



**Πανεπιστήμιο Πειραιώς**  
**Τμήμα Ψηφιακών Συστημάτων**

Μεταπτυχιακό Πρόγραμμα Σπουδών  
Διδακτική της Τεχνολογίας & Ψηφιακά Συστήματα  
Κατεύθυνση Δικτυοκεντρικών Συστημάτων

Μελέτη, σχεδιασμός και ανάπτυξη  
υπηρεσιοστρεφούς πλατφόρμας  
διαχείρισης υπηρεσιών  
επιχειρησιακών διεργασιών σε  
υπολογιστικά νέφη

Κωστάντος Κων/νος ME 10089

## Περίληψη

Το υπολογιστικό νέφος αναφέρεται σε υπολογιστικό παράδειγμα που μετατρέπει τους υπολογιστικούς πόρους, το δίκτυο και τους πόρους αποθήκευσης σε υπηρεσία. Πλέον έχει μεγάλη απήχηση στη αγορά και χρησιμοποιείται όλο και περισσότερο από επιχειρήσεις και οργανισμούς. Το υπολογιστικό νέφος παρέχει τρία βασικά μοντέλα υπηρεσιών την υποδομή σαν υπηρεσία που είναι δυνατότητα που παρέχεται στον χρήστη να έχει πρόσβαση σε υπολογιστικούς, αποθηκευτικούς, δικτυακούς και άλλους στοιχειώδεις πόρους, πάνω στους οποίους μπορεί να εγκαθιστά και να εκτελεί λογισμικό, το οποίο περιλαμβάνει τόσο εφαρμογές όσο και λειτουργικά συστήματα, την πλατφόρμα σαν υπηρεσία που παρέχει δυνατότητες στο χρήστη να εγκαθιστά στην υποδομή ενός νέφους εφαρμογές δικής του δημιουργίας είτε άλλες εφαρμογές, χρησιμοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες και εργαλεία που διατίθενται από τον πάροχο και το λογισμικό σαν υπηρεσία που δίνει την δυνατότητα σε έναν καταναλωτή να χρησιμοποιήσει τις εφαρμογές ενός παρόχου που εκτελούνται σε μια υποδομή νέφους.

Η υπηρεσιοστρεφής αρχιτεκτονική παρέχει σημαντικά πλεονεκτήματα στην ανάπτυξη εφαρμογών όπως την επαναχρησιμοποίηση προγραμματιστικών μονάδων που επιτρέπει την γρήγορη ανάπτυξη και δημιουργία καινούριων επιχειρησιακών διαδικασιών, την δυνατότητα γρήγορης και εύκολης αλλαγής υφιστάμενων επιχειρησιακών διαδικασιών που στηρίζονται σε ένα σύνολο υπηρεσιών, την δυνατότητα για σταδιακή ανάπτυξη ενός πλήρους συστήματος που είναι και το βασικό μέρος της ολοκλήρωσης ενός συστήματος σε υπηρεσιοστρεφή αρχιτεκτονική.

Το υπολογιστικό νέφος υποστηρίζει την κοινή χρήση, τη διασύνδεση και τη χρήση διαφόρων πόρων, ενσωματωμένων στο πλαίσιο ενός δυναμικού υπολογιστικού συστήματος. Η διαχείριση των λειτουργιών της ροής εργασίας κάποιας εφαρμογής σε υπολογιστικό νέφος απαιτεί την ενορχήστρωση των επιχειρησιακών διαδικασιών που τη συνθέτουν. Στην εργασία αυτή θα ασχοληθούμε με την ανάπτυξη μίας υπηρεσιοστρεφούς πλατφόρμας εκτέλεσης ροών εργασίας σε περιβάλλον υπολογιστικού νέφους. Η αρχιτεκτονική της πλατφόρμας είναι σχεδιασμένη καθ' αυτόν τον τρόπο ώστε να είναι εύκολα επεκτάσιμη σε ένα περιβάλλον υπολογιστικού νέφους. Τα πειραματικά αποτελέσματα ήταν αρκετά ενθαρρυντικά ότι αυτή η αρχιτεκτονική μπορεί να χρησιμοποιηθεί σε περιβάλλον υπολογιστικού νέφους.

## Πίνακας Περιεχομένων

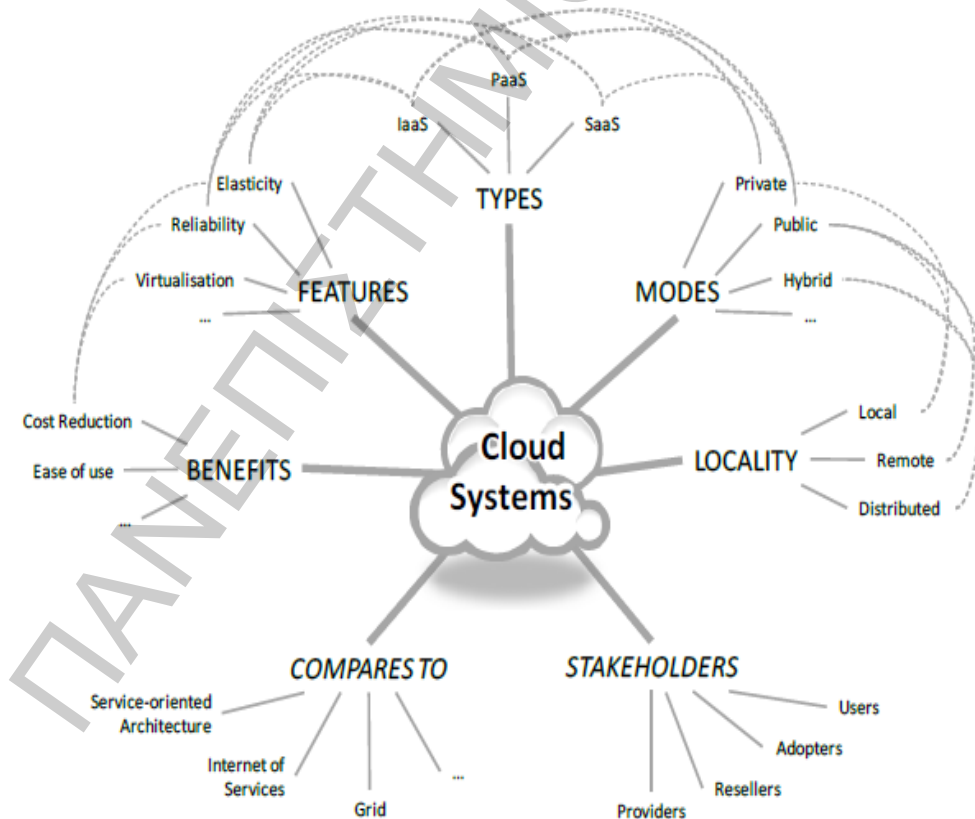
Περίληψη.....	2
1 Εισαγωγή .....	5
2 Υπολογιστικά νέφη.....	7
2.1 Γενικές πληροφορίες για τα υπολογιστικά νέφη.....	7
2.1.1 Βασικά χαρακτηριστικά:.....	7
2.1.2 Μοντέλα Υπηρεσιών: .....	8
2.1.3 Μοντέλα ανάπτυξης:.....	9
2.2 Πάροχοι νεφών.....	10
2.2.1 Υποδομή σαν υπηρεσία .....	10
2.2.2 Πλατφόρμα σαν υπηρεσία .....	17
2.3 Cross Platform Cloud APIs .....	19
3 Ροές Εργασίας .....	20
3.1 Ορισμοί.....	20
3.2 Κατηγορίες .....	21
3.3 Υπάρχοντα Συστήματα .....	23
4 Πλατφόρμα Διαχείρισης.....	24
4.1 Αρχιτεκτονική υλοποίησης.....	24
4.1.1 Κεντρική βάση δεδομένων.....	25
4.1.2 Drools .....	27
4.1.3 JBPM .....	27
4.1.4 BPMN.....	28
4.1.5 Java EE 6 .....	29
4.1.6 Jboss Application Server .....	29
4.1.7 MySQL.....	29
4.1.8 RESTeasy.....	29
4.2 Ροή Εργασίας.....	30
5 Περιβάλλον Υπολογιστικού νέφους.....	32
5.1 Χαρακτηριστικά.....	32
5.1.1 Πολλαπλά εργαλεία διαχείρισης .....	32
5.1.2 Ετερογενείς υποδομές και πόροι.....	32
5.1.3 Ανάθεση υπολογιστικών πόρων κατ' εντολή.....	33
5.1.4 Εικονικά μηχανήματα και τύποι στιγμιότυπων .....	33
5.1.5 Αποθήκευση .....	33

5.1.6	Δικτυακοί πόροι .....	34
5.1.7	Διαχείριση ελέγχου και επίβλεψη .....	34
5.1.8	Ελαστικότητα.....	35
5.1.9	Μηχανισμός ειδοποιήσεων.....	35
5.1.10	Πλαισιοποίηση .....	35
5.1.11	Περιγραφή αρχιτεκτονικής του περιβάλλοντος υπολογιστικού νέφους.....	35
5.2	Τοπολογία Πειράματος .....	37
5.3	Προσομοιώσεις .....	38
5.3.1	Προσομοίωση 1.....	39
5.3.2	Προσομοίωση 2.....	42
6	Συμπεράσματα .....	45
7	Βιβλιογραφία.....	46

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΝ

## 1 Εισαγωγή

Το υπολογιστικό νέφος (cloud computing) εξελίσσεται πολύ γρήγορα και είναι ένα από τα πιο ελπιδοφόρα παραδείγματα στον χώρο των τηλεπικοινωνιών και των πληροφοριακών συστημάτων (Information and Communication Technology -ICT). Η χρησιμότητά του για χρήστες και επιχειρήσεις είναι πλέον ευρέως αποδεκτή [1]. Ο όρος νέφος προέρχεται από την δεκαετία του 90 αναφερόμενος στις τηλεπικοινωνιακές δομές που επέτρεπαν τον διαμοιρασμό του φορτίου (“telecom clouds”) [2], και ακόμα πιο πριν σύμφωνα με την Wikipedia, το θεώρημα του υπολογιστικού νέφους είχε προβληθεί από μία ομιλία του John McCarthy (το 1961) όπου ανέφερε ότι το “time-sharing” μεταξύ υπολογιστών θα μπορούσε να οδηγήσει στο να παρέχονται οι υπολογιστικοί πόροι σαν δημόσια υπηρεσία [3]. Το 2001 ο όρος υιοθετήθηκε σε μία παρουσίαση για το .net πλαίσιο για να αναφερθεί στους υπολογιστικούς πόρους που συνθέτουν το internet,[4] αλλά έγινε ευρέως γνωστός από την δημοσίευση της Amazon για το Elastic Compute Cloud EC2 το 2006 [5]. Στις μέρες μας το υπολογιστικό νέφος αναφέρεται σαν ένα υπολογιστικό πρότυπο όπου η θεμελίωση του είναι να προσφέρει σαν υπηρεσίες υπολογιστικούς πόρους και εφαρμογές, συχνά αναφέρεται σαν XaaS (Everything as a Service). Ενώ υπάρχουν πολλοί ορισμοί και ονομασίες για το υπολογιστικό νέφος η ακόλουθη εικόνα (Εικόνα 1) συλλαμβάνει τον κύριο όγκο των όρων και των χαρακτηριστικών που συνθέτουν ένα υπολογιστικό νέφος.[5]



Εικόνα 1. Υπολογιστικό νέφος.

Τα συστήματα ροών εργασίας χρησιμοποιούνται κατά βάση από οργανισμούς κι επιχειρήσεις. Είναι συστήματα που προσφέρουν τη δυνατότητα σε κάποιον οργανισμό να διαχειρίζεται τη ροή της εργασίας μέσα στα πλαίσια λειτουργίας του. Πιο συγκεκριμένα, τα συστήματα αυτά χρησιμεύουν στον καθορισμό, την εκτέλεση, την παρακολούθηση και την ανακατεύθυνση των περιπτώσεων εργασίας μέσα σε ένα εργασιακό περιβάλλον.

Τα συστήματα ροών εργασίας περιέχουν δυο βασικά στοιχεία :

- Τη μονάδα μοντελοποίησης της ροής εργασίας
- Τη μονάδα εκτέλεσης της ροής εργασίας

Η μονάδα μοντελοποίησης αναφέρεται στους διαχειριστές

του συστήματος. Τους επιτρέπει να ορίζουν δραστηριότητες και διεργασίες, να τις αναλύουν και να τις προσομοιώνουν, καθώς και να τις αναθέτουν σε ανθρώπινο δυναμικό.

Η μονάδα εκτέλεσης παρέχει στον τελικό χρήστη του συστήματος μια διεπαφή εκτέλεσης και προσομοίωσης, ενώ περιέχει και μια μηχανή ροής εργασίας η οποία βοηθά ή/και εκτελεί την καθοδήγηση των διεργασιών και δραστηριοτήτων μέσα στο περιβάλλον. Μια εφαρμογή αποκαλείται εφαρμογή ροής εργασίας (workflow application) όταν διάφοροι άνθρωποι, ή και εφαρμογές, συμμετέχουν στην επεξεργασία δεδομένων της εφαρμογής, ώστε να ολοκληρωθεί μια περίπτωση της διεργασίας που περιγράφει η εφαρμογή. Με άλλα λόγια, κάποια περίπτωση εργασίας είναι απαραίτητο να περάσει από διάφορα στάδια επεξεργασίας, είτε από μια εφαρμογή είτε από ανθρώπινο παράγοντα, προκειμένου να θεωρηθεί ως ένα ολοκληρωμένο στιγμιότυπο της συνολικής διεργασίας.

Μια διεργασία πάντα εμπεριέχει διαρκείς αλλαγές κι ενημερώσεις των στοιχείων που διακινούνται στα πλαίσιά της. Παρ' όλα αυτά, είναι ιδιαίτερα εύκολο να εισαχθεί κάποιος παράγοντας στη λειτουργία μιας εφαρμογής, αφού δε χρειάζεται να γίνουν σημαντικές αλλαγές στο μοντέλο.

## 2 Υπολογιστικά νέφη

### 2.1 Γενικές πληροφορίες για τα υπολογιστικά νέφη

Το Υπολογιστικό Νέφος είναι ένα μοντέλο για την ενεργοποίηση ευέλικτης, κατόπιν αιτήματος και διαρκούς πρόσβασης σε ένα σύνολο διαμοιρασμένων και διαμορφώσιμων υπολογιστικών πόρων (πχ. δίκτυα, εξυπηρετητές, αποθηκευτικοί χώροι, εφαρμογές και υπηρεσίες) οι οποίοι μπορούν πολύ γρήγορα να προβλεφθούν και διατεθούν με ελάχιστο διαχειριστικό κόστος ή αλληλεπίδραση χρήστη - πάροχου.

Αυτό το μοντέλο Νέφους αποτελείται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα υπηρεσιών και τέσσερα μοντέλα ανάπτυξης τα οποία παρουσιάζουμε και αναλύουμε παρακάτω.

#### 2.1.1 Βασικά χαρακτηριστικά:

##### 2.1.1.1 Αυτο-εξυπηρέτηση κατόπιν αιτήματος

Ένας χρήστης μπορεί μονομερώς να εφοδιαστεί και να “καταναλώσει” υπολογιστικές υπηρεσίες, όπως χρόνο χρήσης σε εξυπηρετητές και πρόσβαση σε δικτυακό αποθηκευτικό χώρο, αυτοματοποιημένα, χωρίς να απαιτείται προσωπική αλληλεπίδραση με τον κάθε πάροχο αυτών των υπηρεσιών ξεχωριστά.

##### 2.1.1.2 Ευρεία δικτυακή πρόσβαση

Οι δυνατότητες και οι υπηρεσίες αυτές είναι διαθέσιμες και προσβάσιμες μέσω του δικτύου χρησιμοποιώντας μηχανισμούς που προάγουν την χρήση ετερογενών προγραμμάτων/πλατφόρμων πελάτη (κινητά τηλέφωνα, ταμπλέτες, φορητοί υπολογιστές, σταθμούς εργασίας κ.α.).

##### 2.1.1.3 Συγκέντρωση πόρων

Οι υπολογιστικοί πόροι ενός παρόχου είναι συγκεντρωμένοι έτσι ώστε να εξυπηρετούν πολλαπλούς χρήστες/καταναλωτές, χρησιμοποιώντας μοντέλα πολλαπλής μίσθωσης (multi-tenancy), ο καθένας από τους οποίους έχει διαφορετικούς φυσικούς ή εικονικούς πόρους δυναμικά εκχωρημένους σε αυτόν σύμφωνα με τις δικές του απαιτήσεις. Η έννοια της ανεξαρτησίας από την τοποθεσία, όσον αφορά τους υπολογιστικούς πόρους, έχει την σημασία ότι ο πελάτης δεν έχει γνώση ή έλεγχο σχετικά με την ακριβή θέση των παρεχομένων πόρων αλλά μπορεί να θέσει σχετικούς περιορισμούς σε υψηλότερο επίπεδο αοριστίας (πχ. χώρα, πολιτεία, κέντρο δεδομένων). Παραδείγματα τέτοιων πόρων συμπεριλαμβάνουν αποθηκευτικό χώρο, υπολογιστική ισχύ, μνήμη και χρήση δικτύου.

##### 2.1.1.4 Άμεση ελαστικότητα

Οι υπηρεσίες αυτές των Νεφών μπορούν να εκχωρηθούν και διατεθούν με έναν ελαστικό και μερικές φορές αυτόματο τρόπο. Αυτό σημαίνει ότι μπορούν να αυξήσουν ή μειώσουν τις δυνατότητες τους και την χωρητικότητα τους αναλόγως με την ζήτηση. Για τον καταναλωτή, οι δυνατότητες που του διατίθενται συχνά φαίνονται να είναι απεριόριστες και μπορούν να πιστωθούν σε οποιαδήποτε ποσότητα ανά πάσα στιγμή.

#### **2.1.1.5 Υπηρεσία μετρήσεων**

Τα συστήματα Νεφών μπορούν αυτόματα να ελέγχουν και να βελτιστοποιούν την χρήση των πόρων αξιοποιώντας δυνατότητες μετρήσεων σε κάποιο επίπεδο αφαίρεσης (abstraction) κατάλληλο στις διάφορες παρεχόμενες υπηρεσίες. Η χρήση των παρεχομένων πόρων παρακολουθείται, ελέγχεται και αναφέρεται, παρέχοντας έτσι διαφάνεια τόσο για τον πάροχο όσο και για τον καταναλωτή.[6]

### **2.1.2 Μοντέλα Υπηρεσιών:**

#### **2.1.2.1 Λογισμικό ως Υπηρεσία (Software as a Service - SaaS)**

Η δυνατότητα που παρέχεται σε έναν καταναλωτή να χρησιμοποιήσει τις εφαρμογές ενός παρόχου που εκτελούνται σε μια υποδομή Νέφους. Οι εφαρμογές αυτές είναι προσβάσιμες από διάφορες συσκευές πελατών είτε μέσω διεπαφών, όπως ένα πρόγραμμα περιήγησης διαδικτύου (thin client), είτε μέσω ολοκληρωμένων προγραμμάτων. Ο καταναλωτής δεν διαχειρίζεται ούτε ελέγχει την υποδομή Νέφους συμπεριλαμβανομένων του δικτύου, εξυπηρετητών, λειτουργικών συστημάτων αλλά ούτε και τις δυνατότητες των εφαρμογών, με πιθανή εξαίρεση μόνο περιορισμένες ρυθμίσεις των εφαρμογών αυτών που σχετίζονται με τον ίδιο τον χρήστη. [6]

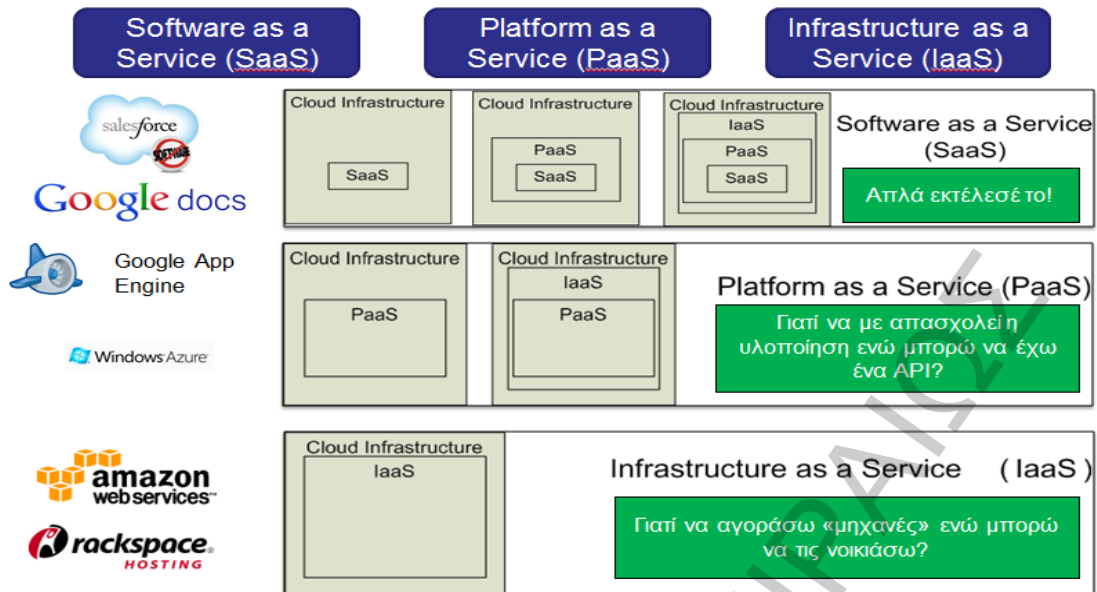
#### **2.1.2.2 Πλατφόρμαως Υπηρεσία (Platform as a Service - PaaS)**

Παρέχονται δυνατότητες στο χρήστη να εγκαθιστά στην υποδομή ενός Νέφους εφαρμογές δικής του δημιουργίας είτε άλλες εφαρμογές, χρησιμοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες και εργαλεία που διατίθενται από τον πάροχο. Ο χρήστης δεν διαχειρίζεται ή ελέγχει την υπάρχουσα υποδομή Νέφους συμπεριλαμβανομένων του δικτύου, των εξυπηρετητών, των λειτουργικών συστημάτων ή του χώρου αποθήκευσης. Παρόλα αυτά μπορεί να έχει έλεγχο πάνω στις εγκατεστημένες εφαρμογές όπως και σε πιθανές ρυθμίσεις του περιβάλλοντος φιλοξενίας των εφαρμογών αυτών.[6]

#### **2.1.2.3 Υποδομήως Υπηρεσία (Infrastructure as a Service - IaaS)**

Η δυνατότητα που παρέχεται στον χρήστη να έχει πρόσβαση σε υπολογιστικούς, αποθηκευτικούς, δικτυακούς και άλλους στοιχειώδεις πόρους, πάνω στους οποίους μπορεί να εγκαθιστά και να εκτελεί λογισμικό, το οποίο περιλαμβάνει τόσο εφαρμογές όσο και λειτουργικά συστήματα. Ο χρήστης δεν διαχειρίζεται ή ελέγχει την υπάρχουσα υποδομή Νέφους συμπεριλαμβανομένων του δικτύου, των εξυπηρετητών, των λειτουργικών συστημάτων ή του χώρου αποθήκευσης. Μπορεί παρόλα αυτά να έχει περιορισμένο έλεγχο πάνω σε δικτυακά στοιχεία (π.χ. τοίχος προστασίας, τρόπο πρόσβασης).[6]





Εικόνα 2. Μοντέλα υπηρεσιών.

