



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανώνυμη Κοινότητα Ανταλλαγής Πληροφοριών Μεταξύ
Κινητών Τερματικών Σε Πραγματικό Χρόνο**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Κ. Λιαπής

Χρήστος Α. Ηλιόπουλος

Επιβλέπων : Θεολόγου Μιχαήλ
Καθηγητής Ε.Μ.Π

Αθήνα, Ιούνιος 2006



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανώνυμη Κοινότητα Ανταλλαγής Πληροφοριών Μεταξύ Κινητών Τερματικών Σε Πραγματικό Χρόνο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Κ. Λιαπής

Χρήστος Α. Ηλιόπουλος

Επιβλέπων : Θεολόγου Μιχαήλ
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31^η Μήνα Έτος.

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

.....
Ον/μο Μέλος Δ.Ε.Π
Ιδιότητα Μέλους Δ.Ε.Π

Αθήνα, Ιούνιος 2006

.....
Δημήτριος Κ. Λιαπής

.....
Χρήστος Α. Ηλιόπουλος

Διπλωματούχοι Ηλεκτρολόγοι Μηχανικοί και Μηχανικοί Υπολογιστών Ε.Μ.Π.

Copyright © Δημήτριος Λιαπής 2006.

Copyright © Χρήστος Ηλιόπουλος 2006.

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 τους συγγραφείς.

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

Περίληψη

Η συγκεκριμένη διπλωματική εργασία περιγράφει την αρχιτεκτονική και τον τρόπο λειτουργίας μιας πλατφόρμας η οποία προσφέρει την δυνατότητα παροχής υπηρεσιών σε πραγματικό χρόνο, σε χρήστες κινητών τερματικών. Οι υπηρεσίες αυτές είναι διαθέσιμες στα μέλη μιας κοινότητας. Τα μέλη της κοινότητας, χρήστες κινητών συσκευών- τερματικών, χρησιμοποιούν υπηρεσίες που παρέχονται από την κοινότητα και περιέχουν πληροφορίες που προέρχονται από τα υπόλοιπα μέλη της κοινότητας. Οι πληροφορίες δεν βρίσκονται αποθηκευμένες σε ένα κεντρικό εξυπηρετητή αλλά στις τερματικές συσκευές των μελών και λαμβάνονται από αυτές όταν ζητηθεί από κάποιο άλλο μέλος. Ιδιαίτερη αναφορά γίνεται στις οντότητες που λαμβάνουν μέρος κατά την ανταλλαγή πληροφορίας καθώς και στις μεταξύ τους σχέσεις. Επίσης αναλύονται θέματα ασφαλείας, ανωνυμίας των μελών καθώς και συμφόρησης του δικτύου κινητών τηλεπικοινωνιών. Τέλος, γίνεται μελέτη που αφορά την υλοποίηση της περιγραφόμενης πλατφόρμας βασισμένη στις καταλληλότερες σύγχρονες τεχνολογίες.

Λέξεις Κλειδιά: Service Oriented Architecture, Anonymous Community, Mobile Terminal Querying, UDDI, SMS, WAP, WAP Push, Jini Technology.

Abstract

The current dissertation suggests the development of a platform for the provision of real-time services exchange between mobile terminals. Services can be requested by registered members of a community who are users of mobile terminals. The provided services' conveyed information derives from other members of the community. Instead of information being stored in a database, each member provides real-time information on demand. Furthermore, the involved entities along with their relationships are outlined. Security, anonymity and network congestion issues are taken into account. Finally, an implementation study occurs based on the state of the art and appropriate technologies.

Keywords: Service Oriented Architecture, Anonymous Community, Mobile Terminal Querying, UDDI, SMS, WAP, WAP Push, Jini Technology.

Ευχαριστίες

Αισθανόμαστε καταρχήν την ανάγκη να ευχαριστήσουμε από καρδιάς τις οικογένειές μας που μας στηρίζουν στο διαρκή αγώνα μας για γνώση. Χωρίς την κατανόηση και υποστήριξή τους ο αγώνας μας αυτός θα ήταν πολύ δυσκολότερος.

Ευχαριστούμε επίσης όλους τους καθηγητές μας στο Εθνικό Μετσόβιο Πολυτεχνείο, που αυτά τα 5 χρόνια μας δίδαξαν, μας ενέπνευσαν και μας μεταλαμπάδευσαν τη δίψα για γνώση. Ιδιαίτερα, ευχαριστούμε τον Καθηγητή Μιχαήλ Θεολόγου για την άψογη συνεργασία μας.

Τέλος, θέλουμε να αποδώσουμε θερμές ευχαριστίες στον κ.Χρήστο Δεσινιώτη, χωρίς τη βοήθεια του οποίου αυτή η εργασία δεν θα μπορούσε να ολοκληρωθεί.

Ηλιόπουλος Χρήστος
Λιαπής Δημήτριος

Αθήνα, Ιούλιος 2006

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	8
ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ	10
ΣΥΝΤΜΗΣΕΙΣ.....	11
1- ΕΙΣΑΓΩΓΗ.....	14
1.1- ΣΚΟΠΟΣ	15
1.2- ΔΟΜΗ	15
2- ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ ΣΤΗΝ ΠΑΡΟΧΗ ΥΠΗΡΕΣΙΩΝ ΑΝΩΝΥΜΩΝ ΚΟΙΝΟΤΗΤΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ	17
3- Η ΑΝΩΝΥΜΗ ΚΟΙΝΟΤΗΤΑ ΑΝΤΑΛΛΑΓΗΣ ΠΛΗΡΟΦΟΡΙΩΝ ΜΕΤΑΞΥ ΚΙΝΗΤΩΝ ΤΕΡΜΑΤΙΚΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ	19
3.1- Η ΕΝΝΟΙΑ ΤΗΣ ΚΟΙΝΟΤΗΤΑΣ	19
3.2- Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΠΛΑΤΦΟΡΜΑ ΑΝΤΑΛΛΑΓΗΣ ΠΛΗΡΟΦΟΡΙΩΝ	20
3.3- ΓΕΝΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	22
3.4- Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ	24
3.5- ΕΣΩΤΕΡΙΚΗ ΔΟΜΗ ΤΩΝ ΟΝΤΟΤΗΤΩΝ	27
3.5.1- <i>O Service Requestor</i>	27
3.5.2- <i>O Service Provider</i>	29
3.5.3- <i>H Application Service</i>	31
3.5.4- <i>O Community Server</i>	33
3.5.5- <i>To Network Module</i>	36
3.6- ΤΑ ΣΕΝΑΡΙΑ	37
3.6.1- <i>Αίτηση SR για μια Υπηρεσία</i>	37
3.6.2- <i>Δημοσίευση μιας υπηρεσίας από SP και AS</i>	39
3.6.3- <i>Διαδικασία έγγραφης ενός μέλους στην κοινότητα (registration)</i>	40
3.6.4- <i>Οι Διεπαφές των SRs' και SPs'</i>	41
3.7- ΠΙΘΑΝΕΣ ΕΦΑΡΜΟΓΕΣ ΒΑΣΙΣΜΕΝΕΣ ΣΤΗ ΠΡΟΤΕΙΝΟΜΕΝΗ ΙΔΕΑ	44
3.7.1- <i>Υπηρεσία εύρεσης Ταξί</i>	44
3.7.2- <i>Υπηρεσία εύρεσης κίνησης σε Clubs</i>	46
3.7.3- <i>Υπηρεσία εύρεσης Συντροφιάς (Flirt Service)</i>	48
3.7.4- <i>Υπηρεσία δημοσκοπήσεων (Poll Service)</i>	50
3.7.5- <i>Υπηρεσία Location-Based διαφημίσεων (Location-Based Advertising and Marketing)</i>	52
3.8- ΘΕΜΑΤΑ ΣΥΜΦΟΡΗΣΗΣ ΔΙΚΤΥΟΥ	54
3.9- ΘΕΜΑΤΑ ΑΣΦΑΛΕΙΑΣ	57
4- ΕΠΙΛΟΓΕΣ ΥΛΟΠΟΙΗΣΗΣ.....	59
4.1- Η ΤΕΧΝΟΛΟΓΙΑ JINI	59
4.1.1- <i>Αναζήτηση υπηρεσίας</i>	59
4.1.2- <i>Εγγραφή υπηρεσίας στην υπηρεσία αναζήτησης (LookUp Service)</i>	61
4.1.3- <i>Διακίνηση πληροφορίας στο δίκτυο</i>	63
4.1.4- <i>Πολιτική Ασφαλείας στο Jini</i>	64
4.1.5- <i>Πλεονεκτήματα της τεχνολογίας Jini</i>	65
4.1.6- <i>Μειονεκτήματα της τεχνολογίας Jini</i>	66
4.1.7- <i>Εφαρμογή της τεχνολογίας Jini στη προτεινόμενη πλατφόρμα</i>	66
4.2- ΤΟ ΠΡΩΤΟΚΟΛΛΟ UDDI	68
4.2.1- <i>Αρχιτεκτονική του UDDI</i>	69
4.2.2- <i>UDDI Data Model</i>	69
4.2.3- <i>Αναζήτηση Πληροφορίας</i>	73
4.2.4- <i>Εγγραφή/Τροποποίηση Πληροφορίας</i>	74
4.2.5- <i>Διακίνηση πληροφορίας στο δίκτυο</i>	76
4.2.6- <i>Πολιτική Ασφαλείας στο UDDI</i>	76
4.2.7- <i>Πλεονεκτήματα του πρωτοκόλλου UDDI</i>	78
4.2.8- <i>Μειονεκτήματα του πρωτοκόλλου UDDI</i>	78
4.2.9- <i>Εφαρμογή του πρωτοκόλλου UDDI στη προτεινόμενη πλατφόρμα</i>	78

4.3 ΤΟ ΠΡΩΤΟΚΟΛΛΟ HTTPS.....	79
4.3.1 Το πρωτόκολλο SSL.....	79
4.3.2 Εφαρμογή του πρωτοκόλλου HTTPS στην προτεινόμενη πλατφόρμα.....	80
4.4- ΥΠΗΡΕΣΙΑ ΣΥΝΤΟΜΩΝ ΜΗΝΥΜΑΤΩΝ (SMS).....	80
4.4.1- Πως λειτουργεί η υπηρεσία SMS	81
4.4.2- Επικοινωνία εφαρμογών με μηνύματα SMS	82
4.4.3- Πλεονεκτήματα της Υπηρεσίας SMS.....	83
4.4.4- Μειονεκτήματα της Υπηρεσίας SMS	84
4.4.5- Εφαρμογή της υπηρεσίας SMS στη προτεινόμενη πλατφόρμα.....	84
4.5- ΤΟ ΠΡΩΤΟΚΟΛΛΟ WAP.....	85
4.5.1- Η στοιβα πρωτοκόλλων του WAP	85
4.5.2- Η τεχνολογία WAP Push	87
4.5.3- Στοιχεία δικτύου για υποστήριξη WAP	91
4.5.5- Εφαρμογή του πρωτοκόλλου WAP στη προτεινόμενη πλατφόρμα.....	91
4.6- Η ΤΕΧΝΟΛΟΓΙΑ JSP	93
4.6.1- Το Δυναμικό Web.....	93
4.6.2- Το client-side scripting	93
4.6.3- Το πρωτόκολλο HTTP.....	94
4.6.4- Αποθήκευση και ανάκτηση των Cookies	95
4.6.5- Οι Σύνοδοι Χρήστη (User Sessions).....	95
4.6.6- Κωδικοποίηση των URLs.....	96
4.6.7- JavaBeans	96
4.6.8- Εφαρμογή της τεχνολογίας JSP στην προτεινόμενη πλατφόρμα.....	97
4.7- Η ΤΕΧΝΟΛΟΓΙΑ PHP.....	98
4.7.1- Επικύρωση (Authentication) του HTTP με την PHP	99
4.7.2- Συμφόρηση δικτύου και αποδοτικότητα.....	99
4.7.3- Ασφάλεια.....	99
4.7.4- Εφαρμογή της PHP στην προτεινόμενη πλατφόρμα.....	100
4.8- Η ΤΕΧΝΟΛΟΓΙΑ ASP.NET	100
4.8.1- Η .NET πλατφόρμα.....	101
4.8.2- Πρόσβαση σε μια Βάση Δεδομένων από μια σελίδα ASP	102
4.8.3- Εφαρμογή της ASP στην προτεινόμενη πλατφόρμα.....	103
5- ΕΠΙΛΟΓΟΣ	105
5.1- ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ	105
5.2- ΥΠΟΣΤΗΡΙΖΟΜΕΝΕΣ ΥΠΗΡΕΣΙΕΣ ΒΑΣΕΙ ΤΩΝ ΠΙΘΑΝΩΝ ΕΠΕΚΤΑΣΕΩΝ	106
5.2.1- Υπηρεσία εύρεσης διαδρομής.....	106
5.2.2- Υπηρεσία διαμεσολαβητικής πληρωμής (Proxy Pay).....	106
5.2.3- Mobile Multiplayer Gaming	108
5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ	109
ΠΑΡΑΡΤΗΜΑ	110
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ	114

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1. Η αρχιτεκτονική της Ανώνυμης Κοινότητας Ανταλλαγής Πληροφοριών μεταξύ Κινητών Τερματικών σε Πραγματικό Χρόνο	25
Σχήμα 2. Η δομή του Service Requestor	29
Σχήμα 3. Η δομή του Service Provider	31
Σχήμα 4. Η δομή του Application Service	33
Σχήμα 5. Η δομή του Community Server	36
Σχήμα 6 . Η δομή του Network Module	37
Σχήμα 7. Διάγραμμα διαδικασίας αναζήτησης και χρήσης υπηρεσίας	38
Σχήμα 8. Διάγραμμα διαδικασίας εγγραφής υπηρεσιών.....	39
Σχήμα 9. Διάγραμμα διαδικασίας εγγραφής μέλους.....	40
Σχήμα 10. SR's και SP's Διεπαφές (στη στοίβα πρωτοκόλλων) και μεταξύ τους επικοινωνία	41
Σχήμα 11. Δείγμα Ροής Πληροφορίας στον Client του SP.....	43
Σχήμα 12. Διάγραμμα διαδικασίας εύρεσης ταξί.....	45
Σχήμα 13. Διάγραμμα διαδικασίας εύρεσης κίνησης σε club.....	47
Σχήμα 14. Διάγραμμα διαδικασίας εύρεσης συντροφιάς	49
Σχήμα 15. Διάγραμμα διαδικασίας δημοσκόπησης	51
Σχήμα 16. Διάγραμμα υπηρεσίας Location-Based Διαφημίσεων	53
Σχήμα 17. Οι τρεις οντότητες της τεχνολογίας Jini	59
Σχήμα 18. Επικοινωνία οντοτήτων στη τεχνολογία Jini.....	63
Σχήμα 19. Διάγραμμα εγγραφής, αναζήτησης και χρήσης υπηρεσιών	68
Σχήμα 20. Η δομή του <i>bussinessEntity</i>	70
Σχήμα 21. Η δομή του <i>bussinessService</i>	71
Σχήμα 22. Η δομή του <i>bindingTemplate</i>	72
Σχήμα 23. Η δομή του tModel	73
Σχήμα 24. Η δομή του GSM που υποστηρίζει SMS.....	81
Σχήμα 25. Η δομή μηνύματος SMS.....	82
Σχήμα 26. Η τιμές που μπορεί να πάρει το Information-Element-Identifier	83
Σχήμα 27. Η στοίβα των πρωτοκόλλων του WAP	86
Σχήμα 28. Οι τεχνολογίες pull και push	87
Σχήμα 29. Οι οντότητες και τα πρωτόκολλα που χρησιμοποιούνται στο WAP push	88
Σχήμα 30. Ένα δίκτυο WAP.....	91
Σχήμα 31. Τεχνολογία ASP/ ADO	101
Σχήμα 32. Παράδειγμα υλοποίησης web σελίδας της κοινότητας με τεχνολογία ASP	104

ΣΥΝΤΜΗΣΕΙΣ

- **3G** 3rd Generation (Mobile Communications Networks)
- **ADO** ActiveX Data Objects
- **API** Application Programming Interface
- **ASP** Active Server Pages
- **ATM** Asynchronous Transfer Mode
- **B2C** Business-to-Client model
- **B2B** Business-to-Business model
- **B3G** Beyond 3rd Generation
- **BPEL** Business Process Execution Language
- **BSC** Base Station Controller
- **BSS** Base Station System
- **BTS** Base Transceiver Station
- **CA** Certification Authority
- **CAS** Cell Administration Server
- **CGI** Common Gateway Interface
- **cHTML** Compact HTML
- **CSD** Circuit Switched Data
- **CORBA** Common Object Request Broker Architecture
- **DB** Database
- **DOM** Document Object Model
- **DTD** Document Type Definition
- **FTP** File Transfer Protocol
- **EJB** Enterprise JavaBeans
- **GPRS** General Packet Radio Service
- **GPS** Global Positioning System
- **GSM** Global System for Mobile communication
- **GUI** Graphical User Interface
- **HLR** Home Location Register
- **HTML** Hypertext Markup Language
- **HTTP** Hyper Text Transfer Protocol
- **HTTPS** Hyper Text Transfer Protocol over SSL
- **ID** Identification
- **IMAP** Internet Message Access Protocol
- **IP** Internet Protocol
- **IRC** Internet Relay Chat
- **J2ME** Java 2 Micro edition
- **JAR** Java Archive
- **JDK** Java Development Kit
- **JERI** Java Extensible Remote Invocation
- **JINI** Jini is not initials (!)
- **JRMP** Java Remote Method Protocol
- **JSP** Java Server Page
- **JVM** Java Virtual Machine
- **LUS** Look Up Service
- **MD5** Message Digest 5
- **MIME** Multipurpose Internet Mail Extensions
- **MMORPG** Massive Multiplayer Online Role Playing Games

- **MMS** Multimedia Message Service
- **MSC** Mobile Switching Center
- **MSISDN** Mobile Station International ISDN Number
- **NM** Network Module
- **NNTP** Network News Transfer Protocol
- **OTA** Over The Air
- **P2P** Peer to Peer
- **PAP** Push Access Protocol
- **PDA** Personal Digital Assistant
- **PDU** Protocol Data Unit
- **PHP** Hypertext PreProcessor
- **POP3** Post Office Protocol Version 3
- **PPG** Push Proxy Gateway
- **PI** Push Initiator
- **RSA** Rivest, Shamir and Adleman. Inventors of the encryption algorithm

- **RMI** Remote Method Invocation
- **RPC** Remote Procedure Call
- **SAX** Simple API for XML
- **SDU** Service Data Unit
- **SMC** Short Message Center
- **SME** Short Message Entity
- **SMS** Short Message Service
- **SMTP** Simple Mail Transfer Protocol
- **SNMP** Simple Network Management Protocol
- **SOA** Service Oriented Architecture
- **SOAP** Simple Object Access Protocol
- **SP** Service Provider
- **SQL** Structured Query Language
- **SR** Service Requestor
- **SSL** Secure Sockets Layer
- **TCP** Transmission Control Protocol
- **UDDI** Universal Description Discovery and Integration Protocol
- **UDH** User Data Header
- **UML** Unified Modeling Language
- **UMTS** Universal Mobile Telecommunications System
- **URI** Uniform Resource Identifier
- **URL** Uniform Resource Locator
- **USSD** Unstructured Supplementary Service Data
- **VBScript** Microsoft Visual Basic Scripting Edition
- **VLR** Visitor Location Register
- **WAE** Wireless Application Environment
- **WAP** Wireless Application Protocol
- **WDP** Wireless Datagram Protocol
- **WiFi** Wireless Fidelity
- **WiMAX** The name commonly given to the IEEE 802.16 Standard

- **WML** Wireless Markup Language
- **WLAN** Wireless Local Area Network
- **WSP** Wireless Session Protocol

- **WSDL** Web Service Description Language
- **WTLS** Wireless Transport Layer Security
- **WTP** Wireless Transaction Protocol
- **WWW** World Wide Web
- **XHTML** Extensible Hypertext Markup Language
- **XML** Extensible Markup Language
- **XSL** Extensible Stylesheet Language

1- ΕΙΣΑΓΩΓΗ

Τα κινητά τερματικά, ιδιαίτερα στις μέρες μας, μέσα από το πλήθος των εγγενών δυνατοτήτων που παρέχουν, προσφέρουν το αναγκαίο τεχνολογικό υπόβαθρο για την ανάπτυξη και παροχή πλούσιων υπηρεσιών και εφαρμογών. Η διαρκής τους εξέλιξη σε επίπεδο υλικού (hardware) δίνει τη δυνατότητα να ενσωματώνουν πληθώρα χαρακτηριστικών σε όλο και μικρότερο μέγεθος. Σε περίοδο ωριμότητας φαίνεται να εισέρχονται και οι εφαρμογές που αναπτύσσονται για κινητά τερματικά, προσφέροντας με εύχρηστο τρόπο πρόσβαση σε νέες καινοτόμες υπηρεσίες και πληροφορίες που προηγουμένως ήταν προβληματική ή αδύνατη. Μέσω του λογισμικού αναδεικνύονται οι πραγματικές δυνατότητες των σύγχρονων κινητών τερματικών που σε συνδυασμό με τον εύληπτο τρόπο παρουσίασης των πληροφοριών στους τελικούς χρήστες, καθιστούν τους τελευταίους ενεργούς συμμετέχοντες στις τεχνολογικές εξελίξεις.

Τα κινητά τερματικά, μέσα από το μικρό τους μέγεθος και το πλήθος των εργασιών που δύνανται να διεκπεραιώσουν, έχουν πλέον καταξιωθεί στις συνειδήσεις των χρηστών, όχι μονάχα σαν συσκευές πρόσβασης σε δίκτυα κινητής τηλεφωνίας αλλά ως χρηστικά μηχανήματα που βοηθούν και δίνουν λύσεις σε καθημερινά προβλήματα. Η ευρεία αποδοχή τους, ακόμα και στην Ελληνική κοινωνία (70% σύμφωνα με την Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων [1] για το έτος 2005), είναι απλώς ενδεικτική και σκιαγραφεί τις δυνατότητες και τα περιθώρια που διανοίγονται για την ανάπτυξη εφαρμογών και υπηρεσιών στην Ελληνική αγορά. Επιπροσθέτως, αν λάβουμε υπόψη τη σύγκλιση του δικτύου κινητής τηλεφωνίας και του διαδικτύου, μια σχέση που θεμελιώνεται διαρκώς και σταθερά, αυξάνουμε την εμβέλεια και διευρύνουμε το πλήθος των πιθανών παρεχόμενων λύσεων προς όφελος πάντα του τελικού καταναλωτή. Πλέον, οι παρεχόμενες υπηρεσίες μπορούν να εσωκλείουν την πληροφορία της εκάστοτε θέσης του χρήστη. Η ανταλλαγή μηνυμάτων μέσω κειμένου μπορεί να διανθιστεί με ανταλλαγή δεδομένων κάθε μορφής. Ο ίδιος ο χρήστης μπορεί να γίνει τόσο πάροχος όσο και αποδέκτης υπηρεσιών.

Μια νέα εποχή διανοίγεται για την κινητή τηλεφωνία και τις υπηρεσίες που βασίζονται σε αυτή καθώς η ευελιξία, η ευκολία και η πολύχρηστικότητα, στοιχεία δηλαδή που άπτονταν μέχρι πρότινος μονάχα του κόσμου των προσωπικών υπολογιστών, έρχονται πλέον και ενσωματώνονται στα κινητά τερματικά. Το μικρό τους μέγεθος, η φορητότητά τους και η καθολική αποδοχή τους αφήνουν περιθώρια ανάπτυξης υπηρεσιών που θα ξεφεύγουν από τα στενά πλαίσια του όρου «εφαρμογές» και θα εντάσσονται αρμονικά σε κοινωνικές δομές, υπηρετώντας με αυτόν τον τρόπο βαθύτερα και ουσιαστικότερα το αγαθό της επικοινωνίας. Οι κινητές επικοινωνίες έρχονται πλέον να υπηρετήσουν τον τεχνολογικό πολιτισμό.

1.1- ΣΚΟΠΟΣ

Ο σκοπός της προκείμενης εργασίας είναι να αναλύσει τα θέματα που ανακύπτουν και αφορούν μια κοινότητα χρηστών κινητών συσκευών. Η αλληλεπίδραση μεταξύ των μελών-χρηστών της κοινότητας πραγματοποιείται σε επίπεδο υπηρεσιών. Προσθήκοντας σημασία κρίνεται η εξασφάλιση της ανωνυμίας μεταξύ των μελών της κοινότητας. Πιο συγκεκριμένα, κάθε μέλος μπορεί να είναι είτε πάροχος μιας υπηρεσίας είτε αποδέκτης μιας παρεχόμενης υπηρεσίας είτε και τα δύο. Η μελέτη της δομής και της αρχιτεκτονικής λαμβάνει υπόψιν πολλά είδη υπηρεσιών, συμπεριλαμβανομένου υπηρεσίες στις οποίες η πληροφορία της θέσης του εκάστοτε συμβάλλοντος μέλους κρίνεται απαραίτητη για τη διεκπεραίωση της διαδικασίας σε πραγματικό χρόνο. Το θεωρητικό μοντέλο που περιγράφεται, μπορεί να εφαρμοστεί σε οποιοδήποτε ασύρματο δίκτυο στο οποίο τα μέλη της κοινότητας να χρησιμοποιούν κινητές τερματικές συσκευές.

Συγκεκριμένα αναλύονται και προτείνονται λύσεις σχετικά με:

- Την εγγραφή, τη διαγραφή των μελών καθώς και τον έλεγχο της δικτυακής κοινότητας.
- Την αρχιτεκτονική που θα επιτρέπει την ανάπτυξη και παροχή των υπηρεσιών.
- Θέματα επικοινωνίας των μελών της κοινότητας μέσω ανταλλαγής πληροφοριών με τρόπο ώστε να διασφαλίζεται η ανωνυμία του παρόχου καθώς και του αποδέκτη της πληροφορίας.
- Τρόπους δημοσίευσης και παροχής των υπηρεσιών από τα μέλη της κοινότητας στα μέλη της κοινότητας.
- Τρόπους επιλογής της επιθυμητής υπηρεσίας από τα μέλη.
- Υπηρεσίες που χρησιμοποιούν την θέση και κατάσταση του χρήστη.

Εκτός σκοπού αυτής της εργασίας είναι θέματα κοστολόγησης χρήσης και παροχής της κάθε υπηρεσίας αν και θα αναφερθούν στη συνέχεια κάποια κίνητρα τα οποία μπορούν να ωθήσουν τους χρήστες κινητών συσκευών στη χρήση των υπηρεσιών της κοινότητας.

1.2- ΔΟΜΗ

Στο δεύτερο κεφάλαιο γίνεται αναφορά της παρούσης τεχνολογικής πραγματικότητας σχετικά πάντα με θέματα που άπτονται του κεντρικού άξονα της διπλωματικής καθώς και παρουσίαση παρεμφερών (ή απλά σχετιζόμενων) προϊόντων και ερευνητικών προγραμμάτων. Γίνεται μια ανάλυση των ομοιοτήτων και των διαφορών κάθε προγράμματος με την παρούσα εργασία.

Στο τρίτο κεφάλαιο πραγματοποιείται μια πιο εμπειριστατωμένη περιγραφή του θέματος με το οποίο καταπιάνεται η εργασία. Προτείνονται διάφορες εφαρμογές που μπορούν να υλοποιηθούν με βάση το

περιγραφόμενο μοντέλο και αναλύονται οι προδιαγραφές που πρέπει να τηρούνται ώστε μια τέτοια δικτυακή κοινότητα παροχής υπηρεσιών (δια)δικτύου να μπορεί να λειτουργεί ομαλά και αποδοτικά. Στη συνέχεια παρουσιάζεται υπό τη μορφή σχεδιαγράμματος η γενική αρχιτεκτονική, βασισμένη σε διάφορα πιθανά σενάρια όπως και η ροή μηνυμάτων μεταξύ των οντοτήτων που λαμβάνουν μέρος με τις προδιαγραφές της γλώσσας Unified Modeling Language (UML).

Το τέταρτο κεφάλαιο ασχολείται με τα διαφορά τεχνικά θέματα που υπεισέρχονται κατά την υλοποίηση της προτεινόμενης αρχιτεκτονικής και αφορούν τις τεχνολογίες και τις μεθόδους επικοινωνίας των τερματικών συσκευών των χρηστών, του δικτύου και του κεντρικού συντονιστή της κοινότητας. Αναλύονται όλες οι πιθανές λύσεις και γίνεται ιδιαίτερη μνεία στην καταλληλότερη.

Στο πέμπτο και τελευταίο κεφάλαιο της εργασίας παρουσιάζονται τα γενικά πλεονεκτήματα και οι πιθανές ελλείψεις της προτεινόμενης αρχιτεκτονικής. Ακόμα περιγράφονται θέματα ενδεχόμενης βελτίωσης ή επέκτασης του προτεινόμενου μοντέλου ώστε να προσαρμόζεται αρμονικά στις μελλοντικές τάσεις που θα επικρατήσουν στο τομέα των τηλεπικοινωνιών και του διαδικτύου. Η εργασία ολοκληρώνεται με τα εξαγόμενα συμπεράσματα και γενικές παρατηρήσεις έτσι όπως αυτές προέκυψαν καθ' όλη την ενασχόληση με τις σύγχρονες τεχνολογίες και την όλη πορεία της εργασίας.

2- ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ ΣΤΗΝ ΠΑΡΟΧΗ ΥΠΗΡΕΣΙΩΝ ΑΝΩΝΥΜΩΝ ΚΟΙΝΟΤΗΤΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

Μεγάλος αριθμός ερευνητών, εταιριών πληροφορικής και τηλεπικοινωνιών προέβλεψαν την συνεχώς αυξανόμενη επιρροή των κινητών συσκευών στη καθημερινή μας ζωή. Η εύρυθμη λειτουργία αυτών των συσκευών στα τηλεπικοινωνιακά συστήματα έδωσε ώθηση στη ανάπτυξη αρκετών εφαρμογών για κινητές συσκευές.

Οι παραπάνω εφαρμογές ασχολούνται κυρίως με τη δημιουργία blogs. Τα blogs είναι τοποθεσίες στο διαδίκτυο όπου ο διαχειριστής τους έχει την ευχέρεια να παρουσιάζει δεδομένα που αφορούν τον ίδιο και θέλει να τα δημοσιοποιήσει.

Μια τέτοια εφαρμογή είναι και το Cosmorion [\[2\]](#). Το Cosmorion επιτρέπει στους χρήστες του να εισάγουν και να δημοσιεύουν δεδομένα (κείμενα, φωτογραφίες, video) είτε από κινητές συσκευές είτε από προσωπικούς υπολογιστές. Επίσης επιτρέπει την ανάγνωση και αναπαραγωγή αυτών των δεδομένων και με τους δυο προαναφερθέντες τρόπους. Έτσι αναλαμβάνει την μετατροπή δεδομένων που καταχωρήθηκαν από κινητή συσκευή σε μορφή που να μπορεί να είναι καταλλήλως προσβάσιμα και εμφανίσιμα σε προσωπικούς υπολογιστές καθώς και το αντίθετο.

Το Streethive [\[3\]](#) αποτελεί μια ακόμα τέτοια εφαρμογή. Σε αυτή τη κοινότητα ο κάθε χρήστης χρησιμοποιώντας ένα χάρτη καθορίζει την τοποθεσία στην οποία βρίσκεται. Επίσης στέλνει σαν πληροφορία εκτός της θέσης του και κάποια σχόλια καθώς και φωτογραφίες. Έτσι κάποιος μπορεί να αναζητήσει ένα συγκεκριμένο χρήστη βάση του ονόματος του και να διαβάσει τα σχόλια και τις φωτογραφίες που έχει δημοσιεύσει καθώς και το στίγμα του στο χάρτη. Επίσης αναζήτηση μπορεί να γίνει και με την περιοχή ή/και με βάση των θεμάτων που μας ενδιαφέρουν.

Στο ίδιο πλαίσιο με το Streethive κινείται και η εφαρμογή Dodgeball [\[4\]](#). Είναι μια υπηρεσία που συνδυάζει blogs, εύρεση φίλων και σχολιασμό συγκεκριμένων τοποθεσιών σε μια πόλη. Από την ιστοσελίδα του dodgeball τα μέλη εγγράφονται στη κοινότητα και αποκτούν το δικό τους χώρο που μπορεί να περιλαμβάνει μια φωτογραφία τους καθώς και ότι άλλο θέλει να δημοσιεύσει ένα μέλος σε μορφή κειμένου. Επίσης υπάρχουν όλα τα μέρη στα οποία έχει πάει ο συγκεκριμένος χρήστης καθώς και ένα σχολιασμό από μέρος του για την κάθε τοποθεσία. Από τη κινητή του συσκευή ένα μέλος στέλνει ένα μήνυμα στη διεύθυνση του dodgeball αναφέροντας του τη θέση του. Αμέσως η υπηρεσία ενημερώνει τα υπόλοιπα μέλη της κοινότητας που έχει ορίσει ο χρήστης ως φίλους του για τη παρουσία του στο συγκεκριμένο μέρος. Φίλοι των φίλων του συγκεκριμένου χρήστη καθώς και μέλη (μέχρι 5) τα οποία ο χρήστης έχει δηλώσει ότι τρέφει ιδιαίτερα αισθήματα και βρίσκονται σε απόσταση 10 οικοδομικών τετραγώνων θα ειδοποιηθούν επίσης. Ακόμα ένα μέλος μπορεί να στείλει ένα μήνυμα με το

όνομα μιας τοποθεσίας στη πόλη και η υπηρεσία να του απαντήσει με τη διεύθυνση της συγκεκριμένης τοποθεσίας.

Μια πιο σχετική υπηρεσία με την παρούσα εργασία είναι το MoniTap [\[5\]](#). Το Monitar ανέπτυξε μια πλατφόρμα στην οποία χρησιμοποιούνται υπηρεσίες με τις οποίες νοσοκομεία και κλινικές στέλνουν πληροφορίες σχετικές με τις φαρμακευτικές αγωγές ασθενών που πήραν εξιτήριο από αυτές. Το σύστημα με την βοήθεια παρόχων τηλεπικοινωνιακών υπηρεσιών, αναλαμβάνει να υπενθυμίζει στους ασθενείς σχετικά με τη λήψη της απαραίτητης φαρμακευτικής αγωγής τους μέσω SMS. Έτσι αν κάποιος ξεχάσει να πάρει τη φαρμακευτική του αγωγή την ώρα που προβλέπεται, ειδοποιείται μέσω μηνύματος στο κινητό.

Στον ερευνητικό τομέα το πρόγραμμα DBGLOBE [\[6\]](#) παρουσιάζει μια αρχιτεκτονική με την οποία κινητές συσκευές μπορούν να είναι πάροχοι και δέκτες πληροφορίας. Η πληροφορία παρέχεται υπό τη μορφή υπηρεσιών μεταξύ των κινητών συσκευών. Μια υλοποίηση αυτού του ερευνητικού προγράμματος είναι το MobiShare.

Στο MobiShare [\[7\]](#) ο δισδιάστατος χώρος είναι διαιρεμένος σε κυψέλες όπως γίνεται και στην κινητή τηλεφωνία. Τα τερματικά όμως επικοινωνούν μέσω access points χρησιμοποιώντας την τεχνολογία WLAN. Σε κάθε κυψέλη υπάρχει ένας εξυπηρετητής της κυψέλης που ονομάζεται Cell Administration Server (CAS). Ο CAS είναι υπεύθυνος ώστε να γνωρίζει για κάθε συσκευή ανά πάσα στιγμή τη θέση της μέσω GPS συστήματος, που πρέπει να υπάρχει σε κάθε τερματικό, την κατάσταση της συσκευής καθώς και εάν παρέχει κάποιες υπηρεσίες ή πληροφορίες για τις υπηρεσίες αυτές. Επίσης κάθε CAS γνωρίζει τις διευθύνσεις γειτονικών του CAS ώστε όταν μια κινητή συσκευή ζητήσει μια συγκεκριμένη υπηρεσία που δεν υπάρχει στο CAS που έγινε το ερώτημα τότε αυτό να την αναζητά σε γειτονικούς CAS. Το CAS είναι υπεύθυνο επίσης για να δίνει διευθύνσεις δικτύου στα τερματικά, να κάνει έλεγχο πιστοποίησης του κάθε τερματικού, να καταχωρεί υπηρεσίες που παρέχουν τερματικά που βρίσκονται υπό την εποπτεία του μαζί με μια μικρή περιγραφή της κάθε υπηρεσίας. Όταν ένας χρήστης ενδιαφέρεται για μια υπηρεσία ζητά από το CAS στο οποίο είναι συνδεδεμένος να του επιστρέψει ένα κατάλογο με υπηρεσίες που αντιστοιχούν σε κάποιες λέξεις κλειδιά που στέλνει ο ίδιος ο χρήστης. Έτσι ο χρήστης όταν επιλέξει την υπηρεσία που τον ενδιαφέρει ζητά και λαμβάνει από το CAS τη διεύθυνση δικτύου της συσκευής που παρέχει αυτή την υπηρεσία. Ακολουθώντας επικοινωνεί απευθείας με την συσκευή αυτή και κάνει χρήση της συγκεκριμένης υπηρεσίας.

3- Η ΑΝΩΝΥΜΗ ΚΟΙΝΟΤΗΤΑ ΑΝΤΑΛΛΑΓΗΣ ΠΛΗΡΟΦΟΡΙΩΝ ΜΕΤΑΞΥ ΚΙΝΗΤΩΝ ΤΕΡΜΑΤΙΚΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

3.1- Η ΕΝΝΟΙΑ ΤΗΣ ΚΟΙΝΟΤΗΤΑΣ

Η έννοια της κοινότητας δεν είναι καινούργια. Από τεχνολογικής άποψης συντροφεύει το διαδίκτυο από τη στιγμή της δημιουργίας του και εκφράζει την ανάγκη του ανθρώπου για συνεύρεση και συσπείρωση σε κοινωνικές ομάδες που τα μέλη τους μοιράζονται κοινά χαρακτηριστικά γνώρισμα. Σε τεχνολογικό-υπολογιστικό επίπεδο η απαρχή έγινε με το διαδίκτυο και έκτοτε δεν έπαψε να εξελίσσεται. Διαδικτυακά, σχηματίστηκαν τόσες κοινότητες όσες και οι ασχολίες του ανθρώπου. Στη σύγχρονή μας εποχή, οι ψηφιακές κοινότητες έχουν μετεξελιχθεί σε τεχνολογικά κατασκευάσματα ικανά να παρέχουν κάθε είδους απάντηση σε οποιονδήποτε ενδιαφερόμενο. Έχοντας περάσει από πολλές φάσεις της εξελικτικής τους πορείας τόσο σε ασύγχρονο (groups, newsgroups, fora, blogs) όσο και σύγχρονο επίπεδο (IRC, chatting channels, instant messaging channels) πλέον έχουν να επιδείξουν έναν οργανωμένο τρόπο τόσο αναζήτησης όσο και καταχώρησης πάσης φύσεως πληροφορίας.

Καθώς η ζωή των δικτυακών κοινοτήτων ακολουθεί και συνδέεται στενά με τη ζωή του διαδικτύου, διαρκώς εξελίσσεται ώστε να ανταποκρίνεται στα νέα δεδομένα, τα οποία στο χώρο της τεχνολογίας και δει του διαδικτύου, αλληλοδιαδέχονται αδιακόπως το ένα το άλλο. Στη σύγχρονή μας εποχή, η έκταση του διαδικτύου είναι τέτοια που επέβαλε από νωρίς στις κοινότητες να αναπτύξουν μηχανισμούς σωστής οργάνωσης και αναζήτησης της πληροφορίας έτσι όπως αυτή διακινείται ανάμεσα στα μέλη της. Συνεπώς καθίσταται πασιφανές ότι η επιτυχία των ψηφιακών κοινοτήτων, έτσι όπως αυτή εκφράζεται από την ευρεία αποδοχή τους στον ψηφιακό κόσμο, ανταποκρίνεται σε μια από τις σημαντικότερες ανάγκες του σύγχρονου πολιτισμένου ατόμου: εκείνη της επικοινωνίας. Πρόκειται για μια κοινωνική εκδήλωση του ανθρώπου παρόμοια με την ενστικτώδη παρόρμησή του για ενσωμάτωσή σε κοινωνικές ομάδες στην πραγματική ζωή του. Το διαδίκτυο έδωσε τη ευκαιρία στον άνθρωπο να ανάγει την επικοινωνία του σε άλλο επίπεδο. Πλέον δεν περιορίζεται από στενά γεωγραφικά πλαίσια, καθώς πολύ εύκολα γίνεται κοινωνός πολιτισμού. Ασυνείδητα συμβάλλει στη ψηφιακή διακίνηση πάσης φύσεως πληροφορίας μεταξύ των λαών. Βέβαια το μέγεθος του διαδικτύου σε συνδυασμό με το κατεξοχήν φιλελεύθερο και δημοκρατικό καθεστώς που επικρατεί στις διακινούμενες απόψεις, εγείρει θέματα αξιοπιστίας τα οποία δεν θα απασχολήσουν την προκείμενη εργασία. Θεωρητικά, όλα τα μέλη μιας κοινότητας συμμετέχουν σε αυτήν με διττή υπόσταση: δεν γίνονται μονάχα αποδέκτες της πληροφορίας αλλά παρέχουν και πληροφορία. Κατά σύμβαση, όλα τα μέλη της κοινότητας είναι πρόθυμα να συμβάλλουν με κάποιο τρόπο στη λειτουργία της κοινότητας.

Ένα άλλο θέμα αφορά την κινητή τηλεφωνία και το ποσοστό διείσδυσης της στις σύγχρονες κοινωνίες, έτσι όπως αναπτύχθηκε και στην εισαγωγή. Η φορητότητα είναι ένα από τα μεγαλύτερα πλεονεκτήματα δεδομένου του λιλιπούτειου μεγέθους των τερματικών συσκευών κινητής τηλεφωνίας. Πληθώρα τόσο εμπορικών προγραμμάτων όσο και ερευνητικών έργων, στηρίζεται στον προσδιορισμό και την εξόρυξη της θέσης του εκάστοτε χρήστη μέσω διάφορων μηχανισμών.

Η παρούσα διπλωματική καταδεικνύει τη δυνατότητα δημιουργίας κοινοτήτων καθώς και ανάπτυξης υπηρεσιών οι οποίες μπορούν να προσφέρονται μεταξύ χρηστών κινητών τερματικών συσκευών. Ιδιαίτερο ρόλο στην επινόηση αυτή διαδραμάτισε η διαρκής σύγκλιση των κινητών τηλεπικοινωνιών και του διαδικτυακού κόσμου, η οποία συνδυάζει τη φορητότητα και την πληροφορία του γεωγραφικού στίγματος του πρώτου με τις ευκολίες του τελευταίου. Το διαδίκτυο φέρει τη σταθερότητα, την τεχνογνωσία σε επίπεδο πρωτοκόλλων, τη φιλοσοφία των κοινοτήτων ενώ η τηλεπικοινωνιακή κινητή πραγματικότητα αντιπροσωπεύει το καθ' ολικά αποδεκτό και τη φορητότητα των τερματικών της συσκευών. Κατά την περιγραφή της αρχιτεκτονικής και του μοντέλου σημειώνεται σε ποιες περιπτώσεις αυτό είναι βιώσιμο. Πιο συγκεκριμένα, παρουσιάζεται σε ποια ασύρματα δίκτυα μπορεί να υλοποιηθεί (δίκτυα κινητής τηλεφωνίας, ασύρματα δίκτυα που βασίζονται σε συγκεκριμένα πρωτόκολλα) και ποιές πρέπει να είναι οι προϋποθέσεις για τις τερματικές συσκευές.

Σε όλα τα στάδια της παρουσίασης του μοντέλου, μέσω συγκεκριμένων τεχνικών θα εξασφαλίζεται η ανωνυμία των μελών της κοινότητας, θεμελιώδες στοιχείο που χαρακτηρίζει όλες τις ψηφιακές κοινότητες. Συγκεκριμένα, αν και το μοντέλο που περιγράφεται έχει ως πηγή και αποδέκτη τον καταναλωτή, διακατέχεται από την αρχή πώς σημασία έχει η πληροφορία που διακινείται και όχι αυτός που τη διακινεί. Άλλωστε το κάθε μέλος μπορεί να ταυτοποιείται μέσω ενός κωδικού ονόματος (username), το οποίο και θα είναι μοναδικό. Ακόμη, κάθε μέλος μπορεί να είναι τόσο πάροχος όσο και αποδέκτης υπηρεσίας, είτε ταυτόχρονα είτε εναλλάσσοντας μεταξύ των δύο αυτών γνωρισμάτων όποτε το επιθυμεί. Εξάλλου μια τέτοια ελευθερία μπορεί να βοηθήσει στην ανταλλαγή πληροφορίας σε πραγματικό χρόνο.

3.2- Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΠΛΑΤΦΟΡΜΑ ΑΝΤΑΛΛΑΓΗΣ ΠΛΗΡΟΦΟΡΙΩΝ

Μέσω της προτεινόμενης πλατφόρμας, τα μέλη της κοινότητας ανταλλάσσουν μεταξύ τους πληροφορία και πιο συγκεκριμένα πληροφορία που βρίσκεται ενθυλακωμένη σε υπηρεσίες. Κάθε μέλος δύναται όποτε το επιθυμεί να διαθέτει την υπηρεσία του στους ενδιαφερομένους της κοινότητας. Παράλληλα, μπορεί να αναζητήσει μια υπηρεσία που πιθανόν να χρειάζεται, μεταξύ των ήδη διαθέσιμων από τα άλλα μέλη-παρόχους. Κάθε υπηρεσία ξεκινά και καταλήγει στα μέλη της κοινότητας. Η όλη διαδικασία είναι απλοϊκή, χωρίς να απαιτεί πολύ χρόνο από τα ενδιαφερόμενα μέλη. Ο προτεινόμενος μηχανισμός δεν επιβαρύνει τα κινητά τερματικά και η επικοινωνία εξασφαλίζεται με τον πιο αποδοτικό τρόπο,

σεβόμενη το περιορισμένο εύρος ζώνης του ασύρματου δικτύου στο οποίο υλοποιείται.

Η διακίνηση της πληροφορίας είναι κεντριοποιημένη. Η ίδια η πληροφορία όμως είναι κατανοητή. Ένας μεσάζοντας (διακομιστής) προωθεί, εγγράφει, διαγράφει, πιστοποιεί, αποθηκεύει και διακινεί την πληροφορία μεταξύ των μελών της κοινότητας. Ο ρόλος του είναι πολύ σημαντικός καθώς με αυτοματοποιημένο τρόπο διεκπεραιώνει μια σειρά εργασιών για την εύρυθμη λειτουργία της κοινότητας. Πρόκειται να κρατά στοιχεία που αφορούν τις διαθέσιμες υπηρεσίες ώστε γρήγορα να τις προωθεί στους εκάστοτε ενδιαφερόμενους. Ο διακομιστής είναι εκείνος που επιλαμβάνεται της ευθύνης να κρατά στοιχεία σχετικά με το αν ο πάροχος μιας υπηρεσίας είναι συνδεδεμένος τη στιγμή που θα κληθεί να παράσχει την υπηρεσία του σε ένα άλλο μέλος. Παράλληλα, μέσω της απλής καταγραφής των διαθέσιμων υπηρεσιών, ο διακομιστής κρατά στοιχεία ώστε να βοηθά και να καθιστά ευκολότερη τη διαδικασία αναζήτησης μιας υπηρεσίας. Τέλος, ειδικοί μηχανισμοί τίθενται σε λειτουργία όταν ο ασύγχρονος τρόπος επικοινωνίας κρίνεται επιβεβλημένος.

Πιο συγκεκριμένα, η προτεινόμενη από την παρούσα εργασία πλατφόρμα, έχει ως πυρήνα της αρχιτεκτονικής της δύο βασικές θεματικές ενότητες- ακρογωνιαίους λίθους. Η πρώτη ενότητα αφορά στον σχηματισμό μιας ζωντανής κοινότητας χρηστών τερματικών συσκευών, οι οποίοι χαίρουν της δυνατότητας διακίνησης υπηρεσιών. Η δεύτερη ενότητα έχει να κάνει με την τεχνολογική υλοποίηση αυτής της διαπροσωπικής διάδρασης, δηλαδή με ποιους τρόπους διακινείται η εγκαλούμενη ή εγκληθείσα υπηρεσία στους κόλπους μιας κοινότητας.

