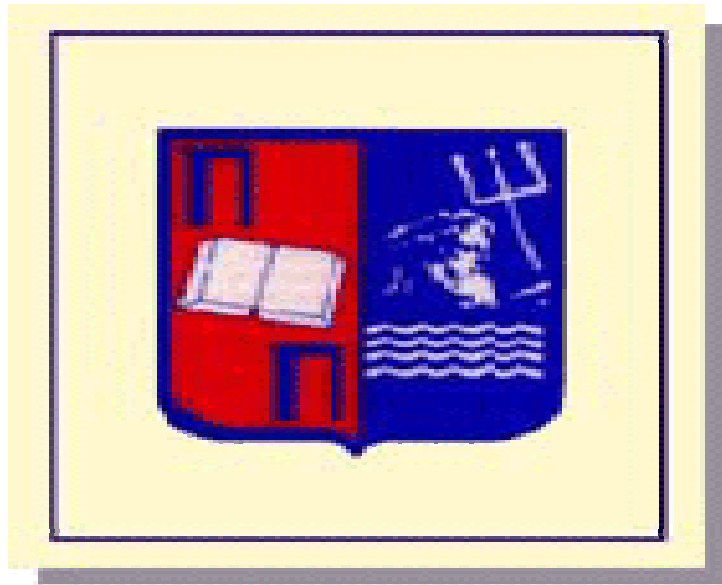


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Ψηφιακών Συστημάτων



**«Διαχείριση κατανεμημένου πολυμεσικού
περιεχομένου με χρήση υπηρεσιοστρεφών
αρχιτεκτονικών »**

ΨΥΧΟΓΥΙΟΣ ΕΥΣΤΑΘΙΟΣ

Η εργασία υποβάλλεται για την μερική κάλυψη των απαιτήσεων με στόχο την
απόκτηση του Μεταπτυχιακού Διπλώματος Σπουδών στα Ψηφιακά Συστήματα
του Πανεπιστήμιο Πειραιώς

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	2
ΠΕΡΙΛΗΨΗ	4
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....	5
ΕΙΣΑΓΩΓΗ	6
ΜΕΡΟΣ ΠΡΩΤΟ	7
ΚΕΦΑΛΑΙΟ 1⁰ - ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ	7
<i>ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΕΙΣΑΓΩΓΗ</i>	<i>7</i>
<i>ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΠΛΕΟΝΕΚΤΗΜΑΤΑ.....</i>	<i>9</i>
<i>ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΠΑΡΑΔΕΙΓΜΑΤΑ.....</i>	<i>10</i>
<i>ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΤΙ ΑΛΛΑΖΕΙ;.....</i>	<i>11</i>
<i>ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΤΟ ΜΕΛΛΟΝ ΤΟΥΣ ΣΤΗΝ ΕΛΛΑΔΑ ..</i>	<i>12</i>
<i>ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΟΡΙΣΜΟΣ</i>	<i>14</i>
<i>ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΜΟΝΤΕΛΟ</i>	<i>15</i>
<i>ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ WEB SERVICES</i>	<i>19</i>
<i>ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΧΡΗΣΗ ΤΩΝ WEB SERVICES</i>	<i>22</i>
<i>ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΣΥΜΠΕΡΑΣΜΑΤΑ</i>	<i>23</i>
ΚΕΦΑΛΑΙΟ 2⁰ – JAVA MEDIA FRAMEWORK.....	24
<i>JMF –JAVA MEDIA FRAMEWORK – ΕΙΣΑΓΩΓΗ</i>	<i>24</i>
<i>JMF –JAVA MEDIA FRAMEWORK – ΕΦΑΡΜΟΓΕΣ.....</i>	<i>25</i>
<i>ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JMF.....</i>	<i>25</i>
<i>ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JMF.....</i>	<i>26</i>
<i>JMF –JAVA MEDIA FRAMEWORK – ΑΡΧΙΤΕΚΤΟΝΙΚΗ</i>	<i>27</i>
<i>ΑΝΑΛΥΣΗ ΣΗΜΑΝΤΙΚΟΤΕΡΩΝ ΟΡΩΝ ΤΟΥ JMF</i>	<i>31</i>
<i>RTP – REAL TIME TRANSFER PROTOCOL.....</i>	<i>34</i>
ΜΕΡΟΣ ΔΕΥΤΕΡΟ	35
ΚΕΦΑΛΑΙΟ 1⁰- ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ	35
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΕΙΣΑΓΩΓΗ.....</i>	<i>35</i>
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ.....</i>	<i>37</i>
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΗ ΟΘΟΝΕΣ</i>	
<i>ΕΦΑΡΜΟΓΗΣ.....</i>	<i>43</i>
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ - ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΗ ΑΝΑΛΥΣΗ</i>	
<i>ΚΛΑΣΕΩΝ.....</i>	<i>51</i>
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΑΝΑΛΥΣΗ ΥΠΗΡΕΣΙΩΝ ΠΑΓΚΟΣΜΙΟΥ</i>	
<i>ΙΣΤΟΥ.....</i>	<i>56</i>
<i>ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΑΝΑΛΥΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ</i>	<i>58</i>
ΜΕΡΟΣ ΤΡΙΤΟ	64
ΚΕΦΑΛΑΙΟ 1⁰ – P2P ΔΙΚΤΥΑ	64
<i>P2P ΔΙΚΤΥΑ ΕΙΣΑΓΩΓΗ</i>	<i>64</i>
<i>ΤΕΧΝΟΛΟΓΙΑ GNUTELLA</i>	<i>65</i>
<i>ΤΕΧΝΟΛΟΓΙΑ FAST TRACK.....</i>	<i>68</i>
<i>ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΑΣ</i>	<i>69</i>

ΚΕΦΑΛΑΙΟ 2 ⁰ – ΔΥΣΚΟΛΙΕΣ ΥΛΟΠΟΙΗΣΗΣ & ΠΡΟΟΠΤΙΚΕΣ ΕΞΕΛΙΞΗΣ..	70
ΔΥΣΚΟΛΙΕΣ – ΠΡΟΒΛΗΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ	70
ΠΡΟΟΠΤΙΚΕΣ ΕΞΕΛΙΞΗΣ ΚΑΙ ΒΕΛΤΙΩΣΗΣ.....	72
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	73
URL's	73
ΣΥΓΓΡΑΜΜΑΤΑ -ΔΗΜΟΣΙΕΥΣΕΙΣ	73

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

ΠΕΡΙΛΗΨΗ

Το παρόν σύγγραμμα αποτελεί το παραδοτέο της πτυχιακής εργασίας του μεταπτυχιακού φοιτητή Ψυχογιού Ευστάθιου και αφορά στην προσπάθεια υλοποίησης ενός συστήματος ανταλλαγής αρχείων πολυμέσων με την χρήση της τεχνολογίας υπηρεσιών παγκόσμιου ιστού.

Λέξεις κλειδιά

1. **Web Services**
2. **SOAP (Simple Object Architecture Protocol)**
3. **Java**
4. **JMF (Java Media FrameWork)**
5. **XML (Extensive Markup Language)**
6. **HTTP (Hyper Text Transfer Protocol)**
7. **P2P (Peer to Peer)**
8. **Apache Tomcat**
9. **My Sql**

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Ακολουθεί κατάλογος με τα σχήματα του παρόντος συγγράμματος:

1. Σχήμα 1.1.1 -Παράδειγμα web services
2. Σχήμα 1.1.2 - Web Service Logos
3. Σχήμα 1.1.3 - Κλήση Web services
4. Σχήμα 1.1.4 –SOAP envelope
5. Σχήμα 1.1.5– Σχεδιάγραμμα λειτουργίας Web Service
6. Σχήμα 1.1.6– Οθόνη WSDL
7. Σχήμα 1.2,1– Οθόνη JMF STUDIO
8. Σχήμα 1.2.2– Αρχιτεκτονική JMF
9. Σχήμα 1.2.3-JMF stack
10. Σχήμα 1.2.4- **Player**
11. Σχήμα 1.2.5- **Processor**
12. Σχήμα 1.2.6– Επεξεργασία σήματος
13. Σχήμα 1.2.7 – **JMF logo**
14. Σχήμα 1.2.8- **Player**
15. Σχήμα 1.2.9- **Processor**
16. Σχήμα 2.1.1– Αρχιτεκτονική υπηρεσίας
17. Σχήμα 2.1.2– Ανταλλαγή αρχείων σε P2P επίπεδο
18. Σχήμα 2.1.3– Διάγραμμα περιπτώσεων χρήσης
19. Σχήμα 2.1.4– Επικοινωνία με βάση Δεδομένων
20. Σχήμα 2.1.5 – Κεντρική οθόνη
21. Σχήμα 2.1.6 – Προβολή μηνυμάτων
22. Σχήμα 2.1.7- Οθόνη προβολής μενού File & Help
23. Σχήμα 2.1.8– Οθόνη προβολής μενού Tools
24. Σχήμα 2.1.9– Οθόνη προσθήκης αρχείων στην βιβλιοθήκη
25. Σχήμα 2.1.10– Οθόνη προσθήκης αρχείων στην βιβλιοθήκη
26. Σχήμα 2.1.11– Οθόνη προβολής της βιβλιοθήκης
27. Σχήμα 2.1.12- Οθόνη τροποποίησης εγγραφής
28. Σχήμα 2.1.13– Οθόνη προβολής του ιστορικού των αρχείων μου
29. Σχήμα 2.1.14 - Οθόνη προβολής network settings
30. Σχήμα 2.1.15-Διάγραμμα Οντοτήτων Συσχετίσεων
31. Σχήμα 2.1.16 -Διάγραμμα Οντοτήτων Συσχετίσεων
32. Σχήμα 2.1.17 – Πίνακες της βάσης
33. Σχήμα 2.1.18 - Πίνακας user
34. Σχήμα 2.1.19- Πίνακας music file
35. Σχήμα 2.1.20- Πίνακας video file

ΕΙΣΑΓΩΓΗ

Στόχος μας είναι η ανάπτυξη μιας ολοκληρωμένης υπηρεσίας ανταλλαγής αρχείων πολυμέσων (**multimedia file exchange service**) –με την χρήση υπηρεσιών παγκόσμιου ιστού (**web services**). Η υπηρεσία αυτή θα πρέπει να υποστηρίζει τόσο **ανταλλαγή αρχείων (file downloading)** όσο και **file streaming** μεταξύ των χρηστών.

Το παρόν σύγγραμμα αποτελεί συνοδευτικό του κώδικα και των αρχείων υλοποίησης της υπηρεσίας μας .

Είναι δομημένο σε 3 μέρη.

- Ø Στο πρώτο μέρος γίνεται μια επισκόπηση των βασικότερων τεχνολογιών που χρησιμοποιήσαμε. Απαρτίζεται από 2 κεφάλαια. Στο πρώτο γίνεται αναφορά στην **τεχνολογία των υπηρεσιών παγκόσμιου ιστού**. Απαρτίζεται από 10 παραγράφους. Επικεντρώνεται κυρίως στην χρήση των υπηρεσιών αυτών στην χώρα μας καθώς και στα πλεονεκτήματα που αποκομίζουμε από αυτήν. Τέλος γίνεται αναφορά στο βασικότερο μοντέλο υλοποίησης των υπηρεσιών αυτών. Στο δεύτερο κεφάλαιο ασχολούμαστε με το **Java Media Framework** – το οποίο και χρησιμοποιούμε για το streaming των αρχείων πολυμέσων. Απαρτίζεται από 7 παραγράφους ενώ γίνεται αναφορά και στο πρωτόκολλο μεταφοράς που χρησιμοποιούμε –**Real Transfer Protocol**.
- Ø Στο δεύτερο μέρος απαρτίζεται επίσης από 2 κεφάλαια. Στο πρώτο γίνεται ανάλυση της αρχιτεκτονικής, της λειτουργίας και των κλάσεων που υλοποιούν την υπηρεσία μας. Απαρτίζεται από 6 παραγράφους. Στο δεύτερο κεφάλαιο παρουσιάζονται οι κυριότερες δυσκολίες που συναντήσαμε για την ολοκλήρωση της παρούσας εργασίας καθώς και οι λύσεις που υιοθετήσαμε.
- Ø Στο τρίτο και τελευταίο μέρος γίνεται μια ανάλυση της τεχνολογίας P2P (Peer to Peer) και γίνεται σύγκριση του συστήματος μας με άλλα ήδη υπάρχοντα. Κλείνοντας γίνεται σύντομη αναφορά στις δυσκολίες που συναντήσαμε καθώς και στις προοπτικές εξέλιξης του συστήματος μας.

ΜΕΡΟΣ ΠΡΩΤΟ

ΚΕΦΑΛΑΙΟ 1^ο - ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΕΙΣΑΓΩΓΗ



Σήμερα το Internet αποτελεί για τις επιχειρήσεις ένα μέσο προβολής και δημοσιοποίησης των δραστηριοτήτων τους. Τα τελευταία χρόνια όμως πολλές επιχειρήσεις παγκοσμίως άρχισαν να χρησιμοποιούν το Internet ως μέσο παροχής υπηρεσιών προς τους πελάτες τους αλλά και προς άλλες επιχειρήσεις.

Με την εξέλιξη αυτή δημιουργήθηκε ένα σύνολο από θέματα που είχαν να κάνουν με την ευχρηστία, τη λειτουργικότητα και την απλότητα των παρεχόμενων υπηρεσιών. Και ενώ οι προσπάθειες που έγιναν για να δοθούν λύσεις ήταν πολλές, το αποτέλεσμα ήταν το ίδιο: υπήρχε ένα μεγάλο σύνολο από έτοιμες υπηρεσίες στο Internet αλλά η χρησιμοποίησή τους ήταν πάρα πολύ μικρή.

Πρόσφατα όμως μία αρχιτεκτονική που ήδη έχει θεσπιστεί και από το world wide web consortium [www.w3c.org], με το όνομα web service, ήρθε να δώσει λύση στα προβλήματα ευχρηστίας και λειτουργικότητας των ηλεκτρονικών υπηρεσιών. Μέσα από την αρχιτεκτονική των web services ορίζεται ένα σύνολο από προδιαγραφές και κανόνες με τους οποίους είναι δυνατή η δημιουργία και η παροχή ηλεκτρονικών υπηρεσιών μέσα στο Internet με αρκετά απλό τρόπο.

Έτσι τα τελευταία τρία χρόνια οι ηλεκτρονικές υπηρεσίες πήραν νέα διάσταση και πλέον κάθε επιχείρηση μπορεί σχετικά εύκολα να δημιουργεί και να παρέχει υπηρεσίες στο Internet αλλά μπορεί με ακόμα μεγαλύτερη ευκολία να χρησιμοποιεί έτοιμες υπηρεσίες.

Η τεχνολογία των web services αφορά συνήθως στους προγραμματιστές εφαρμογών στο Internet και όχι στους απλούς χρήστες και χειριστές ηλεκτρονικών υπολογιστών. Όμως η γνώση της ύπαρξης της συγκεκριμένης τεχνολογίας και του γεγονότος ότι η ενσωμάτωσή της αποτελεί μία σχετικά εύκολη διαδικασία και θα πρέπει να περάσει σε κάθε δραστήριο επιχειρηματία στη χώρα μας. Με αυτό τον τρόπο οι επενδύσεις στην πληροφορική και στο Internet θα είναι πραγματικά αξιόλογες και ανταγωνιστικές τόσο για τα Ελληνικά όσο και για τα παγκόσμια δεδομένα.

Στην ενότητα αυτή θα προσπαθήσουμε να κάνουμε μία συνοπτική αναφορά στα web services και τα τεχνικά χαρακτηριστικά τους. Κατόπιν θα αναφερθούμε σε κάποιες πηγές στο Internet στις οποίες θα μπορέσουμε να βρούμε αρκετά παραδείγματα από web services. Έπειτα θα δούμε πως κάποια ελληνική επιχείρηση θα μπορέσει να παρέχει μία υπηρεσία ως web service αλλά και ποια διαδικασία πρέπει να ακολουθήσει μία άλλη επιχείρηση για να καταφέρει να χρησιμοποιήσει κάποια έτοιμη υπηρεσία από το Internet.

Τέλος θα αναφέρουμε τις τελευταίες εξελίξεις όσο αφορά στην τεχνολογία των web services και τις εκτιμήσεις για τί μπορεί να προσφέρει αυτή η αρχιτεκτονική στον κόσμο των επιχειρήσεων.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΠΛΕΟΝΕΚΤΗΜΑΤΑ

Η αρχιτεκτονική των web services παρέχει αρκετά πλεονεκτήματα τα βασικότερα από τα οποία αναφέρονται παρακάτω:

- Ø Διαλειτουργικότητα. Ένα web service παρέχει ανεξαρτησία τόσο από λειτουργικό σύστημα όσο και από το hardware. Οποιοδήποτε πρόγραμμα που συμβαδίζει με αυτή τη τεχνολογία μπορεί πολύ εύκολα να προσπελάσει μία τέτοια υπηρεσία.
- Ø Ενσωμάτωση. Σε ένα υπάρχον λογισμικό σύστημα που λειτουργεί μέσα στο Internet η δημιουργία ενός web service δεν απαιτεί αλλαγές στον μηχανισμό του συστήματος.
- Ø Διαθεσιμότητα και δημοσίευση. Οι πληροφορίες για τα web services δημοσιεύονται οπότε η εύρεση και η χρήση τους μπορεί να είναι ταχύτατες.
- Ø Επέκταση. Ένα έτοιμο web service είναι δυνατό να ανανεωθεί με εύκολο τρόπο παρέχοντας έτσι επιπρόσθετες υπηρεσίες στους χρήστες του.
- Ø Μικρό κόστος δημιουργίας και χρήσης. Εφόσον σε ένα λογισμικό σύστημα υπάρχει ήδη κάποια διαδικασία που χρειάζεται να επεκταθεί σε on-line υπηρεσία, η δημιουργία του web service κοστίζει ελάχιστα. Επίσης το κόστος ενσωμάτωσης ενός web service σε κάποιο [website](#) ή σε δικτυακή εφαρμογή είναι πάρα πολύ μικρό.

Χρήση λογισμικών συστημάτων. Όλα τα λογισμικά συστήματα και ειδικότερα τα websites που χρησιμοποιούν έτοιμες υπηρεσίες γίνονται πιο λειτουργικά και πιο φιλικά αφού παρέχουν περισσότερες υπηρεσίες στους χρήστες.

ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΠΑΡΑΔΕΙΓΜΑΤΑ

Υπάρχει μία μεγάλη λίστα από έτοιμα web services που θα μπορούσε να χρησιμοποιήσει κανείς, ακόμα και εντελώς δωρεάν.

Ψάχνοντας για παράδειγμα στη διεύθυνση <http://www.webservicelist.com/> μπορούμε να βρούμε μία πληθώρα από web services συνοδευόμενα από το wsdl file καθώς και με on-line demo της λειτουργίας τους.

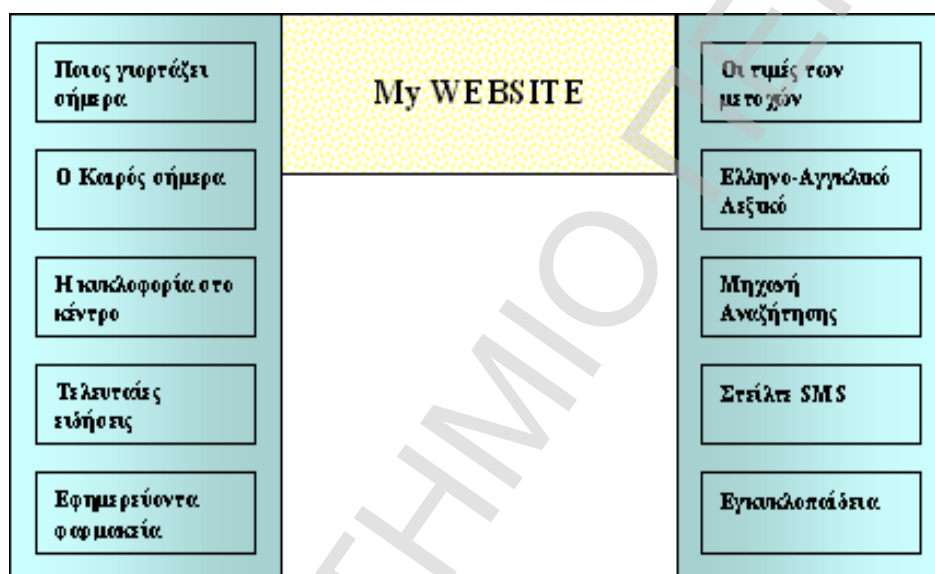
Στον πίνακα που ακολουθεί αναγράφονται κάποια ενδεικτικά όπως τα πήραμε από την παραπάνω διεύθυνση.

Web Service	URL
Country Population Lookup Service Δέχεται το όνομα μίας χώρας και επιστρέφει τον πληθυσμό της.	http://www.cs.uga.edu/~sent
BN Quote Service Δέχεται το ISBN ενός βιβλίου και επιστρέφει την τιμή του.	http://www.xmethods.com/help/addspecs.html
Rich Credit Card Validator Ελέγχει τα στοιχεία μίας πιστωτικής κάρτας αν αντιστοιχούν σε υπάρχουσα πιστωτική.	http://www.richsolutions.com
SOAP SMS Στέλνει κείμενα των χρηστών ως sms σε κινητά.	http://www.redcoal.com
Google Web Service Απαντά σε ερωτήσεις εύρεσης σελίδων και επιστρέφει τα αποτελέσματα σε μορφή επεξεργάσιμη.	http://www.google.com/apis
FreshScore Δίνει τα αποτελέσματα των αγώνων σε πραγματικό χρόνο.	http://www.freshscore.com/service/FreshScoreLiveScores.asmx

ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΤΙ ΑΛΛΑΖΕΙ;

Η λογική της δημιουργίας των πάντων από το μηδέν είναι επιχειρηματικά πλέον ξεπερασμένη. Το κόστος δημιουργίας και συντήρησης μίας υπηρεσίας συνήθως είναι πολύ πιο μεγάλο από το κόστος αγοράς ή ενοικίασης της ίδιας υπηρεσίας από κάποιους τρίτους.

Ακόμα και για τη κατασκευή ενός δυναμικού website η δυνατότητα ενσωμάτωσης μέσα στο site extra πληροφοριών χρησίμων για τους επισκέπτες είναι πλέον πολύ μεγαλύτερη και πιο εύκολη από ποτέ. Αλήθεια ένα website το οποίο να περιέχει τις παρακάτω υπηρεσίες πόσο δύσκολο τελικά είναι να υλοποιηθεί;



Σχήμα 1.1.1 – Παράδειγμα web services

Κάποιοι μπορεί να σκέπτονται πως θα πρέπει να δημιουργήσουν ένα μεγάλο portal από την αρχή και τρομάζουν μόνο με την ιδέα. Κάποιοι άλλοι μπορεί να αναρωτιούνται πως θα μπορέσουν να γνωρίζουν κάθε ημέρα τον καιρό ή τις πτήσεις των αεροπλάνων. Παρόλα αυτά η απάντηση στην πρόκληση για τη δημιουργία ενός τέτοιου website είναι θετική.

Το μόνο που πρέπει να κάνει εδώ ο κατασκευαστής του website είναι να ψάξει για να βρει Internet web services δωρεάν ή και επί πληρωμή που προσφέρουν αυτές τις υπηρεσίες. Κατόπιν θα πρέπει να συμπληρώσει για κάθε κουτάκι στην σελίδα του κάποιες γραμμές εντολών οι οποίες επικοινωνούν με τον αντίστοιχο SOAP server, παίρνουν τις πολύτιμες πληροφορίες και τέλος τις εμφανίζουν με ωραίο, γραφικό τρόπο. Το αποτέλεσμα θα είναι η δημιουργία ενός website με υπηρεσίες οι οποίες φαίνονται στους χρήστες σαν να λειτουργούν στο ίδιο το site.

Αν και τελικά δημιουργήσαμε μία ιστοσελίδα η οποία στην ουσία δημιουργείται τμηματικά από κλήσεις σε ξένες υπηρεσίες, ο τελικός χρήστης θα βλέπει πάντα το καλύτερο δυνατό αποτέλεσμα και εμείς θα έχουμε ξοδέψει ελάχιστο χρόνο και χρήμα κάνοντας ταυτόχρονα την ιστοσελίδα μας πιο ελκυστική.

ΥΠΗΡΕΣΙΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ – ΤΟ ΜΕΛΛΟΝ ΤΟΥΣ ΣΤΗΝ ΕΛΛΑΔΑ

Ενώ με την αρχιτεκτονική των web services αντιμετωπίστηκαν τα περισσότερα προβλήματα συμβατότητας και διαλειτουργικότητας μεταξύ των δικτυακών λογισμικών συστημάτων, υπάρχουν ακόμα αρκετά ανοικτά θέματα που έχουν να κάνουν με την ασφάλεια, την ποιότητα επικοινωνίας και με τον αυτόματο τρόπο αναζήτησης και ενσωμάτωσης έτοιμων υπηρεσιών σε ένα λογισμικό σύστημα.

Έτσι η τεχνολογία των web services εξελίσσεται συνεχώς προτείνοντας και θεσπίζοντας νέα standards πάνω στον τρόπο ανάπτυξης και λειτουργίας των υπηρεσιών δίνοντας μάλιστα ιδιαίτερη έμφαση σε αυτό που οι ξένοι επιστήμονες ονομάζουν «semantic web» δηλαδή στη σημασιολογική ερμηνεία όλων των κόμβων πληροφορίας του παγκόσμιου ιστού.

Καλύπτοντας με αυτόν τον τρόπο και τις απαιτήσεις για την αυτόματη σημασιολογικά ερμηνεία μιας on-line υπηρεσίας από ένα σύστημα η επιστήμη προχωρά σε νέες δομές αυτόματης ανακάλυψης και εκμετάλλευσης των web services.

Ομως αυτή η τεχνολογική εξέλιξη που συμβαίνει στον κόσμο, δεν έχει βρει την ευρεία αποδοχή των Ελληνικών εταιρειών πράγμα που κάνει την χρήση των αμιγώς ελληνικών web services να σπανίζει. Για να επιβεβαιώσει κανείς αυτήν την παρατήρηση μπορεί απλά να δοκιμάσει να αναζητήσει κάποιο Ελληνικό web service ή έστω βιβλιογραφία στα Ελληνικά για αυτήν την αρχιτεκτονική και θα μάλλον θα απογοητευτεί από τα αποτελέσματα.

Υπάρχει λοιπόν ένας λόγος παραπάνω για τις επιχειρήσεις στην χώρα μας να επενδύσουν προς αυτήν την κατεύθυνση. Με αυτό τον τρόπο θα αποδείξουν ότι είναι τόσο ανταγωνιστικές και σύγχρονες όσο και τεχνολογικά ενήμερες για τα τις πλατφόρμες παροχής υπηρεσιών στον κόσμο.

Είναι βέβαιο πως και στην Ελλάδα η λογική της παροχής on-line ηλεκτρονικών υπηρεσιών με την μορφή των web services θα ακολουθήσει την πορεία των υπόλοιπων χωρών του κόσμου. Οι Έλληνες επιχειρηματίες πλέον θα πρέπει να βλέπουν το Internet και ως ένα μέσο προώθησης των υπηρεσιών τους αλλά και εκμετάλλευσης των υπαρχόντων υπηρεσιών.

Ακόμα και στο επίπεδο της κατασκευής ενός website μίας επιχείρησης, θα πρέπει να υπάρχει πάντα μέσα στους κατασκευαστές το «μικρόβιο» της δημιουργίας κάποιας υπηρεσίας αλλά και της χρήσης κάποιας έτοιμης.

Για παράδειγμα όλες οι ξενοδοχειακές επιχειρήσεις θα επιθυμούσαν να αποκτήσουν on-line υπηρεσία κράτησης δωματίων. Αν αυτή η υπηρεσία βρίσκεται μόνο στο website της επιχείρησης τότε η επιχείρηση μειώνει δραματικά τον αριθμό των πιθανών επισκεπτών.

Σε αντίθεση αν παράλληλα με το site ο κατασκευαστής δημιουργήσει και ένα web service με αυτή την υπηρεσία τότε δίνει το δικαίωμα σε όλα τα μεγάλα τουριστικά - ενημερωτικά sites αλλά και συνεργαζόμενες επιχειρήσεις να κλείνουν αυτόματα δωμάτια.

Επειδή λοιπόν όλες οι επενδύσεις γίνονται με σκοπό το άμεσο ή έμμεσο κέρδος, κάθε επιχειρηματίας μπορεί εύκολα να καταλάβει πως η δημιουργία και η χρήση των web services στην ηλεκτρονική του επιχείρηση έχει τόσο άμεσα όσο και έμμεσα οφέλη.

