



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Εφαρμογή των Mobile Agents για την
ανάπτυξη Virtual Home Environment
και
Location Dependent Services**

Διπλωματική Εργασία

**Βασίλειος Αλ. Μπαούσης
Α.Μ.: Μ195**

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υπεύθυνος Καθηγητής: Λ. Μεράκος

Φεβρουάριος 2001

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	1
ΜΕΡΟΣ Α'	3
VIRTUAL HOME ENVIRONMENT	5
1.1 ΕΙΣΑΓΩΓΗ.....	5
1.2 Η ΕΝΝΟΙΑ ΤΟΥ VHE	5
1.3 ΟΡΙΣΜΟΣ ΤΟΥ VHE.....	6
1.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ VHE	6
1.4.1 Υπηρεσίες στο VHE	7
1.5 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΓΙΑ ΤΟ VHE.....	8
1.5.1 Πλευρά του χρήστη	8
1.5.2 Πλευρά του διαχειριστή	9
1.6 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ	10
1.6.1 Γενική Αρχιτεκτονική για Ασύρματες υπηρεσίες δεδομένων.....	10
1.6.2 Αρχιτεκτονική χρήση	10
1.6.3 Αρχιτεκτονική Server.....	12
1.6.4 Το USER Agent	13
1.7 ΠΡΩΤΟΚΟΛΛΑ	14
1.7.1 Εισαγωγή.....	14
1.7.2 Πρωτόκολλα σχετιζόμενα με το SS7	14
1.7.3 IP Based Πρωτόκολλα και συστήματα.....	16
1.8. ΑΠΑΙΤΗΣΕΙΣ ΑΣΦΑΛΕΙΑΣ ΚΑΙ ΠΡΩΤΟΚΟΛΛΑ.....	19
1.8.1 Απαιτήσεις ασφάλειας στο VHE.....	19
1.8.2 Ανάλυση των πρωτοκόλλων ασφαλείας	21
ΜΕΡΟΣ Β'	25
ΣΥΣΤΗΜΑΤΑ ΠΑΓΚΟΣΜΙΟΥ ΔΟΡΥΦΟΡΙΚΟΥ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ (GPS).....	27
2.1 ΓΕΝΙΚΑ	27
2.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΛΕΙΤΟΥΡΓΙΚΩΝ ΤΜΗΜΑΤΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ GPS.....	27
2.3 ΔΕΚΤΕΣ	28
2.4 ΑΡΧΗ ΕΝΤΟΠΙΣΜΟΥ- ΤΡΟΠΟΙ ΜΕΤΡΗΣΗΣ	28
2.5 ΤΡΟΠΟΙ ΕΝΤΟΠΙΣΜΟΥ	29
2.5.1 Απόλυτος Εντοπισμός.....	29
2.5.2 Σχετικός Εντοπισμός	29
ΛΥΣΕΙΣ ΚΙΝΗΤΗΣ ΘΕΣΗΣ (MOBILE LOCATION SOLUTION MLS SOFTWARE) ..	31
3.1 ΕΙΣΑΓΩΓΗ.....	31
3.2 ΣΥΣΤΗΜΑΤΑ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΕ ΤΕΡΜΑΤΙΚΑ	32
<i>network Assisted Global Positioning System (A-GPS)</i>	32
<i>Enhanced Observed Time Difference (E-OTD)</i>	33
3.3 ΣΥΣΤΗΜΑΤΑ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΕ ΔΙΚΤΥΑ	34
<i>Μέθοδος παγκόσμιας ταυτότητας κελιού και καθυστέρησης χρόνου μετάδοσης (Cell Global Identity and Timing Advance, CGI-TA)</i>	34
<i>Μέθοδος μέτρησης του χρόνου άφιξης (Uplink Time Of Arrival, UL-TOA)</i>	35
3.4 ΕΦΑΡΜΟΓΕΣ ΘΕΣΗΣ	36
3.5 ΣΧΕΤΙΚΑ ΜΕ ΤΑ ΣΥΣΤΗΜΑΤΑ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ.....	36
3.6 ΤΥΠΟΠΟΙΗΣΗ	37
3.7 Η ΛΥΣΗ ΤΗΣ ERICSSON	37
ΜΕΡΟΣ Γ'	39
SOFTWARE AGENTS	41

MOBILE AGENTS	45
5.1 ΟΡΙΣΜΟΣ ΤΩΝ MOBILE AGENT	46
5.2 Το ΥΠΟΛΟΓΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ MOBILE AGENT	47
5.3 Η ΔΟΜΗ ΤΟΥ MOBILE AGENT	49
ΜΕΘΟΔΟΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΚΑΙ ΝΕΑ ΠΡΟΣΕΓΓΙΣΗ	53
ΠΛΕΟΝΕΚΤΗΜΑΤΑ	57
ΑΣΦΑΛΕΙΑ.....	61
8.1 ΠΕΡΙΠΤΩΣΕΙΣ ΑΣΦΑΛΕΙΑΣ	61
8.2 ΤΑΞΙΝΟΜΗΣΗ ΤΩΝ ΕΠΙΘΕΣΕΩΝ	62
8.2.1 Παθητικές Επιθέσεις.....	62
8.2.2 Ενεργές Επιθέσεις.....	62
8.3 ΥΠΗΡΕΣΙΕΣ ΑΣΦΑΛΕΙΑΣ –ΤΡΕΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ.....	63
ΕΦΑΡΜΟΓΕΣ	65
ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΠΟΛΛΕΣ ΤΟΠΟΘΕΣΙΕΣ	66
ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΦΙΛΤΡΑΡΙΣΜΑ	66
ΈΛΕΓΧΟΣ (MONITORING)	66
ΔΙΑΠΡΑΓΜΑΤΕΥΣΕΙΣ	66
ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ	66
ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΕΣ ΥΠΗΡΕΣΙΕΣ	67
ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ	67
ΨΥΧΑΓΩΓΙΑ	67
Η ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ MOBILE AGENTS	69
10.1 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	69
10.2 MOBILE AGENT ΣΥΣΤΗΜΑΤΑ.....	69
10.3 MOBILE AGENT ΠΡΩΤΟΚΟΛΛΑ	72
10.4 PROJECTS ΒΑΣΙΣΜΕΝΑ ΣΤΗΝ MOBILE AGENT ΤΕΧΝΟΛΟΓΙΑ.....	73
MOBILE AGENTS ΚΑΙ Η ΓΛΩΣΣΑ JAVA	75
11.1 AGENT ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ JAVA	75
11.2 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ JAVA	76
ΜΕΡΟΣ Δ'	77
AGLETS	79
12.1 ΟΡΙΣΜΟΣ	79
12.2 AGLETS ΚΑΙ APPLETS.....	79
12.3 Το ΜΟΝΤΕΛΟ ΤΟΥ AGLET	80
12.4 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΟΥ AGLET	81
12.5 Το EVENT ΜΟΝΤΕΛΟ ΤΟΥ AGLET.....	82
12.6 Το ΕΠΙΚΟΙΝΩΝΙΑΚΟ ΜΟΝΤΕΛΟ ΤΟΥ AGLET	83
AGLET API	85
13.1 Το AGLET API.....	85
13.2 Η ΥΛΟΠΟΙΗΣΗ ΤΟΥ AGLET	88
Η ΕΠΙΚΟΙΝΩΝΙΑ ΤΩΝ AGLETS.....	91
14.1 ΤΡΟΠΟΙ ΑΝΤΑΛΛΑΓΗΣ ΜΗΝΥΜΑΤΩΝ	91
14.2 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΜΗΝΥΜΑΤΩΝ	92
Η ΔΟΜΗ ΤΟΥ AGLET ΚΑΙ Η ΜΕΤΑΦΟΡΑ ΤΟΥ.....	95
15.1 Η ΔΟΜΗ ΤΟΥ AGLET	95
15.2 Η ΜΕΤΑΦΟΡΑ ΤΩΝ ΚΛΑΣΕΩΝ	96

Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ AGLET ΣΥΣΤΗΜΑΤΟΣ.....	99
16.1 AGLETS RUNTIME LAYER	99
16.2 COMMUNICATION LAYER	100
16.2.1 <i>Communication API</i>	100
16.2.2 <i>Η Αρχιτεκτονική του Communication Layer</i>	100
16.3 AGENT TRANSFER PROTOCOL.....	101
ΜΕΡΟΣ Ε'	105
ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ.....	107
17.1 ΕΙΣΑΓΩΓΗ.....	107
17.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ VHE ΚΑΙ LOCATION DEPENDENT SERVICES.	107
17.3 ΚΥΡΙΕΣ ΟΝΤΟΤΗΤΕΣ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΗΝ ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ VHE ΚΑΙ LOCATION DEPENDENT SERVICES ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΩΝ AGLETS.	109
ΑΝΑΛΥΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ.....	113
18.1 ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	113
18.2 ΔΙΑΔΙΚΑΣΙΑ ΔΗΜΙΟΥΡΓΙΑΣ ΕΝΟΣ ΧΡΗΣΤΗ ΚΑΙ ΕΓΓΡΑΦΗ ΤΟΥ ΣΤΟ ΔΙΚΤΥΟ ΚΑΙ ΠΙΣΤΟΠΟΙΗΣΗ ΧΡΗΣΤΗ.....	114
18.3 ΣΥΝΔΡΟΜΗ ΣΕ ΚΑΠΟΙΑ ΥΠΗΡΕΣΙΑ	117
18.4 ΧΡΗΣΗ ΥΠΗΡΕΣΙΩΝ	120
18.4.1 <i>Χρήση υπηρεσίας Μετεωρολογικών πληροφοριών</i>	123
18.4.2 <i>Χρήση υπηρεσίας παροχής χαρτών περιοχής χρήστη</i>	123
18.4.3 <i>Χρήση υπηρεσίας παροχής πληροφοριών θέσης κινητού χρήστη</i>	124
ΥΛΟΠΟΙΗΣΗ.....	127
19.1 ΚΩΔΙΚΑΣ.....	127
19.2 ΕΚΤΕΛΕΣΗ.....	127
ΕΠΙΛΟΓΟΣ.....	141
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ.....	143
ΠΑΡΑΡΤΗΜΑ.....	147

ΕΙΣΑΓΩΓΗ

Η βιομηχανία τηλεπικοινωνιών τις σημερινές ημέρες σχεδιάζει την εξέλιξη και την εφαρμογή της τρίτης γενιάς συστημάτων κινητής επικοινωνιών όπως το Παγκόσμιο Σύστημα Κινητών Τηλεπικοινωνιών (Universal Telecommunication System -UMTS) το οποίο θα βασίζεται σε state-of-the-art τεχνολογίες. Η τεχνολογία των Mobile Agents (Mobile Agent Technology) κερδίζει έδαφος στις τηλεπικοινωνίες, καθώς η τεχνολογία αυτή έχει αρχίσει να τυποποιείται (Mobile Agent System Interoperability Facility -MASIF).

Το UMTS θα προσφέρει μεγαλύτερη πρόσβαση, διαλειτουργικότητα και ευλυγισία από τα σημερινά δίκτυα. Ως το σύστημα τρίτης γενιάς, θα καθιστά δυνατή την ολοκλήρωση των υπηρεσιών των επικοινωνιακών συστημάτων, σε ρυθμούς, μέχρι 2Mbits/s, και υπόσχεται τον συνδυασμό τους με multimedia εφαρμογές τόσο για κινητούς όσο και για σταθερούς χρήστες.

Επίσης το UMTS θα προσφέρει στους χρήστες του την δυνατότητα να έχουν πρόσβαση στο δικό τους σετ υπηρεσιών από οπουδήποτε και εάν έχουν πρόσβαση στο δίκτυο, είτε δηλαδή από κινητό ή σταθερό δίκτυο.

Στο πλαίσιο αυτό και για να μην είναι περιορισμένοι οι διαχειριστές και οι δημιουργοί και παροχείς των υπηρεσιών, αλλά και για να επιτευχθεί όσο το δυνατό μεγαλύτερος ανταγωνισμός στην ανοικτή αγορά των επικοινωνιών, αποφασίστηκε να μην τυποποιηθούν οι υπηρεσίες που θα προσφέρονται στο UMTS, πράγμα που είχε γίνει στα δίκτυα δεύτερης γενιάς (το γνωστό μας Global System for Mobile Communications -GSM). Σύμφωνα με αυτό το σκεπτικό, θα τυποποιηθούν και θα καθοριστούν οι δυνατότητες των υπηρεσιών.

Η παροχή μη τυποποιημένων υπηρεσιών πάνω από διαφορετικά δίκτυα παρουσιάζει αναμφίβολα πολλές δυσκολίες που για να ξεπεραστούν απαιτείται η υιοθέτηση καινοτομιών και νέων τεχνικών. Αυτά τα προβλήματα εντοπίστηκαν από την International Telecommunication Union (ITU) και έχουν παρθεί μέτρα που θα επιτρέπουν την παροχή υπηρεσιών διαμέσου διαφορετικών δικτύων (service roaming).

Η έννοια-τεχνολογία που θα παρέχει την μεταφερσιμότητα των υπηρεσιών ονομάζεται Virtual Home Environment (VHE). Σύμφωνα με αυτό, κάθε υπηρεσία πάντα θα παρουσιάζεται η ίδια στον χρήστη ανεξάρτητα από το δίκτυο που έχει πρόσβαση ο χρήστης.

Η τεχνολογία των mobile agents παρέχει πολλές δυνατότητες για αποτελεσματική παροχή και εφαρμογή των υπηρεσιών αυτών. Η τεχνολογία αυτή έχει τη δυνατότητα να ικανοποιήσει τις απαιτήσεις αυτές και να πραγματοποιήσει το VHE. Τα mobile agents είναι ένα είδος λογισμικού, που συνδυάζουν τα χαρακτηριστικά των software agents, και επιπλέον, διαθέτουν την ιδιότητα της κινητικότητας. Το κύριο χαρακτηριστικό των mobile agents, είναι ότι παρέχουν στις μηχανές του δικτύου υψηλό βαθμό ευελιξίας όσον αφορά στη χρήση πόρων, επεξεργαστών και υπηρεσιών.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η παρουσίαση της έννοιας του VHE και η παροχή υπηρεσιών που έχουν το ίδιο look and feel ανεξάρτητα από το σημείο που έχει πρόσβαση ο χρήστης, είδος δικτύου και τερματικού. Οι υπηρεσίες αυτές εξαρτώνται από το γεωγραφικό σημείο που έχει πρόσβαση ο χρήστης μιας και πρόκειται για Location Dependent Services.

Οι υπηρεσίες που παρουσιάζονται είναι:

- Παροχή Μετεωρολογικών Πληροφοριών ανάλογα με το γεωγραφικό σημείο που βρίσκεται κάθε φορά ο χρήστης (Π.χ εάν βρίσκεται με ένα κινητό τηλέφωνο στο κελί μιας κεραιάς κινητής τηλεφωνίας στην Πάτρα του παρέχεται ο τοπικός καιρός της Πάτρα, στην Αθήνα της Αθήνας κλπ).

- Παροχή Πληροφοριών Θέσης. Στην υπόψη υπηρεσία χρησιμοποιείται ο αλγόριθμος υπολογισμού θέσης που εφαρμόζεται στο δορυφορικό σύστημα του Global Positioning System -GPS, μόνο που στην περίπτωση αυτή αντί για "αποστάσεις" από τέσσερις δορυφόρους, μετρώνται "αποστάσεις" από τους Σταθμούς Βάσης και από άλλα σταθερά σημεία με γνωστές

συντεταγμένες. Οι συντεταγμένες του χρήστη παρουσιάζονται σε προβολή Universal Transverse Mercator (UTM) και σε αναφορά γεωδαιτικού World Geodetic System 1984 (WGS 84) το οποίο πλέον είναι και το standard γεωδαιτικό σύστημα αναφορών όλων των νέων χαρτών και αναφορών στην αεροπλοΐα και ναυσιπλοΐα.

• Παροχή Χαρτών Περιοχής. Αυτή η υπηρεσία θα μπορούσε να θεωρηθεί σαν συνέχεια των δύο προηγούμενων αφού από την στιγμή που είναι γνωστές οι συντεταγμένες και ο Σταθμός Βάσης του χρήστη είναι δυνατόν να διαπιστωθεί το μέρος το οποίο βρίσκεται, οπότε και να του παρασχεθεί ο αντίστοιχος χάρτης. Οι χάρτες που παρέχονται στην εφαρμογή που περιλαμβάνεται στην υπόψη διπλωματική εργασία είναι κλίμακας 1:1.000.000 σε προβολή Universal Transverse Mercator (UTM) και σε αναφορά γεωδαιτικού World Geodetic System 1984 (WGS 84).

Η εργασία αποτελείται πέντε μέρη όπου στα δύο πρώτα αναλύονται η έννοια του VHE και οι βασικές αρχές και χαρακτηριστικά του GPS ενώ στο τρίτο και τέταρτο αναλύεται η τεχνολογία των mobile agents και των Aglets. Στο πέμπτο μέρος παρουσιάζεται η εφαρμογή που υλοποιεί το VHE και τις υπηρεσίες του. Αναλυτικότερα τα επιμέρους τμήματα της παρούσας διπλωματικής εργασίας περιλαμβάνουν :

Στο πρώτο μέρος, παρουσιάζεται το ο ορισμός και η έννοια του VHE, διάφορα σενάρια υπηρεσιών και η ανάλυση των απαιτήσεων που τίθενται για την ολοκλήρωση του, τόσο από την πλευρά του απλού χρήστη όσο και από την πλευρά των διαχειριστών του δικτύου. Επίσης παρουσιάζονται διάφορες αρχιτεκτονικές και πρωτόκολλα και εξετάζεται η δυνατότητα εφαρμογής τους για την πραγματοποίηση του VHE. Τέλος, αναφέρονται πρωτόκολλα που μπορούν να εφαρμοστούν για να παρέχουν ασφάλεια στις επικοινωνίες και στην παροχή των υποστηριζόμενων υπηρεσιών.

Στο δεύτερο μέρος παρουσιάζεται η έννοια του Παγκόσμιου Εντοπισμού Θέσης (Global Positioning System -GPS), η παροχή δηλαδή πληροφοριών θέσης και γενικά υπηρεσιών που εξαρτώνται από την γεωγραφική θέση του χρήστη. Γίνεται μια συνοπτική περιγραφή του GPS, των συχνοτήτων που χρησιμοποιεί καθώς και των τρόπων - μεθόδων εντοπισμού. Στο τέλος αυτής της ενότητας παρουσιάζονται οι λύσεις κινητής θέσης της Ericsson, που περιλαμβάνουν διάφορους τρόπους υπολογισμού της θέσης ενός χρήστη, είτε με την βοήθεια του δορυφορικού συστήματος του GPS, είτε με το δίκτυο του GSM, είτε με συνδυασμό τους.

Στο τρίτο μέρος, παρουσιάζεται το μοντέλο των mobile agents, σε αντιπαράθεση με τις μεθόδους της τρέχουσας τεχνολογίας και τα πλεονεκτήματά τους. Θίγεται το θέμα της ασφάλειας, που αποτελεί τροχοπέδη στο ρυθμό ευρείας αποδοχής τους και οι τρόποι αντιμετώπισης των κινδύνων. Επίσης, παρουσιάζονται τα σημαντικότερα mobile agent συστήματα και διάφορα έργα εφαρμογών τους, από όπου αναδύονται τα προτερήματα της γλώσσας προγραμματισμού Java για την υλοποίησή τους.

Το τέταρτο μέρος ασχολείται με τα Aglets, τα οποία είναι Java mobile agents. Παρουσιάζεται αναλυτικά η τεχνολογία των Aglets, το Aglet API, ο τρόπος επικοινωνίας τους και ο τρόπος μεταφοράς του κώδικα και της κατάστασής τους από υπολογιστή σε υπολογιστή του δικτύου. Επίσης, περιγράφεται η τεχνολογία των Aglet συστημάτων και το ATP (Agent Transfer Protocol), το οποίο είναι και το πρωτόκολλο επικοινωνίας τους.

Το πέμπτο μέρος αφορά στην εφαρμογή των Aglets για την υποστήριξη της έννοιας του VHE στο UMTS. Αναλύονται τα στάδια της λειτουργίας του Aglet συστήματος και τα μηνύματα που ανταλλάσσονται μεταξύ των aglets. Τέλος, παρουσιάζεται ο τρόπος υλοποίησής του συστήματος και τα αποτελέσματα της εκτέλεσής του.

Στο Παράρτημα παρουσιάζεται ο κώδικας της εφαρμογής.

Τα Aglets, και γενικότερα τα mobile agents, προσθέτουν δυναμισμό, ευελιξία και απόδοση στα συστήματα των δικτύων. Αποτελούν ένα νέο παράδειγμα των κατανεμημένων υπολογισμών και αναμένεται να επικρατήσουν στα περιβάλλοντα των μελλοντικών υπολογιστικών συστημάτων.

ΜΕΡΟΣ Α'

Virtual Home Environment

1.1 Εισαγωγή

Η αυξανόμενη αγορά των κινητών δικτύων με τους ολοένα και περισσότερους συνδρομητές με περισσότερες απαιτήσεις οδήγησαν στην ανάπτυξη και τυποποίηση της τρίτης γενιάς κινητών συστημάτων.

Η διαδικασία τυποποίησης του Universal Mobile Telecommunication System (UMTS) είναι ακόμη σε εξέλιξη, αλλά τα κύρια σημεία της διαδικασίας είναι να τυποποιηθούν όσο το δυνατό λιγότερα στοιχεία του, έτσι ώστε το σύστημα να είναι πιο ευέλικτο (Π.χ οι υπηρεσίες δεν θα τυποποιηθούν αλλά μόνο τα εργαλεία ανάπτυξης τους).

Πρώτα από όλα το Virtual Home Environment αποτελεί βασική έννοια του UMTS και καθορίζει το πλαίσιο μέσα στο οποίο έξυπνες υπηρεσίες και εφαρμογές μπορούν να αναπτυχθούν και να εφαρμοστούν από τους παροχείς των υπηρεσιών μέσα στο UMTS. Κάθε κομμάτι του VHE έχει συγκεκριμένες προαπαιτήσεις που πρέπει να ικανοποιηθούν. Αυτές αναλυτικά είναι:

Από την πλευρά του τελικού χρήστη:

- Πρόσβαση σε μια μεγάλη ποικιλία υπηρεσιών οι οποίες θα ικανοποιούν τις απαιτήσεις του και θα προσφέρονται από του διαχειριστές του συστήματος ή τους παροχείς υπηρεσιών. Αυτές οι υπηρεσίες θα είναι διαθέσιμες ανεξάρτητα από το δίκτυο που θα είναι ο χρήστης και το τερματικό που θα χρησιμοποιεί.
- Απλοποιημένη παροχή υπηρεσιών και αναβάθμιση των υπηρεσιών αυτών διαμέσου της δυνατότητα του "κατεβάσματος" νέων υπηρεσιών με την όσο πιο μικρή ανάμειξη του χρήστη.

Από την πλευρά των διαχειριστών του συστήματος:

- Ευελιξία για ανάπτυξη υπηρεσιών, οι οποίες θα ικανοποιούν τις απαιτήσεις του τελικού χρήστη σε μικρό χρονικό διάστημα, που μπορεί να διαφέρουν θεμελιωδώς μεταξύ διαφορετικών δικτύων (ασύρματα, δορυφορικά, κυψελωτά κλπ. δίκτυα) χωρίς την αλλαγή ή τροποποίηση της υποδομής του δικτύου.

Από την πλευρά των παροχέων υπηρεσιών:

- Παροχή μέσων για την δημιουργία νέων υπηρεσιών. Αυτά τα μέσα πρέπει να καθιστούν ικανούς τους παροχείς υπηρεσιών να καθορίζουν την λειτουργία του δικτύου και του τερματικού για την υλοποίηση της υπηρεσίας όπως και την παρουσίαση της υπηρεσίας στο χρήστη.

1.2 Η έννοια του VHE

Η αρχική έννοια του VHE παρουσιάστηκε στο Memorandum of Understanding (MoU) του GSM τον Ιανουάριο του 1995. Είχε προταθεί σαν η λύση για την εξέλιξη από διάφορα διαφορετικά συστήματα κινητών δεύτερης γενιάς σε ένα σύστημα το οποίο θα ήταν παγκοσμίως αποδεκτό και τυποποιημένο, το IMT -2000. Αυτή η λύση θεωρήθηκε σαν το μέσο, για τους χρήστες και τους παροχείς υπηρεσιών, που θα τους επέτρεπε την δημιουργία και εφαρμογή υπηρεσιών που θα είχαν την ίδια μορφή ακόμη και σε διαφορετικά δίκτυα.

Επίσης θα μπορούσε να ειπωθεί ότι οι περιορισμοί και τα μειονεκτήματα των υπάρχοντων κινητών τερματικών και των προσφερομένων υπηρεσιών από τα παρόντα δίκτυα

ήταν και το έναυσμα το οποίο οδήγησε στην ανάπτυξη νέων τεχνολογιών και υπηρεσιών οι οποίες θα προσφέρονται ανεξάρτητα από το που ο χρήστης χρησιμοποιεί το τερματικό του. Με αυτόν τον τρόπο ο χρήστης θα μπορεί να έχει πρόσβαση σε προηγμένες υπηρεσίες ανεξάρτητα από:

- το δίκτυο που βρίσκεται,
- τον τύπο της πρόσβασης του σε αυτό,
- και τη τερματική συσκευή που χρησιμοποιεί.

Μια νέα εποχή ανταγωνισμού θα ανοίξει μεταξύ των διαφορετικών δικτύων και των παροχών υπηρεσιών.

1.3 Ορισμός του VHE

Αυτή η παράγραφος παρουσιάζει διάφορους ορισμούς του VHE όπως αυτοί προτάθηκαν από διαφορετικά διεθνή fora. Οι ορισμοί αυτοί κάνουν αναφορά σε παρόμοιες ιδέες. Οι πιο κοινές λέξεις που χρησιμοποιούνται είναι:

- Φορητότητα (Portability) των υπηρεσιών καθώς και προσωπική φορητότητα υπηρεσιών
- Τρίτη γενιά
- Διατήρηση της διεπαφής χρήστη-εφαρμογών (User Interface) μεταξύ διαφορετικών περιβαλλόντων (δικτύων)

	Ορισμοί
ETSI- 3GPP	Το VHE είναι μια έννοια συστήματος για προσωπική φορητότητα (portability) υπηρεσιών μεταξύ διαφορετικών δικτύων και τερματικών (ETSI TR 22.70)
GSM- MoU	Το VHE είναι μια έννοια συστήματος για προσωπική φορητότητα (portability) υπηρεσιών στην Τρίτη Γενιά μεταξύ διαφορετικών δικτύων.(GSM MoU TG 21)
ITU/IMT 2000	Το VHE είναι η δυνατότητα με την οποία σε ένα χρήστη προσφέρεται η ίδια υπηρεσία στο επισκεπτόμενο δίκτυο όπως στο δίκτυο που έχει κάνει συνδρομή (ITU Q 1711)
UMTS Forum	VHE σημαίνει ότι ο χρήστης θα έχει την ίδια διεπαφή και περιβάλλον υπηρεσίας ανεξάρτητα από την θέση του (προσωπική φορητότητα και interface, ανεξάρτητα από το δίκτυο που παρέχει πρόσβαση)(UMTS Forum -Report #1)

Ο παρακάτω ορισμός περιλαμβάνει όλους τους παραπάνω ορισμούς

Το VHE είναι ένα περιβάλλον, το οποίο παρουσιάζει στον χρήστη τις προσφερόμενες υπηρεσίες με την ίδια μορφή που αυτός είναι συνηθισμένος (common look and feel interface and service experience) ανεξάρτητα από την θέση, δίκτυο και τύπο τερματικού. Το VHE βασίζεται σε τυποποιημένες δυνατότητες υπηρεσιών και προσωπικά χαρακτηριστικά τα οποία με συνέπεια παρουσιάζονται στον χρήστη ώστε να νοιώθει ότι βρίσκεται στο οικείο δίκτυο (home Network) ακόμη και εάν βρίσκεται σε άλλα δίκτυα. Απόλυτη ασφάλεια θα προσφέρεται διαφανώς μεταξύ των δικτύων πρόσβασης και δικτύων κορμού.

1.4 Χαρακτηριστικά του VHE

Κάποια θεμελιώδη χαρακτηριστικά που πηγάζουν από την έννοια του VHE είναι η προσωποποίηση των υπηρεσιών, η κλιμάκωση των υπηρεσιών και η φορητότητα των υπηρεσιών. Βασίζομενος σε αυτά τα χαρακτηριστικά ένας χρήστης μπορεί να έχει πρόσβαση σε ένα μεγάλο εύρος υπηρεσιών. Τα χαρακτηριστικά υπηρεσιών του VHE αναλύονται παρακάτω.

Προσωποποίηση Υπηρεσίας(Service Personalization)

Η έννοια του VHE δημιουργεί την ανάγκη για την τυποποίηση ορισμένων βασικών προαπαιτήσεων όπως η διαλειτουργικότητα μεταξύ διαφορετικών τεχνολογιών δικτύων, την ενδολειτουργία μεταξύ των διαφόρων επιπέδων και την ασφαλή ανταλλαγή δεδομένων που

αφορούν τον χρήστη. Παρόλα αυτά, το βασικό χαρακτηριστικό του VHE (πχ οι νέες υπηρεσίες) δεν μπορούν να τυποποιηθούν λόγω του ανταγωνιστικού περιβάλλοντος που αντιμετωπίζουν οι διαχειριστές των δικτύων και οι παροχείς υπηρεσιών. Αναμένεται ότι ένα ελάχιστο σετ υπηρεσιών και οι αντίστοιχες προδιαγραφές τους θα συμφωνηθεί μεταξύ των διαχειριστών των δικτύων και των παροχέων υπηρεσιών. Επιτακτικός είναι όμως ο καθορισμός των μέσων-εργαλείων για την δημιουργία, λειτουργία, έλεγχο και φορητότητα των υπηρεσιών.

Με τις προαναφερθείσες λειτουργίες, ο χρήστης θα έχει πρόσβαση στην λειτουργία και έλεγχο των υπηρεσιών για να μπορεί να τις προσωποποιήσει και δυναμικά να τις τροποποιήσει. Λόγω του μεγάλου αριθμού των προσφερόμενων υπηρεσιών, πρέπει να υπάρχουν ευέλικτες μέθοδοι και φιλικές στο χρήστη για να μπορεί ο χρήστης να συνθέσει ένα προφίλ το οποίο θα αντικατοπτρίζει τις απαιτήσεις του. Έτσι μπορούν να προβλεφθούν πολλά σενάρια παροχής υπηρεσιών όπως οι παροχείς υπηρεσιών να προσφέρουν στον χρήστη τα μέσα για την επιλογή των προσφερόμενων υπηρεσιών, αυτών που πραγματικά ικανοποιούν τις απαιτήσεις του και τα τεχνικά χαρακτηριστικά των συσκευών που χρησιμοποιούνται καθώς και την επιλογή της ποιότητας παροχής της υπηρεσίας (QoS). Σύμφωνα με αυτό το σενάριο ο χρήστης είναι ελεύθερος να κατασκευάσει, σε αντικειμενοστραφές περιβάλλον, το δικό του προφίλ υπηρεσιών με την απλή επιλογή των απαιτούμενων συστατικών (πλήρης προσωποποίηση υπηρεσιών). Μια εναλλακτική και πιο απλή λύση είναι ο περιορισμός του χρήστη στο να επιλέγει από ένα περιορισμένο σετ προκαθορισμένων υπηρεσιών (περιορισμένη προσωποποίηση υπηρεσιών).

Κλιμάκωση Υπηρεσίας(Service Scalability)

Αναφέρεται στη διαδικασία της προσαρμογής των δεδομένων που παράγονται από μια εφαρμογή, έτσι ώστε να χρησιμοποιείται όσο το δυνατό λιγότερο εύρος ζώνης. Αυτή η διαδικασία μπορεί να είναι χρήσιμη για δύο λόγους:

- Το δίκτυο που εξυπηρετεί τον χρήστη μπορεί να έχει συμφόρηση σε ένα κόμβο και κατάλληλες ενέργειες πρέπει να γίνουν για να ελαττωθεί ο ρυθμός των bit που παράγονται από μια εφαρμογή. Έχει βρεθεί ότι η κλιμάκωση υπηρεσίας προτιμάται από την παροχή της υπηρεσίας με την απώλεια κάποιων πακέτων δεδομένων.
- Ο χρήστη αιτείται πρόσβαση σε συγκεκριμένες εφαρμογές με συγκεκριμένες απαιτήσεις σε εύρος ζώνης και QoS οι οποίες δεν μπορούν να ικανοποιηθούν από το δίκτυο.

Φορητότητα Υπηρεσίας (Service Portability)

Οι χρήστες επιθυμούν οι υπηρεσίες που τους προσφέρονται να έχουν το ίδιο παρουσιαστικό και συμπεριφορά (ίδιο look and feel) είτε προσωπική είτε όχι, ανεξάρτητα από το δίκτυο ή την συσκευή που χρησιμοποιείται. Η πραγματοποίηση της απαιτούμενης ανεξαρτησίας δικτύου και τερματικού είναι αρκετά δύσκολη λόγω της μεγάλης ποικιλίας τόσο των κινητών όσο και των σταθερών τερματικών αλλά και των διαφορετικών δικτύων. Διαφορετικά δίκτυα/τερματικά με διαφορετικές δυνατότητες θα έχουν επίπτωση στην δημιουργία και ανάπτυξη των υπηρεσιών γιατί η δημιουργία του κώδικα που χρειάζεται μια υπηρεσία πρέπει να γίνει με τέτοιο τρόπο έτσι ώστε να μπορεί να προσαρμοστεί (σε πραγματικό χρόνο ή σε μη πραγματικό χρόνο) στις αλλαγές των δυνατοτήτων των δικτύων και τερματικών.

1.4.1 Υπηρεσίες στο VHE

Οι υπηρεσίες στο VHE μπορούν να ομαδοποιηθούν σε διάφορες κατηγορίες. Αυτές θα μπορούσαν να είναι :

Βασικές Υπηρεσίες οι οποίες θα προσφέρονται στα δίκτυα 3G είναι audio, video, fax transfer, μεταφορά δεδομένων, Internet services, ειδικά Web -browsing, e-mail/ voice mail, paging, messaging και συνδυασμός αυτών δηλαδή multimedia. Μερικές βασικές υπηρεσίες είναι τυποποιημένες γιατί η ενδολειτουργία με άλλα συστήματα θεωρήθηκε ως αναγκαία. Αυτές οι υπηρεσίες είναι:

- Φωνή

- Επείγοντες κλήσεις
- Υπηρεσία μικρών μηνυμάτων
- Fax

Συμπληρωματικές Υπηρεσίες χρησιμοποιούνται για να συμπληρώσουν και να προσωποποιήσουν τις βασικές υπηρεσίες τηλεπικοινωνιών. Έτσι δεν μπορούν να προσφερθούν σαν μόνες τους. Παραδείγματα συμπληρωματικών υπηρεσιών είναι:

- Αναγνώριση κλήσεων
- Προώθηση κλίσης
- Φραγμός κλίσεων

Value Added Υπηρεσίες οι οποίες δεν είναι τυποποιημένες. Καθορίζονται από τους παροχείς υπηρεσιών ή άλλους εξουσιοδοτημένους διαχειριστές και βασίζονται στις δυνατότητες των υπηρεσιών σαν κοινή πλατφόρμα. Μερικές από αυτές τις υπηρεσίες θα μπορούσαν να είναι:

- Video on demand
- Virtual Banking
- Mobile Office

1.5 Ανάλυση απαιτήσεων για το VHE

Οι απαιτήσεις για την πραγματοποίηση του VHE αναλύονται παρακάτω πρώτα από την πλευρά του χρήστη και έπειτα από την πλευρά του διαχειριστή του συστήματος.

1.5.1 Πλευρά του χρήστη

Αυτή η παράγραφος παρουσιάζει τις απαιτήσεις από την πλευρά του απλού χρήστη.

- Παρουσίαση του VHE

Οι υπηρεσίες του VHE πρέπει να παρουσιάζονται σε μια μορφή που να είναι εύκολα κατανοητές από τον χρήστη. Πρέπει να περιλαμβάνονται δηλαδή γραφικά όπου το κινητό τερματικό έχει την δυνατότητα ή σαν κάποια λίστα από εντολές σε τερματικά με περιορισμένες δυνατότητες.

- Δίκτυα και η υποστηριζόμενη ποιότητα επικοινωνίας

Ο χρήστης θα μπορεί να επιλέγει το πως το δίκτυο και η προσφερόμενη ποιότητα επικοινωνίας θα παρουσιάζονται σε αυτόν για διαφορετικές κλίσεις και υπηρεσίες. Το VHE θα παρέχει αυτές τις πληροφορίες σε μια γραφική μορφή όπου το κινητό τερματικό έχει την δυνατότητα ή σαν κάποια λίστα από εντολές σε τερματικά με περιορισμένες δυνατότητες. Ο διαχειριστής του δικτύου θα μπορεί να προσδιορίσει ποιες είναι οι δυνατότητες του εκάστοτε τερματικού.

- Κοστολόγηση Υπηρεσιών

Ο χρήστης θα μπορεί να προσδιορίσει πόσο κοστίζει μια υπηρεσία ή να εμφανίζεται στο τερματικό του πόσο μια υπηρεσία θα κοστίσει. Επίσης ο χρήστης θα μπορεί να αποφασίζει πως αυτές οι πληροφορίες θα παρουσιάζονται (τόνος, εικονίδιο, μήνυμα) γενικά ή ανά υπηρεσία.

- Επιλογή Υπηρεσιών

Ο χρήστης θα μπορεί εύκολα να βρίσκει ποιες υπηρεσίες είναι διαθέσιμες ή όχι, ειδικά όταν βρίσκεται σε διαφορετικό δίκτυο.

- Τροποποίηση Υπηρεσιών

Ο χρήστης θα μπορεί να τροποποιεί τα χαρακτηριστικά μια υπηρεσίας (προφίλ υπηρεσίας) από οποιοδήποτε τερματικό είτε γενικά είτε ανά κλήση. Αυτό φυσικά θα έχει αντίκτυπο στο διαχειριστή του δικτύου που θα πρέπει να υποστηρίζει τις αιτήσεις του χρήστη όπως επίσης να διαπραγματεύεται με το δίκτυο που έχει συνδρομή ο χρήστης για θέματα ασφάλειας, πρόσβασης σε υπηρεσίες καθώς και για θέματα χρέωσης.

- Πληροφορίες Θέσης

Αυτή η λειτουργία καταγράφει την θέση του τερματικού. Ο χρήστης θα είναι σε θέση να αποδεσμεύει τις πληροφορίες θέσης του ή όχι.

- Υπηρεσίες Θέσης

Ο χρήστης θα μπορεί να έχει πρόσβαση σε προσωπικές υπηρεσίες που βασίζονται στην θέση που έχει αυτός. Πχ θα μπορεί να έχει πρόσβαση σε υπηρεσίες που είναι σχετικές με την θέση που έχει όπως πληροφορίες για το κοντινότερο ξενοδοχείο ή εστιατόριο.

- Περιοδική σε άλλα δίκτυα (Roaming)

Ο χρήστης θα μπορεί να διατρέχει οποιοδήποτε δίκτυο, στα οποία υπάρχει, φυσικά, σχετική εμπορική συμφωνία, από όπου θα μπορεί να έχει πρόσβαση στις προσωπικές του υπηρεσίες τις οποίες χρησιμοποιούσε στο οικείο δίκτυο. Η φορητότητα των χαρακτηριστικών των υπηρεσιών στα διαφορετικά δίκτυα προσφέρεται από το VHE. Αυτό έχει αντίκτυπο και περιπλοκές για τους διαχειριστές των δικτύων ειδικά όταν το οικείο δίκτυο προσφέρει περισσότερες υπηρεσίες από το επισκεπτόμενο (visited) δίκτυο.

- Διαχείριση κλήσεων

Ο καλούμενος θα μπορεί να έχει διαχείριση κλήσεων όπως και αντίστροφη χρέωση όπου χρεώνεται ο καλών και όχι ο χρήστης.

- Πολυμέσα

Ο χρήστης θα μπορεί να έχει πρόσβαση σε υπηρεσίες πολυμέσων, οι οποίες περιλαμβάνουν ήχο, video και δεδομένα.

1.5.2 Πλευρά του διαχειριστή

Πρώτα από όλα από την πλευρά του διαχειριστή του δικτύου, το VHE πρέπει να παρέχει την δυνατότητα για την ανάπτυξη υπηρεσιών σε πολύ μικρό χρονικό διάστημα, που μπορεί να διαφέρουν ριζικά από δίκτυο σε δίκτυο (πχ ασύρματα, κινητής τηλεφωνίας, δορυφορικά δίκτυα κλπ), χωρίς να απαιτούνται τροποποιήσεις από τους προμηθευτές της τεχνικής υποδομής των δικτύων αυτών.

Δεύτερον απαιτείται ο διαχειριστής να προσφέρει προϊόντα και υπηρεσίες σε καλύτερη ποιότητα από αυτές που προσφέρουν οι ανταγωνιστές τους. Ο διαχειριστής θα πρέπει να είναι δημιουργικός και εφευρετικός για την δημιουργία νέων υπηρεσιών.

Τέλος, για την πραγματοποίηση του VHE, πρέπει να τονιστεί ότι ιδιαίτερες απαιτήσεις υπάρχουν για τους διαχειριστές των δικτύων για θέμα διαλειτουργικότητας. Ο διαχειριστής θα πρέπει να ξέρει:

- Που να βρει τον κατάλληλο παροχέα υπηρεσιών να χρεώσει
- Ποιόν να χρεώσει και πως, για την κίνηση και για την σηματοδότηση
- Πως να διερμηνεύσει την διαπραγμάτευση για ποιότητα παροχής της υπηρεσίας (QoS)
- Πως να διερμηνεύσει την διευθυνσιοδότηση
- Πως να διερμηνεύσει τα δεδομένα των αιτούμενων υπηρεσιών
- Πως να διερμηνεύσει τις υπηρεσίες που προσφέρονται
- Πως να δρομολογήσει τα εισερχόμενα σήματα
- Πως να διερμηνεύσει το User Interface που λαμβάνει από το τερματικό του χρήστη, σε πληροφορία ελέγχου συνόδου ή σε κάτι άλλο
- Πως να διατηρεί προσωρινά δεδομένα σε συγχρονισμό με δεδομένα που έχει ο παροχέας της υπηρεσίας.

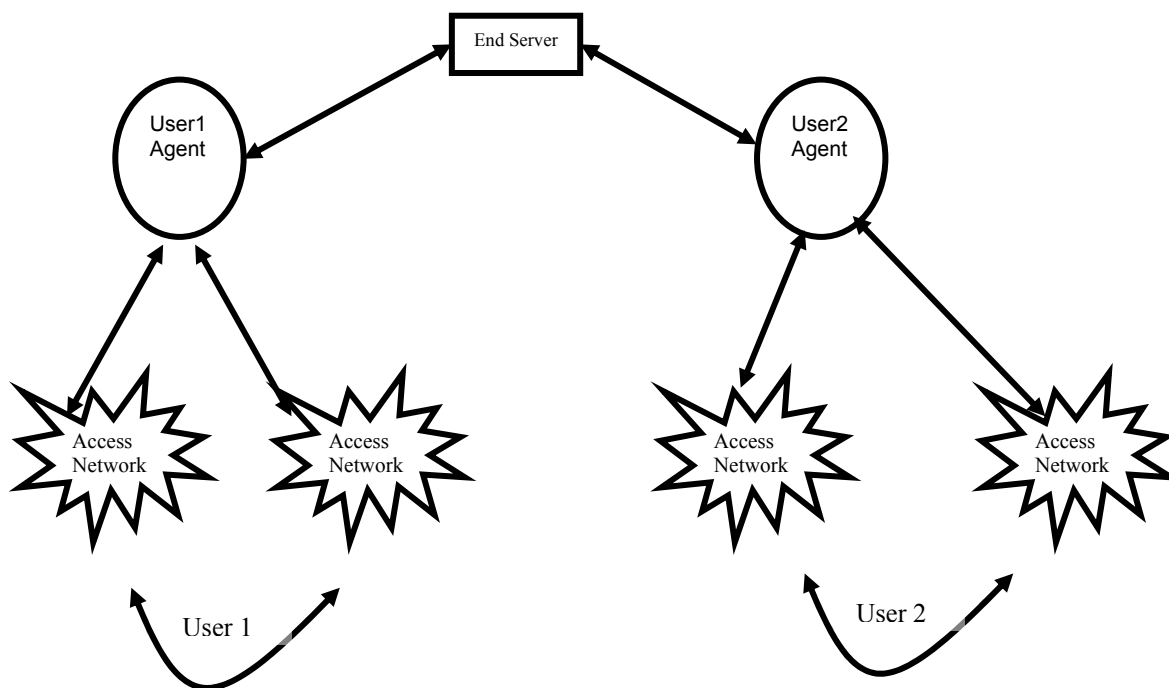
Το πηγαίο (originate) δίκτυο συγκεκριμένα θα πρέπει να ξέρει:

- Πως να χρεώσει για την κίνηση που μεταφέρεται από το δίκτυο του
- Πως να λειτουργήσει με τα άλλα δίκτυα
- Πως να χρεώσει για την σηματοδότηση
- Πως να δρομολογήσει τις διάφορες συνόδους
- Πως να δρομολογήσει τα σήματα
- Πως να διατηρήσει προσωρινά δεδομένα εάν αυτό είναι αναγκαίο.

1.6 Προσδιορισμός Αρχιτεκτονικών

1.6.1 Γενική Αρχιτεκτονική για Ασύρματες υπηρεσίες δεδομένων

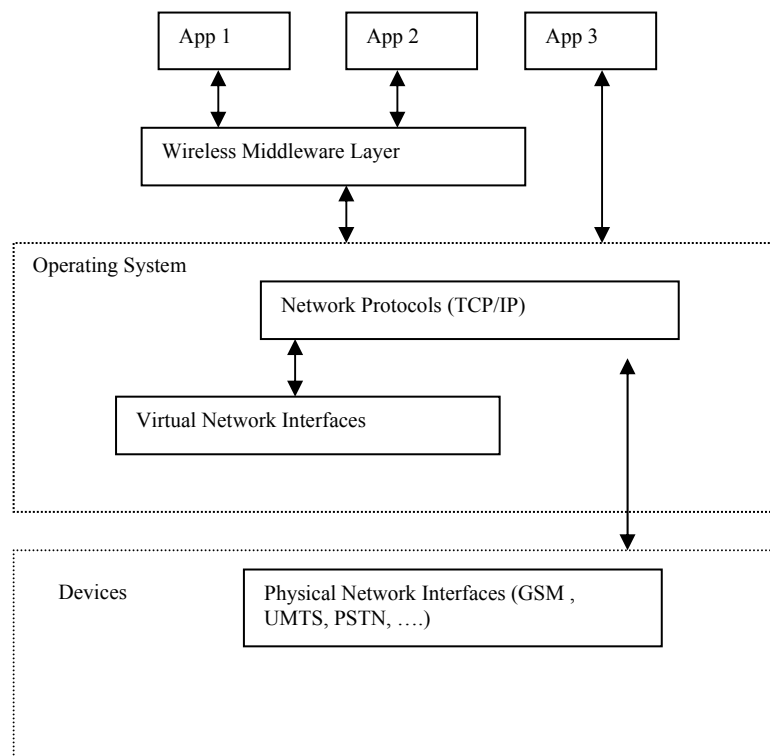
Παρακάτω παρουσιάζεται μια αρχιτεκτονική για ασύρματες υπηρεσίες δεδομένων (Σχήμα 1). Στο υψηλότερο επίπεδο αυτής της αρχιτεκτονικής υπάρχει μια δομή που αποτελείται από τρία επίπεδα. Στο υψηλότερο επίπεδο, υπάρχει ο κώδικας για κάθε υπηρεσία και περιέχεται σε κάθε server, ο οποίος είναι υπεύθυνος για να προσφέρει την επεξεργασία για να προσφερθεί η υπηρεσία. Στην μέση βρίσκεται μια συλλογή από agents χρηστών, καθένα από τα οποία είναι ο ενδιάμεσος μεταξύ του server και του χρήστη. Κάθε agent μπορεί να θεωρηθεί σαν την λογική αντιπροσώπευση του χρήστη στο Server. Κρύβει έτσι το δίκτυο και την κίνηση του χρήστη, έτσι ο server να μη χρειάζεται να γνωρίζει λεπτομέρειες για τους χρήστες. Εξαρτάται από τα agents και την διαφάνεια την οποία έχουν αυτά, το εάν ο server γνωρίζει ότι πρόκειται για κινητούς χρήστες. Στην βάση της προτεινόμενης αρχιτεκτονικής είναι οι χρήστες των υπηρεσιών που χρησιμοποιούν συγκεκριμένες συσκευές με καθορισμένες δυνατότητες με τις οποίες είναι προσαρτημένοι με μια συγκεκριμένη πρόσβαση στο δίκτυο.



Σχήμα 1. Αρχιτεκτονική υπηρεσιών.

1.6.2 Αρχιτεκτονική χρήστη

Εξαιτίας της αυστηρής και περιοριστικής φύσης της ασύρματης πρόσβασης (μεγαλύτερη καθυστέρηση, συχνή διακοπή, ριπαιούς και τυχαίους ρυθμούς σφαλμάτων) σε σύγκριση με την ενσύρματη πρόσβαση, ο χρήστης πρέπει να παρέχει την επιπρόσθετη λειτουργικότητα που απαιτείται. Το διάγραμμα (Σχήμα 2) παρουσιάζει την αρχιτεκτονική του χρήστη, ενώ παρακάτω αναλύονται τα συστατικά του μέρη.



Σχήμα 2 Αρχιτεκτονική χρήστη εφαρμογής

Wireless Middleware Layer - Data Plane. Υποστηρίζονται ασύγχρονη και σύγχρονη μεταφορά δεδομένων. Το επίπεδο αντιμετωπίζει τις ασύγχρονες "ανακοινώσεις" (notifications) σαν να είναι μέρος των εφαρμογών, όπως επίσης και τις ενδιάμεσες λεπτομέρειες επικοινωνίας που είναι συνηθισμένες σε ασύρματες επικοινωνίες.

Transaction Layer: Παρέχει υποστήριξη στις αλληλεπιδράσεις αίτησης - απόκρισης που απαιτούνται από πολλές εφαρμογές. Οι απαιτούμενες λειτουργίες πραγματοποιούν τον σχηματισμό των μηνυμάτων, για τις επανεκπομπές αιτήσεων και αποκρίσεων (απαιτούμενες για την ασταθή ασύρματη επικοινωνία) και για ασφάλεια κλπ.

Adaptation Layer: Ο σκοπός αυτού του επιπέδου είναι στο να καθιστά ικανές τις εφαρμογές να "τρέχουν" με τον πιο αποτελεσματικό τρόπο με τους υπολογιστικούς πόρους που τους έχουν διατεθεί για αυτό το σκοπό. Το επίπεδο χειρίζεται τις λεπτομέρειες του δικτύου και διαιρείται σε δύο υποεπίπεδα:

Upper Adaptation Layer :Παρέχει υποστήριξη σε μια εφαρμογή για να χειρίζεται τα χαρακτηριστικά της, τα δεδομένα απόδοσης της και τον σχηματισμό των δεδομένων. Επίσης ειδοποιεί για αλλαγή των πόρων του συστήματος και βοηθά στην αρχικοποίηση νέων διαδρομών δεδομένων που ζητούνται από αιτήσεις των εφαρμογών. Οι συνέπειες του υποεπιπέδου αυτού μπορεί να είναι εμφανείς στον τελικό χρήστη (πχ υποβάθμιση video λόγω μικρού bandwidth κλπ).

Lower Adaptation Layer :Λειτουργεί μαζί με το δίκτυο για να παρέχει στις εφαρμογές διαφάνεια στην προσαρμογή του συστήματος (όπως κωδικοποίηση σφαλμάτων, συμπίεση) όταν συμβαίνει μεταπομπή(hand-off).

Wireless Middleware Layer- Control Plane

State Management: Αυτό το υποσύνολο λειτουργιών μέσα στο Service Client, χειρίζεται τις λειτουργίες μεταφοράς των εφαρμογών όταν συμβαίνει αλλαγή δικτύου. Αυτές οι λειτουργίες μπορεί να περιλαμβάνουν : handshake μεταξύ του client και του τελικού εξυπηρέτη, ανταλλαγή

πληροφοριών κατάστασης, διαφανή επανεκκίνηση της εφαρμογής χρησιμοποιώντας τις πληροφορίες κατάστασης, τερματισμό μιας εφαρμογής κλπ.

Profile Management: Παρέχει ομοιόμορφη διεπαφή πρόσβασης σε διαφορετικές πληροφορίες. Αποτελείται από το προφίλ του συνδρομητή, τις προτιμήσεις του, τις δυνατότητες της συσκευής και του δικτύου κλπ.

Network Management: Εντοπίζει τα δίκτυα στα οποία υπάρχει πρόσβαση κάθε στιγμή και αρχικοποιεί μονοπάτια προσαρμογής δεδομένων ή μεταπομπές σε άλλα δίκτυα.

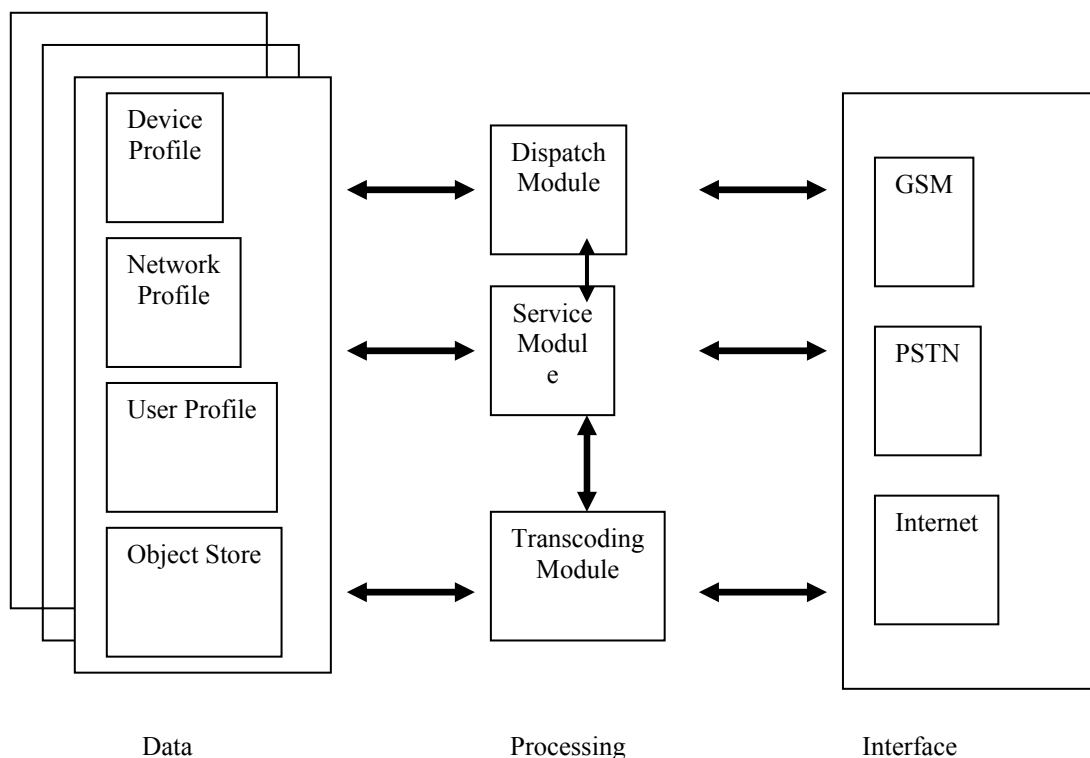
Device Management: Παρέχει διεπαφή ανεξάρτητη από την συσκευή.

1.6.3 Αρχιτεκτονική Server

Η αρχιτεκτονική του Server φαίνεται στο σχήμα 3. Αποτελείται από τρία μεγάλα τμήματα: την διεπαφή, την επεξεργασία και τα δεδομένα, τα οποία αναλύονται στη συνέχεια.

Διεπαφή:

Το τμήμα της διεπαφής αποτελείται από ένα σετ διεπαφών δικτύων επιτρέποντας έτσι στα μηνύματα να αποστέλλονται και να λαμβάνονται από διαφορετικά δίκτυα (Short Message Service -SMS, Cellular Digital Packet Data - CDPD, General Packet Radio System- GPRS, Personal Switched Telephone Network- PSTN και Internet). Όταν ένα μήνυμα παραληφθεί από έναν αποστολέα, το είδος του δικτύου δεν έχει σημασία.



Σχήμα 3 Αρχιτεκτονική τελικού εξυπηρέτη

Data Component

Είναι οργανωμένο ανάλογα με τον χρήστη. Καθένα από αυτό περιέχει τα χαρακτηριστικά της συσκευής, του δικτύου και πληροφορίες κατάστασης.

Processing Code

Είναι ο πυρήνας του server και παρέχει το κώδικα για τις υπηρεσίες. Αποτελείται από τρία υποεπίπεδα: το Dispatch Module, το Service Module και το Transcoding Module.

Dispatch Module: Το υποεπίπεδο αυτό παίρνει τρεις σημαντικές αποφάσεις: α) ποια υπηρεσία να καλέσει, β) ποιου είδους προσαρμογή να εκτελέσει (εαν χρειάζεται), και γ) σε ποιο δίκτυο να αποστείλει μήνυμα. Αυτές οι αποφάσεις εξαρτώνται από τις προτιμήσεις του χρήστη και από την διαθεσιμότητα του δικτύου και της συσκευής.

Service Module: Παρέχει την έννοια της χρήσης μιας μόνο εφαρμογής στο χρήστη, με το να παρακολουθεί την πρόσβαση του χρήστη στο δίκτυο. Επίσης ελέγχει τις πληροφορίες κατάστασης που είναι μόνιμες στον χρήστη (όπως πχ bookmarks, cookies και ιστορικό Web Browsing) και μπορούν να μεταφερθούν όταν αυτός αλλάζει συσκευή.

Transcoding Module: Παρέχει την κατάλληλη προσαρμογή όταν ο χρήστης αλλάζει συσκευή ή δίκτυο. Υπάρχουν δύο βασικοί τύποι μηχανισμών transcoding: α) αλλαγή περιεχομένου πληροφοριών και β) αλλαγή media format.

Η αλλαγή περιεχομένου πληροφοριών μπορεί να ενεργοποιηθεί από την αλλαγή παραδείγματος χάρη της συσκευής το οποίο μπορεί να επηρεάσει τους κάτωθι παράγοντες:

- Διαθέσιμο εύρος δικτύου το οποίο έχει ως συνέπεια σημαντική καθυστέρηση στην μεταφερόμενη πληροφορία
- Υπολογιστική δύναμη
- Χωρητικότητα μνήμης
- Δυνατότητες παρουσίασης των δεδομένων (display capabilities)

Αλλαγή media format μπορεί να προκληθεί από ασυμβατότητα μέσω (πχ μετακίνηση από laptop σε ένα συμβατικό PSTN τηλέφωνο).

1.6.4 To USER Agent

Τα user agents εκτελούν κυρίως λειτουργίες ελέγχου. Μπορούν να παρέχουν πληροφορίες για τον χρήστη στο υπόλοιπο σύστημα. Επίσης μπορούν να αντιπροσωπεύουν τον χρήστη όταν είναι αποσυνδεδεμένος από το δίκτυο. Η πραγματική διαδρομή δεδομένων μεταξύ του τελικού server και του χρήστη μπορεί να περνά ή και να μην περνά από το αντίστοιχο user agent. Εάν το user agent παρέχει value -added λειτουργίες, τότε η διαδρομή αυτή των δεδομένων περνά μέσα από το user agent. Αυτός κρατά πληροφορίες για την κατάσταση της συσκευής του χρήστη και λειτουργεί σαν τον τερματισμό της γραμμής για τα πρωτόκολλα επικοινωνίας.

Παρόλα αυτά, επειδή η καθυστέρηση είναι βασικός παράγοντας, η δρομολόγηση των δεδομένων μέσω του user agent μπορεί να μην είναι αποδεκτή. Εξαρτάται από το σύστημα, το μέγεθος της κυκλοφορίας από την σηματοδότηση και από το εάν το user agent μπορεί να μεταναστεύσει από μόνο του όπως και οι πραγματικοί χρήστες.

Το user agent μπορεί να έχει ένα σταθερό και ένα επεκταμένο τμήμα. Το πρώτο τμήμα ασχολείται με βασικές λειτουργίες μηνυμάτων τα οποία πάντα απαιτούνται. Επίσης υποστηρίζει λειτουργίες που έχουν σχέση με το προφίλ του χρήστη, για τον έλεγχο της κατάστασής του. Αυτές οι λειτουργίες αναμένεται να είναι ίδιες σε όλους τους χρήστες. Το επεκταμένο τμήμα είναι προγραμματιζόμενο να εκτελεί συγκεκριμένα καθήκοντα που επιθυμεί κάποιος χρήστης. Αποτελείται από τα User Agent programs (UA) τα οποία κατηγοριοποιούνται σε δύο κατηγορίες: α) τα προγράμματα διαμόρφωσης και β) τα προγράμματα εφαρμογής. Τα προγράμματα διαμόρφωσης εφαρμόζονται στις διαδρομές των δεδομένων, ενώ τα δεύτερα περιέχουν δέσμες προγραμμάτων που χειρίζονται μηνύματα ενός συγκεκριμένου σχήματος.

1.7 Πρωτοκόλλα

1.7.1 Εισαγωγή

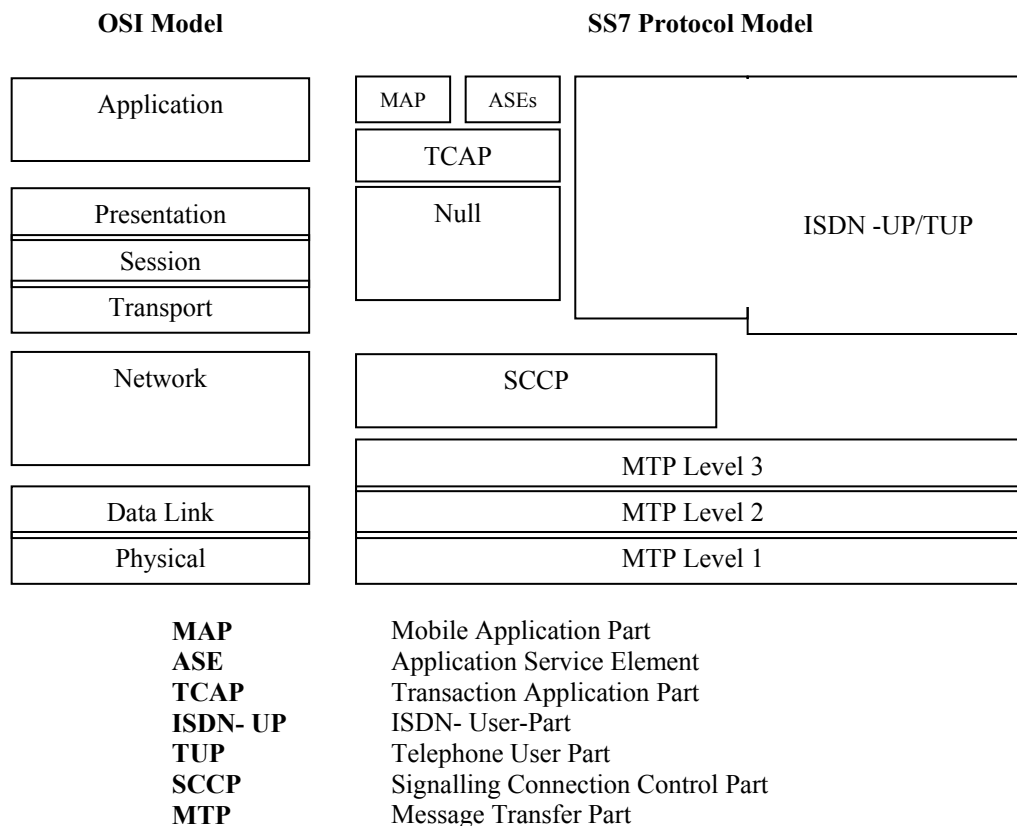
Η παράγραφος αυτή παρουσιάζει μια συνοπτική περιγραφή για μερικά σημαντικά πρωτόκολλα που χρησιμοποιούνται σήμερα και αναμένεται να παίξουν σημαντικό ρόλο για τον καθορισμό του Virtual Home Environment.

Υπάρχουν τουλάχιστον δύο προσεγγίσεις για την χρησιμοποίηση αυτών των πρωτοκόλλων. Η μία είναι τα πρωτοκόλλα που προέρχονται από τα παραδοσιακά δίκτυα τηλεπικοινωνιών, τα οποία περιλαμβάνουν πρωτόκολλα βασισμένα σε σηματοδοσία που ήδη χρησιμοποιούνται σε σταθερά και κινητά δίκτυα επικοινωνιών ενώ η δεύτερη προσέγγιση είναι τα πρωτοκόλλα που βασίζονται στο IP.

1.7.2 Πρωτόκολλα σχετιζόμενα με το SS7

Signalling System 7 (SS7)

Στις σύγχρονες τηλεπικοινωνίες, η σηματοδοσία μπορεί να ορισθεί σαν το σύστημα το οποίο καθιστά ικανή την αποθήκευση προγραμμάτων, ανταλλαγή μηνυμάτων ελέγχου, πρόσβαση και χρήση βάσεων δεδομένων μέσα στο δίκτυο, επικοινωνία με άλλους νοήμονες κόμβους μέσα στο δίκτυο με τους οποίους μπορεί να γίνει ανταλλαγή μηνυμάτων για την εγκαθίδρυση, παρακολούθηση μιας συνδιάλεξης, ανταλλαγή πληροφοριών ελέγχου για μια κλήση/ σύνδεση που απαιτούνται για κατανεμημένη επεξεργασία και έλεγχο. Στο σχήμα 4 παρουσιάζεται η δομή του SS7 πρωτοκόλλου και πως αυτή η δομή σχετίζεται με τα επίπεδα του Open System Interconnection (OSI).



Σχήμα 4. Η δομή του πρωτοκόλλου SS7.

Το SS7 καλύπτει το σύνολο της σηματοδότησης των κλασικών τηλεπικοινωνιών και για αυτό είναι υποψήφιο πρωτόκολλο για να προσφέρει VHE δυνατότητες. Συγκεκριμένα τα TCAP & MAP μπορούν να είναι χρήσιμα: το πρώτο για την υποστήριξη συναλλαγών με IN υπηρεσίες και την δυνατότητα για εκτέλεση ρουτινών σε κάποιο άλλο κόμβο. Το MAP μπορεί να προσφέρει λειτουργίες σηματοδότησης για το κινητό domain (το οποίο είναι και το κύριο domain για το VHE). Πιθανές βελτιώσεις στο MAP μπορούν να υποστηρίξουν όλες τις λειτουργίες του UMTS.

MAP

Το MAP (Mobile Application Part) είναι ένα σετ πρωτοκόλλων για την υλοποίηση του ελέγχου της κινητικότητας του χρήστη στο GSM. Για να το επιτύχει αυτό είναι απαραίτητο για τους κόμβους μεταγωγής (MCS, GMSC) να έχουν κάποιο 'διάλογο' με τις βάσεις δεδομένων πράγμα που πραγματοποιείται με την ανταλλαγή MAP μηνυμάτων. Τα μηνύματα αυτά χρησιμοποιούνται για την ενημέρωση των βάσεων δεδομένων για την θέση του κινητού σταθμού, για να μεταφέρουν πληροφορίες (πχ προφίλ χρήστη) από μία βάση σε μια άλλη μέσα στο δίκτυο και στο να αντλούν πληροφορίες από βάσεις δεδομένων για το εάν κάποιος χρήστης είναι εξουσιοδοτημένος να έχει πρόσβαση στους πόρους του συστήματος κλπ. Το γεγονός ότι είναι ένα σετ πρωτοκόλλων σημαίνει ότι περιέχονται αρκετές διεπαφές πράγμα που το κάνει πολύπλοκο.

Από την στιγμή που το MAP προσφέρει έλεγχο της κινητικότητας του χρήστη στο GSM και το ότι είναι το πιο διαδεδομένο πρωτόκολλο για αυτό τον σκοπό, μπορεί να χρησιμοποιηθεί και σε άλλα συστήματα και μπορεί να γίνει ένα χρήσιμο εργαλείο για το VHE- με την έννοια μιας παγκόσμιας κοινωνίας- για έλεγχο της κινητικότητας των χρηστών.

