

**Εθνικό και Καποδιστριακό  
Πανεπιστήμιο Αθηνών  
Σχολή Θετικών Επιστημών  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Το Μοντέλο Αυτόματης Προώθησης στις  
Εξαρτώμενες από τη Θέση Υπηρεσίες**

Βασίλειος Τσέτσος  
Α.Μ: 1309

Οδυσσέας Σέκκας  
ΑΜ:1274

**Επιβλέπων Καθηγητής: Λάζαρος Μεράκος**

-Αθήνα-  
Ιούνιος 2003



# ΠΡΟΛΟΓΟΣ

Η παρούσα προπτυχιακή εργασία εκπονήθηκε στο πλαίσιο του Προπτυχιακού Προγράμματος του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Το αντικείμενο μελέτης της είναι οι Εξαρτώμενες από την Θέση Υπηρεσίες σε ασύρματα δίκτυα, δίνοντας ειδικότερη έμφαση στο μοντέλο Αυτόματης Προώθησης (Push). Η έμφαση αυτή έχει την μορφή παρατηρήσεων και επισημάνσεων, που αφορούν στο συγκεκριμένο μοντέλο, σε κάθε επιμέρους θέμα των υπηρεσιών που μελετούνται. Η μελέτη για το μοντέλο αυτό συμπληρώνεται με την δημιουργία ενός παραδειγματικού χρονοδρομολογητή (scheduler) ο οποίος θα μπορούσε να αποτελεί συστατικό μέρος μιας πλατφόρμας παροχής τέτοιων υπηρεσιών.

Το έναυσμα για τη μελέτη του συγκεκριμένου θέματος προήλθε από την παρακολούθηση των μαθημάτων της κατεύθυνσης Τηλεπικοινωνιών και ειδικότερα των μαθημάτων που αναφέρονται στα δίκτυα υπολογιστών.

Θα θέλαμε να ευχαριστήσουμε θερμά τον υπεύθυνο Καθηγητή του τομέα Επικοινωνιών και Επεξεργασίας Σήματος, κ. Λάζαρο Μεράκο, για την υποστήριξη που μας παρείχε όταν αυτή χρειάστηκε. Επίσης ευχαριστούμε τον ερευνητή μηχανικό του εργαστηρίου δικτύων επικοινωνιών (CNL), κ. Ιωάννη Πρίγγουρη, για την ουσιαστική και καθοριστική συνδρομή του κατά την υλοποίηση του προγραμματιστικού μέρους της εργασίας. Χωρίς αυτήν, η δημιουργία του χρονοδρομολογητή θα ήταν πολύ πιο δύσκολη. Τέλος θα θέλαμε να ευχαριστήσουμε και τον Δρ. Ευστάθιο Χατζηευθυμιάδη, για την καθοδήγηση και την πολύτιμη συνεισφορά του σε όλη τη διάρκεια εκπόνησης της εργασίας. Η αδιάκοπη παρακολούθηση της προόδου της πτυχιακής εργασίας και οι εύστοχες επισημάνσεις του συντέλεσαν, σε μεγάλο βαθμό, στη διαμόρφωση του τελικού αποτελέσματος.

Αθήνα, Ιούνιος 2003

Βασίλειος Τσέτσος

Οδυσσέας Σέκκας



# Περιεχόμενα

<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>5</b>
<b>ΜΕΡΟΣ 1 - ΕΞΑΡΤΩΜΕΝΕΣ ΑΠΟ ΤΗ ΘΕΣΗ ΥΠΗΡΕΣΙΕΣ</b> .....	<b>6</b>
1.1 ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ .....	7
1.1.1 Βασικοί Ορισμοί.....	7
1.1.2 Ιστορία .....	7
1.1.3 Αυτόματη Προώθηση.....	9
1.1.4 Παραδείγματα Υπηρεσιών Θέσης .....	11
1.1.5 Γενική Αρχιτεκτονική .....	14
1.1.5.1 Οντότητες Συστήματος .....	14
1.1.5.2 Αρχιτεκτονική σε επίπεδο υπηρεσιών και λογισμικού.....	17
1.1.5.3 Αρχιτεκτονική στο GSM.....	18
1.2 ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΠΡΟΤΥΠΑ .....	20
1.2.1 Εντοπισμός Θέσης.....	20
1.2.1.1 Εντοπισμός θέσης σε GSM δίκτυα .....	20
1.2.1.1.1 Τεχνικές βασισμένες στο δίκτυο (network-based techniques) .....	20
α) Cell ID .....	20
β) Cell ID + Timing Advance (TA).....	21
γ) Uplink Time Of Arrival (UL-TOA).....	22
δ) Angle Of Arrival (AOA).....	23
1.2.1.1.2 Τεχνικές βασισμένες στη κινητή συσκευή (terminal-based techniques).....	24
α) Enhanced Observed Time Difference (E-OTD).....	24
β) Assisted GPS (A-GPS).....	25
1.2.1.2 Εντοπισμός θέσης σε ασύρματα τοπικά δίκτυα .....	26
1.2.1.2.1 RSS-based Location Fingerprinting.....	27
1.2.1.3 Σχόλια στις τεχνικές εντοπισμού θέσης .....	28
1.2.2 Το Περιεχόμενο στις Εξαρτώμενες από τη Θέση Υπηρεσίες .....	30
1.2.2.1 Γεωκωδικοποίηση.....	30
1.2.2.2 Γεωγραφικά Συστήματα Πληροφοριών (GIS).....	31
1.2.2.2.1 Εισαγωγή Στοιχείων .....	32
1.2.2.2.2 Δομή Βάσης Δεδομένων.....	32
1.2.2.2.3 Ανάλυση Χώρου.....	34
1.2.2.2.4 Επερωτήσεις και Αναζήτηση Δεδομένων.....	35
1.2.2.2.5 Παρουσίαση Αποτελεσμάτων Αναζήτησης.....	36
1.2.2.3 Τρόποι Αναπαράστασης Περιεχομένου .....	36
1.2.2.3.1 Χάρτες και Χαρτογραφική Διαδικασία.....	38
1.2.2.3.2 Οι Γλώσσες GML και SVG.....	41
1.2.2.3.3 Φωνητικές Υπηρεσίες.....	42
1.2.2.3.4 Τρόποι Αναπαράστασης Περιεχομένου και Εξαρτώμενες από τη Θέση Υπηρεσίες .....	43
1.2.2.4 Χωρικές Βάσεις Δεδομένων .....	45
1.2.3 Διεπαφές.....	46
1.2.3.1 SMS .....	47
1.2.3.1.1 Γενικά .....	47
1.2.3.1.2 Αρχιτεκτονική.....	48
1.2.3.1.3 Το πρωτόκολλο SMPP (Short Message Peer to Peer protocol) .....	50
1.2.3.1.4 Εφαρμογές SMS .....	52
1.2.3.2 WAP .....	53
1.2.3.2.1 Γενικά .....	53
1.2.3.2.2 Αρχιτεκτονική.....	54
1.2.3.2.3 WAP Push .....	55
α) Push Access Protocol (PAP) .....	56
β) Push Proxy Gateway (PPG) .....	57
γ) Push Over-The-Air Protocol.....	58
1.2.3.3 Σύγκριση SMS-WAP .....	58

1.2.3.4	Νέες τεχνολογίες διεπαφών (VoiceXML).....	59
1.2.4	1.2.4 Η Τυποποίηση στις Υπηρεσίες Θέσης.....	60
1.2.4.1	Η σημασία των Ανοικτών Προτύπων.....	60
2.2.4.2	Οι Οργανισμοί Τυποποίησης.....	62
1.2.4.3	Mobile Location Protocol.....	63
1.3	ΕΙΔΙΚΑ ΘΕΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ.....	65
1.3.1	1.3.1 Επιχειρηματικά Θέματα.....	65
1.3.1.1	Επιχειρηματικοί ρόλοι.....	65
1.3.1.2	Επιχειρηματικά Μοντέλα.....	68
1.3.2	1.3.2 Χρέωση.....	71
1.3.2.1	Οικονομικά θέματα.....	72
1.3.2.1.1	Οι νέες απαιτήσεις.....	72
1.3.2.1.2	Ροές Χρέωσης.....	73
1.3.2.1.3	Είδη χρεώσεων.....	74
1.3.2.1.4	Χρέωση του τελικού χρήστη.....	75
1.3.2.2	Τεχνικά θέματα.....	76
1.3.3	1.3.3 Ιδιωτικότητα.....	77
1.3.3.1	Θέματα και Προβλήματα που ανακύπτουν.....	78
1.3.3.2	Ειδικές Περιπτώσεις.....	80
1.3.3.3	Μέτρα προστασίας της ιδιωτικότητας.....	80
1.4	ΠΛΑΤΦΟΡΜΕΣ ΓΙΑ ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ.....	84
1.4.1	LocatioNet Platform [40].....	84
1.4.2	Cellebrity Platform [41].....	86
1.4.3	LoL@ Platform [42].....	86
1.4.4	Cellpoint Platform [43].....	87
1.4.5	IBM's WebSphere Everyplace Server [44].....	89
1.4.6	Webraska SmartZone Geospatial Platform [45].....	90
1.4.7	PoLoS [46].....	91
1.5	ΤΟ ΠΑΡΟΝ ΚΑΙ ΤΟ ΜΕΛΛΟΝ ΤΩΝ ΥΠΗΡΕΣΙΩΝ ΘΕΣΗΣ.....	94
1.5.1	Σημερινά προβλήματα και πιθανές λύσεις.....	94
1.5.2	Προβλέψεις για το Μέλλον.....	96
<b>ΜΕΡΟΣ 2 - ΠΑΡΑΔΕΙΓΜΑΤΙΚΟΣ ΧΡΟΝΟΔΡΟΜΟΛΟΓΗΤΗΣ.....</b>		<b>97</b>
2.1	ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ.....	98
2.1.1	Εξυπηρετές Εφαρμογών (Application Servers).....	99
2.1.2	Enterprise JavaBeans (EJB).....	100
2.1.2.1	Γενικά στοιχεία.....	100
2.1.2.2	Οι τύποι των EJBs.....	102
2.1.3	HTTP Servlets.....	103
2.1.4	Η κλάση Timer της Java.....	104
2.1.5	Διαχείριση με JMX.....	106
2.1.5.1	JMX (Java Management Extensions).....	106
2.1.5.2	MBeans (Managed Beans).....	107
2.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	109
2.2.1	Διάγραμμα Κλάσεων (Class Diagram).....	110
2.2.2	Ακολουθιακά Διαγράμματα (Sequence Diagrams).....	110
2.2.2.1	Η μέθοδος initialize.....	111
2.2.2.2	Η μέθοδος schedule (χρονικός σκανδαλισμός).....	111
2.2.2.3	Η μέθοδος schedule (χωρικός σκανδαλισμός).....	112
2.2.2.4	Η μέθοδος remove.....	113
2.2.2.5	Η χρονοδρομολόγηση μιας διεργασίας.....	113
2.2.3	Διάγραμμα συστατικών (Component Diagram).....	114
<b>ΑΝΑΦΟΡΕΣ - ΠΗΓΕΣ.....</b>		<b>116</b>
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....</b>		<b>119</b>
<b>ΠΑΡΑΡΤΗΜΑ - Ο ΚΩΔΙΚΑΣ ΤΟΥ ΠΑΡΑΔΕΙΓΜΑΤΙΚΟΥ ΧΡΟΝΟΔΡΟΜΟΛΟΓΗΤΗ.....</b>		<b>121</b>

# Εισαγωγή

Τα τελευταία χρόνια συντελούνται πολύ μεγάλες αλλαγές στις ασύρματες επικοινωνίες. Η διεύρυσή τους στις καταναλωτικές αγορές είναι πολύ μεγάλη, ενώ δεν απέχουμε πολύ και από τις επόμενες γενιές ασύρματων δικτύων που υπόσχονται να μεταφέρουν το περιβάλλον του Internet στο κινητό τερματικό μας. Μέσα σε αυτό το πλαίσιο άρχισαν να αναπτύσσονται νέες υπηρεσίες για κινητούς χρήστες, οι οποίες προσανατολίζονται κυρίως στην παροχή δεδομένων. Οι πιο εξελιγμένες και πιο πολλά υποσχόμενες από αυτές είναι οι Εξαρτώμενες από τη Θέση Υπηρεσίες. Σε αυτή την πτυχιακή εργασία γίνεται μια προσπάθεια να μελετηθούν τα σημαντικότερα επιμέρους θέματα που συνιστούν τις υπηρεσίες αυτές. Ιδιαίτερη βαρύτητα δίνεται σε ότι αφορά το μοντέλο Αυτόματης Προώθησης (πώς υλοποιείται, τι ιδιαιτερότητες έχει, τι προβλήματα εισάγει κλπ).

Έτσι στην πρώτη ενότητα του πρώτου μέρους παρουσιάζονται μερικά γενικά και εισαγωγικά στοιχεία για τις υπηρεσίες αυτές καθώς και η γενική αρχιτεκτονικά τους (ενότητα 1.1). Στην συνέχεια (ενότητα 1.2) παρουσιάζονται οι κυριότερες τεχνολογίες και πρότυπα που χρησιμοποιούνται για τον εντοπισμό της θέσης του κινητού τερματικού, την διαχείριση και την παρουσίαση του περιεχομένου και τις πιο συνηθισμένες διαπαφές προς τον τελικό χρήστη (SMS, WAP). Επίσης, στην ίδια ενότητα αναφέρεται συνοπτικά και η σημερινή κατάσταση στον χώρο της τυποποίησης. Στην ενότητα 1.3 θίγονται θέματα που σχετίζονται με τα επιχειρηματικά μοντέλα που μπορούν να εφαρμοστούν στις υπηρεσίες θέσης, με τις πολιτικές χρέωσης και με το πολύ σοβαρό ζήτημα της ιδιωτικότητας. Τέλος, στην ενότητα 1.4 παρουσιάζονται οι πιο γνωστές εμπορικές πλατφόρμες που μπορούν να υποστηρίξουν υπηρεσίες θέσης.

Στο δεύτερο μέρος βρίσκεται ό,τι σχετίζεται με τον παραδειγματικό χρονοδρομολογητή που υλοποιήθηκε. Πιο συγκεκριμένα, αναφέρονται κάποια βασικά στοιχεία για τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή του (ενότητα 2.1) και η αρχιτεκτονική της σχεδίασής του με διαγράμματα UML (ενότητα 2.2). Ο κώδικάς του βρίσκεται στο παράρτημα που υπάρχει στο τέλος της εργασίας.

Η εργασία ολοκληρώνεται με τις αναφορές που έγιναν στο κείμενο και με τις κυριότερες πηγές που χρησιμοποιήθηκαν προκειμένου να βρεθεί το υλικό για το κείμενο. Όπου είναι δυνατό γίνονται αναφορές και σε δικτυακούς τόπους. Τέλος, παρέχονται και οι πλήρεις ονομασίες των κυριότερων συντομογραφιών που χρησιμοποιήθηκαν στην παρούσα εργασία.

## **Μέρος 1**

### **Εξαρτώμενες από τη Θέση Υπηρεσίες**



## 1.1 Γενικά Στοιχεία

### 1.1.1 Βασικοί Ορισμοί

Οι Εξαρτώμενες από τη Θέση Υπηρεσίες (Location Based Services) μπορούν να περιγραφούν σαν «εφαρμογές» που εκτελούνται και παρέχουν κάποια λειτουργικότητα στον κινητό χρήστη (εξ' ου και ο όρος «υπηρεσίες»), μετά από κατάλληλο γεωγραφικό σκανδαλισμό. Δηλαδή η εκτέλεσή τους ξεκινάει μετά από συγκεκριμένες αλλαγές της θέσης του χρήστη. Οι Εξαρτώμενες από την Θέση Υπηρεσίες θα αναφέρονται στο υπόλοιπο κείμενο και σαν και σαν «υπηρεσίες θέσης» ή «LBS υπηρεσίες» (ο όρος LBS είναι συντομογραφία του αγγλικού όρου «Location Based Services»)

Στο κείμενο θα χρησιμοποιηθούν και οι παρακάτω όροι που εξηγούνται εδώ:

- Με τον όρο «περιεχόμενο» (content) θα εννοούνται τα δεδομένα που χρησιμοποιούνται για τις ανάγκες των υπηρεσιών θέσης (γεωγραφικές ή άλλες πληροφορίες).
- Με τον όρο «χρήστης» θα αναφέρεται ο τελικός χρήστης ενός ασύρματου δικτύου. Ο όρος εστιάζεται κυρίως σε χρήστες κινητών κυψελωτών δικτύων, επειδή σε αυτά κυρίως αναπτύσσονται οι υπηρεσίες θέσης και επειδή αυτά είναι τα πιο διαδεδομένα ασύρματα δίκτυα. Επίσης ο όρος «συνδρομητής» αναφέρεται κυρίως σε όσους χρήστες έχουν εγγραφεί σε μια υπηρεσία.
- Ο όρος «παίκτης» αναφέρεται σε μία εταιρία ή άλλο φορέα που συμμετέχει στην παροχή υπηρεσιών. Είναι δηλαδή ουσιαστικά το στιγμιότυπο ενός επιχειρηματικού ρόλου. Για παράδειγμα σαν «παίκτης» αναφέρονται τόσο ο παροχέας του ασύρματου δικτύου όσο και ο παροχέας της υπηρεσίας.

Τέλος θα πρέπει να σημειωθεί ότι η παρούσα μελέτη των LBS υπηρεσιών αναφέρεται κυρίως σε κυψελωτά δίκτυα. Αυτό δεν σημαίνει ότι δεν μπορούν αυτές να εφαρμοστούν σε άλλους τύπους δικτύων που υποστηρίζουν κινητικότητα (mobility) όπως για παράδειγμα στα ασύρματα τοπικά δίκτυα (WLANs). Αυτά τα τελευταία δίκτυα όμως, δεν αφορούν μεγάλο αριθμό χρηστών και γενικά είναι περιορισμένης χρησιμότητας (όχι σε ποιότητα αλλά σε ποσότητα). Εξάλλου η τοπική φύση αυτών των δικτύων, που συνεπάγεται περιορισμένη κινητικότητα, δεν επιτρέπει την δημιουργία πολύ καινοτόμων και χρήσιμων υπηρεσιών.

### 1.1.2 Ιστορία

Το πρώτο ενδιαφέρον για τις υπηρεσίες θέσης εμφανίστηκε λίγο μετά την εγκατάσταση και αποδοχή του GSM (δηλαδή στα μέσα της δεκαετίας του 1990). Οι

παροχείς τηλεπικοινωνιών βλέποντας ότι η διείσδυση της κινητής τηλεφωνίας θα έφθανε σε κορεσμό άρχισαν να στρέφονται σε άλλους τρόπους αύξησης των εσόδων τους (Average Revenue Per User, ARPU). Ο πιο κατάλληλος τρόπος ήταν βέβαια οι υπηρεσίες προστιθέμενης αξίας, στις οποίες συγκαταλέγονται και οι υπηρεσίες θέσης.

Το πρώτο βήμα για την ανάπτυξή τους έγινε το 1996 όταν η Ομοσπονδιακή Επιτροπή Επικοινωνιών (Federal Communications Commission, FCC) των ΗΠΑ υποχρέωσε τους παροχείς τηλεπικοινωνιών να παρέχουν στις αρμόδιες αρχές (αστυνομία, πυροσβεστική, νοσοκομεία) πληροφορία για την θέση αυτού που έκανε μια κλήση έκτακτης ανάγκης (E911) προς αυτές. Η οδηγία αυτή προέβλεπε αρχικά τον εντοπισμό του καλούντος με βάση μόνο την ταυτότητα της κυψέλης από την οποία έκανε την κλήση. Μετά από την πρώτη αυτή φάση θα έπρεπε οι παροχείς των δικτύων να παρέχουν πληροφορία θέσης με ακρίβεια:

- 100 μέτρων για το 67% των κλήσεων και 300 μέτρων για το 95% των κλήσεων, αν χρησιμοποιούταν εντοπισμός θέσης βασισμένος στο δίκτυο (βλ. ενότητα «Εντοπισμός Θέσης»).
- 50 μέτρων για το 67% των κλήσεων και 150 μέτρων για το 95% των κλήσεων, αν χρησιμοποιούταν εντοπισμός θέσης βασισμένος στο τερματικό.

Οι παροχείς των ΗΠΑ εφόσον εγκατέστησαν την υποδομή που απαιτούνταν, αποφάσισαν να την εκμεταλλευτούν και εμπορικά (η παροχή πληροφορίας θέσης στο E911 ήταν μη κερδοσκοπική). Έτσι άρχισαν να αναδεικνύονται και να υλοποιούνται οι πρώτες υπηρεσίες θέσης. Η Ευρώπη έφτασε στο ίδιο σημείο αλλά από διαφορετικό δρόμο. Οι βιομηχανίες και οι παροχείς της Ευρώπης άρχισαν να αναπτύσσουν τις υπηρεσίες αυτές για τους λόγους που αναφέρθηκαν στην αρχή της παρούσας ενότητας (κορεσμός του GSM). Εξάλλου στην Ευρώπη δεν υπήρξε αντίστοιχη δράση με του FCC.

Οι πρώτες αρχιτεκτονικές των συστημάτων που υποστήριζαν τέτοιες υπηρεσίες μπορούν να χαρακτηριστούν σαν μονολιθικές (1997-2001). Οι παροχείς των δικτύων «άνοιξαν» σημεία πρόσβασης (διεπαφές) προς τα υποσυστήματα εντοπισμού θέσης που διέθεταν. Όσοι διέθεταν γεωγραφικό περιεχόμενο μπορούσαν μέσω αυτών των διεπαφών να δημιουργήσουν υπηρεσίες θέσης. Οι διεπαφές δεν στηρίζονταν σε ανοικτά και αναγνωρισμένα πρότυπα και το κύριο λόγο είχε ο ιδιοκτήτης του δικτύου. Στην συνέχεια (2001- σήμερα), οι παροχείς των δικτύων αντιλήφθηκαν ότι έπρεπε να υπάρξει μια ευρύτερη συνεργασία με τις εταιρίες-παροχείς του χώρου αυτού και έτσι άρχισε μια πιο συστηματική προσπάθεια υλοποίησης νέων αρχιτεκτονικών. Αυτές στηρίζονται σε σχεδιαστικά μοντέλα (design) που επιτρέπουν την ευελιξία και την αναπροσαρμογή του όλου συστήματος στις απαιτήσεις των νέων εφαρμογών και υπηρεσιών.

Οι πρώτες υπηρεσίες θέσης άρχισαν να εμφανίζονται στην Ευρώπη το 1998. Σε αυτές συμπεριλαμβάνονται η πλοήγηση (navigation) του χρήστη (ακόμη και με την βοήθεια χαρτών), η εύρεση των πιο κοντινών σημείων ενδιαφέροντος (ATMs, εστιατόρια, φαρμακεία), η παροχή πληροφοριών για την περιοχή του χρήστη (χάρτης, τουριστικά αξιοθέατα, καιρός κ). Επίσης έχουν υλοποιηθεί και κάποιες υπηρεσίες που αφορούν κυρίως επιχειρήσεις, όπως η διαχείριση στόλου (fleet management) σε συνδυασμό και με την χρήση τηλεματικής. Στις ΗΠΑ η πρώτη υπηρεσία ήταν οι κλήσεις έκτακτης ανάγκης E911 που άρχισαν να παρέχονται το 1997.

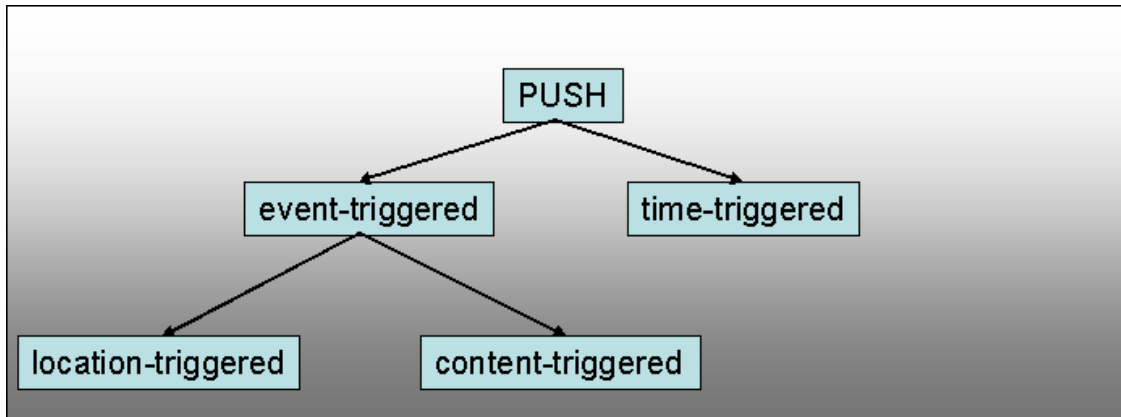
### 1.1.3 Αυτόματη Προώθηση

Η Αυτόματη Προώθηση (θα αναφέρεται και ως Push για συντομία) δεν είναι ένα νέο μοντέλο παροχής δικτυακών υπηρεσιών. Μια από τις πρώτες μορφές της είναι η επικεφαλίδα “Refresh” που ανέπτυξαν οι κατασκευαστές περιηγητών διαδικτύου (web browsers) και αποτελεί επέκταση του πρωτοκόλλου HTTP 1.1. Φυσικά η επικεφαλίδα αυτή αποτελούσε απλώς μια προσομοίωση αυτόματης προώθησης αφού τελικά χρησιμοποιείται το κλασικό μοντέλο πελάτη-εξυπηρετή. Στον χώρο των κινητών επικοινωνιών οι πρώτες εφαρμογές Push υλοποιήθηκαν με το πρωτόκολλο SMS. Στην συνέχεια ο οργανισμός WAP Forum, αφού αντιλήφθηκε την σημασία του και τις δυνατότητες της εμπορικής εκμετάλλευσής του, ενσωμάτωσε το Push στην έκδοση 1.2 του πρωτοκόλλου WAP.

Τι ακριβώς όμως σημαίνει ο όρος Push; Με τον όρο αυτό εννοούμε την αποστολή περιεχομένου σε ένα χρήστη δικτύου, χωρίς ο ίδιος να την έχει ζητήσει ρητά την στιγμή που την λαμβάνει. Ουσιαστικά είναι σαν να «σπρώχνει» το δίκτυο την πληροφορία στον χρήστη, εξ’ ου και το όνομα Push. Το μοντέλο αυτό είναι ακριβώς το αντίθετο από το Pull. Το τελευταίο είναι το μοντέλο που ακολουθείται κατά κύριο λόγο στο διαδίκτυο (client-server ή request/response). Στο Pull ο χρήστης κάνει αίτηση για να λάβει κάποιο περιεχόμενο και το δίκτυο του στέλνει σαν απάντηση το περιεχόμενο αυτό ή ένα μήνυμα αποτυχίας. Κατ’ αναλογία μια Push υπηρεσία είναι μια εφαρμογή λογισμικού που γνωρίζει τι είδους περιεχόμενο ενδιαφέρει τον κάθε χρήστη της και του το παραδίδει όταν αυτό είναι διαθέσιμο ή τον πληροφορεί από που μπορεί να το πάρει. Οι υπηρεσίες αυτές άρχισαν να αποκτούν όλο και μεγαλύτερη αποδοχή από τις κινητές επικοινωνίες σαν υπηρεσίες επιπρόσθετης αξίας (value added services).

Βέβαια στο Push μοντέλο το δίκτυο δεν στέλνει αυθαίρετα στον χρήστη μηνύματα (με την ευρύτερη έννοια). Πρέπει προηγουμένως αυτός να έχει δηλώσει την επιθυμία του για παροχή των εν λόγω υπηρεσιών. Αυτό συνήθως συμπεριλαμβάνει την εγγραφή (registration) ή συνδρομή του (subscription) στην Push υπηρεσία και τον σαφή καθορισμό των όρων του συμβολαίου. Η επιθυμία του χρήστη όμως, δεν γίνεται πάντα σεβαστή και πολλές εταιρίες εκμεταλλεύονται τέτοιες υπηρεσίες για διαφημιστικούς κυρίως λόγους. Με τον τρόπο αυτό ενοχλούν τον συνδρομητή που πρέπει να αναγκαστεί να φιλτράρει πλέον το Push περιεχόμενο που λαμβάνει (βλ. και ενότητα «Ιδιωτικότητα»).

Μια υπηρεσία Push μπορεί να λειτουργεί με τρεις τρόπους: να την σκανδαλίζει κάποιο συμβάν (event-triggered), ή να την σκανδαλίζει κάποιος χρονοπρογραμματιστής (time-triggered). Το συμβάν του πρώτου τρόπου μπορεί να είναι είτε η αλλαγή της θέσης του χρήστη ή κάποιας άλλης οντότητας που τον ενδιαφέρει -πχ φίλος του- (location-triggered) είτε η διάθεση/δημοσίευση κάποιου περιεχομένου που έχει αξία για τον χρήστη (content-triggered). Όλοι οι παραπάνω τρόποι συνοψίζονται στην παρακάτω εικόνα:



**Εικόνα 1: Οι τρόποι σκανδαλισμού μιας Push υπηρεσίας**

Το location-triggered Push, που αποτελεί και το κύριο θέμα της εργασίας αυτής, έχει να κάνει με την αποστολή μηνυμάτων όταν ικανοποιείται κάποια χωρική συνθήκη (spatial condition) που έχει ορίσει ο χρήστης κατά την εγγραφή του σε μια υπηρεσία. Στο time-triggered Push, που επίσης τίγεται σε αυτήν την εργασία στα πλαίσια του παραδειγματικού χρονοπρογραμματιστή (scheduler) που υλοποιήθηκε, η συνθήκη που πρέπει να ικανοποιηθεί για την εκτέλεση της υπηρεσίας είναι χρονική. Για παράδειγμα, μπορεί ο χρήστης να λάβει κάποια πληροφορία μια φορά σε μια συγκεκριμένη χρονική στιγμή ή σε τακτά χρονικά διαστήματα, που έχει επίσης καθορίσει προκαταβολικά κατά την εγγραφή του.

Η τρίτη περίπτωση, το content-triggered Push, εμπεριέχει την εκτέλεση της υπηρεσίας όταν γίνουν διαθέσιμα κάποια δεδομένα ή κάποιες πληροφορίες που ενδιαφέρουν τον χρήστη. Αυτό μπορεί να υλοποιηθεί με διάφορους τρόπους. Ένας από αυτούς [1] είναι με μια αρχιτεκτονική Publisher/Subscriber (P/S). Σε αυτήν ο Publisher ενημερώνει το σύστημα για την παρουσία νέου περιεχομένου. Οι Subscribers έχουν δηλώσει την επιθυμία τους να ειδοποιούνται, ή και να τους αποστέλλεται, όταν υπάρχει νέο περιεχόμενο που να τους ενδιαφέρει. Η ενημέρωσή τους γίνεται από το σύστημα και μπορεί να είναι και ασύγχρονη, με την χρήση κατάλληλων ουρών, στην περίπτωση που ο συνδρομητής δεν έχει ενεργοποιημένη την συσκευή του. Από τα παραπάνω βλέπουμε ότι το Push μπορεί να απευθύνεται και σε πολλούς χρήστες ταυτόχρονα (multicast). Αυτό εξάλλου υποστηρίζεται και από το WAP.

Για να λειτουργήσει όμως σωστά και αποτελεσματικά το μοντέλο αυτόματης προώθησης πρέπει να πληρούνται κάποιες προϋποθέσεις. Η πρώτη και πιο σημαντική είναι να υποστηρίζεται από το δίκτυο ποιότητα υπηρεσίας (QoS). Οι περισσότερες υπηρεσίες Push είναι άχρηστες αν δεν παραληφθούν άμεσα από τον χρήστη (πχ ειδοποίηση για κυκλοφοριακή κίνηση). Βέβαια η ποιότητα υπηρεσίας είναι πιο δύσκολο να εξασφαλιστεί στα δίκτυα που υποστηρίζουν υπηρεσίες Push απ' ότι σε αυτά που προσφέρουν μόνο τις κλασικές υπηρεσίες Pull. Αυτό συμβαίνει επειδή η δικτυακή «συμπεριφορά» του μοντέλου Push έχει κάποιες ιδιαιτερότητες που δεν την καθιστούν προβλέψιμη (και όπως είναι γνωστό η πρόβλεψη παίζει μεγάλο ρόλο στην εξασφάλιση ποιότητας υπηρεσίας). Μία από αυτές είναι το ότι η δικτυακή κίνηση που προκαλείται είναι απρόβλεπτη, αφού δεν εξαρτάται τόσο από τον ίδιο τον χρήστη όσο από την στατιστική συμπεριφορά (στο πεδίο του χρόνου) του συμβάντος που σκανδαλίζει την Push υπηρεσία. Το δίκτυο μπορεί να υποπέσει σε ακόμα χειρότερη κατάσταση αν οι

Push υπηρεσίες αιτηθούν ξαφνικά από μεγάλο πλήθος συνδρομητών. Επίσης οι Push υπηρεσίες συνήθως μεταφέρουν περιεχόμενο. Και εφόσον αυτό μπορεί να είναι όσο ογκώδες θέλει ο χρήστης, το δίκτυο διατρέχει μεγαλύτερο κίνδυνο να υπερφορτωθεί (αυτή είναι μια ιδιαιτερότητα γενικά των υπηρεσιών προστιθέμενης αξίας).

Για να γίνει πιο κατανοητή η έννοια της αυτόματης προώθησης παρατίθεται και ένα σενάριο χρήσης της: Ο συνδρομητής εγγράφεται σε μια υπηρεσία πρόγνωσης καιρού. Κατά την εγγραφή του δηλώνει ότι θέλει να πληροφορείται για την πρόγνωση των τοπικών καιρικών συνθηκών κάθε πρωί στις 8:00. Η συγκεκριμένη υπηρεσία κάθε πρωί στις 8:00 ζητάει από το δίκτυο του συνδρομητή να της αποσταλεί η θέση του. Με την πληροφορία που αποκτά κάνει μια επερώτηση στη μετεωρολογική βάση με τις προγνώσεις για όλες τις περιοχές. Έτσι βρίσκει την πρόγνωση που χρειάζεται ο συνδρομητής και του την αποστέλλει με μήνυμα SMS. Ακόμα και αν η τερματική συσκευή του συνδρομητή είναι απενεργοποιημένη στις 8:00, η υπηρεσία SMS εγγυάται ότι αυτός θα λάβει το μήνυμα αργότερα.

Η κύρια χρήση του Push για πολύ καιρό ήταν η κοινοποίηση (notification) ή προειδοποίηση (alert) του χρήστη για κάποιο συμβάν. Σήμερα με τις νέες τεχνολογικές δυνατότητες που υπάρχουν, προτείνονται και υλοποιούνται συνεχώς νέες υπηρεσίες. Οι βασικότερες από αυτές θεωρείται ότι είναι αυτές που εξαρτώνται από την θέση. Μερικές τέτοιες αναφέρονται στην ενότητα που ακολουθεί.

#### 1.1.4 Παραδείγματα Υπηρεσιών Θέσης

Σήμερα έχουν υλοποιηθεί αρκετές υπηρεσίες θέσης και έχουν προταθεί ακόμα περισσότερες. Οι κυριότερες υπηρεσίες είναι:

- Διαφημίσεις (Advertising)
- Ιχνηλασία (Tracking)
- Πλοήγηση (Navigation)
- Σημεία Ενδιαφέροντος (Points of Interest)
- Υπηρεσίες Έκτακτης Ανάγκης (Emergency Services)
- Διασκέδαση (Entertainment)
- Χρέωση και Διαχείριση (Billing and Management)

Ακολουθεί συνοπτική περιγραφή των προαναφερθέντων υπηρεσιών.

##### **Διαφημίσεις (Advertising)**

Μπορεί να χωριστεί σε δύο κατηγορίες: **Push advertising** και **Pull advertising**. Η πρώτη κατηγορία περιγράφει την περίπτωση στην οποία το κινητό τερματικό δέχεται διαφημιστικά μηνύματα ανάλογα με την περιοχή στη οποία βρίσκεται (π.χ. μηνύματα μαγαζιών, εστιατορίων κ.τ.λ.) *αυτόματα* χωρίς ο ιδιοκτήτης του να έχει προηγουμένως ζητήσει την υπηρεσία. Η δεύτερη κατηγορία είναι ακριβώς ίδια με την πρώτη με την διαφορά ότι ο χρήστης έχει τώρα ζητήσει μέσω του τερματικού του την υπηρεσία πληρώνοντας το αντίστοιχο αντίτιμο για τα διαφημιστικά μηνύματα που λαμβάνει. Στις διαφημίσεις συμπεριλαμβάνονται και οι μικρές αγγελίες (yellow pages).

### **Ιχνηλασία (Tracking)**

Η κατηγορία αυτή περιλαμβάνει υπηρεσίες που σχετίζονται με την παρακολούθηση και τον εντοπισμό αυτοκινήτων, κατοικίδιων ακόμη και μικρών παιδιών. Ειδικά η περίπτωση της παρακολούθησης των αυτοκινήτων, γνωστή και ως διαχείριση στόλου (fleet management), χρησιμοποιείται από εταιρείες που έχουν έναν στόλο αυτοκινήτων των οποίων είναι γνωστή η γεωγραφική τους θέση ανά πάσα στιγμή σε πραγματικό χρόνο. Το γεγονός αυτό καθιστά δυνατή την εύκολη διαχείριση του στόλου με άμεσο αποτέλεσμα την καλύτερη εκμετάλλευση του αλλά και την μείωση του κόστους λειτουργίας του.

### **Πλοήγηση (Navigation)**

Μία ακόμη σημαντική κατηγορία είναι η Πλοήγηση (Navigation). Ειδικότερες περιπτώσεις αυτής της κατηγορίας είναι πληροφορίες για την κατάσταση της οδικής κυκλοφορίας, όπως επίσης και για διάφορα γεγονότα που σχετίζονται με αυτή (π.χ. ατυχήματα, κλειστοί δρόμοι κτλ.). Όλα αυτά συμπληρώνονται και από την δυνατότητα που έχει ο χρήστης να λαμβάνει οδηγίες για την συντομότερη διαδρομή μεταξύ δύο σημείων που επιλέγει. Αν το σύστημα είναι αρκετά εξελιγμένο τότε δεν λαμβάνεται μόνο υπ' όψιν η γεωγραφική απόσταση μεταξύ των δύο αυτών σημείων αλλά και πληροφορίες για την κατάσταση της κυκλοφορίας με σκοπό την ελαχιστοποίηση του χρόνου της διαδρομής.

### **Σημεία Ενδιαφέροντος (Points of Interest)**

Υπάρχουν επίσης και εφαρμογές που προσφέρουν πληροφορίες για αναζητήσεις συγκεκριμένων σημείων που ενδιαφέρουν τον χρήστη. Αυτές οι αναζητήσεις μπορεί να είναι του τύπου «Ποια είναι τα κοντινότερα εστιατόρια;» ή «Ποια είναι τα κοντινότερα φαρμακεία;». Σε αυτές θα μπορούσαμε να συμπεριλάβουμε και την εύρεση οικείων προσώπων (friend finders) της μορφής «Που βρίσκεται αυτή την στιγμή ο αδερφός μου;» ή «Ποιοι φίλοι μου βρίσκονται σε ακτίνα 500 μ. από εμένα;».

### **Υπηρεσίες Έκτακτης Ανάγκης ( Emergency Services )**

Μια εξίσου σημαντική εφαρμογή των LBS είναι αυτή της δυνατότητας εντοπισμού της θέσης ενός ατόμου ή ενός οχήματος όταν αυτό εκπέμπει σήμα έκτακτης ανάγκης με αποτέλεσμα την αποστολή άμεσης βοήθειας προς αυτό. Ειδικά για την περίπτωση των οχημάτων, μπορεί να εξοπλιστούν με συσκευές ανίχνευσης έτσι ώστε σε περίπτωση που κλαπούν να είναι εύκολος ο εντοπισμός τους.

### **Διασκέδαση ( Entertainment )**

Η περιορισμένη διαθεσιμότητα αλλά και το κόστος των διαφόρων συσκευών εντοπισμού θέσης είχε παλιότερα σαν αποτέλεσμα την μη χρησιμοποίηση αυτών σε εφαρμογές που ανήκουν στην ευρύτερη κατηγορία της διασκέδασης (entertainment). Με την ανάπτυξη όμως τεχνικών εντοπισμού θέσης για δίκτυα GSM, αλλά και την αύξηση του εύρους ζώνης με τα δίκτυα νέων γενιών, έχουμε παράλληλα και την ανάπτυξη αυτών των εφαρμογών, όπως η ανταλλαγή μηνυμάτων μεταξύ ατόμων που βρίσκονται στον ίδιο χώρο ή υπηρεσίες ραντεβού (dating services) κα. Από την εμπειρία στον χώρο των υπολογιστών, μπορεί κανείς να υπολογίζει ότι θα αναπτυχθούν και αρκετά παίγνια, στα

οποία ένας παράγοντας που θα επηρεάζει την εξέλιξή τους θα είναι, ίσως, η θέση του χρήστη.

### **Χρέωση και Διαχείριση (Billing and Management)**

Η πληροφορία θέσης των συνδρομητών μπορεί να χρησιμοποιηθεί από τους παροχείς των δικτύων για υποστήριξη ευέλικτων και προοδευτικών συστημάτων χρέωσης των πελατών τους. Έτσι μπορεί ένας παροχέας να χρεώνει τους συνδρομητές του με επίπεδη χρέωση (flat rate) όταν τηλεφωνούν από το σπίτι τους, για να τους παρακινήσει να το κάνουν, και να τους χρεώνει με χρονοχρέωση όταν κάνουν κλήσεις από αλλού. Επίσης μπορεί η ίδια πληροφορία να χρησιμοποιηθεί για την καλύτερη διαχείριση και λειτουργία του δικτύου από τον ίδιο τον παροχέα. Όταν ένας παροχέας ξέρει με αρκετή ακρίβεια τα στατιστικά της κίνησης και θέσης των συνδρομητών του μπορεί να οργανώσει και να επεκτείνει καλύτερα το δίκτυό του προτού αυτό να υπερφορτωθεί.

Εκτός από τις παραπάνω γενικές υπηρεσίες που μπορούν να υλοποιηθούν με βάση και τα δυο μοντέλα, Push και Pull, υπάρχουν και κάποιες που θα ταίριαζαν καλύτερα στο μοντέλο Αυτόματης Προώθησης, χωρίς αυτό να σημαίνει ότι δεν θα μπορούσαν να υλοποιηθούν και με το άλλο μοντέλο. Αυτές είναι:

- *Ειδήσεις:* πολλές υπηρεσίες ειδήσεων – από εφημερίδες μέχρι τηλεοπτικούς σταθμούς – είναι ήδη προσπελάσιμες μέσω του Internet. Το ειδησεογραφικό περιεχόμενό τους αλλάζει συνεχώς και έτσι θα επωφελούνταν από την χρήση ενός Push μοντέλου σε συνδυασμό με τους κατάλληλους μηχανισμούς φιλτραρίσματος.
- *Πρόγνωση καιρού:* ο καιρός αλλάζει συνεχώς και έτσι μπορεί να αξιοποιηθεί σαν περιεχόμενο για μια Push υπηρεσία. Σε συνδυασμό με κατάλληλα φίλτρα μπορεί να παρέχει πολύτιμη ενημέρωση σε πολλές κατηγορίες χρηστών (πχ να ειδοποιεί κάποιον που ασχολείται με την ναυσιπλοΐα για τις εναλλαγές της παλίρροιας).
- *Ανακοινώσεις κυκλοφοριακής κίνησης:* τα μποτιλιαρίσματα στους δρόμους είναι απρόβλεπτα και πολλοί θα ήθελαν να ενημερώνονται αυτόματα γι' αυτά ανάλογα με την ώρα της ημέρας, την περιοχή τους κλπ.
- *Ενημερώσεις για δρομολόγια:* πολλά δρομολόγια μέσω μεταφοράς ακυρώνονται ή καθυστερούν για διάφορους λόγους. Το αποτέλεσμα είναι πάντα εκνευρισμός και ταλαιπωρία για τους άτυχους χρήστες τους. Μια υπηρεσία Push θα μπορούσε να μετριάσει αυτές τις συνέπειες με έγκαιρη ενημέρωση.
- *Ενημέρωση τοπικών πολιτιστικών δραστηριοτήτων:* πολλοί θα ενδιαφέρονταν να ειδοποιούνται αυτόματα για τις πολιτιστικές δραστηριότητες της περιοχής τους. Αυτές μπορεί να είναι μουσικές συναυλίες, νέες θεατρικές παραστάσεις, κινηματογραφικές προβολές, καλλιτεχνικές εκθέσεις και οτιδήποτε άλλο.
- *Ενημέρωση από την τοπική αυτοδιοίκηση:* πολλές αρχές τοπικής αυτοδιοίκησης έχουν αρχίσει να κάνουν χρήση νέων τεχνολογιών για να χειριστούν καλύτερα και πιο άμεσα τα προβλήματα των πολιτών και να τους παρέχουν καλύτερες υπηρεσίες. Μια τέτοια υπηρεσία μπορεί να είναι και η ενημέρωση για τοπικά θέματα, όπως την προσωρινή διακοπή της υδροδότησης λόγω έργων.

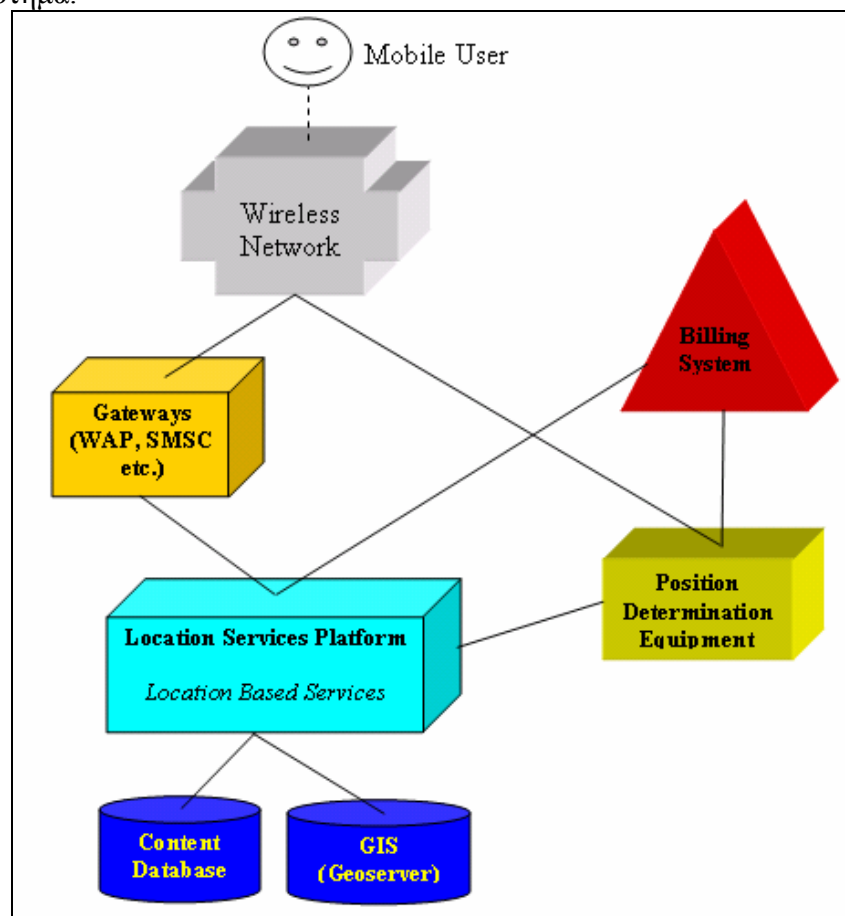
- *Αποφυγή μπλόκων*: φυσικά δεν γίνεται να μην υπάρχουν και «πονηρές» υπηρεσίες! Ο χρήστης θα μπορούσε να ενημερώνεται αυτόματα για τυχόν μπλόκα της τροχιάς στην περιοχή του. Αυτή όμως είναι μια υπηρεσία που μάλλον θα αργήσουμε να δούμε!

## 1.1.5 Γενική Αρχιτεκτονική

Η αρχιτεκτονική ενός συστήματος παροχής υπηρεσιών θέσης μπορεί να μελετηθεί σε πολλά επίπεδα. Στην παρούσα ενότητα θα περιγραφούν οι οντότητες ενός τέτοιου συστήματος, η αρχιτεκτονική σε επίπεδο υπηρεσιών-λογισμικού και ένα παράδειγμα αρχιτεκτονικής για GSM δίκτυα.

### 1.1.5.1 Οντότητες Συστήματος

Στην εικόνα 2 φαίνονται οι οντότητες (μονάδες) από τις οποίες αποτελείται ένα τέτοιο σύστημα.



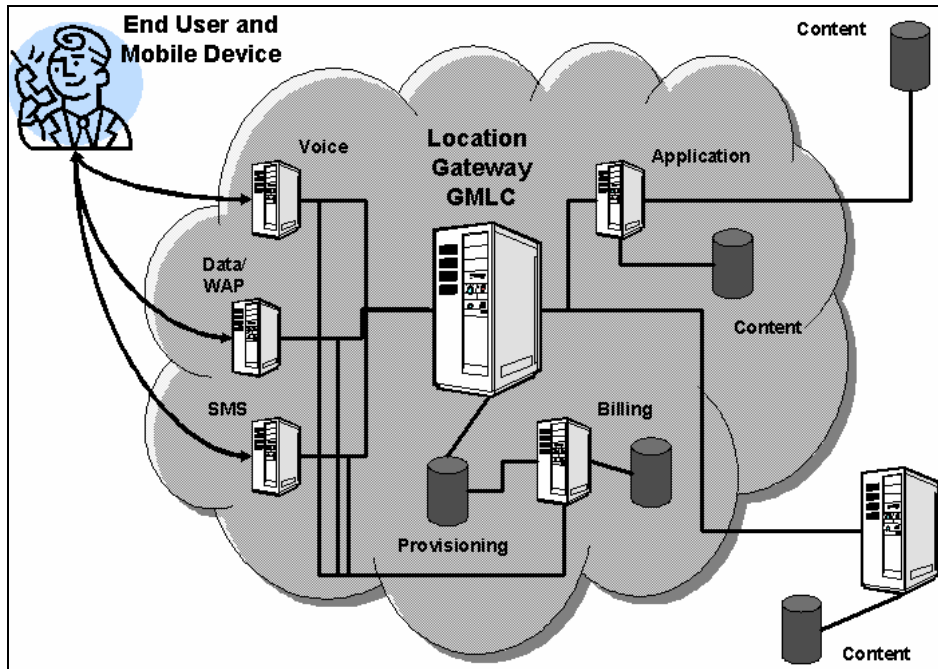
Εικόνα 2: Οι φυσικές οντότητες ενός συστήματος παροχής υπηρεσιών θέσης



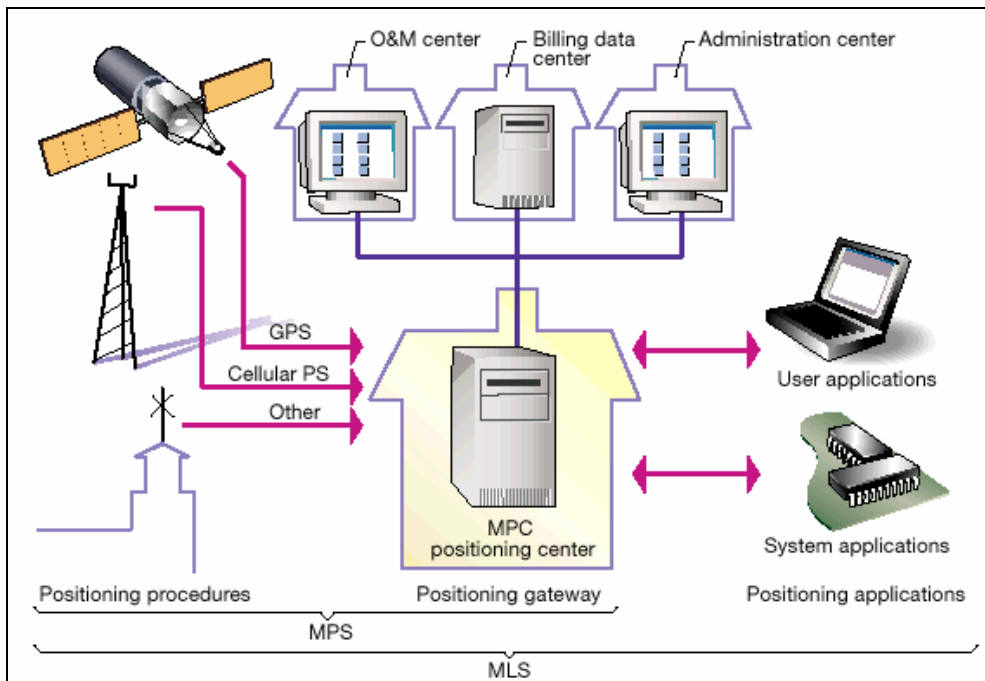
Αυτές οι οντότητες είναι:

- **το ασύρματο δίκτυο (wireless network):** είναι το μέσο επικοινωνίας μεταξύ του συνδρομητή-χρήστη και των υπηρεσιών θέσης. Αυτό συμμετέχει σε πολλές διαδικασίες, μεταξύ των οποίων είναι το να παρέχει δεδομένα από την HLR στην πλατφόρμα και να συμβάλει στον υπολογισμό της πληροφορίας θέσης. Το δίκτυο δεν αποτελεί περιορισμό για τις υπηρεσίες θέσης (μπορούν να υλοποιηθούν και σε δίκτυα όλων των γενιών)
- **ο εξοπλισμός προσδιορισμού θέσης (position determination equipment):** είναι ο εξοπλισμός που χρησιμοποιείται για την υλοποίηση των τεχνικών εντοπισμού θέσης. Περιλαμβάνει μονάδες όπως οι LMUs (Location Measurement Units), που εξαρτώνται από την επιλεγμένη τεχνική, αλλά και τον υπάρχον εξοπλισμό του δικτύου (GMSC και SMLC για GSM δίκτυα και MPC για ANSI δίκτυα). Έτσι ο διαχωρισμός της από το δίκτυο δεν είναι πολύ σαφής.
- **η πλατφόρμα υπηρεσιών θέσης (location services platform):** αυτή είναι ο συνδετικός κρίκος μεταξύ των υπόλοιπων οντοτήτων και αναλαμβάνει την εκτέλεση των υπηρεσιών (location based services).
- **οι υπηρεσίες που εξαρτώνται από την θέση (location based services):** οι υπηρεσίες αυτές είναι το τελικό αποτέλεσμα που προκύπτει από την κατάλληλη αλληλεπίδραση όλων των υπόλοιπων οντοτήτων.
- **το σύστημα χρέωσης (billing system):** μπορεί να συλλέγει στοιχεία από οποιεσδήποτε οντότητες ανάλογα με το συνολικό σχεδιασμό του συστήματος.
- **οι βάσεις περιεχομένου (Content database και GIS):** αυτές παρέχουν τις γεωγραφικές πληροφορίες στην πλατφόρμα για την εκτέλεση των υπηρεσιών θέσης.

Στο επόμενο σχήματα φαίνονται άλλες δυο απόψεις της γενικής αρχιτεκτονικής (η εικόνα αναφέρεται στην ολοκληρωμένη λύση της Ericsson [2]) :



Εικόνα 3: Η γενική αρχιτεκτονική ενός συστήματος υπηρεσιών θέσης

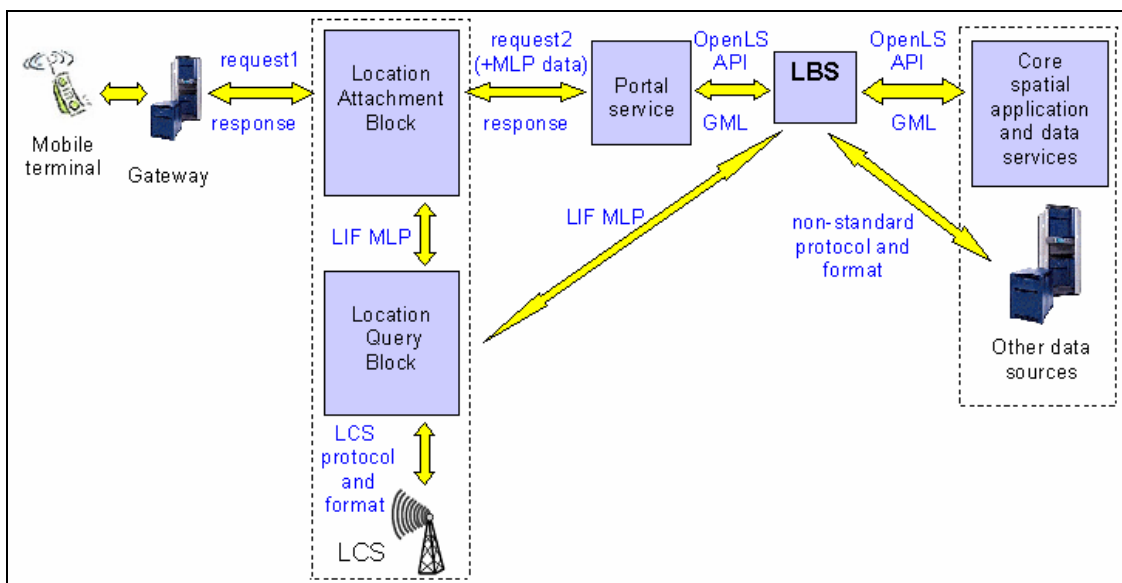


Εικόνα 4: Η αρχιτεκτονική της Ericsson για ένα σύστημα υπηρεσιών θέσης (MLS: Mobile Location Solution, MPC: Mobile Location Center, MPS: Mobile Positioning Server)

### 1.1.5.2 Αρχιτεκτονική σε επίπεδο υπηρεσιών και λογισμικού

Σε αυτήν την υποενότητα θα δούμε πως δημιουργείται η απάντηση της υπηρεσίας (response) στην αίτηση του χρήστη (request). Η περιγραφή που ακολουθεί, και που αποτελεί ένα σενάριο Pull υπηρεσίας, έχει σαν σκοπό να αναδείξει τις επιμέρους υπηρεσίες και εφαρμογές που συνεργάζονται για την παροχή μιας ολοκληρωμένης υπηρεσίας στον χρήστη. Πριν την περαιτέρω ανάλυση αξίζει να επισημανθεί ότι το μεγαλύτερο μέρος της ανταλλαγής δεδομένων σε ένα τέτοιο σύστημα βασίζεται στην XML κωδικοποίηση των μηνυμάτων που ανταλλάσσονται. Επίσης σαν μέσο μεταφοράς χρησιμοποιείται κυρίως το HTTP Post. Τέλος, η συγκεκριμένη αρχιτεκτονική δεν αποτελεί την μόνη αποδεκτή αλλά μία από τις πιο δημοφιλείς.

