ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

ΠΡΟΤΥΠΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΔΗΜΟΣΙΕΥΣΗ ΚΑΙ ΣΥΛΛΟΓΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΣΤΑ ΚΟΙΝΩΝΙΚΑ ΜΕΣΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ ΣΠΥΡΑΚΗ

Επιβλέπων : **Δημήτριος Ασκούνης**
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος, 2011

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

ΠΡΟΤΥΠΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΔΗΜΟΣΙΕΥΣΗ ΚΑΙ ΣΥΛΛΟΓΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ ΣΤΑ ΚΟΙΝΩΝΙΚΑ ΜΕΣΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΠΑΝΑΓΙΩΤΗ ΣΠΥΡΑΚΗ

Επιβλέπων : Δημήτριος Ασκούνης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ***9η Νοεμβρίου 2011.***

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Γρηγόριος Μέντζας Καθηγητής Ε.Μ.Π.	Ιωάννης Ψαρράς Καθηγητής Ε.Μ.Π.	Δημήτριος Ασκούνης Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος, 2011

(Υπογραφή)

.....

ΠΑΝΑΓΙΩΤΗΣ ΣΠΥΡΑΚΗΣ

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας υπήρξε η διερεύνηση και μελέτη διαφορετικών υπαρχουσών τεχνολογιών και αρχιτεκτονικών που μπορούν να εφαρμοστούν στον παγκόσμιο ιστό προκειμένου να δημιουργηθεί ένα σύστημα ικανό να λαμβάνει και να αποστέλλει περιεχόμενο σε πραγματικό χρόνο από και προς τα κοινωνικά μέσα δικτύωσης. Για το σκοπό αυτό εξετάσθηκαν αφενός διαδεδομένες και ώριμες λύσεις όπως τα Simple Object Access Protocol (SOAP) και REpresentational State Transfer (REST) με πιο κατάλληλο το δεύτερο και αφετέρου ένα νέο πρωτόκολλο με την ονομασία eXtensible Messaging and Presence Protocol (XMPP) το οποίο αναπτύσσεται την τελευταία δεκαετία. Ύστερα από διεξοδική ανάλυση των πτυχών καθενός επιλέχθηκε το XMPP ως ιδανικό για τη δημιουργία ενός συστήματος που λειτουργεί ως υπηρεσία Δημοσιεύσεων-Συνδρομών (Publish-Subscribe) όπου διατηρούνται κόμβοι ώστε να μπορεί ένας χρήστης να δημοσιεύσει ένα μήνυμα στα κοινωνικά μέσα αλλά και να δεχθεί όλα τα μηνύματα αλληλεπίδρασης της αρχικής δημοσίευσης μέσα από αυτούς σε πραγματικό χρόνο.

Συγκεκριμένα, το σύστημα παρέχει τη δυνατότητα δημοσίευσης μηνυμάτων από μία εφαρμογή πελάτη XMPP στο Facebook και την λήψη των ειδοποιήσεων για κάθε νέο σχόλιο (comment) ή εκδήλωση προτίμησης (like) πίσω σε αυτή. Για τη διασύνδεση με το Facebook χρησιμοποιήθηκε το Graph API (web API του Facebook) τόσο για την ταυτοποίηση της πρόσβασης της εφαρμογής του συστήματος στον προσωπικό λογαριασμό χρήστη μέσω OAuth όσο και για την δυνατότητα δημοσίευσης μέσω αυτού στο προφίλ και την ανάκτηση πληροφοριών από αυτό. Το XMPP παρέχει ένα τρόπο ανταλλαγής δομημένης πληροφορίας σε πραγματικό χρόνο και προσφέρει μεγάλες δυνατότητες επεκτασιμότητας. Πέραν όμως της επεκτασιμότητας του ίδιου του πρωτοκόλλου, η δομή του υλοποιημένου συστήματος επιτρέπει τόσο την επέκταση του σε άλλα κοινωνικά μέσα δικτύωσης όσο και την προσθήκη νέων υποσυστημάτων. Το σύστημα αποτελεί μία βάση στην οποία μπορούν να δομηθούν ακόμα πιο σύνθετα συστήματα ειδοποιήσεων με εφαρμογές στον παγκόσμιο ιστό και στα κοινωνικά μέσα δικτύωσης .

Λέξεις Κλειδιά: Κοινωνικά Μέσα, Παγκόσμιος Ιστός, Πρωτόκολλα Πραγματικού Χρόνου, Υπηρεσίες Ιστού, XMPP, Δημοσίευση, Συνδρομή, Σύστημα Ειδοποιήσεων, REST

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The scope of this thesis was the research and study of different existing technologies and architectures implemented in worldwide web in order to create a software system capable of both receiving content from social media platforms and sending content back at them in real-time. On one hand, existing mature web architecture solutions like Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) were studied, with the latter being the most widespread, and on the other hand a freshly new protocol which is being developed during the last decade was examined, named eXtensible Messaging and Presence Protocol (XMPP). After thoroughly analyzing the aspects of each architectural approach the XMPP protocol was selected as the most appropriate for the creation of a system that acts like a Publish-Subscribe service where preserved nodes allow specific users to publish messages on social media and receive all of their feedback messages through the nodes' realtime notifications.

In particular, the system provides the capability of posting messages on Facebook and receiving their feedback as Facebook "comments" or "likes" by using an XMPP client software. The Facebook Graph web API was used for accomplishing the authorization procedure of the user's Facebook account using OAuth as well as for gaining access to its publishing and feed retrieving capabilities. The XMPP protocol provides a way of exchanging structured data in realtime and offers a great amount of extension capabilities. Apart from the protocol's built-in potential for extension, the structure of the developed system allows the connection to several other social media platforms or new independent components. This system can be used as a foundation on which more complex notification systems, that function within the scope of the world wide web and the social media, can be built.

Keywords: Social Media, Worldwide Web, Realtime protocols, Web Services, XMPP, Publish, Subscribe, Notification System, REST

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή Δημήτριο Ασκούνη καθώς και τον υποψήφιο διδάκτορα Ιωσήφ Αλβέρτη που με βοήθησαν σημαντικά στην εκπόνηση της διπλωματικής αυτής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω και το συμφοιτητή Μιχάλη Πετυχάκη για τη συμβολή του καθόλη τη διάρκεια ανάπτυξης της εργασίας αυτής.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Περιεχόμενα

1

Εισαγωγή

1.1 Χώρος Εφαρμογής

Με την εμφάνιση του web 2.0 ως εξέλιξη της πρότερης στατικής μορφής του γνωστού σε όλους world wide web (Όπως προτάθηκε και παρουσιάστηκε αρχικά το 1990 από τον Tim Burners Lee) ανέκυψαν οι έννοιες της άμεσης επικοινωνίας και συνεργασίας μεταξύ τελικών χρηστών του παγκόσμιου ιστού καθώς επίσης της διαλειτουργικότητας και διαδραστικότητας σε ιστοχώρους. Διαδικτυακές υπηρεσίες και εφαρμογές άρχισαν να σχεδιάζονται με βάση νέες αρχιτεκτονικές προκειμένου οι υπηρεσίες που προσφέρουν στον τελικό χρήστη να διευρυνθούν. Ως υπηρεσία ιστού (web service) το W3C [27] ορίζει ένα σύστημα λογισμικού που επιτρέπει την διαλειτουργική επικοινωνία μεταξύ δύο μηχανημάτων (εφαρμογών) σε ένα δίκτυο [11]. Ο επικρατέστερος τρόπος υλοποίησης μίας υπηρεσίας σήμερα είναι η REST αρχιτεκτονική με την εμφάνιση της οποίας άρχισε και η ανάπτυξη των Web APIs, δηλαδή των διαπροσωπειών διαδικτυακών εφαρμογών . Τα Web APIs μετέφεραν την ποιότητα των παρεχόμενων υπηρεσιών σε άλλο επίπεδο καθώς αποτελούν μία διαπροσωπεία την οποία μπορεί να χρησιμοποιήσει μία εξωτερική εφαρμογή ή ένας τελικός χρήστης για να έχει πρόσβαση σε δεδομένα της ίδιας της υπηρεσίας. Τα παραπάνω συνετέλεσαν στο γεγονός ότι οι τελικοί χρήστες του διαδικτύου μπόρεσαν πλέον να αλληλεπιδρούν μεταξύ τους καθώς οι υπηρεσίες τις οποίες χρησιμοποιούσαν τους παρείχαν την πρόσθετη δυνατότητα να επεμβαίνουν άμεσα σε αυτές. Κατ'αυτόν τον τρόπο ο παγκόσμιος ιστός κατακλύσθηκε από διαδικτυακές εφαρμογές και υπηρεσίες οι οποίες ήταν

ικανές να επιτρέπουν τη συμμετοχή του χρήστη και τη δυναμική τροποποίηση περιεχομένου από αυτόν. Σημαντικότερη εφαρμογή των παραπάνω αποτελούν οι ιστότοποι κοινωνικών μέσων δικτύωσης που παρέχουν ολοκληρωμένη λειτουργικότητα ως προς τη δημοσίευση-διαχείριση προφίλ χρηστών με συγκεκριμένη ή γενικευμένη θεματολογία και δίνουν τη δυνατότητα της δημιουργίας περιεχομένου από τους χρήστες άμεσα προσβάσιμο από τρίτους και διαθέσιμο προς σχολιασμό – επεξεργασία. Ως αποτέλεσμα των παραπάνω δημιουργήθηκε ένα ιστός από διαφορετικές πλατφόρμες και δίκτυα όπου οι ίδιοι οι χρήστες φρόντισαν να εκθέσουν υλικό που αφορά την καθημερινότητά τους και σχετίζεται με προϊόντα, υπηρεσίες και οτιδήποτε άλλο τους ενδιαφέρει καθώς και τις εντυπώσεις τους για τα παραπάνω. Όλο αυτό το υλικό που συνεχώς αυξάνεται και ανανεώνεται αποτέλεσε αντικείμενο μελέτης για πολλούς καθώς η εξαγωγή συμπερασμάτων από την επεξεργασία του μπορεί να αποδειχθεί ιδιαίτερα ωφέλιμη για την αξιολόγηση, την βελτίωση ή τον ανασχεδιασμό μίας υπηρεσίας, ενός προϊόντος ή του κοινωνικού προφίλ μίας ομάδας.

Είναι, λοιπόν, εμφανές ότι η επικοινωνία μεταξύ χρηστών ανά τον κόσμο και ο διαμοιρασμός περιεχομένου στον παγκόσμιο ιστό εξελίχθηκε σε μία νέα μορφή στην οποία σημαντικός παράγοντας διαμόρφωσης υπήρξε η αμεσότητα με την οποία ο χρήστης έχει πρόσβαση στο περιεχόμενο που τον αφορά αλλά και ο χρόνος στον οποίο λαμβάνει ενημερώσεις και ειδοποιήσεις για νέο περιεχόμενο ή αλληλεπιδράσεις από τρίτους. Στα πλαίσια αυτής της σημαντικής εξέλιξης η συγκεκριμένη εργασία εξετάζει σύγχρονες τεχνολογίες που παρέχουν το τεχνικό υπόβαθρο για την υλοποίηση συστημάτων που εξασφαλίζουν τη μεταφορά της παραγόμενης πληροφορίας στον χρήστη σε σχεδόν πραγματικό χρόνο.

1.2 Αντικείμενο

Αντικείμενο της διπλωματικής αποτέλεσε η διερεύνηση εναλλακτικών τεχνολογιών με σκοπό την επιλογή της πλέον κατάλληλης για την υλοποίηση ενός συστήματος που θα λειτουργεί σαν εργαλείο για τη δημιουργία αλλά και την συνεχή και απρόσκοπτη συγκέντρωση περιεχομένου από τα κοινωνικά μέσα δικτύωσης προκειμένου να είναι αργότερα διαθέσιμο προς μελέτη και επεξεργασία. Ειδικότερα, πρόκειται για τη δημιουργία μίας εφαρμογής δημοσίευσης στρατηγικής στα κοινωνικά μέσα δικτύωσης στηριζόμενη στο πρωτόκολλο επικοινωνίας πραγματικού χρόνου XMPP. Η τεράστια προσέλκυση χρηστών σε πλατφόρμες κοινωνικών μέσων δικτύωσης καταδεικνύει την ανάγκη για αξιοποίηση και ενσωμάτωση νέων τεχνολογιών και πρωτοκόλλων σε ότι αφορά την ανάπτυξη εφαρμογών με σκοπό την

επικοινωνία με τα επικρατέστερα social media api . Η υλοποίηση αυτή επιδεικνύει τόσο τις δυνατότητες του συγκεκριμένου πρωτοκόλλου όσο και τη σημασία της ενσωμάτωσης τεχνολογιών και αρχιτεκτονικών real time instant messaging στη διαχείριση και προώθηση περιεχομένου από και προς τα social media. Πρόκειται για ένα σύστημα το οποίο επιτρέπει τη δημιουργία διαφορετικών κόμβων θεμάτων σε καθέναν από τους οποίους δημοσιεύονται μηνύματα με σκοπό να μεταδοθούν έπειτα στα κοινωνικά μέσα και στη συνέχεια κάθε σχολιασμός που γίνεται σε αυτά να επιστρέφεται σαν δημοσίευση σε ένα κόμβο αλληλεπίδρασης . Η επικοινωνία από τον κόμβο προς τα κοινωνικά μέσα και αντίστροφα καθώς και από τον εκδότη μίας δημοσίευσης προς τον κόμβο πραγματοποιείται με τη χρήση του πρωτοκόλλου XMPP επιτυγχάνοντας έτσι τη μετάδοση των μηνυμάτων σε πραγματικό χρόνο και παρέχοντας το πλεονέκτημα της υψηλής κλιμάκωσης. Κάθε κόμβος αναπαρίσταται σε μία βάση δεδομένων όπου καταγράφονται οι συνδρομητές του, οι επιτρεπόμενοι εκδότες και οι δημοσιεύσεις του. Κάθε οντότητα που συμμετέχει στο παραπάνω σύστημα, δηλαδή, ο εκδότης, το σύστημα που συντηρεί τους κόμβους και η διεπαφή που επικοινωνεί άμεσα με το api των κοινωνικών μέσων δικτύωσης αποτελεί συγχρόνως μία οντότητα XMPP η οποία επικοινωνεί με τις υπόλοιπες σε πραγματικό χρόνο. Κατορθώνουμε με αυτόν τον τρόπο να συλλέγουμε άμεσα περιεχόμενο από την αλληλεπίδραση στην κάθε διαφορετική δημοσίευση το οποίο αποθηκεύεται σε ένα κόμβο και μπορεί να αποσταλλεί σαν ειδοποίηση πραγματικού χρόνου στον εκδότη. Η επιτήρηση για αλληλεπίδραση σε μία ορισμένη δημοσίευση, η αποστολή μίας άλλης δημοσίευσης σε νέο κόμβο από άλλο εκδότη και η αποστολή ειδοποίησης για νέο περιεχόμενο σχολιασμού σε έναν τρίτο εκδότη είναι διαδικασίες που μπορεί το σύστημα να πραγματοποιήσει ταυτόχρονα καθώς η αρχιτεκτονική στην οποία στηρίζεται είναι ασύγχρονη και έχει σχεδιασθεί στη βάση της σαν σύστημα ανταλλαγής μηνυμάτων & ειδοποιήσεων σε πραγματικό χρόνο μεταξύ εκατομμύρια χρηστών.

1.3 Οργάνωση κειμένου

Στο κεφάλαιο 2 γίνεται εκτενής αναφορά σε τεχνολογίες που πρωταγωνιστούν στο χώρο εφαρμογής της συγκεκριμένης εργασίας. Στο κεφάλαιο 3 γίνεται η ανάλυση των απαιτήσεων για την πραγματοποίηση της εφαρμογής με τη βοήθεια διαγραμμάτων και σεναρίων χρήσης. Στο κεφάλαιο 4 προδιαγράφεται η σχεδίαση του συστήματος με τη χρήση διαγράμματος κλάσεων, λειτουργιών διαπροσωπείας και unified data model. Η υλοποίηση του συστήματος περιγράφεται στο κεφάλαιο 5. Το κεφάλαιο 6 αποτελεί τον επίλογο της εργασίας ενώ στο

κεφάλαιο 7 καταγράφονται οι αναφορές. Ακολουθεί το κεφάλαιο 8 με δύο παραρτήματα : Α) Οδηγός εγκατάστασης εργαλείων και Β) Οδηγός χρήσης της εφαρμογής.

2

Θεωρητικό υπόβαθρο

2.1 Κοινωνικά Μέσα

Τα κοινωνικά μέσα δικτύωσης είναι εφαρμογές βασισμένες στην ύπαρξη του παγκόσμιου ιστού των οποίων οι χρήστες μπορούν να επικοινωνούν μεταξύ τους με διαδραστικούς τρόπους και ανταλλάσοντας πληροφορία μέσα από ένα δυναμικά μεταβαλλόμενο γραφικό περιβάλλον. Δημιουργήθηκαν με σκοπό να αποτελέσουν ένα υπερσύνολο των υπαρχόντων τρόπων κοινωνικής αλληλεπίδρασης και να δώσουν την ευκαιρία στους χρήστες να παράγουν συνεχώς περιεχόμενο που προέρχεται από τις προσωπικές προτιμήσεις και συνήθειες τους, να το διαθέτουν στο πλαίσιο που αυτά παρέχουν και με αυτόν τον τρόπο οι υπόλοιποι χρήστες να αλληλεπιδρούν με αυτό. Είναι επόμενο, η ύπαρξη και η διάδοση των ποικίλων εφαρμογών κοινωνικών μέσων δικτύωσης να έχει προσδώσει νέα διάσταση στην επικοινωνία μέσω του παγκόσμιου ιστού καθώς ο κάθε χρήστης του παγκόσμιου ιστού μπορεί να διατηρεί ένα προσωπικό χώρο μέσω του λογαριασμού του στον οποίο απεικονίζεται η δραστηριότητά του η οποία είναι ανάλογη της θεματολογίας της συγκεκριμένης υλοποίησης. Στην ουσία αποτελούν μία αναπαράσταση του κοινωνικού ιστού σε μία μορφή ηλεκτρονικής κοινότητας όπου κάθε άτομο είναι μία αυτόνομη οντότητα που μπορεί να διαθέτει στο σύνολο τα δεδομένα που επιθυμεί προκειμένου να υπάρχει διάλογος και ανταλλαγή απόψεων. Τα κοινωνικά μέσα μπορεί να έχουν διαφορετικές μορφές με βασικά εξειδικευμένα παραδείγματα τα forums, τα blogs, οι πλατφόρμες micro-blogging, οι βιβλιοθήκες wiki, οι πλατφόρμες με υλικό βίντεο και εικόνας .

Οι Kaplan και Haenlein [5] δημιούργησαν ένα σχήμα κατηγοριοποίησης για τους διάφορους τύπους κοινωνικών μέσων σε άρθρο των Businness Horizons 2010 σύμφωνα με το οποίο τα κατατάσσουν στις παρακάτω κατηγορίες:

- συνεργατικά συστήματα (collaborative projects – π.χ. Wikipedia)
- πλατφόρμες δημοσίευσης άρθρων (ιστολόγια) και συμπυκνωμένων άρθρων (blogs και microblogs - Twitter)
- κοινότητες περιεχομένου (π.χ. YouTube)
- πλατφόρμες κοινωνικής δικτύωσης με γενικευμένη θεματολογία (π.χ. Facebook)
- εικονικοί κόσμοι παιχνιδιών
- εικονικοί κοινωνικοί κόσμοι [5]

Παρακάτω αναφέρεται ως παράδειγμα ένα από τα πιο διαδεδομένα μέσα από την κατηγορία των μέσων κοινωνικής δικτύωσης, το Facebook.

Facebook

Το Facebook είναι μία κοινωνική πλατφόρμα η οποία δημιουργήθηκε αρχικά το 2004 σαν ένα ηλεκτρονικό κοινωνικό δίκτυο όπου μέλη ήταν μόνο φοιτητές ενός και στη συνέχεια περισσοτέρων πανεπιστημίων ενώ στη συνέχεια το 2006 επέτρεψε την πρόσβαση σε όλους τους χρήστες του παγκόσμιου ιστού με μόνο περιορισμό την ύπαρξη ενός πραγματικού e-mail και το όριο ηλικίας. Από το 2009 και ύστερα η επισκεψιμότητα του άρχισε να αυξάνεται σταθερά .

Η πλατφόρμα του Facebook είναι μία ιστοσελίδα όπου ο χρήστης αφού συνδεθεί με τα προσωπικά του διαπιστευτήρια (credentials) έχει πρόσβαση στο προσωπικό του προφίλ το οποίο εμπλουτίζεται με πληροφορία όπως φωτογραφίες, λίστες ενδιαφερόντων, πληροφορίες επικοινωνίας και άλλα. Οι χρήστες μπορούν να επικοινωνήσουν με φίλους τους και άλλους χρήστες μέσω δημόσιων ή και προσωπικών μηνυμάτων ενώ είναι διαθέσιμη και δυνατότητα για άμεση συνομιλία (chat). Μπορούν ακόμα να συμμετέχουν σε ομάδες κοινών ενδιαφερόντων ή σελίδες προτίμησης (like pages). Όσον αφορά την προστασία της ιδιωτικότητας το Facebook επιτρέπει στο χρήστη να ρυθμίσει το επίπεδο ορατότητας του περιεχομένου σε τρίτους .

Στον ιστό του Facebook κάθε οντότητα, δηλαδή, κάθε ενεργό και δυναμικό στοιχείο που παρουσιάζεται στην ιστοσελίδα της πλατφόρμας αποτελεί ένα αντικείμενο που αλληλεπιδρά με τα υπόλοιπα. Έτσι κάθε χρήστης μπορεί να συνδεθεί με κάποιον άλλο με το να γίνει «φίλος» μαζί του ώστε να μπορεί να μοιράζεται το περιεχόμενο που δημοσιεύει στην πλατφόρμα. Για χρήστες που δεν έχουν την παραπάνω σχέση, το δημόσιο προφίλ και

ορισμένες πρόσθετες επιλογές που ρυθμίζουν την ορατότητα του περιεχομένου των διαφόρων αντικειμένων του χρήστη (φωτογραφικό άλμπουμ, πληροφορίες χρήστη, δημοσιεύσεις) φροντίζουν για την διασφάλιση της επιθυμητής ιδιωτικότητας. Τα ποικίλα αντικείμενα που συσχετίζονται με ένα χρήστη αποτελούν συνδετικούς κρίκους μεταξύ χρηστών καθώς λειτουργούν σαν αφορμές για αλληλεπίδραση μεταξύ τους παρέχοντας τελικά ένα πραγματικά πλούσιο σε περιεχόμενο κοινωνικό δίκτυο. Χαρακτηριστικά παραδείγματα είναι : ο χώρος σε κάθε σελίδα του προφίλ χρήστη που επιτρέπει την δημοσίευση μηνυμάτων ορατών στους φίλους (wall), τα συμβολικά μηνύματα απλής ειδοποίησης σε άλλους χρήστες (pokes,σαν εικονική «αφύπνιση» χρήστη), οι φωτογραφίες οι οποίες μπορεί να οργανωθούν σε άλμπουμ (photos), οι εκδηλώσεις προτίμησης (likes) και η κατάσταση του χρήστη (status) [7] .

Όπως και πολλά άλλα μέσα κοινωνικής δικτύωσης το Facebook διαθέτει τη δυνατότητα διασύνδεσης με άλλες ιστοσελίδες μέσω μίας υπερ-σύνδεσης με τη σήμανση "share" που οδηγεί στην ιστοσελίδα του και ο χρήστης προσθέτει στο προφίλ του περιεχόμενο που σχετίζεται με τη σελίδα στην οποία επέλεξε τη δυνατότητα διαμοιρασμού. Εκτός όμως από την κοινή μορφή του διαμοιρασμού, το Facebook δημιούργησε την έννοια της εκδήλωσης προτίμησης σε μία οποιαδήποτε σελίδα του παγκόσμιου ιστού η οποία υλοποιείται με μία υπερ-σύνδεση στο σημείο προτίμησης η οποία απεικονίζεται στο αντικείμενο "like" του ιστού του Facebook και εμφανίζεται σαν μία ένδειξη στο περιεχόμενο του προφίλ χρήστη. Με αυτόν τον τρόπο γίνεται πολύ εύκολη η ανταλλαγή και ο διαμοιρασμός δεδομένων από τον υπόλοιπο παγκόσμιο ιστό προς το Facebook και ταυτόχρονα καθίσταται πολύ απλή και άμεση η διαδικασία παραγωγής περιεχομένου που υποδηλώνει προτιμήσεις και τάσεις από ένα σύνολο ανθρώπων ώστε να υποστεί την κατάλληλη επεξεργασία από ειδικούς για την εξαγωγή ωφέλιμων συμπερασμάτων. Παράλληλα, η έντονη παρουσία του συγκεκριμένου κοινωνικού μέσου άρχισε να γίνεται εμφανής από την υιοθέτηση σε ποικίλες ιστοσελίδες ενσωματωμένων τμημάτων κώδικα που αποτελούν επέκταση του Facebook και ονομάζονται Facebook plugins αλλά και της πρόσθετης επιλογής σύνδεσης με εισαγωγή διαπιστευτηρίων Facebook account σε αυτές (Facebook login). Τα παραπάνω είναι εμφανές ότι επιδεικνύουν με τον καλύτερο δυνατό τρόπο το βαθμό διείσδυσης ορισμένων μέσων στον παγκόσμιο ιστό όμως είναι σημαντικό να προσέξουμε ότι αυτές οι πτυχές τους ενεργοποιούν τους χρήστες να εισάγουν με ευκολία και αμεσότητα περιεχόμενο από τον υπόλοιπο ιστό στην πλατφόρμα. Το Νοέμβριο του 2010 το Facebook ανακοίνωσε την δημιουργία ενός πρόσθετου χαρακτηριστικού με την ονομασία Facebook Messages το οποίο θα παρείχε τη δυνατότητα ανταλλαγής γραπτών μηνυμάτων, ανταλλαγής μηνυμάτων πραγματικού χρόνου και e-mail. Ο κάθε χρήστης μπορούσε να λάβει στον προσωπικό του φάκελο μηνυμάτων μήνυμα το οποίο είχε αποσταλεί στη διεύθυνση <user>@facebook.com αποκτώντας έτσι υπόσταση mail

account μέσω του facebook [7]. Λίγο αργότερα έγινε διαθέσιμη και η υπηρεσία ανταλλαγής μηνυμάτων σε πραγματικό χρόνο μεταξύ χρηστών του Facebook (Facebook chat) μέσα από ένα πελάτη στον περιηγητή (browser client) διαθέτοντας όλα τα απαραίτητα χαρακτηριστικά που αφορούν την διαθεσιμότητα των χρηστών και την καθυστερημένη παράδοση μηνυμάτων στην περίπτωση που ο χρήστης είναι εκτός σύνδεσης. Κατ'αυτόν τον τρόπο το Facebook δημιούργησε ένα δικό του σύστημα εξυπηρετητή-πελάτη για ανταλλαγή μηνυμάτων σε πραγματικό χρόνο παρόμοιο με διάφορες άλλες ήδη υπάρχουσες υλοποιήσεις. Εφαρμογές με πολλαπλή πρόσβαση σε διαφορετικά συστήματα instant messaging (IM-στιγμιαίων μηνυμάτων) προσέθεσαν σαν επιλογή την προσθήκη λογαριασμού Facebook chat κάνοντας χρήση των διαπιστευτηρίων χρήστη από το Facebook . Αντίστοιχες εφαρμογές που τρέχουν σε λογισμικό εξελιγμένων κινητών τηλεφώνων έχουν σήμερα διαδεδομένη χρήση καθώς αποτελούν τον πιο άμεσο, αξιόπιστο και εύκολο τρόπο για άμεση καθημερινή επικοινωνία.

Το ένα από τα πιο διαδεδομένα μέσα το οποίο και αναφέρθηκε παραπάνω αποτελεί έναν ιστό από αντικείμενα, δηλαδή, τα διάφορα χαρακτηριστικά και δυνατότητες που το απαρτίζουν, στα οποία ένας οποιοσδήποτε χρήστης μπορεί να έχει πρόσβαση μέσα από ένα web API. Το web API αποτελεί τη διαπροσωπεία μέσω της οποίας μπορεί κανείς προγραμματιστικά (και όχι μόνο) να ζητήσει από τον πυρήνα του κάθε συστήματος την αποστολή αναπαραστάσεων όλων των αντικειμένων του . Πρακτικά, αυτό σημαίνει ότι μία εφαρμογή, η οποία μπορεί να τρέχει σε διάφορες πλατφόρμες λογισμικού ηλεκτρονικών υπολογιστών ή κινητών, αποκτά πρόσβαση (μέσω μηχανισμών ασφαλείας) πραγματοποιώντας αιτήματα προς το web API του μέσου και λαμβάνοντας δεδομένα από αυτό αξιοποιώντας την αρχιτεκτονική υπηρεσιών ιστού REST (REpresentational State Transfer). Οι δυνατότητες που παρέχει η ύπαρξη ενός web API το οποίο είναι διαθέσιμο και ανοικτό σε όλους αποτελούν πρόκληση για οποιονδήποτε επιθυμεί να αξιοποιήσει την δύναμη των παραπάνω μέσων και συνεισφέρουν σημαντικά στην διεύρυνση της κοινότητας των χρήστων και όσων ασχολούνται με την ανάπτυξη ανάλογων εφαρμογών.

2.2 Επικοινωνία σε Πραγματικό Χρόνο

Η έννοια της επικοινωνίας σε σχεδόν πραγματικό χρόνο μέσω του διαδικτύου απέκτησε μορφή υλοποίησης σύγχρονη και συμβατή με τον παγκόσμιο ιστό στα μέσα της δεκαετίας του 1990 με πιο γνωστές εφαρμογές το PowWow, το ICQ και το AOL Instant Messenger τα οποία αποτέλεσαν τις πρώτες μαζικές λύσεις τέτοιου είδους συστημάτων ανταλλαγής άμεσων μηνυμάτων (συστήματα instant-messaging - IM) και γνώρισαν μεγάλη επιτυχία έχοντας

εκατομμύρια εγγεγραμμένους χρήστες. Αργότερα ακολούθησαν και άλλες εταιρίες οι οποίες δημιούργησαν δικά τους πρωτόκολλα επικοινωνίας πραγματικού χρόνου και δικό τους λογισμικό πελάτη (MSN Messenger, Yahoo! Messenger) μέσα από το οποίο αποκτούσε πρόσβαση ο χρήστης στο συγκεκριμένο σύστημα. Ωστόσο, οι χρήστες που επιθυμούσαν να χρησιμοποιήσουν παραπάνω από μία πλατφόρμες ήταν υποχρεωμένοι να συνδέονται με περισσότερα του ενός προγράμματα πελάτη [8]. Το 2000 εμφανίστηκε ένα νέο ανοικτό πρωτόκολλο με την ονομασία Jabber με σκοπό να εξαλείψει το προηγούμενο μειονέκτημα. Το γεγονός ότι το Jabber είχε πρότυπα ανοικτά και διαθέσιμα προς όλους αμέσως έδινε τη δυνατότητα να αναπτυχθούν διαφορετικά προγράμματα πελάτη που όμως να υπακούν σε συγκεκριμένη αρχιτεκτονική (πελάτη-εξυπηρετητή) είτε για τη δημιουργία τοπικών δικτύων ανταλλαγής μηνυμάτων ή δικτύων ευρύτερου μεγέθους τα οποία πλέον θα μπορούσαν να συνδεθούν μεταξύ τους αξιοποιώντας το χαρακτηριστικό της διασύνδεσης εξυπηρετητών. Αργότερα μετονομάστηκε σε eXtensible Messaging and Presence Protocol (XMPP) και προσέφερε επιπρόσθετα τη δυνατότητα διασύνδεσης και με άλλα δίκτυα IM όπως το Google Talk. Παράλληλα με το XMPP εμφανίστηκαν προγράμματα πελάτη IM τα οποία περιλάμβαναν εγκατεστημένες βιβλιοθήκες για επικοινωνία με καθένα από τα γνωστά δίκτυα IM προκειμένου να μπορεί ο χρήστης να έχει πρόσβαση σε αυτά από ένα μόνο πρόγραμμα πελάτη με κοινό γραφικό interface (διαπροσωπεία). Στόχος της δημιουργίας του XMPP ήταν να δημιουργηθεί μία ανεπτυγμένη κοινότητα που θα έχει επαφή και ενασχόληση με ένα ενιαίο πρωτόκολλο όσον αφορά την επικοινωνία πραγματικού χρόνου τόσο ώστε κάθε χρήστης IM να αποτελεί οντότητα η οποία ανήκει σε ένα ευρύ και ανοικτό δίκτυο XMPP αλλά και να αξιοποιηθούν οι μεγάλες δυνατότητες που μπορεί η συγκεκριμένη τεχνολογία να προσφέρει στη βάση της δομημένης επικοινωνίας πραγματικού χρόνου σε συστήματα ειδοποιήσεων, διαχείρισης ασύγχρονης μεταφοράς πληροφορίας αλλά και οποιασδήποτε προσαρμοσμένης ειδικής χρήσης μέσω των επεκτάσεων που μπορεί να δεχθεί.

