



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Πλατφόρμας Αναζήτησης Υπηρεσιών Διαδικτύου με Χρήση Ομότιμων Οντοτήτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιουλιανός Γ. Νότος

Επιβλέπων : Ιάκωβος Στ. Βενιέρης
Αναπλ. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2005



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη Πλατφόρμας Αναζήτησης Υπηρεσιών Διαδικτύου με Χρήση Ομότιμων Οντοτήτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιουλιανός Γ. Νότος

Επιβλέπων : Ιάκωβος Στ. Βενιέρης
Αν. Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 4^η Φεβρουαρίου 2005.

.....
Ι. Βενιέρης
Αν. Καθηγητής Ε.Μ.Π.

.....
Γ. Στασινόπουλος
Καθηγητής Ε.Μ.Π.

.....
Δ. Κακλαμάνη
Επ. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2005

.....
Ιουλιανός Γ. Νότος

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

Copyright © Ιουλιανός Νότος, 2005

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Ο σκοπός της διπλωματικής εργασίας αυτής ήταν η ανάπτυξη μιας πλατφόρμας αναζήτησης υπηρεσιών Διαδικτύου με χρήση ομότιμων οντοτήτων. Τα τελευταία χρόνια έχουν μπει δυναμικά στο προσκήνιο υπηρεσίες που προσφέρονται μέσω διαδικτύου. Οι λεγόμενες υπηρεσίες διαδικτύου (web services), αποτελούν μια εξελιγμένη μορφή τέτοιων υπηρεσιών. Πρόκειται για υπηρεσίες οι οποίες περιγράφονται με την χρήση της εκτατής γλώσσας σήμανσης XML και που μπορούν να εκτελεστούν σε οποιαδήποτε πλατφόρμα λογισμικού. Ο κύκλος ζωής μιας τέτοιας υπηρεσίας αφορά την δημιουργία, την ανακοίνωση, την ανακάλυψη και την χρησιμοποίησή της. Οι δύο πρώτες λειτουργίες αφορούν τους παρόχους υπηρεσιών ενώ οι δύο επόμενες τους χρήστες – καταναλωτές αυτών.

Συγκεκριμένα στην διπλωματική αυτή προτείνουμε ένα καινούργιο τρόπο για την ανακάλυψη υπηρεσιών διαδικτύου. Κάνοντας χρήση τεχνολογίας ομότιμων οντοτήτων (Peer-to-Peer) δημιουργούμε μια πλατφόρμα στην οποία οι παραγωγοί ανακοινώνουν τις υπηρεσίες που προσφέρουν ενώ οι καταναλωτές από την άλλη ανακαλύπτουν και χρησιμοποιούν την υπηρεσία που επιθυμούν. Επιπλέον ο εκάστοτε πελάτης της υπηρεσίας δεν χρειάζεται να γνωρίζει τίποτε ως προς το πώς θα χρησιμοποιήσει την υπηρεσία που επιθυμεί. Ο τρόπος κλήσης της υπηρεσίας δημιουργείται αυτόματα αφού βρεθεί η υπηρεσία.

Η υλοποίηση της πλατφόρμας αυτής γίνεται σε γλώσσα προγραμματισμού Java και με τη χρήση της Java υλοποίησης του πλαισίου JXTA.

Λέξεις κλειδιά: υπηρεσίες διαδικτύου, XML, ομότιμες οντότητες, ανακάλυψη υπηρεσιών, Java, JXTA.

Abstract

The aim of this thesis was the development of a platform for discovering web services using peer-to-peer.

The last years have entered dynamically in the limelight services that are offered via the Internet. The so-called web services constitute an evolved form of such services. These services are described with the usage of Extensible Markup Language, XML, and they can be executed in any software platform. The life circle of a service of that kind concerns its creation, its statement, its discovery and utilization. The first two operations concern service providers and the two next functions concern their consumers.

Concretely in this thesis we propose a new way for discovering web services. Using the technology of peer entities (Peer-to-Peer) we create a platform in which providers announce the services that offer while the consumers on the other hand discover and use the service they wish. Moreover each service's customer does not need to know nothing as for how will he use the service he wishes. The service invocation is created automatically after its discovery.

The concretization of this platform becomes in Java programming language, using the Java implementation of JXTA framework.

Keywords: web services, XML, peer entities, service discovery, Java, JXTA.

Αφιερώνεται
στον πατέρα μου Γιώργο,
τη μητέρα μου Ρούσα και
τον αδερφό μου Ανέστη.

Ευχαριστίες

Ένα μεγάλο ευχαριστώ χρωστάω στην οικογένειά μου για την καλή ανατροφή που μου προσέφερε!!!

Στους καθηγητές μου, όλους μαζί κι έναν προς έναν ξεχωριστά, για τη μόρφωση που μου προσέφεραν, όλα αυτά τα χρόνια στο Πολυτεχνείο, και κυρίως τον κύριο Ι. Βενιέρη για την καλή μας συνεργασία.

Επιπλέον στον κύριο Λ. Κουτσολουκά για τις συζητήσεις, τις παρατηρήσεις και τις συμβουλές.

ΠΕΡΙΕΧΟΜΕΝΑ

Γενικές Πληροφορίες	Σελίδα
1.1 Εισαγωγή	14
1.2 Υπηρεσίες Διαδικτύου, ως τώρα	14

ΜΕΡΟΣ Ι – ΣΥΝΘΕΣΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΔΙΑΔΙΚΤΥΟΥ

1. Εισαγωγή	22
2. Η εκτατή γλώσσα σήμανσης (Extensible Markup Language – XML)	22
2.1 Επικύρωση εγγράφων XML	24
2.2 Ανάλυση XML εγγράφων	27
2.3 Παρουσίαση – μετασχηματισμοί XML	27
3. Πρωτόκολλο πρόσβασης απλού αντικειμένου (Simple Object Access Protocol – SOAP)	30
3.1 Μηνύματα SOAP	31
3.2 SOAP σε δράση	33
4. Γλώσσα περιγραφής υπηρεσιών Διαδικτύου (Web Service Definition Language – WSDL)	35
4.1 Περιγραφή ανάπτυξης υπηρεσίας Διαδικτύου (Web Service Deployment Descriptor WSDD)	38
5. Παγκόσμια Περιγραφή Ανακάλυψη και Ολοκλήρωση (Unified Description, Discovery & Integration – UDDI)	38

ΜΕΡΟΣ ΙΙ – ΤΕΧΝΟΛΟΓΙΑ ΟΜΟΤΙΜΩΝ ΟΝΤΟΤΗΤΩΝ

1. Εισαγωγή	42
1.1 Από την αρχιτεκτονική πελάτης εξυπηρετητής στην τεχνολογία ομότιμων οντοτήτων (Peer-to-Peer –P2P)	43
1.2 Περιγραφή της τεχνολογίας P2P	47
1.3 Γιατί είναι σημαντική η τεχνολογία P2P	50
2. Η εξέλιξη της τεχνολογίας P2P	51
2.1 Στιγμαϊαία μηνύματα	52
2.2 Διανομή αρχείων	53
2.3 Κατανεμημένος υπολογισμός	54
3. Εισαγωγή στην πλατφόρμα JXTA	54
3.1 Επισκόπηση της πλατφόρμας JXTA	57
3.2 Ομότιμες οντότητες (peers)	58
3.3 Ομάδες ομότιμων οντοτήτων (peer groups)	59
3.4 Μεταφορά στο δίκτυο	60
3.5 Υπηρεσίες	61
3.6 Διαφημίσεις	61
3.7 Πρωτόκολλα	62
3.8 Ονομασία οντοτήτων	62
4. Πρωτόκολλα της JXTA	64
4.1 Πρωτόκολλο τελικού σημείου	64
4.1.1 Επισκόπηση πρωτοκόλλου	65
4.1.2 Υπηρεσία τελικού σημείου	66
4.1.3 Αποστολή μηνύματος	66
4.1.4 Σύνδεση του πρωτοκόλλου τελικού σημείου με τη Java	67
4.2 Πρωτόκολλο για ραντεβού	67
4.2.1 Διαφημίσεις Ραντεβού	68
4.2.2 Διάδοση μηνυμάτων	69
4.3 Πρωτόκολλο επίλυσης ομότιμων οντοτήτων	69
4.3.1 Ονομασία χειριστών	70
4.3.2 Μηνύματα ερώτησης	71
4.3.3 Μηνύματα απάντησης	72
4.3.4 Σύνδεση με τη Java	73

4.4 Πρωτόκολλο πληροφόρησης ομότιμων οντοτήτων	73
4.4.1 Μηνύματα ερώτησης	74
4.4.2 Μηνύματα απάντησης	74
4.4.3 Σύνδεση με τη Java	76
4.5 Πρωτόκολλο δέσμευσης σωλήνα	77
4.5.1 Επισκόπηση πρωτοκόλλου	77
4.5.2 Διαφήμιση σωλήνα	78
4.5.3 Σύνδεση με τη Java	79
4.6 Πρωτόκολλο ανακάλυψης	80
4.6.1 Επισκόπηση πρωτοκόλλου	80
4.6.2 Μηνύματα αναζήτησης	81
4.6.3 Μηνύματα απάντησης	82
4.6.4 Πολιτικές και ποιότητα υπηρεσίας	83
4.6.5 Σύνδεση της Java με το πρωτόκολλο ανακάλυψης	84
 ΜΕΡΟΣ ΙΙΙ – ΑΝΑΠΤΥΞΗ ΠΛΑΤΦΟΡΜΑΣ	
1. Εισαγωγή	87
2. Επισκόπηση αρχιτεκτονικής	87
2.1 Δημιουργία διαφήμισης για τις υπηρεσίες Διαδικτύου	89
2.2 Ανταλλαγή μηνυμάτων για λήψη πληροφοριών	91
2.3 Πολιτικές ασφαλείας και ποιότητα υπηρεσίας	92
3. Υλοποίηση στη Java	93
3.1 Αναζήτηση ομάδας	94
3.2 Εγγραφή	95
3.3 Ανακοίνωση – αναζήτηση διαφήμισης	95
3.4 Διευκρίνιση διαφήμισης	95
3.5 Λειτουργία αναφοράς στην χρήση της υπηρεσίας	96
3.6 Σύνθεση της πλατφόρμας και επίδειξη	97
4. Σχεδιαγράμματα UML	104
4.1 Διάγραμμα περιπτώσεων χρήσης	104
4.2 Ακολουθιακό διάγραμμα	104
4.2 Διάγραμμα δραστηριοτήτων	105
5. Ανοιχτά θέματα	107
 ΠΑΡΑΡΤΗΜΑ	109
 ΑΝΑΦΟΡΕΣ	113
 ΒΙΒΛΙΟΓΡΑΦΙΑ	115

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΜΕΡΟΣ Ι

2.1 Περίπτωση βιογραφικού σημειώματος	23
2.2 Επικύρωση με DTD	25
2.3 Τροποποίηση του αρχικού βιογραφικού για να δεχθεί επικύρωση με DTD	25
2.4 Επικύρωση με XML-Schema	26
2.5 Τροποποίηση του αρχικού βιογραφικού για να δεχθεί επικύρωση με XML-Schema	27
2.6 XSL μετασχηματισμοί	28
2.7 Παρουσίαση του XML βιογραφικού ως HTML	28
2.8 XSL που μετασχηματίζει το αρχικό XML	29
3.1 Διαστρωμάτωση α – κωδικοποίηση β – μεταφοράς στο Διαδίκτυο	30
3.2 Δομή ενός μηνύματος SOAP	31
3.3 Μορφή ενός φακέλου SOAP	31
3.4 Μορφή μιας επικεφαλίδας SOAP	32
3.5 Παράδειγμα αναφοράς λάθους	32
3.6 SOAP σε δράση, γενική εικόνα.	33
3.7 Ένα ολόκληρο SOAP μήνυμα αίτησης για την χρησιμοποίηση της μεθόδου gateDate της υπηρεσίας JuliansDateServer για την ώρα. Τα στοιχεία που βρίσκονται στην κορυφή είναι τα στοιχεία επικεφαλίδας HTTP Headers	33
3.8 Απάντηση στο μήνυμα της προηγούμενης αίτησης για την ώρα, που προσφέρει η υπηρεσία JuliansDateServer μέσω της μεθόδου της getDate	34
3.9 SOAP φάκελος αίτησης ζητώντας την τιμή της μετοχής, με σύμβολο “SDAQ”, στη μέθοδο getQuote της υπηρεσίας StockQuoteService που προσφέρει ενημέρωση μετοχών, α) πάνω αίτηση, β) κάτω αναφορά σφάλματος	34
4.1 Δομή ενός WSDL εγγράφου	37
5.1 Βασικές δομές δεδομένων και μεταξύ τους σχέσεις στο UDDI	39
5.2 Η UDDI και ο ρόλος της στην ανακάλυψη υπηρεσιών Διαδικτύου	40

ΜΕΡΟΣ ΙΙ

1.1 Το Διαδίκτυο, συνένωση όλων των δικτύων	43
1.2 Αρχιτεκτονική πελάτης εξυπηρετητής	45
1.3 Επίπεδη αρχιτεκτονική με χρήση ομότιμων οντοτήτων	46
1.4 Λειτουργία DNS	48
1.5 Δακτύλιος με κουπόνι	49
1.6 Συγκεντρωτική τοπολογία	49
1.7 Κατανεμημένη τοπολογία	50
1.8 Υβριδική τοπολογία	50
3.1 Η διαστρωμάτωση των πρωτοκόλλων της JXTA	56
3.2 Η αρχιτεκτονική της JXTA	57
3.3 Παρουσίαση γενικού τρόπου σχηματισμού JXTA id κατά ABNF	64
3.4 Ειδικά jxta id για την Java υλοποίηση της JXTA παρουσίαση κατά ABNF.	65
4.2.1 XML αναπαράσταση διαφήμισης ραντεβού	69
4.2.2 Η δομή που ακολουθεί το μήνυμα ραντεβού	70
4.3.1 Ο τρόπος σχηματισμού του ονόματος χειριστή	71
4.3.2 Τρόπος σχηματισμού ερώτησης, πρωτόκολλο επίλυσης	72
4.3.3 Τρόπος σχηματισμού απάντησης, πρωτόκολλο επίλυσης	73
4.4.1 Τρόπος σχηματισμού ερώτησης, πρωτόκολλο πληροφόρησης	75
4.4.2 Τρόπος σχηματισμού απάντησης, πρωτόκολλο πληροφόρησης	76
4.5.1 Τρόπος σχηματισμού ενός σωλήνα	79
4.5.2 Τρόπος σχηματισμού μηνύματος επίλυσης σωλήνα	79
4.6.1 Τρόπος σχηματισμού ερώτησης αναζήτησης	82
4.6.2 Τρόπος σχηματισμού απάντησης ερώτησης αναζήτησης	83

ΜΕΡΟΣ ΙΙΙ

2.1 Πλατφόρμα αναζήτησης υπηρεσιών Διαδικτύου με χρήση P2P	88
2.2 Τρόπος σχηματισμού διαφημίσεων για υπηρεσίες Διαδικτύου	89
2.3 Τρόπος σχηματισμού μηνύματος αίτησης προς τον προμηθευτή	91
3.1 Συναλλαγή με ομάδες ομότιμων οντοτήτων	94
3.2 Τρόπος δημιουργίας μιας διαφήμισης σωλήνα	95
3.3 WSDL έγγραφο που περιγράφει την υπηρεσία για την ώρα	98
3.4 Διαφήμιση που χρησιμοποιήθηκε για την υπηρεσία	98
3.5 Η έξοδος από το πρόγραμμα διεπαφής χρήστη	99
3.6 WSDL έγγραφο που περιγράφει την υπηρεσία για τα ζώδια	99
3.7 Διαφήμιση που χρησιμοποιήθηκε για την υπηρεσία για τα ζώδια	100
3.8 Αλληλεπίδραση του πελάτη με την υπηρεσία διαδικτύου	100
3.9 WSDL έγγραφο που περιγράφει την υπηρεσία για την ενημέρωση σχετικά με μετοχές	101
3.10 Πρόγραμμα διεπαφής του προμηθευτή των υπηρεσιών	102
3.11 Πρόγραμμα πελάτη για αναζήτηση και χρησιμοποίηση	103
4.1 Σχεδιάγραμμα χρήσης	104
4.2 Γενικό ακολουθιακό διάγραμμα χρησιμοποίησης υπηρεσιών διαδικτύου	105
4.3 Ακολουθιακό διάγραμμα αναζήτησης υπηρεσιών διαδικτύου στο P2P δίκτυο	106
4.4 Διάγραμμα δραστηριοτήτων α) προμηθευτή, β) καταναλωτή	107

ΕΙΣΑΓΩΓΗ

1. Εισαγωγή

Η καθιέρωση μιας εταιρικής παρουσίας στο Διαδίκτυο, μέσω μιας εταιρικής πύλης Διαδικτύου, και η ανάπτυξη των διαύλων επικοινωνίας μεταξύ εταιρειών και πελατών είναι και χρονοβόρα και αρκετά δαπανηρή διαδικασία. Έπειτα από την επίτευξη αυτών, θα πρέπει οι επιχειρήσεις να προσφέρουν αρκετό περιεχόμενο, το οποίο να το ενημερώνουν τακτικά, και ιδιαίτερα χαρακτηριστικά γνωρίσματα από την δικτυακή τους πύλη ώστε να ενθαρρύνουν επισκέπτες, πελάτες ή τους συνεργάτες τους να επιστρέψουν και να επισκεφτούν ξανά της πύλη τους αυτή.

Οι περισσότεροι ιστοχώροι, οι οποίοι προσελκύουν πολλούς επισκέπτες χρησιμοποιούν τεχνικές συλλογής περιεχομένου για να εμπλουτίσουν την προσφορά τους. Για να επεκτείνουν τις λειτουργίες τους βασίζονται στις υπηρεσίες διαδικτύου. Γνωστά παραδείγματα είναι το Yahoo! [URL-1], η Netscape[URL-2], MSN [URL-1] κτλ – τα περισσότερα από τα χαρακτηριστικά γνωρίσματά τους προέρχονται από άλλους ιστοχώρους: ταξίδια, ενημέρωση, καιρός, αγορές, χάρτες και αναζητήσεις πληροφοριών χρησιμοποιώντας τις υπηρεσίες της Google μηχανής αναζήτησης. Αυτά τα χαρακτηριστικά γνωρίσματα είναι ενσωματωμένα σε δικτυακές πύλες, λ.χ. στη www.yahoo.com, ως εγγενείς υπηρεσίες που προσφέρονται σε πληθώρα επισκεπτών ανά τον κόσμο και σε καθημερινή βάση. Με αυτόν τον τρόπο ο κάθε προμηθευτής, διαθέτοντας την υπηρεσία του, π.χ. στο Yahoo!, μπορεί να εκθέσει τον εαυτό του σε περισσότερους υποψήφιους πελάτες και να αυξήσει τα εισοδήματά του, επιτρέποντας παράλληλα στο Yahoo! να προσφέρει περισσότερες υπηρεσίες στους χρήστες του.

Το απλό αυτό παράδειγμα επεξηγεί τη βασική έννοια των υπηρεσιών Διαδικτύου. Το πρόβλημα είναι ότι, μέχρι τώρα, η ενσωμάτωση υπηρεσιών από τις διαφορετικές επιχειρήσεις ήταν πολύ δύσκολη υπόθεση. Ενώ τα περίπλοκα μεσομικά (middleware) εξωτερικής ολοκλήρωσης είναι πολύ χρήσιμα για την ένα προς ένα ολοκλήρωση, η πολλαπλή ολοκλήρωση είναι ένα διαφορετικό επίπεδο σκέψης που απαιτεί συγκεκριμένη ανάπτυξη για κάθε νέα υπηρεσία για να ενσωματωθεί [IJEC][URL-18]. Ομοίως, για τους φορείς παροχής υπηρεσιών, η πρόσθεση ενός νέου συνεργάτη να σημαίνει απλή προσαρμογή του πηγαίου του κώδικα στη υπάρχουσες υπηρεσίες που προσφέρει [KYEE].

Οι καλές ειδήσεις είναι ότι οι υπηρεσίες διαδικτύου δεν είναι πλέον μόνο μια έννοια αλλά μια τεχνολογική πραγματικότητα. Τα πρώτα κεφάλαια αυτής της ιστορίας έχουν γραφτεί ήδη από μερικούς προμηθευτές λογισμικού (IBM, Microsoft, Sun Microsystems, Apache).

1.1 Υπηρεσίες Διαδικτύου, έως τώρα

Οι απόψεις για το τι είναι οι υπηρεσίες διαδικτύου είναι πάμπολλες και συχνά διαφορετικές. Σύμφωνα με τον οργανισμό World Wide Web Consortium (W3C)[URL-15], οι υπηρεσίες διαδικτύου είναι «ένα σύστημα λογισμικού, που περιγράφεται από ένα URL, η διεπαφή του οποίου περιγράφεται με XML. Η περιγραφή του μπορεί να ανακαλυφθεί από άλλα συστήματα λογισμικού. Τέτοια συστήματα μπορεί να αλληλεπιδρούν με την υπηρεσία Διαδικτύου με ένα τρόπο

που υποδεικνύεται από τον ορισμό της, χρησιμοποιώντας μηνύματα XML που μεταφέρονται με κάποιο πρωτόκολλο του Διαδικτύου». Από την άλλη ο Mark Colan της IBM, υποστηρίζει πως οι υπηρεσίες Διαδικτύου είναι «βασισμένες στο Διαδίκτυο εφαρμογές που εκτελούν έναν συγκεκριμένο επιχειρησιακό στόχο και προσαρμόζονται σε ένα συγκεκριμένο τεχνικό σχήμα». Έτσι, εάν ορισμένες διαδικασίες από τις εφαρμογές μπορούν να επικαλεσθούν μέσω του Διαδικτύου, μέσα σε μια μέθοδο και σε ένα τυποποιημένο σχήμα, έχουμε έναν εξυπηρετητή υπηρεσιών Διαδικτύου. Ομοίως, εάν καλέσουμε ορισμένες διαδικασίες εξωτερικές, ως προς στις εφαρμογές μας που εκτελούνται στο τοπικό μας υπολογιστή, μέσω Διαδικτύου, έχουμε τον πελάτη των υπηρεσιών Διαδικτύου. Ο καθένας καταλαβαίνει ότι για αυτού του είδους της δικτύωσης, χρειάζονται τυποποιημένα σχήματα και μέθοδοι. Όλα αυτά υπάρχουν και είναι διαθέσιμα, για μερικά χρόνια τώρα.

Το SOAP (Simple Object Access Protocol) είναι ένα πρωτόκολλο κλήσης απομακρυσμένης διαδικασίας (Remote Procedure Call - RPC [ISO11578]) που χρησιμοποιεί τα τυποποιημένα πρωτόκολλα Διαδικτύου για τη μεταφορά - είτε HTTP [RFC2616] για τις σύγχρονες κλήσεις είτε SMTP [RFC821] για τις ασύγχρονες κλήσεις. Το SOAP χρησιμοποιεί την XML για να περιγράψει το σχήμα των στοιχείων θα διαβιβασθούν κάθε φορά. SOAP σημαίνει πρωτόκολλο πρόσβασης απλού αντικειμένου, αν και μερικοί το αναφέρουν και ως "προσανατολισμένο προς τις υπηρεσίες πρωτόκολλο αρχιτεκτονικής" (Service Oriented Architecture Protocol) [COL].

Φυσικά, αυτό είναι όλο πολύ καλό, αλλά το SOAP είναι ακόμα ακριβώς ένα RPC, που καλεί τις χαμηλού επιπέδου λειτουργίες και που αφήνει λίγο πολύ όλα στους υπεύθυνους για την ανάπτυξη λογισμικών συστημάτων. Προκειμένου να γίνει ευκολότερη η χρησιμοποίηση, αναπτύχθηκε ένα σχήμα για την περιγραφή των υπηρεσιών που μπορούν να επικαλεσθούν από το SOAP, η γλώσσα περιγραφής υπηρεσιών διαδικτύου WSDL (Web Service Description Language).

Η WSDL μπορεί να θεωρηθεί ως συμπλήρωμα στο SOAP, δεδομένου ότι διευκολύνει τη διαλειτουργικότητα μεταξύ των υπηρεσιών Διαδικτύου. Όπως η γλώσσα καθορισμού διεπαφών (Interface Definition Language – IDL), που ενεργεί ως περιγραφική υπηρεσία με την αρχιτεκτονική μεσάζοντα ζήτησης συνηθισμένου αντικειμένου (Common Object Request Broker Architecture – CORBA) [OMG96], η γλώσσα περιγραφής υπηρεσιών διαδικτύου (WSDL) είναι μια XML σύνταξη για να περιγράψει τις υπηρεσίες Διαδικτύου. Οι προδιαγραφές για την WSDL προέρχονται από μια κοινή πρωτοβουλία από τη Microsoft, την IBM και Ariba. Όλο και περισσότερες εφαρμογές SOAP υποστηρίζουν αυτήν την γλώσσα περιγραφής. Με την WSDL, οι εφαρμογές που χρησιμοποιούν το SOAP μπορούν να διαμορφώσουν από μόνες τους τις ανταλλαγές μεταξύ των υπηρεσιών Διαδικτύου, κρύβοντας τις περισσότερες από τις χαμηλού επιπέδου τεχνικές λεπτομέρειες.

Το SOAP δεν στηρίζεται αποκλειστικά στα καθιερωμένα πρότυπα, αλλά και αναγνωρίζεται ευρέως από τους σημαντικότερους μάλιστα φορείς της αγοράς όπως η Microsoft, η IBM, η Sun, και ακόμη και η Oracle. Ακόμη και η πρωτοβουλία της Microsoft στο .NET έχει το SOAP στον πυρήνα της [URL-10], αν και το πρωτόκολλο έχει παρθεί ως μελέτη του W3C κι έχει πληθώρα υλοποιήσεων σε ανοικτό πηγαίου κώδικα.

Μέσα σε αυτό το πλαίσιο, το SOAP και η WSDL μπορούν να θεωρηθούν ως κύρια κομμάτια στο γρίφο, αλλά δεν διαμορφώνουν το ολόκληρο σχήμα από μόνα τους. Όπως συνέβη για τις δικτυακές πύλες που στόχευαν και τελικά δέχθηκαν πληθώρα από επισκέπτες, οι υπηρεσίες Διαδικτύου αφιερώνονται στις επιχειρήσεις και τίθεται ως στόχος να αντεπεξέλθουν σε μια τεράστια ανάπτυξη, εάν λάβουμε υπόψη μας τις προβλέψεις [PRE-1,2]. Οι επαγγελματίες της βιομηχανίας, που έρχονται αντιμέτωποι με μια εξαιρετικά πλούσια προσφορά στην οποία είναι δύσκολο να προσδιοριστεί οτιδήποτε, θα πρέπει έπειτα να αντιμετωπίσουν το αιώνιο θέμα «που και πώς κάποιος μπορεί να βρει τις πληροφορίες που αναζητά». Οι ανάγκες της αγοράς είναι μια απλή υπηρεσία που επιτρέπει στους φορείς:

- Να βρουν το σωστό συνεργάτη (αυτός που προσφέρει στην υπηρεσία που χρειαζόμαστε)
- Να λάβουν τις απαραίτητες πληροφορίες ώστε να εμπλακούν σε ηλεκτρονικές συναλλαγές με το συνεργάτη (π.χ. να έχουν πρόσβαση σε ένα έγγραφο WSDL).

Εδώ είναι όπου εμπλέκεται η καθολική περιγραφή, ανακάλυψη, και ολοκλήρωση (UDDI). Η UDDI ήταν το αποτέλεσμα σκέψης αρκετών από τους γίγαντες της βιομηχανίας τεχνολογίας πληροφορικής (η IBM, η Sun, η Microsoft και Ariba, οι τρεις ίδιες που ήταν πίσω από WSDL και το SOAP και αυτό δεν είναι καμία σύμπτωση). Το UDDI είναι στην ουσία η προσπάθεια καταγραφής των υπηρεσιών που προσφέρονται στον Διαδίκτυο σε μια απέραντη παγκόσμια XML βάση δεδομένων, ένας κατάλογος πληροφοριών ως προς της υπηρεσίες [URL-19]. Πολλές επιχειρήσεις υποσχέθηκαν να υποστηρίξουν τη UDDI. Το πρόγραμμα έχει κατορθώσει ήδη να συλλέξει μαζί επιχειρήσεις όπως SAP, Compaq, Dell, ICG, Rational, Accenture, Cargill, Clarus, CommerceOne, Sabre, Tibco, WebMethods. Όλοι οι σχετικοί φορείς το θεώρησαν σημαντικό βήμα προς τα εμπρός.

Η UDDI προσφέρει ένα τεχνικό πλαίσιο που είναι ανεξάρτητο από τις πλατφόρμες λογισμικού και ανοιχτού κώδικα, ώστε οι επιχειρήσεις να μπορούν να βρουν η μια την άλλη, να καθορίσουν πώς θα αλληλεπιδράσουν στο διαδίκτυο και να καθορίσουν το πώς οι πληροφορίες θα μοιραστούν χρησιμοποιώντας ένα παγκόσμιο σύστημα εγγραφής. Το αποτέλεσμα αυτού του προγράμματος, σύμφωνα με τους υποστηρικτές του, θα είναι ότι οι επιχειρήσεις θα είναι σε θέση να εισαγάγουν τον κόσμο του ηλεκτρονικού επιχειρείν πολύ πιο γρήγορα στην αγορά. Έτσι το τεχνικό σχήμα που μπορούμε να καταρτίσουμε για τις υπηρεσίες Διαδικτύου έχει ως εξής:

- XML: σχήμα για την ανταλλαγή στοιχείων και την περιγραφή δεδομένων
- SOAP: πρωτόκολλο για την κλήση των υπηρεσιών Διαδικτύου
- WSDL: σχήμα για την περιγραφή των υπηρεσιών Διαδικτύου
- UDDI: κεντρική οργάνωση για την εγγραφή, εύρεση και χρησιμοποίηση των υπηρεσιών Διαδικτύου

Και χάρη στο SOAP, έχουμε τελικά το δίαυλο που θα μας επιτρέψει να αναπτύξουμε σχέσεις πελάτη - εξυπηρετητή μεταξύ των εφαρμογών και μέσω Διαδικτύου με έναν ενιαίο τρόπο - και που είναι μια σημαντική καινοτομία! Οι προηγούμενες προσπάθειες απέτυχαν είτε επειδή δεν ήταν απλές, είτε επειδή

δεν δέχθηκαν ενιαία υποστήριξη. Όχι ότι οι αρχιτεκτονικές CORBA, DCOM ή ακόμα και RMI είναι “βαριές”, αργές και χρειάζονται ιδιαίτερη προσοχή κατά τη χρήση τους και για το λόγο αυτό απορρίφθηκαν από τους υπεύθυνους ανάπτυξης εφαρμογών Διαδικτύου. Είναι προ πάντων επειδή κανένας δεν μπόρεσε φυσικά να διασχίσει το εμπόδιο ζωνών ηλεκτρονικής ασφαλείας (firewall)[FWP]. Το SOAP δεν υφίσταται αυτήν την αναπηρία, επειδή είναι βασισμένο στο HTTP και αυτό του δίνει ένα μεγαλύτερο πλεονέκτημα από την άποψη της συμμόρφωσης με τα πραγματικά πρότυπα. Επιπλέον του γεγονότος ότι αποτελεί ανοικτό πρότυπο και δέχθηκε πολλαπλή υποστήριξη.

Έτσι στην ερώτηση για το ποια είναι η αληθινή φύση των υπηρεσιών διαδικτύου, η απάντηση είναι πως αποτελούν μια αρχιτεκτονική πελάτη – εξυπηρετητή ανάμεσα σε εφαρμογές και μέσω του διαδικτύου. Το επόμενο βήμα είναι να αναρωτηθεί κανείς για τον πραγματικό τους ρόλο. Τα εμπορικά τμήματα των φορέων της βιομηχανίας δίνουν έμφαση στους ακόλουθους στόχους:

- Στην ανταλλαγή εξατομικευμένων πληροφοριών μεταξύ συνεργατών (e-Business) [WSEB-1,2].
- Να δημοσιεύσουν και να προσφέρουν εφαρμογές σε ένα έτοιμο προς χρήση σχήμα.

Η αγορά είναι ανοιχτή για μια καινούργια κατηγορία εφαρμογών που θα προωθούν την ολοκλήρωση των ενδοεπιχειρησιακών υπηρεσιών σε ένα ενιαίο καθολικό σχήμα και θα κάνουν προσιτά επιχειρησιακά συστήματα πληροφοριών σε κλίμακα Διαδικτύου [DWS].

Πριν μια δεκαετία περίπου είδαμε τη ραγδαία ανάπτυξη που έλαβε η δημιουργία δικτυακών τόπων. Σήμερα γινόμαστε μάρτυρες της ανάπτυξης υπηρεσιών διαδικτύου.

1.2 Υπηρεσίες διαδικτύου, μια αποκεντρωμένη προσέγγιση

Το γενικό σχήμα όπως αναφέρεται παραπάνω είναι αρκετά καλό. Έχουμε την XML για να περιγράψουμε χρήσιμη πληροφορία, την γλώσσα WSDL να περιγράψουμε τη διεπαφή με την υπηρεσία, το SOAP για να επικοινωνήσουμε με την υπηρεσία και τέλος τον «χρυσό οδηγό» UDDI στο οποίο καταγράφονται οι υπηρεσίες διαδικτύου. Άρα το θέμα της χρησιμοποίησης μιας τέτοιας υπηρεσίας είναι απλή υπόθεση. Αρκεί να μεταβούμε, με κάποιο τρόπο, στο δικτυακό τόπο που προσφέρεται αυτή η υπηρεσία του «χρυσού οδηγού» και να αναζητήσουμε για υπηρεσίες που επιθυμούμε. Φυσικά εδώ προϋποθέτουμε ότι γνωρίζουμε τον δικτυακό αυτό τόπο και τον τρόπο χρησιμοποίησης της UDDI. Αν δεν τον γνωρίζουμε, τότε θα πρέπει να την αναζητήσουμε και αυτή την υπηρεσία για να κάνουμε έπειτα αναζήτηση για την υπηρεσία που μας ενδιαφέρει.

Θα αναρωτηθεί κανείς, γιατί να αναζητήσουμε μια επιπλέον υπηρεσία και να μην αναζητήσουμε κατευθείαν την υπηρεσία που μας ενδιαφέρει; Επίσης τι συμβαίνει όταν για κάποιους λόγους τεθεί εκτός αυτή η υπηρεσία καταλόγου, απορρίπτονται όσοι την χρησιμοποιούν; Επιπλέον, πόσο έγκαιρη και έγκυρη πληροφορία μπορεί να υπάρχει σε ένα κατάλογο ακόμη και αν αυτός ενημερώνεται περιοδικά και μάλιστα αρκετά συχνά, δεδομένου ότι οι επιχειρήσεις ενημερώνουν συνεχώς τις υπηρεσίες τους, τις αναβαθμίζουν, τις

θέτουν διαθέσιμες, τις θέτουν εκτός λειτουργίας και φέρνουν νέες, κατά βούληση. Βέβαια το θέμα της αποθήκευσης και της επεξεργασίας δεν αποτελούν εμπόδιο αφού το κόστος για αυτά έχει μειωθεί σημαντικά τα τελευταία χρόνια. Αν και υπήρχε σαν ιδέα η χρήση ομότιμων οντοτήτων για τη δημιουργία ενός σχήματος για την αναζήτηση υπηρεσιών διαδικτύου, τελικά η UDDI είναι η λύση που επικράτησε. Αυτό γιατί μέχρι πρόσφατα δεν είχε γίνει κάτι σημαντικό στην τεχνολογία ομότιμων οντοτήτων. Οι πρώτες εφαρμογές που έφεραν τον όρο στο προσκήνιο ήταν κάποιες επεκτάσεις των παραδοσιακών τεχνικών πελάτης εξυπηρετητής. Η τεχνολογία όμως αναπτύσσεται σταθερά και αναμένεται να φέρει επανάσταση στη διακίνηση των πληροφοριών.

Οι πρώτοι υπεύθυνοι για την ανάπτυξη του ARPANET και του διαδικτύου επέτρεψαν στον εαυτό τους να ονειρευτούν τη ημέρα που οι υπολογιστές σε όλο τον κόσμο όταν θα συνδέονταν μοιράζοντας τους πόρους τους μεταξύ τους. Το 1962, ένας από εκείνους, J.C.R. Licklider του MIT, έγραψε μια διάσημη σειρά υπομνημάτων στην οποία περιέγραψε ένα «διαγαλαξιακό δίκτυο» των διασυνδεόμενων υπολογιστών. Το όραμά του οδήγησε στη δημιουργία του προγράμματος ελέγχου δικτύων (network control program NCP), του πρώτου πρωτοκόλλου δικτύωσης host-host και του προδρόμου στο TCP / IP. Η έννοια “host - host” που καλούμε τώρα peer-to-peer ήταν κρίσιμη στην ανάπτυξη του διαδικτύου. Κάθε υπολογιστής στο δίκτυο ήταν ίσος. Κάθε υπολογιστής θα μπορούσε να έχει πρόσβαση στους πόρους οποιουδήποτε άλλου υπολογιστή στο δίκτυο ενώ ταυτόχρονα θα καθιστούσε τους δικούς του πόρους διαθέσιμους. Η επικοινωνία μεταξύ των host ήταν επίσης ίση. Κανένας υπολογιστής δεν θα θεωρούνταν ως πελάτης ή συστατικό κάποιου άλλου υπολογιστή, και όλοι οι υπολογιστές θα μοιράζονταν πάνω κάτω ίσο εύρους ζώνης.

Διάφορα γεγονότα έχουν συνωμοτήσει να αλλάξουν το τοπίο Διαδικτύου από πρώτιστα peer-to-peer στη τώρα πιο γνωστή αρχιτεκτονική πελάτης - εξυπηρετητής. Το διαδίκτυο έχει γίνει βαθμιαία εμπορικότερο, και οι εταιρίες χτίζουν τοίχους ηλεκτρονικής προστασίας (firewalls) γύρω από τις πληροφορίες τους για να ελέγξουν την πρόσβαση. Τα εκατομμύρια των ανθρώπων «εισέρχονται» στο Διαδίκτυο χρησιμοποιώντας τους υπολογιστές γραφείου που δεν μπορούν να αναμετρηθούν με τη δύναμη των εξυπηρετητών που διαμορφώνουν τη ραχοκοκαλιά του διαδικτύου. Και πολλές δημοφιλείς εφαρμογές και υπηρεσίες διαδικτύου, συμπεριλαμβανομένου του World Wide Web και του FTP, είναι βασισμένες σε μια αρχιτεκτονική πελατών εξυπηρετητών.

Στα τελευταία έτη, εντούτοις, η τεχνολογία ομότιμων οντοτήτων (peer-to-peer ή P2P) άλλη μια φορά έρχεται για να ασκήσει μια βαθιά επίδραση στο διαδίκτυο και τη διανομή των πληροφοριών και των πόρων. Η τεχνολογία P2P εισήλθε επιθετικά στο προσκήνιο, αλλά υπάρχει μια ευρεία ποικιλία των απόψεων ως προς το τι είναι ακριβώς P2P.

Ο Clay Shirky (σύμβουλος διαδικτύου, συγγραφέας, και ομιλητής) είπε ότι «P2P είναι μια κατηγορία εφαρμογών που εκμεταλλεύεται την αποθήκευση των πόρων, τους κύκλους του επεξεργαστή, ανθρώπινη παρουσία-χειριστή που είναι διαθέσιμα στις άκρες του Διαδικτύου.»

Ο Li Gong της Sun Microsystems [LIGON-1,2] έγραψε «ο όρος δικτύωση P2P εφαρμόζεται σε ένα ευρύ φάσμα των τεχνολογιών που αυξάνουν πολύ τη χρησιμοποίηση των πληροφοριών, του εύρους ζώνης, και των πόρων

υπολογισμού στο Διαδίκτυο. Συχνά, αυτές οι P2P τεχνολογίες υιοθετούν ένα βασισμένο στο δίκτυο ύψος υπολογισμού που ούτε αποκλείει ούτε εξαρτάται εγγενώς από τα συγκεντρωμένα σημεία ελέγχου.»

Οι Ed Dumbill της XML.com λέει, «P2P είναι ο χρήστης.»

Για να καταλάβουμε όμως τη διαφορά μεταξύ του τρόπου που γίνονται τώρα τα πράγματα και πώς η τεχνολογία P2P θα μπορούσε να παρέχει τις πιο χρήσιμες και γερές λύσεις, ας εξετάσουμε το ακόλουθο παράδειγμα. Συνήθως για να βρούμε κάποιες συγκεκριμένες πληροφορίες στο διαδίκτυο, πηγαίνουμε σε μια μηχανή αναζήτησης, πχ το Google που είναι και η πιο διαδεδομένη, και υποβάλλουμε μια ερώτηση αναζήτησης. Τις περισσότερες φορές, λαμβάνουμε έναν κατάλογο αρκετές χιλιάδες αποτελεσμάτων, πολλά από τα οποία είναι ανεξάρτητα, είναι ξεπερασμένα, ή χειρότερα ακόμα, δείχνουν σε πόρους που δεν υπάρχουν πλέον. Είναι πραγματικά στενάχωρο πως η πιο ευφυής μηχανή αναζήτησης σήμερα στο Διαδίκτυο μπορεί να έχει τέτοια αποτελέσματα. Ένα από τα προβλήματα όμως με την τρέχουσα λύση μηχανών αναζήτησης βρίσκεται στη συγκέντρωση της γνώσης και των πόρων. Το Google στηρίζεται σε μια κεντρική βάση δεδομένων που ενημερώνεται καθημερινά «χτενίζοντας» το διαδίκτυο για τις νέες πληροφορίες. Λόγω του αριθμού καταχωρήσεων των ιστοσελίδων στη βάση δεδομένων της (περισσότερες από 2 δισεκατομμύρια), δεν ενημερώνεται κάθε καταχώρηση κάθε φορά. Ως αποτέλεσμα αυτής της ανεπάρκειας, οι πληροφορίες στη βάση δεδομένων Google να μην απεικονίζουν τις πιο έγκυρες διαθέσιμες πληροφορίες κάθε φορά, μικραίνοντας κατά συνέπεια τη χρησιμότητα των αποτελεσμάτων της για οποιαδήποτε δεδομένη ερώτηση αναζήτησης.

Η τεχνολογία μηχανών αναζήτησης έχει κι άλλα διάφορα μειονεκτήματα, όπως:

- Απαιτεί πολύ εξοπλισμό. Το Google, παραδείγματος χάριν, τρέχει μια συστάδα Linux, 10.000 μηχανών για να παρέχει την υπηρεσία του.
- Αν η μηχανή αναζήτησης βγει εκτός (π.χ. λόγω μιας διακοπής, ή αναβάθμισης λειτουργίας των μηχανών αυτών), οι πληροφορίες όλης της μηχανής αναζήτησης είναι μη διαθέσιμες στους χρήστες της.
- Λόγω του μεγέθους του διαδικτύου, η μηχανή αναζήτησης δεν μπορεί να παρέχει έναν περιεκτικό δείκτη σε κάθε ιστοσελίδα του διαδικτύου.
- Οι μηχανές αναζήτησης δεν μπορούν να συνδεθούν με τις πληροφορίες που αποθηκεύονται στη βάση δεδομένων ενός εταιρικού ιστοχώρου, που σημαίνει ότι η μηχανή αναζήτησης δεν μπορεί να “δει” κάποιες πληροφορίες.

Μια παρόμοια υπηρεσία αναζήτησης θα μπορούσε να εφαρμοστεί χρησιμοποιώντας την P2P τεχνολογία, που αυξάνει την αποδοτικότητα με τις πρόσθετες επιθυμητές ιδιότητες [LIGON-1]. Φανταστείτε λοιπόν εάν ο καθένας θα μπορούσε να τρέξει έναν προσωπικό εξυπηρετητή ιστοσελίδων σε έναν υπολογιστή γραφείου! Υποθέστε ότι, εκτός από την εξυπηρέτηση του περιεχομένου από τη μηχανή του χρήστη, αυτός ο υπολογιστής είχε την ικανότητα να επεξεργαστεί ερωτήσεις και να δίνει απαντήσεις για τα έγγραφα που χειρίζονται από τον ίδιο. Ο εξυπηρετητής αυτός, ενός χρήστη θα μπορούσε να λάβει μια ερώτηση από μια ισότιμη οντότητα, να ελέγξει τα έγγραφα που διαχειρίζεται για μια αντιστοιχία, και αποκρίνεται στην ερώτηση με έναν κατάλογο ταιριάσματος των εγγράφων. Ο εξυπηρετητής του χρήστη θα ήταν

αρμόδιος για την ευρετηρίαση των εγγράφων που κατέστησε ο ίδιος διαθέσιμα και επομένως θα ήταν σε θέση τις πιο εξακριβωμένες, ενημερωμένες πληροφορίες για τα έγγραφα του χρήστη σε καθένα που υποβάλλει μια ερώτηση αναζήτησης. Η εργασία αυτή της δεικτοδότησης στα έγγραφα ενός χρήστη θα ήταν πιο εύκολη να αναπτυχθεί, να συντηρηθεί, να βελτιωθεί και να ελεγχθεί από το αυτό που έχει να αντιμετωπίζει το Google, μερικές ιστοσελίδες ενός χρήστη σε σχέση με τα δισεκατομμύρια των σελίδων εκατομμυρίων χρηστών. Οι εταιρίες από την άλλη θα μπορούσαν να παρέχουν διόδους να συνδέσουν τις βάσεις δεδομένων των ιστοχώρων τους με το P2P δίκτυο, για να παρέχουν την πρόσβαση στις πληροφορίες που οι μηχανές αναζήτησης δεν μπορούν να φθάσουν.

Το σύστημα αυτό θα είχε ένα επιπλέον πλεονέκτημα, εάν ο εξυπηρετητής του χρήστη αποσυνδέονταν από το δίκτυο, η υπηρεσία αναζήτησης θα γινόταν επίσης μη διαθέσιμη οι άλλοι χρήστες που ψάχνουν στο δίκτυο δεν θα λάμβαναν τα αποτελέσματα για τους πόρους που ήταν μη διαθέσιμοι. Έτσι όταν κάποιος θα ψάχνει για τις πληροφορίες, θα είναι σχεδόν σίγουρος ότι οποιοδήποτε αποτέλεσμα έβρισκε, αυτό θα ήταν διαθέσιμο εκείνη τη στιγμή, μειώνοντας το σπαταλημένο χρόνο αναζήτησης.

Μια τέτοια μηχανή αναζήτησης αναπτύσσεται χρησιμοποιώντας την P2P τεχνολογία πάνω στην πλατφόρμα JXTA της Sun Microsystems. Περισσότερες πληροφορίες υπάρχουν στην υπόλοιπη διπλωματική.

Προχωρώντας ακόμη παραπέρα την ιδέα αυτή, φανταστείτε κάποιος να είναι σε θέση να προσφέρει δυναμικά πληροφορίες για τις υπηρεσίες που ο ίδιος προσφέρει μέσω του διαδικτύου. Μια επιχείρηση ή κάποιος πάροχος υπηρεσιών θα μπορούσε να προσφέρει όπως παραπάνω πληροφορίες για τις υπηρεσίες του σε όποιον αναζητά υπηρεσίες διαδικτύου ή να καθιστά αυτές τις πληροφορίες διαθέσιμες σε όποιον τις αναζητά. Κάθε υποψήφιος μελλοντικός χρήστης να είναι σε θέση να τις ανακαλύπτει και να τις χρησιμοποιεί δυναμικά για να καλεί την υπηρεσία που επιθυμεί. Η ιδέα αυτή είναι λίγο πιο πολύπλοκη από την προηγούμενη λόγω της «ρευστής» φύσης που έχει η πληροφορία, σε αντιπαράθεση με τη στατική δομή των εγγράφων ιστοσελίδων. Η πληροφορία για την ύπαρξη υπηρεσιών διαδικτύου αναζητείται σε ένα δίκτυο ομότιμων οντοτήτων ενώ η πρόσβαση στην υπηρεσία γίνεται έξω από το δίκτυο αυτό.

Όπως την επανάσταση επιτραπέζιου υπολογιστή στα μέσα της δεκαετίας του '80, η τεχνολογία P2P υπόσχεται να επιφέρει επανάσταση στην ανταλλαγή των πληροφοριών.

