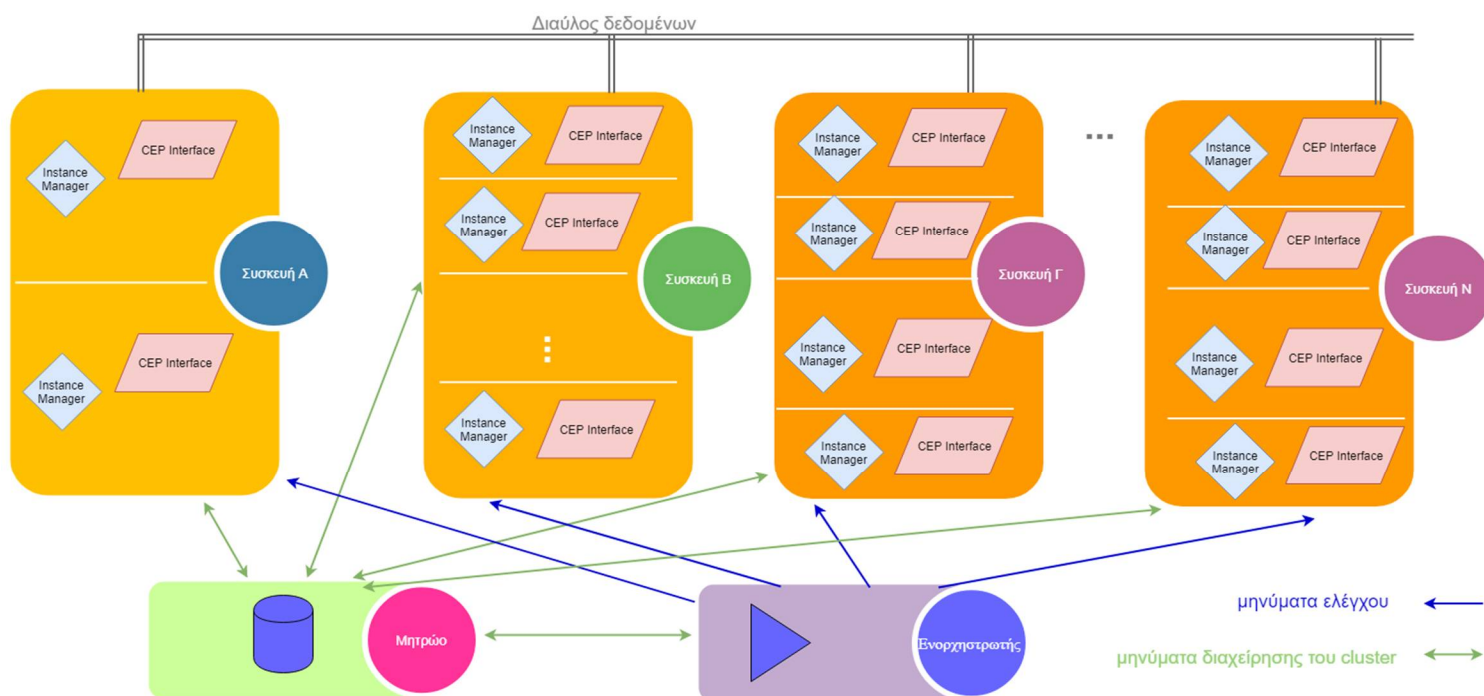
- **Τελεστές προσανατολισμένους στα δεδομένα:** η έξοδος του υπολογισμού, εάν υπάρχει, εξαρτάται από την πραγματική πλειάδα συν το αποτέλεσμα ενός ερωτήματος που εκτελείται σε εξωτερικούς χώρους αποθήκευσης δεδομένων. Αυτοί οι τελεστές επιτρέπουν σε συνεχόμενα ερωτήματα να λαμβάνουν αποφάσεις βάσει ιστορικών δεδομένων καθώς και να υλοποιούν ενδιάμεσα ή/και τελικά αποτελέσματα ερωτημάτων CEP.

4.4 Παραλληλοποίηση του Complex Event Processing

Συμπλέγματα CEP είναι το σύνολο των μηχανημάτων που διατίθενται για την εκτέλεση του μηχανισμού λειτουργίας CEP. Στην Εικόνα 13 εμφανίζεται μια υψηλού επιπέδου παρουσίαση ενός συμπλέγματος CEP που επισημαίνει τα κύρια στοιχεία της αρχιτεκτονικής που είναι ο ενορχηστρωτής CEP (CEP Orchestrator), ο Διαχειριστής των instances (Instance Manager), τα CEP Interfaces και το μητρώο (Registry).

Τα CEP instances είναι τα βασικά στοιχεία της αρχιτεκτονικής. Ένα instance είναι μια διαδικασία που μπορεί να εκτελέσει τους τελεστές ροής που συνθέτουν ένα συνεχές ερώτημα ή δευτερεύον ερώτημα (τμήμα ενός ερωτήματος). Τα CEP instances είναι πολύ ευέλικτες δομές που έχουν φτιαχτεί με αυτό ακριβώς το στόχο. Μπορούν να

χρησιμοποιηθούν για την εκτέλεση πολλών ερωτημάτων σε ένα μόνο νήμα, καθώς και για την εκτέλεση ενός μεμονωμένου τελεστή χρησιμοποιώντας μια ουρά/στοίβα νημάτων.



Εικόνα 13 Αρχιτεκτονική για παράλληλη διαχείριση του CEP

Ο Instance Manager φροντίζει για τη ρύθμιση των παραμέτρων των CEP interfaces, όπως απαιτείται. Επίσης, αντιστοιχίζει τα ερωτήματα/υπό-ερωτήματα που λαμβάνονται από τον ενορχηστρωτή στα CEP interfaces.

Ο ενορχηστρωτής CEP είναι υπεύθυνος για τη διανομή ερωτημάτων/δευτερευόντων ερωτημάτων μεταξύ των Instance Managers που είναι

διαθέσιμα στο cluster και την παρακολούθηση αυτού για αποτυχίες και συμφορήσεις επιδόσεων. Τέλος, το μητρώο είναι ένα σύμπλεγμα βασισμένο στο Apache Zookeeper [108] που χρησιμοποιείται για τον συντονισμό του ενορχηστρωτή με τα CEP Instances και επίσης για να αποθηκεύει όλες τις παραμέτρους και τις ρύθμισης παραμέτρων που απαιτούνται από κάθε CEP Instance για την εκτέλεση των ερωτημάτων τους.

4.5 Ροές δεδομένων

Οι συνεχόμενες ροές ερωτημάτων είναι μη κυκλικά γραφήματα, όπου οι κόμβοι είναι τελεστές ροών και οι ακμές είναι οι ροές. Σε ένα σύμπλεγμα(cluster) CEP ένα συγκεκριμένο ερώτημα μπορεί να εκτελεστεί χρησιμοποιώντας διαφορετικές υλοποιήσεις. Στην απλούστερη ανάπτυξη, ολόκληρο το ερώτημα εκτελείται από το ίδιο μοναδικό νήμα (thread) ενός CEP instance. Εάν το ερώτημα χρειάζεται πρόσθετη επεξεργαστική ισχύ, μπορεί να εξακολουθήσει να αναπτύσσεται επιτρέποντας στο συγκεκριμένο CEP instance να παραγάγει περισσότερα νήματα για την εξυπηρέτηση του ερωτήματος. Επιπλέον, πολύ πολύπλοκα και απαιτητικά συνεχόμενα ερωτήματα μπορεί να διαιρεθούν σε δευτερεύοντα ερωτήματα, όπου κάθε δευτερεύον ερώτημα εκτελείται σε μια διαφορετικό CEP instance που βρίσκεται στο ίδιο μηχάνημα ή διανέμεται μεταξύ του συμπλέγματος του CEP. Τα CEP instances επικοινωνούν μεταξύ τους χρησιμοποιώντας ροές από tuples. Οι ροές μεταφέρουν τα tuples που παράγονται από την έξοδο ενός χειριστή στον ακόλουθο κατά σειρά χειριστή του γραφήματος. Λόγω της ποικιλίας και της πολυπλοκότητας των διαφορετικών αναπτύξεων που είναι διαθέσιμες για ένα ερώτημα, οι ροές πρέπει να έχουν τη δυνατότητα να μεταφέρουν τα tuples μεταξύ των CEP instances που εκτελούνται είτε

στο ίδιο νήμα, είτε στον ίδιο υπολογιστή είτε μεταξύ των διεργασιών. Για το σκοπό αυτό, ορίζονται τρεις κατηγορίες ροής:

- **Τοπική ροή δεδομένων:** ροή που χρησιμοποιείται για την αποστολή tuples στα CEP instances που τρέχουν στο ίδιο νήμα.

- **Μετάδοση ροής μεταξύ thread:** ροή που χρησιμοποιείται για την αποστολή των tuples σε CEP instances που εκτελούνται στον ίδιο υπολογιστή.

- **Ροή από το δίκτυο:** ροή που χρησιμοποιείται για την αποστολή tuples στα CEP instances που εκτελούνται σε διαφορετικούς υπολογιστές.

Η πρώτη κατηγορία είναι η μόνη κατηγορία που δεν χρειάζεται σειριοποιήσει και αποσειριοποιήσει της πλειάδα κατά τη διάρκεια της επικοινωνίας. Όλες οι κατηγορίες ροής έχουν την απαίτηση ότι οι πλειάδες που μεταφέρονται μεταξύ αυτών να έχουν το ίδιο σχήμα. Ένα σχήμα είναι το σύνολο των πεδίων που καθορίζουν μια πλειάδα. Αξίζει να σημειωθεί ότι οι χειριστές και οι ροές είναι εντελώς αποσυνδεδεμένες, δηλαδή η εργασία του χειριστή δεν αλλάζει όταν διαφοροποιούνται οι επιλογές για τη διάσχιση και τη κατεύθυνση της ροής που επιλέγεται για την επικοινωνία. Ο Instance Manager είναι το δομικό στοιχείο που είναι υπεύθυνο για τη ρύθμιση των παραμέτρων των CEP instances με τις κατηγορίες των ροών που αποφασίζονται από τον ενορχηστρωτή.

Εκτός των τριών κατηγοριών που παρουσιάστηκαν μπορεί η συγκεκριμένη υλοποίηση να επιτρέπει επίσης τον ορισμό των προσαρμοσμένων χειριστών. Στην παρακάτω ενότητα παρέχονται λεπτομέρειες για τους προκαθορισμένους τελεστές που

επικεντρώνονται τόσο στην κύρια λειτουργία που παρέχουν όσο και στις παραμέτρους διαμόρφωσης τους.

4.6 Τελεστές Γεγονότων

Οι τελεστές που απευθύνονται σε γεγονότα χρησιμοποιούνται για να φιλτράρουν και να μετατρέπουν μεμονωμένες στοιβάδες. Οι εξαγόμενες στοιβάδες, εάν υπάρχουν, εξαρτώνται μόνο από τα δεδομένα που περιέχονται στην τρέχουσα πλειάδα.

4.6.1 Τελεστής Map

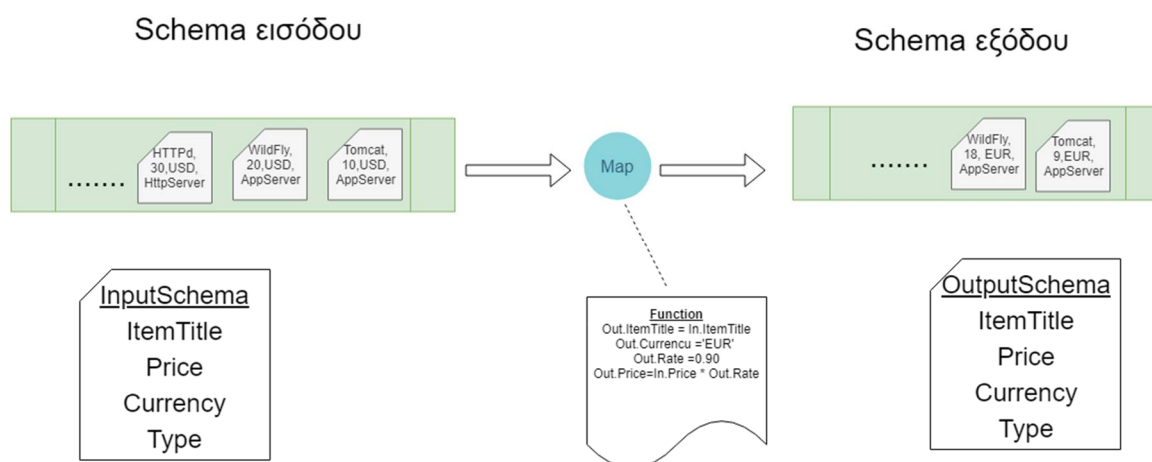
- **Λογική:** ο τελεστής συσχέτισης συνδέει κάθε πλειάδα εισόδου με πλειάδα εξόδου όπου πάνω σε αυτή εφαρμόζει κάποιους μετασχηματισμούς πεδίου. Ζεύγη παραμέτρων προσδιορίζουν τον τρόπο με τον οποίο τα πεδία των εισόδων μετατρέπονται στα αντίστοιχα πεδία της πλειάδας εξόδου. Ταυτόχρονα απαιτείται τουλάχιστον ένας μετασχηματισμός να γίνει για να είναι ορθή αυτή η πράξη.

- **Ρύθμιση παραμέτρων:** για κάθε πεδίο της ροής εξόδου, ο τελεστής πρέπει να ρυθμιστεί με ένα ζεύγος που καθορίζει το όνομα του πεδίου εξόδου και μια έκφραση πάνω στη ροή εισόδου.

- **Είσοδος:** μία ροή εισόδου

- **Έξοδος:** μία ροή εξόδου. Το σχήμα (schema) της ροής εξόδου μπορεί να είναι διαφορετικό από το σχήμα της ροής εισόδου και συνάγεται από τους μετασχηματισμούς που χρησιμοποιούνται από το τελεστή map.

Παράδειγμα: μετατροπή κάθε τιμής των προϊόντων που αφορούν server για υπηρεσίες που είναι για τη δόμηση εφαρμογών από δολάρια ΗΠΑ (USD) που είναι κοστολογημένα σε ευρώ. Υποθέτοντας 1 USD να είναι ίσο με 0.90 ευρώ



Εικόνα 14 Παράδειγμα του Τελεστή Map

4.6.2 Τελεστής Filter

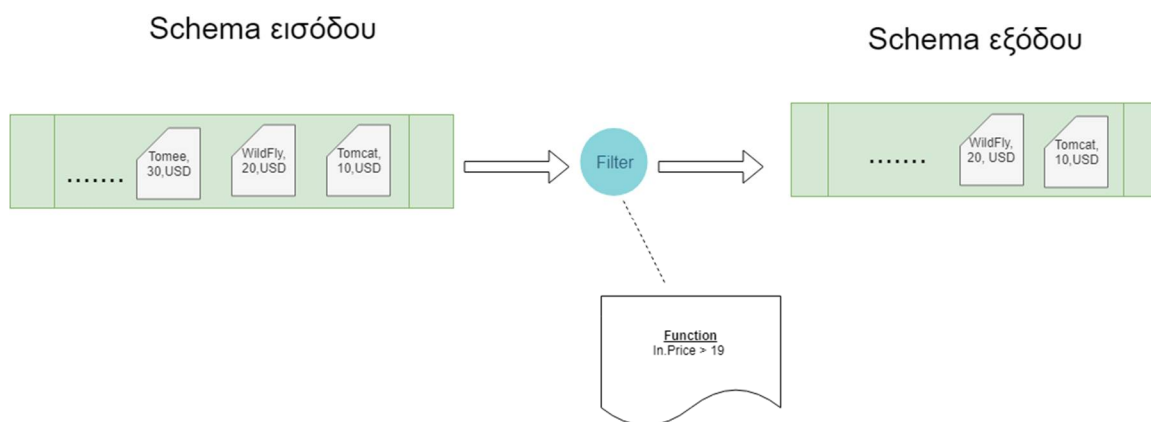
- **Λογική:** ο τελεστής φίλτρου αποδίδει στη ροή εξόδου όλα τα στοιχεία εισόδου που ικανοποιούν ένα κατηγορήμα. Οι στιβάδες που δεν ικανοποιούν το κατηγορήμα απορρίπτονται.

- **Διαμόρφωση:** μια δυαδική έκφραση πάνω από τη ροή εισόδου.

- **Είσοδος:** μία ροή εισόδου.

- **Εξόδος:** μία ροή εξόδου με το ίδιο σχήμα της ροής εισόδου.

Παράδειγμα: Αφαιρέστε από τη ροή όλα τα προϊόντα των οποίων η τιμή είναι μικρότερη από 19 Δολάρια. Τέτοια ερωτήματα μπορούν να φανούν ιδιαίτερα χρήσιμα σε συστήματα επιλογής φθηνότερης τιμής, είτε ακολουθώντας το απλό παράδειγμα χρησιμοποιώντας ένα τελεστή είτε συνδυασμό αυτών σε κάθε χαρακτηριστικό που απαρτίζει τη τιμή.



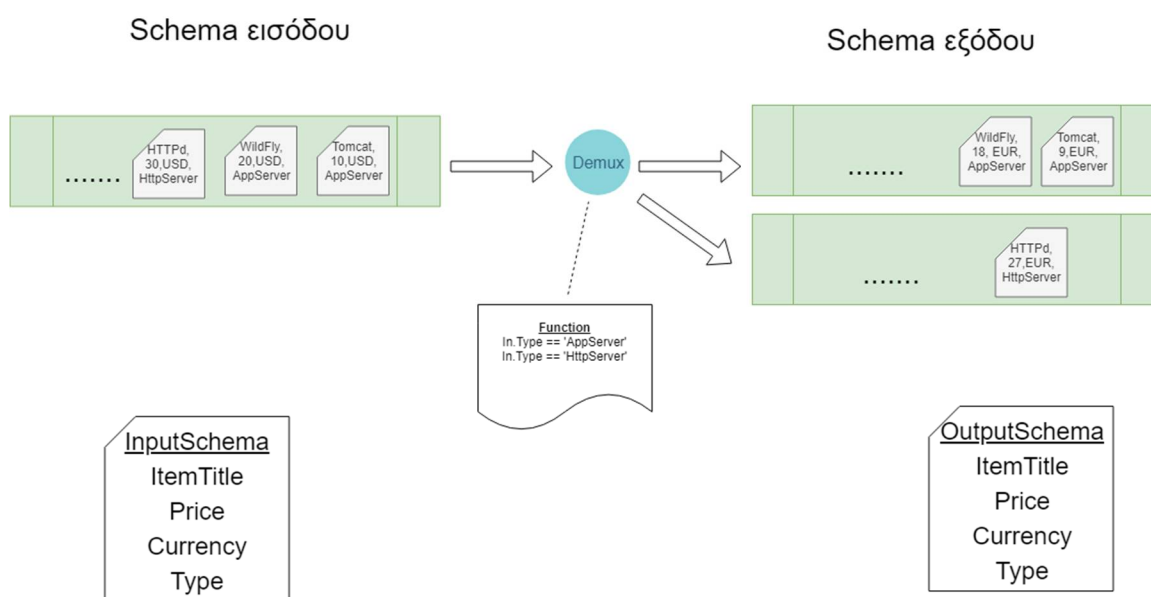
Εικόνα 15 Παράδειγμα του τελεστή Filter

4.6.3 Τελεστής Demux

- **Λογική:** ο τελεστής Demux αποστέλλει την πλειάδα εισόδου σε όλες τις ροές εξόδου των οποίων το κατηγορήμα ικανοποιείται.
- **Ρύθμιση παραμέτρων:** μια λίστα με εκφράσεις δυαδικής τιμής, μία για κάθε ροή εξόδου
- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** μια ροή εξόδου για κάθε κατηγορία όλα με το ίδιο schema της ροής εισόδου.

Παράδειγμα: διαίρεση μιας ροής στοιχείων με διαφορετικά τύπου προϊόντα που μπορεί να αντιπροσωπεύουν είτε να χρησιμεύουν σε διαφορετικού είδους προβλήματα είτε υλοποιήσεις.



Εικόνα 16 Παράδειγμα του τελεστή Demux

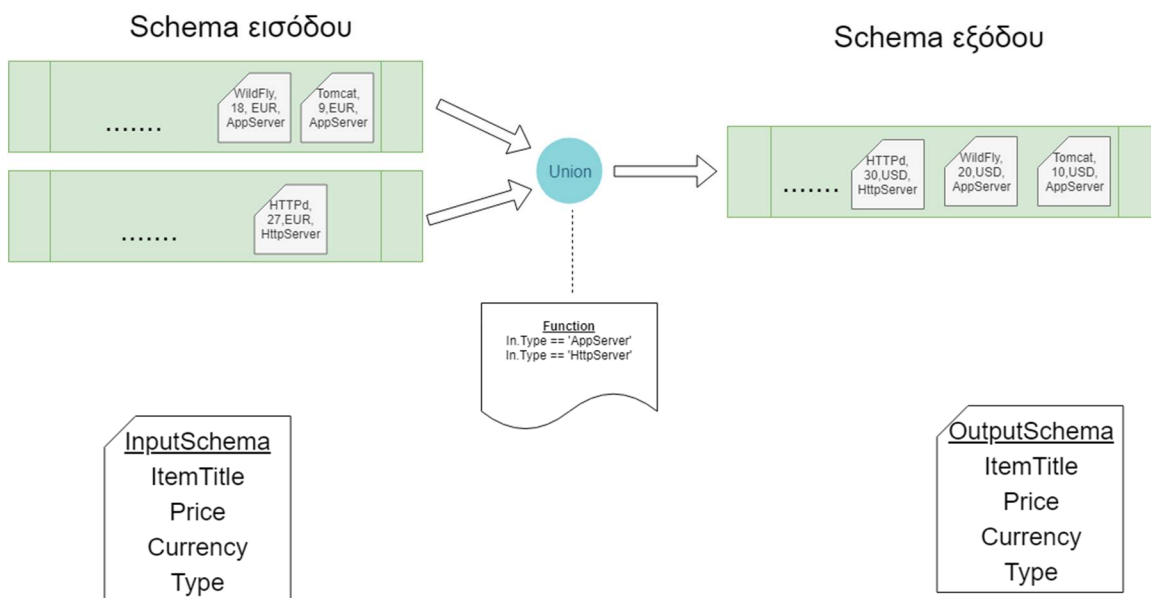
4.6.4 Τελεστής Union

- **Λογική:** ο διαχειριστής της Ένωσης συγχωνεύει τα στοιχεία πολλών ροών εισόδου με το ίδιο σχήμα σε μια μεμονωμένη ροή εξόδου.

- **Ρύθμιση παραμέτρων:** δεν υπάρχει ρύθμιση παραμέτρων για αυτόν τον τελεστή.

- **Είσοδος:** μία ή περισσότερες ροές με το ίδιο σχήμα.
- **Έξοδος:** μία ροή εξόδου με το ίδιο σχήμα με τις εισροές.

Παράδειγμα: συγχώνευση ροών στοιχείων με διαφορετικό τύπο server σε μία ροή που περιλαμβάνει όλα τα στοιχεία σε μια κοινή ροή.



Εικόνα 17 Παράδειγμα του τελεστή Union

4.7 Τελεστές προσαρμοσμένοι με χρήση παραθύρου

Οι τελεστές που ακολουθούν το μοντέλο της χρήσης παραθύρου παράγουν έναν ορισμένο αριθμό από πλειάδες που βασίζεται σε μια ροή πλειάδων που αποθηκεύονται προσωρινά σε μια δομή μνήμης που ονομάζεται συρόμενο παράθυρο. Συρόμενα παράθυρα

αποθηκεύουν πλειάδες σύμφωνα με χωρικές ή χρονικές συνθήκες. Συγκεκριμένα, τα συρόμενα παράθυρα ορίζονται από τρεις παραμέτρους:

- **Μέγεθος του παραθύρου:** ορίζει τη χωρητικότητα του παραθύρου.
- **Πρόοδος:** καθορίζει πόσο σύρεται το παράθυρο όταν αυτό γεμίσει.
- **Τύπος παραθύρου:** καθορίζει τον τρόπο ολίσθησης του παραθύρου, είτε με βάση το χρόνο είτε με τον αριθμό των αποθηκευμένων πλειάδων.

Για τα χωρικά παράθυρα, η παράμετρος μεγέθους αντιπροσωπεύει το μέγιστο αριθμό των πλειάδων που μπορούν να διατηρηθούν στο παράθυρο. Για τα χρονικά παράθυρα, η μονάδα μέτρησης είναι ο χρόνος και συνήθως τα δευτερόλεπτα όπου το παράθυρο μετατοπίζεται και λαμβάνει τη νέα πλειάδα απομακρύνοντας τη παλαιά τη στιγμή που η διαφορά στο χρόνο μεταξύ της νεότερης πλειάδας και της παλαιότερης είναι μεγαλύτερη από το μέγεθος του παραθύρου. Τόσο για τα χρονικά όσο και για τα χωρικά συρόμενα παράθυρα μοιραία θα πρέπει να γίνει απομάκρυνση των παλαιότερων πλειάδων για να μπορέσει να υπάρξει χώρος για τις νεότερες πλειάδες όταν αυτός ο αριθμός φτάσει να αντιστοιχεί στο μέγιστο μέγεθος του παραθύρου πριν από τη ολίσθηση.

4.7.1 Τελεστής Aggregate

- **Λογική:** ο τελεστής συγκεντρωτικών αποτελεσμάτων αξιολογεί μία ή περισσότερες συναρτήσεις συγκεντρωτικών αποτελεσμάτων πάνω από συρόμενα παράθυρα που ορίζονται στη ροή εισόδου του. Κατορθώνει να εξάγει το τρέχον άθροισμα

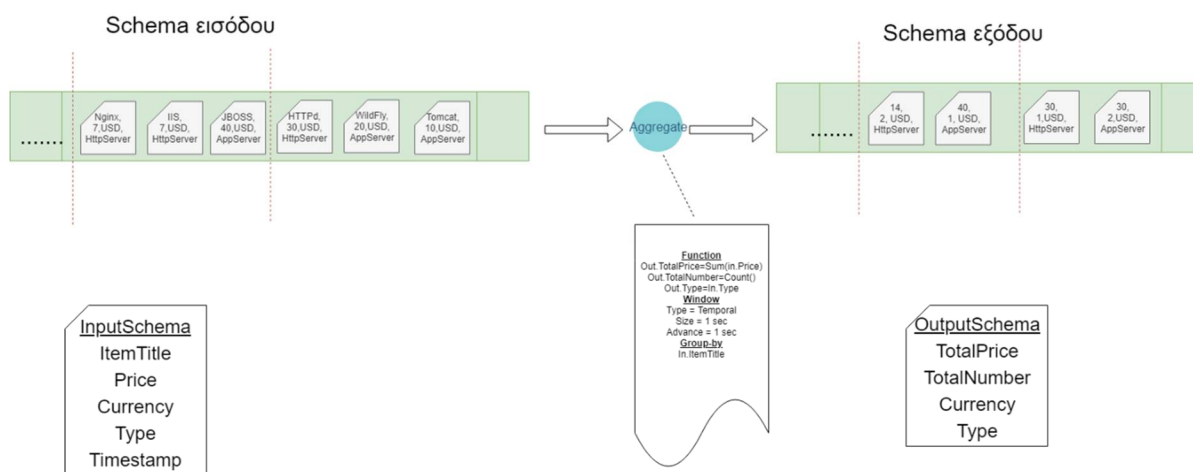
λίγο πριν από τη ολίσθηση του παραθύρου. Ο τελεστής αποτελείται τουλάχιστον από μία συνάρτηση συγκεντρωτικών αποτελεσμάτων.

- **Ρύθμιση παραμέτρων:** για κάθε πεδίο της ροής εξόδου, το άθροισμα πρέπει να ρυθμιστεί με ένα ζεύγος που καθορίζει το όνομα του πεδίου εξόδου και μια συνάρτηση συγκεντρωτικών αποτελεσμάτων. Επιπλέον, οι παράμετροι για τη ρύθμιση του συρόμενου παραθύρου είναι απαραίτητες (τύπος, μέγεθος και πρόοδος) εκτός από τις προαιρετικές παραμέτρους group-by, στη περίπτωση που απαιτείται ομαδοποίηση.

- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** μία ροή εξόδου της οποίας το σχήμα εξαρτάται από την έξοδο που παράγεται από τις συναρτήσεις συγκεντρωτικών αποτελεσμάτων.

Παράδειγμα: Στη περίπτωση που έχουμε μια ροή από κόστη από υπηρεσίες cloud που αποστέλλουν την αξία των υπηρεσιών που προσφέρονται στο πελάτη από στιγμή που



Εικόνα 18 Παράδειγμα του τελεστή Aggregate

χρησιμοποιούνται μπορεί ο τελεστής aggregate να υπολογίσει βασισμένος στο προκαθορισμένο από το χρήστη παράθυρο το συνολικό κόστος υπηρεσιών ανά τύπο υπηρεσίας. Η επιλογή του παραθύρου μπορεί να καθοριστεί σε τέτοια χρονική περίοδο ώστε να μπορεί ο χρήστης να βγάλει σημαντικά συμπεράσματα για τις επιλογές που έχει κάνει. Αντίστοιχα εάν αυτό χρησιμοποιείται σαν είσοδο σε ένα σύστημα ελέγχου για την καλύτερη επιλογή υπηρεσιών (service monitoring) είτε σε ένα ΑΙ σύστημα που προσαρμόζεται ανάλογα με τις ανάγκες ή τη πρότερη γνώση αυτός ο χρόνος θα μπορούσε να μειωθεί αρκετά μιας και είναι ικανός να επεξεργαστεί από ένα μηχάνημα/μηχανισμό επιλογής βέλτιστης λύσης. Οι συγκεντρωτικές αυτές λύσεις μπορούν να διευκολύνουν στη καλύτερη παρουσίαση και ανάλυση μιας μέτρησης που θα έχει φυσικό νόημα για το τελικό αποδέκτη.

4.7.2 Τελεστής Join

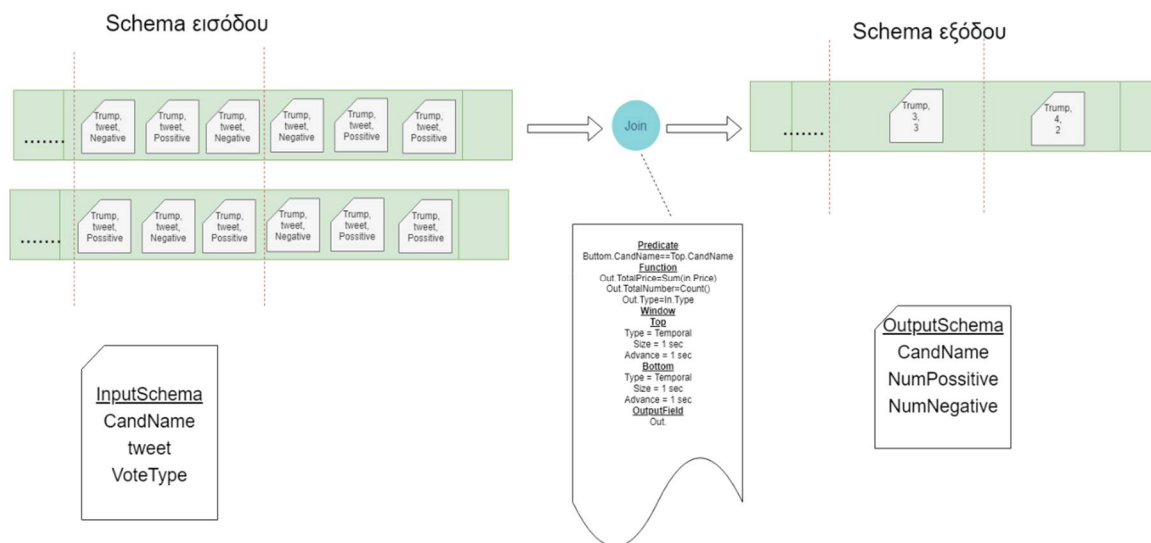
- **Λογική:** ο τελεστής συνδέσμου συσχετίζει τα στοιχεία από δύο ροές εισόδου. Χρησιμοποιώντας δύο συρόμενα παράθυρα ζευγαρώνει κάθε πλειάδα από μια ροή με τις πλειάδες από την άλλη ροή να διατηρούνται. Για να είναι σωστά δομημένο και καθορισμένο, ο τελεστής χρησιμοποιεί μόνο χρονικά παράθυρα και πριν από την αντιστοίχιση των στοιχείων, μετακινεί το παράθυρο σύμφωνα με τη χρονική σήμανση που ελήφθη στη ροή εισόδου.