Σήμερα, η σημαντική αύξηση της συχνότητας χρήσης των κοινωνικών μέσων δικτύωσης και η εκτεταμένη πρόσβαση των χρηστών στον παγκόσμιο ιστό από διαφορετικά σημεία (υπολογιστής σε γραφείο/σπίτι, φορητός υπολογιστής, κινητό τηλέφωνο) έχει δημιουργήσει την ανάγκη για συνεχή διαθεσιμότητα στο δίκτυο (online χρήστης) προκειμένου να υπάρχει διάλογος μεταξύ χρηστών αλλά και επαφή με τα κοινωνικά μέσα σε πραγματικό χρόνο. Ειδικότερα, με την ανάπτυξη ολοκληρωμένων λύσεων υλισμικού και λογισμικού σε κινητά τηλέφωνα με δυνατότητες κοντά σε αυτές ενός προσωπικού υπολογιστή (γνωστά ως smartphones) οι χρήστες απέκτησαν πρόσβαση σε εφαρμογές κοινωνικών μέσων δικτύωσης και άρχισαν να χρησιμοποιούν αυτά σαν εργαλεία για χρονικά άμεση αλληλεπίδραση με το κοινωνικό τους δίκτυο χωρίς τον περιορισμό της διαθεσιμότητας στο δίκτυο αναλόγως της

θέσης τους. Έτσι, ταυτόχρονα δημιουργήθηκε η ανάγκη για επικοινωνία πραγματικού χρόνου μέσω γραπτών μηνυμάτων από τις φορητές συσκευές αξιοποιώντας την σύνδεση δεδομένων/internet που παρέχει το κάθε δίκτυο κινητής τηλεφωνίας ή την ασύρματη δικτύωση σε ένα τοπικό δίκτυο που έχει πρόσβαση στον παγκόσμιο ιστό. Δημιουργήθηκαν έτσι εφαρμογές που υποστηρίζουν πληθώρα διαφορετικών πρωτοκόλλων IM και μπορούν να εγκατασταθούν σε διαφορετικές πλατφόρμες smartphones όπως android, blackberry OS, iOS , WP7. Οι περισσότερες από αυτές υποστηρίζουν και το πρωτόκολλο XMPP δίνοντας δυνατότητα πρόσβασης σε οποιοδήποτε τοπικό (προσωπικό/εταιρικό) ή ευρύτερο (jabber servers) δίκτυο .

2.3 Αρχιτεκτονικές στο Διαδίκτυο

2.3.1 Representational State Transfer

Το Representational State Tranfer (REST) [9] είναι ένα αρχιτεκτονικό στυλ για κατανεμημένα συστήματα υπερμέσων (hypermedia), όπως ο Παγκόσμιος Ιστός. Οι όροι «Representational State Transfer» και «REST» παρουσιάστηκαν το 2000 στη διδακτορική διατριβή του Roy Fielding [1].

Η αρχιτεκτονική REST αποτελείται από εξυπηρετητές (servers) και πελάτες(clients). Οι clients εκκινούν την αποστολή αιτημάτων στους servers οι οποίοι με τη σειρά τους τα επεξεργάζονται προκειμένου να επιστρέψουν κατάλληλες απαντήσεις. Κάθε αίτημα ή μήνυμα προς τους εξυπηρετητές αποστέλλεται σε συγκεκριμένο αντικείμενο της εφαρμογής το οποίο ονομάζεται πόρος (resource). Κάθε πόρος έχει μοναδική διεύθυνση στην οποία αντιστοιχίζεται το κάθε αίτημα. Οι απαντήσεις εμφανίζουν στον πελάτη αναπαραστάσεις συγκεκριμένων πόρων για τη δεδομένη χρονική στιγμή του αιτήματος, αποτελούν δηλαδή ένα στιγμιότυπό τους. Η ονομασία της αρχιτεκτονικής REST προέρχεται από την έννοια της μεταφοράς κατάστασης στον πελάτη. Με κάθε νέο αίτημα επιστρέφεται μία ανάλογη απόκριση στον πελάτη με τη μορφή αναπαράστασης με αποτέλεσμα να μεταβαίνει αυτός σε μία νέα κατάσταση .

Βασικές αρχές του REST είναι :

- Μία υλοποίηση REST αποτελείται από συστατικά δικτύου (components) τα οποία διαιρούνται σε πόρους
- Κάθε πόρος διευθυνσιοδοτείται μοναδικά με ένα καθολικό ταυτοποιητή (URI) (βλ. παράγραφος 5.2.1.1 [1])

- Η μεταφορά κατάστασης μεταξύ του πελάτη και του πόρου πραγματοποιείται μέσα από μία ομοιόμορφη διεπαφή (uniform interface) (βλ. παράγραφος 5.1.5 [1]) , η οποία χαρακτηρίζεται από:
    - Ένα καθορισμένο σύνολο αυστηρά ορισμένων λειτουργιών
    - Ένα σύνολο από τύπους περιεχομένου

    Τα δύο παραπάνω συστατικά αναδεικνύουν τη σημασιολογία της επικοινωνίας καθώς επίσης αποτελούν το πρότυπο για την ανταλλαγή πληροφορίας.

Σύμφωνα με τις παραγράφους 5.1.2,5.1.3,5.1.4,5.1.6 [1] η εφαρμογή της αρχιτεκτονικής γίνεται σε πρωτόκολλα τα οποία :

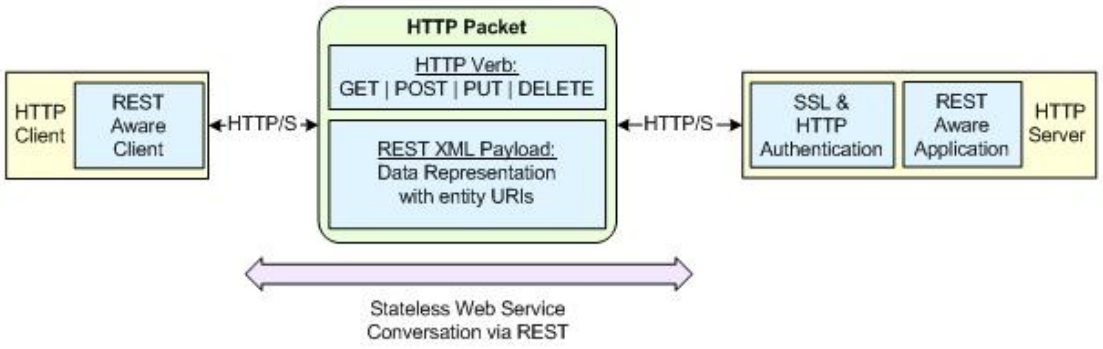
- είναι Πελάτη-Εξυπηρέτη (Client-server)
- είναι άνευ κατάστασης (Stateless)
- δίνουν δυνατότητα αποθήκευσης περιεχομένου σε κρυφές μνήμες (Cacheable)
- υποστηρίζουν διαστρωμάτωση (Layered)

Η δυνατότητα αποθήκευσης σε κρυφές μνήμες παρέχεται μέσα από το ίδιο το μήνυμα απόκρισης που αποστέλλει ο εξυπηρετητής στο περιεχόμενο του οποίου αναφέρεται εάν θα πρέπει να αποθηκευτεί προσωρινά.

Αναπτύχθηκε παράλληλα με το πρωτόκολλο HTTP/1.1 ενώ η ύπαρξη του παγκόσμιου ιστού αποτελεί την μεγαλύτερη υλοποίηση συστήματος που ακολουθεί τις αρχές της χωρίς ωστόσο να αποτελεί το μοναδικό πεδίο εφαρμογής .

Στο πεδίο αυτής της διπλωματικής μας αφορά άμεσα η εφαρμογή της αρχιτεκτονικής αυτής στον παγκόσμιο ιστό. Σε αυτή την εφαρμογή, το πρωτόκολλο HTTP αποτελεί τη διεπαφή μέσα από την οποία ο πελάτης επικοινωνεί με τους πόρους και δέχεται τις αναπαραστάσεις τους. Οι πόροι είναι σελίδες που φιλοξενούνται στον εξυπηρετητή, στατικές ή δυναμικές (κώδικας on-demand) και η αναφορά σε αυτούς γίνεται με ένα καθολικό ταυτοποιητή (URI) που στη συγκεκριμένη περίπτωση είναι μία διεύθυνση URL. Κάθε αίτημα στο συγκεκριμένο URL συνεπάγεται την διαδικασία επεξεργασίας στον εξυπηρετητή και την αποστολή μίας απόκρισης στον πελάτη .

Εικόνα 1 Σχήμα αρχιτεκτονικής REST [33]



Ακολουθώντας τις αρχές και τους κανόνες του REST είναι δυνατή η δημιουργία μίας υπηρεσίας ιστού. Μία τέτοια αντιπροσωπευτική υλοποίηση υπακούει στις εξής αρχές [4]:

- Αυστηρή και σαφής χρήση του πρωτοκόλλου HTTP

  Απαιτείται να γίνεται σαφής και σύμφωνη με το πρωτόκολλο χρήση των μεθόδων του. Αυτό σημαίνει ότι πρέπει να υπάρχει αυστηρή αντιστοίχιση ανάμεσα στις λειτουργίες δημιουργίας, ανάγνωσης, ενήμερωσης και διαγραφής (create-read-update-delete CRUD operations) και στις HTTP μεθόδους. Έτσι, σύμφωνα με αυτήν την αντιστοίχιση με την POST μέθοδο δημιουργούμε έναν πόρο στον εξυπηρετητή, με την GET ανακαλούμε ένα πόρο, με την PUT ενημερώνουμε την κατάσταση ενός πόρου και με την DELETE διαγράφουμε ένα πόρο.

- Η πλευρά του εξυπηρετητή χαρακτηρίζεται από απουσία κατάστασης (stateless)

  Η δομή ενός δικτύου που υποστηρίζει την REST αρχιτεκτονική πρέπει να σχεδιάζεται με παράμετρο τις έντονα αυξανόμενες απαιτήσεις σε ζήτηση και μέγεθος. Κατ'αυτόν τον τρόπο προκειμένου να υποστηρίξει την υψηλή κλιμάκωση περιλαμβάνει πληθώρα συστατικών δικτύου (ενδιάμεσους εξυπηρετητές, μεσολαβητές και πύλες) ικανών να διαχειριστούν μεγάλο αριθμό αιτημάτων από χρήστες. Αυτό προϋποθέτει ότι κάθε αίτημα πρέπει να είναι ανεξάρτητο και ολοκληρωμένο, δηλαδή, να περιέχει όλες τις παραμέτρους, τα δεδομένα και το περιεχόμενο που απαιτείται προκειμένου να παραχθεί μία απάντηση σε αυτό. Ο εξυπηρετητής δεν χρειάζεται να ανακτά ή να συγχρονίζει δεδομένα που αντιστοιχούν στην κάθε συνδερία επικοινωνίας με τον χρήστη με αποτέλεσμα να μειώνεται ο φόρτος του και να βελτιώνεται η κλιμάκωση. Επίσης, κάθε αίτημα μεταξύ ενός πελάτη και του εξυπηρετητή προορισμού μπορεί να προωθηθεί από πληθώρα συστατικών δικτύου όπως αυτά που προαναφέρθηκαν χωρίς όμως να υπάρχει ορατότητα στο περιεχόμενο του από αυτά. Κάθε οντότητα μπορεί να έχει πρόσβαση

μόνο στο μήνυμα που αποστέλλει ή λαμβάνει. Η παραπάνω πτυχή της αρχιτεκτονικής αναφέρεται ως ιδιότητα «διαστρωμάτωσης»(Layering).

- Τα URIs της υπηρεσίας πρέπει να έχουν μορφή που να υποδηλώνει το νόημα και να ακολουθεί το πρότυπο μίας δομής καταλόγου.

  Κάθε πόρος της υπηρεσίας ιστού πρέπει να έχει URI που να δείχνει έμμεσα την έννοια ύπαρξης και λειτουργίας του. Για να είναι ακόμα πιο βοηθητικό και ξεκάθαρο στη χρήση είναι σκόπιμο η διεύθυνση (url) να είναι σε μορφή που να δηλώνει τη διαδρομή σε ένα κατάλογο με τις υποενότητές του όπως η παρακάτω:

  http://www.myservice.org/discussion/2008/12/10/{topic}

- Μεταφορά αναπαραστάσεων των πόρων σε μορφή internet media types όπως XML ή JSON.

Όπως αναφέρθηκε και παραπάνω η απόκριση που δίνεται από τον εξυπηρετητή στον πελάτη είναι η αναπαράσταση ενός πόρου. Αυτό σημαίνει ότι το αίτημα πρέπει να επιστρέφει ένα στιγμιότυπο της κατάστασης του πόρου και των χαρακτηριστικών του τη χρονική στιγμή που πραγματοποιείται το αίτημα από τον πελάτη. Το στιγμιότυπο αυτό μπορεί να έχει απλά τη μορφή μίας εγγραφής σε μία βάση δεδομένων και να επιστρέφεται αυτή ως XML αρχείο κατά το αίτημα της ή να είναι αναπαράσταση ενός αντικειμένου σε ένα μοντέλο δεδομένων που χρησιμοποιείται από το σύστημα. Στην τελευταία περίπτωση είναι φυσικό ότι τα αντικείμενα που απαρτίζουν το μοντέλο συνδέονται μεταξύ τους με σχέσεις αλληλεπίδρασης και είναι σκόπιμο αυτές οι αλληλεπιδράσεις να απεικονίζονται στο περιεχόμενο μίας αναπαράστασης δίνοντας τη δυνατότητα να παρέχονται τα URIs των συσχετιζόμενων αντικειμένων. Παράδειγμα τέτοιας αναπαράστασης σε μορφή κειμένου XML αποτελεί η παρακάτω:

Πίνακας 1 Παράδειγμα αναπαράστασης αντικειμένου σε XML

```
<?xml version="1.0"?>
<discussion date="{date}" topic="{topic}">
  <comment>{comment}</comment>
  <replies>
    <reply from=joe@mail.com href="/discussion/topics/{topic}/joe"/>
    <reply
from="bob@mail.com" href="/discussion/topics/{topic}/bob"/>
  </replies>
</discussion>
```

Σύμφωνα με την παράγραφο 5.3.1 [1], ο διαχωρισμός των ασχολιών του πελάτη και του εξυπηρετητή απλοποιεί την υλοποίηση των συστατικών, μειώνει την πολυπλοκότητα της

σημασιολογίας των διασυνδετών, βελτιώνει την αποτελεσματικότητα των ρυθμίσεων απόδοσης και αυξάνει το επίπεδο κλιμάκωσης των συστατικών εξυπηρετητή.

2.3.2 Simple Object Access protocol (SOAP)

Το SOAP, αρχικά ορισμένο ως Simple Object Access Protocol, είναι μία προδιαγραφή πρωτοκόλλου για ανταλλαγή δομημένης πληροφορίας στην εφαρμογή των υπηρεσιών ιστού σε δίκτυα υπολογιστών. Βασίζεται στο πρότυπο XML (extensible markup language) για την προδιαγραφή μηνύματος του και συνήθως είναι εξαρτώμενο από άλλα πρωτόκολλα επιπέδου εφαρμογής με κύρια τα RPC (remote procedure call) και HTTP, για διαπραγμάτευση και μετάδοση μηνύματος[10].

Η δημιουργία του SOAP προέκυψε από την ανάγκη για δημιουργία ενός αξιόπιστου τρόπου επικοινωνίας μεταξύ εφαρμογών σε ένα ευρύτερο δίκτυο που να ξεπερνά όμως τους φραγμούς που θέτουν η ασυμβατότητα μεταξύ διαφορετικών λογισμικών και τα προγράμματα ασφαλείας στην διαδικτυακή επικοινωνία. Οι εφαρμογές μπορούν να επικοινωνήσουν μεταξύ τους μέσω RPC – remote procedure calls μεταξύ αντικειμένων όπως συμβαίνει με τα πρότυπα DCOM και CORBA έχοντας τους παραπάνω περιορισμούς. Το πρωτόκολλο HTTP αποτελεί μία ιδανική βάση για την επικοινωνία εφαρμογών καθώς είναι φιλικό προς τα προγράμματα ασφαλείας και υποστηρίζεται από όλους του φυλλομετρητές και εξυπηρετητές του παγκόσμιου ιστού. Το SOAP αποτελεί ένα πρότυπο επικοινωνίας μέσω HTTP που διασφαλίζει την βασική απαίτηση για εύκολη ανταλλαγή πληροφορίας μεταξύ εφαρμογών ιστού. Το πρωτόκολλο SOAP προσφέρει την δυνατότητα να υποστηρίζει διαφορετικά πρωτόκολλα μεταφοράς όπως το SMTP, JMS και το HTTP. Ωστόσο, πιο διαδεδομένη είναι η χρήση SOAP μέσω του πρωτοκόλλου HTTP καθώς είναι συμβατή με την σημερινή υποδομή του παγκόσμιου ιστού[12].

Το SOAP είναι ικανό να σχηματίσει το επίπεδο – θεμέλιο μίας στοίβας πρωτοκόλλου υπηρεσίας ιστού παρέχοντας ένα βασικό περιβάλλον-πλαίσιο επικοινωνίας μέσω μηνυμάτων πάνω στο οποίο μπορούν να χτιστούν υπηρεσίες ιστού.

Αυτό το βασισμένο σε XML πρωτόκολλο αποτελείται από τρία μέρη : τον φάκελο , ο οποίος καθορίζει το περιεχόμενο του μηνύματος και τον τρόπο χειρισμού του, ένα σύνολο από κανόνες κωδικοποίησης για την διατύπωση στιγμιοτύπων τύπων δεδομένων (καθοριζόμενων από την εφαρμογή) και μία σύμβαση για αναπαράσταση κλήσεων διαδικασιών και απαντήσεων. [10]

Ένα παράδειγμα μηνύματος SOAP είναι το παρακάτω:

Πίνακας 2 Παράδειγμα μηνύματος SOAP

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Header>
 <m:Transxmlns:m="http://www.w3schools.com/transaction/"
 soap:mustUnderstand="1">234
 </m:Trans>
</soap:Header>

<soap:Body>
 <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
  <m:Item>Apples</m:Item> </m:GetPrice>
</soap:Body>
</soap:Envelope>
```

Το στοιχείο <soap:Envelope> είναι εκείνο που καθορίζει ότι το συγκεκριμένο κείμενο XML είναι μήνυμα SOAP. Το xml namespace soap είναι υποχρεωτικό και πρέπει να έχει πάντα την συγκεκριμένη τιμή ενώ το encodingstyle καθορίζει τους τύπους δεδομένων του αρχείου και δεν έχει προεπιλεγμένη τιμή. [14]

Το στοιχείο <soap:Header> επικεφαλίδας είναι προαιρετικό και παρέχει πληροφορίες σχετικές με την εφαρμογή όπως πιστοποίηση ή διαδικασία πληρωμής. Το SOAP ορίζει τρία χαρακτηριστικά για την επικεφαλίδα τα οποία ρυθμίζουν τον τρόπο επεξεργασίας του μηνύματος από τον παραλήπτη : mustUnderstand (ορίζει εάν ο παραλήπτης πρέπει να αναγνωρίσει το συγκεκριμένο στοιχείο-παιδί της επικεφαλίδας), actor (ορίζει αν το συγκεκριμένο στοιχείο-παιδί απευθύνεται στον συγκεκριμένο παραλήπτη) και encodingStyle. [15]

Το στοιχείο <soap:Body> περιέχει το βασικό περιεχόμενο του μηνύματος το οποίο προορίζεται για τον απόλυτο παραλήπτη του. Τα άμεσα στοιχεία-παιδιά αυτού του στοιχείου μπορούν να επιδεχθούν xml namespaces καθοριζόμενα από την εφαρμογή (άσχετα από το SOAP namespace) .[17]

Τα μηνύματα SOAP επιδέχονται ακόμα τη μεταφορά στοιχείων soap-fault για διαχείριση σφαλμάτων και κατάστασης μηνυμάτων.

Ως παράδειγμα του τρόπου κλήσης διαδικασιών SOAP,ένα μήνυμα μπορεί να αποσταλλεί σε μία ιστοσελίδα που υποστηρίζει υπηρεσίες ιστού όπως μία βάση δεδομένων ανάπτυξης ακινήτων, με τις απαιτούμενες παραμέτρους προς αναζήτηση. Στη συνέχεια η ιστοσελίδα θα επιστρέψει ένα έγγραφο στο πρότυπο του XML με τα κατάλληλα δεδομένα ως αποτέλεσμα (π.χ. τιμές,τοποθεσία κ.λ.π.). Έχοντας τα δεδομένα να επιστρέφονται σε μία τυποποιημένη, αναγνώσιμη από μηχανή μορφή μπορούν έτσι να ενσωματωθούν άμεσα σε οποιαδήποτε ιστοσελίδα ή εφαρμογή. Η προδιαγραφή του SOAP ορίζει το πλαίσιο ανταλλαγής μηνυμάτων το οποίο αποτελείται από τις παρακάτω συνιστώσες:[10]

- Το μοντέλο επεξεργασίας SOAP το οποίο ορίζει τους κανόνες που διέπουν την επεξεργασία ενός μηνύματος SOAP
- Το μοντέλο επεκτασιμότητας SOAP το οποίο ορίζει τις ιδέες που υλοποιούν τα SOAP features και τα SOAP modules
- Το πλαίσιο που καθορίζει τη διαπραγμάτευση και την προσαρμογή στο υποκείμενο πρωτόκολλο που χρησιμοποιείται για τη μεταφορά πληροφορίας από ένα άκρο σε ένα άλλο
- Το μοντέλο δομής ενός SOAP μηνύματος

Το μοντέλο επεξεργασίας SOAP περιγράφει ένα κατανεμημένο μοντέλο επεξεργασίας που απαρτίζεται από τους συμμετέχοντες του, δηλαδή, τους κόμβους SOAP και τους τρόπους με τους οποίους ένας δέκτης SOAP επεξεργάζεται ένα μήνυμα. Ως κόμβοι SOAP ορίζονται οι παρακάτω:[10]

- Αποστολέας SOAP : ένας κόμβος SOAP ο οποίος μεταδίδει ένα μήνυμα.
- Παραλήπτης SOAP : ένας κόμβος SOAP ο οποίος λαμβάνει ένα μήνυμα.
- Διαδρομή μηνύματος SOAP : ένα σύνολο κόμβων SOAP το οποίο διατρέχει ένα συγκεκριμένο μήνυμα.
- Ενδιάμεσος κόμβος SOAP : ένας κόμβος SOAP ο οποίος αποτελεί κόμβο αποστολέα και κόμβο παραλήπτη. Επεξεργάζεται την επικεφαλίδα SOAP τα οποία απευθύνονται σε εκείνον και λειτουργεί ως προωθητής του μηνύματος στον απόλυτο παραλήπτη του.
- Απόλυτος παραλήπτης SOAP : ένας κόμβος SOAP ο οποίος αποτελεί τον τελικό προορισμό ενός μηνύματος SOAP. Είναι υπεύθυνος για την επεξεργασία των περιεχομένων του σώματος του μηνύματος (SOAP body) αλλά και όποιων τμημάτων της επικεφαλίδας απευθύνονται σε αυτόν. Σε ορισμένες περιπτώσεις ένα μήνυμα SOAP μπορεί να μην φτάσει σε έναν απόλυτο παραλήπτη για παράδειγμα εξαιτίας ενός προβλήματος σε έναν ενδιάμεσο κόμβο. Ένας απόλυτος παραλήπτης δεν δύναται να αποτελεί συγχρόνως ενδιάμεσο κόμβο για το ίδιο μήνυμα SOAP .

Συμπεραίνουμε ότι το πρωτόκολλο SOAP αποτελεί σε απλή περιγραφή ένα πρότυπο για ανταλλαγή μηνυμάτων XML μεταξύ κόμβων. Η απλότητα της μορφής του μηνύματος λόγω της χρήσης του προτύπου XML και η δυνατότητα χρήσης διαφορετικών xml namespaces στο στοιχείο <soap:body> και ακολούθως στα στοιχεία παιδιά αυτού επιτρέπει την αναφορά σε προκαθορισμένες μεθόδους και μεταβλητές ενός συστήματος προκειμένου να πραγματοποιούνται αιτήματα και αποκρίσεις ώστε να εξυπηρετηθούν οι λειτουργίες μίας υπηρεσίας ιστού. Τα παρακάτω μηνύματα απεικονίζουν την εφαρμογή της παραπάνω δυνατότητας μέσα από μία αλληλεπίδραση μεταξύ δύο κόμβων (αίτημα χρήστη – απόκριση υπηρεσίας) :[16]

Πίνακας 3 Αίτημα SOAP

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

Πίνακας 4 Απόκριση SOAP

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
  </m:GetStockPriceResponse>
</soap:Body>

</soap:Envelope>
```

Στο παραπάνω παράδειγμα ζητείται με τη μέθοδο GetStockPrice η τιμή μίας μετοχής με παράμετρο το όνομά της βάσει ενός xml namespace (="http://www.example.org/stock") η οποία στη συνέχεια επιστρέφεται με το ίδιο namespace και τη μέθοδο GetStockPriceResponse. Παρατηρούμε ότι το περιεχόμενο του στοιχείου <soap:Body> είναι αυτό που καθορίζει τις ενέργειες που πρέπει να εκτελέσει ο κόμβος παραλήπτης .

Το SOAP αποτελεί μία ευνοϊκή λύση για ανταλλαγή μηνυμάτων μέσω HTTP μεταξύ εφαρμογών ιστού καθώς όπως αναφέραμε προσφέρει ευελιξία κατά τη μεταφορά μέσα από προγράμματα ασφαλείας(firewalls) και μεσολαβητές (proxies) όμως προκειμένου να παραμείνει η επικοινωνία γρήγορη τίθεται ο φραγμός στο μέγεθος του μηνύματος που μεταδίδεται. Ακόμα, συμπεραίνουμε ότι πρόκειται για μία τεχνολογία όπου οι ρόλοι μεταξύ των αλληλεπιδρώντων πλευρών είναι αυστηρά καθορισμένοι καθώς υποστηρίζει τη λειτουργία μίας υπηρεσίας την οποία μόνο ο χρήστης-πελάτης (δηλαδή, η μία πλευρά) μπορεί να εκμεταλλευτεί. Έτσι για την υλοποίηση ενός συστήματος ειδοποιήσεων ή ανανέωσης πληροφορίας στην πλευρά του χρήστη απαιτείται η διαδικασία των επίμονων αιτημάτων (Polling) από την πλευρά του προκειμένου να αποκτήσει νέα πληροφορία από την υπηρεσία .

2.4 XMPP πρωτόκολλο

Η ονομασία XMPP αναφέρεται σε μία τεχνολογία η οποία αρχικά δημιουργήθηκε με σκοπό την υλοποίηση της επικοινωνίας μεταξύ χρηστών σε πραγματικό χρόνο μέσω γραπτών μηνυμάτων. Το πρωτόκολλο XMPP (eXtensible Messaging & Presence Protocol) επινοήθηκε στην πρώιμη μορφή του το 1998 από τον Jeremie Miller ο οποίος δημιούργησε και διέθεσε στην ανοικτή κοινότητα ένα server ανοιχτού κώδικα ονομαζόμενο jabberd προκειμένου να επιτραπεί η δυνατότητα στους χρήστες του παγκόσμιου ιστού να επικοινωνούν μεταξύ τους σε πραγματικό χρόνο χρησιμοποιώντας ένα κοινό μέσο εκτός των κλειστών IM συστημάτων ανταλλαγής μηνυμάτων της εποχής (ICQ, MSN, AOL). Λέγοντας κοινό μέσο εννοούμε ένα μέσο ελεύθερο προς όλους, το οποίο περιγράφεται από ενιαίο πρωτόκολλο και ελεύθερη διάθεση λογισμικού πελάτη/εξυπηρετητή καθώς και βιβλιοθήκες για ανάπτυξη των προηγουμένων. Στη συνέχεια, το 2002, το IETF (Internet Engineering Task Force) [25] συγκρότησε μια ομάδα ανάπτυξης του XMPP με σκοπό να διατυπώσει με επίσημη μορφή τα πρωτόκολλα πυρήνα σαν μία τεχνολογία “instant messaging and presence” του IETF.

Ειδικότερα, πρόκειται για μία τεχνολογία η οποία χρησιμοποιεί το πρότυπο open XML (eXtensible Markup Language) ως το κυρίαρχο φορμά για ανταλλαγή πληροφορίας και στην ουσία επιτρέπει τη μεταφορά μικρού μεγέθους αρχείων xml από μία οντότητα σε μία άλλη σχεδόν σε πραγματικό χρόνο. Αναπτύχθηκε σαν ένα open standard, δηλαδή, σαν ένα πρότυπο που οι προδιαγραφές του ορίζουν ανοιχτά πρωτόκολλα τα οποία χρησιμοποιούνται για επικοινωνία μεταξύ δικτυακών οντοτήτων. Όπως το HTTP και η HTML ορίζουν τα πρωτόκολλα και τους τύπους δεδομένων στα οποία βασίζεται ο παγκόσμιος ιστός έτσι το XMPP ορίζει τα πρωτόκολλα και τους τύπους δεδομένων που στηρίζουν τις αλληλεπιδράσεις πραγματικού χρόνου στο internet [2] . Ο πυρήνας της τεχνολογίας του XMPP απεικονίστηκε σε τέσσερις προδιαγραφές RFC (3920, 3921, 3922, 3923) από την ομάδα ανάπτυξης οι οποίες εγκρίθηκαν από την IETF το 2004. Το 2011 τα RFC-3920/3921 αντικαταστάθηκαν από τα RFC-6120/6121 ενώ το 6122 προδιέγραψε τον τύπο της XMPP διεύθυνσης. Ο σχεδιασμός του πυρήνα της τεχνολογίας του XMPP έγινε με κύριο γνώμονα και απαίτηση την επεκτασιμότητα με αποτέλεσμα μέχρι σήμερα να έχει αποτελέσει μία τεχνολογική υποδομή για την πραγματοποίηση όχι μόνο της γραπτής επικοινωνίας πραγματικού χρόνου αλλά της τηλεφωνίας ιστού (VoIP) καθώς επίσης και εφαρμογών και υπηρεσιών πραγματικού χρόνου.