Αυτή η διπλωματική είναι οργανωμένη σε τρία μέρη. Στο πρώτο μέρος αναλύονται οι σχετικές με τις υπηρεσίες διαδικτύου έννοιες. Στο δεύτερο μέρος ξεκαθαρίζονται οι όροι που εμπλέκονται στην τεχνολογία P2P και παρουσιάζεται η πλατφόρμα JXTA της Sun. Τέλος στο τρίτο μέρος συνδυάζουμε τα παραπάνω και προβάλλουμε έναν καινούριο τρόπο όσον αφορά τη αναζήτηση και χρησιμοποίηση υπηρεσιών διαδικτύου. Συγκεκριμένα προτείνουμε μια πλατφόρμα στην οποία κάθε πάροχος υπηρεσιών μπορεί να ανακοινώσει της υπηρεσίες που διαθέτει και ο κάθε υποψήφιος χρήστης αναζητά υπηρεσίες. Επιπλέον ο πελάτης, δεν είναι αναγκασμένος να γνωρίζει το πώς πρέπει να κληθεί η υπηρεσία, το σύστημα από μόνο του δημιουργεί αυτόματα τον τρόπο κλήσής της.

ΜΕΡΟΣ Ι

ΣΥΝΘΕΣΗ ΤΩΝ ΥΠΗΡΕΣΙΩΝ
ΔΙΑΔΙΚΤΥΟΥ

1. Εισαγωγή

Λόγω της πολυπλοκότητας του σημερινού τρόπου ζωής χρειάζεται ένα είδος αυτοματοποίησης, χωρίς την διαρκή παρέμβαση του ανθρώπου, για την ανάκτησης της πληροφορίας. Έτσι το μέλλον του Διαδικτύου βασίζεται στην εγκαθίδρυση ενός πιο αποδοτικού μέσου ανταλλαγής δεδομένων από τις HTML σελίδες που χρησιμοποιούνται ως σήμερα. Αυτό μπορεί να γίνει εφικτό, σε μεγάλο βαθμό, με τις υπηρεσίες διαδικτύου.

Η άφιξη των υπηρεσιών διαδικτύου φέρνει ενδιαφέρουσες αλλαγές στις τρέχουσες τεχνολογίες Διαδικτύου. Εξυπηρετητές εφαρμογών όπως BEA WebLogic Server [URL-4] ή IBM WebSphere [URL-5] ήδη υποστηρίζουν αυτού του νέου είδους εφαρμογές. Εν τω μεταξύ προμηθευτές όπως η Iona [URL-6], Web Methods [URL-7], CapeClear [URL-8] και Idoox [URL-9] ήδη έχουν διανύσει πολύ δρόμο. Τα εργαλεία ανάπτυξης που δημιουργούνται είναι προσανατολισμένα προς τις υπηρεσίες διαδικτύου με τρανταχτό παράδειγμα την πλατφόρμα .NET της Microsoft [URL-10].

Οι τρόποι με τους οποίους οι εφαρμογές διαδικτύου αναπτύσσονται θα επηρεαστούν από τις νέες δυνατότητες των υπηρεσιών Διαδικτύου. Όσον αφορά τη δημιουργία αυτών των εφαρμογών, υψηλού επιπέδου προσεγγίσεις έρχονται στο προσκήνιο. Μέχρι τώρα η προσέγγιση ενός επιχειρησιακού συστατικού αντικειμένου υιοθετήθηκε και σήμερα προωθείται ευρύτερα. Η προδιαγραφή EJB έχει συμβάλει πολύ στο να βελτιώσει την πληροφόρηση των εννοιών όπως η αποτύπωση οντοτήτων – συσχετίσεων.

Για επιπλέον πληροφορίες σε θέματα σχετικά με υπηρεσίες διαδικτύου κάποιος μπορεί να συμβουλευτεί την ιστοσελίδα της IBM [URL-11], τα θέματα της TrendMarkers [URL-12]. Για τα τελευταία νέα στο αντικείμενο αυτό, ο Webservices.org [URL-13] είναι καλή πηγή πληροφόρησης.

2. Η εκτατή γλώσσα σήμανσης (EXtensible Markup Language – XML)

Η γλώσσα XML [URL-14] είναι όπως η γλώσσα σήμανσης υπερκειμένου (HTML). Σε αντίθεση από την HTML όμως, η οποία συγκρατά πληροφορία για την εμφάνιση του περιεχομένου κειμένου, η XML περιέχει μόνο δομημένη πληροφορία. Η XML προέρχεται από την SGML η οποία, πρώτη, ξεχώρισε το περιεχόμενο από την παρουσίαση της πληροφορίας.

Το μεγάλο πλεονέκτημα της XML είναι ότι διαχωρίζει τα δεδομένα από τον τρόπο εμφάνισης και τον χειρισμό τους και επιπλέον οργανώνει τα δεδομένα αυτά ως συστατικά της πληροφορίας που αναπαριστούν. Για το λόγο αυτό η XML χρησιμοποιείται:

- Στην ανταλλαγή πληροφορίας, μέσω εγγράφων, μεταξύ εφαρμογών
- “write one, view anywhere applications” εφαρμογές που μπορούν να προσεγγιστούν από οπουδήποτε
- Τον ορισμό νέων πιο ειδικευμένων γλωσσών σήμανσης, όπως είναι οι WML, NewsML, OWL [URL-14] κτλ.

Ας θεωρήσουμε την περίπτωση του βιογραφικού σημειώματος ενός ατόμου. Αυτό περιλαμβάνει κάποια προσωπικά στοιχεία, την εκπαίδευσή του, την εργασιακή εμπειρία και τα ενδιαφέροντά του. Ένα έγγραφο XML που θα μπορούσε να αναπαριστά την πληροφορία του βιογραφικού σημειώματός αυτού φαίνεται στο παρακάτω σχήμα.

```
<?xml version="1.0" standalone="no"?>
<cv >
  <personal_info>
    <name>
      <firstname>Julian</firstname>
      <lastname>Notos</lastname>
    </name>
    <age>24</age>
    <address type="home">
      <street_address>181 Asklepiou St.</street_address>
      <city>Athens</city>
      <state>Greece</state>
      <zip>11471 </zip>
    </address>
  </personal_info>
  <education>
    <university> NTUA </university>
    <seminars>
      <seminar>
        <theme>WAI </theme>
        <descr/>
      </seminar>
    </seminars>
  </education>
</cv>
```

Σχήμα 2.1 Περίπτωση βιογραφικού σημειώματος

Το XML αυτό έγγραφο περιέχει απλώς και μόνο την πληροφορία που εισάγουμε σε ένα βιογραφικό σημείωμα, χωρίς καμία υπόδειξη για το πώς το τελικό αυτό έγγραφο θα παρουσιαστεί στον ενδιαφερόμενο.

Παρατηρώντας και το παραπάνω έγγραφο πρέπει να επισημάνουμε τα βασικά χαρακτηριστικά.

- Ότι βρίσκεται ανάμεσα στα σύμβολα ‘<’ και ‘>’ (π.χ. το <cv>, το <age> κτλ) ορίζεται ως μια ετικέτα (tag). Κάθε ετικέτα μπορεί να έχει έναν πεπερασμένο αριθμό από γνωρίσματα (attributes), π.χ. η <?xml?> έχει ως γνώρισμα το “version” με τιμή “1.0”, ενώ η ετικέτα <address> έχει ως γνώρισμα το “type” με τιμή “home”. Μια ετικέτα όπως η <?xml?> στην κορυφή του εγγράφου λέγεται ετικέτα επεξεργασίας και περιλαμβάνει οδηγίες επεξεργασίας για το πρόγραμμα που θα κάνει ανάγνωση στο έγγραφο αυτό. Εδώ η ετικέτα αυτή αναφέρει πως το έγγραφο αυτό είναι XML έγγραφο, γραμμένο σύμφωνα με την έκδοση 1.0 της γλώσσας XML.
- Όταν η ετικέτα βρίσκεται χρησιμοποιεί μόνο τα σύμβολα ‘<’ και ‘>’ λέγεται ετικέτα αρχής (π.χ. <cv> , <personal_info>, <name> κτλ) ενώ όταν χρησιμοποιεί τα σύμβολα ‘</’ και ‘>’ λέγεται ετικέτα τέλους, π.χ </cv>, </personal_info>, </name> κτλ.

- Μια ετικέτα αρχής και η αντίστοιχη ετικέτα τέλους ορίζουν ένα στοιχείο (element) όταν ανάμεσά τους υπάρχουν απλά δεδομένα (π.χ. το `<age>...</age>`, `<firstname>...</firstname>`), ενώ όταν η ετικέτα αρχής και η αντίστοιχη ετικέτα τέλους περιλαμβάνουν άλλα στοιχεία (π.χ. τα `<name>... </name>`, `<education> ... </education>`) ή ομάδες τους (π.χ. το `<cv>...</cv>`) τότε έχουμε έναν κόμβο (node). Ό,τι βρίσκεται ανάμεσα σε μια ετικέτα αρχής και τέλους αναφέρεται στον κόμβο ή στο στοιχείο αυτό. Δημιουργείται λοιπόν μια δενδρική δομή στην οποία κάθε κόμβος που περιέχει κι άλλους κόμβους ή στοιχεία λέγεται πατέρας κόμβος, ενώ τα στοιχεία οι κόμβοι που περιέχονται λέγονται παιδιά. Έτσι κάθε κόμβος μπορεί να έχει μόνο έναν πατέρα αλλά πολλά παιδιά.

Οι κανόνες σύνταξης της γλώσσας καθορίζουν ένα βασικό τρόπο ώστε οι αναγνώστες να μπορούν να καταλάβουν τη δομή του εγγράφου. Οι κανόνες αυτοί αναφέρουν πως:

- Κάθε XML έγγραφο θα πρέπει να ξεκινά με την ετικέτα επεξεργασίας που αναφέρει την έκδοση της XML (version="1.0")
- Μια ετικέτα που ανοίγει πρέπει και να κλείνει. Επιτρέπεται το φώλιασμα των ετικετών αλλά όχι η μίξη τους, (π.χ. θεωρείται λάθος το `<name><firstName>Julian</name></firstName>` γιατί υπάρχει μίξη των ετικετών και όχι φώλιασμα).
- Μπορούμε να δημιουργήσουμε όποια ετικέτα επιθυμούμε. Όμως, οι ετικέτες ανοίγματος γράφονται με τον ίδιο τρόπο όπως οι ετικέτες κλεισίματος (π.χ. `<name>...</name>` και όχι `<Name>...</Name>`)
- Μπορούμε να σημειώσουμε όσα γνωρίσματα επιθυμούμε. Όμως τα γνωρίσματα αυτά πρέπει να έχουν κάποια τιμή και η τιμή πρέπει να περικλείεται ανάμεσα στα “ ” (π.χ. `<name genre= “male”>.. </name>` και όχι `<name genre=male>... </name>`).

2.1 Επικύρωση εγγράφων XML

Έτσι η XML μας επιτρέπει να δημιουργήσουμε οποιοδήποτε έγγραφο και να περιγράψουμε οποιαδήποτε πληροφορία. Φυσικά, σε ένα τέτοιο έγγραφο εμείς, που το γράφουμε, καθορίζουμε τη δομή του και μόνο εμείς είμαστε σε θέση να γνωρίζουμε εκ των προτέρων και να την επιβεβαιώσουμε. Σύγχυση, λοιπόν, θα δημιουργούταν αν ο καθένας περιέγραφε την πληροφορία σύμφωνα με τη δική του δομή, γιατί λαμβάνοντας ένα έγγραφο XML δεν μπορούμε να γνωρίζουμε τη δομή ή τους τύπους των στοιχείων.

Για να μπορούμε να ορίσουμε και να επικυρώσουμε την δομή ενός XML εγγράφου υπάρχουν βασικά δυο τεχνολογίες, με τον ορισμό τύπου εγγράφου (Document Type Definition – DTD) και με σχήμα XML (XML-Schema) [XSD2001-1, 2]. Πρόκειται για έγγραφα που καθορίζουν την αποδεκτή δομή των ετικετών έγγραφο XML. Το DTD είναι δανεισμένο από την SGML, το ίδιο το έγγραφο δεν είναι XML έγγραφο και προσπαθεί με αναδρομικό τρόπο να περιγράψει την εις βάθος δομή του XML που επικυρώνει. Αντίθετα το XML-Schema είναι ένα XML αρχείο, αναπτύχθηκε ειδικά για την επικύρωση των XML εγγράφων και αναμένεται να αντικαταστήσει το DTD. Επιπλέον το XML-

Schema περιγράφει και τους τύπους των στοιχείων, ορίζονται νέοι τύποι οι οποίοι μπορούν να επαναχρησιμοποιηθούν.

Παρακάτω δίνουμε τον ορισμό του βιογραφικού κατά DTD και με XML-Schema.

```
<!ELEMENT cv ( personal_info, education?, experience?, hobbies?)>

<!ELEMENT personal_info (name, age?, address*)>
<!ELEMENT name (firstname, lastname)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT address (street_address, city, state, zip)>
<!ATTLIST address type (home | work) "home" >
<!ELEMENT street_address (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT zip (#PCDATA)>

<!ELEMENT education (university?, seminars?)>
<!ELEMENT university (#PCDATA)>
<!ELEMENT seminars (seminar+)>
<!ELEMENT seminar (theme, descr?)>
<!ELEMENT theme (#PCDATA)>
<!ELEMENT descr (#PCDATA)>

<!ELEMENT experience (work+)>
<!ELEMENT work (name, descr?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT descr (#PCDATA)>

<!ELEMENT hobbies (hobby+)>
<!ELEMENT hobby (#PCDATA)>
```

Σχήμα 2.2 Επικύρωση με DTD

Για να δηλώσουμε πως επιθυμούμε την επικύρωση πρέπει να τροποποιήσουμε το XML αρχείο όπως παρακάτω:

```
<?xml version="1.0"?>
<!DOCTYPE cv SYSTEM "DTD_validation_file.dtd">
<cv> ... </cv>
```

Σχήμα 2.3 Τροποποίηση του αρχικού XML βιογραφικού για να δεχθεί επικύρωση με DTD

Δηλαδή, προσθέτουμε την εντολή για επικύρωση και σημειώνουμε το αρχείο που περιέχει το DTD κώδικα επικύρωσης.

```

<?xml version="1.0"?>
<Schema name="ValidationSchema" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:xml-datatypes">
  <AttributeType name="type" dt:type="enumeration" dt:values="home work"/>
  <ElementType name="firstname" content="textOnly"/>
  <ElementType name="lastname" content="textOnly"/>
  <ElementType name="age" content="textOnly"/>
  <ElementType name="street_address" content="textOnly"/>
  <ElementType name="city" content="textOnly"/>
  <ElementType name="state" content="textOnly"/>
  <ElementType name="zip" content="textOnly"/>
  <ElementType name="theme" content="textOnly"/>
  <ElementType name="descr" content="textOnly"/>
  <ElementType name="university" content="textOnly"/>
  <ElementType name="hobby" content="textOnly"/>

  <ElementType name="name" content="eltOnly" order="seq">
    <element type="firstname" minOccurs="1" maxOccurs="1"/>
    <element type="lastname" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="address" content="eltOnly" order="seq">
    <element type="street_address" minOccurs="1" maxOccurs="1"/>
    <element type="city" minOccurs="1" maxOccurs="1"/>
    <element type="state" minOccurs="1" maxOccurs="1"/>
    <element type="zip" minOccurs="1" maxOccurs="1"/>
    <attribute type="type" default="home"/>
  </ElementType>
  <ElementType name="personal_info" content="eltOnly" order="seq">
    <element type="name" minOccurs="1" maxOccurs="1"/>
    <element type="age" minOccurs="1" maxOccurs="1"/>
    <element type="address" minOccurs="1"/>
  </ElementType>
  <ElementType name="seminar" content="eltOnly" order="seq">
    <element type="theme" minOccurs="1" maxOccurs="1"/>
    <element type="descr" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="seminars" content="eltOnly">
    <element type="seminar" minOccurs="1"/>
  </ElementType>
  <ElementType name="education" content="eltOnly" order="seq">
    <element type="university" minOccurs="1" maxOccurs="1"/>
    <element type="seminars" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="work" content="eltOnly" order="seq">
    <element type="theme" minOccurs="1" maxOccurs="1"/>
    <element type="descr" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="experience" content="eltOnly">
    <element type="work" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="hobbies" content="eltOnly">
    <element type="hobby" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="cv" content="eltOnly" order="seq">
    <element type="personal_info" minOccurs="0" maxOccurs="1"/>
    <element type="education" minOccurs="1" maxOccurs="1"/>
    <element type="experience" minOccurs="1" maxOccurs="1"/>
  </ElementType>
</Schema>

```

Σχήμα 2.4 Επικύρωση με XML-Schema

Ενώ το XML βιογραφικό μας πρέπει να αλλάξει κατά το παρακάτω:

```
<?xml version="1.0" standalone="no"?>

<cv xmlns="x-schema: ValidationFile.xml">
...
</cv>
```

Σχήμα 2.5 Τροποποίηση αρχικού XML βιογραφικού

Αυτό που πρέπει να επισημάνουμε είναι ότι ενώ με το DTD η επικύρωση γίνεται από την ρίζα προχωρώντας κατά βάθος σε όλους τους κόμβους, με το XML-Schema γίνεται το αντίστροφο. Ορίζονται πρώτα τα φύλλα, τα απλά στοιχεία, και έπειτα οι κόμβοι πατέρες σαν σύνθεση αυτών, καθώς η διαδικασία προχωρά από τα φύλλα ως την ρίζα.

2.2 Ανάλυση XML

Γενικά υπάρχουν δυο φιλοσοφίες για την ανάλυση των XML αρχείων, η απλή προγραμματιστική διεπαφή εφαρμογών για επεξεργασία XML (Simple API for XML – SAX) και το μοντέλο αντικείμενου εγγράφου (Document Object Model – DOM)[DOM].

Σύμφωνα με την πρώτη φιλοσοφία τα στοιχεία ενός XML εγγράφου εξάγονται και επεξεργάζεται σειριακά, καθώς ο αναλυτής επισκέπτεται τις ετικέτες μια προς μια.

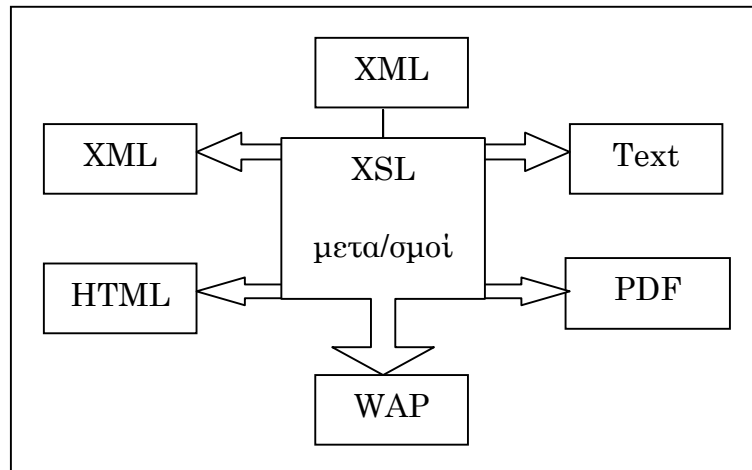
Στη δεύτερη αρχιτεκτονική δημιουργείται η δενδρική δομή που αναπαριστά ένα XML έγγραφο. Το έγγραφο ολόκληρο αναπαρίσταται ως ένα αντικείμενο. Βέβαια, γίνεται αντιληπτό πως, για να γίνει αυτό πρέπει όλες οι ετικέτες να προσπερασθούν τουλάχιστον μια φορά σειριακά για να αποτυπωθεί αυτή η δενδρική δομή.

2.3 Παρουσίαση - μετασχηματισμοί XML

Όπως έχουμε ήδη αναφέρει η XML δεν περιέχει καμία πληροφορία για την παρουσίαση της πληροφορίας. Για να παρουσιαστεί η πληροφορία των XML εγγράφων, κυρίως, σε σελίδες υπερκειμένου υπάρχουν δύο τρόποι: μέσω CSS (Cascading Style Sheets) ή μέσω XSL (EXtensible Stylesheet Language) [XSL].

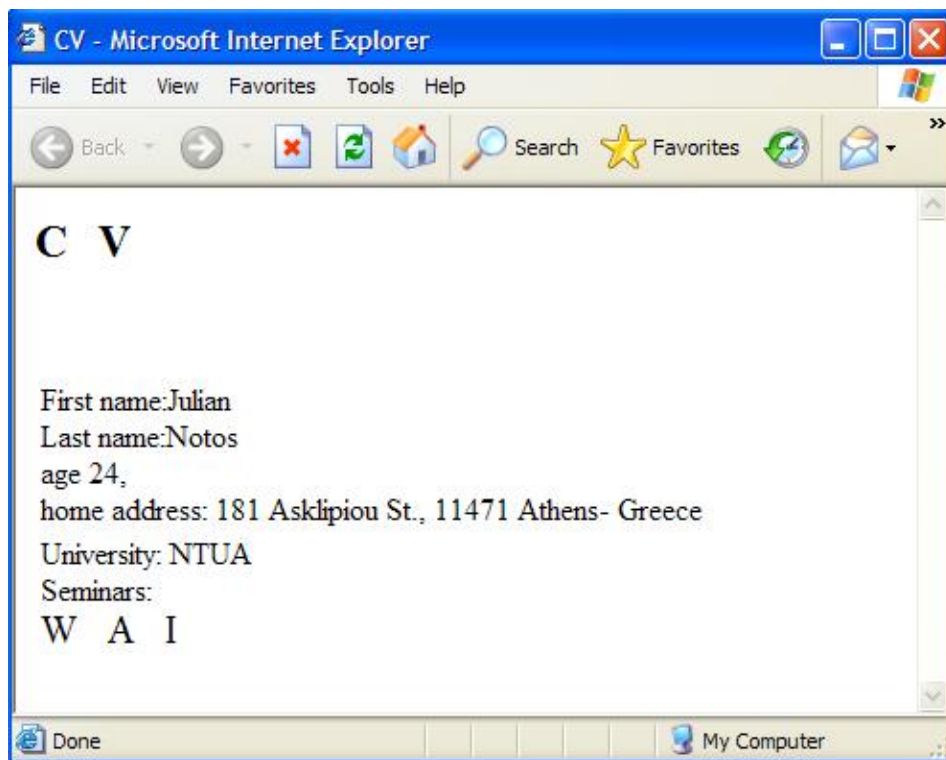
Χρησιμοποιώντας τα φύλλα στυλ, CSS, μπορούμε πολύ εύκολα να ορίσουμε τις ιδιότητες εμφάνισης ενός στοιχείου XML. Τα CSS όμως χρησιμοποιούνται μόνο σε φυλλομετρητές ιστού και δεν παρέχουν πολλές δυνατότητες.

Χρησιμοποιώντας την εκτατή γλώσσα στυλ, XSL, μπορούμε πολύ εύκολα να μετασχηματίσουμε ένα XML έγγραφο σε οποιαδήποτε άλλη γλώσσα σήμανσης. Και το XSL αποτελεί ένα XML έγγραφο κατά βάση και αποτελεί ένα δυνατό εργαλείο για την απεικόνιση XML εγγράφων. Το αρνητικό στοιχείο είναι ότι η γλώσσα αυτή υποστηρίζεται μέχρι στιγμής μόνο από τον Internet Explorer της Microsoft.



Σχήμα 2.6 XSL μετασχηματισμοί

Έτσι με το XSL κείμενο του σχήματος 2.8 μετασχηματίζουμε το αρχικό βιογραφικό σε HTML και παρουσιάζεται στον Internet Explorer όπως παρακάτω.



Σχήμα 2.7 Παρουσίαση του XML βιογραφικού ως HTML

Το XSL έγγραφο που εκτελεί το μετασχηματισμό του βιογραφικού σημειώματος σε HTML και παρουσιάζεται στο σχήμα 2.8.

Ενώ το αρχικό XML πρέπει να μετατραπεί προσθέτοντας μόνο την ετικέτα επεξεργασίας:

```
<?xml-stylesheet href="XSL-file " type="text/xsl"?>
```

στην κορυφή, πριν την εγγραφή των υπόλοιπων στοιχείων.

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<xsl:template>
<xsl:value-of/>
</xsl:template>
<xsl:template match="/">
<html><head><title>CV</title></head>
<body>
<h1 style="bgcolor:#FFFFFF; font-size:18pt; letter-spacing:1.0em"> CV </h1>
<table>
<xsl:for-each select="cv">
<tr><td>
<xsl:apply-templates select="personal_info"/></td></tr><br/>
<tr><td>
<xsl:apply-templates select="education"/></td></tr><br/>
<tr><td>
<xsl:apply-templates select="experience"/></td></tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
<xsl:template match="personal_info">
<xsl:apply-templates select="name"/>
age <xsl:value-of select="age"/>, <br/>
<xsl:apply-templates select="address"/>
</xsl:template>
<xsl:template match="education">
University: <xsl:value-of select="university"/> <br/>
Seminars: <xsl:apply-templates select="seminars"/>
</xsl:template>
<xsl:template match="address">
<xsl:value-of select="@type"/> address:
<xsl:value-of select="street_address"/>,
<xsl:value-of select="zip"/>
<xsl:value-of select="city"/>-
<xsl:value-of select="state"/>
</xsl:template>
<xsl:template match="name">
First name:<xsl:value-of select="firstname"/><br/>
Last name:<xsl:value-of select="lastname"/><br/>
</xsl:template>

<xsl:template match="seminars">
<div style="bgcolor:#FFFFFF; font-size:16pt; letter-spacing:1.0em">
<xsl:value-of select="seminar/theme"/>
<xsl:value-of select="seminar/descr"/></div>
</xsl:template>
</xsl:stylesheet>

```

Σχήμα 2.8 XSL που μετασχηματίζει το αρχικό XML

Έτσι με την XML έχουμε ένα δυνατό τρόπο να περιγράψουμε τα δεδομένα μας και έπειτα να τα μετασχηματίζουμε προγραμματιστικά κατά βούληση, χρησιμοποιώντας οποιεσδήποτε από τις τεχνικές ανάλυσης XML εγγράφων που αναφέρθηκαν ή με την χρησιμοποίηση της XSL να μετασχηματίσουμε σε άλλες μορφές εγγράφων.

3. Το πρωτόκολλο πρόσβασης απλού αντικειμένου (Simple Object Access Protocol – SOAP)

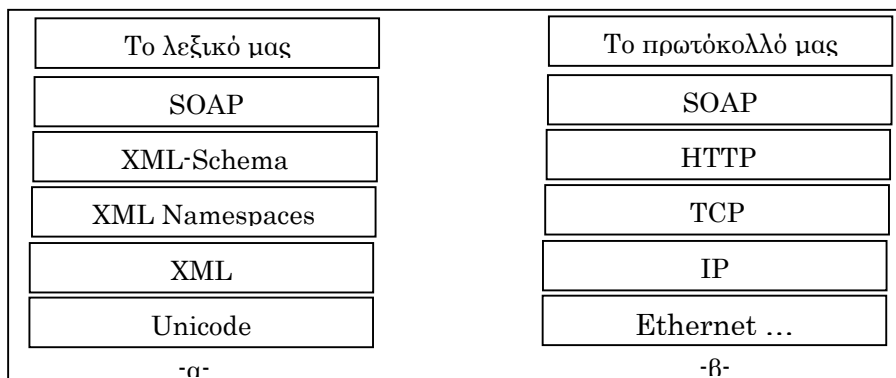
Το πρωτόκολλο SOAP [URL-15,10] αποτελεί ένα πλαίσιο ανταλλαγής μηνυμάτων βασισμένο σε XML. Είναι ειδικά σχεδιασμένο για την ανταλλαγή μηνυμάτων μέσω του διαδικτύου. Είναι απλό στη χρήση και εντελώς ανεξάρτητο από το λειτουργικό σύστημα, γλώσσα προγραμματισμού ή πλατφόρμα καταναμετημένων συστημάτων.

Εκτός από το να παρέχει μια αντιστοίχιση σε ένα επίπεδο μεταφοράς για την ανταλλαγή XML μηνυμάτων μέσω του διαδικτύου με το SOAP μια επιχείρηση μπορεί να δημοσιοποιήσει υπηρεσίες για ανταλλαγή εταιρικών XML δεδομένων και να καθορίσει τις ιδιότητες των ανταλλασόμενων μηνυμάτων που σχετίζονται με την ποιότητα υπηρεσίας.

Πρέπει να επισημάνουμε πως τα επιμέρους τμήματα των δραστηριοτήτων αυτών αφήνονται σε άλλες προδιαγραφές, π.χ. ο ορισμός των τύπων μπορεί να γίνει με XML-Schemas, η δημοσίευση – ανακάλυψη των υπηρεσιών γίνεται με το UDDI, η ασφάλεια υλοποιείται με SSL, https, κτλ. Μια εφαρμογή SOAP:

- είναι σε θέση να ανταλλάσσει πληροφορία με μηνύματα, σύμφωνα με το πρωτόκολλο SOAP,
- αναγνωρίζει τα μέρη του μηνύματος που πρέπει να εκτελεστούν από συγκεκριμένες εφαρμογές
- επεξεργάζεται τα υποχρεωτικά τμήματα των μηνυμάτων
- αφαιρεί τα επεξεργασμένα μέρη και να προωθεί τα μηνύματα στην επόμενη εφαρμογή, εφόσον υπάρχει. Κάθε SOAP μήνυμα ταξιδεύει από τον αποστολέα στους ενδιαμέσους κόμβους και τελικά φτάνει στον προορισμό. Κάθε ενδιαμέσος κόμβος χειρίζεται ένα κομμάτι του μηνύματος και το προωθεί στον επόμενο κόμβο.

Το ίδιο το SOAP δεν ασχολείται με θέματα όπως, η αναφορά στα αντικείμενα, η συλλογή καταναμετημένων απορριμμάτων λόγω μηνυμάτων, ή αποστολή δέσμης μηνυμάτων. Ίσως μελλοντικές υλοποιήσεις του πρωτοκόλλου να το κάνουν.

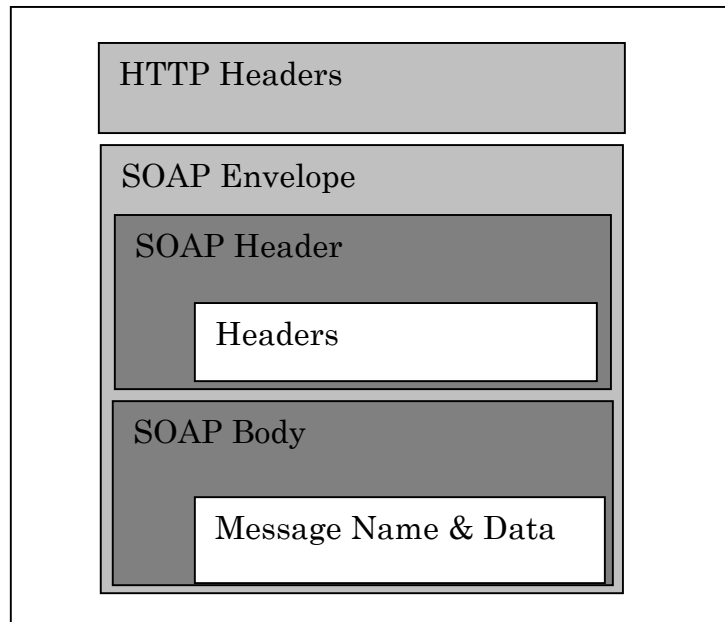


Σχήμα 3.1 Διαστρωμάτωση α- κωδικοποίησης δεδομένων [USA15, 28] β-μεταφοράς στο διαδίκτυο

Η διαστρωμάτωση των πρωτοκόλλων, κατά το σχήμα 3.1 μας επιτρέπει να εκτελούμε τις παραπάνω εργασίες.

3.1 Μηνύματα SOAP

Η δομή ενός μηνύματος SOAP φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.2 Δομή ενός μηνύματος SOAP, θεωρούμε μεταφέρεται με το HTTP

Το όλο μήνυμα είναι χωρισμένο σε δύο μέρη, το *Headers* και το *SOAP Envelope* (φάκελος). Το πρώτο περιλαμβάνει τις επικεφαλίδες του πρωτοκόλλου μεταφοράς, π.χ. HTTP, και τις επικεφαλίδες δέσμευσης αυτού με το SOAP μήνυμα, ενώ το δεύτερο, ο φάκελος SOAP, εσωκλείει τα δεδομένα. Ο φάκελος SOAP με τη σειρά του αποτελείται και αυτός από δύο μέρη, τις επικεφαλίδες SOAP (*SOAP Headers*) που εσωκλείει όλες τις μεμονωμένες επικεφαλίδες (Headers) και τη σώμα του μηνύματος SOAP (*SOAP Body*). Το σώμα του φακέλου (SOAP Body) εσωκλείει με τη σειρά του τα ονόματα και δεδομένα που μεταφέρονται με ένα μήνυμα SOAP.

Όπως βλέπουμε τα SOAP μηνύματα αποτελούνται από δύο στοιχεία, με δύο XML namespaces:

- `http://schemas.xmlsoap.org/envelope` για τη δομή του φακέλου
 - `http://schemas.xmlsoap.org/encoding` για την κωδικοποίηση του μηνύματος.
- Θεωρώντας προθέματα *soap-env* και *soap-enc*, για τα παραπάνω namespaces παρουσιάζουμε τη δομή ενός φακέλου και επικεφαλίδας SOAP.

```
<soap-env:Envelope>
  < soap-env:Header> ...</ soap-env:Header>
  < soap-env:Body>...</ soap-env:Body>
  <myNS:OtherStuff> ...</myNS:OtherStuff>
  <yourNS:OtherStuff> ...</yourNS:OtherStuff>
</ soap-env:Envelope>
```

Σχήμα 3.3 Μορφή ενός φακέλου SOAP

```

<soap-env:Header>
  <myNS:Date>2005-02-21</myNS:Date>
  <myNS:From>nojulian@hotmail.com</myNS:From>
  <myNS:To>myfriend@yahoo.com</myNS:To>
  ...
</soap-env:Header>

```

Σχήμα 3.4 Μορφή μιας επικεφαλίδας SOAP

Το header περιέχει στοιχεία που καθορίζονται από την εφαρμογή, (π.χ. τα myNS:Date, myNS:From, myNS:To) τα οποία ονομάζονται header entries. Επιπλέον το σώμα φακέλου του SOAP μηνύματος έχει την ίδια μορφή με την επικεφαλίδα του σχήματος 3.4, και αποτελείται και αυτό από αντίστοιχα body entries.

Τα μηνύματα λοιπόν που μεταφέρονται μέσω SOAP εσωκλείονται σε φακέλους SOAP (SOAP Envelops). Ο φάκελος SOAP μπορεί να μην έχει επικεφαλίδα (soap-env:Header) αλλά πρέπει να έχει σώμα (soap-env:Body) για να αποτελεί ένα έγκυρο μήνυμα SOAP.

Επιπλέον για την αναφορά λαθών το SOAP ορίζει ένα σύνολο από λάθη:

- VersionMismatch όταν υπάρχει σφάλμα στο Header,
- MustUnderstand όταν δεν έγινε κατανοητό το μήνυμα,
- Client για σφάλμα στο πελάτη
- Server για σφάλμα στον εξυπηρετητή

Πέραν αυτών όμως υπάρχει η δυνατότητα ορισμού νέων τύπων λαθών. Το SOAP ορίζει για τα λάθη, μόνο ένα body entry, το “soap-env:Fault” με στοιχεία τα:

- faultcode : αναφέρεται στο namespace
- faultstring : είναι η συμβολοσειρά λάθους που παράγεται
- faultactor : αυτός που παρήγαγε το λάθος
- detail : λεπτομέρειες για το λάθος

Παράδειγμα ενός λάθους βλέπουμε στο παρακάτω σχήμα:

```

<env:Fault>
  <faultcode>env:MustUnderstand</faultcode>
  <faultstring>SOAP Must Understand Error</faultstring>
</env:Fault>

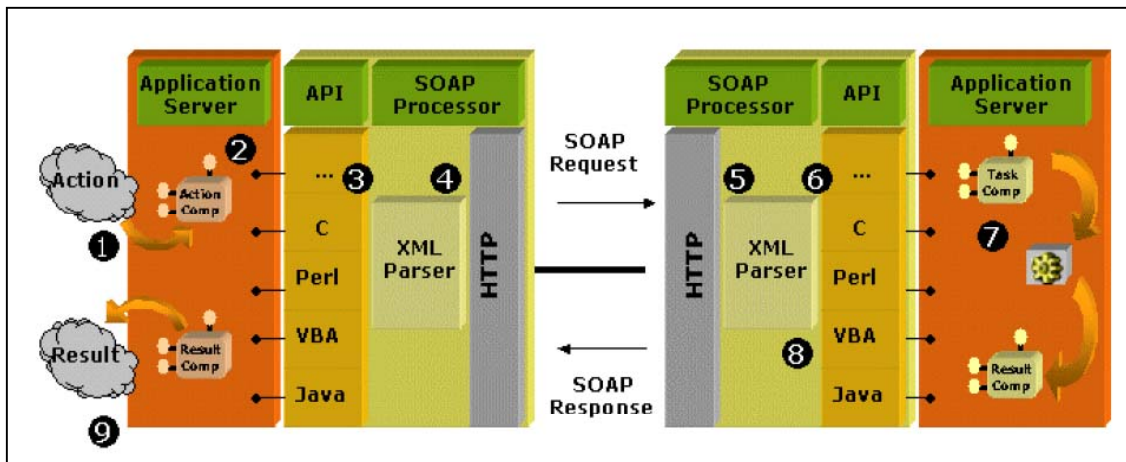
```

Σχήμα 3.5 Παράδειγμα αναφορά λάθους

Η κωδικοποίηση των μηνυμάτων SOAP είναι προαιρετική. Μπορούμε να χρησιμοποιήσουμε οποιοδήποτε δικό μας στυλ κωδικοποίησης για το μήνυμα, ή ακόμα μπορεί να ορισθεί ένα γνώρισμα (*encodingStyle*) για την κωδικοποίηση οποιουδήποτε στοιχείου στο μήνυμα.

3.2 Δράση του SOAP

Το γενικό σχήμα δράσης του SOAP παρουσιάζεται στο σχήμα 3.6.



Σχήμα 3.6 SOAP σε δράση, Action – Result μπορεί να είναι οποιαδήποτε ενέργεια – αποτέλεσμα, αντίστοιχα, π.χ. πρόσβαση σε μια βάση δεδομένων, παρουσίαση – μετασχηματισμός μιας ιστοσελίδας κτλ.

Το σχήμα αυτό αναδεικνύει το γεγονός ότι πολλές εταιρείες έχουν ενσωματώσει στους εξυπηρετητές εφαρμογών τους το πρωτόκολλο SOAP.

Αυτό που μας ενδιαφέρει είναι, η αποστολή και λήψη μηνυμάτων (SOAP Request – Response). Το SOAP χρησιμοποιεί το HTTP για μεταφορά. Για την αποστολή μιας αίτησης, HTTP Request καθορίζεται μια επικεφαλίδα δράσης (action header) που καθορίζει το τελικό σημείο δράσης. Επίσης για την λήψη απαντήσεων γίνεται χρήση του καθιερωμένης απάντησης, HTTP response, για επιτυχή ή αποτυχημένη απάντηση. Το μήνυμα απάντησης πρέπει να είναι ένα μήνυμα SOAP με μια εγγραφή λάθους, SOAP fault, στο σώμα του μηνύματος.

```
POST /axis/JuliansDateServer.jws HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.1
Host: 127.0.0.1
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 386

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getDate soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://DefaultNamespace"/>
  </soapenv:Body>
</soapenv:Envelope>
```

Σχήμα 3.7 Ένα ολόκληρο SOAP μήνυμα αίτησης για την χρησιμοποίηση της μεθόδου gateDate της υπηρεσίας JuliansDateServer για την ώρα. Τα στοιχεία που βρίσκονται στην κορυφή είναι τα στοιχεία επικεφαλίδας HTTP Headers (δες και σχήμα 3.2).

Και το μήνυμα απάντησης που επιστρέφει σε αυτή την αίτηση είναι:

```
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=FE557C3183AE648AB1A4C055C0E148C1; Path=/axis
Content-Type: text/xml; charset=utf-8
Date: Mon, 21 Feb 2005 12:08:05 GMT
Server: Apache-Coyote/1.1
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getDateResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://DefaultNamespace">
      <ns1:getDateReturn xsi:type="xsd:string">
        DATE NOW IS:Mon Feb 21 14:08:05 EET 2005
      </ns1:getDateReturn>
    </ns1:getDateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Σχήμα 3.8 Απάντηση στο μήνυμα της προηγούμενης αίτησης για την ώρα, που προσφέρει η υπηρεσία JuliansDateServer μέσω της μεθόδου της getDate

Επιπλέον έχουμε το παρακάτω μήνυμα λάθους φαίνεται στο σχήμα 3.9 β:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getQuote
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://DefaultNamespace">
      <symbol xsi:type="xsd:string">SDAQ</symbol>
    </ns1:getQuote>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>java.net.UnknownHostException: www.xmltoday.com</faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

Σχήμα 3.9 SOAP φάκελος αίτησης ζητώντας την τιμή της μετοχής, με σύμβολο “SDAQ”, στη μέθοδο getQuote της υπηρεσίας StockQuoteService που προσφέρει ενημέρωση μετοχών, α) πάνω αίτηση, β) κάτω αναφορά σφάλματος

Με αναφορά και στο σχήμα 3.6, παραπάνω, μπορούμε να πούμε ότι πράγματι το SOAP προάγει την διαλειτουργικότητα αφού διαφορετικές συσκευές, λογισμικά και λειτουργικά συστήματα μπορεί να βρίσκονται εκατέρωθεν της επικοινωνίας και να ανταλλάσσουν πληροφορίες χρησιμοποιώντας το. Οι αρχικοί στόχοι του SOAP ήταν:

- τα υπάρχοντα προγράμματα να γίνουν περισσότερο προσιτά στους χρήστες,
 - να μην χρειάζεται επινόηση νέων τεχνολογιών και προδιαγραφών
 - το πρωτόκολλο HTTP ένα από τα δυνατά πρωτόκολλα μεταφοράς
- Χρησιμοποιώντας το HTTP, το SOAP κατάφερε:

- να περάσει μέσα από τις ζώνες ηλεκτρονικής προστασίας (Firewalls)
- να φτάνει στα άκρα του διαδικτύου, όπου φτάνει και το HTTP
- να είναι ασφαλές κάνοντας χρήση HTTPS
- να είναι απλό, χρησιμοποιώντας XML

Βέβαια υπάρχουν το SOAP έχει και μειονεκτήματα. Οι απαιτήσεις του καθορίζουν ότι:

- δεν απαιτεί επικύρωση των μηνυμάτων, π.χ. με DTD
- όταν προκύπτει μήνυμα λάθους η απάντηση λάθους πρέπει να επιστρέφεται
- λόγω της XML τα δεδομένα περιγράφονται με λέξεις, και άρα καταναλώνει εύρος ζώνης

Όπως και να 'χει όμως, το SOAP προάγει την διαλειτουργικότητα μεταξύ συσκευών και είναι απλό. Ακόμη, το πρωτόκολλο είναι ανοιχτό, με την υποστήριξη του W3C και δέχεται και βελτιστοποιήσεις. Αν και υπάρχουν κάποιες υλοποιήσεις και η τρέχουσα έκδοση είναι η "1.2", η έκδοση "1.1" είναι η καθιερωμένη [URL-10] [URL-15]. Στην πλατφόρμα SOAP4J της Apache υπάρχει υποστήριξη, βασικά, για απομακρυσμένη κλήση υπηρεσίας (remote procedure call – RPC). Στην πλατφόρμα .NET της Microsoft τα συστατικά (components) είναι προσιτά δικτυακά μόνο με SOAP. Κάποιοι επίσης προσπαθούν για μια γέφυρα επικοινωνίας μεταξύ SOAP και COM τεχνολογίας [URL-16].

4. Γλώσσα Περιγραφής Υπηρεσιών Διαδικτύου (Web Service Description Language – WSDL)

Αφού το SOAP έγινε διαθέσιμο σαν ένας μηχανισμός ανταλλαγής XML μηνυμάτων μεταξύ των επιχειρήσεων, παρουσιάστηκε η ανάγκη ενός τρόπου για την περιγραφή των εφαρμογών καθώς και του τρόπου με τον οποίο γίνεται η συνεργασία τους.

Η γλώσσα WSDL είναι η γραμματική για την περιγραφή των υπηρεσιών διαδικτύου σαν μια συλλογή από τελικά σημεία πρόσβασης, ικανά να ανταλλάσσουν μηνύματα. Το WSDL είναι η «συνταγή» για την αυτοματοποίηση των λεπτομερειών που συμμετέχουν σε μια συναλλαγή μεταξύ εφαρμογών.

Πρόκειται, ουσιαστικά, για ένα XML schema, που αναπτύχθηκε από την Microsoft και την IBM με σκοπό:

- να περιγράψει τι μπορεί να κάνει η υπηρεσία
- να δηλώσει το που βρίσκεται
- να περιγράψει το πως θα γίνει η χρησιμοποίησή της
- να καθορίσει τον τρόπο σύνδεσης της υπηρεσίας με τα μηνύματα SOAP 1.1, HTTP POST/GET μεθόδους, και τους τύπους των δεδομένων

Σε ένα βαθμό η WSDL δεν είναι διαφορετική από την γλώσσα περιγραφής διεπαφής (Interface Definition Language – IDL) CORBA IDL, ή Microsoft IDL. Όλες χρησιμοποιούνται για τον ορισμό της διεπαφής και τύπους δεδομένων για ένα συγκεκριμένο κομμάτι της επιχειρηματική λογικής. Επιπλέον όμως η WSDL προσφέρει έναν βαθμό επεκτασιμότητας πέραν των άλλων, επιτρέποντας την:

- Να περιγράφει τελικά σημεία πρόσβασης και τα μηνύματά τους, ανεξάρτητα από τον τύπο του μηνύματος και τα πρωτόκολλα δικτύου που χρησιμοποιούνται για τη μεταφορά τους
- Να χειρίζεται τα μηνύματα σαν μια αφηρημένη περιγραφή των δεδομένων που ανταλλάσσονται
- Να χειρίζεται τους τύπους των τελικών σημείων πρόσβασης σαν αφηρημένη συλλογή λειτουργιών των υπηρεσιών διαδικτύου

Σχηματικά ένα WSDL έγγραφο έχει τη δομή του παρακάτω σχήματος, σχήμα 4.1. Πρόκειται για XML περιγραφή των στοιχείων που αναγράφονται:

- Τύποι δεδομένων (<types>): είναι η περιγραφή των τύπων των δεδομένων που χρησιμοποιούνται στα Μηνύματα. Εμφανίζεται μόνο μια ή καμία φορά στο WSDL έγγραφο και μπορεί να περιέχει περισσότερα του ενός στοιχεία για τον καθορισμό των τύπων (XML schema) των δεδομένων που ανταλλάσσονται
- Μηνύματα (<message>): τα στοιχεία αυτά χρησιμοποιούνται να μοντελοποιήσουν την ανταλλαγή δεδομένων σαν τμήμα των υπηρεσιών. Σε ένα WSDL έγγραφο μπορεί να υπάρχουν περισσότερα του ενός τέτοια στοιχεία, καταγράφοντας, τα SOAP κυρίως, μηνύματα που διακινούνται
- Τύποι σημείων πρόσβασης (<PortType>): καθορίζουν ένα υποσύνολο των λειτουργιών που υποστηρίζονται για ένα τελικό σημείο πρόσβασης μιας υπηρεσίας διαδικτύου. Σε κάποιο βαθμό τα στοιχεία αυτά παρέχουν ένα μοναδικό προσδιορισμό σε ένα σύνολο από ενέργειες οι οποίες εκτελούνται από ένα τελικό σημείο πρόσβασης. Μπορεί να υπάρχουν περισσότερα του ενός τέτοια στοιχεία σε ένα WSDL έγγραφο.
- Σύνδεση (<binding>): τα στοιχεία αυτά αποτελούν μια συγκεκριμένη προδιαγραφή για το πρωτόκολλο και τον τύπο των δεδομένων για κάθε στοιχείο «τύπο σημείο πρόσβασης». Μπορεί να υπάρχουν περισσότερα του ενός τέτοια στοιχεία σε ένα WSDL έγγραφο.
- Υπηρεσία (<service>): Τα στοιχεία αυτά τυπικά παρουσιάζονται προς το τέλος κάθε WSDL εγγράφου και προσδιορίζουν μια υπηρεσία διαδικτύου. Μερικά WSDL έγγραφα ωστόσο δεν περιέχουν αυτό το στοιχείο, αφού ο κύριος ρόλος τους είναι η περιγραφή της διεπαφής και αυτό γίνεται από το στοιχείο *Σύνδεση* κατά βάση. Ένα στοιχείο <service> χρησιμοποιείται μόνο όταν περιγράφουμε το πραγματικό τελικό σημείο πρόσβασης της υπηρεσίας.

Μπορεί να υπάρχουν περισσότερα του ενός τέτοια στοιχεία σε ένα WSDL έγγραφο.