Τα συμβαλλόμενα μέλη της κοινότητας έχουν μοναδικά ονόματα, γεγονός που διασφαλίζει την απαραίτητη ανωνυμία. Πρόκειται για το κοινωνικό ψευδώνυμο του κάθε μέλους. Η ταυτότητα του εκάστοτε μέλους εξακριβώνεται διαμέσου ενός κωδικού τον οποίο και γνωρίζει μονάχα το ίδιο το μέλος και κατόπιν αναγκαστικής κοινοποίησης ο φορέας της πλατφόρμας που έχει επιφορτιστεί με τη διαδικασία της πιστοποίησης. Ένας χρήστης κινητής τερματικής συσκευής μπορεί να γίνει μέλος της κοινότητας αφού πρώτα εγγραφεί σε αυτήν λαμβάνοντας έτσι όνομα και συνθηματικό (κωδικό). Στη συνέχεια, ο κεντρικός εξυπηρετητής αναλαμβάνει να διεκπεραιώσει την επικοινωνία μεταξύ των μελών της κοινότητας διασφαλίζοντας με αυτόν τον τρόπο την ανωνυμία.

Κάθε μέλος της κοινότητας που ζητά μια υπηρεσία απευθύνεται στο κεντρικό εξυπηρετητή της κοινότητας αναζητώντας την (υπηρεσία) μέσω ενός ονόματος που είναι μοναδικό. Έτσι μπορούν να συνυπάρχουν πολλές κοινότητες με διαφορετικά ονόματα ή ακόμα και πολλοί εξυπηρετητές για μια κοινότητα. Αυτό δίνει την απαιτούμενη ευελιξία ώστε σε περίπτωση που αλλάξει διεύθυνση δικτύου ένας εξυπηρετητής να μην προκληθεί πρόβλημα στη επικοινωνία. Έτσι ο χρήστης πρέπει να στείλει στον εξυπηρετητή κάποια πληροφορία σχετική με την υπηρεσία που επιθυμεί να χρησιμοποιήσει. Σε περίπτωση που υπάρχουν πολλές διαθέσιμες παρόμοιες υπηρεσίες που μπορεί να τον ενδιαφέρουν, μια λίστα

αναλαμβάνει να τον ενημερώσει σχετικά με τη διαθεσιμότητα αυτών των υπηρεσιών.

Αφού ένας χρήστης καταχωρήσει επιτυχώς το όνομα και το κωδικό του και δεδομένου ότι έχει πιστοποιηθεί σαν εγγεγραμμένο μέλος, μπορεί να προχωρήσει στην χρήση υπηρεσιών. Ο εξυπηρετητής τότε αναλόγως του είδους της αιτούμενης υπηρεσίας, καλεί το χρήστη να αποστείλει επιπλέον πληροφοριακά στοιχεία που ενδεχομένως να απαιτεί η υπηρεσία ώστε να λειτουργήσει σωστά. Ο εξυπηρετητής αναλαμβάνει να διεκπεραιώσει αυτόματα τις απαιτούμενες διεργασίες εκ μέρους του χρήστη τον οποίο και πληροφορεί σχετικά με την έκβαση της αίτησής του. Δηλαδή του αποστέλλει τα αποτελέσματα της υπηρεσίας, ενώ στην περίπτωση που δεν υπάρχουν τέτοια, τον ενημερώνει απλά με ένα μήνυμα επιτυχούς έκβασης.

Από την πλευρά του παρόχου της υπηρεσίας (ο οποίος στην περίπτωσή μας είναι ο τελικός χρήστης) τα πράγματα είναι διαφορετικά. Ο πάροχος αφού εγγραφεί επιτυχώς και πιστοποιηθεί ως μέλος της κοινότητας, αποστέλλει πληροφορία σχετική με την υπηρεσία που έχει σκοπό να προσφέρει διαμέσου της κοινότητας στα υπόλοιπα μέλη. Ακολουθώντας, με διάφορες τεχνικές (που περιγράφονται αναλυτικότερα παρακάτω) διασφαλίζεται το γεγονός ότι εξακολουθεί να προσφέρει την υπηρεσία του. Κάτι τέτοιο πραγματοποιείται με τη βοήθεια των πρωτοκόλλων επικοινωνίας που χρησιμοποιεί ο εξυπηρετητής της κοινότητας. Συνεπώς, ενδεχόμενη αποσύνδεση του παρόχου της υπηρεσίας για οποιονδήποτε λόγο, αυτόματα σημάνει και τον τερματισμό της υπηρεσίας που έως πρότινος διέθετε στα άλλα μέλη της κοινότητας.

3.3- ΓΕΝΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Μια πλατφόρμα δημοσίευσης, αναζήτησης και παροχής υπηρεσιών πρέπει να τηρεί κάποιες απαραίτητες προδιαγραφές, ούτως ώστε να μπορεί να έχει λειτουργική χρήση, οι οποίες περιγράφονται παρακάτω.

α) Διαλειτουργικότητα (Interoperability)

Υποθέτουμε ότι οι χρήστες των υπηρεσιών καθώς και οι πάροχοι τους χρησιμοποιούν κινητές συσκευές διάφορων κατασκευαστών και με ποικίλα λειτουργικά συστήματα. Έτσι θα πρέπει η πλατφόρμα που θα δημιουργήσουμε να είναι ανεξάρτητη λειτουργικού συστήματος και αρχιτεκτονικής του υλικού της συσκευής.

Το μόνο που θα πρέπει να είναι κοινό για όλες τις συσκευές είναι τα πρωτόκολλα επικοινωνίας τα οποία θα πρέπει να είναι κατανοητά τουλάχιστο από τα ζευγάρια οντοτήτων αιτούμενου υπηρεσίας – εξυπηρετητή και πάροχου υπηρεσίας – εξυπηρετητή αφού όπως αναφέρθηκε στη περιγραφή της πλατφόρμας, ο εξυπηρετητής λειτουργεί σαν ενδιάμεσος (proxy) μεταξύ αιτούμενου και παρόχου υπηρεσίας.

β) Απλότητα (Simplicity)

Όσο αφορά τους χρήστες υπηρεσιών, καθώς και τις συσκευές τους, η αναζήτηση και χρήση των υπηρεσιών δεν πρέπει να προϋποθέτει οποιαδήποτε ιδιαίτερη γνώση υπολογιστών. Ο χρήστης πρέπει να χρησιμοποιεί μια διασύνδεση που να του επιτρέπει απλά και εύκολα να αναζητεί και να χρησιμοποιεί υπηρεσίες χωρίς να τον ενδιαφέρει πως υλοποιείται η υπηρεσία σε επίπεδο μεταφοράς και δικτύου.

Το ίδιο ισχύει και για το πάροχο της υπηρεσίας. Με μια εφαρμογή που θα λειτουργεί στο κινητό τερματικό του θα γίνεται αυτόματη εγγραφή στη κοινότητα και παράλληλα διάθεση της υπηρεσίας του στα υπόλοιπα μέλη. Όταν ζητηθεί από ένα πάροχο υπηρεσίας να προσφέρει πληροφορία στη κοινότητα, αναλόγως της υπηρεσίας, είτε η εφαρμογή θα στέλνει πληροφορία χωρίς καμιά ενέργεια από τη πλευρά του ή θα ειδοποιείται για να παράσχει ο ίδιος την πληροφορία που θέλει.

γ) Κλιμακωσιμότητα (Scalability)

Η όλη περιγραφή της ιδέας αναφέρεται σε ένα εξυπηρετητή στον οποίο ενδιαφερόμενα μέλη της κοινότητας παρέχουν και χρησιμοποιούν υπηρεσίες. Η πλατφόρμα θα πρέπει να μπορεί να λειτουργήσει και με πολλούς εξυπηρετητές οι οποίοι να συνεργάζονται. Για παράδειγμα ενδέχεται σε μια κοινότητα να υπάρχουν εξυπηρετητές, κάθε ένας εκ των οποίων να αναλαμβάνει ένα ξεχωριστό είδος υπηρεσιών. Έτσι, οι εξυπηρετητές που συμμετέχουν θα πρέπει να επικοινωνούν αυτόματα και να συνεργάζονται αν η αιτούμενη πληροφορία είναι κατανοημένη ή δεν βρίσκεται σε εκείνον στον οποίο γίνεται η αρχική αίτηση αναζήτησης. Επομένως όλοι οι εξυπηρετητές έχουν δυνητικά τις απαραίτητες πληροφορίες (εφόσον συνεργάζονται- επικοινωνούν μεταξύ τους) και δεν παίζει ρόλο ο εξυπηρετητής στον οποίο γίνεται η αρχική σύνδεση.

Ακόμα πρέπει να είναι δυνατή η χρησιμοποίηση των υπηρεσιών της πλατφόρμας και σε άλλες πλατφόρμες. Δηλαδή να μπορεί κάποιος πάροχος υπηρεσιών σε άλλη πλατφόρμα να χρησιμοποιεί τις υπηρεσίες που είναι καταχωρημένες σε ένα εξυπηρετητή της κοινότητας που περιγράφει η παρούσα εργασία και βάσει αυτών των υπηρεσιών να παρέχει άλλες υπηρεσίες αλλού.

δ) Ευελιξία (Flexibility)

Η συζητούμενη πλατφόρμα για να λειτουργήσει σαν πλατφόρμα παροχής υπηρεσιών πρέπει να παρέχει ευελιξία όσον αφορά το είδος των υπηρεσιών που μπορεί να προσφέρει. Αυτό σημαίνει ότι ο ενδιαμέσος εξυπηρετητής μπορεί να υποστηρίξει κάθε είδος υπηρεσίας π.χ. υπηρεσίες που βασίζονται στη κατάσταση ή τη θέση του χρήστη (context based services, location based services) ή υπηρεσίες ειδοποίησης (Notification Services). Αν η υπηρεσία είναι υπηρεσία ειδοποίησης με γραπτό μήνυμα τότε η πλατφόρμα θα πρέπει να παρέχει ένα τρόπο το μήνυμα που στέλνει ο πάροχος σε κάποιο χρήστη να φτάσει στο τηλεπικοινωνιακό φορέα του

παραλήπτη ούτως ώστε να μπορεί ο τελευταίος να το δει στην οθόνη της κινητής του συσκευής.

ε) Διατήρηση Απορρήτου και Ασφάλεια (Privacy and Security)

Η διατήρηση της ανωνυμίας και η ασφάλεια κατά τη χρήση της περιγραφόμενης πλατφόρμας παίζει καθοριστικό ρόλο για ένα μελλοντικό της χρήστη. Κανείς δεν θα χρησιμοποιούσε μια πλατφόρμα στην οποία θα ήταν δυνατό να διακινηθούν προσωπικά του στοιχεία σε τρίτους. Επίσης κανείς δεν θα διακινδύνευε την ασφάλεια της συσκευής του από κακόβουλους χρήστες της κοινότητας.

Στοιχεία που θα μπορούσαν να διαρρεύσουν από μια ανασφαλή πλατφόρμα είναι η διεύθυνση δικτύου του χρήστη, ο αριθμός της τηλεφωνικής του σύνδεσης καθώς και πληροφορίες για τη συσκευή του.

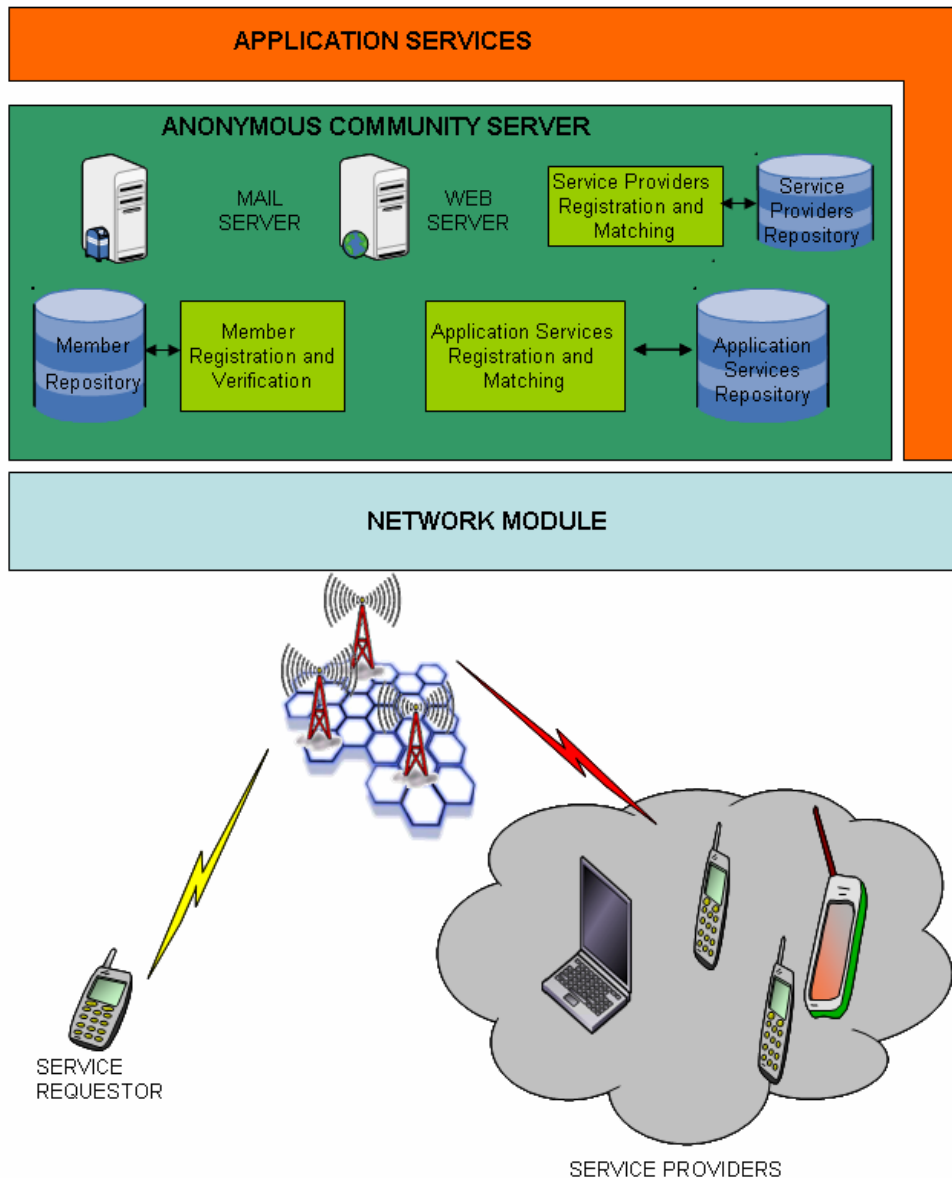
Αυτά είναι αρκετά σε συνδυασμό με άλλα στοιχεία του χρήστη που είναι απαραίτητα για την παροχή υπηρεσιών όπως η γεωγραφική του θέση ή αριθμός πιστωτικής κάρτας να χρησιμοποιηθούν για κακόβουλες ενέργειες.

Η πλατφόρμα θα πρέπει να παρέχει μεθόδους με τις οποίες να αποτρέπει τη ταυτοποίηση του χρήστη μιας υπηρεσίας από το πάροχο της υπηρεσίας αυτής καθώς και το αντίστροφο. Αυτό δεν πρέπει σε καμία περίπτωση να είναι ανασταλτικός παράγοντας για την λειτουργία μιας παρεχόμενης υπηρεσίας. Η πλατφόρμα πρέπει να διαθέτει μεθόδους με τις οποίες αυτό να είναι δυνατό.

Επίσης η πλατφόρμα θα πρέπει σαν αξιόπιστη αρχή να χρησιμοποιεί τα δεδομένα που της παρέχουν οι χρήστες της κατά την εγγραφή τους ή κατά τη χρήση της πλατφόρμας μόνο για δικούς της σκοπούς και με κανένα τρόπο να μην τα κοινοποιεί σε τρίτους.

3.4- Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Στο παρακάτω σχήμα παρουσιάζεται η αρχιτεκτονική του προτεινόμενου μοντέλου διακίνησης της πληροφορίας μεταξύ των μελών της κοινότητας.



Σχήμα 1. Η αρχιτεκτονική της Ανώνυμης Κοινότητας Ανταλλαγής Πληροφοριών μεταξύ Κινητών Τερματικών σε Πραγματικό Χρόνο

Στο πιο πάνω σχήμα απεικονίζονται οι συμβαλλόμενες οντότητες (Service Requestor, Service Provider), το επίπεδο μεταφοράς δικτύου (Network Module), ο διακομιστής της ανώνυμης κοινότητας (Anonymous Community Server) καθώς και η υπηρεσία-εφαρμογή (Application Service).

Ο **Service Requestor (SR)** είναι αυτός που επιθυμεί να κάνει χρήση της υπηρεσίας. Στην ουσία πρόκειται για ένα μέλος της κοινότητας που διαμέσου της τερματικής συσκευής του θέλει να κάνει χρήση μιας εκ των παρεχόμενων από την κοινότητα υπηρεσιών. Οποιοδήποτε μέλος της κοινότητας μπορεί να λάβει τον ρόλο του Service Requestor, αποκτώντας πρόσβαση στη διατιθέμενη πληροφορία μέσω των διακινούμενων υπηρεσιών. Η ύπαρξη του SR είναι άμεσα συνυφασμένη με την ύπαρξη της ίδιας της κοινότητας. Δεν νοείται λειτουργική διακίνηση της προσφερόμενης πληροφορίας χωρίς την ύπαρξη ενός SR-αποδέκτη. Αποτελεί τον διακινητή/

διαμορφωτή των προσφερόμενων υπηρεσιών, αφού ασκώντας το δικαίωμα της επιλογής, κατά μια έννοια καθορίζει και αξιολογεί την παρεχόμενη πληροφορία. Λαμβάνοντας υπόψη τον σημαντικότερο αυτό ρόλο του SR, ο εκάστοτε διαχειριστής του Community Server, μπορεί να διευρύνει τα κοινοτικά πλαίσια, εισάγοντας SRs ενδεχομένως από άλλες κοινότητες (ή απλά επιλεγόμενους χρήστες). Κατ' αυτόν τον τρόπο εξασφαλίζεται η μεγαλύτερη απήχηση των υπηρεσιών. Είναι στη διακριτική ευχέρεια του operator (ο οποίος επιφορτίζεται με την ολική επίβλεψη της προτεινόμενης κοινοτικής πλατφόρμας) ανάλογα με το επιχειρηματικό μοντέλο που ακολουθεί, να καθορίσει/ εκμεταλλευτεί ανάλογα το ποσοστό διείσδυσης εξωτερικών SRs στην κοινοτική οικογένεια.

Ο **Service Provider (SP)** είναι ο πάροχος μιας υπηρεσίας. Πρόκειται ουσιαστικά για μέλος της κοινότητας που μέσω της τερματικής συσκευής του και (πιο συγκεκριμένα) του εξειδικευμένου λογισμικού που έχει εγκαταστήσει σε αυτό, διατίθεται να προσφέρει οποιαδήποτε πληροφορία, σχετική πάντα με την προκείμενη υπηρεσία. Τον ρόλο του SP μπορεί να λάβει οποιοδήποτε μέλος της κοινότητας. Επιπροσθέτως, πολλά διαφορετικά μέλη ενδέχεται να αποτελούν Service Providers της ίδιας υπηρεσίας. Με αυτό τον τρόπο διασφαλίζουμε την ύπαρξη περισσότερων πηγών άντλησης πληροφοριών σχετικά με μία μόνο υπηρεσία, γεγονός που οδηγεί σε εγκυρότερα, πληρέστερα και πιο εξειδικευμένα αποτελέσματα. Ο ρόλος του SP, καθρεφτίζει τη διάθεση προσφοράς στην κοινότητα.

Ο **Community Server (CS)** παρεμβάλλεται μεταξύ του SR και των SPs, εξασφαλίζοντας έτσι την ανωνυμία μεταξύ των μελών της κοινότητας. Βέβαια, εκτελεί και μια σειρά άλλων κρίσιμων λειτουργιών. Καταρχήν, διατηρεί μια βάση δεδομένων με τα στοιχεία όλων των μελών της κοινότητας. Με τη βοήθεια της συγκεκριμένης βάσης, γίνεται ταυτοποίηση/ πιστοποίηση κάθε φορά που κάποιος χρήστης προσπαθεί να εισέλθει στην κοινότητα. Η δεύτερη βάση δεδομένων καταγράφει και διατηρεί ένα κατάλογο με τις διαθέσιμες για χρήση εφαρμογές-υπηρεσίες. Μια τρίτη βάση δεδομένων είναι επιφορτισμένη με τους διαθέσιμους πάροχους υπηρεσίας SPs. Οι πάροχοι υπηρεσιών (SP) μιας συγκεκριμένης εφαρμογής- υπηρεσίας θα πρέπει να έχουν εγκαταστήσει στις τερματικές συσκευές τους την υπηρεσία (κατάλληλο λογισμικό) που σχετίζεται με την εφαρμογή-υπηρεσία και να δηλώσουν την διαθεσιμότητα τους στο CS. Ο SR κάνει χρήση μιας εφαρμογής- υπηρεσίας ώστε να αντλήσει την πληροφορία που χρειάζεται από κάποιον (ή κάποιους) SP(s), που θα έχουν ενεργοποιημένη την υπηρεσία που αντιστοιχεί στην επιλεγόμενη από τον SR εφαρμογή υπηρεσία. Η επικοινωνία SR και SP δεν είναι άμεση καθώς παρεμβάλλεται σε κάθε μορφής συνδιαλλαγή ο CS και η εφαρμογή-υπηρεσία. Ο CS είναι υπεύθυνος για τον εντοπισμό και επιλογή κάθε φορά των SP που θα παρέχουν την πληροφορία που ζητά ένας SR. Φυσικά η επιλογή κάθε φορά γίνεται με βάση το είδος της εφαρμογής υπηρεσίας που χρησιμοποιεί ο SR. Παράλληλα, ο CS χρησιμοποιώντας το Network Module μπορεί να επικοινωνεί με ένα SMS/MMS Gateway, γεγονός που εξασφαλίζει ενδεχόμενη επικοινωνία με τα μέλη, μέσω SMS /MMS ειδοποίησης. Κάτι τέτοιο μπορεί να λαμβάνει χώρα κατά την εγγραφή/ διαγραφή/ αίτηση ενημέρωσης των ίδιων των μελών. Τέλος, Web και Mail Servers θα είναι

έτοιμοι να στηρίξουν την διαδικτυακή ύπαρξη της κοινότητας και να προσφέρουν υποστήριξη, οδηγίες και επικοινωνία με ειδικούς για την αντιμετώπιση προβλημάτων (tutorials, troubleshooting). Ο CS αντιπροσωπεύει τον υπεύθυνο/ διαιτητή/ δρομολογητή πάσης φύσεως πληροφορίας που διακινείται στους κόλπους της κοινότητας.

Η **Application Service (AS)**, είναι η εφαρμογή-υπηρεσία η οποία στηρίζεται από τις υπηρεσίες που παρέχονται μέσα στην κοινότητα από τους SPs. Σκοπός του AS αποτελεί την άντληση πληροφοριών υπό την μορφή υπηρεσιών από τα τερματικά των SPs, αφού έχει δεχθεί μια αίτηση για παροχή υπηρεσίας από ένα SR. Είναι υπεύθυνη να ενημερώνει το CS κάθε φορά που χρειάζεται ο εντοπισμός και επιλογή των SP που θα παρέχουν την πληροφορία που ζητά ένας SR καθώς και για το φιλτράρισμα αυτής της πληροφορίας έτσι ώστε να αποφευχθεί άσκοπη διακίνηση άχρηστης για τον SR πληροφορίας στο δίκτυο.

Το **Network Module (NM)** αντιπροσωπεύει το επίπεδο μεταφοράς και δικτύου με το οποίο γίνεται η επικοινωνία των μελών της κοινότητας (είτε SR είτε SP), των εφαρμογών-υπηρεσιών και του διακομιστή της κοινότητας. Το NM αποτελεί τον ασύρματο ή/και ενσύρματο δίαυλο επικοινωνίας που μπορεί να υλοποιείται κατά περίπτωση διαφορετικά, ανάλογα με την ακολουθούμενη τεχνολογία. Ορισμένα πιθανά NM είναι τα τηλεπικοινωνιακά δίκτυα GSM σε συνδυασμό με το GPRS και UMTS, τα ασύρματα δίκτυα WLAN (WiFi, WiMAX) καθώς και τα ενσύρματα δίκτυα όπως ATM, TCP/IP, οπτικές ίνες, τηλεφωνικό δίκτυο συμπεριλαμβανομένου του εσωτερικού δικτύου των παρόχων τους. Το NM καθρεφτίζει το μέσο μεταφοράς της διακινούμενης πληροφορίας μέσα στην κοινότητα.

Το NM αναλαμβάνει, κάθε φορά που μια εφαρμογή-υπηρεσία, διαμέσου του CS, ζητά παρόχους της υπηρεσίας αυτής, να ειδοποιεί τις συσκευές των παρόχων υπηρεσίας ότι πρέπει να στείλουν πληροφορία στη κοινότητα. Η ειδοποίηση των παρόχων υπηρεσίας μπορεί να γίνει είτε βάση μόνο της διαθεσιμότητας τους, είτε βάση της διαθεσιμότητας και της θέσης του παρόχου (σε αυτή την περίπτωση χρησιμοποιούνται τεχνικές broadcasting σε μια συγκεκριμένη κυψέλη).

3.5- ΕΣΩΤΕΡΙΚΗ ΔΟΜΗ ΤΩΝ ΟΝΤΟΤΗΤΩΝ

3.5.1- Ο Service Requestor

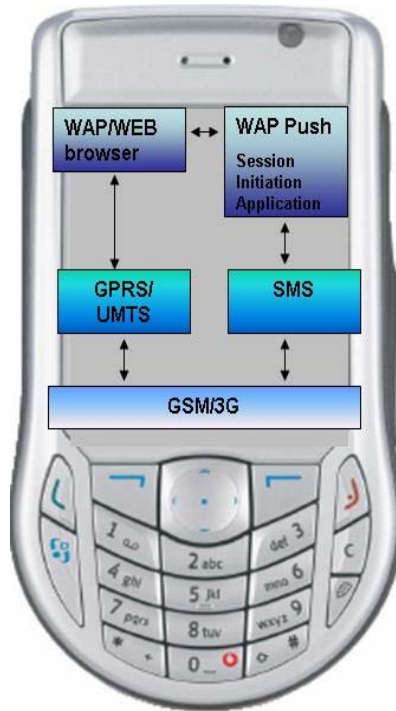
Ένα μέλος της κοινότητας ανάλογα με τις δυνατότητες του κινητού τερματικού που χρησιμοποιεί, μπορεί να χρησιμοποιήσει τις υπηρεσίες που προσφέρει η κοινότητα με διαφορετικούς τρόπους. Οι απαιτήσεις της κοινότητας από ένα Service Requestor είναι τέτοιες ώστε να μπορεί κάποιος να λειτουργήσει μέσα στη κοινότητα χωρίς να χρειαστεί να εγκαταστήσει οποιαδήποτε εφαρμογή στο τερματικό του ή να πρέπει να ασχοληθεί με θέματα που απαιτούν εξειδικευμένες γνώσεις.

Η διασύνδεση του SP με τον Community Server μπορεί να υλοποιηθεί μέσω Web ή WAP. Έτσι στο κινητό τερματικό του Service Requestor πρέπει

να υπάρχει εγκατεστημένος ένας πλοηγός (browser) ο οποίος να υποστηρίζει μια από τις δύο τεχνολογίες. Επίσης για την σύνδεση ενός μέλους, που ζητά μια υπηρεσία από την κοινότητα, με τον κεντρικό εξυπηρετητή της κοινότητας χρειάζεται η δυνατότητα από μέρους της κινητής συσκευής να χρησιμοποιεί τεχνολογίες που επιτρέπουν την αποστολή και λήψη δεδομένων προς και από το διαδίκτυο. Η τεχνολογία GPRS, η οποία είναι ήδη αρκετά διαδεδομένη και υποστηρίζεται σχεδόν από όλα τα δίκτυα κινητών τηλεπικοινωνιών είναι αρκετή για να ικανοποιήσει τις ανάγκες τόσο του Community Server όσο και των Application Services ώστε να επιτευχθεί η αναζήτηση, εύρεση και χρήση των υπηρεσιών που προσφέρουν, από το κινητό τερματικό του Service Requestor. Κινητές συσκευές τρίτης γενιάς (3G) μπορούν επίσης να χρησιμοποιήσουν εναλλακτικά το σύστημα UMTS που προσφέρει μεγαλύτερες ταχύτητες διαδικτυακής πρόσβασης. Πιο εξελιγμένα κινητά τερματικά μπορούν να χρησιμοποιήσουν τεχνολογίες ασύρματης δικτύωσης όπως Wi-Fi για επικοινωνία με το Community Server μέσω ενός WLAN δικτύου παρακάμπτοντας έτσι το δίκτυο κινητών τηλεπικοινωνιών.

Όσον αφορά την ενημέρωση των μελών σχετικά με την κοινότητα, αυτή μπορεί να πραγματοποιηθεί είτε με WAP push είτε με μηνύματα SMS ανάλογα με τη δυνατότητα της συσκευής να υποστηρίξει μια από τις δύο τεχνολογίες. Η αδυναμία ενός μέλους της κοινότητας να λαμβάνει ειδοποιήσεις από τον Community Server δεν αποτρέπει την δυνατότητα ενός μέλους της κοινότητας να χρησιμοποιήσει τις παρεχόμενες υπηρεσίες.

Έτσι μια ελάχιστη απαίτηση, όσον αφορά την εσωτερική δομή του κινητού τερματικού του Service Requestor, είναι να υποστηρίζει τις τεχνολογίες WAP, GPRS καθώς και την δυνατότητα αποστολής και λήψης μηνυμάτων SMS η οποία είναι αναγκαία για την εγγραφή ενός μέλους στη κοινότητα. Οι ελάχιστες αυτές απαιτήσεις ικανοποιούνται πλήρως από μια συνηθισμένη κινητή συσκευή, οπότε η εσωτερική δομή του κινητού τερματικού του Service Requestor ακολουθεί την εσωτερική δομή μιας συνηθισμένης κινητής συσκευής. Διαγραμματικά η δομή του Service Requestor φαίνεται στο πιο κάτω σχήμα.



Σχήμα 2. Η δομή του Service Requestor

3.5.2- O Service Provider

Τα μέλη της κοινότητας που το επιθυμούν μπορούν να γίνουν πάροχοι υπηρεσιών (Service Providers) σε υπηρεσίες-εφαρμογές που επιλέγουν. Για να γίνει αυτό θα πρέπει το μέλος της κοινότητας να εγκαταστήσει το κατάλληλο λογισμικό (software) με το οποίο το κινητό τερματικό του θα μπορεί να στέλνει την απαραίτητη, σε σχέση με την εφαρμογή-υπηρεσία, πληροφορία στο Community Server όταν αυτός το ζητά. Η πληροφορία αυτή μπορεί να βρίσκεται αποθηκευμένη στο κινητό τερματικό του χρήστη ή να την εισάγει ο ίδιος ο χρήστης όταν του ζητηθεί. Στη περίπτωση που η πληροφορία πρέπει να εισαχθεί από τον χρήστη, τότε είτε η εφαρμογή διαθέτει μια κατάλληλη γραφική διασύνδεση με την οποία ο χρήστης εισάγει την απαραίτητη πληροφορία, είτε ο Service Provider με τη χρήση της τεχνολογίας WAP Push παραπέμπεται σε μια σελίδα WML (του πρωτοκόλλου WAP) όπου εισάγει την απαραίτητη πληροφορία. Στη περίπτωση που η πληροφορία αποστέλλεται αυτόματα από την κινητή συσκευή, η ίδια η εφαρμογή διαβάζει την πληροφορία που χρειάζεται να αποστείλει (στίγμα GPS, Cell ID, ταχύτητα) από τον χώρο στον οποίο η κινητή συσκευή καταγράφει τα αναγκαία δεδομένα. Επίσης η εφαρμογή που εγκαθίσταται στο τερματικό του Service Provider αναλαμβάνει να ενημερώνει, ανά τακτά χρονικά διαστήματα, τον Community Server για την διαθεσιμότητα του παρόχου υπηρεσίας να δώσει πληροφορία. Αυτό μπορεί να πραγματοποιηθεί κατόπιν εντολής του SP ή ακόμα και αυτόματα κάθε φορά που η τερματική συσκευή του SP είναι ενεργή.

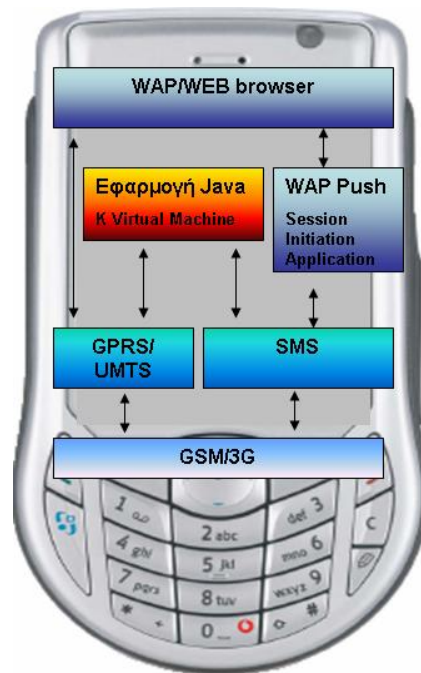
Για να λειτουργήσει η πιο πάνω διαδικασία το κινητό τερματικό του Service Provider πρέπει να έχει την δυνατότητα εγκατάστασης και λειτουργίας του απαιτούμενου λογισμικού. Σε κινητές συσκευές υπάρχουν

δύο τρόποι να επιτευχθεί εγκατάσταση μιας εφαρμογής. Ο πρώτος είναι μέσω σύνδεσης με κάποιο υπολογιστή (ενσύρματη ή ασύρματη) που να μπορεί να κάνει την εγκατάσταση, ενώ ο δεύτερος μέσω του δικτύου κινητών τηλεπικοινωνιών με χρήση της τεχνολογίας OTA. Για OTA εγκατάσταση απαιτείται GPRS σύνδεση του τερματικού με τον εξυπηρετητή που διανέμει το απαραίτητο λογισμικό. Κινητές συσκευές τρίτης γενιάς (3G) μπορούν να χρησιμοποιήσουν εναλλακτικά το σύστημα UMTS που προσφέρει μεγαλύτερες ταχύτητες πρόσβασης. Οι περισσότερες εφαρμογές που αναπτύσσονται για κινητά τερματικά είναι υλοποιημένες στη γλώσσα προγραμματισμού J2ME (Java 2 Micro Edition). Αυτό συμβαίνει λόγω του ότι εφαρμογές που αναπτύσσονται με την J2ME είναι ανεξάρτητες του λειτουργικού συστήματος και του υλικού (hardware) του κινητού τερματικού. Έτσι το κινητό τερματικό του Service Provider πρέπει να υποστηρίζει εφαρμογές Java. Μια συσκευή για να μπορεί να εκτελέσει τέτοιου είδους εφαρμογές πρέπει επίσης να διαθέτει το K Virtual Machine το οποίο απαιτεί τις επιπλέον ακόλουθες δυνατότητες:

- Επεξεργαστή 16 bit /16 MHz ή καλύτερο
- Μνήμη 160-512 KB αφιερωμένη μόνο για την πλατφόρμα Java

Ακόμα, απαιτείται συνδεσιμότητα με την χρήση των τεχνολογιών GPRS ή UMTS για την αναφορά διαθεσιμότητας παροχής υπηρεσίας στον Community Server.

Έτσι μια ελάχιστη απαίτηση όσον αφορά την εσωτερική δομή του κινητού τερματικού του Service Requestor, ο οποίος λειτουργεί με τον προαναφερθέν τρόπο, είναι να υποστηρίζει εγκατάσταση και λειτουργία εφαρμογών της τεχνολογίας Java, σύνδεση στο διαδίκτυο καθώς και την δυνατότητα αποστολής και λήψης μηνυμάτων SMS που ήδη είναι αναγκαία δυνατότητα για τη εγγραφή στη κοινότητα. Επιπλέον απαιτείται υποστήριξη της τεχνολογίας WAP push, στην περίπτωση στην οποία η εισαγωγή της πληροφορίας γίνεται βάσει WML σελίδων. Διαγραμματικά η δομή του Service Provider φαίνεται στο πιο κάτω σχήμα.



Σχήμα 3. Η δομή του Service Provider

3.5.3- Η Application Service

Η εφαρμογή-υπηρεσία (Application Service) αποτελεί την οντότητα η οποία υλοποιεί μια υπηρεσία που προσφέρεται στα μέλη της κοινότητας. Παραδείγματα τέτοιων υπηρεσιών αναφέρονται σε επόμενο κεφάλαιο ([κεφάλαιο 3.7](#)). Οι λειτουργίες που πρέπει να υποστηρίζει μια εφαρμογή υπηρεσία είναι:

- Η αποστολή και λήψη μηνυμάτων SOAP σε UDDI Registry (που βρίσκεται στον Community Server) για την καταχώρηση, τροποποίηση και διαγραφή της εφαρμογής-υπηρεσίας στη κοινότητα.
- Η επικοινωνία με τον Community Server με αποστολή και λήψη μηνυμάτων SOAP έτσι ώστε να ζητά και να λαμβάνει την κατάλληλη πληροφορία που χρειάζεται για να διεκπεραιώσει ένα αίτημα παροχής υπηρεσίας από τον SR.
- Η παροχή κατάλληλης διασύνδεσης στον SR, έτσι ώστε να του παρέχει την υπηρεσία που αιτείται βάσει των δυνατοτήτων του τερματικού του.
- Η επεξεργασία των δεδομένων που λαμβάνει από τους παρόχους υπηρεσίας (SP) διαμέσου του Community Server.
- Η φύλαξη, αναζήτηση και εύρεση πρόσφατων αποτελεσμάτων.
- Η δυνατότητα εγγραφής πληροφοριών από τα μέλη της κοινότητας.

Η εφαρμογή-υπηρεσία παρέχει στον διαχειριστή της τη δυνατότητα να την εγγράψει, να την τροποποιήσει και να την διαγράψει από την κοινότητα μέσω μιας εφαρμογής (UDDI application) που επικοινωνεί με τη UDDI Registry του Community Server. Έτσι η εφαρμογή πρέπει να μπορεί να λαμβάνει, να επεξεργάζεται, να δημιουργεί και να αποστέλλει τα κατάλληλα SOAP μηνύματα (βάσει του API που ορίζει το πρωτόκολλο UDDI) που χρειάζονται. Για το σκοπό αυτό πρέπει να έχει εγκατεστημένη μία SOAP Engine, όπως τον Axis [\[24\]](#) εάν η εφαρμογή υλοποιηθεί σε πλατφόρμα Java,

η οποία αναλαμβάνει την δημιουργία, αποστολή και επεξεργασία SOAP φακέλων καθώς και την ενθυλάκωση σε αυτούς της πληροφορίας που χρειάζεται. Η πληροφορία που ενθυλακώνεται ακολουθεί τους κανόνες της γλώσσας XML, προσφέροντας στον παραλήπτη τη δυνατότητα να διαχειρίζεται την πληροφορία σημασιολογικά. Δυνατότητα διαχείρισης πληροφορίας XML από εφαρμογές δίνουν εργαλεία όπως ο Xerces [23] και ο Xalan [22] (για υλοποιήσεις σε Java). Η εφαρμογή που αναλαμβάνει τις λειτουργίες επικοινωνίας με τη UDDI Registry του Community Server πρέπει να μπορεί να χρησιμοποιεί τα πιο πάνω εργαλεία.

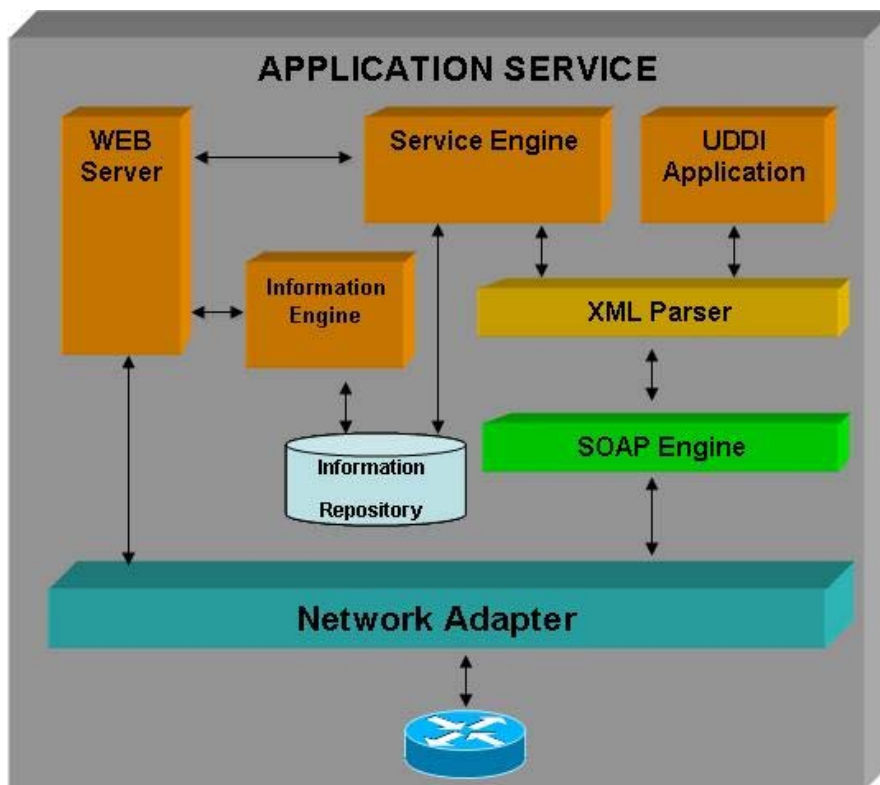
Η δεύτερη εφαρμογή που χρειάζεται μια Application Service για να λειτουργήσει σύμφωνα με την αρχιτεκτονική που προτείνει η εργασία αυτή, αναλαμβάνει να αναζητήσει την πληροφορία που ζητά ένας SR, εάν ο ίδιος το επιθυμεί, από την βάση δεδομένων που διατηρεί (Information Repository) η εφαρμογή- υπηρεσία ενώ στην περίπτωση που αυτό δεν είναι επιθυμητό από τον SR αναλαμβάνει την επικοινωνία με τον Community Server καθώς και την επεξεργασία των δεδομένων που λαμβάνει από αυτόν. Η εφαρμογή αυτή (Service Engine) απαιτεί τη χρήση των ίδιων εργαλείων που αναφέρθηκαν στη προηγούμενη παράγραφο (Axis, Xalan, Xerces) καθώς η επικοινωνία γίνεται με SOAP μηνύματα που περιέχουν την πληροφορία σε κείμενο XML. Η δομή της XML πληροφορίας που ανταλλάσσεται είναι τέτοια ώστε να είναι αναμενόμενη τόσο από τον CS όσο και από την AS ώστε μπορεί να γίνει κατανοητή κατά την επεξεργασία. Η πληροφορία που αποστέλλεται στον Community Server περιέχει ένα μοναδικό αναγνωριστικό της εφαρμογής υπηρεσίας, ένα αριθμό μηνύματος, κάποια κριτήρια που πρέπει να εκπληρώνουν οι πάροχοι υπηρεσίας (Cell ID, Longitude, Latitude, πιθανές ομάδες στις οποίες ανήκουν) καθώς και ένα ελάχιστο αριθμό παρόχων υπηρεσίας που χρειάζονται να ερωτηθούν. Η πληροφορία που λαμβάνει η Application Service περιέχει τον αριθμό μηνύματος του οποίου είναι απάντηση το μήνυμα που στέλνει ο Community Server, τον αριθμό των παρόχων υπηρεσίας που έστειλαν πληροφορία καθώς και τις πληροφορίες αυτές. Τέλος οι πληροφορίες αυτές αποστέλλονται στον SR και ταυτόχρονα καταγράφονται στην βάση δεδομένων της εφαρμογής υπηρεσίας.

Η εφαρμογή Information Engine αναλαμβάνει να καταγράφει στη βάση δεδομένων πιθανές πληροφορίες που θέλει να αποστείλει ένα μέλος της κοινότητας, το οποίο μπορεί και να μην είναι πάροχος υπηρεσίας. Έτσι εάν επιθυμεί κάποιος, μπορεί να στείλει πληροφορίες στην εφαρμογή υπηρεσία χωρίς να του το ζητηθεί προηγουμένως.

Όσο αφορά την επικοινωνία των SR ή των μελών της κοινότητας με την εφαρμογή-υπηρεσία, η κάθε εφαρμογή-υπηρεσία χρειάζεται ένα εξυπηρετητή ιστού (web server) μέσω του οποίου θα παρέχεται η κατάλληλη διασύνδεση του χρήστη (Web/WAP interface) με την εφαρμογή που υλοποιεί την υπηρεσία (Service Engine). Ο εξυπηρετητής πρέπει να υποστηρίζει τα πρωτόκολλα HTTPS και τη στοίβα των πρωτοκόλλων WAP για να είναι δυνατή η ασφαλής επικοινωνία με τον SR. Ένας τέτοιος εξυπηρετητής μπορεί είναι ο Apache Web Server σε συνεργασία με τον Tomcat servlet container (εφόσον η Service Engine υλοποιηθεί σε Java).

Όσον αφορά την WAP διασύνδεση, αυτή μπορεί να μην παρέχεται άμεσα από την Application Service. Οι Service Requestors που χρησιμοποιούν WAP μπορεί να επικοινωνούν με την εφαρμογή-υπηρεσία μέσω ενός WAP Proxy που βρίσκεται στον Community Server και αναλαμβάνει την μετατροπή των πρωτοκόλλων από WAP σε WWW και το αντίθετο.

Τέλος καμιά από τις πιο πάνω τεχνολογίες δεν μπορεί να λειτουργήσει χωρίς σύνδεση στο διαδίκτυο την οποία οφείλει να έχει κάθε Application Service. Διαγραμματικά η δομή της Application Service φαίνεται στο πιο κάτω σχήμα.



Σχήμα 4. Η δομή του Application Service

3.5.4- Ο Community Server

Ο Community Server είναι η κύρια οντότητα της κοινότητας. Επικοινωνεί με όλες τις άλλες οντότητες παρέχοντας τους τις δυνατότητες που χρειάζονται για να λειτουργήσουν στην κοινότητα. Οι λειτουργίες που υποστηρίζει είναι:

- Η παροχή κατάλληλης διασύνδεσης στον SR, έτσι ώστε να του παρέχει εγγραφή και σύνδεση στη κοινότητα καθώς και δυνατότητες αναζήτησης και προβολής των υπηρεσιών που προσφέρονται από την κοινότητα.
- Η παροχή πιστοποιητικών ασφαλείας σε όλες τις οντότητες της κοινότητας.

- Η καταχώρηση, επεξεργασία και διαγραφή των υπηρεσιών που προσφέρονται.
- Η καταχώρηση, εύρεση, επιλογή και η ειδοποίηση παρόχων υπηρεσίας (Service Providers).
- Η καταχώρηση όλων απαραίτητων στοιχείων των μελών της κοινότητας.
- Η αποστολή ενημερωτικών μηνυμάτων στα μέλη της κοινότητας καθώς και σε χρήστες κινητών τερματικών που δεν είναι ακόμα μέλη της κοινότητας.
- Η επικοινωνία με τις εφαρμογές-υπηρεσίες (Application Services) για αποστολή πληροφοριών σε αυτές από τους παρόχους υπηρεσιών (Service Providers).

Ένας χρήστης κινητού τερματικού για να μπορέσει να χρησιμοποιήσει οποιαδήποτε λειτουργία παρέχει η κοινότητα, πρέπει να έχει ήδη εγγραφεί ως μέλος στη κοινότητα. Το μήνυμα SMS που στέλνει ένα υποψήφιο μέλος για να εγγραφεί, μεταφέρεται από το τηλεπικοινωνιακό δίκτυο μέσω του Network Module στον Community Server. Μια εφαρμογή (Member Registration and Verification Application) που βρίσκεται στο Community Server πρέπει να αναλαμβάνει την συλλογή αυτών των μηνυμάτων καθώς και την επικοινωνία με τη βάση δεδομένων που καταχωρούνται τα μέλη της κοινότητας (Member Repository). Η βάση δεδομένων που θα υπάρχουν καταχωρημένα όλα τα μέλη της κοινότητας είναι απλή και μπορεί να υλοποιηθεί με μια βάση δεδομένων όπως την MySQL. Φυσικά με την εφαρμογή αυτή, μέσω Web και WAP διασύνδεσης, θα μπορούν να επικοινωνούν όλα τα μέλη για να γίνεται η κατάλληλη πιστοποίηση βάσει των username και password τους. Αφού πιστοποιηθεί ένα μέλος από την εφαρμογή τότε δίνεται εντολή στην αρχή πιστοποίησης (Certification Authority) έτσι ώστε να του εκδοθεί ένα πιστοποιητικό ασφαλείας. Αφού ο SR λάβει το πιστοποιητικό μπορεί να επικοινωνήσει με τη εφαρμογή-υπηρεσία και να χρησιμοποιήσει τις δυνατότητες της.