### 2.1.3 Μοντέλα ανάπτυξης:

#### 2.1.3.1 Ιδιωτικό Νέφος (Private Cloud)

Η υποδομή του Νέφους παρέχεται για αποκλειστική χρήση από ένα μοναδικό οργανισμό που μπορεί να αποτελείται από πολλούς χρήστες (π.χ. διαφορετικά τμήματα μιας επιχείρησης). Μπορεί επίσης να είναι ιδιοκτησία, να είναι αντικείμενο διαχείρισης και λειτουργίας από την επιχείρηση, από έναν εξωτερικό οργανισμό είτε από ένα συνδυασμό αυτών. Τέλος μπορεί να βρίσκεται εντός ή και εκτός από τις εγκαταστάσεις του οργανισμού αυτού.

#### 2.1.3.2 Κοινοτικό Νέφος (Community Cloud)

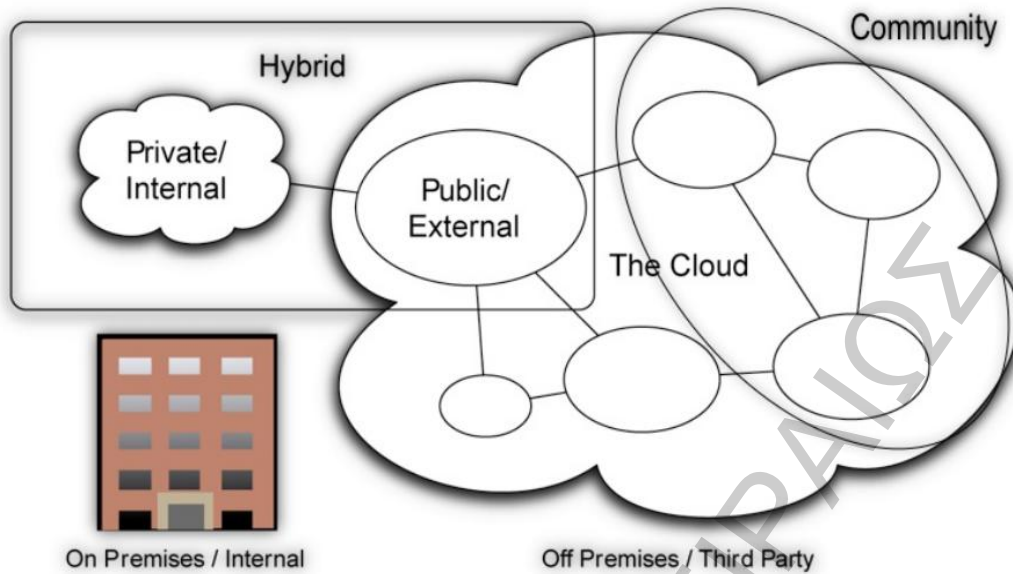
Η υποδομή του Νέφους παρέχεται για αποκλειστική χρήση από τους χρήστες ενός οργανισμού οι οποίοι έχουν κοινά ενδιαφέροντα και ανησυχίες. Μπορεί να είναι ιδιοκτησία και να τελεί υπό την διαχείριση ενός ή περισσότερων οργανισμών μιας κοινότητας, έναν εξωτερικό οργανισμό ή συνδυασμό αυτών. Επίσης, η υποδομή μπορεί να βρίσκεται εντός ή και εκτός των εγκαταστάσεων του οργανισμού αυτού.

#### 2.1.3.3 Δημόσιο Νέφος (Public Cloud)

Η υποδομή του Νέφους παρέχεται για δημόσια χρήση από το κοινό. Μπορεί να είναι ιδιοκτησία και να διαχειρίζεται από έναν επιχειρηματικό, ακαδημαϊκό, κυβερνητικό οργανισμό ή συνδυασμό των παραπάνω. Η υποδομή βρίσκεται στις εγκαταστάσεις του παρόχου αυτού. [6]

#### Υβριδικό Νέφος (Hybrid Cloud)

Η υποδομή του Νέφους αποτελείται από έναν συνδυασμό δύο ή περισσότερων μοντέλων Νεφών (ιδιωτικό, κοινοτικό ή δημόσιο) τα οποία υπάρχουν και λειτουργούν αυτόνομα αλλά είναι διασυνδεδεμένα μέσω τυποποιημένων ή ιδιόκτητων τεχνολογιών που επιτρέπουν την μεταφορά δεδομένων και εφαρμογών (πχ. ενεργοποίηση δευτερεύοντος Νέφους με σκοπό την εξισορρόπηση φόρτου εργασίας – Cloud bursting) [6].



Εικόνα 3. Μοντέλα ανάπτυξης.

## 2.2 Πάροχοι νεφών

### 2.2.1 Υποδομή σαν υπηρεσία

Στο σημείο αυτό θα επισημάνουμε τα δημοφιλέστερα συστήματα δημιουργίας και διαχείρισης Νεφών όπως και τις διεπαφές χρήσης (APIs) που παρέχουν. Εκτός από τους μεγάλους παρόχους εμπορικών Νεφών, έχουν περιγραφεί συστήματα ανοιχτού κώδικα που επιτρέπουν την δημιουργία και διαχείριση εικονικών υποδομών αλλά και την διαλειτουργικότητα με άλλα εμπορικά νέφη.

#### 2.2.1.1 Amazon EC2 API

Η Amazon θεωρείται ως ο πρωτοπόρος πάροχος υπηρεσιών Υπολογιστικού Νέφους στην παρούσα αγορά. Η σχετική υπηρεσία που παρέχει ονομάζεται Amazon Web Services (AWS) και παρουσιάστηκε αρχικά το 2002 [8]. Στην πράξη η AWS είναι ένα περιβάλλον διαδικτυακών υπηρεσιών οι οποίες στο σύνολό τους διαμορφώνουν την πλατφόρμα νέφους της Amazon. Η εταιρία παρέχει επίσης μια πληθώρα σχετικών και εξειδικευμένων υπηρεσιών όπως: Elastic Compute Cloud (Amazon EC2), [9][10] Simple Storage Service (S3)[11], Simple Queue Service (SQS)[12], CloudFront[13], Simple DB[14] και άλλες. Σε αυτό το κεφάλαιο δεν θα επεκταθούμε περισσότερο σε όλες υπηρεσίες και τα προϊόντα της Amazon αλλά θα επικεντρωθούμε στα βασικά χαρακτηριστικά των δύο βασικών υπηρεσιών Νέφους: EC2 (Υπολογιστικό Νέφος) και S3 (Νέφος αποθήκευσης δεδομένων).

Το EC2 είναι μια διαδικτυακή υπηρεσία η οποία επιτρέπει την εγκατάσταση και διαχείριση στιγμιότυπων εξυπηρετητών στα κέντρα δεδομένων και υπολογιστών της Amazon. Η υπηρεσία αυτή παρέχει συγκεκριμένες διεπαφές χρήσης και προγραμματισμού (API) έτσι ώστε κάποιος να μπορεί να ελέγχει τους υπολογιστικούς πόρους, τους περιορισμούς πρόσβασης και γενικά να μπορεί να δημιουργήσει ένα προσωποποιημένο περιβάλλον εφαρμογών. Εκτός από τον πλήρη έλεγχο που παρέχεται, ένα δεύτερο σημαντικό χαρακτηριστικό της EC2 υπηρεσίας είναι η επεκτασιμότητα (*scalability*). Όπως

και το ίδιο το όνομα της υπηρεσίας υπονοεί (EC2: Elastic Cloud), η χωρητικότητα του κάθε πόρου μπορεί να προσαρμοστεί δυναμικά, είτε χειροκίνητα, είτε αυτόματα από την εφαρμογή μέσω του παρεχόμενου API.

Η υπηρεσία S3 της Amazon παρέχει διαδικτυακό αποθηκευτικό χώρο που μπορεί να είναι προσβάσιμος από οποιοδήποτε, είτε άτομο είτε εφαρμογή στο διαδίκτυο. Η υπηρεσία αυτή εμφανίστηκε ως μια εναλλακτική λύση στα συμβατικά, τοπικά συστήματα αποθήκευσης δεδομένων και αποτελεί μια από τις βασικότερες υπηρεσίες της συλλογής AWS της Amazon. Παρόμοια με το την EC, η S3 παρέχει ένα API μέσω του οποίου μπορεί κάποιος να αποθηκεύσει και επανακτήσει δεδομένα από τα κέντρα δεδομένων και υπολογιστών της Amazon. Επιπρόσθετα, η εταιρία εμφανίζεται να διαφημίζει ότι μπορεί να παρέχει 99.99% διαθεσιμότητα και αξιοπιστία στα αποθηκευμένα δεδομένα, όπως αυτό περιγράφεται και στο συμβόλαιο υπηρεσίας (Amazon S3 Service Level Agreement).

Για να κατανοήσουμε καλύτερα την EC2 υπηρεσία, θα εξηγήσουμε σε αυτό το σημείο την ορολογία που η Amazon χρησιμοποιεί για την προδιαγραφή του σχετικού API. Στιγμιότυπο (*instance*) είναι ένας εικονικός εξυπηρετητής που εκτελείται σε ένα φυσικό πόρο στις εγκαταστάσεις της Amazon. Τα πρότυπα πάνω στα οποία η δημιουργία των στιγμιότυπων βασίζεται ονομάζονται Εικόνες Μηχανής της Amazon (*Amazon Machine Images - AMIs*) και συμπεριλαμβάνουν επιλογές του λειτουργικού συστήματος ή άλλες ρυθμίσεις που μπορεί κάποιος να επιλέξει για να προσδιορίσει το συγκεκριμένο στιγμιότυπο που θέλει να χρησιμοποιήσει.

Για να έχει κάποιος πρόσβαση σε ένα στιγμιότυπο, μπορεί να χρησιμοποιήσει τις ελαστικές IP διευθύνσεις της Amazon (*Amazon's Elastic IP Addresses*) οι οποίες είναι στατικές IP διευθύνσεις σχεδιασμένες για δυναμικά υπολογιστικά Νέφη. Αυτή η διεύθυνση είναι συσχετισμένη με το λογαριασμό του στην Amazon και μπορεί να επαναπροσδιοριστεί και να συνδεθεί με διαφορετικό στιγμιότυπο. *Elastic Block Storage (EBS)* είναι ο αποθηκευτικός χώρος για κάθε EC2 στιγμιότυπο στον οποίο κρατείται η κατάσταση (*state*) του κάθε στιγμιότυπου.

Επιπρόσθετα, μπορεί κάποιος να δημιουργήσει μια απεικόνιση της κατάστασης (*snapshot*) μιας συγκεκριμένης στιγμής η οποία θα αποθηκευθεί σε μια S3 υπηρεσία για μεγαλύτερη αντοχή στο χρόνο. Όπως παρουσιάζεται στο Σχήμα 11, η ροή εργασίας όταν χρησιμοποιούμε την EC2 ξεκινά από την αναζήτηση ενός διαθέσιμου δημόσια AMI και την προσαρμογή του στις ανάγκες μας είτε στην δημιουργία ενός από την αρχή. Το επόμενο βήμα είναι η δημιουργία ενός στιγμιότυπου του AMI χρησιμοποιώντας τις προγραμματιστικές διεπαφές (API) της EC2. Το αποτέλεσμα αυτής της διαδικασίας είναι ένα AMI ID που μπορεί να χρησιμοποιηθεί έτσι ώστε να εκτελέσουμε όσα στιγμιότυπα θέλουμε από το επιλεγμένο AMI. Τέλος, μέσω των παρεχομένων από το API εργαλείων, μπορούμε να διαχειριστούμε και να χρησιμοποιήσουμε τα στιγμιότυπα όπως επιθυμούμε σε οποιονδήποτε εξυπηρετητή.

Το API της Amazon EC2 παρέχει πρόσβαση στην διαδικτυακή υπηρεσία είτε μέσω SOAP API είτε μέσω Query API. Στην πρώτη περίπτωση, οι διεπαφές ορίζονται μέσω ενός XML κειμένου περιγραφής υπηρεσίας (*Web Service Description Language - WSDL*). Αφού τα αιτήματα και οι απαντήσεις σε SOAP, στην Amazon EC2, ακολουθούν τα προδιαγεγραμμένα πρότυπα οποιασδήποτε γλώσσας προγραμματισμού (π.χ. Java, C++, C#, Python, Perl και Ruby) η οποία μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών που θα εκτελούνται στο Νέφος της Amazon. Το Query API είναι μια διεπαφή βασισμένη στο μοντέλο REST [15]

που υποστηρίζει λειτουργίες GET και POST κατά την πραγματοποίηση αιτημάτων. Η τεχνολογία αυτή φαίνεται να προτιμάται από τους προγραμματιστές της Amazon και παρέχεται σχεδόν από όλες τις υπηρεσίες AWS.

Πίνακας 1. Χαρακτηριστικά Amazon EC2[16],[17]

Τύπος Υποδομής σαν υπηρεσία	Το Amazon EC2 είναι μία εμπορική υποδομή που λειτουργεί εξολοκλήρου στην υποδομή της Amazon. Ο χρήστης μπορεί να διαχειριστεί εικονικά μηχανήματα, αποθηκευτικό χώρο και αρχεία.
Χαρακτηριστικά διαχείρισης εικονικών μηχανημάτων	Τα εικονικά μηχανήματα λειτουργούν πάνω από XEN. Τα εικονικά περιφερειακά (δίκτυο, δίσκοι, κονσόλα) χρησιμοποιούν τους οδηγούς του XEN paravirtualized.
Απομακρυσμένη διαχείριση	Η προγραμματιστική διεπαφή (API) υποστηρίζει την δημιουργία, επανεκκίνηση και την διαγραφή εικονικών πόρων. Επίσης ένα αρχείο καταγραφής (log) μπορεί να δοθεί από το API.
Υποστήριξη εικονικών λειτουργικών προγραμμάτων (images)	Υπάρχουν έτοιμα εικονικά λειτουργικά προγράμματα όπως Windows ή Linux που προσφέρονται από την Amazon ή από τους συνεργάτες της. Επίσης δίνεται η δυνατότητα να προσθέτουν εικονικά λειτουργικά προγράμματα από τον χρήστη.
Χαρακτηριστικά (Scalability)	Η αυτόματη επεκτασιμότητα μπορεί να ενεργοποιηθεί βασιζόμενη σε μετρικούς δείκτες της υποδομής. Ο χρήστης δημιουργεί μία ομάδα μηχανών αυτόματης επεκτασιμότητας και καθορίζει τις παραμέτρους που θα την ενεργοποιήσουν.

### 2.2.1.2 Rackspace cloud

Η Rackspace είναι ένας εμπορικός πάροχος υποδομής σαν υπηρεσία από το 2006 το οποίο την κάνει να είναι από τους πρώτους παρόχους υποδομής σαν υπηρεσία. Η Rackspace έχει υποδομή μόνο στην Αμερική και σχεδιάζει να δημιουργήσει και στην Ευρώπη (UK). Υπάρχουν τρεις βασικές υπηρεσίες που προσφέρει

- Cloudservers μία υποδομή για εικονικά μηχανήματα [18]
- Cloudfiles ένα σύστημα αποθήκευσης βασισμένο σε αρχεία [19]
- Cloudsites είναι μία πλατφόρμα που προσφέρεται σαν υπηρεσία για την φιλοξενία ιστοσελίδων

Πίνακας 2. Χαρακτηριστικά Rackspace cloud

Τύπος Υποδομής σαν υπηρεσία	Το Amazon EC2 είναι μία εμπορική υποδομή που λειτουργεί εξολοκλήρου στην υποδομή
-----------------------------	--

		της Rackspace. Ο χρήστης μπορεί να διαχειριστεί εικονικά μηχανήματα, αποθηκευτικό χώρο και αρχεία.
Χαρακτηριστικά διαχείρισης εικονικών μηχανημάτων		Τα εικονικά μηχανήματα λειτουργούν πάνω από XEN. Τα εικονικά περιφερειακά (δίκτυο, δίσκοι, κονσόλα) χρησιμοποιούν τους οδηγούς του XEN paravirtualized.
Απομακρυσμένη διαχείριση		Η προγραμματιστική διεπαφή (API) [20] υποστηρίζει την δημιουργία, επανεκκίνηση και την διαγραφή εικονικών πόρων.
Υποστήριξη εικονικών λειτουργικών προγραμμάτων (images)		Υπάρχουν έτοιμα εικονικά λειτουργικά προγράμματα όπως Windows ή Linux που προσφέρονται από την Amazon ή από τους συνεργάτες της. Επίσης δίνεται η δυνατότητα να προστεθούν εικονικά λειτουργικά προγράμματα από τον χρήστη.
Χαρακτηριστικά (Scalability)	επεκτασιμότητας	Δεν υπάρχει δυνατότητα για αυτόματη επεκτασιμότητα.

### 2.2.1.3 VMware vCloud

Η VMware ως πρωτοπόρος εταιρία στις τεχνολογίες εικονικοποίησης (*virtualization*), εξέδωσε το 2009 το προϊόν vCloud [21], μια πλατφόρμα Υπολογιστικού Νέφους βασισμένο στο πρότυπο OVF 1.0 ως το προγραμματιστικό περιβάλλον διεπαφής (API) για την διαχείριση εικονικών πόρων στο Νέφος. Το API του vCloud [22] ήταν αποτέλεσμα συνδυαστικής προσπάθειας της VMware και των συνεργατών της με σκοπό να παραδώσουν μια εύκολη στη χρήση διεπαφή χρήσης υπηρεσιών Νέφους, επεκτάσιμο και βασισμένο σε διάφορα καθιερωμένα πρότυπα όπως XML, HTTP, OVF κ.α. Πιο λεπτομερώς, το vCloud API μπορεί να διαχωριστεί σε δύο μέρη: το διαχειριστικό (*Administrative*) API και το API χρηστών (*Users API*). Το πρώτο χρησιμοποιείται για την δημιουργία, διαχείριση και παρακολούθηση πόρων, χρηστών και ρόλων σε ένα περιβάλλον vCloud, ενώ το δεύτερο παρέχει λειτουργίες περιήγησης και αναζήτησης πόρων όπως και δημιουργίας, μετατροπής και εγκατάστασης λειτουργικών εικονικών συσκευών. Το VMware vCloud API επιτρέπει στους προγραμματιστές εφαρμογών να δημιουργούν προγράμματα-πελάτες για υπηρεσίες vCloud εκμεταλλευόμενοι το *RESTful* προγραμματιστικό παράδειγμα. Έτσι, οι εφαρμογές αυτές επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματα σε XML που αναπαριστούν οντότητες του vCloud.

Πίνακας 3. Χαρακτηριστικά VMwarevCloud

Τύπος Υποδομής σαν υπηρεσία		Το VMware vCloud είναι μία εμπορική υποδομή που επιτρέπει στους χρήστες να δημιουργήσουν ιδιωτικά υπολογιστικά νέφη στη δική τους υπάρχουσα υποδομή.
Χαρακτηριστικά διαχείρισης εικονικών μηχανημάτων		Ο διαχειριστής του οργανισμού μπορεί να δημιουργήσει διάφορα σύνολα από εικονικά μηχανήματα με συγκεκριμένες δυνατότητες ανάλογα με τις ανάγκες του οργανισμού.



Απομακρυσμένη διαχείριση	Υποστηρίζεται ο πλήρης έλεγχος των εικονικών μηχανημάτων.
Υποστήριξη εικονικών λειτουργικών προγραμμάτων (images)	Τα εικονικά λειτουργικά προγράμματα καθορίζονται από τον οργανισμό.
Χαρακτηριστικά επεκτασιμότητας (Scalability)	Δεν υπάρχει δυνατότητα για αυτόματη επεκτασιμότητα.