Σχήμα 4.1 Δομή ενός WSDL εγγράφου

Η δημιουργία λοιπόν ενός εγγράφου WSDL, αν και δύσκολη να γίνει από έναν άνθρωπο, μπορεί να αυτοματοποιηθεί. Πράγματι υπάρχουν πολλά εργαλεία που κάνουν το μετασχηματισμό από τον πηγαίο κώδικα στην αποτύπωση του WSDL αρχείου.

Επιπλέον, εφόσον η WSDL αποτελεί μια αφηρημένη περιγραφή της διεπαφής με την υπηρεσία διαδικτύου γίνεται αντιληπτό ότι, μπορεί να γίνει η υλοποίηση του κώδικα για την κλήση της υπηρεσίας διαδικτύου. Αυτό αποτελεί σημαντικό πλεονέκτημα, αφού για την συναλλαγή με μια υπηρεσία διαδικτύου, αρκεί να ανακαλύψουμε μόνο το WSDL έγγραφο της περιγραφής της. Ο τρόπος κλήσης της υπηρεσίας, ανακτώντας το WSDL έγγραφο περιγραφής, μπορεί να αυτοματοποιηθεί.

Ένα εργαλείο της Java μπορεί να κάνει και τις δύο εργασίες, δοθέντος του πηγαίου κώδικα Java να το μετασχηματίσει στο αντίστοιχο WSDL αρχείο, αλλά και το αντίστροφο, δοθέντος του WSDL εγγράφου που περιγράφει μια υπηρεσία να δημιουργήσει τον πηγαίο κώδικα κλήσης της. Συγκεκριμένα η προδιαγραφή Java της προγραμματιστικής διεπαφής εφαρμογών με XML και απομακρυσμένης κλήσης διαδικασίας (JAX-RPC) [URL-17], που αποτελεί τμήμα του πακέτου της Java σχετικά με υπηρεσίες διαδικτύου, παρέχει εργαλεία σχετικά με τους μετασχηματισμούς αυτούς. Επιπλέον, η Java παρέχει μηχανισμούς για τη μεταγλώττιση και δυναμική χρησιμοποίηση πηγαίου κώδικα χρησιμοποιώντας την μέθοδο της ανάκλασης. Όλ' αυτά μας δίνουν την δυνατότητα να αυτοματοποιήσουμε τη διαδικασία κλήσης μιας υπηρεσίας διαδικτύου, αν γνωρίζουμε το WSDL αρχείο της περιγραφής της ή να δημιουργήσουμε το WSDL αρχείο μιας υπηρεσίας διαδικτύου αν γνωρίζουμε τον πηγαίο κώδικά Java της.

4.1 Περιγραφή ανάπτυξης υπηρεσίας Διαδικτύου (web service deployment descriptor – WSDD)

Η ανάπτυξη της υπηρεσίας διαδικτύου, μετονομάζοντας το Java αρχείο πηγαίου κώδικα από *.java* σε *.jws* και τοποθετώντας το στον AXIS [AXI-1], ώστε να δημιουργηθεί αυτόματα το WSDL της υπηρεσίας, είναι πράγματι εύκολη διαδικασία ανάπτυξης. Έχει όμως ένα μειονέκτημα, πρέπει να γνωρίζουμε και να κατέχουμε τον πηγαίο κώδικα, και έπειτα όλες οι μέθοδοι, που ορίζονται, εκτίθενται στο κοινό κατά την δημιουργία του WSDL εγγράφου. Η περιγραφή ανάπτυξης υπηρεσιών Διαδικτύου, Web Service Deployment Descriptor (WSDD) ήρθε να δώσει τη λύση.

Μια υπηρεσία Διαδικτύου αποτελείται στην ουσία ένα κομμάτι λογισμικού που υλοποιεί την εργασία της υπηρεσίας στηριζόμενο σε μεθόδους που υλοποιούνται μέσα σ' αυτό. Με το WSDD περιγράφουμε ακριβώς ποιες μέθοδοι από την εφαρμογή εκτίθενται στην υπηρεσία του κοινού. Αυτό γίνεται όταν έχουμε τον αντικείμενο (object code), ή εκτελέσιμο, Java κώδικα και περιγράφουμε στο σύστημά μας ποια είναι τα τελικά σημεία πρόσβασης, ποιες είναι οι μέθοδοι που μπορεί να καλέσει ο απομακρυσμένος χρήστης και με ποια δικαιώματα.

Γίνεται κατανοητό ότι, με την περιγραφή αυτή, έχουμε την ευχέρεια να δώσουμε αυθαίρετα ονόματα για αναφορά στις μεθόδους του πηγαίου κώδικα που τελικά εκθέτουμε, ή ακόμα και στην ονομασία της υπηρεσίας. Το WSDL που δημιουργείται κατά την ανάπτυξη μια υπηρεσίας μέσω περιγραφέα WSDD, μπορεί να αναγράφει τελικά ως μεθόδους και υπηρεσίες τις αναφορές αυτές, που εμείς δώσαμε, και ως τελικά σημεία πρόσβασης της υπηρεσίας την πραγματική διεπαφή με το αντικείμενο κώδικα.

5. Παγκόσμια Περιγραφή Ανακάλυψη και Ολοκλήρωση (Universal Description Discovery and Integration – UDDI) [URL-19]

Η ύπαρξη των υπηρεσιών διαδικτύου έχει νόημα μόνο εφόσον οι δυνητικοί χρήστες τους μπορούν να βρουν επαρκείς πληροφορίες για να μπορέσουν τελικά να τις χρησιμοποιήσουν, αυτό αφορά πληροφορίες σχετικά την ανακάλυψη και την χρησιμοποίησή τους.

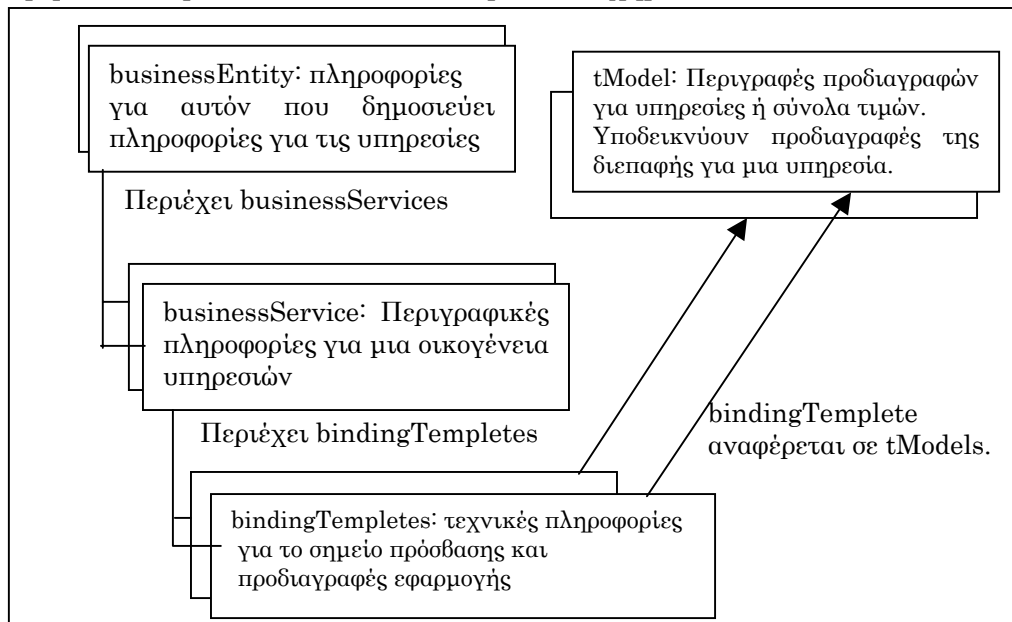
Η καθολική περιγραφή, ανακάλυψη και ολοκλήρωση (UDDI) έχει σκοπό τον καθορισμό ενός συνόλου υπηρεσιών που υποστηρίζουν την περιγραφή και την ανακάλυψη των επιχειρήσεων, οργανισμών, και άλλων φορέων παροχής υπηρεσιών Διαδικτύου, τις υπηρεσίες Διαδικτύου που αυτοί προσφέρουν και τις τεχνικές διεπαφές που μπορούν να χρησιμοποιηθούν για να έχουν πρόσβαση σε αυτές τις υπηρεσίες. Με βάση ένα κοινό σύνολο βιομηχανικών προτύπων, HTTP, XML, XML-Schemas και του SOAP, η UDDI παρέχει μια διαλειτουργική, θεμελιώδη, βασισμένη στις υπηρεσίες Διαδικτύου, υποδομή και ένα περιβάλλον εφαρμογών για τη δημοσίευση και των δημόσιων υπηρεσιών αλλά και των υπηρεσιών που εκτίθενται από το εσωτερικό ιδιωτικό δίκτυο ενός οργανισμού.

Η προδιαγραφή της UDDI καθορίζει ένα μοντέλο πληροφόρησης αποτελούμενο από στιγμιότυπα, μόνιμων αποθηκευμένων δεδομένων, τα οποία ονομάζονται οντότητες.

Οι οντότητες εκφράζονται σε XML και είναι μονίμως αποθηκευμένες από έναν κόμβο UDDI. Κάθε οντότητα έχει τον τύπο της ρίζας του XML εγγράφου, που την περιγράφει. Η προδιαγραφή καθορίζει τους τύπους:

- **businessEntity**: Περιγράφει μια επιχείρηση ή οποιονδήποτε οργανισμό που προσφέρει υπηρεσίες Διαδικτύου
- **businessService**: Περιγράφει μια συλλογή από υπηρεσίες διαδικτύου οι οποίες προσφέρονται από έναν οργανισμό, που περιγράφεται από μια **businessEntity**
- **tModel**: Περιγράφει ένα τεχνικό σχήμα για την περιγραφή μιας υπηρεσίας διαδικτύου, όπως π.χ. έναν τύπο υπηρεσίας, ένα πρωτόκολλο που χρησιμοποιείται από μια υπηρεσία ή μια κατηγορία συστημάτων, προάγοντας έτσι το σκεπτικό της επαναχρησιμοποίησης
- **publisherAssertion**: Περιγράφει, σε σχέση με μια **businessEntity**, τη σχέση που αυτή έχει σε σχέση με μια άλλη **businessEntity**.
- **subscription**: Περιγράφει το διαρκές αίτημα για συγκρατηθούν οι αλλαγές στις οντότητες που περιγράφονται από την συνδρομή

Η δομή των δεδομένων ακολουθεί το παρακάτω σχήμα:



Σχήμα 5.1 Βασικές δομές δεδομένων και μεταξύ τους σχέσεις

Ένας κόμβος UDDI έχει τα παρακάτω χαρακτηριστικά:

- Υποστηρίζει αλληλεπίδραση με τα δεδομένα, που αποθηκεύονται σε αυτόν
- Είναι μέλος μόνο ενός μητρώου UDDI
- Ένας κόμβος UDDI εννοιολογικά έχει πρόσβαση και χειρίζεται ένα πλήρες αντίγραφο δεδομένων UDDI, που διαχειρίζονται από τον κατάλογο του οποίου είναι ένα μέλος. Επιπλέον, ένας κόμβος, είναι το στοιχείο αυτό που χειρίζεται οποιαδήποτε ερώτηση και δημοσιεύει την προγραμματιστική διεπαφή που υποστηρίζεται από τον κόμβο.

Η προδιαγραφή, δεν καθορίζει την φυσική υλοποίηση ενός κόμβου UDDI. Ένας οι περισσότεροι κόμβοι UDDI μπορεί να συνδυαστούν και να σχηματίσουν το μητρώο UDDI. Το μητρώο έχει τα παρακάτω χαρακτηριστικά:

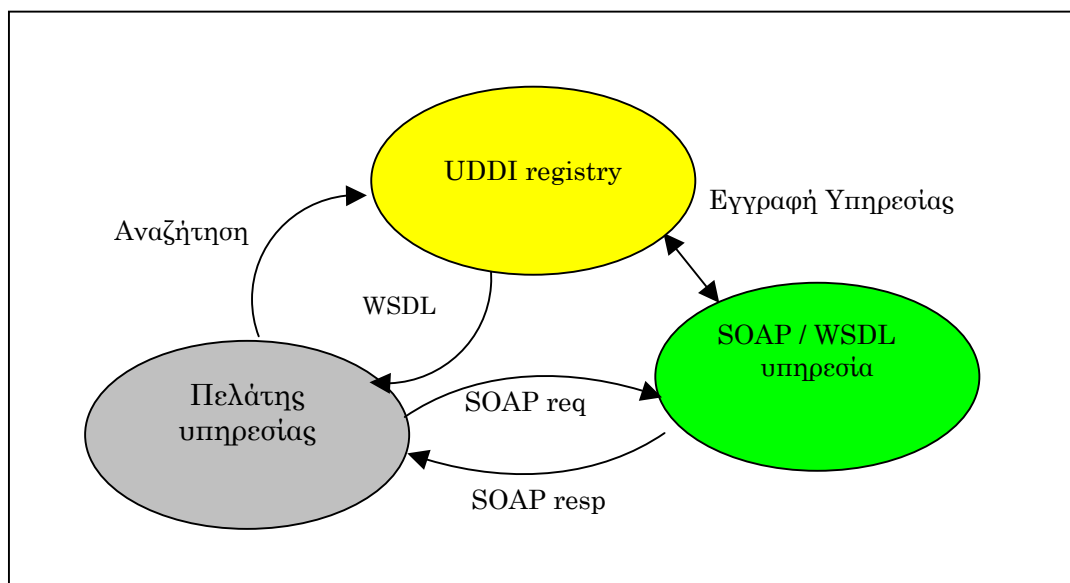
- Αποτελείται από έναν ή περισσότερους κόμβους

- Οι κόμβοι του καταλόγου διαχειρίζονται συλλογικά ένα σύνολο από καλά καθορισμένα δεδομένα. Αυτό, τυπικά, υποστηρίζεται από την αντιγραφή ανάμεσα στους κόμβους του μητρώου που βρίσκονται σε διαφορετικά λειτουργικά συστήματα.
- Το μητρώο πρέπει να δημιουργήσει μια πολιτική αποφάσεων και μπορεί να μεταβιβάσει αυτή την πολιτική στους κόμβους.

Για να έχει νόημα η αναζήτηση των υπηρεσιών θα πρέπει η περιγραφή τους, σε έναν κόμβο UDDI, να περιλαμβάνει πληροφορίες πέρα από τις καθαυτό τεχνικές προδιαγραφές της. Βασικός στόχος του UDDI καταλόγου είναι η παρουσίαση των δεδομένων και των μεταδεδομένων σχετικά με τις υπηρεσίες διαδικτύου. Το μητρώο UDDI προσφέρει έναν μηχανισμό για την ταξινόμηση, δεικτοδότηση και διαχείριση των υπηρεσιών Διαδικτύου, ώστε αυτές να ανακαλυφθούν και να καταναλωθούν. Η τρέχουσα προδιαγραφή της UDDI επιτρέπει:

- Την ανακάλυψη υπηρεσιών Διαδικτύου βασιζόμενοι στον ορισμό μιας αφηρημένης διεπαφής
- Την ανακάλυψη παρόχων υπηρεσιών Διαδικτύου που ταξινομούνται σύμφωνα με ένα γνωστό σχήμα ταξινόμησης ή σύστημα προσδιορισμού
- Τον προσδιορισμό της ασφάλειας και των πρωτοκόλλων μεταφοράς που υποστηρίζονται από μια δεδομένη υπηρεσία Διαδικτύου
- Την αναζήτηση για υπηρεσίες βασιζόμενοι σε μια λέξη κλειδί
- Την προσωρινή αποθήκευση τεχνικών πληροφοριών για μια υπηρεσία Διαδικτύου και έπειτα την ενημέρωση του μητρώου σχετικά με αυτή την πληροφορία κατά το χρόνο εκτέλεσης.

Και με το UDDI η χρησιμοποίησης μια υπηρεσίας Διαδικτύου αποτυπώνεται στο παρακάτω σχήμα.



Σχήμα 5.2 Η UDDI και ο ρόλος της στην ανακάλυψη των υπηρεσιών

Το σχήμα αυτό υποδεικνύει το ρόλο της UDDI στην αναζήτηση των υπηρεσιών διαδικτύου. Ο πελάτης αναζητά εγγεγραμμένες υπηρεσίες στο μητρώο UDDI, αφού

ανακαλύψει την υπηρεσία που επιθυμεί χρησιμοποιεί SOAP και συναλλάσσεται με αυτή. Και έτσι συμπληρώνει το θέμα της ανακάλυψης των υπηρεσιών Διαδικτύου.

Η UDDI καθορίζει το μητρώο κατά το χρόνο εκτέλεσης, όταν οι κόμβοι είναι ενεργοί και συνδέονται μεταξύ τους. Οι διάφοροι κόμβοι όμως μπορεί να μην είναι συνεχώς συνδεδεμένοι. Όταν ελέγχεται το μητρώο για μια υπηρεσία, θα πρέπει να αναμένουμε απάντηση απ' όλους τους κόμβους για να είμαστε σίγουροι για την πιο έγκαιρη πληροφορία.

Επιπλέον η αντιγραφή των δεδομένων μεταξύ των κόμβων δημιουργεί διάφορα προβλήματα. Βασικά, το πρωτόκολλο αντιγραφής μεταξύ των κόμβων υπάρχει για να λύσει το πρόβλημα της «ελλιπής» πληροφόρησης των κόμβων, αλλά η αντιγραφή σε συνδυασμό με το γεγονός ότι οι κόμβοι μπορεί να αποσυνδέονται κατά καιρούς, είναι αυτό που δημιουργεί το πρόβλημα.

Ακόμα, όταν μια υπηρεσία Διαδικτύου παύει να λειτουργεί ή βγαίνει εκτός για κάποιο λόγο ή αναβαθμίζεται θα πρέπει να ενημερωθούν όλοι οι κόμβοι για αυτό, ώστε το μητρώο να παρέχει έγκαιρη πληροφορία.

Τέλος η τεχνολογία που ακολουθεί η UDDI είναι τεχνολογία πελάτη – εξυπηρετητής, από την άποψη ότι, ο κάθε UDDI κόμβος θα πρέπει να γνωρίζει καλά το που θα βρει τους άλλους κόμβους. Άρα όλοι οι κόμβοι θα πρέπει να έχουν σταθερές, μόνιμες και επιλύσιμες διευθύνσεις.

Με όλ' αυτά κάποιες από τις πληροφορίες σε ένα μητρώο UDDI μπορεί να μην είναι και οι πιο έγκυρες πληροφορίες που θα μπορούσαν να προσφέρονται σε έναν πελάτη που αναζητά πληροφορίες για υπηρεσίες. Όμως η UDDI δέχθηκε δυνατή υποστήριξη και μάλιστα από ισχυρές εταιρείες στην βιομηχανία ανάπτυξης υπηρεσιών Διαδικτύου, όπως οι Accenture, Ariba, Commerce One, Fujitsu Limited, Hewlett-Packard, i2 Technologies, Intel, International Business Machines Corporation, Microsoft, Oracle, SAP AG, Sun Microsystems, και VeriSign.

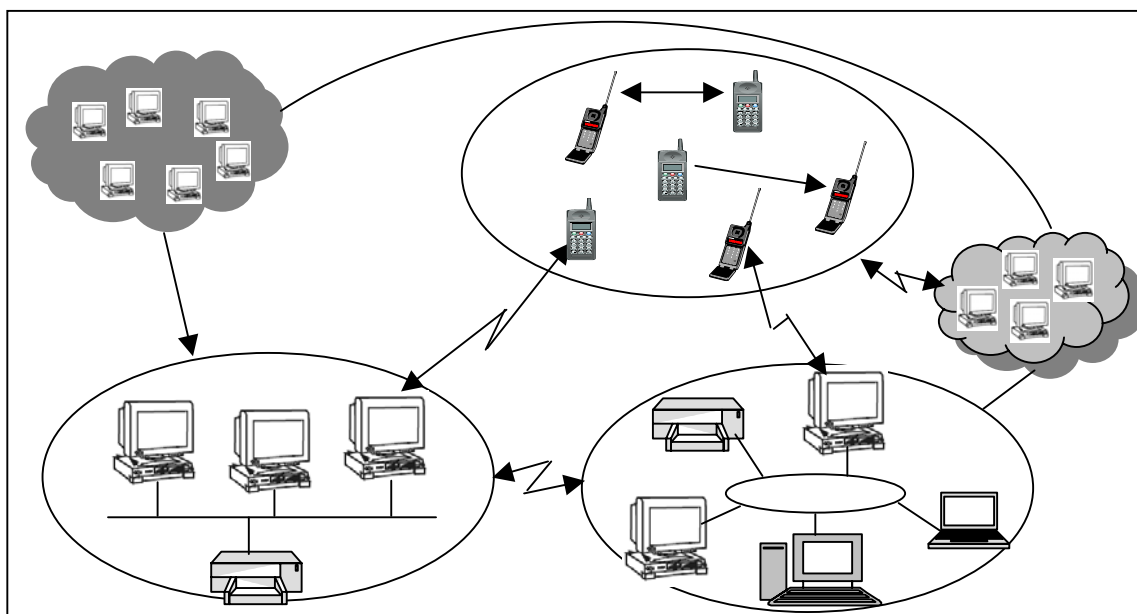
ΜΕΡΟΣ II

ΤΕΧΝΟΛΟΓΙΑ ΟΜΟΤΙΜΩΝ
ΟΝΤΟΤΗΤΩΝ

1. Εισαγωγή

Πολλοί από εμάς τείνουν να σκέφτονται το Διαδίκτυο, αλλά και τα άλλα δίκτυα, ως συστήματα πελατών εξυπηρετητών. Οι υπεύθυνοι για την ανάπτυξη ιστοχώρων ξοδεύουν πολύ χρόνο χτίζοντας δυνατούς ιστοχώρους που προσφέρουν πληροφορίες και υπηρεσίες σε χιλιάδες από τους χρήστες που τους επισκέπτονται καθημερινά.

Εάν όμως πάρουμε για μια στιγμή για να εξετάσουμε το ίδιο το Διαδίκτυο, θα δούμε ότι υπάρχουν εκατομμύρια συσκευές που συνδέονται στο δίκτυο οποιαδήποτε στιγμή. Όλοι οι υπολογιστές συνδέονται θεωρητικά ο ένας με τον άλλο, και οι πληροφορίες που ανταλλάσσονται και αποθηκεύονται μπορούν να προσεγγιστούν από τον οποιοδήποτε. Συνολικά, η τοπολογία ή το σχεδιάγραμμα των υπολογιστών στο διαδίκτυο είναι η ομαδοποίηση των μηχανών που βρίσκονται διάσπαρτες στις διάφορες θέσεις τους, Σχήμα 1.1. Μέσα σε κάθε μια από τις ομάδες ή τα υποδίκτυα, οι συσκευές θα είναι ορατές σε άλλες συσκευές στο υποδίκτυο και μερικές φορές στο εξωτερικό Διαδίκτυο.



Σχήμα 1.1 Το Διαδίκτυο, συνένωση όλων των δικτύων

Μερικές από τις συσκευές αυτές θα εξυπηρετούν αιτήσεις και θα φιλοξενούν πληροφορίες, παραδείγματος χάριν οι μηχανές Yahoo! που εξυπηρετούν με το περιεχόμενο σελίδων διαδικτύου είναι εξυπηρετητές διαδικτύου. Φυλλομετρώντας το Yahoo! από τον υπολογιστή μας μετατρέπουμε τον υπολογιστή μας σε μια «μηχανή πελάτη». Αυτός ο τύπος αλληλεπίδρασης πελατών - εξυπηρετητών συμβαίνει για τις εκατοντάδες χιλιάδες των υπολογιστών συγχρόνως.

Ενώ μια μηχανή πελατών «κοιτάζει» βιαστικά ιστοσελίδες στο Yahoo!, θα μπορούσε παράλληλα να διαμοιράζεται έναν τοπικό δίσκο με τα μέλη ομάδας χρηστών. Σε αυτήν την κατάσταση, η μηχανή μας λειτουργεί ως ένας εξυπηρετητής σε οποιοδήποτε «πελάτη» που προσπαθεί να έχει πρόσβαση στα αρχεία στην τοπικό δίσκο.

Η τεχνολογία ομότιμων οντοτήτων (γνωστή με την Αγγλική της ορολογία peer-to-peer ή P2P) επιτρέπει σε οποιαδήποτε δίκτυο – ενήμερη συσκευή για να παρέχει τις υπηρεσίες σε μια άλλη δίκτυο-ενήμερη συσκευή. Μια συσκευή σε ένα P2P δίκτυο μπορεί να παρέχει την πρόσβαση σε οποιοδήποτε τύπο πόρου που έχει στη διάθεσή της, είτε αυτό είναι έγγραφα, ικανότητα αποθήκευσης, υπολογιστική ισχύ, ή ακόμα και τον άνθρωπο χειριστή της. Αν και ο όρος P2P ηχεί σαν μια πρόσκαιρη μόδα του ηλεκτρονικού εμπορίου, η τεχνολογία είναι μια φυσική επέκταση της φιλοσοφίας του διαδικτύου και της ανάγκης για διοικητική αποκέντρωση. Με τον ίδιο τρόπο που το διαδίκτυο παρέχει την αναζήτηση ονόματος περιοχών (DNS), τον παγκόσμιο ιστό (World Wide Web), το ηλεκτρονικό ταχυδρομείο, και άλλες υπηρεσίες μοιράζοντας την ευθύνη για τις υπηρεσίες αυτές μεταξύ των εκατομμυρίων εξυπηρετητών υπολογιστών, η τεχνολογία P2P έχει την ικανότητα να τροφοδοτήσει ολόκληρο ένα νέο σύνολο γερών εφαρμογών μοιράζοντας τους πόρους που διαδίδονται σε όλες τις γωνίες του διαδικτύου.

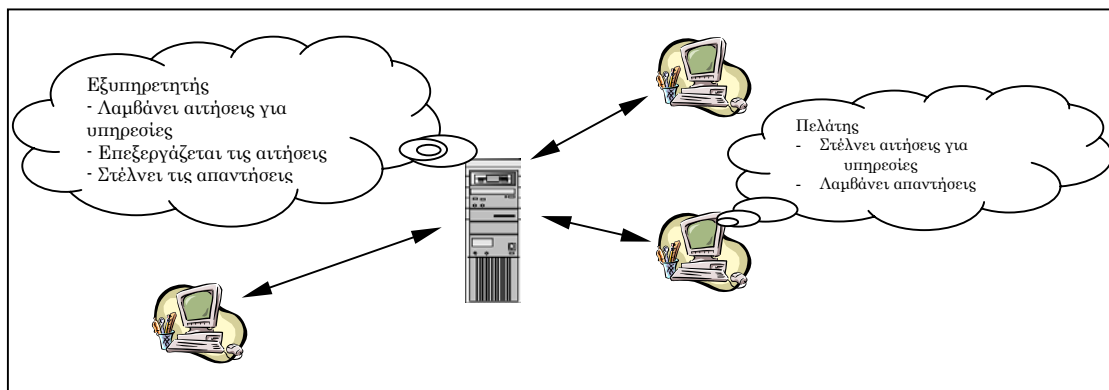
1.1 Από την αρχιτεκτονική πελάτης – εξυπηρετητής στην τεχνολογία ομότιμων οντοτήτων (P2P technology)

Οι περισσότερες υπηρεσίες διαδικτύου διανέμονται χρησιμοποιώντας την παραδοσιακή αρχιτεκτονική πελάτης - εξυπηρετητής, όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 1.1.1). Σε αυτήν την αρχιτεκτονική, υπάρχουν δύο διακριτοί ρόλοι, οι πελάτες που ζητούν υπηρεσίες και οι εξυπηρετητές που προσφέρουν υπηρεσίες. Οι μεν πελάτες είναι εκείνοι που ξεκινούν την διαδικασία όταν επιθυμούν μια υπηρεσία, συνδέονται με έναν εξυπηρετητή χρησιμοποιώντας ένα συγκεκριμένο πρωτόκολλο επικοινωνιών, όπως π.χ. το πρωτόκολλο μεταφοράς αρχείων (FTP), για να λάβουν πρόσβαση σε έναν συγκεκριμένο πόρο, π.χ. τη μεταφόρτωση - «κατέβασμα» ενός αρχείου. Το μεγαλύτερο μέρος της επεξεργασίας στην προσφορά μιας υπηρεσίας εμφανίζεται συνήθως στον εξυπηρετητή, αφήνοντας τον πελάτη σχετικά ξαλαφρωμένο. Οι πιο δημοφιλείς εφαρμογές Διαδικτύου, συμπεριλαμβανομένου του παγκόσμιου ιστού, το FTP, Telnet, και το ηλεκτρονικό ταχυδρομείο, χρησιμοποιούν αυτό το πρότυπο παράδοσης υπηρεσίας.

Δυστυχώς, αυτή η αρχιτεκτονική έχει ένα σημαντικό μειονέκτημα. Δεδομένου ότι ο αριθμός πελατών αυξάνεται, οι απαιτήσεις φορτίων και εύρους ζώνης στον εξυπηρετητή αυξάνονται επίσης, αποτρέποντας τελικά τον εξυπηρετητή από το να χειριστεί πρόσθετους πελάτες. Για να αντεπεξέλθουν στην αυξανόμενη ζήτηση για τις υπηρεσίες τους, οι εταιρείες, προσπαθούν να μοιράσουν την κίνηση προς τους εξυπηρετητές τους χρησιμοποιώντας πολύπλοκες τεχνικές εξισορρόπησης φορτίου (load balancing) και ανανεώνοντας τους υπάρχοντες πόρους τους (αναβάθμιση συσκευών, εύρος ζώνης).

Το πλεονέκτημα αυτής της αρχιτεκτονικής είναι ότι απαιτεί τη λιγότερη υπολογιστική δύναμη από την πλευρά πελατών. Ειρωνικά, οι περισσότεροι χρήστες έχουν επιδιώξει να αναβαθμίσουν τους ηλεκτρονικούς υπολογιστές τους σε επίπεδα που είναι παράλογα ανώτερα για τις δημοφιλέστερες εφαρμογές διαδικτύου, όπως η περιαγωγή στον παγκόσμιο ιστό, ανάκτηση του ηλεκτρονικού ταχυδρομείου και η μεταφορά αρχείων.

Ο πελάτης στην αρχιτεκτονική πελάτης - εξυπηρετητής ενεργεί σε έναν παθητικό ρόλο, ικανό να απαιτεί υπηρεσίες από τους εξυπηρετητές αλλά ανίκανο να προσφέρει υπηρεσίες σε άλλους πελάτες. Το πρότυπο αυτό της παροχής υπηρεσιών, αναπτύχθηκε σε μία εποχή που οι περισσότερες μηχανές στο διαδίκτυο είχαν μια επιλύσιμη στατική διεύθυνση IP, που σημαίνει ότι όλες οι άλλες μηχανές στο διαδίκτυο θα μπορούσαν να βρουν η μια την άλλη εύκολα χρησιμοποιώντας ένα απλό όνομα (όπως www.yahoo.com) και χρησιμοποιώντας έναν γνωστό εξυπηρετητή ονομάτων περιοχών (DNS).



Σχήμα 1.2 Αρχιτεκτονική πελάτης εξυπηρετητής

Καθώς το Διαδίκτυο αυξάνονταν, ο πεπερασμένος αριθμός των διευθύνσεων IP προέτρεψε τους φορείς παροχής υπηρεσιών να αρχίσουν να παραχωρούν δυναμικά τις διευθύνσεις IP στις μηχανές κάθε φορά που συνδέονταν με το δίκτυο μέσω των συνδέσεων διεπιλογών (dial-up). Η δυναμική φύση των διευθύνσεων IP αυτών των μηχανών απέτρεψε τους χρήστες από το να «τρέχουν» προγράμματα για χρήσιμους εξυπηρετητές και να παρέχουν υπηρεσίες. Αν και κάποιος θα μπορούσε ακόμα να «τρέξει» έναν εξυπηρετητή, ακόμη και εκείνος ο χρήστης δεν θα μπορούσε να έχει πρόσβαση σε αυτόν τον εξυπηρετητή εκτός αν ήξερε τη διεύθυνση IP της μηχανής εκ των προτέρων.

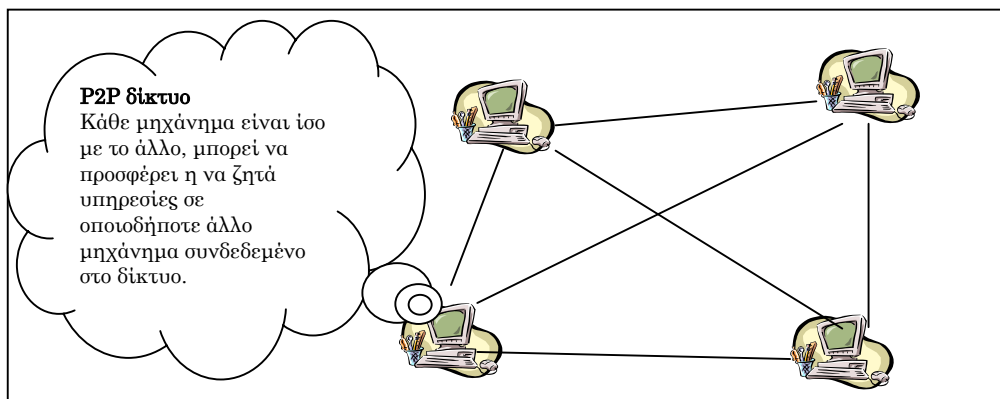
Ένας επιπλέον λόγος ότι οι υπολογιστές των περισσότερων χρηστών δεν μπορούν να τρέξουν εξυπηρετητές, είναι ότι οι μηχανές αυτές συνήθως αποτελούν μέρος ενός ιδιωτικού δικτύου, που οργανώνεται συνήθως από το τμήμα πληροφορικής της εταιρίας τους. Το ιδιωτικό αυτό δίκτυο απομονώνεται συνήθως από το διαδίκτυο από ένα τείχος ηλεκτρονικής προστασίας (firewall), μια συσκευή με σκοπό να αποτρέπει τις αυθαίρετες συνδέσεις προς και από το ιδιωτικό δίκτυο. Οι εταιρίες δημιουργούν συνήθως ένα ιδιωτικό δίκτυο για να εξασφαλίσουν τις ευαίσθητες εταιρικές πληροφορίες καθώς επίσης και για να αποτρέψουν την κατάχρηση ή την κακή χρήση δικτυακών πόρων. Η παρενέργεια αυτής της τεχνολογίας είναι ότι ένας υπολογιστής έξω από το ιδιωτικό δίκτυο δεν μπορεί να συνδεθεί με έναν υπολογιστή μέσα στο ιδιωτικό δίκτυο για να λάβει υπηρεσίες που αυτός προσφέρει.

Αυτές οι συσκευές, αποτελούν τα «άκρα» του διαδικτύου, μηχανές που συνδέονται αλλά ανίκανες να συμμετάσχουν σε ανταλλαγή υπηρεσιών. Για αυτόν τον λόγο, οι περισσότερες χρήσιμες υπηρεσίες (email, ftp, www, news) συγκεντρώνονται σε κεντρικούς υπολογιστές με τις επιλύσιμες διευθύνσεις IP, απ' όπου μπορούν να δώσουν πρόσβαση στον καθένα που ξέρει το ευκολομνημόνευτο όνομα περιοχών του κεντρικού υπολογιστή (λ.χ.

πηγαίνοντας στο www.yahoo.com μπορούμε να κάνουμε χρήση email). Εάν όμως, όλες οι μηχανές στο δίκτυο έτρεχαν και ως εξυπηρετητής και ως πελάτης, θα διαμόρφωσαν την αρχή ενός στοιχειώδους P2P δικτύου.

Υπολογισμοί σε όγκους πληροφοριών που διακινούνται μέσω P2P δικτύων επιδεικνύουν μεγάλες δυνατότητες της P2P τεχνολογίας [COMP-1, 2, 3]. Αν υποθέσουμε για παράδειγμα ότι μόνο 10 χιλιάδες συσκευές (υπολογιστές, υπολογιστές χειρός κτλ) του 1GHz συνδέονται με το Διαδίκτυο σε οποιοδήποτε χρόνο, που καθεμία κατέχει μόνο 100MB του αχρησιμοποίητου χώρου αποθήκευσης, 100bps του αχρησιμοποίητου εύρους ζώνης, και 10% την αχρησιμοποίητη ισχύ επεξεργασίας. Σε οποιοδήποτε χρόνο, αυτοί οι «πελάτες» αντιπροσωπεύουν 1 Terabytes (10.000GB) ανεκμετάλλετου αποθηκευτικού χώρου, 1 Mbps του διαθέσιμου εύρους ζώνης (περίπου 250 KBps), και 10^4 GHz της σπαταλημένης επεξεργαστικής ισχύος!

Αυτές είναι συντηρητικές εκτιμήσεις που υπαινίσσονται μόνο το τεράστιο αναξιοποίητο δυναμικό που αναμένει να εξαπολυθεί από τα «άκρα» του Διαδικτύου. Η τεχνολογία P2P είναι το κλειδί στην πραγματοποίηση αυτής της δυνατότητας, παρέχοντας στις μεμονωμένες μηχανές ένα μηχανισμό για προσφορά υπηρεσιών μιας στην άλλη. Αντίθετα από την αρχιτεκτονική πελατών - εξυπηρετητή, τα δίκτυα P2P δεν στηρίζονται σε έναν κεντρικό υπολογιστή για την παροχή πρόσβασης σε υπηρεσίες, και λειτουργούν συνήθως έξω από το σύστημα ονόματος περιοχών (DNS). Όπως παρουσιάζεται στο παρακάτω σχήμα (Σχήμα 1.1.2), τα δίκτυα P2P αποφεύγουν τη συγκεντρωμένη οργάνωση της αρχιτεκτονικής πελάτη εξυπηρετητή και υιοθετούν αντί αυτού μια επίπεδη, ιδιαίτερα διασυνδεόμενη αρχιτεκτονική. Επιτρέποντας περιοδικά συνδεδεμένους υπολογιστές να βρουν ο ένας τον άλλον, η τεχνολογία P2P επιτρέπει σε αυτές τις μηχανές να ενεργήσουν και ως πελάτες και εξυπηρετητές, ικανές να μπορούν να καθορίζουν τις υπηρεσίες που προσφέρουν στο P2P δίκτυο και να συμμετέχουν σε αυτές τις υπηρεσίες με κάποιο ορισμένο από εφαρμογή τρόπο.



Σχήμα 1.3 Επίπεδη αρχιτεκτονική με χρήση ομότιμων οντοτήτων

Το κύριο πλεονέκτημα των δικτύων P2P είναι ότι μοιράζουν τα καθήκοντα προσφοράς υπηρεσιών μεταξύ όλων των ομότιμων οντοτήτων (peers) - συσκευών στο δίκτυο, πράγμα που αποβάλλει τις διακοπές παροχής υπηρεσιών λόγω ενός ενιαίου σημείου της αποτυχίας, του κεντρικού εξυπηρετητή, παρέχοντας έτσι, μια πιο εξελικτική λύση για την προσφορά των υπηρεσιών. Επιπλέον, τα δίκτυα P2P εκμεταλλεύονται το διαθέσιμο εύρος ζώνης σε

ολόκληρο το δίκτυο με τη χρησιμοποίηση ποικίλων καναλιών επικοινωνίας και με την πλήρωση του εύρους ζώνης στη «άκρη» του διαδικτύου.

Αντίθετα από τις παραδοσιακές επικοινωνίες πελάτης - εξυπηρετητής, στις οποίες οι συγκεκριμένες διαδρομές στους δημοφιλείς προορισμούς μπορούν να υπερφορτωθούν (π.χ. η διαδρομή προς το Amazon.com ή το eBay), η τεχνολογία P2P επιτρέπει την επικοινωνία μέσω ποικίλων διαδρομών δικτύων, με αυτόν τον τρόπο μειώνοντας τη συμφόρηση δικτύων. Η τεχνολογία P2P έχει την ικανότητα της προσφοράς πόρων με υψηλή διαθεσιμότητα και με πολύ χαμηλότερο κόστος, μεγιστοποιώντας τη χρήση των πόρων από κάθε ισότιμη οντότητα που συνδέεται με το P2P δίκτυο. Εκτιμώντας ότι οι λύσεις πελάτης - εξυπηρετητής στηρίζονται στην προσθήκη του δαπανηρού εύρους ζώνης, εξοπλισμού, και διευκολύνσεις τοποθέτησης για να διατηρήσουν μια δυνατή λύση, η τεχνολογία P2P μπορεί να προσφέρει ένα παρόμοιο επίπεδο ευρωστίας με τη εξάπλωση των δικτύων και των πόρων μέσω το P2P δίκτυο. Επιχειρήσεις όπως η Intel και η IBM χρησιμοποιούν ήδη P2P τεχνολογία για να μειώσουν το κόστος για μεταφορά στα έγγραφα και τα λοιπά αρχεία τους σε ολόκληρη την επιχείρηση.

Δυστυχώς όμως, η τεχνολογία P2P πάσχει και αυτή από μερικά μειονεκτήματα λόγω της φύσης της δομής ενός P2P δικτύου. Η διανεμημένη μορφή καναλιών επικοινωνιών στα P2P δίκτυα οδηγεί σε αιτήσεις για υπηρεσίες μη ντετερμινιστικής φύσης. Παραδείγματος χάριν, οι πελάτες που ζητούν τον ακριβή ίδιο πόρο από το P2P δίκτυο, μπορεί να συνδεθούν με τις εξ ολοκλήρου διαφορετικές μηχανές μέσω διαφορετικών διαδρομών επικοινωνίας, με διαφορετικά αποτελέσματα. Αιτήματα που στέλνονται μέσω ενός P2P δικτύου μπορεί να μην οδηγήσουν σε μια άμεση απάντηση και, σε μερικές περιπτώσεις, να μην οδηγήσουν σε οποιαδήποτε απάντηση. Οι πόροι σε ένα P2P δίκτυο μπορούν να εξαφανιστούν κατά περιόδους καθώς αυτοί που φιλοξενούν εκείνους τους πόρους αποσυνδέονται από το δίκτυο, κάτι που έρχεται σε αντιπαράθεση με τις ως τώρα πρακτικές στο διαδίκτυο, όπου οι υπηρεσίες που παρέχονται έχουν τους πόρους τους συνεχώς διαθέσιμους.

Εντούτοις, η τεχνολογία P2P μπορεί να υπερνικήσει όλους αυτούς τους περιορισμούς. Αν και οι πόροι θα εξαφανίζονται κατά περιόδους, μια P2P εφαρμογή μπορεί να εφαρμόσει τη λειτουργία της αντανάκλασης (mirror) για τους δημοφιλέστερους πόρους σε περισσότερες του ενός ισότιμες οντότητες, ελαχιστοποιώντας έτσι αιτήσεις για πρόσβαση στον πόρο της και αυξάνοντας ταυτόχρονα την προσφορά του συγκεκριμένου πόρου. Έτσι μεγαλύτεροι αριθμοί διασυνδεδεμένων ισότιμων οντοτήτων μειώνουν την πιθανότητα ότι ένα αίτημα για μια υπηρεσία θα πάει αναπάντητο. Εν ολίγοις, η ίδια η δομή ενός P2P δικτύου που προκαλεί το πρόβλημα μπορεί να χρησιμοποιηθεί και για να λύσει το πρόβλημα.

1.2 Περιγραφή της τεχνολογίας P2P

Στα περισσότερα P2P συστήματα, η διάκριση των ρόλων μεταξύ ενός κεντρικού υπολογιστή - εξυπηρετητή και ενός πελάτη είναι θολή. Ο ίδιος ο υπολογιστής θα μπορούσε να είναι συνδεδεμένος με άλλους υπολογιστές χρησιμοποιώντας μια τοπολογία δαχτυλιδιών, αλλά όταν αναφερόμαστε σε ένα

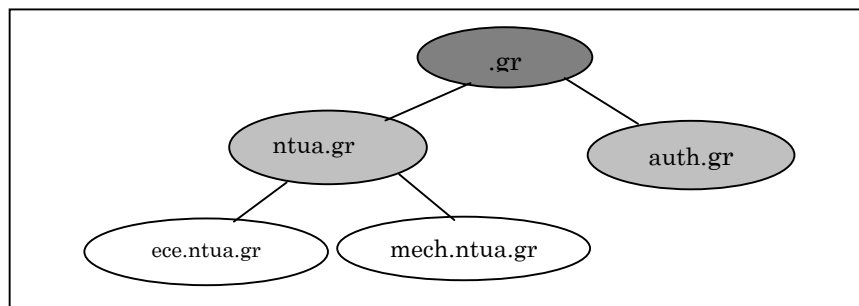
P2P δίκτυο, εννοούμε μια λογική τοπολογία η οποία ορίζει τον τρόπο διανομής πληροφορίας μεταξύ των αυτόνομων οντοτήτων. Έτσι, οι ισότιμες οντότητες (peers) θα μπορούσαν όλες να επικοινωνούν με έναν κεντρικό υπολογιστή για να διανείμουν την πληροφορία, όπως στο Napster κατά την αναζήτηση ενός μουσικού αρχείου, αλλά η τελική «συναλλαγή» θα γίνει μεταξύ των δυο ισότιμων οντοτήτων, εκ των οποίων η μία θα προσφέρει και η άλλη θα «καταναλώνει» την υπηρεσία.

Στις περισσότερες περιπτώσεις, οι ισότιμοι θα συνδεθούν ο ένας με τον άλλο μέσω του Διαδικτύου χρησιμοποιώντας είτε το TCP είτε το HTTP πρωτόκολλο. Όπως γνωρίζουμε, το TCP/IP είναι το θεμελιώδες πρωτόκολλο για τη μεταφορά των πληροφοριών μέσω του Διαδικτύου. Το πρωτόκολλο HTTP χτίζεται πάνω από το TCP/IP και επιτρέπει την επικοινωνία μεταξύ των υπολογιστών για τη μεταφορά υπερκειμένου - ιστοσελίδων. Το HTTP είναι πολύ δημοφιλές για τα P2P συστήματα επειδή οι περισσότερες οργανισμοί επιτρέπουν την μεταφορά υπερκειμένου στην πόρτα 80 [RFC1700], αφού μόνο έτσι μπορούν να προσφέρουν έναν ιστοχώρο πίσω από ένα εταιρικό δίκτυο.

Διάφορες τοπολογίες δικτύων μπορούν να χρησιμοποιηθούν για τη σύνδεση P2P των συστημάτων. Παρακάτω συζητάμε τις σημαντικότερες P2P τοπολογίες δικτύων προκειμένου να εξηγηθεί πώς οι πληροφορίες μπορούν να διαβιβαστούν μεταξύ των οντοτήτων αποτελεσματικά. Πρόκειται για λογικές απεικονίσεις των δικτύων, όπως αυτά διακινούν την πληροφορία, και δεν θα πρέπει να συγχέονται με τις φυσικές τοπολογίες, που διακινούν τα δεδομένα.

Η ιεραρχική τοπολογία

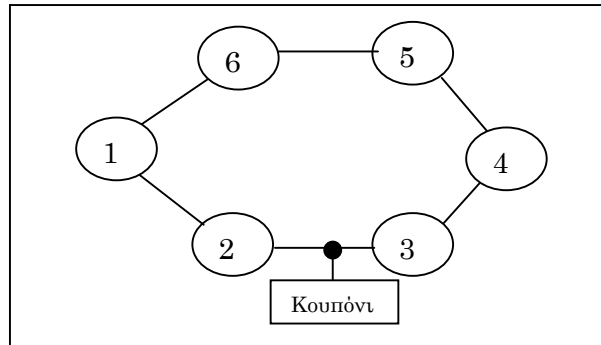
Μια από τις πιο κοινές τοπολογίες είναι η ιεραρχία. Κάθε φορά που δακτυλογραφούμε τη διεύθυνση ενός ιστοχώρου, URL, στον φυλλομετρητή μας, χρησιμοποιείται το σύστημα αντιστοίχισης ονομάτων σε IP διευθύνσεις αποκαλούμενο dns, καλώντας τον εξυπηρετητή ονόματος περιοχών (dns server) για να εντοπιστεί η IP διεύθυνση του μηχανήματος που προσφέρει αυτόν τον ιστόχωρο. Αυτό το σύστημα οργανώνεται σε μια ιεραρχία, με τους κεντρικούς υπολογιστές ρίζας σε τα πολύ ανώτερα επίπεδα. Η τοπολογία ιεραρχίας μοιάζει με το παρακάτω σχήμα. Για αρκετά έτη τώρα, οι κριτικοί έχουν απαιτήσει μια εξέταση της DNS αρχιτεκτονικής επειδή οι κεντρικοί υπολογιστές ρίζας αντιπροσωπεύουν ένα ενιαίο σημείο της αποτυχίας. Εντούτοις, επειδή η πιθανότητα του dns συστήματος να βγαίνει εκτός λειτουργίας είναι μικρή, καμία πραγματική εργασία δεν έχει εμφανιστεί σε αυτήν την περιοχή.