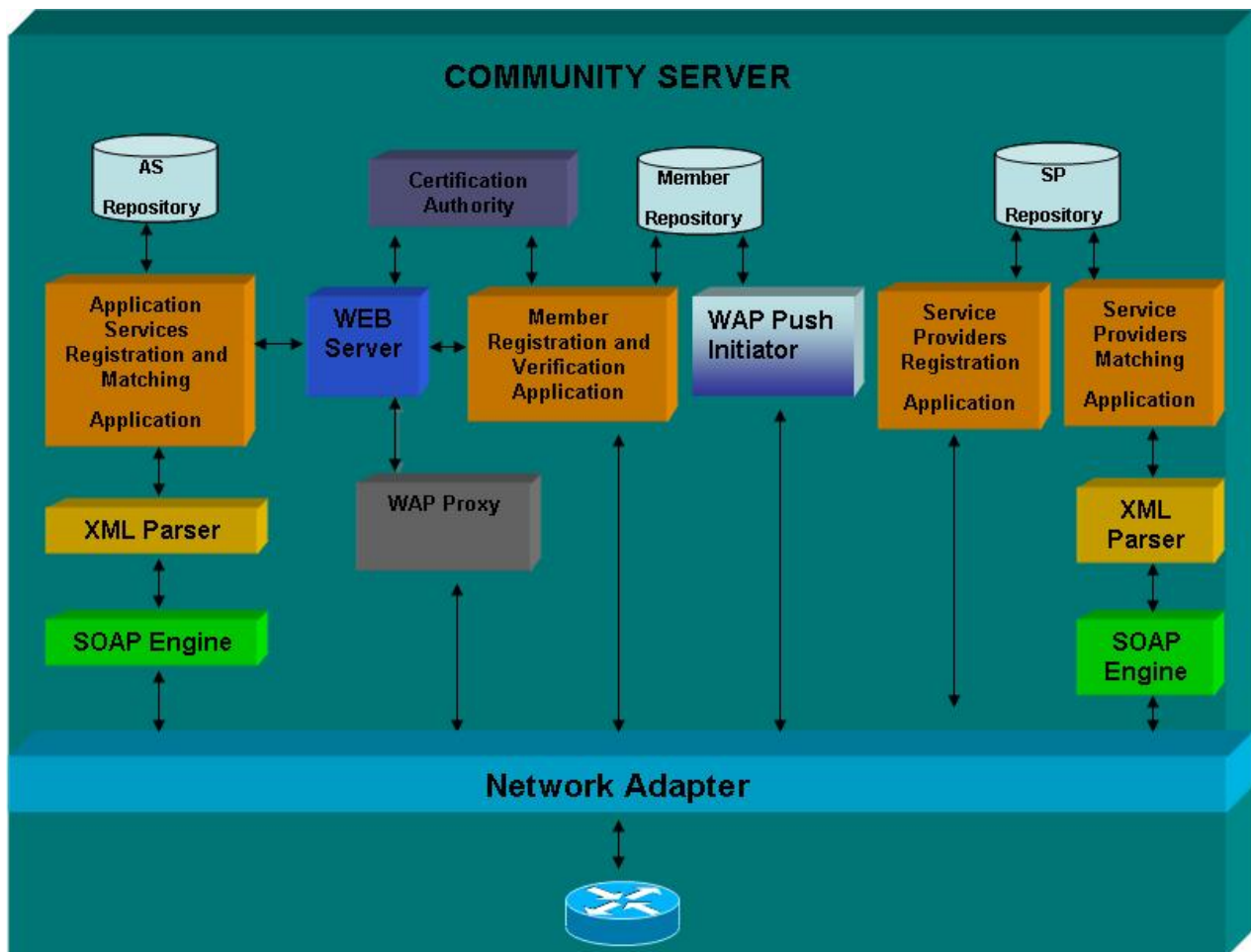
Ο εξυπηρετητής ιστού (Web server) που πρέπει να βρίσκεται στον Community Server αναλαμβάνει την επικοινωνία των SR μέσω μιας διασύνδεσης web. Ένας τέτοιος εξυπηρετητής μπορεί είναι ο Apache Web Server σε συνεργασία με τον Tomcat servlet container (εφόσον η Member Registration and Verification Application υλοποιηθεί σε Java). Όσον αφορά τη δυνατότητα τα μέλη που το επιθυμούν να επικοινωνούν με WAP διασύνδεση, απαιτείται να υπάρχει ένας WAP proxy στον Community Server ούτως ώστε να αναλαμβάνει τη μετατροπή των πρωτοκόλλων από WAP σε WWW και το αντίθετο. Ο WAP proxy του Community Server πρέπει να μπορεί να εξυπηρετεί και τις εφαρμογές-υπηρεσίες που έχουν μόνο web διασυνδέσεις και πρέπει να είναι προσβάσιμες και μέσω WAP διασύνδεσης.

Μια άλλη εφαρμογή που βρίσκεται στον Community Server αναλαμβάνει την επικοινωνία των εφαρμογών-υπηρεσιών με την βάση δεδομένων στην οποία καταχωρούνται οι εφαρμογές υπηρεσίες (Application Services Repository). Η βάση δεδομένων μπορεί να υλοποιηθεί σαν μια UDDI Registry. Έτσι η εφαρμογή (Application Services Registration and Matching Application) πρέπει να μπορεί να λαμβάνει, να επεξεργάζεται, να

δημιουργεί και να αποστέλλει τα κατάλληλα SOAP μηνύματα (βάσει του API που ορίζει το πρωτόκολλο UDDI) που χρειάζονται. Για το σκοπό αυτό πρέπει να έχει εγκατεστημένη μία SOAP Engine, όπως τον Axis εάν η εφαρμογή υλοποιηθεί σε πλατφόρμα Java, η οποία αναλαμβάνει την δημιουργία, αποστολή και επεξεργασία SOAP φακέλων καθώς και την ενθυλάκωση σε αυτούς της πληροφορίας που χρειάζεται. Η πληροφορία που ενθυλακώνεται ακολουθεί τους κανόνες της γλώσσας XML. Δυνατότητα διαχείρισης πληροφορίας XML από εφαρμογές δίνουν εργαλεία όπως ο Xerces και ο Xalan (για υλοποιήσεις σε Java) τα οποία πρέπει να χρησιμοποιεί η εφαρμογή. Ακόμα η εφαρμογή αυτή πρέπει να είναι προσβάσιμη και από τον Web server ούτως ώστε ένας SR να μπορεί μέσω Web/WAP διασύνδεσης να προσπελάζει τις διαθέσιμες εφαρμογές υπηρεσίες (Application Services).

Για την διαθεσιμότητα των παρόχων υπηρεσίας να παρέχουν αν χρειαστεί δεδομένα στον Community Server, απαιτείται η ύπαρξη μιας ακόμα βάσης δεδομένων (Service Providers Repository). Η βάση δεδομένων αυτή μπορεί να υλοποιηθεί από ένα LookUp Service της τεχνολογίας Jini ούτως ώστε να γίνεται δυναμική καταχώρηση και διαγραφή των παρόχων υπηρεσίας. Η εφαρμογή Service Providers Matching Application επικοινωνεί και αναζητά από το Service Providers Repository τους κατάλληλους και διαθέσιμους παρόχους υπηρεσίας και σε συνεργασία με το Network Module στέλνει ειδοποιήσεις στα τερματικά τους χρησιμοποιώντας SMS ή WAP push αναλόγως τις οδηγίες της εφαρμογής υπηρεσίας. Ακόμα αναλαμβάνει να δέχεται τα μηνύματα SOAP από τις διάφορες Application Services που θέλουν πληροφορία, και να στέλνει τις κατάλληλες απαντήσεις περιέχοντας την συλλεγόμενη από τους Service Provider πληροφορία, πάντα χρησιμοποιώντας SOAP μηνύματα. Η εφαρμογή αυτή απαιτεί τη χρήση των ίδιων εργαλείων που αναφέρθηκαν στη προηγούμενη παράγραφο (Axis, Xalan, Xerces).

Όσο αφορά την ενημέρωση των μελών της κοινότητας, αυτό γίνεται είτε με WAP push είτε με SMS μηνύματα αναλόγως της δυνατότητας της συσκευής να υποστηρίξει μια από τις δύο τεχνολογίες. Τέλος από τον Community Server απαιτείται και η ύπαρξη ενός εξυπηρετητή ηλεκτρονικού ταχυδρομείου (e-mail server) ούτως ώστε να γίνεται όλα τα μέλη να έχουν κάποια διεύθυνση ώστε να επικοινωνούν με τους διαχειριστές του Community Server. Διαγραμματικά η δομή του Community Server φαίνεται στο πιο κάτω σχήμα.



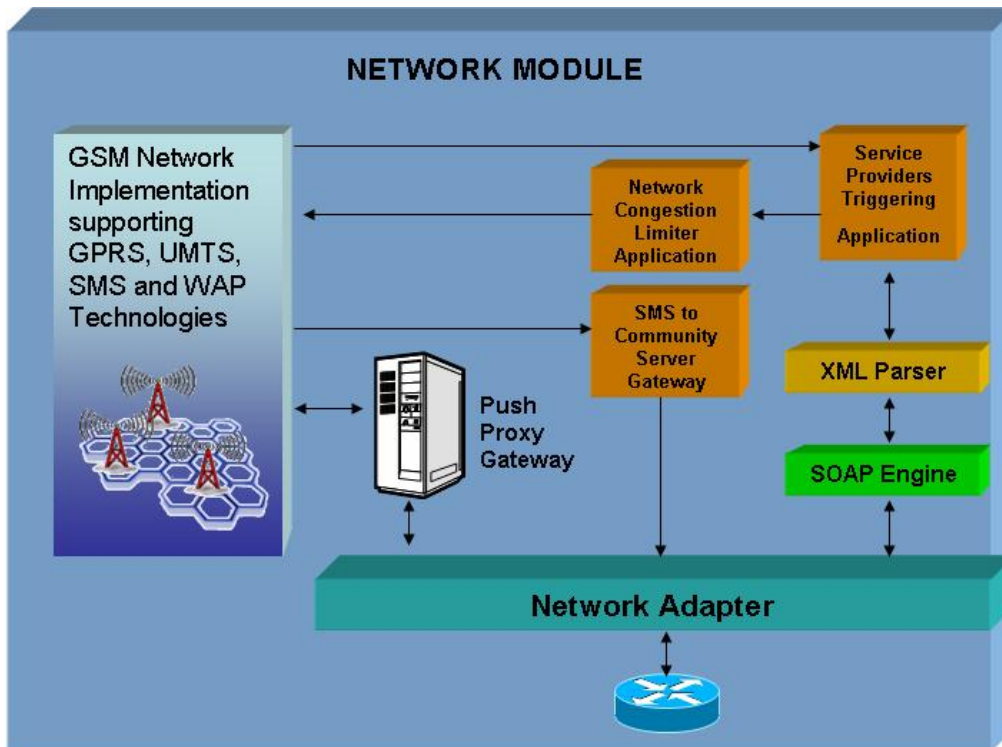
Σχήμα 5. Η δομή του Community Server

3.5.5- To Network Module

Το Network Module αποτελεί την θεμελιώδη υποδομή που επιτρέπει την επικοινωνία των οντοτήτων μεταξύ τους. Όπως και ο Community Server βρίσκεται υπό την διαχείριση του παρόχου του δικτύου κινητών τηλεπικοινωνιών. Οι λειτουργίες που πρέπει να υποστηρίζει το Network Module είναι:

- Υλοποίηση του δικτύου GSM το οποίο υποστηρίζει τις τεχνολογίες GPRS, UMTS, SMS και WAP
- Δυνατότητα αποστολής WAP Push μηνυμάτων.
- Δυνατότητα ειδοποίησης των Service Providers για αποστολή πληροφορίας.
- Μετατροπή μηνυμάτων SMS σε πληροφορία κατανοητή από τον Community Server
- Επικοινωνία με το Community Server μέσω SOAP μηνυμάτων.

Διαγραμματικά η δομή του Network Module φαίνεται στο πιο κάτω σχήμα.

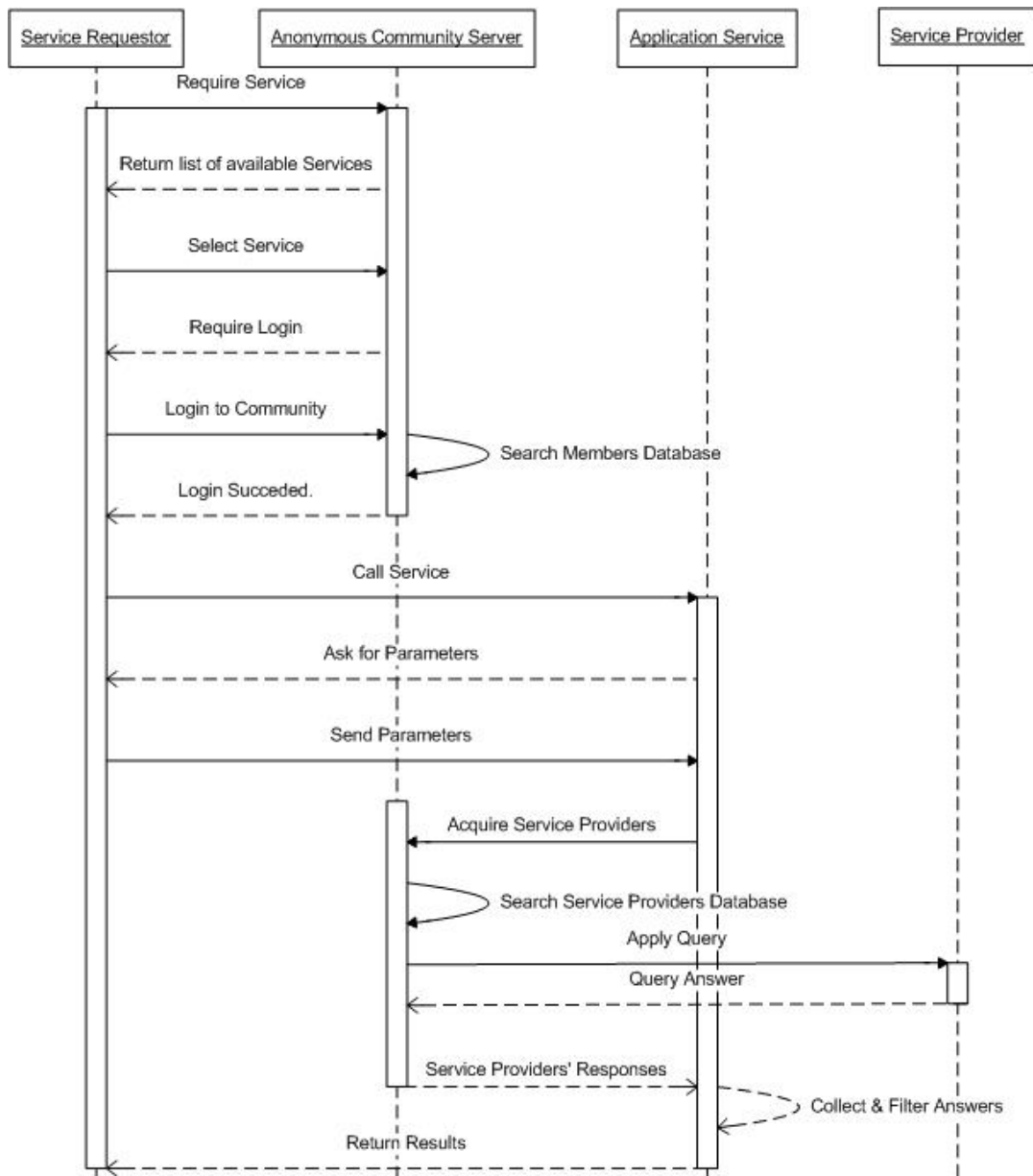


Σχήμα 6 . Η δομή του Network Module

3.6- ΤΑ ΣΕΝΑΡΙΑ

Ακολουθούν τα διαγράμματα ροής της πληροφορίας σε 3 εναλλακτικά σενάρια, που διαρκώς λαμβάνουν χώρα στα πλαίσια της κοινότητας σε επίπεδο υπηρεσιών:

3.6.1- Αίτηση SR για μια Υπηρεσία

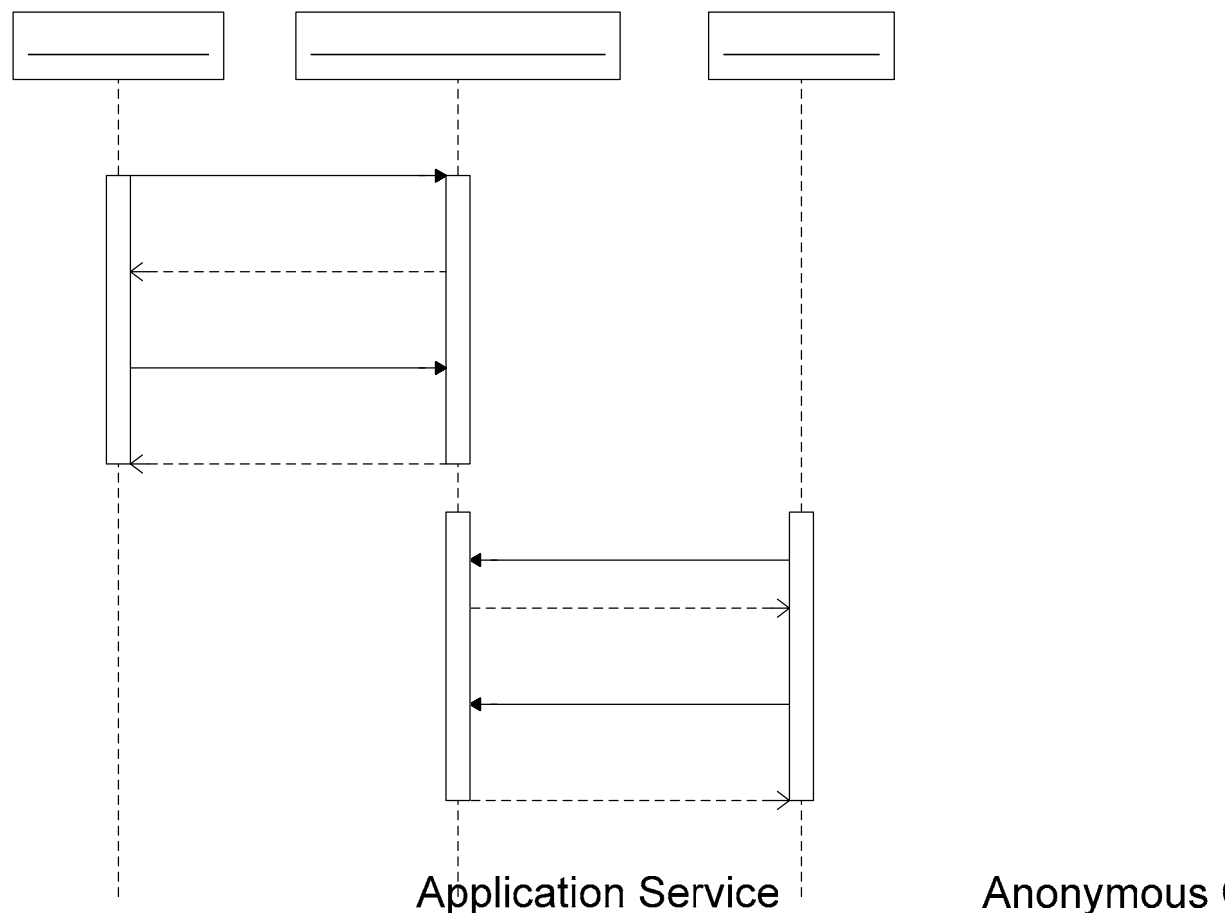


Σχήμα 7. Διάγραμμα διαδικασίας αναζήτησης και χρήσης υπηρεσίας

Όπως φαίνεται και από το πιο πάνω σχήμα, αρχικά ο SR βρίσκεται σε διαδικασία αναζήτησης εφαρμογής-υπηρεσίας. Πολύ πιθανόν να μην γνωρίζει τι ακριβώς θέλει, οπότε με μια λειτουργία αναζήτησης ο CS του επιστρέφει μια λίστα με όλες τις διαθέσιμες εφαρμογές-υπηρεσίες (*List of available Services*) που σχετίζονται με την αναζήτηση του SR. Αφού (και εφόσον) ο SR κάνει μια συγκεκριμένη επιλογή (*Select Service*), θα πρέπει να εισέλθει επιτυχώς στην κοινότητα με τη διαδικασία ανταλλαγής χειραψίας (*Login to Community – Login Succeeded*). Μετά το επιτυχές login, ο SR μπορεί να καλέσει με την εφαρμογή-υπηρεσία (*Call Service*). Αφού σταλούν και οι απαραίτητες (αν υπάρχουν) παράμετροι (*Ask-Send Parameters*) κατόπιν, η AS ψάχνει για τους διαθέσιμους SPs και ειδοποιεί για αυτό το σκοπό το CS (*Acquire Service Providers*). Ο CS αφού πρώτα ενημερωθεί από την αρμόδια βάση δεδομένων (*Search Service Providers Database*) για τους διαθέσιμους πάροχους υπηρεσίας της εφαρμογής-υπηρεσίας τότε

δέχονται σχετική ερώτηση (*Apply Query*) στις τερματικές συσκευές τους και καλούνται να παράσχουν την απαραίτητη πληροφορία. Η τελευταία είτε λαμβάνεται αυτόματα από το κινητό (σε περίπτωση που η πληροφορία ενυπάρχει στο κινητό όπως λ.χ. GPS στίγμα), είτε ο ίδιος ο χρήστης χρειάζεται να την εισάγει (*Query Answer*). Όλες οι απαντήσεις των SPs συγκεντρώνονται τοπικά στο CS (Service Providers' Responses) και κατόπιν στέλνονται και επεξεργάζονται από την AS (*Collect & Filter answers*), η οποία και προωθεί τα αποτελέσματα στον αρχικό SR (*Return Results*). Αυτή είναι η ροή της πληροφορίας, από τη στιγμή της αρχικής ερώτησης του SR μέχρι την έλευση των αποτελεσμάτων που επιστρέφονται από τους SPs και αφού έχουν υποστεί επεξεργασία από την AS.

3.6.2- Δημοσίευση μιας υπηρεσίας από SP και AS

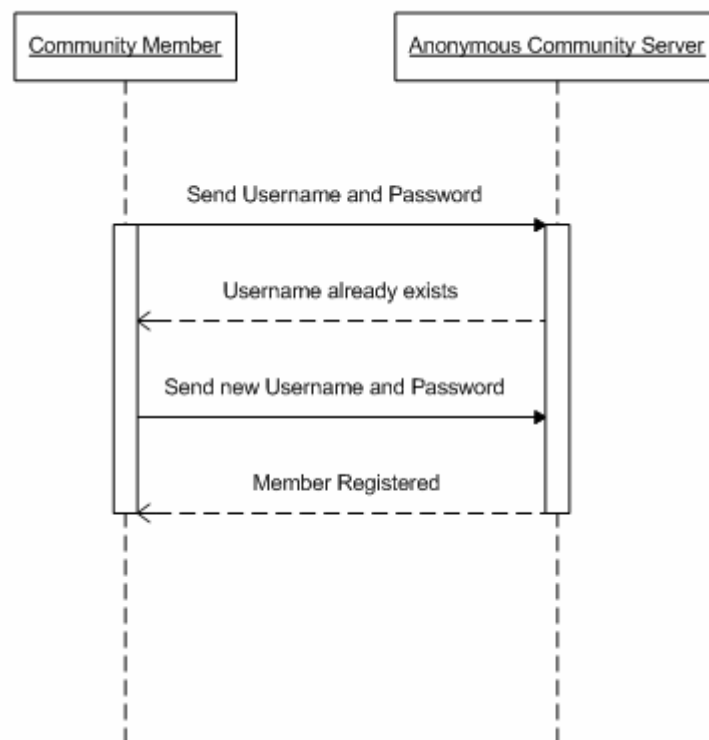


Σχήμα 8. Διάγραμμα διαδικασίας εγγραφής υπηρεσιών

Όπως φαίνεται και από το UML σχήμα, αρχικά η Application Service θα πρέπει να δηλώσει την παρουσία της στον CS. Ανάλογα με την υλοποίηση της πλατφόρμας του προτεινόμενου μοντέλου **Login to Community** να φιλοξενείται μέσα στον CS είτε όχι, γεγονός που καθιστά την ανταλλαγή χειραψίας επιβεβλημένη (*Login to Community – Login Succeeded*). Αφού λοιπόν η AS δημοσιευθεί (*Register Application Service – Service*

Registered), είναι πλέον έτοιμη να δρομολογήσει αμφίδρομη κίνηση μεταξύ SR και SPs. Πιο συγκεκριμένα, θα είναι η AS αυτή που θα λαμβάνει την ευθύνη να είναι ενήμερη σχετικά με τους διαθέσιμους κάθε στιγμή SPs, ώστε να τους υποβάλλει ερωτήσεις διαμέσου του CS, κάθε φορά που θα είναι απαραίτητο. Την στιγμή που κάποιο μέλος επιθυμεί να δηλώσει τη διαθεσιμότητά του ως SP της συγκεκριμένης εφαρμογής-υπηρεσίας, και αφού εισέλθει πρώτα επιτυχώς στην κοινότητα (*Login to Community – Login Succeeded*), λαμβάνει χώρα το registration του service απευθείας στον CS (*Register Service – Service Registered*). Με αυτόν τον τρόπο ολοκληρώνεται επιτυχώς η διαδικασία κοινοποίησης της διαθεσιμότητας ενός SP για μια ήδη δημοσιευμένη εφαρμογή-υπηρεσία (*Registered Application Service*). Πλέον, μπορεί να ικανοποιηθεί μια αίτηση ενός SR καθώς για την επιλεχθείσα του υπηρεσία, υπάρχει ήδη τουλάχιστον ένας SP που δύναται να παράσχει σχετική πληροφορία.

3.6.3- Διαδικασία εγγραφής ενός μέλους στην κοινότητα (registration)



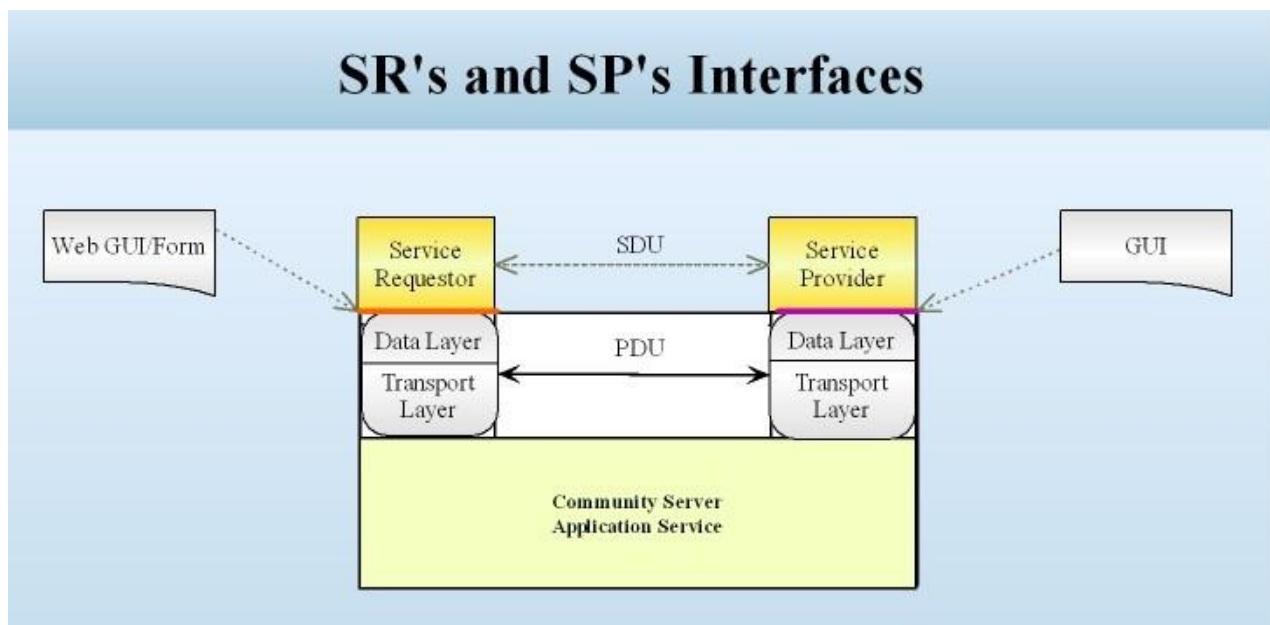
Σχήμα 9. Διάγραμμα διαδικασίας εγγραφής μέλους

Η διαδικασία εγγραφής είναι απαραίτητη και προαπαιτείται ώστε ο χρήστης να γίνει πιστοποιημένο μέλος της κοινότητας. Ως τέτοιο, μπορεί να προσφέρει και να τυγχάνει των ευκολιών που άπτονται της κοινότητας. Το registration απαιτεί να καθορίσει αρχικά ο χρήστης το Username με το οποίο θα είναι γνωστός στην κοινότητα και ένα Password ώστε να διασφαλίζεται η ακεραιότητα της δικτυακής του ταυτότητας (*Send Username and Password*). Στο ενδεχόμενο όπου το Username ήδη υπάρχει, η διαδικασία registration επιστρέφει στον χρήστη μήνυμα *Username already exists*, και ο τελευταίος καλείται να ξαναπροσπαθήσει ώστε να δεσμεύσει Username και Password που δεν υπάρχουν ήδη (*Send new Username and*

Password). Μόλις επιτευχθεί κάτι τέτοιο η CS καλωσορίζει το νέο μέλος της με σχετικό μήνυμα επιτυχούς ολοκλήρωσης της διαδικασίας registration (*Member Registered*). Η διαδικασία του registration λαμβάνει χώρα μια μόνο φορά. Ένα ήδη εγγεγραμμένο μέλος πιστοποιεί την ταυτότητά του μέσω της διαδικασίας login που καλείται να ακολουθήσει κάθε φορά που αποπειράται να εισέλθει στην κοινότητα (σχετικοί clients για τα κινητά τερματικά μπορούν να αυτοματοποιήσουν αυτήν τη διαδικασία).

3.6.4- Οι Διεπαφές των SRs' και SPs'

Τόσο ο Service Requestor όσο και ο Service Provider πρέπει να έχουν μια κατάλληλη διεπαφή ώστε να επικοινωνούν με την κοινότητα και να εκτελούν τις επιθυμητές τους διεργασίες. Στο ακόλουθο σχήμα φαίνονται οι θέσεις των διεπαφών πάνω στη στοίβα των στρωμάτων/ πρωτοκόλλων.



Σχήμα 10. SR's και SP's Διεπαφές (στη στοίβα πρωτοκόλλων) και μεταξύ τους επικοινωνία

Οι SR και SP αποτελούν ομότιμες οντότητες (peer entities) και βρίσκονται στο υψηλότερο σκαλοπάτι των στρωμάτων της πλατφόρμας. Η επικοινωνία τους με την πλατφόρμα με την βοήθεια γραφικής διεπαφής χρήστη (GUI). Τα GUIs θα αναλαμβάνουν να μεταφράζουν την λαμβανόμενη πληροφορία στα χαμηλότερα στρώματα της πλατφόρμας χωρίς ο χρήστης να ανησυχεί για αυτό. Η ροή της πληροφορίας μπορεί να είναι τόσο ημιαμφίδρομη (και προς τις δυο κατευθύνσεις αλλά ποτέ ταυτόχρονα) όσο και αμφίδρομη (ταυτόχρονα και προς τις δυο κατευθύνσεις). Οι SR και SP δεν επικοινωνούν άμεσα. Η επικοινωνία τους είναι νοητή γι' αυτό και στο σχήμα συμβολίζεται με διακεκομμένη γραμμή. Μεταξύ τους ανταλλάζουν SDUs (Service Data Units) δηλαδή Μονάδες Δεδομένων Υπηρεσίας. Η πραγματική επικοινωνία πραγματοποιείται πιο χαμηλά στην στοίβα και αυτό αναπαρίσταται με ανταλλαγή PDUs (Protocol Data Units) δηλαδή Μονάδες Δεδομένων Πρωτοκόλλου.

Ο Service Requestor θα πρέπει να επικοινωνεί πολύ εύκολα και απλά με τον Community Server. Όπως έχει ήδη αναφερθεί, στην αρχή δεν απαιτείται να πιστοποιήσει την ταυτότητά του στην κοινότητα για να έχει πρόσβαση στις τρέχουσες διαθέσιμες υπηρεσίες. Ένας οποιοδήποτε χρήστης κινητής συσκευής μπορεί να πραγματοποιήσει τα πρώτα βήματα εύρεσης/ αναζήτησης επιθυμητής υπηρεσίας και μετέπειτα να επιβεβαιώσει την ταυτότητά του ώστε να λάβει τον ρόλο του SR. Η ευκολία αυτή στην πρόσβαση- αναζήτηση πραγματοποιείται με μια web φόρμα ή γενικότερα ένα web based GUI. Πιθανές σύγχρονες τεχνολογίες που μπορούν να στηρίξουν το web GUI μοντέλο, παρουσιάζονται σε επόμενα κεφάλαια.

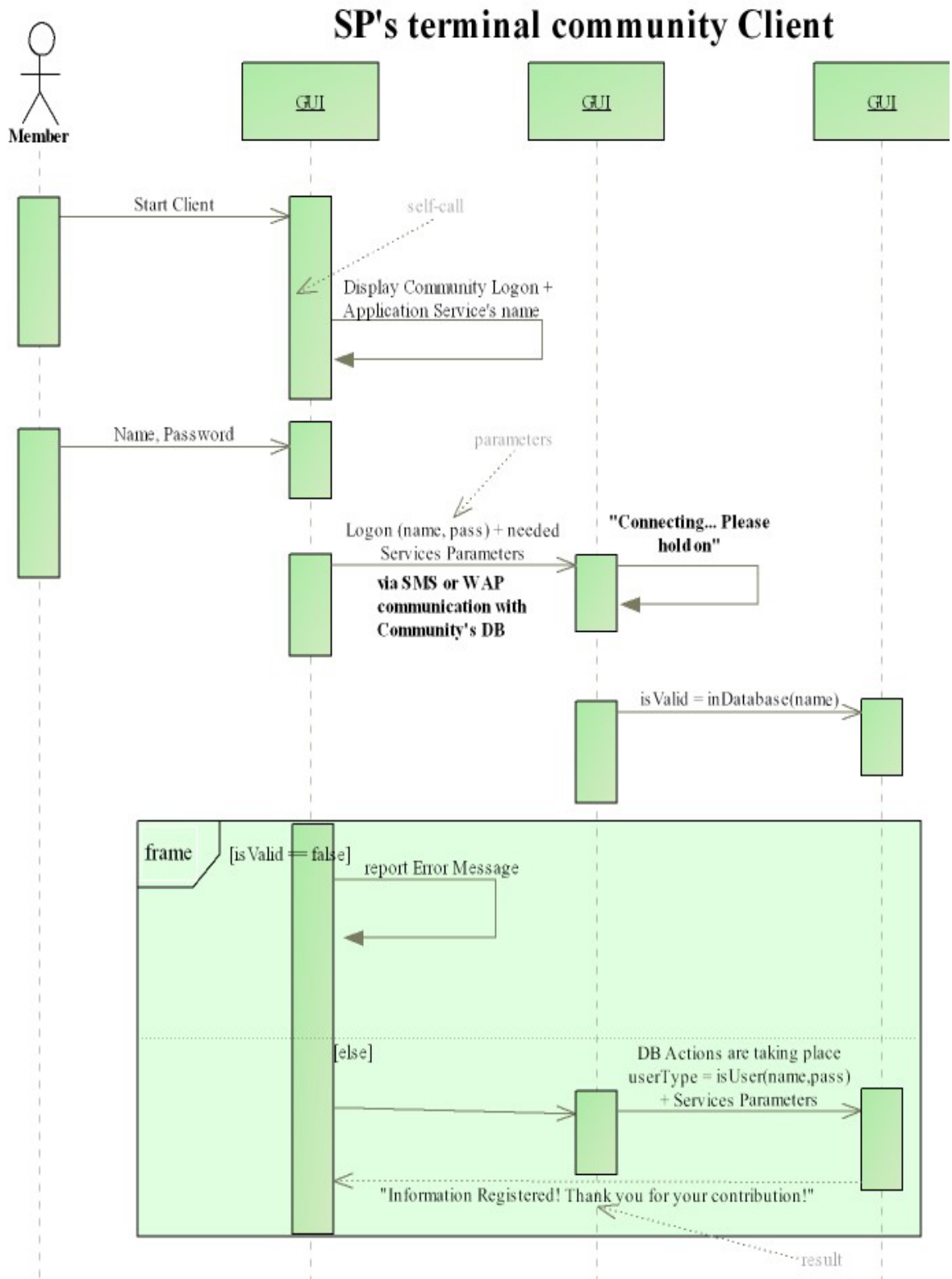
Ο Service Provider, αφού επιλέξει την υπηρεσία στην οποία θα συμμετάσχει ως πάροχος πληροφορίας, κατεβάζει στο τερματικό του τον απαραίτητο client. Μέσω αυτού θα εισάγει όλα τα απαραίτητα στοιχεία που απαιτούνται από τη διαδικασία συλλογής πληροφοριών. Η ύπαρξη του κατάλληλου client στο τερματικό ενός μέλους, χρήζει το τελευταίο ουσιαστικά Service Provider. Ο Client επιτελεί και ακόμα μια εργασία: η ενεργοποίηση του στο τερματικό του Service Provider ισοδυναμεί με *τρέχουσα διαθεσιμότητα* του SP, γεγονός για το οποίο ενημερώνεται ο Community Server. Όταν το μέλος πάψει να επιθυμεί τη συνδρομή του στην υπηρεσία, απενεργοποιώντας τον client εμμέσως ενημερώνει και τον CS για την επιθυμία του αυτή λόγω του ότι δεν θα ανανεώσει την διαθεσιμότητα του μετά το πέρας του χρόνου ανανέωσης (leasing) που καθορίζει ο Community Server. Τότε ο CS διαγράφει το συγκεκριμένο μέλος από τη Βάση Δεδομένων που διατηρεί με τους τρέχοντες παροχείς υπηρεσιών.

Πάνω απ' όλα όμως ο client δεν παύει να είναι μια γραφική διεπαφή χρήστη. Πρόκειται δηλαδή για ένα γραφικό περιβάλλον που εγκαθίσταται στο τερματικό του εκάστοτε ενδιαφερόμενου. Μέσω αυτού ο SP επικοινωνεί με τον CS και εμμέσως με την AS και τον SR. Η γραφική διεπαφή αυτή θα πρέπει να είναι προσεκτικά σχεδιασμένη ώστε με αποδοτικό, ευπαρουσίαστο, ακριβή και απλό τρόπο να αντλεί την απαραίτητη πληροφορία από τον SP.

Πιο συγκεκριμένα, θα πρέπει αρχικά να γίνει μια *Ανάλυση απαιτήσεων*. Αυτή υπαγορεύεται από την υπολογιστική δύναμη των κινητών τερματικών που χρησιμοποιούν οι χρήστες- SPs. Έπειτα ακολουθεί η *Σχεδίαση για το χρήστη*. Εκεί όπου χρησιμοποιείται GUI θα πρέπει να γίνει ορθή *Δομή της επιφάνειας χρήσης*: μενού, διάλογοι, διακόπτες ελέγχου, ρυθμιστές παραμέτρων. Επίσης διάφορες άλλες *Διαδραστικές λειτουργίες* θα πρέπει να λαμβάνονται υπόψη: έλεγχοι, πολλαπλές επιλογές, επιβεβαιώσεις. Τέλος θα πρέπει να εστιαστεί η προσοχή στην *Συνολική παρουσίαση της επιφάνειας χρήσης*: εμφάνιση της οθόνης, χρώματα καθώς και στην μελλοντική *Υποστήριξη του χρήστη*.

Τα αρχεία των γραφικών θα πρέπει να είναι σε κατάλληλη μορφή ώστε να προσαρμόζονται αυτόματα σε κάθε μέγεθος οθόνης θέασης/ παρουσίασης (bitmap files and vector files). Ακόμα, δεδομένου του περιορισμένου τηλεπικοινωνιακού διαύλου θα πρέπει να μεριμνηθεί

κατάλληλη συμπίεση των δεδομένων/ γραφικών (Run-Length-Coding, Lempel-Ziv coding, Huffman coding, GIF, JPEG, JBIG).



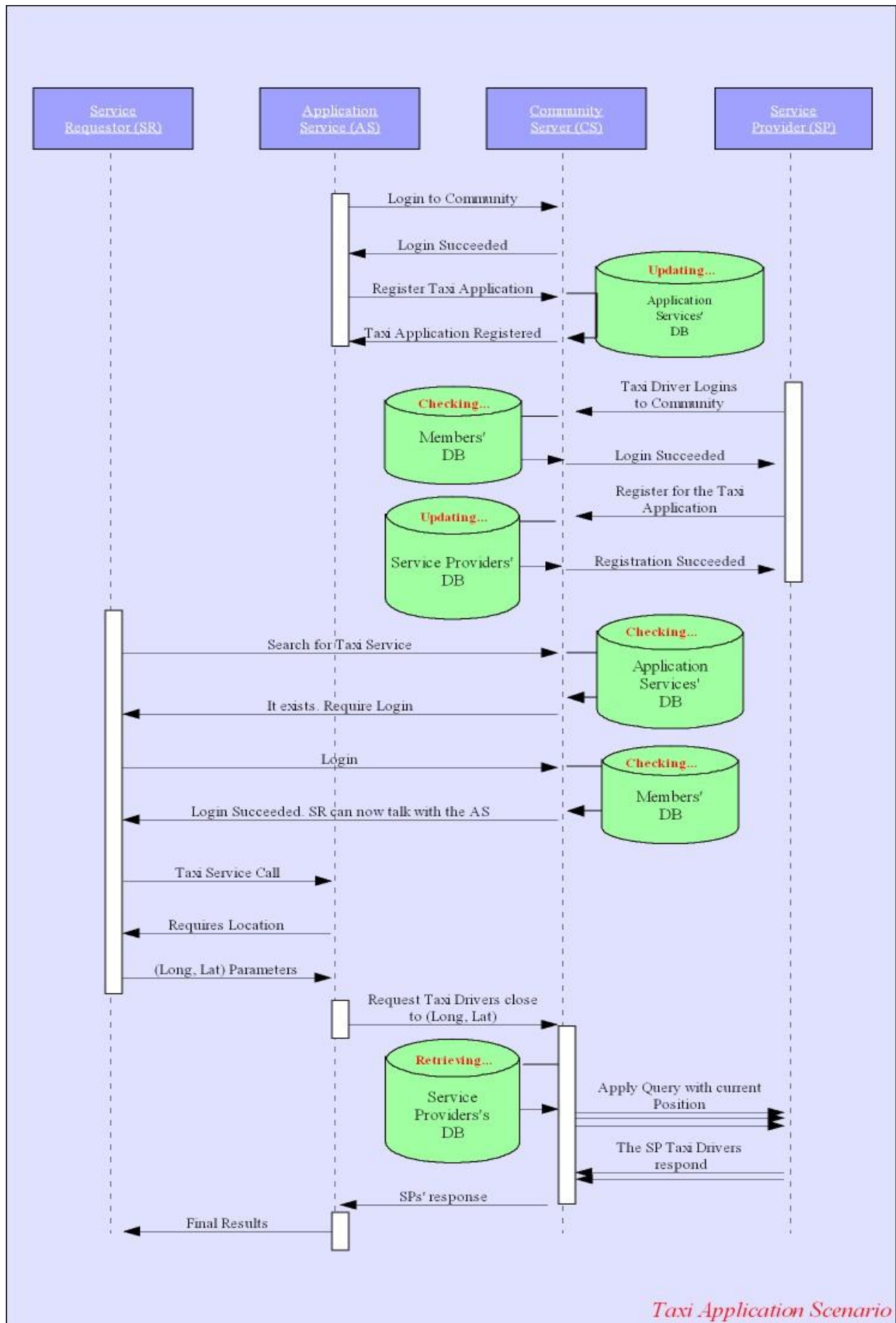
Σχήμα 11. Δείγμα Ροής Πληροφορίας στον Client του SP

3.7- ΠΙΘΑΝΕΣ ΕΦΑΡΜΟΓΕΣ ΒΑΣΙΣΜΕΝΕΣ ΣΤΗ ΠΡΟΤΕΙΝΟΜΕΝΗ ΙΔΕΑ

Παρακάτω περιγράφονται επιγραμματικά, ενδεχόμενα σενάρια-πιθανές εφαρμογές, όπου το προτεινόμενο μοντέλο επικοινωνίας/ανταλλαγής υπηρεσιών βρίσκει εφαρμογή και βάση του οποίου παρέχονται οι εφαρμογές αυτές.

3.7.1- Υπηρεσία εύρεσης Ταξί

Διαγραμματικά, η διαδικασία αναζήτησης και εύρεσης ταξί φαίνεται στο πιο κάτω σχήμα.

