



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Ευφύων Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Κινητή πλατφόρμα για καταγραφή ζωτικών δεδομένων από αισθητήρες στο
«υπολογιστικό νέφος»**

Mobile platform to Monitoring Vital Data from Sensors in cloud Computing

Θεόδωρος Σουλτανόπουλος

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Ευριπίδης Πετράκης, Καθηγητής (Επιβλέπων)

Α. Δελγιαννάκης, Καθηγητής

Δρ. Στέλιος Σωτηριάδης, Επιστημονικός Συνεργάτης

XANIA 2015

Abstract

The idea of the Internet of things, combined with the technology of cloud computing, open new horizons in the field of healthcare. The use of wearable sensors and their possibility of connectivity to the Internet provide significant benefits not only to healthcare organizations, to which they allow an instant, easy and efficient monitoring of the patients' vital data from anywhere they are, leading to cost savings in areas such as equipment and nursing staff, but also to patients themselves, who can now easily and safely perform their daily activities, wearing the appropriate sensors.

The Intelligent Gateway application, taking advantage of the possibilities offered by smartphones that have android software, converts the mobile device to a gateway between the wearable Bluetooth low energy (BLE) sensors and the cloud. In particular, this application allows the collection, processing and storage of vital data that the BLE sensors produce, such as the pulse rate, as well as the transfer of useful information to the cloud through a "smart" and dynamic way. "Smart" because there will be an internal processing of the signal that are received from BLE sensors that declares the way data is send to cloud .Dynamic because our system has the ability to recognize new characteristic values of BLE sensors by editing a XML schema .

Finally, the use of cloud services, through the Intelligate application, allows the reception, processing and storage of data, the recording of the measuring indications of the sensors in real time, the information of the patient by his doctor and several additional functions, combining Generic enablers and computing resources that are accessed online and that are provided by cloud infrastructures, helping in the overall interoperability of the system.

All the levels of the Intelligate system communicate interactively with each other, based on the model of service oriented architecture, giving our system the possibility of scalability that is allowing the addition of extra functions in the future, without affecting the already existing ones.

Περίληψη

Το εγχείρημα του διαδικτύου των πραγμάτων(Internet of things), σε συνδυασμό με την τεχνολογία του υπολογιστικού νέφους(cloud), ανοίγουν νέους ορίζοντες στον τομέα της υγειονομικής περίθαλψης. Η χρήση φορητών αισθητήρων και η δυνατότητα συνδεσιμότητάς τους στο διαδίκτυο παρέχουν σημαντικά οφέλη τόσο στους οργανισμούς παροχής υπηρεσιών υγείας, στους οποίους επιτρέπει την άμεση, εύκολη και αποτελεσματική παρακολούθηση των ζωτικών δεδομένων των ασθενών από όποιο σημείο και αν βρίσκονται, με αποτέλεσμα την εξοικονόμηση κόστους σε τομείς όπως αυτός του υλικού εξοπλισμού και του νοσηλευτικού προσωπικού, όσο και στους ίδιους τους ασθενείς, οι οποίοι μπορούν πλέον, φορώντας τους κατάλληλους αισθητήρες, να εκτελούν τις καθημερινές τους δραστηριότητες με άνεση και ασφάλεια.

Η εφαρμογή Intelligate (Intelligent Gateway), εκμεταλλευόμενη τις δυνατότητες που προσφέρουν οι έξυπνες κινητές συσκευές (smartphones), λογισμικού android, μετατρέπει την κινητή συσκευή σε δίαυλο επικοινωνίας(Gateway) μεταξύ Bluetooth low energy(BLE) φορητών αισθητήρων και νέφους(cloud). Ειδικότερα, η εφαρμογή αυτή επιτρέπει την συλλογή, επεξεργασία και αποθήκευση ζωτικών δεδομένων που παράγουν οι BLE αισθητήρες, όπως ο καρδιακός ρυθμός , καθώς και την μεταφορά χρήσιμων πληροφοριών προς το νέφος με «έξυπνο» και δυναμικό τρόπο. Έξυπνο για το λόγο του ότι τα σήματα που θα λαμβάνονται στην συσκευή θα υπόκεινται μια εσωτερική επεξεργασία ώστε να στέλνονται στο νέφος όταν κατάσταση είναι κρίσιμη ή όταν θα το επιλέξει ο ίδιος ο χρήστης της εφαρμογής. Δυναμικό καθώς η εφαρμογή θα μπορεί να δέχεται την αναγνώριση των χαρακτηριστικών τιμών νέων BLE αισθητήρων με την συμπλήρωση ενός XML σχήματος.

Τελικά, η χρήση των υπηρεσιών του υπολογιστικού νέφους, μέσω της εφαρμογής Intelligate, επιτρέπει τη λήψη, επεξεργασία και αποθήκευση δεδομένων, την καταγραφή των ενδείξεων μέτρησης των αισθητήρων σε πραγματικό χρόνο, την ενημέρωση του ασθενούς από τον γιατρό του και αρκετές επιπλέον λειτουργίες, συνδυάζοντας παρεχόμενες υπηρεσίες γενικού σκοπού(Generic enablers) και υπολογιστικούς πόρους που προσπελαύνονται διαδικτυακά και παρέχονται από υποδομές νέφους βοηθώντας συνολικά στην δια λειτουργικότητα του συστήματος.

Όλα τα επίπεδα του συστήματος Intelligate επικοινωνούν αμφίδρομα μεταξύ τους, με βάση το μοντέλο της υπηρεσιοκεντρικής αρχιτεκτονικής (Service Oriented Architecture),

δίνοντας στο σύστημα μας την δυνατότητα επεκτασιμότητας επιτρέποντας δηλαδή την προσθήκη στο μέλλον επιπλέον λειτουργιών, χωρίς να επηρεάζονται οι ήδη υπάρχουσες.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Πετράκη για την ουσιαστική καθοδήγηση του μέχρι την ολοκλήρωση της διπλωματικής μου εργασίας. Τους βοηθούς του εργαστηρίου Κώστα, Αλέξανδρο, Σπύρο και τον διδακτορικό Στέλιο για την πολύτιμη βοήθεια τους καθώς και όλους τους προπτυχιακούς φοιτητές. Τέλος ευχαριστώ την οικογένεια μου για την υλική και ψυχολογική στήριξη καθ' όλη την διάρκεια των σπουδών μου καθώς και την Βασιλική.

Η διπλωματική εργασία είναι αφιερωμένη στους γονείς μου Μιχαήλ και Ευνομία για την συνεχή στήριξη τους σε όλες μου τις επιλογές.

Περιεχόμενα

Abstract	1
Περίληψη	2
Ευχαριστίες	3
1 Εισαγωγή	7
1.1 Περιγραφή του γενικού Προβλήματος.....	7
1.3 Περιγραφή της λύσης	8
1.4 Δομή της εργασίας	10
2 Υπόβαθρο και υπάρχουσες Τεχνολογίες	11
2.1 Τεχνολογία υπολογιστικού νέφους	11
2.1.1 Ορισμός.....	11
2.1.2 Μοντέλα παροχής υπηρεσιών νέφους	12
2.1.3 Μοντέλα ανάπτυξης υπολογιστικού νέφους.....	14
2.1.4 Πλεονεκτήματα και μειονεκτήματα τεχνολογίας υπολογιστικού νέφους	16
2.1.5 Υποδομές νέφους.....	18
2.1.6 Εικονοποίηση πόρων σε Υπολογιστικό Νέφος	20
2.2 Αισθητήρες στο νέφος και διαδίκτυο των Πραγμάτων	22
2.2.1 Ορισμός Διαδικτύου των πραγμάτων(Internet of things)	22
2.2.2 Διείσδυση επιχειρήσεων στο Διαδίκτυο των πραγμάτων	23
2.2.3 Διαδίκτυο των Πραγμάτων (Internet of Things) και Υπολογιστικό Νέφος (Cloud Computing)	24
2.2.4 Υπηρεσιοκεντρική Αρχιτεκτονική (Service Oriented Architecture) , λειτουργικά μέρη και προσεγγίσεις	25
2.3 Το Πρωτόκολλο Bluetooth Low Energy (BLE).....	29
2.3.1 Ιστορικά στοιχεία και σχέση με το IOT.....	29
2.3.2 Βασικά χαρακτηριστικά και σχεδιαστικές επιλογές του BLE.....	31
2.3.3 Πρωτόκολλο Χαρακτηριστικών (Attribute Protocol) και Γενικό Προφίλ Χαρακτηριστικών (Generic Attribute Profile Protocol)	32
2.4 Λογισμικό Android.....	39