Σχήμα 1.4. Λειτουργία DNS

Η τοπολογία δαχτυλιδιών

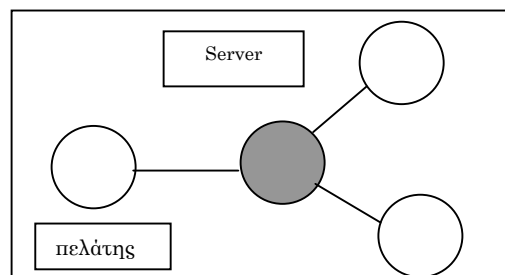
Η τοπολογία δικτύων «δαχτυλίδι με κουπόνι» χρησιμοποιεί την έννοια του κουπονιού, το οποίο περνάει από έναν υπολογιστή στον άλλο σε μια ομάδα υπολογιστών που είναι συνδεδεμένοι σε ένα σχέδιο δαχτυλιδιού. Όταν μια μηχανή λαμβάνει το κουπόνι, μπορεί να στείλει τις πληροφορίες της. Η τοπολογία δαχτυλιδιών δεν χρησιμοποιείται πολύ για την διάδοση της πληροφορίας στα κοινά δίκτυα, αλλά παρέχει ένα ενδιαφέρον μοτίβο για την εξισορρόπηση φορτίου ενός συστήματος ή μιας ιεραρχίας εξυπηρετητών υπολογιστών. Το παρακάτω σχήμα παρουσιάζει μια τοπολογία δαχτυλιδιών.



Σχήμα 1.5. Δακτύλιος με κουπόνι

Τοπολογία πελάτης - εξυπηρετητής ή συγκεντρωτική

Η πιο κοινή τοπολογία είναι ο πελάτης εξυπηρετητής, ή συγκεντρωτική τοπολογία. Η ορολογία του πελάτη - εξυπηρετητή μας ήταν γνωστή για πολλά έτη, ενώ πρόσφατα, ο όρος «συγκεντρωτική» τοπολογία χρησιμοποιείται για να περιγράψει ένα σύστημα στο οποίο ένας ενιαίος υπολογιστής, ο κεντρικός υπολογιστής ή εξυπηρετητής, καθιστά τις υπηρεσίες διαθέσιμες πέρα από το δίκτυο. Οι μηχανές πελατών έρχονται σε επαφή με τον κεντρικό υπολογιστή όταν απαιτούνται οι υπηρεσίες. Προφανώς, όσο περισσότεροι πελάτες είναι στο σύστημα, τόσο «μεγαλύτερος» πρέπει να είναι και ο κεντρικός υπολογιστής για να ανταποκριθεί στις αιτήσεις τους. Προκειμένου να αντιμετωπιστεί ο όγκος κυκλοφορίας από όλους τους αυξανόμενους πελάτες, κάποια στιγμή, θα πρέπει να στηθεί κι άλλος κεντρικός υπολογιστής που να εξυπηρετεί στο ίδιο πράγμα. Το παρακάτω σχήμα παρουσιάζει ένα παράδειγμα τέτοιας τοπολογίας.

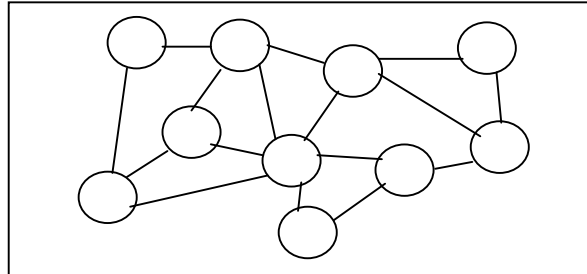


Σχήμα 1.6. Μόνο η κεντρική οντότητα, ο εξυπηρετητής, είναι σε θέση να προσφέρει υπηρεσίες

Η αποκεντρωμένη τοπολογία

Η αποκεντρωμένη τοπολογία είναι μια τοπολογία δικτύων που είναι πιο κοντά στην έννοια peer-to-peer. Όταν ένα πακέτο των πληροφοριών αρχίζει το ταξίδι του στο δίκτυο, είναι βασικά διακινούμενο μέσω μιας αποκεντρωμένης

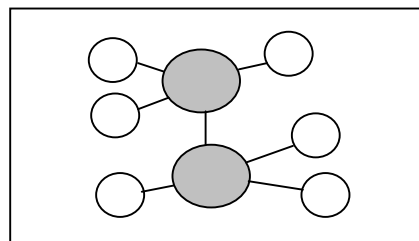
τοπολογίας. Οι πληροφορίες μέσα το ίδιο στο πακέτο λένε σε κάθε κόμβο πού να σταλεί το πακέτο έπειτα. Το σχήμα που ακολουθεί παρουσιάζει ένα παράδειγμα μιας αποκεντρωμένης τοπολογίας δικτύων. Βασικά, όλοι οι υπολογιστές στο σύστημα ενεργούν και ως πελάτες και κεντρικοί υπολογιστές (ισότιμα μεταξύ τους), και διαχειρίζονται αιτήσεις άλλων ενώ ταυτόχρονα κάνοντας τις δικές τους αιτήσεις. Οι εφαρμογές KaZaA και Gnutella χρησιμοποιούν αυτήν την αποκεντρωμένη τοπολογία για τα P2P συστήματά τους.



Σχήμα 1.7. Κατανεμημένη τοπολογία, δεν υπάρχει κεντρικός έλεγχος, όλες οι οντότητες είναι ισότιμες, τα μηνύματα πληροφορίας δρομολογούνται από την επόμενη οντότητα που τα λαμβάνει

Η υβριδική τοπολογία

Η υβριδική τοπολογία που παρουσιάζεται στο παρακάτω σχήμα, αποτελεί μια μίξη ανάμεσα στην συγκεντρωτική και την αποκεντρωμένη τοπολογία. Η πληροφορία ουσιαστικά δρομολογείται αυστηρώς δια μέσου των κεντρικών υπολογιστών. Οι πελάτες όταν χρειάζονται πληροφορίες για κάποιους πόρους θα έρθουν σε επαφή με έναν κεντρικό υπολογιστή για να τις λάβουν, οι πληροφορίες αυτές αφορούν την ύπαρξη των πόρων, το όνομα της ομότιμης οντότητας που κατέχει του πόρους κτλ.. Ο αρχικός πελάτης που ζητά κάποιους πόρους έπειτα θα έρθει σε επαφή με ομότιμη οντότητα που κατέχει τους πόρους αυτούς για να τους λάβει. Με την υβριδική αυτή τοπολογία, ένας υπολογιστής μπορεί να είναι είτε πελάτης είτε ένας κεντρικός υπολογιστής. Αυτή είναι η τοπολογία που χρησιμοποιείται στο σύστημα Napster, ODC ή DC++ μεμονωμένοι υπολογιστές επικοινωνούν με έναν σταθερά εντοπισμένο κεντρικό υπολογιστή για την αναζήτηση και αφού βρουν τον υπολογιστή που κατέχει τα μέσα που τους ενδιαφέρει έρχονται σε επαφή με αυτόν άμεσα για τη μεταφόρτωση μέσων (downloading).



Σχήμα 1.8. Οι κεντρικές οντότητες είναι υπεύθυνες για τη διανομή της πληροφορίας

Αυτή η τοπολογία, μπορούμε να πούμε πως χρησιμοποιείται, κατά κάποιο τρόπο, κατά την ανακάλυψη των πληροφοριών σε ένα καθολικό μητρώο UDDI. Το καθολικό μητρώο UDDI αποτελείται από κόμβους UDDI που ο καθένας τους συγκρατά ορισμένες πληροφορίες για υπηρεσίες διαδικτύου. Ο έλεγχος για την

ύπαρξη υπηρεσιών γίνεται από τους κόμβους UDDI, στους οποίους συνδέονται οι πελάτες. Οι κόμβοι αυτοί έπειτα επικοινωνούν με τους υπόλοιπους κόμβους για την αναζήτηση της πληροφορίας και, με την εύρεση της πληροφορίας, ο πελάτης αλληλεπιδρά κατευθείαν με την υπηρεσία Διαδικτύου. Αυτό όμως δεν θα πρέπει να συγχέεται με την αρχιτεκτονική P2P. Οι κόμβοι γνωρίζουν εκ των προτέρων την ύπαρξη των άλλων κόμβων.

1.3 Γιατί είναι σημαντική η τεχνολογία P2P

Αν και P2P κέρδισε την κακή φήμη σαν ένα μέσο για παράνομη διανομή πνευματικής ιδιοκτησίας, η τεχνολογία P2P έχει να προσφέρει περισσότερα από την εύκολη πρόσβαση στην κλεμμένα video αρχεία ή μουσική.

Η αναβάθμιση των σύγχρονων υπολογιστών σε υψηλά επίπεδα δημιουργεί πόρους στα άκρα του Διαδικτύου που μένουν αναξιοποίητοι. Από την άλλη πολύπλοκες υπολογιστικές εργασίες χρειάζονται περισσότερους πόρους απ' ότι ένας υπολογιστής ή ένα υπολογιστικό σύστημα μπορεί να προσφέρει. Αναφερόμαστε κυρίως σε περιπτώσεις κατανεμημένου υπολογισμού όπου η τεχνολογία P2P είναι πολλά υποσχόμενη.

Στον ευρωπαϊκό οργανισμό για πυρηνικές έρευνες CERN ερευνούν την τεχνολογία P2P ώστε να την προσαρμόσουν στις ανάγκες τους, συγκεκριμένα στα πειράματα φυσικής υψηλής ενέργειας. Τρέχον πρόγραμμα αποτελεί η απόκτηση δεδομένων από το πείραμα συμπαγούς σωληνοειδούς μιονίου, το οποίο εκτελείται στο Big Hadron Collider, ένα μεγάλο δαχτυλίδι στο οποίο υποατομικά σωματίδια επιταχύνονται και έρχονται σε μεταξύ τους σύγκρουση. Τα αποτελέσματα της σύγκρουσης αναλύονται ώστε να απαντηθεί η βασική ερώτηση της φύσης για το ποια είναι η σύσταση της ύλης. Τα αποτελέσματα της σύγκρουσης καταγράφονται από διάφορους ανιχνευτές. Τέτοια πειράματα όταν εκτελούνται συνθέτουν δισεκατομμύρια πηγές πληροφορίας με υψηλούς ρυθμούς παραγωγής δεδομένων που πρέπει να επεξεργαστούν και να αποθηκευτούν για μετέπειτα ανάλυση.

Ένα συνδεδεμένο σύστημα απόκτησης δεδομένων (Data Acquisition - DAQ) για τα πειράματα φυσικής υψηλής ενέργειας πρέπει να συλλέξει, να συνδυάσει και να φιλτραρίσει όλα τα στοιχεία που καταγράφονται από τους κατάλληλους ανιχνευτές. Το πείραμα εκτελείται σε μια ονομαστική συνολική απόδοση δεδομένων 100 Gigabyte/Sec. Τα παραδοσιακά, μονολιθικά συστήματα υπολογισμού δεν έχουν τις απαραίτητες ικανότητες επικοινωνίας και επεξεργασίας να καλύψουν αυτήν την απαίτηση. Επομένως, για να επιτευχθεί η διαδικασία, τίθεται η ανάγκη κατανομής της συγκέντρωσης των δεδομένων σε μια συστάδα ηλεκτρονικών υπολογιστών.

Μια υποδομή μεσισμικού αποκαλούμενη XDAQ (Cross DAQ) [CERN] έχει αναπτυχθεί κατά τη διάρκεια των προηγούμενων ετών για να εκτελέσει αυτή τη διαδικασία. Το XDAQ είναι ένα «κουτί εργαλείων» προγραμματισμού για τα κατανεμημένα συστήματα αποκλήσεων δεδομένων. Παρέχει υπηρεσίες ανεξάρτητες των πλατφόρμων, εργαλεία για την τοπική και απομακρυσμένη διαδικαστική επικοινωνία, τη διαμόρφωση και τον έλεγχο. Ο ρόλος του μεσισμικού είναι να διευκολυνθεί ο σχεδιασμός, ο προγραμματισμός και η διαχείριση εφαρμογών απόκτησης δεδομένων με την παροχή ενός απλού, συνεπούς κατανεμημένου περιβάλλοντος προγραμματισμού. Το εργαλείο αυτό

χτίζεται επάνω στα βιομηχανικά πρότυπα, σε ανοικτά πρωτόκολλα και βιβλιοθήκες. Το XDAQ έχει αρχίσει από το 2000 και έχει χρησιμοποιηθεί από τότε στο Κέντρο Πυρηνικών Μελετών και Ερευνών (CERN) για τις εφαρμογές πρωτοτύπων των τοπικών συστημάτων DAQ.

Αυτήν την περίοδο, διερευνάται η ενσωμάτωση της πλατφόρμας JXTA [CERN] ως ευκολοσύνδετο συστατικό στο πλαίσιο τους, και εξετάζουν τα σενάρια ανακαλύψεων στα μικτά ιδιωτικά - δημόσια δίκτυα για την χρησιμοποίηση οντοτήτων με διαθέσιμους να προσφέρουν, πόρους.

2. Η εξέλιξη της τεχνολογίας P2P

Η τεχνολογία P2P πάντα υπήρχε, αλλά όχι με αυτήν τη μορφή, κεντρικοί υπολογιστές με σταθερές IP διευθύνσεις είχαν πάντα την ικανότητα να επικοινωνήσουν με άλλους κεντρικούς υπολογιστές και να έχουν πρόσβαση στις υπηρεσίες τους. Διάφορες προ p2p εφαρμογές, όπως το ηλεκτρονικό ταχυδρομείο και το σύστημα ονόματος περιοχών στηρίζονται σε αυτές τις ικανότητες που παρέχονται από καταναμημένα δίκτυα, αλλά μια τέτοια εφαρμογή το USENET, εξέχει.

Το USENET δημιουργήθηκε το 1979 από δύο τελειόφοιτους σπουδαστές στο πανεπιστήμιο της Βόρειας Καρολίνα, τον Tom Truscott και Jim Ellis, για να παρέχει έναν τρόπο για δύο υπολογιστές να ανταλλάξουν τις πληροφορίες τις πριν από την ανάπτυξη του Διαδικτύου. Η πρώτη προσπάθεια τους επέτρεπε σε έναν υπολογιστή για να καλεί έναν άλλο υπολογιστή, να ελέγξει για τα νέα αρχεία, και να μεταφορτώσει εκείνα τα αρχεία αυτό έγινε τη νύχτα για να σώσει στις μεγάλης απόστασης τηλεφωνικές δαπάνες. Το σύστημα αυτό εξελίχθηκε στο ογκώδες σύστημα ομάδων πληροφόρησης που είναι σήμερα. Εντούτοις, όσο μεγάλο και είναι, το USENET έχει μερικές ιδιότητες που βοηθούν να το διακρίνουν ως πιθανώς την πρώτη P2P εφαρμογή. Το USENET δεν έχει καμία κεντρική διευθύνουσα αρχή που - κάθε κόμβος διαχειρίζεται τη διανομή του περιεχομένου, και το περιεχόμενο του δικτύου USENET ανανεώνεται (γενικά ή εν μέρει) στους κόμβους του.

Το USENET δεν είναι ένα κομμάτι του λογισμικού ή ένα δίκτυο των κεντρικών υπολογιστών, βέβαια απαιτεί και το λογισμικό και κεντρικούς υπολογιστές για να λειτουργήσει, αυτά τα πράγματα δεν καθορίζουν αληθινά USENET. Στον πυρήνα του, το USENET είναι απλά ένας τρόπος για τις μηχανές να μιλήσουν η μία στην άλλη, για να επιτρέψει σε μηνύματα ειδήσεων να μεταδοθούν και να διαδοθούν πέρα από ένα δίκτυο. Με την παροχή ενός καθορισμένου με σαφήνεια πρωτοκόλλου, το πρωτόκολλο μεταφοράς ειδήσεων δικτύων [IETF - RFC 977], ένας ευρύτερος πιθανός αριθμός μηχανημάτων μπορεί να συμμετέχει ανεξάρτητα για να παρέχει τις υπηρεσίες ειδήσεων. Αυτή η κατανομή των αρμοδιοτήτων είναι που διακρίνει το USENET, που καθιστά την αναγνωρίσιμη ως πρώτο αληθινό, αν και πρώτο στοιχείο, εφαρμογή P2P της τεχνολογίας.

Από USENET, οι δημοφιλέστερες P2P εφαρμογές έχουν εμπέσει στη μια από τρεις σημαντικές κατηγορίες: ανταλλαγή στιγμιαίων μηνυμάτων, μοίρασμα αρχείων και καταναμημένος υπολογισμός.

2.1 Στιγμιαία μηνύματα (instant messages)

Όταν ο Mirabilis ανακοίνωσε το ICQ (www.icq.com) τον Νοέμβριο του 1996, έδωσε στους χρήστες του έναν γρηγορότερο τρόπο να επικοινωνούν με τους φίλους από το παραδοσιακό ηλεκτρονικό ταχυδρομείο. Το ICQ επιτρέπει στους χρήστες να ειδοποιούνται όταν έρχονται ανοιχτή γραμμή οι φίλοι τους και να στείλουν τα στιγμιαία μηνύματα στους φίλους τους αυτούς. Εκτός από την κύρια ικανότητα ανταλλαγής μηνυμάτων, το ICQ επιτρέπει στους χρήστες να ανταλλάσσουν και αρχεία. Αν και είναι ταξινομημένο ως P2P εφαρμογή, το ICQ στηρίζεται σε μια υβριδική τοπολογία αρχιτεκτονικών πελατών - κεντρικών υπολογιστών για να παρέχει την υπηρεσία του. Συγκεκριμένα ένας κεντρικός υπολογιστής ελέγχει ανά πάσα στιγμή ποιοι χρήστες είναι σε ανοιχτή γραμμή και έτοιμοι να ανταλλάξουν μηνύματα. Έτσι όταν νέοι χρήστες έλθουν σε ανοιχτή γραμμή, να ειδοποιηθούν τα ενδιαφερόμενα συμβαλλόμενα μέρη τότε αυτού συνδέθηκαν με το δίκτυο. Όλη η άλλη επικοινωνία μεταξύ των χρηστών διευθύνεται σε έναν P2P τρόπο, με τα μηνύματα να «φρέουν» άμεσα από μια μηχανή του χρήστη σε αυτή του άλλου χωρίς να μεσολαβεί κάποιος κεντρικός υπολογιστής.

Από την εμφάνισή του, το ICQ είχε πολλούς μιμητές, όπως το MSN Messenger (www.messenger.msn.com), το Yahoo! Messenger και τον AOL Messenger (www.messenger.yahoo.com και www.aol.com αντίστοιχα). Αυτές οι εφαρμογές όμως δεν είναι συμβατές μεταξύ τους, επειδή καθεμία στηρίζεται σε ιδιόκτητο πρωτόκολλο επικοινωνίας. Ως αποτέλεσμα αυτού του ασυμβιβάστου, οι χρήστες πρέπει να μεταφορτώσουν, να «κατεβάσουν», το διαφορετικό λογισμικό πελατών (MSN, ICQ, Yahoo, AOL) και να περάσουν από μια χωριστή διαδικασία εγγραφής για κάθε δίκτυο. Επειδή οι περισσότεροι χρήστες επιλέγουν να αποφύγουν αυτήν την δυσχέρεια, αυτά τα δίκτυα έχουν αυξηθεί στις απολύτως χωριστές κοινότητες χρηστών που δεν μπορούν να αλληλεπιδράσουν.

Διάφοροι υπεύθυνοι ανάπτυξης λογισμικού έχουν προσπαθήσει να γεφυρώσουν αυτές τις χωριστές κοινότητες με την αντίστροφη μηχανική των πρωτοκόλλων στιγμιαίων μηνυμάτων και την παραγωγή του νέου λογισμικού πελατών. Μια τέτοια εφαρμογή, Jabber (www.jabber.com), παρέχει τις πύλες σε όλες τις σημαντικές υπηρεσίες στιγμιαίων μηνυμάτων, που επιτρέπουν στους χρήστες για να αλληλεπιδράσει ο ένας με τον άλλον στα διάφορα δίκτυα. Αυτή η προσπάθεια έχει συναντήσει την αντίσταση από τους φορείς παροχής υπηρεσιών, που προέτρεψαν την AOL να αλλάξει το πρωτόκολλο επικοινωνίας της σε μία προσπάθεια να εμποδίσουν οι πελάτες Jabber.

2.2 Διανομή αρχείων

Το Napster (www.napster.com) εμφανίστηκε στο Διαδίκτυο το 1999, παρέχοντας στους χρήστες του την ικανότητα να ανταλλάσσουν μουσικά αρχεία (MP3). Το Napster υιοθετεί μια υβριδική P2P λύση παρόμοια με ICQ, που στηρίζεται σε έναν κεντρικό υπολογιστή για να αποθηκεύσει έναν κατάλογο μουσικών αρχείων στη μηχανή κάθε χρήστη. Αυτός ο κεντρικός υπολογιστής

είναι επίσης αρμόδιος για να επιτρέψει στους χρήστες του να κάνουν μια αναζήτηση στα διαθέσιμα αρχεία για να βρουν ένα συγκεκριμένο αρχείο τραγουδιού και τον οικοδεσπότη του. Η λειτουργία μεταφοράς αρχείων συντονίζεται άμεσα μεταξύ των ισότιμων οντοτήτων χωρίς να μεσολαβεί κάποιος άλλος κεντρικός υπολογιστής. Εκτός από την κύρια λειτουργία διανομής αρχείων, το Napster παρέχει μια λειτουργία συνομιλίας για να επιτρέψει στους χρήστες για να στέλνουν τα μηνύματα κειμένων ο ένας στον άλλο.

Παίρνοντας το σύνθημά του από Napster, αλλά επισημαίνοντας τις νομικές επιπτώσεις της παράβασης πνευματικών δικαιωμάτων, το πρόγραμμα Gnutella (www.gnutelliums.com) παίρνοντας την ιδέα διανομής αρχείων από Napster προχώρησε ένα βήμα παραπέρα και εξάλειψε την ανάγκη να παρέχεται η λειτουργία αναζήτησης σε έναν κεντρικό υπολογιστή. Η ανεξαρτησία κεντρικών υπολογιστών του δικτύου Gnutella, που συνδυάζεται με την ικανότητά της να μοιραστεί οποιοδήποτε τύπο αρχείου, το κάνει μια από τις ισχυρότερες επιδείξεις της P2P τεχνολογίας.

Οι οντότητες στο δίκτυο Gnutella είναι αρμόδιες όχι μόνο για την εξυπηρέτηση των αρχείων, αλλά και για την απάντηση στις ερωτήσεις και τα μηνύματα δρομολόγησης σε άλλες οντότητες. Σημειώστε ότι αν και η εφαρμογή Gnutella δεν απαιτεί έναν κεντρικό υπολογιστή για να παρέχει την αναζήτηση και τη λειτουργία διευκρίνισης των διευθύνσεων IP, η σύνδεση με το δίκτυο Gnutella ακόμα απαιτεί ότι μια οντότητα γνωρίζει τη διεύθυνση IP μιας ήδη συνδεδεμένης οντότητας με το P2P δίκτυο. Για αυτόν τον λόγο, διάφορες οντότητες με στατικές ή επιλύσιμες διευθύνσεις IP έχουν καθιερωθεί για να παρέχουν στις νέες οντότητες μια αφετηρία για την ανακάλυψη άλλων ισότιμων οντοτήτων στο δίκτυο. Η εξάλειψη της αναφοράς σε έναν κεντρικό υπολογιστή έχει φέρει στην επιφάνεια διάφορα νέα ζητήματα όπως:

- Πώς οι ομότιμες οντότητες διανέμουν τα μηνύματα χωρίς πλημμύρα του δικτύου;
- Πώς οι ομότιμες οντότητες παρέχουν το περιεχόμενο ασφαλώς και ανώνυμα;
- Πώς μπορεί το δίκτυο να ενθαρρύνει τη διανομή των πόρων;

Άλλες παραλλαγές P2P συστημάτων διανομής αρχείων, μαζί με τα Freenet (freenet.sourceforge.net), Morpheus (www.musiccity.com), και MojoNation (www.mojonation.net), έχουν έλθει στο χώρο να αντιμετωπίσουν αυτά τα ζητήματα. Κάθε μια από αυτές τις εφαρμογές αντιμετωπίζει ένα συγκεκριμένο ζήτημα. Το Freenet παρέχει την αποκεντρωμένη ανώνυμη αποθήκευση που προστατεύεται από το ισχυρό σύστημα κρυπτογραφία. Το Morpheus παρέχει τις βελτιωμένες ικανότητες αναζήτησης βασισμένες στα μεταδεδομένα που ενσωματώνονται με το κοινό σχήμα μέσων. Το MojoNation χρησιμοποιεί ένα τεχνητό νόμισμα, αποκαλούμενο Mojo, για να επιβάλει τη διανομή των πόρων.

2.3 Κατανεμημένος υπολογισμός

Ο κατανεμημένος υπολογισμός είναι ένας τρόπος να αντιμετωπιστούν τα δύσκολα προβλήματα με το διαχωρισμό του προβλήματος σε διάφορα υποπροβλήματα που μπορούν να λυθούν ανεξάρτητα, από έναν μεγάλο αριθμό υπολογιστών. Αν και οι δημοφιλέστερες εφαρμογές του κατανεμημένου

υπολογισμού δεν είναι P2P λύσεις, είναι σημαντικό να σημειωθεί η σημαντική εργασία που έχει ολοκληρωθεί από τα προγράμματα όπως SETI@Home (setiathome.berkeley.edu) και Distributed.net (distributed.net) και οι επιχειρήσεις όπως οι United Devices (www.ud.com).

Το 1996, ο οργανισμός SETI (Search for ExtraTerrestrial Intelligence) άρχισε να διανέμει στους χρήστες μια εφαρμογή σε στυλ προφύλαξης οθόνης, για να μπορέσει να επεξεργαστεί στοιχεία ραδιοτηλεσκοπίων και να συμβάλει στην αναζήτηση της εξωγήινης μορφής ζωής. Από τότε, έχει δεχθεί υποστήριξη από περισσότερους από 3 εκατομμύριο χρήστες (από τους οποίους περισσότερο από μισό εκατομμύριο είναι ενεργοί συνεισφέροντες). Σε παρόμοιο πρόγραμμα που άρχισε το 1997, Distributed.net χρησιμοποιώντας τη δύναμη υπολογισμού των χρηστών του προσπαθεί να «σπάσει» τα προηγουμένως άθραυστα κρυπτογραφημένα μηνύματα, από την αρχή δε, εξέφρασε την επιθυμία του, σε πρόκληση από τα RSA labs, να «σπάσει» το 56 bits κώδικα μυστικού κλειδιού.

Και στις δύο περιπτώσεις, το λογισμικό πελατών έρχεται σε επαφή με έναν κεντρικό υπολογιστή για να μεταφορτώσει την αντίστοιχη μερίδα του προς επίλυση προβλήματος και έως ότου να λυθεί το πρόβλημα, καμία περαιτέρω επικοινωνία με τον κεντρικό υπολογιστή δεν απαιτείται.

Στο μέλλον, αναμένεται ότι ο καταναμημένος υπολογισμός θα εξελιχθεί για να εκμεταλλευθεί πλήρως την τεχνολογία P2P δημιουργώντας μια αγορά για την εφεδρική δύναμη υπολογισμού.

3. Εισαγωγή της πλατφόρμας JXTA [JXTA]

Όπως φαίνεται από τα παραπάνω οι περισσότερες P2P λύσεις συμφωνούν με κάποια μορφή. Το ICQ παρέχει το στιγμιαίο μήνυμα συν «λίγο» από διανομή αρχείων. Το Napster παρέχει το διανομή αρχείων συν «λίγο» από στιγμιαίο μήνυμα. Ακόμη θα μπορούσαμε να πούμε ότι η Gnutella παρέχει τη διανομή αρχείων, συν λίγο από καταναμημένο υπολογισμό, λόγω του τρόπου που οντότητες δρομολογούν τα μηνύματα μέσω του δικτύου. Δυστυχώς, οι τρέχουσες εφαρμογές P2P τείνουν να χρησιμοποιήσουν τα πρωτόκολλα που είναι ιδιόκτητα και ασυμβίβαστης φύσης μεταξύ τους, πράγμα που μειώνει το πλεονέκτημα που προσφέρεται με τη συλλογή των συσκευών στα P2P δίκτυα. Κάθε δίκτυο διαμορφώνει μια κλειστή κοινότητα, απολύτως ανεξάρτητη από τα άλλα δίκτυα και ανίκανη αναμοχλευτούν οι υπηρεσίες τους.

Μέχρι τώρα, ο ενθουσιασμός της διερεύνησης των δυνατοτήτων P2P της τεχνολογίας έχει επισκιάσει τη σημασία της διαλειτουργικότητας και της επαναχρησιμοποίησης λογισμικού. Για να εξελίξουν την τεχνολογία P2P σε μια ώριμη πλατφόρμα λύσης, οι υπεύθυνοι για την ανάπτυξη πρέπει να επικεντρώσουν εκ νέου τις προσπάθειές τους από τον προγραμματισμό των βασικών δομών P2P δικτύων στη δημιουργία P2P εφαρμογών σε μια στερεά, καθορισμένη με σαφήνεια βάση. Για να κάνουν αυτό, χρειάζονται την ανάπτυξη μιας κοινής γλώσσας επικοινωνίας για να επιτρέψουν στις οντότητες που συμμετέχουν να επικοινωνούν και να εκτελέσουν τις βασικές αρχές της P2P δικτύωσης. Κατανοώντας αυτήν την ανάγκη για μια κοινή γλώσσα, η Sun Microsystems διαμόρφωσε το πρόγραμμα JXTA. Μια μικρή ομάδα ανάπτυξης κάτω από την καθοδήγηση των Bill Joy και Mike Clary, ορίστηκε για να

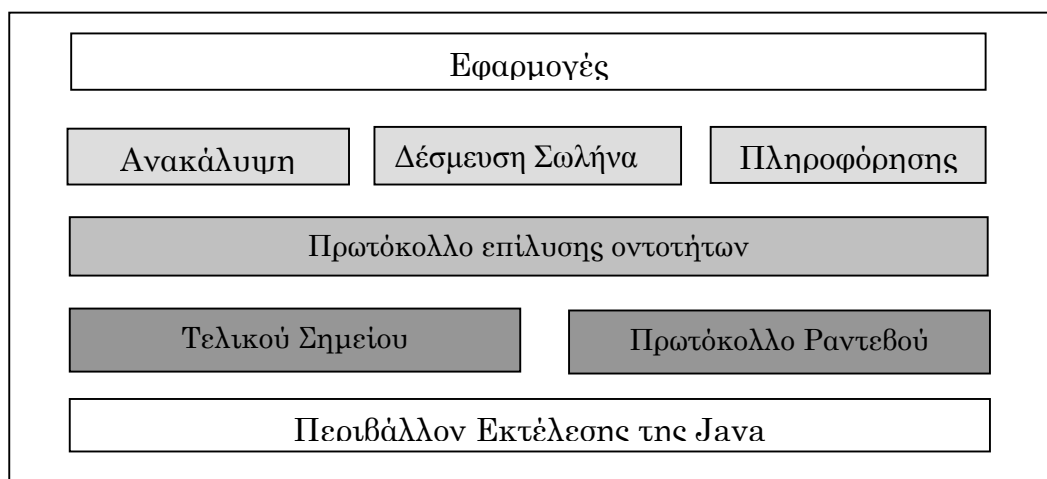
σχεδιάσει μια λύση να εξυπηρετήσει όλες τις P2P εφαρμογές. Στον πυρήνα του, το JXTA είναι απλά ένα σύνολο από προδιαγραφές πρωτοκόλλων, και αυτό είναι που το καθιστά τόσο ισχυρό. Έτσι κάποιος που θέλει να παραγάγει μια νέα P2P εφαρμογή δεν συναντά πλέον τη δυσκολία κατάλληλου σχεδιασμού πρωτοκόλλων για να χειριστεί τις βασικές λειτουργίες της P2P επικοινωνίας.

Το όνομα JXTA προέρχεται από την Αγγλική λέξη juxtapose που σημαίνει αντιπαραθέτω. Με την επιλογή αυτού του ονόματος, η ομάδα ανάπτυξης στη Sun αναγνώρισε ότι οι P2P λύσεις θα υπήρχαν πάντα παράλληλα με τις τρέχουσες λύσεις πελατών - κεντρικών υπολογιστών παρά να τις αντικαταστήσουν εντελώς.

Η πρώτη έκδοση προδιαγραφής πρωτοκόλλων JXTA καθορίζει τις βασικές δομικές μονάδες και τα πρωτόκολλα P2P της δικτύωσης:

- **πρωτόκολλο ανακάλυψης** - επιτρέπει σε δύο οντότητες να ανακαλύψουν άλλες ομότιμες οντότητες στο δίκτυο
- **πρωτόκολλο επίλυσης οντοτήτων** - επιτρέπει στις ομότιμες οντότητες να στέλνουν και να επεξεργαστούν γενικά αιτήματα
- **το πρωτόκολλο για ραντεβού** - χειρίζεται τις λεπτομέρειες της διάδοσης των μηνυμάτων μεταξύ των ομότιμων οντοτήτων
- **πρωτόκολλο πληροφόρησης** - παρέχει στις ομότιμες οντότητες έναν τρόπο να λάβουν πληροφορίες για άλλες οντότητες για το δίκτυο
- **πρωτόκολλο δέσμευσης σωλήνων** - παρέχει έναν μηχανισμό για τη δημιουργία ενός εικονικού καναλιού επικοινωνίας ανάμεσα σε ομότιμες οντότητες
- **το πρωτόκολλο δρομολόγησης τελικού σημείου** - παρέχει ένα σύνολο μηνυμάτων που χρησιμοποιούνται για να επιτρέψουν τη δρομολόγηση μηνυμάτων από μια οντότητα πηγής σε μια οντότητα προορισμού

Παρακάτω παρουσιάζεται η οργάνωση των πρωτοκόλλων αυτών, η μεταξύ τους σχέση και εξάρτηση στην πλατφόρμα JXTA.



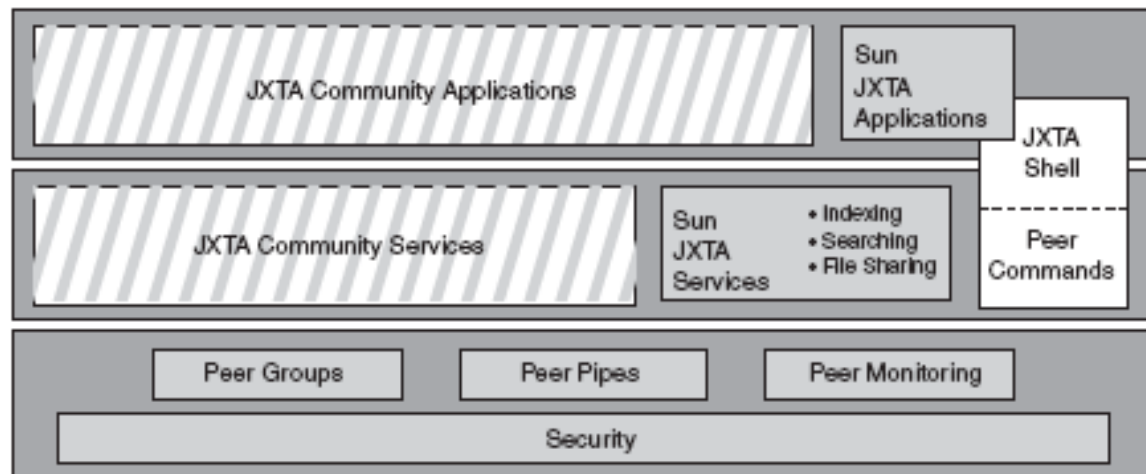
Σχήμα 3.1 η διαστρωμάτωση των πρωτοκόλλων της jxta

Τα πρωτόκολλα JXTA είναι ανεξάρτητα της γλώσσας προγραμματισμού και καθορίζουν ένα σύνολο μηνυμάτων XML για να συντονίσουν κάθε πτυχή της P2P δικτύωσης. Αν και μερικοί υπεύθυνοι για την ανάπτυξη στη P2P κοινότητα διαμαρτυρήθηκαν στη χρήση μιας τέτοιας «φλύαρης» γλώσσας, η

επιλογή της XML επιτρέπει στους εφαρμοστές των πρωτοκόλλων JXTA να χρησιμοποιήσουν τα ήδη υπάρχοντα εργαλεία για ανάλυση και μετασχηματισμό των XML μηνυμάτων. Επιπλέον, η απλότητα των πρωτοκόλλων JXTA το καθιστά πιθανό να εφαρμόσει P2P λύσεις σε οποιαδήποτε συσκευή με μία «ψηφιακή καρδιά», όπως τα κινητά τηλέφωνα ή PDA. Τα πρωτόκολλα αυτά είναι όλα όσα χρειάζεται μια οντότητα για να υπάρξει μέσα σε ένα κατακεντρωμένο περιβάλλον ομότιμων οντοτήτων.

Για να λειτουργήσουν τα πρωτόκολλα αυτά και να δημιουργήσουν ένα ολοκληρωμένο αποκεντρωμένο σύστημα πρέπει να υποβληθεί μια αρχιτεκτονική. Η JXTA ακολουθεί την αρχιτεκτονική του σχήματος 3.2.

Το στρώμα πλατφόρμας, γνωστό και ως πυρήνας JXTA, τοποθετεί τις ελάχιστες και ουσιαστικά πρωταρχικές δομές που είναι κοινές για τη P2P δικτύωση. Περιλαμβάνει τις δομικές μονάδες για να επιτρέψει τους βασικούς μηχανισμούς για τις P2P εφαρμογές, συμπεριλαμβανομένης της ανακάλυψης, της μεταφοράς (συμπεριλαμβανομένης του χειρισμού της ζώνης ηλεκτρονικής προστασίας), της δημιουργίας των ομότιμων οντοτήτων και των ομάδων, και τη σχετική πρωταρχική ασφάλεια.



Σχήμα 3.2 Η αρχιτεκτονική της jxta.

Το στρώμα υπηρεσιών περιλαμβάνει τις υπηρεσίες δικτύων που μπορούν να μην είναι απολύτως απαραίτητες για ένα P2P δίκτυο να λειτουργήσει, αλλά είναι κοινό ή επιθυμητό στο P2P περιβάλλον. Παραδείγματα τέτοιων υπηρεσιών δικτύων περιλαμβάνουν την έρευνα και την ευρετηρίαση, δημιουργία καταλόγου, συστήματα αποθήκευσης, το μοίρασμα αρχείων, τα κατακεντρωμένα συστήματα αρχείων, τη συνάθροιση και την «ενοικίαση» των πόρων, τη μετάφραση πρωτοκόλλων, την αυθεντικότητα, και υπηρεσίες κρυπτογράφησης δημοσίου κλειδιού (PKI).

Το στρώμα εφαρμογών περιλαμβάνει την ενσωμάτωση ολοκληρωμένων εφαρμογών, όπως τη μεταφορά στιγμιαίων μηνυμάτων, τη διανομή εγγράφων και των πόρων, τη διαχείριση και παράδοση περιεχόμενου ψυχαγωγίας, συστήματα ηλεκτρονικού ταχυδρομείου, τα κατακεντρωμένα συστήματα δημοπρασίας, και πολλά άλλα.

Τελικά τον Απρίλιο του 2001, η ομάδα ανάπτυξης, τοποθέτησε το πρόγραμμα JXTA στα χέρια της κοινότητας για την παραπέρα ανάπτυξη με την

υιοθέτηση μιας άδειας βασισμένης στην έκδοση αδειών λογισμικού Apache. Εκτός από τη διατήρηση της v1.0 προδιαγραφής πρωτοκόλλων JXTA, το πρόγραμμα JXTA είναι αρμόδιο για την ανάπτυξη των εφαρμογών αναφοράς στην πλατφόρμα JXTA και του ελέγχου πηγαίου κώδικα για ποικίλα κοινοτικά προγράμματα JXTA. Αυτήν την περίοδο, το πρόγραμμα JXTA έχει μια εφαρμογή αναφοράς διαθέσιμη σε Java, σε C, σε Objective-C, Ruby, και Perl. Αυτή τη στιγμή, το πρόγραμμα JXTA στεγάζει ποικίλα κοινοτικά προγράμματα που εφαρμόζουν την τεχνολογία JXTA στους διαφορετικούς τομείς όπως διαχείριση περιεχομένου, τεχνητή νοημοσύνη, και εξασφάλιση ανώνυμων συστημάτων πλήρωμης.

Στο παράρτημα καταγράφονται όλα τα τρέχοντα προγράμματα της κοινότητας που αναπτύσσονται πάνω στην πλατφόρμα JXTA μαζί με μια σχετική περιγραφή τους.

3.1 Επισκόπηση της πλατφόρμας JXTA

Παρακάτω προσπαθούμε να παρουσιάσουμε τις βασικές έννοιες που συναντάμε στην πλατφόρμα JXTA και να τις τοποθετήσουμε σε ένα γενικό πλαίσιο κοινό για όλα τα P2P δίκτυα. Αυτό το κεφάλαιο εισάγει την ορολογία που χρησιμοποιείται για να περιγράψει τις πτυχές P2P των δικτύων, των συστατικών κοινών για όλες P2P τις λύσεις (συμπεριλαμβανομένων των εκείνων των που δεν χτίζονται χρησιμοποιώντας την τεχνολογία JXTA), και των προβλημάτων και των λύσεων έμφυτων P2P στα δίκτυα.

Η τεχνολογία P2P είναι η λύση στην απλή ερώτηση: Πώς μπορούμε να συνδέσουμε ένα σύνολο συσκευών κατά τέτοιο τρόπο ώστε μπορούν να μοιραστούν τις πληροφορίες, τους πόρους, και τις υπηρεσίες; Στην επιφάνεια, φαίνεται μια απλή ερώτηση, αλλά για να απαντήσει απαιτεί κατάλληλα σε διάφορες υπονοούμενες ερωτήσεις:

- Τι σημαίνει συσκευή;
- Πώς μια «συσκευή» μαθαίνει της παρουσίας μιας άλλης «συσκευής»;
- Πώς μια «συσκευή» οργανώνεται να διευθετήσει κοινές ερωτήσεις;
- Πώς μια συσκευή γνωστοποιεί τις ικανότητές της;
- Ποιες πληροφορίες απαιτούνται για να προσδιορίσουν μοναδικά μια συσκευή;
- Πώς οι συσκευές ανταλλάσσουν τα στοιχεία;

Όλα τα P2P δίκτυα στηρίζονται στα θεμελιώδη στοιχεία για να δώσουν τις απαντήσεις σε τέτοιες ερωτήσεις. Δυστυχώς, πολλά από αυτά τα στοιχεία υποθέτονται ή υπονοούνται από τα ιδιόκτητα P2P δίκτυα και είναι αυστηρώς κωδικοποιημένα στις εφαρμογές πολλών P2P εφαρμογών, με συνέπεια την «ακαμψία» αυτών των λύσεων. Η πλειοψηφία των τρεχουσών P2P λύσεων, παραδείγματος χάριν, υποθέτει τη χρήση του TCP ως μηχανισμό μεταφορών δικτύων και δεν μπορεί να λειτουργήσει σε οποιοδήποτε άλλο περιβάλλον δικτύων. Οι εύκαμπτες P2P λύσεις χρειάζονται μια γλώσσα που δηλώνει ρητά όλες τις μεταβλητές σε οποιαδήποτε P2P λύση.

Παρακάτω αναφέρεται η βασική ορολογία P2P της δικτύωσης όπως αυτή συναντάται στην πλατφόρμα JXTA.

3.2 Ομότιμες οντότητες (peers)

Μια ομότιμη οντότητα (peer) είναι ένας κόμβος σε ένα P2P δίκτυο που διαμορφώνει τη θεμελιώδη μονάδα επεξεργασίας σε οποιασδήποτε P2P λύσης. Μέχρι τώρα, η εικόνα που ίσως έχει σχηματιστεί για μια τέτοια μονάδα, είναι για κάποια εφαρμογή που τρέχει σε έναν υπολογιστή που συνδέεται με ένα δίκτυο όπως στο Διαδίκτυο, αλλά αυτός ο περιορισμένος ορισμός δεν θα συλλάμβανε τον αληθινό ρόλο μιας οντότητας και όλων των πιθανών «ενσάρκώσεων» της. Ένας τέτοιος ορισμός απορρίπτει τη δυνατότητα μια οντότητα να είναι μια εφαρμογή που κατανεμήθηκε διάφορες μηχανές ή ότι είναι μια μικρότερη συσκευή, όπως ένας υπολογιστής χειρός ή ένα κινητό τηλέφωνο, το οποίο συνδέεται με ένα δίκτυο έμμεσα, όπως μέσω Bluetooth με έναν υπολογιστή ήδη συνδεδεμένο. Ένας υπολογιστής μπορεί ακόμη να είναι αρμόδιος για το τρέξιμο των πολλαπλών τέτοιων οντοτήτων. Για να καλύψουμε όλες αυτές τις απόψεις η έννοια της ομότιμης οντότητας ορίζεται ως «Οποιαδήποτε οντότητα ικανή να παράγει κάποια χρήσιμη εργασία και να παρέχει τα αποτελέσματα εκείνης της εργασίας σε μια άλλη οντότητα πέρα από ένα δίκτυο, είτε άμεσα είτε έμμεσα». Ο καθορισμός της χρήσιμης εργασίας εξαρτάται από τον τύπο οντότητας. Τρεις πιθανοί τύποι ισοτίμων οντοτήτων υπάρχουν σε οποιοδήποτε P2P δίκτυο:

- Απλές οντότητες: με σκοπό να εξυπηρετήσουν έναν τελικό χρήστη, επιτρέποντάς του να παρέχει υπηρεσίες από τη συσκευή του και να καταναλώνει υπηρεσίες που παρέχονται από άλλες οντότητες στο δίκτυο.
- Οντότητες για ραντεβού: παρέχουν στις άλλες οντότητες μια θέση στο δίκτυο για να ανακαλύψουν άλλες ομότιμες οντότητες ή και πόρους. Με την αποθήκευση πληροφοριών για τις ισοτίμες οντότητες, βελτιώνουν την απόκριση σε ερωτήματα, συμβάλουν στη μείωση της κυκλοφορίας και παρέχουν καλύτερη υπηρεσία στις μελλοντικές αιτήσεις από τις άλλες απλές οντότητες. Συνήθως τέτοιες οντότητες συναντιούνται έξω από την ζώνη προστασίας ενός ιδιωτικού δικτύου για να συγκερατούν πληροφορίες σχετικά με τις οντότητες στο εσωτερικό του ιδιωτικού δικτύου και πληροφορίες για το πώς να προσεγγιστούν από άλλες οντότητες από το εξωτερικό δίκτυο.
- Οντότητες δρομολογητών: παρέχουν πληροφορίες για τη δρομολόγηση της επικοινωνίας μεταξύ δύο ομότιμων οντοτήτων. Αυτές οι πληροφορίες δρομολόγησης παρέχουν έναν μηχανισμό στο P2P δίκτυο για να αντικαταστήσει το παραδοσιακό DNS, επιτρέποντας σε μια περιοδικά συνδεδεμένη συσκευή με μια δυναμική διεύθυνση IP για να βρεθεί στο δίκτυο όπου κι αν αυτή βρίσκεται. Συνήθως τέτοιου είδους οντότητες συναντούνται έξω από μια ζώνη προστασίας ή μια συσκευή μετάφρασης διεύθυνσης δικτύου (NAT). Με αυτόν το τρόπο παρέχουν ένα μηχανισμό επικοινωνίας με τις ομότιμες οντότητες που βρίσκονται έξω από το ιδιωτικό αυτό δίκτυο να ανακαλύψουν και να επικοινωνήσουν με αυτές που βρίσκονται μέσα, δρομολογώντας κατάλληλα τα μηνύματα.

Κάθε ομότιμη οντότητα στο δίκτυο μπορεί να ενεργήσει ως ένας ή ως περισσότεροι τύποι, με κάθε τύπο καθορίζοντας ένα διαφορετικό σύνολο ευθυνών για την οντότητα στο P2P δίκτυο συνολικά.