- **Ρύθμιση παραμέτρων:** ένα κατηγορημα πάνω από τις ροές εισόδου. Η ρύθμιση παραμέτρων για τα δύο παράθυρα και μια λίστα ζευγών με το όνομα πεδίου εξόδου και τα πεδία που θα ληφθούν από την ένωση των δύο ροών εισόδου.

- **Είσοδος:** δύο ροές εισόδου χωρίς απαιτήσεις στα σχήματα τους.

- **Εξοδος:** μία ροή εξόδου με όλα τα πεδία που καθορίζονται στις διαμορφώσεις.

Παράδειγμα: Στη περίπτωση του αλγορίθμου που επιστρέφει τα αποτελέσματα από τη συναισθηματική φόρτιση που πιστεύεται ότι έχει ένας χρήστης social media γράφοντας το tweet που δημοσιοποιεί στα κοινωνικά δίκτυα, ο τελεστής προσθέτει ανά διάστημα που ορίζεται εκ των προτέρων πόσα αθροιστικά θετικά είτε αρνητικά μηνύματα υπάρχουν σε αυτό το διάστημα. Τέτοια παραδείγματα έχουν ιδιαίτερο ενδιαφέρον σε περιπτώσεις που υπάρχει αρκετά μεγάλη μεταβλητότητα στα αποτελέσματα λόγω κάποιων γεγονότων είτε κάποιας δυναμικής. Η περίπτωση ενός debate είναι μια τέτοια περίπτωση όπου μπορούν να βγουν χρήσιμα συμπεράσματα άμεσα για αυτά που λέει ο κάθε υποψήφιος.



Εικόνα 19 Παράδειγμα του τελεστή Join

4.8 Τελεστές διαχείρισης συστημάτων αποθήκευσης

Οι τελεστές διαχείρισης είναι σε θέση να εκτελέσουν ερωτήματα SQL σε εξωτερικούς χώρους αποθήκευσης δεδομένων. Το CEP μπορεί να υποστηρίξει οποιοδήποτε δομή ή βάση δεδομένων αρκεί να υπάρχει το αντίστοιχο πρόγραμμα οδήγησης, όπως για παράδειγμα ο ODBC driver.

4.8.1 Τελεστής Insert

- **Λογική:** ο τελεστής Insert εκτελεί ένα ερώτημα εισαγωγής SQL στον πίνακα απομακρυσμένου χώρου αποθήκευσης δεδομένων, χρησιμοποιώντας τα πεδία που ορίζονται στη ροή εισόδου

- **Διαμόρφωση:** το όνομα και το σχήμα του πίνακα που υπάρχει στο αποθετήριο δεδομένων προορισμού, μια λίστα με πεδία της ροής εισόδου που θα αποθηκευτούν και προαιρετικά μια συνθήκη WHERE στα πεδία της ροής εισόδου.

- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** δεν υπάρχει ροή εξόδου.

4.8.2 Τελεστής Update

- **Λογική:** ο τελεστής update εκτελεί ένα ερώτημα ενημέρωσης SQL στον πίνακα απομακρυσμένου χώρου αποθήκευσης δεδομένων, χρησιμοποιώντας τα πεδία που ορίζονται στη ροή εισόδου

- **Διαμόρφωση:** όνομα και σχήμα του πίνακα που βρίσκονται στο αποθετήριο δεδομένων προορισμού, καθώς και μια λίστα με τα πεδία της ροής εισόδου προς ενημέρωση και μια συνθήκη WHERE στα πεδία τόσο της ροής εισόδου όσο και του πίνακα προορισμού.

- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** δεν υπάρχει ροή εξόδου.

4.8.3 Τελεστής Delete

- **Λογική:** Ο τελεστής delete εκτελεί ένα ερώτημα διαγραφής SQL στον πίνακα του απομακρυσμένου χώρου αποθήκευσης δεδομένων, χρησιμοποιώντας τα πεδία που ορίζονται στη ροή εισόδου

- **Διαμόρφωση:** όνομα και σχήμα του πίνακα στο αποθετήριο δεδομένων προορισμού και μια συνθήκη WHERE όπου στα πεδία τόσο της ροής εισόδου όσο και του πίνακα προορισμού εφαρμόζεται για τον προσδιορισμό των γραμμών που θα διαγραφούν.

- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** μία ροή εξόδου με ένα σχήμα που είναι αντίστοιχο με το σχήμα του πίνακα του αποθετηρίου.

4.8.4 Τελεστής Select

- **Λογική:** Ο τελεστής select εκτελεί ένα ερώτημα SQL με παραμέτρους τα πεδία της ροής εισόδου στον πίνακα της βάσης προορισμού.

- **Ρύθμιση παραμέτρων:** όνομα και σχήμα του πίνακα της βάσης προορισμού καθώς και ένα σύνολο ονομάτων πεδίων για ανάγνωση από τον πίνακα και προαιρετικά μια συνθήκη WHERE όπου εφαρμόζεται στα πεδία τόσο της ροής εισόδου όσο και του πίνακα προορισμού για τον προσδιορισμό των γραμμών που θα διαβαστούν.

- **Είσοδος:** μία ροή εισόδου.

- **Έξοδος:** μία ροή εξόδου με ένα σχήμα που είναι αντίστοιχο με το σχήμα του συνόλου αποτελεσμάτων επιλογής.

4.9 Τελεστής προσαρμοσμένοι από τους χρήστες

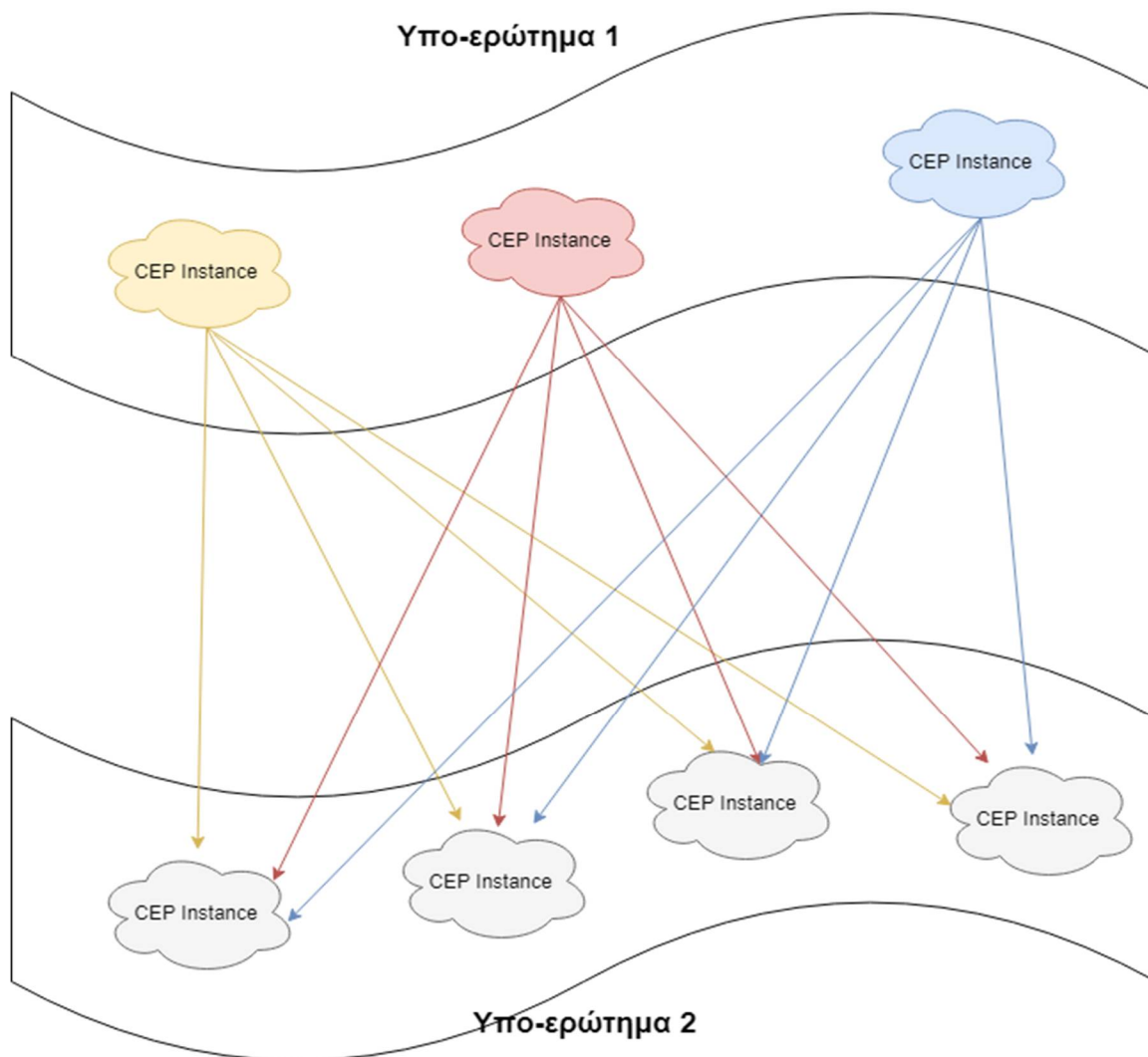
Οι εφαρμογές επιτρέπεται να καθορίζουν και να ενσωματώνουν τελεστές που ορίζονται από το χρήστη στο CEP. Για να διευκολυνθεί η ανάπτυξη αυτών των φορέων, το CEP προσφέρει στις εφαρμογές ένα σύνολο αφηρημένων τελεστών που υλοποιούν βασικές λειτουργίες, όπως η αλληλεπίδραση με συρόμενα παράθυρα και με ροές εισόδου και εξόδου. Η εφαρμογή πρέπει να παρέχει μόνο την επιχειρηματική λογική που απαιτείται για την επεξεργασία των μεμονωμένων πλειάδων που λαμβάνει ο τελεστής.

4.10 Στρατηγικές Παραλληλοποίησης

Με το CEP, η ανάπτυξη ερωτημάτων είναι πλήρως παραμετροποιήσιμη. Αυτό συμβαίνει γιατί, το CEP επιτρέπει την εκτέλεση πολλών ερωτημάτων στο ίδιο νήμα, καθώς και την εκτέλεση κάθε τελεστή ερωτήματος σε διαφορετικούς κόμβους του συμπλέγματος. Όταν οι χρήστες πρέπει να εκτελέσουν σύνθετα ερωτήματα με συγκεκριμένες απαιτήσεις χρόνου απόκρισης, το ερώτημα μπορεί να διαιρεθεί σε δευτερεύοντα ερωτήματα με διαφορετικούς βαθμούς παραλληλισμού. Τα δευτερεύοντα ερωτήματα που απαιτούν περισσότερη επεξεργαστική ισχύ μπορούν να παραλληλοποιηθούν με υψηλότερο βαθμό από τα άλλα δευτερεύοντα ερωτήματα (με inter-operator παραλληλισμό είτε intra-operator παραλληλισμό). Η κύρια πρόκληση στο ερώτημα-παραλληλισμού είναι να εγγυηθεί ότι το αποτέλεσμα της παράλληλης εκτέλεσης είναι το ίδιο με ένα συγκεντρωτικό. Εστιάζοντας

σε ένα δευτερεύον ερώτημα που έγινε από έναν μόνο τελεστή, η προηγούμενη πρόκληση σημαίνει ότι η έξοδος ενός παράλληλου τελεστή πρέπει να είναι ίδια με αυτή του κεντρικού τελεστή. Από τη μία πλευρά, η λύση είναι απλή για τους τελεστές προσανατολισμένους σε συμβάντα και αποθετήρια δεδομένων, επειδή επεξεργάζονται κάθε πλειάδα ανεξάρτητα και οι εκροές τους δεν εξαρτώνται από προηγούμενες υλοποιήσεις. Από την άλλη πλευρά, οι τελεστές που χρησιμοποιούν τα παράθυρα απαιτούν να επεξεργάζονται όλες οι συγκεντρωτικές/συσχετισμένες μεταξύ τους από το ίδιο CEP instance. Για παράδειγμα, εάν ένας τελεστής συγκεντρωτικών αποτελεσμάτων που αντιστοιχεί στο σύνολο των μηνιαίων κινήσεων τραπεζικών λογαριασμών χρηστών είναι παραλληλοποιήσιμος σε τρία CEP instances, πρέπει να βεβαιώνεται ότι όλα τα τμήματα που ανήκουν στον ίδιο λογαριασμό χρήστη πρέπει να υποβληθούν σε επεξεργασία από το ίδιο CEP instance για να παράγει το σωστό αποτέλεσμα.

Για να διασφαλιστεί η ισοδυναμία μεταξύ κεντρικών και παράλληλων ερωτημάτων, πρέπει να δοθεί ιδιαίτερη προσοχή στις επικοινωνίες μεταξύ δευτερευόντων ερωτημάτων. Στην Εικόνα 20 που απεικονίζονται δύο δευτερεύοντα ερωτήματα, υπό-ερώτημα 1 και υπό-ερώτημα 2, με βαθμό παραλληλισμού τρία και τέσσερα αντίστοιχα. Εάν το υπό-ερώτημα 2 δεν περιέχει κανέναν τελεστή που ακολουθεί τη φιλοσοφία του παραθύρου, τα CEP instances στο υπό-ερώτημα 1 μπορεί αυθαίρετα να αποφασίσουν σε ποιο CEP instance του υπό-ερωτήματος 2 να στέλνουν τις εξόδους τους.



Εικόνα 20 Στρατηγική Παραλληλοποίησης

Εάν το υπό-ερώτημα 2 περιέχει έναν τελεστή με χρήση παραθύρου, τα CEP instances στο υπό-ερώτημα 1 πρέπει να στείλουν τη στοιβάδα εξόδου τους σύμφωνα με την ακόλουθη λογική:

Οι έξοδοι του υπό-ερωτήματος 1 αντιστοιχίζονται σε κουβάδες. Αυτή η ανάθεση βασίζεται στα πεδία της στοιβάδας. Δεδομένου ενός αριθμού «N» από διακριτούς κουβάδες, ο κουβάς «n» που αντιστοιχεί σε μια πλειάδα «t» υπολογίζεται με κατακερματισμό (hashing) ενός ή περισσοτέρων πεδίων της στοιβάδας «N». Όλες οι στοιβάδες που ανήκουν σε ένα δεδομένο κουβά αποστέλλονται στον ίδιο CEP instance του υπό-ερωτήματος 2.

Τα πεδία που θα χρησιμοποιηθούν στη συνάρτηση κατακερματισμού εξαρτώνται από τη σημασιολογία των τελεστών με χρήση του παραθύρου που ορίζονται στο υπό-ερώτημα 2 και η αντιστοίχιση μεταξύ των πακέτων και των επόμενων CEP instances που εξαρτώνται από τον αλγόριθμο εξισορρόπησης φόρτου που χρησιμοποιείται από το CEP.

- **Join:** αν το κατηγορηματικό σύνδεσμο έχει τουλάχιστον μία συνθήκη ισότητας θα το ονομάσουμε ως Equi-Join (EJ) αλλιώς θα το ονομάσουμε Cartesian Product (CP). Για τους τελεστές του τύπου EJ, η συνάρτηση hash υπολογίζεται σε όλα τα πεδία που χρησιμοποιούνται στις ισότητες συν τα προαιρετικά πεδία που θα μπορούσαν να εμφανιστούν στον όρο GROUP-BY. Για τελεστές του τύπου CP, η συνάρτηση κατακερματισμού υπολογίζεται σε όλα τα πεδία που χρησιμοποιούνται σε κατηγορηματικό συν τα προαιρετικά πεδία που θα μπορούσε να εμφανιστεί στον όρο GROUP-BY.

- **Aggregate:** τα πεδία που χρησιμοποιούνται στη λειτουργία κατακερματισμού είναι όλα τα πεδία που χρησιμοποιούνται στην παράμετρο GROUP-BY. Με αυτόν τον τρόπο, είναι βέβαιο ότι όλες οι πλειάδες που μοιράζονται τις ίδιες τιμές των

χαρακτηριστικών που καθορίζονται στην παράμετρο GROUP-BY επεξεργάζονται από το ίδιο CEP instance.

Το CEP διαθέτει μια προεπιλεγμένη στρατηγική παραλληλισμού που χρησιμοποιείται για τον διαχωρισμό ενός ερωτήματος σε δευτερεύοντα ερωτήματα, αυτό παρέχεται χωρίς να υπάρχει η ανάγκη ρυθμίσεων διαφόρων παραμέτρων από τη μεριά του χρήστη. Σύμφωνα με αυτήν τη στρατηγική, δημιουργείται ένα νέο δευτερεύον ερώτημα κάθε φορά που πληρούνται μία από τις ακόλουθες συνθήκες κατά τη διάρκεια του ερωτήματος:

- Είναι ένας τελεστής που βασίζεται στη χρήση παραθύρου.
- Είναι ένας τελεστής με περισσότερες από μία ροή εισόδου.
- Όλοι οι τελεστές που εφαρμόζονται πριν το πρώτο τελεστή που κάνει χρήση παραθύρου και λειτουργεί πάνω στο ίδιο δευτερεύον ερώτημα.

4.11 Ελαστικότητα

Ο ενορχηστρωτής του CEP και ο Instance Manager που απεικονίζονται στην Εικόνα 13 είναι τα στοιχεία που είναι αρμόδια για την παροχή ελαστικότητας στο CEP cluster. Συγκεκριμένα, ο Instance Manager παρακολουθεί τοπικά για τη συλλογή πληροφοριών σχετικά με τη χρήση πόρων και μαζεύει στατιστικά στοιχεία από τον ελεγχόμενο CEP instance. Αυτές οι πληροφορίες αποστέλλονται περιοδικά στον ενορχηστρωτή που συναθροίζει όλα τα ληφθέντα δεδομένα σε μια βάση ανά μηχανή. Ο

ενορχηστρωτής επεξεργάζεται τα δεδομένα που συλλέγονται και όταν δεν πληρούν ένα προκαθορισμένο βασικό δείκτη επιδόσεων, μπορεί να αποφασίσει είτε να ρυθμίσει ξανά τις παραμέτρους της προσθήκης/κατάργησης μηχανών σε ερωτήματα είτε να εξισορροπήσει το φορτίο μεταξύ αυτών των μηχανών.

Η επαναρύθμιση παραμέτρων μπορεί να είναι inter-query είτε intra-query. Στην πρώτη περίπτωση, ένα ή περισσότερα ερωτήματα που εκτελούνται στον υπολογιστή A μετακινούνται στον υπολογιστή B όταν ο ενορχηστρωτής ανιχνεύσει ότι τα ερωτήματα που εκτελούνται στον υπολογιστή A οδηγούν στο κορεσμό των πόρων αυτού του μηχανήματος, ενώ υπάρχει ένα μηχάνημα B όπου μπορεί να στείλει και να τρέξουν τα ερωτήματα επεξεργασίας με πολύ λιγότερο φόρτο. Η επαναρύθμιση παραμέτρων των intra-query είναι συγκεκριμένη για ένα παράλληλο καταναεμημένο ερώτημα. Όπως περιγράφεται στην αρχή του κεφαλαίου, οι ροές χωρίζονται σε κουβάδες και οι κουβάδες αντιστοιχίζονται σε τελεστές, η επαναρύθμιση σε επίπεδο intra-query σημαίνει ότι πρέπει να γίνει μεταφορά του κατόχου του κάδου ενός υπερφορτωμένου CEP instance σε ένα λιγότερο φορτωμένο CEP instance. Η επανα-διαμόρφωση μεταξύ ερωτημάτων είναι μια γενίκευση της επαναρύθμισης παραμέτρων ενός ερωτήματος. Στο ακόλουθο παράδειγμα, περιγράφεται ο μηχανισμός ελαστικότητας για τον τελευταίο μηχανισμό που επικεντρώνεται στις περιπτώσεις υποερωτημάτων που πραγματοποιούνται από έναν μόνο τελεστή.

Δεδομένης χρονικής στιγμής "Ts", η μεταφορά της κυριότητας ενός κάδου "b" από το A στο B σημαίνει ότι όλες οι πλειάδες που ανήκουν στο "b" των οποίων η χρονική

σήμανση είναι μικρότερη από "Ts" πρέπει να υποβληθούν σε επεξεργασία από το A, ενώ οι νεότερες πλειάδες πρέπει να υποβληθούν σε επεξεργασία από το B. Η επαναρύθμιση του τελεστή είναι σχετικά απλή διαδικασία, επειδή επεξεργάζεται μία πλειάδα τη στιγμή, έτσι ώστε όταν η πρώτη πλειάδα με χρονική σήμανση μεγαλύτερη από "Ts" πρέπει να επεξεργαστεί, αποστέλλεται στο B και ταυτόχρονα μπορεί να απορριφθεί από τον A. Οι τελεστές με χρήση παραθύρων είναι πιο δύσκολοι λόγω του συρόμενου παραθύρου. Στην πραγματικότητα, τα αποτελέσματα που παράγονται από μια πλειάδα με χρονική σήμανση μεγαλύτερη από το "Ts", ως εκ τούτου ανήκει στο B, θα μπορούσε να εξαρτηθούν από τα παράθυρα που είναι αποθηκευμένα στο συρόμενο παράθυρο που διατηρείται στον A.

Το πρωτόκολλο ελαστικότητας αποτελείται από τις ακόλουθες φάσεις:

1. Ταυτοποίηση των κάδων που πρέπει να μεταφερθούν από τον παλιό στον νέο ιδιοκτήτη.
2. Αναγνώριση της χρονικής σήμανσης "Ts" για να χρησιμοποιηθεί ως σημείο εκκίνησης της επαναδιαμόρφωσης.
3. Μεταφορά της κατάστασης του κάδου από τον παλιό ιδιοκτήτη στο νέο.

Ιδιαίτερη αναφορά και εστίαση πρέπει να δοθεί στο τρίτο, όπου "A" είναι ο παλιός ιδιοκτήτης, "B" ο νέος ιδιοκτήτης, "b" ο κουβάς που μεταφέρεται και "Ts" το σημείο εκκίνησης της επαναδιαμόρφωσης. Όλη η πλειάδα με χρονική σήμανση μεγαλύτερη από "Ts" απορρίπτεται από το "A". Μόλις απορρίψει την πρώτη πλειάδα, το A αρχίζει τη σειριοποίηση της κατάστασης σε σχέση με τον κουβά "b" και το στέλνει στο "B". Αφού οι

"Ts" πλειάδες έχουν σταλεί και στα δύο, A και B, μέχρι να ειδοποιήσει ο A τον Instance Manager του για την ολοκλήρωση της κρατικής μεταφοράς. Όταν ο B ανακατασκευάσει την κατάσταση λήψης, τότε μόνο μπορεί να επεξεργαστεί τα εισερχόμενα, ξεκινώντας από αυτές τις πλειάδες που αποθηκεύονται κατά τη διάρκεια της μεταφοράς κατάστασης.

4.12 Διασυνδεδεμένο σύστημα ανάλυσης πολλών δεδομένων

Με την ευρεία διάδοση των κοινωνικών δικτύων, των υποδομών με αισθητήρες και των έξυπνων συσκευών -είτε αυτά είναι κινητά είτε συσκευές οικιακής χρήσης- τα δεδομένα που συλλέγουν είναι συνεχώς όλο και περισσότερα. Αυτός ο παράγοντας δημιουργεί νέες προκλήσεις που πρέπει να αντιμετωπιστούν ώστε να μπορεί σε πραγματικό χρόνο να γίνει η επεξεργασία των δεδομένων εάν αυτό κρίνεται αναγκαίο. Ταυτόχρονα η δυνατότητα ταχείας αντίδρασης στις μεταβαλλόμενες τάσεις και κατά συνέπεια στα δεδομένα είναι αυτονόητη μιας και τις περισσότερες φορές τα συστήματα που καλούνται να πάρουν αποφάσεις διαχειρίζονται μεγάλα πακέτα μετοχών και χρημάτων. Τέτοιο παράδειγμα είναι τα συστήματα αναγνώρισης συναισθήματος καθώς και η πρόβλεψη των λεγομένων και των αποτελεσμάτων που μπορεί να έχουν αυτά στις μετοχές του χρηματιστηρίου από κολοσσούς όπως η Thomson Reuters και η Bloomberg.

Ερευνητικές προσεγγίσεις για την πρόβλεψη τέτοιων τάσεων, ακολουθούν μεθόδους που βασίζονται στο AutoRegressive Integrated Moving Average (ARIMA) [109]. Άλλες προσεγγίσεις ακολουθούν συναθροιστηκές τεχνικές συνδυάζοντας artificial

neural network, decision tree και k-nearest neighbor [110]. Ταυτόχρονα υπάρχουν προσεγγίσεις [111] που με χρήση SVM καταφέρνουν να προσεγγίσουν τη κίνηση του δείκτη ΝΙΚΚΕΙ 225 με ικανοποιητική ακρίβεια. Τέλος πρόσφατες μελέτες [112] έχουν καταφέρει με επαναλαμβανόμενο νευρωνικό δίκτυο μακράς διάρκειας μνήμης (LSTM) να έχουν πολύ μεγάλη ακρίβεια στα αποτελέσματά τους.

Για την ανάλυση και αξιολόγηση του τρόπου αλλά και του μέσου που μπορούν μονολιθικές εφαρμογές να αντικατασταθούν-τροποποιηθούν με τέτοιο τρόπο ώστε να αντιμετωπίσουν προβλήματα που γεννιούνται από πλημμύρα δεδομένων, παρουσιάζεται η προσέγγιση ανάλυσης συναισθήματος που έχει δομηθεί ακολουθώντας το σχεδιασμό που θα επιτρέψει να υπερκεραστούν αυτά. Συγκεκριμένα η προσέγγιση Giannakopoulos [113] που βασίστηκε η υλοποίηση από τον Tzannetos [114] για την ανάλυση, επεξεργασία και συναισθηματική αξιολόγηση σύντομων μηνυμάτων με χρήση n-gram είναι αυτή που αναδομήθηκε με τέτοιο τρόπο ώστε να ακολουθεί τις αρχές που θα της επιτρέψουν να ανταπεξέλθει σε συστήματα διαχείρισης πολλών δεδομένων.

Η μέθοδος των n-gram γράφων δημιουργήθηκε για την αξιολόγηση μοντέλων που εξάγουν αυτόματα περίληψη κειμένου (text summarization) [115] [116] [117] [118]. Η διαδικασία περιλαμβάνει αρχικά τη δημιουργία μίας περίληψης. Η περίληψη που δημιουργείται αποθηκεύεται σε μια διάταξη n-gram γράφου όπου κατά την δημιουργία αποθηκεύονται στον γράφο γειτονικά n-grams χαρακτήρων από τα κείμενα. Ο γράφος ελαχιστοποιεί τη πληροφορία των αρχικών κειμένων διότι αποθηκεύει την εκάστοτε ακολουθία χαρακτήρων που εμφανίζεται με τον ίδιο τρόπο σε ένα ή περισσότερα κείμενα.

Με αυτό το τρόπο ο γράφος μπορεί να οριστεί ως ένα σύνολο κοινών χαρακτηριστικών που είχαν τα κείμενα που χρησιμοποιήθηκαν στην αρχή βασισμένο σε αυτό το «μοντέλο κειμένου».

Για να δημιουργηθεί αυτό το μοντέλο κειμένου θα πρέπει κάθε κείμενο που υπάρχει να αναπαρασταθεί με τη μορφή του n-gram γράφου. Στη περίπτωση που έχει επιλεγθεί σαν μέγεθος n-gram το τρία τότε διασπάται το κάθε κριτικό-κείμενο βάση αυτού του ορίου. Στη περίπτωση που υπάρχει το κείμενο «εξαιρετικός υποψήφιος» εξάγονται από αυτή τη πρόταση όλα τα 3-grams που παράγονται και αποτελούν την ανάλυση της πρότασης.

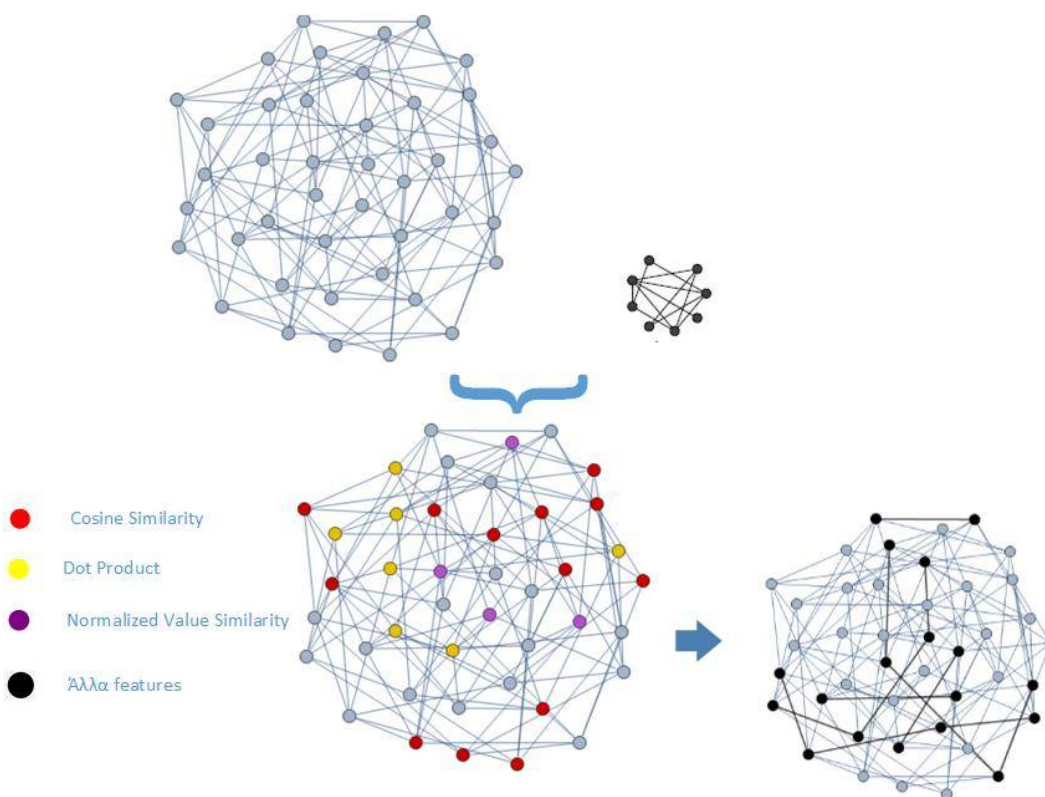
Για κάθε μοναδικό 3-gram δημιουργούμε έναν κόμβο στον αρχικό γράφο. Η χρήση γράφου έχει κύριο σκοπό την αποθήκευση της γειτνίασης των 3-grams, η οποία εμπεριέχει πληροφορία για το πλήθος παρουσίας των χαρακτήρων στο αρχικό κείμενο. Ταυτόχρονα θα πρέπει να οριστεί και το μήκος του παραθύρου που αποτελεί το πλήθος των γειτονικών στο 3-gram χαρακτήρων. Με αυτή τη λογική σε περίπτωση που οποιοδήποτε από τα στοιχεία που αποτελούν το 3-gram χρησιμοποιείται σαν μέρος ενός άλλου 3-gram τότε θεωρούμε ότι αυτά γειτνιάζουν και σαν αποτέλεσμα τοποθετείται μια ακμή ανάμεσα σε αυτά τα δύο. Στη περίπτωση που μεταξύ αυτών των δυο κορυφών υπάρχει ακμή από προηγούμενη περίπτωση, τότε προστίθεται στο συντελεστή εμφάνισης της μια ακόμα εμφάνιση. Αυτός ο συντελεστής ονομάζεται βάρος.

Ακολουθώντας αυτή τη λογική και έχοντας από την αρχή ταξινομήσει τα κείμενα ανάλογα με τη θεματολογία τους μπορούν να δημιουργηθούν δύο είτε τρία μοντέλα (εάν

συμπεριλάβουμε και το ουδέτερο) γράφων. Ένα θετικής πολικότητας (positive graph), και αντίστοιχα έναν γράφο αρνητικής πολικότητας (negative graph).

Το επόμενο βήμα είναι να χρησιμοποιηθούν αυτοί οι γράφοι, για εξαγωγή χαρακτηριστικών για την μηχανική μάθηση. Αυτοί οι γράφοι που βασίζονται σε διαφορετικά κείμενα από αυτά που έχουν χρησιμοποιηθεί για τη δημιουργία του μοντέλου γράφου αντιπροσωπεύεται από ένα σύνολο χαρακτηριστικών (features) το οποίο αποτελεί ένα instance προς ταξινόμηση. Το σύνολο αυτών των χαρακτηριστικών εξάγεται από το κείμενο και σαν κύριο στόχο έχουν να μπορούν να χρησιμοποιηθούν για να δώσουν μια απάντηση για τη πολικότητά -θετική είτε αρνητική- του κειμένου.

Ο γράφος του κειμένου συγκρίνεται με κάθε γράφο πολικότητας, ώστε να εξάγουμε κάποιες τιμές ομοιότητας με τον καθένα από αυτούς (Εικόνα 21).