2.4.1 Το λογισμικό Android και ιστορικά στοιχεία	39
2.4.3 Βασικά συστατικά για την δημιουργία εφαρμογής σε Android	41
2.4.4 Επιπλέον χρήσιμα συστατικά Λογισμικού Android για την δημιουργία της εφαρμογής μας.....	44
3 Intelligate(Intelligent Gateway) και γενικός σχεδιασμός του συστήματος	46
3.1 Σενάριο Χρήσης και λειτουργικότητα της εφαρμογής	46
3.2 Γενική Αρχιτεκτονική	48
3.3 Η αρχιτεκτονική του συστήματος Intelligate	49
3.3.1 Παραγωγοί (Producers).....	50
3.3.2 Έξυπνη Πύλη διαδικτύου(Intelligate).....	52
3.3.3 Διεπαφή διαχείρισης Συστήματος νέφους(Cloud back-end)	53
3.3.4 Καταναλωτές (Consumers).....	54
3.4 Υπηρεσίες γενικού Σκοπού για της ανάγκες του συστήματος Intelligate.....	55
3.4.1 JSON storage GE και ο τρόπος λειτουργίας του	55
3.4.2 Google Cloud messaging GE και ο τρόπος λειτουργίας του	57
3.4.3 Identity Management-KeyRock.....	58
3.5 Ανάλυση ροής δεδομένων και ενεργειών συστήματος Intelligate.....	59
4 Υλοποίηση του συστήματος	64
4.1 Υλοποίηση ενότητας Έξυπνης πύλης Διαδικτύου	64
4.1.1 Εισαγωγή σχήματος αισθητήρα στο σύστημα Intelligate.....	64
4.1.2 Sensor Data Collector	66
4.1.3 Local Storage	68
4.1.4 Connectivity Service	70
4.1.5 Notification Service	75
4.1.6 Application Logic(Intelligate)	76
4.1.7 Intelligate Application και γραφικό περιβάλλον (User Interface)	77
4.2 Υλοποίηση ενότητας Διεπαφή Διαχείρισης Συστήματος Νέφους	83
4.2.1 Παραμετροποίηση εικονικών μηχανών	84
4.2.2 Παραμετροποίηση JSON STORAGE GE	85
4.2.3 Παραμετροποίηση Google Cloud Messaging GE	87
4.2.4 Sensor History GE	90

4.2.5	Connectivity Service	91
4.2.6	Application Logic (cloud) και μηχανισμοί.	91
5	Συμπεράσματα και μελλοντικές επεκτάσεις	98
	Bibliography	102

1 Εισαγωγή

1.1 Περιγραφή του γενικού Προβλήματος

Τα τελευταία χρόνια υπάρχει μια μεγάλη αύξηση της ζήτησης σε υπηρεσίες υγείας ενώ οι ελλείψεις σε ειδικευμένους επαγγελματίες αποτελούν μια από τις πιο δύσκολες προκλήσεις που αντιμετωπίζουν οι πάροχοι υγειονομικής περίθαλψης.

Ειδικά μια κατηγορία η οποία θίγεται επιπλέον άμεσα από τα παραπάνω είναι οι ίδιοι οι ασθενείς, ιδιαίτερα άτομα με κινητικά προβλήματα και άτομα τρίτης ηλικίας όπου η επαφή με το εκάστοτε νοσοκομείο ή ιδιώτη γιατρό πρέπει να είναι άμεση, υποχρεωτική και παράλληλα τακτική. Παραδειγματικά μιλώντας ως υποθέσουμε ότι ένας ασθενής πάσχει από ταχυκαρδία. Θα πρέπει να υπάρχει μια τακτική συνάντηση του με τον εκάστοτε επιμελητή γιατρό του για τον έλεγχο και την πρόοδο της ασθένειας του. Επιπλέον συχνό γεγονός είναι ο ίδιος ο ασθενής να βρίσκεται σε αυστηρά κρίσιμη κατάσταση νοσηλείας όπου η συνεχής παρακολούθηση του είναι επιτακτική αλλά και ταυτόχρονα αδύνατη για έναν μεγάλο αριθμό ασθενών.

Συμπερασματικά μπορεί να γίνει εύκολα κατανοητό ότι για την ομαλή πρόοδο των ασθενών υπάρχει η ανάγκη συνεχούς παρακολούθησης, καταγραφής του ιστορικού συμπτωμάτων και επεξεργασίας των δεδομένων που παράγουν τα κατάλληλα όργανα μέτρησης εντός αλλά και εκτός του χώρου νοσηλείας τους οποιαδήποτε χρονική στιγμή.

Με τον καιρό οι ασθένειες γίνονται όλο και πιο πολύπλοκες και νέες εξελίξεις στην τεχνολογία και στην έρευνα έχουν διευκολύνει την δημιουργία νέων και πιο αποτελεσματικών διαγνώσεων και θεραπευτικών τεχνικών. Πολλές τεχνολογίες αισθητήρων που χρησιμοποιούνται στην βιομηχανία μπορούν να εφαρμοστούν στον τομέα της ιατρικής με το πρόσχημα της συνεχούς παρακολούθησης ειδικών αντιδράσεων ενός ασθενούς. Προς το παρόν οι αισθητήρες απλά καταγράφουν αλλαγές μιας κατάστασης, παρόλα αυτά υπολογίζεται ότι μέσα σε 10 χρόνια θα υπάρχει τεχνολογία αισθητήρων όπου θα μπορούν άμεσα να παρέμβουν στην υγειονομική περίθαλψη ενός ατόμου αλλάζοντας έτσι τους ρόλους των νοσοκομείων, χώρων στα εξωτερικά ιατρεία, και σχέσης γιατρού-ασθενούς. Βιοαισθητήρες¹ (biosensors) θα προκαλέσουν ριζικές αλλαγές στον χώρο της ιατρικής φροντίδας επιτρέποντας σε πολλές περιπτώσεις η ποιότητα της περίθαλψης να βελτιωθεί και ασθενείς να μπορούν να ελέγχονται απομακρυσμένα από τον γιατρό στα σπίτια τους.

¹ <https://en.wikipedia.org/wiki/Biosensor>

Αυτήν την στιγμή στην αγορά εργασίας υπάρχουν αρκετοί τύποι αισθητήρων οι οποίοι θα μπορούσαν να συντελέσουν στον απομακρυσμένο έλεγχο κατάστασης ενός ασθενούς από τον γιατρό του. Τα πρότυπα επικοινωνίας τους διαφέρουν από αισθητήρα σε αισθητήρα. Ένα πρότυπο από αυτά είναι και το Bluetooth low energy² στο οποίο μάλιστα θα γίνει αναλυτική περιγραφή στα επόμενα κεφάλαια. Το επιπλέον πρόβλημα που προκύπτει είναι ο τρόπος μετάβασης των δεδομένων από ένα διαφορετικό πρότυπο επικοινωνίας(Bluetooth low energy) σε ένα άλλο(Πρωτόκολλο διαδικτύου – TCP-IP) ώστε να είναι επιτρεπτή η ακέραια μεταφορά και επεξεργασία των δεδομένων από τον ασθενή στον γιατρό οποιαδήποτε χρονική στιγμή με δυναμικό τρόπο.

1.3 Περιγραφή της λύσης

Για την αντιμετώπιση των παραπάνω προβλημάτων δημιουργήθηκε το σύστημα «έξυπνης πύλης δικτύου» Intelligate (Intelligent Gateway³). Δηλαδή μιας συσκευής με δυνατότητα σύνδεσης στο διαδίκτυο, ώστε να είναι εφικτή η λήψη σημάτων διαφορετικών αισθητήρων και με κατάλληλους μηχανισμούς η αποστολή των δεδομένων σε διαφορετικές υπηρεσίες ανεπτυγμένες στο νέφος(cloud). Το γεγονός του ορισμού «έξυπνη πύλη διαδικτύου» δεν είναι τυχαίο καθώς η συγκεκριμένη πύλη είναι πλέον μια εφαρμογή αναπτυγμένη σε λογισμικό android⁴ η οποία έχει την δυνατότητα να λαμβάνει και να στέλνει σήματα που παράγουν οι αισθητήρες με έξυπνο τρόπο. Αυτό σημαίνει ότι τα σήματα που θα λαμβάνονται στην συσκευή θα υπόκεινται μια εσωτερική επεξεργασία ώστε να στέλνονται σε μια υπηρεσία όταν κατάσταση είναι κρίσιμη ή όταν θα το επιλέξει ο ίδιος ο χρήστης της εφαρμογής.