3.3 Ομάδες ομότιμων οντοτήτων

Πριν από τη JXTA, η ιδιόκτητη και εξειδικευμένη φύση P2P των λύσεων και των σχετικών πρωτοκόλλων τους διαίρεσε τη χρήση του «χώρου» των δικτύων σύμφωνα με την εφαρμογή. Εάν θελήσατε να εκτελέσετε τη διανομή αρχείων, χρησιμοποιήσατε πιθανώς το πρωτόκολλο Gnutella και θα μπορούσατε να επικοινωνήσετε μόνο με άλλες οντότητες χρησιμοποιώντας το πρωτόκολλο Gnutella ομοίως, εάν θελήσατε να στείλετε ένα στιγμιαίο μήνυμα, χρησιμοποιήσατε το ICQ και θα μπορούσατε να επικοινωνήσετε μόνο με άλλες οντότητες που χρησιμοποιούν επίσης το ICQ. Τα ασυμβίβαστα των πρωτοκόλλων διαίρεσαν αποτελεσματικά το «διάστημα» μεταξύ των δικτύων βασισμένο στην εφαρμογή που χρησιμοποιείται από τις σχετικές οντότητες. Εάν εξετάζετε ένα P2P σύστημα στο οποίο όλοι οι πελάτες μπορούν να μιλήσουν το ίδιο σύνολο πρωτοκόλλων, όπως μπορούν σε JXTA, η έννοια μιας ομάδας ομότιμων οντοτήτων είναι απαραίτητη για να υποδιαιρέσει το χώρο των δικτύων. Έτσι, μια *ομάδα ομότιμων οντοτήτων* καθορίζεται ως εξής: «ένα σύνολο ομότιμων οντοτήτων που διαμορφώνεται για να εξυπηρετήσει ένα κοινό συμφέρον ή έναν στόχο που υπαγορεύθηκε από τις σχετικές οντότητες που συμμετέχουν». Οι ομάδες αυτές μπορούν να παρέχουν τις υπηρεσίες στις οντότητες μέλη τους οι οποίοι δεν είναι προσιτοί από άλλες οντότητες στο P2P δίκτυο. Οι ομάδες των ομότιμων οντοτήτων χωρίζουν το P2P δίκτυο σε ομάδες με κοινούς στόχους, βασισμένους στα εξής στοιχεία:

- **Την εφαρμογή στην οποία θέλουν να συνεργαστούν επάνω ως ομάδα.** Μια ομάδα διαμορφώνεται ώστε τα μέλη της να ανταλλάξουν υπηρεσίες, οι οποίες όμως να μην είναι διαθέσιμες σε ολόκληρο το P2P δίκτυο. Ένας λόγος για αυτό θα μπορούσε να είναι η ιδιωτική φύση των στοιχείων που χρησιμοποιούνται από την εφαρμογή.
- **Οι απαιτήσεις ασφάλειας των σχετικών οντοτήτων.** Μια ομάδα μπορεί να χρησιμοποιεί υπηρεσίες επικύρωσης αυθεντικότητας για να περιορίσουν τον αριθμό των οντοτήτων που μπορούν να προσχωρήσουν στην ομάδα και να έχουν πρόσβαση στις υπηρεσίες που προσφέρονται από την ομάδα.
- **Την ανάγκη για πληροφορίες κατάστασης για τα μέλη της ομάδας.** Τα μέλη μιας ομάδας ομότιμων οντοτήτων μπορούν να ελέγχουν άλλα μέλη. Οι πληροφορίες κατάστασης μπορεί να χρησιμοποιούνται για να διατηρήσουν ένα κατώτατο επίπεδο υπηρεσίας για την εφαρμογή της ομάδας.

Τα μέλη της ομάδας μπορούν να παρέχουν πλεονάζουσα πρόσβαση σε μια υπηρεσία, που εξασφαλίζει ότι μια υπηρεσία είναι πάντα διαθέσιμη σε μια ομάδα ομότιμων οντοτήτων εφ' όσον τουλάχιστον ένα μέλος της παρέχει την υπηρεσία.

3.4 Μεταφορά στο δίκτυο

Για να ανταλλάξουν πληροφορίες, οι ομότιμες οντότητες πρέπει να χρησιμοποιήσουν κάποιο μηχανισμό για να χειριστούν τη μετακίνηση δεδομένων στο δίκτυο. Αυτό το στρώμα, αποκαλούμενο στρώμα *μεταφοράς δικτύου*, είναι αρμόδιο για όλες τις πτυχές της μετάδοσης δεδομένων, συμπεριλαμβανομένου του τεμαχισμού αυτών σε πακέτα, προσθέτοντας τις

κατάλληλες επιγραφές σε ένα πακέτο για να ελέγξει τον προορισμό του, και σε μερικές περιπτώσεις, εξασφαλίζει ότι ένα πακέτο φθάνει στον προορισμό του. Μια μεταφορά δικτύων θα μπορούσε να είναι μια χαμηλού επιπέδου μεταφορά, όπως UDP ή το TCP, ή μια υψηλού επιπέδου μεταφορά, όπως το HTTP ή SMTP. Η έννοια της μεταφοράς στα δίκτυα P2P μπορεί να αναλυθεί σε τρία βασικά μέρη:

- **Τελικά σημεία πρόσβασης (endpoints):** η αρχική πηγή ή ο τελικός προορισμός οποιουδήποτε κομματιού του διαβίβασης των στοιχείων πέρα από το δίκτυο. Ένα σημείο τέλους αντιστοιχεί στις διεπαφές δικτύων που χρησιμοποιούνται για να στείλουν και να λάβουν τα στοιχεία.
- **Σωλήνες (pipes):** κατευθυντικά, ασύγχρονα, εικονικά κανάλια επικοινωνιών που συνδέουν δύο ή περισσότερα τελικά σημεία.
- **Μηνύματα (messages):** τα στοιχεία που διαβιβάζονται πέρα από έναν σωλήνα από ένα σημείο τέλους σε άλλο.

Για να μπορέσει μια ομότιμη οντότητα να επικοινωνήσει χρησιμοποιώντας ένα σωλήνα, πρέπει πρώτα να βρει τα τελικά σημεία, ένα για την πηγή του μηνύματος και ένα για κάθε προορισμό του μηνύματος, και έπειτα να τα συνδέσει με τη δέσμευση ενός σωλήνα σε κάθε ένα από τα τελικά σημεία. Με αυτό τον τρόπο, το τελικό σημείο πηγής δεδομένων καλείται *σωλήνας παραγωγής ή εξόδου* και το τελικό σημείο δέκτη καλείται *σωλήνας εισαγωγής ή εισόδου*. Ο ίδιος ο σωλήνας δεν είναι πραγματικά αρμόδιος για να φέρει τα στοιχεία μεταξύ των σημείων τέλους, αποτελεί μόνο μια αφαίρεση που χρησιμοποιείται για να αντιπροσωπεύσει το γεγονός ότι δύο τελικά σημεία συνδέονται. Τα τελικά σημεία πρόσβασης είναι αυτά που παρέχουν πρόσβαση στην διεπαφή των δικτύων για τη διαβίβαση και τη λήψη των δεδομένων. Για να σταλθούν δεδομένα από μια ομότιμη οντότητα σε μία άλλη, η πρώτη συσκευάζει τα δεδομένα που διαβιβάζονται σε ένα μήνυμα και στέλνει το μήνυμα χρησιμοποιώντας έναν σωλήνα παραγωγής στην άλλη άκρη του σωλήνα, η άλλη ομότιμη οντότητα λαμβάνει τα μηνύματα από έναν σωλήνα εισαγωγής και εξάγει τα διαβιβασθέντα δεδομένα.

Σημειώνουμε ότι ένας σωλήνας παρέχει επικοινωνία μόνο προς μια κατεύθυνση, απαιτώντας κατά συνέπεια δύο σωλήνες για να επιτύχει τη διπλής κατεύθυνσης επικοινωνία μεταξύ δύο ομότιμων οντοτήτων.

3.5 Υπηρεσίες

Οι υπηρεσίες παρέχουν τις λειτουργίες που οι ομότιμες οντότητες μπορούν να δεσμεύσουν για να εκτελέσουν τη «χρήσιμη εργασία» για μία απομακρυσμένη ομότιμη οντότητα. Αυτή η εργασία μπορεί να περιλάβει τη μεταφορά ενός αρχείου, την παροχή των πληροφοριών κατάστασης, την εκτέλεση ενός υπολογισμού, ή βασικά να κάνει οτιδήποτε μπορεί να θελήσουμε να κάνει μια οντότητα σε ένα P2P δίκτυο, για την οποίο είναι ικανό να κάνει. Οι υπηρεσίες είναι το κίνητρο για τη συλλογή των συσκευών σε ένα P2P δίκτυο. Χωρίς υπηρεσίες δεν έχουμε ένα P2P δίκτυο αλλά απλώς ένα σύνολο συσκευών ανίκανων να συμμετάσχουν σε οποιαδήποτε συναλλαγή πόρων. Οι υπηρεσίες μπορούν να διαιρεθούν σε δύο κατηγορίες:

- **Υπηρεσίες ομότιμων οντοτήτων:** προσφέρονται από μια συγκεκριμένη ομότιμη οντότητα στο δίκτυο σε άλλες ομότιμες οντότητες. Οι ικανότητες αυτής της υπηρεσίας θα είναι μοναδικές στην ομότιμη οντότητα και θα είναι διαθέσιμες μόνο όταν αυτός συνδέεται με το δίκτυο. Όταν αποσυνδέεται από το δίκτυο, η υπηρεσία δεν θα είναι πλέον διαθέσιμη.
- **Υπηρεσίες ομάδας ομότιμων οντοτήτων:** προσφέρονται από μια ομάδα ομότιμων οντοτήτων στα μέλη της ομάδας. Διάφορα μέλη της ομάδας, κατ' αυτόν τον τρόπο, παρέχοντας περιττή πρόσβαση στην υπηρεσία, θα μπορούσαν να παρέχουν αυτήν την λειτουργία. Εφ' όσον τουλάχιστον ένα μέλος της ομάδας συνδέεται με το δίκτυο, η ομάδα είναι σε θέση να παρέχει την υπηρεσία στα μέλη της.

Το όριο μεταξύ των υπηρεσιών και των εφαρμογών δεν είναι άκαμπτο παρόλο που στην αρχιτεκτονική της JXTA φαίνεται να υπάρχει διαχωρισμός. Μια ολοκληρωμένη εφαρμογή σε έναν πελάτη, μπορεί να αντιμετωπισθεί ως υπηρεσία από έναν άλλο πελάτη. Ολόκληρο το σύστημα σχεδιάζεται για να είναι κλιμακωτό, επιτρέποντας στους υπεύθυνους για την ανάπτυξη για να επιλέξει μια συλλογή των υπηρεσιών και των εφαρμογών που ανταποκρίνεται στις ανάγκες τους.

3.6 Διαφημίσεις

Η έννοια της *διαφήμισης* χρησιμοποιείται όπως ακριβώς γνωρίζουμε από την καθημερινή μας ζωή. Ομότιμες οντότητες προβάλλουν την ύπαρξή τους, τις υπηρεσίες τους ή τους πόρους τους σε μια ομάδα, ή ακόμη την ομάδα σε ολόκληρο το P2P δίκτυο μέσω διαφημίσεων. Μέχρι τώρα, οι εφαρμογές P2P έχουν χρησιμοποιήσει μια άτυπη μορφή διαφημίσεων. Στην εφαρμογή Gnutella λ.χ., τα αποτελέσματα που επιστρέφονται από μια ερώτηση αναζήτησης θα μπορούσαν να θεωρηθούν διαφήμιση που διευκρινίζει τη θέση ενός συγκεκριμένου αρχείου – πόρου στο δίκτυο Gnutella. Αυτές οι πρωτόγονες διαφημίσεις είναι εξαιρετικά περιορισμένες στο σκοπό και την αναφορά τους.

Στον πυρήνα της, η *διαφήμιση* αποτελεί τη δομημένη αντιπροσώπευση μιας οντότητας, μιας υπηρεσίας, ή ενός πόρου που παρέχεται από μια ομότιμη οντότητα ή μια ομάδα ως μέρος ενός P2P δικτύου. Όλες οι δομικές μονάδες που συναντούνται στη JXTA μπορούν να περιγραφούν από τις διαφημίσεις, συμπεριλαμβανομένων των ομότιμων οντοτήτων, τις ομάδες, τους σωλήνες, τα σημεία τέλους, τις υπηρεσίες, και το περιεχόμενο.

3.7 Πρωτόκολλα

Κάθε ανταλλαγή στοιχείων στηρίζεται σε ένα πρωτόκολλο για να οριστεί ρητά ποιο στοιχείο στέλνεται και σε ποια σειρά. Ακόμη και η απλούστερη ανθρώπινη χειρονομία, η χειραψία, στηρίζεται σε ένα άτυπο πρωτόκολλο που καθορίζει πότε είναι αρμόζον να σφίξουμε τα χέρια, ποιο χέρι να χρησιμοποιήσουμε και για πόσο χρόνο. Ένα *πρωτόκολλο* είναι ένας καθορισμένος τρόπος την ανταλλαγή των πληροφοριών μεταξύ δύο ή

περισσότερων συμβαλλόμενων μερών που χρησιμοποιούν κανόνες που έχουν συμφωνηθεί προηγουμένως σχετικά με από όλα τα συμβαλλόμενα μέρη. Σε ένα P2P δίκτυο, τα πρωτόκολλα χρειάζονται για να παρέχουν έναν τρόπο αλληλεπίδρασης με μια οντότητα.

Με τη δεδομένη οργάνωση των πληροφοριών σε διαφημίσεις απλοποιούνται σημαντικά τα πρωτόκολλα που απαιτούνται για τις διάφορες λειτουργίες. Οι διαφημίσεις οι ίδιες υπαγορεύουν τη δομή και την αντιπροσώπευση των στοιχείων, απλοποιώντας έτσι τον καθορισμό ενός πρωτοκόλλου. Έτσι αντί για τη μετακίνηση δεδομένων σε ένα πρωτόκολλο επικοινωνίας, τα πρωτόκολλα οργανώνουν απλά την ανταλλαγή διαφημίσεων που περιέχουν τις απαραίτητες πληροφορίες για να εκτελέσουν κάποια αυθαίρετη λειτουργία.

3.8 Ονομασία οντοτήτων

Τα περισσότερα στοιχεία σε ένα P2P δίκτυο χρειάζονται πληροφορίες που τα προσδιορίζει μεμονωμένα στο δίκτυο:

- **Ομότιμη οντότητα:** χρειάζεται ένα μοναδικό στοιχείο που να την προσδιορίζει, ώστε οι άλλες ομότιμες οντότητες να το χρησιμοποιήσουν για να την εντοπίσουν ή να την ξεχωρίσουν στο δίκτυο. Ο προσδιορισμός μιας ομότιμης οντότητας είναι απαραίτητος ώστε το κάθε μήνυμα που αποστέλλεται να ληφθεί τελικά από το σωστό παραλήπτη.
- **ομάδα ομότιμων οντοτήτων:** μια ομότιμη οντότητα χρειάζεται κάποιο τρόπο ώστε να προσδιορίσει ποια ομάδα θα επιθυμούσε να χρησιμοποιήσει για να γίνει μέλος.
- **σωλήνες:** για να επιτρέψουν την επικοινωνία, μια ομότιμη οντότητα χρειάζεται κάποιο τρόπο να προσδιορίσει ένα σωλήνα που συνδέει κάποια σημεία τέλους στο δίκτυο.
- **περιεχόμενο:** ένα τμήμα περιεχομένου, π.χ. ένα αρχείο ή μια υπηρεσία, πρέπει να έχει μοναδικό προσδιορισμό για να επιτρέψει στις άλλες ομότιμες οντότητες να χρησιμοποιήσουν την τεχνική αντανάκλασης περιεχομένου (mirror) στο δίκτυο, με αυτόν τον τρόπο παρέχοντας την περιττή πρόσβαση στο περιεχόμενο. Οι άλλες ομότιμες οντότητες μπορούν έπειτα να χρησιμοποιήσουν αυτό το μοναδικό προσδιοριστικό παράγοντα για να βρουν το περιεχόμενο σε οποιοδήποτε ομότιμη οντότητα που το κατέχει.

Στα παραδοσιακά P2P δίκτυα, μερικά από αυτά τα προσδιοριστικά έχουν χρησιμοποιήσει τις, ορισμένες από τη μεταφορά, λεπτομέρειες δικτύων π.χ. μια ομότιμη οντότητα θα μπορούσε να προσδιοριστεί από τη διεύθυνση IP της. Εντούτοις, η εξάρτηση από το σύστημα είναι άκαμπτη τεχνική και δεν μπορεί να παρέχει ένα σύστημα του προσδιορισμού που είναι ανεξάρτητο από τη μεταφορά, λειτουργικό σύστημα ή τα πρωτόκολλα δικτύων. Στο ιδανικό P2P δίκτυο, οποιαδήποτε συσκευή πρέπει να είναι ικανή, ανεξάρτητα από τη μεταφορά της λειτουργικών συστημάτων ή δικτύων.

Ένας, ανεξάρτητο από το σύστημα, τρόπος ονομασίας των ομότιμων οντοτήτων είναι μια απαίτηση για ένα εύκαμπτο P2P δίκτυο. Έτσι στην

πλατφόρμα JXTA υιοθετείται ένα τέτοιο σχήμα για τον προσδιορισμό των παραπάνω στοιχείων. Πρόκειται για μια συμβολοσειρά που, όπως ο αριθμός ταυτότητας, προσδιορίζει μονοσήμαντα το κάθε ομότιμη οντότητα, ομάδα, υπηρεσία, σωλήνα ή περιεχόμενο. Ο τρόπος σχηματισμού αυτών των αριθμών γίνεται με μια ψευδοτυχαία ακολουθία γραμμάτων και αριθμών έτσι ώστε η πιθανότητα να υπάρχουν δύο ίδιοι αριθμοί να είναι πολύ μικρή έως μηδαμινή. Στο παρακάτω σχήμα παρουσιάζεται ο γενικός τρόπος, κατά ABNF [RFC2234], με τον οποίο γίνεται αυτό.

<JXTAURN>	::= "urn:" <JXTANS> ":" <JXTAIDVAL>
<JXTANS>	::= "jxta"
<JXTAIDVAL>	::= <JXTAFMT> "-" <JXTAIDUNIQ>
<JXTAFMT>	::= 1 * <URN chars>
<JXTAIDUNIQ>	::= 1 * <URN chars>
<URN chars>	::= <trans> "%" <hex> <hex>
<trans>	::= <upper> <lower> <number> <other> <reserved>
<upper>	::= "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
<lower>	::= "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
<hex>	::= <number> "A" "B" "C" "D" "E" "F" "a" "b" "c" "d" "e" "f"
<number>	::= "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
<other>	::= "(" ")" "+" "," "-" "." ":" "=" "@" ";" "\$" "_" "!" "*" ""
<reserved>	::= "%" "/" "?" "#"

Σχήμα 3.3 Παρουσίαση γενικού τρόπου σχηματισμού JXTA id [RFC2141] κατά ABNF

Ενώ η Java υλοποίηση της προδιαγραφής υιοθετεί σχήμα 3.4.

<JXTAUUIDURN>	::= "urn:" <JXTANS> ":" <JXTAUUIDFMT> "-" <(1*(<hex> <hex>)) <JXTAUUIDIDTYPE>
<JXTAUUIDFMT>	::= "uuid"
<JXTAUUIDIDTYPE>	::= <CODATID> <PEERGROUPID> <PEERID> <PIPEID> <MODULECLASSID> <MODULESPECID>
<CODATID>	::= "01"
<PEERGROUPID>	::= "02"
<PEERID>	::= "03"
<PIPEID>	::= "04"
<MODULECLASSID>	::= "05"
<MODULESPECID>	::= "06"

Σχήμα 3.4 Ειδικά jxta id για την υλοποίηση κατά java κατά ABNF

Με αυτό τον τρόπο υφίσταται η μοναδικότητα ότι μια ομότιμη οντότητα έχει διαφορετικό αριθμό ταυτότητας (peerid) απ' ότι ένα περιεχόμενο (codatid) ή μια ομάδα (peergroup) ή ένας σωλήνας (pipeid).

4. Πρωτόκολλα της JXTA

Τα πρωτόκολλα της jxta αποτελούν τη βάση για να χτίσουμε πάνω τους όλες της υπηρεσίες και τις εφαρμογές μας. Η διεπαφή με το προγραμματιστικό περιβάλλον έχει υλοποιημένα τα πρωτόκολλα ως υπηρεσίες και έτσι παρέχονται στους υπεύθυνους ανάπτυξης εφαρμογών. Παρακάτω αναλύουμε τη λειτουργία των πρωτοκόλλων.

4.1 Πρωτόκολλο τελικού σημείου

Το πρωτόκολλο τελικού σημείου είναι αρμόδιο για τον καθορισμό μιας διαδρομής μεταξύ ομότιμων οντοτήτων στο δίκτυο JXTA. Η υπηρεσία τελικού σημείου είναι υπηρεσία πυρήνα στην εφαρμογή αναφοράς της Java που είναι αρμόδια για την εφαρμογή του πρωτοκόλλου τελικού σημείου. Το πρωτόκολλο τελικού σημείου έχει ως σκοπό να παρέχει το κανάλι επικοινωνίας μεταξύ μιας ομότιμης οντότητας και ενδεχομένως διάφορων άλλων. Τοποθετημένο στο κατώτατο σημείο της διαστρωμάτωσης των πρωτοκόλλων, το πρωτόκολλο τελικού σημείου χειρίζεται τις χαμηλού επιπέδου λεπτομέρειες της επικοινωνίας.

4.1.1 Επισκόπηση του πρωτοκόλλου

Το πρωτόκολλο αυτό είναι αρμόδιο για την παροχή μιας πρόσοψης, που κάνει δύο ομότιμες οντότητες οι οποίες δεν έχουν κάποια άμεση σύνδεση μεταξύ τους να φανεί ότι έχουν. Αυτό κάνει το δίκτυο JXTA να εμφανιστεί να είναι μια τοπολογία δικτύων πολλαπλών σημείων. Έχοντας ένα ξεχωριστό πρωτόκολλο για αυτή την εικονική σύνδεση σημαίνει ότι η υπηρεσία τελικού σημείου δεν χρειάζεται να ξέρει εάν δύο οντότητες συνδέονται άμεσα.

Για να ολοκληρώσει τη διαδικασία σύνδεσης, το πρωτόκολλο καθορίζει διάφορα μηνύματα, που αποτελούνται από ερωτήσεις και απαντήσεις (με τον ίδιο σχεδόν τρόπο όπως σε πολλά από τα άλλα πρωτόκολλα). Τα μηνύματα ερώτησης και απάντησης είναι τα ακόλουθα:

- **Ερώτηση ping:** η υπηρεσία τελικού σημείου στέλνει αυτό το μήνυμα για να καθορίσει εάν υπάρχει μια διαδρομή μεταξύ ομότιμων. Στις περισσότερες καταστάσεις, η σύνδεση θα είναι άμεση και δεν θα περιλάβει τη διάδοση. Όταν μια ομότιμη οντότητα λαμβάνει ένα μήνυμα με ερώτηση ping, αποκρίνεται σε εκείνο το μήνυμα.
- **Απάντηση ping:** όταν μια ομότιμη οντότητα λαμβάνει μια ερώτηση ping, αυτό στέλνει ένα μήνυμα απάντησης ping. Το μήνυμα θα ενημερώσει την ομότιμη οντότητα πηγής ότι μια διαδρομή είναι διαθέσιμη μεταξύ δύο.

- **Ερώτηση διαδρομής:** εάν μια ομότιμη οντότητα πρέπει να στείλει ένα μήνυμα σε μια άλλη, αλλά δεν υπάρχει μια άμεση διαδρομή, τότε στέλνει ένα μήνυμα ερώτησης διαδρομής στην ομότιμη οντότητα που είναι άμεσα συνδεδεμένη με αυτήν ζητώντας για μια διαδρομή προς τον προορισμό.
- **Απάντηση διαδρομής:** εάν μια ομότιμη οντότητα έχει μια διαδρομή προς μια ομότιμη οντότητα για τον οποίο ζητείται η διαδρομή, τότε αυτός στέλνει μια απάντηση διαδρομής σε αυτόν που έκανε την ερώτηση διαδρομής.

Κάθε ένα από τα μηνύματα ερώτησης και απάντησης περιλαμβάνει διάφορα στοιχεία, συμπεριλαμβανομένων των πεδίων:

- **Source:** η διεύθυνση τελικού σημείου της ομότιμης οντότητας πηγής
- **Destination:** η διεύθυνση του τελικού σημείου της πιθανής ομότιμης οντότητας προορισμού
- **LastHop:** η διεύθυνση τελικού σημείου της τελευταίας ομότιμης οντότητας τύπου δρομολογητή από το οποίο πέρασε το μήνυμα
- **NbOfHop:** ο συνολικός αριθμός ομότιμων οντοτήτων από τους οποίους πέρασε το μήνυμα
- **ForwardRoute:** μία προαιρετική λίστα διευθύνσεων τελικού σημείου που μια ομότιμη οντότητα θα πρέπει να στείλει ένα μήνυμα για να φθάσει στον προορισμό
- **ReverseRoute:** μία προαιρετική λίστα διευθύνσεων τελικού σημείου που μια ομότιμη οντότητα θα χρησιμοποιούσε για να αποκριθεί σε ένα μήνυμα που λαμβάνει

4.1.2 Υπηρεσία τελικού σημείου

Η υπηρεσία τελικού σημείου είναι υπηρεσία πυρήνα σε μια ομάδα ομότιμων οντοτήτων που παρέχει τη λειτουργία μεταφοράς μεταξύ των ομότιμων οντοτήτων χρησιμοποιώντας κάποιο πρωτόκολλο μεταφοράς της JXTA. Τα τρέχοντα πρωτόκολλα μεταφοράς της JXTA είναι τα ακόλουθα:

- μεταφορά μέσω HTTP [RFC2616]
- μεταφορά μέσω SMTP [RFC821]
- μεταφορά BEEP [RFC3080][RFC3081]
- μεταφορά μέσω TLS (transport layer security) [RFC2246]

Τα πρωτόκολλα μεταφορών JXTA είναι αρμόδια για τη χαμηλού επιπέδου επικοινωνία πάνω από τα συγκεκριμένα κανάλια επικοινωνίας. Η υπηρεσία τελικού σημείου χρησιμοποιείται από άλλες υπηρεσίες πυρήνα, όπως οι υπηρεσίες επίλυσης οντοτήτων και διάδοσης μηνυμάτων. Με τη χρησιμοποίηση της υπηρεσίας τελικού σημείου, οι ομότιμες οντότητες μπορούν να βεβαιωθούν ότι τα μηνύματα που στέλνονται στο δίκτυο JXTA χρησιμοποιούν την τοπολογία δικτύων που καθορίζεται στην προδιαγραφή. Υπάρχουν τρόποι να παρακαμφθεί η τοπολογία του δικτύου JXTA και να στέλνονται "ακατέργαστα" τα μηνύματα που χρησιμοποιούν οποιαδήποτε τοπολογία απαραίτητη για έναν δεδομένο στόχο. Εάν αναζητήσουμε τις λεπτομέρειες της υπηρεσίας τελικού σημείου, θα γίνει σαφές ότι η υπηρεσία είναι αρμόδια μόνο για την αποστολή των μηνυμάτων από μια ομότιμη οντότητα σε μια άλλη. Οι ομότιμες οντότητες

μπορούν να είναι προσιτές ή μια στην άλλη ή να συνδεθούν μέσω ενός τρίτου. Αυτό έχει νόημα επειδή τα περισσότερα από τα πρωτόκολλα μεταφορών JXTA σχεδιάζονται για την επικοινωνία ένας προς ένα μεταξύ των άμεσα συνδεδεμένων ομότιμων οντοτήτων. Το πρωτόκολλο τελικού σημείου αναλαμβάνει αυτή τη διαδικασία που κάνει τις αποσυνδεδεμένες ομότιμες οντότητες να εμφανιστούν ως συνδεδεμένες.

4.1.3 Αποστολή μηνύματος

Όταν ένα μήνυμα είναι έτοιμο να σταλεί από μια ομότιμη οντότητα σε μια άλλη, η υπηρεσία τελικού σημείου χρειάζεται τη διεύθυνση μιας πηγής και για έναν προορισμό υπό μορφή ταυτότητας. Η ταυτότητα κατασκευάζεται σύμφωνα με το ακόλουθο σχήμα:

όνομα_πρωτόκολλο://διεύθυνση_κατά_πρωτόκολλο/μοναδικό_όνομα_παραλήπτη/μοναδικό_όνομα_περιεχομένου_παραλήπτη

Με βάση το όνομα του πρωτοκόλλου, η υπηρεσία τελικού σημείου επικαλείται ένα πρωτόκολλο τελικού σημείου, όπως το TCP ή το HTTP. Η τιμή του πεδίου *“διεύθυνση_κατά_πρωτόκολλο”* αντιπροσωπεύει τη μηχανή στην οποία στέλνεται το μήνυμα, δηλαδή την διεύθυνση της μηχανής σύμφωνα με το πρωτόκολλο που ακολουθείται, για το πρωτόκολλο TCP, η τιμή αυτού του πεδίου θα ήταν χαρακτηριστικά μια διεύθυνση IP και η πόρτα. Όταν το μήνυμα φθάνει στη μηχανή προορισμού, ένας χειριστής επικαλείται βασισμένος σε μια αλληλουχία τιμών των

μοναδικό_όνομα_παραλήπτη και μοναδικό_όνομα_περιεχομένου.

4.1.4 Σύνδεση του πρωτοκόλλου τελικού σημείου με τη Java

Το πρωτοκόλλο τελικού σημείου υλοποιείται στην Java με το αρχείο EndpointRouter.java (δρομολογητής τελικού σημείου). Σκοπός του είναι να καθοδηγηθεί ένα μήνυμα από ένα το σημείο A σε ένα οποιοδήποτε σημείο X. Με αυτόν τον στόχο, ο δρομολογητής θα κάνει τα απαραίτητα για να ελαχιστοποιηθεί το ποσό της απαραίτητης εργασίας για την ολοκλήρωση του στόχου του. Ένα από τα εργαλεία που χρησιμοποιεί για να ελαχιστοποιήσουν την εργασία είναι η *αποθήκευση διαδρομών*. Όταν απαιτείται μια διαδρομή μεταξύ δύο ομότιμων οντοτήτων, ο δρομολογητής τελικού σημείου ελέγχει αν ήδη έχει αποθηκεύσει τη διαδρομή. Εάν μια ναι, τότε τη χρησιμοποιεί, εάν δεν βρεθεί καμία διαδρομή, τότε στέλνεται στο δίκτυο JXTA το μήνυμα ερώτησης διαδρομής.

Όλα τα μηνύματα από το δρομολογητή τελικού σημείου χρησιμοποιούν και τις υπηρεσίες ανακαλύψεων και τις υπηρεσίες δρομολόγησης. Αυτές οι υπηρεσίες πυρήνα επιτρέπουν στα μηνύματα ερώτησης να δημοσιευθούν και να καθοδηγηθούν σε όλο το δίκτυο JXTA. Μέσα στο δρομολογητή τελικού σημείου υπάρχει μια μέθοδος που εκτελεί αυτή την εργασία της εύρεσης μιας διεύθυνσης τελικού σημείου.

Αυτή η μέθοδος δέχεται την ταυτότητα μιας ομότιμης οντότητας και επιστρέφει τη διεύθυνση τελικού σημείου που μπορεί να χρησιμοποιήσει για να

επικοινωνήσει με μια ομότιμη οντότητα. Τα βήματα στην εύρεση μιας διαδρομής είναι τα ακόλουθα:

1. Ελέγχεται εάν η ομότιμη οντότητα συνδέεται άμεσα· ο δρομολογητής τελικού σημείου θα προσπαθήσει να φθάσει στην ομότιμη οντότητα προορισμού τοπικά. Εάν συνδέεται άμεσα, η διαδρομή βρίσκεται.
2. Εάν καμία άμεση διαδρομή δεν υπάρχει, ελέγχονται οι προηγούμενες διαδρομές. Εδώ ελέγχεται ο αποθηκευτικός χώρος για τη διαδρομή. Εάν μια διαδρομή βρίσκεται, χρησιμοποιείται.
3. Εάν η ομότιμη οντότητα δεν συνδέεται άμεσα και καμία διαδρομή δεν βρίσκεται στον αποθηκευτικό χώρο, η ερώτηση πρέπει να σταλεί στο δίκτυο JXTA για να βρεθεί η διαδρομή. Άρα στέλνεται μια ερώτηση ring.

Αυτή η διαδικασία συμβαίνει κάθε φορά που πρόκειται να σταλθεί ένα μήνυμα στο υπόλοιπο δίκτυο.

4.2 Πρωτόκολλο για ραντεβού

Το πρωτόκολλο ραντεβού είναι σχεδιασμένο για να πολλαπλασιάσει τα μηνύματα μεταξύ των ομότιμων οντοτήτων μέσα σε μια ομάδα. Μια εφαρμογή της προδιαγραφής πρωτοκόλλου επιτρέπει σε μια ομότιμη οντότητα να χρησιμοποιήσει μια υπηρεσία ραντεβού για τον πολλαπλή διάδοση μηνυμάτων και να στηριχθεί στην υπηρεσία για τη διαχείριση της διάδοσης. Μερικές λειτουργίες από τη διαχείριση περιλαμβάνουν τον έλεγχο, τη μεταβλητή «χρόνος ζωής» σε κάθε μήνυμα και την εξασφάλιση ότι τα μηνύματα δεν επαναλαμβάνονται στους τοπικούς βρόχους.

Ο μηχανισμός μεταφοράς που χρησιμοποιείται από τη Java υλοποίηση της JXTA είναι το HTTP και το TCP/IP. Ως επί το πλείστον, αυτοί οι μηχανισμοί δεν σχεδιάζονται για να στείλουν τις πληροφορίες από μια μηχανή σε πολλαπλές άλλες μηχανές, εκτός ίσως από το TCP multicast που είναι για τοπικά δίκτυα. Το πρωτόκολλο ραντεβού δημιουργήθηκε για να διευκολύνει τη μεταφορά των μηνυμάτων στους πολλαπλές ομότιμες οντότητες μέσα στο δίκτυο JXTA.

Μια *ομότιμη οντότητα ραντεβού* είναι απλά μια ομότιμη οντότητα που αποφασίζει να διαδώσει τα μηνύματα που λαμβάνει σε όσες ομότιμες οντότητες «γνωρίζει» και που είναι συνδεδεμένοι μαζί της. Όταν μια ομότιμη οντότητα χρησιμοποιεί την υπηρεσία ανακαλύψεων για να βρει διαφημίσεις στο δίκτυο JXTA, εκείνη η υπηρεσία χρησιμοποιεί την υπηρεσία επίλυσης που εφαρμόζεται που εδρεύει στο πρωτόκολλο επίλυσης ομότιμων οντοτήτων. Η υπηρεσία επίλυσης θα έχει την επιλογή της χρησιμοποίησης είτε του πρωτοκόλλου τελικού σημείου είτε του πρωτοκόλλου ραντεβού, είτε και των δύο. Με τη χρησιμοποίηση του πρωτοκόλλου ραντεβού και, στη συνέχεια, της υπηρεσίας ραντεβού, η υπηρεσία ανακαλύψεων θα κερδίσει ένα πολύ ευρύτερο «κοινό» για γεγονότα ανακάλυψη ή δημοσίευσης.

4.2.1 Διαφημίσεις ραντεβού

Όταν μια ομότιμη οντότητα είτε διαμορφώνεται ως ραντεβού είτε γίνεται δυναμικά τύπου ραντεβού, δημιουργεί μια διαφήμιση ραντεβού για να διαφημίσει την υπηρεσία ραντεβού στις άλλες ομότιμες οντότητες στην ομάδα. Η διαφήμιση περιγράφεται παρακάτω. Τα στοιχεία στη διαφήμιση είναι τα ακόλουθα:

Name: ένα προαιρετικό όνομα για την ομότιμη οντότητα ραντεβού

RdvGroupID: η όμοια ταυτότητα ομάδας με την οποία αυτή η ομότιμη οντότητα ραντεβού δρα

RdvPeerID: ο αριθμός ταυτότητας αυτής της ομότιμης οντότητας ραντεβού

```
<?xml version="1.0" encoding="UTF-8"?>
<jxta:RdvAdvertisement>
  <Name> Julian's Peer</Name>
  <RdvGroupId> PeerGroup ID </RdvGroupId>
  <RdvPeerId>Peer ID </RdvPeerId>
</jxta:RdvAdvertisement>
```

Σχήμα 4.2.1 xml αναπαράσταση μιας διαφήμισης ραντεβού, όπου: Name = ένα προαιρετικό όνομα, PeerGroup ID = το μοναδικό χαρακτηριστικό προσδιοριστικό της ομάδας Peer ID = το μοναδικό χαρακτηριστικό προσδιοριστικό της ομότιμης οντότητας

4.2.1 Διάδοση μηνυμάτων

Όπως αναφέρθηκε νωρίτερα, το πρωτόκολλο ραντεβού πρέπει να ελέγχει τη διάδοση των μηνυμάτων στο δίκτυο προκειμένου να αποφευχθεί η συσσώρευση «απορριμμάτων» κατά τον πολλαπλασιασμό τους. Για την αποφυγή αυτού του ενδεχομένου, ενσωματώνεται μαζί με το μήνυμα που διαδίδεται και ένα μήνυμα ραντεβού που ακολουθεί την παρακάτω δομή ώστε το πραγματικό μήνυμα να ξεχωρίζει από ένα αντίγραφο του που προέκυψε από μια ομότιμη οντότητα ραντεβού.

```
<xs:element name="RendezVousPropagateMessage"
type="jxta:RendezVousPropagateMessage"/>
<xs:complexType name="RendezVousPropagateMessage">
  <xs:element name="MessageId" type="xs:string"/>
  <xs:element name="DestSName" type="xs:string"/>
  <xs:element name="DestSPParam" type="xs:string"/>
  <xs:element name="TTL" type="xs:unsignedInt"/>
  <xs:element name="Path" type="xs:anyURI"
maxOccurs="unbounded"/>
</xs:complexType>
```

Σχήμα 4.2.2 Δομή που ακολουθεί ένα μήνυμα ραντεβού

Τα στοιχεία του μηνύματος είναι τα ακόλουθα:

- **MessageId** : ταυτότητα της JXTA που το συνδέει με το μήνυμα
- **DestSName** : το όνομα του προορισμού

- **DestSParam** : παράμετροι που συνδέονται με τον προορισμό
- **TTL** : ο χρόνος για τον οποίο θεωρείται έγκυρο το μήνυμα
- **Path** : η διεύθυνση του τελευταίου τον οποίο το μήνυμα επισκέφτηκε

Έτσι η κάθε ομότιμη οντότητα που λαμβάνει μηνύματα μπορεί να ξεχωρίσει τα μηνύματα αυτά από τα μηνύματα ραντεβού.

Το πρωτόκολλο ραντεβού είναι ένα σημαντικό μέρος της προδιαγραφής JXTA. Επιτρέπει στα μηνύματα να διαδίδονται μεταξύ των ομότιμων οντοτήτων χρησιμοποιώντας για μεταφορά πρωτόκολλα που δεν είναι σε θέση να διαδώσουν τα μηνύματα σε πολλαπλούς προορισμούς. Σαν πρωτόκολλο μεσολαβητών, το πρωτόκολλο ραντεβού χρησιμοποιείται από το πρωτόκολλο ανακάλυψης και το πρωτόκολλο επίλυσης ομότιμων οντοτήτων.

4.3 Πρωτόκολλο επίλυσης ομότιμων οντοτήτων

Όταν μια οντότητα χρειάζεται να δημιουργήσει ένα μήνυμα για να το στείλει στο δίκτυο σύγχυση θα δημιουργούνται εάν αυτός χρησιμοποιούσε οτιδήποτε πρωτόκολλο είχε διαθέσιμο να στείλει εκείνη τη στιγμή και η ομότιμη οντότητα που λάμβανε το μήνυμα προσπαθούσε να το διευκρινίσει. Η υλοποίηση της προδιαγραφής JXTA ορίζει το πρωτόκολλο επίλυσης ομότιμων οντοτήτων (Peer Resolver Protocol) με σκοπό τη δημιουργία ενός κοινού πλαισίου επικοινωνίας των ομότιμων οντοτήτων μέσω ερωταποκρίσεων.

Η υλοποίηση της προδιαγραφής JXTA στην Java δομεί μια υπηρεσία βασισμένη στο πρωτόκολλο επίλυσης της προδιαγραφής. Η υπηρεσία είναι, κατά σύμβαση, άμεσα συνδεδεμένη με την κάθε ομάδα ομότιμων οντοτήτων. Οι άλλες υπηρεσίες υψηλού επιπέδου (ανακάλυψης, πληροφόρησης και δέσμευσης σωλήνα) βασίζονται πάνω στην υπηρεσία επίλυσης.

Το πρωτόκολλο είναι ένα σύνολο γενικών μηνυμάτων ερώτησης και απάντησης με σκοπό να διευκολύνουν ένα κοινό σύστημα μηνύματος μεταξύ των ομότιμων στο δίκτυο JXTA. Για να μειωθεί το ποσό επεξεργασίας που πρέπει να κάνει μια υπηρεσία, τα μηνύματα ορίζονται και παραδίδονται σε έναν συγκεκριμένο «χειριστή» στην ομότιμη οντότητα. Ο *χειριστής* είναι το όνομα που ορίζεται σε έναν ορισμό που διευκρινίζει το σχήμα του μηνύματος, καθώς επίσης και την απάντηση που μπορεί να εμφανιστεί όταν λαμβάνεται ένα μήνυμα εκείνου του τύπου. Παραδείγματος χάριν, ένας χειριστής να χρησιμοποιήσει την υπηρεσία ραντεβού για να διαδώσει ένα μήνυμα στους πολλαπλάσιες ομότιμες οντότητες ή για να στείλει το μήνυμα σε μια συγκεκριμένη ομότιμη οντότητα.

Έτσι, όπως αναμένεται, η υπηρεσία ανακάλυψης ορίζει έναν χειριστή για να χειριστεί τα μηνύματα ερώτησης και απάντησης ανακαλύψεων. Όταν η υπηρεσία ανακαλύψεων πρέπει να στείλει ένα μήνυμα ερώτησης, περνά το μήνυμα στο πρωτόκολλο επίλυσης. Το πρωτόκολλο αυτό θα τυλίξει το μήνυμα ερώτησης ανακαλύψεων στο μήνυμά του και θα στείλει το νέο μήνυμα σε άλλες ομότιμες οντότητες. Το πρωτόκολλο επίλυσης στους μακρινές ομότιμες οντότητες θα λάβει το μήνυμα, θα το ξετυλίξει και θα το διαβιβάσει έπειτα στον αρμόδιο χειριστή.

4.3.1. Ονομασία χειριστών

Το πρωτόκολλο επίλυσης χρησιμοποιεί το όνομα χειριστών για να μεταφέρει τα μηνύματα που φθάνουν σε, ή που φεύγουν από, μια ομότιμη οντότητα σε μια συγκεκριμένη υπηρεσία τελικού σημείου. Ένας ακροατής (listener) χρησιμοποιείται για να συνδέσει μια υπηρεσία τελικού σημείου με το όνομα ενός χειριστή. Όλες οι λεπτομέρειες αντιμετωπίζονται με τη δημιουργία μιας υπηρεσίας επίλυσης (resolver service). Η τρέχουσα προδιαγραφή δηλώνει ότι όλες οι συνδέσεις του πρωτοκόλλου επίλυσης πρέπει να χρησιμοποιήσουν ένα κοινό σχήμα για την ονομασία. Το σχήμα είναι μια αλληλουχία σειράς του ονόματος υπηρεσιών, η ταυτότητα της ομάδας της ομότιμης οντότητας και μια μοναδική αξία. Για το όνομα χειριστών παρουσιάζεται παρακάτω κανονικό σχήμα.

```
<JXTARSLVRRSQRY> ::= <JXTARSLVRNAM> <JXTAIDVAL> <JXTARSLVRQRYTAG>  
<JXTARSLVRRSRSP> ::= <JXTARSLVRNAM> <JXTAIDVAL> <JXTARSLVRRSPTAG>  
<JXTARSLVRQRYTAG> ::= "ORes"  
<JXTARSLVRRSPTAG> ::= "IRes"  
<JXTARSLVRNAM> ::= "jxta.service.resolver"  
<JXTAIDVAL> ::= JXTA ID
```

Σχήμα 4.3.1 Ο τρόπος σχηματισμού ονόματος χειριστή

Μόνο δύο ακροατές καθορίζονται αυτήν την περίοδο για πρωτόκολλο επίλυσης.

Για τις ερωτήσεις:

jxta.service.resolver[group ID string]ORes

Για τις απαντήσεις:

jxta.service.resolver[group ID string]IRes

Αν και η υπηρεσία θα είναι μοναδική σε μια συγκεκριμένη ομότιμη οντότητα, όλες οι ομότιμες οντότητες μέσα σε μια ομάδα θα χρησιμοποιήσουν την ίδια υπηρεσία. Προφανώς, μπορούμε να δημιουργήσουμε μια ομάδα που δεν χρησιμοποιεί την προεπιλεγμένη εφαρμογή της υπηρεσίας επίλυσης. Σε αυτήν την περίπτωση, τα μηνύματα ερώτησης και απάντησης πρέπει ακόμα να είναι τα ίδια όπως αυτά καθορίζονται από την προδιαγραφή JXTA. Εντούτοις, θα μπορούσαμε να δημιουργήσουμε άλλους χειριστές με τα μοναδικά μηνύματα και ονόματα χειριστών.

4.3.2 Μηνύματα ερώτησης

Όπως αναφέραμε παραπάνω όταν μια ομότιμη οντότητα θα στείλει ένα μήνυμα αναζήτησης σε άλλες ομότιμες οντότητες σε μια ομάδα, θα δημιουργήσει το μήνυμα του με έναν δικό του τρόπο σύμφωνα με το δικό του σχήμα δημιουργίας. Έπειτα το μήνυμα θα δοθεί στην υπηρεσία επίλυσης για την παράδοση στους άλλες ομότιμες οντότητες, και η υπηρεσία επίλυσης θα τυλίξει το μήνυμα ανακάλυψης στο με το μήνυμά της. Στο παρακάτω σχήμα

παρουσιάζεται το XML σχήμα που χρησιμοποιείται για το μήνυμα ερώτησης της υπηρεσίας επίλυσης.

```
<xs:element name="ResolverQuery" type="jxta:ResolverQuery"/>
<xs:complexType name="ResolverQuery">
  <xs:element name="Credential" type="xs:anyType" minOccurs="0"/>
  <xs:element name="SrcPeerID" type="xs:anyURI"/>
  <!-- This could be extended with a pattern restriction -->
  <xs:element name="HandlerName" type="xs:string"/>
  <xs:element name="QueryID" type="xs:string" minOccurs="0"/>
  <xs:element name="Query" type="xs:anyType"/>
</xs:complexType>
```

Σχήμα 4.3.2 Τρόπος σχηματισμού ερώτησης

Τα στοιχεία του μηνύματος είναι τα ακόλουθα:

- **Credential:** Τα διαπιστευτήρια της ομότιμης οντότητας που στέλνει το μήνυμα
- **SrcPeerID:** Η ταυτότητα της ομότιμης οντότητας που επέβαλε την ερώτηση
- **HandlerName:** Ένα όνομα που ορίζει πως η υπηρεσία επίλυσης, της ομότιμης οντότητας που λαμβάνει την ερώτηση, θα επεξεργαστεί την ερώτηση.
- **QueryID:** Ένα μοναδικό προσδιοριστικό που καθορίζει την ερώτηση. Είναι το ίδιο που χρησιμοποιείται στην απάντηση για να προσδιορίσει την ερώτηση στην οποία αντιστοιχεί.
- **Query:** Η αναπαράσταση της ερώτησης.

4.3.3 Μηνύματα απάντησης

- Όταν μια ερώτηση εκτελείται, οι μακρινές ομότιμες οντότητες είναι πολύ πιθανό, αν και δεν είναι υποχρεωτικό, να απαντήσουν στην ερώτηση με μια απάντηση. Οι μακρινές ομότιμες οντότητες που θα απαντήσουν σε μια ερώτηση ανακάλυψης, θα βάλουν την απάντηση σε ένα μήνυμα απάντησης της υπηρεσίας ανακάλυψης, και η υπηρεσία ανακάλυψης θα περάσει το μήνυμα στην υπηρεσία επίλυσης που το παραδίδει στην αιτούσα ομότιμη οντότητα. Η υπηρεσία επίλυσης θα τυλίξει το μήνυμα στο σχήμα μηνυμάτων της, το οποίο παρουσιάζεται στο παρακάτω σχήμα, σχήμα 4.3.3.

```
<xs:element name="ResolverResponse" type="ResolverResponse"/>
<xs:complexType name="ResolverResponse">
  <xs:element name="Credential" type="xs:anyType" minOccurs="0"/>
  <xs:element name="HandlerName" type="xs:string"/>
  <xs:element name="QueryID" type="xs:string" minOccurs="0"/>
  <xs:element name="Response" type="xs:anyType"/>
</xs:complexType>
```

Σχήμα 4.3.3 Τρόπος σχηματισμού απάντησης

Τα στοιχεία στο μήνυμα απάντησης είναι τα ακόλουθα:

- **Credential:** Τα διαπιστευτήρια της ομότιμης οντότητας που στέλνει το μήνυμα
- **HandlerName:** Ένα όνομα που ορίζει πως η υπηρεσία επίλυσης, της ομότιμης οντότητας που λαμβάνει την ερώτηση, θα επεξεργαστεί την ερώτηση.
- **QueryID:** Ένα μοναδικό προσδιοριστικό που καθορίζει την απάντηση. Είναι το ίδιο που χρησιμοποιείται στην ερώτηση και προσδιορίζει την ερώτηση.
- **Response:** Η αναπαράσταση της ερώτησης.