Εικόνα 21 Σύγκριση και ανάλυση γράφων με βάση δείκτες ομοιότητας

Πέρα όμως από τους δείκτες ομοιότητας που αναφέρονται για την σύγκριση γράφων κατά την αξιολόγηση περιλήψεων η δυνατότητα που προσφέρει η μετατροπή του αλγορίθμου ακολουθώντας λογική διάσπαση των διαδικασιών είναι ότι μπορεί να εισάγει και άλλους δείκτες. Αυτό είναι και το σημείο που στο κομμάτι του αλγορίθμου έχουν γίνει οι αλγοριθμικές προσθήκες. Αυτή η λογική διάσπαση του κώδικα για χάριν της σπονδυλωτής αρχιτεκτονικής και της microservice προσέγγισης δίνει το πλεονέκτημα ότι

μπορεί ευκολότερα να τροποποιηθεί κομμάτι του αλγορίθμου ακόμα και εάν δεν είναι γνωστό το υπόλοιπο μέρος αυτού.

Η προσέγγιση όμως ξεπερνά τα στενά πλαίσια του αλγορίθμου και αντιμετωπίζει το πρόβλημα συστημικά επιλέγοντας το κατάλληλο σημείο για την κάθε ενέργεια έχοντας σα γνώμονα την απόδοση του τελικού συστήματος και την ευελιξία να κλιμακωθεί. Στο [119] αναλύονται δύο συστήματα που καθοδηγούν τον τρόπο σκέψης μας, το πρώτο σύστημα είναι γρήγορο, διαισθητικό και συναισθηματικό, ενώ το δεύτερο είναι πιο αργό, συνετό και έλλογο. Γίνεται λοιπόν αντιληπτό ότι η προσπάθεια ανάλυσης και επεξεργασίας ενός κειμένου (είτε μιας πρότασης) θα πρέπει να συμπεριλάβει σαν παράγοντα το σύστημα που οδήγησε τον άνθρωπο στην συγγραφή αυτού. Αυτό το στάδιο προϋποθέτει ανάλυση πλαισίου (context analysis) όπου μέσω της πολυδιάστατης κατηγοριοποίησης των δεδομένων δύναται να βελτιωθεί και να ομογενοποιηθεί η εξαγωγή συναισθημάτων και η ανάλυση του περιεχομένου των κειμένων. Συνεπώς η δημιουργία ενός πολυεπίπεδου αλγορίθμου προϋποθέτει τη κατάλληλη προεπεξεργασία της πληροφορίας και εξαγωγή heuristics (από τα metadata) για τα κείμενα ώστε να μπορέσει να αποδώσει στις δοθείσες συνθήκες βέλτιστα. Η ανάλυση πλαισίου αποτελεί τη προσέγγιση που πρόκειται να ακολουθηθεί για την προεπεξεργασία του περιεχομένου.

Σε ένα γενικότερο πλαίσιο με την προσέγγιση της ανάλυσης πλαισίου το σύστημα πρέπει να επεξεργάζεται κείμενα χρησιμοποιώντας επικουρικά τεχνικές και λεξιλόγια που αποδίδουν τα βέλτιστα για συγκεκριμένους τύπους κειμένων (blog, άρθρο, ετυμολογία, επιστολές), παρέχοντας αυτά τα χαρακτηριστικά σαν heuristics. Η ανάλυση πλαισίου για

την προεπεξεργασία των κειμένων αποτελεί τη σημαντικότερη αλλά όχι τη μοναδική χρήση για το σύστημα. Επιμέρους αναλύσεις πλαισίου μπορούν να υποβοηθήσουν στην εύρεση θεματικής ενότητας μιας πρότασης. Χαρακτηριστικό παράδειγμα είναι στην ανάλυση συναισθήματος που μέσω της εισαγωγής περισσότερων ή πιο αντιπροσωπευτικών features μπορούν να περιγράψουν καλύτερα, είτε να προσφέρουν στον αλγόριθμο τη γνώση για τις ιδιαιτερότητες που έχει. Με αυτό το τρόπο μπορεί να έχει ο αλγόριθμος την κατάλληλη γνώση για τα σωστά features που πρέπει να χρησιμοποιήσει είτε αυτά που πρέπει να εισάγει ακόμα πέρα από τα βασικά που χρησιμοποιεί.

Η ανάλυση πλαισίου λειτουργεί σε δυο στάδια ένα προπαρασκευαστικό και ένα κατά την λειτουργία του αλγορίθμου. Σε καθένα από αυτά τα στάδια το σύστημα αυτό επικοινωνεί με την βάση δεδομένων του συστήματος. Η επικοινωνία αυτή είναι αμφίδρομη και έχει ως κύριο στόχο την εκπαίδευση του συστήματος για να παραχθούν ακριβέστερα αποτελέσματα όσο αυξάνονται τα δεδομένα καθώς επίσης και να “αναθεωρήσει” εάν κάποια δεδομένα είναι αντιπροσωπευτικά για να βρίσκονται στο training set.

Η αποδοτικότητα του συστήματος ανάλυσης πλαισίου και κατ’ επέκταση του αλγορίθμου βασίζονται στην ποιότητα και την ποσότητα των μοτίβων που έχουν παραχθεί στη βάση δεδομένων όσο και στο φιλτράρισμα των δεδομένων που επιλέγονται να χρησιμοποιηθούν. Ένας μεγάλος αριθμός μοτίβων έχει ως αποτέλεσμα το σύστημα ανάλυσης πλαισίου να είναι πολύ λεπτομερειακό στο χαρακτηρισμό της πληροφορίας. Παρόλα αυτά όσο μεγάλος και να είναι ο αριθμός των μοτίβων, η ποιότητα τους είναι ο σημαντικότερος παράγοντας διότι ένα λανθασμένο μοτίβο μπορεί να οδηγήσει σε πολύ

χειρότερα αποτελέσματα. Η συνεχής τροφοδοσία με κείμενα στη βάση δεδομένων καθώς και η χρήση deep learning τεχνικών εξασφαλίζει έλεγχο των μοτίβων τόσο κατά τη δημιουργία τους όσο και κατά την χρήση.

Στην Εικόνα 22 παρουσιάζεται η γενική αρχιτεκτονική του συστήματος που στόχο έχει να αναλύσει και να διαχειριστεί δεδομένα από κοινωνικά δίκτυα και ροές δεδομένων καθώς και να παρέχει πληροφορίες για το θέμα που έχει επιλεγεί να παρακολουθεί.

Το σύστημα θα μπορούσε να διαχωριστεί από πέντε υποσυστήματα

(α) Υπο-Σύστημα διαχείρισης πολλών δεδομένων (Storage Engine & Transactional Manager)

(β) Υπο-Σύστημα συγκέντρωσης απαραίτητων δεδομένων και παρακολούθησης των κοινωνικών δικτύων (Social Media & News Feeds)

(γ) Υπο-Σύστημα που κατανέμει τις διαδικασίες και πραγματοποιεί αναλύσεις (Data Processing Engine)

(δ) Υπο-Σύστημα παροχής μεταδεδομένων σχετικά με το περιεχόμενο (Context Manager)

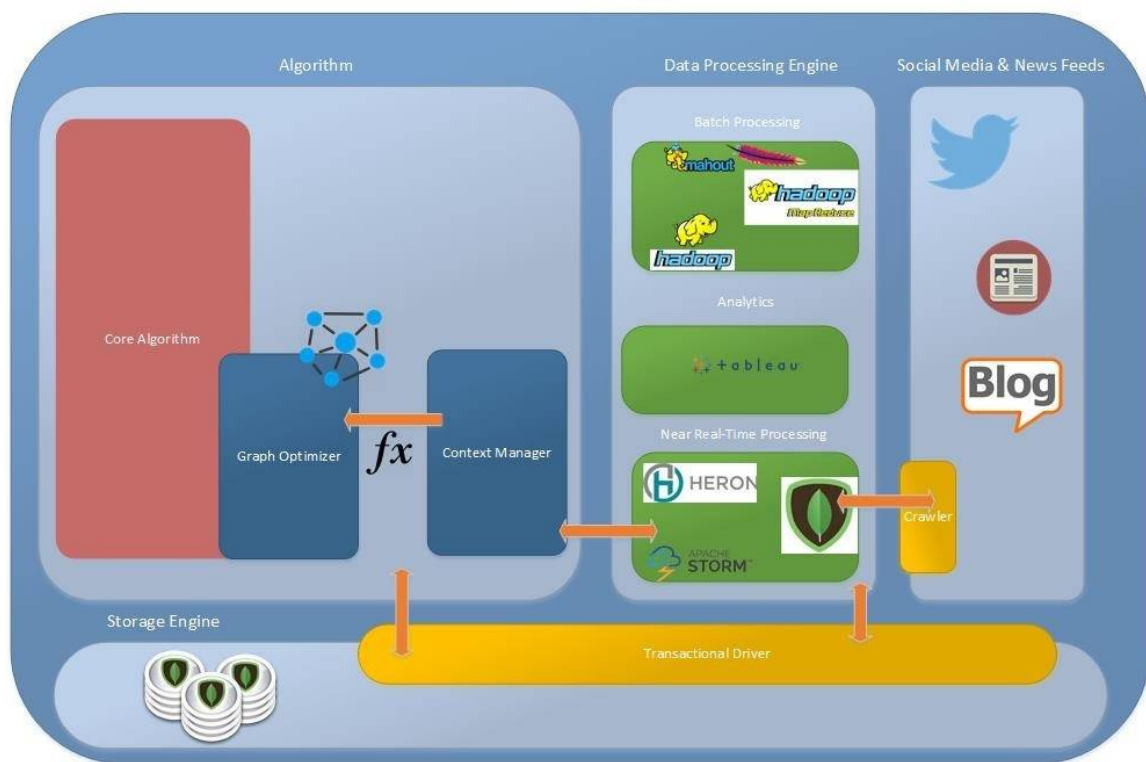
(ε) Υπο-Συστημα αλγορίθμου εξόρυξης πληροφοριών από πολυγλωσσικά δεδομένα (Algorithm)

Το κάθε κομμάτι της αρχιτεκτονικής παρουσιάζει μια ανεξαρτησία. Γνώμονας αυτής της επιλογής είναι η ευκολία διαχείρισης και του ελέγχου του κάθε υπο-συστήματος καθώς και η αύξηση της επαναχρησιμοποίησης πολλών κομματιών (reusability). Ταυτόχρονα ο συνδυασμός με τη προτυποποιημένη μορφή ανταλλαγής δεδομένων μπορεί να προσφέρει την ευελιξία να δοκιμαστούν και άλλες τεχνολογίες και υλοποιήσεις. Για παράδειγμα μπορεί για συγκεκριμένο σενάριο χρήσης η χρησιμοποίηση διαφορετικής βάσης δεδομένων να είναι βέλτιστη, είτε να επιλεγεί διαφορετική μηχανή για real time processing.

Το πρώτο στάδιο (α) είναι αυτό της συλλογής των δεδομένων από τις ροές δεδομένων. Σε αυτό το σύστημα με χρήση crawler μπορεί ανάλογα με τα κριτήρια επιλογής (επιλογή hashtags, επιλογή συγκεκριμένης λέξης είτε brand) να μαζεύει τα δεδομένα που δημιουργούνται [120]. Ταυτόχρονα λόγω του μεγέθους των δεδομένων σε περιπτώσεις υψηλής κίνησης του θέματος συζήτησης θα πρέπει αυτό να μπορεί να ανταπεξέλθει προσφέροντας scalability.

Το δεύτερο στάδιο (β) καθώς και το τρίτο (γ) αποτελούν τα κομμάτια αυτά της αρχιτεκτονικής που καλούνται να διαχειριστούν και να αποθηκεύσουν με βέλτιστο τρόπο τα δεδομένα. Ταυτόχρονα γνώμονας είναι η παραλληλοποίηση και το isolation των ερωτημάτων καθώς η βάση αντιμετωπίζει τόσο ερωτήματα γρήγορης εγγραφής (αποθήκευση tweets από το διαδίκτυο) καθώς όμως και ερωτήματα analytics (range queries από το Batch Processing). Συνεπώς χαρακτηρίζεται από την ικανότητα διαχείρισης μεγάλου όγκου δεδομένων (big-data, web-scale data), την ελεύθερη δομή του συστήματος

δεδομένων (semi structured format, format που ακολουθεί και το twitter –JSON), την εν γένη υποστήριξη της κατανεμημένης φιλοσοφίας (distributed), την υποστήριξη εύκολης επανάληψης-αντιγραφής (replication) των δεδομένων καθώς και την παροχή μηχανισμών που επιτρέπουν την αυτόματη επαναφορά του συστήματος σε περιπτώσεις που κάποιος κόμβος σταματάει να είναι διαθέσιμος. Η ευελιξία του ανοιχτού format καθώς και της ημιδομημένης μορφής του προσφέρει τη δυνατότητα συνεργασίας με πολλές πλατφόρμες διαχείρισης και ανάλυσης δεδομένων. Σε αυτό το στάδιο σημαντικό ρόλο παίζει και η δυνατότητα για τη καλύτερη διαχείριση και φιλτράρισμα των ερωτημάτων με τη χρήση του CEP από τη διεπαφή της διαχείρισης των δεδομένων προς τόσο στον αλγόριθμο όσο και σε υπηρεσίες ανάλυσης και μετα-επεξεργασίας για λόγους στατιστικούς είτε βελτιστοποίησης των παρεχόμενων υπηρεσιών.



Εικόνα 22 Γενική αρχιτεκτονική συστήματος διαχείρισης πολλών δεδομένων για ανάλυση μικρών κειμένων

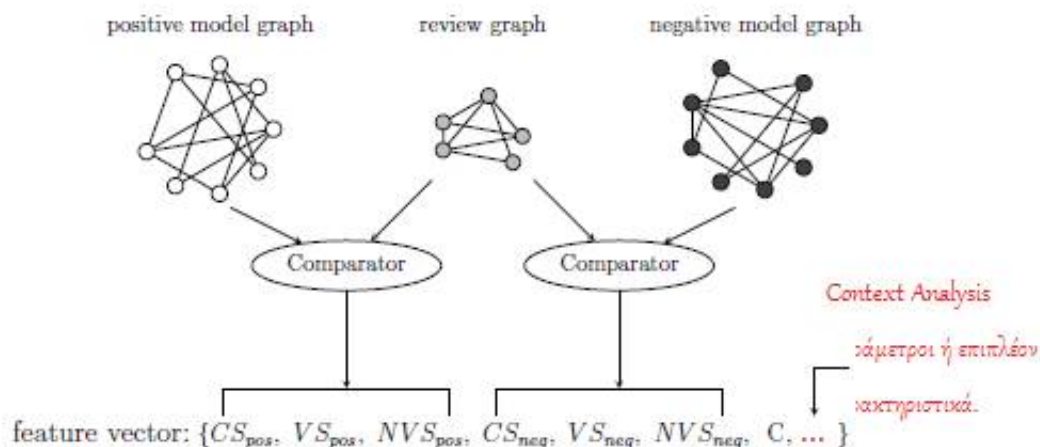
Τέλος τα δύο τελευταία στάδια είναι αυτά του context analysis (Context Manager) (δ) και του αλγορίθμου (ε). Σε αυτά τα στάδια η διαδικασία επεξεργασίας των δεδομένου γίνεται πράξη. Τα δεδομένα κατάλληλα μορφοποιημένα εισάγονται στο σύστημα επεξεργασίας ακολουθώντας την ταχύτητα που μπορεί να υποστηρίξει το σύστημα. Ο context manager αναλαμβάνει να κάνει έναν πρώτο έλεγχο των δεδομένων και να παρέχει τα απαραίτητα metadata στον αλγόριθμο. Τέτοια metadata είναι, η βέλτιστη συνάρτηση με

τα features καθώς και πληροφορίες για τη γλώσσα και το τύπο των δεδομένων της συγκεκριμένης περίπτωσης. Στη συνέχεια ο αλγόριθμος εκμεταλλεύεται αυτά τα μεταδεδομένα ώστε να επιλέξει τους βέλτιστους γράφους από τη βάση που μαζί με τη συνάρτηση των features που του έχει παραμετροποιήσει ο context manager παράγει το βέλτιστο αποτέλεσμα για το συγκεκριμένο request. Η επικοινωνία και οι κλήσεις του συστήματος για τη παραγωγή αποτελεσμάτων είναι ανεξάρτητες η μία από την άλλη με συνέπεια τα στάδια αυτά να μπορούν να παραλληλοποιηθούν με χαρακτηριστική ευκολία.

Επιστρέφοντας στο χαρακτηριστικό πλεονέκτημα της microservice προσέγγισης που είναι ότι μπορεί να διαδραματίσει ευκολότερα ένα πιο ολοκληρωμένο ρόλο στην ανάλυση εντός κειμένου μιας και μπορεί να συμπεριλάβει ακόμα περισσότερες τεχνικές και αναλύσεις που ήταν απαγορευτικές για ένα αλγόριθμο που υποχρεωτικά θα έτρεχε μονολιθικά και θα οδηγούσε σε φυσικούς περιορισμούς διαθέσιμων πόρων, είναι ότι επιτρέπει η προσέγγιση να ακολουθεί τη λογική του Ensemble learning. Με αυτό το τρόπο μπορεί να συνδυάζει τα πλεονεκτήματα που προσφέρουν α) τα Μπεϋζιανά Δίκτυα (Bayesian Networks), η Απλή Μπεϋζιανή Ταξινόμηση (Naive Bayes Classification), η Μεγιστοποίηση Εντροπίας (Maximum Entropy), και τα Νευρωνικά Δίκτυα (Neural Networks) στη παρουσία και συχνότητα όρων ή συστάδων όρων, που αναπαρίστανται από unigrams και n-grams αντίστοιχα μαζί με τη νοηματική πληροφορία βάσει της αναγνώρισης μερών του λόγου μέσα στο κείμενο και του αρνητικού δείκτη ο οποίος είναι δυνατό να τροποποιήσει προς το αντίθετο την συναισθηματική πολικότητα φράσεων και λέξεων, μαζί με β) τη Lexicon-Based προσέγγιση που χρησιμοποιεί λεξικά συναισθήματος

για να προσεγγίσει τη συναισθηματική «κατεύθυνση» της πρότασης ή της παραγράφου. Μια τέτοια ανάλυση μπορεί να εμπλουτιστεί από τη βάση του WordNet λεξικού [121] για την αναζήτηση των συνωνύμων είτε των αντώνυμων τους και γ) η ενσωμάτωση τεχνικών ανάλυσης κειμένου σε μέρη του λόγου (part of speech tagging, POS tagging) όπου οι κλάσεις στις οποίες ομαδοποιούνται οι λέξεις, είναι αυτές που μπορούν να αντληθούν αρκετές πληροφορίες σχετικά με τη λέξη αλλά και τις γειτονικές της [122] ακολουθώντας ένα συγκεκριμένο σύνολο tags όπως αυτό του Penn Treebank Project, [123]. Παράδειγμα είναι το [124] όπου καθίσταται προσπάθεια πρόβλεψης της πολικότητας επιθέτων, ενώ στο [125] η μελέτη αυτή έχει εμπλουτιστεί. Αντίστοιχα όπως και για την εκπαίδευση των n-grams έτσι και για την αξιολόγηση μοντέλων ανάθεσης μερών του λόγου (POS taggers) πρέπει να χρησιμοποιηθούν προκατασκευασμένα σώματα κειμένων (corpora) με προαναγνωρισμένα μέρη του λόγου. Για αυτό το βήμα corpus όπως το Brown corpus [126], Switchboard corpus, και του North American News Text Corpus (AP corpus). Σε [127] καθίσταται προσπάθεια πρόβλεψης της πολικότητας επιθέτων, ενώ σε [125] η μελέτη επεκτείνεται με τον επιπρόσθετο συνδυασμό ουσιαστικών και επιρρημάτων. [129] και WSJ corpus [130] εμπλούτισαν την ανάλυση και τη πιστότητα των κανόνων για την καλύτερη ακρίβεια του feature.

Συγκεντρωτικά όλες αυτές οι τεχνικές συνδυάστηκαν για να προσφέρουν πιο ακριβές αποτέλεσμα. Η σπονδυλωτή δομή της καινούρια δομής του κώδικα προσέφερε τη δυνατότητα να μπορούν να προστεθούν επιπλέον features στο διάλυμα σύγκρισης συναισθηματικής κατηγορίας.



Εικόνα 23 Σύγκριση και ανάλυση γράφων βάση της λίστας των χαρακτηριστικών

Στο διάνυσμα έξι στοιχείων (συν της πολικότητας της κριτικής) προστίθενται δύο ακόμη χαρακτηριστικά, έναν μετρητή «θετικών» κανόνων και έναν μετρητή «αρνητικών» κανόνων, που εξετάζουν την παρουσία κανόνων στο κείμενο (Εικόνα 23). Ένας κανόνας που έχει εμφανιστεί τις περισσότερες φορές σε θετική κριτική θα αυξάνει τον μετρητή «θετικών» κανόνων, ενώ αντίστοιχα ένας κανόνας που έχει εμφανιστεί τις περισσότερες φορές σε αρνητική κριτική θα αυξάνει τον μετρητή «αρνητικών» κανόνων. Οι τιμές των μετρητών δίδονται μαζί με τα χαρακτηριστικά ομοιότητας για την εκπαίδευση του μοντέλου μηχανικής μάθησης.

4.13 Συγκριτική μελέτη

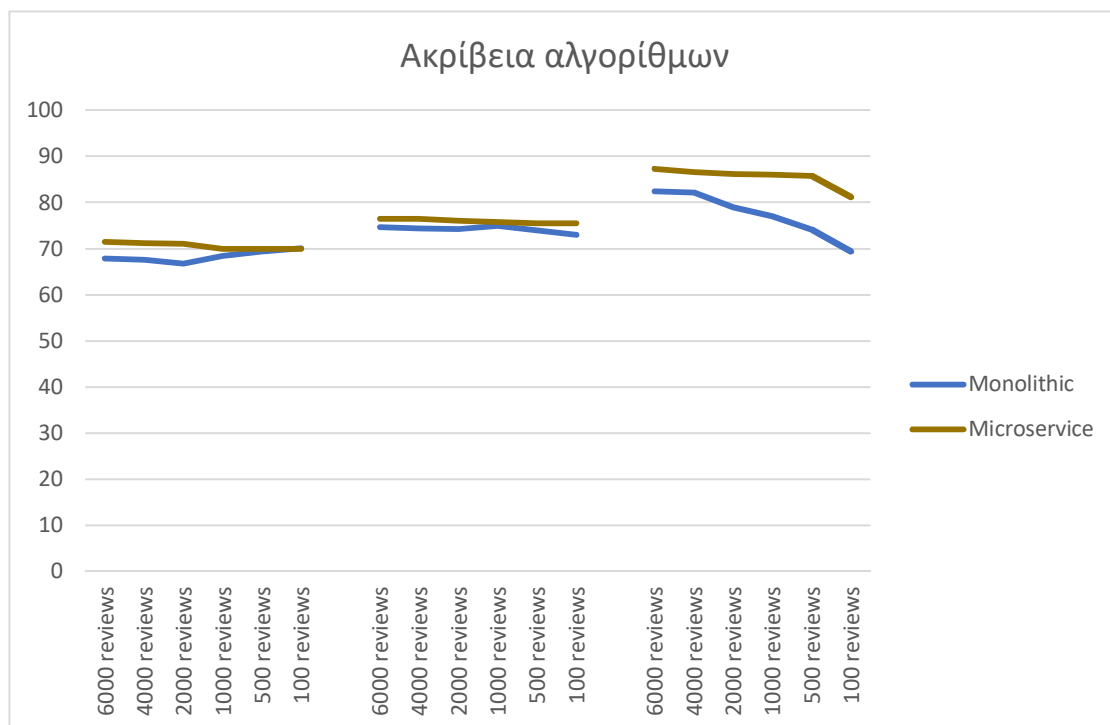
Οι όλο και περισσότερο μεγαλύτερες ανάγκες για ανάλυση δεδομένων από συστήματα που δεν αυξάνονται με τον ίδιο ρυθμό οδηγεί μοιραία είτε στο περιορισμό των δεδομένων που αναλύονται αφήνοντας ένα κομμάτι έξω από την ανάλυση είτε στο να απλουστεύεται αλγοριθμικά η επεξεργασία των δεδομένων για τα μειωθεί ο χρόνος απόκρισης. Με αυτό το τρόπο μειώνετε η ποιότητα της πληροφορίας και ταυτόχρονα η πιστότητα του αποτελέσματος. Στόχος, μιας και δεν είναι δεδομένο ότι όλα τα προγράμματα, αλγόριθμοι είναι εκ φύσεων ελαστικοί στη διαχείριση των διαθέσιμων πόρων, είναι να παρουσιαστεί το πρόβλημα συστημικά δομώντας έτσι τις απαιτήσεις ώστε να μπορεί να ανταπεξέλθει σε πολλές αν όχι όλες τις καταστάσεις. Η σύγκριση βέβαια ενός μονολιθικού αλγορίθμου με τη προτεινόμενη αλλαγή δεν έχει σαν στόχο να δείξει τις καλύτερες επιδόσεις σε αξιοπιστία αλλά να παρουσιάσει ότι επιπλέον χαρακτηριστικά και τεχνικές που ήταν αποτρεπτικές στη μονολιθική λύση μπορούν να εφαρμοστούν και να χρησιμοποιηθούν. Την ίδια στιγμή παρουσιάζεται πως μπορεί να ξεπεράσει τους περιορισμούς πόρων η νέα λύση να κλιμακωθεί ανάλογα με τους διαθέσιμους καταναεμημένους πόρους.

Στην Εικόνα 24 παρουσιάζονται τα αποτελέσματα σε σύγκριση του μονολιθικού αλγορίθμου με το τροποποιημένο αλγόριθμο. Οι δοκιμές βασίστηκαν στις κριτικές τόσο

ταινιών, χρησιμοποιώντας την ανοιχτή βάση που προσφέρει το `imdb` [131], όσο και στις κριτικές εστιατορίων που βασίζονται στα μηνύματα που γράφουν οι χρήστες στο `Yelp` [132]. Δοκιμάζοντας τις δύο περιπτώσεις σε δύο διαφορετικά πεδία μπορούμε να έχουμε καλύτερη εικόνα για τις επιδόσεις τους και στο πώς ανταπεξέρχονται σε διαφορετικούς θεματικούς τομείς. Στο γράφημα έχει βρεθεί ένας μέσος όρος απόδοσης από τις δύο βάσεις πληροφοριών χρησιμοποιώντας 20 διαφορετικά `training sets` για κάθε μια από τις περιπτώσεις. Τα αποτελέσματα που παρουσιάζονται δεν αλλάζουν τη σειρά ακρίβειας μιας και πάντα υπερέχει το `word-gram` ακολουθούμενο από το `4-gram` και τελευταίο βρίσκεται το `3-gram`. Όσο αναφορά τη σύγκριση των δύο περιπτώσεων παρατηρείται ότι πάντα η νέα λύση είναι καλύτερη από τη μονολιθική. Αυτό έχει λογική μιας και η νέα λύση περιλαμβάνει σαν μέρος της την ανάλυση που έχει γίνει από τη παλαιά.

Ένα ακόμα εύρημα είναι ότι η νέα λύση προσφέρει στο σύστημα με την ανάλυση βασισμένη στα `word-grams` αισθητά καλύτερα αποτελέσματα που μπορεί να δικαιολογηθεί από το γεγονός ότι τα νέα `features` που εισήχθησαν εκμεταλλεύονται τη δομή της πρότασης και των λέξεων. Μάλιστα παρατηρείται ότι η νέα προσέγγιση είναι πιο σταθερή στην ακρίβεια των αποτελεσμάτων από ένα αριθμό δειγμάτων και πάνω (500+). Τέλος όσο περισσότερο εκπαιδεύεται ο ταξινομητής, τόσο περισσότερο αυξάνει η διαφορά στην απόδοση της νέας προσέγγισης. Οι επιπλέον μετρητές, λειτουργούν επικουρικά προς την ανάλυση των γράφων, αυτό δηλαδή σημαίνει ότι δεν είναι αναγκαία η παρουσία κανόνων σε κάποιο κείμενο ώστε ο ταξινομητής να αποφανθεί για την κλάση

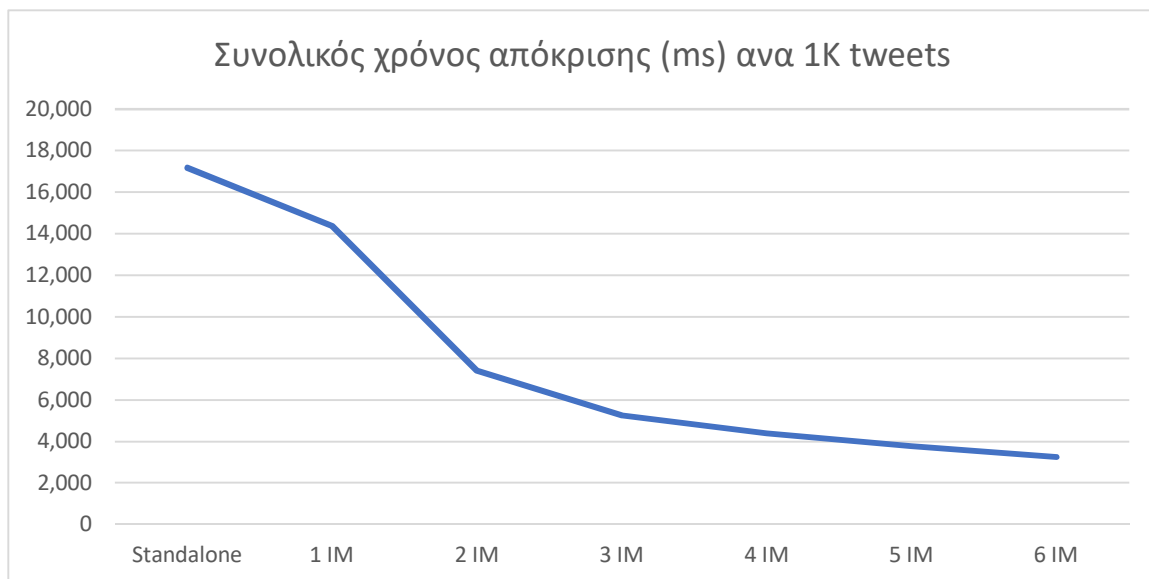
πολικότητας. Φαίνεται όμως από τα αποτελέσματα ότι στη περίπτωση που εισαχθούν επικουρούν στην αύξηση της ακρίβειάς του.