Με την βοήθεια του Intelligate ο ασθενής φορώντας τους αισθητήρες που μετρούν ζωτικής σημασίας δεδομένα(καρδιακούς παλμούς, οξύγνο στο αίμα ...) θα μπορεί μέσω μιας κινητής συσκευής(mobile Gateway) να χρησιμοποιεί μία εφαρμογή όπου θα του επιτρέπει να παρακολουθεί ο ίδιος τις μετρήσεις του αλλά και ταυτόχρονα όλες αυτές οι μετρήσεις να μπορούν να επιβλέπονται και από έναν γιατρό μέσω ενός φυλλομετρητή ο οποίος θα βρίσκεται σε μια διαφορετική τοποθεσία. Ο ασθενής θα μπορεί να κοιτάει τις μετρήσεις του μέσω του έξυπνου κινητού(smart-phone) του , ενώ ο γιατρός μέσω ενός “φυλλομετρητή ιστοσελίδων”⁵(browser) θα μπορεί να βλέπει την πρόοδο κάθε ασθενούς από όπου και αν βρίσκεται την ίδια χρονική στιγμή όπου μπορεί να γίνεται η μέτρηση .

Η διευκόλυνση των παρόχους υγειονομικής περίθαλψης και ασθενών είναι φανερή για τον καθένα για διαφορετικούς λόγους.

² https://en.wikipedia.org/wiki/Bluetooth_low_energy

³ [https://en.wikipedia.org/wiki/Gateway_\(computer_program\)](https://en.wikipedia.org/wiki/Gateway_(computer_program))

⁴ <https://el.wikipedia.org/wiki/Android>

⁵ https://el.wikipedia.org/wiki/Web_browser

- Για τους Οργανισμούς υγειονομικής Περίθαλψης(Νοσοκομεία-Γιατροί-τεχνικοί διαχειριστές)
 - ✓ Μεγαλύτερο κέρδος σε νοσοκομεία ή γιατρούς καθώς δίνεται η δυνατότητα να εξυπηρετούν περισσότερους ασθενείς και απομακρυσμένα μέσω απλά με την χρήση του διαδικτύου.
 - ✓ Εισαγωγή οποιαδήποτε καινούργιου BLE αισθητήρα με την βοήθεια συμπλήρωσης φόρμας για το σχήμα του.
 - ✓ Ο Γιατρός μπορεί να επιβλέπει οποιονδήποτε χρήστη σε πραγματικό χρόνο τις μετρήσεις που παράγουν οι αισθητήρες για κάθε ασθενή και επιπλέον έχει πρόσβαση στο ιστορικό δεδομένων του χρήστη για κάθε τιμή που είχε παράγει ο αισθητήρας.
 - ✓ Ο Γιατρός ειδοποιείται όταν η κατάσταση του ασθενούς είναι κρίσιμη και είναι σε θέση να αποστείλει μήνυμα για την παρότρυνση του ασθενούς σε μία συγκεκριμένη ενέργεια.
 - ✓ Πλέον η γραφειοκρατία περιορίζεται καθώς ο τεράστιος αριθμός ασθενών σε ένα νοσοκομείο θα γίνεται πλέον μέσω αποθήκευσης τους στο νέφος.
 - ✓ Μείωση εξόδων νοσοκομείων καθώς δεν είναι θα υπεύθυνα για την συντήρηση του συστήματος καθώς παρέχεται μέσω της τεχνολογίας νέφους όπου θα αναλυθεί αργότερα.
 - ✓ Έλεγχος και παραμετροποίηση των κανόνων που διέπουν οι αισθητήρες για κάθε χρήστη.

- Για τους ίδιους τους ασθενείς
 - ✓ Οι ασθενείς νιώθουν ασφαλείς καθώς η παρακολούθηση είναι συνεχής και μπορούν και οι ίδιοι να την ορίσουν ενεργοποιώντας απλά την εφαρμογή και την σύνδεση τους με τους αισθητήρες.

- ✓ Δυνατότητα αύξησης της συχνότητας προώθησης δεδομένων στο νέφος άμα οι ίδιοι νιώσουν αδιαθεσία ή παραβούν κάποιο κανόνα που έθεσε ο γιατρός χωρίς να καταλάβουν.
- ✓ Διαρκή επικοινωνία μέσω μηνυμάτων με τον φορέα περίθαλψής τους.
- ✓ Δυνατότητα χρήσης της εφαρμογής για χρονικό διάστημα που ο χρήστης επιθυμεί και κοστολόγηση σύμφωνα με αυτόν τον χρόνο.

1.4 Δομή της εργασίας

Στο **2^ο κεφάλαιο** γίνεται εκτενέστερη αναφορά στην τεχνολογία Υπολογιστικού Νέφους. Δίνεται ο ορισμός και αναλύονται τα πλεονεκτήματα και οι αδυναμίες της τεχνολογίας Νέφους, τί προσφέρει σήμερα στον κόσμο της τεχνολογίας αλλά και πού έχουν εντοπιστεί αδυναμίες. Επιπλέον, γίνεται ανάλυση όσων αφορά τα μοντέλα παροχής υπηρεσιών στο νέφος, τι είναι υπηρεσίες γενικού και ειδικού σκοπού. Αναφερόμαστε στην Υπηρεσιοκεντρική Αρχιτεκτονική και τις τεχνολογίες που χρησιμοποιούμε στην παρούσα διπλωματική εργασία (REST, HTTP, JSON κ.α.). Στη συνέχεια, περιγράφεται η νέα τάση στο διαδίκτυο, Διαδίκτυο των Πραγμάτων (Internet of Things) και συγκεκριμένα του m-health(mobile health), τι μας προσφέρει, πως συσχετίζεται άμεσα με το Υπολογιστικό Νέφος και τον τρόπο λειτουργίας του. Έπειτα γίνεται ενδεικτική αναφορά στο Πρωτόκολλο Επικοινωνίας Bluetooth Low energy με βάση τα χαρακτηριστικά του την διαφορά του από το κλασσικό πρωτόκολλο επικοινωνίας Bluetooth καθώς και για τον τρόπο λειτουργίας μετάδοσης πληροφορίας από μια περιφερειακή συσκευή(αισθητήρας) σε έναν δέκτη(κινητή συσκευή). Προφανώς αναγκαία είναι και η αναφορά των βασικών χαρακτηριστικών των αισθητήρων που χρησιμοποιήθηκαν για την υλοποίηση της συγκεκριμένης εφαρμογής. Στο τέλος του 2^{ου} κεφαλαίου γίνεται αναφορά συγκεκριμένα στο λογισμικό Android. Η αναφορά απασχολεί την βασική αρχιτεκτονική του λογισμικού, τα βασικά συστατικά της εφαρμογής μας καθώς και δευτερεύοντα συστατικά για την ομαλή και αποτελεσματική λειτουργία της εφαρμογής.

Στο **3^ο κεφάλαιο** , αρχικά παρουσιάζεται το σενάριο χρήσης το οποίο καλύπτεται από την εφαρμογή Intelligate. Επιπλέον γίνεται πλέον φανερό το γενικό σχήμα αρχιτεκτονικής του συστήματος και η γενική περιγραφή του στα επιμέρους κομμάτια του σε επίπεδο διεπαφής χρήστη(Front end) και διεπαφής συστήματος(Back end) βάση της υπηρεσιοκεντρικής αρχιτεκτονικής. Στην συνέχεια γίνεται λεπτομερής ανάλυση της ροής δεδομένων με βάση το

γενικό σχήμα αρχιτεκτονικής καθώς και οι λειτουργικές απαιτήσεις του συστήματος που μας απασχόλησε στην διπλωματική μας εργασία.

Στο **4^ο κεφάλαιο** , πλέον προχωράμε στην περιγραφή της υλοποίησης του συστήματος. Γίνεται ειδική ανάλυση στα επιμέρους κομμάτια της γενικής αρχιτεκτονικής. Πλέον κάθε υπηρεσία ειδικού σκοπού(Generic enabler) και υπηρεσιών που έχει υλοποιηθεί ή παραμετροποιηθεί αναλύεται ως ξεχωριστό κομμάτι βάση της υλοποίησης και λειτουργικότητας που της δώσαμε. Έπειτα γίνεται η πλήρης αναφορά του τρόπου υλοποίησης των μηχανισμών που προσφέρει η εφαρμογή σε συνδυασμό με το γραφικό περιβάλλον(User interface) της εφαρμογής σε επίπεδο συσκευής και διαδικτύου.

Τέλος στο **5^ο κεφάλαιο** θα παραθέσουμε τα συμπεράσματα που προέκυψαν από την ενασχόληση μας με την υλοποίηση της εφαρμογής Intelligate καθώς και προτάσεις για μελλοντική δουλειά.