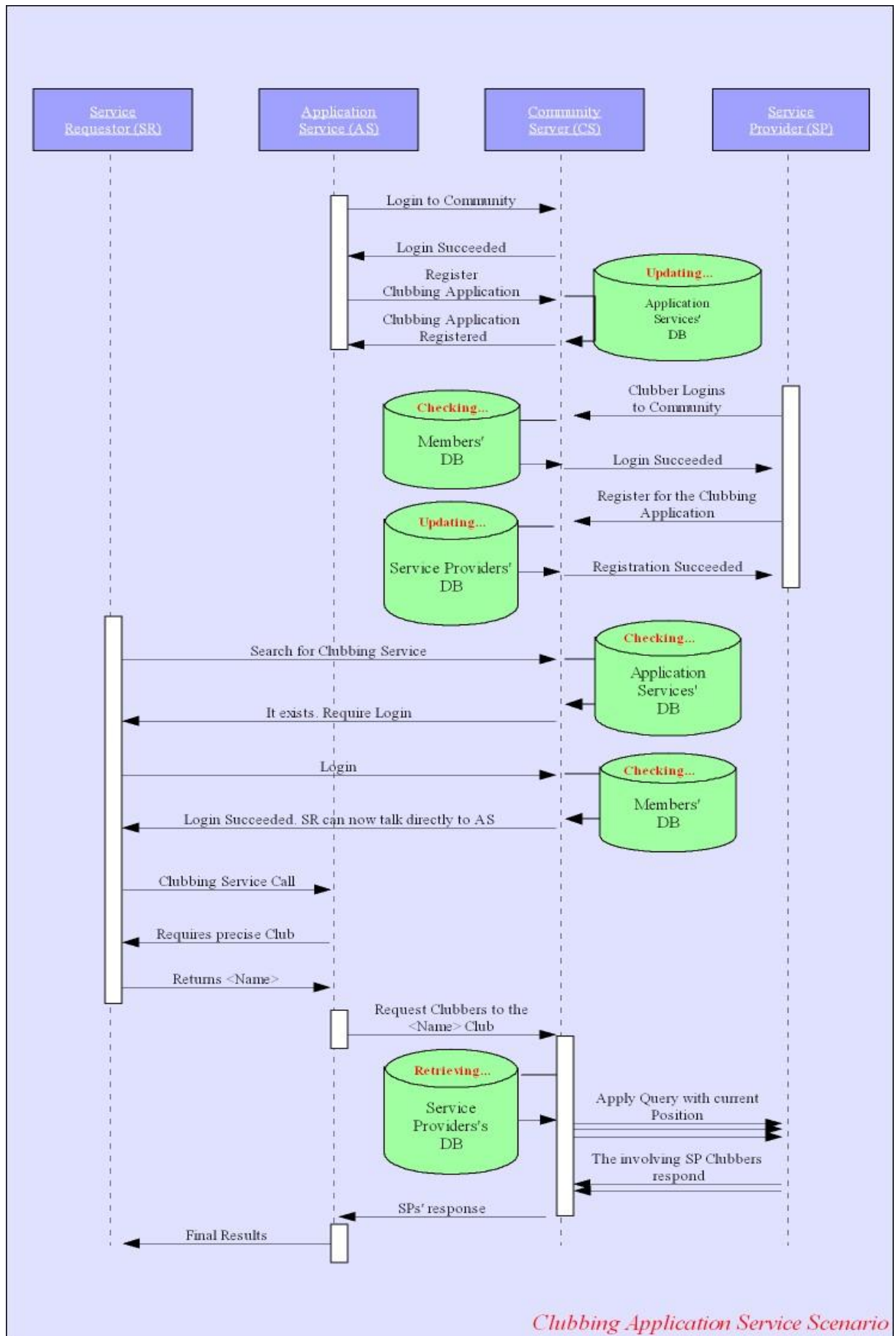


Σχήμα 12. Διάγραμμα διαδικασίας εύρεσης ταξί

Ανεξάρτητος οργανισμός (πιθανότατα ο operator) δημιουργεί μια υπηρεσία που σκοπός της είναι να προσφέρει άμεση εύρεση κοντινότερου διαθέσιμου ταξί για οποιονδήποτε ενδιαφερόμενο. Η ιδέα στηρίζεται στο ότι οι ταξιτζήδες αποτελούν τους SPs ενώ αυτοί που αναζητούν ταξί είναι οι SRs. Αρχικά, η υπηρεσία εύρεσης Ταξί πρέπει να δηλωθεί στην κοινότητα και πιο συγκεκριμένα στον CS. Ο CS ανανεώνει τη λίστα του με τις τρέχουσες διαθέσιμες υπηρεσίες, την οποία και προωθεί σε οποιονδήποτε ενδιαφερόμενο SR. Ακολούθως ο CS αναμένει και δέχεται αιτήσεις από ενδιαφερόμενους ταξιτζήδες- μέλη που επιθυμούν να συμμετάσχουν στην υπηρεσία ως SPs. Το μόνο που έχουν να κάνουν είναι να συνδεθούν με την κοινότητα και να ενημερώσουν τον CS για το ενδιαφέρον τους. Αυτό γίνεται μέσω ειδικού client που βρίσκεται εγκατεστημένος στο τερματικό του SP. Μέσω αυτού δηλώνεται η ενεργή παρουσία του SP για την υπηρεσία και γίνεται εισαγωγή των απαραίτητων παραμέτρων. Ο CS τότε ανανεώνει τη Βάση Δεδομένων με τους SPs και τις αντίστοιχες παραμέτρους τους, για κάθε τρέχουσα υπηρεσία που διατηρεί, και είναι έτοιμος να τους καλέσει κάθε φορά που η υπηρεσία το απαιτεί. Τέλος, έχουμε την ίδια τη διαδικασία αναζήτησης ταξί: ενδιαφερόμενος SR αναζητεί στον CS για υπηρεσία που σχετίζεται με εύρεση ταξί. Ο CS αποκρίνεται πώς υπάρχει διαθέσιμη μια τέτοια υπηρεσία και ζητά από τον SR να εξακριβώσει τα στοιχεία του ώστε να συνεχιστεί η διαδικασία. Αφού ο SR πιστοποιήσει ότι είναι μέλος της κοινότητας, εξουσιοδοτείται να επικοινωνήσει με την ίδια την υπηρεσία εύρεσης Ταξί. Η υπηρεσία του ζητά τη γεωγραφική του θέση, δεδομένο με το οποίο απαντά ο SR (π.χ. Longitude, Latitude). Ακολούθως, η υπηρεσία επικοινωνεί με τον CS και του αναθέτει να επικοινωνήσει με τους SPs της υπηρεσίας, ώστε να βρεθεί ο κοντινότερος διαθέσιμος ταξιτζής στο γεωγραφικό στίγμα (Long, Lat). Ο CS επικοινωνεί με τους τρέχοντες SPs της υπηρεσίας εύρεσης Ταξί και ενημερώνει την υπηρεσία με τα αποτελέσματα. Κάτι τέτοιο μπορεί να διευθετείται και από την επικοινωνία των SPs' clients με τον CS. Τέλος, η ίδια η υπηρεσία επιστρέφει στον SR εκείνον τον διαθέσιμο SP- ταξιτζή που τον εξυπηρετεί καλύτερα.

3.7.2- Υπηρεσία εύρεσης κίνησης σε Clubs

Διαγραμματικά, η διαδικασία αναζήτησης και εύρεσης της κίνησης σε clubs φαίνεται στο πιο κάτω σχήμα.

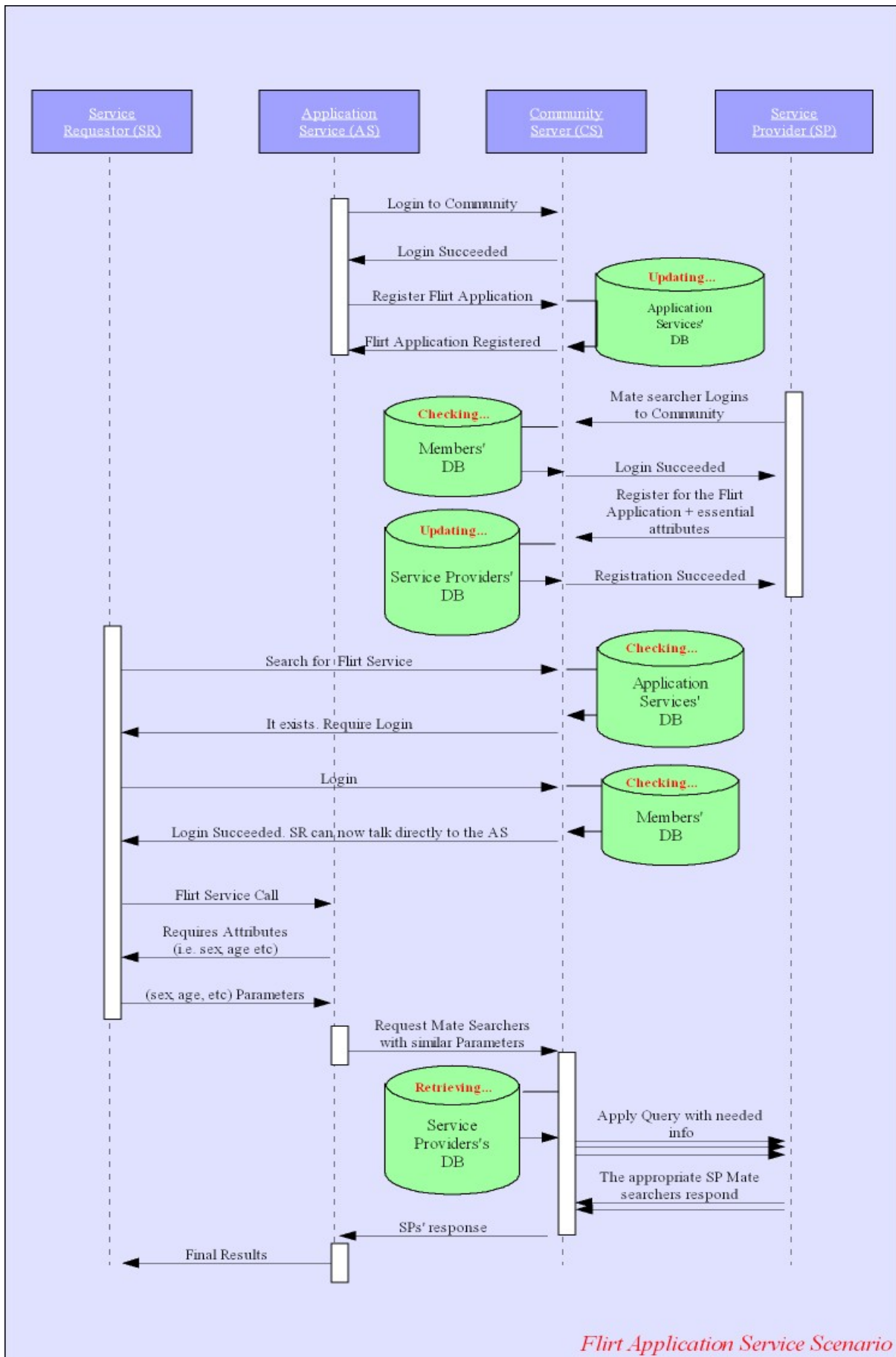


Σχήμα 13. Διάγραμμα διαδικασίας εύρεσης κίνησης σε club

Τόσο ο operator όσο και μια σύμπραξη των νυχτερινών Clubs μπορούν να διαθέσουν αυτή την υπηρεσία στα μέλη της κοινότητας. Η υπηρεσία στηρίζεται στο ότι οι Clubbers- μέλη της κοινότητας γίνονται SPs και μπορούν να ενημερώσουν σχετικά με την τρέχουσα κίνηση και τη “ζωντάνια” μέσα στο Club, κάθε ενδιαφερόμενο SR. Στην αρχή, θα πρέπει η υπηρεσία να δηλωθεί στον CS ώστε να είναι διαθέσιμη και να εμφανίζεται ως τρέχουσα σε οποιαδήποτε κλήση των SRs. Πλέον, ο CS αναμένει τη συμμετοχή των SPs ώστε να μπορεί να λειτουργήσει η υπηρεσία. Οι Clubbers-μέλη της κοινότητας που προθυμοποιούνται να λάβουν μέρος ως SPs, κατά την επίσκεψή τους στα βραδινά κέντρα διασκέδασης, έχουν ενεργοποιήσει τον αντίστοιχο client και περιμένουν ερώτηση ώστε να απαντήσουν. Ο SR είναι ένας οποιοσδήποτε χρήστης που αναζητεί υπηρεσία σχετικά με Clubbing στην κοινότητα. Η διαδικασία αναζήτησης τον οδηγεί στην Υπηρεσία εύρεσης κίνησης σε Clubs, την οποία και αποφασίζει να χρησιμοποιήσει. Ο CS του ζητά να πιστοποιήσει πως είναι εγγεγραμμένο μέλος της κοινότητας. Αφού εισέλθει στην κοινότητα, ο SR μπορεί πλέον να μιλήσει απευθείας με την Υπηρεσία εύρεσης κίνησης σε Clubs. Υποτίθεται πως ο SR θα ζητήσει πληροφορίες για κάποια συγκεκριμένα Clubs που έχει ως πιθανούς προορισμούς. Το όνομα του target- Club είναι αρκετό για την υπηρεσία, η οποία και το προωθεί στον CS. Ο CS είναι πλέον υπεύθυνος να αναζητήσει μεταξύ των SPs της υπηρεσίας, εκείνους που βρίσκονται στο target- Club και να τους υποβάλλει τις απαραίτητες ερωτήσεις μέσω των clients τους. Η πληροφορία που επιστρέφεται/ συλλέγεται από τους εμπλεκόμενους SPs αντικατοπτρίζει την κατάσταση του κόσμου στο target- Club εκείνη τη συγκεκριμένη στιγμή (real- time service). Τα αποτελέσματα πακετάρονται και αποστέλλονται στην Υπηρεσία εύρεσης κίνησης σε Clubs. Η τελευταία, αφού επεξεργαστεί τη λαμβανόμενη πληροφορία, επιστρέφει τα αποτελέσματα στον SR με τρόπο εύληπτο, συνοπτικό και κυρίως κατανοητό.

3.7.3- Υπηρεσία εύρεσης Συντροφιάς (Flirt Service)

Διαγραμματικά, η διαδικασία αναζήτησης και εύρεσης της κίνησης σε club φαίνεται στο πιο κάτω σχήμα.

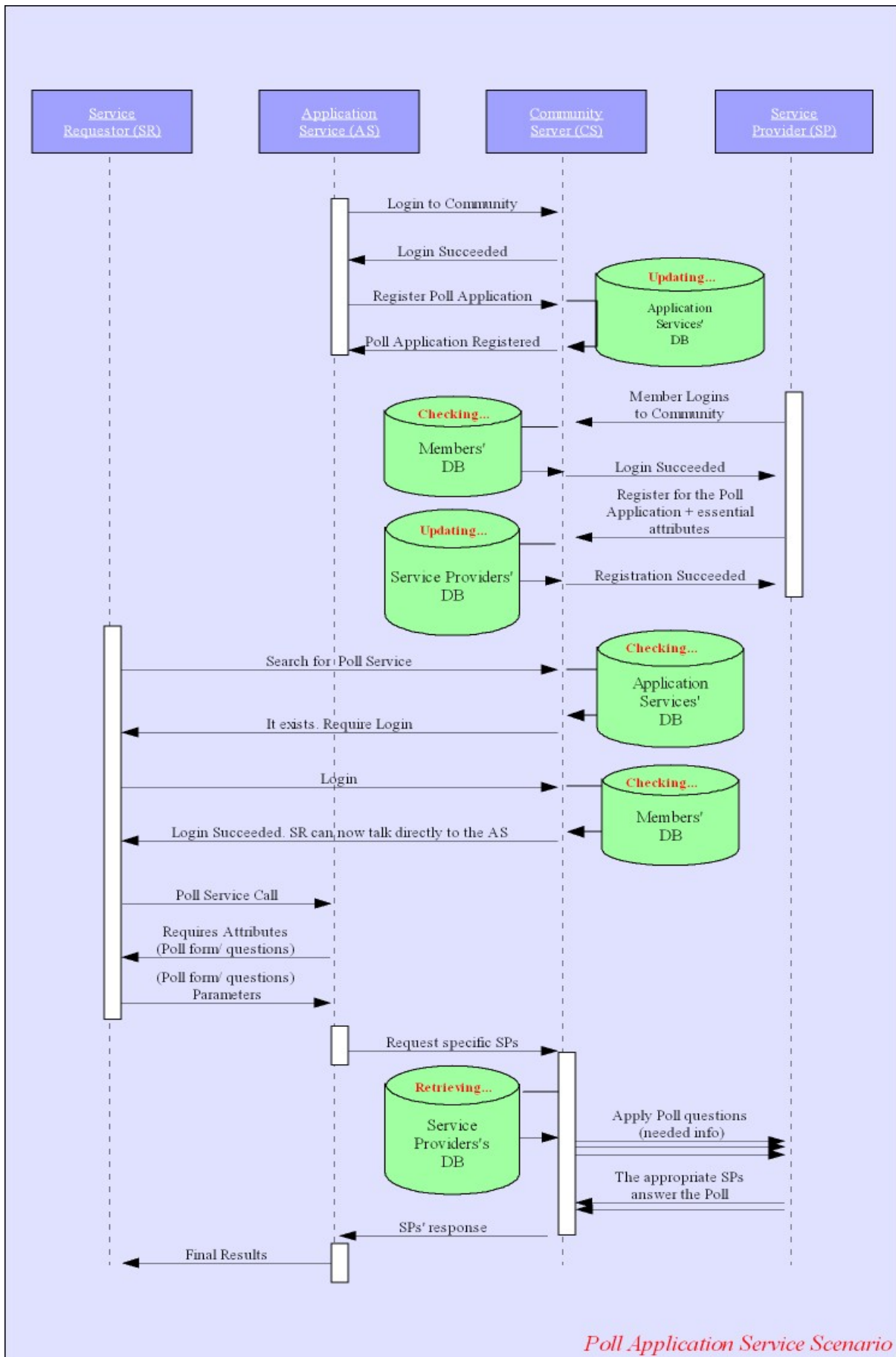


Σχήμα 14. Διάγραμμα διαδικασίας εύρεσης συντροφιάς

Αρχικά η Υπηρεσία Εύρεσης Συντροφιάς θα πρέπει να δηλωθεί στον CS ώστε να είναι διαθέσιμη και να φαίνεται στις τρέχουσες υπηρεσίες της κοινότητας. Όσα από τα μέλη της κοινότητας επιθυμούν να λάβουν μέρος στην υπηρεσία μπορούν πλέον να το κάνουν ως SP της υπηρεσίας. Αφού πρώτα πιστοποιήσουν πώς είναι μέλη της κοινότητας, εγγράφονται ως SP της υπηρεσίας, με την οποία και θα επικοινωνούν εφεξής διαμέσου ειδικού client. Χάρη στον client, οι SP μπορούν να εισάγουν τα απαραίτητα στοιχεία που απαιτούνται από την υπηρεσία (π.χ. ηλικία, γένος, ενδιαφέροντα, κ.α.) και να καταγράφονται έτσι στην κατάλληλη Βάση Δεδομένων που διατηρεί ο CS. Με την ενεργοποίηση του client επιτυγχάνονται δύο πράγματα: 1) εισάγονται οι απαραίτητες παράμετροι από τους SPs, 2) για όσο διάστημα ο client λειτουργεί στο τερματικό του SP, εξασφαλίζεται επικοινωνία του SP με τον CS (με άλλα λόγια, για όσο εκτελείται ο client, δηλώνεται ταυτόχρονα και η διαθεσιμότητα του SP). Έπειτα ακολουθεί η ίδια η διαδικασία αναζήτησης συντρόφου: ενδιαφερόμενος SR αναζητεί στον CS την ύπαρξη υπηρεσία που να σχετίζεται με εύρεση συντροφιάς. Ο CS αποκρίνεται πως διαθέτει τέτοια τρέχουσα υπηρεσία και ζητά από τον SR να εξακριβώσει τα στοιχεία του. Αφού πιστοποιηθεί η ταυτότητά του, ο SR έχει εξουσιοδοτηθεί να μιλήσει απευθείας με την Υπηρεσία εύρεσης συντροφιάς. Η Υπηρεσία αυτή του ζητά τις απαραίτητες παραμέτρους που απαιτούνται ανάλογα με το ταίρι που αναζητεί να βρει. Ο SR εισάγει όλα τα απαραίτητα και αναμένει. Η υπηρεσία τότε απευθυνόμενη στον CS, του δίνει εντολή να αναζητήσει όλους τους τρέχοντες SPs που πληρούν ή καλύπτουν επαρκώς τα χαρακτηριστικά του SR. Ο CS τότε, ελέγχοντας την κατάλληλη Βάση Δεδομένων του, επικοινωνεί με επιλεγμένους SPs της υπηρεσίας, και πιο συγκεκριμένα με τους clients που τρέχουν στις τερματικές συσκευές τους. Αφού λάβει τις απαντήσεις, ο CS τις πακετάρει και τις αποστέλλει για επεξεργασία στην AS της Flirt υπηρεσίας. Η τελευταία, αφού ολοκληρώσει τις απαραίτητες διεργασίες αποστέλλει τα αποτελέσματα στον αναμένοντα SR.

3.7.4- Υπηρεσία δημοσκοπήσεων (Poll Service)

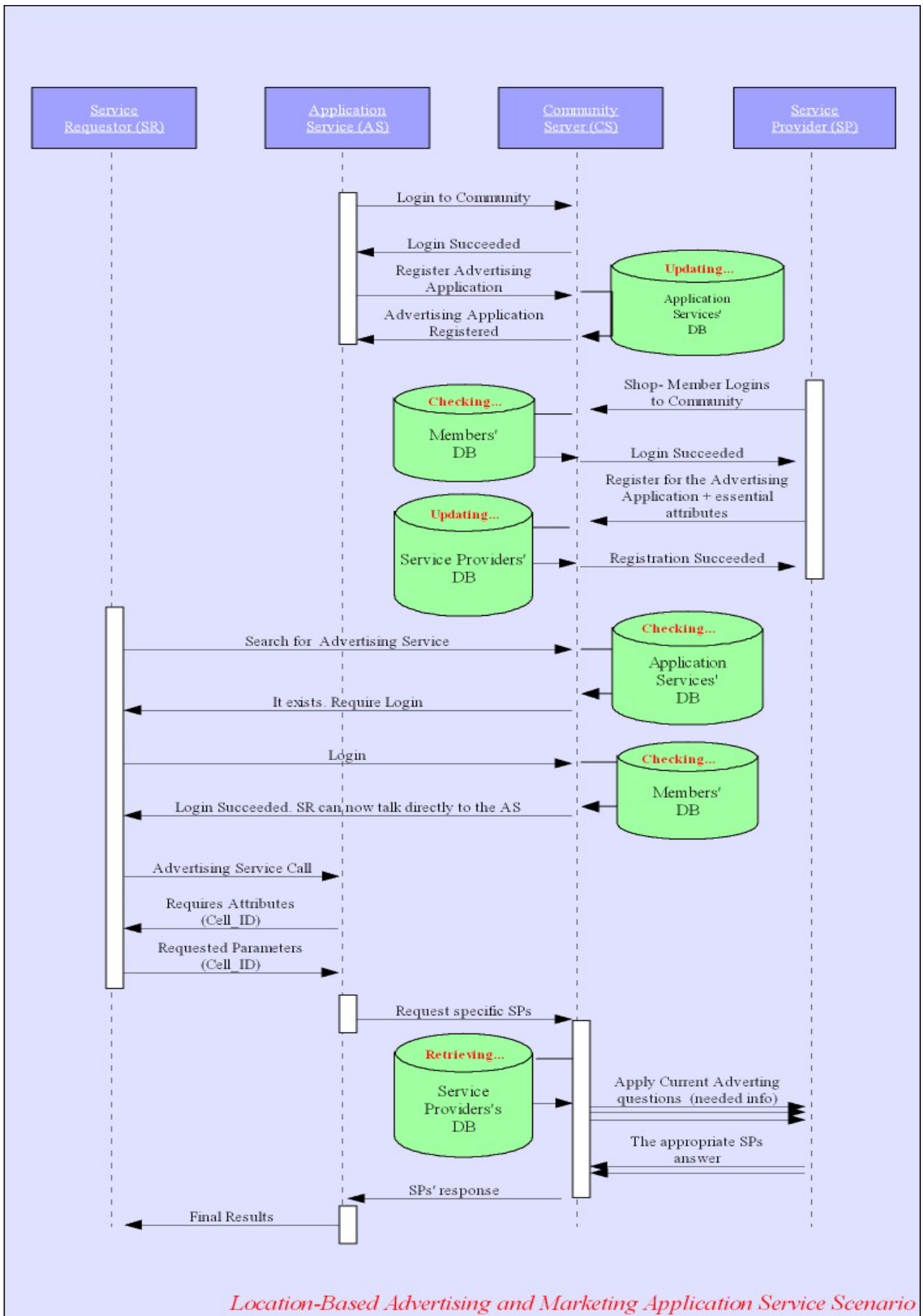
Διαγραμματικά, η διαδικασία που ακολουθείται για την προτεινόμενη υπηρεσία δημοσκοπήσεων, φαίνεται στο παρακάτω σχήμα.



Σχήμα 15. Διάγραμμα διαδικασίας δημοσκόπησης

Η Υπηρεσία πραγματοποίησης σφυγμομετρήσεων στηρίζεται στην πραγματοποίηση δημοσκοπήσεων με συμμετέχοντες τα ενδιαφερόμενα μέλη της κοινότητας. Η σωστή δημοσκόπηση στηρίζεται στα ακόλουθα κριτήρια: 1) το δείγμα πρέπει να είναι τυχαίο, 2) το δείγμα πρέπει να είναι όσο το δυνατόν μεγαλύτερο, 3) θα πρέπει να εξασφαλίζεται ανωνυμία για τους συμμετέχοντες. Όλα τα ανωτέρω εξασφαλίζονται στα πλαίσια την Ανώνυμης Κοινότητας Ανταλλαγής Πληροφοριών μεταξύ Κινητών Τερματικών σε Πραγματικό Χρόνο. Αρχικά θα πρέπει η υπηρεσία να δηλωθεί στον CS, ο οποίος και την συμπεριλαμβάνει στις *τρέχουσες* υπηρεσίες της. Τα μέλη της κοινότητας που επιθυμούν να λάβουν μέρος στις δημοσκοπήσεις, έχουν τη δυνατότητα αυτή κατεβάζοντας τον απαραίτητο client. Πλέον είναι SPs της Υπηρεσίας πραγματοποίησης δημοσκοπήσεων και μέσω του client που έχουν εγκαταστήσει στα κινητά τερματικά τους, ενημερώνουν τον CS για την διαθεσιμότητά τους αυτή. Το θέμα της διαθεσιμότητας των συμμετεχόντων σε κάθε δημοσκόπηση είναι καίριο, και εξασφαλίζεται αυτόματα διαμέσου της επικοινωνίας client – CS. Η ίδια η διαδικασία της δημοσκόπησης έχει ως εξής: οργανισμός που επιθυμεί να πραγματοποιήσει σφυγμομέτρηση (π.χ. πολιτικό κόμμα, τηλεοπτικό κανάλι, εφημερίδα, κοκ) αναζητά τον CS για την ύπαρξη τέτοιας ή παρόμοιας υπηρεσίας. Ο οργανισμός αυτός αποτελεί τον SR της υπηρεσίας. Ο CS αποκρίνεται πως έχει τέτοια υπηρεσία και ζητά από τον SR να πιστοποιήσει τα στοιχεία του. Αφού γίνει αυτό, ο SR εξουσιοδοτείται να μιλήσει απευθείας με την Υπηρεσία πραγματοποίησης δημοσκοπήσεων. Η AS του ζητά τις επιθυμητές ερωτήσεις π.χ. τη φόρμα της δημοσκόπησης. Ακολούθως, η ίδια η AS επικοινωνώντας με τον CS του ζητά να επικοινωνήσει με τους *τρέχοντες* SPs της υπηρεσίας και να τους προωθήσει την απαραίτητη φόρμα συμπλήρωσης. Ο client των SPs αυτών, αναλαμβάνει να λάβει τη φόρμα και να την παρουσιάσει σε κατανοητή μορφή στους SPs. Μετά την ολοκλήρωση της συμπλήρωσης της φόρμας, ο CS συλλέγει τα αποτελέσματα από τους SPs που αποκρίθηκαν και αποστέλλει τα αποτελέσματα στην AS. Η Υπηρεσία πραγματοποίησης δημοσκοπήσεων τότε, αφού επεξεργαστεί τα αποτελέσματα, τα προωθεί στον SR.

3.7.5- Υπηρεσία Location-Based διαφημίσεων (Location-Based Advertising and Marketing)



Location-Based Advertising and Marketing Application Service Scenario
 Σχήμα 16. Διάγραμμα υπηρεσίας Location-Based Διαφημίσεων

Η συγκεκριμένη ιδέα στηρίζεται στην προώθηση διαφημιστικών μηνυμάτων σε χρήστες τερματικών συσκευών ανάλογα με τη θέση τους. Αρχικά θα πρέπει να δημιουργηθεί η AS ονόματι Location-Based Advertising από κάποιον operator ή ανεξάρτητο φορέα. Δουλειά της είναι να δίνει εντολές σχετικά με την προώθηση όλων των κατάλληλων διαφημιστικών που εταιρίες επιθυμούν να προωθήσουν στους τελικούς καταναλωτές-χρήστες ανάλογα βέβαια με τη θέση τους. Αφού η AS δημιουργηθεί, δηλώνεται στον CS. Πλέον ο τελευταίος ενημερώνει κάθε ενδιαφερόμενο για την νέα υπηρεσία που διατηρεί, καθώς την έχει συμπεριλάβει στη βάση δεδομένων των *τρέχουσων* υπηρεσιών του. SP της υπηρεσίας είναι κάθε κατάσταση, εταιρία ή οργανισμός που επιθυμεί να προωθεί location-based διαφημιστικά μηνύματα. Ο SP ενημερώνεται για την ύπαρξη της προκείμενης υπηρεσίας και αποφασίζει να λάβει μέρος σε αυτήν. Μέσω του client που κατεβάζει, μπορεί να εισάγει όλα τα απαραίτητα δεδομένα όπως το γεωγραφικό στίγμα γύρω από το οποίο επιθυμεί να εμφανίζεται σχετικό διαφημιστικό μήνυμα, καθώς και το ίδιο το διαφημιστικό μήνυμα. Η υπηρεσία εν δράση έχει ως εξής: χρήστης τερματικής συσκευής ανακαλύπτει πώς η κοινότητα διατηρεί τρέχουσα υπηρεσία που αφορά σε διαφημιστικά μηνύματα. Στην σύντομη περιγραφή της υπηρεσίας ενημερώνεται πώς οι διαφημίσεις αυτές περιλαμβάνουν και τρέχουσες εκπτώσεις, και προσφορές σε stocks που μεταβάλλονται διαρκώς (*real time*). Ο χρήστης αποφασίζει να κάνει χρήση της υπηρεσίας και εισέρχεται στην κοινότητα ως μέλος και με την ιδιότητα του SR. Μιλώντας απευθείας με την AS, επιλέγει το είδος των διαφημίσεων που επιθυμεί να λαμβάνει (π.χ. ρουχισμός) μέσω μιας κατανοητής και εύχρηστης γραφικής διεπαφής χρήστη. Αφού η AS συλλέγει τις προτιμήσεις του μαζί με τη θέση του (υποθέτουμε ότι ο SR βρίσκεται στη βόλτα του για ψώνια), επικοινωνεί με τον CS και του δίνει εντολή να επικοινωνήσει με τους κατάλληλους SPs. Ο CS αφού ενημερωθεί από την αντίστοιχη βάση δεδομένων που διατηρεί, επικοινωνεί με εκείνους τους SPs που ταυτίζονται με τις προτιμήσεις-επιλογές του SR και που βρίσκονται φυσικά σε μια λογική ακτίνα από το γεωγραφικό του στίγμα. Οι SPs αποκρίνονται (ή ήδη έχουν αποκριθεί καθώς οι ενεργοί SPs αναγράφονται στη βάση δεδομένων των SPs και δηλώνουν αυτόματα την διαθεσιμότητά-παρουσία τους μέσω του κατάλληλου ενεργοποιημένου client για την υπηρεσία) και ο CS αποστέλλει τα αποτελέσματα για επεξεργασία στην AS. Η τελευταία προωθεί τα τελικά αποτελέσματα στον SR.

3.8- ΘΕΜΑΤΑ ΣΥΜΦΟΡΗΣΗΣ ΔΙΚΤΥΟΥ

Κατά τη περιγραφή, σχεδίαση και ανάλυση της εργασίας (για λόγους απλότητας) έχουν γίνει μια σειρά από υποθέσεις-παραδοχές:

- Η καθυστέρηση που προκαλεί το δίκτυο στην επικοινωνία είναι αμελητέα.
- Το διαθέσιμο εύρος ζώνης είναι σταθερό και αμετάβλητο.
- Δεν έχουν ληφθεί υπόψη θέματα κόστους μεταφοράς δεδομένων.

Σε περιορισμένη κλίμακα χρήσης της πλατφόρμας οι επιπτώσεις από την λανθασμένη εκτίμηση χρήσης του δικτύου δεν είναι εμφανείς, αλλά όταν η χρήση αυξάνεται παρατηρούνται φαινόμενα συμφόρησης δικτύου που μπορεί να οφείλονται στους παρακάτω λόγους:

α) Μεγάλος αριθμός χρηστών που ζητούν υπηρεσία:

Οι χρήστες της πλατφόρμας είναι χρήστες κινητών τερματικών. Αυτό σημαίνει ότι κατά την χρήση της πλατφόρμας, κάθε χρήστης που αιτείται μια υπηρεσία δεσμεύει ένα κανάλι επικοινωνίας της κυψέλης στην οποία βρίσκεται. Λόγω του ότι ο αριθμός καναλιών που μπορεί να προσφέρει μια κυψέλη είναι περιορισμένος και ο σχεδιασμός της χωρητικότητας έχει γίνει χωρίς να ληφθούν υπόψη πιθανές υπηρεσίες σαν αυτή που περιγράφεται στην εργασία μας, οποιαδήποτε αίτηση για υπηρεσία, μπορεί να προκαλέσει συμφόρηση στους διαθέσιμους πόρους του δικτύου.

Σε χρονικές περιόδους όπου η χρήση του δικτύου από χρήστες κινητών τερματικών για την παροχή άλλων υπηρεσιών είναι περιορισμένη, δεν προκύπτει κάποιο πρόβλημα από την αναζήτηση και χρήση υπηρεσιών από κάποιο μέλος της κοινότητας. Αντιθέτως συμβάλλει θετικά στη αξιοποίηση του διατιθέμενου εύρους ζώνης (utilization) αφού υπό άλλες συνθήκες τα συγκεκριμένα κανάλια επικοινωνίας θα παρέμεναν αδιάθετα.

Σε περιόδους αιχμής της κίνησης στο δίκτυο όμως, μια αίτηση για υπηρεσία από κάποιο μέλος της κοινότητας μπορεί να προκαλέσει συμφόρηση (π.χ. να μην υπάρχουν πόροι για να εξυπηρετηθούν άλλες υπηρεσίες). Οπότε, πρέπει από τη πλευρά του τηλεπικοινωνιακού παρόχου να γίνει μια ιεράρχηση όσον αφορά τη προτεραιότητα που θα έχει ένα μέλος της κοινότητας κατά την αίτηση για χρήση των υπηρεσιών της πλατφόρμας σε σχέση με τους χρήστες ομιλίας ή χρήστες άλλων διαδικτυακών εφαρμογών.

β) Μεγάλος αριθμός παρόχων υπηρεσιών:

Αν και μεγαλύτερος αριθμός παρόχων υπηρεσιών αυξάνει τις πιθανότητες να ικανοποιηθεί με ακρίβεια μια αίτηση υπηρεσίας, εντούτοις μεγαλύτερος αριθμός παρόχων υπηρεσιών μπορεί να προκαλέσει φαινόμενα συμφόρησης. Όπως στη περίπτωση των μελών της κοινότητας που αιτούνταν υπηρεσίες από τη κοινότητα έτσι και οι πάροχοι υπηρεσιών κατά την επικοινωνία τους με τις υπόλοιπες οντότητες της κοινότητας χρησιμοποιούν πόρους του δικτύου που μπορεί σε περιόδους αυξημένης ζήτησης να προκαλέσουν συμφόρηση στο δίκτυο.

Έτσι για κάθε εφαρμογή-υπηρεσία που προσφέρεται στη κοινότητα θα πρέπει να υπάρχει ένα όριο όσο αφορά τους παρόχους υπηρεσιών. Το όριο αυτό δεν μπορεί να είναι το ίδιο για όλες τις εφαρμογές υπηρεσίες και μπορεί να εξαχθεί από τη φύση της υπηρεσίας. Για παράδειγμα μια υπηρεσία ταξί θα χρειαζόταν στη χειρότερη περίπτωση, αφού θα μπορούσαν να υπάρχουν πολλά ταξί που να παρέχουν την ίδια υπηρεσία, μόνο ένα πάροχο υπηρεσίας (οδηγό ταξί) διαθέσιμο για κάθε αίτηση

υπηρεσίας (κλήση) με παράμετρο ατόμων μικρότερη ή ίση με τέσσερα. Από την άλλη πλευρά, μια υπηρεσία γνωριμιών θα χρειαζόταν σίγουρα πάνω από ένα παρόχους υπηρεσιών. Ακόμα, σε εφαρμογές-υπηρεσίες όπου η τοποθεσία του παρόχου δεν παίζει ρόλο για την ακρίβεια του αποτελέσματος κάθε εφαρμογή-υπηρεσία μπορεί να ρωτά για πληροφορία παρόχους υπηρεσιών που βρίσκονται σε διαφορετικές κυψέλες κατανέμοντας έτσι την επιβάρυνση του δικτύου.

Επίσης κατά την διάρκεια που ένας πάροχος υπηρεσίας είναι διαθέσιμος, λογικά θα έπρεπε να έχει κατειλημμένο ένα δίαυλο επικοινωνίας μέσω του οποίου θα επικοινωνεί με την εφαρμογή-υπηρεσία. Αυτό θα επιβαρύνει σημαντικά το δίκτυο γιατί σε ορισμένες περιπτώσεις θα είχαμε κατειλημμένους διαύλους, τόσους όσοι είναι οι διαθέσιμοι πάροχοι υπηρεσίας, χωρίς να υπάρχουν χρήστες που να αιτούνται τις υπηρεσίες τους. Για να αποφευχθεί αυτό, ο πάροχος υπηρεσίας δηλώνει την διαθεσιμότητα του και μετά θα βρίσκεται σε idle-mode (δεν θα απασχολεί πόρους του δικτύου). Όταν χρειάζεται να παράσχει πληροφορία σε μια εφαρμογή-υπηρεσία, τότε «ξυπνά» με την λήψη μιας ειδοποίησης από την εφαρμογή-υπηρεσία (triggering, WAP-push, smart messaging) και χρησιμοποιεί, αν υπάρχει, ένα διαθέσιμο κανάλι επικοινωνίας για να αποστείλει πληροφορία. Μετά επιστρέφει πάλι σε idle-mode.

γ) Ταυτόχρονη απάντηση από μεγάλο αριθμό παρόχων υπηρεσιών

Το πρόβλημα συμφόρησης που προκύπτει από το μεγάλο αριθμό παρόχων υπηρεσιών δεν εξαλείφεται μόνο θέτοντας τους σε idle-mode. Επιβάρυνση στο δίκτυο μπορεί να προκαλέσει και η ταυτόχρονη μετάδοση πληροφορίας από τους παρόχους υπηρεσίας, μετά από μια ειδοποίηση από την εφαρμογή-υπηρεσία διαμέσου του κεντρικού διακομιστή της κοινότητας (Community Server). Εάν οι πάροχοι υπηρεσίας απαντούν ταυτόχρονα μετά από μια ειδοποίηση, τότε για συγκεκριμένη χρονική περίοδο δεσμεύονται τόσοι δίαυλοι επικοινωνίας όσοι είναι οι διαθέσιμοι πάροχοι υπηρεσίας. Έτσι, κάθε φορά που ένας μέλος της κοινότητας θέλει να χρησιμοποιήσει μια εφαρμογή-υπηρεσία με πολλούς παρόχους υπηρεσίας, θα προκαλούσε, χωρίς να το γνωρίζει, τέτοια συμφόρηση στο δίκτυο που θα καθιστούσε αδύνατη την εξυπηρέτηση χρηστών άλλων υπηρεσιών.

Λύση στο προαναφερόμενο πρόβλημα, μπορεί να δώσει η τακτική με την οποία κάθε πάροχος υπηρεσίας απαντά στην ειδοποίηση με μια μικρή τυχαία καθυστέρηση. Ακόμα, επιπλέον της τακτικής αυτής, η αποστολή ειδοποιήσεων στους παρόχους υπηρεσίας μπορεί να γίνεται σταδιακά. Ο CS ειδοποιεί ένα μικρό αριθμό παρόχων υπηρεσίας και μετά από μια μικρή καθυστέρηση ειδοποιεί άλλους τόσους παρόχους υπηρεσίας μέχρι να τους ειδοποιήσει όλους. Τότε στην καλύτερη περίπτωση όλοι οι πάροχοι υπηρεσιών, που βρίσκονται στην ίδια κυψέλη, χρησιμοποιούν ένα μόνο δίαυλο επικοινωνίας, αποδεσμεύοντας έτσι αρκετούς διαύλους επικοινωνίας για άλλες υπηρεσίες μειώνοντας κατά πολύ την συμφόρηση του δικτύου.

Για παράδειγμα θα μπορούσε κάθε πάροχος υπηρεσίας, σε ένα σύστημα πολυπλεξίας χρόνου και συχνότητας, να στέλνει πληροφορία στη

εφαρμογή-υπηρεσία σε τυχαία πολλαπλάσια της χρονικής θυρίδας του συστήματος. Δηλαδή, σε μια κυψέλη, εάν υποθέσουμε ότι η διάρκεια της χρονοσχισμής είναι 0,577ms (όπως το GSM), κάθε πάροχος υπηρεσίας χρειάζεται 57,7ms για να μεταδώσει την πληροφορία που χρειάζεται (για ευκολία στους υπολογισμούς) και έχουμε 8 χρονοσχισμές ανά συχνότητα, τότε για να απαντήσουν 64 πάροχοι υπηρεσίας οι οποίοι θα ειδοποιούνται ανά ομάδες των 8 παρόχων κάθε φορά, θα χρειαστούν 3,7s για να ολοκληρωθεί η διαδικασία και θα δεσμεύουν μόνο μια συχνότητα (ή 8 διαύλους επικοινωνίας). Η ακριβής μέγιστη συνολική χρονική καθυστέρηση με την οποία θα μπορούσε ένα σύστημα, σαν αυτό που περιγράφεται στην εργασία, να λειτουργήσει εξαρτάται φυσικά, από την ταχύτητα μετάδοσης του δικτύου καθώς και την ανεκτικότητα ενός χρήστη μιας υπηρεσίας στην καθυστέρηση λήψης αποτελέσματος. Είναι ενδεικτικό όμως το μεγάλο κέρδος σε πόρους του δικτύου που απελευθερώνεται έναντι της καθυστέρησης που εισάγεται.

Στην αντίθετη περίπτωση που έχουμε τα ίδια δεδομένα αλλά ο κάθε πάροχος υπηρεσίας απαντά μόλις λάβει την ειδοποίηση, τότε για την χρονική στιγμή που θα μετέδιδε πληροφορία ο κάθε πάροχος, το σύστημα θα ήταν πλήρως κατειλημμένο και η χρήση του από χρήστες άλλων υπηρεσιών θα ήταν αδύνατη.

3.9- ΘΕΜΑΤΑ ΑΣΦΑΛΕΙΑΣ

Όπως έχει αναφερθεί στις γενικές απαιτήσεις που καθιστούν την προτεινόμενη πλατφόρμα λειτουργική ([κεφάλαιο 3.3](#)), σημαντικό ρόλο παίζει η ασφάλεια σε κάθε διαδικασία επικοινωνίας μεταξύ δυο οντοτήτων της κοινότητας. Ο γενικός όρος ασφάλεια δύναται να αναλυθεί σε επιμέρους ιδιότητες που πρέπει να παρέχονται από το σύστημα, ώστε να αποτρέπουν κακόβουλες ενέργειες τόσο από τις ίδιες τις οντότητες όσο και από τρίτους.

Θεωρώντας ότι ο Community Server αποτελεί μια έμπιστη αρχή στη κοινότητα και δεν θα επιχειρούσε οποιαδήποτε κακόβουλη ενέργεια στις υπόλοιπες οντότητες, η αρχιτεκτονική του συστήματος ([κεφάλαιο 3.4](#)) αποτρέπει οποιανδήποτε άμεση επικοινωνία μεταξύ μελών της κοινότητας, που μπορεί να οδηγήσει σε διαρροή ζημιογόνων πληροφοριών. Ακόμα η επικοινωνία μεταξύ των οντοτήτων γίνεται με τέτοιο τρόπο (ανταλλαγή μηνυμάτων που περιέχουν μόνο γραμματοσειρές) που είναι εξαιρετικά δύσκολη η εκτέλεση εχθρικού κώδικα σε κάποια από αυτές. Παρόλα αυτά ενδείκνυται η κάθε οντότητα, μέσω των εφαρμογών που χρησιμοποιεί για να επικοινωνεί με κάποια άλλη οντότητα, να ορίζει μια πολιτική ασφαλείας (security policies, firewalls) που να επιτρέπει ελεγχόμενη προσπάθεια στους πόρους της μόνο από οντότητες που έχουν τα κατάλληλα δικαιώματα. Έτσι διασφαλίζεται η απόλυτη ασφάλεια από κακόβουλες ενέργειες προερχόμενες από οντότητες της κοινότητας.

Όσο αφορά τρίτους που μπορούν να προκαλέσουν προβλήματα στην ασφάλεια των λειτουργιών του συστήματος, αυτά μπορούν να αποφευχθούν με την υιοθέτηση διαφόρων πρακτικών. Η πρώτη πρακτική η οποία

διασφαλίζει την εμπιστευτικότητα στην ανταλλαγή μηνυμάτων μεταξύ των οντοτήτων είναι η κρυπτογράφηση των μεταδιδόμενων μηνυμάτων με τέτοιο τρόπο ώστε ένα μήνυμα που υποκλέπτεται να μην μπορεί να αποκρυπτογραφηθεί. Η συνήθης πρακτική είναι η κρυπτογράφηση με χρήση δημοσίου κλειδιού. Σύμφωνα με αυτή, ο παραλήπτης ενός μηνύματος διαθέτει ένα κλειδί σε όποιον θέλει να επικοινωνήσει μαζί του (δημόσιο κλειδί). Ο αποστολέας θα χρησιμοποιήσει το κλειδί αυτό σε ένα πασίγνωστο αλγόριθμο (π.χ. αλγόριθμος RSA) για να κρυπτογραφήσει τα δεδομένα που θέλει να αποστείλει. Ο παραλήπτης του μηνύματος είναι ο μοναδικός γνώστης του μοναδικού κλειδιού που μπορεί να χρησιμοποιηθεί για να αποκρυπτογραφηθεί το μήνυμα που έχει παραλάβει (ιδιωτικό κλειδί). Έτσι ο αποστολέας είναι σίγουρος ότι ο μόνος που μπορεί να καταλάβει το μήνυμα που έστειλε είναι ο νόμιμος παραλήπτης του.

Μια δεύτερη πρακτική, η οποία διασφαλίζει την ταυτότητα του αποστολέα του μηνύματος είναι η ψηφιακή υπογραφή. Ο αποστολέας του μηνύματος μαζί με το μήνυμα στέλνει και μια σύνοψη του μηνύματος του. Σύνοψη ενός μηνύματος είναι ένα αρκετά μικρότερο σε μέγεθος μήνυμα το οποίο προκύπτει από την εφαρμογή του αρχικού μηνύματος σε ένα αλγόριθμο κατατεμαχισμού (π.χ. MD5). Χαρακτηριστικό του αλγόριθμου κατατεμαχισμού είναι ότι ποτέ δεν προκύπτουν ίδιες συνόψεις μηνυμάτων από δυο διαφορετικά μηνύματα που εφαρμόζονται σε αυτό τον αλγόριθμο. Ενώ σύμφωνα με την πρώτη πρακτική που αναφέρθηκε, το μήνυμα είναι κρυπτογραφημένο βάσει του δημοσίου κλειδιού του παραλήπτη, η σύνοψη του μηνύματος είναι κρυπτογραφημένη βάσει του ιδιωτικού κλειδιού του αποστολέα. Όταν ο παραλήπτης του μηνύματος αποκρυπτογραφήσει το μήνυμα που έλαβε τότε αποκρυπτογραφεί και τη σύνοψη του μηνύματος βάσει του δημοσίου κλειδιού του φαινομένου ως αποστολέα. Ακολουθώντας τον ίδιο αλγόριθμο κατατεμαχισμού (MD5) ένα αντίγραφο του ληφθέντος μηνύματος και το συγκρίνει με τη κρυπτογραφημένη σύνοψη του μηνύματος. Εάν το μήνυμα που τεμάχισε ο παραλήπτης είναι το ίδιο με την σύνοψη του μηνύματος που απέστειλε ο φαινομένου ως αποστολέας τότε πιστοποιείται η ταυτότητα του αποστολέα.

Τέλος για να έχουν αποτέλεσμα οι πιο πάνω πρακτικές ο παραλήπτης πρέπει να είναι σίγουρος ότι το δημόσιο κλειδί του αποστολέα ανήκει όντως σε αυτόν καθώς και το αντίθετο. Λύση σε αυτό το πρόβλημα έρχεται να δώσει η αρχή πιστοποίησης (Certification Authority, CA). Η αρχή πιστοποίησης δημιουργεί ένα πιστοποιητικό που δεσμεύει το δημόσιο κλειδί της ταυτότητας με την ταυτότητα, το υπογράφει ψηφιακά και το διαθέτει σε κάθε ενδιαφερόμενο. Η αρχή πιστοποίησης πρέπει να είναι μια έμπιστη αρχή. Στην πλατφόρμα που προτείνει η εργασία αυτή, ρόλο αρχής πιστοποίησης μπορεί να διαδραματίσει ο Community Server.

4- ΕΠΙΛΟΓΕΣ ΥΛΟΠΟΙΗΣΗΣ

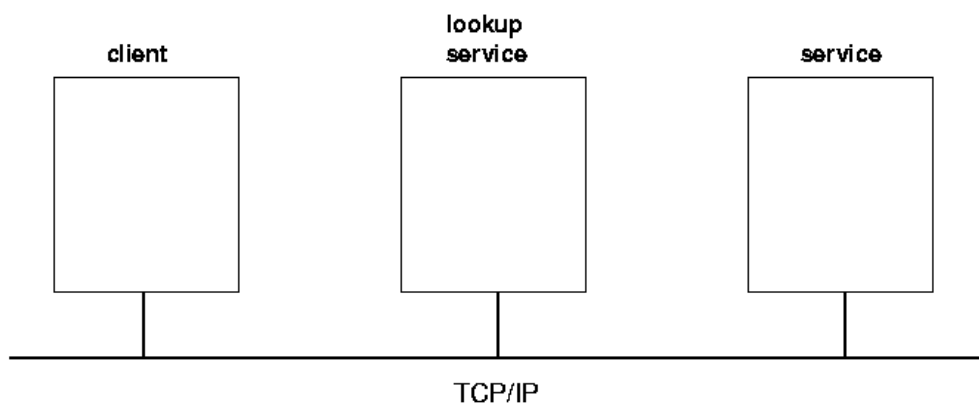
Η εργασία αυτή παραθέτει μια πρόταση για την ανταλλαγή πληροφοριών σε μια κοινότητα. Η πληροφορία που ανταλλάσσεται περιέχεται σε υπηρεσίες. Αναπτύσσει μια υπηρεσιο-στραφή αντίληψη της πληροφορίας και ακολουθεί μια υπηρεσιο-στραφή αρχιτεκτονική (Service Oriented Architecture). Αυτού του είδους η αντίληψη έχει αναπτυχθεί τα τελευταία χρόνια και ήδη έχει αρχίσει να εφαρμόζεται στην αγορά.