#### 2.2.1.4 Eucalyptus

Το Eucalyptus ήταν μια από τις πρώτες εφαρμογές ανοιχτού κώδικα που επικεντρώθηκε στην δημιουργία IaaS Νεφών. Δημιουργήθηκε έτσι ώστε να παρέχει λειτουργίες παρόμοιες με το Amazon Web Services API αλλά διατίθεται ως εφαρμογή open source. Οι χρήστες μπορούν να αλληλεπιδρούν με το Νέφος του Eucalyptus χρησιμοποιώντας τα ίδια εργαλεία που χρησιμοποιούν για να έχουν πρόσβαση στο Amazon EC2. Επιπλέον όμως, παρέχεται ένα Νέφος αποθήκευσης API για την αποθήκευση των γενικών δεδομένων των χρηστών και των VM εικόνων. Συνοψίζοντας, το Eucalyptus παρέχει τα ακόλουθα συστατικά:

- Linux-based controller
- EC2-compatible (SOAP, Query) , S3-compatible (SOAP, REST) CLI και Web portal interfaces
- Xen, KVM, και VMWare backends
- Amazon EBS-compatible virtual storage devices
- Διεπαφή προς το Amazon EC2 public cloud
- Εικονικά δίκτυα (virtual networks)

Η αρχιτεκτονική του Eucalyptus, αποτελεί συστατικό στοιχείο κάθε συστήματος σε υψηλό επίπεδο ως αυτόνομη υπηρεσία Web με τα ακόλουθα στοιχεία ελέγχου:

Node controller (NC): Ελέγχει την εκτέλεση, επιθεώρηση και τον τερματισμό των στιγμιότυπων των εικονικών μηχανών (VMs) στον χώρο που φιλοξενούνται και εκτελούνται.

Cluster controller (CC): Συλλέγουν πληροφορίες σχετικά με τις εκτελέσεις των VMs αλλά και τις προγραμματίζουν σε συγκεκριμένους node controllers (NC), τόσο καλά όσο διαχειρίζονται τις περιπτώσεις εικονικών δικτύων.

Storage controller (SC): Είναι μια υπηρεσία λήψης/αποθήκευσης που εφαρμόζει την διεπαφή του Amazon S3 και παρέχει τον τρόπο αποθήκευσης και πρόσβασης της πληροφορίας που έχει χρησιμοποιηθεί από τον χρήστη.

Cloud controller (CLC): Είναι το σημείο εισόδου για το Νέφος για απλούς χρήστες και διαχειριστές. Θέτει ερωτήματα στους διαχειριστές κόμβων για πληροφορίες και πόρους, παίρνει αποφάσεις προγραμματισμού ενεργειών υψηλού επιπέδου, και εφαρμόζει όλα αυτά κάνοντας αιτήματα (*requests*) στους ελεγκτές συμπλεγμάτων (*cluster controllers*).

**Walrus (W):** Είναι το εξάρτημα του ελεγκτή που διαχειρίζεται την πρόσβαση στις υπηρεσίες αποθήκευσης μέσα στον Eucalyptus. Τα αιτήματα αποστέλλονται στον Walrus χρησιμοποιώντας διεπαφές SOAP ή REST.

Πίνακας 4. Χαρακτηριστικά Eucalyptus [23]

Τύπος Υποδομής σαν υπηρεσία	Το Eucalyptus είναι ένα λογισμικό που επιτρέπει την διαχώριση περιβαλλόντων υπολογιστικών νεφών. Μπορεί είτε να εγκατασταθεί είτε να προσφερθεί από κάποιον πάροχο. Μπορεί να διαχειριστεί είτε παραδοσιακό hardware είτε πόρους που παρέχονται από έναν εξωτερικό πάροχο όπως η Amazon
Χαρακτηριστικά διαχείρισης εικονικών μηχανημάτων	Η εικονοποίηση παρέχεται είτε από XEN είτε από KVM. Επίσης η Enterprise έκδοση επιτρέπει και την χρήση διάφορων τεχνολογιών VMware.
Απομακρυσμένη διαχείριση	Τα εικονικά μηχανήματα μπορούν να είναι προσβάσιμα μόνο από τεχνολογίες του λειτουργικού τους προγράμματος όπως ssh. Το Eucalyptus υποστηρίζει μόνο την εκκίνηση, την διαγραφή και την επανεκκίνηση των μηχανημάτων.
Υποστήριξη εικονικών λειτουργικών προγραμμάτων (images)	Οι χρήστες μπορούν να δημιουργήσουν τα δικά τους εικονικά λειτουργικά συστήματα.
Χαρακτηριστικά επεκτασιμότητας (Scalability)	Δεν υπάρχει δυνατότητα για αυτόματη επεκτασιμότητα.

### 2.2.1.5 Open Nebula

Το Open Nebula είναι μία από τις πιο πλούσιες εφαρμογές ανοιχτού κώδικα για υλοποίηση IaaS. Αρχικά είχε σχεδιαστεί για την διαχείριση εικονικών υποδομών και περιλάμβανε απομακρυσμένες διεπαφές που καθιστούσαν υλοποιήσιμη την κατασκευή δημόσιων Νεφών. Συνολικά τέσσερα APIs είναι διαθέσιμα:

- XML-RPC
- Libvirt
- EC2 (Query) APIs
- OpenNebula Cloud API (OCA)

Η αρχιτεκτονική του περιλαμβάνει διάφορα εξειδικευμένα δομικά συστατικά. Η κύρια ενότητα της αρχιτεκτονικής του περιλαμβάνει φυσικούς εξυπηρετητές και τα *hypervisors* τους, τους κόμβους αποθήκευσης και τα στρώμα δικτύου. Η διαχείριση των εργασιών πραγματοποιείται από οδηγούς οι οποίοι αλληλεπιδρούν με τα APIs των *hypervisors*, με τις συσκευές αποθήκευσης και τις τεχνολογίες δικτύων των δημόσιων Νεφών. Συνοψίζοντας, το Open Nebula έχει τις ακόλουθες δυνατότητες:

- CLI, XML-RPC, EC2-compatible Query και OCA interfaces
- Xen, KVM, και VMware backend

- Διεπαφές σε δημόσια Νέφη (AmazonEC2, ElasticHosts)
- Εικονικά δίκτυα
- Δυναμική ανάθεση πόρων
- Προληπτική δέσμευση πόρων

Δικτύωση: Γενικά, οι υπηρεσίες που αναπτύσσονται στο Νέφος, από μια συστάδα υπολογιστών (*cluster*) προς την κλασική three-tier επαγγελματική εφαρμογή, απαιτούνται πολλαπλές αλληλένδετες εικονικές μηχανές (VMs) με ένα εικονικό δίκτυο εφαρμογών (VAN) να είναι ο συνδεδεμένος κρίκος μεταξύ τους. Το Open Nebula δημιουργεί δυναμικά αυτά τα εικονικά δίκτυα εφαρμογών και ακολουθεί τις MAC διευθύνσεις που χρησιμοποιήθηκαν στο δίκτυο για τις υπηρεσίες των VMs.

Πίνακας 5. Χαρακτηριστικά Open Nebula [24][25].

Τύπος Υποδομής σαν υπηρεσία	Ιδιωτικό , δημόσιο, υβριδικό και ομοσπονδιακό
Χαρακτηριστικά διαχείρισης εικονικών μηχανημάτων	Η εικονοποίηση παρέχεται είτε από XEN είτε από KVM. Επίσης η Enterprise έκδοση επιτρέπει και την χρήση διάφορων τεχνολογιών VMware.
Απομακρυσμένη διαχείριση	Τα εικονικά μηχανήματα μπορούν να είναι προσβάσιμα μόνο από τεχνολογίες του λειτουργικού τους προγράμματος όπως ssh. Το Eucalyptus υποστηρίζει μόνο την εκκίνηση την διαγραφή και την επανεκκίνηση των μηχανημάτων.
Υποστήριξη εικονικών λειτουργικών προγραμμάτων (images)	Οι χρήστες μπορούν να δημιουργήσουν τα δικά τους εικονικά λειτουργικά συστήματα.
Χαρακτηριστικά επεκτασιμότητας (Scalability)	Δεν υπάρχει δυνατότητα για αυτόματη επεκτασιμότητα.

### 2.2.1.6 Open Stack

Το open stack είναι ένα ανοιχτού κώδικα έργο παροχής υποδομής σαν υπηρεσία που ιδρύθηκε από την Rackspace και την NASA το 2010 και τώρα έχει πάνω από σαράντα μέλη. Υπάρχουν δύο υποέργα

- NOVA για υπολογιστικούς πόρους (εικονικά μηχανήματα, δίκτυο) [26]
- Swift για αποθήκευση αρχείων [27]

Αρχικά ο κώδικας του Swift ήταν μία ανοιχτού κώδικα έκδοση της μηχανής της υπηρεσίας Rackspace cloudfiles. Το Nova ξεκίνησε σαν μία ολοκλήρωση του μεταξύ του NASA nebula και του Rackspace cloudserver. Και τα δύο έργα χρησιμοποιούν python γλώσσα προγραμματισμού.



Πίνακας 6. Χαρακτηριστικά Open Stack [28].

Τύπος Υποδομής σαν υπηρεσία	Nova και Swift είναι δωρεάν υπό τους όρους του apache (apache license 2.0). Το Nova και το swift έχουν σχεδιαστεί ώστε να είναι πλήρως επεκτάσιμα.		
Χαρακτηριστικά μηχανημάτων	διαχείρισης	εικονικών	Το Nova υποστηρίζει <ul style="list-style-type: none"> <li>• Xen (απευθείας μέσω του XenAPI)</li> <li>• Qemu (μέσω libvirt)</li> <li>• KVM (μέσω libvirt)</li> <li>• UML (μέσω libvirt)</li> </ul>
Απομακρυσμένη διαχείριση	Οι βασικές καταστάσεις των μηχανημάτων μπορούν να ελεγχθούν μέσω API		
Υποστήριξη προγραμμάτων (images)	εικονικών	λειτουργικών	Οι χρήστες μπορούν να δημιουργήσουν τα δικά τους εικονικά λειτουργικά συστήματα.
Χαρακτηριστικά (Scalability)	επεκτασιμότητας		Δεν υπάρχει δυνατότητα για αυτόματη επεκτασιμότητα.

## 2.2.2 Πλατφόρμα σαν υπηρεσία

### 2.2.2.1 Google App Engine

Η είσοδος της Google στον χώρο των τεχνολογιών υπολογιστικού Νέφους έγινε μέσω της υπηρεσίας Google App Engine [29]. Σε αντίθεση με άλλους παρόχους και λύσεις οι οποίες υλοποιούν το παράδειγμα IaaS (π.χ. Amazon AWS), η υπηρεσία App Engine είναι ένα σύστημα Πλατφόρμα-ως-Υπηρεσία (PaaS). Σε αυτή τη βάση, η Google App Engine είναι μια πλατφόρμα μέσω της οποίας ένας προγραμματιστής μπορεί να εγκαταστήσει και εκτελέσει μία διαδικτυακή εφαρμογή στα κέντρα δεδομένων της Google. Αυτή τη στιγμή η πλατφόρμα υποστηρίζει ανάπτυξη εφαρμογών γραμμένες σε Java [30] και Python [31] άλλες γλώσσες βασισμένες σε JVM όπως Groovy, JRuby και Scala. Εκτός από το περιβάλλον ανάπτυξης (SDK) που παρέχεται, υπάρχουν διαθέσιμες προεκτάσεις προγραμμάτων (*plugins*) για σουίτες όπως το Eclipse. Η διαχείριση δεδομένων στην πλατφόρμα Νέφους της Google υλοποιείται μέσω του Datastore API. Στην περίπτωση ανάπτυξης εφαρμογών σε Java, το Datastore API αποθηκεύει και εκτελεί ερωτήματα (*queries*) πάνω σε αντικείμενα δεδομένων γνωστά ως *οντότητες*. Κάθε οντότητα έχει ένα μοναδικό χαρακτηριστικό κωδικό (*key*) και μια ή περισσότερες ιδιότητες, ως τιμές συγκεκριμένων τύπων δεδομένων. Το Datastore υποστηρίζει τις Java Data Objects (JDO) 2.3 και Java Persistent API (JPA) 1.0 πρότυπες διεπαφές.

Στην περίπτωση ανάπτυξης εφαρμογών με την χρήση Python, το Datastore υλοποιείται μέσω μιας γλώσσας παρόμοιας με SQL η οποία ονομάζεται QGL. Η γλώσσα αυτή δεν υποστηρίζει δηλώσεις Ένωσης (*Join*) καθώς είναι ανέφικτη η εφαρμογή τέτοιων λειτουργιών όταν τα δεδομένα είναι τοποθετημένα σε πολλαπλά μηχανήματα. Υπό αυτές τις συνθήκες, η γλώσσα αυτή δεν είναι μια σχεσιακή βάση δεδομένων αλλά παρόμοιες λειτουργίες μπορούν να υλοποιηθούν μέσω συγκεκριμένων μηχανισμών που παρέχονται από το API της QGL.

Όπως αναφέραμε, η Google App Engine πλατφόρμα παρέχει ένα API και συνολικά ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών είτε μέσω Java είτε μέσω Python. Το

περιβάλλον αυτό προσομοιώνει όλες τις λειτουργίες της Google App Engine και σου δίνει την δυνατότητα να εγκαταστήσεις κατευθείαν την εφαρμογή στο περιβάλλον Νέφους. Η όλη διαδικασία περιγράφεται με τα παρακάτω βήματα:

- Ανάπτυξη της εφαρμογής χρησιμοποιώντας τα διαθέσιμα εργαλεία και APIs
- Καταχώρηση ενός Κωδικού Εφαρμογής (Application ID) στην υπηρεσία Google App Engine μέσω της Κονσόλας Διαχείρισης (Administration Console)
- Μεταφόρτωση των αρχείων της εφαρμογής
- Πρόσβαση στην εφαρμογή μέσω του διαδικτύου, χρησιμοποιώντας συγκεκριμένη διεύθυνση βασισμένη στον Κωδικό Εφαρμογής

Η Google παρέχει δωρεάν χώρο χρήσης στους εξυπηρετητές και κέντρα δεδομένων της, με κάποιους περιορισμούς. Οι προγραμματιστές μπορούν πάντα να επεκτείνουν την χρήση ενεργοποιώντας την χρέωση στον λογαριασμό τους. Ενδεικτικά, στον παρακάτω πίνακα (Πίνακας 7) παρουσιάζονται τα χαρακτηριστικά δωρεάν χρήσης.

Πίνακας 7 Google App Engine free quota

Quota	Limit
Apps per developer	10
Time per request	30 sec
Blobstore size (total file size per app)	1 GB
HTTP response size	10 MB
Datastore item size	1 MB
Application code size	150 MB

#### 2.2.2.2 Azure (.NET)

Η γρήγορη εξέλιξη του Υπολογιστικού Νέφους και των σχετικών τεχνολογιών δεν θα μπορούσε να αφήσει την Microsoft εκτός του κλάδου αυτού. Ως αποτέλεσμα, η Microsoft εξέδωσε την λύση Windows Azure [32] ως ένα λειτουργικό σύστημα που παρέχει υπηρεσίες Νέφους. Στην πραγματικότητα το Azure είναι μια πλατφόρμα υπηρεσιών που επιτρέπει την ανάπτυξη, εγκατάσταση και εκτέλεση εφαρμογών Windows στα κέντρα δεδομένων και υπολογιστών της Microsoft. Οι προγραμματιστές μπορούν να αναπτύξουν τις εφαρμογές χρησιμοποιώντας τις συνηθισμένες γλώσσες προγραμματισμού των Windows (C#, C++, VMκλπ) ή και άλλες γλώσσες που υποστηρίζονται (π.χ. Java, Ruby, PHP, Python) μέσω της σουίτας Microsoft Visual Studio.

Η πλατφόρμα Azure αποτελείται από τρία βασικά στοιχεία: Υπηρεσία Υπολογισμού (Compute Service), Υπηρεσία Αποθήκευσης (Storage Service) και Στρώμα Εφαρμογών (Application Fabric). Το πρώτο παρέχει τις απαραίτητες διεπαφές και υποστήριξη για την εκτέλεση των εφαρμογών οι οποίες μπορούν να έχουν πολλαπλά εγκατεστημένα στιγμιότυπα. Όλες οι εφαρμογές μπορούν να έχουν πρόσβαση σε πόρους δεδομένων κάνοντας χρήση της Υπηρεσίας Αποθήκευσης μέσω διεπαφών REST. Η υπηρεσία αυτή παρέχει αντικείμενα BLOB για την αποθήκευση μεγάλων δυαδικών αντικειμένων, πινάκων και ουρών για την διαχείριση δεδομένων. Για εφαρμογές πιο απαιτητικές όσον αφορά την διαχείριση δεδομένων, η πλατφόρμα παρέχει το SQL Azure Database, ένα σύστημα Νέφους για διαχείριση δεδομένων (DBMS). Το σύστημα αυτό βασίζεται στο παλιότερο Microsoft SQL Server αλλά υποστηρίζει ένα διαχειριστικό περιβάλλον στο Νέφος. Η πρόσβαση των

δεδομένων μπορεί να πραγματοποιηθεί μέσω των διεπαφών ADO.NET ή άλλων διεπαφών πρόσβασης των Windows.

Η υπηρεσία για την υποστήριξη της υποδομής του Νέφους στο Windows Azure πραγματοποιείται μέσω του Στρώματος Εφαρμογών (*Application Fabric*). Κάθε εφαρμογή μπορεί να δημιουργήσει διεπαφές πρόσβασης (*endpoints*) χρησιμοποιώντας το εργαλείο Service Bus του Στρώματος Εφαρμογών έτσι ώστε να μπορούν να χρησιμοποιηθούν από άλλες εφαρμογές του Νέφους ή και ανεξάρτητες εφαρμογές. Η διασύνδεση ενός προγράμματος πελάτη REST προς μία εφαρμογή ελέγχεται από το εργαλείο *Access Control* του Στρώματος Εφαρμογών. Οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές είτε με ρόλο *Web* είτε με ρόλο *Worker*, και να προσδιορίσουν έτσι πόσα αντίγραφα στιγμιότυπων θέλουν να εκτελέσουν στις εικονικές μηχανές των Windows. Αυτές οι εικονικές μηχανές (VMs) δεν δημιουργούνται από τον προγραμματιστή αλλά παρέχονται από το Azure σύστημα (συγκεκριμένα το *hypervisor* υποσύστημα) το οποίο είναι ειδικά σχεδιασμένο για λειτουργία σε περιβάλλον Νέφους. Οι εφαρμογές *Web* είναι συνήθως υλοποιημένες σε ASP.NET περιβάλλον ενώ τα στιγμιότυπα *Worker* είναι εργασίες που αλληλεπιδρούν με τις διαδικτυακές εφαρμογές μέσω της Υπηρεσίας Αποθήκευσης.

### 2.3 Cross Platform Cloud APIs

Εκτός από τους παρόχους νεφών με συγκεκριμένα APIs υπάρχουν διεπαφές προγραμματισμού για Υπολογιστικά Νέφη ανεξάρτητα πλατφόρμας. Το Jclouds [33] είναι πλαίσιο ανάπτυξης εφαρμογών για Νέφη ανοιχτού κώδικα και βασισμένο στην Java. Χρησιμοποιώντας τις βιβλιοθήκες αυτού του συστήματος, κάποιος μπορεί να αναπτύξει προγράμματα και εφαρμογές συμβατά με διάφορους παρόχους Νέφους όπως Amazon, VMWare, Azure και άλλους. Το Deltacloud [34] είναι ένα επίσης API ανοιχτού κώδικα βασισμένο στο REST, διατίθεται από την RedHat και είναι συμβατό με το EC2, Rackspace, GoGRID και άλλους παρόχους. Στην ίδια λογική, το libcloud [35] είναι μια βιβλιοθήκη της Python, που επιτρέπει στις εφαρμογές να αλληλεπιδρούν με διαφορετικές υλοποιήσεις Υπολογιστικών Νεφών. Διατίθεται υπό την άδεια λογισμικού *Apache Software License* και εξυπηρετεί μέσω συγκεκριμένων οδηγών (*drivers*) τα περισσότερα εμπορικά Νέφη.

### 3 Ροές Εργασίας

Όπως έχει ήδη αναφερθεί, με βάση διεπισδυτικά πρότυπα διαδικτύου, το υπολογιστικό νέφος επιτρέπει την κοινή χρήση υπολογιστικών και πληροφοριακών πόρων εκτός των ορίων επιμέρους τμημάτων ή οργανισμών, με ασφαλή και εξαιρετικά αποτελεσματικό τρόπο. Το νέφος υποστηρίζει την κοινή χρήση, τη διασύνδεση και τη χρήση διαφόρων πόρων, ενσωματωμένων στο πλαίσιο ενός δυναμικού υπολογιστικού συστήματος.

Η διαχείριση των λειτουργιών της ροής εργασίας κάποιας εφαρμογής σε υπολογιστικού νέφους απαιτεί την ενορχήστρωση των προαναφερθέντων κατανεμημένων πόρων **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** Σε αυτό το πλαίσιο, η ροή εργασίας είναι σημαντικός παράγοντας για τη σύνθεση των εφαρμογών σε πλέγματα, προσφέροντας αρκετά πλεονεκτήματα **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, όπως:

- Δυνατότητα ανάπτυξης δυναμικών εφαρμογών με ενορχήστρωση κατανεμημένων πόρων
- Αξιοποίηση πόρων που βρίσκονται σε συγκεκριμένο τομέα με στόχο την αύξηση της απόδοσης ή τη μείωση των δαπανών εκτέλεσης
- Εκτέλεση πολλών απομακρυσμένων διαχειριστικών τομέων με στόχο την απόκτηση συγκεκριμένων δυνατοτήτων επεξεργασίας
- Ενοποίηση πολλών ομάδων που συμμετέχουν στη διαχείριση διαφορετικών τμημάτων της ροής εργασίας, ενισχύοντας έτσι τις συνεργασίες μεταξύ οργανισμών

#### 3.1 Ορισμοί

Καθώς η ροή εργασίας είναι μια ευρεία τεχνολογική έννοια στη συγκεκριμένη παράγραφο διευκρινίζεται η ορολογία που αφορά τους ορισμούς της ροής εργασίας που χρησιμοποιούνται στη συνέχεια της παρούσης διατριβής.

