



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Σύστημα διαχείρισης εμπιστοσύνης-φήμης σε ομόσπονδες ερευνητικές υποδομές

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

**Κωνσταντίνου Παπαδάκη-Βλαχοπαπαδόπουλου**

**Επιβλέπων :** Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

## Σύστημα διαχείρισης εμπιστοσύνης-φήμης σε ομόσπονδες ερευνητικές υποδομές

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**Κωνσταντίνου Παπαδάκη-Βλαχοπαπαδόπουλου**

**Επιβλέπων :** Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16<sup>η</sup> Φεβρουαρίου 2017.

(Υπογραφή)

.....  
Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Θεοδώρα Βαρβαρίγου  
Καθηγήτρια Ε.Μ.Π.

(Υπογραφή)

.....  
Ιωάννα Ρουσσάκη  
Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

(Υπογραφή)

.....

**Κωνσταντίνος Παπαδάκης-Βλαχοπαπαδόπουλος**

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

Copyright © Κωνσταντίνος Παπαδάκης-Βλαχοπαπαδόπουλος, Φεβρουάριος 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved

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

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

## Περίληψη

Στα πλαίσια του Διαδικτύου του Μέλλοντος αλλά και της ανάπτυξης νέων τεχνολογιών για αυτό, έχει δοθεί μεγάλο βάρος από την ερευνητική κοινότητα στην πειραματικά οδηγούμενη έρευνα. Για την πραγματοποίηση πειραματικά οδηγούμενης έρευνας χρειάζεται ευρεία χρήση ετερογενών ερευνητικών υποδομών ώστε να αντιμετωπιστούν με τον καλύτερο δυνατό τρόπο οι προκλήσεις η πολυπλοκότητα και η ετερογένεια του ίδιου του διαδικτύου. Για αυτόν ακριβώς τον λόγο έχουν δημιουργηθεί ομοσπονδίες ερευνητικών υποδομών. Η παρούσα εργασία εστιάζει σε λιγότερο ερευνημένο τομέα των ομόσπονδων ερευνητικών υποδομών. Σε αυτόν της διαχείρισης της φήμης και της εμπιστοσύνης των ερευνητικών υποδομών και των παρεχόμενων υπηρεσιών/πόρων η οποία και θα επιτρέψει στους ερευνητές χωρίς πρότερη άμεση εμπειρία με τις υποδομές να επιλέξουν τον καταλληλότερο συνδυασμό τους, με βάση τις υπηρεσίες/πόρους που παρέχουν και την φήμη τους για την λειτουργία τους. Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος διαχείρισης εμπιστοσύνης/φήμης σε ομοσπονδίες από ερευνητικές υποδομές. Αφού μελετήθηκαν διάφορα μοντέλα φήμης κυρίως για ομότιμα δίκτυα επιλέχθηκε η υλοποίηση του κατάλληλου μηχανισμού υπολογισμού και διαχείρισης φήμης με βάση τον οποίο υλοποιήθηκε σύστημα διαχείρισης εμπιστοσύνης/φήμης, ακολουθώντας το πρότυπο σχεδιασμού Μοντέλου-Προβολής-Ελεγκτή. Το σύστημα διαχείρισης εμπιστοσύνης/φήμης επιτρέπει την πλήρη διαχείριση των παραμέτρων καθορισμού της φήμης των ετερογενών ερευνητικών υποδομών, την υποβολή αξιολογήσεων αλλά και τον υπολογισμό της φήμης/εμπιστοσύνης. Η προτεινόμενη υλοποίηση του συστήματος είναι άμεσα εφαρμόσιμη σε ομοσπονδία ερευνητικών υποδομών αλλά παράλληλα αποτελεί και μια βάση για μελλοντικές προτάσεις μηχανισμών υπολογισμού και διαχείρισης φήμης για ομοσπονδίες ερευνητικών υποδομών.

**Λέξεις Κλειδιά:** φήμη, εμπιστοσύνη, σύστημα διαχείρισης εμπιστοσύνης/φήμης ερευνητική υποδομή, ομοσπονδοποίηση

## **Abstract**

Within the Future Internet and the research and development of new technologies for it, there has been a great interest from the research community towards test driven research. In order for this kind of research to be successful, it is absolutely necessary to combine a wide variety of heterogeneous testbeds so that it can face the challenges and complexity of the internet. For this exact reason federation of testbeds have been created and developed. In this thesis we focus in the less researched and documented aspect of federated testbeds, that is trust and reputation management of testbeds and the services and resources that they supply. This will allow to experimenters to choose efficiently, even with no prior experience with any of the testbeds, the correct combination of them basing the decision on the services/resources the testbeds supply, the experiment requirements and most significantly the reputation scores of every testbed for each service/resource. The goal of this thesis is the development of a reputation based trust management system for federated experimental infrastructures. In this context, we studied different frameworks and proposals for reputation based trust management mainly designed for peer to peer networks and finally chose an appropriate system for federated testbeds. The selected reputation evaluation algorithm was used for implementing the actual system, following a Model-View-Controller approach. The reputation based trust management system supports rating, management and reputation calculation functionality. The proposed implementation is applicable to existing environments of federated testbeds. Moreover it provides a basis for developing alternative reputation based trust management systems for federated testbeds.

**Keywords:** reputation, trust, reputation based trust management system, testbed, federation

## Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων πληροφορικής στη διάρκεια των ακαδημαϊκών ετών 2015-2016, 2016-2017 και επισφραγίζει τις σπουδές μου στο Εθνικό Μετσόβιο Πολυτεχνείο. Θα ήθελα ιδιαιτέρως να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παπαβασιλείου Συμεών για την καθοδήγηση του και για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον και επίκαιρο αντικείμενο. Επίσης θα ήθελα να ευχαριστήσω την Δρ. Παπαγιάννη Χρύσα, Ε.ΔΙ.Π στον τομέα Επικοινωνιών Ηλεκτρονικής και Συστημάτων Πληροφορικής για τον χρόνο που μου αφιέρωσε και την άριστη συνεργασία καθώς και τον Δρ. Δεχουνιώτη Δημήτρη για όλη την πολύτιμη βοήθεια που μου παρείχε.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για όλη την στήριξη και κατανόηση και τον αδερφό μου για την υποστήριξη όποτε συναντούσα δυσκολίες.

Η παρούσα εργασία αφιερώνεται στους παραπάνω, όλους μου τους φίλους και όσους συναντηθήκαμε όλα αυτά τα χρόνια στους δρόμους και στα αμφιθέατρα προκειμένου να μπορέσει η γενιά μας να κρατήσει το κεφάλι ψηλά.

## Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1.1	Εμπιστοσύνη και Φήμη στο Διαδίκτυο .....	1
1.1.2	Ομόσπονδες Ερευνητικές Υποδομές.....	4
1.2	Αντικείμενο διπλωματικής .....	6
1.2.1	Το πρόβλημα .....	6
1.2.2	Συνεισφορά.....	7
1.3	Οργάνωση κειμένου .....	8
2	Εμπιστοσύνη και φήμη σε ομότιμα δίκτυα .....	9
2.1	Τα ομότιμα δίκτυα.....	9
2.2	Διαχείριση εμπιστοσύνης βασισμένης στην φήμη σε ομότιμα δίκτυα.....	11
2.2.1	Κεντρικοποιημένα συστήματα διαχείρισης φήμης.....	11
2.2.2	Κατανεμημένα συστήματα διαχείρισης φήμης. ....	13
2.3	Μηχανισμοί διαχείρισης φήμης.....	14
2.3.1	Η εμπιστοσύνη στο eBay .....	14
2.3.2	Ο αλγόριθμος EigenTrust.....	16
2.3.3	Ο αλγόριθμος PBTrust .....	17
2.3.3	Ο αλγόριθμος ROCQ.....	22
3	Θεωρητικό υπόβαθρο .....	26
3.1	Federated Trust and User Experience Framework .....	26
3.2	Η Υπηρεσία στο FTUE .....	27
3.3	Ο αλγόριθμος του FTUE .....	28
4	Σχεδίαση/Υλοποίηση.....	33
4.1	Αρχιτεκτονική Μοντέλου-Προβολής-Ελεγκτή .....	33
4.2	Διεπαφή Δεδομένων Παρακολούθησης .....	36
4.3	Αρχιτεκτονική Συστήματος.....	36
4.4	Βάσεις Δεδομένων.....	39
5	Τεχνικές λεπτομέρειες.....	41
5.1	Διαγράμματα ακολουθίας μηνυμάτων .....	41



5.1.1	Προσθήκη ερευνητικής υποδομής και των υπηρεσιών της .....	41
5.1.2	Λήψη φήμης συγκεκριμένης ερευνητικής υποδομής .....	43
5.1.3	Διαγραφή ερευνητικής υποδομής και των υπηρεσιών της .....	44
5.1.4	Ανανέωση των στοιχείων μιας ερευνητικής υποδομής ή/και των υπηρεσιών της	45
5.1.5	Λήψη της φήμης κάθε ερευνητικής υποδομής .....	47
5.1.6	Έλεγχος βαθμολόγησης.....	48
5.1.7	Υποβολή αξιολόγησης – υπολογισμός φήμης/αξιοπιστίας. ....	49
5.2	Πλατφόρμες και προγραμματιστικά εργαλεία.....	50
6	Μελλοντικές επεκτάσεις.....	51
6.1	Ενσωμάτωση σε ομοσπονδία ερευνητικών υποδομών και δεδομένα παρακολούθησης .....	51
6.2	Εμπλουτισμός του αλγορίθμου υπολογισμού φήμης .....	51
7	Βιβλιογραφία .....	53
	Παράρτημα Α – περιγραφή rest-api .....	55

# ***1 Εισαγωγή***

## ***1.1.1 Εμπιστοσύνη και Φήμη στο Διαδίκτυο***

Η εμπιστοσύνη ενός προσώπου σε ένα άλλο. Ενός προσώπου σε μια υπηρεσία. Η φήμη ενός καταστήματος ή ενός οργανισμού, αποτελεί έννοια μεγάλης σημασίας συνολικά για τον τρόπο με τον οποίο αναπτύσσονται επαφές, συναλλαγές, ανταλλαγές. Φυσικά αυτό δεν είναι κάτι το οποίο πρωτοεμφανίστηκε στο διαδίκτυο. Αντιθέτως αποτελεί θεμελιώδες στοιχείο για την πραγματοποίηση επιλογών στην κοινωνία. Θα το εξετάσουμε λοιπόν αρχικά με ορισμένα απλά παραδείγματα από την καθημερινή ζωή. Έστω ότι θέλουμε να αγοράσουμε ένα συγκεκριμένο αντικείμενο. Η επιλογή μας για το κατάστημα/εταιρεία από το οποίο θα το αγοράσουμε θα βασιστεί είτε στο ιστορικό των προσωπικών μας αγορών και την πρότερη εμπειρία μας, είτε αν δεν υπάρχει δική μας παλαιότερη εμπειρία, στις ευρέως γνωστές πληροφορίες που υπάρχουν για τα προϊόντα του καταστήματος, την εξυπηρέτηση που παρέχει και κατ' επέκταση τη φήμη του. Ακόμη, κατά τη διάρκεια συναλλαγής με πιστωτική κάρτα, τόσο ο πελάτης όσο και ο πωλητής εμπιστεύονται ένα ηλεκτρονικό σύστημα για την διεκπεραίωση της συναλλαγής. Αν το σύστημα απορρίψει την κάρτα ο πελάτης είναι συνήθως ύποπτος και όχι το σύστημα συναλλαγής. Η αξία της φήμης και της εμπιστοσύνης μεγαλώνει στο διαδίκτυο. Ειδικά σε περιπτώσεις όπου δύο οντότητες δεν έχουν πραγματοποιήσει καμία συναλλαγή μεταξύ τους και ταυτόχρονα ο πελάτης/χρήστης διαθέτει ελλιπείς πληροφορίες σχετικά με τις παρεχόμενες υπηρεσίες ή αγαθά. Αυτή η συνθήκη, υποχρεώνει τον χρήστη να αναλάβει κάποιο ρίσκο κατά την επιλογή του αφού αυτή θα γίνει χωρίς να έχει κάποια γνώση για την επίδοση ή την ποιότητα αυτής. Π.χ να πληρώσει για προϊόντα πριν τα δει ή τα παραλάβει, να κατεβάσει κάποιο αρχείο χωρίς να γνωρίζει αν είναι κακόβουλο ή όχι κ.ο.κ. Ο χρήστης έτσι βρίσκεται σε μειονεκτική θέση αφού δεν έχει ούτε την δυνατότητα διαπροσωπικής επαφής με τον πάροχο του προϊόντος ή της υπηρεσίας, ούτε τον φυσικό έλεγχο ενός προϊόντος (π.χ. το άγγιγμα ενός ρούχου για την διαπίστωση της ποιότητας του υφάσματος, τον έλεγχο της αυθεντικότητας της μάρκας ενός προϊόντος κ.ο.κ). Αντιθέτως ο πάροχος, είναι σε πλεονεκτική θέση, αφού οποίος ορίζει και επομένως γνωρίζει ακριβώς τι θα λάβει (π.χ. το ακριβές αντίτιμο για ένα συγκεκριμένο προϊόν).

Η ανεπάρκεια του χρήστη που δημιουργείται από αυτήν την έλλειψη πληροφοριών μπορεί να μετριαστεί μέσω της εμπιστοσύνης (trust) και της φήμης (reputation) του πάροχου/πωλητή. Η κεντρική ιδέα είναι ότι ακόμη κι αν ο χρήστης δεν μπορεί να δοκιμάσει προκαταβολικά ένα προϊόν, αρχείο ή υπηρεσία, μπορεί να έχει την αυτοπεποίθηση ότι αυτά θα είναι ακριβώς ό,τι περίμενε, εφόσον εμπιστεύεται τον πάροχο/πωλητή. Έτσι, προτού ληφθεί κάποια ενέργεια, ο χρήστης μπορεί να νιώθει ασφάλεια για την επιλογή του, παρά την αβεβαιότητα και της ελλιπείς πληροφορίες που μπορεί να υπάρχουν.

Σε αυτό το σημείο θα ορίσουμε την εμπιστοσύνη(trust) και την φήμη(reputation) σύμφωνα με [1].

### **Ορισμός 1 Εμπιστοσύνη**

*Η εμπιστοσύνη είναι ο βαθμός κατά τον οποίο μία οντότητα είναι πρόθυμη να βασιστεί σε κάτι ή κάποιον σε μία δεδομένη κατάσταση, με ένα αίσθημα ασφάλειας, ακόμα και αν υπάρχει ενδεχόμενο αρνητικών επιπτώσεων.*

Η εμπιστοσύνη δηλαδή έχει να κάνει με το προσωπικό κριτήριο του καθενός και κατά πόσο με βάση αυτό θα επιλέξει να βασιστεί σε κάποιον άλλον. Το πρόβλημα που προκύπτει όμως είναι το πως δημιουργείται η εμπιστοσύνη όταν δεν υπάρχει κάποια άμεση προηγούμενη εμπειρία μιας οντότητας, με την οντότητα για την οποία καλείται να λάβει μια απόφαση. Σε αυτό το πρόβλημα έρχονται να απαντήσουν μηχανισμοί εμπιστοσύνης βασισμένοι στην φήμη.

### **Ορισμός 2 Φήμη**

*Η φήμη είναι αυτό που γενικά λέγεται ή πιστεύεται για την συμπεριφορά και την τοποθέτηση ενός ατόμου ή πράγματος*

Η διαφορά μεταξύ της εμπιστοσύνης και της φήμης μπορεί να γίνει εμφανής με το παρακάτω απλό παράδειγμα:

- “Σε εμπιστεύομαι λόγω της καλής σου φήμης” (1)
- “Σε εμπιστεύομαι παρά την κακή σου φήμη” (2)

Υποθέτοντας ότι αυτές οι δύο προτάσεις αναφέρονται σε πανομοιότυπες συναλλαγές βλέπουμε ότι στην πρώτη περίπτωση (1) η εμπλεκόμενη οντότητα γνωρίζει την φήμη της άλλης οντότητας και βασίζει την εμπιστοσύνη του σε αυτήν. Ενώ η δεύτερη περίπτωση (2) μας δείχνει ότι η εμπλεκόμενη οντότητα διαθέτει προσωπική γνώση για την άλλη οντότητα, π.χ. μέσω φιλικής σχέσης, και με βάση αυτήν απορρίπτει την όποια φήμη υπάρχει για αυτήν. Με αυτό το παράδειγμα επομένως καταλαβαίνουμε ότι η εμπιστοσύνη τελικά, είναι υποκειμενική και προσωπική και βασίζεται σε διαφορετικούς παράγοντες και ότι κάποιοι από αυτούς έχουν μεγαλύτερο βάρος από άλλους. Η προσωπική εμπειρία με μία οντότητα για

παράδειγμα, συνήθως παίζει μεγαλύτερο ρόλο σε σχέση με τις αναφορές για την εμπιστοσύνη τρίτων στην οντότητα αυτή. Επομένως η φήμη, μπορεί να θεωρηθεί το συλλογικό μέτρο της αξιοπιστίας μίας οντότητας, το οποίο προκύπτει από τις αναφορές και τις αξιολογήσεις των μελών μιας κοινότητας. Υπάρχουν δύο δομικές διαφορές μεταξύ ενός παραδοσιακού και ενός διαδικτυακού περιβάλλοντος στο πώς μπορεί να χρησιμοποιηθεί η φήμη και η εμπιστοσύνη.

1. Πρώτον όπως ήδη αναφέραμε τα παραδοσιακά στοιχεία που μας οδηγούν να εμπιστευτούμε ή να διαπιστώσουμε την φήμη κάποιου στον πραγματικό κόσμο λείπουν σε διαδικτυακά περιβάλλοντα οπότε πρέπει να δημιουργηθούν ηλεκτρονικά υποκατάστατά.
2. Η επικοινωνία και ο διαμοιρασμός πληροφορίας στον πραγματικό κόσμο σχετικά με την φήμη κάποιου είναι σχετικά δύσκολη και συνήθως περιορίζεται σε κλειστές κοινότητες.

Έτσι ένας μηχανισμός διαχείρισης, υπολογισμού και διαμοιρασμού φήμης θα πρέπει να μπορεί :

1. Να αντικαταστήσει στοιχεία του πραγματικού κόσμου που μας οδηγούν στην διαπίστωση της φήμης κάποιου με κατάλληλα στοιχεία, αναλόγως και με το είδος της εφαρμογής, τα οποία είναι ικανά για να υπολογιστεί το μέτρο της φήμης για κάτι.
2. Να αξιοποιεί το διαδίκτυο και τις υπάρχουσες τεχνικές υποδομές ώστε να μπορεί αποτελεσματικά να συλλέγει την απαραίτητη πληροφορία για τον υπολογισμό της φήμης ώστε να εξυπηρετεί την καλύτερη λήψη αποφάσεων από τους χρήστες.

Παρά τις πολλές διαφορετικές προτάσεις και προσεγγίσεις που υπάρχουν σχετικά με το πως διαμορφώνεται και υπολογίζεται η φήμη μίας οντότητας -κάποιοι από τους οποίους παρουσιάζονται και παρακάτω- η βασική ιδέα παραμένει η ίδια. Μετά από την ολοκλήρωση οποιασδήποτε συναλλαγής (χρηματική, δεδομένα κ.ο.κ) οι εμπλεκόμενες οντότητες αξιολογούν η μία την άλλη. Ο συμψηφισμός, με τρόπο που ορίζει το όποιο χρησιμοποιούμενο μοντέλο φήμης, όλων των αξιολογήσεων που έχει λάβει μια συγκεκριμένη οντότητα, μας δίνει την τιμή φήμης της συγκεκριμένης οντότητας, η οποία μπορεί να βοηθήσει κάθε μελλοντικό χρήστη να αποφασίσει στο κατά πόσον θα εμπιστευθεί η όχι την συγκεκριμένη οντότητα για κάποια συναλλαγή. Επομένως για κάθε τύπο διαδικτυακής συναλλαγής (χρηματική, αρχείων και δεδομένων, χρήση κάποια υποδομής/υπηρεσίας κ.ο.κ) είναι πολύ σημαντικό να υπάρχει το κατάλληλο μοντέλο εμπιστοσύνης για την διαδικτυακή κοινότητα στην οποία αναφέρεται έτσι ώστε να παρέχεται στον χρήστη η κατάλληλη πληροφορία με αξιοπιστία και ασφάλεια έτσι ώστε:

- να μπορεί να γίνει αποτελεσματική διαχείριση μεγάλων κατανεμημένων συστημάτων.
- Να πραγματοποιείται απρόσκοπτα και με ασφάλεια το ηλεκτρονικό εμπόριο.
- Να διασφαλίζεται η πραγματική ταυτότητα ενός χρήστη ή αρχείου.
- Να επιλέγει αποτελεσματικά ένας χρήστης την καταλληλότερη για αυτόν υπηρεσία η προϊόν.

Χαρακτηριστικά παραδείγματα τέτοιων μοντέλων συναντάμε κατά κύριο λόγο σε εφαρμογές και ομότιμα δίκτυα (peer to peer) καθώς και σε πλατφόρμες ηλεκτρονικού εμπορίου όπως το eBay [29] το Amazon [30] κ.λ.π. τα οποία θα αναλυθούν παρακάτω.

### ***1.1.2 Ομόσπονδες Ερευνητικές Υποδομές***

Στις μέρες μας και ακόμη περισσότερο όσο προχωράει η υιοθέτηση υποδομών και πρακτικών (cloud, internet of things κ.ο.κ.) του διαδικτύου του μέλλοντος (Future Internet) ολοένα και αποκεντρώνονται οι υποδομές και οι λειτουργίες που χρησιμοποιεί ο χρήστης και οι διάφορες εργασίες πραγματοποιούνται όλο και πιο κατανεμημένα. Για παράδειγμα, σε προηγούμενες φάσεις του διαδικτύου, η δυνατότητα κάποιου χρήστη να πραγματοποιήσει μια εργασία στον υπολογιστή του βασιζόταν κατά βάση σε δύο παράγοντες.