Έτσι έχουν αναπτυχθεί τεχνολογίες με τις οποίες μια τέτοια πρόταση όπως αυτή που περιγράφει η εργασία να μπορούν να υλοποιηθούν.

4.1- Η ΤΕΧΝΟΛΟΓΙΑ Jini

Το Jini είναι μια βιβλιοθήκη από κλάσεις Java που επιτρέπουν σε κάποιον να αναπτύξει τεχνολογίες με βάση τη υπηρεσιο-στραφή αντίληψη.

Υιοθετεί τρεις ξεχωριστές οντότητες: τον πάροχο της υπηρεσίας (Service provider), τον αιτούμενο υπηρεσία (Service requestor) και την υπηρεσία αναζήτησης (LookUp Service).



Σχήμα 17. Οι τρεις οντότητες της τεχνολογίας Jini

4.1.1- Αναζήτηση υπηρεσίας

Όταν κάποιος αναζητά μια υπηρεσία αποστέλλεται στη LookUp Service (LUS). Το LUS μπορεί να έχει μια γνωστή διεύθυνση δικτύου, οπότε ο service requestor να ξέρει πού να τη βρει ή ο service requestor να ψάξει να βρει όλες τις LUS που υπάρχουν στο δίκτυο του. Η αναζήτηση αυτή τυπικά περιορίζεται σε τοπικά δίκτυα οπότε στη περίπτωση αυτής της εργασίας ένας service requestor θα πρέπει να ξέρει τη διεύθυνση του LUS της κοινότητας του.

Αφού βρει ένα LUS, ο service requestor τότε «κατεβάζει» ένα αντικείμενο από αυτή την υπηρεσία το οποίο θα του δώσει τη δυνατότητα να αναζητήσει την υπηρεσία που θέλει. Το αντικείμενο αυτό είναι ένα

αντικείμενο Java που ανήκει στη κλάση ServiceRegistrar. Η κλάση αυτή όσον αφορά την αναζήτηση υπηρεσιών παρέχει τη μέθοδο lookup με δυο διαφορετικούς τρόπους κλήσης αυτής της μεθόδου όπως φαίνεται παρακάτω:

```
public Class ServiceRegistrar {
public java.lang.Object lookup(ServiceTemplate tmpl)
throws java.rmi.RemoteException;
public ServiceMatches lookup(ServiceTemplate tmpl,
int maxMatches)
throws java.rmi.RemoteException;
}
```

Η διαφορά των δυο κλήσεων της μεθόδου lookup είναι ότι η κλήση της μεθόδου χρησιμοποιώντας μια παράμετρο επιστρέφει ένα αντικείμενο της υπηρεσίας ενώ ο δεύτερος τρόπος επιστρέφει μια συλλογή από αντικείμενα υπηρεσιών, τόσα όσα και η τιμή της μεταβλητής maxMatches.

Η παράμετρος που χρησιμοποιείται για να κληθεί η μέθοδος lookup είναι ένα αντικείμενο της κλάσης ServiceTemplate. Η κλάση αυτή περιέχει κάποιες μεταβλητές με τις οποίες διευκρινίζεται στο LUS για ποιες υπηρεσίες γίνεται αναζήτηση.

```
package net.jini.core.lookup;

public Class ServiceTemplate {
public ServiceID serviceID;
public java.lang.Class[] serviceTypes;
public Entry[] attributeSetTemplates;

ServiceTemplate(ServiceID serviceID,
java.lang.Class[] serviceTypes,
Entry[] attrSetTemplates);
}
```

Όπως φαίνεται αναζήτηση μιας υπηρεσίας μπορεί να γίνει αν ξέρουμε την ταυτότητα της υπηρεσίας που έχει ανατεθεί μοναδικά σε κάθε υπηρεσία που παρέχεται (serviceID). Τέτοιου είδους αναζητήσεις μπορούν να επιστρέψουν το πολύ ένα αντικείμενο υπηρεσίας.

Όμως δεν μπορούμε να βασιστούμε στο γεγονός ότι ο service requestor θα γνωρίζει το serviceID της υπηρεσίας που αναζητεί. Έτσι αναζήτηση μπορεί να γίνει αναφέροντας σε ποια κλάση ανήκει το αντικείμενο υπηρεσίας που αναμένουμε να λάβουμε ώστε να μπορούμε να εκτελέσουμε την υπηρεσία που αναζητούμε.

Λόγω του ότι μια υπηρεσία μπορεί να χρησιμοποιεί πολλές κλάσεις, οπότε να χρειαστεί να λάβουμε πολλά αντικείμενα υπηρεσίας, στέλνουμε στο LUS ένα πίνακα που περιέχει όλες τις κλάσεις που θέλουμε.

Ακόμα και αν δεν γνωρίζουμε τις κλάσεις των αντικειμένων που περιμένουμε μπορούμε να κάνουμε αναζήτηση κάποιας υπηρεσίας χρησιμοποιώντας διάφορα γνωρίσματα αυτής της υπηρεσίας όπως το όνομα

της, τη διεύθυνση του service provider, το όνομα του κατασκευαστή που υλοποίησε την υπηρεσία κ.α. Αυτό γίνεται δυνατό αν καταχωρήσουμε στο πίνακα attributeSetTemplates ότι από τα πιο πάνω χαρακτηριστικά είναι γνωστό.

Με τη πιο πάνω μεθοδολογία ο service requestor λαμβάνει ένα αντικείμενο της υπηρεσίας που θέλει να χρησιμοποιήσει. Καλώντας μεθόδους του αντικειμένου αυτού το ίδιο το αντικείμενο εκτελεί μεθόδους στο service provider έτσι γίνεται χρήση της υπηρεσίας του service provider. Παρατηρούμε δηλαδή ότι το αντικείμενο της υπηρεσίας αποτελεί ένα proxy μεταξύ service requestor και provider.

Για να μπορέσει ο χρήστης να καλέσει μεθόδους αυτού του αντικειμένου χρειάζεται να ξέρει και την κλάση της οποίας είναι instance αυτό το αντικείμενο. Για να μπορέσει να γίνει αυτό ο service provider πρέπει να διαθέτει τις κλάσεις που χρειάζονται σε ένα HTTP server ούτως ώστε ο service requestor να μπορεί να τις κατεβάσει στη συσκευή του και να κάνει χρήση του proxy αντικειμένου.

4.1.2- Εγγραφή υπηρεσίας στην υπηρεσία αναζήτησης (LookUp Service)

Για να δημοσιεύσει μια υπηρεσία ο service provider ακολουθεί παρόμοια μεθοδολογία με αυτή που ακολουθά και ο service requestor για να αναζητήσει μια υπηρεσία.

Πρώτα πρέπει να βρει ένα LookUp Service. Όπως και για τον service requestor μπορεί να γίνει είτε με unicast δηλαδή γνωρίζοντας τη διεύθυνση δικτύου του LUS είτε με multicast εάν δεν γνωρίζει τη διεύθυνση δικτύου του LUS ή αν επιθυμεί να εγγράψει την υπηρεσία του σε πολλά LUS.

Αφού βρει ένα LookUp Service λαμβάνει ένα αντικείμενο το οποίο του δίνει τη δυνατότητα να εγγράψει τη υπηρεσία του στο LookUp Service. Το αντικείμενο αυτό είναι ένα αντικείμενο Java που ανήκει στη κλάση ServiceRegistrar. Η κλάση αυτή όσο αφορά την εγγραφή υπηρεσιών παρέχει τη μέθοδο register με δυο διαφορετικά ορίσματα:

```
package net.jini.core.lookup;

public Class ServiceRegistrar {
public ServiceRegistration register(ServiceItem item,
long leaseDuration)
throws java.rmi.RemoteException;
}
```

Το πρώτο όρισμα είναι ένα αντικείμενο της κλάσης ServiceItem. Όπως είχαμε αναφέρει και παραπάνω αυτό το αντικείμενο έχει τρεις μεταβλητές. Για τη πρώτη μεταβλητή τη serviceID ο πάροχος έχει δυο επιλογές: είτε να την αφήσει κενή, όποτε το LUS να της αναθέσει κάποια τιμή, είτε εάν ο πάροχος θέλει να εγγράψει την υπηρεσία του σε πολλά LUS να της δώσει κάποια τιμή. Κάνοντας το δεύτερο ο πάροχος αποσαφηνίζει το γεγονός ότι

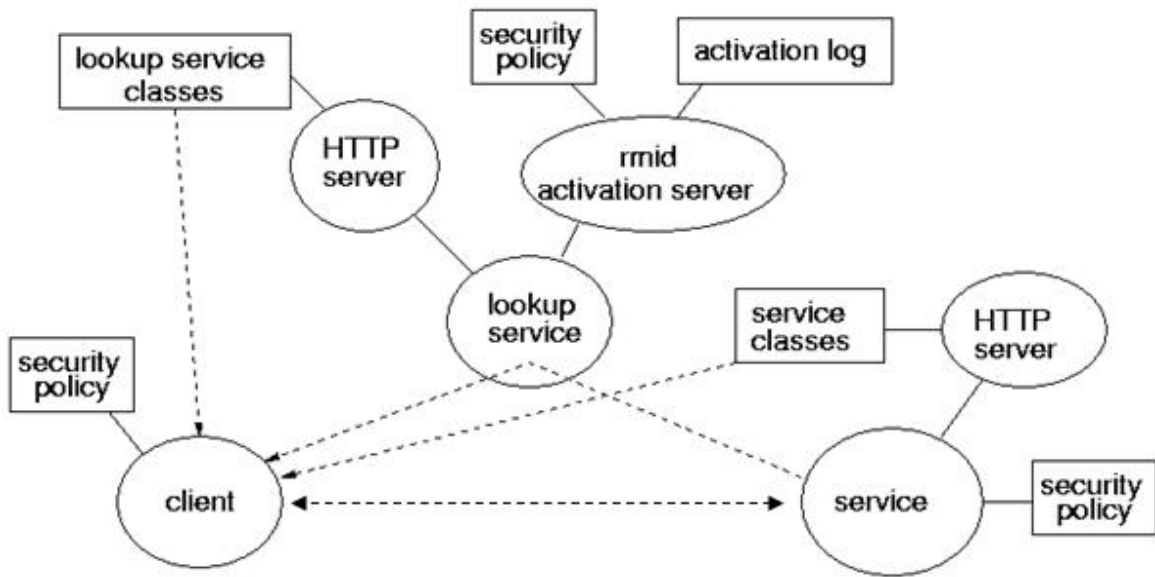
μια υπηρεσία που βρίσκεται σε διαφορετικά LUS αλλά έχει το ίδιο serviceID πρόκειται για την ίδια υπηρεσία. Η δεύτερη μεταβλητή του αντικειμένου της κλάσης ServiceItem είναι ένας πίνακας που ο πάροχος αναφέρει της κλάσεις της υπηρεσίας που θα προσφέρει. Η τρίτη μεταβλητή είναι και αυτή ένας πίνακας που περιέχει χαρακτηριστικά γνωρίσματα της υπηρεσίας όπως το όνομα της, τη διεύθυνση του service provider, το όνομα του κατασκευαστή που υλοποίησε την υπηρεσία κ.ά.

Το δεύτερο όρισμα της μεθόδου register είναι ένας αριθμός ο οποίος αναφέρει το χρόνο σε χιλιοστά του δευτερολέπτου που επιθυμεί ο πάροχος να είναι διαθέσιμη η υπηρεσία του. Μετά τη πάροδο αυτού του χρόνου η υπηρεσία διαγράφεται από το LookUp Service εκτός και αν ο ίδιος ο πάροχος κάνει ανανέωση της υπηρεσίας. Πρέπει να τονιστεί ότι ο χρόνος αυτός (leaseDuration) δεν είναι ο χρόνος που θα είναι διαθέσιμη η υπηρεσία αλλά ο χρόνος που επιθυμεί ο πάροχος της υπηρεσίας να είναι διαθέσιμη η υπηρεσία του. Το LUS αποφασίζει ποιος θα είναι ο χρόνος μετά τον οποίο αν δεν γίνει ανανέωση θα διαγραφεί η υπηρεσία.

Όταν καλέσει ο service provider τη μέθοδο register με τα ορίσματα που εξηγήσαμε πιο πάνω θα λάβει από το LUS ένα αντικείμενο της κλάσης ServiceRegistration. Η κλάση αυτή επιτρέπει στο service provider να ελέγχει της πληροφορίες της υπηρεσίας του όπως αυτή έχει εγγραφεί στο LUS. Μπορεί να αλλάξει οποιαδήποτε πληροφορία είχε δώσει αρχικά κατά την εκτέλεση της μεθόδου register. Επίσης παρέχονται μέθοδοι με τις οποίες ο service provider μπορεί ο να ενημερωθεί για το χρόνο που χρειάζεται να γίνεται ανανέωση της υπηρεσίας του ούτως ώστε να μην διαγραφεί από το LUS. Παρέχεται και μέθοδος με την οποία μπορεί να γίνει και ανανέωση της υπηρεσίας.

Εάν ο service provider δεν είχε καθορίσει serviceID για την υπηρεσία του κατά την εγγραφή της του δίνεται η δυνατότητα με την μέθοδο getServiceID να ενημερωθεί για το ποιο serviceID το έχει παραχωρήσει το LUS.

Επιγραμματικά παραθέτουμε ένα σχηματικό διάγραμμα του τρόπου με τον οποίο λειτουργεί η τεχνολογία Jini.



Σχήμα 18. Επικοινωνία οντοτήτων στη τεχνολογία Jini.

4.1.3- Διακίνηση πληροφορίας στο δίκτυο.

Όπως φαίνεται και στο σχήμα 14 η τεχνολογία Jini παρέχει τη δυνατότητα ανάπτυξης εφαρμογών που ακολουθούν την υπηρεσιο-στραφή αρχιτεκτονική (Service Oriented Architecture). Η κατακεντρωμένη αυτή αρχιτεκτονική απαιτεί την ανταλλαγή πληροφορίας μεταξύ των τριών οντοτήτων που εμφανίζονται σε μια υλοποίηση της Service Oriented Architecture.

Όταν ένας χρήστης ή ο πάροχος μιας υπηρεσίας ξεκινά μια αναζήτηση σε ένα LookUp Service, όπως αναφέρθηκε προηγουμένως, δέχεται αντικείμενα κλάσεων μέσω των οποίων εκτελεί μεθόδους για να κάνει την αναζήτηση ή την εγγραφή μιας υπηρεσίας. Για να μπορέσει να χρησιμοποιήσει αυτά τα αντικείμενα πρέπει γνωρίζει και τις κλάσεις στα οποία ανήκουν αυτά τα αντικείμενα. Έτσι ένα LookUp Service πρέπει να παρέχει αυτή την πληροφορία στους χρήστες του.

Για τον σκοπό αυτό κάθε LookUp Service έχει ένα εξυπηρετητή (web server) μέσω του οποίου μπορεί ένας πάροχος ή χρήστης μιας υπηρεσίας να προμηθεύεται τις κλάσεις που χρειάζεται. Το ίδιο ισχύει και για το πάροχο της υπηρεσίας. Πρέπει να διαθέτει τις κλάσεις που αποτελούν την υπηρεσία του σε κάθε χρήστη της υπηρεσίας που τις χρειάζεται.

Έτσι η υπηρεσία Jini χρησιμοποιεί το πρωτόκολλο HTTP για την μεταφορά των απαιτούμενων κλάσεων μεταξύ των τριών οντοτήτων που την αποτελούν. Συνήθως οι κλάσεις αυτές βρίσκονται στον εξυπηρετητή σε μορφή αρχείου JAR.

Η μεταφορά των αντικειμένων όμως λόγω του ότι είναι αντικείμενα Java ακολουθεί την τεχνολογία RMI (Remote Method Invocation). Το RMI αρχικά χρησιμοποιούσε μόνο το πρωτόκολλο JRMP (Java Remote Method Protocol). Το πρωτόκολλο αυτό χρησιμοποιεί το πρωτόκολλο TCP/IP για τη μεταφορά στο δίκτυο.

Έτσι αν μπορούμε να βλέπουμε τι διακινείται πάνω στο δίκτυο κατά τη διάρκεια μιας ανταλλαγής αντικειμένων μεταξύ δυο οντοτήτων, θα δούμε ότι γίνεται ανταλλαγή TCP πακέτων τα οποία περιέχουν σε μορφή byte τα δεδομένα.

Η τελευταία έκδοση του Jini παρουσιάζει το πρωτόκολλο JERI (Java Extensible Remote Invocation). Αυτό μας δίνει τη δυνατότητα να χρησιμοποιήσουμε άλλα πρωτόκολλα όπως τα HTTP, HTTPS για ασφαλείς συνδέσεις καθώς και άλλα πρωτόκολλα που χρησιμοποιούν πιστοποίηση και κρυπτογράφηση των δεδομένων όπως το Kerberos.

Επίσης η επικοινωνία μεταξύ του αντικειμένου (proxy) που χρησιμοποιεί ο χρήστης της υπηρεσίας και της «πραγματικής» υπηρεσίας που βρίσκεται στον υπολογιστή του παρόχου μπορεί να γίνει με οποιονδήποτε πρωτόκολλο ακολουθεί το μοντέλο client-server.

Θεωρητικά ο τρόπος επικοινωνίας μεταξύ του proxy της υπηρεσίας και της υπηρεσίας στο πάροχο είναι ασυσχέτιστος με το τρόπο επικοινωνίας του παρόχου της υπηρεσίας και του χρήστη της υπηρεσίας με το LookUp Service. Στη πραγματικότητα αν κάποια από τις δυο μορφές επικοινωνίας χρησιμοποιεί κάποιο πρωτόκολλο που απαιτεί κρυπτογράφηση ή /και πιστοποίηση ενώ η άλλη μορφή χρησιμοποιεί ανασφαλή πρωτόκολλα τότε θα υπάρξει πρόβλημα στη λειτουργία του συστήματος και θα πάρουμε Java exceptions με αποτέλεσμα να μην επιτευχθεί επιτυχής χρήση της υπηρεσίας.

4.1.4- Πολιτική Ασφαλείας στο Jini

Η ασφάλεια κατέχει σημαντικό ρόλο στην ανάπτυξη καταναμημένων συστημάτων. Θα ήταν πολύ επικίνδυνο να μπορούσε ο καθένας να προσφέρει μια υπηρεσία που να εκτελεί εντολές στη συσκευή του χρήστη της υπηρεσίας. Για παράδειγμα μια υπηρεσία θα μπορούσε να εκτελεί την εντολή format στο δίσκο του χρήστη με καταστροφικά αποτελέσματα.

Το Jini προσφέρει βασική ασφάλεια βασισμένη στο μοντέλο ασφαλείας του JDK 1.2. Αυτό χρησιμοποιεί ένα αντικείμενο της κλάσης [SecurityManager](#) το οποίο επιτρέπει ή απαγορεύει τη προσπέλαση από συγκεκριμένες πηγές.

Επειδή χρησιμοποιείται RMI για την επικοινωνία των οντοτήτων όσο αφορά την εγγραφή και αναζήτηση υπηρεσιών πρέπει να εγκαταστήσουμε ένα [SecurityManager](#) και συγκεκριμένα ένα [RMISecurityManager](#) . Αυτό χρησιμοποιεί ένα αρχείο πολιτικής ασφαλείας (policy file) ούτως ώστε να γνωρίζει σε ποιους επιτρέπεται η προσπέλαση του συστήματος. Αυτό είναι απαραίτητο και για τις τρεις οντότητες μιας SOA.

Ένα αρχείο πολιτικής ασφαλείας (policy file) φορτώνεται κατά την εκκίνηση μιας εφαρμογής που το χρειάζεται. Δηλώνεται σαν παράμετρος της JVM (Java Virtual Machine) με τον ακόλουθο τρόπο.

```
java -Djava.security.policy="policy.all" ...
```

όπου το `policy.all` είναι το αρχείο πολιτικής ασφαλείας

Αυτό το αρχείο μπορεί να περιέχει κάποιες δηλώσεις ασφαλείας όπως τι παρακάτω:

```
grant {
  permission java.net.SocketPermission "224.0.1.85", "connect,accept";
  permission java.net.SocketPermission "*.edu.au:80", "connect";
}
```

Με τις πιο πάνω δηλώσεις επιτρέπεται η προσπέλαση για συγκεκριμένες διαδικασίες όπως η προσπέλαση σε sockets από συγκεκριμένες διευθύνσεις, σε όλες ή σε συγκεκριμένες θύρες όπως π.χ. 80. Θα μπορούσαμε να επιτρέψουμε παντός είδους προσπέλαση από συγκεκριμένες διευθύνσεις με τη παρακάτω δήλωση

```
grant codebase "http://sunshade.dstc.edu.au/classes/" {
  permission java.security.AllPermission "", "";
}
```

Ακόμα μπορούμε να ζητήσουμε ψηφιακή υπογραφή συνημμένη με τον κώδικα με τη παρακάτω δήλωση

```
grant signedBy "sysadmin" {
  permission java.security.AllPermission "", "";
}
```

4.1.5- Πλεονεκτήματα της τεχνολογίας Jini

- Παρέχει δυναμική καταχώρηση των υπηρεσιών λαμβάνοντας έτσι υπόψιν τυχόν αδυναμίες του δικτύου. Με αυτό το τρόπο όταν ο πάροχος της υπηρεσίας δεν είναι διαθέσιμος, τότε ούτε και η υπηρεσία του είναι διαθέσιμη για αναζήτηση σε τυχόν χρήστες της.
- Ο χρήστης της υπηρεσίας χειρίζεται την υπηρεσία σα να εκτελεί μεθόδους σε αντικείμενα που βρίσκονται στη συσκευή του. Δεν τον ενδιαφέρει πως και με πια πρωτόκολλα γίνεται η επικοινωνία του με το πάροχο της υπηρεσίας.
- Επειδή είναι υλοποιημένη σε Java οποιονδήποτε πρόβλημα παρουσιαστεί στο δίκτυο θα προκαλέσει ένα Java exception το οποίο ο χρήστης της υπηρεσίας μπορεί να το χειριστεί και έτσι να αποτρέψει γενικότερα προβλήματα που μπορούν να επηρεάσουν είτε το πάροχο της υπηρεσίας είτε άλλους χρήστες της υπηρεσίας.

- Η τεχνολογία αυτή παρέχει δυνατότητες πιστοποίησης του κάθε χρήστη με γνωστά πρωτόκολλα ασφαλείας καθώς και δυνατότητες κωδικοποίησης των μηνυμάτων που κυκλοφορούν πάνω στο δίκτυο.

4.1.6- Μειονεκτήματα της τεχνολογίας Jini

- Είναι εξολοκλήρου υλοποιημένη σε Java οπότε υπηρεσίες που έχουν υλοποιηθεί σε άλλες γλώσσες προγραμματισμού δεν μπορούν να εγγραφούν. Επίσης το ίδιο ισχύει και για τους service requestors. Πρέπει να γράφουν τις εφαρμογές τους σε Java.
- Έχουμε υπερφόρτωση του δικτύου με τη συνεχή διακίνηση αντικειμένων.
- Η τεχνολογία αυτή χρειάζεται ένα περίπλοκο περιβάλλον για να λειτουργήσει. Ο κάθε service provider και το κάθε LUS πρέπει να διαθέτουν και ένα HTTP server στον οποίο να είναι διαθέσιμες οι κλάσεις τους.
- Είναι ευάλωτη σε εχθρικό κώδικα που μπορεί κάποιος να τον παρέχει υπό τη μορφή μια χρήσιμης υπηρεσίας. Έτσι και οι τρεις οντότητες που λαμβάνουν μέρος σε αυτή τη τεχνολογία (service requestor, service provider και LookUp Service) θα πρέπει να χρησιμοποιούν πολιτικές ασφαλείας, πράγμα που κάνει την υλοποίηση πιο πολύπλοκη.

4.1.7- Εφαρμογή της τεχνολογίας Jini στη προτεινόμενη πλατφόρμα

Λόγω της δυνατότητας της τεχνολογίας Jini να παρέχει δυναμική καταχώρηση υπηρεσιών, είναι ιδανική για να υλοποιήσει το κομμάτι που αφορά την διαθεσιμότητα των παρόχων υπηρεσιών για κάθε εφαρμογή υπηρεσία.

Έτσι η βάση δεδομένων που βρίσκεται στο CS και είναι υπεύθυνη για τη καταχώρηση των διαθέσιμων SPs είναι στη πραγματικότητα ένα LookUp Service. Ο κάθε πάροχος υπηρεσίας δηλώνει την διαθεσιμότητα του στη κοινότητα εγγράφοντας μια Jini υπηρεσία στο LookUp Service όπως θα έκανε ένας Service Provider στο Jini.

Η υπηρεσία αυτή δεν χρειάζεται να έχει κάποια λειτουργικότητα και θα μπορούσε να είναι ακόμα και κενή. Σημασία έχει στο Serviceltem να εμφανίζεται το όνομα της εφαρμογής-υπηρεσίας με την οποία συνεργάζεται ο SP καθώς και ένα αναγνωριστικό του SP(στο GSM σύστημα χρησιμοποιούμε το MSISDN). Επίσης καθοριστικό ρόλο παίζει και η διάρκεια για την οποία πρέπει η υπηρεσία να διαθέσιμη στη κοινότητα. Μετά τη λήξη του χρόνου αυτού ο SP πρέπει να ανανεώσει τη διαθεσιμότητα του, αλλιώς θεωρείται μη διαθέσιμος. Έτσι η επιλογή του

leaseDuration πρέπει να επιλεγθεί ώστε να είναι τόσο μεγάλη ώστε να μην επιβαρύνεται το δίκτυο από συνεχείς ανανεώσεις και τόσο μικρή ώστε να μειωθεί η πιθανότητα ένας πάροχος υπηρεσίας που δεν είναι πλέον διαθέσιμος, να φαίνεται στο LookUp Service ως διαθέσιμος. Αφού ο SP έχει επιτυχώς δηλώσει τη διαθεσιμότητά του τότε μπορεί να αποσυνδεθεί από το δίκτυο και να περάσει σε idle-mode.

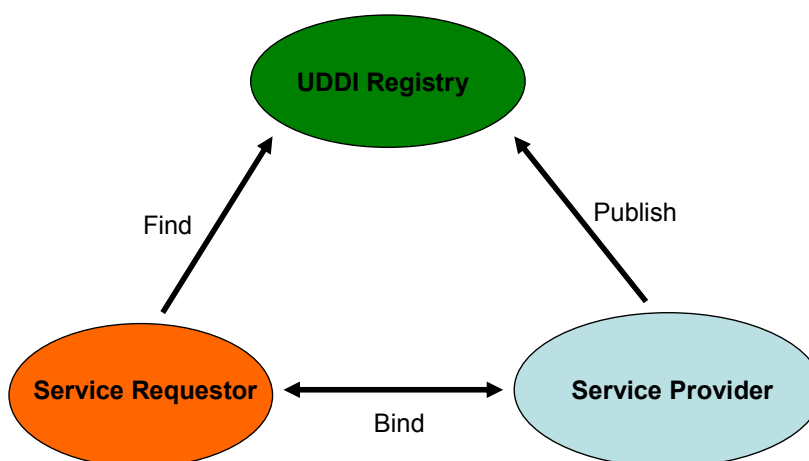
Όταν μια εφαρμογή-υπηρεσία χρειάζεται πληροφορία από κάποιους SPs τότε επικοινωνεί με το CS ενημερώνοντας τον τι είδους υπηρεσίες αναζητεί. Ο CS τότε λειτουργεί σαν Service Requestor στο LookUp Service και αναζητά υπηρεσίες Jini που πληρούν τα κριτήρια αναζήτησης τα οποία έθεσε η εφαρμογή-υπηρεσία. Αφού ο CS βρει τους ανάλογους της υπηρεσίας –εφαρμογής παρόχους υπηρεσίας τότε αντί να εκτελέσει οποιαδήποτε μέθοδο στον κάθε ένα, όπως θα περιμέναμε να γίνει, απλώς διαβάζει από το ServiceItem του κάθε πάροχου υπηρεσίας το MSISDN του. Έτσι τώρα ο CS είναι σε θέση, αφού γνωρίζει τα MSISDN των διαθέσιμων πάροχων υπηρεσίας, να επικοινωνήσει και να ζητήσει από το Network Module να ειδοποιηθούν οι πάροχοι υπηρεσίας που έχει επιλέξει.

4.2- ΤΟ ΠΡΩΤΟΚΟΛΛΟ UDDI

Το UDDI (Universal Description Discovery and Integration protocol) παρέχει μια τεχνική προδιαγραφή για την περιγραφή, ανακάλυψη και ολοκλήρωση υπηρεσιών που παρέχονται μέσω του διαδικτύου. Δημιουργήθηκε σαν φυσικό επακόλουθο από την ανάπτυξη των Web Services (που χρησιμοποιούν το πρωτόκολλο SOAP) καθώς και του πρωτοκόλλου WSDL (Web Services Description Language) χωρίς όμως να περιορίζεται μόνο σε αυτού του είδους τις υπηρεσίες.

Το UDDI μπορεί να χρησιμοποιηθεί για να περιγράψει απλές δικτυακές υπηρεσίες όπως μια ιστοσελίδα ή μια διεύθυνση ηλεκτρονικού ταχυδρομείου μέχρι και υπηρεσίες που χρησιμοποιούν SOAP, CORBA και Java RMI.

Υποστηρίζει σύμφωνα με τη SOA τρεις ξεχωριστές οντότητες: τον πάροχο της υπηρεσίας (Service Provider), τον αιτούμενο υπηρεσία (Service Requestor) και την υπηρεσία περιγραφής και αναζήτησης υπηρεσιών (UDDI Registry).



Σχήμα 19. Διάγραμμα εγγραφής, αναζήτησης και χρήσης υπηρεσιών

Όπως φαίνεται στο σχήμα 15 ο SP δηλώνει μια περιγραφή της υπηρεσίας που διαθέτει στο UDDI Registry. Η περιγραφή αυτή περιέχει όλα τα απαραίτητα στοιχεία που θα χρειαστεί ο SR για να χρησιμοποιήσει την υπηρεσία αυτή. Ο SR με τη σειρά του θα αναζητήσει μια υπηρεσία μέσω του UDDI Registry ώστε να λάβει όλες τις πληροφορίες που χρειάζεται σχετικά με τις υπηρεσίες που τον ενδιαφέρουν και ακολούθως να επιλέξει και χρησιμοποιήσει την υπηρεσία που τον ενδιαφέρει.

Η πληροφορία που μπορεί να συσχετίζεται με μια υπηρεσία σύμφωνα με τις προδιαγραφές του UDDI χωρίζεται σε τρεις κατηγορίες ανάλογες του τηλεφωνικού καταλόγου:

White Pages

Όπως και σε κάθε τηλεφωνικό κατάλογο οι «λευκές σελίδες» παρέχουν ένα ονομαστικό ευρετήριο έτσι και εδώ οι «λευκές σελίδες» παρέχουν ένα ονομαστικό ευρετήριο των επιχειρήσεων (Service Providers) οι οποίες θέλουν ή έχουν ήδη καταχωρήσει υπηρεσίες. Περιέχουν πληροφορίες όπως το όνομα της επιχείρησης, μια περιγραφή της επιχείρησης, διεύθυνση και τηλέφωνα επικοινωνίας. Επίσης μπορεί να περιέχουν και αναγνωριστικά των επιχειρήσεων.

Yellow Pages

Αυτές οι «σελίδες» κατηγοριοποιούν τις επιχειρήσεις ή τις υπηρεσίες ανάλογα με την κατηγορία στην οποία ανήκουν. Για παράδειγμα μπορεί να περιέχονται δεδομένα όπως βιομηχανίες, προϊόντα ή γεωγραφικούς κώδικες, όλα ομαδοποιημένα σύμφωνα με καθορισμένες κατηγορίες.

Green Pages

Αυτές οι σελίδες περιλαμβάνουν την τεχνική περιγραφή μιας υπηρεσίας. Αυτό συνήθως περιλαμβάνει έναν δείκτη σε μια διεύθυνση όπου υπάρχουν τεχνικές προδιαγραφές σχετικά με την υπηρεσία και μια διεύθυνση για να καλέσει κάποιος την υπηρεσία.

4.2.1- Αρχιτεκτονική του UDDI

Το UDDI χωρίζεται σε τρία μέρη :

1. Το UDDI Data Model: Είναι μια τεχνική περιγραφή, ένα XML Schema, η οποία καθορίζει το τρόπο με τον οποίο περιγράφονται οι υπηρεσίες και οι επιχειρήσεις.

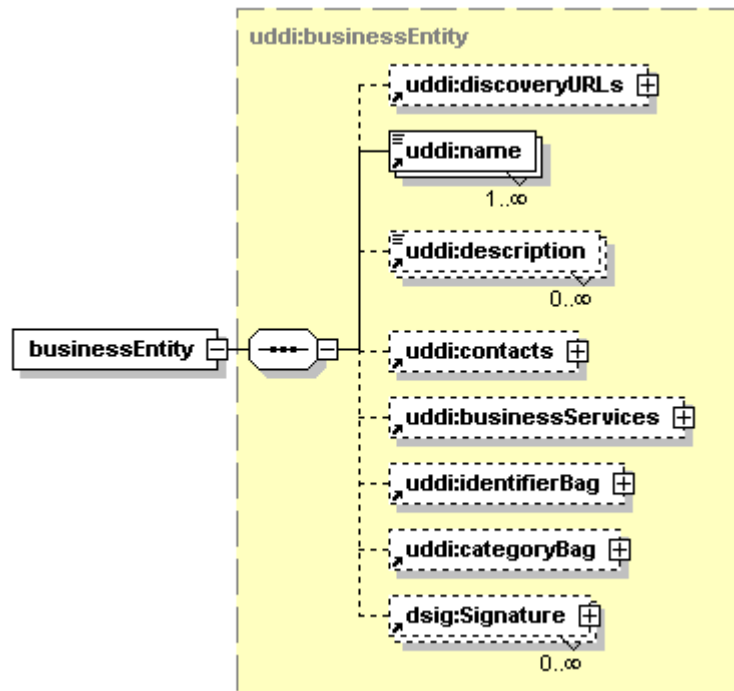
2. Το UDDI API: Αποτελεί ένα API που βασίζεται στο SOAP με το οποίο μπορούμε να αναζητήσουμε και να καταχωρήσουμε υπηρεσίες.

3. Το UDDI cloud services: Είναι το σύνολο όλων των ιστοθέσεων που υλοποιούν το πρωτόκολλο UDDI και γίνεται συγχρονισμός όλων των πληροφοριών που υπάρχουν σε κάθε κόμβο UDDI.

4.2.2- UDDI Data Model

Το UDDI Data Model αποτελείται από τέσσερις βασικούς τύπους δόμησης των δεδομένων:

α) *businessEntity*: Περιέχει πληροφορία σχετική με την επιχείρηση. Κάθε *businessEntity* ξεχωρίζει μοναδικά από το *businessKey*.



Σχήμα 20. Η δομή του *bussinessEntity*

Όπως φαίνεται στο σχήμα 16 το *bussinessEntity* αποτελείται από πολλά elements. Από αυτά μόνο το *name* είναι υποχρεωτικό να υπάρχει. Το *name* περιέχει το όνομα της επιχείρησης και μέσα σε ένα *bussinessEntity* μπορούν να υπάρχουν πολλά διαφορετικά στοιχεία ονομάτων (*name elements*) που να αντιστοιχούν στην ίδια επιχείρηση ή οργανισμό.

Το στοιχείο *discoveryURLs* περιέχει διάφορες διευθύνσεις ιστοσελίδων της επιχείρησης.

Το στοιχείο *description* περιέχει μια περιγραφή της επιχείρησης ή του οργανισμού.

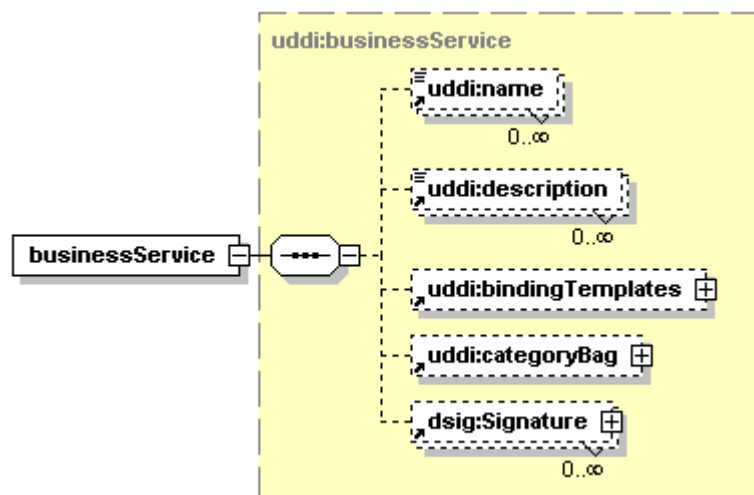
Το στοιχείο *contacts* περιέχει πληροφορίες σχετικά με στοιχεία ατόμων επικοινωνίας της επιχείρησης όπως διευθύνσεις, τηλέφωνα, τύπος εργασίας κτλ.

Το στοιχείο *identifierBag* περιέχει πληροφορία με την οποία *bussinessEntity* δομές μπορούν να βρεθούν κατά την αναζήτηση με βάση διεθνώς τυποποιημένα αναγνωριστικά επιχειρήσεων.

Το στοιχείο *categoryBag* περιέχει πληροφορία με την οποία *bussinessEntity* δομές μπορούν να βρεθούν κατά την αναζήτηση με βάση διεθνώς τυποποιημένα αναγνωριστικά κατηγορίας επιχειρήσεων.

Το στοιχείο *Signature* επιτρέπει στην επιχείρηση που διαφημίζει σε ένα UDDI Registry ένα οποιοδήποτε από τους βασικούς τύπους να τους υπογράψει ψηφιακά ώστε να διασφαλίζεται η αυθεντικότητά τους.

β) *bussinessService*: Περιέχει πληροφορία σχετική με την παρεχόμενη υπηρεσία. Κάθε υπηρεσία ξεχωρίζει μοναδικά με το *ServiceKey*.



Σχήμα 21. Η δομή του *businessService*

Όπως φαίνεται στο σχήμα 17 το *businessService* αποτελείται από μη υποχρεωτικά στοιχεία.

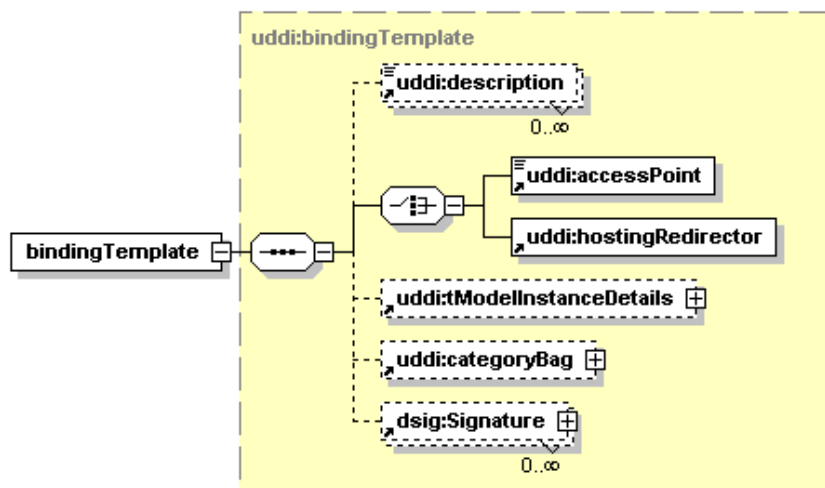
Το στοιχείο *name* περιέχει το όνομα της παρεχόμενης υπηρεσίας και μέσα σε ένα *businessService* μπορούν να υπάρχουν πολλά διαφορετικά στοιχεία ονομάτων (*name elements*) που να αντιστοιχούν στην ίδια υπηρεσία.

Το στοιχείο *description* περιέχει μια σύντομη περιγραφή της υπηρεσίας.

Το στοιχείο *categoryBag* περιέχει πληροφορία με την οποία *businessService* δομές μπορούν να βρεθούν κατά την αναζήτηση με βάση διεθνώς τυποποιημένα αναγνωριστικά κατηγορίας επιχειρήσεων στις οποίες απευθύνεται η υπηρεσία.

Το στοιχείο *Signature* επιτρέπει στην επιχείρηση που διαφημίζει σε ένα UDDI Registry ένα οποιοδήποτε από τους βασικούς τύπους να τους υπογράψει ψηφιακά ώστε να διασφαλίζεται η αυθεντικότητά τους.

γ) *bindingTemplate*: Περιέχει πληροφορίες σχετικά με το που και πως μπορούμε να έχουμε πρόσβαση σε μια υπηρεσία. Κάθε *bindingTemplate* αναγνωρίζεται μοναδικά με το *bindingKey*.



Σχήμα 22. Η δομή του *bindingTemplate*

Όπως φαίνεται στο σχήμα 18 το στοιχείο *accessPoint* είναι υποχρεωτικό. Το στοιχείο αυτό περιέχει τη διεύθυνση στην οποία μπορεί να βρεθεί μια υπηρεσία, ή να βρεθεί η διεύθυνση μιας υπηρεσίας. Το στοιχείο *hostingRedirector* υπάρχει μόνο για λόγους συμβατότητας με τη δεύτερη έκδοση του πρωτοκόλλου και δεν πρέπει να υπάρχει ένα έχουμε στοιχείο *accessPoint*.

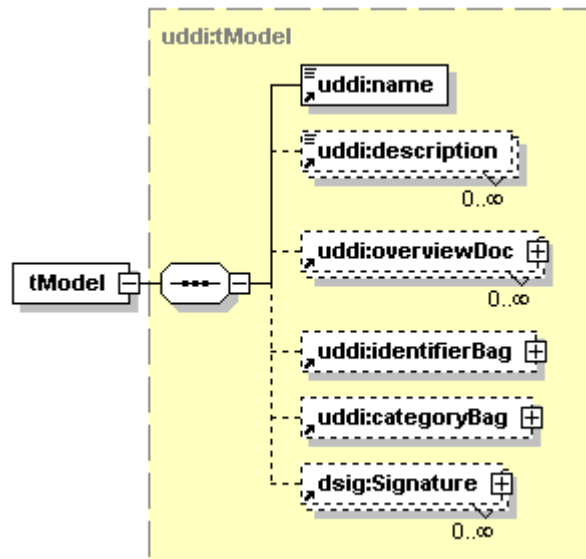
Το στοιχείο *description* περιέχει μια σύντομη περιγραφή, σε μορφή κειμένου, του *bindingTemplate*.

Το στοιχείο *tModelInstanceDetails* περιέχει μια συλλογή από *tModelKey* τα οποία είναι μοναδικά αναγνωριστικά για διάφορες δομές *tModel*. Το σύνολο όλων των *tModelKey* που χρησιμοποιεί μια υπηρεσία ονομάζεται και «τεχνικό αποτύπωμα» της υπηρεσίας.

Το στοιχείο *categoryBag* περιέχει πληροφορία σχετικά με την κατηγορία στην οποία *bindingTemplate* δομές ανήκουν. Για παράδειγμα μια *bindingTemplate* δομή μπορεί να είναι δοκιμαστική (*test*) ή έτοιμη για χρήση (*production*).

Το στοιχείο *Signature* επιτρέπει στην επιχείρηση που διαφημίζει σε ένα UDDI Registry ένα οποιοδήποτε από τους βασικούς τύπους να τους υπογράψει ψηφιακά ώστε να διασφαλίζεται η αυθεντικότητά τους.

δ) *tModel*: Περιέχει πληροφορίες σχετικά με το που μπορούμε να έχουμε πρόσβαση σε τυχόν εξωτερικές τεχνικές προδιαγραφές της υπηρεσίας. Κάθε *tModel* αναγνωρίζεται μοναδικά με το *tModelKey*



Σχήμα 23. Η δομή του tModel

Όπως φαίνεται στο σχήμα 19 στο *tModel*, το στοιχείο *name* είναι υποχρεωτικό να υπάρχει. Το στοιχείο *name* πρέπει να εκφράζεται σαν ένα URI.

Το στοιχείο *description* περιέχει μια σύντομη περιγραφή, σε μορφή κειμένου, του *tModel*.

Το στοιχείο *overviewDoc*, περιέχει το URL στο οποίο υπάρχει η εξωτερική τεχνική περιγραφή καθώς και μια περιγραφή για το πως πρέπει να χρησιμοποιηθεί κατά τη χρήση της υπηρεσίας.

Το στοιχείο *identifierBag* περιέχει αναγνωριστικά του *tModel* σε διάφορα συστήματα ταξινόμησης των *tModel*.

Το στοιχείο *categoryBag* περιέχει πληροφορίες σχετικά με πιθανές κατηγορίες στις οποίες ανήκει το *tModel*.

Το στοιχείο *Signature* επιτρέπει στην επιχείρηση που διαφημίζει σε ένα UDDI Registry ένα οποιοδήποτε από τους βασικούς τύπους να τους υπογράψει ψηφιακά ώστε να διασφαλίζεται η αυθεντικότητά τους.

Εκτός από τις βασικές δομές δεδομένων που αναφέραμε υπάρχουν ακόμα δυο δομές δεδομένων στο μοντέλο πληροφορίας του UDDI. Το *publisherAssertion* το οποίο παρουσιάζει πιθανές σχέσεις μεταξύ διαφόρων *businessEntity* όπως για παράδειγμα μητρική ή θυγατρική εταιρία (parent-child) και το *operationalInfo* το οποίο περιέχει πληροφορία σχετικά με την δημιουργία και τροποποίηση κάθε βασικής δομής δεδομένων όπως πότε, από ποιον και σε πιο κόμβο UDDI εκδόθηκε ένας βασικός τύπος.

4.2.3- Αναζήτηση Πληροφορίας

Αναζήτηση πληροφορίας σχετικά με υπηρεσίες σε μια UDDI Registry είναι δυνατή μέσω αποστολής ενός μηνύματος SOAP από τον SR στο UDDI

Registry. Τότε το UDDI Registry απαντά με ένα μήνυμα SOAP το οποίο περιέχει την πληροφορία που ζήτησε ο SR.

Αναλόγως της πληροφορίας που αναζητά, ο SR μέσα στο μήνυμα SOAP μπορεί να χρησιμοποιήσει τα ακόλουθα στοιχεία :

find_binding: χρησιμοποιείται για να αναζητήσουμε πληροφορία που βρίσκεται σε δομή τύπου *bindingTemplate*.

find_business: χρησιμοποιείται για να αναζητήσουμε πληροφορία που βρίσκεται σε δομή τύπου *businessEntity*.

find_relatedBusinesses: χρησιμοποιείται για να αναζητήσουμε πληροφορία που βρίσκεται σε δομή τύπου *publisherAssertion*.

find_service: χρησιμοποιείται για να αναζητήσουμε πληροφορία που βρίσκεται σε δομή τύπου *businessService*.

find_tModel: χρησιμοποιείται για να αναζητήσουμε πληροφορία που βρίσκεται σε δομή τύπου *tModel*.

get_bindingDetail: χρησιμοποιείται για να αναζητήσουμε με βάση ένα *bindingKey* πληροφορία που βρίσκεται σε δομή τύπου *bindingTemplate*.

get_businessDetail: χρησιμοποιείται για να αναζητήσουμε με βάση ένα *businessKey* πληροφορία που βρίσκεται σε δομή τύπου *businessEntity*.

get_operationalInfo: χρησιμοποιείται για να αναζητήσουμε με βάση ένα *entityKey* δηλαδή ένα εκ των *bindingKey*, *businessKey*, *serviceKey*, *tModelKey* πληροφορία που βρίσκεται σε δομή ανάλογου με το *entityKey* τύπου.

get_serviceDetail: χρησιμοποιείται για να αναζητήσουμε με βάση ένα *serviceKey* πληροφορία που βρίσκεται σε δομή τύπου *businessService*.

get_tModelDetail: χρησιμοποιείται για να αναζητήσουμε με βάση ένα *tModelKey* πληροφορία που βρίσκεται σε δομή τύπου *tModel*.