INAP

Το INAP είναι τυπικό για τις υπηρεσίες των Intelligent Network (IN) για σταθερά δίκτυα. Για κινητά δίκτυα οι υπηρεσίες των Intelligent Network (IN) καλύπτονται από το CAMEL (Customized Applications for Mobile network Enhanced Logic). Το CAMEL είναι πρότυπη διαδικασία της ETSI GSM για να παρέχει υποστήριξη roaming στις υπηρεσίες των Intelligent Network οι οποίες ελέγχονται από το οικείο δίκτυο. Το CAMEL βασίζεται στο INAP CS-1. Ενώ το INAP επιτρέπει την χρήση των IN υπηρεσιών σε σταθερά δίκτυα, το CAMEL επιτρέπει στους συνδρομητές των ασύρματων επικοινωνιών να έχουν πρόσβαση σε IN υπηρεσίες ενώ κινούνται.

CAMEL

Το CAMEL (Customized Applications for Mobile network Enhanced Logic) είναι ένα εργαλείο για την παροχή στους συνδρομητές λειτουργικών υπηρεσιών ακόμα και όταν αυτοί είναι εκτός του οικείου δικτύου. Αντιπροσωπεύει την προσπάθεια των IN να αναβαθμιστούν σε GSM και για αυτό δεν πρέπει να θεωρείται σαν συμπληρωματική υπηρεσία αλλά σαν χαρακτηριστικό των IN δικτύων. Με άλλα λόγια είναι ένας μηχανισμός δικτύου που προσθέτει IN λειτουργίες στα GSM δίκτυα και έχει ως σκοπό να τυποποιήσει τις δυνατότητες των υπηρεσιών παρά τις υπηρεσίες. Μπορεί να θεωρηθεί σαν το πρώτο βήμα προς το VHE, που γίνεται από την ETSI για το UMTS, και σαν ένα υποσύνολο του δεύτερου.

Γενικά τα IN δίκτυα έχουν πρακτική εφαρμογή στα σύγχρονα δίκτυα και αυτό είναι το πλεονέκτημα τους. Καθορίζουν επίσης ένα πρωτοποριακό σετ προτύπων ενώ και οι διεπαφές μεταξύ των οντοτήτων του δικτύου έχουν τυποποιηθεί. Ο έλεγχος και τα δεδομένα των υπηρεσιών είναι ξεχωριστά, πράγμα που καθιστά δυνατή την εφαρμογή νέων υπηρεσιών με ένα αρθρωτό (modular) τρόπο. Επίσης επιτρέπουν global roaming μεταξύ κινητών και σταθερών δικτύων για την μεταφορά των προφίλ των χρηστών.

1.7.3 IP Based Πρωτόκολλα και συστήματα

Το Internet αποτελεί μια συλλογή δικτύων τα οποία επικοινωνούν χρησιμοποιώντας ένα κοινό σετ από πρωτόκολλα, που επίσημα ονομάζεται TCP/IP (από τα δύο βασικά πρωτόκολλα που περιλαμβάνονται στο τυποποιημένο σετ πρωτοκόλλων). Εξαιτίας της ευλυγισίας του το TCP/IP έγινε το de facto πρότυπο δικτύου και είναι συσχετισμένο με πολλές δικτυακές υπηρεσίες όπως e-mail, μεταφορά αρχείων και το WWW (World Wide Web).

Η πρόκληση για ένα επεκταμένο Internet και τις εμφανιζόμενες νέες υπηρεσίες είναι να δημιουργηθούν TCP/IP πρωτόκολλα δυνατά να υποστηρίξουν τις νέες υπηρεσίες και τις απαιτήσεις για κίνηση του χρήστη.

Το πρωτόκολλο IP

Παρακάτω αναλύονται ορισμένα πρωτόκολλα που χρησιμοποιούνται σήμερα πάνω από το IP και πως τα χαρακτηριστικά τους μπορούν να προσαρμοστούν στην χρήση των κινητών συστημάτων επικοινωνίας.

User Datagram Protocol (UDP)

Αυτό το πρωτόκολλο χρησιμοποιεί το IP για να παρέχει αναξιόπιστη, χωρίς σύνδεση, παροχή υπηρεσιών. Η βασική διαφορά του με το IP συνίσταται στο ότι όταν το IP μεταφέρει δεδομένα μεταξύ δύο κόμβων, το UDP χρησιμοποιείται για να καθορίσει-αναγνωρίσει την εφαρμογή σε αυτόν το Host για την οποία μεταφέρει τα δεδομένα. Αυτό αποτελεί και το χαρακτηριστικό, δηλαδή η δυνατότητα αποστολής δεδομένων σε μια συγκεκριμένη εφαρμογή σε ένα συγκεκριμένο Host με όσο το δυνατό λιγότερους μηχανισμούς πρωτοκόλλων, που μπορεί να είναι χρήσιμο στα ασύρματα δίκτυα του UMTS. Ταυτόχρονα η μη αξιόπιστη παροχή των μηνυμάτων είναι και το μειονέκτημα του, πράγμα που κάνει τις εφαρμογές που τρέχουν πάνω από το UDP να αναλαμβάνουν αυτές την διεκπεραίωση της δουλειά αυτής, αυξάνοντας έτσι την πολυπλοκότητα του. Επίσης μεγάλη διακύμανση έχει η αξιοπιστία του σε ραδιοζεύξεις.

Transmission Control Protocol (TCP)

Αυτό το πρωτόκολλο παρέχει αμφίδρομη και αξιόπιστη μεταφορά δεδομένων και παροχή υπηρεσιών μεταξύ δύο Host. Το πρωτόκολλο αυτό σχεδιάστηκε και βελτιώθηκε για το ενσύρματο Internet και είναι ευρέως διαδεδομένο και εφαρμοσμένο σε πολλών ειδών δίκτυα: τοπικά, εθνικά και παγκόσμια.

Το TCP δεν μπορεί να ικανοποιήσει όμως κάποιες ειδικές ανάγκες της ασύρματης επικοινωνίας, όπως η απώλεια πακέτων, που στο Internet αντιμετωπίζεται σαν συμφόρηση σε κάποιο ενδιάμεσο κόμβο, ενώ στην ασύρματη τηλεφωνία οι διακοπές και η απώλεια δεδομένων μπορεί να οφείλεται σε κακές συνθήκες της ραδιοζεύξης ή στην έλλειψη καναλιών κατά την διάρκεια των ωρών αιχμής. Στην περίπτωση αυτή το TCP υποθέτει ότι κάποιος δρομολογητής έχει συμφόρηση και ελαττώνει την εκπομπή, μειώνει το μέγεθος του παραθύρου εκπομπής στο μισό και αυξάνει τον χρόνο επανεκπομπής εκθετικά. Αυτό λειτουργεί αποτελεσματικά στο Internet αλλά όταν η απώλεια των πακέτων οφείλεται σε προσωρινή απορρόφηση του σήματος ή άλλες ατμοσφαιρικές καταστάσεις - περίπτωση ασύρματης επικοινωνίας- τότε η θεραπεία χειροτερεύει την αρρώστια.

Αυτό οδηγεί στο συμπέρασμα ότι το TCP είναι βαρύ πρωτόκολλο για να είναι αποτελεσματικό στο UMTS, αλλά δεν θα πρέπει να αποκλεισθεί οριστικά διότι μπορεί να ελαχιστοποιηθεί η επίδραση του στην μεταφορά, με την χρησιμοποίηση κάποιων αλλαγών στα ανώτερα επίπεδα του πχ το HTTP-NG αντί του HTTP, που περιλαμβάνει και τα χαρακτηριστικά της μεταφοράς δεδομένων των ασύρματων επικοινωνιών.

File Transfer Protocol (FTP)

Όπως δηλώνει και το όνομα του FTP, είναι ένα πρωτόκολλο για την μεταφορά αρχείων και όχι υποστήριξη συνεχόμενων ροών δεδομένων. Το FTP τρέχει πάνω από το TCP, όποτε όλα τα προηγούμενα για το TCP ισχύουν και στο FTP. Επίσης είναι μεγάλο πρωτόκολλο για το UMTS πράγμα που θα είναι ίσως πρόβλημα στα σχετικά μικρά τερματικά με περιορισμένο αποθηκευτικό χώρο. Επίσης δεν περιλαμβάνει πιστοποίηση χρήστη και έλεγχο πρόσβασης σε κάποιο ικανοποιητικό επίπεδο.

Το TFTP (Trivial File Transfer Protocol) είναι πολύ πιο "ελαφρύ" πρωτόκολλο από το FTP, το οποίο χρησιμοποιεί μια σύνδεση UDP σε αντίθεση με το FTP που χρησιμοποιεί δύο συνδέσεις TCP και κρίνεται ακατάλληλο για ασύρματη χρήση.

Το SFTP (Simple File Transfer Protocol) είναι σχεδιασμένο για να καλύψει το κενό μεταξύ FTP και TFTP. Είναι πιο απλό από το FTP άρα και κατά συνέπεια πιο αναποτελεσματικό από αυτό και χρησιμοποιεί μια σύνδεση TCP.

Οι διαφορετικές εκδόσεις του FTP είναι πιθανές για χρήση στο UMTS για μεταφορά αρχείων και μπορούν να χρησιμοποιηθούν και σε ορισμένες υπηρεσίες.

Simple Mail Transfer Protocol (SMTP)

Το SMTP είναι ένα πρωτόκολλο ταχυδρομείου που τρέχει πάνω από οποιοδήποτε πρωτόκολλο μεταφοράς, περιλαμβανομένου και του TCP. Ελαχιστοποιεί την χρήση του διαθέσιμου bandwidth πράγμα που είναι σημαντικό στην ασύρματη επικοινωνία. Από την άλλη μεριά η αποστολή ενός μηνύματος με το SMTP απαιτεί πολλές ανταλλαγές εντολών /απαντήσεων. Επίσης το SMTP καταναλώνει αρκετό εύρος ζώνης όταν χρησιμοποιείται για την μεταφορά οτιδήποτε άλλης πληροφορίας εκτός 7-bit ASCII πληροφορίας. Έτσι όλα τα δεδομένα θα πρέπει να υποστούν περαιτέρω επεξεργασία πράγμα που απαιτεί επιπρόσθετη επεξεργασία, μνήμη τόσο στον αποστολέα όσο και στον παραλήπτη. Η κωδικοποίηση των 7 σημαντικών bit σε 8 σημαντικά bit μεγαλώνει το μέγεθος των δεδομένων κατά 14% πράγμα που δεν είναι συχνά αποδεκτό.

Real-time Transport Protocol (RTP)

Το RTP παρέχει λειτουργίες μεταφοράς από την μια στην άλλη άκρη του δικτύου, που είναι κατάλληλες για την εκπομπή δεδομένων σε πραγματικό χρόνο, video ή δεδομένα προσομοίωσης, πάνω από multicast ή unicast υπηρεσίες δικτύου.

Το RTP δεν είναι ένα ολοκληρωμένο πρωτόκολλο και προορίζεται για το επίπεδο εφαρμογής, παρά να εφαρμοστεί σαν ξεχωριστό πρωτόκολλο.

Mobile IP

Σε ένα κανονικό router κάθε διεύθυνση IP είναι συνδυασμένη με μια σταθερή θέση στο δίκτυο. Όταν ο προορισμός ενός πακέτου είναι ένας κινητός κόμβος, αυτό σημαίνει ότι κάθε νέο σημείο σύνδεσης που γίνεται από τον κόμβο αυτό συνδυάζεται με ένα νέο αριθμό δικτύου και μια IP διεύθυνση καθιστώντας την διαφανή κινητικότητα (mobility) αδύνατη.

Το Mobile IP είναι ένα πρότυπο, που προτάθηκε από ομάδα εργασίας της Internet Engineering Task Force (IETF) και σχεδιάστηκε για να λύσει αυτό το πρόβλημα επιτρέποντας στον κινητό κόμβο να έχει δύο IP διευθύνσεις: μια σταθερή (Home Address) και μια προσωρινή (care-of address) η οποία αλλάζει σε κάθε σημείο σύνδεσης. Η πρώτη χρησιμοποιείται πχ για την αναγνώριση συνδέσεων TCP ενώ η δεύτερη μπορεί να θεωρηθεί σαν η "τοπογραφικά σημαντική διεύθυνση του κινητού" κόμβου και δείχνει τον αριθμό του δικτύου, φανερώνοντας έτσι το σημείο σύνδεσης του κινητού κόμβου σε σχέση με την τοπολογία του δικτύου. Η Home Address κάνει να φαίνεται ότι ο κινητός κόμβος είναι συνεχώς έτοιμος να δεχθεί δεδομένα στο δικό του Home δίκτυο, ενώ το Mobile IP απαιτεί την ύπαρξη ενός κόμβου στο δίκτυο καλούμενο home agent. Όταν ο κινητός κόμβος δεν είναι συνδεδεμένος στο home δίκτυο (είναι δηλαδή σε

κάποιο ξένο δίκτυο) τότε το home agent λαμβάνει όλα τα πακέτα δεδομένων και κανονίζει να τα παραδώσει στον κινητό κόμβο στο σημείο στο οποίο έχει συνδεθεί.

Το μεγαλύτερο πλεονέκτημα του mobile IP είναι ότι διαχειρίζεται την κινητικότητα του χρήστη (mobility) στο επίπεδο του δικτύου και αφήνει ανέπαφα τα υπόλοιπα επίπεδα, επιτρέποντας έτσι την υπάρχουσα υποδομή δρομολόγησης, τους ακίνητους κόμβους, καθώς και τις υπάρχουσες εφαρμογές να μπορούν να λειτουργούν χωρίς καμία αλλαγή.

Το mobile IP είναι ένα πρωτόκολλο χτισμένο πάνω από το IP και το οποίο κάνει τη κινητικότητα (mobility) διαφανή στις εφαρμογές. Η κινητικότητα του χρήστη (mobility) μέσα στα υποδίκτυα (πχ. handovers μεταξύ Base Stations) θα χειρίζεται από το υποκείμενο δίκτυο.

Μειονεκτήματα του mobile IP είναι ο υψηλός φόρτος στο Home υποδίκτυο και η μεγάλη καθυστέρηση στην μεταδιδόμενη κίνηση στο κινητό κόμβο. Μπορεί φυσικά να αντιμετωπιστεί εάν στην ανταλλαγή των δεδομένων ο κόμβος που στέλνει τα δεδομένα μάθει την κινητή διεύθυνση (care-of address) του κινητού κόμβου και να στέλνει τα δεδομένα έτσι κατευθείαν σε αυτήν και όχι διαμέσου του home agent.

Με την χρησιμοποίηση του mobile IP δεν υπάρχει ανάγκη για διαμόρφωση του κινητού τερματικού με νέα IP διεύθυνση όταν κινείται από ένα δίκτυο σε άλλο. Αυτό είναι ένα από τα βασικά χαρακτηριστικά του VHE. Το τερματικό μπορεί να στέλνει και να λαμβάνει IP πακέτα δεδομένων ακόμα και μετά το μεταπομπή. Επίσης είναι δυνατή η ικανοποίηση της απαίτησης του VHE για παροχή υπηρεσιών χωρίς καμία διαφορά σε όλα τα δίκτυα.

IPv6

Το IPv6 είναι η επόμενη γενιά του IP πρωτοκόλλου και είναι γνωστό και ως IPng (Internet Protocol Next Generation) και καλύπτει τις ανεπάρκειες του IPv4, προσφέρει νέα βελτιωμένα χαρακτηριστικά όπως μεγαλύτερο μέγεθος και χώρο για διευθύνσεις και βελτιωμένα formats πακέτων. Θα καθιστά δυνατή την υψηλή απόδοση, την βαθμωτή λειτουργία μεταξύ των δικτύων και θα προσφέρει μια καλά δομημένη, αποδοτική, προσαρμοστική και ιεραρχική δρομολόγηση. Επίσης στις βασικές του προδιαγραφές περιλαμβάνεται κρυπτογράφηση και πιστοποίηση των πηγών εκπομπής δεδομένων. Τέλος θα παρέχει κίνηση σε πραγματικό χρόνο για πχ videoconference.

Το IPv6 υποστηρίζει κινητικότητα (Mobile IPv6 πρωτόκολλο) και επιλύει το πρόβλημα της διαφανούς δρομολόγησης των πακέτων δεδομένων προς και από τους κινητούς κόμβους όταν αυτοί δεν είναι στο οικείο δίκτυο.

Το πρωτόκολλο Mobile IPv6 είναι κατάλληλο για κινητικότητα τόσο σε ομογενή όσο και σε ετερογενή δίκτυα. Το IPv6 χρησιμοποιεί το IPsec και μπορεί διαμέσου του VHE να προσφερθούν Internet υπηρεσίες.

SIP (Session Initiation Protocol)

Χρησιμοποιείται για σηματοδότηση σε Internet multimedia conferencing, τηλεφωνία και διανομή πολυμέσων. Το SIP μπορεί να δημιουργήσει, να τροποποιήσει και να τερματίσει συνόδους μεταξύ δύο ή περισσότερων συμμετεχόντων. Οι χρήστες μπορεί να είναι είτε άνθρωποι ή ρομπότ που διαχειρίζονται συστήματα αποθήκευσης πολυμέσων.

Το SIP επιτρέπει την διαπραγμάτευση και την επιλογή τερματικού, επιτρέπει στους servers να παρέχουν IN υπηρεσίες, είναι ανεξάρτητο των πρωτοκόλλων μεταφοράς και τέλος επιτρέπει κινητικότητα των χρηστών με την εκπροσώπηση και την αποστολή των αιτήσεων στην νέα θέση του χρήστη.

Μειονεκτήματα του είναι το μεταβλητό μέγεθος των SIP μηνυμάτων που μερικές φορές μπορεί να υπερβούν την χωρητικότητα μερικών τερματικών καθώς και ότι θέλει ενοποίηση με άλλα πρωτόκολλα για να παρέχει μια ολοκληρωμένη υπηρεσία.

Το VHE μπορεί να θεωρηθεί σαν ένα καταναμημένο προφίλ χρήστη και το SIP μπορεί να χρησιμοποιηθεί για την διαπραγμάτευση και την διανομή του. Παρέχει στο VHE συμπληρωματικές πληροφορίες για τις υπηρεσίες καθώς και δυνατότητες σηματοδότησης για να δοθούν πληροφορίες που αφορούν τον χρήστη στο επισκεπτόμενο δίκτυο.

H.323

Το H.323 είναι ένα τυποποιημένο πρωτόκολλο από την ITU-T για επικοινωνίες πολυμέσων πάνω από δίκτυα και είναι μέρος από μια μεγάλη σειρά τυποποιημένων πρωτοκόλλων που καθιστούν ικανή την τηλεδιάσκεψη πάνω από WAN περιλαμβανομένου του Internet. Το πρότυπο περιλαμβάνει υποδομή για ήχο, video και επικοινωνία δεδομένων πάνω από IP.

Πλεονεκτήματα του H.323

- Το πρωτόκολλο H.323 είναι ευρέως διαδεδομένο και έχει καλά διαπιστευτήρια, πχ υποστήριξη πολλών εταιριών όπως Microsoft, Intel και Netscape.
- Είναι δοκιμασμένο και πιστοποιημένο. (το Microsoft Net Meeting το χρησιμοποιεί).
- Είναι ολοκληρωμένο: περιλαμβάνει έλεγχο κλήσεων, διαχείριση πολυμέσων και εύρους ζώνης από σημείο σε σημείο και από σημείο σε πολλά σημεία όπως επίσης και διεπαφές μεταξύ δικτύων με άλλα δίκτυα.
- Υποστηρίζει μια πληθώρα τερματικών.
- Η υποστήριξη των τερματικών περιλαμβάνει και τερματικά με διαφορετικές δυνατότητες.
- Είναι καλά αναβαθμισμένο σε PSTN πχ εμπορικά διαθέσιμα είναι τα PIGs (PSTN to Internet Gateway)
- Είναι σχεδιασμένο να αντισταθμίζει την καθυστέρηση των δικτύων, επιτρέποντας στους χρήστες να χρησιμοποιούν εφαρμογές πολυμέσων χωρίς να αλλάξουν την υποδομή του δικτύου τους.
- Παρέχει διαλειτουργικότητα μεταξύ συσκευών καθώς και μεταξύ εφαρμογών.
- Είναι ανεξάρτητο από το υλικό και το λειτουργικό σύστημα
- Το H.323 είναι έτοιμο προς χρήση

Μειονεκτήματα του H.323

- Είναι πολύπλοκο
- Η δρομολόγηση των κλήσεων είναι πολύπλοκη.
- Η εγκαθίδρυση των κλίσεων είναι επίπονη: υπάρχει συντελεστής καθυστέρησης περίπου έξι με επτά Round Trip Delays (RTDs) μόνο για τα μηνύματα σηματοδότησης. Στην ασύρματη επικοινωνία είναι σημαντικό να έχουμε όσο το δυνατό μικρότερη σηματοδότηση για διαφύλαξη του εύρους ζώνης.
- Δεν υπάρχουν εχέγγυα για QoS.
- Δυσκολία τυποποίησης
- Η επεξεργασία που απαιτείται από τα τερματικά είναι μεγάλη.
- Περιορισμός χρήσης φωνής. Όλα τα τερματικά θα πρέπει να υποστηρίζουν επικοινωνία φωνής: ενώ η υποστήριξη video και δεδομένων είναι προαιρετικά. Το VHE προβλέπει πιο διαφοροποιημένα τερματικά.
- Δυσκολία ελέγχου.

1.8. Απαιτήσεις Ασφάλειας και Πρωτόκολλα

1.8.1 Απαιτήσεις ασφάλειας στο VHE

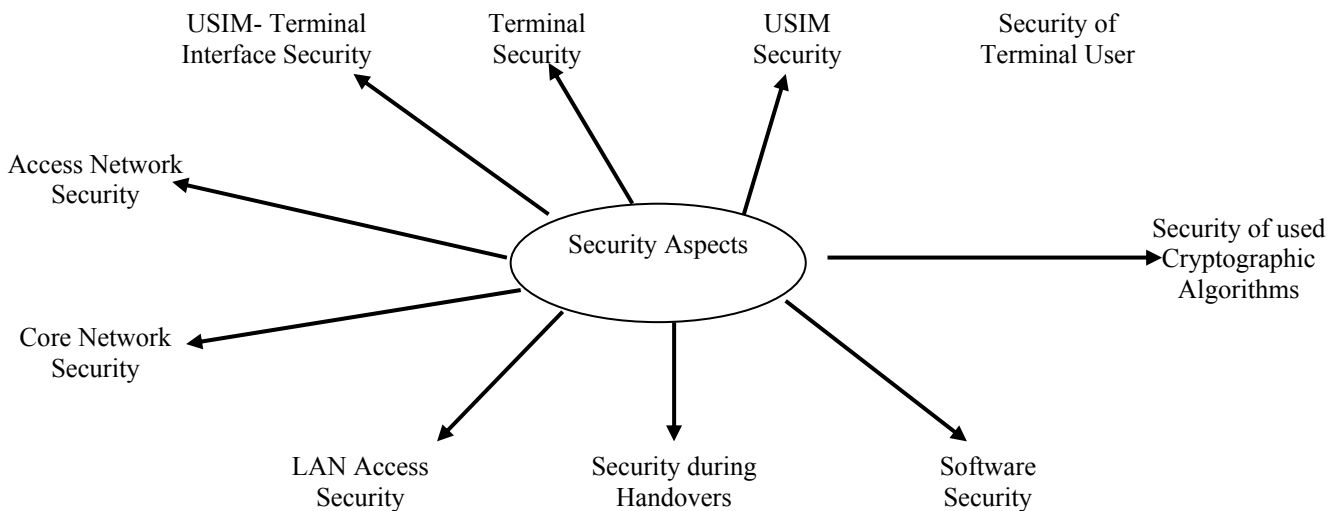
Η ασφάλεια είναι βασικό τμήμα της έννοιας του VHE, για να προσφέρονται στον χρήστη υπηρεσίες με ασφάλεια και με σωστή χρέωση. Οι παρούσες τεχνολογίες ασφάλειας είναι επαρκείς για να ικανοποιήσουν νέες υπηρεσίες όπως ηλεκτρονικό εμπόριο ή σχήματα πιστοποίησης.

Πρέπει να δηλώνεται στον χρήστη ποιες υπηρεσίες είναι ασφαλείς και ποιες όχι. Η ένδειξη αυτή πρέπει να είναι σε κάποια τυποποιημένη μορφή και θέση στην συσκευή του χρήστη.

Πέντε γενικά χαρακτηριστικά ασφάλειας πρέπει να παρέχονται:

- Εμπιστευτικότητα
- Ακεραιότητα
- Πιστοποίηση
- Μη άρνηση παροχής υπηρεσιών
- Διαθεσιμότητα

Όλα τα στοιχεία ασφάλειας που αναφέρονται παραπάνω καλούνται πυρήνες -κρατήρες (craters) ασφάλειας. Τα κύρια χαρακτηριστικά ασφαλείας του VHE παρουσιάζονται στο σχήμα 5 τα οποία και αναλύονται παρακάτω.



Σχήμα 5 Θέματα Ασφαλείας στο VHE

Ασφάλεια Τερματικών (Terminal Security)

Τα τερματικά των χρηστών θα πρέπει να είναι ασφαλή, το οποίο σημαίνει ότι όλο το υλικό και λογισμικό που είναι σημαντικό για την ασφάλεια πρέπει να προέρχεται από αξιόπιστη πηγή.

Ασφάλεια Κάρτας USIM (USIM Security)

Η USIM χρησιμοποιείται για να πιστοποιεί τον χρήστη και να αποθηκεύει δεδομένα και προγράμματα σχετικά με τον χρήστη. Θα πρέπει να είναι αδύνατο να εξαχθούν από την USIM τα προσωπικά κλειδιά. Η κρυπτογράφηση "δημόσιου κλειδιού" του GSM-SIM μπορεί να εφαρμοστεί. Επίσης διαπιστευμένα προγράμματα πρέπει να προφυλάσσονται από τροποποιήσεις.

Εάν η κίνηση σε διαφορετικά τερματικά επιτρέπεται (δηλαδή μετακινώντας την USIM από μια κινητή συσκευή σε ένα PC), η USIM θα πρέπει να περιέχει προσωπικά κλειδιά για την πιστοποίηση του τερματικού. Το επίπεδο ασφάλειας μπορεί να χρησιμοποιηθεί από τον παροχέα υπηρεσιών για να κάνει τις απαιτούμενες προσαρμογές στις προσφερόμενες υπηρεσίες.

Φυσική Ασφάλεια Διεπαφής USIM (USIM- Terminal interface security)

Η USIM πρέπει πάντα να έχει φυσική σύνδεση με την συσκευή δηλαδή δεν θα είναι δυνατό να την βγάλουμε ή να την αλλάξουμε κατά την διάρκεια μιας συνόδου.

Ασφάλεια Πρόσβασης (Access network security)

Οι διεπαφές των δικτύων και των PSTN πρέπει να προστατεύονται από μεγαλύτερου επιπέδου πρωτόκολλα ασφαλείας όπως IPsec με διαχείριση κλειδιού.

Ασφάλεια Δικτύων Κορμού (Core network security)

Τα δίκτυα κορμού IP και PSTN δεν θα πρέπει να θεωρούνται ασφαλή.

Ασφάλεια κατά τη Μεταπομπή σε Άλλο Δίκτυο (Security during handovers)

Όταν χρησιμοποιείται mobile IP σαν πρωτόκολλο, η προφανής λύση για την από άκρη σε άκρη ασφάλεια είναι το IPsec με ISAKMP (Internet Security Association and Key Management Protocol). Επίσης χαρακτηριστικά ασφαλείας των διαφόρων εφαρμογών θα μπορούν να χρησιμοποιηθούν.

Ασφάλεια Πρόσβασης σε Απομακρυσμένα δίκτυα (LAN access security)

Υπάρχουν δύο τουλάχιστον πρακτικοί τρόποι για την πρόσβαση σε απομακρυσμένο δίκτυο από ένα τερματικό: α) Χρησιμοποιώντας ασφαλές link μεταξύ GPRS και απομακρυσμένο δίκτυο ή β) με την χρήση IPsec με ISAKMP για να δημιουργήσουμε ένα Virtual Private Network (VPN) μεταξύ του τερματικού και του απομακρυσμένου δικτύου. Η πρώτη λύση απαιτεί συμφωνία μεταξύ των διαχειριστών του GPRS και των διαχειριστών του δικτύου.

Ασφάλεια Λογισμικού (Software security)

Κάποια φερέγγυα αρχή πρέπει να εγγυάται για το λογισμικό που θα φορτώνεται και χρησιμοποιείται.

Ασφάλεια Χρησιμοποιούμενων Αλγορίθμων Κρυπτογράφησης (Security of used cryptographic algorithms)

Όλα τα κρυπτογραφικά πρωτόκολλα πρέπει να είναι σχεδιασμένα έτσι ώστε να είναι ανεξάρτητα από τους αλγόριθμους που χρησιμοποιούνται. Αυτό σημαίνει ότι μπορούν να αλλαχθούν οι χαμηλού επιπέδου αλγόριθμοι μετέπειτα, εαν υπάρχει κάποιο πρόβλημα στην ασφάλεια. Μόνο τυποποιημένα και γνωστά πρωτόκολλα και αλγόριθμοι θα πρέπει να χρησιμοποιηθούν (όπως IPsec, ISAKMP, SHA, 3DES)

Ασφάλεια Τερματικού Χρήστη (Security of terminal user)

Η ασφάλεια του τερματικού του χρήστη σχετίζεται με τις λειτουργίες χρέωσης και τα δικαιώματα πρόσβασης στις υπηρεσίες του VHE. Ανάλογες συγκλίσεις- συμβάσεις μεταξύ των GSM, GPRS και UMTS πρέπει να γίνουν. Η πρόσβαση στις υπηρεσίες θα πρέπει να επιτρέπεται στον χρήστη σύμφωνα με την ταυτότητα του (USIM) και σύμφωνα με το επίπεδο ασφαλείας του τερματικού του. Ιδιαίτερη προσοχή πρέπει να δοθεί στην ασφάλεια που πρέπει να έχουν οι ενημερώσεις του VHE.

1.8.2 Ανάλυση των πρωτοκόλλων ασφαλείας

Παραπάνω αναλύθηκαν οι απαιτήσεις ασφαλείας για το VHE οι οποίες θα πρέπει να εφαρμοστούν στην επιλογή των πρωτοκόλλων για να προστατευθούν τα δεδομένα στα UMTS δίκτυα, οι διεπαφές των δικτύων αυτών και των τερματικών που θα χρησιμοποιούνται.

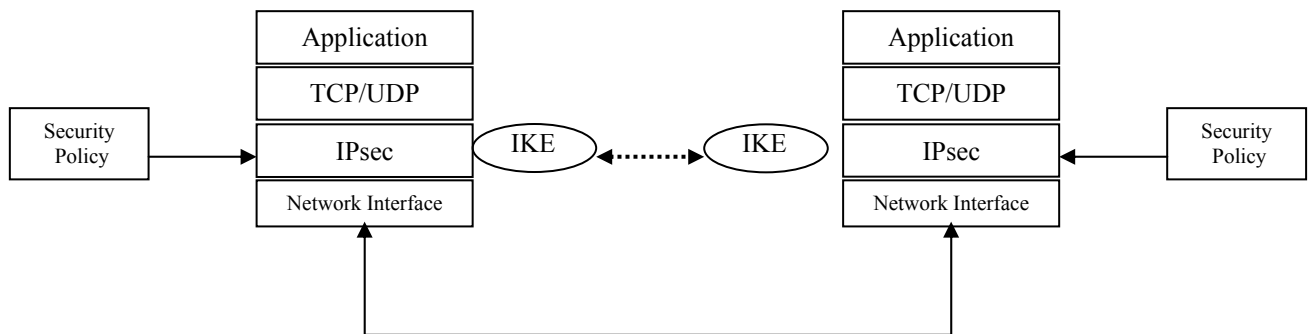
Η απόδοση αυτών των πρωτοκόλλων μπορεί να μετρηθεί από την ασφάλεια των αλγορίθμων κρυπτογράφησης, υπολογιστικής ευελιξίας και της διαθεσιμότητάς τους. Οι αλγόριθμοι αυτοί πρέπει να είναι συμβατοί με τα υψηλότερα και χαμηλότερα επίπεδα πρωτοκόλλων όπου εφαρμόζονται.

Παρακάτω αναλύονται συνοπτικά τα προτεινόμενα πρωτόκολλα ασφαλείας για χρήση στο περιβάλλον του VHE.

IPsec

Το IPsec είναι ένα πρωτόκολλο ασφαλείας που βασίζεται στο IP για την παροχή ασφαλών επικοινωνιών μεταξύ δύο IP κόμβων. Απαιτείται υποστήριξη λειτουργιών ασφαλείας και στους δύο κόμβους στην στοίβα του IP. Η ασφάλεια παρέχεται με την πιστοποίηση και την κρυπτασφάλιση των πακέτων που εκπέμπονται. Το IPsec χρησιμοποιεί συμμετρική κρυπτογράφηση και απαιτεί την χρήση του ίδιου κλειδιού κρυπτασφάλισης και πιστοποίησης και στα δύο άκρα. Κλιμακούμενο πρωτόκολλο διαχείρισης των κλειδιών σαν το IKE χρησιμοποιείται για να δημιουργεί συμμετρικά κλειδιά για την στοίβα του IPsec.

Το IPsec παρέχει μια ασφαλή πλατφόρμα για τις εφαρμογές που τρέχουν πάνω από αυτό και η λειτουργία του είναι εύληπτη και ξεκάθαρη. Η επικεφαλίδα Authentication Header (AH) χρησιμοποιείται για να παρέχει connectionless πιστοποίηση για την ακεραιότητα και πηγή των δεδομένων για IP datagrams και προαιρετικά να παρέχει προστασία από επαναλήψεις. Το Encapsulating Security Payload (ESP) χρησιμοποιείται για να παρέχει μερικές από τις ακόλουθες υπηρεσίες: εμπιστευτικότητα, προέλευση δεδομένων, πιστοποίηση, προστασία από επαναλήψεις κλπ. Η επικεφαλίδα AH προστίθεται μετά την IP επικεφαλίδα και πριν από οποιαδήποτε δεδομένα ανωτέρων επιπέδων στο IP datagram. Στο σχήμα 6 φαίνεται η θέση του IPsec σε μια τυπική στοίβα πρωτοκόλλων.



Σχήμα 6 Θέση του IPsec σε μια τυπική στοίβα πρωτοκόλλων

Καταλληλότητα του IPsec για το VHE

Το IPsec υποστηρίζει την έννοια του VHE με την προσφορά μιας τυποποιημένης πλατφόρμας, πάνω από το IP, στις εφαρμογές για να επικοινωνούν με ασφάλεια. Το IPsec μπορεί να εφαρμοστεί σε οποιοδήποτε κινητό κόμβο, σταθερό κόμβο ή δρομολογητή που υποστηρίζει το πρωτόκολλο. Το IPsec θα είναι ιδιαίτερα χρήσιμο όταν εφαρμόζεται σε ένα κινητό κόμβο που υποστηρίζει mobile IP και IPsec.

Εφαρμογές του IPsec για το IPv4 που είναι σήμερα εμπορικά διαθέσιμες για πολλές πλατφόρμες.

Ένας περιορισμός του IPsec είναι η μη ύπαρξη πρωτοκόλλου για ασφαλή multicast κίνηση.

IKE

Το IKE είναι πρωτόκολλο διαχείρισης κλειδιών, που παρέχει δηλαδή κλειδιά συνόδων για το IPsec. Το πρωτόκολλο είναι σχεδιασμένο και δοκιμασμένο από την IETF όπως το IPsec, ενώ το επίπεδο ασφαλείας του εξαρτάται από τους επιλεγόμενους αλγόριθμους και τα μήκη των κλειδιών που έχουν τεθεί από τις ιδιότητες του Host. Το IKE συνήθως χρησιμοποιεί κρυπτογράφηση δημόσιου κλειδιού για να πιστοποιεί τους συνδιαλαγόμενους κόμβους.

Πρέπει να σημειωθεί ότι το IKE απαιτεί περισσότερη επεξεργαστική δύναμη από το IPsec λόγω των δαπανηρών σε επεξεργασία δημοσίων κλειδιών. Η συνολική καθυστέρηση (η εγκαθίδρυση μιας σύνδεσης IPsec) είναι το άθροισμα του χρόνου των λειτουργιών των δημοσίων κλειδιών και ο χρόνος της καθυστέρησης Round-Trip που απαιτείται από το IKE για

την ανταλλαγή μηνυμάτων. Η παρατηρούμενη καθυστέρηση είναι της τάξης του 0.5 δευτερόλεπτου έως και μερικά δευτερόλεπτα εξαρτώμενης πάντα από το επιλεγμένο mode για την επεξεργασία IKE και την καθυστέρηση πηγαινε -έλα μεταξύ των δύο επικοινωνούντων Host.

WTLS (Wireless Transport Layer Security)

Το WTLS είναι ένα πρωτόκολλο που παρέχει ασφάλεια σε WAP εφαρμογές. Έχει προέλθει από το TLS και βασίζεται στο SSL 3.0 και παρέχει περίπου την ίδια λειτουργία με το IPsec και IKE.

Παρόλα αυτά το πρωτόκολλο είναι στενά συνδεδεμένο με το περιβάλλον του WAP και έτσι είναι φτιαγμένο για μεταφορά μικρών ταχυτήτων και αργών αλληλεπιδράσεων. Επίσης λαμβάνεται υπόψη η περιορισμένη μνήμη των κινητών τερματικών με την ελαχιστοποίηση των εφαρμοζόμενων αλγορίθμων και την επιλογή μικρού μεγέθους αλγορίθμων

X.509

Κάθε σύστημα που χρησιμοποιεί δημόσια κλειδιά κρυπτασφάλισης για την πιστοποίηση των διαφόρων οντοτήτων απαιτεί την διαθεσιμότητα των κλειδιών αυτών με ένα ασφαλή τρόπο. Αυτή η διαθεσιμότητα συνήθως παρέχεται χρησιμοποιώντας τα τυποποιημένα πιστοποιητικά X.509 της ITU. Ένα πιστοποιητικό X.509 χρησιμοποιείται μαζί με την "ταυτότητα" της οντότητας και το αντίστοιχο δημόσιο κλειδί της οντότητας.

Η X.509v3 είναι η τελευταία έκδοση του πρωτοκόλλου.

ΜΕΡΟΣ Β'

ΣΥΣΤΗΜΑΤΑ ΠΑΓΚΟΣΜΙΟΥ ΔΟΡΥΦΟΡΙΚΟΥ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ (GPS)

2.1 Γενικά.

Το Global Positioning System (GPS) αποτελεί το νεότερο σύστημα προσδιορισμού της θέσης σημείων στη γήινη επιφάνεια με την βοήθεια τεχνητών δορυφόρων. Παρέχει μετά από κατάλληλη επεξεργασία μέσω ηλεκτρονικού υπολογιστή, τη θέση σημείων σε καρτεσιανές συντεταγμένες στο γεωκεντρικό σύστημα αναφοράς των δορυφόρων.

Το δορυφορικό GPS σχεδιάστηκε και αναπτύσσεται από τις ΗΠΑ για να ικανοποιήσει κυρίως στρατιωτικές ανάγκες ναυσιπλοΐας. Το GPS αποτελείται από τρία λειτουργικά τμήματα. Το τμήμα του διαστήματος, το οποίο αποτελείται από τους δορυφόρους NAVSTAR, που εκπέμπουν σήματα και κωδικοποιημένες πληροφορίες στην περιοχή των μικροκυμάτων (L-Band). Το επίγειο τμήμα ελέγχου, το οποίο παρακολουθεί και κατευθύνει τη λειτουργία κάθε δορυφόρου του συστήματος. Τέλος το τμήμα χρηστών που παρακολουθεί τα σήματα των δορυφόρων και αποτελείται από όλους τους χρήστες που μπορούν να εκμεταλλευθούν τις δυνατότητες του συστήματος για ναυσιπλοΐα, γεωδαισία και άλλες χρήσεις.

2.2 Περιγραφή των λειτουργικών τμημάτων του συστήματος GPS.

Το τμήμα του διαστήματος αποτελείται από 24 δορυφόρους, από τους οποίους οι 21 είναι επιχειρησιακοί και 3 είναι εφεδρικοί. Οι δορυφόροι αυτοί βρίσκονται σε έξι όμοια κατανεμημένα τροχιακά επίπεδα. Σε κάθε τροχιακό επίπεδο λειτουργούν 4 δορυφόροι, οι οποίοι περιστρέφονται σε σχεδόν κυκλικές και έχουν γωνία κλίσης 55 μοίρες με τον ισημερινό με περίοδο ολοκλήρωσης μιας περιστροφής γύρω από την Γη 12 ώρες. Το ύψος της τροχιάς είναι περίπου 20.200 χιλιόμετρα. Ο δορυφορικός αυτός σχηματισμός εξασφαλίζει δυνατότητες ορατότητας από τους χρήστες οποιαδήποτε χρονική στιγμή με καλή γεωμετρική διάταξη 5 μοιρών ή και ψηλότερα από τον ορίζοντα, σε οποιαδήποτε θέση πάνω από την Γη.

Οι δορυφόροι εκπέμπουν κωδικοποιημένο σήμα σε δύο φέρουσες συχνότητες, την L_1 και L_2 που είναι $154 f_0$ και $120 f_0$ αντίστοιχα, όπου f_0 είναι η βασική συχνότητα και είναι ίση με 10,23MHz. Το μήκος κύματος της L_1 είναι περίπου 20 cm ενώ της L_2 είναι περίπου 25 cm. Στα φέροντα κύματα προστίθενται σαν δυαδικές διαμορφώσεις δύο φάσεων οι κώδικες ψευδοτυχαίου θορύβου P και C/A καθώς και ένα δυαδικό μήνυμα 1500 ψηφίων. Ο κώδικας P μεταδίδεται στην L_1 και στην L_2 . Η συχνότητα του είναι 10.23 MHz και το μήκος κύματος του είναι 29,31 m. Ο κώδικας αυτός παράγεται με την μέθοδο αλγορίθμου που επαναλαμβάνεται κάθε 266,4 ημέρες. Σε κάθε δορυφόρο αντιστοιχίζεται ένα εβδομαδιαίο μοναδικό τμήμα από τον κώδικα, χωρίς να υπάρχει επικάλυψη με τους άλλους δορυφόρους. Κάθε μεσάνυχτα σε χρόνο του συστήματος GPS από το Σάββατο προς την Κυριακή, ο κώδικας αρχίζει να μεταδίδεται από την αρχή. Ο κώδικας C/A μεταδίδεται μόνο από την L_1 και επαναλαμβάνεται κάθε 1msec, έχει συχνότητα 1,023MHz και μήκος κύματος 293,1 m. Το δυαδικό μήνυμα, μεταδίδεται και στις δύο συχνότητες L_1 και L_2 . Έχει συχνότητα $f_0/204.600$, μήκος κύματος 6 Mm, η διάρκεια του είναι 30 sec και διαιρείται σε πέντε τμήματα διάρκειας 6 sec το καθένα. Τα δεδομένα του μηνύματος περιλαμβάνουν πληροφορία για τη λειτουργία του δορυφόρου, πληροφορία χρόνου για τη μεταφορά της παρακολούθησης του σήματος από τον C/A κώδικα στον P κώδικα, τις παραμέτρους για τον υπολογισμό της διόρθωσης χρονομέτρου, τα τροχιακά στοιχεία του δορυφόρου (εφημερίδα του δορυφόρου) και τις διορθώσεις για την καθυστέρηση διάδοσης του σήματος λόγω ιονόσφαιρας. Επιπλέον περιέχουν πληροφορίες για τις λιγότερο ακριβείς τροχιές

και την κατάσταση λειτουργίας όλων των υπόλοιπων δορυφόρων στο σχηματισμό, που χρειάζεται για τον προγραμματισμό της παρακολούθησης και των άλλων δορυφόρων, ενώ έχει προβλεφθεί χώρος για την διαμόρφωση και εκπομπή ειδικών μηνυμάτων.

Το τμήμα ελέγχου αποτελείται από πέντε σταθμούς παρακολούθησης (5 Monitor Stations) τον κεντρικό σταθμό ελέγχου (Master Control Station) καθώς και τρεις σταθμούς επικοινωνίας (3 Upload Ground Stations). Κύριος σκοπός αυτού του τμήματος είναι να παρακολουθεί συχνά τους δορυφόρους, να ελέγχει την καλή λειτουργία τους, να προσδιορίζει την τροχιά τους και να ενημερώνει τις πληροφορίες που εκπέμπουν οι δορυφόροι.

Το τμήμα των χρηστών αποτελείται από όλους τους χρήστες, οι οποίοι με την χρήση κατάλληλων δεκτών και με μετρήσεις προς τους δορυφόρους σε συνδυασμό με τα στοιχεία των εφημερίδων μπορούν οποιαδήποτε χρονική στιγμή να προσδιορίσουν με ακρίβεια τη θέση, την ταχύτητα και την κατάσταση του χρονομέτρου του δέκτη.

2.3 Δέκτες

Οι μετρήσεις στο σύστημα GPS είτε είναι μετρήσεις φάσης φέροντος κύματος είτε μετρήσεις κώδικα, γίνονται με την βοήθεια των δεκτών. Ένας δέκτης αποτελείται από ένα ή περισσότερα κανάλια. Το κανάλι ενός δέκτη αποτελείται από την ραδιοσυχνότητα, τα ψηφιακά ηλεκτρονικά και το πρόγραμμα, που απαιτούνται ώστε ο δέκτης να παρακολουθεί το σήμα του δορυφόρου GPS σε μια από τις δύο φέρουσες συχνότητες.

Κάθε κανάλι μπορεί να είναι είτε τετραγωνισμού είτε συσχέτισης. Το κανάλι τετραγωνισμού πολλαπλασιάζει το λαμβανόμενο σήμα για να παράγει μια δεύτερη αρμονική του φέροντος κύματος που δεν έχει τον κώδικα, ενώ το κανάλι της συσχέτισης χρησιμοποιεί ένα κύκλωμα εγκλωβισμού καθυστέρησης για να διατηρήσει μια ευθυγράμμιση μεταξύ του αντιγράφου του κώδικα του GPS που παράγεται στον δέκτη και του εισερχόμενου κώδικα.

2.4 Αρχή Εντοπισμού- Τρόποι Μέτρησης

Η θέση του παρατηρητή προσδιορίζεται με μέτρηση των αποστάσεων προς τέσσερις δορυφόρους, με την ευθυγράμμιση του κώδικα που εκπέμπεται από τον δορυφόρο και τον κώδικα που παράγεται στο δέκτη. Το γεγονός ότι δεν αρκούν αποστάσεις από τρεις δορυφόρους για τον προσδιορισμό της θέσης X,Y,Z οφείλεται στο ότι το χρονόμετρο του παρατηρητή δεν είναι ιδιαίτερα ακριβές και κατά συνέπεια η απόσταση που προσδιορίζεται δεν είναι σωστή. Γι' αυτό και θα πρέπει το σφάλμα του χρονομέτρου να θεωρηθεί σαν συμπληρωματικός άγνωστος.

Μετρήσεις στο σύστημα GPS μπορούν να γίνουν με δύο τρόπους, είτε με μετρήσεις φάσης του φέροντος σήματος, είτε με την χρήση του κώδικα. Στις μετρήσεις κώδικα μετρώνται ψευδοαποστάσεις, ενώ στις μετρήσεις φέροντος σήματος μετριέται η φάση του σήματος.

Ψευδοαπόσταση είναι η διαφορά του χρόνου που απαιτείται για να ευθυγραμμιστεί (συσχετιστεί) ένα αντίγραφο του κώδικα του GPS το οποίο παράγεται στον δέκτη, με το εισερχόμενο σήμα του GPS, που μετατρέπεται σε απόσταση αφού πρώτα πολλαπλασιαστεί με την ταχύτητα του φωτός.

Η μέτρηση φάσης φέροντος σήματος είναι η διαφορά μεταξύ της φάσης του εισερχόμενου στον δέκτη φέροντος κύματος του GPS και της φάσης μιας ονομαστικά σταθερής συχνότητας που παράγεται στον δέκτη. Μπορεί να συσχετιστεί με την ψευδοαπόσταση, μόνο αφού προσδιοριστεί η ασάφεια της φάσης.

Ασάφεια φάσης είναι η αβεβαιότητα της αρχικής μέτρησης που εισάγεται σαν συστηματικό σφάλμα σε όλες τις μετρήσεις και μέχρι κάποια διακοπή. Πιο αναλυτικά, σε κάθε στιγμή μετριέται η φάση του φέροντος κύματος και στην μέτρηση αυτή προστίθεται η αρχική ακτίνα δηλαδή η ασάφεια φάσης.

2.5 Τρόποι Εντοπισμού

2.5.1 Απόλυτος Εντοπισμός

Ο απόλυτος εντοπισμός αφορά τον προσδιορισμό των συντεταγμένων ενός μόνο δέκτη από μια σειρά μετρήσεων. Μπορεί να είναι είτε στατικός είτε κινηματικός.

Ο στατικός εντοπισμός αναφέρεται σε περιόδους παρατηρήσεων μιας ή και περισσότερων ωρών για τις οποίες η κίνηση του δέκτη είναι ασήμαντη, ενώ ο κινηματικός υπονοεί σημαντική κίνηση του δέκτη σε ένα μικρό τμήμα της περιόδου ενός δορυφόρου GPS, που είναι δώδεκα ώρες.

Οι συντεταγμένες θέσης, τόσο στον στατικό όσο και στον κινηματικό εντοπισμό, μπορούν να υπολογιστούν είτε σε πραγματικό χρόνο, δηλαδή κατά την διάρκεια των μετρήσεων, είτε εκ των υστέρων.

Οι ακρίβειες, οι οποίες επιτυγχάνονται στον απόλυτο εντοπισμό θέσης μετά από αρκετή ώρα παρατηρήσεων με την χρήση ενός δέκτη ο οποίος μετράει σε κώδικα, είναι της τάξης των 25 μέτρων εάν οι μετρήσεις γίνονται στον C/A κώδικα, ενώ αν οι μετρήσεις γίνονται στον P ακρίβεια γίνεται 10 μέτρα περίπου

2.5.2 Σχετικός Εντοπισμός

Όπως φαίνεται από τα παραπάνω, ο απόλυτος εντοπισμός θέσης δεν καλύπτει τις απαιτήσεις ακρίβειας των τοπογραφικών και γεωδαιτικών εφαρμογών. Γι' αυτό το λόγο χρησιμοποιείται ο σχετικός εντοπισμός θέσης, ο οποίος αναφέρεται στον προσδιορισμό των σχετικών συντεταγμένων (ΔX , ΔY , ΔZ) μεταξύ δύο ή περισσότερων δεκτών που παρακολουθούν τους ίδιους δορυφόρους.

Ο στατικός σχετικός εντοπισμός θέσης αφορά τον προσδιορισμό των σχετικών θέσεων ενός δικτύου σημείων. Απαιτείται η χρήση τουλάχιστον δύο δεκτών, οι οποίοι να μετράνε φάση φέροντος κύματος είτε στην L_1 είτε στην L_2 (σε περίπτωση που η βάση είναι μεγαλύτερη από τα 20 χιλιόμετρα τότε πρέπει οι δέκτες να μετράνε και στις δύο συχνότητες) ταυτόχρονα προς τουλάχιστον τέσσερις δορυφόρους. Η διάρκεια των παρατηρήσεων κυμαίνεται από 45 λεπτά και πάνω, ανάλογα με την απόσταση των δεκτών και το πλήθος των δορυφόρων που είναι διαθέσιμοι κατά την διάρκεια των μετρήσεων.

Ο υπολογισμός των διανυσμάτων βάσης μεταξύ των σημείων γίνεται εκ των υστέρων. Η ακρίβεια που πετυχαίνεται στον στατικό σχετικό εντοπισμό θέσης μπορεί να φθάσει και το 1ppm.

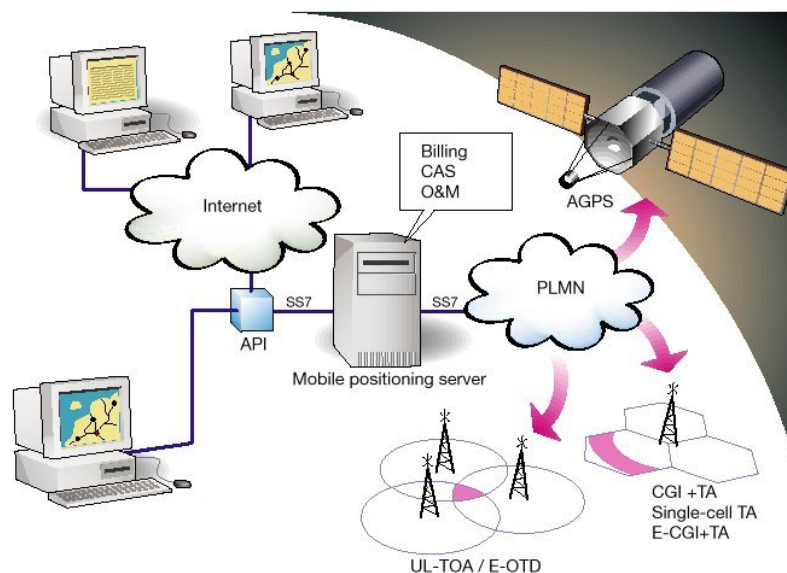
Κατά τον κινηματικό σχετικό εντοπισμό με έναν ή περισσότερους δέκτες παρακολούθησης σε γνωστές ακίνητες θέσεις, προσδιορίζεται η θέση ενός ή περισσότερων δεκτών σε κίνηση. Οι δέκτες μετρούν φάσεις στην L_1 ή την L_2 και ο υπολογισμός των θέσεων γίνεται εκ των υστέρων. Η ακρίβεια που πετυχαίνεται μπορεί να φθάσει τα μερικά εκατοστά.

Ο κινηματικός σχετικός εντοπισμός θέσης σε πραγματικό χρόνο απαιτεί επιπλέον να υπάρχει κάποιος βασικός σταθμός, ο οποίος να εκπέμπει προς όλους τους κινητούς δέκτες ταυτόχρονα με τα σήματα των δορυφόρων και τις διορθώσεις. Στην περίπτωση αυτή του κινηματικού σχετικού εντοπισμού θέσης σε πραγματικό χρόνο γίνονται μετρήσεις κώδικα και η ακρίβεια που πετυχαίνεται κυμαίνεται από 2 έως 5 μέτρα.

ΛΥΣΕΙΣ ΚΙΝΗΤΗΣ ΘΕΣΗΣ (MOBILE LOCATION SOLUTION MLS SOFTWARE)

3.1 Εισαγωγή

Ο όρος Mobile Location Solution (MLS) είναι ορολογία που χρησιμοποιείται από την Ericsson για ένα σύστημα εντοπισμού θέσης, περιλαμβάνοντας εφαρμογές, με το οποίο καθορίζεται η γεωγραφική θέση των συνδρομητών των εταιριών κινητής τηλεφωνίας και η παροχή σε αυτούς με σχετικές πληροφορίες και υπηρεσίες (Σχήμα 7).



Σχήμα 7. Το σύστημα MLS σχεδιάστηκε για να χειρίζεται μια μεγάλη ποικιλία τρόπων εντοπισμού και διεπαφών εφαρμογών.

Το σκεπτικό για την υλοποίηση του συστήματος αυτού είναι ο συνδυασμός των μηχανισμών προσδιορισμού θέσης με πληροφορίες σχετικές με την γεωγραφική θέση του συνδρομητή, πετυχαίνοντας έτσι την παροχή προσωπικών υπηρεσιών διαμέσου των κινητών τηλεφώνων ή οποιονδήποτε κινητών μονάδων τηλεπικοινωνίας.

Στις ΗΠΑ, νομικά θέματα επέδρασαν σαν κινητήριες δυνάμεις πίσω από τον καθορισμό του προτύπου, για το σύστημα προσδιορισμού θέσης του GSM. Ο κύριος πρωταγωνιστής σε αυτή την αγορά είναι η Federal Communication Commission (FCC). Οι υπηρεσίες και η βιομηχανία συμφώνησαν για το μέρος και τα χαρακτηριστικά του συστήματος προσδιορισμού θέσης του GSM το οποίο θα χρησιμοποιηθεί στην αγορά. Οι απαιτήσεις δεν έχουν ακόμα τελειώσει, μα ένα διάγραμμα υλοποίησης και κάποια χαρακτηριστικά έχουν περιγραφεί. Οι κανονισμοί της FCC διαφέρουν μεταξύ των λύσεων που βασίζονται σε τερματικά και σε αυτές που βασίζονται σε δίκτυα. Οι λύσεις που βασίζονται σε τερματικά συνδέονται με τις πληροφορίες θέσης που είναι αποθηκευμένες στο τερματικό ή στην κάρτα SIM. Αυτοί οι μηχανισμοί εντοπισμού θέσης απαιτούν ένα νέο τερματικό, μια νέα κάρτα SIM ή και τα δύο.

Στην πραγματικότητα, αυτό σημαίνει ότι όταν το σύστημα εγκατασταθεί θα πρέπει όλοι οι συνδρομητές να αντικαταστήσουν τις συσκευές ή την κάρτα SIM τους για να χρησιμοποιήσουν τις νέες υπηρεσίες. Η εισχώρηση στην αγορά θα μεγαλώσει με την αντικατάσταση των κινητών συσκευών και των καρτών SIM μετά από μια περίοδο τεσσάρων με πέντε χρόνων. Παραδείγματα από λύσεις βασισμένες σε τερματικά είναι το βοηθούμενο από δίκτυο παγκόσμιο σύστημα εντοπισμού (Network Assisted Global Positioning System , A-GPS), SIM toolkit, και την αναβαθμισμένη παρατηρούμενη διαφορά χρόνου (Enhanced Observed Time Difference, E-OTD).

Σε αντίθεση με τα παραπάνω, οι μηχανισμοί εντοπισμού θέσης που βασίζονται σε δίκτυα δεν απαιτούν πληροφορίες εντοπισμού να υπάρχουν στην κινητή συσκευή, το οποίο φυσικά σημαίνει 100% εισχώρηση στην αγορά. Παραδείγματα μηχανισμών εντοπισμού θέσης που βασίζονται σε δίκτυα είναι η μέθοδος της παγκόσμιας ταυτότητας κελιού και καθυστέρησης χρόνου μετάδοσης (Cell Global Identity and Timing Advance, CGI-TA) και η μέθοδος μέτρησης του χρόνου άφιξης (Uplink Time Of Arrival, UL-TOA). Επειδή τα συστήματα εντοπισμού θέσης που βασίζονται σε τερματικά και αυτά που βασίζονται σε δίκτυα έχουν διαφορετικά χαρακτηριστικά, η FCC διατύπωσε διαφορετικές απαιτήσεις για το καθένα. Σήμερα αυτά που ικανοποιούν τις απαιτήσεις αυτές είναι το A-GPS για το πρώτο σύστημα ενώ η μέθοδος UL-TOA για το δεύτερο. Παρακάτω αναλύονται οι μέθοδοι και των δύο προσεγγίσεων.

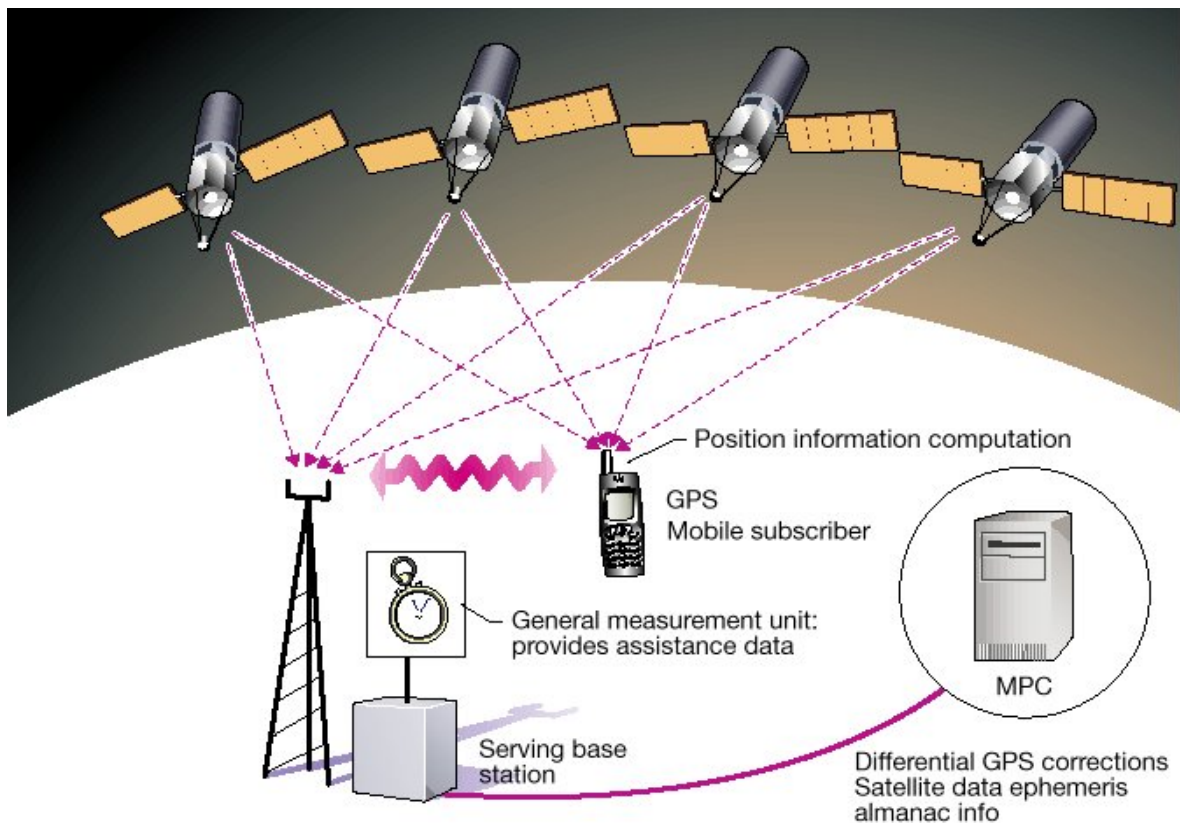
3.2 Συστήματα εντοπισμού θέσης που βασίζονται σε τερματικά

network Assisted Global Positioning System (A-GPS)

Το σύστημα παγκόσμιου εντοπισμού θέσης (Global Positioning System-GPS) χρησιμοποιείται κυρίως για πλοήγηση. Συσκευές GPS υπάρχουν σε αυτοκίνητα, πλοία και αεροπλάνα. Μια συσκευή GPS λαμβάνει σήμα από τέσσερις ή περισσότερους δορυφόρους. Κάθε σήμα περιέχει μια χρονοσφραγίδα και μια περιγραφή για την θέση του δορυφόρου. Συγκρίνοντας αυτή την πληροφορία ο δέκτης GPS μπορεί να υπολογίσει την δική του θέση (αναλυτικότερα στο κεφάλαιο του GPS). Τα κύρια μειονεκτήματα του GPS είναι ότι τα σήματα των δορυφόρων είναι σχετικά ασθενή και μπορεί μερικές φορές να μην παρέχουν επαρκή κάλυψη σε κάποια περιβάλλοντα (αστικές περιοχές ή σε ανώμαλη γήινη επιφάνεια). Παρόλα αυτά, το δίκτυο του GSM μπορεί να παράσχει βοηθητική πληροφορία το οποίο δίνει ολοκληρωμένους δέκτες GPS, καλύτερη κάλυψη από το μόνο του GPS σύστημα. Οι επιπλέον δυνατότητες φαίνονται στον παρακάτω πίνακα.

Τύπος βοήθειας	Κέρδος – Νέες δυνατότητες
Δορυφορικές πληροφορίες	Βελτιώνει τον χρόνο που πρέπει να υπολογιστεί και την ευαισθησία ή και τα δύο. Το πρώτο πετυχαίνεται ελαττώνοντας την ανάγκη για αποδιαμόρφωση των μηνυμάτων πλοήγησης.
Ακρίβεια Συχνότητας	Βελτιώνει τον χρόνο που πρέπει να υπολογιστεί
Εκτίμηση θέσης	Αρχικοποιεί την διαδικασία υπολογισμού θέσης και βελτιώνει την λήψη δευτερευόντων και επακόλουθων σημάτων
Διαφορική διόρθωση στις μετρήσεις GPS	Βελτιώνει την ακρίβεια των εκτιμήσεων θέσης(10 με 20 μέτρα)
Αναφορά χρόνου	Βελτιώνει τον χρόνο που πρέπει να υπολογιστεί για όλους τους δέκτες. Βελτιώνει την ευαισθησία για δέκτες σε φτωχό περιβάλλον σήματος.

Διαφορετικά είδη μονάδων μέτρησης θέσης (Location Measurement Units, LMU) χρησιμοποιούνται για την συλλογή βοηθητικών δεδομένων. Προκειμένου να παρασχεθούν δορυφορικές πληροφορίες καθώς και διαφορική διόρθωση GPS , κάθε LMU πρέπει κάθε να αναπυχθεί κάθε **300 Km μέσα στο** δίκτυο. Αυτή η αναβάθμιση προσφέρει ακρίβεια μέχρι 10 ως 20 μέτρα. Για την περαιτέρω αύξηση της κάλυψης του GPS μια πιο ακριβής αναφορά χρόνου πρέπει να χρησιμοποιηθεί πράγμα που σημαίνει την ανάπτυξη μιας LMU κάθε περίπου τρίτο BTS (Σχήμα 8).



Σχήμα 8. Διάγραμμα της μεθόδου Advanced GPS (A-GPS)

Enhanced Observed Time Difference (E-OTD)

Η μέθοδος της αναβαθμισμένης παρατηρούμενης διαφοράς χρόνου, βασίζεται στην μέτρηση OTD μεταξύ των αφίξεων των ριπών από παρακείμενα ζεύγη ληπτών-εκπομπών που υπάρχουν στους σταθμούς βάσης. Η κινητή συσκευή μετρά την OTD καθώς επίσης κανονικές ριπές, dummy και συγχρονισμού. Επειδή τα πακέτα εκπομπής του BTS δεν συγχρονίζονται θα πρέπει να μετράται και η σχετική διαφορά χρόνου (Relative Time Difference, RTD). Για να επιτύχουμε ακριβή τριγωνισμό πρέπει να κάνουμε μετρήσεις για το OTD και RTD από τρία τουλάχιστον διαφορετικά ζεύγη δεκτών-εκπομπών που βρίσκονται σε διαφορετικά γεωγραφικά σημεία. Βασιζόμενοι στις μετρήσεις των τιμών της OTD, η γεωγραφική θέση του τερματικού μπορεί να υπολογιστεί μέσα στο δίκτυο είτε από το ίδιο το κινητό τερματικό με την προϋπόθεση ότι έχει διαθέσιμες όλες τις απαιτούμενες πληροφορίες. Η μέθοδος E-OTD μπορεί να είναι είτε:

- Βοηθούμενη από το δίκτυο, όπου στην περίπτωση αυτή το κινητό τερματικό μετρά το OTD σήμα και υπολογίζει την γεωγραφική του θέση (για να το κάνει αυτό, το δίκτυο πρέπει να παράσχει πρόσθετες πληροφορίες, όπως τις συντεταγμένες των BTS από όπου γίνονται οι μετρήσεις και τις τιμές των μετρήσεων RTD)
- Βοηθούμενη από την κινητή συσκευή, όπου στην περίπτωση αυτή η κινητή συσκευή μετρά το OTD σήμα και αναφέρει τις μετρήσεις στο δίκτυο, το οποίο τότε υπολογίζει την θέση της κινητής συσκευής. Η ακρίβεια είναι της τάξης των 60 μέτρων σε κατοικημένες περιοχές και 200 μέτρα σε αραιοκατοικημένες περιοχές.