2 Υπόβαθρο και υπάρχουσες Τεχνολογίες

2.1 Τεχνολογία υπολογιστικού νέφους

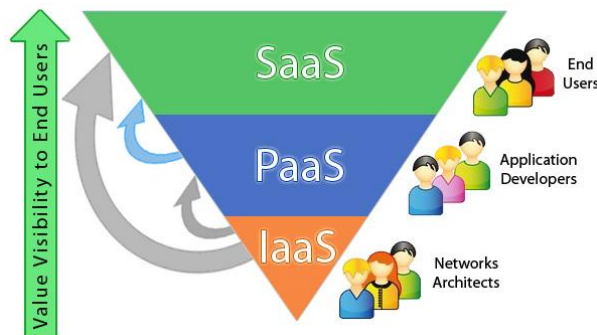
2.1.1 Ορισμός

Υπολογιστικό νέφος ονομάζεται η κατ' αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων(δίκτυο, εξυπηρετητές ,εφαρμογές και υπηρεσίες) με υψηλή ευελιξία , ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση [3].

Στο υπολογιστικό νέφος η αποθήκευση, η επεξεργασία και η χρήση δεδομένων, λογισμικού και υπηρεσιών γίνεται διαδικτυακά, μέσω απομακρυσμένων υπολογιστών σε υπολογιστικά κέντρα (Datacenter) για τα οποία ο χρήστης δεν γνωρίζει που βρίσκονται. Υπηρεσίες όπως η κατ' αίτηση παροχή εικονικών μηχανών, το διαδικτυακό ηλεκτρονικό ταχυδρομείο ή τα κοινωνικά δίκτυα βασίζονται στην τεχνολογία του. Αυτό το μοντέλο αποτελείται από τρία μοντέλα παροχής υπηρεσιών και τέσσερα μοντέλα ανάπτυξης.

2.1.2 Μοντέλα παροχής υπηρεσιών νέφους

Όπως αναφέραμε και στην προηγούμενη ενότητα το μοντέλο του υπολογιστικού νέφους αποτελείται από τρία μοντέλα παροχής υπηρεσιών τα οποία συνοψίζονται στο Σχήμα 1 και εξηγούνται στο παρακάτω κείμενο.



Σχήμα 1-Μοντέλα Παροχής υπηρεσιών νέφους

- **Υπολογιστική Υποδομή ως Υπηρεσία (Infrastructure as a Service - IaaS):** Αυτή η μορφή Νέφους δίνει πρόσβαση σε εικονικές πλατφόρμες υποδομών (hardware), που περιλαμβάνουν τα μηχανήματα, το δίκτυο και τα αποθηκευτικά μέσα. Με αυτή τη μορφή, οι αρχιτέκτονες διαδικτύου μπορούν να δημιουργήσουν τη δική τους Υπολογιστική Υποδομή πάνω στην οποία είναι οι ίδιοι υπεύθυνοι στο να εγκαταστήσουν, να διατηρήσουν και να τις παρέχουν σε άλλους εξαρτώμενους χρήστες. Η υποδομή είναι το χαμηλότερο επίπεδο και είναι ένα μέσο για να παρέχεται η επεξεργασία, η αποθήκευση, το δίκτυο και άλλοι βασικοί υπολογιστικοί πόροι σαν δεδομένες υπηρεσίες δια μέσου του διαδικτύου. Οι εξυπηρετητές, τα αποθηκευτικά συστήματα, οι δικτυακοί διακόπτες (network switches), οι δρομολογητές (routers) και άλλα συστήματα, χειρίζονται διάφορους τύπους φόρτου εργασίας, από μια σειρά προγραμμάτων που εκτελούνται χωρίς παρέμβαση χρήστη μέχρι και την αποθήκευση

στον εξυπηρετητή. Οι πάροχοι των υπηρεσιών Νέφους μπορούν να έχουν τα δικά τους λειτουργικά συστήματα και λογισμικό για την υποκείμενη υπολογιστική υποδομή που χρησιμοποιούν στο Νέφος. Παραδείγματα αυτού του μοντέλου: Amazon EC2⁶, Rackspace⁷ κ.α.

- **Βάση ως Υπηρεσία (Platform as a Service – PaaS):** Παρέχει πρόσβαση σε ένα προγραμματιστικό περιβάλλον ή σε ένα περιβάλλον εργασίας, με δυνατότητα επεκτάσιμης υπολογιστικής ισχύος και δομές δεδομένων ενσωματωμένες σε αυτό. Με το PaaS οι χρήστες μπορούν να αναπτύξουν και να εκτελέσουν τις δικές τους εφαρμογές μέσα στο περιβάλλον που τους παρέχεται από τον πάροχο της υπηρεσίας. Αυτό το μεσαίο επίπεδο παρέχει υψηλότερο επίπεδο αφαίρεσης και υπηρεσίες για την ανάπτυξη, εφαρμογή, φιλοξενία και διατήρηση εφαρμογών στο ίδιο ενσωματωμένο περιβάλλον. Αυτό το επίπεδο παρέχει ένα περιβάλλον εργασίας και τα μέσα για την τοποθέτηση εφαρμογών που χρησιμοποιούν γλώσσες προγραμματισμού που υποστηρίζει ο πάροχος του Νέφους. Παραδείγματα αυτού του μοντέλου: Google App Engine⁸, Windows Azure⁹.
- **Λογισμικό ως Υπηρεσία (Software as a Service – SaaS)** Παρέχει πρόσβαση σε μια συλλογή από εφαρμογές λογισμικού. Οι πάροχοι SaaS προσφέρουν πρόσβαση σε συγκεκριμένες εφαρμογές που ελέγχονται και εκτελούνται στην υποδομή των παρόχων. Το SaaS συχνά αναφέρεται σαν “Software on Demand” (Λογισμικό σε ζήτηση) και είναι η γνωστότερη μορφή Νέφους που σήμερα καταναλώνουν οι περισσότεροι χρήστες. Παραδείγματα αυτού του μοντέλου: Google Apps, Yahoo Mail, Dropbox κ.α.

Όπως βλέπουμε λοιπόν και από το ‘Σχήμα 1’ και στα τρία μοντέλα παροχής υπηρεσιών τα δίκτυα, ο αποθηκευτικός χώρος, οι εξυπηρετητές και ο μηχανισμός της εικονικοποίησης διαχειρίζονται από τον πάροχο της υπηρεσίας. Στο μοντέλο Υπολογιστική Υποδομή ως Υπηρεσία ο καταναλωτής έχει το δικαίωμα διαχείρισης των Εικονικών Μηχανών-Λειτουργικών Συστημάτων, των δεδομένων, των εφαρμογών και του συνδετικού λογισμικού που μεσολαβεί μεταξύ αυτών και του λειτουργικού συστήματος. Στο μοντέλο Βάση ως Υπηρεσία ο καταναλωτής διαχειρίζεται μόνο τα δεδομένα του και τις εφαρμογές που ο ίδιος έχει παράξει.

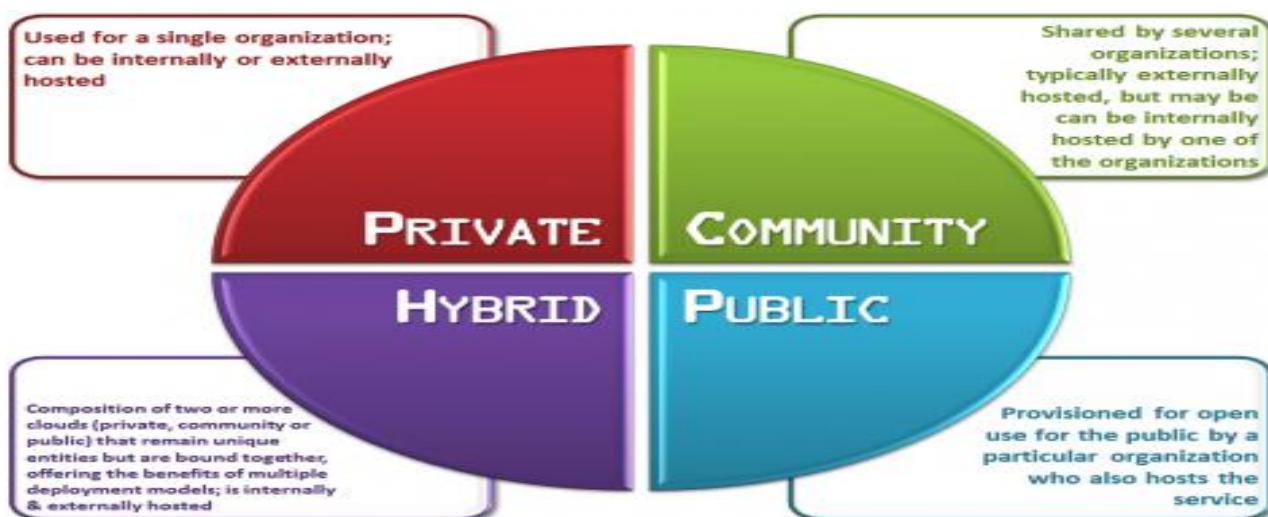
⁶ <http://aws.amazon.com/ec2/>

⁷ <http://www.rackspace.com/managed-cloud/>

⁸ <https://cloud.google.com/appengine/docs>

2.1.3 Μοντέλα ανάπτυξης υπολογιστικού νέφους

Το Υπολογιστικό Νέφος, όπως αναφέραμε χαρακτηρίζεται ως προς το μοντέλο ανάπτυξης της υποδομής. Τα τέσσερα μοντέλα είναι τα Δημόσιο Νέφος (Public Cloud), Ιδιωτικό Νέφος (Private Cloud), Κοινοτικό Νέφος (Community Cloud), Υβριδικό Νέφος (Hybrid Cloud) [3,4,6].