2.4.1 Αρχιτεκτονική

Η αρχιτεκτονική του XMPP ακολουθεί το μοντέλο πελάτη-εξυπηρετητή. Κατ' αυτόν τον τρόπο η δημιουργία ενός συστήματος που υλοποιεί αυτήν την τεχνολογία σε ένα δίκτυο (τοπικό ή μή) συνίσταται στην ύπαρξη ενός ή περισσότερων εξυπηρετητών και στην δημιουργία ενός ή περισσότερων πελατών. Στην πιο απλή μορφή του λειτουργεί ως σύστημα ανταλλαγής μηνυμάτων πραγματικού χρόνου μεταξύ χρηστών εγγεγραμμένων σε ένα συγκεκριμένο εξυπηρετητή. Υπεύθυνος για τη διαχείριση της επικοινωνίας μεταξύ των πελατών είναι ο εξυπηρετητής μέσω του οποίου γίνεται η εκκίνηση και διατήρηση της σύ-νδεσης ανάμεσα σε εκείνον και τον πελάτη. Εκτός από την ανταλλαγή μηνυμάτων μεταξύ των χρηστών μεγάλη σημασία στην επικοινωνία τους έχει η διαθεσιμότητα στο δίκτυο (Network availability) κάθε οντότητας. Η συνεχής ενημέρωσή της υλοποιείται με την αποστολή των μηνυμάτων παρουσίας (presence) μεταξύ τους με σκοπό τον προσδιορισμό της κατάστασής τους. Τέτοιου είδους μηνύματα αποστέλλονται από τον χρήστη στον εξυπηρετητή ο οποίος με τη σειρά του παράγει και αποστέλλει ένα ανάλογο μήνυμα σε όσους χρήστες έχουν τον αποστολέα στους καταλόγους επαφών τους. Οι συγκεκριμένοι κατάλογοι διατηρούνται και ενημερώνονται συνεχώς από τον εξυπηρετητή. Είναι μία αποκεντρωμένη αρχιτεκτονική καθώς οι πελάτες είναι οντότητες υπεύθυνες μόνο για διαδικασίες όπως :

- η σύνταξη και αποστολή μηνυμάτων τα οποία θα περιγραφούν εκτενέστερα παρακάτω και μπορούν να έχουν τη μορφή ενός απλού μηνύματος κειμένου με παραλήπτη έναν οποιονδήποτε χρήστη (message) ή ενός μηνύματος παρουσίας ή ενός μηνύματος πληροφορίας-αιτήματος που προσφέρει υπηρεσίες από τον εξυπηρετητή στον πελάτη (iq)
- η λήψη μηνυμάτων από τον εξυπηρετητή που ανήκουν σε μία από τις προαναφερθείσες κατηγορίες

ενώ ο εξυπηρετητής έχει τις παρακάτω βασικές αρμοδιότητες :

- διαχείριση των λογαριασμών των χρηστών και έλεγχος για σύνδεση χρήστη πριν από κάθε εκκίνηση μετάδοσης πληροφορίας λήψη των τριών διαφορετικών ειδών μηνυμάτων από τους χρήστες
- Επεξεργασία και απόκριση στα μηνύματα των χρηστών ανάλογα με το είδος του εισερχόμενου αιτήματος. Αν πρόκειται για μήνυμα με περιεχόμενου κειμένου (payload) το προωθεί στον αντίστοιχο παραλήπτη που ορίζεται μέσα σε αυτό ενώ αν είναι μήνυμα παρουσίας ενημερώνει τους χρήστες που έχουν τον συγκεκριμένο χρήστη-αποστολέα στους καταλόγους τους με την κατάσταση στην οποία βρίσκεται .Τέλος, εάν πρόκειται για μήνυμα που έχει τη μορφή αιτήματος ή ρύθμισης κάποιας υπηρεσίας ο εξυπηρετητής επεξεργάζεται το αίτημα και επιστρέφει στον αποστολέα την κατάλληλη απάντηση .

Έτσι,η υποδομή του διαδικτύου σε ανταλλαγή άμεσων μηνυμάτων και παρουσίας συνίσταται στην ύπαρξη εκατοντάδων χιλιάδων εξυπηρετητών που τρέχουν λογισμικό όπως ο Openfire [38] και ejabberd και εκατομμυρίων πελατών-χρηστών οι οποίοι τρέχουν λογισμικό όπως τα Adium,Gajim και Pidgin χρησιμοποιώντας το πρωτόκολλο XMPP .

Οι βασικοί όροι που περιγράφουν τις οντότητες που εμφανίζονται στη συγκεκριμένη αρχιτεκτονική και οι οποίοι θα χρησιμοποιηθούν παρακάτω είναι : οι χρήστες, τα domains, οι διευθύνσεις των οντοτήτων XMPP σε ένα δίκτυο, οι πόροι των χρηστών, τα xmpp URI's και η ροή δεδομένων XML (XML stream) .[2]

Σε κάθε χρήστη αντιστοιχεί ένας λογαριασμός και μία μοναδική ταυτότητα στο δίκτυο που ονομάζεται ταυτότητα JID . Η ταυτότητα αυτή αποτελεί τη διεύθυνση του χρήστη στο δίκτυο XMPP. Με βάση αυτήν καθίσταται δυνατή η αναγνώριση του χρήστη από τον εξυπηρετητή, η προσθήκη του ως νέα επαφή στον κατάλογο επαφών άλλων χρηστών και η ενημέρωση της παρουσίας-κατάστασής του. Με την εγκατάσταση ενός λογισμικού εξυπηρετητή ορίζεται η

ονομασία του συγκεκριμένου τομέα (domain) στον οποίο θα συνδέονται οι χρήστες που έχουν λογαριασμούς και συνεπώς πρόσβαση στον εξυπηρετητή. Η ονομασία αυτή αποτελεί τμήμα της διεύθυνσης JID. Η τυπική μορφή μιας διεύθυνσης JID είναι user@domain και ονομάζεται bare JID. Πρόσθετα, υπάρχει η δυνατότητα χρήσης της διεύθυνσης στη μορφή user@domain/resource (full JID). Το πρότυπο RFC-6122 ορίζει για το XMPP, όπως και για άλλα πρωτόκολλα, τη μορφή του σχήματος URI (ενός καθολικού αναγνωριστικού, δηλαδή) ως: xmpp:user@domain/resource για κάθε οντότητα XMPP στον παγκόσμιο ιστό. Ως resource ορίζεται ο συγκεκριμένος πόρος του χρήστη, δηλαδή, το συγκεκριμένο μέσο με το οποίο έχει συνδεθεί ο χρήστης με τον εξυπηρετητή. Είναι ένα αναγνωριστικό που χρησιμοποιείται προκειμένου να καθορίσει που θα δρομολογηθούν τα μηνύματα καθώς ένας χρήστης έχει την δυνατότητα με ένα λογαριασμό να συνδέεται από διαφορετικά σημεία χρησιμοποιώντας διαφορετικές συσκευές ή διαφορετικό λογισμικό πελάτη . Τέλος, η ύπαρξη του resource συνεπάγεται τη δυνατότητα ενημέρωσης παρουσίας του χρήστη για κάθε διαφορετικό σημείο σύνδεσης. Το αναγνωριστικό resource μπορεί να έχει απλά τη μορφή μίας συμβολοσειράς. Με τον όρο ροή μετάδοσης XML (XML stream) αναφερόμαστε στην βασική τεχνολογική υποδομή του XMPP. Όταν ένας χρήστης-πελάτης εκκινεί μία <<συνεδρία>> με ένα εξυπηρετητή ανοίγει μία long-lived TCP σύνδεση και έπειτα διαπραγματεύεται την δημιουργία ενός καναλιού μετάδοσης XML με αυτόν. Ένα αντίστοιχο κανάλι με αντίθετη κατεύθυνση ανοίγει και ο εξυπηρετητής. Μόλις η διαπραγμάτευση του καναλιού πραγματοποιηθεί μπορεί να γίνει ανταλλαγή μεταξύ των δύο άκρων τριών διαφορετικών εξειδικευμένων τύπων μηνυμάτων XML (XML stanzas,όπως αναφέρονται στο RFC-3920) : <message/>, <presence/> και <iq/> τα οποία αναφέρθηκαν παραπάνω και θα εξηγηθούν αναλυτικά παρακάτω .[2]

Ακολουθεί ένα απλοποιημένο παράδειγμα (πηγή:[2]) της ροής XML που απεικονίζει το διάλογο μεταξύ ενός χρήστη με έναν άλλο καθώς και την παρελκόμενη επικοινωνία με τον εξυπηρετητή .

Πίνακας 5 Παράδειγμα ροής XML – XMPP Stream

```
C: <?xml version='1.0'?>
     <stream:stream
        to='example.com'
        xmlns='jabber:client'
        xmlns:stream='http://etherx.jabber.org/streams'
        version='1.0'>
  S: <?xml version='1.0'?>
     <stream:stream
        from='example.com'
        id='someid'
        xmlns='jabber:client'
        xmlns:stream='http://etherx.jabber.org/streams'
```

```
      version='1.0'>
... encryption, authentication, and resource binding ...
C:  <message from='juliet@example.com'
        to='romeo@example.net'
        xml:lang='en'>
C:    <body>Art thou not Romeo, and a Montague?</body>
C:  </message>
S:  <message from='romeo@example.net'
        to='juliet@example.com'
        xml:lang='en'>
S:    <body>Neither, fair saint, if either thee dislike.</body>
S:  </message>
C: </stream:stream>
S: </stream:stream>
```

Είναι σημαντικό να τονίσουμε, όπως αναφέρεται και στο RFC-6120 [26], πως η βασική μονάδα/δομή δεδομένων του πρωτοκόλλου είναι το μήνυμα XML, το οποίο στην ουσία αποτελεί τμήμα ενός αρχείου XML που μεταδίδεται μέσω μίας συνεχούς ροής. Η ροή XML αποτελεί το μέσο μεταφοράς της πληροφορίας από σημείο σε σημείο στο δίκτυο .

Σύμφωνα με το XMPP foundation οι τεχνολογίες που βρίσκονται στον πυρήνα του XMPP είναι [20]:

- το βασικό επίπεδο της ροής μετάδοσης XML αρχείων .
- η κρυπτογράφηση καναλιού χρησιμοποιώντας ασφάλεια επιπέδου μεταφοράς TLS (Transport Layer Security)
- ισχυρό έλεγχο πιστοποίησης χρησιμοποιώντας το πρωτόκολλο SASL (Simple Authentication and Security Layer)
- χρήση του προτύπου κωδικοποίησης χαρακτήρων UTF-8 για πλήρη υποστήριξη Unicode χαρακτήρων, περιλαμβάνοντας έτσι πλήρως διεθνή διευθυνσιοδότηση (“internationalized addresses”)
- ενσωματωμένη πληροφορία σχετικά με τη διαθεσιμότητα στο δίκτυο (παρουσία)
- συνδρομές σε παρουσία χρηστών με αμφίδρομο έλεγχο πιστοποίησης
- κατάλογος επαφών με δυνατότητα εμφάνισης παρουσίας-κατάστασης.

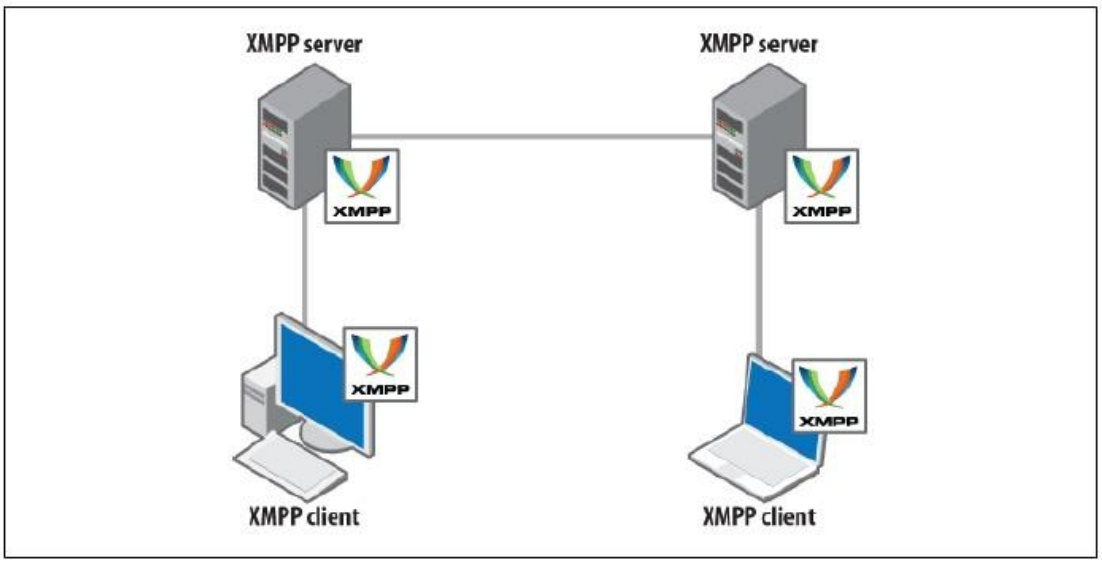
Τα βήματα που περιγράφουν μία διαδικασία σύνδεσης ενός χρήστη σε έναν εξυπηρετητή και τον τερματισμό αυτής σύμφωνα με το RFC-6120 [26] είναι :

- προσδιορισμός της διεύθυνσης IP και της πόρτας στην οποία πρέπει να συνδεθεί ο πελάτης-χρήστης, η οποία μπορεί να προκύπτει και από ένα έγκυρο όνομα domain (χρησιμοποιώντας έτσι την DNS υπηρεσία)
- εκκίνηση μίας σύνδεσης TCP (Transmission Control Protocol)

- δημιουργία ενός καναλιού μετάδοσης XML μέσω της TCP σύνδεσης
- η διαπραγμάτευση κατά προτίμηση ασφάλειας TLS για κρυπτογράφηση καναλιού
- πιστοποίηση χρησιμοποιώντας τον μηχανισμό SASL
- αντιστοίχιση ενός resource στο συγκεκριμένο κανάλι μετάδοσης
- ανταλλαγή ενός μη περιορισμένου αριθμού μηνυμάτων XML (XML stanzas) με άλλες οντότητες XMPP στο δίκτυο
- κλείσιμο του καναλιού μετάδοσης XML
- τερματισμός της σύνδεσης TCP .

Η αρχιτεκτονική XMPP επιτρέπει ακόμα τη σύνδεση εξυπηρετητών XMPP μεταξύ τους (inter-domain/inter-server) με ανάλογο τρόπο με αυτόν που περιγράφηκε παραπάνω. Είναι, λοιπόν , εφικτό χρήστες που έχουν λογαριασμό σε έναν συγκεκριμένο εξυπηρετητή να είναι διαθέσιμοι προς επικοινωνία σε χρήστες ενός διαφορετικού εξυπηρετητή που έχει ρυθμιστεί ώστε να επικοινωνεί με τον αρχικό και να δρομολογεί αναλόγως τα μηνύματα.

Εικόνα 2 Διάγραμμα αρχιτεκτονικής πελάτη-εξυπηρετητή XMPP (inter-domain) Πηγή: XMPP:The Definitive Guide[2]



Όπως αναφέρθηκε παραπάνω, μία βασική ιδιότητα που καθιστά το πρωτόκολλο XMPP ενδιαφέρον είναι η δυνατότητα επεκτασιμότητας. Με τον όρο αυτό εννοούμε ότι είναι εφικτή η χρήση αλλά και η δημιουργία επεκτάσεων, δηλαδή, προτύπων που περιγράφουν τη μορφή XML στοιχείων που μπορούν να περιληφθούν μέσα στις τρεις μορφές μηνυμάτων (message,presence,iq) με σκοπό να προσφέρουν πρόσθετη λειτουργικότητα στην επικοινωνία. Δεδομένου ότι το XMPP είναι μία αμιγώς XML τεχνολογία η υλοποίηση των επεκτάσεων γίνεται με τη χρήση των XML namespaces. Με αυτά προδιαγράφεται το είδος και η δομή του payload (φορτίου) που μπορεί να εμπεριέχει ένα οποιοδήποτε stanza ώστε να

υπακούει σε ένα συγκεκριμένο πρότυπο επέκτασης και να είναι «κατανοητό» σαν μήνυμα από τον παραλήπτη. Η αντιστοίχιση της επέκτασης γίνεται τόσο στο όνομα του στοιχείου-παιδιού (που εμπεριέχεται στο stanza) όσο και στο namespace . Παράδειγμα χρήσης των namespaces αποτελεί το παρακάτω <message/> stanza :

Πίνακας 6 Παράδειγμα μηνύματος message stanza για χρήση xml namespaces

```
<message from="you@yourdomain.tld/home"
to="friend@theirdomain.tld"
type="chat">
<body>I love this movie I saw last night, it's awesome!</body>
<html xmlns="http://jabber.org/protocol/xhtml-im">
<body xmlns="http://www.w3.org/1999/xhtml">
<p>
I <em>love</em>, this new movie I saw last night,
it's <strong>awesome</strong>!
</p>
</body>
</html>
</message>
```

Στην περίπτωση αυτή επέκταση αποτελεί το στοιχείο <html/> έχοντας ως χαρακτηριστικό το namespace xmlns="http://jabber.org/protocol/xhtml-im". Πρόκειται για την επέκταση XHTML-IM XEP-0071 η οποία χρησιμοποιείται για τη ρύθμιση του τρόπου εμφάνισης-παρουσίασης του σώματος κειμένου ενός <message/> stanza.

Μέχρι στιγμής έχουν αναπτυχθεί δεκάδες επεκτάσεις από την κοινότητα ανάπτυξης του XMPP ενώ κατά μεγάλη πλειοψηφία τέτοιες επεκτάσεις δημοσιεύονται από το XMPP Standards Foundation .

Ωστόσο, ένας χρήστης μπορεί εύκολα να ορίσει δικές του επεκτάσεις για προσωπική χρήση προκειμένου να προσδώσει πρόσθετη χρηστικότητα στις εφαρμογές του.

Ασφάλεια Σύνδεσης [26]

Το XMPP στηρίζει την ασφάλεια της επικοινωνίας στο πρωτόκολλο πιστοποίησης SASL και κρυπτογράφησης TLS. Κατά την εκκίνηση μίας ροής XML μεταξύ δύο μερών και ύστερα από την αμφίπλευρη αποστολή των επικεφαλίδων <stream:stream …> XML η πλευρά που αποτελεί τον δέκτη της σύνδεσης αποστέλλει (διαφημίζει) τα stream features, δηλαδή, τις απαιτήσεις ή προτιμήσεις που έχει όσον αφορά την κρυπτογράφηση και την πιστοποίηση της σύνδεσης. Ένα ενδεικτικό τέτοιο μήνυμα είναι το ακόλουθο :

Πίνακας 7 Ενδεικτικό τμήμα XML ροής – stream features για απαίτηση ή όχι κρυπτογράφησης

```
<stream:features>
        <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
        <required/>
        </starttls>
</stream:features>
```

Το συγκεκριμένο μήνυμα υποδηλώνει ότι ο δέκτης της σύνδεσης απαιτεί την κρυπτογράφησή της (TLS) καθώς περιέχει το στοιχείο <required/>. Στη συνέχεια ο εκκινητής της σύνδεσης πρέπει να αποστείλει μήνυμα που σηματοδοτεί την αρχή της διαπραγμάτευσης TLS σαν το εξής :

<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>

Για να είναι επιτυχής η «χειραψία» (TLS handshake) θα πρέπει ο δέκτης να απαντήσει με ένα μήνυμα XML <proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/> αλλιώς να δηλώσει την αποτυχία λόγω σφάλματος αποστέλλοντας το μήνυμα : <failure xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>.

Η αποστολή του παραπάνω μηνύματος είναι προαιρετική. Στην περίπτωση που δεν παραλείπεται η προηγούμενη διαδικασία η διαπραγμάτευση της πιστοποίησης SASL ακολουθεί σε συνέχεια αυτής. Πρώτα όμως γίνεται επανεκκίνηση της ροής (stream) ώστε ακόμα και η διαδικασία της πιστοποίησης να είναι σε κρυπτογραφημένη μετάδοση . Κατ'αυτόν τον τρόπο γίνεται αρχικά η ανταλλαγή επικεφαλίδων <stream:stream> για μία ακόμη φορά από τα δύο μέρη και έπειτα ο δέκτης αποστέλλει ένα μήνυμα <stream:features> αυτή τη φορά όμως για να διαφημίσει τις προτιμήσεις σχετικά με το μοντέλο πιστοποίησης SASL. Ενδεικτικό μήνυμα είναι το παρακάτω :

Πίνακας 8 Ενδεικτικό τμήμα XML ροής – stream features για διαφήμιση μηχανισμού ασφαλείας

```
<stream:features>
        <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
          <mechanism>EXTERNAL</mechanism>
          <mechanism>SCRAM-SHA-1-PLUS</mechanism>
          <mechanism>SCRAM-SHA-1</mechanism>
          <mechanism>PLAIN</mechanism>
        </mechanisms>
</stream:features>
```

Δηλώνονται με σειρά προτεραιότητας οι προτιμώμενοι τρόποι πιστοποίησης SASL. Ο εκκινητής οφείλει να αποστείλει ένα μήνυμα <auth/> με το κατάλληλο namespace αλλά και με τον τρόπο πιστοποίησης που υποστηρίζεται προκειμένου να αρχίσει η διαπραγμάτευση.

Πίνακας 9 XML τμήμα μηνύματος για επιλογή τρόπου πιστοποίησης

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
                mechanism='PLAIN'>AGp1bGlldAByMG0zMG15cjBtMzA=</auth>
```

Με τη σειρά του ο δέκτης ξεκινάει μια διαδικασία πρόκλησης (challenge-response) κατά την οποία αποστέλλει μήνυμα το οποίο μπορεί να περιέχει δεδομένα χαρακτήρων XML (σύμφωνα με το μοντέλο πιστοποίησης) και είναι σαν το παρακάτω:

Πίνακας 10 XML τμήμα μηνύματος για εκκίνηση διαδικασίας πρόκλησης

```
<challenge xmlns='urn:ietf:params:xml:ns:xmpp-sasl' …/>
```

Ο εκκινητής απαντά με ένα αντίστοιχο μήνυμα <response/> του οποίου το περιεχόμενο είναι μία απόκριση που παράγεται σε σχέση με το περιεχόμενο του μηνύματος <challenge/> και συμμορφώνεται με το μοντέλο πιστοποίησης που έχει οριστεί κατά την διαπραγμάτευση .

Την επικύρωση της επιτυχημένης σύνδεσης μέσω SASL πραγματοποιεί το παρακάτω μήνυμα:

Πίνακας 11 XML τμήμα μηνύματος για επικύρωση σύνδεσης XML stream

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

Ενώ σε αντίθετη περίπτωση αποστέλλεται από τον δέκτη το ένα μήνυμα όπως :

Πίνακας 12 XML τμήμα μηνύματος για αποτυχία σύνδεσης XML stream

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
            <not-authorized/>
</failure>
```

2.4.2 Μηνύματα [2]

Παρακάτω περιγράφονται εκτενέστερα οι τρείς τύποι μηνυμάτων και δίνονται παραδείγματα της μορφής τους.

Message

Είναι ο τύπος μηνύματος που χρησιμοποιείται για άμεση επικοινωνία(IM), ομαδική συζήτηση(groupchat), ειδοποιήσεις και μηνύματα ανακοινώσεων. Παράδειγμα ενός μηνύματος τύπου <message/> αποτελεί το παρακάτω xml κείμενο :

Πίνακας 13 Παράδειγμα μηνύματος <message/>

```
<message from="user1@server/mobile"
to="user2@server"
type="chat">
<body>Hello friend!</body>
<subject>greetings</subject>
</message>
```

Παρατηρούμε ότι το στοιχείο γονέας του μηνύματος XML έχει ένα όρισμα (attribute) με την ονομασία type. Οι τιμές του συγκεκριμένου ορίσματος μεταβάλλουν το είδος του μηνύματος message, δηλαδή, τον σκοπό χρήσης και ύπαρξής του και μπορούν να είναι ένα από τα παρακάτω :

- normal : πρόκειται για απλά και μεμονωμένα μηνύματα από ένα χρήστη σε έναν άλλο των οποίων η απάντηση δεν είναι απαραίτητα αναμενόμενη. Παραδίδεται άμεσα στον παραλήπτη ή αποθηκεύεται στον εξυπηρετητή εάν δεν είναι συνδεδεμένος (Offline). Αποτελεί τον προεπιλεγμένο τύπο <message/> stanza.
- chat : είναι τα μηνύματα που ανταλλάσσονται σε μία «συνεδρία» ανταλλαγής άμεσων μηνυμάτων μεταξύ δύο οντοτήτων XMPP, δηλαδή, σε μία συνομιλία (chat) ανάμεσα σε δύο χρήστες ενός συστήματος IM. Το IM λογισμικό πελάτη εμφανίζει τα συγκεκριμένα μηνύματα σε μεμονωμένο παράθυρο στη μορφή ενός διαλόγου.
- groupchat : είναι τα μηνύματα που ανταλλάσσονται σε μία συνομιλία μεταξύ πολλών χρηστών (multi-user chat room) σε ένα σύστημα IM. Τα μηνύματα αυτά δρομολογούνται από τον εξυπηρετητή σε μία εξειδικευμένη οντότητα XMPP (όπως XMPP component) η οποία φιλοξενεί multi-user chat rooms. Με τη σειρά του αποστέλλει εξερχόμενα μηνύματα προς όλους τους συμμετέχοντες.
- headline : είναι τα μηνύματα που αποστέλλονται με σκοπό την ενημέρωση ή ειδοποίηση του παραλήπτη οπότε δεν αναμένεται απόκριση σε αυτά. Το εκάστοτε λογισμικό πελάτη οφείλει να μην επιτρέπει στο χρήστη να απαντήσει σε ένα τέτοιο μήνυμα . Κατά βάση τα μηνύματα αυτά αποστέλλονται σε όλους τους διαθέσιμους (Online) χρήστες καθώς δεν υπάρχει δυνατότητα για αποθήκευση στον εξυπηρετητή και αποστολή αργότερα σε εκείνους που ήταν Offline .
- error : τα συγκεκριμένα μηνύματα αποστέλλονται σε περίπτωση που προκύπτει κάποιο σφάλμα σε σχέση με ένα μήνυμα το οποίο έχει αποσταλεί προηγουμένως. Η οντότητα που διαπιστώνει την ύπαρξη του προβλήματος επιστρέφει το μήνυμα αυτού του είδους.

Κάθε μήνυμα <message/> έχει ορίσματα from και to τα οποία ορίζουν τον αποστολέα και τον παραλήπτη του μηνύματος σε μορφή διεύθυνσης JID όπως ορίστηκε παραπάνω. Στο τελικό μήνυμα που θα λάβει ο παραλήπτης και το οποίο αποστέλλεται στην πραγματικότητα από τον εξυπηρετητή στον παραλήπτη και όχι αυτούσιο από τον αποστολέα στον παραλήπτη το πεδίο from έχει συμπληρωμένη την πραγματική διεύθυνση του η οποία αναγνωρίζεται από τον εξυπηρετητή και έτσι αποφεύγεται το address spoofing (η εξαπάτηση με ψευδή στοιχεία αποστολέα). Προαιρετικά, είναι δυνατό να υπάρχει και όρισμα με την ετικέτα id για εντοπισμό και παρακολούθηση ενός μηνύματος .

Ένα μήνυμα, όμως , έχει ως σκοπό να μεταφέρει «φορτίο» όπως ονομάζονται τα υπόλοιπα στοιχεία-παιδιά του XML αρχείου .

Οι προδιαγραφές του XMPP ορίζουν σαν βασικό τέτοιο φορτίο τα στοιχεία <body/> και <subject/> . Είναι δυνατό να υπάρχουν διάφορες μορφές φορτίου που δεν περιορίζονται στις παραπάνω.

Σε περίπτωση που το πεδίο “to” δεν είναι συμπληρωμένο με την full JID του παραλήπτη το μήνυμα αποστέλλεται στο resource του με τη μεγαλύτερη τιμή προτεραιότητας (priority : ακέραια τιμή). Η τιμή αυτή ορίζεται στη ρύθμιση του λογαριασμού χρήστη που αντιστοιχεί σε κάθε συγκεκριμένο resource .

Περιγράψαμε παραπάνω τη βασική μορφή και λειτουργία των <mesage/> stanzas τα οποία όμως μπορούν να έχουν εμπλουτισμένη πληροφορία χάρη στις δυνατότητες επεκτασιμότητας του πρωτοκόλλου XMPP. Η προδιαγραφή επέκτασης XEP-0085 προσθέτει τη λειτουργία των chat state notifications, δηλαδή, των ειδοποιήσεων κατάστασης εν μέσω μίας συνομιλίας. Περιγράφει ως καταστάσεις τις εξής :

- starting (εκκίνηση): εκκίνηση συζήτησης χωρίς να έχει συμμετάσχει ακόμα η άλλη πλευρά
- active (ενεργός): ενεργή συμμετοχή σε μία συζήτηση χωρίς όμως να πραγματοποιείται πληκτρολόγηση κάποιου μηνύματος
- composing (πληκτρολογεί): πληκτρολόγηση μηνύματος προς άμεση αποστολή
- paused (σε παύση): η διαδικασία πληκτρολόγησης ενός μηνύματος βρίσκεται σε παύση
- inactive (ανενεργός): δεν έχει συμμετάσχει στη συζήτηση για ένα ικανό χρονικό διάστημα
- gone (απουσιάζει): έχει πραγματοποιηθεί λήξη της συζήτησης από τη μία πλευρά όπως θα συνέβαινε με το κλείσιμο του παραθύρου διαλόγου της εφαρμογής IM .

Η υλοποίηση των ανωτέρω απεικονίζεται στο περιεχόμενο ενός <message/> stanza εμπλουτισμένου με chat state notifications.

Είναι σαν ένα συμβατικό μήνυμα με την προσθήκη ενός στοιχείου XML με όνομα μία από τις καταστάσεις που προαναφέρθηκαν και χαρακτηριστικό το XML namespace (xmlns="http://jabber.org/protocol/chatstates"). Παράδειγμα αποτελεί το παρακάτω:

Πίνακας 14 Παράδειγμα μηνύματος <message/> με xmlns="http://jabber.org/protocol/chatstates"

```
<message from="you@yourdomain/work"
to="daughter@yourdomain"
type="chat">
<body>Hi honey!</body>
<active xmlns="http://jabber.org/protocol/chatstates"/>
</message>
```

Presence

Είναι ο τύπος μηνυμάτων παρουσίας <presence/> τα οποία περιέχουν πληροφορία για τη διαθεσιμότητα ενός χρήστη στο δίκτυο και την κατάσταση στην οποία βρίσκεται (status). Με αυτόν τον τρόπο μπορεί κάποιος χρήστης να ενημερώνεται για τη διαθεσιμότητα των χρηστών που υπάρχουν στον κατάλογο επαφών του και αναλόγως να επικοινωνήσει μαζί τους. Η παραπάνω δυνατότητα ενημέρωσης υπάρχει αμφίδρομα μόνο αφότου ο χρήστης αποστείλει αίτημα για συνδρομή στην διαθεσιμότητά των άλλων και εκείνοι με τη σειρά τους αποδεχθούν απαντώντας με ένα συγκεκριμένο μήνυμα <presence/>. Το παρακάτω μήνυμα αποτελεί μία αίτηση από τον user1 για την δυνατότητα να παρακολουθεί τη διαθεσιμότητα και τις αλλαγές της κατάστασης του user2 :

Πίνακας 15 Ενδεικτικό μήνυμα <presence/> type='subscribe'

```
<presence from="user1@domain" to="user2@domain" type="subscribe"/>
```

Ακολούθως, ο user2 μπορεί να αποδεχθεί την αίτηση αποστέλλοντας το παρακάτω <presence/> stanza με το χαρακτηριστικό type να έχει την τιμή subscribed (αντιθέτως θα αποστείλει μήνυμα με type=unsubscribed):

Πίνακας 16 Ενδεικτικό μήνυμα <presence/> type='subscribed'

```
<presence from="user2@domain" to="user1@domain " type="subscribed"/>
```

Συνηθίζεται από το λογισμικό πελάτη IM να ακολουθεί η αυτόματη αποστολή ενός μηνύματος <presence/> τύπου "subscribe" και από τον user2 στον user1 προκειμένου να υπάρχει αμφίδρομη συνδρομή στη διαθεσιμότητα .