- Το διαθέσιμο υλικό του υπολογιστή(hardware)
- Το κατά πόσο διαθέτει ο χρήστης το κατάλληλο λογισμικό(software)

Αντίθετα σήμερα, όπως αναφέρθηκε και αρχικά αυτό τείνει να αναιρεθεί. Ειδικά σε κάποιες περιπτώσεις είναι δυνατό ένας χρήστης να πραγματοποιεί διάφορες εργασίες με ελάχιστους πόρους (μνήμη, επεξεργαστική ισχύ κ.λ.π) με μόνη προϋπόθεση την ύπαρξη μιας αξιόπιστης και σχετικά γρήγορης σύνδεσης με το διαδίκτυο. Για παράδειγμα έχει την δυνατότητα να αποθηκεύει μεγάλη ποσότητα δεδομένων σε κάποια υπηρεσία cloud ή να επεξεργάζεται κείμενα σε κάποια διαδικτυακή εφαρμογή (π.χ. google docs) λ.χ. Η λίστα των δυνατοτήτων αυτών καθημερινά αυξάνεται και πολλαπλασιάζεται. Αυτό επιτρέπει στους χρήστες να μην επενδύουν κεφάλαια για την αγορά υλικού και λογισμικού για την ανάπτυξη εφαρμογών. Στο πλαίσιο της ανάπτυξης γενικότερα του διαδικτύου του μέλλοντος, έχει δοθεί μεγάλο βάρος από την ακαδημαϊκή και ερευνητική κοινότητα στην ανάπτυξη πειραματικών τεχνικών ευρείας κλίμακας για την πειραματικά οδηγούμενη έρευνα. Υπό αυτό το πρίσμα δημιουργείται η ανάγκη για την δημιουργία υποδομών και εγκαταστάσεων οι οποίες θα μπορούν να υποστηρίξουν τέτοιου είδους έρευνα στα πλαίσια μια ανοιχτής και συγκροτημένης ομοσπονδίας. Μέχρι στιγμής, σε πειραματικό επίπεδο, χρησιμοποιούνται σε μεγάλο βαθμό προσομοιώσεις (simulations) και εξομοιώσεις (emulations). Οι πρώτες

αποτελούν πολύτιμο εργαλείο καθώς συνήθως είναι και οικονομικές και επιτρέπουν την γρήγορη εξαγωγή συμπερασμάτων για την συμπεριφορά, την απόδοση και την κλιμάκωση τεχνολογιών μέσω προσεγγίσεων. Οι εξομοιώσεις, προσπαθούν να αναπαραγάγουν πλήρως τη συμπεριφορά πραγματικών συστημάτων για την μέτρηση και την εξαγωγή των αντίστοιχων συμπερασμάτων. Παρ' ότι και οι δύο αυτές προσεγγίσεις αποτελούν πολύ σημαντικά εργαλεία για τους ερευνητές, λόγω των προσεγγίσεων και τις ελαχιστοποίησης των χρησιμοποιούμενων μέσων (για λογούς κόστους κ.λπ.), πολλές φορές αδυνατούν να μοντελοποιήσουν και να εξομοιώσουν/προσομοιώσουν κατ' επέκταση τις δυναμικές και την πολυπλοκότητα που θα συναντούσε ένα πειραματικό σενάριο στον πραγματικό κόσμο. Αυτό δημιουργεί την ανάγκη πραγματοποίησης πειραμάτων σε πραγματικές εγκαταστάσεις υπό πραγματικές συνθήκες σε ευρεία κλίμακα. Αυτού του είδους τα πειράματα πραγματοποιούνται πάνω σε ερευνητικές υποδομές (testbeds, experimental infrastructures). Το μεγάλο στοίχημα για την έρευνα στο διαδίκτυο του μέλλοντος είναι η αποτελεσματική ομοσπονδοποίηση γεωγραφικά διάσπαρτων ερευνητικών υποδομών προκειμένου να μπορούν, από την μία να έχουν πρόσβαση σε αυτές πολλοί ερευνητές και να διενεργούνται ποικίλα πειραματικά σενάρια και ταυτοχρόνως να δοθεί η δυνατότητα συνδυασμού ερευνητικών υποδομών για την δημιουργία πολύπλοκων και ποικιλόμορφων πειραμάτων τα οποία θα ανταποκρίνονται αντιστοίχως στην πολυπλοκότητα και την ετερογένεια του ίδιου του διαδικτύου. Σε αυτό το πλαίσιο αναπτύχθηκαν πλατφόρμες ομόσπονδων ερευνητικών υποδομών όπως το geni (<https://portal.geni.net/>) [2] και το fed4fire (<https://www.fed4fire.eu/>) [3]. Οι πλατφόρμες αυτές έχουν υλοποιηθεί με βάση την αρχιτεκτονική Slice Based Federation Architecture (SFA) [4], η οποία προσπαθεί να καταστήσει εφικτή την αποτελεσματική πρόσβαση σε πόρους διάφορων ερευνητικών υποδομών για την πραγματοποίηση πειραμάτων μεγάλης κλίμακας στα πλαίσια ομοσπονδίας ερευνητικών υποδομών. Η αρχιτεκτονική αυτή καθορίζει το ελάχιστο σύνολο διεπαφών και τύπων δεδομένων που διευκολύνουν τη δια λειτουργικότητα των συστατικών στοιχείων μιας ομοσπονδίας ερευνητικών υποδομών. Κάθε πάροχος δηλώνει την πολιτική του για την κατανομή πόρων και τη χρήση τους στο δίκτυο. Παράλληλα, οι ερευνητές για να πραγματοποιήσουν κάποιο πείραμα δεσμεύουν για κάποιο διάστημα πόρους και αποκτούν πρόσβαση σε αυτούς. Το πιο κομβικό κομμάτι αυτής της αρχιτεκτονικής είναι η έννοια της φέτας πόρων (slice), η οποία επί της ουσίας δημιουργείται από ένα σύνολο πόρων, τους οποίους έχουν συνδυάσει και έχουν αποκτήσει πρόσβαση ερευνητές, για τη διενέργεια συγκεκριμένων πειραμάτων. Δίνεται με αυτόν τον τρόπο η δυνατότητα στους ερευνητές να συνδυάσουν πόρους (resources) διαθέσιμους σε πληθώρα ερευνητικών υποδομών (testbeds), βελτιώνοντας έτσι τη δυνατότητα κλιμάκωσης και την ποικιλομορφία των διεξαγόμενων

πειραμάτων. Καθορίζοντας λοιπόν πλήρως την λειτουργικότητα γύρω από τη φέτα πόρων, ερευνητές και πάροχοι είναι ελεύθεροι να συνεργαστούν απρόσκοπτα.

Εν τέλει, η ανάπτυξη ομόσπονδων ερευνητικών υποδομών για την πειραματικά οδηγούμενη έρευνα, αποτελεί μια πολύ σημαντική αξιοποίηση του υπό διαμόρφωση διαδικτύου του μέλλοντος γιατί:

- Δίνει την δυνατότητα πρόσβασης και διενέργειας πειραμάτων σε ερευνητές, σε πραγματικές εγκαταστάσεις, στις οποίες δεν έχουν φυσική πρόσβαση η ιδιοκτησία
- Επιτρέπει τον συνδυασμό διαφορετικών και ποικιλόμορφων, γεωγραφικά διάσπαρτων υποδομών, προκειμένου να μπορούν αποτελεσματικά να γίνουν πειράματα σύνθετα με τις δυναμικές και την πολυπλοκότητα του πραγματικού κόσμου.

## ***1.2 Αντικείμενο διπλωματικής***

### ***1.2.1 Το πρόβλημα***

Όπως αναφέραμε και προηγουμένως, οι ομόσπονδες ερευνητικές υποδομές, αποτελούν πολύ σημαντικό εργαλείο για την εξέλιξη της πειραματικά οδηγούμενης έρευνας. Το μεγάλο του προτέρημα όμως, δηλαδή η δυνατότητα δημιουργίας “φέτας” (slice), με το συνδυασμό διαφορετικών υπολογιστικών και δικτυακών πόρων για την υποστήριξη πολύπλοκων πειραμάτων ευρείας κλίμακας, δημιουργεί και ένα πρόβλημα για τους ερευνητές/χρήστες. Ποιες ερευνητικές υποδομές είναι η καταλληλότερες να συνδυαστούν για κάθε ξεχωριστό πείραμα; Ειδικά για κάποιον ερευνητή ο οποίος χρησιμοποιεί για πρώτη φορά κάποια πλατφόρμα από ομόσπονδες ερευνητικές υποδομές και δεν έχει καμία άμεση και προσωπική προηγούμενη εμπειρία από καμία ερευνητική υποδομή με ποιον τρόπο θα μπορέσει να επιλέξει τον κατάλληλο συνδυασμό πόρων ώστε και να έχει επιτυχία το πείραμα του και ταυτόχρονα να μη δεσμεύσει πόρους για ένα αποτυχημένο πείραμα; Μπορούμε να ορίσουμε δύο κριτήρια τα οποία μπορούν να βοηθήσουν τον χρήστη/ερευνητή να επιλέξει.

1. Τις υπηρεσίες που παρέχει η κάθε ερευνητική υποδομή, με απλά λόγια δηλαδή του τι είδους πειράματα μπορούν να πραγματοποιηθούν σε κάθε ερευνητική υποδομή (κίνηση δικτύου, επεξεργαστική ισχύ κ.λπ.)
2. Τις επιδόσεις, την σταθερότητα και την αξιοπιστία κάθε ερευνητικής υποδομής στην κάθε υπηρεσία.

Το πρώτο κριτήριο δεν αποτελεί πρόβλημα καθώς στις πλατφόρμες αυτές κάθε ερευνητική υποδομή “διαφημίζει” τις υπηρεσίες τις οποίες υποστηρίζει και παρέχει. Το δεύτερο κριτήριο αποτελεί το βασικό πρόβλημα καθώς δεν υπάρχει κάποιος τεχνικός τρόπος ο οποίος να μπορεί με αποτελεσματικότητα να μετρήσει τις επιδόσεις. Αυτό συμβαίνει γιατί δεν αρκεί απλά η ταχύτητα δικτύου ή το εάν έχει αποσυνδέσεις μια ερευνητική υποδομή για να συμπεράνει κάποιος το αν είναι η καταλληλότερη για το πείραμα του καθώς κάθε πείραμα ανάλογα και με την φύση του επηρεάζεται με διαφορετικό τρόπο από τέτοιες παραμέτρους. Για παράδειγμα αν μία ερευνητική υποδομή χάνει την σύνδεση της το 10% του χρόνου μπορεί να έχει τελείως διαφορετικό αντίκτυπο σε ένα πείραμα εάν η διακοπή αυτή είναι συνεχόμενη ή αποτελείται από πολλές μικρές διακοπές κατά τη διάρκεια του πειράματος. Έτσι χρειάζεται κάποιος πιο σύνθετος και πολυπαραμετρικός τρόπος αξιολόγησης των υποδομών από την απλή καταγραφή τεχνικών μετρήσεων για τις επιδόσεις των υποδομών και αυτό το πρόβλημα ακριβώς προσπαθεί να επιλύσει και η συγκεκριμένη εργασία.

### ***1.2.2 Συνεισφορά***

Για την επίλυση του παραπάνω προβλήματος στα πλαίσια της συγκεκριμένης εργασίας, εμπλουτίσαμε την καταγραφή τεχνικών μετρήσεων (δεδομένων παρακολούθησης των πειραμάτων) με τις έννοιες της φήμης και της εμπιστοσύνης. Οι έννοιες της φήμης και της εμπιστοσύνης είναι πολύ σημαντικές στο διαδίκτυο καθώς επιτρέπουν στους χρήστες να πραγματοποιούν επιλογές χωρίς πρότερη εμπειρία, με βάση το ιστορικό της εμπειρίας προηγούμενων χρηστών. Στην περίπτωση μας, η φήμη χρησιμοποιείται για να καταγραφεί η συλλογική ποιότητα της εμπειρίας των χρηστών/ερευνητών με τις ερευνητικές υποδομές ώστε να διευκολυνθεί η μελλοντική επιλογή τόσο νέων όσο και παλιών χρηστών/ερευνητών. Το σύστημα που υλοποιήθηκε στην παρούσα εργασία προκειμένου να γίνουν πράξη τα παραπάνω αποτελεί μια διεπαφή για την πραγματοποίηση αξιολογήσεων από τους χρήστες, την διαχείριση των ερευνητικών υποδομών από τους διαχειριστές και την παροχή των τιμών φήμης στους χρήστες. Επιπρόσθετα η διαδικτυακή εφαρμογή μας πίσω από την διεπαφή μας (backend), υλοποιεί μηχανισμό υπολογισμού φήμης. Για την πραγματοποίηση των παραπάνω ακολουθήθηκε συνοπτικά η εξής διαδικασία.

1. Μελετήσαμε την έννοια της φήμης και της εμπιστοσύνης στο διαδίκτυο.
2. Μελετήσαμε διαφορετικές προτάσεις για τον τρόπο υπολογισμού της φήμης και της εμπιστοσύνης στο διαδίκτυο.



3. Επιλέχθηκε ο καταλληλότερος για την περίπτωση των ομόσπονδων ερευνητικών υποδομών και υλοποιήθηκε ως μέρος του συστήματος διαχείρισης εμπιστοσύνης/φήμης, με τη ανάπτυξη της αντίστοιχης διαδικτυακή εφαρμογής ακολουθώντας το πρότυπο σχεδιασμού Μοντέλου-Προβολής-Ελεγκτή.
4. Μελετήσαμε τις απαιτήσεις της ενσωμάτωσης του συστήματος διαχείρισης εμπιστοσύνης/φήμης σε μία ομόσπονδη πλατφόρμα ερευνητικών υποδομών και προσομοιώσαμε τη λειτουργία του με τη δημιουργία mocups των λοιπών εμπλεκόμενων συστημάτων.

### ***1.3 Οργάνωση κειμένου***

Το υπόλοιπο κείμενο αναπτύσσεται με την εξής λογική. Στο κεφάλαιο 2 παρουσιάζουμε σχετικές εργασίες. Παρουσιάζονται δηλαδή διάφορα μοντέλα υπολογισμού φήμης εστιάζοντας σε ομότιμα δίκτυα για την κατανόηση γενικότερα της διαδικασίας υπολογισμού φήμης αλλά και την αναζήτηση ομοιοτήτων και διαφορών με τον θεωρητικό πυρήνα της παρούσας εργασίας. Στο κεφάλαιο 3 περιγράφεται αναλυτικά ο μηχανισμός υπολογισμού φήμης που επιλέξαμε να υλοποιήσουμε ως μέρος του συστήματος διαχείρισης εμπιστοσύνης/φήμης. Στο κεφάλαιο 4 παρουσιάζονται οι σχεδιαστικές λεπτομέρειες και η επισκόπηση της αρχιτεκτονικής του συστήματος μας. Στο κεφάλαιο 5 παρουσιάζονται αναλυτικά όλες οι κλήσεις του συστήματος μας με την μορφή διαγραμμάτων ακολουθίας και επεξηγούνται αναλυτικά καθώς παρουσιάζονται με την μορφή πίνακα τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν. Στο κεφάλαιο 7 παρουσιάζεται η βιβλιογραφία και ακολουθούν ως παράρτημα όλες οι κλήσεις της εφαρμογής με πλήρη παραδείγματα για το πώς αυτές χρησιμοποιούνται.

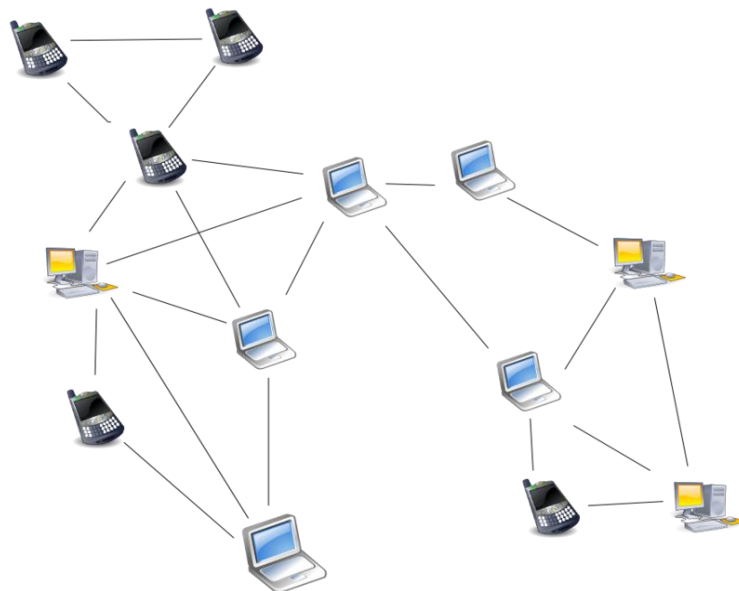
## 2 Εμπιστοσύνη και φήμη σε ομότιμα δίκτυα

### 2.1 Τα ομότιμα δίκτυα

Οι ομότιμες (peer to peer – p2p) διαδικτυακές κοινότητες μπορούν να ιδωθούν ως πραγματικά καταναλωμένες υπολογιστικές εφαρμογές στις οποίες οι χρήστες (peers) επικοινωνούν απευθείας μεταξύ τους ανταλλάσσοντας πληροφορίες, κατανέμοντας εργασίες, μοιράζοντας αρχεία ή πραγματοποιώντας συναλλαγές. Οι χρήστες (peers) αποτελούν ταυτόχρονα πάροχοι και καταναλωτές πόρων σε αντίθεση με το παραδοσιακό μοντέλο πελάτη-εξυπηρετητή (client-server) κατά το οποίο η κατανάλωση και η παροχή πόρων είναι ξεκάθαρα διαχωρισμένες. Τα ομότιμα δίκτυα τυπικά συγκροτούνται με δυναμικό τρόπο με ad-hoc προσθήκες κόμβων. Με βάση το με ποιόν τρόπο συνδέονται οι κόμβοι μεταξύ τους και την αρχιτεκτονική του δικτύου τα ομότιμα δίκτυα χωρίζονται κυρίως σε δύο κατηγορίες

- Τα αδόμητα (unstructured)
- Τα δομημένα (structured)

Τα αδόμητα ομότιμα δίκτυα δεν επιβάλλουν κάποια συγκεκριμένη δομή από σχεδίασης. Αντίθετα σχηματίζονται από κόμβους οι οποίοι τυχαία σχηματίζουν συνδέσεις μεταξύ τους.

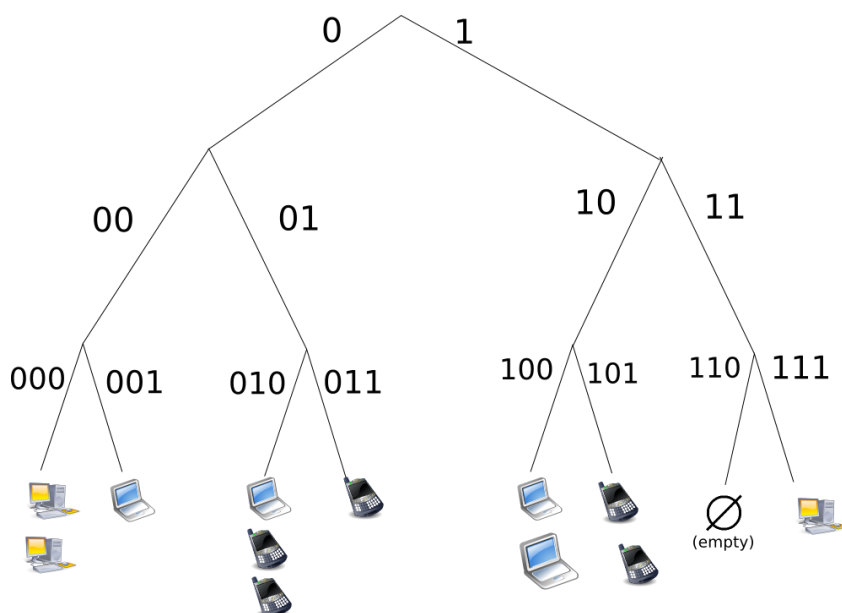


Σχήμα 2.1 – Παράδειγμα τοπολογίας αδόμητου p2p δικτύου

Μια τέτοια τοπολογία φαίνεται καθαρά στο παραπάνω σχήμα (Σχ 2.1)

Λόγω της έλλειψης συγκεκριμένης προσχεδιασμένης δομής τα αδόμητα δίκτυα αποκτούν το πλεονέκτημα της εύκολης δημιουργίας αλλά και της υψηλής ανθεκτικότητας όταν πολλοί χρήστες συχνά εισέρχονται και εξέρχονται από το δίκτυο. Η έλλειψη δομής όμως αποτελεί και το αδύναμο τους σημείο καθώς δυσκολεύει πολύ την αναζήτηση πληροφοριών/αρχείων καθώς το αίτημα πρέπει να περάσει από όλο το δίκτυο κάνοντας για αυτόν το λόγο μεγάλη χρήση πόρων χωρίς να διασφαλίζεται ταυτόχρονα ότι το αίτημα θα διεκπεραιωθεί ολοκληρωτικά αφού κατά την αναζήτηση σπάνιου περιεχομένου που κατέχουν λίγοι χρήστες, δεν υπάρχει εγγύηση ότι το αίτημα θα φτάσει σε όλους τους χρήστες.

Από την άλλη τα δομημένα ομότιμα δίκτυα είναι οργανωμένα σε συγκεκριμένες τοπολογίες και το πρωτόκολλο διασφαλίζει ότι κάθε κόμβος μπορεί αποτελεσματικά να αναζητήσει το δίκτυο για κάποιο συγκεκριμένο αρχείο, ακόμη κι αν αυτό είναι πολύ σπάνιο. Παρ' όλα αυτά για να μπορούν να δρομολογούν αποδοτικά την κίνηση δικτύου, οι κόμβοι στα δομημένα δίκτυα πρέπει να διατηρούν λίστες των γειτονικών κόμβων οι οποίοι πληρούν συγκεκριμένα κριτήρια. Αυτό τα κάνει ιδιαίτερα ευάλωτα όταν πολλοί χρήστες εισέρχονται και εξέρχονται συχνά από το δίκτυο. Τοπολογία δομημένου ομότιμου δικτύου με χρήση πίνακα Distributed Hash Table (DHT) μπορούμε να δούμε στο ακόλουθο σχήμα.



Σχήμα 2.2 – Παράδειγμα τοπολογίας δομημένου p2p δικτύου

## ***2.2 Διαχείριση εμπιστοσύνης βασισμένης στην φήμη σε ομότιμα δίκτυα***

Χωρίς να απαιτείται να επεκταθούμε παραπάνω στα ομότιμα δίκτυα είναι εμφανές ότι υπάρχει ανάγκη διαχείρισης της εμπιστοσύνης και της φήμης σε αυτά. Το χαρακτηριστικό τους στοιχείο άλλωστε, δηλαδή η δυναμική δημιουργία τους από διάφορους ξένους μεταξύ τους χρηστών, καθιστά από την μία απαραίτητη την διαδικασία αυτή για την αποφυγή και πρόληψη κακόβουλων ενεργειών και ταυτόχρονα αποτελεί μεγάλης σπουδαιότητας διαδικασία λόγω και της αποκεντρωμένης φύσης των ομότιμων συστημάτων. Η συγκεκριμένη αναγκαιότητα δεν είναι κάτι το οποίο δεν αντιλαμβάνονται και οι ίδιοι οι απλοί καθημερινοί χρήστες του διαδικτύου. Οι χρήστες του amazon για παράδειγμα ελέγχουν τις προηγούμενες κριτικές και αξιολογήσεις των χρηστών στα προϊόντα που επιθυμούν για να επιλέξουν, οι χρήστες του eBay στο τέλος κάθε δημοπρασίας η αγοραπωλησίας βαθμολογούν ο ένας τον άλλον και σχηματίζονται οι βαθμοί φήμης του καθ' ενός, ενώ σε file sharing δίκτυα όπως το Gnutella ελέγχουν τις αξιολογήσεις αρχείων πριν αρχίζουν να κατεβάζουν κάποιο για να αποφύγουν αρχεία κακόβουλο περιεχομένου (Trojans, worms κλπ). Στο πρόβλημα της σωστής αξιολόγησης στα διαφορετικά αυτά δίκτυα αποτέλεσε η λύση της διαχείρισης της εμπιστοσύνης με βάση την φήμη (reputation based trust management). Η διάφορες λύσεις και προτάσεις οι οποίες υπάρχουν μπορούμε να πούμε ότι χωρίζονται σε δύο βασικές κατηγορίες. Τα κεντροποιημένα συστήματα διαχείρισης φήμης και τα κατακεντρωμένα συστήματα διαχείρισης φήμης.

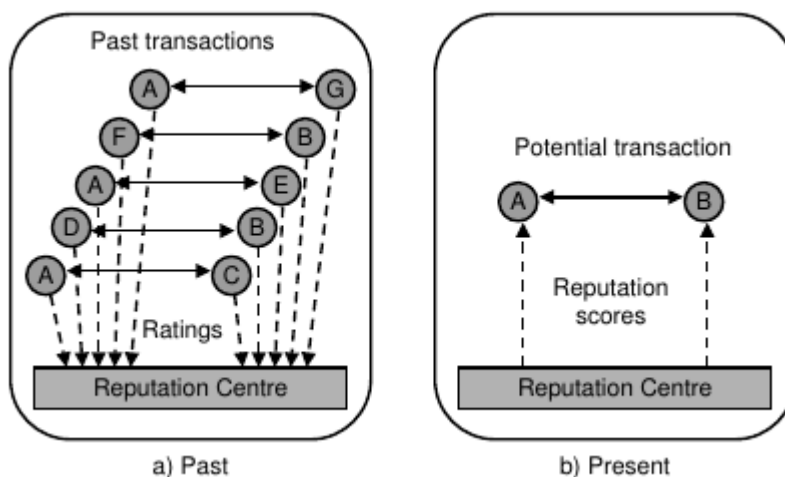
### ***2.2.1 Κεντροποιημένα συστήματα διαχείρισης φήμης.***

Στα κεντροποιημένα συστήματα διαχείρισης φήμης, οι πληροφορίες για τις επιδόσεις ενός συγκεκριμένου συμμετέχοντα συλλέγονται ως βαθμολογίες από τα υπόλοιπα μέλη της κοινότητας που είχαν κάποια εμπειρία με τον συγκεκριμένο συμμετέχοντα. Η κεντρική αρχή διαχείρισης φήμης η οποία και συλλέγει όλες τις αξιολογήσεις, υπολογίζει με βάση το χρησιμοποιούμενο μηχανισμό διαχείρισης φήμης, την τιμή φήμης κάθε συμμετέχοντα στην κοινότητα και καθιστά την τιμή φήμης όλων των συμμετεχόντων δημόσια διαθέσιμη. Έτσι οι συμμετέχοντες μπορούν να χρησιμοποιήσουν τις τιμές αυτές για να λάβουν αποφάσεις εντός της κοινότητας, όπως για παράδειγμα το αν θα προβούν σε κάποια συγκεκριμένη συναλλαγή με μία άλλη οντότητα εντός της κοινότητας. Η γενική ιδέα είναι ότι οι συναλλαγές στις οποίες συμμετέχουν χρήστες με υψηλή τιμή φήμης, είναι πολύ πιο πιθανό ή και σχεδόν βέβαιο ότι θα εξελιχθούν πολύ καλύτερα από άλλες στις οποίες συμμετέχουν οντότητες με

χαμηλό βαθμό φήμης. Στο παρακάτω σχήμα (Σχ 2.3) μπορούμε να δούμε την λειτουργία ενός τυπικού κεντρικοποιημένου πλαισίου διαχείριση φήμης (centralized reputation framework) όπου ως A και B συμβολίζουμε δύο χρήστες με ιστορικό συναλλαγών στο παρελθόν οι οποίοι σκέφτονται να πραγματοποιήσουν κάποια συναλλαγή στο παρόν. Έπειτα από κάθε συναλλαγή που είχαν στο παρελθόν οι δύο χρήστες υπέβαλλαν βαθμολογίες για την συμπεριφορά και επίδοση του καθ' ένα στην κάθε συναλλαγή. Η κεντρική αρχή διαχείρισης φήμης συλλέγει τις αξιολογήσεις και ανανεώνει στη συνέχεια της τιμή φήμης του καθένα συναρτήσει των τελευταίων βαθμολογιών. Οι τιμές αυτές καθίστανται, όπως είπαμε και προηγουμένως, διαθέσιμες στο δίκτυο και προσβάσιμες σε κάθε χρήστη, έτσι ώστε να μπορούν να αποφασίσουν σχετικά με τις μελλοντικές τους συναλλαγές.