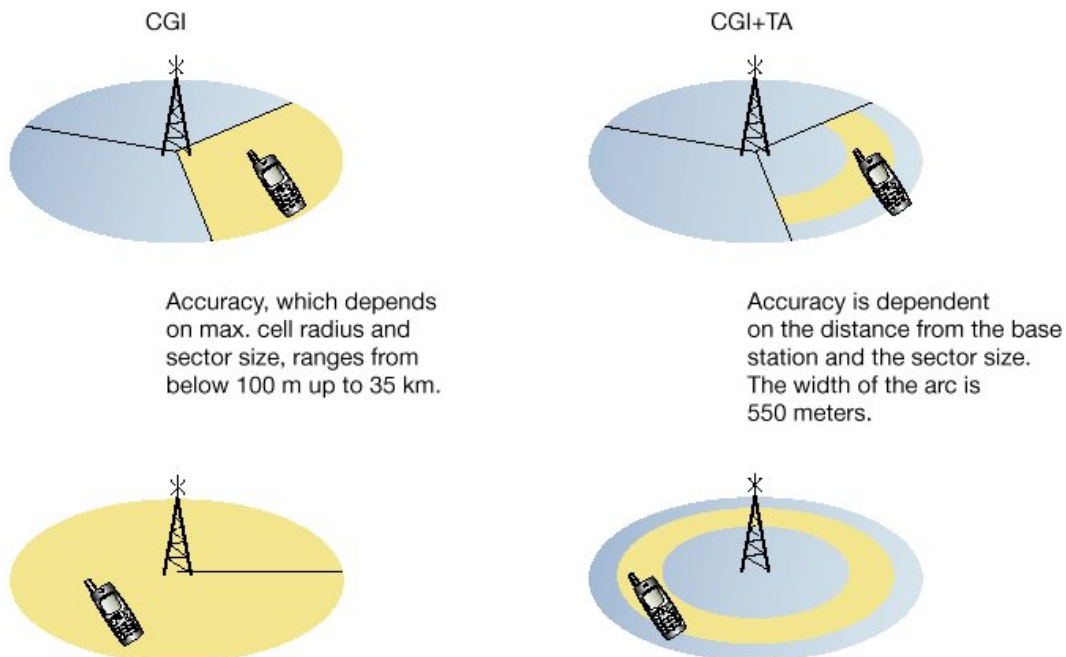
3.3 Συστήματα εντοπισμού θέσης που βασίζονται σε δίκτυα

Υπάρχουν δύο μέθοδοι που βασίζονται σε δίκτυα και είναι η μέθοδος της παγκόσμιας ταυτότητας κυψέλης και καθυστέρησης χρόνου μετάδοσης (Cell Global Identity and Timing Advance, CGI-TA) και η μέθοδος καθυστέρησης του χρόνου μετάδοσης (Uplink Time Of Arrival, UL-TOA) που περιγράφονται αναλυτικά παρακάτω.

Μέθοδος παγκόσμιας ταυτότητας κελιού και καθυστέρησης χρόνου μετάδοσης (Cell Global Identity and Timing Advance, CGI-TA)

Η μέθοδος εντοπισμού θέσης σε μια κυψέλη με την αύξηση χρόνου χρησιμοποιεί την παγκόσμια ταυτότητα της κελιού (Cell Global Identity, CGI) και την καθυστέρηση χρόνου μετάδοσης (Time Advance, TA) σαν παραμέτρους για τον καθορισμό της θέσεως των κινητών συσκευών. Η παράμετρος CGI καθορίζει το κελί του δικτύου μέσα στο οποίο βρίσκεται η κινητή συσκευή. Το κελί αυτό μπορεί να είναι είτε κυκλικό είτε τριγωνικό. Η παράμετρος TA είναι μια εκτίμηση για την απόσταση (σε βήμα των 250 μέτρων) της κινητής συσκευής από τον σταθμό βάσης. Η μέτρηση βασίζεται στην καθυστέρηση πρόσβασης από την αρχή μιας χρονοθυρίδας και την άφιξη των ριπών από την κινητή συσκευή. Η ακρίβεια της μεθόδου αυτής κυμαίνεται ανάλογα με το μέγεθος του κελιού. Η ακτίνα του κελιού μπορεί να κυμαίνεται από 100 μέτρα μέχρι 35 χιλιόμετρα (CGI). Το πλάτος ενός τόξου είναι 550 μέτρα.

Όπως φαίνεται στο σχήμα 9 η μέθοδος παγκόσμιας ταυτότητας κελιού και καθυστέρησης χρόνου μετάδοσης είναι απλώς μια αναβάθμιση του CGI. Η εκτιμώμενη θέση αναφέρεται σε γεωγραφικό πλάτος και μήκος και σε ένα σχήμα αβεβαιότητας μέσα στο οποίο η κινητή συσκευή βρίσκεται.



Σχήμα 9. Μέθοδος παγκόσμιας ταυτότητας κελιού και καθυστέρησης χρόνου μετάδοσης (CGI-TA)

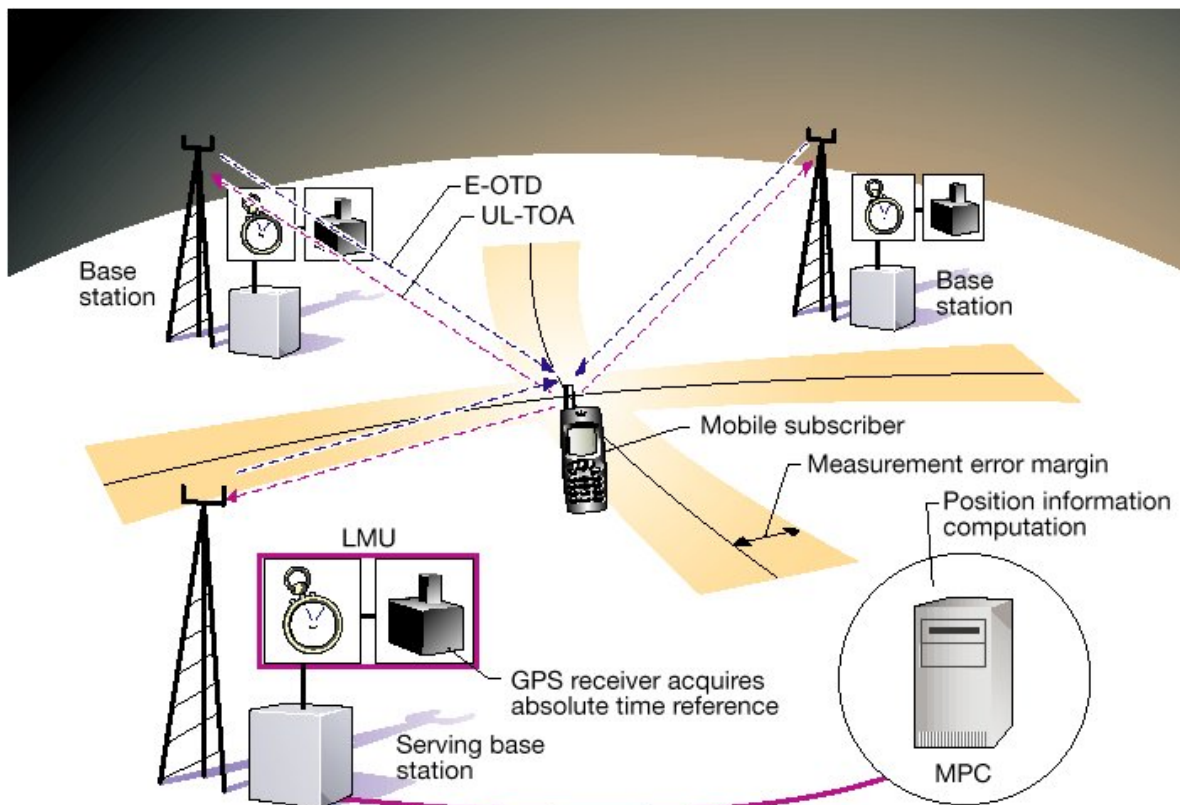
Μέθοδος μέτρησης του χρόνου άφιξης (Uplink Time Of Arrival, UL-TOA).

Η μέθοδος μέτρησης του χρόνου άφιξης (Uplink Time Of Arrival, UL-TOA) βασίζεται στη μέτρηση του χρόνου άφιξης ενός σήματος της κινητής συσκευής προς τέσσερις ή περισσότερους σταθμούς μέτρησης. Το σήμα αυτό μπορεί να είναι μια ακολουθία τυχαίων ριπών πρόσβασης ή ακόμα και κανονικές ριπές. Η μέθοδος μέτρησης του χρόνου άφιξης για τον εντοπισμό θέσης δουλεύει με όλες τις υπάρχοντες κινητές συσκευές και άρα δεν χρειάζεται καμία τροποποίηση σε αυτές. Οι μονάδες μέτρησης θέσης (Location Measurement Units, LMU) που βρίσκονται στους σταθμούς βάσης λαμβάνουν τις ριπές και υπολογίζουν την τιμές του χρόνου άφιξης αυτών (UL-TOA). Το κέντρο υπολογισμού θέσης (Mobile Position Center, MPC) υπολογίζει την διαφορά χρόνου άφιξης (TDOA) με την αφαίρεση τιμών ζευγών UL-TOA.

Προϋπόθεση για τον καθορισμό θέσης με την χρήση της μεθόδου μέτρησης του χρόνου άφιξης είναι οι ακόλουθες:

- Οι γεωγραφικές συντεταγμένες των μονάδων μέτρησης θέσης είναι γνωστές.
- Η απαρχή του χρόνου μεταξύ των μονάδων μέτρησης θέσης είναι γνωστό, πχ με την χρήση απόλυτου GPS χρόνου στις μονάδες μέτρησης θέσης για τον καθορισμό της πραγματικής χρονικής διαφοράς (Real Time Difference).

Το κέντρο υπολογισμού θέσης (MPC) παρέχει μια εκτίμηση θέσης και μια εκτίμηση αβεβαιότητας στην εφαρμογή. Η ακρίβεια αυτής της μεθόδου είναι κυμαινόμενη ανάλογη με το περιβάλλον και τον αριθμό των μονάδων μέτρησης θέσης. Η ακρίβεια αυτή κυμαίνεται συνήθως μεταξύ 50 μέτρα σε αστικές περιοχές και 150 μέτρα σε μη αστικές περιοχές (Σχήμα 10).



Σχήμα 10. Διάγραμμα της μεθόδου μέτρησης χρόνου άφιξης.

3.4 Εφαρμογές θέσης

Κατηγορίες υπηρεσιών

Οι υπηρεσίες θέσης κατηγοριοποιούνται με τον τύπο της εφαρμογής

- **Υπηρεσίες πληροφόρησης**

Οι υπηρεσίες πληροφόρησης κάνουν χρήση των τραπεζών πληροφοριών όπου οι πληροφορίες φιλτράρονται σύμφωνα με την σχετική θέση του χρήστη και των εφαρμογών που αυτός χρησιμοποιεί. Παραδείγματα υπηρεσιών πληροφόρησης περιλαμβάνουν πληροφορίες για τοπικές yellow pages, συμβάντα και καλλιτεχνικά δρώμενα κλπ.

- **Υπηρεσίες εντοπισμού**

Διαφορετικές υπηρεσίες μπορούν να χρησιμοποιήσουν τις πληροφορίες θέσης για τον εντοπισμό του κινητού τερματικού, να παράσχουν ασφάλεια, να αποτρέψουν κλοπές κλπ. Οι υπηρεσίες αυτές περιλαμβάνουν τον εντοπισμό ενός κλεμμένου αυτοκινήτου, και τον εντοπισμό κάποιου σε μια αραιοκατοικημένη περιοχή που χρειάζεται βοήθεια.

- **Υπηρεσίες έλεγχου πόρων**

Οι εφαρμογές έλεγχου πόρων χρησιμοποιούνται για τον έλεγχο του στόλου μεταφορικών αυτοκινήτων, εμπορικών πλοίων κλπ. Παραδείγματα αυτών των εφαρμογών είναι ο έλεγχος ενός στόλου Ταξί, η διεύθυνση ενός στόλου πλοίων κλπ

- **Πλοήγηση**

Οι εφαρμογές πλοήγησης χρησιμοποιούνται για να ενημερώσουν τους χρήστες για το πως μπορούν να κινηθούν καλύτερα από ένα σημείο Α σε ένα σημείο Β. Τέτοιες εφαρμογές μπορούν να προσαρμοστούν σε εφαρμογές για πλοήγηση για αυτοκίνητο ή πεζού ανθρώπου.

- **Άλλες Υπηρεσίες**

Κάποιες εφαρμογές θέσης όπως οργάνωση δικτύου, υπηρεσίες παροχής χαρτών δεν ανήκουν σε καμία από τις παραπάνω κατηγορίες. Παραδείγματα από αυτές τις υπηρεσίες περιλαμβάνουν χρέωση των χρηστών ανάλογα με το που βρίσκονται, χρησιμοποίηση δικτύου κλπ.

3.5 Σχετικά με τα συστήματα εντοπισμού θέσης

Υπάρχουν δύο είδη συστημάτων προσδιορισμού θέσης

- Επικαλυπτόμενα συστήματα
- Ολοκληρωμένα συστήματα

Τα επικαλυπτόμενα συστήματα είναι χτισμένα πάνω από υπάρχοντα δίκτυα. Επίσης επειδή χρησιμοποιούν λίγους πόρους του δικτύου, αυτά τα συστήματα μπορούν να εφαρμοστούν σε μικτά δίκτυα. Παρόλα αυτά, τα επικαλυπτόμενα συστήματα δεν πρόκειται να τυποποιηθούν, διότι δεν είναι σίγουρα για το μέλλον και ούτε ο σχεδιασμός τους δεν έχει τις αναγκαίες προϋποθέσεις. Γενικά, η αγορά για αυτά τα συστήματα δεν έχει τα απαιτούμενα εχέγγυα.

Όταν η διαδικασία τυποποίησης ολοκληρωθεί, θα είναι δυνατό να αναπτυχθούν δίκτυα τα οποία θα είναι συμβατά με όλα τα άλλα. Επίσης η λύση βασισμένη σε δίκτυα έχει τα πλεονεκτήματα ότι σχεδιάστηκε για να :

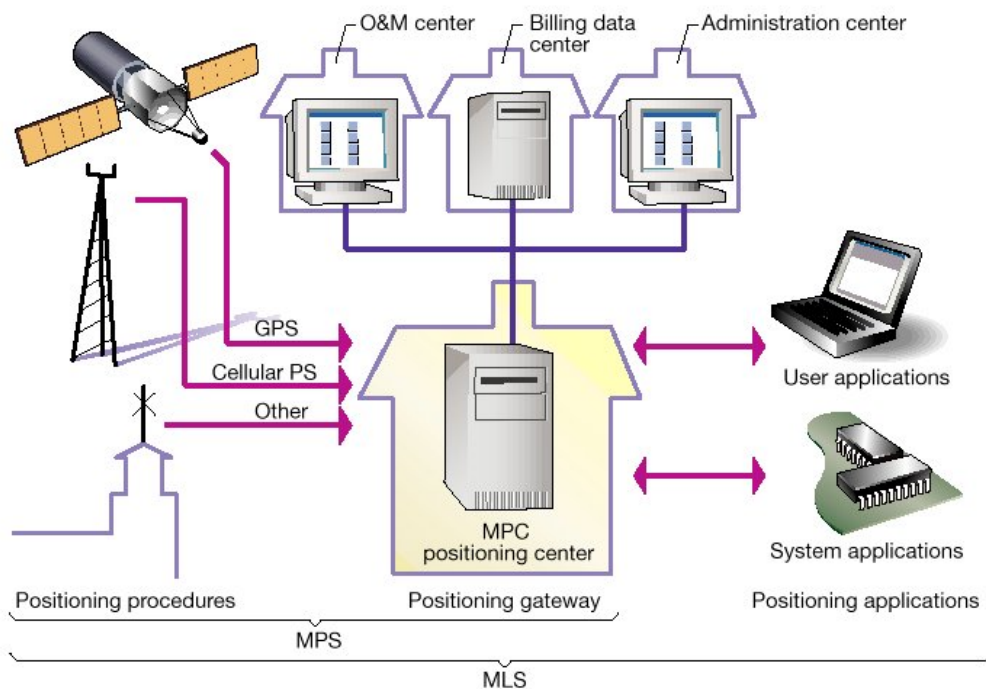
- Ικανοποιεί τις νομικές και εμπορικές απαιτήσεις
- Να ακολουθήσει τον δρόμο της εξέλιξης του GSM προς την τρίτη γενιά συστημάτων.

3.6 Τυποποίηση

Οι τεχνικές προσδιορισμού θέσης CGI+TA και UL-TOA τυποποιήθηκαν το Μάιο 1999 και σχέδια έχουν γίνει και για την τυποποίηση του E-OTD και A-GPS. Η τεχνική της αναβάθμισης της κάρτας SIM δεν πρόκειται να τυποποιηθεί.

3.7 Η λύση της ERICSSON

Η λύση της Ericsson για τα κινητά συστήματα προσδιορισμού θέσης αποτελείται από τρία λογικά υποσυστήματα (Σχήμα 11).



Σχήμα 11. Η λύση της ERICSSON (MLS), όπου ένα κινητό σύστημα εντοπισμού έχει εντοπισθεί σε ένα δίκτυο.

1. Το υποσύστημα προσδιορισμού θέσης μπορεί να χρησιμοποιήσει μια ποικιλία από τεχνικές για τον καθορισμό και την παροχή των γεωγραφικών συντεταγμένων:
 - Το κυψελωτό σύστημα προσδιορισμού θέσης όπως πχ UL-TOA και E-OTD.
 - Το βασισμένο σε δίκτυο παγκόσμιο σύστημα εντοπισμού(A-GPS).
 - Άλλες τεχνικές, όπως η τεχνική της αναβάθμισης της κάρτας SIM.
2. Το gateway υποσύστημα προσδιορισμού (MPC)που λειτουργεί σαν ενδιάμεσο μεταξύ του PLMN(Public Land Mobile Network) και της υπηρεσίας εντοπισμού θέσης (LCS-client), ανακτά δεδομένα από τα υποσυστήματα προσδιορισμού θέσης και τα μετατρέπει σε γεωγραφικές πληροφορίες για τον LCS-client. Επίσης παρέχει στον διαχειριστή ένα GUI(Graphical User Interface) συνήθως βασισμένο σε Java για τον έλεγχο του κάθε κόμβου-όπου στην πρόταση αυτή της Ericsson αυτοί οι κόμβοι καλούνται Mobile Positioning Center Tool (MPC-tool).
Σημειώνεται ότι επειδή το MPC είναι ενδιάμεσο κέντρο ελέγχει και μπορεί να εγγράψει την χρήση μιας συγκεκριμένης εφαρμογής προσδιορισμού θέσης, επιτρέποντας έτσι τους διαχειριστές να χρεώνουν τους χρήστες μέσω εφαρμογών που είναι εγκαταστημένες στο MPC είτε μέσω διεπαφών με το σύστημα χρέωσης.

3. Το υποσύστημα LCS-client, το οποίο είναι υποσύστημα του MPC περιέχει εφαρμογές οι οποίες κάνουν χρήση των γεωγραφικών πληροφοριών. Εσωτερικές εφαρμογές (όπως τηλέφωνα ανάγκης) είναι κωδικοποιημένες μέσα στο GSM σύστημα σύμφωνα με την τυποποίηση του GSM. Εξωτερικές εφαρμογές παρέχονται στο σύστημα από τους πωλητές του συστήματος, τους διαχειριστές κ.λ.π. Μερικές εφαρμογές μπορεί να είναι ανάλογες με τις απαιτήσεις των χρηστών.

ΜΕΡΟΣ Γ'

SOFTWARE AGENTS

Ο όρος **agent**, που σημαίνει πράκτορας ή αντιπρόσωπος, πηγάζει από το Λατινικό ρήμα *agere* δηλαδή καθοδηγώ, ενεργώ ή κάνω. Η σημερινή έρευνα και τεχνολογία έχει εισαγάγει τον όρο αυτό στην ανάπτυξη του λογισμικού και των υπολογιστικών συστημάτων, στην προσπάθεια αυτοματοποίησης της αλληλεπίδρασης ανθρώπου- μηχανής και βελτίωσης των επικοινωνιών μέσω υπολογιστικών δικτύων. Στο χώρο αυτό, η οντότητα του *agent* υλοποιείται ως πρόγραμμα, γι' αυτό καλείται *software agent*.

Με την βασική έννοια, ένα *software agent* είναι λογισμικό που ενεργεί για λογαριασμό κάποιου χρήστη, προκειμένου να φέρει σε πέρας μία συγκεκριμένη εργασία που του έχει ανατεθεί. Οι ερευνητές, θέλοντας να προσδιορίσουν ακριβέστερα την έννοια του *software agent*, έχουν δώσει διάφορους ορισμούς, αφαιρετικούς και περιγραφικούς, επηρεαζόμενοι κυρίως από τα υπό μελέτη συστήματα. Ένας πιο σαφής ορισμός συνοψίζεται ως εξής:

Software agent είναι μία οντότητα λογισμικού που λειτουργεί συνεχώς και αυτόνομα μέσα σε ένα περιβάλλον, στο οποίο φιλοξενούνται κι άλλα *agents* και διεργασίες. Η απαίτηση για συνέχεια και αυτονομία προέρχεται από την επιθυμία του *agent* να δραστηριοποιείται με ευέλικτο και έξυπνο τρόπο, να ανταποκρίνεται στις αλλαγές του περιβάλλοντος, χωρίς να χρειάζεται κάθε φορά την ανθρώπινη καθοδήγηση ή παρέμβαση. Το *agent*, μαθαίνει από την εμπειρία του, επικοινωνεί και συνεργάζεται με άλλα *agents* και μπορεί να μετακινείται από μέρος σε μέρος προκειμένου να ολοκληρώσει τις εργασίες του.

Σήμερα, τα περισσότερα *software agents* δεν έχουν τόσο γενικό χαρακτήρα, αλλά εξειδικεύονται ανάλογα με τη χρήση τους. Ο όρος *software agent* μπορεί να θεωρηθεί σαν μία ομπρέλα που καλύπτει μία περιοχή από ειδικούς τύπους *agent*. Σύμφωνα με τις απαιτήσεις της εκάστοτε περίπτωσης, κάθε *agent* μπορεί να κατέχει, σε μεγαλύτερο ή μικρότερο βαθμό, ιδιότητες όπως:

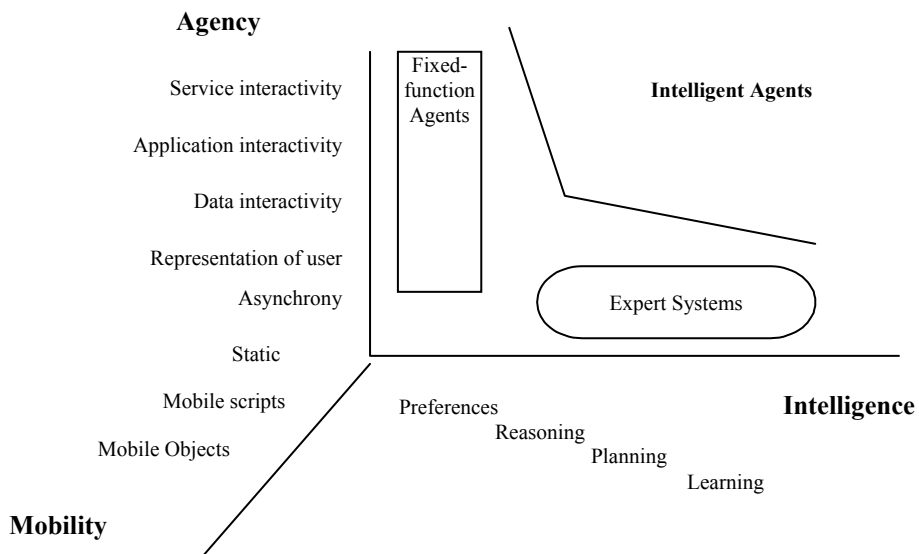
- **Αντιδραστικότητα (Reactivity).** Η ικανότητα να ανιχνεύει, να ενεργεί επιλεκτικά και να ανταποκρίνεται, τις κατάλληλες χρονικές στιγμές, σε αλλαγές του περιβάλλοντος.
- **Αυτονομία (Autonomy).** Προδραστική (proactive) συμπεριφορά. Η ικανότητα να αυτενεργεί, καθοδηγούμενο από τους στόχους του.
- **Συνεργατικότητα (Collaborative behavior).** Η ικανότητα να συνεργάζεται με άλλα *agents* για την επίτευξη κοινού στόχου.
- **Επικοινωνιακή ικανότητα (“High-level” communication ability).** Η ικανότητα να επικοινωνεί με χρήστες ή *agents* με γλώσσα που μοιάζει περισσότερο στην ανθρώπινη παρά στην τυπική γλώσσα συμβόλων των πρωτοκόλλων από πρόγραμμα σε πρόγραμμα.
- **Συμπερασματική ικανότητα (Inferential capability).** Η δυνατότητα να ενεργεί, ακόμη και με αφηρημένο προσδιορισμό εργασίας, χρησιμοποιώντας προηγούμενη γνώση γενικών σκοπών και μεθόδων για ευελιξία. Μπορεί να προχωράει πέρα από τις δεδομένες πληροφορίες, έχοντας συγκεκριμένα μοντέλα, του εαυτού του, του χρήστη ή/ και άλλων *agents*.
- **Χρονική συνέχεια (Temporal continuity).** Η διατήρηση της ταυτότητας και της κατάστασης του σε μεγάλες χρονικές περιόδους.
- **Προσωπικότητα (Personality).** Εκδήλωση συμπεριφοράς ανθρώπινου χαρακτήρα, όπως η συγκίνηση.
- **Προσαρμοστικότητα (Adaptivity).** Η δυνατότητα να μαθαίνει και να βελτιώνεται βάσει της εμπειρίας του.
- **Κινητικότητα (Mobility).** Η δυνατότητα να μετακινείται με δική του απόφαση από μία πλατφόρμα σε άλλη.

Η IBM τοποθετεί τα *software agents* σε χώρο που ορίζεται από τρεις διαστάσεις: Agency, Intelligence, Mobility (Σχήμα 12).

Agency είναι ο βαθμός της αυτονομίας και της «εξουσίας» που δίνεται στο agent. Μπορεί ποσοτικά να μετρηθεί από τη φύση της αλληλεπίδρασης του agent με άλλες οντότητες του συστήματος. Κατ' ελάχιστο, το agent θα πρέπει να εκτελείται ασύγχρονα. Ο βαθμός agency αυξάνεται, όσο ο agent μπορεί να αλληλεπιδρά με δεδομένα, εφαρμογές, υπηρεσίες κλπ.

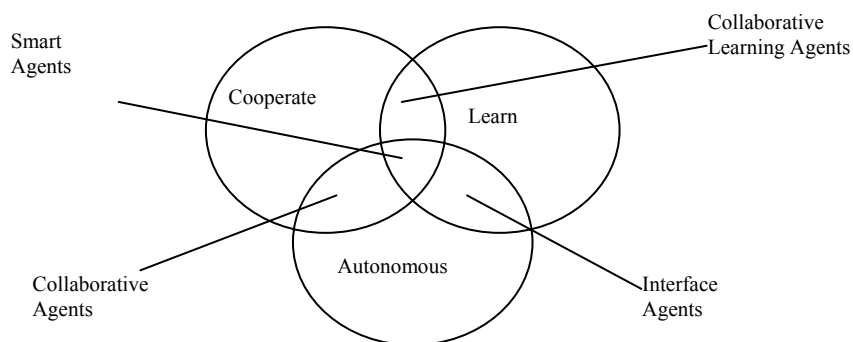
Intelligence είναι ο βαθμός της ικανότητας μάθησης του agent. Στον ελάχιστο βαθμό, μπορεί να λειτουργεί βάσει των προτιμήσεων του χρήστη. Σε υψηλότερα επίπεδα, το agent περιλαμβάνει ένα μοντέλο χρήστη, συγκεκριμένη συλλογιστική, έως και ικανότητα να μαθαίνει από το περιβάλλον του.

Mobility είναι ο βαθμός στον οποίο τα agents ταξιδεύουν διαμέσω του δικτύου. Οι κινητοί κώδικες (mobile scripts) συντίθενται σε μία μηχανή και μεταφέρονται σε άλλη για την εκτέλεση. Τα κινητά αντικείμενα (mobile objects) μεταβαίνουν από μηχανή σε μηχανή διακόπτοντας την εκτέλεση, μεταφέροντας μαζί τους πληροφορίες κατάστασης.



Σχήμα 12: Απεικόνιση των agents σε χώρο τριών διαστάσεων

Μία προτεινόμενη τυπολογία για τα software agents φαίνεται στο σχήμα 13. Η κατηγοριοποίηση των software agents, βάσει των κυρίων ιδιοτήτων τους, κάνει ευδιάκριτους διάφορους τύπους agents (Πίνακας.1), αν και υπάρχει μία σχετικότητα που έγκειται στις διαβαθμίσεις των ιδιοτήτων που εμφανίζονται.



Σχήμα 13 : Τυπολογία των agents βάσει των ιδιοτήτων τους

Static	Mobile
Deliberative	Reactive
Collaborative	Collaborative
Interface	Learning
Hybrid	Information
	/Internet
	Smart

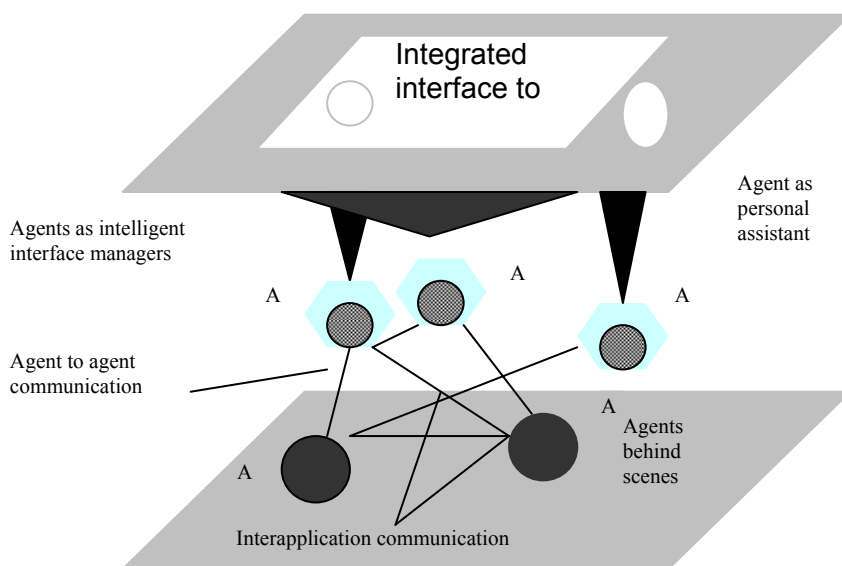
Πίνακας 1: Κατηγορίες των software agents.

Μία άλλη κατηγοριοποίηση των agents προκύπτει σύμφωνα με τις δομές ελέγχου, τα περιβάλλοντα (πχ. βάσεις δεδομένων, σύστημα αρχείων, δίκτυο), τις γλώσσες προγραμματισμού στις οποίες γράφεται το λογισμικό και στις εφαρμογές τους.

Ενώ η αρχική μελέτη πάνω στα software agents υποκινήθηκε από ερευνητές που μελετούσαν υπολογιστικά έξυπνα μοντέλα για καταναμημένα συστήματα, δύο ακόμη ενδιαφέροντα πρακτικής φύσεως επηρέασαν την περαιτέρω ανάπτυξή τους: 1) η απλοποίηση της πολυπλοκότητας των καταναμημένων υπολογισμών και 2) το ξεπέρασμα των περιορισμών των διεπαφών χρήστη. Οι δύο αυτοί παράγοντες, βασικά, μπορούν να θεωρηθούν, ως συνέχιση της τάσης για μεγαλύτερη αφαίρεση των διεπαφών ανάμεσα στις διάφορες υπηρεσίες.

Πριν μερικά χρόνια, η έξυπνη διαλειτουργικότητα μεταξύ των συστημάτων λογισμικού αποτέλεσε θέμα πρωτεύουσας σημασίας. Η εξέλιξη πολλών αντικειμενοστραφών προσεγγίσεων και η ανάπτυξη πάνω σε αποδεκτά πρότυπα (TCP/IP, HTTP, IIOP, ODBC) προσέφεραν το βασικό επίπεδο για τη συνεργασία πολλών συστημάτων και υπηρεσιών. Βρισκόμαστε, πλέον, στο μέσο της μετάβασης από το δικτυακό λειτουργικό σύστημα, στους υπολογισμούς (computing) μέσω Internet και intranets. Όσο προχωράει η μετάβαση αυτή, τόσο περισσότερο «γόνιμο έδαφος» αποκτούν οι δικτυακές υπηρεσίες που επιτυγχάνουν τη διαλειτουργικότητα, ανεξάρτητα λειτουργικού συστήματος.

Υψηλότερου επιπέδου διαλειτουργικότητα απαιτεί τη γνώση των ικανοτήτων του συστήματος, έτσι ώστε να μπορεί, μεταξύ των συστημάτων, να λάβει χώρα ασφαλής ανάθεση πόρων, σχεδιασμός εργασιών, εκτέλεση και έλεγχος. Η νέα τεχνολογία προτείνει ένα intelligent agent που να παίζει το ρόλο του διαχειριστή πόρων. Ένα επόμενο βήμα είναι η ολοκλήρωση ενός ή περισσότερων agents σε κάθε ένα σύστημα.



Σχήμα 14: Η αρχιτεκτονική ενός agent συστήματος

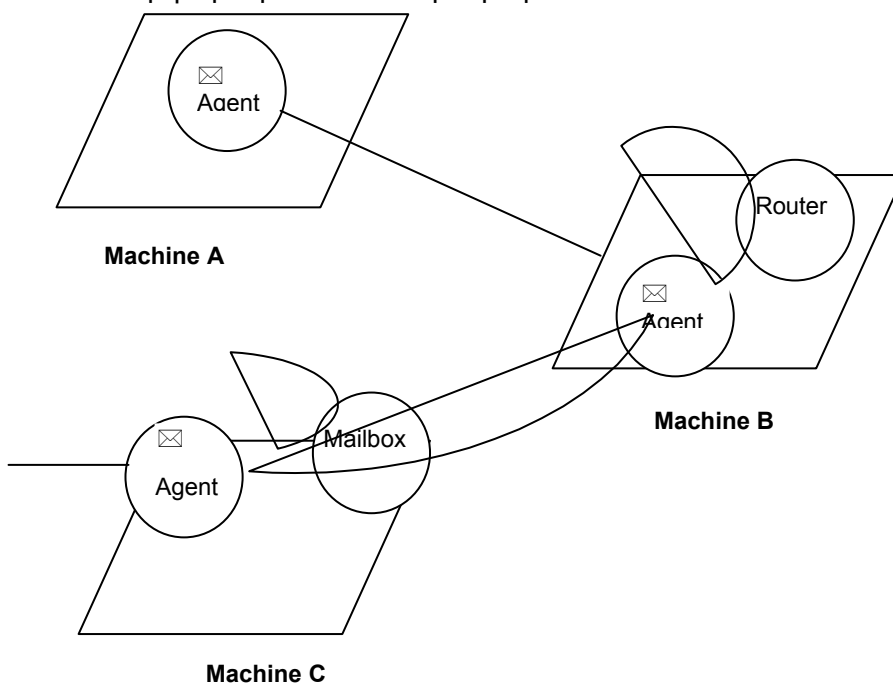
Εξάλλου, πολλά πλεονεκτήματα των σημερινών διεπαφών απευθείας χειρισμού (*direct manipulation interfaces*) αρχίζουν να αποδυναμώνονται, καθώς οι υπολογιστικές εργασίες φτάνουν υψηλές κλίμακες πολυπλοκότητας. Μία προσπάθεια προσαρμογής στη νέα πραγματικότητα αποτελεί η πρόταση για έμμεσο διοικητικό τρόπο αλληλεπίδρασης (*indirect management style*). Σύμφωνα με την προσέγγιση αυτή, οι χρήστες δεν είναι υποχρεωμένοι να δίνουν μία -μία τις εντολές στον υπολογιστή για τις εργασίες τους, αλλά η ευελιξία και η εξυπνάδα των *software agents*, τους επιτρέπει να δίνουν τη γενική καθοδήγηση και να μην ασχολούνται με τις λεπτομέρειες για τη διεκπεραίωσή τους.

Το σχήμα 14 απεικονίζει τους διάφορους ρόλους που μπορούν να παίξουν τα *software agents* σε μία ανάλογη αρχιτεκτονική συστήματος. Συνήθως, οι εφαρμογές απαιτούν υπηρεσίες μέσω των *agents*, οι οποίες συσχετίζονται περισσότερο με τις προθέσεις του χρήστη παρά με συγκεκριμένες υλοποιήσεις. Καθώς τα *software agents* εξελίσσονται από στατικές σε κινητές οντότητες, αναμένεται να προκύψει ένας επαναπροσδιορισμός των καταναμημένων δικτυακών συστημάτων, αλλά και σε όλο το εύρος του παγκόσμιου ιστού (WWW).

Mobile Agents

Ο τρόπος ζωής και οι απαιτήσεις του σύγχρονου ανθρώπου αποτελούν καθοριστικά στοιχεία για την ανάπτυξη και την εξέλιξη των διάφορων τεχνολογιών. Η ψηφιακή τεχνολογία των agents συνιστά νέες οντότητες λογισμικού που εμφανίζονται να έχουν συμπεριφορά παρόμοια με την ανθρώπινη, έτσι ώστε, πραγματικά, να διευκολύνουν τους χρήστες στην καθημερινή τους ζωή. Η κινητικότητα, ως γενικό χαρακτηριστικό ζωής, προσδίδει δυναμισμό στα software agents, που, με την ιδιότητα αυτή, λέγονται mobile agents και έρχονται να αλλάξουν το υπάρχον σκηνικό στο περιβάλλον των υπολογιστών.

Ένα απλό παράδειγμα που απεικονίζει τη φιλοσοφία των mobile agents φαίνεται στο σχήμα 15. Ένα mobile agent μεταφέρει ένα e-mail μήνυμα. Μεταβαίνει πρώτα σε έναν δρομολογητή (router) και μετά, στο ταχυδρομικό κουτί (mailbox) του παραλήπτη. Το agent μπορεί να εκτελεί πολύπλοκη επεξεργασία σε κάθε ενδιάμεση μηχανή, έτσι ώστε να επιβεβαιώνει ότι το μήνυμα φτάνει στον προορισμό του.



Σχήμα 15 Ένα mobile agent που μεταφέρει e-mail μηνύματα

Γενικά, δεν είναι όλα τα agents κινούμενα. Υπάρχουν τα στατικά (stationary) agents, τα οποία εκτελούνται μόνο στο σύστημα στο οποίο ξεκινούν την εκτέλεσή τους και επικοινωνούν με άλλα agents μέσω των συνηθών τρόπων επικοινωνίας (RPC, messaging). Αντιθέτως, τα κινούμενα (mobile) agents δεν είναι «δεμένα» σε ένα σύστημα, αλλά είναι ελεύθερα να ταξιδεύουν μεταξύ των υπολογιστών του δικτύου. Μεταφέρουν μαζί τους τον κώδικα και την κατάσταση τους και μπορούν να εκτελούνται σε οποιοδήποτε περιβάλλον του δικτύου, ανεξάρτητα από το αρχικό περιβάλλον εκτέλεσής τους.

5.1 Ορισμός των Mobile Agent

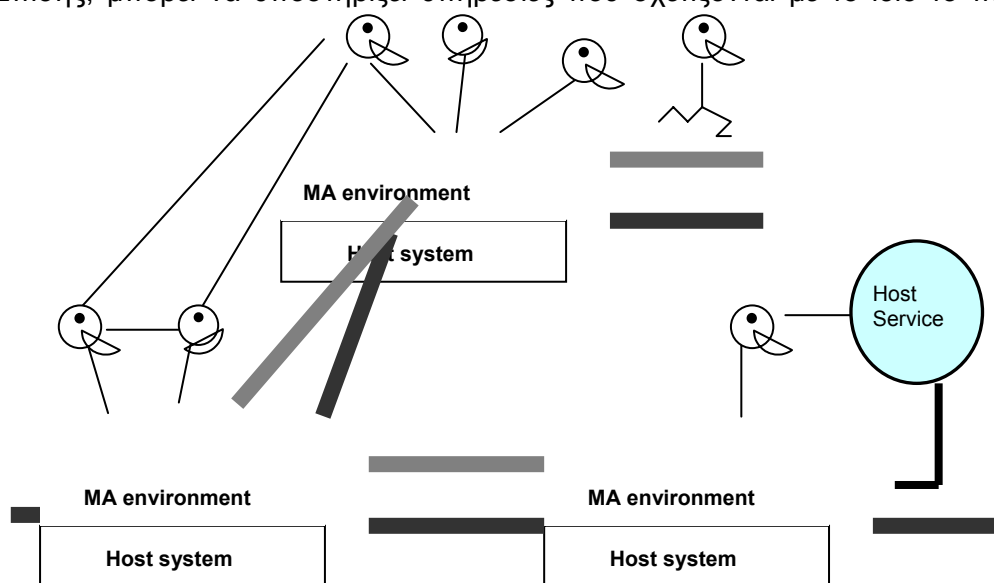
Ένα mobile agent είναι ένα software agent, το οποίο μπορεί να μετακινείται από υπολογιστή σε υπολογιστή μέσα σε ένα ομογενές ή ετερογενές περιβάλλον. Το πρόγραμμα επιλέγει πότε και πού να «μεταναστεύσει». Μπορεί να ξεκινήσει τη λειτουργία του σε ένα σημείο, να μετακινηθεί σε άλλη μηχανή και να ολοκληρώσει την λειτουργία του σε μία νέα μηχανή.

Ένα mobile agent περιλαμβάνει :

- **Κώδικα (Code)** : Το πρόγραμμα που καθορίζει την συμπεριφορά του agent.
- **Κατάσταση (State)** : Εσωτερικές μεταβλητές του agent κλπ., οι οποίες του επιτρέπουν να συνεχίζει τις λειτουργίες του μετά από την μετακίνησή του σε άλλον υπολογιστή.
- **Ιδιότητες (Attributes)** : Πληροφορίες που το περιγράφουν, όπως ο κωδικός του (identifier), η πηγή του, ο ιδιοκτήτης του, οι απαιτήσεις σε πόρους, τα κλειδιά πιστοποίησης, η ιστορία κίνησής του, οι περιορισμοί περιβάλλοντος κλπ. Μερικές ιδιότητες καθορίζονται από το ίδιο το agent, ενώ άλλες δεν μπορούν να αλλάζουν από το agent.
- **Τοποθεσία (Location)** : Πληροφορίες που προσδιορίζουν την τρέχουσα θέση του agent, σε κάθε host μηχανή που επισκέπτεται.

Το mobile agent κληρονομεί αρκετά από τα χαρακτηριστικά του software agent. Πρέπει όμως να τονιστεί ότι, πρόκειται για μία οντότητα λογισμικού η οποία «ζει» μέσα σε συγκεκριμένο περιβάλλον λογισμικού, το οποίο καλείται **Mobile Agent (MA) environment**.

Ένα mobile agent environment είναι ένα σύστημα λογισμικού, το οποίο κατανέμεται πάνω σε ένα δίκτυο ομογενές ή ετερογενών υπολογιστών (Σχήμα 16). Ο πρωταρχικός του σκοπός είναι να παρέχει ένα περιβάλλον μέσα στο οποίο μπορούν να εκτελούνται τα mobile agents. Επίσης, μπορεί να υποστηρίζει υπηρεσίες που σχετίζονται με το ίδιο το περιβάλλον,



Σχήμα 16: Το περιβάλλον των mobile agents.

υπηρεσίες των συστημάτων πάνω στα οποία έχει δομηθεί το mobile agent environment και υπηρεσίες που επιτρέπουν την πρόσβαση σε άλλα mobile agent συστήματα. Το mobile agent environment χτίζεται στην κορυφή ενός host συστήματος. Τα mobile agents ταξιδεύουν μεταξύ τέτοιων συστημάτων. Επικοινωνούν μεταξύ τους είτε τοπικά είτε απομακρυσμένα. Επικοινωνία επίσης, μπορεί να λάβει χώρα μεταξύ ενός mobile agent ή/ και μιας υπηρεσίας του συστήματος.

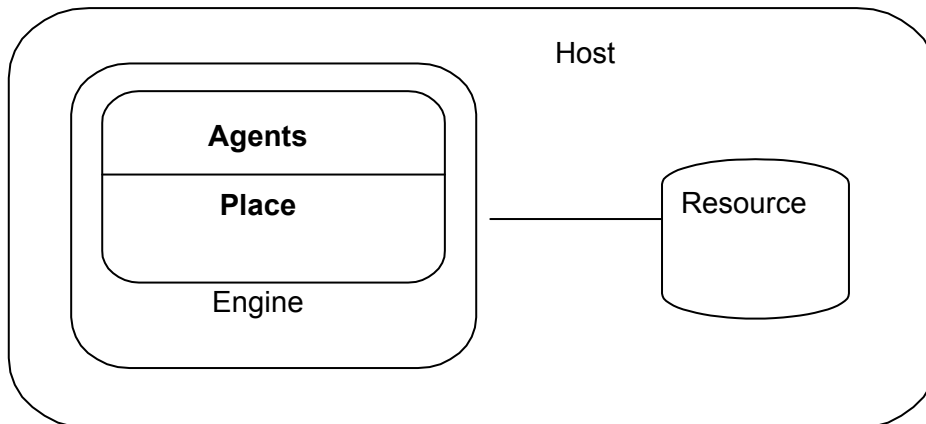
Ο όρος *place* συγκεντρώνει τις πρωταρχικές ιδιότητες που περιγράφουν το περιβάλλον λειτουργίας των mobile agents. Τέσσερις είναι οι παράγοντες που πρωταγωνιστούν σε ένα **agent place**:

Engine: Παίζει το ρόλο μίας εικονικής μηχανής (virtual machine) για τα places και τα agents. Είναι μία φυσική οντότητα που παρέχει συνδέσμους με το δίκτυο και άλλους πόρους που διατίθενται από τον host.

Resources: Οι τοπικοί πόροι και οι υπηρεσίες, όπως τα δίκτυα, οι βάσεις δεδομένων, οι επεξεργαστές, η μνήμη, οι δίσκοι και άλλες hardware και software υπηρεσίες.

Location: Συνίσταται στο συνδυασμό του ονόματος ενός place μέσα στο οποίο εκτελείται το agent και της δικτυακής διεύθυνσης της μηχανής την οποία καταλαμβάνει το place.

Principals: Ένα place έχει δύο κύρια στοιχεία: Τον κατασκευαστή (manufacturer), που παρέχει την υλοποίηση του place, και τον ιδιοκτήτη (place master), που αντιπροσωπεύει το πρόσωπο ή οργανισμό που είναι υπεύθυνος για τη λειτουργία του place. Στο σχήμα 17 φαίνεται η σχέση του place και του Engine.



Σχήμα 17: Η σχέση μεταξύ των Place και Engine

Ένας συγκεκριμένος υπολογιστής του δικτύου μπορεί να «φιλοξενεί» πολλαπλές engines. Κάθε engine μπορεί να περιλαμβάνει πολλά places, με μοναδικά ονόματα, και κάθε place μπορεί να περιέχει πολλά agents.

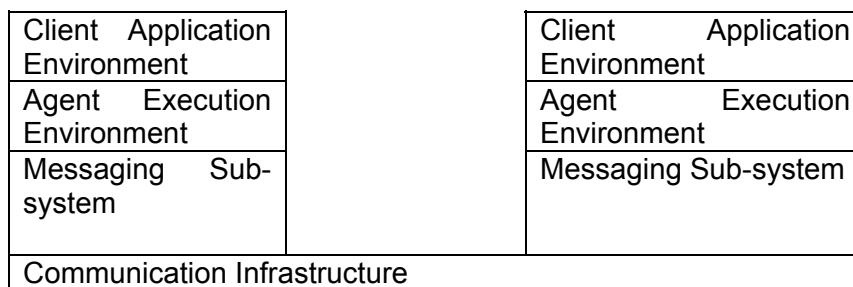
5.2 Το Υπολογιστικό Περιβάλλον του Mobile Agent

Η ιδέα μετάδοσης εκτελέσιμων προγραμμάτων μεταξύ clients και servers είναι αρκετά διαδομένη τα τελευταία χρόνια. Τα mobile agents μπορούν να θεωρηθούν ως επέκταση των μεθόδων απομακρυσμένης μετάδοσης κώδικα ή απομακρυσμένης υποβολής εργασιών. Η βασική ιδέα των mobile agent υπολογισμών περιγράφεται ως εξής:

Ο υπολογιστής του client συνίσταται σε ένα περιβάλλον λειτουργικού συστήματος (πχ. OS/2 ή Microsoft Windows) που περιέχει μία ή περισσότερες εφαρμογές για αλληλεπίδραση με έναν απομακρυσμένο server. Οι εφαρμογές μπορεί να αφορούν την αναζήτηση και ανάκτηση πληροφοριών, την επίτευξη συναλλαγών, τους πελάτες ηλεκτρονικού ταχυδρομείου, κλπ. Αυτές οι εφαρμογές είναι «δεμένες» με ένα περιβάλλον εκτέλεσης για mobile agents.

Μέσω των APIs, μία εφαρμογή μπορεί να περάσει παραμέτρους σε διάφορες κλάσεις agent προγραμμάτων, και αντίστροφα, τα agent προγράμματα μπορούν να επιστρέφουν παραμέτρους στα προγράμματα εφαρμογών. Οι κλάσεις μπορούν είτε να αποτελούν μέρος του βασικού περιβάλλοντος εκτέλεσης για mobile agents είτε να αποτελούν agents καταναμημένα με τις εφαρμογές του λειτουργικού συστήματος, είτε agents που έλαβε ο client από έναν server ή ένα άλλο ομότιμο κόμβο του δικτύου.

Συνήθως όμως, μπορεί να μην υπάρχει κανένα πρόγραμμα εφαρμογής, αλλά τα προγράμματα των agents να είναι αυτά που μαζεύουν πληροφορίες από οποιαδήποτε θύρα εισόδου και τις παρουσιάζουν στη διεπαφή χρήστη. Σε αυτή την περίπτωση, τα προγράμματα των agents ή το περιβάλλον εκτέλεσης για τα agents, πρέπει να είναι συνδεδεμένο με τις βιβλιοθήκες διεπαφής χρήστη στην μηχανή του client.



Σχήμα 18: Η βασική ιδέα του mobile agent υπολογιστικού περιβάλλοντος.

Επίσης, το περιβάλλον εκτέλεσης χρειάζεται να είναι συνδεδεμένο με διάφορες λειτουργίες του λειτουργικού συστήματος, όπως τον διαχειριστή μνήμης, το χρονιστή, το σύστημα αρχείων κλπ. Ειδικότερα, χρειάζεται να είναι συνδεδεμένο με την υπηρεσία μεταφοράς μηνυμάτων, έτσι ώστε να στέλνει τους mobile agents μέσω της επικοινωνιακής υποδομής. Το περιβάλλον εκτέλεσης των agents μπορεί να έχει πρόσβαση σε πολλά διαφορετικά agent προγράμματα, τα οποία παρέχουν διαφορετικές υπηρεσίες στις client εφαρμογές. Στο σχήμα 18 φαίνεται η βασική ιδέα του mobile agent υπολογιστικού περιβάλλοντος.

Όταν μία εφαρμογή χρειάζεται να στείλει ένα mail ή να κάνει μία συναλλαγή, συλλέγει τις απαιτούμενες πληροφορίες και τις περνά μέσω του API στο περιβάλλον εκτέλεσης των agents (agent execution environment). Αυτό αρχικοποιεί την εκτέλεση ενός στιγμιότυπου (instance) μιας συγκεκριμένης κλάσης agent, σαν μία διεργασία μέσα στο περιβάλλον. Η διεργασία ενδέχεται να είναι μία διεργασία ή ένα νήμα του λειτουργικού συστήματος ή/και να διοικείται από ένα σύνολο νημάτων του περιβάλλοντος εκτέλεσης των agents.

Το πρόγραμμα του agent μπορεί να δομηθεί από διεργασίες ή από κλάσεις αντικειμένων. Σε κάθε περίπτωση, το agent έχει πρόσβαση σε λειτουργίες μέσα στο περιβάλλον εκτέλεσης, όπως είναι οι διεργασίες των εφαρμογών, του λειτουργικού συστήματος ή άλλων υποσυστημάτων ή agent προγραμμάτων. Το πρόγραμμα του agent μπορεί να γραφτεί είτε σε μία γλώσσα μηχανής είτε σε μία μεταφρασμένη γλώσσα. Τα προγράμματα μεταφρασμένων γλωσσών περιφέρονται με ασφάλεια ευκολότερα από ότι η γλώσσα μηχανής και επιτρέπουν στο agent να περιέχει αναφορές σε διαδικασίες ή κλάσεις που ίσως να μην είναι παρούσες στο σύστημα στο οποίο βρίσκεται, αλλά να είναι διαθέσιμες στο σύστημα προορισμού του.

Οι πληροφορίες που διατίθενται από μία εφαρμογή, γίνονται δεκτές από το agent ως μέρος της αρχικοποίησής του (initialization), και σε συγκεκριμένο σημείο της διαδικασίας εκτέλεσης, εκτελεί εντολές που μπορούν να προκαλέσουν διάφορα αποτελέσματα, όπως:

- Το τρέχων agent να αναβάλλει την τρέχουσα διεργασία ή να δημιουργήσει ένα καινούριο agent στο ίδιο περιβάλλον εκτέλεσης για να την εκτελέσει.
- Η διεργασία που αναβλήθηκε ή η νέα διεργασία, περιλαμβάνοντας την κατάσταση (state), τη στοίβα (stack), το σωρό (heap) και όλες τις εξωτερικές αναφορές της, να συγκροτηθεί μέσα σε ένα μήνυμα, εκφρασμένο σε μία μορφή ανεξάρτητη μηχανής. Το μήνυμα μπορεί να σταλεί στον τελικό προορισμό, όπου συλλέγεται από το υποσύστημα μηνυμάτων του server, στο περιβάλλον εκτέλεσης για agents.
- Το agent να μεταβεί από τον client που έχει κάποια αίτηση - για πληροφορίες, συναλλαγές ή διανομή ηλεκτρονικού ταχυδρομείου- στον server που είναι σε θέση να την ικανοποιήσει.
- Το agent, κατά την διάρκεια εκτέλεσής του στο server, περνά τις πληροφορίες που «κρατάει» από τις εφαρμογές του client σε διαδικασίες προγραμμάτων του server, λαμβάνει νέες πληροφορίες και επιστρέφει. Κατά την ολοκλήρωση του σταδίου αυτού, μπορεί να:
 - περιμένει για κάποιο συμβάν εφαρμογής στο server .
 - τερματίσει την εκτέλεσή του.
 - επαναλάβει την μετάβαση σε άλλον server, είτε πηγαίνοντας το ίδιο είτε δημιουργώντας έναν άλλον agent για να μεταβεί.

Ειδικότερα, το ίδιο το agent μπορεί να κάνει τις ενέργειες ανάκτησης πληροφοριών και εάν η υπηρεσία δεν είναι διαθέσιμη ή ικανοποιητική, να αποφασίζει, βάσει των δεδομένων που

έχει λάβει από τον τρέχοντα server, αν θα μεταβεί σε κάποιον άλλον server στη συνέχεια ή θα επιστρέψει τα αποτελέσματα στον client.

5.3 Η Δομή του Mobile Agent

Η οντότητα του mobile agent εμπεριέχει στη δομή της πολλά είδη μοντέλων, που ορίζουν, στην ολότητά τους, το συγκεκριμένο τρόπο λειτουργίας και συμπεριφοράς του agent, βάσει του οποίου εκδηλώνει ιδιότητες που το χαρακτηρίζουν. Τα είδη των μοντέλων αυτών είναι τα εξής:

- agent μοντέλο,
- μοντέλο κύκλου ζωής,
- υπολογιστικό μοντέλο,
- μοντέλο ασφάλειας,
- επικοινωνιακό μοντέλο και
- μοντέλο πλοήγησης.

Το mobile agent environment επιτρέπει την πλειοψηφία των μοντέλων αυτών.

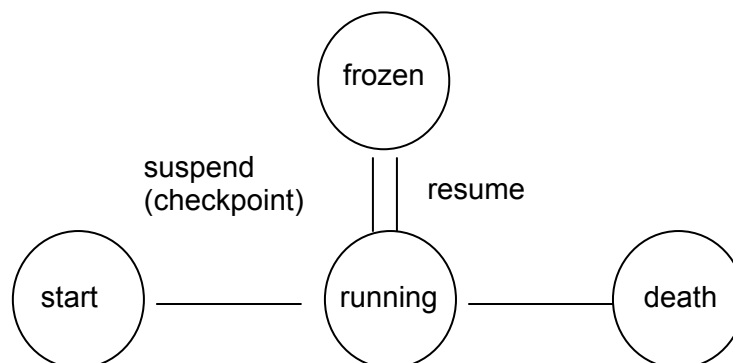
(i) Agent Μοντέλο (Agent Model)

Το μοντέλο αυτό καθορίζει την εσωτερική δομή του ευφυούς μέρους του mobile agent. Βάσει των πρωταρχικών ιδιοτήτων ενός software agent, καθορίζει την αυτονομία, τα χαρακτηριστικά μάθησης και συνεργασίας του agent, την προδραστική (proactive) και αντιδραστική (reactive) φύση του.

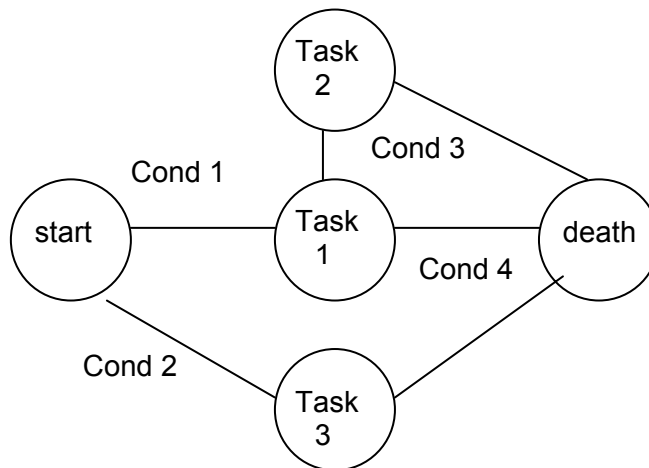
(ii) Μοντέλο Κύκλου Ζωής (Life-cycle Model)

Αυτό το μοντέλο καθορίζει τις διαφορετικές καταστάσεις ενός mobile agent στη διάρκεια εκτέλεσής του και τα γεγονότα που προκαλούν την μετάβασή του από την μία κατάσταση στην άλλη. Έτσι, σχετίζεται στενά και με το υπολογιστικό μοντέλο που περιγράφει το «πώς» συμβαίνει η εκτέλεση. Σήμερα, τα πιο δημοφιλή μοντέλα κύκλου ζωής είναι το *persistent process model*, που υιοθετήθηκε στο πλαίσιο Telescript και στο AgentTCL, και το *task based model*, που οριοθετήθηκε στα Aglets.

Το persistent process μοντέλο, ξεκινά με μία αρχική κατάσταση (start state), προχωράει σε μία κατάσταση εκτέλεσης (running state), όπου εκτελείται μία persistent διεργασία και περνάει σε μία κατάσταση θανάτου, (death state) όπου η διεργασία τερματίζεται (Σχήμα 19). Όταν ένα mobile agent μετακινείται από τον έναν κόμβο στον άλλον, κρατά το σημείο που σταματάει την εκτέλεση και εισέρχεται σε μία παγωμένη κατάσταση (frozen state). Φτάνοντας στον κόμβο προορισμού, όπου η διεργασία συνεχίζεται, το agent επανεισέρχεται στην κατάσταση εκτέλεσης στο σημείο στο οποίο την είχε σταματήσει. Αυτό το είδος κύκλου ζωής είναι το πιο δυναμικό και ευέλικτο, εφόσον όλα τα άλλα μοντέλα κύκλου ζωής μπορούν να βασιστούν πάνω του και να εξελιχθούν.



Σχήμα 19: Κύκλος ζωής σύμφωνα με το μοντέλο της persistent διεργασίας.



Σχήμα 20: Κύκλος ζωής σύμφωνα με το task based μοντέλο.

Το task based μοντέλο ξεκινάει με μία αρχική κατάσταση (start state). Βασιζόμενο σε ένα σύνολο από συνθήκες, προχωράει στη συνέχεια σε ένα δίκτυο εργασιών (Σχήμα 20). Κάθε εργασία έχει την δική της κατάσταση. Ωστόσο, όταν το agent μετακινείται σε έναν νέο κόμβο η κατάσταση της τρέχουσας εκτελούμενης εργασίας χάνεται. Προτού το agent μετακινηθεί, πρέπει να υποδείξει την πρώτη εργασία που πρέπει να αρχίσει όταν επανα-υλοποιηθεί στον κόμβο προορισμού. Η ευελιξία αυτής της προσέγγισης είναι μειωμένη γιατί χάνεται πληροφορία κατάστασης κατά την διάρκεια μεταφοράς.

(iii) Υπολογιστικό Μοντέλο (Computational Model)

Το υπολογιστικό μοντέλο καθορίζει πώς ένα mobile agent εκτελείται όταν βρίσκεται στην ενεργή κατάσταση (running state). Ο υπολογισμός λαμβάνει χώρα μέσα σε ένα περιβάλλον και διευκολύνεται από κάποιο είδος επεξεργαστή, όπως είναι η CPU ενός υπολογιστή ή ένας πιο αφηρημένος επεξεργαστής όπως η Java virtual machine.

Μέρος του μοντέλου αυτού αποτελεί κι ένα σύνολο από πρωταρχικές εντολές (instructions) που πρέπει να καθοριστούν. Αυτό το σύνολο καθορίζει τις υπολογιστικές ικανότητες ενός agent και περιλαμβάνει εντολές διαχείρισης δεδομένων και οδηγίες ελέγχου νημάτων.

Οι προγραμματιστές των mobile agents, έχουν πρόσβαση στα άλλα μοντέλα τους μέσω του υπολογιστικού μοντέλου. Αυτό σημαίνει ότι η δομή του επηρεάζει τα υπόλοιπα μοντέλα. Σε ένα πολύ γενικό και δυναμικό υπολογιστικό μοντέλο, πρέπει να ληφθεί υπόψη η επίδραση αυτού στον κύκλο ζωής, στην ασφάλεια, στην επικοινωνία και στην πλοήγηση του mobile agent.

(iv) Μοντέλο Ασφάλειας (Security Model)

Το θέμα της ασφάλειας στα δίκτυα αποτελεί ερευνητικό ζήτημα μεγάλης σπουδαιότητας σήμερα. Η ελεύθερη μετακίνηση των mobile agents στο δίκτυο μπορεί να επιφέρει διαφόρων ειδών προβλήματα.

Οι απειλές της ασφάλειας με τη χρήση των mobile agents μπορούν να χωριστούν στις ακόλουθες κατηγορίες: Η πρώτη έχει να κάνει με την προστασία των mobile agents από καταστροφικούς hosts. Η δεύτερη περιλαμβάνει την προστασία των host κόμβων από καταστροφικούς mobile agents. Η τρίτη αφορά στην προστασία του δικτύου. Εκτενέστερη ανάλυση των θεμάτων ασφάλειας γίνεται παρακάτω σε ξεχωριστή ενότητα.

Το πλαίσιο Telescript παρέχει τρεις μηχανισμούς που μπορούν να εφαρμοστούν σε διαφορετικές βαθμίδες:

➤ **Προστασία και ασφάλεια του Agent:** Επιτρέπει την ασφαλή αλληλεπίδραση μεταξύ των agents και μεταξύ των agents και των hosts. Αυτό επιτυγχάνεται χρησιμοποιώντας

authenticated οντότητες, προστατευμένες αναφορές, συγκεκριμένα permissions και πρωτόκολλα επικοινωνίας.

➤ **Προστασία του συστήματος:** Ελέγχει την πρόσβαση σε πόρους του συστήματος. Αυτό επιτυγχάνεται επιβάλλοντας πρόσβαση σε αυτούς τους πόρους μέσω διαμεσολαβητών (proxy objects). Οι proxies που υποστηρίζονται είναι: External File (για την πρόσβαση αρχείων), External Handle (για την πρόσβαση δικτύου) και Control Manager (για τις λειτουργίες διαχείρισης).

➤ **Προστασία του δικτύου:** Παρέχει δυνατότητες ελέγχου της αυθεντικότητας, επίπεδα εξουσιοδότησης και privacy επικοινωνίας μέσω της χρήσης ασφαλών καναλιών, εξωτερικών περιορισμών και συγκεκριμένων μεθόδων πολιτικής ανά περιοχή.

Ομοίως και η Java και η Safe-Tcl παρέχουν παρόμοια στοιχεία για την ασφάλεια.

(v) Επικοινωνιακό Μοντέλο (Communication Model)

Τα mobile agents θα είχαν περιορισμένη χρήση αν δεν μπορούσαν να επικοινωνούν με άλλες οντότητες μέσα σε ένα περιβάλλον. Τέτοιες οντότητες μπορεί να είναι οι χρήστες (σταθεροί ή κινούμενοι), το περιβάλλον εκτέλεσης τους και άλλα συστήματα, όπως τα καταμεμημένα συστήματα της αρχιτεκτονικής CORBA.

Επικοινωνία απαιτείται κατά την πρόσβαση σε εξωτερικές υπηρεσίες, στη διάρκεια συνεργασίας και συντονισμού των agents με άλλες οντότητες και στην ανταγωνιστικότητα μεταξύ τους.

Ένα πρωτόκολλο αποτελεί ένα επικοινωνιακό μοντέλο. Υπάρχει μεγάλη ποικιλία πρωτοκόλλων, τα οποία διαφοροποιούνται ανάλογα με τους τύπους των οντοτήτων που επικοινωνούν. Γι' αυτό το λόγο, τα mobile agents χρειάζονται παραπάνω από ένα επικοινωνιακά μοντέλα. Τότε, όμως, γεννάται και η ανάγκη μετάφρασης μεταξύ των πρωτοκόλλων. Τέτοιες δυνατότητες μπορεί να φέρει εσωτερικά ένα mobile agent ή να παρέχονται εξωτερικά από μία υπηρεσία.

(vi) Μοντέλο Πλοήγησης (Navigation Model)

Αυτό το μοντέλο έχει να κάνει με όλες τις πτυχές της κινητικότητας του agent και αποτελεί το κομμάτι μεγίστου ενδιαφέροντος των τελευταίων ετών. Οι περισσότεροι ερευνητές συμφωνούν ότι για την πλοήγηση είναι απαραίτητα τα παρακάτω:

➤ Συμβάσεις ονομασίας (naming conventions) για όλες τις οντότητες μέσα στο mobile agent σύστημα (agents, hosts, υπηρεσίες, κλπ.).

➤ Πρόσβαση σε πληροφορίες ανεξάρτητα του απομακρυσμένου mobile agent περιβάλλοντος.

➤ Ικανότητα μετάβασης του mobile agent σε μία ανασταλθείσα (suspended) κατάσταση του κύκλου ζωής και ετοιμότητα μετακίνησης σε απομακρυσμένο host.

➤ Ικανότητα λήψης ενός suspended mobile agent από έναν απομακρυσμένο host και επανασύστασής του μέσα σε ένα νέο περιβάλλον.

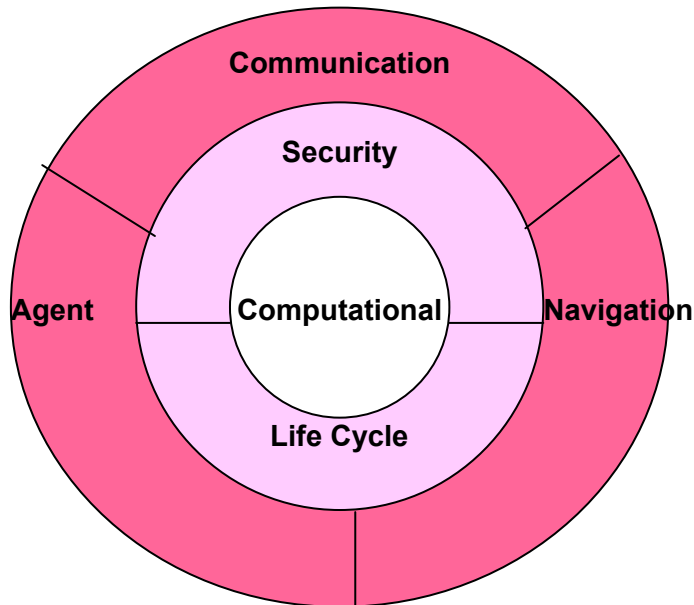
➤ Υπηρεσίες καταλόγων και αναφορών που βοηθούν τα agents στην ανακάλυψη νέων σχετικών υπηρεσιών που μπορούν να χρησιμοποιήσουν. Σύγχρονα συστήματα είναι τα X.500, CORBA Trader, LDAP και FIPA.

➤ Υπηρεσίες πληροφοριών της τοπολογίας δικτύου μπορούν να παρέχουν πληροφορίες για την κατάσταση του δικτύου. Τέτοιες πληροφορίες βοηθούν το agent να λάβει αποφάσεις σχεδιασμού βασισμένο στην ποιότητα διαφορετικών κομματιών του δικτύου.

Αν και καθένα από τα παραπάνω μοντέλα αναπτύχθηκαν ξεχωριστά, στην πραγματικότητα είναι ισχυρά αλληλεξαρτώμενα και ολοκληρωμένα. Στο σχήμα 21 απεικονίζεται η σύνθεση της δομής ενός απλού mobile agent.