- Ø Χρησιμοποιώντας web services μειώνεται δραματικά ο χρόνος και το κόστος υλοποίησης των websites και γενικότερα των δικτυακών λογισμικών εφαρμογών. Ακόμα οι εφαρμογές γίνονται πιο χρηστικές και πιο ελκυστικές στους χρήστες.
- Ø Δημιουργώντας web services αυξάνεται ραγδαία ο αριθμός των πιθανών πελατών αφού παρέχονται οι υπηρεσίες της επιχείρησης σε όλο τον δικτυακό κόσμο και μπορούν να χρησιμοποιηθούν με πολύ απλό τρόπο.

Σε μία εποχή που προωθούνται οι επιχειρηματικές συνεργασίες και το άνοιγμα της αγοράς μέσα από το Internet, οι Ελληνικές εταιρείες δεν θα πρέπει να αρκестούν σε ένα μικρό βήμα προς τα εκεί. Όσο οι σχεδιασμοί των λογισμικών συστημάτων λαμβάνουν χώρα πάνω στο Internet και όσο το μέλλον της επιχειρηματικής Ελλάδας στρέφεται στον παγκόσμιο ιστό, τα web services πρέπει να αποτελούν βασικά στοιχεία στον σχεδιασμό των προγραμματισμών και απαίτηση από τους επιχειρηματίες.



Σχήμα 1.1.2 - Web Service Logos

Παρακάτω θα προχωρήσουμε σε μια πιο τεχνική ανάλυση της τεχνολογίας των web services αναλύοντας την αρχιτεκτονική, σχεδιασμό, τον τρόπο δημιουργίας και φυσικά των τρόπο λειτουργίας τους.

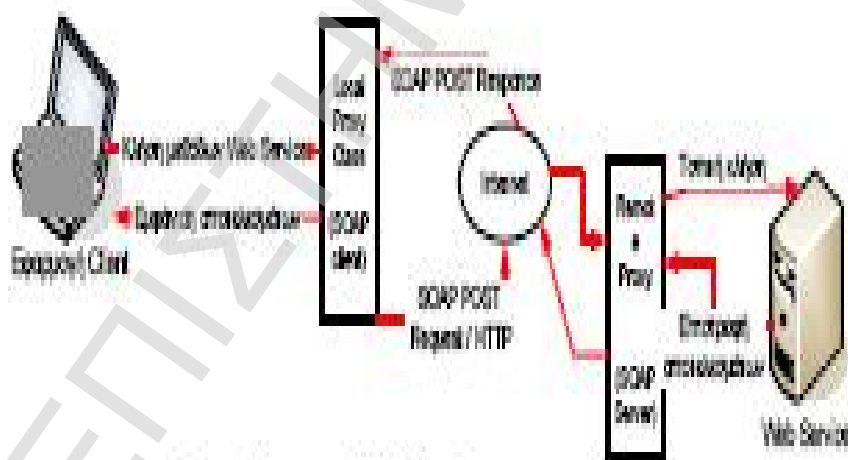
ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΟΡΙΣΜΟΣ

Το Web service είναι ένα λογισμικό σύστημα που αναγνωρίζεται από ένα URI [IETF RFC 2396] και που το περιβάλλον διεπαφής (interface) του καθώς και οι δράσεις του ορίζονται πλήρως και περιγράφονται σε eXtensible Markup Language (XML) μορφή.

Το web service μπορεί να βρεθεί και να χρησιμοποιηθεί εύκολα από άλλα λογισμικά συστήματα. Αυτά τα συστήματα μπορούν να αλληλεπιδρούν με το web service χρησιμοποιώντας υποχρεωτικά τρόπους επικοινωνίας μέσω του Internet και αλλάζοντας πληροφορίες σε μορφή XML.

Παρόλο που ο ορισμός φαίνεται αρκετά γενικός και δεν περιορίζει τη χρήση των web services με συγκεκριμένα πρωτόκολλα, μετά από μία κοινή αποδοχή των μεγαλύτερων εταιρειών λογισμικού στον κόσμο (Microsoft, IBM, Sun κ.α.) της αρχιτεκτονικής αυτής, έχει πλέον καθοριστεί ένα πιο συγκεκριμένο μοντέλο σύμφωνα με το οποίο θα πρέπει οι εταιρείες να παράγουν και να χρησιμοποιούν τα web services.

Δηλαδή στην ουσία ως υπηρεσία παγκόσμιου ιστού ορίζουμε την διαδικασία ανταλλαγής μηνυμάτων κωδικοποιημένων σε XML πάνω από κάποιο ήδη υπάρχων πρωτόκολλο του διαδικτύου. Παρακάτω αναλύεται το μοντέλο αυτό ανταλλαγής μηνυμάτων.



Σχήμα: Κλήση των Web Services

Σχήμα 1.1.3 - Κλήση Web services

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΜΟΝΤΕΛΟ

Υπάρχουν αρκετά προτεινόμενα μοντέλα και αρχιτεκτονικές για την υλοποίηση ενός web service. Το πιο συνηθισμένο από αυτά – που χρησιμοποιήθηκε για την ανάπτυξη του συστήματός μας -προδιαγράφεται ως εξής:

Για την επικοινωνία χρησιμοποιείται συνήθως το πρωτόκολλο HTTP, το ίδιο δηλαδή πρωτόκολλο που χρησιμοποιούν και οι κοινοί browsers για την πλοήγηση στο Internet. Πιο απλά τα δεδομένα μεταφέρονται όπως ακριβώς μεταφέρονται και οι ιστοσελίδες μέσω ενός HTTP stream.

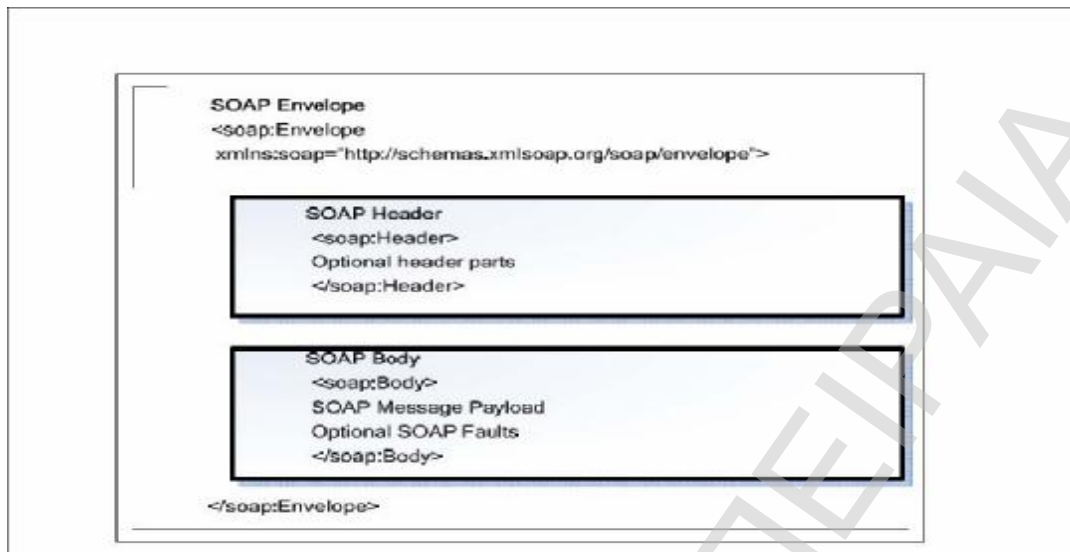
Βασιζόμενο πάνω στο HTTP πρωτόκολλο (όχι αναγκαστικά) χρησιμοποιείται ένα άλλο πρωτόκολλο που ονομάζεται SOAP (Simple Object Access Protocol). Με τη χρήση του πρωτοκόλλου αυτού υλοποιείται ένας εξυπηρετητής (server) ο SOAP server. Για κάθε μέθοδο της υπηρεσίας που θέλουμε να προσφέρουμε, ορίζουμε μία αντιστοιχία με μία λειτουργία του SOAP Server. Κατόπιν, το σύνολο των λειτουργιών του SOAP Server καθώς και τα υπόλοιπα χαρακτηριστικά του, περιγράφονται σε ένα αρχείο που ονομάζεται WSDL (Web Service Description Language) file και το οποίο δημοσιεύεται στο Internet έτσι ώστε να δίνεται η δυνατότητα σε κάθε ενδιαφερόμενο χρήστη να μπορεί άμεσα να χρησιμοποιήσει την υπηρεσία.

Το SOAP (Simple Object Access Protocol) είναι επίσης ένα άλλο standard της W3C [[W3C Simple Object Access Protocol \(SOAP\)](#)] και χρησιμοποιείται αρκετά με στόχο την αποστολή απλών αντικειμένων (αρχείων, εφαρμογών, κλπ.) σε XML μορφή. Γι' αυτό και κάθε web service που χρησιμοποιεί SOAP μπορεί να λάβει αιτήσεις για συγκεκριμένες λειτουργίες απλά δεχόμενο αντικείμενα σε XML. Η χρήση του SOAP πρωτοκόλλου γίνεται συνήθως πάνω από το πρωτόκολλο HTTP αλλά μπορεί να λειτουργήσει και με άλλα πρωτόκολλα (π.χ. FTP, SMTP, κα). Η ιδέα για την δημιουργία ενός πρωτοκόλλου όπως το SOAP ξεκίνησε στα τέλη της δεκαετίας του 90, και συγκεκριμένα το 1997. Τότε μεγάλες εταιρείες, όπως η Microsoft, άρχισαν να διερευνούν κατά πόσο ο κατακευματισμένος υπολογισμός μπορεί να βασιστεί στη γλώσσα XML. Ο σκοπός της έρευνας αυτής ήταν να γίνει εφικτή η επικοινωνία μεταξύ των εφαρμογών μέσω απομακρυσμένων κλήσεων διαδικασιών (Remote Procedure Calls – RPCs), χρησιμοποιώντας απλά πρωτόκολλα δικτύου, όπως το HTTP.

Το σημαντικό λοιπόν είναι ότι καθώς η έρευνα ξεκίνησε από τον κατακευματισμένο υπολογισμό, το SOAP είναι το πλέον κατάλληλο πρωτόκολλο αφού ειδικεύεται σε τέτοια περιβάλλοντα. Το SOAP παρέχει τα κάτωθι χαρακτηριστικά και μηχανισμούς:

- Ø **Μηχανισμός για τον ορισμό της πληροφορίας στην επικοινωνία** – Στο SOAP, όλη η πληροφορία είναι καταχωρημένη σε ένα ξεκάθαρο και ταυτοποιήσιμο SOAP μήνυμα (SOAP message). Αυτό γίνεται μέσω ενός SOAP φακέλου (SOAP envelope), που περικλείει όλες τις απαραίτητες πληροφορίες. Ένα μήνυμα SOAP, μπορεί να έχει σώμα (body), το οποίο περιέχει πληροφορία σε XML δομή. Επίσης, μπορεί να διαθέτει έναν αριθμό από επικεφαλίδες (headers), οι οποίες ενσωματώνουν επιπλέον πληροφορίες έξω από το σώμα του μηνύματος.
- Ø **Διεργασιακό μοντέλο** – Αυτό το μοντέλο ορίζει ένα σύνολο κανόνων βάσει των οποίων γίνονται οι διαπραγματεύσεις μεταξύ των SOAP μηνυμάτων και του λογισμικού. Διακρίνεται από την απλότητά του.
- Ø **Μηχανισμός για αντιμετώπιση σφαλμάτων** – Το SOAP παρέχει το μηχανισμό αυτό με τη μορφή των SOAP faults, τα οποία όταν χρησιμοποιούνται μπορεί να προσδιοριστεί η πηγή που προκάλεσε το σφάλμα. Επιπλέον, παρέχουν τη δυνατότητα ανταλλαγής διαγνωστικών πληροφοριών μεταξύ των μελών της επικοινωνίας.
- Ø **Μοντέλο επεκτασιμότητας** – Το μοντέλο αυτό χρησιμοποιεί SOAP επικεφαλίδες για την υλοποίηση προεκτάσεων πάνω στο SOAP. Οι επικεφαλίδες περιέχουν κομμάτια από δεδομένα με δυνατότητα επέκτασης, τα οποία ταξιδεύουν μαζί με το μήνυμα και μπορούν να γίνουν στόχος για επέκταση σε συγκεκριμένους κόμβους του δικτύου.
- Ø **Ευέλικτος μηχανισμός για αναπαράσταση δεδομένων** – Ο μηχανισμός αυτός επιτρέπει στα δεδομένα σε οποιαδήποτε σειριακή μορφή κι αν βρίσκονται να αναπαρασταθούν σε XML μορφή.
- Ø **Σύμβαση για αναπαράσταση των RPCs και των απαντήσεών τους σαν SOAP μηνύματα** – Οι απομακρυσμένες κλήσεις διαδικασιών είναι αρκετά διαδεδομένες στον κατανεμημένο προγραμματισμό/ υπολογισμό και μπορούν να αναπαρασταθούν καλά μέσω του μηχανισμού αυτού σαν SOAP μηνύματα.
- Ø **Πρωτόκολλο εγκαθίδρυσης σύνδεσης** – Το πρωτόκολλο αυτό ορίζει μια αρχιτεκτονική για την οικοδόμηση συνδέσεων επικοινωνίας ώστε να είναι εφικτή η ανταλλαγή SOAP μηνυμάτων πάνω σε μέσα μεταφοράς και επικοινωνιακά κανάλια. Χρησιμοποιείται το HTTP, καθώς είναι το πιο διαδεδομένα χρησιμοποιούμενο πρωτόκολλο στο διαδίκτυο.

Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα της δομής ενός μηνύματος SOAP:



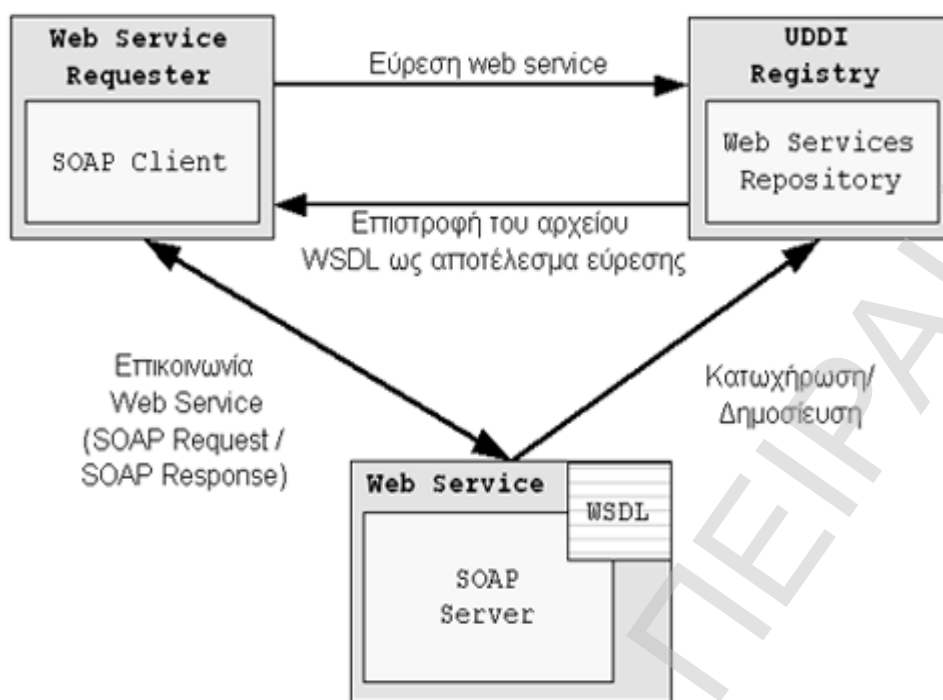
Σχήμα 1.1.4 – SOAP envelope

Πλέον, οι περισσότερες γλώσσες προγραμματισμού από την Delphi v7 μέχρι το Visual Studio .net, την PHP, την JSP και άλλες πολλές, υποστηρίζουν τη δημιουργία SOAP Servers με πάρα πολύ απλό τρόπο. Το μόνο που πρέπει να κάνει κανείς είναι να καθορίσει τις λειτουργίες που πρέπει να γίνουν όταν ο server δεχθεί κάποιο αίτημα προς εξυπηρέτηση.

Η WSDL (Web Service Description Language) [[W3C Web Services Description Language \(WSDL\)](http://www.w3.org/TR/wsdl/)] είναι μία γλώσσα σε XML μορφή η οποία περιγράφει απόλυτα ένα web service. Έτσι για κάθε ένα web service που δημιουργείται, αντίστοιχα πρέπει να δημιουργείται ένα αρχείο WSDL στο οποίο θα καταγράφονται όλες οι πληροφορίες για το ίδιο το service.

Πιο συγκεκριμένα εκεί καταγράφεται το πού βρίσκεται ο server (σε πια διεύθυνση), ποιες λειτουργίες υποστηρίζει καθώς και πως δέχεται και πως επιστρέφει τα δεδομένα για κάθε λειτουργία. Παραδείγματα αρχείων WSDL μπορούν να βρεθούν σε κάθε διεύθυνση που αναφέρεται σε web service. Π.χ. <http://www.capescience.com/webservices>

Οι προγραμματιστές και εδώ δεν θα πρέπει να ανησυχούν πολύ αφού υπάρχουν πάρα πολλά διαθέσιμα εργαλεία που δημιουργούν αυτόματα ένα wsdl αρχείο παράλληλα με τη δημιουργία του SOAP server.



Σχήμα 1.1.5 – Σχεδιάγραμμα λειτουργίας Web Service

Τέλος, το **UDDI (Universal Description, Discovery and Integration)** αποτελεί ένα πρωτόκολλο καταχώρησης για **web services**. Χρησιμοποιείται για να μπορούμε να παρέχουμε πληροφορίες για τα **web services**.

Κάθε καταχώρηση περιέχει το **wsdl** αρχείο και τη διεύθυνση που λειτουργεί η υπηρεσία στο **Internet**. Επιπρόσθετα σε κάθε καταχώρηση υπάρχουν και διάφορες άλλες πληροφορίες για την υπηρεσία που σχετίζονται με τον ιδιοκτήτη της, την πολιτική του κ.α.

Υπάρχουν διαφορετικοί τύποι καταχωρήσεων μίας υπηρεσίας. Πιο συγκεκριμένα υπάρχουν καταχωρήσεις που μπορούν να γίνουν για υπηρεσίες από όλο τον κόσμο και που απευθύνονται σε όλο τον κόσμο, αλλά και καταχωρήσεις που απευθύνονται μόνο σε εξειδικευμένες επιχειρήσεις προωθώντας έτσι και το **B2B** μοντέλο συνεργασίας. Τέλος υπάρχουν και καταχωρήσεις υπηρεσιών για πιο εξειδικευμένες περιπτώσεις.

Γνωρίζοντας όμως μία επιχείρηση το **UDDI** μπορεί με πολύ απλό τρόπο να αναζητήσει και να βρει το αρχείο **wsdl** για κάποια συγκεκριμένη υπηρεσία και κατόπιν να την εκμεταλλευτεί εξίσου απλά και γρήγορα.

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ WEB SERVICES

Υπάρχουν αρκετές πλατφόρμες στις οποίες μπορεί να βασιστεί κανείς για τη δημιουργία μιας υπηρεσίας παγκόσμιου ιστού. Από τη μεριά της Microsoft, οι έτοιμες λύσεις που δίνει το περιβάλλον Visual Studio .Net έχουν προσελκύσει πολλούς προγραμματιστές για την δημιουργία τέτοιου είδους υπηρεσιών . Επίσης άλλες μεγάλες εταιρίες όπως η IBM και η ORACLE χρησιμοποιούν τα δικά τους προγραμματιστικά εργαλεία. Εκτός από τα εμπορικά πακέτα, οι περισσότερες γλώσσες προγραμματισμού, όπως για παράδειγμα η Java και η PHP, έχουν ενσωματώσει την δυνατότητα για αυτόματη δημιουργία δημοσίευση και ανάκληση υπηρεσιών παγκόσμιου ιστού.

Τέλος υπάρχει μια σειρά από εργαλεία που διατίθεται δωρεάν στο διαδίκτυο και απλουστεύουν σημαντικά τον προγραμματισμό Διαδικτυακών Υπηρεσιών. Ακολουθεί λίστα με τα σημαντικότερα από αυτά..

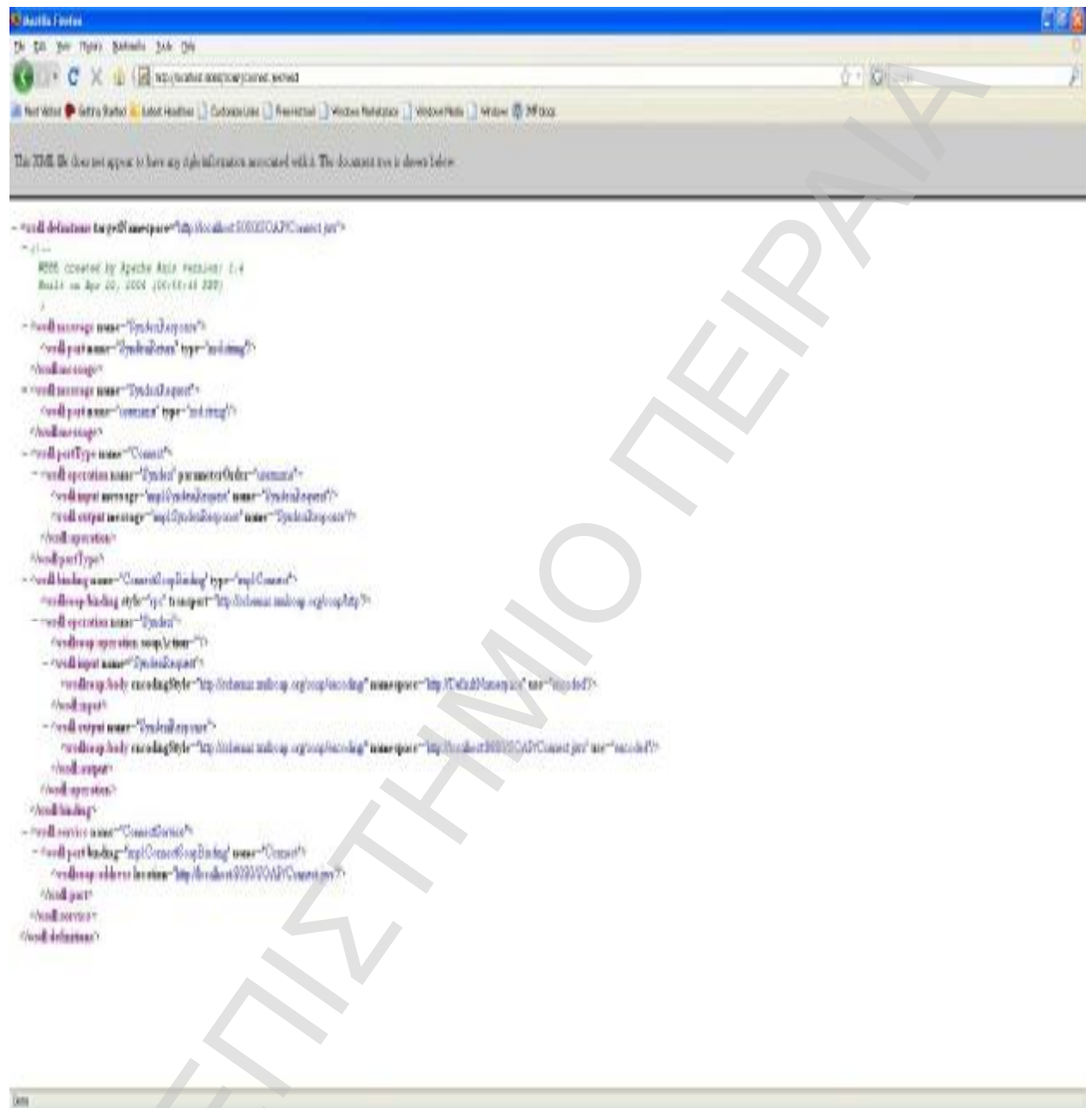
Ø Apache AXIS

Ο οργανισμός Apache κατασκεύασε ένα εργαλείο συμβατό με το SOAP 1.1, το οποίο ονομάστηκε Apache SOAP και το οποίο αργότερα μετεξελίχθηκε σε περιβάλλον ανάπτυξης Java, με το όνομα Apache Axis. Για να λειτουργήσει σε ένα σύστημα, το Apache Axis απαιτεί την εγκατάσταση ενός διακομιστή εφαρμογών, όπως είναι ο Apache Tomcat, και έναν σαρωτή XML (XML parser). Αρχικά δημιουργείται η εφαρμογή της Διαδικτυακής Υπηρεσίας σε γλώσσα Java και κατόπιν δίνεται η δυνατότητα στον διαχειριστή του διακομιστή να δημοσιεύσει επιλεκτικά κάποιες από τις μεθόδους της εφαρμογής ή και όλες καθώς και να ρυθμίσει διάφορες άλλες σχετικές παραμέτρους. Το αρχείο WSDL περιγραφής της υπηρεσίας μας παράγεται αυτόματα.. Περισσότερες πληροφορίες μπορούν να βρεθούν στο <http://ws.apache.org/axis/>

Η δημιουργία ενός java web service με την χρήση του AXIS είναι αρκετά εύκολη.

- Ø Δημιουργούμε ένα αρχείο .jws (java web service) στο οποίο ορίζουμε μια κλάση που υλοποιεί τις ζητούμενες υπηρεσίες στις μεθόδους της. Στο αρχείο αυτό κάνουμε χρήση των βιβλιοθηκών του πακέτου org.apache.axis.
- Ø Δημιουργούμε ένα φάκελο για το project μας στο /webapps του καταλόγου εγκατάστασης του Apache Tomcat . Τοποθετούμε σε αυτόν το .jws αρχείο μας
- Ø Tomcat δημιουργεί αυτόματα ένα .wsdl αρχείο το οποίο περιγράφει το .jws αρχείο μας. Για παράδειγμα - θέλουμε να δημιουργήσουμε ένα web service που αναλαμβάνει την σύνδεσή μας σε μια απομακρυσμένη βάση δεδομένων . Ας ονομάσουμε την σχετική κλάση μας Connect. Οπότε έχουμε την δημιουργία του αρχείου Connect.jws το οποίο και τοποθετούμε στον φάκελο /webapps/SOAP.

Αν λοιπόν ανοίξουμε ένα web browser και καλέσουμε το <http://localhost:8080/SOAP/Connect.jws> θα δούμε την παρακάτω οθόνη:



Σχήμα 1.1.6 – Οθόνη WSDL

Η οθόνη αυτή στην ουσία προβάλλει το WSDL αρχείο περιγραφής της διαδικτυακής υπηρεσίας Connect που μόλις δημιουργήσαμε. Το συγκεκριμένο αρχείο περιγράφει όλες τις μεθόδους που έχουμε ορίσει στην Connect. Συνεπώς γίνεται πολύ εύκολη η πρόσβαση από τον εκάστοτε προγραμματιστή στο αρχείο περιγραφής μιας υπηρεσίας για την δημιουργία κάποιας εφαρμογής πρόσβασης σε αυτή. Το αρχείο αυτό όπως προαναφέραμε δημιουργείται αυτόματα από τον εξυπηρετητή.

Ø IBM web service Toolkit

Η άλλη μεγάλη εταιρία που έχει συμβάλει τα μέγιστα στον σχεδιασμό, ανάπτυξη και καθιέρωση των τεχνολογιών των Διαδικτυακών Υπηρεσιών, διαθέτει και αυτή ένα δικό της σύνολο εργαλείων με όνομα Web Services Toolkit 3.0. Στο πακέτο υπάρχει ενσωματωμένη μια περιορισμένη έκδοση του διακομιστή εφαρμογών WebSphere της IBM. Το toolkit παρέχει την λειτουργικότητα που παρέχουν και τα προηγούμενα πακέτα, όπως για παράδειγμα αυτόματη ή χειρωνακτική δημιουργία του αρχείου περιγραφής των Διαδικτυακών Υπηρεσιών που δημιουργήσαμε(WSDL).

Αυτό που πρέπει να τονισθεί, διότι κάνει το πακέτο να ξεχωρίζει από τα προηγούμενα, είναι η πολύ καλή τεκμηρίωση και το πλούσιο συνοδευτικό υλικό που περιλαμβάνει.