Τα δύο θεμελιώδη χαρακτηριστικά των κεντρικοποιημένων συστημάτων διαχείρισης φήμης είναι:

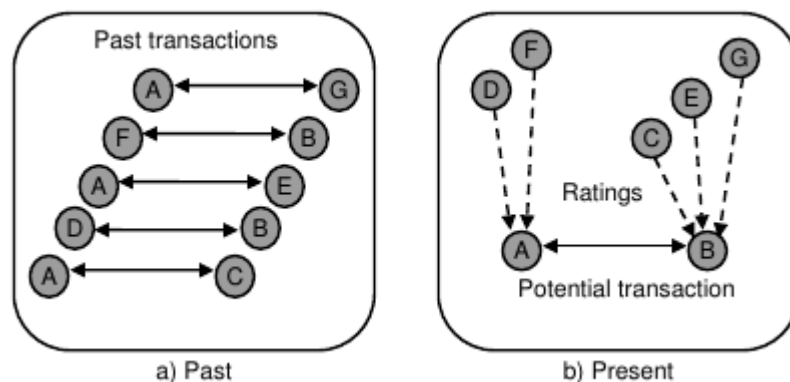
- τα κεντρικοποιημένα πρωτόκολλα επικοινωνίας τα οποία επιτρέπουν στους συμμετέχοντες να παρέχουν αξιολογήσεις για τις συναλλαγές τους και τους συμμετέχοντες σε αυτά στην κεντρική αρχή καθώς και την παροχή των τιμών φήμης κάθε χρήστη σε κάθε χρήστη
- Ένα μηχανισμό υπολογισμού της φήμης τον οποίο χρησιμοποιεί η κεντρική αρχή για να παράγει τις τιμές φήμης του κάθε χρήστη βασιζόμενη στις κατατεθειμένες αξιολογήσεις και σε κάποιες περιπτώσεις αναλόγως και με τον χρησιμοποιούμενο μηχανισμό σε συνδυασμό και με άλλες πληροφορίες.



Σχήμα 2.3 – Σχηματική αναπαράσταση κεντρικοποιημένου πλαισίου διαχείρισης φήμης

### 2.2.2 Κατανεμημένα συστήματα διαχείρισης φήμης.

Υπάρχουν συστήματα και περιβάλλοντα στα οποία ένα κατανεμημένο σύστημα διαχείρισης φήμης (δηλαδή χωρίς καμία κεντροκοποιημένη αρχή ή λειτουργία) είναι καταλληλότερα από ένα κεντροκοποιημένο σύστημα. Σε ένα κατανεμημένο σύστημα δεν υπάρχει κάποια κεντρική τοποθεσία στην οποία είτε υποβάλλουν τις βαθμολογίες τους οι χρήστες είτε ανακτούν τιμές φήμης από αυτήν. Αντιθέτως, μπορεί να αποθηκεύονται και να υποβάλλονται κατανεμημένα οι αξιολογήσεις, είτε μπορεί απλώς κάθε χρήστης να κρατά αρχείο της άποψης του για κάθε εμπειρία που είχε με άλλες οντότητες και να παρέχει αυτές τις πληροφορίες μετά από απαίτηση κάποιου άλλου χρήστη. Μία οντότητα A η οποία σκέφτεται να πραγματοποιήσει κάποια συναλλαγή με μια οντότητα B θα πρέπει να βρει που είναι κατανεμημένα αποθηκευμένες οι αξιολογήσεις είτε να προσπαθήσει να αποκτήσει τις αξιολογήσεις από όσο το δυνατόν περισσότερα μέλη της κοινότητας που είχαν κάποια άμεση εμπειρία με την οντότητα B και αντίστροφως ο B για τον A. Η δεύτερη διαδικασία παρουσιάζεται σχηματικά παρακάτω. Μετά την συλλογή των αξιολογήσεων υπολογίζεται η τιμή φήμης με βάσει τις αξιολογήσεις που κατάφερε να συλλέξει η κάθε οντότητα. Σε περίπτωση που για παράδειγμα η οντότητα A είχε στο παρελθόν κάποια απ' ευθείας εμπειρία με την οντότητα B τότε η εμπειρία αυτή και η αντίστοιχη της αξιολόγηση θα συμπεριληφθεί φυσικά στον υπολογισμό της τιμής φήμης και πιθανότατα και με μεγαλύτερο βάρος αναλόγως και με τον μηχανισμό υπολογισμού των τιμών φήμης που χρησιμοποιείται.



Σχήμα 2.4 – Σχηματική αναπαράσταση κατανεμημένου συστήματος διαχείρισης φήμης

Τα δύο θεμελιώδη χαρακτηριστικά των καταναμημένων συστημάτων διαχείρισης φήμης είναι:

- Ένα καταναμημένο πρωτόκολλο επικοινωνίας το οποίο επιτρέπει στους συμμετέχοντες να αποκτούν αξιολογήσεις από τα μέλη της κοινότητας.
- Έναν μηχανισμό υπολογισμού των τιμών φήμης τον οποίο χρησιμοποιεί κάθε ξεχωριστός πράκτορας για να υπολογίσει και να παράξει τις τιμές φήμης για τα άλλα μέλη της κοινότητας για τα οποία ενδιαφέρεται, με βάση τις ληφθέντες αξιολογήσεις ή και άλλες πληροφορίες αναλόγως και με το τι μηχανισμός υπολογισμού χρησιμοποιείται.

## **2.3 Μηχανισμοί διαχείρισης φήμης.**

Τα συστήματα διαχείρισης φήμης συνήθως βασίζονται σε δημόσιες πληροφορίες προκειμένου να αντικατοπτρίσουν την γενική άποψη της κοινότητας. Κάποια άλλα παίρνουν και δημόσιες και ιδιωτικές πληροφορίες ως είσοδο (ιδιωτική πληροφορία ονομάζουμε την πληροφορία η οποία παράχθηκε από προσωπική εμπειρία και θεωρείται πιο αξιόπιστη κατά γενική ομολογία).

Για να μπορεί ένα σύστημα διαχείρισης φήμης να συμβουλεύει αποτελεσματικά τους χρήστες σε αξιόπιστες επιλογές θα πρέπει να μπορεί με αποτελεσματικότητα να:

- Να συλλέγει πληροφορίες για την συμπεριφορά και τις συναλλαγές κάθε κόμβου.
- Να υπολογίζει την φήμη κάθε κόμβου με βάση τα στοιχεία
- Να συγκρίνει/παρουσιάζει τις τιμές αυτές.

Στη συνέχεια θα παρουσιάσουμε μοντέλα που χρησιμοποιούνται σε ομότιμα περιβάλλοντα.

### **2.3.1 Η εμπιστοσύνη στο eBay**

Το eBay είναι ίσως η πιο δημοφιλής ιστοσελίδα διαδικτυακών δημοπρασιών όπου ο καθένας μπορεί να βάλει αντικείμενα προς πώληση και οι ενδιαφερόμενοι να πλειοδοτήσουν για αυτά. Ο μηχανισμός διαχείρισης φήμης του eBay, γνωστός ως “eBay feedback forum” δίνει στον αγοραστή και στον πωλητή την δυνατότητα να αξιολογήσουν ο ένας τον άλλο ως θετικό (positive), αρνητικό (negative) ή ουδέτερο (neutral) μετά το πέρας κάθε συναλλαγής. Οι αξιολογήσεις αυτές καταχωρούνται ως 1 για θετική αξιολόγηση, -1 για αρνητική και 0 για ουδέτερη. Ακόμη οι χρήστες έχουν την δυνατότητα να εκφράσουν την γνώμη τους και με μορφή σχολίου πέραν της σκέτης βαθμολογίας με σχόλια όπως “πολύ καλή εξυπηρέτηση” σε περίπτωση θετικής γνώμης, ή “αποφεύγετε τον συγκεκριμένο πωλητή” σε περίπτωση αρνητικής εμπειρίας. Ο μηχανισμός του eBay χρησιμοποιεί κεντρική αρχή διαχείρισης της

φήμης, όπου σε αυτή συλλέγονται όλες οι βαθμολογίες και υπολογίζονται οι τιμές φήμης. Η τιμή φήμης του κάθε χρήστη υπολογίζεται από το άθροισμα των θετικών αξιολογήσεων με τις αρνητικές. Προκειμένου το eBay να παρέχει πληροφορίες όχι μόνο για την συνολική συμπεριφορά ενός χρήστη αλλά και για την πιο πρόσφατη μόνο, η κεντρική αρχή παρέχει στους χρήστες τιμές φήμης που προκύπτουν από το άθροισμα των βαθμολογιών για τα χρονικά διαστήματα: α) τελευταίοι έξι μήνες, β) τελευταίος μήνας και γ) τελευταία εβδομάδα. Έχει διαπιστωθεί με εμπειρικές μελέτες [6] ότι σε γενικές γραμμές οι βαθμολογίες που υποβάλλονται είναι εντυπωσιακά θετικές. Οι αγοραστές παρέχουν αξιολογήσεις για τους πωλητές σε ποσοστό 51,7% των πραγματοποιούμενων συναλλαγών ενώ οι πωλητές παρέχουν αξιολογήσεις για τους αγοραστές σε ποσοστό 60.6% επί του συνόλου των πραγματοποιούμενων συναλλαγών. Απ' όλες τις αξιολογήσεις που υποβάλλονται στο σύστημα, οι αρνητικές αποτελούν ποσοστό μικρότερο του 1%, οι ουδέτερες βρίσκονται σε ποσοστό μικρότερο του 0.5% και οι θετικές καταλαμβάνουν περίπου το 99% των αξιολογήσεων. Επίσης διαπιστώθηκε ότι υπάρχει μεγάλη συσχέτιση μεταξύ των αξιολογήσεων των αγοραστών και των πωλητών. Παρατηρήθηκε δηλαδή σε μεγάλο βαθμό, ότι στις θετικές αξιολογήσεις υπήρχε ανταπόδοση με θετική αξιολόγηση, ενώ σε αρνητικές αξιολογήσεις επιστρεφόταν η αρνητική ως “αντίποινα”. Αυτό είναι αρκετά προβληματικό διότι ο στόχος κάθε μηχανισμού διαχείρισης φήμης είναι πρωτίστως να λαμβάνει ειλικρινείς αξιολογήσεις. Μια λύση για το συγκεκριμένο πρόβλημα θα ήταν η μονόδρομη αξιολόγηση, δηλαδή να υπάρχει η δυνατότητα αξιολόγησης μόνο του πωλητή από τον αγοραστή και όχι το αντίστροφο. Ένα άλλο σύνηθες πρόβλημα σε συστήματα διαχείριση φήμης είναι οι πολλαπλές ψεύτικες αξιολογήσεις προκειμένου να αυξηθεί πλαστά η τιμή φήμης ενός χρήστη. Κάτι το οποίο προλαμβάνεται στην συγκεκριμένη περίπτωση διότι ο χρήστης έχει δικαίωμα να βαθμολογήσει ένα χρήστη μόνο μετά το πέρας κάποια συναλλαγής μαζί του και ενώ θα μπορούσε κάποιος να πει ότι αυτό το μέτρο μπορεί να παρακαμφθεί δημιουργώντας πλασματικές συναλλαγές, αυτό θα ήταν κάτι κοστοβόρο λόγω της προμήθειας που παίρνει από τις συναλλαγές το eBay. Παρ' όλα αυτά, οι άδικες βαθμολογίες για πραγματικές συναλλαγές είναι κάτι το οποίο δεν μπορεί να αποφευχθεί με τον συγκεκριμένο μηχανισμό. Το σύστημα φήμης του eBay είναι αρκετά πρωτόγονο και μπορεί σε πολλές περιπτώσεις να είναι παραπλανητικό. Με τόσες λίγες αρνητικές αξιολογήσεις, ένας χρήστης με 100 θετικές και 10 αρνητικές αξιολογήσεις, ενώ θα έπρεπε να φαίνεται λιγότερο αξιόπιστος σε σχέση με έναν χρήστη με 90 θετικές και καμία αρνητική αξιολόγηση, στο σύστημα του eBay λαμβάνουν την ίδια τιμή φήμης. Παρά τα ελαττώματα του και την πρωτόγονη φύση του, το σύστημα φήμης του eBay έχει επιδράσει πολύ θετικά στον αντίκτυπο που έχει συνολικότερα το eBay στην ψηφιακή αγορά καθώς οι χρήστες αντιδρούν πολύ καλά στο σύστημα του eBay και το στοιχείο διαχείρισης φήμης που διαθέτει.



### 2.3.2 Ο αλγόριθμος *EigenTrust*

Ο αλγόριθμος *EigenTrust* σχεδιάστηκε για την αποφυγή μη αυθεντικών ή και κακόβουλων αρχείων σε ομότιμα συστήματα διαμοιρασμού αρχείων. Ο *EigenTrust* διαχειρίζεται την φήμη σε ομότιμο σύστημα καθορίζοντας μια μοναδική καθολική τιμή εμπιστοσύνης (trust value) για κάθε ομότιμο κόμβο με βάση το ιστορικό προηγούμενων συναλλαγών. Ο υπολογισμός και η λήψη των τιμών φήμης πραγματοποιείται κατανεμημένα χωρίς την χρήση κάποιας κεντρικής αρχής διαχείρισης φήμης. Έτσι κάθε κόμβος αποθηκεύει μια τοπική τιμή εμπιστοσύνης για κάθε κόμβο με τον οποίο έχει πραγματοποιήσει κάποια συναλλαγή. Ως τοπική τιμή φήμης  $s_{ij}$  ορίζουμε το άθροισμα των τιμών αξιολόγησης (trust value συμβολίζεται ως  $tr$ ) για κάθε μεταφόρτωση αρχείου του κόμβου  $i$  από τον κόμβο  $j$ .

$$s_{ij} = \sum tr_{ij} \quad (2.1)$$

Κάθε φορά όπου ένας κόμβος  $i$  λαμβάνει ένα αρχείο από ένα κόμβο  $j$ , ο  $i$  μπορεί να δώσει μια τιμή  $tr(i,j)$  για να αξιολογήσει την συναλλαγή με κριτήριο την ποιότητα του αρχείου που έλαβε αλλά και το κατά πόσον αυτό ανταποκρινόταν σε αυτό που έλεγε ότι είναι. Στην περίπτωση που μιλάμε για γνήσιο αρχείο τότε αυξάνει την τοπική τιμή εμπιστοσύνης ενώ στην αντίθετη περίπτωση την μειώνει.

Προκειμένου να μπλοκαριστούν κακόβουλοι χρήστες από το να υποβάλλουν ψευδής υψηλές και θετικές τιμές εμπιστοσύνης σε κακόβουλους χρήστες είτε από το να δίνουν χαμηλές τιμές εμπιστοσύνης σε καλόβουλους χρήστες καθιστώντας έτσι τον μηχανισμό αναξιόπιστο, οι τιμές τοπικής εμπιστοσύνης κανονικοποιούνται. Έτσι η κανονικοποιημένη τοπική τιμή εμπιστοσύνης  $c_{ij}$  ορίζεται ως εξής:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)} \quad (2.2)$$

Όπου στον παρονομαστή έχουμε το άθροισμα όλων των αξιολογήσεων που έχει δώσει ο κόμβος  $i$  στις συναλλαγές του με κάθε κόμβο  $j$ . Έτσι όλες οι τιμές κανονικοποιούνται σε τιμές στο διάστημα μεταξύ 0 και 1 κάτι που διευκολύνει και τους υπολογισμούς αφού δεν χρειάζεται να κανονικοποιηθούν οι τελικές καθολικές τιμές εμπιστοσύνης των κόμβων.

Για να γίνει η συλλογή και η συνάθροιση πρέπει κάθε κόμβος  $i$  να παραλάβει μετά από αίτημα του από όλους του γνωστούς του κόμβους τη γνώμη τους για τους κόμβους με τους οποίους ο  $i$  δεν έχει αλληλεπιδράσει. Μετά από αυτήν την διαδικασία κάθε κόμβος  $i$  λαμβάνει

την τοπική τιμή εμπιστοσύνης  $t_{ik}$  του κόμβου  $i$  για τον κόμβο  $k$ , με βάση τη γνώμη των γνωστών κόμβων του  $i$  για τον κάθε κόμβο  $k$ , η οποία υπολογίζεται ως εξής:

$$\sum_j c_{ij}c_{jk} \quad (2.3)$$

Η παραπάνω σχέση μπορεί να διατυπωθεί και με συμβολισμό μητρών. Εάν ορίσουμε  $C$  την μήτρα  $[c_{ij}]$  και  $\vec{t}_i$  το διάνυσμα που περιέχει τιμές  $t_{ik}$  τότε θα ισχύει  $\vec{t}_i = C^T \vec{c}_i$ .

Οι τιμές αυτές αν και προσδίδουν στον κόμβο  $i$  μια οπτική για το δίκτυο αρκετά ευρύτερη της προσωπικής του εμπειρίας περιορίζεται στην εμπειρία των γνωστών του κόμβων. Προκειμένου να αποκτήσει ευρύτερη άποψη, ένας κόμβος θα πρέπει να ζητήσει την άποψη και των γνωστών κόμβων των γνωστών του κόμβων ( $\vec{t}_i = (C^T)^2 \vec{c}_i$ ). Εάν συνεχίσει με αυτόν τον τρόπο ( $\vec{t}_i = (C^T)^n \vec{c}_i$ ) θα αποκτήσει πλήρη άποψη για το δίκτυο μετά από  $n$  επαναλήψεις (με την προϋπόθεση ότι η μήτρα  $C$  είναι αμειώτη και απεριοδική). Εάν το  $n$  είναι μεγάλο τότε το διάνυσμα εμπιστοσύνης  $\vec{t}_i$  θα συγκλίνει στο ίδιο διάνυσμα για κάθε κόμβο  $i$ , στο αριστερό κύριο διάνυσμα του  $C$ . Με άλλα λόγια το διάνυσμα  $\vec{t}_i$  είναι ένα καθολικό διάνυσμα εμπιστοσύνης.

### 2.3.3 Ο αλγόριθμος *PBTrust*

Σε αυτήν την ενότητα θα αναλύσουμε τον αλγόριθμο *PBTrust*. Αν και ο αλγόριθμος αυτός χρησιμοποιεί ένα πολύ διαφορετικό μοντέλο για τον υπολογισμό της τιμής φήμης και της συνολικής διαχείρισης τους (σε σχέση με το *FTUE* το οποίο χρησιμοποιείται στην παρούσα εργασία και αναλύεται παρακάτω) παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς είναι σχεδιασμένο στο να αναγνωρίζει τις πιο αξιόπιστες υπηρεσίες σε συστήματα προσανατολισμένα στην παροχή υπηρεσιών (*service oriented*). Αυτό είναι σημαντικό γιατί στις ομοσπονδίες ερευνητικών υποδομών, οι ερευνητικές υποδομές χωρίζονται με βάση τις υπηρεσίες τις οποίες παρέχουν και στο πλαίσιο της παρούσας εργασίας λαμβάνουν ξεχωριστή τιμή φήμης για κάθε διαφορετική υπηρεσία. Το *PBTrust* (*Priority Based Trust*) είναι ένα μοντέλο υπολογισμού φήμης με βάση την προτεραιότητα (το βάρος που δίνεται σε κάποιο χαρακτηριστικό). Σύμφωνα λοιπόν με το μοντέλο *PBTrust* μια υπηρεσία μπορεί να περιγράψει σειρά χαρακτηριστικών όπως για παράδειγμα τιμή, χρόνος, ποιότητα κ.ο.κ. Για διαφορετικά αιτήματα χρηστών, η προτεραιότητα κάθε υπηρεσίας μπορεί να είναι διαφορετική, υπό την έννοια δηλαδή ότι αναλόγως με την εργασία που θέλει να εκτελέσει ένας χρήστης κάποιο χαρακτηριστικό είναι σημαντικότερο από κάποιο άλλο. Προκειμένου να αντιμετωπιστεί η σχέση μεταξύ των χαρακτηριστικών και των αντίστοιχων προτεραιοτήτων περιγράφεται η κάθε υπηρεσία με έναν επίσημο τρόπο. Ας υποθέσουμε ότι υπάρχουν  $n$

χαρακτηριστικά τα οποία χρησιμοποιούνται για να πραγματοποιηθεί η περιγραφή μιας υπηρεσίας και κάθε χαρακτηριστικό ζητείται με μία συγκεκριμένη προτεραιότητα ως συνθήκη για την ολοκλήρωση της υπηρεσίας. Κάθε υπηρεσία αναπαρίσταται από  $n$  χαρακτηριστικά και τις αντίστοιχες προτεραιότητες τους.

### Ορισμός 1

Η περιγραφή μιας υπηρεσίας συμβολίζεται με  $SDes$  (Service description) και αναπαρίσταται με την ακόλουθη μορφή πίνακα

$$SDes = \begin{pmatrix} A_1 & A_2 & A_3 & \dots & A_n \\ W_1 & W_2 & W_3 & \dots & W_n \end{pmatrix} \quad (2.4)$$

Όπου με  $A_i$  συμβολίζουμε το  $i$  χαρακτηριστικό και με  $W_i$  την προτεραιότητα του χαρακτηριστικού.

Για τις προτεραιότητες ισχύει:

$$\sum_{i=1}^n W_i = 1 \quad (2.5)$$

### Ορισμός 2

Οι αξιολογήσεις (*Ratings*) αναπαριστούν την βαθμολογία που έχει δώσει σε έναν πάροχο υπηρεσίας για μια συγκεκριμένη υπηρεσία ένας χρήστης. Οι αξιολογήσεις συμβολίζονται με  $R_1, R_2, \dots, R_n$  όπου κάθε  $R_i$  αναπαριστά τον βαθμό του  $i$  χαρακτηριστικού της υπηρεσίας. Οι τιμές του κάθε  $R_i$  κυμαίνονται στο διάστημα  $[0,100]$  όπου το 0 αντιπροσωπεύει τη χειρότερη επίδοση και το 100 την καλύτερη επίδοση του χαρακτηριστικού  $i$ .

### Ορισμός 3

Η εμπειρία εξυπηρέτησης (*service experience*) ενός πάροχου για μια συγκεκριμένη υπηρεσία και ορίζεται ως  $Exp = \langle SRate, SNum \rangle$ , όπου το  $SRate$  αναπαριστά το ποσοστό επιτυχίας του πάροχου για την συγκεκριμένη υπηρεσία και το  $SNum$  τον συνολικό αριθμό επιτυχιών για την ίδια υπηρεσία.

### Ορισμός 4

Το αίτημα υπηρεσίας (*service request*) ορίζεται ως  $SR = \langle CID, SDes, RN, Sthreshold \rangle$  όπου  $CID$  η ταυτότητα (ID number) του χρήστη που καταναλώνει την υπηρεσία,  $SDes$  η υπηρεσία ως ένας  $2 \times n$  πίνακας όπως αυτός ορίστηκε στον ορισμό 2 και  $Sthreshold$  το κατώφλι του ποσοστού επιτυχίας ενός παρόχου προκειμένου να πληροί τις προϋποθέσεις για την παρεχόμενη υπηρεσία.

## Ορισμός 5

Η αναφορά Rf ορίζεται ως  $RF = \langle RfID, SDes, Ratings, T \rangle$  όπου RfID είναι η ταυτότητα (ID number) του διαιτητή (βαθμολογητή), SDes όπως ορίζεται και στον ορισμό 1 είναι η περιγραφή της υπηρεσίας από τον διαιτητή, Ratings η επίδοση (βαθμολογία) κάθε χαρακτηριστικού της υπηρεσίας (βλ ορισμό 2) όπως δόθηκε από τον διαιτητή, και τέλος, T η χρονική στιγμή της ολοκλήρωσης της υπηρεσίας.

## Ορισμός 6

Η απάντηση υπηρεσίας (service reply) ορίζεται ως  $SR = \langle SPID, RfSet, Exp \rangle$ , όπου SPID η ταυτότητα (ID number) του πάροχου υπηρεσίας (service provider), RfSet το σετ αναφορών συμπεριλαμβανομένων πολλών καλών αναφορών που δόθηκαν στον πάροχο από άλλους διαιτητές στο παρελθόν και τέλος EXP, η εμπειρία εξυπηρέτησης (service experience) η οποία όπως ορίστηκε στον ορισμό 3 αναπαριστά τη συνολική επίδοση σε μια υπηρεσία.

Το PBTrust αποτελείται από 4 μονάδες. Την μονάδα αιτήματος (Request Module), την μονάδα απάντησης (Reply Module), την μονάδα υπολογισμού εμπιστοσύνης με βάση την προτεραιότητα (Priority-Based Trust Calculation), και την μονάδα αξιολόγησης (Evaluation Module).