Συμπεραίνουμε ότι, το μοντέλο συνδρομής σε διαθεσιμότητα αποτελεί μία απλή και εξειδικευμένη μέθοδο publish-subscribe (δημοσίευσης - συνδρομής). Στην βασική του μορφή το μήνυμα <presence/> δείχνει απλά εάν κάποιος χρήστης είναι συνδεδεμένος στο δίκτυο ή όχι (online-offline). Παρέχει ωστόσο, την πρόσθετη δυνατότητα για ενημέρωση της κατάστασης του χρήστη με τέσσερις διαφορετικές και προκαθορισμένες τιμές οι οποίες εσωκλείονται στο στοιχείο <show/> του XML μηνύματος και είναι οι παρακάτω :

- Chat: χρήστης διαθέσιμος για συζήτηση
- Away: δηλώνει ότι ο χρήστης είναι απομακρυσμένος για σύντομο χρονικό διάστημα από το σημείο πρόσβασης ,δηλαδή, την εφαρμογή πελάτη IM
- Xa (extended away): δηλώνει ότι ο χρήστης είναι απομακρυσμένος για μεγάλο χρονικό διάστημα
- Dnd (do not disturb): δηλώνει ότι ο χρήστης είναι απασχολημένος και δεν επιθυμεί να επικοινωνήσει κάποιος μαζί του .

Από τις παραπάνω καταστάσεις οι chat, away και xa συνήθως παράγονται και εμφανίζονται αυτόματα από την εφαρμογή πελάτη IM. Παράλληλα, το στοιχείο <status/> μπορεί να εσωκλείει μία εξατομικευμένη περιγραφή, δηλαδή, ένα κείμενο που μπορεί να γράψει ο ίδιος ο χρήστης όπως «θα επιστρέψω σύντομα».

Σε εφαρμογές IM το λογισμικό πελάτη εμφανίζει στον κατάλογο επαφών (ο οποίος στην ορολογία του XMPP ονομάζεται roster) την διαθεσιμότητα καθεμίας. Τη στιγμή που ο χρήστης συνδέεται με τον εξυπηρετητή, το λογισμικό πελάτη αποστέλλει στον εξυπηρετητή μήνυμα <presence/> δηλώνοντας ότι είναι διαθέσιμος, δηλαδή, συνδεδεμένος . Ο εξυπηρετητής ελέγχει τις επαφές του καταλόγου και αποστέλλει ειδοποίηση με μήνυμα <presence/> σε κάθε έναν με ενεργή συνδρομή συμπληρώνοντας πάντα την full JID στο πεδίο from. Για την ενημέρωση της διαθεσιμότητας των επαφών ο εξυπηρετητής αποστέλλει ένα μήνυμα όπως το παρακάτω για καθεμία χωριστά:

Πίνακας 17 Ενδεικτικό μήνυμα <presence/> type='probe'

```
<presence from="user1@domain/notebook"
to="friend@domain"
type="probe"/>
```

Με αυτόν τον τρόπο η κάθε επαφή στην οποία έχει ενεργή συνδρομή ο χρήστης user1 ενεργοποιείται και αποστέλλει μήνυμα <presence/> εάν είναι Online, διαφορετικά ο εξυπηρετητής της αποστέλλει μήνυμα σαν το παρακάτω:

Πίνακας 18 Ενδεικτικό μήνυμα <presence/> type='unavailable'

```
<presence from="an_offline_friend@domain/desktop"
to="user1@domain/iphone"
type="unavailable">
<delay xmlns="urn:xmpp:delay"
stamp="2011-06-26T18:29:10Z"/>
</presence>
```

δηλώνοντας έτσι την τελευταία χρονική στιγμή που υπήρξε διαθέσιμη.

Ένα ενδεικτικό μήνυμα <presence/> με στοιχεία που αφορούν την κατάσταση του χρήστη είναι :

Πίνακας 19 Ενδεικτικό μήνυμα <presence/> με πληροφορίες για την κατάσταση (status) του χρήστη

```
<presence from="user@domain/iphone">
<show>xa</show>
<status>guess i'm not online…</status>
</presence>
```

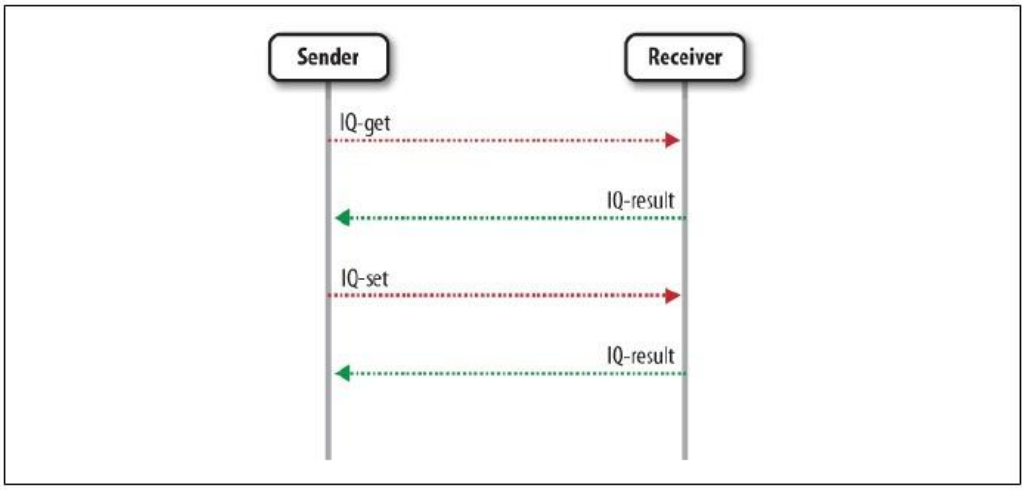
Iq (info/query)

Τα μηνύματα iq είναι μηνύματα που παρέχουν μία δομή για αλληλεπιδράσεις μεταξύ των χρηστών και του εξυπηρετητή ή μίας οντότητας XMPP στη μορφή αιτημάτων – αποκρίσεων. Σε αντίθεση με τα <message/> stanzas τα iq μπορεί να έχουν ένα μόνο συγκεκριμένο φορτίο το οποίο έχει τυποποιημένη δομή και προσδιορίζει επακριβώς την ενέργεια που πρέπει να εκτελεστεί από τον παραλήπτη. Επίσης, σε κάθε οντότητα που αποστέλλει ένα μήνυμα iq πρέπει να επιστρέφεται μία απάντηση με ιδίου τύπου μήνυμα είτε αυτό παράγεται από ένα χρήστη ή από τον εξυπηρετητή. Στο στοιχείο πατέρα του XML stanza είναι απαραίτητη η ύπαρξη του ορίσματος id το οποίο χρησιμοποιείται στην αντιστοίχιση του αιτήματος στην απόκριση. Το όρισμα type δέχεται τις παρακάτω τιμές :

- get : στην περίπτωση αυτή η αιτούμενη οντότητα ζητά πληροφορίες όπως απαιτήσεις για την εγγραφή και τη δημιουργία ενός λογαριασμού.

- set : στην περίπτωση αυτή η αιτούμενη οντότητα στέλνει πληροφορίες ή πραγματοποιεί ένα νέο αίτημα.
- result : η αποκρινόμενη οντότητα επιστρέφει μία απάντηση σε ένα iq stanza με τύπο get όπως τα στοιχεία που αποστέλλει ένας χρήστης για να εγγραφεί σε έναν εξυπηρετητή. Ο τύπος αυτός χρησιμοποιείται και στην περίπτωση απλής απάντησης αναγνώρισης σε ένα αίτημα iq stanza με τύπο set .
- error : η αποκρινόμενη οντότητα ή οποιαδήποτε ενδιάμεση (όπως ένας εξυπηρετητής) ειδοποιεί την αιτούμενη οντότητα ότι δεν ήταν δυνατή η επεξεργασία του αιτήματος τύπου get ή set .

Εικόνα 3 Τα μηνύματα iq προσφέρουν μία δομημένη αλληλεπίδραση μεταξύ πελάτη και εξυπηρετητή
Πηγή:XMPP:The Definitve Guide[2]



Ένα παράδειγμα αλληλεπίδρασης μεταξύ χρήστη και εξυπηρετητή μέσω iq μηνυμάτων είναι η αίτηση στον εξυπηρετητή με ένα μήνυμα τύπου get σχετικά με τον κατάλογο επαφών του χρήστη . Το συγκεκριμένο μήνυμα έχει κενό φορτίο(payload). Στη συνέχεια ο εξυπηρετητής αποκρίνεται με ένα μήνυμα τύπου result το οποίο σαν φορτίο φέρει από ένα στοιχείο <item/> για κάθε μία επαφή.

Το αρχικό μήνυμα που αποστέλλει ο χρήστης είναι :

Πίνακας 20 Ενδεικτικό μήνυμα <iq/> για αίτηση του καταλόγου επαφών χρήστη (roster)

```
<iq from="user@domain/iphone"
id=" pk1456z "
to="user@domain"
type="get">
<query xmlns="jabber:iq:roster"/>
</iq>
```

και ακολουθεί η απάντηση από τον εξυπηρετητή :

Πίνακας 21 Ενδεικτικό μήνυμα <iq/> ως απάντηση με τον κατάλογο επαφών χρήστη (roster)

```
<iq from="user@domain"
id="pk1456z"
to=" user@domain/iphone "
type="result">
<query xmlns="jabber:iq:roster">
<item jid="usr1@domain"/>
<item jid="usr2@domain"/>
<item jid="usr3@domain"/>
<item jid="usr4@domain"/>
</query>
</iq>
```

Παρατηρούμε ότι το είδος του φορτίου προσδιορίζεται από το χαρακτηριστικό xmlns του στοιχείου <query/>, δηλαδή, το namespace, το οποίο έχει την τιμή "jabber:iq:roster" .

Σε συνέχεια του παραδείγματος, ένας χρήστης θα μπορούσε να προσθέσει μία επαφή στο roster του αποστέλλοντας το παρακάτω iq μήνυμα :

Πίνακας 22 Ενδεικτικό μήνυμα <iq/> για προσθήκη επαφής χρήστη

```
<iq from="user@domain/iphone "
id="ku267gh3"
to="user@domain"
type="set">
<query xmlns="jabber:iq:roster">
<item jid="newfriend@domain"/>
</query>
</iq>
```

Ο εξυπηρετητής επιστρέφει ως αναγνώριση του αιτήματος το παρακάτω μήνυμα χωρίς φορτίο .

Πίνακας 23 Ενδεικτικό μήνυμα <iq/> ως αποτέλεσμα αναγνώρισης αιτήματος

```
<iq from="user@domain"
id="ku267gh3"
to="user@domain/iphone"
type="result"/>
```

2.4.3 Επέκταση πρωτοκόλλου Publish-Subscribe [XEP-0060] [2]

Μία ιδιαίτερα σημαντική επέκταση του XMPP στην οποία στηρίζεται και η υλοποίηση της συγκεκριμένης διπλωματικής είναι το πρότυπο XEP-0060 το οποίο προδιαγράφει τη

λειτουργία μίας υπηρεσίας δημοσιεύσεων/ειδοποιήσεων γνωστό ως Publish-Subscribe. Βασικά συστατικά που αλληλεπιδρούν σε μία τέτοια υπηρεσία είναι ο κόμβος στον οποίο αντιστοιχούν και αποθηκεύονται οι δημοσιεύσεις/ειδοποιήσεις, ο συνδρομητής ο οποίος λαμβάνει ένα μήνυμα με το περιεχόμενο της ειδοποίησης μόλις γίνει μία δημοσίευση στον κόμβο στον οποίο είναι εγγεγραμμένος και ο χρήστης που είναι εξουσιοδοτημένος να δημοσιεύει νέα μηνύματα σε έναν κόμβο.

Το XEP-0060 περιγράφει τη σύνταξη των μηνυμάτων που απαιτούνται προκειμένου να πραγματοποιηθούν οι ενέργειες δημιουργίας και διαχείρισης ενός κόμβου, εγγραφής και απεγγραφής ενός χρήστη ως συνδρομητή ενός κόμβου, δημοσίευσης ενός μηνύματος σε ένα κόμβο και ειδοποίησης ενός συνδρομητή.

Παρακάτω παρατίθενται ενδεικτικά μηνύματα XMPP από κάθε κατηγορία ενεργειών της υπηρεσίας publish-subscribe (Πηγή: [2]):

Publish: Δημοσίευση ενός μηνύματος (ή αντικειμένου item) από τον αποστολέα σε ένα κόμβο

Πίνακας 24 Παράδειγμα μηνύματος <iq/> publish Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="adult@holiday.lit/car"
id="wpd7x937"
to="pubsub.holiday.lit"type="set">
<pubsub xmlns="http://jabber.org/protocol/pubsub">
<publish node="are-we-there-yet">
<item>
<there xmlns="http://holiday.lit/there-yet" status="true"/>
</item>
</publish>
</pubsub>
</iq>
```

Subscribe / Unsubscribe : Εγγραφή / Κατάργηση εγγραφής χρήστη από συνδρομητή σε ορισμένο κόμβο. Η εγγραφή ενός χρήστη σε ένα κόμβο έχει ως αποτέλεσμα την αποστολή ειδοποιήσεων σε αυτόν με περιεχόμενο τα αντικείμενα Items που έχουν δημοσιευθεί προηγουμένως στον κόμβο αυτό.

Πίνακας 25 Παράδειγμα μηνύματος <iq/> subscribe Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="alice@wonderland.lit/rabbithole"
id="gh921nx3"
to="notify.wonderland.lit"
type="set">
<pubsub xmlns="http://jabber.org/protocol/pubsub">
<subscribe node="queenly_proclamations" jid="alice@wonderland.lit"/>
```

```
</pubsub>
</iq>
```

Και έπειτα αποστέλλεται η επικύρωση της επιτυχημένης συνδρομής:

Πίνακας 26 Παράδειγμα μηνύματος <iq/> subscription='subscribed' Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="notify.wonderland.lit"
id="gh921nx3"
to="alice@wonderland.lit/rabbithole"
type="result">
<pubsub xmlns="http://jabber.org/protocol/pubsub">
<subscription node="queenly_proclamations" jid="alice@wonderland.lit"
subscription="subscribed"/>
</pubsub>
</iq>
```

Πίνακας 27 Παράδειγμα μηνύματος <iq/> unsubscribe Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="alice@wonderland.lit/rabbithole"
id="vd923k66"
to="notify.wonderland.lit"
type="set">
<pubsub xmlns="http://jabber.org/protocol/pubsub">
<unsubscribe node="queenly_proclamations" jid="alice@wonderland.lit"/>
</pubsub>
</iq>
```

Και η επικύρωση της επιτυχημένης απεγγραφής από τη συνδρομή:

Πίνακας 28 Παράδειγμα μηνύματος <iq/> result Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="notify.wonderland.lit"
id="vd923k66"
to="alice@wonderland.lit/rabbithole"
type="result"/>
```

Create / Delete node : Δημιουργία / διαγραφή κόμβου στον οποίο πραγματοποιούνται δημοσιεύσεις και εγγράφονται συνδρομητές.

Πίνακας 29 Παράδειγμα μηνύματος <iq/> create node Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="queen@wonderland.lit/throne"
id="cr561nd0"
to="notify.wonderland.lit"
type="set">
<pubsub xmlns="http://jabber.org/protocol/pubsub">
<create node="queenly_proclamations"/>
</pubsub>
</iq>
```

Πίνακας 30 Παράδειγμα μηνύματος <iq/> delete node Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="queen@wonderland.lit/throne"
id="d3l3t41t"
to="notify.wonderland.lit"
type="set">
<pubsub xmlns="http://jabber.org/protocol/pubsub#owner">
<delete node="queenly_proclamations"/>
</pubsub>
</iq>
```

Πίνακας 31 Παράδειγμα μηνύματος <message/> event delete node Πηγή : XMPP:The Definitive Guide[2]

```
<message from="notify.wonderland.lit" to="knave@wonderland.lit">
<event xmlns="http://jabber.org/protocol/pubsub#event">
<delete node="queenly_proclamations"/>
</event>
</message>
```

Request items / disco : Με τα μηνύματα <iq/> αυτού του είδους ο αποστολέας χρησιμοποιεί το κατάλληλο namespace “http://jabber.org/protocol/disco#info” για να

Ζητήσει πληροφορίες σχετικά με τις υπηρεσίες του παραλήπτη. Ο παράλήπτης απαντά αποστέλλοντας ένα μήνυμα <iq/> type=’result’ μήνυμα με περιεχόμενο τις διαθέσιμες υπηρεσίες. Αν αναφέρεται η υπηρεσία pubsub τότε ο χρήστης (αρχικός αποστολέας) αποστέλλει μήνυμα με το namespace “http://jabber.org/protocol/disco#items” για να ζητήσει πληροφορίες σχετικά με τους κόμβους που διαθέτει η υπηρεσία.

Πίνακας 32 Παράδειγμα μηνυμάτων <iq/> disco info type=’get’ & ‘result’ Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="knave@wonderland.lit/croquetlawn"
to="notify.wonderland.lit"
id="d1nfg39e"
type="get">
<query xmlns="http://jabber.org/protocol/disco#info"/>
</iq>

<iq from="notify.wonderland.lit"
id="d1nfg39e"
to="knave@wonderland.lit/croquetlawn"
type="result">
<query xmlns="http://jabber.org/protocol/disco#info">
<identity category="pubsub" type="service"/>
<feature var="http://jabber.org/protocol/pubsub"/>
</query>
</iq>
```

Πίνακας 33 Παράδειγμα μηνυμάτων <iq/> disco items type='get' & 'result' Πηγή : XMPP:The Definitive Guide[2]

```
<iq from="knave@wonderland.lit/croquetlawn"
id="nb74fg13"
to="notify.wonderland.lit"
type="get">
<query xmlns="http://jabber.org/protocol/disco#items"/>
</iq>

<iq from="notify.wonderland.lit"
id="nb74fg13"
to="knave@wonderland.lit/croquetlawn"
type="result">
<query xmlns="http://jabber.org/protocol/disco#items">
<item jid="notify.wonderland.lit"
node="blogregator"
name="Weblogs"/>
<item jid="notify.wonderland.lit"
node="croquet_results"
name="Results from croquet games"/>
</query>
</iq>
```

Πίνακας 34 Παράδειγμα μηνύματος <message/> event με φορτίο items Πηγή : XMPP:The Definitive Guide[2]

```
<message from="pubsub.holiday.lit" to="child@holiday.lit">
<event xmlns="http://jabber.org/protocol/pubsub#event">
<items node="are-we-there-yet">
<item id="bc42su0a93">
<there xmlns="http://holiday.lit/there-yet" status="true"/>
</item>
</items>
</event
</message>
```

2.4.4 XMPP Component ([3], σελίδα 7)

Εκτός από τη βασική XMPP οντότητα, δηλαδή, τον προσδιοριζόμενο από την ταυτότητα JID χρήστη (πελάτης, XMPP client), το πρωτόκολλο XMPP με την ύπαρξη της επέκτασης Jabber Component Protocol XEP-0114[28] προσφέρει τη δυνατότητα δημιουργίας και χρήσης εξωτερικών συστατικών εξυπηρετητή (external server components) ως ανεξάρτητες εξωτερικές οντότητες συνδεόμενες άμεσα με τον εξυπηρετητή με χρήση του ονόματος τους και ενός κωδικού για πιστοποίηση. Η ύπαρξη αυτής της δυνατότητας έχει σκοπό να προσδώσει πρόσθετη λειτουργικότητα στον εξυπηρετητή με τη δημιουργία νέων αυτόνομων υπηρεσιών μέσω των XMPP components. Κάθε component αποτελεί μία ξεχωριστά διευθυνσιοδοτούμενη οντότητα αποκτώντας διεύθυνση στη μορφή “component.myserver” εάν υποθέσουμε ότι ένας χρήστης που ανήκει στον ίδιο εξυπηρετητή έχει JID “user@myserver” και συνεπώς αναπαρίσταται στο υπόλοιπο δίκτυο σαν ένας υπό-τομέας του εξυπηρετητή (sub-domain).

Κατά βάση, οι εξυπηρετητές XMPP δεν διαχειρίζονται περισσότερα θέματα πέραν της δρομολόγησης XML stanza μεταξύ διασυνδεδεμένων XMPP οντοτήτων. Συνεπώς, με τη δημιουργία ενός component είναι εφικτή η υλοποίηση οποιασδήποτε πρόσθετης υπηρεσίας η οποία μπορεί να λειτουργεί εξωτερικά από τον εξυπηρετητή. Στην ουσία, το βασικό γνώρισμα ενός component που επιτρέπει τη λειτουργία του αυτόνομα είναι το γεγονός ότι είναι ένας υπό-τομέας του εξυπηρετητή στον οποίο μπορούν να δρομολογηθούν διάφοροι τύποι μηνυμάτων stanza άμεσα, δηλαδή, χώρις την επεξεργασία και τη διαχείριση τους πρώτα από τον εξυπηρετητή. Με αυτόν τον τρόπο, είναι πολύ εύκολο να μειωθεί σημαντικά ο φόρτος επεξεργασίας μηνυμάτων ενός εξυπηρετητή και ο ρόλος του να περιοριστεί μόνο στην προώθηση αυτών. Μία συνηθισμένη πρακτική που αξιοποιεί αυτές τις δυνατότητες είναι

η διαχείριση του καταλόγου επαφών και κατ'επέκταση των μηνυμάτων παρουσίας (presence stanza) μέσα από το ίδιο το component. Αυτό σημαίνει ότι όλες οι διαδικασίες διαχείρισης μηνυμάτων παρουσίας (presence handling) υλοποιούνται από το XMPP component και όχι από τον εξυπηρετητή. Στις περισσότερες περιπτώσεις αποτελεί αναγκαία λύση καθώς οι εξυπηρετητές XMPP δεν μπορούν να διαχειριστούν καταλόγους επαφών με μέγεθος της τάξης των χιλιάδων εγγραφών .

Κλασικό παράδειγμα XMPP component αποτελεί η δημιουργία «δωματίων» συζήτησης με πολλούς χρήστες στα συστήματα IM (Multi-user chat-room). Ο εξυπηρετητής επιτρέπει σε ένα component να δρομολογεί εσωτερικά και να διαχειρίζεται stanzas από μόνο του. Έτσι, το component μπορεί να δημιουργεί υπό-τμήματα με ξεχωριστές διευθύνσεις όπως «δωμάτια» και χρήστες.

Συμπεραίνουμε, λοιπόν, ότι η χρήση των XMPP components είναι ιδιαίτερα σημαντική στο σχεδιασμό ενός συστήματος το οποίο προορίζεται για αυξημένες απαιτήσεις κλιμάκωσης χρήσης και κατά συνέπεια προϋποθέτει την ελάφρυνση του φόρτου που επιβαρύνει τον εξυπηρετητή ώστε να διασφαλιστεί η αξιοπιστία και η ταχύτητα της υλοποίησης. Καταλήγουμε, λοιπόν, ότι το component αποτελεί ένα είδος εξειδικευμένου εξυπηρετητή για την παροχή μίας υπηρεσίας και ως αποτέλεσμα έχει συχνά την ονομασία του υπό-εξυπηρετητή (sub-server) .

2.4.5 Σύγκριση της βασικής τεχνολογίας υπηρεσιών ιστού (REST) με το πρωτόκολλο XMPP

Συμπεραίνουμε από τα παραπάνω ότι το XMPP είναι ένα καλά σχεδιασμένο πρωτόκολλο με ποικίλες δυνατότητες και διαφορετικές εκφάνσεις στον τρόπο χρήσης του. Ωστόσο, κάθε εφαρμογή αυτού έχει ως κύριο γνώμονα την λειτουργία ενός μοντέλου πελάτη – εξυπηρετητή σε πραγματικό χρόνο και μεγάλη κλίμακα. Σε αυτή την εργασία εξετάζεται ως κατάλληλη τεχνολογία για την υποστήριξη μίας υπηρεσίας ιστού που αλληλεπιδρά με τα κοινωνικά μέσα δικτύωσης. Μέχρι σήμερα, όμως, είναι ευρέως διαδεδομένη αλλά και αδιαμφισβήτητα επιτυχημένη η χρήση της αρχιτεκτονικής REST ως βασικής μεθοδολογίας λειτουργίας μίας οποιασδήποτε υπηρεσίας ιστού με υψηλές απαιτήσεις σε αξιοπιστία και σε μέγεθος κλιμάκωσης. Οι πυρήνες των δύο παραπάνω προσεγγίσεων για τη δημιουργία μίας υπηρεσίας ιστού απέχουν στο σχεδιασμό τους σημαντικά ενώ οι πιο κρίσιμες διαφορές εντοπίζονται στις παρακάτω παραγράφους.

Προσέγγιση για δημιουργία υπηρεσίας-ιστού:

Αρχικά, θα πρέπει να επισημάνουμε ότι στα πλαίσια της εργασίας αυτής επιλέχθηκε μία σχετικά νέα τεχνολογία (XMPP) για την υποστήριξη μίας νέας υπηρεσίας που θα μπορούσε να είχε υλοποιηθεί εναλλακτικά και κατά ένα τρόπο παραδοσιακά με την αρχιτεκτονική REST. Το γεγονός αυτό, όμως, δεν σημαίνει ότι συγκρίνονται δύο άναλογα αντικείμενα καθώς η REST προσέγγιση είναι μία συγκεκριμένη αρχιτεκτονική και μεθοδολογία ενώ το XMPP είναι ένα πρωτόκολλο. Συγκεκριμένα, όπως αναφέρθηκε και παραπάνω μία REST προσέγγιση για υπηρεσία ιστού υποννοεί την χρήση της αρχιτεκτονικής αυτής στη βάση του πρωτοκόλου HTTP. Επιπλέον, είναι ξεκάθαρο πως το πρωτόκολλο XMPP σχεδιάστηκε αρχικά για να υποστηρίζει την εύκολη ανάπτυξη συστημάτων IM καθώς όλα τα βασικά συστατικά ενός τέτοιου συστήματος όπως τυποποίηση μηνυμάτων, διαχείριση διαθεσιμότητας-παρουσίας, αποθήκευση και διαχείριση καταλόγου επαφών κάθε χρήστη στον εξυπηρετητή, βρίσκονται ενσωματωμένα ως προδιαγραφές και πρότυπα στον πυρήνα του. Κατ΄αυτόν τον τρόπο καθίσταται ιδανική τεχνολογία για την δημιουργία συστημάτων ανταλλαγής μηνυμάτων και αποστολής ενημερώσεων/ειδοποιήσεων πραγματικού χρόνου μεταξύ δύο ή περισσότερων χρηστών προσφέροντας πληθώρα πρόσθετων λειτουργιών λόγω της επεκτασιμότητάς του κάτι το οποίο δεν μπορεί να υποστηρίξει η αρχιτεκτονική REST καθώς η προδιαγραφή των αρχών του είναι πολύ απλή σε σχέση με του XMPP και δεν περιέχει έτοιμες λύσεις για κάτι αντίστοιχο.

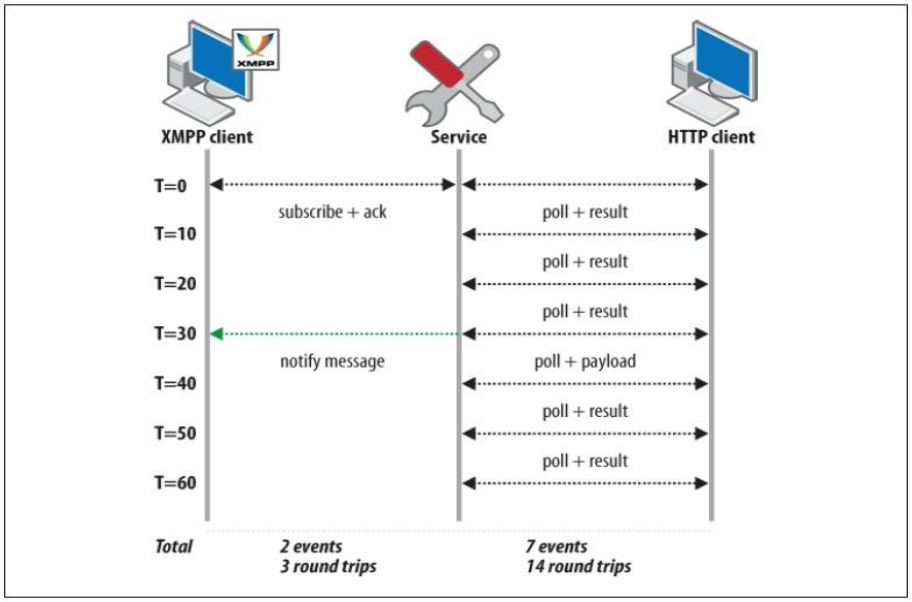
Ταχύτητα-Λειτουργία μετάδοσης

Δεδομένου ότι η «γέννηση» του XMPP είχε σκοπό την υποστήριξη συστημάτων IM για τα οποία η επικοινωνία και ανταλλαγή πληροφορίας σε πραγματικό χρόνο είναι απαιτούμενη, είναι γεγονός ότι το πρωτόκολλο αυτό εξασφαλίζει την άμεση και απρόσκοπτη ανταλλαγή μηνυμάτων μικρού μεγέθους τόσο μεταξύ πελάτη και εξυπηρετητή όσο και μεταξύ δύο πελατών. Η αρχιτεκτονική XMPP στηρίζεται κατά βάση στην έννοια των μόνιμων-επίμονων συνδέσεων (persistent). Για τη δημιουργία ενός καναλιού επικοινωνίας δημιουργείται στην πραγματικότητα ένα TCP socket το οποίο παραμένει ανοικτό όσο ένας χρήστης είναι συνδεδεμένος με τον εξυπηρετητή. Αυτό συνεπάγεται ότι η επικοινωνία μπορεί να είναι αφενός ασύγχρονη σε αντίθεση με την REST αρχιτεκτονική όπου κάθε αίτημα από τη μία πλευρά προκαλεί μία απόκριση από την άλλη και αφετέρου ταχύτατη (σχεδόν σε πραγματικό χρόνο) για κάθε ένα κανάλι χωριστά. Εκεί εντοπίζεται μία ακόμη μεγάλη διαφορά ανάμεσα στις δύο αρχιτεκτονικές, στο σημείο όπου η ύπαρξη των επίμονων συνδέσεων TCP αποδεικνύεται ιδιαίτερα αποδοτική για την κίνηση στο δίκτυο στην περίπτωση μεγάλου φόρτου (εκκατομμυρίων μηνυμάτων), η οποία μάλιστα γίνεται προς τις δύο κατευθύνσεις χωρίς να διακόπτεται η ροή της πληροφορίας.