Εννοείται πως για να ολοκληρωθεί μια επικοινωνία, χρειάζονται κι άλλα στοιχεία, όπως τα «στοιχεία» της ομότιμης οντότητας που επέβαλε την ερώτηση ή την απάντηση. Αυτά τα στοιχεία εσωκλείονται στο πεδίο “Query” ή “Response” αντίστοιχα όταν σε ένα μήνυμα επίλυσης και επεξεργάζονται από τις ανώτερου επιπέδου υπηρεσίες. Η υπηρεσία επίλυσης ασχολείται από με τα παραπάνω μηνύματα ερώτησης και απάντησης.

4.3.4 Σύνδεση της Java με το πρωτόκολλο επίλυσης

Η Java υλοποίηση της πλατφόρμας JXTA παρέχει υλοποιημένη μια υπηρεσία που βρίσκεται στο πακέτο `net.jxta.impl.resolver` ως υπηρεσία ομάδας ομότιμων οντοτήτων αποκαλούμενη `ResolverService`. Έτσι εάν χρειάζεται να εισέλθουμε σε όλες τις λεπτομέρειες χαμηλού επιπέδου, μπορούμε να χρησιμοποιήσουμε την υπηρεσία αυτή.

Για όλες τις βασικές και θεμελιώδεις διαδικασίες ερώτησης – απάντησης που βρίσκονται στις προδιαγραφές JXTA, το μόνο που χρειάζεται είναι η προεπιλεγμένη υπηρεσία επίλυσης στην προεπιλεγμένη ομάδα ομότιμων οντοτήτων (`NetPeerGroup`). Η κλάση `GenericResolver` είναι μια διεπαφή που καθορίζει τις μεθόδους για και τα μηνύματα ερώτησης και απάντησης. Η διεπαφή υπηρεσίας `ResolverService`, που προέρχεται από `GenericResolver`, εφαρμόζει τις μεθόδους για και τους χειριστές βασισμένους στη διεπαφή `QueryHandler`. Αντικείμενα της κλάσης χειριστή, `QueryHandler`, θα λάβουν ως είσοδο μηνύματα ερωτήσεων και απαντήσεων, δηλαδή αντικείμενα των κλάσεων `ResolverQueryMsg` και `ResolverResponseMsg`, όταν πρέπει να επικαλεσθεί η υπηρεσία επίλυσης για έναν συγκεκριμένο χειριστή.

4.4 Πρωτόκολλο πληροφόρησης ομότιμων οντοτήτων

Οι περισσότερες εφαρμογές διαδικτύου, υπάρχει πάντα η επιθυμία να γνωρίζουμε το τι κάνουν οι απομακρυσμένες ομότιμες οντότητες. Ίσως επιθυμούμε να γνωρίζουμε απλώς τα ονόματά τους, αλλά ακόμα ίσως είναι αναγκαίο να μάθουμε πόσο καιρό ομότιμη αυτή οντότητα είναι ενεργή, πότε έστειλε το τελευταίο μήνυμα και ακόμη πόσα δεδομένα στάλθηκαν από αυτήν.

Η προδιαγραφή JXTA περιλαμβάνει ένα πρωτόκολλο αποκαλούμενο πρωτόκολλο πληροφόρησης ομότιμων οντοτήτων για να χειριστεί τα αιτήματα

πληροφόρησης για τις μακρινές ομότιμες οντότητες. Όπως και τα άλλα πρωτόκολλα της πλατφόρμας JXTA, και το πρωτόκολλο πληροφόρησης χρησιμοποιεί μια σειρά μηνυμάτων που μεταφέρονται μεταξύ δύο ή περισσότερων ομότιμων οντοτήτων.

Έτσι όταν μια ομότιμη οντότητα εντοπίζεται στο δίκτυο μπορεί να ερωτηθεί για την κατάστασή της. Το πρωτόκολλο αυτό είναι ένα προαιρετικό πρωτόκολλο και οι ομότιμες οντότητες δεν είναι απαραίτητο ότι θα απαντήσουν σε ερωτήσεις αυτού του πρωτοκόλλου. Η αξιόπιστη μεταφορά είναι και αυτή προαιρετική για το πρωτόκολλο πληροφόρησης. Έτσι πολλά μηνύματα πληροφοριών μπορούν να σταλούν και κανένα, ένα ή πολλές απαντήσεις μπορεί να παραληφθούν σε οποιαδήποτε ερώτηση.

Το πρωτόκολλο πληροφόρησης βασίζεται πάνω στο στρώμα του πρωτοκόλλου επίλυσης ομότιμων οντοτήτων.

4.4.1 Μηνύματα ερώτησης

Όπως αναφέραμε παραπάνω όταν μια ομότιμη οντότητα θα στείλει ένα μήνυμα ερώτησης για πληροφορίες σε μια άλλη ομότιμη οντότητα, μέσα σε μια ομάδα, θα δημιουργήσει το μήνυμα του με έναν δικό του τρόπο σύμφωνα με το δικό του σχήμα δημιουργίας. Έπειτα το μήνυμα θα δοθεί στην υπηρεσία επίλυσης για την παράδοση στις άλλες ομότιμες οντότητες, και η υπηρεσία επίλυσης θα τυλίξει το μήνυμα αυτό με το μήνυμά της. Στο παρακάτω σχήμα παρουσιάζεται το XML σχήμα που χρησιμοποιείται για το μήνυμα ερώτησης της υπηρεσίας πληροφόρησης.

```
<xs:element name="PeerInfoQueryMessage"
type="jxta:PeerInfoQueryMessage"/>

<xs:complexType name="PeerInfoQueryMessage">
  <xs:sequence>
    <xs:element name="sourcePid" type="jxta:JXTAID"/>
    <xs:element name="targetPid" type="jxta:JXTAID"/>
    <!--if not present then the response is the general peerinfo-->
    <xs:element name="request" type="xs:anyType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Σχήμα 4.4.1 Τρόπος σχηματισμού μηνυμάτων ερωτήσεων

Τα στοιχεία είναι τα ακόλουθα:

- **sourcePid**: Ο αριθμός ταυτότητας της ομότιμης οντότητας που θέτει την ερώτηση
- **targetPid**: Ο αριθμός ταυτότητας της ομότιμης οντότητας που ερωτάται
- **request**: Προαιρετικό στοιχείο, δομή μιας ερώτησης

Το στοιχείο request αποτελεί τη βάση αυτού του μηνύματος ερώτησης. Εάν το στοιχείο αυτό είναι κενό, η ομότιμη οντότητα προορισμού θα υποθέσει ότι ο αιτών θέλει μόνο τις προεπιλεγμένες πληροφορίες, και ότι οι πληροφορίες

επιστρέφονται βασισμένος στο μήνυμα απάντησης. Εάν η ομότιμη οντότητα ενδιαφέρεται για τις πρόσθετες πληροφορίες, ένα μήνυμα ερώτησης μπορεί να εγκλειστεί στο στοιχείο request. Όμως, η εφαρμογή του πρωτοκόλλου πληροφόρησης, στην ομότιμη οντότητα που λαμβάνει τις ερωτήσεις πληροφόρησης, θα πρέπει να είναι σε θέση να «καταλάβει» και να επεξεργαστεί κατάλληλα το αίτημα πληροφοριών.

4.4.2 Μήνυμα απάντησης

Το πρωτόκολλο πληροφόρησης το οποίο παρέχει καθορισμένες πληροφορίες για την τρέχουσα κατάσταση μιας ομότιμης οντότητας, μηνύματα που δέχτηκε και έστειλε, το χρόνο που έλαβε και έστειλε το τελευταίο μήνυμα καθορίζει τον παρακάτω τρόπο σχηματισμού απαντήσεων σε ερωτήματα για πληροφόρηση.

Τα στοιχεία που το απαρτίζουν είναι τα εξής:

- **sourcePid** Ο αριθμός ταυτότητας της ομότιμης οντότητας που θέτει την ερώτηση
- **targetPid** Ο αριθμός ταυτότητας της ομότιμης οντότητας που ερωτάται
- **uptime** Ο σχετικός χρόνος σε millisecond για τον οποίο η υπηρεσία πληροφόρησης στην ερωτούμενη ομότιμη οντότητα άρχισε να εκτελείται. Οι ομότιμες οντότητες πρέπει να δώσουν αυτήν τη πληροφορία, αλλά μπορεί να επιλέξουν να μη το κάνουν αν η πληροφορία δεν είναι διαθέσιμη ή αν θα παρουσιάσει κάποιο θέμα ασφαλείας.
- **timestamp** Η απόλυτη χρονική στιγμή την οποία δημιουργήθηκε η απάντηση μετρούμενη από την 1^η Ιανουαρίου 1970, ώρα 00:00 GTM. Ομοίως οι ομότιμες οντότητες πρέπει να δίνουν αυτή τη πληροφορία, αλλά μπορούν να επιλέξουν να μην το κάνουν αν η πληροφορία δεν είναι διαθέσιμη ή για λόγους ασφαλείας.
- **response** Περιέχει μια ενδεχόμενη απάντηση σε ένα προηγούμενο μήνυμα ερώτησης του πρωτοκόλλου. Αναφέραμε πως το πρωτόκολλο πληροφόρησης στηρίζεται πάνω στο πρωτόκολλο επίλυσης. Έτσι για να υπάρχει αντιστοίχιση μιας απάντησης σε μια ερώτηση του πρωτοκόλλου πληροφόρησης, το στοιχείο QueryId που συναντάμε στο μήνυμα ερώτησης του πρωτοκόλλου επίλυσης, θα πρέπει να είναι το ίδιο στην απάντηση με αυτό της ερώτησης.
- **traffic** Το στοιχείο αυτό κρατά πληροφορίες για την κίνηση δικτύου στην ομότιμη οντότητα που απαντά. Και αυτό το στοιχείο είναι προαιρετικό.
- **lastIncomingMessageAt** Ο απόλυτος χρόνος στον οποίο η ομότιμη οντότητα έλαβε το τελευταίο έγκυρο μήνυμα. Μετράται σε milisecond από την 1^η Ιανουαρίου 1970, 00:00 GTM. Οι ομότιμες οντότητες πρέπει να παρέχουν αυτό το στοιχείο, αλλά για λόγους ασφαλείας ή αν η πληροφορία αυτή δεν είναι διαθέσιμη.
- **lastOutgoingMessageAt** Ο απόλυτος χρόνος στον οποίο η ομότιμη οντότητα έστειλε το τελευταίο έγκυρο μήνυμα. Μετράται σε milisecond από την 1^η Ιανουαρίου 1970, 00:00 GTM. Οι ομότιμες οντότητες πρέπει να παρέχουν αυτό το στοιχείο, αλλά για λόγους ασφαλείας ή αν η πληροφορία αυτή δεν είναι διαθέσιμη.

- **in** Αν είναι παρόν, περιέχει διάφορα στοιχεία που περιγράφουν την εισερχόμενη κίνηση από διάφορα τελικά σημεία.
ο **transport** Ο αριθμός των bytes που παραλήφθηκαν από ένα συγκεκριμένο τελικό σημείο και το όνομα του τελικού σημείου
- **out** Αν είναι παρόν, περιέχει διάφορα στοιχεία που περιγράφουν την εξερχόμενη κίνηση από διάφορα τελικά σημεία.
ο **transport** Ο αριθμός των bytes που παραλήφθηκαν από ένα συγκεκριμένο τελικό σημείο και το όνομα του τελικού σημείου.

```
<xs:element name="PeerInfoResponse" type="jxta:PeerInfoResponse"/>
<xs:complexType name="PeerInfoResponseMessage">
  <xs:sequence>
    <xs:element name="sourcePid" type="jxta:JXTAID"/>
    <xs:element name="targetPid" type="jxta:JXTAID"/>
    <xs:element name="uptime" type="xs:unsignedLong" minOccurs="0"/>
    <xs:element name="timestamp" type="xs:unsignedLong" minOccurs="0"/>
    <xs:element name="response" type="xs:anyType" minOccurs="0"/>
    <xs:element name="traffic" type="jxta:piptraffic" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="piptraffic">
  <xs:sequence>
    <xs:element name="lastIncomingMessageAt" type="xs:unsignedLong" minOccurs="0"/>
    <xs:element name="lastOutgoingMessageAt" type="xs:unsignedLong" minOccurs="0"/>
    <xs:element name="in" type="jxta:piptrafficinfo" minOccurs="0"/>
    <xs:element name="out" type="jxta:piptrafficinfo" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="piptrafficinfo">
  <xs:sequence>
    <xs:element name="transport" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:unsignedLong">
            <xs:attribute name="Expiration" type="xs:anyURI"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Σχήμα 4.4.2 Τρόπος σχηματισμού μηνυμάτων απάντησης

4.4.3 Σύνδεση της Java με το πρωτόκολλο πληροφόρησης

Η Java υλοποίηση της πλατφόρμας JXTA παρέχει με το πακέτο `net.jxta.impl.peer` την υλοποίηση του πρωτοκόλλου, δηλαδή την υπηρεσία υποβολής και λήψης με τα μηνύματα ερώτησης και απάντησης. Έτσι εάν

επιθυμούμε να δημιουργήσουμε τις δικές μας πληροφορίες για την κατάσταση των ομότιμων οντοτήτων και να έχουμε πρόσβαση στις πληροφορίες αυτές, μπορούμε να κάνουμε χρήση αυτού του πακέτου.

Για όλες τις βασικές και θεμελιώδεις διαδικασίες ερώτησης – απάντησης που βρίσκονται στις προδιαγραφές JXTA, το μόνο που χρειάζεται είναι η προεπιλεγμένη υπηρεσία πληροφόρησης στο πακέτο `net.jxta.peer`. Η κλάση `PeerInfoService` είναι μια διεπαφή που καθορίζει τις μεθόδους για και τα μηνύματα ερώτησης και απάντησης.

Η υπηρεσία πληροφόρησης δημιουργείται και ένα αντικείμενο ακροατή (`listener`), όπως ορίζεται από την κλάση `PeerInfoListener`, που συνδέεται με αυτήν. Όταν μια ερώτηση πληροφόρησης στέλνεται, το αντικείμενο ακροατή θα περιμένει μια απάντηση. Όταν επιστρέφεται μια απάντηση, αυτή είναι εγκλεισμένη σε ένα αντικείμενο `PeerInfoEvent`. Το αντικείμενο `PeerInfoEvent` περιλαμβάνει μια μέθοδο για εξάγουμε ένα αντικείμενο `PeerInfoAdvertisement`. Μέσα στο αντικείμενο `PeerInfoAdvertisement` υπάρχουν μέθοδοι για να εξάγουμε τα διάφορα στοιχεία του αντικειμένου χωρίς να πρέπει να εκτελεσθεί XML ανάλυση.

4.5 Πρωτόκολλο δέσμευσης σωλήνων

Ένα από τα πιο πολύ χρησιμοποιημένα πρωτόκολλα της JXTA προδιαγραφής είναι το πρωτόκολλο δέσμευσης σωλήνα. Βασίζεται στην υπόθεση ότι μια ομότιμη οντότητα θα δημιουργήσει ένα σωλήνα εισόδου και επιθυμεί οι απομακρυσμένοι ομότιμες οντότητες να συνδεθούν με το σωλήνα για να ανταλλάξουν πληροφορίες.

Το πρωτόκολλο δέσμευσης σωλήνα καθορίζει έναν σωλήνα και καθορίζει πως αυτός θα χρησιμοποιηθεί ανάμεσα σε ομότιμες οντότητες μέσα σε μια ομάδα ομότιμων οντοτήτων. Όπως έχουμε ήδη αναφέρει, ο σωλήνας είναι μια αφηρημένη έννοια για να περιγράψει ένα κανάλι επικοινωνίας βασισμένο σε ένα πρωτόκολλο μεταφοράς του διαδικτύου, όπως το HTTP ή το TCP/IP.

4.5.1 Επισκόπηση του πρωτοκόλλου

Το πρωτόκολλο δέσμευσης σωλήνα αποτελεί μια υπηρεσία υψηλού επιπέδου για να επιτρέψει στα μηνύματα να μεταφερθούν από ομότιμη οντότητα σε ομότιμη οντότητα. Για να διευκολυνθεί η μετακίνηση πληροφοριών σε ένα σωλήνα το πρωτόκολλο αυτό στηρίζεται πάνω στο πρωτόκολλο τελικού σημείου και το πρωτόκολλο επίλυσης για να λάβει πληροφορίες για τη διαδρομή επικοινωνίας μεταξύ ομότιμων οντοτήτων. Η τρέχουσα προδιαγραφή πρωτοκόλλου περιγράφει τρεις διαφορετικούς τύπους σωλήνων:

- **Unicast** Μια μονόδρομη σύνδεση μεταξύ δύο ομότιμων οντοτήτων. Η προδιαγραφή δηλώνει ότι ο σωλήνας αυτού του τύπου είναι αναξιόπιστος και μη ασφαλής. Οι περισσότερες εφαρμογές θα χρησιμοποιήσουν το TCP/IP για το σωλήνα Unicast. Στην προδιαγραφή και την εφαρμογή της Java, η περιγραφή για αυτόν τον τύπο σωλήνων ονομάζεται `JxtaUnicast`.

- **SecureUicast** Μια μονόδρομη σύνδεση μεταξύ δύο ομότιμων οντοτήτων. Η σύνδεση είναι ασφαλής και χρησιμοποιεί TLS (Transport Layer Security) για την κρυπτογράφηση. Στη Java εφαρμογή της JXTA προδιαγραφής η περιγραφή γι' αυτόν τον τύπο ονομάζεται JxtaUicastSecure.
- **Propagate** Μια σύνδεση από έναν προς πολλές απομακρυσμένες ομότιμες οντότητες. Η σύνδεση είναι αναξιόπιστη και μη ασφαλής. Στη Java εφαρμογή της JXTA προδιαγραφής η περιγραφή γι' αυτόν τον τύπο ονομάζεται JxtaPropagate.

Η Java εφαρμογή της JXTA περιλαμβάνει και αξιόπιστους και αμφίδρομους σωλήνες, αλλά και αυτοί στηρίζονται σε μονόδρομους και αναξιόπιστους σωλήνες όπως ορίζονται από την προδιαγραφή.

4.5.2 Διαφήμιση σωλήνα

Μια ομότιμη οντότητα χρησιμοποιεί διαφημίσεις σωλήνα για να περιγράψει τα χαρακτηριστικά ενός σωλήνα. Έτσι μια διαφήμιση ενός σωλήνα περιέχει όλες τις απαραίτητες πληροφορίες για έναν σωλήνα.

Τα στοιχεία είναι τα εξής:

- **Name:** Ένα προαιρετικό όνομα
- **ID:** Ο μοναδικός αριθμός ταυτότητας του σωλήνα
- **Type:** Πρόκειται για το τύπο σωλήνα JxtaUicast, JxtaUicastSecure, ή JxtaPropagate

Το σχήμα σύμφωνα με το οποίο σχηματίζονται οι σωλήνες φαίνεται παρακάτω:

```
<xs:element name="PipeAdvertisment" type="jxta:PipeAdvertisment"/>

<xs:complexType name="PipeAdvertisment">
  <xs:sequence>
    <xs:element name="Id" type="jxta:JXTAID"/>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="Name" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Σχήμα 4.5.1 Τρόπος σχηματισμού ενός σωλήνα

Το πρωτόκολλο δέσμευσης σωλήνα θα πρέπει να κάνει αρκετή δουλειά ώστε τελικά να συνδέσει δύο ομότιμες οντότητες χρησιμοποιώντας ένα σωλήνα. Μια ομότιμη οντότητα θα δημιουργήσει ένα σωλήνα εισόδου και μια διαφήμιση για αυτόν, την οποία θα τη δημοσιεύσει στην ομάδα του. Μια απομακρυσμένη ομότιμη οντότητα θα ανακαλύψει τον σωλήνα εισαγωγής μέσω ενός μηνύματος αναζήτησης ή κατευθείαν από της πληροφορίες που παρέχει η τοπική μας ομότιμη οντότητα. Η απομακρυσμένη ομότιμη οντότητα δεν μπορεί να υποθέτει πως η τοπική μας ομότιμη οντότητα έχει δημιουργήσει πραγματικά το σωλήνα και τον παρέχει έτοιμο προς χρήση.


```

<xs:element name="jxta:PipeResolver" type="jxta:PipeResolver"/>

<xs:simpleType name="PipeResolverMsgType">
  <xs:restriction base="xs:string">
    <!-- QUERY -->
    <xs:enumeration value="Query"/>
    <!-- ANSWER -->
    <xs:enumeration value="Answer"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="PipeResolver">
  <xs:sequence>
    <xs:element name="MsgType" type="jxta:PipeResolverMsgType"/>
    <xs:element name="PipeId" type="jxta:JXTAID"/>
    <xs:element name="Type" type="xs:string"/>

    <!-- used in the query -->
    <xs:element name="Cached" type="xs:boolean" default="true" minOccurs="0"/>
    <xs:element name="Peer" type="jxta:JXTAID" minOccurs="0"/>

    <!-- used in the answer -->
    <xs:element name="Found" type="xs:boolean"/>
    <!-- This should refer to a peer adv, but is instead a whole doc -->
    <xs:element name="PeerAdv" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```

Σχήμα 4.5.2 Τρόπος σχηματισμού μηνύματος επίλυσης σωλήνα

Έτσι η απομακρυσμένη ομότιμη οντότητα προτού συνδεθεί στέλνει στην απομακρυσμένη ομότιμη οντότητα ένα μήνυμα επίλυσης σωλήνα για να βρεθεί το τελικό σημείο του σωλήνα εισαγωγής. Το ίδιο σχήμα μηνύματος χρησιμοποιείται και ως ερώτηση και ως απάντηση.

Τα στοιχεία του μηνύματος επίλυσης σωλήνα είναι τα εξής:

- **MsgType:** Μπορεί να έχει την τιμή είτε «*Query*» είτε «*Answer*». Έχει την τιμή «*Query*» όταν μια απομακρυσμένη ομότιμη οντότητα στέλνει το μήνυμα σε μια τοπική ομότιμη οντότητα και την τιμή «*Answer*» όταν η τοπική μας ομότιμη οντότητα απαντά στην ερώτηση της αρχικής ομότιμης οντότητας.
- **PipeID:** Ο JXTA αριθμός ταυτότητας του σωλήνα για τον οποίο στέλνεται η ερώτηση
- **Type:** Ο τύπος του σωλήνα JxtaUnicast, JxtaUnicastSecure, ή JxtaPropagate.
- **Cached:** Λαμβάνει δύο τιμές, η *false* σημαίνει πως η απάντηση δεν πρέπει να προέρχεται από τον χώρο αποθήκευσης της ομότιμης οντότητας αλλά από τρέχοντα δεδομένα.
- **Peer:** Είναι ο JXTA αριθμός ταυτότητας της ομότιμης οντότητας που πρέπει να απαντήσει στην ερώτηση. Αν η τιμή δεν είναι παρούσα και το στοιχείο *cached* είναι *true* τότε μια ενδιαμέσση ομότιμη οντότητα μπορεί να απαντήσει στην ερώτηση. Σε μηνύματα απάντησης το στοιχείο αυτό δείχνει τις ομότιμες οντότητες που γνωρίζουν το σωλήνα.
- **Found:** Η τιμή *true* δείχνει ότι ο σωλήνας βρέθηκε

- **PeerAdv:** Η διαφήμιση της ομότιμης οντότητας που δημιουργήσε τελικά το σωλήνα εισόδου.

4.5.3 Σύνδεση της Java με το πρωτόκολλο πληροφόρησης

Η Java υλοποίηση της πλατφόρμας JXTA παρέχει με το πακέτο `net.jxta.pipe` για την υλοποίηση του πρωτοκόλλου. Το πρωτόκολλο υλοποιείται σαν υπηρεσία ομάδας, στην οποία η τρέχουσα ομότιμη οντότητα είναι μέλος. Η υλοποίηση του πρωτοκόλλου είναι υπηρεσία πυρήνα στην προεπιλεγμένη ομάδα (`NetPeerGroup`), ονομάζεται `PipeService` και παρέχει την ικανότητα να δημιουργήσουμε σωλήνες εισαγωγής και σωλήνες εξαγωγής από διαφημίσεις σωλήνα και να συνδεθούμε με αυτούς.

4.6 Πρωτόκολλο ανακάλυψης ομότιμων οντοτήτων

Το πρωτόκολλο ανακάλυψης ομότιμων οντοτήτων χρησιμοποιείται για την ανακάλυψη και την ανακοίνωση οποιουδήποτε πόρου που μπορεί να παρέχει μια ομότιμη οντότητα. Οι πόροι, όπως έχουμε ήδη αναφέρει, παρουσιάζονται σαν διαφημίσεις. Ένας πόρος μπορεί να είναι μια ομότιμη οντότητα, μια ομάδα, ένας σωλήνας, μια υπηρεσία ή οτιδήποτε που μπορεί να παρουσιαστεί με μια διαφήμιση. Έτσι κάθε διαθέσιμος πόρος πρέπει να αναπαρίσταται από μια διαφήμιση.

Η αρχική πρόθεση ήταν το πρωτόκολλο ανακάλυψης να παρέχει μια ουσιαστική υποδομή για την οικοδόμηση υψηλού επιπέδου υπηρεσιών ανακάλυψης. Σε πολλές περιπτώσεις οι πληροφορίες ανακαλύψεων είναι γνωστές καλύτερα από μια υψηλού επιπέδου υπηρεσία, επειδή η υπηρεσία μπορεί να έχει μια καλύτερη γνώση της τοπολογίας ομάδας. Το πρωτόκολλο παρέχει ένα βασικό μηχανισμό για την ανακάλυψη διαφημίσεων, ενώ ταυτόχρονα επιτρέπει σε υψηλού επιπέδου υπηρεσίες να συμμετέχουν στην διαδικασία της ανακάλυψης.

Οι υπηρεσίες πρέπει να υποδεικνύουν χαρακτηριστικά ώστε να βελτιώσουν την διαδικασία της ανακάλυψης, π.χ. να αποφασίζουν για το ποιες διαφημίσεις είναι χρήσιμο να αποθηκεύονται στον δίσκο ώστε να μην επαναλαμβάνεται η διαδικασία της ανακάλυψης.

Το πρωτόκολλο ανακάλυψης χρησιμοποιεί το πρωτόκολλο επίλυσης για να στέλνει ερωτήσεις και απαντήσεις διαδρομών.

4.6.1 Επισκόπηση Πρωτοκόλλου

Το πρωτόκολλο ανακάλυψης δουλεύει σε δύο επίπεδα, πρώτα σε τοπικό επίπεδο, ερευνώντας το χώρο αποθήκευσης της τοπικής ομότιμης οντότητας, και σε επίπεδο ομάδας, για την αναζήτηση στο δίκτυο. Και στις δύο περιπτώσεις η

αιτούσα ομότιμη οντότητα καλεί μια μέθοδο της υπηρεσία αναζήτησης, η οποία υλοποιεί το συγκεκριμένο πρωτόκολλο. Οι παράμετροι αναζήτησης είναι οι εξής:

- Ένα χαρακτηριστικό που χρησιμοποιείται σαν κλειδί
- Η τιμή αυτού του κλειδιού
- Ένας αριθμός για το μέγιστο αριθμό των απαντήσεων από κάθε ομότιμη οντότητα

Στην περίπτωση τοπικής αναζήτησης η υπηρεσία αναζήτησης θα ψάξει στον τοπικό χώρο αποθήκευσης να βρει κάποιο ταίριαγμα στην ερώτησης αναζήτησης. Ο τοπικός αποθηκευτικός χώρος είναι ο χώρος στον οποίο αποθηκεύονται διαφημίσεις που στέλνονται από άλλες ομότιμες οντότητες. Το ταίριαγμα στην ερώτηση αναζήτησης σημαίνει να βρεθεί μια διαφήμιση που να έχει ένα στοιχείο όπως το κλειδί και η τιμή του στοιχείου αυτού να είναι ίση με την τιμή του κλειδιού. Η υπηρεσία αναζήτησης, στην περίπτωση τοπικής αναζήτησης, δεν χρειάζεται να δημιουργήσει κάποιο μήνυμα ερώτησης, απλά θα αναζητηθούν οι παράμετροι στις βασικές δομές δεδομένων στις οποίες αποτυπώνονται τα δεδομένα των διαφημίσεων. Φυσικά, εγείρονται ερωτήματα για περιπτώσεις παλιών διαφημίσεων, για τις οποίες οι πόροι που παριστάνουν δεν υπάρχουν στο δίκτυο.

Στην περίπτωση απομακρυσμένης αναζήτησης η αιτούσα ομότιμη οντότητα θα στείλει ένα μήνυμα αναζήτησης στις άλλες ομότιμες οντότητες στο δίκτυο χρησιμοποιώντας το πρωτόκολλο επίλυσης. Οι ομότιμες οντότητες που λαμβάνουν την ερώτηση, θα εξετάσουν τις διαφημίσεις τους μήπως βρουν κάποιο ταίριαγμα. Αν βρουν, τότε θα στείλουν την διαφήμιση στην οποία το βρήκαν, αλλά να μην υπερβαίνουν το μέγιστο αριθμό, ενώ αν δεν βρουν δεν απαντούν. Η αιτούσα ομότιμη οντότητα τελικά αποθηκεύει τις διαφημίσεις που του ήλθαν, στον τοπικό αποθηκευτικό χώρο αυξάνοντας τον όγκο του, και έπειτα επιλέγει να χρησιμοποιήσει οποιαδήποτε από αυτές επιθυμεί.

4.6.2 Μηνύματα αναζήτησης

Ένα μήνυμα αναζήτησης θα δημιουργηθεί για κάθε απομακρυσμένη αναζήτηση σύμφωνα με τον τρόπο σχηματισμού στο σχήμα 4.6.1.

Τα στοιχεία του μηνύματος αυτού είναι τα εξής:

- **Type** Ο τύπος της διαφήμισης. Μόνο διαφημίσεις του ζητούμενου τύπου μπορούν να αναζητηθούν. Δυνατές τιμές είναι:
 - ο 0 Όταν πρόκειται για διαφημίσεις ομότιμων οντοτήτων (Peer Adv)
 - ο 1 Όταν πρόκειται για διαφημίσεις ομάδων (Peergroup Adv)
 - ο 2 Όταν πρόκειται για οποιαδήποτε άλλου είδους διαφήμιση (pipe, module, ορισμένη από τον χρήστη διαφήμιση κτλ.)
- **Threshold:** Ο μέγιστος αριθμός των διαφημίσεων που επιθυμούμε να επιστρέψει η κάθε ομότιμη οντότητα. Ο συνολικός αριθμός των απαντήσεων που θα λάβουμε εξαρτάται από τις ομότιμες οντότητες που απαντούν και τον αριθμό των διαφημίσεων που ο καθένας τους στέλνει. Αν ο τύπος της διαφήμισης είναι 0 (Peer Advertisements) και το κατώφλι (Threshold) είναι

0, τότε η ερώτηση έχει ειδικό νόημα στόχος της είναι να συλλέξει τέτοιου είδους διαφημίσεων. Έτσι κάθε ομότιμη οντότητα πρέπει να απαντήσει σε μια τέτοια ερώτηση ακόμη κι αν στην απάντηση δεν περικλείει κάποιο αποτέλεσμα.

```
<xs:element name="DiscoveryQuery" type="jxta:DiscoveryQuery"/>

<xs:simpleType name="DiscoveryQueryType">
  <xs:restriction base="xsd:string">
    <xs:enumeration value="0"/><!-- peer -->
    <xs:enumeration value="1"/><!-- group -->
    <xs:enumeration value="2"/><!-- adv -->
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="DiscoveryQuery">
  <xs:sequence>
    <xs:element name="Type" type="jxta:DiscoveryQueryType"/>
    <xs:element name="Threshold" type="xs:unsignedInt" minOccurs="0"/>
    <xs:element name="Attr" type="xs:string" minOccurs="0"/>
    <xs:element name="Value" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Σχήμα 4.6.1 Τρόπος σχηματισμού ερώτησης αναζήτησης

- **PeerAdv:** Αν είναι παρόν τότε είναι η διαφήμιση της ομότιμης οντότητας που κάνει την ερώτηση.
- **Attr:** Πρόκειται για το στοιχείο που θα αναζητηθεί σε όλες τις διαφημίσεις
- **Value:** Αφορά την τιμή του παραπάνω χαρακτηριστικού στοιχείου που βρέθηκε σε μια διαφήμιση. Οποιοδήποτε από τα δύο αυτά στοιχεία μπορεί να είναι παρόν ή απών. Αν είναι απών κάποιο από αυτά (ή και τα δύο), τότε η ομότιμη οντότητα που απαντά στέλνει ως απάντηση ένα σύνολο από διαφημίσεις (το πολύ ως τον αριθμό που περιέχεται στο *Threshold*), του συγκεκριμένου τύπου (όπως περιγράφεται από το *Type*), στις οποίες βρήκε να υπάρχει κάποιο από τα ζητούμενα δεδομένα (ή όλες τις διαφημίσεις αν και τα δύο, *Attr* και *Value*, είναι απών). Αν και τα δύο αυτά στοιχεία είναι παρόν, τότε μόνο διαφημίσεις που περιέχουν ως στοιχείο το *Attr* με τιμή αυτού το *Value* είναι νόμιμο να βρεθούν. Η παράμετρος αναζήτησης μπορεί να ξεκινά ή και να τελειώνει με το χαρακτήρα “*”. Σε αυτή την περίπτωση θα αναζητηθούν οι τιμές των χαρακτηριστικών *Attr* που περιέχουν σαν ακολουθία την τιμή του *Value*, αντικαθιστώντας το κάθε “*” με οποιαδήποτε σύμβολα χαρακτήρων.

4.6.3 Μηνύματα απάντησης

Ο τρόπος απάντησης σε ένα μήνυμα αναζήτησης δημιουργείται κατά το παρακάτω σχήμα.

```

<xs:element name="DiscoveryResponse" type="jxta:DiscoveryResponse"/>
<xs:complexType name="DiscoveryResponse">
  <!-- this should be an enumeration -->
  <xs:element name="Type" type="xs:string"/>
  <xs:element name="Count" type="xs:unsignedInt" minOccurs="0"/>
  <xs:element name="PeerAdv" type="xs:anyType" minOccurs="0">
    <xs:attribute name="Expiration" type="xs:unsignedLong"/>
  </xs:element>
  <xs:element name="Attr" type="xs:string" minOccurs="0"/>
  <xs:element name="Value" type="xs:string" minOccurs="0"/>
  <xs:element name="Response" type="xs:anyType" maxOccurs="unbounded">
    <xs:attribute name="Expiration" type="xs:unsignedLong"/>
  </xs:element>
</xs:complexType>

```

Σχήμα 4.6.2 Τόπος σχηματισμού απάντησης σε ερώτηση αναζήτησης

Τα στοιχεία της απάντησης είναι τα εξής:

- **Type:** Ο τύπος της διαφήμισης που επιστέφει ως απάντηση
- **Count:** Ο συνολικός αριθμός των στοιχείων Response στο μήνυμα απάντησης.
- **PeerAdv:** Η διαφήμιση της ομότιμης οντότητας που απαντά.
 - ο **Expiration:** Ένα γνώρισμα του στοιχείου PeerAdv που δείχνει το συνολικό αριθμό σε milliseconds μέχρι να *λήξει* η διαφήμιση.
- **Attr και Value:** Είναι η τιμές των στοιχείων που χρησιμοποιήθηκαν στην αναζήτηση της επιστρεφόμενης διαφήμισης.
- **Response:** Περιλαμβάνει το σύνολο των απαντήσεων που επιστρέφει μια ομότιμη οντότητα, τόσες διαφημίσεις όσες σημειώνονται στο στοιχείο *count*.
 - ο **Expiration:** Ένα γνώρισμα του στοιχείου Response που δείχνει το συνολικό αριθμό σε milliseconds μέχρι να *λήξει* η επιστρεφόμενη διαφήμιση.

4.6.4 Πολιτικές και ποιότητα υπηρεσίας

Το πρωτόκολλο ανακάλυψης δεν εγγυάται ότι οι ομότιμες οντότητες που θα λάβουν μια ερώτηση θα αποκριθούν, ούτε ότι αυτοί που θα αποκριθούν θα τιμήσουν το όριο των διαφημίσεων. Η καλύτερη προσπάθεια καταβάλλεται μόνο στο ταίριασμα της ερώτησης με τα δεδομένα στον αποθηκευτικό χώρο της ομότιμης οντότητας που λαμβάνει την ερώτηση. Από κει και πέρα δεν υπάρχει καμία εγγύηση ότι η ομότιμη οντότητα αυτός θα απαντήσει. Είναι σημαντικό να επισημανθεί ότι η απάντηση σε ένα μήνυμα αναζήτησης (DiscoveryQuery) είναι προαιρετική και μια ομότιμη οντότητα δεν είναι αναγκασμένη να απαντήσει.

Ένα έμπιστο μέσο μεταφορά είναι επίσης προαιρετικό με το πρωτόκολλο αναζήτησης. Πολλαπλά ερωτήματα αναζήτησης μπορεί να σταλούν να ληφθούν πολλές, μια ή και καμία απάντηση.

Μια ομότιμη οντότητα μπορεί να λάβει απαντήσεις (DiscoveryResponse) οι οποίες να μην αντιστοιχούν σε ερωτήσεις αναζήτησης (DiscoveryQuery) που αυτός έθεσε αρχικά. Αυτός ο μηχανισμός αν και φαίνεται εκ πρώτης όψης λάθος,

παρέχει την δυνατότητα να δημοσιεύουμε τους πόρους μας σε απομακρυσμένες ομότιμες οντότητες.

Το πρωτόκολλο ανακάλυψης παρέχει έναν γενικό μηχανισμό στις υπηρεσίες να αναζητούν στο δίκτυο πόρους, θέτοντας και λαμβάνοντας ερωτήσεις. Υλοποιημένη σαν υπηρεσία βοηθά τις άλλες υπηρεσίες, αναλαμβάνοντας την αποθήκευσή τους στον τοπικό χώρο αποθήκευσης και τη λήξη των διαφημίσεων.

Η πραγματική εργασία θέσης, προώθησης και λήψης των ερωτήσεων αναζήτησης γίνεται από το πρωτόκολλο επίλυσης.

4.6.5 Σύνδεση της Java με το πρωτόκολλο ανακάλυψης

Μια υλοποίηση του πρωτοκόλλου σαν υπηρεσία υπάρχει στο πακέτο *net.jxta.impl.discovery*. Το πακέτο *net.jxta.discovery* παρέχει πρόσβαση στο πρωτόκολλο αναζήτησης υλοποιημένο ως υπηρεσία αναζήτησης για την προεπιλεγμένη ομάδα ομότιμων οντοτήτων. Αυτή η υπηρεσία αναλαμβάνει όλες τις διαδικασίες αναζήτησης. Έτοιμο για την αναζήτηση διαφημίσεων η υπηρεσία αυτή θα πρέπει να ανακτηθεί από την προεπιλεγμένη ομάδα (NetPeerGroup), στην οποία η ομότιμη οντότητα είναι μέλος, και έπειτα μέσω μεθόδων επιτρέπει την αναζήτηση αποθηκευμένων διαφημίσεων, στον τοπικό χώρο αποθήκευσης αλλά και απομακρυσμένων διαφημίσεων, στην ίδια ομάδα ομότιμων οντοτήτων.

Η διαφορά μεταξύ αυτών των δύο πακέτων είναι ότι, το δεύτερο πακέτο παρέχει την διεπαφή προς την υπηρεσία αντί την υπηρεσία, αντί για την καθεαυτό υλοποιημένη υπηρεσία σε μια Java κλάση, όπως συμβαίνει στο πρώτο πακέτο.

ΜΕΡΟΣ ΙΙΙ

ΥΛΟΠΟΙΗΣΗ ΠΛΑΤΦΟΡΜΑΣ

1. Εισαγωγή

Η ανάπτυξη δικτυακών υπηρεσιών προάγει την ολοκλήρωση των εφαρμογών σε ένα καθολικό σχήμα. Πρόβλημα όμως αντιμετωπίζει ο κάθε πελάτης μια τέτοιας υπηρεσίας, όταν δεν γνωρίζει τη θέση της, για να την χρησιμοποιήσει. Έπειτα, η χρησιμοποίησή της είναι μια άλλη υπόθεση. Ο προμηθευτής πρέπει να έχει προβλέψει μια διεπαφή με τον χρήστη της υπηρεσίας για την συναλλαγή. Αν όχι τότε ο χρήστης θα πρέπει να κατέχει προγραμματιστικές γνώσεις ώστε να καταφέρει να την χρησιμοποιήσει τελικά, ενώ η διαδικασία μπορεί να αυτοματοποιηθεί για κάθε υπηρεσία.

Αν και υπήρχε ως ιδέα η χρήση ομότιμων οντοτήτων για την ανάπτυξη ενός τρόπου αναζήτησης δικτυακών υπηρεσιών, οι επιχειρήσεις οδηγήθηκαν τελικά στη χρήση του καταλόγου UDDI. Η αλήθεια είναι πως η τεχνολογία ομότιμων οντοτήτων είναι σχετικά καινούρια, ενώ οι διάφορες εφαρμογές των τελευταίων χρόνων αποτελούν ένα υβρίδιο μεταξύ κεντροποιημένων και κατανεμημένων υπηρεσιών.

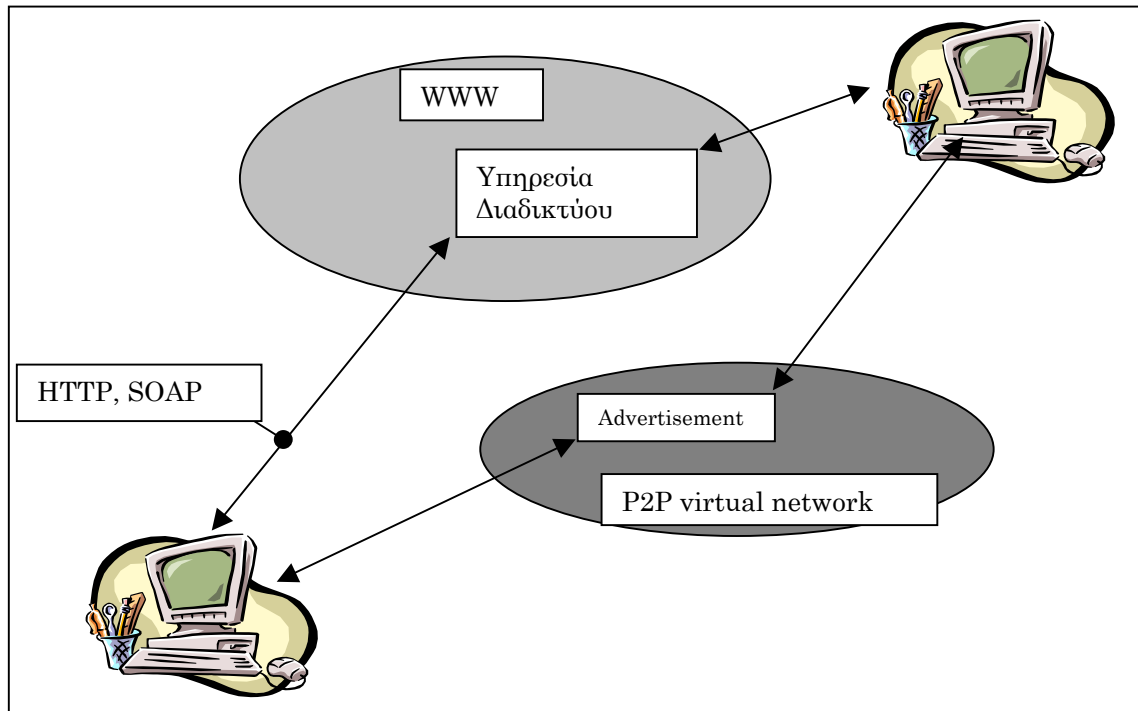
Η λύση UDDI είναι αρκετά καλή και υποστηρίζεται από πολλές και μεγάλες εταιρείες. Σε μια «παγκόσμια» βάση δεδομένων συγκρατούνται λεπτομερείς οδηγίες για τις υπηρεσίες, τις μεθόδους τους και τον τρόπο συναλλαγής μαζί τους. Αυτό όμως ίσως είναι περιττό σε έναν βαθμό γιατί οι υπηρεσίες διαδικτύου είναι αυτοπεριγραφόμενες, ένας από τους λόγους που τις έκανε και δημοφιλείς. Η γνώση της ύπαρξης μιας υπηρεσίας σχετικής με το αντικείμενο που επιθυμούμε είναι αρκετή τις περισσότερες φορές.

Με την πλατφόρμα Jxta έχουμε ένα δυνατό εργαλείο για την ανάπτυξη μιας πλατφόρμας αναζήτησης δικτυακών υπηρεσιών με χρήση ομότιμων οντοτήτων. Έχουμε τα πρωτόκολλα με τα οποία μπορούμε να αναπτύξουμε την πλατφόρμα ομότιμων οντοτήτων και η διαδικασία της αναζήτησης εμπίπτει στην χρησιμοποίηση του πρωτοκόλλου αναζήτησης διαφημίσεων, αρκεί να αναπτύξουμε μια δομή πληροφορίας που παριστά μια υπηρεσία διαδικτύου με μια διαφήμιση. Οι επιχειρήσεις μπορούν να χρησιμοποιήσουν αυτή τη δομή για να δηλώσουν την προσφορά μια υπηρεσίας διαδικτύου, ενώ οι καταναλωτές μπορούν να αναζητούν τέτοιες δομές. Αφού βρεθεί η διεύθυνση της υπηρεσίας μεταφορτώνεται στον τοπικό δίσκο το αρχείο που περιγράφει την υπηρεσία με την WSDL και δημιουργείται αυτόματα ο τρόπος κλήσης της. Βέβαια η κλήση της υπηρεσίας γίνεται μέσω SOAP και έξω από το δίκτυο ομότιμων οντοτήτων όπου βρέθηκε η πληροφορία ύπαρξής της.

2. Επισκόπηση Αρχιτεκτονικής

Η αρχιτεκτονική της πλατφόρμας απεικονίζεται στο Σχήμα 2.1. Σκοπίμως δεν έχουμε σημειώσει τον ρόλο του κάθε υπολογιστή. Εκ πρώτης όψεως φαίνεται πως ο πελάτης της υπηρεσίας είναι αυτός που χρησιμοποιεί το πρωτόκολλο HTTP για μεταφορά και το SOAP να συναλλάσσεται με την υπηρεσία διαδικτύου. Ο καθένας τους, όμως, θα μπορούσε να δρα και ως πελάτης και ως προμηθευτής υπηρεσιών διαδικτύου. Οπότε ο ρόλος που παίζει το κάθε

μηχάνημα εξαρτάται από το αν παρέχει κάποια υπηρεσία ή αν αναζητά να καταναλώσει υπηρεσίες.



Σχήμα 2.1 Πλατφόρμα αναζήτησης υπηρεσιών διαδικτύου με χρήση P2P

Από τα προλεγόμενα παρουσιάζεται ότι κάθε ομότιμη οντότητα αντιστοιχεί σε έναν προμηθευτή ή έναν καταναλωτή υπηρεσιών ενώ μια συσκευή, όπως ένας υπολογιστής λ.χ., μπορεί να εκτελεί πολλές ομότιμες οντότητες ταυτόχρονα, η καθεμία από τις οποίες θα μπορούσε με τη σειρά της να προσφέρει ή να καταναλώνει πληροφορίες για πολλαπλές υπηρεσίες διαδικτύου.

Οι διαδικασίες που αφορούν λοιπόν μια υπηρεσία Διαδικτύου συνοψίζονται τώρα στα εξής βήματα:

Προμηθευτής

Δημιουργεί την υπηρεσία, την θέτει έτοιμη προς χρήση.

Ανακοινώνει την ύπαρξή της στο P2P δίκτυο με μια διαφήμιση

Πελάτης

Αναζητεί την ύπαρξη υπηρεσιών, για τις ανάγκες του – διαφημίσεις

Χρησιμοποιεί την υπηρεσία

Βέβαια ο έλεγχος πρέπει να γίνεται από την εφαρμογή διεπαφής χρήστη. Δηλαδή, από τη μεριά του προμηθευτή της υπηρεσίας, πρέπει να γίνεται έλεγχος για το αν η διαφήμιση που δημιουργείται πράγματι αντιστοιχεί σε μια υπηρεσία διαδικτύου. Ενώ από τη μεριά του καταναλωτή της υπηρεσίας πρέπει να γίνει ο έλεγχος για τον αν η πρόσφατη διαφήμιση, που ανακαλύφθηκε, αντιστοιχεί σε μια υπηρεσία διαδικτύου προτού γίνει η μεταφόρτωση του WSDL αρχείου και κάθε άλλη παραπέρα ενέργεια για χρήση της υπηρεσίας.

2.1 Δημιουργία διαφήμισης για δικτυακές υπηρεσίες

Έχουμε ήδη αναφέρει πως στην πλατφόρμα Jxta η διακίνηση των πληροφοριών γίνεται μέσω διαφημίσεων, XML έγγραφα που διακινούνται μεταξύ των ομότιμων οντοτήτων. Σκοπός μιας διαφήμισης, για μας, είναι η ανακοίνωση της ύπαρξης μιας υπηρεσίας διαδικτύου. Φυσικά έχουμε μια «τεχνική» περιγραφή της υπηρεσίας από το WSDL αρχείο της. Αυτή όμως η περιγραφή δεν αρκεί για να περιγράψει την λειτουργία της υπηρεσίας, δηλαδή το τι αυτή προσφέρει. Αυτό αποτελεί μεταπληροφορία για μια συγκεκριμένη υπηρεσία. Το wsdl δεν περιλαμβάνει από μόνο του αυτή πληροφορία. Για τον λόγο αυτό πρέπει να αναπτύξουμε μια δομή, η οποία να περιέχει αρκετή μεταπληροφορία για μια υπηρεσία, τι προσφέρει, το που βρίσκεται κτλ. Ως εκ τούτου:

- Κάθε διαφήμιση υπηρεσίας διαδικτύου θα πρέπει να περιλαμβάνει αρκετή πληροφορία για την υπηρεσία για την εύκολη αναζήτηση της στο δίκτυο. Ο κάθε χρήστης που αναζητά υπηρεσίες συνήθως δεν γνωρίζει τίποτε για αυτές, εκτός ίσως λέξεις κλειδιά, για αυτό που επιθυμεί να κάνει. Άρα μια διαφήμιση πρέπει να περιέχει μια περιγραφή της υπηρεσίας.
- Κάθε διαφήμιση πρέπει να περιλαμβάνει κάποια στοιχεία για αυτόν που την εξέδωσε. Μπορεί να μην περιέχει αυτές τις πληροφορίες αν π.χ. δεν είναι διαθέσιμες, αλλά σε αυτή την περίπτωση θα πρέπει να υπάρχει η δυνατότητα παροχής τους από τον εκδότη της.
- Πρέπει να έχει ένα μοναδικό χαρακτηριστικό ώστε ο εκδότης της να την ξεχωρίζει ανάμεσα σε πολλές διαφημίσεις,
- Ένα χαρακτηριστικό που να προσδιορίσει μια κατηγορία όμοιων υπηρεσιών, επιτρέποντας π.χ. την αναφορά σε μια κοινή διεπαφή
- Πρέπει να έχει σχετικά μικρό μέγεθος, δεδομένου ότι σε ένα δίκτυο θα μετακινούνται συνεχώς τέτοιες διαφημίσεις, περισσότερες πληροφορίες μπορούν να αναζητηθούν στον εκδότη της.

Λαμβάνοντας υπόψη τα παραπάνω, η δομή μιας διαφήμισης για την ύπαρξη μιας υπηρεσίας, που προτείνουμε, φαίνεται στο παρακάτω σχήμα.