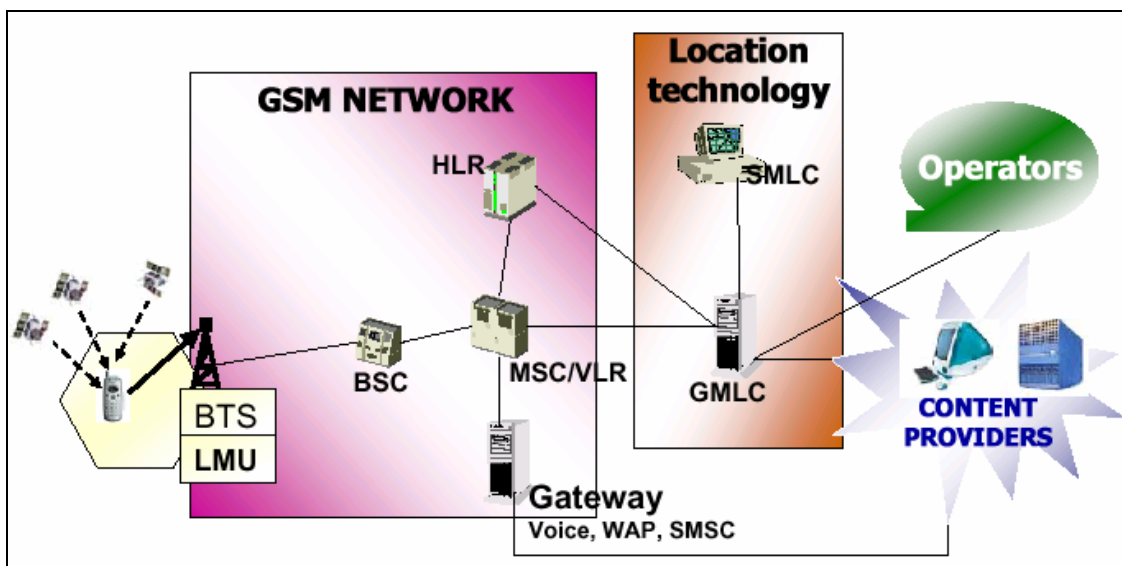
Ο κινητός χρήστης, λοιπόν, κάνει μια αίτηση για παροχή υπηρεσίας θέσης. Το δίκτυο, μαζί με την αίτησή του, προωθεί και την πληροφορία που αφορά την θέση του προς την πλατφόρμα η οποία εκτελεί την συγκεκριμένη υπηρεσία.. Αυτή η επισύναψη της θέσης δεν είναι βέβαια υποχρεωτικό να γίνει σε αυτό το στάδιο αφού μπορεί και αργότερα η υπηρεσία (LBS) να ζητήσει την θέση του χρήστη. Για την ανάκτηση της θέσης πάντως, χρησιμοποιείται το Mobile Location Protocol (MLP, βλ. παράγραφο 1.2.4.3 για σύντομη περιγραφή). Η εύρεση αυτής της πληροφορίας γίνεται από μια υπηρεσία που έχει άμεση επικοινωνία με το σύστημα εντοπισμού (LCS, LoCation Service). Η υπηρεσία πύλης (portal service) είναι, όπως υποδηλώνει και το όνομά της, το σημείο συσχέτισης μεταξύ της υπηρεσίας θέσης (LBS) και του υπόλοιπου δικτύου. Αυτή επικοινωνεί με διάφορα APIs με την υπηρεσία θέσης αλλά το πιο σημαντικό είναι αυτό του OpenLS (βλ. παράγραφο 1.2.4.2 για σύντομη περιγραφή). Η υπηρεσία θέσης αποκτά πρόσβαση στο περιεχόμενο που την ενδιαφέρει να στείλει στον χρήστη μέσω τυποποιημένων γλωσσών (GML) ή μη τυποποιημένων (proprietary) διεπαφών και κωδικοποιήσεων δεδομένων (format). Τελικά, το περιεχόμενο συλλέγεται στην πλατφόρμα και μετά από κατάλληλη μορφοποίησή του, αποστέλλεται στον χρήστη. Όλη η παραπάνω διαδικασία απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 5: Αρχιτεκτονική υπηρεσιών και πρωτοκόλλων λογισμικού.

### 1.1.5.3 Αρχιτεκτονική στο GSM

Οι περισσότερες υπηρεσίες θέσης σήμερα έχουν υλοποιηθεί σε GSM δίκτυα, καθώς αυτό είναι το πιο διαδεδομένο σύστημα κινητών επικοινωνιών στον κόσμο. Στην παρακάτω εικόνα φαίνεται η διάρθρωση των στοιχείων ενός τέτοιου δικτύου που υποστηρίζει και υπηρεσίες θέσης. Οι κυριότερες μονάδες του, όσον αφορά τις υπηρεσίες αυτές, είναι το SMLC (Serving Mobile Location Centre) και το GMLC (Gateway Mobile Location Centre). Τα κέντρα αυτά καθορίστηκαν από οργανισμούς όπως ο ETSI και ο 3GPP για τις ανάγκες των υπηρεσιών θέσης.

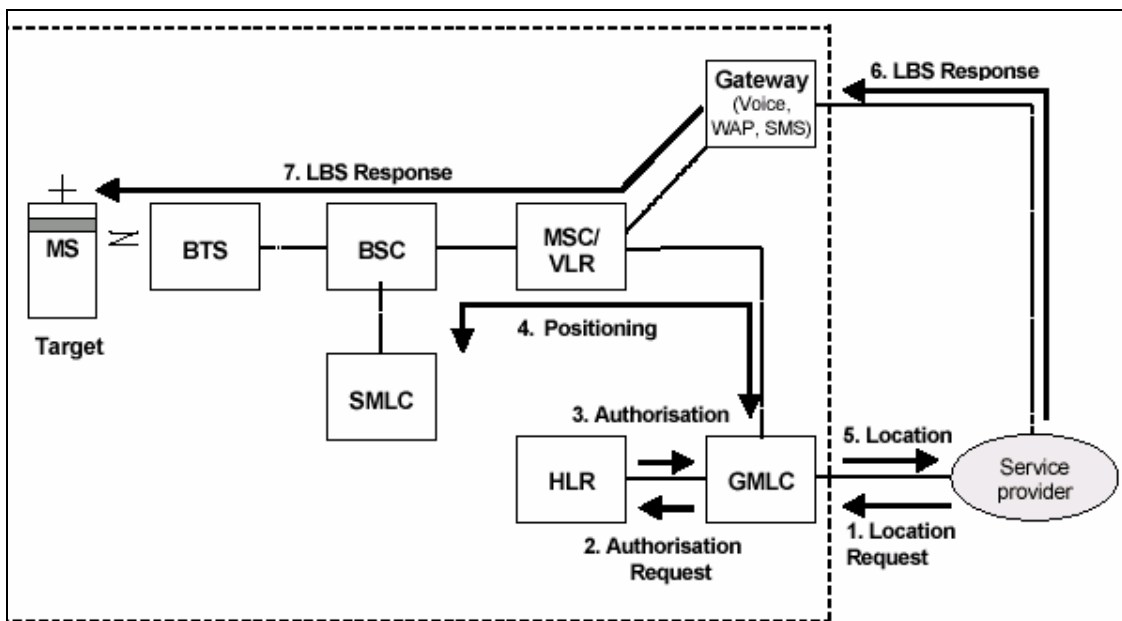


Εικόνα 6: Αρχιτεκτονική υπηρεσιών θέσης για GSM δίκτυο.

Το GMLC εξυπηρετεί σαν η κύρια ενδιάμεση πύλη για τις εφαρμογές θέσης λαμβάνοντας αιτήσεις από τις εφαρμογές, συλλέγοντας την πληροφορία θέσης από το δίκτυο ή από το SMLC και παρέχοντάς την στις υπηρεσίες και εφαρμογές που την χρειάζονται και που είναι εξουσιοδοτημένες να την χρησιμοποιήσουν. Σε ένα PLMN δίκτυο (Public Land Mobile Network) μπορούν να υπάρχουν περισσότερα από ένα GMLC. Το GMLC χρησιμοποιείται και στις προδιαγραφές του UMTS. Το SMLC εξυπηρετεί τις αιτήσεις για την εύρεση της θέσης των κινητών τερματικών και συνδυάζει τις πληροφορίες που συλλέγει από το δίκτυο και από τον εξοπλισμό εντοπισμού θέσης. Μπορεί να είναι μια ξεχωριστή οντότητα του δικτύου ή μια λειτουργικότητα ενσωματωμένη στο BSC (Base Station Controller). Το SMLC μπορεί να ελέγχει ένα πλήθος από LMUs. Ουσιαστικά είναι το σημείο στο οποίο υπολογίζεται η τελική θέση του κινητού.

Για να γίνει πιο κατανοητή η αρχιτεκτονική στο GSM και η διαδικασία της Αυτόματης Προώθησης στα πλαίσια αυτής της αρχιτεκτονικής ακολουθεί το παρακάτω σενάριο εκτέλεσης μιας Push υπηρεσίας σε GSM δίκτυο (απεικονίζεται στην εικόνα 7):

1. Η υπηρεσία θέσης ζητάει την θέση του χρήστη για να μπορέσει να συλλέξει το σχετικό περιεχόμενο και να το παραδώσει στον χρήστη. Η αίτηση για την θέση του αποστέλλεται στο GMLC.
- 2,3. Γίνεται η πιστοποίηση των στοιχείων του συνδρομητή με την βοήθεια της HLR.
4. Το GMLC ζητάει από το SMLC την θέση του χρήστη. Το SMLC χρησιμοποιώντας τις LMU μονάδες υπολογίζει την θέση του και την αποστέλλει στο GMLC.
5. Το GMLC στέλνει την πληροφορία θέσης στη πλατφόρμα όπου εκτελείται η υπηρεσία θέσης. Εκεί συλλέγεται και μορφοποιείται το απαραίτητο περιεχόμενο.
- 6,7. Το αποτέλεσμα της εκτέλεσης της υπηρεσίας στέλνεται στην κατάλληλη πύλη (gateway) και από εκεί στην τερματική συσκευή του συνδρομητή.



Εικόνα 7: Push σενάριο σε ένα GSM δίκτυο.

## 1.2 Τεχνολογίες και Πρότυπα

### 1.2.1 Εντοπισμός Θέσης

Το σημαντικότερο ίσως στοιχείο σε ένα σύστημα LBS είναι οι τεχνικές για εντοπισμό θέσης (positioning techniques). Η ακρίβεια με την οποία μπορεί να εντοπιστεί ένας κινητός χρήστης είναι αυτή που θα καθορίσει την ποιότητα των παρεχόμενων υπηρεσιών. Από την άλλη πλευρά, το κόστος εφαρμογής αυτών των τεχνικών είναι ο παράγοντας που θα καθορίσει το πότε και σε τι βαθμό θα αρχίσουν να χρησιμοποιούνται ευρέως οι LBS υπηρεσίες, ώστε να αποφέρουν τα προσδοκώμενα κέρδη στους παροχείς υπηρεσιών και την επιθυμητή λειτουργικότητα στους χρήστες.

Οι τεχνικές που θα περιγραφούν παρακάτω ξεκίνησαν να αναπτύσσονται είτε λόγω κυβερνητικών παρεμβάσεων (πχ η FCC υποχρέωσε τους παροχείς κινητών επικοινωνιών να υποστηρίξουν εντοπισμό θέσης για τις κλήσεις στο E911 [3]) είτε λόγω εμπορικών αναγκών. Μερικές από αυτές τις τεχνικές αποτελούν ήδη πρότυπα (πχ οι CI+TA και UL-TOA για GSM δίκτυα τυποποιήθηκαν τον Μάιο του 1999), ενώ άλλες προβλέπεται να γίνουν.

Για κάθε τύπο ασύρματου δικτύου έχουν αναπτυχθεί διαφορετικές τεχνικές που εκμεταλλεύονται τα ιδιαίτερα χαρακτηριστικά του. Εμείς θα μελετήσουμε τις κυριότερες τεχνικές που υποστηρίζονται από το GSM και θα αναφέρουμε σύντομα στην κατάσταση που επικρατεί σήμερα για τα ασύρματα τοπικά δίκτυα (WLANs), καθώς αυτά τα δύο είδη δικτύων έχουν το μεγαλύτερο μερίδιο της αγοράς των ασύρματων δικτύων.

Γενικά οι τεχνικές εντοπισμού θέσης διαχωρίζονται σε δύο κατηγορίες: σε αυτές που υποστηρίζονται από το τερματικό του χρήστη (terminal-based ή handset-based) και σε αυτές που υποστηρίζονται από το δίκτυο (network-based). Φυσικά μεγαλύτερο επιχειρηματικό ενδιαφέρον έχουν οι δεύτερες αφού δεν επιβάλλουν μετατροπές στην τερματική συσκευή και άρα μπορούν να εξασφαλίσουν πολύ μεγάλη διείσδυση στην αγορά από την πρώτη κιόλας ημέρα που θα γίνουν διαθέσιμες οι υπηρεσίες. Οι πρώτες όμως (terminal-based) έχουν γενικά μεγαλύτερη ακρίβεια.

#### 1.2.1.1 Εντοπισμός θέσης σε GSM δίκτυα

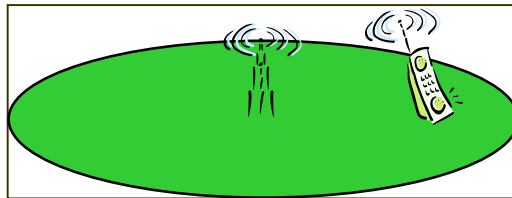
##### 1.2.1.1.1 Τεχνικές βασισμένες στο δίκτυο (network-based techniques)

###### α) Cell ID

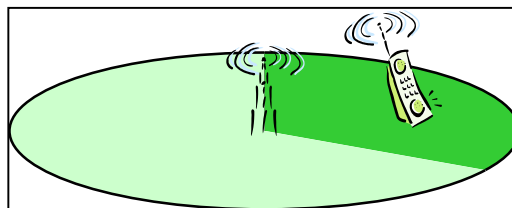
Η μέθοδος αυτή είναι η απλούστερη όσον αφορά στο κόστος εφαρμογής της αλλά και η λιγότερο ακριβής όσον αφορά στην ακρίβεια με την οποία γίνεται ο εντοπισμός της θέσης του κινητού τερματικού. Το κινητό εντοπίζεται από τον σταθμό βάσης με τον οποίο επικοινωνεί (εικόνα 8). Αυτό είναι δυνατό γιατί κάθε σταθμός βάσης, άρα κάθε

κυψέλη, έχει μια μοναδική ταυτότητα (Cell ID). Αυτή η μέθοδος απαιτεί πολύ μικρές αλλαγές στο λογισμικό (software) του δικτύου ενώ δεν απαιτεί καθόλου αλλαγές στον εξοπλισμό (hardware) του. Όμως η ποιότητα που παρέχει δεν είναι κατάλληλη για όλες τις εφαρμογές αφού το κινητό μπορεί να είναι οπουδήποτε μέσα στην κυψέλη, και οι κυψέλες έχουν τυπικές ακτίνες από 200 m – 35km. Έτσι αυτή η μέθοδος μπορεί να εφαρμοστεί καλύτερα σε περιοχές με πολύ μικρές κυψέλες (πυκνοκατοικημένες πόλεις).

Μέχρι τώρα θεωρήσαμε ότι κάθε σταθμός βάσης αποτελείται από μία πανκατευθυντική κεραία (omnidirectional antenna). Αν υπάρχουν πολλές κατευθυντικές κεραίες όμως, μπορούμε να ξέρουμε και με ποια συγκεκριμένη κεραία επικοινωνεί το κινητό. Έτσι μπορούμε να περιορίσουμε την θέση του σε ένα συγκεκριμένο τομέα (sector) της κυψέλης. Αν, για παράδειγμα, υπάρχουν τρεις κεραίες που καθεμία καλύπτει το ένα τρίτο της κυψέλης, τότε αυτή χωρίζεται σε τρεις κυκλικούς τομείς γωνίας  $120^\circ$ , όπως φαίνεται και από την εικόνα 9. Η τεχνική Cell ID αναφέρεται και σαν CGI (Cell Global Identity).



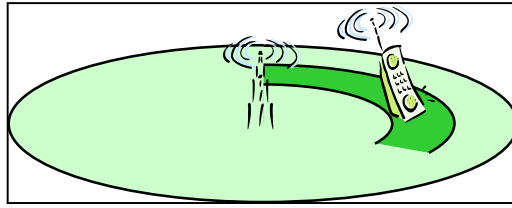
**Εικόνα 8: Η τεχνική Cell ID**



**Εικόνα 9: Cell ID και τομεοποίηση**

## **β) Cell ID + Timing Advance (TA)**

Η μέθοδος αυτή εκτός από το Cell ID κάνει χρήση και της προπορείας χρόνου (Timing Advance, TA) του GSM. Εφόσον η παράμετρος αυτή είναι μέρος των προδιαγραφών του GSM, δεν χρειάζεται καμία αλλαγή στον εξοπλισμό του δικτύου. Το TA είναι μια παράμετρος ελέγχου που χρησιμοποιείται για να μετρηθεί ο χρόνος μιας πλήρους διαδρομής (round trip time) μεταξύ του σταθμού βάσης και του κινητού έτσι ώστε να γίνουν διορθώσεις στον χρονοισμό του κινητού και το σήμα που φτάνει στον σταθμό βάσης να είναι συγχρονισμένο με το τοπικό ρολόι του σταθμού. Η τεχνική αυτή παρέχει λίγο μεγαλύτερη ακρίβεια από την απλή Cell ID.



**Εικόνα 10: Η τεχνική Cell ID και Timing Advance**

### γ) Uplink Time Of Arrival (UL-TOA)

Αυτή είναι άλλη μια τεχνική που στηρίζεται στην μέτρηση χρόνων. Το σήμα του κινητού προς τον σταθμό βάσης (εξ' ου και ο όρος «Uplink») λαμβάνεται από γεωγραφικά απομακρυσμένους δέκτες που ονομάζονται TOA LMUs (Location Measurement Units). Αυτοί οι δέκτες είναι σταθεροί και συνήθως βρίσκονται στο ίδιο μέρος με τους σταθμούς βάσης. Οι χρόνοι άφιξης του σήματος στους LMUs μετρούνται, και με την μέθοδο της τριγωνοποίησης (triangulation) υπολογίζεται η θέση του κινητού (εικόνα 11). Για να λειτουργήσει αυτή η μέθοδος πρέπει το δίκτυο να «αναγκάσει» το κινητό να κάνει μια αίτηση μεταπομπής. Αυτές οι περιπτώσεις κατά τα άλλα αιτήσεις όμως, προκαλούν επιπρόσθετη κίνηση και παρεμβολές στο δίκτυο (κάθε κινητό πρέπει να επικοινωνήσει με τρεις τουλάχιστον LMUs).

Ο χρόνος άφιξης δίνεται από τον παρακάτω τύπο:

$$TOA = TL + TP + TT + TE$$

TL	χρόνος αναφοράς του LMU
TP	καθυστέρηση διάδοσης από κινητό μέχρι LMU
TT	χρονική στιγμή εκπομπής
TE	σφάλματα στην μέτρηση (θόρυβος, παρεμβολή, φαινόμενα multipath κα.)

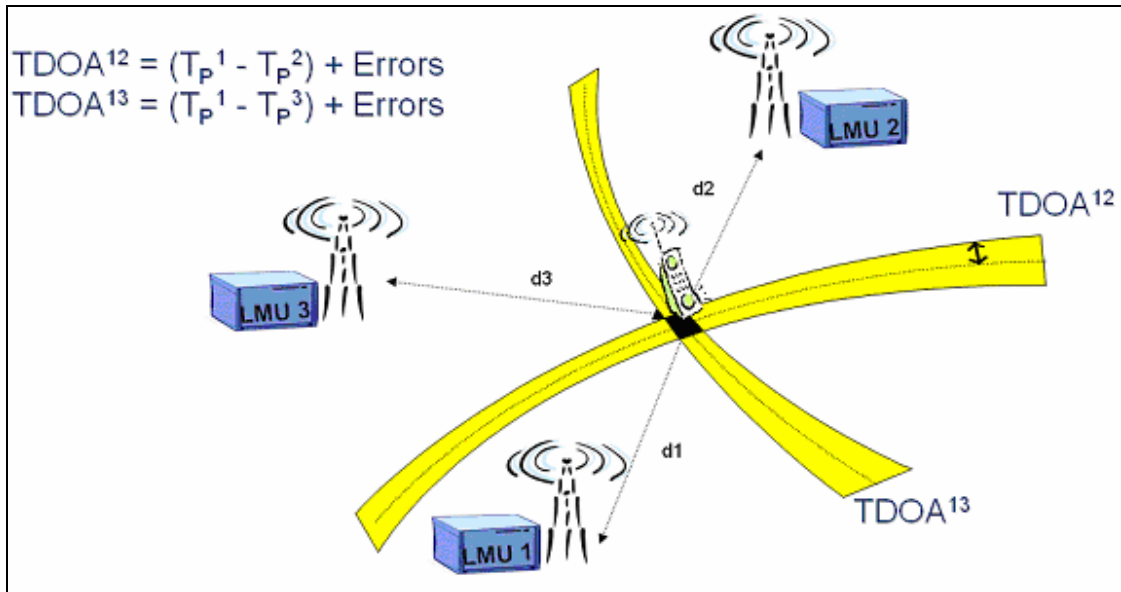
Οι χρόνοι άφιξης από δύο LMUs συνδυάζονται για να δώσουν την διαφορά χρόνου αφίξεων (Time Deference Of Arrival ή TDOA). Έτσι έχουμε:

$$TDOA_{12} = TOA_1 - TOA_2 = (TL_1 - TL_2) + (TP_1 - TP_2) + (TT_1 - TT_2) + (TE_1 - TE_2)$$

Αν όλοι οι LMU έχουν ένα κοινό ρολόι αναφοράς (πχ με σήμα χρονισμού από GPS) τότε η διαφορά  $TL_1 - TL_2$  απαλείφεται. Ο συνδυασμός δύο TDOA μας δίνει την θέση του κινητού με κάποιο περιθώριο λάθους, όπως φαίνεται στην εικόνα 11.

Η τεχνική αυτή προσφέρει αρκετά μεγάλη ακρίβεια αφού για το 67% των περιπτώσεων μπορεί να προσδιορίζει την θέση του κινητού με περιθώριο λάθους λιγότερο από 125 μέτρα. Ένα άλλο μειονέκτημα αυτής της τεχνικής (εκτός από την πλεονάζουσα κυκλοφορία στο δίκτυο) είναι η μεγάλη πολυπλοκότητα και το υψηλό κόστος των LMUs καθώς και το γεγονός ότι απαιτείται από ένας LMU σε κάθε σταθμό βάσης.

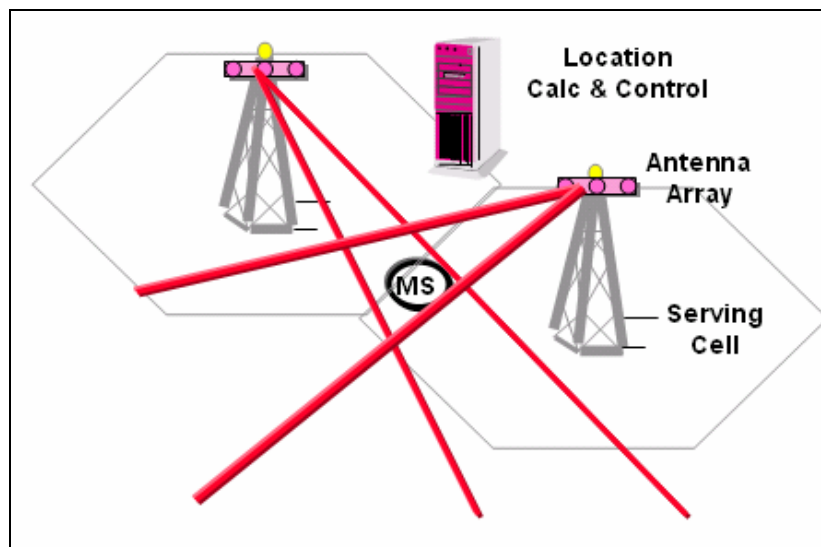




Εικόνα 11: Η τεχνική Uplink Time of Arrival

δ) **Angle Of Arrival (AOA)**

Η τεχνική αυτή βασίζεται στον γεωμετρικό υπολογισμό της θέσης του κινητού, δεδομένης της γωνίας υπό την οποία φτάνει το σήμα σε τουλάχιστον δύο σταθμούς βάσης. Για να μπορέσει κάθε σταθμός βάσης να γνωρίζει την γωνία υπό την οποία έφτασε σε αυτόν το σήμα του κινητού, πρέπει να είναι εξοπλισμένος με μια συστοιχία κατευθυντικών κεραιών (antenna array). Ξέροντας ποια κεραία εξυπηρέτησε το κινητό και ποιο κυκλικό τομέα καλύπτει κάθε κεραία, μπορούμε να υπολογίσουμε κάποια όρια όσον αφορά την θέση του κινητού σταθμού (MS). Φυσικά αυτές οι κεραιές αυξάνουν το κόστος του δικτύου.



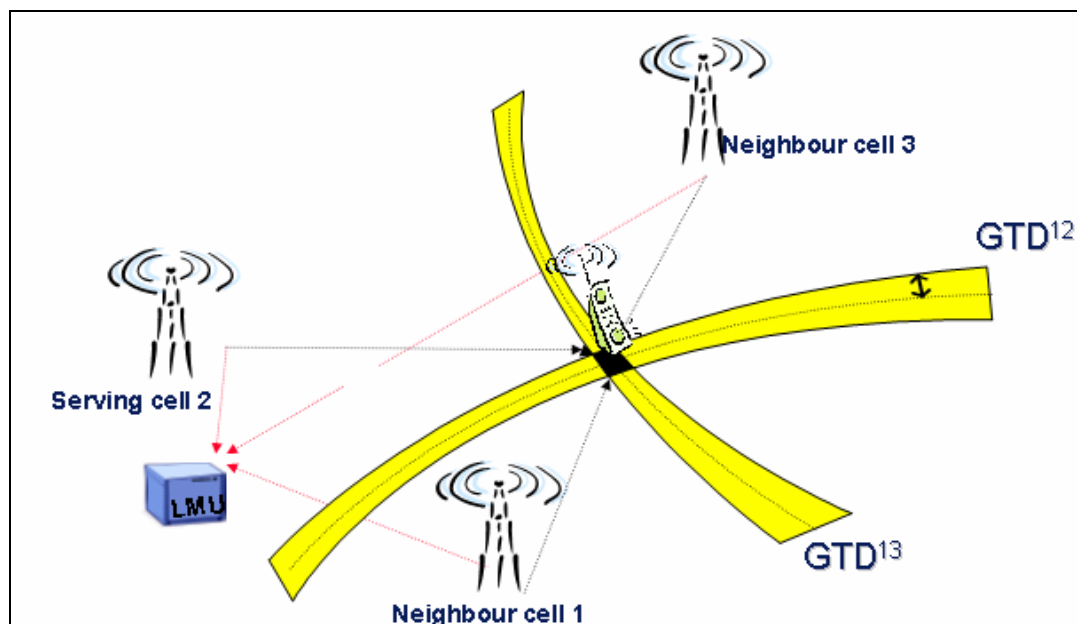
Εικόνα 12: Η τεχνική Angle of Arrival

Όπως φαίνεται και από το παραπάνω σχήμα, η θέση του κινητού είναι προσεγγιστικά το τραπέζιο που σχηματίζεται από την τομή των δύο γωνιών. Αν χρησιμοποιήσουμε και επιπλέον πληροφορία, όπως ότι το κινητό εξυπηρετείται από τον δεξιό σταθμό βάσης, μπορούμε να μειώσουμε ακόμα περισσότερο την αβεβαιότητα για την ακριβή θέση του κινητού.

### 1.2.1.1.2 Τεχνικές βασισμένες στη κινητή συσκευή (terminal-based techniques)

#### α) Enhanced Observed Time Difference (E-OTD)

Η τεχνική αυτή, ομοίως με την UL-TOA, στηρίζεται στις μετρήσεις χρόνων και στο γεγονός ότι ο χρόνος διάδοσης ενός σήματος είναι ανάλογος της απόστασης που διανύει το σήμα. Η διαφορά από την UL-TOA είναι ότι ο υπολογισμός της χρονικής διαφοράς άφιξης (TDOA), ή όπως συνηθίζεται να λέγεται της «παρατηρούμενης χρονικής διαφοράς» (Observed Time Difference ή OTD), γίνεται στο κινητό με ειδικό λογισμικό που πρέπει να είναι εγκατεστημένο σε αυτό. Η OTD είναι η χρονική διαφορά μεταξύ καταιγισμών (bursts) που εκπέμπονται από το σταθμό βάσης ο οποίος εξυπηρετεί το κινητό (serving base station) και από τουλάχιστον άλλους δύο γειτονικούς σταθμούς βάσης. Οι καταιγισμοί αυτοί είναι σήματα που εκπέμπονται από το BCCH (Broadcast Control Channel) κανάλι των σταθμών βάσης. Οι σταθμοί βάσης πρέπει να είναι συγχρονισμένοι μεταξύ τους, αλλιώς απαιτούνται LMUs όπως και στην UL-TOA μέθοδο.



Εικόνα 13: Η τεχνική E-OTD

Η βασική εξίσωση της τεχνικής αυτής είναι η:

$$GTD = OTD - RTD$$

όπου:

GTD (Geometric Time Difference) είναι η χρονική διαφορά που οφείλεται στους διαφορετικούς χρόνους μετάδοσης μεταξύ του κινητού και των σταθμών βάσης

OTD είναι η χρονική διαφορά που μετρήθηκε από το κινητό, μεταξύ των καταιγισμών από δύο σταθμούς βάσης.

RTD (Real Time Difference) είναι η διαφορά συγχρονισμού μεταξύ των σταθμών βάσης, δηλαδή η σχετική διαφορά μεταξύ των χρόνων εκπομπής των καταιγισμών

Η GTD είναι αυτή που περιέχει την πραγματική πληροφορία για την θέση του κινητού αφού ισχύει:

$$GTD = [d(MS,BTS1) - d(MS,BTS2)] / c$$

όπου  $c$  η ταχύτητα του φωτός και  $d(MS,BTSx)$  η απόσταση μεταξύ του κινητού σταθμού (MS) και του σταθμού βάσης X (BTSx).

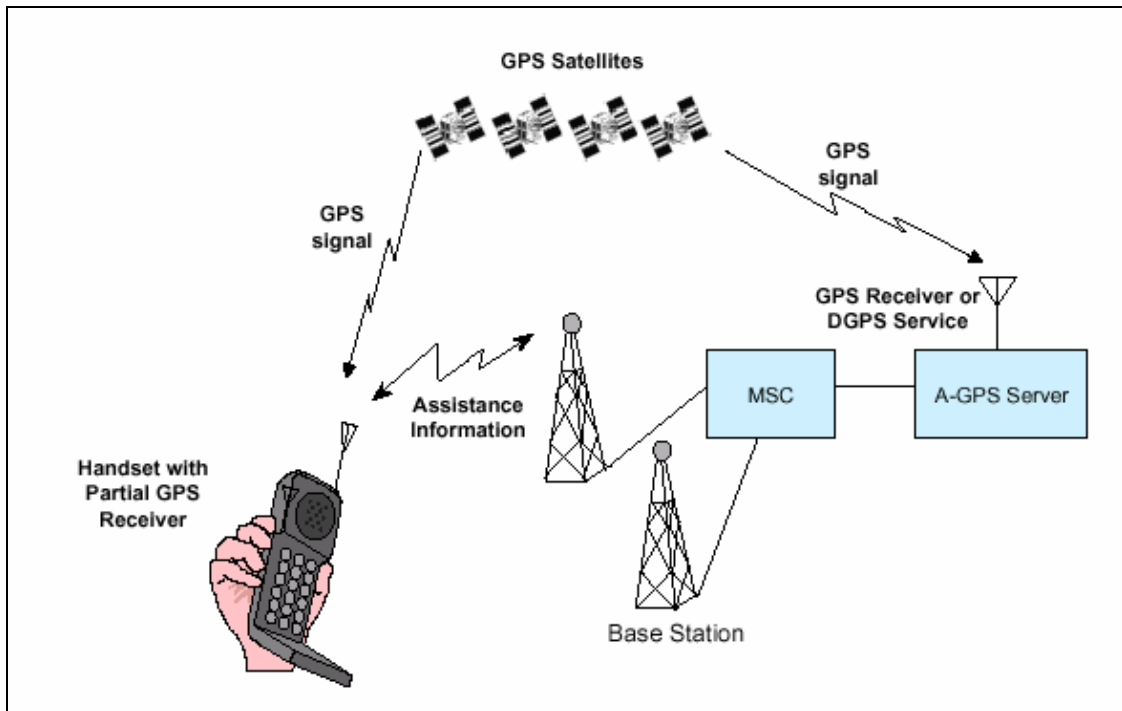
Η παραπάνω σχέση μας δίνει μια υπερβολή. Έτσι ξέροντας τουλάχιστον δύο GTDs μπορούμε να προσδιορίσουμε την θέση του κινητού (εικόνα 13). Στην πράξη όμως χρειάζονται περισσότεροι γειτονικοί σταθμοί βάσης για να επιτύχουμε μια ικανοποιητική ακρίβεια, κάτι που μπορεί να μην είναι δυνατό για μια αραιοκατοικημένη αγροτική (rural) περιοχή.

## β) Assisted GPS (A-GPS)

Το GPS (Global Positioning System) είναι σήμερα το πιο ακριβές σύστημα πλοήγησης και εντοπισμού θέσης. Η λειτουργία του είναι απλή: Ο ειδικός δέκτης λαμβάνει σήμα από 4 δορυφόρους και υπολογίζοντας τον χρόνο που έκανε να φτάσει το σήμα του καθενός, υπολογίζει την απόστασή του από τον κάθε δορυφόρο και άρα την ακριβή του θέση στις τρεις διαστάσεις. Το σύστημα αυτό έχει γενικά καλές επιδόσεις, έχει όμως και κάποια προβλήματα. Επειδή το σήμα από τους δορυφόρους είναι αδύναμο, δεν πρέπει να υπάρχουν εμπόδια ανάμεσα στον δέκτη και στους δορυφόρους. Έτσι αν ο δέκτης βρίσκεται κάτω από φυλλώματα δέντρων ή μέσα σε κτίρια ή δίπλα σε πολύ υψηλά κτίρια (urban canyons), η ακρίβεια του συστήματος μειώνεται αρκετά.

Την λύση σε αυτό το πρόβλημα έρχεται να δώσει το A-GPS. Σε αυτό το σύστημα, ο GPS δέκτης βρίσκεται σε ένα κινητό τερματικό σταθμό κάποιου άλλου τύπου ασύρματου δικτύου (τυπικά σε μια συσκευή κυψελωτού δικτύου). Το ασύρματο δίκτυο έχει και αυτό GPS δέκτες σε σταθερές τοποθεσίες (τυπικά κάθε 300 περίπου χιλιόμετρα). Οι δέκτες αυτοί (A-GPS location servers) παρέχουν στο κινητό κάποιες επιπλέον πληροφορίες που το βοηθούν να προσδιορίσει με μεγαλύτερη ακρίβεια την θέση του. Τέτοιες πληροφορίες μπορεί να είναι είτε διορθώσεις που αφορούν στη μετατόπιση Doppler λόγω της μετακίνησης των δορυφόρων, την οποία δεν μπορεί να αντληθεί το κινητό, είτε πληροφορίες για την διαμόρφωση του σήματος από τον δορυφόρο, οι οποίες

προκύπτουν από κάποιες προβλέψεις που είναι σε θέση να κάνει ο location server. Ο υπολογισμός της τελικής θέσης του κινητού μπορεί να γίνει είτε στο ίδιο, είτε στον location server.



**Εικόνα 14: Η τεχνική A-GPS**

Εκτός από μεγάλη ακρίβεια (της τάξης των λίγων δεκάδων μέτρων, ακόμα και σε εσωτερικούς χώρους), το A-GPS δίνει την δυνατότητα στο κινητό να αποδιαμορφώνει και σήματα πολύ μικρής ισχύος (πχ που έχουν υποστεί περισσότερη από 20dB εξασθένιση). Επίσης ο τελικός υπολογισμός θέσης είναι πολύ γρήγορος (100-1000 φορές γρηγορότερος από το συμβατικό GPS). Όμως προϋποθέτει ειδικό εξοπλισμό τόσο στο δίκτυο (location servers), όσο στην κινητή συσκευή (GPS δέκτη), κάτι που δυσκολεύει την άμεση διείσδυσή του στην αγορά για υπηρεσίες προστιθέμενης αξίας. Έτσι σήμερα η χρήση του περιορίζεται κυρίως στις υπηρεσίες έκτακτης ανάγκης (emergency services).

### **1.2.1.2 Εντοπισμός θέσης σε ασύρματα τοπικά δίκτυα**

Τα δίκτυα κινητών επικοινωνιών καλύπτουν το μεγαλύτερο μέρος των χωρών στις οποίες είναι εγκατεστημένα, όμως η ακρίβεια που παρέχουν δεν είναι πάντα η απαιτούμενη για κάποιες υπηρεσίες (πχ ο εντοπισμός γιατρών ή ασθενών μέσα σε ένα νοσοκομείο). Η σημαντικότερη αιτία που δεν επιτρέπει στα κυψελωτά δίκτυα και στα παραδοσιακά GPS συστήματα να παρέχουν υψηλής ακρίβειας εντοπισμό θέσης σε εσωτερικούς χώρους είναι τα φαινόμενα διάδοσης μέσω πολλαπλών διαδρομών που παρατηρούνται (multipath propagation). Για τέτοιες υπηρεσίες, που απαιτούν εντοπισμό

θέσης σε εσωτερικούς χώρους (indoor positioning) με μεγάλη ακρίβεια, μπορούν να χρησιμοποιηθούν άλλοι τύποι δικτύων όπως είναι τα ασύρματα τοπικά δίκτυα με κυριότερους αντιπροσώπους τα IEEE 802.11 και HIPERLAN/2 και τα προσωπικά δίκτυα (Personal Area Networks ή PANs) όπως το Bluetooth.

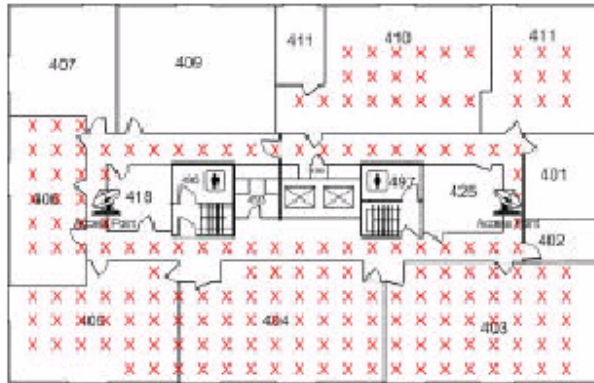
Τα WLANs γνωρίζουν τα τελευταία χρόνια μια ραγδαία τεχνολογική εξέλιξη που συνοδεύεται και από αντίστοιχα μεγάλη ζήτηση στην αγορά δικτύων. Έτσι μπορούμε να χρησιμοποιήσουμε τα πολλά ήδη εγκατεστημένα WLANs ή να εγκαταστήσουμε νέα με μικρό σχετικά κόστος, για παροχή υπηρεσιών εξαρτώμενων από τη θέση.

Στα WLANs μπορούν να εφαρμοστούν μερικές από τις τεχνικές εντοπισμού θέσης που είδαμε ότι εφαρμόζονται και στα κυβελωτά δίκτυα, αλλά υλοποιημένες διαφορετικά για κάθε τύπο WLAN ανάλογα με τις τεχνικές πρόσβασης (access mechanisms) που καθένα υποστηρίζει στο MAC επίπεδό του. Έτσι για παράδειγμα υπάρχουν προτάσεις για τεχνικές TOA (και γενικότερα για τεχνικές που βασίζονται στην εξάρτηση του χρόνου διάδοσης από την απόσταση) σε 802.11 δίκτυα. Η εφαρμογή όμως τεχνικών του τύπου AOA είναι πολύ πιο δύσκολη αφού σε εσωτερικούς χώρους δεν υπάρχει πάντα οπτική επαφή με το σημείο πρόσβασης (line of sight). Στη συνέχεια θα περιγράψουμε μια τεχνική που στηρίζεται στην ισχύ του σήματος που λαμβάνει κάθε κινητός χρήστης.

#### **1.2.1.2.1 RSS-based Location Fingerprinting**

Η μέθοδος αυτή έχει ήδη εφαρμοστεί και μπορεί κανείς να βρει λεπτομερείς αναλύσεις σχετικά με αυτήν [4]. Ένα από τα σημαντικότερα πλεονεκτήματά της είναι ότι δεν απαιτεί πολλή επιπλέον δικτυακή υποδομή, συγκρατώντας έτσι το κόστος υλοποίησης της σε χαμηλά επίπεδα. Ακολουθεί η περιγραφή της λειτουργίας της.

Όταν εγκαταστήσουμε το δίκτυο, μετράμε το RSS (Received Signal Strength) ενός κινητού τερματικού το οποίο τοποθετείται σε συγκεκριμένες θέσεις που απέχουν συγκεκριμένο διάστημα η μια από την άλλη. Οι μετρήσεις γίνονται σε τρία ή και περισσότερα σημεία πρόσβασης (Access Points ή APs) και αποθηκεύονται σε μια τυπική βάση δεδομένων. Έτσι δημιουργείτε ένα πλέγμα από σημεία του εσωτερικού χώρου που μας ενδιαφέρει (τα κόκκινα σημάδια στην εικόνα 15). Επίσης παίρνουμε μετρήσεις και για διαφορετικές ώρες της μέρας, αφού η παρουσία ανθρώπων επηρεάζει το RSS. Τελικά έχουμε δημιουργήσει μια βάση με «αποτυπώματα» (fingerprints) της RSS τιμής στο πεδίο του χώρου.



**Εικόνα 15: Το πλέγμα που προκύπτει από τις μετρούμενες RSS τιμές**

Όταν θέλουμε να υπολογίσουμε την θέση ενός κινητού, μετράμε την τιμή του RSS σε τρία (ή περισσότερα) APs και συγκρίνουμε την τιμή αυτή με τις τιμές στην βάση δεδομένων. Η θέση του κινητού τότε είναι: είτε η θέση του «αποτυπώματος» στην βάση το οποίο έχει την μικρότερη διαφορά από το μετρημένο «αποτύπωμα», είτε η θέση που προκύπτει αν διαλέξουμε τα  $k$  «αποτυπώματα» με την μικρότερη διαφορά από το μετρημένο και υπολογίσουμε μια μέση τιμή των συντεταγμένων τους

Η ακρίβεια που μας δίνει αυτή η τεχνική εξαρτάται κυρίως από το πόσο απέχουν τα αρχικά RSS «αποτυπώματα» που μετρήσαμε. Συνήθως το περιθώριο λάθους είναι μέχρι 2 μέτρα.. Παρόμοια συστήματα που έχουν υλοποιηθεί είναι το RADAR [5] και το Ekahau [6].

### 1.2.1.3 Σχόλια στις τεχνικές εντοπισμού θέσης

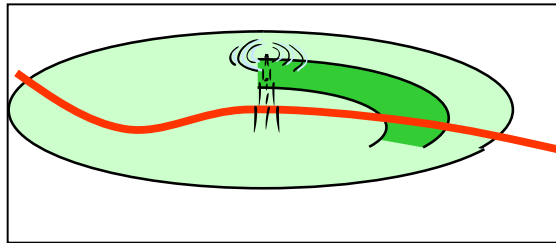
Ένας συγκεντρωτικός πίνακας που παρουσιάζει τυπικές επιδόσεις των τεχνικών που προαναφέρθηκαν είναι ο παρακάτω (οι τιμές αναφέρονται σε αστικό περιβάλλον):

Τεχνική	Ακρίβεια (μέτρα)
Cell ID + TA	50-500
TOA	40-150
E-OTD	50-125
AOA	~ 40
A-GPS	5-50
RSS-Location Fingerprinting	1-2

**Πίνακας 1: Ακρίβεια των κυριότερων μεθόδων εντοπισμού**

Σε πραγματικά συστήματα πολλές φορές χρησιμοποιούνται συνδυασμοί των παραπάνω τεχνικών, ώστε να επιτευχθεί ακόμη μεγαλύτερη ακρίβεια. Επίσης οι συντεταγμένες που υπολογίζονται από τις παραπάνω μεθόδους μπορούν να συνδυαστούν και με στοιχεία από βάσεις γεωγραφικών δεδομένων έτσι ώστε να αποκλειστούν ή να προβλεφθούν κάποιες θέσεις. Για παράδειγμα αν ξέρουμε την θέση ενός κινητού σε μια αγροτική περιοχή με περιθώριο λάθους 200 μέτρα και ξέρουμε ότι στην συγκεκριμένη

περιοχή βρίσκεται μόνο ένας μεγάλος δρόμος στον οποίο έχουν πρόσβαση οι άνθρωποι, μπορούμε με σχετική ασφάλεια να αποκλείσουμε τις θέσεις που δεν ανήκουν στο δρόμο (εικόνα 16).



**Εικόνα 16: Συνδυασμός πληροφορίας εντοπισμού με τοπικά γεωγραφικά δεδομένα για βελτίωση της ακρίβειας**

Η προτυποποίηση (standardisation) των τεχνικών εντοπισμού θέσης έχει αρχίσει για τα κυψελωτά δίκτυα. Για τα δίκτυα GSM έχουν γίνει πρότυπα οι τεχνικές TOA, E-OTD, AOA και A-GPS (στις ΗΠΑ μάλιστα, ο FCC έχει υποχρεώσει τους παροχείς κινητών επικοινωνιών να τα υποστηρίξουν). Επίσης προβλέπονται και τεχνικές για τα δίκτυα 3<sup>ης</sup> γενιάς (A-GPS, TDOA). Περισσότερες πληροφορίες μπορούν να βρεθούν στην αναφορά [7]. Η κατάσταση όμως δεν είναι τόσο καλή για τα τοπικά ασύρματα δίκτυα όπου προς το παρόν επικρατούν οι λύσεις που προτείνονται από ιδιωτικές εταιρίες (proprietary solutions). Αυτό φυσικά καθυστερεί την εξέλιξη των υπηρεσιών που εξαρτώνται από την θέση.

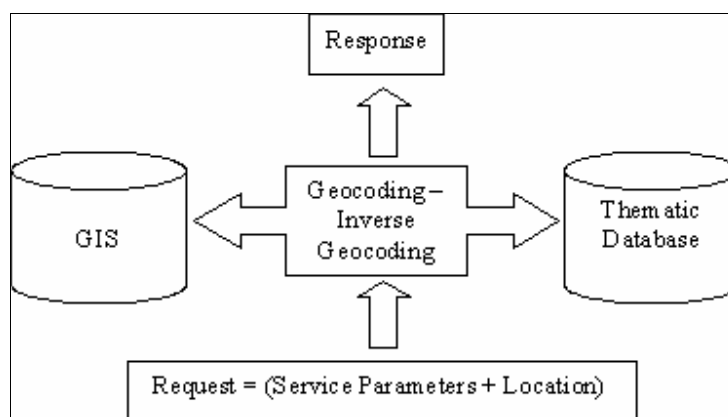
Το μοντέλο αυτόματης προώθησης που μελετάμε δεν είναι εντελώς ανεξάρτητο από την τεχνική εντοπισμού θέσης την οποία χρησιμοποιεί το ασύρματο δίκτυο. Πιο συγκεκριμένα, ο διαρκής έλεγχος της θέσης του κινητού χρήστη, που γίνεται για να αποφασιστεί πότε θα σκανδαλισθεί (trigger) η παρεχόμενη υπηρεσία, είναι κάτι που από την φύση του καταναλώνει αρκετούς πόρους (υπολογιστική ισχύ, χωρητικότητα δικτύου). Η κατάσταση μπορεί να γίνει αρκετά χειρότερη αν το ίδιο το κινητό υπολογίζει την θέση του. Τότε θα πρέπει σε τακτά χρονικά διαστήματα να κάνει αυτόν τον υπολογισμό (χρήση των υπολογιστικών πόρων της τερματικής συσκευής) και να την αναφέρει σε έναν κεντρικό σταθμό (χρήση των ραδιοπόρων της ασύρματης ζεύξης). Το τελευταίο είναι ιδιαίτερα σημαντικό αφού το ασύρματο δίκτυο αποκτά μεγαλύτερο φόρτο, ο οποίος μάλιστα είναι και αρκετά δύσκολο να μοντελοποιηθεί και να αναλυθεί στατιστικά, σε αντίθεση με άλλα είδη φόρτων όπως αυτός των αναμενόμενων τηλεφωνικών κλήσεων ανά περιοχή ή ανά ώρα. Άρα οι τεχνικές που βασίζονται στο δίκτυο είναι καταλληλότερες για υπηρεσίες που ανήκουν στο μοντέλο αυτόματης προώθησης.

Η τρίτη γενιά ασύρματων δικτύων κινητής τηλεφωνίας (UMTS) έχει την δυνατότητα να υποστηρίξει ακόμα πιο ακριβείς τεχνικές (βασισμένες στην μέθοδο WCDMA) καθώς και τεχνικές που μοιάζουν με αυτές του GSM.

## 1.2.2 Το Περιεχόμενο στις Εξαρτώμενες από τη Θέση Υπηρεσίες

Η δεύτερη μεγάλη διαφορά των υπηρεσιών που εξαρτώνται από την θέση σε σχέση με τις άλλες υπηρεσίες κινητών επικοινωνιών είναι οι αυξημένες ανάγκες σε περιεχόμενο (content) – η πρώτη διαφορά είναι φυσικά η εξάρτηση από την θέση του χρήστη. Οι υπόλοιπες υπηρεσίες χρησιμοποιούν στατική πληροφορία την οποία συνήθως μπορεί να διαθέσει ο ίδιος ο παροχέας του δικτύου. Στις υπηρεσίες που μελετάμε όμως, οι πληροφορίες που φθάνουν στον χρήστη πρέπει να δημιουργηθούν δυναμικά ανάλογα με την θέση του. Έτσι υπάρχουν πλέον και άλλοι «παίκτες» που παρέχουν εξειδικευμένο περιεχόμενο. Τα δεδομένα τους προέρχονται είτε από Γεωγραφικά Συστήματα Πληροφοριών (GIS), είτε από απλές βάσεις δεδομένων, οι οποίες όμως πρέπει απαραίτητα να περιέχουν και κάποια γεωγραφική πληροφορία για να είναι αξιοποιήσιμες από χωρικά εξαρτώμενες υπηρεσίες.

Όπως φαίνεται και από το παρακάτω σχήμα, τα GIS συστήματα συνεργάζονται με άλλες ανεξάρτητες βάσεις. Η συνεργασία τους αυτή γίνεται κυρίως στα πλαίσια της Γεωκωδικοποίησης (Geocoding) ή της Αντίστροφης Γεωκωδικοποίησης (Inverse Geocoding). Επειδή αυτές οι διαδικασίες είναι πολύ χρήσιμες σε τέτοιου είδους υπηρεσίες, ακολουθεί μια πιο αναλυτική περιγραφή (θα περιγραφεί μόνο η Γεωκωδικοποίηση καθώς η άλλη διαδικασία είναι απλώς η αντίστροφή της).



Εικόνα 17: Σχέσεις μεταξύ των οντοτήτων παροχής περιεχομένου

### 1.2.2.1 Γεωκωδικοποίηση

Με τον όρο αυτό αναφερόμαστε στην διαδικασία κατά την οποία συνδυάζονται εγγραφές από δύο βάσεις δεδομένων: μια βάση που περιέχει εγγραφές με δεδομένα και διευθύνσεις (χωρίς πληροφορία για ακριβή τοποθεσία) και μια άλλη που περιέχει χάρτες ή, πιο γενικά, συντεταγμένες διευθύνσεων. Έτσι η δεύτερη βάση (που συνήθως ανήκει σε ένα GIS) χρησιμοποιείται σαν σύστημα αναφοράς ενώ η πρώτη περιέχει τα πραγματικά δεδομένα που ενδιαφέρουν τον χρήστη (θεματική βάση ή βάση χαρακτηριστικών). Το



λογισμικό που χρησιμοποιείται στη Γεωκωδικοποίηση συνδέει εγγραφές από τις δύο βάσεις με βάση τα κοινά ονόματα δρόμων ή τις κοινές διευθύνσεις. Το τελικό αποτέλεσμα είναι ότι τα δεδομένα της πρώτης βάσης αποκτούν και ακριβείς συντεταγμένες που συνήθως είναι το γεωγραφικό πλάτος και μήκος. Κατόπιν αυτές μπορεί να χρησιμοποιηθούν για περαιτέρω αναλύσεις που πιθανόν να απαιτούνται από την εφαρμογή.

Για παράδειγμα, πριν από την Γεωκωδικοποίηση μπορεί να έχουμε την παρακάτω εγγραφή:

Όνομα	Διεύθυνση	Ταχ. Κωδ.	X συντεταγμένη	Y συντεταγμένη
<b>Κώστας Μαχαίρας</b>	<b>Παπάγου 4</b>	<b>163 44</b>	<b>Κενό</b>	<b>Κενό</b>

Μετά την διαδικασία που περιγράφηκε θα έχουμε την παρακάτω εγγραφή:

Όνομα	Διεύθυνση	Ταχ. Κωδ.	X συντεταγμένη	Y συντεταγμένη
<b>Κώστας Μαχαίρας</b>	<b>Παπάγου 4</b>	<b>163 44</b>	<b>-122.345</b>	<b>47.234</b>

Όλη αυτή η διαδικασία μπορεί να γίνει με βάση διευθύνσεις ή ταχυδρομικούς κώδικες. Στην περίπτωση των υπηρεσιών που εξαρτώνται από την θέση, ίσως είναι προτιμότερη η δεύτερη αφού και η ακρίβεια των τεχνικών εντοπισμού θέσης δεν είναι τόσο μεγάλη σήμερα ώστε να μπορεί να γίνει διάκριση σε επίπεδο διεύθυνσης, δηλαδή σε λίγα μόνο μέτρα. Τέλος στις υπηρεσίες αυτές προφανώς μεγαλύτερη χρησιμότητα έχει η αντίστροφη διαδικασία αφού στα δεδομένα που έχει στην διάθεσή του το σύστημα, συμπεριλαμβάνεται η θέση του κινητού τερματικού και ζητούνται πληροφορίες που προϋποθέτουν την εύρεση της διεύθυνσής του.

### 1.2.2.2 Γεωγραφικά Συστήματα Πληροφοριών (GIS)

Τα GIS (Geographic Information System) είναι συστήματα αποθήκευσης, ανάλυσης και αναπαράστασης γεωγραφικών δεδομένων. Ένας πιο ακριβής ορισμός μπορεί να προκύψει αν αναλύσουμε ετυμολογικά την ονομασία τους [8]:

1. **Γεωγραφικό:** Το σύστημα αναφέρεται σε στοιχεία που σχετίζονται με την γεωγραφική κλίμακα και αναφέρονται με κάποιο σύστημα συντεταγμένων σε θέσεις στην επιφάνεια της γης. Επομένως, οι χωρικές οντότητες και η γεωγραφική θέση τους αποτελούν τον θεμέλιο λίθο του συστήματος.
2. **Σύστημα:** Είναι ένα περιβάλλον που επιτρέπει την διαχείριση των στοιχείων (εισαγωγή, ανάλυση, και απόδοση) καθώς και την αναζήτηση απαντήσεων σε ερωτήσεις που τίθενται.
3. **Πληροφοριών:** Το σύστημα χρησιμοποιείται για να θέσει ερωτήσεις για τα στοιχεία της γεωγραφικής βάσης, λαμβάνοντας πληροφορίες για τον γεωγραφικό κόσμο. Αυτό αντιπροσωπεύει την γνωστή διαδικασία από στοιχεία σε πληροφορία.

Στο μεγαλύτερο διάστημα της ιστορίας των GIS, τέσσερις ήταν οι κύριοι στόχοι που καθόριζαν το περιεχόμενο των γεωγραφικών στοιχείων:

- ο φυσικός χώρος
- η ιδιοκτησία
- ο πληθυσμός
- η παραγωγή

Με την πάροδο του χρόνου, αυτοί απέκτησαν μια πολυπλοκότητα και πολυσυνθετότητα. Αυτό ισχύει γιατί σήμερα τα GIS χρησιμοποιούνται και σε εφαρμογές γενικής χρήσης και όχι μόνο από ειδικούς (πχ Τοπογράφους). Ένα τέτοιο λαμπρό παράδειγμα αποτελούν οι εφαρμογές που ανέπτυξε το Seattle [9]. Αυτά είναι υπηρεσίες οι οποίες παρέχονται κυρίως στις δημοτικές και κυβερνητικές αρχές, αλλά και στους πολίτες και επαγγελματίες, του Seattle και που στηρίζονται σε ένα GIS που έχει αρχίσει να δημιουργείται εδώ και πολλά χρόνια και συλλέγει σε μια κεντρική βάση δεδομένων όλες τις απαιτούμενες πληροφορίες. Έτσι γενικά μπορούμε να πούμε ότι τα δεδομένα για τα GIS, αναφορικά με το περιεχόμενό τους, διακρίνονται σε τέσσερις κατηγορίες: στοιχεία που αφορούν το φυσικό χώρο, τους ανθρώπους( πληθυσμό κλπ), τις οικονομικές δομές και τις κοινωνικές δομές και διαδικασίες.

Η χρήση των GIS συμπεριλαμβάνει τις εξής διαδικασίες: συλλογή και εισαγωγή δεδομένων, αποθήκευση και διαχείριση στην βάση δεδομένων, μετασχηματισμός στοιχείων και ανάλυση στοιχείων ή/και χώρου, αναζήτηση στοιχείων, παρουσίαση των αποτελεσμάτων της αναζήτησης. Παρακάτω θα αναφερθούμε σε καθεμιά από αυτές τις διαδικασίες ξεχωριστά (σε άλλες συνοπτικά και σε άλλες πιο αναλυτικά).

#### **1.2.2.2.1 Εισαγωγή Στοιχείων**

Για την εισαγωγή των γεωγραφικών στοιχείων χρησιμοποιούνται συνήθως έτοιμοι χάρτες όπου με διάφορες μεθόδους (άλλες αυτοματοποιημένες και άλλες όχι) μετατρέπονται σε ψηφιακή μορφή κατάλληλη για αποθήκευση σε Η/Υ. Της διαδικασίας αυτής, όπως είναι προφανές, προηγείται η διαδικασία της γεωδαισίας(geodetics), η οποία ασχολείται με την μέτρηση και αποτύπωση των γεωγραφικών οντοτήτων σε τοπολογικούς χάρτες.