Η αρχιτεκτονική REST εφαρμοσμένη με «παραδοσιακό τρόπο» σε μία υπηρεσία ιστού συνεπάγεται ότι για την ανανέωση πληροφορίας από τον εξυπηρετητή χρειάζεται η πραγματοποίηση αιτημάτων HTTP GET ανά τακτά χρονικά διαστήματα προκειμένου να αποσταλεί από τον εξυπηρετητή η νέα αναπαράσταση ενός πόρου (διαδικασία polling). Ωστόσο, όπως είναι φυσικό, ορισμένα μόνο από τα παραπάνω αιτήματα έχουν ως απόκριση μία πραγματικά ανανεωμένη και τροποποιημένη αναπαράσταση του πόρου με αποτέλεσμα να προστίθεται φόρτος στο δίκτυο χωρίς όφελος και συγχρόνως να μην γίνεται η ενημέρωση σε πραγματικό χρόνο καθώς τα χρονικά διαστήματα ανανέωσης δεν μπορούν να συρρικνωθούν τόσο ώστε να προσεγγίσουν το χρονική απόκριση του XMPP. Αντίθετα, με τη χρήση του πρωτοκόλλου XMPP είναι εξασφαλισμένη η στιγμιαία ενημέρωση για νέα πληροφορία από τον εξυπηρετητή στον πελάτη μόλις αυτή δημιουργηθεί (push notifications).

Στα πλαίσια της συγκεκριμένης εργασίας όπου εξετάζεται η υλοποίηση ενός συστήματος ειδοποιήσεων πραγματικού χρόνου, με αμφίδρομη ανταλλαγή δεδομένων με τα κοινωνικά μέσα δικτύωσης η χρήση του πρωτοκόλλου XMPP ενδείκνυται προκειμένου να επιτευχθεί η επικοινωνία σε πραγματικό χρόνο. Εναλλακτικές λύσεις που υιοθετούν την αρχιτεκτονική REST και προσεγγίζουν την απόκριση των υπηρεσιών σε πραγματικό χρόνο έχουν εφαρμοστεί με χαρακτηριστική αυτή του Simple Update Protocol από την FriendFeed.com [23][24]. Η παραπάνω λύση προβλέπει ότι σε κάθε τροφοδοσία νέου περιεχομένου ειδήσεων (Atom ή RSS feed) που την υλοποιεί θα πρέπει να περιέχεται ένα στοιχείο XML <link/> στο οποίο θα αναγράφεται το SUP-ID του συγκεκριμένου feed και το url ενός SUP feed. Με τη φράση SUP feed ορίζεται μία αναπαράσταση δεδομένων που δηλώνει τα feed στα οποία έχουν πραγματοποιηθεί αλλαγές (ανανεώσεις). Συνεπώς, ο αναγνώστης του feed έχει τη δυνατότητα να γνωρίζει ακριβώς ποια είναι τα θέματα που έχουν ανανεωθεί προκειμένου να στείλει αίτημα μόνο για αυτά. Μπορεί, λοιπόν, η διαδικασία του polling να περιοριστεί σε πιο αραιά χρονικά διαστήματα και να γίνεται εντατικό polling στο SUP feed. Κατ' αυτόν τον τρόπο εκτιμάται ότι το συνολικό μέγεθος του feed polling θα μειωθεί έως και 90% , ενώ οι ανανεώσεις θα εμφανίζονται δέκα φορές νωρίτερα .

Εικόνα 4 Σχήμα σύγκρισης XMPP μετάδοσης με διαδικασία polling (REST) Πηγή: XMPP:The Definitive Guide [2]



Εφαρμοσμένες λύσεις-Κοινότητα και υλικό

Από την αρχή της εφαρμογής της αρχιτεκτονικής REST στον παγκόσμιο ιστό, δηλαδή, στο πρωτόκολλο HTTP, τα σημαντικά πλεονεκτήματα σχεδιασμού της άρχισαν να αναδεικνύονται μέσα από τη χρήση της στη δημιουργία υπηρεσιών ιστού (web services). Οι αρχές του REST έχουν χρησιμοποιηθεί για μεγάλο χρονικό διάστημα σε πληθώρα εφαρμογών υπηρεσιών ιστού προκειμένου να ανταλλάσσονται άμεσα και με εύκολο τρόπο δεδομένα μεταξύ χρηστών και εξυπηρετητών από site με υποδομές ικανές να διαχειρίζονται πολλά αιτήματα στη μονάδα του χρόνου και να επιστρέφουν δεδομένα στον χρήστη τα οποία εμφανίζονται σε ένα web interface. Ιδιαίτερη σημασία έχει το γεγονός ότι τέτοιες υποδομές έχουν αναπτυχθεί σε μεγάλη κλίμακα και έχουν λειτουργήσει απρόσκοπτα και αξιόπιστα για χρονικό διάστημα πολλών ετών χάριν στην απλότητα των αρχών του REST, στην αποδοτικότητά, τη φορητότητα και την ευελιξία τροποποίησης του. Κατ΄επέκταση υπάρχει μία πολύ ανεπτυγμένη κοινότητα που ασχολείται με υλοποιήσεις REST καθώς επίσης και διαθέσιμο υλικό σε κώδικα, βιβλιοθήκες και έτοιμες λύσεις που έχουν εφαρμοστεί και ωριμάσει στον παγκόσμιο ιστό [21]. Όπως είναι φυσικό, μία υπηρεσία ιστού όπως αυτές που προσφέρει μία πλατφόρμα κοινωνικών μέσων δικτύωσης ή μία οποιαδήποτε πλατφόρμα που στηρίζεται στην προσωπική συμμετοχή των χρηστών με ατομικούς λογαριασμούς δεν είναι αναγκαίο να υποστηρίζει απόκριση σε πραγματικό χρόνο καθώς κάτι τέτοιο δεν απαιτείται από τον ίδιο τον χρήστη όταν εκείνος χρησιμοποιεί το web interface της. Αντιθέτως, το πρωτόκολλο XMPP διαθέτει αρκετά πιο περιορισμένη κοινότητα που απασχολείται με υλοποιήσεις καθώς αναπτύσσεται μόνο την τελευταία δεκαετία.

Επεκτασιμότητα

Όπως αναφέρθηκε και παραπάνω κύριο χαρακτηριστικό του XMPP είναι η επεκτασιμότητα του, η οποία πλέον των δυνατοτήτων που δίνει με τις υπάρχουσες επεκτάσεις επιτρέπει στον developer να δημιουργήσει πάνω σε μία τυποποιημένη βάση προκειμένου να υλοποιήσει επιτυχώς υπηρεσίες ιστού που υποστηρίζουν επικοινωνία πραγματικού χρόνου. Είναι συνεπώς εύκολο για κάποιον να δημιουργήσει ένα σύστημα που θα αποτελείται από τη μία πλευρά από ένα backend, δηλαδή, την υποδομή πίσω από τον εξυπηρετητή που θα επεξεργάζεται τα αιτήματα και θα παράγει τις απαντήσεις προς αποστολή και από την άλλη από τελικούς χρήστες (end-users) με πρόσβαση σε εφαρμογή πελάτη ή σε web interface που θα εκτελεί κώδικα πελάτη με σκοπό να παρέχονται υπηρεσίες πραγματικού χρόνου στους χρήστες. Η υλοποίηση μίας υπηρεσίας σαν την παραπάνω σε επίπεδο μίας κλασικής υπηρεσίας web μπορεί να είναι πιο δύσκολη απ' ότι με την χρήση της REST προσέγγισης όμως μπορεί να προσδώσει πολλές πρόσθετες δυνατότητες οι οποίες είναι ενσωματωμένες σε αυτό (και που δεν υπάρχουν έτοιμες στο REST) όπως : remote commands / data forms / publish-subscribe service / components (bots που τρέχουν απομακρυσμένα με real time επικοινωνία) .

Ασφάλεια σύνδεσης

Σε επίπεδο ασφάλειας της σύνδεσης και οι δύο αρχιτεκτονικές παρέχουν λύσεις που εγγυώνται την ασφαλή και προστατευμένη από εξωτερικές επιθέσεις επικοινωνία. Στο XMPP η πιστοποίηση πριν την εκκίνηση μίας σύνδεσης εξασφαλίζεται με την χρήση κάποιου από τους μηχανισμούς challenge-respone που παρέχει το SASL και επιπρόσθετα μπορεί να υποστηρίξει κρυπτογράφηση του καναλιού επικοινωνίας με το TLS. Επειδή η αρχιτεκτονική του REST υλοποιείται στον παγκόσμιο ιστό πάνω από το πρωτόκολλο HTTP η ασφάλεια στην σύνδεση κατοχυρώνεται με τη χρήση πιστοποιητικών που εκδίδονται από αρχές πιστοποίησης (π.χ. VeriSign) όσον αφορά την πλευρά του εξυπηρετητή και με το SSL/TLS για την κρυπτογράφηση και προστασία του καναλιού σε όλη τη διάρκεια της σύνδεσης (HTTPS - secure). Ωστόσο, το πρωτόκολλο HTTP συναντά πολλές φορές εμπόδια λόγω της εκτεταμένης χρήσης τοιχών ασφαλείας (Firewalls) και της ύπαρξης του NAT(network address translation), δηλαδή, της αντιστοίχισης μίας εξωτερικής διεύθυνσης IP μίας σύνδεσης στην εσωτερική που αφορά ένα συγκεκριμένο μηχάνημα. Αντίθετα, το XMPP είναι ιδιαίτερα «φιλικό» ως προς τη χρήση των τοιχών ασφαλείας (Firewalls) καθώς χρησιμοποιεί επίμονες TCP συνδέσεις μεταξύ χρήστη και εξυπηρετητή.([3] σελ.28)

Διευθυνσιοδότηση

Ένα ακόμη βασικό πλεονέκτημα που προσφέρει το XMPP είναι ότι έχει ενσωματωμένη στον πυρήνα του τη διευθυνσιοδότηση σύμφωνα με όσα αναφέρθηκαν στις παραγράφους που αφορούν την τεχνολογία του XMPP. Κάθε οντότητα έχει μοναδική ταυτότητα και γι'αυτό μπορεί να υποστηριχθεί η ασύγχρονη επικοινωνία τόσο μεταξύ πελάτη και εξυπηρετητή όσο και μεταξύ πελατών ή components (bots). Επιπρόσθετα, η ύπαρξη της διεύθυνσης full JID δίνει τη δυνατότητα για ευελιξία του χρήστη μέσα από διαφορετικές συσκευές (λογισμικό πελάτη) και καθορισμό προτεραιότητας.

Στην αρχιτεκτονική REST υπάρχει η έννοια της διεύθυνσης η οποία αντιστοιχεί σε ένα πόρο συστήματος με σκοπό κάποιος χρήστης ή εφαρμογή από οποιοδήποτε σημείο και χωρίς κάποια συγκεκριμένη διεύθυνση (application layer) να μπορούν να αποστείλουν ένα αίτημα (HTTP GET) ώστε να επιστραφεί σε αυτούς μία αναπαράσταση του συγκεκριμένου πόρου. Συμπεραίνουμε λοιπόν ότι δεν υπάρχει η δυνατότητα να επικοινωνήσουν δύο οντότητες στον παγκόσμιο ιστό με άμεσο και εύκολα υλοποιήσιμο τρόπο.

Σύμφωνα με τα παραπάνω είναι εύλογο να συμπεράνουμε ότι και οι δύο προσεγγίσεις έχουν γνωρίσματα που λειτουργούν ευεργετικά στην υλοποίηση συστημάτων διαφορετικού είδους .

Η **REST** προσέγγιση είναι ιδανική για υπηρεσίες ιστού στις οποίες απαιτείται αξιοπιστία και αμεσότητα αλλά η μετάδοση σε πραγματικό χρόνο δεν προσφέρει ιδιαίτερο επιπλέον όφελος ενώ η πολυπλοκότητα στο άκρο του πελάτη πρέπει να παραμένει χαμηλή και το φόρτο να τον επομίζεται μόνο ο εξυπηρετητής. Επίσης, σε ότι αφορά απλές υλοποιήσεις η λιτότητα της αρχιτεκτονικής αυτής μπορεί να οδηγήσει σε μεγαλύτερη ταχύτητα απ'ότι το XMPP ενώ το χαρακτηριστικό γνώρισμα της απουσίας κατάστασης στον εξυπηρετητή σημαίνει ότι είναι ευνοϊκή η αύξηση της κλιμάκωσης.

Αντιθέτως, το **XMPP** αποτελεί καθαρή και αξιόπιστη λύση για την δημιουργία συστημάτων όπου η επικοινωνία σε πραγματικό χρόνο είναι καίριας σημασίας, η μοναδική διευθυνσιοδότηση κάθε χρήστη είναι αναγκαία για την μεταξύ τους επαφή, η συχνότητα αμφίδρομης ανταλλαγής περιεχομένου από πληθώρα χρηστών είναι ιδιαίτερα αυξημένη και ειδικές λύσεις άμεσης επικοινωνίας απαιτούνται για την υποστήριξη ενός ευρύτερου συστήματος ή υπηρεσίας όπως αυτό που εξετάζουμε στην συγκεκριμένη διπλωματική εργασία. Το XMPP αναδεικνύεται ως αναγκαία λύση για τέτοια συστήματα καθώς οποιεσδήποτε άλλες RESTful λύσεις σαν αυτές που αναφέρθηκαν παραπάνω τείνουν στην ουσία να προσεγγίσουν την αρχιτεκτονική του XMPP και δείχνουν εμφανώς ότι χρειάζεται να υποστηρίζουν γνωρίσματα όπως η ύπαρξη κατάστασης (statefulness) στον εξυπηρετητή ή

οι προωθούμενες ειδοποιήσεις από αυτόν (push notifications) τα οποία ωστόσο από κάποια άλλη σκοπιά θα μπορούσαν να εκτιμηθούν ως μειονεκτήματα .([3] σελ.28)

Συγκεντρωτικός πίνακας σύγκρισης

	Αρχιτεκτονική REST	**Πρωτόκολλο XMPP**
Ταχύτητα – Λειτουργία μετάδοσης	Συγχρονισμένη επικοινωνία μεταξύ πελάτη και εξυπηρετητή, ανανέωση πληροφορίας με διαδικασία polling, δεν υποστηρίζεται δυνατότητα για άμεση επαφή μεταξύ χρηστών.	Ασύγχρονη και αμφίδρομη επικοινωνία σε πραγματικό χρόνο μέσω TCP socket, ανανέωση πληροφορίας όταν αυτή δημιουργείται, ενσωματωμένη υποδομή για άμεση επικοινωνία μεταξύ χρηστών.
Κοινότητα – Διαθέσιμο υλικό	Ευρέως εφαρμοσμένη τεχνική σχεδιασμού υπηρεσιών ιστού σε επιτυχημένα συστήματα υψηλής κλιμάκωσης με εξασφάλιση αξιόπιστης λειτουργίας. Μεγαλύτερη ωριμότητα.	Εφαρμοσμένο σε συστήματα υπηρεσιών ιστού έχοντας ειδική εφαρμογή στην υλοποίηση του τμήματος διαχείρισης και ενημέρωσης πληροφορίας από τους εξυπηρετητές στους χρήστες σε πραγματικό χρόνο. Καινούρια τεχνολογία.
Επεκτασιμότητα	Πρόκειται για μία απλή μεθοδολογία και τεχνική δημιουργίας υπηρεσιών, συνεπώς δεν περιλαμβάνει δυνατότητες για επιπλέον εξειδικευμένες επεκτάσεις.	Η δημιουργία του πρωτοκόλλου στα θεμέλια της XML δομής μηνυμάτων καθιστά ιδανικά ευνοϊκή τη δημιουργία επεκτάσεων από το χρήστη αλλά και τη χρήση έτοιμων λύσεων.
Ασφάλεια σύνδεσης	Δυνατότητα χρήσης πιστοποιητικών για τους εξυπηρετητές και κρυπτογράφηση καναλιού (HTTPS-SSL/TLS)	Ανώτερο επίπεδο ασφάλειας. Πιστοποίηση χρήστη μέσω SASL και δυνατότητα κρυπτογράφησης σύνδεσης μέσω TLS.

Διευθυνσιοδότηση	Απουσία προδιαγραφής για διευθυνσιοδότηση χρήστη .	Πλήρης και μοναδική διευθυνσιοδότηση χρήστη με δυνατότητα χρήσης ενός λογαριασμού από διαφορετικά σημεία ταυτόχρονα.

2.4.6 Εφαρμογές XMPP στον παγκόσμιο ιστό

Παρά το γεγονός ότι το XMPP δημιουργήθηκε ως πρωτόκολλο το 2002 και η κοινότητα του αναπτύσσεται μόνο την τελευταία δεκαετία υπάρχουν ποικίλες εφαρμογές του στον παγκόσμιο ιστό ορισμένες από τις οποίες γνώρισαν επιτυχία και διάδοση σε ευρύ κοινό. Αναφέρονται παρακάτω οι σημαντικότερες από αυτές, καθώς αποτελούν παράδειγμα της αξιοπιστίας και ταχύτητας που προσφέρει το πρωτόκολλο σε υλοποιήσεις μεγάλης κλίμακας:

- **Facebook Chat**: [29] Η υλοποίηση ενός συστήματος IM μέσω XMPP όπου κάθε χρήστης του Facebook αποτελεί και οντότητα XMPP έχοντας στον κατάλογο επαφών (roster) τους φίλους από το Facebook (Facebook Friends). Το σύστημα αυτό προσφέρει στους χρήστες τη δυνατότητα να το χρησιμοποιούν είτε από το web interface του Facebook ή από οποιοδήποτε πρόγραμμα πελάτη (για υπολογιστή ή πλατφόρμα smartphone) που υποστηρίζει XMPP αρκεί να εισάγει τα διαπιστευτήρια του λογαριασμού του (Facebook account) .
- **Superfeedr**: [30] Αποτελεί την υλοποίηση ενός συστήματος παροχής ειδοποιήσεων πραγματικού χρόνου μέσω XMPP σε τελικούς χρήστες που έχουν δηλώσει τους συνδέσμους (hyperlinks) από κανάλια ενημερώσεων ιστοσελίδων (feeds) που τους ενδιαφέρει να λαμβάνουν. Τα feeds αποτελούν ειδοποιήσεις σε μορφή κειμένου με περιορισμένο μέγεθος από ιστοσελίδες που αποσκοπούν να ενημερώνουν τακτικά το ενδιαφερόμενο κοινό τους. Το superfeedr παρέχει και στις δύο πλευρές (ιστοσελίδες και χρήστες ως εκδότες και συνδρομητές αντίστοιχα) τη δυνατότητα να αποστέλλουν και να λαμβάνουν, αντίστοιχα, το περιεχόμενο που θέλουν συγκεντρωτικά και σε πραγματικό χρόνο.
- **Twitter**: [31] Το Twitter, το οποίο αποτελεί την πιο ανεπτυγμένη πλατφόρμα Micro-blogging (Μίκρο-ιστολόγιο), έχει δώσει τη δυνατότητα σε εφαρμογές και υπηρεσίες να χρησιμοποιούν μία διεπαφή με το API της βασισμένη στο πρωτοκόλλο XMPP. Έτσι, μπορούν να λαμβάνουν ενημερώσεις πολύ μεγάλου πλήθους μαζικά σε

πραγματικό χρόνο καθώς το API του Twitter και το API της εκάστοτε υπηρεσίας αλληλεπιδρούν σαν δύο οντότητες XMPP .

3

Ανάλυση απαιτήσεων

Στην εργασία αυτή υλοποιούμε ένα σύστημα δημοσίευσης περιεχομένου στα κοινωνικά μέσα και παρακολούθησης των αλληλεπιδράσεων της δημοσίευσης. Όλα αυτά συμβαίνουν σε σχεδόν πραγματικό χρόνο ενώ στον πυρήνα της υλοποίησης βρίσκεται μία σημαντική επέκταση του XMPP, η υπηρεσία publish-subscribe XEP-0060. Η επέκταση αυτή προδιαγράφει την αρχιτεκτονική ενός συστήματος ικανού να διαχειριστεί συνεχείς δημοσιεύσεις με σκοπό την προώθησή τους ως ειδοποιήσεις σε κατάλληλους συνδρομητές. Στο σύστημα επιδρούν εξωτερικά ο χρήστης εκδότης μίας δημοσίευσης και ο διαχειριστής χρήστης. Ο χρήστης εκδότης δημοσιεύει ένα νέο μήνυμα σε κόμβο της υπηρεσίας publish-subscribe (δημοσίευσης-συνδρομής) το οποίο στη συνέχεια δημοσιεύεται στο κοινωνικό μέσο δικτύωσης Facebook προκειμένου να επιστραφεί στην υπηρεσία υλικό αλληλεπίδρασης (ως δημοσίευση στον κόμβο αλληλεπίδρασης που αντιστοιχεί στον αρχικό). Βασικό γνώρισμα της παραπάνω διαδικασίας είναι η επικοινωνία όλων των συστατικών μεταξύ τους σε πραγματικό χρόνο μέσω του πρωτοκόλλου XMPP.

3.1 Συστατικά συστήματος και Βασικές λειτουργίες

Τα βασικά **συστατικά μέρη** του συστήματος και οι οντότητες που επιδρούν εξωτερικά σε αυτό είναι τα παρακάτω:

- **Εξυπηρετητής XMPP (XMPP server):** Πρόκειται για τον εξυπηρετητή του πρωτοκόλλου XMPP ο οποίος είναι υπεύθυνος για τη διαχείριση των λογαριασμών χρηστών, την ταυτοποίηση των διαπιστευτηρίων τους (xmpp credentials, δηλαδή, jid και κωδικό πρόσβασης), την καταγραφή της δραστηριότητάς τους και τέλος την διακίνηση των μηνυμάτων XML (stanzas).
- **Υπηρεσία Publish-Subscribe (δημοσίευσης-συνδρομής):** Η υπηρεσία Publish-Subscribe λειτουργεί σαν ένας εξυπηρετητής μηνυμάτων που αφορούν τη δημιουργία και διαχείριση κόμβων αλλά και τη δημοσίευση μηνυμάτων σε κόμβους της. Η διαχείριση των κόμβων περιλαμβάνει τον καθορισμό συνδρομητών και επιτρεπόμενων εκδοτών σε κάθε έναν από αυτούς. Συνδρομητής ορίζεται ο χρήστης ο οποίος έχει εγγραφεί ως παραλήπτης άμεσης ενημέρωσης για κάθε νέο μήνυμα που δημοσιεύεται σε έναν κόμβο ενώ επιτρεπόμενος εκδότης είναι ο χρήστης ο οποίος μπορεί να δημοσιεύσει σε έναν συγκεκριμένο κόμβο.Για να επιτευχθεί η υλοποίηση του μοντέλου Publish-subscribe χρησιμοποιείται μία ενσωματωμένη βάση δεδομένων στην οποία αποθηκεύονται κόμβοι που αναπαριστούν διαφορετικά θέματα ή καμπάνιες και στους οποίους προστίθενται νέες δημοσιεύσεις. Όπως αναφέρθηκε και παραπάνω κάθε κόμβος έχει καθορισμένους συνδρομητές οι οποίοι σε κάθε νέα προσθήκη δημοσίευσης ενημερώνονται σε πραγματικό χρόνο. Στην ίδια βάση αποθηκεύονται και οι κόμβοι στους οποίους δημοσιεύονται οι αλληλεπιδράσεις από τα κοινωνικά μέσα . Σε κάθε κόμβο δημοσίευσης αντιστοιχεί ένας κόμβος αλληλεπίδρασης στον οποίο δημοσιεύει μία εφαρμογή διασύνδεσης με τα κοινωνικά μέσα (social media connector) και στον οποίο είναι συνδρομητής ο αρχικός εκδότης.
- **Εφαρμογή διασύνδεσης με κοινωνικά μέσα (Social media connectors-clients):** Είναι το συστατικό του συστήματος υπεύθυνο για την πρόσβαση στα κοινωνικά μέσα. Τα μηνύματα που προέρχονται από την υπηρεσία Publish-Subscribe προωθούνται και δημοσιεύονται στο κοινωνικό μέσο Facebook ενώ η παρακολούθηση (tracking) των αλληλεπιδράσεων μιας υπάρχουσας δημοσίευσης γίνεται με την επιστροφή αυτών στην υπηρεσία Publish-Subscribe. Η διασύνδεση ενός social media connector με την υπηρεσία Publish-Subscribe γίνεται μέσω του XMPP.
- **Εφαρμογή πελάτη XMPP - Εκκινητής καμπάνιας/εκδότης (XMPP Client – campaign initiator):** Αποτελεί το βασικό εξωτερικό δράστη του συστήματος, δηλαδή, τον χρήστη ο οποίος μέσω του συστήματος δημοσιεύει πληροφορία σε κάποιο κόμβο με σκοπό να φτάσει αυτή στα κοινωνικά μέσα και να δεχθεί στη συνέχεια σαν μηνύματα message stanza ειδοποιήσεις που απεικονίζουν την αλληλεπίδραση που υπήρξε στη συγκεκριμένη δημοσίευση. Αφού συνδεθεί στον εξυπηρετητή XMPP με την ταυτότητα του (JID) αποστέλλει στην διεύθυνση του pubsub component προτυποποιημένα μηνύματα με τα οποία μπορεί να εκτελέσει τις παρακάτω ενέργειες:

1. Σύνδεση στην υπηρεσία Publish-Subscribe αποστέλλωντας μήνυμα connect.
2. Δημιουργία ενός νέου κόμβου δημοσιεύσεων.
3. Κατάργηση ενός υπάρχοντος κόμβου.
4. Δημοσίευση σε έναν από τους υπάρχοντες κόμβους στους οποίους ο συγκεκριμένος χρήστης έχει πρόσβαση είτε επειδή είναι δημιουργός του ή ο διαχειριστής τον έχει ορίσει ως επιτρεπόμενο εκδότη.
5. Αποσύνδεση από την υπηρεσία Publish-Subscribe αποστέλλοντας μήνυμα disconnect.

- **Εφαρμογή πελάτη XMPP-Διαχειριστής (XMPP Admin) :** Αποτελεί εξωτερικό δράστη του συστήματος που δύναται να διαμορφώσει βασικές ρυθμίσεις της υπηρεσίας Publish-Subscribe. Αφού συνδεθεί στον εξυπηρετητή XMPP με την ταυτότητα του (JID) αποστέλλει στην διεύθυνση της υπηρεσίας προτυποποιημένα μηνύματα με τα οποία μπορεί να εκτελέσει τις παρακάτω ενέργειες :
  1. Σύνδεση στην υπηρεσία Publish-Subscribe αποστέλλωντας μήνυμα connect.
  2. Δημιουργία ενός νέου κόμβου δημοσιεύσεων.
  3. Δημιουργία ενός νέου κόμβου αλληλεπιδράσεων.
  4. Κατάργηση ενός υπάρχοντος κόμβου.
  5. Αποσύνδεση από την υπηρεσία Publish-Subscribe αποστέλλοντας μήνυμα disconnect.
  6. Εγγραφή χρήστη με το JID του σε κόμβο ως συνδρομητή.
  7. Εγγραφή χρήστη με το JID του σε όλους τους κόμβους ως συνδρομητή.
  8. Κατάργηση εγγραφής χρήστη με το JID του από κόμβο ως συνδρομητή.
  9. Εγγραφή επιτρεπόμενου εκδότη σε κόμβο.
  10. Κατάργηση εγγραφής επιτρεπόμενου εκδότη από κόμβο.
  11. Προσθήκη συνδρομητή σε whitelist.

Οι βασικές **λειτουργίες** που επιτελεί το σύστημα είναι:

- **Δημοσίευση (Publishing) σε κόμβους της υπηρεσίας Publish-Subscribe:** Ο χρήστης αποκτά πρόσβαση στο σύστημα μέσω ενός συμβατικού προγράμματος πελάτη XMPP και έπειτα δημοσιεύει σε κάποιον κόμβο ένα μήνυμα. Αυτό συνεπάγεται την αποθήκευση του μηνύματος στη βάση δεδομένων ως στοιχείο (notification Item) ενός συγκεκριμένου κόμβου.

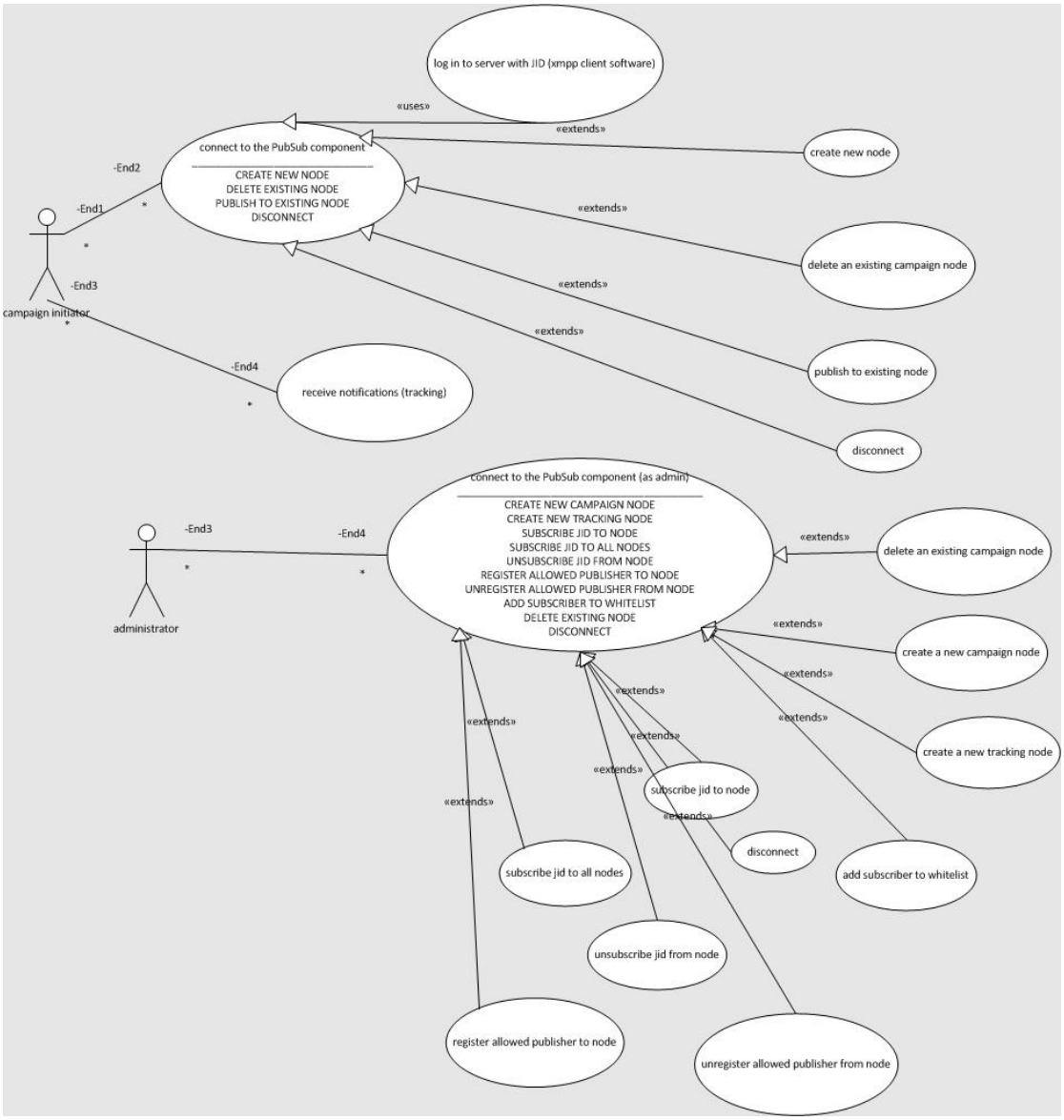
- **Ειδοποίηση των συνδρομητών:** Σε συνέχεια της παραπάνω διαδικασίας ο συγκεκριμένος κόμβος ειδοποιεί τους εγγεγραμμένους συνδρομητές αποστέλλοντας μήνυμα message stanza με xml namespace pubsub event. Από προεπιλογή, το σύστημα καταχωρεί τους social media connectors ως εγγεγραμμένους συνδρομητές σε κάθε κόμβο.
- **Επιστροφή περιεχόμενου αλληλεπίδρασης στον δημιουργό:** Οι social media connectors είναι συγχρόνως clients οι οποίοι ελέγχουν περιοδικά για αλληλεπιδράσεις σε κάθε νέα δημοσίευση και τις επιστρέφουν μέσω μηνυμάτων message stanza στον δημιουργό.