### A. Μονάδα Αιτημάτων

Ο σκοπός της μονάδας αιτημάτων (Request Module) είναι η δημιουργία αιτήματος υπηρεσίας (Service Request) με βάση το αίτημα ενός καταναλωτή. Για παράδειγμα ο καταναλωτής C αιτείται σε πλατφόρμα ηλεκτρονικού εμπορίου μια υπηρεσία που περιγράφεται από τρία χαρακτηριστικά, (κόστος, ταχύτητα και ποιότητα) με τις προτεραιότητες (0.3, 0.5, 0.2) αντίστοιχα. Επίσης ο καταναλωτής C αιτείται 2 αναφορές για έναν πιθανό πάροχο με το ποσοστό επιτυχίας του πάροχου για την συγκεκριμένη υπηρεσία να είναι τουλάχιστον 70%.

Με βάση αυτά η μονάδα αιτημάτων θα παράξει μια συγκεκριμένη περιγραφή, βάσει του ορισμού 1, την ακόλουθη:

$$SDes = \begin{pmatrix} Cost & Speed & Quality \\ 0.3 & 0.5 & 0.2 \end{pmatrix} \quad (2.6)$$

Έπειτα, ένα αίτημα θα παραχθεί με βάση την περιγραφή και τις προδιαγραφές του καταναλωτή στη μορφή που ορίζει ο ορισμός 4.

Δηλαδή,  $SR = \langle C, SDes, 2, 0.7 \rangle$

Το παραπάνω παράδειγμα θα χρησιμοποιηθεί για την επεξήγηση και των υπόλοιπων μονάδων (modules).

## **B. Μονάδα Απόκρισης**

Όταν ένας πιθανός πάροχος P μπορεί να παρέχει την υπηρεσία, δηλαδή πληροί τα κριτήρια του καταναλωτή C, ο πάροχος P θα δώσει τις ακόλουθες πληροφορίες: την ταυτότητα του πάροχου (Provider ID), δύο αναφορές (reference reports), καθώς και την εμπειρία εξυπηρέτησης (service experience). Έστω λοιπόν ότι ο P έχει λάβει 3 αναφορές για προηγούμενες επιδόσεις του για την ίδια υπηρεσία από διαφορετικούς καταναλωτές οι οποίες αναπαρίστανται από ένα σετ  $\{Rf1, Rf2, Rf3\}$ , και κάθε στοιχείο του σετ είναι στην μορφή που ορίζει ο ορισμός 5. Ο πάροχος P θα επιλέξει τις δύο καλύτερες αναφορές που αντιπροσωπεύουν προηγούμενες επιδόσεις του με την υπηρεσία, έστω η Rf2 και Rf3. Ας υποθέσουμε ότι το ποσοστό επιτυχίας του P στη συγκεκριμένη υπηρεσία είναι 70% και οι συνολικές του επιτυχίες είναι 35. Οι πληροφορίες που θα δώσει ο P απαντώντας στο αίτημα του καταναλωτή C θα είναι η ακόλουθη (βλ. ορισμό 6).

$SR = \langle P, \{Rf2, Rf3\}, (0.7, 35) \rangle$

Εάν παραπάνω από ένας πάροχος της συγκεκριμένης υπηρεσίας είναι διαθέσιμοι και προτίθενται να παρέχουν την υπηρεσία, η παρούσα μονάδα (Reply Module) θα παράξει παραπάνω από μία απαντήσεις

## **Γ. Μονάδα Υπολογισμού Εμπιστοσύνης με βάση την Προτεραιότητα**

Η μονάδα υπολογισμού εμπιστοσύνης με βάση την προτεραιότητα (Priority-based Trust Calculation Module) αποτελεί τον πυρήνα του μοντέλου PBTrust. Ο κύριος σκοπός αυτής της μονάδας είναι να υπολογίσει τις τιμές εμπιστοσύνης των πιθανών πάροχων βασιζόμενη στις αναφορές τρίτων, την εμπειρία εξυπηρέτησης (service experience) που παρέχουν, την χρονική τιμή της αναφοράς και τέλος τις ομοιότητες μεταξύ της ζητούμενης περιγραφής της υπηρεσίας και των αναφορών ως προς την προτεραιότητα του πάροχου σε ίδια χαρακτηριστικά. Αυτές οι τιμές εμπιστοσύνης θα βοηθήσουν τον καταναλωτή να επιλέξει τον καλύτερο πάροχο τον οποίο μπορεί να εμπιστευτεί ο καταναλωτής για να ολοκληρώσει την υπηρεσία που επιθυμεί. Οι τελικές τιμές εμπιστοσύνης για κάθε πιθανό πάροχο παράγονται από αρκετά αποτελέσματα υπολογισμών σε τέσσερις οπτικές, τις ακόλουθες.

- Την εμπειρία του παρόχου στην υπηρεσία
- Την ομοιότητα της κατανομής των προτεραιοτήτων στα χαρακτηριστικά μεταξύ της αναφοράς για τον πάροχο και τη ζητούμενη υπηρεσία,
- την καταλληλότητα του πιθανού παρόχου για την ζητούμενη υπηρεσία
- Την χρονική απόδοση των αξιολογήσεων τρίτων

Οι οπτικές αυτές συνεισφέρουν στην τελική τιμή φήμης με διαφορετικό τρόπο η κάθε μία και ορίζονται με διαφορετικούς μαθηματικούς τύπους. Ο ακριβής τρόπος υπολογισμού, δεν θα

αναλυθεί περισσότερο στα πλαίσια της συγκεκριμένης εργασίας καθώς δεν μας ενδιαφέρει τόσο αυτό το κομμάτι από την συγκεκριμένη πρόταση για την διαχείριση της φήμης και της εμπιστοσύνης, αλλά περισσότερο μας ενδιαφέρει η αντίληψη του διαχωρισμού και της αξιολόγησης παρόχων με βάση την υπηρεσία, η οποία μάλιστα μπορεί να είναι και παράγωγο ενός η και περισσότερων χαρακτηριστικών. Ο τρόπος υπολογισμού επεξηγείται αναλυτικά στην σχετική δημοσίευση όπως σημειώνεται και στο τμήμα της βιβλιογραφίας της εργασίας για όποιον επιθυμεί να εμβαθύνει στο συγκεκριμένο μοντέλο.

#### **Δ. Μονάδα Αξιολόγησης**

Η μονάδα αξιολόγησης (Evaluation Module) περιέχει δύο στοιχεία. Ένα είναι να παράξει αναφορά από έναν καταναλωτή για έναν πάροχο με βάση την επίδοση σε μία ολοκληρωμένη υπηρεσία και το άλλο είναι η ανανέωση του αρχείου της εμπειρίας εξυπηρέτησης (service experience) ενός παρόχου όπου υπάρχει καινούργια αναφορά για αυτόν.

##### **1) Παραγωγή αναφοράς**

Θα χρησιμοποιήσουμε το ίδιο παράδειγμα που χρησιμοποιήσαμε και στο Request Module και στο Reply Module για να επιδείξουμε πως παράγονται αναφορές από τη συγκεκριμένη μονάδα. Μετά την ολοκλήρωση της ζητούμενης υπηρεσίας, ο καταναλωτής C αξιολογεί την επίδοση του παρόχου P. Το αποτέλεσμα της αξιολόγησής αναπαρίσταται ως αναφορά με βάση τον ορισμό 5 όπως φαίνεται παρακάτω

$Rf = \langle C, SDes, \langle 60, 40, 90 \rangle, 12/7/2008 \rangle$

Η παραπάνω αναφορά δείχνει την αξιολόγηση όπως προέκυψε από τον καταναλωτή C για τις υπηρεσίες SDes οι οποίες ολοκληρώθηκαν στις 12 Ιουλίου του 2008. Από την βαθμολογία του καταναλωτή C μπορούμε να δούμε ότι δεν ήταν ικανοποιημένος με την ταχύτητα της υπηρεσίας (δεύτερο χαρακτηριστικό) , ήταν πολύ ικανοποιημένος με την ποιότητα της υπηρεσίας (τρίτο χαρακτηριστικό) και ήταν ικανοποιημένος με το κόστος της υπηρεσίας (πρώτο χαρακτηριστικό)

##### **2) Ανανέωση της εμπειρίας εξυπηρέτησης**

Η ανανέωση της εμπειρίας εξυπηρέτησης (service experience) βασίζεται στην κρίση του καταναλωτή για την προσφάτως ολοκληρωμένη υπηρεσία από τον πάροχο. Το αποτέλεσμα της κρίσης μπορεί να λάβει δύο τιμές. Είτε επιτυχία, είτε αποτυχία ανάλογα και με την επίδοση του παρόχου. Η τιμή Exp του service experience περιέχει δύο στοιχεία SNum και SRate (βλ ορισμό 3). Το SNum και το Srate μπορούν να ανανεωθούν από τους ακόλουθους τύπους

$$SNum = \begin{cases} SNum' + 1 & \text{judgement : success} \\ Snum' & \text{judgement : fail} \end{cases} \quad (2.7)$$

$$SRate = \begin{cases} \frac{SNum' + 1}{Srate' + 1} & \text{judgement : success} \\ \frac{SNum'}{Srate' + 1} & \text{judgement : fail} \end{cases} \quad (2.8)$$

Όπου το  $SNum'$  και το  $Srate'$  αντιπροσωπεύουν τον συνολικό αριθμό επιτυχιών και το ποσοστό επιτυχίας πριν την ανανέωση αντιστοίχως.

Έστω ότι ο καταναλωτής  $C$  είναι ικανοποιημένος από την υπηρεσία που παρέχει ο πάροχος  $P$ , ο  $C$  θα δώσει αξιολόγηση ως ‘επιτυχημένη’. Σε αυτήν την περίπτωση θα χρησιμοποιηθούν οι τύποι (2.7) και (2.8) για την ανανέωση του αρχείου του  $P$  από (0.7, 35) σε (0.706, 36).

Στην περίπτωση που δεν είναι ικανοποιημένος με την υπηρεσία ο  $C$  θα δώσει αξιολόγηση ως “αποτυχημένη” για τον  $P$  και την συγκεκριμένη υπηρεσία. Σε αυτήν την περίπτωση, με χρήση των τύπων (2.7), (2.8) θα ανανεωθεί το αρχείο του με τις τιμές (0.686, 35).

### 2.3.3 Ο αλγόριθμος ROCQ

Σε αυτήν την ενότητα θα παρουσιάσουμε τον αλγόριθμο ROCQ [9] όπως σχεδιάστηκε για ομότιμα δίκτυα ο οποίος παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς αποτελεί και την βάση πάνω στην οποία στηρίχθηκε ο μηχανισμός υπολογισμού τιμής φήμης που χρησιμοποιήθηκε στην συγκεκριμένη εργασία. Ο αλγόριθμος ROCQ αν και σχεδιάστηκε κυρίως για ομότιμες εφαρμογές και δίκτυα, τροποποιώντας την αρχιτεκτονική και την υλοποίηση μπορεί να χρησιμοποιηθεί σε άλλου τύπου δίκτυα και κοινότητες είτε με κεντρική αρχή διαχείρισης της φήμης είτε χωρίς.

Στην περίπτωση μας με τις κατάλληλες τροποποιήσεις στον αλγόριθμο όπως αυτός περιγράφεται στο επόμενο κεφάλαιο υλοποιήθηκε για ομοσπονδία ερευνητικών υποδομών με χρήση κεντρικής αρχής διαχείρισης φήμης.

Ο αλγόριθμος ROCQ όπως διατυπώθηκε από τους Anurag Garg Roberto Battiti έχει τέσσερα βασικά συστατικά στοιχεία.

- Φήμη (Reputation)
- Άποψη (Opinion)
- Αξιοπιστία (Credibility)
- Ποιότητα (Quality)

τα οποία και θα παρουσιάσουμε αναλυτικότερα παρακάτω.

### Άποψη

Ένας κόμβος σχηματίζει άποψη (opinion) για το κατά πόσο έμεινε ικανοποιημένος από κάποια συναλλαγή στην οποία συμμετείχε. Με τον συμβολισμό  $O_{ij}^k$  ως την άποψη του κόμβου  $i$  για την συναλλαγή  $k$  με το χρήστη  $j$ .

Ορίζεται ακόμα η μέση άποψη του κόμβου η οποία συμβολίζεται ως  $O_{ij}^{avg}$  η οποία είναι η μέση άποψη του κόμβου  $i$  για κάθε συναλλαγή του με τον κόμβο  $j$ . Η μέση άποψη υπολογίζεται ως απλός μέσος όρος της άποψης του κόμβου για κάθε ξεχωριστή συναλλαγή διά του συνολικού αριθμού ( $N_{ij}$ ) των συναλλαγών όπως φαίνεται στην από κάτω σχέση.

$$O_{ij}^{avg} = \frac{\sum_k O_{ij}^k}{N_{ij}} \quad (2.9)$$

Κάθε άποψη  $O_{ij}^k$  παίρνει τιμές στο διάστημα  $[0,1]$ .

### Φήμη

Η φήμη (reputation) ενός κόμβου  $i$  είναι αποτέλεσμα της συνάθροισης των απόψεων διάφορων άλλων χρηστών για τον χρήστη  $i$ . Η τιμή φήμης αντιπροσωπεύει την μέση “ικανοποίηση” που εκτιμάται, από όλες τις απόψεις του σε ένα σύστημα, ότι θα λάβει ένας κόμβος αν εμπλακεί σε κάποια συναλλαγή με τον χρήστη  $i$ . Η τιμή φήμης ενός κόμβου κανονικοποιείται ώστε να κυμαίνεται στο διάστημα  $[0,1]$  με το 0 να σημαίνει ότι ο χρήστης αυτός εκτιμάται ότι δεν θα συμπεριφερθεί ποτέ ικανοποιητικά, και το 1 ότι ο χρήστης εκτιμάται ότι θα συμπεριφερθεί πάντα ικανοποιητικά. Αν και το παραπάνω παράδειγμα αναφέρεται σε συμπεριφορά χρηστών ομότιμων δικτύων το μοντέλο μπορεί να προσαρμοστεί και σε άλλες χρήσεις. Όπως αναφέρουν και οι δημιουργοί του σε άλλες περιπτώσεις μπορεί να αντιπροσωπεύει την ποιότητα υπηρεσίας που παρέχει ένας κόμβος ή στην περίπτωση μας (με τις κατάλληλες τροποποιήσεις ειδικά) να αντιπροσωπεύει το πόσο καλά εκτιμάται ότι θα συμπεριφερθεί μια ερευνητική υποδομή σε μία συγκεκριμένη υπηρεσία που παρέχει. Η φήμη ενός χρήστη υπολογίζεται χρησιμοποιώντας τις απόψεις όλων των κόμβων που έχουν εμπλακεί μαζί του στο παρελθόν. Οι απόψεις σταθμίζονται από την αξιοπιστία (credibility) του χρήστη και την ποιότητα (quality). Σε ένα ομότιμο δίκτυο πλήρως καταναμημένο η φήμη θα υπολογίζεται από κάθε κόμβο ξεχωριστά ελλείψει μιας κεντρικής αρχής διαχείρισης της φήμης. Έτσι η φήμη ενός κόμβου  $j$  υπολογίζεται σε ένα κόμβο  $m$  ως εξής:

$$R_{mj} = \frac{\sum_i O_{ij}^{avg} \cdot C_{mi} \cdot Q_{ij}}{\sum_i C_{mi} \cdot Q_{ij}} \quad (2.10)$$



Όπου  $R_{mj}$  είναι η φήμη του κόμβου  $j$ ,  $C_{mi}$  η αξιοπιστία του κόμβου  $i$  σύμφωνα με τον κόμβο  $m$ ,  $O_{ij}^{avg}$  είναι η μέση άποψη για το κόμβο  $j$  από τον κόμβο  $i$  και  $Q_{ij}$  η τιμή της ποιότητας του κόμβου  $j$  όπως την ανέφερε ο κόμβος  $i$ .

### Αξιοπιστία

Εάν ένας κόμβος υπολόγιζε την φήμη ενός άλλου παίρνοντας απλά τον μέσο όρο των απόψεων των άλλων κόμβων για αυτόν, το σύστημα θα ήταν ιδιαίτερα διαβλητό σε διαφόρων ειδών επιθέσεις καθώς δεν θα υπήρχε κανένας τρόπος να διαχωριστούν οι αληθείς από τις ψευδείς βαθμολογίες. Η μεταβλητή αξιοπιστίας ενός κόμβου είναι ανεξάρτητη από την τιμή φήμης του καθώς ένας χρήστης μπορεί να συμπεριφέρεται καλοπροαίρετα στις συναλλαγές του, ή να έχει καλές επιδόσεις στην παροχή κάποιας υπηρεσίας αλλά ταυτόχρονα να δίνει κακοπροαίρετες κριτικές σε άλλους κόμβους. Στον μηχανισμό του ROCQ η αξιοπιστία (credibility) ενός κόμβου χρησιμοποιείται για να σταθμιστεί η άποψη που καταθέτει για κάποιο κόμβο. Εάν ένας κόμβος δίνει λανθασμένες κριτικές για άλλους κόμβους η αξιοπιστία του μειώνεται και οι απόψεις του επομένως όπως φαίνεται και από την σχέση (2.10) έχουν περιορισμένη επίδραση στον υπολογισμό της φήμης κάποιου άλλου κόμβου. Ομοίως αν με συνέπεια δίνει σωστές κριτικές η αξιοπιστία του ανεβαίνει. Οι τιμές αξιοπιστίας βασίζονται στις άμεσες εμπειρίες ενός κόμβου και αντίθετα με την άποψη (opinion) δεν διαμοιράζονται σε άλλους κόμβους. Οι τιμές αξιοπιστίας κανονικοποιούνται ώστε να κυμαίνονται στο διάστημα  $[0,1]$ . Η τιμή αξιοπιστίας  $C_{ij}$  του κόμβου  $j$ , αντιπροσωπεύει την αυτοπεποίθηση του κόμβου  $i$ , ως προς την ειλικρίνεια η μη της άποψης του κόμβου  $j$  για άλλους κόμβους. Κάθε άποψη του κόμβου  $j$  σταθμίζεται από την τιμή αξιοπιστίας όπως αναφέραμε και προηγουμένως. Κάθε φορά που ένας κόμβος ανανεώνει την τιμή αξιοπιστίας κάποιου κόμβου χρησιμοποιεί την τιμή της ποιότητας (quality) για να αποφασίσει το ποσό της μεταβολής. Όταν ένας κόμβος εκφράζει άποψη για έναν άλλο κόμβο για πρώτη φορά η τιμή αξιοπιστίας του τίθεται στο 0.5. Από αυτήν τη στιγμή κι έπειτα η αξιοπιστία του προσαρμόζεται σύμφωνα με τις παρακάτω σχέσεις.

$$C_{mi}^{k+1} = \begin{cases} C_{mi}^k + \frac{(1 - C_{mi}^k)Q_{ij}}{2} & \text{if } |R_{mj} - O_{ij}^{avg}| < s_{mj} \\ C_{mi}^k - \frac{C_{mi}^k \cdot Q_{ij}}{2} & \text{if } |R_{mj} - O_{ij}^{avg}| > s_{mj} \end{cases} \quad (2.11)$$

Όπου  $C_{mi}^k$  είναι η αξιοπιστία του κόμβου  $i$  στα μάτια του κόμβου  $m$  μετά από  $k$  για τον κόμβο  $m$ ,  $O_{ij}^{avg}$  η άποψη του κόμβου  $i$  για τον κόμβο  $j$ ,  $Q_{ij}$  η σχετική τιμή ποιότητας, και  $R_{mj}$  η τιμή φήμης του κόμβου  $j$  όπως την υπολόγισε ο κόμβος  $m$  και τέλος  $s_{mj}$  η τυπική απόκλιση των απόψεων που έχουν αναφερθεί στον κόμβο  $m$  για τον κόμβο  $j$ . Έτσι αν η

άποψη του  $i$  ταιριάζει με αυτή των υπόλοιπων χρηστών ( $|R_{mj} - O_{ij}^{avg}| < s_{mj}$ ), η τιμή αξιοπιστίας αυξάνεται, αν πάλι απέχει ( $|R_{mj} - O_{ij}^{avg}| > s_{mj}$ ) η τιμή αξιοπιστίας μειώνεται.

### **Ποιότητα**

Σε αντίθεση με άλλους μηχανισμούς εμπιστοσύνης, ο μηχανισμός του ROCQ επιτρέπει σε κάθε κόμβο να προσδιορίσει την αυτοπεποίθηση του σε σχέση με την αξιολόγηση του.

Αυτό εξυπηρετεί δύο σημαντικούς σκοπούς. Πρώτον, η παροχή λάθος αξιολογήσεων μπορεί να μειώσει την αξιοπιστία ενός κόμβου. Επομένως, ένας κόμβος μπορεί να μειώσει την τιμή ποιότητας όταν δεν είναι πολύ σίγουρος για κάποια του άποψη και έτσι σε περίπτωση που η άποψη του είναι όντως λανθασμένη, η αξιοπιστία του θα μειωθεί λιγότερο. Όπως και όλες οι άλλες παράμετροι, η ποιότητα είναι κανονικοποιημένη ώστε να κυμαίνεται στο διάστημα  $[0,1]$ .

## 3 Θεωρητικό υπόβαθρο

### 3.1 Federated Trust and User Experience Framework

Η μελέτη Reputation-Based Trust in federated testbeds utilizing user experience [10] (από εδώ και στο εξής θα αναφερόμαστε σε αυτό ως FTUE) αποτελεί την αλγοριθμική βάση και πρακτικά τον μηχανισμό με τον οποίο υπολογίζει η πλατφόρμα μας αφενός την φήμη της κάθε ερευνητικής υποδομής και αφετέρου την αξιοπιστία κάθε χρήστη με συνδυασμό (i) της εμπειρίας του (user experience) μέσω αξιολόγησης την οποία αποστέλλει σε πλατφόρμα κατά τη διάρκεια χρήσης της υποδομής, με (ii) τα “δεδομένα παρακολούθησης” (monitoring data) για κάθε ερευνητική υποδομή κατά τη διάρκεια του πειράματος.

Βασικά πλεονεκτήματα του FTUE είναι:

- Ότι σχεδιάστηκε και προορίζεται όχι για την απλή αξιολόγηση μεμονωμένων ερευνητικών υποδομών αλλά για ομοσπονδίες από ερευνητικές υποδομές.
- Ότι είναι ανθεκτικό σε κακόβουλες αξιολογήσεις κι επιθέσεις.

Το παρόν πλαίσιο (framework) προσπαθεί να λύσει ένα βασικό πρόβλημα σε πλατφόρμες ομόσπονδων ερευνητικών υποδομών. Αυτό της φήμης των ερευνητικών υποδομών και της εμπιστοσύνης των χρηστών.

Στην παρούσα εργασία και φυσικά στο συγκεκριμένο πλαίσιο, ο στόχος είναι κάθε χρήστη/ερευνητή να διευκολύνεται, να επιλέξει την καταλληλότερη ερευνητική υποδομή για το πείραμα του, από την φήμη της κάθε ερευνητικής υποδομής για την υπηρεσία ή τους πόρους που παρέχει. Για να επιτευχθεί αποτελεσματικά αυτό το FTUE επεκτείνει την έννοια της φήμης συμπεριλαμβάνοντας και την έννοια της ποιότητας της εμπειρίας του χρήστη (Quality of Experience – QoE), η οποία ως μέγεθος υποδεικνύει το επίπεδο ικανοποίησης του χρήστη από την παρεχόμενη/ους υπηρεσία/πόρους. Έτσι, με βάση το QoE και την ποιότητα της υπηρεσίας (Quality of Service – QoS) υπολογίζεται η τιμή της φήμης (με τρόπο που θα αναλυθεί παρακάτω) κάθε παρεχόμενης υπηρεσίας στην ομοσπονδία. Έτσι, με βάση αυτές ο κάθε ερευνητής μπορεί να επιλέξει από μια μεγάλη ποικιλία από ερευνητικές υποδομές οι οποίες μπορεί να διαφέρουν σε συγκεκριμένα ποιοτικά χαρακτηριστικά και να επιλέξει με βάση την ιδιαίτερη σημασία που δίνει σε συγκεκριμένα επιμέρους χαρακτηριστικά και τη φήμη αυτών.

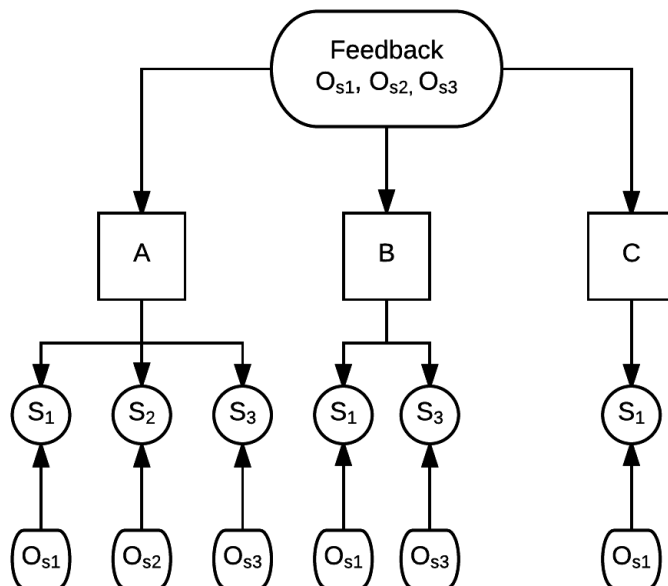
### 3.2 Η Υπηρεσία στο FTUE

Για την αξιολόγηση των ερευνητικών υποδομών το FTUE τα χωρίζει σε σεντ υπηρεσιών που παρέχουν. Μία υπηρεσία στο FTUE είναι ένα τεχνικό χαρακτηριστικό το οποίο απορρέει από την ετερογενή φύση της κάθε ερευνητικής υποδομής και των πόρων της.