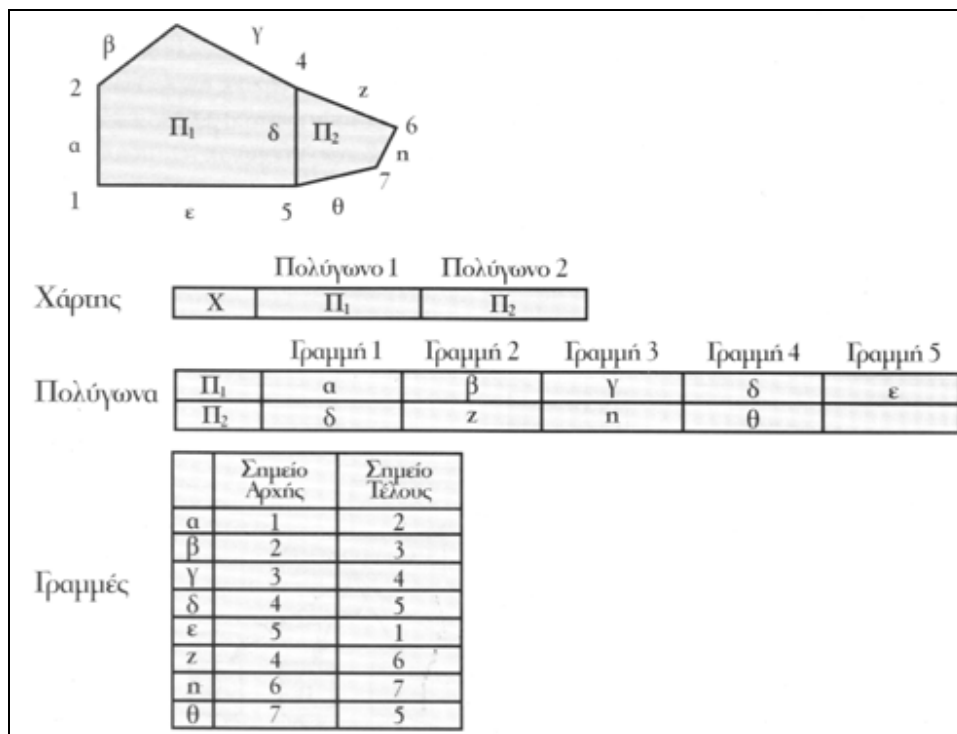
#### **1.2.2.2.2 Δομή Βάσης Δεδομένων**

Η τακτική που ακολουθείται στην αποθήκευση των δεδομένων στα GIS ορίζει ότι σε κάθε γεωγραφικό χάρτη αντιστοιχίζονται ένα ή περισσότερα στρώματα (layers). Κάθε στρώμα αποτελεί ένα αυτόνομο θεματικό χάρτη (εικόνα 20). Βέβαια τα δεδομένα του μπορούν να συνδυαστούν με αυτά οποιουδήποτε άλλου επιπέδου.

Στα GIS μπορούν να χρησιμοποιηθούν όλοι οι βασικοί τύποι βάσεων δεδομένων γιατί ο καθένας έχει τα δικά του πλεονεκτήματα. Έτσι τα ιεραρχικά συστήματα είναι χρήσιμα για την επιμέρους διαίρεση χωρικών δεδομένων σε διαχειρίσιμα θέματα ή περιοχές, με τρόπο ώστε να γίνεται εφικτή η γεωγραφική ανάλυση και χαρτογράφηση συνεχών περιοχών. Η δικτυακή προσέγγιση είναι ιδεώδης για την τοπολογική απόδοση διανυσματικών στοιχείων (σημείων, γραμμών και πολυγώνων). Η σχεσιακή προσέγγιση

είναι κατάλληλη για την ανάκληση οντοτήτων με βάση τα χαρακτηριστικά τους. Η αντικειμενοστραφής σχεδίαση είναι χρήσιμη όταν οι οντότητες έχουν κοινά χαρακτηριστικά ή αλληλεπιδρούν μεταξύ τους [8]. Τέλος τα τελευταία χρόνια έχουν αναπτυχθεί και χωρικές βάσεις δεδομένων, τις οποίες θα δούμε με περισσότερες λεπτομέρειες σε επόμενη ενότητα.

Βέβαια είναι φυσικό ότι η ανάπτυξη που γνώρισαν τα σχεσιακά συστήματα βάσεων δεδομένων (RDBMS) τα τελευταία χρόνια δεν πέρασε απαρατήρητη και από τον χώρο των GIS. Έτσι σήμερα προτιμάται μια σχεσιακή προσέγγιση. Σε αυτήν τα χωρικά δεδομένα καθώς και τα θεματικά χαρακτηριστικά κάθε περιοχής οργανώνονται σε σχεσιακούς πίνακες (εικόνες 18 και 19). Κάθε γεωγραφική εγγραφή σε αυτούς τους πίνακες προσδιορίζεται από ένα χωρικό κλειδί (spatial key). Κάθε θεματική εγγραφή αντιστοιχίζεται με την γεωγραφική πληροφορία μέσω ενός ξένου κλειδιού (foreign key).



Εικόνα 18: Σχεσιακή αναπαράσταση γεωμετρικών δεδομένων [8]

Σημείο			
point id	x-cord	y-cord	
Αλυσίδα			
chain id	point id	seq.no	
Πολύ-γραμμή			
polyline id	chain id		
Πολύγωνο			
polygon id	chain id		
Σημειακό στοιχείο			
pt feature id	point id		
Γραμμικό στοιχείο			
line feature id	polyline id		
Επιφανειακό στοιχείο			
area feature id	polygon id		
Σύνθετο στοιχείο			
comp feature id	pt feature id	line feature id	area feature id

Εικόνα 19: Αποθήκευση γεωμετρικής πληροφορίας σε σχεσιακούς πίνακες [8]

### 1.2.2.2.3 Ανάλυση Χώρου

Η ανάλυση των δεδομένων, όπως είναι λογικό, αποτελεί το κεντρικό λειτουργικό σημείο των GIS συστημάτων. Οι μέθοδοι ανάλυσης που υποστηρίζονται από ένα τέτοιο σύστημα καθώς και η απόδοσή τους, είναι οι κύριοι παράγοντες που θα καθορίσουν την αξία και χρησιμότητά του. Εξάλλου αυτές οι μέθοδοι είναι που δίνουν και την δυνατότητα για περίπλοκες επερωτήσεις και αναζητήσεις σε ένα τέτοιο σύστημα. Οι κυριότερες αναλυτικές διαδικασίες είναι οι εξής:

1. **Επιλογή χαρακτηριστικών:** Χρησιμοποιώντας κάποιες ιδιότητες ενός θεματικού χάρτη μπορούμε να ανακτήσουμε επιπλέον πληροφορίες. Η αναζήτησή τους γίνεται με κριτήρια τύπου Boolean (AND, OR, XOR, NOT) και με αριθμητικές συγκρίσεις (<, >, =, ≠).
2. **Παραγωγή νέων χαρακτηριστικών με ποσοτικές διαδικασίες:** Οι διαδικασίες αυτές μπορεί να είναι είτε αριθμητικές και τριγωνομετρικές πράξεις, είτε στατιστικές μέθοδοι (μέσος όρος, διασπορά κλπ).
3. **Ανάλυση εγγύτητας:** Αυτές οι διαδικασίες αφορούν την δημιουργία μιας περιφέρειας (ενός νέου πολυγώνου), η οποία ορίζεται με βάση την εγγύτητά της σε μια υπάρχουσα οντότητα. Οι πιο διαδεδομένες από αυτές είναι η δημιουργία ζωνών επιρροής (buffer), τα πολύγωνα Thiessen (που δημιουργούνται γύρω από ένα σύνολο σημείων και εκπροσωπούν την περιοχή ευθύνης του καθενός απ’

- αυτά), η εγγύτητα (βρίσκει ποια σημεία ενός στρώματος βρίσκονται πιο κοντά σε κάποια σημεία ενός δεύτερου στρώματος), και η απόσταση από σημείο.
4. **Επικάλυψη:** Είναι ίσως η πιο θεμελιώδης διαδικασία ανάλυσης σε ένα GIS και αυτό γιατί η έννοια των θεματικών στρωμάτων και της επικάλυψής τους αποτελούν την πεμπτούσια των συστημάτων αυτών. Κατά την διαδικασία αυτή συνδυάζονται δύο ή περισσότερα στρώματα και με τη χρήση κάποιου τελεστή της άλγεβρας Boolean (Ένωση, Ταυτότητα, Τομή) παράγεται ένα νέο στρώμα.
  5. **Διαχείριση:** Οι κύριες μέθοδοι διαχείρισης είναι η αφαίρεση (erase), το κόψιμο (clip), ο διαμελισμός (split) και η ενημέρωση (update) των χωρικών δεδομένων στην βάση δεδομένων.
  6. **Ανάλυση Ορίων:** Οι διαδικασίες αυτές αναφέρονται στην επεξεργασία των οντοτήτων ενός θεματικού στρώματος, οι οποίες οδηγούν στην αλλαγή συγκεκριμένων ορίων του στρώματος αυτού. Οι κυριότερες είναι η ένωση γειτονικών χαρτών (append,marjoin), η αφαίρεση επιμηκών πολυγώνων και η αφαίρεση γραμμών. Τα επιμήκη πολύγωνα προκύπτουν κατά τις πράξεις μεταξύ στρωμάτων στα οποία δεν συμπίπτουν οι θέσεις των οντοτήτων λόγω κακής ψηφιοποίησης.

#### 1.2.2.2.4 Επερωτήσεις και Αναζήτηση Δεδομένων

Γενικά, οι επερωτήσεις (queries) στις οποίες ένα GIS μπορεί να απαντήσει κατά την διάρκεια της διαδικασίας της χωρικής ανάλυσης, διακρίνονται σε πέντε χαρακτηριστικές κατηγορίες:

1. **Γεωγραφία:** «Τι υπάρχει στην περιοχή...»  
Η ουσία της επερώτησης βρίσκεται στο γεγονός, ότι οφείλουμε να γνωρίζουμε τι υπάρχει σε κάθε υποσύνολο-χωρική ενότητα της περιοχής μελέτης. Η ταυτότητα ενός τόπου μπορεί να περιγραφεί με διάφορους τρόπους, όπως για παράδειγμα, με το όνομα της τοποθεσίας, κάποιο γεωγραφικό κωδικό ή το γεωγραφικό μήκος και πλάτος του.
2. **Αναζήτηση βάσει κριτηρίων:** «Που βρίσκεται...»  
Η δεύτερη αυτή επερώτηση είναι ουσιαστικά η αντίστροφη της πρώτης και απαιτεί στοιχεία χωρικής ανάλυσης για να απαντηθεί. Αντί της ταυτότητας ενός συγκεκριμένου τόπου, ζητείται να βρεθεί ο γεωμετρικός τόπος μέσα στον οποίο ικανοποιούνται ορισμένες συνθήκες. Ένα τέτοιο παράδειγμα είναι: «που βρίσκεται δασική έκταση με εμβαδόν μεγαλύτερο από 100 στρέμματα που απέχει λιγότερο από 60 χιλ. από το αστικό κέντρο και 2 χιλ. από το οδικό δίκτυο»
3. **Τάσεις:** «Ποια η μεταβολή...»  
Η επερώτηση αυτή προϋποθέτει την απάντηση των δύο πρώτων, καθώς αναζητά τις διαφορές που παρουσιάζονται, λαμβάνοντας υπόψη και το χρονικό πεδίο.
4. **Πρότυπα:** «Από ποια χωρικά πρότυπα χαρακτηρίζεται...»  
Στη συγκεκριμένη περίπτωση αναζητούνται συσχετισμοί και νόμοι οι οποίοι διέπουν φαινόμενα που συμβαίνουν ταυτόχρονα (ή είναι το ένα συνέπεια του άλλου) και αφορούν έναν συγκεκριμένο χώρο.
5. **Διαδικασίες:** «Τι θα συνέβαινε αν...»

Αυτού του είδους οι επερωτήσεις προϋποθέτουν συνολικότερη επιστημονική ανάλυση, αφού δεν αρκεί μόνο η γεωγραφική πληροφορία για να δοθούν απαντήσεις.

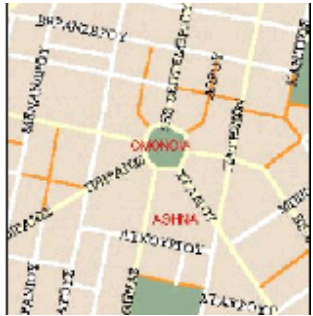


Είναι πολύ σημαντικό οι δημιουργοί των εφαρμογών και υπηρεσιών που χρησιμοποιούν GIS σαν τον κύριο παροχέα περιεχομένου, να γνωρίζουν τις δυνατότητες αναζήτησης δεδομένων που έχουν. Έτσι μπορούν να εκμεταλλευθούν καλύτερα αυτά τα συστήματα και να δημιουργήσουν πιο εξειδικευμένες και προηγμένες εφαρμογές.

#### **1.2.2.2.5 Παρουσίαση Αποτελεσμάτων Αναζήτησης**

Αφού αναλυθούν τα δεδομένα ενός GIS και παραχθεί η απάντηση στην επερώτηση που τέθηκε, πρέπει αυτή να επιστρέψει στον χρήστη του GIS σε μια μορφή που να μπορεί να γίνει αντιληπτή από αυτόν. Η μορφή αυτή εξαρτάται από τον τύπο του χρήστη (υπολογιστικό σύστημα ή άνθρωπος) και έχουν προταθεί πολλές μορφές αναπαράστασης περιεχομένου. Με αυτό το θέμα θα ασχοληθούμε στην επόμενη ενότητα.

#### **1.2.2.3 Τρόποι Αναπαράστασης Περιεχομένου**

Οι βασικοί τρόποι με τους οποίους μπορεί να αναπαρασταθεί το σχετικό με την υπηρεσία περιεχόμενο είναι δύο: είτε ως γραπτό κείμενο, που μπορεί να έχει και την μορφή λιστών ή επιλογέων (menus), είτε σαν εικόνες που μπορεί να αναπαριστούν χάρτες ή άλλα στοιχεία (πχ φωτογραφίες των πιο κοντινών αρχαιολογικών μνημείων) που μπορεί να ενδιαφέρουν τον χρήστη. Αξίζει σε αυτό το σημείο να αναφερθεί ότι εκτός από τις δυο προηγούμενες διεπαφές, έχουν δημιουργηθεί και υπηρεσίες που χρησιμοποιούν φωνή για να απαντήσουν στις αιτήσεις του χρήστη. Έτσι υπάρχουν εφαρμογές φωνητικής πλοήγησης που υλοποιούνται με τεχνολογίες όπως η VoiceXML. Λίγα λόγια για τέτοιες υπηρεσίες θα αναφερθούν στο τέλος της ενότητας. Το ποιος τρόπος θα χρησιμοποιηθεί κάθε φορά εξαρτάται από το είδος της υπηρεσίας, τις πηγές δεδομένων που έχουν στην διάθεσή τους οι δημιουργοί της υπηρεσίας και φυσικά τις ικανότητες της κινητής συσκευής (βλ. και ενότητα 1.2.3). Παρακάτω παρατίθενται παραδείγματα των δύο βασικών τρόπων αναπαράστασης:

Παρεχόμενη υπηρεσία	Αποτέλεσμα που επιστρέφει η υπηρεσία	Εικόνα												
Δείξε μου τον χάρτη της περιοχής γύρω από την Ομόνοια.	Χάρτης													
Θέλω την διαδρομή με λεωφορεία από το σπίτι μου στο πανεπιστήμιο.	Διαδρομή													
Βρες τον φίλο μου με κωδικό M1	Σημείο-Θέση													
Δώσε μου λεπτομερείς οδηγίες για το πως θα πάω από εδώ που βρίσκομαι στην κοντινότερη Εθνική Τράπεζα	Καθοδήγηση	<p>Straight on ΝΕΣΟΓΕΙΩΝ  Turn right onto ΚΥΠΡΙΣΗΝ ΑΓΩΝΙΣΤΩΝ  Turn right onto ΚΥΠΡΙΣΗΝ ΑΓΩΝΙΣΤΩΝ  Straight on ΝΕΣΟΓΕΙΩΝ  Turn left onto ΝΕΣΟΓΕΙΩΝ  Straight on ΚΑΥΑΛΑ ΑΣΠ.  Turn right onto ΚΑΥΑΛΑ ΑΣΠ.  Straight on ΚΑΥΑΛΑ ΑΣΠ.  Stay to the left on ΚΑΥΑΛΑ ΑΣΠ.  Stay to the right on ΚΑΥΑΛΑ ΑΣΠ.  Turn right onto ΤΕΑΤΡΗΣ ΑΜΑΡ.  Straight on ΤΕΑΤΡΗΣ ΑΜΑΡ.</p>												
Δείξε μου το κοντινότερο εστιατόριο και τον τιμοκατάλογό του	Λίστα	<p><b>CHICKENLAND, TROIAS 5 St.</b></p> <table border="1"> <tbody> <tr> <td>BOWL CHILI</td> <td>\$ 3.25</td> </tr> <tr> <td>BOWL CHILI AND BEANS</td> <td>\$ 2.75</td> </tr> <tr> <td>BOWL GREEN CHILE STEW</td> <td>\$ 3.50</td> </tr> <tr> <td>BEEF TACO PLATE (3)</td> <td>\$ 4.50</td> </tr> <tr> <td>CHEESE ENCHILADA PLATE</td> <td>\$ 4.50</td> </tr> <tr> <td>BEEF ENCHILADA PLATE</td> <td>\$ 5.25</td> </tr> </tbody> </table>	BOWL CHILI	\$ 3.25	BOWL CHILI AND BEANS	\$ 2.75	BOWL GREEN CHILE STEW	\$ 3.50	BEEF TACO PLATE (3)	\$ 4.50	CHEESE ENCHILADA PLATE	\$ 4.50	BEEF ENCHILADA PLATE	\$ 5.25
BOWL CHILI	\$ 3.25													
BOWL CHILI AND BEANS	\$ 2.75													
BOWL GREEN CHILE STEW	\$ 3.50													
BEEF TACO PLATE (3)	\$ 4.50													
CHEESE ENCHILADA PLATE	\$ 4.50													
BEEF ENCHILADA PLATE	\$ 5.25													

Πίνακας 2: Οι κυριότεροι τρόποι αναπαράστασης χωρικού περιεχομένου

Όσον αφορά τα γραπτά αποτελέσματα δεν υπάρχει κάτι ιδιαίτερο που πρέπει να τηχτεί καθώς η διαδικασία είναι απλή: συγκεντρώνονται οι πληροφορίες από τις βάσεις δεδομένων, συντάσσονται με κάποιον τρόπο και αποστέλλονται στον τελικό χρήστη. Η σύνταξη της απάντησης (response) θα βασιστεί κυρίως στην διεπαφή που υποστηρίζει η κινητή συσκευή (WAP ή SMS). Τα γραφικά αποτελέσματα όμως έχουν κάποιες ιδιαιτερότητες αφού εκτός από το περιεχόμενο πρέπει να αποφασιστεί και η μορφοποίηση (format) της τελικής απάντησης. Για εικόνες στατικές (φωτογραφίες και εικόνες που δεν δημιουργούνται δυναμικά από συστήματα GIS) μπορούν να χρησιμοποιηθούν τα format που χρησιμοποιούνται και στο διαδίκτυο (JPEG, GIF). Το πλεονέκτημά τους έναντι άλλων (πχ. BMP) είναι ότι υποστηρίζουν συμπίεση των δεδομένων. Για δυναμικά δημιουργούμενα γραφικά όμως, τα πράγματα είναι λίγο πιο περίπλοκα. Ο «κύριος εκπρόσωπος» αυτής της κατηγορίας είναι οι χάρτες που δημιουργούνται σε πραγματικό χρόνο βασιζόμενοι στην τοποθεσία του χρήστη.

### 1.2.2.3.1 Χάρτες και Χαρτογραφική Διαδικασία

Ο χάρτης είναι παραδοσιακά το κύριο μέσο μετάδοσης της γεωγραφικής πληροφορίας. Κι αυτό γιατί ο χάρτης οριζόμενος ως η γραφική αναπαράσταση της γεωγραφίας (μορφή του χώρου, χωρικά φαινόμενα και χωρικές αλληλοσυσχετίσεις) μιας περιοχής, παρέχει μια άμεση εποπτεία του χώρου, με ελεγμένη ακρίβεια και πληρότητα. Έχοντας ήδη προσδιορίσει ποια είναι η διαδικασία που ακολουθείται για την ανάλυση και συλλογή των δεδομένων που θα απεικονίζει ένας χάρτης, μένει να μελετήσουμε τις τεχνικές που χρησιμοποιούνται για την κατασκευή του. Οι τεχνικές αυτές έχουν σημασία γιατί από αυτές θα καθοριστούν η επεξεργασία που μπορούμε να κάνουμε στο χάρτη (πχ μεγέθυνση) και η επιτυχής μετάδοση της πληροφορίας στον τελικό χρήστη, άρα σε τελική ανάλυση οι υπηρεσίες που μπορούν να προσφερθούν σε αυτόν.

Στην δημιουργία ενός χάρτη υπεισέρχονται δύο κύριες διαδικασίες: (α) η σύνθεση των γραφικών στοιχείων του και (β) η τελική οπτικοποίηση-μορφοποίησή του.

#### α) Σύνθεση γραφικών στοιχείων

Τα βασικά γραφικά στοιχεία ενός χάρτη είναι τα σημεία, οι γραμμές και οι επιφάνειες. Εκτός από αυτά, σπουδαίο ρόλο στην επιτυχία και πληρότητα ενός χάρτη παίζουν και τα λεγόμενα «Χαρτογραφικά Στοιχεία». Σε αυτά ανήκουν τα παρακάτω:

- η κλίμακα  
καθορίζει το επίπεδο ακρίβειας ενός χάρτη και συνυπολογίζεται στην λεπτομέρεια της γραφικής απεικόνισης
- η προβολή  
το ελλειψοειδές στερεό (πχ Γη) μπορεί να προβληθεί σε μία από τις εξής επιφάνειες: την επίπεδη επιφάνεια, τον κύλινδρο, κα τον κώνο. Η πιο συνηθισμένη είναι η κυλινδρική, αφού το ανάπτυσμά της είναι μια επίπεδη ορθογώνια επιφάνεια που ταιριάζει απόλυτα με τα μέσα απεικόνισης (οθόνες).
- τα σύμβολα

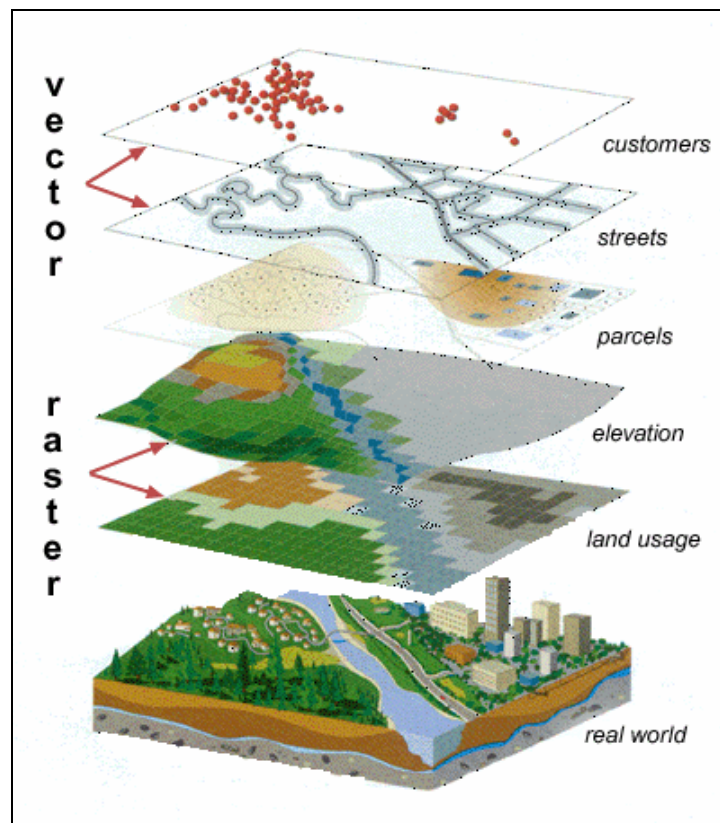


τα πιο συνήθη σύμβολα είναι αυτά που προσδιορίζουν μέγεθος, προσανατολισμό και σχήμα. Επίσης διαφορετικά χρώματα και γραμματοσειρές χρησιμοποιούνται για την διάκριση γεωγραφικών στοιχείων και χαρακτηριστικών.

Τέλος μια τελευταία κατηγορία στοιχείων που απεικονίζονται στους χάρτες είναι τα «Εποπτικά Στοιχεία». Αυτά περιλαμβάνουν τον τίτλο του χάρτη, το υπόμνημα, δείκτη κλίμακας και τον δείκτη προσανατολισμού.

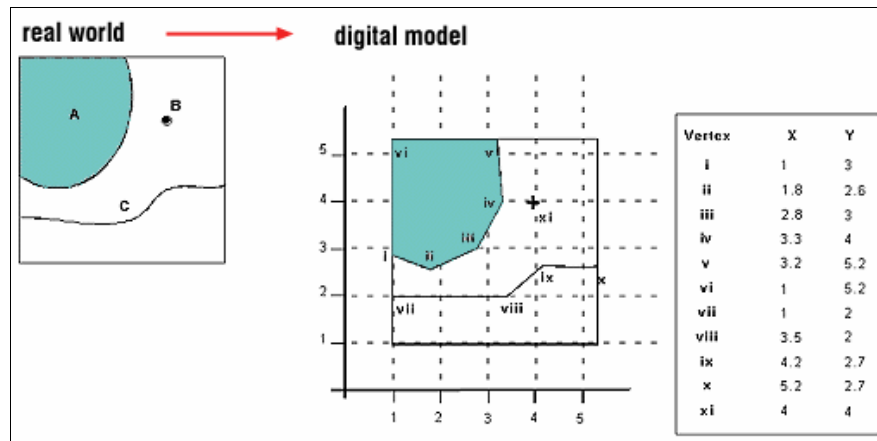
## β) Οπτικοποίηση – Μορφοποίηση Γεωγραφικών Δεδομένων

Η διαδικασία αυτή σχετίζεται με την μέθοδο μορφοποίησης των δεδομένων που αποτελούν τον χάρτη. Τα δύο κύρια μοντέλα δεδομένων και άρα χαρτών που υποστηρίζονται από τα GIS είναι το Ψηφιδωτό(Raster) και το Διανυσματικό(Vector). Παραδείγματα τέτοιων χαρτών φαίνονται στην εικόνα 20.



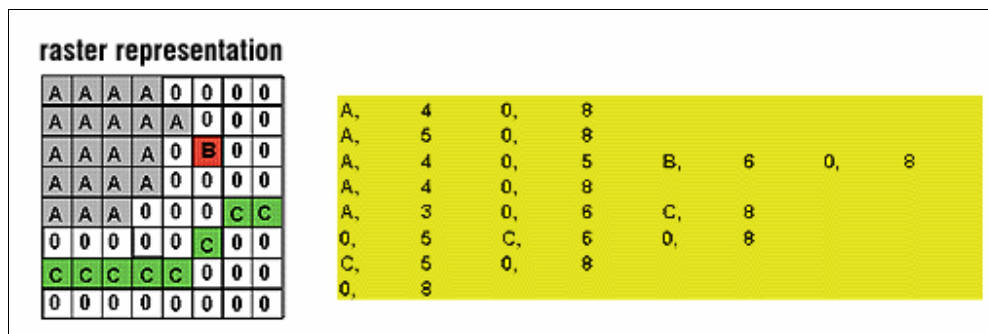
Εικόνα 20: Η διάταξη των χαρτών σε στρώματα και οι κύριοι τύποι χαρτών (vector,raster)

Στους Vector χάρτες, οι γεωγραφικές πληροφορίες αναπαρίστανται με σημεία (συντεταγμένες), γραμμές και πολύγωνα. Καθένα από αυτά τα στοιχεία αποτελείται από ένα σύνολο συντεταγμένων (όπως έχουμε δει και σε προηγούμενη παράγραφο). Ένας τρόπος με τον οποίο μπορούν να δημιουργηθούν Vector χάρτες φαίνεται στην εικόνα 21.



Εικόνα 21: Δημιουργία vector χαρτών

Στους Raster χάρτες, κάθε περιοχή διαιρείται σε στήλες και γραμμές οι οποίες δημιουργούν ένα πλέγμα. Κάθε κυψέλη (cell) αυτού του πλέγματος πρέπει να είναι ορθογώνια αλλά όχι απαραίτητα τετράγωνη. Επίσης κάθε κυψέλη περιέχει τόσο πληροφορία για τις συντεταγμένες της όσο και μια τιμή για κάποιο χαρακτηριστικό που εξαρτάται από το θέμα του χάρτη (πχ υψόμετρο). Οι κυψέλες αυτές αντιστοιχίζονται τελικά στα pixels που θα εμφανιστούν στην οθόνη. Στο επόμενο σχήμα φαίνεται ένας Raster χάρτης καθώς και ένας τρόπος που θα μπορούσαν να αποθηκευτούν τα δεδομένα του (σε αυτό το παράδειγμα χρησιμοποιείται run-length κωδικοποίηση).



Εικόνα 22: Η αναπαράσταση και κωδικοποίηση των δεδομένων σε έναν raster χάρτη

Κάθε είδος χάρτη έχει τις δικές του ιδιότητες καθώς και τα πλεονεκτήματα ή μειονεκτήματά του. Στον πίνακα 3 φαίνονται τα γενικά πλεονεκτήματα/μειονεκτήματα του κάθε είδους.

	raster	vector
Ακρίβεια	X	✓
Μεγαλύτερη ομοιότητα με παραδοσιακούς χάρτες	X	✓
Μικρές αποθηκευτικές απαιτήσεις	X	✓

Αναπαράσταση ακριβούς τοπολογίας	X	✓
Ευκολία υπολογισμών	✓	X
Εύκολη ανανέωση	✓	X
Αναπαράσταση συνεχούς χώρου	✓	X
Τρισδιάστατη αναπαράσταση	✓	X
Αναπαράσταση ορίων περιοχών	X	✓

**Πίνακας 3: Γενικά πλεονεκτήματα και μειονεκτήματα των μοντέλων Raster και Vector**

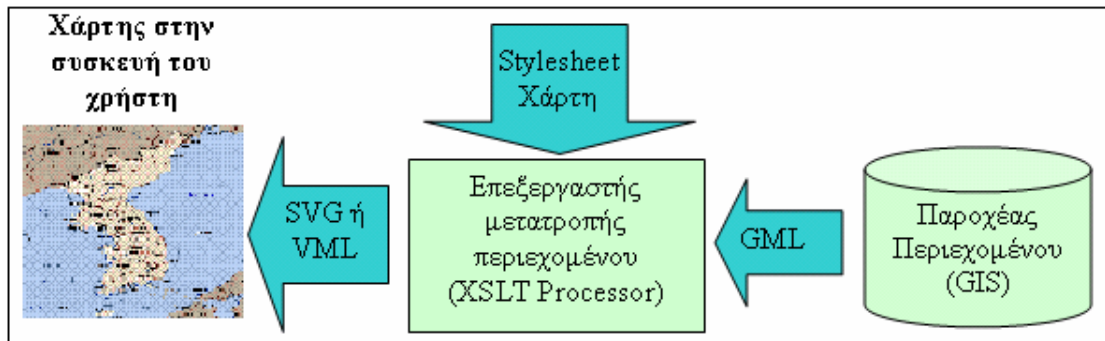
Οι ψηφιδωτοί χάρτες είναι λογικό ότι θα φθάνουν στην κινητή συσκευή με την μορφή bitmap εικόνων. Εξάλλου υπάρχει μια τέλεια αναλογία ανάμεσά τους. Για τους διανυσματικούς χάρτες η κατάσταση είναι πιο περίπλοκη, αφού ένα τέτοιο μοντέλο δεν υποστηρίζεται από κανένα format εικόνων που να χρησιμοποιείται σε δικτυακές εφαρμογές (πχ JPEG, GIF). Έτσι έχουν αναπτυχθεί κάποιες γλώσσες αναπαράστασης διανυσματικών δεδομένων που βασίζονται στην XML και υποστηρίζονται από πολλά GIS συστήματα. Ανάμεσα τους είναι η GML [10], η POIX [11] και η NVML [12]. Οι γλώσσες αυτές δεν χρησιμοποιούνται για την γραφική αναπαράσταση των δεδομένων. Όπως και η XML, έτσι και αυτές περιγράφουν μόνο την δομή και τα χαρακτηριστικά (attributes) των δεδομένων. Για την οπτικοποίηση χρησιμοποιούνται διάφορες άλλες γλώσσες που έχουν προτυποποιηθεί από τους οργανισμούς που ασχολούνται με την ανάπτυξη του Internet (W3C). Κάποιες από αυτές είναι η SVG [13] και η VML [14]. Το τελικό στάδιο βέβαια και στην περίπτωση των διανυσματικών χαρτών είναι η ψηφιοποίησή τους (rasterisation). Αυτή η διαδικασία είναι αναγκαία καθώς όλες οι συσκευές απεικόνισης που χρησιμοποιούνται σήμερα έχουν σαν βασικά στοιχεία απεικόνισης τα pixels. Στις επόμενες παραγράφους θα αναφέρουμε συνοπτικά τα κύρια χαρακτηριστικά της GML και της SVG.

### 1.2.2.3.2 Οι Γλώσσες GML και SVG

Η GML είναι ένα πρότυπο για αναπαράσταση γεωγραφικών πληροφοριών που βασίζεται στην XML. Αναπτύχθηκε από το OpenGIS Consortium (OGC) για να βοηθήσει στην ανταλλαγή χωρικών δεδομένων μέσα από το Internet. Τα GML έγγραφα αποτελούνται από γεωμετρικά δεδομένα και θεματικά χαρακτηριστικά (features) όπως οι οντότητες σπίτι, δρόμος, χώρα κ.α. Τα βασικά γεωμετρικά δεδομένα που υποστηρίζονται από την GML 2.0 είναι το σημείο, η γραμμή και το πολύγωνο. Πιο πολύπλοκα μπορούν να προκύψουν συνδυάζοντας πολλές γεωμετρίες. Με κάθε θεματικό χαρακτηριστικό συσχετίζονται κάποιες ιδιότητες, τις οποίες ο δημιουργός του εγγράφου είναι ελεύθερος να καθορίσει, και κάποιες γεωμετρίες.

Η GML, όπως προαναφέρθηκε, δεν περιέχει καθόλου πληροφορία για την οπτικοποίηση των δεδομένων αυτών. Για να δημιουργηθούν χάρτες από GML έγγραφα χρειάζεται επιπλέον ένα εργαλείο που θα μετατρέψει τα δεδομένα σε κάποια μορφή που θα είναι κατανοητή από τον browser ή γενικά το λογισμικό που χρησιμοποιεί η συσκευή

του χρήστη. Το εργαλείο αυτό για παράδειγμα μπορεί να είναι ένας επεξεργαστής XSLT (XML Transformation Language Processor). Τέτοιοι επεξεργαστές είναι ο Xalan [15] και ο Saxon [16]. Τα XSLT stylesheets χαρτών που χρησιμοποιούνται από τον επεξεργαστή δίνουν την τελική μορφοποίηση στο GML έγγραφο (SVG, VML). Αυτή η διαδικασία γίνεται καλύτερα αντιληπτή από το παρακάτω σχήμα.



Εικόνα 23: Διαδικασία κατασκευής χάρτη

Η SVG (Scalable Vector Graphics) είναι μια γλώσσα για περιγραφή δισδιάστατων δεδομένων σε XML. Η SVG υποστηρίζει τρεις τύπους γραφικών αντικειμένων: διανυσματικά σχήματα (γραμμές, καμπύλες κα), εικόνες και κείμενο. Η SVG υποστηρίζει και κινούμενη εικόνα (animation) που μπορεί να οριστεί και να εκτελεστεί είτε δηλωτικά (με στοιχεία markup) είτε με κώδικα (scripting) τον οποίο ορίζει το πρότυπο. Τα γραφικά της SVG μπορούν να προβληθούν σε διάφορες αναλύσεις χωρίς να μειωθεί πολύ η ακρίβεια και η ομαλότητά τους (smoothness). Αυτό μπορεί να φανεί πολύ χρήσιμο στον χώρο των κινητών επικοινωνιών αφού υπάρχει η δυνατότητα, το γράφημα που απεικονίζεται να καταλαμβάνει πλήρως την οθόνη της κινητής συσκευής η οποία συνήθως διαφέρει από κατασκευαστή σε κατασκευαστή.

Όπως είναι γνωστό στα ψηφιδωτά γραφικά μπορούν να εφαρμοστούν με εύκολο τρόπο φίλτρα και εφέ, όπως σκιές και θολώματα. Η γλώσσα SVG υποστηρίζει τον δηλωτικό καθορισμό φίλτρων που μπορούν να εφαρμοστούν στην συσκευή του χρήστη σε πραγματικό χρόνο. Αυτά είναι ορισμένα με τέτοιο τρόπο που τα γραφικά να μην χάνουν την δυνατότητα που είχαν για μεγέθυνση/σμίκρυνση και απεικόνιση σε πολλαπλές αναλύσεις. Τέλος να αναφέρουμε ότι η παραγωγή της τελικής ψηφιδωτής εικόνας που θα προβληθεί, για την περίπτωση της SVG, μπορεί να γίνεται από τον ίδιο τον browser ή από κατάλληλα plug-ins (πχ Adobe SVG Viewer) ή και από Java προγράμματα και applets.

### 1.2.2.3.3 Φωνητικές Υπηρεσίες

Εκτός από τους κλασικούς οπτικούς τρόπους αναπαράστασης πληροφορίας έχουν αναπτυχθεί και άλλοι τρόποι που έχουν σαν κύριους στόχους να είναι πιο αποτελεσματικοί και να δίνουν καλύτερη εμπειρία στον χρήστη (advanced user experience). Ένας από αυτούς, που φυσικά να μπορεί να εφαρμοστεί σε υπηρεσίες κινητών επικοινωνιών, είναι η φωνητική «αναπαράσταση» των δεδομένων. Με τον όρο

αυτό εννοούμε ότι τόσο η αίτηση για μια υπηρεσία όσο και η απάντηση είναι σε μορφή φωνής. Η απάντηση φυσικά εννοείται ότι θα δημιουργείται δυναμικά, αφού αλλιώς η υπηρεσία θα εκτελούσε χρέη αυτόματου τηλεφωνητή. Μερικά παραδείγματα φωνητικών υπηρεσιών είναι τα εξής:

- **Ψυχαγωγία-Διασκέδαση** (παιχνίδια, ωροσκόπια, κριτικές ταινιών): Αναμένεται ότι αυτές οι υπηρεσίες θα αποτελούν το μεγαλύτερο μέρος των φωνητικών υπηρεσιών.
- **Προηγμένες υπηρεσίες καταλόγου** που θα παρέχουν αριθμούς τηλεφώνου, ωράρια καταστημάτων ή διευθύνσεις και κατευθύνσεις.
- **Οικονομικές** που θα συμπεριλαμβάνουν χρηματιστηριακά νέα και αγορές ή δοσοληψίες με τράπεζες.
- **Σχετικές με ταξίδια:** θα μπορούσαν να περιέχουν επισημάνσεις για καθυστέρηση πτήσης, δελτία καιρού, κυκλοφοριακές συνθήκες και λεπτομερή πλοήγηση κατά την οδήγηση (turn-by-turn driving instructions).

Ένα σημαντικό πλεονέκτημα των υπηρεσιών αυτών είναι ότι είναι πολύ εύκολες να χρησιμοποιηθούν από άπειρους χρήστες τερματικών συσκευών αλλά και από άτομα με αναπηρίες. Από την άλλη πλευρά όμως, υπάρχουν κάποια εγγενή προβλήματα που εμποδίζουν ακόμα την εξάπλωση αυτού του τύπου διεπαφής με τον χρήστη. Ένα από αυτά είναι ότι οι φωνητικές υπηρεσίες απαιτούν σχετικά μεγάλο εύρος ζώνης από το δίκτυο. Ένα άλλο είναι οι δυσκολίες που υπάρχουν στην επεξεργασία φυσικής γλώσσας και φωνής. Τέλος εμπόδια υπάρχουν και από την επιχειρηματική πλευρά αφού τα παραδοσιακά επιχειρηματικά μοντέλα (δηλαδή αυτά που στηρίζονται στις διαφημίσεις) δεν μπορούν να εφαρμοστούν εύκολα. Έρευνες έδειξαν ότι οι χρήστες δεν είναι πρόθυμοι να χρησιμοποιήσουν φωνητικές υπηρεσίες αν πρόκειται να ακούν ηχητικές διαφημίσεις.

Η τεχνολογία που χρησιμοποιείται σήμερα, και έχει γίνει ευρέως αποδεκτή, για φωνητικές εφαρμογές είναι η VoiceXML [17]. Η γλώσσα αυτή συνδυάζει την XML με προηγμένη αναγνώριση φωνής. Η έκδοση 1.0 που δημιουργήθηκε από εταιρίες όπως οι AT&T, IBM, Lucent Technologies και Motorola το, έγινε πρότυπο το 2000 από τον W3C.

#### **1.2.2.3.4 Τρόποι Αναπαράστασης Περιεχομένου και Εξαρτώμενες από την Θέση Υπηρεσίες**

Σε αυτήν τη παράγραφο θα δούμε πως συνδέονται όλα τα παραπάνω με τις υπηρεσίες που μελετάμε και αν υπάρχουν κάποια θέματα που να αφορούν ειδικά στο μοντέλο Αυτόματης Προώθησης. Καταρχήν πριν φτάσουμε στον τρόπο αναπαράστασης της πληροφορίας πρέπει να λάβουμε σοβαρά υπόψη μας τα θέματα που προκύπτουν από τις περιορισμένες ικανότητες προβολής (display) της κινητής συσκευής. Αυτό βέβαια ισχύει κυρίως για τα κινητά τηλέφωνα, αφού τα PDAs και οι φορητοί υπολογιστές έχουν οθόνες ικανοποιητικών διαστάσεων, χρωμάτων και αναλύσεων. Στα κινητά τηλέφωνα πρέπει να γίνεται πολύ προσεκτικά και μεθοδικά ο σχεδιασμός της διεπαφής (design).

Μια γενική αρχή είναι ότι πρέπει κάθε υπηρεσία να υποστηρίζει αρκετούς εναλλακτικούς τρόπους παρουσίασης του περιεχομένου, αν δεν θέλουμε να αποκλείσουμε προκαταβολικά κάποιες ομάδες χρηστών. Αυτή η αρχή υποδεικνύει ότι πρέπει να λαμβάνεται ειδική μέριμνα και για κατηγορίες χρηστών με ειδικές ανάγκες και αναπηρίες. Το κείμενο πρέπει να είναι σωστά δομημένο σε λίστες ή μενού ανάλογα με το περιεχόμενο. Επίσης πρέπει να συνυπολογίζεται το γεγονός ότι οι οθόνες των κινητών μπορούν να προβάλουν μόνο λίγες γραμμές κειμένου έτσι ώστε να γίνεται σωστός διαμερισμός του συνολικού κειμένου.

Οι χάρτες που παρουσιάζονται στον χρήστη πρέπει να είναι απλοί με την έννοια ότι πρέπει να περιέχουν τα ελάχιστα στοιχεία που απαιτούνται για την σωστή πληροφόρηση του χρήστη. Για να γίνει αυτό πρέπει να εξαντλούνται όλες οι διαδικασίες ανάλυσης χώρου που διαθέτει το GIS σύστημα. Προσοχή χρειάζεται και στην επιλογή των χρωμάτων αφού κάθε συσκευή διαθέτει διαφορετικό αριθμό υποστηριζόμενων χρωμάτων. Επίσης πολλά χρώματα σε μικρή επιφάνεια προκαλούν χρωματική ρύπανση και καθιστούν δυσδιάκριτα τα στοιχεία του χάρτη. Σε κάθε περίπτωση πρέπει να υπάρχει και η δυνατότητα προβολής των χαρτών σε μονόχρωμες οθόνες. Αυτό ίσως να απαιτεί την περαιτέρω επεξεργασία του χάρτη ώστε να κάποιες χρωματικές πληροφορίες να αντικατασταθούν από σύμβολα ή μοτίβα (patterns).

Ένα άλλο θέμα όσον αφορά τους χάρτες είναι η χρήση διανυσματικών ή ψηφιδωτών γραφικών. Τα διανυσματικά γραφικά πλεονεκτούν στο ότι έχουν μικρότερες αποθηκευτικές απαιτήσεις και άρα μικρότερες απαιτήσεις εύρους ζώνης. Επίσης μπορούν να υποστηρίξουν πιο εύκολα κινούμενη εικόνα, η οποία ως γνωστόν δημιουργεί καλύτερα αισθητικά αποτελέσματα. Όμως έχουν και σημαντικά μειονεκτήματα που είναι και ο λόγος για την μη ευρεία χρήση τους. Όπως προαναφέρθηκε μπορούν να ψηφιδωποποιηθούν είτε στον server που θα παρέχει την υπηρεσία είτε στη τερματικά συσκευή από ειδικό λογισμικό. Το λογισμικό αυτό αυξάνει το κόστος της υπηρεσίας και στην περίπτωση που υπάρχει στη συσκευή (πχ στον browser) αυξάνει και τον υπολογιστικό φόρτο της.

Το μοντέλο Αυτόματης Προώθησης, εκτός των άλλων, εμπεριέχει και την τακτική αποστολή δεδομένων στην κινητή συσκευή μέσα από διαδικασίες όπως η χρονοδρομολόγηση (time-scheduling) και ο χωρικός σκανδαλισμός (location triggering). Πολλές από τις υπηρεσίες που χρησιμοποιούν αυτό το μοντέλο είναι και χρονικά κρίσιμες (time-critical). Αυτό το τελευταίο σημαίνει ότι απαιτείται γρήγορη απόκριση και άρα μικροί όγκοι μεταφερόμενων δεδομένων, αφού η μεταφορά δεδομένων μέσα από το δίκτυο αποτελεί το κυριότερο σημείο συμφόρησης (bottleneck). Αν τα δεδομένα είναι χάρτες, η υιοθέτηση ενός διανυσματικού τρόπου αναπαράστασης μπορεί να ανταποκριθεί καλύτερα στις παραπάνω απαιτήσεις από τον ψηφιδωτό τρόπο. Πιο συγκεκριμένα, αν έχουμε τακτικές προωθήσεις χαρτών στους οποίους κάποια οντότητα είναι κινούμενη (συνήθως ο ίδιος ο χρήστης), συμφέρει να χρησιμοποιήσουμε κάποια γλώσσα διανυσματικής αναπαράστασης όπως η GML. Κάθε φορά θα πρέπει να αλλάζουμε μόνο τα στοιχεία που μεταβλήθηκαν από την προηγούμενη προώθηση. Αυτό σημαίνει ότι η ενημέρωση του χάρτη θα γίνεται με διαδικασίες επεξεργασίας κειμένου (text processing), αφού οι γλώσσες αυτές είναι βασισμένες στην XML, και ότι θα μεταφέρονται μέσω του ασύρματου δικτύου μόνο τα μεταβλητά στοιχεία. Φυσικά αυτό δεν ισχύει γενικά, αφού προϋποθέτει ότι η τελική οπτικοποίηση γίνεται στο τερματικό και ότι το υπόλοιπο μέρος του χάρτη μένει σταθερό, αλλά είναι μια δυνατότητα που μας

δίνεται από την υπάρχουσα τεχνολογία. Τα ψηφιδωτά γραφικά απαιτούν μεθόδους επεξεργασίας εικόνας (image processing) για να ενημερώσουν μόνο ένα μέρος του χάρτη, οι οποίες είναι σαφώς πιο πολύπλοκες.

#### **1.2.2.4 Χωρικές Βάσεις Δεδομένων**

Μια Χωρική Βάση Δεδομένων (ΧΒΔ, Spatial Database) χρησιμοποιείται για την αποθήκευση χωρικών αντικειμένων που αναπαρίστανται με χωρικούς τύπους δεδομένων και σχετίζονται με χωρικές συσχετίσεις. Τα χωρικά δεδομένα φέρουν πληροφορίες σχετικές με την τοπολογία και τις αποστάσεις των αντικειμένων αυτών. Για την οργάνωσή τους χρησιμοποιούνται κατάλληλες δομές ευρετηρίων και για την προσπέλασή τους κατάλληλες επερωτήσεις. Όλα τα παραπάνω συνοψίζουν τις διαφορές που υπάρχουν ανάμεσα στις ΧΒΔ και στις συνήθεις (σχεσιακές) βάσεις δεδομένων.

Οι ΧΒΔ προέκυψαν από το επιτυχημένο σχεσιακό μοντέλο με κύριο σκοπό να εξυπηρετήσουν άλλες τεχνολογίες, όπως είναι τα GIS. Από το γεγονός αυτό και όσα έχουν προαναφερθεί για τα GIS, γίνεται προφανής η σημασία τους για τα συστήματα που παρέχουν υπηρεσίες εξαρτώμενες από την θέση.

Σήμερα πολλά πανεπιστήμια και ερευνητικές ομάδες δραστηριοποιούνται στο πεδίο των τεχνικών που μπορούν να χρησιμοποιηθούν σε τέτοιου είδους βάσεις (πχ spatial data mining). Επίσης πολλές εταιρίες έχουν ήδη παρουσιάσει συστήματα διαχείρισης βάσεων δεδομένων (DBMS) ενώ άλλες έχουν προσαρτήσει λειτουργίες διαχείρισης χωρικών δεδομένων στα υπάρχοντα συστήματά τους (τέτοια παραδείγματα αποτελούν τα εργαλεία Oracle Spatial [18] και Oracle Locator[19] που μας επιτρέπει να αποθηκεύουμε, να προσπελάζουμε και να αναλύουμε τέτοια δεδομένα γρήγορα και αποτελεσματικά).

Οι βασικοί γεωμετρικοί τύποι από τους οποίους αποτελούνται όλα τα στοιχεία μιας τέτοιας βάσης είναι:

- Σημείο (Point)
- Γραμμή (Line)
- Πολύγωνο (Polygon)

Το σημείο είναι ένα στοιχείο που αποτελείται από 2 συντεταγμένες (οι οποίες μπορεί να είναι και γεωγραφικό μήκος και πλάτος). Η γραμμή αποτελείται από ένα ζευγάρι σημείων και μπορεί να σχηματίσει μία σειρά γραμμών (line string). Το πολύγωνο αποτελείται από ένα σύνολο συνδεδεμένων γραμμών οι οποίες όμως πρέπει να σχηματίζουν ένα κλειστό δακτύλιο. Τα μεμονωμένα σημεία μπορούν, για παράδειγμα, να αναπαριστούν κάποια γεωγραφικά σημεία ιδιαίτερης σημασίας για τον χρήστη της βάσης δεδομένων, όπως τηλεφωνικούς θαλάμους ή φανάρια. Όμοια οι σειρές γραμμών μπορούν να αναπαριστούν δρόμους και τα πολύγωνα γεωγραφικά όρια. Βέβαια οι δυνατότητες αναπαράστασης δεν σταματούν εδώ αφού μπορούν να συνδυαστούν τα παραπάνω στοιχεία για να προκύψουν πιο περίπλοκες δομές (είναι πολύ συχνή η συνεργασία των ΧΒΔ με εφαρμογές CAD/CAM, που ως γνωστόν χρησιμοποιούνται για να

μοντελοποιούν αρκετά περίπλοκες δομές στον χώρο). Τέλος μπορεί να υπάρχουν και ειδικές τύποι δεδομένων που αναπαριστούν επίπεδα στον χώρο (layers).

Επειδή τα χωρικά δεδομένα αναπαρίστανται με ειδικούς τύπους δεδομένων, για την οργάνωση τους χρησιμοποιούνται άλλοι τύποι ευρετηρίων από τους συνηθισμένους. Οι δομές αυτές είναι συνήθως πολυδιάστατα δέντρα. Μερικά παραδείγματα είναι τα τετραδικά δέντρα (quad trees), τα R-δέντρα και τα R\*-δέντρα. Τα R-δέντρα είναι από τα πιο πολυχρησιμοποιημένα. Κάθε κόμβος-φύλλο των δέντρων αυτών είναι της μορφής (I, P), όπου I το ελάχιστο ορθογώνιο στο οποίο μπορεί να περιέχεται το χωρικό στοιχείο της ΧΒΔ στο οποίο δείχνει ο δείκτης P. Οι υπόλοιποι κόμβοι είναι πλειάδες που αποτελούνται από στοιχεία της μορφής (I, PC), όπου I είναι το ελάχιστο ορθογώνιο στο οποίο μπορούν να περιέχονται όλα τα ορθογώνια των παιδιών του συγκεκριμένου κόμβου. PC είναι ο δείκτης που δείχνει στο κόμβο-παιδί του συγκεκριμένου κόμβου. Οι υπόλοιπες ιδιότητες των δέντρων αυτών μοιάζουν πολύ με τις ιδιότητες των B-δέντρων.

Οι πράξεις που μπορούν να εφαρμοστούν στα χωρικά δεδομένα είναι επερωτήσεις (πχ βρες τα N γειτονικά στοιχεία ενός στοιχείου), συνενώσεις (spatial joins), έλεγχος για επικάλυψη στοιχείων διαφορετικών επιπέδων (πχ η λειτουργία map overlay των GIS) και άλλες. Βέβαια όλες αυτές οι πράξεις απαιτούν πολύ περίπλοκους αλγορίθμους έτσι ώστε να ελαχιστοποιηθεί η (αρκετά) μεγάλη χρονική πολυπλοκότητα τους. Η ανακάλυψη τέτοιων αλγορίθμων είναι μια από τις σημαντικές ερευνητικές περιοχές στις ΧΒΔ.

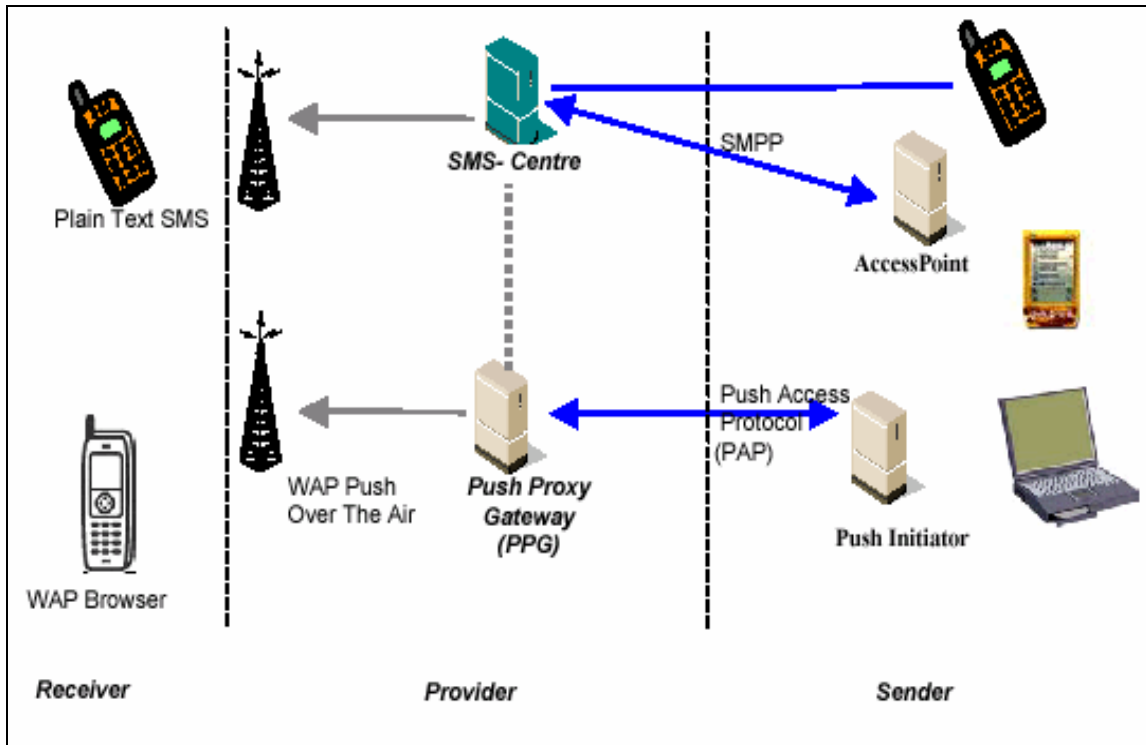
### 1.2.3 Διεπαφές

Με τον όρο διεπαφή (interface) εννοούμε γενικά τον τρόπο με τον οποίο παρουσιάζεται ένα σύνολο δεδομένων σε ένα τερματικό. Ειδικότερα όμως μία διεπαφή αποτελείται από ένα σύνολο πρωτοκόλλων (αν πρόκειται για δικτυακή εφαρμογή) και προγραμμάτων τα οποία καθορίζουν με κάθε λεπτομέρεια την όλη δομή, με τελικό σκοπό την μεταφορά των δεδομένων και την εμφάνισή τους στον τελικό χρήστη.

Στην περίπτωση των ασύρματων επικοινωνιών (άρα και των LBS υπηρεσιών) το τερματικό του χρήστη είναι αυτό που καθορίζει κατά ένα μεγάλο ποσοστό και την διεπαφή που θα χρησιμοποιηθεί. Οι τεχνικές προδιαγραφές του κινητού τερματικού παίζουν πολύ σημαντικό ρόλο.

Σήμερα οι δύο κυριότεροι τρόποι παρουσίασης των δεδομένων σε μια ασύρματη συσκευή είναι το SMS και το WAP. Η επόμενη εικόνα περιγράφει ενοποιημένα αυτούς τους δύο τρόπους:





Εικόνα 24: SMS και WAP διεπαφή στο μοντέλο Push

### 1.2.3.1 SMS

#### 1.2.3.1.1 Γενικά

Η υπηρεσία των γραπτών μηνυμάτων Short Message Service, γνωστή παγκοσμίως απλά ως SMS [20], είναι μία ασύρματη υπηρεσία που δίνει την δυνατότητα ανταλλαγής μηνυμάτων, με περιεχόμενο αλφαριθμητικά και σύμβολα, μεταξύ χρηστών της κινητής τηλεφωνίας αλλά και με εξωτερικά συστήματα όπως e-mail, voice mail systems, mobile banking, voting κ.α. Μετά την εμφάνισή της το 1991 στην Ευρώπη σαν standard του συστήματος GSM έγινε ταχύτατα παγκοσμίως αποδεκτή.

Η υπηρεσία χρησιμοποιεί ένα κέντρο, το Short Message Service Center (SMSC), που έχει τον ρόλο ενός συστήματος τύπου store-and-forward. Το ασύρματο δίκτυο μεταφέρει τα γραπτά μηνύματα μεταξύ του κέντρου και των ασύρματων τερματικών προσφέροντας εγγυημένη παράδοση των δεδομένων ανακαλύπτοντας οποιοδήποτε πρόβλημα προκύψει και αποθηκεύοντας το μήνυμα έως ότου αυτό το πρόβλημα ρυθμιστεί.