Περιέχει έτοιμα παραδείγματα σύνθετων εφαρμογών που βασίζονται σε Διαδικτυακές Υπηρεσίες και μελέτες περιπτώσεων (case studies) για την επιτυχή αξιοποίηση και εισαγωγή των σχετικών τεχνολογιών στον κύκλο συναλλαγών μιας σύγχρονης επιχείρησης.

Ø Microsoft SOAP Toolkit 2

Με το MS Toolkit 2 είναι δυνατή στον προγραμματιστή η αξιοποίηση των Διαδικτυακών Υπηρεσιών σε πλατφόρμες Windows στις οποίες δεν έχει εγκατασταθεί το .NET Framework. Μπορεί να ενσωματωθεί στο περιβάλλον του Visual Studio 6.

Υποστηρίζει το SOAP έκδοση 1,1, την WSDL έκδοση 1,1 και το UDDI. Προσφέρει συστατικά τόσο πελάτη όσο και διακομιστή. Υπάρχει η δυνατότητα για αλληλεπίδραση των Διαδικτυακών Υπηρεσιών με στοιχεία COM και για το λόγο αυτό παράγονται αυτόματα δύο αρχεία, ένα με επέκταση .wsdl και ένα με επέκταση .wsml (Web Services Meta Language).

Υπάρχει επίσης και το συστατικό Remote Object Proxy Engine (ROPE) για την αυτοματοποίηση της διαδικασίας παραγωγής μηνυμάτων για την επικοινωνία με απομακρυσμένα αντικείμενα των Διαδικτυακών Υπηρεσιών. Τέλος για την απλούστευση της διαδικασίας δημιουργίας και επεξεργασίας των XML εγγράφων που στέλνονται ως SOAP μηνύματα, χρησιμοποιείται το SOAP Messaging Object (SMO) framework

Ø PHP Web Service Tools

Μία δωρεάν λύση προσφέρεται και στους προγραμματιστές δυναμικών ιστοσελίδων που χρησιμοποιούν την γλώσσα PHP. Υπάρχουν έτοιμες βιβλιοθήκες (π.χ. nusoap) που μπορούν να χρησιμοποιηθούν από οποιοδήποτε για να δημιουργήσει απλά έναν SOAP server και να τον καλέσει φτιάχνοντας ένα ολοκληρωμένο web service.

Για την υλοποίηση της παρούσας εργασίας χρησιμοποιήσαμε την πλατφόρμα την κοινότητας ανοιχτού λογισμικού (**open source**) του **Apache AXIS** σε συνδυασμό με την γλώσσα προγραμματισμού **Java**.

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΧΡΗΣΗ ΤΩΝ WEB SERVICES

Μέχρι στιγμής είδαμε τον τρόπο λειτουργίας και κατασκευής μιας υπηρεσίας παγκόσμιου ιστού. Κλείνοντας λοιπόν θα δούμε πώς κάνουμε χρήση ενός web service σαν το παραπάνω – υλοποιημένο σε εξυπηρετητή Apache Tomcat - δημιουργώντας έναν java client.

Η όλη διαδικασία περιλαμβάνει 3 απλά βήματα:

- Ø κάνουμε χρήση των βιβλιοθηκών του πακέτου org.apache.axis.
- Ø κάνουμε απομακρυσμένη κλήση (remote procedure call) μιας μεθόδου του αρχείου Connect.jws που δημιουργήσαμε πιο πάνω περνώντας τα απαραίτητα ορίσματα.
- Ø η μέθοδος που υλοποιεί το web service μας επιστρέφει πίσω το ανάλογο αντικείμενο το οποίο και μπορούμε να χρησιμοποιήσουμε κατά το δοκούν.

Όπως γίνεται αντιληπτό η χρήση ενός web service υλοποιημένου σε AXIS αποτελεί μια ειδική περίπτωση απομακρυσμένης κλήσης συνάρτησης και μοιάζει αρκετά με το γνωστό μας Java RMI.

Το AXIS επίσης κρύβει την πολυπλοκότητα του SOAP από τον προγραμματιστή , αφού αναλαμβάνει την υλοποίηση του και το parsing των αντικειμένων σε XML.Ο προγραμματιστής το μόνο που έχει να κάνει είναι μέσω του wsdl αρχείου να δει τι ορίσματα παίρνει σαν είσοδο η μέθοδος που υλοποιεί το web service τι επιστρέφει το web service που θέλει να καλέσει την ip διεύθυνση του server πάνω στον οποίο τρέχει.

Είναι πλέον έτοιμος να καλέσει το web service μέσω ενός αντικειμένου της κλάσης Call που παρέχεται από το AXIS και ορίζεται στο πακέτο org.apache.axis το οποίο και συμπεριλάβαμε στον client μας.

Η δημιουργία του αρχείου περιγραφής της υπηρεσίας μας γίνεται αυτόματα όπως προείπαμε από τον ίδιο τον εξυπηρετητή.

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ραγδαία ανάπτυξη του διαδικτύου οδήγησε στην παρουσία όλο και περισσότερων οργανισμών - επιχειρήσεων σε αυτό μέσω της τεχνολογίας του παγκόσμιου ιστού (WWW – World Wide Web) . Το γεγονός αυτό κατάστησε την ανάγκη δημιουργίας ενός μοντέλου παροχής υπηρεσιών προς τους καταναλωτές-πελάτες μέσω του διαδικτύου (διαδικτυακές υπηρεσίες – web services).

Κάτι τέτοιο έγινε δυνατόν με την ανάπτυξη της γλώσσας XML (Extensive Markup Language). Η XML αποτελεί ένα πρώτης τάξεως εργαλείο για την περιγραφή, μοντελοποίηση και κωδικοποίηση σύνθετων δομών δεδομένων σε ένα απλό αρχείο κειμένου.

Συνεπώς η ανταλλαγή XML μηνυμάτων πάνω από ένα ήδη υπάρχον πρωτόκολλο του διαδικτύου αποτελεί την ιδανική λύση μεταφοράς σύνθετου περιεχομένου από κάποιον πάροχο προς τον καταναλωτή μιας υπηρεσίας.

Ακριβώς αυτή την λειτουργία υλοποιεί το SOAP(Simple Object Access Protocol) το οποίο αποτελεί και την πιο διαδεδομένη λύση ανάπτυξης υπηρεσιών παγκόσμιου ιστού.

Πρόκειται για ένα πρωτόκολλο ανταλλαγής XML μηνυμάτων πάνω από κάποιο ήδη υπάρχον πρωτόκολλο του διαδικτύου όπως το HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol) ή το SMTP (Simple Mail Transfer Protocol)

Το συγκεκριμένο πρωτόκολλο έχει γίνει standard της W3C και έχει υιοθετηθεί από την πλειοψηφία των μεγάλων εταιρειών – οργανισμών στον χώρο του διαδικτύου και της πληροφορικής.

Κλείνοντας σε ότι αφορά την δημιουργία και χρήση web services υλοποιημένων σε SOAP, οι προγραμματιστές έχουν πλέον στην διάθεσή τους μια πληθώρα επιλογών από εργαλεία και ολοκληρωμένα περιβάλλοντα ανάπτυξης εφαρμογών –IDEs (Integrated Development Environments).

Για προγραμματιστές δυναμικών ιστοσελίδων που θέλουν αν εντάξουν κάποιο web service σε αυτές πολύ καλές λύσεις αποτελούν η πλατφόρμα της Microsoft (.net, ASP) καθώς και η PHP.

Για standalone εφαρμογές από την άλλη υπάρχει μια πληθώρα λύσεων σε περιβάλλον java με κυριότερες την πλατφόρμα του Eclipse και το AXIS plug in για τον Apache Tomcat.

ΚΕΦΑΛΑΙΟ 2^ο – JAVA MEDIA FRAMEWORK

JMF –JAVA MEDIA FRAMEWORK – ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια ο όρος «πολυμέσα» (**multimedia**) έχει μπει για τα καλά στην ζωή και την καθημερινότητά μας. Με τον όρο αυτό εννοούμε διάφορα μέσα μεταφοράς πληροφοριών συνδυασμένα μεταξύ τους. Ο πιο κοινός συνδυασμός φυσικά είναι αυτός της εικόνας με τον ήχο και το κείμενο – που άλλωστε αποτελούν τα βασικότερα μέσα μεταφοράς πληροφοριών.

Δύο κυρίως παράγοντες οδήγησαν στην αλματώδη ανάπτυξη της τεχνολογίας των πολυμέσων:

- ∅ Η ανάπτυξη της ψηφιακής τεχνολογίας
- ∅ Η ανάπτυξη των δικτύων υπολογιστών και του Διαδικτύου

Η μετάβαση από την αναλογική στην ψηφιακή εποχή επέτρεψε την αποθήκευση και επεξεργασία μεγάλου όγκου πληροφοριών, η οποία πριν ήταν πολύ δύσκολη και δαπανηρή.

Η ανάπτυξη του διαδικτύου έκανε εφικτή την ανταλλαγή μεγάλου όγκου πληροφοριών εύκολα και γρήγορα μεταξύ των χρηστών.

Έτσι πολύ απλά γεννήθηκε η ανάγκη δημιουργίας εφαρμογών που να υποστηρίζουν ροή διαφόρων τύπων δεδομένων (**multimedia streaming**) σε πραγματικό ή μη χρόνο (**real time**). Για μια γλώσσα προγραμματισμού που χρησιμοποιείται κατά κόρον για την ανάπτυξη εφαρμογών στο διαδίκτυο όπως – η **Java**- αποτέλεσε επιτακτική ανάγκη η υποστήριξη πολυμέσων.

Το **JMF -Java Media Framework API** –προεκτείνει λοιπόν την γλώσσα προγραμματισμού **Java** ώστε οι εφαρμογές της να υποστηρίζουν ήχο και βίντεο (πολυμέσα). Πιο συγκεκριμένα επιτρέπει εγγραφή (**capture**), αναπαραγωγή, μετάδοση, λήψη και αποθήκευση διαφόρων τύπων **media** μέσω της χρήσης της γλώσσας **java**.



Σχήμα 1.2.1 – Οθόνη JMF STUDIO

JMF –JAVA MEDIA FRAMEWORK – ΕΦΑΡΜΟΓΕΣ

Παρακάτω παρατίθενται μερικές από τις πιο κοινές εφαρμογές του **JMF**:

- Ø Αποστολή **media** (βίντεο ή/και ήχου) μέσω του **RTP**
- Ø Λήψη **media** μέσω **RTP**
- Ø Εξαγωγή και αποθήκευση **media** που παρελήφθησαν
- Ø Πρόσβαση σε ανεξάρτητα **video frames** ενός **Processor**
- Ø Δημιουργία ενός **Movie** αρχείου από μία λίστα εικόνων
- Ø Επεξεργασία ληφθέντων **media**
- Ø Αναπαραγωγή αρχείων βίντεο και ήχου
- Ø Λήψη βίντεο και/ ή ήχου μέσω κάμερας και μικροφώνου και αποστολή μέσω **Internet**
- Ø Ζωντανή εκπομπή ραδιοτηλεοπτικών προγραμμάτων

Όπως μπορούμε να δούμε το **JMF** υποστηρίζει την δικτυακή μετάδοση πολυμέσων μέσω ενός πρωτοκόλλου επιπέδου μεταφοράς, το **RTP –Real Transfer Protocol** – ανάλυση του οποίου γίνεται παρακάτω.

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΓΙΑ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JMF

Για την εγκατάσταση του **JMF** σε κάποιον υπολογιστή ισχύουν οι παρακάτω απαιτήσεις συστήματος:

- **166 MHz Pentium, 160 MHz PowerPC, ή 166 MHz UltraSparc 32 MB RAM** ή μεγαλύτερη Προαιρετικά, μια κατάλληλη κάρτα ήχου για αναπαραγωγή ήχου.
- **Windows 95/98/NT 4.0/2000, Solaris/SPARC** και κάθε συμβατό με **Java** λειτουργικό σύστημα.
- **JDK (Java Development Kit) 1.1** ή επόμενη έκδοση.

ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JMF

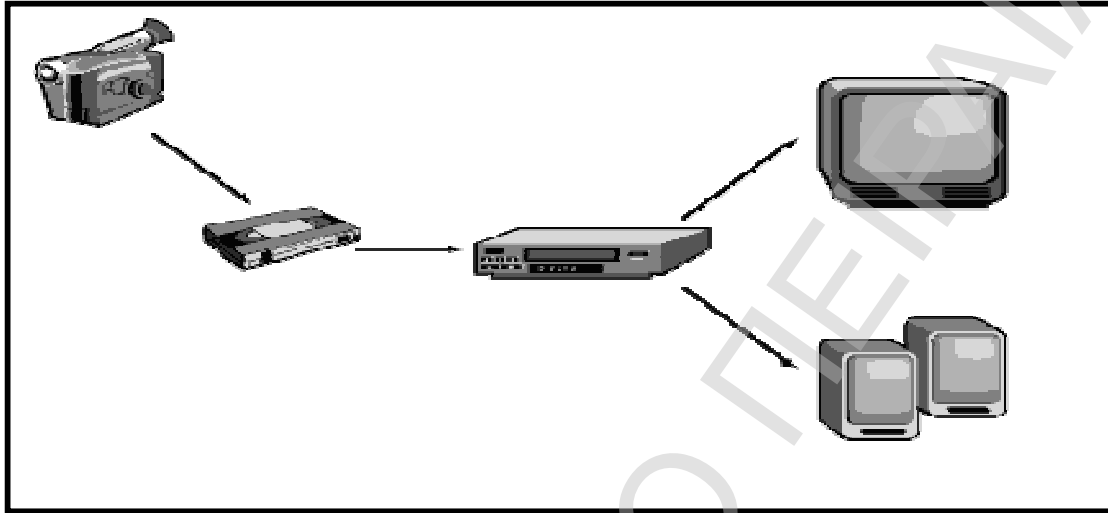
Εφόσον το μηχάνημα μας καλύπτει τις παραπάνω ελάχιστες απαιτήσεις συστήματος, η εγκατάσταση και λειτουργία του **JMF** αποτελεί μια σχετικά απλή διαδικασία:

- Ø Από την ιστοσελίδα της [Java](#) κατεβάζουμε το σχετικό εκτελέσιμο αρχείο. Για να κατεβάσετε το **JMF** πατήστε [εδώ](#).
- Ø Τρέχουμε το αρχείο.
- Ø Ελέγχουμε αν το **Path** και το **ClassPath** έχουν τροποποιηθεί ως εξής: **set PATH=%JMFDIR%\lib;%PATH%** και **set CLASSPATH=%JMFDIR%\lib\jmf.jar;%JMFDIR%\lib\sound.jar;%CLASSPATH%**.

Για να σιγουρευτούμε τέλος ότι η εγκατάσταση του **JMF** είναι επιτυχής, μπορούμε να «τρέξουμε» το **applet** που παρέχει η **Java** και ελέγχει εάν οι κλάσεις και οι βιβλιοθήκες του **JMF** έχουν εγκατασταθεί σωστά. Για την πραγματοποίηση του σχετικού ελέγχου πατήστε [εδώ](#).

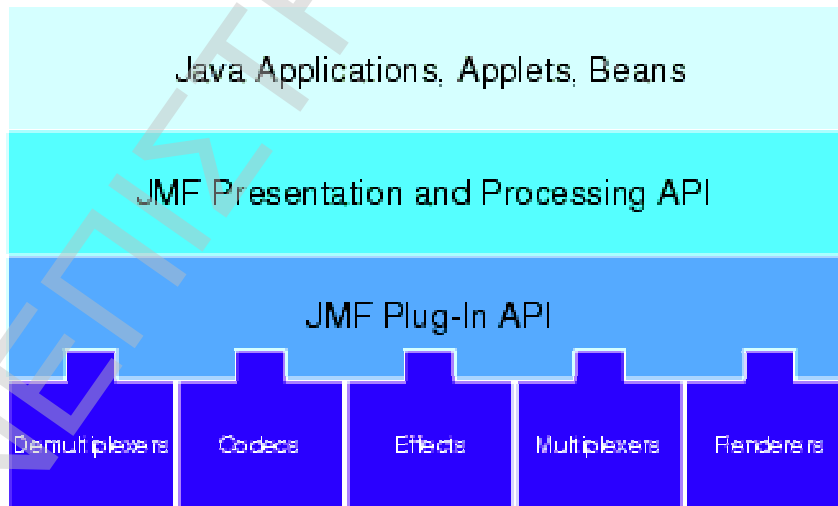
JMF –JAVA MEDIA FRAMEWORK – ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Το βασικό μοντέλο του **JMF** θυμίζει σε μεγάλο βαθμό το μοντέλο ενός κοινού **Video Player (VCR)**.



Σχήμα 1.2.2 – Αρχιτεκτονική JMF

Το **JMF** παρέχει μια διεπαφή για αναπαραγωγή πολυμεσικών αρχείων, όπως φαίνεται στο παρακάτω σχήμα.



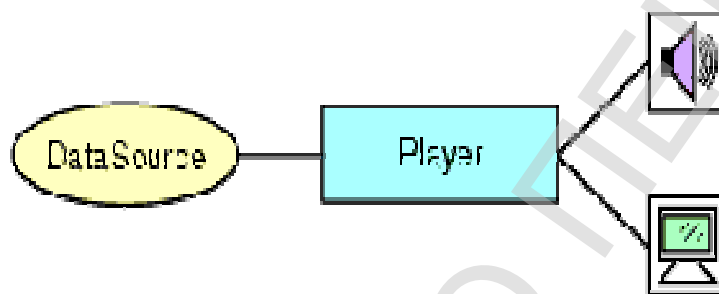
Σχήμα 1.2.3 JMF stack

Το **JMF** χρησιμοποιεί ένα βασικό πρότυπο. Ένα **datasource** περιβάλλει το ρεύμα (**stream**) μέσω των και ένας **player** παρέχει τους μηχανισμούς επεξεργασίας και ελέγχου του ρεύματος αυτού.

Τα **datasources** και οι **players** είναι σημαντικά μέρη του **JMF API** για τη διαχείριση της σύλληψης, παρουσίασης και επεξεργασίας των πολυμέσων. Εκτενέστερη ανάλυση των 2 αυτών εννοιών γίνεται παρακάτω.

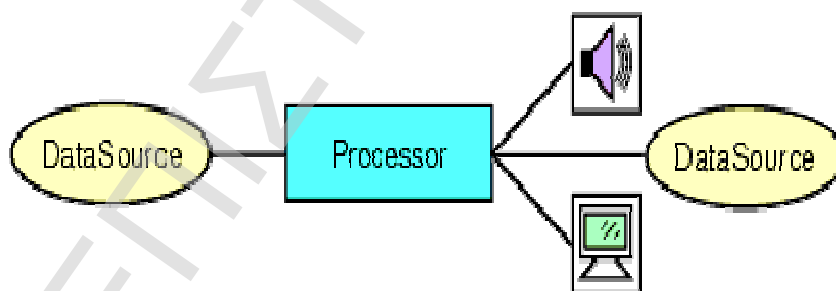
Το **JMF** παρέχει επίσης ένα χαμηλότερου επιπέδου **API** που υποστηρίζει την διαφανή ολοκλήρωση και την εύκολη εισαγωγή νέων επεκτάσεων.

Ένας **Player** επεξεργάζεται μια ροή πολυμεσικών δεδομένων και την παρουσιάζει σε ακριβή χρονικά διαστήματα. Τα πολυμεσικά δεδομένα παρέχονται από ένα **DataSource**. Η παρουσίαση γίνεται στην κατάλληλη συσκευή, ανάλογα με το είδος των πολυμεσικών δεδομένων όπως απεικονίζεται παρακάτω.



Σχήμα 1.2.4 - Player

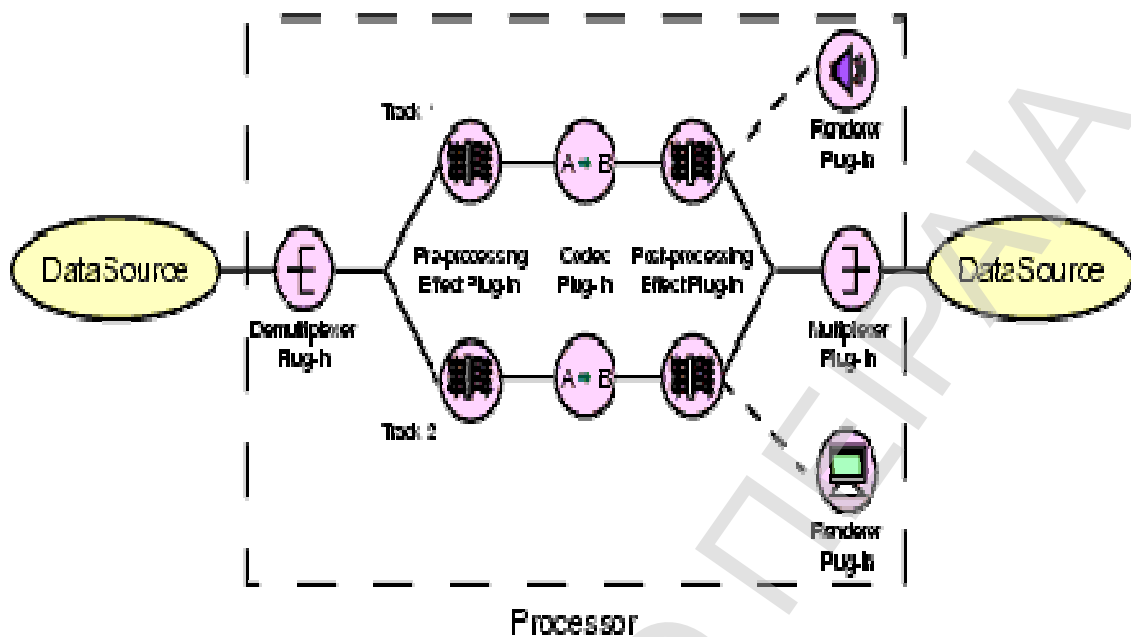
Για την παρουσίαση κάποιων πολυμεσικών δεδομένων μπορεί να χρησιμοποιηθεί και ένας **Processor**. Είναι ένας εξειδικευμένος τύπος **Player** που παρέχει δυνατότητα ελέγχου σχετικά με την επεξεργασία που συμβαίνει στην ροή των πολυμεσικών δεδομένων.



Σχήμα 1.2.5 -Processor

Η επεξεργασία που γίνεται από τους **Players** είναι προκαθορισμένη, αλλά οι **Processors** επιτρέπουν στον προγραμματιστή να καθορίσει το είδος της επεξεργασίας που εφαρμόζεται στα πολυμεσικά δεδομένα. Αυτό επιτρέπει την εφαρμογή την δημιουργία εφέ και την σύνθεση περισσότερων ροών σε πραγματικό χρόνο – παρέχοντας έτσι ευελξία στον εκάστοτε προγραμματιστή.

Η επεξεργασία των δεδομένων που λαμβάνει ο **Processor** από το **DataSource** γίνεται σε συγκεκριμένα στάδια.



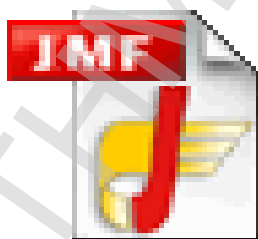
Σχήμα 1.2.6 – Επεξεργασία σήματος

- Ø **Demultiplexing (αποπολύπλεξη)** είναι η διαδικασία διαχωρισμού του εισερχόμενου stream. Εάν υπάρχουν πολλαπλά κανάλια (tracks) , εξάγονται ξεχωριστά. Για παράδειγμα, ένα αρχείο QuickTime μπορεί να αποπολυπλεκτεί σε ξεχωριστά κανάλια ήχου και εικόνας. Η αποπολύπλεξη γίνεται αυτόματα εφόσον το εισερχόμενο stream περιέχει πολυπλεγμένα δεδομένα.
- Ø **Pre - Processing (προ-επεξεργασία)** είναι η διαδικασία εφαρμογής αλγορίθμων εφέ στα κανάλια που εξάγονται από το εισερχόμενο stream.
- Ø **Transcoding (μετεγγραφή)** είναι η διαδικασία μετατροπής των πολυμεσικών δεδομένων κάθε καναλιού από μια μορφή σε μια άλλη. Εάν πρόκειται για μετατροπή ενός καναλιού από συμπιεσμένη μορφή ονομάζεται αποκωδικοποίηση. Η αντίστροφη διαδικασία λέγεται κωδικοποίηση.
- Ø **Post - Processing (μετά-επεξεργασία)** είναι η διαδικασία εφαρμογής εφέ στο αποκωδικοποιημένο κανάλι πολυμεσικών δεδομένων.
- Ø **Multiplexing (πολυπλεξία)** είναι η διαδικασία σύμπτυξης των μετακωδικοποιημένων πολυμεσικών δεδομένων σε ένα μόνο εξερχόμενο stream δεδομένων. Για παράδειγμα, ξεχωριστά κανάλια ήχου και εικόνας μπορούν να πολυπλεχτούν σε ένα MPEG -1 stream.
- Ø **Rendering (εμφάνιση)** είναι η διαδικασία παρουσίασης των πολυμεσικών δεδομένων στο χρήστη.

Η επεξεργασία σε κάθε φάση γίνεται από διαφορετικά επεξεργαστικά στοιχεία. Αυτά ονομάζονται plug-ins. Υπάρχουν πέντε τύποι plug-ins:

- Ø **Demultiplexer** – Επεξεργάζεται πολυμεσικά streams όπως WAV, MPEG ή QuickTime. Εάν το stream είναι πολυπλεγμένο, τότε εξάγει τα ξεχωριστά κανάλια.
- Ø **Effect** – Εφαρμόζει αλγορίθμους εφέ σε ένα κανάλι δεδομένων. Το αποτέλεσμα έχει την ίδια μορφή με την αρχική.
- Ø **Codec** – Κάνει κωδικοποίηση και αποκωδικοποίηση δεδομένων. Το αποτέλεσμα έχει διαφορετική μορφή από το αρχικό.
- Ø **Multiplexer** – Συνδυάζει πολλαπλά εισερχόμενα κανάλια σε ένα μοναδικό εξερχόμενο stream ως εξερχόμενο DataSource.
- Ø **Renderer** – Επεξεργάζεται τα πολυμεσικά δεδομένα σε ένα κανάλι, και τα παρουσιάζει σε μια συσκευή εξόδου.

Ακολουθεί μια εκτενέστερη ανάλυση των βασικότερων όρων- στοιχείων του JMF.

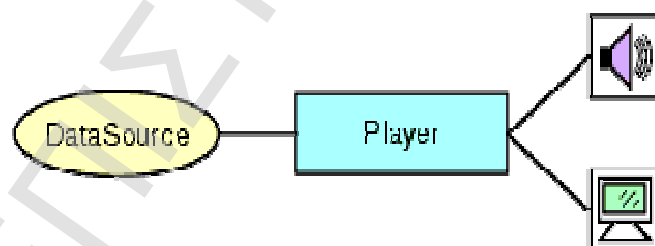


Σχήμα 1.2.7 JMF logo

ΑΝΑΛΥΣΗ ΣΗΜΑΝΤΙΚΟΤΕΡΩΝ ΟΡΩΝ ΤΟΥ JMF

Στην παράγραφο αυτή θα αναλύσουμε τους σημαντικότερους όρους της πλατφόρμας του JMF, όπως είναι οι Players, τα DataSource και τα Media Streams.