Ο πυρήνας του βασίζεται στο υπολογιστικό μοντέλο. Αυτό έχει μεγάλη επίδραση στα υπόλοιπα μοντέλα. Καθορίζει πώς αντιμετωπίζονται άλλα agents, οι hosts και οι πόροι που αφορούν το μοντέλο ασφάλειας. Έτσι, σε όλες τις τρέχουσες mobile agent υλοποιήσεις που βασίζονται στη γλώσσα Java (τα Aglets είναι ένα από τα συστήματα αυτά) έχει υιοθετηθεί το task based μοντέλο κύκλου ζωής. Από την άλλη πλευρά, το υπολογιστικό μοντέλο των AgentTcl και Telescript είναι τέτοιο που υποστηρίζει ένα persistent process μοντέλο κύκλου ζωής. Το μοντέλο του κύκλου ζωής καθορίζει τις έγκυρες καταστάσεις για το agent. Τα μοντέλα ασφάλειας και κύκλου ζωής είναι και τα δύο δομημένα πολύ κοντά στον πυρήνα.

Το εξωτερικό στρώμα περιέχει το επικοινωνιακό μοντέλο, το μοντέλο πλοήγησης και το agent μοντέλο. Το agent μοντέλο καθορίζει το ευφρές κομμάτι του mobile agent, όπως τις λειτουργίες μάθησης και συνεργασίας. Το επικοινωνιακό μοντέλο στηρίζεται ισχυρά στο μοντέλο ασφάλειας. Το μοντέλο πλοήγησης είναι επίσης εξαρτώμενο από το μοντέλο ασφάλειας, καθώς το mobile agent κινείται από κόμβο σε κόμβο. Η διαδικασία μεταφοράς συχνά προκαλεί μετάβαση του agent σε διαφορετική κατάσταση του κύκλου ζωής, όπως συμβαίνει στα Aglets.



Σχήμα 21: Η δομή ενός mobile agent

ΜΕΘΟΔΟΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΚΑΙ ΝΕΑ ΠΡΟΣΕΓΓΙΣΗ

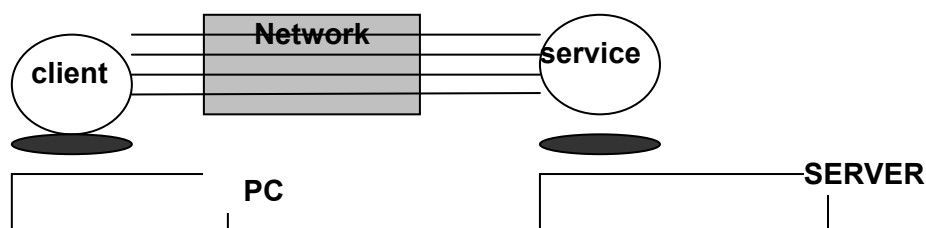
Πριν από τα mobile agents, είναι γνωστό ότι, άλλες διαδικασίες αλληλεπίδρασης και επικοινωνίας αναπτύχθηκαν και εξακολουθούν να εφαρμόζονται κατά κόρον στα καταναμημένα και γενικότερα, στα συστήματα δικτύων.

Υπάρχουν πολλοί τρόποι αλληλεπίδρασης, που υλοποιούν το κυρίαρχο μοντέλο πελάτη- εξυπηρετητή (client-server) με διαφορετικές μεθόδους. Οι σχετικές μέθοδοι είναι messaging, datagrams, sockets και Remote Procedure Call (RPC). Η κύρια διαφορά μεταξύ αυτών έγκειται στη χρήση ασύγχρονων ή σύγχρονων πρωτοκόλλων. Στο εξής, με τον όρο *messaging* θα γίνεται αναφορά στις ασύγχρονες αλληλεπιδράσεις client-server ενώ, με τον όρο *RPC* θα γίνεται αναφορά στις άλλες περιπτώσεις.

Στις δύο αυτές μορφές μεθόδων, ο client και ο server ανταλλάσσουν δεδομένα που είναι προς επεξεργασία από συγκεκριμένες διαδικασίες στην απομακρυσμένη CPU. Καμία πλευρά δεν ορίζει πώς να γίνει η επεξεργασία των δεδομένων. Αμφότερες, όμως, γνωρίζουν τις ικανότητες των απομακρυσμένων διεργασιών. Τα mobile agents διαφέρουν στο σημείο αυτό, επειδή ανταλλάσσουν είτε δεδομένα είτε διαδικασίες και εκμεταλλεύονται τις διαδικασίες είτε του client είτε του server.

Στην περίπτωση του *RPC*, ο client και server ανοίγουν ένα επικοινωνιακό κανάλι μεταξύ της εφαρμογής στον client και της διεργασίας του server. Οι *RPC* παράμετροι περνούν σε μία ρουτίνα διεπαφής, η οποία τις μετατρέπει σε μορφή κατάλληλη για μετάδοση και μετά, στέλνονται αποκλειστικά στη διεργασία του server. Τα *RPC* μηνύματα λαμβάνονται από την ανάλογη ρουτίνα διεπαφής στο server, αναλύονται και περνούν στην κατάλληλη διαδικασία του. Η διαδικασία επεξεργάζεται τις παραμέτρους και επιστρέφει τα αποτελέσματα στην client διεργασία. Και οι δύο πλευρές χρησιμοποιούν κοινή διεπαφή, ανεξάρτητα από την πιθανή ετερογένεια του υλικού ή του λογισμικού του λειτουργικού συστήματος (Σχήμα 22).

Ενώ, μία τοπική κλήση διαδικασίας εκτελείται το πολύ σε λίγα microseconds (χωρίς να περιληφθεί ο χρόνος εκτέλεσης της ίδιας της διαδικασίας), η *RPC* εισάγει χρονική επιβάρυνση (overhead) λόγω, της μετατροπής των δεδομένων για την μετάδοση, της μετάδοσης αυτών και κατόπιν, της ανάλυσής τους, γεγονός που επιφέρει καθυστέρηση. Όπως η τοπική κλήση διαδικασιών, η *RPC* είναι σύγχρονη. Ο client «αναβάλλει» την διεργασία, διατηρώντας την κατάσταση της στο σημείο που σταμάτησε, έως ότου λάβει την επιστρεφόμενη *RPC* από τον server. Για την ασφάλεια στην client-server επικοινωνία χρησιμοποιούνται οι μηχανισμοί κρυπτογράφησης και πιστοποίησης, οι οποίοι, όμως, προσθέτουν ακόμη μεγαλύτερη χρονική επιβάρυνση.



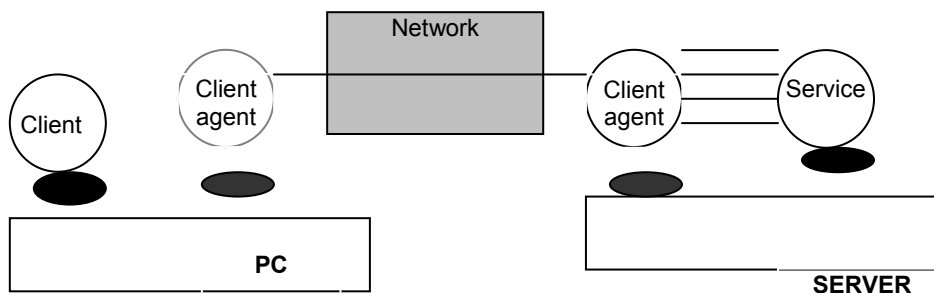
Σχήμα 22: Οι υπολογισμοί πελάτη-εξυπηρετητή με τη χρήση της μεθόδου

Η τεχνική *messaging* αποτελεί εξέλιξη των συστημάτων ηλεκτρονικού ταχυδρομείου και των προηγούμενων καταναμημένων υπολογιστικών σχημάτων, στα οποία οι εφαρμογές επικοινωνούσαν μέσω αρχείων ή μέσω διασωληνώσεων (pipes). Η εφαρμογή του client συνθέτει ένα μήνυμα, με τυπικά δομημένο κείμενο, για να παραδοθεί στον κατάλληλο, για τον συγκεκριμένο τύπο μηνύματος, επεξεργαστή λογισμικού. Τα *messaging* συστήματα διαθέτουν δική τους υπηρεσία μεταφοράς των μηνυμάτων. Ο τύπος του επεξεργαστή υποδεικνύεται στην επικεφαλίδα του μηνύματος. Το μήνυμα, γενικώς, προωθείται ανεξάρτητα από τον client, με την έννοια ότι αυτός μπορεί να μην γνωρίζει την ακριβή διεύθυνση δικτύου του server. Η ανάλυση των διευθύνσεων γίνεται σε ενδιάμεσα βήματα επεξεργασίας.

Το *messaging* είναι ασύγχρονο. Ο client, από την στιγμή που στέλνει το μήνυμα στο messaging υποσύστημα, συνεχίζει την εκτέλεσή του. Αν στο μέλλον λάβει ένα μήνυμα απάντησης από τον server, πρέπει να επανέλθει στην κατάσταση της εφαρμογής για την επεξεργασία της απάντησης. Η καθυστέρηση είναι μεγαλύτερη και λιγότερο προβλέψιμη από ότι στην περίπτωση *RPC*. Επομένως, το messaging είναι λιγότερο αποτελεσματικό για την επικοινωνία ενός προς έναν. Ωστόσο, για την επικοινωνία ενός προς πολλούς, όπως συνήθως συμβαίνει στην εξυπηρέτηση των clients από τους servers, η απόδοση είναι μεγαλύτερη, γιατί ο client δεν χρειάζεται να αναβάλλει την εκτέλεσή του περιμένοντας την απάντηση. Για την ασφαλή ανταλλαγή μηνυμάτων χρησιμοποιείται η κρυπτογράφηση και η πιστοποίηση της αυθεντικότητας των μελών επικοινωνίας. Αυτά επιφέρουν αύξηση της χρονικής επιβάρυνσης, αν και ο ρόλος της δεν είναι το ίδιο κρίσιμος με την προηγούμενη περίπτωση.

Η νέα πραγματικότητα εισάγει το **Remote Programming (RP)**. Η RP προσέγγιση επιτρέπει σε έναν υπολογιστή, όχι μόνο να καλεί διαδικασίες ενός άλλου απομακρυσμένου υπολογιστή, αλλά και να παρέχει τις διαδικασίες προς εκτέλεση. Επιτρέπει, δηλαδή, όχι μόνο τη μεταφορά του ελέγχου, αλλά και του κώδικα προς εκτέλεση.

Κάθε μήνυμα που μεταφέρει το δίκτυο συνίσταται στην διαδικασία που πρόκειται να εκτελεστεί από τον απομακρυσμένο υπολογιστή, καθώς και στα δεδομένα που εισάγονται ως παράμετροι της διαδικασίας. Η διαδικασία, μπορεί να είναι μία που έχει ξεκινήσει στον αποστολέα υπολογιστή ή απλά έχει συνεχιστεί στον αποστολέα από προηγούμενη επικοινωνία, και καλείται να συνεχιστεί στον παραλήπτη, δηλαδή στον απομακρυσμένο υπολογιστή. Η «μετανάστευση» σε άλλον host μπορεί να είναι είτε παθητική, με την έννοια ότι κάποιος άλλος την επιβάλλει, είτε ενεργή, όπου η ίδια η διαδικασία αποφασίζει πότε και πού θα μεταβεί (Σχήμα 23).



Σχήμα 23: Οι υπολογισμοί πελάτη-εξυπηρετητή με τη χρήση του Remote Programming.

Οι δύο υπολογιστές που ακολουθούν την **RP** προσέγγιση, εκ των προτέρων συμφωνούν σε μία γλώσσα επικοινωνίας. Η γλώσσα περιλαμβάνει εντολές, που επιτρέπουν σε μία διαδικασία να λαμβάνει αποφάσεις, να ελέγχει και να αλλάζει την κατάστασή της, ενώ ταυτόχρονα, να καλεί άλλες διαδικασίες του υπολογιστή στον οποίο βρίσκεται. Τέτοιες κλήσεις είναι τοπικές. Η διαδικασία μαζί με την κατάστασή της, αποτελούν ένα mobile agent, δίνοντας έμφαση στο γεγονός ότι αντιπροσωπεύουν τον υπολογιστή «αποστολέα», αν και βρίσκονται στον απομακρυσμένο υπολογιστή «παραλήπτη».

Το ιδιαίτερο χαρακτηριστικό του **Remote Programming** είναι ότι, κατόπιν μεταφοράς του agent από το δίκτυο, ο υπολογιστής του χρήστη και του server μπορούν να αλληλεπιδρούν

χωρίς να χρησιμοποιούν το δίκτυο. Αυτό αναδεικνύει πολλά πλεονεκτήματα χρήσης των mobile agents σε ευρύ φάσμα εφαρμογών που χρησιμοποιούν τα δίκτυα και ειδικότερα, το Internet για την επικοινωνία.

Πλεονεκτήματα

Γενικά, δεν υπάρχουν λειτουργίες που ακολουθούν το μοντέλο client-server, οι οποίες να μην υλοποιούνται αποκλειστικά από την τεχνολογία των mobile agents. Ωστόσο, σε κάθε περίπτωση, τα mobile agents παρουσιάζουν πλεονεκτήματα στα στάδια σχεδιασμού, υλοποίησης και εκτέλεσης. Τα πλεονεκτήματα αυτά μπορούν να χαρακτηριστούν μεγάλης σπουδαιότητας για την αναμόρφωση της τεχνολογίας λογισμικού σύμφωνα με τις ανάγκες της σύγχρονης και μελλοντικής πραγματικότητας.

➤ **Αποδοτικότητα.** Τα mobile agents καταναλώνουν λιγότερους πόρους δικτύου, εφόσον μετακινούν το πρόγραμμα στα δεδομένα κι όχι τα δεδομένα στο πρόγραμμα. Επίσης, μπορούν να χειρίζονται μεγάλη ποσότητα δεδομένων, εφόσον αυτά είναι αποθηκευμένα στον εκάστοτε υπολογιστή και η επεξεργασία τους από τα agents γίνεται τοπικά.

➤ **Μείωση της κίνησης δικτύου.** Τα περισσότερα πρωτόκολλα επικοινωνιών αναμειγνύουν διάφορες αλληλεπιδράσεις, ιδιαίτερα όταν εφαρμόζονται και τα κατάλληλα μέτρα ασφάλειας. Αυτό προκαλεί μεγάλη κίνηση δικτύου. πχ. Στην περίπτωση ασφαλούς RPC, για μια περίπλοκη συναλλαγή λαμβάνουν χώρα μερικές δεκάδες ανταλλαγές μηνυμάτων. Με τα mobile agents κάποιος μπορεί να στείλει τον αντιπρόσωπό του στον απομακρυσμένο host και οι αλληλεπιδράσεις να λάβουν χώρα τοπικά, γεγονός ιδιαίτερα σημαντικό για τα δίκτυα περιορισμένου εύρους ζώνης.

➤ **Ασύγχρονη αυτόνομη αλληλεπίδραση.** Οι εργασίες που απαιτούν απομακρυσμένη αλληλεπίδραση μπορούν να κωδικοποιηθούν μέσα στα mobile agents και κατόπιν, αυτά να μετακινηθούν στον κατάλληλο host. Τα mobile agents λειτουργούν ασύγχρονα και ανεξάρτητα από το πρόγραμμα που τα έστειλε. Ένα τέτοιο παράδειγμα είναι το εξής: Ένας χρήστης στέλνει το αυτόνομο agent του να ψάξει για πληροφορίες μέσα στο δίκτυο και έπειτα, διακόπτει την σύνδεση. Οποιαδήποτε άλλη στιγμή επανασυνδεθεί μπορεί να λάβει τα αποτελέσματα του agent.

➤ **Αλληλεπίδραση με συστήματα πραγματικού χρόνου.** Οι οντότητες πραγματικού χρόνου, όπως το λογισμικό που ελέγχει έναν ΑΤΜ μεταγωγέα ή ένα σύστημα ασφάλειας σε πυρηνικές εγκαταστάσεις, απαιτούν άμεσες αντιδράσεις στις αλλαγές του περιβάλλοντός τους. Ο έλεγχος των οντοτήτων αυτών μέσω ενός τεράστιου δικτύου, επιφέρει σημαντικές καθυστερήσεις. Σε κρίσιμες περιπτώσεις (όπως τον έλεγχο πυρηνικού συστήματος), τέτοιες καθυστερήσεις είναι ανεπίτρεπτες. Τα ευφυή mobile agents παρέχουν την εναλλακτική λύση διότι μπορούν να αποσπαστούν από ένα κεντρικό σύστημα, για να ελέγχουν τα συστήματα πραγματικού χρόνου τοπικά και επιπλέον, να εκτελούν εντολές καθοδήγησης από το κεντρικό σύστημα. Άλλο παράδειγμα αποτελεί ο έλεγχος λογισμικού για διαστημικές εξερευνήσεις του ηλιακού συστήματος. Το πρόγραμμα του agent που εκτελείται τοπικά, ακόμη κι αν μεταφράζεται, έχει σχετικά μικρή και περιορισμένη καθυστέρηση, οπότε μπορεί να εκμεταλλευτεί ευκαιρίες για τη διόρθωση των σφαλμάτων.

➤ **Υποστήριξη ετερογενών περιβαλλόντων.** Οι υπολογιστές και τα δίκτυα πάνω στα οποία δομούνται τα συστήματα mobile agent έχουν ετερογενή χαρακτήρα. Η χρήση μιας scripting γλώσσας για ανταλλαγή προγραμμάτων και δεδομένων, καθιστά την αναπαράσταση τους ανεξάρτητη πλατφόρμας, από την στιγμή που το script περιβάλλον έχει εισαχθεί στις αντίστοιχες πλατφόρμες. Έτσι τα mobile agents είναι ανεξάρτητα υπολογιστών και δικτύων, και επιτυγχάνουν τη διαφάνεια στη λειτουργικότητα.

➤ **Δρομολόγηση βάσει του σημασιολογικού περιεχομένου.** Οι σημερινές client-server αλληλεπιδράσεις απαιτούν λεπτομερή γνώση των λειτουργιών των εφαρμογών τους, των πρωτοκόλλων επικοινωνίας και των διευθύνσεων των δύο μερών. Τα mobile agents

διευκολύνουν την διαδικασία αυτή απελευθερώνοντας τον client και τον χρήστη από τέτοιες «έννοιες». Ένας χρήστης μπορεί να διατυπώσει μία αίτηση για κάποια πληροφορία ή υπηρεσία σε φυσική γλώσσα και η αίτηση να υποβληθεί σε ένα mobile agent. Το τελευταίο αναμορφώνει την αίτηση σύμφωνα με το λεξιλόγιο και την σύνταξη της γλώσσας επικοινωνίας των agents. Μπορεί, επίσης, να συμβουλευτεί άλλους agents ή να χρησιμοποιήσει δικό του ευρετήριο για να αναγνωρίσει έναν ή περισσότερους servers που είναι ικανοί να τον εξυπηρετήσουν. Το mobile agent μεταβαίνει σε καθέναν από αυτούς, προωθεί την αίτηση και επιστρέφει τα αποτελέσματα στον χρήστη. Έτσι, η αίτηση που υποβάλλεται από τον χρήστη, δρομολογείται βάσει του σημασιολογικού περιεχομένου της.

➤ **Ευρωστία και ανεκτικότητα στα λάθη.** Τα mobile agents έχουν την ικανότητα να αντιδρούν αυτόνομα σε αλλαγές του περιβάλλοντός τους. Η δυναμική προσαρμογή τους σε διάφορες καταστάσεις, καθιστά εύκολη τη διαμόρφωση ανεκτικής συμπεριφοράς τους σε λάθη, ιδιαίτερα σε ένα καταναμημένο περιβάλλον. Όσον αφορά την ευρωστία, πλεονεκτούν από δύο απόψεις: Από πλευράς ανταλλαγής μηνυμάτων, παρέχουν αξιόπιστη μεταφορά, χωρίς να απαιτούν αξιόπιστη επικοινωνία. Από πλευράς ανάκτησης δεδομένων, τα mobile agents μπορούν να χειριστούν την κατάσταση στην οποία ο ζητούμενος server δεν είναι διαθέσιμος ή είναι εκτός υπηρεσίας, αφού έχουν τη δυνατότητα να μεταναστεύουν σε άλλους κατάλληλους servers, που τους γνωρίζουν είτε από μόνοι τους είτε κατόπιν επικοινωνίας τους με άλλους agents.

➤ **Διευκόλυνση της ανάκτησης πληροφοριών.** Είναι αποδοτικότερο για το πρόγραμμα που ανακτά πληροφορίες, να πηγαίνει στην πηγή των δεδομένων, αντί να στέλνονται τα δεδομένα στο πρόγραμμα, ειδικότερα όταν το πρόγραμμα αυτό μπορεί, επιπλέον, να φιλτράρει και να συνοψίζει τα δεδομένα. Ένα agent έχει τη δυνατότητα να διαμορφώνει αυτόματα ευρετήριο δεδομένων, ξεχωρίζοντας τα ενδιαφέροντα για τον χρήστη ή/και μπορεί να αναγνωρίζει δεδομένα που ενδιαφέρουν άλλα agents και να τα πληροφορεί για αυτά. Η διαφορά μεταξύ της ανάκτησης πληροφοριών βάσει της σημασιολογίας τους και βάσει απλών λέξεων κλειδιών, συνίσταται στο ποσό της πληροφορίας που μπορεί να περάσει από το σύστημα για να του επιτρέψει να γίνει ένα πραγματικά ποιοτικό «έμπειρο σύστημα». Θα μπορούσε κάποιος να αντιπαραθέσει προγράμματα φιλτραρίσματος τα οποία, μαζί με την αίτηση του χρήστη στέλνουν ένα προφίλ του χρήστη από προηγούμενες σχετικές αναζητήσεις. Όμως, το ποσό των δεδομένων που διακινούνται στο δίκτυο αυξάνεται, ειδικότερα όταν αυτό το ποσό πρέπει να διανεμηθεί σε πολλούς servers.

➤ **Εξατομίκευση της συμπεριφοράς του server.** Οι προγραμματιστές των mobile agents μπορούν να επεκτείνουν την λειτουργικότητα που προσφέρεται από τα προγράμματα των διαθέσιμων servers, έτσι ώστε να προσαρμόζονται στις συνήθεις εργασίες του χρήστη-client. Στον τομέα των Ευφυών Δικτύων, τα mobile agents παρέχουν τη δυνατότητα εξατομίκευσης της συμπεριφοράς οντοτήτων του δικτύου (πχ. των δρομολογητών) με την δυναμική προσαρμογή τους σε νέα συμπεριφορά, ανάλογα με τις τρέχουσες συνθήκες.

➤ **Αποτελούν ένα χρήσιμο μοντέλο ανάπτυξης.** Ο σχεδιασμός και η δόμηση των καταναμημένων συστημάτων μπορεί να γίνει ευκολότερα με τη χρήση των mobile agents. Αυτό, γιατί είναι από τη φύση τους καταναμημένα αντικείμενα, τα οποία, όμως, έχουν εκτελέσιμο περιεχόμενο και κινούνται, παρέχοντας μία νέα άποψη ενός καταναμημένου συστήματος,

➤ **Υποστήριξη των κινούμενων πελατών.** Η τεχνολογία των mobile agents ευνοεί τους χρήστες κινητών συσκευών. Οι κινητές συσκευές, όπως οι υπολογιστές laptop, τα notebooks κλπ., έχουν χαρακτηριστικά, όπως:

- Η σύνδεσή τους με το δίκτυο παρουσιάζει διαλείψεις και η πρόσβαση σε έναν server μπορεί να διακόπτεται (γρήγορη αλλαγή θέσεων, ποικιλία QoS παραμέτρων).
- Χρησιμοποιούν, συνήθως, συνδέσεις περιορισμένου εύρους ζώνης.
- Έχουν περιορισμένες ικανότητες αποθήκευσης και επεξεργασίας.

Το πλεονέκτημα των mobile agents συνίσταται στην δυνατότητα του client, καθόσον δεν είναι συνδεδεμένος, να αναπτύξει έναν agent με συγκεκριμένο στόχο (πχ. αίτηση για μία υπηρεσία που παρέχεται από κάποιον server), να τον στείλει στον προορισμό κατά την διάρκεια μιας συνόδου σύνδεσης και μετά, να αποσυνδεθεί. Την απάντηση μπορεί να την λάβει αργότερα, σε μία επόμενη σύνοδο σύνδεσης. Επίσης, το mobile agent μπορεί να αναλάβει την ανάκτηση των

πληροφοριών, το φιλτράρισμά τους και την επιστροφή του αποτελέσματος στον client. Έτσι, αφενός οι πληροφορίες που μεταφέρονται μέσω του δικτύου μειώνονται, αφετέρου, η κινητή συσκευή απελευθερώνεται από την διαδικασία επεξεργασίας μεγάλου όγκου δεδομένων.

Είναι φανερό λοιπόν, ότι η ανάπτυξη των mobile agents μπορεί να συμβάλλει ουσιαστικά στη λύση προβλημάτων και τη βελτίωση της υπάρχουσας τεχνολογίας, ανοίγοντας νέες δυνατότητες στους χρήστες της. Το τίμημα, όμως, για τα οφέλη που επιφέρουν τα mobile agents αποτελεί η πρόκληση στο θέμα της ασφάλειας, παράγοντας αντισταθμιστικός για το γρήγορο ρυθμό αποδοχής τους.

Ασφάλεια

Το θέμα της ασφάλειας στα δίκτυα και, ειδικότερα, στο Internet, έχει γίνει στις μέρες μας μείζονος σημασίας, περισσότερο από κάθε άλλη περίοδο. Τα mobile agents, όπως αναφέρθηκε αναλυτικά παραπάνω, προσφέρουν πολλά πλεονεκτήματα και διευκολύνσεις, στις συναλλαγές και σε άλλες εφαρμογές του δικτυακού περιβάλλοντος. Ωστόσο, η ασφάλεια είναι ένας από τους παράγοντες κλειδιά για την εξέλιξη της mobile agent τεχνολογίας. Όπως κάθε άλλος εκτελέσιμος κώδικας, τα mobile agents μπορούν να αποτελέσουν απειλή για ένα σύστημα. Αλλά, και τα ίδια είναι εκτεθειμένα σε απειλές από το σύστημα που τα φιλοξενεί, κατάσταση που διαφέρει από τα προβλήματα που αντιμετωπίζονται με τα παραδοσιακά συστήματα ασφαλείας.

Ένα παράδειγμα, όπου είναι φανεροί οι κίνδυνοι, είναι το εξής: Ένας χρήστης στέλνει ένα mobile agent για να κάνει κράτηση εισιτηρίου σε ένα ταξιδιωτικό γραφείο. Ένας «κακοπροαίρετος» host μπορεί να κλέψει από το agent και να χρησιμοποιήσει εμπιστευτικές πληροφορίες του χρήστη. Μπορεί κάποιος τρίτος να ξεγελάσει το mobile agent (WebSproofing), οπότε να επιστρέψει λανθασμένα αποτελέσματα. Με τον ίδιο τρόπο, το agent μπορεί να προσποιηθεί στον server, παριστάνοντας το agent του χρήστη στο server ή να προκαλέσει ζημιά στο host, περίπτωση στην οποία η ευθύνη αποδίδεται στο χρήστη.

8.1 Περιπτώσεις Ασφάλειας

Στο πρόβλημα της ασφάλειας που αφορά τα mobile agents διακρίνονται οι εξής περιπτώσεις:

α) Προστασία των mobile agents

- Απομακρυσμένος host απειλεί το agent: Ένα agent, καθώς κινείται, επισκέπτεται έναν μη έμπιστο server που προσπαθεί να εξάγει ιδιωτικές πληροφορίες ή να εξαπατήσει το agent.
- Agent απειλεί άλλο agent: Ένα agent αλληλεπιδρά με κάποιο από τα agents του χρήστη είτε για να εξάγει ιδιωτικές πληροφορίες είτε για να σταματήσει την εκτέλεσή του.
- Μη εξουσιοδοτημένες οντότητες απειλούν το agent: Μία κακοπροαίρετη οντότητα μπορεί να τροποποιήσει τα μηνύματα που ανταλλάσσονται μεταξύ agent-server ή να μπλοκάρει την επικοινωνία μεταξύ τους ή να αποκρύψει το περιεχόμενο των μηνυμάτων από το agent.

β) Προστασία των κόμβων από «κακόβουλα» agents

Ένα mobile agent σύστημα είναι ένα ανοικτό σύστημα, οπότε οι hosts υπόκεινται σε ποικιλία απειλών:

- Εισερχόμενα agents απειλούν τον host : Ένας εισερχόμενο agent που επισκέπτεται έναν server προσπαθεί να προσπελάσει δεδομένα ή να αποκτήσει πρόσβαση σε ιδιωτικά αρχεία του host ή ακόμη και να διακόψει τη λειτουργία του server.
- Μη εξουσιοδοτημένα τρίτα πρόσωπα απειλούν τον host: Μία κακοπροαίρετη οντότητα μπορεί να στείλει μεγάλο αριθμό από “spam” agents στον server για να τον θέσουν εκτός εξυπηρέτησης.

Ένα mobile agent είναι μοναδικό για κάθε κόμβο στον οποίο εκτελείται ο κώδικάς του. Κατά την εκτέλεσή του, αυτόματα έχει πρόσβαση σε μερικούς από τους πόρους του host. Με αυτό το επίπεδο πρόσβασης τα mobile agents μπορούν να κάνουν επιθέσεις τροποποιώντας άλλα τοπικά agents, διαδίδοντας ιούς, παριστάνοντας άλλους χρήστες κ.α.

γ) Προστασία του δικτύου

- Εισερχόμενο agent απειλεί το δίκτυο: Ένα mobile agent αρχίζει να πολλαπλασιάζεται και καθώς ταξιδεύει εξαπλώνεται προσπαθώντας να υπερχειλίσει το δίκτυο από agents.

Έτσι, από την mobile agent τεχνολογία αναδύουν νέα θέματα προβληματισμού για την ασφάλεια, αν και, με τη χρήση της υπάρχουσας τεχνολογίας ασφαλείας, μπορούν να λυθούν τα

περισσότερα από αυτά. Επιπλέον, με κατάλληλα μοντέλα ασφάλειας στη δομή των εκάστοτε mobile agents μπορούν να αντιμετωπιστούν οι ιδιόμορφες νέες καταστάσεις.

8.2 Ταξινόμηση των Επιθέσεων

Οι πιθανές επιθέσεις που σχετίζονται με τα mobile agents και τους hosts που επισκέπτονται διακρίνονται σε δύο κατηγορίες:

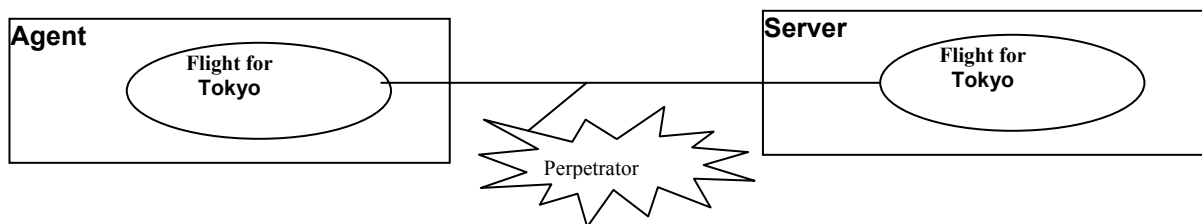
- Στις **παθητικές επιθέσεις**, που δεν τροποποιούν τα agents ή άλλες πληροφορίες.
- Στις **ενεργές επιθέσεις**, που επιδρούν στα agents.

Σε μία ενεργή επίθεση, μπορεί κανείς να αντιληφθεί ότι κάτι συμβαίνει στο agent του, ενώ δεν υπάρχει τέτοια περίπτωση σε μία παθητική επίθεση.

8.2.1 Παθητικές Επιθέσεις

Συνήθως είναι επιθέσεις εναντίον των επικοινωνούντων συστημάτων με agents και των ίδιων των agents που μεταφέρονται στο δίκτυο. Επειδή δεν γίνονται αλλαγές στα agents, τέτοιες επιθέσεις είναι δύσκολο να ανιχνευτούν. Αυτές μπορεί να είναι:

Eavesdropping: Αυτό το είδος επίθεσης συνήθως χρησιμοποιεί ένα πρόγραμμα που λέγεται επόπτης επικοινωνίας (communication monitor). Ένας επόπτης παρακολουθεί τις πληροφορίες που ανταλλάσσονται μεταξύ των agent συστημάτων και «συλλαμβάνει» agents και μηνύματα που ενδεχομένως περιέχουν χρήσιμες πληροφορίες (Σχήμα 24). Για να προστατευτούν οι agent χρήστες πρέπει να χρησιμοποιούν την κρυπτογράφηση.



Σχήμα 24: Η παθητική επίθεση Eavesdropping.

Ανάλυση Κίνησης: Είναι μια διαφοροποιημένη μορφή του *eavesdropping* που επιτρέπει σε αυτόν που επιτίθεται να αναλύει τη μορφή της κίνησης των agents που ανταλλάσσονται μεταξύ των συστημάτων (πχ. αλλαγές στο ποσό της κίνησης ή αλλαγές στην συχνότητα εισερχόμενων-εξερχόμενων agents), έτσι ώστε να κάνει υποθέσεις ή να εξαγάγει συμπεράσματα από τις μορφές αυτές. Αυτό το είδος επίθεσης είναι επικίνδυνο, ακόμη κι αν τα περιεχόμενα είναι κρυπτογραφημένα.

8.2.2 Ενεργές Επιθέσεις

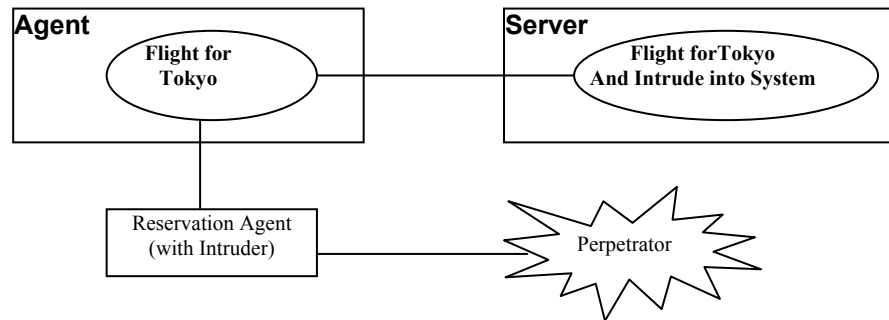
Οι επιθέσεις αυτές προκαλούν μεγάλη ποικιλία απειλών, που εκτείνονται από την απλή τροποποίηση των δεδομένων των mobile agents έως την εισβολή «κακών» mobile agents σε άγνωστους servers. Αυτές μπορεί να είναι:

Μη εξουσιοδοτημένη πρόσβαση: Ένα mobile agent αποκτά πρόσβαση σε πληροφορίες που δεν του επιτρέπονται. Αυτό μπορεί να συμβεί αν το agent «μεταμφιεστεί» σε έναν έμπιστο χρήστη ή να εκτελείται σε μη ασφαλές περιβάλλον.

Masquerade: Μία οντότητα προσποιείται ότι είναι μία άλλη. Για παράδειγμα, ένα agent εισέρχεται σε έναν server παριστάνοντας φαινομενικά ένα έμπιστο πρόσωπο ή οργανισμό. Αν πετύχει να εξαπατήσει το host, μπορεί να χρησιμοποιήσει δωρεάν κάποια υπηρεσία ή να κλέψει εμπιστευτικές πληροφορίες που θα κρατούσε η υποτιθέμενη οντότητα.

Trojan Horse: Ένα *Trojan horse* είναι ένα agent που εκτελείται από νόμιμο χρήστη, αλλά κάνει κάτι διαφορετικό από αυτό που ο χρήστης περιμένει ή ισχυρίζεται. Ένας εισβολέας

μπορεί εύκολα να δημιουργήσει έναν τέτοιο agent, ο οποίος φαίνεται αθώο, αλλά είναι κακό για τους χρήστες του (Σχήμα 25).



Σχήμα 25: Η ενεργή επίθεση Trojan Horse.

Τροποποίηση: Ένα agent ή ένα μήνυμα μεταξύ των agent συστημάτων διαγράφεται ή αλλάζεται κατά την διάρκεια της μεταφοράς. Αυτό, επίσης, μπορεί να γίνει κι από κάποιο host που επισκέπτεται το agent, ο οποίος του αλλάζει τα δεδομένα που έχει συλλέξει από προηγούμενους hosts. Οποιαδήποτε πληροφορία που τροποποιείται απρόσμενα με τέτοιο τρόπο, μπορεί να αλλάξει την συμπεριφορά του agent σε «κακόβουλη» ή απλώς, να κάνει το agent να επιστρέψει λανθασμένα αποτελέσματα.

Αντιγραφή: Ένα mobile agent που στέλνεται νόμιμα σε έναν host, αντιγράφεται και επαναχρησιμοποιείται από κάποιον παράνομο χρήστη. Ο τελευταίος μπορεί να λάβει τα ίδια αποτελέσματα ή να παρενοχλήσει τον server.

Εξόντωση Πόρων: Ένας πόρος χρησιμοποιείται τόσο πολύ από κάποιον που, τελικά, διακόπτεται η εξυπηρέτηση προς τους άλλους χρήστες. Αυτή η επίθεση λέγεται και denial of service (DOS). Ο πόρος μπορεί να είναι το εύρος ζώνης δικτύου, η μνήμη ή η κεντρική μονάδα επεξεργασίας του server. Ένα παράδειγμα είναι η χρήση agents που διαρκώς κλωνοποιούνται και γρήγορα γεμίζουν τον server ή το δίκτυο.

Αρνηση Αναγνώρισης: Σε μία επικοινωνία για συναλλαγή, το ένα μέλος αρνείται ότι έγινε συναλλαγή. Για παράδειγμα, το agent ενός χρήστη αγοράζει κάτι μέσω κάποιου agent server και αργότερα, ο server αρνείται ότι έγινε η συναλλαγή.

8.3 Υπηρεσίες Ασφάλειας –Τρέχουσα Κατάσταση

Οι υπηρεσίες ασφάλειας που διαθέτει η υπάρχουσα εφαρμοσμένη τεχνολογία μπορούν να συμβάλλουν σημαντικά στην προστασία των mobile agents και των servers από τις επιθέσεις. Παρακάτω παρουσιάζονται οι γνωστότερες από αυτές τις υπηρεσίες που εφαρμόζονται για την ασφάλεια των agent συστημάτων.

Πιστοποίηση (Authentication). Πριν την αποδοχή ενός εισερχόμενου mobile agent σε ένα σύστημα, ο διαχειριστής του ίσως θέλει να γνωρίζει σε ποιόν ανήκει το agent. Σε αυτή την περίπτωση απαιτείται authentication του agent. Πριν την εκτέλεση ενός εισερχόμενου mobile agent, ο server χρειάζεται να γνωρίζει ποιος είναι υπεύθυνος για την υλοποίηση του mobile agent, δηλαδή απαιτεί authentication του κώδικα. Για τους σκοπούς αυτούς χρησιμοποιούνται συνήθως οι ψηφιακές υπογραφές. Όμως, κι ο χρήστης χρειάζεται να αποδεικνύει την αυθεντικότητά του στον δεδομένο server ή το αντίστροφο, ο server να αποδεικνύει ότι είναι αυτός που ισχυρίζεται. Αυτό γίνεται με την κρυπτογράφηση δημοσίου κλειδιού (public-key encryption) ή με τη χρήση κάποιου password.

Ακεραιότητα (Integrity). Προκειμένου ο χρήστης να βασίζεται στις υπηρεσίες ή τα αποτελέσματα ενός mobile agent, πρέπει να είναι σίγουρος ότι κανένας δεν έχει πειράξει την κατάσταση και τον κώδικά του. Ο έλεγχος της ακεραιότητας του agent είναι η τεχνική επαλήθευσης στο ότι δεν έχουν γίνει τροποποιήσεις του κώδικα και της κατάστασης του agent

από τρίτους. Πραγματοποιείται, επίσης, με την κρυπτογράφηση και την χρήση ψηφιακής υπογραφής, αν και ο έλεγχος του agent είναι ακόμη θέμα που δεν έχει ξεκάθαρη λύση.

Εμπιστευτικότητα (Confidentiality). Ένα mobile agent μπορεί να μεταφέρει εμπιστευτικές πληροφορίες για να διαβαστούν από κάποιον συγκεκριμένο server ή agent. Για την τήρηση της μυστικότητας από τους άλλους servers ή agents, το mobile agent μπορεί να ζητήσει από τον server να το μεταφέρει με μυστικό τρόπο (πχ. κρυπτογράφηση, χρήση SSL) για την αποφυγή του eavesdropping.

Εξουσιοδότηση (Authorization). Με τον τρόπο αυτό μπορεί να καθοριστεί και να επιβληθεί έλεγχος στο agent όσον αφορά στην πρόσβαση πληροφοριών και στην δυνατότητα χρήσης των υπηρεσιών του server.

Αποδοχή Αναγνώρισης (Nonrepudiation). Βάσει της υπηρεσίας αυτής ούτε ο agent ούτε ο server μπορούν να αρνηθούν την ταυτότητά τους σε μία συναλλαγή που έγινε κατά την επικοινωνία τους.

Ακρόαση (Auditing). Η υπηρεσία auditing καταγράφει τις ενέργειες ενός agent που σχετίζονται με την ασφάλεια για περαιτέρω μελλοντική παρακολούθηση.

Ωστόσο, υπάρχουν όρια στα μέτρα ασφαλείας των mobile agents. Η προστασία τους στηρίζεται στον server τον οποίο επισκέπτονται και στο περιβάλλον υποδοχής τους. Αν και είναι αυτόνομα, δεν είναι ανεξάρτητα και χρειάζονται επεξεργαστές για να μπορούν να εκτελούν τα προγράμματά τους. Ο server είναι αποκλειστικά υπεύθυνος για την εκτέλεση του mobile agent. Έχει την δυνατότητα να μην του επιτρέψει να μεταβεί σε έναν ανταγωνιστικό server. Ένα mobile agent δεν μπορεί να παράγει μυστικές πληροφορίες χωρίς την βοήθεια του μεταφραστή (του server) κι έτσι, δεν μπορεί να διατηρήσει κρυφό από τον server οτιδήποτε δημιουργεί στην πλατφόρμα του. Αυτό σημαίνει, ότι τα mobile agents δεν μπορούν να διατηρήσουν κρυφές, από τον server, τις επικοινωνίες τους με τα άλλα agents.

Επιπλέον, στην προσπάθεια για την επαλήθευση του agent (ότι δεν τροποποιήθηκε) η χρήση της ψηφιακής υπογραφής δεν εγγυάται για την συμπεριφορά του agent. Η κατάσταση ενός mobile agent ποικίλει στην διάρκεια του κύκλου ζωής του κι επομένως, είναι πολύ δύσκολο να επαληθευτεί.

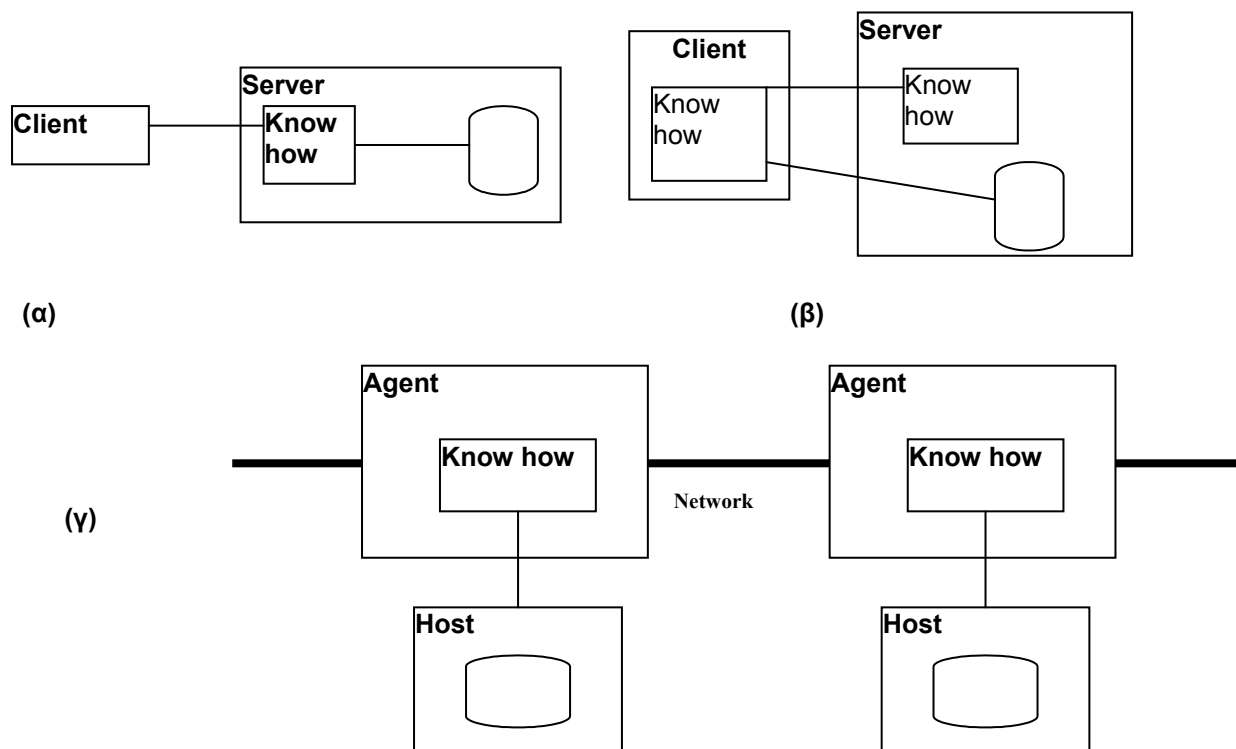
Παρόλα αυτά, πέρα από τις μεθόδους ασφαλείας που αναφέρθηκαν, το δομικό μοντέλο των mobile agents εμπεριέχει ένα μοντέλο ασφαλείας. Ανάλογα με το χώρο εφαρμογής της mobile agent τεχνολογίας, κάθε υλοποίηση προτείνει το δικό της μοντέλο ασφαλείας, το οποίο καθορίζει συγκεκριμένες πολιτικές για τα agents και τους hosts, παρέχοντας πρόσθετες μεθόδους ασφαλείας για τα mobile agent συστήματα.

ΕΦΑΡΜΟΓΕΣ

Τα mobile agents παρουσιάζουν ιδιαίτερα χαρακτηριστικά και πλεονεκτήματα σε συγκεκριμένες, διακριτές περιοχές μελέτης. Όμως, για κάθε μία από αυτές μπορεί να βρεθεί αντίστοιχη λύση της παραδοσιακής τεχνολογίας που δεν απαιτεί την χρήση των mobile agents και δεν εμφανίζει μεγάλα μειονεκτήματα. Το γεγονός αυτό αποτέλεσε αντικείμενο συζήτησης για τους ερευνητές της συγκεκριμένης τεχνολογίας, για να αναδειχθεί το αθροιστικό πλεονέκτημα των mobile agents:

- Παρέχουν ένα «διαχυτικό», ανοικτό, γενικευμένο πλαίσιο για ανάπτυξη και εξατομίκευση των υπηρεσιών δικτύου.
- Κανένας από τους εναλλακτικούς τρόπους υλοποίησης από μόνος του δεν συνδυάζει την λειτουργικότητα που προσφέρει το πλαίσιο δομής του mobile agent.
- Επιπροσθέτως στην αποτελεσματική υποστήριξη των υπαρχόντων υπηρεσιών, η τεχνολογία των mobile agents εισάγει νέες παραγωγικές υπηρεσίες δικτύων κι επομένως, νέους ορίζοντες στον τομέα των επιχειρήσεων.
- Δεδομένου ότι παρέχουν αποτελεσματικό τρόπο για την εύρεση υπηρεσιών και πληροφοριών, τα mobile agents ενισχύουν την πλευρά του χρήστη και αναμένεται να κυριαρχήσουν στην κοινότητα του Internet

Με άλλα λόγια, η σπουδαιότητα των mobile agents έγκειται στην ευρύτητα των δυνατών εφαρμογών τους, δίνοντας έναν μοναδικό τρόπο υλοποίησης, ταυτόχρονα, της παραδοσιακής κατακευματισμένης επεξεργασίας, του μοντέλου client/server και του μοντέλου κινητού κώδικα (Σχήμα 26).



Σχήμα 26: (α) Το μοντέλο client-server. (β) Το μοντέλο κινητού κώδικα (Code-on-Demand). (γ) Το παράδειγμα των mobile agents.

Η τεχνολογία των mobile agents επιτρέπει την ευελιξία εφαρμογής της σε πληθώρα περιπτώσεων, οι γενικότερες των οποίων παρουσιάζονται παρακάτω:

Συλλογή δεδομένων από πολλές τοποθεσίες

Μία από τις κυριότερες διαφορές μεταξύ κινητού κώδικα – όπως είναι τα applets- και ενός mobile agent, είναι ότι ο κώδικας μπορεί να μεταφερθεί από ένα σημείο A σε ένα άλλο σημείο B, ενώ το agent μπορεί να ταξιδεύει ακολουθιακά, σε πολλά μέρη. Ένα παράδειγμα αυτής της εφαρμογής είναι ένα backup εργαλείο δικτύου, που περιοδικά πρέπει να ψάχνει σε κάθε υπολογιστή του δικτύου για κάθε νέο δίσκο που πιθανώς να έχει προστεθεί. Εδώ, ένα mobile agent μπορεί να περιφέρεται στο δίκτυο, συλλέγοντας πληροφορίες για την backup κατάσταση κάθε υπολογιστή και στο τέλος, να επιστρέφει στην αφετηρία του δίνοντας μία αναφορά Ανάλογο παράδειγμα, είναι η χρήση των mobile agents στη διακίνηση πληροφοριών μεταξύ των συνεργατών ενός οργανισμού, ανεξάρτητα των ετερογενών συστημάτων ή των ειδικών εφαρμογών τους.

Ανίχνευση και φιλτράρισμα

Με δεδομένη τις συνεχώς αυξανόμενες πληροφορίες που διατίθενται στο Internet και σε άλλα δίκτυα, η συλλογή χρήσιμων πληροφοριών, απαιτεί ψάξιμο μέσα σε τεράστιο όγκο δεδομένων. Το φιλτράρισμα των άσχετων πληροφοριών είναι, συνήθως, χρονοβόρα διαδικασία. Ένα mobile agent μπορεί να επισκέπτεται πολλές σελίδες, να ψάχνει, βάσει συγκεκριμένου κριτηρίου, στις διαθέσιμες πληροφορίες και να κατασκευάζει ένα ευρετήριο από συνδέσμους για αυτές που ταιριάζουν στο κριτήριο αναζήτησης που χρησιμοποιεί. Παράδειγμα τέτοιας εφαρμογής, είναι ένας mobile agent που γνωρίζει τις προτιμήσεις του χρήστη και συλλέγει από το δίκτυο ενδιαφέρουσες πληροφορίες για λογαριασμό του.

Έλεγχος (Monitoring)

Πολλές φορές, οι πληροφορίες διατίθενται διάσπαρτα, όχι στο εύρος του χώρου, αλλά στο μήκος του χρόνου. Συνεχώς νέες πληροφορίες δημοσιεύονται στο δίκτυο. Τα mobile agents μπορούν να προγραμματιστούν από τον χρήστη, ώστε να περιμένουν στον προορισμό τους μέχρι να διατεθούν οι σχετικές πληροφορίες που αναζητούν. Για παράδειγμα, ένα agent μπορεί να πάει σε κάποιον stock market host, να περιμένει μέχρι την καλύτερη προσφορά στην τιμή κάποιου προϊόντος και μετά, να το αγοράσει για τον χρήστη του. Ο χρήστης δεν χρειάζεται να είναι συνεχώς συνδεδεμένος στο δίκτυο. Το agent μπορεί να περιμένει έως ότου ο χρήστης επανασυνδεθεί για να του δώσει την ανάλογη απάντηση.

Διαπραγματεύσεις

Εκτός από το ψάξιμο σε βάσεις δεδομένων και αρχεία, τα mobile agents έχουν τη δυνατότητα να συλλέγουν πληροφορίες μέσω της αλληλεπίδρασής τους με άλλα agents. Με τον τρόπο αυτό, μαθαίνουν τους διαθέσιμους servers που πρέπει να επισκεφτούν για να εξυπηρετήσουν τις προτιμήσεις του χρήστη τους. Έτσι, αν κάποιος θέλει να σχεδιάσει μία συνάντηση με άλλα πρόσωπα, μπορεί να στείλει το mobile agent του, να αλληλεπιδράσει με τα agents αντιπροσώπων των υπολοίπων προσώπων που θέλει ο χρήστης να καλέσει. Το κάθε agent περιέχει πληροφορίες για το καθημερινό πρόγραμμα του χρήστη του. Τα agents διαπραγματεύονται και καθορίζουν τον αποδεκτό από όλους χρόνο συνάντησης.

Παράλληλη Επεξεργασία

Εφόσον τα mobile agents έχουν την ικανότητα να δημιουργούν κλώνους, μία δυνατή εφαρμογή τους, είναι η χρήση τους στη διαχείριση διεργασιών παράλληλης επεξεργασίας. Αν κάποιος υπολογισμός απαιτεί τόσο μεγάλο χρόνο επεξεργασίας ώστε να είναι αναγκαίος ο διαμερισμός της εργασίας σε πολλούς επεξεργαστές, μία υποδομή υπολογιστών που «φιλοξενούν» mobile agents μπορεί να υλοποιήσει την παράλληλη επεξεργασία.

Τηλεπικοινωνιακές υπηρεσίες

Η υποστήριξη και η διαχείριση τηλεπικοινωνιακών υπηρεσιών, χαρακτηρίζεται από το δυναμικό ανασχηματισμό του δικτύου, προς εξυπηρέτηση των χρηστών του. Το φυσικό μέγεθος αυτών των δικτύων και οι αυστηρές απαιτήσεις λειτουργίας τους, καθιστούν αναγκαία τη χρήση τεχνολογίας που δίνει τη δυνατότητα της ευελιξίας σε συνδυασμό με την απόδοση. Τα mobile agents, μπορούν να εφαρμοστούν με ποικίλους τρόπους στο συγκεκριμένο τομέα, όπως στη διαχείριση του δικτύου, στην υλοποίηση του location management και διαφόρων στρατηγικών διαχείρισης βάσεων δεδομένων των τηλεπικοινωνιακών συστημάτων.

Ηλεκτρονικό Εμπόριο

Αντιπροσωπευτικό παράδειγμα εφαρμογής των mobile agents, αποτελεί το ηλεκτρονικό εμπόριο. Ένα mobile agent μπορεί να κάνει παραγγελίες, ακόμη και αγορές για τον χρήστη του, όπως στην εξής περίπτωση: Ο χρήστης επιθυμεί να κάνει ένα ταξίδι και στέλνει ένα agent για αγοράσει το εισιτήριο. Το agent επισκέπτεται βάσεις δεδομένων με προγράμματα πτήσεων και τιμές για διάφορες αεροπορικές εταιρίες, βρίσκει την καλύτερη τιμή και ώρα, κάνει κράτηση και πληρώνει με το νόμισμα της πιστωτικής κάρτας του χρήστη.

Το ηλεκτρονικό εμπόριο μπορεί να εξελιχθεί και μεταξύ των agents. Έστω ότι υπάρχει ένας agent host που είναι αφοσιωμένος στις πωλήσεις και αγορές αυτοκινήτων. Ο χρήστης που θέλει να αγοράσει ένα αυτοκίνητο, δίνει στο mobile agent τις προτιμήσεις του, συμπεριλαμβάνοντας την τιμή και κάποια στρατηγική διαπραγμάτευσης, και στέλνει το agent στον συγκεκριμένο host. Εκεί, πιθανώς να συναντήσει άλλα agents, που πωλούν κάποιο αυτοκίνητο. Το mobile agent μπορεί να διαπραγματευτεί μαζί τους και να αντιδράσει γρήγορα σε μία ευκαιρία, ανταγωνιζόμενος κι άλλα agents αγοραστής. Τελικά, ανάλογα με το πρόγραμμά του, μπορεί είτε να κλείσει την συμφωνία είτε να επιστρέψει στον χρήστη μία αναφορά για να κάνει αυτοπροσώπως τον διακανονισμό.

Ψυχαγωγία

Τα mobile agents έχουν εφαρμογή στην ψυχαγωγία. Μπορούν να εκπροσωπούν τους παίκτες ενός παιχνιδιού και να συναγωνίζονται για λογαριασμό των χρηστών τους. Ο κάθε πραγματικός παίκτης προγραμματίζει το mobile agent του με μία στρατηγική και το στέλνει σε έναν game host. Αν ο game host εκτελείται σε κάποιον υπολογιστή του Las Vegas, το agent μπορεί να παίζει για να κερδίσει χρήματα.

Κάθε είδους εφαρμογή εκμεταλλεύεται διαφορετικά τη λειτουργικότητα των mobile agents, εν συγκρίσει με τις υπάρχουσες τεχνολογίες, και αναδεικνύει τις ξεχωριστές δυνατότητες τους. Οι εφαρμογές που αναφέρθηκαν παραπάνω, είναι γενικές, με την έννοια ότι μπορούν να ταιριάξουν ανάλογα, στις ειδικότερες περιπτώσεις που παρουσιάζονται, σε επιμέρους τομείς της επιστήμης, της εργασίας, της αγοράς και της εκπαίδευσης.

Η τεχνολογία των Mobile Agents

Η τεχνολογία των mobile agents περιέχει τρία κύρια συστατικά:

- **Μία γλώσσα προγραμματισμού**, που επιτρέπει στους προγραμματιστές εφαρμογών να προγραμματίζουν τα agents και τα μέρη που επισκέπτονται.
- **Ένα mobile agent σύστημα**, που παρέχει μία εικονική μηχανή (virtual machine) για την γλώσσα και διαχειρίζεται την εκτέλεση των agents.
- **Τα agent πρωτόκολλα**, τα οποία επιτρέπουν στα συστήματα διαφορετικών υπολογιστών να ανταλλάσσουν agents.

10.1 Η Γλώσσα Προγραμματισμού

Μία agent γλώσσα προγραμματισμού, επιτρέπει στους προγραμματιστές εφαρμογών αλληλεπίδρασης, να καθορίζουν τους αλγόριθμους που θα ακολουθούν τα agents και τις πληροφορίες που θα μεταφέρουν, καθώς ταξιδεύουν στο δίκτυο.

Συνήθως, τα τμήματα των εφαρμογών περιλαμβάνουν το στατικό λογισμικό στους υπολογιστές των χρηστών, για την αλληλεπίδραση των agents με τους χρήστες, και το στατικό λογισμικό στους servers, για την αλληλεπίδραση των συστημάτων που φιλοξενούν τα mobile agents με βάσεις δεδομένων. Τα τμήματα αυτά των εφαρμογών μπορούν να γραφτούν σε μια τυπική γλώσσα προγραμματισμού. Τα agents και οι διεπαφές των places που επισκέπτονται, γράφονται σε agent γλώσσα προγραμματισμού.

Μία agent γλώσσα προγραμματισμού διαθέτει χαρακτηριστικά που διευκολύνουν την επικοινωνία των εφαρμογών. Κατά κύριο λόγο, όμως, πρέπει να είναι:

Πλήρης. Οποιοσδήποτε αλγόριθμος να μπορεί να εκφραστεί στη γλώσσα αυτή. Ένα agent μπορεί να προγραμματιστεί να παίρνει αποφάσεις, να χειρίζεται περιπτώσεις εξαιρέσεων, να συγκεντρώνει, να αναλύει και να δημιουργεί νέες πληροφορίες.

Δυναμική. Να επιτρέπει στα mobile agents να μεταφέρουν αντικείμενα κλάσεων από μέρος σε μέρος. Ακόμη κι αν οι κλάσεις των αντικειμένων δεν είναι γνωστές στον προορισμό, τα αντικείμενα έχουν λειτουργικότητα, γιατί οι κλάσεις μεταφέρονται μαζί με το agent.

Συνεχής (Persistent). Όπου κι αν πηγαίνει, το agent και οι πληροφορίες που μεταφέρει, πρέπει να είναι ασφαλώς αποθηκευμένες σε μη πτητική μνήμη. Έτσι, το mobile agent εξακολουθεί να υφίσταται, ακόμη και σε περιπτώσεις προβλημάτων στη λειτουργία των υπολογιστών.

Επικεντρωμένη στην επικοινωνία. Να υπάρχουν συγκεκριμένες εντολές της γλώσσας που επιτρέπουν στο mobile agent να διεκπεραιώνει σύνθετες δικτυακές λειτουργίες, όπως είναι η μεταφορά, η πλοήγηση, ο έλεγχος πρόσβασης, η πιστοποίηση κλπ.

10.2 Mobile Agent Συστήματα

Ένα mobile agent σύστημα είναι ένα πρόγραμμα που υλοποιεί την agent γλώσσα προγραμματισμού, φιλοξενεί mobile agents και επιτρέπει την εκτέλεσή τους.

Τα πιο πρόσφατα mobile agent συστήματα έχουν δομηθεί στο ανώτερο επίπεδο Java συστημάτων και Tcl/tk συστημάτων. Ωστόσο, έχουν γίνει προσπάθειες να αναπτυχθούν συστήματα από τη βάση προς την κορυφή. Το Telescript είναι ένα από αυτά. Στον πίνακα 2 φαίνονται τα συστήματα, που αποτελούν τη βάση για την ανάπτυξη της mobile agent

τεχνολογίας. Ο πίνακας 3 περιέχει τα σημαντικότερα mobile agent συστήματα που έχουν υλοποιηθεί.

Γλώσσα προγ/μού	Οργανισμός
Java	Sun Microsystems
Tcl/tk	Sun Microsystems
Obliq	Digital

Πίνακας 2: Συστήματα που αποτελούν τη βάση για την ανάπτυξη της τεχνολογίας των mobile agents.

Σύστημα	Βασισμένο	Οργανισμός
Agent Tcl (ATC)	Tcl/tl	Dartmouth College
Telescript	custom	General Magic, Inc.
ARA	Tcl, C++, Java	University of Kaiserslautern
Mole	Java	University of Stuggart
TACOMA	Tcl	Cornell University and the University of Tromso
TKQML	Tcl, KQML	University of Maryland
MOA	Java, OS migration technology	Open Management Group
Concordia	Java	Mitsubishi Electric Information Technology Center America
Odyssey	Java	General Magic, Inc.
Voyager	Java	ObjectSpace
Aglets	Java	IBM Japan

Πίνακας 3: Τα σπουδαιότερα mobile agent συστήματα.

Η **Obliq** είναι μία λεξική, μη τυπική, μεταφρασμένη γλώσσα που υποστηρίζει κατανεμημένους αντικειμενοστραφείς υπολογισμούς. Αναμιγνύει τη χρήση πολλαπλών νημάτων ελέγχου μέσα σε έναν χώρο διεύθυνσης, πολλαπλούς χώρους διευθύνσεων σε μία μηχανή, ετερογενείς μηχανές σε ένα τοπικό δίκτυο και πολλαπλά δίκτυα μέσα στο Internet. Τα Obliq αντικείμενα έχουν κατάσταση και είναι τοπικά σε ένα site. Οι Obliq υπολογισμοί μπορούν να περιάγονται στο δίκτυο, όσο διατηρούνται οι δικτυακές συνδέσεις. Η Obliq είναι ένα από τα παλαιότερα (1995) mobile agent συστήματα. Σήμερα έχει αποδειχθεί ως χρήσιμη για την τυποποίηση των αλγορίθμων των mobile agents.

Το περιβάλλον της **Java** είναι μία νέα προσέγγιση στις κατανεμημένες υπολογιστικές εφαρμογές. Είναι μία απλή και φιλική αντικειμενοστραφής γλώσσα προγραμματισμού. Διαμορφώνει ένα μεταφέσιμο, αξιόπιστο και ικανοποιητικά ασφαλές σύστημα. Επιτρέπει την δυναμική εκτέλεση μεταφρασμένων προγραμμάτων και τον συγχρονισμό πολλαπλών νημάτων. Υποστηρίζει δικτυακές λειτουργίες και κατανεμημένα συστήματα μέσω των RMI (Remote object Invocation) και OS (Object Serialization). Δεν περιέχει mobile agents, αλλά καθιστά ικανή την ανάπτυξη της τεχνολογίας τους.

Το **AgentTcl** είναι ένα απλό, ανεξάρτητο πλατφόρμας, mobile agent σύστημα. Είναι βασισμένο στο Tcl/tk scripting σύστημα, γι' αυτό δεν διαθέτει την ευελιξία και την δύναμη μίας τυπικής γλώσσας προγραμματισμού, όπως είναι η Java. Κύριο συστατικό του αποτελεί ένας server που τρέχει σε κάθε μηχανή και επιτρέπει σε ολόκληρη την κατάσταση εκτέλεσης να μετακινείται. Τόσο το μοντέλο πλοήγησης όσο και το επικοινωνιακό μοντέλο, υλοποιούνται με απλές εντολές (agent_jump, agent_send, agent_receive, agent_meet) και επιτρέπουν την ασύγχρονη επικοινωνία.

Η **Telescript** είναι μία αντικειμενοστραφής mobile agent γλώσσα. Είναι μία πλατφόρμα που επιτρέπει τη δημιουργία ενεργών, κατανεμημένων εφαρμογών δικτύου. Κύρια στοιχεία της είναι οι agents, τα μέρη (places) και η εντολή "go". Οι agents ταξιδεύουν από μέρος σε μέρος χρησιμοποιώντας την εντολή "go". Η Telescript είναι ολόκληρη ένα σύστημα ανεξάρτητο πλατφόρμας, που συνίσταται σε μία γλώσσα κι έναν μεταφραστή, την Telescript engine. Τα μέρη είναι ισοδύναμα με το mobile agent environment. Το υπολογιστικό μοντέλο είναι η ίδια η Telescript γλώσσα και το agent μοντέλο είναι ένα Telescript αντικείμενο. Το agent διατηρεί την κατάσταση εκτέλεσης κατά τη διάρκεια της μετακίνησης, δίνοντας ένα ευέλικτο μοντέλο κύκλου ζωής.

Αν και, τεχνικά, είναι ένα καλά μελετημένο mobile agent σύστημα, έχει μεγάλο κόστος. Απαιτεί σημαντικό ποσό υπολογιστικών πόρων (96MB μνήμης) και επιπλέον, είναι ακριβό λογισμικό, παράγοντες που αναστέλλουν τη γενική αποδοχή χρήσης του ως mobile agent σύστημα.

Το **ARA (Agents for Remote Action)** είναι μία άλλη πλατφόρμα για mobile agents σε ετερογενή δίκτυα. Αντιλαμβάνεται τα mobile agents ως πρόγραμμα που έχει την ικανότητα να αλλάζει μηχανή host κατά την διάρκεια της εκτέλεσης, ενώ διατηρεί την εσωτερική του κατάσταση.

Σκοπό έχει να παρέχει λειτουργικότητα και ταυτόχρονα, να κυριαρχήσει σε όσο το δυνατόν περισσότερα προγραμματιστικά μοντέλα. Γι' αυτό, παρέχει έναν πυρήνα που υφίσταται πάνω από μη τροποποιημένο λειτουργικό σύστημα του host. Αυτός ο πυρήνας συνδέεται με έναν τροποποιημένο μεταφραστή της γλώσσας στον προορισμό, προκειμένου να παρέχει ευκολίες σε προγράμματα της γλώσσας αυτής. Με άλλα λόγια, υποστηρίζει ένα inter-agent σύστημα επικοινωνίας βασισμένο στον κοινό πυρήνα. Οι λειτουργίες κινητικότητας και ασφάλειας δεν αναμειγνύονται με την κανονική εκτέλεση του προγράμματος. Μεταφραστές γλωσσών προγραμματισμού που έχουν προσαρμοστεί στο ARA σύστημα είναι οι : C, C++, Java, Tcl. Ωστόσο, αυτός ο τύπος των mobile agents δεν είναι κατάλληλος για περιπτώσεις όπως οι κινητές επικοινωνίες.

Το **TACOMA (The Troms And COrnell Moving Agents project)** εστιάζει στην υποστήριξη των mobile agents από λειτουργικά συστήματα και στο πώς τα agents μπορούν να χρησιμοποιηθούν για τη λύση προβλημάτων που συνήθως προέρχονται από τα λειτουργικά συστήματα. Υπάρχουν τρεις εκδόσεις του: Η πρώτη βασίζεται στα UNIX και Tcl-TCP, και τα agents είναι γραμμένα σε γλώσσα Tcl/TK. Η δεύτερη βασίζεται στα UNIX και TCP, και τα agents είναι γραμμένα σε Tcl/TK και σε C. Η τρίτη έκδοση υποστηρίζει agents που είναι γραμμένα σε C, Tcl/TK, Perl, Python και Scheme. Το όλο σύστημα είναι γραμμένο σε C.