3.2 Διάγραμμα Περίπτωσης Χρήσης (Use Case Diagram)

Το παρακάτω διάγραμμα χρήσης απεικονίζει το σύστημα στο βασικό σενάριο χρήσης καθώς και την δράση των εξωτερικών οντοτήτων σε αυτό, δηλαδή, του δημιουργού μίας δημοσίευσης ή του διαχειριστή χρήστη.

Στην ουσία το σύστημα επιτρέπει την πρόσβαση σε έναν εγγεγραμμένο διαχειριστή ή σε έναν εγγεγραμμένο εκδότη (εκκινητή μίας νέας καμπάνιας). Η αναγνώριση καθενός από τους δύο χρήστες γίνεται με βάση την ταυτότητα XMPP (JID). Το σύστημα αναλόγως του χρήστη που συνδέεται σε αυτό δέχεται συγκεκριμένες εντολές μέσα από μηνύματα XMPP από τα οποία μπορεί να πραγματοποιήσει τις κύριες ενέργειες που προσφέρει το σύστημα, δηλαδή, δημιουργία κόμβου δημοσίευσης και δημοσίευση μηνύματος αλλά και άλλες διαχειριστικές όπως για παράδειγμα εγγραφή σε κόμβο, κατάργηση εγγραφής, κατάργηση κόμβου . Η λίστα διαχειριστών είναι μία προκαθορισμένη λίστα με διευθύνσεις JID ενώ οι επιτρεπόμενοι εκδότες αποθηκεύονται στη βάση δεδομένων καθώς για να υπάρχει η δυνατότητα σύνδεσης καθενός από τους εκδότες στο σύστημα πρέπει πρώτα να έχουν οριστεί ως επιτρεπόμενοι εκδότες σε κάποιο κόμβο τουλάχιστον μία φορά από έναν διαχειριστή.

Εικόνα 5 Διάγραμμα Περίπτωσης Χρήσης (Use Case Diagram – UML 2.0)



3.3 Σενάριο Χρήσης

Όπως αναφέρθηκε και παραπάνω το σύστημα δέχεται τη σύνδεση δύο ειδών χρηστών και επεξεργάζεται ορισμένες προτυποποιημένες εντολές για την εκτέλεση όλων των πιθανών ενεργειών που προσφέρει το σύστημα. Παρακάτω, περιγράφονται το βασικό και εναλλακτικό σενάριο χρήσης στα οποία κατ'αντιστοιχία: α) δημιουργείται από το διαχειριστή ένας νέος κόμβος δημοσίευσης στον οποίο εγγράφει ως εκδότη τον εκκινητή μίας καμπάνιας για να συνδεθεί στη συνέχεια εκείνος και να δημοσιεύσει στον συγκεκριμένο κόμβο ένα μήνυμα και β) συνδέεται ένας εκδότης με τουλάχιστον έναν υπάρχοντα κόμβο στον οποίο του έχει

επιτραπεί η δημοσίευση προκειμένου να δημιουργήσει ο ίδιος ένα νεό ή να δημοσιεύσει στον προαναφερόμενο ένα νέο μήνυμα.

Κείμενο βασικού σεναρίου χρήσης:

1. Ο διαχειριστής του συστήματος συνδέεται από τον προσωπικό του XMPP client με την υπηρεσία Publish-Subscribe προσθέτοντας το στη λίστα επαφών και αποστέλλωντας σε αυτό ένα μήνυμα .
2. Δημιουργεί ένα κόμβο δημοσίευσης αποστέλλοντας ένα μήνυμα δημιουργίας κόμβου. Κατά τη δημιουργία ορίζεται αυτόματα ο client του Facebook connector ως συνδρομητής στο node 1.
3. Δημιουργεί ένα κόμβο αλληλεπίδρασης στην παραπάνω δημοσίευση αποστέλλοντας ένα μήνυμα δημιουργίας κόμβου. Κατά τη δημιουργία ορίζεται αυτόματα ο client του Facebook connector ως επιτρεπόμενος εκδότης στο node 2.
4. Ορίζει τον επιτρεπόμενο εκδότη αποστέλλοντας .
5. Προσθέτει στην whitelist τον εκδότη του συγκεκριμένου κόμβου ώστε να επιτραπεί ο καθορισμός του ως συνδρομητή στον κόμβο αλληλεπίδρασης .
6. Εγγράφει τον εκδότη του node 1 ως συνδρομητή στον κόμβο αλληλεπιδράσεων node 2 .
7. Ο διαχειριστής του συστήματος αποσυνδέεται από την υπηρεσία Publish-Subscribe.
8. Ο εκκινητής μίας καμπάνιας δημοσιεύσεων σε κοινωνικά μέσα συνδέεται από τον προσωπικό του XMPP client με την υπηρεσία Publish-Subscribe προσθέτοντας το στη λίστα επαφών και αποστέλλοντας σε αυτό ένα μήνυμα .
9. Στη συνέχεια δημιουργεί μία νέα δημοσίευση η οποία θα αποτελέσει ένα νέο post στο Facebook account του εκδότη .
10. Το παραπάνω μήνυμα παραλαμβάνεται από την υπηρεσία Publish-Subscribe δημοσιεύεται στον κόμβο node 1 και ενημερώνεται ο Facebook connector client ως συνδρομητής.
11. Στη συνέχεια ο Facebook connector client συνδέεται μέσω του API του Facebook με αυτό και δημιουργεί ένα post στο wall του χρήστη-εκδότη.
12. Ταυτόχρονα ξεκινάει να ελέγχει ανά πολύ μικρά χρονικά διαστήματα εάν υπάρχουν αλληλεπιδράσεις στο παραπάνω Post, δηλαδή, comments ή likes .
13. Για κάθε ένα που λαμβάνει από τα αιτήματα ελέγχου στο api του Facebook το αποστέλλει ως δημοσίευση στο node 2 (αλληλεπιδράσεων) με τη μορφή μηνύματος iq (stanza) τύπου <publish/> .

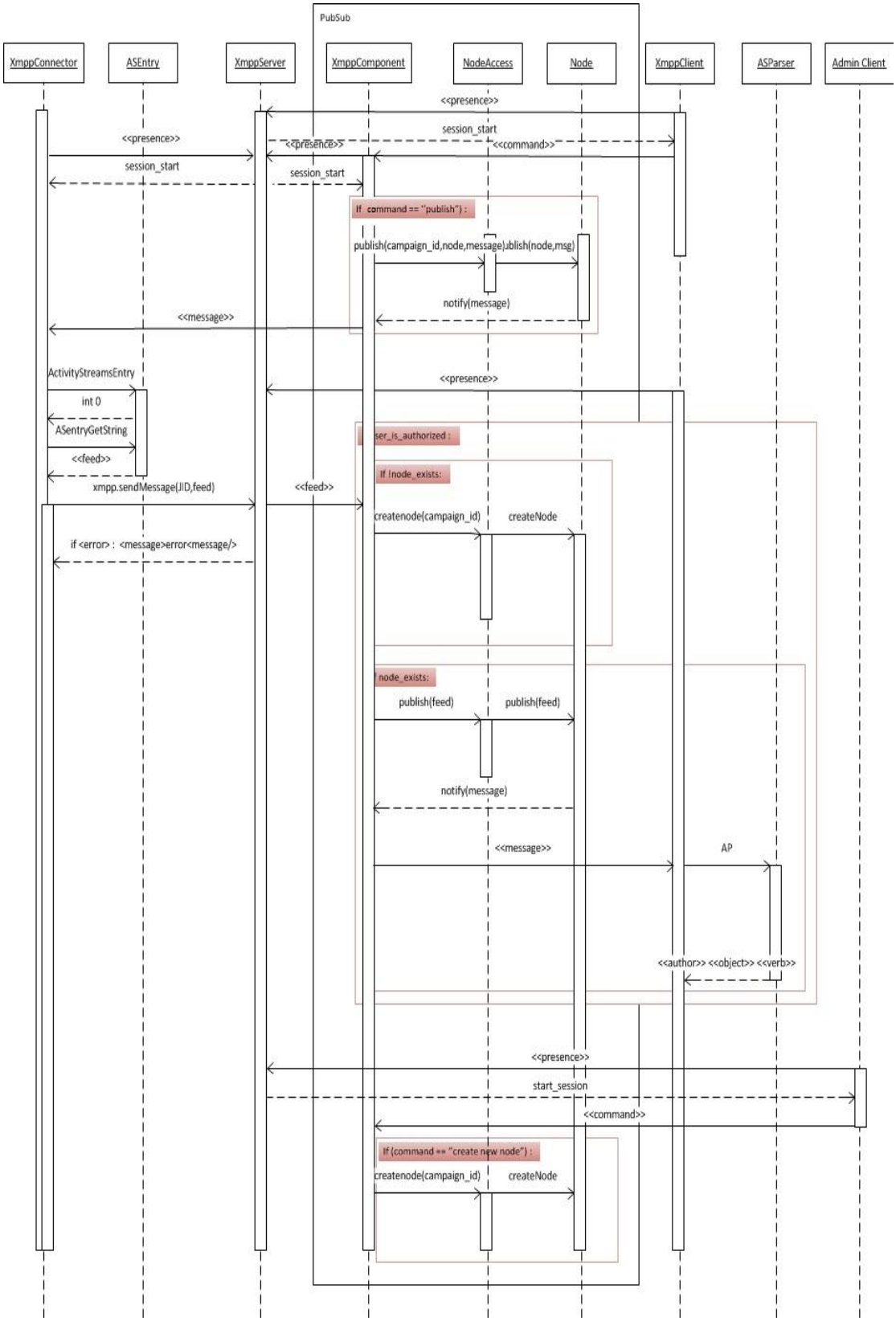
14. Η υπηρεσία Publish-Subscribe ενημερώνει τον συνδρομητή του κόμβου node 2, δηλαδή, τον εκδότη της δημοσίευσης.
15. Ο διαχειριστής αποσυνδέεται από την υπηρεσία Publish-Subscribe.

Εναλλακτικό σενάριο χρήσης κατά το οποίο ο εκδότης είναι ήδη εγγεγραμμένος σαν εκδότης ενός υπάρχοντος κόμβου: Στο σενάριο αυτό ο χρήστης συνδέεται στην υπηρεσία Publish-Subscribe έχοντας τη δυνατότητα να αποστείλει οποιοδήποτε από τα προτυποποιημένα μηνύματα τα οποία αφορούν τον εκδότη.

1. Ο εκκινητής μίας καμπάνιας δημοσιεύσεων σε κοινωνικά μέσα συνδέεται από τον προσωπικό του XMPP client με την υπηρεσία Publish-Subscribe προσθέτοντας το στη λίστα επαφών και αποστέλλοντας σε αυτό ένα μήνυμα .
2. Στο βήμα αυτό ο εκδότης μπορεί είτε να δημοσιεύσει ένα μήνυμα σε υπάρχοντα κόμβο στον οποίο είναι εγγεγραμμένος (βλ. βήματα 3-9) ή να δημιουργήσει ένα νέο κόμβο δημοσιεύσεων με ένα μήνυμα δημιουργίας κόμβου. Έτσι, δημιουργείται αυτόματα ένα νέος κόμβος αλληλεπιδράσεων node 4, ο εκδότης της δημοσίευσης γίνεται επιτρεπόμενος εκδότης του node 3 και συνδρομητής του node 4, ο Facebook connector client γίνεται συνδρομητής του node 3 και επιτρεπόμενος εκδότης του node 4.
3. Στη συνέχεια δημιουργεί μία νέα δημοσίευση η οποία θα αποτελέσει ένα νέο post στο Facebook account του εκδότη .
4. Το παραπάνω μήνυμα παραλαμβάνεται από την υπηρεσία Publish-Subscribe δημοσιεύεται στον κόμβο node 3 και ενημερώνεται ο Facebook connector client ως συνδρομητής.
5. Στη συνέχεια ο Facebook connector client συνδέεται μέσω του API του Facebook με αυτό και δημιουργεί ένα post στο wall του χρήστη-εκδότη.
6. Ταυτόχρονα ξεκινάει να ελέγχει ανά πολύ μικρά χρονικά διαστήματα εάν υπάρχουν αλληλεπιδράσεις στο παραπάνω Post, δηλαδή, comments ή likes .
7. Για κάθε ένα που λαμβάνει από τα αιτήματα ελέγχου στο api του Facebook το αποστέλλει ως δημοσίευση στο node 4 (αλληλεπιδράσεων) με τη μορφή μηνύματος iq (stanza) τύπου <publish/> .
8. Η υπηρεσία Publish-Subscribe ενημερώνει τον συνδρομητή του κόμβου node 4, δηλαδή, τον εκδότη της δημοσίευσης.
9. Ο εκδότης αποσυνδέεται από την υπηρεσία Publish-Subscribe.

3.4 Ακολουθιακό Διάγραμμα

Εικόνα 6 Ακολουθιακό Διάγραμμα (Sequence Diagram – UML 2.0)



Το παραπάνω διάγραμμα αλληλεπίδρασης UML 2.0 σε συνδυασμό με το παραπάνω διάγραμμα χρήσης χρησιμοποιήθηκε για να προδιαγράψει τις λειτουργικές απαιτήσεις του συστήματος. Περιλαμβάνονται τα βασικά συστατικά του συστήματος με τη μορφή αντικειμένων (στιγμιοτύπων κλάσεων) που αλληλεπιδρούν μεταξύ τους είτε με μηνύματα (όπως <<presence>> XML stanza) ή με κλήσεις συγκεκριμένων μεθόδων άλλων αντικειμένων.

Βασικές οντότητες οι οποίες αναφέρονται και στην παράγραφο 3.1 ως συστατικά του συστήματος είναι κατ'αντιστοιχία τα αντικείμενα : XmppServer ως xmpp εξυπηρετητή, XmppComponent ως το component Δημοσίευσης-Συνδρομής, XmppConnector ως την εφαρμογή πελάτη XMPP για σύνδεση με κοινωνικά μέσα , XmppClient ως την XMPP εφαρμογή του εκδότη και AdminClient ως την εφαρμογή XMPP διαχειριστή. Τα υπόλοιπα αποτελούν στιγμιότυπα κλάσεων που λειτουργούν σαν δομικά συστατικά στα πλαίσια της οργάνωσης του κώδικα της υλοποίησης.

4

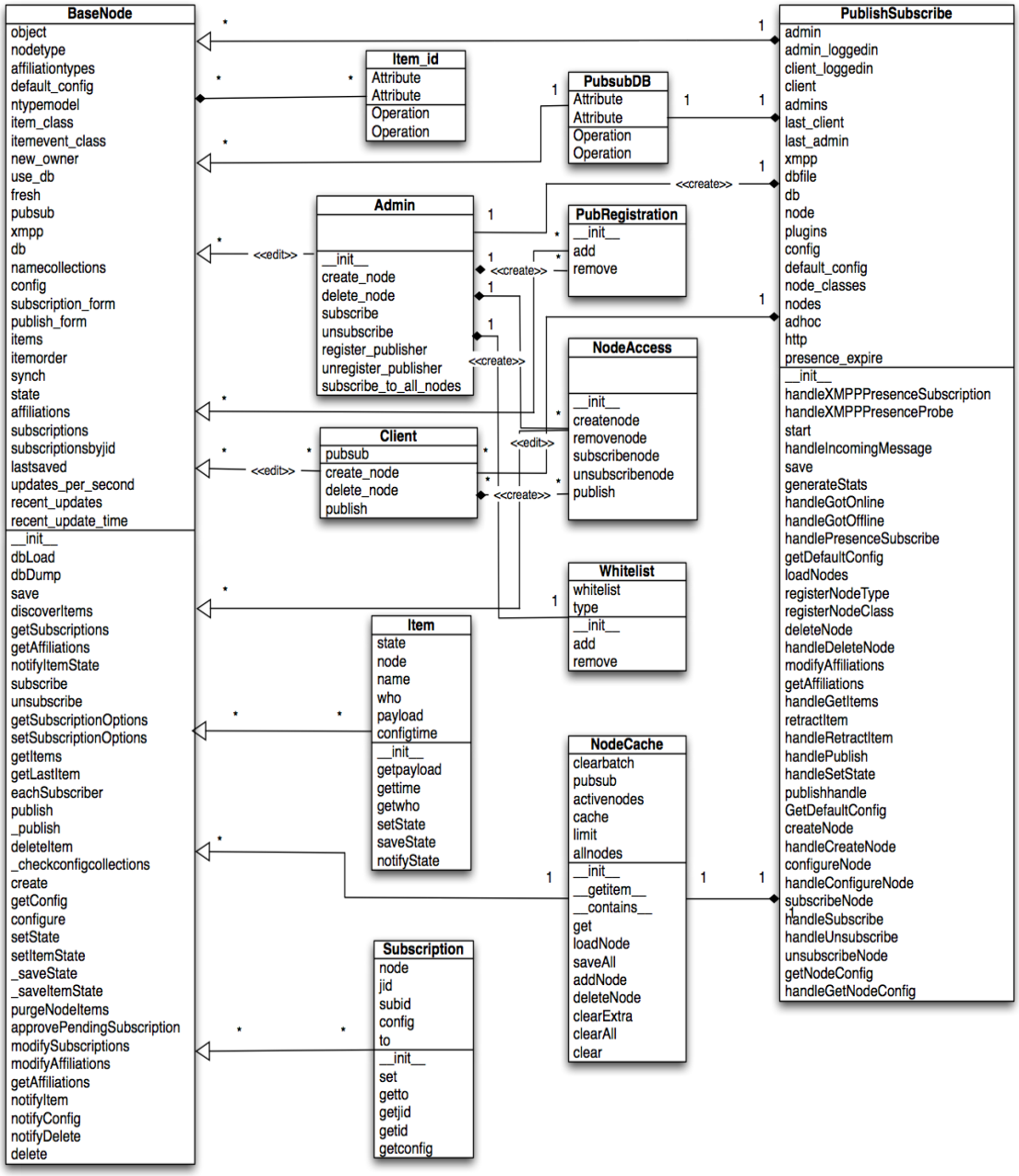
Σχεδίαση συστήματος

Στο κεφάλαιο αυτό περιγράφεται με χρήση της γλώσσας UML η στατική δομή του συστήματος. Βασικά εργαλεία που χρησιμοποιήθηκαν είναι το Διάγραμμα Κλάσης, οι λειτουργίες διαπροσωπείας και το μοντέλο Uniform Data. Το διάγραμμα κλάσης είναι ένα μοντέλο που αφορά την λεπτομερή στατική δομή του λογισμικού που υλοποιήθηκε, οι λειτουργίες διαπροσωπείας περιγράφουν τον τρόπο επικοινωνίας ενός εξωτερικού συστήματος λογισμικού ή δράστη με το σύστημα και το μοντέλο Uniform Data απεικονίζει τις βασικές αναπαραστάσεις δεδομένων του συστήματος.

4.1 Διάγραμμα Κλάσης

Το διάγραμμα κλάσης περιέχει τις κλάσεις του λογισμικού με λεπτομερή καταγραφή των ορισμάτων που δέχονται ως παραμέτρους αλλά και τις μεθόδους που ανήκουν σε αυτές. Απεικονίζονται αναλυτικά οι σχέσεις που συνδέουν τις κλάσεις μεταξύ τους με κύριο διαχωρισμό σε σχέσεις συσσώρευσης (aggregation, δηλαδή, σχέση μεταξύ σύνθετου όλου και συστατικών μερών) και γενίκευσης (generalization,δηλαδή, σχέση μεταξύ γενικής και ειδικής οντότητας).

Εικόνα 7 Διάγραμμα κλάσης



4.2 Λειτουργίες Διαπροσωπείας

Οι λειτουργίες διαπροσωπείας συνοψίζουν τους τρόπους και τις μεθόδους με τις οποίες μπορεί μία εξωτερική οντότητα να αλληλεπιδράσει με το σύστημα. Συγκεκριμένα, η εξωτερική οντότητα μπορεί να είναι μία απλή εφαρμογή πελάτη XMPP αλλά και λογισμικό

που επιδρά άμεσα στο σύστημα που υλοποιήθηκε με κλήση μεθόδων στον πυρήνα του λογισμικού.

Η διαπροσωπεία Publish/Subscribe

Χρησιμοποιώντας τις παρακάτω μεθόδους μπορεί κανείς είτε να επέμβει άμεσα στην υπηρεσί publish-subscribe (δημοσιεύσεις,συνδρομές) ή να στείλει iq stanzas τα οποία θα περιέχουν ως XML στοιχείο παιδί ένα στοιχείο <pubsub/> με το κατάλληλο xml namespace "http://jabber.org/protocol/pubsub" και στη συνέχεια ένα XML στοιχείο παιδί publish/subscribe/unsubscribe τα οποία λαμβάνονται από το component και επεξεργάζονται κατάλληλα το καθένα .

Πίνακας 35 Πίνακας Publish-Subscribe Interface

Μέθοδος (method)	**Όρισμα (Parameter)**	**Επιστρέφει(Returns)**
publish() με τη μέθοδο αυτή ένα XMPP component είναι ικανό να δημοσιεύσει ένα μήνυμα στο pubsub component (αν και εφόσον είναι εξουσιοδοτημένο)	**string** node: μοναδική ταυτότητα (ID) του κόμβου στον οποίο δημοσιεύεται το μήνυμα **string** item: μήνυμα (XML στοιχείο) προς δημοσίευση **int** id: μοναδική ταυτότητα της δημοσίευσης **string** who: μοναδική ταυτότητα του εκδότη (JID)	**node.publish** η μέθοδος αυτή είναι ικανή να προσθέσει ένα νέο μήνυμα (αντικείμενο Item) για ένα συγκεκριμένο κόμβο και να καλέσει μία άλλη μέθοδο υπεύθυνη για την αποστολή της ειδοποίησης δημοσίευσης σε όλους τους συνδρομητές του κόμβου
subscribeNode() με αυτή τη μέθοδο ένα XMPP component ή client δύναται να εγγραφεί ως συνδρομητής σε συγκεκριμένο κόμβο για να λαμβάνει ειδοποιήσεις όταν νέα μηνύματα	**string** node: μοναδική ταυτότητα (ID) του κόμβου στον οποίο δημοσιεύεται το μήνυμα	**nodes[node].subscribe(jid,to)** η μέθοδος αυτή ενημερώνει τις συνδρομές του συγκεκριμένου αντικειμένου του κόμβου (node) προσθέτοντας στον πίνακα της βάσης έναν ακόμη συνδρομητή

(Items) είναι διαθέσιμα	**string** jid: μοναδική ταυτότητα (JID) του συνδρομητή **string** who: περιγραφή του συνδρομητή **string** to: περιγραφή του κόμβου	
unsubscribeNode() με αυτή τη μέθοδο ένας XMPP client/component μπορεί να καταργήσει την συνδρομή του από ένα κόμβο	**string** node: μοναδική ταυτότητα(ID) του κόμβου **string** jid: μοναδική ταυτότητα (JID) του συνδρομητή **string** who περιγραφή του χρήστη **int** subid μοναδική ταυτότητα της συνδρομής	**boolean** επιστρέφει λογική τιμή True εάν ο κόμβος ήδη υπάρχει,διαφορετικά επιστρέφει False

4.3 Uniform Data Model

Το μοντέλο Uniform Data περιγράφει τα κλάσεις και τα ορίσματα αυτών που αποτελούν τις βασικές αναπαραστάσεις δεδομένων στο λογισμικό. Παρακάτω παρατίθεται παράδειγμα του μηνύματος δημοσίευσης XML ως τύπος δεδομένου αλλά και οι ιδιότητες των κλάσεων κόμβου και συνδρομής σε γλώσσα προγραμματισμού Python .

Πίνακας 36 Δεδομένα Uniform Data Model

Data
• PubSub Items • PubSub BaseNode • PubSub Subscriptions

4.3.1 Publish-Subscribe Items (Αντικείμενο που αναπαριστά το μήνυμα προς δημοσίευση)

Πίνακας 37 Πίνακας Publish-Subscribe Items

Name	Description
Node	Η τιμή αυτή τύπου string αναπαριστά τη μοναδική ταυτότητα του κόμβου στον οποίο δημοσιεύεται το μήνυμα
Who	Η μοναδική ταυτότητα (jid: user@domain/resource) του δημιουργού της δημοσίευσης
Name	Η μοναδική ταυτότητα (τιμή τύπου string) της δημοσίευσης
payload	Το μήνυμα που δημοσιεύεται σε μορφή συμβολοσειράς (τιμή τύπου string) περιλαμβάνεται μέσα σε ένα <entry> xml στοιχείο το οποίο περικλείεται από ένα <item> xml στοιχείο
config	Τιμή τύπου String η οποία προσθέτει λειτουργικότητα και ρυθμισιμότητα όπως η ενεργοποίηση της δυνατότητας για «επίμονες» δημοσιεύσεις ,δηλαδή, δημοσιεύσεις οι οποίες αποστέλλονται ως ειδοποιήσεις σε μεταγενέστερη χρονική στιγμή από τη στιγμή δημιουργίας τους σε συνδρομητές που γίνονται αργότερα διαθέσιμοι στο δίκτυο

Παράδειγμα:

Πίνακας 38 Παράδειγμα μηνύματος xmpp iq stanza και ιδιότητες κλάσης Items

xmpp iq stanza	<iq from="user@domain/resource" id="ma019r58" to="subdomain.domain" type="set"> <pubsub xmlns="http://jabber.org/protocol/pubsub">

	<publish node="campaign-node"> <item> <entry xmlns="http://www.w3.org/2005/Atom"> <title>a new item!</title> <summary>publishing!</summary> <id>unique id</id> <published>2000-12-13T18:30:02Z</published> <updated>2000-12-13T18:30:02Z</updated> </entry> </item> </publish> </pubsub> </iq>
Python Class properties	class Item(object): self.node = node self.name = name self.who = who self.payload = payload self.config = config

Μία οντότητα μπορεί να δημοσιεύσει μία δραστηριότητα στον εξυπηρετητή αποστέλλοντας ένα μήνυμα τύπου pubsub <publish> με χαρακτηριστικό τον κόμβο στον οποίο δημοσιεύεται και ένα <entry > στοιχείο παιδί .

Για την ανανέωση (update) ενός αντικειμένου Item η οντότητα στέλνει ξανά ένα μήνυμα τύπου <publish> στο οποίο όμως το στοιχείο <item> έχει ως χαρακτηριστικό το id της αρχικής δημοσίευσης.

Μία οντότητα μπορεί να διαγράψει τη δραστηριότητα που έχει δημοσιευτεί στον εξυπηρετητή στέλνοντας ένα μήνυμα iq με το κατάλληλο namespace (pubsub) και ένα στοιχείο παιδί XML <retract /> το οποίο περιλαμβάνει ένα στοιχείο παιδί <item /> με το item id σαν χαρακτηριστικό .

Παράδειγμα:

Πίνακας 39 Παράδειγμα μηνύματος xmpp iq stanza για διαγραφή στοιχείου από κόμβο

```
<iq
from="user@domain/resource"
id="ma019r58"
to="subdomain.domain"
type="set">
<pubsub xmlns='http://jabber.org/protocol/pubsub'>
  <retract node='urn:xmpp:microblog:0'>
    <item id='urn:uuid:0bfb71a4-d8fd-4410-b119-199c3596f296' />
  </retract>
</pubsub>
</iq>
```

4.3.2 Publish-Subscribe BaseNode (Ενας κόμβος για κάθε μία εκστρατεία στα κοινωνικά μέσα)

Πίνακας 40 Πίνακας Publish-Subscribe BaseNode

Name	Description
Name	Η τιμή αυτή τύπου string αναπαριστά τη μοναδική ταυτότητα του κόμβου τον οποίο αναπαριστά το στιγμιότυπο της κλάσης BaseNode
use_db	Μία Boolean τιμή που ορίζει εάν θα γίνει χρήση της ενσωματωμένης βάσης δεδομένων sqlite3
Db	Ένα στιγμιότυπο της κλάσης DB (database)
Items	Μία λίστα προσωρινά αποθηκευμένων αντικειμένων προερχόμενων από τη βάση δεδομένων και τα οποία αντιστοιχούν στον συγκεκριμένο κόμβο
pubsub	Η κλάση πατέρας του PUBSUB component περνιέται σαν όρισμα προκειμένου να χρησιμοποιηθούν οι βασικές της μέθοδοι
Xmpp	Το αντικείμενο xmpp component (sleekxmpp.Componentxmpp()) υπεύθυνο για τη δημιουργία της σύνδεσης με τον openfire server και τη διαχείριση επικοινωνίας πραγματικού χρόνου μέσω του XMPP πρωτοκόλλου
subscriptions	Μία λίστα από αντικείμενα συνδρομών στον συγκεκριμένο κόμβο χρησιμοποιούμενη σαν προσωρινά αποθηκευμένη λίστα των πιο πρόσφατων

Παράδειγμα:

Πίνακας 41 Ιδιότητες κλάσης BaseNode

Python Class properties	class BaseNode(object): self.use_db = use_db self.pubsub = pubsub self.xmpp = self.pubsub.xmpp self.db = db self.name = name self.items = {} self.subscriptions = {}

4.3.3 PubSub Subscriptions (Ο τρόπος αποθήκευσης-αναπαράστασης των συνδρομών)

Τα παρακάτω είναι ιδιότητες της κλάσης Subscriptions υπεύθυνη για την προσωρινή αποθήκευση των δεδομένων μίας συνδρομής σε ένα στιγμιότυπο ώστε να χρησιμοποιηθεί από τη βάση δεδομένων για προσπέλαση και αποθήκευση (μέθοδοι της κλάσης db) .