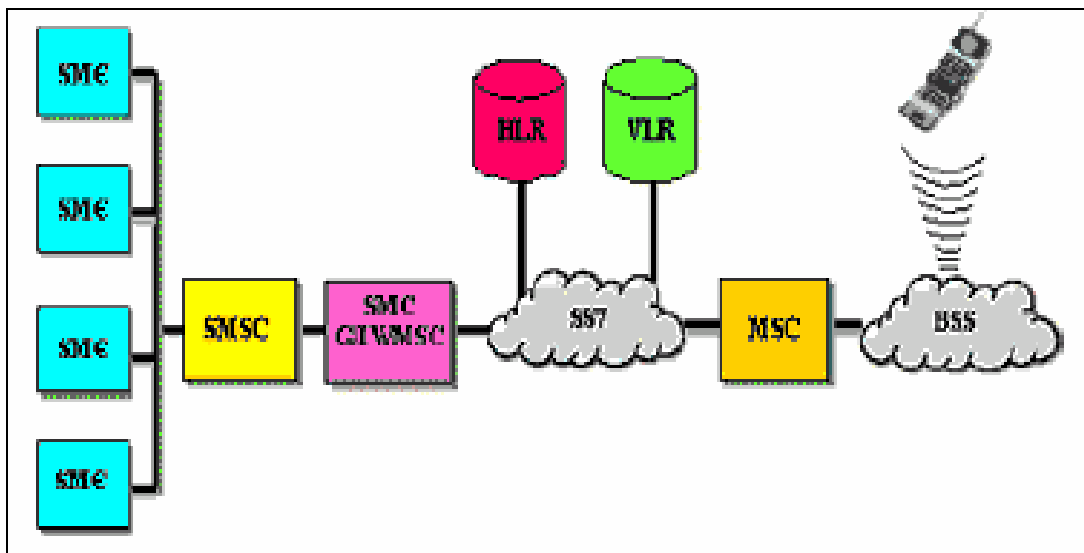
Το κυριότερο χαρακτηριστικό της υπηρεσίας είναι ότι ένα κινητό μπορεί να λάβει ή να στείλει μηνύματα οποιαδήποτε ώρα (υπό την προϋπόθεση ότι είναι ενεργό) ανεξάρτητα αν εκείνη τη στιγμή βρίσκεται σε εξέλιξη κλήση για φωνή ή δεδομένα. Ένα

ακόμη χαρακτηριστικό είναι η μη επιβάρυνση του δικτύου αφού απαιτεί μικρό εύρος ζώνης.

Στα αρχικά στάδια υπήρχε περιορισμένη λειτουργικότητα αλλά αργότερα με την ανάπτυξη της τεχνολογίας εμφανίστηκαν νέες υπηρεσίες όπως e-mail, fax, e-banking αλλά και για εφαρμογές όπως ενεργοποίηση της κάρτας SIM κα. Επίσης η υπηρεσία χρησιμοποιείται πολλές φορές και από το ίδιο το σύστημα για ανταλλαγή σημάτων ελέγχου χωρίς ο χρήστης να αντιλαμβάνεται κάτι.

### 1.2.3.1.2 Αρχιτεκτονική

Η αρχιτεκτονική και τα δομικά στοιχεία της υπηρεσίας φαίνονται στην παρακάτω εικόνα:



Εικόνα 25: Αρχιτεκτονική του SMS

- **SME (Short Message Entities)**  
Είναι μια οντότητα που μπορεί να στείλει ή να λάβει μηνύματα. Η θέση της μπορεί να βρίσκεται στο δίκτυο, στο κινητό ή σε κάποιο άλλη εξωτερική οντότητα
- **SMSC(Short Message Service Center)**  
Το κέντρο αυτό είναι υπεύθυνο για την αποστολή μηνυμάτων μεταξύ μιας οντότητας (SME) και του κινητού σταθμού.
- **SMS – Gateway / Interworking Mobile Switching Center**  
Το κέντρο SMS – Gateway Mobile Switching Center (SMS-GMWC) έχει την δυνατότητα λήψης μηνυμάτων από ένα SMSC, εξετάζει την βάση HLR (Home Location Register) για πληροφορίες σχετικά με τη θέση και προωθεί το μήνυμα στο αντίστοιχο κέντρο MSC (Mobile Switching Center) το οποίο είναι υπεύθυνο

για τον αντίστοιχο τελικό παραλήπτη. Το κέντρο SMS Interworking Mobile Switching Center (SMS-IW MSC) έχει την δυνατότητα λήψης μηνυμάτων από το δίκτυο προωθώντας τα στο κατάλληλο SMSC.

- **Home Location Register**

Είναι μια βάση δεδομένων που χρησιμοποιείται για μόνιμη αποθήκευση και διαχείριση των δεδομένων των συνδρομητών και των αντίστοιχων προφίλ του καθενός. Όταν εξετάζεται από ένα SMSC η βάση παρέχει πληροφορίες σχετικά με τη θέση του αντίστοιχου συνδρομητή. Επίσης ενημερώνει το ίδιο κέντρο, που προηγουμένως απέτυχε στην αποστολή ενός μηνύματος λόγω του ότι το κινητό ήταν απενεργοποιημένο, ότι τώρα είναι ενεργό και μπορεί να στείλει το μήνυμά του.

- **Mobile Switching Center**

Εκτελεί και ελέγχει όλες τις διεργασίες που έχουν να κάνουν με δρομολόγηση των κλήσεων από και προς άλλα τηλεφωνικά συστήματα και συστήματα δεδομένων.

- **Visitor Location Register**

Η VLR (Visitor Location Register) είναι μια βάση που περιέχει προσωρινή πληροφορία για τους συνδρομητές και η οποία χρησιμοποιείται για λειτουργίες όπως η περιαγωγή (roaming). Οι πληροφορίες αυτές χρησιμοποιούνται από τα κέντρα MSC.

- **Base Station System**

Όλες οι λειτουργίες που σχετίζονται με την ραδιοδιάδοση εκτελούνται από το σύστημα του σταθμού βάσης BSS. Αποτελείται από ελεγκτές (Base Station Controllers, BSCs) και πομποδέκτες (Base Transceiver Stations, BTSs), και η κύρια ευθύνη του είναι η μετάδοση φωνής και δεδομένων μεταξύ κινητών σταθμών.

- **Mobile Station**

Είναι το ασύρματο τερματικό το οποίο μπορεί να λαμβάνει και να στέλνει μηνύματα SMS καθώς και φωνή.

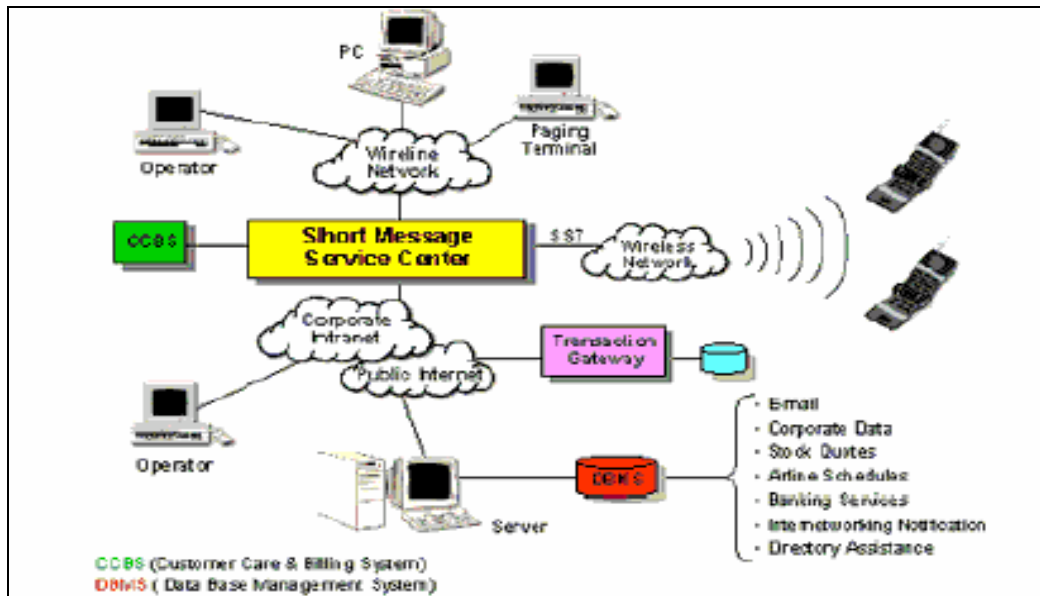
Η υποδομή του δικτύου για μεταφορά σημάτων ελέγχου βασίζεται στο σύστημα SS7 (Signaling System 7). Η υπηρεσία SMS κάνει χρήση του MAP (Mobile Application Part), που ορίζει τους μηχανισμούς επικοινωνίας στα ασύρματα δίκτυα, και χρησιμοποιεί επίσης τις υπηρεσίες του TCAP (Transaction Capabilities Application Part).

Μπορούν να διακριθούν δύο τύποι μηνυμάτων SMS: το προερχόμενο από τον κινητό σταθμό (Mobile-Originated Short Message, MO-SM) και το τερματιζόμενο στον κινητό σταθμό (Mobile-Terminated Short Message, MT-SM). Τα μηνύματα MO-SM μεταφέρονται από το κινητό στα κέντρα SMSC και τελικά καταλήγουν σε άλλα κινητά ή κατευθύνονται σε δίκτυα ηλεκτρονικού ταχυδρομείου (electronic mail networks). Τα MT-SM μεταφέρονται από τα κέντρα SMSC στις ασύρματες συσκευές. Στα κέντρα αυτά μπορούν να φτάσουν μέσω των MO-SM μηνυμάτων από συνδρομητές ή από άλλες πηγές όπως συστήματα φωνητικού ή ηλεκτρονικού ταχυδρομείου (voice-mail or e-mail systems).

Για ένα μήνυμα MT-SM, επιστρέφεται πάντα μία αναφορά στο κέντρο SMSC από το οποίο εστάλη, ενημερώνοντας το για την επιτυχή παράδοση του μηνύματος στο τερματικό ή για την αποτυχία, προσδιορίζοντας ταυτόχρονα και τον λόγο. Αντίστοιχα

για ένα μήνυμα MO-SM, επιστρέφεται πάντα μία αναφορά στο τερματικό από το οποίο εστάλη, προσδιορίζοντας την επιτυχία ή την αποτυχία αποστολής του στο κέντρο SMSC.

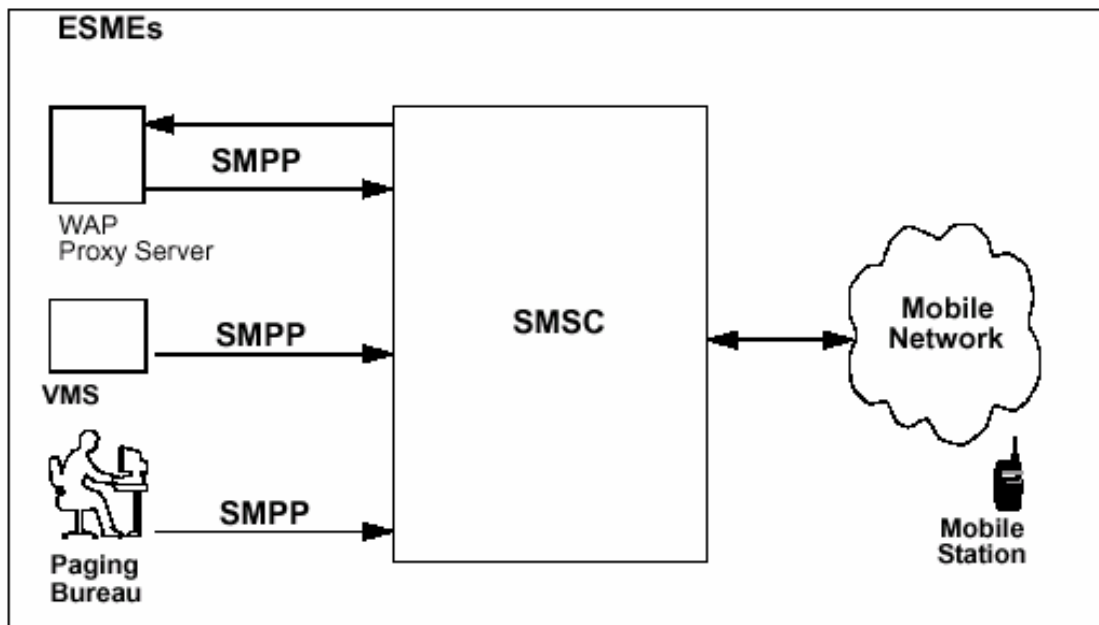
Η παρακάτω εικόνα αποτυπώνει την όλη δομή του συστήματος καθώς και τα εξωτερικά στοιχεία με τα οποία υπάρχει επικοινωνία, όπως βάσεις δεδομένων κ.τ.λ.



Εικόνα 26: Η σχέση του SMS υποσυστήματος με το υπόλοιπο δίκτυο

### 1.2.3.1.3 Το πρωτόκολλο SMPP (Short Message Peer to Peer protocol)

Το πρωτόκολλο SMPP [21] Είναι ένα ανοικτό πρότυπο (open industry standard) που σχεδιάστηκε για να απλοποιήσει την σύνδεση και την επικοινωνία για την μεταφορά SMS μεταξύ των κέντρων SMSC και μίας εξωτερικής οντότητας ESME (External Short Message Entity) όπως έναν WAP Proxy Server ή EMail Gateway. Η θέση του φαίνεται καθαρά στην παρακάτω εικόνα:



Εικόνα 27: Το πρωτόκολλο SMPP

Η έκδοση 3.4 του πρωτοκόλλου υποστηρίζει τεχνολογίες δικτύων όπως GSM, IS-95 (CDMA), ANSI-136 (TDMA), iDEN.

Χρησιμοποιώντας το πρωτόκολλο SMPP, μία ESME μπορεί να στείλει και να λάβει μηνύματα προς και από ένα κέντρο SMSC πάνω από δικτυακή σύνδεση TCP/IP ή X.25. Η οντότητα μπορεί να ακυρώσει ή να αντικαταστήσει μηνύματα που έχουν ήδη σταλεί σε ένα κέντρο μέσω του πρωτοκόλλου.

Οι σημαντικότερες λειτουργίες που υποστηρίζει το SMPP είναι οι εξής:

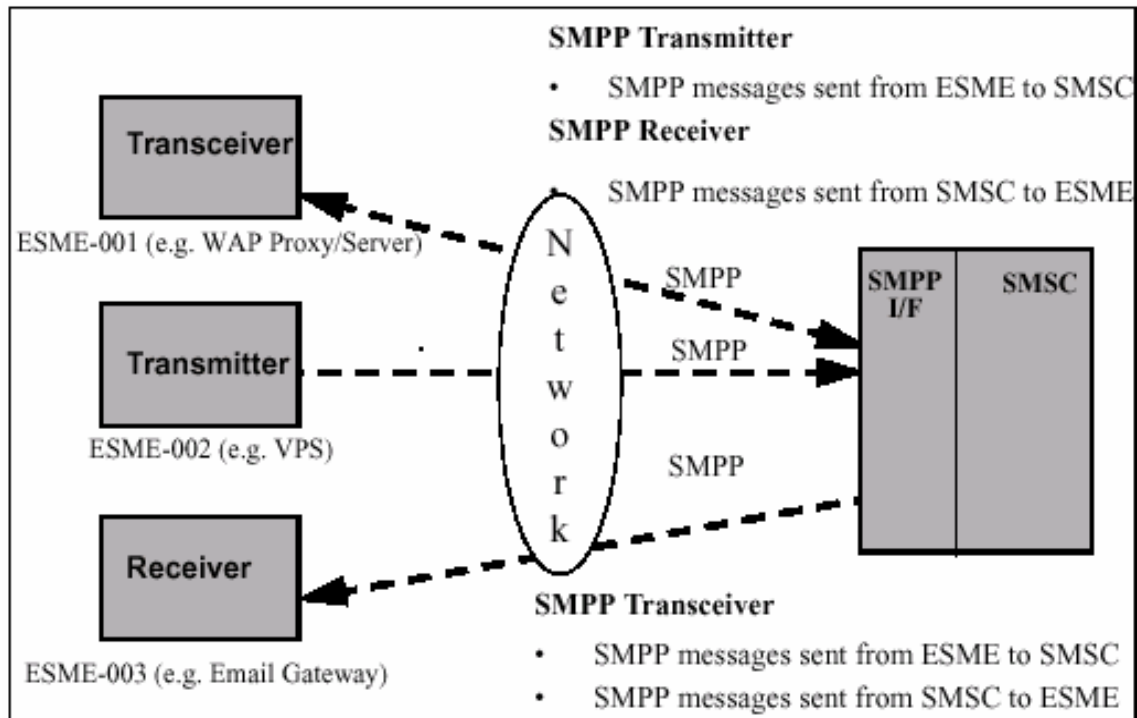
- Μεταφέρει μηνύματα από μία οντότητα ESME σε πολλούς προορισμούς μέσω των κέντρων SMSC.
- Μία οντότητα ESME μπορεί να λαμβάνει μηνύματα μέσω των κέντρων SMSC από κινητούς σταθμούς ή άλλες οντότητες.
- Επερώτηση για την κατάσταση ενός μηνύματος που είναι αποθηκευμένο στο κέντρο SMSC.
- Ακύρωση ή αντικατάσταση μηνυμάτων.
- Χρονοπρογραμματισμός ενός μηνύματος (ημερομηνία και ώρα παράδοσης).
- Επιλογή του τύπου του μηνύματος (datagram ή store-and-forward).
- Καθορισμός της προτεραιότητας αποστολής του μηνύματος.
- Καθορισμός της χρονικής ισχύος του μηνύματος.

Η ανταλλαγή των μηνυμάτων μπορεί να χωριστεί σε τρεις κατηγορίες:

1. Μηνύματα που στέλνονται από μία ESME (Transmitter) σε ένα κέντρο SMSC.
2. Μηνύματα που στέλνονται από ένα κέντρο SMSC σε μία ESME (Receiver) .

3. Μηνύματα που στέλνονται από μία ESME (Transceiver) σε ένα κέντρο SMSC και μηνύματα που στέλνονται από ένα κέντρο SMSC σε μία ESME (Transceiver).

Στην παρακάτω εικόνα απεικονίζονται οι τρεις αυτές κατηγορίες



Εικόνα 28: Οι τρεις τύποι μηνυμάτων στο SMPP

#### 1.2.3.1.4 Εφαρμογές SMS

Ο αρχικός σχεδιασμός προέβλεπε η υπηρεσία να υποστηρίζει ανταλλαγή μικρού μήκους μηνυμάτων. Αυτός ο τρόπος ανταλλαγής της πληροφορίας χρησιμοποιείται σήμερα ευρύτατα. Ο λόγος που γίνεται αυτό έγκειται στο γεγονός ότι πλήθος εφαρμογών χρειάζονται μικρή ποσότητα πληροφορίας ώστε να λειτουργήσουν. Έτσι έχουμε χαμηλό κόστος. Μεγάλο μέρος των υπηρεσιών θέσης ανήκει στην κατηγορία αυτή, π.χ. η ιχνηλασία (Tracking) μίας κινούμενης οντότητας (αυτοκινήτου) απαιτεί ανταλλαγή πληροφορίας μικρού μεγέθους όπως τις γεωγραφικές συντεταγμένες, μια συγκεκριμένη ώρα ή άλλες παραμέτρους.

Μία άλλη κατηγορία εφαρμογών είναι αυτές που έχουν να κάνουν με συναλλαγές, μέσω του κινητού, με την τράπεζα ή γενικά με άλλες οντότητες. Αν εξαιρέσουμε το θέμα της δυσπιστίας, ο τρόπος αυτός μειώνει σημαντικά το κόστος και από την πλευρά του πελάτη αλλά και της τράπεζας. Πολύ σημαντικό ρόλο παίζει και στην επικοινωνία μεταξύ δύο (γνωστό σε όλους μας σχεδόν) ή περισσοτέρων (chat) ατόμων, που με την βοήθεια του κινητού στέλνουν και λαμβάνουν γραπτά μηνύματα.

Τέλος θα ήταν παράληψη να μην αναφέρουμε και την εφαρμογή που έχει η υπηρεσία στην ανάγνωση του ηλεκτρονικού ταχυδρομείου (e-mail) αλλά και η συνεισφορά στην ανάπτυξη των διαδικτυακών αγορών μέσω του κινητού.

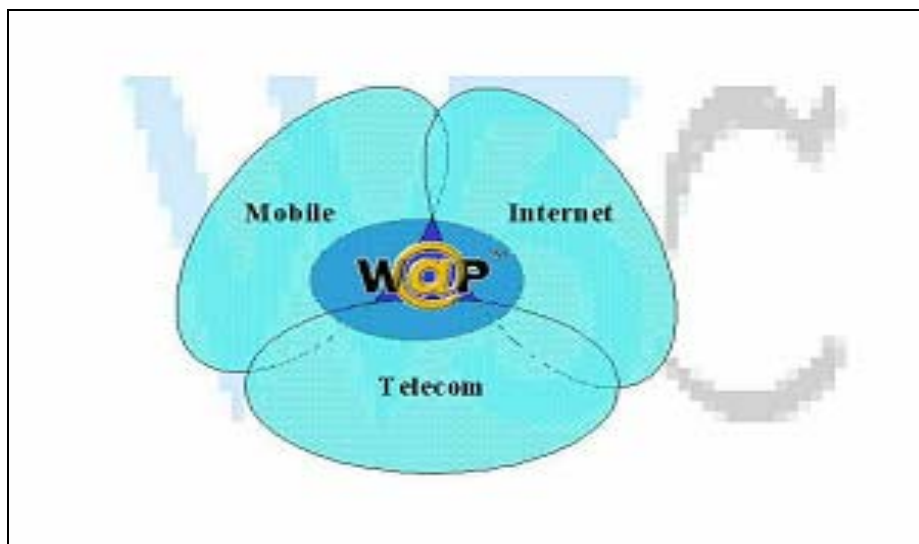
### 1.2.3.2 WAP

#### 1.2.3.2.1 Γενικά

Το WAP (Wireless Application Protocol) [22] είναι ένα ανοιχτό διεθνές πρότυπο (standard) για την παρουσίαση και την παράδοση πληροφορίας και υπηρεσιών σε ψηφιακά κινητά τηλέφωνα, PDAs (Personal Digital Assistants) και άλλα ασύρματα τερματικά. Το πρωτόκολλο αυτό αναπτύχθηκε από τον οργανισμό WAP Forum μετά από τους στόχους που ετέθησαν και οι οποίοι ήταν οι εξής:

- Να εξασφαλιστεί η συμβατότητα των προϊόντων που παράγονται από εταιρίες της ασύρματης βιομηχανίας, έτσι ώστε να υπάρχει ανάπτυξη της αγοράς.
- Να γίνει δυνατή η πρόσβαση σε υπηρεσίες του Internet από τις κινητές συσκευές.
- Να δημιουργηθεί ένα παγκόσμια αποδεκτό πρωτόκολλο έτσι ώστε να είναι συμβατό σε όλες τις τεχνολογίες των δικτύων.

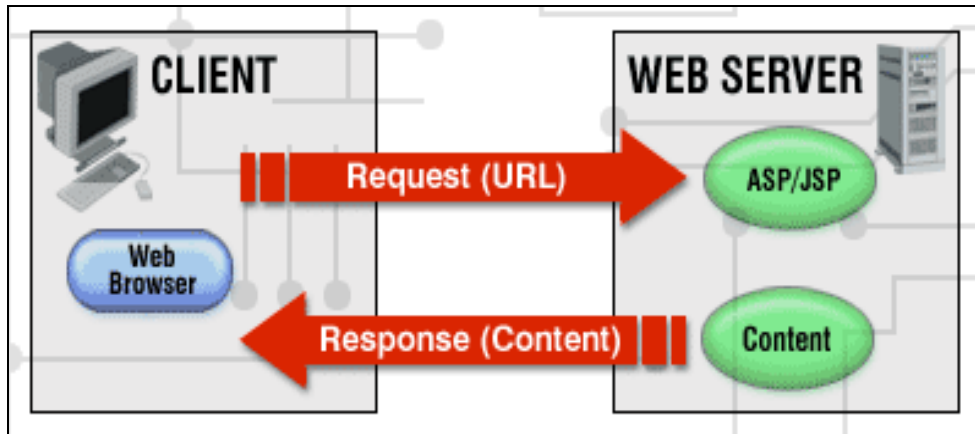
Είναι προφανές λοιπόν ότι το πρωτόκολλο συνδέεται άμεσα με τις τεχνολογίες του Internet και αποτελεί τον συνδετικό κρίκο μεταξύ αυτών των τεχνολογιών και της ασύρματης μετάδοσης της πληροφορίας, όπως χαρακτηριστικά απεικονίζει η παρακάτω εικόνα.



Εικόνα 29: Η σχέση του WAP με τις υπόλοιπες τεχνολογίες

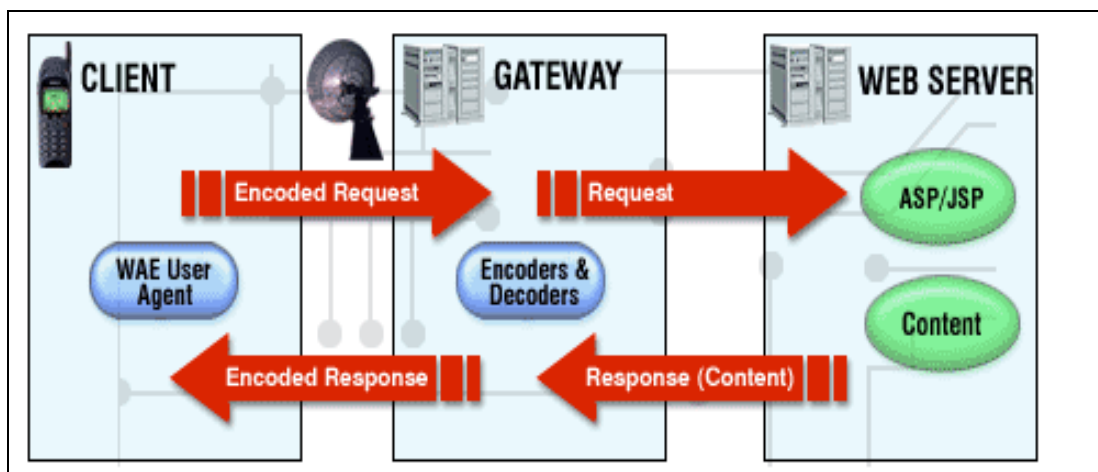
### 1.2.3.2.2 Αρχιτεκτονική

Στον χώρο του Internet, το μοντέλο που ακολουθείται στην παράδοση της πληροφορίας είναι αυτό του εξυπηρετούμενου – εξυπηρετή (client-server). Η παρακάτω εικόνα απεικονίζει αυτόν τον τρόπο μεταξύ του εξυπηρετή (web server) και του εξυπηρετούμενου (client). Ο εξυπηρετούμενος κάνει μία αίτηση (request) προς τον εξυπηρετή που με τη σειρά του απαντά με το περιεχόμενο (response).



Εικόνα 30: Το μοντέλο client-server στο Internet

Η στενή διασύνδεση του WAP με τον χώρο του Internet έχει σαν αποτέλεσμα η γενική αρχιτεκτονική να είναι παρόμοια με αυτήν της παραπάνω εικόνας, με μερικά επιπρόσθετα στοιχεία, όπως φαίνεται. Η ύπαρξη μίας πύλης (gateway), ή αλλιώς WAP proxy, κρίνεται απαραίτητη αφού αποτελεί τον ενδιάμεσο μεταξύ του εξυπηρετούμενου (που τώρα είναι μια ασύρματη συσκευή) και του εξυπηρετή. Η επικοινωνία της πύλης με το τερματικό γίνεται με βάση τα πρωτόκολλα του WAP ενώ με τον εξυπηρετή γίνεται με βάση τα πρωτόκολλα του Internet.



Εικόνα 31: Η θέση του WAP στο μοντέλο client-server



Τα βήματα που ακολουθούνται για την διεκπεραίωση μίας αίτησης είναι τα εξής :

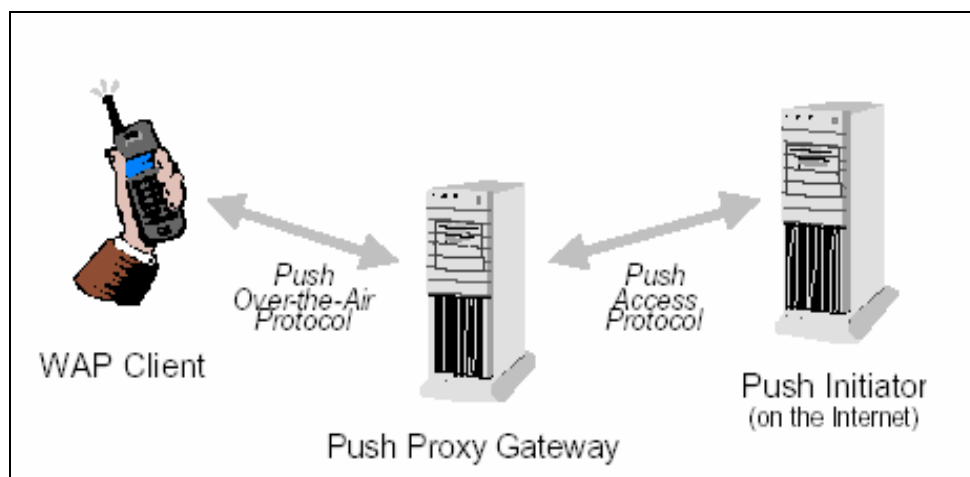
1. Ο εξυπηρετούμενος ξεκινά μια σύνδεση υποβάλλοντας την διεύθυνση IP της πύλης WAP και τη διεύθυνση URL του εξυπηρετή (Web Server). Τα δεδομένα που διακινούνται σε αυτό το στάδιο είναι κωδικοποιημένα.
2. Η πύλη WAP μετά από την επεξεργασία και την αποκωδικοποίηση υποβάλλει ένα HTTP αίτημα προς τον εξυπηρετή (Web Server).
3. Ο εξυπηρετής επιστρέφει στην πύλη τα δεδομένα που αιτήθηκαν σε μορφή WML (Wireless Markup Language).
4. Η πύλη WAP ελέγχει τα WML δεδομένα και τα κωδικοποιεί. Στη συνέχεια στέλνει την απάντηση στον εξυπηρετούμενο.
5. Ο εξυπηρετούμενος λαμβάνει την WAP απάντηση.

Η ύπαρξη ενός WAP Proxy μπορεί να βελτιστοποιεί την επικοινωνία μεταξύ των δύο πλευρών. Αν και η έκδοση WAP 2.0 δεν απαιτεί αυτή τη χρήση ο WAP Proxy είναι αναγκαίος για την λειτουργία της αυτόματης προώθησης, όπως θα αναφερθούμε παρακάτω. [23]

### 1.2.3.2.3 WAP Push

Η λειτουργία αυτόματης προώθησης στο WAP δίνει τη δυνατότητα στον *Push Initiator (PI)* να μεταδώσει *push* περιεχόμενο και οδηγίες παράδοσης σε έναν *Push Proxy Gateway (PPG)*, ο οποίος στη συνέχεια πρέπει να παραδώσει το περιεχόμενο στον πελάτη (Wap client) σύμφωνα με τις οδηγίες παράδοσης.

Ο PI είναι μια εφαρμογή που λειτουργεί σε έναν συνηθισμένο web server και επικοινωνεί με τον PPG χρησιμοποιώντας το πρωτόκολλο *Push Access Protocol (PAP)*. Ο PPG, με τη σειρά του, χρησιμοποιεί το πρωτόκολλο *Push Over-The-Air (OTA)* για να παραδώσει το *push* περιεχόμενο στον πελάτη όπως φαίνεται και στην εικόνα.



Εικόνα 32: Τα μέρη του WAP Push

## α) Push Access Protocol (PAP)

Το πρωτόκολλο PAP χρησιμοποιείται για την επικοινωνία ανάμεσα στον Push Initiator που ξεκινά την αποστολή μηνυμάτων και τον υπολογιστή που επεξεργάζεται τα Push μηνύματα πριν τα στείλει στους τελικούς χρήστες (Push Proxy Gateway). Το πρωτόκολλο αυτό έχει σχεδιαστεί έτσι ώστε να είναι ανεξάρτητο από το υποκείμενο πρωτόκολλο μεταφοράς και μπορεί να χρησιμοποιηθεί ουσιαστικά πάνω από οποιοδήποτε πρωτόκολλο που επιτρέπει τη μεταφορά MIME τύπων δεδομένων μέσω του Internet. Το αρχικό πρωτόκολλο που έχει καθοριστεί ως πρωτόκολλο μεταφοράς για το PAP είναι το HTTP αλλά μπορούν να χρησιμοποιηθούν και άλλα όπως για παράδειγμα το SMTP. Το PAP πρωτόκολλο χρησιμοποιείται για να μεταφέρει εκτός από το περιεχόμενο που θα λάβει ο χρήστης και πληροφορίες ελέγχου τις οποίες χρησιμοποιεί ο PPG. Η πληροφορία αυτή εκφράζεται χρησιμοποιώντας τη γλώσσα XML [24].

### Λειτουργίες του PAP

Το PAP πρωτόκολλο καθορίζει τις ακόλουθες λειτουργίες :

- **Push Submission:**

Ο σκοπός της λειτουργίας αυτής είναι η παράδοση ή η αντικατάσταση ενός Push μηνύματος από έναν PI σε έναν PPG ο οποίος θα πρέπει να το παραδώσει στον user-agent μιας συσκευής στο ασύρματο δίκτυο.

- **Result Notification:**

Η λειτουργία αυτή είναι υποχρεωτική σε έναν PPG για να ενημερώσει τον PI για το τελικό αποτέλεσμα μιας αίτησης αποστολής μηνύματος, εφόσον έχει ζητηθεί από τον PI.

- **Push Cancellation:**

Τη λειτουργία αυτή την ξεκινά ο Push Initiator με μια αίτηση που επιχειρεί να ακυρώσει την αποστολή ενός προηγούμενου μηνύματος. Ο PPG απαντά αν η αίτηση ήταν επιτυχής ή όχι. Η πληροφορία διακινείται με XML έγγραφα.

- **Push Replacement:**

Η λειτουργία της αίτησης για αποστολή ενός Push μηνύματος είναι δυνατόν να προκαλέσει, εφόσον το ζητήσει ο PI, την αντικατάσταση ενός προηγούμενου μηνύματος. Είναι δυνατόν να προσδιοριστεί αν το νέο μήνυμα θα πρέπει να σταλεί σε αυτούς τους παραλήπτες που δεν έχουν λάβει το αρχικό μήνυμα ή αν το νέο μήνυμα θα πρέπει να σταλεί σε όλους. Σε κάθε περίπτωση το αρχικό μήνυμα ακυρώνεται για εκείνους τους παραλήπτες στους οποίους δεν έχει παραδοθεί.

- **Status Query:**

Η λειτουργία αυτή επιτρέπει στον PI να ζητήσει την τρέχουσα κατάσταση ενός μηνύματος που έχει σταλεί προηγουμένως.

- **Client Capabilities Query:**

Ο PI ρωτά τον PPG για τις δυνατότητες μιας συγκεκριμένης συσκευής [25].

## Διευθυνσιοδότηση

Γενικά οι διευθύνσεις που χρησιμοποιούνται στην όλη διαδικασία είναι οι εξής:

- Η διεύθυνση του PPG
- Η διεύθυνση της ασύρματης συσκευής
- Η διεύθυνση του PI

Ειδικά για την διεύθυνση της ασύρματης συσκευής ακολουθείται η παρακάτω μορφή:

*WAPPUSH=<διεύθυνση\_χρήστη\_ή\_συσκευής>/TYPE=<τύπος διεύθυνσης>@<προσδιοριστής\_PPG>*

όπου <τύπος διεύθυνσης> πρέπει να είναι:

- USER: για έναν “προσδιοριστή χρήστη” (οι προσδιοριστές χρήστη είναι αυθαίρετες τιμές που αντιστοιχούν σε διευθύνσεις ασύρματου δικτύου με τρόπο που δεν είναι καθορισμένος)
- IPv4 ή IPv6: για τις αντίστοιχες διευθύνσεις
- PLMN (Public Land Mobile Network): για διευθύνσεις που είναι τηλεφωνικοί αριθμοί
- <προσδιοριστής\_PPG> πρέπει να είναι:
- ο προσδιοριστής του ονόματος του domain με μορφή ppg.φορέας.com που καθορίζει τον υπολογιστή όπου βρίσκεται ο PPG. [25]

## Η μορφή του Push μηνύματος

Κάθε Push μήνυμα περιέχει επικεφαλίδες και σώμα. Πιο συγκεκριμένα το μήνυμα αυτό περιέχει μια οντότητα ελέγχου (control entity), μια οντότητα περιεχομένου (content entity) και προαιρετικά μια οντότητα δυνατοτήτων (capabilities entity).

1. Το *control entity* είναι ένα XML έγγραφο που περιέχει πληροφορίες ελέγχου τις οποίες χρησιμοποιεί ο PPG για να επεξεργαστεί το μήνυμα για παράδοση.
2. Το *content entity* αντιπροσωπεύει το περιεχόμενο που θα σταλεί στην ασύρματη συσκευή σε WML.
3. Το *capabilities entity* περιέχει τις δυνατότητες του τερματικού που υποθέτει ο PI. Ο PPG μπορεί να χρησιμοποιήσει την πληροφορία αυτή για να επικυρώσει ότι το μήνυμα είναι κατάλληλο για αυτό το τερματικό

## β) Push Proxy Gateway (PPG)

Ο PPG παίζει πολύ σημαντικό ρόλο στο μοντέλο του Wap Push. Οι σημαντικότερες λειτουργίες του μπορούν να συνοψισθούν ως εξής:

- Αναγνώριση και εξακρίβωση της ταυτότητας ενός Push Initiator
- Έλεγχος πρόσβασης
- Υπηρεσίες προσδιορισμού της ταυτότητας του χρήστη
- Αποδοχή και απόρριψη μηνυμάτων

- Υπηρεσία διαχείρισης μηνύματος
- Κωδικοποίηση, Μεταγλώττιση και Συμπίεση
- Μετάδοση σε έναν ή περισσότερους χρήστες και μετατροπές διευθύνσεων
- Υπηρεσία επερώτησης των δυνατοτήτων του τερματικού [23]

### γ) Push Over-The-Air Protocol

Το PAP χρησιμοποιείται για την μεταφορά των δεδομένων από τον *PI* στον *PPG* ενώ το Push Over-The-Air (OTA) πρωτόκολλο είναι το τμήμα του Push μοντέλου που είναι υπεύθυνο για τη μεταφορά του περιεχομένου από τον PPG στο τερματικό του χρήστη. Μπορεί να λειτουργήσει πάνω από το HTTP (OTA-HTTP) ή το WSP (Wireless Session Protocol) γνωστότερο και σαν OTA-WSP [23].

### 1.2.3.3 Σύγκριση SMS-WAP

Έχοντας δει τους δύο τύπους των διεπαφών που μπορούν να χρησιμοποιηθούν στις Υπηρεσίες Εξαρτώμενες από τη Θέση, είναι η κατάλληλη στιγμή να τους συγκρίνουμε και να αναφέρουμε τα πλεονεκτήματα και τα μειονεκτήματά τους καθώς επίσης να αναφερθούμε και σε νέες τεχνολογίες που εμφανίζονται στο προσκήνιο. Η υπηρεσία SMS έχει πολλά πλεονεκτήματα εξαιτίας των οποίων έγινε δημοφιλής:

- Χαμηλή χρέωση
- Οικειότητα
- Κατάλληλο για απλά μηνύματα-ειδοποιήσεις χωρίς μεγάλο φόρτο στο δίκτυο
- Καταξιωμένο εμπορικά
- Δεν προϋποθέτει έξτρα εξοπλισμό στο κινητό

Όμως η τεχνολογία WAP υπερτερεί σημαντικά έναντι της υπηρεσίας SMS σε μερικούς τομείς. Τα πλεονεκτήματα αυτά αναλύονται ευθύς αμέσως.

#### Ενεργό Περιεχόμενο (Active Content)

Το κυριότερο πλεονέκτημα είναι η ικανότητα να σταλεί στο κινητό ένα «ενεργό» περιεχόμενο δηλαδή ένα περιεχόμενο που δίνει στον χρήστη τη δυνατότητα να ενεργοποιήσει εφαρμογές όπως το άνοιγμα μιας εικόνας ή την επίσκεψή του σε ένα URL. Καθιστά το κινητό έναν διαφυλλιστή (micro-browser) που εμφανίζει το περιεχόμενο στην γλώσσα WML.

#### Ευρύ σύνολο δυνατοτήτων

Το πρωτόκολλο PAP προσφέρει ένα ευρύ σύνολο δυνατοτήτων οι οποίες δεν είναι διαθέσιμες για το SMS όπως:

- Δυνατότητα πολλαπλής αποστολής του ίδιου μηνύματος (Multi-recipient addressing).
- Δυνατότητα αποστολής μηνύματος χρησιμοποιώντας αριθμό τηλεφώνου ή μία εναλλακτική διεύθυνση ορισμένη από το χρήστη (alternate addressing).
- Δυνατότητα εκτέλεσης συγκεκριμένων εφαρμογών στο τερματικό του χρήστη.
- Έλεγχος αν κάποια ενέργεια του χρήστη διεκόπη.
- Δυνατότητα επερώτησης των τεχνικών ικανοτήτων του τερματικού έτσι ώστε να προωθήσει κατάλληλα το μήνυμα.
- Δυνατότητα αποστολής μηνυμάτων MIME .
- Προτεραιότητα αποστολής.
- Αντικατάσταση μηνύματος που έχει ήδη σταλεί.

### **Έλεγχος κρυφής μνήμης (Cache Control)**

Ένα επίσης πλεονέκτημα είναι η δυνατότητα ελέγχου της μνήμης αποθήκευσης στο τερματικό. Αυτό επιτρέπει την επαναχρησιμοποίηση προηγούμενων δεδομένων που έχει ήδη επισκεφθεί ο χρήστης με την προϋπόθεση ότι αυτά δεν έχουν αλλάξει. Αν π.χ. ένα μήνυμα MIME αποτελείται από ένα μίγμα SI (Service Indication) και WML τότε μόνο το SI θα παρουσιαστεί στο χρήστη ενώ τα άλλα δεδομένα θα αποθηκευτούν στην μνήμη.

### **Μηχανισμοί παράδοσης (delivery mechanisms)**

Οι μηχανισμοί παράδοσης μπορεί να χρησιμοποιήσουν SMS σαν μέσο επικοινωνίας με το τερματικό ή να επικοινωνήσουν απευθείας. Αν χρησιμοποιηθεί SMS τότε ο PPG έχει την δυνατότητα να τμηματοποιήσει ένα μεγάλο μήνυμα σε μικρότερα τα οποία στη συνέχεια μετά την αποστολή επανενώνονται. Επίσης ο PPG επιτρέπει λειτουργίες όπως:

- Ταυτοποίηση του χρήστη
- Αναφορά παράδοσης
- Αποστολή πάνω από ασφαλές κανάλι (secure channel) [26]

### **1.2.3.4 Νέες τεχνολογίες διεπαφών (VoiceXML)**

Η VoiceXML συνδυάζει την XML με την αναγνώριση φωνής προσφέροντας έτσι μία διαδραστική (interactive) πρόσβαση μέσω του τερματικού στο Internet. Μετά από μία συνεργασία των AT&T, IBM, Lucent Technologies και Motorola, η VoiceXML 1.0 υιοθετήθηκε τον Μάρτιο του 2000 από το World Wide Web Consortium. Μερικές από τις εφαρμογές της τεχνολογίας αυτής είναι :

- Διασκέδαση (Entertainment) όπως παιχνίδια, ωροσκόπιο.
- Υπηρεσίες καταλόγου (Enhanced directory services) που παρέχουν αριθμούς τηλεφώνου και δίνουν οδηγίες.
- Οικονομικές υπηρεσίες (Financial services) όπως χρηματιστήριο.

- Υπηρεσίες ταξιδιών όπως ώρες πτήσεων ,καθυστερήσεις κ.τ.λ.

## **Λειτουργία**

Χρησιμοποιείται ένα τηλέφωνο για σύνδεση στην Φωνητική Πύλη (Voice Portal) που είναι σαν μία παραδοσιακή πύλη με την διαφορά ότι η περιήγηση γίνεται με την φωνή και όχι με το ποντίκι. Στη συνέχεια η φωνή μεταφράζεται από το κατάλληλο λογισμικό αναγνώρισης φωνής και επεξεργάζεται από τον Web Server. Τέλος ο χρήστης ακούει τις οδηγίες που επιστρέφονται. Γνωστά παραδείγματα μίας τέτοιας τεχνολογίας στη χώρα μας είναι η φωνητική πύλη MyCosmos.

Βέβαια υπάρχουν ακόμη πολλά προβλήματα που σχετίζονται κυρίως με την αναγνώριση φωνής και την επεξεργασία ώστε να προκύψουν οι κατάλληλες εντολές. Έρευνα της Cahners In-State Group δείχνει οι συναλλαγές που θα σχετίζονται με τις Φωνητικές Πύλες θα ξεπερνούν τα \$1.6 δισεκατομμύρια το 2005 [27]

## **1.2.4 Η Τυποποίηση στις Υπηρεσίες Θέσης**

### **1.2.4.1 Η σημασία των Ανοικτών Προτύπων**

Σήμερα τον κυριότερο ρόλο σε ένα σύστημα παροχής LBS υπηρεσιών διαδραματίζει ο παροχέας του ασύρματου δικτύου. Αυτός χρησιμοποιεί ένα μη τυποποιημένο (proprietary) υποσύστημα εντοπισμού θέσης που μπορεί να χρησιμοποιεί οποιαδήποτε από τις τεχνικές που προαναφέρθηκαν. Η ανταλλαγή δεδομένων μεταξύ των υποσυστημάτων (GIS, βάσεις δεδομένων, υποσυστήματα χρέωσης) γίνεται επίσης με μη τυποποιημένες (proprietary) διεπαφές. Τέλος, οι παροχείς εφαρμογών (ASPs) αναπτύσσουν τις υπηρεσίες τους σε πλατφόρμες που τους δίνονται έτοιμες και είναι διαφορετικές για κάθε δίκτυο. Από όλα τα παραπάνω γίνεται αντιληπτό ότι η συμβατότητα μεταξύ διαφορετικών συστημάτων μπορεί να υλοποιηθεί μόνο με «κατά περίπτωση» λύσεις. Αυτό σημαίνει την δημιουργία και υποστήριξη πολλών διαφορετικών τεχνολογιών και πρωτοκόλλων. Όμως αυτό δεν είναι σε καμία περίπτωση η καλύτερη λύση αφού το κόστος τέτοιων λύσεων είναι απαγορευτικό. Η βιομηχανία των ασύρματων υπηρεσιών όμως, χάρη στην εμπειρία από άλλους τομείς της πληροφορικής και των τηλεπικοινωνιών, εντόπισε το ανωτέρω πρόβλημα και δραστηριοποιήθηκε για την δημιουργία και κοινή υιοθέτηση Ανοικτών Προτύπων (Open Standards).

Οι ιδιότητες αυτών των προτύπων, που αποτελούν ταυτόχρονα και τα πλεονεκτήματά τους, είναι σχετικά γνωστές και συνοψίζονται στις παρακάτω:

- *Δημόσια και μη κερδοσκοπική διαθεσιμότητα:* Όλοι έχουν το δικαίωμα να υλοποιούν τα Ανοικτά Πρότυπα χωρίς την καταβολή (σημαντικού) αντιτίμου.
- *Μεγιστοποίηση των επιλογών του τελικού χρήστη:* Με την χρήση τους δημιουργείται μια πλουραλιστική αγορά προϊόντων και υπηρεσιών που χαρακτηρίζεται από κυμαινόμενες και ανταγωνιστικές τιμές.

- *Έλλειψη διάκρισης:* Οι οργανισμοί που διαχειρίζονται τα Ανοικτά Πρότυπα δεν υποστηρίζουν ιδιαίτερα και αθέμιτα κανέναν από αυτούς που υλοποιούν τα πρότυπα. Το μόνο που παρέχουν είναι αντικειμενική πιστοποίηση συμβατότητας μεταξύ των προτύπων και των εφαρμογών.

Έτσι με την χρήση των Ανοικτών Προτύπων επιτυγχάνεται θεμιτός και «υγιής» οικονομικός ανταγωνισμός που έχει σαν αποτέλεσμα την μείωση του κόστους χρήσης των υπηρεσιών για τον τελικό χρήστη, την μείωση της υλοποίησης για τον παροχέα και την βελτίωση των παρεχόμενων υπηρεσιών. Επιπλέον, κάθε παροχέας υπηρεσιών έχει κέρδη ανάλογα των υπηρεσιών που παρέχει και όχι ανάλογα της προσπάθειάς του να αποκλείσει άλλους παροχείς από την αγορά (δημιουργία μονοπωλίων).

Ειδικότερα για τις υπηρεσίες θέσης, τα πλεονεκτήματα της επικράτησης Ανοικτών Προτύπων θα είναι:

- Όλες οι κινητές συσκευές θα λειτουργούν παντού και κάθε ώρα.
- Υποστήριξη όλων των τεχνικών εντοπισμού θέσης.
- Καλά καθορισμένες μέθοδοι *παραγωγής* μεταξύ διαφορετικών δικτύων.
- Υποστήριξη *ασφάλειας* και *χρέωσης* σε εθνικό (και ίσως διεθνές) επίπεδο.
- *Συμβατότητα* μεταξύ δικτύων διαφορετικών γενιών (2G, 2.5G, 3G).
- Εύκολη συνεργασία σε επίπεδο ανάπτυξης εφαρμογών και υπηρεσιών.
- Υποστήριξη *πολλών επιχειρηματικών μοντέλων* και *επιλογή* του καταλληλότερου.

Σήμερα στην συγκεκριμένη βιομηχανία χρησιμοποιούνται αρκετά Ανοικτά Πρότυπα και Ανοικτές Τεχνολογίες με αποδεδειγμένη αξία. Μερικά από αυτά είναι τα εξής:

- Τα πρωτόκολλα του Internet (TCP/IP, HTTP και άλλα πρωτόκολλα του W3C και του IETF)
- Η γλώσσα προγραμματισμού Java
- Η πλατφόρμα ανάπτυξης επιχειρηματικών εφαρμογών J2EE
- Η πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές J2ME
- Η γλώσσα XML και πολλές επεκτάσεις της (GML κα)
- Το πρωτόκολλο WAP
- Το σύστημα GSM και συστήματα που προτείνονται από οργανισμούς όπως οι 3GPP και ETSI
- Τα επιμέρους πρωτόκολλα που προτείνονται από οργανισμούς όπως ο OGC, LIF κλπ.
- Πρωτόκολλα και συστήματα που προτείνονται από τον οργανισμό IEEE (πχ 802.11)
- Το πρωτόκολλο-διεπαφή OSA/Parlay

Βασικά οι πιο πολλοί οργανισμοί που ασχολούνται με συναφή θέματα προσπαθούν να χρησιμοποιήσουν και να επεκτείνουν με τον καλύτερο δυνατό τρόπο τις τεχνολογίες του Internet. Αυτό συμβαίνει όχι μόνο επειδή είναι ήδη υλοποιημένες αλλά

και επειδή έχουν χρησιμοποιηθεί επιτυχώς. Φυσικά η αξιοποίησή τους δεν θα ήταν δυνατή αν δεν ήταν και οι ίδιες «ανοικτές».

#### **2.2.4.2 Οι Οργανισμοί Τυποποίησης**

Τα πλεονεκτήματα που αναφέρθηκαν προηγούμενα αποτελούν οράματα για το μέλλον των υπηρεσιών που εξαρτώνται από την θέση. Όμως έχουν ήδη αρχίσει οι διαδικασίες και έχουν τεθεί οι βάσεις για την δημιουργία και χρήση εξειδικευμένων Ανοικτών Προτύπων, που θα καθορίζουν τις διεπαφές μεταξύ των υποσυστημάτων που αυτές απαιτούν για την υλοποίησή τους. Κεντρικής σημασίας οντότητες για αυτή την πρωτοβουλία είναι οι οργανισμοί που συγκροτήθηκαν για την ανάπτυξη της επιθυμητής συμβατότητας (interoperability). Οι σημαντικότεροι από αυτούς είναι ο LIF [28] (Location Inter-Operability Forum) και ο OGC [29] (Open GIS Consortium). Άλλοι οργανισμοί είναι ο WLIA [30] (Wireless Location Industry Association) και ο OMA [31] (Open Mobile Alliance).

Ο LIF είναι μια παγκόσμια πρωτοβουλία που σχηματίστηκε το Σεπτέμβριο του 2000 με σκοπό την ανάπτυξη και προώθηση κοινών λύσεων για την βιομηχανία των υπηρεσιών θέσης. Οι προτάσεις του LIF είναι ανεξάρτητες και από δικτυακά πρωτόκολλα και από τεχνολογίες εντοπισμού θέσης. Κύριος στόχος του είναι να υποδεικνύει και να αντιμετωπίζει προβλήματα συμβατότητας και να επιταχύνει την ανάπτυξη και εμπορική εκμετάλλευση των υπηρεσιών αυτών. Ο LIF επικεντρώνει το ενδιαφέρον του στο πως πρέπει να είναι ένα ενδιάμεσο λογισμικό (middleware) που θα προσφέρει ομοιόμορφες διεπαφές ανάμεσα στους παροχείς των δικτύων και στους υπόλοιπους «παίκτες». Επίσης αναπτύσσει διεπαφές σε επίπεδο εφαρμογής. Το σημαντικότερο έργο του είναι το πρότυπο MLP (Mobile Location Protocol) το οποίο θα περιγραφεί παρακάτω. Οποιαδήποτε εταιρία μπορεί να συμμετάσχει στις δραστηριότητές του, συμπεριλαμβανομένων παροχέων περιεχομένου και εφαρμογών, εταιριών δικτυακής υποδομής, κατασκευαστών τερματικών συσκευών και παροχέων κινητών επικοινωνιών. Σήμερα ανήκουν σε αυτόν πάνω από 150 μέλη τα οποία συμμετέχουν και σε άλλους οργανισμούς της ασύρματης βιομηχανίας, όπως ο OGC. Ο LIF τελεί υπό καθεστώς συνένωσης με τον OMA.

Ο OGC είναι ένας διεθνής οργανισμός που τον αποτελούν πάνω από 250 μέλη (εταιρίες, κυβερνητικοί οργανισμοί και πανεπιστήμια). Τα Ανοικτά Πρότυπα και οι διεπαφές που δημοσιεύει, έχουν σαν στόχο να δώσουν την δυνατότητα στον Παγκόσμιο Ιστό (WWW), στις LBS υπηρεσίες και γενικότερα στις ασύρματες υπηρεσίες, να αξιοποιήσουν με έναν ομοιόμορφο τρόπο τα GIS συστήματα που υπάρχουν. Έτσι βοηθάει όσους αναπτύσσουν εφαρμογές, να προσπελάσουν πολύπλοκες χωρικές πληροφορίες. Άρα ενώ ο LIF ασχολείται κυρίως με την συμβατότητα σε επίπεδο δικτύου, ο OGC ασχολείται με την συμβατότητα σε επίπεδο εφαρμογών, παρουσίασης και δεδομένων.

Στα πλαίσια των δραστηριοτήτων του ο OGC, δημιούργησε μια «πρωτοβουλία» (initiative) με την ονομασία OpenLS (Open Location Services) [32]. Αυτή είναι αφιερωμένη στην ανάπτυξη προτύπων που θα καθιστούν δυνατή την χρήση της θέσης και άλλες χωρικές πληροφορίες από το περιβάλλον του ασύρματου Internet. Ο κύριος



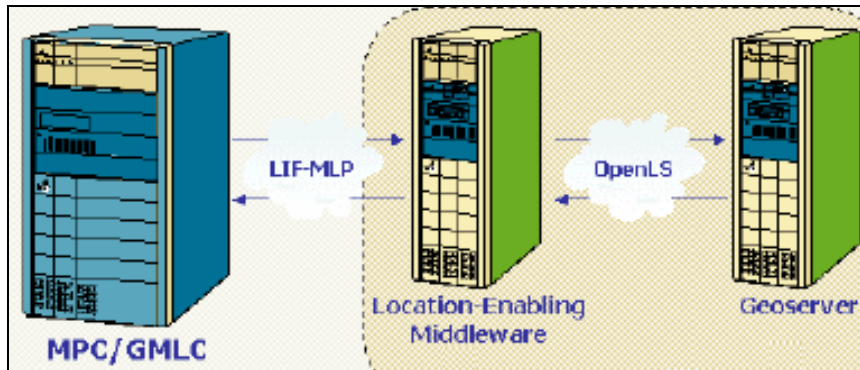
σκοπός της είναι να παράγει Ανοικτά Πρότυπα που θα αξιοποιηθούν για συμβατές υπηρεσίες θέσης. Τα πρότυπα αυτά είναι είτε πρωτόκολλα είτε διεπαφές. Το OpenLS API είναι μια διεπαφή που φιλοδοξεί να αντικαταστήσει όλες τις άλλες εμπορικές και μη τυποποιημένες διεπαφές που έχουν χρησιμοποιηθεί μέχρι σήμερα, για την αλληλεπίδραση των GIS servers (geoservers) με το λογισμικό που παρέχει την θέση (εικόνα 33).

Ένας άλλος οργανισμός που δεν συστάθηκε αρχικά για την δημιουργία προτύπων, αλλά τελικά ασχολείται και με αυτά, είναι ο WLIA. Αυτός είναι η «φωνή» της βιομηχανίας που ασχολείται με τις ασύρματες υπηρεσίες θέσης. Τα μέλη του παρέχουν υλικό (hardware & infrastructure), λογισμικό, υπηρεσίες και άλλα προϊόντα που σχετίζονται με την θέση του κινητού χρήστη. Ο WLIA προσφέρει αναφορές (references) και πληροφόρηση τόσο στο ευρύ κοινό όσο και στις εταιρίες. Είναι υπεύθυνος για την εκπροσώπηση των μελών του σε νομικούς, ρυθμιστικούς και διαχειριστικούς φορείς. Επίσης εξυπηρετεί σαν ένα φόρουμ για την ανάπτυξη προτύπων και προδιαγραφών που αφορούν σε σημαντικά θέματα, όπως για παράδειγμα η ιδιωτικότητα (privacy).

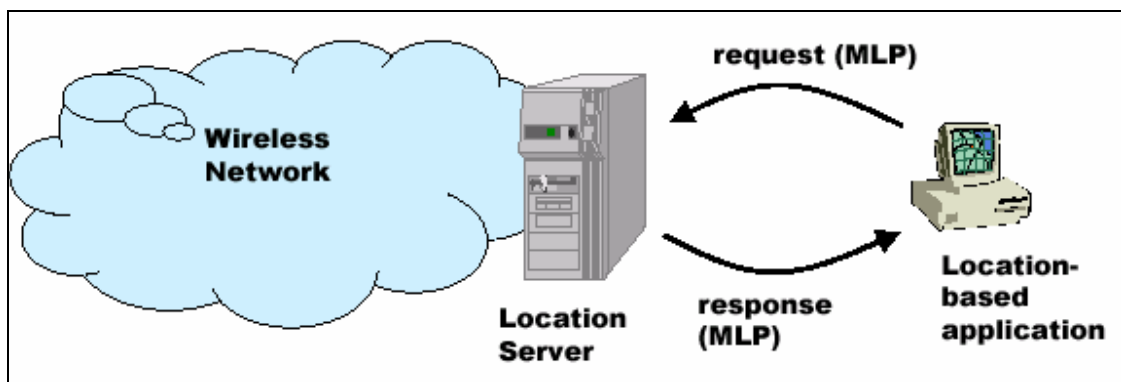
Τέλος ο OMA είναι ένας οργανισμός που επιδιώκει την άρση των εμποδίων που καθιστούν σήμερα τις κινητές επικοινωνίες ανίκανες να λειτουργήσουν σε παγκόσμια κλίμακα. Γι' αυτό το σκοπό ο OMA δημιουργεί υψηλής ποιότητας Ανοικτά Πρότυπα και προδιαγραφές, σύμφωνα με τις απαιτήσεις της αγοράς. Επίσης παρέχει ελέγχους συμβατότητας. Οι αρχές που τον διέπουν, εκτός από τα Ανοικτά Πρότυπα, είναι ότι το επίπεδο εφαρμογής πρέπει να είναι ανεξάρτητο του φορέα (GSM, GPRS, EDGE), ανεξάρτητο από το Λειτουργικό Σύστημα και πρέπει να υπάρχει πλήρης συμβατότητα μεταξύ των πλατφόρμων και των εφαρμογών ώστε να μπορεί να υποστηριχθεί διεθνής περιαγωγή (roaming). Ας σημειωθεί εδώ ότι ο OMA δεν είναι ένας οργανισμός που ιδρύθηκε για τις ανάγκες των υπηρεσιών θέσης, όπως οι προηγούμενοι, αλλά προϋπήρχε αυτών και απλώς τις ενσωμάτωσε εφόσον κι αυτές ανήκουν στην ασύρματη βιομηχανία.