4.2.4- Εγγραφή/Τροποποίηση Πληροφορίας

Εγγραφή ή τροποποίηση πληροφορίας σχετικά με υπηρεσίες σε μια UDDI Registry είναι δυνατή μέσω αποστολής ενός μηνύματος SOAP από τον SP στο UDDI Registry. Τότε το UDDI Registry απαντά με ένα κενό μήνυμα SOAP στη περίπτωση επιτυχούς εγγραφής.

Αναλόγως της πληροφορίας που θέλει να εγγράψει, ο SP στο μήνυμα SOAP μπορεί να χρησιμοποιήσει τα ακόλουθα στοιχεία:

add_publisherAssertions: χρησιμοποιείται για να προσθέσουμε ένα *publisherAssertion* μέσα σε ένα ήδη υπάρχον σύνολο συσχετίσεων μιας *businessEntity*.

delete_binding: χρησιμοποιείται για να διαγράψουμε με βάση το *bindingKey* ένα ή περισσότερα *bindingTemplate*.

delete_Business: χρησιμοποιείται για να διαγράψουμε με βάση το *businessKey* ένα ή περισσότερα *businessEntity* από το UDDI Registry.

delete_publisherAssertions: χρησιμοποιείται για να διαγράψουμε ένα ή περισσότερα *publisherAssertion* από ένα ήδη υπάρχον σύνολο συσχετίσεων μιας *businessEntity*.

delete_service: χρησιμοποιείται για να διαγράψουμε ένα ή περισσότερα *businessService* από το UDDI Registry και από μια ήδη υπάρχον *businessEntity*.

delete_tModel: χρησιμοποιείται για να διαγράψουμε ένα ή περισσότερα *tModel*. Δεν γίνεται διαγραφή της πληροφορίας απλώς τα διαγραφέντα *tModel* δεν μπορούν να αναζητηθούν με τη μέθοδο *find_tModel*.

get_assertionStatusReport: χρησιμοποιείται για να βρούμε την κατάσταση τυχόν συσχετίσεων του *businessEntity* ενός SP. Επιτυχής χρήση επιστρέφει λίστα με τις καταστάσεις των ζητούμενων σχέσεων.

get_publisherAssertions: χρησιμοποιείται για να βρούμε τυχόν συσχετίσεις του *businessEntity* ενός SP. Επιτυχής χρήση επιστρέφει λίστα με τις ζητούμενες δομές *publisherAssertion*.

get_registeredInfo: χρησιμοποιείται για να βρούμε όλα τα *businessEntity* και *tModel* που ελέγχονται από ένα SP. Επιτυχής χρήση επιστρέφει λίστα με τις ζητούμενες δομές.

save_binding: χρησιμοποιείται για εισαγωγή ή για τροποποίηση ενός ή περισσότερων *bindingTemplate*. Το νέο *bindingTemplate* πρέπει να υπάρχει μέσα στο *save_binding*. Επιτυχής χρήση επιστρέφει λίστα με τα καινούργια *bindingTemplate*.

save_service: χρησιμοποιείται για εισαγωγή ή για τροποποίηση ενός ή περισσότερων *businessService*. Το νέο *businessService* πρέπει να υπάρχει μέσα στο *save_service*. Επιτυχής χρήση επιστρέφει λίστα με τα καινούργια *businessService*.

save_tModel: χρησιμοποιείται για εισαγωγή ή για τροποποίηση ενός ή περισσότερων *tModel*. Το νέο *tModel* πρέπει να υπάρχει μέσα στο *save_tModel*. Επιτυχής χρήση επιστρέφει λίστα με τα καινούργια *tModel*.

set_publisherAssertions: χρησιμοποιείται για την διαχείριση όλων των *publisherAssertion* που αφορούν ένα SP.

4.2.5- Διακίνηση πληροφορίας στο δίκτυο

Η αναζήτηση και η εγγραφή πληροφορίας σε μια UDDI Registry όπως έχουμε δει υλοποιείται μέσω μηνυμάτων SOAP. Ένα μήνυμα SOAP είναι ένα κείμενο γραμμένο σε XML. Η προδιαγραφή του SOAP δεν καθορίζει τίποτα άλλο παρά έναν απλό φάκελο βασισμένο στην XML για να μεταφέρει τις διάφορες πληροφορίες μαζί με ένα σύνολο κανόνων για μετάφραση και μετατροπή των διαφόρων τύπων δεδομένων σε XML εκφράσεις. Ο σχεδιασμός του SOAP το κάνει κατάλληλο για μια μεγάλη ποικιλία εφαρμογών που χρησιμοποιούν μηνύματα για επικοινωνία και για πολλά ολοκληρωμένα μοντέλα. Το SOAP χρησιμοποιεί το πρωτόκολλο HTTP για την μεταφορά του. Έτσι σε επίπεδο μεταφοράς το πρωτόκολλο που χρησιμοποιείται είναι το πρωτόκολλο μεταφοράς του HTTP δηλαδή το TCP.

Μεταφορά πληροφορίας στο δίκτυο όμως δεν παρατηρείται μόνο κατά την αναζήτηση ή εγγραφή πληροφοριών σε μια UDDI Registry αλλά και κατά την λειτουργία μιας υπηρεσίας.

Αν και η συνηθέστερη πρακτική των υπηρεσιών είναι η επικοινωνία να γίνεται με SOAP υπάρχουν και άλλοι τρόποι. Το UDDI υποστηρίζει όλων των ειδών τις υπηρεσίες έτσι κατά την χρήση μιας υπηρεσίας μπορούν να χρησιμοποιηθούν πρωτόκολλα όπως SMTP για υπηρεσίες που ενεργοποιούνται μέσω αποστολής e-mail, FTP για υπηρεσίες ανταλλαγής αρχείων καθώς και πρωτόκολλα ασφαλείας όπως HTTPS. Επίσης άλλες υπηρεσίες μπορεί να χρησιμοποιούν Java RMI για επικοινωνία οπότε χρησιμοποιούνται τα πρωτόκολλα JRMP και JERI όπως γίνεται στην τεχνολογία Jini.

4.2.6- Πολιτική Ασφαλείας στο UDDI

Το UDDI παρέχει πληροφορίες σχετικά με υπηρεσίες. Οι πληροφορίες αυτές δεν πρέπει να είναι προσβάσιμες από κάποιους οι οποίοι είναι αναρμόδιοι (μη εξουσιοδοτημένοι) ή έχουν κακόβουλες προθέσεις. Έτσι κατά την αναζήτηση, την εγγραφή και τροποποίηση πληροφοριών σε μια UDDI Registry απαιτείται η ταυτοποίηση του χρήστη πριν γίνει οτιδήποτε.

Ταυτοποίηση ενός χρήστη γίνεται με αποστολή σε ένα κόμβο UDDI, που είναι εγγεγραμμένος χρήστης, ενός μηνύματος SOAP. Το μήνυμα αυτό πρέπει να περιέχει την εντολή *get_authToken*.

Με αυτό το μήνυμα ένας χρήστης ζητά από ένα κόμβο να του παραχωρήσει πρόσβαση στο UDDI Registry. Μέσα στο μήνυμα σαν attributes του *get_authToken* απαιτούνται δυο attributes:

α) το *UserID* το οποίο έχει σαν τιμή το μοναδικό αναγνωριστικό που έχει αναθέσει ο κόμβος σε ένα συγκεκριμένο χρήστη.

β) το *cred* το οποίο έχει σαν τιμή τον κωδικό που αντιστοιχεί στο κάθε χρήστη.

Εάν ο χρήστης είναι εγγεγραμμένος τότε θα λάβει σαν απάντηση στο μήνυμα που έστειλε, ένα μήνυμα SOAP που θα περιέχει ένα κλειδί πρόσβασης (*authToken*). Το κλειδί αυτό πρέπει να το ενθυλακώνει ο χρήστης σε κάθε μήνυμα που στέλνει προς τη UDDI Registry ώστε να του επιστραφεί ένα μήνυμα με την πληροφορία που ζήτησε.

Από την πλευρά ενός πάροχου υπηρεσιών που θέλει να διαφημίσει μια υπηρεσία, η διαδικασία είναι η ίδια. Η μόνη διαφορά με ένα χρήστη που αναζητά υπηρεσίες είναι ότι ο πάροχος υπηρεσιών έχει πρόσβαση μόνο στα δικά του δεδομένα. Σε καμία περίπτωση δεν μπορεί να προσθέσει, να διαγράψει ή να τροποποιήσει υπηρεσίες στις οποίες δεν είναι ο εκδότης τους. Ακόμα και όταν κάποιος νόμιμα δηλώσει για το *bussinessEntity* το οποίο έχει εκδώσει, μια σχέση (*publisherAssertion*) με κάποιο *bussinessEntity* εταιρίας που δεν είναι ο ίδιος εκδότης, τότε η σχέση δεν είναι νόμιμη. Η σχέση μπορεί να γίνει νόμιμη και να παρουσιάζεται κατά την αναζήτηση, μόνο όταν ο εκδότης του δεύτερου *bussinessEntity* χρησιμοποιήσει τα αναγνωριστικά του *UserID* και *cred*, λάβει ένα *authToken* και επιβεβαιώσει με ένα μήνυμα SOAP που περιέχει την εντολή *set_publisherAssertions*, ότι η σχέση μεταξύ των δυο εταιριών ή οργανισμών ισχύει.

Αφού ένας χρήστης που αναζητά υπηρεσίες ή ένας πάροχος υπηρεσιών που διαφημίζει υπηρεσίες μέσω ενός UDDI Registry τελειώσει με την επικοινωνία του με το UDDI Registry πρέπει να ζητήσει να καταστραφεί το κλειδί πρόσβασης που του έχει δοθεί. Αυτό μπορεί να το κάνει στέλνοντας ένα μήνυμα που περιέχει την εντολή *discard_authToken* και ενθυλακώνοντας το *authToken* που του έχει δοθεί. Απάντηση κενού μηνύματος σημαίνει επιτυχής ακύρωση του *authToken*. Εάν ο χρήστης αμελήσει να ακυρώσει ένα *authToken*, ο κόμβος UDDI θα ακυρώσει αυτόματα το συγκεκριμένο κλειδί πρόσβασης μετά την πάροδο ορισμένου χρόνου. Ο χρόνος με την πάροδο του οποίου ακυρώνεται ένα κλειδί, είναι μεταβλητός από κόμβο σε κόμβο και εξαρτάται από την πολιτική ασφαλείας του κάθε κόμβου.

Για να αποτραπεί η κλοπή ενός *authToken* κατά την διακίνηση του στο δίκτυο, αποστολές μηνυμάτων που περιέχουν κλειδιά πρόσβασης ή και των κωδικών του χρήστη γίνονται με το πρωτόκολλο HTTPS το οποίο κρυπτογραφεί τα SOAP μηνύματα κατά την διακίνηση τους στο δίκτυο.

Όσο αφορά την αυθεντικότητα των πληροφοριών των οποίων εγγράφονται σε μια UDDI Registry, αυτές μπορούν να πιστοποιηθούν με την χρήση ψηφιακής υπογραφής. Ο εκδότης της πληροφορίας υπογράφει ψηφιακά τα δεδομένα που καταχωρεί στη UDDI Registry έτσι ώστε κάποιος που χρησιμοποιεί τα δεδομένα αυτά, να μπορεί να εξακριβώσει ότι προέρχονται από την εταιρία ή τον οργανισμό που φαίνεται μέσα από το *bussinessEntity* ότι προέρχονται. Η χρήση της ψηφιακής υπογραφής των πληροφοριών ακολουθεί την τεχνική προδιαγραφή XML-Signature Syntax and Processing.

4.2.7- Πλεονεκτήματα του πρωτοκόλλου UDDI

- Παρέχει πληροφορίες σχετικά με οποιοδήποτε τύπο υπηρεσιών, από απλές ιστοσελίδες μέχρι Web Services.
- Είναι ανεξάρτητο από γλώσσες προγραμματισμού.
- Έχει καθιερωθεί σαν προδιαγραφή, έτσι χρησιμοποιείται από ένα ευρύ φάσμα παρόχων υπηρεσιών.
- Είναι απλό και ασφαλές στη χρήση.
- Χρησιμοποιεί τα ευρέως διαδεδομένα πρωτόκολλα SOAP, XML και WSDL τα υποστηρίζονται από όλες τις εταιρίες ανάπτυξης λογισμικού.

4.2.8- Μειονεκτήματα του πρωτοκόλλου UDDI

- Δεν παρέχει υπηρεσίες αλλά πληροφορίες για τις υπηρεσίες.
- Είναι δύσκολο να αυτοματοποιηθεί η αναζήτηση και χρήση των υπηρεσιών που περιγράφονται.
- Δεν παρέχει με κάποιο μηχανισμό έλεγχο αν υπάρχουν οι υπηρεσίες που διαφημίζονται.

4.2.9- Εφαρμογή του πρωτοκόλλου UDDI στη προτεινόμενη πλατφόρμα

Το πρωτόκολλο UDDI λόγω του ότι δεν παρέχει τη δυνατότητα στο χρήστη να χρησιμοποιεί κατευθείαν της υπηρεσίες που διαφημίζονται αλλά παρέχει πληροφορίες για αυτές καθώς και λόγω του ότι χρησιμοποιεί πρωτόκολλα που είναι ευρέως διαδεδομένα στην ανάπτυξη καταναμημένων εφαρμογών όπως το SOAP, XML και WSDL, είναι δυνατόν να χρησιμοποιηθεί για να υλοποιήσει τη διαδικασία καταχώρισης, αναζήτησης και χρήσης των εφαρμογών-υπηρεσιών.

Χρησιμοποιώντας ένα UDDI Registry υλοποιείται η βάση δεδομένων που καταχωρεί τις εφαρμογές-υπηρεσίες και βρίσκεται στο CS. Έτσι η κάθε εφαρμογή-υπηρεσία μπορεί να καταχωρηθεί στη βάση δεδομένων της κοινότητας χρησιμοποιώντας μηνύματα SOAP που ακολουθούν τις προδιαγραφές που ορίζει το API του πρωτοκόλλου UDDI (όπως περιγράφηκε στην [κεφάλαιο 4.2.4](#)).

Οι πληροφορίες που καταχωρεί κάθε εφαρμογή- υπηρεσία, σε ένα UDDI Registry, είναι μεταξύ άλλων η διεύθυνση δικτύου που βρίσκεται η υπηρεσία καθώς και οι τεχνικές προδιαγραφές που είναι αναγκαίες για χρήση μιας υπηρεσίας-εφαρμογής (WSDL). Οι πληροφορίες αυτές

παρέχουν στο CS (που σε αυτή τη περίπτωση λειτουργεί σαν χρήστης που αναζητά υπηρεσίες) τη δυνατότητα μέσω του UDDI API αναζήτησης υπηρεσιών ([κεφάλαιο 4.2.3](#)) να τις παρουσιάζει σε ένα SR μέσω μιας εύχρηστης διεπαφής (π.χ. web interface). Έτσι ο χρήστης μιας υπηρεσίας (SR) μπορεί εύκολα και απλά να αναζητά να βρίσκει και να χρησιμοποιεί αυτές τις υπηρεσίες.

Φυσικά αφού ο SR αναζητεί υπηρεσίες μέσω μιας εφαρμογής που βρίσκεται στο τερματικό του (π.χ. browser), η κάθε εφαρμογή-υπηρεσία, για να διατηρήσει την απλότητα στη όλη διαδικασία, πρέπει να προσφέρει τη χρήση της υπηρεσίας της με τον ίδιο τρόπο με τον οποίο έγινε και η αναζήτηση.

4.3 ΤΟ ΠΡΩΤΟΚΟΛΛΟ HTTPS

Το HTTPS είναι το πρωτόκολλο που χρησιμοποιείται για ασφαλείς συνδέσεις σε ένα εξυπηρετητή διαδικτύου (web server) με το οποίο είναι δυνατή η κρυπτογράφηση και η ταυτοποίηση στην επικοινωνία. Χρησιμοποιώντας το πρωτόκολλο HTTPS αντί του HTTP έχει ως αποτέλεσμα ένα μήνυμα να προωθείται στη ασφαλή θύρα επικοινωνίας 443 αντί στη γνωστή θύρα 80 που χρησιμοποιούν συμβατικά όλοι οι εξυπηρετητές διαδικτύου. Η σύνοδος τότε ακολουθεί μεθόδους που καθορίζει ένα πρωτόκολλο ασφαλείας. Το HTTPS κρυπτογραφεί τα δεδομένα της συνόδου χρησιμοποιώντας το πρωτόκολλο SSL (Secure Socket Layer) εξασφαλίζοντας έτσι την απαιτούμενη ασφάλεια από επιθέσεις διαφόρων ειδών.

4.3.1 Το πρωτόκολλο SSL

Το SSL είναι ένα πρωτόκολλο το οποίο δίνει τη δυνατότητα να κρυπτογράφησης και αποκρυπτογράφησης δεδομένων κατά τη διάρκεια μιας συνόδου. Το πρωτόκολλο αυτό βρίσκεται μεταξύ επιπέδου εφαρμογής (HTTP) και μεταφοράς(TCP/IP) όσον αφορά τη στοίβα πρωτοκόλλων. Για να λειτουργήσει μια ασφαλής σύνοδος χρησιμοποιώντας το SSL χρειάζονται τα παρακάτω στοιχεία:

- Ο πελάτης της συνόδου πρέπει να έχει τις δυνατότητες να χρησιμοποιήσει το πρωτόκολλο.
- Ο πελάτης και ο εξυπηρετητής της συνόδου πρέπει να έχουν ένα πιστοποιητικό το οποίο να συνδέει τα δημόσια κλειδιά τους με τη ταυτότητα τους.
- Τα πιστοποιητικά πρέπει να φέρουν την ψηφιακή υπογραφή μιας αρχής πιστοποίησης, έμπιστης και από τις δύο πλευρές.

Το SSL είναι σήμερα το επικρατέστερο πρωτόκολλο που χρησιμοποιείται για ασφαλή επικοινωνία. Δυνατότητες χρησιμοποίησης του πρωτοκόλλου αυτού βρίσκονται προεγκατεστημένες σε όλα τα προγράμματα πλοήγησης στο διαδίκτυο.

4.3.2 Εφαρμογή του πρωτοκόλλου HTTPS στην προτεινόμενη πλατφόρμα

Το πρωτόκολλο HTTPS είναι δυνατόν να υλοποιήσει τη διαδικασία που είναι αναγκαία για την ασφάλεια της προτεινόμενης πλατφόρμας (κεφάλαιο 3.8) όσο αφορά κακόβουλες ενέργειες από εξωγενής της κοινότητας οντότητες.

Καταρχήν ο Community Server θα πρέπει να λειτουργήσει σαν αρχή πιστοποίησης και διανομής κλειδιών κρυπτογράφησης. Για μεγαλύτερη εμπιστοσύνη, ο Community Server πρέπει να έχει ένα πιστοποιητικό από κάποια από τις παγκοσμίου εμπιστοσύνης αρχές πιστοποίησης. Ένα μέλος της κοινότητας όταν καταχωρεί το username και το password του κρυπτογραφεί τα δεδομένα αυτά βάσει του δημοσίου κλειδιού του Community Server. Το δημόσιο αυτό κλειδί βρίσκεται στο πιστοποιητικό του Community Server.

Όταν ένα μέλος της κοινότητας πιστοποιηθεί σαν έγκυρο μέλος θα του παραχωρείται ένα πιστοποιητικό ψηφιακά υπογεγραμμένο από το Community Server. Τέτοια πιστοποιητικά πρέπει να εκδοθούν και σε κάθε εφαρμογή-υπηρεσία η οποία είναι εγγεγραμμένη στη κοινότητα. Έτσι κατά την επικοινωνία ενός μέλους μιας κοινότητας με μια εφαρμογή υπηρεσία εκτός από τη κρυπτογράφηση των δεδομένων της συνόδου θα παρέχεται και η αναγκαία ταυτοποίηση των δύο επικοινωνούντων οντοτήτων.

4.4- ΥΠΗΡΕΣΙΑ ΣΥΝΤΟΜΩΝ ΜΗΝΥΜΑΤΩΝ (SMS)

Η υπηρεσία σύντομων μηνυμάτων είναι ένας μηχανισμός παράδοσης σύντομων μηνυμάτων σε δίκτυα κινητών τηλεπικοινωνιών. Το μήνυμα μπορεί να περιέχει μόνο γραπτό κείμενο το οποίο δεν υπερβαίνει τους 160 λατινικούς χαρακτήρες. Η αποστολή μηνύματος ακολουθεί τη μέθοδο της αποθήκευσης και προώθησης προς και από τα κινητά τερματικά. Ένα μήνυμα αφού σταλεί, αποθηκεύεται σε ένα κέντρο σύντομο μηνυμάτων (Short Message Center, SMC) και κατόπιν διαβιβάζεται στο κινητό τερματικό του παραλήπτη. Έτσι σε περίπτωση που η συσκευή του παραλήπτη είναι εκτός δικτύου, το μήνυμα φυλάσσεται και αποστέλλεται αργότερα.

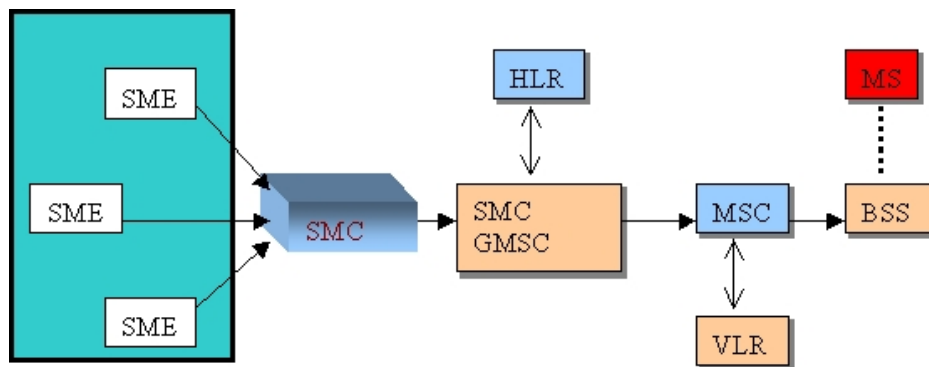
Επιπλέον, ένα ενδιαφέρον γνώρισμα της υπηρεσίας είναι ότι εάν το επιθυμεί ο αποστολέας, μπορεί να λαμβάνει μια ειδοποίηση μετά από την αποστολή ενός μηνύματος, που να επιβεβαιώνει την παράδοση του μηνύματος στον προοριζόμενο παραλήπτη. Η υπηρεσία SMS κατά τη λειτουργία της δεν χρησιμοποιεί τα κανάλια επικοινωνίας του δικτύου που χρησιμοποιούνται για αποστολή φωνής και δεδομένων άλλα το κανάλι σηματοδότησης του δικτύου. Με αυτό το τρόπο τα μηνύματα SMS μπορούν να σταλούν ή να παραληφθούν κατά τη διάρκεια που ο αποστολέας ή ο παραλήπτης συμμετέχουν σε μια σύνοδο ομιλίας ή δεδομένων.

Η υπηρεσία SMS υποστηρίζεται από τις εθνικές και διεθνείς υπηρεσίες περιαγωγής κάνοντας εφικτή την αποστολή και παραλαβή

μηνυμάτων SMS προς και από όλους τους χρήστες του συστήματος κινητών τηλεπικοινωνιών GSM σε οποιονδήποτε σημείο του πλανήτη.

4.4.1- Πως λειτουργεί η υπηρεσία SMS

Στο σχήμα 20 φαίνεται μια τυπική οργάνωση των τηλεπικοινωνιακών στοιχείων που συνθέτουν ένα δίκτυο GSM το οποίο υποστηρίζει την υπηρεσία SMS.



Σχήμα 24. Η δομή του GSM που υποστηρίζει SMS

Το SMC (Short Message Center) είναι η οντότητα που κάνει την εργασία της αποθήκευσης και προώθησης των μηνυμάτων προς και από το κινητό τερματικό. Ο αποστολέας ή παραλήπτης του μηνύματος (Short Message Entity, SME) μπορεί να βρίσκεται είτε σε σταθερό δίκτυο (π.χ. internet εάν του παρέχει αυτή τη δυνατότητα ο πάροχος του δικτύου) ή σε δίκτυο κινητής τηλεφωνίας χρησιμοποιώντας το κινητό τερματικό του.

Το SMC GMSC είναι μια πύλη MSC (gateway MSC) που μπορεί να λαμβάνει μηνύματα SMS. Η πύλη MSC είναι το στοιχείο ενός δικτύου κινητής τηλεφωνίας, το οποίο αναλαμβάνει την επαφή με άλλα δίκτυα. Κατά τη λήψη ενός μηνύματος SMS από το Short Message Center, το GMSC χρησιμοποιεί το SS7 δίκτυο για να ρωτήσει την τρέχουσα θέση του κινητού τερματικού του παραλήπτη από το HLR (Home Location Register).

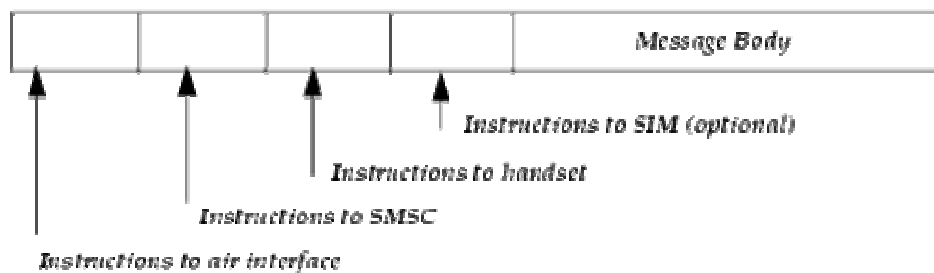
Το HLR είναι η κύρια βάση δεδομένων σε ένα δίκτυο κινητών τηλεπικοινωνιών. Φυλάσσει τις συνδρομητικές πληροφορίες του κινητού τερματικού καθώς επίσης και τις πληροφορίες δρομολόγησης του συνδρομητή, δηλ. τη περιοχή (που καλύπτεται από ένα MSC) που βρίσκεται τη δεδομένη στιγμή. Το GMSC είναι έτσι ικανό να στείλει το μήνυμα στο σωστό MSC.

Το MSC (Mobile Switching Center) είναι η οντότητα σε ένα δίκτυο GSM που αναλαμβάνει τις συνδέσεις μεταξύ των κινητών τερματικών ή μεταξύ των κινητών τερματικών και του δικτύου σταθερής τηλεφωνίας.

Ένα VLR (Visitor Location Register) αντιστοιχεί σε κάθε MSC και περιέχει προσωρινές πληροφορίες για κάθε κινητό τερματικό που βρίσκεται στη περιοχή κάλυψης του MSC. Το VLR περιέχει πληροφορία που προσδιορίζει μοναδικά ένα κινητό και τη κυψέλη (ή μια ομάδα κυψελών)

όπου το κινητό τερματικό βρίσκεται. Χρησιμοποιώντας τις πληροφορίες αυτές από το VLR, το MSC είναι σε θέση να αποστείλει το μήνυμα SMS στο αντίστοιχο BSS (Base Station System, BSC + BTSs), το οποίο διαβιβάζει το μήνυμα SMS στο κινητό τερματικό του παραλήπτη. Το BSS αποτελείται από τους σταθμούς βάσης, οι οποίοι στέλνουν και λαμβάνουν δεδομένα προς και από τα κινητά τερματικά. Για την αποστολή των δεδομένων του μηνύματος SMS, χρησιμοποιείται το κανάλι σηματοδότησης της κυψέλης, έτσι το κινητό τερματικό μπορεί να λάβει τα μηνύματα ακόμα και κατά τη διάρκεια μιας κλήσης ομιλίας ή δεδομένων.

Η γενική δομή ενός μηνύματος SMS μπορεί να φανεί στο πιο κάτω σχήμα [13].



Σχήμα 25. Η δομή μηνύματος SMS

4.4.2- Επικοινωνία εφαρμογών με μηνύματα SMS

Εκτός από την επικοινωνία μεταξύ χρηστών κινητών τερματικών, τα τελευταία χρόνια έχει αναπτυχθεί και η επικοινωνία εφαρμογών που λειτουργούν πάνω στη κινητή συσκευή, με εφαρμογές που βρίσκονται κάπου απομακρυσμένα (συνήθως εφαρμογές του τηλεπικοινωνιακού παρόχου) χρησιμοποιώντας μηνύματα SMS. Έτσι έχουν αναπτυχθεί και πολλές υπηρεσίες που έχουν σαν μέσο επικοινωνίας μηνύματα SMS (Nokia Smart Messaging, WAP push).

Η κεφαλίδα ενός μηνύματος SMS αποτελείται από μια σειρά από bits που σκοπό έχουν τη ρύθμιση διαφόρων παραμέτρων τόσο των στοιχείων του δικτύου (SMC, SMS GMSC, MSC) όσο και στις ίδιες τις κινητές συσκευές. Ένα από αυτά τα bits καθορίζει εάν το κείμενο στο μήνυμα του χρήστη περιέχει μόνο το κείμενο που έχει εισάγει ο αποστολέας ή περιέχεται και κάποια κεφαλίδα (User Data Header, UDH) πριν από το κείμενο. Το bit αυτό ονομάζεται User-Data-Header-Indicator (UDHI) και εάν η τιμή του είναι 0 τότε το μήνυμα περιέχει μόνο το μήνυμα που έστειλε ο αποστολέας ενώ αν είναι 1 τότε το μήνυμα που έστειλε ο αποστολέας ακολουθεί μετά από μια κεφαλίδα. Η κεφαλίδα αυτή χρησιμοποιείται συνήθως όταν ο αποστολέας θέλει να στείλει ένα μήνυμα που ξεπερνά τους 160 χαρακτήρες, οπότε το μήνυμα πρέπει να χωριστεί σε μικρότερα μηνύματα και ανασυντεθεί στη κινητή συσκευή του παραλήπτη.

Μια κεφαλίδα UDH μπορεί να περιέχει ορισμένα προκαθορισμένα στοιχεία. Κάθε στοιχείο αποτελείται από τρία υποχρεωτικά πεδία:

- Information-Element-Identifier το οποίο καθορίζει τι τύπου είναι το στοιχείο
- Length of Information Element το οποίο καθορίζει το μέγεθος του στοιχείου
- Information Element Data το οποίο περιέχει την πληροφορία του στοιχείου της κεφαλίδας

Το Information-Element-Identifier μπορεί να πάρει διάφορες τιμές με τη σημασία που φαίνεται στο παρακάτω σχήμα .

VALUE (hex)	MEANING
00	Concatenated short messages, 8-bit reference number
01	Special SMS Message Indication
02	Reserved
03	Value not used to avoid misinterpretation as <LF> character
04	Application port addressing scheme, 8 bit address
05	Application port addressing scheme, 16 bit address
06	SMSC Control Parameters
07	UDH Source Indicator
08	Concatenated short message, 16-bit reference number
09	Wireless Control Message Protocol
0A-6F	Reserved for future use
70-7F	SIM Toolkit Security Headers
80 - 9F	SME to SME specific use
A0 - BF	Reserved for future use
C0 - DF	SC specific use
E0 - FF	Reserved for future use

Σχήμα 26. Η τιμές που μπορεί να πάρει το Information-Element-Identifier

Για να λάβει μια εφαρμογή που λειτουργεί σε ένα κινητό τερματικό ένα μήνυμα SMS πρέπει, όπως αναφέραμε, να υπάρχει στο μήνυμα μια κεφαλίδα UDH η οποία να περιέχει ένα στοιχείο του τύπου Application port addressing scheme. Δηλαδή το Information-Element-Identifier του στοιχείου θα έχει τη δεκαεξαδικής μορφή τιμή 04 ή 05. Το Information Element Data πρέπει να περιέχει την θύρα στην οποία ακούει η εφαρμογή που βρίσκεται εγκατεστημένη στο κινητό τερματικό του παραλήπτη. Τότε το περιεχόμενο του μηνύματος που έστειλε ο αποστολέας διαβάζεται από την αρμόδια εφαρμογή και όχι από τον χρήστη της συσκευής.

4.4.3- Πλεονεκτήματα της Υπηρεσίας SMS

- ο Είναι μια απλή και βολική μέθοδος για ανταλλαγή μικρής σε μέγεθος πληροφορίας μεταξύ χρηστών κινητών συσκευών.

- Ο μηχανισμός αποστολής και λήψης μηνυμάτων εκτός του ότι εξοικονομεί χρόνο στο χρήστη της υπηρεσίας είναι η φθηνότερη από άποψη χρέωσης μέθοδος επικοινωνίας.
- Σε πολλές περιπτώσεις ενδείκνυται η αποστολή μηνύματος παρά η φωνητική συνομιλία.
- Μπορεί να χρησιμοποιηθεί και σαν ένα αρκετά ελκυστικό μέσο διαφήμισης
- Προσφέρει ένα μηχανισμό πάνω στον οποίο μπορούν να αναπτυχθούν πρωτότυπες υπηρεσίες.
- Επειδή χρησιμοποιεί το κανάλι σηματοδότησης της κυψέλης δεν επιβαρύνει τη συμφόρηση του δικτύου.

4.4.4- Μειονεκτήματα της Υπηρεσίας SMS

- Τα μηνύματα μπορούν να περιέχουν μόνο απλό κείμενο. Δεν είναι δυνατή η αποστολή εικόνας και ήχου.
- Είναι περιορισμένο σε μέγεθος (δεν μπορεί να ξεπεράσει τους 160 χαρακτήρες) το μήνυμα.
- Ο ρυθμός μετάδοσης ενός μηνύματος SMS στο δίκτυο είναι πολύ αργός.
- Ο μηχανισμός εισαγωγής χαρακτήρων στα κινητά περιορίζει τη ευκολία στη χρήση.

4.4.5- Εφαρμογή της υπηρεσίας SMS στη προτεινόμενη πλατφόρμα

Η υπηρεσία SMS λόγω της ευκολίας που προσφέρει όσο αφορά την ανταλλαγή σύντομων γραπτών μηνυμάτων μπορεί να χρησιμοποιηθεί κατά την επικοινωνία ενός μέλους της κοινότητας με τον Community Server αλλά και για να επικοινωνεί ο Community Server με τους παρόχους υπηρεσίας (SP). Ακόμα με τη σύμφωνη γνώμη του παραλήπτη ο Community Server μπορεί να ενημερώνει τα μέλη της κοινότητας ή ακόμα και άτομα που δεν έχουν εγγραφεί στη κοινότητα σχετικά με καινούργιες υπηρεσίες που παρέχονται ή κάποιες προσφορές όσο αφορά τη χρέωση των υπηρεσιών.

Ένα υποψήφιο μέλος της κοινότητας μπορεί να εγγραφεί αποστέλλοντας το username με το οποίο θέλει να αναγνωρίζεται στη κοινότητα καθώς και το password του μέσω ενός μηνύματος SMS. Το Network Module το οποίο παραλαμβάνει το μήνυμα από το κινητό τερματικό του χρήστη αποστέλλει στον Community Server το περιεχόμενο του μηνύματος μαζί με το MSISDN του αποστολέα. Ο Community Server τότε καταχωρεί το νέο μέλος στη βάση δεδομένων υπάρχουν καταχωρημένα όλα τα μέλη της κοινότητας.

Αφού γίνει η καταχώρηση επιτυχώς τότε ο Community Server αποστέλλει στο νέο μέλος ένα μήνυμα SMS με το οποίο τον καλωσορίζει στη κοινότητα. Αν για οποιονδήποτε λόγο η διαδικασία δεν ολοκληρωθεί κανονικά ο Community Server θα στείλει μήνυμα αναφέροντας τι προκάλεσε σφάλμα στη διαδικασία.

Όταν μια εφαρμογή-υπηρεσία ζητήσει από το Community Server πληροφορία για τη διεκπεραίωση ενός αιτήματος κάποιου SR τότε ο Community Server πρέπει να ενημερώσει τους παρόχους υπηρεσίας της εφαρμογής-υπηρεσίας που είναι διαθέσιμοι να στείλουν πληροφορία. Ο Community Server στέλνει ένα μήνυμα SMS με παραλήπτες τις συσκευές των παρόχων υπηρεσίας. Το συγκεκριμένο μήνυμα SMS περιέχει UDH με το οποίο δίνεται η δυνατότητα, χωρίς να το καταλάβει ο διαθέσιμος πάροχος υπηρεσίας, να ενεργοποιηθεί η εφαρμογή που έχει εγκαταστήσει στο τερματικό του και να αποστείλει ένα απαντητικό μήνυμα που περιέχει τη πληροφορία που χρειάζεται η εφαρμογή υπηρεσία. Η συγκεκριμένη υλοποίηση της ενημέρωσης των παρόχων υπηρεσίας δίνει τα εξής πλεονεκτήματα :

α) Εάν το παρέχει το τηλεπικοινωνιακό δίκτυο να γίνεται αποστολή μηνύματος SMS με broadcasting σε όλους τους διαθέσιμους παρόχους-υπηρεσίας που βρίσκονται στην ίδια κυψέλη

β) Η όλη επικοινωνία δεν χρησιμοποιεί κανένα από τους διαύλους επικοινωνίας που μπορεί να χρησιμοποιηθούν για υπηρεσίες φωνής και δεδομένων, μειώνοντας έτσι τη συμφόρηση στο δίκτυο.

4.5- ΤΟ ΠΡΩΤΟΚΟΛΛΟ WAP

Το WAP (Wireless Access Protocol) έχει σαν στόχο την παροχή περιεχομένου που βρίσκεται στο διαδίκτυο και εξειδικευμένων υπηρεσιών σε κινητά τερματικά. Η οικογένεια των πρωτοκόλλων που το αποτελούν παρέχει τη δυνατότητα λειτουργίας του WAP πάνω από διαφόρων ειδών ασύρματα δίκτυα και την απεικόνιση ιστοσελίδων σε κινητές συσκευές περιορισμένης ανάλυσης εικόνας και περιορισμένου εύρους ζώνης. Οι κινητές συσκευές που υποστηρίζουν WAP παρέχουν στους χρήστες τους τη δυνατότητα να λαμβάνουν και να αποκρίνονται σε μηνύματα ηλεκτρονικού ταχυδρομείου (e-Mail), πρόσβαση σε βάσεις δεδομένων και επικοινωνία με πληροφορία που βρίσκεται στο διαδίκτυο.

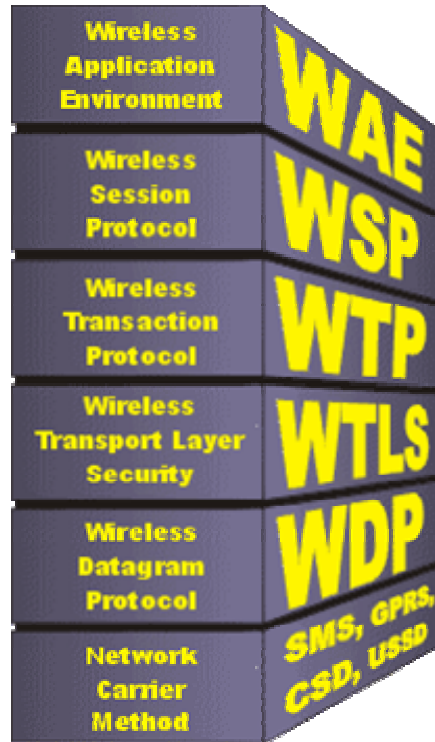
4.5.1- Η στοίβα πρωτοκόλλων του WAP

Τα πρωτόκολλα που αποτελούν το WAP μπορούν να χωριστούν σε κατηγορίες ανάλογα του επιπέδου επικοινωνίας στο οποίο το κάθε ένα από αυτά επενεργεί.

- Επίπεδο εφαρμογής: WAE (Wireless Application Environment)
- Επίπεδο συνόδου: WSP (Wireless Session Protocol)
- Επίπεδο συναλλαγής: WTP (Wireless Transaction Protocol)

- Επίπεδο ασφάλειας: WTLS (Wireless Transport Layer Security)
- Επίπεδο μεταφοράς: WDP (Wireless Datagram Protocol)

Η στοίβα των πρωτοκόλλων του WAP μπορεί να αναπαρασταθεί σχηματικά όπως φαίνεται στο σχήμα 23.



Σχήμα 27. Η στοίβα των πρωτοκόλλων του WAP

Το κάθε επίπεδο της στοίβας επικοινωνεί με το αμέσως χαμηλότερο του επίπεδο. Αυτό δίνει τη δυνατότητα σε όσους αναπτύσσουν λογισμικό να δημιουργούν εφαρμογές και υπηρεσίες για ένα συγκεκριμένο επίπεδο.

Το Wireless Application Environment βρίσκεται στο υψηλότερο σημείο της αρχιτεκτονικής του WAP. Είναι υπεύθυνο και παρέχει μια διασύνδεση που επιτρέπει την επικοινωνία με τον χρήστη. Περιέχει την γλώσσα ανάπτυξης περιεχομένου WML (Wireless Markup Language) καθώς και την γλώσσα ανάπτυξης κώδικα WMLScript, η οποία ενθυλακώνεται στο περιεχόμενο μιας WML σελίδας.

Το Wireless Session Protocol περιέχει όλες τις προδιαγραφές που χρειάζονται για τη πραγματοποίηση μιας συνόδου. Καταρχήν καθορίζει εάν η σύνοδος θα γίνει βάσει μιας συνδεδειστροφούς (connection-oriented) υπηρεσίας ή μιας ασυνδεδειστροφούς (connectionless-oriented) υπηρεσίας. Στη πρώτη περίπτωση έχουμε μετά την εκκίνηση μιας συνόδου, ανταλλαγή δεδομένων προς και από το κινητό τερματικό και αφού ολοκληρωθεί η μεταφορά δεδομένων τερματισμό της συνόδου (GPRS, CSD, HTTP). Σε αυτή τη περίπτωση το WSP επικοινωνεί και στέλνει δεδομένα στο επίπεδο που βρίσκεται το WTP. Εάν η σύνοδος γίνεται βάσει ασυνδεδειστροφούς υπηρεσίας (SMS, USSD) τότε το WSP επικοινωνεί και στέλνει δεδομένα στο επίπεδο που βρίσκεται το WDP.

Το Wireless Transaction Protocol εκτελεί τον απαραίτητο έλεγχο συμφόρησης που χρειάζεται κατά τη διάρκεια μιας συνδεδεσιστρεφούς συνόδου. Έτσι εξασφαλίζεται η ομαλή και εύρυθμη ανταλλαγή των δεδομένων. Επίσης κατηγοριοποιεί κάθε αίτημα συναλλαγής το οποίο μπορεί να ανήκει σε μια από τις πιο κάτω κατηγορίες:

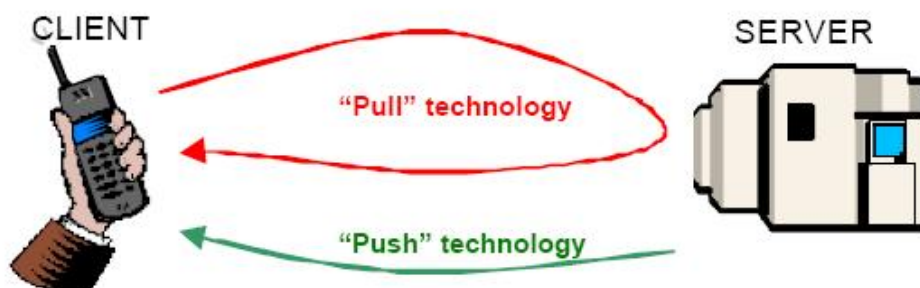
- α) αξιόπιστο με ανταλλαγή πληροφορίας και από τα δυο μέλη της συνόδου
- β) αξιόπιστο με προσφορά πληροφορίας μόνο από το ένα μέλος της συνόδου
- γ) αναξιόπιστο με προσφορά πληροφορίας μόνο από το ένα μέλος της συνόδου

Το Wireless Transport Layer Security διασφαλίζει ασφάλεια κατά την ανταλλαγή των δεδομένων. Χρησιμοποιείται συνήθως σε εφαρμογές ηλεκτρονικού εμπορίου (m-commerce) και τραπεζικών εφαρμογών (WAP Banking). Παρέχει μεθόδους κρυπτογράφησης και ταυτοποίησης καθώς και διασφαλίζει την ακεραιότητα των δεδομένων.

Το Wireless Datagram Protocol διασφαλίζει την επικοινωνία των προηγούμενων πρωτοκόλλων με το δίκτυο. Αναλαμβάνει την εύκολη προσαρμογή του WAP σε διάφορους τύπους δικτύων. Αυτό το πρωτόκολλο καθιστά όλη τη στοίβα των πρωτοκόλλων του WAP ανεξάρτητη του δικτύου στο οποίο υλοποιείται.

4.5.2- Η τεχνολογία WAP Push

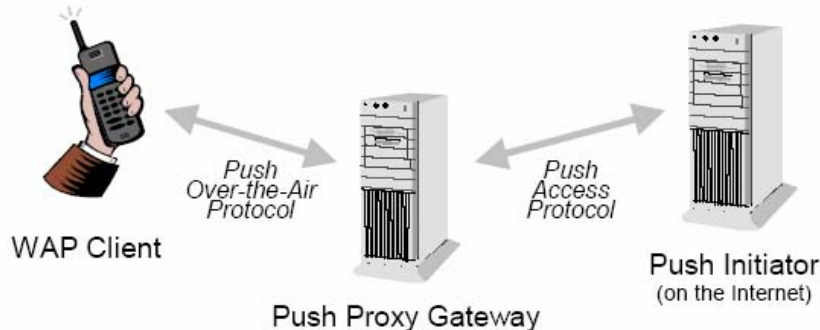
Στο μοντέλο πελάτη-εξυπηρετητή (client-server) ο πελάτης ζητά μια υπηρεσία ή πληροφορία από τον εξυπηρετητή ο οποίος αποκρίνεται στέλνοντας τη ζητούμενη πληροφορία στον πελάτη. Αυτό είναι γνωστό ως τεχνολογία pull, δηλαδή ο πελάτης «τραβά» τη πληροφορία που θέλει από τον εξυπηρετητή. Σε αντίθεση με αυτή τη τεχνολογία βρίσκεται η τεχνολογία push η οποία είναι και αυτή βασισμένη στο μοντέλο πελάτη-εξυπηρετητή. Η διαφορά όμως έγκειται στο γεγονός ότι ο εξυπηρετητής μεταδίδει πληροφορία στο πελάτη χωρίς ο τελευταίος να έχει κάνει οποιοδήποτε αίτημα παροχής της πληροφορίας από τον εξυπηρετητή.



Σχήμα 28. Οι τεχνολογίες pull και push

Κατά τη αποστολή πληροφορίας μέσω WAP χρησιμοποιώντας τη τεχνολογία push η οντότητα που θέλει να στείλει τη πληροφορία (Push

Initiator, PI), στέλνει ένα μήνυμα που περιέχει τη πληροφορία καθώς και οδηγίες παράδοσης της σε ένα Push Proxy Gateway (PPG). Ακολούθως βάσει των οδηγιών παράδοσης το PPG παραδίδει τη πληροφορία στο κινητό τερματικό του παραλήπτη, το οποίο πρέπει να υποστηρίζει τη τεχνολογία WAP push. Η διαδικασία φαίνεται στο σχήμα 25.



Σχήμα 29. Οι οντότητες και τα πρωτόκολλα που χρησιμοποιούνται στο WAP push

Ο *Push Initiator* είναι μια εφαρμογή που λειτουργεί σε ένα συνηθισμένο εξυπηρετητή ιστού (web server). Χρησιμοποιεί το πρωτόκολλο Push Access Protocol (PAP) για να επικοινωνεί με το PPG.