Παραδείγματα υπηρεσιών:

- διαθεσιμότητα κόμβου (availability)
- επιδόσεις δικτύου (delay (καθυστέρηση), bandwidth (εύρος ζώνης) κ.λ.π)
- επιδόσεις υπολογιστικών πόρων (CPU, μνήμη, δίσκοι)

Για την διεξαγωγή των πειραμάτων τους, οι ερευνητές δεσμεύουν πόρους από τις ερευνητικές υποδομές. Μετά την ολοκλήρωση κάθε πειράματος ο ερευνητής καλείται να απαντήσει αρκετές ερωτήσεις σχετικά με την εμπειρία του. Οι ερωτήσεις αυτές περιλαμβάνουν συγκεκριμένα ερωτήματα για κάθε υπηρεσία που χρησιμοποιήθηκε. Έτσι για κάθε  $\lambda$  υπηρεσίες προκύπτουν  $\lambda$  απαντήσεις που από εδώ και στο εξής θα τις αποκαλούμε απόψεις (opinions). Οι βαθμολογίες δίνονται ως υποκειμενική ποιότητα (Mean Opinion Score - MOS) [27] λόγω και της ευρείας χρήσης του για την μέτρηση του QoE, το οποίο λαμβάνει τιμές από το 1 (χειρότερο) μέχρι το 5 (καλύτερο). Η διαδικασία αξιολόγησης είναι εμφανής στο παράδειγμα του σχήματος παρακάτω (Σχ 3.1). Ας πάρουμε έναν χρήστη ο οποίος χρησιμοποιεί για το πείραμα του 3 ερευνητικές υποδομές τις A, B, C και τρεις υπηρεσίες  $S_1, S_2, S_3$ . Στο τέλος κάθε πειράματος, ο χρήστης καλείται αξιολογήσει και τις τρεις υπηρεσίες, με τη μορφή των απόψεων (opinions)  $O_{s1}, O_{s2}, O_{s3}$ . Αυτές οι απόψεις αντιστοιχίζονται στις σχετικές υπηρεσίες. Κάθε άποψη επομένως συμπυκνώνει τις πληροφορίες για κάθε ερευνητική υποδομή που συμμετέχει στο πείραμα και υποστηρίζει την συγκεκριμένη υπηρεσία. Όπως αποδίδεται και σχηματικά, η άποψη  $O_{s1}$  περιλαμβάνει την άποψη του ερευνητή για την υπηρεσία  $S_1$  σε όλες τις ερευνητικές υποδομές (A,B,C). Η άποψη  $O_{s2}$  θα αντιστοιχηθεί στην υπηρεσία  $S_2$  του A, ενώ η άποψη  $O_{s3}$  στην  $S_3$  στις ερευνητικές υποδομές A, B.



Σχήμα 3. 1 – Σχήμα αξιολόγησης ερευνητικών υποδομών/υπηρεσιών

### 3.3 Ο αλγόριθμος του FTUE

Ο αλγόριθμος τους FTUE βασίζεται στις αρχές του αλγόριθμου ROCQ όπως διατυπώθηκε από τους Garg and Battiti. Ο αλγόριθμος αυτός είχε αρχικά σχεδιαστεί για ομότιμα συστήματα, στο FTUE έχουν γίνει πολλές τροποποιήσεις προκειμένου να μπορεί να προσαρμοστεί σε ένα ομόσπονδο περιβάλλον από ερευνητικές υποδομές. Το μοντέλο του αποτελείται από ένα σετ από  $K$  ερευνητικές υποδομές τα οποία παρέχουν  $S$  υπηρεσίες και τέλος  $N$  χρήστες, όπου κάθε χρήστης  $i \in N$  πραγματοποιεί  $M_i$  πειράματα. Κάθε ερευνητική υποδομή υποστηρίζει μία ή και περισσότερες από τις υπηρεσίες  $S$ . Επίσης κάθε χρήστης  $i \in N$ , κατά την ολοκλήρωση του πειράματος  $j \in M_i$  παρέχει την άποψη του (opinion), όπως αναφέρεται και παραπάνω, για την συνολική εμπειρία του σε σχέση με τη χρήση της υπηρεσίας  $s \in S$ . Η άποψη του για τη συγκεκριμένη υπηρεσία θα συμβολίζεται ως  $O_{ij}^S$ . Έπειτα το πλαίσιο αναθέτει κάθε υπηρεσία στις αντίστοιχες ερευνητικές υποδομές που την παρέχουν και συμμετείχαν στο πείραμα ( $O_{ij}^{kS}$ ).

Έπειτα από την συλλογή και ανάθεση των απόψεων, κάθε άποψη συγκρίνεται με τα αντίστοιχα δεδομένα παρακολούθησης (το  $D_j^{kS}$  συμβολίζει τα δεδομένα παρακολούθησης για την υπηρεσία  $s \in S$  της ερευνητικής υποδομής  $k \in K$  για το  $j$  πείραμα). Δεδομένου του αποτελέσματος της σύγκρισης, μεταβάλλεται κατάλληλα η τιμή αξιοπιστίας (Credibility) του χρήστη αναλόγως με τον αν η άποψη του συγκλίνει με τα δεδομένα παρακολούθησης η όχι.

Οι απόψεις (Opinions) σταθμίζονται με τις τιμές αξιοπιστίας (Credibility) και ποιότητας (Quality) και συνυπολογίζοντας προκύπτει η τιμή εμπιστοσύνης (Trust)  $T_{ks}$  για κάθε υπηρεσία  $s$  της ερευνητικής υποδομής  $k$ .

Για λόγους απλότητας, για την ολοκληρωμένη και αναλυτική παρουσίαση των μεταβλητών και του συστήματος, θα χρησιμοποιηθεί μια υπηρεσία την οποία θα συμβολίζουμε ως  $S$ , ενώ με  $t$  θα συμβολίζουμε τον αριθμό επανάληψης.

### Άποψη

Η άποψη (Opinion) Εκφράζει τον βαθμό ικανοποίησης για μια υπηρεσία  $S$  του χρήστη  $i$  για το  $j$  πείραμα και συμβολίζεται ως εξής:

$$O_{ij}^{kS} \{1 \leq i \leq N, 1 \leq j \leq M_i, k \in K\} \quad (3.1)$$

Όπου  $N$  είναι ο αριθμός χρηστών του συστήματος και  $M_i$  ο αριθμός των πειραμάτων που έχει εκτελέσει ο χρήστης  $i$ . Η άποψη  $O_{ij}^{kS}$  παίρνει τιμές που ανήκουν στο διάστημα  $[0,1]$  (κανονικοποιημένη μορφή του MOS).

### Ποιότητα

Η ποιότητα (Quality) αντιπροσωπεύει την εμπιστοσύνη του χρήστη που παρέχει την αξιολόγηση. Αν ένας χρήστης δεν είναι σίγουρος για την άποψη του, η αξιοπιστία του (Credibility) και η τελική τιμή της φήμης του (reputation) θα τροποποιηθεί ανάλογα. Η ποιότητα του χρήστη  $i$  για το  $j$  πείραμα αναπαρίσταται ως  $Q_{ij}$  και παίρνει τιμές που ανήκουν στο διάστημα  $[0,1]$  (κανονικοποιημένη μορφή του MOS).

### Αξιοπιστία

Η αξιοπιστία (Credibility) χρησιμοποιείται ώστε να αποτυπωθεί ως μεταβλητή, το αν ένας χρήστης παρέχει αληθής η ψευδείς κρίσεις/αξιολογήσεις για τις υπηρεσίες των ερευνητικών υποδομών. Κάθε χρήστης διαθέτει μία ξεχωριστή τιμή αξιοπιστίας, η οποία και προσαρμόζεται ανάλογα με την ειλικρίνεια των αξιολογήσεων του. Αν η τιμή της αξιοπιστίας ενός χρήστη είναι χαμηλή, και επομένως ο χρήστης δεν είναι έμπιστος, η άποψη του παίζει μικρό ρόλο στην αξιολόγηση του συστήματος και αντίστροφα, αν έχει υψηλό βαθμό αξιοπιστίας τότε η άποψη του έχει και μεγαλύτερο βάρος. Η αξιοπιστία (credibility) του κάθε χρήστη παίρνει τιμές στο διάστημα  $[0,1]$  και ενημερώνεται μετά το πέρας κάθε αξιολόγησης με βάση κάποια διαφορετικά σενάρια τα οποία θα αναλυθούν παρακάτω.

Δεδομένου του γεγονότος ότι για την εκτέλεση ενός μόνου πειράματος μπορεί να χρησιμοποιηθούν περισσότερες από μία ερευνητικές υποδομές, το πλαίσιο καλείται να προσαρμοστεί σε διαφορετικές καταστάσεις και συμπεριφορές χρηστών που μπορεί να προκύψουν. Για αυτόν τον λόγο έχουν αναγνωριστεί και τυποποιηθεί τα πιθανά σενάρια και

για κάθε ένα από αυτά έχει καθορισθεί και ο κατάλληλος μαθηματικός τύπος για την ενημέρωση της αξιοπιστίας του χρήστη/ερευνητή. Για κάθε διαφορετική περίπτωση η άποψη  $O_{ij}^{kS}$  που παρέχει ο χρήστης  $i$  συγκρίνεται με τα δεδομένα παρακολούθησης  $D_j^{kS}, k \in K$  με βάση ένα κατώφλι  $e$  το οποίο ορίζεται καταλλήλως ανάλογα και με τα χαρακτηριστικά της υπηρεσίας που αξιολογείται.

### Σενάριο I

Με δεδομένο ένα πείραμα το οποίο εκτελείται από τον χρήστη  $i$ , το οποίο εκτείνεται σε μια σειρά από ερευνητικές υποδομές και την υπηρεσία  $S$ , η οποία συνδέεται, με μία ή και περισσότερες από αυτές τις ερευνητικές υποδομές.

Αν η απόλυτη τιμή τις διαφορές τους τη δοσμένης άποψης  $O_{ij}^{kS}$  με τα (κανονικοποιημένα) δεδομένα παρακολούθησης  $D_j^{kS}$  δεν είναι μεγαλύτερη του κατωφλίου  $e$ , για κάθε ερευνητική υποδομή  $k \in K$  το οποίο και εμπλέκεται με το πείραμα, τότε η αξιοπιστία του χρήστη αυξάνεται με τον ακόλουθο τρόπο:

$$C_i^{t+1} = C_i^t + \frac{(1 - C_i^t)Q_{ij}}{2}, \quad \text{if } |D_j^{kS} - O_{ij}^{kS}| \leq e, \quad \forall k \in K \quad (3.2)$$

Το σενάριο αυτό αντιστοιχεί στην περίπτωση όπου οι αξιολογήσεις των ερευνητών ταιριάζουν με τα δεδομένα παρακολούθησης για κάθε ερευνητική υποδομή.

### Σενάριο II

Στο σενάριο αυτό, το απόλυτο της διαφοράς της δοσμένης άποψης  $O_{ij}^{kS}$  με τα δεδομένα παρακολούθησης  $D_j^{kS}$  είναι μεγαλύτερο από το κατώφλι  $e$  για κάθε ερευνητική υποδομή  $k \in K$  στο εμπλεκόμενο στο πείραμα. Επιπρόσθετα, η τιμή των δεδομένων παρακολούθησης για όλες τις εμπλεκόμενες ερευνητικές υποδομές είναι είτε όλες υψηλότερες από τις τιμές των αντίστοιχων απόψεων ( $D_j^{kS} - O_{ij}^{kS} > 0$ ), είτε όλες χαμηλότερες ( $D_j^{kS} - O_{ij}^{kS} < 0$ ). Τότε η αξιοπιστία του χρήστη μειώνεται με τον ακόλουθο τρόπο.

$$C_i^{t+1} = C_i^t - \frac{C_i^t Q_{ij}}{2}, \quad \text{if } |D_j^{kS} - O_{ij}^{kS}| > e, \quad \forall k \in K$$

$$\text{if } (D_j^{kS} - O_{ij}^{kS} > 0) \vee (D_j^{kS} - O_{ij}^{kS} < 0), \forall k \in K \quad (3.3)$$

Το σενάριο αυτό ανταποκρίνεται στην περίπτωση όπου οι αξιολογήσεις του ερευνητή διαφέρουν από τα δεδομένα παρακολούθησης για κάθε ερευνητική υποδομή

### Σενάριο III

Σε αυτό το σενάριο, η αξιολόγηση του ερευνητή βρίσκεται στο μέσον της κλίμακας (3 στα 5) και η απόλυτη διαφορά της δοσμένης άποψης  $O_{ij}^{kS}$  με τα δεδομένα παρακολούθησης  $D_i^{kS}$  είναι μεγαλύτερη από το κατώφλι  $e$  για κάθε ερευνητική υποδομή  $k \in K$  εμπλεκόμενο στο πείραμα ( $|D_j^{kS} - O_{ij}^{kS}| > e$ ). Ακόμη η τιμή των δεδομένων παρακολούθησης μίας τουλάχιστον ερευνητικής υποδομής  $k \in K$  είναι μεγαλύτερη της τιμής της άποψης του ερευνητή ( $D_j^{nS} - O_{ij}^{kS} > 0, n \in K$ ), και η τιμή των δεδομένων παρακολούθησης μίας τουλάχιστον ερευνητικής υποδομής είναι χαμηλότερη από την αντίστοιχη άποψη του ερευνητή ( $D_j^{mS} - O_{ij}^{kS} < 0, m \in K$ ).

Σε αυτήν την περίπτωση η αξιοπιστία υπολογίζεται με τον ακόλουθο τρόπο:

$$C_i^{t+1} = C_i^t + \frac{(1 - C_i^t)Q_{ij}}{2},$$
$$\text{if } |D_j^{kS} - O_{ij}^{kS}| > e, \forall k \in K \wedge O_{ij}^{kS} = 3 \wedge$$
$$\exists n \in K: (D_j^{nS} - O_{ij}^{kS}) > 0 \wedge \exists m \in K: (D_j^{mS} - O_{ij}^{kS}) < 0, m \in K \quad (3.4)$$

Το παρόν σενάριο ανταποκρίνεται στην περίπτωση όπου οι ερευνητές αξιολογούν με συντηρητικό τρόπο δίνοντας ως βαθμό την μέση τιμή την ώρα που οι παρεχόμενες υπηρεσίες είναι είτε χειρότερες είτε καλύτερες.

### Σενάριο IV

Στην συγκεκριμένη περίπτωση, η απόλυτη διαφορά της δοσμένης άποψης  $O_{ij}^{nS}$  με τα δεδομένα παρακολούθησης  $D_i^{nS}$  είναι μεγαλύτερη του κατωφλίου  $e$  για τουλάχιστον μια ερευνητική υποδομή  $n \in K$  που συμμετέχει στο πείραμα. Παράλληλα, υπάρχει τουλάχιστον μια ερευνητική υποδομή  $m \in K$ , όπου η απόλυτη διαφορά της δοσμένης άποψης  $O_{ij}^{mS}$  με τα δεδομένα παρακολούθησης  $D_i^{mS}$  δεν είναι μεγαλύτερη του κατωφλίου  $e$ . Σε αυτήν την περίπτωση λοιπόν, η αξιοπιστία του χρήστη αυξάνεται, ενώ για τις ερευνητικές υποδομές  $L \subset K$  όπου τα δεδομένα παρακολούθησης αποκλίνουν από την άποψη του χρήστη, παίρνουμε μια νέα τροποποιημένη άποψη. Η αξιοπιστία και η νέα άποψη υπολογίζονται με τον ακόλουθο τρόπο.

$$\begin{cases} C_i^{t+1} = C_i^t + \frac{(1 - C_i^t)Q_{ij}}{2} \\ O_{ij}^{lS} = O_{ij}^{lS} + (D_j^{lS} - O_{ij}^{lS})M_w, \forall l \in L \\ \exists n \in K: |D_j^{nS} - O_{ij}^{nS}| > e \wedge \exists m \in K: |D_j^{mS} - O_{ij}^{mS}| < e \end{cases} \quad (3.5)$$



Η νέα τροποποιημένη άποψη βρίσκεται ανάμεσα στην τιμή της αρχικής άποψης και της τιμής των δεδομένων παρακολούθησης με τιμή που εξαρτάται από την μεταβλητή  $M_u$  η οποία και υποδεικνύει τον βαθμό απομόνωσης των κακόβουλων χρηστών και παίρνει τιμές από (0,1]. Αυτή η νέα τροποποιημένη άποψη χρησιμοποιείται για να υπολογιστεί η φήμη της δεδομένης υπηρεσίας

*Το σενάριο αυτό ανταποκρίνεται στην περίπτωση όπου τουλάχιστον μια ερευνητική υποδομή έχει διαφορετική συμπεριφορά από τις υπόλοιπες εμπλεκόμενες με το πείραμα.*

### **Φήμη**

Η φήμη  $R^{kS}$  της υπηρεσίας  $S$  της ερευνητικής υποδομής  $k \in K$  προκύπτει από το αποτέλεσμα της άθροισης όλων των απόψεων με βάση το αντίστοιχο βάρος της κάθε μίας όπως αυτό προκύπτει από τους δείκτες αξιοπιστίας και ποιότητας.

$$R^{kS} = \frac{\sum_{i=0}^N \sum_{j=0}^{M_i} O_{ij}^{kS} C_{ij} Q_{ij}}{\sum_{i=0}^N \sum_{j=0}^{M_i} C_{ij} Q_{ij}}, \forall k \in K \quad (3.6)$$

*Η διαδικασία υπολογισμού πραγματοποιείται με την εκτέλεση δυο διαδοχικών σειρών όπου η εξωτερική σειρά αφορά όλους τους χρήστες, και η εσωτερική όλα τα πειράματα  $M_i$  του χρήστη  $i$  που χρησιμοποιούν πόρους της ερευνητικής υποδομής  $k$ .*

## **4 Σχεδίαση/Υλοποίηση**

Σε αυτό το κεφάλαιο θα επεξηγηθούν αναλυτικά οι σχεδιαστικές λεπτομέρειες αλλά και οι λεπτομέρειες της αρχιτεκτονικής της εφαρμογής μας και των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της,

### **4.1 Αρχιτεκτονική Μοντέλου-Προβολής-Ελεγκτή**

Η εφαρμογή μας υλοποιήθηκε χρησιμοποιώντας το πλαίσιο εφαρμογών διαδικτύου (web application framework) Ruby on Rails [16]. Το πλαίσιο αυτό ακολουθεί το πρότυπο σχεδιασμού Μοντέλου-Προβολής-Ελεγκτή (Model View Controller - MVC) [17]. Βασική αρχή του προτύπου είναι ο διαχωρισμός και η διακριτότητα των επιμέρους λειτουργιών του συστήματος. Ο διαχωρισμός τόσο στο πλαίσιο ανάπτυξης εφαρμογών διαδικτύου Ruby on Rails τόσο και σε όλα σχεδόν τα αντίστοιχα πλαίσια γίνεται με την εξής λογική.

#### **Μοντέλο**

Το μοντέλο αναπαριστά τις πληροφορίες και τα δεδομένα της βάσης δεδομένων. Το μοντέλο είναι ανάλογα και με τον σχεδιασμό του κάθε επιμέρους MVC πλαισίου διαδικτυακών εφαρμογών, ανεξάρτητο από την βάση δεδομένων αυτή καθ' αυτή. Το μοντέλο επίσης είναι υπεύθυνο για τον έλεγχο εγκυρότητας των προς υποβολή στην βάση δεδομένων. Η συνήθης πρακτική για το μοντέλο στα MVC είναι η αντιστοιχία κάθε πίνακα στη βάση δεδομένων με ένα μοντέλο.

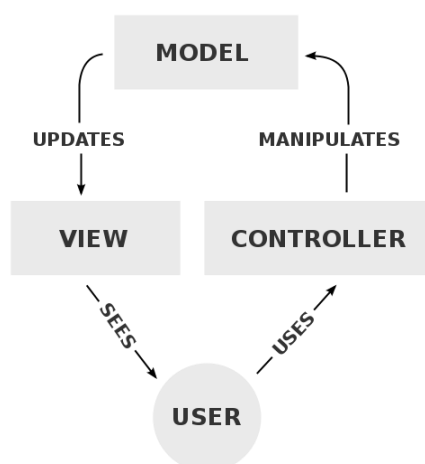
#### **Προβολή**

Η προβολή αποτελεί το στρώμα παρουσίασης της εφαρμογής. Το στρώμα της προβολής ευθύνεται για την απόδοση των μοντέλων από αντικείμενα της χρησιμοποιούμενης γλώσσας προγραμματισμού σε διάφορα πρότυπα για την παρουσίαση τους στο χρήστη είτε σε κάποια εφαρμογή η διεπαφή π.χ. HTML, XML JSON κ.ο.κ.

## Ελεγκτής

Ο Ελεγκτής συνδέει το μοντέλο με την προβολή. Ο ελεγκτής γνωρίζει πως να επεξεργαστεί τα δεδομένα που προέρχονται από το μοντέλο και πως να τα περάσει αυτά στην προβολή.

## Η αρχιτεκτονική του μοντέλου – προβολή – ελεγκτή (Model-View-Controller)



Σχήμα 4.1 – παράδειγμα αλληλεπίδρασης των τμημάτων ενός MVC

Παρ' ότι τα πλαίσια με βάση την αρχιτεκτονική MVC έχουν ως στόχο την διάκριση των λειτουργιών και την μικρότερη τους εξάρτηση για να επιτύχουν αφενός καθαρό και κατανοητό κώδικα, αφετέρου την δυνατότητα επεξεργασίας του κάθε τμήματος ξεχωριστά χωρίς να υπάρχει κάποια σύγκρουση μεταξύ των διαφόρων τμημάτων τα τμήματα πρέπει να επικοινωνούν με συγκεκριμένο τρόπο προκειμένου να λειτουργεί ορθά και γρήγορα η όποια εφαρμογή.

Όπως βλέπουμε και στο παραπάνω σχήμα (Σχ. 4.1) ξεκινώντας για παράδειγμα από ένα αίτημα χρήστη προς την εφαρμογή μας, αφού αυτή δρομολογηθεί στον κατάλληλο ελεγκτή, ο ελεγκτής επικοινωνεί με το μοντέλο και γίνονται οι απαραίτητες ανανεώσεις/προσθήκες/αναζητήσεις ή όποια άλλη εργασία απαιτείται από την βάση, το αποτέλεσμα που σχετίζεται με αυτές τις εργασίες μεταβιβάζεται στην προβολή η οποία αποδίδει τα δεδομένα αυτά στο κατάλληλο πρότυπο για την παρουσίαση τους στο χρήστη.

Για τον σχεδιασμό και την υλοποίηση της εφαρμογής μας κάναμε μερικές τροποποιήσεις στις αρχικές προεπιλογές του Ruby on Rails. Αρχικά, καθώς η εφαρμογή μας σχεδιάστηκε ως αποκλειστικά-API εφαρμογή (δεν παράγονται HTML σελίδες) (API-only application) πραγματοποιήθηκαν ορισμένες βασικές τροποποιήσεις.

- Με την βοήθεια του gem rails-api[18] βελτιστοποιήσαμε το σύστημα μας για την χρήση του ως API αποκλειστικά, καθώς έτσι φορτώνονται μόνο τα απαραίτητα κομμάτια για την εφαρμογή μας ώστε να είναι ελαφρύτερη και γρηγορότερη.