### **1.2.4.3 Mobile Location Protocol**

Σαν δείγμα ενός Ανοικτού Προτύπου θα παρουσιάσουμε το Mobile Location Protocol (MLP) που προτάθηκε και αναπτύσσεται από τον LIF. Σήμερα έχει φτάσει αισίως στην 3η έκδοσή του. Το MLP είναι ένα πρωτόκολλο επιπέδου εφαρμογής, που χρησιμοποιείται για να ληφθεί η θέση των κινητών σταθμών (κινητά τηλέφωνα, PDAs κα). Ο σκοπός του είναι να ορίσει μια απλή και ασφαλή μέθοδο πρόσβασης (διεπαφή) που να επιτρέπει στις εφαρμογές θέσης, να αναζητούν χωρικές πληροφορίες από ένα ασύρματο δίκτυο, ανεξάρτητα από τις διεπαφές ασύρματης διάδοσης και τις τεχνικές εντοπισμού θέσης (εικόνας 33 και 34). Οι προδιαγραφές του πρωτοκόλλου καθορίζουν ένα API (Application Programming Interface) για τον εξυπηρέτη θέσης (location server), είτε αυτός είναι ένα GMLC (Gateway Mobile Location Center, στα δίκτυα GSM, GPRS, UMTS) είτε ένα MPC (Mobile Positioning Center, στα ANSI δίκτυα). Το API αυτό βασίζεται σε τεχνολογίες του Internet, όπως το HTTP, το SSL/TLS και η XML.



Εικόνα 33: Ο ρόλος του MLP και του OpenLS API



Εικόνα 34: Η χρήση του MLP

Μερικά άλλα χαρακτηριστικά του πρωτοκόλλου αυτού είναι η υποστήριξη ιδιωτικότητας και πιστοποίησης (authentication) για να διασφαλιστεί ότι προστατεύονται τα προσωπικά δεδομένα των χρηστών, δηλαδή η ακριβής θέση τους, και ότι δεν θα φτάσουν σε μη εξουσιοδοτημένες οντότητες. Η χρήση του MLP μειώνει το χρόνο ανάπτυξης και διάθεσης των υπηρεσιών θέσης στους συνδρομητές. Επίσης οι εταιρίες που ασχολούνται με τις παραπάνω υπηρεσίες και που θα υιοθετήσουν το MLP μπορούν να ελπίζουν σε καλύτερη συνεργασία με άλλες εταιρίες του χώρου.

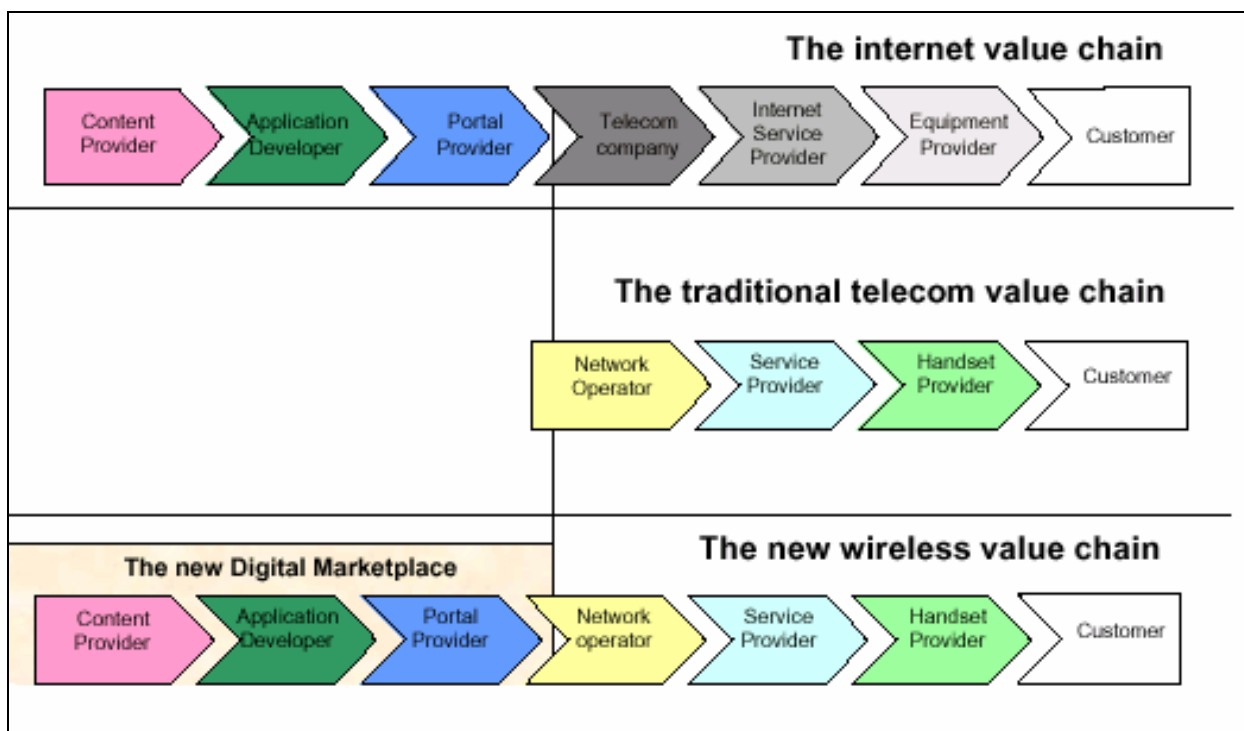
## 1.3 Ειδικά Θέματα Διαχείρισης και Επιχειρηματικής Πολιτικής

### 1.3.1 Επιχειρηματικά Θέματα

#### 1.3.1.1 Επιχειρηματικοί ρόλοι

Ένα μεγάλο μέρος των ασύρματων επικοινωνιών, άρα και των υπηρεσιών θέσης, έχει να κάνει με τα επιχειρηματικά μοντέλα (business models) δηλαδή με τις σχέσεις που αναπτύσσονται μεταξύ των διαφόρων «παικτών», καθένας από τους οποίους προσφέρει συγκεκριμένες υπηρεσίες και σε συνδυασμό συνθέτουν την απαιτούμενη από τον τελικό χρήστη LBS υπηρεσία.

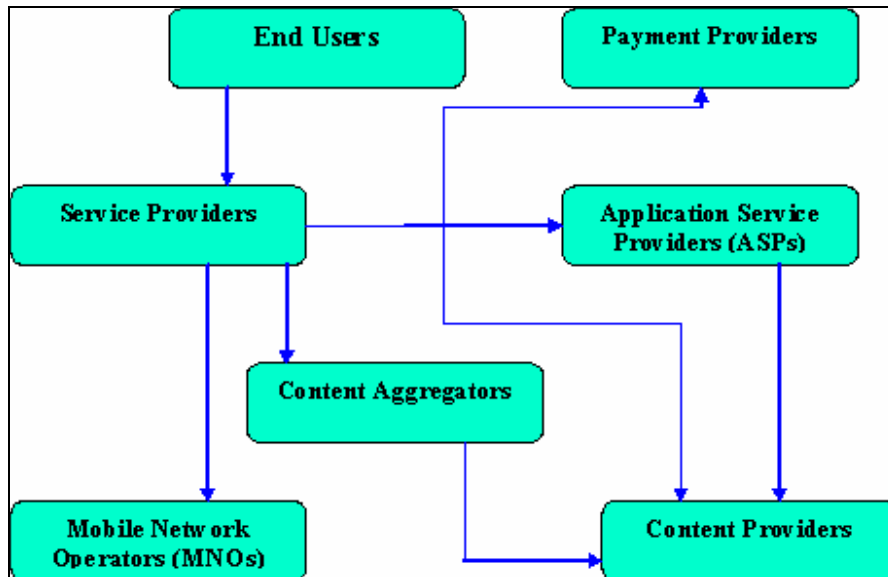
Τον τελευταίο καιρό, με την ανάπτυξη του m-commerce, εμφανίζονται στο προσκήνιο νέες εταιρείες οι οποίες φιλοδοξούν να παίξουν σημαντικό ρόλο στην νέα αλυσίδα αξίας (value chain) που προκύπτει από τον συνδυασμό ήδη υπάρχοντων (Εικόνα 35).



Εικόνα 35: Η δημιουργία της αλυσίδας αξίας για τις νέες ασύρματες υπηρεσίες

Ουσιαστικά θα υπάρχει ένας διαχωρισμός μεταξύ των ρόλων που θα αναλαμβάνει κάθε εταιρεία. Το παρακάτω σχήμα δείχνει πως τα διαφορετικά μέρη συνδυάζονται και

επικοινωνούν μεταξύ τους έχοντας ξεχωριστό ρόλο. Ακόμη βέβαια δεν είναι ξεκάθαρο πως θα υλοποιηθούν αυτές οι σχέσεις και οι συμφωνίες.



Εικόνα 36: Οι ρόλοι στις υπηρεσίες προστιθέμενης αξίας

Τα βέλη με μπλε χρώμα παριστάνουν ροές εισοδήματος (revenue streams) μεταξύ των ρόλων ως αντίτιμο προσφερόμενων υπηρεσιών [33]. Ακολουθεί μια σύντομη περιγραφή των κυριότερων οντοτήτων που εμπλέκονται στα επιχειρηματικά μοντέλα.

### Τελικοί Χρήστες (End Users)

Οι τελικοί χρήστες/πελάτες είναι ο πιο σημαντικός παράγοντας στην όλη υπόθεση διότι από αυτούς εξαρτάται αν μία υπηρεσία θα γίνει αποδεκτή από το ευρύ κοινό και θα αποφέρει κέρδη στις εταιρείες που την προωθούν. Οι πελάτες είναι αυτοί που συντηρούν ουσιαστικά την αλυσίδα αξίας (value chain), αφού έμμεσα τα εισοδήματα όλων των άλλων που συνιστούν την αλυσίδα, προέρχονται από αυτούς. Βέβαια υπάρχουν αρκετές απαιτήσεις εκ μέρους τους για θέματα ασφαλείας και εμπιστευτικότητας.

### Παροχέας κινητού δικτύου (Mobile Network Operator)

Όπως προαναφέρθηκε, ο φορέας του δικτύου σήμερα επιτελεί ένα ιδιαίτερο ρόλο προσπαθώντας να καλύψει τα σημαντικότερα και πιο επικερδή τμήματα της αλυσίδας αξίας, αντικαθιστώντας πολλούς άλλους ρόλους. Θεωρητικά μπορεί να καλύψει όλους τους άλλους ρόλους αλλά κάτι τέτοιο δεν θα οδηγούσε σε υπηρεσίες μεγάλης αξίας και ποιότητας. Στην παρούσα φάση των υπηρεσιών θέσης πάντως, πλεονεκτεί έναντι των άλλων παικτών διότι μπορεί και ελέγχει την πρόσβαση που έχει στην πληροφορία ο τελικός χρήστης.

### Παροχείς Περιεχομένου (Content Providers)

Αποτελούν τις οντότητες που αναλαμβάνουν να βρουν την πληροφορία (περιεχόμενο) που ζητήθηκε και να την προωθήσουν προς τον τελικό χρήστη. Αυτό το

περιεχόμενο μπορεί να είναι ειδήσεις, πληροφορίες καταλόγου, σημεία ενδιαφέροντος, πληροφορίες για ψυχαγωγία/διασκέδαση ή οποιεσδήποτε άλλες πληροφορίες σχετικές με τις LBS υπηρεσίες.

Αυτό γίνεται με το να επικοινωνούν με τους κατόχους της πληροφορίας (content owners) ,να την αγοράζουν και στη συνέχεια να την χρησιμοποιούν. Στις περισσότερες περιπτώσεις, βέβαια, οι ίδιοι αποτελούν και τους κατόχους της πληροφορίας. Η προώθηση της γίνεται απευθείας στους φορείς του δικτύου ή σε συνεργασία με κάποιον content aggregator.

### **Παροχέας Εφαρμογών (Application Service Provider, ASP)**

Αυτό τον ρόλο έχουν οι εταιρείες εκείνες που αναλαμβάνουν την υλοποίηση, φιλοξενία και διαχείριση εφαρμογών για λογαριασμό άλλων που συνιστούν την αλυσίδα αξίας. Πιο συγκεκριμένα, εστιάζουν στη δημιουργία και παροχή πλατφόρμων για LBS υπηρεσίες και γενικά υπηρεσιών κινητής τηλεφωνίας. Έτσι αναλαμβάνουν την φιλοξενία των υπηρεσιών κάνοντας χρήση της δικιάς τους υποδομής. Επίσης παρέχουν συστήματα μηνυμάτων (messaging), διαχείρισης πληρωμών.

Ο λόγος ύπαρξής τους οφείλεται στο γεγονός ότι οι εταιρείες που αγοράζουν τα προϊόντα τους δεν έχουν την εμπειρία αλλά και τους πόρους να δημιουργήσουν τέτοιου είδους εφαρμογές.

Σημαντική πηγή εσόδων για τους ASPs αποτελεί η διαφήμιση. Πολλές διαφημιστικές εταιρείες (sponsors) συνάπτουν συμφωνίες με τους ASPs και τους παρέχεται ο απαιτούμενος διαφημιστικός χώρος για την φιλοξενία. Οι διαφημιστικές αυτές εταιρείες παρεμβάλλονται ουσιαστικά μεταξύ του διαφημιζόμενου και του ASP [34].

### **Παροχείς Υπηρεσιών (Service Providers)**

Είναι αυτοί που παρέχουν την τελική υπηρεσία προστιθέμενης αξίας στον χρήστη και διενεργούν τις τυχόν δοσοληψίες μαζί του. Χρησιμοποιούν τις πλατφόρμες και τις άλλες εφαρμογές που διαθέτουν οι ASPs για να προωθήσουν τις υπηρεσίες τους. Είναι οι μόνοι που είναι ορατοί στον χρήστη εκτός από τον παροχέα του δικτύου.

Ένας τρόπος με τον οποίο μπορούν να προωθήσουν τις υπηρεσίες τους είναι μέσω των Mobile Portals. Αυτά ,ουσιαστικά, αποτελούν μία παραλλαγή των Internet portals (όπως το Yahoo!) και αποτελούν μια ολοκληρωμένη λύση για την παροχή υπηρεσιών ανάκτησης πληροφορίας, συναλλαγών/δοσοληψιών και υπηρεσιών. Το κύριο πλεονέκτημά τους έναντι των Internet Portals είναι η αξιοποίηση της πληροφορίας θέσης.

Τα έσοδα τους λοιπόν συνδέονται άμεσα με τους τελικούς χρήστες και προκύπτουν συνήθως από μηνιαίες συνδρομές, έσοδα από διαφημίσεις αλλά και διάφορες συμφωνίες διαμοιρασμού εσόδων με τους παροχείς περιεχομένου (content providers).

### **Συναθροιστές Περιεχομένου (Content Aggregators)**

Λειτουργούν σαν ενδιάμεσες οντότητες μεταξύ των content providers και των αυτών που προωθούν την πληροφορία. Συγκεντρώνουν και συνδυάζουν πληροφορία από πολλαπλές πηγές. Εστιάζουν κυρίως στην δημιουργία προστιθέμενης αξίας.

### **Παροχείς Πληρωμών (Payment Providers/ Facilitators)**

Το ρόλο αυτό έχουν οι εταιρείες εκείνες που παρέχουν εξειδικευμένες υπηρεσίες χρέωσης και αναλαμβάνουν την διαχείριση των πληρωμών. Σήμερα αυτό το ρόλο τον έχουν αναλάβει οι εταιρείες δικτύων κινητής τηλεφωνίας. Ο τρόπος χρέωσης είναι κυρίως με μηνιαίες εισφορές ή μέσω προπληρωμένων τηλεφωνικών καρτών. Το κυριότερο πρόβλημα είναι αυτό της συμβατότητας και των τυποποιήσεων.

### **1.3.1.2 Επιχειρηματικά Μοντέλα**

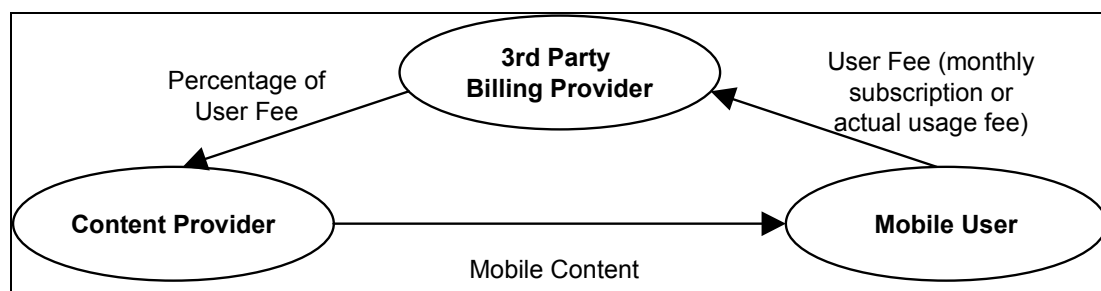
Επειδή η αγορά των LBS είναι ακόμη σε πρώιμο στάδιο υπάρχουν πολλές ιδέες για διαφορετικά επιχειρηματικά μοντέλα. Μερικά από αυτά βέβαια είναι σε πειραματικό στάδιο. Πολλές εταιρείες πειραματίζονται με τέτοιου είδους μελέτες. Ένα από τα μεγαλύτερα ζητήματα που αντιμετωπίζουν είναι τα θέματα χρέωσης. Παρακάτω παρουσιάζονται τα κυριότερα επιχειρηματικά μοντέλα. Ας σημειωθεί εδώ ότι ένα επιχειρηματικό μοντέλο θα πρέπει να έχει άμεση σχέση με τις LBS εφαρμογές που πρόκειται να αναπτυχθούν, δηλαδή δεν θα ήταν φρόνιμο (και δυνατό) να έχουμε ένα που να ταιριάζει με όλες τις εφαρμογές.

#### **Βασικό (Basic/Core Business Model)**

Αυτό το μοντέλο προϋποθέτει ότι οι υπηρεσίες που εμπλέκονται είναι πρόσθετες (additive) στη βασική υπηρεσία (π.χ. υπηρεσία φωνής) σε μία «ελεύθερη» βάση, επειδή αυτό θα συνέβαλε σημαντικά στην προώθηση της εταιρείας με αποτέλεσμα την προσέλκυση νέων συνδρομητών. Παραδείγματα: Direct Connect (Nextel), AOL Instant Messenger (AIM) [35].

#### **«Παραδοσιακό» (“Traditional” ή User Fee Business Model)**

Αυτό το μοντέλο υποθέτει ότι κάθε πελάτης θα πληρώνει μία εισφορά/ τέλος (fee) για την υπηρεσία που του παρέχεται.



**Εικόνα 37: Το παραδοσιακό μοντέλο**

Οι κατηγορίες που προκύπτουν είναι :

- Καθορισμένου Τέλους (Fixed Fee): Ουσιαστικά είναι υπηρεσίες πρόσθετες στις βασικές (Βασικό Μοντέλο) που θα προσφέρονται με βάση μία καθορισμένη εισφορά/τέλος (συνήθως μηνιαία), και οι πελάτες θα πληρώνουν γι' αυτές επιπρόσθετα με τις βασικές υπηρεσίες (φωνή).
- Με βάση τη χρήση (Per Use): Υπηρεσίες που προσφέρονται και θα πληρώνονται ανάλογα με τη χρήση, δηλαδή ανάλογα με τον χρόνο που χρησιμοποιούνται ή σε πολλές περιπτώσεις ανάλογα με τον αριθμό των χρηστών που τις χρησιμοποιούν ταυτόχρονα (π.χ. *conference bridges*) [35,36]

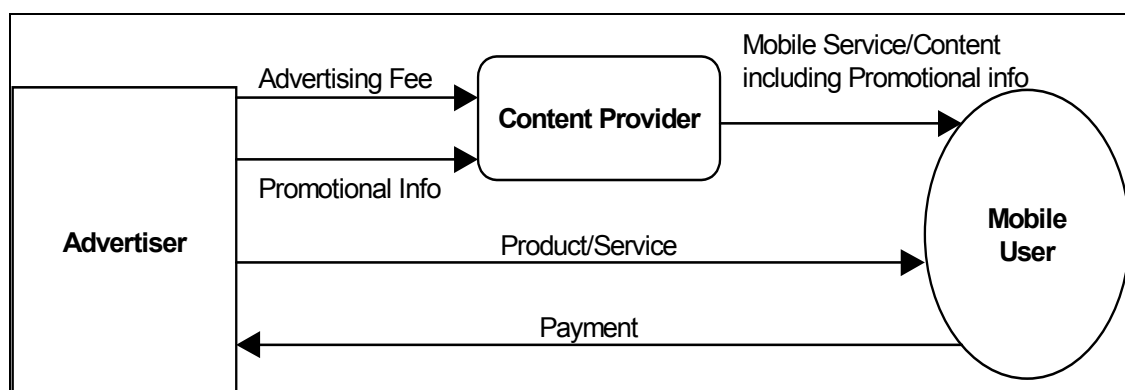
Τα παραπάνω μοντέλα ταιριάζουν περισσότερο στη σημερινή κατάσταση όπου ο δικτυακός φορέας (Network Operator) επιτελεί ουσιαστικά το κυριότερο ρόλο. Η υιοθέτηση όμως αυτών για όλο το φάσμα των υπηρεσιών θέσης μελλοντικά θα ήταν ατυχής. Ίσως θα μπορούσαν να χρησιμοποιηθούν για ένα υποσύνολο αυτών. Όσο υπάρχει διαρκής ανάπτυξη τα μοντέλα που πιθανώς θα επικρατήσουν είναι το Διαφημιστικό μοντέλο (Advertising/ Sponsor-Based Business Model), το Εξαρτώμενο από το Περιεχόμενο και τις Δοσοληψίες μοντέλο (Context-Based/Transaction-Oriented Business Model), το μοντέλο Διαμοιρασμού Εισοδήματος (Revenue Sharing Business Model).

### Advertising Business Model

Η διαφήμιση στον χώρο των ασύρματων επικοινωνιών έχει διαφορετικό χαρακτήρα από ότι π.χ. στο Internet λόγω του ότι τα τερματικά δεν έχουν οθόνες ικανοποιητικού μεγέθους που θα βοηθούσαν στην καλύτερη παρουσίαση. Έτσι ακολουθούνται διαφορετικοί τρόποι παρουσίασης.

Ένας τρόπος είναι να παρουσιάζονται στους χρήστες διαφημίσεις σε μορφή SMS ή WAP που έχουν σχέση με τις επερωτήσεις τους (π.χ. αν αναζητούν ένα μέρος για να γευματίσουν να παρουσιάζονται διαφημίσεις των κοντινότερων εστιατορίων).

Σύμφωνα με το μοντέλο αυτό ο διαφημιστής (advertiser) πληρώνει τον παροχέα περιεχομένου (content provider) για να προσθέσει διαφημιστικά μηνύματα στο περιεχόμενο που διανέμεται στους χρήστες. Η βασική μορφή αυτού του μοντέλου παρουσιάζεται στην παρακάτω εικόνα .



Εικόνα 38: Το διαφημιστικό μοντέλο

Στην πραγματικότητα, στο παραπάνω σχήμα μπορούν να εισέλθουν και άλλοι παίκτες όπως οι service providers, οι mobile network operators και οι application service providers. Οι εισφορές/τέλη (fees) μπορούν να υπολογιστούν με διαφορετικούς τρόπους. Η απλούστερη μορφή είναι ο διαφημιζόμενος να πληρώνει το αντίτιμο για την παρουσίαση της διαφήμισης του για μια συγκεκριμένη χρονική περίοδο. Μία άλλη προσέγγιση είναι οι εισφορές βασιζόμενες στην κίνηση(traffic based) κατά την οποία η πληρωμή γίνεται με βάση την συχνότητα (αριθμό) των διαφημιστικών μηνυμάτων.

Οι διαφημίσεις μπορούν να φτάσουν στον τελικό χρήστη με δύο τρόπους push και pull. Κατά τον πρώτο τρόπο ο χρήστης θα δέχεται μηνύματα στο κινητό του αυτόματα κυρίως με την μορφή SMS. Στον δεύτερο τρόπο ο χρήστης θα δέχεται τα διαφημιστικά μηνύματα σε συνδυασμό με το περιεχόμενο της επερώτησής του.

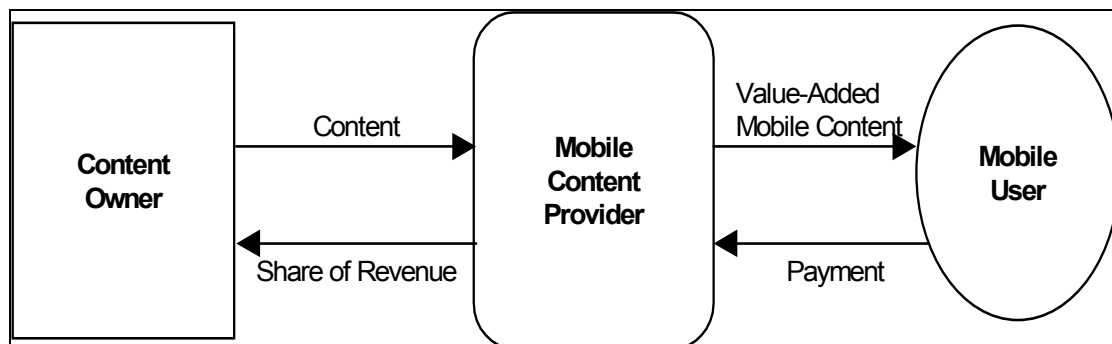
Ο κύριος λόγος που αυτό το μοντέλο είναι πιο αποδοτικό στην ασύρματη επικοινωνία από ότι στην ενσύρματη, είναι ότι στηρίζεται σε διαφημίσεις που έχουν σχέση με την θέση που βρίσκεται ο χρήστης και έτσι είναι πιο πιθανό ότι θα τον ενδιαφέρουν. Αυτό το πλεονέκτημά τους μπορεί να ενισχυθεί και με την χρήση προφίλ χρηστών που μπορούν να δημιουργηθούν [34].

### Context-Based/Transaction-Oriented Business Model

Είναι ένα ακόμη πιθανό μοντέλο που οι εισφορές θα έχουν σχέση αποκλειστικά με το περιεχόμενο που διακινείται προς τον τελικό χρήστη αλλά και τις δοσοληψίες που αυτός εκτελεί. Έχει πολύ στενή σχέση με το προηγούμενο μοντέλο και θα μπορούσε να αποτελέσει υποπερίπτωση του. Ο διαχωρισμός τους δεν είναι και τόσο ξεκάθαρος [35].

### Revenue-Sharing Business Model

Το κύριο χαρακτηριστικό αυτού του μοντέλου είναι η συλλογή των εισφορών από τον χρήστη και ο διαμοιρασμός τους σε όλα τα άλλα μέρη που βρίσκονται στην αλυσίδα αξίας (value chain) π.χ. content providers, mobile network operators κ.α.



Εικόνα 39: Το μοντέλο διαμοιρασμού εισοδήματος

Η «πώληση» περιεχομένου κατευθείαν στον τελικό χρήστη/πελάτη από μία και μόνο εταιρεία δεν είναι πάντα εύκολο γεγονός. Η υπηρεσία δεν θα είναι και τόσο ανταγωνιστική. Αυτό δίνει την δυνατότητα ύπαρξης και άλλων εταιρειών οι οποίες σε συνδυασμό θα προσφέρουν μία υπηρεσία που θα προσέλκυε αρκετού πελάτες [34].



Το πλεονέκτημα για τον τελικό χρήστη είναι ότι με την αύξηση του ανταγωνισμού μεταξύ όλων αυτών των εταιρειών δεν θα υπάρχει μόνο μείωση του κόστους αλλά και βελτίωση της ποιότητας της προσφερόμενης υπηρεσίας κάτι που μόνος του ο φορέας του δικτύου δεν θα κατάφερνε. Άρα λοιπόν υπάρχει ανάγκη από πολλούς «παίκτες».

### 1.3.2 Χρέωση

Με τον ερχομό των νέων υπηρεσιών προστιθέμενης αξίας προκλήθηκαν πολλές αλλαγές στο «οικοσύστημα» των κινητών επικοινωνιών. Καταρχήν άλλαξε το είδος των παρεχόμενων υπηρεσιών: στις υπηρεσίες φωνής προστέθηκαν και οι υπηρεσίες δεδομένων. Οι τεχνολογίες που χρησιμοποιούνται στην ασύρματη βιομηχανία, επίσης, άλλαξαν και αλλάζουν. Έτσι γίνεται σταδιακά η μετάβαση από τα δίκτυα μεταγωγής κυκλώματος (2G) στα δίκτυα μεταγωγής πακέτων (2.5G και 3G) που βασίζονται στο IP πρωτόκολλο. Ακόμη, αυτές οι νέες υπηρεσίες απαιτούν την συνεργασία πολλών εταιρών και την υιοθέτηση νέων επιχειρησιακών μοντέλων που θα ορίζουν την μεταξύ τους συνεργασία.

Όλες αυτές οι νέες συνθήκες επιβάλουν και την ανάπτυξη νέων συστημάτων χρέωσης (ο όρος «σύστημα» χρησιμοποιείται και με την τεχνολογική και με την επιχειρηματική έννοια). Για παράδειγμα πρέπει να γίνει το πέρασμα από την χρέωση που βασίζεται στον χρόνο (χρονοχρέωση), σε νέα μοντέλα χρέωσης που βασίζονται κυρίως στην ποσότητα του περιεχομένου (ογκοχρέωση) και την αξία του. Πριν όμως από κάθε ενασχόληση με θέματα χρέωσης καλό είναι να έχει ξεκαθαριστεί η σημασία των εξής εννοιών: *χρέωση, τιμολόγηση, καθορισμός τιμολογιακής πολιτικής και λογιστική τακτοποίηση.*

- Χρέωση (charging): Κατά την διαδικασία της χρέωσης συλλέγεται η απαραίτητη πληροφορία από τις αντίστοιχες οντότητες του συστήματος παροχής υπηρεσιών (δίκτυο, application servers κλπ).
- Τιμολόγηση (billing): Κατά την διαδικασία της τιμολόγησης οι πληροφορίες που έχουν συλλεχθεί από τη χρέωση, επεξεργάζονται με βάση τα αντίστοιχα μοντέλα χρέωσης, έτσι ώστε να υπολογιστεί η χρέωση για μια συγκεκριμένη υπηρεσία.
- Λογιστική Τακτοποίηση (accounting): Σε περιπτώσεις περιαγωγής (roaming) θα πρέπει να καταμεριστούν τα έσοδα στους δύο παροχείς. Η διαδικασία αυτή ονομάζεται «Λογιστική Τακτοποίηση». Επιπλέον στα συστήματα που παρέχουν υπηρεσίες προστιθέμενης αξίας (value added services) από ανεξάρτητους παροχείς υπηρεσιών, το “accounting” θα είναι υπεύθυνο για να μοιραστούν τα έσοδα ανάμεσα στον παροχέα του δικτύου και στους παροχείς των υπηρεσιών.
- Καθορισμός Τιμολογιακής Πολιτικής (pricing): Κατά την διαδικασία αυτή καθορίζονται οι παράμετροι που θα λαμβάνονται υπόψη τόσο κατά την διαδικασία του “billing” όσο και κατά την διαδικασία του “accounting”. Επίσης

καθορίζεται και το μοντέλο χρέωσης που θα ακολουθηθεί (πχ επίπεδη χρέωση, χρονοχρέωση, ογκοχρέωση κλπ).  
Η παρούσα ενότητα ασχολείται με τις παραπάνω διαδικασίες τόσο από οικονομική όσο και από τεχνική σκοπιά. Τα θέματα που θίγονται, όπως είναι φυσικό, έχουν πολλά κοινά σημεία με τις υπόλοιπες υπηρεσίες προστιθέμενης αξίας, εκτός από τις υπηρεσίες θέσης.

### **1.3.2.1 Οικονομικά θέματα**

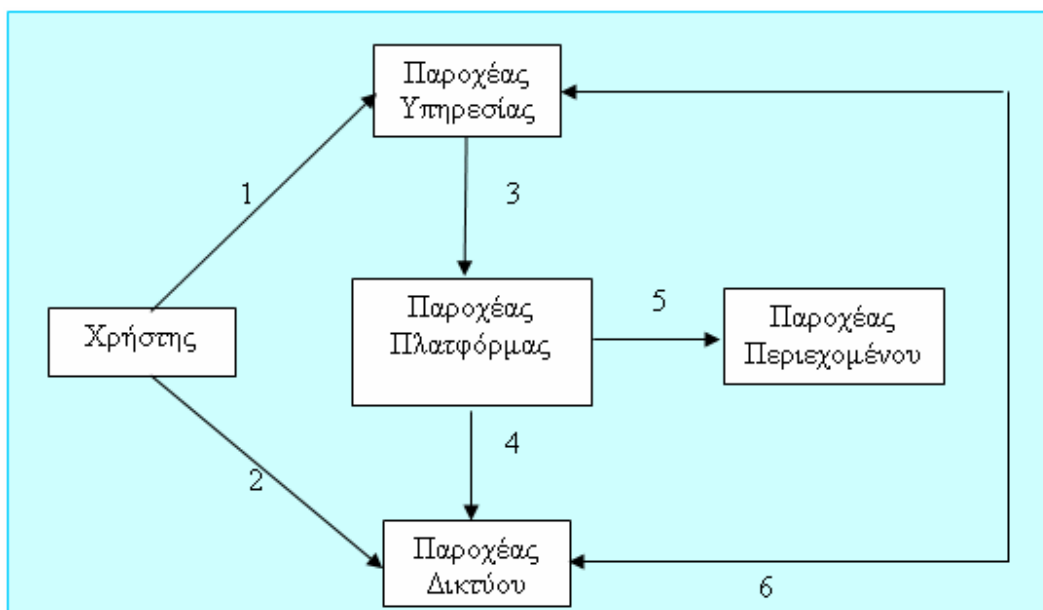
#### **1.3.2.1.1 Οι νέες απαιτήσεις**

Οι νέες υπηρεσίες έχουν και νέες απαιτήσεις από τα συστήματα χρέωσης. Μερικές από αυτές είναι:

- Γρήγορη και εύκολη προσθήκη νέων υπηρεσιών  
Οι νέες υπηρεσίες πρέπει να μπορούν να ενσωματωθούν εύκολα στην υπάρχουσα χρέωση ενός χρήστη και να είναι διαθέσιμες το γρηγορότερο δυνατόν.
- Όχι επιπλέον συμβόλαια για την χρήση νέων υπηρεσιών  
Οι χρήστες δυσανασχετούν με την ύπαρξη πολλών συμβολαίων και γενικά με τις πολύπλοκες διαδικασίες και την γραφειοκρατία.
- Έγκυρη και έγκαιρη ενημέρωση για το κόστος χρήσης μιας υπηρεσίας  
Πρέπει να παρέχεται πλήρης και έγκυρη ενημέρωση σχετικά με τα ζητήματα χρέωσης. Αυτή μπορεί να γίνει πριν, κατά τη διάρκεια ή μετά τη χρήση της υπηρεσίας
- Κλιμακούμενη χρέωση ανάλογα με:
  1. την ποιότητα της παρεχόμενης υπηρεσίας  
Για τις υπηρεσίες θέσης η ποιότητα αυτή μεταφράζεται σε ακρίβεια εντοπισμού θέσης και σε υψηλή αξία περιεχομένου.
  2. την δέσμευση και χρήση των πόρων του δικτύου  
Πολύ απαιτητικές υπηρεσίες (σε όγκο δεδομένων που μεταφέρουν ή υπολογιστική ισχύ που καταναλώνουν) πρέπει να χρεώνονται με άλλα κριτήρια
  3. το προφίλ του χρήστη  
Η χρέωση πρέπει να λαμβάνει υπόψη και το προφίλ του χρήστη, ώστε να γίνεται με βάση και τα προσωπικά του χαρακτηριστικά. Αυτό μπορεί να οδηγήσει σε διαφορετικές χρεώσεις για την ίδια ακριβώς υπηρεσία.
  4. την ίδια την υπηρεσία
  5. τα περιεχόμενα/πληροφορίες που πήρε ο χρήστης απ' αυτήν  
Πρέπει να λαμβάνεται υπόψη τόσο η ποσότητα όσο και η ποσότητα των παρεχόμενων ή χρησιμοποιούμενων υπηρεσιών
  6. την θέση του χρήστη  
Έχουν αρχίσει ήδη να εφαρμόζονται σποραδικά, διαφορετικές χρεώσεις ανάλογα με την θέση του κινητού χρήστη.

### 1.3.2.1.2 Ροές Χρέωσης

Ανάμεσα στους εταίρους ενός συστήματος που παρέχει υπηρεσίες περιεχομένου, υπάρχουν κάποιες ροές χρέωσης (revenue ή charging flows), οι οποίες δεν υπήρχαν στα συστήματα φωνής και παραδοσιακών υπηρεσιών (SMS). Στην παρακάτω εικόνα διακρίνονται αυτές οι ροές.



Εικόνα 40: Οι ροές χρέωσης ανάμεσα στους «παίκτες» των υπηρεσιών θέσης

Οι αριθμημένες ροές χρέωσης είναι:

- 1) Ο παροχέας της υπηρεσίας μπορεί να χρεώνει άμεσα (ή έμμεσα μέσω των ροών 6 και 2) τον τελικό χρήστη για την χρήση της υπηρεσίας, ειδικά στην περίπτωση που γίνονται δΟΣΟΛΗΨΙΕΣ στο πλαίσιο της (πχ αγορά εισιτηρίων)
- 2) Ο παροχέας του ασύρματου δικτύου χρεώνει τον συνδρομητή του για την χρήση του ασύρματου δικτύου σαν μέσο μεταφοράς της πληροφορίας. Αυτή η χρέωση όπως θα δούμε παρακάτω μπορεί να γίνει με διάφορους τρόπους.
- 3) Ο παροχέας της πλατφόρμας στην οποία εκτελούνται οι υπηρεσίες μπορεί να χρεώσει τον παροχέα της υπηρεσίας για διάφορες λειτουργίες:
  - i) για την δημιουργία της υπηρεσίας (αν αυτή λαμβάνει χώρα στην πλατφόρμα).
  - ii) για την εκτέλεση της υπηρεσίας. Η χρέωση αυτή μπορεί να είναι ανάλογη και της επιθυμητής επίδοσης της υπηρεσίας. Για παράδειγμα ταυτόχρονη υποστήριξη πολλών χρηστών με μικρή καθυστέρηση, μπορεί να είναι κριτήριο για μεγαλύτερη χρέωση.
  - iii) για την προσπέλαση περιεχομένου (δεδομένα γεωγραφικής θέσης, δεδομένα από GIS) που απαιτεί η υπηρεσία. Οι τιμές μπορεί να εξαρτώνται από τον

αριθμό και την ποιότητα των αιτήσεων προς τον παροχέα του δικτύου και προς τους παροχείς περιεχομένου.

- 4) Ο παροχέας του δικτύου μπορεί να χρεώσει τον παροχέα της πλατφόρμας για την χρήση της πληροφορίας θέσης του χρήστη. Αυτή η χρέωση μπορεί να είναι είτε επίπεδη (flat fee) είτε να εξαρτάται από τον αριθμό των αιτήσεων που κάνει ο παροχέας της πλατφόρμας για πληροφορία θέσης. Βέβαια κρίνεται σκόπιμο να υπάρχουν στο μεταξύ τους συμβόλαιο και εγγυήσεις για την ποιότητα της εξυπηρέτησης (ελάχιστος και μέγιστος αριθμός αιτήσεων ανά δευτερόλεπτο, καθυστέρηση, ακρίβεια εντοπισμού κτ).
- 5) Ο παροχέας της πλατφόρμας χρεώνεται από τον παροχέα περιεχομένου για τα δεδομένα που αντλεί από αυτόν.
- 6) Ο παροχέας δικτύου και ο παροχέας της υπηρεσίας μπορούν να κάνουν μια συμφωνία για διαμοιρασμό των εσόδων από την χρήση της υπηρεσίας.

Από τα παραπάνω προκύπτει ότι ο χρήστης θα πρέπει να χρεώνεται μόνο από τον παροχέα του δικτύου ή έστω και από τον παροχέα της υπηρεσίας. Οι υπόλοιποι «παίκτες» πρέπει να είναι «διαφανείς» σε αυτόν, αφού εξάλλου δεν έχουν άμεση οικονομική συσχέτιση μαζί του. Το μοντέλο χρέωσης που θα υιοθετηθεί από τους παροχείς για τις μεταξύ τους χρεώσεις πρέπει να ακολουθεί κάποιες αρχές:

- πρέπει να μπορεί να διασφαλιστεί μερίδιο των εσόδων για όλους τους εμπλεκόμενους
- η χρέωση πρέπει να γίνεται σε πραγματικό χρόνο και
- να επιτρέπεται η ακύρωση (rollback) των πληρωμών όταν οι υπηρεσίες δεν μπορούν να φτάσουν για κάποιο λόγο στον τελικό χρήστη.

Στην διαμόρφωση της τελικής χρέωσης μεταξύ των επιχειρηματικών εταίρων μπορούν να συμπεριληφθούν και άλλα κριτήρια. Έτσι μπορεί να συνδυαστεί με τα παραπάνω και κάποιο μοντέλο χρέωσης:

- ανά εγγεγραμμένο χρήστη
- ανά ώρα συνεδρίας (session)
- ανά όγκο δεδομένων (πχ ανά Megabyte)
- ανά προτεραιότητα της υπηρεσίας
- ανά πλήθος και είδος δοσοληψιών (transactions)

### **1.3.2.1.3 Είδη χρεώσεων**

Σε αυτό το σημείο πρέπει να διαχωρίσουμε τα είδη των χρεώσεων που συναντιούνται σε ένα τέτοιο σύστημα. Ο διαχωρισμός γίνεται με βάση την οντότητα που χρεώνεται κάθε φορά. Έτσι έχουμε τις χρεώσεις μεταφοράς (transport charges), τις χρεώσεις πληροφορίας θέσης (location information charges) και τις χρεώσεις των ιδίων των υπηρεσιών (service charges).

#### **Χρεώσεις μεταφοράς**

Είναι οι χρεώσεις που αποφασίζονται αποκλειστικά από τον παροχέα του δικτύου κινητών επικοινωνιών. Οι παράμετροι που τις καθορίζουν είναι η ποσότητα των

μεταφερόμενων δεδομένων, η ποιότητα εξυπηρέτησης (QoS), η διάρκεια των μεταφορών, η ώρα της ημέρας κατά την οποία γίνονται οι μεταφορές, η κινητικότητα του χρήστη και η θέση του. Επίσης η χρέωση πρέπει να ρυθμίζεται ευέλικτα ανάλογα με τον φορέα που χρησιμοποιείται (IP, GPRS, SMS κα). Ο τελικός υπολογισμός της χρέωσης μεταφοράς γίνεται με βάση τις πολιτικές τιμολόγησης (pricing), όπως αυτές καθορίζονται από το συμβόλαιο του συγκεκριμένου χρήστη με τον τηλεπικοινωνιακό παροχέα.

#### **Χρεώσεις πληροφορίας θέσης**

Αυτές είναι οι χρεώσεις τις οποίες υφίστανται όσοι έχουν πρόσβαση στις πληροφορίες θέσης, που μπορεί να παράγει ο παροχέας του δικτύου με την τεχνολογία εντοπισμού θέσης που διαθέτει. Καταρχήν εξαρτώνται από την ποιότητα της αιτούμενης εξυπηρέτησης (QoS), που μπορεί να είναι η ακρίβεια σε μέτρα και η επιθυμητή καθυστέρηση της απάντησης. Επίσης σημασία έχει και το αν έχει αιτηθεί μια απλή θέση χρήστη ή συνεχής παρακολούθηση (tracking). Στην δεύτερη περίπτωση συνυπολογίζονται η συνολική διάρκεια της παρακολούθησης και η συχνότητα με την οποία ανιχνεύεται η θέση του χρήστη.

#### **Χρεώσεις υπηρεσίας**

Αυτές φυσικά είναι διαφορετικές για κάθε υπηρεσία. Η χρέωση πρέπει να είναι δυνατή ανεξάρτητα από τον τρόπο με τον οποίο ο χρήστης έχει πρόσβαση στην υπηρεσία (WAP, SMS, WWW, φωνητικές κλήσεις). Ο τρόπος όμως μπορεί να αποτελεί κριτήριο διαφορετικής χρέωσης. Επιπρόσθετα, η χρέωση μπορεί να εξαρτάται από τα πρωτόκολλα που χρησιμοποιεί η υπηρεσία (πχ TCP/UDP, HTTP/FTP). Το τελικό κόστος της υπηρεσίας θα καθορίσουν και άλλοι παράγοντες, όπως το περιεχόμενο που χρησιμοποιείται, το προφίλ του χρήστη και η συχνότητα παροχής της.

### **1.3.2.1.4 Χρέωση του τελικού χρήστη**

Σήμερα οι πιο συνηθισμένες μέθοδοι χρέωσης του τελικού χρήστη για υπηρεσίες προστιθέμενης αξίας είναι η επίπεδη χρέωση (flat-rate) και η ογκοχρέωση. Η πρώτη, παρόλο που είναι αρκετά αποδεκτή από τους χρήστες, φαίνεται ότι θα εγκαταλειφθεί από τους παροχείς των δικτύων αφού δεν αντικατοπτρίζει την πραγματική αξία των παρεχόμενων υπηρεσιών. Η υιοθέτηση μιας τέτοιας χρέωσης δεν μπορεί πάντα να καλύψει τα λειτουργικά έξοδα του δικτύου. Εξάλλου, η επίπεδη χρέωση δεν μπορεί να εφαρμοστεί με μεγάλη επιτυχία σε δίκτυα κινητών επικοινωνιών για δύο λόγους. Πρώτον οι διαθέσιμοι πόροι που είναι προσβάσιμοι από τον χρήστη (δεδομένα, υπηρεσίες) είναι περιορισμένοι ακόμα, και δεύτερον η απρόβλεπτη κινητικότητα σε συνδυασμό με την υπερβολική χρήση του δικτύου για υπηρεσίες δεδομένων (η οποία δεν περιορίζεται από την επίπεδη χρέωση), μπορούν να αυξήσουν την χρήση του δικτύου σε τέτοιο βαθμό που να υποβαθμιστεί η ποιότητα υπηρεσίας του (QoS) και να αυξηθεί ο βαθμός εξυπηρέτησης (GoS).

Η ογκοχρέωση, που εφαρμόζεται σήμερα στα δίκτυα 2.5G, δεν είναι ούτε αυτή μια αποδεκτή μέθοδος χρέωσης για υπηρεσίες που βασίζονται στο περιεχόμενο. Δεν

πρέπει να συνδέεται η ποσότητα της πληροφορίας με την ποιότητά και την αξία της. Αυτή η ισοπέδωση που προκαλεί μια τέτοια μέθοδος χρέωσης δεν θα επιτρέψει την δημιουργία πραγματικά ποιοτικών υπηρεσιών. Οι παροχείς της εικόνας 40 θέλουν αποκλειστικά να αυξήσουν το κέρδος τους και αν το μέτρο χρέωσης είναι ο όγκος, τότε θα στραφούν σε «ογκώδεις» και όχι χρήσιμες και εξελιγμένες υπηρεσίες.

Παρόλα τα πλεονεκτήματα που προσφέρουν οι διάφορες μέθοδοι χρέωσης καμία δεν φαίνεται ότι μπορεί να εφαρμοστεί σε αποκλειστικό επίπεδο. Αντιθέτως πρέπει κάθε φορέας να δημιουργήσει το κατάλληλο συνδυασμό που θα του επιφέρει το επιθυμητό κέρδος, ενθαρρύνοντας ταυτόχρονα και την εξάπλωση της χρήσης τέτοιων υπηρεσιών. Για παράδειγμα μπορεί κάποιος παροχέας δικτύου να εφαρμόσει επίπεδη χρέωση για να καλύψει τα λειτουργικά έξοδα του δικτύου, ενώ χρέωση βασιζόμενη στο περιεχόμενο για να προσδώσει επιπλέον αξία στις υπηρεσίες. Ένα μάλλον επιτυχημένο μοντέλο χρέωσης (κρίνοντας εκ του αποτελέσματος) έχει εφαρμόσει η ιαπωνική εταιρία NTT DoCoMo για το σύστημα i-mode. Ο χρήστες πληρώνουν μηνιαία κάποιο πάγιο τέλος για την χρήση του δικτύου (χρέωση μεταφοράς πακέτων), ένα πάγιο τέλος για την υπηρεσία i-mode, ένα πάγιο τέλος που καθορίζεται από τους παροχείς δεδομένων (για τα i-menu) και επιπλέον χρεώνονται ανάλογα με το είδος του περιεχομένου που έχουν διακινήσει. Μερικές από τις κατηγορίες περιεχομένου είναι: μήνυμα ηλεκτρονικού ταχυδρομείου, εικόνες, ενημέρωση για τιμές μετοχών και μεταφορά χρημάτων.

Μελετώντας σε ένα άλλο επίπεδο την χρέωση του τελικού χρήστη, παρουσιάζονται δύο πιθανά μοντέλα: η προπληρωμή (prepaid) και η μεταπληρωμή (postpaid). Η προπληρωμή υλοποιείται με την χρήση των γνωστών καρτών της κινητής τηλεφωνίας. Οι χρήστες που χρεώνονται προκαταβολικά αποτελούν πλέον μεγάλο μερίδιο των συνολικών χρηστών (στην Ευρώπη είναι ήδη η πλειοψηφία) αλλά προσφέρουν στις αντίστοιχες εταιρίες μικρότερα έσοδα από τους χρήστες που χρεώνονται με μεταπληρωμή (δηλαδή με συμβόλαια). Η προπληρωμή έχει κάποια πλεονεκτήματα όπως το ότι δεν απαιτεί πολύπλοκους αλγόριθμους χρέωσης και ότι προσφέρει ανωνυμία στους χρήστες. Από την άλλη πλευρά όμως είναι πιο ακριβή και η ανωνυμία που προσφέρει δεν συμφωνεί με την φύση πολλών από τις νέες υπηρεσίες, που προϋποθέτουν χρήση του προφίλ του χρήστη. Επίσης απαιτεί ενημέρωση των χρηστών και του παροχέα σε πραγματικό χρόνο (real time), σχετικά με την πιστωτική κατάσταση του χρήστη. Σε κάθε περίπτωση ο χρήστης πρέπει να έχει καλή και έγκαιρη ενημέρωση από τον παροχέα του σχετικά με την τιμολόγηση αλλά και την τρέχουσα κατάστασή του.

### **1.3.2.2 Τεχνικά θέματα**

Η ανάπτυξη συστημάτων που να υποστηρίζουν τις παραπάνω αρκετά περίπλοκες μορφές χρέωσης σίγουρα δεν είναι εύκολη υπόθεση. Υπάρχουν αρκετά στοιχεία που διαφοροποιούν αυτά τα συστήματα από τα προηγούμενα μονολιθικά που χρησιμοποιούνταν στα ασύρματα δίκτυα δεύτερης γενιάς. Είναι προφανές ότι τα νέα συστήματα πρέπει να είναι πολύ ευέλικτα και να μπορούν να προσαρμοστούν εύκολα ανάλογα με τις επιχειρηματικές ανάγκες που προκύπτουν. Τα συστήματα αυτά συνήθως δεν μπορούν να υλοποιηθούν σαν επεκτάσεις των ήδη υπάρχοντων συστημάτων. Ούτε είναι αποτελεσματικό οι εταιρίες λογισμικού να αναπτύσσουν εμπορικά συστήματα

χρέωσης γενικού σκοπού. Με αυτόν τον τρόπο δεν θα μπορεί να υποστηριχτεί εύκολα η περιαγωγή και η συνεργασία των παροχών, αλλά και δεν θα μπορεί να επιτευχθεί η επιθυμητή ευελιξία που απαιτείται. Αντίθετα πρέπει να αρχίσουν να δημιουργούνται πρότυπα που θα καθορίζουν την δομή και τις διεπαφές αυτών των συστημάτων και να χρησιμοποιούνται τα ήδη υπάρχοντα (πχ OSA/Parlay [37]).

Τα συστήματα αυτά απαιτούν συλλογή από πολλά στοιχεία του δικτύου. Κεντρικής σημασίας είναι οι συσκευές μέτρησης (measurement devices). Αυτές είναι συσκευές που μετρούν παραμέτρους όπως η διάρκεια, ο όγκος των δεδομένων που μεταφέρονται, ο τύπος και η αξία της υπηρεσίας και η ποιότητα υπηρεσίας. Η μέτρηση και συλλογή των παραπάνω στοιχείων πρέπει να γίνεται κατανοημένα και όχι μόνο στον παροχέα του δικτύου. Εξάλλου αυτός, συνήθως, δεν θα είναι σε θέση να πάρει όλες τις μετρήσεις που απαιτούνται. Στο τέλος πρέπει να συνυπολογίζονται όλα τα στοιχεία σε κάποιο κεντρικό σημείο και να γίνεται η πλήρης διαδικασία της χρέωσης (charging, billing, accounting). Ο υπολογισμός της χρέωσης κάθε υπηρεσίας πρέπει να γίνεται σε πραγματικό χρόνο. Αυτό απαιτεί φυσικά καλό σχεδιασμό και επαρκή εξοπλισμό γιατί εμπλέκονται πολλές μεταφορές δεδομένων και υπολογισμοί σε αυτήν την διαδικασία.

Αναλύοντας με πιο μεγάλη λεπτομέρεια ένα τέτοιο σύστημα βλέπουμε ότι ο καλύτερος τρόπος με τον οποίο μπορεί να υλοποιηθεί είναι με διατήρηση αρχείων ημερολόγιου (logging files). Αυτά τα αρχεία είναι θεμιτό να διατηρούνται σε όλους τους εμπλεκόμενους παροχείς. Άλλα από αυτά είναι πιο χρήσιμα για τον υπολογισμό της χρέωσης (αυτά του παροχέα της πλατφόρμας και του παροχέα του δικτύου) και άλλα μπορούν να χρησιμοποιηθούν για επαλήθευση και έλεγχο της συνολικής διαδικασίας. Για παράδειγμα ο παροχέας υπηρεσίας μπορεί να χρεώνεται με βάση τα logs του παροχέα της πλατφόρμας για την χρήση της αλλά μπορεί να κρατάει και δικά του logs ή να χρησιμοποιεί αυτά του παροχέα δικτύου για επαλήθευση. Με ορθές διαδικασίες ελέγχου θα δημιουργηθεί ένα κλίμα εμπιστοσύνης και σωστής συνεργασίας μεταξύ των πολλών εμπλεκόμενων παροχών.

Κλείνοντας με το θέμα της χρέωσης, να επισημάνουμε ξανά ότι είναι ίσως το πιο ουσιαστικό σημείο για την γρήγορη διεύθυνση των υπηρεσιών που εξαρτώνται από την θέση. Οι κατάλληλες επιλογές μετά από προσεκτική μελέτη όλων των σχετιζόμενων παραμέτρων θα οδηγήσουν σε βιώσιμα επιχειρηματικά μοντέλα. Και φυσικά πάντα οδηγός πρέπει να είναι η υποστήριξη της ευελιξίας. Αυτή επιτάσσεται από το γεγονός ότι ακόμα δεν είναι γνωστή η κατεύθυνση που θα ακολουθήσουν αυτές οι υπηρεσίες.

### 1.3.3 Ιδιωτικότητα

Σίγουρα ένα από τα σοβαρότερα προβλήματα που πρέπει να λύσει η βιομηχανία των ασύρματων υπηρεσιών θέσης είναι αυτό της ιδιωτικότητας (privacy). Φυσικά κάθε χρήστης, από την στιγμή που κάνει χρήση μιας κινητής συσκευής γνωρίζει και αποδέχεται ότι ο παροχέας του δικτύου θα ξέρει την θέση του, τουλάχιστον σαν απόσταση από τον σταθμό βάσης που τον εξυπηρετεί. Αυτό όμως απέχει πολύ από την χρήση, και πολύ περισσότερο την διάθεση σε τρίτους, της «ακριβούς» θέσης του, που εντοπίζεται πλέον με πιο εξελιγμένες μεθόδους στην περίπτωση των υπηρεσιών θέσης.

Και αν ακόμη δεχτούμε ότι δεν τίθεται θέμα παραβίασης της ιδιωτικότητας σε υπηρεσίες έκτακτης ανάγκης όπως το E911 (μια υπόθεση που δεν είναι κοινώς αποδεκτή), σίγουρα ανακύπτουν κάποια θέματα για τις υπόλοιπες εμπορικές υπηρεσίες και εφαρμογές, τα οποία αναλύονται παρακάτω.

### **1.3.3.1 Θέματα και Προβλήματα που ανακύπτουν**

Καταρχήν δημιουργούνται κάποια εύλογα ερωτήματα όσον αφορά την χρήση των πληροφοριών που σχετίζονται με την θέση του χρήστη. Μερικά από αυτά είναι:

- Ποιος θα έχει πρόσβαση σε αυτές τις πληροφορίες;
- Για πόσο θα υπάρχουν στην διάθεση όσων τις χρησιμοποιούν;
- Που θα βρίσκονται αποθηκευμένες;
- Τι γίνεται αν κάποιος δεν θέλει να εντοπίζεται συνεχώς από τον τηλεπικοινωνιακό του φορέα;

Οι παραπάνω ερωτήσεις είναι απόλυτα δικαιολογημένες αν τις εξετάσουμε υπό το πρίσμα της «αοριστίας και ασάφειας» που επικρατεί σήμερα στον συγκεκριμένο χώρο. Αυτό σημαίνει ότι δεν έχουν καθοριστεί και οριστικοποιηθεί ακόμη τα επιχειρηματικά μοντέλα που θα χρησιμοποιηθούν, άρα δεν είναι γνωστό ποιοι «παίκτες» θα συμμετάσχουν στην παροχή τέτοιων υπηρεσιών. Επίσης δεν έχουν αποφασισθεί τα νομικά πλαίσια μέσα στα οποία θα επιτρέπεται να χρησιμοποιηθούν οι «ευαίσθητες» πληροφορίες των χρηστών. Το κυριότερο όμως είναι ότι καθώς αποτελούν μια εντελώς νέα γενιά τεχνολογιών και υπηρεσιών, δεν ξέρει κανείς, ούτε μπορεί να προβλέψει, όλες τις πιθανές παραβιάσεις της ιδιωτικότητας.