Όσο αφορά το γενικό ορισμό, η ομάδα WfMC (Workflow Management Coalition) δίνει τον εξής ορισμό [36]: *"Ροή εργασίας είναι η αυτοματοποίηση μιας επιχειρηματικής διαδικασίας, στο σύνολό της ή εν μέρει, κατά την οποία έγγραφα, πληροφορίες ή εργασίες μεταφέρονται μεταξύ των συμμετεχόντων προκειμένου να εκτελεστεί κάποια δράση, σύμφωνα με ένα σύνολο διαδικαστικών κανόνων."*

Διαφορετικά, η ροή εργασίας μπορεί να οριστεί ως η ενορχήστρωση ενός συνόλου δραστηριοτήτων με στόχο την επίτευξη ενός πολύπλοκου στόχου, ενώ στα περιβάλλοντα υπολογιστικού νέφους, η ροή εργασίας περιλαμβάνει διαδικασίες εφαρμογών, επιχειρηματικές διαδικασίες και διαδικασίες υποδομής [37]. Η ροή εργασίας είναι ένας αρχιτεκτονικά σημαντικός παράγοντας για τη δυναμική διαλειτουργικότητα και προσαρμογή σε διαφορετικά επιχειρηματικά μοντέλα, που μπορούν να συμπεριληφθούν στη γενική έννοια πολιτικές ροής εργασίας, και πλαίσια ανάπτυξης. Τα μοντέλα ροής εργασίας χρησιμοποιούνται προκειμένου να οριστεί μια ροή εργασίας, τόσο σε επίπεδο εργασιών όσο και σε επίπεδο δομής.

Η έρευνα που διεξάγεται για την τεχνολογία υπολογιστικού νέφους έχει διατυπώσει ακόμη δύο ορισμούς, που είναι προσανατολισμένοι κυρίως προς την επιστήμη των υπολογιστών:

Workflow είναι ένα σχέδιο αλληλεπίδρασης διάφορων επιχειρηματικών διεργασιών, χωρίς να αντιστοιχεί απαραίτητα σε ένα καθορισμένο σύνολο από επιχειρηματικές διεργασίες. Όλες αυτές οι αλληλεπιδράσεις μπορεί να είναι μεταξύ υπηρεσιών (services) που λειτουργούν σε ένα μόνο κέντρο δεδομένων ή μεταξύ ενός εύρους από διαφορετικές πλατφόρμες και υλοποιήσεις οπουδήποτε.

Grid Workflow είναι ένας εύκολος τρόπος δημιουργίας νέων υπηρεσιών συνδυάζοντας ήδη υπάρχουσες υπηρεσίες. Μια νέα υπηρεσία (service) μπορεί να δημιουργηθεί και να χρησιμοποιηθεί καταχωρώντας τον ορισμό ενός workflow σε μία workflow engine.

Στη συνέχεια και βάσει των παραπάνω ορισμών παρουσιάζονται οι κατηγορίες των ροών εργασίας.

### 3.2 Κατηγορίες

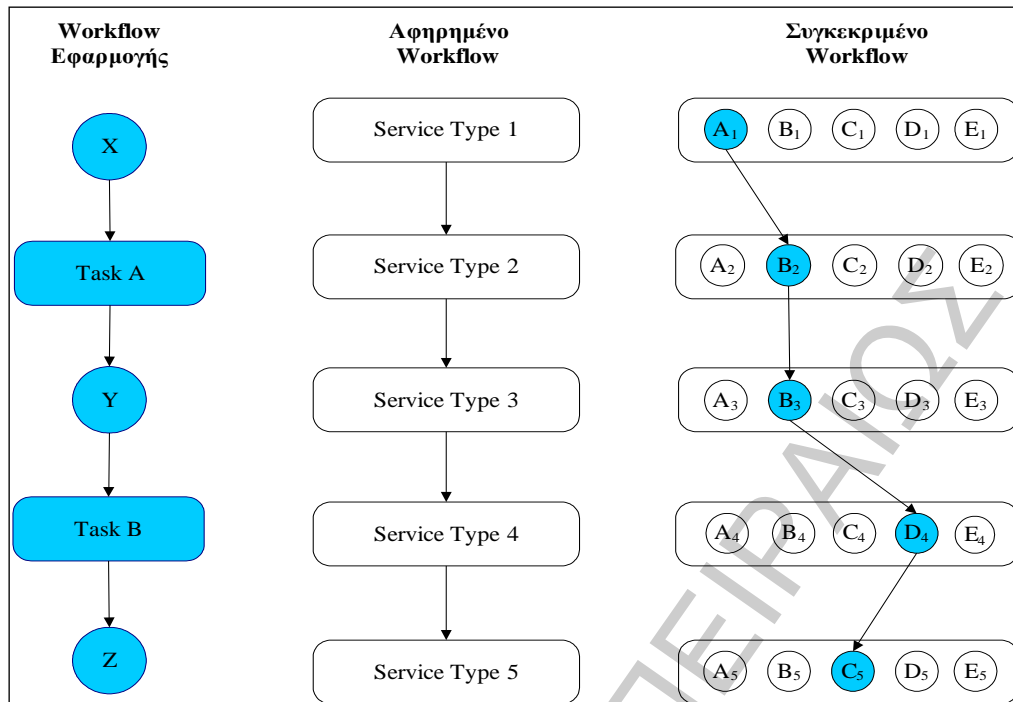
Οι ροές εργασίας παίζουν πολύ σημαντικό ρόλο στην αρχιτεκτονική του υπολογιστικού νέφους κυρίως όσον αφορά την ευέλικτη και δυναμική συνάθροιση των υπηρεσιών που προσφέρονται μέσω του υπολογιστικού νέφους. Χρησιμεύει στην αναπαράσταση και σύνθεση της λειτουργικής και επιχειρηματικής λογικής σε μια εφαρμογή.

Επιπρόσθετα οι ροές εργασίας χωρίζονται και ανάλογα με το αν είναι γνωστό πώς ακριβώς θα υλοποιηθούν οι διεργασίες που αυτά περιλαμβάνουν. Σε αυτή την περίπτωση έχουμε τις ακόλουθες δύο κατηγορίες - αφηρημένες (abstract) και συγκεκριμένες (concrete) **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε., Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.:**

Συγκεκριμένη ροή εργασίας (Concrete workflow) - αναφέρονται επίσης και ως εκτελέσιμες ροές εργασίας στη βιβλιογραφία [40] - ονομάζεται μία σύνθεση υπηρεσιών, η οποία παρέχει τόσο σημασιολογικές πληροφορίες όσο και πληροφορίες εκτέλεσης, οι οποίες αφορούν τη σύνθεση του workflow, τόσο από την πλευρά των υπηρεσιών, που το αποτελούν, όσο και από την πλευρά της πλήρους σύνθεσής του. Ένα συγκεκριμένο workflow μπορεί να θεωρηθεί ως ένα μονοπάτι, που προκύπτει από την επιλογή στιγμιοτύπων των υπηρεσιών έτσι ώστε να επιτευχθεί το workflow της εφαρμογής (application workflow).

Αφηρημένη ροή εργασίας (abstract workflow). Για να κατασκευαστεί ένα συγκεκριμένο workflow, συνδυάζεται το workflow εφαρμογής (application workflow) με τις παραμέτρους ποιότητας υπηρεσίας (QoS). Ένα ενδιάμεσο βήμα προς το στόχο αυτό αποτελεί η κατασκευή ενός αφηρημένου workflow, το οποίο προκύπτει ως η σύνθεση των υπηρεσιών που υλοποιούν την εφαρμογή. Παρέχονται όμως μόνο πληροφορίες για το είδος της υπηρεσίας που χρειάζεται σε κάθε βήμα καθώς και οι εισοδοί και έξοδοι αυτού. Δεν περιλαμβάνει πληροφορίες για το ποιός κόμβος συγκεκριμένα θα είναι αυτός που τελικά θα παρέχει την υπηρεσία.

Οι προαναφερθείσες κατηγορίες παρουσιάζονται στο σχήμα που ακολουθεί (Σχήμα 1):



Σχήμα 1: Ορισμοί ροής εργασίας

Αναλύοντας περισσότερο τις παραπάνω κατηγορίες, στα αφηρημένα μοντέλα, οι εργασίες περιγράφονται με μη συγκεκριμένη μορφή χωρίς να γίνεται αναφορά σε συγκεκριμένους πόρους πλέγματος για την εκτέλεση των εργασιών αυτών ενώ παράλληλα δίνεται η δυνατότητα στους χρήστες να ορίσουν ροές εργασίας με ευέλικτο τρόπο, απομονώνοντας τις λεπτομέρειες εκτέλεσης. Επίσης, τα αφηρημένα μοντέλα παρέχουν μόνο πληροφορίες για την έννοια των υπηρεσιών και σχετικά με τον τρόπο σύνθεσης της ροής εργασίας. Επομένως, είναι εφικτή η κοινή χρήση των περιγραφών της ροής εργασίας μεταξύ των χρηστών του πλέγματος, γεγονός που είναι κρίσιμης σημασίας για τους συμμετέχοντες σε εικονικούς οργανισμούς.

Στα συγκεκριμένα μοντέλα, οι εργασίες της ροής εργασίας αντιστοιχίζονται σε συγκεκριμένους πόρους και, επομένως, το μοντέλο παρέχει πληροφορίες σχετικά με την έννοια των υπηρεσιών και την εκτέλεση σχετικά με τον τρόπο σύνθεσης της ροής εργασίας, τόσο για τις υποστάσεις υπηρεσιών όσο και για τη συνολική σύνθεση (για παράδειγμα σχέσεις στη ροή δεδομένων, δομές ροής ελέγχου). Σε αντιστοιχία με τα αφηρημένα μοντέλα και σε σχέση με τους εικονικούς οργανισμούς, οι εργασίες που περιλαμβάνονται σε ένα συγκεκριμένο μοντέλο μπορεί επίσης να αναφέρονται σε αιτήματα μετακίνησης δεδομένων με στόχο τη δημοσιοποίηση των προσφάτως εξαγμένων δεδομένων στον εικονικό οργανισμό [38]. Πρέπει να διευκρινιστεί ότι οι υποστάσεις υπηρεσιών δεν αντιστοιχούν απαραίτητα σε πόρους του πλέγματος εφόσον, εντός κάποιου πόρου, μπορεί να υπάρχουν και να εκτελούνται περισσότερες από μια υποστάσεις υπηρεσιών.

Με βάση τους ορισμούς των μοντέλων ροής εργασίας, οι εργασίες στα αφηρημένα μοντέλα είναι φορητές και μπορούν να απεικονισθούν σε οποιαδήποτε κατάλληλη υπηρεσία πλέγματος με χρήση κατάλληλων μηχανισμών ανίχνευσης και απεικόνισης. Σε αυτή την κατεύθυνση, ένα από τα θέματα της παρούσης διατριβής είναι ένας νέος μηχανισμός απεικόνισης ροής εργασίας, το αποτέλεσμα του οποίου είναι μια συγκεκριμένη ροή



εργασίας. Η συγκεκριμένη ροή εργασίας μπορεί να αντιμετωπιστεί ως "διαδρομή", μια επιλογή υποστάσεων υπηρεσιών από τύπους υπηρεσιών με συγκεκριμένη σειρά προκειμένου να επιτευχθεί ο στόχος της εκτέλεσης της ροής εργασίας μιας συγκεκριμένης εφαρμογής.

### 3.3 Υπάρχοντα Συστήματα

Στην παρακάτω παράγραφο αναφέρονται μερικά συστήματα διαχείρισης ροών εργασίας.

Το Taverna είναι ένα σύστημα διαχείρισης ροών εργασίας το οποίο ακολουθεί κεντροποιημένη αρχιτεκτονική και θέτει ερωτήματα για την επεκτασιμότητα του συστήματος. Το Taverna υποστηρίζει υπηρεσίες διαδικτύου (web services) αλλά δεν παρέχει κάποιο μηχανισμό ποιότητας της υπηρεσίας (Quality of service- QoS). Παρά όλα αυτά το σύστημα παρέχει παρακολούθηση των ροών εργασίας που εκτελούνται και ένα φιλικό περιβάλλον χρήσης για την διαχείρισή τους. Επίσης υποστηρίζει και ανοχή λαθών επιτρέποντας την επανάληψη κάποιας υπηρεσίας που απέτυχε. [41]

Το Askalon επικεντρώνεται σε αποδοσοστρεφείς εφαρμογές. Το έργο ακολουθεί μία αποκεντροποιημένη αρχιτεκτονική αλλά με ένα κεντρικό μηχανισμό λήψης αποφάσεων. Οι χρήστες μπορούν να ορίσουν υψηλού επιπέδου περιορισμούς και ιδιότητες που προσδιορίζουν τον χρόνο εκτέλεσης και η ροή εργασίας προγραμματίζεται βάση την πρόβλεψη απόδοσης. Το Askalon παρέχει παρακολούθηση της ροής εργασίας που εκτελείται αλλά δεν επιτρέπει παρεμβάσεις σε αυτήν. Έλεγχος σημείων και τεχνικές μετάπτωσης χρησιμοποιούνται για ανοχή λαθών. [42]

Το σύστημα Amadeus ακολουθεί μία κεντροποιημένη αρχιτεκτονική. Παράμετροι ποιότητας και κόστους υποστηρίζονται με σκοπό να βρεθεί η βέλτιστη λύση, καθώς επιτρέπει και την χρήση SLAs για να επιτυγχάνεται συμφωνία μεταξύ του χρήστη και του παρόχου της υπηρεσίας. Δεν παρέχει καμία μορφή ανοχής λαθών και ελέγχου κατά την εκτέλεση της ροής εργασίας. [43]

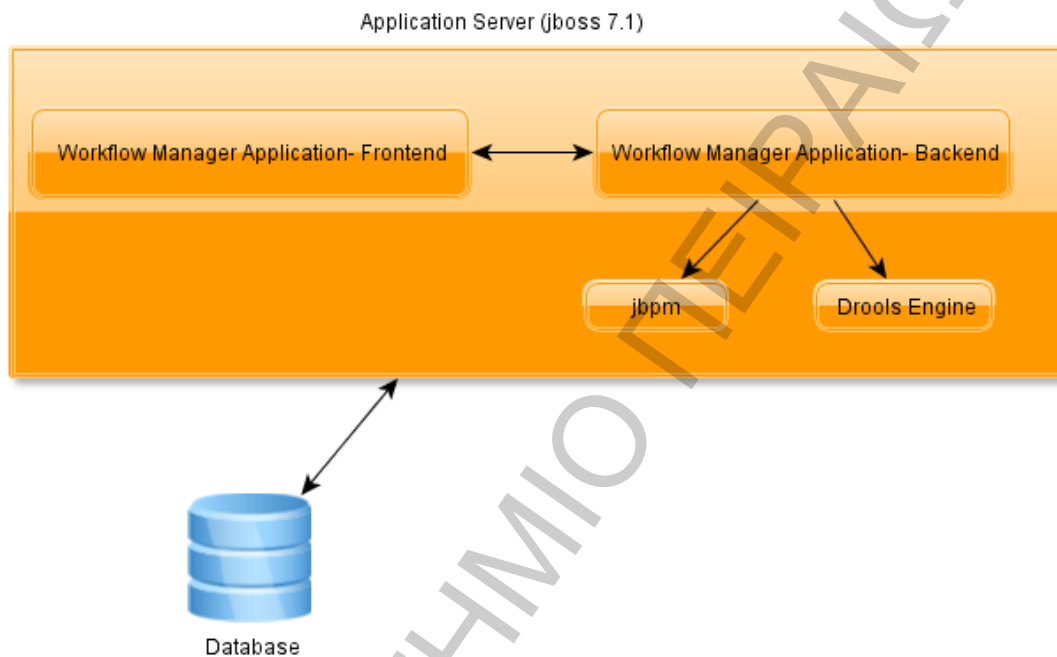
Το GrADS βασίζεται στο σύνολο εργαλείων του Globus (Globus toolkit) και στοχεύει σε εφαρμογές με μεγάλο φόρτο υπολογιστικών και επικοινωνιακών πόρων. Υποστηρίζει τον προσδιορισμό της ροής εργασίας η οποία αναλύεται και οι εξαρτήσεις μεταξύ των μερών προσδιορίζονται. Αυτό βοηθάει στην παραλληλοποίηση των εργασιών και αλγόριθμοι προγραμματισμού μπορούν να χρησιμοποιηθούν. Το GrADS ακόμη υποστηρίζει ποιότητα της υπηρεσίας μέσω πρόβλεψης του χρόνου εκτέλεσης της ροής εργασίας από την ανάλυση ιστορικών δεδομένων. [44]

Το Kepler σύστημα ροών εργασίας είναι ένα ανοιχτού κώδικα λογισμικό το οποίο επεκτείνει την εργασία του Ptolemy II [45] για να υποστηρίξει επιστημονικές εφαρμογές που χρησιμοποιούν προσέγγιση ροών δεδομένων. Το κύριο του χαρακτηριστικό βασίζεται σε επεξεργαστικά βήματα που ονομάζονται «actors» που έχουν καλά ορισμένες εισόδους και εξόδους. Ο χρήστης μπορεί να σχεδιάζει ροές εργασίας με το να επιλέγει κατάλληλους «actors» και να τους συνδέει μεταξύ τους μέσω ενός γραφικού περιβάλλοντος. Μία μονάδα που ονομάζεται «director» καθορίζει την εκτέλεση ολόκληρης της ροής εργασίας. Το σύστημα Kepler παρέχει διάφορους μηχανισμούς ανοχής λαθών με κυριότερο το ότι μπορεί ο χρήστης να καθορίσει «actors» για των έλεγχο εξαιρέσεων. [46]

## 4 Πλατφόρμα Διαχείρισης

### 4.1 Αρχιτεκτονική υλοποίησης

Η εφαρμογή διαχείρισης ροών εργασίας (Workflow Manager Application) υλοποιήθηκε σε πλατφόρμα Sun Java EE 6 , ο εξυπηρετητής εφαρμογών που χρησιμοποιήθηκε (application server) είναι jboss 7.1 και για την υλοποίηση χρησιμοποιήθηκαν επίσης τα υποσυστήματα Jboss drools (μηχανή κανόνων – rules engine) και jbpm ( σύστημα διαχείρισης επιχειρησιακών διαδικασιών).

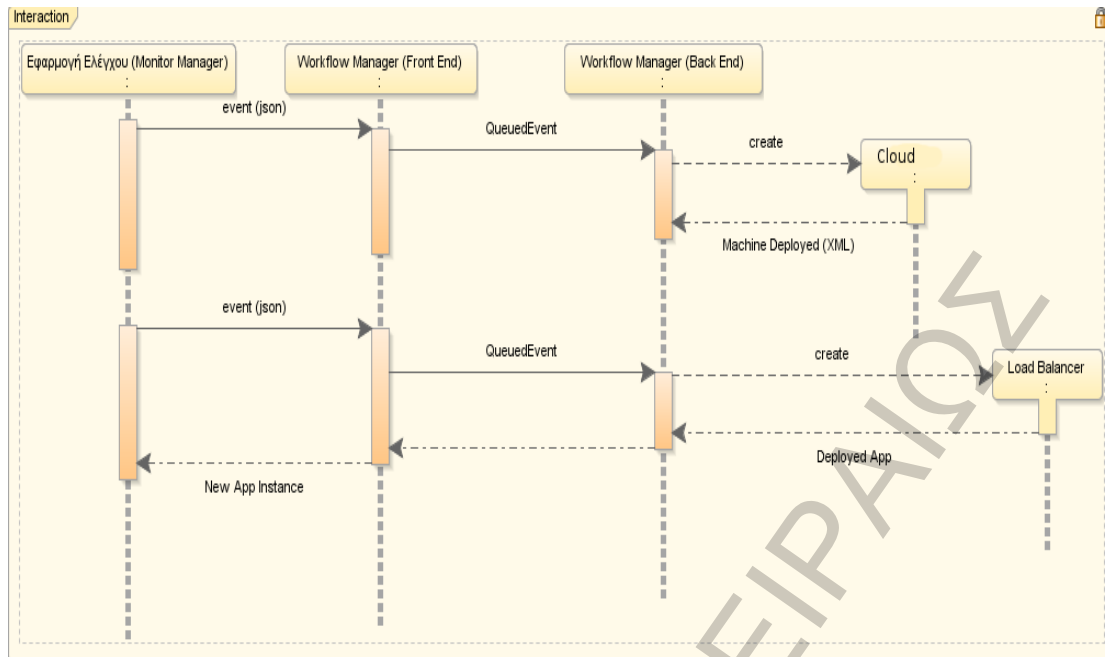


Εικόνα 4. Αρχιτεκτονική Πλατφόρμας

Η πλατφόρμα επεξεργάζεται ροές εργασίας ανάλογα με τις κλήσεις που δέχεται. Μία εξωτερική εφαρμογή (εφαρμογή ελέγχου) στέλνει αιτήματα και ανάλογα με το είδος του αιτήματος εκτελείται και η κατάλληλη ροή εργασίας. Στην παρακάτω εικόνα βλέπουμε την αλληλεπίδραση μεταξύ της εξωτερικής εφαρμογής , της πλατφόρμας , της υποδομής και της εφαρμογής που βλέπει ο χρήστης. Η επικοινωνία μεταξύ της εξωτερικής εφαρμογής ελέγχου και της πλατφόρμας γίνεται μέσω μηνυμάτων σε μορφή JSON (Javascript Object Notation)

Η επικοινωνία μεταξύ της πλατφόρμας και του περιβάλλοντος υπολογιστικού νέφους γίνεται με ανταλλαγή μηνυμάτων xml.