- Ως προεπιλογή το τμήμα της προβολής διαχειρίζεται από αρχεία ERB (Ruby ERB Script Files) τα οποία αποτελούν αρχεία HTML τα οποία δίνουν την δυνατότητα ενσωμάτωσης κώδικα Ruby. Η εφαρμογή μας απαντάει στα αιτήματα των χρηστών σε καθαρή μορφή JSON, για την διαχείριση του τμήματος της προβολής χρησιμοποιήθηκε το gem jbuilder [19] το οποίο επιτρέπει την μορφοποίηση των JSON απαντήσεων.
- Το πλαίσιο ανάπτυξης Ruby on Rails υποστηρίζει Object Relational Mapping [20] (ORM) παρέχοντας την δυνατότητα αναπαράστασης και τροποποίησης δεδομένων της βάσης δεδομένων με την μορφή αντικειμένων σε αντικειμενοστραφείς γλώσσες όπως η Ruby στην περίπτωση μας. Αυτό γίνεται συνήθως με την αναπαράσταση ενός πίνακα της βάσης ως αντικείμενου. Για παράδειγμα αν έχουμε ένα πίνακα στην βάση που κρατάει στοιχεία των τηλεφωνικών επαφών χρηστών, η εγγραφή των επαφών ενός συγκεκριμένου χρήστη θα μπορεί να γίνει αντικείμενο με το όνομα χρήστη για παράδειγμα το οποίο περιλαμβάνει τα εξής χαρακτηριστικά/πεδία, το όνομα χρήστη, λίστα τηλεφώνων και λίστα διευθύνσεων. Το Ruby on Rails από προεπιλογή χρησιμοποιεί το ενσωματωμένο σε αυτό Active Record ως ORM [21]. Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε το sequel ORM [14] καθώς παρέχει καθώς είναι πιο σταθερό και αξιόπιστο, ταχύτερο και πιο προηγμένο σε πολλές του λειτουργίες από το Active Record [22] [23].
- Τέλος, το Ruby on Rails βασίζεται σε συγκεκριμένες ορθές πρακτικές για την εύκολη, αποτελεσματική και αποδοτική λειτουργία του. Την “σύμβαση υπεράνω της διαμόρφωσης ρυθμίσεων” (convention over configuration) , δηλαδή ως βασική αρχή την τήρηση κάποιων βασικών συμβάσεων στην δημιουργία και ονοματολογία των ελεγκτών και των μοντέλων π.χ. οι οποίες μας επιτρέπουν την γρήγορη και απροβλημάτιστη λειτουργικότητα χωρίς ιδιαίτερες ρυθμίσεις. Την αρχή “μην επαναλαμβάνεις τον εαυτό σου” (Don’t repeat yourself – DRY) και την αρχή του “λεπτού ελεγκτή και παχύ μοντέλου” (fat model and skinny controller). Προκειμένου να είναι ο κώδικας μας DRY (να μην ξαναγράψουμε κώδικα για αντίστοιχη λειτουργία που επαναλαμβάνεται αλλά να επαναχρησιμοποιείται το αρχικό μας κομμάτι) και του καθαρού ελεγκτή υιοθετήσαμε μια εναλλακτική σχεδιαστική λογική αυτή των υπηρεσιών [24] [25]. Με αυτόν τον τρόπο καταφέραμε να δημιουργήσουμε “λεπτούς” ελεγκτές, οι οποίοι πραγματοποιούν σύνθετες εργασίες εκατοντάδων γραμμών κώδικα. Αυτό καθίσταται δυνατό διότι στον ελεγκτή συμπεριλάβαμε μόνο τις απολύτως απαραίτητες εντολές και τις κλήσεις των υπηρεσιών για την εκτέλεση των πιο σύνθετων εργασιών, στις οποίες και βρίσκεται το μεγάλο τμήμα του κώδικα. Έτσι όπως φαίνεται και από το απόσπασμα κώδικα στο Σχήμα 4.2 κρατήσαμε τον ελεγκτή μας μικρό και αποφύγαμε την επανάληψη τμημάτων κώδικα

κρατώντας τον ελεγκτή μας DRY. Οι υπηρεσίες που καλούνται στο Σχήμα 4.2 είναι οι `ConvertTime`, `GenerateEid`, `SubmitRating`, `CalculateReputation`.

```
def userqoe
  @start_ts, @end_ts = ConvertTime.call(rate_params)
  @eid = GenerateEid.call(@resources) if !@resources.nil?
  @user, @exp_id, @testbeds_array, @services_array = SubmitRating.call(@user_urn, @slice_urn, @start_ts, @end_ts, @resources, @eid, @user_eval)
  if @user == nil
    render :json => "Experiment exists", status: 400
    return
  end
  @test = CalculateReputation.call(@exp_id, @slice_urn, @user_urn, @start_ts, @end_ts, @user, @testbeds_array, @services_array)
  render :json => "The operation completed successfully"
end
```

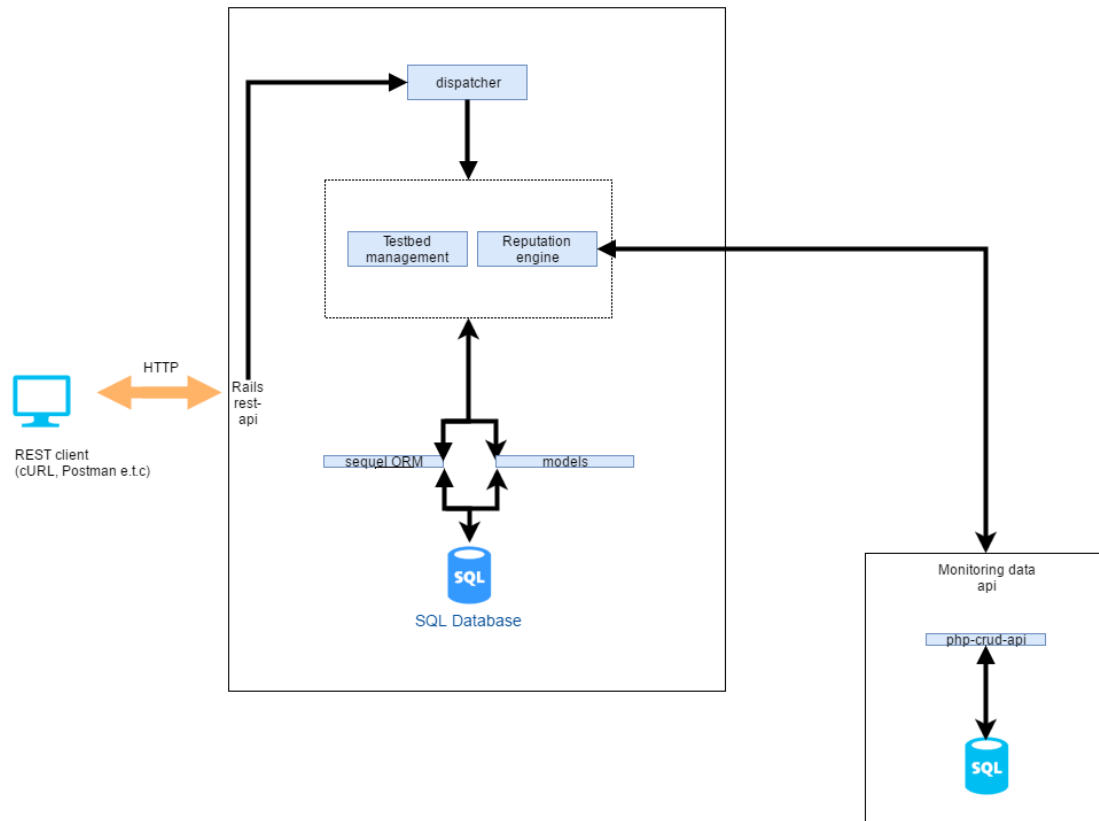
Σχήμα 4.2 - Απόσπασμα κώδικα με χρήση υπηρεσιών

## 4.2 Διεπαφή Δεδομένων Παρακολούθησης

Για την προσομοίωση της προγραμματιστικής διεπαφής των δεδομένων παρακολούθησης χρησιμοποιήθηκε ένα πολύ απλό CRUD (create, read, update, delete) API υλοποιημένο σε PHP [13] καθώς το μόνο που χρειαζόμαστε από αυτό είναι η προσθήκη νέων τιμών στη βάση δεδομένων παρακολούθησης, και η λήψη των τιμών αυτών από την εφαρμογή μας. Για την προσθήκη τιμών δεδομένων παρακολούθησης στο API δημιουργήσαμε script σε Ruby το οποίο προσθέτει περιοδικά δεδομένα παρακολούθησης (π.χ. κάθε 5 λεπτά).

## 4.3 Αρχιτεκτονική Συστήματος

Σε αυτήν την ενότητα θα παρουσιάσουμε την αρχιτεκτονική του συνολικού συστήματος, όλα τα τμήματα του και τον τρόπο που επικοινωνούν αυτά μεταξύ τους καθώς και το ποιες λειτουργίες εξυπηρετεί η εφαρμογή μας και με ποιον τρόπο σχεδιαστικά υλοποιήθηκαν αυτές. Όπως φαίνεται και στο παρακάτω σχήμα η εφαρμογή μας απαντάει σε HTTP αιτήματα, τα οποία ανάλογα το τελικό τους σημείο (API endpoint) και το είδος της κλήσης (POST, GET κ.λ.π) δρομολογούνται στον αντίστοιχο ελεγκτή και στην αντίστοιχη ενέργεια (action) του ελεγκτή αυτού. Για την υλοποίηση της εφαρμογής μας χρησιμοποιήσαμε δύο ελεγκτές. Έναν για την διαχείριση των αιτημάτων αναφορικά με τις ερευνητικές υποδομές (Testbed management) και έναν για τον μηχανισμό υπολογισμού της φήμης και την διαχείριση αυτών (Reputation engine). Επίσης παρουσιάζεται στο σχήμα και το δεύτερο API το οποίο είναι υπεύθυνο για την παροχή των δεδομένων παρακολούθησης (προσομοίωση) στην βασική μας εφαρμογή για τον υπολογισμό της τιμής φήμης σύμφωνα με το FTUE όπως αναλύθηκε στο κεφάλαιο 3.



Σχήμα 4. 3 – Σχήμα αρχιτεκτονικής συστήματος

Με τους παρακάτω πίνακες παρουσιάζονται αναλυτικά οι δύο ελεγκτές και οι ενέργειες τους.

*Testbed\_controller.rb (Testbed Management)*

A/A	Όνομα ενέργειας (action)	Περιγραφή	Χρήστης
1	create	Προσθήκη νέας ερευνητικής υποδομής και των υπηρεσιών της	Διαχειριστής
2	update	Ανανέωση των παραμέτρων ερευνητικής υποδομής και των υπηρεσιών της	Διαχειριστής
3	destroy	Διαγραφή ερευνητικής υποδομής και των υπηρεσιών της	Διαχειριστής

*Πίνακας 4.1 – Ενέργειες testbed\_controller.rb*

Ο συγκεκριμένος ελεγκτής αναφέρεται στους διαχειριστές ενός συστήματος φήμης για ομόσπονδες ερευνητικές υποδομές όπως φαίνεται και από τον πίνακα. Για τις ενέργειες του συγκεκριμένου ελεγκτή δεν χρησιμοποιήθηκαν υπηρεσίες και δεν απαιτείται καμία αλληλεπίδραση με τη διεπαφή παρακολούθησης δεδομένων.

A/A	Όνομα ενέργειας (action)	Περιγραφή	Χρήστης
1	show	Λήψη της τιμής φήμης όλων των ερευνητικών υποδομών	Ερευνητής
2	Show_unrated	Έλεγχος αν συγκεκριμένο/α πείραμα/ατα είναι βαθμολογημένα	Ερευνητής
3	userqoe	Υποβολή αξιολόγησης – υπολογισμός φήμης/αξιπιστίας.	Ερευνητής
4	show	Λήψη της τιμής φήμης μίας συγκεκριμένης ερευνητικής υποδομής	Ερευνητής

Πίνακας 4.2 – Ενέργειες reputation\_controller.rb

Ο συγκεκριμένος ελεγκτής αναφέρεται στους ερευνητές. Η ενέργεια userqoe υλοποιεί υπηρεσίες όπως φαίνεται και στο σχήμα 4.2. Όλες οι υπηρεσίες βρίσκονται στην διεύθυνση /app/services/ ενώ οι ελεγκτές στην διεύθυνση /app/controllers/. Για την ολοκλήρωση της ενέργειας userqoe η υπηρεσία calculate\_reputation ζητάει από την εξωτερική διεπαφή δεδομένων παρακολούθησης (προσομοίωση) τα δεδομένα παρακολούθησης για το πείραμα που αξιολόγησε ο χρήστης για τον υπολογισμό της φήμης. Τα στάδια εξυπηρέτησης όλων των παραπάνω κλήσεων (και των δύο ελεγκτών) παρουσιάζονται με την μορφή διαγραμμάτων ακολουθίας μηνυμάτων στο κεφάλαιο 6.

### **Πιστοποίηση**

Για την εφαρμογή μας χρειαζόμαστε κάποιο σύστημα πιστοποίησης προκειμένου να διασφαλιστούν οι εξής δύο βασικές ανάγκες της κεντρικής αρχής διαχείρισης φήμης:

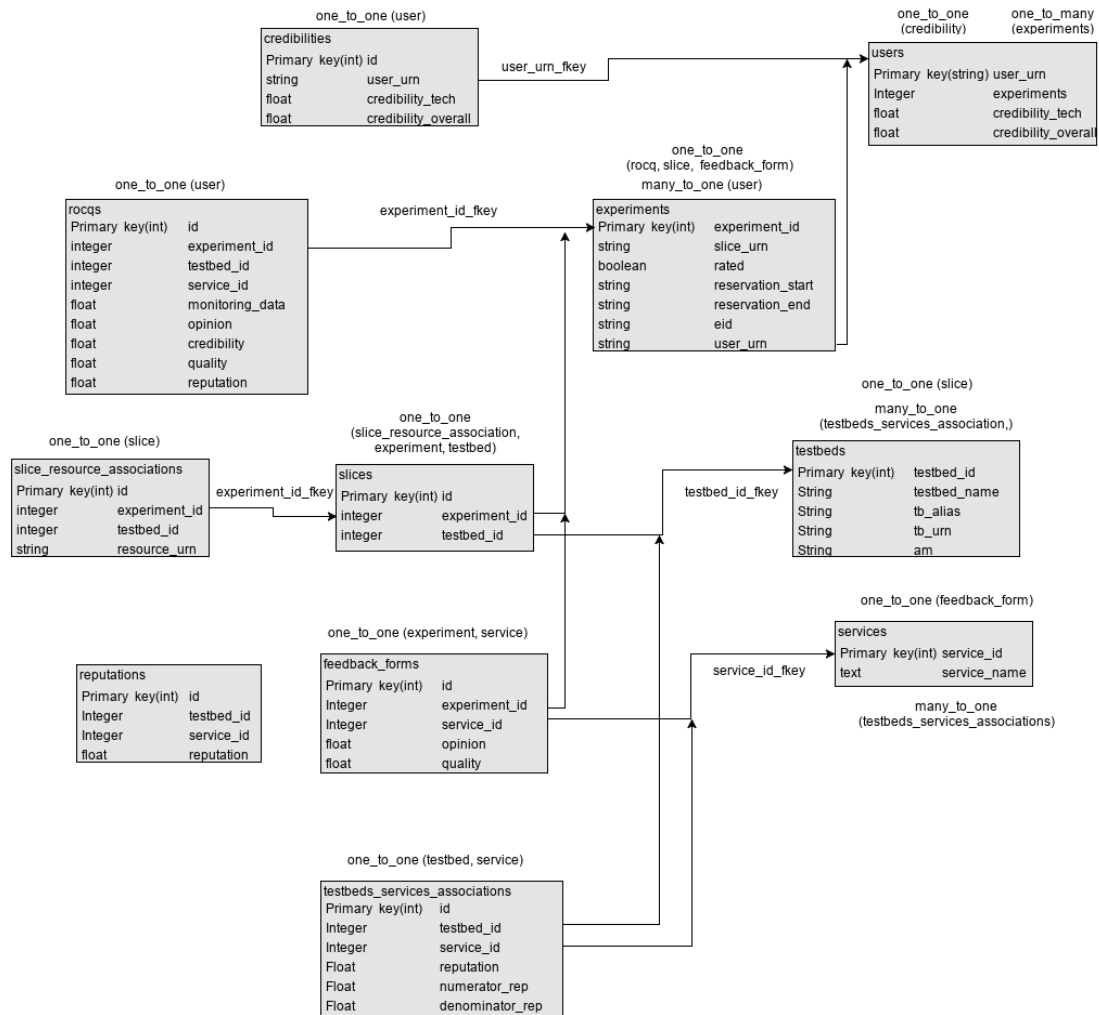
- Την διαχείριση ρόλων των χρηστών στην εφαρμογή μας.
- Την αποτροπή υποβολής αξιολογήσεων από κακόβουλους χρήστες οι οποίοι δεν έχουν πρόσβαση στην ομοσπονδία ερευνητικών υποδομών στην οποία αναφέρεται η εφαρμογή μας.

Για την εξασφάλιση αυτών των κριτηρίων υλοποιήθηκε σύστημα πιστοποίηση χρηστών (user authentication) με χρήση κλειδιού (token) [26]. Με αυτόν τον τρόπο αρχικά προστατεύουμε την εφαρμογή μας από κακόβουλες αξιολογήσεις από χρήστες οι οποίοι δεν έχουν πρόσβαση στην ομοσπονδία ερευνητικών υποδομών αφού σε μία πλήρη υλοποίηση κανένας χρήστης χωρίς κωδικούς και πρόσβαση στην ομοσπονδία δε θα μπορεί να έχει πρόσβαση στο κλειδί. Ταυτόχρονα με την διάκριση των κλειδιών σε χρηστών και διαχειριστών εξασφαλίζεται η αποτροπή χρήσης των διαχειριστικών ενεργειών από απλούς ερευνητές/χρήστες. Για την προστασία των κλειδιών των χρηστών και συνολικά των δεδομένων που αποστέλλονται, όλες οι συνδέσεις στην εφαρμογή μας γίνονται υποχρεωτικά με χρήση ασφαλούς σύνδεσης (HTTPS) και χρήση TLS με πιστοποιητικό Let's Encrypt [28]

## 4.4 Βάσεις Δεδομένων

Προκειμένου η εφαρμογή μας να μπορεί να υπολογίζει και να παρέχει τις τιμές φήμης και αξιοπιστίας θα πρέπει να διαθέτει προφανώς και τα κατάλληλα δεδομένα. Συγκεκριμένα η βάση μας περιλαμβάνει πληροφορίες σχετικά με:

1. Τις ερευνητικές υποδομές και τις υπηρεσίες τους οι οποίες βρίσκονται υπό αξιολόγηση, δηλαδή όλες τις διαθέσιμες υπηρεσίες και τις τιμές φήμης και εμπιστοσύνης τους.
2. Τους χρήστες, τις τιμές ποιότητας και αξιοπιστίας τους
3. Πληροφορίες για τα πειράματα που έχουν εκτελεσθεί.



Σχήμα 4. 4 – Σχήμα βάσης του reputation rest-api

Για την υλοποίηση της βάσης μας χρησιμοποιήθηκε η sqlite3 και στο επίπεδο της εφαρμογής χρησιμοποιήθηκε το sequel ως ORM για όλες τις εργασίες που σχετίζονται με την βάση.

Το σχήμα της βάσης αποτυπώνεται στο Σχήμα 4.4. Πέρα από τους πίνακες που περιέχει, τα κελιά και τους τύπους δεδομένων, καταγράφονται και οι συσχετίσεις (associations) που



χρησιμοποιήθηκαν στο sequel για τον προσδιορισμό της σχέσης των διαφορετικών πινάκων και το κλειδί (foreign key) με το οποίο σχετίζονται.

Για την βάση η οποία προσομοιώνει τον server με τα δεδομένα παρακολούθησης χρησιμοποιήθηκε και πάλι sqlite3. Η βάση αυτή αποτελείται ουσιαστικά από ένα πίνακα για κάθε υπηρεσία ο οποίος έχει και το όνομα της υπηρεσίας, την τεχνική παράμετρο που παρακολουθείται ώστε να εξαχθεί η τιμή των δεδομένων παρακολούθησης, και την χρονική στιγμή στην οποία έγινε ο έλεγχος. Έτσι για παράδειγμα (Σχ. 4.5) για την υπηρεσία της διαθεσιμότητας (availability), δηλαδή το ποσοστό χρόνου στο οποίο ο κόμβος ήταν ενεργός στο σχήμα είναι ως εξής:

Availability	
Primary key(int)	id
string	node
integer	up
last_check	integer

Σχήμα 4. 5 - Πίνακας των δεδομένων παρακολούθησης

Έτσι για κάθε στιγμή που ελέγχεται μια ερευνητική υποδομή το πεδίο up θα πάρει την τιμή 0 η 1 αναλόγως με την κατάσταση της ερευνητικής υποδομής και το last\_check πεδίο θα πάρει την χρονική στιγμή του ελέγχου σε unix time.

## **5 Τεχνικές λεπτομέρειες**

Σε αυτήν την ενότητα θα παρουσιάσουμε την λειτουργικότητα της Rest διεπαφής και του μηχανισμού υπολογισμού της φήμης και της αξιοπιστίας μέσω διαγραμμάτων ακολουθίας. Θα παρουσιάσουμε έτσι την ακολουθία ενεργειών και μηνυμάτων για κάθε μία από τις κλήσεις που υποστηρίζεις η εφαρμογή μας όπως αυτές περιγράφονται στο παράρτημα Α

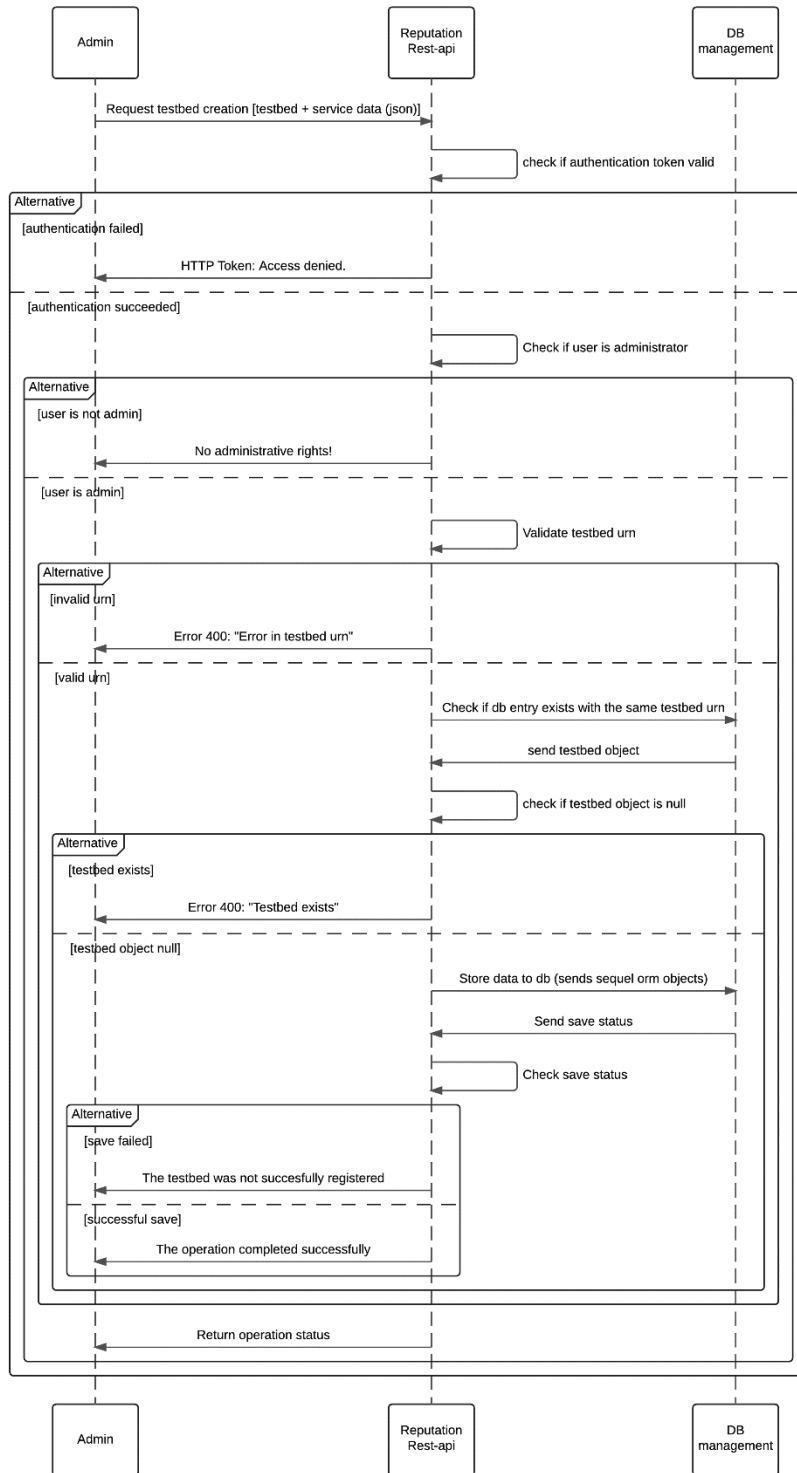
### **5.1 Διαγράμματα ακολουθίας μηνυμάτων**

Σε αυτό το τμήμα όπως είπαμε και προηγουμένως θα παρουσιάσουμε με διαγράμματα ακολουθίας (message sequence diagrams) όλες τις κλήσεις και λειτουργίες που υποστηρίζει η εφαρμογή μας με όλα τα πιθανά σενάρια εξέλιξης τους. Ταυτόχρονα γίνεται διάκριση στο ποιες κλήσεις αφορούν τους απλούς ερευνητές/χρήστες (Users/Experimenters) και ποιες τους διαχειριστές (admins) . Οι κλήσεις θα παρουσιαστούν με την ίδια σειρά που καταγράφονται και παρουσιάζονται στο Παράρτημα Α (Rest-api documentation).