Επιπρόσθετα, παρουσιάζονται και κάποιοι προβληματισμοί στην χρήση των υπηρεσιών θέσης μέσα στο περιβάλλον μιας επιχείρησης. Πιο συγκεκριμένα ορισμένοι θεωρούν ότι το να τους παρακολουθούν συνεχώς οι προϊστάμενοί τους, συνήθως μέσω ενός εταιρικού δικτύου (πχ τύπου WLAN) που υποστηρίζει υπηρεσίες θέσης και υπηρεσιών όπως διαχείριση στόλου (fleet management), είναι σαφής παραβίαση της ιδιωτικότητας τους. Ούτε αυτός είναι κάποιος παράλογος προβληματισμός, αν αναλογιστούμε την κατάσταση που επικρατεί σήμερα σε πολλές εταιρίες (αυστηρή επιτήρηση, δημιουργία φόβου υπό την απειλή απολύσεως και άλλες παραβιάσεις των εργατικών δικαιωμάτων).

Γενικά, η ιδιωτικότητα απειλείται από αυτές τις υπηρεσίες κυρίως με δύο τρόπους: α) *ικανότητα για προηγμένη επιτήρηση και παρακολούθηση*, και β) *δημιουργία και επεξεργασία βάσεων δεδομένων που περιέχουν λεπτομερείς προσωπικές πληροφορίες*. Η πληροφορία της γεωγραφικής θέσης ενός ατόμου είναι πολύ σπουδαία για όποιον παρακολουθεί αυτό το άτομο επειδή είναι ήδη σε ψηφιακή μορφή (είναι ευκολότερη η περαιτέρω επεξεργασία της), και επειδή αυτός που παρακολουθεί δεν μπορεί να γίνει αντιληπτός από τον παρακολουθούμενο. Κανένας άλλος τρόπος παρακολούθησης δεν συνδυάζει αυτά τα πλεονεκτήματα. Η παρακολούθηση γίνεται ακόμη πιο αποδοτική, και άρα πιο ανεπιθύμητη για τον παρακολουθούμενο, αν το ασύρματο δίκτυο παρακολουθεί σε συνεχή βάση τους χρήστες του, και αν οι τεχνολογίες εντοπισμού θέσης γίνουν πιο ακριβείς από τις σημερινές (κάτι που λογικά δεν θα αργήσει πολύ να γίνει). Έχουν παρουσιαστεί ήδη πολλές περιπτώσεις παράνομης παρακολούθησης (συνήθως με την



χρήση GPS) και τίποτα δεν εγγυάται ότι οι νέες υπηρεσίες δεν θα χρησιμοποιηθούν και για τέτοιους σκοπούς. Μερικά από τα άτομα που έχουν χρησιμοποιήσει (σύμφωνα με τον Τύπο) τέτοιες μεθόδους είναι κυνηγοί επικηρυγμένων, ιδιωτικοί αστυνομικοί, άτομα με προσωπικές διαφορές (πχ απατημένοι σύζυγοι) κα.

Όμως η δεύτερη απειλή είναι μάλλον η πιο σημαντική. Αυτή συμπεριλαμβάνει την αποθήκευση, επεξεργασία ή και διανομή σε τρίτους, της πληροφορίας θέσης του χρήστη. Αν η αποθήκευση των δεδομένων αυτών γίνεται για μεγάλο χρονικό διάστημα, μπορεί με κατάλληλη επεξεργασία τους να δημιουργηθεί ένα προφίλ για τον χρήστη. Αυτό το προφίλ μπορεί να είναι πολιτικό, εμπορικό ή οτιδήποτε άλλο ανάλογα με τους σκοπούς αυτού που το δημιουργεί. Στην απλούστερη και πιο «αθώα» περίπτωση θα χρησιμοποιηθεί για σκοπούς μάρκετινγκ και διαφήμισης. Αν τα δεδομένα αυτά διασταυρωθούν και με άλλα προσωπικά στοιχεία, όπως πληροφορίες πιστωτικών καρτών, μπορούν να προκύψουν πιο λεπτομερείς πληροφορίες για τις συνήθειες του χρήστη απ' ό,τι είναι σήμερα δυνατό. Σύμφωνα με πιο κακόβουλα σενάρια, αυτά τα προφίλ μπορούν να διανεμηθούν και να χρησιμοποιηθούν από τρίτους, τους οποίους ο χρήστης ίσως να μην μπορεί να προσεγγίσει.

Οι πληροφορίες που συνάγονται με τις παραπάνω μεθόδους χρησιμοποιούνται συχνά για διαφημιστικούς σκοπούς. Έτσι έχουμε φτάσει σήμερα στο σημείο, να έχει νόημα να ορίσουμε τη φράση “wireless spam” (και “sms spam”, αν αναφερόμαστε σε κινητή τηλεφωνία). Οι όροι αυτοί αναφέρονται στην αυτόματη προώθηση διαφημιστικών μηνυμάτων, που μπορεί να σχετίζονται και με την τρέχουσα θέση του χρήστη, τα οποία δεν αιτήθηκαν από αυτόν. Αυτό, σε ορισμένες περιπτώσεις, μπορεί να γίνει πολύ ενοχλητικό και να δράσει ακούσια σαν «άρνηση εξυπηρέτησης» (Denial of Service).

Ένα άλλο θέμα είναι η προστασία και ασφάλεια των ευαίσθητων δεδομένων τόσο στα αποθηκευτικά μέσα όσο και κατά τη μεταφορά τους μέσα από δικτυακές ζεύξεις. Ακόμα και αν τηρούνται όλοι οι κανόνες σωστής και νόμιμης χρήσης των δεδομένων αυτών από τους εμπλεκόμενους φορείς και εταιρίες, τα δεδομένα δεν είναι ασφαλή αν περνούν μέσα από ανασφαλή συστήματα. Έτσι δεν πρέπει να ξεχνάμε ότι «μια αλυσίδα είναι τόσο δυνατή, όσο είναι ο πιο αδύναμος κρίκος της». Φυσικά αυτό είναι ένα γενικότερο θέμα της πληροφορικής και των τηλεπικοινωνιών και πρέπει η εμπειρία και γνώση που υπάρχει σε αυτούς τους τομείς να εφαρμοστεί αυστηρά και στην περίπτωση των υπηρεσιών θέσης.

Επίσης, σημασία δεν έχει μόνο η γεωγραφική πληροφορία που σχετίζεται με τη θέση του χρήστη, να μην χρησιμοποιείται παρά την θέλησή του αλλά και να είναι έγκυρη. Η εκούσια και δόλια αλλοίωση των προσωπικών δεδομένων του χρήστη από κάποιον που έχει πρόσβαση σε αυτά, μπορεί να προκαλέσει προβλήματα στον χρήστη, αν αυτά στη συνέχεια χρησιμοποιηθούν από τρίτους.

Τέλος, αξίζει να αναφερθεί κανείς και στο πώς συσχετίζεται το μοντέλο αυτόματης προώθησης, που αναμένεται να εφαρμοστεί ευρέως στα επόμενα χρόνια, με τα θέματα ιδιωτικότητας. Η αυτόματη προώθηση στηρίζεται στην συνεχή παρακολούθηση της θέσης του χρήστη. Αυτό από μόνο του μπορεί να είναι πρόβλημα, επειδή διατηρούνται πλήρη και συνεχή στοιχεία των κινήσεων του χρήστη. Μεγαλύτερη διάρκεια παραμονής των στοιχείων αυτών σημαίνει μεγαλύτερο κίνδυνο για υποκλοπή τους. Επίσης, λόγω της πληρότητάς τους, μπορούν να οδηγήσουν σε πολύ πιο ακριβή και λεπτομερειακά προφίλ χρηστών μετά από κατάλληλη επεξεργασία. Επίσης η δυνατότητα

για αυτόματη προώθηση μπορεί να γίνει, και πιθανότατα θα γίνει σύμφωνα με την μέχρι σήμερα εμπειρία, τρόπος μη αιτούμενης και επιθυμητής διαφήμισης.

### **1.3.3.2 Ειδικές Περιπτώσεις**

Παρόλο που στην πλειοψηφία των περιπτώσεων, επικρίνεται η χρήση της θέσης ενός χρήστη για σκοπούς άσχετους με την παρεχόμενη υπηρεσία, υπάρχουν και κάποιες εξαιρέσεις. Μία από αυτές είναι η χρήση τέτοιων στοιχείων για εξαγωγή στατιστικών στοιχείων και συμπερασμάτων. Για παράδειγμα, ένας τηλεπικοινωνιακός παροχέας μπορεί να χρησιμοποιήσει τέτοια πληροφορία για καλύτερη οργάνωση και διαχείριση του δικτύου του (κυψέλες με μεγάλη κίνηση, «νεκρά» σημεία κτλ). Φυσικά σε αυτή την περίπτωση δεν θα πρέπει να χρησιμοποιούνται τα στοιχεία ταυτοποίησης των χρηστών στο σύστημα. Για να διασφαλιστεί ακόμα καλύτερα αυτό, έχουν προταθεί κάποιες λύσεις, όπως αντιστοίχιση τυχαίων αριθμών στις τερματικές συσκευές. Μία άλλη παρόμοια εφαρμογή είναι η καταγραφή κυκλοφοριακής κίνησης και η ενημέρωση των οδηγών. Η συγκεκριμένη υπηρεσία, που εφαρμόστηκε πειραματικά με επιτυχία στην Φινλανδία και σε άλλες χώρες, εκτιμά την κίνηση από τα κυψελωτά τηλέφωνα των οδηγών και τους ενημερώνει πάλι μέσω αυτών για την τρέχουσα κυκλοφοριακή κατάσταση.

Μια άλλη εξαίρεση μπορεί να είναι οι περιπτώσεις που τέτοια πληροφορία χρησιμοποιείται από κυβερνητικές και δικαστικές αρχές για την έρευνα εγκλημάτων ή απλών παραπόνων. Φυσικά θα πρέπει η πληροφορία θέσης που θα χρησιμοποιηθεί να αφορά μόνο το άτομο που κατέφυγε στις αρχές αυτές. Τέλος, προωθείται σε παγκόσμιο επίπεδο, στις ΗΠΑ έχει ήδη ξεκινήσει (E911), ο αυτόματος και χωρίς έγκριση εντοπισμός όσων κάνουν κλήσεις έκτακτης ανάγκης. Υπάρχει αντίλογος σε αυτή την πρόταση αλλά στην μεγάλη πλειοψηφία τους οι χρήστες πιστεύουν ότι κάτι τέτοιο είναι θεμιτό.

### **1.3.3.3 Μέτρα προστασίας της ιδιωτικότητας**

Ήδη πολλοί φορείς που σχετίζονται με τις υπηρεσίες θέσεις έχουν εντοπίσει τα παραπάνω θέματα και έχουν δραστηριοποιηθεί ώστε να αντεπεξέλθουν στις επιθυμίες αλλά και στα δικαιώματα των χρηστών. Κρίνεται αναγκαίο να επικρατήσει μια γενικότερη αντίληψη, ότι τα δικαιώματα αυτά πρέπει να αντιστοιχούν σε υποχρεώσεις από την πλευρά της βιομηχανίας. Μόνο έτσι θα αναπτυχθεί μια σχέση εμπιστοσύνης ανάμεσα στους χρήστες και στους παροχείς, η οποία θα προωθήσει την αποδοχή και ανάπτυξη των υπηρεσιών θέσης. Επίσης πρέπει να λαμβάνονται όχι μόνο προληπτικά μέτρα, αλλά και να υπάρχουν μηχανισμοί επιβολής ποινών στους παραβάτες. Οι κύριοι φορείς που μπορούν να επιβάλουν τέτοια μέτρα είναι οι Αρχές και οι Οργανισμοί, οι

παροχείς του ασύρματου δικτύου, οι υπόλοιποι «παίκτες» και οι κατασκευαστές κινητών τερματικών συσκευών.

Μερικοί από τους οργανισμούς που ασχολούνται και με θέματα ασφάλειας και ιδιωτικότητας είναι ο CTIA (Cellular Telecommunication Industry Association) [38], ο WLIA και ο OPA (Online Privacy Alliance) [39]. Οι βασικές θέσεις του CTIA είναι:

- να ενημερώνονται οι χρήστες-πελάτες σχετικά με την συλλογή και χρήση των πληροφοριών θέσης τους.
- να παρέχεται στον χρήστη η ευχέρεια να συναινέσει για την συλλογή τέτοιων πληροφοριών πριν αυτές χρησιμοποιηθούν.
- να διασφαλίζεται η ασφάλεια και η ακεραιότητα των δεδομένων και να δίνεται στον χρήστη πρόσβαση σε αυτά.

Ο WLIA επεκτείνει αυτές τις θέσεις και συμπληρώνει ότι πρέπει να γίνεται κάθε δυνατή προσπάθεια για να διασφαλιστεί ότι τα δεδομένα θέσης είναι ακριβή. Πιο συγκεκριμένα, υποχρεώνει τα μέλη του να αναφέρουν ρητά την πολιτική ιδιωτικότητας (privacy policy) που υιοθετούν και η οποία πρέπει να είναι συνεπής με την δική του. Τα σχετικά κείμενα πρέπει να είναι εύκολα προσβάσιμα σε όλους τους χρήστες. Κάθε μέλος του πρέπει να δημοσιοποιεί πως δημιουργείται και χρησιμοποιείται η πληροφορία θέσης από αυτό και πως προστατεύει την ιδιωτικότητα των χρηστών. Επίσης πρέπει να αποφεύγεται η συλλογή τέτοιας πληροφορίας όπου δεν αποτελεί λειτουργικό στοιχείο της παρεχόμενης εφαρμογής ή υπηρεσίας. Τέλος δεν πρέπει αυτά τα δεδομένα να μένουν αποθηκευμένα περισσότερο απ' όσο χρειάζεται για τον σκοπό που συλλέχθηκαν ή δημιουργήθηκαν.

Όσον αφορά την συμμετοχή τρίτων «παικτών», θα πρέπει να υπάρχει ένα καλώς ορισμένο πλαίσιο μέσα στο οποίο θα πρέπει να γίνεται η διακίνηση γεωγραφικής πληροφορίας μεταξύ του τηλεπικοινωνιακού παροχέα και των τρίτων (παροχείς περιεχομένου, εφαρμογών κτλ). Μέρος αυτού του πλαισίου θα πρέπει να είναι και ό,τι σχετίζεται με την ενημέρωση του τελικού χρήστη. Δηλαδή οι κανόνες που διέπουν την χρήση τέτοιας πληροφορίας δεν θα πρέπει να μένουν στο εταιρικό επίπεδο αλλά η συνεργασία των εταιριών να είναι διαφανής για τον χρήστη.

Γενικά οι περισσότερες προσπάθειες των οργανισμών, εμπορικών ή κυβερνητικών, προσανατολίζονται στην προώθηση της αυτο-ρύθμισης (self-regulation). Οι τηλεπικοινωνιακοί φορείς είναι αυτοί που έχουν τις μεγαλύτερες δυνατότητες για διασφάλιση της ιδιωτικότητας. Καταρχήν έχουν ένα βασικό κίνητρο: είναι οι κύριοι εμπορικοί εκμεταλλευτές αυτών των νέων τεχνολογιών, τουλάχιστον προσωρινά, και έτσι ενδιαφέρονται αρκετά για την αποδοχή τους από το αγοραστικό κοινό. Πέρα από αυτό όμως, έχουν και την μοναδική ιδιότητα να είναι αυτοί που δημιουργούν τα δεδομένα θέσης του χρήστη.

Σήμερα έχει γίνει σχεδόν ευρέως αποδεκτό ένα μοντέλο διασφάλισης της ιδιωτικότητας που βασίζεται σε τρεις έννοιες: Standard Opt-in, Confirmed Opt-in και Opt-out. Οι έννοιες αυτές δεν είναι καινούριες. Προέρχονται από τον χώρο του μάρκετινγκ που βασίζεται σε αποστολή διαφημιστικών μηνυμάτων ηλεκτρονικού ταχυδρομείου. Στις υπηρεσίες θέσης όμως το νόημά τους αναπροσαρμόζεται κάπως.

Το "Standard Opt-in" (ή "Single Opt-in") ορίζεται σαν μια διαδικασία που απαιτεί ενεργή επιλογή από την πλευρά του χρήστη, για να εκφράσει την συναίνεσή του σε κάθε μεμονωμένη χρήση μιας υπηρεσίας. Για παράδειγμα, το «πάτημα» της επιλογής «Εντόπισέ με» σε μια ασύρματη υπηρεσία θέσης είναι έκφραση συγκατάθεσης για

εντοπισμό. Ένα άλλο παράδειγμα αυτής της διαδικασίας είναι το να δώσει ο χρήστης άδεια για συνεχή εντοπισμό του μέσω κάποιας παρόμοιας επιλογής.

Το “Confirmed Opt-in” (ή “Double Opt-in”) είναι μια επέκταση της προηγούμενης διαδικασίας και ενέχει και την ρητή επικύρωση της άδειας (permission) που ο χρήστης δίνει σε μια υπηρεσία. Αυτή μπορεί να γίνεται είτε τη στιγμή που ο χρήστης αιτείται την υπηρεσία, ή σαν μια διαδικασία επιβεβαίωσης ότι ο χρήστης έχει ήδη δώσει την άδειά του για παροχή της υπηρεσίας, για καθορισμένο χρονικό διάστημα που του έχει γνωστοποιηθεί. Μια τέτοια άδεια μπορεί να δοθεί κατά την ενεργοποίηση μιας υπηρεσίας θέσης, κατά την εγγραφή του χρήστη στην υπηρεσία ή οποιαδήποτε άλλη στιγμή. Επίσης η άδεια πρέπει να είναι σε μια μορφή που να μπορεί να χρησιμοποιηθεί από την διαδικασία επιβεβαίωσης (κείμενο, ηλεκτρονική μορφή κα). Άρα το “Confirmed Opt-in” εισάγει ακόμη ένα επίπεδο προστασίας του χρήστη.

Το “Opt-out” ορίζεται σαν η διαδικασία με την οποία ο χρήστης με δική του ενέργεια αρνείται να δώσει την άδειά του για εντοπισμό ή αποσύρεται από την τρέχουσα υπηρεσία, ανεξάρτητα αν έχει προηγουμένως κάνει “Opt-in”. Για να είναι επιτυχημένη μια τέτοια διαδικασία πρέπει να έχει υποστηρίζεται σωστά από την υπηρεσία. Δηλαδή να μπορεί ο χρήστης να την σταματήσει όποτε θέλει, να έχει άμεσο αποτέλεσμα και να μπορεί να επιλεγεί εύκολα (χωρίς την περιήγηση σε πολύπλοκα μενού). Στη γενική περίπτωση, προτιμάται το “Opt-in” από το “Opt-out”.

Εκτός από τους τηλεπικοινωνιακούς φορείς και παροχείς υπηρεσιών όμως, μπορούν να προστατεύσουν τον χρήστη και οι κατασκευαστές των τερματικών συσκευών. Ένας τρόπος με τον οποίο μπορεί να υλοποιηθεί αυτό είναι να εκπέμπει το κινητό ένα επιπλέον σήμα που να είναι μέρος του σήματος ταυτοποίησης (unit identification signal), το οποίο ούτως ή άλλως εκπέμπεται περιοδικά. Το επιπλέον αυτό σήμα θα υποδεικνύει αν ο χρήστης επιθυμεί να εντοπίζεται η συσκευή του. Η καλύτερη όμως λύση είναι το κινητό να έχει τον πλήρη έλεγχο της θέσης του. Αυτό μπορεί να γίνει μόνο αν η τεχνική εντοπισμού θέσης την οποία χρησιμοποιεί το ασύρματο δίκτυο βασίζεται, ή τουλάχιστον υποβοηθείται, από το κινητό. Αυτό σημαίνει ότι πρέπει να χρησιμοποιηθούν τεχνικές εντοπισμού θέσης σαν τις A-GPS και E-OTD, στις οποίες ο χρήστης θα μπορεί να επιλέξει αν θα σταλεί η θέση του στο υπόλοιπο δίκτυο. Μια τέτοια λύση όμως δεν μπορεί να εφαρμοστεί άμεσα καθώς το κόστος της είναι απαγορευτικό.

Μερικά από τα μέτρα που προαναφέρθηκαν φαίνονται να υπόσχονται υψηλό βαθμό προστασίας της ιδιωτικότητας. Από την άλλη πλευρά όμως, η υλοποίησή τους σε ένα σύστημα που βασίζεται στην αυτόματη προώθηση δεν φαίνεται να είναι εφαρμόσιμη ή επιθυμητή. Για παράδειγμα, η διαδικασία του “Confirmed Opt-in” μπορεί να είναι αρκετά αποτελεσματική σε περιπτώσεις Pull υπηρεσιών, αλλά δύσκολα θα την δεχόταν κάποιος χρήστης που έχει εγγραφεί σε μια υπηρεσία αυτόματης προώθησης. Στην πραγματικότητα όταν κάποιος εγγράφεται σε μια τέτοια υπηρεσία δέχεται και το γεγονός ότι η θέση του θα παρακολουθείται συνεχώς (ή έστω τακτικά) από το δίκτυο. Έτσι του φαίνεται πολύ ενοχλητικό να το επικυρώνει αυτό συνέχεια. Αυτή ακριβώς η συνεχής «παρακολούθηση» είναι και ο λόγος που δεν έχει μεγάλο νόημα η χρήση τεχνολογιών όπως το A-GPS. Τέτοιες τεχνολογίες σε ένα περιβάλλον αυτόματης προώθησης απλά επιβαρύνουν με φόρτο σηματοδότησης το δίκτυο.

Όλα τα παραπάνω μέτρα αναφέρονται κυρίως στην πρόληψη πιθανών παραβιάσεων της ιδιωτικότητας. Εκτός από αυτά, όμως, θα πρέπει να υπάρχουν και μηχανισμοί επιβολής κυρώσεων στους παραβάτες των αποφασισμένων κανονισμών.

Αυτοί οι κανονισμοί μπορεί να έχουν την μορφή οδηγιών από βιομηχανικούς οργανισμούς (WLIA) ή να είναι επίσημοι νόμοι. Σήμερα το νομικό πλαίσιο για αυτά τα θέματα είναι μάλλον ανεπαρκές και δεν μπορεί να παρακολουθήσει τις ταχύτατες τεχνολογικές εξελίξεις. Έτσι δεν μπορούμε να βασιζόμαστε σήμερα σε ένα πλήρη και αυστηρό δικαστικό μηχανισμό επιβολής ποινών. Οι υπόλοιποι οργανισμοί όμως που σχετίζονται με τις υπηρεσίες θέσης μπορούν και πρέπει να επιβάλουν κυρώσεις. Μερικοί έχουν ήδη μεριμνήσει γι' αυτό. Ο WLIA, για παράδειγμα, προβλέπει κάποια σώματα που θα ασχολούνται με τέτοιες υποθέσεις (Committee on Standards and Self-Regulation, Board of Directors), τα οποία έχουν την αρμοδιότητα ακόμα και να διαγράψουν ένα μέλος-παραβάτη ή να το αναφέρουν σε άλλες αρμόδιες Αρχές. Πάντως προκειμένου να υπάρχει κοινή αποδοχή των όποιων κυρώσεων, αυτές πρέπει να συμφωνούν και με το σχετικό Δίκαιο.

Η πιο ουσιαστική από όλες τις λύσεις όμως, ας μην ξεχνάμε ότι είναι η ανθρώπινη ευσυνειδησία. Όπως και με όλες τις άλλες τεχνολογίες, έτσι και στις υπηρεσίες θέσης ο ανθρώπινος παράγοντας είναι αυτός που θα καθορίσει τα αρνητικά και θετικά τους. Τέλος, όπως είδαμε, η ιδιωτικότητα και η λειτουργικότητα μπορεί μερικές φορές να συγκρούονται. Καθώς οι υπηρεσίες αυτές βρίσκονται ακόμη στην αρχική τους φάση, μόνο ο χρόνος μπορεί να δείξει τι θα θέσουν οι χρήστες σαν μεγαλύτερη προτεραιότητα: την αυξημένη λειτουργικότητα των υπηρεσιών ή την μέγιστη διασφάλιση της ιδιωτικότητάς τους, με πιθανώς μικρότερη υποστηριζόμενη λειτουργικότητα.

## 1.4 Πλατφόρμες για Υπηρεσίες Θέσης

Πολλές εταιρείες τηλεπικοινωνιών ανά τον κόσμο σε συνεργασία με πανεπιστημιακά ιδρύματα έχουν αναπτύξει πλατφόρμες οι οποίες δίνουν την δυνατότητα στους παροχείς να προσφέρουν Υπηρεσίες Εξαρτώμενες από τη Θέση. Οι πιο γνωστές πλατφόρμες αυτή τη στιγμή είναι οι:

1. LocatioNet Platform
2. Celebriety Platform
3. LoL@ platform
4. Cellpoint Platform
5. IBM's WebSphere Everyplace Server
6. Webraska SmartZone Geospatial Platform
7. PoLoS (υπό ανάπτυξη)

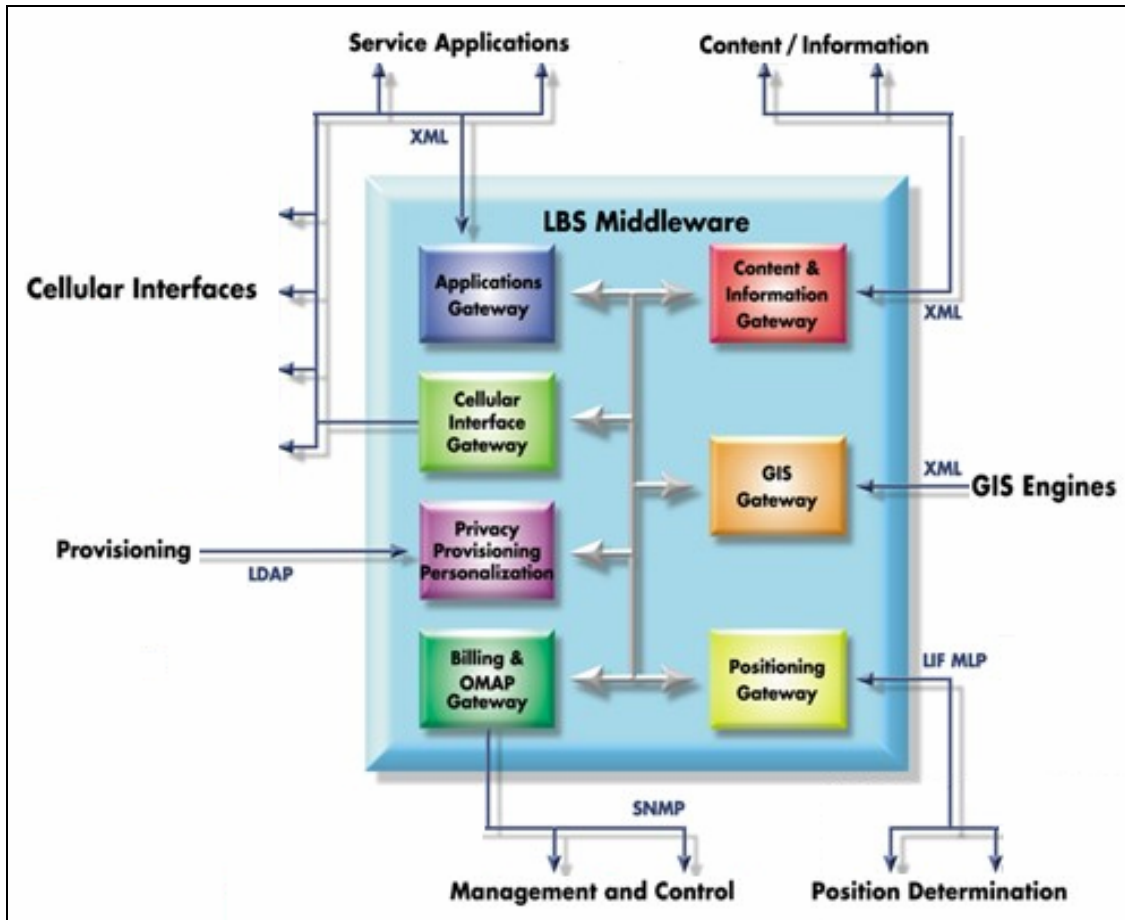
### 1.4.1 LocatioNet Platform [40]

Η πλατφόρμα αυτή προσφέρει μία ολοκληρωμένη λύση για τους παροχείς ασύρματων υπηρεσιών επιβοηθούμενη από ένα σύστημα GIS αλλά και πολλές άλλες υπηρεσίες. Επιπρόσθετες υπηρεσίες αλλά και εξοπλισμός όπως για τα συστήματα εντοπισμού θέσης (positioning) μπορούν να προσφερθούν κάλλιστα από άλλες εταιρείες.

Η δομή της είναι κεντρικοποιημένη (centralized), προσφέροντας διαχείριση και έλεγχο όλων των επιμέρους δομικών στοιχείων του πυρήνα, αλλά και των εφαρμογών που σχετίζονται με τις προσφερόμενες υπηρεσίες, μέσω μίας ενοποιημένης διεπαφής. Μπορούμε να την διακρίνουμε στα εξής μέρη:

- Μία πραγματικού χρόνου μηχανή διανομής υπηρεσιών (real-time service delivery engine), η οποία σαν ρόλο έχει την διαχείριση όλων των σχετιζόμενων ως προς μία υπηρεσία στοιχείων, όπως εφαρμογές της συγκεκριμένης υπηρεσίας, αναγνώριση θέσης κ.α.
- Μία διεπαφή API και ένα σύνολο από εργαλεία ανάπτυξης λογισμικού SDK τα οποία προσφέρουν την δυνατότητα και στον παροχέα των ασύρματων υπηρεσιών αλλά και στις άλλες εμπλεκόμενες εταιρείες να αναπτύξουν εφαρμογές για υπηρεσίες Εξαρτώμενες από την Θέση.
- Τέλος το σύστημα GIS που προσφέρει υπηρεσίες γεωδαισίας, και καθοδήγησης.

Η αρχιτεκτονική της πλατφόρμας απεικονίζεται στην παρακάτω εικόνα



**Εικόνα 41: Αρχιτεκτονική της Πλατφόρμας LocationNet**

Η επικοινωνία με τα συστήματα GIS και τις βάσεις δεδομένων επιτυγχάνεται με την βοήθεια της XML η οποία χρησιμοποιείται επίσης και για την διεπαφή μεταξύ των διαφορετικών δομικών στοιχείων της πλατφόρμας.

Οι ασύρματες συσκευές (τερματικά) μπορούν να έχουν πρόσβαση στην πλατφόρμα μέσω SMS, φωνής, WAP ή HTTP πρωτοκόλλων. Είναι λοιπόν αυτονόητο ότι οι χρήστες θα πρέπει να είναι εξοπλισμένοι με συσκευές τέτοιες που να υποστηρίζουν τα προαναφερθέντα. Κάθε συσκευή έχει ορισμένα τεχνικά χαρακτηριστικά τα οποία είναι γνωστά στο σύστημα όπως ανάλυση οθόνης, μέγεθος κ.α.

Η τεχνολογία που χρησιμοποιείται για τον εντοπισμό της θέσης (positioning) μπορεί να προσφέρεται από το ίδιο το δίκτυο (Cell ID, AOA/ TDOA, E-OTD, STK) ή να γίνεται με την βοήθεια GPS.

Οι πιο γνωστές υπηρεσίες που προσφέρει αυτή η πλατφόρμα είναι :

- Πλοήγηση (Navigation)
- Διαφημιστικές υπηρεσίες (Mobile Commerce)
- Διαχείριση Στόλου (Fleet Management)
- Ιχνηλασία (Tracking)

## 1.4.2 **Celebrity Platform [41]**

Αποτελεί πρόταση της εταιρείας CT Motion και δίνει την δυνατότητα για την δημιουργία και την διαχείριση των εξαρτώμενων από τη θέση υπηρεσιών (LBS). Παρέχει υπηρεσίες πραγματικού χρόνου, όπου πολλές εφαρμογές επικοινωνούν και ανταλλάσσουν δεδομένα μεταξύ τους με δυναμικό τρόπο ανάλογα με το προφίλ του χρήστη αλλά και με τα τεχνικά χαρακτηριστικά του τερματικού του. Αυτές οι εφαρμογές αναπτύσσονται από την CT Motion αλλά και από άλλες εταιρείες.

Η Celebrity Platform διαχειρίζεται έναν αριθμό από διεπαφές με κέντρα SMS (SMS centers), συστήματα τιμολόγησης, πύλες WAP (WAP gateways) και συστήματα διαχείρισης δικτύων. Επίσης, πληροφορίες και δεδομένα για τις προσφερόμενες υπηρεσίες μπορούν να είναι προσβάσιμα μέσω του Internet. Η πλατφόρμα συμπεριλαμβάνει μία διεπαφή (API) και ένα σύνολο από εργαλεία ανάπτυξης λογισμικού (SDK), τα οποία προσφέρουν την δυνατότητα ανάπτυξης των εφαρμογών για τις υπηρεσίες. Η επικοινωνία με τα τερματικά μπορεί να επιτευχθεί μέσω SMS, WAP ή με το πρωτόκολλο IP, ενώ για την πρόσβαση στην πλατφόρμα χρησιμοποιούνται κινητά ή PDAs.

Το σύστημα διαθέτει επίσης και την μονάδα εντοπισμού θέσης (positioning) που δίνει την δυνατότητα στις εφαρμογές να γνωρίζουν τη θέση του χρήστη. Οι τεχνολογίες που χρησιμοποιούνται είναι οι: Cell ID, Enhanced Cell ID, GMLC και GPS. Η επιλογή εξαρτάται από δυνατότητα που έχει το τερματικό του χρήστη. Λαμβάνονται υπ' όψιν πολλές παράμετροι όπως η ανάλυση της οθόνης, το κόστος κ.α.

## 1.4.3 **LoL@ Platform [42]**

Η LoL@ (Local Location Assistant) είναι ένα ερευνητικό πρόγραμμα συνεργασίας μεταξύ εταιρειών τηλεπικοινωνιών και του πανεπιστημίου FTW (Forschungszentrum Telekommunikation Wien) της Αυστρίας. Προσφέρει μία εικόνα για τις μελλοντικές UMTS υπηρεσίες και τις απαιτήσεις τους. Αυτό που προτείνεται είναι μία αρχιτεκτονική για υπηρεσίες, οι οποίες θα σχετίζονται με πολυμέσα (multimedia) όπως εξελιγμένες διεπαφές για τον τελικό χρήστη, αναγνώριση φωνής κ.α. Η αρχιτεκτονική χωρίζεται σε τρία μέρη: στο τερματικό, στο δίκτυο και στον server (terminal, mobile network and server domain).

Το πρώτο μέρος (terminal domain) αποτελείται από το κινητό τερματικό του χρήστη, την κάρτα SIM και ένα περιβάλλον εκτέλεσης εφαρμογών όπως web browser ή media player. Το δεύτερο μέρος (mobile network) είναι το δίκτυο μέσω του οποίου επιτυγχάνεται η επικοινωνία με το τερματικό. Η πλατφόρμα χρησιμοποιεί υποδομή βασισμένη στη μεταγωγή πακέτων (packet switched) όπως GSM/GPRS και UMTS. Τέλος το τρίτο μέρος (server domain) περιλαμβάνει όλα εκείνα τα δομικά στοιχεία που επεκτείνουν την λειτουργικότητα του συστήματος όπως τις βάσεις δεδομένων του δικτύου (HLR/HSS, VLR, κα) και τον εξοπλισμό για τον εντοπισμό της θέσης. Όλα αυτά είναι τυποποιημένα από το 3GPP και χρησιμοποιούνται πρωτόκολλα όπως το MAP (Mobile Application Part).



Η επικοινωνία μεταξύ των διαφόρων συστημάτων πραγματοποιείται χρησιμοποιώντας CORBA, ενώ παράλληλα χρησιμοποιούνται τυποποιημένα πρωτόκολλα όπως TCP/IP, Bluetooth αλλά και το πρωτόκολλο IP/HTTP που υποστηρίζει την επικοινωνία για τα τετραγενικά της τρίτης γενιάς (3G).

Η θέση του τετραγενικού ανακτάται χρησιμοποιώντας τεχνολογίες όπως GPS, Assisted GPS, Bluetooth καθώς και αλγόριθμους βασισμένους στις πληροφορίες της κυψέλης (Cell ID). Το σύστημα είναι σε θέση να επιλέξει την πιο κατάλληλη τεχνολογία ανάλογα με την περίπτωση. Πέραν τούτου όμως μπορεί να υπάρχουν πιο ακριβή αποτελέσματα για τη θέση χρησιμοποιώντας την ιχνηλασία (tracking) η οποία διατηρεί ένα ιστορικό για τις προηγούμενες θέσεις του κινητού έτσι ώστε να μην υπάρχουν αμφιβολίες για την παρούσα θέση. Επιπλέον χρησιμοποιείται και ο διαδραστικός εντοπισμός θέσης (interactive positioning) κατά τον οποίο ο χρήστης από μόνος του επιβεβαιώνει την γεωγραφική του θέση.

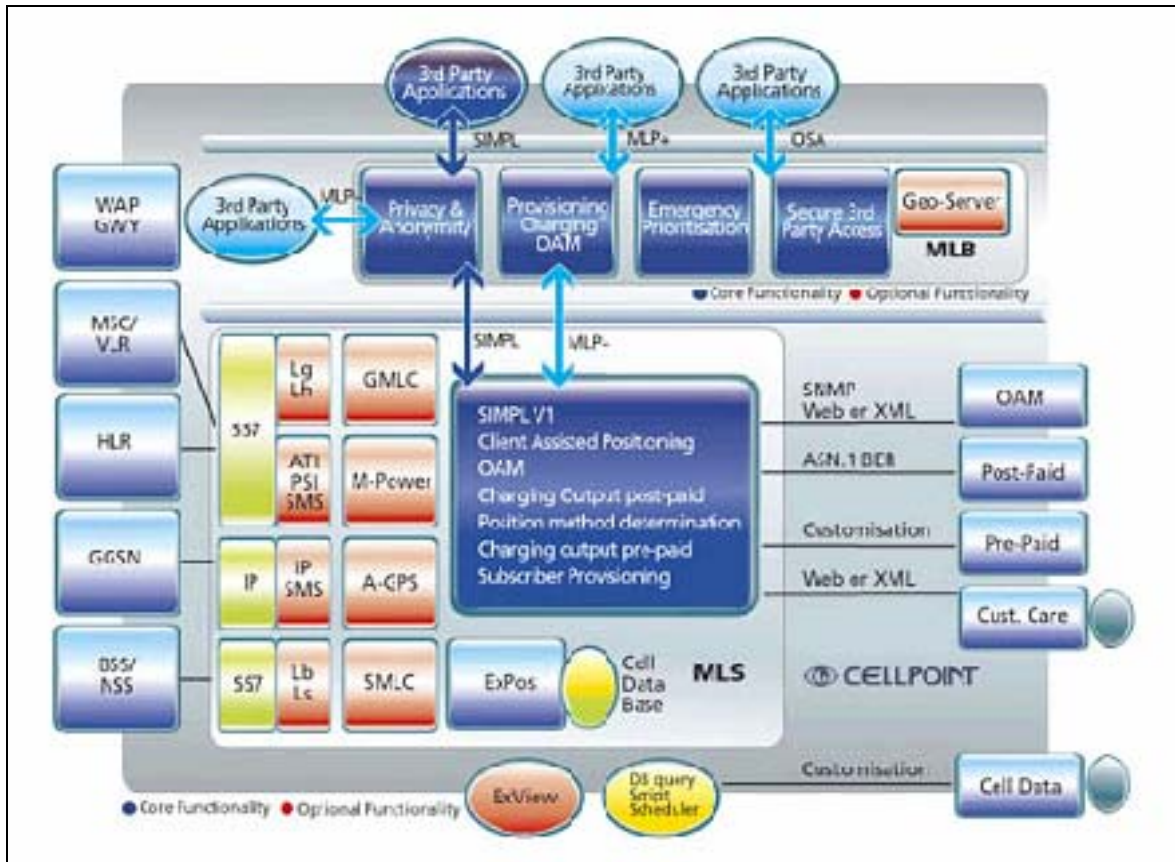
Η πλατφόρμα ουσιαστικά αποτελεί μία υπηρεσία πλοήγησης (navigation) για τις πόλεις προσφέροντας πληροφορίες για σημεία ενδιαφέροντος όπως μουσεία, εστιατόρια κ.α. Τα απαραίτητα συστατικά της είναι ο εντοπισμός θέσης (positioning), σε συνδυασμό με επικοινωνίες πολυμέσων (multimedia communications) και την αναγνώριση και επεξεργασία φωνής.

#### **1.4.4 Cellpoint Platform [43]**

Η πλατφόρμα αυτή δίνει την δυνατότητα στον παροχέα να προσφέρει υπηρεσίες εξαρτώμενες από τη θέση. Το όλο σύστημα αποτελείται από τα εξής μέρη:

- Τον Mobile Location Broker (MLB) που διαχειρίζεται την γνησιότητα (authentication) και την εξουσιοδότηση (authorization) των χρηστών.
- Τον mPower Location Provider που παρέχει την διεπαφή με το δίκτυο GSM μέσω του πρωτοκόλλου MAP (Mobile Application Part) και περιέχει λειτουργίες για την ανάκτηση της θέσης μέσω του δικτύου αυτού.
- Τον Mobile Location Server (MLS) που υπολογίζει την θέση του τετραγενικού μέσω των πληροφοριών που του παρέχονται από το δίκτυο αλλά περιέχει επίσης λειτουργίες διαχείρισης και τιμολόγησης.

Η αρχιτεκτονική της πλατφόρμας παρουσιάζεται στην επόμενη εικόνα



Εικόνα 42: Αρχιτεκτονική της Πλατφόρμας Cellpoint

Η επικοινωνία με τα πληροφοριακά συστήματα και τις βάσεις δεδομένων βασίζεται στην γλώσσα XML και στο πρωτόκολλο SNMP ενώ οι διεπαφές μεταξύ διαφόρων δομικών στοιχείων βασίζονται αποκλειστικά στην XML. Για την επικοινωνία με τα τελεματικά χρησιμοποιούνται SMS, HTTP και WAP.

Οι τεχνολογίες εντοπισμού θέσης είναι οι εξής :

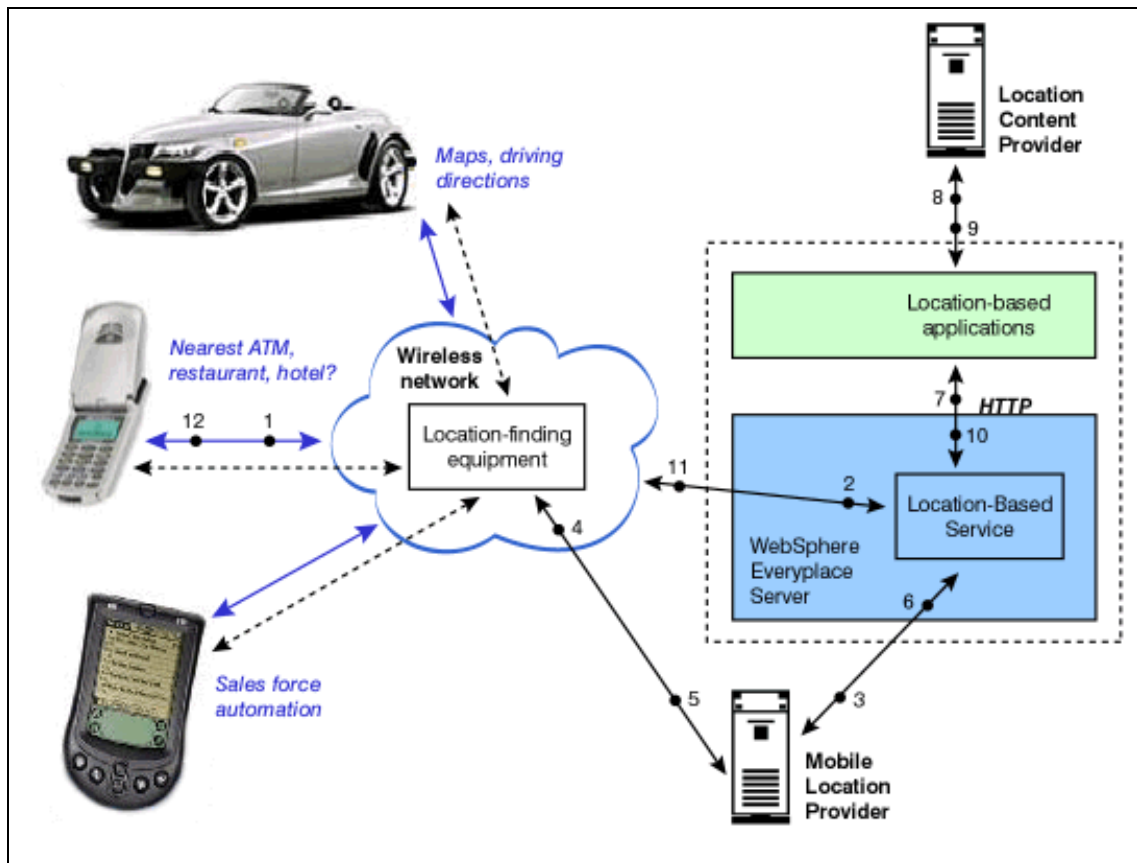
- mPower (τεχνολογία της Cellpoint )
- A-GPS
- E-OTD
- Cell ID

Οι υπηρεσίες που προσφέρονται είναι οι:

- Σημεία ενδιαφέροντος (Points of Interest)
- Αναζήτηση προσώπου (Friend Finder)
- Πλοήγηση (Navigation)
- Διαχείριση Στόλου (Fleet Management and Control)

### 1.4.5 IBM's WebSphere Everyplace Server [44]

Αποτελεί μία πλατφόρμα που εκτείνει τις υπάρχουσες εφαρμογές για το ασύρματο ηλεκτρονικό εμπόριο (mobile e-business). Η αρχιτεκτονική παρουσιάζεται στη επόμενη εικόνα .



Εικόνα 43: Αρχιτεκτονική της Πλατφόρμας WebSphere Everyplace Server

Η επικοινωνία με τα υπόλοιπα πληροφοριακά συστήματα αλλά και μεταξύ των ίδιων των δομικών μερών βασίζεται στην γλώσσα XML. Για πρόσβαση στην πλατφόρμα μπορούν να χρησιμοποιηθούν κινητά τηλέφωνα με την χρήση SMS και WAP καθώς και PDAs. Τα πρωτόκολλα που χρησιμοποιούνται είναι κυρίως τα WAP, HTTP/IP, SNPPP και SMTP.

Η τεχνολογία που χρησιμοποιείται για τον εντοπισμό της θέσης μιας τερματικής συσκευής είναι κυρίως η Cell ID.

Οι υπηρεσίες που προσφέρει η πλατφόρμα είναι οι εξής:

- Σημεία ενδιαφέροντος (Points of Interest)
- Αναζήτηση προσώπου (Friend Finder)
- Πλοήγηση (Navigation )
- Υπηρεσίες έκτακτης ανάγκης (Emergencies)

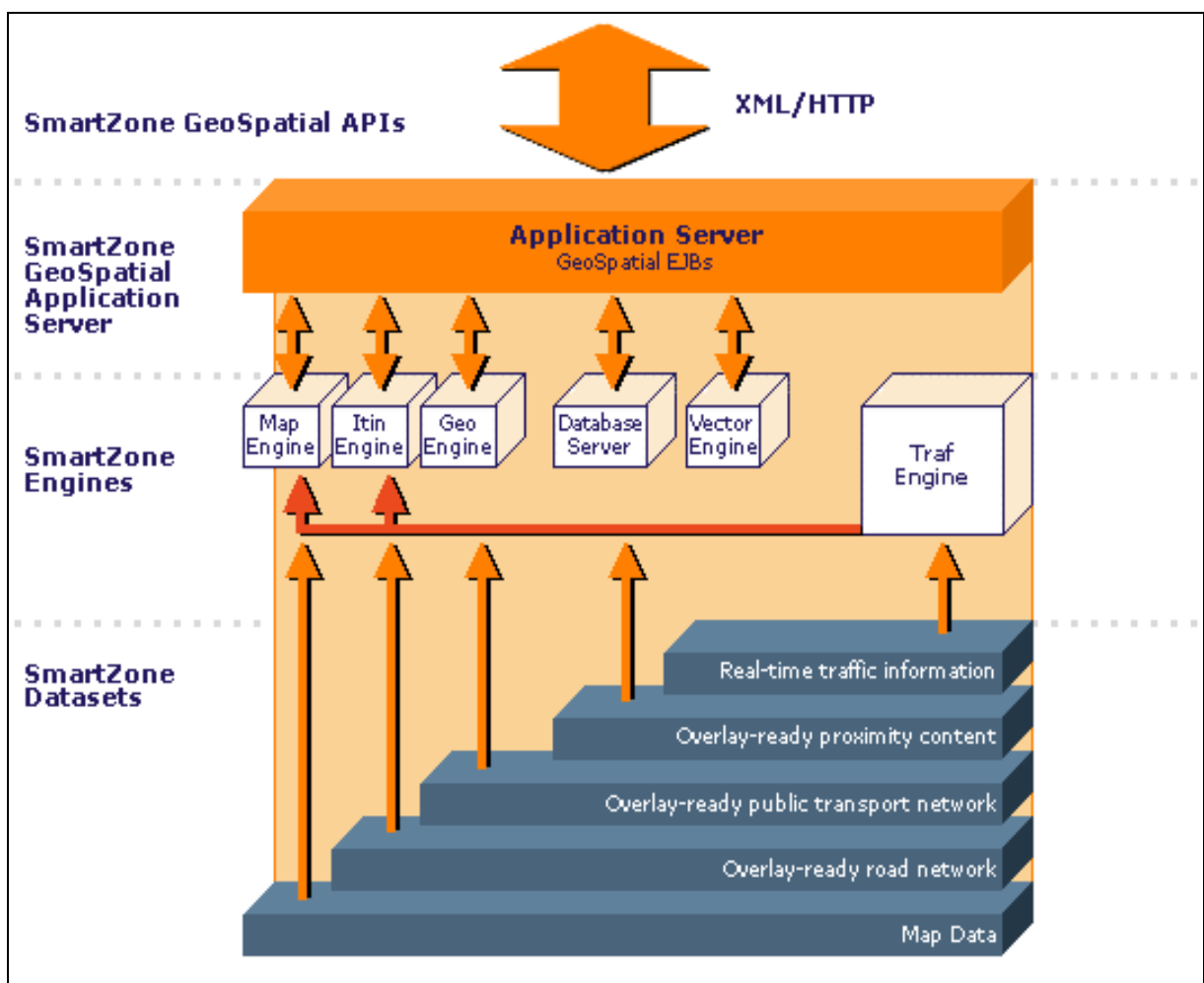
## 1.4.6 Webraska SmartZone Geospatial Platform [45]

Είναι μία πλατφόρμα που βασίζεται σε μια καταναμημένη αρχιτεκτονική τριών επιπέδων (three-tier distributed architecture) και περιλαμβάνει αποκλειστικά τεχνολογίες και ανοιχτά πρότυπα (open standards) όπως τα J2EE, CORBA, JDBC κ.α.

Αποτελείται από τα εξής τρία μέρη:

- SmartZone GeoSpatial Application Server
- SmartZone GeoSpatial Engines
- SmartZone Datasets

Η εικόνα που ακολουθεί αποτυπώνει λεπτομερέστερα την αρχιτεκτονική της πλατφόρμας.



Εικόνα 44: Η πλατφόρμα Smartzone Geospatial

Η επικοινωνία με τα πληροφοριακά συστήματα υλοποιείται με την βοήθεια XML, HTTP ή JDBC ενώ με τα άλλα στοιχεία χρησιμοποιείται κατά βάση CORBA. Η πλατφόρμα είναι συμβατή με τα συστήματα GSM, GPRS, EDGE, CDMA, W-CDMA

και είναι προσπελάσιμη από κινητά, PDAs, smartphones χρησιμοποιώντας SMS, WAP ή XML/HTTP.

Τέλος ο εντοπισμός της θέσης πραγματοποιείται με τεχνολογίες όπως Cell ID, Network Triangulation και GPS.

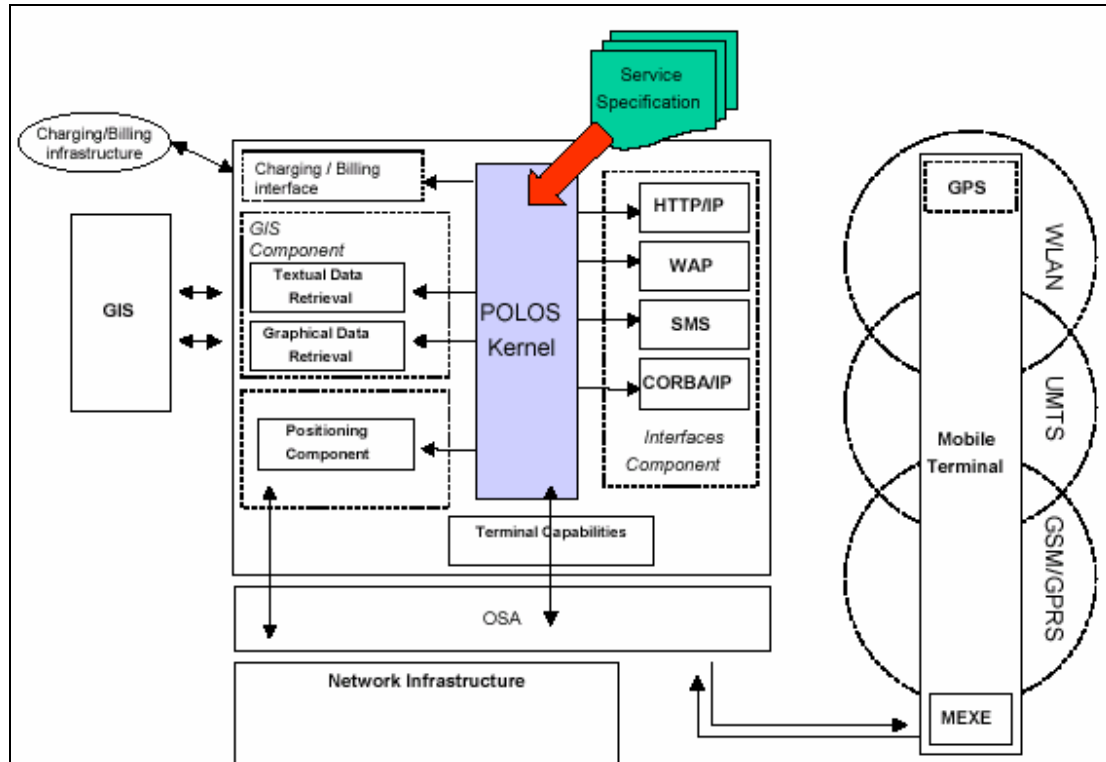
Οι υπηρεσίες που προσφέρει είναι :

- Υπηρεσίες εύρεσης προσώπου ή αντικειμένου (Finder services)
- Υπηρεσίες δρομολόγησης και πλοήγησης (Routing and Navigation services)
- Υπηρεσίες για το κοινό (Community applications)

### 1.4.7 PoLoS [46]

Η πλατφόρμα αυτή αναπτύσσεται αυτή τη στιγμή στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών σε συνεργασία με πολλές ελληνικές και ξένες εταιρείες τηλεπικοινωνιών. Αποσκοπεί στο να προσφέρει εξίσου στον παροχέα των υπηρεσιών (service provider) αλλά και στον τελικό χρήστη όλη την λειτουργικότητα για εκτέλεση των LBS.

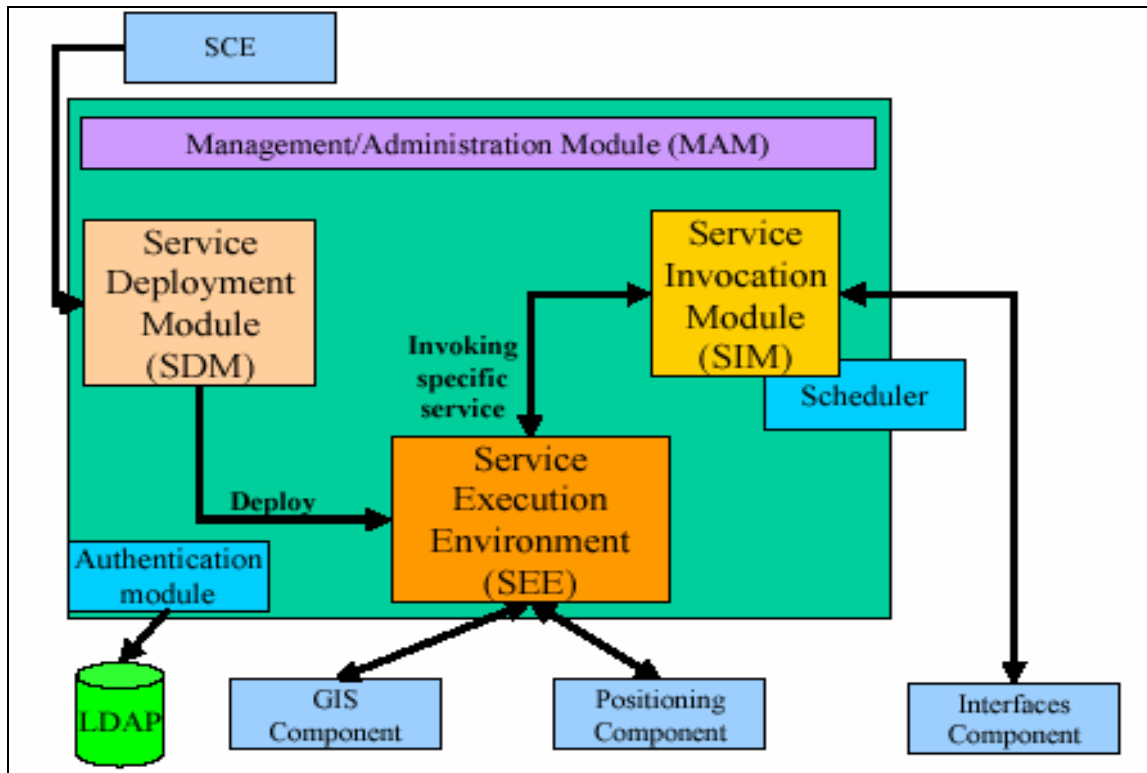
Η αρχιτεκτονική του PoLoS αποτελείται από δύο κυρίως οντότητες (όπως φαίνεται και στη εικόνα): τον Πυρήνα ( PoLoS Kernel) και τα Περιφερειακά Δομικά Στοιχεία ( Peripheral Components).



Εικόνα 45: Η Αρχιτεκτονική της πλατφόρμας PoLoS

## Ο Πυρήνας

Ο Πυρήνας αποτελεί το πιο σημαντικό στοιχείο της όλης πλατφόρμας που έχει σαν σκοπό να συντονίζει τα υπόλοιπα μέρη έτσι ώστε να προσφερθεί στον τελικό χρήστη μία υπηρεσία LBS. Η παρακάτω εικόνα δείχνει την αρχιτεκτονική του Πυρήνα.