Το **MOA (Mobile Objects and Agents)** αποτελεί ένα σύστημα που υποστηρίζει την ανάπτυξη mobile agents γραμμένων σε Java . Έχει υλοποιηθεί στην κορυφή της Java Virtual Machine, χωρίς να επιφέρει τροποποιήσεις σε αυτή. Ο αρχικός σκοπός του ήταν να υποστηρίζει την επικοινωνία κατά την μετακίνηση των agents και να παρέχει έλεγχο των πόρων απέναντι σε επιθέσεις που προκαλούν denial of service. Αργότερα στους στόχους προστέθηκαν η συμμόρφωση με το μοντέλο των Java Beans, το οποίο προσθέτει ευελιξία και προσαρμοστικότητα στο agent σύστημα και στις εφαρμογές, και η διαλειτουργικότητα μεταξύ agent συστημάτων.

Το **Concordia** είναι ένα ολοκληρωμένο σύστημα ανάπτυξης και διαχείρισης mobile agent εφαρμογών δικτύου για κάθε συσκευή που υποστηρίζει Java. Τα mobile agents είναι γραμμένα σε Java, χρησιμοποιούν τις υπηρεσίες του Concordia περιβάλλοντος για να μετακινούνται σε κατανεμημένες μηχανές του δικτύου και να έχουν πρόσβαση στις υπηρεσίες τους. Διαχειριστές καθορίζουν ποιες υπηρεσίες είναι διαθέσιμες για ποιά agents και ποιους χρήστες. Τα agents μπορούν να χωρίζουν μία εργασία σε τμήματα, να την διεξάγουν παράλληλα στα καταλληλότερα μέρη και να συνεργάζονται για το συνολικό αποτέλεσμα. Το Concordia σύστημα μπορεί να ολοκληρωθεί με βάσεις δεδομένων και με CORBA αντικείμενα, παρέχοντας αξιόλογες εφαρμογές.

Το **Voyager** είναι μία πλατφόρμα ανάπτυξης Java agents που επιτρέπει σε Java προγραμματιστές να δημιουργούν εφαρμογές δικτύων, χρησιμοποιώντας παραδοσιακές, αλλά και προχωρημένες προγραμματιστικές τεχνικές για agents. Κύριο συστατικό του είναι το Virtual Object που επιτρέπει την διαχείριση οποιουδήποτε αντικείμενου ως κατανεμημένο αντικείμενο,

το οποίο μπορεί να μετακινείται όχι μόνο μεταξύ agent servers, αλλά και σε Java runtime περιβάλλοντα άλλων virtual objects.

Το **Aglet** σύστημα αντιπροσωπεύει το επόμενο βήμα εξέλιξης για εκτελέσιμο κώδικα στο Internet, εισάγοντας την ιδέα προγράμματος που μπορεί να μεταφέρεται μαζί με την πληροφορία κατάστασης. Τα Aglets είναι αντικείμενα της Java, τα οποία μπορούν, ενώ εκτελούνται σε έναν host, απότομα να σταματούν την εκτέλεση, να μεταβαίνουν σε άλλον απομακρυσμένο host και να συνεχίζουν την εκτέλεσή τους εκεί. Τα aglets αποτελούν mobile agents, γιατί υποστηρίζουν την αυτόνομη εκτέλεση, την δυναμική δρομολόγηση κατά την διαδρομή τους και παρέχουν πλήρεις μηχανισμούς μηνυμάτων και ιδιοτήτων. Ανοίγουν προοπτικές για μελλοντικές εφαρμογές, ειδικότερα μέσω της συνεργασίας τους με την OMG.

Η παρούσα διπλωματική εργασία υλοποιεί ένα mobile agent σύστημα για την παρουσίαση του concept του VHE του UMTS και την προσφορά value added services με τη χρήση των Aglets. Αναλυτικότερη περιγραφή του Aglet συστήματος γίνεται σε επόμενη ενότητα.

Τα συστήματα που παρουσιάστηκαν παραπάνω διαφέρουν σημαντικά ως προς την αρχιτεκτονική και την υλοποίηση. Για την επίτευξη της διαλειτουργικότητας κάποια κομμάτια της mobile agent τεχνολογίας θα έπρεπε να καθοριστούν ως πρότυπα. Πέντε εταιρίες (Crystaliz, General Magic Inc., GMD Fokus, IBM Corporation και Open Group) ανέπτυξαν συλλογικά την πρόταση για **Mobile Agent System Interoperability Facility (MASIF)** και την έφεραν προς συζήτηση στο Object Management Group (OMG).

Το **MASIF** καθορίζει τις διεπαφές μεταξύ των agent συστημάτων. Δεν αφορά στην διαλειτουργικότητα μεταξύ των γλωσσών των mobile agents. Περιορίζεται στη διαλειτουργικότητα μεταξύ agent συστημάτων που είναι γραμμένα στην ίδια γλώσσα, πιθανώς από διαφορετικούς προμηθευτές. Δεν προσπαθεί να καθορίσει τα πρότυπα για λειτουργίες των agents, αλλά, για τις διεπαφές στο επίπεδο των agent συστημάτων και συγκεκριμένα, για τις εξής τέσσερις περιοχές:

- Τη διαχείριση των agents διαφορετικών συστημάτων (Agent management).
- Τη μεταφορά των agents ελεύθερα μεταξύ των συστημάτων διαφορετικού τύπου (Agent transfer)
- Τα ονόματα των agents και των agent συστημάτων για την μεταξύ τους αναγνώριση. (Agent and agent system names)
- Τον τύπο των agent συστημάτων και τη σύνταξη της τοποθεσίας τους (Agent system type and location syntax.)

10.3 Mobile Agent Πρωτόκολλα

Τα mobile agent πρωτόκολλα επιτρέπουν σε δύο mobile agent συστήματα να επικοινωνούν. Μπορούν να λειτουργούν πάνω από διάφορα δίκτυα μεταφοράς, περιλαμβάνοντας αυτά που βασίζονται στα TCP/IP πρωτόκολλα του Internet, στο X.25 interface των οργανισμών τηλεφωνίας ή ακόμη στο ηλεκτρονικό ταχυδρομείο.

Γενικά, τα mobile agent πρωτόκολλα λειτουργούν σε δύο επίπεδα. Το χαμηλότερο επίπεδο αφορά στην μεταφορά των agents. Το υψηλότερο επίπεδο καθορίζει την κωδικοποίηση (encoding) και το serialization τους, και ακολούθως, την αποκωδικοποίηση (decoding) και το deserialization.

Οι agent κανόνες για serialization ορίζουν πώς ένα agent σύστημα κωδικοποιεί ένα agent, την διαδικασία του και την κατάστασή του, ως δυαδικά (binary) δεδομένα και πώς μπορεί να παραλείπει κομμάτια της κωδικοποίησης για την βελτίωση της απόδοσης. Αν και τα agent συστήματα μπορούν να διατηρούν agents σε διαφορετικά formats προς εκτέλεση, πρέπει να χρησιμοποιούν ένα πρότυπο format για τη μεταφορά. Το πρωτόκολλο μεταφοράς καθορίζει πώς δύο συστήματα, αρχικά ελέγχουν την αυθεντικότητα μεταξύ τους (πχ. χρησιμοποιώντας κρυπτογράφηση) και μετά, μεταφέρουν την κωδικοποίηση του agent το ένα σύστημα στο άλλο. Τα mobile agent συστήματα που έχουν υλοποιηθεί παρουσιάζουν διαφοροποιήσεις στα πρωτόκολλα που χρησιμοποιούν.

10.4 Projects βασισμένα στην Mobile Agent Τεχνολογία

Ο πίνακας 4 παραθέτει μία αντιπροσωπευτική λίστα από projects που χρησιμοποιούν την mobile agent τεχνολογία. Τα περισσότερα αφορούν στις επικοινωνίες, γεγονός που αντικατοπτρίζει την γενική αποδοχή εφαρμογής της στους συγκεκριμένους τομείς.

Σύστημα	Περιγραφή	Οργανισμός
InAMoS	Αφορά στο ηλεκτρονικό εμπόριο και στους κινητούς χρήστες. Τα mobile agents αντιπροσωπεύουν καταναλωτές σε market τοποθεσίες	TU Berlin
MagicCap	Χρησιμοποιεί την Telescript τεχνολογία για την υλοποίηση ενός έξυπνου προσωπικού συστήματος επικοινωνίας που επιτρέπει διάφορες μορφές επικοινωνίας (mail, fax, τηλέφωνο κλπ.) με τον χρήστη, ανεξάρτητα της γεωγραφικής του θέσης	General Magic
IBM-AMP	Δημιουργία περιβάλλοντος, έτσι ώστε, τα mobile agents να υλοποιούν ασφαλείς, απομακρυσμένες εφαρμογές σε μεγάλα δημόσια δίκτυα με πολλές κινητές συσκευές (laptops, PDAs). Η κεντρική ιδέα του στηρίζεται σε ένα Agent Meeting Point σε κάθε μηχανή, το οποίο παρέχει προϋποθέσεις και ευκολίες για την λειτουργία των agents.	IBM
Magna	Αρχιτεκτονική CORBA και Java που υποστηρίζει το Remote Programming και RPCs ως συμπληρωματικές μεθόδους, με σκοπό την εφαρμογή της στις υπηρεσίες τηλεπικοινωνιών παρέχοντας "Intelligence on Demand".	GMD Fokus
Stormcast	Παρέχει σημαντικές πληροφορίες για τον καιρό σε μετεωρολογικό γραφείο στη Norway, με χρήση mobile agents (Tacoma) που μεταβαίνουν σε servers τεράστιων μετεωρολογικών σταθμών από όπου συγκεντρώνουν πληροφορίες.	Cornell University and the University of Troms
Perpetuum Mobile Porocura Project	Αφορά στην εφαρμογή της mobile agent (Java) τεχνολογίας για τη διαχείριση δικτύων. Τα mobile agents του συστήματος καλούνται Neglets και μπορούν να συμβάλλουν στην ανακάλυψη και διόρθωση σφαλμάτων, στην επέκταση, τροποποίηση υπηρεσιών, στην ασφάλεια του δικτύου κλπ.	Carleton University
Messegers	Σύστημα (CORBA και Java) σχεδιασμένο για γενικού σκοπού κατανεμημένες εφαρμογές. Βασίζεται στους Messegers, αυτόνομα μηνύματα που μεταφέρουν την συμπεριφορά τους σε κάθε κόμβο που επισκέπτονται.	University of California
Plangent	Αποτελεί σύστημα από έξυπνα mobile agents που σχεδιάζουν τις ενέργειες που πρέπει να κάνουν για να ικανοποιούν τις απαιτήσεις του χρήστη και τροποποιούν το πλάνο τους σύμφωνα με τις τρέχουσες συνθήκες σε κάθε κόμβο. Συνδυασμός Cyberagent και Aglets τεχνολογίας	Toshiba Corp.

Πίνακας 4 : Projects εφαρμογής της mobile agent τεχνολογίας.

MOBILE AGENTS και Η ΓΛΩΣΣΑ Java

Η Java είναι μία object-oriented γλώσσα προγραμματισμού που ικανοποιεί τις απαιτήσεις ανάπτυξης εφαρμογών που επικοινωνούν μεταξύ τους. Έχει συμβάλλει δυναμικά στην εξέλιξη του Web και προσφέρει μοναδικές δυνατότητες που ευνοούν την ανάπτυξη της mobile agent τεχνολογίας .

11.1 Agent Χαρακτηριστικά της Java

Οι περισσότερες υλοποιήσεις mobile agent συστημάτων χρησιμοποιούν την Java ως γλώσσα προγραμματισμού. Οι λόγοι που την καθιστούν κυρίαρχη γλώσσα είναι τα agent χαρακτηριστικά της:

Ανεξάρτητη πλατφόρμας

Η Java είναι σχεδιασμένη να λειτουργεί σε ετερογενή περιβάλλοντα. Προκειμένου να είναι δυνατή η εκτέλεση εφαρμογών Java οπουδήποτε στο δίκτυο, ο compiler γεννάει byte κώδικα ουδέτερο αρχιτεκτονικής. Για να εκτελεστεί ο κώδικας σε έναν υπολογιστή πρέπει να είναι παρόν το Java runtime system. Οι τύποι δεδομένων και οι βιβλιοθήκες της δεν εξαρτώνται από τον επεξεργαστή ή το λειτουργικό σύστημα του υπολογιστή. Αυτό επιτρέπει την ανάπτυξη mobile agents ανεξάρτητα από τους τύπους των υπολογιστών στους οποίους θα τρέξουν.

Ασφαλής εκτέλεση

Με την Java, τα προγράμματα δεν μπορούν να έχουν πρόσβαση σε private δεδομένα αντικειμένων αν δεν τους επιτρέπεται η πρόσβαση, γεγονός που αποτρέπει μια σημαντική κατηγορία ιών. Το Java runtime system επιβεβαιώνει κάθε φορά ότι ο κώδικας του προγράμματος δεν είναι σε θέση να καταστρέψει τα βασικά στοιχεία της Java. Παρέχει επίσης, security manager για τον έλεγχο των μη ασφαλών λειτουργιών και τον καθορισμό των permissions για το δεδομένο πρόγραμμα. Γενικά, η αρχιτεκτονική ασφάλειας της Java καθιστά το σύστημα ικανό να φιλοξενήσει untrusted agents, εφόσον δεν μπορούν να «επέμβουν» στον host ή να έχουν πρόσβαση σε ιδιωτικές πληροφορίες.

Δυναμικό Class loading

Αυτός ο μηχανισμός επιτρέπει στη virtual machine να φορτώνει και να ορίζει κλάσεις στο χρόνο εκτέλεσης. Παρέχει έναν προστατευτικό χώρο ονομάτων για κάθε agent, έτσι ώστε να εκτελούνται με ασφάλεια, ανεξάρτητα μεταξύ τους. Ο μηχανισμός αυτός είναι επεκτάσιμος και επιτρέπει σε κλάσεις να φορτώνονται μέσω του δικτύου.

Προγραμματισμός πολλαπλών νημάτων

Τα agents εξ' ορισμού είναι αυτόνομα. Καθένα εκτελείται ανεξάρτητα από τους άλλα που βρίσκονται στον ίδιο χώρο ονομάτων. Η εκτέλεση καθενός agent με την δική του διεργασία καλείται νήμα εκτέλεσης και είναι αυτό που διαθέτει την αυτονομία. Η Java, όχι μόνο επιτρέπει τον προγραμματισμό πολλαπλών νημάτων, αλλά και τον συγχρονισμό τους, καθιστώντας δυνατή την αλληλεπίδραση των agents.

Object Serialization

Η Java παρέχει ένα serialization μηχανισμό, ο οποίος μπορεί να αναπαριστά την κατάσταση ενός αντικειμένου σε μία serialized μορφή, αρκετά λεπτομερειακή ώστε το αντικείμενο να μπορεί να επαναδομηθεί αργότερα. Η serialized μορφή του αντικειμένου πρέπει να είναι ικανή να αναγνωρίζει την Java κλάση από την οποία προήλθε η κατάσταση του αντικειμένου και να επαναφέρει την συγκεκριμένη κατάσταση σε μία νέα περίπτωση. Ένα αντικείμενο, συνήθως, αναφέρεται σε άλλα αντικείμενα. Για την διατήρηση της δομής του αντικειμένου, πρέπει κι αυτά να αποθηκεύονται και να ανακτώνται μαζί με το αντικείμενο. Ο μηχανισμός αυτός (serialization –deserialization) αποτελεί κλειδί για τη λειτουργία των mobile agents.

Reflection

Ο κώδικας της Java μπορεί να ανακαλύπτει πληροφορίες για τα πεδία, τις μεθόδους και τους constructors των φορτωμένων κλάσεων και να χρησιμοποιεί reflected πεδία, μεθόδους και constructors για τη λειτουργία των αντίστοιχων τμημάτων των αντικειμένων, στα πλαίσια των περιορισμών ασφάλειας. Έτσι καλύπτεται η απαίτηση των agents να συμπεριφέρονται έξυπνα όσον αφορά τους εαυτούς τους και άλλους agents.

11.2 Μειονεκτήματα της Java

Τα χαρακτηριστικά που προαναφέρθηκαν καθιστούν το σύστημα της γλώσσας Java το πλέον κατάλληλο για την δημιουργία mobile agents. Παρόλα αυτά, η συγκεκριμένη γλώσσα παρουσιάζει ορισμένα μειονεκτήματα:

Ελλιπής υποστήριξη για τον έλεγχο πόρων

Το σύστημα της γλώσσας Java δεν παρέχει κανέναν τρόπο για τον έλεγχο των πόρων που «καταναλώνει» ένα Java αντικείμενο. Για παράδειγμα, ένα agent μπορεί να ξεκινήσει ένα συνεχές loop καταναλώνοντας κύκλους επεξεργασίας και μνήμη. Η Java δεν παρέχει τρόπο ώστε ο host να περιορίζει τους πόρους που χρησιμοποιεί ένα αντικείμενο ή νήμα. Αυτό σχετίζεται άμεσα με τον τύπο επίθεσης που ορίζεται ως denial of service.

Ελλιπής προστασία των αναφορών

Οι public μέθοδοι ενός αντικειμένου Java είναι διαθέσιμες και σε άλλα αντικείμενα που περιέχουν αναφορές στις συγκεκριμένες μεθόδους. Επειδή δεν υπάρχει η έννοια της προστατευμένης αναφοράς, μερικά αντικείμενα έχουν πρόσβαση σε μεγαλύτερο σύνολο public μεθόδων από ότι άλλα. Δεν υπάρχει τρόπος ώστε ένα agent να ρυθμίζει άμεσα και να ελέγχει ποιοι άλλα agents χρησιμοποιούν τις μεθόδους του. Επίσης, δεν υπάρχει η έννοια της κατοχής των αναφορών σε ένα συγκεκριμένο αντικείμενο. Αυτό σημαίνει ότι αν κάποιο agent περιέχει μία αναφορά στο δικό μας agent, αναπόφευκτα ανοίγει ένα loop-hole για να κρατά alive το δικό μας agent, παρά τη θέλησή μας.

Δεν υποστηρίζεται η συνέχιση της Κατάστασης Εκτέλεσης.

Έως σήμερα, στη Java είναι αδύνατη η ανάκτηση ολόκληρης της execution state ενός αντικειμένου. Πληροφορίες όπως η κατάσταση του program counter και του frame stack είναι απαγορευμένες περιοχές για τα Java προγράμματα. Έτσι, για να συνεχίσει ένας mobile agent την εκτέλεσή του σε μία απομακρυσμένη μηχανή πρέπει να βασίζεται σε εσωτερικές τιμές ιδιοτήτων και εξωτερικά γεγονότα που να τον καθοδηγούν.

Τα τελευταία αρνητικά χαρακτηριστικά της Java λαμβάνονται σοβαρά υπόψη στην δημιουργία mobile agent συστημάτων, τα οποία, με τη σειρά τους, παρέχουν κατάλληλους συμπληρωματικούς μηχανισμούς, αφενός για την αντιμετώπιση των μειονεκτημάτων της γλώσσας προγραμματισμού και αφετέρου, για την ικανοποίηση των στόχων τους.

ΜΕΡΟΣ Δ'

AGLETS

12.1 Ορισμός

Η λέξη **Aglet** σημαίνει “**lightweight agent**”, με τον ίδιο τρόπο που το **applet** σημαίνει “**lightweight application**”. Ο όρος **aglet** προκύπτει από το συνδυασμό των λέξεων **agent** και **applet**. Το **aglet** αποτελεί το επόμενο σκαλοπάτι στην εξέλιξη του εκτελέσιμου κώδικα στο Internet, εισάγοντας την έννοια του κώδικα προγράμματος που μπορεί να μεταφέρεται μαζί με την πληροφορία κατάστασης.

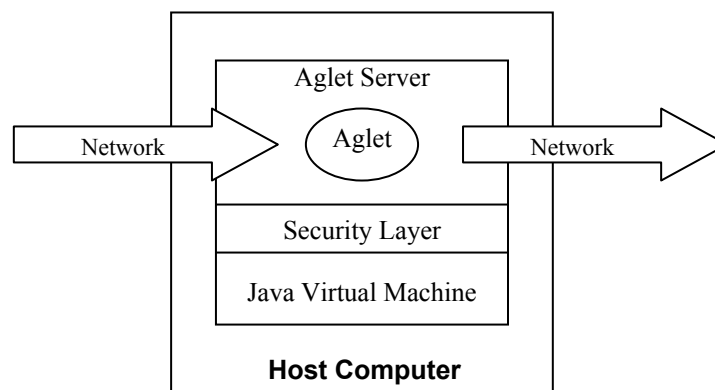
Τα **aglets** είναι **mobile Java agents** που μπορούν να μετακινούνται από μία μηχανή του Internet σε άλλη. Ένα **aglet** που εκτελείται σε μία μηχανή, μπορεί ξαφνικά να σταματήσει την εκτέλεση, να μεταβεί σε άλλη μηχανή και να συνεχίσει την εκτέλεσή του στη νέα τοποθεσία. Συμπεριφέρεται αυτόνομα και υποστηρίζει τη δυναμική δρομολόγηση κατά την πλοήγησή του στο δίκτυο.

12.2 Aglets και Applets

Τα **aglets** αποτελούν μία γενίκευση και επέκταση του μοντέλου κινητού κώδικα που έγινε γνωστό από τα **Java applets**. Ένα **applet** είναι κώδικας που μπορεί να μετακινείται διαμέσω του δικτύου από έναν **server** σε έναν **client**. Το **aglet** είναι ένα πρόγραμμα που, καθώς τρέχει, μπορεί να μεταναστεύει από τον ένα **host** στον άλλον, μεταφέροντας την κατάστασή του. Για το λόγο αυτό, μπορεί να ταξιδεύει ακολουθιακά σε πολλές τοποθεσίες του δικτύου, επιστρέφοντας τελικά στον αρχικό **host** της αφητηρίας του.

Τα **applets** τρέχουν στο περιβάλλον μίας **Java host εφαρμογής**: Ένας **Web browser** ενεργοποιεί την **Java εφαρμογή** για να φιλοξενήσει όποια **applets** χρειαστεί, κατά την πλοήγηση του χρήστη από σελίδα σε σελίδα. Η **Java εφαρμογή** εγκαθιστά έναν **security manager** για να επιβάλλει περιορισμούς στις δραστηριότητες των μη έμπιστων **applets**. Η εφαρμογή, για να «κατεβάσει» τα αρχεία κλάσεων ενός **applet**, δημιουργεί **class loaders** που γνωρίζουν το πώς να ζητήσουν τα αρχεία από έναν **HTTP server**.

Παρόμοια, ένα **aglet** προϋποθέτει ένα **aglet host περιβάλλον** να τρέχει στον εκάστοτε υπολογιστή, προτού μεταβεί σε αυτόν (Σχήμα 27). Κάθε **aglet host** εγκαθιστά έναν **security manager** που επιβάλλει περιορισμούς στις ενέργειες των μη έμπιστων **aglets**. Οι **hosts** «φορτώνουν» τα **aglets** μέσω **class loaders**, οι οποίοι γνωρίζουν πώς να ανακτήσουν τα αρχεία κλάσεων και την κατάσταση ενός **aglet** από έναν απομακρυσμένο **aglet host**.



Σχήμα 27: Ο **Aglet Server** φιλοξενεί τα **Aglets** στο κατάλληλο περιβάλλον.

12.3 Το Μοντέλο του Aglet

Το μοντέλο του aglet είναι σχεδιασμένο με τέτοιο τρόπο, ώστε να εκμεταλλεύεται τα agent χαρακτηριστικά της γλώσσας Java και ταυτόχρονα, να ξεπερνά τα μειονεκτήματά της. Σύμφωνα με το μοντέλο αυτό, το mobile agent είναι ένα κινητό αντικείμενο, το οποίο έχει αυτοέλεγχο, είναι καθοδηγούμενο από events και επικοινωνεί με τη χρήση μηνυμάτων.

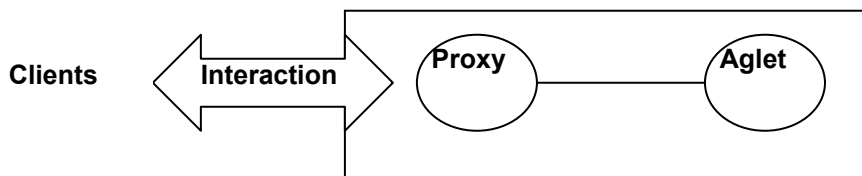
Το μοντέλο ορίζει ένα σύνολο αφαιρέσεων για την συμπεριφορά των mobile agents, από τις οποίες, θεμελιώδεις είναι οι εξής:

Aglet

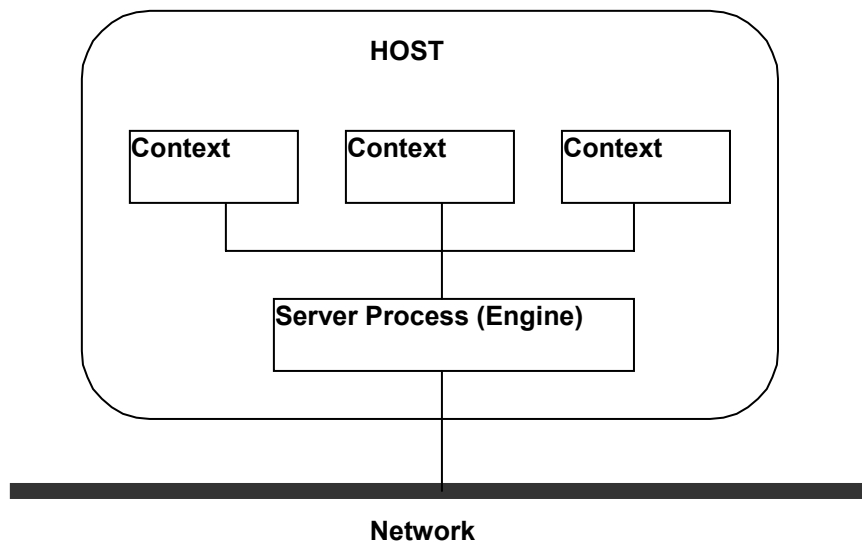
Ένα aglet είναι ένα κινητό αντικείμενο της Java που επισκέπτεται aglet hosts του δικτύου. Είναι αυτόνομο, γιατί εκτελείται με δικό του νήμα εκτέλεσης, μετά την άφιξή του στον host, και αντιδραστικό (reactive), γιατί μπορεί να ανταποκρίνεται στα εισερχόμενα μηνύματα.

Proxy

Ο proxy αποτελεί το διαμεσολαβητή ενός aglet. Εξυπηρετεί ως περίβλημα που προστατεύει το aglet από άμεση πρόσβαση στις public μεθόδους του και παρέχει διαφάνεια για την πραγματική θέση του aglet. Αυτό σημαίνει, ότι το aglet και οι proxies του είναι ξεχωριστές οντότητες, έτσι ώστε ο τοπικός proxy να κρύβει την απομάκρυνση του aglet. (Σχήμα 28)



Σχήμα 28: Σχέση μεταξύ Aglet και Proxy



Σχήμα 29: Σχέση μεταξύ των Host, Server Process και Contexts.

Context

Context είναι το περιβάλλον ζωής και εργασίας του aglet. Αντιπροσωπεύει την έννοια του όρου place που προαναφέρθηκε. Είναι ένα σταθερό αντικείμενο που παρέχει τρόπους διατήρησης και διαχείρισης των ενεργών aglets (running aglets) σε ένα ομοιόμορφο περιβάλλον εκτέλεσης, όπου το host σύστημα είναι προστατευμένο από «κακοπροαίρετα» aglets. Κάθε κόμβος μπορεί να τρέχει πολλαπλές server διεργασίες (engines) και κάθε server μπορεί να φιλοξενεί πολλαπλά περιβάλλοντα (contexts). Τα contexts έχουν συγκεκριμένο όνομα κι έτσι, η θέση τους μπορεί να προσδιοριστεί από το συνδυασμό της διεύθυνσης του server και του ονόματός τους. (Σχήμα 29)

Identifier

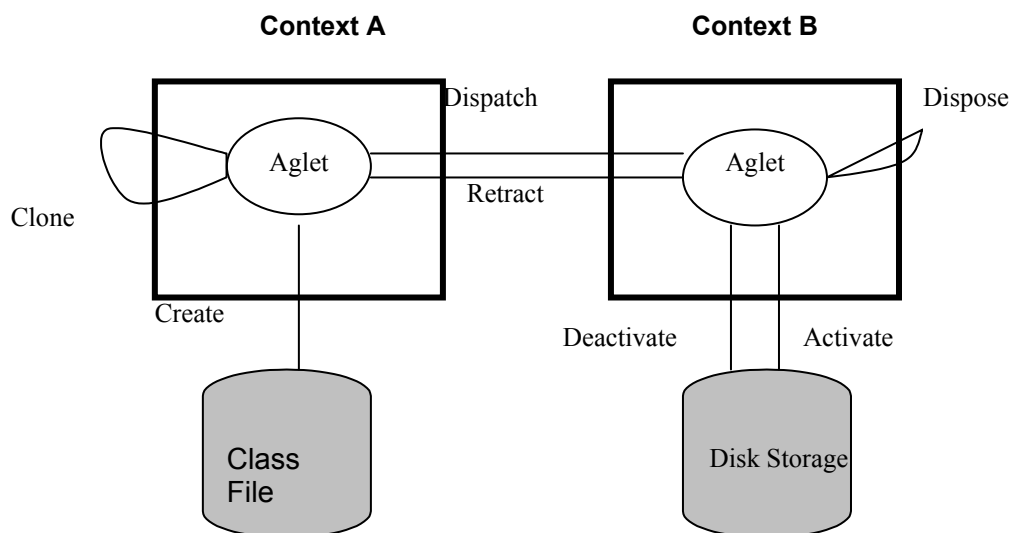
Ο identifier είναι ο κωδικός αριθμός ταυτότητας του κάθε aglet. Ο identifier είναι καθολικά μοναδικός και αμετάβλητος σε όλη τη διάρκεια ζωής του aglet.

12.4 Ο Κύκλος Ζωής του Aglet

Βάσει προσεκτικής ανάλυσης της ζωής και του θανάτου των mobile agents προέκυψε η συμπεριφορά που υποστηρίζει το μοντέλο του aglet αντικειμένου (Σχήμα 30).

Υπάρχουν δύο τρόποι για να έρθει στη ζωή ένα aglet. Ο ένας τρόπος είναι η δημιουργία ενός νέου instance του aglet αντικειμένου (creation). Ο άλλος, είναι η δημιουργία αντίγραφου ενός aglet που προϋπάρχει (cloning). Το κλωνοποιημένο aglet έχει την ίδια κατάσταση με το αρχικό, αλλά διαφορετικό AgletID αντικείμενο, που σημαίνει ότι έχει διαφορετική ταυτότητα.

Η κινητικότητα των aglets επιτυγχάνεται είτε ενεργητικά είτε παθητικά. Το aglet μπορεί να μετακινηθεί, από το context στο οποίο τρέχει σε άλλα, κατά τη διάρκεια της εκτέλεσής του. Τα contexts μπορεί να βρίσκονται είτε στην ίδια virtual machine είτε σε διαφορετικές. Στην ενεργητική περίπτωση, το aglet σπρώχνει τον εαυτό του από τον τρέχοντα host σε έναν απομακρυσμένο (dispatching). Στην παθητική περίπτωση, ένας απομακρυσμένος host μπορεί να τραβήξει το aglet μακριά από την τρέχουσα θέση του (retracting).



Σχήμα 30: Το μοντέλο του κύκλου ζωής του Aglet

Θεμελιώδεις λειτουργίες ενός aglet

Creation : Η δημιουργία λαμβάνει χώρα μέσα σε ένα context. Στο νέο aglet εκχωρείται ένα identifier. Το aglet εισάγεται στο context και αρχικοποιείται. Αρχίζει την εκτέλεσή του από την στιγμή που έχει επιτυχώς αρχικοποιηθεί.

Cloning : Η κλωνοποίηση του aglet παράγει ένα σχεδόν ίδιο αντίγραφο του αρχικού aglet μέσα στο ίδιο context. Οι μόνες διαφορές τους είναι ο identifier και το γεγονός ότι η εκτέλεση του προγράμματός του ξεκινά πάλι από την αρχή στο νέο aglet. Δεν κλωνοποιούνται τα νήματα εκτέλεσης.

Dispatching : Με τον τρόπο αυτό το aglet «σπρώχνεται» από το τρέχον context του σε ένα άλλο context προορισμού, όπου ξεκινά από την αρχή την εκτέλεσή του. Δεν μεταναστεύουν τα νήματα εκτέλεσης.

Retraction: Το retraction ενός aglet τραβάει το aglet από το τρέχον context και το εισάγει στο context από το οποίο προκλήθηκε.

Activation / Deactivation : Η απενεργοποίηση (deactivation) του aglet είναι η ικανότητα να σταματά την εκτέλεσή του προσωρινά και να αποθηκεύει την κατάστασή του σε ένα secondary storage. Η ενεργοποίησή του (activation) το επαναφέρει στο ίδιο context, για να συνεχίσει την εκτέλεση.

Disposal : Το disposal του aglet ισοδυναμεί με την καταστροφή του, δηλαδή, σταματά την εκτέλεση του aglet και το αφαιρεί από το περιβάλλον εκτέλεσης του.

Πίνακας 5: Οι θεμελιώδεις λειτουργίες ενός aglet.

Όταν εκτελούνται τα aglets καταναλώνουν πόρους. Όλα τα mobile aglet αντικείμενα μπορούν να είναι persistent, ώστε να μετατρέπονται σε ένα bit-stream. Αυτό το stream μπορεί να αποθηκευτεί, επιτρέποντας στα aglets να γίνονται ανενεργά προσωρινά, για συγκεκριμένα milliseconds (deactivation). Με τον τρόπο αυτό απελευθερώνουν τους πόρους, μέχρις ότου περάσει το ορισμένο χρονικό διάστημα ή έως ότου άλλο πρόγραμμα ζητήσει την ενεργοποίησή τους (activation). Το aglet επανέρχεται ενεργό, σε running mode, στο ίδιο context στο οποίο βρισκόταν τη στιγμή που απενεργοποιήθηκε (Πίνακας 5).

Σε αντίθεση με τα Java αντικείμενα, τα οποία αυτόματα απελευθερώνονται μέσω του garbage collector, το aglet αντικείμενο, μόνο του μπορεί να αποφασίζει αν θα πεθάνει ή όχι. Μπορεί, επίσης, ο χρήστης του να το καταστρέψει (dispose). Τότε καλείται συγκεκριμένη μέθοδος του aglet για την κατάλληλη ολοκλήρωση της ενεργής κατάστασής του και τον ομαλό τερματισμό της λειτουργίας του.

12.5 Το Event Μοντέλο του Aglet

Το προγραμματιστικό μοντέλο του aglet βασίζεται σε γεγονότα (events). Το event μοντέλο δίνει την ευκαιρία στους προγραμματιστές να πραγματοποιούν ενέργειες σε κάθε λειτουργία των aglets. Συγκεκριμένα, τους επιτρέπει να εισάγουν **listeners** στο aglet, οι οποίοι ανταποκρίνονται σε συγκεκριμένου είδους γεγονότα στη διάρκεια ζωής του aglet.

Υπάρχουν τρία είδη listeners:

- **Clone Listener**: Ακούει τα γεγονότα κλωνοποίησης (Clone Events). Ο listener μπορεί να προγραμματιστεί ώστε να υλοποιεί συγκεκριμένες ενέργειες, όταν το aglet πρόκειται να κλωνοποιηθεί, όταν δημιουργείται ο κλώνος ή/και μετά την κλωνοποίηση.

- **Mobility Listener:** Αναπαυκρίνεται σε γεγονότα κινητικότητας (Mobility Events). Μπορεί να χρησιμοποιηθεί όταν ένα aglet πρόκειται να γίνει dispatched σε ένα άλλο context, όταν πρόκειται να γίνει retracted από ένα context ή/και όταν φτάνει σε ένα νέο context.
- **Persistence Listener:** Ακούει τα γεγονότα συνέχισης (Persistency Events). Επιτρέπει στον προγραμματιστή, να προσθέτει ενέργειες που θέλει να γίνονται, όταν το aglet πρόκειται να απενεργοποιηθεί (deactivated) και αμέσως μετά την ενεργοποίησή του.

Γενικότερα, ένας listener είναι ένα αντικείμενο που υλοποιεί μία συγκεκριμένη διεπαφή. Η διεπαφή αυτή, ορίζει μία ή περισσότερες μεθόδους, που καλούνται ως ανταπόκριση σε κάθε τύπο γεγονότος που χειρίζεται. Συνδυάζει διαφορετικές μεθόδους για κάθε τύπο γεγονότος που αντιπροσωπεύει η event κλάση, ανάλογα, βέβαια, και με τη σημασιολογία τους (Πίνακας 6). Για παράδειγμα, ο CloneListener ορίζει τρεις μεθόδους –onCloning, onClone, onCloned- μία για κάθε τύπο γεγονότος που αντιπροσωπεύει η κλάση CloneEvent.

When	Event Object	Listener	Method called
About to be cloned	CloneEvent	CloneListener	OnCloning
Clone is created	CloneEvent	CloneListener	OnClone
After the clone was created	CloneEvent	CloneListener	OnCloned
About to be dispatched	MobilityEvent	MobilityListener	OnDispatching
About to be retracted	MobilityEvent	MobilityListener	OnReverting
After arrived at the destination	MobilityEvent	MobilityListener	OnArrival
About to be deactivated	PersistencyEvent	PersistencyListener	OnDeactivating
After activated	PersistencyEvent	PersistencyListener	OnActivation

Πίνακας 6: Οι Listeners, οι τύποι των γεγονότων στα οποία ανταποκρίνονται και οι μέθοδοι που χρησιμοποιούν.

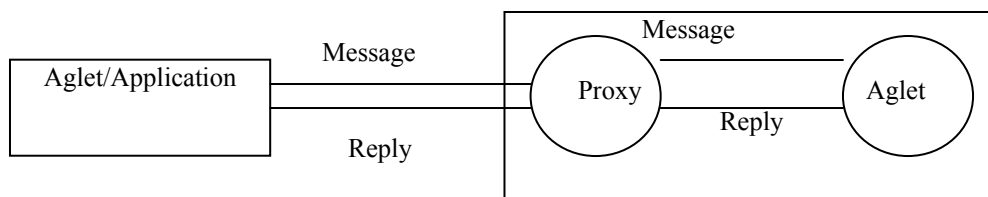
Το aglet event API προσπαθεί να πετύχει μία ισορροπία μεταξύ των διαφορετικών τύπων listener διεπαφών και των ξεχωριστών τύπων διεπαφής για κάθε είδος γεγονότος. Για το λόγο αυτό οι διεπαφές listener συνοδεύονται από ένα σύνολο *adapter* κλάσεων, τις

- **CloneAdapter**
- **MobilityAdapter**
- **PersistencyAdapter**

Οι κλάσεις αυτές επιτρέπουν στους προγραμματιστές να ξαναγράφουν και να προσαρμόζουν, μόνο εκείνες τις μεθόδους του listener που τους ενδιαφέρουν.

12.6 Το Επικοινωνιακό Μοντέλο του Aglet

Τα aglets επικοινωνούν μέσω μηνυμάτων. Το επικοινωνιακό μοντέλο, επιτρέπει στα aglets να δημιουργούν και να ανταλλάσσουν μηνύματα με ευέλικτους τρόπους. Εξ' ορισμού, το aglet messaging δεν εξασφαλίζει τον ταυτόχρονο χειρισμό μηνυμάτων. Αυτό σημαίνει ότι τα aglets χειρίζονται όλα τα μηνύματα ένα προς ένα (Σχήμα 31). Το Aglet API, όπως θα φανεί παρακάτω, παρέχει τις μεθόδους για την αντιμετώπιση της ταυτοχρονίας.



Σχήμα 31: Το μοντέλο επικοινωνίας μεταξύ των Aglets

Εκτός από τα **Aglet** και **AgletProxy** στο επικοινωνιακό μοντέλο εμφανίζονται άλλα τρία βασικά συστατικά:

➤ **Message.** Ένα μήνυμα είναι ένα αντικείμενο που ανταλλάσσεται μεταξύ των aglets. Επιτρέπει τόσο το σύγχρονο όσο και το ασύγχρονο πέρασμα μηνυμάτων, τα οποία χρησιμοποιούνται για την ανταλλαγή πληροφοριών και τη συνεργασία των aglets.

➤ **Future reply.** Μία μελλοντική απάντηση χρησιμοποιείται ως χειρισμός που επιτρέπει στον αποστολέα ενός μηνύματος να λαμβάνει την απάντηση ασύγχρονα.

➤ **Reply set.** Ένα σύνολο απαντήσεων μπορεί να περιέχει πολλά αντικείμενα μελλοντικών απαντήσεων και χρησιμοποιείται για να λαμβάνει αποτελέσματα καθώς αυτά γίνονται διαθέσιμα. Με το αντικείμενο Reply set, ο αποστολέας μπορεί να επιλέξει να λάβει την πρώτη απάντηση και να αγνοήσει τα υπόλοιπα αποτελέσματα.

Aglet API

13.1 To Aglet API

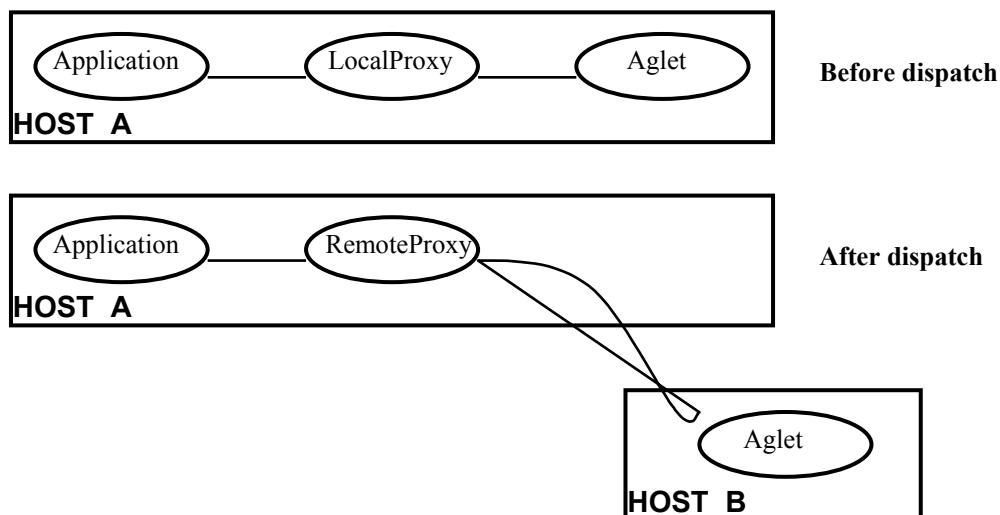
Το **Aglet API** είναι ένα πακέτο Java κλάσεων και διεπαφών, που καθορίζουν την κύρια λειτουργικότητα των aglet mobile agents. Το Aglet API είναι απλό και ευέλικτο. Παρακάτω παρουσιάζονται οι κυριότερες κλάσεις και διεπαφές του, καθώς και η συσχέτιση αυτών στον κύκλο ζωής του aglet.

com.ibm.aglet.Aglet

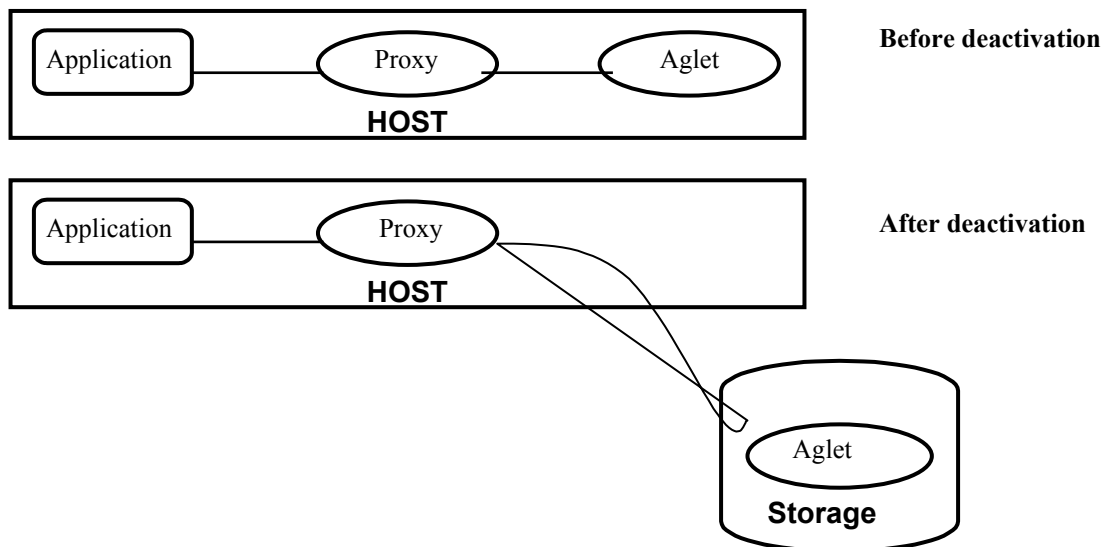
Η κλάση Aglet είναι μία αφηρημένη κλάση, που χρησιμοποιείται ως βάση για την ανάπτυξη όλων των aglets. Ορίζει μεθόδους για τον έλεγχο της κινητικότητας και του κύκλου ζωής ενός mobile agent, καθώς επίσης και μεθόδους για την προσαρμογή της συμπεριφοράς του agent στις απαιτήσεις των εφαρμογών. Οι μέθοδοι καλούνται από το σύστημα, όταν συμβαίνουν συγκεκριμένα γεγονότα στον κύκλο ζωής του aglet. πχ.

- *Aglet.dispatch (URL)* : προκαλεί την μετακίνηση ενός aglet από τον τοπικό host στον προορισμό που δίνεται ως παράμετρος.
- *Aglet.deactivate(long time)* : επιτρέπει σε ένα aglet να αποθηκευτεί σε secondary storage.
- *Aglet.clone()* : αναδημιουργεί ένα νέο instance του aglet που έχει την κατάσταση του αρχικού aglet.
- *Aglet.dispose()* : καταστρέφει και απομακρύνει το aglet από το τρέχον περιβάλλον του.

Η κλάση Aglet χρησιμοποιείται επίσης για την πρόσβαση στις ιδιότητες που σχετίζονται με ένα aglet. Το αντικείμενο `com.ibm.aglet.AgletInfo`, που είναι το αποτέλεσμα της μεθόδου `Aglet.getAgletInfo()`, περιέχει ενδογενείς ιδιότητες, όπως η ώρα και η βάση (codebase) δημιουργίας του aglet, και δυναμικές ιδιότητες, όπως είναι ο χρόνος άφιξης στον προορισμό του και η διεύθυνση του τρέχοντος περιβάλλοντος.



Σχήμα 32: Ο Proxy παρέχει διαφάνεια ως προς τη θέση του. Aglet Aglet Proxy. Dispatch (URLdestination)



Σχήμα 33: Απενεργοποίηση / Ενεργοποίηση ενός Aglet μέσω του Proxy του. `AgletProxy.deactivate(long duration)`.

com.ibm.aglet.AgletProxy

Η διεπαφή `AgletProxy` παρέχει έναν κοινό τρόπο πρόσβασης και χειρισμού του `aglet`. Η υλοποίησή της δίνεται από βιβλιοθήκη. Υπάρχουν διάφοροι τρόποι για να αποκτήσει κανείς την `AgletProxy` διεπαφή (interface):

- `AgletContext.getAgletProxies()` : δίνει έναν αριθμό από proxies που βρίσκονται μέσα στο context.
- `AgletContext.getAgletProxy(AgletID)` ή
- `Aglet.getAgletProxy(String contextName, AgletID)` : επιστρέφει το αντικείμενο `AgletProxy` για ένα συγκεκριμένο `aglet`.
- `Aglet.getProxy()` : επιστρέφει σε ένα `aglet` τον δικό του proxy.
- Μέθοδοι όπως οι `AgletContext.createAglet`, `AgletContext.retractAglet`, `AgletContext.clone`, επιστρέφουν το αντικείμενο `AgletProxy` του νέου `aglet`.
- Το αντικείμενο `AgletProxy` μπορεί να τεθεί σαν όρισμα μέσα σε ένα αντικείμενο `Message` και να σταλεί σε ένα `aglet` τοπικά ή απομακρυσμένα.

Η διεπαφή παρέχει πολλές μεθόδους, ώστε κάποιος— ο προγραμματιστής ή το ίδιο το `aglet` ή άλλο `aglet`-να μπορεί ρυθμίζει και να ελέγχει τη συμπεριφορά ενός `aglet`. Ανάλογα παραδείγματα φαίνονται στα σχήματα 32, 33.

Όταν καλείται, το αντικείμενο proxy, συμβουλευείται τον `SecurityManager` για να ελέγξει αν το συγκεκριμένο context επιτρέπεται να έχει πρόσβαση στη μέθοδο του (`AgletProxy.method`).

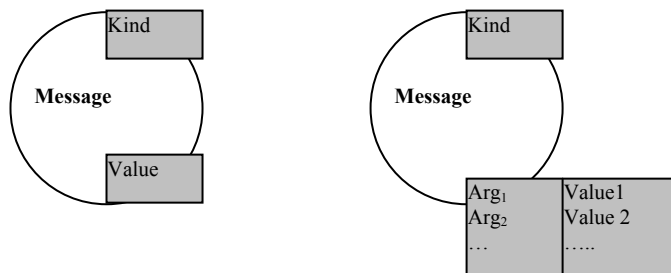
com.ibm.aglet.AgletContext

Το `AgletContext` αποτελεί μία διεπαφή στο περιβάλλον εκτέλεσης που περιέχει το `aglet`. Οποιοδήποτε `aglet` μπορεί να αποκτήσει αναφορά στο δικό του `AgletContext` αντικείμενο, μέσω της μεθόδου `Aglet.getAgletContext()`. Μπορεί, εν συνεχεία, να την χρησιμοποιήσει :

- για να λάβει τοπικές πληροφορίες, όπως τη διεύθυνση του hosting context (πχ. *AgletContext.getHostingURL()*), τα *AgletProxies* μέσα σε αυτό
 - για να θέσει ή να λάβει ιδιότητες του συγκεκριμένου context (πχ. *AgletContext.getProperty(key, value)*)
 - για να δημιουργήσει ένα καινούριο aglet μέσα στο context (πχ. *AgletContext.createAglet(URLcodebase, code, init)*)
 - για να στείλει ένα multicast μήνυμα στους συνδρομητές του ίδιου context (πχ. *AgletContext.multicastMessage(Message)*)
 - για να τραβήξει ένα aglet, που είχε προηγουμένως απομακρυνθεί με τη μέθοδο *AgletProxy.dispatch(URL)* σε μακρινό host, μέσα στο context (πχ. *AgletContext.retractAglet(remoteContextURL, agletID)*).
- Υπεύθυνη για την υλοποίηση της διεπαφής αυτής είναι η runtime βιβλιοθήκη.

com.ibm.aglet.Message

Τα aglets επικοινωνούν ανταλλάσσοντας αντικείμενα της κλάσης *Message*. Ένα αντικείμενο *Message* περιλαμβάνει ένα *String* αντικείμενο που καθορίζει το είδος (*kind*) του μηνύματος και προαιρετικά, άλλα αντικείμενα ως παραμέτρους. Αναλυτικά η δομή ενός αντικείμενου *Message* φαίνεται στο σχήμα 34.



Σχήμα 34: Τα αντικείμενα *Message*

Η αποστολή του μηνύματος μπορεί να γίνει με την κλήση μίας από τις ακόλουθες μεθόδους:

```
Object      AgletProxy.sendMessage(Message msg)
FutureReply AgletProxy.sendFutureMessage(Message msg)
void       AgletProxy.sendOnewayMessage(Message msg)
```

Το αντικείμενο *Message* περνά ως παράμετρος στη μέθοδο του aglet: *Aglet.handleMessage(Message msg)*. Η μέθοδος αυτή χειρίζεται τα εισερχόμενα μηνύματα και επιστρέφει *true* ή *false* στον αποστολέα, ανάλογα αν το aglet χειρίστηκε το μήνυμα ή όχι.

com.ibm.aglet.FutureReply

Το αντικείμενο *FutureReply* που ορίζεται από τη *FutureReply* διεπαφή, επιστρέφεται από την ασύγχρονη μέθοδο αποστολής μηνυμάτων και χρησιμοποιείται, αργότερα, για τη λήψη του αποτελέσματος ασύγχρονα. Με αυτό το *interface*, ο παραλήπτης μπορεί να καθορίσει αν η απάντηση είναι διαθέσιμη και να περιμένει για το αποτέλεσμα μέσα σε συγκεκριμένο χρονικό διάστημα, έτσι ώστε να συνεχίσει μετά την εκτέλεσή του, σε περίπτωση που η απάντηση δεν επιστραφεί.

Περαιτέρω ανάλυση του messaging γίνεται σε επόμενη ενότητα.

13.2 Η Υλοποίηση του Aglet

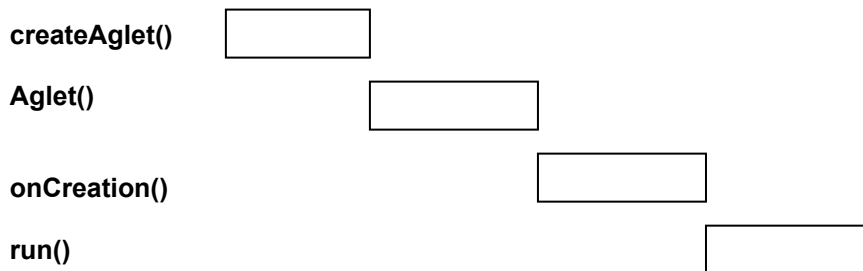
Κάθε aglet κλάση που ορίζεται από το χρήστη, επεκτείνει την αφηρημένη κλάση `com.ibm.aglet.Aglet`, αλλά δεν πρέπει να τροποποιεί τις προκαθορισμένες Aglet API μεθόδους. Μπορεί να υλοποιεί μία `java.io.Externalizable` διεπαφή. Μέθοδοι που μπορούν να αλλαχθούν από μία υποκλάση, επιτρέποντας την προσαρμογή της συμπεριφοράς του aglet, είναι οι εξής:

➤ `public void Aglet.onCreate(Object init)`

Αρχικοποιεί ένα νέο Aglet. Η μέθοδος καλείται μόνο μία φορά στον κύκλο ζωής ενός aglet.

➤ `public void Aglet.run()`

Αποτελεί το σημείο εισαγωγής στο νήμα αυτόνομης εκτέλεσης του aglet.



Σχήμα 35: Διάγραμμα των φάσεων δημιουργίας του Aglet

Αρχικά, στη φάση δημιουργίας του aglet (Σχήμα 35), καλείται η μέθοδος `createAglet` μέσα στο `context`. Η εκτέλεση αυτής –που συμβολίζεται με οριζόντια μπάρα στο σχήμα 35 θα δημιουργήσει ένα στιγμιότυπο (instance) της κλάσης `Aglet`, το οποίο προκαλεί τον `aglet constructor` (`Aglet()`). Όταν το aglet instance δομηθεί, καλείται η μέθοδος `onCreation()`, η οποία ολοκληρώνεται αμέσως πριν ξεκινήσει η εκτέλεση της μεθόδου `run()`.

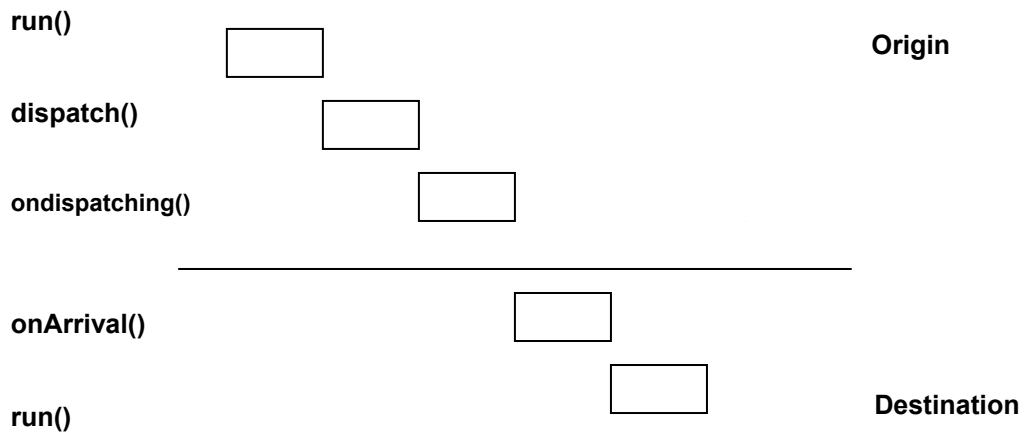
Όταν ένα aglet μεταναστεύει, κλωνοποιείται ή απενεργοποιείται, μετατρέπεται σε ένα bit-stream, και αργότερα, επανέρχεται πάλι στην αρχική του κατάσταση. Το aglet χρησιμοποιεί το `ObjectSerialization` της Java για να επιτυγχάνει τη μετατροπή αυτή (`Serialization / Deserialization`), περιλαμβάνοντας όλα τα αντικείμενα στα οποία έχει αναφορές. Όλα τα αντικείμενα στα οποία γίνεται άμεση ή έμμεση αναφορά από ένα aglet, πρέπει να υλοποιούν είτε το `java.io.Serializable` είτε το `java.io.Externalizable` interface, διαφορετικά η αναφορά τους πρέπει να δηλώνεται ως *transient*.

Όταν ένα τέτοιο αντικείμενο είναι ένα `proxy object`, τότε, από τη στιγμή που σειριακοποιείται, αντιγράφεται και δεν μπορεί πλέον να είναι διαμοιράσιμο από πολλαπλά aglets. Όταν ένα `proxy object` μεταναστεύει, μπορεί να διατηρεί την αναφορά στο πραγματικό aglet -κρατά το aglet id και τη διεύθυνσή του- ακόμη κι αν μεταφέρεται σε απομακρυσμένο host ή απενεργοποιείται. Αυτό συμβαίνει, εφόσον το aglet εμμένει στην ίδια τοποθεσία. Όμως, ο `AgletProxy` δεν μπορεί να κρατά τα ίχνη περιφερόμενων aglets. Από τη στιγμή που το aglet μεταβαίνει σε άλλο host (`dispatched`), το `proxy` αντικείμενο, που προηγουμένως αναφερόταν σε αυτό το aglet, δεν είναι πλέον έγκυρο (`valid`).

Η μέθοδος `Aglet.dispatch(URL destination)` υπαγορεύει σε ένα aglet μεταναστεύσει σε μία απομακρυσμένη μηχανή. Το event μοντέλο του aglet παρέχει ένα απλό σχήμα για τον έλεγχο της `dispatching` διεργασίας. Όταν το aglet γίνεται `dispatched`, καλούνται τα κατάλληλα `listener` αντικείμενα. Πρώτα αναβάλλεται το νήμα εκτέλεσης, μετά καλείται η `ObjectSerialization` διαδικασία. Το aglet με τα συσχετισμένα αντικείμενα μετατρέπονται στην κατάλληλη μορφή, μεταφέρονται στον προορισμό, το νήμα «σκοτώνεται» και οι σχετικοί πόροι απελευθερώνονται. Αν το aglet έχει δημιουργήσει περισσότερα από ένα νήματα, τα νήματα αυτά τερματίζονται, επίσης, αυτόματα. Για την αποφυγή προβλημάτων από μη αναμενόμενο τερματισμό, το aglet πρέπει να υλοποιεί ένα `listener interface` που να τερματίζει ομαλά τα νήματα στις μεθόδους του.

Καλώντας τη μέθοδο *dispatch()*, αμέσως ενεργοποιείται η μέθοδος *onDispatching()* του listener (Σχήμα 36). Με την ολοκλήρωση της τελευταίας μεθόδου, όλα τα νήματα τερματίζονται και το aglet μεταφέρεται στο απομακρυσμένο URL. Η πρώτη μέθοδος που ενεργοποιείται στον προορισμό είναι η μέθοδος *onArrival()* του listener, ακολουθούμενη από τη μέθοδο *run()*.

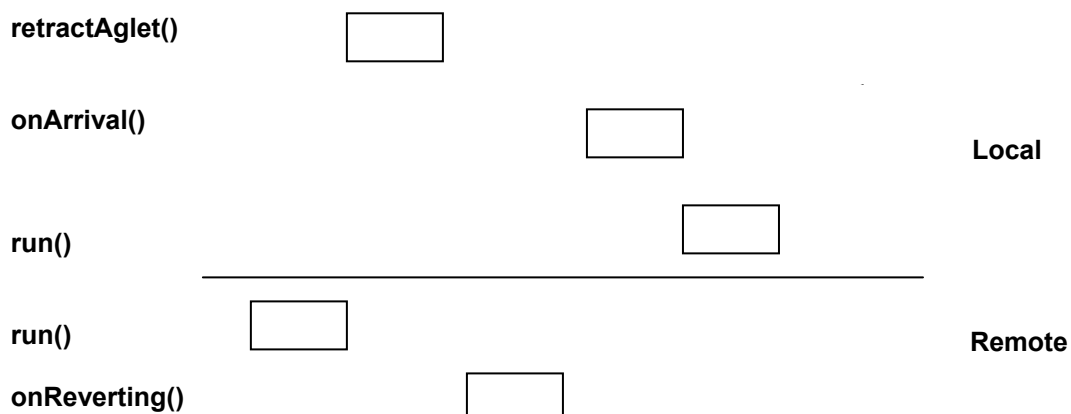
- *public void MobilityAdapter.onDispatching(MobilityEvent event)*
- *public void MobilityAdapter.onArrival(MobilityEvent event)*



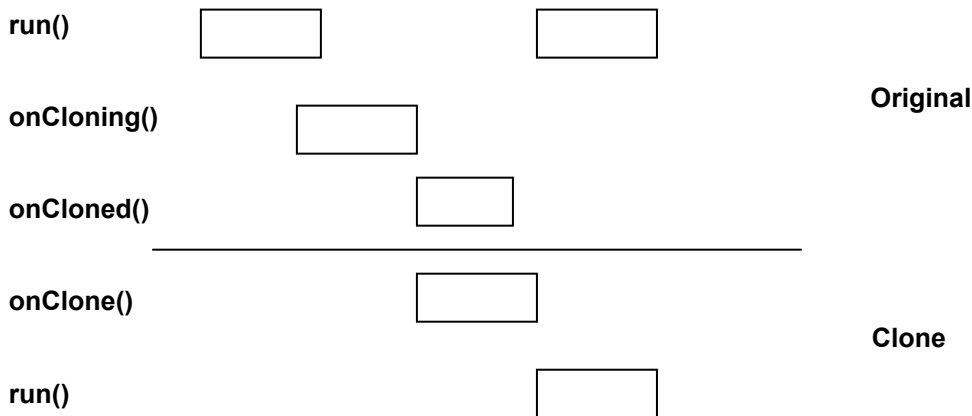
Σχήμα 36: Διάγραμμα των φάσεων της διαδικασίας dispatching

Πολλές φορές, ο χρήστης ενός aglet, ενδέχεται, να μην περιμένει το aglet να γυρίσει από μόνο του στον αρχικό host, αλλά να προτιμά να το τραβήξει πίσω. Αυτό γίνεται με τη χρήση της μεθόδου *AgletContext.retractAglet(URL contextAddress, AgletID id)*. Ο *MobilityListener* υποστηρίζει δύο μεθόδους για το χειρισμό της διαδικασίας retraction: τη μέθοδο *onArrival()* και τη μέθοδο *onReverting()*. Η προσπάθεια retracting ενός απομακρυσμένου aglet, ενεργοποιεί τη μέθοδο *onReverting()*, ως προειδοποίηση για την προετοιμασία του aglet (Σχήμα 37). Με την ολοκλήρωση της μεθόδου, όλα τα νήματα έχουν τερματιστεί. Η εκτέλεση της μεθόδου *onArrival()* τελειώνει πριν αρχίσει η μέθοδος *run()*.

- *public void MobilityAdapter.onReverting(MobilityEvent event)*



Σχήμα 37: Διάγραμμα των φάσεων της διαδικασίας retraction



Σχήμα 38: Διάγραμμα των φάσεων της διαδικασίας κλωνοποίησης

Η διαδικασία κλωνοποίησης ενός aglet (Σχήμα 38) δημιουργεί ένα αντίγραφο του αρχικού aglet μέσα στο τρέχον context του. Ο CloneListener ανταποκρίνεται στα γεγονότα κλωνοποίησης και παρέχει τρεις μεθόδους για την προσαρμογή και τον έλεγχο της όλης διαδικασίας. Στην αρχή καλείται η μέθοδος *Aglet.clone()*, η οποία ενεργοποιεί την μέθοδο *onCloning()* στον CloneAdapter του aglet που κλωνοποιείται. Μετά την ολοκλήρωση αυτής, η μέθοδος *clone* συνεχίζεται και δημιουργεί το δίδυμο aglet. Η πρώτη μέθοδος, που ενεργοποιείται στο νέο aglet, ξεχωριστά, είναι η *onClone()* μέθοδος στον δικό του CloneAdapter, ακολουθούμενη από την *run()* μέθοδο. Ταυτόχρονα, η μέθοδος *onCloned()* δείχνει ότι έχει πραγματοποιηθεί κλωνοποίηση στο αρχικό aglet.

- `public void CloneAdapter.onCloning(CloneEvent event)`
- `public void CloneAdapter.onClone(CloneEvent event)`
- `public void CloneAdapter.onCloned(CloneEvent event)`

Με παρόμοιο τρόπο, ο χειρισμός της λειτουργίας deactivation / activation γίνεται με τις αντίστοιχες μεθόδους του PersistenceAdapter, οι οποίες υλοποιούν τις ενέργειες που πρέπει να γίνουν ως ανταπόκριση στην αίτηση απενεργοποίησης ή ενεργοποίησης.

Εφόσον το aglet ολοκληρώσει τις εργασίες του, πρέπει να γίνει η σωστή διαδικασία καταστροφής του (*dispose*). Τα περισσότερα aglets διεκπεραιώνουν τις εργασίες τους ως επισκέπτες σε απομακρυσμένους servers και γι' αυτό, είναι υποχρεωμένα να ελαχιστοποιούν την δική τους κατανάλωση σε πόρους. Όταν ένα aglet καταστρέφεται, το περιβάλλον του απελευθερώνει όλους τους πόρους που του είχαν ανατεθεί. Η μέθοδος που παρέχει το AgletAPI για το σκοπό αυτό, είναι η *Aglet.dispose()*, με την οποία ένα aglet καταστρέφει τον εαυτό του. Τότε, αμέσως ενεργοποιείται η μέθοδος *onDisposing()*, με την οποία τερματίζονται όλα τα νήματα που είχαν δημιουργηθεί από το aglet -συμπεριλαμβανομένης και της αρχικής κλήσης *dispose()* - και το aglet εξαφανίζεται.

Η επικοινωνία των aglets

Τα aglets επικοινωνούν μέσω μηνυμάτων, με τον ίδιο τρόπο είτε τοπικά είτε απομακρυσμένα. Χρησιμοποιούν ένα σχήμα ανταλλαγής μηνυμάτων βασισμένο σε αντικείμενα, το οποίο είναι ανεξάρτητο θέσης, επεκτάσιμο, πλούσιο και υποστηρίζει τόσο τον σύγχρονο όσο και τον ασύγχρονο τρόπο επικοινωνίας. Τα αντικείμενα που περνούν ως παράμετροι των μηνυμάτων, μπορούν να είναι οποιοδήποτε τύπου Java που υποστηρίζει το ObjectSerialization.

Γενικά, για το χειρισμό των μηνυμάτων, τα aglets υλοποιούν **handlers** που λειτουργούν μόνο για τα είδη μηνυμάτων που δηλώνουν ότι καταλαβαίνουν. Η μέθοδος *Aglet.handleMessage(Message msg)* επιτρέπει στο aglet να ανταποκρίνεται στα μηνύματα που τού στέλνονται, επιστρέφοντας true ή false, ανάλογα με το αν επιτυγχάνεται ή όχι ο χειρισμός τους.

14.1 Τρόποι Ανταλλαγής Μηνυμάτων

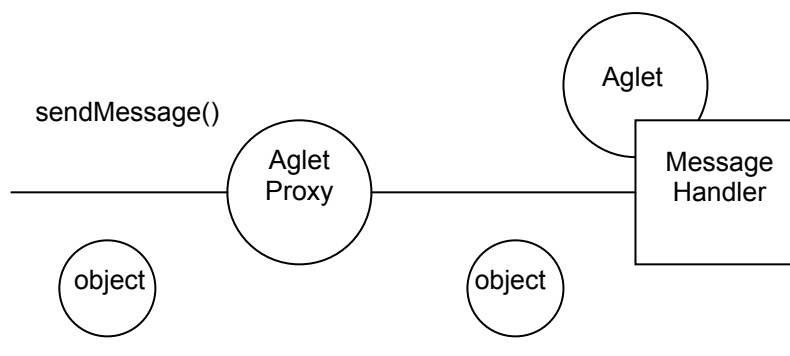
Σύγχρονος τρόπος

Στον σύγχρονο τρόπο ανταλλαγής μηνυμάτων, ο αποστολέας ενός μηνύματος σταματά προσωρινά την εκτέλεσή του έως ότου λάβει απάντηση.