- Ø **DataSource:** περιέχει τα media streams (video, audio ή συνδυασμό των δύο), τη «διεύθυνση» των media και το σχετικό πρωτόκολλο για παράδοση. Το απαιτούμενο δεδομένο εισόδου για τη δημιουργία του είναι η «διεύθυνση» των media (JMF MediaLocator ή URL). Υπάρχουν δύο κατηγορίες: Pull Data Source (όταν ο client αρχίζει την μετάδοση) και Push Data Source (όταν ο server αρχίζει την μετάδοση).
- Ø **Media Streams:** ένα media stream είναι στην ουσία media data του προς μετάδοση αρχείου. Μπορεί να αποτελείται από πολλά κανάλια, τα οποία ονομάζονται Tracks. Streams με πολλά Tracks ονομάζονται multiplexed ή complex Media Streams και με την διαδικασία demultiplexing «αποπλέκουμε» τα διάφορα Tracks. Ο τύπος του Track (Track's type) καθορίζει τον τύπο των δεδομένων, αν είναι δηλαδή video ή audio. Αντιστοίχως με το DataSource, υπάρχουν δύο κατηγορίες Media Streams: Pull (όταν ο client αρχίζει την μετάδοση) και Push (όταν ο server αρχίζει την μετάδοση).
- Ø **Capture Devices:** συσκευές λήψης των media, π.χ. κάμερα, μικρόφωνο κτλ.
- Ø **DataSink:** διεπιφάνεια για αντικείμενα που δέχονται τα εισερχόμενα δεδομένα και τα αποθηκεύουν σε κάποιο προορισμό (συνήθως σε κάποιο φάκελο).
- Ø **Player:**



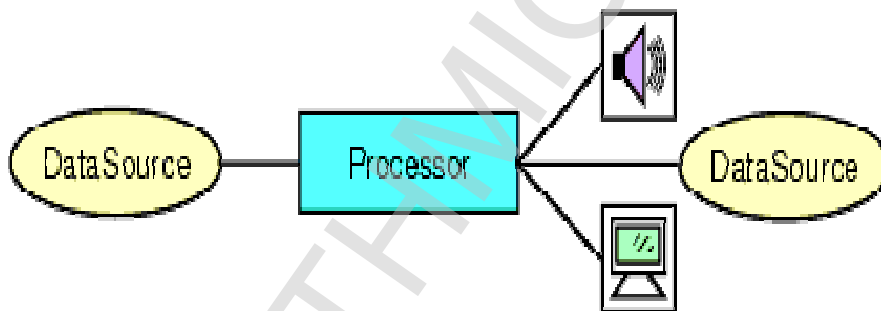
Σχήμα 1.2.8 - Player

έναν **Player** δέχεται σαν δεδομένα εισόδου ένα **stream** δεδομένων **video** ή **audio** και τα παρουσιάζει στην οθόνη ή στα ηχεία του H/Y. Ωστόσο, οι **Players** λειτουργούν σαν μαύρα κουτιά, με την έννοια ότι δεν μας επιτρέπουν επεξεργασία των **input streams** ή έλεγχο επί της προβολής. Ένας **Player** μπορεί να βρίσκεται σε μία εκ των παρακάτω έξι καταστάσεων:

- Ø **Unrealized:** η κατάσταση στην οποία εισέρχεται ο Player μόλις δημιουργηθεί. Δεν γνωρίζει τίποτα για τα media του.
- Ø **Realizing:** ο Player εισέρχεται σε αυτήν την κατάσταση όταν η μέθοδος Realize καλείται. Στην φάση αυτή ο Player καθορίζει τα

χαρακτηριστικά του, δηλαδή τις απαιτήσεις του και τα υποστηριζόμενα formats.

- Ø **Realized:** στην κατάσταση αυτή ο Player γνωρίζει τα χαρακτηριστικά του και μπορεί να παρέχει visual components και controls.
 - Ø **Prefetching:** ο Player ετοιμάζεται να παρουσιάσει τα media του. Στην φάση αυτή ο Player προ-φορτώνει τα προς παρουσίαση media data και δεσμεύει απαραίτητες πηγές.
 - Ø **Prefetched:** στην κατάσταση αυτή ο Player είναι έτοιμος να ξεκινήσει.
 - Ø **Started:** ο Player έχει ξεκινήσει και μπορεί να αρχίσει η παρουσίαση των media.
- Ø **Processor:** Πρόκειται για έναν Player ο οποίος έχει κατά βάση τα ίδια χαρακτηριστικά με τον Player. Ωστόσο, ο Processor δίνει την δυνατότητα στον χρήστη να επέμβει και να ελέγξει την επεξεργασία των media data.



Σχήμα 1.2.9 - Processor

Επίσης ο Processor μπορεί να στείλει τα εξαγόμενα δεδομένα (output data) είτε για παρουσίαση (όπως και ο Player), είτε σε ένα DataSource, το οποίο μπορεί να αποτελέσει είσοδο για ένα νέο Player ή Processor. Να επισημανθεί ότι ο Processor πέρα από τις έξι καταστάσεις του Player μπορεί να βρίσκεται σε ακόμη δύο καταστάσεις, οι οποίες τοποθετούνται μετά την Unrealized και πριν την Realizing κατάσταση. Οι δύο λοιπόν αυτές καταστάσεις είναι:

- Ø **Configuring:** ο Processor εισέρχεται σε αυτήν την κατάσταση όταν καλείται η μέθοδος configure (C). Στην φάση αυτή ο Processor συνδέεται (επικοινωνεί) με το DataSource, αποπλέκει το εισερχόμενο stream και συγκεντρώνει πληροφορίες για το format των εισερχόμενων δεδομένων.
- Ø **Configured:** ο Processor εισέρχεται στη φάση αυτή όταν έχει συνδεθεί με το DataSource και έχει καθοριστεί το format των δεδομένων. Ο Processor πλέον μπορεί, καλώντας την μέθοδο getTrackControls, να «πάρει» τα TrackControl αντικείμενα για το καθένα κανάλι (track) του media stream. Μέσω αυτών των αντικείμενων μπορούμε να

προγραμματίσουμε τον Processor ώστε να επεξεργαστεί τα δεδομένα μας όπως εμείς θέλουμε. Συγκεκριμένα μπορούμε να προσδιορίσουμε τα Effect, Codec και Renderer Plug-ins.

Ø **Managers:** διαμεσολαβητικά αντικείμενα τα οποία σε μεγάλο βαθμό καθορίζουν την συμπεριφορά και την αλληλεπίδραση μεταξύ των αντικειμένων, μέσω των οποίων πραγματοποιούνται οι διάφορες λειτουργίες και επεξεργασίες επί των δεδομένων. Είναι ο συνδετικός κρίκος μεταξύ Java και JMF, καθώς επιτρέπουν στις εφαρμογές των νέων διεπιφανειών και δυνατοτήτων (του **JMF**) να χρησιμοποιούνται, χωρίς κανένα πρόβλημα, με ήδη υπάρχουσες κλάσεις (της **Java**).

Το **JMF** χρησιμοποιεί τέσσερις **Managers**:

1. **Manager:** χειρίζεται την δημιουργία των Players, των Processors, των DataSources και των Datasinks.
2. **Package Manager:** διατηρεί λίστα πακέτων που περιέχουν κλάσεις JMF.
3. **Capture Device Manager:** διατηρεί λίστα διαθέσιμων capture συσκευών.
4. **PlugIn Manager:** διατηρεί λίστα διαθέσιμων plug-ins του JMF για επεξεργασία των media. π.χ. Multiplexers, Demultiplexers, Codecs, Effects και Renderers.

Ø **Event Model:** δομημένος μηχανισμός αναφοράς αλλαγών/ γεγονότων, ώστε τα προγράμματα του JMF να ενημερώνονται για την τρέχουσα κατάσταση των δεδομένων και να μπορούν να αντιμετωπίσουν τυχόν λάθη. Όταν ένα αντικείμενο JMF θέλει να αναφέρει την τρέχουσα κατάσταση αποστέλλει ένα MediaEvent. Υπάρχουν πολλοί τύποι MediaEvents και για κάθε ένα τέτοιο τύπο το JMF ορίζει μια διεπιφάνεια η οποία λαμβάνει αυτές τις ειδοποιήσεις, καλώντας την μέθοδο addListener.

Ø **Session:** είναι, θα λέγαμε, η ταυτότητα της επικοινωνίας μεταξύ δύο υπολογιστών. Στο αντικείμενο αυτό περιέχονται οι διευθύνσεις των επικοινωνούντων υπολογιστών, οι θύρες μέσω των οποίων θα πραγματοποιηθεί η επικοινωνία, το format των προς μετάδοση δεδομένων και κάποια λοιπά χαρακτηριστικά της επικοινωνίας (π.χ. FrameSize).

Κλείνοντας ακολουθεί μια εκτενή ανάλυση του RTP (Real Time Transfer Protocol) βασικότερου δικτυακού πρωτοκόλλου επιπέδου μεταφοράς (Transport Layer) που χρησιμοποιεί το JMF για την μετάδοση των πολυμεσικών δεδομένων μέσω διαδικτύου.

RTP – REAL TIME TRANSFER PROTOCOL

Το RTP – συντομογραφία για το Real Time Transport Protocol - (Πρωτόκολλο μεταφοράς σε πραγματικό χρόνο) αποτελεί ένα πρωτόκολλο επιπέδου μεταφοράς (Transport Layer) που ορίζει ένα τυπικό μορφότυπο πακέτου για την παράδοση ήχου και εικόνας μέσω του διαδικτύου (δικτυακή μετάδοση πολυμέσων).

Ορίζεται στο RFC 1889. Δημιουργήθηκε από τον όμιλο Audio Video Transport Working και δημοσιεύτηκε για πρώτη φορά το 1996.

Παρέχει end-to-end λειτουργίες δικτύου μεταφορών όντας κατάλληλο για εφαρμογές πραγματικού χρόνου μετάδοσης δεδομένων (real time applications), όπως αρχεία ήχου και βίντεο ή προσομοίωση των δεδομένων, πάνω από multicast ή μοναδικής διανομής υπηρεσιών δικτύου. Όπως σε κάθε πρωτόκολλο βασισμένο πάνω στο IP (Internet Protocol) έτσι και το RTP υπόκειται σε καθυστερήσεις μετάδοσης καθώς και απώλειες πακέτων.

Το **RTP** σε αντίθεση με το **TCP** δεν αντιμετωπίζει πόρο κράτησης και δεν εγγυάται την ποιότητα των παρεχόμενων υπηρεσιών για την παρακολούθηση σε πραγματικό χρόνο των υπηρεσιών. Για τον λόγο αυτό έχουμε την προσθήκη ενός νέου πρωτοκόλλου του **RTCP** που καθιστά δυνατή την παρακολούθηση της ροής και παράδοσης των δεδομένων. Έτσι οι εφαρμογές που μεταδίδουν δεδομένα μέσω **RTP** Μπορούν συλλέγοντας δεδομένα από το **RTCP** να βελτιώσουν την ποιότητα των παρεχόμενων υπηρεσιών τους (**QoS control**).

Τα δυο αυτά πρωτόκολλα λοιπόν -RTP και [RTCP](#) - είναι στενά συνδεδεμένα μεταξύ τους. Το RTP πραγματοποιεί την παράδοση των πραγματικών δεδομένων και το [RTCP](#) χρησιμοποιείται για την ανάδραση σχετικά με την ποιότητα της υπηρεσίας.

Με μια γρήγορη ματιά στα δημοφιλέστερα προγράμματα αναπαραγωγής πολυμέσων –όπως το QuickTime της Apple, το Windows Media Player της Microsoft, το Real Player και το Skype μπορεί κάποιος να διαπιστώσει το πόσο μεγάλη απήχηση έχει το RTP και πόσο ευρεία είναι η διάδοσή του. Όλες οι εφαρμογές (applications) που κάνουν μετάδοση πολυμέσων μέσω διαδικτύου – υποστηρίζουν την μετάδοση δεδομένων μέσω RTP. Στην ουσία έχουμε την δυνατότητα εγκαθίδρυσης ενός RTP session δηλώνοντας την ip διεύθυνση του άλλου άκρου επικοινωνίας.

Σημαντικό στοιχείο αποτελεί και το γεγονός πως η γλώσσα Java παρέχει μηχανισμούς υποστήριξης του RTP μέσω του JMF. Πιο συγκεκριμένα, υπάρχει η διεπιφάνεια RTPManager (πρώην SessionManager) η οποία είναι υπεύθυνη για την δημιουργία, την αρχικοποίηση, την συμμετοχή και το κλείσιμο ενός RTP Session.

ΜΕΡΟΣ ΔΕΥΤΕΡΟ

ΚΕΦΑΛΑΙΟ 1^ο- ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΕΙΣΑΓΩΓΗ

Στόχος μας είναι η ανάπτυξη μιας ολοκληρωμένης υπηρεσίας ανταλλαγής αρχείων πολυμέσων (**multimedia file exchange service**) με την χρήση υπηρεσιών παγκόσμιου ιστού (**web services**). Η υπηρεσία αυτή θα πρέπει να υποστηρίζει τόσο **ανταλλαγή αρχείων (file downloading)** όσο και **file streaming** μεταξύ των χρηστών.

Σε πρώτο στάδιο λοιπόν θα πρέπει να καθορίσουμε:

- Ø **Τον τρόπο πρόσβασης των χρηστών στην υπηρεσία** . Ο χρήστης θα αποκτά πρόσβαση στην υπηρεσία μέσω μιας αυτόνομης εφαρμογής που θα τρέχει τοπικά στο τερματικό του. Η εφαρμογή αυτή θα καλεί με την σειρά της κάποιες υπηρεσίες παγκόσμιου ιστού, παρέχοντας έτσι πρόσβαση στην υπηρεσία. Για την πρόσβαση στην υπηρεσία ο χρήστης θα χρησιμοποιεί κάποιο ψευδώνυμο το οποίο και θα ορίζει ο ίδιος μέσω της εφαρμογής.
- Ø **Την πολιτική πρόσβασης στην υπηρεσία μας**. Δεν θα απαιτείται κάποια εγγραφή στην υπηρεσία καθώς η πρόσβαση σε αυτή θα είναι απολύτως ελεύθερη. Αρκεί ο χρήστης να μπορεί να τρέξει την εφαρμογή μας σε έναν υπολογιστή συνδεδεμένο στο διαδίκτυο.
- Ø **Τον τρόπο ανταλλαγής αρχείων και πληροφοριών μεταξύ των χρηστών**. Οι χρήστες θα αποθηκεύουν την πληροφορία σχετικά με τα αρχεία που διαθέτουν προς ανταλλαγή κεντρικά σε μια βάση **δεδομένων**. Η ανταλλαγή όμως των αρχείων θα γίνεται σε επίπεδο P2P και δεν θα υλοποιείται κεντρικά από την βάση μας. Στην ουσία λοιπόν κάθε χρήστης θα αποστέλλει έναν κατάλογο - με τα αρχεία που διαθέτει στην υπηρεσία - κεντρικά σε μια βάση δεδομένων στην οποία θα έχουν πρόσβαση όλοι οι συνδεδεμένοι χρήστες.
- Ø **Τον αριθμό, τον ρόλο και τις λειτουργίες των υπηρεσιών παγκόσμιου ιστού που θα χρησιμοποιήσουμε**. Θα πρέπει να καθοριστεί ο αριθμός των υπηρεσιών παγκόσμιου ιστού που θα χρησιμοποιήσουμε. Επίσης, θα πρέπει να μοντελοποιηθούν οι λειτουργίες τους. Τέλος θα πρέπει να γίνει επιλογή της πλατφόρμας υλοποίησής τους.

Στην συνέχεια σε δεύτερο στάδιο θα πρέπει προσδιορίσουμε την αρχιτεκτονική υλοποίηση της υπηρεσίας μας. Επίσης θα πρέπει να καταγραφούν οι λειτουργίες που θα μπορεί να εκτελέσει κάθε χρήστης και στην συνέχεια να μοντελοποιηθούν. Ιδιαίτερη μνεία θα πρέπει να γίνει και σε κάποια ζητήματα ασφαλείας και τους τυχόν περιορισμούς που προκύπτουν από αυτά.

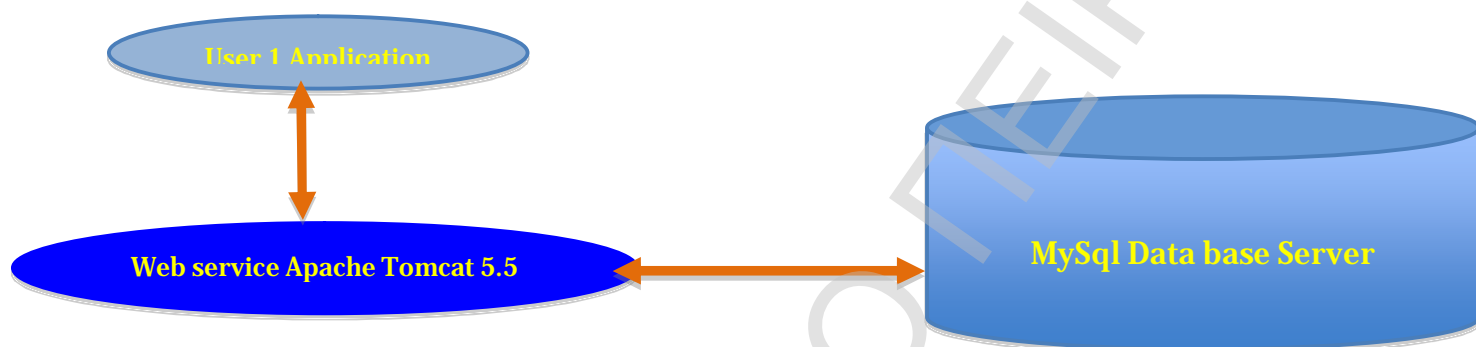
Κλείνοντας προχωράμε στο στάδιο υλοποίησης. Εδώ έχουμε υλοποίηση όσων καθορίσαμε στην πρώτη φάση. Θα πρέπει να επιλεγεί η γλώσσα υλοποίησης του συστήματος καθώς και το πρωτόκολλο υλοποίησης των υπηρεσιών παγκόσμιου ιστού.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Λαμβάνοντας υπόψη τα παραπάνω καταλήξαμε σε μια αρχιτεκτονική **3 επιπέδων** όπως φαίνεται στο παρακάτω σχήμα.

Οι πληροφορίες των ενεργών χρηστών θα αποθηκεύονται κεντρικά σε μια **βάση δεδομένων** όπως έχουμε ήδη αναφέρει. Οι χρήστες θα αποκτούν πρόσβαση στα δεδομένα της βάσης με την χρήση κάποιων υπηρεσιών παγκοσμίου ιστού (**web services**).



Σχήμα 2.1.1 – Αρχιτεκτονική υπηρεσίας

Δεν θα έχουμε φυσική αποθήκευση των αρχείων προς ανταλλαγή στην βάση μας . **Η ανταλλαγή των αρχείων γίνεται θα σε επίπεδο P2P**. Μόλις κάποιος χρήστης αποσυνδεθεί από την υπηρεσία αυτομάτως θα διαγράφεται και η καταχώρηση του από την βάση. Η ταυτοποίηση κάθε συνδεδεμένου χρήστη θα γίνεται με βάση την δικτυακή του διεύθυνση (**ip address**) και κάποιο ψευδώνυμο με το οποίο θα συνδέεται στην υπηρεσία.

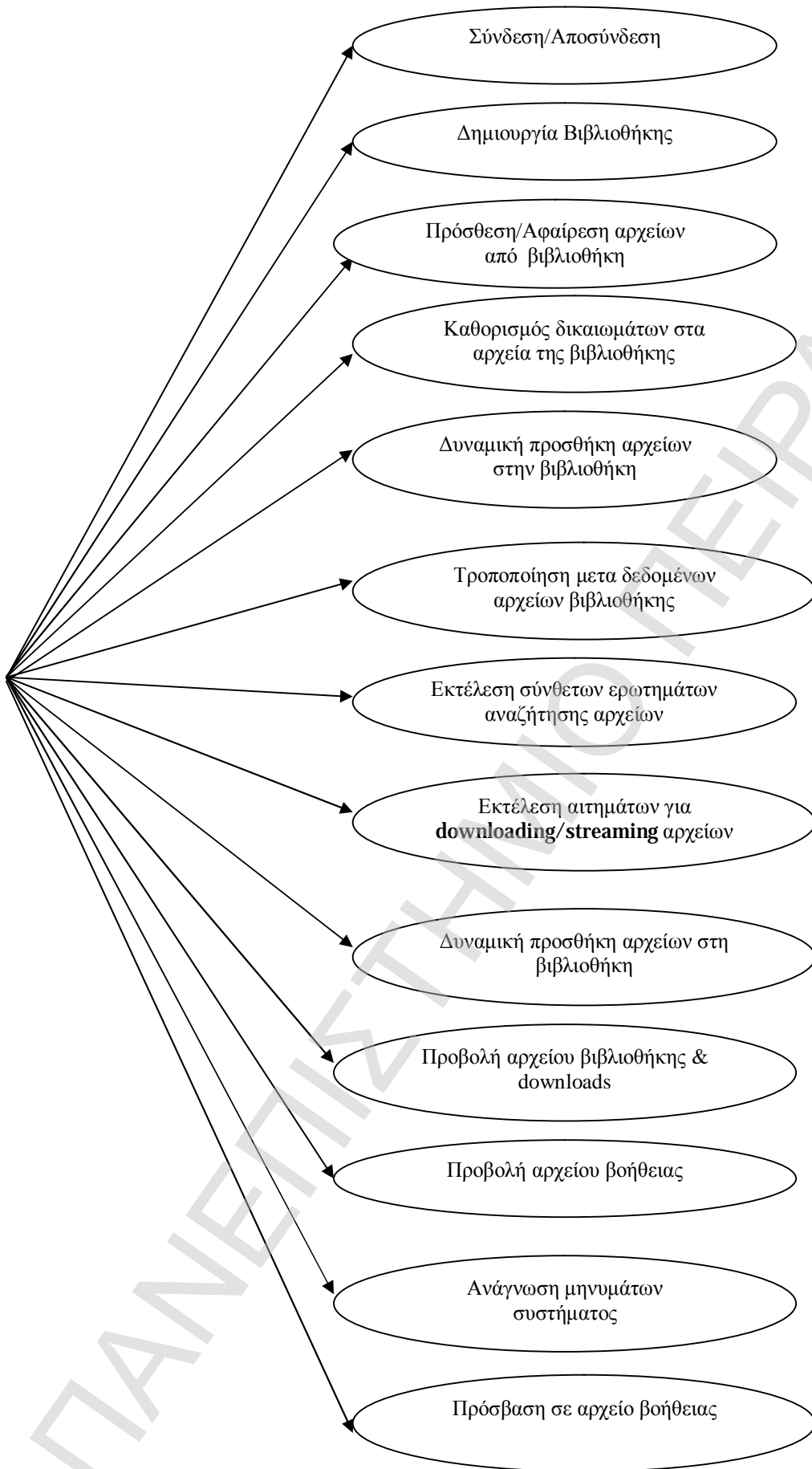


Σχήμα 2.1.2– Ανταλλαγή αρχείων σε P2P επίπεδο

Επιλέγοντας το μοντέλο αυτό η υπηρεσία μας καθίσταται αρκετά ασφαλής και αξιόπιστη καθώς οι χρήστες δεν έρχονται σε απευθείας επαφή με την κεντρική βάση δεδομένων μας. Επίσης, πετυχαίνουμε και μεγάλο βαθμό διαφάνειας καθώς οι χρήστες δεν έρχονται σε επαφή με την πολυπλοκότητα του συστήματος.

Σε ότι αφορά την εφαρμογή που θα τρέχει τοπικά ο χρήστης στον υπολογιστή του, θα περιλαμβάνει **κάποιες οθόνες με μενού και κουμπιά** υλοποιώντας όλες τις βασικές λειτουργίες που θα μπορεί να εκτελέσει . Παρακάτω ακολουθεί μια λίστα με την καταγραφή των βασικότερων λειτουργιών αυτών.

1. Για κάθε χρήστη ορίζουμε 2 καταστάσεις στις οποίες μπορεί να βρίσκεται - **συνδεδεμένος ή αποσυνδεδεμένος** από την υπηρεσία. Επομένως ο χρήστης θα μπορεί να εκτελεί ενέργειες **σύνδεσης/αποσύνδεσης**.
2. Ο κάθε **συνδεδεμένος** στην υπηρεσία χρήστης θα δημιουργεί και θα διαθέτει μια **βιβλιοθήκη (Library)** με τον κατάλογο των αρχείων που διαθέτει προς ανταλλαγή. Στη βιβλιοθήκη αυτή, για κάθε αρχείο του καταλόγου θα συμπεριλαμβάνονται τα στοιχεία που το χαρακτηρίζουν καθώς και τα δικαιώματα προσπέλασης που του έδωσε ο χρήστης.
3. Ο χρήστης θα πρέπει να μπορεί να **προσθέσει** και να **αφαιρέσει** αρχεία από την βιβλιοθήκη (Library).
4. Ο χρήστης θα πρέπει να μπορεί να **καθορίσει δικαιώματα** για κάθε αρχείο είτε για streaming είτε για downloading.
5. Ο χρήστης θα πρέπει να μπορεί να προσθέσει αρχεία στην βιβλιοθήκη ενώ είναι ήδη συνδεδεμένος στην υπηρεσία (δυναμική προσθήκη αρχείων).
6. Για κάθε αρχείο που προσθέτει ο χρήστης στην βιβλιοθήκη θα πρέπει να γίνεται ανάκτηση **των μετα-δεδομένων (meta data)** του αρχείου αυτού. Θα πρέπει να μπορεί να δει και να τροποποιήσει τα μετα-δεδομένα που μας ενδιαφέρουν.
7. Ο χρήστης θα πρέπει να μπορεί να εκτελεί **σύνθετες αναζητήσεις** με βάση κάποια συγκεκριμένα κριτήρια. Επειδή μιλάμε για αρχεία πολυμέσων τα κριτήρια θα αφορούν κατά βάση κάποια μετα δεδομένα – συγκεκριμένα για κάθε αρχείο.
8. Ένας χρήστης θα πρέπει να μπορεί να δει τις πληροφορίες της βιβλιοθήκης του.
9. Για κάθε χρήστη θα πρέπει να υπάρχει ένα ιστορικό με τα αρχεία που κατέβασε. Ένας λοιπόν θα πρέπει να μπορεί να έχει πρόσβαση το ιστορικό αυτό των **downloads** τα οποία έχει πραγματοποιήσει.
10. Ο χρήστης θα πρέπει να μπορεί να εκτελέσει ένα αίτημα για **streaming ή για download** ενός αρχείου από τα αποτελέσματα της αναζήτησης που εκτέλεσε.
11. Ο χρήστης θα πρέπει να μπορεί να εκτελεί **σύνθετες αναζητήσεις** με βάση κάποια συγκεκριμένα κριτήρια. Επειδή μιλάμε για αρχεία πολυμέσων τα κριτήρια θα αφορούν κατά βάση κάποια μετα δεδομένα – συγκεκριμένα για κάθε αρχείο.
12. Ο χρήστης θα πρέπει να μπορεί να λαμβάνει **ευδιάκριτα μηνύματα** από το σύστημα, όπως για την κατάσταση στην οποία βρίσκεται ή τα αποτελέσματα της αναζήτησης του.
13. Ο χρήστης θα πρέπει να μπορεί να έχει πρόσβαση σε κάποιο εγχειρίδιο χρήσης της εφαρμογής (**Help Menu**).



Σχήμα 2.1.3 – Διάγραμμα περιπτώσεων χρήσης

Όπως βλέπουμε λοιπόν κάθε χρήστης δημιουργεί ένα **Library** έγγραφο στο οποίο μπορεί να προσθέσει όσα αρχεία θέλει να διαμοιραστεί με τους υπόλοιπους χρήστες. Το έγγραφο αυτό θα βρίσκεται αποθηκευμένο σε έναν φάκελο με το όνομα Library. Σε κάθε αρχείο που προσθέτει ο χρήστης στο έγγραφο βιβλιοθήκης μπορεί να δώσει δικαιώματα είτε για **streaming** είτε για **downloading**.

Διακρίνουμε δυο μεγάλες κατηγορίες αρχείων που μπορεί να επιλέξει να προσθέσει ο χρήστης στο έγγραφο βιβλιοθήκης:

1. Αρχεία video-movies

2. Αρχεία ήχου-μουσικής

Για κάθε αρχείο που ο χρήστης προσθέτει στην βιβλιοθήκη (Library), αποθηκεύονται κάποια μετα -δεδομένα (meta data). Πιο συγκεκριμένα για κάθε αρχείο ήχου έχουμε:

- | | |
|--------------------------------------|---|
| <input type="checkbox"/> Τίτλος | <input type="checkbox"/> Είδος μουσικής |
| <input type="checkbox"/> Καλλιτέχνης | <input type="checkbox"/> Χρονολογία |
| <input type="checkbox"/> Άλμπουμ | |

Για κάθε αρχείο βίντεο/ταινία αντιστοίχως έχουμε:

- | | |
|-------------------------------------|--|
| <input type="checkbox"/> Τίτλος | <input type="checkbox"/> Είδος ταινίας |
| <input type="checkbox"/> Ηθοποιός | <input type="checkbox"/> Χρονολογία |
| <input type="checkbox"/> Σκηνοθέτης | |

Επιπλέον, για κάθε αρχείο έχουμε την συλλογή και κάποιων φυσικών χαρακτηριστικών του όπως:

- | | |
|---|---|
| <input type="checkbox"/> Μέγεθος αρχείου | <input type="checkbox"/> Τύπος αρχείου |
| <input type="checkbox"/> Διαδρομή (File Path) | <input type="checkbox"/> Δικαιώματα πρόσβασης |

Για κάθε μια κατηγορία αρχείων ο χρήστης μπορεί να εκτελέσει αναζήτηση με βάση κάποια κριτήρια. Πιο συγκεκριμένα για αρχεία μουσικής μπορεί εκτελεστεί αναζήτηση με κριτήριο:

- | | |
|--------------------------------------|--|
| <input type="checkbox"/> Καλλιτέχνης | <input type="checkbox"/> Είδος μουσικής |
| <input type="checkbox"/> Άλμπουμ | <input type="checkbox"/> Τίτλος τραγουδι |

Για τα αρχεία βίντεο τα αντίστοιχα κριτήρια αναζήτησης είναι:

- | | |
|---|--|
| <input type="checkbox"/> Ηθοποιός | <input type="checkbox"/> Σκηνοθέτης |
| <input type="checkbox"/> Τίτλος ταινίας | <input type="checkbox"/> Είδος ταινίας |

Τα αρχεία που κατεβάζει ο χρήστης μέσω της υπηρεσίας αποθηκεύονται σε έναν φάκελο με το όνομα «downloads». Στο φάκελο αυτό θα υπάρχει το αντίστοιχο **Downloads** στο οποίο καταγράφεται το ιστορικό των αρχείων που έχει κατεβάσει ο χρήστης.