Πίνακας 42 Πίνακας Publish-Subscribe Subscriptions

Name	Description
node	Η τιμή αυτή τύπου string αναπαριστά τη μοναδική ταυτότητα του κόμβου στον οποίο ανήκει η συνδρομή
jid	Η μοναδική ταυτότητα (jid: user@domain/resource) του συνδρομητή (client ή component) .
subid	Η μοναδική ταυτότητα της συνδρομής

Παράδειγμα:

Πίνακας 43 Ιδιότητες κλάσης Subscriptions

Python Class properties	class Subscription(object): self.node = node self.jid = jid self.subid = subid

5

Υλοποίηση

Για το υλοποιημένο σύστημα λογισμικού του οποίου οι απαιτήσεις και η σχεδίαση περιγράφηκαν στα κεφάλαια 3 και 4 απαιτήθηκε η χρήση των παρακάτω συστημάτων :

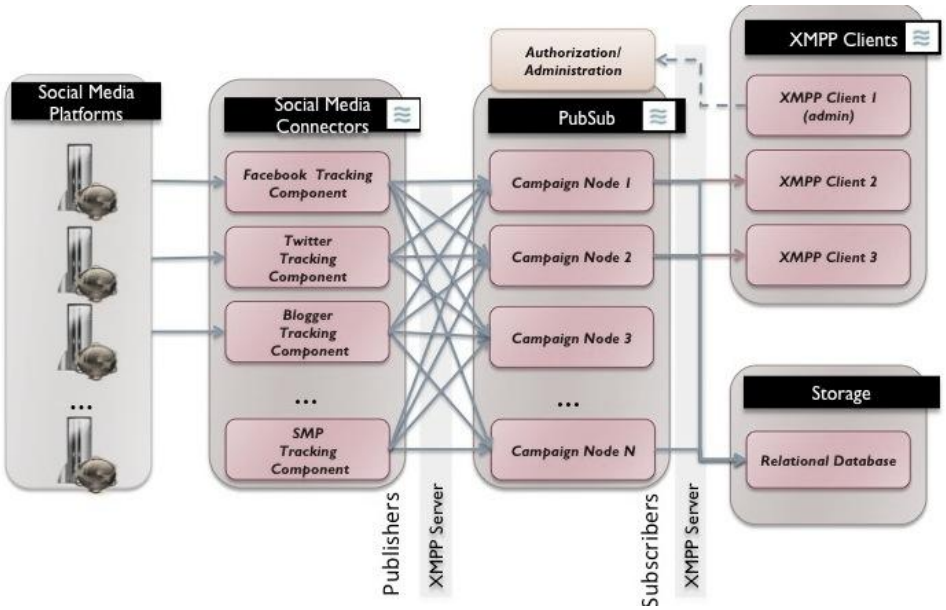
- **Εξυπηρετητής XMPP (XMPP server):** ο εξυπηρετητής XMPP Openfire ο οποίος χρησιμοποιεί μία εξωτερική βάση δεδομένων MySQL για την αποθήκευση των λογαριασμών χρηστών. Είναι υπεύθυνος για τη διαχείριση των λογαριασμών χρηστών, την ταυτοποίηση των διαπιστευτηρίων τους (xmpp credentials, δηλαδή, JID και κωδικό πρόσβασης), την καταγραφή της δραστηριότητάς τους και τέλος την διακίνηση των μηνυμάτων XML (stanzas).
- **Εξωτερικό XMPP component Δημοσίευσης-Συνδρομής (Publish-subcribe external component)**: η υπηρεσία Publish-Subscribe η οποία αποτελεί ένα πακέτο κώδικα το οποίο περιλαμβάνει αφενός όλες τις μεθόδους αναπαράστασης και διαχείρισης των βασικών τύπων δεδομένων που αναφέρθηκαν στο κεφάλαιο 4 και τους οποίους αναλαμβάνει να αποθηκεύσει στην ενσωματωμένη βάση δεδομένων και αφετέρου ένα XMPP component το οποίο είναι ένας σύνθετος XMPP πελάτης. Το XMPP component έχει την ιδιότητα να διατηρεί επίμονη σύνδεση (persistent) με τον εξυπηρετητή Openfire και να επιτελεί αυτοματοποιημένες ενέργειες όπως να διαχειρίζεται μηνύματα message και iq stanza συμβατά με το πρότυπο επέκτασης XEP-0060 [18] και αφετέρου να παρέχει τη δυνατότητα να επιδράσει σε αυτήν ο χρήστης προκειμένου να ρυθμίσει τις παραμέτρους

της ή να δημοσιεύσει σε αυτήν (εκδότης/διαχειριστής). Στην υλοποίησή μας χρησιμοποιείται σαν διεπαφή της υπηρεσίας publish-subscribe η οποία παρέχει δυνατότητα σύνδεσης με το XMPP δίκτυο.

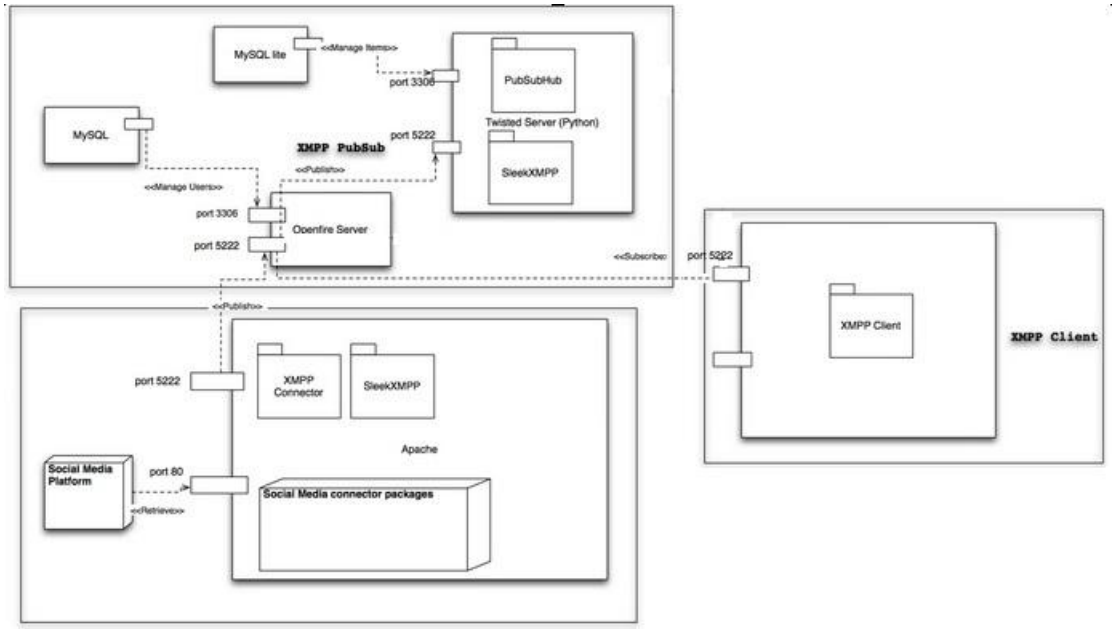
- **Εφαρμογή πελάτη XMPP διασύνδεσης με κοινωνικά μέσα (Social media connectors-clients)**: Ένα ανεξάρτητο εκτελέσιμο πρόγραμμα το οποίο αποτελεί την εφαρμογή πελάτη XMPP και τη διασύνδεση με τα κοινωνικά μέσα και επιτελεί αφενός τη λειτουργία ανταλλαγής μηνυμάτων XML stanza σε πραγματικό χρόνο από και προς το XMPP component που περιέχει η υπηρεσία Publish-Subscribe είτε πρόκειται για δημοσίευση από αυτήν στο Facebook ή για ενημέρωση αλληλεπίδρασης από το Facebook στην υπηρεσία και αφετέρου τη λειτουργία της επικοινωνίας με το API του Facebook όπως περιγράφεται αναλυτικά στην παράγραφο 5.2
- **Εφαρμογή πελάτη XMPP - Εκκινητής καμπάνιας (XMPP Client – campaign initiator) και εφαρμογή πελάτη XMPP-Διαχειριστής (XMPP Admin)** : Ένας συμβατικός XMPP client (Psi, Pidgin,Adium) για την σύνδεση στον εξυπηρετητή με τα διαπιστευτήρια του εκδότη ή διαχειριστή και την προσθήκη στον κατάλογο επαφών του το XMPP component Publish-Subscribe προκειμένου να είναι εφικτή η αποστολή ανάλογων μηνυμάτων. Οι ενέργειες που μπορεί να επιτελέσει ο κάθε χρήστης αναφέρονται στην προδιαγραφή των απαιτήσεων στο κεφάλαιο 3.

Παρακάτω ακολουθούν το σχήμα αναπαράστασης της αρχιτεκτονικής του συστήματος και το ψηφιδικό διάγραμμα της υλοποίησης (Component Diagram UML 2.0).

Εικόνα 8 Αναπαράσταση αρχιτεκτονικής συστήματος



Εικόνα 9 Ψηφιδικό διάγραμμα υλοποίησης (component diagram UML 2.0)



5.1 Βιβλιοθήκες

Η υλοποίηση των συστατικών του συστήματος έγινε με τη γλώσσα προγραμματισμού Python 2.6. Για την δημιουργία των διεπαφών XMPP σε κάθε συστατικό χρησιμοποιήθηκε η ευρέως χρησιμοποιούμενη βιβλιοθήκη SleekXMPP η οποία υποστηρίζει πλήρως την υλοποίηση των προδιαγραφών του πρωτοκόλλου και των επέκτασεων που δέχεται ενώ για την δημιουργία της υπηρεσίας Publish-Subscribe βάση αποτέλεσε το SleekPUBSUB το οποίο είναι μία υλοποίηση της επέκτασης XEP-0060 του XMPP σε γλώσσα Python η οποία στηρίζεται στο SleekXMPP. Για τη διασύνδεση με το Graph API του Facebook χρησιμοποιήθηκε το επίσημο Python SDK που επιτρέπει την πρόσβαση μέσω καθορισμένων μεθόδων στο προφίλ ενός χρήστη . Επειδή υπήρξε διαθέσιμο το παραπάνω υλικό σε γλώσσα Python και η κοινότητα και ο διάλογος στον παγκόσμιο ιστό γύρω από εφαρμογές του XMPP (forums,blogs και documentations) έχει ασχοληθεί ιδιαίτερα με τη συγκεκριμένη γλώσσα επιλέχθηκε αυτή ως γλώσσα υλοποίησης των διαφόρων υποσυστημάτων. Ακόμα, η Python είναι μία γλώσσα που διευκολύνει τη δημιουργία πρωτοτύπων αλλά συγχρόνως υποστηρίζει την ανάπτυξη υψηλής κλίμακας εφαρμογών.

Εξειδικευμένη αναφορά στις βασικές βιβλιοθήκες και συστατικά που χρησιμοποιήθηκαν ακολουθεί παρακάτω:

SleekXMPP (https://github.com/fritzy/SleekXMPP/wiki): Είναι μία ευέλικτη βιβλιοθήκη σχετικά με τα XMPP component/client/server για Python 2.6 και 3+ που παρέχει ευκολία στη χρήση και δέχεται κάθε δυνατή επέκταση XMPP (XEP) σαν αυτόνομο plugin.

Εναλλακτικά μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις προτεινόμενες από την παρακάτω διεύθυνση βιβλιοθήκες http://xmpp.org/xmpp-software/libraries/ προκειμένου να πραγματοποιήσει σύνδεση μέσω XMPP μεταξύ συστημάτων που έχουν υλοποιηθεί σε διαφορετικές γλώσσες προγραμματισμού .

SleekPUBSUB (https://github.com/fritzy/SleekPubsub): Είναι μία ευέλικτη βιβλιοθήκη σχετικά με την υλοποίηση της επέκτασης Publish-Subscribe XEP-0060 [18] του πρωτοκόλλου XMPP για Python 2.6 και 3+ που παρέχει ευκολία στη χρήση και στηρίζεται στη βιβλιοθήκη SleekXMPP. Παρέχει όλες τις βασικές μεθόδους που απαιτούνται για τη δημιουργία και διαχείριση κόμβων δημοσιεύσεων καθώς επίσης και αποθήκευση αυτών σε ενσωματωμένη βάση δεδομένων SQLite.

Openfire Server (http://www.igniterealtime.org/projects/openfire/): Πρόκειται για έναν XMPP εξυπηρετητή γραμμένο σε γλώσσα Java, με ευκολία στην εγκατάσταση και διαχείριση. Προσφέρει αξιοπιστία και απρόσκοπτη λειτουργία και είναι συμβατός με την υλοποίηση μας χωρίς να επηρεάζεται από τη διαφορά γλώσσας προγραμματισμού. Είναι η τελευταία από μια σειρά επιτυχημένων και μαζικά χρησιμοποιημένων εκδόσεων σε ολόκληρο τον κόσμο.

Facebook Python-SDK (https://github.com/facebook/python-sdk): Η επίσημη βιβλιοθήκη ανάπτυξης εφαρμογών που μπορούν να επικοινωνήσουν με το Graph API του Facebook χρησιμοποώντας τη γλώσσα προγραμματισμού Python.

MySQL Server: Είναι ένα ανοικτό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο μπορεί να τρέχει σε ποικίλες πλατφόρμες λογισμικών και αποτελεί εξυπηρετητή παρέχοντας πρόσβαση σε διαφορετικές βάσεις από πολλούς χρήστες.

Είναι μία από τις πιο δημοφιλείς και εύκολες στη χρήση βάσεις και παρέχει δυνατότητα ενσωμάτωσης με σχεδόν οποιαδήποτε γλώσσα προγραμματισμού .

Στην υλοποίηση μας χρησιμοποιείται από τον εξυπηρετητή στο ίδιο μηχάνημα ως βάση αποθήκευσης και διαχείρισης των λογαριασμών των εγγεγραμμένων σε αυτόν χρηστών/component .

SQLite: Είναι μία βιβλιοθήκη λογισμικού γραμμένη σε Python η οποία υλοποιεί μία βάση δεδομένων χωρίς την ανάγκη εγκατάστασης εξυπηρετητή και χωρίς απαίτηση για ρυθμίσεις πέραν της αρχικοποίησής της. Είναι η πιο ευρέως χρησιμοποιημένη βάση στον κόσμο.

Στην υλοποίησή μας χρησιμοποιείται σαν βάση αποθήκευσης όλων των δεδομένων που χρησιμοποιεί το Publish-Subcribe component. Διατηρούνται, δηλαδή, οι πίνακες που αφορούν τους κόμβους, τις συνδρομές, τους εκδότες και τις δημοσιεύσεις.

5.2 Υλοποίηση εφαρμογής διασύνδεσης με το Facebook

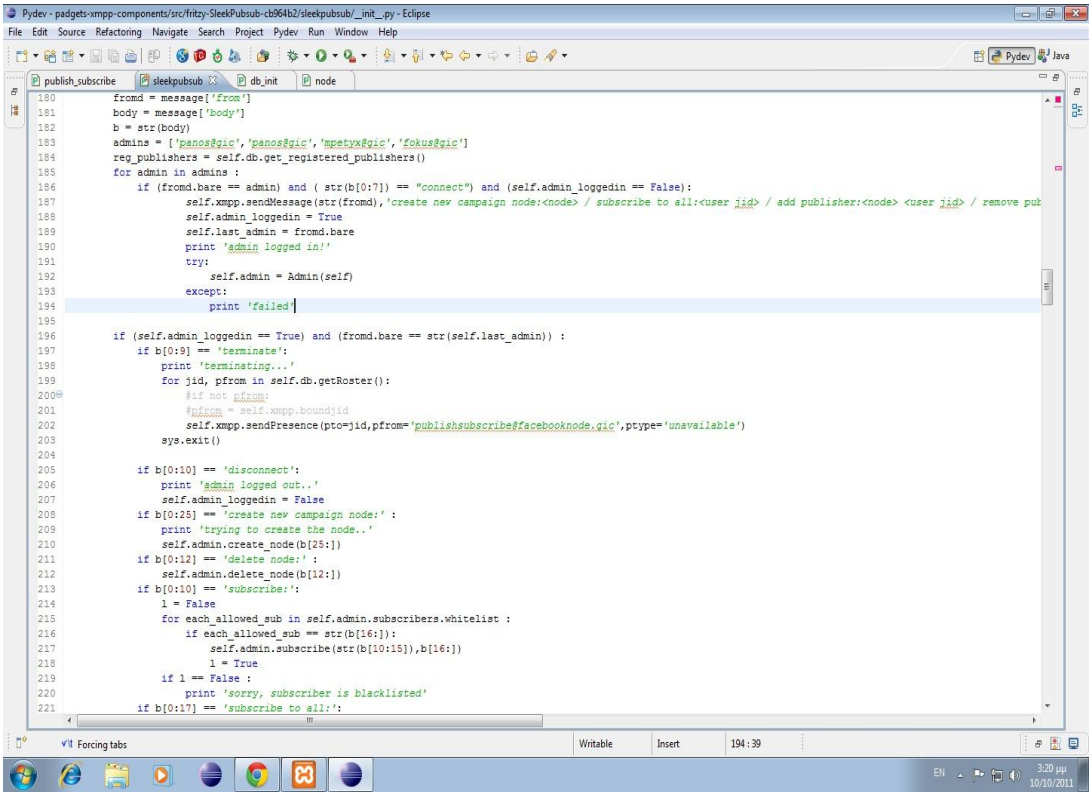
Για τη δημιουργία της εφαρμογής διασύνδεσης με το κοινωνικό μέσο δικτύωσης Facebook υλοποιήθηκε μία εφαρμογή σε γλώσσα προγραμματισμού Python της οποίας η βασική κλάση είναι ένας πελάτης XMPP που χρησιμοποιεί την βιβλιοθήκη sleekxmpp. Κατά την εκκίνηση εκτέλεσης της εφαρμογής καλείται ένα αντικείμενο κλάσης το οποίο είναι υπεύθυνο για την σύνδεση με το API του Facebook. Το συγκεκριμένο API έχει ένα ιδιαίτερα αυστηρό σύστημα πιστοποίησης [36] καθώς υποστηρίζει το πρωτόκολλο OAuth 2.0 [37] . Σύμφωνα με αυτό κάθε εφαρμογή έχει δικό της μοναδικό αναγνωριστικό (APP_ID) και κωδικό (APP_SECRET) με τα οποία μπορεί μέσω μίας «χειραψίας»μέσω μηνυμάτων HTTP με το graph.facebook.com (http handshake) να αποκτήσει ένα αναγνωριστικό της συγκεκριμένης συνεδρίας σύνδεσης γνωστό ως Acces Token προκειμένου να έχει πρόσβαση στο λογαριασμό χρήστη του οποίου τα στοιχεία συμπληρώθηκαν στο login του Facebook. Στο συγκεκριμένο τμήμα κώδικα τρέχει ένας ενσωματωμένος HTTP εξυπηρετητής (HTTP server) ο οποίος δέχεται τις απαντήσεις στα αιτήματα HTTP GET και POST που πραγματοποιούνται για την ολοκλήρωση της προαναφερθείσας διαδικασίας. Στη συνέχεια κάθε μήνυμα που λαμβάνει η βασική κλάση πελάτη XMPP ως συνδρομητής σε κάποιο κόμβο δημοσίευσης, δημοσιεύεται χρησιμοποιώντας μεθόδους του Python Facebook SDK [35] ως post στο wall του χρήστη του οποίου εισήχθηκαν τα στοιχεία λογαριασμού (username και password) και αρχίζει η εκτέλεση της κλάσης παρακολούθησης αλληλεπίδρασης στη δημοσίευση με το συγκεκριμένο αναγνωριστικό (id). Κάθε νέο σχόλιο αλληλεπίδρασης (Facebook like ή comment) αποστέλλεται από τον πελάτη XMPP σαν δημοσίευση στον κόμβο αλληλεπίδρασης που αντιστοιχεί στον αρχικό κόμβο δημοσίευσης.

5.3 Ανάπτυξη

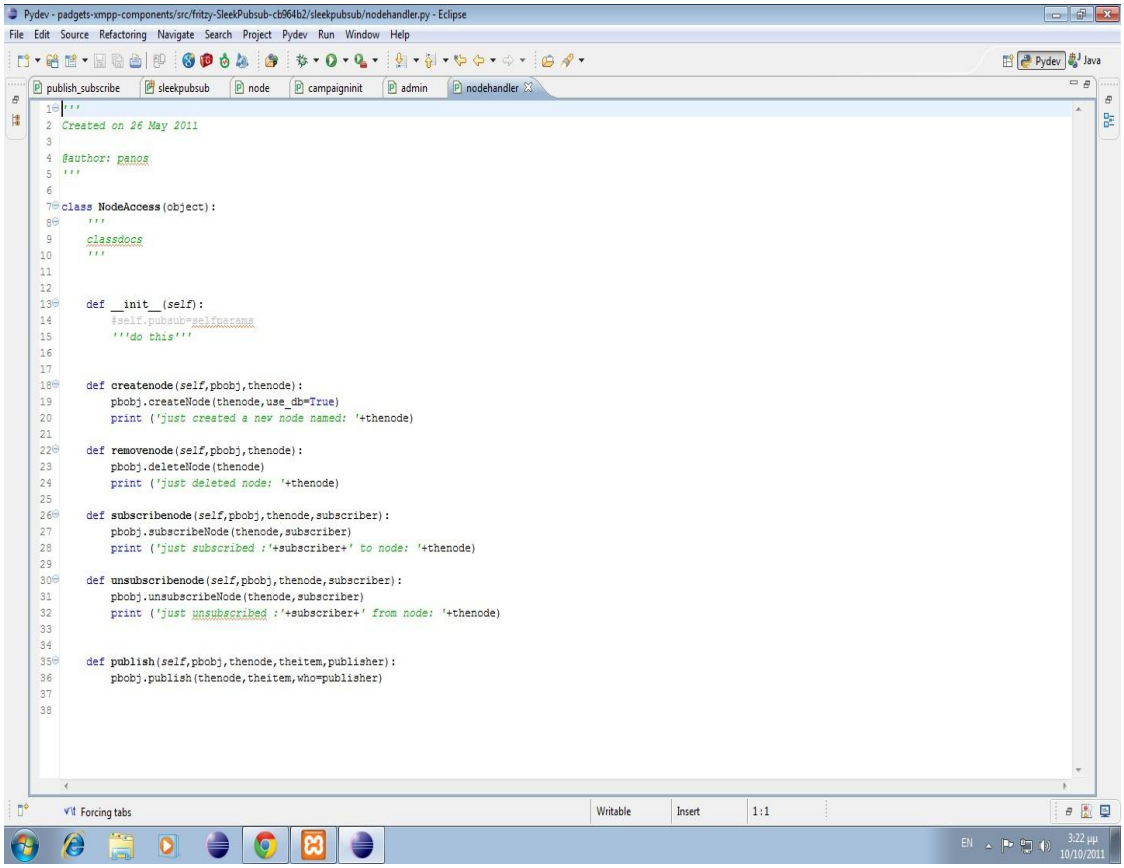
Για την ανάπτυξη του συστήματος σε γλώσσα Python χρησιμοποιήθηκε το IDE PyDev (http://www.pydev.org) το οποίο εγκαταστάθηκε στο framework του Eclipse IDE (http://eclipse.org) .

Παρακάτω φαίνονται ορισμένα **στιγμιότυπα στο Eclipse** από τις βασικές κλάσεις που προστέθηκαν :

Εικόνα 10 Στιγμιότυπα από τμήματα κώδικα στο περιβάλλον Eclipse



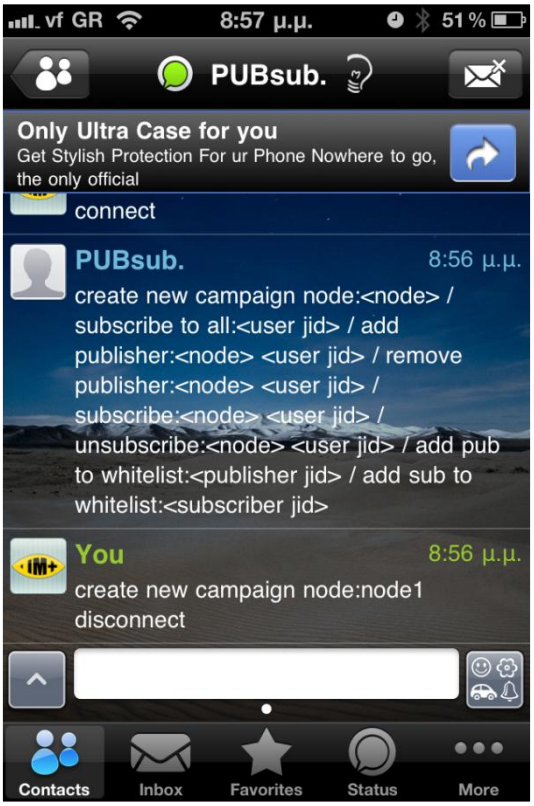
Εικόνα 11 Στιγμιότυπα από τμήματα κώδικα στο περιβάλλον Eclipse



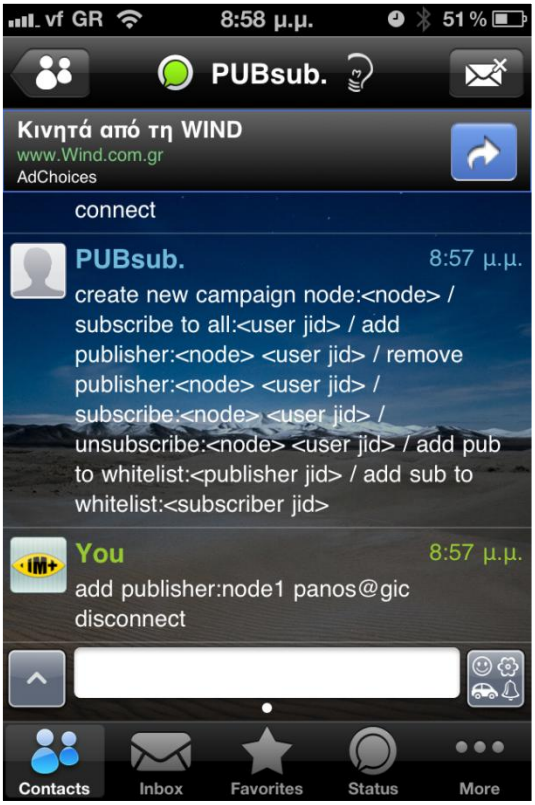
Παρακάτω ακολουθούν στιγμιότυπα από την **χρήση του συστήματος**:

Σύνδεση στο publish subscribe component (με εφαρμογή client XMPP/iOS platform) από το λογαριασμό διαχειριστή και δημιουργία υποτυπώδους κόμβου με σκοπό την εγγραφή του χρήστη ως επιτρεπόμενου εκδότη (<add publisher>).

Εικόνα 12 Σύνδεση στο Pubsub component και δημιουργία νέου κόμβου από το διαχειριστή



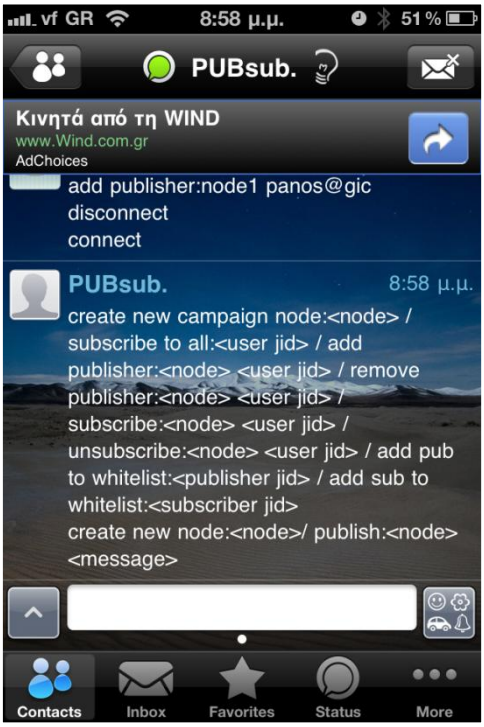
Εικόνα 13 Καθορισμός του χρήστη panos@gic ως επιτρεπόμενου εκδότη στο node1



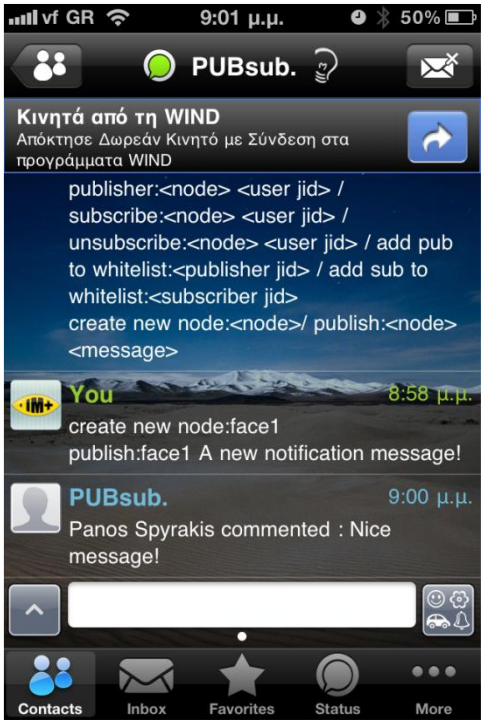
Ύστερα από την αποσύνδεση του χρήστη ως διαχειριστή αποστέλλει ξανά connect προκειμένου να συνδεθεί ως διαχειριστής και εκδότης . Στο βοηθητικό μήνυμα περιέχονται πλέον και οι εντολές : “create new node:<node>” και “publish:<node> <message>”. Με αυτόν τον τρόπο μπορεί να δημιουργήσει ένα νέο κόμβο (<face1>) και αυτόματα να δημιουργηθεί ένας κόμβος αλληλεπίδρασης που θα ενημερωθεί από το Facebook component.

Ο εκδότης δημοσιεύει το μήνυμα «A new notification message!» στον κόμβο face1 και ενημερώνεται ο Facebook connector ο οποίος πραγματοποιεί ένα post στο wall του χρήστη στο Facebook. Αμέσως εμφανίζεται η ενημέρωση στον εκδότη ότι ένας χρήστης έκανε σχόλιο στο αρχικό του Post.

Εικόνα 14 Σύνδεση στο pubsub component από τον χρήστη - εκδότη



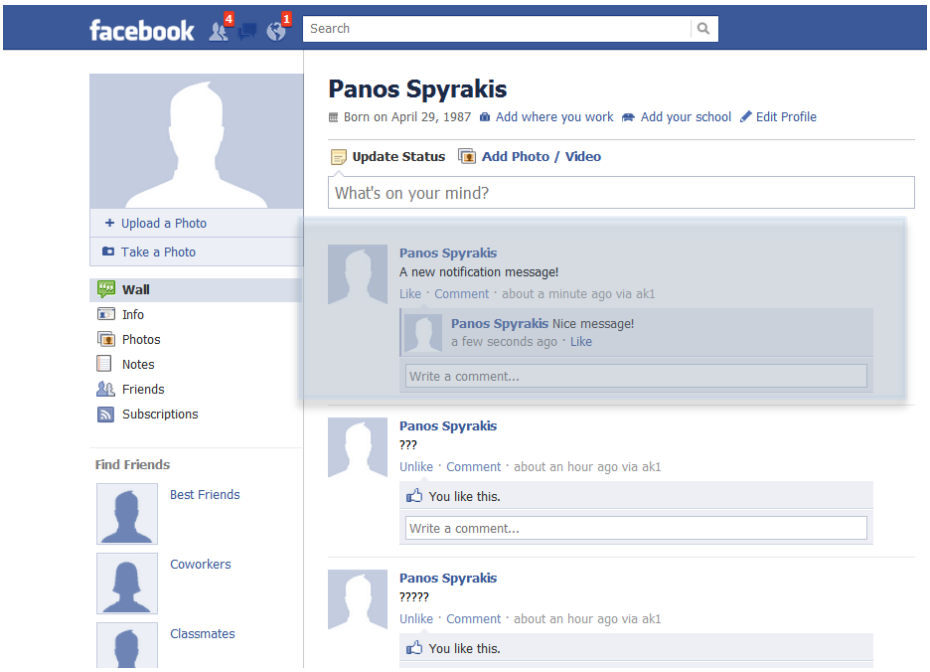
Εικόνα 15 Δημιουργία ενός κόμβου για δημοσίευση (post) στο Facebook και πραγματοποίηση δημοσίευσης. Το σχόλιο στην αρχική δημοσίευση στο Facebook επιστρέφει στον εκδότη σε πραγματικό χρόνο



Ακολουθούν τα στιγμιότυπα που δείχνουν την αλληλεπίδραση στο post του χρήστη στο Facebook.