#### **5.1.1 Προσθήκη ερευνητικής υποδομής και των υπηρεσιών της.**

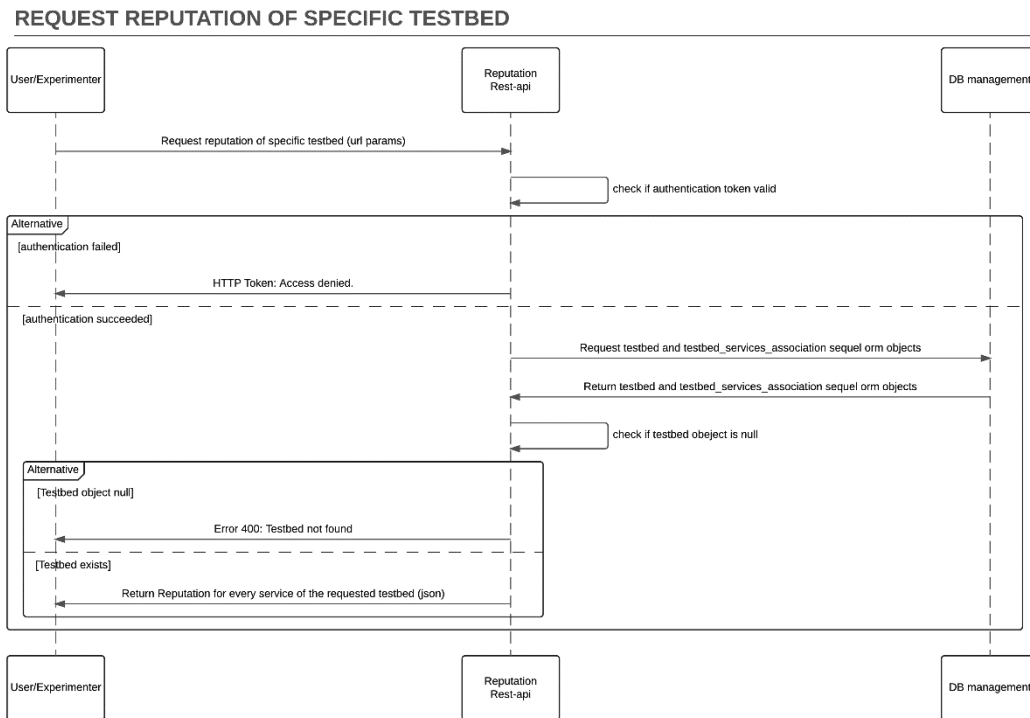
Στο παρακάτω σχήμα (5.1) παρουσιάζεται αναλυτικά η διαδικασία και τα βήματα για την προσθήκη μίας νέας ερευνητικής υποδομής και των υπηρεσιών της. Αρχικά ο διαχειριστής με χρήση κατάλληλου εργαλείου-προγράμματος πελάτη (client), αποστέλλει τα στοιχεία της ερευνητικής υποδομής και των υπηρεσιών της. Στο πρώτο στάδιο η εφαρμογή μας ελέγχει αν το κλειδί (token) που έστειλε ο χρήστης υπάρχει, αν δεν υπάρχει τότε η εφαρμογή επιστρέφει κατάλληλο μήνυμα στον χρήστη και η κλήση τελειώνει. Αν υπάρχει, ελέγχει στη συνέχεια αν το κλειδί ανήκει σε διαχειριστή αφού η ενέργεια αυτή αναφέρεται αποκλειστικά σε διαχειριστές. Αν δεν είναι, η κλήση τερματίζεται με αποστολή κατάλληλου μηνύματος στον χρήστη. Αντίθετα, συνεχίζεται κανονικά η κλήση. Έπειτα, η εφαρμογή μας ελέγχει την ορθότητα των urn που απέστειλε ο διαχειριστής και στην περίπτωση λάθους διακόπτει την διαδικασία και στέλνει στον διαχειριστή το κατάλληλο μήνυμα λάθους όπως φαίνεται και στην εικόνα. Έπειτα γίνεται έλεγχος ότι η συγκεκριμένη υποδομή δεν είναι ήδη καταχωρημένη στη βάση, και σε περίπτωση που είναι καινούργια συνεχίζει με την διαδικασία αποθήκευση της σε αυτήν. Αλλιώς επιστρέφει μήνυμα στον διαχειριστή ότι είναι καταχωρημένη ήδη. Τέλος, εφ' όσον και η εγγραφή στη βάση δεδομένων μας είναι επιτυχής, επιστρέφει μήνυμα ότι η διαδικασία ολοκληρώθηκε με επιτυχία.

## ADD TESTBED AND ITS SERVICES



Σχήμα 5.1 - Προσθήκη ερευνητικής υποδομής και των υπηρεσιών της

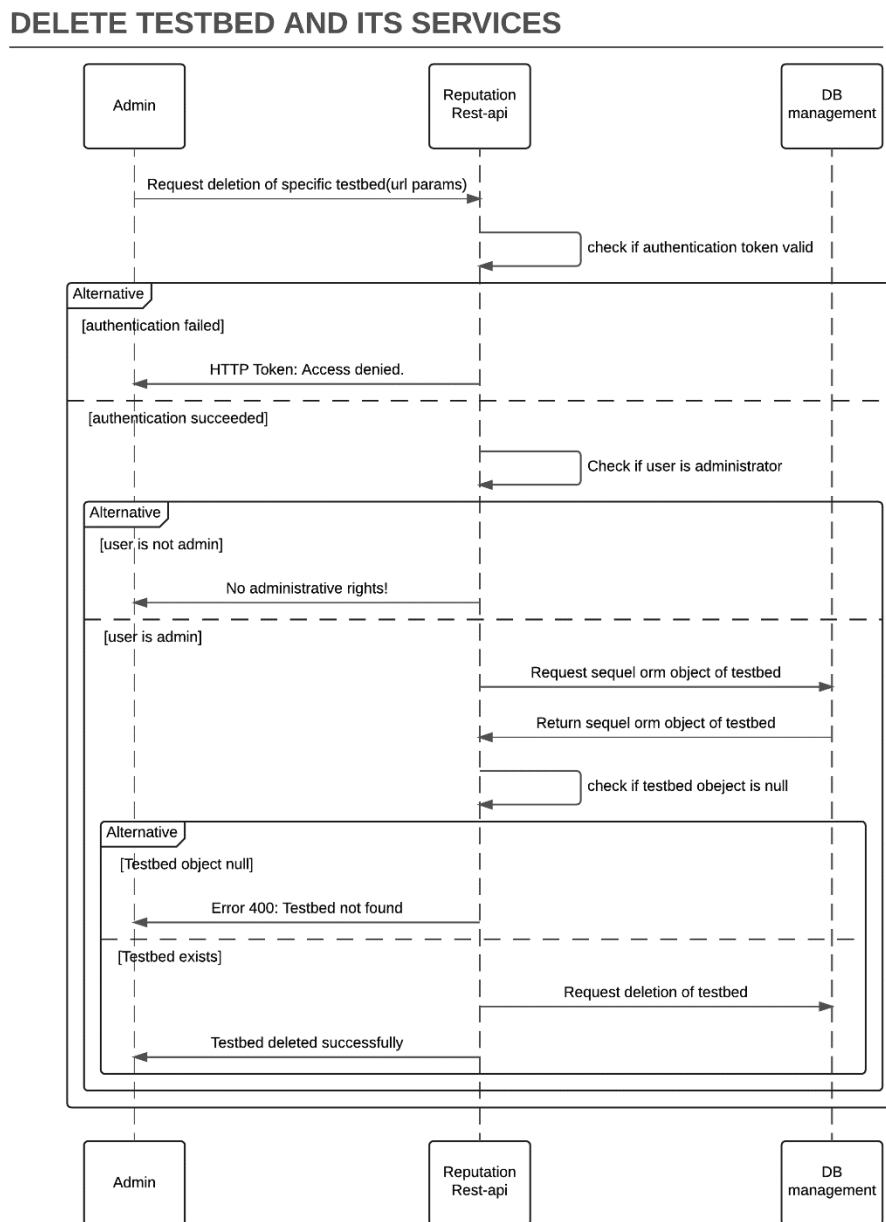
## 5.1.2 Λήψη φήμης συγκεκριμένης ερευνητικής υποδομής



Σχήμα 5.1 - Λήψη φήμης συγκεκριμένης ερευνητικής υποδομής

Στο συγκεκριμένο σενάριο χρήσης, ο χρήστης/ερευνητής επιθυμεί να μάθει την φήμη μιας συγκεκριμένης ερευνητικής υποδομής. Αρχικά γίνεται κλήση προς το rest-api κατά την οποία ο χρήστης στέλνει το url της ερευνητικής υποδομής που τον ενδιαφέρει ως παράμετρο στο url της συγκεκριμένης κλήσης. Επίσης αποστέλλεται ως επικεφαλίδα (header) το κλειδί (token) και η εφαρμογή μας πιστοποιεί τον χρήστη. Σε περίπτωση μη έγκυρου κλειδιού αποστέλλεται στο χρήστη σχετικό μήνυμα. Έπειτα, η εφαρμογή μας ζητάει από την βάση να επιστρέψει σε μορφή αντικειμένου από τον αντίστοιχο πίνακα (testbeds βλ σχήμα 4.4) την εγγραφή που αντιστοιχεί στην ερευνητική υποδομή που ζήτησε ο χρήστης. Ακολούθως, γίνεται έλεγχος από την εφαρμογή μας αν το αντικείμενο αυτό είναι κενό (η ζητούμενη από τον χρήστη εγγραφή δεν υπάρχει) ή όχι. Αν είναι κενό, επιστρέφει μήνυμα ότι η ζητούμενη ερευνητική υποδομή δεν υπάρχει. Αλλιώς επιστρέφει σε μορφή json την φήμη της ερευνητικής υποδομής για κάθε υπηρεσία που παρέχει και υποστηρίζει.

### 5.1.3 Διαγραφή ερευνητικής υποδομής και των υπηρεσιών της



Σχήμα 5.2 - Διαγραφή ερευνητικής υποδομής και των υπηρεσιών της

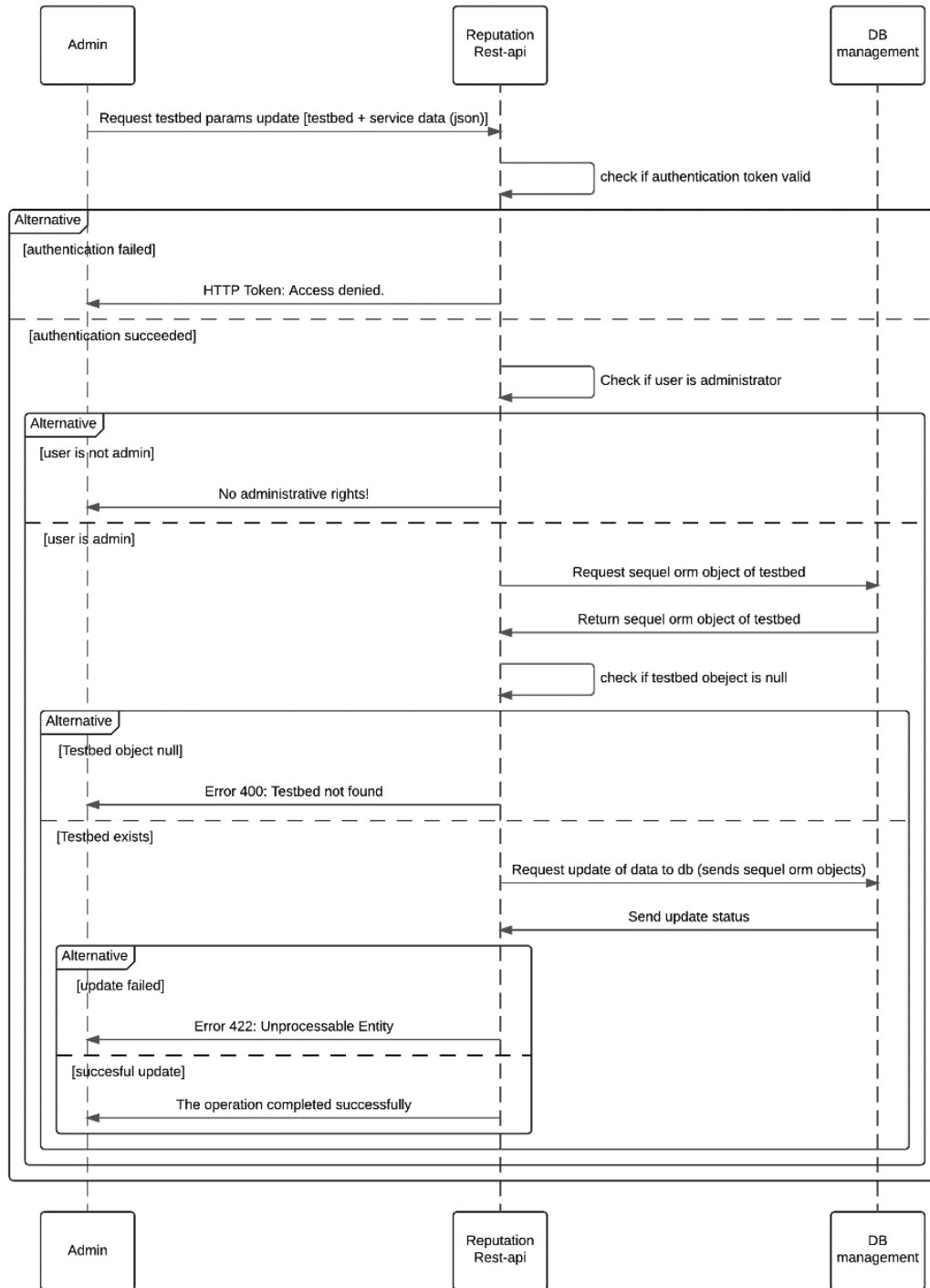
Για την διαγραφή μιας ερευνητικής υποδομής και των υπηρεσιών της ο διαχειριστής στέλνει το url της ερευνητικής υποδομής ως παράμετρο στην διεύθυνση που απαντάει η διεπαφή μας για την συγκεκριμένη λειτουργία (βλ Παράρτημα Α) καθώς και ένα κλειδί (token) ως επικεφαλίδα (header). Στην συνέχεια ελέγχεται η εγκυρότητα του κλειδιού και ο ρόλος του χρήστη (ερευνητής ή διαχειριστής) όπως περιγράφεται και στην ενότητα 5.1.1. Έπειτα ζητείται από τη βάση σε μορφή αντικειμένου η αντίστοιχη εγγραφή στην βάση στον πίνακα

testbeds (βλ σχήμα 4.4). Αν το αντικείμενο είναι κενό, η ερευνητική υποδομή που ζητήθηκε δεν υπάρχει και η εφαρμογή μας επιστρέφει στον διαχειριστή το αντίστοιχο μήνυμα. Σε περίπτωση που υπάρχει γίνεται αίτημα στην βάση για την διαγραφή της ερευνητικής υποδομής και των υπηρεσιών της. Με την ολοκλήρωση της διαγραφής, η εφαρμογή επιστρέφει μήνυμα ότι η διαδικασία ολοκληρώθηκε με επιτυχία.

### ***5.1.4 Ανανέωση των στοιχείων μιας ερευνητικής υποδομής ή/και των υπηρεσιών της***

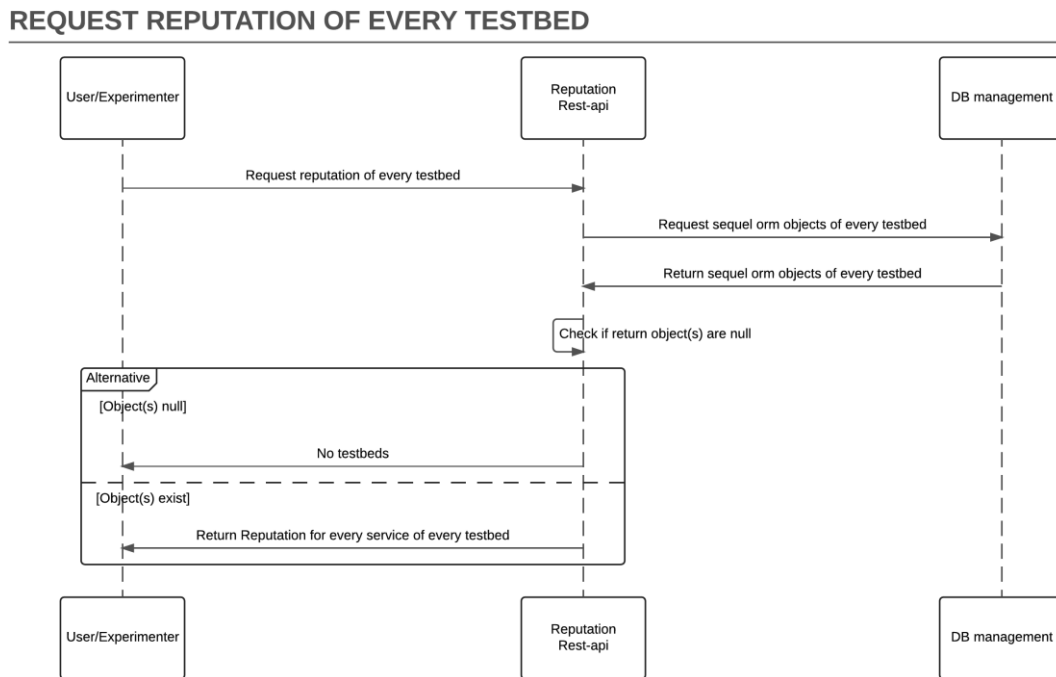
Η συγκεκριμένη λειτουργία της εφαρμογής μας απευθύνεται και πάλι σε κάποιον διαχειριστή. Ο διαχειριστής με χρήση του κατάλληλου εργαλείου/προγράμματος πελάτη (client) στέλνει σε μορφή json όπως υποδεικνύεται και στο Παράρτημα Α. Αρχικά όπως σε κάθε κλήση διαχειριστή ελέγχεται η εγκυρότητα του κλειδιού και ο ρόλος του χρήστη. Έπειτα με την ίδια διαδικασία που αναφέραμε και παραπάνω και φαίνεται και στο σχήμα 5.4, ελέγχεται αν η ερευνητική υποδομή της οποίας θέλει ο διαχειριστής να ανανεώσει τα στοιχεία της υπάρχει στο σύστημα και στην περίπτωση που δεν υπάρχει η εφαρμογή μας επιστρέφει στον διαχειριστή το κατάλληλο μήνυμα σφάλματος. Έπειτα ζητείται στην βάση η ανανέωση των παραμέτρων (αποστέλλονται με την μορφή αντικειμένου) και μετά επιστρέφεται στον διαχειριστή το κατάλληλο μήνυμα ανάλογα με την επιτυχημένη ή όχι έκβαση της ανανέωσης των παραμέτρων.

## UPDATE TESTBED AND SERVICE



Σχήμα 5.3 - Ανανέωση των στοιχείων μιας ερευνητικής υποδομής ή/και των υπηρεσιών της

### 5.1.5 Λήψη της φήμης κάθε ερευνητικής υποδομής

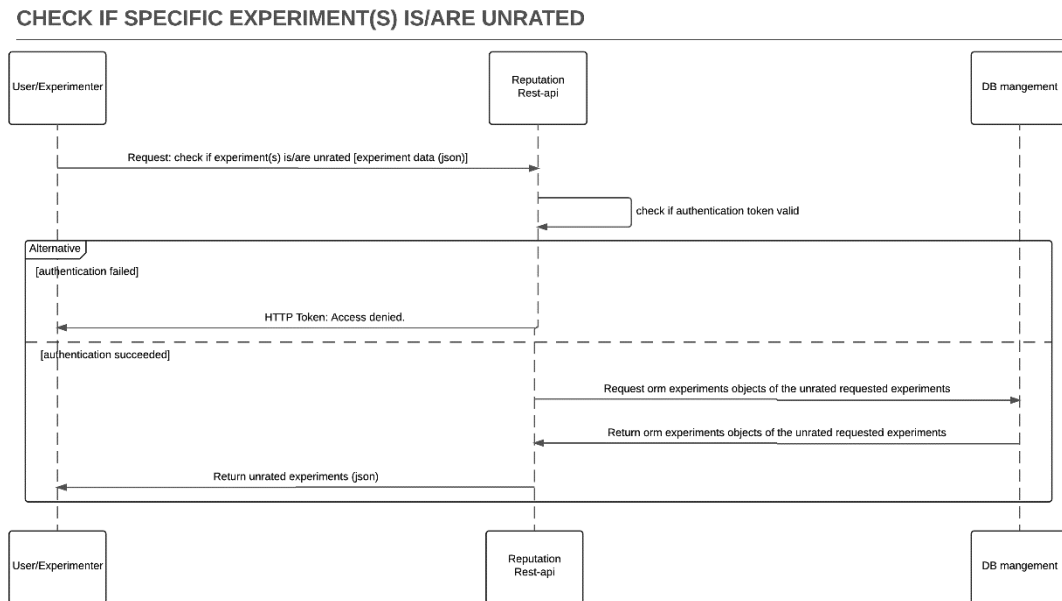


Σχήμα 5.4 - Λήψη της φήμης κάθε ερευνητικής υποδομής

Η κλήση αυτή πραγματοποιείται από απλούς χρήστες που θέλουν να μάθουν την φήμη κάθε ερευνητικής υποδομής για κάθε υπηρεσία που παρέχουν. Η κλήση γίνεται χωρίς παραμέτρους με απλή κλήση GET στην κατάλληλη διεύθυνση όπως φαίνεται και στο παράρτημα Α. Μετά την κλήση του χρήστη, ζητείται από την βάση με μορφή αντικειμένου κάθε εγγραφή στον πίνακα testbeds (βλέπε σχήμα 4.4). Σε περίπτωση που δεν υπάρχει καμία εγγραφή και ο πίνακας αντικειμένων που θα επιστραφεί είναι κενός η διαδικασία σταματάει και η εφαρμογή μας στέλνει στον χρήστη το κατάλληλο μήνυμα. Σε αντίθετη περίπτωση η εφαρμογή μας επιστρέφει σε μορφή json την φήμη κάθε ερευνητικής υποδομής για κάθε υπηρεσία που παρέχει.



## 5.1.6 Έλεγχος βαθμολόγησης

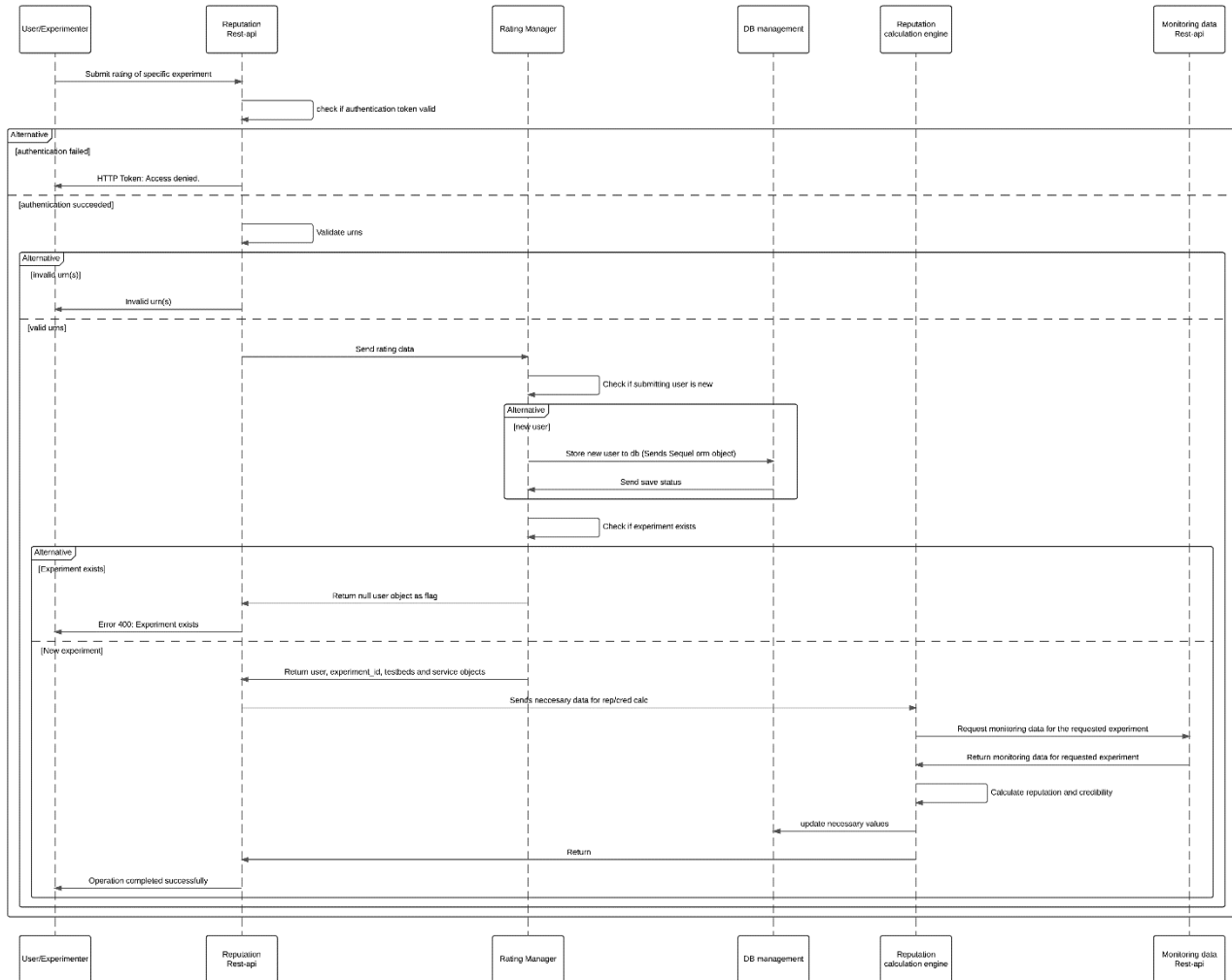


Σχήμα 5.5 - Έλεγχος αν συγκεκριμένο/α πείραμα/ατα είναι βαθμολογημένα

Με αυτήν την λειτουργία, ο χρήστης μπορεί να ελέγξει αν για κάποιο από τα πειράματα που έχει εκτελέσει δεν έχει υποβάλλει βαθμολογία στο σύστημα. Ο χρήστης με χρήση κατάλληλου προγράμματος/εργαλείου πελάτη (client) στέλνει σε μορφή json τα στοιχεία των πειραμάτων τα οποία θέλει να ελέγξει. Αρχικά ελέγχεται η εγκυρότητα του κλειδιού (token) που στάλθηκε ως επικεφαλίδα (header) χωρίς να ελέγχεται ο ρόλος του χρήστη αφού η κλήση αυτή δεν είναι διαχειριστική. Σε περίπτωση μη έγκυρου κλειδιού αποστέλλεται κατάλληλο μήνυμα στο χρήστη. Έπειτα η εφαρμογή μας ζητάει από την βάση να επιστραφούν σε μορφή αντικειμένου οι εγγραφές του πίνακα experiments (βλ. σχήμα βάσης) οι οποίες έχουν ακριβώς τα στοιχεία που έστειλε ο χρήστης και στο πεδίο βαθμολογημένο (rated) έχουν την τιμή ψευδές. Έπειτα η εφαρμογή μας στέλνει τα πειράματα αυτά πίσω στον χρήστη σε μορφή json. Στην περίπτωση που όλα τα πειράματα που απέστειλε ο χρήστης είναι βαθμολογημένα θα επιστρέψει κενό.