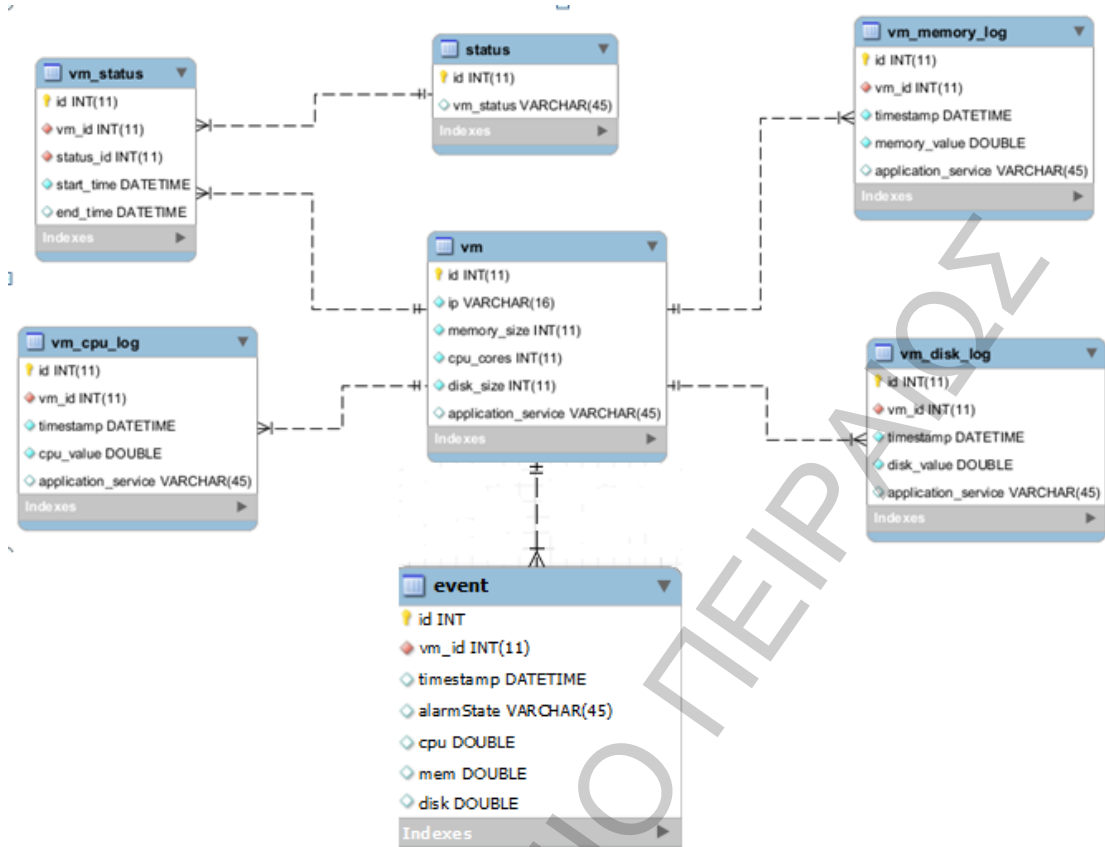
Εικόνα 5 Διάγραμμα αλληλεπίδρασης.

Όταν η εφαρμογή ελέγχου δημιουργήσει ένα αντικείμενο τύπου event το αποστέλλει στην front – end εφαρμογή της πλατφόρμας διαχείρισης ροών εργασίας η οποία το αποστέλλει στο κατάλληλο στιγμιότυπο της back – end εφαρμογής της πλατφόρμας διαχείρισης ροών εργασίας, ανάλογα με το σε ποία τοποθεσία στο υπολογιστικό νέφος βρίσκεται ο υπολογιστικός πόρος που δημιούργησε το event. Η back – end εφαρμογή επεξεργάζεται το event σύμφωνα με τους κανόνες που έχουν υλοποιηθεί και εκτελεί την ανάλογη ροή εργασίας. Μετά την ολοκλήρωση της ροής εργασίας ενημερώνει την front – end εφαρμογή η οποία με την σειρά της ενημερώνει την εφαρμογή ελέγχου.

#### 4.1.1 Κεντρική βάση δεδομένων

Η βάση δεδομένων διατηρεί όλη τη απαραίτητη πληροφορία, που χρειάζονται τόσο ο διαχειριστής ροής εργασιών, όσο και ο διαχειριστής ελέγχου και επίβλεψης. Η βάση δεδομένων διατρέχεται από όλα τα αρχιτεκτονικά τμήματα του μηχανισμού ελέγχου και επίβλεψης. Διατηρεί πληροφορία σχετικά με:

- Τη ταυτότητα των εικονικών μηχανημάτων και των τεχνικών χαρακτηριστικών τους.
- Τη κατάσταση των μηχανημάτων (ενεργά/ανενεργά).
- Το ιστορικό των μετρήσεων όλων των εικονικών μηχανημάτων για κάθε παράμετρο ξεχωριστά (CPU, RAM, disk)



Εικόνα 6. Κεντρική βάση δεδομένων

Παρακάτω αναλύονται τα πεδία κάθε πίνακα ξεχωριστά και τι είδους πληροφορία διατηρούν:

#### Event

Ο πίνακας που αποθηκεύει τα events που δημιουργούνται από την εφαρμογή ελέγχου και λαμβάνονται από την πλατφόρμα. Ο πίνακας αποθηκεύει πληροφορία για το ποιό μηχάνημα δημιούργησε το event, την χρονική στιγμή που το δημιούργησε καθώς και για το είδος του event. Επίσης, αποθηκεύει πληροφορίες και για τις τιμές του επεξεργαστή, της μνήμης και του δίσκου του μηχανήματος.

#### Status

Ένας παραμετρικός πίνακας που περιγράφει τις 2 πιθανές καταστάσεις των εικονικών μηχανημάτων. Περιέχει το πεδίο κλειδί id. Το κύριο πεδίο μπορεί να πάρει τιμές "ACTIVE" "INACTIVE".

#### VM

Ο πίνακας που διατηρεί τη ταυτότητα των εικονικών μηχανημάτων. Εκτός από το πεδίο κλειδί, περιέχει τα εξής πεδία:

- Το πεδίο Ip που περιέχει την IP διεύθυνση του εικονικού μηχανήματος.
- Το πεδίο memory\_size που περιέχει το συνολικό μέγεθος της μνήμης RAM του εικονικού μηχανήματος.

- Το πεδίο `cpu_cores` που περιέχει τον αριθμό των πυρήνων επεξεργαστή που διαθέτει το εικονικό μηχάνημα.
- Το πεδίο `disk_size` που περιέχει το μέγεθος του σκληρού δίσκου σε MB, που διαθέτει το εικονικό μηχάνημα.
- Το πεδίο `application_service` το οποίο είναι προαιρετικό και περιέχει τη διεύθυνση της υπηρεσία μίας οποιασδήποτε εφαρμογής που μπορεί να λειτουργεί στο εικονικό μηχάνημα και να χρειάζεται να ελέγχεται από το μηχανισμό.

#### Vm Status

Ο πίνακας περιέχει πληροφορίες σχετικά με τη κατάσταση των εικονικών μηχανημάτων. Ο πίνακας κάνει αναφορά στο πίνακα `vm` (1..1) μέσω του πεδίου `vm_id`, και στο πίνακα `status` (1..1) μέσω του πεδίου `status_id`. Επίσης περιέχει τα πεδία `start_timestamp` και `end_timestamp`. Το πεδίο `start_timestamp` περιγράφει τη χρονική στιγμή την οποία το εικονικό μηχάνημα εκκίνησε τη λειτουργία του, ενώ το πεδίο `end_timestamp` περιέχει τη χρονική στιγμή την οποία το εικονικό μηχάνημα τερμάτισε τη λειτουργία του.

#### Vm cpu log

Ο πίνακας περιέχει τις μετρήσεις επεξεργαστικού φόρτου των εικονικών μηχανημάτων. Η πληροφορία αυτή διατηρείται στο πεδίο `cpu_value`. Το πεδίο `vm_id` αποτελεί αναφορά στο πίνακα `vm` (1..\*). Τέλος το πεδίο `timestamp` περιέχει τη χρονική στιγμή κατά την οποία έγινε η μέτρηση.

#### Vm memory log

Ο πίνακας περιέχει τις μετρήσεις σχετικά με τη χρησιμοποίηση της μνήμης RAM των εικονικών μηχανημάτων. Η πληροφορία αυτή διατηρείται στο πεδίο `memory_value`. Το πεδίο `vm_id` αποτελεί αναφορά στο πίνακα `vm` (1..\*). Τέλος το πεδίο `timestamp` περιέχει τη χρονική στιγμή κατά την οποία έγινε η μέτρηση.

#### Vm disk log

Ο πίνακας περιέχει τις μετρήσεις σχετικά με τη χρησιμοποίηση του αποθηκευτικού χώρου των εικονικών μηχανημάτων. Η πληροφορία αυτή διατηρείται στο πεδίο `disk_value`. Το πεδίο `vm_id` αποτελεί αναφορά στο πίνακα `vm` (1..\*). Τέλος το πεδίο `timestamp` περιέχει τη χρονική στιγμή κατά την οποία έγινε η μέτρηση.

### **4.1.2 Drools**

Το σύστημα `drools` είναι ένα σύστημα διαχείρισης επιχειρησιακών κανόνων που χρησιμοποιεί τον αλγόριθμο `Rete`. Το `drools` υποστηρίζει το πρότυπο `JSR-94` για την μηχανή κανόνων για την κατασκευή, συντήρηση και επιβολή επιχειρησιακών πολιτικών για έναν οργανισμό, μία εφαρμογή ή μία υπηρεσία. [47]

### **4.1.3 JBPM**

Το σύστημα `jbpm` είναι μία ανοιχτού κώδικα μηχανή ροών εργασίας υλοποιημένη σε γλώσσα προγραμματισμού `java` που μπορεί και εκτελεί επιχειρησιακές διαδικασίες που περιγράφονται από `BPMN 2` (`Business Process Model and Notation`). Γενικά το σύστημα `jbpm` δέχεται μία γραφική διαδικασία ως είσοδο, η διαδικασία αποτελείται από μέρη τα οποία συνδέονται με ροές ακολουθίας. Το `jbpm` στηρίζεται στην εικονική μηχανή διαδικασιών (`Process Virtual Machine - PVM`) όπου ανήκει στον οργανισμό `JBoss` και υποστηρίζει πολλαπλές γλώσσες διαδικασιών. [48]

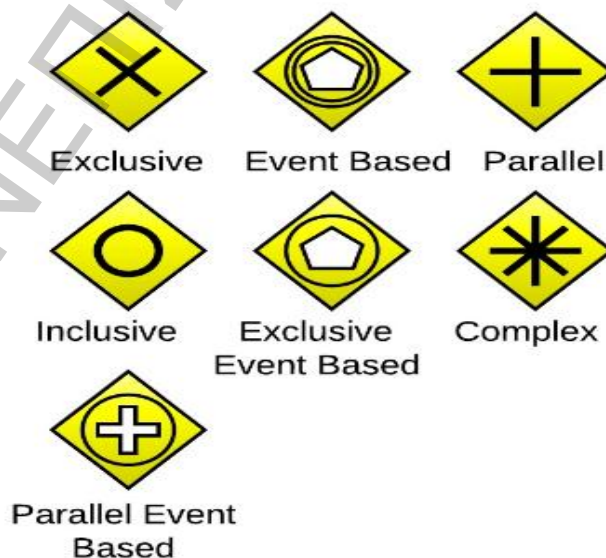
#### 4.1.4 BPMN

BPMN (Business Process Model and Notation) είναι μία γραφική αναπαράσταση για επιχειρησιακές διαδικασίες για ένα μοντέλο επιχειρησιακών διαδικασιών. Επίσης το BPMN είναι ένα πρότυπο για την μοντελοποίηση των επιχειρησιακών διαδικασιών που παρέχει μία γραφική σήμανση των διαδικασιών ανάλογη με αυτή των διαγραμμάτων δραστηριοτήτων της UML (Unified Modeling Language). Ο σκοπός του BPMN είναι να παρέχει διαχείριση των επιχειρησιακών διαδικασιών τόσο για τεχνικά καταρτισμένους χρήστες όσο και για χρήστες που κατέχουν απλώς την επιχειρησιακή λογική. Το BPMN απαρτίζεται από τέσσερις βασικές κατηγορίες στοιχείων οι οποίες είναι [49]:

- Τα αντικείμενα ροής (flow objects)
  - Δραστηριότητες (Activities) όπως παρουσιάζονται στην εικόνα 7 και μπορεί να είναι τμήματα εργασίας (μηχανής ή ανθρώπου), υποδιεργασίες, συναλλαγές και κλήσεις.
  - Πύλες (Gateways) που παρουσιάζονται στην εικόνα 8 και είναι διάφοροι λογικοί τελεστές.
  - Μηνύματα (Events) όπως αρχής και τέλους (Εικόνα 9).
- Τα αντικείμενα σύνδεσης (connecting objects) που αναφέρονται σε σειριακή ροή , ροή μηνυμάτων ή σε σχέσεις (Εικόνα 10)
- Swim lanes, σήμανση που οπτικά διαχωρίζει λογικά τμήματα της διαδικασίας.
- Artifacts που αναφέρονται σε αντικείμενα δεδομένων, ομάδες και στοιχεία σήμανσης.



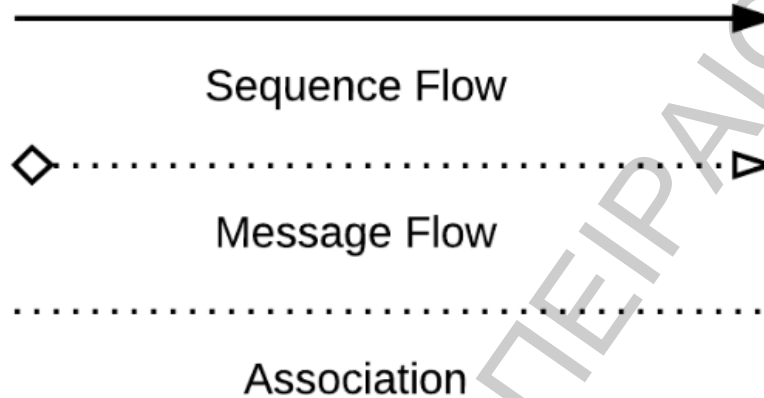
Εικόνα 7. Activities



Εικόνα 8. Gateways



Εικόνα 9. Events



Εικόνα 10. Connecting objects

#### 4.1.5 Java EE 6

Η Java Enterprise Edition είναι η πλατφόρμα της Oracle για την ανάπτυξη εφαρμογών και υπηρεσιών διαδικτύου, και άλλες πολυεπίπεδες, επεκτάσιμες, αξιόπιστες και ασφαλείς εφαρμογές. Η Java EE είναι επέκταση της Java SE. Η πλατφόρμα υιοθετεί μία σχεδίαση, κατά ένα μεγάλο βαθμό βασισμένη σε πολλαπλά τμήματα λογισμικού τα οποία τρέχουν σε έναν εξυπηρετητή εφαρμογών (application server). Η εφαρμογές που υποστηρίζονται από τη πλατφόρμα Java EE είναι υλοποιημένες στη γλώσσα προγραμματισμού Java.

#### 4.1.6 Jboss Application Server

Αποτελεί ένα εξυπηρετητή εφαρμογών για την υποστήριξη εφαρμογών Java EE. Είναι γραμμένος στη γλώσσα προγραμματισμού Java και γι' αυτό υποστηρίζεται από όλα τα περιβάλλοντα που έχουν εγκατεστημένη την πλατφόρμα Java.

#### 4.1.7 MySQL

Μία ανοικτού κώδικα πλατφόρμα για σχεσιακές βάσεις δεδομένων. Στηρίζεται στη γλώσσα επερωτήσεων SQL και είναι πολύ δημοφιλής για εφαρμογές διαδικτύου.

#### 4.1.8 RESTeasy

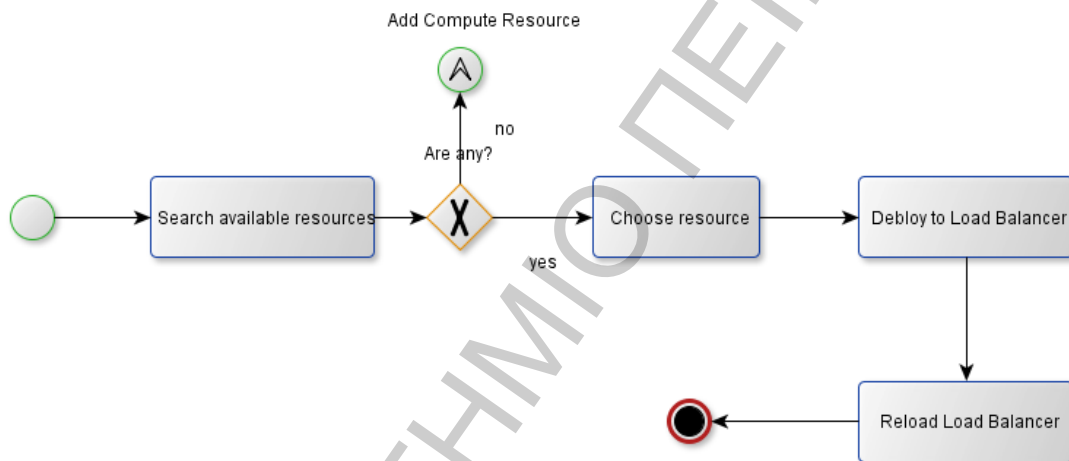
Αποτελεί την υλοποίηση για το πρότυπο RESTful υπηρεσιών JAX-RS για εφαρμογές διαδικτύου JavaEE. Το πακέτο RESTeasy προσφέρεται απευθείας από τον εξυπηρετητή εφαρμογών Jboss, και προσφέρει μία πληθώρα εργαλείων για την εύκολη υλοποίηση υπηρεσιοστρεφών εφαρμογών, καθώς και εφαρμογών πελατών (client) που καλούν διαδικτυακές RESTful υπηρεσίες.

## 4.2 Ροή Εργασίας

Όταν έρθει ένα σήμα ότι η εφαρμογή χρειάζεται και άλλους πόρους ώστε να εξυπηρετήσει ομαλά τα αιτήματα ενεργοποιείται μία ροή εργασίας η οποία ελέγχει εάν υπάρχουν διαθέσιμοι υπολογιστικοί πόροι. Εφόσον υπάρχουν υπολογιστικοί πόροι ενημερώνει τον εξισορροπητή φορτίου να συμπεριλάβει και αυτούς τους πόρους (Εικόνα 11).

Αρχείο τύπου event.json :

```
{ "timestamp": "2013-01-15 18:52", "alarmState": "red",  
  "vmIP": "192.168.21.2", "cpu": 0.1, "mem": 1.0, "disk": 0.5 }
```



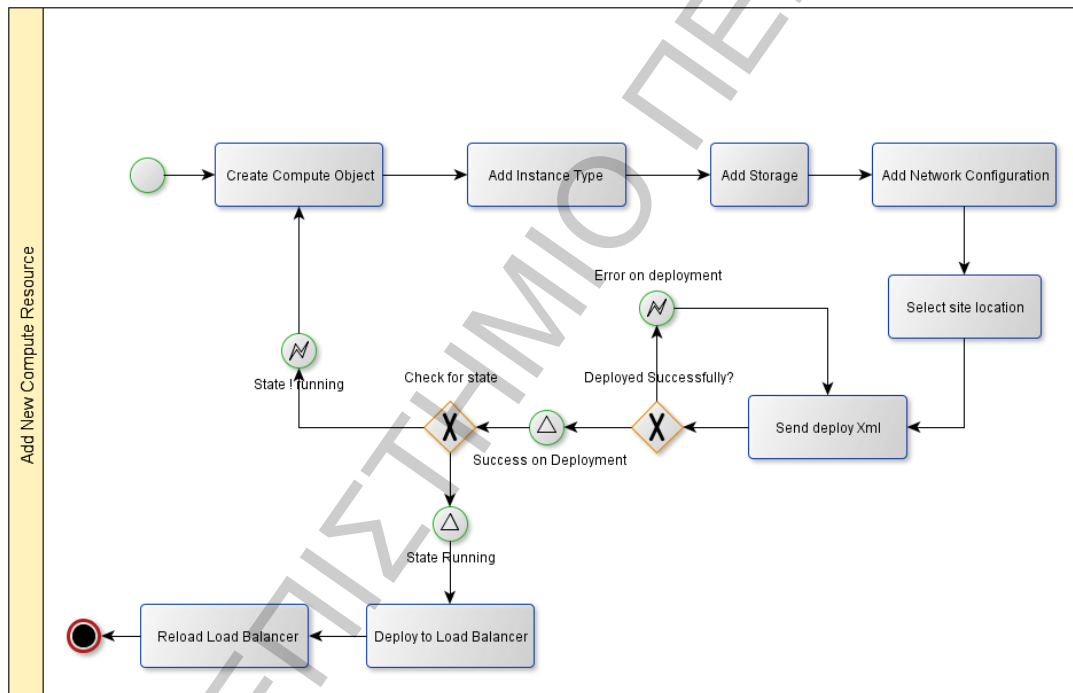
Εικόνα 11 Βασική ροή εργασίας.