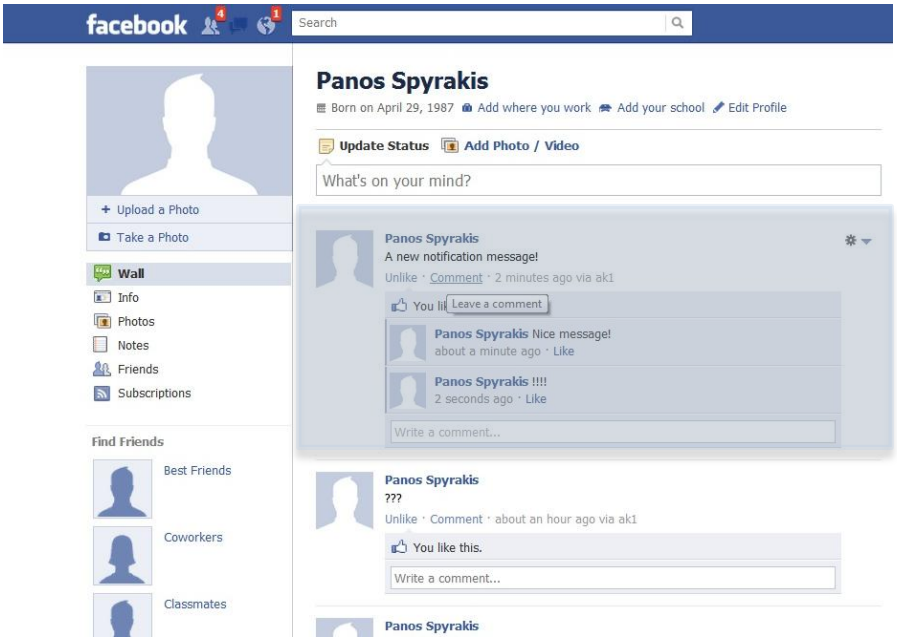
Εικόνα 16 Εμφάνιση δημοσίευσης (post) στο wall του χρήστη στο Facebook

Εικόνα 17 Εμφάνιση σχολίου στην αρχική δημοσίευση που πραγματοποιήθηκε μέσω Pubsub component στο Facebook

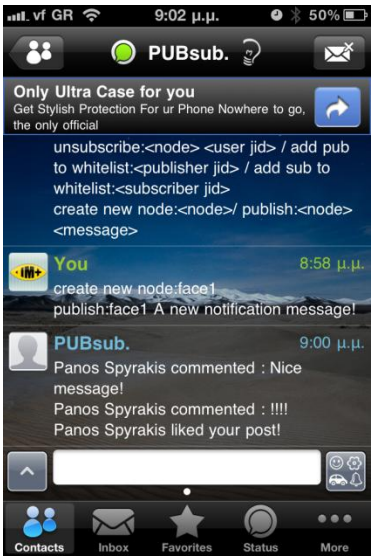


Οποιαδήποτε επόμενη ενέργεια σχολιασμού επιστρέφεται στον εκδότη σαν ειδοποίηση:

Εικόνα 18 Απεικόνιση της δημιουργίας νέου σχολίου και Facebook Like στην αρχική δημοσίευση



Εικόνα 19 Απεικόνιση της ειδοποίησης του εκδότη σε πραγματικό χρόνο για το σχόλιο και για το Facebook Like



5.4 Συμπεράσματα

Ο χρόνος εκπόνησης της εργασίας ήταν δύο εξάμηνα. Στην πρώτη φάση η εργασία ξεκίνησε με τη μελέτη δύο βασικών βιβλίων που αφορούσαν την τεχνολογία του XMPP [2][3] αλλά και την έρευνα σε σχέση με τεχνολογίες που έχουν ως βάση το πρωτόκολλο HTTP. Παράλληλα, πραγματοποιήθηκε αφενός η αναζήτηση πληροφοριών σχετικά με τεχνολογίες υπηρεσιών ιστού που έχουν εφαρμοστεί στα πλαίσια του web 2.0 και με τα δημοφιλέστερα κοινωνικά μέσα δικτύωσης και αφετέρου η εξοικίωση συγκεκριμένα με το API της πλατφόρμας του Facebook. Στη συνέχεια, κατά τη διάρκεια του δευτέρου εξαμήνου, έγινε η προδιαγραφή των αναγκών και απαιτήσεων ενός συστήματος ειδοποιήσεων πραγματικού χρόνου με τη βοήθεια διαγραμμάτων σε γλώσσα UML 2.0 και ακολούθησε η υλοποίηση του συστήματος που περιγράφεται στα κεφάλαια 3, 4 και 5. Τα προαναφερθέντα βιβλία που αφορούν την ταχνολογία XMPP σε συνδυασμό με την αναζήτηση εναλλακτικών λύσεων σε άλλες γλώσσες και βιβλιοθήκες υλοποίησης όπως της Ignite Realtime [32] οδήγησαν στην επιλογή της γλώσσας Python και στην ενσωμάτωση των προαναφερθέντων βιβλιοθηκών ως καταλληλότερες και πιο ευέλικτες για την ανάπτυξη του συστήματος που προδιαγράφηκε. Χρήσιμες στην έρευνα και προετοιμασία της εργασίας υπήρξαν ποικίλες ιστοσελίδες όπως κοινότητες ελεύθερου λογισμικού οι οποίες αφορούσαν άμεσα υλοποιήσεις σε XMPP με χρήση της Python, ιστολόγια ατόμων που έχουν ασχοληθεί σημαντικά με την τεχνολογία του XMPP και έχουν έργο τους διαθέσιμο και βασικοί οργανισμοί όπως ο XSF [28] και ο Ignite

Realtime [32]. Υπήρχαν διαθέσιμα τόσο παραδείγματα χρήσης της τεχνολογίας του XMPP και του Facebook API με κώδικα σε παραπάνω από μία γλώσσες όσο συμβουλές και διάλογος αναφορικά με τις βιβλιοθήκες, τη χρήση συγκεκριμένων κλάσεων, το περιβάλλον ανάπτυξης, τη ρύθμιση του εξυπηρετητή, τη χρήση επεκτάσεων και την χρήση του Facebook SDK. Ωστόσο, στη συγκεκριμένη φάση έγινε εμφανής η διαφορά στην ωριμότητα μεταξύ της φρέσκιας τεχνολογίας XMPP και της παλαιώτερης REST καθώς οι πηγές αναζήτησης στον παγκόσμιο ιστό ήταν ιδιαίτερα περιορισμένες συγκριτικά με τη δεύτερη. Δεν ήταν εύκολη η εύρεση υλικού που να προσεγγίζει συγκεκριμένο απόσπασμα της υλοποίησης και να συνδυάζει τις παραμέτρους που είχαν τεθεί από το σύστημα της εργασίας όπως γλώσσα προγραμματισμού, πακέτο βιβλιοθηκών και επεκτάσεις XMPP.

Βασικές δυσκολίες στην ομαλή ανάπτυξη του συστήματος αποτέλεσαν η παραμετροποίηση του τοίχους ασφαλείας δικτύου (firewall) του λειτουργικού συστήματος (Windows 7 Professional) του μηχανήματος στο οποίο εκτελείται ο εξυπηρετητής XMPP και η υπηρεσία Publish-Subscribe και η κατανόηση και δημιουργία ενός πλήρως λειτουργικού XMPP component. Η πρώτη δυσκολία επιλύθηκε με προσθήκη ενός κανόνα που επιτρέπει κάθε δικτυακή κίνηση σε παραπάνω από μία «πόρτες» (ports : 5222, 5223, 7070, 5060, 7443) αλλά και προσθήκη ακόμη δύο κανόνων που επιτρέπουν την δικτυακή κίνηση στις διεργασίες openfired.exe και openfire-service.exe που αφορούν τη λειτουργία του εξυπηρετητή Openfire. Η δεύτερη δυσκολία υπήρξε εξαιτίας της σχετικά περιορισμένης χρήσης των XMPP components από χρήστες στον παγκόσμιο ιστό με αποτέλεσμα να μην υπάρχει ιδιαίτερο υλικό υποστήριξης. Το βασικό πρόβλημα εντοπίστηκε αφενός στο τμήμα που αφορά τη αυτόνομη διαχείριση του καταλόγου επαφών του component και την ανταλλαγή μηνυμάτων παρουσίας και αφετέρου στη σύνταξη μηνυμάτων XML συμβατών με το πρότυπο επέκτασης Publish-Subscribe XEP-0060.

Είναι σημαντικό να τονιστεί η ανεξαρτησία των υποσυστημάτων ως βασικό χαρακτηριστικό της τεχνολογίας του XMPP καθώς επιτρέπει ελευθερία στην επιλογή γλώσσας και πλατφόρμας λογισμικού για την υλοποίηση καθενός από τα υποσυστήματα που θα αλληλεπιδράσουν μεταξύ τους. Αυτό συμβαίνει καθώς στο δίκτυο XMPP οι οντότητες χρησιμοποιούν για την επικοινωνία μεταξύ τους το πρωτόκολλο ως μέσο οπότε για παράδειγμα είναι εφικτό ο εξυπηρετητής να είναι υλοποιημένος σε γλώσσα JAVA, μία εφαρμογή πελάτη να είναι υλοποιημένη σε Python ενώ μία άλλη εφαρμογή πελάτη να είναι σε web interface (διαπροσωπεία ιστού) σε JavaScript . Στη συγκεκριμένη εργασία όλα τα υποσυστήματα αναπτύχθηκαν σε Python για λόγους ευελιξίας και χρήσης ενός κοινού πλαισίου ανάπτυξης (framework) χωρίς όμως να σημαίνει ότι δεν υπήρξαν εξίσου καλές

εναλλακτικές λύσεις ή ότι δεν υπάρχει ελευθερία στην επεκτασιμότητα ανεξαρτήτως πλατφόρμας λογισμικού .

6

Επίλογος

6.1 Συμπεράσματα

Με την ολοκλήρωση της εργασίας διαπιστώθηκε πραγματικά ότι συστήματα σαν και εκείνο που προδιαγράφηκε στα κεφάλαια 3, 4 και 5 μπορούν να αναπτυχθούν σωστά και να λειτουργούν αξιόπιστα για μεγάλο αριθμό χρηστών και αιτημάτων μόνο με χρήση της αρχιτεκτονικής του πρωτοκόλλου XMPP. Το σύστημα το οποίο αναπτύχθηκε αποτελείται από διαφορετικά συστατικά τα περισσότερα από τα οποία εκτελούνται σε ανεξάρτητα μηχανήματα, δηλαδή, κάθε υποσύστημα διαθέτει τους δικούς του υπολογιστικούς πόρους. Συνεπώς, η αρχιτεκτονική δημιουργεί μία κατανεμημένη δομή των πόρων για εξασφάλιση της εγγυημένης απόδοσης και απρόσκοπτης ανταπόκρισης του συστήματος στις απαιτήσεις. Ωστόσο, βασική πτυχή της λειτουργίας του συστήματος είναι η ασύγχρονη και μαζική ανταλλαγή μηνυμάτων μεταξύ όλων των συστατικών προκειμένου να γίνεται η εξυπηρέτηση των διαδικάσιών αλλά και η επίτευξη αυτής σε πραγματικό χρόνο. Χωρίς, λοιπόν, το πρωτόκολλο XMPP η εξασφάλιση της λειτουργίας δεν θα ήταν εφικτή. Οποιεσδήποτε άλλες εναλλακτικές δεν είναι ικανές να παρέχουν ούτε την αξιοπιστία στην περίπτωση μεγάλης κλιμάκωσης ούτε και το αποτέλεσμα μίας εμπειρίας πραγματικού χρόνου στον χρήστη . Το XMPP, δηλαδή, αποτελεί την ιδανική διαπροσωπεία (interface) για τη δρομολόγηση ολόκληρου του φόρτου επικοινωνίας μεταξύ των υποσυστημάτων.

Στο πλαίσιο της συγκεκριμένης εργασίας ο χρήστης είτε ως διαχειριστής ή ως εκδότης έχει πρόσβαση στο σύστημα μέσα από μία εφαρμογή πελάτη XMPP η οποία εκτελείται είτε σε κάποιο λειτουργικό ηλεκτρονικού υπολογιστή (windows, linux, Mac OS) ή σε λειτουργικό smartphone (Android, iOS, WP7) και μέσω αυτής απολαμβάνει μία εμπειρία ενημέρωσης πραγματικού χρόνου. Ωστόσο, αυτό ήταν μία επιλογή που έγινε στο πλαίσιο των απαιτήσεων της συγκεκριμένης υλοποίησης. Η ελευθερία που προσφέρει το ανοικτό πρωτόκολλο XMPP και η επεκτασιμότητα του προσφέρουν πληθώρα δυνατοτήτων για την επέκταση της εμπειρίας ενός υπάρχοντος συστήματος όπως αυτό που περιγράφηκε στα κεφάλαι 3, 4 και 5.

Χαρακτηριστικό και ενδιαφέρον παράδειγμα αποτελεί η ύπαρξη έτοιμων λύσεων εφαρμογών πελάτη (XMPP client) που εκτελούνται σε περιβάλλον ενός οποιουδήποτε περιηγητή ιστοσελίδων (web browser) . Αυτό οφείλεται στην εναλλακτική δυνατότητα λειτουργίας του XMPP πάνω από το πρωτόκολλο HTTP (XMPP over HTTP) μέσω μίας επέκτασης γνωστής με την ονομασία XMPP over BOSH [34]. Η σημασία της χρήσης τέτοιου είδους εφαρμογών πελάτη είναι ιδιαίτερα μεγάλη καθώς επιτρέπει τη δημιουργία δυναμικού περιεχομένου προερχόμενο για παράδειγμα από σχολιασμούς ή ειδήσεις σε απόκριση πραγματικού χρόνου μέσα σε μία ιστοσελίδα. Έτσι, εμπλουτίζεται η εμπειρία του χρήστη κυρίως σε συνάρτηση με τη χρήση ιστοσελίδων κοινωνικών μέσων δικτύωσης.

Πέραν όμως από την επεκτασιμότητα που προσφέρει στον πυρήνα του το XMPP, ο τρόπος με τον οποίο έχουν δομηθεί τα ανεξάρτητα υποσυστήματα προσφέρει δυνατότητες επέκτασης με την έννοια της προσθήκης επιπλέον συστατικών όπως η εφαρμογή πελάτη ιστού (XMPP web client) ή εφαρμογές διασύνδεσης με άλλα κοινωνικά μέσα δικτύσωσης όπως το Twitter και το Blogger. Έτσι, το μεγάλο πλεονέκτημα της αρχιτεκτονικής του συστήματος είναι ότι μπορεί να δεχθεί άμεσα και να υποστηρίξει τέτοιες εφαρμογές διασύνδεσης οι οποίες μπορούν να αναπτυχθούν σε οποιαδήποτε γλώσσα προγραμματισμού χρησιμοποιώντας το κατάλληλο SDK και έχοντας μία διεπαφή πελάτη XMPP επίσης ανεπτυγμένη σε οποιαδήποτε γλώσσα προγραμματισμού.

6.2 Μελλοντικές προεκτάσεις

Το σύστημα που αναπτύχθηκε στη συγκεκριμένη εργασία είναι ικανό να δεχθεί πληθώρα επεκτάσεων τόσο σε σχέση με την XMPP πτυχή του όσο και με την κατανεμημένη δομή της

αρχιτεκτονικής του όπως αναφέρθηκε στην παράγραφο 6.1 . Κατ' αυτόν τον τρόπο θα μπορούσε να αποτελέσει τη βάση για ποικίλα συστήματα όπως ένα σύστημα ενημερώσεων πραγματικού χρόνου το οποίο να συλλέγει πληροφορία από συγκεκριμένα προφίλ σε διάφορα κοινωνικά μέσα δικτύωσης, να την συγκεντρώνει σε μία βάση δεδομένων και στη συνέχεια να εμφανίζονται σε μία ιστοσελίδα όλες οι αλληλεπιδράσεις άμεσα, σαν ειδοποιήσεις απεσταλμένες από τον εξυπηρετητή χωρίς αίτημα (push notifications). Για παράδειγμα θα μπορούσε μία εταιρία να εμφανίζει στην ιστοσελίδα της "feedback" σε πραγματικό χρόνο δεδομένα από τη δραστηριότητα «φίλων» της από το Facebook στη σελίδα αυτής και παράλληλα αποτελέσματα από δημοσιεύσεις χρηστών στο Twitter σε σχέση με τη συγκεκριμένη εταιρία ή τον χώρο στον οποίο δραστηριοποιείται. Τέλος, θα ήταν ιδιαίτερα ωφέλιμη η χρήση του υλοποιημένου συστήματος ως εργαλείο γεφύρωσης της πληροφορίας από διαφορετικά κοινωνικά μέσα δικτύωσης με ανταλλαγή της πληροφορίας που διαφέρει ή απουσιάζει προκειμένου να υπάρχει κοινή αναφορά αλλά και εμπλουτισμός του περιεχομένου σε όλα τα ευρέως αποδεκτά και χρησιμοποιούμενα μέσα .

7

Βιβλιογραφία - Αναφορές

[1] Architectural Styles and the Design of Network-based Software Architectures by Roy Thomas Fielding,Chapter 5,2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[2] XMPP: The Definitive Guide, by Peter Saint-Andre, Kevin Smith, and Remko Tronçon, 2009

[3] Professional XMPP Programming with JavaScript and jQuery, by Jack Moffitt,2010

[4] IBM technical library,RESTFUL web services:The basics,2008, http://www.ibm.com/developerworks/webservices/library/ws-restful/

[5] Wikipedia, Social Media ,2/10/11, http://www.wikipedia.org/Social_media

[6] Wikipedia, Twitter ,2/10/11, http://www.wikipedia.org/Twitter

[7] Wikipedia, Facebook ,2/10/11, http://www.wikipedia.org/Facebook

[8] Wikipedia,InstantMessaging,15/9/11, http://en.wikipedia.org/wiki/Instant_messaging

[9] Wikipedia,REST,10/8/11, http://en.wikipedia.org/wiki/REST

[10] Wikipedia,SOAP,15/8/11, http://en.wikipedia.org/wiki/SOAP

[11] W3C,Web Services Architecture, http://www.w3.org/TR/ws-arch/

[12] W3Schools,SOAP, http://www.w3schools.com/soap/soap_intro.asp

[13] W3Schools,SOAP, http://www.w3schools.com/soap/soap_syntax.asp

[14] W3Schools,SOAP, http://www.w3schools.com/soap/soap_envelope.asp

[15] W3Schools, http://www.w3schools.com/soap/soap_header.asp

[16] W3Schools, http://www.w3schools.com/soap/soap_example.asp

[17] W3Schools, http://www.w3schools.com/soap/soap_body.asp

[18] XMPP Standards Foundation (XSF), XEP-0060: Publish-Subscribe, http://xmpp.org/extensions/xep-0060.html

[19] Wikipedia, Web Service,25/9/11,http://en.wikipedia.org/wiki/Web_service

[20] XMPP Standards Foundation (XSF),XMPP Technologies:Core,17/7/11, http://xmpp.org/about-xmpp/technology-overview/core/

[21] Gen2Phen Knowledge Center, http://www.gen2phen.org/post/web-service-architecture-simple-rest-vs-complex-soap

[22] Joshua Sachter's Blog,2008, http://joshua.schachter.org/2008/07/beyond-rest.html
[23] FriendFeed Blog,,2008, http://blog.friendfeed.com/2008/08/simple-update-protocol-fetch-updates.html
[24] Roy Fielding's Blog,2008, http://roy.gbiv.com/untangled/2008/paper-tigers-and-hidden-dragons
[25] IETF,Internet Engineering Task Force, http://www.ietf.org/
[26] IETF,[RFC-6120],March 2011 , http://www.ietf.org/rfc/rfc6120.txt
[27] World Wide Web Consortium, http://www.w3c.org
[28] XMPP Standards Foundation (XSF),XEP-0114: Jabber Component Protocol, http://xmpp.org/extensions/xep-0114.html
[29] Facebook Chat, 28/10/2011, http://developers.facebook.com/docs/chat/
[30] Superfeedr, 28/10/2011, http://superfeedr.com/documentation
[31] Twitter Blog, 2008, http://blog.twitter.com/search?q=xmpp
[32] Ignite Realtime, http://www.igniterealtime.org/
[33] Dion Hinchcliffe site,2005,http://hinchcliffe.org/archive/2005/02/12/171.aspx
[34] XMPP Standards Foundation (XSF),XEP-0206: XMPP Over BOSH,http://xmpp.org/extensions/xep-0206.html
[35] Facebook Python SDK, https://github.com/facebook/python-sdk
[36] Facebook Graph API authentication, http://developers.facebook.com/docs/authentication/
[37] OAuth 2.0 Authorization Protocol, http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf
[38] Openfire Server, http://www.igniterealtime.org/projects/openfire/

8

Παραρτήματα

Παράρτημα Α – Οδηγίες εγκατάστασης της υπηρεσίας Publish-Subscribe

Για την εγκατάσταση του συστήματος Publish-Subscribe σε περιβάλλον Windows και την εκτέλεση των βασικών λειτουργιών του, απαιτείται να ακολουθηθούν τα παρακάτω βήματα εγκατάστασης.

Αρχικά, χρειάζεται να επιλέξουμε από το web administration interface του Openfire στη σελίδα *Server Settings* το υπομενού *External Components*, να ενεργοποιήσουμε τη δυνατότητα σύνδεσης στον εξυπηρετητή εξωτερικών XMPP components και να ορίσουμε συγκεκριμένα την ονομασία, τον κωδικό και την πόρτα (port:5060) του Publish-Subscribe component. Αντίστοιχα, τα στοιχεία αυτά πρέπει να ενημερωθούν στο αρχείο *config.txt* στο φάκελο του project *Publish-Subscribe component* .

Στον εξυπηρετητή θα πρέπει να έχουν δημιουργηθεί ακόμα οι λογαριασμοί χρηστών των :

Διαχειριστή, εκδότη και XMPP Social Media Connector.

Ο Διαχειριστής μπορεί να τροποποιηθεί αλλάζοντας τη λίστα *admins=[]* στο αρχείο *__init__.py* που βρίσκεται στο φάκελο *sleekpubsub* του project *Publish-Subscribe component.*

Εγκατάσταση και εκτέλεση Publish-Subscribe component:

- Λήψη και εγκατάσταση της *Python 2.7.2* από την ιστοσελίδα http://python.org/download/
- Προσθήκη της διαδρομής του φακέλου όπου εγκαταστάθηκε η Python (C:\Python27) στις "environment variables" του χρήστη και συγκεκριμένα στη μεταβλητή PATH.
- Λήψη και εγκατάσταση του πακέτου *setuptools* από την ιστοσελίδα http://pypi.python.org/pypi/setuptools .
- Προσθήκη της διαδρομής του αρχείου *easy_install.exe* (C:\Python27\Scripts\easy_install.exe) στις "environment variables" του χρήστη και συγκεκριμένα στη μεταβλητή PATH.
- Εκτέλεση στη γραμμή εντολών (Windows cmd.exe) της εντολής : *easy_install sleekxmpp* για εγκατάσταση του πακέτου SleekXMPP στη βιβλιοθήκη της Python.
- Από τη γραμμή εντολών (Windows cmd.exe) ορίζουμε την τρέχουσα διαδρομή σε αυτή του φακέλου του project *Publish-Subscribe component* και εκτελούμε την εντολή:
  python publish_subscribe.py

Έτσι, αρχίζει η εκτέλεση της υπηρεσίας Publish-Subscribe .

Εγκατάσταση και εκτέλεση XMPP Social media Connector:

- Αρχικά, απαιτείται η δημιουργία μίας νέας εφαρμογής Facebook η οποία συνδέεται με ένα συγκεκριμένο λογαριασμό χρήστη (http://developers.facebook.com/docs/appsonfacebook/tutorial/#create). Η συγκεκριμένη εφαρμογή χαρακτηρίζεται μοναδικά από την ταυτότητα της *APP_ID* και ένα κωδικό *APP_SECRET* . Τα στοιχεία αυτά χρησιμοποιούνται κατά την εκτέλεση της εφαρμογής *XMPP Social Media Connector* όταν επιχειρείται η σύνδεση της με το Facebook Graph API και γίνεται η πιστοποίηση μέσω OAuth προκειμένου να ληφθεί το *ACCES_TOKEN* ως αναγνωριστικό πρόσβασης στη συγκεκριμένη συνεδρία σύνδεσης.
- Παράλληλα, κατά τη διάρκεια απόπειρας σύνδεσης με το Facebook GraphAPI ο χρήστης καλείται να εισάγει σε ένα παράθυρο περιηγητή ιστού τα στοιχεία του λογαριασμού Facebook στον οποίο θα επιδρά ο *XMPP Social media Connector.*
- Για την εκτέλεση του *XMPP Social Media Connector*, από τη γραμμή εντολών (Windows cmd.exe) ορίζουμε την τρέχουσα διαδρομή σε αυτή του φακέλου του και εκτελούμε την εντολή:
  python xmpp_socialmedia_connector.py

Έτσι, εκτελείται ο XMPP Social media Connector και αναμένει μηνύματα δημοσιεύσεων.

Παράρτημα Β – Εγχειρίδιο χρήσης

Για να γίνει ορθή χρήση του συστήματος Publish-Subscribe απαιτείται η εγκατάσταση ενός XMPP client όπως Pidgin, Psi, Adium. Ο χρήστης αφού εγκαταστήσει τον παραπάνω client εισάγει τα στοιχεία του XMPP λογαριασμού που διατηρεί στον XMPP εξυπηρετητή (όνομα, JID, κωδικός και ρυθμίσεις σύνδεσης) και αποκτά σύνδεση με αυτόν. Στην επιλογή προσθήκης φίλου στον κατάλογο επαφών (+add buddy) εισάγει την μοναδική ταυτότητα JID του Publish Subscribe component : publishsubscribe@facebooknode.<server name> και αποδέχεται το αίτημα για ενημέρωση του component με την παρουσία του(presence subscription). Έπειτα το component εμφανίζεται στη λίστα επαφών απ'όπου μπορεί να του αποστείλει προτυποποιημένα μηνύματα για τη διαχείριση ή την δημιουργία δημοσιεύσεων σε κόμβους.

Δημιουργία κόμβων από το διαχειριστή και δημοσίευση μηνύματος στο Publish subscribe component από έναν εκδότη:

1. Ο διαχειριστής του συστήματος συνδέεται από τον προσωπικό του XMPP client με το Publish-Subscribe component προσθέτοντας το στη λίστα επαφών και αποστέλλωντας σε αυτό ένα μήνυμα «connect».
2. Δημιουργεί ένα κόμβο δημοσίευσης αποστέλλοντας ένα μήνυμα «create new campaign node:node1». Κατά τη δημιουργία ορίζεται αυτόματα ο client του Facebook connector ως συνδρομητής στο node 1.
3. Δημιουργεί ένα κόμβο αλληλεπίδρασης στην παραπάνω δημοσίευση αποστέλλοντας ένα μήνυμα «create new tracking node:node2». Κατά τη δημιουργία ορίζεται αυτόματα ο client του Facebook connector ως επιτρεπόμενος εκδότης στο node 2.
4. Ορίζει τον επιτρεπόμενο εκδότη αποστέλλοντας ένα μήνυμα «add publisher:node1 <JID of publisher>»
5. Προσθέτει στην whitelist τον εκδότη του συγκεκριμένου κόμβου ώστε να επιτραπεί ο καθορισμός του ως συνδρομητή στον κόμβο αλληλεπίδρασης : «add sub to whitelist:<JID of publisher>»
6. Εγγράφει τον εκδότη του node 1 ως συνδρομητή στον κόμβο αλληλεπιδράσεων node 2 με το παρακάτω μήνυμα «subscribe:node2 <JID of publisher>»
7. Ο διαχειριστής του συστήματος αποσυνδέεται από το Publish-Subscribe component αποστέλλοντας μήνυμα «disconnect»

8. Ο εκκινητής μίας καμπάνιας δημοσιεύσεων σε κοινωνικά μέσα συνδέεται από τον προσωπικό του XMPP client με το Publish-Subscribe component προσθέτοντας το στη λίστα επαφών και αποστέλλοντας σε αυτό ένα μήνυμα «connect».
9. Στη συνέχεια δημιουργεί μία νέα δημοσίευση η οποία θα αποτελέσει ένα νέο post στο Facebook acoount του εκδότη αποστέλλοντας ένα μήνυμα «publish:node1 <text to be published>»
10. Το παραπάνω μήνυμα παραλαμβάνεται από το Publish-Subscribe component δημοσιεύεται στον κόμβο node 1 και ενημερώνεται ο Facebook connector client ως συνδρομητής.
11. Στη συνέχεια ο Facebook connector client συνδέεται μέσω του API του facebook με αυτό και δημιουργεί ένα post στο wall του χρήστη-εκδότη.
12. Ταυτόχρονα ξεκινάει να ελέγχει ανά πολύ μικρά χρονικά διαστήματα εάν υπάρχουν αλληλεπιδράσεις στο παραπάνω Post, δηλαδή, comments ή likes .
13. Για κάθε ένα που λαμβάνει από τα αιτήματα ελέγχου στο api του Facebook το αποστέλλει ως δημοσίευση στο node 2 (αλληλεπιδράσεων) με τη μορφή μηνύματος iq (stanza) τύπου <publish/> .
14. Το Publish-Subscribe component ενημερώνει τον συνδρομητή του κόμβου node 2, δηλαδή, τον εκδότη της δημοσίευσης.
15. Ο διαχειριστής αποσυνδέεται από το Publish-Subscribe component αποστέλλοντας μήνυμα «disconnect»

Δημιουργία νέων κόμβων και δημοσίευση μηνύματος από έναν εκδότη με ήδη επιτρεπόμενη δυνατότητα έκδοσης σε υπάρχοντα κόμβο:

Στο σενάριο αυτό ο χρήστης συνδέεται στο Publish-Subscribe component έχοντας τη δυνατότητα να αποστείλει οποιοδήποτε από τα προτυποποιημένα μηνύματα τα οποία αφορούν τον εκδότη.

1. Ο εκκινητής μίας καμπάνιας δημοσιεύσεων σε κοινωνικά μέσα συνδέεται από τον προσωπικό του XMPP client με το Publish-Subscribe component προσθέτοντας το στη λίστα επαφών και αποστέλλοντας σε αυτό ένα μήνυμα «connect».
2. Στο βήμα αυτό ο εκδότης μπορεί είτε να δημοσιεύσει ένα μήνυμα σε υπάρχοντα κόμβο στον οποίο είναι εγγεγραμμένος (βλ. βήματα 3-9) ή να δημιουργήσει ένα νέο κόμβο δημοσιεύσεων με ένα μήνυμα δημιουργίας κόμβου «create new node:node3». Έτσι, δημιουργείται αυτόματα ένα νέος κόμβος αλληλεπιδράσεων node 4,ο εκδότης της

δημοσίευσης γίνεται επιτρεπόμενος εκδότης του node 3 και συνδρομητής του node 4, ο Facebook connector client γίνεται συνδρομητής του node 3 και επιτρεπόμενος εκδότης του node 4.

3. Στη συνέχεια δημιουργεί μία νέα δημοσίευση η οποία θα αποτελέσει ένα νέο post στο Facebook account του εκδότη αποστέλλοντας ένα μήνυμα «publish:node3 <text to be published>»
4. Το παραπάνω μήνυμα παραλαμβάνεται από το Publish-Subscribe component δημοσιεύεται στον κόμβο node 3 και ενημερώνεται ο Facebook connector client ως συνδρομητής.
5. Στη συνέχεια ο Facebook connector client συνδέεται μέσω του API του Facebook με αυτό και δημιουργεί ένα post στο wall του χρήστη-εκδότη.
6. Ταυτόχρονα ξεκινάει να ελέγχει ανά πολύ μικρά χρονικά διαστήματα εάν υπάρχουν αλληλεπιδράσεις στο παραπάνω Post, δηλαδή, comments ή likes .
7. Για κάθε ένα που λαμβάνει από τα αιτήματα ελέγχου στο api του Facebook το αποστέλλει ως δημοσίευση στο node 4 (αλληλεπιδράσεων) με τη μορφή μηνύματος iq (stanza) τύπου <publish/> .
8. Το Publish-Subscribe component ενημερώνει τον συνδρομητή του κόμβου node 4, δηλαδή, τον εκδότη της δημοσίευσης.
9. Ο εκδότης αποσυνδέεται από το Publish-Subscribe component αποστέλλοντας μήνυμα «disconnect»

Ευρετήριο εικόνων

Ευρετήριο πινάκων