Εικόνα 24 Ακρίβεια αποτελεσμάτων ανάλογα με το μέγεθος των δεδομένων εκπαίδευσης

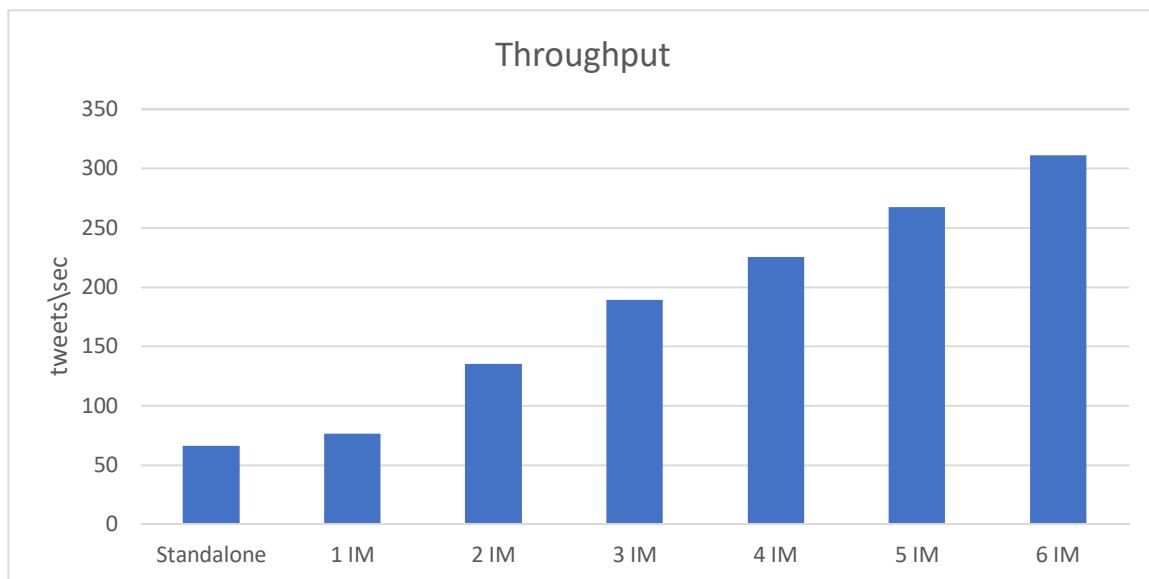
Αναφορικά με τις επιδόσεις σε περιβάλλον όπου πρέπει να επεξεργαστεί μεγάλο όγκο δεδομένων το πλεονέκτημα που μπορεί να προσφέρει η τροποποιημένη έκδοση είναι ότι έχει τη δυνατότητα να διασπάσει τις πράξεις και τις διεργασίες ώστε να εκτελούνται παράλληλα σε διαφορετικούς instance managers. Η δεύτερη δομική αλλαγή που έχει

προκύψει βασίζεται στο γεγονός ότι ο αλγόριθμος μπορεί να αφήσει τον CEP να πραγματοποιήσει τη διάσπαση των ερωτημάτων και να τις αναθέσει σε διαφορετικό σύστημα για εκτέλεση. Το πρόγραμμα οδήγησης κρύβει από τον αλγόριθμο/πρόγραμμα την πολυπλοκότητα του συμπλέγματος CEP και παρουσιάζει το σύμπλεγμα ως ένα μαύρο κουτί που οι εφαρμογές μπορούν να χρησιμοποιήσουν για να τρέξει παράλληλα. Οι εφαρμογές μπορούν να δημιουργήσουν και να αναπτύξουν συνεχόμενα ερωτήματα χρησιμοποιώντας το πρόγραμμα οδήγησης καθώς και να εγγραφούν στις ροές προέλευσης και να εγγραφούν σε ροές εξόδου αυτών των ερωτημάτων. Κατά την ανάπτυξη, το πρόγραμμα οδήγησης φροντίζει για τον διαχωρισμό ενός ερωτήματος σε δευτερεύοντα ερωτήματα και τα αναπτύσσει στο σύμπλεγμα CEP. Ορισμένα από αυτά τα δευτερεύοντα ερωτήματα μπορεί να είναι παραλληλισμένα αν χρειαστεί. Αυτό ακριβώς είναι και ένας από τους λόγους που στην Εικόνα 25 η αύξηση των επιδόσεων είναι ανάλογη με την αύξηση των IM του συστήματος. Η εγγραφή/εγγραφή σε ροές προέλευσης/εξόδου ενός ερωτήματος δεν είναι ασήμαντη όταν το συνεχές ερώτημα είναι παραλληλισμένο, επειδή η ίδια η ροή εξαπλώνεται μεταξύ όλων των instances του ερωτήματος/δευτερευόντων ερωτημάτων. Χρησιμοποιώντας το πρόγραμμα οδήγησης, οι αιτήσεις δεν αντιμετωπίζουν το πρόβλημα του να αναζητούν μια θέση ροής ούτε τη διαχείριση της για το παραλληλισμό της ροής, επειδή το πρόγραμμα οδήγησης κρύβει επίσης αυτή την πολυπλοκότητα και παρουσιάζει στην εφαρμογή μια απλή τοπική ροή για να τροφοδοτηθεί.



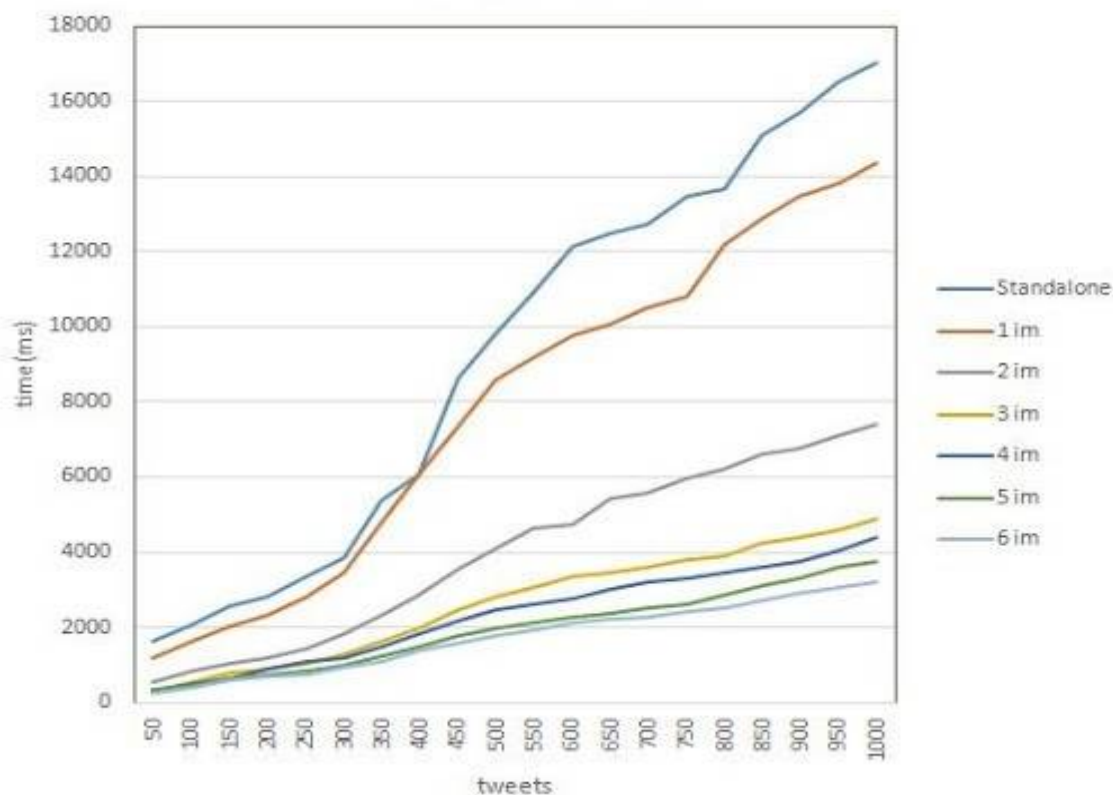
Εικόνα 25 Απόδοση συστήματος με χρήση πολλών IM

Ανάλογος του χρόνου απόκρισης είναι και ο συνολικός αριθμός tweets που μπορούν να επεξεργαστούν στη μονάδα του χρόνου. Στην Εικόνα 26 παρουσιάζεται πως καταφέρνει να ανταπεξέλθει και να ξεπεράσει την εν γέννη μη πλήρως παραλληλοποιημένη δομή του αλγορίθμου το σύστημα και να λειτουργήσει κοντά σε επιδόσεις με το απόλυτο. Η συγκεκριμένη μέτρηση έγινε με ένα cluster από AMD Opteron επεξεργαστών και επικοινωνία με infiniband μεταξύ των διαφορετικών clusters. Η μνήμη του συστήματος ήταν 256GB και η χρήση της λόγω της κατανομής δεν ξεπέρασε το 30% της χρησιμότητας. Κύριος λόγος για αυτό, ήταν ότι η τροποποιημένη έκδοση στο κομμάτι της εκτέλεσης να αποτελείται κυρίως από filter και map τελεστές.



Εικόνα 26 Συνολικός όγκος επεξεργαζόμενων tweets ανά μονάδα χρόνου

Στην Εικόνα 27 παρουσιάζεται πως κάθε διαφορετικό προφίλ εκτέλεσης ανταποκρίνεται σε όλο και μεγαλύτερες ροές δεδομένων. Τα κύρια ευρήματα είναι ότι στη περίπτωση της μη παραλληλοποιημένης νέας εκδοχής ο χρόνος εκτέλεσης σε σχέση με τον αριθμό των tweets είναι γραμμικός. Στον αντίποδα υπάρχει μια ελαφρά καλύτερη επίδοση στα συστήματα που χρησιμοποιούν παραπάνω από ένα Instance Manager. Συγκεκριμένα στην εκδοχή που υπάρχουν έξι instance managers η απόδοση παραμένει ελαφρώς καλύτερη από τη γραμμική (με 500 tweets ο χρόνος απόκρισης ήταν περίπου 2000 ms ενώ με το διπλάσιο αριθμό tweets περίπου 3800).



Εικόνα 27 Χρόνος επεξεργασίας tweets ανάλογα με τον αριθμό των IM

Επιστρέφοντας όμως στο πρωταρχικό πρόβλημα που δεν είναι άλλο από την επεξεργασία σε πραγματικό χρόνο, την ανίχνευση μοτίβων συμβάντων σε ροές δεδομένων το CEP μπορεί να κάνει την αντιστοίχιση με περιπτώσεις ακόμα και μεγάλης κλιμάκωσης. Σε αυτό συνεπικουρεί και η δυνατότητα που έχει και απορρίπτει αμέσως όλα τα δεδομένα που δεν σχετίζονται με το ερώτημα που έχει δοθεί.

4.14 Σύνοψη

Αυτό το κεφάλαιο περιγράφει την αρχιτεκτονική του μηχανισμού επεξεργασίας πολύπλοκων γεγονότων (CEP). Το CEP είναι μια μηχανή που έχει τη δυνατότητα να τρέχει παράλληλα και κατανεμημένα, που σημαίνει ότι είναι σε θέση να υποστηρίξει scale-up και scale-out. Ο σκοπός είναι η επίτευξη ενός ελαστικού CEP με αποτελεσματική δυνατότητα κλιμάκωσης και μη διασπαστική ελαστικότητα. Το σύστημα CEP έχει ένα σύνολο διαφορετικών στοιχείων, δηλαδή, CEP instances, πρόγραμμα διαχείρισης instances, ενορχηστρωτές, αξιόπιστα μητρώα καθώς και οδηγούς (driver που κρύβουν από τις εφαρμογές την πολυπλοκότητα του υποκείμενου συστήματος). Τα CEP instances είναι εργασία του μηχανισμού CEP που είναι σε θέση να φιλοξενούν ερωτήματα και δευτερεύοντα ερωτήματα και να επεξεργάζονται συμβάντα που στοχεύουν σε αυτά. Παρέχουν μια διασύνδεση για την ανάπτυξη ερωτημάτων και δευτερευόντων ερωτημάτων και επεξεργάζονται ροές από αυτές. Ο πιο δομημένος τρόπος για τη παρουσίαση αυτών των ερωτημάτων αλλά και τη δομή των δεδομένων είναι με τη χρήση JSON που βοηθούν στο καθορισμό των ερωτημάτων. Ο instance manager είναι το στοιχείο που χειρίζεται τη ρύθμιση παραμέτρων. Είναι αυτός δηλαδή που γνωρίζει τι ερωτήματα είναι και ποια είναι δευτερεύοντα ερωτήματα που αναπτύσσονται σε κάθε instance και ποιες ροές είναι συνδεδεμένες μεταξύ τους. Ο ενορχηστρωτής (orchestrator) είναι το καθολικό στοιχείο που ενώνει το σύμπλεγμα CEP. Όταν λάβει εντολή από το διαχειριστή για να αναπτύξει και να αναρρέσει την ανάπτυξη ερωτημάτων και επίσης παίρνει την απόφαση να ρυθμίσει ξανά το σύμπλεγμα για σκοπούς εξισορρόπησης φόρτου και ελαστικότητας. Το αξιόπιστο μητρώο είναι όπου η διαμόρφωση του CEP. Τέλος, το πρόγραμμα οδήγησης παρέχει ένα

API για εφαρμογές προγράμματος-πελάτη για να αλληλεπιδρούν με τα ερωτήματα, υποστηρίζοντας τόσο την ένεση και την απορρόφηση των περισσότερων και την ανάπτυξη των ερωτημάτων στο σύστημα.

Η δυνατότητα κλιμάκωσης του CEP επιτυγχάνεται με τρία είδη παραλληλισμό: παραλληλισμό μεταξύ ερωτημάτων (εκτελώντας διαφορετικά ερωτήματα σε διαφορετικούς κόμβους), εντός ερωτήματος παραλληλισμό μεταξύ τελεστών (εκτελώντας διαφορετικούς τελεστές ενός ερωτήματος σε διαφορετικούς κόμβους) και εσωτερικό ερώτημα ενδο-τελεστή παραλληλισμού (εκτελώντας έναν δεδομένο τελεστή ενός ερωτήματος σε πολλούς κόμβους).

Το CEP είναι επίσης ελαστικό. Το κύριο χαρακτηριστικό του είναι ότι η ελαστικότητα δεν είναι διασπαστική. Αυτό σημαίνει ότι ενώ το σύστημα αλλάζει δυναμικά, δεν σταματά ούτε διαταράσσει την τακτική επεξεργασία των συμβάντων, έτσι ώστε οι εφαρμογές-πελάτες δεν υποφέρουν από υποβάθμιση στο QoS.

ΚΕΦΑΛΑΙΟ 5

Μικροπηρεσίες

Η αρχιτεκτονική των μικροπηρεσιών, όπως ορίζεται στο NIST SP 800-180 [133], χρησιμοποιείται για τον σχεδιασμό και την ανάπτυξη εφαρμογών που αξιοποιούν ευέλικτες προσεγγίσεις ανάπτυξης λογισμικού. Η ασφάλεια των εφαρμογών πρέπει να εξετάζεται σε όλο τον κύκλο ζωής της ανάπτυξης του λογισμικού μιας και μόνο έτσι θα μπορεί να πιστοποιηθεί αλλά και να επιβεβαιώσει ότι έχει ακολουθήσει όλα τα απαραίτητα πρότυπα ασφάλειας και ποιότητας ελέγχου. Από την άλλη το NIST 800-160 [134], καθορίζει την ανάγκη για αξιόπιστα ασφαλή συστήματα που βασίζονται σε μια μεγάλη ποικιλία αναγκών των μικροπηρεσιών. Αυτές οι απαιτήσεις χρησιμεύουν για το καθορισμό των βέλτιστων πρακτικών για την εφαρμογή μιας ασφαλούς αρχιτεκτονικής μικροπηρεσιών που πρέπει να ακολουθεί ώστε να διαβεβαιώνει το τελικό χρήστη για τη πιστότητα και την ασφάλεια των αποτελεσμάτων των μεθόδων και των διεργασιών της.

5.1 Εισαγωγή

Μια αρχιτεκτονική σχεδίασης μικρουπηρεσιών είναι μια χαρακτηριστική μέθοδος για την ανάπτυξη ενός ενιαίου συστήματος λογισμικού σαν μια σουίτα μικρών υπηρεσιών. Η δημοσίευση του NIST SP800-180, ορίζει τις μικρουπηρεσίες ως «ένα βασικό στοιχείο που προκύπτει από την αρχιτεκτονική αποσύνθεση των συστατικών της εφαρμογής σε χαλαρά συζευγμένα μοτίβα που αποτελούνται από αυτόνομες υπηρεσίες που επικοινωνούν μεταξύ τους χρησιμοποιώντας ένα τυπικό πρωτόκολλο επικοινωνίας και ένα σύνολο σαφώς καθορισμένων API, ανεξάρτητα από οποιονδήποτε προμηθευτή, προϊόν ή τεχνολογία". Αυτή η αρχιτεκτονική αποτελείται από ένα ελάχιστο σύνολο υπηρεσιών, η οποία παρέχει ευελιξία, αλλά ταυτόχρονα εισάγει προκλήσεις, καθώς οι υπηρεσίες μπορούν να εγγραφούν σε διαφορετικές γλώσσες προγραμματισμού και να χρησιμοποιούν διαφορετικές τεχνολογίες αποθήκευσης δεδομένων.

5.2 Αρχιτεκτονική Προσανατολισμένη στην Υπηρεσία (SOA)

Η έννοια της SOA συνεχίζει να καθορίζει τις υπηρεσίες που παράγονται είτε σε εταιρικό είτε σε επιχειρησιακό περιβάλλον. Στο [135] ορίζεται ως ένα αρχιτεκτονικό μοτίβο που "μπορεί να πραγματοποιηθεί μέσω οποιασδήποτε κατάλληλης τεχνολογικής πλατφόρμας". Η βάση του προσανατολισμού της υπηρεσίας δεν βρίσκεται στην τεχνολογική νομολογία, αλλά θεωρητικά στον διαχωρισμό των απαιτήσεων, που σημαίνει ότι τα μεγάλα προβλήματα χωρίζονται σε μικρότερα, διακριτά συναφή μέρη, με τη μορφή υπηρεσιών [136]. Το πρότυπο του SOA στη συνέχεια βοηθά στο σχεδιασμό, την ανάπτυξη και τη διαχείριση αυτών των υπηρεσιών στο πληροφοριακό περιβάλλον. Ως εκ τούτου,

όπως ορίζεται στο [135]: «το SOA αντιπροσωπεύει μια ανοικτή, ευέλικτη, επεκτάσιμη, ομόσπονδη, συνθέσιμη αρχιτεκτονική που αποτελείται από αυτόνομα, με δυνατότητα QoS, ετερογενής προμηθευτές υπηρεσιών, διαλειτουργικό, και με δυνητικά επαναχρησιμοποιήσιμες υπηρεσίες».

Για την αυστηρή δομή και ανάλυση θα πρέπει να οριστούν όλα τα σκέλη του ορισμού, οπότε ως Υπηρεσία μπορεί να θεωρηθεί: ένα αυτόνομο κομμάτι της επιχειρηματικής λειτουργικότητας, με σαφή σκοπό [137]. Οι υπηρεσίες είναι μοντελοποιημένες από μια συγκεκριμένη επιχειρηματική διαδικασία. Όσο αναφορά το προσανατολισμό σε μια υπηρεσιοστρεφή λογική, σύμφωνα με το [138] είναι μια συλλογή από επιχειρησιακή λογική που είναι ενσωματωμένη σε ένα στοιχείο (component) του λογισμικού. Αυτό το στοιχείο παρέχεται ως υπηρεσία στους καταναλωτές υπηρεσιών, όπου ο καταναλωτής της υπηρεσίας μπορεί επίσης να είναι κάλλιστα και μια άλλη υπηρεσία. Ταυτόχρονα αυτά τα ανεξάρτητα στοιχεία, όταν είναι συνδεδεμένα, μπορούν να παρέχουν την πλήρη υποστήριξη για μια συγκεκριμένη επιχειρηματική διαδικασία κάτω από ένα μηχανισμό διακυβέρνησης και ελέγχου τους [139]. Τέλος ως εταιρική αρχιτεκτονική, ακολουθώντας τον ορισμό του [140] έχει να κάνει με την αρχιτεκτονική μέθοδο που μπορεί να χρησιμοποιηθεί για να μεταφράσει την επιχειρηματική λογική και στρατηγική για να καθοδηγήσει αποτελεσματικά τις υλοποιήσεις ανάπτυξης του λογισμικού.

5.3 Εταιρικό τοπίο διακυβέρνησης

Η επιτυχής ανάπτυξη του κύκλου ζωής των SOA (μοντελοποίηση, συναρμολόγηση, ανάπτυξη και διαχείριση) γίνεται στο πλαίσιο αυτού που ονομάζεται ως SOA Governance [141]. Η διακυβέρνηση του SOA είναι η διαδικασία καθορισμού της αλυσίδας αρμοδιοτήτων και επικοινωνιών, πολιτικών, μετρήσεων και μηχανισμών ελέγχου που επιτρέπουν στους ανθρώπους να εκτελούν τις αρμοδιότητές τους.

Συγκεντρωτικά τα χαρακτηριστικά είναι ότι α) Η διακυβέρνηση είναι μια διαδικασία που εφαρμόζεται, και όχι ένα προϊόν που αγοράζεται [142]. β) Η ίδια η διακυβέρνηση έχει ένα σύνολο φάσεων: σχεδιασμός, καθορισμός, ενεργοποίηση και μέτρηση και γ) το SOA ορίζει τέσσερις βασικούς τύπους υπηρεσιών που είναι:

Οι *επιχειρηματικές υπηρεσίες* είναι αφηρημένες κύριες υπηρεσίες που καθορίζουν τις βασικές επιχειρηματικές λειτουργίες. Συνήθως αναπαρίστανται μέσω XML, γλώσσα ορισμού υπηρεσιών Web (WSDL) ή γλώσσα εκτέλεσης επιχειρηματικής διαδικασίας (BPEL).

Οι *εταιρικές υπηρεσίες* υλοποιούν τη λειτουργικότητα που ορίζεται από τις επιχειρηματικές υπηρεσίες. Βασίζονται σε υπηρεσίες εφαρμογών και υπηρεσίες υποδομής για την εκπλήρωση επιχειρηματικών αιτημάτων.

Οι *υπηρεσίες εφαρμογών* είναι λεπτομερές υπηρεσίες που είναι συνδεδεμένες με ένα συγκεκριμένο περιβάλλον εφαρμογής. Αυτές οι υπηρεσίες μπορούν να προβληθούν απευθείας μέσω ενός αποκλειστικού περιβάλλοντος εργασίας χρήστη.

Οι υπηρεσίες παροχής υπηρεσιών υλοποιούν μη λειτουργικές εργασίες όπως η πιστοποίηση, ο έλεγχος, η ασφάλεια και η καταγραφή. Μπορούν να προβληθούν είτε από υπηρεσίες εφαρμογής είτε από υπηρεσίες επιχειρήσεων.

5.4 Χαρακτηριστικά της SOA

Επεκτείνοντας το μοντέλο που παρουσιάστηκε από τον [143] από επτά σε εννιά αρχές του SOA σχεδιασμού μπορεί να περιλάβει με αυτό το τρόπο όλες τις καινούριες αλλαγές που φέρνει σε αυτή την αρχιτεκτονική η ιδέα των μικροπηρεσιών. Αναλυτικά αυτές που πρέπει να ακολουθούνται είναι:

1. Διαλειτουργικότητα-οι υπηρεσίες θα πρέπει να χρησιμοποιούν πρότυπα που επιτρέπουν στους συνδρομητές να χρησιμοποιούν την υπηρεσία. Επιπλέον, με αυτόν τον τρόπο επιτυγχάνεται ευκολότερη ολοκλήρωση.
2. Χαλαρή σύζευξη- οι υπηρεσίες ελαχιστοποιούν τις εξαρτήσεις μεταξύ τους.
3. Αφαιρετικότητα υπηρεσιών-αυτό σημαίνει ότι οι υπηρεσίες αποκρύπτουν τη λογική που υλοποιούν από τον εξωτερικό κόσμο
4. Διαχείριση πόρων/δυνατότητα επαναχρησιμοποίησης της υπηρεσίας- η προγραμματιστική λογική διαιρείται σε υπηρεσίες με σκοπό τη μεγιστοποίηση της επαναχρησιμοποίησης
5. Εντοπισμός υπηρεσίας-οι υπηρεσίες θα πρέπει να μπορούν να εντοπιστούν (συνήθως σε ένα μητρώο υπηρεσιών)

6. Διαρθρωτική ανεξαρτησία/αυτονομία υπηρεσιών-οι υπηρεσίες πρέπει να ελέγχουν τους πόρους που καταναλώνει/βασίζεται η υπηρεσία.
7. Σύνθεση υπηρεσίας/αυτοαναπροσαρμογής- οι υπηρεσίες πρέπει να σπάνε τις μεγάλες εργασίες σε μικρότερες
8. Granularity/Service statelessness- ιδανικά, οι υπηρεσίες θα πρέπει να είναι stateless.
9. Ποιότητα υπηρεσιών- οι υπηρεσίες συμμορφώνονται με ένα σύμβολο σε επίπεδο υπηρεσίας μεταξύ του παρόχου υπηρεσιών και του πελάτη.

5.4.1 Περιορισμοί στη SOA

Δεδομένου ότι ο προσανατολισμός της υπηρεσίας σημαίνει διαφορετικά πράγματα σε διαφορετικούς ανθρώπους, οι προκλήσεις της SOA αρχιτεκτονικής μπορούν να προβληθούν μέσω τριών τομέων: επιχειρήσεις, τεχνική/μηχανική και τις λειτουργίες. Ταυτόχρονα υπάρχει μια τεράστια παραλλαγή της εφαρμογής, των χαρακτηριστικών, των περιγραφών και των δεδομένων των υπηρεσιών, επομένως, παραμένει προβληματική η αποτελεσματική διαχείριση των υπηρεσιών.

Για να ερευνηθεί μια εφαρμογή για την ταξινόμηση των περιοχών που υπάρχουν προκλήσεις που σχετίζονται με συστήματα αρχιτεκτονικής με προσανατολισμό προς την

υπηρεσία, ο παρακάτω πίνακας παρουσιάζει τις προκλήσεις για την υλοποίηση της SOA κατηγοριοποιημένες σε καθέναν από τους τρεις τομείς:

<u>Τομέας</u>	<u>Προκλήσεις</u>
Επιχείρηση	Ορισμός των προτύπων
Επιχείρηση	Προϋπόθεση κατανόησης
Επιχείρηση	Σύνθεση υπηρεσίας
Τεχνική/μηχανική	Ασφάλεια και ακεραιότητα
Τεχνική/μηχανική	Συμβατότητα υπηρεσιών
Τεχνική/μηχανική	Μετεγκατάσταση λογισμικού παλαιού τύπου
Τεχνική/μηχανική	Επικοινωνία σε πραγματικό χρόνο
Τεχνική/μηχανική	Αξιοπιστία και δοκιμές
Λειτουργίες	Οι υπηρεσίες συνεργάζονται
Λειτουργίες	Διακυβέρνηση της πολυπλοκότητας του SOA
Λειτουργίες	Υπηρεσίες παρακολούθησης

Πίνακας 2 Περιοχές τομέα για προκλήσεις της SOA

Στο τομέα της ασφάλειας συγκεκριμένα πέρα από τους παραπάνω περιορισμούς, η SOA δεν ανταποκρίνεται βέλτιστα στα ακόλουθα σενάρια:

1. *Ασφάλεια εφαρμογών παλαιού τύπου (legacy code)*. Παρόλο που οι παλαιού τύπου υπηρεσίες των προγραμμάτων μπορούν να εξωτερικευτούν με τη βοήθεια SOA προσαρμογών, ο σχεδιαστής πρέπει να συνυπολογίσει τους περιορισμούς του υπάρχοντος μοντέλου ασφαλείας. Ο προτεινόμενος προσαρμογέας μπορεί να μην έχει την εσωτερική πληροφορία λειτουργικότητας του κώδικα οπότε να μην μπορεί να δομήσει την υπηρεσία έτσι όπως θα έπρεπε έχοντας σα γνώμονα τα ίδια επίπεδα ασφαλείας.

2. *Χαλαρή σύζευξη υπηρεσιών και εφαρμογών.* Η ασφάλεια του SOA δεν πρέπει να παραβιάζει τις συνολικές αρχές σχεδιασμού λογισμικού, όπως τη χαλαρή σύζευξη των στοιχείων(components) της λύσης. Η υπηρεσία έχει ως στόχο να προσφέρει μια βιώσιμη εφαρμογή, προστατεύοντας τον καταναλωτή υπηρεσιών από τις λεπτομέρειες σχεδιασμού.

3. *Υπηρεσίες που λειτουργούν διαμέσου οργανωτικών ορίων.* Η παραδοσιακή περιμετρική ασφάλεια (enterprise security boundary) μπορεί να μην επαρκεί για τον μετριασμό των ρίσκων που παρουσιάζονται από δια-οργανικές αλληλεπιδράσεις. Ένα σύνολο ελέγχων μπορεί να είναι απαραίτητο για τη διατήρηση της συμμόρφωσης με τις εταιρικές πολιτικές ασφάλειας.

4. *Δυναμικές σχέσεις αξιοπιστίας.* Οι συμμετέχοντες στο SOA πρέπει να εγκαθιδρύσουν αμοιβαία εμπιστοσύνη (ενδεχομένως, και με τους υπεύθυνους για τη φιλοξενία και τη συντήρηση των μητρώων των SOA). Δεδομένου ότι οι σχέσεις αυτές είναι δυναμικής φύσεως, η εμπιστοσύνη είναι, πιθανόν, και αυτή να ακολουθεί αυτή τη δυναμική.

5. *Σύνθετες υπηρεσίες.* Η σύνθεση και η συνάθροιση υπηρεσιών είναι οι δύο μορφές συσχέτισης υπηρεσιών. Η σύσταση της υπηρεσίας μπορεί να παραβιάζει την αρχή της ανεξαρτησίας. Ως εκ τούτου, η ενορχήστρωση είναι η προτιμητέα στρατηγική σύνδεσης υπηρεσιών.

6. *Πρέπει να συμμορφώνεται με έναν αυξανόμενο κατάλογο προτύπων.* Η τυποποίηση των SOA αποτελεί ανησυχία λόγω του αυξανόμενου αριθμού των προτύπων και κανονισμών. Ωστόσο, αυτό δεν διαφέρει από οποιαδήποτε άλλη πλατφόρμα που στόχο έχει τη διασύνδεση υπηρεσιών.

7. *Η ευελιξία.* Οι SOA λύσεις προορίζονται να είναι ευέλικτες και προσαρμόσιμες. Μπορεί να βελτιώσουν και να μειώσουν το χρόνο που χρειάζεται για να βγουν στην αγορά οι παρεχόμενες διεργασίες και οι επιχειρησιακές δραστηριότητες. Η ανάλυση των χαρτοφυλακίων, ο προγραμματισμός για τη μετάβαση σε νέα τροποποιημένη έκδοση και η αρχιτεκτονική διακυβέρνηση είναι σε θέση να παρέχουν στην επιχειρηματική αρχιτεκτονική ευκαιρίες για αλλαγή και συγχώνευση στρατηγικών επιχειρήσεων και στόχων. Με τη μόχλευση των υπηρεσιών που αφορούν τις υπηρεσίες, η στρατηγική του κύκλου προγραμματισμού των επιχειρήσεων μπορεί να μετατραπεί σε ένα πλάνο για συγκεκριμένες πρωτοβουλίες αλλαγών. Στη συνέχεια, ο κύκλος ζωής της SOA αρχιτεκτονικής κατευθύνει την υλοποίηση ενός ή περισσότερων συγκεκριμένων έργων που έχουν οριστεί από το εκάστοτε πλάνο (roadmap). Ως εκ τούτου, οι λύσεις της SOA πρέπει να προσαρμοστούν και να επεκταθούν ανάλογα, προκειμένου να καταστήσουν τις επιχειρηματικές διαδικασίες σχετικές, εξατομικευμένες και ανταποκρινόμενες [144].

5.4.2 Ελλείψεις στη SOA

Καθώς η SOA αρχιτεκτονική θεωρείται μια μοντέρνα προσέγγιση αφαιρετικότητας, εξακολουθεί να υποθέτει ότι υπάρχει μια κεντρικά διαχειριζόμενη αφαιρετική πλατφόρμα. Ορισμένες από τις δυνητικές ελλείψεις ευθυγραμμίζονται με τους περιορισμούς και τις δυνατότητες που παρέχουν οι μικρουπηρεσίες, όπως:

1. Χρησιμοποιώντας το service bus, για να επιτευχθεί SOA αξιοπιστία, η δυνατότητα κλιμάκωσης, έρχεται με ορισμένους περιορισμούς που δεν συναντιούνται στις μικρουπηρεσίες.
2. Η SOA θεωρείται ότι έχει αντίκτυπο στην απόδοση, ωστόσο, αυτό αποδίδεται σε μεγάλο βαθμό στο πρωτόκολλο, και όχι στην τυχόν μη ορθή υλοποίηση της υπηρεσίας. Κακώς ενορχηστρωμένη υπηρεσία, ωστόσο, μπορεί να παραβιάσει και τα SLA/QSA.
3. Η άρτια και ορθά δηλωμένη περιγραφή της υπηρεσίας είναι απαραίτητη για τη χρηστικότητα της. Συνεπώς, αναμένεται ότι ο κύκλος ζωής των υπηρεσιών ελέγχεται από την κατάλληλη διαδικασία διακυβέρνησης.
4. Ο έλεγχος του SOA είναι πιθανό να είναι τόσο σύνθετος όσο κάθε άλλη κατανομημένη λύση, όπου ο στόχος του καταναλωτή, έχοντας συγκεκριμένα SLA/QSA, εξαρτάται σε μεγάλο βαθμό από την ικανότητα των υπο-υπηρεσιών να διατηρήσουν το OLA (Operational-Level Agreement).

5.4.3 Πως οι μικρουπηρεσίες επεκτείνουν τη SOA

Το μονολιθικό μοντέλο ανάπτυξης εφαρμογών αναστέλλει την ικανότητα των οργανισμών να εξελίσσονται ταχέως σε επιχειρηματικά σενάρια. Για αυτό το λόγο και θεωρείτε ξεπερασμένο. Όλες οι παραλλαγές της πολύ-κλιμακούμενης αρχιτεκτονικής αναπτύχθηκαν και δημιουργήθηκαν ως βιώσιμες και προτιμώμενες εναλλακτικές λύσεις για το μονολιθικό ομόλογό της. Μια αρχιτεκτονική μικρουπηρεσιών, παρουσιάζει εξαιρετική ευελιξία, επιτρέπει την ταχεία εξέλιξη και, δυνητικά, μειώνει το κόστος ανάπτυξης στο ίδιο επίπεδο πολυπλοκότητας. Οι μικρουπηρεσίες είναι μια αναδυόμενη αρχιτεκτονική προσέγγιση που επεκτείνει τις πρακτικές των SOA και ξεπερνά τις παραδοσιακές ελλείψεις της SOA.