Λαμβάνοντας υπόψη όλα τα παραπάνω, καταλήξαμε πως την **ιδανικότερη λύση** για την υλοποίηση των δυο αρχείων καταλόγων –**Library** και **Downloads** – μας την προσφέρει η **XML**.

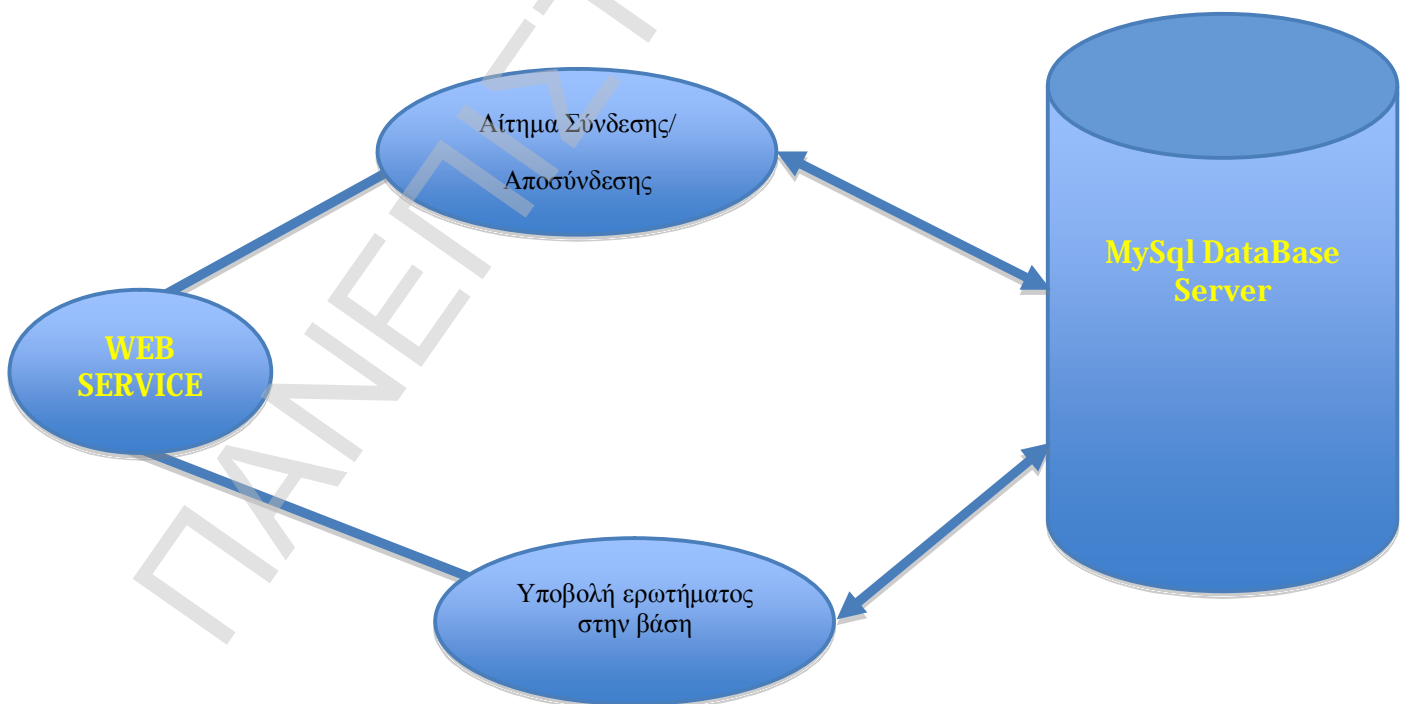
Η **XML** μας παρέχει την δυνατότητα ορισμού σύνθετων δομών δεδομένων όπως – **music file, video file** - των οποίων τα χαρακτηριστικά μπορούμε να προσδιορίσουμε με συγκεκριμένες τιμές σε ένα απλό αρχείο κειμένου. Περισσότερα για την λειτουργία της XML υπάρχει στο κεφάλαιο ανάλυσης των τεχνολογιών που χρησιμοποιήσαμε.

Δηλαδή, στην ουσία η απαιτούμενη πληροφορία για τα αρχεία κάθε χρήστη αποθηκεύεται σε ένα **απλό xml έγγραφο**, το οποίο και επεξεργάζεται κάποιες υπηρεσίες παγκόσμιου ιστού – εξάγοντας τα απαραίτητα δεδομένα για αποθήκευση στην κεντρική βάση μας.

Οι υπηρεσίες παγκόσμιου ιστού που χρησιμοποιούμε δρουν σαν ενδιάμεσο μεταξύ του χρήστη και της βάσης δεδομένων. Δεν χρειάζεται να βρίσκονται στον ίδιο υπολογιστή με την βάση μας – αφού μπορούν πολύ εύκολα να την καλέσουν απομακρυσμένα. Συνεπώς οποιοδήποτε αλλαγή αποφασιστεί να γίνει στην βάση δεδομένων μας, αυτή περνά απαρατήρητη από του χρήστες.

Επομένως, οι υπηρεσίες παγκόσμιου ιστού που χρησιμοποιούμε διεκπεραιώνουν τα ερωτήματα – αιτήματα των χρηστών προς την κεντρική βάση δεδομένων της υπηρεσίας μας. Μπορούμε να διακρίνουμε τις παρακάτω κατηγορίες ερωτημάτων – αιτημάτων προς την βάση δεδομένων:

- Ø Αίτηση σύνδεσης με την υπηρεσία μας (Connection Request)
- Ø Αίτηση αποσύνδεσης από την υπηρεσία μας (Disconnection Request)
- Ø Υποβολή ερώτησης για την εύρεση αρχείων με συγκεκριμένα χαρακτηριστικά



Σχήμα 2.1.4 – Επικοινωνία με βάση Δεδομένων

Οι υπηρεσίες παγκόσμιου ιστού είναι γραμμένες στην γλώσσα Java που και υλοποιούνται σε έναν εξυπηρετητή **Apache Tomcat έκδοση 5.5**.

Η κλήση των υπηρεσιών παγκόσμιου ιστού από την εφαρμογή του χρήστη γίνεται με την χρήση **SOAP (Simple Object Access Protocol)** που αποτελεί και την πιο διαδεδομένη λύση στον χώρο αυτό.

Για την χρήση μιας υπηρεσίας παγκόσμιου ιστού η εφαρμογή θα πρέπει φυσικά να γνωρίζει την δικτυακή διεύθυνση του εξυπηρετητή που είναι υλοποιημένη. Στην προκρινόμενη περίπτωση η εφαρμογή μας την διαβάζει από ένα εξωτερικό αρχείο κείμενου (PATH.txt). Συνεπώς, σε περίπτωση αλλαγής του εξυπηρετητή στον οποίο υλοποιούνται οι υπηρεσίες μας είναι πολύ εύκολη η αναβάθμιση της εφαρμογής μας.

Σε ότι αφορά τώρα το μέγεθος των δεδομένων που ανταλλάσει μια υπηρεσία παγκόσμιου ιστού με την εφαρμογή του χρήστη, έχει ελαχιστοποιηθεί με την κωδικοποίηση όλης της πληροφορίας σε ένα απλό **xml έγγραφο**.

Έτσι ελαχιστοποιούμε την καθυστέρηση στην ροή των δεδομένων από τον χρήστη προς την υπηρεσία μας. Επίσης ελαχιστοποιούμε τον φόρτο στον εξυπηρετητή μας καθιστώντας την όλη υπηρεσία πιο αποδοτική.

Μια από τις υπηρεσίες παγκόσμιου ιστού που υλοποιήσαμε αναλαμβάνει την υποβολή ερωτημάτων αναζήτησης στην βάση και την επιστροφή των αντίστοιχων αποτελεσμάτων. Τα αποτελέσματα αποθηκεύονται επίσης σε ένα xml έγγραφο με το όνομα **Results**. Η προβολή τους γίνεται με την χρήση **JTables**. Ανάμεσα στα άλλα στα αποτελέσματα περιλαμβάνονται τα στοιχεία του χρήστη στον οποίο ανήκει – ψευδώνυμο και ip address. Έτσι η μεταφορά του αρχείου μπορεί να γίνει σε επίπεδο **P2P** χωρίς κάποια κεντρική μεσολάβηση.

Στην συνέχεια ακολουθεί παρουσίαση και ανάλυση του κάθε επιπέδου της αρχιτεκτονικής μας.

Ξεκινάμε από την γραφική διεπαφή των χρήστη. Γίνεται παρουσίαση των βασικότερων οθονών που καλύπτουν τις περιπτώσεις χρήσης που ορίσαμε. Ακολουθεί μια ανάλυση των κλάσεων που υλοποιούν τις συγκεκριμένες λειτουργίες της εφαρμογής του χρήστη. Συνεχίζουμε με ανάλυση των υπηρεσιών παγκόσμιου ιστού που χρησιμοποιούμε.

Τέλος κλείνουμε με μια ανάλυση της κεντρικής βάσης δεδομένων που χρησιμοποιούμε για την ανταλλαγή των δεδομένων των ενεργών χρηστών.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΗ ΟΘΟΝΕΣ ΕΦΑΡΜΟΓΗΣ

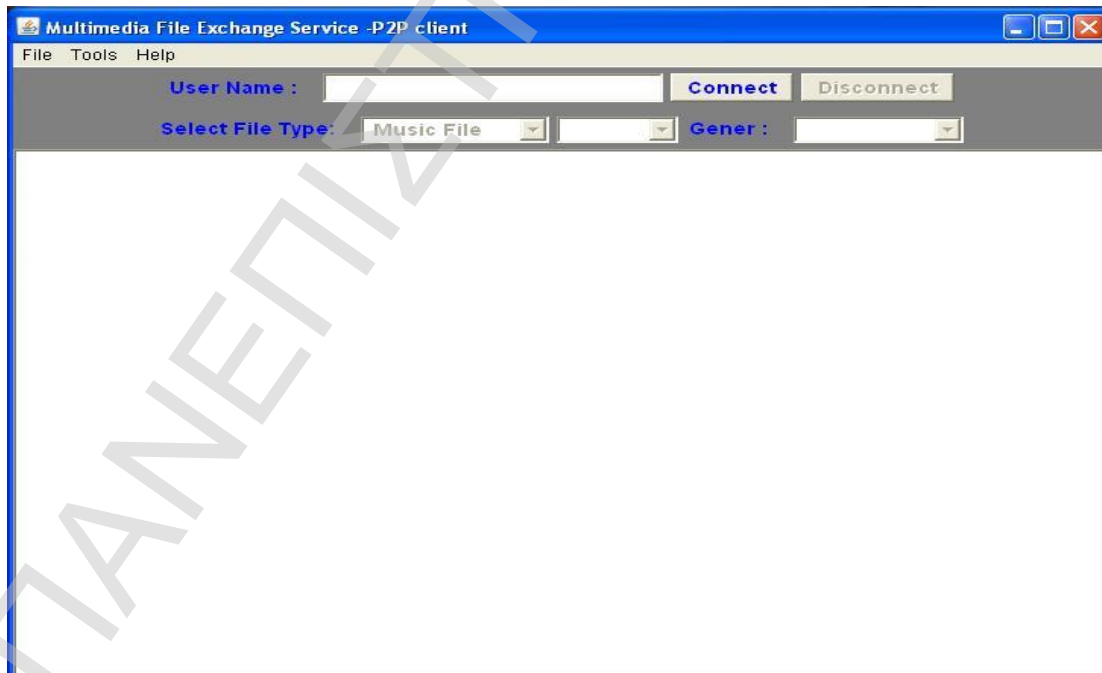
Για την υλοποίηση της εφαρμογής του χρήστη επιλέξαμε την γλώσσα προγραμματισμού Java για **τρεις κυρίως λόγους**:

- Ø Διότι συνεργάζεται **άψογα** με την **xml**, παρέχοντας ένα ισχυρό μηχανισμό προσπέλασης των εγγράφων και εξαγωγής δεδομένων από αυτά.
- Ø Διότι υπάρχουν **πληθώρα** επιλογών για την υλοποίηση κλήση και χρήση παγκόσμιου ιστού μέσω Java.
- Ø Διότι είναι **μεταφέρσιμη** οπότε η εφαρμογή μας θα είναι συμβατή με οποιαδήποτε πλατφόρμα και οποιοδήποτε λειτουργικό σύστημα.

Ο χρήστης αποκτά πρόσβαση στην υπηρεσία μέσω ενός Java GUI το οποίο περιλαμβάνει εννέα οθόνες και δομείται **από 20 κλάσεις** που αναλαμβάνουν:

- Ø την κλήση των Web services
- Ø την παρουσίαση μηνυμάτων λαθών
- Ø την παρουσίαση των αποτελεσμάτων αναζήτησης
- Ø την εγκαθίδρυση P2P συνδέσεων με άλλους χρήστες

Παρακάτω βλέπουμε την κεντρική οθόνη της γραφικής διεπιφάνειας του χρήστη, η οποία υλοποιείται από την **κλάση Client**.

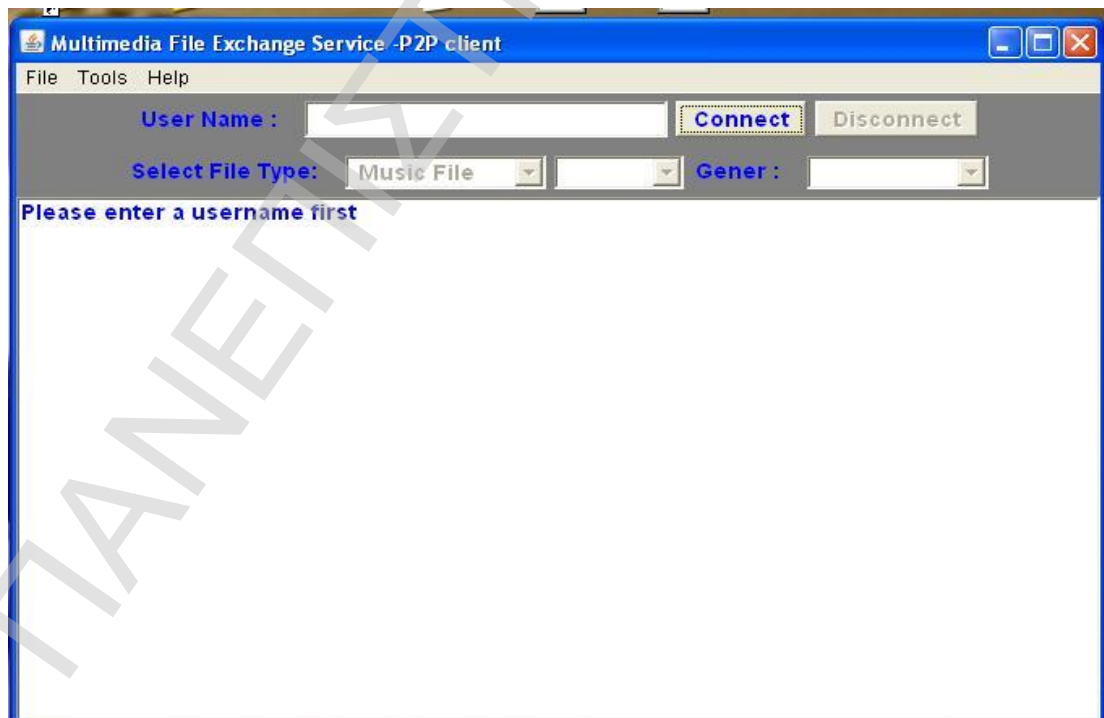


Σχήμα 2.1.5 – Κεντρική οθόνη

Όπως βλέπουμε η πρώτη οθόνη αποτελείται από ένα βασικό παράθυρο το οποίο περιλαμβάνει:

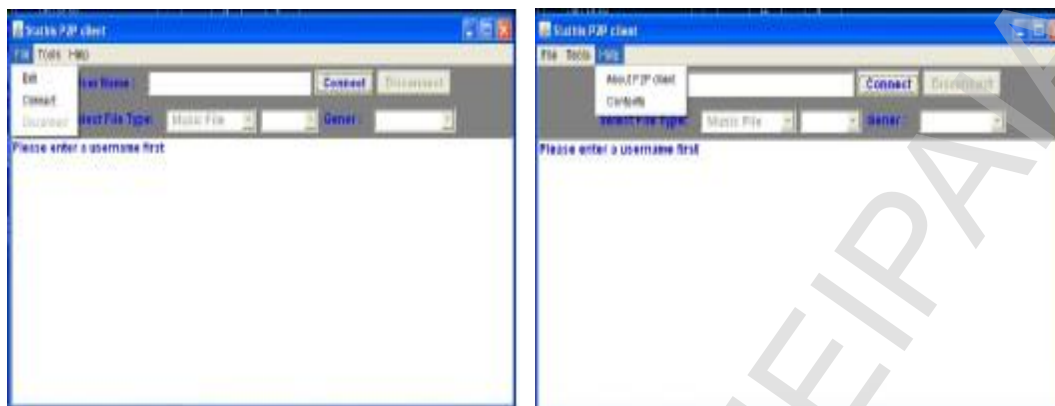
- Ø Μια γραμμή εργασίας (tool bar) με τρία βασικά μενού επιλογών. **File, Tools, Help.**
- Ø Ένα πεδίο κειμένου με την ετικέτα **User name** στο οποίο ο χρήστης πληκτρολογεί το ψευδώνυμο (user name) με το οποίο θα συνδεθεί στην υπηρεσία.
- Ø Δύο κουμπιά με τις ετικέτες **Connect και Disconnect** αντίστοιχα που αναλαμβάνουν την σύνδεση και την αποσύνδεση του χρήστη από την υπηρεσία. Μόνο το κουμπί σύνδεσης είναι ενεργό σε αυτή την πρώτη οθόνη. Το κουμπί αποσύνδεσης ενεργοποιείται με την επιτυχή σύνδεση του χρήστη στην υπηρεσία.
- Ø Τρεις λίστες επιλογής όπου ο χρήστης καθορίζει τα κριτήρια αναζήτησης. Σε αυτή την πρώτη οθόνη καμία από τις λίστες δεν είναι ενεργή. Ενεργοποιούνται με την επιτυχή σύνδεση του χρήστη στην υπηρεσία.
- Ø Μια οθόνη για την προβολή των μηνυμάτων της εφαρμογής καθώς και των αποτελεσμάτων της αναζήτησης.

Παρακάτω για παράδειγμα φαίνεται το αντίστοιχο μήνυμα σε περίπτωση που ο χρήστης προσπαθήσει να συνδεθεί χωρίς να προσδιορίσει κάποιο user name. Η λειτουργία της οθόνης προβολής των μηνυμάτων υλοποιείται από την **κλάση Display.**



Σχήμα 2.1.6 – Προβολή μηνυμάτων

Στις δυο παρακάτω εικόνες φαίνονται οι επιλογές των τριών βασικών μενού που περιλαμβάνονται στην μπάρα εργασίας (tool bar) – **File, Tools, Help**.



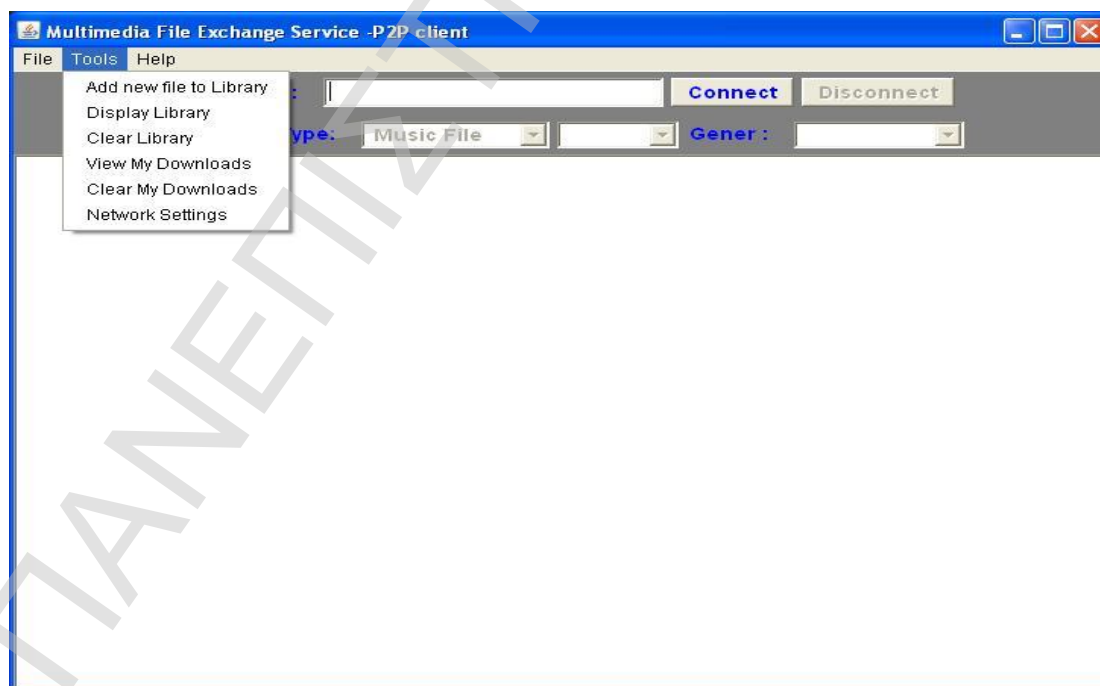
File menu screenshot

Help menu screenshot

Σχήμα 2.1.7 – Οθόνη προβολής μενού File & Help

Όπως φαίνεται παραπάνω τα μενού επιλογών File (Αρχείο) και Help (Βοήθεια) είναι σχετικά μικρά και περιλαμβάνουν από δυο επιλογές το κάθε ένα:

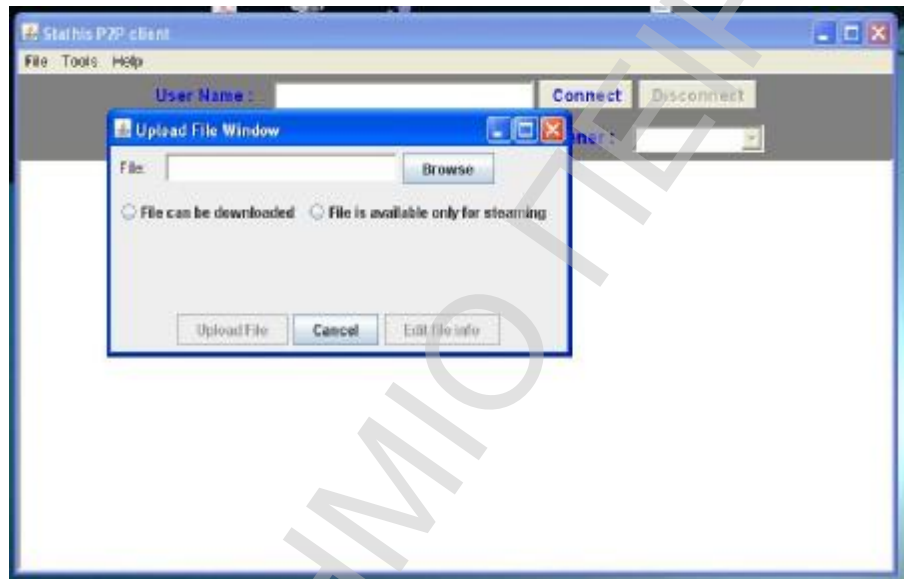
- | | |
|----------------------|--------------------|
| Ø Exit | Ø About P2P client |
| Ø Connect/Disconnect | Ø Contents |



Σχήμα 2.1.8 – Οθόνη προβολής μενού Tools

Αντιθέτως το μενού **επιλογών Tools (Εργαλεία)** αποτελεί και το **βασικό μενού** του παραθύρου μας περιλαμβάνοντας 5 επιλογές

- Ø **Add new file to Library** – Προσθήκη νέου αρχείου στο Library. Με την επιλογή αυτή ο χρήστης μεταβαίνει στην οθόνη προσθήκης νέου αρχείου στη Library – η οποία υλοποιείται από την **κλάση Add**. Μέσω της συγκεκριμένης οθόνης ο χρήστης περιηγείται (browses) στο τοπικό σύστημα αρχείων του υπολογιστή ώστε να επιλέξει το αρχείο που επιθυμεί να προσθέσει στον κατάλογο της βιβλιοθήκης.



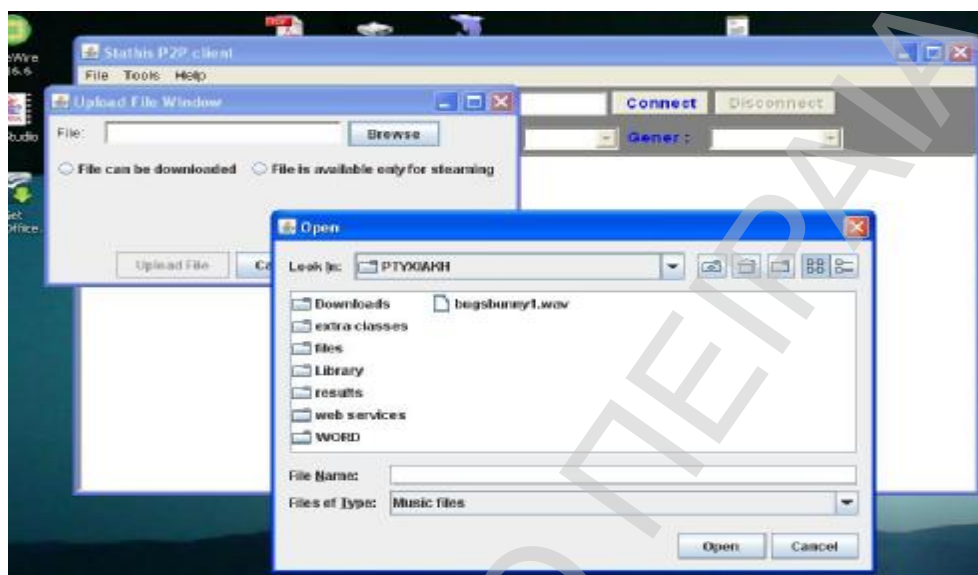
Σχήμα 2.1.9 – Οθόνη προσθήκης αρχείων στην βιβλιοθήκη

Όπως φαίνεται και από την παραπάνω εικόνα η συγκεκριμένη οθόνη περιλαμβάνει 4 απλά κουμπιά και ένα κουμπί επιλογής (radio button). Από τα 4 κουμπιά μόνο το κουμπί περιήγησης αρχείων (Browse button) είναι ενεργό. Ο χρήστης πατώντας το κουμπί αυτό μεταβαίνει στην οθόνη περιήγησης και επιλογής αρχείων από όπου μπορεί να επιλέξει το αρχείο που επιθυμεί να προσθέσει στην Library. Για το παράθυρο περιήγησης αρχείων έχουμε υλοποιήσει 3 φίλτρα αρχείων (file filters).