Το *Push Access Protocol* χρησιμοποιεί τυποποιημένα πρωτόκολλα τα οποία εφαρμόζονται στο διαδίκτυο. Η γλώσσα XML χρησιμοποιείται για να εκφραστούν οι οδηγίες παράδοσης του περιεχομένου, ενώ το περιεχόμενο μπορεί να είναι οποιοσδήποτε γνωστός τύπος πληροφορίας (MIME Types). Το πρωτόκολλο PAP είναι σχεδιασμένο έτσι ώστε να είναι ανεξάρτητο από τα πρωτόκολλα μεταφοράς. Μπορεί να χρησιμοποιηθεί οποιοδήποτε πρωτόκολλο μεταφοράς μπορεί να μεταφέρει MIME Type πληροφορίες. Το PAP υποστηρίζει τις ακόλουθες λειτουργίες:

- Αποστολή Push (PI προς PPG)
- Αναφορά αποτελέσματος (PPG προς PI)
- Ακύρωση μηνύματος Push (PI προς PPG)
- Αντικατάσταση μηνύματος Push (PI προς PPG)
- Ερώτηση κατάστασης (PI προς PPG)
- Ερώτηση για τις δυνατότητες του παραλήπτη (PI προς PPG)

Ο σκοπός της αποστολής Push (Push Submission) είναι η παράδοση ενός μηνύματος push στο PPG και από εκεί στη κατάλληλη εφαρμογή του κινητού τερματικού του παραλήπτη. Ένα μήνυμα push περιέχει υποχρεωτικά τις οντότητες ελέγχου (control entity) και περιεχομένου (content entity) και προαιρετικά την οντότητα δυνατοτήτων του παραλήπτη (capabilities entity). Η οντότητα ελέγχου είναι κείμενο σε μορφή XML που περιέχει οδηγίες παράδοσης με παραλήπτη το PPG. Στην οντότητα περιεχομένου βρίσκεται η πληροφορία που θα λάβει η τερματική συσκευή του παραλήπτη. Η οντότητα δυνατοτήτων του παραλήπτη περιέχει τις δυνατότητες που πρέπει να έχει το κινητό τερματικό του παραλήπτη του μηνύματος ούτως ώστε να μπορεί να χρησιμοποιήσει τις πληροφορίες που του στέλνει. Εάν οι δυνατότητες που έχει το τερματικό του παραλήπτη, όπως τις γνωρίζει ο PPG, δεν επιτρέπουν την χρήση της πληροφορίας για τον σκοπό που έχει σταλεί, τότε ο PPG θα απορρίψει το μήνυμα. Η

απάντηση ενός μηνύματος push είναι ένα κείμενο XML το οποίο αναφέρει επιτυχία ή αποτυχία της διαδικασίας αποστολής μηνύματος στο PPG. Η επιτυχία ή αποτυχία της διαδικασίας εξαρτάται από την επικύρωση ή όχι της οντότητας ελέγχου, βάσει ενός αρχείου επικύρωσης του κειμένου XML που αναμένεται (DTD). Πρέπει να αναφερθεί ότι όσοι και αν είναι οι παραλήπτες του συγκεκριμένου περιεχομένου η απάντηση του PPG είναι μόνο μία για κάθε αποστολή push από τον PI.

Η αναφορά αποτελέσματος (Result Notification) έχει ως αποτέλεσμα την ενημέρωση του Push Initiator, αφού προηγουμένως το έχει ζητήσει, σχετικά με την έκβαση της αποστολής ενός μηνύματος push. Η αναφορά αυτή στέλνεται σε συγκεκριμένο URI που έχει διευκρινιστεί από τον PI. Ένα Result Notification επιστρέφεται στον PI όταν το μήνυμα έχει πάψει να θεωρείται υποψήφιο για μετάδοση, δηλαδή είτε έχει παραδοθεί κανονικά, είτε λόγω προβλήματος δεν έχει παραδοθεί στο κινητό τερματικό του παραλήπτη. Η αναφορά περιέχει ένα κείμενο σε XML με την έκβαση της αποστολής του μηνύματος και προαιρετικά πληροφορία που έστειλε ο χρήστης του κινητού τερματικού.

Η ακύρωση διαδικασίας push (Push Cancellation) είναι ένα κείμενο σε XML που απαιτεί από τον PPG την ακύρωση ενός μηνύματος που έστειλε ο PI προηγουμένως. Ο PPG απαντά με ένα κείμενο σε XML αναφέροντας εάν η ακύρωση ήταν επιτυχής.

Η αντικατάσταση μηνύματος Push (Push Replacement) γίνεται αποστέλλοντας ένα νέο μήνυμα και ζητώντας να αντικατασταθεί το παλιό. Υπάρχει η δυνατότητα ο PI που επιθυμεί την αντικατάσταση του μηνύματος, να διευκρινίσει εάν το νέο μήνυμα θα σταλεί σε όλους τους παραλήπτες ή μόνο σε όσους δεν το έχουν λάβει.

Η ερώτηση κατάστασης (Status Query) είναι ένα κείμενο XML με το οποίο ο PI ζητά από τον PPG να ενημερωθεί για την εξέλιξη μιας διαδικασίας αποστολής μηνύματος Push. Ο PPG απαντά, με ένα μήνυμα XML το οποίο περιέχει την τρέχουσα κατάσταση της διαδικασίας.

Η ερώτηση για τις δυνατότητες του παραλήπτη (Client Capabilities Query) είναι ένα κείμενο σε XML με το οποίο ο PI ζητά από τον PPG να ενημερωθεί για τις δυνατότητες μιας συγκεκριμένης κινητής συσκευής που είναι συνδεδεμένη στο δίκτυο. Ο PPG απαντά με ένα μήνυμα XML το οποίο αποτελείται από δύο μέρη. Το πρώτο μέρος αναφέρεται στο αποτέλεσμα του ερωτήματος ενώ το δεύτερο περιέχει τις δυνατότητες της συσκευής σύμφωνα με το WAP User Agent Profile vocabulary.

Ο *Push Proxy Gateway* αναλαμβάνει τη μεταφορά του περιεχομένου ενός μηνύματος push από το διαδίκτυο στο δίκτυο κινητών τηλεπικοινωνιών. Οι υπηρεσίες που προσφέρει στη διαδικασία αποστολής ενός WAP push μηνύματος είναι:

- Έλεγχο πρόσβασης με ταυτοποίηση και εξουσιοδότηση των διαπιστευμένων Push Initiators.
- Έλεγχο σφαλμάτων και ανάλυση των μηνυμάτων των Push Initiators.
- Υπηρεσίες εύρεσης και αναγνώρισης δυνατοτήτων τερματικών συσκευών
- Συμπίεση και κωδικοποίηση συγκεκριμένων τύπων δεδομένων για αποδοτικότερη αποστολή τους μέσω του δικτύου κινητών τηλεπικοινωνιών(Over the Air, OTA).
- Μετατροπή πρωτοκόλλων.

Ακόμα, αναλόγως της υλοποίησης του, μπορεί να υποστηρίξει broadcasting και multicasting μεθόδους παράδοσης σε πολλούς παραλήπτες.

Το Push Over the Air Protocol αναλαμβάνει τη μετάδοση του περιεχομένου ενός μηνύματος push από τον PPG στους παραλήπτες του και στις εφαρμογές που χρησιμοποιούν. Είναι σχεδιασμένο να λειτουργεί πάνω από τα πρωτόκολλα HTTP (OTA-HTTP) και WSP (OTA-WSP). Το πρωτόκολλο OTA-WSP μπορεί να χρησιμοποιήσει όλες τις μεθόδους μετάδοσης και τις δυνατότητες που υποστηρίζει το WSP καθώς και αναγκαίες προεκτάσεις τους προσθέτοντας καινούργιες επιπλέον κεφαλίδες στα WAP μηνύματα. Μπορεί να χρησιμοποιήσει είτε συνδεοστρεφείς είτε ασυνδεοστρεφείς υπηρεσίες. Το πρωτόκολλο OTA-HTTP χρησιμοποιείται κυρίως σε συνεργασία με TCP/IP συνδέσεις. Μπορεί να χρησιμοποιήσει μόνο συνδεοστρεφείς υπηρεσίες. Η παράδοση του περιεχομένου γίνεται με την μέθοδο HTTP POST με την οποία ο PPG αναλαμβάνει ρόλο πελάτη ενώ η τερματική συσκευή του χρήστη ρόλο εξυπηρετητή.

Υπάρχουν δύο μέθοδοι για την εγκατάσταση μιας ενεργής σύνδεσης TCP μεταξύ PPG και κινητού τερματικού ούτως ώστε να σταλεί ένα μήνυμα push. Η πρώτη μέθοδος επιτρέπει στο PPG να ξεκινήσει μια σύνδεση με ένα συγκεκριμένο τερματικό (PPG Originated TCP, PO-TCP) εφόσον το τερματικό είναι συνδεδεμένο με το δίκτυο (ή μπορεί να το συνδέσει ο PPG) και ο PPG γνωρίζει την IP διεύθυνση του. Η δεύτερη μέθοδος επιτρέπει στο τερματικό να ξεκινήσει μια σύνδεση με τον PPG (Terminal Originated TCP, TO-TCP) και συνήθως χρησιμοποιείται σε συνεργασία με μηνύματα SIR (Session Initiation Request).

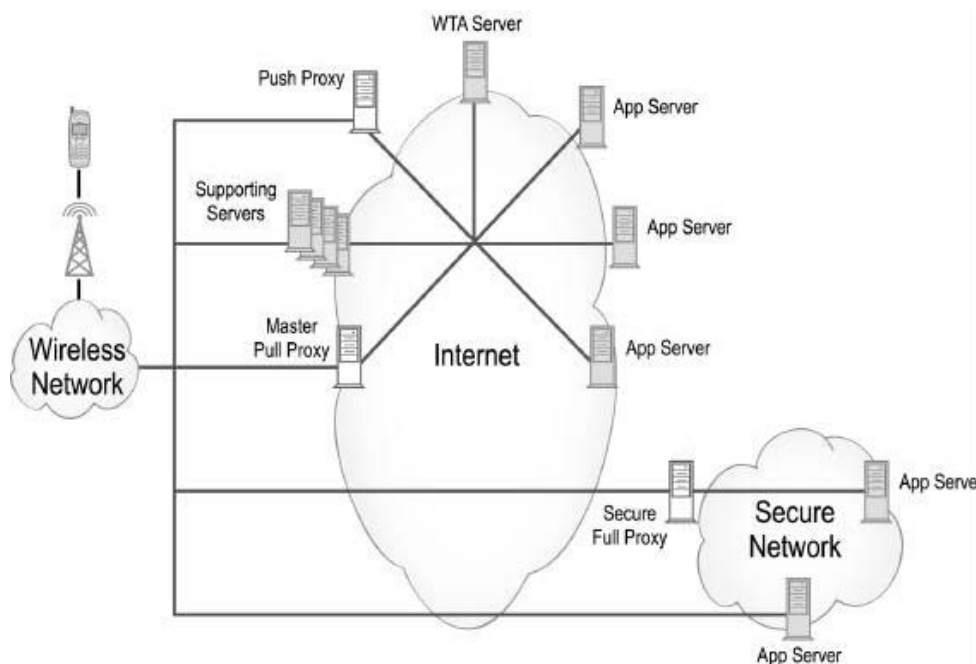
Είτε χρησιμοποιείται OTA-HTTP είτε OTA-WSP είναι επιθυμητό, τις παραπάνω φορές, τη διαδικασία εγκατάστασης σύνδεσης με το δίκτυο να ξεκινά από το τερματικό. Αυτό συμβαίνει επειδή συνήθως μόνο το τερματικό μπορεί να ξεκινήσει μια σύνδεση WSP ή επειδή ο PPG δεν γνωρίζει την IP διεύθυνση του τερματικού εάν αυτό είναι ήδη συνδεδεμένο. Για το σκοπό αυτό εγκατεστημένη στο τερματικό πρέπει να υπάρχει η εφαρμογή SIA (Session Initiation Application). Η εφαρμογή SIA «ακούει» σε μηνύματα SIR τα οποία στέλνει ο PPG στο τερματικό και ακολούθως ενεργοποιεί τον κατάλληλο φορέα, όπως του το υποδεικνύει το μήνυμα SIR, για να εγκαταστήσει μια σύνδεση επικοινωνίας (OTA-HTTP ή OTA-WSP) με το PPG.

Το μήνυμα SIR χρησιμοποιεί ασυνδεοστρεφείς υπηρεσίες για να σταλεί σε ένα τερματικό και η συνηθέστερη είναι η υπηρεσία SMS. Πρέπει

να απαιτείται προσοχή ούτως ώστε ένα μήνυμα SIR να χωράει σε ένα μήνυμα SMS. Η υπηρεσία SMS έχει επιλεγεί λόγω του ότι υποστηρίζεται από όλα τα δίκτυα GSM, προσφέρει διευθυνσιοδότηση χρησιμοποιώντας το MSISDN του τερματικού και μπορεί να προσφέρει μια αξιόπιστη μεταφορά του μηνύματος.

4.5.3- Στοιχεία δικτύου για υποστήριξη WAP

Ένα τερματικό που υποστηρίζει WAP μπορεί να επικοινωνήσει με ένα εξυπηρετητή (Application Server) είτε μέσω διαφόρων ενδιάμεσων εξυπηρετητών (proxies) είτε κατευθείαν. Οι ενδιάμεσοι εξυπηρετητές αναλαμβάνουν την μετατροπή των πρωτοκόλλων του WAP σε πρωτόκολλα WWW (HTTP, TCP) ούτως ώστε να επιτρέπουν στον χρήστη να χρησιμοποιεί το διαδίκτυο. Ένα τυπικό δίκτυο WAP φαίνεται στο παρακάτω σχήμα.



Σχήμα 30. Ένα δίκτυο WAP.

Οι proxies μπορεί να βρίσκονται είτε στο δίκτυο κινητών τηλεπικοινωνιών είτε να είναι ανεξάρτητοι εξυπηρετητές έτσι ώστε να παρέχονται στο τερματικό δυνατότητες συνυφασμένες με το ασύρματο δίκτυο (τηλεφωνία, εύρεση θέσης, διαχείριση συσκευής) καθώς και δυνατότητες βελτιστοποίησης της επικοινωνίας μεταξύ κινητού τερματικού και εξυπηρετητή (Application Server). Για την ασφαλή επικοινωνία μεταξύ ενός τερματικού με ένα εξυπηρετητή μπορεί να χρησιμοποιηθούν proxies που βρίσκονται σε ασφαλές δίκτυο (Secure Network). Ασφαλής επικοινωνία μπορεί να γίνει και απευθείας με ένα εξυπηρετητή που μπορεί να παρέχει αυτή τη δυνατότητα.

4.5.5- Εφαρμογή του πρωτοκόλλου WAP στη προτεινόμενη πλατφόρμα

Το πρωτόκολλο WAP λαμβάνει υπόψιν τις περιορισμένες δυνατότητες του μεγαλύτερου αριθμού κινητών συσκευών στην αγορά. Επίσης αποτελείται από μια ολοκληρωμένη σειρά από πρωτόκολλα τα οποία παρέχουν όλες τις δυνατότητες (έλεγχος συμφόρησης, κρυπτογράφηση δεδομένων, ταυτοποίηση) που πρέπει να υποστηρίζονται κατά την ανταλλαγή δεδομένων μεταξύ ενός κινητού τερματικού και ενός εξυπηρετητή είτε αυτός βρίσκεται στο διαδίκτυο είτε στο τηλεπικοινωνιακό δίκτυο. Το γεγονός αυτό το καθιστά αναγκαίο για την υλοποίηση μιας εύχρηστης διασύνδεσης ενός μέλους της κοινότητας τόσο με τον Community Server όσο και με το Application Service η οποία να απευθύνεται σε ένα ευρύ φάσμα χρηστών κινητών τερματικών. Το γεγονός αυτό όμως δεν αποτρέπει χρήστες που χρησιμοποιούν κινητές συσκευές με αυξημένες δυνατότητες να χρησιμοποιούν διασυνδέσεις με πιο πολλές δυνατότητες.

Επίσης το πρωτόκολλο WAP λόγω του ότι είναι ανεξάρτητο του τηλεπικοινωνιακού δικτύου στο οποίο εφαρμόζεται παρέχει τη δυνατότητα στον Community Server να επικοινωνεί με τα μέλη της κοινότητας όποιος και αν είναι ο φορέας του δικτύου (SMS, IP πακέτα). Οι χρήστες που ζητούν υπηρεσία (SR) και δεν έχουν τη δυνατότητα σύνδεσης με το διαδίκτυο μπορούν να επικοινωνούν με ασυνδεδειστροφή τρόπο (SMS) με ένα WAP Proxy το οποίο θα βρίσκεται στον Community Server και θα αναλαμβάνει τη μετατροπή των πρωτοκόλλων WWW (HTTP, TCP/IP) σε πρωτόκολλα WAP. Το WAP Proxy μπορεί να χρησιμοποιήσει και ένας SR ο οποίος μπορεί να συνδεθεί με το διαδίκτυο, αλλά η συσκευή του δεν έχει τη δυνατότητα να χρησιμοποιήσει τα πρωτόκολλα WWW και έχει τη δυνατότητα χρήσης των WAP πρωτοκόλλων (οι πιο διαδεδομένες κινητές συσκευές ανήκουν σε αυτή την κατηγορία).

Εκτός από την επικοινωνία για αίτηση παροχής υπηρεσίας ο Community Server, εγκαθιστώντας μια εφαρμογή Push Initiator και δεδομένης της υποστήριξης του Network Module το οποίο θα υλοποιεί ένα Push Proxy Gateway, μπορεί να επικοινωνεί με τα μέλη της κοινότητας όποτε το επιθυμεί χωρίς προηγουμένως να πρέπει να το ζητήσουν τα ίδια τα μέλη. Αυτό προσφέρει τη δυνατότητα στο Community Server εκτός από την ενημέρωση των μελών της κοινότητας, που θα μπορούσε να χρησιμοποιηθούν SMS, να προσφέρει σε χρήστες κινητών τερματικών που δεν είναι μέλη της κοινότητας τη δυνατότητα να πάρουν μια ιδέα από τις υπηρεσίες που παρέχει η κοινότητα έτσι ώστε να γίνουν και αυτοί εάν το επιθυμούν μέλη της κοινότητας.

Όσο αφορά τους παρόχους υπηρεσίας (SP), όταν μια εφαρμογή υπηρεσία ζητά πληροφορία η οποία απαιτεί από τον SP να την εισάγει ο ίδιος την ώρα που θα του ζητηθεί, μπορούν να χρησιμοποιήσουν το WAP, το οποίο μέσω WML σελίδων να δίνει τη δυνατότητα να καταχωρούν την πληροφορία που τους ζητείται. Αυτό το είδος ειδοποίησης των παρόχων υπηρεσίας δεν έρχεται σε αντίθεση με εφαρμογή της υπηρεσίας SMS που αναφέρθηκε στο προηγούμενο κεφάλαιο. Η ειδοποίηση των διαθέσιμων παρόχων όταν πρόκειται να χρησιμοποιήσουν WAP, γίνεται με την τεχνολογία WAP push με μηνύματα SIR που χρησιμοποιούν μηνύματα SMS για την μεταφορά τους.

4.6- Η ΤΕΧΝΟΛΟΓΙΑ JSP

Οι *JSPs* (*JavaServer Pages*) είναι μια τεχνολογία που έχει δημιουργηθεί από την εταιρεία Sun Microsystems για να μπορεί να δημιουργεί δυναμικό περιεχόμενο (dynamic content) στο Web. Πρόκειται για HTML έγγραφα (ιστοσελίδες) τα οποία αναμειγνύονται με τη γλώσσα προγραμματισμού Java, η οποία και έχει τη δυνατότητα να παρέχει (δημιουργεί) αυτό το δυναμικό περιεχόμενο.

Οι JSPs είναι μια εφαρμογή στην πλευρά του server (server-side application), που σημαίνει ότι δέχονται μια αίτηση (request) και παράγουν μια απόκριση ή απάντηση (response). Σε γενικές γραμμές, οι αιτήσεις γίνονται από έναν Web client και η απόκριση είναι ένα παραγόμενο HTML έγγραφο το οποίο στέλνεται πίσω στον Web client. Επειδή οι JSPs είναι μια εφαρμογή στην πλευρά του server, έχουν πρόσβαση σε πηγές (resources) στον server, όπως είναι τα Servlets, JavaBeans, EJBs, αλλά και σε βάσεις δεδομένων.

Οι ιστοσελίδες που περιέχουν κώδικα JSP (*JavaServer Pages*), δηλ. ανάμιξη κώδικα HTML με κώδικα Java, μετατρέπονται σε Servlets πριν εκτελεσθούν στον server. Ένα Servlet, στη γενική του μορφή, είναι μια τάξη της Java που υλοποιεί (implements) το interface Servlet και δέχεται αιτήσεις (requests) και παράγει (δημιουργεί) αποκρίσεις (responses). Οι αιτήσεις μπορεί να προέρχονται από τάξεις της Java, από Web clients ή και από άλλα Servlets.

4.6.1- Το Δυναμικό Web

Τα προγράμματα υποστήριξης δυναμικού web βρίσκονται στον server και δέχονται αιτήσεις (requests), χρησιμοποιούν τις πηγές που βρίσκονται στην πλευρά του server (server-side resources) και δημιουργούν μια σελίδα HTML σαν απόκριση (response) στην αίτηση του χρήστη. Επειδή αυτά τα προγράμματα καθορίζουν το HTML έγγραφο που στέλνεται πίσω στον client, οι χρήστες λαμβάνουν αποκρίσεις (responses) που ανταποκρίνονται στις συγκεκριμένες τους απαιτήσεις.

4.6.2- To client-side scripting

Μια άλλη τεχνολογία που εισήχθη στη δομή του Δυναμικού Web είναι το *client-side scripting*. Οι scripting languages είναι διερμηνευόμενες (interpreted) γλώσσες, δηλ. δεν μεταγλωττίζονται πριν από την εκτέλεση, αλλά ελέγχονται (διερμηνεύονται) κατά μία γραμμή τη φορά. Τέτοιες γλώσσες είναι η JavaScript και η VBScript. Το client-side scripting απελευθέρωσε λίγο από δουλειά τον server καθώς έδωσε τη δυνατότητα σε μικρές εφαρμογές, όπως είναι η επικύρωση των στοιχείων μιας φόρμας (form validation), να λαμβάνουν χώρα στον client. Όταν ο client προσπαθεί να υποβάλει (submit) μια φόρμα, μπορεί να εκτελεσθεί ένα script για τον έλεγχο συμπλήρωσης όλων των απαραίτητων πεδίων (fields).

Αν λείπει μια τιμή από ένα πεδίο, μπορεί να εμφανισθεί ένα μήνυμα στον client για να γνωρίζει ο χρήστης ότι είναι απαραίτητη η καταχώρηση κάποιας τιμής στο συγκεκριμένο πεδίο. Η απλή αυτή δουλειά μπορεί να λάβει χώρα στον client χωρίς να απασχοληθεί καθόλου ο server. Η JavaScript είναι η scripting language του Netscape και η VBScript είναι η scripting language του Internet Explorer. Η VBScript αποτελεί ένα υποσύνολο της γνωστής γλώσσας προγραμματισμού Visual Basic και το σημαντικότερο μειονέκτημά της είναι ότι εργάζεται μόνο στον Internet Explorer, ενώ η JavaScript μπορεί να χρησιμοποιηθεί και στον Netscape Navigator και στον Internet Explorer.

4.6.3- Το πρωτόκολλο HTTP

Το WWW σχεδιάστηκε γύρω από έναν απλό μηχανισμό αίτησης/απόκρισης (request/response). Οι browsers που εκτελούνται στα μηχανήματα των clients αιτούν τα HTML έγγραφα και οι servers αποκρίνονται παραδίδοντας αυτά τα έγγραφα. Οι κανόνες και οι μηχανισμοί για τη μεταφορά των HTML εγγράφων αποτελούν το πρωτόκολλο *HTTP (Hypertext Transfer Protocol)*. Οι servers συνήθως δέχονται αιτήσεις από πολλούς διαφορετικούς clients και ασχολούνται μ' έναν συγκεκριμένο client μέχρις ότου ικανοποιήσουν μια αίτηση με μια απόκριση. Μετά απ' αυτό, οι servers ξεχνούν αυτόν τον client και συνεχίζουν για να ικανοποιήσουν αποκρίσεις από άλλους clients.

Το HTTP δεν παρέχει μηχανισμούς για να μπορέσει να διακρίνει έναν client από τον επόμενο, έτσι ώστε αν δύο αιτήσεις συμβεί να προέλθουν από τον ίδιο client, ο server να μην είναι σε θέση να το καταλάβει. Μ' αυτή την έννοια, το HTTP αναφέρεται σαν ένα stateless πρωτόκολλο. Αυτό το χαρακτηριστικό επέβαλε έναν περιορισμό στο τι μπορεί να κάνει ένα Web site. Οι servers δεν είναι σε θέση να θυμούνται στοιχεία πληροφορίας για έναν συγκεκριμένο χρήστη εφόσον δεν υπάρχει κάποιος τρόπος διαχωρισμού των χρηστών μεταξύ τους.

Ο server θα πρέπει να μπορεί να παρακολουθεί ποιες αιτήσεις ανήκουν σε ποιον χρήστη. Ο server θα πρέπει να είναι σε θέση να διατηρεί την κατάσταση (state) του κάθε χρήστη αλλά το HTTP δεν το επιτρέπει αυτό. Για να μπορέσει να λυθεί αυτό το πρόβλημα, η Netscape επινόησε την ιδέα του cookie. Τα cookies είναι κομμάτια πληροφοριών και συσχετισμένων μηχανισμών που αποθηκεύουν πληροφορίες που έχουν σχέση με τον χρήστη στα μηχανήματα των clients εξ ονόματος του server. Οι πληροφορίες αυτές στέλνονται μετά στον server κάθε φορά που ο browser κάνει μια αίτηση στον ίδιο server.

Οι πληροφορίες σ' ένα cookie μπορεί να περιέχουν ο,τιδήποτε και σε γενικές γραμμές χρησιμοποιούνται για να διακρίνουν τους διάφορους χρήστες που έχουν πρόσβαση στον ίδιο Web server. Ένα cookie μπορεί να περιέχει στοιχεία όπως το username, ένα ID, προσωπικές πληροφορίες ή ο,τιδήποτε άλλο που η εφαρμογή (application) του Web αποφασίσει να χρησιμοποιήσει για να μπορεί να παρακολουθεί συγκεκριμένους χρήστες.

Μια εφαρμογή του Web ορίζει ένα cookie ζητώντας από τον browser να δημιουργήσει ένα cookie στο μηχάνημα του client όπου εκτελείται ο browser. Το πλήθος των πληροφοριών που περιέχει ένα cookie καθώς και ο αριθμός των cookies που μπορούν να γίνουν επιτρεπτά είναι συνήθως περιορισμένα επειδή ένας server δεν θα πρέπει να αφήνεται ελεύθερος να χρησιμοποιεί το μηχάνημα του client σαν απομακρυσμένη συσκευή αποθήκευσης.

Τα JSPs παρέχουν τρεις τρόπους για να υπάρχει μια επικοινωνία ανάμεσα στον browser στην πλευρά του client και την εφαρμογή του server : τα cookies, τις συνόδους (sessions) και την κωδικοποίηση των URLs.

4.6.4- Αποθήκευση και ανάκτηση των Cookies

Τα cookies δημιουργούνται από την εφαρμογή στην πλευρά του server, δηλ. το JSP, και αποθηκεύονται στο μηχάνημα του client εξ ονόματος του JSP. Τα JSPs μπορούν να προσθέσουν cookies στο μηχάνημα του client χρησιμοποιώντας τη μέθοδο (συνάρτηση) *addCookie()* του αντικειμένου *response*. Η μέθοδος *addCookie()* λαμβάνει ένα στιγμιότυπο (instance) της τάξης (class) *Cookie*, η οποία μπορεί να περιέχει πληροφορίες σχετικά με τον χρήστη.

Το μηχάνημα του client αποθηκεύει τα cookies σ' έναν φάκελο που ορίζεται από τον browser που εκτελείται στο μηχάνημα του client. Τα cookies ανακτώνται από την JSP από το μηχάνημα του client όταν ο client στείλει μια αίτηση στο JSP. Τα JSPs έχουν πρόσβαση στα cookies χρησιμοποιώντας τη μέθοδο *getCookies()* του αντικειμένου *request*. Η μέθοδος *getCookies()* επιστρέφει έναν πίνακα από αντικείμενα *Cookie*. Τα cookies παριστάνονται από την τάξη *Cookie* σαν αντικείμενα τάξης στην πλευρά του server.

4.6.5- Οι Σύνοδοι Χρήστη (User Sessions)

Το interface *HttpSession* δίνει στα JSPs τη δυνατότητα να αποθηκεύσουν και να μοιραστούν πληροφορίες για μια αλληλεπίδραση μ' έναν χρήστη που βρίσκεται σε εξέλιξη. Μπορούμε να έχουμε πρόσβαση στο interface *HttpSession* μέσω της μεθόδου *getSession()* του interface *HttpServletRequest*.

Μέσα σ' ένα JSP το αντικείμενο *session* υλοποιεί ένα στιγμιότυπο του interface *HttpSession*. Το αντικείμενο *session* χρησιμοποιείται σαν ένα σχεδιάγραμμα για τα JSPs μιας εφαρμογής e-commerce ώστε να παρακολουθούν τα προσωρινά δεδομένα του χρήστη με τον οποίο αλληλεπιδρούν. Τα δεδομένα θα πρέπει να είναι μόνιμα αποθηκευμένα σε μια βάση δεδομένων.

Ο κάθε μεμονωμένος χρήστης που ξεκινάει μια αλληλεπίδραση μ' ένα οποιοδήποτε JSP μιας εφαρμογής συσχετίζεται μ' ένα μοναδικό αντικείμενο *session*. Αυτό γίνεται δημιουργώντας ένα μοναδικό ID για τον κάθε χρήστη

που αλληλεπιδρά με την εφαρμογή και αποθηκεύοντάς το σαν ένα cookie στο μηχάνημα του πελάτη (client). Σε κάθε αίτηση (request) που κάνει ο πελάτης προς την εφαρμογή του server, χρησιμοποιείται το cookie που περιέχει το ID του πελάτη ώστε να μπορέσει να ξεχωρίσει ανάμεσα από πολλούς άλλους χρήστες. Το ID χρησιμοποιείται για να μπορέσει να καθορίσει το σωστό στιγμιότυπο του session για τον αντίστοιχο χρήστη.

Αυτός ο μηχανισμός επιτρέπει στα JSPs να διατηρούν μια ιδιωτική συνομιλία με τον κάθε χρήστη καθώς αυτός πηγαίνει από σελίδα σε σελίδα. Τα JSPs έχουν τη δυνατότητα να αποθηκεύσουν και να ανακτήσουν τα προσωρινά δεδομένα ενός session χρησιμοποιώντας τις μεθόδους `setAttribute()` και `getAttribute()`. Αυτές οι μέθοδοι επιτρέπουν στους developers να συνδέσουν ένα αντικείμενο που περιέχει δεδομένα του χρήστη σ' ένα string ώστε να μπορούν αυτά να ανακτηθούν με το όνομα.

4.6.6- Κωδικοποίηση των URLs

Τα cookies και τα sessions δουλεύουν μόνο αν οι χρήστες έχουν ενεργοποιήσει τα cookies στους browsers τους. Πολλοί χρήστες απενεργοποιούν τα cookies για διάφορους λόγους, όπως αν δεν θέλουν να παρακολουθούνται ή αν δεν θέλουν να αφήνουν ίχνη στα μηχανήματά τους για το ποιες ιστοσελίδες έχουν επισκεφθεί. Αν και ο χρήστης έχει κάθε δικαίωμα να το κάνει αυτό, είναι γεγονός ότι μερικές εφαρμογές είναι αδύνατο να τρέξουν χωρίς να κρατάνε κάποιες πληροφορίες για τους χρήστες. Μερικά sites απαιτούν από τους χρήστες να έχουν ενεργοποιήσει τα cookies στους browsers τους ώστε να μπορέσουν να δούψουν κάποια στοιχεία του site.

Οι εφαρμογές του Web σε γενικές γραμμές δεν θα πρέπει να παίρνουν ως δεδομένο το ότι τα cookies είναι ενεργοποιημένα και θα πρέπει να χρησιμοποιούν την κωδικοποίηση URL για να διατηρούν μια επικοινωνία με τις εφαρμογές του client.

Το ID του session μπορεί να αποθηκευθεί είτε σ' ένα cookie ή στο κωδικοποιημένο URL. Το κωδικοποιημένο URL που θα προκύψει θα περιέχει ένα ID που χρησιμοποιείται για να συσχετίσει τον χρήστη μ' ένα αντικείμενο `HttpSession`. Έτσι, ακόμη κι αν δεν είναι ενεργοποιημένα τα cookies, η εφαρμογή του Web μπορεί να κάνει χρήση ενός αντικειμένου session που συσχετίζεται με συγκεκριμένους χρήστες.

Η μέθοδος `encodeURL()` είναι σε θέση να αναγνωρίσει αν το URL θα πρέπει να κωδικοποιηθεί ή όχι. Αν τα cookies είναι ενεργοποιημένα, η μέθοδος `encodeURL()` δεν θα κωδικοποιήσει το URL και θα επιστρέψει το αρχικό URL. Διαφορετικά, θα κωδικοποιήσει (ενσωματώσει) ένα session ID μέσα στο URL.

4.6.7- JavaBeans

Τα *JavaBeans* είναι τάξεις της Java που ικανοποιούν την προδιαγραφή *JavaBean*. Αυτή η προδιαγραφή καθιερώνει κατευθυντήριες

οδηγίες για την ανάπτυξη κώδικα σε Java που να είναι εύκολο να ξαναχρησιμοποιηθεί σαν λειτουργικά μπλοκ σε πολλά σενάρια.

Από τεχνική άποψη, η JSP τεχνολογία κάνει όλη τη δουλειά ενός JavaBean. Αλλά στην πραγματικότητα τα JSPs σχεδιάστηκαν για να χρησιμοποιηθούν ως το επίπεδο παρουσίασης, το οποίο αποτελεί εκείνο το κομμάτι μιας εφαρμογής που έχει να κάνει με την άμεση αλληλεπίδραση με τον χρήστη. Αλλά, αυτή η εργασία είναι αρκετά πολύπλοκη. Υπάρχουν και άλλες πολύπλοκες εργασίες στην ανάπτυξη εφαρμογών, όπως είναι οι υπολογισμοί και η επεξεργασία των βάσεων δεδομένων που πρέπει να λάβουν χώρα για να ικανοποιηθούν οι αιτήσεις από την αλληλεπίδραση με τον χρήστη. Η καλύτερη τακτική είναι ο διαχωρισμός του επιπέδου παρουσίασης από τους υπολογισμούς και την επεξεργασία των βάσεων δεδομένων. Ο πολύ μεγάλος φόρτος υπολογισμών που συμβαίνει στο παρασκήνιο θα πρέπει να χειρισθεί από κάποιον μηχανισμό που να είναι αφοσιωμένος σ' αυτή τη δουλειά, όπως είναι τα JavaBeans. Η τάξη της Java που υλοποιεί τα JavaBeans θα έχει συνήθως έναν αριθμό από μεταβλητές, οι οποίες αναφέρονται σαν οι ιδιότητες των JavaBeans. Για καθεμία από τις μεταβλητές που χρειάζεται η προδιαγραφή της JavaBean, θα πρέπει να παρέχονται αντίστοιχες μέθοδοι *set* και *get* για να ορίζεται και να διαβάζεται, αντίστοιχα, την τιμή της μεταβλητής.

Οι εφαρμογές του Web διαιρούνται συχνά σε δύο επίπεδα: το *presentation layer* και το *business logic layer*. Το *presentation layer* είναι αυτό που βλέπει ο χρήστης και αλληλεπιδρά μαζί του σ' έναν browser. Το *business logic layer* αποτελείται από πολλές τάξεις της Java που συνεργάζονται μεταξύ τους για να μοντελοποιήσουν και να εκτελέσουν τη ροή εργασίας πχ του CS. Τα JSPs εμπίπτουν στην κατηγορία του *presentation layer* και τα JavaBeans στην κατηγορία του *business logic layer*. Τα JSPs προορίζονται να χρησιμοποιηθούν για τη δημιουργία δυναμικών ιστοσελίδων. Ο ενσωματωμένος κώδικας της Java στα *scriptlets* της JSP θα πρέπει να χρησιμοποιείται για να αποφασίσει προγραμματιστικά για τη διάταξη και το περιεχόμενο της σελίδας. Για σοβαρές υπολογιστικές εργασίες, μπορούμε να εξετάσουμε την περίπτωση δημιουργίας εξειδικευμένου κώδικα σε Java, όπως είναι τα JavaBeans. Τα *scriptlets* μπορούν να δημιουργήσουν στιγμιότυπα των τάξεων της Java και να τα χρησιμοποιήσουν ως αντικείμενα.

4.6.8- Εφαρμογή της τεχνολογίας JSP στην προτεινόμενη πλατφόρμα

Η JSP τεχνολογία, αποτελεί το ευέλικτο εκείνο μοντέλο που μπορεί να στηρίξει την επικοινωνία SR<->CS καθώς και SR<->AS στο *server-side scripting* κομμάτι. Παρέχοντας δυναμικό- προσωποποιημένο web, αποτελεί ουσιαστικά μια γραφική διεπαφή χρήστη (GUI), μέσω της οποίας ο SR έρχεται σε επικοινωνία με την κοινότητα. Βασισόμενη στο HTTP πρωτόκολλο, η JSP τεχνολογία παρέχει διαλειτουργικότητα και πρόσβαση στα *resources* του CS (Βάσεις Δεδομένων μελών, υπηρεσιών-εφαρμογών, παροχών υπηρεσιών). Τέλος, εξαιτίας της φύσης της, παρέχει/ δημιουργεί προσωποποιημένα στοιχεία με τρόπο ασφαλή.

Όσον αφορά στα JavaBeans: Οι φόρμες της HTML χρησιμοποιούνται για να συγκεντρώσουν πληροφορίες από τους χρήστες και αυτές οι πληροφορίες μπορούν να δοθούν (σταλούν) σε JSPs με τη μορφή ζευγαριών ονόματος/τιμής στο αντικείμενο request.

Τα JavaBeans μπορούν να χρησιμοποιηθούν για να μαζέψουν τις τιμές των παραμέτρων request σε μια συνεκτική και λογική μονάδα. Αυτό μπορεί να γίνει εξάγοντας τις τιμές των παραμέτρων στο αντικείμενο request που στέλνονται από τη φόρμα και δίνοντας τιμές στις ιδιότητες (properties) ενός κατάλληλα ορισμένου JavaBean.

Πιο συγκεκριμένα, κατά την εγγραφή/ είσοδο των μελών στην κοινότητα, η μετακίνηση των δεδομένων από τη φόρμα στον CS μπορεί να γίνεται μέσω JavaBeans. Ακόμα, όλα τα απαραίτητα δεδομένα που θα συλλέγονται από τους clients των SPs μπορούν να πηγαίνουν σε μορφή JavaBeans στον CS.

4.7- Η ΤΕΧΝΟΛΟΓΙΑ PHP

Η *PHP*, (*Hypertext PreProcessor*), είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting). Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα δημιουργίας δυναμικών παραγόμενων ιστοσελίδων. Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον server.

Στο πιο βασικό επίπεδο, η PHP μπορεί να κάνει ό,τι και τα άλλα προγράμματα της τεχνολογίας δυναμικών ιστοσελίδων, όπως επεξεργασία των δεδομένων μιας φόρμας, δημιουργία δυναμικού περιεχομένου ιστοσελίδων ή αποστολή και λήψη cookies. Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Η PHP παρέχει επίσης υποστήριξη για συνομιλία μ' άλλες υπηρεσίες, χρησιμοποιώντας πρωτόκολλα όπως τα IMAP, SNMP, NNTP, POP3 ή και το HTTP.

Η PHP υποστηρίζει τα HTTP cookies όπως ορίζεται από τις προδιαγραφές της Netscape. Τα cookies είναι ένας μηχανισμός για να αποθηκεύονται δεδομένα στον απομακρυσμένο browser και έτσι να είναι δυνατή η παρακολούθηση ή να αναγνώριση των χρηστών. Ο ορισμός των cookies γίνεται με τη συνάρτηση *SetCookie()*.

Τα cookies αποτελούν μέρος του HTTP header, έτσι η συνάρτηση *SetCookie()* πρέπει να κληθεί πριν σταλεί κάποια έξοδος στον browser. Αυτός είναι ο ίδιος περιορισμός που ισχύει και για τη συνάρτηση *Header()*. Τα cookies που στέλνονται στον server από τον client μετατρέπονται αυτόματα σε μια μεταβλητή της PHP όπως συμβαίνει με τα δεδομένα των μεθόδων GET και POST.

4.7.1- Επικύρωση (Authentication) του HTTP με την PHP

Η επικύρωση (authentication) του HTTP στην PHP είναι διαθέσιμη μόνο όταν εκτελείται σαν ένα Apache module και δεν είναι συνεπώς διαθέσιμη στο CGI. Σ' ένα script της PHP σ' ένα Apache module, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση Header() για να στείλουμε ένα μήνυμα "Authentication Required" στον browser του χρήστη (πελάτη) για να εμφανίσει (pop up) ένα παράθυρο εισόδου Username/Password.

Αφού ο χρήστης έχει καταχωρήσει ένα username και ένα password, θα κληθεί ξανά το URL που περιέχει το PHP script με τις μεταβλητές `$PHP_AUTH_USER`, `$PHP_AUTH_PW` και `$PHP_AUTH_TYPE` που περιέχουν αντίστοιχα το user name, το password και τον τύπο επικύρωσης (authentication type).

4.7.2- Συμφόρηση δικτύου και αποδοτικότητα

Η ισχύς της PHP βρίσκεται στην τεράστια βιβλιοθήκη της από ενσωματωμένες συναρτήσεις που περιέχει, οι οποίες δίνουν τη δυνατότητα υλοποίησης πολύ πολύπλοκων εργασιών χωρίς να χρειάζεται εγκατάσταση νέων βιβλιοθηκών όπως συμβαίνει με άλλες δημοφιλείς server-side γλώσσες όπως είναι η Perl.

Η PHP, ενώ είναι γενικά γρήγορη και αποδοτική, παρ' όλα αυτά επιβαρύνει τον χρόνο φόρτωσης και το φορτίο εργασίας του μηχανήματος πάνω στο οποίο τρέχει ο server. Σε sites με μεγάλη κυκλοφορία, αυτό το φορτίο μπορεί να αυξηθεί σε μη αποδεκτά επίπεδα. Χρησιμοποιώντας την PHP στον παρασκήνιο και κάνοντας χρήση των *ημιδυναμικών (semi-dynamic)* σελίδων δίνεται λύση στο πρόβλημα της συμφόρησης δικτύου και δεν φορτώνεται ο server τόσο πολύ. Τέλος, ένα εξαιρετικά ισχυρό χαρακτηριστικό της PHP είναι η ικανότητά της να στέλνει εύκολα μηνύματα email με δυναμικά παραγόμενο περιεχόμενο.

4.7.3- Ασφάλεια

Τα scripts της PHP περιέχουν μερικές φορές ευαίσθητες πληροφορίες όπως usernames, passwords. Ενώ υπάρχει η δυνατότητα ρύθμισης της MySQL έτσι ώστε το username και το password που χρησιμοποιούνται από την PHP να μην μπορούν να χρησιμοποιηθούν από ενδεχόμενους τρίτους, ορίζοντας το πεδίο Host στον πίνακα user, καταφέρνουμε το username και το password να προστατεύονται από ένα αυξημένο επίπεδο ασφάλειας.

Σε περίπτωση που η PHP σταματήσει να εργάζεται στον server, τότε οι σελίδες της PHP θα αντιμετωπισθούν σαν απλά αρχεία κειμένου (text files) μαζί μ' όλο τον κώδικα της PHP, καθώς και το password, και θα είναι ορατές σ' όλους.

Προστασία από αυτήν την παράβαση ασφάλειας θα υπάρχει εάν όλος ο κώδικας (ο σχετικός με την ασφάλεια) τοποθετηθεί σ' ένα αρχείο include και αυτό το αρχείο με την σειρά του σ' ένα directory που δεν αποτελεί

μέρος της δομής καταλόγων του Web server. Προσθέτοντας αυτό το directory στη ρύθμιση include_path της PHP (στο php.ini), θα είναι δυνατόν να γίνεται απευθείας αναφορά στα αρχεία με τη συνάρτηση include() της PHP.

4.7.4- Εφαρμογή της PHP στην προτεινόμενη πλατφόρμα

Οι φόρμες μπορούν να υλοποιηθούν με την PHP τεχνολογία καθώς και η διαδικασία επικύρωσης. Ακόμα, η ευρεία υποστήριξη τόσο διαφορετικών Βάσεων Δεδομένων όσο και ευρείας ποικιλίας πρωτοκόλλων, καθιστούν την PHP τεχνολογία κατάλληλη για την δικτυακή επικοινωνία SR<->CS καθώς και SR<->AS. Τέλος, η ασφάλεια που παρέχει, οι λύσεις σε θέματα συμφόρησης δικτύου και η αποδοτικότητα καθιστούν την PHP μια πολύ καλή επιλογή στο server-side scripting κομμάτι.

4.8- Η ΤΕΧΝΟΛΟΓΙΑ ASP.NET

ASP.NET (Active Server Page).NET είναι μια μεταγλωττισμένη .NET προγραμματιστική πλατφόρμα και τεχνολογία που μπορεί να χρησιμοποιηθεί στον server για να δημιουργηθούν δικτυακές εφαρμογές με ιδιαίτερο βάρος στις βάσεις δεδομένων. Η ASP.NET επωφελείται από το .NET Framework, το οποίο παρέχει εκατοντάδες κλάσεις που χειρίζονται XML, κείμενο ως είσοδο, έλεγχο των εισαγομένων από τον χρήστη, επεξεργασία εικόνας κ.α. Η .NET στρατηγική είναι ανεξάρτητη από συγκεκριμένη προγραμματιστική γλώσσα καθώς είναι συμβατή με πάρα πολλές προγραμματιστικές γλώσσες (25 στο σύνολο συμπεριλαμβανομένων των Visual Basic.NET, C# και Jscript.NET).

Οι διαδικτυακές υπηρεσίες (Web Services), που αποτέλεσαν σημαντικό παράγοντα στη δημιουργία του .NET, επεκτείνει την ιδέα επαναχρησιμοποίησης λογισμικού διαμέσου του Internet, επιτρέποντας ουσιαστικά την πρόσβαση σε λογισμικό εγκατεστημένο σε απομακρυσμένο υπολογιστή ή πλατφόρμα. Έτσι κάθε εφαρμογή αποτελείται από ανεξάρτητα κομμάτια τα οποία και συγγράφονται από ανεξάρτητους προγραμματιστές.

Η ASP.NET, είναι μια τεχνολογία για την αποστολή στον client δυναμικό διαδικτυακό περιεχόμενο (dynamic Web content), συμπεριλαμβανομένου XHTML, Dynamic HTML, client-side scripts όπως και Java Applets.

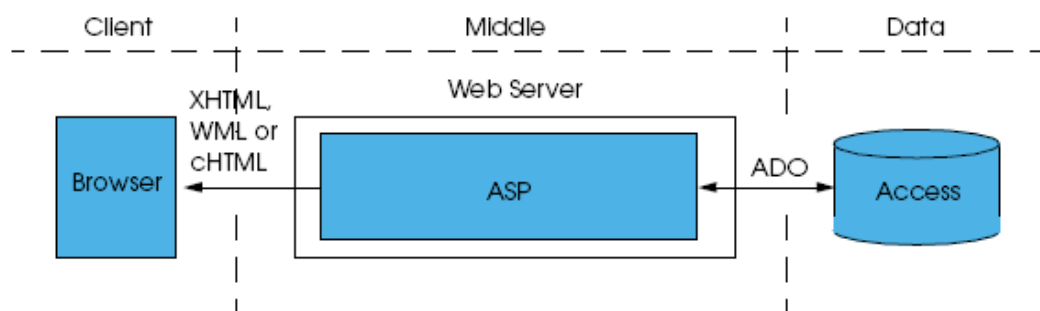
Τα αρχεία σε ASP.NET πλατφόρμα αναφέρονται συνήθως και ως δικτυακές φόρμες (Web Forms) καθώς επεξεργάζονται εισαγόμενα από φόρμα. Δεδομένα που εισάγονται σε μία φόρμα μπορούν να αποσταλούν στον server, να επεξεργαστούν και τέλος να επιστραφούν στον client σε διαφορετικό format. *Για παράδειγμα είναι ιδανικό στον έλεγχο στοιχείων ώστε ένα μέλος(είτε SR είτε SP) να εξακριβώσει την ταυτότητά του και αν εισέρθει στην κοινότητα.*

Μια δικτυακή φόρμα (Web Form) περιλαμβάνει όλα τα απαραίτητα GUI στοιχεία ώστε ο χρήστης να πλοηγείται άνετα και να εισάγει τα

δεδομένα που απαιτούνται με ευκολία και χωρίς προαπαιτούμενη τεχνολογική εμπειρία.