Παρόλο που η *microservice* αρχιτεκτονική δεν εισάγει τίποτα ριζικά νέο, αποτελεί τη λογική εξέλιξη της SOA και υποστηρίζει σύγχρονα σενάρια επιχειρηματικής χρήσης. Παρόμοια με τη SOA, μια προσέγγιση μικρουπηρεσιών αποσυνθέτει μεμονωμένες εφαρμογές σε πολλαπλές, διακριτές υπηρεσίες και μονάδες λειτουργικότητας, που υλοποιούνται από τις ατομικές μικρουπηρεσίες χρησιμοποιώντας την αρχή της ενιαίας ευθύνης. Σε αυτή την αρχιτεκτονική ενώνονται για να σχηματίσουν μια χρήσιμη δραστηριότητα επιχειρηματικής διαδικασίας και να δημιουργήσουν μια ροή υπηρεσιών ώστε να τη παρέχουν στο τελικό χρήστη. Οι πρόσθετες αρχές βοηθούν να παρέχουν αυτόνομες, χαλαρά συνδεδεμένες, ανθεκτικές και χαμηλής συνεκτικότητας λύσεις.

Η προσέγγιση των μικρουπηρεσιών εισάγει λοιπόν επιπλέον πρακτικές και αρχές στην αρχιτεκτονική δομή του SOA. Τα κύρια χαρακτηριστικά είναι α) η αντιστοίχιση του περιεχομένου και της πληροφορίας (Context Mapping) β) η χαλαρή ένωση των υπηρεσιών και η υψηλή συνοχή σε κάθε αυτοτελής μικρο-υπηρεσία. γ) η αρχή του Shared-nothing architecture δ) η δυνατότητα ότι μπορεί να υλοποιηθεί όλη η μικρο-υπηρεσία χωρίς την εξάρτηση κάποιου άλλου συνδέσμου (full-stack development) ε) και η δυνατότητα της παράλληλης ταυτόχρονης υλοποίησης της υπηρεσίας μιας και μπορεί το κάθε κομμάτι μικρο-υπηρεσία να υλοποιείται ταυτόχρονα.

Συγκριτικά οπότε υπάρχει μια στροφή προς μια προσέγγιση που είναι πιο κοντά στη λογική της διαχείρισης των δεδομένων και των ενεργειών που πρέπει να γίνονται σε αυτά. Από την ιδέα της SOA “προσέφερε όσο περισσότερα μπορείς” τώρα αυτό τροποποιείται και ακολουθείται μια πιο συντηρητική προσέγγιση που κάνει ευκολότερο τον έλεγχο των όλων διαδικασιών. Δηλαδή γίνεται “προσέφερε όσα λιγότερα γίνεται”. Μπορεί αυτό να φαίνεται αντίθετο αλλά στην ουσία δεν είναι, μιας και οι υπηρεσίες και ο διαχειριστής της αρχιτεκτονικής είναι ελεύθεροι να επιλέξουν αυτά που θα εκθέσουν είτε στο διαδίκτυο είτε στους πιο κλειστούς κύκλους υπηρεσιών εντός της εταιρίας. Με αυτό το τρόπο διαφυλάσσεται η πληροφορία από άθελα αρχιτεκτονικά σφάλματα που θα μπορούσαν να εκθέσουν επιχειρησιακά μυστικά και δεδομένα. Στο κομμάτι της μεταφοράς ακολουθείται ένα πιο ελαφρύ μοντέλο ανταλλαγής μηνυμάτων σε σύγκριση με το Enterprise Service Bus (ESB) βασισμένο κυρίως σε HTTP/REST καθώς και σε συστήματα ουρών. Ταυτόχρονα επεκτείνει και κάνει πιο διαδεδομένη τη χρήση των containers μιας

και δομικά έχει οριστεί με τέτοιο τρόπο που αυτές οι δυο τεχνολογίες να συμπλέουν. Τέλος στο κομμάτι της διαχείρισης προσφέρει μια πιο χαλαρή αντιμετώπιση σε σχέση με τη SOA ακολουθώντας τις τάσεις αποκεντροποίησης των υπηρεσιών και της διαχείρισης που ισχύουν. Σε αυτό βοηθά και η επέκταση σε μη σχεσιακές βάσεις δεδομένων, βάσεις που είναι ιδιαίτερα δημοφιλής στις μικρουπηρεσίες λόγω της πιο δεδομενοστραφής προσέγγισης τους.

5.4.4 Πλεονεκτήματα των μικρουπηρεσιών

Οι περισσότερες διαφορές μεταξύ των Μικρουπηρεσιών και της SOA βασίζονται στο γεγονός ότι προσπάθησε να βελτιώσει μερικά προβλήματα που φάνηκε ότι αντιμετώπιζε το μοντέλο. Χωρίζοντας ξανά σε κατηγορίες ανάλογα με την εκάστοτε λογική παρουσιάζονται ανά κατηγορίες τα πλεονεκτήματα που εισάγονται.

Όντας μια αρχιτεκτονική που βασίστηκε στη έννοια και στις ανάγκες των επιχειρηματικών αναγκών στον Πίνακα 3 Επιχειρηματικά πλεονεκτήματα Πίνακα 3 παρουσιάζονται τα πλεονεκτήματα που προσφέρει σε όποιον οργανισμό ακολουθήσει αυτό το μοντέλο λειτουργίας.

<p>Ευθυγράμμιση με την επιχείρηση</p>	<p>Μια μικρουπηρεσία βασίζεται σε μια μοναδική ατομική επιχειρηματική λειτουργία που διαθέτει δυνατότητες ατομικής επιχειρηματικής δραστηριότητας, η οποία ανήκει σε μια ενιαία επιχειρηματική μονάδα. Έτσι, είναι σε θέση να</p>
--	--

	συγχρονίζεται με τις ανάγκες της επιχειρηματικής μονάδας εξελισσόμενη ανάλογα με αυτή.
Γίνεται ευέλικτη και ανταποκρινόμενη στις επιχειρηματικές ανάγκες	Δεν χρειάζεται να παρθεί μια ολόκληρη εφαρμογή για να ενημερωθεί ή να κλιμακωθεί απλά αυτή η δομική αλλαγή περιορίζεται σε ένα μόνο στοιχείο της. Κάθε υπηρεσία μπορεί να ενημερωθεί ή να κλιμακωθεί ανεξάρτητα. Αυτό μας δίνει τη δυνατότητα να ανταποκριθούμε ταχύτερα.
Ενεργοποίηση ομάδων	Μια σειρά από κατανεμημένες και αυτόνομες ομάδες που χρησιμοποιούν το πρότυπο συνεχούς ενοποίησης (CI) και συνεχούς ανάπτυξης (CD) είναι σε θέση να εργαστούν σε ανεξάρτητες υπηρεσίες με σχεδόν συνεχή ενοποίηση του λογισμικού, με την αυτοματοποιημένη ανάπτυξη του ενσωματωμένου κώδικα.
Δυνατότητα υψηλής διαδραστικότητας από τον χρήστη σε όλα τα στάδια δίχως να επηρεάζει την διαθεσιμότητα των υπηρεσιών	Η ανθεκτικότητα που παρέχεται μέσω της παραλληλίας των μικρουπηρεσιών, που ενεργοποιείται από την ανεξαρτησία τους, επιτρέπει την ταχεία ανακατεύθυνση και συντήρηση. Αυτά τα χαρακτηριστικά οδηγούν σε υψηλή διαθεσιμότητα και απρόσκοπτη αλληλεπίδραση του χρήστη.
Εγγενώς διαλειτουργικές υπηρεσίες	Οι υπηρεσίες συνήθως υλοποιούνται με τη χρήση υποχρεωτικών σαφώς καθορισμένων, δημοσιευμένων διασυνδέσεων, σύμφωνα με καθορισμένους κανόνες υπηρεσιών. Έτσι, ο χρόνος και το κόστος που συνδέεται με τη σύνθεση των αιτήσεων μειώνεται, επιτρέποντας ταχύτερη ανταπόκριση στις αλλαγές στο marketplace ή στην είσοδο σε μια νέα αγορά

Πίνακας 3 Επιχειρηματικά πλεονεκτήματα

Στο κομμάτι των τεχνικών χαρακτηριστικών και των αρχιτεκτονικών δομών που ένας αναλυτής είτε προγραμματιστής θα μπορούσε να εκμεταλλευτεί, μπορούν να θεωρηθούν καινοτόμες αλλαγές στο τρόπο δημιουργίας του συστήματος, η δυνατότητα πολύ γλωσσικότητας στο προγραμματισμό, όπου επιτρέπει τη συλλογή και την επιλογή της καλύτερης γενιάς τεχνολογίας, αλλά και εργαλείων και πλατφόρμων. Στην φιλοσοφία των μικρουπηρεσιών δεν υπάρχει προεπιλογή είτε ‘κλείδωμα’ γλώσσας ή τεχνολογίας, καθώς κάθε υπηρεσία λειτουργεί ανεξάρτητα, και με αυτό το τρόπο μπορεί να επιλεγεί οποιαδήποτε γλώσσα ή τεχνολογία για να την ανάπτυξη. Πρέπει όμως να διασφαλιστεί ότι είναι σταθερές οι μέθοδοι και τα κατηγορήματα που το API έχει, καθώς και να κλειδώσει η αναμενόμενη έξοδο τους. Ακόμα ένα τεχνικό πλεονέκτημα είναι ότι κάθε υπηρεσία μπορεί να αναπτυχθεί ανεξάρτητα. Αυτό συμβαίνει γιατί από τη δομή του το σύστημα έχει φτιαχτεί με τέτοιο τρόπο ώστε να έχει αναλυθεί και να έχει διαχωρίσει πολλές -εάν όχι όλες- τις υπηρεσίες σε ανεξάρτητες οντότητες. Αυτή η επιλογή προσφέρει ακόμα το πλεονέκτημα ότι μετά την συγγραφή του κώδικα μιας υπηρεσίας, αυτή μπορεί να χρησιμοποιηθεί σε άλλα έργα, όπου απαιτείται η ίδια λειτουργικότητα. Τέλος όσο αναφορά τα τεχνικά πλεονεκτήματα θα πρέπει να τονιστεί εκτός από το κομμάτι της συγγραφής του κώδικα, τα πλεονεκτήματα στο κομμάτι της μορφοποίησης του, των αλλαγών, της επαλήθευσης και της ενσωμάτωσης στο όλο σύστημα. Εκεί η αρχιτεκτονική μικρουπηρεσιών επιτρέπει τη συνεχή παράδοση(continuous delivery) και λειτουργεί ακολουθώντας και υποστηρίζοντας τις τεχνικές που προγράμματα CI-CD προσπαθούν να προσφέρουν.

Στη κατηγορία με τα επιχειρησιακά οφέλη μπορεί να απαριθμηθεί η δυνατότητα εάν μια υπηρεσία αποτύχει, τότε η αποτυχία της δεν έχει καθολικό αντίκτυπο. Αυτό βοηθά ιδιαίτερα στον εντοπισμό σφαλμάτων. Ακόμα τα επιμέρους κομμάτια μπορούν να αναπτυχθούν σε πολλαπλούς διακομιστές ή ακόμη και σε πολλαπλά κέντρα δεδομένων, καθώς υπάρχει η δυνατότητα να λειτουργεί πολύ καλά η αρχιτεκτονική δομή με ενορχηστρωτές για containers όπως το Kubernetes, DC/OS και το Docker Swarm.

5.4.5 Προκλήσεις

Το πιο σημαντικό μειονέκτημα καθώς και η πρόκληση που υπάρχει είναι η πολυπλοκότητα στο επιχειρησιακό περιβάλλον ως αποτέλεσμα πολλών μικροπηρεσιών που αναπτύσσονται στην παραγωγή, συμπεριλαμβανομένης της συνέπειας των δεδομένων που πρέπει να υπάρχει, της παρακολούθησης των υπηρεσιών, της αντιμετώπισης των επιμέρους προβλημάτων και της διασφάλισης της ασφάλειας των containers. Η διαχείριση μιας πληθώρας καταναμημένων υπηρεσιών σε μεγάλη κλίμακα είναι δύσκολη μιας και

α) Οι ομάδες έργου πρέπει να ανακαλύπτουν εύκολα τις υπηρεσίες ως πιθανούς υποψηφίους πελάτες. Οι υπηρεσίες αυτές θα πρέπει να παρέχουν τεκμηρίωση, κονσόλες δοκιμών ώστε έτσι η επαναχρησιμοποίηση να είναι σημαντικά ευκολότερη σε σύγκριση με την κατασκευή από την αρχή. Αυτό αποτελεί μια μεγάλη επιχειρηματική και επιχειρησιακή πρόκληση μιας και αλλάζει εντελώς το τρόπο όχι μόνο που σκέφτεται ο προγραμματιστής αλλά και το τρόπο που γράφει κώδικα.

β) Οι αλληλεξαρτήσεις μεταξύ των υπηρεσιών πρέπει να παρακολουθούνται στενά. Ο χρόνος διακοπής των υπηρεσιών, οι διακοπές λειτουργίας, οι αναβαθμίσεις υπηρεσιών μπορούν να έχουν διαδοχικές μεταγενέστερες επιπτώσεις και ο αντίκτυπος αυτός θα πρέπει να αναλυθεί προληπτικά. Αυτό αποτελεί μια τεχνική πρόκληση μιας και εργαλεία monitoring τέτοιας κλίμακας δεν υπάρχουν είτε δεν είναι αποδοτικά αφού καταναλώνουν αρκετούς από τους πόρους των συστημάτων με αποτέλεσμα να δημιουργούν στο τέλος μεγαλύτερο πρόβλημα από ότι αυτό που είχαν καλεστεί να λύσουν.

γ) Ο κύκλος ζωής των υπηρεσιών SDLC θα πρέπει να είναι πολύ αυτοματοποιημένος. Αυτό απαιτεί τεχνολογία, όπως τεχνολογίες αυτοματισμού ανάπτυξης και πλαίσια CI-CD. Παρόλα αυτά το σημαντικότερο είναι η πειθαρχία από τις ομάδες προγραμματιστών και επιχειρήσεων. Το πρόβλημα αυτό είναι μια επιχειρησιακή πρόκληση που επεκτείνεται σε συστημικό πρόβλημα που περιέχει εκτός από τεχνικά χαρακτηριστικά και ανθρωποκεντρικά .

Πέρα όμως από τις τρεις βασικές προκλήσεις υπάρχουν και επιμέρους τεχνικές που μπορούν να ενταχθούν ανάλογα με την ανάλυση σε κάποιες από τις βασικές. Στο κομμάτι της διαχείρισης αυτών των υπηρεσιών σε κλίμακα, ορισμένες προκλήσεις εγείρονται λόγω των διαφοροποιήσεων που υπάρχουν. Αυτές είναι: α) ο εντοπισμός (Service Discovery) και τεκμηρίωση υπηρεσίας, β) η ανοχή σε σφάλματα (Fault tolerance) γ) η ποιότητα υπηρεσιών (Quality of Service), δ) Η ασφάλεια των υπηρεσιών ε) οι αιτήσεις ιχνηλασιμότητας (Request traceability) και στ) η διαλογή αποτυχιών και σφαλμάτων.

Τέλος δομικά οι μικρουπηρεσίες σαν μοντέλο αρχιτεκτονικής υποχρεώνει την αλλαγή κουλτούρας και στρατηγικής σε όλα τα κομμάτια ανάλυσης και συγγραφής του κώδικα. Από το κομμάτι της σχεδίασης μέχρι το κομμάτι του ελέγχου και των δοκιμών πρέπει η φιλοσοφία να παραμένει η ίδια χωρίς να προσαρμόζεται στις ανάγκες της μορφοποίησης και τις απλής διάσπασης των διαδικασιών. Οπότε συστημικά θα πρέπει να λογιστούν παράγοντες όπως:

α) *Η επιλογή του σωστού μεγέθους υπηρεσίας* - Σπάζοντας την μονολιθική εφαρμογή ή δημιουργώντας μικρουπηρεσίες από το μηδέν, είναι πολύ σημαντικό να επιλεγεί η σωστή λειτουργικότητα για μια υπηρεσία. Για παράδειγμα, αν δημιουργηθεί μια μικρουπηρεσία για κάθε λειτουργία ενός μονολιθικού μοντέλου, τότε θα καταλήξουμε με πολλές μικρές υπηρεσίες, που θα φέρουν περιττή πολυπλοκότητα.

β) *Η δοκιμή του κώδικα και της αρχιτεκτονικής* - Με πολλές υπηρεσίες και την αλληλεξάρτησή τους, μερικές φορές γίνεται δύσκολο να γίνει από άκρη end-to-end testing μιας μικρουπηρεσίας.

γ) *Η δια-υπηρεσιακή επικοινωνία* - Η επικοινωνία μεταξύ των υπηρεσιών μπορεί να είναι πολύ δαπανηρή εάν δεν υλοποιηθεί σωστά. Υπάρχουν επιλογές όπως π.χ. το message passing είτε το RPC και πρέπει να επιλεγεί αυτό που ταιριάζει στην απαίτηση και έχει την ελάχιστη επιβάρυνση.

ε) *Η διαχείριση βάσεων δεδομένων* - Όταν πρόκειται για την αρχιτεκτονική των μικρουπηρεσιών, μπορεί να αποφασιστεί να εφαρμοστεί μια βάση δεδομένων τοπική σε

για μικρουπηρεσία. Ωστόσο, για να κλείσει ένας επιχειρησιακός βρόχος, ενδέχεται να χρειαστούν αλλαγές και σε άλλες σχετικές βάσεις δεδομένων. Αυτό μπορεί να δημιουργήσει προβλήματα (π.χ. διαμερισματοποίηση των βάσεων δεδομένων).

στ) *Η εγκατάσταση των μικρουπηρεσιών* - Μπορεί εύκολα να αναπτυχθεί μια μονολιθική εφαρμογή. Ωστόσο, για να αναπτυχθεί μια μικρουπηρεσία, πρέπει να χρησιμοποιήσουμε ένα καταναμημένο περιβάλλον όπως το Kubernetes.

ζ) *Η παρακολούθηση* - Η παρακολούθηση μεμονωμένων υπηρεσιών σε περιβάλλον μικρουπηρεσιών μπορεί να αποτελέσει πρόκληση. Αυτή η πρόκληση αντιμετωπίζεται και αναπτύσσεται από ένα νέο σύνολο εργαλείων, όπως το Sysdig το Datadog, για την παρακολούθηση και τον εντοπισμό σφαλμάτων στις μικρουπηρεσίες.

5.5 Blueprint

Τα στοιχεία και οι προκλήσεις που υπάρχουν δημιουργούν την ανάγκη για ένα σύστημα διαχείρισης και αυστηρής ανάλυσης των υπηρεσιών ώστε να μπορεί να παρουσιαστεί και να αναλυθεί κάθε στοιχείο διεξοδικά. Αυτό θα επιτρέψει μια διαφανή διαδικασία επιλογών υπηρεσιών σαν προϊόντα σε συστήματα αγορών όπως αυτό που παρουσιάστηκε και αναλύθηκε. Η δυνατότητα αυτή θα δημιουργήσει ένα καινούριο χώρο

για αναζήτηση και ανάλυση υπηρεσιών που θα ακολουθεί όμως το μοντέλο της MSA (MicroService Architecture). Αυτά τα προϊόντα θα μπορούν να φιλοξενοούνται σε containers είτε θα υπάρχει ακριβής ανάλυση για τη δυνατότητα εγκατάστασης κάτω από τις εγκαταστάσεις του πελάτη.

Οι containers εφαρμογών και η αρχιτεκτονική των μικροπηρεσιών χρησιμοποιούνται για τον σχεδιασμό, την ανάπτυξη και τη σχεδίαση εφαρμογών που αξιοποιούν ευέλικτες προσεγγίσεις ανάπτυξης λογισμικού. Το βασικό όμως θέμα είναι ότι και η ασφάλεια πρέπει να ενσωματωθεί σε αυτές τις προσεγγίσεις ανάπτυξης. Για αυτό το λόγο πρέπει να υπάρχει ένας προσδιορισμός και μια σειρά από συστάσεις και πρακτικές βελτιώσεων για την αντιμετώπιση των προκλήσεων όσον αφορά τη διασφάλιση των application containers τόσο από τη μεριά του προγραμματιστή όσο και του χειριστή και του αρχιτέκτονα.

Στην αγορά υπηρεσιών, οι ανάγκες σε λογισμικό είναι πιο απαιτητικές από ποτέ. Οι εταιρικές εφαρμογές πρέπει να υποστηρίζουν συχνά μεταβαλλόμενες απαιτήσεις όπως ακόμα να έχουν λύσει ζητήματα κλιμάκωσης, γρήγορη υιοθέτηση αλλαγών και γρήγορες διαδικασίες παράδοσης και ανάπτυξης λογισμικού. Ο τρόπος σχεδίασης με μικροπηρεσίες, παρέχει τη δυνατότητα να σχεδιάσει και να διαχειρίζεται λύσεις λογισμικού που είναι πιο ευθυγραμμισμένες με τις μεταβαλλόμενες επιχειρηματικές ανάγκες, και αυτός είναι ο σημαντικότερος λόγος που οι επιχειρήσεις την ενστερνίζονται σταδιακά. Σε ένα απαιτητικό και δυναμικό επιχειρηματικό σενάριο, οι επιχειρήσεις πρέπει να ανταποκριθούν γρήγορα στις μεταβαλλόμενες ανάγκες, αλλά τα μονολιθικά συστήματα

λογισμικού δεν τους επιτρέπουν να το πράξουν. Οι μικρουπηρεσίες είναι πιο ευέλικτες και ανταποκρίνονται στις μεταβαλλόμενες καταστάσεις. Η προσέγγιση του συνδυασμού μικρών ανεξάρτητων συστατικών-υπηρεσιών, επιτρέπει στις εφαρμογές να έχουν υψηλή διαθεσιμότητα, δυνατότητα κλιμάκωσης και δυνατότητες απομόνωσης βλαβών. Ένα μίγμα από διάφορες τεχνολογικές στοίβες προσφέρει όλα τα οφέλη που απαιτεί ένα σύγχρονο λογισμικό.

Παρόλα αυτά η αρχιτεκτονική των μικρουπηρεσιών αντιμετωπίζει μείζονες προκλήσεις. Αυτές οι προκλήσεις θα μπορούσαν να οδηγήσουν έναν δυνητικό αγοραστή να απορρίψει μία μικρουπηρεσία όταν του δίνεται η επιλογή. Ανεξάρτητα από το πόσα οφέλη θα έφερνε η αρχιτεκτονική των μικρουπηρεσιών σε ένα σύστημα και ένα προϊόν στο σύνολό του, εξακολουθούν να υπάρχουν ορισμένα "μειονεκτήματα" που πρέπει να ληφθούν υπόψη. Μια τέτοια πρόκληση, για παράδειγμα, είναι η έλλειψη παρακολούθησης για τις αλληλεξαρτήσεις (εάν υπάρχουν) μεταξύ των υπηρεσιών. Μέχρι τώρα, το θέμα αυτό αποτέλεσε τεχνική πρόκληση. Ο χρόνος διακοπής μιας υπηρεσίας και οι αναβαθμίσεις της θα πρέπει να παρακολουθούνται, ώστε οι κατάλληλες ενέργειες να λαμβάνονται έγκαιρα. Αυτό το μειονέκτημα μπορεί να εξαλειφθούν εάν η ομάδα ανάπτυξης της εταιρείας (ο αγοραστής της μικρουπηρεσίας) έχει πλήρη πρόσβαση σε κάθε υπηρεσία, έτσι ώστε να είναι σε θέση να κάνει κάθε αναγκαία ενέργεια. Επιπλέον, η ανάπτυξη μιας βελτιωμένης αρχιτεκτονικής μικρουπηρεσιών (σε σύγκριση με τις προηγούμενες αρχιτεκτονικές/λύσεις και τα μειονεκτήματά τους) θα ήταν χρήσιμη, με βάση ένα ενημερωμένο/αναβαθμισμένο μοντέλο σύνδεσης μεταξύ των υπηρεσιών.

Με την ακραία ανάπτυξη του υπολογιστικού νέφους τα τελευταία χρόνια, οι υπηρεσίες υπολογιστικού νέφους έχουν εξελιχθεί σε άυλης αξίας αγαθά των επιχειρήσεων. Οι περισσότερες από τις εταιρικές εφαρμογές στρέφονται σε λύσεις υπηρεσιών cloud όπως το IaaS (υποδομή ως υπηρεσία), PaaS (πλατφόρμα ως υπηρεσία) και CaaS (κοντέινερ ως υπηρεσία) σε λύσεις χωρίς διακομιστή, όπως το FaaS (λειτουργία ως υπηρεσία), προκειμένου να επιτευχθεί κλιμάκωση και ανθεκτικότητα στη διαθεσιμότητα των υπηρεσιών. Ωστόσο, πρόκειται για ένα σημείο όπου η ποικιλία τόσο στις υπηρεσίες υπολογιστικού νέφους όσο και στους παρόχους υπολογιστικού νέφους αυξάνεται κατά τέτοιον τρόπο, ώστε το σενάριο μιας εφαρμογής δομημένης από μικροπηρεσίες που αποτελείται από υπηρεσίες και τεχνολογίες cloud μεικτών προμηθευτών να καθίσταται αναπόφευκτη. Μέχρι τώρα, ο συνδυασμός και ενορχήστρωση των διαφόρων υπηρεσιών cloud από διαφορετικούς παρόχους αποδεικνύεται ότι είναι ένα πολύ δύσκολο, και στην περίπτωση ορισμένων υπηρεσιών cloud που είναι αυστηρά εξαρτώμενη από την πλατφόρμα, π.χ. FaaS, είναι σχεδόν αδύνατο.

Η κύρια πρόκληση της αρχιτεκτονικής μικροπηρεσιών (MSA) είναι η δόμηση της έτσι ώστε να μπορεί να ακολουθήσει όλες τις αρχές των μικροπηρεσιών, όπως τα χαρακτηριστικά της αμεταβλητότητας (Immutability), του διαχωρισμού των υπηρεσιών και των λογικών οντοτήτων (Separation of concerns), της ατομικότητας (Atomicity), της ανθεκτικότητας (Durability) και της επαναχρησιμοποίησης (Reusability).

Η ανάλυση του προβλήματος από τη γενική εικόνα στα επιμέρους επίπεδα, όταν η πλειονότητα των εντοπισμένων λειτουργικών συστατικών συμμορφώνονται με τις

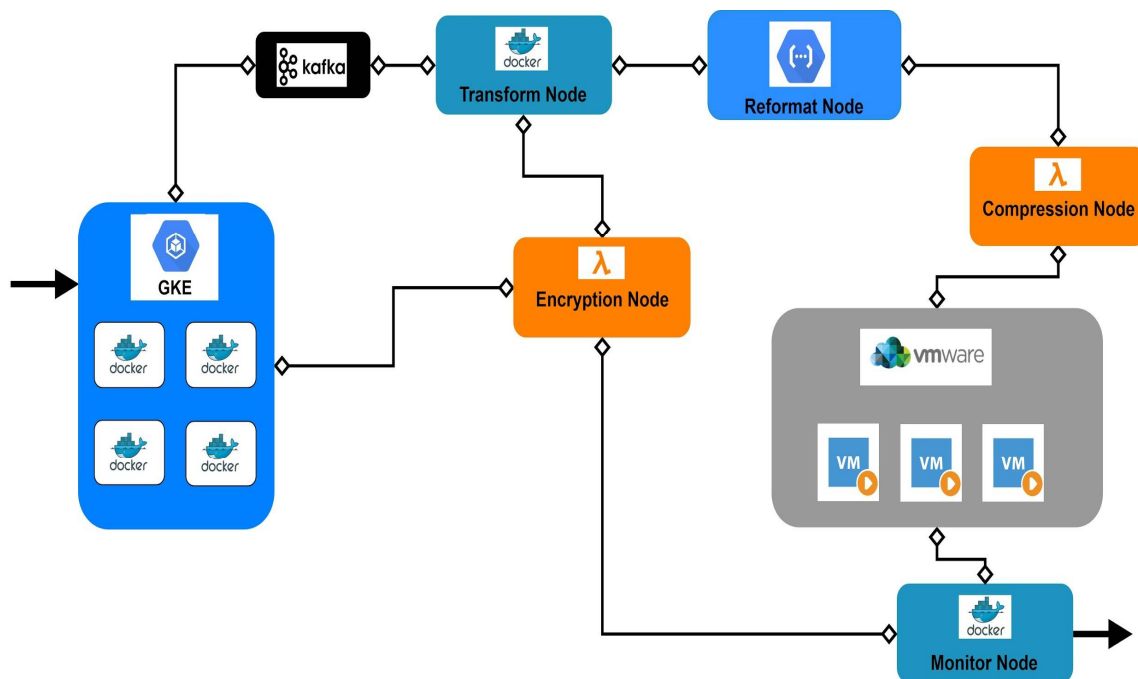
παραπάνω αρχές, είναι μια επαναληπτική (και μάλλον κουραστική) διαδικασία. Ταυτόχρονα η πολυπλοκότητα υπακούει στην αρχή της διατήρησης, όπου δηλαδή για τα κλειστά συστήματα η πολυπλοκότητα ενός συγκεκριμένου συστήματος παραμένει σταθερή. Απλοποιώντας τα επιμέρους στοιχεία, μετατοπίζεται η πολυπλοκότητα στον υπεύθυνο επεξεργασίας (ή τον ενορχηστρωτή) που είναι το στοιχείο που είναι υπεύθυνο για τη συμπεριφορά και την σωστή λειτουργία της συνολικής λύσης.

Παρόλα αυτά τα θέματα, η MSA έχει τη δυνατότητα ταχείας αντίδρασης στους συνεχώς μεταβαλλόμενους επιχειρηματικούς κινδύνους με ταχείες τροποποιήσεις στις επιχειρηματικές λύσεις. Ωστόσο, η ευελιξία αυτή έρχεται με το κόστος της θέσπισης σοβαρών θεμελίων στη βάση της επιχείρησης.

5.5.1 Ενορχήστρωση υπηρεσιών

Προκειμένου οι πλατφόρμες και οι εφαρμογές να έχουν ανταγωνιστικό πλεονέκτημα, πρέπει να εκτελούν πρόσθετες ενέργειες, όπως η παρακολούθηση δεδομένων, η μορφοποίηση δεδομένων και η σύνδεση τους με πολλαπλές προελεύσεις πηγών δεδομένων. Αν και αυτές οι ενέργειες δεν είναι οι κύριες λειτουργίες που εξυπηρετεί το λογισμικό, είναι υποχρεωτικές στην εποχή των μεγάλων δεδομένων όπου τα θέματα απόδοσης, ελέγχου, απορρήτου και διαθεσιμότητας είναι επιβεβλημένα. Επιπλέον, είναι σημαντικό να υπάρχει ευελιξία ώστε να χρησιμοποιούνται τα καλύτερα διαθέσιμα στοιχεία από τον πάροχο που είναι ειδικός στον τομέα χωρίς την ανάγκη να

υλοποιηθούν από τον προγραμματιστή. Έτσι, εργασίες όπως η ενίσχυση της προστασίας της ιδιωτικής ζωής και των προσωπικών δεδομένων, η αύξηση της εμπιστοσύνης των χρηστών στους παρόχους υπηρεσιών cloud, η απλοποίηση της διακυβέρνησης, η διαχείριση κινδύνων και η συμμόρφωση των υπηρεσιών cloud μπορούν να προβάλλονται ως στοιχεία που ακολουθούν τα τελευταία στάνταρτ από τους ειδικούς τομέα που η αποκλειστική τους δουλειά είναι αυτό το πράγμα. Αυτοί είναι μερικοί από τους λόγους για τους οποίους οι αρχιτεκτονικές που βασίζονται στην MSA είναι όλο και πιο σημαντικοί καθώς παρέχει αρθρωτότητα που ενδεχομένως απλουστεύει την ανάπτυξη και τον επανασχεδιασμό των προσπαθειών.