Σχήμα 2 – Μοντέλα ανάπτυξης Υπολογιστικού νέφους

- **Δημόσιο Νέφος (Public Cloud)** :Το Δημόσιο Νέφος αποτελεί ένα σύνολο από υπολογιστικούς πόρους, οι οποίοι διατίθενται μέσω του διαδικτύου και είναι διαθέσιμοι σε όλο το κοινό. Τα πλεονεκτήματα αυτής της υποδομής είναι η χρέωση με βάση το μοντέλο «Pay as you go»¹⁰, η μεγάλη ευελιξία λόγω της άμεσης παροχής υπηρεσιών, η ύπαρξη κλιμάκωσης σε μεγαλύτερη ή μικρότερη χωρητικότητα σύμφωνα με τις ανάγκες του χρήστη χωρίς ο ίδιος να γνωρίζει από πού παράγονται . Ωστόσο τα δημόσια νέφη είναι πιο ευάλωτα σε θέμα ασφάλειας από τα ιδιωτικά νέφη. Εταιρίες κολοσσοί όπως η Amazon¹¹, Google¹² και Microsoft¹³ αποτελούν παρόχους δημόσιου νέφους.

Το δημόσιο νέφος είναι μια ιδανική επιλογή όταν

¹⁰ https://en.wikipedia.org/wiki/Pay-as-you-go_tax

¹¹ https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud

¹² https://en.wikipedia.org/wiki/Google_Cloud_Platform

¹³ https://en.wikipedia.org/wiki/Cloud_computing

- ✓ Υπάρχει απασχόληση με τυποποιημένο φόρτο εργασίας για τις εφαρμογές που χρησιμοποιούνται από πολλούς ανθρώπους, όπως το ηλεκτρονικό ταχυδρομείο
- ✓ Όταν θέλουμε να παράξουμε κώδικα για το στήσιμο και την δημιουργία μιας εφαρμογής με την βοήθεια υπηρεσιών που μας προσφέρει το δημόσιο νέφος.
- ✓ Χρήση επιπλέον υπολογιστικών πόρων σε ώρες αιχμής.
- ✓ Όταν θέλουμε να κάνουμε εργασίες συνεργασίας με άλλες εταιρίες.
- **Ιδιωτικό Νέφος (Private Cloud)**

Οι υπηρεσίες παρέχονται σε χρήστες που ανήκουν σε έναν συγκεκριμένο οργανισμό σε αντίθεση με το δημόσιο νέφος που απευθύνεται σε όλους. Τέτοιου είδους λύσεις υιοθετούν εταιρείες που επιθυμούν τον πλήρη έλεγχο των υπηρεσιών, οι οποίες ακολουθούν ένα κοινό νομικό πλαίσιο παροχής ώστε να διασφαλίζεται η ασφάλειά τους. Σημαντικό χαρακτηριστικό είναι το μεγάλο κόστος δημιουργίας και λειτουργίας μιας τέτοιας υποδομής.

Το ιδιωτικό νέφος είναι μια ιδανική επιλογή όταν

- ✓ Το κέντρο δεδομένων(datacenter) θα πρέπει να γίνει πιο αποτελεσματικό σε πόρους(μνήμη, επεξεργαστική ισχύ κτλ).
- ✓ Είναι αναγκαία η παροχή ιδιωτικών υπηρεσιών νέφους.
- ✓ Είναι χρήσιμη συνοχή σε όλες τις υπηρεσίες που προσφέρονται από τον οργανισμό.
- **Κοινοτικό Νέφος (Community Cloud)**

Η υποδομή Κοινοτικού Νέφους (Community Cloud) είναι διαμοιρασμένη σε πολλούς οργανισμούς και εξυπηρετεί μια συγκεκριμένη κοινότητα η οποία έχει σαν κοινό τόπο κάποιο ενδιαφέρον ή στόχο. Αυτή την υποδομή μπορεί να διαχειρίζεται κάποιος από τους αναφερθέντες οργανισμούς, ή από έναν τρίτο οργανισμό ή επιχείρηση.

Το κοινοτικό νέφος είναι μια ιδανική επιλογή όταν

- ✓ Κυβερνητικοί οργανισμοί χρειάζονται να διαμοιράζονται πόρους.
- ✓ Είναι αναγκαία η δημιουργία ενός κοινού συστήματος πρόσβασης σε προφίλ πελατών και εργαζομένων(Health Insurance Portability and Accountability ¹⁴)για μια ομάδα νοσοκομείων ή κλινικών

¹⁴ https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act

- **Υβριδικό Νέφος (Hybrid Cloud)**

Το Υβριδικό Νέφος είναι το μοντέλο που συνδυάζει όλα τα παραπάνω Νέφη, ώστε να δημιουργηθεί μια υποδομή. Η χρήση αυτού του μοντέλου παρέχει σε ορισμένες περιπτώσεις μεγαλύτερη ευελιξία σε επιχειρήσεις, δεδομένου ότι είναι εφικτή μια ποικιλία συνδυασμών των πόρων. Οι συνδυασμοί αυτοί επιτρέπουν την αποτελεσματικότερη αντιμετώπιση φόρτου εργασίας και μεγάλων απαιτήσεων υπολογιστικών πόρων. Μια ιδανική επιλογή για το υβριδικό νέφος είναι το γεγονός μιας εταιρίας να θέλει να χρησιμοποιήσει μια εφαρμογή SaaS αλλά ανησυχεί για την ασφάλεια που της παρέχει.

2.1.4 Πλεονεκτήματα και μειονεκτήματα τεχνολογίας υπολογιστικού νέφους

Η αυξανόμενη ζήτηση του κοινού για το Υπολογιστικό Νέφος δείχνει ότι αυτή η νέα τεχνολογία προσφέρει πολλά οφέλη στους χρήστες και τις επιχειρήσεις.

- **Μειωμένο Κόστος:** Οι χρήστες που επιλέγουν να χρησιμοποιήσουν εφαρμογές ή υπηρεσίες που παρέχονται από το Νέφος επωφελούνται οικονομικά. Η συντήρηση και οι αναβαθμίσεις του λογισμικού είναι αποκλειστική ευθύνη του πάροχου και όχι του χρήστη. Έτσι αποφεύγεται η πρόσληψη νέου εξειδικευμένου προσωπικού από τις επιχειρήσεις. Επίσης, ο εξοπλισμός που χρειάζεται μια εταιρεία μειώνεται, καθώς οι υπηρεσίες φιλοξενούνται σε απομακρυσμένους εξυπηρετητές και αρκεί οποιαδήποτε συσκευή με πρόσβαση στο διαδίκτυο ώστε αυτές να χρησιμοποιηθούν.
- **Ευελιξία:** Η πρόσβαση σε υπηρεσίες που παρέχει το Νέφος πραγματοποιείται από οποιαδήποτε τοποθεσία ή χρονική στιγμή ζητήσει ο χρήστης, με δεδομένη την ύπαρξη σύνδεσης στο δίκτυο.
- **Ελαστικότητα :** Ο χρήστης έχει τη δυνατότητα να διαχειρίζεται τους πόρους που του παρέχει το Νέφος σύμφωνα με τις ανάγκες του. Με αυτόν τον τρόπο οι καταναλωτές μπορούν να αυξάνουν υπολογιστική ισχύ (αποθηκευτικό χώρο, μνήμη κ.α.) ανά τακτά χρονικά διαστήματα ή να απελευθερώνουν πόρους όταν πλέον δεν είναι χρήσιμοι.
- **Χρέωση ανά χρήση:** Οι παρεχόμενοι πόροι είναι μετρήσιμοι και κοστολογούνται σύμφωνα με το χρονικό διάστημα χρήσης τους. Έτσι, η κατανάλωση και η χρέωσή τους ελέγχεται πλήρως από τον χρήστη. Για επιχειρήσεις που χρησιμοποιούν εφαρμογές με

βάση τα ωράρια λειτουργίας τους αυτό το μοντέλο παροχής υπηρεσιών ενδείκνυται καθώς μειώνει τα έξοδα τους σε σημαντικό βαθμό.