- Ø Αρχεία μουσικής
- Ø Αρχεία βίντεο
- Ø Όλα τα αρχεία

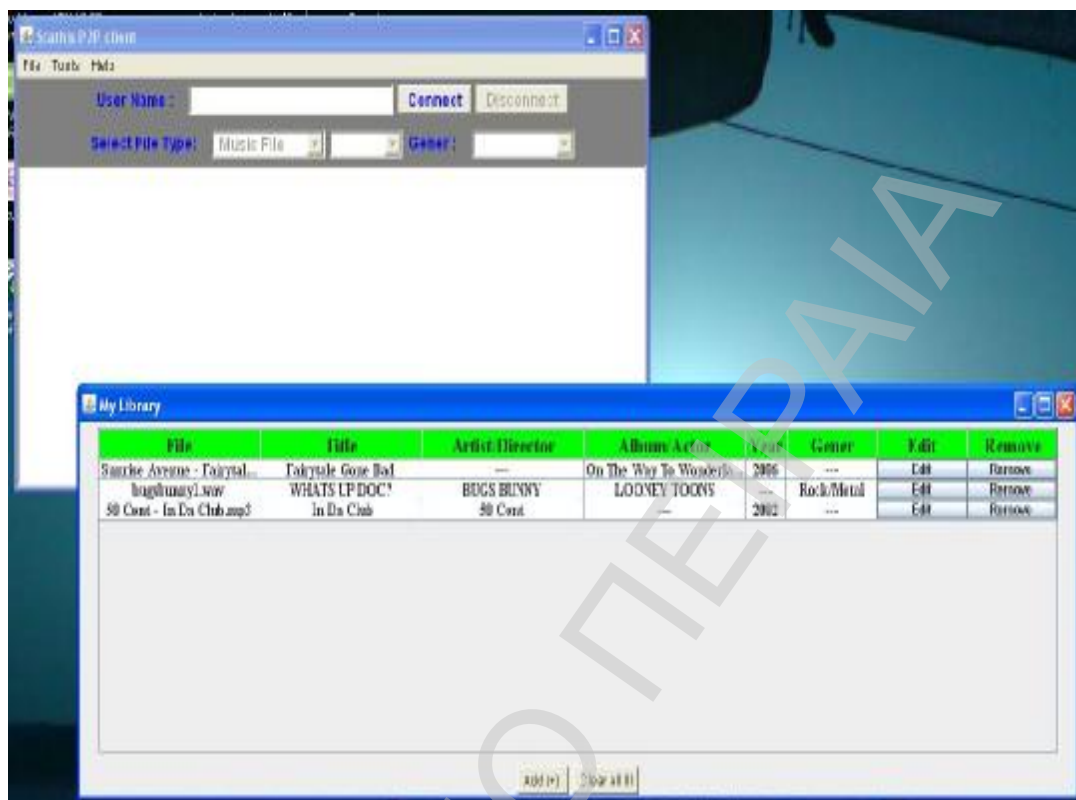
Αφού ο χρήστης επιλέξει το αρχείο που θέλει να προσθέσει μεταβαίνει στην αρχική οθόνη – με όλα τα κουμπιά ενεργοποιημένα αυτή την φορά. Μπορεί λοιπόν κάλλιστα να καθορίσει τα δικαιώματα προσπέλασης για αυτό το αρχείο (radio button) ή να αλλάξει τα μετα

δεδομένα του αρχείου. Η αλλαγή των μετα δεδομένων αυτών υλοποιείται από την κλάση Edit – η οποία αναλύεται παρακάτω.



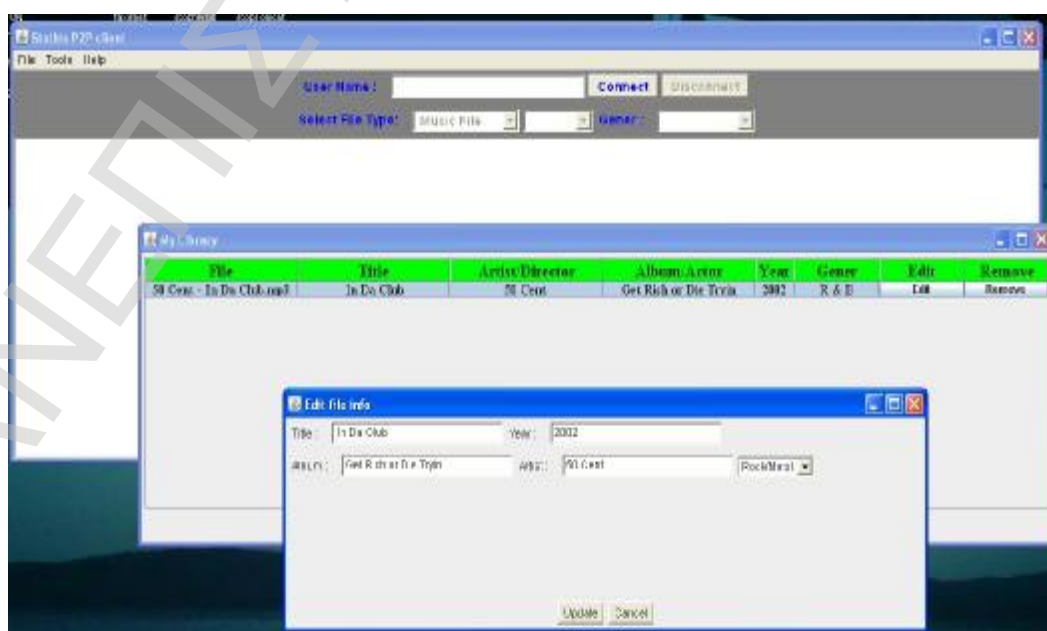
Σχήμα 2.1.10 – Οθόνη προσθήκης αρχείων στην βιβλιοθήκη

Ø **Display Library** – Προβολή των περιεχομένων του Library. Με την επιλογή αυτή ο χρήστης μεταβαίνει στην οθόνη προβολής των περιεχομένων της Library – η οποία υλοποιείται από την **κλάση Library**. Η παρουσίαση των δεδομένων γίνεται με την χρήση **JTable**. Στην οθόνη προβολής των δεδομένων μου για κάθε εγγραφή υπάρχουν 2 ενεργά κουμπιά – **Τροποποίησης (Edit)** και **Διαγραφής (Remove)** της κάθε εγγραφής. Επιλέγοντας το πρώτο ο χρήστης μεταβαίνει στην οθόνη τροποποίησης της συγκεκριμένης εγγραφής που υλοποιείται από την **κλάση Edit**. Επιλέγοντας αντιστοίχως το κουμπί διαγραφής της εκάστοτε εγγραφής αυτή αυτομάτως διαγράφεται από την βιβλιοθήκη.



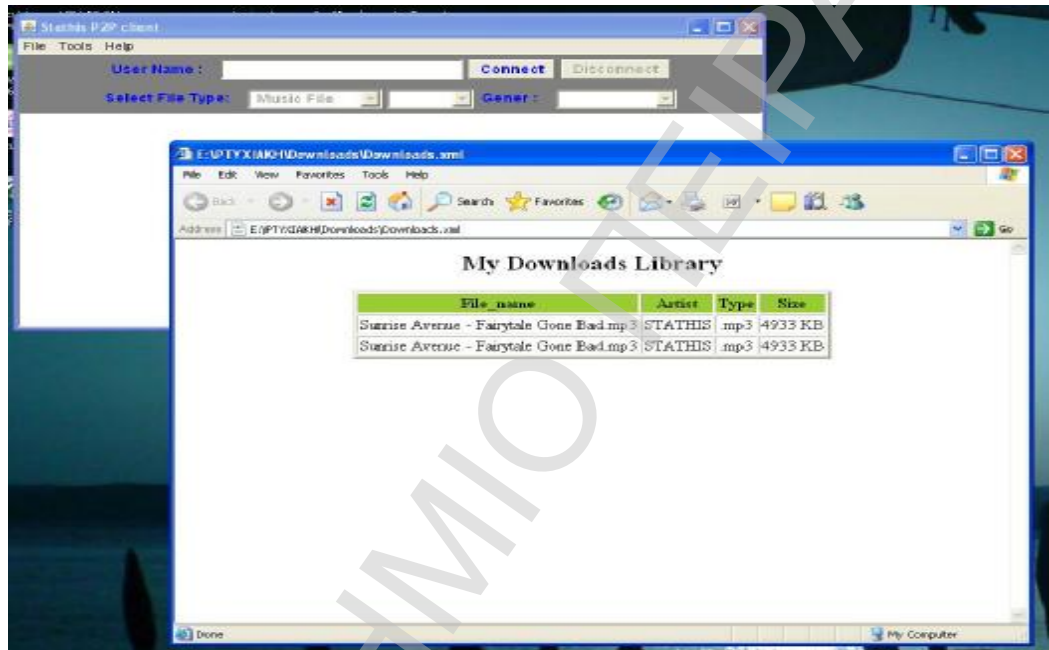
Σχήμα 2.1.11 – Οθόνη προβολής της βιβλιοθήκης

Ø **Clear Library** - Διαγραφή όλων των περιεχομένων του Library. Με την επιλογή αυτή ο χρήστης διαγράφει τα περιεχόμενα της Library. Η υλοποίηση της λειτουργίας αυτής γίνεται από την **κλάση Clear**.



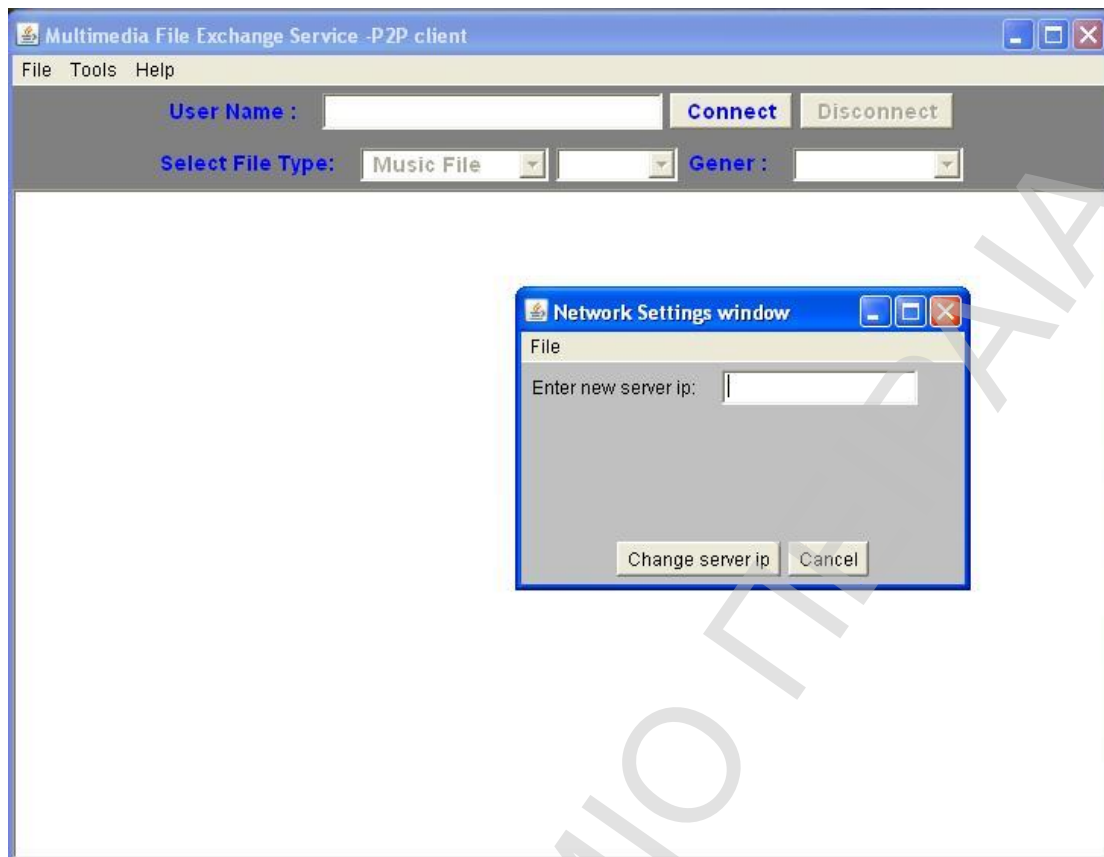
Σχήμα 2.1.12 – Οθόνη τροποποίησης εγγραφής

- Ø **View My Downloads** – Προβολή όλων των αρχείων που έχει κατεβάσει ο χρήστης. Η προβολή των συγκεκριμένων περιεχομένων δεν γίνεται από κάποια αυτόνομη κλάση. Τα δεδομένα προβάλλονται σε μια σελίδα html. Μέσω φυσικά του φυλλομετρητή (web browser) του συστήματος. Ο κατάλογος με τα αρχεία που έχει κατεβάσει ο χρήστης βρίσκεται στο αρχείο Downloads.xml. Για την προβολή λοιπόν στον web browser χρησιμοποιούμε έναν μετασχηματισμό xml.



Σχήμα 2.1.13 – Οθόνη προβολής του ιστορικού των αρχείων μου

- Ø **Clear My Downloads** - Διαγραφή των περιεχομένων του αρχείου που περιλαμβάνει τα αρχεία που έχει κατεβάσει ο χρήστης. Με την επιλογή αυτή ο χρήστης διαγράφει τα περιεχόμενα του αρχείου Downloads.xml. Η υλοποίηση της λειτουργίας αυτής γίνεται από επίσης από την κλάση **Clear**.
- Ø **Network Settings** – Ρυθμίσεις δικτύου από όπου ο χρήστης μεταβαίνει στην οθόνη αλλαγής δικτυακής διεύθυνσης του εξυπηρετητή όπως φαίνεται παρακάτω Στην οθόνη αυτή ο χρήστης μπορεί να πληκτρολογήσει την νέα διεύθυνση του εξυπηρετητή η οποία και θα αποθηκευτεί στο αρχείο PATH.txt.



Σχήμα 2.1.14 – Οθόνη προβολής network settings

Αυτές λοιπόν είναι οι βασικότερες οθόνες της εφαρμογής μας που υλοποιούν και τις βασικές λειτουργικές απαιτήσεις που ορίσαμε πιο πάνω.

Ακολουθεί ανάλυση των κλάσεων που απαρτίζουν την εφαρμογή μας καθώς και των υπηρεσιών παγκόσμιου ιστού που υλοποιήσαμε.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ - ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΗ ΑΝΑΛΥΣΗ ΚΛΑΣΕΩΝ

Όπως έχουμε προαναφέρει η εφαρμογή μας δομείται από **20 κλάσεις Java**. Αυτές με την σειρά τους μπορούμε να τις χωρίσουμε σε 4 μεγάλες κατηγορίες.

- | | |
|----------------------------|--------------------------------|
| ∅ Client side – κλάσεις | ∅ Result related κλάσεις |
| ∅ P2P- RTP related κλάσεις | ∅ Web service consumer κλάσεις |

Στην πρώτη κατηγορία έχουμε τις κλάσεις εκείνες που υλοποιούν τις οθόνες συστήματος που αναλύσαμε πιο πάνω και εκτελούν τις λειτουργίες των κουμπιών της κάθε οθόνης.

Στην δεύτερη κατηγορία έχουμε **τις 4 κλάσεις** που αναλαμβάνουν την δημιουργία των P2P συνδέσεων και την υλοποίηση του streaming.

Στις Result related κλάσεις έχουμε **τις 2 κλάσεις** που σχετίζονται με την υλοποίηση των κουμπιών σε έναν JTable (**ButtonRenderer** και **ButtonEditor**). Έχουμε επίσης και την κλάση προβολής των αποτελεσμάτων της εκάστοτε αναζήτησης.

Κλείνοντας στην τελευταία κατηγορία έχουμε **5 κλάσεις** – μια για κάθε υπηρεσία παγκόσμιου ιστού – που αναλαμβάνουν την κλήση των υπηρεσιών αυτών.

Ακολουθεί μια εκτενέστερη ανάλυση των κλάσεων ανά κατηγορία.

∅ Client side – κλάσεις

Client.java: Υλοποιεί το βασικό μου GUI που περιλαμβάνει το κεντρικό παράθυρο με όλες τις λειτουργίες που μπορεί να διαλέξει ο χρήστης. Περιλαμβάνει δύο βασικές μεθόδους την **Build_Window()** και την **actionPerformed()**. Η πρώτη υλοποιεί την κατασκευή του παραθύρου. Η δεύτερη χειρίζεται τις ενέργειες του χρήστη και ενεργοποιείται με το πάτημα ενός κουμπιού. Μέσω της μεθόδου αυτής αναλόγως με την ενέργεια του χρήστη καλείται και η αντίστοιχη κλάση. Αποτελεί την εκτελέσιμη κλάση της εφαρμογής μου και στην ουσία οποιαδήποτε αντικείμενα άλλης κλάσης καλούνται μέσα από αυτή.

Add.java – εκτελεί την προσθήκη νέων στοιχείων στην Library του χρήστη. Συμπεριλαμβάνει την δημιουργία της οθόνης προσθήκης νέου αρχείου στην βιβλιοθήκη καθώς και των 3 φίλτρων επιλογής αρχείων (File Choosers) της αντίστοιχης οθόνης. Αναλαμβάνει επίσης την προσθήκη της αντίστοιχης εγγραφής στο Library.κάνοντας **parsing** στο αντίστοιχο xl έγγραφο. Περιλαμβάνει 2 βασικές μεθόδους την **actionPerformed**– και την **Prosthiki()**. Η πρώτη ενεργοποιείται με το πάτημα ενός κουμπιού και χειρίζεται τις ενέργειες του χρήστη. Αναλόγως την ενέργεια εκτελείται στο σώμα της ο απαραίτητος κώδικας. Επίσης θα πρέπει να αναφερθεί πως καλεί την κλάση **ID3Read.java** για την ανάκτηση των απαραίτητων μετα δεδομένων του αρχείου που προσθέτουμε στην βιβλιοθήκη. Η δεύτερη μέθοδος

καλείται μέσω της πρώτης όταν χρήστης πατήσει το κουμπί Edit ή το Add. Εκτελεί την δημιουργία του νέου κόμβου για το αρχείο μας στον κατάλογο της βιβλιοθήκης.

Display.java – υλοποιεί την δημιουργία οθόνης προβολής μηνυμάτων στο κάτω μέρος του παραθύρου της εφαρμογής. Περιλαμβάνει κυρίως 2 μεθόδους την **showMessage()** και την **clearDisplay()**. Η πρώτη αναλαμβάνει την προβολή των μηνυμάτων ελέγχου ενώ η δεύτερη υλοποιεί έναν καθαρισμό της οθόνης. Επίσης συγκεκριμένη κλάση κάνει χρήση της διεπαφής **MouseListener** κάνοντας την ενεργή στην χρήση του ποντικού από τον χρήστη.

ID3Read.java – διαβάζει την ID3 ετικέτα από αρχεία μουσικής που προσθέτει ο χρήστης στο Library. Η κλάση αυτή δέχεται σαν εισροή (Input) το όνομα ενός αρχείου μουσικής και υλοποιεί ανάγνωση των μετα-δεδομένων του. Από αυτά επιλέγουμε όσα μας ενδιαφέρουν καθώς το πρότυπο ID3 περιλαμβάνει μια πληθώρα ετικετών. Καλείται από την κλάση Add κατά την διάρκεια της φάσης προσθήκης αρχείων στην βιβλιοθήκη.

Library.java - Η κλάση αυτή αναλαμβάνει την προβολή της Library του χρήστη μέσω ενός JTable. Περιλαμβάνει 2 μεθόδους την **Create_Table()** και την **actionPerformed ()**. Η πρώτη αναλαμβάνει την κατασκευή ενός JTable με δεδομένα από το αρχείο της βιβλιοθήκης μου κάνοντας parsing το αντίστοιχο έγγραφο Library.xml.. Η δεύτερη αναλαμβάνει την διαγραφή ή ενημέρωση μιας εγγραφής της βιβλιοθήκης με την κλήση των κλάσεων **Edit και Remove** αντίστοιχα.

Edit.java – Η κλάση αυτή επιτρέπει στον χρήστη να κάνει κάποιες αλλαγές σε κάποιες από τις εγγραφές που έχουν ήδη καταχωρηθεί στην βιβλιοθήκη. Αυτό γίνεται μέσα από μια ξεχωριστή οθόνη όπου φαίνονται τα στοιχεία της αντίστοιχης εγγραφής τα οποία και μπορεί να τροποποιήσει ο χρήστης. Περιλαμβάνει 3 βασικές μεθόδους την **Update()** την **actionPerformed ()** και την **boolean match()** . Η πρώτη αναλαμβάνει την ενημέρωση της αντίστοιχης εγγραφής στο αρχείο βιβλιοθήκης με τις αλλαγές που έκανε ο χρήστης. Η δεύτερη ενεργοποιείται με το πάτημα κάποιου κουμπιού και περιλαμβάνει τον κώδικα για την υλοποίηση των ενεργειών του χρήστη. **Η τελευταία συνάρτηση αναλαμβάνει την επικύρωση των δεδομένων** που εισάγει ο χρήστης σε κάποιο από τα πεδία . Ελέγχει για την παρουσία ειδικών χαρακτήρων οι οποίοι θα μπορούσαν να προκαλέσουν κάποιο πρόβλημα κατά την εισαγωγή μιας εγγραφής στην βάση δεδομένων μου. Δέχεται ως όρισμα 6 αλφαριθμητικά και επιστρέφει μια λογική μεταβλητή ελέγχου Αν λοιπόν ο χρήστης εισάγει τον ειδικό χαρακτήρα «'» τότε η μεταβλητή ελέγχου γίνεται αληθής και επιστρέφεται στην συνάρτηση που κάλεσε την match (). Στην περίπτωση αυτή έχω την εκτύπωση του ανάλογου μηνύματος στην οθόνη μου.

Clear.java – Η κλάση αυτή καθαρίζει τελείως το Library διαγράφοντας όλα τα αρχεία από αυτή. Περιλαμβάνει μόνο μια μέθοδο η οποία και κάνει parsing το αρχείο Library.xml διαγράφοντας όλες του τις εγγραφές.

Remove.java – Επιτρέπει την διαγραφή συγκεκριμένου στοιχείου από την Library με βάση το όνομα αρχείου. Αποτελεί μια υποπερίπτωση της Clear που αναλύσαμε πιο πάνω. Στην ουσία ελέγχει κάθε μια εγγραφή με μια συνθήκη και εάν αυτή επαληθευτεί προχωράει στην διαγραφή της συγκεκριμένης εγγραφής.

Settings.java – Η κλάση αυτή υλοποιεί μια οθόνη μέσω της οποίας δίνεται η δυνατότητα στον χρήστη να καθορίσει την δικτυακή διεύθυνση του εξυπηρετητή στον οποίο υλοποιούνται οι υπηρεσίες παγκόσμιου ιστού. Όπως έχουμε ήδη αναφέρει η εφαρμογή διαβάζει την συγκεκριμένη διεύθυνση από ένα απλό αρχείο κειμένου. Στην ουσία λοιπόν ενημερώνεται το συγκεκριμένο αρχείο από τον χρήστη. Η οθόνη που υλοποιεί απαρτίζεται από ένα πεδίο κειμένου και 2 κουμπιά. Ένα κουμπί ακύρωσης και ένα άλλο επιβεβαίωσης της αλλαγής. Περιλαμβάνει μια βασική μέθοδο την **actionPerformed ()**. Αυτή αναλαμβάνει την εκτέλεση των ενεργειών του χρήστη όταν πατήσει κάποιο από τα 2 κουμπιά της οθόνης μας.

Ø P2P- RTP related κλάσεις

AVreceive2.java – Η κλάση αυτή υλοποιεί τον απαραίτητο RTP Receiver για το streaming αρχείων πολυμέσων. Ακούει για κάποιο εισερχόμενο Stream στις ports 42060 (για ήχο- Audio Streaming) και 42062 (για εικόνα –Video Streaming).

Η κλάση αυτή βρίσκεται διαθέσιμη **στην ιστοσελίδα της Sun Micro Systems** (<http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/solutions/AVReceive2.java>) στο δικτυακό χώρο που είναι αφιερωμένος για το JMF.

AVTransmit2.java - Η κλάση αυτή υλοποιεί την μετάδοση RTP sessions στις θύρες 42060 και 42063 για ήχο και βίντεο αντίστοιχα ενώ δέχεται αίτηση για Streaming στις θύρες 42050 και 42052 (για λόγους συμβατότητας σε λειτουργία localhost- έγινε η αλλαγή των Ports κατά 10). Η κλάση αυτή βρίσκεται διαθέσιμη **στην ιστοσελίδα της Sun Micro Systems** (<http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/solutions/AVReceive2.java>) στο δικτυακό χώρο που είναι αφιερωμένος για το JMF. Απαρτίζεται από τις παρακάτω 7 βασικές μεθόδους. **Start()**, **stop()**, **createProcessor()**, **createTransmitter()**, **checkForVideoSizes()**, **setJPEGQuality()**, **waitForState()**.

FTP.java - Η κλάση αυτή υλοποιεί το βασικό Server socket που ακούει για εισερχόμενες συνδέσεις στην **TCP port 8000**. Περιλαμβάνει κυρίως 2 μεθόδους την **handleNewClient ()** και την **Termination()**. Η πρώτη χειρίζεται τις εισερχόμενες συνδέσεις ενώ η δεύτερη αναλαμβάνει τον τερματισμό μιας σύνδεσης. Πιο συγκεκριμένα η **handleNewClient ()** χειρίζεται μια σύνδεση αναλόγως εάν πρόκειται για streaming ή απλά ένα download request. Στην πρώτη περίπτωση καλείται η κλάση **AVTransmit2.java** για την μετάδοση της ζητούμενης ροής (stream). Στην δεύτερη περίπτωση πάλι κάνουμε αποστολή του αρχείου που ζητήθηκε μέσω ενός **Java ObjectInputStream**.

Connect.java – Η κλάση αυτή υλοποιεί το **client socket** για την ανταλλαγή των αρχείων ή το streaming αντίστοιχα. Δέχεται σαν εισροή την ip διεύθυνση του host στον οποίο βρίσκεται το αρχείο πολυμέσων το οποίο θέλουμε. Στην περίπτωση μεταφοράς του αρχείου αυτή γίνεται με την χρήση ενός **Java ObjectInputStream** και ενός **FileOutputStream** ενώ γίνεται και η αντίστοιχη προσθήκη στο αρχείο **Downloads.xml**. Στην περίπτωση που έχουμε file streaming αυτό υλοποιείται σε ένα ξεχωριστό νήμα εκτέλεσης στο οποίο και έχουμε κλήση της κλάσης **AVreceive2.java** για την λήψη της ζητούμενης ροής. Περιλαμβάνει στην ουσία μια

μέθοδο -πέρα από τον Constructor – Την Run() για την εκτέλεση του νήματος στην περίπτωση του αιτήματος για Streaming.

Ø Result related κλάσεις

Result.java – Η κλάση αυτή προβάλλει τα αποτελέσματα σε ένα JTable. Αναλαμβάνει την δημιουργία ενός πίνακα με δεδομένα από το αρχείο Results.xml στο οποίο και έχουν καταχωρηθεί τα αποτελέσματα της αναζήτησης.

ButtonEditor.java - Η κλάση αυτή σε συνδυασμό με την ButtonRenderer υλοποιεί την χρήση κουμπιών στα JTable μοντέλα μου. Έτσι οι πίνακες γίνονται πλέον διαδραστικοί. Περιλαμβάνει 2 βασικές μεθόδους την **getTableCellEditorComponent()** και την **getCellEditorValue()**. Αναλόγως το κουμπί, καλείται και η αντίστοιχη κλάση. Η κλάση αυτή χρησιμοποιείται από τις 2 κλάσεις που κάνουν χρήση JTable – **Result και Library**

ButtonRenderer.java - Η κλάση αυτή σε συνδυασμό με την ButtonEditor υλοποιεί την χρήση κουμπιών στα JTable μοντέλα μου.

Ø Web service consumer κλάσεις

Οι κλάσεις της κατηγορίας αυτής κάνουν χρήση των πακέτων **org.apache.axis.client.Call** και **org.apache.axis.client.Service** μέσω των οποίων και μπορούν να προσπελάσουν τις υπηρεσίες παγκόσμιου ιστού του συστήματος μας. Όπως έχουμε ήδη αναφέρει οι κλάσεις αυτές διαβάζουν την δικτυακή διεύθυνση του εξυπηρετητή -στον οποίο τρέχουν οι υπηρεσίες παγκόσμιου ιστού που κατασκευάσαμε- από ένα αρχείο κειμένου (PATH.txt). Ακολουθεί μια πιο εκτενής ανάλυση των κλάσεων αυτών.

Disconnect.java - Η κλάση αυτή κάνει χρήση του web service Disconnect που αναλαμβάνει την αποσύνδεση του χρήστη από την βάση δεδομένων καλώντας μεθόδους του java web service Disconnect.jws. Καλείται με το πάτημα του αντίστοιχου κουμπιού Disconnect από τον χρήστη. Στην περίπτωση επιτυχούς αποσύνδεσης του χρήστη προβάλλεται το ανάλογο μήνυμα στην κεντρική οθόνη της εφαρμογής.