Για την αποστολή μηνυμάτων χρησιμοποιείται η μέθοδος *AgletProxy.sendMessage(Message message)*, η οποία παίρνει το αντικείμενο Message ως όρισμα και στέλνει το μήνυμα στο aglet παραλήπτη, μέσω του AgletProxy του. Ο AgletProxy του παραλήπτη λειτουργεί ως πύλη προς την οντότητα του aglet και ταυτόχρονα, παρέχει διαφάνεια όσον αφορά στην τοποθεσία του.

Τα μηνύματα λαμβάνονται από τον message handler του παραλήπτη aglet. Πρώτη εργασία αυτού είναι να προσδιορίσει το είδος του μηνύματος, με μεθόδους όπως *Message.getKind()* ή *Message.sameKind(String)*. Μετά, μπορεί να εξαγάγει το/α όρισμα/τα αυτού, χρησιμοποιώντας τη μέθοδο *Message.getArg()* (Σχήμα 39).

Ο message handler χρησιμοποιεί το ίδιο το εισερχόμενο μήνυμα για να στείλει την απάντηση. Έτσι, η μέθοδος *Message.sendReply*, με ή χωρίς ορίσματα, χρησιμοποιείται ως απάντηση επιβεβαίωσης στο εισερχόμενο μήνυμα.



Σχήμα 39: Σύγχρονος τρόπος ανταλλαγής μηνυμάτων

Ασύγχρονος τρόπος

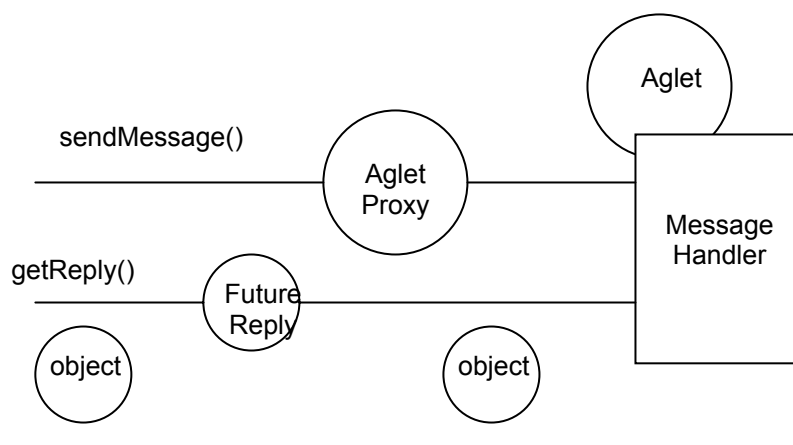
Σύμφωνα με αυτόν, ο αποστολέας ενός μηνύματος συνεχίζει την εκτέλεσή του, και αργότερα, λαμβάνει την απάντηση. Τα μελλοντικού τύπου μηνύματα δεν μπλοκάρουν την τρέχουσα εκτέλεση.

Το ασύγχρονο messaging, όπως προαναφέρθηκε στο Aglet API, βασίζεται στην ιδέα του **futureobject**. Ο αποστολέας χρησιμοποιεί την μέθοδο *AgletProxy.sendMessage(Message message)*, η οποία επιστρέφει ένα αντικείμενο τύπου *FutureReply* στον αποστολέα, ως υπόσχεση για μία μελλοντική απάντηση. Ο αποστολέας μπορεί να καθορίσει στο ενεργό νήμα να περιμένει ένα συγκεκριμένο χρονικό διάστημα για την απάντηση ή/και αργότερα, να ελέγξει αν η απάντηση έχει έρθει, ώστε να την λάβει (Σχήμα 40).

```
void FutureReply.waitForReply(long duration)
```

```
boolean FutureReply.isAvailable()
```

```
Object FutureReply.getReply()
```



Σχήμα 40: Ασύγχρονος τρόπος επικοινωνίας

Υπάρχει και η περίπτωση του **Oneway**-τύπου μηνύματος, το οποίο είναι ασύγχρονο, αλλά δεν επιστρέφει απάντηση:

```
AgletProxy.sendOnewayMessage(new Message ("question"))
```

14.2 Διαχείριση των Μηνυμάτων

Κάθε **aglet** φέρει ένα **MessageManager**, ο οποίος ελέγχει τον χειρισμό των εισερχόμενων μηνυμάτων. Τα μηνύματα εισέρχονται σε μία ουρά και ο **MessageManager** τα προωθεί, ένα κάθε φορά, στον **message handler** του **aglet**, με την ίδια σειρά με την οποία εισέρχονται. Επίσης, επιβεβαιώνει ότι το επόμενο μήνυμα δεν προωθείται πριν γίνει ο χειρισμός του τρέχοντος μηνύματος. Με άλλα λόγια, ο **message manager** σειριακοποιεί το χειρισμό μηνυμάτων.

Όμως, παρέχει κι ένα σύνολο μεθόδων που επιτρέπουν στο **aglet** να προσαρμόζει την συμπεριφορά του **message manager**, καθιστώντας το ικανό να θέτει προτεραιότητες σε μηνύματα, να επιτρέπει τον παράλληλο χειρισμό και τον συγχρονισμό μηνυμάτων.

Προτεραιότητες μηνυμάτων

Ένα **aglet** μπορεί να καθορίσει συγκεκριμένη προτεραιότητα, που συσχετίζεται με το είδος των μηνυμάτων. Καταρχήν, αποκτά πρόσβαση στον δικό του **message manager**, μέσω της μεθόδου *Aglet.getMessageManager()*. Κατόπιν, μπορεί να θέσει υψηλή προτεραιότητα για ένα συγκεκριμένο είδος μηνύματος, το οποίο τίθεται σε υψηλότερες θέσεις της ουράς

μηνυμάτων, μπροστά από τα μηνύματα μικρότερης προτεραιότητας: *MessageManager.setPriority (String kind, int priority)*

Παράλληλος χειρισμός

Με την κλήση της μεθόδου *MessageManager.exitMonitor()*, το aglet μπορεί να συνεχίσει το νήμα που χειρίζεται το τρέχον μήνυμα, ενώ ταυτόχρονα, να επιτρέψει το ξεκίνημα ενός νέου νήματος με ένα εισερχόμενο μήνυμα. Με αυτό τον τρόπο, δίνεται η δυνατότητα να υλοποιείται μία background εργασία, στη διάρκεια χειρισμού μηνυμάτων.

Συγχρονισμός μηνυμάτων

Πολλές φορές, το νήμα που χειρίζεται ένα μήνυμα, χρειάζεται να περιμένει, μέχρι η κατάσταση του aglet να ικανοποιήσει κάποιες συνθήκες ή να πληροφορήσει το επόμενο μήνυμα για τις τρέχουσες συνθήκες. Γι' αυτό, μπορεί να αναβάλλει την εκτέλεσή του, χωρίς να μπλοκάρει τα υπόλοιπα εισερχόμενα μηνύματα. Αυτό γίνεται με τη χρήση των μεθόδων:

Aglet.waitForMessage(), η οποία αναβάλλει την εκτέλεση του τρέχοντος νήματος μέχρις ότου ένα άλλο μήνυμα να του δώσει εντολή να συνεχίσει.

Aglet.notifyMessage(), η οποία ενεργοποιεί το νήμα που περιμένει για να συνεχίσει την εκτέλεσή του. Το νήμα που προκαλεί την ενεργοποίηση τίθεται στην αρχή της ουράς και περιμένει μέχρι το «παλιό» νήμα να ολοκληρώσει την εκτέλεσή του.

Aglet.notifyAllMessages(), η οποία ενεργοποιεί όλα τα νήματα που έχουν αναβληθεί ένα προς ένα. Αυτά τίθενται στην αρχή της ουράς και συνεχίζουν την εκτέλεσή τους ξεκινώντας από το παλαιότερο.

Multicasting

Προκειμένου να είναι εφικτός ο συντονισμός των δραστηριοτήτων πολλών aglets και η συνεργασία τους, δεν αρκεί μόνο η ένα-προς-έναν επικοινωνία. Για παράδειγμα, ένα aglet που εισέρχεται σε ένα νέο περιβάλλον δεν γνωρίζει τις οντότητες που υπάρχουν εκεί. Όμως πρέπει με κάποιο τρόπο να γνωστοποιήσει την άφιξή του στους άλλους προκειμένου να συνεργαστούν. Το aglet context επιτρέπει την ανταλλαγή μηνυμάτων ένα-προς-πολλά, μέσα στο ίδιο περιβάλλον. Δύο είναι οι βασικές ιδέες για το message multicasting :

- Τα aglets γίνονται «συνδρομητές», δηλώνουν συμμετοχή, για ένα ή και περισσότερα multicast μηνύματα, με τη μέθοδο *Aglet.subscribeMessage(String kind)* .
- Υλοποιούν handlers για τα είδη αυτών των μηνυμάτων.

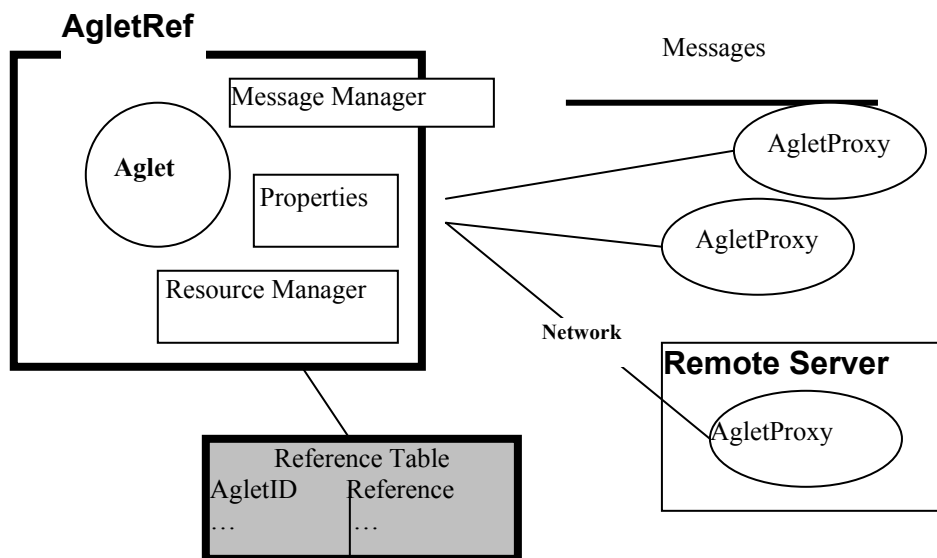
Η αποστολή ενός μηνύματος προς όλους τους συνδρομητές του ίδιου context, γίνεται ως εξής: *AgletContext.multicastMessage(Message message)* , η οποία επιστρέφει ένα αντικείμενο τύπου *ReplySet*. Η κλάση *ReplySet* επιτρέπει τη διαχείριση των *FutureReply* απαντήσεων από πολλαπλά aglets. Έτσι, ο αποστολέας του μηνύματος μπορεί να λάβει τις πιθανές απαντήσεις από πολλούς συνδρομητές.

Μία αξιοσημείωτη διαφορά μεταξύ της απλής και της multicasting ανταλλαγής μηνυμάτων είναι ότι η πρώτη είναι διαφανής ως προς τη θέση, σε αντίθεση με την δεύτερη που είναι τοπική μέσα σε ένα περιβάλλον εκτέλεσης.

Η δομή του Aglet και η μεταφορά του

15.1 Η Δομή του Aglet

Εστιάζοντας περισσότερο στη δομή που περιβάλλει ένα aglet, διαφαίνεται ένα σύνθετο αντικείμενο που αποτελείται από το aglet αντικείμενο και επιπροσθέτως, άλλα σημαντικά συστατικά.



Σχήμα 41: Η δομή του αντικειμένου Aglet

Όπως φαίνεται στο σχήμα 41, κεντρικό στοιχείο της δομής αποτελεί το αντικείμενο **AgletRef**, το οποίο θεωρείται ως μια εσωτερική αναπαράσταση του αντικειμένου aglet. Το AgletRef υλοποιεί τις βασικές λειτουργίες ενός aglet που ορίζονται από την αφηρημένη κλάση Aglet. Διατηρεί τις aglet ιδιότητες (*properties*), όπως την ταυτότητα και τη διεύθυνση του aglet. Είναι ένα αντικείμενο αναφοράς, το οποίο αποθηκεύεται στον πίνακα αναφοράς των aglets (*aglet reference table*). Ο τελευταίος κρατάει την αντιστοιχία κάθε AgletID με το πραγματικό aglet instance και χρησιμοποιείται για την εύρεση των aglets. Τα AgletProxy αντικείμενα παρέχουν έμμεση αναφορά στα aglets μέσω του αντικειμένου AgletRef.

Επίσης, ενσωματώνει συστατικά όπως τον **MessageManager** και τον **ResourceManager**:

- Ο **MessageManager**, όπως προαναφέρθηκε, είναι υπεύθυνος για τη διαχείριση των εισερχόμενων μηνυμάτων και γεγονότων.
- Ο **ResourceManager** είναι υπεύθυνος για τη διαχείριση των πόρων που προσπελούνται ή καταναλώνονται από τα aglets.

Προς το παρόν, στην υλοποίηση των aglets, ο ResourceManager υποστηρίζει την διαχείριση των νημάτων και των παραθύρων αλληλεπίδρασης που ανοίγονται ή κλείνονται από ένα aglet, όχι όμως και την διαχείριση των αρχείων συστήματος ή των sockets δικτύου. Όταν ένα aglet μεταβαίνει σε άλλον host, απενεργοποιείται ή καταστρέφεται, οι πόροι που δεσμεύει

πρέπει να απελευθερωθούν. Αφήνεται, λοιπόν, στον **garbage collector** να αποφασίσει γι' αυτό. Όμως, σε αντίθεση με τα συνήθη αντικείμενα της Java, τα aglets είναι ενεργά, με τα δικά τους νήματα ελέγχου και ποτέ δεν είναι αυτόματα garbage-collected. Έτσι, ο προγραμματιστής πρέπει να φροντίσει για την καταστροφή του aglet και την απελευθέρωση των πόρων.

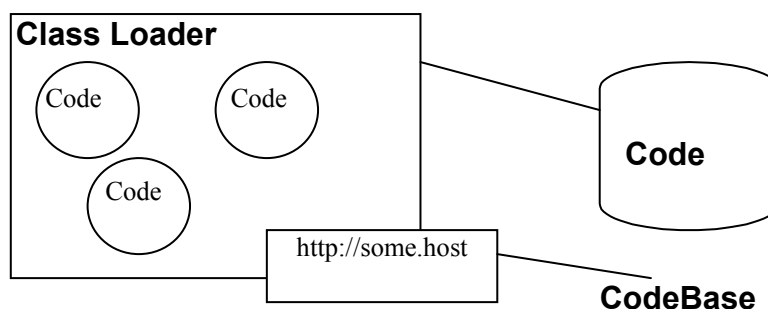
Στην περίπτωση αυτή, το αντικείμενο AgletRef σβήνεται από τον πίνακα αναφοράς. Επίσης, η εσωτερική αναφορά σε αυτό το aglet και στα συσχετιζόμενα συστατικά του, όπως τον MessageManager ή τις ιδιότητες, τίθεται null, οπότε ο garbage collector μπορεί να «καθαρίσει» τα αντικείμενα αυτά.

15.2 Η Μεταφορά των Κλάσεων

Στα mobile agent συστήματα, οι κλάσεις ενός agent πρέπει να είναι διαθέσιμες στον server στον οποίο δημιουργείται, εκτελείται ή μεταβαίνει. Έτσι, το mobile σύστημα, χρειάζεται είτε να φορτώνει τον byte κώδικα κατ' απαίτηση είτε να τον μεταφέρει μαζί με το agent. Το πλαίσιο των Aglets συνδυάζει και τα δύο σχήματα, ενώ παράλληλα, χρησιμοποιεί μία cache για να μειώσει το φόρτωμα των μη απαραίτητων κλάσεων.

Οι αναγκαίες κλάσεις για τη δημιουργία και την εκτέλεση ενός aglet, καθορίζονται δυναμικά στην διάρκεια της εκτέλεσης. Αυτό μπορεί να συμβεί και σε συγκεκριμένα σημεία της εκτέλεσης, όπως για παράδειγμα κατά την επίσκεψη του aglet σε άλλους hosts.

Η Java παρέχει μία ειδική κλάση που ονομάζεται **ClassLoader**, η οποία είναι ικανή να προσδιορίζει κάθε νέα κλάση στον byte κώδικα και να χειρίζεται όλες τις κλάσεις που χρησιμοποιούνται μέσα στην εκάστοτε κλάση. Τα aglets διαθέτουν την υποκλάση **AgletClassLoader**. Κάθε aglet συσχετίζεται με έναν συγκεκριμένο class loader, ο οποίος διαχειρίζεται όλες τις κλάσεις που απαιτούνται από το aglet (Σχήμα 42). Ένας class loader μπορεί να διαχειρίζεται πολλά aglet instances.



Code Base	Location
file://c:/some/path	υποκατάλογος του τοπικού συστήματος αρχείων
atp://some.host/some/host	υποκατάλογος ενός απομακρυσμένου aglet server
http://some.host/some/path	υποκατάλογος ενός Web server

Σχήμα 42: Ο κώδικας (Code) και η βάση του κώδικα (CodeBase)

Ο AgletClassLoader διατηρεί έναν πίνακα, *cache table*, για τις κλάσεις που αρχικά προσδιορίζονται από τον class loader. Εκεί αποθηκεύονται όλες οι κλάσεις που απαιτούν να φορτώσουν τα aglets από την βάση του. Στην περίπτωση αυτή, ο AgletClassLoader, αρχικά, ψάχνει για μία κλάση μέσα σε αυτόν τον πίνακα. Αν δεν την βρει εκεί, ζητά από τον class loader του συστήματος να την φορτώσει από το CLASSPATH. Αν κι αυτό αποτύχει, ψάχνει για τον cached bytecode που κρατά ο CacheManager. Διαφορετικά, ορίζει μία νέα κλάση από την βάση του κώδικα (CodeBase), δηλαδή από τον (τοπικό ή απομακρυσμένο) υποκατάλογο που περιέχει τον compiled κώδικα της κλάσης. Αυτή αποθηκεύεται, επίσης, στο cache table για

ενδεχόμενη χρήση του κι από άλλα aglets, συμβάλλοντας με τον τρόπο αυτό στη μείωση της κίνησης του δικτύου και στην βελτίωση της απόδοσης.

Στο πλαίσιο των aglets, οι Java κλάσεις κατηγοριοποιούνται ανάλογα με την προέλευσή τους και η διαχείρισή τους γίνεται με διαφορετικούς τρόπους. Υπάρχουν, λοιπόν, οι εξής κατηγορίες κλάσεων:

- Archived classes, κλάσεις συγκεντρωμένες μέσα σε ένα JAR αρχείο.
- Code Base classes, κλάσεις που φορτώνονται από τη βάση του aglet κώδικα.
- System classes, κλάσεις που φορτώνονται από το CLASSPATH.
- Άλλες κλάσεις που φορτώνονται από βάσεις άλλων aglets, όταν για παράδειγμα, ένα μήνυμα που δέχεται το aglet, περιέχει αντικείμενα κλάσεων του αποστολέα.

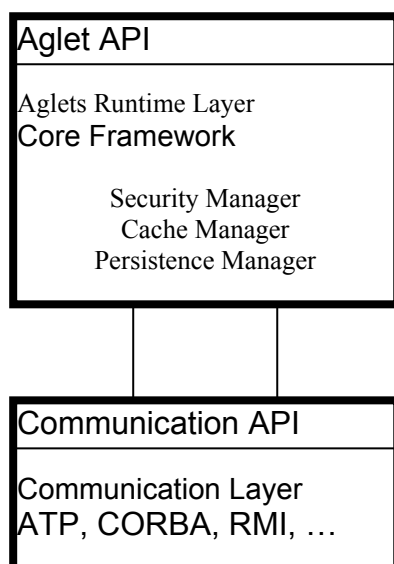
Όταν ένα aglet μεταναστεύει ή απενεργοποιείται, ο byte κώδικας των εν χρήση codebase κλάσεων, αποθηκεύεται σε ένα stream, κατά τη διαδικασία του ObjectSerialization, και μετά, στέλνεται στον προορισμό ή στο secondary storage. Αντιθέτως, οι system κλάσεις ποτέ δεν αποθηκεύονται σε stream και δεν μεταφέρονται. Η Aglet δομή, προϋποθέτει ότι όλες οι system κλάσεις που χρησιμοποιούνται από ένα aglet είναι διαθέσιμες στους servers, στους οποίους μεταβαίνει. Αν δεν είναι διαθέσιμες, η διαδικασία του dispatching αποτυγχάνει.

Από την άλλη πλευρά, δεν επιτρέπεται σε ένα aglet να φορτώνει κλάσεις από δύο διαφορετικές βάσεις κώδικα (codebases), για λόγους ασφαλείας. Οπότε, όταν ένα aglet φτάνει στον host προορισμού ή ενεργοποιείται, γίνεται ανάκτηση του byte κώδικα από το stream, ο οποίος χρησιμοποιείται για ορίσει τις κλάσεις που χρειάζονται για την επαναδόμηση του aglet. Αν αυτές υπάρχουν ήδη στην τοπική cache, χρησιμοποιούνται οι παλιές αντί των νέων. Παρόμοια συμβαίνει και στην περίπτωση κλωνοποίησης ενός aglet, όπου ο κλώνος, χρησιμοποιεί τον ίδιο class loader με του αρχικού aglet, γι' αυτό και συνίσταται από το ίδιο ακριβώς σύνολο κλάσεων.

Κατά την μεταφορά του byte κώδικα, εισερχόμενος ή εξερχόμενος από το τρέχον περιβάλλον, ενεργοποιείται ο SecurityManager για να ελέγξει αν επιτρέπεται το aglet να μεταφέρει τον κώδικα. Ομοίως συμβαίνει, όταν το aglet μεταφέρει αρχεία από άλλους servers.

Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ AGLET ΣΥΣΤΗΜΑΤΟΣ

Η αρχιτεκτονική του συστήματος των aglets συνίσταται σε δύο επίπεδα και δύο APIs που καθορίζουν διεπαφές για την πρόσβαση στις λειτουργίες τους (Σχήμα 43).



Σχήμα 43: Η αρχιτεκτονική του Aglet συστήματος

Το runtime layer αποτελεί την υλοποίηση του Aglet API και ορίζει την συμπεριφορά των API συστατικών, όπως είναι τα AgletProxy και AgletContext. Το επίπεδο αυτό είναι απαραίτητο για την κατάλληλη διαχείριση και εκτέλεση των aglets.

Το communication layer είναι κυρίως υπεύθυνο για την μεταφορά και τη λήψη ενός serialized agent σε κάποιον προορισμό. Υποστηρίζει, επίσης, την επικοινωνία μεταξύ των agents και ευκολίες στη διαχείρισή τους.

16.1 Aglets Runtime Layer

Το επίπεδο αυτό αποτελείται από δύο μέρη:

- έναν πυρήνα (Core Framework) και
- ένα σύνολο υποσυστατικών διαχείρισης του συστήματος.

Το πλαίσιο του πυρήνα περιλαμβάνει θεμελιώδεις μηχανισμούς για την εκτέλεση των aglets, όπως:

- Αρχικοποίηση και serialization/deserialization των aglets.
- Φόρτωμα των κλάσεων και μεταφορά.
- Διαχείριση των αναφορών και garbage collection.

Οι υπηρεσίες αυτές ενδέχεται να διαφέρουν ανάλογα με τις απαιτήσεις και τα περιβάλλοντα των εφαρμογών. Για το λόγο αυτό, έχουν σχεδιαστεί τα υποσυστατικά, τα οποία είναι επεκτάσιμα και ευπροσάρμοστα παρέχοντας διευκολύνσεις στους προγραμματιστές.

Ο **PersistenceManager** είναι υπεύθυνος για την αποθήκευση και την επανάκτηση απενεργοποιημένων (serialized) aglets. Αποθηκεύει τον byte κώδικα και την κατάσταση του agent σε ένα secondary storage, όπως είναι μία βάση δεδομένων ή ένα σύστημα αρχείων.

Ο **CacheManager** είναι υπεύθυνος για τη διατήρηση του byte κώδικα που χρησιμοποιείται από το aglet. Επειδή ο aglet κώδικας, χρειάζεται να μεταφέρεται, όταν το aglet μετακινείται μεταξύ των servers, ο μηχανισμός cache συμβάλλει στη βελτίωση της απόδοσης του agent.

Ο **SecurityManager** είναι υπεύθυνος για την προστασία των hosts και των aglets από κακόβουλες οντότητες. Για κάθε ευαίσθητη λειτουργία απαιτείται συμβούλευση από τον SecurityManager, για να ελέγξει αν επιτρέπεται στο aglet να διεκπεραιώσει την απαιτούμενη λειτουργία. Το συστατικό αυτό βασίζεται στην κλάση SecurityManager του συστήματος της γλώσσας Java.

Τα παραπάνω συστατικά διαχείρισης ορίζονται είτε ως διεπαφές είτε ως αφηρημένες κλάσεις, οπότε οι δημιουργοί aglet servers μπορούν να τα υλοποιούν ή να τα επεκτείνουν, για να τα ταιριάζουν στο περιβάλλον της εκάστοτε εφαρμογής. Προς το παρόν δεν είναι δημοσίως διαθέσιμα στους προγραμματιστές aglet εφαρμογών.

16.2 Communication Layer

Το runtime layer δεν διαθέτει εσωτερικό μηχανισμό για την μεταφορά των aglets στο δίκτυο. Χρησιμοποιεί το **communication API** που παρέχει γενικευμένο τρόπο επικοινωνίας μεταξύ των agent συστημάτων. Αυτό το API και η υλοποίησή του συνιστούν έναν agent μηχανισμό μεταφοράς και επικοινωνίας, ανεξάρτητο από τα Aglets. Ο διαχωρισμός της runtime υλοποίησης των aglets και του μηχανισμού μεταφοράς, σημαίνει ότι τα aglets δεν βασίζονται σε μοναδικό είδος μηχανισμού μεταφοράς. Η σημερινή υλοποίηση του communication layer των Aglets πραγματοποιείται βάσει του **Agent Transfer Protocol (ATP)**.

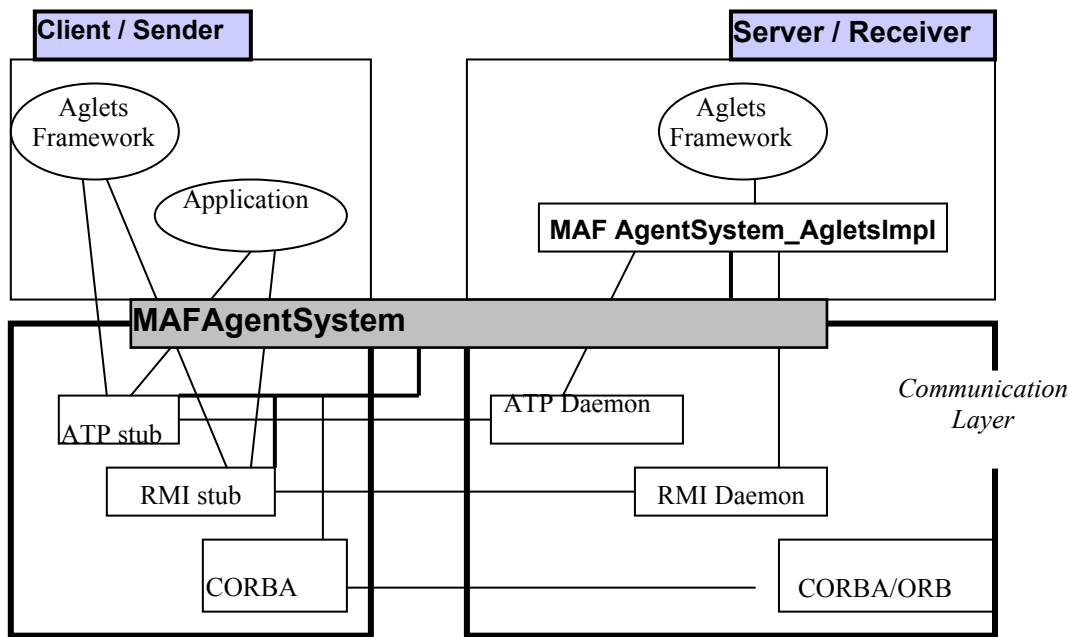
16.2.1 Communication API

Το communication API που χρησιμοποιείται από το aglet runtime περιβάλλον, προέρχεται από το πρότυπο OMG MASIF (Mobile Agent System Interoperability Facility), το οποίο επιτρέπει τη διαλειτουργικότητα μεταξύ διαφόρων agent συστημάτων. Αυτό το API, παρέχει διεπαφές και καθορίζει έναν κοινό τρόπο αναπαράστασης των δεδομένων σε Java, τα οποία συμμορφώνονται με την IDL, που ορίζεται στο MASIF πρότυπο.

Αν και οι MASIF διεπαφές αφορούν σε CORBA αντικείμενα, οι διεπαφές των Aglets δεν βασίζονται στο CORBA. Ορίζονται ως κανονικές Java κλάσεις, διεπαφές ή αφηρημένες κλάσεις, που λειτουργούν ως κοινό περίβλημα για τα πρωτόκολλα που χρησιμοποιούνται. Έτσι, είναι δυνατό να χρησιμοποιούνται πολλά είδη πρωτοκόλλων, διαφορετικά από το CORBA/IIOP, όπως είναι το ATP και το RMI.

16.2.2 Η Αρχιτεκτονική του Communication Layer

Η αρχιτεκτονική του communication layer των Aglets φαίνεται στο σχήμα 44. Η κλάση *com.ibm.maf.MAAgentSystem* είναι μία αφηρημένη κλάση που ορίζει ένα σύνολο μεθόδων ισοδύναμο με της MASIF διεπαφής. Δύο ήδη κλάσεων επεκτείνουν αυτή την αφηρημένη κλάση. Η μία, είναι μία κλάση υλοποίησης του agent συστήματος και η άλλη, είναι ένα stub αντικείμενο που μεταφέρει αιτήσεις στον προορισμό. Ένα agent σύστημα πρέπει να έχει ένα stub αντικείμενο για κάθε πρωτόκολλο που υποστηρίζει. Οι εφαρμογές ή οι clients, χρησιμοποιούν ένα συγκεκριμένο stub αντικείμενο για δεδομένο πρωτόκολλο, προκειμένου να στείλουν μία αίτηση στον προορισμό.



Σχήμα 44 : Η αρχιτεκτονική του Communication Layer

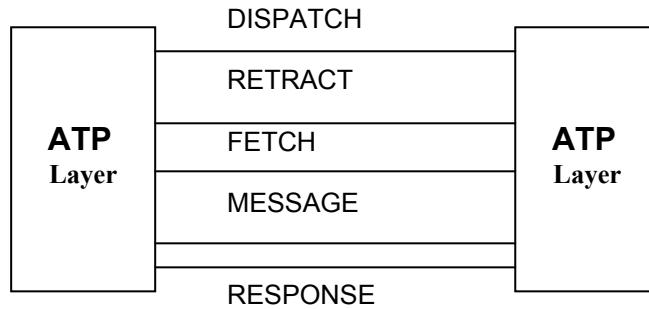
Από την άλλη πλευρά, ένας agent server εμπεριέχει μία υλοποίηση του MAFAgentSystem που χειρίζεται τις αιτήσεις. Τα Aglets έχουν την κλάση υλοποίησης `com.ibm.aglets.MAFAgentSystem_AgletsImpl`. Ένας server, γενικότερα, έχει έναν ή περισσότερους δαίμονες που δέχονται αιτήσεις και μπορεί να υποστηρίζει πολλαπλά πρωτόκολλα, έχοντας διαφορετικούς δαίμονες που χειρίζονται το κάθε πρωτόκολλο. Όταν ο δαίμονας δέχεται αιτήσεις, τις προωθεί στο `MAFAgentSystem_AgletsImpl`.

16.3 Agent Transfer Protocol

Το **Agent Transfer Protocol (ATP)** είναι ένα απλό πρωτόκολλο επιπέδου εφαρμογής, για την μεταφορά των agents μεταξύ κατακευμασμένων συστημάτων του δικτύου, που βασίζονται σε agents. Αν και τα mobile agents ενδέχεται να είναι προγραμματισμένα σε διαφορετικές γλώσσες και για ειδικές διαφορετικές πλατφόρμες, το ATP παρέχει την δυνατότητα χειρισμού της κινητικότητας των agents με έναν ομοιόμορφο και γενικευμένο τρόπο:

- Μία μηχανή μπορεί να φιλοξενεί διάφορους τύπους mobile agents, υπό την προϋπόθεση ότι υποστηρίζει τις αντίστοιχες agent πλατφόρμες.
- Κάθε agent πλατφόρμα πρέπει να περιλαμβάνει έναν handler ATP μηνυμάτων.
- Ένα ATP μήνυμα φέρει επαρκείς πληροφορίες για την αναγνώριση της εκάστοτε ειδικής agent πλατφόρμας και την κλήση του ATP handler της για το χειρισμό του.
- Κάθε μηχανή που φιλοξενεί agents διαθέτει μία υπηρεσία βασισμένη στο ATP, συστατικό ικανό για την αποστολή και λήψη των agents από απομακρυσμένες μηχανές, μέσω του ATP πρωτοκόλλου. Η agent υπηρεσία έχει μοναδική διεύθυνση αναγνώρισης, ανεξάρτητη των ειδικών τύπων πλατφόρμας που υποστηρίζονται από τη μηχανή. Μία μηχανή μπορεί να εκτελεί πολλαπλές agent υπηρεσίες.

Το σχήμα 45 δείχνει το παράδειγμα μίας μηχανής, που υποστηρίζει δύο agent πλατφόρμες και μία μοναδική ATP-based agent υπηρεσία, η οποία έχει μοναδική διεύθυνση.



Σχήμα 45: Οι ATP μέθοδοι αίτησης

Το ATP πρωτόκολλο βασίζεται στο παράδειγμα request/response μεταξύ των agent υπηρεσιών. Μία agent υπηρεσία A εγκαθιστά σύνδεση με μία B, στέλνει μία αίτηση στην B και περιμένει για την απάντηση. Μία αίτηση περιέχει μία γραμμή αίτησης, η οποία καθορίζει τη μέθοδο αίτησης, την έκδοση του πρωτοκόλλου και τον απαιτούμενο πόρο, ακολουθούμενη από ένα μήνυμα, παρόμοιο με MIME τύπου, το οποίο περιέχει πληροφορίες του αποστολέα και πιθανώς κάποιο περιεχόμενο στον κορμό του. Μία απάντηση περιλαμβάνει μία γραμμή κατάστασης, που δηλώνει την επιτυχία ή τον κωδικό λάθους και την έκδοση του πρωτοκόλλου, ακολουθούμενη από ένα μήνυμα που περιέχει πληροφορίες του αποστολέα και περιεχόμενο. Το ATP ορίζει τέσσερις συγκεκριμένες μεθόδους αίτησης για τις agent υπηρεσίες:

Dispatch

Η μέθοδος αυτή, ζητά από το agent σύστημα προορισμού, να επαναδομήσει το agent από το περιεχόμενο της αίτησης και να αρχίσει να το εκτελεί. Αν η αίτηση είναι επιτυχής, ο αποστολέας πρέπει να τερματίσει το agent και να απελευθερώσει τους πόρους που καταναλώνονται από αυτόν.

Retract

Με τη μέθοδο αυτή, ζητείται από το agent σύστημα προορισμού, να στείλει τον καθορισμένο agent πίσω στον αποστολέα. Αν το agent μεταφερθεί επιτυχώς, ο παραλήπτης της αίτησης πρέπει να τερματίσει το agent και να απελευθερώσει τους συσχετιζόμενους με αυτό πόρους.

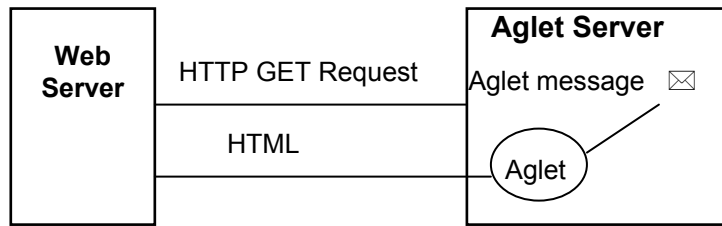
Fetch

Η μέθοδος αυτή είναι παρόμοια με την GET μέθοδο του HTTP. Ζητά από τον παραλήπτη να ανακτήσει και να στείλει οποιαδήποτε αναγνωρίσιμη πληροφορία.

Message

Χρησιμοποιείται για να περάσει ένα μήνυμα στο agent, που χαρακτηρίζεται από ένα agentID, και να επιστρέψει μία τιμή απάντησης. Αν και, το πρωτόκολλο ATP υιοθετεί μία φόρμουλα αίτησης/ απάντησης, δεν θέτει κανόνες για το σχήμα επικοινωνίας μεταξύ των agents.

Συνήθως, ένας ATP server προσπαθεί να δημιουργήσει μία άμεση σύνδεση με τους hosts του δικτύου. Ωστόσο, ένα agent δεν μπορεί να γίνει άμεσα dispatched ή retracted μέσω ενός firewall. Γι' αυτό, το ATP υποστηρίζει το μηχανισμό HTTP tunneling. Με την τεχνική αυτή, μία ATP αίτηση μπορεί να σταλθεί έξω από ένα firewall ως μία αίτηση HTTP POST. Η απάντηση ανακτάται πάλι ως μία HTTP απάντηση. Παρόμοια, ο ATP server μπορεί να διαμορφωθεί, ώστε να δέχεται HTTP-wrapped ATP αιτήσεις με περιεχόμενο τύπου application/ x-atp και να ανταποκρίνεται αντίστοιχα (Σχήμα 46).



Σχήμα 46: Οι πελάτες του Web εξυπηρετούνται από aglets που δέχονται HTTP wrapped ATP αιτήσεις.

ΜΕΡΟΣ Ε'

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

17.1 Εισαγωγή

Η επιτυχία του GSM οφείλεται μεταξύ άλλων στην δυνατότητα του χρήστη-συνδρομητή να μπορεί να κινείται σε διαφορετικά δίκτυα-και έτσι σε διαφορετικές χώρες-χρησιμοποιώντας μια μόνο συνδρομή. Αυτό φυσικά σημαίνει ότι ο συνδρομητής έχει ένα μόνο νούμερο κλήσης και λαμβάνει ένα λογαριασμό από το δίκτυο το οποίο έχει κάνει συνδρομή. Τα σημαντικότερα χαρακτηριστικά των μελλοντικών χρηστών κινητών υπηρεσιών θα είναι:

- Εύκολος χειρισμός των επιθυμητών υπηρεσιών, συμπεριλαμβανόμενου και της δυνατότητας τροποποίησης της υπηρεσίας ώστε να ικανοποιεί πλήρως τις απαιτήσεις του χρήστη, αλλαγή του "look and feel" στις παρεχόμενες υπηρεσίες και την συνδρομή "on demand".
- "Παγκόσμια" κάλυψη, διαθεσιμότητα και σταθερή απόδοση των προσφερόμενων υπηρεσιών
- Κατανοητή χρέωση από ένα μόνο παροχέα υπηρεσιών (home network -service provider)

Όμως τα μελλοντικά τηλεπικοινωνιακά συστήματα - δίκτυα δεν μπορεί να είναι ομοιογενή και έτσι αυτά τα χαρακτηριστικά δεν θα μπορούν να επιτευχθούν εύκολα. Ο σκοπός του VHE είναι να επιλύσει αυτά τα προβλήματα με το να παρέχει στο δίκτυο που επισκέπτεται ο χρήστης πληροφορίες σχετικά με τον χρήστη, το δίκτυο που έχει συνδρομή ο χρήστης, τις προτιμήσεις του και τις υπηρεσίες που θέλει να έχει πρόσβαση καθώς και άλλες χρήσιμες πληροφορίες.

Η πραγματοποίηση και η εκτέλεση κάθε υπηρεσίας μπορεί να διαφέρει από δίκτυο σε δίκτυο, το VHE είναι αυτό που θα παρέχει την δυνατότητα στον χρήστη να έχει πρόσβαση στην υπηρεσία που επιθυμεί με τον ίδιο τρόπο και με την ίδια μορφή που του παρέχεται στο δίκτυο που έχει κάνει συνδρομή.

Γενικά το VHE υποστηρίζει διαδικασίες έτσι ώστε σε ένα επισκεπτόμενο δίκτυο να παρέχονται υπηρεσίες από άλλα δίκτυα, με την ίδια παρουσία και συμπεριφορά (ίδιο look and feel) και με τις ίδιες δυνατότητες.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η παρουσίαση της έννοιας του VHE χρησιμοποιώντας την τεχνολογία των Mobile Agents, για την επίτευξη των λειτουργιών και την ανταλλαγή των απαραίτητων μηνυμάτων μεταξύ των διάφορων οντοτήτων που απαρτίζουν το δίκτυο, με σκοπό την απρόσκοπτη παροχή υπηρεσιών στους χρήστες.

Ήδη στο πρώτο και στο δεύτερο μέρος έχουν αναλυθεί-παρουσιαστεί αντίστοιχα η έννοια του VHE και της έννοιας του GPS (αντιπροσωπευτική των Location dependent services που αναλύονται και περιγράφονται στην παρούσα εφαρμογή). Στην συνέχεια αναλύεται η αρχιτεκτονική που χρησιμοποιήθηκε σε αυτή την εφαρμογή.

17.2 Αρχιτεκτονική εφαρμογής VHE και Location dependent Services.

Προκειμένου να επιτευχθεί το VHE επιβάλλεται η επικοινωνία μεταξύ διαφόρων ανομοιογενών δικτύων. Από την πλευρά του χρήστη το VHE θα επιτρέπει, όπως έχει ήδη αναλυθεί, την διαφανή χρήση υπηρεσιών στις οποίες έχει κάνει συνδρομή στο home του δίκτυο.

Από την πλευρά του διαχειριστή του συστήματος πρέπει να παρέχονται τα μέσα για την δημιουργία, εκτέλεση, ενημέρωση, τροποποίηση και απρόσκοπτη παροχή των υπηρεσιών αυτών. Επίσης για την ολοκλήρωση του VHE απαιτούνται διάφορες τροποποιήσεις και εμπορικές συμφωνίες μεταξύ των διαχειριστών των δικτύων και των παροχέων υπηρεσιών.

Υπάρχουν διάφορα σενάρια αρχιτεκτονικών για την δημιουργία του VHE που μπορούν να κατηγοριοποιηθούν ανάλογα με τα παρακάτω κριτήρια

- Θέση του ελέγχου υπηρεσίας
- Σχήμα κλήσης της υπηρεσίας
- Θέση των δεδομένων στο δίκτυο
- Διαθεσιμότητα υπηρεσίας
- Επιπρόσθετα μέτρα ασφάλειας

Σύμφωνα με αυτά τα κριτήρια υπάρχουν διαφορετικά σενάρια για το VHE και ένα εξ' αυτών είναι το **Shadow Home Service**. Σύμφωνα με αυτό το σενάριο όλες οι πληροφορίες και αρχεία (όπως το προφίλ του χρήστη για την παρεχόμενη υπηρεσία, τα δεδομένα, το απαιτούμενο πρόγραμμα για την υπηρεσία κλπ) προσωρινά "επισκιάζονται" ή μεταφέρονται από το home network του χρήστη στο δίκτυο στο οποίο βρίσκεται και αιτεί την υπηρεσία. Η υπηρεσία εκτελείται στο επισκεπτόμενο δίκτυο για την επίτευξη του VHE. Εδώ υλοποιείται επακριβώς αυτό το σενάριο και αποτελεί τη βάση της εφαρμογής αυτής για την παρουσίαση του VHE.

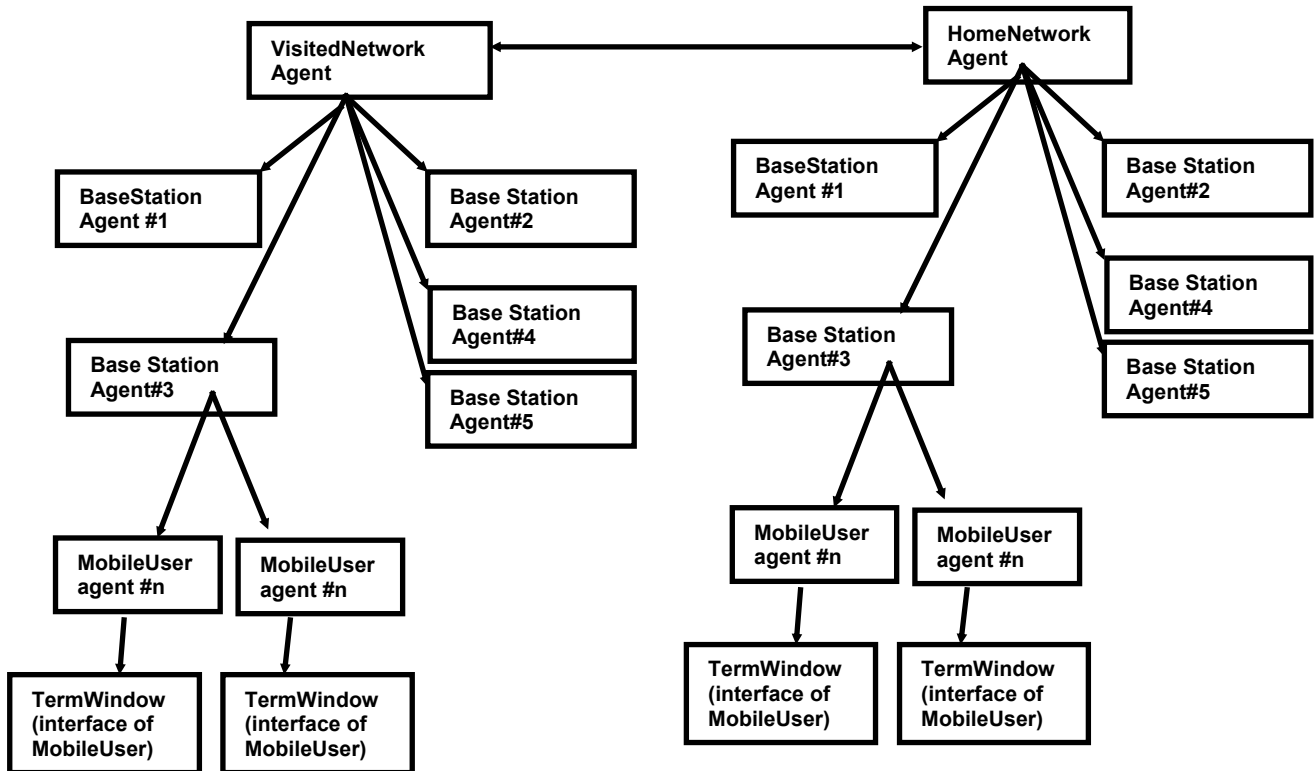
Στην παρούσα εργασία προκειμένου να επιτευχθεί το VHE καθώς και η παροχή υπηρεσιών που εξαρτώνται από το γεωγραφικό τόπο που βρίσκεται ο χρήστης (location dependent services) χρησιμοποιείται η τεχνολογία των Mobile Agents. Πρόκειται για ένα multi-agent σύστημα, το οποίο περιλαμβάνει agents (aglets) τα οποία επιτελούν διαφορετικές λειτουργίες. Τα aglets αυτά είναι στατικά και κινητά ανάλογα με τον ρόλο τους.

Το agent περιβάλλον αποτελείται από:

- Το Aglet HomeNetwork agents.
- Έναν αριθμό σταθμών βάσης (BaseStation) που δημιουργούνται από το HomeNetwork και ονομάζονται BaseStation
- Το Aglet VisitedNetwork agents
- Έναν αριθμό σταθμών βάσης (BaseStation) που δημιουργούνται από το VisitedNetwork, τα οποία είναι ακριβώς ίδια με τα προηγούμενα με διαφορετικές παραμέτρους αρχικοποίησης.
- Τους κινητούς χρήστες
- Το Aglet ServiceAgent που είναι υπεύθυνο για την παροχή των υπηρεσιών

Το Aglet HomeNetwork είναι το πρώτο που δημιουργείται και είναι υπεύθυνο για την αρχικοποίηση όλου του Aglet συστήματος. Αυτό γίνεται με την δημιουργία των πέντε BaseStations που απαρτίζουν το home δίκτυο του χρήστη. Έπειτα το ίδιο Aglet δημιουργεί το VisitedNetwork Aglet. Αυτό στην συνέχεια μπορεί να δημιουργήσει τα δικά του πέντε BaseStations. Παρακάτω αναλύονται οι λειτουργίες των Aglet αυτών, οι μεταξύ τους σχέσεις και τα μηνύματα που ανταλλάσσονται μεταξύ τους.

Όσον αναφορά στην υπηρεσία παροχής πληροφοριών θέσης του κινητού χρήστη έχει εφαρμοστεί η μέθοδος της μέτρησης του χρόνου άφιξης (UL-TOA) που αναλύθηκε στην παράγραφο 3.3. Στο σχήμα 47 φαίνεται η διάταξη του δικτύου και της ιεραρχίας μεταξύ των agent που χρησιμοποιούνται.



Σχήμα 47. Διάταξη δικτύου και ιεραρχία μεταξύ των agent.

17.3 Κύριες οντότητες για την εφαρμογή και την παρουσίαση του VHE και Location dependent Services με την χρήση των Aglets.

Σε αυτή την παράγραφο παρουσιάζονται αναλυτικά οι κύριες οντότητες που συμμετέχουν σε αυτή την εφαρμογή καθώς και η σχηματική αναπαράσταση όλων των σχέσεων μεταξύ όλων των οντοτήτων.

Οι κύριες οντότητες είναι:

HomeNetwork

Η οντότητα αυτή είναι ένα σταθερό-ακίνητο Aglet δουλειά του οποίου είναι να αρχικοποιεί το δίκτυο με το να δημιουργεί τα BaseStations του δικτύου και να τα αποστέλλει στους αντίστοιχους Aglet Servers. Επίσης, διαχειρίζεται όλα τα μηνύματα από και προς το δίκτυο (όπως registration σε κάποια υπηρεσία του χρήστη, authentication του χρήστη ο οποίος έχει πρόσβαση στο δίκτυο, παροχή υπηρεσιών, μηνύματα διαχείρισης από και προς τα Base Stations και από και προς άλλα δίκτυα). Παρέχει την δυνατότητα στον χρήστη να κάνει συνδρομή σε κάποια υπηρεσία ή το να χρησιμοποιήσει κάποια υπηρεσία από άλλο δίκτυο. Επίσης παρέχει την δυνατότητα παρακολούθησης της κίνησης σε ένα οποιοδήποτε Aglet Server, δηλαδή να έχει μια εποπτεία και εικόνα του server αυτού, παρέχοντας πληροφορίες για το πόσα και ποιά Aglets είναι ενεργά σε αυτόν τον server, ώρα δημιουργίας τους, από ποιο server δημιουργήθηκαν και σε ποιον ανήκουν.

VisitedNetwork

Η οντότητα αυτή είναι παρόμοια με το HomeNetwork και παρέχει τις ίδιες λειτουργίες με αυτό. Σε αντίθεση όμως από το δεύτερο αυτό δεν είναι σταθερό Aglet αλλά δημιουργείται από

το HomeNetwork και έπειτα αποστέλλεται στο κατάλληλο Aglet Server. Από την στιγμή που θα φθάσει μπορεί και αυτό να δημιουργήσει δικά του BaseStations και να λειτουργήσει όπως ακριβώς και το HomeNetwork με όλες τις λειτουργίες που αναφέρθηκαν στην προηγούμενη παράγραφο.

BaseStation

Η οντότητα αυτή προσομοιάζει ένα Base Station δικτύου της κινητής τηλεφωνίας. Δημιουργείται από το Home Network ή από το Visited Network και έπειτα μεταναστεύει σε κάποιο Aglet Server όπου παραμένει μέχρι το τέλος της λειτουργίας του. Το Aglet αυτό παρέχει πληροφορίες σε κάθε Mobile User Aglet το οποίο δημιουργείται ή έρχεται από κάποιο άλλο Server, αποτελεί τον ενδιάμεσο σε όλες τις επικοινωνίες του χρήστη προς το δίκτυο και από το δίκτυο προς αυτόν.

Παρέχει πληροφορίες για τις προσφερόμενες υπηρεσίες (όπως τις συντεταγμένες του για τον υπολογισμό της θέσης του Mobile User , την τοποθεσία στην οποία βρίσκεται για την παροχή των μετεωρολογικών πληροφοριών και χαρτών).

MobileUser

Το Aglet αυτό αναπαριστά τον κινητό χρήστη. Είναι υπεύθυνο για την διαχείριση μηνυμάτων προς το δίκτυο όπως μηνυμάτων για authentication του χρήστη, για συνδρομή σε κάποια υπηρεσία, αίτηση για κάποια υπηρεσία καθώς και την παροχή στον χρήστη των διαφόρων υπηρεσιών για τις οποίες έχει κάνει συνδρομή σε άλλα δίκτυα. Όλα αυτά παρέχονται στον χρήστη διαμέσου ενός user friendly interface το οποίο λέγεται TermWindow. Μετά την δημιουργία ή την άφιξη σε κάποιο Aglet Server αμέσως επικοινωνεί με το Aglet BaseStation για να δηλώσει την παρουσία του και να διαβάσει ορισμένες παραμέτρους από το δεύτερο για να μπορεί να αποκαταστήσει την επικοινωνία του με το BaseStation και κατ' επέκταση με το δίκτυο. Έπειτα ακολουθεί η διαδικασία authentication και η αίτηση παροχής κάποιας υπηρεσίας.

ServiceAgent

Το Aglet αυτό, όπως και το όνομα του υποδηλώνει, είναι αυτό που παρέχει τις διάφορες υπηρεσίες στον χρήστη. Δημιουργείται κάθε φορά από την οντότητα αυτή του δικτύου που μπορεί να παράσχει την υπηρεσία. Οντότητες οι οποίες μπορούν έτσι να δημιουργήσουν ένα ServiceAgent είναι α) το HomeNetwork Agent β) το VisitedNetwork Agent και γ) οποιοδήποτε BaseStation Agent.

Το agent αυτό με κάθε αίτηση για παροχή οποιοσδήποτε υπηρεσίας από ένα χρήστη, μετακινείται από τον Aglet Server που βρίσκεται η οντότητα που το δημιουργεί, στον Aglet Server που βρίσκεται το Aglet MobileUser που αναπαριστά τον χρήστη. Οι δυο Aglet Servers αυτοί μεταξύ των οποίων μετακινείται το Agent ServiceAgent μπορεί να βρίσκονται στον ίδιο υπολογιστή, σε υπολογιστές που ανήκουν σε ένα δίκτυο ή και σε δύο υπολογιστές που επικοινωνούν μέσω Internet. Η λειτουργία του ServiceAgent ουσιαστικά περιλαμβάνει την μετακίνηση του κώδικα που είναι αναγκαίος για την εκτέλεση μιας υπηρεσίας στην οντότητα αυτή του alget δικτύου που υποβάλλει αίτηση για μια υπηρεσία.

MyProxy

Το Aglet αυτό είναι μια επιπρόσθετη λειτουργία του HomeNetwork ή του VisitedNetwork για να μπορούν να εποπτεύσουν ένα Aglet Server ή και περισσότερους. Δημιουργείται από ένα από τα δυο αναφερθέντα Aglet μέσω του interface τους.

Αφού δημιουργηθεί, στην συνέχεια δημιουργεί ένα άλλο Aglet, που ονομάζεται MySlave, το οποίο αποστέλλει στον Aglet Server που επιθυμούμε να εποπτεύσουμε. Αυτό μόλις φθάσει στον προορισμό στέλνει πίσω στο MyProxy Aglet μια αναφορά με τα στοιχεία των Aglet, που υπάρχουν σε αυτό τον Aglet Server.

TermWindow

Η οντότητα αυτή είναι η διεπαφή του χρήστη. Είναι ένα Aglet που παρέχει στον χρήστη την δυνατότητα να κάνει login στο δίκτυο και να χρησιμοποιήσει οποιαδήποτε υπηρεσία.

CoordinateCalculator

Η οντότητα αυτή είναι μια κοινή κλάση που υπολογίζει την θέση του κινητού χρήστη. Χρησιμοποιεί τον αλγόριθμο υπολογισμού θέσης του GPS όπως αυτός περιγράφεται στην [49].

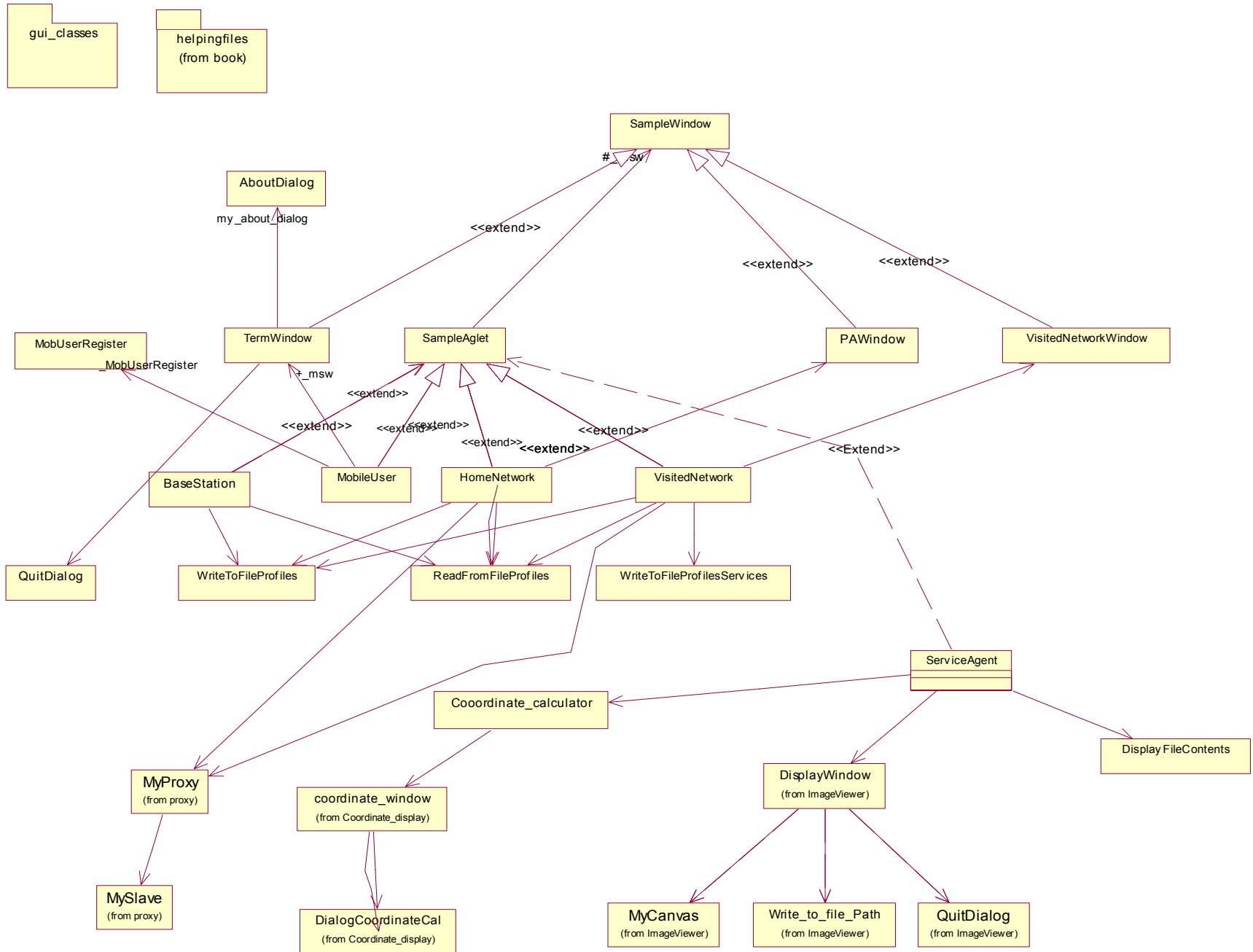
Η κλάση αυτή καλείται από το `coordinate_window` και παίρνει ως παραμέτρους την τοποθεσία των τριών γειτονικών BaseStations και του BaseStation που βρίσκεται ο χρήστης για τον υπολογισμό της θέσης του κινητού χρήστη. Ο υπόψη αλγόριθμος παίρνει σαν είσοδο τις γεωγραφικές θέσεις των BaseStations κάθε περιοχής ενώ οι αποστάσεις (ουσιαστικά η απόσταση του χρήστη εκφρασμένη σε χρόνο αφού έχει γίνει η κατάλληλη αντιστοίχιση) είναι τυχαίες (έχει χρησιμοποιηθεί η κλάση `Random` της `Java`) και τα αποτελέσματα είναι κάθε φορά διαφορετικά. Ο αλγόριθμος εμφανίζει ορισμένες μικρές αποκλίσεις από την πραγματική θέση του χρήστη.

DisplayWindow

Είναι το παράθυρο στο οποίο παρουσιάζονται οι χάρτες των περιοχών, όταν ο χρήστης χρησιμοποιεί την υπηρεσία παροχής χαρτών. Οι χάρτες αυτοί είναι γεωγραφικά προσανατολισμένοι ενώ παρέχονται με μεγάλη ακρίβεια οι συντεταγμένες του χάρτη καθώς μετακινείται το ποντίκι πάνω από τον χάρτη για κάθε σημείο του. Με ένα οποιοδήποτε κλικ πάνω σε αυτό το χάρτη παρουσιάζονται οι συντεταγμένες του pixel της εικόνας πάνω στο οποίο κάναμε κλικ.

Συνολικά όλα τα Aglets, οι κλάσεις και οι μεταξύ τους σχέσεις φαίνονται στο παρακάτω γράφημα (Σχήμα 48).

Σχήμα 48. Σχέσεις Χρησιμοποιούμενων Κάδρων

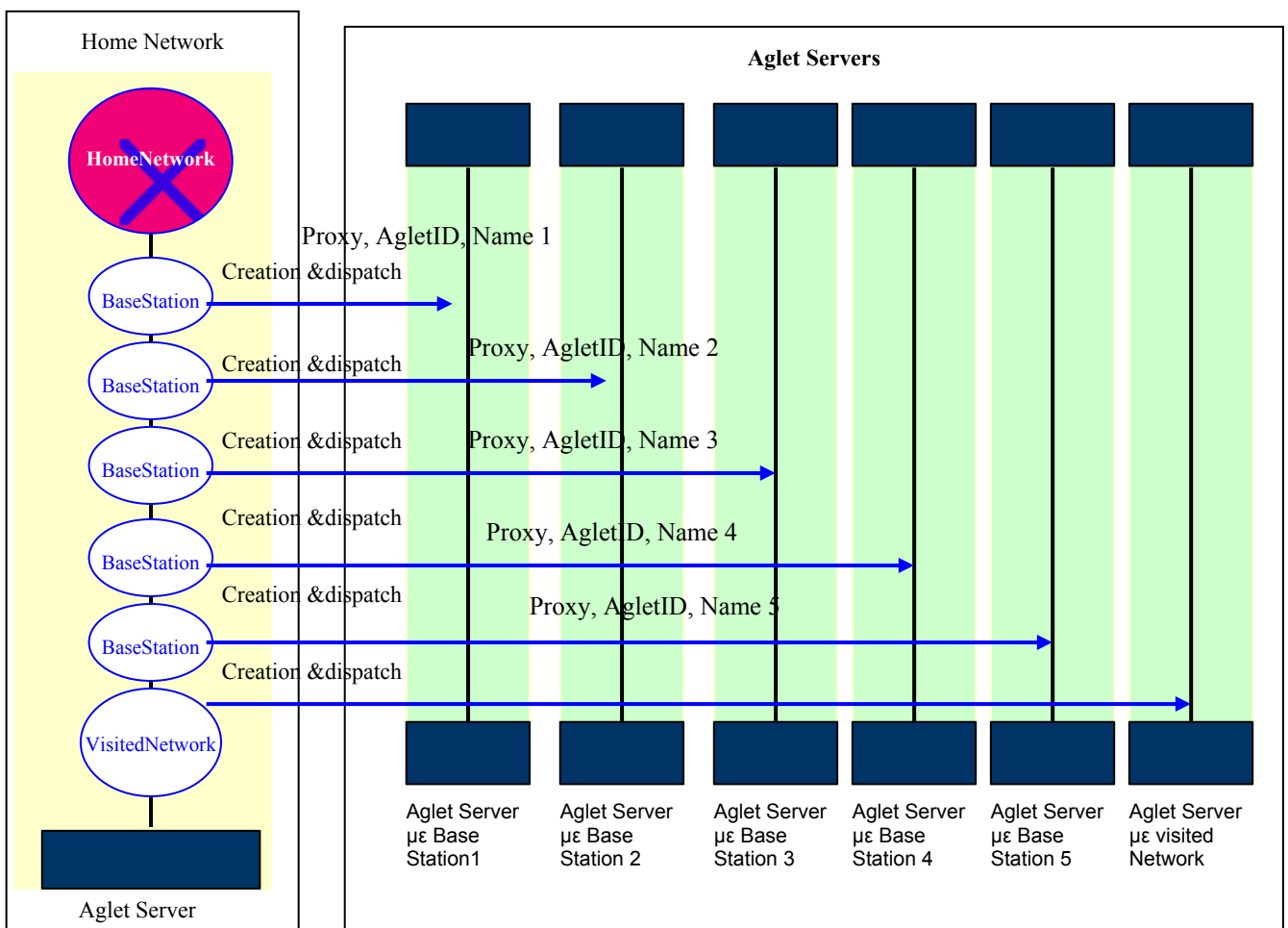


ΑΝΑΛΥΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Στην ενότητα αυτή παρουσιάζονται με λεπτομέρεια οι διάφοροι μηχανισμοί, καθώς επίσης, και όλες οι οντότητες των aglets που συμμετέχουν στην εφαρμογή. Οι μηχανισμοί-διαδικασίες μπορούν να κατηγοριοποιηθούν σε τρεις κατηγορίες: α) Μηχανισμοί αρχικοποίησης συστήματος β) Διαδικασίες Registration στο δίκτυο και πιστοποίησης χρήστη γ). Διαδικασία συνδρομής σε υπηρεσίες δ) Χρήση υπηρεσιών. Αναλυτικότερα παρουσιάζονται στις επόμενες παραγράφους.

18.1 Αρχικοποίηση του συστήματος

Η εκκίνηση του συστήματος ξεκινά με την δημιουργία του Aglet HomeNetwork Όπως προαναφέρθηκε, το HomeNetwork δημιουργεί τα aglets BaseStation, τα οποία στέλνει, με τη μέθοδο dispatch, στις διευθύνσεις των σταθμών βάσης του δικτύου, ένα σε κάθε διεύθυνση-Aglet Server (Σχήμα 49). Περνά παραμετρικά σε όλα τα BaseStation το Proxy, το AgletID και το όνομα του στο BaseStation που δημιουργεί. Οι δύο πρώτες παράμετροι χρησιμοποιούνται για την επικοινωνία των BaseStation με το HomeNetwork ενώ το τελευταίο χρησιμοποιείται για να καθορίσει το γεωγραφικό τόπο που θα λειτουργήσει το κάθε BaseStation. Τα τελευταία παραμένουν ενεργά σε κάθε έναν σταθμό, προκειμένου να διαχειρίζονται τα μηνύματα από και προς τους χρήστες.



Η ίδια ακριβώς διαδικασία συνεχίζεται από το VisitedNetwork Agent για την δημιουργία των δικών του BaseStations και την αποστολή τους στους αντίστοιχους Aglet Servers.