Εάν δεν υπάρχουν διαθέσιμοι υπολογιστικοί πόροι ξεκινάει μία καινούρια ροή εργασίας έτσι ώστε να δημιουργηθούν νέοι υπολογιστικοί πόροι και να ενημερωθεί ο εξισορροπητής φορτίου. Στην συγκεκριμένη ροή εργασίας δημιουργείται ένα αντικείμενο τύπου "Compute Object" και στην συνέχεια του προσθέτουμε ένα τύπο στιγμιότυπου (επεξεργαστική ισχύ, μέγεθος μνήμης ) στην συνέχεια προστίθεται αποθηκευτικός χώρος και παραμετροποιείται το δίκτυο, γίνεται επιλογή τοποθεσίας στο νέφος και αποστέλλεται ένα αίτημα με μένα αρχείο xml που περιγράφει τον καινούριο υπολογιστικό πόρο προς δημιουργία. Εφόσον ο καινούριος υπολογιστικός πόρος δημιουργηθεί επιτυχώς γίνεται έλεγχος για την κατάσταση του μηχανήματος μόλις η κατάσταση είναι έτοιμη (running) τότε ενημερώνεται ο εξισορροπητής φορτίου για τον καινούριο πόρο.

Αρχείο για την δημιουργία νέου υπολογιστικού πόρου :

```

<compute>
  <name>test</name>
  <instance_type>medium</instance_type>
  <host>vmhost0</host>
  <disk>
    <storage href="/locations/storages"/>
    <type>OS</type>
    <target>hda</target>
  </disk>
  <nic>
    <network href="/locations/networks"/>
  </nic>
  <context>
  </context>
  <link href="/locations " rel="location"/>
</compute>
    
```



Εικόνα 12. Ροή εργασίας προσθήκη υπολογιστικού πόρου.



## 5 Περιβάλλον Υπολογιστικού νέφους

Η αρχιτεκτονική του περιβάλλοντος υπολογιστικού νέφους έχει σχεδιαστεί για να υποστηρίζει ερευνητικά πειράματα σε εφαρμογές, υπηρεσίες και συστήματα. Παρέχει συγκεκριμένες λειτουργίες, όπως

- Διαχείριση ελέγχου, σε επίπεδο υποδομής και virtual machine.
- Διαχείριση πειραμάτων με τη χρήση αρχείων περιγραφής πειραμάτων (descriptors/XML).
- Ελαστικότητα.
- Διαχείριση πόρων για την εγκατάσταση και τη λειτουργία εφαρμογών σε μία ομάδα διαφορετικών εικονοποιημένων μηχανημάτων.

### 5.1 Χαρακτηριστικά

Το περιβάλλον υπολογιστικού νέφους προσφέρει τις ακόλουθες λειτουργίες και χαρακτηριστικά:

#### 5.1.1 Πολλαπλά εργαλεία διαχείρισης

Για την δημιουργία πειραμάτων και πόρων, μπορούν να χρησιμοποιηθούν:

- Εντολές πρωτοκόλλου HTTP, χρησιμοποιώντας έτοιμα εργαλεία (cURL).
- Η δικτυακή διεπαφή του περιβάλλοντος υπολογιστικού νέφους. Πρόκειται για μία διαδικτυακή εφαρμογή, εύκολη στη χρήση. Η εφαρμογή δίνει την δυνατότητα στο χρήστη να δημιουργεί νέα πειράματα βήμα-βήμα, καθώς επίσης και να καθορίζει τις αρχικές παραμέτρους που αφορούν τους διαθέσιμους πόρους για το πείραμα.
- Ένα αρχείο περιγραφής πειράματος (Experiment descriptor). Ο χρήστης μπορεί να δημιουργήσει ένα αρχείο, το οποίο περιγράφει το πείραμα, σε μορφή JSON ή OVF. Η υποδομή διαθέτει έναν διαχειριστή πειραμάτων, ο οποίος μεταφράζει αυτό το αρχείο και δεσμεύει τους πόρους. Η λειτουργία αυτή επιτρέπει στο χρήστη να δημιουργεί πειράματα, χωρίς να έχει γνώση OCCl και HTTP μηνυμάτων.
- Η υπηρεσία Restfully, η οποία προσφέρει μία διεπαφή υπηρεσιών REST, σε γλώσσα προγραμματισμού Ruby. Σκοπός της είναι να παρέχει ένα επίπεδο ανταλλαγής HTTP μηνυμάτων μεταξύ χρήστη και εξυπηρετητή, καλύπτοντας τις λεπτομέρειες του πρωτοκόλλου. Η υπηρεσία επιτρέπει στο χρήστη να δημιουργεί αρχεία (scripts), που μπορούν να αυτοματοποιήσουν την διαδικασία δημιουργίας και εγκατάστασης πειραμάτων.
- Εργαλεία εντολών κονσόλας (CLIttools). Πρόκειται για εργαλεία τα οποία δίνουν την δυνατότητα στο χρήστη να επικοινωνεί με την διεπαφή του περιβάλλοντος υπολογιστικού νέφους, μέσω της παραδοσιακής γραμμής εντολών.

#### 5.1.2 Ετερογενείς υποδομές και πόροι

Το περιβάλλον υπολογιστικού νέφους περιλαμβάνει επιμέρους υποδομές υπολογιστικών νεφών, οι οποίες είναι γεωγραφικά κατανομημένες ανά την Ευρώπη, προσφέροντας ετερογενείς πόρους. Οι υποδομές είναι η EPPC στο Εδιμβούργο, η HPCCells στο Bristol της



Αγγλίας, η Inria στη Rennes της Γαλλίας, η HLRStην Στουτγάρδη της Γερμανίας, η IBBT VirtualWall στη Ghent του Βελγίου, και η PSNCστο Poznan της Πολωνίας.

Κάθε περιοχή προσφέρει πόρους (compute,storage,network), οι οποίοι μπορούν να χρησιμοποιηθούν για την δημιουργία πειραμάτων, αλλά και εικονικών μηχανημάτων. Κάποιες υποδομές προσφέρουν πρόσθετες υπηρεσίες, όπως δυναμική ανάθεση πόρων κατά εντολή (on-request resources, στην Inria) και προσομοίωση δικτύου (στο VirtualWall). Ο χρήστης, αφού πιστοποιήσει την ταυτότητα του, μπορεί να έχει πρόσβαση στους πόρους αυτούς μέσω SSH.

### 5.1.3 Ανάθεση υπολογιστικών πόρων κατ' εντολή

Η υποδομή Inria προσφέρει την δυνατότητα στο χρήστη να δεσμεύει πόρους την στιγμή που τους χρειάζεται. Παρέχει μία μεγάλη ποσότητα πόρων (162 κόμβοι/1800 πυρήνες). Έτσι ο χρήστης έχει την ευελιξία να δημιουργεί μεγάλης κλίμακας πειράματα, όπως επίσης και να έχει περισσότερο έλεγχο στις παραμέτρους του πειράματος, καθώς μπορεί να επιλέξει σε ποιόν φυσικό κόμβο να δεσμεύσει πόρους και να εκτελέσει το πείραμα του.

### 5.1.4 Εικονικά μηχανήματα και τύποι στιγμιότυπων

Το περιβάλλον υπολογιστικού νέφους προσφέρει πολλαπλά στιγμιότυπα για εικονικά μηχανήματα. Τα στιγμιότυπα βασίζονται στο linux λειτουργικό σύστημα Debian Squeeze, τα οποία ποικίλλουν σε αποθηκευτικό χώρο. Ο αποθηκευτικός χώρος μπορεί να επεκταθεί με την προσάρτηση επιπρόσθετου αποθηκευτικού χώρου (blockstorage), ο οποίος προσφέρεται ως ξεχωριστός τύπος πόρων.

Οι χρήστες μπορούν να δημιουργούν πολλαπλά εικονικά μηχανήματα, χρησιμοποιώντας τα στιγμιότυπα τύπου Debian, να εγκαταστήσουν και να τροποποιήσουν λογισμικό, και να τα αποθηκεύσουν. Η υποδομή, προς το παρόν, δεν προσφέρει τη δυνατότητα στο χρήστη να δημιουργήσει δικά του στιγμιότυπα και να ανεβάσει στην υποδομή, παρ' όλα αυτά έχει την δυνατότητα να αποθηκεύει μία κατάσταση (snapshot) ενός εικονικού μηχανήματος, αφού έχει εγκαταστήσει/τροποποιήσει λογισμικό, και να την χρησιμοποιήσει ως στιγμιότυπο για να δημιουργήσει καινούργια εικονικά μηχανήματα.

Όταν ο χρήστης δημιουργεί εικονικά μηχανήματα, ορίζει το τύπο στιγμιότυπου, ο οποίος ποικίλλει σε αριθμό πυρήνων και μέγεθος μνήμης. Ο τύπος στιγμιότυπου επιλέγεται είτε από μία λίστα (small,medium,large), είτε τον ορίζει ο χρήστης επιλέγοντας ο ίδιος τον αριθμό των πυρήνων και το μέγεθος της μνήμης.

### 5.1.5 Αποθήκευση

Όπως αναφέρθηκε και προηγουμένως, τα έτοιμα στιγμιότυπα παρέχονται με ένα συγκεκριμένο μέγεθος αποθηκευτικού χώρου. Αν ο χρήστης δεν ικανοποιείται από αυτό το μέγεθος, μπορεί να επεκτείνει τον αποθηκευτικό χώρο των εικονικών μηχανημάτων του πειράματος του, προσαρτώντας επιπλέον αποθηκευτικούς πόρους. Ο χρήστης μπορεί να καθορίσει το τύπο του αποθηκευτικού τμήματος που θέλει να δημιουργήσει (ext3, jfs, reiserfs), και φυσικά το μέγεθος του τμήματος. Είναι επίσης δυνατόν, να προσαρτηθεί το νέο τμήμα ως άμεση επέκταση του αποθηκευτικού χώρου του λειτουργικού συστήματος, του εικονικού μηχανήματος και όχι ως ξεχωριστή εγκατάσταση.

Εξ' ορισμού, τα τμήματα των αποθηκευτικών χώρων που δημιουργούνται αντιγράφονται όταν προσαρτούνται σε εικονικά μηχανήματα, και αντιστοίχως καταστρέφονται όταν τα εικονικά μηχανήματα τερματίζουν την λειτουργία τους. Ένω άλλοι πόροι (υπολογιστικοί, δικτύου) είναι δυνατό να δημιουργηθούν μόνο κατά την διάρκεια ενός πειράματος, οι αποθηκευτικοί πόροι μπορούν να δημιουργούνται ξεχωριστά και να είναι διαθέσιμοι και όταν ένα πείραμα λήξει. Επιπρόσθετα, είναι δυνατόν τα δεδομένα που αποθηκεύονται σε ένα τμήμα αποθηκευτικού χώρου που έχει δημιουργήσει ο χρήστης, κατά τη διάρκεια λειτουργίας ενός εικονικού μηχανήματος, να εγγράφονται μόνιμα στο τμήμα αυτό. Έτσι ο χρήστης μπορεί να χρησιμοποιήσει το τμήμα αυτό, με τα δεδομένα, και να το προσαρτήσει σε νέα εικονικά μηχανήματα που θα δημιουργήσει.

Τέλος, ένας άλλος ειδικός τύπος αποθηκευτικού τμήματος είναι ο διαμοιρασμένος αποθηκευτικός χώρος. Προς το παρόν, μπορούν μόνο να δημιουργηθούν στην υποδομή IBBT. Αντίθετα με τους άλλους τύπους αποθηκευτικών πόρων, οι διαμοιρασμένοι αποθηκευτικοί πόροι, είναι προσπελάσιμοι από πολλαπλά εικονικά μηχανήματα την ίδια στιγμή. Εάν ένα τμήμα αποθηκευτικού χώρου, είναι διαμοιρασμένο, και έχει δημιουργηθεί εκτός του εύρους ενός πειράματος, τότε είναι δυνατόν, αυτό το τμήμα να είναι προσπελάσιμο από εικονικά μηχανήματα, από διαφορετικά πειράματα.

#### 5.1.6 Δικτυακοί πόροι

Οι υποδομές που αποτελούν το περιβάλλον υπολογιστικού νέφους, διασυνδέονται μέσω του δημόσιου διαδικτύου (public Internet), όμως μόνο μερικοί πάροχοι υποδομών έχουν την δυνατότητα να παρέχουν δημόσιες IPv4 διευθύνσεις. Ως εκ τούτου, το περιβάλλον υπολογιστικού νέφους λειτουργεί ένα ιδιωτικό εικονικό δίκτυο (VPN) για να δρομολογεί την κίνηση μεταξύ των υποδομών.

#### 5.1.7 Διαχείριση ελέγχου και επίβλεψη

Το περιβάλλον υπολογιστικού νέφους παρέχει εκτεταμένη πληροφορία, σχετικά με τη κατάσταση των εικονικών πόρων σε διάφορα πειράματα, καθώς επίσης και για την κατάσταση των φυσικών κόμβων, στους οποίους εγκαθίστανται τα εικονικά μηχανήματα. Αυτό είναι ένα μοναδικό χαρακτηριστικό του περιβάλλοντος υπολογιστικού νέφους, που επιτρέπει στο χρήστη να συσχετίζει παρατηρήσεις που αφορούν εικονικά μηχανήματα, με παρατηρήσεις και συμβάντα που συμβαίνουν σε φυσικούς κόμβους.

Η πλατφόρμα παρακολούθησης που προσφέρεται από την υποδομή, βασίζεται στην υπηρεσία ελέγχου ανοικτού κώδικα Zabbix. Το λογισμικό αποτελείται από δύο κύρια προγράμματα. Το πρόγραμμα Zabbix Server και το πρόγραμμα Zabbix Agent. Το πρόγραμμα Zabbix Server, αναφέρεται και ως συγκεντρωτής (Aggregator), και εγκαθίσταται σε ξεχωριστό εικονικό μηχανήμα, ενώ οι "πελάτες" (agents), εγκαθίστανται σε κάθε εικονικό μηχανήμα που δημιουργεί ο χρήστης κατά την διάρκεια ενός πειράματος. Ο συγκεντρωτής αναλαμβάνει να συλλέξει πληροφορίες που στέλνει κάθε εφαρμογή πελάτη, δίνοντας μία ενοποιημένη εικόνα σχετικά με την κατάσταση των πόρων ενός πειράματος. Εκτενής αναφορά στην πλατφόρμα και την αρχιτεκτονική ελέγχου και επίβλεψης θα γίνει στο επόμενο υποκεφάλαιο.

### 5.1.8 Ελαστικότητα

Ως ελαστικότητα αναφέρεται η δυνατότητα να προσθέτονται και να αφαιρούνται δυναμικά πόροι, σύμφωνα με το φόρτο και τις ανάγκες ενός εικονικού μηχανήματος ή πειράματος. Το περιβάλλον υπολογιστικού νέφους παρέχει ένα ξεχωριστό εικονικό μηχάνημα με μία μηχανή Ελαστικότητας βασισμένη στο HAProxy και το Kamailio.

Η «ελαστικότητα ως υπηρεσία» (EaaS) στο περιβάλλον υπολογιστικού νέφους, χρησιμοποιεί την πληροφορία που συλλέγει από τα εικονικά μηχανήματα ο Zabbix Aggregator, και δυναμικά προσθέτει ή αφαιρεί πόρους, σύμφωνα με μία ομάδα κανόνων ελαστικότητας, που ορίζονται από το χρήστη.

### 5.1.9 Μηχανισμός ειδοποιήσεων

Κατά την εκτέλεση του πειράματος, και ιδιαίτερα αν είναι ενεργοποιημένος ο μηχανισμός της ελαστικότητας, είναι πολύ σημαντικό για το χρήστη να μπορεί να επιβλέπει αλλαγές και συμβάντα που συμβαίνουν στο πείραμα. Το περιβάλλον υπολογιστικού νέφους προσφέρει τη δυνατότητα στους πελάτες να λαμβάνουν ειδοποιήσεις όταν η κατάσταση του πειράματος ή των πόρων του πειράματος αλλάζει, όταν αυτοί δημιουργούνται, ανανεώνονται, ή καταστρέφονται. Οι αλλαγές κατάστασης είναι διαθέσιμες ως συμβάντα σε μία ουρά μηνυμάτων η οποία χρησιμοποιεί το RabbitMQ.

Οι καταστάσεις του πειράματος είναι σύμφωνες με το κύκλο ζωής του πειράματος, και οι καταστάσεις των πόρων είναι σύμφωνες με τις δηλωμένες καταστάσεις OCCI.

### 5.1.10 Πλαισιοποίηση

Τα στοιχεία πλαισιοποίησης του OCCI μπορούν να χρησιμοποιηθούν για να μεταφέρουν τιμές αρχικοποίησης για ένα πείραμα, σε μορφή από ζευγών τιμών-κλειδιά. Αυτό χρησιμοποιείται από τις υποδομές του περιβάλλοντος υπολογιστικού νέφους και το πλαίσιο λειτουργίας του, αλλά είναι επίσης διαθέσιμο και στους χρήστες για να ορίζουν, π.χ.:

- Την IP διεύθυνση του εικονικού μηχανήματος όπου είναι εγκατεστημένος ο Zabbix Aggregator.
- Οποιοσδήποτε παραμέτρους ελέγχου και επίβλεψης.
- Κανόνες ελαστικότητας, εάν αυτή χρησιμοποιείται.
- Προγράμματα τα οποία εκτελούνται μετά την εγκατάσταση εικονικού μηχανήματος.
- Πρόσθετα κλειδιά SSH, για να επιτρέπεται η πρόσβαση σε ένα πείραμα από πολλαπλούς χρήστες.

### 5.1.11 Περιγραφή αρχιτεκτονικής του περιβάλλοντος υπολογιστικού νέφους

Ο διαχειριστής ροής εργασιών είναι σε θέση να λαμβάνει γεγονότα που παράγει ο αποτιμητής. Έχει αναπτυχθεί στην πλατφόρμα Drools, ώστε να υπάρχει η δυνατότητα διαχείρισης κανόνων. Ο διαχειριστής ροής εργασιών ρυθμίζεται και στη συνέχεια τα εικονικά μηχανήματα καλούν τις υπηρεσίες που βασίζονται σε ένα αρχείο περιγραφής ροής εργασιών, το οποίο εκτός από αυτή τη πληροφορία, περιέχει πληροφορία σχετικά με το ποια βήματα πρέπει να ακολουθηθούν σε κάθε πιθανό συμβάν. Τα βήματα αυτά περιλαμβάνουν κλήσεις άλλων υπηρεσιών ή αλλαγές στη ρύθμιση ήδη ενεργών υπηρεσιών. Ο συντάκτης ροής εργασιών είναι σε θέση να αντιδρά σε πιθανά λάθη, με τη βοήθεια

συγκεκριμένων βημάτων χειρισμού λαθών, τα οποία ορίζονται στη περιγραφή ροής εργασιών. Αυτά μπορεί να είναι αρκετά απλά, όπως επανάληψη ή αποτυχία, ή πιο περίπλοκα, που μπορεί να ορίζονται από τον ίδιο τον προγραμματιστή και μπορούν να περιέχουν κλήσεις σε καινούργιες υπηρεσίες ή την επαναρύθμιση ήδη υπαρχόντων. Είναι πολύ σημαντικό να διαχωρίζονται οι διαφορές μεταξύ συμβάντων και σφαλμάτων. Το συμβάν είναι μία γνωστή και αναμενόμενη κατάσταση του συστήματος, στην οποία ο διαχειριστής ροής εργασιών μπορεί να αντιδράσει ακολουθώντας κάποια προκαθορισμένα βήματα. Τα σφάλματα από την άλλη πλευρά, είναι άγνωστες και απρόβλεπτες καταστάσεις. Σε τέτοιες περιπτώσεις, πρέπει να πραγματοποιηθούν γενικού τύπου λειτουργίες, όπως η αποθήκευση της κατάστασης ενός εικονικού μηχανήματος. Υπάρχει η δυνατότητα να οριστούν συγκεκριμένες δομές στη περιγραφή ροής εργασιών, καθώς επίσης και ειδικές περιπτώσεις, για περιστατικά επανάληψης ή αποτυχίας. Η επανάληψη π.χ μπορεί να προστεθεί ως μία εύκολη λύση για να αντιμετωπίζονται προβλήματα που εμφανίζονται σποραδικά.

Ο μηχανισμός διαχείρισης ελέγχου, είναι κάτι παραπάνω από ένα απλό εργαλείο μέτρησης, καθώς παρέχει συγκεντρωτικές λειτουργίες στη συλλογή δεδομένων της υποδομής. Ο μηχανισμός είναι σε θέση να συγκεντρώνει πληροφορία σε επίπεδο εφαρμογής, συστάδας εικονικών μηχανημάτων και υπολογιστικού νέφους. Ο λόγος που υπάρχει αυτή η πολυεπίπεδη προσέγγιση είναι ότι όσο περισσότερα επίπεδα εισάγονται, τόσο μεγαλύτερο είναι το κόστος επικοινωνίας.

Συνεπώς το σύστημα διαχωρίζει διαφορετικά επίπεδα συγκέντρωσης πληροφορίας. Αναλυτικότερα, μπορεί να οριστούν ξεχωριστοί κανόνες για κάθε εφαρμογή, συστάδα εικονικών μηχανημάτων ή υποδομής υπολογιστικού νέφους συνδυασμένοι με ένα πλαίσιο χρόνου. Παρ' όλα αυτά, μπορεί να γίνει μερική συγκέντρωση πληροφορίας σε χαμηλότερα επίπεδα. Για κάθε εικονικό μηχάνημα υπάρχει ένα στιγμιότυπο του μηχανισμού ελέγχου, και είναι υπεύθυνο για τη συλλογή συμβάντων για το μηχάνημα. Κάθε στιγμιότυπο περιλαμβάνει ένα συλλέκτη πληροφορίας για όλες της εφαρμογές που είναι εγκατεστημένες στο μηχάνημα, ένα μηχανισμό κανόνων που αλλάζει τα χρονικά διαστήματα μεταξύ ελέγχων, ανάλογα με τη κρισιμότητα των συμβάντων, καθώς και έναν αποστολέα που στέλνει τη πληροφορία στον διαχειριστή ελέγχου. Τα τμήματα λογισμικού του στιγμιότυπου είναι ανεπτυγμένα στη γλώσσα προγραμματισμού Java. Ξεχωριστό τμήμα συλλέγει πληροφορίες σχετικά με τη κατάσταση του εικονικού μηχανήματος, και με τη σειρά του στέλνει τη πληροφορία αυτή στο τοπικό υποσύστημα παρακολούθησης. Η τμηματοποίηση του λογισμικού παρακολούθησης σε κάθε μηχάνημα επιτρέπει σε άλλα συστήματα μέτρησης να εγκατασταθούν και να χρησιμοποιηθούν ως πηγές πληροφορίας.