DownloadClient.java - Η κλάση αυτή κάνει χρήση του web service που επιστρέφει τα αποτελέσματα της αναζήτησης του χρήστη - καλώντας μεθόδους του java web service Download.jws. Τα αποτελέσματα επιστρέφονται με την μορφή ενός δέντρου δεδομένων (**DOM tree**). Γίνεται επεξεργασία τους και αποθήκευση στο αρχείο **Results.xml**. Από εκεί αποκτά πρόσβαση στα δεδομένα η κλάση **Result** τα οποία και παρουσιάζει μέσω ενός JTable.

Register.java – Η κλάση αυτή αναλαμβάνει την σύνδεση του χρήστη με την υπηρεσία καλώντας μεθόδους του java web service Connect.jws. Η συγκεκριμένη υπηρεσία παγκόσμιου ιστού αναλαμβάνει την καταχώρηση του χρήστη στην κεντρική βάση δεδομένων ώστε αυτός να αποκτήσει πρόσβαση στην υπηρεσία. Στην περίπτωση που αυτή είναι επιτυχής επιστρέφεται και το ανάλογο μήνυμα το οποίο

προβάλλεται στην κεντρική οθόνη μέσω της κλάσης **Display**. Στην περίπτωση που αυτή είναι ανεπιτυχής προβάλλεται το αντίστοιχο μήνυμα `Problem connecting to service`.

UploadFile.java – Η κλάση αυτή αναλαμβάνει την ενημέρωση της βάσης με τα καινούρια αρχεία που παρέχει για upload ο χρήστης καλώντας την μέθοδο `uploadFile` του java web service `Uploadt.jws`. Καλείται στην περίπτωση που ο χρήστης έχει ήδη συνδεθεί στην υπηρεσία και αποφασίζει να ανανεώσει την βιβλιοθήκη του προσθέτοντας σε αυτή κάποιο αρχείο.

UploadLibrary.java – Η κλάση αυτή αναλαμβάνει την ενημέρωση της βάσης με τα αρχεία που παρέχει για upload ο χρήστης μόλις συνδεθεί καλώντας την μέθοδο `uploadLibrary` του java web service `Upload.jws`. Καλείται αυτόματα στο παρασκήνιο μετά την επιτυχή εκτέλεση της κλάσης **Register**.

Ακολουθεί εκτενής ανάλυση των 4 υπηρεσιών παγκόσμιου ιστού που κατασκευάσαμε.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΑΝΑΛΥΣΗ ΥΠΗΡΕΣΙΩΝ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

Ο χρήστης επικοινωνεί με την υπηρεσία μας καλώντας **4 υπηρεσίες παγκόσμιου ιστού (web services)**. Οι υπηρεσίες αυτές κατασκευάστηκαν με την χρήση της γλώσσας προγραμματισμού **Java**.

Εφόσον πρόκειται για υπηρεσίες παγκόσμιου ιστού θα πρέπει να υλοποιηθούν σε έναν ανάλογο εξυπηρετητή (web server). Ο συγκεκριμένος εξυπηρετητής θα πρέπει φυσικά να υποστηρίζει υλοποίηση εφαρμογών στην γλώσσα Java. Στην περίπτωση μας επιλέξαμε τον **Apache Tomcat web server έκδοση 5.5**. Πρόκειται για έναν ανοιχτού λογισμικού -εξυπηρετητή – ο οποίος φυσικά διατίθεται δωρεάν. Αναπτύχθηκε από το Apache Foundation και αποτελεί παραλλαγή του κλασικού Apache web server ώστε να υποστηρίζει εφαρμογές στην γλώσσα Java. Για περισσότερες πληροφορίες μπορεί κάποιος να ανατρέξει στο <http://tomcat.apache.org/>. Εκτενέστερη ανάλυση για τον **Apache Tomcat** γίνεται στο κεφάλαιο με τις τεχνολογίες που χρησιμοποιήσαμε.

Για την υλοποίηση όμως μιας υπηρεσίας παγκόσμιου ιστού από έναν εξυπηρετητή απαιτείται η υποστήριξη του αντίστοιχου πρωτοκόλλου υλοποίησης –στην δική μας περίπτωση του **SOAP (Simple Object Access Protocol)**. Επομένως θα πρέπει να επιλεγεί κάποια από τις διαθέσιμες πλατφόρμες υλοποίησης για τον συγκεκριμένο εξυπηρετητή.

Πιο συγκεκριμένα επιλέξαμε την λύση του **Apache AXIS** που αποτελεί ένα **plugin** για την υλοποίηση java web services στον **Apache Tomcat**. Αναπτύχθηκε από το **Apache foundation**, και αποτελεί μια καλή λύση για την ανάπτυξη υπηρεσιών σε open source περιβάλλον. Περισσότερες πληροφορίες μπορούν να βρεθούν στο <http://ws.apache.org/axis/>.

Οι 4 υπηρεσίες μας αναλαμβάνουν την επικοινωνία του χρήστη με την βάση δεδομένων και την καταχώρηση των πληροφοριών σχετικά με κάθε χρήστη σε αυτή. Στην ουσία δηλαδή αποτελούν έναν ενδιάμεσο μεταξύ των χρηστών και της βάσης δεδομένων στην αρχιτεκτονική των 3 επιπέδων που έχουμε υιοθετήσει.

Παρακάτω ακολουθεί ανάλυση των 4 αυτών υπηρεσιών. Πρόκειται για τις υπηρεσίες

- Ø Connect
- Ø Disconnect
- Ø Download
- Ø Upload

Ø Connect

Η συγκεκριμένη υπηρεσία παγκόσμιου ιστού αναλαμβάνει την σύνδεση του χρήστη με την υπηρεσία και την καταχώρηση του στην βάση δεδομένων – πιο συγκεκριμένα στον πίνακα των ενεργών χρηστών.

Περιλαμβάνει μια μέθοδο την **Syndesi()**. Η μέθοδος αυτή δέχεται σαν εισροή ένα αλφαριθμητικό που αναπαριστά το ψευδώνυμο (**user name**) του χρήστη. Στην

συνέχεια αναλαμβάνει την ταυτοποίηση της δικτυακής διεύθυνσης του υπολογιστή μέσω του οποίου καλεί ο χρήστης την υπηρεσία (**ip address**). Τέλος αναλαμβάνει την σύνδεση με την βάση δεδομένων και την εκτέλεση του ερωτήματος καταχώρησης της εγγραφής του χρήστη στην βάση. Στην ουσία δημιουργείται μια καταχώρηση στον πίνακα **User** για κάθε χρήστη με βάση τον συνδυασμό ψευδώνυμου και της δικτυακής διεύθυνσης.

Ø Upload

Η συγκεκριμένη υπηρεσία παγκόσμιου ιστού αναλαμβάνει την καταχώρηση στην βάση δεδομένων των στοιχείων της βιβλιοθήκης (**Library**) ενός χρήστη που μόλις συνδέθηκε επιτυχώς. Συνεπάγεται λοιπόν πώς καλείται αμέσως μετά την εκτέλεση της υπηρεσίας **Connect**.

Περιλαμβάνει 2 μεθόδους την **uploadLibrary()** και την **uploadFile()**. Η πρώτη δέχεται σαν εισροή ένα αλφαριθμητικό και έναν πίνακα με bytes. Στην συνέχεια αναλαμβάνει την σύνδεση με την βάση δεδομένων. Ακολουθεί μια ανάλυση των δεδομένων που έχει λάβει σαν εισροή. Πιο συγκεκριμένα τα δεδομένα που προκύπτουν από τον πίνακα – καταγράφονται σε ένα **προσωπικό αρχείο .xml** για να μπορούμε να τα προσπελάσουμε.. Καθώς λοιπόν προσπελαύνουμε τα δεδομένα από το αρχείο αυτό εκτελούμε για κάθε κόμβο μια εντολή Inset στην βάση μας

Η δεύτερη μέθοδος δέχεται σαν εισροή 4 αλφαριθμητικά και έναν ακέραιο τα οποία και περιγράφουν το αρχείο το οποίο θέλουμε να προσθέσουμε στην βάση μας.

Ø Download

Η συγκεκριμένη υπηρεσία παγκόσμιου ιστού αναλαμβάνει την εκτέλεση των ερωτημάτων αναζήτησης που θέτει ο χρήστης και την επιστροφή των αντίστοιχων αποτελεσμάτων

Περιλαμβάνει μια μέθοδο την **getFile()**. Η μέθοδος αυτή δέχεται σαν εισροή 4 αλφαριθμητικά – ένα για κάθε κριτήριο αναζήτησης το οποίο θέτει ο χρήστης .Στην συνέχεια αναλαμβάνει την σύνδεση με την βάση δεδομένων και ανάκτηση των δεδομένων από αυτή .Τέλος έχω την δημιουργία ενός xml εγγράφου με τα αποτελέσματα μου.

Ø Disconnect

Η συγκεκριμένη υπηρεσία παγκόσμιου ιστού αναλαμβάνει την επιτυχή αποσύνδεση ενός χρήστη από την υπηρεσία.

Περιλαμβάνει μια μέθοδο **Aposyndesi()** που δέχεται σαν εισροή ένα αλφαριθμητικό που αντιστοιχεί στο ψευδώνυμο του χρήστη που κάνει χρήση της υπηρεσίας. Στην συνέχεια ακολουθεί σύνδεση με την βάση δεδομένων. Κλείνοντας έχουμε εκτέλεση των ερωτημάτων διαγραφής από την βάση των εγγραφών που αφορούν τον συγκεκριμένο χρήστη. Τα ερωτήματα εκτελούνται με κλειδί το ψευδώνυμο του χρήστη.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ – ΑΝΑΛΥΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Όπως λοιπόν έχουμε προαναφέρει το σύστημά μου δομείται από μια αρχιτεκτονική 3 επιπέδων όπου στο επίπεδο 3 ορίζεται μια **σχεσιακή βάση δεδομένων**. Στην βάση αυτή αποθηκεύονται τα στοιχεία για τους συνδεδεμένους χρήστες .

Για την μοντελοποίηση της βάσης ορίζουμε 3 οντότητες.

- Ø Την οντότητα User- που αναφέρεται στους συνδεδεμένους χρήστες . Ο κάθε χρήστης διακρίνεται από έναν μοναδικό συνδυασμό **ψευδώνυμου (user name)** και **δικτυακής διεύθυνσης (ip address)**.
- Ø Την οντότητα Music File – που αναφέρεται στα αρχεία μουσικής που διαθέτουν οι χρήστες στην υπηρεσία είτε για download είτε για streaming. Το κάθε αρχείο διακρίνεται από έναν μοναδικό συνδυασμό **του ονόματος του και του χρήστη στον οποίο ανήκει (name, user_name)**.
- Ø Την οντότητα Video File – που αναφέρεται αντίστοιχα στα αρχεία video που διαθέτουν οι χρήστες στην υπηρεσία . Το κάθε αρχείο διακρίνεται από έναν μοναδικό συνδυασμό **του ονόματος του και του χρήστη στον οποίο ανήκει (name, user_name)**.

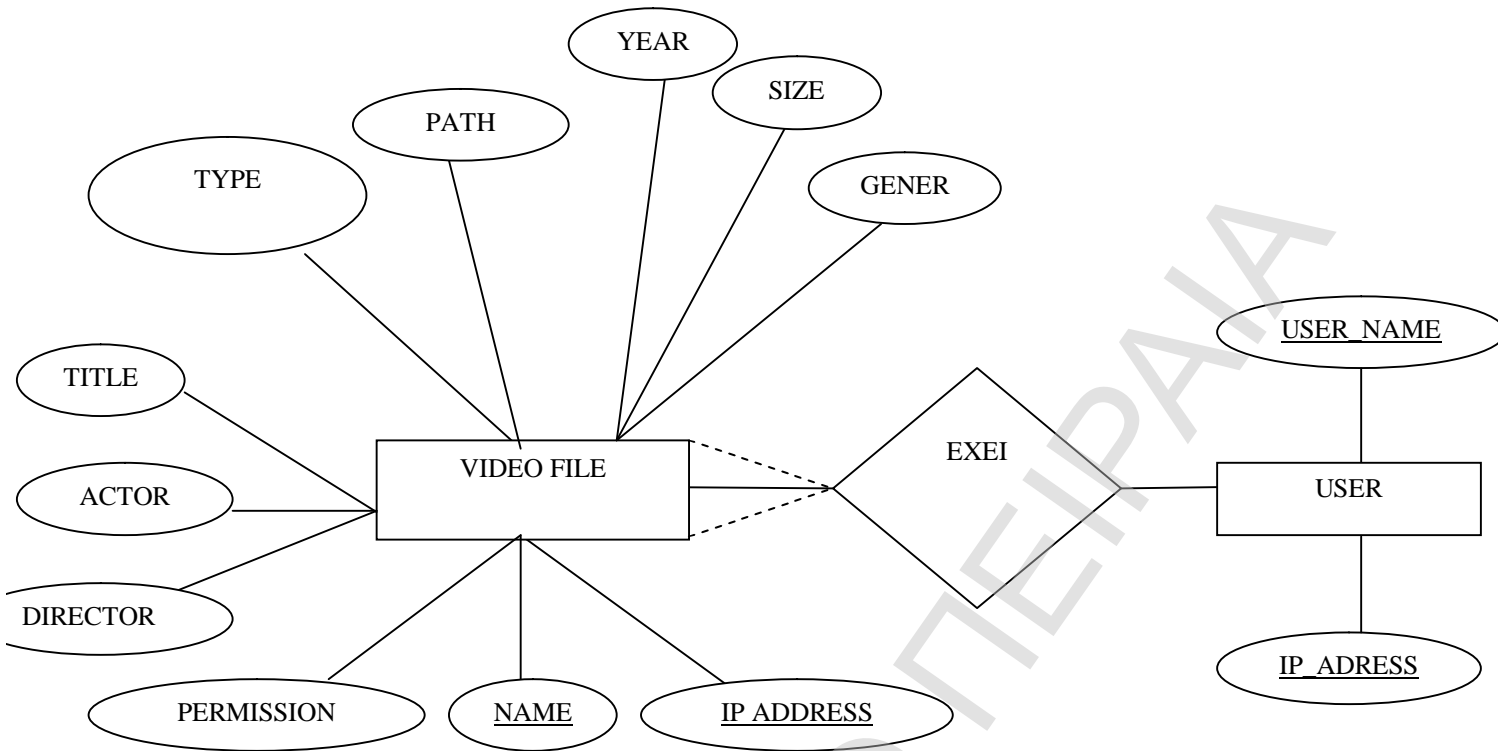
Διακρίνουμε 2 συσχετίσεις μεταξύ των οντοτήτων User και Music File και μεταξύ των User και Video File. Οι 2 αυτές συσχετίσεις είναι 1προς πολλά και περιγράφονται στο αντίστοιχο διάγραμμα οντοτήτων – Συσχετίσεων που ακολουθεί.

Σύμφωνα λοιπόν με τα παραπάνω δεν υπάρχει κάποιος περιορισμός σε ότι αφορά τα ονόματα των χρηστών (usernames)- καθώς 2 χρήστες που συνδέονται από διαφορετικά τερματικά μπορούν να χρησιμοποιούν το ίδιο ψευδώνυμο. Η ταυτοποίηση των χρηστών - όπως προαναφέραμε- προκύπτει από συνδυασμό του ψευδώνυμου και της δικτυακής διεύθυνσης του κάθε χρήστη.

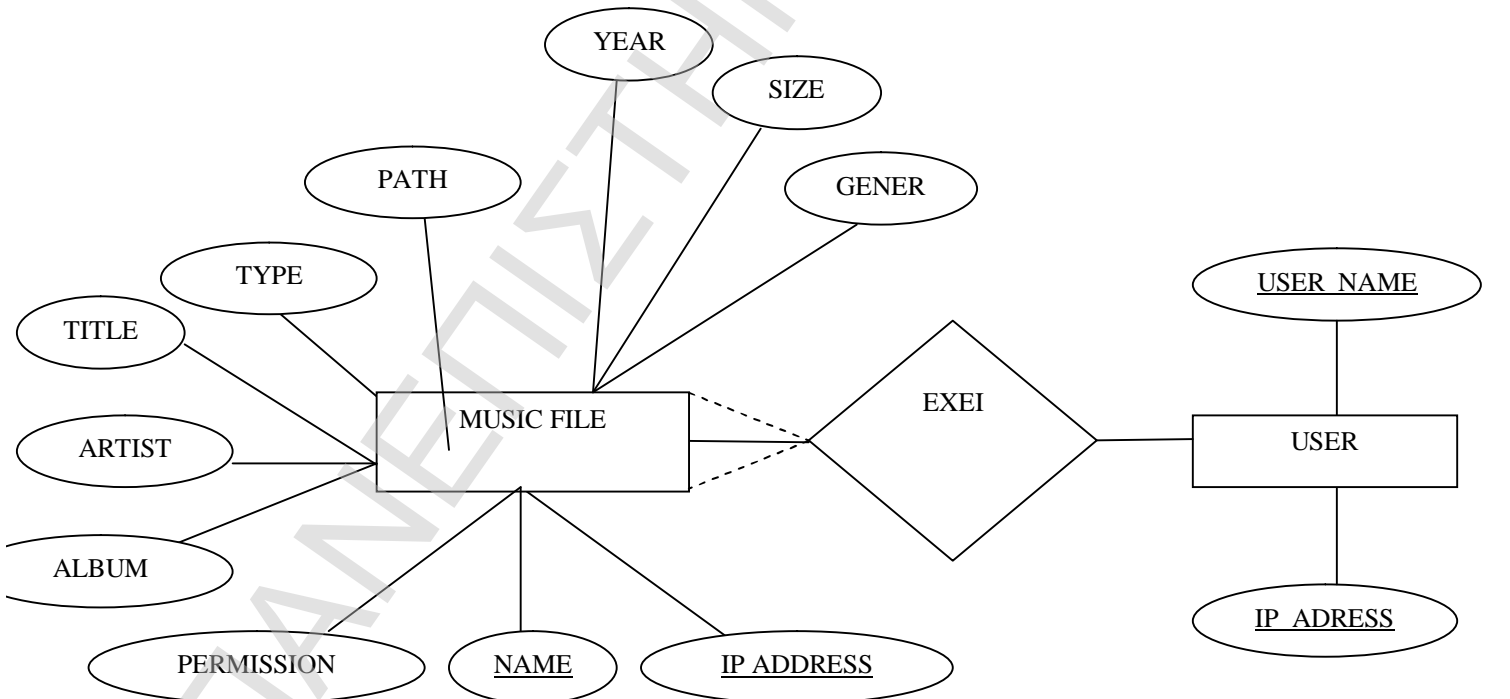
Συνεχίζοντας δεν υπάρχει επίσης κάποιος περιορισμός στο όνομα των αρχείων που διαμοιράζονται οι χρήστες. Η ταυτοποίηση ενός αρχείου γίνεται με βάση τον χρήστη στον οποίο ανήκει.

Καταλήγουμε λοιπόν σε ένα σχεσιακό μοντέλο με 3 πίνακες:

- Ø Τον πίνακα **Music File** όπου αποθηκεύονται στοιχεία για τα αρχεία μουσικής που διατίθενται από τους χρήστες είτε για streaming είτε για downloading. Αποθηκεύονται οι αντίστοιχες πληροφορίες για κάθε αρχείο όπως προκύπτουν από την βιβλιοθήκη του ιδιοκτήτη τους.
- Ø Τον πίνακα **Video File** όπου αποθηκεύονται στοιχεία για τα αρχεία βίντεο που διατίθενται από τους χρήστες είτε για streaming είτε για downloading Αποθηκεύονται οι αντίστοιχες πληροφορίες για κάθε αρχείο όπως προκύπτουν από την βιβλιοθήκη του ιδιοκτήτη τους.
- Ø Και τον πίνακα **User** όπου καταχωρούνται οι πληροφορίες για κάθε συνδεδεμένο χρήστη.



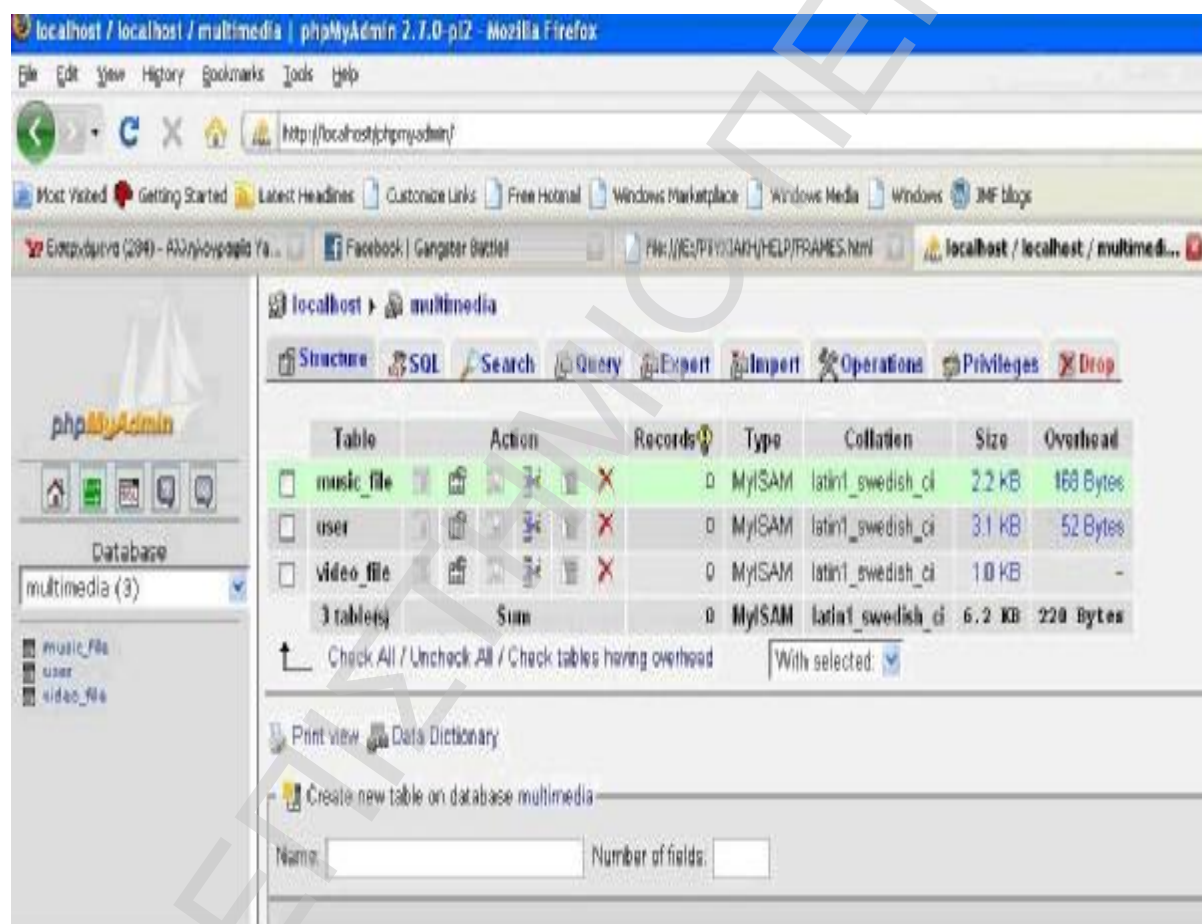
Σχήμα 2.1.15 – Διάγραμμα Οντοτήτων Συσχετίσεων



Σχήμα 2.1.16 – Διάγραμμα Οντοτήτων Συσχετίσεων

Για την υλοποίηση της Βάσης Δεδομένων που ορίσαμε πιο πάνω επιλέξαμε το Σύστημα Διαχείρισης Βάσεων Δεδομένων της **My SQL- έκδοση 5.0.18**. Πρόκειται για ένα από τα δημοφιλέστερα συστήματα βάσεων δεδομένων για ανάπτυξη διαδικτυακών εφαρμογών. Διατίθεται δωρεάν στο διαδίκτυο από την ιστοσελίδα της MySQL. Για περισσότερες πληροφορίες μπορεί κάποιος να ανατρέξει στο <http://www.mysql.com/>. Συνεργάζεται άψογα με τον εξυπηρετητή παγκόσμιου ιστού που επιλέξαμε – Apache Tomcat – ενώ παρέχει και πληθώρα γραφικών εργαλείων διαχείρισης και υποβολής ερωτημάτων.

Για την υλοποίηση της υπηρεσίας μας επιλέξαμε ένα από αυτά – **το phpMyAdmin** έκδοση 2.7.0. Πρόκειται για ένα εργαλείο γραμμένο στην γλώσσα προγραμματισμού PHP – το οποίο συνεπάγεται προς τρέχει πάνω από το πρωτόκολλο Http (Hyper Text Transfer Protocol).



Σχήμα 2.1.17 – Πίνακες της βάσης μας

Όπως μπορούμε να δούμε και από την παραπάνω εικόνα αποφασίσαμε να ονομάσουμε την βάση μας **multimedia** δηλώνοντας έτσι πως χρησιμοποιείται για την διαχείριση πληροφορίας πολυμεσικού περιεχομένου. Ο κώδικας δημιουργίας της βάσης βρίσκεται γραμμένος σε ένα αρχείο κειμένου με το όνομα **BASI.sql**.

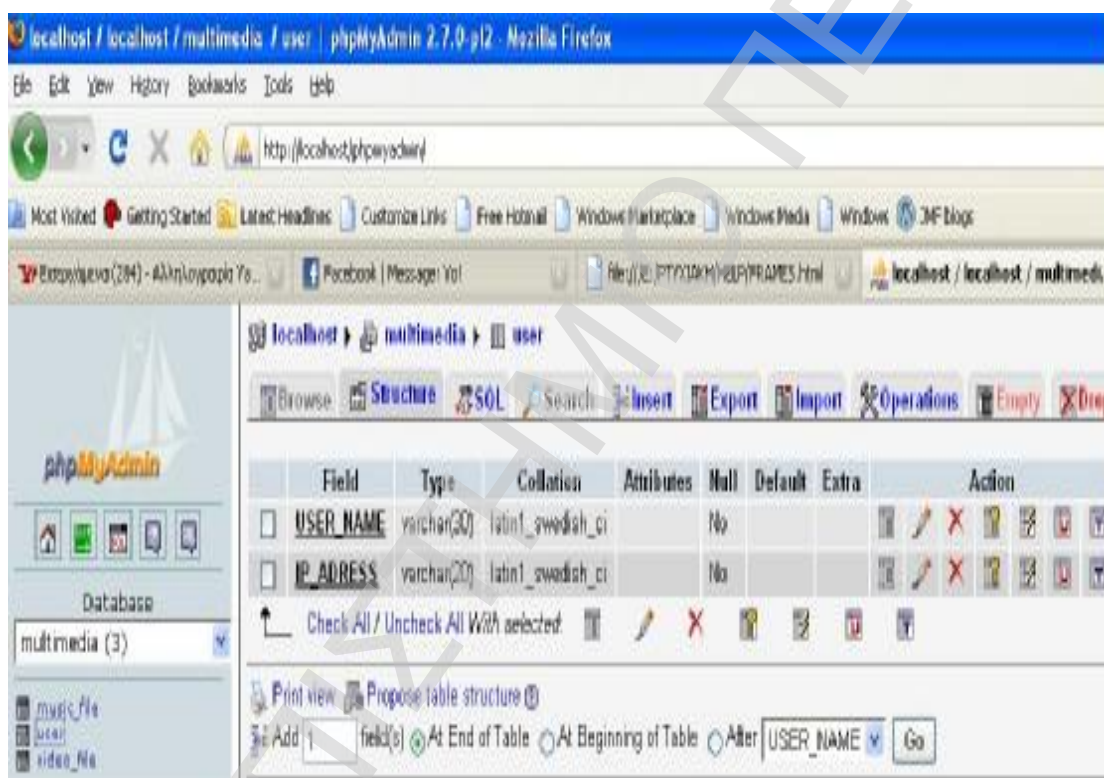
Παρακάτω ακολουθεί ανάλυση των 3 πινάκων που απαρτίζουν την βάση μας με τις αντίστοιχες οθόνες – ξεκινώντας από τον πίνακα των χρηστών.

Ø Πίνακας των Χρηστών

Στον πίνακα αυτό καταχωρούνται τα δεδομένα για κάθε χρήστη που συνδέεται με επιτυχία στην υπηρεσία μας. Όπως προκύπτει από το διάγραμμα οντοτήτων συσχετίσεων που ορίσαμε πιο πάνω η οντότητα χρήστη περιλαμβάνει 2 βασικά χαρακτηριστικά.