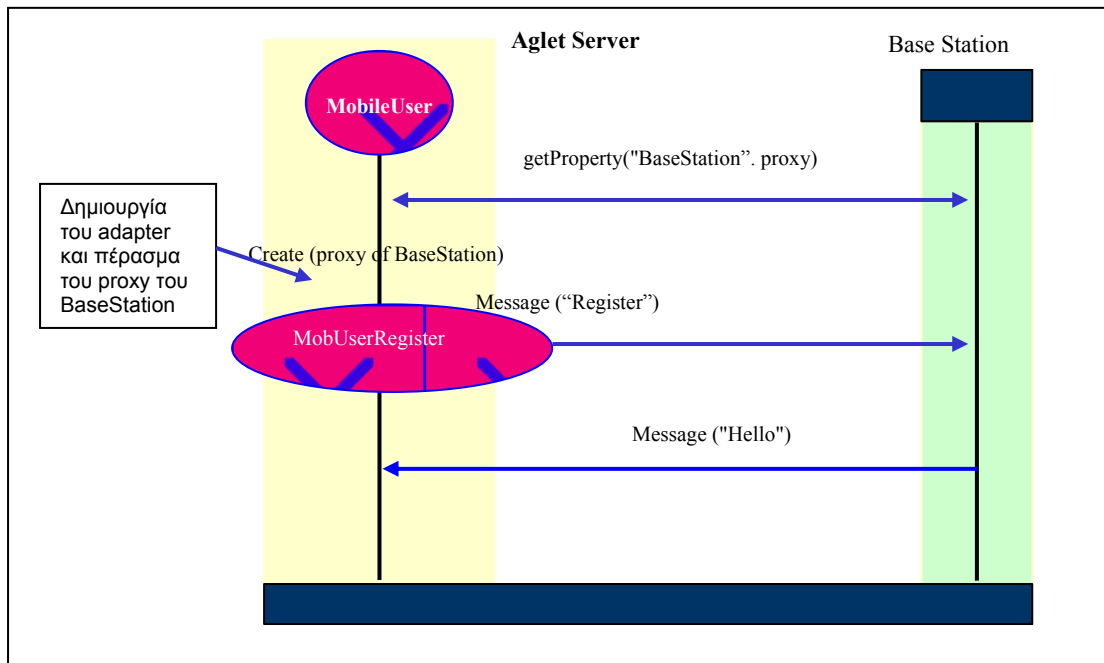
Τα BaseStations όταν φθάσουν στους Aglet Servers στέλνουν ένα μήνυμα "Hello" πίσω στο HomeNetwork και στο VisitedNetwork για να δηλώσουν ότι έφθασαν , ότι λειτουργούν κανονικά και είναι έτοιμα να δεχθούν κίνηση.

18.2 Διαδικασία δημιουργίας ενός χρήστη και εγγραφή του στο δίκτυο και πιστοποίηση χρήστη

Η δημιουργία ενός χρήστη γίνεται με την επιλογή από το μενού του Aglet Server, της εντολής create. Με την εντολή αυτή δημιουργούμε έτσι το Aglet που ονομάζεται MobileUser που εκπροσωπεί τον κινητό χρήστη στο δίκτυο το οποίο περιγράφεται. Από την στιγμή της δημιουργίας του χρήστη πραγματοποιούνται ορισμένες λειτουργίες για την εγγραφή του στο δίκτυο και την δημιουργία συνδέσεων που θα επιτρέπουν την αμφίδρομη επικοινωνία, με την μορφή μηνυμάτων, από το χρήστη προς το δίκτυο και αντίστροφα.

Η πρώτη ενέργεια του Aglet MobileUser από την στιγμή που θα δημιουργηθεί σε έναν Aglet Server όπου βρίσκεται και τρέχει ένα Aglet BaseStation είναι να διαβάσει τις ιδιότητες του BaseStation και ιδιαίτερα αυτήν που αφορά το proxy του. Όπως έχει αναφερθεί, ο proxy κάθε Aglet είναι η οντότητα μέσω της οποίας επικοινωνούν όλες οι άλλες οντότητες με αυτό το Aglet.

Από την στιγμή που το Aglet MobileUser διαβάσει τον proxy του BaseStation, τότε καλεί το Aglet MobUserRegister. Το δεύτερο Aglet είναι ο Adapter του Aglet MobileUser και είναι υπεύθυνο για την εγγραφή του στο δίκτυο όταν δημιουργείται ή φθάνει σε κάποιο Aglet Server καθώς και για την διαγραφή του όταν καταστρέφεται ή μετακινείται σε άλλο Server. Στην προκειμένη περίπτωση δημιουργίας, το MobileUser αρχικοποιεί τον adapter του , τον MobUserRegister, και έπειτα περνά σε αυτόν το proxy του BaseStation για να μπορέσει ο adapter να στείλει το μήνυμα για εγγραφή. Το BaseStation σε απάντηση αυτού του μηνύματος στέλνει στο MobUserRegister ένα μήνυμα " Hello". Η διαδικασία αυτή φαίνεται στο σχήμα 50.



Σχήμα 50 Δημιουργία και εγγραφή χρήστη.

Η επόμενη φάση μετά την εγγραφή στο δίκτυο είναι η πιστοποίηση του χρήστη. Από το interface του MobileUser, το TermWindow ζητείται η πληκτρολόγηση του ονόματος και του

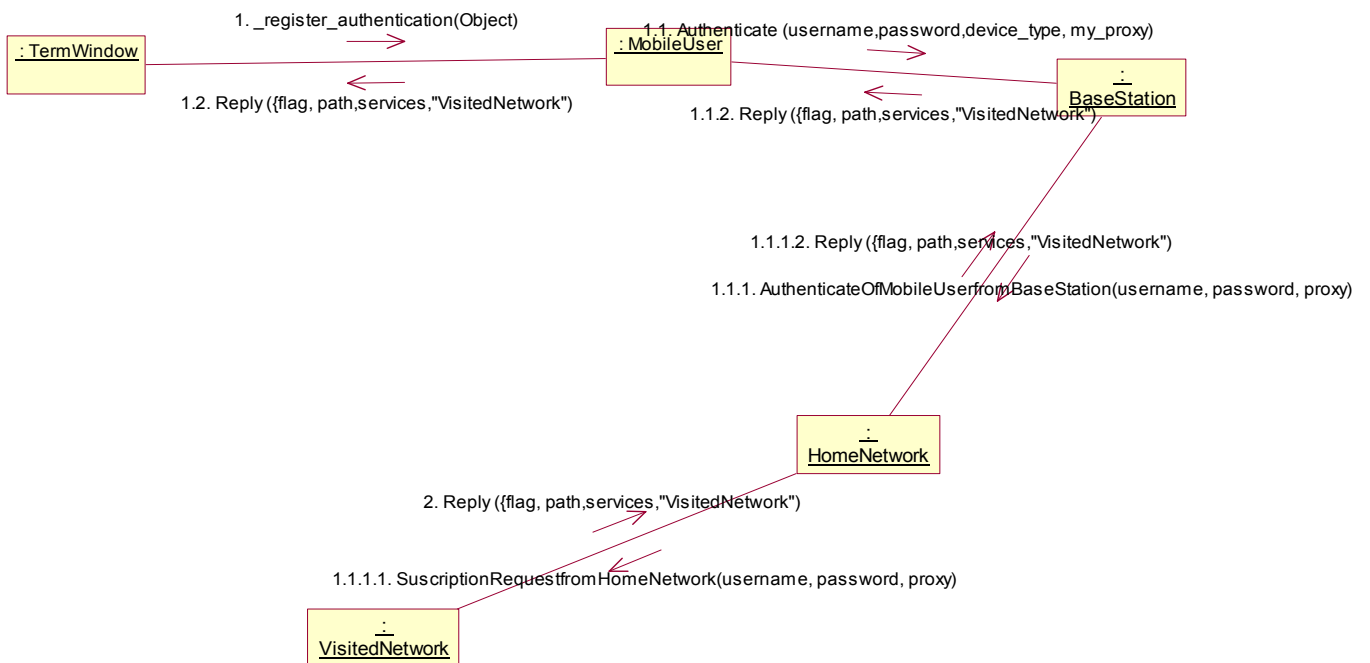
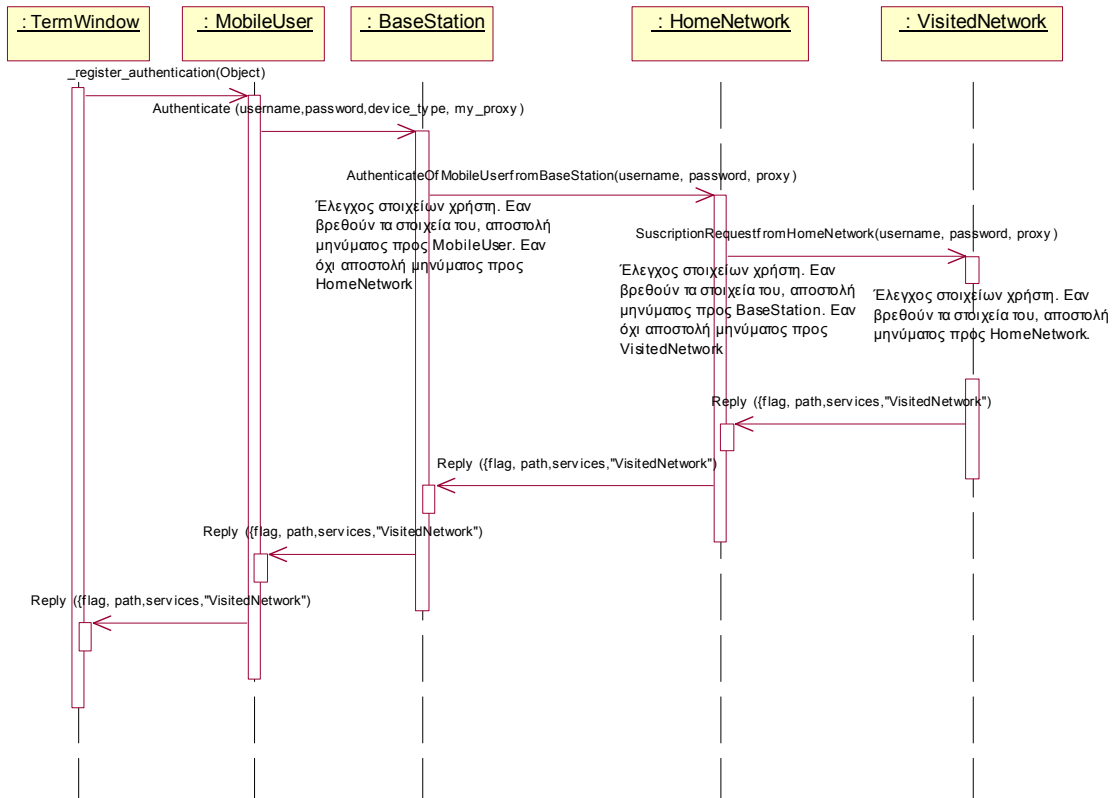
κωδικού του χρήστη. Επίσης ζητείται και το είδος της συσκευής που προτίθεται να χρησιμοποιήσει για αυτήν την σύνοδο. Με αυτές τις τρεις πληροφορίες (όνομα, κωδικό χρήστη και είδος συσκευής) το TermWindow καλεί την μέθοδο `_register_authentication` του `MobileUser`. Η μέθοδος αυτή δημιουργεί ένα μήνυμα το οποίο περιέχει τις τρεις παραπάνω πληροφορίες και επιπλέον το proxy του `MobileUser` και το αποστέλλει στο `BaseStation` που έχει επικοινωνήσει με την προηγούμενη διαδικασία.

Το `BaseStation` αφού λάβει αυτό το μήνυμα ελέγχει εάν έχει τέτοια στοιχεία στο αρχείο `User_Profiles.txt` μέσα στο οποίο υπάρχουν οι πληροφορίες για τους χρήστες που είναι υπεύθυνο το `BaseStation`. Εάν βρει αυτά τα στοιχεία τότε η επόμενη ενέργεια είναι να κοιτάξει εάν υπάρχει κάποιο αρχείο που να περιγράφει το προφίλ αυτού του χρήστη (στην περίπτωση μας στο αρχείο αυτό περιλαμβάνονται οι υπηρεσίες που έχει κάνει συνδρομή). Εάν το βρει τότε τα στοιχεία του αρχείου αυτού περιλαμβάνονται στο μήνυμα που αποστέλλεται πίσω στο `MobileUser`. Σε αυτό το μήνυμα περιλαμβάνονται επίσης, ένα `Flag` το οποίο δηλώνει στο `MobileUser` ότι πραγματοποιήθηκε η διαδικασία πιστοποίησης του χρήστη, ένα "μονοπάτι " στον υπολογιστή που τρέχει ο `Aglet Server` με το `BaseStation`, καθώς και ένα `string` "`BaseStation`" το οποίο δηλώνει ότι η πιστοποίηση έγινε από το `BaseStation`.

Εάν τα στοιχεία αυτά δεν βρεθούν στο `BaseStation` τότε αυτό δημιουργεί ένα μήνυμα με τις ίδιες παραμέτρους όπως αυτό που δέχθηκε από το `MobileUser` και το αποστέλλει στο `Aglet HomeNetwork`.

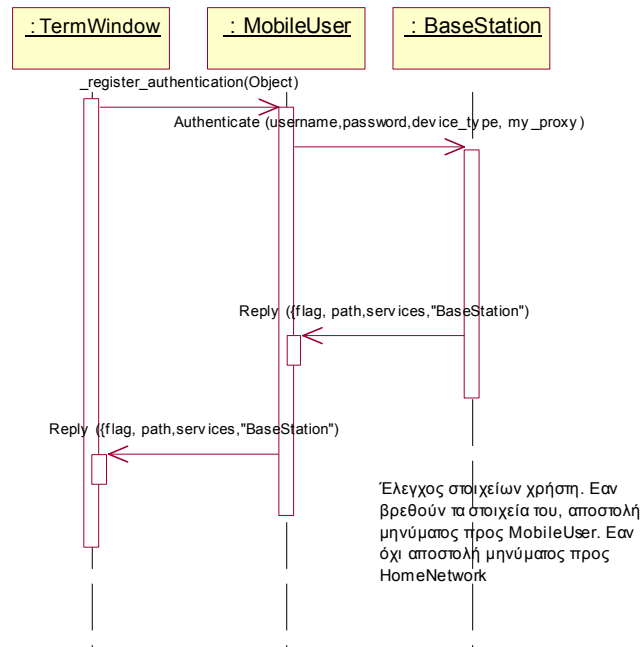
Το `HomeNetwork` επαναλαμβάνει την ίδια διαδικασία ανεύρεσης των στοιχείων του χρήστη στο δικό του αρχείο `User_Profiles.txt`. Εάν είναι επιτυχής η διαδικασία αυτή στέλνει μήνυμα-απάντηση στο `BaseStation` με τις ίδιες παραμέτρους όπως αυτού του μηνύματος που στέλνει το `BaseStation` στο `MobileUser` μόνο που η τιμή του `string` είναι τώρα "`HomeNetwork`".

Θα πρέπει επίσης να αναφερθεί ότι εναλλακτικά αντί κάθε οντότητα του δικτύου να εξετάζει τα διάφορα αρχεία (όπως `User_Profiles.txt` και `User_name.txt`) για να διαπιστώσει τα στοιχεία και τις προτιμήσεις του χρήστη, θα μπορούσε να χρησιμοποιηθεί μια βάση δεδομένων στην οποία η πρόσβαση θα γινόταν μέσω της τεχνολογίας `JDBC`. Στην παρούσα εργασία χρησιμοποιήθηκε η απλούστερη μορφή διότι οι χρήστες είναι λίγοι και η λειτουργία αυτή μπορεί να υλοποιηθεί με απλά αρχεία. Σε πιο σύνθετες μορφές και με περισσότερους χρήστες, περισσότερα εμπλεκόμενα δίκτυα η χρήση βάσεων δεδομένων και της τεχνολογίας `JDBC` θα ήταν επιβεβλημένη.



Σχήματα 51 - 52 Διαδικασία πιστοποίησης χρήστη (από το Aglet VisitedNetwork)

Στο σχήμα 51 φαίνεται η πιο extreme περίπτωση όπου την διαδικασία πιστοποίησης την επιτελεί το Aglet VisitedNetwork., ενώ στο σχήμα 52 φαίνεται αναλυτικά η ανταλλαγή των μηνυμάτων αυτής της διαδικασίας. Στο σχήμα 53 φαίνεται η πιο απλή περίπτωση όπου η διαδικασία πιστοποίησης γίνεται από το Aglet BaseStation



Σχήμα 53 Διαδικασία πιστοποίησης χρήστη (από το Aglet BaseStation)

Τέλος εάν και στο HomeNetwork δεν βρεθούν τα στοιχεία του χρήστη, τότε αυτό δημιουργεί ένα μήνυμα με τις ίδιες παραμέτρους όπως αυτό που δέχθηκε και το αποστέλλει στο Aglet VisitedNetwork. Το VisitedNetwork επαναλαμβάνει την ίδια διαδικασία ανεύρεσης των στοιχείων του χρήστη στο δικό του αρχείο User_Profiles.txt. Εάν είναι επιτυχής η διαδικασία αυτή στέλνει μήνυμα-απάντηση στο HomeNetwork με τιμή string " VisitedNetwork

Πρέπει να αναφερθεί επίσης ότι η διαδικασία είναι παρόμοια εάν ο χρήστης δημιουργηθεί σε ένα Aglet Server όπου τρέχει ένα BaseStation του VisitedNetwork. Η μόνη παραλλαγή είναι η αλληλουχία αποστολής των μηνυμάτων από την μια οντότητα στην άλλη. Στη διαδικασία που προηγουμένως περιγράφηκε η ακολουθία των μηνυμάτων ήταν: TermWindow->MobileUser->BaseStation -> HomeNetwork -> VisitedNetwork ενώ σε αυτή την περίπτωση είναι TermWindow -> MobileUser->BaseStation -> VisitedNetwork ->HomeNetwork.

Με την διαδικασία που περιγράφηκε σε αυτή την παράγραφο, δημιουργείται ο χρήστης σε κάποιον Aglet Server και γίνεται η αναγνώριση του από το δίκτυο καθώς και η εύρεση των προτιμήσεων των υπηρεσιών δηλαδή που έχει κάνει συνδρομή και του παρέχεται η δυνατότητα να τις χρησιμοποιήσει. Στην επόμενη παράγραφο αναλύεται η διαδικασία εγγραφής- συνδρομής ενός χρήστη σε κάποια υπηρεσία που παρέχεται από ένα δίκτυο.

18.3 Συνδρομή σε κάποια υπηρεσία

Στην παράγραφο αυτή περιγράφεται η διαδικασία σύμφωνα με την οποία ένας χρήστης αφού έχει εισέλθει στο δίκτυο, μπορεί να εγγραφεί σε κάποια υπηρεσία. Οι υπηρεσίες που είναι διαθέσιμες στο δίκτυο εμφανίζονται στο παράθυρο του χρήστη όπως φυσικά και οι υπηρεσίες στις οποίες έχει κάνει ήδη συνδρομή από κάποιο άλλο δίκτυο. Ο χρήστης μπορεί να επιλέξει την υπηρεσία που θέλει να κάνει εγγραφή -συνδρομή, να την προσθέσει στο προφίλ του και να μπορεί να την χρησιμοποιήσει όταν επιθυμεί.

Η διαδικασία ξεκινά από το TermWindow το οποίο είναι το interface του Aglet MobileUser. Αφού ο χρήστης επιλέξει κάποια υπηρεσία στην οποία θέλει να κάνει συνδρομή-εγγραφή γίνεται ένας έλεγχος για το εάν έχει κάνει ήδη συνδρομή σε αυτή την υπηρεσία. Οι πληροφορίες για το σε ποιές υπηρεσίες έχει κάνει συνδρομή παρέχονται από την προηγούμενη διαδικασία (Δημιουργία και Πιστοποίηση του χρήστη). Εάν διαπιστωθεί ότι ο χρήστης έχει ήδη εγγραφεί στην υπηρεσία που θέλει εκ νέου να κάνει εγγραφή, ενημερώνεται για αυτό το γεγονός και δεν του επιτρέπεται να κάνει νέα εγγραφή.

Στην περίπτωση που δεν έχει συνδρομή στην επιλεγμένη υπηρεσία, τότε το TermWindow καλεί την μέθοδο του Aglet MobileUser, `_Subscription_to_service(args)`. Σαν παράμετροι περνιούνται το όνομα του χρήστη, η υπηρεσία που θέλει να κάνει συνδρομή, το "μονοπάτι" στον υπολογιστή που τρέχει ο Aglet Server με την οντότητα που γνωστοποιήθηκε στο MobileUser σύμφωνα με την προηγούμενη διαδικασία (Δημιουργία και Πιστοποίηση του χρήστη), καθώς και το όνομα αυτής της οντότητας(πχ BaseStation, HomeNetwork ή VisitedNetwork). Η οντότητα αυτή είναι εκείνη που έκανε την πιστοποίηση του χρήστη και έχει τις πληροφορίες για το προφίλ του και κατά συνέπεια αυτή η οντότητα πρέπει να κάνει και την ενημέρωση στα αντίστοιχα αρχεία.

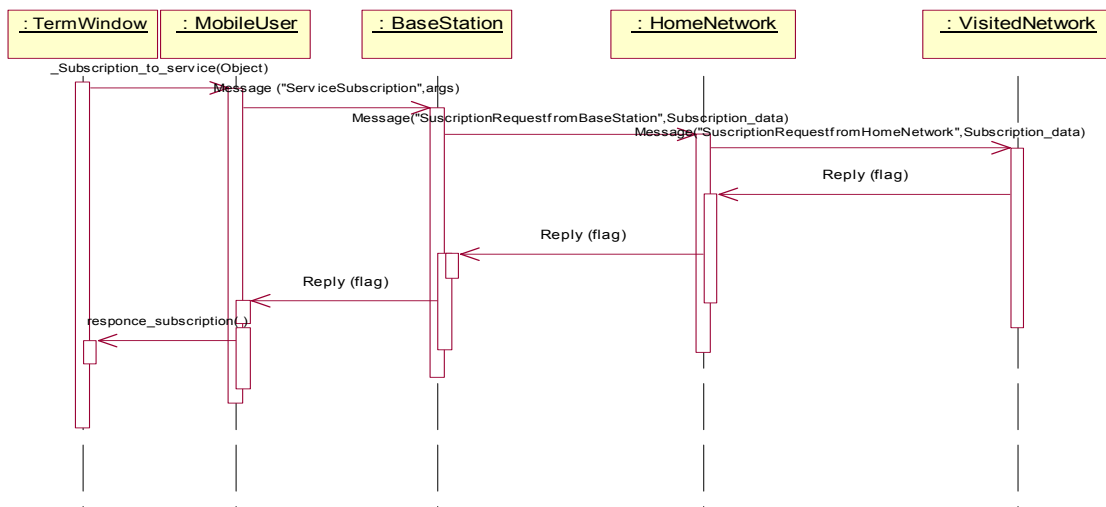
Η μέθοδος `_Subscription_to_service` του Aglet MobileUser, από την στιγμή που κληθεί, δημιουργεί ένα μήνυμα { Message ("ServiceSubscription",arguments) } με τις ίδιες παραμέτρους που έλαβε από το TermWindow προσθέτοντας μόνο το proxy του και το αποστέλλει στο BaseStation. Το τελευταίο με το που λάβει το μήνυμα, ελέγχει πρώτα εάν είναι υπεύθυνο αυτό για να επιτελέσει αυτή την λειτουργία. Ο έλεγχος αυτός γίνεται με το πεδίο του μηνύματος που αναφέρει ποια οντότητα είναι υπεύθυνη για την λειτουργία πιστοποίησης του χρήστη και ενημέρωσης του προφίλ του (αυτό το πεδίο είναι το τελευταίο του μηνύματος που αναλύθηκε προηγουμένως). Εάν το BaseStation είναι υπεύθυνο για αυτή την λειτουργία τότε αποστέλλει πίσω στο Aglet MobileUser ένα μήνυμα το οποίο περιέχει ένα Flag που δηλώνει ότι η διαδικασία εγγραφής στην υπηρεσία ήταν επιτυχής από το BaseStation, ενώ ταυτόχρονα ενημερώνει το αρχείο που περιέχει τις υπηρεσίες που έχει συνδρομή ο χρήστης. Το αρχείο αυτό έχει όνομα το όνομα του χρήστη (username.txt)

Στην περίπτωση που δεν είναι υπεύθυνο το BaseStation, η διαδικασία συνεχίζεται με την αποστολή ενός μηνύματος(Message("SubscriptionRequestfromBaseStation",Subscription_data)) από το τελευταίο προς το HomeNetwork. Οι παράμετροι (Subscription_data) είναι οι ίδιοι με το μήνυμα που έλαβε το BaseStation από το MobileUser. Το HomeNetwork αφού λάβει το μήνυμα επιτελεί και αυτό τον ίδιο έλεγχο που έγινε στο BaseStation για να εξακριβώσει εάν είναι υπεύθυνο να επιτελέσει την λειτουργία συνδρομής. Εάν είναι υπεύθυνο για αυτή την λειτουργία τότε αποστέλλει πίσω ένα μήνυμα το BaseStation, το οποίο το προωθεί στο MobileUser, το οποίο περιέχει ένα Flag που δηλώνει ότι η διαδικασία εγγραφής στην υπηρεσία ήταν επιτυχής από το HomeNetwork . Το τελευταίο Aglet κάνει την ενημέρωση του αρχείου username.txt.

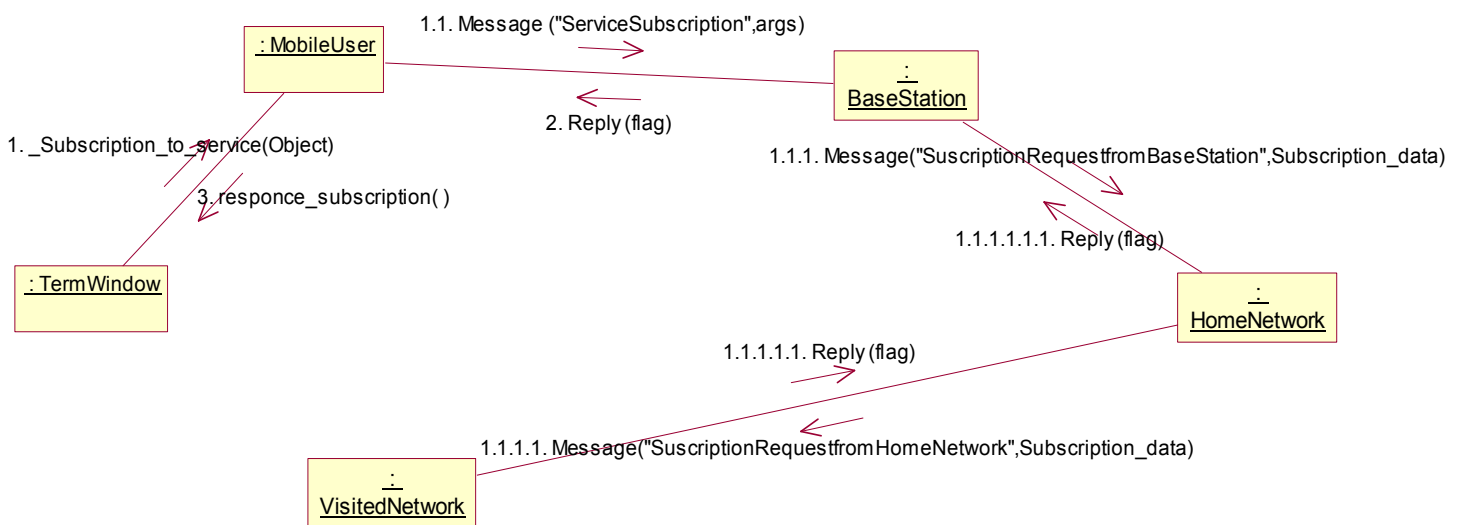
Εάν και το τελευταίο δεν είναι υπεύθυνο για την λειτουργία με την σειρά του στέλνει ένα μήνυμα (Message("SubscriptionRequestfromHomeNetwork",Subscription_data), το οποίο είναι το ίδιο με τα δύο προηγούμενα μηνύματα , στο VisitedNetwork. Γίνεται ο ίδιος έλεγχος που έγινε και στα δύο προηγούμενα Aglets και αποστέλλει στο HomeNetwork ένα Flag που δηλώνει ότι η διαδικασία εγγραφής στην υπηρεσία ήταν επιτυχής από το VisitedNetwork και κάνει την ενημέρωση του αρχείου username.txt.. Σε αυτό το σημείο τελειώνει αυτή η διαδικασία και δεν αποστέλλονται άλλα μηνύματα διότι τουλάχιστον μια οντότητα από την αλυσίδα που αναφέρθηκε (BaseStation-> HomeNetwork-> VisitedNetwork) θα έχει επιτελέσει την λειτουργία συνδρομής. Το τελευταίο είναι βέβαιο διότι μια από αυτές τις οντότητες είχε κάνει την πιστοποίηση του χρήστη κατά την διάρκεια της προηγούμενης διαδικασίας που περιγράφηκε στην προηγούμενη παράγραφο (Διαδικασία δημιουργίας ενός χρήστη και εγγραφή του στο δίκτυο και πιστοποίηση χρήστη).

Πρέπει να αναφερθεί επίσης ότι η διαδικασία είναι παρόμοια εάν ο χρήστης βρίσκεται σε ένα Aglet Server όπου τρέχει ένα BaseStation του VisitedNetwork. Η μόνη παραλλαγή είναι η αλληλουχία αποστολής των μηνυμάτων από την μια οντότητα στην άλλη. Στη διαδικασία που προηγουμένως περιγράφηκε η ακολουθία των μηνυμάτων ήταν: TermWindow->MobileUser->BaseStation -> HomeNetwork -> VisitedNetwork ενώ σε αυτή την περίπτωση είναι TermWindow -> MobileUser->BaseStation -> VisitedNetwork ->HomeNetwork.

Στο σχήμα 54 παρουσιάζεται η αλληλουχία ανταλλαγής μηνυμάτων που απαιτούνται για την εγγραφή ενός χρήστη σε μια υπηρεσία ενώ στο σχήμα 55 φαίνεται το Collaboration διάγραμμα της ίδιας διαδικασίας



Σχήμα 54. Διαδικασία συνδρομής σε κάποια υπηρεσία (από το Aglet VisitedNetwork)



Σχήμα 55. Collaboration διάγραμμα διαδικασίας συνδρομής σε κάποια υπηρεσία (από το Aglet VisitedNetwork)

18.4 Χρήση Υπηρεσιών

Στην παράγραφο αυτή αναλύονται τα μηνύματα που ανταλλάσσονται προκειμένου να προσφερθεί μια υπηρεσία στον χρήστη. Όπως έχει αναφερθεί οι προσφερόμενες υπηρεσίες είναι : α) Μετεωρολογικές πληροφορίες για την περιοχή όπου βρίσκεται ο χρήστης β) Παροχή χάρτου περιοχής χρήστη γ) Υπηρεσία παροχής πληροφοριών θέσης του χρήστη.

Η διαδικασία ως ένα σημείο είναι κοινή για όλες τις υπηρεσίες και για αυτό το λόγο πρώτα περιγράφεται το κοινό μέρος της διαδικασίας.

Η διαδικασία εκκίνησης των υπηρεσιών ξεκινά από το TermWindow. Ο χρήστης επιλέγει την υπηρεσία την οποία θέλει να χρησιμοποιήσει είτε από την λίστα των υπηρεσιών που έχει κάνει συνδρομή είτε από την λίστα των υπηρεσιών που του διατίθενται από το δίκτυο πατώντας το κουμπί Start Service..

Αφού επιλέξει την υπηρεσία, τότε το TermWindow καλεί την μέθοδο του Aglet MobileUser, `_Start_service((arguments_for_service))`. Σαν παράμετροι χρησιμοποιούνται, η υπηρεσία που θέλει να χρησιμοποιήσει, το είδος της συσκευής που χρησιμοποιεί καθώς και το όνομα της οντότητας του δικτύου που έκανε την πιστοποίηση του χρήστη κατά την διαδικασία *Δημιουργίας και Πιστοποίησης του χρήστη*.

Στο MobileUser γίνεται έλεγχος εάν υπάρχει ήδη ενεργό κάποιο Aglet που να είναι υπεύθυνο για την παροχή υπηρεσιών (αυτό το Aglet ονομάζεται ServiceAgent και θα αναλυθεί παρακάτω η λειτουργία του). Εάν ήδη υπάρχει και εάν προσφέρει την ίδια υπηρεσία με την αιτούμενη τότε αποστέλλεται ένα μήνυμα (`Message(arguments_for_service)`) σε αυτό το Aglet (ServiceAgent). Το μήνυμα αυτό αιτεί την παροχή της υπηρεσίας που επιθυμεί ο χρήστης.

Εάν υπάρχει κάποιο ενεργό ServiceAgent και προσφέρει άλλη υπηρεσία τότε αυτό καταστρέφεται από το Aglet MobileUser και ταυτόχρονα αποστέλλεται μήνυμα στο BaseStation το οποίο ζητά την παροχή της υπηρεσίας. Το μήνυμα αυτό περιέχει τις εξής παραμέτρους: α) Αιτούμενη υπηρεσία, β) είδος συσκευής, γ) υπεύθυνη οντότητα για την παροχή της υπηρεσίας και τέλος δ) το proxy του MobileUser.

Στο BaseStation επιτελούνται δύο έλεγχοι. Ο πρώτος έλεγχος γίνεται εάν για να εξακριβωθεί εάν ήταν αυτό που έκανε την πιστοποίηση του χρήστη και κατά συνέπεια είναι και υπεύθυνο για την παροχή της υπηρεσίας (την δημιουργία δηλαδή του ServiceAgent). Ο δεύτερος έλεγχος γίνεται για να διαπιστωθεί εάν για την αιτηθείσα υπηρεσία από το χρήστη, έχει ορισθεί από το δίκτυο να την παρέχει το BaseStation ανεξάρτητα εάν ήταν αυτό που έκανε την πιστοποίηση του χρήστη. Εάν συμβαίνει ένα από τα δυο προηγούμενα τότε καλείται η μέθοδος του BaseStation που ονομάζεται `_services_on_baseStation()` με παραμέτρους ίδιες με αυτές του μηνύματος από το MobileUser. Η μέθοδος αυτή δημιουργεί το Aglet ServiceAgent και το αποστέλλει με την μέθοδο **dispatch** στο Aglet Server του MobileUser (στην περίπτωση αυτή είναι ο ίδιος ο Aglet Server που τρέχει τόσο το Aglet MobileUser όσο και το Aglet BaseStation). Σε αυτό περνιούνται σαν ορίσματα οι παράμετροι που αναφέρθηκαν προηγουμένως.

Εάν κανένα από τα δύο δεν συμβαίνει, το BaseStation δεν έχει καμία αρμοδιότητα για παροχή της συγκεκριμένης υπηρεσίας, τότε αποστέλλεται ένα μήνυμα `{Message("Start_A_Service_from_BaseStation",arguments_for_service) }` στο HomeNetwork . Σε αυτό γίνονται οι ίδιοι έλεγχοι όπως και στο BaseStation. Εάν βρεθεί ότι είναι υπεύθυνο για την παροχή της υπηρεσίας τότε καλείται η μέθοδος `services_on_HomeNetwork(arguments_for_service)` η οποία αναλαμβάνει να δημιουργήσει το Aglet ServiceAgent και να το αποστέλλει με την μέθοδο `dispatch` στο Aglet Server του MobileUser.

Σε περίπτωση που και αυτό δεν είναι υπεύθυνο για την παροχή αυτής της υπηρεσίας, τότε αποστέλλει μήνυμα `{Message("Start_A_Service_from_HomeNetwork",arguments_for_service) }` στο VisitedNetwork το οποίο τελικά θα δημιουργήσει το Aglet ServiceAgent και θα το αποστέλλει στο Aglet Server του MobileUser, διότι είναι και το τελευταίο στην αλυσίδα των Aglet που είναι υπεύθυνο για την παροχή υπηρεσιών.

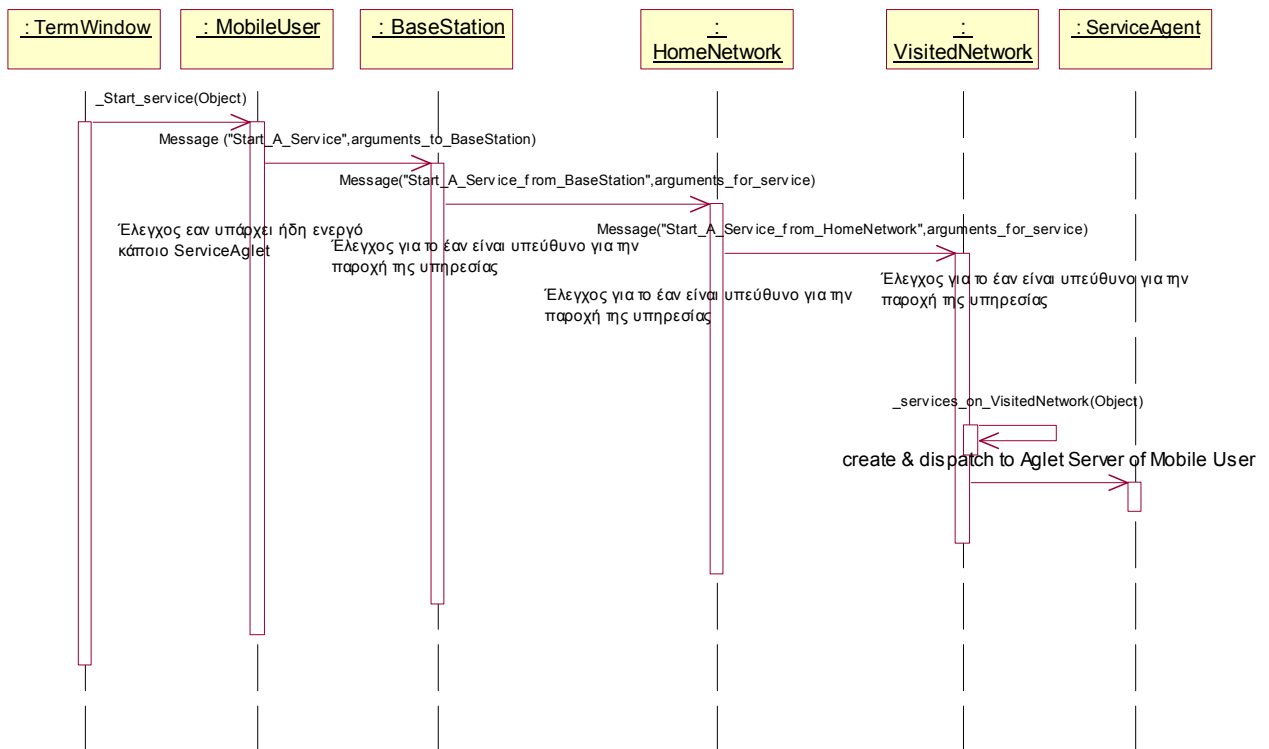
Η ως τώρα περιγραφείσα διαδικασία, είχε ως αποτέλεσμα την δημιουργία και την αποστολή του Aglet ServiceAgent στον Aglet Server του MobileUser. Από την στιγμή που

αφίχθει το Aglet ServiceAgent τότε ο χρήστης πρέπει να πατήσει δεύτερη φορά το κουμπί Start Service και τότε η υπηρεσία που είχε επιλέξει εμφανίζεται στην οθόνη του. Σε περίπτωση που στο μεσοδιάστημα, αίτησης της πρώτης υπηρεσίας και άφιξης του ServiceAgent, έχει επιλέξει άλλη υπηρεσία η διαδικασία επαναλαμβάνεται από την αρχή για την δημιουργία και αποστολή εκ νέου του Aglet ServiceAgent. Αυτό είναι απαραίτητο διότι η νέα υπηρεσία που έχει επιλέξει ο χρήστης μπορεί να παρέχεται από άλλη οντότητα του δικτύου και άρα πρέπει αυτή η οντότητα να δημιουργήσει το κατάλληλο ServiceAgent.

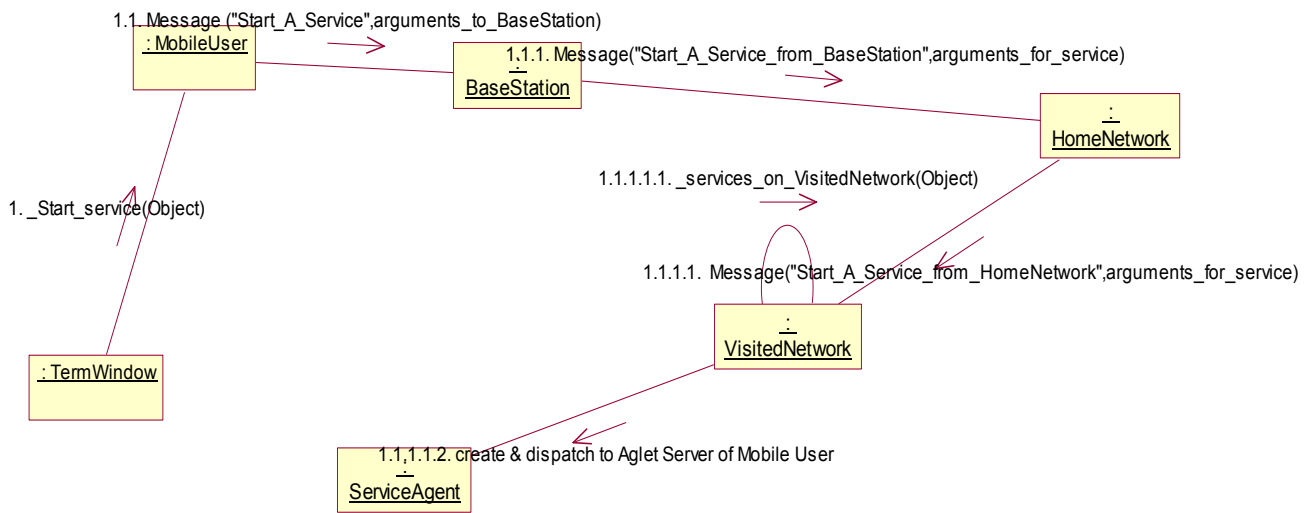
Όταν ο χρήστης πατά πάλι το κουμπί Start Service στο TermWindow, καλεί την μέθοδο του MobileUser _Start_service((arguments_for_service)) με τις ίδιες παραμέτρους όπως και στην αρχή. Στον σχετικό έλεγχο στο MobileUser διαπιστώνεται ότι υπάρχει ενεργό κάποιο ServiceAgent και προσφέρει αυτή την υπηρεσία. Τότε το μήνυμα(Message(arguments_for_service)) αποστέλλεται στο ServiceAgent. Το μήνυμα αυτό αιτεί την παροχή της υπηρεσίας που επιθυμεί ο χρήστης.

Όταν φθάσει το μήνυμα από το MobileUser στο Service Agent γίνεται ένας έλεγχος για να διαπιστωθεί ποια υπηρεσία ζητείται. Αφού διαπιστωθεί από το μήνυμα ποια υπηρεσία έχει αιτηθεί, τότε καλείται και ο αντίστοιχος μηχανισμός που θα παράσχει την υπηρεσία.

Η διαδικασία που περιγράφηκε για την παροχή υπηρεσιών στο χρήστη είναι κοινή για όλες τις υπηρεσίες. Αναλυτικότερα οι διαφοροποιήσεις των κάθε υπηρεσιών περιγράφονται στις επόμενες παραγράφους. Στο σχήμα 56 φαίνεται το γράφημα ανταλλαγής μηνυμάτων αυτής της διαδικασίας ενώ στο σχήμα 57 φαίνεται το Collaboration διάγραμμα της.



Σχήμα 56 .Διαδικασία παροχής υπηρεσιών (από το VisitedNetwork)



Σχήμα 57 . Collaboration διάγραμμα διαδικασίας παροχής υπηρεσιών (από το VisitedNetwork)

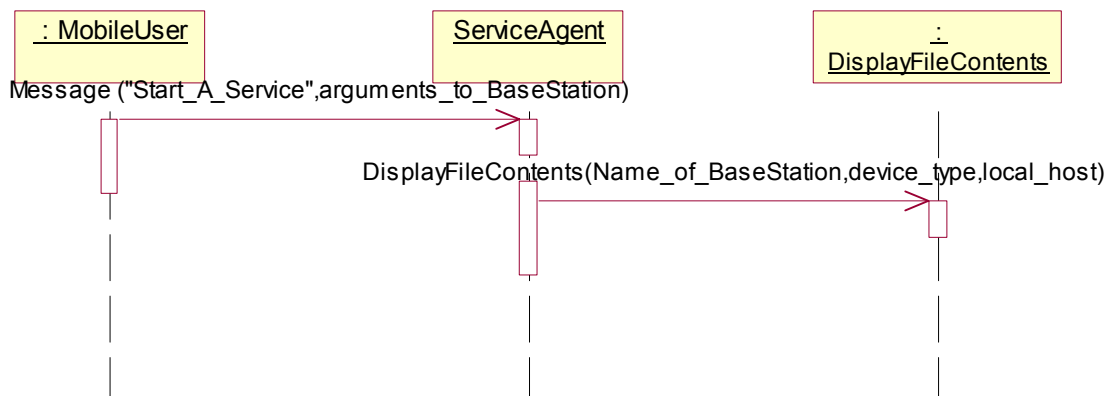
18.4.1 Χρήση υπηρεσίας Μετεωρολογικών πληροφοριών

Η υπηρεσία αυτή παρέχει στον χρήστη πληροφορίες για τον καιρό τον οποίο παρατηρείται και προβλέπεται στην περιοχή στην οποία βρίσκεται ο χρήστης. Τα μετεωρολογικά στοιχεία και η πρόγνωση που έχουν χρησιμοποιηθεί στην εργασία αυτή είναι της 15 και 16 Ιανουαρίου 2001.

Όταν το ServiceAgent λάβει μήνυμα να παράσχει την υπηρεσία αυτή, καλεί την κλάση (class) που ονομάζεται DisplayFileContents (Name_of_BaseStation,device_type,local_host). Τα ορίσματα που έχει είναι το όνομα του BaseStation- από το μήνυμα που έλαβε (για να γνωρίσει ποια αρχεία πρέπει να ανοίξει), το είδος της συσκευής που χρησιμοποιεί ο χρήστη - επίσης από το μήνυμα που έλαβε- και τέλος το όνομα του υπολογιστή που ανήκει στο δίκτυο που τρέχει η Aglet εφαρμογή μας.

Η κλάση DisplayFileContents ανοίγει το κατάλληλο αρχείο που περιέχει τις μετεωρολογικές πληροφορίες της περιοχής που βρίσκεται ο χρήστης και τις παρουσιάζει σε ένα παράθυρο που δημιουργεί. Το παράθυρο και οι παρουσιαζόμενες πληροφορίες σε αυτό αναλύονται στο επόμενο κεφάλαιο.

Το διάγραμμα των μηνυμάτων που ανταλλάσσονται φαίνονται στο σχήμα 58.



Σχήμα 58. Διάγραμμα των μηνυμάτων για την υπηρεσία μετεωρολογικών πληροφοριών

18.4.2 Χρήση υπηρεσίας παροχής χαρτών περιοχής χρήστη

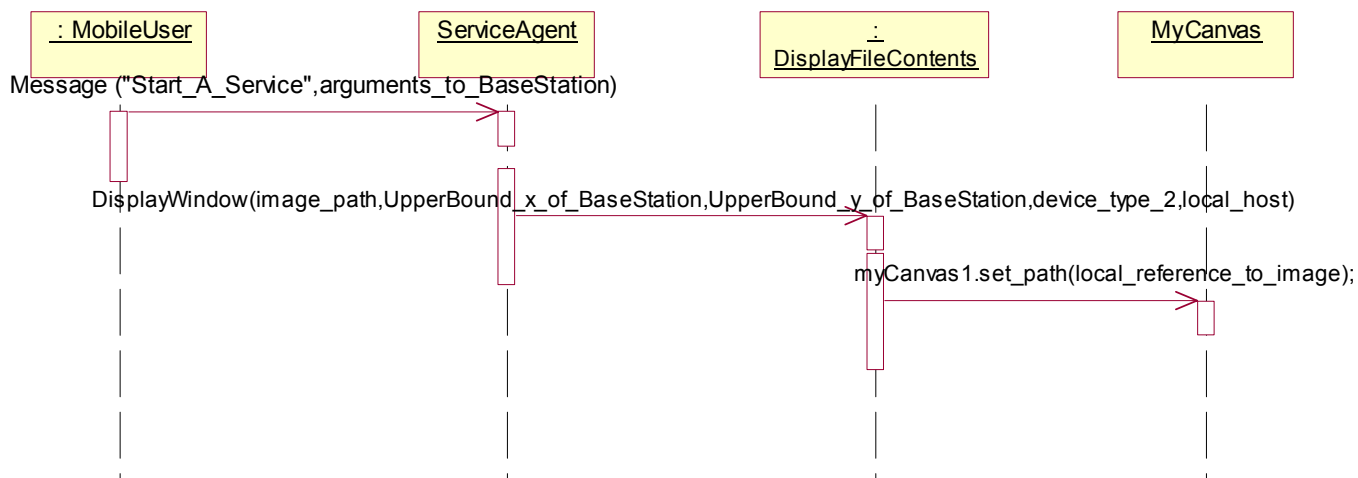
Η υπηρεσία αυτή παρέχει στο χρήστη χάρτη της περιοχής όπου βρίσκεται. Οι χάρτες που χρησιμοποιήθηκαν είναι γεωγραφικά προσανατολισμένοι και παρέχονται με τις αντίστοιχες συντεταγμένες τους που έχουν ακρίβεια 200 μέτρα.

Όταν το ServiceAgent αφιχθεί στο Aglet Server που βρίσκεται ο MobileUser διαβάζει τις ιδιότητες του BaseStation που αφορούν τις συντεταγμένες των σημείων που περικλείουν την περιοχή του, το όνομα του BaseStation (που στην πραγματικότητα είναι η γεωγραφική περιοχή στην οποία λειτουργεί) καθώς και τις ακριβείς συντεταγμένες που βρίσκεται το BaseStation.

Όταν το ServiceAgent λάβει μήνυμα να παράσχει την υπηρεσία αυτή, καλεί την κλάση που ονομάζεται DisplayWindow. Αυτή παίρνει σαν παραμέτρους το path στο οποίο θα βρει τον χάρτη που πρέπει να ανοίξει (ο οποίος είναι ένα αρχείο .jpg), τις συντεταγμένες του πάνω αριστερά σημείου του χάρτη αυτού, το είδος της συσκευής που χρησιμοποιείται καθώς και το όνομα του υπολογιστή που δημιούργησε το ServiceAgent. Η κλάση DisplayWindow στην συνέχεια καλεί την μέθοδο set_path(local_reference_to_image) της κλάσης MyCanvas με όρισμα το Path του jpg που αντιστοιχεί στο χάρτη της περιοχής.

Το interface της υπηρεσίας αυτής καθώς και η ανάλυση των πληροφοριών που παρέχονται μέσω αυτής παρουσιάζονται στο επόμενο κεφάλαιο.

Το διάγραμμα των μηνυμάτων που ανταλλάσσονται για την επίτευξη αυτής της υπηρεσίας φαίνονται στο σχήμα 59.



Σχήμα 59. Διάγραμμα των μηνυμάτων για την υπηρεσία παροχής χαρτών.

18.4.3 Χρήση υπηρεσίας παροχής πληροφοριών θέσης κινητού χρήστη

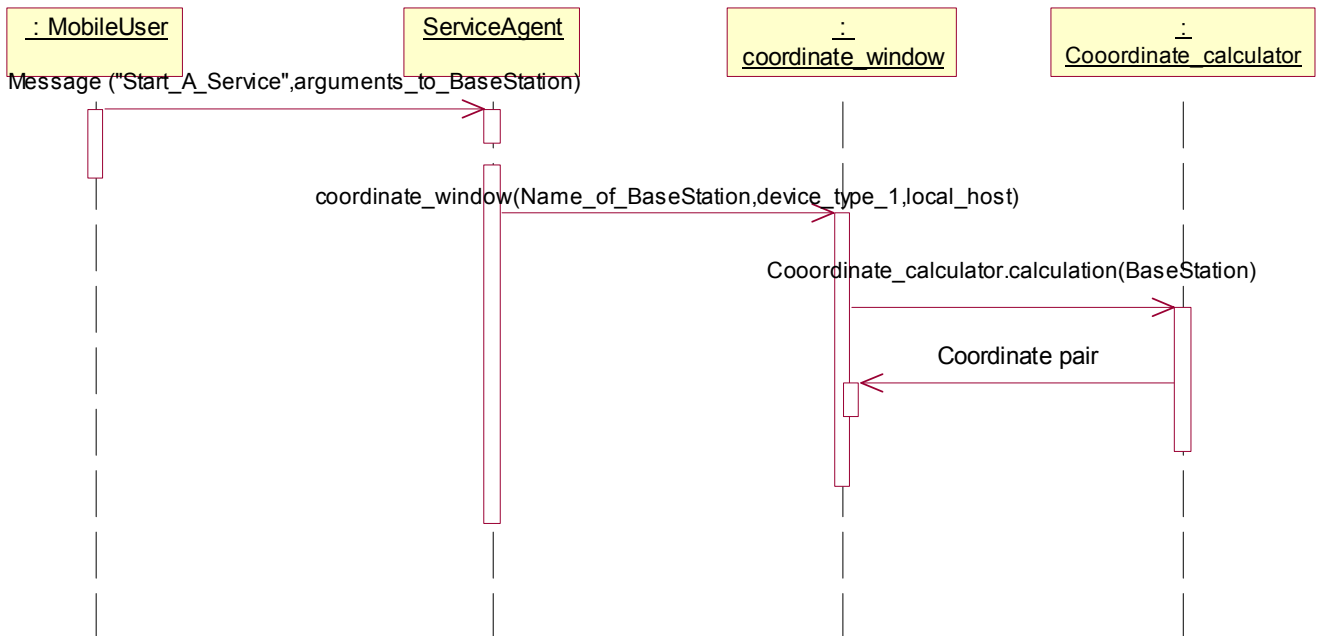
Η υπηρεσία αυτή παρέχει στον χρήστη την γεωγραφική θέση στην οποία βρίσκεται. Η γεωγραφική θέση παρέχεται σε μέτρα με αντίστοιχη αναφορά των συντεταγμένων στον άξονα x και στον άξονα Y συντεταγμένων σε προβολή UTM (Universal Transverse Mercator) και γεωδαιτικό σύστημα αναφοράς το WGS 84 (World Geodetic System 1984)

Όταν το ServiceAgent λάβει μήνυμα να παράσχει την υπηρεσία αυτή, καλεί την κλάση που ονομάζεται coordinate_window. Αυτή παίρνει σαν παραμέτρους το όνομα του BaseStation, τον τύπο της συσκευής που χρησιμοποιείται καθώς και το όνομα του υπολογιστή που δημιούργησε το ServiceAgent. Η κλάση coordinate_window είναι το interface αυτής της υπηρεσίας. Η υπόψη κλάση στην συνέχεια καλ εί την μέθοδο calculation της κλάσης Cooordinate_calculator με όρισμα το όνομα του BaseStation. Η κλάση Cooordinate_calculator έχει προκαθορισμένες συντεταγμένες για τρία σημεία γύρω από την περιοχή κάθε BaseStation τις οποίες μαζί με τις συντεταγμένες του εκάστοτε BaseStation τις χρησιμοποιεί σαν είσοδο στον αλγόριθμο υπολογισμού θέσης του χρήστη. Μαζί με τα τέσσερα αυτά ζεύγη συντεταγμένων χρησιμοποιούνται και τέσσερις τυχαίοι αριθμοί (παράγονται από την Random της java) και αναπαριστούν τον χρόνο που χρειάζεται το σήμα της συσκευής του χρήστη να πάει στους γύρω σταθμούς βάσης (τα τρία προκαθορισμένα σημεία με τις γνωστές συντεταγμένες) και να έρθει, διαιρεμένο δια του δύο. Ο χρόνος αυτός εάν διαιρεθεί με την ταχύτητα διάδοσης των ηλεκτρομαγνητικών κυμάτων δίνει την απόσταση που απέχει ο κινητός χρήστης από τα τέσσερα σημεία. Οι τυχαίες αποστάσεις και οι προκαθορισμένες συντεταγμένες των γύρω σταθμών βάσης μαζί με τις συντεταγμένες του σταθμού βάσης που βρίσκεται ο κινητός χρήστης χρησιμοποιούνται στον αλγόριθμο για την εύρεση της γεωγραφικής θέσης του χρήστη στον χώρο. Οι πληροφορίες θέσης που παρέχονται είναι μόνο για τους άξονες X και Y και παραλείπονται οι πληροφορίες για τον άξονα Z διότι για τον υπολογισμό αυτών χρειάζονται επιπλέον στοιχεία όπως πληροφορίες για την μορφολογία του εδάφους της περιοχής.

Αφού εκτελεστεί ο αλγόριθμος υπολογισμού θέσης τα αποτελέσματα του εμφανίζονται στο παράθυρο coordinate_window. Πρέπει να σημειωθεί ότι ο αλγόριθμος που χρησιμοποιήθηκε είναι ο ίδιος που χρησιμοποιούν οι εμπορικά διαθέσιμες συσκευές GPS. Θέτοντας πραγματικές συντεταγμένες (όπως και αυτές που χρησιμοποιήθηκαν στην εφαρμογή) και χρόνοι- αποστάσεις παρατηρήθηκαν κάποιες αποκλίσεις από τα πραγματικά σημεία αναφοράς.

Το interface της υπηρεσίας αυτής καθώς και η ανάλυση των πληροφοριών που παρέχονται μέσω αυτής παρουσιάζονται στο επόμενο κεφάλαιο.

Το διάγραμμα των μηνυμάτων που ανταλλάσσονται για την επίτευξη αυτής της υπηρεσίας φαίνονται στο σχήμα 60.



Σχήμα 60. Διάγραμμα των μηνυμάτων για την υπηρεσία παροχής πληροφοριών θέσης χρήστη

ΥΛΟΠΟΙΗΣΗ

19.1 Κώδικας

Για την υλοποίηση της παρούσας εφαρμογής με την τεχνολογία των Aglet χρησιμοποιήθηκε η έκδοση των Aglet ASDK 1.0.3, που λειτουργεί με εκδόσεις 1.1.x της Java. Χρησιμοποιήθηκε η έκδοση JDK 1.1.8 της Java. Το πρόγραμμα για την υλοποίηση του Aglet συστήματος περιλαμβάνει τον κώδικα των aglets, καθώς επίσης και προγράμματα κλάσεων που χρησιμοποιούνται από τις οντότητες αυτές. Όλες οι κλάσεις του κώδικα του συστήματος δίνονται στο Παράρτημα Α.

Αξίζει να σημειωθεί ότι, στο ASDK 1.0.3, τα aglets κατηγοριοποιούνται σε trusted και untrusted. Τα aglets θεωρούνται ότι είναι trusted, όταν η Code Base τους είναι τοπική, ενώ untrusted, όταν προέρχονται από διαφορετικούς aglet servers. Ένα trusted aglet έχει το δικαίωμα να δημιουργεί άλλα trusted aglets μέσα στον ίδιο aglet server. Για την κάθε aglet κατηγορία, ορίζονται ειδικά authorizations, που αποθηκεύονται σε προστατευμένα αρχεία και υποκαταλόγους, όπως τα \$HOME.aglets, \$CLASSPATH, \$AGLET_PATH και \$AGLET_EXPORT_PATH. Ο Tahiti (που είναι και ο Aglet Server) παρέχει τη δυνατότητα ρύθμισης των επιλογών ασφάλειας που αφορούν το FileSystem, το NetworkSystem και AgletProperties. Έτσι, πριν την εκτέλεση του προγράμματος, ορίζονται τα authorizations για το διάβασμα ή/και το γράψιμο αρχείων, από συγκεκριμένους καταλόγους και υποκαταλόγους, τόσο για τα trusted όσο και για τα untrusted aglets. Στην υπόψη εφαρμογή έχει απενεργοποιηθεί ο security manager των Aglets. Αυτό μπορεί να επιτευχθεί είτε δίνοντας την εντολή "agletsd -nosecurity -port number of port (έστω 1000) είτε θέτοντας στις ρυθμίσεις των Aglet στο αρχείο "Aglet_HOME/./aglets/security /trusted.file και untrusted.file (που ουσιαστικά είναι οι ιδιότητες που θέτουμε στον Aglet Server για το τι μπορούν να κάνουν τα trusted και untrusted aglets) τις παρακάτω ρυθμίσεις:

```
file.read= true
window.modal=true
window.warning=true
file.write= true
enable.rmserver=false
window.open=true
```

Στο μέρος Δ, αναλύθηκαν εκτενώς η τεχνολογία των aglets ενώ εδώ παρουσιάζονται αναλυτικά οι επιμέρους οντότητες που απαρτίζουν την εφαρμογή προσομοίωσης της έννοιας του VHE και την παροχή των εξαρτωμένων από την περιοχή που βρίσκεται ο χρήστης υπηρεσιών(Location dependent services).

19.2 Εκτέλεση

Στην παρούσα εφαρμογή χρησιμοποιήθηκαν Aglet Servers προκειμένου το Aglet σύστημα να προσομοιώσει τη λειτουργία των διαφόρων οντοτήτων του δικτύου. Απαιτήθηκαν πολλοί Aglet servers, οι οποίοι αποτελούν το κατάλληλο περιβάλλον για την εκτέλεση των aglets. Εδώ χρησιμοποιείται η πλατφόρμα των Windows 98. Η ανεξαρτησία πλατφόρμας τόσο της Java όσο και των Aglet επιτρέπει την χρήση οποιασδήποτε άλλης πλατφόρμας τόσο Windows based (NT, Millenium ή 2000);όσο και σε άλλα λειτουργικά συστήματα όπως OS2, Unix κλπ,

Η παρούσα προσομοίωση περιλαμβάνει τουλάχιστον δώδεκα Aglet Servers, δύο (2) για να τρέχουν σε αυτούς το Aglet HomeNetwork και VisitedNetwork, πέντε (5) για τα Aglet

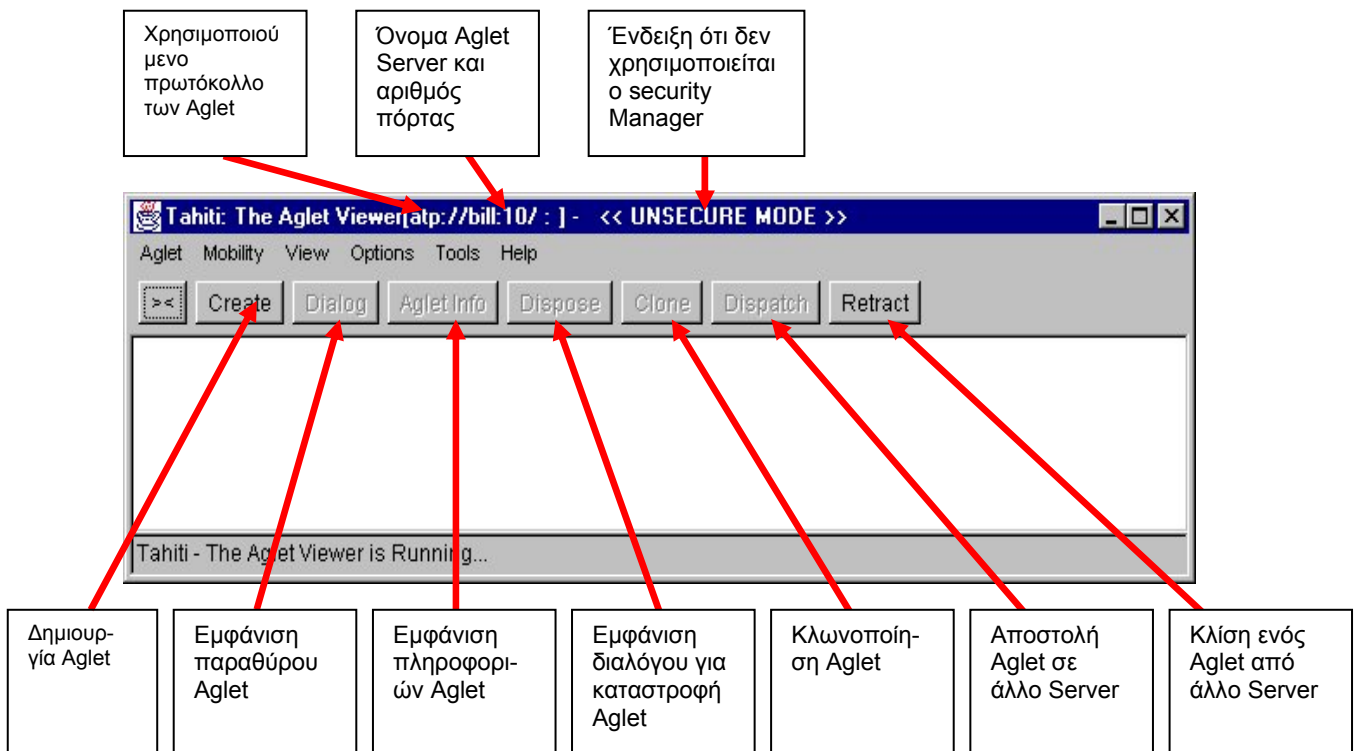
BaseStations του HomeNetwork και πέντε(5) για τα Aglet BaseStations του VisitedNetwork. Τα ονόματα και οι αριθμοί των χρησιμοποιούμενων πορτών των aglet server που αναφέρονται παρακάτω είναι τυχαίοι και αναφέρονται με αυτά μόνο για σκοπούς παρουσίασης (τα ονόματα **bill, elena** έχει προκύψει τυχαία). Επίσης τα ονόματα αυτά μπορούν να τροποποιηθούν κατά το setting των Aglet όπως και οι αριθμοί των πορτών που χρησιμοποιούνται όταν ξεκινάμε έναν τέτοιο server. Για την εκτέλεση του προγράμματος του όλου συστήματος, ενεργοποιούνται οι εξής aglet servers :

atp://bill: 10, ο οποίος προσομοιώνει το περιβάλλον των aglets στον κόμβο του HomeNetwork
atp://bill: 1000, atp://bill: 2000, atp://bill: 3000, atp://bill: 4000, atp://bill: 5000 μέσα στους οποίους τρέχουν τα aglets BaseStation που αναπαριστούν τους σταθμούς βάσης του δικτύου του HomeNetwork

atp://elena: 10, ο οποίος προσομοιώνει το περιβάλλον των aglets στον κόμβο του VistedNetwork

atp://elena: 1000, atp://elena: 2000, atp://elena: 3000, atp://elena: 4000, atp://elena: 5000 μέσα στους οποίους τρέχουν τα aglets BaseStation που αναπαριστούν τους σταθμούς βάσης του δικτύου του VistedNetwork

Το interface του Aglet Server (που ονομάζεται Tahiti) φαίνεται στο σχήμα 61 :

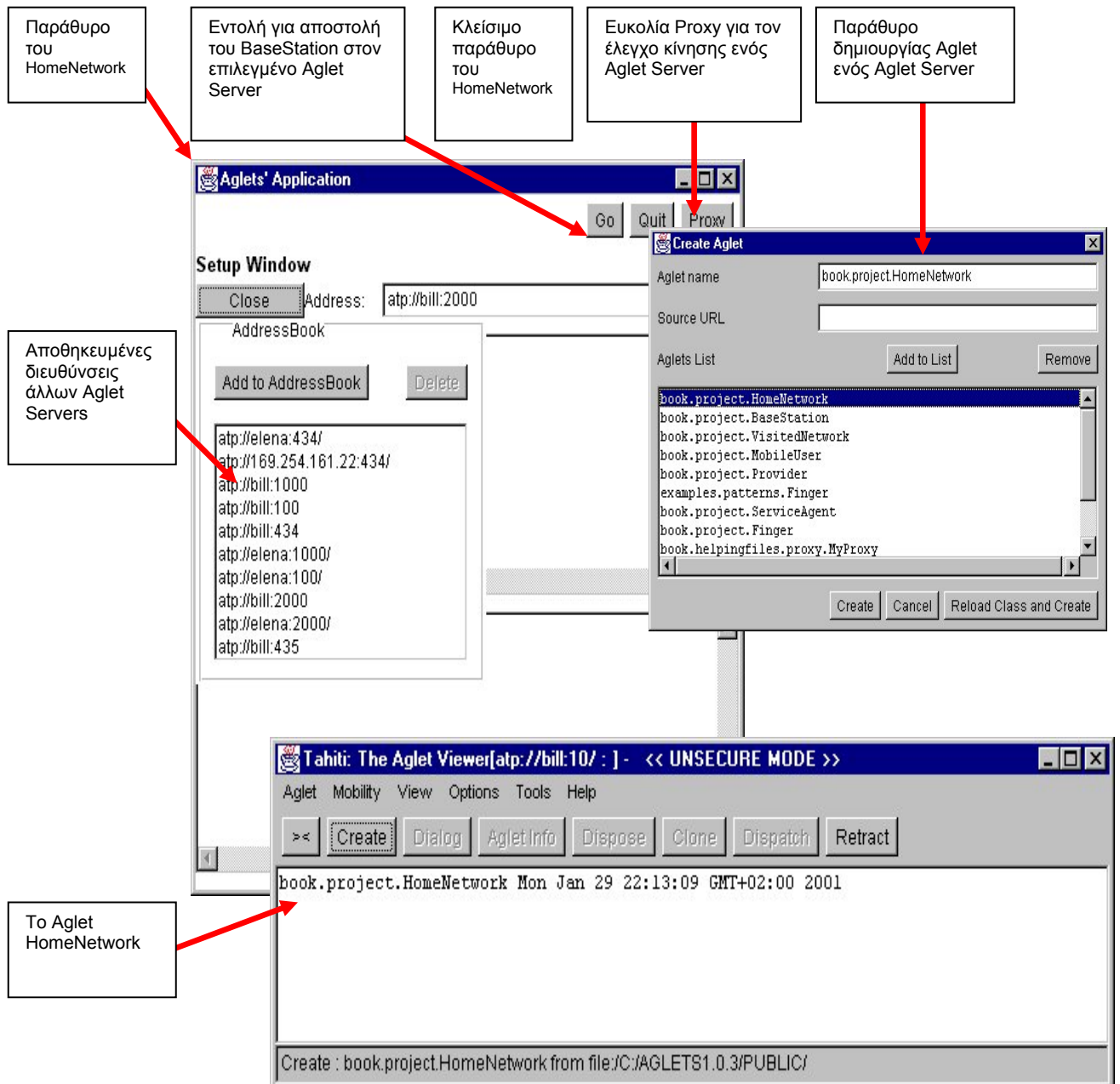


Σχήμα 61. Το interface του Tahiti Aglet Server

Ο server αυτός παρέχει ορισμένες λειτουργίες για την διαχείριση, δημιουργία και καταστροφή των Aglet, όπως αναλύονται στο γράφημα του σχήματος 61.