```
<xs:element name="WebServiceAdv" type="jxta:WebServiceAdv"/>
<xs:complexType name="WebServiceAdvertisement">
  <xs:element name="ServiceID" type="jxta:JXTAID" />
  <xs:element name="InfoPipeID" type="jxta:JXTAID" />
  <xs:element name="ProviderID" type="jxta:JXTAID" />
  <xs:element name="ServiceName" type="xs:String" maxOccurs="1"/>
  <xs:element name="ServiceDesc" type="xs:String" maxOccurs="1"/>
  <xs:element name="ServiceURL" type="xs:anyURI" maxOccurs="1"/>
  <xs:element name="ServiceVersion" type="xs:String" maxOccurs="1"/>
  <xs:element name="ProviderName" type="xs:String" maxOccurs="1"/>
  <xs:element name="ProviderDesc" type="xs:String" maxOccurs="1"/>
  <xs:element name="ProvidersURL" type="xs:anyURI" maxOccurs="1"/>
</xs:complexType>
```

Σχήμα 2.2 Τρόπος σχηματισμού διαφημίσεων για υπηρεσίες διαδικτύου

Τα στοιχεία είναι τα εξής:

ServiceID: Είναι ένα CodatID. Στην υλοποίηση της πλατφόρμας μας το στοιχείο αυτό χρησιμοποιείται για να προσδιορίσει μοναδικά τη διαφήμιση μιας υπηρεσίας, π.χ. όταν ένας πελάτης ρωτά για λεπτομέρειες στον προμηθευτή πρέπει ο προμηθευτής να καταλάβει σε ποια διαφήμιση αναφέρεται. Το στοιχείο αυτό στην ουσία προσδιορίζει ένα Codat. Τα Codats στην Jxta είναι αντικείμενα που περιλαμβάνουν και κώδικα και δεδομένα. Σε μια πιο αφηρημένη προσέγγιση τα Codats αυτά μπορεί να προσδιορίσουν ένα κοινό πρότυπο που ακολουθεί η υπηρεσία διαδικτύου. Τα Codats πρέπει να δημιουργούνται από την πλατφόρμα μας αφού εξεταστεί η υπηρεσία και διαπιστωθεί πως δεν υπάρχει άλλο στο δίκτυο, εάν υπάρχει τότε γίνεται αναφορά σε αυτό, ενώ για τον προσδιορισμό της διαφήμισης από τον προμηθευτή θα πρέπει να καταγράφεται πάντα το όνομα και η έκδοση.

InfoPipeID: Πρόκειται για ένα PipeID. Στην ουσία προσδιορίζει έναν σωλήνα από τον οποίο μπορούν να αναζητηθούν πληροφορίες για την υπηρεσία. Το στοιχείο αυτό πρέπει να υπάρχει σε κάθε διαφήμιση.

ProviderID: Πρόκειται για ένα PeerID. Προσδιορίζει μια ομότιμη οντότητα και άρα έναν προμηθευτή υπηρεσιών διαδικτύου. Η τιμή αυτού του στοιχείου πρέπει να παραχωρείται αυτόματα από το σύστημα όταν εκδότης της διαφήμισης θα προσπαθήσει να διαφημίσει την υπηρεσία του και να υπάρχει σε κάθε διαφήμιση.

ServiceName: Πρόκειται για ένα όνομα, το όνομα της υπηρεσίας. Ο κάθε πάροχος υπηρεσιών δίνει το δικό του όνομα στις υπηρεσίες του για να τις ξεχωρίσει. Το στοιχείο αυτό μπορεί να μην υπάρχει σε μια διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα επικοινωνίας.

ServiceDesc: Πρόκειται για μια περιγραφή της υπηρεσίας. Μια καλή περιγραφή της υπηρεσίας είναι ο προσδιορισμός λέξεων κλειδιά, οι οποίες είναι χαρακτηριστικές για την υπηρεσία. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα.

ServiceURL: Πρόκειται για την ηλεκτρονική διεύθυνση από την οποία μπορεί να προσεγγιστεί η υπηρεσία. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα. Σε αυτή της περίπτωση ο προμηθευτής πρέπει να την παρέχει ώστε ο πελάτης να έχει τελικά πρόσβαση στην υπηρεσία.

ServiceVersion: Πρόκειται για τον αριθμό έκδοσης της υπηρεσίας. Σε συνδυασμό με το όνομα της υπηρεσίας ο προμηθευτής προσδιορίζει επακριβώς της υπηρεσία του. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα. Σε αυτή της περίπτωση ο προμηθευτής μπορεί να την παρέχει ώστε ο πελάτης να ξεχωρίσει από προηγούμενες εκδόσεις.

ProviderName: Πρόκειται για το όνομα του προμηθευτή, την επωνυμία του παρόχου της υπηρεσίας. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα.

ProviderDesc: Πρόκειται για την περιγραφή του παρόχου της υπηρεσίας. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα.

ProvidersURL: Πρόκειται για την ηλεκτρονική διεύθυνση του παρόχου της υπηρεσίας. Μπορεί να μην υπάρχει στην διαφήμιση αλλά μπορεί να αναζητηθεί μέσω του σωλήνα.

Για διάφορους λόγους μπορεί να μην αναγράφονται κάποια στοιχεία σε μια διαφήμιση. Όταν όμως ένας προμηθευτής ερωτηθεί για μια συγκεκριμένη διαφήμιση θα πρέπει να είναι σε θέση να «ξεκαθαρίσει» τις απορίες των πελατών του σχετικά με την υπηρεσία. Για λόγους ασφαλείας όμως μπορεί να επιλέξει να μην παραχωρήσει κάποια στοιχεία. Αυτό όμως εντάσσεται στην πολιτική που ακολουθεί η ομότιμη οντότητα.

2.2 Ανταλλαγή μηνυμάτων για λήψη πληροφοριών

Όταν μια διαφήμιση είναι ελλιπής σε πληροφορίες μια ομότιμη οντότητα μπορεί να αναζητήσει αυτές από τον εκδότη της διαφήμισης. Ο τρόπος επικοινωνίας μεταξύ του πρέπει να γίνει έτσι ώστε να γίνει κατανοητό ένα από τα συμβαλλόμενα μέρη, με ένα πρωτόκολλο ανταλλαγής μηνυμάτων. Τα μηνύματα που ανταλλάσσονται για την επικοινωνία φαίνονται παρακάτω.

```
<xs:element name="ServiceAdvRequest" type="jxta:ServiceAdvRequest"/>
<xs:simpleType name="MsgType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Query"/>
    <xs:enumeration value="Answer"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ServiceAdvRequest">
  <xs:element name="WebServiceAdv" type="jxta:WebServiceAdv"/>
  <xs:element name="PeerIdentity" type="jxta:Cred"/>
  <xs:element name="Pipe" type="jxta:PipeAdv" maxOccurs="1"/>
</xs:complexType>
```

Σχήμα 2.3 Τρόπος σχηματισμού μηνύματος αίτησης προς το προμηθευτή

Όπου τα στοιχεία είναι τα εξής:

MsgType: Καθορίζει αν πρόκειται για ερώτηση ή αν πρόκειται για απάντηση

WebServiceAdv: Είναι η διαφήμιση για την οποία ζητούνται πληροφορίες όταν η διαφήμιση είναι ελλιπής και η διαφήμιση με όλα τα στοιχεία της όταν είναι ολοκληρωμένη.

PeerIdentity: Στο μήνυμα ερώτησης, καταγράφεται στο στοιχείο αυτό η ταυτότητα αυτού που ζητά πληροφορίες και στο μήνυμα απάντησης καταγράφονται τα διαπιστευτήρια του προμηθευτή της απάντησης. Πρέπει να σημειώσουμε εδώ ότι η ταυτότητα είναι διαφορετική από την διαφήμιση ομότιμης οντότητας.

Pipe: Είναι μια διαφήμιση σωλήνα. Χρησιμοποιείται μόνο στο μήνυμα ερώτησης ώστε ο προμηθευτής, ο οποίος ρωτάται, να γνωρίσει που θα απαντήσει.

2.3 Πολιτικές ασφαλείας και ποιότητα υπηρεσίας στην Java υλοποίηση της Jxta

Σε ένα δίκτυο ομότιμων οντοτήτων η κάθε οντότητα έχει πρόσβαση στη δημιουργία διαφημίσεων και την αποστολή μηνυμάτων. Για να εξασφαλιστεί κάποιος βαθμός ασφάλειας και ποιότητας υπηρεσίας θα πρέπει υιοθετηθούν κάποιες πολιτικές διασφάλισης ποιότητας.

Η ανάπτυξη διαπιστευτηρίων αποτελούν το πρωταρχικό βήμα εγκαθίδρυσης ασφάλειας και ταυτότητας επικοινωνίας μέσα στην jxta πλατφόρμα. Τα πακέτα `net.jxta.membership` και `net.jxta.credential` αποτελούν το μηχανισμό για την επίτευξη αυτού του στόχου. Το πρώτο περιλαμβάνει τις λειτουργίες για λήψη ταυτότητας, δήλωσης ταυτότητας, και δράσης ως διαπιστευμένο ων, ενώ στο δεύτερο ορίζεται η έννοια των διαπιστευτηρίων. Συγκεκριμένα στο δεύτερο πακέτο υπάρχει η προνομιούχος λειτουργία (Privileged Operation) η οποία αποτελεί ένα προσδιοριστικό για μια λειτουργία η της οποίας χρήση είναι περιορισμένη. Η προνομιούχος λειτουργία αντιπροσωπεύεται ως πιστοποιητικό έτσι ώστε οι χρήστες και οι προμηθευτές να μπορούν να αποτρέψουν την αναρμόδια δημιουργία των προνομίων. Κάθε λειτουργία συνδέεται με έναν πάροχο (offerer). Αυτός είναι η ταυτότητα αυτού που δημιουργήσε τη λειτουργία.

Η λήψη ταυτότητας γίνεται κατά την εγγραφή μιας ομότιμης οντότητας στην ομάδα ομότιμων οντοτήτων που επιθυμεί. Έτσι η κάθε ομότιμης οντότητα μπορεί να σημειώνει σε μια λίστα πρόσβασης με ποιους επιθυμεί να συναλλάσσεται και ποιους όχι.

Η κάθε ομότιμη οντότητα ακολουθεί την δική του πολιτική ασφαλείας. Είναι λοιπόν στο χέρι της να επιλέξει να συναλλάσσεται με ομότιμες οντότητες χωρίς ταυτότητα. Σίγουρα δεν απαντά σε μηνύματα στα οποία δεν αναγράφεται η διαφήμιση, ή αυτή δεν είναι δική του διαφήμιση, ή δεν αναγράφεται η διαφήμιση σωλήνα για να απαντήσει.

Φυσικά επιχειρήσεις επιθυμούν να διαφημίσουν το προϊόν τους σε μεγαλύτερη μερίδα του κοινού, αλλά υπάρχουν και κακόβουλοι οι οποίοι μπορεί να διαφημίσουν υπηρεσίες που δεν υπάρχουν ή ακόμα θα προσπαθήσουν «πνίξουν» το δίκτυο, διακινώντας μεγάλους όγκους από άχρηστες διαφημίσεις. Εδώ εγείρονται διάφορα προβλήματα για το κατά πόσο έγκυρα είναι οι διαφημίσεις που ανακαλύπτονται και αν πράγματι αυτές αντιστοιχούν σε πραγματικές υπηρεσίες διαδικτύου. Για τους λόγους αυτούς λοιπόν πρέπει να γίνεται έλεγχος πριν την χρήση μιας διαφήμισης πρέπει να γίνεται έλεγχος, ταυτότητας του προμηθευτή και ύπαρξης της υπηρεσίας.

Ακόμα η πλατφόρμα Jxta προβλέπει ήδη τη δυνατότητα λήξης των διαφημίσεων, για να μην περιφέρονται άσκοπα στο δίκτυο. Με αυτόν τον τρόπο η διαδικασία της ανακάλυψης ή της δημοσίευσης θα πρέπει να επαναλαμβάνεται περιοδικά από τις διάφορες ομότιμες οντότητες.

Η πλατφόρμα μας είναι σε θέση να εξασφαλίσει λύσεις για τέτοια θέματα ενώ ταυτόχρονα να είναι αποδοτική και ευέλικτη.

3. Υλοποίηση στην Java

Η υλοποίηση της πλατφόρμας έγινε σε Java. Συγκεκριμένα υπάρχουν δύο ήδη κλάσεων από της οποίες η μία για τον προμηθευτή (BusinessServer), και η άλλη τον καταναλωτή (ServiceConsumer). Ο προμηθευτής συνήθως διαθέτει υπηρεσίες που θέλει να διαφημίσει ενώ ο καταναλωτής που εισέρχεται σε στο δίκτυο επιθυμεί να καταναλώσει υπηρεσίες τις οποίες όμως δεν γνωρίζει εκ των προτέρων. Έτσι λοιπόν, ο κάθε καταναλωτής, επιθυμεί να ανακαλύψει υπηρεσίες με βάση κάποιες λειτουργίες που θέλει η υπηρεσία να του προσφέρει. Και οι δύο αυτές κλάσεις αποτελούν μια ομότιμη οντότητα σε ένα δίκτυο ομότιμων οντοτήτων. Ο ρόλος που επιλέγει κάποιος εξαρτάται από το εάν προσφέρει ή καταναλώνει υπηρεσίες.

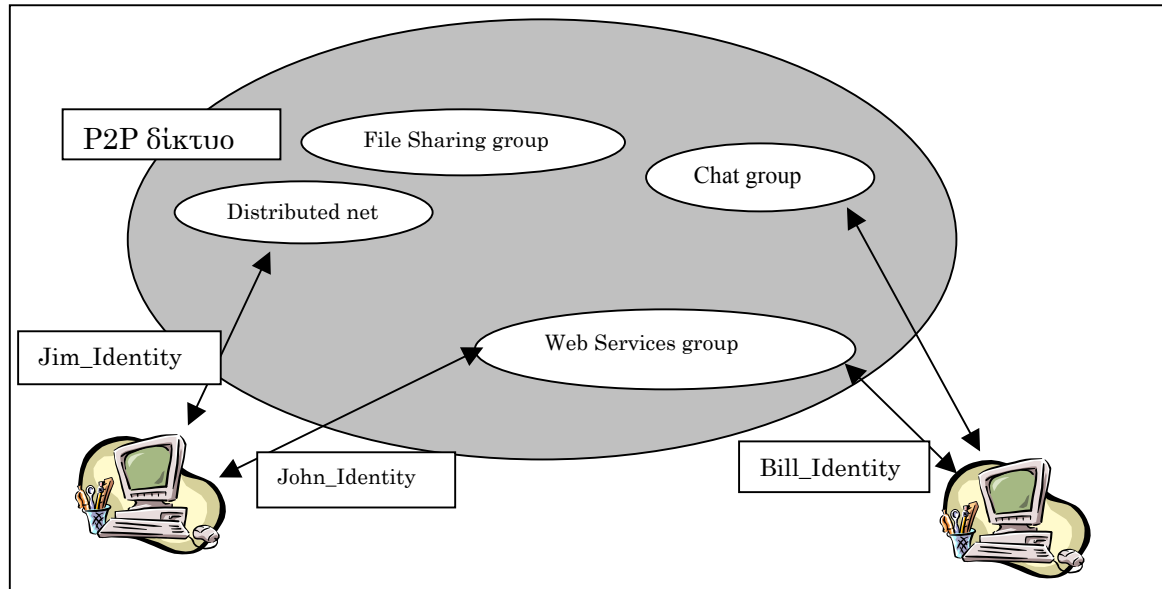
Οι λειτουργίες που πρέπει να εκτελεί κάθε ομότιμη οντότητα στο δίκτυο είναι οι εξής:

Προμηθευτής

- Αναζητεί μια συγκεκριμένη ομάδα ομότιμων οντοτήτων σε όλο το P2P δίκτυο.
- Εγγράφεται στο δίκτυο, αποκτώντας την δική του ταυτότητα.
- Διαφημίζει υπηρεσίες
- Αναμένει μηνύματα για διευκρινίσεις σε διαφημίσεις που εξέδωσε
- Στέλνει απαντήσεις σε ερωτήσεις διευκρίνησης

Καταναλωτής

- Αναζητεί μια συγκεκριμένη ομάδα ομότιμων οντοτήτων σε όλο το P2P δίκτυο με βάση κάποια γενικά χαρακτηριστικά των υπηρεσιών που επιθυμεί
- Εγγράφεται στη συγκεκριμένη ομάδα, αποκτώντας την δική του ταυτότητα
- Αναζητά υπηρεσίες
- Στέλνει τυχόν μηνύματα για διευκρινίσεις για υπηρεσίες
- Λαμβάνει απαντήσεις σε ερωτήσεις διευκρίνησης διαφήμισης



Σχήμα 3.1 Τρόπος συναλλαγής με ομάδες ομότιμων οντοτήτων. Αφού ανακτήσουν την ταυτότητα εγγράφονται στην ομάδα και έπειτα μπορούν να εκτελέσουν τις άλλες λειτουργίες

3.1 Αναζήτηση ομάδας

Σαν πρώτο βήμα, και για τον προμηθευτή και για τον καταναλωτή, γίνεται αναζήτηση της κατάλληλης ομάδας ομότιμων οντοτήτων στην οποία μπορούν να αναζητηθούν πληροφορίες για υπηρεσίες διαδικτύου. Η ομάδα αυτή ορίζει εξάλλου και το είδος της πληροφορίας που διακινείται στην ομάδα. Η αναζήτηση μπορεί να γίνει με λέξεις κλειδιά για αυτό που επιθυμούμε να κάνουμε, π.χ. *υπηρεσίες διαδικτύου, στιγμιαία μηνύματα, καταμετρημένο υπολογισμό κτλ.* Σκοπός της δημιουργίας μια ομάδων είναι να περιοριστεί το εύρος αναζήτησης, αφού σε κάθε ερώτηση αναζήτησης ρωτούνται μόνο τα μέλη της ομάδας και όχι τα μέλη όλου του δικτύου.

Η πλατφόρμα JXTA μας δίνει την δυνατότητα να δημιουργήσουμε φωλιασμένες ομάδες, παιδιά υπερομάδων. Κατ' αυτόν το τρόπο μπορούμε να δημιουργήσουμε μια δενδρική δομή, ξεκινώντας από την προεπιλεγμένη ομάδα (NetPeerGroup), να δημιουργήσουμε την σχετική ομάδα με τις υπηρεσίες διαδικτύου (WebServicesNet), να δημιουργήσουμε την υποομάδα της (WebServicesProvidersNet) στην οποία έχουν πρόσβαση μόνο οι προμηθευτές δικτυακών υπηρεσιών κ.ο.κ. Η δημιουργία των ομάδων είναι πολύ σημαντική ιδιότητα γιατί κατ' αυτόν το τρόπο θα μπορούσε σε κάθε ομάδα ομότιμων οντοτήτων να διακινείται διαφορετικό είδος πληροφορίας. Εξάλλου η πρόσβαση της κάθε ομότιμης οντότητας περιορίζεται στα όρια μια ομάδας.

Στην υλοποίησή μας δεν προχωρήσαμε την ιδέα του φωλιάσματος των ομάδων σε μεγάλο βάθος. Δημιουργήσαμε μια υποομάδα (WebServicesNet) της προεπιλεγμένης ομάδας (NetPeerGroup) στην οποία οι προμηθευτές και οι καταναλωτές ανακοινώνουν και αναζητούν αντίστοιχα διαφημίσεις για την ύπαρξη υπηρεσιών διαδικτύου.

Η αναζήτηση γίνεται χρησιμοποιώντας το πρωτόκολλο αναζήτησης, μέσω της υπηρεσίας αναζήτησης, όπως αυτή είναι υλοποιημένη στη Java για την προεπιλεγμένη ομάδα ομότιμων οντοτήτων. Εάν δεν υπάρχει τέτοια ομάδα, ο προμηθευτής πρέπει να την δημιουργήσει, ώστε να παρέχει την πληροφόρηση των υπηρεσιών στους υποψήφιους πελάτες, ενώ ο καταναλωτής θα διαπιστώσει πως δεν υπάρχουν δηλωμένες υπηρεσίες διαδικτύου στη συγκεκριμένη ομάδα.

3.2 Εγγραφή

Σαν δεύτερο βήμα, και για τον προμηθευτή και για τον καταναλωτή, είναι η εγγραφή στην κατάλληλη ομάδα ομότιμων οντοτήτων. Η διαδικασία αυτή αφορά την απόδοση διαπιστευτηρίων στην ομότιμη οντότητα που εγγράφεται τελικά στην ομάδα. Τα διαπιστευτήριά μιας ομότιμης οντότητας είναι στην ουσία η ταυτότητά της, η εγγύηση ότι έχει το δικαίωμα να υπάρχει ως μέλος στη συγκεκριμένη ομάδα ομότιμων οντοτήτων. Αυτό είναι αναγκαίο ώστε να αποφεύγονται σωρεία προβλημάτων ασφαλείας γιατί ο καθένας θα γνωρίζει με ποιόν συναλλάσσεται.

Στην υλοποίησή μας η ομάδα στην οποία εγγράφεται ο προμηθευτής καταγράφονται τα διαπιστευτήρια κάποιων προνομιούχων ομάδων χρηστών. Καθένας ο οποίος δεν έχει διαπιστευτήρια, αποκτά την *nobody* ταυτότητα κατά την εγγραφή του στο δίκτυο.

3.3 Ανακοίνωση – αναζήτηση διαφήμισης

Μετά την εγγραφή του κάθε ομότιμης οντότητας στην ομάδα, ο κάθε προμηθευτής έχει την ικανότητα να ανακοινώνει τις διαφημίσεις του, ενώ ο κάθε καταναλωτής έχει την ικανότητα να αναζητά διαφημίσεις. Η αναζήτηση διαφημίσεων από τους καταναλωτές βασίζεται σε λέξεις κλειδιά. Οι λειτουργίες αυτές γίνονται μέσω του πρωτοκόλλου αναζήτησης, όπως αυτό είναι υλοποιημένο σαν υπηρεσία στην Java υλοποίηση της Jxta. Τα πρωτόκολλα της Jxta μας εξασφαλίζουν αναζήτηση πέρα από το τοπικό μας δίκτυο, στις «*άκρες*» του Διαδικτύου και πίσω από εταιρικούς τοίχους ηλεκτρονικής προστασίας. Άρα έχουμε ένα αποτελεσματικό τρόπο ανακάλυψης.

3.4 Διευκρίνιση διαφήμισης

Η διευκρίνιση διαφημίσεων γίνεται μόνο σε οντότητες με ταυτότητα και από οντότητες ταυτότητα. Η διευκρίνιση διαφημίσεων από οντότητες χωρίς ταυτότητα, είναι επικίνδυνο επειδή μπορεί να αποσκοπεί στην κατανάλωση εύρους ζώνης ενός από τους δύο συμμετέχοντες στην επικοινωνία.

Η ανταλλαγή μηνυμάτων γίνεται μέσω σωλήνων επικοινωνίας. Η δημιουργία σωλήνων επικοινωνίας γίνεται από την υπηρεσία δέσμευσης σωλήνων με την παροχή μιας διαφήμισης σωλήνα. Ο τρόπος που δημιουργείται μια διαφήμιση σωλήνα φαίνεται στο παρακάτω σχήμα:

Με στοιχεία:

- **Id:** Ο αριθμός ταυτότητας του σωλήνα
- **Type:** Τύπος σωλήνα, JxtaUnicast, JxtaUnicastSecure, JxtaPropagate
- **Name:** Ένα προαιρετικό όνομα

```
<xs:element name="PipeAdvertisement" type="jxta:PipeAdvertisement"/>

<xs:complexType name="PipeAdvertisement">
  <xs:element name="Id" type="jxta:JXTAID" />
  <xs:element name="Type" type="xs:string" />
  <xs:element name="Name" type="xs:string" />
</xs:complexType>
```

Σχήμα 3.2 Τρόπος δημιουργίας μιας διαφήμισης σωλήνα

Για να μην υπάρχει σύγχυση από την πλευρά των καταναλωτών, η διαφήμιση δημιουργείται με στοιχείο το *Id*, το οποίο το λαμβάνεται από την διαφήμιση της υπηρεσίας και συγκεκριμένα το στοιχείο *InfoPipeId*, και τον τύπο, ο οποίος είναι JxtaUnicast.

Η δέσμευση σωλήνα λοιπόν γίνεται από τη διαφήμιση που δημιουργείται τη στιγμή που ανακαλύπτεται η διαφήμιση της υπηρεσίας και η επικοινωνία με τον εκδότη γίνεται άμεσα με την ανταλλαγή μηνυμάτων με τη μορφή που ορίζονται παραπάνω.

3.5 Λειτουργία αναφοράς στην χρήσης της υπηρεσίας

Ανακαλύπτοντας μια διαφήμιση που ταιριάζει στις προτιμήσεις μας αυτό που κυρίως μας ενδιαφέρει είναι η πρόσβαση στην τοποθεσία της υπηρεσίας στο διαδίκτυο. Ένας χρήστης, αφού ανακαλύψει την υπηρεσία που τον ενδιαφέρει πρέπει να βρει κάποιο τρόπο να την χρησιμοποιήσει. Υπάρχει λοιπόν και μια άλλη κλάση (WSDL2JavaBuilder) για να εκτελέσει τη συγκεκριμένη εργασία. Συγκεκριμένα η κλάση αυτή εκτελεί τις παρακάτω εργασίες:

- Μετασχηματίζει το WSDL αρχείο σε πηγαίο Java κώδικα. Ο μετασχηματισμός αυτός γίνεται κατά τον τρόπο που η υπηρεσία ορίζει, με σκοπό τη δημιουργία του κώδικα πελάτη της υπηρεσίας.
- Μεταγλωττίζει το σε πηγαίο Java κώδικα σε αντικείμενο κώδικα, έτοιμο προς εκτέλεση.
- Δημιουργεί μια αναφορά στις μεθόδους της υπηρεσίας.
- Παρουσιάζει τις μεθόδους αυτές στο χρήστη και περιμένει την επιλογή του
- Εκτελεί τη μέθοδο που ο χρήστης επέλεξε.

Οι διαδικασίες που περιγράφονται παραπάνω είναι αρκετά πολύπλοκες για να τις χειριστεί μια και μόνο κλάση. Για το σκοπό αυτό δημιουργήθηκε μια άλλη κλάση (LocalDOMHandler), ένας αναλυτής που μας παρέχει όλες τις τεχνικές λεπτομέρειες (μεταβλητές εισόδου, εξόδου και τους τύπους τους στις μεθόδους της υπηρεσίας) αναλύοντας το WSDL αρχείο που μόλις μεταφορτώθηκε στον τοπικό δίσκο χρησιμοποιώντας JDOM. Φυσικά αυτές τι

πληροφορίες μπορούμε να τις αποκτήσουμε με την μέθοδο της αντανάκλασης που προσφέρει η Java για αναφορά σε κλάσεις και αντικείμενα κατά το χρόνο εκτέλεσης. Ο αναλυτής αυτός δημιουργήθηκε, κυρίως, για την περίπτωση η υπηρεσία έχει αναπτυχθεί στο τελικό σημείο πρόσβασής της μέσω περιγραφέας ανάπτυξης (WSDD). Σε αυτή την περίπτωση είναι δύσκολο να προσδιοριστεί το όνομα της υπηρεσίας κατά το μετασχηματισμό του WSDL αρχείου σε Java κώδικα γιατί μπορεί οι ενδείξεις για το όνομα της κλάσης που υλοποιεί τον πελάτη της υπηρεσίας να είναι λανθασμένες. Αυτή την δυσκολία άρει ο αναλυτής *LocalDOMHandler* που αναλύει το WSDL αρχείο που μεταφορτώνεται στον τοπικό δίσκο και ανακαλύπτει το πραγματικό όνομα της Java κλάσης που προκύπτει από το μετασχηματισμό, του WSDL αρχείου, στο πηγαίο κώδικα του πελάτη. Επιπλέον η ομάδα ανάπτυξης του AXIS εργάζεται στο να ανακαλύπτει τρόπους, ώστε να εμφυτεύει πληροφορίες σχετικά με τις υπηρεσίες στα WSDL έγγραφα [AXI-1]. Με δεδομένο κάτι τέτοιο, ένα πρόγραμμα που αναλύει το WSDL αρχείο και εξάγει αυτή την μεταπληροφορία σχετικά με την υπηρεσία κάνει το πρόγραμμα πελάτη πιο εύχρηστο. Παράλληλα με την παρουσίαση των μεθόδων, που κάνουμε εδώ, θα μπορούσαμε να παρουσιάσουμε και πληροφορίες σχετικά με τον σκοπό τους στην υπηρεσία.

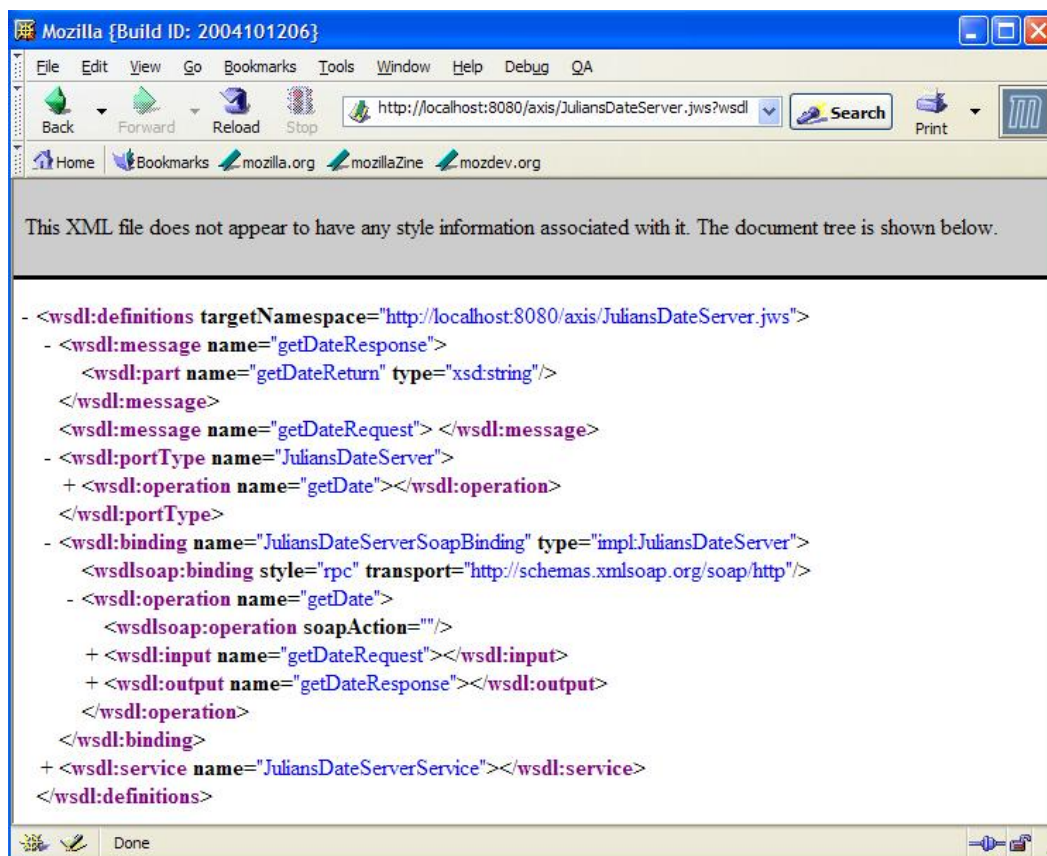
Έχοντας αυτό το γενικό τρόπο εκτέλεσης μιας υπηρεσίας μπορούμε να εκτελέσουμε οποιαδήποτε υπηρεσία διαδικτύου στην οποία έχουμε πρόσβαση στο WSDL αρχείο της.

3.6 Σύνθεση της πλατφόρμας και επίδειξη

Για να υλοποιήσουμε τις υπηρεσίες διαδικτύου υιοθετήσαμε το πρότυπο JWS (Java Web Services) και τον εύκολο τρόπο δημιουργίας υπηρεσίας διαδικτύου [AXI-1]. Χρησιμοποιήσαμε σαν SOAP μηχανή το AXIS (Apache EXtensible Interaction System, έκδοση 1.1)[AXI-1] και σαν εξυπηρετητή τον Tomcat servlet-JSP container (έκδοση 4.1.29)[TOM-1]. Το AXIS προσφέρει την δυνατότητα εύκολης δημιουργίας της υπηρεσίας, κατά την οποία ο πηγαίος Java κώδικας τοποθετείται στη μηχανή του AXIS και από κει, αυτόματα, μεταγλωττίζεται σε αντικείμενο κώδικα και δημιουργείται το WSDL αρχείο που περιγράφει την υπηρεσία.

Συγκεκριμένα αναπτύξαμε τρεις κλάσεις που προσφέρουν τρεις διαφορετικές υπηρεσίες, η μία αναφέρει την ημερομηνία και την ώρα (JulianDateServer), μια άλλη κοιτά σε μια βάση δεδομένων για να προσφέρει ενημέρωση σχετικά με τα εβδομαδιαία ζώδια (JulianSignService) και τέλος μια υπηρεσία που προσφέρει ενημέρωση σε τιμές μετοχών (StockQuoteService). Παρακάτω βλέπουμε τα WSDL έγγραφα που περιγράφουν τις υπηρεσίες, όπως όταν τις επισκεφτήκαμε με τον φυλλομετρητή ιστού Mozilla web browser.

Παρατηρώντας τη δομή των WSDL αυτών εγγράφων, αυτή ακολουθεί τη δομή που παρουσιάσαμε στο Μέρος-I παράγραφο 4.



Σχήμα 3.3 WSDL έγγραφο που περιγράφει την υπηρεσία για την ώρα. Βλέπουμε ότι δεν χρησιμοποιείται κάποιος ιδιαίτερος τύπος, επιπλέον στην υπηρεσία υπάρχει μόνο μια μέθοδος (operation name="getDate") που υλοποιεί την υπηρεσία και διακινεί δύο μηνύματα (Request – Response)

Η διαφήμιση που χρησιμοποιήθηκε για αυτήν στο JXTA δίκτυο είναι:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:WebServiceAdvertisement>
<jxta:WebServiceAdvertisement xmlns:jxta="http://jxta.org">
  <ServiceName> Date Server </ServiceName>
  <ServiceDesc>Provides information on current date and time. </ServiceDesc>
  <ServiceURL> http://localhost:8080/axis/JuliansDateServer.jws </ServiceURL>
  <InfoPipeID> urn:jxta:uuid-
D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
  </InfoPipeID>
  <ProviderID> urn:jxta:uuid-
59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
  </ProviderID>
  <ProviderName> Julian inc </ProviderName>
  <ProviderDesc> 'Julian inc' is a company focused on developing web services
  </ProviderDesc>
</jxta:WebServiceAdvertisement>
```

Σχήμα 3.4 Διαφήμιση που χρησιμοποιήθηκε για την υπηρεσία

Ο πελάτης χρησιμοποίησε λέξεις κλειδιά *date* και *time* για να την εντοπίσει στο JXTA δίκτυο και το πρόγραμμα διεπαφής χρήστη επέστρεψε:

Found web service adv:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:WebServiceAdvertisement>
<jxta:WebServiceAdvertisement xmlns:jxta="http://jxta.org">
  <ServiceName> Date Server </ServiceName>
  <ServiceDesc> Provides information on current date and time. </ServiceDesc>
  <ServiceURL> http://localhost:8080/axis/JuliansDateServer.jws </ServiceURL>
  <InfoPipeID>urn:jxta:uuid-
D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
  </InfoPipeID>
  <ProviderID>urn:jxta:uuid-
59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
  </ProviderID>
  <ProviderName> Julian inc </ProviderName>
  <ProviderDesc> 'Julian inc' is a company focused on developing web services
  </ProviderDesc>
</jxta:WebServiceAdvertisement>
```

Path created=localhost/axis/JuliansDateServer.jws

WSDL file successfully downloaded to: localhost/axis/JuliansDateServer.jws

WSDL file can be found to: localhost/axis/JuliansDateServer.jws/JuliansDateServer.wsdl.xml

WSDL2Java created classes to path: localhost/axis/JuliansDateServer.jws

Our service class name is: JuliansDateServer

Compile succeed, trying to run...

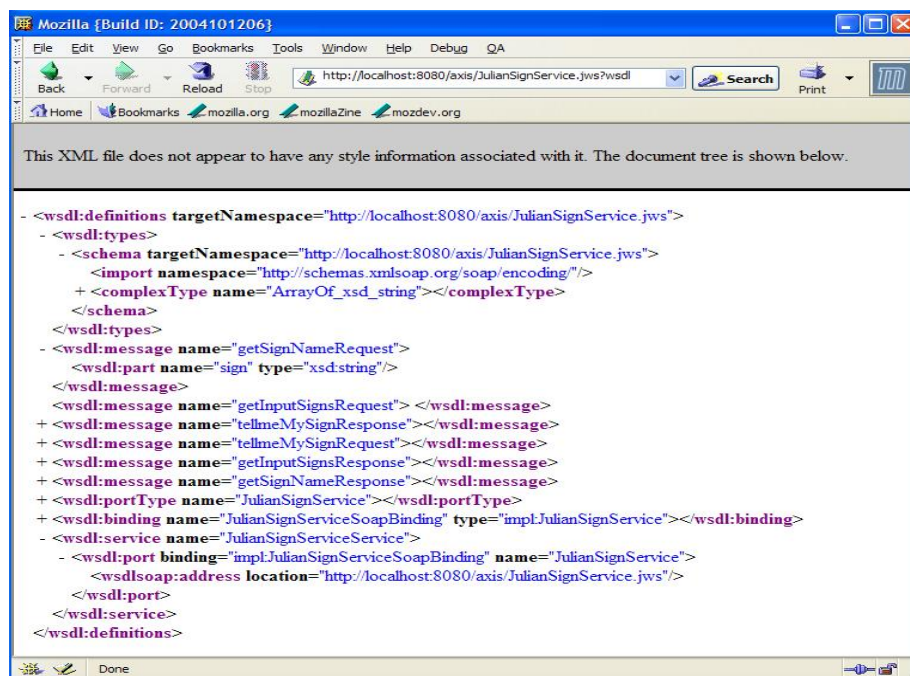
localhost.axis.JuliansDateServer.jws.JuliansDateServerServiceLocator

The service you're trying to use provide the following method (s):

getDate

Please enter the one you want to use!

Σχήμα 3.5 Έξοδος από το πρόγραμμα διεπαφής χρήστη.
Η υπηρεσία παροχής πληροφοριών σχετικά με τα εβδομαδιαία ζώδια.



Σχήμα 3.6 WSDL έγγραφο που περιγράφει την υπηρεσία για τα ζώδια. Βλέπουμε ότι χρησιμοποιείται ένας πολύπλοκος τύπος (types), στην κορυφή, ακόμα στην υπηρεσία υπάρχουν «έμφυτες» τρεις μέθοδοι και η καθεμία διακινεί δύο μηνύματα (Request – Response)

Η διαφήμιση που χρησιμοποιήθηκε στο JXTA δίκτυο φαίνεται στο παρακάτω σχήμα:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:WebServiceAdvertisement>
<jxta:WebServiceAdvertisement xmlns:jxta="http://jxta.org">
  <ServiceName> Star Signs Server </ServiceName>
  <ServiceDesc> Information on this week star signs. </ServiceDesc>
  <ServiceURL> http://localhost:8080/axis/JulianSignService.jws
</ServiceURL>
  <InfoPipeID>urn:jxta:uuid-
D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
</InfoPipeID>
  <ProviderID>urn:jxta:uuid-
59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
</ProviderID>
  <ProviderName> Julian inc </ProviderName>
  <ProviderDesc> 'Julian inc' is a company focused on developing web services
</ProviderDesc>
</jxta:WebServiceAdvertisement>

```

Σχήμα 3.7 Διαφήμιση που χρησιμοποιήθηκε για την υπηρεσία των ζωδίων

Το πρόγραμμα διεπαφής πελάτη χρησιμοποιήθηκε σαν λέξη κλειδί το “*Star Sign**” (το “***” είναι σύμβολο μπαλαντέρ και τοποθετείται για να σημάνει οτιδήποτε συμβολοσειρά κι αν ακολουθεί), για το όνομα της υπηρεσίας, που προφανώς γνώριζε και αυτό επέστρεψε:

```

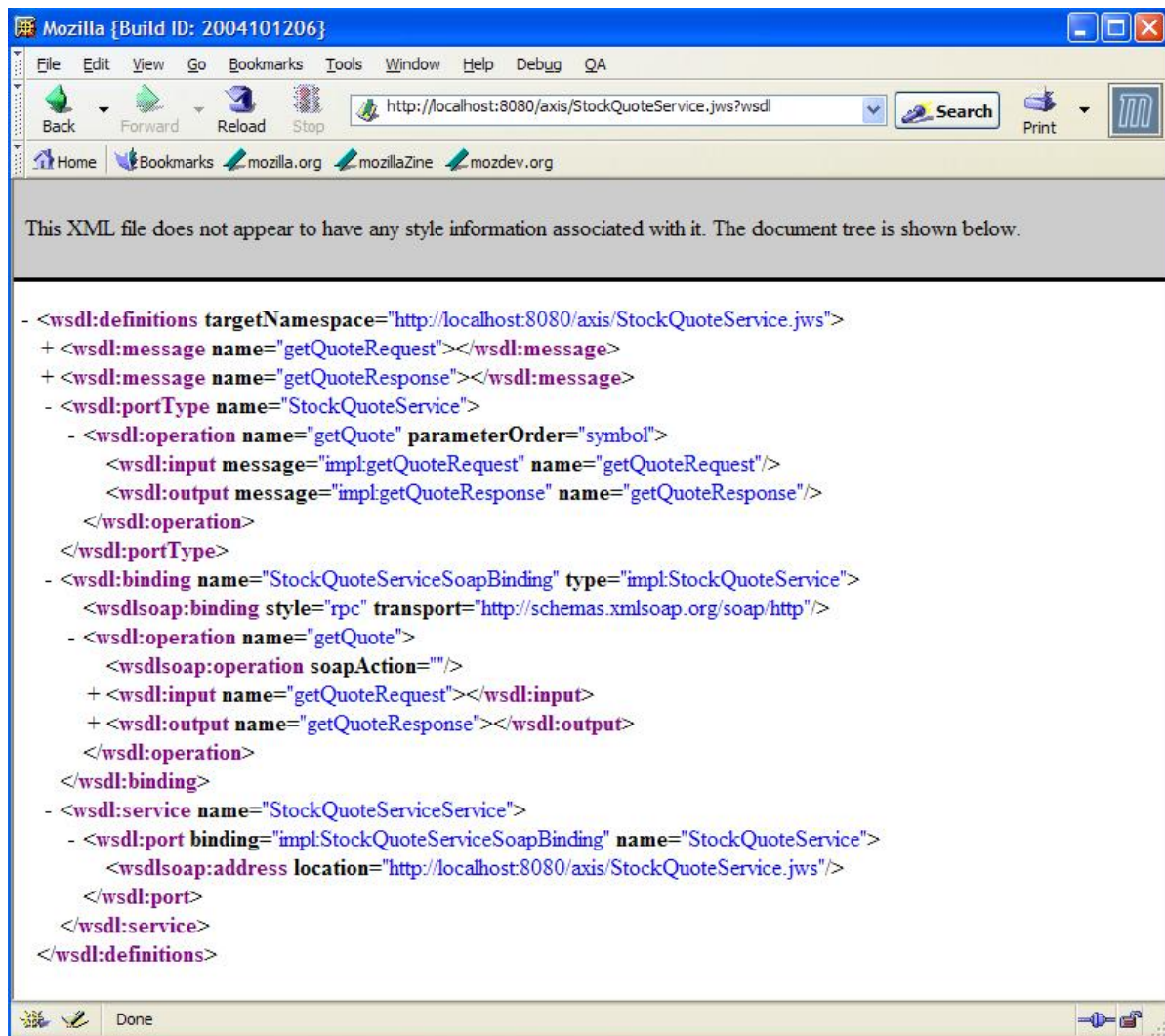
.Found ws:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:WebServiceAdvertisement>
<jxta:WebServiceAdvertisement xmlns:jxta="http://jxta.org">
  <ServiceName> Star Signs Server </ServiceName>
  <ServiceDesc> Information on this week star signs. </ServiceDesc>
  <ServiceURL> http://localhost:8080/axis/JulianSignService.jws </ServiceURL>
  <InfoPipeID>urn:jxta:uuid-
D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
</InfoPipeID>
  <ProviderID>urn:jxta:uuid-
59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
</ProviderID>
  <ProviderName> Julian inc </ProviderName>
  <ProviderDesc> 'Julian inc' is a company focused on developing web services
</ProviderDesc>
</jxta:WebServiceAdvertisement>

Path created=localhost/axis/JulianSignService_jws
WSDL file succesfully downloaded to: localhost/axis/JulianSignService_jws
WSDL file can be found to:localhost/axis/JulianSignService_jws/JulianSignService_wsdl.xml
WSDL2Java created classes to path:localhost/axis/JulianSignService_jws
Our service class name is:JulianSignService
Compile succeed, trying to run...
localhost.axis.JulianSignService_jws.JulianSignServiceServiceLocator
The service you're trying to use provide the following method (s):
    getInputSigns
    getSignName
    tellmeMySign
Please the one you want to use!
tellmeMySign
Building a request:
Please enter a String
Pis
Method 'tellmeMySign' is ready to be invoked.
We got the following response:
    Phishes: You are the luckiest of all. This week signs love you!