Εικόνα 28 Διάγραμμα υπηρεσιών

Μια πλατφόρμα, ακολουθώντας αυτή τη δομή, αποτελείται από μικρές και ελαφριές υπηρεσίες που είναι χαλαρά συζευγμένες, παρέχοντας έτσι ένα αποτέλεσμα που στον κόσμο της αρχιτεκτονικής λογισμικού περιγράφεται ως ‘fine-grained’. Αυτές οι υπηρεσίες μπορούν να έχουν διαφορετικά περιβάλλοντα χρόνου εκτέλεσης και μπορούν να αναπτυχθούν χρησιμοποιώντας μια ποικιλία μοντέλων ανάπτυξης σύννεφων από πολλές υπηρεσίες υπολογιστικού νέφους. Λαμβάνοντας υπόψη ότι όλο και περισσότερες

εφαρμογές οργανώνονται με παρόμοιο τρόπο, ομαδοποιώντας μια πληθώρα κατασκευαστικών στοιχείων και υπηρεσιών υπολογιστικού νέφους, αυτές τείνουν να έχουν ιεραρχική πολυπλοκότητα, καθιστώντας σαφές ότι ένα ενοποιημένο πρότυπο είναι ουσιώδους σημασίας. Ο σχεδιασμός και η ενορχήστρωση αυτών των εφαρμογών είναι βέβαιο ότι θα είναι πιο περίπλοκη διαδικασία οπότε η δημιουργία ενός προτύπου καθορισμού προδιαγραφών μπορεί να είναι πολύ χρήσιμο. Αυτό μπορεί να είναι ένα προσχέδιο που συγκεντρώνει και οργανώνει τις βασικές διαρθρωτικές πληροφορίες μιας εφαρμογής όπου περιγράφει τον container που θα ενορχηστρώνει την ανάπτυξη και την εκτέλεσή του. Με αυτόν τον τρόπο μια εταιρεία μπορεί να διαχειριστεί την ταχύτητα και την πολυπλοκότητα των εφαρμογών της, κατανέμοντας το συνολικό φόρτο εργασίας. Με αυτό το τρόπο επομένως χαμηλώνει το κόστος και αυξάνει την αποδοτικότητα σε όλες τις υποδομές της όπως επίσης βελτιστοποιεί τη μέθοδο δομής και συγγραφής των έργων μαζί με τη διαδικασία δοκιμών και επικύρωσης αυτού.

Η αξιοποίηση της ύπαρξης αυτού του προτύπου επιτρέπει να ληφθεί πλήρως ο έλεγχος των συστημάτων και μπορεί να ανοίξει μια ολόκληρη νέα αγορά, όπου τα εξαρτήματα, ως δομοστοιχεία, μπορούν να δημιουργηθούν και να διατεθούν από οποιονδήποτε και να επισυνάπτονται εύκολα σε υπάρχουσες εφαρμογές. Η διαφάνεια αυτού του προτύπου καθιστά δυνατή τη χρήση εργαλείων τρίτων για να χρησιμοποιούνται από τις εφαρμογές ως μέσο ελέγχου του μητρώου ασφάλειας, εμπιστοσύνης και αξιοπιστίας, καθώς και για τη διασφάλιση της συμμόρφωσης τόσο των λειτουργικών όσο

και των μη λειτουργικών ιδιοτήτων της εφαρμογής, ακολουθώντας όμως ορισμένους κανονισμούς.

5.5.2 Ενοποιημένο πρότυπο μικροπηρεσιών

Στη προηγούμενη ενότητα υπογραμμίστηκε η ανάγκη ενός ενοποιημένου πρωτοτύπου, που θα ονομάζεται Blueprint, που θα λειτουργήσει ως περιγραφή για κάθε τύπο εφαρμογής μικροπηρεσιών. Ο κύριος σκοπός του Blueprint είναι να τυποποιήσει τον τρόπο με τον οποίο οι εφαρμογές των μικροπηρεσιών είναι δομημένες και να τις περιγράψει με τέτοιο τρόπο, έτσι με την υιοθέτηση αυτού του προτύπου περιγραφής, τέτοιες εφαρμογές να μπορούν να αναπτυχθούν και να εννοχηστρωθούν με ανεξάρτητο τρόπο. Όπως αναφέρεται μέσω αυτής της τυποποίησης, οποιαδήποτε εφαρμογή μικροπηρεσιών μπορεί να “συσκευαστεί” ως προϊόν, έτσι μπορεί εύκολα να συνδυαστεί και να χρησιμοποιηθεί ως αυτόνομη ενότητα από άλλες εφαρμογές που ανοίγουν με αυτόν τον τρόπο τη διαδρομή για μια νέα αγορά.

Το Blueprint που παρουσιάζεται θα χωριστεί σε τέσσερα βασικά τμήματα. Στη συνέχεια, κάθε ενότητα θα διαιρεθεί σε περισσότερα μέρη, ώστε να κατανοηθεί καλύτερα το προϊόν ως σύνολο. Το Blueprint θα πρέπει να είναι μια σειρά κατηγοριών (τμήματα), όπου αναλύει και περιγράφει τις λειτουργίες, τις ανάγκες, τις άδειες, τις ιδιότητες του προϊόντος και κάθε είδους πληροφορίες που θα πρέπει να αφορούν τον δυνητικό αγοραστή.

Επιπλέον, κάθε ένα από αυτά τα νεοσύστατα τμήματα θα πρέπει να είναι, κατά κάποιον τρόπο, "αυτόνομα" και να είναι σε θέση να σταθούν ως ένα μεμονωμένο μέρος από το Blueprint. Ένα ακόμα σημαντικό στοιχείο της δομής του Blueprint, είναι ότι προσπαθεί να ακολουθήσει την οργανωτική δομή μιας εταιρίας οπότε κάθε ενότητα θα πρέπει να αναφέρεται (συνήθως) σε κάποιον μέσα στην εταιρεία (που είναι-στο τέλος-ο δυνητικός αγοραστής). Για να κατανοηθεί καλύτερα η αρχιτεκτονική του Blueprint, θα μπορούσε να συγκριθεί με ένα δέντρο με φύλλα. Το Blueprint (ως σύνολο) είναι ο κεντρικός κόμβος. Κάθε ενότητα οδηγεί σε έναν άλλο κόμβο, ο οποίος στην πραγματικότητα είναι μία ενότητα ως μεμονωμένη μονάδα. Στο εσωτερικό του κόμβου κάθε ενότητας, μπορεί κανείς να βρει όλες τις απαραίτητες πληροφορίες σχετικά με αυτό. Με λίγα λόγια, όποιος επιθυμεί να λάβει πληροφορίες σχετικά με ένα τμήμα του Blueprint, θα διαβάσει μόνο έναν κόμβο από το Blueprint. Φυσικά, όλα τα τμήματα μπορούν να βρεθούν στον κεντρικό κόμβο, το οποίο είναι το πλήρες Blueprint. Στην περίπτωσή μας (δηλαδή, η γενική προσέγγιση), το Blueprint χωρίζεται σε τέσσερις ενότητες, πράγμα που σημαίνει ότι είναι δέντρο κόμβων και αποτελείται από τέσσερις κόμβους (εκτός από τον κύριο κόμβο).

Το πρώτο τμήμα του Blueprint περιέχει γενικές πληροφορίες σχετικά με το σύστημα σύνθεσης των υπηρεσιών, όπως μια επιχειρηματική περιγραφή της αίτησης, τη νομοθετική συμμόρφωση, τις άδειες, την κατηγορία και μια αφηρημένη αναφορά στις εισροές και τις εκροές της. Θα πρέπει να εισαγάγει συνοπτικά στον αναγνώστη και τον δυνητικό αγοραστή του "προϊόντος" (η υπηρεσία που αποτελείται από λογισμικό), όλα τα

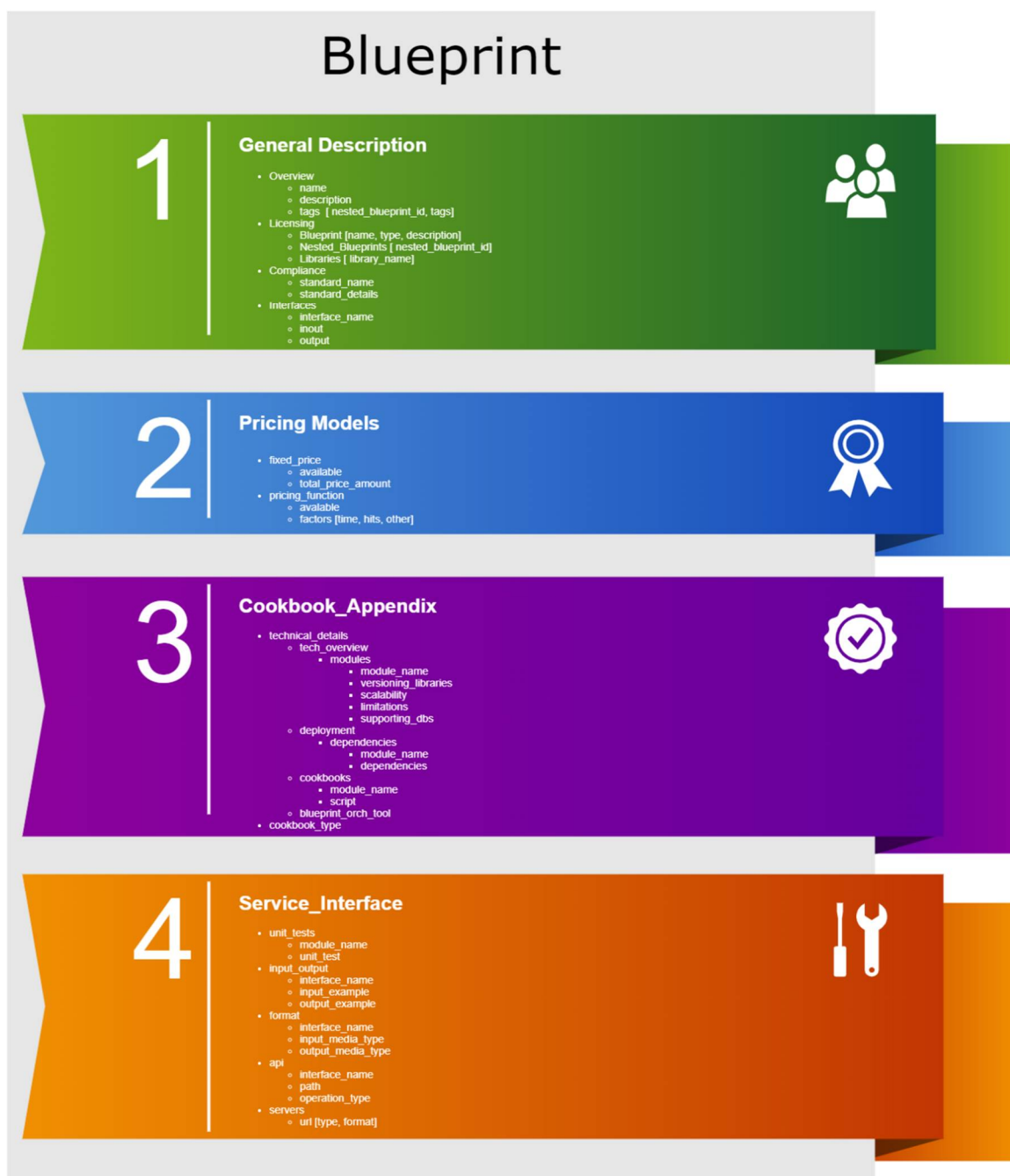
βασικά στοιχεία που το περιγράφουν. Αυτό το τμήμα δεν θα πρέπει να περιγράφει λεπτομερώς τους τομείς των μικροϋπηρεσιών. Ο κύριος στόχος του πρώτου τμήματος είναι να παράσχει όλες τις πληροφορίες που απαιτούνται για τον υπεύθυνο λήψης αποφάσεων (όπως τα ανώτερα στελέχη της εταιρείας) προκειμένου να εξεταστεί το ενδεχόμενο απόκτησης του προϊόντος.

Το δεύτερο τμήμα του Blueprint περιγράφει την τιμολόγηση του προϊόντος στο σύνολό του καθώς και όλα τα οικονομικά θέματα που το αφορούν. Το προϊόν αποτελείται από μια σειρά από διαφορετικές μικροϋπηρεσίες, σε συνδυασμό με ένα (ή περισσότερα) API. Μια τέτοια αρχιτεκτονική καθιστά υποχρεωτική την κατασκευή ενός συγκεκριμένου μοντέλου τιμολόγησης. Αν και ένα μοντέλο θα μπορούσε να είναι αρκετό για να εξηγήσει τις εξαρτήσεις τιμολόγησης στον υπεύθυνο για τις αγορές στην εταιρεία, φαίνεται καλύτερη ιδέα να εφαρμοστούν δύο είδη καθεστώτων τιμολόγησης, προκειμένου να δοθεί καλύτερη εικόνα για το συνολικό κόστος που απαιτείται για την απόκτηση του προϊόντος.

Η τρίτη ενότητα του Blueprint περιλαμβάνει μια αφηρημένη τεχνική επισκόπηση σχετικά με την τήρηση ιστορικού εκδόσεων, τις βιβλιοθήκες, τη δυνατότητα κλιμάκωσης, τους περιορισμούς και τις βάσεις δεδομένων υποστήριξης για κάθε μικροϋπηρεσία. Επιπλέον, αναλύει την ακολουθία πλήρους ανάπτυξης του προϊόντος, όπως οι εξαρτήσεις μεταξύ των μικροϋπηρεσιών και, φυσικά, την ίδια την ακολουθία. Επιπλέον, αυτή η ενότητα μπορεί να απαριθμεί ένα είτε περισσότερα cookbooks για κάθε μικροϋπηρεσία. Ακόμα αναλύει όλα τα στοιχεία ενορχήστρωσης του Blueprint. Με αυτόν τον τρόπο, ένας

πιθανός αγοραστής θα λάβει την «πλήρη εικόνα» των τεχνικών απαιτήσεων της μικροϋπηρεσίας.

Η τέταρτη ενότητα του Blueprint αναφέρεται στους προγραμματιστές μιας εταιρείας. Αυτοί που είναι υπεύθυνοι για τη σωστή λειτουργία όλων των λογισμικών μέσα σε μια εταιρεία είναι οι προγραμματιστές. Εξασφαλίζουν ότι όλα λειτουργούν σύμφωνα με το σχέδιο και διασφαλίζουν τη σταθερότητα για κάθε είδος προϊόντος που εκτελείται στα υπολογιστικά συστήματα. Αυτό σημαίνει ότι η ομάδα θα πρέπει να είναι σε κατάσταση εγρήγορσης, ώστε το λογισμικό της εταιρείας να λειτουργεί πάντα σωστά. Στην περίπτωση του προϊόντος των μικροϋπηρεσιών, η ομάδα προγραμματιστών θα πρέπει να κατανοήσει βαθιά πώς λειτουργεί. Η ενότητα αυτή εξηγεί όλα τα συστατικά στοιχεία του προϊόντος για να δώσει στους προγραμματιστές μια πλήρη επισκόπηση.



Εικόνα 29 Πρότυπο Blueprint

Το Blueprint, αποτελείται από ένα αρχείο JSON που δομείται από το δημιουργό του προϊόντος υπηρεσίας. Η δομή βασισμένη σε JSON είναι σημαντική γιατί είναι εύκολο στην κατανόηση και την σειριοποίησή του. Επιπλέον, τα αρχεία JSON είναι τα πρότυπα για τις αλληλο-επικοινωνίες μεταξύ APIs οπότε είναι επακόλουθο να δομηθεί με τέτοιο τρόπο το πλήρες έγγραφο. Για παράδειγμα, ένας αντιπρόσωπος της εταιρείας θα μπορεί να πάρει τα αρχεία JSON και να συγκρίνει εύκολα τα πεδία που βρίσκει πιο σημαντικά. Στο τέλος, θα μπορούν να επιλέξουν το Blueprint που ταιριάζει καλύτερα στις ανάγκες της εταιρείας, συγκρίνοντας μόνο τα κατάλληλα πεδία που τους ενδιαφέρουν από το αρχείο. Η ύπαρξη λοιπόν ενός αρχείου JSON θα κάνει τη διαδικασία της (κατάλληλης) επιλογής microservice ευκολότερη για τους πιθανούς αγοραστές.

5.5.2.1 Πρώτο τμήμα

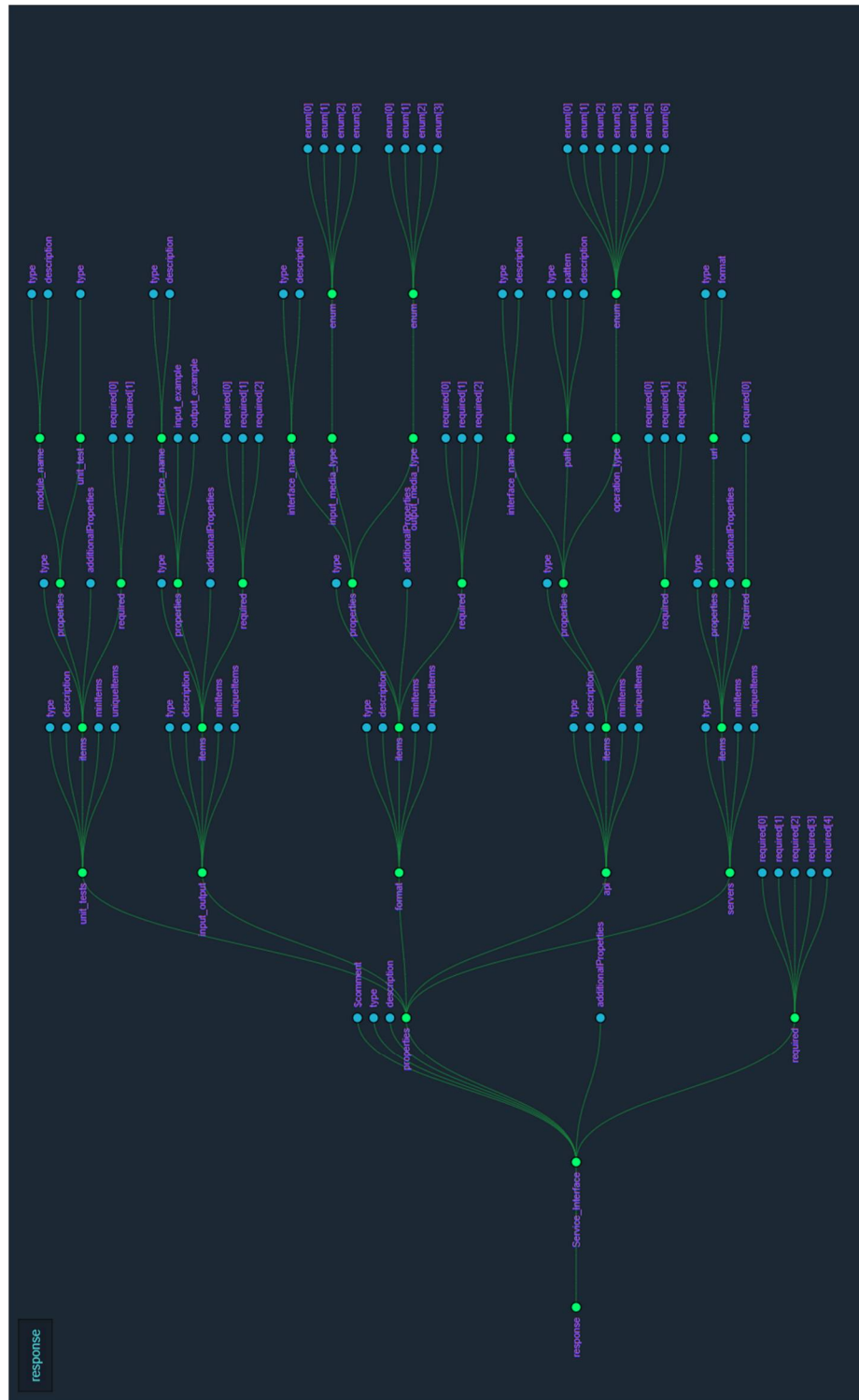
Το πρώτο τμήμα του Blueprint είναι κυρίως μια γενική περιγραφή της υπηρεσίας. Δηλαδή, αναλύει την πραγματική φύση του συστήματος και έχει σα στόχο να δώσει γενικές πληροφορίες για το στόχο και τη λογική αυτού. Απευθύνεται συνήθως στον υπεύθυνο πολιτικής και οργάνωσης της εταιρείας όπου μπορεί να δει συνοπτικά και χωρίς πολλές τεχνικές λεπτομέρειες τη περιγραφή της. Θα μπορούσε κανείς να πει ότι η πρώτη ενότητα είναι παρόμοια με μια επιχειρηματική περιγραφή της εφαρμογής. Αυτή χωρίζεται σε μια σειρά από μέρη, προκειμένου να βοηθήσει τον υπεύθυνο να κατανοήσει το προϊόν. Πιο συγκεκριμένα, αυτά τα μέρη είναι η κύρια και γενική περιγραφή των μικροϋπηρεσιών που αποτελούν το προϊόν, η αδειοδότηση, η συμμόρφωση, η είσοδος/έξοδος που έχει το προϊόν.

Section I General Description	
Overview	Αποτελείται από απλό κείμενο που περιλαμβάνει λεπτομέρειες σχετικά με τη λειτουργικότητα, την περιγραφή και τη λογική της υπηρεσίας.
Licensing	Πληροφορίες σχετικά με την αδειοδότηση της υπηρεσίας. Αυτό θα μπορούσε να ακολουθήσει τις κοινές άδειες που υπάρχουν διαθέσιμες (όπως LGPL, GPL, MIT, BSD, Boost, CC-BY) ή να χρησιμοποιήσει ιδιόκτητη άδεια που θα μπορούσε να έχει μια εταιρεία.
Compliance	Αποτελεί μια περιγραφή σε επίπεδο συμμόρφωσης με πρότυπα και διεθνείς πολιτικές και πρακτικές.
Interfaces	Λεπτομερείς πληροφορίες σχετικά με τα δεδομένα εισόδου και εξόδου της υπηρεσίας. Αυτά μπορεί να είναι παραπάνω από ένα καθώς μπορεί να προσφέρει με διαφορετικό τρόπο τα δεδομένα ανάλογα με τον τελικό αποδέκτη της υπηρεσίας.

Πίνακας 4 Πρώτο τμήμα του **Blueprint**

Στο συγκεκριμένο τμήμα θα πρέπει σύμφωνα με τις πρακτικές καλής λειτουργίας των μικροϋπηρεσιών που έχουν περιγράψει στο “Threat Model for microservices in containerized environments” και το “Best Practices in Implementing a Secure

Microservices Architecture” από την CSA, να περιέχει μέσα στα τμήματα αυτά και ένα μοναδικό αναγνωριστικό, το οποίο διαφοροποιεί το προϊόν. Ακόμα σημαντικό είναι να περιλαμβάνει ένα κείμενο που θα έχει λεπτομέρειες σχετικά με τη λειτουργικότητα, την περιγραφή και τη λογική της υπηρεσίας. Σε αυτό, η ομάδα πίσω από το προϊόν θα πρέπει να παρέχει γενικές πληροφορίες και μια περιγραφή σχετικά με τις υπηρεσίες και τα δεδομένα. Για παράδειγμα, θα μπορούσαν να δώσουν μια απάντηση για καθεμία από τις ακόλουθες ερωτήσεις: ποια δεδομένα εμπλέκονται στην εφαρμογή; Ποια είναι η αξία των δεδομένων; Ποιες είναι οι υπηρεσίες που έχουν πρόσβαση στα δεδομένα; Αναφορικά με το λειτουργικό σύστημα θα πρέπει να περιγράφονται λεπτομέρειες του, όπου θα πρέπει να αναφέρεται τόσο το host λειτουργικό καθώς και η διανομή του λειτουργικού συστήματος. Επακόλουθο τόσο των λειτουργικών όσο και του ίδιου του κώδικα είναι η αδειοδότηση, που θα πρέπει να εξηγεί συνοπτικά τις λεπτομέρειες σχετικά με τη χορήγηση αδειών του προϊόντος. Αυτές οι λεπτομέρειες ενδέχεται να ποικίλλουν ανάλογα με τις μικροϋπηρεσίες που περιλαμβάνει.



Εικόνα 30 Δομή JSON σχήματος του πρώτου τμήματος

Πέρα από τη ολοκληρωμένη περιγραφή και το τρόπο και το θέμα που απευθύνεται η υπηρεσία σημαντικό είναι να περιγράφουν χαρακτηριστικά που έχουν να κάνουν με το κύκλο ζωής του και τη στρατηγική της αναβάθμισης, που καθορίζει το σχέδιο αναβάθμισης του λειτουργικού συστήματος τόσο για παραδοσιακές ενημερώσεις, όσο και για τις ενημερώσεις ασφαλείας και τις πολιτικές για backports. Σύμφωνα με τις απαιτήσεις προστασίας του λειτουργικού συστήματος του Host μηχανήματος που αναφέρονται στο «Threat Model for microservices in containerized environments» της CSA, ο πωλητής του προϊόντος θα πρέπει να αναφέρει τόσο τον κεντρικό υπολογιστή όσο και τη διανομή του λειτουργικού συστήματος.

Επιπλέον, θα πρέπει να συμπεριληφθεί μια μικρή παράγραφος, η οποία θα αναλύσει τη στρατηγική επικαιροποίησης του λειτουργικού συστήματος. Ακόμα, η ομάδα πίσω από το προϊόν θα πρέπει να παρέχει γενικές πληροφορίες και μια περιγραφή σχετικά με τις υπηρεσίες και τα δεδομένα. Για παράδειγμα, θα μπορούσαν να δοθούν λεπτομέρειες για καθεμία από τις ακόλουθες ερωτήσεις: ποια δεδομένα εμπλέκονται στην εφαρμογή; Ποια είναι η αξία των δεδομένων; Ποιες είναι οι υπηρεσίες που έχουν πρόσβαση στα δεδομένα; Τέλος ακολουθώντας τις προτάσεις του «Authorization and Access Control on Microservices» της CSA η εξουσιοδότηση και η πιστοποίηση στην αρχιτεκτονική των μικροϋπηρεσιών θα πρέπει να καθοριστούν σε αυτό το κομμάτι του Blueprint.

Ιδίως μετά την δημιουργία και δοκιμή, οι ομάδες ασφαλείας θα πρέπει να πραγματοποιούν πιστοποίηση για κάθε μία από τις υπηρεσίες που βρίσκονται μέσα στο container/VM. Αυτό μπορεί να γίνει διασφαλίζοντας ότι έχουν εφαρμοστεί όλες οι

απαραίτητες ενέργειες για μια σωστή εγκατάσταση των κομματιών. Με αυτόν τον τρόπο ο αγοραστής θα έχει μια σαφή εικόνα του προϊόντος ως πιστοποιημένο πακέτο μικροϋπηρεσιών. Επιπλέον, τα container πρέπει να αποθηκεύονται σε κεντρικά αποθετήρια όπου ως αρχεία, θα πρέπει να περιλαμβάνουν την ημερομηνία δημιουργίας τους, όσο και συγκεκριμένες ετικέτες, πληροφορίες που θα βοηθήσει τη διαδικασία της κατηγοριοποίησης, τον εντοπισμό, την ανάκτηση και την επαναχρησιμοποίηση τους όταν αυτό χρειάζεται. Τέλος θα πρέπει να δηλώνεται το όνομα και το πιστοποιητικό κάθε μητρώου, προκειμένου να κατανοηθεί αν το μητρώο των container είναι αρκετά αξιόπιστη ή όχι πηγή.

5.5.2.2 Δεύτερο τμήμα

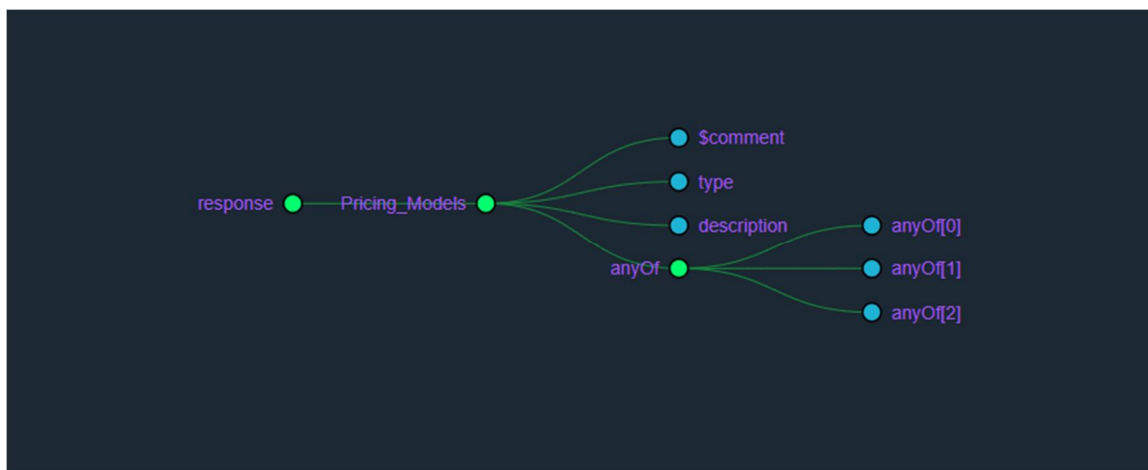
Το δεύτερο τμήμα του Blueprint θα πρέπει να παρέχει όλες τις πληροφορίες που απαιτούνται για την εκτελεστική εξουσία μιας εταιρείας, η οποία έχει την πρωταρχική ευθύνη της διαχείρισης των οικονομικών της και όλων των θεμάτων που έχουν να κάνουν με τις οικονομικές κατευθύνσεις της εταιρίας.

Section II Pricing Models	
Pricing as a Service (pricing_function)	Το προϊόν θα πρέπει να περιγράφεται ως μια υπηρεσία, που αποτελείται από πολλά και διαφορετικά συστατικά.

Pricing as a Structure (fixed_price)	Το προϊόν θα πρέπει να αναλυθεί ως ολοκληρωμένη δομή εξαρτημάτων.

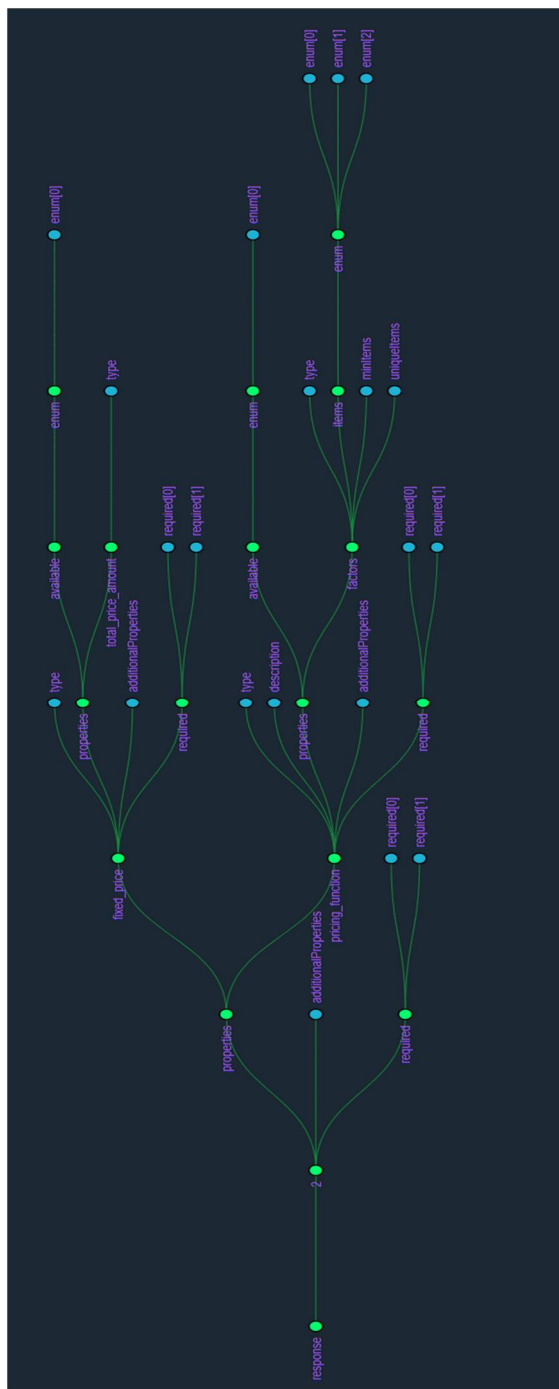
Πίνακας 5 Δεύτερο τμήμα του Blueprint

Το προϊόν αποτελείται από μια σειρά από διαφορετικές μικροϋπηρεσίες, σε συνδυασμό με ένα (ή περισσότερα) API. Μια τέτοια αρχιτεκτονική καθιστά υποχρεωτική την κατασκευή ενός συγκεκριμένου μοντέλου τιμολόγησης. Αν και ένα μοντέλο θα μπορούσε να είναι αρκετό για να εξηγήσει τις εξαρτήσεις τιμολόγησης στον οικονομικό αναλυτή μιας εταιρείας, φαίνεται καλύτερη ιδέα να εφαρμοστούν δύο είδη καθεστώτων τιμολόγησης προκειμένου να δοθεί καλύτερη εικόνα για το συνολικό κόστος που απαιτείται για την απόκτηση του προϊόντος.