Όσον αφορά τον αποτιμητή, τα συμβάντα ακολουθούν ένα απλό σχήμα περιλαμβάνοντας μερικά στατικά πεδία, καθώς επίσης και πεδία τα οποία σχετίζονται με το συμβάν που δημιουργήθηκε. Οι κανόνες παρέχονται ως είσοδος στο μηχανισμό ελέγχου και παρακολούθησης, ώστε αυτός να καθορίσει το πως θα αντιμετωπίζεται κάθε συμβάν. Περιλαμβάνουν 3 κύριους τομείς:

Καθορισμός θέματος κανόνα. Σε αυτό το σημείο ορίζεται ένα όνομα όπου θα λειτουργεί ως μοναδικό αναγνωριστικό για να καταγραφεί στην ουρά κανόνων. Επιπροσθέτως, στο ίδιο τμήμα ορίζονται το επίπεδο συγκέντρωσης και η διάρκεια του.

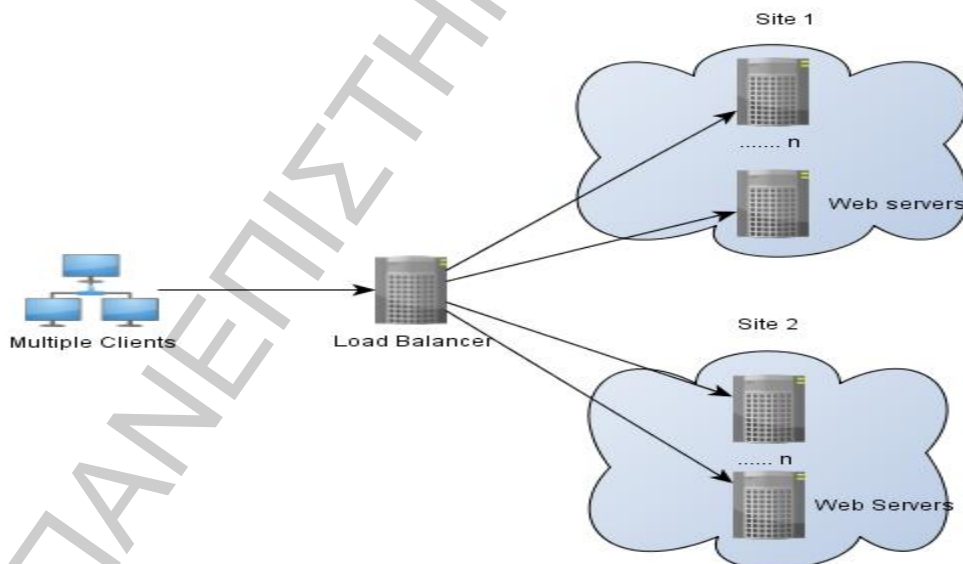
Το τμήμα του κανόνα που παρέχει βασικές λογικές παραμέτρους στα πεδία των συμβάντων. Αν ένα ορισμένο πεδίο ενός κανόνα δεν υπάρχει σε ένα συμβάν, τότε αυτό απορρίπτεται.

Το τμήμα που θα εκτελεί συγκεκριμένες λειτουργίες ανάλογα με τα συμβάντα και θα παραθέτει τα αποτελέσματα σε μία ουρά αποτελεσμάτων.

Κάθε τοποθεσία φυσικών υποδομών, παρέχει το δικό της υποδίκτυο δημόσιων και τοπικών διευθύνσεων IP, και έτσι είναι δυνατή η επικοινωνία πόρων μεταξύ διαφορετικών υποδομών, επιτρέποντας έτσι τη διενέργεια *cross-site* πειραμάτων.

## 5.2 Τοπολογία Πειράματος

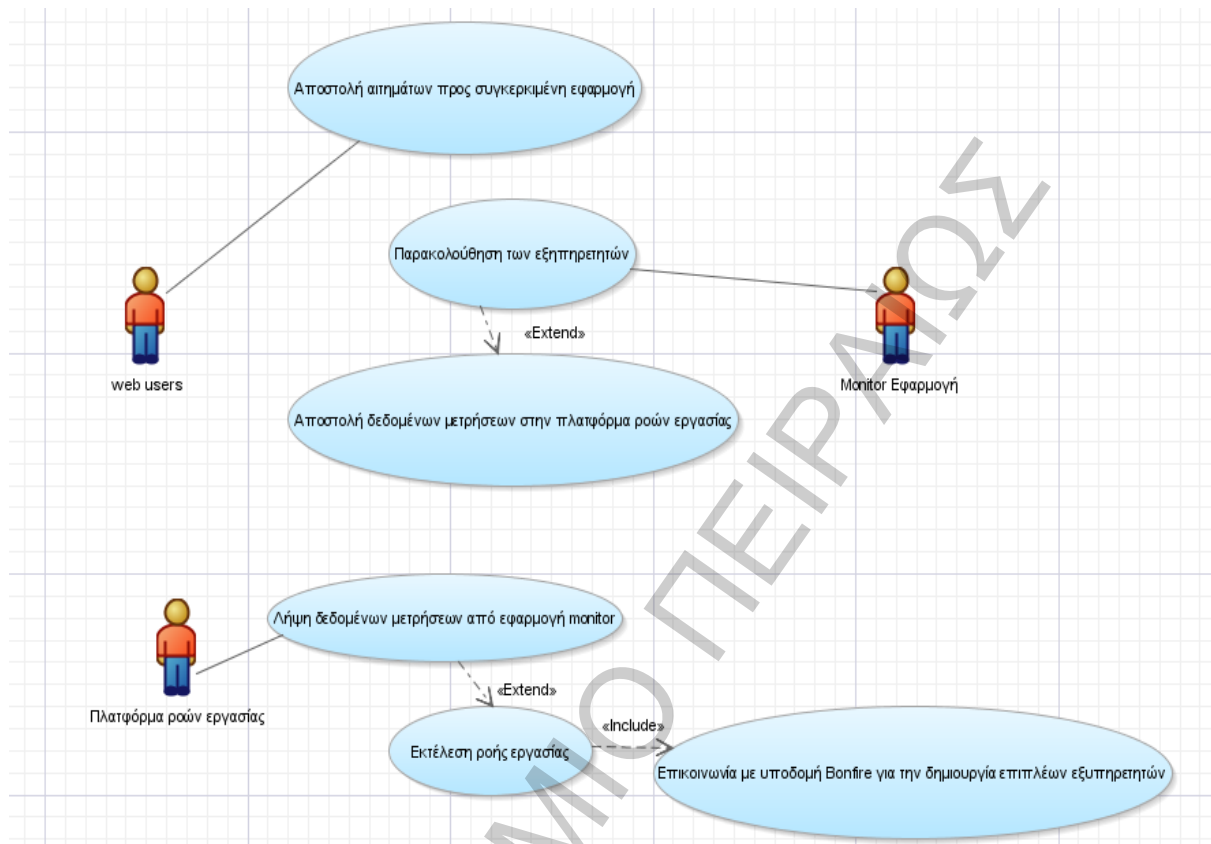
Το πείραμα που διεξήχθητε είχε ως σκοπό να παρατηρήσει την συμπεριφορά μίας εφαρμογής ιστού (στην προκειμένη περίπτωση τον έλεγχο εγκυρότητας ενός αριθμού φορολογικού μητρώου) όταν υπάρχει αρκετά μεγάλος φόρτος αιτημάτων προς την εφαρμογή αυτή. Αναλυτικότερα, όπως φαίνεται και στην εικόνα 13 προσομοιώθηκαν πολλαπλοί πελάτες που στέλνουν τα αιτήματά τους σε έναν εξισορροπητή φορτίου (load balancer) ο οποίος με την σειρά του τα μοιράζει σε πολλαπλούς εξυπηρετητές που φέρουν ένα στιγμιότυπο της εφαρμογής. Οι εξυπηρετητές μπορούν να βρίσκονται στην ίδια τοποθεσία στο νέφος ή σε διαφορετικές.



Εικόνα 13 Τοπολογία πειράματος

Από αυτή την πειραματική διαδικασία θα μελετήσουμε την συμπεριφορά της πλατφόρμας διαχείρισης ροών εργασίας που αναπτύχθηκε για να συμπεράνουμε κατά πόσο εξυπηρετεί αυτή η αρχιτεκτονική.

### 5.3 Προσομιώσεις



Εικόνα 14. Σενάριο χρήσης.

Στο σενάριο που προσομοιάσαμε όπως φαίνεται και στην εικόνα 14, χρήστες του διαδικτύου που προσομοιώθηκαν (η προσομοίωση έγινε με το εργαλείο Apache JMeter) έστειλαν αιτήματα προς μία εφαρμογή, η εφαρμογή Monitor παρακολουθούσε συνεχώς τους εξυπηρετητές και αποστέλλε τα δεδομένα μετρήσεων στην πλατφόρμα ροών εργασίας σε μορφή event object όπως έχει αναφερθεί στην ενότητα της αρχιτεκτονικής της πλατφόρμας. Στη συνέχεια η πλατφόρμα ροών εργασίας ανάλογα με τον τύπο του event εκτελούσε και την ανάλογη ροή εργασίας και όταν ήταν απαραίτητο ενημέρωνε την υποδομή του υπολογιστικού νέφους για επιπλέον υπολογιστικούς πόρους. Δύο είδη event χρησιμοποιήθηκαν, yellow event και red event. Το yellow event όταν αποστέλλονταν από την εφαρμογή monitor μας ενημέρωνε ότι η υπολογιστικοί πόροι που χρησιμοποιούνται από την συνολική υποδομή του πειράματος έχουν ξεπεράσει ένα κατώφλι που οριστικό από την εφαρμογή monitor και το red event μας ενημερώνει ότι οι υπολογιστικοί πόροι έχουν φθάσει σε κρίσιμο σημείο (πάλι βάση ενός ορίου) και πρέπει να δοθούν και άλλοι. Οι κανόνες που χρησιμοποιήθηκαν στην πλατφόρμα ροών εργασίας ήταν οι εξής :

- Μόλις λάβουμε ένα yellow event από έναν υπολογιστικό πόρο να ξεκινήσει η ροή εργασίας που δημιουργεί νέους υπολογιστικούς πόρους στο περιβάλλον νέφους

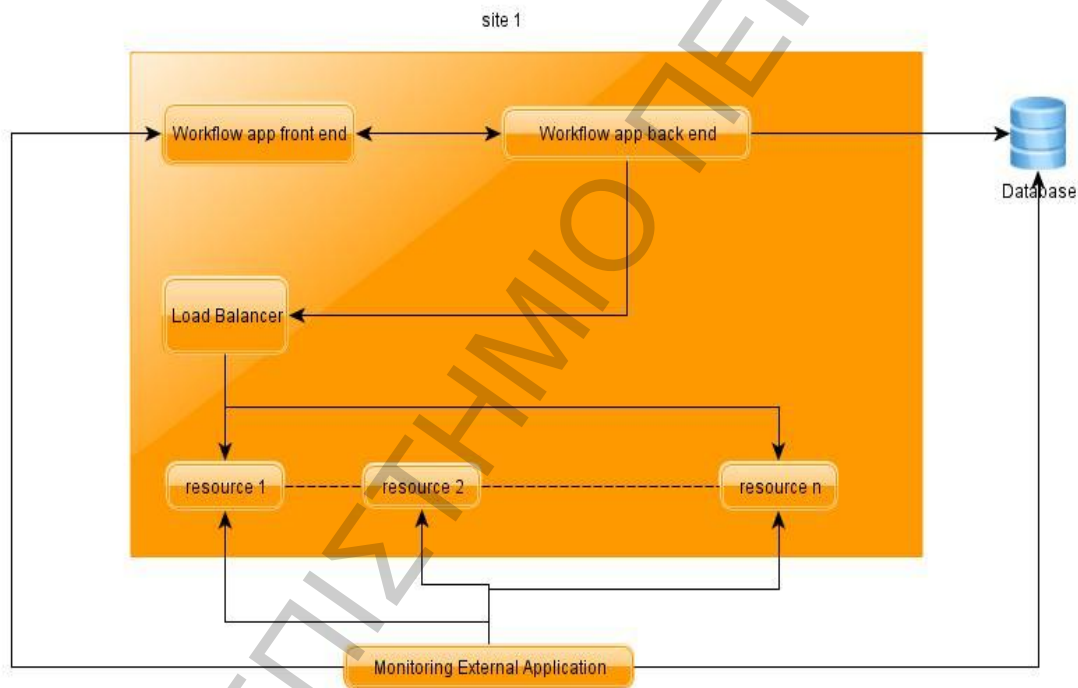


- Μόλις λάβουμε ένα red event από ένα υπολογιστικό πόρο να ξεκινήσει η ροή εργασίας για να δημιουργηθεί ένα καινούριο στιγμιότυπο της εφαρμογής και να αρχίσει να εξυπηρετεί τα αιτήματα των χρηστών.

Σκοπός του πειράματος είναι να παρατηρήσουμε κατά πόσο η πλατφόρμα ροών εργασίας που αναπτύχθηκε μπορεί να διαχειριστεί αυτούς του κανόνες αποτελεσματικά έτσι ώστε να εξασφαλίσει μία ομαλή συμπεριφορά για την εξυπηρέτηση των αιτημάτων των χρηστών προς την εφαρμογή.

### 5.3.1 Προσομοίωση 1

Στην πρώτη προσομοίωση που κάναμε χρησιμοποιήθηκε μία τοποθεσία στο περιβάλλον υπολογιστικού νέφους. Σε αυτή την τοποθεσία βρισκόταν όλα τα υποσυστήματα του πειράματος. Οι αρχικές συνθήκες του πειράματος ήταν με δύο εξυπηρετητές διαδικτύου στην τοποθεσία και έναν εξισορροπητή φόρτου.



Εικόνα 15. Προσομοίωση 1 Application components.

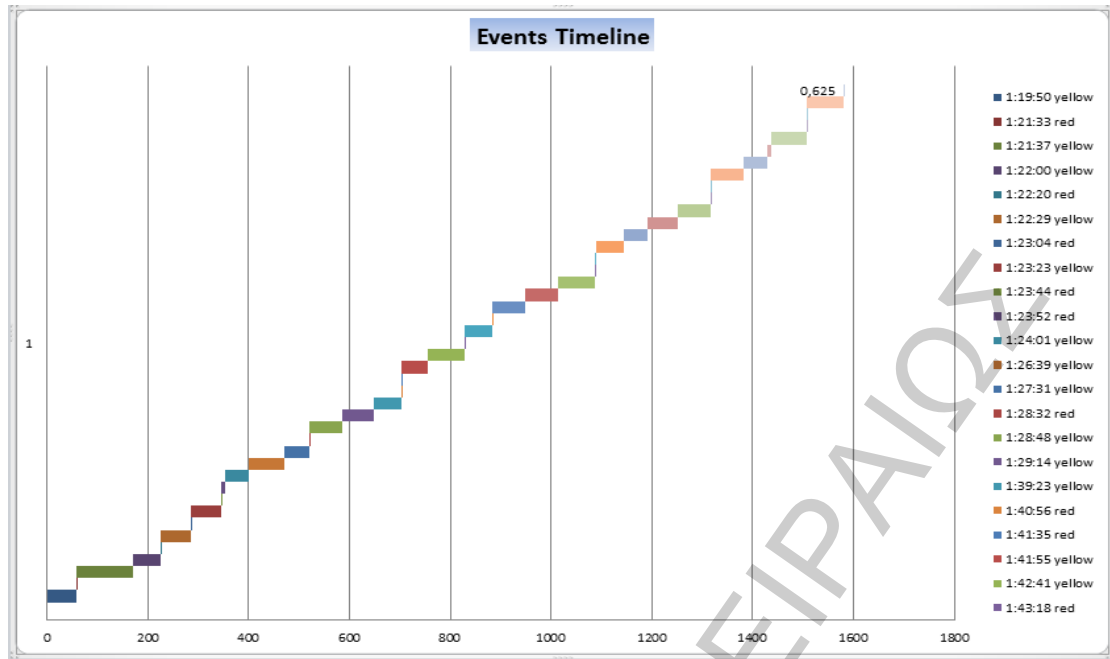
Τα αιτήματα προς την εφαρμογή αρχίσανε από 140 το δευτερόλεπτο και σταδιακά κλιμακώθηκαν μέχρι τα 700 το δευτερόλεπτο. Όπως παρατηρούμε και στον πίνακα 8 δεν υπήρχε επικάλυψη μεταξύ των events και ο χρόνος επεξεργασίας κυμαίνονταν ανάλογα με την ροή εργασίας (εάν χρειαζόταν να δημιουργηθούν νέοι υπολογιστική πόροι ή όχι).

Πίνακας 8. Συνολικά δεδομένα πειράματος

Αναγνωριστικό VM	Ώρα Λήψης Event	Τύπος Event	Χρόνος Επεξεργασίας Event (seconds)
24	1:19:50	yellow	58,629
24	1:21:33	red	0,278

55	1:21:37	yellow	112,418
24	1:22:00	yellow	54,909
24	1:22:20	red	0,419
56	1:22:29	yellow	59,571
24	1:23:04	red	0,605
24	1:23:23	yellow	58,565
24	1:23:44	red	0,29
56	1:23:52	red	7,401
24	1:24:01	yellow	48,445
56	1:26:39	yellow	70,445
57	1:27:31	yellow	47,513
24	1:28:32	red	0,323
60	1:28:48	yellow	64,775
61	1:29:14	yellow	64,118
24	1:39:23	yellow	53,308
24	1:40:56	red	0,49
55	1:41:35	red	0,307
66	1:41:55	yellow	52,574
59	1:42:41	yellow	72,546
56	1:43:18	red	0,31
56	1:43:29	yellow	54,098
59	1:44:14	red	2,044
55	1:44:20	yellow	64,232
59	1:44:46	yellow	64,57
62	1:45:36	yellow	72,832
55	1:46:09	red	0,303
56	1:46:15	red	2,128
56	1:46:25	yellow	54,972
55	1:46:54	yellow	47,633
68	1:47:46	yellow	61,093
69	1:48:42	yellow	64,013
55	1:49:08	red	0,338
56	1:49:15	red	0,522
55	1:49:20	yellow	65,301
56	1:49:58	yellow	47,803
68	1:50:36	red	7,351
24	1:50:45	yellow	70,235
55	1:51:12	red	0,361
24	1:51:22	red	0,258
55	1:51:43	yellow	71,841
62	1:52:16	red	0,625





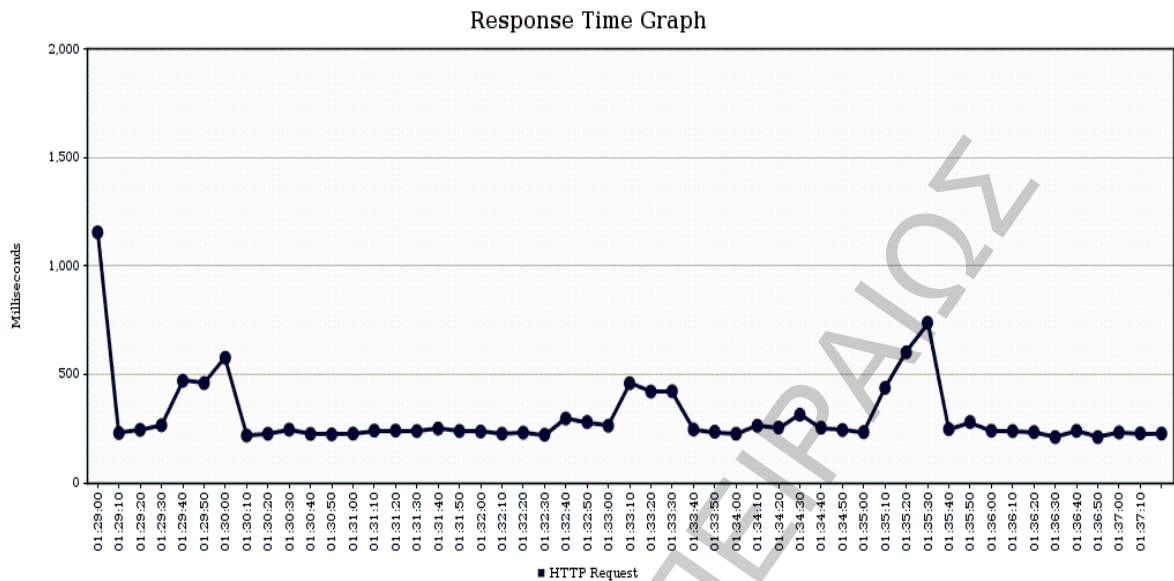
Εικόνα 16. Χρονοροή των events.