- **Αύξηση παραγωγικότητας:** Οι υπηρεσίες εξυπηρετούν παράλληλα και την ίδια χρονική στιγμή πολλούς χρήστες. Μια εταιρεία που απασχολεί μεγάλο αριθμό εργαζομένων έχει τη δυνατότητα να διαμερίσει αρχεία από ένα κεντρικό Η/Υ σε όλους τους εργαζόμενους ταυτόχρονα και όχι στον καθένα ξεχωριστά. Αυτό έχει σαν αποτέλεσμα την γρήγορη ενημέρωση των χρηστών και κατ' επέκταση την όσο το δυνατόν καλύτερη απόδοσή τους.

Όπως σε κάθε νέα τεχνολογία, έτσι και το Υπολογιστικό Νέφος διχάζει και προβληματίζει τους καταναλωτές, για το κατά πόσο είναι μια επαρκής και ασφαλής λύση γι' αυτούς. Πέρα από τα πλεονεκτήματα που προσφέρει το Νέφος υπάρχουν και αρκετά ερωτήματα για τα αρνητικά αποτελέσματα που είναι πιθανόν αυτό να προκαλέσει. Μερικά από αυτά είναι « Τα δεδομένα είναι ασφαλή; » , « Σε περίπτωση πτώσης του δικτύου μπορώ να έχω πρόσβαση στις υπηρεσίες; ». Θα προσπαθήσουμε παρακάτω να αναλύσουμε και να δώσουμε απαντήσεις στα κενά που έχουν παρατηρηθεί.

- **Ασφάλεια και Μυστικότητα:** Όταν αναφερόμαστε στο Υπολογιστικό Νέφος, το ερώτημα που γεννάται είναι, αν τα δεδομένα που φιλοξενούνται σε αυτό είναι ασφαλή. Είναι γεγονός ότι οι χρήστες είναι ιδιαίτερα επιφυλακτικοί στο να εμπιστευτούν ευαίσθητα δεδομένα σε τρίτους, καθώς η ασφάλεια τους δεν διασφαλίζεται πλήρως από κανέναν πάροχο και δεν γίνονται γνωστές στους καταναλωτές τεχνικές λεπτομέρειες, όπως το πού βρίσκονται τα δεδομένα τους . Σύμφωνα με μελέτες που έχουν πραγματοποιηθεί υπάρχει πάντα ο κίνδυνος ευαίσθητα δεδομένα να εκτίθενται δημόσια στο διαδίκτυο. Έτσι συμπεραίνουμε ότι σε μια εποχή που οι συνέπειες και τα πιθανά κόστη από πιθανές αστοχίες αυξάνονται, οι εταιρείες που διαχειρίζονται εμπιστευτικά δεδομένα πρέπει να αναπτύξουν καλύτερους τρόπους για την αξιολόγηση των πρακτικών ασφαλείας και μυστικότητας στις υπηρεσίες Υπολογιστικού Νέφους, ώστε να κερδίσουν την εμπιστοσύνη του κοινού.
- **Διαγραφή δεδομένων:** Η ασφαλής και αποτελεσματική διαγραφή των δεδομένων κατά τον τερματισμό της συνεργασίας πελάτη-παρόχου δεν διασφαλίζεται. Δεν γίνεται γνωστή στον χρήστη η μέθοδος με την οποία διαγράφονται τα δεδομένα του και η ύπαρξη κάποιου αντίγραφου αυτών, που μπορεί να χρησιμοποιηθεί από μη εξουσιοδοτημένους χρήστες του Νέφους.

- **Νομικά ζητήματα:** Η συμμόρφωση με το νόμο είναι ένα πολύ σημαντικό στοιχείο μεταξύ του καταναλωτή και του πάροχου από τη στιγμή που η νομοθεσία στο ζήτημα του Υπολογιστικού Νέφους θεωρείται προβληματική. Ο χρήστης, σήμερα, δεν γνωρίζει επαρκώς τα δικαιώματά του απέναντι στον πάροχο και ποιά είναι τα δικαιώματα του πάροχου πάνω στα δεδομένα του. Το ζήτημα αυτό εντείνεται από τη στιγμή που οι υποδομές Νέφους που φιλοξενούν τα δεδομένα δεν βρίσκονται σε ένα σημείο του πλανήτη αλλά διαμερίζονται ανά τον κόσμο, καθώς δεν υπάρχει ένα ενιαίο νομοθετικό πλαίσιο για την προστασία του καταναλωτή.
- **Χρήση δικτύου:** Όπως έχουμε αναφέρει η πρόσβαση στις υπηρεσίες Νέφους πραγματοποιείται από οποιαδήποτε συσκευή έχει πρόσβαση στο διαδίκτυο, κάτι το οποίο διευκολύνει τον χρήστη. Το πλεονέκτημα αυτό, μετατρέπεται σε μειονέκτημα σε περίπτωση πιθανής διακοπής του δικτύου, καθώς γίνεται αδύνατη η πρόσβαση σε υπηρεσίες ή αρχεία που ο καταναλωτής επιθυμεί να προσπελάσει.

2.1.5 Υποδομές νέφους

2.1.5.1 Openstack

Το Openstack [7] είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο επιτρέπει την δημιουργία μιας υποδομής Υπολογιστικού Νέφους. Οι θεμελιωτές της δημιουργίας αυτού του λογισμικού είναι οι Rackspace Hosting και η NASA, όταν το 2010 αποφάσισαν να δουλέψουν πάνω σε ένα κοινό έργο για την υλοποίηση υποδομής Υπολογιστικού Νέφους. Σήμερα, είναι διαδεδομένο σε όλο τον κόσμο καθώς πάνω από διακόσιες εταιρείες χρησιμοποιούν Openstack στα Νέφη τους όπως είναι η Yahoo¹⁵, Intel¹⁶, Oracle¹⁷ κ.α. Τα συστήματα που είναι σχεδιασμένα στα πρότυπα του Openstack αποτελούνται από μια κεντρική αρχιτεκτονική και από επιμέρους μικρότερα κομμάτια κώδικα που είναι υπεύθυνα για τον έλεγχο του μεγάλου όγκου υπολογιστικών πόρων που διαχειρίζονται. Εύκολη και άμεση είναι και αλληλεπίδραση του χρήστη πάνω στο λογισμικό καθώς μέσω κάποιας κονσόλας τερματικού ή από διεπαφές προγραμματισμού εφαρμογών (APIS) που παρέχονται μπορεί να έχει τον πλήρη έλεγχο τον φόρτου εργασίας της υποδομής και την ορθή διαμέριση των πόρων.

¹⁵ <http://www.yahoo.com>

¹⁶ <http://www.intel.com/content/www/us/en/cloud-computing/intel-cloud-based-solutions.html>