```

Σχήμα 3.8 Αλληλεπίδραση πελάτη με την υπηρεσία διαδικτύου

Τέλος η υπηρεσία πληροφόρησης μετοχών περιγράφεται παρακάτω:



Σχήμα 3.9 WSDL έγγραφο που περιγράφει την υπηρεσία ενημέρωσης σχετικά με μετοχές. Βλέπουμε ότι δεν χρησιμοποιείται κάποιος πολύπλοκος τύπος (type), επιπλέον στην υπηρεσία υπάρχουν έμφυτη μια μέθοδος και διακινεί δύο μηνύματα (Request – Response)

Το πρόγραμμα διεπαφής του προμηθευτή της υπηρεσίας, κατά τον χρόνο εκτέλεσης φαίνεται στο σχήμα 3.10. Το πρόγραμμα παρουσιάζει την ομάδα των ομότιμων οντοτήτων στην οποία εγγράφεται ή δημιουργεί (αν είναι ο πρώτος που εισέρχεται στην ομάδα) ώστε να προσφέρει την πληροφόρηση μέσω διαφημίσεων. Βέβαια στο σημείο αυτό πρέπει να εξασφαλίζεται ότι ο πάροχος της υπηρεσίας θα έχει πρώτα δημιουργήσει και αναπτύξει την υπηρεσία του στον δικτυακό του τόπο προτού την ανακοινώσει στο δίκτυο JXTA.

Ενώ το πρόγραμμα διεπαφής πελάτη φαίνεται στο σχήμα 3.11.

```

Attempting to Discover the Web Services PeerGroup
Found the Web Services PeerGroup advertisement & joined that group
Web Services group (WebServicesNet) is on-line.
This is its advertisement:
jxta:PGA :
  GUID : urn:jxta:uuid-D1D77D9D0B0B4FFBB4D293DFB60C5AD602
  MSID : urn:jxta:uuid-DEADBEEFDEAFBABAFAFEEDBABE000000010306
  Name : WebServicesNet
  Desc : WebServicesNet, a group suitable for dealing with web services.

Creating a web service advertisement and a pipe advertisement...
Publishing a web service advertisement:
jxta:WebServiceAdvertisement ( xmlns:jxta="http://jxta.org" ) :
  ServiceName : Stock Exchange Service
  ServiceDesc : Providing daily information on stock signs
  ServiceURL : http://localhost:8080/axis/StockQuoteService.jws
  InfoPipeID : urn:jxta:uuid-
D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
  ProviderID : urn:jxta:uuid-
59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
  ProviderName : Julian inc
  ProviderDesc : 'Julian inc' is a company focused on developing web services
  ProvidersURL : http://julianinc.com

Creating input pipe to handle client's request.->Done!

publishing in local cache...
remote publishing ...
Group:'WebServicesNet' is up.
Provider:'Julian inc' is up and waiting for requests.
Waiting for messages to arrive...
-

```

Σχήμα 3.10 Πρόγραμμα διεπαφής του προμηθευτή των υπηρεσιών διαδικτύου, η παρουσίαση των διαφημίσεων γίνεται σαν κείμενο αντί για XML

Παρατηρώντας τα σχήματα αυτά (πελάτη και προμηθευτή) θα πρέπει να επισημάνουμε πως δεν υπάρχει τρόπος να περιγράψουμε την εργασία που εκτελεί η κάθε μέθοδος σε μια υπηρεσία, ειδικά όταν η υπηρεσία διαθέτει περισσότερες από μία μεθόδους, που ίσως είναι αναγκαίο για τον πελάτη. Η ομάδα ανάπτυξης του AXIS [AXI-1] εργάζεται αυτή την περίοδο σε αυτό το αντικείμενο, να φωλιάσει δηλαδή μεταπληροφορία στο WSDL έγγραφο που περιγράφει την υπηρεσία από τεχνική σκοπιά. Κάτι τέτοιο βέβαια σίγουρα θα διευκολύνει πολύ τους παραγωγούς εφαρμογών διαδικτύου, αφού με έναν αναλυτή (όπως ο LocalDOMHandler) να εξάγουν αυτή την μεταπληροφορία και να την παρουσιάσουν στο χρήστη προτού αυτός επιλέξει κάποια από τις μεθόδους. Αυτό όμως αποτελεί ένα ανοιχτό θέμα για το μέλλον.

```

Attempting to Discover the Web Services PeerGroup
Web Services group (WebServicesNet) is on-line.
This is its advertisement:
jxta:PGA :
    GID : urn:jxta:uuid-D1D77D9D0B0B4FFBB4D293DFB60C5AD602
    MSID : urn:jxta:uuid-DEADBEEFDEAFBABAFEEDBABE000000010306
    Name : WebServicesNet
    Desc : WebServicesNet, a group suitable for dealing with web services.

Joining peer group...
Successfully joined group WebServicesNet
Credential:
jxta:Cred ( xml:space="preserve" type="jxta:NullCred" xmlns:jxta="http://jxta.org" ) :
    PeerGroupID : urn:jxta:uuid-D1D77D9D0B0B4FFBB4D293DFB60C5AD602
    PeerID : urn:jxta:uuid-
59616261646162614A78746150325033A8BC026402D94C8B9573FB7F8BA701D403
    Identity : nobody

Getting the discovery, pipe and rendezvous service.->Done!
Please enter some characteristics of the web service you are looking for in the following format:
'-wid WebserviceID' or '-wnm WebServiceName' or '-wdc 'a_Key_word'
'-pid ServiceProviderId or '-pnm ProvidersName or '-pdc ProvidersDescription'
or at least enter the choice on witch discovery will relay on ex. '-wid' or '-wnm' or '-wdc'
'-wnm Stock Exchange*'
Attempting to discover web service advertisements
You search will based on:ServiceName, while searching for: 'Stock Exchange*'
on: WebServicesNet
.Found ws:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jxta:WebServiceAdvertisement>
<jxta:WebServiceAdvertisement xmlns:jxta="http://jxta.org">
    <ServiceName> Stock Exchange Service</ServiceName>
    <ServiceDesc> Providing daily information on stock signs </ServiceDesc>
    <ServiceURL> http://localhost:8080/axis/StockQuoteService.jws </ServiceURL>
    <InfoPipeID>
        urn:jxta:uuid-D1D77D9D0B0B4FFBB4D293DFB60C5AD67BF56CEFEF9349A7BE9D9CDCCE7A55AF04
    </InfoPipeID>
    <ProviderID>
        urn:jxta:uuid-59616261646162614A7874615032503318CF09FE76454D63AC14DB6C114230FE03
    </ProviderID>
    <ProviderName> Julian inc </ProviderName>
    <ProviderDesc> 'Julian inc' is a company focused on developing web services
    </ProviderDesc>
    <ProvidersURL> http://julianinc.com </ProvidersURL>
</jxta:WebServiceAdvertisement>
Path created=localhost/axis/StockQuoteService.jws
WSDL file succesfully downloaded to: localhost/axis/StockQuoteService.jws
WSDL file can be found to:localhost/axis/StockQuoteService.jws/StockQuoteService_wsdl.xml
WSDL2Java created classes to path:localhost/axis/StockQuoteService.jws
Our service class name is:StockQuoteService
Compile succeed, trying to run...
localhost.axis.StockQuoteService.jws.StockQuoteServiceServiceLocator
The service you're trying to use provide the following method (s):
    getQuote
Please the one you want to use!

```

Σχήμα 3.11 Πρόγραμμα πελάτη για την αναζήτηση και χρησιμοποίηση υπηρεσίας σχετικά με πληροφορίες για μετοχές

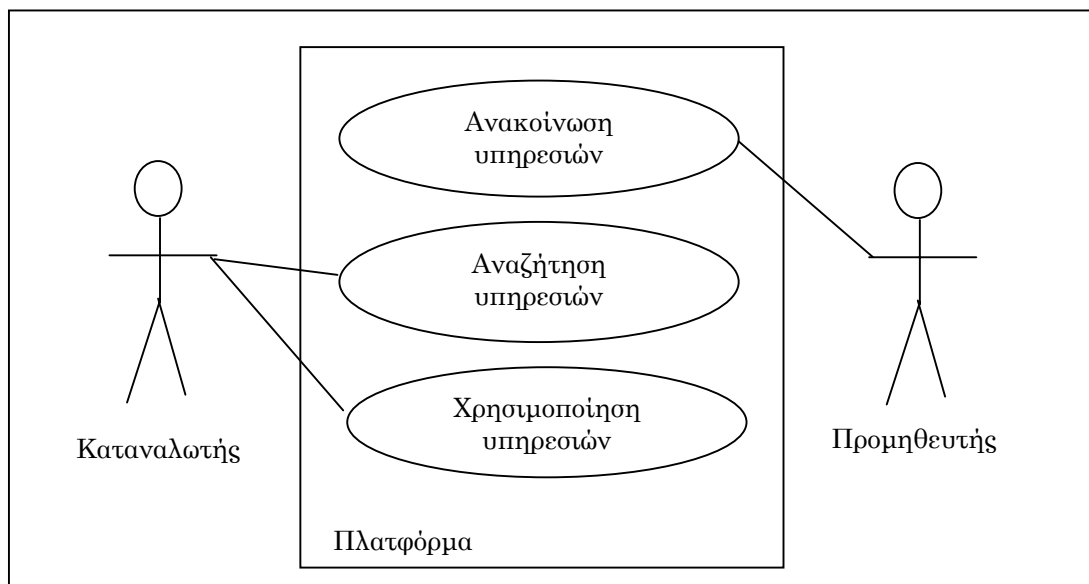
4. Σχεδιαγράμματα UML

Η ενοποιημένη γλώσσα μοντελοποίησης UML (Unified Modeling Language) είναι μια γραφηματική γλώσσα μοντελοποίησης γενικής χρήσης. Παρακάτω παρουσιάζουμε διάφορα σχεδιαγράμματα κατά UML για να δώσουμε μια εικόνα της πλατφόρμας μας.

4.1 Διάγραμμα περιπτώσεων χρήσης

Το διάγραμμα περιπτώσεων χρήσης είναι ένας τρόπος για να περιγραφούν οι υπηρεσίες που προσφέρει ένα σύστημα, σε ποιους και με τη συνεργασία ποιών.

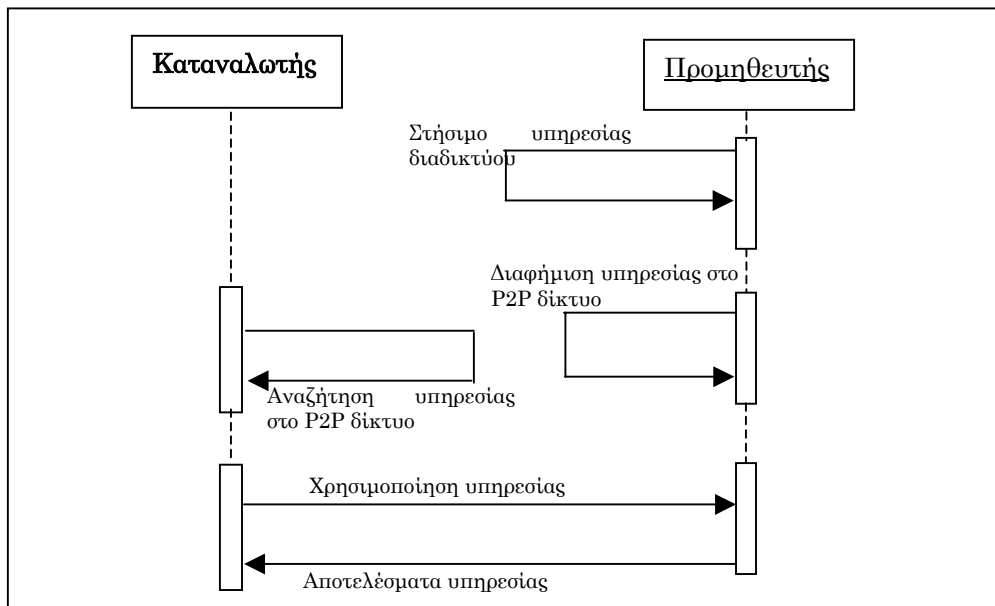
Οι υπηρεσίες που προσφέρει το σύστημά μας είναι για τους προμηθευτές να ανακοινώνουν τις υπηρεσίες διαδικτύου τους και για τους καταναλωτές να αναζητούν και να ανακαλύψουν τις υπηρεσίες διαδικτύου των προμηθευτών και επιπλέον να χρησιμοποιούν επιτυχώς τις υπηρεσίες αυτές.



Σχήμα 4.1 Σχεδιάγραμμα περίπτωσης χρήσης

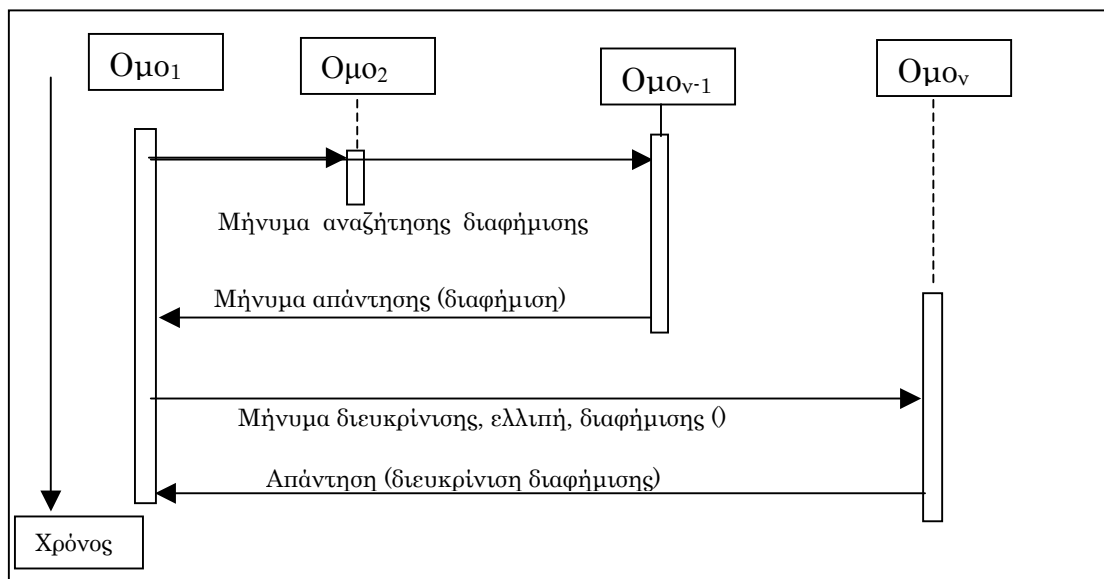
4.2 Ακολουθιακό διάγραμμα

Το ακολουθιακό διάγραμμα περιγράφει την πλήρη εκτέλεση μιας ορισμένης εργασίας σε ένα συγκεκριμένο σενάριο. Σενάριο είναι ο τρόπος εκτέλεσης μιας εργασίας στην οποία λαμβάνουν μέρους συγκεκριμένα αντικείμενα κλάσεων από αυτές που η αντίστοιχη εργασία απαιτεί να συνεργάζονται.



Σχήμα 4.2 Γενικό ακολουθιακό διάγραμμα χρησιμοποίησης δικτυακών υπηρεσιών

Θεωρούμε το παρακάτω σενάριο αναζήτησης μιας υπηρεσίας διαδικτύου. Αφού η αναζήτηση στον τοπικό χώρο αποθήκευσης δεν επιστρέψει κάποιο αποτέλεσμα, ένα μήνυμα αναζήτησης μιας διαφήμισης υπηρεσίας διαδικτύου εκπέμπεται στην ομάδα ομότιμων οντοτήτων στην οποία η αιτούσα οντότητα είναι μέλος. Θεωρούμε πως δεν υπάρχει διάδοση μέσω τρίτων, με πολλαπλασιασμό των μηνυμάτων ερώτησης. Το μήνυμα αυτό λαμβάνεται από τις διάφορες ομότιμες οντότητες σε διαφορετικές χρονικές στιγμές, ενώ όταν μια ομότιμη οντότητα γνωρίζει μια διαφήμιση, που ταιριάζει κάπως, στο μήνυμα αναζήτησης η διαφήμιση αυτή επιστρέφεται στην αιτούσα ομότιμη οντότητα. Εάν η διαφήμιση είναι ελλιπής δημιουργείται ένας σωλήνα επικοινωνίας με τον εκδότη της διαφήμισης της υπηρεσίας και ένα μήνυμα διευκρίνισης στέλνεται σε αυτόν. Αυτός που θα απαντήσει πρώτος σε μια ερώτηση αναζήτησης μπορεί να είναι μια ομότιμη οντότητα διαφορετική από τον εκδότη της διαφήμισης, η οποία έχει λάβει με κάποιο τρόπο, μια διαφήμιση υπηρεσίας διαδικτύου που την έχει αποθηκεύσει στον τοπικό της αποθηκευτικό χώρο και ταιριάζει στην ερώτηση αναζήτησης του καταναλωτή. Ο προμηθευτής απαντά με μια ολοκληρωμένη διαφήμιση υπηρεσίας.

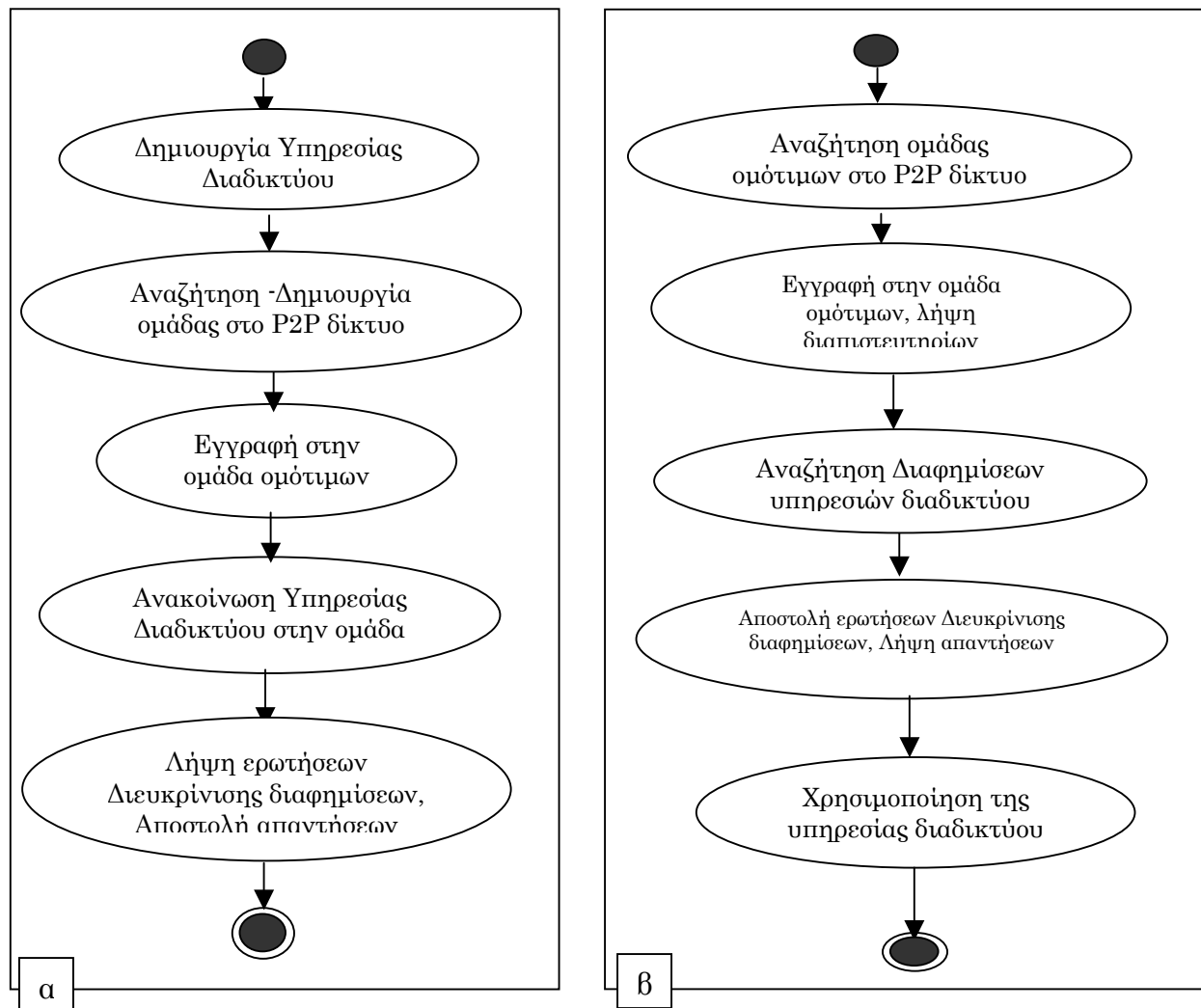


Σχήμα 4.3 Ακολουθιακό διάγραμμα αναζήτησης διαφήμισης δικτυακής υπηρεσίας

4.3 Διάγραμμα δραστηριοτήτων

Το διάγραμμα δραστηριοτήτων περιγράφει το πώς παρέχεται η γενική λειτουργία μιας οντότητας με βάση τις απλούστερες λειτουργίες που ονομάζονται δραστηριότητες.

Οι δραστηριότητες για τις οποίες είναι υπεύθυνος ένας παραγωγός είναι η δημιουργία της υπηρεσίας διαδικτύου και η ανακοίνωσή της στην ομάδα ομότιμων οντοτήτων στην οποία είναι μέλος. Η ανακοίνωση βέβαια των υπηρεσιών είναι μια σύνθετη δραστηριότητα, περιλαμβάνει την αναζήτηση ομάδων ομότιμων οντοτήτων, την εγγραφή στην ομάδα και την αποστολή μηνυμάτων διευκρίνισης για τις ελλιπείς διαφημίσεις. Μόνο όταν ληφθεί μια πλήρης διαφήμιση, από έναν καταναλωτή, μπορεί η δραστηριότητα της ανακοίνωσης να θεωρηθεί πετυχημένη. Από την άλλη οι δραστηριότητες για τις οποίες είναι υπεύθυνος ένας καταναλωτής είναι η εύρεση της υπηρεσίας διαδικτύου στην ομάδα ομότιμων οντοτήτων στην οποία είναι μέλος και η χρησιμοποίησή της. Και εδώ η εύρεση διαφημίσεων υπηρεσιών διαδικτύου είναι μια σύνθετη δραστηριότητα, περιλαμβάνει την αναζήτηση ομάδων ομότιμων οντοτήτων, την εγγραφή στην ομάδα και την αποστολή μηνυμάτων διευκρίνισης για ελλιπείς διαφημίσεις. Μόνο όταν ληφθεί μια πλήρης διαφήμιση, από έναν καταναλωτή, μπορεί η δραστηριότητα της αναζήτησης να θεωρηθεί πετυχημένη.



Σχήμα 4.4 Λιάγραμμα δραστηριοτήτων α) προμηθευτή, β) καταναλωτή

Αν και στο σχήμα αυτό η χρησιμοποίηση της υπηρεσίας, από τον καταναλωτή, αποτυπώνεται ως μια δραστηριότητα αυτή αναλύεται σε επιμέρους δραστηριότητες. Αυτές είναι η μεταφόρτωση του WSDL εγγράφου της περιγραφής της υπηρεσίας, ο σχηματισμός του πηγαίου κώδικα πελάτη, η μεταγλώττιση του πηγαίου κώδικα πελάτη σε αντικείμενο κώδικα, η δημιουργία αναφοράς στις μεθόδους του κώδικα πελάτη και τέλος η επίκληση της μεθόδου που υλοποιεί την υπηρεσία.

5. Ανοιχτά Θέματα

Στην τρέχουσα έκδοση της πλατφόρμας που αναπτύξαμε εγείρονται διάφορα θέματα. Πρωταρχικό θέμα είναι ο έλεγχος, ότι δηλαδή κάθε διαφήμιση αντιστοιχεί σε μια υπηρεσία διαδικτύου. Η πλατφόρμα Jxta ακολουθεί ένα σύστημα αποθήκευσης πληροφοριών, για την αποθήκευση διαφημίσεων, με σκοπό τη μείωση της κίνησης στο δίκτυο. Αυτό μπορεί να προκαλέσει

προβλήματα κατά την αναζήτηση, γιατί οι αποθηκευμένες πληροφορίες μπορεί να είναι παρωχημένες και να μην αντιστοιχούν σε πραγματικές υπηρεσίες. Η λήξη των διαφημίσεων δημιουργεί πρόβλημα στην δικτυακή αναζήτηση, δεδομένου ότι έτσι θα πρέπει να στέλνονται συνεχώς νέες διαφημίσεις, δηλαδή όγκος δεδομένων, που δεσμεύουν εύρος ζώνης.

Το ίδιο αποθηκευτικό σύστημα δημιουργεί κι άλλα προβλήματα. Η αποθήκευση των διαφημίσεων βασίζεται κατά κύριο λόγο στον αριθμό ταυτότητας (ID) που διαθέτουν. Ως συνέπεια αυτού, δύο ίδιες διαφημίσεις με διαφορετικό όμως αριθμό ταυτότητας αποθηκεύονται ως δύο διαφορετικές διαφημίσεις, αυξάνοντας τελικά τον όγκο των αποθηκευμένων πληροφοριών. Για να λυθεί αυτό το πρόβλημα της άχρηστης αποθήκευσης, θα πρέπει οι διαφημίσεις με μη σταθερά χαρακτηριστικά, όπως διαφημίσεις ομότιμων οντοτήτων, σωλήνων, ή υπηρεσιών P2P, να έχουν μικρό χρόνο ζωής. Κάθε φορά που δημιουργείται μια ομότιμη οντότητα αυτοστιγμεί ανατίθεται σε αυτήν ένας τυχαίος αριθμός ταυτότητας, διαφορετικός κάθε φορά. Αν και είναι προγραμματιστικά δύσκολο μια οντότητα να μπορέσει να κρατήσει τον ίδιο αριθμό, στην κοινότητα JXTA αναφέρεται πως είναι δυνατόν. Ο αριθμός ταυτότητας αυτός, δεν πρέπει να συγχέεται με την ταυτότητα της ομότιμης οντότητας, και επίσης δεν πρέπει η αυθεντικότητα μια οντότητας να στηρίζεται σε αυτό. Όταν πάλι δηλώνονται σωλήνες ή υπηρεσίες δεν πρέπει να αναθέτονται νέοι αριθμοί σωλήνα (PipeIds) αλλά να χρησιμοποιούνται παλιοί, γνωστοί αριθμοί. Αντίθετα, διαφημίσεις που δεν αλλάζουν συνεχώς, όπως διαφημίσεις ομάδας ομότιμων οντοτήτων, διαφημίσεις σωλήνα ή διαφημίσεις υπηρεσιών διαδικτύου, θα πρέπει να έχουν μεγάλη διάρκεια ζωής ώστε να μην μετακινούνται συχνά στο δίκτυο. Δηλαδή οι προμηθευτές πρέπει να χρησιμοποιούν τους ίδιους αριθμούς ταυτότητας στις διαφημίσεις τους όταν αλλάζουν απλώς κάποια χαρακτηριστικά των διαφημίσεων και όχι των υπηρεσιών που διαφημίζουν. Με αυτόν τον τρόπο περιορίζουμε το πρόβλημα αποθήκευσης πολλαπλών αντιτύπων που αντιστοιχούν στα ίδια δεδομένα.

Τέλος ανοικτό μένει το ζήτημα ελέγχου αποστολής μηνυμάτων ανακάλυψης – δημοσίευσης διαφημίσεων. Δεν υπάρχει τρόπος να αποτραπεί μια ομότιμη οντότητα να δημοσιεύσει συνεχώς νέες διαφημίσεις, δεσμεύοντας εύρος ζώνης από όλες της ομότιμες οντότητες που συμμετέχουν σε μια ομάδα ομότιμων οντοτήτων. Ο έλεγχος της κίνησης δικτύου που παράγει μια ομότιμη οντότητα μπορεί να εξακριβωθεί μέσω του πρωτοκόλλου πληροφόρησης. Ο συνεχής έλεγχος όμως για όλες τις ομότιμες οντότητες σε μια ομάδα θα προκαλούσε μεγαλύτερη σύγχυση.

Παράρτημα

Παρακάτω καταγράφουμε όλα τα προγράμματα που έχουν δημιουργηθεί πάνω στην πλατφόρμα JXTA με αλφαβητική σειρά.

aisland

Στόχος του είναι η ανάπτυξη ενός συστήματος για τη δημιουργία και τη διανομή κινητών αντιπροσώπων. Το γενικό πλαίσιο εκθέτει τα αντικείμενα σε έναν υπεύθυνο για την ανάπτυξη αντιπροσώπων, ο οποίος μπορεί να «κολλήσει» τις ενότητες αυτές χρησιμοποιώντας μαζί μια script γλώσσα, αρχικά την JavaScript. Οι κινητοί αντιπρόσωποι ταξιδεύουν μεταξύ διαφορετικών AIlands. Εάν ένας πράκτορας ζητήσει μια ενότητα άγνωστη σε ένα AIland, η πλατφόρμα θα ρωτήσει το χρήστη για την άδεια να φορτώσει τον κώδικα από μια ομότιμη οντότητα. Τα aislands μπορούν να διατηρούν σχέσεις εμπιστοσύνης.
Ιστοσελίδα: <http://aisland.jxta.org>

allhands

Στόχος του είναι να δημιουργηθεί μια πλατφόρμα στην οποία τα γεγονότα που δημιουργούνται, δημιουργούν αυτόματα μια βάση γνώσης καταγράφοντας τις ερωταποκρίσεις που διακινούνται, στην οποία μπορούν να υποβληθούν ερωτήσεις.
Ιστοσελίδα: <http://allhands.jxta.org>

cms Το σύστημα διαχείρισης περιεχομένου (cache manager) της JXTA.

Ιστοσελίδα: <http://cms.jxta.org>

chinook

Το πρόγραμμα αυτό αποτελεί μια P2P υπηρεσία βιοπληροφορικής τεχνολογίας. Αυτή τη στιγμή το chinook υλοποιεί πάνω από 25 υπηρεσίες ανάλυσης.
Ιστοσελίδα: <http://chinook.jxta.org>

angelopeerrendezvous

Το πρόγραμμα έχει στόχο να αναπτύξει ένα πλήρες p2p βασισμένο λογισμικό για τη ενδοεπιχειρησιακή επικοινωνία. Οι επιχειρήσεις απαγορεύουν-αποθαρρύνουν τη χρήση των προγραμμάτων επικοινωνίας με στιγμιαία μηνύματα όπως το MSN messenger και Yahoo messenger και εν ενώ υπάρχουν λογισμικό για τη ενδοεπιχειρηματική επικοινωνία, αυτά είναι εξαιρετικά ακριβά.
Ιστοσελίδα: <http://angelopeerrendezvous.jxta.org>

compute-power-market

Μια κατανεμημένη υπολογιστική πλατφόρμα που χρησιμοποιεί μια οικονομική προσέγγιση για υπολογιστικούς τους πόρους στα άκρα του P2P δικτύου. Συγκεκριμένα ο χρήστης της εφαρμογής έχει πρόσβαση σε υπολογιστική ισχύ με μεγάλη ευκολία και πρόσβαση.
Ιστοσελίδα: <http://compute-power-market.jxta.org>

do

Αποτελεί μια εφαρμογή των κατανεμημένων (με αντιγραφή) αντικειμένων.
Ιστοσελίδα: <http://do.jxta.org>

edutella

Βασισμένη στο RDF υποδομή μεταδεδομένων για P2P τις εφαρμογές.
Ιστοσελίδα: <http://edutella.jxta.org>

fanal

Ένα εργαλείο για P2P εφαρμογές, ειδικά για χάρτες κατανεμημένων λογισμικών. Ιστοσελίδα: <http://fanal.jxta.org>

gameplatform

Το πρόγραμμα αυτό προορίζεται να παρέχει μια πλατφόρμα παιχνιδιών για δικτυακά παιχνίδια που χρησιμοποιούν τη JXTA. Υπεύθυνοι για την ανάπτυξη παιχνιδιών μπορούν να αναπτύξουν τα νέα παιχνίδια με το API που παρέχεται από αυτό το πρόγραμμα. Για τους παίκτες, η πλατφόρμα αυτή επιτρέπει σε έναν παίκτη να ανακαλύψει άλλους παίκτες (αντίπαλους ή συνεργάτες) σε ένα παιχνίδι και να παίξει μαζί τους.
Ιστοσελίδα: <http://gameplatform.jxta.org>

gnougat

Το πρόγραμμα αυτό προσπαθεί να αναπτύξει ένα πλήρως αποκεντρωμένο σύστημα αποθήκευσης αρχείων.
Ιστοσελίδα: <http://gnougat.jxta.org>

gnovella

Ένα πειραματικό σύστημα για την ανάπτυξη οντοτήτων για την αποθήκευση, την ανάκτηση και δεικτοδότηση εγγράφων χρησιμοποιώντας τη JXTA. Ιστοσελίδα: <http://gnovella.jxta.org>

go: Είναι ένα παιχνίδι βασισμένο στα πρωτόκολλα της JXTA.
Ιστοσελίδα: <http://go.jxta.org>

halu

Το πρόγραμμα αυτό προσπαθεί να εξερευνήσει πώς η jxta μπορεί να χρησιμοποιηθεί να παρέχει μια λύση που λαμβάνει υπόψη όλες οι πτυχές της δημοσίευσης και κατανάλωσης των πολυμεσικών δεδομένων σε ένα P2P δίκτυο.
Ιστοσελίδα: <http://halu.jxta.org>

jnushare: Προσφέρει τη δυνατότητα στους χρήστες του να μοιράζονται αρχεία χρησιμοποιώντας το GISP (<http://gis.jxta.org>, global info sharing protocol)
Ιστοσελίδα: <http://jnushare.jxta.org>

jsearch: Το πρόγραμμα αυτό προσπαθεί να λύσει το πρόβλημα της αναζήτησης ιστοσελίδων που δεν μπορούν να προσεγγιστούν από τις τρέχουσες μηχανές αναζήτησης. Ιστοσελίδα: <http://jsearch.jxta.org>

jxta-httpd

Το πρόγραμμα αυτό παρέχει ένα σύνολο υπηρεσιών και εργαλείων παρέχοντας δημοσιότητα διαδικτύου χρησιμοποιώντας τη JXTA.

Ιστοσελίδα: <http://jxta-httpd.jxta.org>

juxtaprose

Το πρόγραμμα έχει δύο στόχους:

- Την δημιουργία μιας αυτοοργανωμένης, αποκεντρωμένης ανοιχτής κοινότητας
- Την παροχή οπτικών μέσων για την παρουσίαση υπηρεσιών και πόρων που θα τα καθιστά εύκολα για αναζήτηση και πρόσβαση.

Ιστοσελίδα: <http://juxtaprose.jxta.org>

jxcube

Το Jxta eXtreme Cube -jxcube, αποτελεί μια πλήρη αποκεντρωμένη πλατφόρμα συνεργασίας παρέχοντας συγχρονισμένη και ασύγχρονη συνεργασία με ανταλλαγή μηνυμάτων, αρχείων ή ημερολογίου. Ακόμη υποστηρίζει την πρόσθεση επιπλέον λειτουργιών (plugins) που οι επιχειρηματίες μπορούν να αναπτύσσουν.

Ιστοσελίδα: <http://jxcube.jxta.org>

metasearch

Σκοπό έχει την δημιουργία στρώματος αναζήτηση μεταδεδομένων βασισμένο στην υπηρεσία διαχείρισης περιεχομένου της JXTA.

Ιστοσελίδα: <http://metasearch.jxta.org>

myjxta, myjxta2: Γραφική επίδειξη της πλατφόρμας JXTA.

Ιστοσελίδα: <http://myjxta.jxta.org> και <http://myjxta2.jxta.org> αντίστοιχα.

p2pconference

Βασισμένο σε ένα υπάρχον πρόγραμμα, το eWorkshop, του CeBASE (www.cebase.org), χρησιμοποιεί μια βασισμένη στο WEB, εφαρμογή συνομιλίας. Το eWorkshop είναι δομημένο για να προσαρμόσει τις ανάγκες ενός εργαστηρίου χωρίς να γίνει μια αβίαστη, απευθείας σύνδεση συνομιλίας.

Ιστοσελίδα: <http://p2pconference.jxta.org>

parallax

Το parallax στοχεύει να παρέχει την πρόσβαση σε μια βιώσιμη δομή βιβλιοθήκης για τις ψηφιακές εικόνες, γραφικά, μαζί με ανάμεικτες χρησιμότητες χρησιμοποιώντας τα πρωτόκολλα JXTA. Αυτές οι λειτουργίες θα επεκτείνουν τις βασικές έννοιες JXTA σε «είναι» σχέσεις. Παραδείγματος χάριν, μέλη της βιβλιοθήκης (ομάδα Parallax) θα είναι καθένας καταναλωτής ή κέντρο γνώσης που οι (οντότητες ραντεβού), που μπορούν είτε να μοιράσουν είτε να μεταφορτώσουν περιεχόμενο, φεύγουν ή εισέρχονται (δημόσια ή ιδιωτικά) σε τμήματα ή αναζητούν για περιεχόμενο, καταναλωτές ή άλλα κέντρα γνώσης.

Ιστοσελίδα: <http://parallax.jxta.org>

radiojxta

Αποτελεί την προσπάθεια ανάπτυξης ενός συστήματος για παράδοση ήχου σε δίκτυα jxta. Ιστοσελίδα: <http://radiojxta.jxta.org>

rosettachat

Ένα πρόγραμμα για τη μεταφορά στιγμιαίων μηνυμάτων. Ιστοσελίδα: <http://resettachat.jxta.org>

sentinel

Σκοπός του προγράμματος είναι η ανάπτυξη μιας P2P εφαρμογής για την παρακολούθηση δικτύων, συστημάτων και εφαρμογών. Ιστοσελίδα: <http://sentinel.jxta.org>

shareddiary

Το πρόγραμμα υλοποιεί ένα κατανεμημένο ημερολόγιο για μια ομάδα εργασίας. Ιστοσελίδα: <http://shareddiary.jxta.org>

soundjack

Σκοπός είναι η εκπομπή ζωντανής μουσικής ανάμεσα σε έναν αριθμό από ομότιμες οντότητες. Ιστοσελίδα: <http://soundjack.jxta.org>

tradingcenter

Είναι μια P2P εφαρμογή για τη δημοσίευση και την αναζήτηση προσφορών και πλειοδοτήσεων και την έναρξη και συμμετοχή σε πλειστηριασμούς. Ιστοσελίδα: <http://tradingcenter.jxta.org>

venezia-gondola

Στόχος του προγράμματος είναι να δημιουργηθεί μια P2P εφαρμογή, ένας ελεγκτής. Είναι ένα πείραμα για να δείξει ότι σε ένα P2P δίκτυο και το περιεχόμενο και η λογική συναλλαγής διανέμονται στην άκρη του δικτύου. Η λογική της P2P συναλλαγής πρέπει να αποκεντρωθεί και να συντονιστεί ανάμεσα στις ομότιμες οντότητες σε αντίθεση με το καθιερωμένο πρότυπο όπου η λογική και η ροή της δουλειάς συναλλαγής ρυθμίζονται σε μια κεντρική θέση. Ιστοσελίδα: <http://venezia-gondola.jxta.org>

vop2p

Αυτό το πρόγραμμα διερευνά τη δυνατότητα να δημιουργήσει ένα εντελώς αποκεντρωμένο τηλεφωνικό δίκτυο, ένα τηλεφωνικό δίκτυο χωρίς μια ιεραρχία του κατανεμητές και καμίας απαίτησης για κεντρικό έλεγχο λαθών. Ιστοσελίδα: <http://vop2p.jxta.org>

ΑΝΑΦΟΡΕΣ

- [AXI-1] <http://ws.apache.org/axis/index.html>
<http://ws.apache.org/axis/java/user-guide.html>
- [TOM-1] <http://jakarta.apache.org/tomcat>
- [CERN] <http://www.cern.ch/xdaq> Το σύστημα cross-data acquisition
<http://cmsdoc.cern.ch/> Το πείραμα σωληνοειδούς μιονίου
- [COMP-1] “Framework for Peer-to-Peer Distributed Computing in a Heterogeneous, Decentralized Environment”, Jerome Verbeke, Neelakanth Nadgir, Greg Ruetsch, Ilya Sharapov, Sun Microsystems, Inc.
- [COMP-2] “Peer-to-Peer Computing” Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja1, Jim Pruyne, Bruno Richard, Sami Rollins 2 ,Zhichen Xu HP Laboratories Palo Alto March 8th , 2002
- [COMP-3] Analyzing peer-to-peer traffic across large networks
Subhabrata Sen and Jia Wang, AT&T Labs–Research
- [COL-SOA] SOA and Web Services for Architects and Developers,
Mark Colan, IBM Group.
- [DWS] “Web Services 2002 Market Milestone Report”, Delphi Group White paper.
- [FWP] Network Security: A simple guide to firewalls, White Paper, 3Com Corporation
www.3com.com
- [DOM] <http://www.w3.org/TR/DOM-Level-2-Core>
- [IJECE] Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, Special Issue of the International Journal of Electronic Commerce (IJECE)
- [JXTA] <http://www.jxta.org> Ο υπεύθυνος οργανισμός για την ανάπτυξη της πλατφόρμας JXTA.
- [KYEE] Integrating Amazon Web Services with your Notes databases, Ken Yee,
<http://www.ibm.com/developerworks>
- [LIGON-1] Li. Gong, JXTA: A Network Programming Environment, Internet Computing Online, May/June 2001, Vol.5 No.3, pp. 88-95.
- [LIGON-2] Project JXTA: A Technology Overview, Li Gong, Sun Microsystems
- [OMG96] Common Object Request Broker Architecture and Specification. Object Management Group Document Number 96.03.04.
- [PRE-1] Web Services Market Assessment: “Web Services Markets: Market Strategies, Opportunities, and Forecasts 2002-2007”, Wintergreen Research,

[PRE-2] Patricia Seybold Group's Executive Series: "An Executive's Guide to Web Services: How to Optimize Web Services Investments to Improve Your Customer Experience" By the Patricia Seybold Group

[RFC821] Simple Mail Transfer Protocol (SMTP) . Jonathan B. Postel. August 1982. IETF RFC. 821. Internet Engineering Task Force.

[RFC1700] Assigned Numbers, J. Reynolds, Postel, Network Working Group RFC1700, October 1994

[RFC2141] URN Syntax . R. Moats. May 1997. IETF RFC. 2141. Internet Engineering Task Force.

[RFC2234] Augmented BNF for Syntax Specifications: ABNF . P. Overell and D. Crocker. November 1997. IETF RFC. 2234. Internet Engineering Task Force.

[RFC2246] The TLS Protocol: Version 1.0. Tim Dierks and Christopher Allen. January 1999. IETF RFC. 2246. Internet Engineering Task Force.

[RFC2616] Hypertext Transfer Protocol -- HTTP/1.1 . Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, and Larry Masinter. June 1999. IETF RFC. 2616. Internet Engineering Task Force.

[RFC3080] The Blocks Extensible Exchange Protocol Core . Marshall T. Rose. March 2001. IETF RFC. 3080. Internet Engineering Task Force.

[RFC3081] The Blocks Extensible Exchange Protocol Core . Marshall T. Rose. March 2001. IETF RFC. 3081. Internet Engineering Task Force.

[URL-1] <http://www.yahoo.com>

[URL-2] <http://www.netscape.com>

[URL-3] <http://www.msn.com>

[URL-4] <http://e-docs.bea.com/wls/docs81/webserv/wlw.html>

[URL-5] <http://www.alphaworks.ibm.com/tech/wsde>

[URL-6] <http://www.iona.com>

[URL-7] <http://www.webmethods.com>

[URL-8] <http://www.capeclear.com>

[URL-9] <http://www.idoox.com>

[URL-10]

<http://msdn.microsoft.com/netframework>

<http://msdn.microsoft.com/msdnmag/issues/0300/soap/soap.asp>

Πληροφορίες για το .NET

<http://msdn.microsoft.com/xml/general/soapspec-v1.asp>

Πληροφορίες και προδιαγραφές για το SOAP

[URL-11] <http://www.ibm.com/developerworks>

[URL-12] <http://www.techmetrix.com/trendmarkers/topics>

[URL-13] <http://www.webservices.org>

[URL-14]

<http://www.w3.org> Για την επίσημη προδιαγραφή XML

<http://www.xml.org> Στη θέση αυτή περιέχονται άρθρα και παραδείγματα

http://www.xml.org/xmlorg_registry/ Λίστα με διάφορες γλώσσες σήμανσης που προέκυψαν από την XML
<http://msdn.microsoft.com/workshop/xml/articles/xmlmanifesto.asp>
“XML Manifesto”, by Don Box.

[URL-15]

<http://www.w3.org/2002/ws>
<http://www.w3.org/2002/TR/SOAP> Για τις προδιαγραφές του SOAP
<http://www.w3.org/TR/wsdl> Η γλώσσα wsdl

[URL-16] <http://www.zaks.demon.co.uk/com/soap.html>

[URL-17] <http://java.sun.com/xml/jaxrpc/>

Η προδιαγραφή JAX-RPC, μετασχηματισμός κώδικα Java σε XML και αντίστροφα

[URL-18] <http://ws.apache.org/axis/java/reading.html>

Άρθρα σχετικά με την διαλειτουργικότητα.

[URL-19] <http://www.uddi.org>, <http://uddi.org/pubs/uddi-v3.0.1-20031014.htm>

Ο οργανισμός UDDI και η τρέχουσα προδιαγραφή “3.0.1”.

[USA15] Unicode Standard Annex #15: Unicode Normalization Forms . Mark Davis and Martin Dürst. 26 March 2002. Unicode Standard Annex. 15. The Unicode Consortium. <http://www.unicode.org/unicode/reports/tr15/>

[USA28] Unicode Standard Annex #28: Unicode 3.2 . The Unicode Consortium. March 27 2002. Unicode Standard Annex. 28. The Unicode Consortium.
<http://www.unicode.org/unicode/reports/tr28/tr28-3.html>

[WSA] “Web Services Conceptual Architecture” By Heather Kreger, IBM Software Group, May 2001

[WSEB-1] “Web Services: The Next Horizon for e-business” Mike Conner, IBM

[WSEB-2] “Dynamic e-business: Using Web services to transform business” by Mark Colan, Web Services and XML Chief Advocate IBM

[XSD2001-1] XML Schema Part 1: Structures. Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. 2 May 2001. W3C.
<http://www.w3.org/TR/xmlschema-1/>

[XSD2001-2] XML Schema Part 2: Datatypes. Paul V. Biron and Ashok Malhotra. 2 May 2001. W3C. <http://www.w3.org/TR/xmlschema-2/>

[XSL] <http://www.w3.org/TR/xslt> Προδιαγραφή για μετασχηματισμούς XSL

Βιβλιογραφία

“Professional XML”, by Mark Birbeck, et. al., Wrox Press

“Programming Web Services with XML-RPC” St. Laurent, Simon, Edd Dumbill, and Joe Johnston, (O'Reilly Internet Series) O'Reilly & Associates

“Programming Web Services with SOAP” Doug Tidwell, James Snell, Pavel Kulchenko, O'Reilly & Associates Publishing, December 2001

“Java Web Services” David Chappell, Tyler Jewell, Publisher: O'Reilly & Associates, March 2002

“Understanding Web Services, XML, WSDL, SOAP and UDDI” Eric Newcomer, (David Chappell Series Editor) Addison Wesley Press

“Designing Web Services with the J2EETM 1.4 Platform JAX-RPC, SOAP, and XML Technologies” Inderjeet Singh, Sean Brydon, Greg Murray, Vijay Ramachandran, Thierry Violleau, Beth Stearns, Sun Microsystems Inc. Blueprints 2004

“Building Internet Firewalls” Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman, O'Reilly & Associates, Second Edition, June 2000

“Project JXTA 2.0 Super-Peer Virtual Network” Bernard Traversat, Ahkil Arora, Mohamed Abdelaziz, Mike Duigou, Carl Haywood, Jean-Christophe Hugly, Eric Pouyoul, Bill Yeager, Project JXTA: Sun Microsystems, Inc.

“JXTA”, Brendon J. Wilson - New Riders Publishing, 2002

“JXTA: Java™ P2P Programming” by Daniel Brookshier, Darren Govoni, Navaneeth Krishnan, Juan Carlos Soto - Sams Publishing, 2002

“Mastering JXTA: Building Java Peer-to-Peer Applications” by Joseph D. Gradecki Publisher: John Wiley & Sons, 2002

“The UML Profile for Framework Architectures” Marcus Fontoura, Wolfgang Pree , Bernhard Rumpe, Publisher: Addison Wesley, 2001

“Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example” Doug Rosenberg, Kendall Scott, Publisher: Addison Wesley, 2001.

“UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language” Martin Fowler, Kendall Scott, Publisher: Addison Wesley, 1999.