Εικόνα 31 Γενική δομή JSON σχήματος του δεύτερου τμήματος

Το πρώτο μοντέλο τιμολόγησης περιγράφει το προϊόν ως υπηρεσία, αποτελούμενο από πολλά και διαφορετικά συστατικά. Υπάρχει μεγάλη πιθανότητα ο τελικός χρήστης (πιθανότατα μια εταιρεία) να χρειαστεί λιγότερες από όλες τις διαθέσιμες υπηρεσίες που παρέχονται από το προϊόν. Ως εκ τούτου, αυτό το μοντέλο τιμολόγησης θα πρέπει να περιγράφει όλες τις διαθέσιμες υπηρεσίες και να αναφέρει το κόστος που απαιτείται για την εφαρμογή κάθε μίας από αυτές. Κάθε αρχική υπηρεσία που ορίζεται στο Blueprint θα πρέπει να τιμολογείται, ώστε ο οικονομικός αναλυτής της εταιρείας να αποφασίσει ποια από τις διαθέσιμες υπηρεσίες είναι συμφέρουσα. Ένα τέτοιο μοντέλο τιμολόγησης θα έδινε στους δυνητικούς αγοραστές τη δυνατότητα να επιλέξουν προσεκτικά τις μικροϋπηρεσίες που θεωρούν απαραίτητες και να καθορίσουν από μόνοι τους την τελική τιμή του προϊόντος.



Εικόνα 32 Εσωτερική δομή του σχήματος σχετικά με τα μοντέλα κόστους

Εκτός από τις εταιρείες, οι επιχειρήσεις μικρότερης κλίμακας προσελκύονται από το προϊόν, δεδομένου ότι θα είναι σε θέση να μειώσουν το κόστος αποκλείοντας όλες τις υπηρεσίες που είναι περιττές για τον σκοπό τους.

Το δεύτερο μοντέλο τιμολόγησης απευθύνεται σε προϊόντα που παρουσιάζονται ως πλήρης δομή κατασκευαστικού στοιχείου. Κάθε μικροϋπηρεσία θα πρέπει να περιγράφεται λεπτομερώς, μαζί με το κόστος εφαρμογής της. Με αυτόν τον τρόπο, ένας δυνητικός αγοραστής θα έχει τη δυνατότητα να μειώσει το τελικό κόστος του προϊόντος προσθέτοντας ή αφαιρώντας υπηρεσίες. Το τρέχον μοντέλο όμως προσεγγίζει ένα προϊόν ως μια τελική δομή. Όλα τα ενσωματωμένα στοιχεία θα εξακολουθήσουν να αναλύονται, αλλά ο δυνητικός αγοραστής δεν θα έχει τη δυνατότητα να κάνει αλλαγές στην αρχιτεκτονική του προϊόντος. Ωστόσο, αυτό το μοντέλο τιμολόγησης δεν θα περιλαμβάνει μόνο μία τελική τιμή, αλλά μια σειρά από διαφορετικές, ανάλογα με τις πιθανές κατασκευές του προϊόντος. Οπότε το μοντέλο θα περιγράψει μια σειρά προκαθορισμένων υλοποιήσεων προϊόντων που θα ποικίλλουν για να ικανοποιούν διαφορετικές ανάγκες, προϋπολογισμούς και απαιτήσεις. Οι δυνητικοί αγοραστές θα έχουν τη δυνατότητα να επιλέξουν τη δομή του προϊόντος που είναι κατάλληλη για τις ανάγκες τους, αλλά δεν θα είναι σε θέση να επιλέξουν τις υπηρεσίες μία προς μία.

Κάθε ενδιαφερόμενος θα επιλέξει το μοντέλο τιμολόγησης που θεωρεί ότι θα είναι πιο κατάλληλο για το προϊόν του και την αγορά στόχευσης. Η προσέγγιση του ζητήματος της τιμολόγησης με δύο διαφορετικούς τρόπους δίνει στον κάτοχο του Blueprint (και του προϊόντος) τη δυνατότητα να πουλήσει τη δημιουργία του με τον τρόπο που θέλει.

5.5.2.3 Τρίτο τμήμα

Το τρίτο σκέλος του Blueprint αναφέρεται στη διαδικασία εγκατάστασης της υπηρεσίας σαν λογισμικό και προσφέρει πληροφορίες κυρίως σε αυτούς που καλούνται να το διαχειριστούν. Οπότε η κύρια εστίαση είναι σε τεχνολογικά ζητήματα εντός της υπηρεσίας.

Section III Overview	
Technological details	<ul style="list-style-type: none"> ● Module name ● Versioning_Libraries ● Scalability ● Limitations ● Supporting Databases
Deployment	<ul style="list-style-type: none"> ● Dependencies ● Sequence
Cookbooks	<ul style="list-style-type: none"> ● Module name ● Script
Blueprint Orchestrating Tools	<ul style="list-style-type: none"> ● Vendor Name
Cookbook Type	<ul style="list-style-type: none"> ● Name

Πίνακας 6 Τρίτο τμήμα του Blueprint

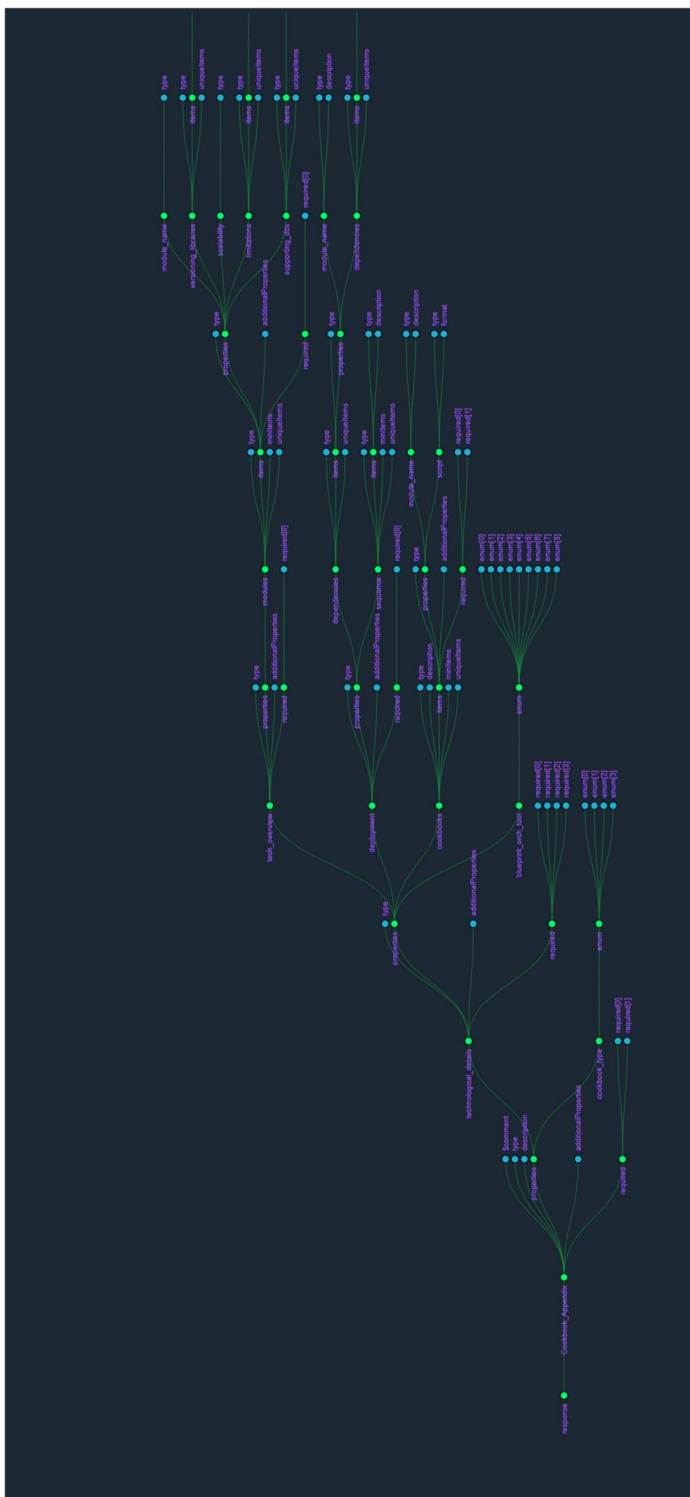
Τα στοιχεία που θα μπορούσαν να συμπεριληφθούν σε αυτήν την ενότητα πρέπει να αποσαφηνίζουν πώς μπορεί να εγκατασταθεί αυτή η υπηρεσία, τι απαιτούμενα έχει και εξαρτήσεις καθώς και το εργαλείο που θα καλεστεί να κάνει αυτή τη διαχείριση. Συγκεκριμένα περιλαμβάνει τη *τεχνική επισκόπηση*, όπου είναι ένας πίνακας που περιέχει τεχνικές λεπτομέρειες για κάθε μικροϋπηρεσία. Σε αυτό το κομμάτι συμπεριλαμβάνονται για κάθε στοιχείο της υπηρεσίας το όνομα του, οι εκδόσεις από τις βιβλιοθήκες που χρειάζεται, η δυνατότητα κλιμάκωσης, οι περιορισμοί και οι βάσεις δεδομένων είτε η ουρά υποστήριξης κάθε μονάδας μικροεξυπηρέτησης. Με λίγα λόγια, η τεχνική επισκόπηση θα είναι μια συστοιχία που περιλαμβάνει ενότητες όπου κάθε ενότητα θα περιλαμβάνει τις προαναφερθείσες τιμές. Όσον αφορά την *ακολουθία ανάπτυξης*, είναι ένα αντικείμενο που καθορίζει τις λεπτομέρειες σχετικά με την ανάπτυξη του προϊόντος. Πρέπει να περιέχει τις εξαρτήσεις μεταξύ μικροϋπηρεσιών και την ίδια την ακολουθία για την ανάπτυξη του. Η πρώτη τιμή, οι εξαρτήσεις μεταξύ μικροϋπηρεσιών, είναι ένα πεδίο που θα αναφέρει εν συντομία όλες τις εξαρτήσεις μεταξύ των ενοτήτων του προϊόντος. Η δεύτερη τιμή, η ακολουθία ανάπτυξης, αναλύει την ακολουθία ανάπτυξης του προϊόντος. Τρίτο στοιχείο είναι το *Cookbook* (appendix), που περιέχει δύο τιμές, τα όνομα του στοιχείου καθώς και λεπτομέρειες εγκατάστασης αυτού. Το πεδίο *module name* καθορίζει τα ακριβές - μοναδικό- όνομα που περιγράφει το στοιχείο. Η δομή του είναι με τέτοια μορφή που θα μπορούσε κάλλιστα να συμπεριλάβει ακόμα και ένα ολόκληρο blueprint. Το πεδίο *script* περιλαμβάνει όλη τη διαδικασία και τις τεχνικές λεπτομέρειες σε γλώσσα που μπορεί να γίνει αντιληπτή από τις πλατφόρμες διαχείρισης υπηρεσιών και containers. Τέτοια γλώσσα είναι για παράδειγμα η YAML που μπορεί να χρησιμοποιηθεί από εργαλεία όπως το

puppet, το ansible και το chef για να περιγράψει το component που με τη σειρά του θα δοθεί προς διαχείριση στη πλατφόρμα. Σημαντικό δομικό στοιχείο είναι η εργαλειοθήκη ενορχήστρωσης blueprint, όπου αναλύει συνοπτικά τα εργαλεία ανάπτυξης και ενορχηστρώνει τις μικροϋπηρεσίες από το προϊόν ώστε να ακολουθεί συγκεκριμένη διαδρομή για την εκτέλεση της υπηρεσίας. Τέλος για να αναγνωστούν ορθά τα scripts υπάρχει μια επιπλέον εγγραφή που είναι αυτή που δηλώνει το τύπο της πλατφόρμας αλλά και της γλώσσας που έχουν γραφτεί αυτά. Τέτοια είναι το Chef, το Puppet και η Ansible. Βέβαια το σχήμα είναι ανεξάρτητο και δεν περιορίζει με πιο εργαλείο θα γραφτούν γι' αυτό και μπορεί ο δημιουργός να ορίσει το δικό του εργαλείο δημιουργίας των cookbooks.

Το κομμάτι αυτό του blueprint θα πρέπει να αναλύσει τα τεχνικά οφέλη του προϊόντος. Τα οφέλη αυτά είναι (μεταξύ άλλων) η προσέγγιση Polyglot, η οποία επιτρέπει τη συλλογή και την επιλογή της καλύτερης τεχνολογίας, εργαλείων, πλατφόρμων και το γεγονός ότι η αρχιτεκτονική μικροϋπηρεσιών επιτρέπει τη συνεχή παράδοση εκδόσεων (continues delivery). Επιπλέον, ένα ακόμη όφελος είναι ότι κάθε υπηρεσία σε ένα προϊόν μπορεί να αναπτυχθεί ανεξάρτητα. Αυτό σημαίνει ότι το ίδιο μπορεί να ισχύσει και με την εγγραφή του κώδικα μιας υπηρεσίας. Αυτή η δυναμική του κατακερματισμού, διευκολύνει γιατί μπορεί το κάθε στοιχείο να χρησιμοποιηθεί σε άλλα έργα όπου απαιτείται η ίδια λειτουργικότητα.

Όσον αφορά στον πωλητή και το κομμάτι της ασφάλειας του προϊόντος θα πρέπει να συμπεριλάβει λεπτομέρειες για να ενεργοποιήσει ή να απενεργοποιήσει το vTPM,

χαρακτηριστικό που βρίσκεται στη λίστα των προτάσεων για περιγραφή από το Best Practices in Implementing a Secure Microservices Architecture της CSA.



Εικόνα 33 Δομή JSON σχήματος του τρίτου τμήματος

Επιπλέον, ο πυρήνας θα πρέπει να δηλώνεται, να αναλύεται και να περιγράφεται, μαζί με το σχέδιο ενημέρωσης/αναβάθμισης. Στόχος είναι, ο πωλητής να αναφέρει τόσο την φιλοξενία όσο και τη διανομή του λειτουργικού συστήματος (αναφέρεται επίσης στην πρώτη ενότητα του blueprint). Με αυτόν τον τρόπο, ο αντιπρόσωπος της εταιρείας που θα το δει θα γνωρίζει αν αυτή η επιλογή λειτουργικού συστήματος ταιριάζει με τις ανάγκες και τα πρότυπα που ακολουθεί.

Ένα ακόμα σημαντικό στοιχείο είναι η συχνότητα και το χρονοδιάγραμμα στο οποίο θα πραγματοποιηθούν οι μελλοντικές επικαιροποιήσεις. Παράλληλα μια ακόμη πληροφορία που αξίζει να συμπεριληφθεί στην τρίτη ενότητα του blueprint και συγκεκριμένα στα *limitations* είναι η παράμετρος που καθορίζει αν το SELINUX είναι ενεργοποιημένο ή όχι. Τέλος, κάθε κομμάτι (container) θα πρέπει να δηλώσει αν χρησιμοποιεί τη κοινόχρηστη μνήμη του δίσκου, ώστε ο υπεύθυνος να γνωρίζει το ποσό που θα πρέπει να δεσμεύσει και να ξέρει πώς να το διαχειριστεί.

Σύμφωνα ξανά με τις προτάσεις για τις βέλτιστες πρακτικές ασφάλειας των container σε Μικροϋπηρεσίες από τη CSA προκειμένου να προστατευθεί η ακεραιότητα του συστήματος και του περιβάλλοντος λειτουργίας, το λειτουργικό σύστημα θα πρέπει να ενημερώνεται συνεχώς για θέματα ευπάθειας container. Επιπλέον, θα πρέπει να υπάρχει ένα χαρακτηριστικό που θα αναφέρει τα επίπεδα κρισιμότητας του προϊόντος (και των μικροϋπηρεσιών του). Με αυτόν τον τρόπο θα αποφεύγεται η ανάμειξη εφαρμογών με διαφορετικά επίπεδα εμπιστευτικότητας και κρισιμότητας.

Έχοντας τον ίδιο γνώμονα ένα ακόμα τμήμα αυτού του blueprint θα πρέπει να αναλύσει τα πρότυπα πρωτοκόλλου του προϊόντος. Για παράδειγμα, θα πρέπει να είναι σαφές αν η λειτουργική μονάδα χρησιμοποιεί το SPIFFE ή JSON Web Token ή είτε ακόμη και τα δύο. Αυτά τα χαρακτηριστικά επηρεάζουν και άλλους παράγοντες όπως για παράδειγμα (παράμετροι ασφαλείας) τους χρόνους ανάπτυξης και της ενοποίησης της ροής εργασίας CI-CD.

Όπως περιγράφεται στην αρχή, υπάρχουν σοβαρά μειονεκτήματα όταν πρόκειται για την ανάπτυξη και τη διαχείριση εφαρμογών που βασίζονται στην αρχιτεκτονική MSA, καθώς μπορούν να οικοδομήσουν ένα πολύ πολύπλοκο και πληθωρικό σύστημα. Η αντιμετώπιση των εξαρτήσεων μεταξύ διαφορετικών υπηρεσιών, η διαχείριση της αυξημένου φόρτου που απαιτείται από κάθε μικροϋπηρεσία για την κατασκευή και παράδοση της καθώς και οι υποστηρικτικές ανάγκες όπως οι λειτουργίες επαναφοράς και οι συναλλαγές είναι μερικές από τις προκλήσεις που πρέπει να αντιμετωπιστούν όταν αναπτύσσονται εφαρμογές με γνώμονα τις μικροϋπηρεσίες. Για τον χειρισμό τέτοιων ζητημάτων, το τρίτο τμήμα του blueprint έχει σχεδιαστεί για να συμπεριλαμβάνει όλες τις απαραίτητες πληροφορίες για την ενορχήστρωση και τις διαδικασίες ανάπτυξης κάθε ενότητας της εφαρμογής. Γενικά, απαριθμεί, από τεχνικά πεδία, όλα τα cookbooks και τις απαιτήσεις συστήματος κάθε container/μικροϋπηρεσίας, καθώς και την ακολουθία ανάπτυξης των container που πρέπει να ακολουθείται για την εκτέλεση της υπηρεσίας.

5.5.2.4 Τέταρτο τμήμα

Το τέταρτο και τελευταίο τμήμα του blueprint στοχεύει να βοηθήσει και να δώσει πληροφορίες στους τεχνικούς/προγραμματιστές. Αυτό το κομμάτι αναλύει τις τεχνικές διεπαφές που έχει το blueprint ώστε να μπορεί να αναγνωρίζεται εύκολα εάν είναι τεχνικά συμβατό με αυτό που θα συνδυαστεί.

Section IV Overview	
Module name	Unique ID
Unit test	<ul style="list-style-type: none"> • Module name • Unit tests
Input Output	<ul style="list-style-type: none"> • List of interface descriptions • Input example • Output example
Format	<ul style="list-style-type: none"> • Interface name • Input media type • Output media type

API	<ul style="list-style-type: none"> • Interface name • Path • Operation type
Servers	<ul style="list-style-type: none"> • URL

Πίνακας 7 Τέταρτο τμήμα του Blueprint

Ο τύπος των χαρακτηριστικών που θα μπορούσαν να συμπεριληφθούν στο αρχείο JSON από αυτή την τέταρτη ενότητα είναι το εκάστοτε *όνομα* του στοιχείου που περιέχει την υπηρεσία καθώς και τα *τεστ*, που γίνονται είναι για να βρεθούν τυχόν ελαττώματα στη συμπεριφορά και τη λειτουργικότητα του προϊόντος. Ακόμα υπάρχει το πεδίο που περιγράφει τη σύνταξη της υπηρεσίας μαζί με την *Είσοδο/Εξοδο*, που θα αναφέρει την είσοδο και την έξοδο του προϊόντος για τους προγραμματιστές, ώστε να γνωρίζουν τους ακριβείς τύπους τους. Ταυτόχρονα περιγράφεται το API, που αναφέρει αναλυτικά τις υπηρεσίες και τα πρωτόκολλα του προϊόντος μαζί με τα στοιχεία του *URL/URI* που πρέπει να χρησιμοποιηθεί για να βρεθεί η υπηρεσία αυτή.

Στο τελευταίο τμήμα θα ληφθεί υπόψιν και θα παρουσιαστούν οι τεχνικές προκλήσεις που παρουσιάζονται στις πρακτικές ασφάλειας μικροϋπηρεσιών από τη CSA. Αυτές οι προκλήσεις είναι οι δοκιμές των υπηρεσιών, η επικοινωνία τους (επιλογές επικοινωνίας όπως message passing είτε RPC) και η διαχείριση της/των βάσεων δεδομένων. Όσον αφορά την τελευταία πρόκληση, όταν πρόκειται για την αρχιτεκτονική των μικροϋπηρεσιών, η αναπτυσσόμενη ομάδα μπορεί να αποφασίσει να εφαρμόσει μια τοπική βάση δεδομένων σε μια μικροϋπηρεσία. Ωστόσο, ενδέχεται να χρειαστούν αλλαγές σε άλλες σχετικές βάσεις δεδομένων, κάτι που μπορεί να δημιουργήσει προβλήματα.

Ακόμα η είσοδος/έξοδος κάθε μικροϋπηρεσίας/μικροκοντέινερ/εικονικής μηχανής θα πρέπει να αναλύεται στο τμήμα του blueprint. Εκτός όμως από τα σημεία διεπαφών θα πρέπει να αναφέρονται και να αναλύονται λεπτομέρειες σχετικά με το API. Όσον αφορά, το εκάστοτε κομμάτι της υπηρεσίας θα πρέπει κάθε κοντέινερ να αναφέρει κάθε υπηρεσία που εκθέτει μιας και μια τέτοια δήλωση είναι χρήσιμη, ώστε ο χρήστης να αναγνωρίσει ποιες υπηρεσίες θα αναπτυχθούν μετά την εκτέλεση του blueprint. Με αυτόν τον τρόπο θα είναι σε θέση να υπολογίσει τον κίνδυνο που μπορεί να προκύψει στο μέλλον λόγω αυτών.

5.6 Συνοψη

Στο κεφάλαιο αυτό παρουσιάστηκε μια ενιαία μορφή δομής για την αναπαράσταση και τη περιγραφή μικροϋπηρεσιών. Το πρότυπο -Blueprint- βασίστηκε στα

αποτελέσματα της ανάλυσης των Microservices από την έρευνα +++, καθώς και στα αναγκαία χαρακτηριστικά που παρουσιάστηκαν και αναφέρονται στα +++. Πιο συγκεκριμένα, κάθε τμήμα αναφέρεται σε ένα συγκεκριμένο ρόλο της επιχειρησιακής δομής, ο οποίος θα μπορούσε να κατανοήσει καλύτερα τα περιεχόμενα του εκάστοτε τμήματος. Η απλότητα του είναι κάτι που διευκολύνει το διαχωρισμό σε κομμάτια αλλά και βοηθά στην εύκολη κατανόηση του. Από όλα τα κομμάτια μόνο το τελευταίο έχει τεχνικές λεπτομέρειες που ίσως χρειάζονται προγραμματιστικές γνώσεις για να γίνει αντιληπτό. Όπως αναφέρθηκε προηγουμένως, το τμήμα 1 έχει σκοπό να παρέχει γενικές πληροφορίες σε έναν από τους υπεύθυνους χάραξης στρατηγικής της εταιρείας, όπου συνήθως είναι αυτός που είναι υπεύθυνος για το όραμα της. Η ενότητα 2 παρέχει όλες τις πληροφορίες που απαιτούνται για τη χάραξη της οικονομικής πολιτικής που θα ακολουθηθεί για αυτήν την υπηρεσία. Το τμήμα 3 έχει ως στόχο να γίνει αντιληπτό ο τρόπος, η μέθοδος και οι τεχνικές απαιτήσεις για την εγκατάσταση η χρήση της υπηρεσίας. Τελευταίο αλλά εξίσου σημαντικό, το Τμήμα 4 είναι γραμμένο έχοντας υπόψη τους προγραμματιστές και ουσιαστικά είναι η εικόνα της υπηρεσίας προς τα έξω. Περιέχει όλες τις ζωτικές πληροφορίες σχετικά με το προϊόν και τις διεπαφές του. Συνοψίζοντας το blueprint παρέχει όλες τις ζωτικές πληροφορίες (σχετικά με τη μικροϋπηρεσία) στον δυνητικό αγοραστή/χρήστη ώστε να έχει μια πλήρη εικόνα για αυτό. Ξεπερνώντας το κλασικό μοντέλο που περιγράφει μόνο τις τεχνικές πληροφορίες ο νέος τρόπος προγραμματισμού ακολουθώντας τη λογική των μικροϋπηρεσιών απαιτεί να λογίζονται και άλλοι παράγοντες όπως το κόστος της υπηρεσίας, ο τρόπος εκτέλεσης της και οι

μέθοδοι χρήσης της. Για αυτό το λόγο ο παραδοσιακός τρόπος περιγραφής δεν είναι αρκετός.

ΚΕΦΑΛΑΙΟ 6

Επίλογος

Η όλο και αυξανόμενη εμφάνιση των πολλών δεδομένων σαν πρόβλημα στις δικτυακές κυρίως- εφαρμογές μαζί με τη σχεδόν ταυτόχρονα συνειδητοποίηση της σημαντικότητας τους, δημιούργησε ένα κενό στο χώρο της πληροφορικής. Η αδυναμία αυτή ήταν πολυδιάστατη μιας και ο τομέας ήταν απροετοίμαστη να το διαχειριστεί τόσο μεθοδολογικά όσο και συστημικά. Το θέμα βέβαια δεν περιορίζεται μόνο στην επεξεργασία αλλά και στην επιλογή και αποθήκευση των δεδομένων αφού ούτε εκεί δεν υπήρχαν οι κατάλληλες υποδομές. Οπότε τόσο σε τεχνικό όσο και σε επίπεδο φιλοσοφίας οι αλλαγές που έπρεπε και πρέπει να γίνουν θα έπρεπε να είναι ριζοσπαστικές.

Αναγνωρίζοντας τα σημεία που πρέπει να αναδειχτούν και να επιλυθούν μπορεί να ορίσει κανείς ότι σα πρώτο βήμα πρέπει να ξεκινήσει από τη ρίζα του προβλήματος. Αυτό το σημείο δεν είναι άλλο από τα ίδια τα δεδομένα και πιο συγκεκριμένα από τις πηγές όπου αντλούνται ώστε να επεξεργαστούν και να μετατραπούν σε πληροφορία. Η διαχείριση και η επεξεργασία των πηγών θα πρέπει να προσφέρει υπηρεσίες εύκολης διαχείρισης και προβολής τους, συστήματα ελέγχου του φόρτου και της ροής των δεδομένων που

πλημυρίζουν την εφαρμογή καθώς και συστήματα που αναλαμβάνουν να διαχωρίσουν και να προ-επεξεργαστούν τα δεδομένα ώστε να φιλτραριστούν και να φτάσουν στην εφαρμογή μόνο αυτά που πραγματικά είναι αναγκαία. Ταυτόχρονα πρέπει να ελεγχθούν για την ορθότητα και τη πιστότητα τους ώστε να μην επηρεάζουν την αποδοτικότητα της εφαρμογής. Παραδείγματα τόσο από μη σωστά δεδομένα που μπορεί να δωθούν για εκπαίδευση [145] όσο και από πιο πολλά και υπερβολικά δεδομένα που πλημμύρισαν το σύστημα [146] [147] [148] [149] [150] [151] είναι μερικές μόνο από τις περιπτώσεις που δείχνουν την ανάγκη παρουσίασης μιας πλήρους μεθοδολογίας που να διαχειρίζεται από τη πηγή μέχρι τη διάδοση τα δεδομένα και τις πληροφορίες. Στη μεθοδολογία που παρουσιάστηκε υπάρχει μια αυστηρή διαδικασία στη ροή των δεδομένων που προσφέρει καθώς και μια ευελιξία στη χρήση των τελικών δεδομένων. Η χρήση οπότε των δεδομένων και των ενεργειών που πρέπει να γίνουν σε αυτά είναι άμεσα συνυφασμένη με το ακριβές σημείο που πρέπει να πραγματοποιηθεί αυτή η ενέργεια, βάση του μοντέλου της μεθοδολογίας που παρουσιάστηκε.

Παρόλη την αυστηρή κατευθυντήρια γραμμή που πρέπει να ακολουθηθεί όσο αναφορά τα δεδομένα, ένα ακόμα σημείο όπου πρέπει να τονιστεί και να αναδειχθεί είναι το κομμάτι των πρότερων ενεργειών και του φιλτραρίσματος που μπορεί να γίνει πριν φτάσουν τα δεδομένα (κυρίως σε περιπτώσεις ροών δεδομένων) στον αλγόριθμο επεξεργασίας. Σε αυτό το στάδιο, βασικό ρόλο παίζει το Complex Event Processing που αναλαμβάνει να διαχωρίσει σε κατηγορίες, να φιλτράρει και κυρίως να δημιουργήσει το πλάνο εκτέλεσης και επεξεργαστεί αυτών. Ένας ακόμα στόχος είναι σε ένα καταναμημένο

περιβάλλον να μετατρέψει και να μορφοποιήσει τις μεθόδους με τέτοιο τρόπο ώστε να μπορούν να επεξεργαστεί και να εκμεταλλευτεί τους διαθέσιμους πόρους στο μέγιστο τους.

Το επόμενο δομικό βήμα είναι η αρχιτεκτονική της ίδιας της εφαρμογής όπου θα πρέπει και αυτή να ακολουθεί τις αρχές και τη λογική που εισήγαγαν τα πολλά δεδομένα. Λόγο τις ανομοιογένειας αλλά και τις έλλειψης γνώσης εκ των προτέρων για το που μπορεί να δημιουργηθεί κώλυμα λόγω φόρτου η αρχιτεκτονική που ταιριάζει σε αυτές τις απαιτήσεις είναι αυτή των μικροπηρεσιών. Η σπονδυλωτή αρχιτεκτονική μπορεί «λογικά» να διαχωρίσει κάθε ενέργεια και να την αναθέσει σε διαφορετική υπηρεσία ώστε να προσφέρει τη μέγιστη αξιοπιστία στο σύστημα. Δημιουργώντας τα υποσύστημα που αποτελούν την αρχιτεκτονική σαν ανεξάρτητες οντότητες, που μοναδικό περιορισμό έχουν το πρότυπο(format) για την είσοδο και έξοδο τους, καταφέρνει να μπορεί να προσφερθεί το κάθε κομμάτι 'και' σαν ανεξάρτητο λογισμικό για τη συγκεκριμένη λειτουργία που δημιουργήθηκε. Ταυτόχρονα προσφέρει μια τρομερή ευελιξία για τον προγραμματιστή αυτή η σπονδύλωση μιας και επιτρέπει να δοκιμάζονται οι υπηρεσίες με τις μέθοδος τους στοχεύοντας και αλλάζοντας μόνο το υποσύστημα που αυτοί ενδιαφέρονται να υλοποιήσουν ή να τροποποιήσουν. Δοκιμάστηκε, τροποποιήθηκε και υλοποιήθηκε η μετατροπή μιας παραδοσιακής εφαρμογής αναγνώρισης συναισθημάτων στη νέα φιλοσοφία ακολουθώντας τη λογική που εισάγουν τα πολλά δεδομένα. Σε αυτό τον αλγόριθμο, που ουσιαστικά μετατράπηκε σε σύστημα διαχείρισης μικροπηρεσιών, δοκιμάστηκε η απόδοση και η πιστότητα του τόσο με βάσεις δεδομένων αναφοράς που

αφορούν κριτικές ταινιών (imdb movie dataset) όσο και κριτικές εστιατορίων (yelp benchmark dataset). Πέρα όμως από τα καθαρά πλεονεκτήματα απόδοσης η αλλαγή προσέφερε και επιπλέον δυνατότητες. Μια από αυτές είναι η προσαρμοστικότητα και η ευελιξία να μπορούν να προστεθούν με το μικρότερο δυνατό κόστος χρόνου και μετατροπών νέες υπηρεσίες που θα λειτουργούν επικουρικά στο σύστημα. Ακόμα και κομμάτια που είναι στενά συνδεδεμένα με τον αλγόριθμο όπως το context analysis, η διαδικασία αυτή μπορεί να μοντελοποιηθεί και να εισαχθεί ως παράμετρος, οπότε παρόλο που αυτό αποτελεί μέλημα του στενού πυρήνα του αλγορίθμου αυτό δεν αποτελεί αποτρεπτικό παράγοντα για να εισαχθεί χωρίς μεγάλο κόστος αλλαγών.