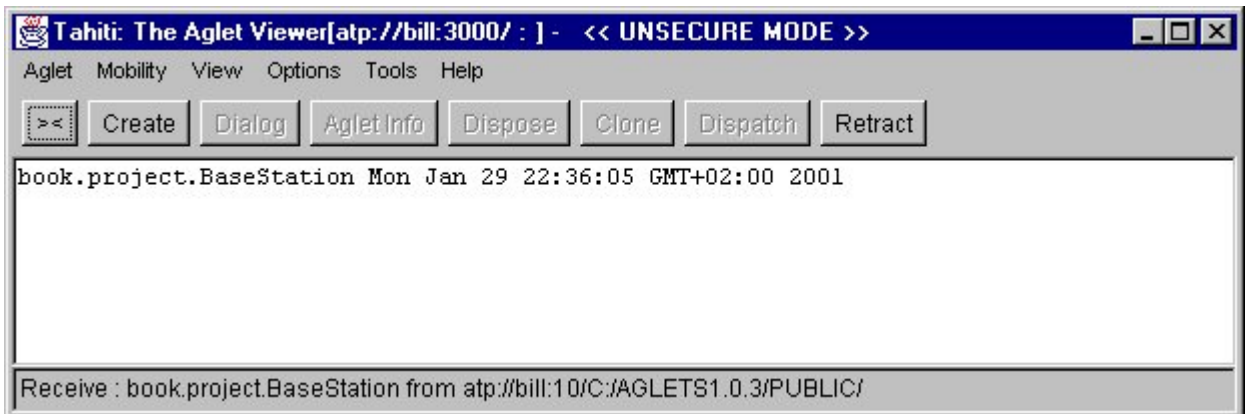
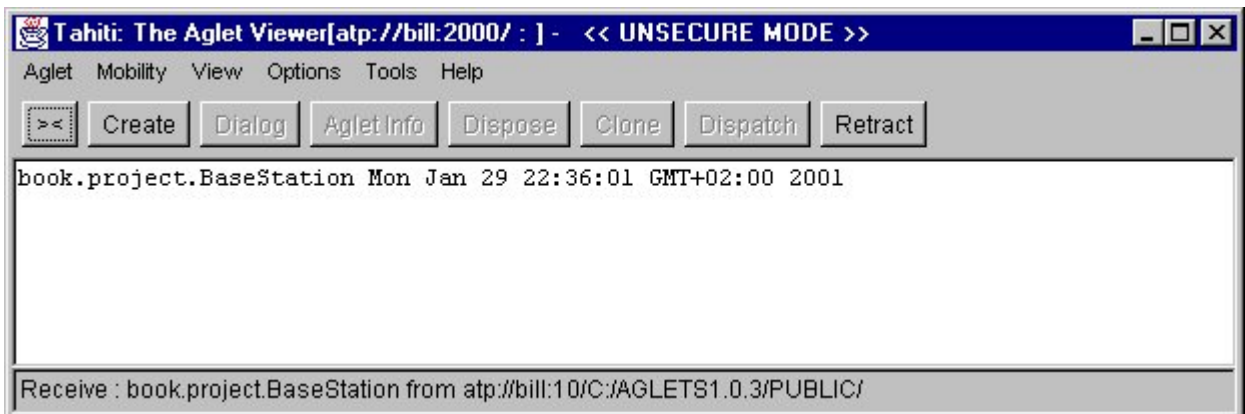
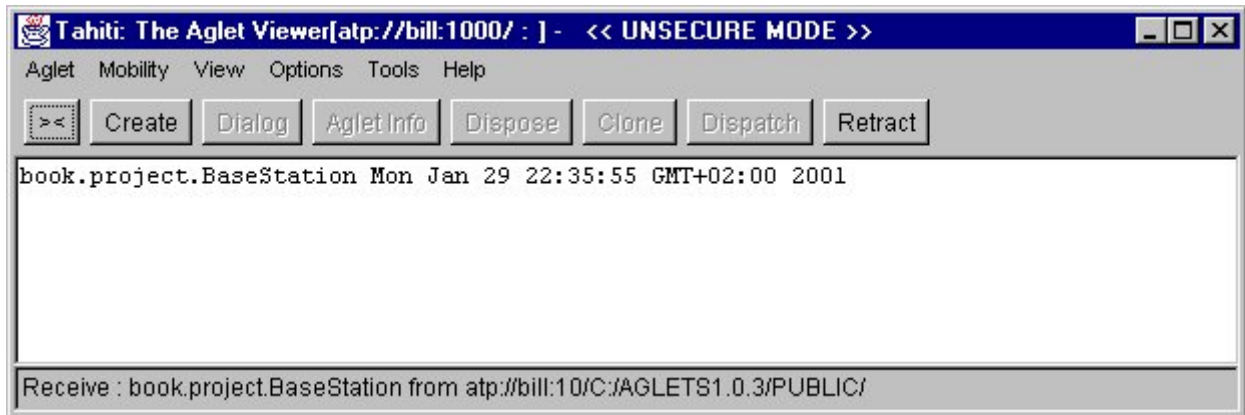
Η εφαρμογή ξεκινά με την δημιουργία του HomeNetwork στον Aglet Server bill :10

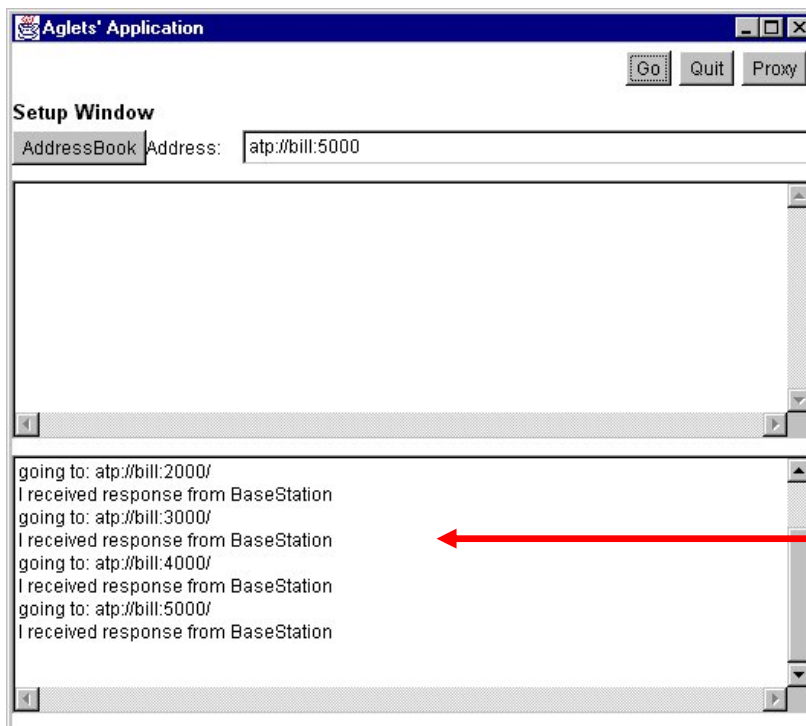
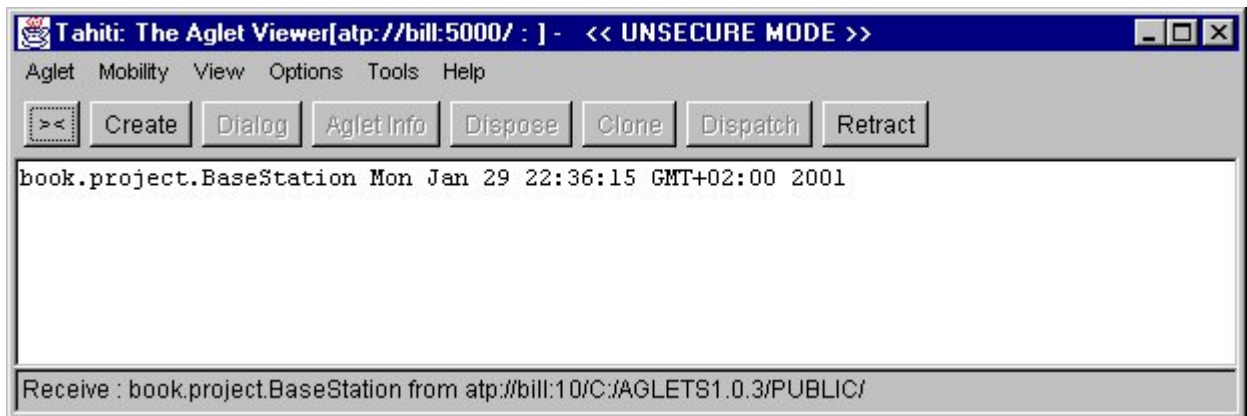
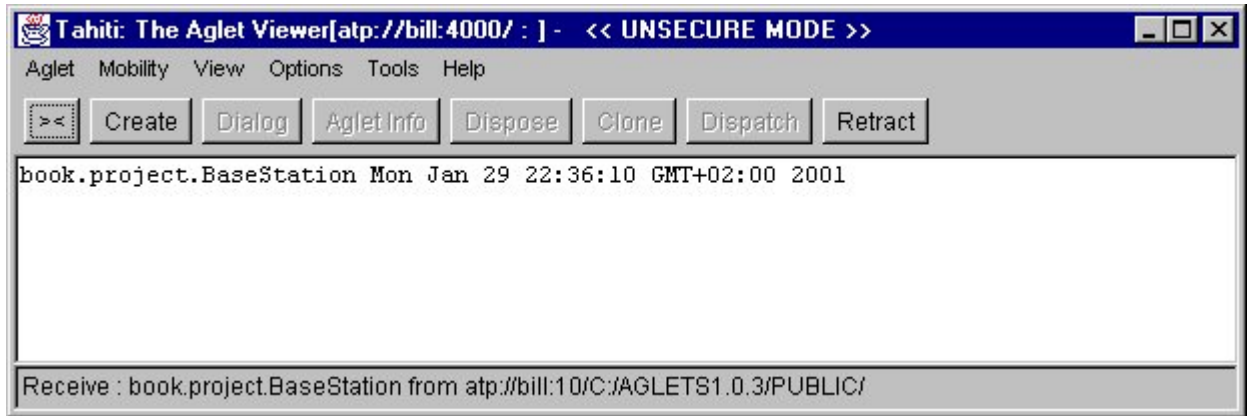
Από την στιγμή που θα δημιουργηθεί το HomeNetwork, εμφανίζεται το παράθυρο του που προτρέπει να δώσουμε τις διευθύνσεις των Aglet Server που θα πάνε τα πέντε BaseStation. (Σχήμα 62)



Σχήμα 62: Δημιουργία του HomeNetwork και των πέντε BaseStations.

Επιλέγουμε πέντε διαθέσιμες διευθύνσεις και αποστέλλουμε τα αντίστοιχα BaseStation. Με την επιλογή της κάθε διεύθυνσης πατάμε και το κουμπί **Go** για να αποστείλει το HomeNetwork με την μέθοδο `dispath` τα BaseStation στους επιλεγμένους Servers, που υποτίθεται ότι είναι ήδη ενεργοποιημένοι. Αυτοί οι servers μπορεί να είναι σε οποιοδήποτε υπολογιστή του δικτύου που μπορεί να τρέξει η εφαρμογή. Το Aglet, BaseStation με το που θα φθάσει στο server που το αποστέλλουμε στέλνει μήνυμα στο HomeNetwork ότι έφθασε στον server. Στο σχήμα 63 φαίνονται οι servers μετά την αποστολή των πέντε BaseStations μαζί με το παράθυρο του HomeNetwork.

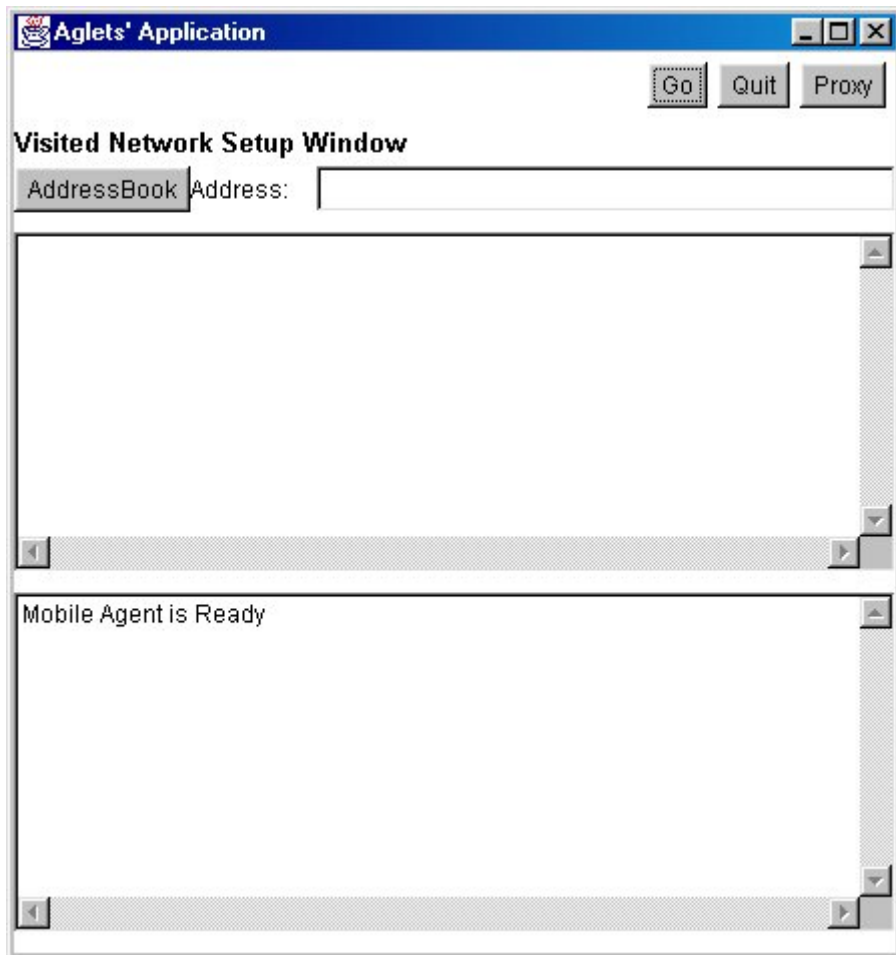
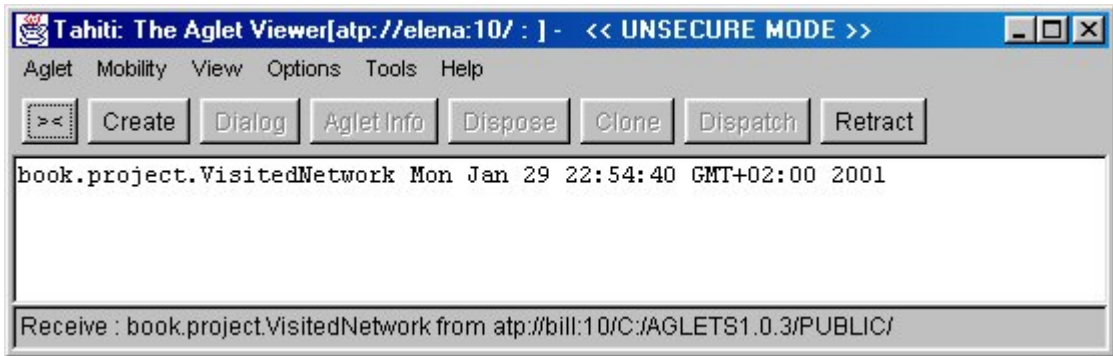




Περιοχή εμφάνισης μηνυμάτων από τα BaseStation για το ότι έφθασαν στους Aglet Servers

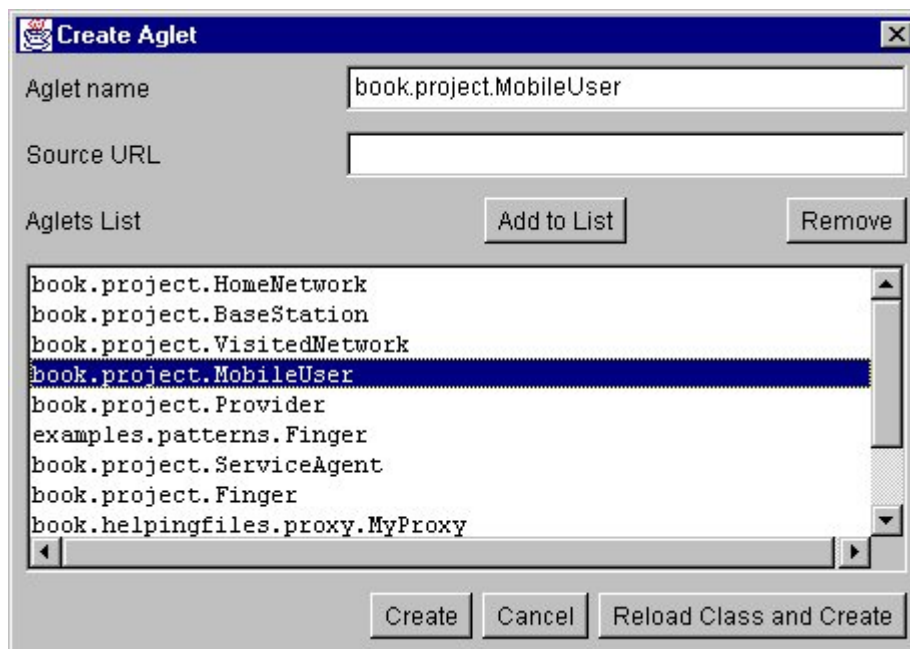
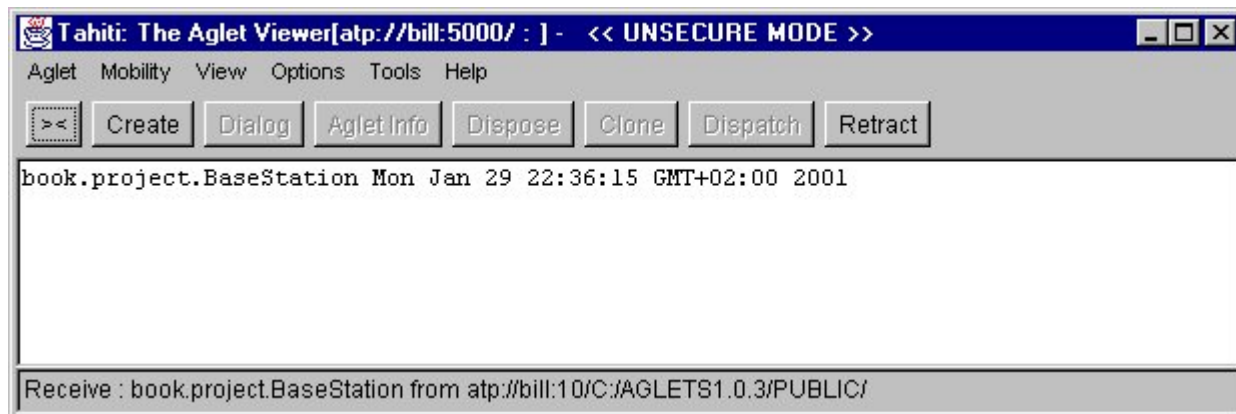
Σχήμα 63: Δημιουργία και αποστολή των BaseStation στους Aglet Servers και το παράθυρο του HomeNetwork μετά την ολοκλήρωση αυτής της διαδικασίας.

Αφού δημιουργηθούν και αποσταλούν τα BaseStations το HomeNetwork δημιουργεί και το Visited Network με ακριβώς την ίδια διαδικασία. Εδώ αποστέλλεται σε άλλον υπολογιστή που ανήκει στο δίκτυο και ονομάζεται elena ενώ ο Aglet Server που τρέχει σε αυτό τον υπολογιστή ονομάζεται elena, με χρησιμοποιούμενη πόρτα την 10. Από την στιγμή που θα φθάσει εκεί ξεκινά την ίδια διαδικασία με το HomeNetwork με την δημιουργία των πέντε BaseStation του. Παρακάτω εμφανίζεται το interface του VisitedNetwork καθώς και ο Aglet server του (Σχήμα 64).



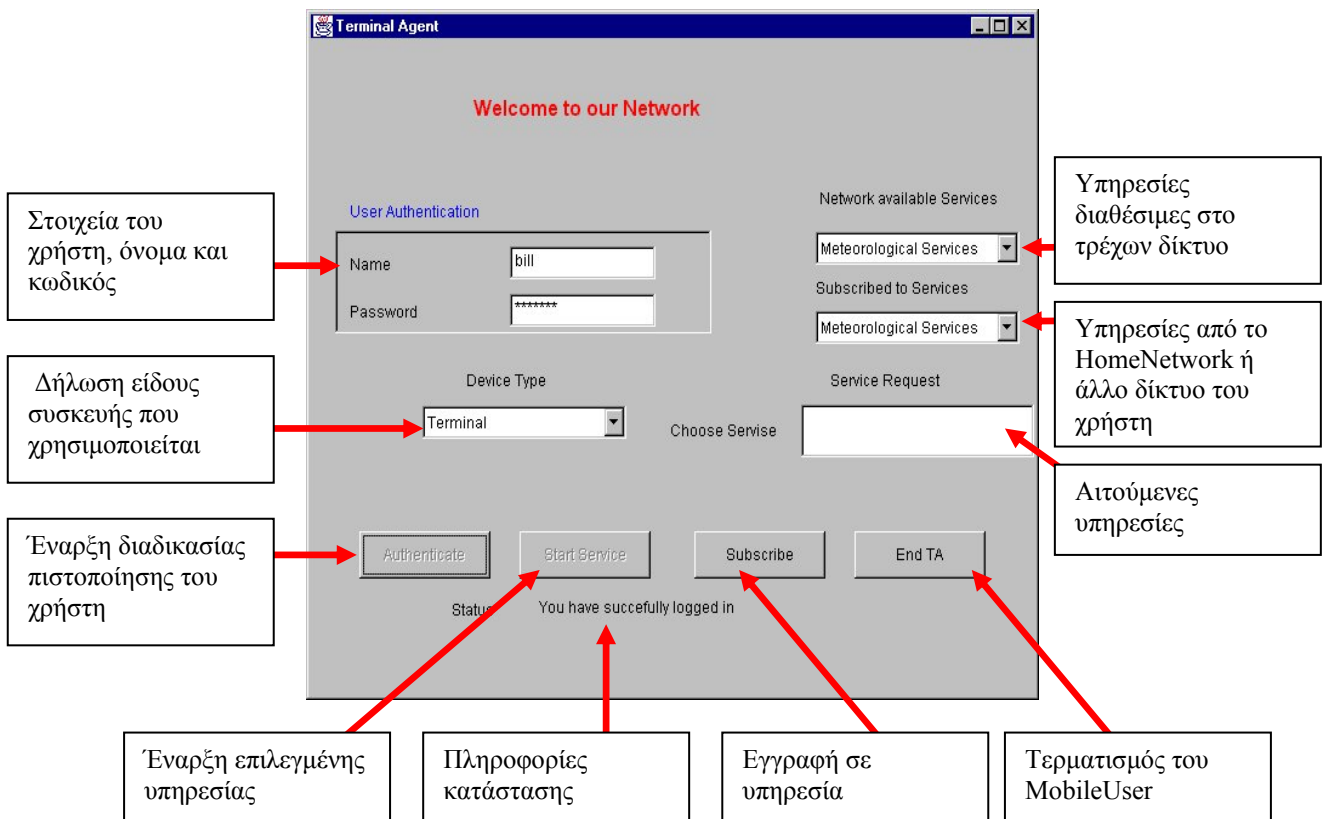
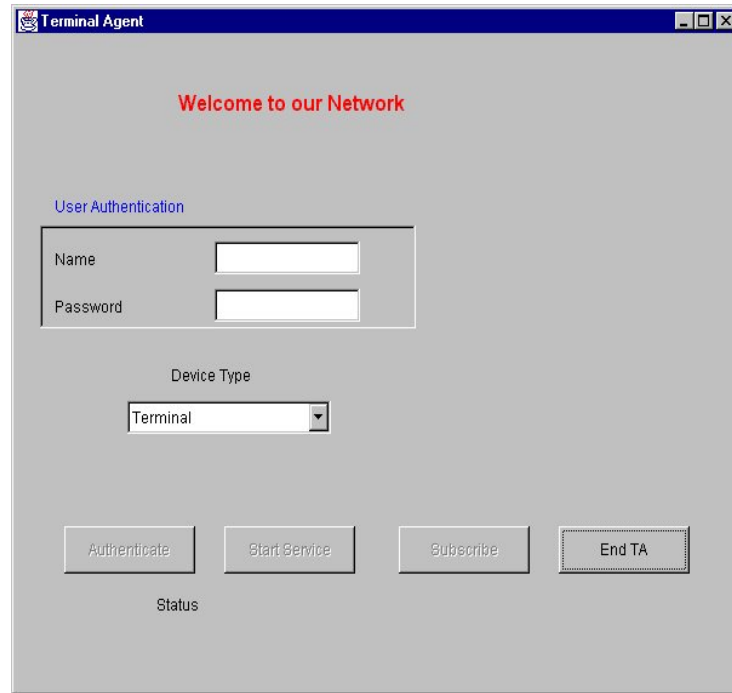
Σχήμα 64. Το interface του VisitedNetwork και ο Aglet Server πάνω στον οποίο τρέχει.

Όταν δημιουργηθούν και όλα τα BaseStation του VisitedNetwork το δίκτυο είναι έτοιμο πλέον να δεχθεί τους διάφορους χρήστες που θα το χρησιμοποιήσουν. Η δημιουργία ενός χρήστη μπορεί να γίνει από οποιοδήποτε Aglet Server του δικτύου πάνω στον οποίο τρέχει ένα Aglet BaseStation. Η δημιουργία γίνεται με την μέθοδο create του Aglet Server όπως φαίνεται στο σχήμα 65.



Σχήμα 65: Δημιουργία του Mobile User σε ένα τυχαίο Aglet Server.

Μετά την δημιουργία του Aglet του Mobile User αμέσως εμφανίζεται το παράθυρο μέσα στο οποίο μας ζητείται να δώσουμε το user id μας καθώς και το password μας κάνουμε login στο δίκτυο. Επίσης μετά την διαδικασία πιστοποίησης του χρήστη εμφανίζονται όλες οι προσφερόμενες υπηρεσίες του δικτύου που βρισκόμαστε καθώς και οι υπηρεσίες που σύμφωνα με το προφίλ του έχει κάνει συνδρομή και μπορεί να χρησιμοποιήσει σε αυτό το δίκτυο. Στο σχήμα 66 εμφανίζεται το παράθυρο του Mobile User πριν και μετά την πιστοποίηση του χρήστη.



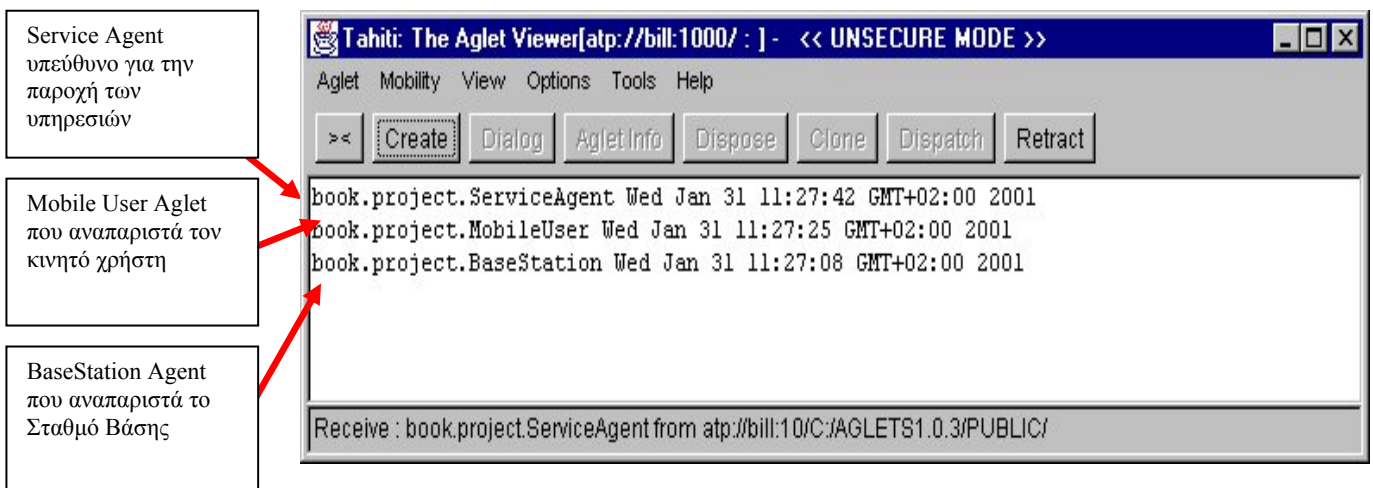
Σχήμα 66. Το interface του Mobile User πριν και μετά το authentication του χρήστη και επεξήγηση των λειτουργιών του.

Από την στιγμή που θα πραγματοποιηθεί η πιστοποίηση του χρήστη, όπως φαίνεται και στο παραπάνω σχήμα, στο παράθυρο του Mobile User εμφανίζονται οι διαθέσιμες υπηρεσίες του δικτύου και λειτουργίες που μπορεί να επιτελέσει ο χρήστης. Κατά την διάρκεια της πιστοποίησης του χρήστη, δηλώνεται και το είδος της χρησιμοποιούμενης συσκευής. Εδώ ο χρήστης έχει τις επιλογές: Τερματικό, Mobile phone και PDA. Σε όλα τα interfaces που χρησιμοποιούνται, από το σημείο αυτό και έπειτα, αναφέρεται το είδος της συσκευής που χρησιμοποιείται από τον χρήστη.

Η ανταλλαγή των μηνυμάτων που απαιτούνται για την πιστοποίηση του χρήστη και την έναρξη παροχής μιας υπηρεσίας αναλύθηκε σε προηγούμενη παράγραφο.

Η αίτηση μιας υπηρεσίας γίνεται με την επιλογή μιας υπηρεσίας από μια από τις δύο λίστες και κατόπιν πατώντας το κουμπί **Start Service**. Αυτό έχει ως αποτέλεσμα την δημιουργία του Service Agent από τον κόμβο που έκανε πιστοποίηση του χρήστη ή από κάποια οντότητα του δικτύου (είτε κάποιο BaseStation είτε από το HomeNetwork ή το VisitedNetwork) που είναι υπεύθυνο για αυτή την υπηρεσία.

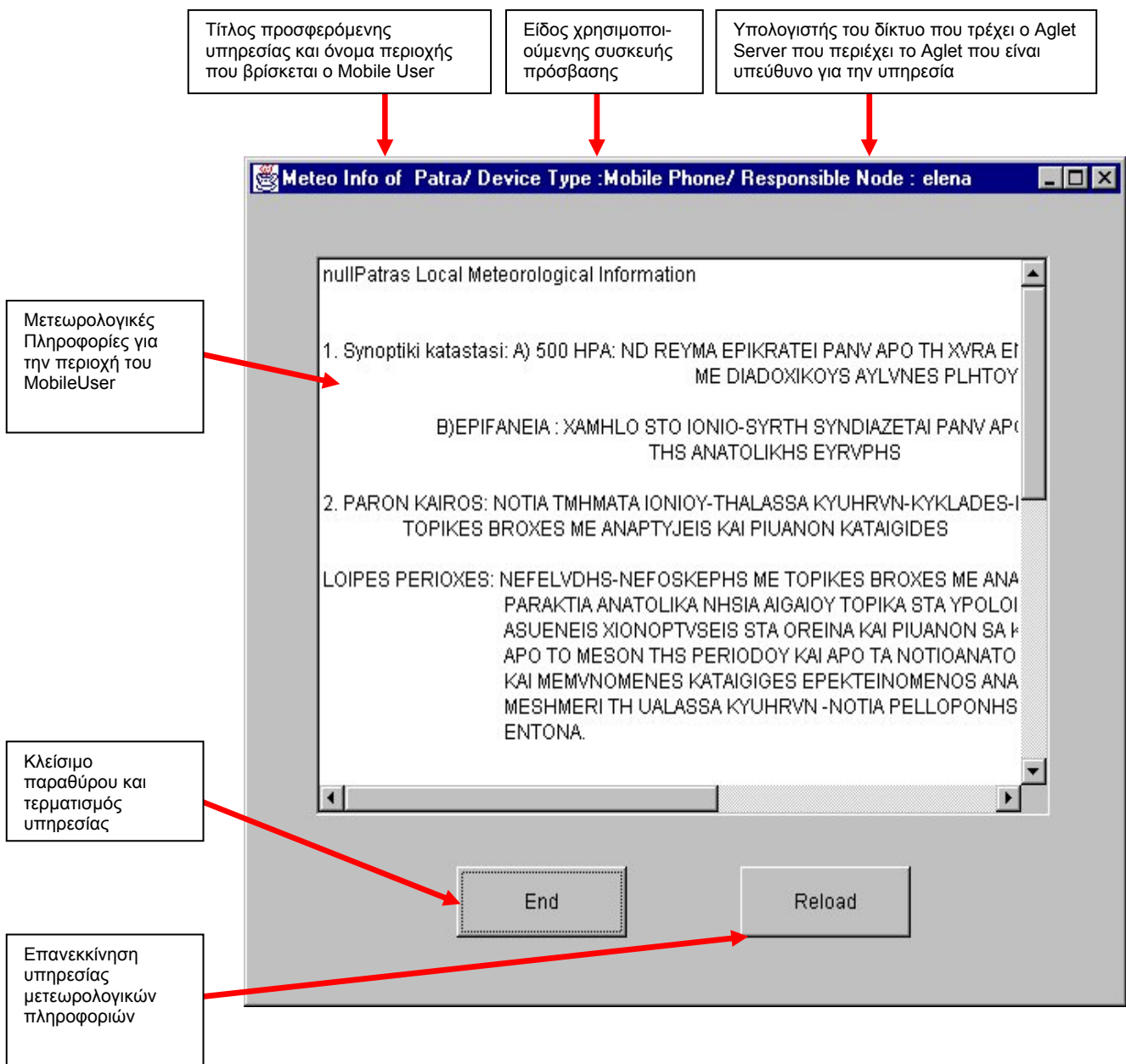
Στο σχήμα 67 φαίνεται ένας Aglet Server με τα Aglet που τρέχουν σε αυτόν αφού ο Mobile User έχει ζητήσει την παροχή της υπηρεσίας Μετεωρολογικών Πληροφοριών.



Σχήμα 67: Ένας Aglet Server μετά την αίτηση του Mobile User για την παροχή της υπηρεσίας Μετεωρολογικών Πληροφοριών

Στην συνέχεια από την στιγμή που θα φθάσει το ServiceAgent στον Aglet Server που βρίσκεται ο Mobile User πατώντας το κουμπί **Start Service** για δεύτερη φορά τότε εμφανίζεται το παράθυρο της αιτηθείσας υπηρεσίας.

Στο σχήμα 68 φαίνεται το παράθυρο της υπηρεσίας παροχής Μετεωρολογικών Πληροφοριών με αναλυτική περιγραφή των παρεχόμενων πληροφοριών. Στην συγκεκριμένη υπηρεσία οι μετεωρολογικές πληροφορίες παρουσιάζονται με την τυπική μορφή που χρησιμοποιείται από την Εθνική Μετεωρολογική Υπηρεσία (EMY) για την περιγραφή και πρόβλεψη του καιρού Ελλάδος και τοπικού καιρού που είναι διάρκειας 24 ωρών.

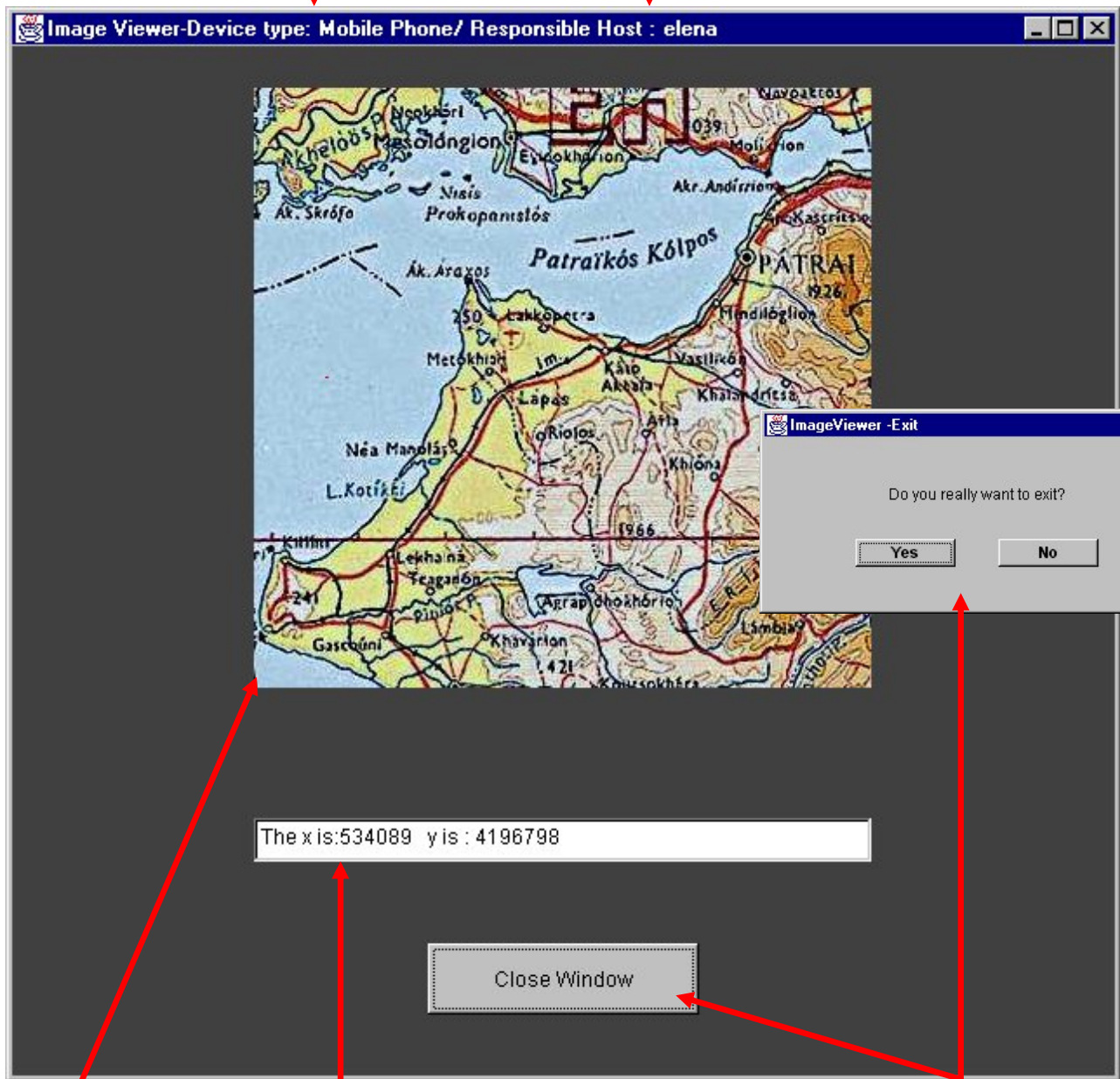


Σχήμα 67: Παράθυρο υπηρεσίας Μετεωρολογικών Πληροφοριών

Με την ίδια ακριβώς διαδικασία ενεργοποιούνται και οι υπόλοιπες υπηρεσίες που υλοποιούνται στην παρούσα εργασία δηλαδή η παροχή χαρτών περιοχής καθώς και η υπηρεσία παροχής των συντεταγμένων της θέσης που βρίσκεται ο κινητός χρήστης. Στο σχήμα 69 εμφανίζεται και αναλύεται η υπηρεσία παροχής χάρτου περιοχής του κινητού χρήστη.

Είδος χρησιμοποιούμενης συσκευής πρόσβασης

Υπολογιστής του δικτύου που τρέχει ο Aglet Server που περιέχει το Aglet που είναι υπεύθυνο για την υπηρεσία



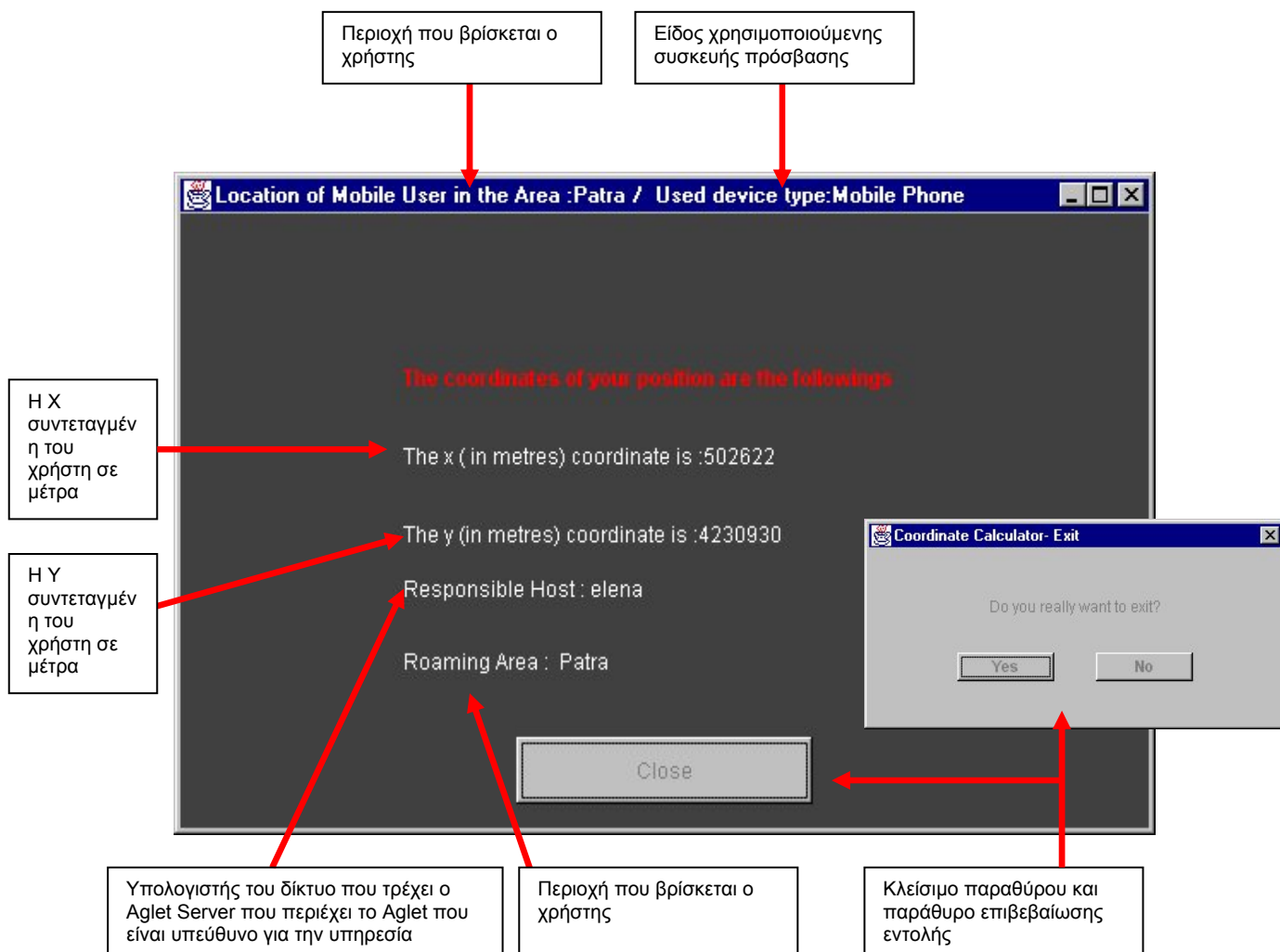
Χάρτης γεωγραφικά προσανατολισμένος της περιοχής που βρίσκεται ο Mobile User.

Καθώς κινούμε το ποντίκι πάνω από την εικόνα στο σημείο αυτό εμφανίζονται οι πραγματικές συντεταγμένες της περιοχής πάνω από την οποία βρίσκεται το ποντίκι. Οι συντεταγμένες αυτές είναι εκφρασμένες σε μέτρα σε προβολή UTM και σφαιροειδές WGS 84

Κλείσιμο παραθύρου και παράθυρο επιβεβαίωσης εντολής

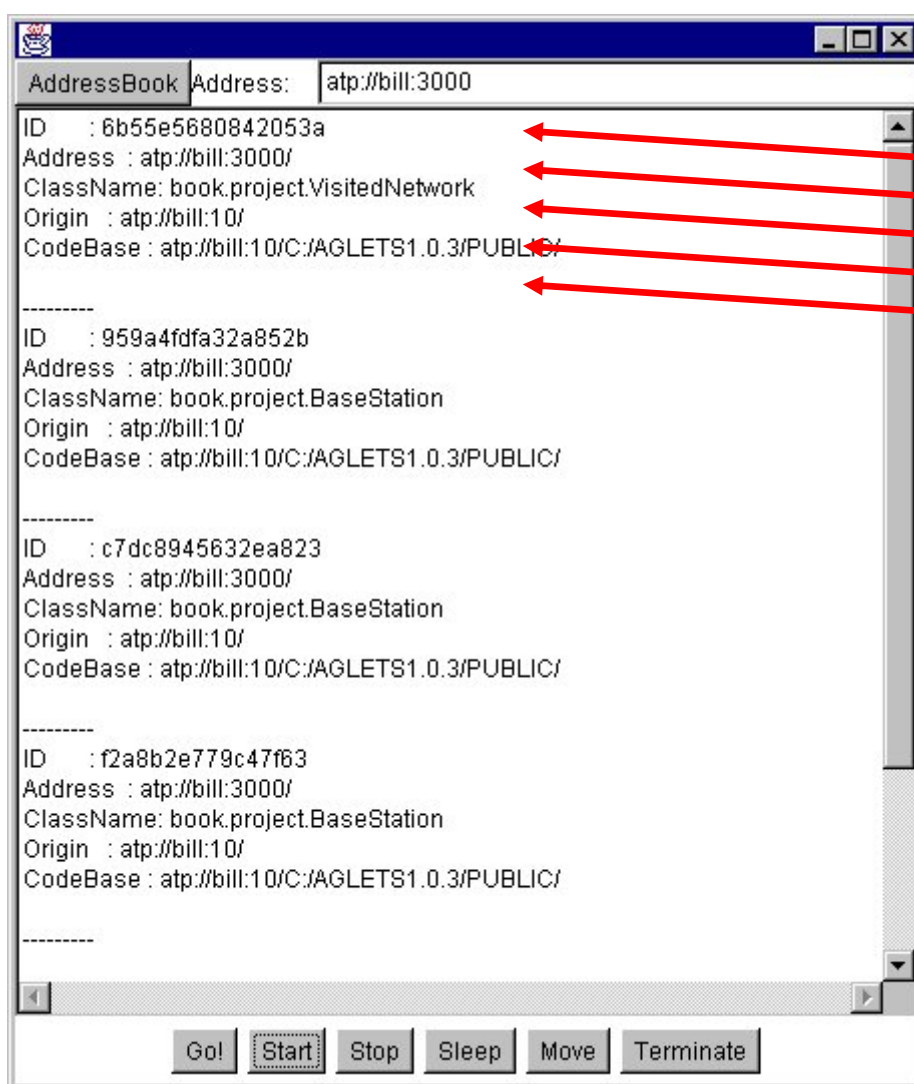
Σχήμα 69: Παράθυρο υπηρεσίας παροχής χαρτών Μετεωρολογικών Πληροφοριών

Παρακάτω εμφανίζεται το παράθυρο της υπηρεσίας παροχής των συντεταγμένων της θέσης που βρίσκεται ο κινητός χρήστης (Σχήμα 70)

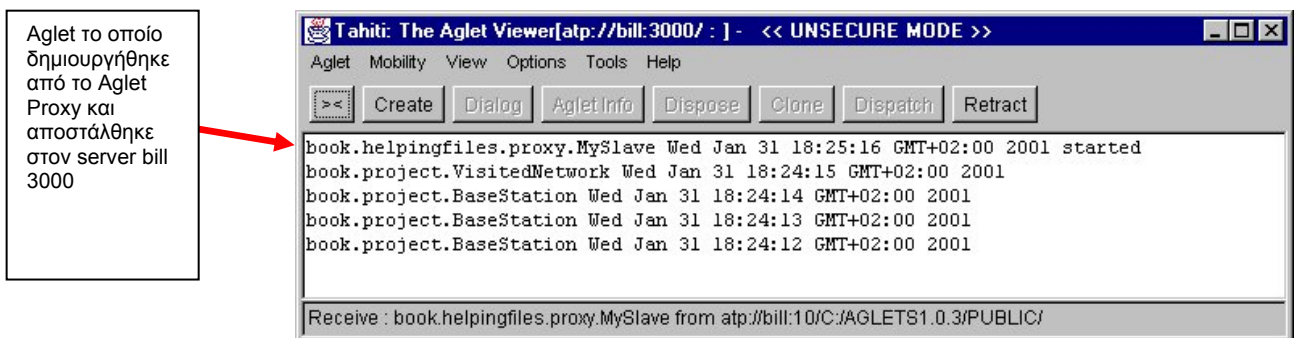
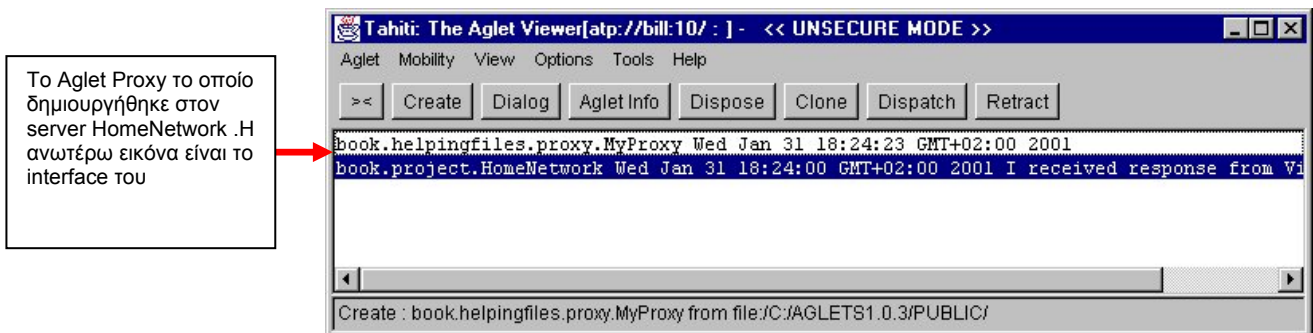
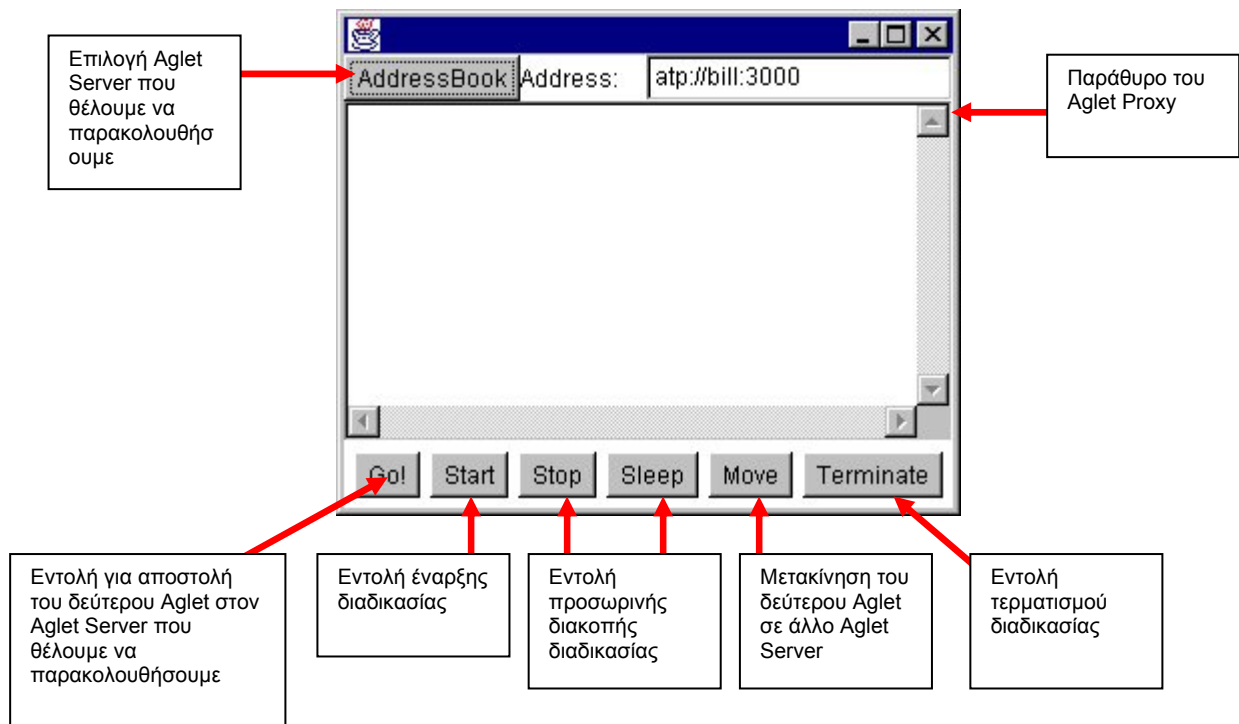


Σχήμα 70: Παράθυρο υπηρεσίας παροχής Πληροφοριών θέσης του κινητού χρήστη.

Τέλος πρέπει να αναφερθεί και η ευκολία που παρέχεται τόσο από το HomeNetwork Aglet όσο και από το VisitedNetwork Aglet σύμφωνα με την οποία, μπορεί να παρακολουθηθεί η κίνηση ενός οποιοδήποτε Aglet Server που είναι ενεργός μέσα στο δίκτυο. Παρέχονται έτσι πληροφορίες για τα Aglets που τρέχουν σε αυτό το Server, το ID τους, ο Aglet Server στον οποίο τρέχουν, το Server που δημιουργήθηκαν και που βρίσκεται ο κώδικας τους. Οι πληροφορίες αυτές παρέχονται ανά τακτά χρονικά διαστήματα ή εάν κάποιο συμβάν προκύψει (π.χ δημιουργία, άφιξη ή καταστροφή ενός Aglet). Η ευκολία αυτή υλοποιείται με την δημιουργία ενός Aglet το οποίο στη συνέχεια δημιουργεί ένα άλλο Aglet το οποίο αποστέλλεται στον Aglet Server που θέλουμε να παρακολουθήσουμε. Το πρώτο Aglet παρέχει και το κατάλληλο interface για την παροχή των πληροφοριών που αναφέρθηκαν καθώς και για τον έλεγχο της ευκολίας αυτής. Στο σχήμα 71 φαίνεται το interface και η λειτουργία του Aglet Proxy



Πληροφορίες που παρέχονται από τη διαδικασία Proxy.
 Το Aglet ID
 Ο Aglet Server
 Ο Aglet Server που δημιουργήθηκε το υπόψη Aglet
 Που βρίσκεται ο κώδικας του Aglet



Σχήμα 71: Η διεπαφή και η λειτουργία του Aglet Proxy.

ΕΠΙΛΟΓΟΣ

Η παροχή "προσωπικών" υπηρεσιών στους χρήστες σε οποιοδήποτε δίκτυο είναι ένα δύσκολο πρόβλημα το οποίο διογκώνεται όταν αυτοί οι χρήστες είναι χρησιμοποιούν διαφορετικά μέσα για να έχουν πρόσβαση (κινητές συσκευές, τερματικά, κλπ) και από διαφορετικά δίκτυα. Η παρούσα διπλωματική εργασία παρουσίασε την έννοια του VHE και των Location dependent services με την βοήθεια των mobile agents. Διαπιστώθηκε ότι η εφαρμογή του VHE μπορεί να υλοποιηθεί με την τεχνολογία των Mobile Agents. Επίσης ο συνδυασμός VHE και Mobile Agents επιτρέπει την εύκολη και ταχύτατη ανάπτυξη νέων υπηρεσιών.

Τα mobile agents είναι οντότητες λογισμικού που χαρακτηρίζονται από την ικανότητα να μετακινούνται με αυτονομία μεταξύ των μηχανών δικτύων και να ολοκληρώνουν σταδιακά την εκτέλεσή τους, οδηγούμενοι από τους στόχους τους. Συνδυάζουν ιδιότητες των software agents, όπως την αντιδραστικότητα, τη συνεργασία, την προσαρμοστικότητα και την επικοινωνιακή δυνατότητα, και επιπροσθέτως, εισάγουν την έννοια του Remote Programming.

Τα κυριότερα πλεονεκτήματα που παρουσιάζουν είναι τα εξής: μειώνουν την κίνηση του δικτύου, μπορούν να αλληλεπιδρούν με οντότητες πραγματικού χρόνου, ενθυλακώνουν πρωτόκολλα, εκτελούνται ασύγχρονα και αυτόνομα, προσαρμόζονται δυναμικά, είναι ανεκτικά στα λάθη και δεν επηρεάζονται από την ετερογένεια των υπολογιστών και των δικτύων. Το θέμα της ασφάλειας, με τους κινδύνους που εγκυμονεί για τους υπολογιστές, τα δίκτυα και τα ίδια τα agents, μπορεί να αντιμετωπιστεί με την εφαρμογή της υπάρχουσας τεχνολογίας ασφάλειας, σε συνδυασμό με τα μοντέλα ασφάλειας των mobile agents.

Η σπουδαιότητα των mobile agents έγκειται στην ευρύτητα των εφαρμογών τους, δίνοντας έναν μοναδικό τρόπο υλοποίησης, ταυτόχρονα, της παραδοσιακής κατανεμημένης επεξεργασίας, του μοντέλου client/server και του μοντέλου κινητού κώδικα. Έτσι, μπορούν να εφαρμοστούν, στην συλλογή κατανεμημένων πληροφοριών, στην ανίχνευση και το φιλτράρισμα πληροφοριών, στην προσωπική βοήθεια και ψυχαγωγία, στην παράλληλη επεξεργασία, στο ηλεκτρονικό εμπόριο, στις τηλεπικοινωνιακές υπηρεσίες, και σε πολλούς επιμέρους τομείς της επιστήμης, της τεχνολογίας, της εργασίας, της εκπαίδευσης και της αγοράς.

Υπάρχουν πολλά mobile agent συστήματα, τα οποία έχουν κατά καιρούς υλοποιηθεί σε διάφορες γλώσσες προγραμματισμού. Η Java προσφέρει agent χαρακτηριστικά που την καθιστούν κυρίαρχη γλώσσα για την υλοποίηση των συστημάτων αυτών. Τα aglets είναι Java mobile agents, που εκμεταλλεύονται τα χαρακτηριστικά της γλώσσας και επιπλέον, με το μοντέλο τους, ξεπερνούν τις ελλείψεις της. Ως τα κατ' εξοχήν network mobile objects, αποτελούν το μέλλον των κατανεμημένων υπολογισμών και αναμένεται να κυριαρχήσουν στα intranets και Internet.

Το VHE είναι ένας από τους επιθυμητούς στόχους, για τα σύγχρονα τηλεπικοινωνιακά συστήματα, που υπόσχονται την ενσωμάτωση mobile multimedia δυνατοτήτων. Η τεχνολογία των aglets και γενικότερα, των mobile agents, μπορεί να εγγραφεί την υλοποίησή του, ενισχύοντας τόσο την πλευρά των κινητών τερματικών όσο και την πλευρά των δικτύων. Η τεχνολογία των Mobile Agents έχει ορισμένες ιδιότητες που μπορούν να εκμεταλλευθούν για την υλοποίηση του VHE. Οι σημαντικότερες από αυτές είναι ότι τα Mobile Agents μπορούν να εφαρμοστούν και να εκτελεστούν σε οποιαδήποτε πλατφόρμα και έτσι παρέχουν την διαλειτουργικότητα που απαιτείται για την επίτευξη του VHE καθώς επίσης ότι τα Mobile Agents επιτρέπουν την μεταφορά κώδικα που είναι απαραίτητη για την προσφορά υπηρεσιών του VHE.

Στην παρούσα εργασία παρουσιάστηκε η έννοια του VHE καθώς και υπηρεσίες που εξαρτώνται από την θέση που βρίσκεται ο κινητός χρήστης (Location Dependent Services). Πρέπει να σημειωθεί ότι οι υπηρεσίες του VHE όπως οι βασικές υπηρεσίες, οι συμπληρωματικές υπηρεσίες και οι value added υπηρεσίες μπορούν να αναπτυχθούν και να εφαρμοσθούν χρησιμοποιώντας την τεχνολογία Mobile Agents και την αρχιτεκτονική που χρησιμοποιήθηκε στην παρούσα εργασία.

BIBΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

1. Markus Breust, Lars Hagen, Thomas Magedanz, 'Impacts of Mobile Agent Technology on Communications System Evolution', IKV++ GmbH Technical University of Berlin, From IEEE 8/1998
2. Thomas Magedanz, K. Rothermel, S. Krause, 'Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?', INFOCOM 3/96
3. Network Functional Model for IMT-2000, Study report /Telecommunication Standardization Sector, ITU 6/1998
4. Jens Hartmann, Garmelita Gorg, Peyman Farjami, 'Agent Technology for the UMTS VHE concept', WOWMOM 98, Dallas USA
5. UMTS Network Aspects, Project P920, EURESCOM
6. Shaw Green, Leon Hurst, Brenda Nagle, Padraig Cunningham, Fergal Somers, Richard Evans, 'Software Aspects: A review', 27 May 1997,
7. Markus Breugst and Thomas Magedanz, 'Mobile Agents- Enabling Technology for Active Intelligent Network Implementation', IKV++ GmbH Technical University of Berlin, IEEE 5-6/1998
8. T. Magedanz, R. Popescu-Zeletin, 'Towards "Intelligence on Demand" – On the Impacts of Intelligent Agents on IN', Conference 11/1996
9. Services and Service Capabilities, ETSI, UMTS, UMTS 22.05
10. Services aspects and Service principles, ETSI, UMTS, UMTS 22.01
11. Terminal and smart card concepts, ETSI, UMTS, UMTS 22.07
12. M. Breugst, I. Busse, S. Covaci, T. Magedanz, 'Grasshopper: A Mobile Agent Platform for IN Based Service Environments', IEEE IN Workshop, Bordeaux, France, May 10-13/1998
13. Services aspects Virtual Home Environment (VHE), ETSI, UMTS, UMTS 22.70
14. VHE Support in the (ISDN Evolved) UMTS Core Network, ETSI
15. UMTS Network Aspects: Service Scenario for the VHE concept, EURESCOM, PIR.21
16. UMTS Network Aspects Service Scenario for the VHE concept, EURESCOM, ANNEX PIR.21
17. UMTS Network Aspects Architecture, Protocol analysis and Requirements for the VHE model, EURESCOM
18. L. Hagen, J. Mauersberger, C. Weckerle, 'Mobile agent based service subscription and customisation using the UMTS VHE', *Computer Communications*, 23,
19. Goran Swedberg, 'Ericsson's mobile location solution', Ericsson
20. Yariv Aridor, Danny B. Lange, 'Agent Design Patterns: Elements of Agent Application Design'
21. T. Magedanz, A. Karmouch, 'Mobile Software Agents for Telecommunication Applications', *Computer Communications*, 23, 2000, pp.705-707
22. A. Puliafito, O. Tomarchio, 'Using mobile agents to implement flexible network management strategies', *Computer Communications*, 23, 2000, pp.708-719
23. D.Gavalas, D.Greenwood, M.Granbari, M. O'Mahony, 'Advanced network monitoring applications based on mobile /intelligent agent technology', *Computer Communications*, 23, 2000, pp.720-730
24. F. Chatzipapadopoulos, G. De Zen, T. Magedanz, I.S Venieris, F. Zizza, 'Harmonised Internet and PSTN service provisioning', *Computer Communications*, 23, 2000, pp.731-739
25. J. Tang, T. White, B. Pagurek, R. Glitho, 'Advanced service architecture for H.323 Internet Protocol Telephony', *Computer Communications*, 23, 2000, pp.740-753
26. P. Farjami, C. Gorg, F. Bell, 'Advanced service provisioning based on mobile agents', *Computer Communications*, 23, 2000, pp.754-760

27. N. Kawaguchi, K. Toyama, Y. Inagaki, 'Magnet: ad hoc network system based on mobile agents', *Computer Communications*, 23, 2000, pp.761-768
28. L.M Silva, G. Soares, P. Martins, V. Batista, L. Santos, 'Comparing the performance of mobile agents systems: a study of benchmarking', *Computer Communications*, 23, 2000, pp.769-778
29. Gunter Karjoth, Danny B. Lange , Mitsuru Oshima, 'A security model for aglets', IEEE Internet Computing, 1997
30. Danny B. Lange, 'Mobile Objects and Mobile Agents: The Future of Distributed Computing?'
31. Danny B. Lange, Mitsuru Oshima, 'Mobile Agents with Java: The Aglet API'
32. Mario Magnanelli, Moira Norrie, 'Databases for Agents and agents for Databases'
33. Gregor Joeris, 'Decentralised and flexible Workflow Enactment based on Task Co-ordination Agents',
34. Scott. A Deloach, 'Multiagent Systems Engineering: A methodology and Language for Designing Agent Systems'
35. Robert Gray, David Kotz, George Cybenco, Daniela Rus, 'Mobile agents: Motivations and State of the art systems'
36. Gal. Kaminka, David V. Pynadath, Milind Tambe, 'A Fly on the wall: Monitoring Agent Organisations by Eavesdropping', *American Association for Artificial Intelligence*, 2000
37. Mei Jie, Kamal Karlapalem, Frederick Lochovsky, 'A multi-agent Framework for expertise Location'
38. Tsvi Kuflik, Peretz Shoval, 'User Profile Generation for Intelligent Information Agents Research in Progress',
39. Esther Lee, Matthew Lih, Morris Brill, 'Agent Based Satellite Ground Systems'
40. Stefan Pleisch, 'State of the Art of Mobile Agent Computing - Security, Fault Tolerance and Transaction Support', IBM Research Laboratory, Zurich, 1999
41. T. Papaioannou, J. Edwards, 'Using Mobile Agents to improve the Alignment Between Manufacturing and its Support Systems', Leicestershire, 1998
42. St. Papastavrou, George Samaras, Evaggelia Pitoura, 'Mobile Agents for WWW Distributing Database Access'
43. Charles Petrie, 'Agent Based Software Engineering'
44. Achiro Lida, Takashi Nishigaya , Koso Murakami, 'DUET: An Agent Based Personal Communications Network'
45. Milind Tambe, David V. Pynadath, Nicolas Chauvat, 'Building Dynamic Agent Organisations in Cyberspace'
46. Heecheol Jeon, Charles Petrie, Mark Cutkosky, 'JATLite: A java Agent Infrastructure with Message Routing'
47. Antonella Di Stefano Corrado Santoro, NetChaser, 'Agent Support for Personal Mobility'
48. Lassaad Gannoun, Lulien Francioli, Stanislav Chachkov, Frederic Schutz, Jarle G. Hulaas, Jurgen Harms, 'Domain Name eXchange: A mobile Agent Based shared registry System', University of Geneva
49. Αλγόριθμος υπολογισμού θέσης του GPS, <http://ralph.bucher.home.att.net/project.html>

ΠΑΡΑΡΤΗΜΑ