¹⁷ <https://cloud.oracle.com/home>

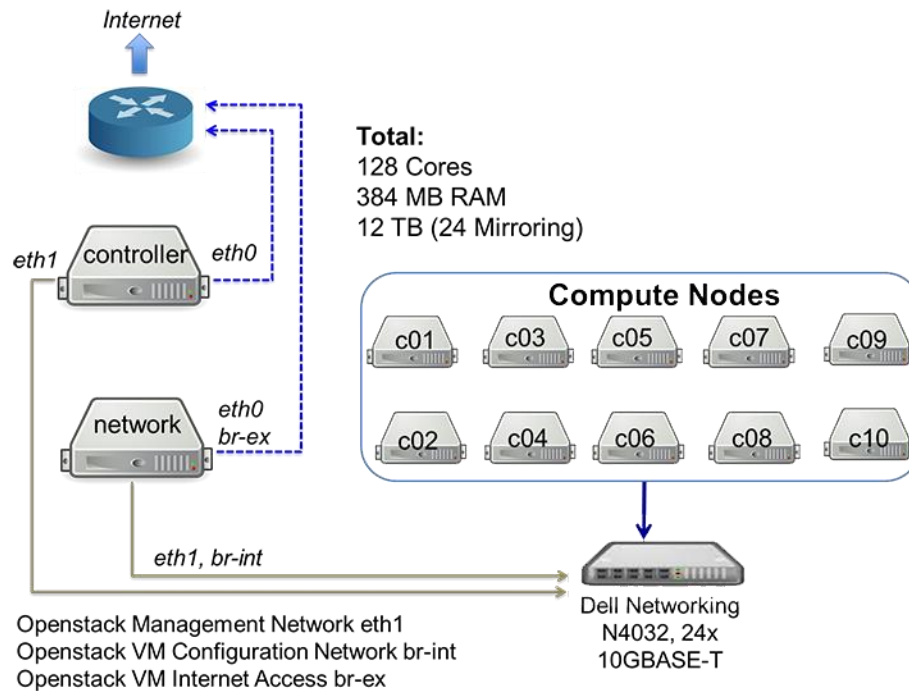
2.1.5.2 Υποδομή Fiware

Το FIWARE είναι μια μη εμπορική πλατφόρμα που προσφέρει υπηρεσίες γενικού σκοπού (GEs) αλλά και ειδικού σκοπού (SEs) ακολουθούμενες από απλές διεπαφές προγραμματισμού εφαρμογών (APIs) ώστε να είναι δυνατή η υλοποίηση «έξυπνων» εφαρμογών. Οι Υπηρεσίες γενικού σκοπού (Generic Enablers) παρέχονται από υποδομές Υπολογιστικού Νέφους ως SaaS για την ανάπτυξη εφαρμογών. Επιπλέον, δίνεται η δυνατότητα στον χρήστη να αποκτήσει την κατάλληλη τεχνογνωσία ώστε να εξοικειωθεί με τις υπηρεσίες καθώς παρέχονται πειράματα και αποτελέσματα των υπηρεσιών αυτών από χρήστες που ήδη έχουν κάνει χρήση αυτών. Οι υπηρεσίες διανέμονται δωρεάν και είναι δημόσιες ώστε το κοινό να έχει πρόσβαση σε αυτές. Το γεγονός αυτό κάνει την πλατφόρμα ελκυστική στους χρήστες που θέλουν να δημιουργήσουν δικές τους εφαρμογές. Οι παροχές του FIWARE δεν σταματάνε στην διανομή υπηρεσιών με την μορφή λογισμικού αλλά δίνει την δυνατότητα σε προγραμματιστές να αποκτήσουν υπηρεσίες ως υποδομή (IaaS) δημιουργώντας εικονικές μηχανές και δεσμεύοντας υπολογιστικούς πόρους (μνήμη, επεξεργαστική ισχύ, αποθηκευτικό χώρο).

2.1.5.3 Υποδομή Intellicloud

Η υποδομή Υπολογιστικού Νέφους Intellicloud, σχεδιάστηκε, υλοποιήθηκε και συντηρείται από το εργαστήριο Ευφών Συστημάτων. Σκοπός της υποδομής αυτής είναι η παροχή υπολογιστικών πόρων, για την ανάπτυξη εφαρμογών στο Νέφος. Η υποδομή φιλοξενείται εξολοκλήρου στο Πολυτεχνείο Κρήτης και αυτή τη στιγμή περιλαμβάνει 128 πυρήνες επεξεργαστικής ισχύς, 384 GB RAM και 12 TB σε σκληρό δίσκο (25 TB σε εικονικό σκληρό δίσκο). Το λογισμικό και η αρχιτεκτονική που είναι υπεύθυνο για τη λειτουργία του Intellicloud είναι βασισμένο σε Openstack Grizzly¹⁸.

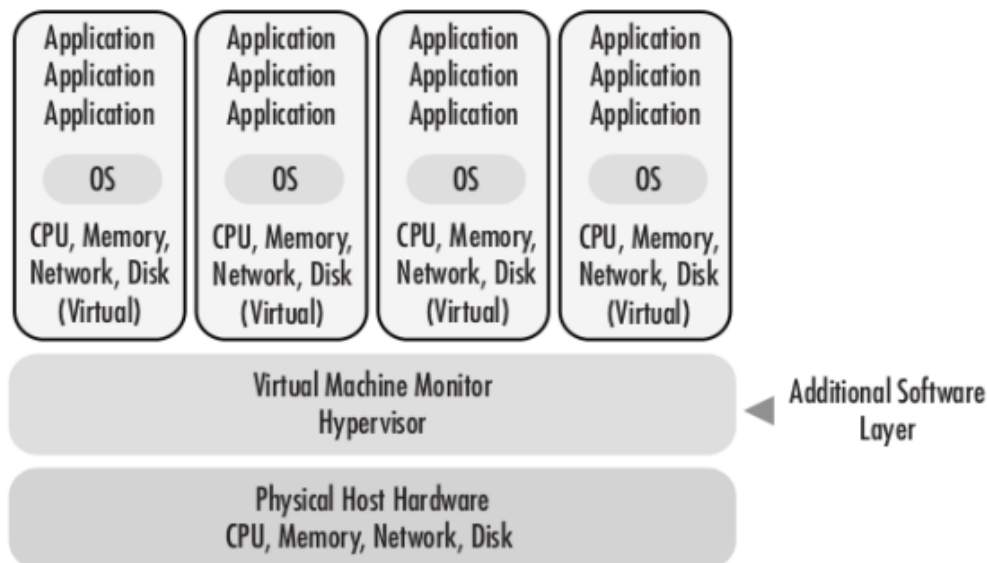
¹⁸ <https://www.openstack.org/software/grizzly/>



Σχήμα 3 Αρχιτεκτονική Intellicloud

2.1.6 Εικονοποίηση πόρων σε Υπολογιστικό Νέφος

Στο Υπολογιστικό Νέφος ιδιαίτερη βαρύτητα δίνεται από κάθε πάροχο στον διαμοιρασμό των υπολογιστικών πόρων, έτσι ώστε οι καταναλωτές να παραμένουν ευχαριστημένοι με τις παρεχόμενες υπηρεσίες. Το λογισμικό το οποίο είναι υπεύθυνο για το πώς θα διανεμηθούν οι διαθέσιμοι πόροι του Υπολογιστικού Νέφους ονομάζεται επιμελητής εικονικών μηχανών (hypervisor) .



**ΣΧΗΜΑ 4: Η ΣΧΕΣΗ ΜΕΤΑΞΥ ΦΥΣΙΚΗΣ ΥΠΟΔΟΜΗΣ -
HYPERVISOR - ΕΙΚΟΝΙΚΩΝ ΜΗΧΑΝΩΝ**

Το πρόσθετο στρώμα λογισμικού που διακρίνεται στο σχήμα 4 μεταξύ της φυσικής υποδομής (CPU, μνήμη, δίκτυο) και των εικονικών μηχανών αναφέρεται ως Επιμελητής Εικονικών Μηχανών – Hypervisor. Στην ουσία πρόκειται για ένα λειτουργικό σύστημα το οποίο εξυπηρετεί τις αιτήσεις που κάνουν οι εικονικές μηχανές προς τη φυσική υποδομή για τη χρήση συγκεκριμένων υπολογιστικών πόρων.

Το σχήμα 4 μοντελοποιεί την αλληλεπίδραση τριών οντοτήτων της Υποδομής Νέφους που παίζουν καθοριστικό ρόλο στην εξυπηρέτηση των καταναλωτών

- Οι **Εικονικές Μηχανές** αποτελούν ένα σύνολο από εικονικούς πόρους (εικονική επεξεργαστική ισχύ, εικονική μνήμη, εικονικός αποθηκευτικός χώρος κ.α), ένα Φιλοξενούμενο Λειτουργικό Σύστημα και τις εφαρμογές που εκτελούνται πάνω σ' αυτό το Λειτουργικό Σύστημα.
- Ο **Επιμελητής Εικονικών Μηχανών**, ο οποίος διανέμει τους πόρους στις Εικονικές Μηχανές ανάλογα με τις απαιτήσεις τους.
- Οι **Φυσικοί Πόροι**, αποτελούν την πραγματική επεξεργαστική ισχύ, μνήμη, αποθηκευτικό χώρο και δίκτυο που διαθέτει η Υποδομή Νέφους.

Η Υποδομή Νέφους του εργαστηρίου μας, Intellicloud είναι βασισμένη στην ανοιχτού κώδικα πλατφόρμα Openstack και ο Επιμελητής Εικονικών Μηχανών που βασίζεται η όλη λειτουργία της υποδομής είναι ο KVM Hypervisor¹⁹.

2.2 Αισθητήρες στο νέφος και διαδίκτυο των Πραγμάτων

2.2.1 Ορισμός Διαδικτύου των πραγμάτων(Internet of things)

“Καθημερινά πράγματα “ξυπνούν”, “γνωρίζουν” το φυσικό τους περιβάλλον, επικοινωνούν την πληροφορία που συλλέγουν, λαμβάνουν πληροφορία από εξωτερικές πηγές”[8]

Αυτή ή τελευταία πρόταση , μας βοηθάει να κατανοήσουμε τον ορισμός της έκφρασης Διαδικτύου των πραγμάτων(Internet of Things).Πιο συγκεκριμένα το διαδίκτυο των πραγμάτων είναι ένα εγχείρημα, μια ιδέα όπου έχει την βάση του στην σύνδεση διάφορων μικρών συσκευών ή οχημάτων με ενσωματωμένους αισθητήρες και εξοπλισμό διασύνδεσης(tablets,κινητά τηλέφωνα, κάμερες ,ανιχνευτές-ιχνηλάτες σώματος(body trackers) και αναρίθμητες άλλες συσκευές) τόσο μεταξύ τους όσο και με τον κατασκευαστή, για να λαμβάνουν και να μεταδίδουν σχετικά δεδομένα με στόχο να προσφέρουν περισσότερες υπηρεσίες και πρόσθετη αξία. Δηλαδή, πρόκειται για ένα αναπτυσσόμενο δίκτυο των καθημερινών αντικειμένων - από τις βιομηχανικές μηχανές έως τα καταναλωτικά αγαθά -που μπορεί να μοιράζεται πληροφορίες και να ολοκληρώνει εργασίες, ενώ ο χρήστης των αντικειμένων μπορεί να είναι απασχολημένος με άλλες δραστηριότητες.