Εικόνα 46: Ο πυρήνας του PoLoS

## Τα Περιφερειακά Δομικά Στοιχεία

Τρία βασικά περιφερειακά στοιχεία διακρίνονται στην αρχιτεκτονική του PoLoS όπως δείχνει και η Εικόνα 2.4.7.2:

- Το στοιχείο GIS, που είναι υπεύθυνο για την επικοινωνία με τις βάσεις δεδομένων GIS και προσφέρει πληροφορίες εξίσου σε γραφική μορφή ή μορφή κειμένου.
- Το στοιχείο του εντοπισμού θέσης (Positioning component), που παρέχει στο σύστημα τις απαραίτητες πληροφορίες για την θέση του χρήστη που ζήτησε μια συγκεκριμένη υπηρεσία.
- Το στοιχείο των διεπαφών ( Interfaces component), που είναι υπεύθυνο για την επικοινωνία της πλατφόρμας με οντότητες έξω από αυτήν όπως π.χ. με το κινητό.

Οι τεχνολογίες εντοπισμού θέσης (positioning) είναι παρόμοιες με αυτές των άλλων πλατφόρμων που περιγράφηκαν προηγουμένως δηλαδή:

- Cell ID

- TOA. Time of Arrival.
- E-OTD. Enhanced Observed Time Difference.
- Network triangulation.
- GPS

## 1.5 Το Παρόν και το Μέλλον των Υπηρεσιών Θέσης

Οι υπηρεσίες θέσης παρόλο που αντιμετωπίστηκαν με πολύ ενθουσιασμό από τους τηλεπικοινωνιακούς φορείς όταν πρωτοεμφανίστηκαν, δεν προκαλούν τον ίδιο ενθουσιασμό στους χρήστες των κινητών δικτύων σήμερα. Οι λόγοι είναι πολλοί και αρκετοί από αυτούς έχουν θιχτεί σε προηγούμενες ενότητες. Σε αυτήν την ενότητα θα τους συνοψίσουμε και θα επισημάνουμε τα βήματα που πρέπει να γίνουν ώστε να υπάρξει αυξημένη διείσδυση και λειτουργικότητα των υπηρεσιών αυτών.

### 1.5.1 Σημερινά προβλήματα και πιθανές λύσεις

Τα προβλήματα που αντιμετωπίζουν σήμερα οι LBS υπηρεσίες είναι πολλά και διαφορετικών τύπων. Τα πιο πολλά από αυτά είναι τεχνικής φύσης. Ένα πολύ σημαντικό τεχνικό θέμα είναι η τεχνολογία εντοπισμού θέσης. Τα συστήματα εντοπισμού θέσης βρίσκονται ακόμη σε πρώιμο στάδιο και όσον αφορά τις ικανότητές τους και όσον αφορά την εξάπλωση και διαθεσιμότητά τους. Οι υπάρχουσες τεχνικές εντοπισμού θέσης μπορούν να εντοπίσουν ένα τερματικό με προσέγγιση τουλάχιστον 50 μέτρα και δεν αποδίδουν καλά σε εσωτερικούς χώρους. Έτσι πολλές από τις πιο χρήσιμες υπηρεσίες που ανήκουν στο μοντέλο των LBS δεν μπορούν να εφαρμοστούν σήμερα (πχ οδική πλοήγηση). Επίσης δεν έχει εξαπλωθεί η εγκατάσταση συστημάτων εντοπισμού θέσης στα υπάρχοντα δίκτυα και δεν έχει αρχίσει η ευρεία διάθεση τερματικών συσκευών που να υποστηρίζουν εντοπισμό θέσης (location-enabled handsets). Βέβαια για να ανοίξει ο δρόμος για την διάθεση τέτοιων τεχνολογιών θα πρέπει η ασύρματη βιομηχανία να προχωρήσει στην τυποποίηση τους.

Βέβαια και αν ακόμη υπήρχαν τα κατάλληλα συστήματα εντοπισμού, η υπάρχουσα δικτυακή τεχνολογία δεν επιτρέπει την ανάπτυξη «εξειλιγμένων» υπηρεσιών αφού δεν υποστηρίζει πλήρως την μεταγωγή πακέτων και υψηλούς ρυθμούς μεταφοράς δεδομένων. Αυτό βέβαια αναμένεται να αλλάξει με τον ερχομό των δικτύων 3<sup>ης</sup> γενιάς (IMT-2000, CDMA2000), ο οποίος όμως έχει καθυστερήσει ήδη αρχικά. Επίσης θα πρέπει να υπάρχει και η κατάλληλη υποδομή για υποστήριξη περιαγωγής (roaming), χρέωσης (charging/billing) και «ανοιχτές» διεπαφές που να επιτρέπουν την αλληλεπίδραση μεταξύ των δικτύων και αυτών που αναπτύσσουν τις υπηρεσίες.

Ένα άλλο θέμα είναι οι κινητές συσκευές. Αυτές, εκτός από τεχνικές εντοπισμού θέσης, θα πρέπει να παρέχουν την δυνατότητα για ποιοτική και προσιτή πρόσβαση στις υπηρεσίες. Αυτό συνεπάγεται μεγάλες και έγχρωμες οθόνες, εύχρηστη σχεδίαση και φυσικά μεγάλη διάρκεια ενεργειακής αυτονομίας. Οι συσκευές αυτές επίσης δεν θα πρέπει να έχουν πολύ μεγάλο κόστος για τον τελικό χρήστη, ώστε να διεισδύσουν γρήγορα και να προωθήσουν την χρήση των υπηρεσιών θέσης. Ας μην ξεχνάμε ότι είναι πολύ σημαντικό για κάθε νέα τεχνολογία ή υπηρεσία να μπορέσει να χρησιμοποιηθεί γρήγορα ώστε να αποσβεσθεί η επένδυση που έγινε γι' αυτήν.



Ο συνδετικός κρίκος σε ένα σύστημα υπηρεσιών θέσης είναι αναμφισβήτητα η πλατφόρμα πάνω στην οποία αυτές θα αναπτυχθούν και θα εκτελούνται. Η πλατφόρμα αυτή θα πρέπει να είναι ένα ευέλικτο προϊόν λογισμικού το οποίο θα επιτρέπει την παροχή υπηρεσιών οι οποίες να πληρούν τα κύρια κριτήρια των χρηστών: έγκαιρη και έγκυρη παροχή που να λαμβάνει υπόψη στοιχεία προσωποποίησης (personalisation). Αυτό αποτελεί ενεργό ερευνητικό τομέα της σχετικής βιομηχανίας που έχει αποδώσει αρκετές λύσεις μέχρι τώρα (βλ. ενότητα 2.4). Βέβαια αυτή η προσπάθεια πρέπει να έχει πάντα σαν σκοπό την δημιουργία ευέλικτων πλατφόρμων οι οποίες να στηρίζονται σε δοκιμασμένα πρότυπα και πρωτόκολλα και οι οποίες να καλύπτουν πλήρως τις απαιτήσεις των εμπλεκόμενων φορέων/εταίρων.

Εκτός από τα τεχνικά προβλήματα που αντιμετωπίζουν οι LBS υπηρεσίες, υπάρχουν και άλλα ζητήματα που έχουν να κάνουν με το περιεχόμενο που παρέχεται από αυτές, τις επιχειρηματικές πολιτικές των εμπλεκόμενων «παικτών» και την στάση των χρηστών. Καταρχήν σήμερα, παρόλο που υπάρχει διαθέσιμο περιεχόμενο που να μπορεί να χρησιμοποιηθεί σε τέτοιες υπηρεσίες, δεν είναι κατάλληλα οργανωμένο για τέτοια χρήση. Επίσης αυτό θα πρέπει να είναι και ακριβές. Πολλές εταιρίες έχουν υποσχεθεί την ενημέρωση βάσεων δεδομένων σε πραγματικό χρόνο και με ακριβή στοιχεία (χάρτες, διευθύνσεις, κυκλοφορική κίνηση, καιρός, πωλήσεις προϊόντων), αλλά οι υποσχέσεις αυτές έχουν υλοποιηθεί σε πολύ μικρό βαθμό. Πρέπει να γίνει από όλους αντιληπτό ότι η πληροφορία θέσης από μόνη της δεν είναι ικανή να δώσει την επιθυμητή αξία στις LBS υπηρεσίες. Όσο πολυτιμότερο γίνεται το περιεχόμενο τόσο μεγαλύτερη αξία θα αποκτούν και οι υπηρεσίες θέσης.

Οι διάφοροι «παίκτες» θα πρέπει να λάβουν κάποια μέτρα για να αντιμετωπίσουν τα εμπόδια που παρουσιάζονται και να είναι ανταγωνιστικοί [33]. Καταρχήν, η επιδίωξη της διαφοροποίησης μεταξύ τους είναι πολύ σημαντική επειδή όλοι οι παροχείς υπηρεσιών θα έχουν στην διάθεσή τους σχεδόν το ίδιο περιεχόμενο και άρα δεν θα τους διαφοροποιεί αυτό. Επιπρόσθετα, κάθε παροχέας υπηρεσιών θα πρέπει να αναλύσει την αγορά στην οποία απευθύνεται έτσι ώστε να βρει τις κατάλληλες υπηρεσίες και εφαρμογές που θα μπορέσουν να χρησιμοποιηθούν από το αγοραστικό κοινό. Η κατάλληλη επιλογή τους είναι αυτή που θα αυξήσει τα έσοδα του και θα συντελέσει στην ανάπτυξη όλης της σχετικής βιομηχανίας. Επίσης οι διάφοροι φορείς θα πρέπει να μελετήσουν και να κατανοήσουν τα νέα επιχειρηματικά μοντέλα που παρουσιάζονται μαζί με τις υπηρεσίες θέσης. Η κατάλληλη πολιτική χρεώσεων και συνεργασίας μεταξύ των φορέων είναι αυτή που θα προωθήσει τις LBS υπηρεσίες.

Ένα από τα πιο σημαντικά θέματα, το οποίο έχει ήδη θιχτεί, είναι η προστασία της ιδιωτικότητας και της πληροφορίας θέσης του χρήστη. Οι πιο πολλοί συνδρομητές είναι επιφυλακτικοί απέναντι στις υπηρεσίες αυτές γιατί δεν ξέρουν αν η θέση τους μπορεί να χρησιμοποιηθεί και για δόλιους σκοπούς. Φυσικά αυτός ο προβληματισμός είναι απόλυτα δικαιολογημένος. Οι εταιρίες θα πρέπει να εξοικειώσουν τους χρήστες με τις νέες αυτές υπηρεσίες σταδιακά. Οι πρώτες υπηρεσίες θα πρέπει να δίνουν την αίσθηση στον χρήστη ότι η θέση του δεν θα υποκλαπεί ή θα χρησιμοποιηθεί καταχρηστικά. Έτσι θα πρέπει να αποφευχθούν οι αυτόματες προωθήσεις (κυρίως διαφημιστικές) χωρίς την αίτηση του χρήστη (spam). Κατόπιν όσο θα εξαπλώνονται οι υπηρεσίες θέσης τόσο θα νοιώθουν πιο άνετοι οι χρήστες και θα υποχωρούν οι φόβοι τους. Ας μην ξεχνάμε ότι πάντα κάθε νέα τεχνολογία αντιμετωπίζεται διστακτικά στην αρχή της χρήσης της (πχ barcode).

Όπως ισχύει και με όλα τα θέματα όμως, έτσι και στις υπηρεσίες θέσης οι χρήστες-πελάτες είναι αυτοί που θα αποφασίσουν αν τελικά οι υπηρεσίες αυτές είναι μια πραγματική ή πλασματική ανάγκη τους. Φυσικά ακόμη και αν είναι πλασματική ανάγκη, μπορεί να επιτευχθεί αρκετά μεγάλη διείσδυση με τη βοήθεια ευνοϊκής χρέωσης. Κι αυτό διότι οι χρήστες εκτός από την αξία της ίδιας της υπηρεσίας θα εκτιμήσουν και το χαμηλό κόστος της.

## 1.5.2 Προβλέψεις για το Μέλλον

Οι μεγάλοι οργανισμοί που ασχολούνται με τις LBS υπηρεσίες (OGC [47], LIF) έχουν ήδη θέσει τα μελλοντικά τους σχέδια. Στόχος σχεδόν όλων είναι ο καθορισμός κοινά αποδεκτών προτύπων που θα υλοποιηθούν από την βιομηχανία. Οι τηλεπικοινωνιακοί φορείς προετοιμάζουν το πεδίο για τον ερχομό των νέων υπηρεσιών. Ακόμα και η νέα τεχνολογία MMS συνδράμει στην εξοικείωση των χρηστών με τις νέες υπηρεσίες δεδομένων που θα επέλθουν στο σύντομο μέλλον.

Παρόλη όμως την προετοιμασία των εμπλεκόμενων φορέων για την παροχή υπηρεσιών θέσης, οι σύγχρονες προβλέψεις για το μέλλον είναι πιο απαισιόδοξες από τις αρχικές προβλέψεις, όσον αφορά στα έσοδα που αυτές θα φέρουν στους παροχείς τους. Για παράδειγμα μια αναφορά της εταιρίας στατιστικών ερευνών In-stat/MDR [48] προβλέπει ότι οι υπηρεσίες αυτές θα κάνουν περίπου μια δεκαετία να αποφέρουν κέρδη στους τηλεπικοινωνιακούς φορείς. Αυτό σημαίνει ότι η διείσδυσή τους θα είναι αργή, κάτι το οποίο, λόγω των προβλημάτων που θίχτηκαν προηγούμενα, είναι δικαιολογημένο και αναμενόμενο. Αυτή η αργή αποδοχή των υπηρεσιών προκαλείται και από το γενικότερο οικονομικό κλίμα που επικρατεί γύρω από τις νέες τεχνολογίες. Στην αμερικανική αγορά τηλεπικοινωνιών η κατάσταση είναι πιο καλή από τεχνικής πλευράς και υποδομής, αλλά το γεγονός ότι ο FCC «εξανάγκασε» τους φορείς να υποστηρίξουν εντοπισμό και διαχείριση θέσης, τους έχει φέρει σε δύσκολη οικονομική θέση. Στην Ευρώπη τέτοιες πρωτοβουλίες παρόμοιες με του FCC αναμένονται να προταθούν και να αρχίσουν να υλοποιούνται μέσα στην επόμενη πενταετία (οι κανονισμοί αυτοί φέρουν την ονομασία E112) [49].

Αυτή η διστακτικότητα των χρηστών-πελατών, όμως, δεν προβλέπεται να σταματήσει και να αποτραβήξει τους διάφορους «παίκτες» από την ανάπτυξη τέτοιων υπηρεσιών. Ο λόγος είναι ότι οι υπηρεσίες θέσης θεωρούνται ως η καλύτερη ευκαιρία για αύξηση του εισοδήματός τους, ακόμα και να αυτό αργήσει να γίνει. Έτσι στο προσεχές μέλλον αναμένεται να παρέχονται όλο και πιο εξελιγμένες υπηρεσίες που θα προσπαθούν να ικανοποιήσουν τις ανάγκες των χρηστών. Γενικά πάντως, οι υπηρεσίες αυτές είναι μάλλον οι μόνες που θα μπορούσαν να χαρακτηριστούν ως «killer apps» (δηλ. «επαναστατικές» εφαρμογές), γιατί είναι οι μόνες που εκμεταλλεύονται πλήρως το μεγαλύτερο πλεονέκτημα των ασύρματων επικοινωνιών: την κινητικότητα.

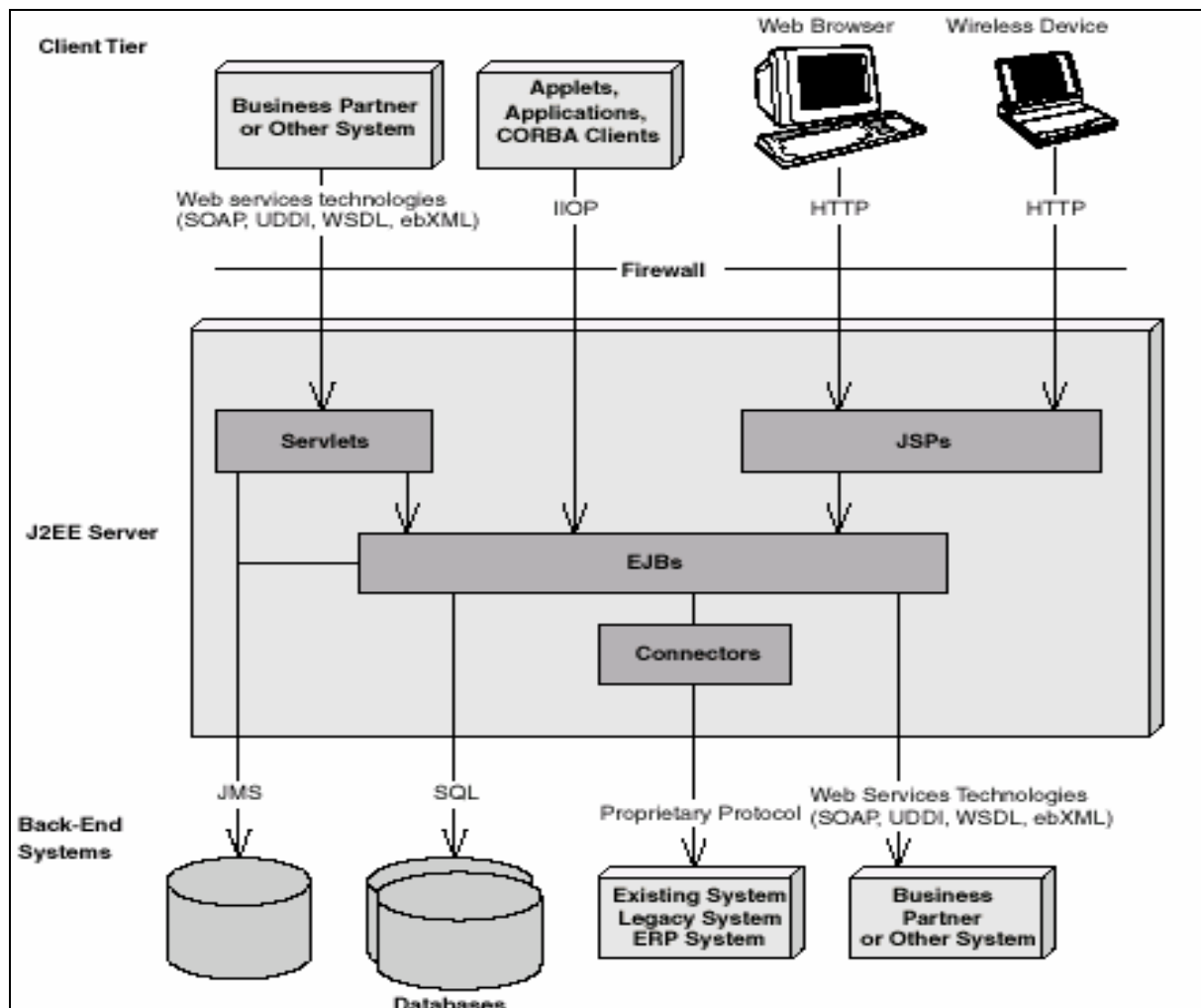
## **Μέρος 2**

### **Παραδειγματικός Χρονοδρομολογητής**

## 2.1 Τεχνολογίες που χρησιμοποιήθηκαν

Πριν την παρουσίαση του παραδειγματικού χρονοδρομολογητή (Scheduler) θα προηγηθεί μια σύντομη αναφορά των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξή του. Αυτό κρίνεται σκόπιμο για την καλύτερη κατανόηση της υλοποίησής του. Η λειτουργικότητά του και η αρχιτεκτονική του παρουσιάζονται αναλυτικά μετά από αυτήν την εισαγωγική ενότητα των τεχνολογιών

Η υλοποίηση του παραδειγματικού χρονοδρομολογητή έγινε στη γλώσσα προγραμματισμού Java και πιο συγκεκριμένα στη πλατφόρμα ανάπτυξης εφαρμογών J2EE της Sun Microsystems. Η γλώσσα Java [50] είναι η πιο διαδεδομένη αντικειμενοστραφής γλώσσα προγραμματισμού και παρέχει μεγάλα πλεονεκτήματα όπως ανεξαρτησία από πλατφόρμα εκτέλεσης και έτοιμες βιβλιοθήκες για όλα σχεδόν τα πεδία εφαρμογών. Η πλατφόρμα J2EE απευθύνεται σε επιχειρηματικές εφαρμογές πολλών στρωμάτων (multitier enterprise applications) και περιλαμβάνει ένα σύνολο από τεχνολογίες λογισμικού (εικόνα 47).



Εικόνα 47: Η γενική αρχιτεκτονική μιας εφαρμογής J2EE

Μεταξύ αυτών των τεχνολογιών είναι οι:

**Enterprise JavaBeans (EJB):** Είναι συστατικά (components) που λειτουργούν στο πλαίσιο ενός Component Container το οποίο υποστηρίζει, μεταξύ άλλων, δοσοληψίες (transactions), ασφάλεια (security), επεκτασιμότητα (scalability)

**Java Servlets:** Είναι κλάσεις που επεκτείνουν την λειτουργικότητα ενός Web server δίνοντας την δυνατότητα για δυναμική δημιουργία περιεχομένου (dynamic content).

**JavaServer Pages (JSP):** Είναι μια επέκταση των servlets που συνδυάζει markup tags και Java scripts αποτελώντας έτσι μια πιο φιλική τεχνολογία για τους σχεδιαστές διαδικτυακών τόπων (web designers) από τα servlets που απευθύνονται κυρίως σε προγραμματιστές.

**Java Naming and Directory Interface (JNDI):** Είναι ένα περιβάλλον ονομάτων (naming environment) από το οποίο μπορούν να τροποποιηθούν παραμετρικά (δηλαδή χωρίς αλλαγές στον κώδικά τους) τα διάφορα αντικείμενα που έχουν πρόσβαση σε αυτό.

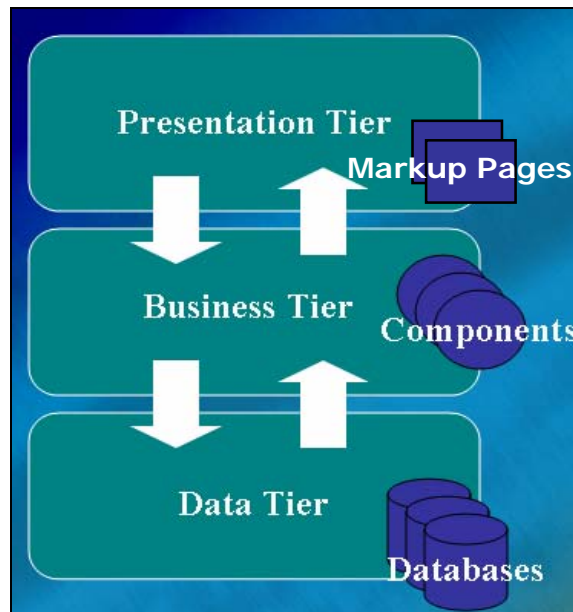
**JDBC:** Είναι μια τεχνολογία που παρέχει πλήρη αλληλεπίδραση μεταξύ των Java αντικειμένων και μιας βάσης δεδομένων. Η αλληλεπίδραση αυτή μάλιστα είναι ανεξάρτητη από τον κατασκευαστή της βάσης.

**Java Transaction (JTA):** Αυτή η τεχνολογία επιτρέπει στις εφαρμογές να πραγματοποιούν δοσοληψίες με τρόπο ανεξάρτητο από συγκεκριμένες υλοποιήσεις.

**Java Message Service (JMS):** Η υπηρεσία αυτή παρέχει την δυνατότητα στις εφαρμογές να αποκτούν πρόσβαση σε συστήματα μηνυμάτων διαφόρων ειδών.

## 2.1.1 Εξυπηρετές Εφαρμογών (Application Servers)

Η πλατφόρμα J2EE απαιτεί την εκτέλεση των εφαρμογών μέσα στο πλαίσιο ειδικά κατασκευασμένων εξυπηρετών που λέγονται Application Servers. Αν χωρίζαμε μια επιχειρηματική εφαρμογή σε τρία στρώματα : στρώμα παρουσίασης (presentation tier), στρώμα επιχειρηματικής λογικής (business tier ή middle tier) και στρώμα δεδομένων (data tier) τότε οι Application Servers ανήκουν στο middle tier και γι' αυτό αναφέρονται συχνά και ως middleware (εικόνες 47 και 48).



**Εικόνα 48: Τα αρχιτεκτονική τριών στρωμάτων (3-tier)**

Τα κυριότερα λειτουργικά χαρακτηριστικά που θα πρέπει να διαθέτουν οι J2EE Application Servers είναι τα παρακάτω:

- Web/JSP/Servlet Container και EJB Container
- Μηχανισμοί ασφάλειας (authentication, authorization)
- Μηχανισμοί επικοινωνίας με άλλα συστήματα (όπως ουρές μηνυμάτων, RDBMSs, συστήματα δοσοληψιών και legacy συστήματα)
- Μηχανισμοί ανάκαμψης (failover)
- Μηχανισμοί υποστήριξης ποιότητας υπηρεσιών (caching, διαχείριση μνήμης, instance/connection pooling)
- Μηχανισμοί διαχείρισης (αντικείμενα διαχείρισης, ημερολόγιο)

## 2.1.2 Enterprise JavaBeans (EJB)

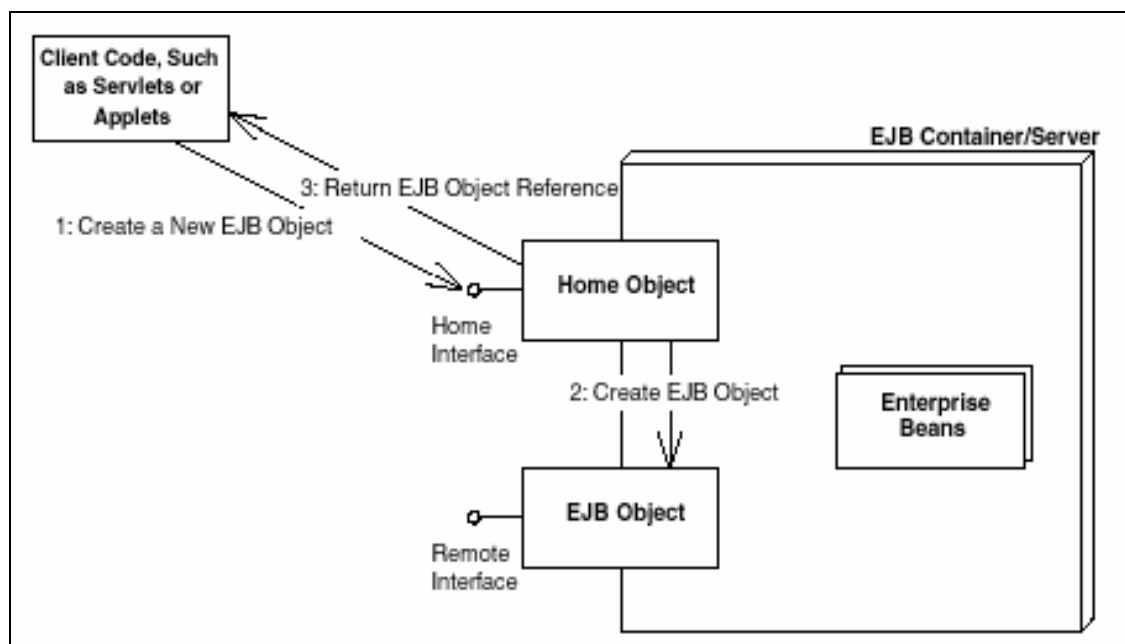
### 2.1.2.1 Γενικά στοιχεία

Η αρχιτεκτονική EJB ορίζει ένα μοντέλο κατανεμημένων αντικειμένων (components) που χρησιμοποιούνται για την ανάπτυξη ασφαλών και επεκτάσιμων εφαρμογών που υποστηρίζουν δοσοληψίες και μεγάλο πλήθος ταυτόχρονων χρηστών. Τα τρία σημαντικότερα πλεονεκτήματα αυτής της αρχιτεκτονικής είναι τα εξής:

- 1) Έχει γίνει αποδεκτή σαν πρότυπο (standard) από την βιομηχανία λογισμικού.

- 2) Είναι ένα ανοικτό πρότυπο διαθέσιμο σε όλους. Έτσι υπάρχει πλήρης συμβατότητα ανάμεσα σε όσους τηρούν τις προδιαγραφές. Για παράδειγμα μπορούμε να εκτελέσουμε τα αντικείμενά μας με όποιον Application Server θέλουμε, αρκεί αυτός να είναι συμβατός με τις προδιαγραφές J2EE.
- 3) Γρηγορότερη ανάπτυξη των εφαρμογών. Τα EJBs περιέχουν μόνο την λογική της εφαρμογής, αφήνοντας τον Application Server να ασχοληθεί με τις υπόλοιπες λεπτομέρειες(ασφάλεια, δοσοληψίες, caching κλπ).

Ένα αντικείμενο EJB αποτελείται από δύο τουλάχιστον Java διεπαφές (home και remote interfaces), από την κλάση που υλοποιεί το Bean και από ένα αρχείο XML (deployment descriptor). Η εικόνα 49 δείχνει τα παραπάνω στοιχεία και πώς αυτά αλληλεπιδρούν μεταξύ τους.



**Εικόνα 49: Η αλληλεπίδραση μεταξύ των διεπαφών του EJB και ενός εξυπηρετούμενου**

### Enterprise Bean

Είναι η κλάση της Java που υλοποιεί τις λειτουργίες (μεθόδους) τις οποίες θα εκτελεί το αντικείμενο EJB.

### Home interface

Κάθε enterprise bean που δημιουργείται πρέπει να έχει ένα σχετιζόμενο με αυτό Home interface το οποίο χρησιμοποιείται από τους εξυπηρετούμενους (clients) για να βρουν ένα ήδη υπάρχον στιγμιότυπο (instance) του EJB ή να δημιουργήσουν ένα νέο.

### Remote interface

Η διεπαφή αυτή ουσιαστικά παρουσιάζει στον έξω κόσμο τις μεθόδους που υλοποιούνται στην κλάση Enterprise Bean. Παίζει τον ίδιο ρόλο με την διεπαφή CORBA IDL.

*Σημείωση:* Πολλές φορές χρησιμοποιούνται οι διεπαφές LocalHome και Local αντί για την Home και την Remote, για καλύτερη απόδοση. Οι local διεπαφές έχουν τον ίδιο ρόλο με τις remote αλλά μπορούν να χρησιμοποιηθούν μόνο από εξυπηρετούμενους που βρίσκονται στο ίδιο Container με το κληθέν EJB (πχ άλλα EJBs).

### **Deployment descriptor**

Είναι ένα αρχείο XML που περιέχει πληροφορίες για το αντικείμενο EJB, στο οποίο μπορούμε να καθορίσουμε κάποιες λεπτομέρειες του αντικειμένου παραμετρικά (χωρίς να επέμβουμε στον κώδικά του). Τα χαρακτηριστικά που μπορούμε να δηλώσουμε είναι:

- Τα ονόματα των δύο διεπαφών, Home και Remote.
- Το όνομα της Home διεπαφής στο δέντρο JNDI.
- Διαδικασίες δοσοληγίας (transaction) για κάθε μέθοδο του αντικειμένου EJB.

### **EJB-Jar file**

Είναι ένα αρχείο με επέκταση *.jar*, που περιέχει την κλάση που υλοποιεί το Bean, τις δύο διεπαφές, home και remote (ή τις αντίστοιχες local), καθώς και το αρχείο XML (deployment descriptor).

## **2.1.2.2 Οι τύποι των EJBs**

Στις προδιαγραφές που έχει δημοσιεύσει η Sun Microsystems για το μοντέλο EJB 2.0 ορίζονται τρεις τύποι αντικειμένων EJB:

### **Session Beans**

Μοντελοποιούν τις διαδικασίες μιας επιχείρησης. Αυτές οι διαδικασίες μπορεί να είναι από αλγόριθμοι υπολογισμών μέχρι κλήσεις σε άλλα πληροφοριακά συστήματα. Για παράδειγμα μπορούν να κάνουν διακρίβωση στοιχείων πελατών, δημιουργία τιμολογίων κ.α. Παριστάνουν πράξεις πάνω σε αποθηκευμένα δεδομένα αλλά όχι τα ίδια τα δεδομένα. Τα session beans διαχωρίζονται σε *stateless* και *stateful*.

Τα *stateful* χρησιμοποιούνται όταν το αντικείμενο πρέπει να «θυμάται» κάποιες πληροφορίες σχετικές με τον πελάτη που το χρησιμοποιεί, για όση ώρα το χρησιμοποιεί. Έτσι ένα αντικείμενο που μοντελοποιεί την αγορά προϊόντων με την χρήση καλαθιού αγορών, θα υλοποιούταν ως *stateful session bean* αφού πρέπει να «θυμάται» τι έχει επιλέξει για αγορά ο χρήστης.

Τα *stateless* είναι απλούστερα. Δεν πραγματοποιούν καμία διαδικασία «διαλόγου» με τον πελάτη και έτσι είναι εύκολα επαναχρησιμοποιούμενα από τον server.

### **Entity Beans**

Μοντελοποιούν τα δεδομένα μιας επιχείρησης. Αναπαριστούν πληροφορίες που υπάρχουν καταχωρημένες σε βάσεις δεδομένων. Για την ακρίβεια κάθε μεταβλητή τους αντιστοιχεί σε μια στήλη ενός πίνακα σχεσιακής βάσης δεδομένων. Για παράδειγμα, ένα entity bean μπορεί να αναπαριστά ένα τιμολόγιο, μια παραγγελία, ένα προϊόν ή ένα



κατάλογο πελατών. Υπάρχουν δύο κατηγοριών entity beans: τα *Container Managed Persistence* (CMP) και τα *Bean Managed Persistence* (BMP).

Για τα CMP, ο Application Server είναι υπεύθυνος για την μόνιμη αποθήκευση των δεδομένων στην βάση (persistence) και ο προγραμματιστής δεν ασχολείται καθόλου με αυτήν την διαδικασία. Οι σχετικές πληροφορίες βρίσκονται στο αρχείο deployment descriptor. Τα CMP μειώνουν το χρόνο ανάπτυξης ενός EJB όπως επίσης και το μέγεθος του απαιτούμενου κώδικα που πρέπει να γραφεί από τον προγραμματιστή.

Για τα BMP όμως, ο προγραμματιστής πρέπει να γράψει τον κώδικα που θα αποθηκεύει (ή ανακτά) τα δεδομένα του αντικειμένου στη βάση. Αυτό απαιτεί το γράψιμο JDBC κώδικα για την επικοινωνία με τη βάση. Ο προγραμματιστής βέβαια σε αυτή την περίπτωση έχει πλήρη έλεγχο για το πώς λειτουργεί το Bean.

### **Message-driven Beans**

Αυτά τα αντικείμενα μοιάζουν με τα session beans ως προς το ότι και αυτά αντιπροσωπεύουν διαδικασίες. Η διαφορά έγκειται στο ότι μπορούν να κληθούν μόνο μέσω μηνυμάτων (δηλαδή ασύγχρονα). Αντίθετα, τα session beans καλούνται σαν απομακρυσμένα αντικείμενα μέσω του RMI-ΠΙΟΡ API (είναι μια επέκταση του RMI). Όποια διεργασία σε μια επιχείρηση μπορεί να αναβληθεί για κάποιο διάστημα ή δεν απαιτείται να επιστρέψει κάποιο αποτέλεσμα άμεσα, μπορεί να υλοποιηθεί με message-driven beans.

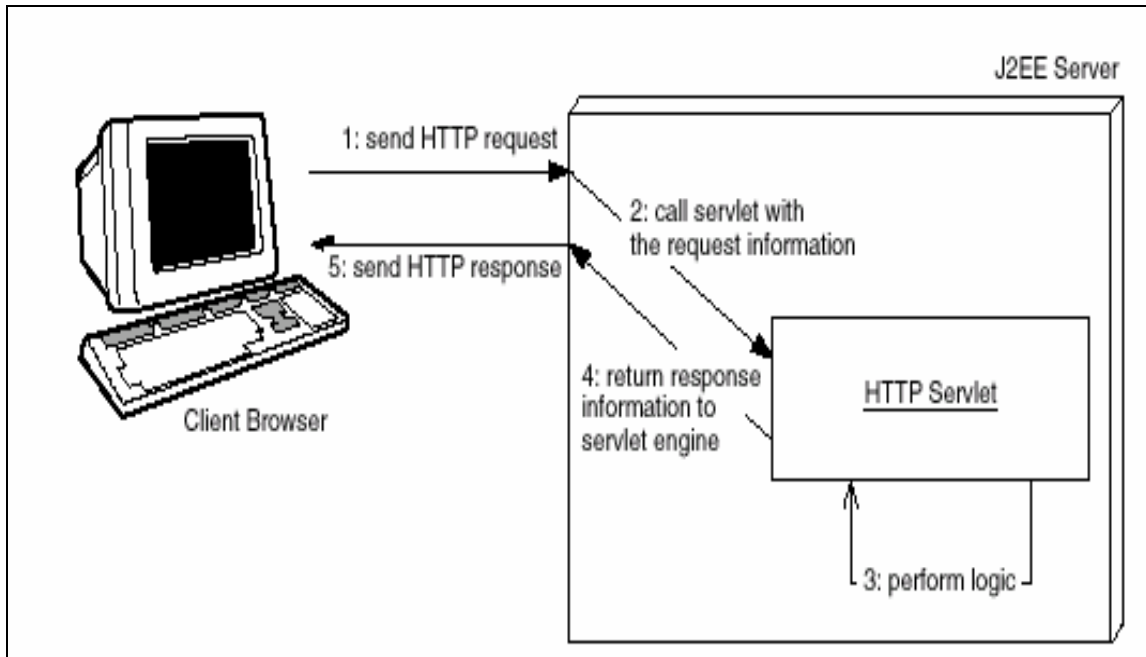
Στην υλοποίηση του παραδειγματικού χρονοδρομολογητή χρησιμοποιήθηκαν μόνο session και entity beans.

## **2.1.3 HTTP Servlets**

Τα HTTP Servlets [51] είναι μονάδες κώδικα (modules) που εκτελούνται σε ένα εξυπηρετή (server) ο οποίος υποστηρίζει HTTP κλήσεις (request/response). Ο εξυπηρετής αυτός ακολουθεί τις προδιαγραφές J2EE και αναφέρεται συνήθως σαν *servlet engine*. Μία κλήση request προέρχεται από ένα απομακρυσμένο εξυπηρετούμενο (client) και περιέχει δεδομένα τα οποία στέλνει στον server. Μία κλήση response είναι τα δεδομένα που επιστρέφει ο server στον client, από τον οποίο δέχτηκε την request.

Ένα servlet είναι ουσιαστικά ένα αντικείμενο της Java το οποίο δέχεται σαν είσοδο μία κλήση request ,αναλύει τα δεδομένα ,εκτελεί ορισμένες διαδικασίες και επιστρέφει τα αποτελέσματα πίσω στο client με μια κλήση response. Αποτελεί τον ενδιάμεσο μεταξύ του client και των βάσεων δεδομένων που βρίσκονται στην πλευρά του server.

Η παρακάτω εικόνα δείχνει την λειτουργία ενός servlet.



**Εικόνα 50: Οι κλήσεις μεταξύ ενός HTTP servlet και ενός εξυπηρετούμενου**

Η όλη διαδικασία θυμίζει την λειτουργία των CGI scripts με την διαφορά όμως ότι τα τελευταία πρέπει να επανεκκινούνται για κάθε κλήση request ενώ τα servlets επαναχρησιμοποιούνται συνεχώς για να εξυπηρετήσουν πολλές κλήσεις. Αυτό αυξάνει σημαντικά την απόδοση του συστήματος γιατί ο server δεν είναι αναγκασμένος να επανεκκινεί μία νέα διεργασία κάθε φορά που λαμβάνει μια κλήση request. Πλεονεκτούν έναντι των CGI scripts και στον τομέα της ασφάλειας αλλά και στο ότι μπορούν να αναπτυχθούν πολύ πιο εύκολα, κάνοντας την όλη διαδικασία να κοστίζει πολύ λιγότερο. Ένα ακόμη πλεονέκτημα τους είναι ότι υλοποιούνται σε Java και έτσι μπορούν να εκμεταλλευτούν όλα τα πλεονεκτήματα της Java. Υπάρχει όμως και το μειονέκτημα ότι απαιτούν γνώσεις Java, κάτι που δεν ικανοποιεί τους σχεδιαστές διαδικτυακών τόπων (web designers).

Ας σημειωθεί ότι υπάρχουν πολλοί τύποι servlets επειδή είναι ανεξάρτητα πρωτοκόλλων. Π.χ. θεωρητικά μπορείς να δημιουργήσεις κανείς ένα FTP servlet, ένα POP servlet, και ούτω καθεξής.

Στην υλοποίηση του παραδειγματικού χρονοδρομολογητή ο RequestHandler() είναι αυτός που παίζει τον ρόλο του client και εκτελεί μία κλήση request προς το GServlet.

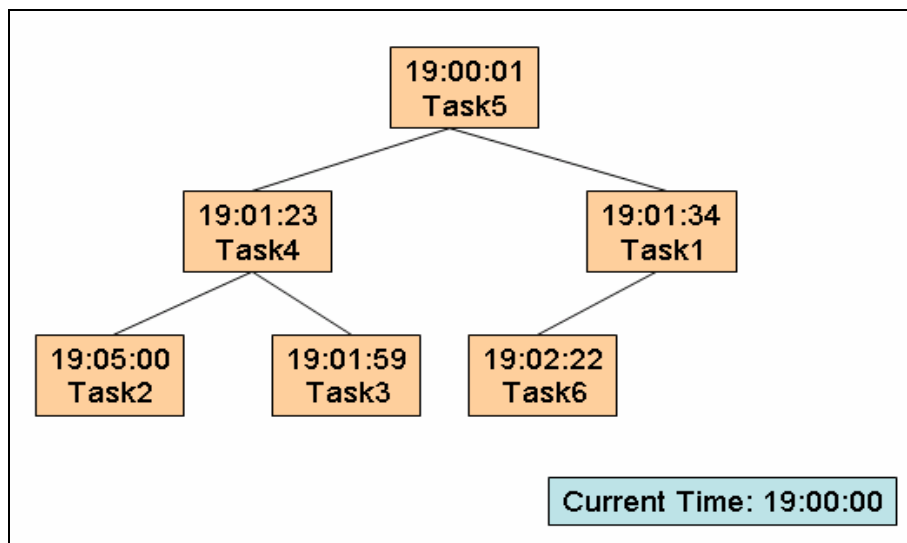
## 2.1.4 Η κλάση Timer της Java

Η κλάση *java.util.Timer* ουσιαστικά ορίζει έναν χρονοδρομολογητή (scheduler) ο οποίος είναι υπεύθυνος για την εκτέλεση των διεργασιών (tasks) που του έχουν ανατεθεί. Τα tasks αυτά πρέπει να είναι αντικείμενα της κλάσης *java.util.TimerTask* και μπορούν να εκτελεστούν μία μόνο φορά ή περιοδικά. Κάθε αντικείμενο Timer είναι ένα νήμα

εκτέλεσης που εκτελείται στο παρασκήνιο (background thread). Αυτό αναλαμβάνει να εκτελέσει όλα τα tasks που του έχουν ανατεθεί σειριακά.

Σύμφωνα με την τεκμηρίωση της Sun για την πλατφόρμα J2SE, μπορούν να ανατεθούν χιλιάδες tasks σε κάθε Timer [52]. Βέβαια για να μπορεί να πραγματοποιηθεί αυτό και να εκτελούνται όλα τα tasks στην ώρα τους, πρέπει να αποτελούνται από κώδικα που εκτελείται γρήγορα. Δηλαδή είναι καλό να αποφεύγονται tasks που απαιτούν μεγάλη υπολογιστική ισχύ. Αν κάποιο task χρειαστεί αρκετό χρόνο για να ολοκληρωθεί, τα επόμενα tasks θα καθυστερήσουν να εκτελεστούν και όταν τελικά φτάσει η σειρά τους, θα εκτελεστούν πολύ γρήγορα το ένα μετά το άλλο (αυτό είναι ιδιαίτερα φανερό και ενοχλητικό όταν ο Timer χρησιμοποιείται για αναπαραγωγή κινούμενης εικόνας). Γενικά η κλάση αυτή δεν εγγυάται ότι τα προς εκτέλεση tasks θα εκτελεστούν στην ώρα που προγραμματίστηκαν.

Όσον αφορά την υλοποίηση του Timer, χρησιμοποιεί ένα ισοζυγισμένο δυαδικό σωρό για να αναπαραστήσει την ουρά αναμονής των tasks. Ένας σωρός είναι ένα δέντρο που στον κόμβο-ρίζα έχει το πιο μικρό στοιχείο (ή το πιο μεγάλο, ανάλογα με το συγκεκριμένο πρόβλημα που καλείται να λύσει) και όλοι οι κόμβοι-παιδιά ενός κόμβου  $K$  έχουν μεγαλύτερη τιμή (ή αντίστοιχα μικρότερη) από τον  $K$ . Στην περίπτωση του Timer, στην ρίζα του σωρού τοποθετείται κάθε φορά το task του οποίου ο χρόνος επόμενης εκτέλεσης απέχει λιγότερο από την τρέχουσα χρονική στιγμή. Ένας τέτοιος σωρός απεικονίζεται στο παρακάτω σχήμα:



Εικόνα 51: Παράδειγμα σωρού που χρησιμοποιείται στην κλάση Timer

Ο λόγος που επιλέχθηκε ένας τέτοιος τύπος δεδομένων για την αναπαράσταση αυτής της ουράς αναμονής είναι ότι ο χρόνος που απαιτείται για να βρεθεί το επόμενο task προς εκτέλεση είναι σταθερός και ότι η χρονική πολυπλοκότητα του αλγορίθμου εισαγωγής ενός νέου task είναι  $O(\log_2 n)$ , όπου  $n$  ο αριθμός των tasks που έχουν ήδη ανατεθεί στον Timer.

Η εκτέλεση ενός task από τον χρονοδρομολογητή μπορεί να γίνει με δύο τρόπους: είτε με εκτέλεση σταθερής καθυστέρησης (fixed-delay execution) είτε με εκτέλεση σταθερού ρυθμού (fixed-rate execution). Στην πρώτη περίπτωση, ο Timer

φροντίζει έτσι ώστε κάθε εκτέλεση ενός περιοδικού task να απέχει από την προηγούμενη τέτοιο χρόνο που να προσεγγίζει όσο περισσότερο γίνεται την περίοδο. Αν μια εκτέλεση καθυστερήσει για κάποιο λόγο, οι επόμενες εκτελέσεις του ίδιου task θα καθυστερήσουν επίσης. Έτσι μακροπρόθεσμα η συχνότητα των εκτελέσεων θα είναι ελαφρώς μικρότερη από το αντίστροφο της επιθυμητής και προκαθοριζόμενης περιόδου. Στην δεύτερη περίπτωση, ο Timer φροντίζει έτσι ώστε τα tasks να εκτελούνται με όσο το δυνατό μικρότερη απόκλιση από την προκαθορισμένη χρονική στιγμή, η οποία όμως αυτή την φορά δεν εξαρτάται από την προηγούμενη εκτέλεση αλλά από την πρώτη εκτέλεση. Αυτός ο τρόπος εκτέλεσης αναφέρεται σε tasks που πρέπει να εκτελούνται σε απόλυτες χρονικές στιγμές (πχ στις 5 το απόγευμα κάθε ημέρα) και όχι σε σχετικές (πχ κάθε μισή ώρα).

Στον παραδειγματικό χρονοδρομολογητή χρησιμοποιείται ο πρώτος τρόπος εκτέλεσης, αφού συνήθως οι υπηρεσίες τις οποίες μελετάμε απαιτούν μια «ομαλότητα» (smoothness) στην εκτέλεσή τους. Η ομαλότητα αυτή είναι απαραίτητη για να μην αντιλαμβάνεται ο τελικός χρήστης παρενέργειες που μπορεί να προκύψουν από τον αυξημένο φόρτο στο σύστημα στο οποίο εκτελείται ο χρονοδρομολογητής. Τέλος, ολοκληρώνοντας την μελέτη του Timer, πρέπει να αναφέρουμε ότι η κλάση TimerTask είναι απλώς μια υλοποίηση (implementation) του interface Runnable και άρα είναι ένα νήμα εκτέλεσης στο οποίο εκτελείται η μέθοδος run(). Η μέθοδος αυτή περιέχει τον κώδικα που θα εκτελείται περιοδικά.

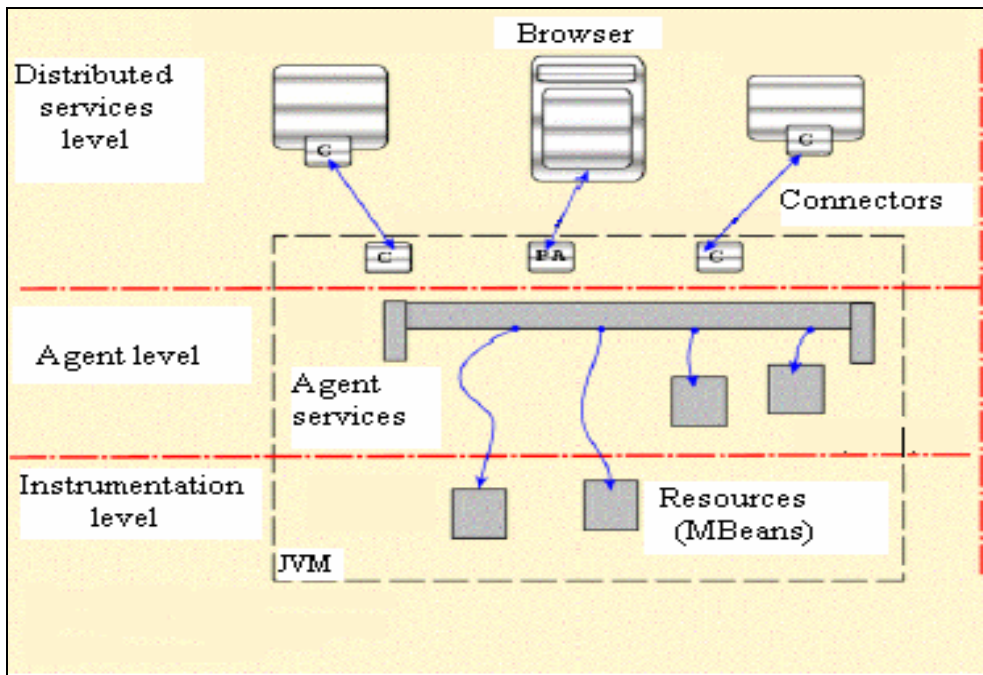
## 2.1.5 Διαχείριση με JMX

### 2.1.5.1 JMX (Java Management Extensions)

Είναι μια τεχνολογία που παρέχει διευκολύνσεις για διαχείριση και παρακολούθηση στοιχείων υλικού και λογισμικού με την βοήθεια της Java [53]. Επίσης συνεργάζεται με έναν μεγάλο αριθμό ήδη υπαρχόντων διαχειριστικών προτύπων. Η παρακάτω εικόνα παρουσιάζει την τριεπίπεδη JMX-αρχιτεκτονική καθώς και τις σχέσεις που υπάρχουν.

Τα τρία επίπεδα είναι

- *Instrumentation level*: Οι προς διαχείριση πόροι. Περιέχει τα MBeans (standard, dynamic, open, και model) και τις κλάσεις μεταδεδομένων (MBean metadata classes)
- *Agent level*: Οι πράκτορες (agents) που ελέγχουν τα αντικείμενα/πόρους του προηγούμενου επιπέδου. Αυτό το επίπεδο περιέχει τον εξυπηρετή MBean (MBean server) και τις υπηρεσίες πράκτορα (Agent services)
- *Distributed services level*: Οι καταναεμημένες υπηρεσίες (distributed services), ο μηχανισμός με τον οποίο οι εφαρμογές (browsers) αλληλεπιδρούν με τους πράκτορες.



Εικόνα 52: Τα τρία επίπεδα της αρχιτεκτονικής JMX

### 2.1.5.2 MBeans (Managed Beans)

Ένα MBean είναι ένα αντικείμενο Java που υλοποιεί μία διεπαφή (management interface) και ακολουθεί ορισμένα σχεδιαστικά πρότυπα. Παρουσιάζει/εκθέτει (exposes) όλες τις απαραίτητες πληροφορίες και τις λειτουργίες που χρειάζεται μία εφαρμογή διαχείρισης (management application) για να ελέγχει κάποιους πόρους (resources). Η διεπαφή που υλοποιείται αποτελείται από τα παρακάτω στοιχεία:

- Τα ιδιοχαρακτηριστικά (attributes)
- Τις λειτουργίες (operations)
- Τις ειδοποιήσεις (notifications)
- Τους constructors για την κλάση MBean

#### Ιδιοχαρακτηριστικά

Είναι ένα πολύ ζωτικό κομμάτι της διεπαφής του MBean. Τα ιδιοχαρακτηριστικά περιγράφουν τους προς διαχείριση πόρους. Π.χ. αν ο προς διαχείριση πόρος είναι ένας εκτυπωτής τότε τα ιδιοχαρακτηριστικά θα περιγράφουν τα χαρακτηριστικά του εκτυπωτή (είδος εκτύπωσης, κατάσταση, ποσότητα διαθέσιμου μελανιού, έλλειψη χαρτιού κα). Ορίζονται με τις λέξεις κλειδιά get και set που προηγούνται του ονόματος δίνοντας δικαιώματα ανάγνωσης και γραφής αντίστοιχα.

#### Λειτουργίες

Μπορούν να περιγραφούν σαν ενέργειες που εκτελούνται πάνω στους προς διαχείριση πόρους. Είναι μέθοδοι που μπορούν να έχουν πολλαπλές παραμέτρους και να επιστρέφουν προαιρετικά μία τιμή.

### Ειδοποιήσεις

Παρέχουν στα MBeans δυνατότητες επικοινωνίας και αποστολή πληροφοριών. Υλοποιούν την `javax.management.NotificationBroadcaster` διεπαφή.

### Constructors

Τα MBeans μπορούν να φορτωθούν δυναμικά μέσα στους JMX πράκτορες. Αυτό επιτυγχάνεται χρησιμοποιώντας τους constructors που συμπεριλαμβάνονται στον ορισμό της διεπαφής και ορίζει ο καθένας μια διαφορετική συμπεριφορά για το αντικείμενο MBean .

Η προδιαγραφή JMX ορίζει τέσσερις τύπους MBeans για να υποστηρίξει διάφορες ανάγκες:

- **Standard MBeans:** Είναι ο πιο κοινός τύπος. Χρησιμοποιούν έναν απλό τρόπο ονοματολογίας και ορίζουν μία «στατική» διεπαφή διαχείρισης.
- **Dynamic MBeans:** Υλοποιούν την `javax.management.DynamicMBean` διεπαφή και εκθέτουν την διεπαφή διαχείρισης τους κατά την εκτέλεση για καλύτερη απόδοση και γιατί οι προς διαχείριση πόροι δεν είναι γνωστοί από πριν.
- **Open MBeans:** Είναι ουσιαστικά μια προέκταση των Dynamic MBeans.
- **Model MBeans:** Είναι επίσης μια προέκταση των Dynamic MBeans. Υλοποιούν την `javax.management.modelmbean.ModelMBean` διεπαφή. Τα Model MBeans απλοποιούν την διαχείριση των πόρων παρέχοντας μία απλή και κατανοητή προς τον χρήστη συμπεριφορά [54].

## 2.2 Αρχιτεκτονική

Ο παραδειγματικός Scheduler είναι ένα κομμάτι λογισμικού (module) το οποίο αποτελεί κεντρικό στοιχείο για την διαδικασία της αυτόματης προώθησης σε μια πλατφόρμα που υποστηρίζει αυτό το μοντέλο. Μπορεί να εκτελεί διεργασίες (tasks) που του έχουν ανατεθεί σε τακτά χρονικά διαστήματα ή σε προκαθορισμένες χρονικές στιγμές. Αυτό επιτυγχάνεται με την βοήθεια της κλάσης PolosTimer που ουσιαστικά υλοποιεί την κλάση java.util.Timer (για την ακρίβεια αποτελεί μια wrapper class αυτής). Οι διεργασίες που του ανατίθενται αντιπροσωπεύουν τις υπηρεσίες (LBS). Τον Scheduler τον υλοποιεί το session bean SchedulerBean.

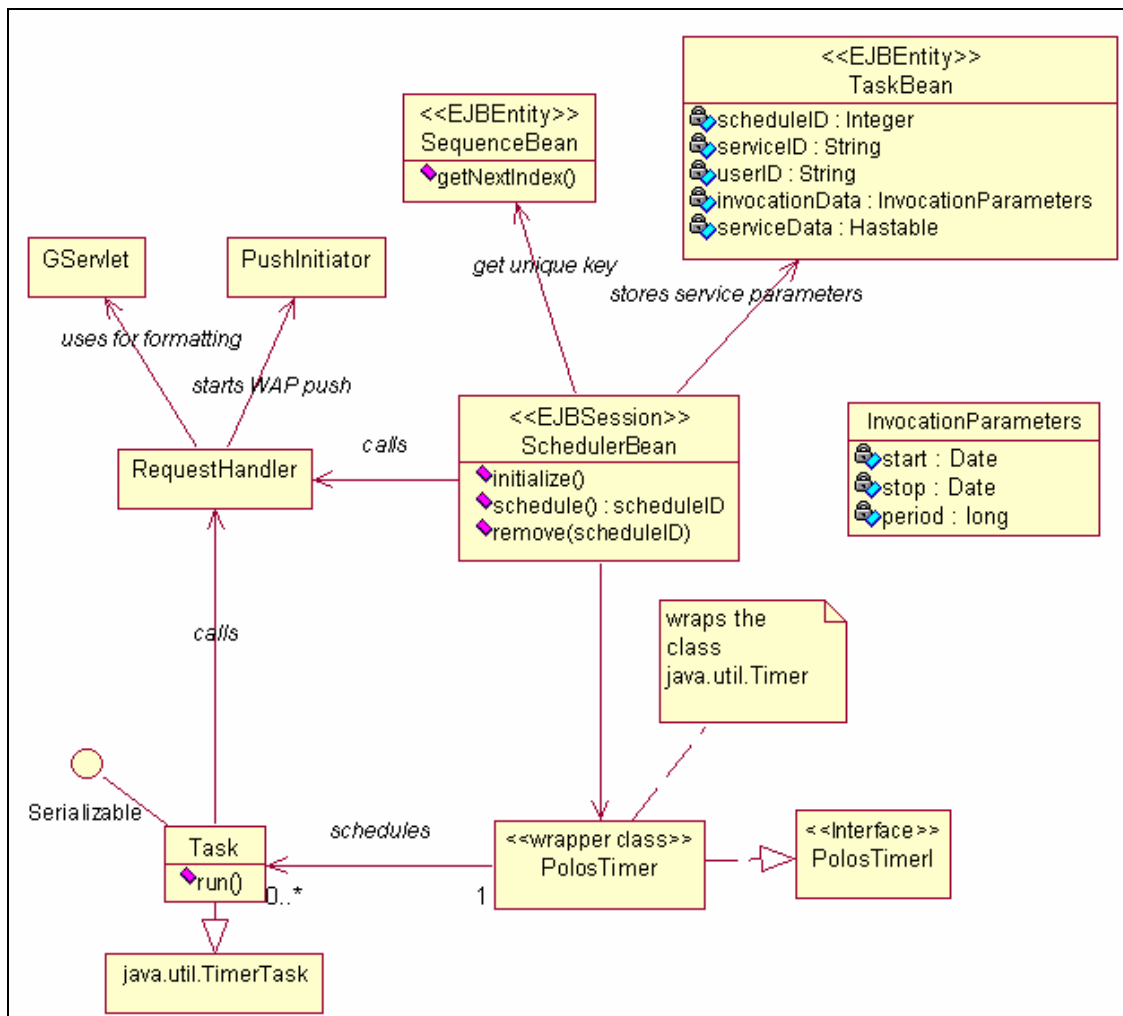
Ο Scheduler είναι υπεύθυνος για την χρονοδρομολόγηση αλλά και την αποθήκευση (persistence) των διεργασιών, για την περίπτωση κατάρρευσης του συστήματος. Για να αποθηκευθούν οι διεργασίες (tasks) χρησιμοποιείται το entity bean TaskBean, που τα χαρακτηριστικά του (attributes) είναι οι παράμετροι της κάθε διεργασίας, δηλαδή η ημερομηνία εκκίνησης και τερματισμού, η περίοδος εκτέλεσης, η ταυτότητα (ID) του χρήστη και της υπηρεσίας, και οι παράμετροι κλήσης της υπηρεσίας. Το πρωτεύον κλειδί για την αποθήκευση αποκτάται από το SequenceBean.