Πρόσφατο χαρακτηριστικό παράδειγμα που μπορεί να παρουσιάσει τα προβλήματα διαχείρισης μεγάλων δεδομένων καθώς και το θέμα της ανάλυσης ή εύρεσης κάποιου χαρακτηριστικού στοιχείου (heuristic) είναι η υπόθεση με τα Panama papers. Έγινε αντιληπτό στο δημοσιογραφικό κλάδο ότι πρέπει να υπάρχει επικουρούμενη βοήθεια από την επιστήμη των υπολογιστών ώστε να καταφέρουν να ανταπεξέλθουν σε μια τόσο μεγάλη σε αριθμό εγγράφων υπόθεση. Ταυτόχρονα φάνηκε ότι είναι ανθρωπίνως αδύνατο να επεξεργαστούν σε σύντομο χρονικό διάστημα τα παραγόμενα έγγραφα από την τράπεζα παρόλο που υπήρξε συνεργασία μεγάλων δημοσιογραφικών οίκων. Η εξαγωγή χρήσιμων στοιχείων και συμπερασμάτων δεν πήρε μόνο πάρα πολύ χρόνο αλλά κόστισε αρκετά, δεδομένο που κάνει αποτρεπτικές πολλές τέτοιου τύπου έρευνες για τα δημοσιογραφικά κανάλια. Μερικά προβλήματα είναι ο μικρός αριθμός εργαλείων για την εξεύρεση ευρημάτων και στοιχείων που θα προσδώσουν κάποιο έναυσμα για να ξεκινήσει

η δημοσιογραφική έρευνα αλλά και συνάμα η δυσκολία χρήσης των συστημάτων αυτών από μη εξειδικευμένο προσωπικό όπως οι δημοσιογράφοι. Η ανάγκη ευχρηστίας ακρίβειας και εμπλουτισμού της γκάμας των αλγορίθμων που μπορούν να προσφέρουν μεταδεδομένα είτε πληροφορίες για το περιεχόμενο (όπως ανάλυση συναισθήματος) κρίνεται αναγκαία.

Θα πρέπει να τονιστεί σε αυτό το σημείο το πρόβλημα της πολυγλωσσίας στα δεδομένα και των εργαλείων που κατά κανόνα μπορούν να ανταπεξέλθουν σε ένα περιορισμένο αριθμό γλωσσών. Το αποτέλεσμα είναι να δημιουργούνται δεδομένα δύο ταχυτήτων που προκαλούν μια στρέβλωση και ανακριβή εικόνας των αποτελεσμάτων μιας και «απαξιώνουν» μια ομάδα από χρήστες είτε έγγραφα που χρησιμοποιούν αυτές τις γλώσσες. Ταυτόχρονα μεγαλώνει ο προβληματισμός για τον κίνδυνο ότι οι άνθρωποι σταδιακά ζουν και ενημερώνονται από “ψηφιακά σύμπαντα” που οι αλγόριθμοι έχουν δημιουργήσει και παρέχουν μεμονωμένη ενημέρωση. Αυτή η εξάρτηση από τους αλγορίθμους -στη περίπτωση που δεν είναι πλήρης και αληθής- οδηγεί στο να περιορίζεται η οπτική γωνία των ανθρώπων όπως ακόμα και να «στενεύουν» οι πεποιθήσεις τους.

Απόρροια αυτών των προβλημάτων είναι η προσπάθεια συλλόγων και ομάδων όπου ζητούν πλέον να μπου στο μικροσκόπιο των κρατών και των κοινωνιών οι αλγόριθμοι και η τεχνητή νοημοσύνη. Στη κατεύθυνση του εμπλουτισμού των αλγορίθμων με περισσότερα δεδομένα όπου θα δώσουν μια πιο αντιπροσωπευτική ματιά στο κόσμο που δημιουργεί ο αλγόριθμος, η τεχνική που παρουσιάζεται θα μπορούσε να προσφέρει τη λύση. Η χρήση της σαν μετα-εργαλείο μπορεί σε συνδυασμό με υπάρχουσες μεθόδους και

αλγορίθμους να κάνει προσβάσιμα τα δεδομένα και να περιορίσει τις «αλγοριθμικές διακρίσεις» σε βάρος ανθρώπων ιδεών και των γεγονότων.

Στη προσπάθεια αυτή ρόλο παίζει και η πρόσβαση στις υπηρεσίες καθώς και η ευκολία δημιουργίας και συνδυασμός τους. Στο κομμάτι αυτό ακολουθώντας τις προτεινόμενες πρακτικές υλοποίησης μικρουπηρεσιών από τη CSA παρουσιάστηκε μια πρότυπη μορφή και περιγραφή τι πρέπει κάθε μικρουπηρεσία να έχει. Το περιεχόμενο αυτής της περιγραφής -Blueprint- διαχειρίζεται τόσο τεχνικές λεπτομέρειες όπως το τρόπο να γίνει η εγκατάσταση, η τεχνική διαχείριση της εφαρμογής και ο τρόπος και η μέθοδος που τεχνικά θα επικοινωνεί όσο και επιχειρηματικές λεπτομέρειες όπως το κόστος, η μέθοδος υπολογισμού του και γενικές πληροφορίες για το ίδιο το προϊόν σαν υπηρεσία (αδειδοτήσεις και περιγραφή). Αυτή η δομή επιτρέπει τη μετάβαση από την εποχή που ο κώδικας περιοριζόταν ή περιγραφόταν καθαρά με τεχνικούς όρους στην εποχή που ουσιαστικά ο κώδικας θα πρέπει να περιγράφεται και με όρους επιχειρηματικούς και υπηρεσιοστρεφής. Η μετάβαση αυτή θα οδηγήσει σε μια εντελώς διαφορετική κουλτούρα στο τρόπο προγραμματισμού αλλά και λογικής στο τρόπο δημιουργίας και χρήσης των προγραμμάτων και των εφαρμογών.

ΚΕΦΑΛΑΙΟ 7

Βιβλιογραφία

- [1] *EMC Digital Universe with Research & Analysis by IDC, The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things.*
- [2] *10 Key Marketing Trends for 2017, IBM Marketing Cloud,*
http://comsense.consulting/wp-content/uploads/2017/03/10_Key_Marketing_Trends_for_2017_and_Ideas_for_Exceeding_Customer_Expectations.pdf.
- [3] *DATA NEVER SLEEPS 6.0, How much data is generated every minute?,* https://www.domo.com/assets/downloads/18_domo_data-never-sleeps-6+verticals.pdf.

- [4] *Wall street daily*, March 2012,
<http://www.wallstreetdaily.com/2012/03/21/forever-growth-trends-and-five-stocks-set-to-profit/>.
- [5] *CERN Accelerating science*,
<http://public.web.cern.ch/public/en/LHC/Computing-en.html>.
- [6] <http://www.youtube.com/yt/press/statistics.html>.
- [7] <http://agbeat.com/tech-news/how-carriers-gather-track-and-sell-your-private-data>, August 2019.
- [8] <http://www.statisticbrain.com/twitter-statistics/>, August 2019.
- [9] http://www.information-management.com/issues/21_5/big-data-is-scaling-bi-and-analytics-10021093-1.html, August 2019.
- [10] D. Laney, «3d Data managment: controlling data volume, velocity and variety,» p. Appl. Delivery Strategies Meta Group (949), 2001.

- [11] A. S. a. J. Gray, « Science in an exponential world,» *Nature*, 440 (2006), pp. 23-24.
- [12] C. Lynch, « Big data: how do your data grow?,» *Nature*, 455 (7209) (2008), pp. 28-29.
- [13] M. A. Khan, M. F. Uddin and N. Gupta, "Seven V's of Big Data understanding Big Data to extract value," Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education, Bridgeport, CT, 2014, pp. 1-5..
- [14] Sun, Zhaohao. (2018). 10 Bigs: Big Data and Its Ten Big Characteristics. 10.13140/RG.2.2.31449.62566..
- [15] K. Borne, "Top 10 Big Data Challenges – A Serious Look at 10 Big Data V's," April 2014. [Online]. Available: <https://www.mapr.com/blog/top-10-big-data-challenges-%E2%80%93-serious-look-10-big-data-v%E2%80%99s..>
- [16] A. McAfee and E. Brynjolfsson, "Big data: The management revolution," *Harvard Business Review*, pp. 61-68, 2012..

- [17] IBM, "The Four V's of Big Data," 2015. [Online]. Available: <http://www.ibmbigdatahub.com/infographic/four-vs-bigdata..>
- [18] Arockia Panimalar, Varnekha Shree, Veneshia Kathrine, «The 17 V's Of Big Data,» *International Research Journal of Engineering and Technology(IRJET)*, 2017 Volume: 04 Issue: 09.
- [19] Eric Savitz, Gartner: Top 10 Strategic Technology Trends for 2013, October 2012. <http://www.forbes.com/sites/ericsavitz/2012/10/23/gartner-top-10-strategic-technology-trends-for-2013/>.
- [20] Eric Savitz, Gartner: 10 Critical Tech Trends for the Next Five Years, October 2012. <http://www.forbes.com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five-years/>..
- [21] American Institute of Physics (AIP). 2010. College Park, MD, (<http://www.aip.org/fyi/2010/>).

- [22] Mervis, J. 2012. “Agencies Rally to Tackle Big Data”, *Science*, 336(4):22, June 6, 2012.
- [23] *Fan Yan. Big Data Security and Privacy Protection [J]. Electronic Technology and Software Engineering, 2016(1): 227..*
- [24] *Feng Dengguo, Zhang Min, Li Yu. Big Data Security and Privacy Protection[J]. Chinese Journal of Computers, 2014, 37(1): 246-258..*
- [25] *Huo Honghua. Exploration of Security and Privacy Protection Technology in the Big Data Era[J]. Cyber Security Technology and Applications, 2016,11(05):79-88..*
- [26] *Luo Ying. Research on Big Data Security and Privacy Protection [J]. Information Communication, 2016(1):162-163..*
- [27] *Wei Kaimin, Weng Jian, Ren Kui. A Survey of Big Data Security Protection Technology. Journal of Network and Information Security, 2016, 2(4)..*

- [28] Marr, Bernard. "Really big data at walmart: Real-time insights from their 40+ petabyte data cloud." *Forbes.com* 23 (2017)..
- [29] SAS Whitepaper (2012) *Big Data Meets Big Data Analytics: Three Key Technologies for Extracting Real-Time Business Value from the Big Data That Threatens to Overwhelm Traditional Computing Architectures.*
- [30] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams,, Katherine A. YelickThe Landscape of Parallel Computing Research: A View from Berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley (2006).
- [31] Mehul Shah, Parthasarathy Ranganathan, Jichuan Chang, Niraj Tolia, David Roberts, Trevor MudgeData dwarfs: Motivating a Coverage Set for Future Large Data Center Workloads, , Technical Report HPL-2010-115, Hewlett Packard Laboratories, November 8 2010.

- [32] *S. Ghemawat, H. Gobioff, S. Leung, The Google File System, 19th ACM Symposium on Operating Systems Principles, Lake George, 2003..*
- [33] *J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, 6th Symposium on Operating System Design and Implementation, San Francisco, 2004..*
- [34] *M. Burrows, The Chubby Lock Service for Loosely-Coupled Distributed Systems, 7th Symposium on Operating System Design and Implementation, Seattle, 2006..*
- [35] *F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, R. Gruber, Bigtable: A Distributed Storage System for Structured Data, 7th Symposium on Operating System Design and Implementation, Seattle, 2006..*
- [36] *Wardman, D. Bringing Big Data to the Enterprise. 2013. Electronic Copy. Available online: ftp://public.dhe.ibm.com/software/os/systemz/pdf/09_-_Dan_Wardman_-*

_Bring_Big_Data_to_the_Enterprise_.pdf (accessed on 16 October 2019)..

- [37] Reinsel, D.; Gantz, J.; Rydning, J. *Data Age 2025: The Evolution of Data to Life-Critical*. SEAGATE. 2017. Electronic Copy. Available online: <https://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf>, (accessed on 16 October 2019).
- [38] Frontex. *Common Integrated Risk Analysis Model (CIRAM)-Comprehensive Update*. Capacity4dev Project. 2014. Electronic Copy. Available online: <https://europa.eu/capacity4dev/file/21158/download?token=D9Gkxx6U> (accessed on 16 October 2019)..
- [39] Hellenic Military Force. *GC 31-14 Control of Information; Hellenic Force Publishing House: Athens, Greece, 2005*.
- [40] Rudner, M. *Protecting Critical Energy Infrastructure Through Intelligence*. *Int. J. Intell. CounterIntell*. 2008, 21, 635–660..

- [41] Lee, S.; Shon, T. *Open source intelligence base cyber threat inspection framework for critical infrastructures. In Proceedings of the 2016 Future Technologies Conference (FTC), San Francisco, CA, USA, 6–7 December 2016.*
- [42] Gao, J.; Xie, C.; Tao, C. *Big Data Validation and Quality Assurance—Issues, Challenges, and Needs. In Proceedings of the 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, UK, 29 March–2 April 2016.*
- [43] Gani, A.; Siddiqa, A.; Shamshirband, S.; Hanum, F. *A survey on indexing techniques for big data: taxonomy and performance evaluation. Knowl. Inf. Syst. 2015, 46, 241–284.*
- [44] Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. *Deep learning applications and challenges in big data analytics. J. Big Data 2015, 2, 1.*
- [45] Fang, H. *Managing data lakes in big data era: What’s a data lake and why has it become popular in data management ecosystem. In Proceedings of the 2015 IEEE International Conference on Cyber*

Technology in Automation, Control, and Intelligent Systems (CYBER),
Shenyang, China, 8–12 June 2015.

- [46] O’Leary, D.E. *Embedding AI and Crowdsourcing in the Big Data Lake. IEEE Intell. Syst.* 2014, 29, 70–73..
- [47] Moody, D.L.; Kortink, M.A. *From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. In Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW’2000), Stockholm, Sweden, 5–6, June, Volume 28, pp. 5.1–5.12..*
- [48] A. Lenk, M. Klems, J. Nimis, S. Tai and T. Sandholm. 2009. *What’s inside the Cloud? An architectural map of the Cloud landscape. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD ’09). IEEE Computer, Society, Washington, DC, USA, 23-31..*
- [49] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Br, *Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility..*

- [50] P. Mell and T. Grance, *NIST Definition of Cloud Computing, Version 1.5, 2009, available online at: <http://csrc.nist.gov/groups/SNS/cloud-computing>.*
- [51] A. Menychtas, G. Kousiouris, D. Kyriazis, T. Varvarigou, "Minimizing technical complexities in emerging cloud computing platforms", *Proceedings (LNCS) of the Cloud Computing: Project and Initiatives, Ischia, Italy, 31 August- 3 September 2010.*
- [52] C. Legner, "Is there a Market for Web Services?, " *Service-Oriented Computing - ICSOC 2007 Workshops - Lecture Notes in Computer Science, E. Di Nitto and M. Ripeanu, eds., Vienna, Austria: Springer Berlin / Heidelberg, 2007, pp. 29-42.*
- [53] A. Menychtas, S. Garcia Gomez, A. Giessmann, A. Gatzioura, K. Stanoevska, J. Vogel, and V. Moulos, 2011. *A marketplace framework for trading cloud-based services. In Proceedings of the 8th international conference on Economics of Grids, Clouds, Systems,, and Services (GECON'11).*

- [54] M. P. Papazoglou and W. Heuvel, "Business Process Development - Life Cycle Methodology," *Communications of the ACM*, vol. 50, 2007, pp. 79-86..
- [55] M. Eurich, A. Giessmann, T. Mettler, and K. Stanoevska-Slabeva, "Revenue Streams of Cloud- based Platforms: Current State and Future Directions," *Proceedings of the Seventeenth Americas Conference on Information Systems (AMCIS)*, Detroit, Michigan: AISel., 2011, p. 10..
- [56] Future Internet Architecture (FIArch) Group, "Future Internet Design Principles", European Commission, http://ec.europa.eu/information_society/activities/foi/docs/fiarchdesignprinciples- v1.pdf, 2012..
- [57] Future Internet Architecture (FIArch) Group, "Future Internet Design Principles", European Commission, http://ec.europa.eu/information_society/activities/foi/docs/fiarchdesignprinciples- v1.pdf, 2012..
- [58] D. Kyriazis, A. Menychtas and T. Varvarigou, "Grid Workflows with encompassed Business Relationships: An approach establishing

Quality of Service Guarantees” in book “Quantitative Quality of Service for Grid Computing”, IGI Global Publishers, 2009..

- [59] Caracas, A., & Altmann, J. (2007, November). *A pricing information service for grid computing. In Proceedings of the 5th international workshop on Middleware for grid computing: held at the ACM/IFIP/USENIX 8th International Middleware Conference (p. 4).*, ACM/IFIP/USENIX 8th International Middleware Conference (p. 4). ACM.
- [60] Samimi, P., & Patel, A. (2011, March). *Review of pricing models for grid & cloud computing. In Computers & Informatics (ISCI), 2011 IEEE Symposium on (pp. 634-639). IEEE..*
- [61] C. Weinhardt, A. Anandasivam, B. Blau and J. Stöber, “*Business Models in the Service World,*” *IT Professional*, vol. 11, no. 2, pp. 28-33, March/April 2009..
- [62] G.M. Giaglis, S. Klein, and R.M. O’Keefe, “*The role of intermediaries in electronic marketplaces: developing a contingency model,*” *Information Systems Journal*, vol. 12, 2002, pp. 231-246..

- [63] *D. Neumann, J. Stoesser, A. Anandasivam, N. Borissov, SORMA-building an open grid market for grid resource allocation, in: Proceedings of the 4th International Workshop on Grid Economics and Business Models, Rennes, France, 2007..*
- [64] *J. Altmann, C. Courcoubetis, G. D. Stamoulis, M. Dramitinos, T. Rayna, M. Risch, C. Bannink, "GridEcon - A Market Place for Computing Resources," GECON 2008, Workshop on Grid Economics and Business Models, Springer LNCS, Las Palmas, Spain, August 2008..*
- [65] *Buyya R, Ranjan R, Calheiros RN. InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP, 2010), Busan, South Korea. Springer: Germany, 21–23 May 2010; 328–336..*
- [66] *E. Gillett, E. G. Brown, J. Staten and C. Lee, "Future View: The New Tech Ecosystems Of Cloud, Cloud Services, And Cloud Computing," Forrester, for Vendor Strategy Professionals, 28 August 2008..*

- [67] H. Li and J. Jeng, "CCMarketplace: a marketplace model for a hybrid cloud," *CASCON '10 Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, 2010..
- [68] Zhang, M., Ranjan, R., Nepal, S., Menzel, M., & Haller, A. (2012). *A declarative recommender system for cloud infrastructure services selection. In Economics of Grids, Clouds, Systems, and Services (pp. 102-113). Springer Berlin Heidelberg..*
- [69] Oberle, K., Cherubini, D., & Cucinotta, T. (2013). *End-to-End Service Quality for Cloud Applications. In Economics of Grids, Clouds, Systems, and Services (pp. 228-243). Springer International Publishing..*
- [70] Yaqub, E., Yahyapour, R., Wieder, P., & Lu, K. (2012). *A protocol development framework for SLA negotiations in cloud and service computing. In Economics of Grids, Clouds, Systems, and Services (pp. 1-15). Springer Berlin Heidelberg..*
- [71] Y. Bakos, "The Emerging Role of Electronic Marketplaces on the Internet," *Communications of the ACM*, vol. 41, 1998, pp. 35-42.

- [72] Dinh Khoa Nguyen, Francesco Lelli, Yehia Taher, Michael Parkin, Mike P. Papazoglou, and Willem-Jan van den Heuvel. 2011. *Blueprint template support for engineering cloud-based services. In Proceedings of the 4th European conference on Towards a service, based internet(ServiceWave'11)..*
- [73] Nguyen D.K., Lelli F., Papazoglou M.P., van den Heuvel W.-J. *Blueprinting Approach in Support of Cloud Computing. Future Internet. 2012; 4(1):322-346..*
- [74] Garcia-Gomez, S., Jimenez-Ganan, M., Taher, Y., Momm, C., Junker, F., Biro, J., Menychtas, A., Andrikopoulos, V., Strauch, S. *Challenges for the comprehensive management of Cloud Services in a PaaS framework. Scalable Computing: Practice and Experience., Selected Papers from the International Workshop on Clouds for Business and Business for Clouds. Vol 13, No 3, 2012..*
- [75] S. García-Gómez et al., "4CaaS: Comprehensive Management of Cloud Services through a PaaS," 2012 IEEE 10th International

Symposium on Parallel and Distributed Processing with Applications, Leganes, 2012, pp. 494-499..

- [76] *Distributed Management Task Force DMTF. Open virtualization format specification. Specification DSP0243 v1.0.0d. Technical report, available online at: <https://www.coinor.org/OS/publications/optimizationServicesFramework2008.pdf>.*
- [77] *P. Mell and T. Grance, NIST Definition of Cloud Computing, Version 15, 2009, available online at: <http://csrc.nist.gov/groups/SNS/cloud-computing>.*
- [78] *M. Morris, M. Schindehutte, and J. Allen, "The entrepreneur's business model: toward a unified perspective," Journal of Business Research, vol. 58(6), 2005, pp. 726-735..*
- [79] *M. Armbrust, A. Fox, R. Griffith, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communications of the ACM, vol. 53, 2010, pp. 50-58..*

- [80] *A. Menychtas, S. Garcia Gomez, A. Giessmann, A. Gatzioura, K. Stanoevska, J. Vogel, and V. Moulos, 2011. A marketplace framework for trading cloud-based services. In Proceedings of the 8th international conference on Economics of Grids, Clouds, Systems,, and Services (GECON'11)..*
- [81] *F. Junker, J. Vogel, K. Stanoevska, Aggregating Price Models for Composite Services in Cloud Service Marketplaces, The 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, Leganés, Madrid, 10-13 July 2012..*
- [82] *S. García-Gómez et al., "4CaaS: Comprehensive Management of Cloud Services through a PaaS," 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, Leganes, 2012, pp. 494-499..*
- [83] *<https://attic.apache.org/>.*
- [84] *<https://libcloud.apache.org/>.*

- [85] C.-S. Peng, S.-K. Chen, J.-Y. Chung, A. Roy-Chowdhury, and V. Srinivasan. *Accessing existing business data from the world wide web. IBM systems journal*, 37(1):115–132, 1998..
- [86] "2.101 Definitions", *U.S. Federal Acquisition Regulations*, archived from the original on 2017-01-30, retrieved 2017-02-01.
- [87] Zhang X., Schiffman J., Gibbs S., Kunjithapatham A., Jeong S.: *Securing elastic applications on mobile devices for cloud computing, Proceedings of the 2009 ACM workshop on Cloud computing security*, 2009..
- [88] Haifei Li, Jun-Jang Jeng, "CCMarketplace: a marketplace model for a hybrid cloud," *CASCON '10 Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, 2010..
- [89] I. Chamodrakas, D. Batis, D. Martakos, "Supplier selection in electronic marketplaces using satisficing and fuzzy AHP," *Expert Systems with Applications*, vol. 37, issue 1, pp. 490-498, January 2010..

- [90] Avinash Lakshman and Prashant Malik. 2010. *Cassandra: a decentralized structured storage system*. *SIGOPS Oper. Syst. Rev.* 44, 2 (April 2010), 35-40..
- [91] *NIST Definition of Cloud Computing*, Peter Mell and Tim Grance, Version 15, 2009, available online at: <http://csrc.nist.gov/groups/SNS/cloud-computing>.
- [92] Sagawa C., Yoshida H., Take R., Shimada J., "Cloud Computing Based on Service-Oriented Platform", *Fujitsu Sci. Tech. J.*, Vol. 45, No. 3, pp 283-289. July 2009.
- [93] *Puppet*, <http://www.puppetlabs.com>.
- [94] *Chef*, <https://www.chef.io/get-chef/>.
- [95] Ozsu M. T., Valduriez P.: *Principles of Distributed Databases* (2nd edition), Prentice-Hall (1999).
- [96] Morris R., Lin D., "Variance of Aggregated WebTraffic", *IEEE INFOCOM 2000*, Tel Aviv, March 2000, pages 360-366.

- [97] *Vaquero L.M., Merino L.R., Buyya R., "Dynamically scaling applications in the cloud", ACM SIGCOMM Computer Communication Review, Vol. 41, No. 1., 2011, pp. 45-52.*
- [98] *Galán F., Sampaio A., Rodero-Merino L., Loy I., Gil V., Vaquero L. M. , Wusthoff M., "Service specification in cloud environments based on extensions to open standards," in Fourth International Conference on COMMunication System softWARE and middlewaRE, (COMSWARE 2009), Jun. 2009..*
- [99] *Borealis: <http://cs.brown.edu/research/borealis/public/>.*
- [100] *Infosphere: <http://www-01.ibm.com/software/data/infosphere/>.*
- [101] *Esper: <http://esper.codehaus.org/>.*
- [102] *Storm: <http://storm-project.net>.*
- [103] *S4: <http://incubator.apache.org/s4/>.*

- [104] *V. Gulisano, R. Jiménez-Peris, M. Patiño-Martínez, C. Soriente, P. Valduriez: StreamCloud: An Elastic and Scalable Data Streaming System. IEEE Trans. Parallel Distrib. Syst. 23(12): 2351-2365 (2012).*
- [105] *<https://leanbigdata.eu/>.*
- [106] *Donald Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, and Stanley B. Zdonik. Monitoring streams - a new class of data management applications. In 28th International Conference, on Very Large Data Bases (VLDB'02), pages 215-226, 2002.*
- [107] *Arvind Arasu, Shivnath Babu, and Jennifer Widom. The cql continuous query language: semantic foundations and query execution. VLDB J., 15(2):121-142, 2006.*
- [108] *<https://zookeeper.apache.org/>.*
- [109] *Ping-Feng Pai and Chih-Sheng Lin. 2005. A hybrid ARIMA and support vectormachines model in stock price forecasting. Omega33, 6 (2005), 497–505., Jung-Hua Wang and Jia-Yann Leu. 1996. Stock market*

trend prediction using ARIMA-based neural networks. In Neural Networks, 1996., IEEE International Conference on, Vol. 4. IEEE, 2160–2165.

- [110] *Bo Qian and Khaled Rasheed. 2007. Stock Market Prediction with Multiple Classifiers. Applied Intelligence 26, 1 (2007), 25–33.*

- [111] *Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. 2005. Forecasting stockmarket movement direction with support vector machine., Computers & Operations Research 32, 10 (2005), 2513–2522..*

- [112] *Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market., In Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2823–2824..*

- [113] *George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. ACM Trans. Speech Lang. Process. 5, 3, Article 5 (October 2008), 39 pages.*

- [114] Δ. Τζαννέτος, *Ανάλυση Συναισθήματος Σε Δεδομένα Κοινωνικών Δικτύων Με Χρήση Γράφων N-γραμμάτων*, <http://artemis.cslab.ece.ntua.gr:8080/jspui/handle/123456789/12585>.
- [115] *Giannakopoulos, George & Karkaletsis, Vangelis & Vouros, George & Stamatopoulos, Panagiotis. (2008). Summarization system evaluation revisited: N-gram graphs. ACM Trans. Speech Lang. Process.. 5. 1-39. 10.1145/1410358.1410359..*
- [116] *Giannakopoulos, George & Kiomourtzis, George & Karkaletsis, Vangelis. (2014). Newsum: "N-gram graph"-based summarization in the real world. 10.4018/978-1-4666-5019-0.ch009..*
- [117] *EL-Haj, Mahmoud & Giannakopoulos, George & li, Lei & Forascu, Corina. (2013). Multi-document multilingual summarization corpus preparation, Part 1: Arabic, English, Greek, Chinese, Romanian..*
- [118] *Elhadad, Michael & Miranda-Jiménez, Sabino & Steinberger, Josef & Giannakopoulos, George. (2013). Multi-document multilingual summarization corpus preparation, Part 2: Czech, Hebrew and Spanish..*

- [119] *Σκέψη, αργή και γρήγορη, Daniel Kahneman.*
- [120] *V. Moulos, F. Aisopos, P. Kranas, S. Stamokostas, A. Evangelinou, M. Kardara, J. Violos, T. Varvarigou and A. Psychas, A Real Time Sentiment Analysis Algorithm using Streaming Data from Twitter, Abstractions and Use Cases of converged Big Data,, Telecom and IoT technologies Workshop, EuCNC 2016.*
- [121] *<https://wordnet.princeton.edu/>.*
- [122] *Jurafsky D., Martin J. H., Speech and Language Processing, 3rd ed. (draft), 2017..*
- [123] *Santorini B., Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision), July 1990..*
- [124] *Hatzivassiloglou V., McKeown K., Predicting the semantic orientation of adjectives, in: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association, for Computational Linguistics, pp. 174-181.*

- [125] Turney P., *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, in: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, July 2002, pp.417-424.
- [126] Kučera H., Francis W. N., *Computational analysis of present-day American English*, Brown University Press, Providence, 1967..
- [127] Hatzivassiloglou V., McKeown K., *Predicting the semantic orientation of adjectives*, in: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the, Association for Computational Linguistics*, pp. 174-181.
- [128] Turney P., *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, in: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, July 2002,, pp. 417-424.

- [129] Chapman S., *Key Ideas in Linguistics and the Philosophy of Language*, Edinburgh University Press, 2009..
- [130] Liberman M., *Text on Tap: the ACL/DCI*, in: *Proceedings of the 1989 DARPA Speech and Natural Language Workshop*, Cape Cod, Massachusetts, 1989..
- [131] <https://ai.stanford.edu/~amaas/data/sentiment/>.
- [132] <https://www.yelp.com/dataset/challenge>.
- [133] *NIST Definition of Microservices, Application Containers and System Virtual Machines*,
https://csrc.nist.gov/CSRC/media/Publications/sp/800-180/draft/documents/sp800-180_draft.pdf.
- [134] *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems*,
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1.pdf>.

- [135] Max Knutsson, Max Knutsson, *Challenges of Service-Oriented Architecture (SOA)*, <https://pdfs.semanticscholar.org/23ef/7b2abcfaed46f37ba7330c1f4409ca8aff6a.pdf>, June, 2015.
- [136] Janek Richter, Dirk Basten, *Exploring the Influence of Service-oriented Architectures on Organizational Agility – A Case Study*.
- [137] Stojanovic, Tim & Ballinger, Rhoda & Lalwani, Chandra. (2004). *Successful integrated coastal management: Measuring it with research and contributing to wise practice. Ocean & Coastal Management. 47. 273-298. 10.1016/j.ocecoaman.2004.08.001..*
- [138] V. Alkkiomäki, *The role of service-oriented architecture as a part of the business model*, <http://dx.doi.org/10.1504/IJBIS.2016.074764>.
- [139] Joachim, N., Beimborn, D., and Weitzel, T. 2013. "The Influence of SOA Governance Mechanisms on IT, Flexibility and Service Reuse," *The Journal of Strategic Information Systems* (22:1), pp. 86–101.

- [140] Zachman, John A., *The Framework for Enterprise Architecture: Background, Description and Utility*.
- [141] Joachim, N. 2011. "A Literature Review of Research on Service-Oriented Architectures (SOA).", Proceedings of the 17th Americas Conference on Information Systems, V. Sambamurthy, and M.: Tanniru (eds.), Detroit, Michigan. August 4-7 2011, Association for Information Systems, pp. 1–11, Characteristics, Adoption Determinants, Governance Mechanisms, and Business Impact," in .
- [142] *Management, Oracle Reference Architecture - Information*, <https://www.oracle.com/technetwork/topics/entarch/oracle-ra-info-mgmt-r3-1-1980395.pdf>.
- [143] Thomas Erl. 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA..
- [144] The Open Group Guide, *SOA Source Book*.

- [145] Broad, Ellen. "Who gets held accountable when a facial recognition algorithm fails?." *IQ: The RIM Quarterly* 34.4 (2018): 18..
- [146] Dietterich, Tom. "Overfitting and undercomputing in machine learning." *ACM computing surveys* 27.3 (1995): 326-327..
- [147] Wu, Hao, and Jonathan L. Shapiro. "Does overfitting affect performance in estimation of distribution algorithms." *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006..
- [148] Ghasemian, Amir, Homa Hosseinmardi, and Aaron Clauset. "Evaluating overfit and underfit in models of network community structure." *IEEE Transactions on Knowledge and Data Engineering* (2019)..
- [149] Merkulov, Daniil, and Ivan Oseledets. "Empirical study of extreme overfitting points of neural networks." *arXiv preprint arXiv:1906.06295* (2019)..

- [150] Arbain, Ain Najwa, and B. Yushalinie Pillay Balakrishnan. "A Comparison of Data Mining Algorithms for Liver Disease Prediction on Imbalanced Data." *International Journal of Data Science and Advanced Analytics* 1.1 (2019): 1-11..
- [151] Liardet, Pierre, et al., eds. *Artificial Evolution: 6th International Conference, Evolution Artificielle, EA 2003, Marseilles, France, October 27-30, 2003, Revised Selected Papers*. Vol. 2936. Springer Science & Business Media, 2004..
- [152] *Andrew, Christopher. "Codebreaking and signals intelligence." Intelligence and National Security* 1.1 (1986): 1-5..