## 5.1.7 Υποβολή αξιολόγησης – υπολογισμός φήμης/αξιοπιστίας.

### SUBMIT RATING - UPDATE REPUTATION/CREDIBILITY



Σχήμα 5.6 - Υποβολή αξιολόγησης – υπολογισμός φήμης/αξιοπιστίας

Ο χρήστης/ερευνητής υποβάλλει την βαθμολογία του με χρήση κατάλληλου προγράμματος/εργαλείου σε μορφή json. Αρχικά η εφαρμογή ακολουθεί την διαδικασία πιστοποίησης απλού χρήστη όπως περιγράφεται και στην ενότητα 5.1.2. Αφού πιστοποιηθεί ο χρήστης, η εφαρμογή μας ελέγχει την εγκυρότητα των απεσταλμένων urns και επιστρέφει μήνυμα σφάλματος στην περίπτωση που αποτύχει κάποιο στον έλεγχο. Σε αντίθετη περίπτωση αποστέλλονται τα κατάλληλα δεδομένα για την υποβολή της αξιολόγησης στον διαχειριστή βαθμολογιών (Rating manager). Σε αυτό το στάδιο ελέγχεται αν ο χρήστης που υποβάλλει την αξιολόγηση είναι νέος ή αν είναι ήδη καταχωρημένος στο σύστημα. Στην περίπτωση που είναι νέος, στέλνονται τα στοιχεία τους στην βάση προς αποθήκευση. Έπειτα γίνεται έλεγχος αν έχει αξιολογηθεί και καταχωρηθεί το συγκεκριμένο πείραμα που έστειλε ο

χρήστη. Στην περίπτωση όπου το πείραμα έχει ξανά καταχωρηθεί και αξιολογηθεί, στέλνεται στην εφαρμογή μας ένα κενό αντικείμενο χρήστη, ως σήμα, και η εφαρμογή μας επιστρέφει στον χρήστη μήνυμα, ότι το πείραμα υπάρχει και έχει αξιολογηθεί. Στην περίπτωση που είναι καινούργιο πείραμα ο rating manager επιστρέφει ως αντικείμενα όλα τα στοιχεία του χρήστη, των ερευνητικών υποδομών και των υπηρεσιών που εμπλέκονται στο πείραμα με την μορφή αντικειμένου. Έπειτα η εφαρμογή μας αποστέλλει στον μηχανισμό υπολογισμού φήμης (reputation calculation engine) όλα τα απαραίτητα δεδομένα για τον υπολογισμό των τιμών φήμης και αξιοπιστίας. Στο επόμενο βήμα ο μηχανισμός υπολογισμού φήμης αιτείται και λαμβάνει να δεδομένα παρακολούθησης για το συγκεκριμένο πείραμα από την rest διεπαφή για τα δεδομένα παρακολούθησης. Στην συνέχεια υπολογίζονται οι τιμές όπως περιγράφεται στο κεφάλαιο 3. Γίνεται αίτημα στην βάση για την ανανέωση των τιμών φήμης αξιοπιστίας και όλων των παραμέτρων που άλλαξαν από την διαδικασία αυτή. Τέλος, ο μηχανισμός υπολογισμού φήμης επιστρέφει, και η εφαρμογή μας αποστέλλει μήνυμα στον χρήστη ότι η διαδικασία ολοκληρώθηκε.

## 5.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Στον πίνακα 5.1 ακολουθεί η περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση του rest-api και της κεντρικής αρχής διαχείρισης φήμης καθώς και της προσομοίωσης της αρχής παροχής δεδομένων παρακολούθησης.

A/A	Τεχνολογίες	Έκδοση Λογισμικού	Σχετιζόμενες Συνιστώσες Συστήματος
1	Ruby	2.2.0	Reputation and testbed management – rest api
2	Ruby on Rails	4.2.6	
3	Sqlite3 gem	1.3.11	database
4	Sequel	4.34.0	Orm db management
5	Sequel-rails gem	0.9.12	
6	Rails-api	0.4.0	Ruby on Rails api only optimization
7	Urn	2.0	Urn validation
8	Jbuilder	2.0	Reputation api json response formatting
9	PHP-CRUD-API	-	Monitoring data rest-api
10	PHP	5.3.10	
11	Let's Encrypt gem	0.0.9	

Πίνακας 5.1 – Προγραμματιστικά εργαλεία

## **6 Μελλοντικές επεκτάσεις**

Σε αυτήν την ενότητα θα παρουσιάσουμε κάποιες ιδέες για την μελλοντική επέκταση της δουλειάς που εκπονήθηκε στο πλαίσιο της συγκεκριμένης εργασίας θεωρητικά και σε επίπεδο υλοποίησης

### **6.1 Ενσωμάτωση σε ομοσπονδία ερευνητικών υποδομών και δεδομένα παρακολούθησης**

Ο μηχανισμός υπολογισμού και διαχείρισης φήμης και η διεπαφή του με ελάχιστες τροποποιήσεις μπορεί να ενσωματωθεί σε κάποια ομοσπονδία ερευνητικών υποδομών όπως το Fed4FIRE<sup>1</sup>. Το κομμάτι της υλοποίησης μας, το οποίο θα χρειαστεί επανασχεδιασμό είναι το κομμάτι των δεδομένων παρακολούθησης τα οποία και προσομοιώθηκαν στο πλαίσιο της παρούσας εργασίας. Σε ένα πραγματικό ομόσπονδο περιβάλλον θα χρειαστεί να ληφθούν διάφορες σχεδιαστικές επιλογές για την πλήρη ενσωμάτωση και υλοποίηση του μηχανισμού μας σε σχέση με τα δεδομένα παρακολούθησης. Η πρώτη επιλογή αφορά αν θα υπάρχει μια κεντρική διεπαφή για την αποστολή των δεδομένων παρακολούθησης όλων των ερευνητικών υποδομών είτε κάθε ερευνητική υποδομή θα διαθέτει τη δική της διεπαφή για την παροχή των δεδομένων παρακολούθησης των πειραμάτων που διεξάγονται σε αυτές. Κάθε μία από τις παραπάνω επιλογές έχει διαφορετικά προτερήματα τα οποία και θα πρέπει να αξιολογηθούν αναλυτικά για να επιλεγεί η βέλτιστη υλοποίηση. Σε μία πρώτη ανάγνωση η κεντρική διεπαφή καθιστά απλούστερη την υλοποίηση του μηχανισμού διαχείρισης φήμης καθώς θα χρειαστεί λιγότερες τροποποιήσεις και παραμετροποιήσεις παρ' όλα αυτά θα καταστήσει το σύστημα δυσκολότερα επεκτάσιμο με νέες ερευνητικές υποδομές κάτι που θεωρείται μεγάλο μειονέκτημα καθώς η επεκτασιμότητα είναι στόχος των ομόσπονδων ερευνητικών υποδομών.

### **6.2 Εμπλουτισμός του αλγορίθμου υπολογισμού φήμης**

Ένα δεύτερο επίπεδο στο οποίο μπορεί να επεκταθεί και να εμπλουτιστεί η παρούσα εργασία είναι ο μηχανισμός υπολογισμού διαχείρισης φήμης. Ο μηχανισμός που

---

<sup>1</sup> [www.fed4fire.eu](http://www.fed4fire.eu)

χρησιμοποιείται στη παρούσα εργασία (ROCQ) χρησιμοποιεί τις αξιολογήσεις χρηστών για τον υπολογισμό των τιμών φήμης με το μαθηματικό μοντέλο που περιγράφεται στο κεφάλαιο 3 και χρησιμοποιεί τα δεδομένα παρακολούθησης για να αυξομειώνει την αξιοπιστία των χρηστών με βάση το προφίλ της κάθε αξιολόγησης τους (π.χ. κακόβουλη, συντηρητική). Οι ομόσπονδες πλατφόρμες ερευνητικών υποδομών χρησιμοποιούν συμφωνίες στάθμης ποιότητας υπηρεσίας (Service Level Agreement – SLA) [31]. Τα SLA επί της ουσίας είναι συμβόλαια που συνάπτονται για την τήρηση συγκεκριμένων προδιαγραφών στην παροχή υπηρεσιών. Τέτοια χαρακτηριστικά μπορεί να είναι η διαθεσιμότητα, η ταχύτητα δικτύου κ.ο.κ. τα οποία είναι και κρίσιμο να τηρούνται για την ομαλή διεξαγωγή των πειραμάτων στις ερευνητικές υποδομές. Έτσι, καθώς στα πλαίσια των ομόσπονδων ερευνητικών καταγράφονται και αποθηκεύονται όλες οι παραβάσεις των SLA από τις ερευνητικές υποδομές, οι παραβάσεις αυτές θα μπορούσαν να συνδράμουν στον υπολογισμό των τιμών φήμης. Αυτός ο εμπλουτισμός πέρα του ότι θα καταστήσει πιο πλήρη το μηχανισμό υπολογισμού φήμης αφού θα λαμβάνει υπόψη περισσότερες παραμέτρους, δυνητικά θα μπορεί να δώσει και αντιπροσωπευτικές τιμές φήμης ακόμα και χαμηλό αριθμό αξιολογήσεων ανά ερευνητική υποδομή.

## 7 Βιβλιογραφία

- [1] Jøsang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2), 618-644.
- [2] Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., ... & Seskar, I. (2014). GENI: A federated testbed for innovative network experiments. *Computer Networks*, 61, 5-23.
- [3] Gavras, A., Karila, A., Fdida, S., May, M., & Potts, M. (2007). Future internet research and experimentation: the FIRE initiative. *ACM SIGCOMM Computer Communication Review*, 37(3), 89-92.
- [4] Peterson, L., Ricci, R., Falk, A., & Chase, J. (2010). Slice-based federation architecture. *Ad Hoc Design Document* (July 2008), 1.
- [5] Peer-to-peer Systems, Wikipedia (February 2017)  
<https://en.wikipedia.org/wiki/Peer-to-peer>
- [6] Resnick, P., Zeckhauser, R., Swanson, J., & Lockwood, K. (2006). The value of reputation on eBay: A controlled experiment. *Experimental economics*, 9(2), 79-101.
- [7] Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003, May). The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web* (pp. 640-651). ACM.
- [8] Su, X., Zhang, M., Mu, Y., & Sim, K. M. (2010, December). Pbtrust: A priority-based trust model for service selection in general service-oriented environments. In *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on* (pp. 841-848). IEEE.
- [9] Garg, A., & Battiti, R. (2004). The reputation, opinion, credibility and quality (ROCQ) scheme.
- [10] Kapoukakis, A., Kafetzoglou, S., Androulidakis, G., Papagianni, C., & Papavassiliou, S. (2014, December). Reputation-Based Trust in federated testbeds utilizing user experience. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)* (pp. 56-60). IEEE.
- [11] Mekouar, L., Iraqi, Y., & Boutaba, R. (2010). Reputation-based trust management in peer-to-peer systems: taxonomy and anatomy. In *Handbook of Peer-to-Peer Networking* (pp. 689-732). Springer US.
- [12] Shmatikov, V., & Talcott, C. (2005). Reputation-based trust management. *Journal of Computer Security*, 13(1), 167-190.
- [13] PHP CRUD API documentation, (February 2017), Available at:  
<https://github.com/mevdschee/php-crud-api>
- [14] Sequel: The Database Toolkit for Ruby, (February 2017), Available at  
<http://sequel.jeremyevans.net/>
- [15] Ripeanu, M. (2001, August). Peer-to-peer architecture case study: Gnutella network. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on* (pp. 99-100). IEEE.

- [16] Ruby on Rails Documentation, (February 2017), Available at:  
<http://api.rubyonrails.org/>
- [17] Model-View-Controller, (February 2017), Available at:  
[https://en.wikibooks.org/wiki/Ruby\\_on\\_Rails/Getting\\_Started/Model-View-Controller](https://en.wikibooks.org/wiki/Ruby_on_Rails/Getting_Started/Model-View-Controller)
- [18] Rails API gem Documentation, (February 2017), Available at:  
<https://github.com/rails-api/rails-api>
- [19] Jbuilder gem Documentation, (February 2017), Available at:  
<https://github.com/rails/jbuilder>
- [20] Object Relational Mapping, Wikipedia, (February 2017), Available at:  
[https://en.wikipedia.org/wiki/Object-relational\\_mapping](https://en.wikipedia.org/wiki/Object-relational_mapping)
- [21] Active Record Documentation, (February 2017), Available at:  
[http://api.rubyonrails.org/files/activerecord/README\\_rdoc.html](http://api.rubyonrails.org/files/activerecord/README_rdoc.html)
- [22] Ode to sequel, (July 2015), Available at:  
<https://twin.github.io/ode-to-sequel/>
- [23] Sequel vs Active Record ORM Benchmark, (September 2015), Available at:  
<http://www.railsmine.net/2015/09/sequel-vs-active-record-orm-benchmark-postgresql.html>
- [24] Service Objects in Rails, (September 2014), Available at:  
<https://www.netguru.co/blog/service-objects-in-rails-will-help>
- [25] Gourmet Service Objects, (April 2014), Available at:  
<http://brewhouse.io/blog/2014/04/30/gourmet-service-objects.html>
- [26] Token Based Authentication in Rails, (February 2014), Available at:  
<https://www.codeschool.com/blog/2014/02/03/token-based-authentication-rails/>
- [27] Mean Opinion Score, Wikipedia, (February 2017)  
[https://en.wikipedia.org/wiki/Mean\\_opinion\\_score](https://en.wikipedia.org/wiki/Mean_opinion_score)
- [28] About Let's Encrypt, Available at:  
<https://letsencrypt.org/about/>
- [29] eBay e-commerce platform, Available at:  
<http://www.ebay.com/>
- [30] Amazon e-commerce platform, Available at:  
<https://www.amazon.com/>
- [31] Service Level Agreement, Wikipedia, (February 2017), Available at:  
[https://en.wikipedia.org/wiki/Service-level\\_agreement](https://en.wikipedia.org/wiki/Service-level_agreement)

## Παράρτημα Α – περιγραφή rest-api

<b>POST /api/testbed</b>
<b>Description:</b> add testbed and its services (admin)
<b>Request example:</b> <pre>curl -X POST -H "Content-Type: application/json" -d '{     "testbed": "urn:publicid:IDN+ple+authority+cm",     "alias": "PLE",     "services": ["Overall", "Availability"],     "type": "PLC" }' "http://newvy.netmode.ntua.gr/api/testbed/"</pre>
<b>Parameters (with value example):</b> (Content-type: application/json): <pre>{     "testbed": "string(testbed urn)",     "alias": "string",     "services": ["string", "string"]    "experiments": [         {             "user_urn": "urn:publicid:IDN+fed4fire:global+user+chrisap",             "slice_urn": "urn:publicid:IDN+fed4fire:global:review+slice+reservation",             "start_time": "2015-12-07T15:00:00+02:00",             "end_time": "2015-12-07T15:30:00+02:00"         },         {             "user_urn": "urn:publicid:IDN+fed4fire:global+user+chrisap",             "slice_urn": "urn:publicid:IDN+fed4fire:global:review+slice+exp1",             "start_time": "2015-12-06T17:00:00+02:00",             "end_time": "2015-12-06T19:00:00+02:00"         }     ] },     "type": "string" }</pre>
<b>Response Representations:</b> i) Success <b>Code 200 (OK):</b> "The operation completed successfully" ii) Error codes and messages <b>Code 400 (Bad request):</b> "Testbed exists" <b>Code 500</b> (Internal server error)

<b>GET /api/reputation/show?testbed=&lt;TESTBED_URN&gt;</b>
<b>Description:</b> request reputation of specific testbed (user)
<b>Parameters (with value example):</b> testbed=[string(testbed_urn)] example: testbed=urn:publicid:IDN+ple+authority+cm
<b>Request example:</b>



<http://newvy.netmode.ntua.gr/api/reputation/show?testbed=urn:publicid:IDN+ple+authority+cm>

**Response Representations:** i) Success

**Code 200 (OK):**

```
{
  "Testbed_alias": [
    {
      "Service name": "Reputation of service"
    },
    {
      "Service name": "Reputation of service"
    }
  ]
}
```

ii) Error codes and messages

**Code 400 (Bad request):** "Testbed not found"

**Code 500 (Internal server error)**

**DEL /api/testbed/<TESTBED\_URN>**

**Description:** delete testbed and its services (admin)

**Parameters (with value example):** [string(testbed\_urn)]

example: urn:publicid:IDN+ple+authority+cm

**Request example:**

curl -X DELETE

"http://newvy.netmode.ntua.gr/api/testbed/urn:publicid:IDN+ple+authority+cm"

**Response Representations:** i) Success

**Code 200 (OK):** "Testbed deleted successfully":

ii) Error codes and messages

**Code 400 (Bad request):** "Testbed not found"

**Code 500 (Internal server error)**

**PUT /api/testbed/**

**Description:** update testbed and its services parameters (admin)

**Parameters (with value example):** (Content-type: application/json)

```
{ "testbed": "string(testbed_urn)",
  "alias": "string(testbed_alias)",
  "type": "string(AM)",
  "services": [ "string(servicename)",
               "....",
               "string(servicename)" ]
}
```

**Example:** (Content-type: application/json)

```
{ "testbed": "urn:publicid:IDN+ple+authority+cm", "services": [ "Availability" ] }
```

**Request example:**

```
curl -X PUT -H "Content-Type: application/json" -d
'{"testbed":"urn:publicid:IDN+ple+authority+cm","services":["Cpu']}'
http://newvy.netmode.ntua.gr/api/testbed/"
```

**Response Representations:** i) Success

**Code 200 (OK):** "The operation completed successfully":

ii) Error codes and messages

**Code 400 (Bad request):** "Testbed not found"

**Code 500 (Internal server error)**

### GET /api/reputation/showrep

**Description:** request reputation of every testbed (user)

**Request example:**

http://newvy.netmode.ntua.gr/api/reputation/showrep

**Response Representations:** i) Success (with example)

**Code 200 (OK):**

Content-type: application/json

```
{
  "testbeds": [
    {
      "testbed_name": "wall1.ilabt.iminds.be",
      "services": [
        {
          "Overall":1
        },
        {
          "Availability":1
        }
      ]
    },
    {...},
  ]
}
```

ii) Error codes and messages

**Code 500 (Internal server error)**

### POST /api/reputation/exp

**Description:** check if specific experiment(s) is/are unrated (user)

**Parameters (with value example):** (Content-type: application/json)

```
{
  "experiments":[
    {
      "user_urn":"string(user_urn)",
      "slice_urn":"string(slice_urn)",
      "start_time":"string(reservation_start(rfc3339))",
      "end_time":"string(reservation_start(rfc3339))"
    }
  ]
}
```

**Example:** (Content-type: application/json)

```
{
  "experiments": [
    {
      "user_urn": "urn:publicid:IDN+fed4fire:global+user+chrisap",
      "slice_urn": "urn:publicid:IDN+fed4fire:global:review+slice+reservation",
      "start_time": "2015-12-07T15:00:00+02:00",
      "end_time": "2015-12-07T15:30:00+02:00"
    },
    {
      "user_urn": "urn:publicid:IDN+fed4fire:global+user+chrisap",
      "slice_urn": "urn:publicid:IDN+fed4fire:global:review+slice+exp1",
      "start_time": "2015-12-06T17:00:00+02:00",
      "end_time": "2015-12-06T19:00:00+02:00"
    }
  ]
}
```

**Request example:**

```
curl -X POST -H "Content-Type: application/json" -d
'{"experiments":[{"user_urn":"urn:publicid:IDN+fed4fire:global+user+chrisap","slice_urn":
urn:publicid:IDN+fed4fire:global:review+slice+reservation","start_time":"2015-12-
07T15:00:00+02:00","end_time":"2015-12-
07T15:30:00+02:00"},{"user_urn":"urn:publicid:IDN+fed4fire:global+user+chrisap","slice_u
rn":"urn:publicid:IDN+fed4fire:global:review+slice+exp1","start_time":"2015-12-
06T17:00:00+02:00","end_time":"2015-12-06T19:00:00+02:00"}]}'
"http://newvy.netmode.ntua.gr/api/reputation/exp"
```

**Response Representations:** i) Success (with example)

**Code 200 (OK):**

Content-type: application/json

```
{
  "Unrated": [
    {
      "experiment_id": 2,
      "slice_urn": "urn:publicid:IDN+fed4fire:global:review+slice+exp1",
      "rated": false,
      "reservation_start": "2015-12-06T17:00:00+02:00",
      "reservation_end": "2015-12-06T19:00:00+02:00",
      "eid": "2",
      "user_urn": "urn:publicid:IDN+fed4fire:global+user+chrisap"
    }
  ]
}
```

ii) Error codes and messages

**Code 500 (Internal server error)**

**POST /api/reputation/userqoe**

**Description:** submit rating – update reputation/credibility

**Parameters (with value example):** (Content-type: application/json)

```

{
  "user_urn": "string(user_urn)",
  "slice_urn": "string(slice_urn)",
  "start_time": "string(reservation_start (rfc3339))",
  "end_time": "string(reservation_end(rfc3339))",
  "resources": ["string(resource_urn)],
  "user_eval": {"string(service_name)": int(1-5)(rating),
  "string(service_name)": int(1-5)(rating)}
}

```

**Example:** (Content-type: application/json)

```

{
  "user_urn": "urn:publicid:IDN+wall2.ilabt.iminds.be+user+bvermeu2",
  "slice_urn": "urn:publicid:IDN+wall2.ilabt.iminds.be:berlin+slice+rep6",
  "start_time": "2016-04-04T12:35:48+01:00",
  "end_time": "2016-04-04T13:36:00+01:00",
  "resources": ["urn:publicid:IDN+wall1.ilabt.iminds.be+sliver+102798",
  "urn:publicid:IDN+wall1.ilabt.iminds.be+sliver+102799"],
  "user_eval": {"overall":5, "quality":5, "availability":5}
}

```

**Request example:**

```

curl -X POST -H "Content-Type: application/json" -d '{
  "user_urn": "urn:publicid:IDN+wall2.ilabt.iminds.be+user+bvermeu2",
  "slice_urn": "urn:publicid:IDN+wall2.ilabt.iminds.be:berlin+slice+rep6",
  "start_time": "2016-04-04T12:35:48+01:00",
  "end_time": "2016-04-04T13:36:00+01:00",
  "resources": ["urn:publicid:IDN+wall1.ilabt.iminds.be+sliver+102798",
  "urn:publicid:IDN+wall1.ilabt.iminds.be+sliver+102799"],
  "user_eval": {"overall":5, "quality":5, "availability":5}
}' "http://newvy.netmode.ntua.gr/api/reputation/userqoe"

```

**Response Representations:** i) Success (with example)

**Code 200 (OK):** "The operation completed successfully"

ii) Error codes and messages

**Code 400 (Bad request):** "Experiment exists"

**Code 500 (Internal server error)**