Κατά την εκτέλεση μιας διεργασίας (task) καλείται ο RequestHandler, που ουσιαστικά παίζει τον ρόλο ενός υποθετικού (dummy) χρήστη, με ορίσματα την υπηρεσία και τα δεδομένα που καθορίζουν τις παραμέτρους της αποστολής στον τελικό χρήστη. Αφού του επιστραφούν τα τελικά μορφοποιημένα δεδομένα της υπηρεσίας από το GServlet τότε προχωρεί στη προώθηση (push) αυτών των δεδομένων. Το GServlet, το οποίο δεν παρουσιάζεται εδώ, είναι ένα servlet που αναλαμβάνει να μορφοποιήσει το περιεχόμενο της υπηρεσίας ανάλογα με το τι διεπαφή θα χρησιμοποιηθεί (WAP ή SMS).

Βέβαια υπάρχει περίπτωση ο RequestHandler να κληθεί απευθείας από το SchedulerBean. Αυτή είναι η περίπτωση που δεν χρειάζεται χρονοδρομολόγηση αφού η αυτόματη προώθηση σκανδαλίστηκε από την θέση του χρήστη (positioning).

Παρακάτω παρουσιάζονται τα διαγράμματα κλάσεων (class), ακολουθίας (sequence) και εξαρτημάτων (component).

## 2.2.1 Διάγραμμα Κλάσεων (Class Diagram)



Εικόνα 53: Το διάγραμμα κλάσεων του Scheduler

Η μόνη άγνωστη οντότητα εδώ είναι η κλάση PushInitiator. Αυτή είναι μια κλάση που αναλαμβάνει την διαδικασία της αυτόματης προώθησης.

## 2.2.2 Ακολουθιακά Διαγράμματα (Sequence Diagrams)

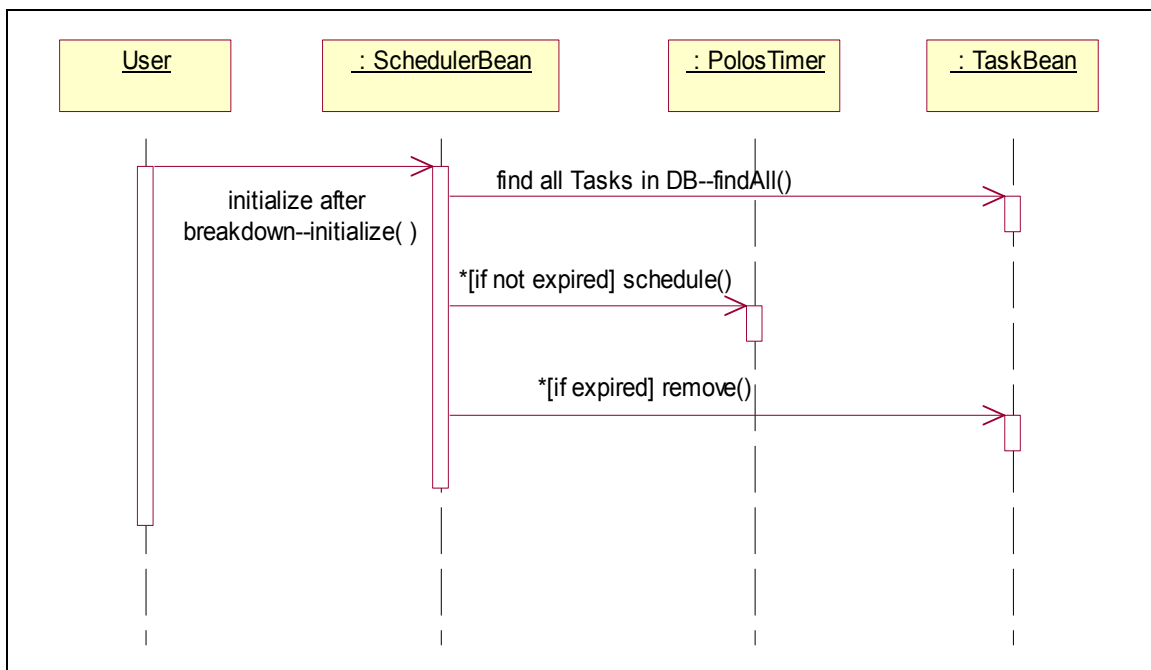
Στα διαγράμματα αυτά παρουσιάζονται οι μέθοδοι που υλοποιούνται στο SchedulerBean και μπορούν να κληθούν από χρήστες καθώς και η διαδικασία κατά την οποία μια διεργασία (task) στέλνει τα δεδομένα της υπηρεσίας στον RequestHandler, ακυρώνεται ή τίθεται σε αυτήν νέος χρόνος εκτέλεσης (Timer Function).



Οι μέθοδοι του SchedulerBean είναι οι initialize(), schedule() και remove(). Η schedule() εμφανίζεται με δύο τρόπους: είτε αναθέτει (schedules) ένα task που χρησιμοποιεί αυτόματη προώθηση με χρονικό σκανδαλισμό είτε αναθέτει κάποιο που προέκυψε από χωρικό σκανδαλισμό(positioning)

### 2.2.2.1 Η μέθοδος initialize

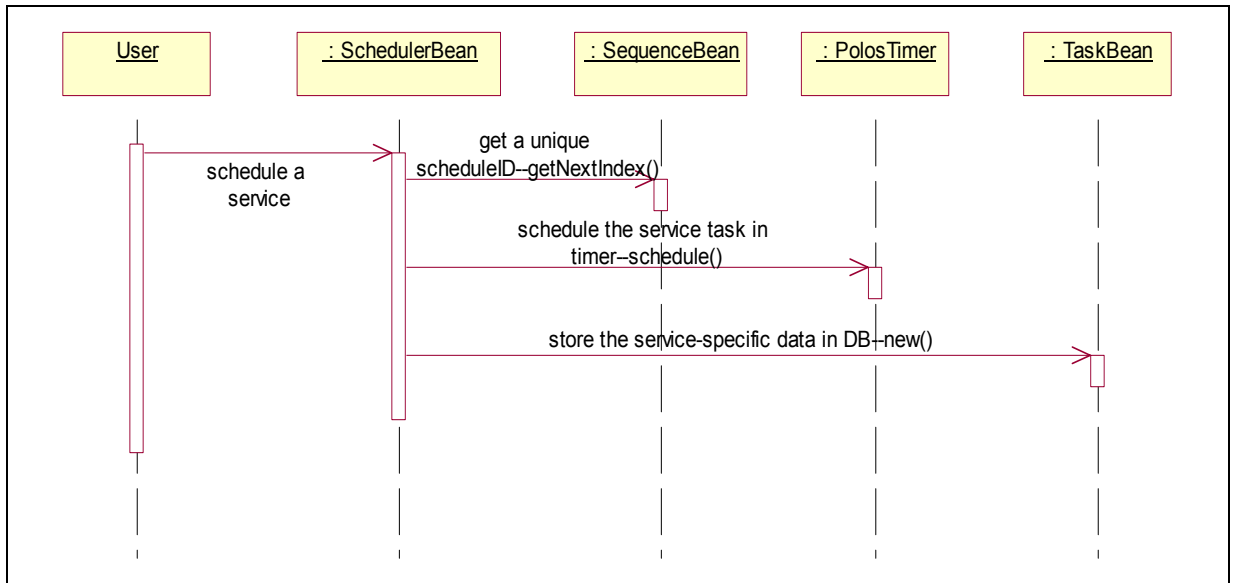
Η μέθοδος αυτή έχει σαν σκοπό την αρχικοποίηση του Scheduler και άρα την επαναχρονοδομολόγηση όλων των αποθηκευμένων διεργασιών στον PolosTimer. Χρησιμοποιείται συνήθως μετά από την κατάρρευση του J2EE server στον οποίο εκτελείται ο Scheduler. Διαβάζει την εγγραφή κάθε διεργασίας από το entity TaskBean και αν η διεργασία δεν έχει λήξει τότε την χρονοδρομολογεί αλλιώς την διαγράφει.



Εικόνα 54: Το ακολουθιακό διάγραμμα της διαδικασίας αρχικοποίησης (initialize)

### 2.2.2.2 Η μέθοδος schedule (χρονικός σκανδαλισμός)

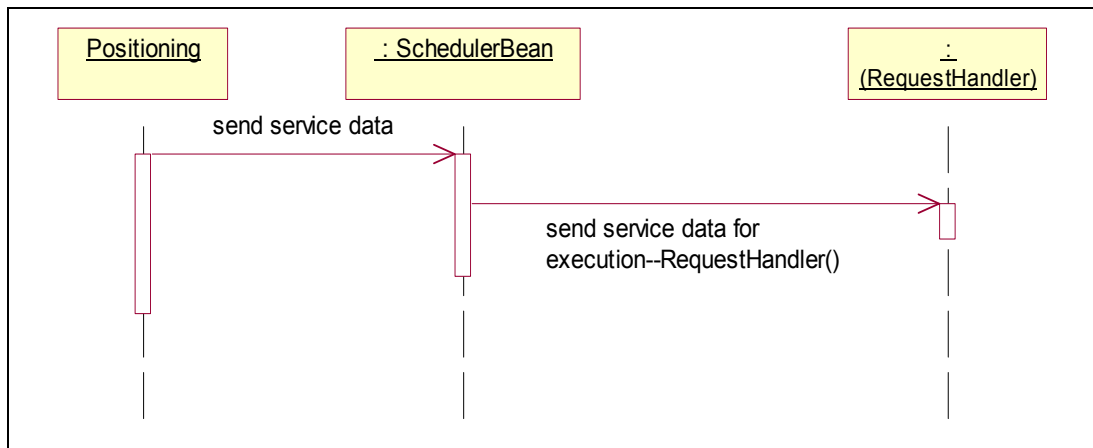
Αυτή η μέθοδος καλείται όταν μία διεργασία πρέπει να χρονοδρομολογηθεί για να εκτελείται περιοδικά. Όταν κληθεί ανακτά ένα μοναδικό πρωτεύον κλειδί, για την αποθήκευση της διεργασίας, από το SequenceBean, την χρονοδρομολογεί στον PolosTimer και τελικά την αποθηκεύει στη βάση (TaskBean).



**Εικόνα 55:** Το ακολουθιακό διάγραμμα της ανάθεσης (schedule) μιας διεργασίας που σκανδαλίζεται χρονικά

### 2.2.2.3 Η μέθοδος *schedule* (χωρικός σκανδαλισμός)

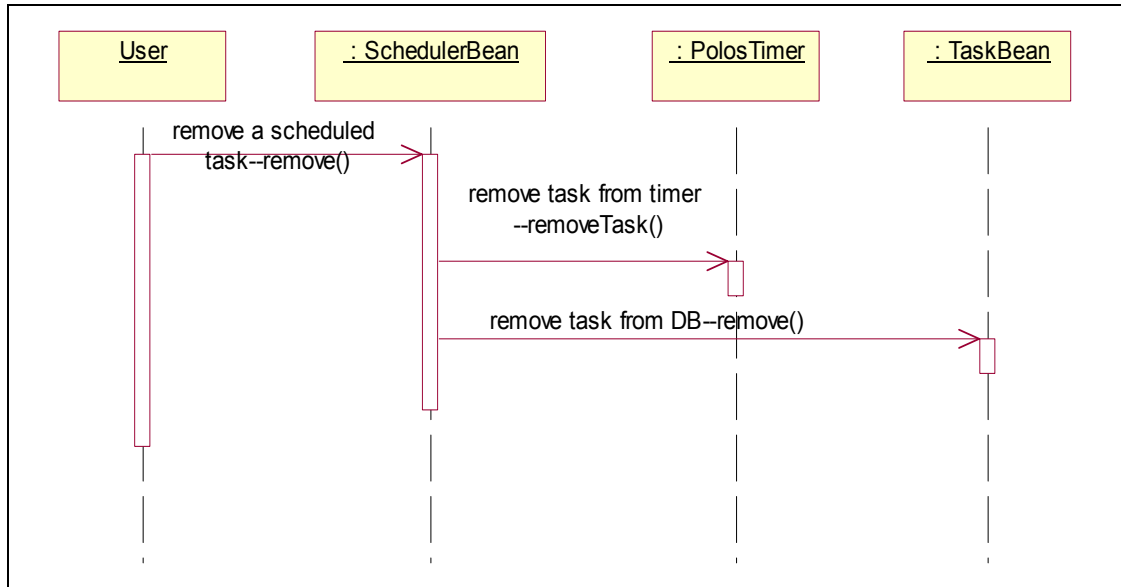
Στην περίπτωση αυτή η διεργασία είναι τέτοιου είδους που δεν χρειάζεται να χρονοδρομολογηθεί αλλά να εκτελεστεί αμέσως. Τότε στέλνονται τα δεδομένα της αιτούμενης υπηρεσίας κατευθείαν στον RequestHandler.



**Εικόνα 56:** Το ακολουθιακό διάγραμμα της ανάθεσης (schedule) μιας διεργασίας που σκανδαλίζεται χωρικά

#### 2.2.2.4 Η μέθοδος *remove*

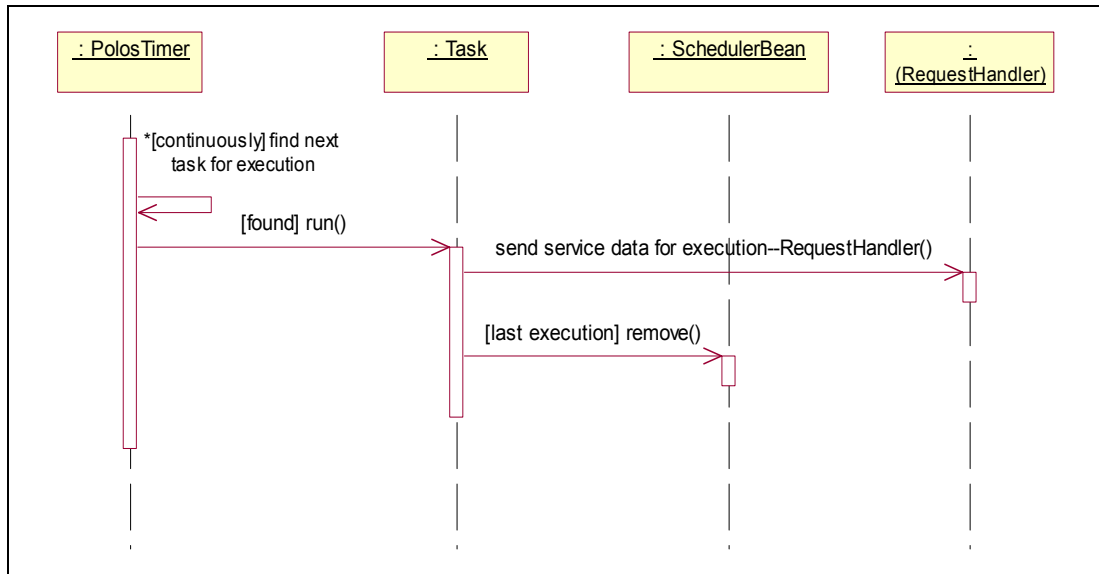
Η μέθοδος αυτή καλείται όταν μία ήδη χρονοδρομολογημένη διεργασία πρέπει να διαγραφεί από τον PolosTimer και την βάση και να μην εκτελεστεί ξανά.



Εικόνα 57: Το ακολουθιακό διάγραμμα της διαγραφής (*remove*) μιας διεργασίας από τον Scheduler

#### 2.2.2.5 Η χρονοδρομολόγηση μιας διεργασίας

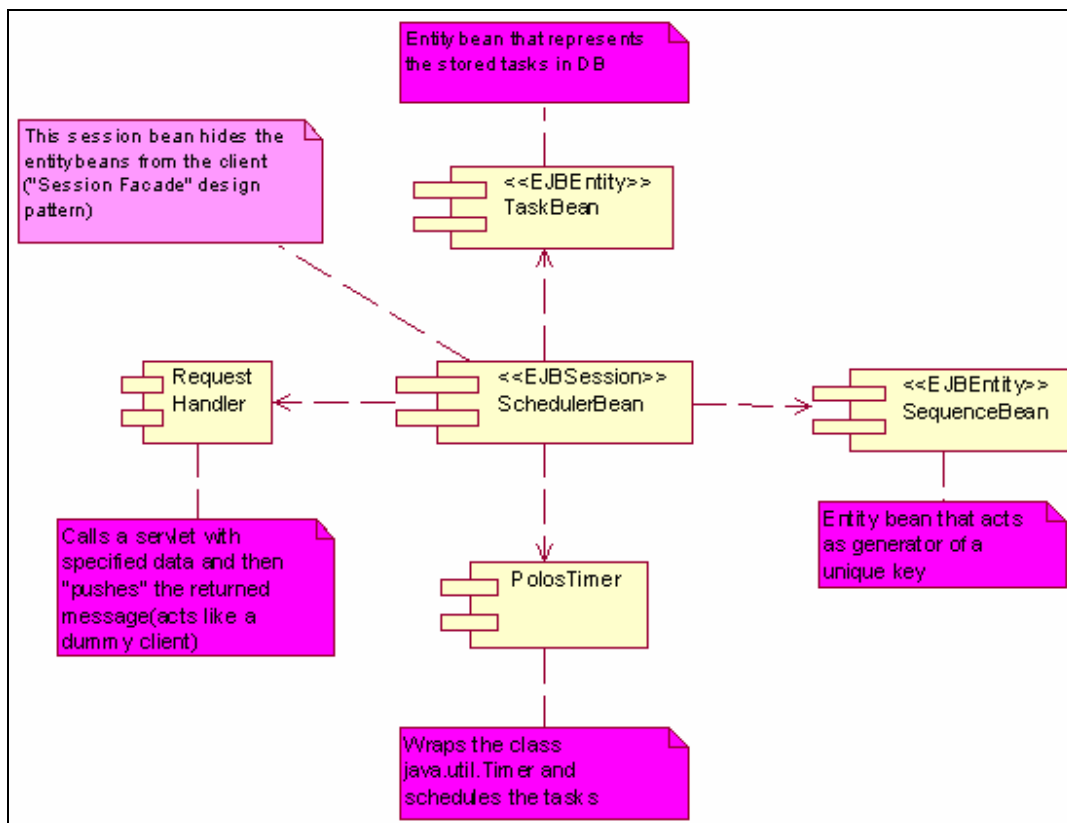
Εδώ εμφανίζεται η ακολουθία των βημάτων κατά την εκτέλεση μιας χρονοδρομολογημένης διεργασίας. Αρχικά καλείται ο RequestHandler, παίρνοντας ως ορίσματα τις παραμέτρους της συγκεκριμένης υπηρεσίας. Στη συνέχεια ελέγχεται από την κλάση Task αν έχει λήξει η προκαθορισμένη διάρκεια παροχής της υπηρεσίας αυτής. Σ' αυτήν την περίπτωση καλείται η μέθοδος *remove()* του SchedulerBean και η αντίστοιχη διεργασία διαγράφεται τόσο από τον PolosTimer όσο και από την βάση δεδομένων (TaskBean).



**Εικόνα 58:** Το ακολουθιακό διάγραμμα της εκτέλεσης μιας διεργασίας από τον Scheduler

### 2.2.3 Διάγραμμα συστατικών (Component Diagram)

Στο διάγραμμα της επόμενης σελίδας (εικόνα 59) παρουσιάζονται τα βασικά συστατικά (components) της εφαρμογής. Τα συστατικά αυτά μπορεί να βρίσκονται όλα μέσα στον ίδιο application server ή σε διαφορετικά μέρη. Για παράδειγμα ο PolosTimer μπορεί να βρίσκεται έξω από τον server στον οποίο βρίσκονται τα EJBs και να προσπελάζεται μέσω RMI κλήσεων.



**Εικόνα 59: Το διάγραμμα των κύριων συστατικών του παραδειγματικού χρονοδρομολογητή**

# Αναφορές - Πηγές

## Αναφορές

- [1]. Ivana Podnar, Manfred Hauswirth and Mehdi Jazayeri, "Mobile Push: Delivering Content to Mobile Users", Proc. IEEE 22<sup>nd</sup> International Conference on Distributed Computing Systems Workshops, 2002
- [2]. "Ericsson's mobile location solution", Göran Swedberg, Ericsson Review No. 04, 1999.  
[http://www.ericsson.com/about/publications/review/1999\\_04/files/19990406.pdf](http://www.ericsson.com/about/publications/review/1999_04/files/19990406.pdf)
- [3]. FCC Enhanced 911 <http://www.fcc.gov/911/enhanced/>
- [4]. P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On Indoor Position Location With Wireless LANs," The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2002), Lisbon, Portugal, September 2002
- [5]. P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System" *Proc. IEEE INFOCOM 2000*, Vol. 2, pp. 775-784, March 2000.  
[www.ekahau.com](http://www.ekahau.com)
- [6]. GSM documents : 03.71, 04.35, 04.71, 09.31
- [7]. 3GPP documents: TS25.305/331/215/133/171
- [8]. Γεωγραφικά Συστήματα Πληροφοριών και Ανάλυση Χώρου, Κωστής Κουτσόπουλος, Εκδόσεις Παπασωτηρίου 2002
- [9]. Seattle GIS WEB, [www.ci.seattle.wa.us/gis](http://www.ci.seattle.wa.us/gis)
- [10]. GML (Geography Markup Language), <http://opengis.net/gml/>
- [11]. POIX (Point-of-interest exchange language) – <http://www.w3.org/TR/poix/>
- [12]. NVML (Navigation markup language) – <http://www.w3.org/TR/NVML>
- [13]. SVG (Scalable Vector Graphics) - <http://www.w3.org/TR/SVG/>
- [14]. VML (Vector Markup Language) - <http://www.w3.org/TR/NOTE-VML>
- [15]. Xalan - <http://xml.apache.org/xalan/overview.html>
- [16]. Saxon - <http://users.iclway.co.uk/mhkay/saxon/>
- [17]. VoiceXML – [www.voicexml.org](http://www.voicexml.org)
- [18]. Oracle Spatial – [otn.oracle.com/products/spatial/content.html](http://otn.oracle.com/products/spatial/content.html)
- [19]. Oracle Locator Datasheet – [otn.oracle.com/products/spatial/htdocs/data\\_sheet\\_9i/9iR2\\_locator\\_ds.html](http://otn.oracle.com/products/spatial/htdocs/data_sheet_9i/9iR2_locator_ds.html)
- [20]. SMS Tutorial, *Web ProForum Tutorials* by International Engineering Consortium  
<http://www.iec.org>
- [21]. SMPP, <http://smsforum.net/>
- [22]. WAP, [www.wapforum.org](http://www.wapforum.org)
- [23]. Ορισμός και Ιστορία του WAP, [www.askforwap.gr](http://www.askforwap.gr)
- [24]. WAP Push Architectural Overview,  
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-250-PushArchOverview-20010703-a.pdf>

- [25]. Push Access Protocol , <http://www1.wapforum.org/tech/terms.asp?doc=WAP-247-PAP-20010429-a.pdf>
- [26]. *Comparison of WAP Push and Short Message Service (SMS)* by Fergus Wills  
Openwave Systems Inc. <http://www.openwave.com>
- [27]. Jupitermedia Corporation
- [28]. Location Inter-Operability Forum (LIF) <http://www.openmobilealliance.org/lif>
- [29]. Open GIS Consortium (OGC), [www.opengis.com](http://www.opengis.com)
- [30]. Wireless Location Industry Association, [www.wliaonline.com](http://www.wliaonline.com)
- [31]. Open Mobile Alliance (OMA), [www.openmobilealliance.org](http://www.openmobilealliance.org)
- [32]. OpenLS, [www.opensls.org](http://www.opensls.org)
- [33]. Location Based Services: Considerations and Challenges, [www.northstream.se](http://www.northstream.se)
- [34]. *POLOS\_UoA\_WP12\_BM\_v01 Survey of Existing Business Models* , Fotini Hadjiefthymiades
- [35]. LBS BUSINESS MODELS, <http://www.davidhwilliams.com>
- [36]. IDC, [www.idc.com](http://www.idc.com)
- [37]. Parlay Group, [www.parlay.org](http://www.parlay.org)
- [38]. Cellular Telecommunicationw Industry Association (CTIA) [www.wow-com.com](http://www.wow-com.com)
- [39]. Online Privacy Alliance (OPA), [www.privacyalliance.org](http://www.privacyalliance.org)
- [40]. LocationNet Platform, <http://www.locationnet.com>
- [41]. Celebriety Platform, <http://www.ctmotion.com>
- [42]. LoL@ Platform, <http://www.ftw.at/wtt01/docs/ProjektC1.pdf>
- [43]. Cellpoint Platform, <http://www.cellpoint.com>
- [44]. IBM's WebSphere Everyplace Server, <http://www.ibm.com>
- [45]. Webraska SmartZone Geospatial Platform, <http://www.webraska.com>
- [46]. PoLoS Platform, <http://www.polos.org>
- [47]. Emerging Technology Summit Series - ETS I: Location-Based Services presented by [GITA](http://www.gita.org) and the Open GIS Consortium. October 24-25, 2002, <http://www.opensls.org/dvd1/ets1/Closing.pdf>
- [48]. In-stat/MDR, [www.instat.com](http://www.instat.com)
- [49]. GI News E112: 'Europe states its position', (September 2002), [http://www.ginews.co.uk/0902\\_18\(1\).html](http://www.ginews.co.uk/0902_18(1).html)
- [50]. Official Java Site, Sun Microsystems, <http://java.sun.com>
- [51]. Core Servlets and JSPs, Prentice Hall and Sun Microsystems, <http://www.coreservlets.com>
- [52]. Java Timer documentation, <http://java.sun.com/j2se/1.4.1/docs/api/java/util/Timer.html>
- [53]. *Managing Java Applications with JMX*, J. Steven Perry, O'Reilly
- [54]. *JBoss Administration and Development Second Edition*, Scott Stark and The JBoss Group

## **Άλλες Πηγές**

Εκτός από τα κείμενα που βρέθηκαν στις παραπάνω αναφορές, χρησιμοποιήθηκαν και τα παρακάτω:

Permanent Reference Document: SE.23, GSM Association,  
[www.gsmworld.com/technology/applications/location.shtml](http://www.gsmworld.com/technology/applications/location.shtml)

“Professional EJB”, Wrox Publications

“Mastering Enterprise JavaBeans 2<sup>nd</sup> Edition”, Ed Roman, Scott Ambler, Tyler Jewell,  
Wiley Computer Publishing (διαθέσιμο σε μορφή PDF στην διεύθυνση  
[www.theserverside.com](http://www.theserverside.com) )



## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

3GPP	3 <sup>rd</sup> Generation Partnership Project
A-GPS	Assisted GPS
ANSI	American National Standards Institute
AOA	Angle Of Arrival
API	Application Programming Interface
ARPU	Average Revenue Per User
ASP	Application Service Provider
BTS	Base Tranceiver Station
CGI	Cell Global Identity
CTIA	Cellular Telecommunicationw Industry Association
DBMS	DataBase Management System
EDGE	Enhanced Data-Rates for GSM Evolution
E-OTD	Enhanced OTD
ETSI	European Telecommunication
FCC	Federal Communications Commision
GIS	Geographic Information System
GML	Geography Markup Language
GMLC	Gateway Mobile Location Centre
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HLR	Home Location Register
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronic Enginners
IETF	Internet Enginering Task Force
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
LBS	Location Based Services
LCS	LoCation Service
LIF	Location Interoperability Forum
LMU	Location Measurement Unit
LS	Location Service
MLP	Mobile Location Protocol
MPC	Mobile Positioning Center
OGC	Open GIS Consortium
OMA	Open Mobile Alliance
OPA	Open Privacy Alliance
OSA	Open Service Access
OTD	Observed Time Difference
PDA	Personal Digital Assistant
PLMN	Public Land Mobile Network
RSS	Received Signal Strength

SMLC	Serving Mobile Location Centre
SMPP	Short Message Peer to Peer protocol
SMS	Short Message Services
SMSC	Short Message Service Centre
SVG	Scalable Vector Graphics
TA	Timing Advance
TDOA	Time Difference Of Arrival
UL-TOA	Uplink Time Of Arrival
UMTS	Universal Mobile Telecommunications System
VML	Vector Markup Language
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network
WLIA	Wireless Location Industry Association
XML	Extensible Markup Language
XSLT	XML Transformation Language

# Παράρτημα

## Ο κώδικας του παραδειγματικού χρονοδρομολογητή

Στο παράρτημα αυτό παρουσιάζεται ο κώδικας σε Java του παραδειγματικού Scheduler. Τα EJBs έχουν υλοποιηθεί με την βοήθεια του εργαλείου Xdoclet, το οποίο κάνει αυτόματη παραγωγή κώδικα με βάση συγκεκριμένα Javadoc tags, που πρέπει να γράψει κατάλληλα ο προγραμματιστής.

### Class InvocationParameters

```
package org.polos.kernel.scheduler;

import java.io.Serializable;
import java.util.Date;

public class InvocationParameters implements Serializable {

    private Date start;
    private Date stop;
    private long period;

    public InvocationParameters(Date start, Date stop, long period) {
        this.start=start;
        this.stop=stop;
        this.period=period;
    }

    public Date getStart() {
        return start;
    }

    public void setStart(Date start) {
        this.start=start;
    }

    public Date getStop() {
        return stop;
    }

    public void setStop(Date stop) {
```

```

        this.stop=stop;
    }

    public long getPeriod() {
        return period;
    }

    public void setPeriod(long period) {
        this.period=period;
    }
}

```

## Class PolosTimer

```
package org.polos.kernel.scheduler;
```

```

import java.util.Timer;
import java.util.TimerTask;
import java.util.Hashtable;
import java.util.Date;
import java.rmi.server.UnicastRemoteObject;

```

```
public class PolosTimer extends UnicastRemoteObject implements PolosTimerI{
```

```

    private Hashtable hash;
    private Timer timer;

```

```

    public PolosTimer() throws java.rmi.RemoteException {
        hash=new Hashtable();
        timer=new Timer();
    }

```

```

    public void schedule(Integer ID, Task task, Date firstTime, long period) {
        try {
            timer.schedule(task, firstTime, period);
            hash.put(ID, task);
        }
        catch(Exception e){e.printStackTrace();}
    }

```

```

    public void schedule(Integer ID, Task task, Date firstTime) {
        try {
            timer.schedule(task, firstTime);
            hash.put(ID, task);
        }
    }

```

```

    }
    catch(Exception e) {e.printStackTrace();}
}

public void removeTask(Integer ID) {
    Task task;
    try {
        task=(Task)hash.get(ID);
        if(task!=null)
            task.cancel();
        hash.remove(ID);
    }
    catch(Exception e) {e.printStackTrace();}
}
}

```

## Interface PolosTimerI

```
package org.polos.kernel.scheduler;
```

```
import java.rmi.Remote;
```

```
import java.rmi.RemoteException;
```

```
import java.util.Date;
```

```
public interface PolosTimerI extends Remote {
```

```
    public void schedule(Integer ID, Task task, Date firstTime, long period)
        throws RemoteException;
```

```
    public void schedule(Integer ID, Task task, Date firstTime)
        throws RemoteException;
```

```
    public void removeTask(Integer ID)
        throws RemoteException;
```

```
}
```

## Class RequestHandler

```
package org.polos.kernel.scheduler;
```

```
import org.polos.kernel.scheduler.SchedulerException;
```

```
import org.polos.ifs.pushEnvel.PushMessage;
```

```
import java.io.*;
```

```

import java.lang.*;
import java.text.*;
import java.net.*;
import java.util.*;
import javax.naming.*;
import org.apache.log4j.Logger;

public class RequestHandler {

    //TIP:if you want to use the following String as the URL uncomment the
    //appropriate line in the methods GETcall() and POSTcall()

    private static final String GSERVLET_URL="http://localhost:8080/ifshttp/XMLpack";
    private static final String USER_ADDRESS = "user-address";
    private static final String SECOND_RESPONSE_TYPE = "second-response-type";
    private static final String SERVICE_TYPE = "service-type";
    private static final String REQUEST_METHOD = "rqst-method";
    private static final String SERVICE_DATA = "service-data";
    private static final String USER_ID = "user-id";
    private static final String SERVICE_ID = "service-id";
    private static Logger logger=Logger.getLogger(RequestHandler.class.getName());

    public static void handleRequest (String userID, String serviceID, Hashtable serviceData) {
        String result=null;
        try {
            result = GETcall(userID, serviceID, serviceData);
            //result = POSTcall(userID, serviceID, serviceData);
        }
        catch(Exception e){
            logger.error("RequestHandler: Error in calling Gservlet:"+ e.getMessage());
            return;
        }

        //call the WAP Push interface
        try {
            PushMessage.PushData(userID, result, (String)serviceData.get(USER_ADDRESS));
        }
        catch(Exception e){
            logger.error("RequestHandler: Error in WAP Push function");
        }
    }

    private static String GETcall(String userID, String serviceID, Hashtable serviceData)
        throws Exception{

        //create the Query String

```

```

String qstring = new String("?");
String tmp=null;
qstring += SERVICE_ID+"="+serviceID+"&";
qstring += USER_ID+"="+userID+"&";
if((tmp=(String)serviceData.get(USER_ADDRESS))!=null)
    qstring += USER_ADDRESS+"="+tmp+"&";
else
    qstring += USER_ADDRESS+"&";
if((tmp=(String)serviceData.get(SECOND_RESPONSE_TYPE))!=null)
    qstring += SECOND_RESPONSE_TYPE+"="+tmp+"&";
else
    qstring += SECOND_RESPONSE_TYPE+"&";
if((tmp=(String)serviceData.get(SERVICE_TYPE))!=null)
    qstring += SERVICE_TYPE+"="+tmp+"&";
else
    qstring += SERVICE_TYPE+"&";
if((tmp=(String)serviceData.get(REQUEST_METHOD))!=null)
    qstring += REQUEST_METHOD+"="+tmp+"&";
else
    qstring += REQUEST_METHOD+"&";
if((tmp=(String)serviceData.get(SERVICE_DATA))!=null)
    qstring += SERVICE_DATA+"="+tmp;
else
    qstring += SERVICE_DATA+"=";

//get the GServlet basic url from the JNDI tree
//TIP: The env-entry GServletURL is declared in the ejb-jar.xml of Scheduler
String address;
try {
    Context ctx = new InitialContext();
    logger.info(ctx);
    address = (String) ctx.lookup("java:comp/env/scheduler/GServletURL");
}
catch(NamingException ne){
    logger.info("Unable to retrieve GServlet URL from JNDI.Trying from local variable");
    address = new String(GSERVLET_URL);
}

//create the url connection
String urlname = address+qstring;
URLConnection connection=null;
try {
    URL url= new URL(urlname);
    connection =(URLConnection) url.openConnection();
    connection.setRequestMethod("GET");
}

```

```

        connection.connect();
    }
    catch(Exception e){
        logger.error("RequestHandler: Cannot connect to GServlet");
        throw new Exception("RequestHandler: Cannot connect to GServlet");
    }

    //get the GServlet response
    String result="";
    try{
        BufferedReader in = new BufferedReader
            (new InputStreamReader(connection.getInputStream()));
        String temp;
        while ((temp = in.readLine()) != null)
            result = result + temp;
        in.close();
    }
    catch(Exception e){
        logger.error("RequestHandler: Cannot receive the GServlet response");
        throw new Exception("RequestHandler: Cannot receive the GServlet response");
    }
    return result;
}

```

```

private static String POSTcall(String userID, String serviceID, Hashtable serviceData)
throws Exception {
    //create the Parameters
    String tmp=null;
    String data = SERVICE_ID+"="+serviceID+"&";
    data += USER_ID+"="+userID+"&";
    if((tmp=(String)serviceData.get(USER_ADDRESS))!=null)
        data += USER_ADDRESS+"="+tmp+"&";
    else
        data += USER_ADDRESS+"=&";
    if((tmp=(String)serviceData.get(SECOND_RESPONSE_TYPE))!=null)
        data += SECOND_RESPONSE_TYPE+"="+tmp+"&";
    else
        data += SECOND_RESPONSE_TYPE+"=&";
    if((tmp=(String)serviceData.get(SERVICE_TYPE))!=null)
        data += SERVICE_TYPE+"="+tmp+"&";
    else
        data += SERVICE_TYPE+"=&";
    if((tmp=(String)serviceData.get(REQUEST_METHOD))!=null)
        data += REQUEST_METHOD+"="+tmp+"&";
    else
        data += REQUEST_METHOD+"=&";
}

```



```

if((tmp=(String)serviceData.get(SERVICE_DATA))!=null)
    data += SERVICE_DATA+"="+tmp;
else
    data += SERVICE_DATA+"=";

//write the Parameters to a Stream
ByteArrayOutputStream byteStream = new ByteArrayOutputStream(512);
PrintWriter out = new PrintWriter(byteStream, true);
out.print(data);
out.flush();

//get the GServlet basic url from the JNDI tree
//TIP: The env-entry GServletURL is declared in the ejb-jar.xml of Scheduler
String urlname;
try{
    Context ctx = new InitialContext();
    urlname = (String) ctx.lookup("java:comp/env/scheduler/GServletURL");
}
catch(NamingException ne){
    logger.info("Unable to retrieve GServlet URL from JNDI.Trying from local variable");
    urlname = new String(GSERVLET_URL);
}
//create the url connection and send the request
URLConnection connection=null;
try{
    URL url= new URL(urlname);
    connection =(URLConnection) url.openConnection();
    connection.setUseCaches(false);
    connection.setDoOutput(true);
    connection.setRequestMethod("POST");
    connection.setRequestProperty("Content-Length",String.valueOf(byteStream.size()));
    byteStream.writeTo(connection.getOutputStream());
}
catch(Exception e){
    logger.error("RequestHandler: Cannot connect to GServlet");
    throw new Exception("RequestHandler: Cannot connect to GServlet");
}

//get the GServlet response
String result="";
try{
    BufferedReader in = new BufferedReader
        (new InputStreamReader(connection.getInputStream()));
    String temp;
    while ((temp = in.readLine()) != null)
        result = result + temp;
}

```

```

        in.close();
    }
    catch(Exception e){
        logger.error("RequestHandler: Cannot receive the GServlert response");
        throw new Exception("RequestHandler: Cannot receive the GServlet response");
    }
    return result;
}
}
}

```

## Class SchedulerBean

```
package org.polos.kernel.scheduler;
```

```
import java.util.Date;
import java.util.Hashtable;
import java.util.Iterator;
```

```
import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.FinderException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;
```

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.naming.NoInitialContextException;
```

```
import org.apache.log4j.Logger;
import org.apache.log4j.Level;
```

```
/**
 * SchedulerBean
 *
 * @ejb:bean name="Scheduler"
 *     jndi-name="ejb/polos/kernel/scheduler/Scheduler"
 *     view-type="remote"
 *     type="Stateless"
 *     cmp-version="2.x"
 *
 * @ejb:ejb-ref ejb-name="Task" view-type="local"
 * @ejb:ejb-ref ejb-name="Sequence" view-type="local"
 * @ejb:env-entry name="scheduler/MaxDuration" type="java.lang.Integer" value="7"

```

```
*/
```

```
public class SchedulerBean implements SessionBean {
```

```
    private PolosTimerI scheduler=null;
```

```
    private Integer scheduleID=null;
```

```
    private static Logger logger=Logger.getLogger("polos.scheduler");
```

```
/**
```

```
 * @ejb:interface-method
```

```
*/
```

```
public void initialize() throws SchedulerException {
```

```
    TaskLocal task;
```

```
    Integer scheduleID;
```

```
    String userID, serviceID;
```

```
    InvocationParameters invocationData;
```

```
    Date firstDate, stopDate;
```

```
    long period;
```

```
    Hashtable serviceData;
```

```
    Task newTask;
```

```
    Iterator i;
```

```
    try {
```

```
        Context ctx =new InitialContext();
```

```
        //find Taskbean
```

```
        TaskHomeLocal taskhome =(TaskHomeLocal)
```

```
            ctx.lookup("ejb/polos/kernel/scheduler/Task" );
```

```
        i = taskhome.findAll().iterator();
```

```
        //schedule each Task in DB (some tasks are excluded)
```

```
        while (i.hasNext()) {
```

```
            task = (TaskLocal) i.next();
```

```
            scheduleID = task.getScheduleID();
```

```
            userID = task.getUserID();
```

```
            serviceID = task.getServiceID();
```

```
            invocationData = task.getInvocationData();
```

```
            firstDate = invocationData.getStart();
```

```
            stopDate = invocationData.getStop();
```

```
            period = invocationData.getPeriod();
```

```
            serviceData = task.getServiceData();
```

```
            if((new Date()).after(firstDate) && period!=0) {
```

```
                //Task has been executed at least one time
```

```
                //Set new firstDate for execution
```

```
                firstDate.setTime(((new Date()).getTime()-
```

```

        firstDate.getTime()/period+1)*period+firstDate.getTime());

        if(firstDate.after(stopDate)){
            //we don't need to schedule the task so remove it from DB
            task.remove();
            continue;
        }
        invocationData.setStart(firstDate);
    }
    newTask=new Task(scheduleID,userID,serviceID,invocationData,serviceData);

    //put newTask in timer
    try {
        if(period==0)
            scheduler.schedule(scheduleID,newTask,firstDate);
        else
            scheduler.schedule(scheduleID,newTask,firstDate,period);
        logger.debug("task with ID="+scheduleID.toString()+" scheduled");
    }
    catch(Exception e) {
        logger.error("Cannot schedule task with ID="+scheduleID.toString()+" in
            Timer");
        throw new SchedulerException("Initialize: Cannot schedule task in
            Timer");
    }

    }//end while
    logger.info("Polos Scheduler initialized");
}
catch(NoInitialContextException e) {
    logger.error("Cannot create an InitialContext");
    throw new SchedulerException("Initialize: Cannot create an InitialContext");
}
catch(NamingException e) {
    logger.error("Naming Error");
    throw new SchedulerException("Initialize: Naming Error");
}
catch(FinderException e) {
    logger.error("Error in finding Tasks in DB ");
    throw new SchedulerException("Initialize: Error in finding Tasks in DB ");
}
catch(Exception e) {
    logger.error("Unknown error in initializing Scheduler");
    logger.error("Thrown exception: "+e.toString());
    throw new SchedulerException("Initialize: Unknown error in initializing Scheduler");
}
}

```

```
}
```

```
/**
```

```
* @ejb:interface-method
```

```
*/
```

```
public Integer schedule(String serviceID, String userID, InvocationParameters invocationData,  
    Hashtable serviceData) throws SchedulerException {
```

```
    try {
```

```
        Context ctx = new InitialContext();
```

```
        //request from Positioning
```

```
        if(invocationData==null) {
```

```
            //Call RequestHandler
```

```
            RequestHandler rh = new RequestHandler();
```

```
            rh.handleRequest(userID,serviceID,serviceData);
```

```
            logger.info("Execution of task from POS with userID="+userID+" and  
                serviceID="+serviceID);
```

```
            return null;
```

```
        }
```

```
        else {
```

```
            //check and if needed convert the invocation parameters
```

```
            //convert N?? to E??
```

```
            if(invocationData.getStart()==null) {
```

```
                invocationData.setStart(new Date());
```

```
                //if it was a NNN treat it like it was from Positioning
```

```
                if(invocationData.getStop()==null && invocationData.getPeriod()==0){
```

```
                    //Call RequestHandler
```

```
                    RequestHandler rh = new RequestHandler();
```

```
                    rh.handleRequest(userID,serviceID,serviceData);
```

```
                    logger.info("Immediate execution of task with userID="+userID+"  
                        and serviceID="+serviceID);
```

```
                    return null;
```

```
                }
```

```
            }
```

```
            //convert ENE to EEE
```

```
            if(invocationData.getPeriod()!=0 && invocationData.getStop()==null) {
```

```
                Integer maxDuration = (Integer)
```

```
                    tx.lookup("java:/comp/env/scheduler/MaxDuration");
```

```
                long currTime=System.currentTimeMillis();
```

```
                Date newStop = new Date(currTime+
```

```
                    24*3600000*maxDuration.longValue());
```

```
                logger.debug("new stop date: "+ newStop.toString());
```

```
                invocationData.setStop(newStop);
```

```
            }
```

```

//convert EEN to EEE
if(invocationData.getPeriod()==0 && invocationData.getStop()!=null) {
    long newPeriod=invocationData.getStop().getTime()-
        invocationData.getStart().getTime();
    invocationData.setPeriod(newPeriod);
    //the new stop Date is a fraction of the period
    //greater than the period (3/4*Period)
    //so as to handle possible delays in the
    //execution of the task by the Timer
    Date newDate=new Date(invocationData.getStop().getTime()+
        3*newPeriod/4);
    invocationData.setStop(newDate);
}
//else
//get scheduleID from the Sequence Bean
try {
    SequenceHomeLocal seqhome = (SequenceHomeLocal)
        ctx.lookup("ejb/polos/kernel/scheduler/Sequence");
    SequenceLocal seq = seqhome.findByPrimaryKey("Scheduler");
    scheduleID = seq.getNextIndex();
}
catch(FinderException e) {
    logger.error("Cannot find Sequence Generator");
    throw new SchedulerException("Schedule: Cannot find Sequence Generator"); }
//create Task
Task newtask = new Task(scheduleID,serviceID,userID,invocationData,
    serviceData);
try {
    //schedule periodic Task in Timer
    if(invocationData.getPeriod()!=0) {

        scheduler.schedule(scheduleID,newtask,invocationData.getStart(),
            invocationData.getPeriod());
        //schedule non-periodic Task in Timer
    }
    else
        scheduler.schedule(scheduleID,newtask,invocationData.getStart());
}
catch(Exception e) {
    logger.error("Cannot schedule task with ID="+scheduleID.toString()+" in
        Timer");
    throw new SchedulerException("Schedule: Cannot schedule task with
        ID="+scheduleID.toString()+" in Timer");
}
//store Task in the DB table
TaskHomeLocal taskhome = (TaskHomeLocal) ctx.lookup

```

```

        ("ejb/polos/kernel/scheduler/Task");
        TaskLocal task = taskhome.create(scheduleID,userID,serviceID,invocationData,
            serviceData);
        logger.info("task with ID="+scheduleID.toString()+" scheduled");
        return scheduleID;
    }
    catch(NoInitialContextException e) {
        logger.error("Cannot create an InitialContext");
        throw new SchedulerException("Schedule: Cannot create an InitialContext");
    }
    catch(NamingException e) {
        logger.error("Naming Error");
        throw new SchedulerException("Schedule: Naming Error");
    }
    catch(CreateException e) {
        logger.error("Cannot create task with ID="+scheduleID.toString());
        throw new SchedulerException("Schedule: Cannot create task with
            ID="+scheduleID.toString());
    }
    catch(Exception e) {
        logger.error("Unknown error in scheduling Task");
        logger.error("Thrown exception: "+e.toString());
        throw new SchedulerException("Schedule: Unknown error in scheduling Task");
    }
}

/**
 * @ejb:interface-method
 */
public void remove(Integer scheduleID) throws SchedulerException {
    Context ctx = null;
    try {
        //remove from Timer
        scheduler.removeTask(scheduleID);
        logger.debug("task with ID="+scheduleID.toString()+" removed from Timer");
    }
    catch(Exception e) {
        logger.error("Cannot remove task with ID="+scheduleID.toString()+" from Timer");
        throw new SchedulerException("Remove: Cannot remove Task with
            ID="+scheduleID.toString()+" from Timer");
    }

    try {
        ctx = new InitialContext();
        //remove from DB table

```

```

        TaskHomeLocal taskhome = (TaskHomeLocal) ctx.lookup
            ("ejb/polos/kernel/scheduler/Task");
        TaskLocal task = taskhome.findByPrimaryKey(scheduleID);
        task.remove();
        logger.debug("task with ID="+scheduleID.toString()+" removed from DB");
        logger.info("task with ID="+scheduleID.toString()+" removed from Polos Scheduler");
    }
    catch(NoInitialContextException e) {
        logger.error("Cannot create an InitialContext");
        throw new SchedulerException("Remove: Cannot create an InitialContext");
    }
    catch(NamingException e) {
        logger.error("Naming Error");
        throw new SchedulerException("Remove: Naming Error");
    }
    catch(FinderException e) {
        logger.error("Cannot find task with PK="+scheduleID.toString());
        throw new SchedulerException("Remove: Cannot find task with
            PK="+scheduleID.toString());
    }
    catch(Exception e) {
        logger.error("Unknown error in removing task with ID="+scheduleID.toString());
        logger.error("Thrown exception: "+e.toString());
        throw new SchedulerException("Remove: Unknown error in removing task with
            ID="+scheduleID.toString());
    }
}

/**
 * @ejb:create-method
 */

public void ejbCreate() throws CreateException {
    try {
        Context ctx = new InitialContext();
        scheduler = (PolosTimerI) ctx.lookup("Scheduler");
    }
    catch(Exception e) {
        logger.error("Unknown error in Scheduler creation");
        logger.error("Thrown exception: "+e.toString());
        throw new CreateException();
    }
}

```



```

public void setSessionContext(SessionContext ctx)throws EJBException {}

public void ejbRemove() throws EJBException {}

public void ejbActivate() throws EJBException {}

public void ejbPassivate() throws EJBException {}
}

```

### **Class SchedulerException**

```

package org.polos.kernel.scheduler;

public class SchedulerException extends Exception {
    public SchedulerException() {}

    public SchedulerException(String message) {
        super(message);
    }
}

```

### **Class SequenceBean**

```

package org.polos.kernel.scheduler;

import java.rmi.RemoteException;

import javax.ejb.EJBException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.ejb.RemoveException;

import javax.ejb.CreateException;

/**
 * @author Administrator
 *
 * @ejb:bean name="Sequence"
 *         local-jndi-name="ejb/polos/kernel/scheduler/Sequence"

```

```

*     view-type="local"
*     type="CMP"
*     cmp-version="2.x"
*     primkey-field="sequence"
*
*
* @ejb:transaction type="Required"
* @ejb:pk
*     class="java.lang.String"
*
*/

```

```

public abstract class SequenceBean implements EntityBean {

```

```

    public EntityContext ejbContext;
    /**
     * @see javax.ejb.EntityBean#ejbActivate()
     */
    public void ejbActivate() throws EJBException, RemoteException {
    }

    /**
     * @see javax.ejb.EntityBean#ejbLoad()
     */
    public void ejbLoad() throws EJBException, RemoteException {
    }

    /**
     * @see javax.ejb.EntityBean#ejbPassivate()
     */
    public void ejbPassivate() throws EJBException, RemoteException {
    }

    /**
     * @see javax.ejb.EntityBean#ejbRemove()
     */
    public void ejbRemove()
        throws RemoveException, EJBException, RemoteException {
    }

    /**
     * @see javax.ejb.EntityBean#ejbStore()
     */
    public void ejbStore() throws EJBException, RemoteException {
    }

```

```

/**
 * @see javax.ejb.EntityBean#setEntityContext(EntityContext)
 */
public void setEntityContext(EntityContext ctx)
    throws EJBException, RemoteException {

   .ejbContext = ctx;
}

/**
 * @see javax.ejb.EntityBean#unsetEntityContext()
 */
public void unsetEntityContext() throws EJBException, RemoteException {

}

/**
 * @ejb:create-method
 */
public String ejbCreate(String sequence, Integer index) throws CreateException {
    this.setSequence(sequence);
    this.setIndex(index);
    return null;
}

/**
 * Post create-method for ejbCreate
 */
public void ejbPostCreate(String sequence, Integer index) throws CreateException {

}

/**
 * Returns the index value to be used in the next insertion
 * for the table in our DB with name similar to the sequence
 *
 * @ejb:interface-method
 */
public Integer getNextIndex() {

    this.setIndex(new Integer(this.getIndex().intValue()+1));
    return this.getIndex();
}

```

```

/**
 * @ejb:pk-field generate="false"
 * @ejb:persistent-field
 */
public abstract String getSequence();
public abstract void setSequence(String sequence);

/**
 * @ejb:interface-method
 * @ejb:persistent-field
 */
public abstract Integer getIndex();

public abstract void setIndex(Integer index);
}

```

## Class Task

```
package org.polos.kernel.scheduler;
```

```

import java.io.Serializable;
import java.util.Date;
import java.util.Hashtable;
import java.util.Properties;
import java.util.TimerTask;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.rmi.PortableRemoteObject;
import org.apache.log4j.Logger;

```

```

public class Task extends TimerTask implements Serializable {
    private String serviceID;
    private Hashtable serviceParams;
    private InvocationParameters invocParams;
    private String userID;
    private Integer scheduleID;
    private static Logger logger=Logger.getLogger("polos.scheduler");

    public Task(Integer scheduleID,String serviceID, String userID, InvocationParameters invocParams,
        Hashtable serviceParams) {
        super();
        this.serviceID=serviceID;
        this.userID=userID;
        this.serviceParams=serviceParams;
        this.invocParams=invocParams;
    }
}

```

```

        this.scheduleID=scheduleID;
    }

    public void run() {
        //if ENN then execute it and do cleanup from the DB
        if(invocParams.getPeriod()==0) {
            //Call RequestHandler
            RequestHandler.handleRequest(userID,serviceID,serviceParams);
            logger.info("Execution of task with scheduleID=" + this.scheduleID.toString());
            try {
                Context ctx = new InitialContext();
                SchedulerHome home = (SchedulerHome) PortableRemoteObject.
                    narrow(ctx.lookup("ejb/polos/kernel/scheduler/Scheduler"),
                        SchedulerHome.class);
                SchedulerRemote scheduler = home.create();
                scheduler.remove(scheduleID);
            }
            catch(Exception e) {
                logger.error("Task: Cannot remove the expired task with
                    scheduleID="+scheduleID.toString()+" from Scheduler");
                logger.error("Thrown exception: "+e.toString());
            }
        }
        else
            //if it is an EEE then check if it has expired
            if(new Date().after(invocParams.getStop())) {
                try{
                    Context ctx = new InitialContext();
                    SchedulerHome home = (SchedulerHome) PortableRemoteObject.
                        narrow(ctx.lookup("ejb/polos/kernel/scheduler/Scheduler"),
                            SchedulerHome.class);
                    SchedulerRemote scheduler = home.create();
                    scheduler.remove(scheduleID);
                }
                catch(Exception e) {
                    logger.error("Task: Cannot remove the expired task with scheduleID="+
                        scheduleID.toString()+ " from Scheduler");
                    logger.error("Thrown exception: "+e.toString());
                }
            }
        else {
            //Call RequestHandler
            RequestHandler.handleRequest(userID,serviceID,serviceParams);
            System.out.println("Execution of task with schedule ID="+this.scheduleID.toString());
        }
    }
}

```

## Class TaskBean

```
package org.polos.kernel.scheduler;
```

```
import javax.ejb.*;
```

```
import java.util.Hashtable;
```

```
/**  
 * @ejb:bean name="Task"  
 *     local-jndi-name="ejb/polos/kernel/scheduler/Task"  
 *     view-type="local"  
 *     type="CMP"  
 *     reentrant="True"  
 *     cmp-version="2.x"  
 *     primkey-field="scheduleID"  
 *     schema="TaskBean"  
 *  
 * @ejb:finder signature="Collection findByUserID(java.lang.String userID)" query="SELECT OBJECT(a)  
 * FROM TaskBean AS a WHERE a.userID = ?1"  
 *  
 * @ejb:finder signature="Collection findByServiceID(java.lang.String serviceID)" query="SELECT  
 * OBJECT(a) FROM TaskBean AS a WHERE a.serviceID = ?1"  
 *  
 * @ejb:finder signature="Collection findAll()" query="SELECT OBJECT(a) FROM TaskBean AS a"  
 *  
 * @ejb:pk  
 *     class="java.lang.Integer"  
 *  
 * @ejb:transaction type="Required"  
 * @jboss:table-name "TaskBean"  
 */
```

```
public abstract class TaskBean implements EntityBean {
```

```
    protected EntityContext ctx;  
    public void ejbActivate() {}  
    public void ejbRemove() {}  
    public void ejbPassivate() {}  
    public void ejbLoad() {}  
    public void ejbStore() {}  
    public void setEntityContext(EntityContext ctx) {  
        this.ctx = ctx;  
    }  
    public void unsetEntityContext() {  
        this.ctx = null;  
    }  
}
```

```
public void.ejbPostCreate(Integer scheduleID, String userID, String serviceID,InvocationParameters
    invocationData,Hashtable serviceData) {}
```

```
/**
```

```
 * @ejb:create-method
```

```
 */
```

```
public Integer.ejbCreate(Integer scheduleID, String userID, String serviceID,InvocationParameters
    invocationData,Hashtable serviceData)throws CreateException {
```

```
    setScheduleID(scheduleID);
    setUserID(userID);
    setServiceID(serviceID);
    setInvocationData(invocationData);
    setServiceData(serviceData);
    return scheduleID;
```

```
}
```

```
/**
```

```
 * @ejb:interface-method
```

```
 * @ejb:persistent-field
```

```
 */
```

```
public abstract String.getUserID();
```

```
/**
```

```
 * @ejb:interface-method
```

```
 */
```

```
public abstract void setUserID(String userID);
```

```
/**
```

```
 * @ejb:interface-method
```

```
 * @ejb:persistent-field
```

```
 */
```

```
public abstract String.getServiceID();
```

```
/**
```

```
 * @ejb:interface-method
```

```
 */
```

```
public abstract void setServiceID(String serviceID);
```

```
/**
 * @ejb:interface-method
 * @ejb:persistent-field
 */
public abstract InvocationParameters getInvocationData();
```

```
/**
 * @ejb:interface-method
 */
public abstract void setInvocationData(InvocationParameters invocationData);
```

```
/**
 * @ejb:interface-method
 * @ejb:persistent-field
 */
public abstract Hashtable getServiceData();
```

```
/**
 * @ejb:interface-method
 */
public abstract void setServiceData(Hashtable serviceData);
```

```
/**
 * @ejb:interface-method
 * @ejb:persistent-field
 */
public abstract Integer getScheduleID();

public abstract void setScheduleID(Integer scheduleID);
```

```
}
```