1. ψευδώνυμο και
2. δικτυακή διεύθυνση

Ο συνδυασμός των 2 αυτών αποτελεί και το κλειδί του πίνακα καθώς μπορεί να ορίσει έναν χρήστη μονοσήμαντα.



Σχήμα 2.1.18 – Πίνακας user

Όταν ένας χρήστης κάνει αίτηση αποσύνδεσης από την υπηρεσία, η υπηρεσία παγκόσμιου ιστού Disconnect εκτελεί ένα ερώτημα διαγραφής με βάση το ψευδώνυμο και την ip address του χρήστη. Γίνεται διαγραφή του χρήστη από τον πίνακα των χρηστών καθώς και διαγραφή των αρχείων μουσικής και βίντεο από τους αντίστοιχους πίνακες **με βάση τους περιορισμούς ξένου κλειδιού** – για τον συγκεκριμένο χρήστη

Ακολουθεί ανάλυση των πινάκων των αρχείων μουσικής και βίντεο αντίστοιχα

Ø Πίνακας αρχείων μουσικής

Στον πίνακα αυτό καταχωρούνται τα δεδομένα για κάθε αρχείο μουσικής που διαμοιράζεται κάθε χρήστης. Τα δεδομένα αυτά προκύπτουν από επεξεργασία του αντίστοιχου αρχείου βιβλιοθήκης από την υπηρεσία παγκόσμιου ιστού **Connect** που υλοποιεί την σύνδεση του χρήστη στην υπηρεσία. Όπως φαίνεται και από το διάγραμμα οντοτήτων συσχετίσεων για κάθε αρχείο έχω 11 χαρακτηριστικά.

1. όνομα
2. τίτλος
3. καλλιτέχνης
4. άλμπουμ
5. χρονολογία
6. μέγεθος αρχείου
7. μονοπάτι αρχείου (file path)
8. τύπος αρχείου
9. δικαιώματα αρχείου
10. είδος μουσικής
11. δικτυακή διεύθυνση

Όπου δικτυακή διεύθυνση αναφέρεται στην ip address Του χρήστη στον οποίο ανήκει το αρχείο. Επομένως πρόκειται για αναφορά ξένου κλειδιού στον πίνακα των χρηστών.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> NAME	varchar(60)	latin1_swedish_ci		No			[Icons]
<input type="checkbox"/> PATH	varchar(250)	latin1_swedish_ci		No			[Icons]
<input type="checkbox"/> SIZE	int(10)			No			[Icons]
<input type="checkbox"/> TYPE	varchar(20)	latin1_swedish_ci		No			[Icons]
<input type="checkbox"/> TITLE	varchar(60)	latin1_swedish_ci		Yes	NULL		[Icons]
<input type="checkbox"/> ARTIST	varchar(60)	latin1_swedish_ci		Yes	NULL		[Icons]
<input type="checkbox"/> ALBUM	varchar(60)	latin1_swedish_ci		Yes	NULL		[Icons]
<input type="checkbox"/> GENER	varchar(20)	latin1_swedish_ci		Yes	NULL		[Icons]
<input type="checkbox"/> YEAR	varchar(20)	latin1_swedish_ci		Yes	NULL		[Icons]
<input type="checkbox"/> IP_ADDRESS	varchar(20)	latin1_swedish_ci		No			[Icons]
<input type="checkbox"/> PERMISSION	varchar(20)	latin1_swedish_ci		No			[Icons]

Σχήμα 2.1.19 – Πίνακας music file

Ø Πίνακας αρχείων βίντεο

Στον πίνακα αυτό καταχωρούνται τα δεδομένα για κάθε αρχείο βίντεο που διαμοιράζεται κάθε χρήστης. Τα δεδομένα αυτά προκύπτουν από επεξεργασία του αντίστοιχου αρχείου βιβλιοθήκης από την υπηρεσία παγκόσμιου ιστού Connect που υλοποιεί την σύνδεση του χρήστη στην υπηρεσία. Όπως φαίνεται και από το διάγραμμα οντοτήτων συσχετίσεων για κάθε αρχείο έχω 11 χαρακτηριστικά.

1. όνομα
2. τίτλος
3. ηθοποιός
4. σκηνοθέτης
5. χρονολογία
6. μέγεθος αρχείου
7. τύπος αρχείου
8. δικαιώματα αρχεία
9. δικτυακή διεύθυνση
10. είδος ταινίας
11. μονοπάτι αρχείου (file path)

Όπου δικτυακή διεύθυνση αναφέρεται στην ip address Του χρήστη στον οποίο ανήκει το αρχείο. Επομένως πρόκειται για αναφορά ξένου κλειδιού στον πίνακα των χρηστών.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> NAME	varchar(60)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> PATH	varchar(250)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> SIZE	int(10)			No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> TYPE	varchar(20)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> TITLE	varchar(60)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> DIRECTOR	varchar(60)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> ACTOR	varchar(60)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> GENER	varchar(20)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> YEAR	varchar(20)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> IP_ADDRESS	varchar(20)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]
<input type="checkbox"/> PERMISSION	varchar(20)	latin1_swedish_ci		No			[edit] [delete] [insert] [refresh] [duplicate] [copy]

Σχήμα 2.1.20 – Πίνακας video file

ΜΕΡΟΣ ΤΡΙΤΟ

ΚΕΦΑΛΑΙΟ 1^ο – P2P ΔΙΚΤΥΑ

P2P ΔΙΚΤΥΑ ΕΙΣΑΓΩΓΗ

Σε αντίθεση με τα κλασικά κατανεμημένα συστήματα, τα **P2P δίκτυα** σκοπεύουν να συναθροίσουν έναν μεγάλο αριθμό από υπολογιστές οι οποίοι συνδέονται και αποσυνδέονται από το δίκτυο τακτικά. Στα αυθεντικά **P2P** συστήματα, ανεξάρτητοι υπολογιστές επικοινωνούν μεταξύ τους και μοιράζονται πληροφορίες **χωρίς να χρησιμοποιούν κάποιον εξυπηρετητή**. Ένα χαρακτηριστικό αυτής της οικογένειας συστημάτων είναι ότι "χτίζουν" στο επίπεδο εφαρμογής ένα ιδεατό δίκτυο με τους δικούς του μηχανισμούς δρομολόγησης. Η τοπολογία αυτού του ιδεατού δικτύου και οι μηχανισμοί δρομολόγησης που χρησιμοποιούνται έχουν σημαντική επίδραση στην αποδοτικότητα της εφαρμογής.

Σήμερα υπάρχουν διαθέσιμα μια πληθώρα προγραμμάτων **P2P (peer to peer)** για την ανταλλαγή αρχείων μεταξύ των χρηστών τους αρκετά από τα οποία βρίσκονται διαθέσιμα δωρεάν μέσω του διαδικτύου. Τα προγράμματα αυτά μπορούν να διακριθούν σε 2 μεγάλες κατηγορίες ανάλογα με την τεχνολογία ανταλλαγής αρχείων που χρησιμοποιούν.

Έχουμε 2 βασικά πρωτόκολλα:

Ø το **Gnutella**

Ø το **Fast Track**

Ακολουθεί μια σύντομη ανάλυση των 2 αυτών τεχνολογιών καθώς και παρουσίαση των βασικότερων προγραμμάτων (**Clients**) που χρησιμοποιούν κάθε μια από τις τεχνολογίες αυτές.

Εδώ αξίζει να σημειωθεί πώς έχει κατά καιρούς συνδεθεί η συγκεκριμένη τεχνολογία **P2P** συστημάτων με ζητήματα που αφορούν τα πνευματικά δικαιώματα για αρχεία μουσικής και ταινιών. Αρκετά λοιπόν από τα προγράμματα που θα αναφερθούν παρακάτω έχουν γίνει στόχος μηνύσεων από εταιρείες της μουσικής βιομηχανίας και του θεάματος και έχουν αναγκαστεί να έρθουν σε συμφωνία μαζί τους.

Στην συνέχεια κάνουμε μια σύνοψη στα βασικά χαρακτηριστικά του δικού μας **P2P** συστήματος τα οποία και το ξεχωρίζουν από τα υπόλοιπα.

ΤΕΧΝΟΛΟΓΙΑ GNUTELLA

Το πρωτόκολλο **Gnutella** είναι ένα ανοιχτό πρωτόκολλο που χρησιμοποιείται κυρίως για διαμοίραση αρχείων. Ο όρος **Gnutella** επίσης επιλέχθηκε για το ιδεατό δίκτυο από υπολογιστές που έχουν πρόσβαση στο διαδίκτυο και τρέχουν κάποια εφαρμογή που μιλάει το πρωτόκολλο **Gnutella**. Οι κόμβοι του **Gnutella**, ονομάζονται **servents** και εκτελούν λειτουργίες που σχετίζονται τόσο με εξυπηρετητές (**servers**) όσο και με πελάτες (**clients**). Παρέχουν διεπαφές μέσω των οποίων οι χρήστες μπορούν να κάνουν ερωτήσεις και να δουν τα αποτελέσματα, να δεχτούν ερωτήσεις από άλλους **servents**, να ελέγξουν αν μπορούν να απαντήσουν, και αν ναι, να στείλουν τα αποτελέσματα. Οι ίδιοι κόμβοι είναι υπεύθυνοι για την διαχείριση της κίνησης η οποία υπάρχει και χρησιμοποιείται για την ακεραιότητα του δικτύου.

Για να μπορέσει ένας νέος κόμβος να συνδεθεί στο **Gnutella** πρέπει πρώτα να συνδεθεί με έναν από ένα πλήθος γνωστών κόμβων οι οποίοι είναι πάντοτε διαθέσιμοι (π.χ **router.limewire.com**). Έχοντας συνδεθεί στο δίκτυο (έχοντας μία ή περισσότερες ανοιχτές συνδέσεις με κόμβους ήδη συνδεδεμένους στο **Gnutella**), οι κόμβοι στέλνουν μηνύματα για να αλληλεπιδρούν μεταξύ τους. Τα μηνύματα μπορεί να γίνονται **broadcast** (π.χ στέλνονται σε όλους τους κόμβους με τους οποίους ο αποστολέας έχει ανοιχτές συνδέσεις) ή απλά να διαδίδονται προς τα πίσω (**back-propagated**).

Αρκετά χαρακτηριστικά του πρωτοκόλλου επιτυγχάνουν αυτόν τον μηχανισμό **broadcast/back-propagation**. Πρώτον, κάθε μήνυμα έχει ένα μοναδικό αριθμό. Δεύτερον, κάθε κόμβος κρατάει στην μνήμη του τα πιο πρόσφατα μηνύματα που δρομολόγησε και χρησιμοποιείται για να αποτρέψει το **re-broadcasting** και να κάνει εφικτή την υλοποίηση του **back-propagation**. Τρίτον, τα μηνύματα έχουν ένα πεδίο **Time-to-live (TTL)** και ένα πεδίο "**Hops taken**".

Τα μηνύματα που επιτρέπονται στο δίκτυο είναι **PING**, **PONG**, **QUERY**, **QUERYHIT** και **PUSH**. Τα μηνύματα **PING** και **PONG** χρησιμοποιούνται για την συντήρηση του δικτύου ενώ τα **QUERY**, **QUERYHIT** και **PUSH** για την αναζήτηση αρχείων. Η παραλαβή των αρχείων γίνεται χρησιμοποιώντας το πρωτόκολλο **HTTP** με απευθείας σύνδεση των δύο κόμβων (αυτού που έχει το αρχείο και αυτού που θέλει να το αποκτήσει).

Ανακεφαλαιώνοντας, για να γίνει μέλος του δικτύου, ένας κόμβος πρέπει να ανοίξει μία ή περισσότερες συνδέσεις με κόμβους οι οποίοι είναι ήδη στο δίκτυο. Στο δυναμικό περιβάλλον όπου το δίκτυο **Gnutella** λειτουργεί, κόμβοι συχνά συνδέονται και αποσυνδέονται και οι συνδέσεις είναι αναξιόπιστες. Για να συμβιβαστεί με αυτό το περιβάλλον, αφού έχει συνδεθεί με το δίκτυο, ένας κόμβος περιοδικά στέλνει μηνύματα **PING** στους γείτονες του (γείτονες είναι οι κόμβοι με τους οποίους έχει ανοιχτές συνδέσεις) για να ανακαλύψει και άλλους κόμβους. Χρησιμοποιώντας αυτή την πληροφορία, ένας μη συνδεδεμένος κόμβος μπορεί σχεδόν πάντα να συνδεθεί με το δίκτυο. Οι κόμβοι αποφασίζουν που θα συνδεθούν στο δίκτυο βασιζόμενοι μόνο σε τοπική πληροφορία και έτσι συνθέτουν ένα δυναμικό δίκτυο από ανεξάρτητες οντότητες. Αυτό το ιδεατό

δίκτυο έχει τους **servents** για κόμβους του και τις ανοιχτές **TCP** συνδέσεις για συνδέσμους(**links**).

Ακολουθεί ένας πίνακας με τα βασικά χαρακτηριστικά των βασικότερων εφαρμογών που βασίζονται στην τεχνολογία **Gnutella**.

Name	Platform	License	Last Release	Heritage
Acquisition	Mac OS X		2.0 (build 209)	LimeWire
Apollon	Unix -KDE	GNU GPL	1.0.2.1 (2005-05-08)	giFT
BearFlix	Windows		7.0.0.56374 (22 October 2008; 12 days ago)	BearShare
BearShare	Windows	Proprietary	5.2.5.3	Original work
Cabos	Java	GNU GPL	0.8 (2008-09-27)	LimeWire
FilesWire (P2P)	Java	Proprietary	Beta 1.1 (2007)	Original Work
FrostWire	Java	GNU GPL	4.17.0	LimeWire
giFT	Cross-platform	GNU GPL	0.11.8.1 (2004-11-27)	Original Work
Gnucleus/GnucDNA	Windows	GNU GPL	2.2.0.0 (2005-06-17)	Original Work
Gtk-gnutella	Unix-like, Mac OS X	GNU GPL	0.96.5 (2008-04-02)	Original Work
iMesh	Windows		Unknown	GnucDNA
KCeasy	Windows	GNU GPL	Unknown	giFT
Kiwi Alpha	Windows		Unknown	GnucDNA

LimeWire	Java	GNU GPL	4.18.8	Original Work
Morpheus	Windows		5.5.1	GnucDNA
MP3 Rocket	Java	GNU GPL	5.0.3	LimeWire
Phex	Java	GNU GPL	3.2.4.105 (2008-06-15)	Original Work
Poisoned	Mac OS X	GNU GPL	0.5191 (August 8, 2006)	giFT
Shareaza	Windows	GNU GPL	2.4.0.0 (01 October 2008; 33 days ago)	Original Work
Symella	Symbian OS	GNU GPL	1.40 (2006-11-31)	Original Work
XFactor	Mac OS X	GNU GPL	0.93	giFT

ΤΕΧΝΟΛΟΓΙΑ FAST TRACK

Πρόκειται για μια τεχνολογία που προηγήθηκε χρονικά αυτής της **Gnutella** ενώ έπεται αυτής του [Napster](#) και έχει βασικό της εκπρόσωπο το λογισμικό [Kazaa](#). Το **2003** αποτελούσε το δημοφιλέστερο δίκτυο ανταλλαγής αρχείων με 2,4 εκατομμύρια ενεργούς χρήστες.

Δημιουργήθηκε τον Μάρτιο του **2001** από τον Σουηδό [Niklas Zennström](#), τον Δανό [Janus Friis](#) καθώς και μια ομάδα Εσθονών προγραμματιστών ηγούμενων από τον [Jaan Tallinn](#).

Το **FastTrack** είναι η αποκαλούμενη δεύτερη γενιά **P2P** πρωτοκόλλων. **It uses [supernodes](#) to improve scalability.**

Για να επιτρέψει το κατέβασμα αρχείων από πολλαπλές πηγές ο, **FastTrack** υλοποιεί τον αλγόριθμο κατακερματισμού [UUHash](#). Ενώ μπορεί να ελέγξει πολύ μεγάλα αρχεία σε μικρό σχετικά χρονικό διάστημα, μπορεί να επιτρέψει σε κάποια αρχεία να περάσουν χωρίς έλεγχο. Αρκετοί μεταξύ των οποίων, και η [RIAA](#), κατάφεραν να εκμεταλλευτούν την αδυναμία αυτή και να εισάγουν κατεστραμμένα αρχεία στο δίκτυο.

Γενικά οι τεχνολογία αυτή τα τελευταία έχει ατονήσει χάνοντας έδαφος ολοένα και περισσότερο προς όφελος των δικτύων τεχνολογίας **Gnutella**.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΑΣ

Τα συστήματα που χρησιμοποιούν τις 2 παραπάνω τεχνολογίες που αναλύσαμε είναι εξολοκλήρου **P2P** και δεν μεσολαβεί επικοινωνία καθόλου με κάποια κεντρική βάση δεδομένων ή κάποιον εξυπηρετητή.

Αντιθέτως θα μπορούσε να πει κάποιος πώς το σύστημά μας αποτελεί μια υβριδική περίπτωση **P2P** δικτύου. Η ανταλλαγή των αρχείων καθώς και το **streaming** γίνεται με τους 2 κόμβους να επικοινωνούν απευθείας. Όμως η διαχείριση της πληροφορίας κάθε κόμβου του δικτύου γίνεται κεντρικά με την μεσολάβηση κάποιου εξυπηρετητή παγκόσμιου ιστού και μιας βάσης δεδομένων. Αυτό καθιστά ευκολότερο τον έλεγχο των χρηστών που συνδέονται ως προς την ταυτότητά τους καθώς και το περιεχόμενο που διαθέτουν προς κοινή χρήση καθιστώντας το σύστημά μας έως ένα σημείο **ασφαλέστερο**.

Η αρχιτεκτονική αυτή του συστήματος μας δυστυχώς περιορίζει τον χρήστη στο κατέβασμα αρχείων από έναν συγκεκριμένο κόμβο κάθε φορά. Αυτό συνεπάγεται μείωση στην ταχύτητα μεταφοράς του αρχείου.

Το βασικό όμως χαρακτηριστικό που το κάνει να διαφέρει από αυτά των δικτύων **Gnutella** και **Fast Track** είναι ότι βασίζεται στην τεχνολογία **των υπηρεσιών παγκόσμιου ιστού (web services)**. Μέσω λοιπόν της δημοσίευσης των υπηρεσιών αυτών είναι δυνατή η υλοποίηση διαφορετικών **client** προγραμμάτων.

Ένα άλλο χαρακτηριστικό της εφαρμογής του χρήστη είναι ότι είναι γραμμένη εξ' ολοκλήρου σε **Java** κάτι που την καθιστά συμβατή με όλες της πλατφόρμες που υποστηρίζουν την τεχνολογία της **Java (cross platform)**.

ΚΕΦΑΛΑΙΟ 2^ο – ΔΥΣΚΟΛΙΕΣ ΥΛΟΠΟΙΗΣΗΣ & ΠΡΟΟΠΤΙΚΕΣ ΕΞΕΛΙΞΗΣ

ΔΥΣΚΟΛΙΕΣ – ΠΡΟΒΛΗΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Κατά διάρκεια εκπόνησης της παρούσας εργασίας προέκυψαν αρκετά ζητήματα σχεδίασης και υλοποίησης που έπρεπε να λυθούν. Στο κομμάτι αυτό λοιπόν θα κάνουμε μια σύντομη αναφορά στα βασικότερα από αυτά, στις αποφάσεις που εξήφθησαν καθώς και στα εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν για την επίλυση τους.

- Ø **Η πρώτη δυσκολία που προέκυψε ήταν η παραμετροποίηση του εξυπηρετητή για την υποστήριξη του SOAP και την δυνατότητα υλοποίησης υπηρεσιών παγκόσμιου ιστού.**
- Ø Το βασικότερο ζήτημα που προέκυψε στο ξεκίνημα της εργασίας ήταν η επιλογή της αρχιτεκτονικής του συστήματος μας. Έπρεπε να ληφθεί απόφαση εάν θα υπήρχε κεντρική αποθήκευση των δεδομένων των χρηστών σε κάποια βάση δεδομένων ή οι χρήστες θα αντάλλασαν τις πληροφορίες αυτές μεταξύ τους με τα κατάλληλα μηνύματα. Όπως έχουμε ήδη παρουσιάσει επιλέχθηκε η λύση της κεντρικής αποθήκευσης σε κάποια βάση δεδομένων.
- Ø Ένα από τα βασικότερα ζητήματα που προέκυψε ήταν ο τρόπος αποθήκευσης των αρχείων κάθε χρήστη καθώς και των αντίστοιχων μεταδιδόμενων για κάθε αρχείο. Θα έπρεπε με κάποιο τρόπο η πληροφορία αυτή να αποθηκεύεται κεντρικά στην βάση δεδομένων σύμφωνα με την αρχιτεκτονική που επιλέξαμε. Την λύση όπως έχουμε αναφέρει μας την έδωσε η XML.
- Ø **Η μεγαλύτερη δυσκολία υπήρξε στην προβολή των περιεχομένων της βιβλιοθήκης με λειτουργικότητα κουμπιών.** Όπως αναφέραμε πιο πάνω για την υλοποίηση της βιβλιοθήκης επιλέχθηκε η λύση της XML. Για την προβολή λοιπόν ενός xml εγγράφου η πιο εύκολη λύση είναι ο μετασχηματισμός του σε ένα html έγγραφο. Η λύση αυτή όμως παρουσιάζει περιορισμό στην προσθήκη κουμπιών με λειτουργικότητα για την κλήση των απαραίτητων κλάσεων JAVA. (Βλέπε κουμπιά download και streaming στο πίνακα προβολής της βιβλιοθήκης). Έτσι λοιπόν επιλέχθηκε η λύση της δημιουργίας JTABLES που παρέχεται στο πακέτο advanced swing.
- Ø Δυσκολία στην διαχείριση μνήμης σε ότι αφορά το video streaming. Για την υλοποίηση των συνδέσεων μεταξύ των χρηστών καθώς και την κλήση των απαραίτητων υπηρεσιών παγκόσμιου ιστού ήταν απαραίτητη η δημιουργία αρκετών νημάτων εκτέλεσης (threads) κάτι που έκανε την διαχείριση μνήμης αρκετά πιο δύσκολη.

- Ø Άλλη μια δυσκολία που συναντήσαμε ήταν η **ελλιπής υποστήριξη από την Sun Microsystems της πλατφόρμας του JMF**. Αποτέλεσμα αυτής ήταν η μη υποστήριξη αρκετών δημοφιλών τύπων αρχείων πολυμέσων.
- Ø Τέλος δυσκολία προέκυψε και με τον χρονοισμό πομπού και δέκτη σε ότι αφορά την διαδικασία του streaming καθώς και στον τερματισμό των συνόδων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

ΠΡΟΟΠΤΙΚΕΣ ΕΞΕΛΙΞΗΣ ΚΑΙ ΒΕΛΤΙΩΣΗΣ

Στο κομμάτι αυτό ακολουθεί μια λίστα με ενδεχόμενες βελτιώσεις και επεκτάσεις που θα μπορούσαν να γίνουν στην υπηρεσία μας εξελίσσοντάς την ακόμα περισσότερο:

- Ø Βελτιστοποίηση κώδικα Client εφαρμογής για την καλύτερη χρήση και διαχείριση μνήμης και bandwidth.
- Ø Προσθήκη δυνατότητας καταγραφής και ελέγχου συμβάντων με την διατήρηση ανάλογων αρχείων (log files).
- Ø Πιθανή μεταφορά της διαδικασίας εγκαθίδρυσης συνδέσεων μεταξύ των χρηστών στο επίπεδο των υπηρεσιών παγκόσμιου ιστού. Μια τέτοια ενέργεια θα καθιστούσε τον πρόγραμμα Client αρκετά πιο ελαφρύ μεταφέροντας φόρτο στον εξυπηρετητή παγκόσμιου ιστού.
- Ø Προσθήκη δυνατότητας επικοινωνίας χρηστών με μηνύματα κειμένου με την ενοποίηση αντίστοιχης chat εφαρμογής. Μια τέτοια εφαρμογή υπάρχει ήδη υλοποιημένη σε Java από τον συγγραφέα της παρούσας εργασίας και θα μπορούσε να ενσωματωθεί με την κατάλληλη τροποποίηση.
- Ø Προσθήκη δυνατότητας προσωρινής αποθήκευσης των ανταλλασσόμενων αρχείων καθώς και κατέβασμα από πολλαπλές πηγές.
- Ø Πιθανή δυνατότητα επιλογής άλλης πιο αξιόπιστης πλατφόρμας για την λειτουργία του streaming.

ΒΙΒΛΙΟΓΡΑΦΙΑ

URL's

1. <http://ws.apache.org/axis/>
2. <http://java.sun.com/>
3. <http://java.sun.com/javase/technologies/desktop/media/jmf/>
4. <http://www.mysql.com/>
5. http://www.phpmyadmin.net/home_page/index.php
6. <http://www.w3.org/TR/soap/>
7. <http://www.w3schools.com/soap/default.asp>
8. <http://www.w3schools.com/xml/default.asp>
9. http://www.go-online.gr/ebusiness/specials/article.html?article_id=213
10. <http://www.3cx.gr/voip-sip/rtp.php>
11. <http://java.sun.com/products/java-media/jmf/2.1.1/guide/index.html>
12. <http://www.freegr.gr/freenuke/modules.php?name=News&file=article&sid=170>
13. <http://www.bengoodger.com/software/webservices/webservices.html>

ΣΥΓΓΡΑΜΜΑΤΑ -ΔΗΜΟΣΙΕΥΣΕΙΣ

1. **Media Streaming με χρήση Java και JMF** – Πανεπιστήμιο Μακεδονίας -Τμήμα: Εφαρμοσμένης Πληροφορικής -Μάθημα: Ανάπτυξη δικτύων - **ΚΑΤΕΡΤΖΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ** - Επιβλέπων Καθηγητής: Π. Φουληράς
2. «**Τεχνολογία Web Services**» - ΤΕΙ Μεσολογγίου – Σχολή Διοίκησης & Οικονομίας –Τμήμα Εφαρμογών Πληροφορικής στην Διοίκηση και

την Οικονομία – Κυριακούλιας Κωνσταντίνος, Ντελής Επαμεινώνδας - Επιβλέπων Καθηγητής: Συρμακέσης Σπύρος

3. Διαχείριση ψηφιακών αντικειμένων –Σχεδιασμός, Ανάπτυξη και Υλοποίηση Συστήματος – Πανεπιστήμιο Πατρών – Σαλούρος Δημήτριος
4. **Java Video and Audio in Consumer Devices: JMF and MM API** - Helsinki University of Technology - Pablo César and J. Luc Lamadon
5. **JAVA MEDIA FRAMEWORK – JMF** – Πανεπιστήμιο Αιγαίου – Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων –Στάρδης Μανώλης
6. Ανάπτυξη Εφαρμογής Πραγματικού Χρόνου για επικοινωνία πελατών μέσω Διαδικτύου – Μεταπτυχιακό Πρόγραμμα σπουδών Τμήματος Πληροφορικής Πανεπιστημίου Μακεδονίας – Υφαντής Γεώργιος Θεσσαλονίκη 2007
7. Ανακάλυψη της τοπολογίας του δικτύου Gnutella και μελέτη της απόδοσης του - Πανεπιστήμιο Κρήτης - Τμήμα Επιστήμης Υπολογιστών - Κωνσταντίνος Ξυνίδης 2002
8. **ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΛΟΜΕΝΩΝ – Ramakrishnan Gehrke** – Εκδόσεις Τζιόλα
9. **Java Network Programming 3rd Edition** - O'Reilly - Elliotte Rusty Harold
10. **Core Java™ 2 Volume I** - Prentice Hall PTR –Cay S. Horstmann, Gary Cornell
11. **Core Java™ 2 Volume I** - Prentice Hall PTR –Cay S. Horstmann, Gary Cornell
12. Πολυμέσα Θεωρία και Πράξη – Κωδικοποίηση Μέσων και Επεξεργασία Περιεχομένου – Ralf Steinmetz, Klara Nahrstedt – Εκδόσεις Γκιούρδας
13. **Multimedia στη Θεωρία και στην Πράξη -2^η έκδοση**-Εκδόσεις Τζιόλα – Ομάδα Εργαστηρίου Πολυμέσων Α.Π.Θ
14. **Οδηγός της XML με παραδείγματα** – Benoit Marchal – Εκδόσεις Γκιούρδας