Στην συνολική διάρκεια του πειράματος χρησιμοποιήθηκαν 25 web servers που είχαν στιγμιότυπα της εφαρμογής (εικόνα 17).

ws136050420:/locations/uk-epcc/computes/16649	small	1	1024MiB node-v9	172.18.240.189	OK	RUNNING	
ws136050416:/locations/uk-epcc/computes/16648	small	1	1024MiB node-v9	172.18.240.188	OK	RUNNING	
ws136050412:/locations/uk-epcc/computes/16647	small	1	1024MiB node-v9	172.18.240.187	OK	RUNNING	
ws136050406:/locations/uk-epcc/computes/16646	small	1	1024MiB node-v9	172.18.240.186	OK	RUNNING	
ws136050402:/locations/uk-epcc/computes/16645	small	1	1024MiB node-v9	172.18.240.185	OK	RUNNING	
ws136050398:/locations/uk-epcc/computes/16644	small	1	1024MiB node-v9	172.18.240.184	OK	RUNNING	
ws136050393:/locations/uk-epcc/computes/16643	small	1	1024MiB node-v9	172.18.240.183	OK	RUNNING	
ws136050388:/locations/uk-epcc/computes/16642	small	1	1024MiB node-v9	172.18.240.182	OK	RUNNING	
ws136050386:/locations/uk-epcc/computes/16641	small	1	1024MiB node-v9	172.18.240.181	OK	RUNNING	
ws136050380:/locations/uk-epcc/computes/16640	small	1	1024MiB node-v9	172.18.240.180	OK	RUNNING	
ws136050376:/locations/uk-epcc/computes/16639	small	1	1024MiB node-v9	172.18.240.179	OK	RUNNING	
ws136050371:/locations/uk-epcc/computes/16638	small	1	1024MiB node-v9	172.18.240.178	OK	RUNNING	
ws136050356:/locations/uk-epcc/computes/16637	small	1	1024MiB node-v9	172.18.240.177	OK	RUNNING	
ws136050295:/locations/uk-epcc/computes/16636	small	1	1024MiB node-v9	172.18.240.176	OK	RUNNING	
ws136050292:/locations/uk-epcc/computes/16635	small	1	1024MiB node-v9	172.18.240.175	OK	RUNNING	
ws136050285:/locations/uk-epcc/computes/16634	small	1	1024MiB node-v9	172.18.240.174	OK	RUNNING	
ws136050279:/locations/uk-epcc/computes/16633	small	1	1024MiB node-v9	172.18.240.173	OK	RUNNING	
ws136050264:/locations/uk-epcc/computes/16632	small	1	1024MiB node-v9	172.18.240.172	OK	RUNNING	
ws136050260:/locations/uk-epcc/computes/16631	small	1	1024MiB node-v9	172.18.240.171	OK	RUNNING	
ws136050254:/locations/uk-epcc/computes/16630	small	1	1024MiB node-v9	172.18.240.170	OK	RUNNING	
ws136050252:/locations/uk-epcc/computes/16629	small	1	1024MiB node-v9	172.18.240.169	OK	RUNNING	
ws136050249:/locations/uk-epcc/computes/16628	small	1	1024MiB node-v9	172.18.240.168	OK	RUNNING	
ws136050239:/locations/uk-epcc/computes/16627	small	1	1024MiB node-v9	172.18.240.167	OK	RUNNING	
ws136046578:/locations/uk-epcc/computes/16625	small	1	1024MiB node-v9	172.18.240.166	OK	RUNNING	
wsNode	/locations/uk-epcc/computes/16504	small	1	1024MiB node-v9	172.18.240.95	OK	RUNNING
lb	/locations/uk-epcc/computes/16498	small	1	1024MiB BonFIRE Debian Squee	172.18.240.92	OK	RUNNING
controller	/locations/uk-epcc/computes/16492	large	2	4096MiB controller	172.18.240.89	OK	RUNNING

Εικόνα 17. Το σύνολο των εξυπηρετητών που χρησιμοποιήθηκαν.

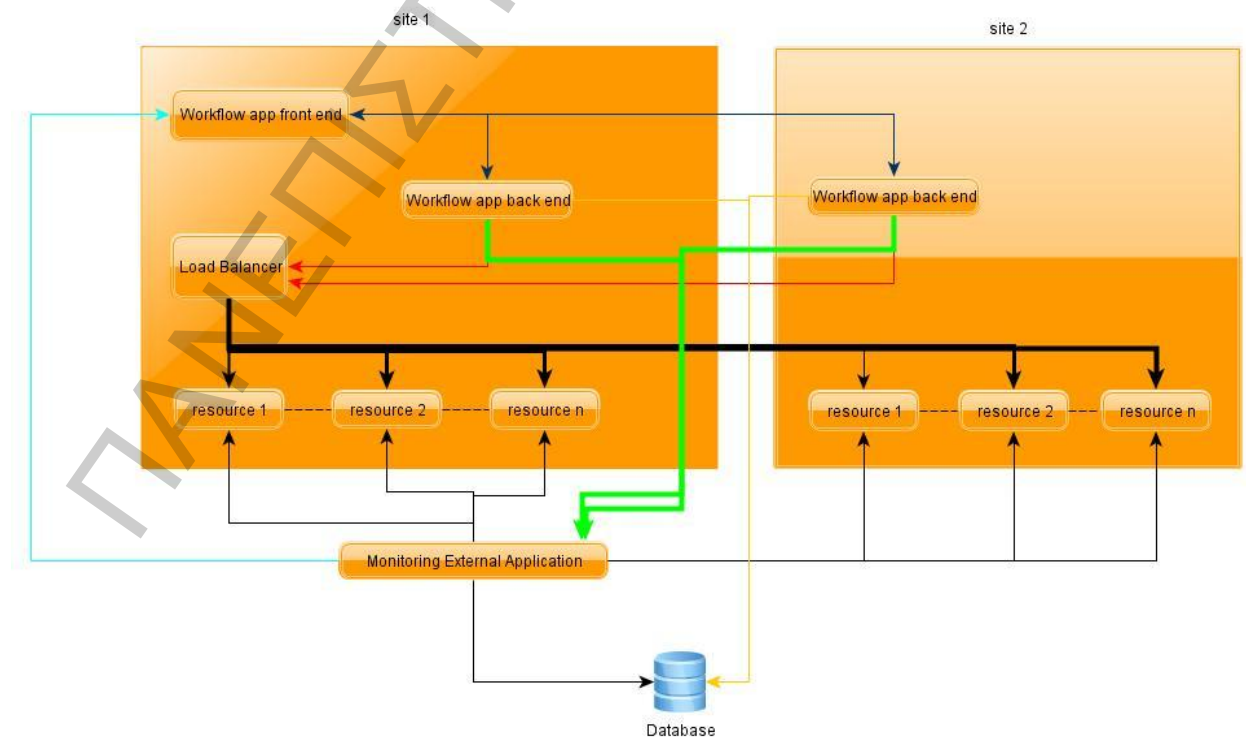
Όπως παρατηρούμε από την εικόνα 18 ο χρόνος απόκρισης της εφαρμογής κυμαίνεται σε απολύτως ικανοποιητικά όρια.



Εικόνα 18. Χρόνος απόκρισης εφαρμογής.

### 5.3.2 Προσομοίωση 2

Στην δεύτερη προσομοίωση που κάναμε χρησιμοποιήθηκαν δύο τοποθεσίες στο περιβάλλον υπολογιστικού νέφους. Σε κάθε τοποθεσία βρισκόταν υποσυστήματα της πλατφόρμας ρών εργασίας και εξυπηρετητές διαδικτύου που φιλοξενούσαν στιγμιότυπα της εφαρμογής. Οι αρχικές συνθήκες του πειράματος ήταν με δύο εξυπηρετητές διαδικτύου (έναν σε κάθε τοποθεσία) έναν εξισορροπητή φόρτου.



Εικόνα 19. Προσομοίωση 2 Application components.

Στους πίνακες που ακολουθούν περιγράφονται αναλυτικά τα events που λήφθηκαν για τον κάθε εξυπηρετητή διαδικτύου.

Πίνακας 9. Δεδομένα για τον εξυπηρετητή 1

VM	Ώρα λήψης Event	Ώρα Εκτέλεσης Event	Τύπος Event	Χρόνος που μεσολάβησε μεταξύ των events (yellow-red)
1	11:37:44	11:38:59	yellow	
1	11:49:59	11:49:59	red	11 sec
1	11:50:12	11:51:21	yellow	
1	11:50:32	11:50:32	red	-49 sec
1	11:51:01	11:52:06	yellow	
1	11:53:22	11:53:22	red	16 sec
1	11:53:51	11:55:09	yellow	
1	11:54:46	11:54:46	red	-23 sec
1	11:58:04	11:59:51	yellow	
1	0:00:26	0:00:26	red	35 sec
1	0:06:07	0:07:45	yellow	

Πίνακας 10. Δεδομένα για τον εξυπηρετητή 2

VM	Ώρα λήψης Event	Ώρα Εκτέλεσης Event	Τύπος Event	Χρόνος που μεσολάβησε μεταξύ των events (yellow-red)
2	11:50:40	11:52:09	yellow	
2	11:53:26	11:53:26	red	0:01:17
2	11:53:31	11:55:06	yellow	
2	11:57:58	11:57:58	red	0:02:52
2	11:58:25	11:59:51	yellow	
2	11:58:54	11:58:54	red	- 59 sec
2	11:59:00	12:01:30	yellow	
2	0:07:02	0:07:02	red	00:05:32

Πίνακας 11. Δεδομένα για τον εξυπηρετητή 3

VM	Ώρα λήψης Event	Ώρα Εκτέλεσης Event	Τύπος Event	Χρόνος που μεσολάβησε μεταξύ των events (yellow-red)
3	11:55:14	11:56:43	yellow	
3	11:58:49	11:58:49	red	0:02:06
3	11:59:26	12:01:30	yellow	

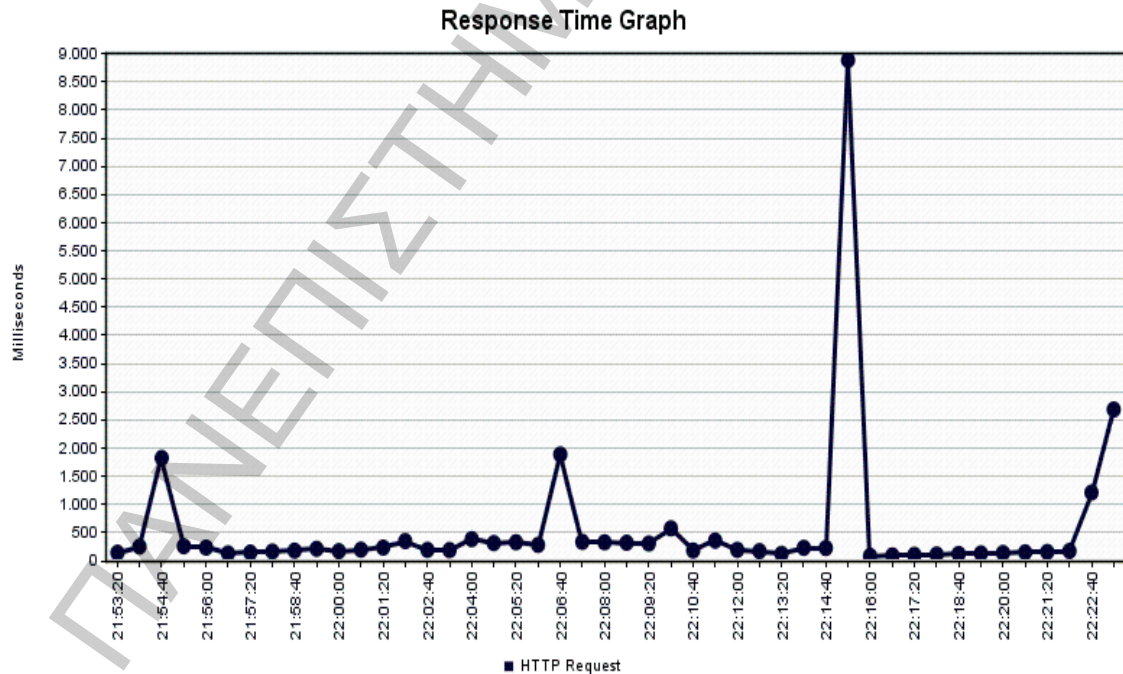
Πίνακας 12. Δεδομένα για τον εξυπηρετητή 4

VM	Ώρα λήψης Event	Ώρα Εκτέλεσης Event	Τύπος Event	Χρόνος που μεσολάβησε μεταξύ των events (yellow-red)
4	11:57:23	11:59:51	yellow	
4	0:00:18	0:00:18	red	00:00:27
4	0:06:39	0:08:16	yellow	

Πίνακας 13. Δεδομένα για τον εξυπηρετητή 5

VM	Ώρα λήψης Event	Ώρα Εκτέλεσης Event	Τύπος Event	Χρόνος που μεσολάβησε μεταξύ των events (yellow-red)
5	11:54:51	11:56:40	yellow	
5	0:01:02	0:01:02	red	00:04:22

Όπως παρατηρούμε υπήρχαν επικαλύψεις σε τρεις μόνο περιπτώσεις στην συνολική διάρκεια του πειράματος, οι οποίες όμως δεν επηρέασαν σημαντικά την απόδοση της εφαρμογής.



Εικόνα 20. Χρόνος απόκρισης της εφαρμογής.

## 6 Συμπεράσματα

Στην εργασία αυτή αναπτύξαμε μία υπηρεσιοστρεφή πλατφόρμα διαχείρισης επιχειρησιακών διαδικασιών σε περιβάλλον υπολογιστικού νέφους. Η πλατφόρμα σχεδιάστηκε έτσι ώστε να δέχεται μηνύματα μέσω υπηρεσιών διαδικτύου να επεξεργάζεται την κατάλληλη ροή εργασίας ανάλογα με τον τύπο του μηνύματος που έχει δεχθεί, στηριζόμενη σε ένα σύνολο κανόνων και να επικοινωνεί εκ νέου πάλι μέσω υπηρεσιών διαδικτύου με εξωτερικές εφαρμογές. Η αρχιτεκτονική της πλατφόρμας αποτελείται από δύο κύρια μέρη το front-end και το back-end. Το front-end διαχειρίζεται την επικοινωνία με τις εξωτερικές εφαρμογές και το back-end διαχειρίζεται το σύστημα κανόνων και τον μηχανισμό εκτέλεσης των ροών εργασίας. Το πλεονέκτημα αυτής της αρχιτεκτονικής είναι ότι μπορούμε να δημιουργήσουμε μία επεκτάσιμη υποδομή της πλατφόρμας σε ένα υπολογιστικό νέφος με ανεξάρτητα τμήματα διαχείρισης της επικοινωνίας με εξωτερικές εφαρμογές και διαχείρισης της εκτέλεσης των ροών εργασίας. Τα πειραματικά αποτελέσματα μας έδωσαν ενθαρρυντικά στοιχεία ότι αυτή η αρχιτεκτονική μπορεί να χρησιμοποιηθεί για επεξεργασία επιχειρησιακών διαδικασιών σε περιβάλλοντα υπολογιστικού νέφους.

## 7 Βιβλιογραφία

1. R. Buyya, C. S. Yeo, S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", 10th IEEE International Conference on High Performance Computing and Communications, 2008
2. A. Malis, "Routing over Large Clouds (ROLC) Charter", Meeting minutes of the 32nd IETF meeting, available online at: <http://www.ietf.org/proceedings/32/charters/rolc-charter.html>
3. Wikipedia, "John McCarthy (computer scientist)", available online at: [http://en.wikipedia.org/wiki/John\\_McCarthy\\_\(computer\\_scientist\)](http://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist))
4. New York Times, "Internet Critic Takes on Microsoft", available online at: <http://www.nytimes.com/2001/04/09/technology>
5. J.Barr, "Amazon EC2 Beta", available online: [http://aws.typepad.com/aws/2006/08/amazon\\_ec2\\_beta.html](http://aws.typepad.com/aws/2006/08/amazon_ec2_beta.html)
6. Cloud Computing Expert Group Report, "The Future of Cloud Computing", European Commission, available online at: <http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf>
7. Γρηγόριος Χ. Κατσαρός, «Τεχνολογίες διαχείρισης υπολογιστικών υποδομών και εφαρμογών σε υπηρεσιοστρεφείς αρχιτεκτονικές και περιβάλλοντα Νεφών» 2012
8. Amazon AWS , <http://aws.amazon.com/ec2>
9. Amazon EC2 API, <http://docs.amazonwebservices.com/AWSEC2/2009-11-30/APIReference/>
10. Amazon EC2 , <http://aws.amazon.com/ec2>
11. Amazon S3 , <http://aws.amazon.com/s2>
12. Amazon SQS , <http://aws.amazon.com/sqs>
13. CloudFrnt , <http://aws.amazon.com/cloudfront>
14. Simple DB , <http://aws.amazon.com/simpledb>
15. R. T. Fielding. "Architectural styles and the design of network-based software architectures". PhD thesis, University of California, Irvine, 2000
16. Amazon AutoScaling <http://aws.amazon.com/autoscaling/>
17. Amazon Elastic Load Balancing <http://aws.amazon.com/elasticloadbalancing/>
18. Rackspace CloudServers , [http://www.rackspacecloud.com/cloud\\_hosting\\_products/servers/](http://www.rackspacecloud.com/cloud_hosting_products/servers/)
19. Rackspace CloudFiles, [http://www.rackspacecloud.com/cloud\\_hosting\\_products/files/](http://www.rackspacecloud.com/cloud_hosting_products/files/)
20. Rackspace CloudServers API reference, <http://docs.rackspacecloud.com/servers/api/cs-devguide-latest.pdf>
21. VMware vCloud, <http://www.vmware.com/products/vcloud/>

22. vCloud API Programming Guide,  
[http://communities.vmware.com/servlet/JiveServlet/previewBody/12463-102-1-13007/vCloud\\_API\\_Guide.pdf](http://communities.vmware.com/servlet/JiveServlet/previewBody/12463-102-1-13007/vCloud_API_Guide.pdf)
23. Eucalyptus Block Storage,  
[http://open.eucalyptus.com/wiki/EucalyptusBlockStoreInteracting\\_v1.6](http://open.eucalyptus.com/wiki/EucalyptusBlockStoreInteracting_v1.6)
24. <http://www.opennebula.org/>
25. Carzaniga, A., Rosenblum, D. S. and Wolf, A. L., “Design and Evaluation of a Wide-Area Event Notification Service”, in: ACM Transactions on Computer Systems, 19:3 (332-383), 2001.
26. OpenStack Nova project <http://nova.openstack.org>
27. OpenStack Swift project <http://swift.openstack.org/>
28. OpenStack Nova architecture and concepts overview  
<http://nova.openstack.org/nova.concepts.html>
29. Google App Engine, <http://code.google.com/appengine/>
30. Google App Engine Java API, <http://code.google.com/appengine/docs/java/overview.html>
31. Google App Engine Python API, <http://code.google.com/appengine/docs/python>
32. Windows Azure, <http://www.microsoft.com/windowsazure/windowsazure/>
33. jClouds, <http://code.google.com/p/jclouds/>
34. Deltacloud, <http://www.deltacloud.org/>
35. libCloud, <http://ci.apache.org/projects/libcloud>
36. Workflow Management Coalition, Terminology & Glossary, Document Number WPMC-TC-1011, Issues 3.0, February 1999.
37. I. Altintas, A. Birnbaum, K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier, and B. Ludaescher, “A Framework for the Design and Reuse of Grid Workflows”, International Workshop on Scientific Applications on Grid Computing (SAG'04), LNCS 3458, Springer, 2005
38. E. Deelman, J. Blythe, Y. Gil, and C. Kesselman, “Workflow Management in GriPhyN”, The Grid Resource Management, Kluwer, Netherlands, 2003
39. E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. H. Su, K. Vahi, M. Livny, “Pegasus: Mapping Scientific Workflow onto the Grid”, Across Grids Conference 2004, Nicosia, Cyprus, 2004
40. B. Ludäscher, I. Altintas, and A. Gupta, “Compiling Abstract Scientific Workflows into Web Service Workflows”, 15th International Conference on Scientific and Statistical Database Management, Cambridge, Massachusetts, USA., IEEE CS Press, pp. 241-244, Los Alamitos, CA, USA., July 09-11, 2003
41. T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, T. Carver, M. Greenwood, K. Glover, M. R.Pocock, A. Wipat, P. Li, Taverna: a tool for the composition and enactment of bioinformaticsworkflows, Bioinformatics 20 (17) (2004) 3045–3054



42. T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. S. Jr, H. L. Truong, ASKALON: a tool set forcluster and Grid computing, *Concurrency and Computation: Practice and Experience* 17 (2005)143–169
43. I. Brandic, S. Pllana, S. Benkner, Specification, planning, and execution of QoS-aware Grid work-flows within the Amadeus environment, *Concurr. Comput. : Pract. Exper.* 20 (2008) 331–345
44. F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy,C. Kesselman, J. Mellor-Crummey, D. Reed, L. Torczon, R. Wolski, The GrADS Project: SoftwareSupport for High-Level Grid Application Development, *International Journal of High PerformanceComputing Applications(JHPCA)* 15 (4) (2001) 327–344
45. J. Eker, J. W. Janneck, E. A. Lee, L. Jie, L. Xiaojun, J. Ludvig, S. Neuendorffer, S. Sachs,X. Yuhong, Taming heterogeneity - the Ptolemy approach, *Proceedings of the IEEE* 91 (1) (2003)127–144
46. B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, Y. Zhao,Scientific workflow management and the Kepler system: *Research Articles, Concurrency and Com-putation: Practice and Experience* 18 (10) (2006) 1039–1065
47. Drools, <http://en.wikipedia.org/wiki/Drools>
48. Jbpm, <http://en.wikipedia.org/wiki/JBPM>
49. BPMN 2, <http://en.wikipedia.org/wiki/BPMN>
50. Παναγιώτης Τουρλάκης , «Ανάλυση Απόδοσης συστημάτων ροής εργασίας» 2006