Το διαδίκτυο (Web) λειτουργεί πάνω στο HTTP το οποίο και είναι ένα στατικό πρωτόκολλο υπό την έννοια ότι δεν διατηρεί την σύνδεση ώστε ο Web Server να γνωρίζει την κατάσταση του client. Αυτό σημαίνει ότι οι Web Servers δεν έχουν δυνατότητα να προσδιορίσουν εάν το request προέρχεται από έναν συγκεκριμένο client ή εάν μια σειρά από requests προέρχονται από τον ίδιο χρήστη. Υπάρχουν διάφοροι τρόποι με τους οποίους προσδιορίζει κανείς την ταυτότητα ενός χρήστη: με τη βοήθεια των Session ID και των Cookies.

Η τεχνολογία ADO παρέχει πρόσβαση στις βάσεις δεδομένων μιας ιστοσελίδας.



Σχήμα 31. Τεχνολογία ASP/ ADO

4.8.1- Η .NET πλατφόρμα

Γενικά για την .NET πλατφόρμα:

- Ενσωμάτωση τεχνολογιών Internet – Δίνει τη δυνατότητα ανάπτυξης λύσεων οι οποίες αξιοποιούν σε μέγιστο βαθμό την «οικουμενικότητα» του Internet και τις δυνατότητες επικοινωνίας κατ' απαίτηση.
- Ταχύτητα στην αγορά (time to market) – Οι λύσεις μπορούν να αναπτυχθούν και να τεθούν σε επιχειρησιακή λειτουργία γρήγορα και χωρίς να απαιτείται επανεκπαίδευση των developers. Οι διατιθέμενες υπηρεσίες και η λειτουργικότητα που αντιπροσωπεύουν ελαχιστοποιούν τον απαιτούμενο κώδικα των εφαρμογών.
- Αξιοπιστία & υψηλή απόδοση – Παρέχει κλιμάκωση, αξιοπιστία και ευκολία στη διαχείριση για περιβάλλοντα hosting και λύσεις e-commerce και line-of-business.
- Αληθινή διαλειτουργικότητα (interoperability) – Υποστήριξη «ανοιχτών» standards και πρωτοκόλλων όπως η XML και το Single Object Access Protocol (SOAP), έτσι ώστε να καθίσταται δυνατή η ολοκλήρωση με συστήματα άλλων κατασκευαστών.
- Μειωμένη πολυπλοκότητα – Ενοποίηση των βασικών υπηρεσιών κατευθείαν με το λειτουργικό σύστημα και διάθεση αυτών με ενοποιημένο τρόπο μέσα από υπηρεσίες Web. Ελαχιστοποίηση της απαιτούμενης προσπάθειας των τεχνικών να λειτουργούν ως «system integrators», έτσι

ώστε να μπορούν να επικεντρωθούν στην επίλυση επιχειρηματικών προβλημάτων.

- Ανεξαρτησία γλώσσας προγραμματισμού, εργαλείων και hardware – Παροχή ενός μοντέλου υπηρεσιών Web τελείως ανεξάρτητων από τη γλώσσα υλοποίησης, έτσι ώστε να μπορούν οι developers να χρησιμοποιούν «το κατάλληλο εργαλείο για την κατάλληλη δουλειά».

4.8.2- Πρόσβαση σε μια Βάση Δεδομένων από μια σελίδα ASP

Αρχικά θα πρέπει να δημιουργηθεί μια σύνδεση βάσης δεδομένων (database connection). Ακολουθώς άνοιγμα της σύνδεσης της βάσης δεδομένων (database connection). Δημιουργία ενός ADO recordset και επακόλουθο άνοιγμα αυτού. Εξαγωγή των χρήσιμων δεδομένων από το recordset. Κλείσιμο του recordset και της σύνδεσης.

Το αντικείμενο ADO Connection χρησιμοποιείται για να καθιερωθεί μια σύνδεση βάσης δεδομένων.

Οι πιο κοινές Μέθοδοι και Ιδιότητες :

Μέθοδοι

- *Close*, κλείνει μια σύνδεση.
- *Execute*, εκτελεί ένα ερώτημα (query), εντολή ή διαδικασία.
- *Open*, ανοίγει μια σύνδεση.

Ιδιότητες

- *Mode*, ορίζει ή επιστρέφει την άδεια πρόσβασης του provider.
- *Provider*, ορίζει ή επιστρέφει το όνομα του provider.
- *State*, επιστρέφει μια τιμή που περιγράφει αν η σύνδεση είναι ανοικτή ή κλειστή.

Για να είναι δυνατή η ανάγνωση των δεδομένων μιας βάσης δεδομένων, πρέπει αυτά πρώτα να φορτωθούν σ' ένα recordset. Γενικά το αντικείμενο Recordset (Recordset Object) χρησιμοποιείται για να περιέχει ένα σύνολο εγγραφών από έναν πίνακα μιας βάσης δεδομένων.

Οι πιο κοινές Μέθοδοι και Ιδιότητες :

Μέθοδοι (Methods)

- *AddNew*, δημιουργεί μια νέα εγγραφή.
- *Close*, κλείνει ένα Recordset.
- *Delete*, διαγράφει την τρέχουσα εγγραφή ή ομάδα εγγραφών.
- *MoveFirst*, πηγαίνει στην πρώτη εγγραφή.
- *MoveLast*, πηγαίνει στην τελευταία εγγραφή.
- *MoveNext*, πηγαίνει στην επόμενη εγγραφή.
- *MovePrevious*, πηγαίνει στην προηγούμενη εγγραφή.
- *Open*, ανοίγει ένα Recordset.

- *Update*, αποθηκεύει όλες τις αλλαγές.

Ιδιότητες (Properties)

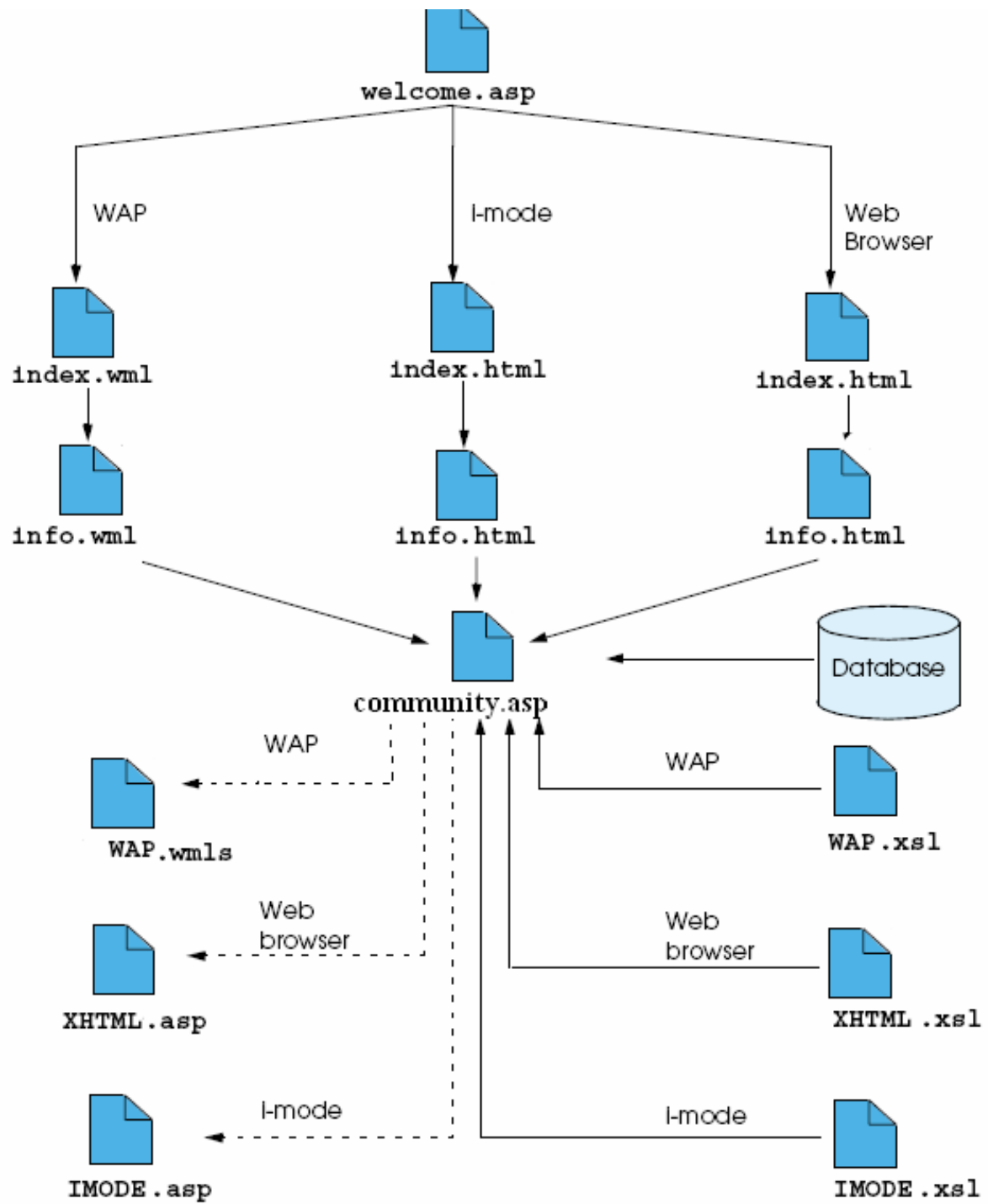
- *BOF*, επιστρέφει true αν η τρέχουσα εγγραφή βρίσκεται πριν από την πρώτη εγγραφή, αλλιώς επιστρέφει false.
- *EOF*, επιστρέφει true αν η τρέχουσα εγγραφή βρίσκεται μετά από την τελευταία εγγραφή, αλλιώς επιστρέφει false.
- *Fields*, περιέχει όλα τα αντικείμενα πεδίου (field objects) του Recordset Object.
- *RecordCount*, επιστρέφει το πλήθος των εγγραφών που υπάρχουν σ' ένα Recordset Object.
- *Sort*, καθορίζει μια λίστα χωρισμένων με κόμμα ονομάτων πεδίων με βάση τα οποία θα ταξινομηθεί το Recordset.

Ο πιο κοινός τρόπος για να εμφανίσουμε τα δεδομένα ενός recordset, είναι να τα εμφανίσουμε σ' έναν html πίνακα. Τέλος, με τη βοήθεια της SQL, τα δεδομένα που πρέπει να εμφανισθούν σε μια HTML σελίδα μπορούν να φιλτραριστούν και να ταξινομηθούν.

4.8.3- Εφαρμογή της ASP στην προτεινόμενη πλατφόρμα

Με τη χρήση της ASP μπορούμε να τροποποιήσουμε δυναμικά το περιεχόμενο της ιστοθέσης που θα επισκέπτεται ο SR για την αναζήτηση της επιθυμητής του υπηρεσίας. Ακόμα, η ASP τεχνολογία θα απαντά σε ερωτήσεις χρηστών ή σε δεδομένα που υποβάλλονται από φόρμες HTML. Με τη χρήση της ASP θα γίνεται η πρόσβαση σε δεδομένα ή τις βάσεις δεδομένων του CS και να επιστρέφει τα αποτελέσματα. Ακόμα θα προσαρμόζει την ιστοσελίδα ώστε να την κάνει πιο χρήσιμη στους χρήστες. Τα πλεονεκτήματα από την χρήση της ASP αντί των ανταγωνιστικών τεχνολογιών CGI και Perl είναι η απλότητα και η ταχύτητα. Στο πολύ ευαίσθητο θέμα της ασφάλειας, η ASP αποδεικνύεται πολύ καλή καθώς ο ASP κώδικας δεν μπορεί να ειδωθεί από τον εκάστοτε browser. Εφόσον τα αρχεία ASP επιστρέφονται σαν απλή HTML, μπορούν να ειδωθούν από οποιονδήποτε browser. Τέλος στο θέμα της συμφόρησης, ο έξυπνος ASP προγραμματισμός μπορεί να ελαχιστοποιήσει την κυκλοφορία στο δίκτυο (network traffic).

Η ADO είναι ένα προγραμματιστικό interface για να έχουμε πρόσβαση στα δεδομένα μιας βάσης δεδομένων. Ο CS της κοινότητας διατηρεί τις ακόλουθες βάσεις: members' DB, Application Services' DB, Service Providers' DB. Η επικοινωνία με αυτές θα είναι συχνή, με σκοπό την άντληση/ καταγραφή/ επεξεργασία δεδομένων. Η πρόσβαση στην ADO πραγματοποιείται μέσα από την τεχνολογία ASP (Active Server Pages).



Σχήμα 32. Παράδειγμα υλοποίησης web σελίδας της κοινότητας με τεχνολογία ASP

5- ΕΠΙΛΟΓΟΣ

5.1- ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η πλατφόρμα που περιγράφει αυτή η εργασία δεν αποτελεί σε καμία περίπτωση ολοκληρωμένο προϊόν το οποίο να μπορεί να διοχετευθεί στην αγορά. Αποτελεί περισσότερο μια διερεύνηση του τρόπου εφαρμογής των τεχνολογιών Service Oriented Architecture και Web Services σε κοινότητες χρηστών που χρησιμοποιούν δίκτυα κινητών τηλεπικοινωνιών, πληρώντας ταυτόχρονα τις βασικές προϋποθέσεις που ένα ολοκληρωμένο σύστημα πρέπει να διαθέτει.

Μια επέκταση που το παρόν σύστημα επιδέχεται είναι η δυνατότητα παροχής του απαραίτητου λογισμικού, στα μέλη που θέλουν να είναι Service Providers, από μέρους του Community Server. Η εργασία αναφέρει ότι για να μπορέσει ένα μέλος της κοινότητας να είναι Service Provider πρέπει να εγκαταστήσει την απαραίτητη εφαρμογή στο κινητό τερματικό που χρησιμοποιεί. Θα προσέφερε μεγάλη ευελιξία στα μέλη που το επιθυμούν, από τον ιστοχώρο της κοινότητας να μπορούν να εγκαταστήσουν αυτόματα και over the air την απαραίτητη εφαρμογή η οποία θα τους δίνει τη δυνατότητα να είναι Service Providers της εφαρμογής υπηρεσίας που επιλέγουν. Έτσι θα μπορούσε ένας πάροχος υπηρεσίας να ανανεώνει την εφαρμογή που έχει εγκαταστήσει σύμφωνα με τις βελτιώσεις που θα γίνονται συνεχώς στις παρεχόμενες υπηρεσίες. Ακόμα θα έδινε τη δυνατότητα στον Community Server να επικοινωνεί με τις εφαρμογές αυτές και να διαχειρίζεται διάφορα προβλήματα και τις απαραίτητες ρυθμίσεις που πιθανόν να χρειάζεται η κάθε εφαρμογή. Η δυνατότητα διαχείρισης των τερματικών συσκευών παρέχεται μέσω της τεχνολογίας του WAP Provisioning και είναι υλοποιήσιμη στη προτεινόμενη πλατφόρμα αφού υποστηρίζει τα πρωτόκολλα WAP. Απαραίτητη φυσικά είναι η συνεργασία των διαχειριστών του Community Server με τους αντίστοιχους του Application Service καθώς και μιας ομάδας ανάπτυξης λογισμικού ώστε να μπορεί να υλοποιηθεί η δυνατότητα αυτή.

Πιθανή επέκταση της προτεινόμενης πλατφόρμας είναι επίσης και η ανάπτυξη εφαρμογών υπηρεσιών η οποίες θα λαμβάνουν την απαραίτητη πληροφορία που χρειάζονται, όχι από διαθέσιμους service providers αλλά από άλλες εφαρμογές υπηρεσίες που είτε αποτελούν υφιστάμενες εφαρμογές υπηρεσίες είτε είναι απλές υπηρεσίες που παρέχονται από άλλους φορείς. Έτσι θα προσφέρονται στο χρήστη ολοκληρωμένες υπηρεσίες οι οποίες θα αποτελούνται από διαφορετικές υπηρεσίες οι οποίες είναι διαθέσιμες ξεχωριστά η μία από την άλλη. Τη δυνατότητα αυτή προσφέρει η Business Process Execution Language (BPEL) την οποία θα πρέπει να υποστηρίζει ο Community Server και καθορίζει τη σειρά εκτέλεσης των λειτουργιών σε μια συλλογή από υπηρεσίες, τα δεδομένα που ανταλλάσσονται μεταξύ τους, ποιοι συνέταιροι παίρνουν μέρος και πώς εμπλέκονται μέσα στη διαδικασία.

5.2- ΥΠΟΣΤΗΡΙΖΟΜΕΝΕΣ ΥΠΗΡΕΣΙΕΣ ΒΑΣΕΙ ΤΩΝ ΠΙΘΑΝΩΝ ΕΠΕΚΤΑΣΕΩΝ

5.2.1- Υπηρεσία εύρεσης διαδρομής

Μέλος της κοινότητας διατείνεται ότι δοθέντος ενός γεωγραφικού στίγματος αφετηρίας και ενός στίγματος προορισμού, δύναται μέσω της υπηρεσίας του να υπολογίσει τη συντομότερη διαδρομή η οποία μάλιστα αποτυπώνεται στο επιστρεφόμενο αρχείο εικόνας. Έτερο μέλος της κοινότητας, αναζητά υπηρεσία που να σχετίζεται με οδικό χάρτη ώστε από το σημείο Α στο οποίο βρίσκεται, να μεταβεί όσο το δυνατόν γρηγορότερα στο σημείο Β. Η διαδικασία αναζήτησης τον οδηγεί να επιλέξει την προηγούμενη υπηρεσία. Αρκεί να δηλώσει την περιοχή στην οποία θέλει να φτάσει καθώς η πληροφορία του τρέχοντος στίγματος του (αφετηρία) αντλείται αυτόματα. Ο διακομιστής ενημερώνεται σχετικά με την αίτηση χρήσης μιας εκ των τρεχουσών υπηρεσιών και μαζί με τα δεδομένα εισόδου γίνεται άμεση προώθηση στο μέλος που εκδίδει την υπηρεσία. Ο πάροχος της προκείμενης υπηρεσίας έχει στα χέρια του λοιπόν δύο γεωγραφικά σημεία και καλείται να αποτυπώσει σε ένα χάρτη τη συντομότερη διαδρομή μεταξύ τους. Ο χάρτης όμως επί του οποίου εφαρμόζει τον δικό του ευρετικό αλγόριθμο συντομότερης διαδρομής, είναι προϊόν μιας άλλης υπηρεσίας που προωθεί κομμάτια-τμήματα του χάρτη γύρω από κάποια δοθέντα γεωγραφικά σημεία. Χρησιμοποιώντας λοιπόν ως είσοδο την είσοδο που έλαβε από την τρέχουσα υπηρεσία και στην οποία τελεί ως πάροχος, κινεί διαδικασίες αναζήτησης και χρήσης άλλης υπηρεσίας στην οποία είναι αποδέκτης αυτή τη φορά. Επί του τμήματος χάρτη που λαμβάνει, εφαρμόζει τον ευρετικό αλγόριθμο και δυο κουκίδες συνδεδεμένες με μια γραμμή, εκείνη της συντομότερης διαδρομής, αποτυπώνονται στο χάρτη. Ως αρχείο εικόνας ο τελευταίος, ακολουθεί την πορεία του προς τον διακομιστή για να καταλήξει τελικά στον αρχικό αιτούντα.

5.2.2- Υπηρεσία διαμεσολαβητικής πληρωμής (Proxy Pay)

Υπηρεσία διαμεσολαβητικής πληρωμής (Proxy pay). Μπορούμε να εκμεταλλευτούμε την ευχέρεια που μας παρέχει το κοινοτικό μοντέλο και σε επίπεδο αγοροπωλησιών. Η ιδέα είναι πως κάθε μέλος που επιθυμεί να περατώσει μια συναλλαγή μέσα στα πλαίσια της κοινότητας, μπορεί να το πράξει χωρίς να χρειάζεται να παραθέτει λ.χ. τον αριθμό της πιστωτικής του κάρτας κάθε φορά. Το μέσο πληρωμής (λ.χ. internet banking, αριθμός πιστωτικής κάρτας) παρατίθεται με ασφαλή τρόπο μονάχα μια φορά, αποφεύγοντας έτσι την διακίνηση ευαίσθητων δεδομένων, διαδικασία που ακολουθείται σε κάθε δικτυακή συναλλαγή. Η αντίστοιχη υπομονάδα του CS ενεργοποιείται και περατώνει τη συναλλαγή κάθε φορά που το μέλος εκδηλώνει την επιθυμία να προβεί σε μία συγκεκριμένη αγοραπωλησία. Υπάρχουν βέβαια και άλλες εναλλακτικές που στηρίζονται στο P2P μοντέλο, στο B2C/ B2B μοντέλο και τέλος στο eCash. Η συγκεκριμένη τακτική κρύβει και οικονομικά οφέλη για τον operator, ο οποίος μπορεί να συνάψει διακανονισμούς με το έτερον άκρο της συναλλαγής.

- 1^η εκδοχή: Πραγματοποίηση αγορών σε σουπερ-μάρκετ. Μέλος της κοινότητας επιθυμεί να κάνει τα καθιερωμένα ψώνια του από το σουπερ-μάρκετ. Η έλλειψη όμως μεταφορικού μέσου τον κάνει να αναζητήσει αντίστοιχη ή σχετική υπηρεσία στην κοινότητά του. Την λύση έρχεται να δώσει η διαθέσιμη υπηρεσία «Σούπερ-Μάρκετ». Όπως πληροφορείται από τη σύντομη περιγραφή, πρόκειται για μια υπηρεσία στην οποία τελούν ως παροχείς όλες οι γνωστές αλυσίδες σουπερ-μάρκετ της επικράτειας. Το μέλος σκέφτεται να δοκιμάσει τη συγκεκριμένη υπηρεσία. Θα είναι και η πρώτη φορά που θα χρησιμοποιήσει τη διαμεσολαβιτική πληρωμή, μια δυνατότητα που του προσφέρει η κοινότητα (ήδη το μέλος, έχοντας προβλέψει το πιθανό ενδεχόμενο να πραγματοποιήσει δικτυακές/ κοινοτικές αγορές, κατά τη διαδικασία εγγραφής του στην κοινότητα, είχε κοινοποιήσει με ασφαλή τρόπο και τον αριθμό της πιστωτικής του κάρτας). Μέσω της φιλικής γραφικής διεπαφής χρήστη, επιλέγει την προτιμητέα του αλυσίδα σουπερ-μάρκετ. Η αυτόματη εξόρυξη της θέσης του θα επιτρέψει την πραγματοποίηση της διαταγής του από το κοντινότερο διαθέσιμο κατάστημα της αλυσίδας. Η πλοήγηση στα διαθέσιμα προς αγορά αγαθά είναι εύκολη. Η αναζήτηση προϊόντων μπορεί να γίνει είτε κατά όνομα είτε κατά κατηγορία. Απαραίτητη είναι και η επιθυμητή ποσότητα κάθε προϊόντος. Η ολοκλήρωση των αγορών φέρνει το μέλος μπροστά σε μια συνολική παρουσίαση των προϊόντων, των ποσοτήτων τους και της συνολικής (μέχρι εκείνη την ώρα) τιμής. Το μέλος βεβαιώνεται πως αγόρασε όλα τα απαραίτητα, και προχωρά στην αγορά. Το πάτημα του κουμπιού «Αγορά», μεταφράζεται ως χρέωση του αριθμού πιστωτικής κάρτας κατά το συνολικό ποσό με αποδέκτη το συγκεκριμένο κατάστημα αγορών. Η μεταφορά των προϊόντων στο χώρο του χρήστη είναι δωρεάν, αφού έχει ξεπεράσει ένα χρηματικό όριο πλαφόν, και πραγματοποιείται εντός 24ώρων ωρών. Όπως βλέπουμε το μέλος δεν χρειάζεται να παραχωρήσει ούτε τον αριθμό της πιστωτικής του κάρτας (το έχει πράξει εξάλλου μια και μοναδική φορά), ούτε την περιοχή κατοικίας του (είτε εξορύσσεται ως γεωγραφικό στίγμα είτε έχει παράσχει αυτήν την πληροφορία κατά την εγγραφή του στην κοινότητα).
- 2^η εκδοχή: Αυτόματη πληρωμή διοδίων. Μέλος της κοινότητας μετακινείται πολύ με το αυτοκίνητό του, και χρειάζεται σχετικά συχνά να χρησιμοποιεί οδικές αρτηρίες που απαιτούν διόδια, αλλά σε συχνότητα τέτοια που η αγορά μηνιαίας κάρτας δεν είναι η πλέον οικονομική λύση. Αναζητώντας τις νέες διαθέσιμες υπηρεσίες από την κοινότητα βλέπει μια νέα είσοδο με όνομα «Αυτόματη Πληρωμή Διοδίων». Μέσω της παρεχόμενης περιγραφής πληροφορείται πως ο operator του, έχει έρθει σε διακανονισμό με το υπουργείο μεταφορών (το οποίο μέσω του παραρτήματος των διοδίων, τελεί ως πάροχος υπηρεσίας), προσφέροντας τη δυνατότητα αυτόματης πληρωμής του αντιτίμου των διοδίων κάθε φορά που πλησιάζει κανείς στην είσοδο αυτών. Απαραίτητες προϋποθέσεις: α) να είναι κανείς μέλος της κοινότητας (σε αντίθετη περίπτωση εξάλλου δεν θα έβλεπε καν τη διαθέσιμη υπηρεσία), β) το κινητό τερματικό να υποστηρίζει σύστημα εντοπισμού της θέσης από δορυφόρο (λ.χ. Galileo, GPS, πιθανοτικός προσδιορισμός από την ένταση λαμβανόμενου σήματος από τους σταθμούς βάσης) και γ)

συμμετοχή στην υπηρεσία διαμεσολαβητικής πληρωμής. Το μέλος ενθουσιάζεται και αποφασίζει να δοκιμάσει την υπηρεσία. Η πρότερη εμπειρία του από τη χρήση της διαμεσολαβητικής πληρωμής, συμβάλλει στην απόφασή του αυτή. Πλέον, κάθε φορά που θα πλησιάζει σε είσοδο διοδίων, θα γίνεται αντιληπτός μέσω του συστήματος εντοπισμού, θα επιβεβαιώνεται η συμμετοχή του στην υπηρεσία και θα πραγματοποιείται αυτόματα η πληρωμή του σχετικού αντιτίμου από τον συνδεδεμένο αριθμό πιστωτικής κάρτας. Στην συγκεκριμένη περίπτωση ο αιτών της υπηρεσίας που πληρεί τις προϋποθέσεις και δύναται να συμμετάσχει στην υπηρεσίας, μπορεί να προέρχεται και έξω από την ίδια την κοινότητα. Η προτεινόμενη πλατφόρμα επιτρέπει κάτι τέτοιο, αλλά τον κύριο λόγο τον έχει ο operator και το επιχειρηματικό μοντέλο που υιοθετεί.

5.2.3- Mobile Multiplayer Gaming

Το mobile gaming είναι ιδιαίτερα διαδεδομένο και τυγχάνει ανταπόκρισης τόσο μικρών όσο και μεγαλύτερων ηλικιών. Πλέον όλα τα κινητά τηλέφωνα περιλαμβάνουν κάποια ενσωματωμένα παιχνίδια, ενώ υπάρχει η δυνατότητα για φόρτωμα ακόμη περισσότερων. Στον κόσμο του desktop gaming, το διαδίκτυο έθεσε νέες διαστάσεις και πλέον με τη βοήθειά του μιλάμε για MMORPG (Massive Multiplayer Online Role Playing Games). Πρόκειται για διαδικτυακά παιχνίδια που επιτρέπουν την ταυτόχρονη συμμετοχή πολλών παιχτών, οι οποίοι και αλληλεπιδρούν, προσφέροντας έτσι μοναδικές εμπειρίες και πραγματικό εθισμό.

Το mobile gaming εξαρτάται άμεσα από τις δυνατότητες των κινητών τερματικών. Όσο το hardware θα βελτιώνεται τόσο τα mobile games θα εξελίσσονται. Σε θέματα τηλεπικοινωνιακού διαύλου τώρα, το multiplayer κομμάτι της προτεινόμενης υπηρεσίας θέτει επιπρόσθετες απαιτήσεις. Και εδώ ωστόσο, τα πράγματα διαρκώς επεκτείνονται/ εξελίσσονται προσφέροντας στους τελικούς χρήστες διαρκώς αυξανόμενες ταχύτητες πρόσβασης στον δίαυλο. Η σωστή εκμετάλλευση και υλοποίηση του mobile multiplayer gaming μπορεί να φέρει την επανάσταση, και πιθανόν να αποτελέσει την killer application της γενιάς κινητών τηλεφώνων που θα το φιλοξενήσουν.

Το προτεινόμενο μοντέλο υποστηρίζει το Mobile Multiplayer Gaming. Αρχικά θα πρέπει να δημιουργηθεί μια AS που θα σχετίζεται με ένα συγκεκριμένο παιχνίδι. Θα υπάρχουν τόσες AS όσα και τα διαθέσιμα δικτυακά παιχνίδια. Ακολούθως θα πρέπει η AS να δηλωθεί στον CS ορίζοντας έτσι την διαθεσιμότητά της. Οι χρήστες που επιθυμούν να λάβουν μέρος σε ένα παιχνίδι, αποτελούν τόσο SRs όσο και SPs (και αυτό επειδή επιθυμούν να παίξουν ένα παιχνίδι -SR- και ταυτόχρονα να δηλώσουν τη διαθεσιμότητά τους κατά την αναζήτηση παιχτών πριν από την έναρξη ενός παιχνιδιού -SP-). Ο CS αναλαμβάνει να ενημερώσει τις απαραίτητες βάσεις δεδομένων του για την μελλοντική επικοινωνία με τους χρήστες. Η AS επικοινωνεί με τον CS ώστε να ενημερωθεί σχετικά με την ύπαρξη επαρκούς αριθμού διαθέσιμων παικτών για την εκκίνηση ενός αγώνα.

Θεωρούμε δεδομένο ότι όλοι οι παίκτες έχουν εγκατεστημένο στη συσκευή τους το επιθυμητό παιχνίδι. Έτσι, το παιχνίδι πολλών παικτών πραγματοποιείται μέσω μιας διαρκούς επικοινωνίας SR/SP <->CS<->AS, μέσω του GUI που προσφέρει το εκάστοτε παιχνίδι. Με αυτόν τον τρόπο μπορούν να υποστηριχθούν τόσο turn-based games (π.χ. τάβλι, σκάκι) που είναι λιγότερο απαιτητικά σε ταχύτητα σύνδεσης όσο και real-time games (π.χ. ποδόσφαιρο, αγώνες ταχύτητας κοκ).

5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η συγκεκριμένη εργασία ανέπτυξε τη δυνατότητα χρήσης υπηρεσιών από κινητά τερματικά χρησιμοποιώντας τόσο τις δυνατότητες που προσφέρουν τα δίκτυα κινητών τηλεπικοινωνιών όσο και το διαδίκτυο. Με την προτεινόμενη πλατφόρμα οι υπηρεσίες του διαδικτύου και η service oriented architecture που μέχρι τώρα αναπτύσσονται με βάση τις ανάγκες των επιχειρήσεων και απευθύνονται μόνο σε αυτές τώρα φτάνουν σε όλους τους χρήστες κινητών συσκευών χωρίς να προϋποθέτει συσκευές με υπερβολικές δυνατότητες.

Η ανάπτυξη της πλατφόρμας δεν αφορά ειδικά μια συγκεκριμένη υπηρεσία παρέχοντας έτσι τη δυνατότητα ανάπτυξης πολλών διαφορετικών υπηρεσιών. Επίσης οι πληροφορίες που ανταλλάσσονται δεν είναι στατικές αλλά έχουν δυναμικό χαρακτήρα και επηρεάζονται από την κατάσταση και θέση του παρόχου τους προσφέροντας έτσι ένα έμμεσο τρόπο επικοινωνίας μεταξύ των χρηστών κινητών συσκευών και εξαγωγή αναλόγων συμπερασμάτων. Έτσι οι κινητές συσκευές αποκτούν τη δυνατότητα αναζήτησης και εύρεσης πληροφορίας η οποία προέρχεται από χρήστες κινητών συσκευών και απευθύνεται σε χρήστες κινητών συσκευών, δυνατότητα που προσέφερε στο διαδίκτυο και την ευρεία του αποδοχή

Οι τεχνολογίες που χρησιμοποιήθηκαν ακολουθούν πρότυπα τα οποία έχουν καθιερωθεί στην ανάπτυξη λογισμικού για δικτυακές εφαρμογές καθιστώντας τη πλατφόρμα υλοποιήσιμη και εύκολα διαθέσιμη στο ευρύ κοινό. Επίσης δίνει τη ευελιξία στην πλατφόρμα να συνδυάζει υφιστάμενες υπηρεσίες χωρίς είναι αναγκαία η υλοποίηση των υπηρεσιών αυτών από την αρχή.

ΠΑΡΑΡΤΗΜΑ

Στο σημείο αυτό περιγράφονται περιληπτικά διάφορα εργαλεία και πλατφόρμες ανάπτυξης λογισμικού το οποία αναφέρονται σε διάφορα σημεία της εργασίας.

ASP

Η γλώσσα προγραμματισμού **ASP** (*Active Server Pages*) έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Με την τεχνολογία της ASP δημιουργούνται δυναμικές ιστοσελίδες (dynamic web pages).

Java Platform

Η Java είναι μια πλατφόρμα ανάπτυξης λογισμικού βασισμένη στον αντικεινοστραφή (Object Oriented) προγραμματισμό. Υπάρχουν τρεις διαφορετικές εκδόσεις της πλατφόρμας αναλόγως των συσκευών στις οποίες απευθύνονται οι εφαρμογές. Η Java 2 Standard Edition (J2SE) η οποία απευθύνεται σε προσωπικούς υπολογιστές και εξυπηρετητές, η Java 2 Enterprise Edition (J2EE) η οποία χρησιμοποιείται σε μεγάλης έκτασης τεχνολογικά έργα και η Java 2 Micro Edition (J2ME) η οποία απευθύνεται σε εφαρμογές συσκευών περιορισμένων δυνατοτήτων όπως τα κινητά τηλέφωνα. Πλεονέκτημα της πλατφόρμας αυτής είναι η ανεξαρτησίας της από τα λειτουργικά συστήματα. Αυτό επιτυγχάνεται λόγω του ότι οι εφαρμογές που αναπτύσσονται δεν επικοινωνούν κατευθείαν με το λειτουργικό σύστημα ενός υπολογιστή αλλά μέσω μιας εικονικής μηχανής, της Java Virtual Machine για τις εκδόσεις J2SE και J2EE καθώς και της K Virtual Machine για την έκδοση J2ME.

Axis

Ο Axis είναι μια εργασία του οργανισμού Apache η οποία προσφέρει ένα API για την δημιουργία, επεξεργασία και αποστολή μηνυμάτων SOAP. Επίσης προσφέρει δυνατότητες ανάπτυξης ολοκληρωμένων υπηρεσιών Web Services καθώς και διάφορα εργαλεία ελέγχου και δοκιμής των υπηρεσιών αυτών. Τα εργαλεία αυτά δεν περιορίζονται μόνο στο πρωτόκολλο SOAP αλλά και σε όλα τα πρωτόκολλα που χρειάζεται να υλοποιήσει κάποιος για να αναπτύξει Web Services όπως το WSDL.

Xerces

Ο Xerces είναι μια εργασία του οργανισμού Apache η οποία προσφέρει APIs για την ανάλυση, δημιουργία και επικύρωση κειμένων XML. Υποστηρίζει όλα τα διεθνή πρότυπα που δημιουργούνται και αφορούν την XML όπως τα W3C's XML Schema recommendation version 1.0, το DOM Level 2 version 1.0, καθώς και το SAX Version 2.

Xalan

Ο Xalan είναι μια εργασία του οργανισμού Apache η οποία προσφέρει XSLT επεξεργασία για το μετασχηματισμό εγγράφων XML σε HTML, XML σε κείμενο, ή άλλους τύπους εγγράφων XML. Ο Xalan χρησιμοποιεί τον Xerces για ανάλυση XML και μπορεί να διαμορφωθεί για να χρησιμοποιήσει άλλους αναλυτές XML. Αυτή τη στιγμή αντιπροσωπεύει τις πλήρεις εφαρμογές των W3C συστάσεων για τους μετασχηματισμούς XSL (XSLT) και τη γλώσσα πορειών XML (XPath). Η δεύτερη έκδοση είναι αυτήν την περίοδο υπό ανάπτυξη.

MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάση δεδομένων βασισμένο σε SQL (Structured Query Language). Μπορεί να λειτουργήσει σε όλα τα λειτουργικά συστήματα και διαθέτει API για πολλές γλώσσες προγραμματισμού. Επίσης υποστηρίζει ταυτόχρονη χρήση από πολλούς διαφορετικούς χρήστες χρησιμοποιώντας παράλληλες διεργασίες.

Apache HTTP Server

Ο Apache HTTP Server είναι μια εργασία του οργανισμού Apache και λειτουργεί σαν εξυπηρετητής διαδικτύου. Όπως όλα τα προϊόντα της Apache είναι ελεύθερα διαθέσιμος. Υποστηρίζει ασφαλείς συνδέσεις (HTTPS) και είναι ο ποιο διαδεδομένος εξυπηρετητής στο διαδίκτυο.

Tomcat

Ο Tomcat είναι μια εργασία του οργανισμού Apache και λειτουργεί σαν εξυπηρετητής εφαρμογών. Παρέχει την δυνατότητα απομακρυσμένης εκτέλεσης εφαρμογών Java (servlets) καθώς και σελίδων που περιέχουν κώδικα Java (JSP). Μπορεί να λειτουργήσει αυτόνομα αφού περιέχει ένα Web server ή σε συνεργασία με τον Apache HTTP Server και άλλους εξυπηρετητές διαδικτύου.

Web Service

Μια δικτυακή υπηρεσία (Web Service) είναι μια κλάση (class) αποθηκευμένη σε ένα μηχάνημα το οποίο μπορεί να προσπελαστεί από ένα άλλο μηχάνημα διαμέσου ενός δικτύου. Εξαιτίας αυτής της σχέσης, το μηχάνημα στο οποίο είναι αποθηκευμένη η δικτυακή υπηρεσία ονομάζεται απομακρυσμένο μηχάνημα (remote machine). Η εφαρμογή που επιθυμεί πρόσβαση στην δικτυακή υπηρεσία, στέλνει μια μέθοδο κλήσης και τα arguments της στο απομακρυσμένο μηχάνημα, το οποίο με τη σειρά του επεξεργάζεται την κλήση και στέλνει μια απάντηση στον αρχικό καλούντα. Αυτό το είδος κατανεμημένου συστήματος ευνοεί διάφορα συστήματα όπως τερματικά με αργή επεξεργαστική ισχύ, περιορισμένη μνήμη ή διαθέσιμους πόρους, τερματικά που δεν έχουν άμεση πρόσβαση σε συγκεκριμένα

δεδομένα ή δεν μπορούν να πραγματοποιήσουν συγκεκριμένους υπολογισμούς. Ένα άλλο πλεονέκτημα των δικτυακών υπηρεσιών είναι ότι ο κώδικας και τα δεδομένα μπορούν να αποθηκεύονται σε κάποιον άλλο υπολογιστή. Για παράδειγμα, μια δικτυακή υπηρεσία μπορεί να δηλώνεται σε μια συγκεκριμένη τοποθεσία ώστε να πραγματοποιεί διάφορες συνήθεις ερωτήσεις σε βάσεις δεδομένων. Η βάση δεδομένων μάλιστα, περιλαμβάνεται στο μηχάνημα που φιλοξενεί και τη δικτυακή υπηρεσία. Ο client, που μπορεί να είναι εγκατεστημένος σε κινητά τερματικά, δεν χρειάζεται να έχει άμεση πρόσβαση ή να αποθηκεύει δεδομένα στις βάσεις δεδομένων. Επίσης, η δικτυακή υπηρεσία καθορίζει τον απαραίτητο κώδικα για αυτόν. Οι μέθοδοι σε μια δικτυακή υπηρεσία εκτελούνται διαμέσου της Remote Procedure Call (RPC).

XML/ XSL/ DOM/ SAX/ DTD

Τα XML (Extensible Markup Language) έγγραφα περιέχουν μόνο δεδομένα, όχι πληροφορίες παρουσίασης, συνεπώς οι εφαρμογές που επεξεργάζονται XML αρχεία πρέπει να αποφασίσουν πώς θα χειριστούν ή θα εμφανίσουν τα δεδομένα του αρχείου. Για παράδειγμα ένα PDA (personal digital assistant) αντιμετωπίζει διαφορετικά ένα XML αρχείο από ένα ασύρματο τηλέφωνο ή έναν υπολογιστή. Οι προγραμματιστές συνήθως χρησιμοποιούν την Extensible Stylesheet Language (XSL) για να προσδιορίσουν τον τρόπο παρουσίασης στις διάφορες πλατφόρμες. Η επεξεργασία ενός XML εγγράφου απαιτεί απαραίτητο λογισμικό που ονομάζεται XML parser. Οι parsers ελέγχουν τη σύνταξη του XML αρχείου και επιτρέπουν στις εφαρμογές να επεξεργάζονται marked-up δεδομένα. Οι XML parsers μπορούν να υποστηρίξουν είτε Document Object Model (DOM) είτε Simple API for XML (SAX). Οι parsers βασισμένοι σε DOM, δημιουργούν μια δενδρική δομή που περιέχει XML δεδομένα στη μνήμη. Οι parsers που βασίζονται σε SAX, επεξεργάζονται XML δεδομένα και παράγουν γεγονότα (events) κάθε φορά που ο parser συναντά tags, κείμενο, σχόλια κοκ. Αυτά τα εγείρομενα γεγονότα περιέχουν δεδομένα από το XML αρχείο. Οι εφαρμογές μπορούν να ακούν αυτά τα γεγονότα και να αντλούν τα εκάστοτε δεδομένα από το XML αρχείο. Ένα XML αρχείο μπορεί να αναφέρεται σε ένα Document Type Definition (DTD) ή αλλιώς schema, που προσδιορίζει τη σωστή δομή ενός XML αρχείου. Κάθε φορά που ένα XML αρχείο αναφέρεται σε ένα DTD ή ένα schema, ορισμένοι parsers (validating parsers) διαβάζουν το DTD/schema και ελέγχουν αν το XML αρχείο ακολουθεί τη δομή που καθορίζεται από το DTD/schema. Όταν το XML αρχείο συμμορφώνεται με το DTD/schema, το XML αρχείο χαρακτηρίζεται ως valid.

Cookies

Ένας δημοφιλής τρόπος να αλληλεπιδράσει κανείς προσωποποιημένα με μια δικτυακή σελίδα είναι με την βοήθεια των Cookies. Ένα Cookie είναι ένα

αρχείο κειμένου το οποίο αποθηκεύεται από τον δικτυακό τόπο επίσκεψης στο τερματικό του χρήστη- client. Το Cookie αποθηκεύεται μια φορά στο τερματικό (την πρώτη). Έπειτα ενεργοποιείται αυτόματα κάθε φορά που ο χρήστης επανεπισκέπτεται τον ίδιο δικτυακό τόπο. Η συλλεγόμενη πληροφορία που περιέχει είναι μια *ανώνυμη* καταγραφή τόσο δεδομένων όσο και προσωπικών προτιμήσεων, τα οποία χρειάζονται ώστε στο μέλλον ο δικτυακός τόπος να είναι προσωποποιημένος για τον συγκεκριμένο χρήστη. Μια πολύ κοινή χρήση των cookies είναι για να αποθηκεύσουν το όνομα χρήστη (user name) και τον μυστικό κωδικό ή συνθηματικό (password) που χρησιμοποιείται κάθε φορά που ο χρήστης εισέρχεται σε ένα portal. Η δυνατότητα αυτή διευκολύνει καθώς δε χρειάζεται κάθε φορά εισαγωγή των απαραίτητων στοιχείων (user name και password). Σε ότι αφορά την ασφάλεια, είναι στην διακριτική ευχέρεια του τελικού χρήστη να επιτρέψει ή όχι την ύπαρξη/ φιλοξενία των cookies στο τερματικό του σύστημα. Σε περίπτωση απαγόρευσης των cookies, ενδέχεται ο δικτυακός τόπος να μην λειτουργεί σωστά.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

1. Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων
www.eett.gr/
2. Cosmorion :Developer Uses Web Services to Extend Blogging to Mobile Phones
<http://members.microsoft.com/CustomerEvidence/Search/EvidenceDetails.aspx?EvidenceID=3437&LanguageID=1>
3. StreetHive <http://www.streethive.com/>
4. dodgeball.com : mobile social software
<http://www.dodgeball.com/>
5. MyTAP Biosciences :Web Service–based Solution Uses Mobile Phones to Remind Patients to Take Medication
<http://members.microsoft.com/CustomerEvidence/Search/EvidenceDetails.aspx?EvidenceID=3768&LanguageID=1>
6. DBGlobe : A Data - centric Approach to Global Computing (IST - 2001 - 32645) <http://softsys.cs.uoi.gr/dbglobe/index1.html>
7. MobiShare: Sharing Context-Dependent Data and Services from Mobile Sources
[Efstratios Valavanis](#), Athens University of Economics and Business
[Christopher Ververidis](#), Athens University of Economics and Business
[Michalis Vazirgianis](#), Athens University of Economics and Business
[George C. Polyzos](#), Athens University of Economics and Business
[Kjetil Norvag](#), Athens University of Economics and Business
<http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/wi/2003/1932/00/1932toc.xml&DOI=10.1109/WI.2003.1241203>
8. Jan Newmarch's Guide to Jini Technologies,
<http://jan.netcomp.monash.edu.au/java/jini/tutorial/Jini.xml>
9. Jini.org – The Community Resource for Jini Technology
<http://www.jini.org/>
10. UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, Dated 20041019
<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>
11. Web Services Essentials - Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, 1st edition, book, by Ethan Cerami, publisher O'Reilly, year 2002
12. Computer Networking: A Top-Down Approach Featuring the Internet, 2nd edition, James F. Kurose, Keith W. Ross, publisher Pearson Education, 2003.
13. SMS—Short but Sweet by Tom Clements February 2003
<http://developers.sun.com/techtopics/mobility/midp/articles/sms/>
14. Short Message Service: What, How and Where? by Puneet Gupta
<http://www.wirelessdevnet.com/channels/sms/features/sms.html>
15. 3GPP TS 03.40 V7.5.0 (2001-12), Third Generation Partnership Program, Technical Specification Group Terminals, Technical Realization of Short Message Service (SMS), Release 1998.

- http://www.3gpp.org/ftp/Specs/archive/03_series/03.40/0340-750.zip
16. How WAP Works by Jeff Tyson
<http://electronics.howstuffworks.com/wireless-internet3.htm>
 17. Wireless Application Protocol Architecture Specification
WAP-210-WAPArch-20010712-a
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-210-WAPArch-20010712-a.pdf>
 18. Push Architectural Overview WAP-250-PushArchOverview-20010703-a
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-250-PushArchOverview-20010703-a.pdf>
 19. Provisioning architecture overview WAP-182-ProvArch-20010314-a
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-182-PROVARCH-20010314-a.pdf>
 20. Java Technology <http://java.sun.com/>
 21. J2ME CLDC and K virtual machine: Frequently Asked Questions, Sun Developer Network (SDN)
<http://java.sun.com/products/cldc/faqs.html>
 22. The Apache Xalan Project <http://xalan.apache.org/index.html>
 23. Apache Xerces Project <http://xerces.apache.org/index.html>
 24. Apache Web Services Project – Axis <http://ws.apache.org/axis/>
 25. MySQL AB : The World’s most popular open source database
<http://www.mysql.com/>
 26. The Apache HTTP Server Project. <http://httpd.apache.org/>
 27. Apache Tomcat <http://httpd.apache.org/>
 28. Internet & World Wide Web How to Program- International Edition, Deitel, Third Edition.
 29. Java and Soap. Building Web Services in Java, Robert Erglander.
 30. Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI. SAMS. By: Steven Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Glen Daniels, Yuichi Nakamura, Ryo Neyama, December 2001
 31. Service-Oriented Architecture. A field guide to integrating XML and Web Services, Thomas Erl, Prentice Hall
 32. Understanding Web Services. XML, WSDL, SOAP, and UDDI. Eric Newcomer. Independent Technology Guides.
 33. Real World XML Web Services. Yasser Shohoud. Addison Wesley
 34. Professional PHP programming. Wrox Press. Jesus Castagnetto, Harish Rawat, Sascha Schumann, Chris Scollo, Deepak Veliath
 35. PHP Cookbook, O’Reilly. David Skiar, Adam Trachtenberg
 36. ASP In a Nutshell. A desktop Quick Reference. O’Reilly. A. Keyton Weissinger
 37. .NET Framework Essentials. 2nd Edition. O’Reilly. Thuan Tai & Hoang Q. Lam
 38. ASP.NET Bible.