Όλες αυτές οι συνδεδεμένες συσκευές που απαρτίζουν σήμερα το Διαδίκτυο των Πραγμάτων, χρησιμοποιούνται προκειμένου να συλλέξουν πληροφορίες από το περιβάλλον και την ανθρώπινη δραστηριότητα. Στόχος είναι η βελτίωση της ποιότητας ζωής των ανθρώπων και η αποφυγή απρόβλεπτων γεγονότων.

Η συλλογή δεδομένων κρίνεται απαραίτητη στους παρακάτω τομείς:

¹⁹http://www.linux-kvm.org/page/Main_Page

- **Μεταφορές :** Έξυπνες λύσεις στον κλάδο των μεταφορών, επιτυγχάνουν μείωση της κίνησης στους δρόμους, μείωση της κατανάλωσης καυσίμων, θέτουν προτεραιότητες στα προγράμματα επισκευής των οχημάτων, και σώζουν ζωές
- **Ανανεώσιμες πηγές ενέργειας :** Έξυπνα ηλεκτρικά δίκτυα που ενσωματώνουν περισσότερες ανανεώσιμες πηγές ενέργειας, βελτιώνουν την αξιοπιστία των συστημάτων και μειώνουν τις χρεώσεις των καταναλωτών, προσφέροντας έτσι φθηνότερη ηλεκτρική ενέργεια
- **Υγείας :** Απομακρυσμένη παρακολούθηση ασθενών, που παρέχει εύκολη πρόσβαση στην ιατροφαρμακευτική περίθαλψη, βελτιώνοντας την ποιότητα των υπηρεσιών, αυξάνοντας τον αριθμό των ατόμων που εξυπηρετούνται και εξοικονομώντας χρήματα.
- **Βιομηχανία :** Αισθητήρες μηχανών παρακολούθησης, που εντοπίζουν και προβλέπουν ζητήματα συντήρησης, αναπλήρωσης αποθεμάτων, και ορίζουν ακόμη και προτεραιότητες στα χρονοδιαγράμματα των εργασιών συντήρησης, για επισκευές και περιφερειακές εργασίες
- **Αυτοματοποιημένα σπίτια :** Έλεγχος όλων των ηλεκτρικών συσκευών ενός σπιτιού

2.2.2 Διείσδυση επιχειρήσεων στο Διαδίκτυο των πραγμάτων

Θα πρέπει να γίνει κατανοητό, πως όλη η εξέλιξη σημαντικών τομέων που αποτελούν την καθημερινότητα των ανθρώπων παίρνει μια άλλη πορεία για την ακρίβεια έχει ήδη αρχίσει, καθώς σύντομα, τα αυτοκίνητά , τα σπίτια, οι μεγάλες συσκευές και ακόμη και οι δρόμοι των πόλεων θα πρέπει να συνδέονται στο Internet δημιουργώντας αυτό το δίκτυο αντικειμένων που ονομάζεται Internet of Things, ή IoT για συντομία, αποτελούμενο από εκατομμύρια αισθητήρες και συσκευές που παράγουν συνεχείς ροές δεδομένων. Οι εταιρίες έχουν ήδη αρχίσει να μετακινούνται προς αυτήν την ιδέα και να επενδύουν τεράστια χρηματικά ποσά. Είναι ευνόητο πως οι πρώτοι που θα ενδιαφερθούν θα είναι οι εταιρείες οι οποίες δραστηριοποιούνται στον βιομηχανικό κλάδο "big data" καθώς και στον κλάδο των "analytics", όπως είναι οι IBM²⁰ και Cisco²¹, καθώς και εταιρείες από τον χώρο του οικιακού εξοπλισμού, όπως είναι η GE²².

Συγκεκριμένα σύμφωνα με τα παραπάνω παρατίθενται ορισμένα ενδεικτικά στοιχεία

²⁰ <http://www.ibm.com/gr-el/>

²¹ <http://www.cisco.com/>

²² <http://www.ge.com/>

- Η συνολική προστιθέμενη αξία από το Internet of things σε όλους τους κλάδους, θα φτάσει τα 1.9 τρισεκατομμύρια δολάρια ως το 2020 παγκοσμίως σύμφωνα με την Gartner.
- Πενήντα δισεκατομμύρια συσκευές θα συνδεθούν με το Internet μέχρι το 2020 προβλέπει η Cisco.
- Η αγορά των μηχανημάτων για εξ αποστάσεως παρακολούθηση ασθενών (remote patient monitoring), διπλασιάστηκε από το 2007 ως το 2011, και αναμένεται να διπλασιαστεί ξανά ως το 2016.
- Οι πόλεις επρόκειτο να δαπανήσουν \$41 τρισεκατομμύρια μέσα στα επόμενα 20 χρόνια, στην αναβάθμιση των υποδομών του Internet of Things σύμφωνα με την Intel.

2.2.3 Διαδίκτυο των Πραγμάτων (Internet of Things) και Υπολογιστικό Νέφος (Cloud Computing)

Το πρόβλημα της διαχείρισης και αποθήκευσης του τεράστιου όγκου δεδομένων που παράγουν αυτές οι συσκευές έρχεται να λύσει η τεχνολογία του Υπολογιστικού Νέφους. Όπως αναλύσαμε σε προηγούμενες παραγράφους του κεφαλαίου 2 τα οφέλη που μας προσφέρει το Υπολογιστικό Νέφος είναι πολλαπλά, οπότε οι δύο αυτές έννοιες δεν θα μπορούσαν παρά να είναι άμεσα συνδεδεμένες μεταξύ τους. «Έξυπνες» εφαρμογές που αναπτύσσονται στα πλαίσια μιας «έξυπνης» πόλης, θα πρέπει να έχουν την δυνατότητα της ευελιξίας και ελαστικότητας πλεονεκτήματα που αυτή τη στιγμή μόνο η τεχνολογία Υπολογιστικού Νέφους προσφέρει. Επίσης, η χρέωση ανά χρήση που προσφέρει στους καταναλωτές την κάνει ακόμα πιο ελκυστική στην χρήση της σε τέτοιου είδους εφαρμογές αφού μια συσκευή-αισθητήρας μπορεί να στέλνει συγκεκριμένες ώρες της ημέρας δεδομένα. Για παράδειγμα, αν ένας ασθενής ο οποίος παρακολουθείται από τον γιατρό του φορώντας φορητούς αισθητήρες τότε ο γιατρός θα μπορεί να ειδοποιείται μόνο στις κρίσιμες καταστάσεις του ασθενούς και όχι κάθε χρονική στιγμή που ο αισθητήρας παράγει μια μέτρηση.

Πολλές εταιρίες που δραστηριοποιούνται στην παροχή υπηρεσιών Υπολογιστικού Νέφους ήδη έχουν αρχίσει την ανάπτυξη ειδικών εργαλείων για την αποθήκευση, επεξεργασία και ανάλυση δεδομένων που προέρχονται από συσκευές IoT. Η ανάπτυξη ειδικών υπηρεσιών σε περιβάλλοντα Υπολογιστικού Νέφους για την διαχείριση των συσκευών του Διαδικτύου των Πραγμάτων βρίσκεται ήδη σε ανάπτυξη και αυτές οι υπηρεσίες είναι άμεσα διαθέσιμες σε δημιουργούς λογισμικού για την δημιουργία καινοτόμων εφαρμογών. Ειδικότερα, νεοσύστατες εταιρίες στον χώρο της τεχνολογίας (start-ups) έχουν την μεγαλύτερη ανάγκη από την

τεχνολογία που προσφέρουν οι πάροχοι μέσω SaaS και PaaS για συσκευές του Διαδικτύου των Πραγμάτων λόγω του χαμηλού κόστους.

2.2.4 Υπηρεσιοκεντρική Αρχιτεκτονική (Service Oriented Architecture) , λειτουργικά μέρη και προσεγγίσεις .

2.2.4.1 Βασική Λειτουργία και Ορισμός

Το σύστημα το οποίο θα αναπτύξουμε θα βασίζεται στην αρχιτεκτονική ανάπτυξης εφαρμογών “Service Oriented Architecture” ή αλλιώς “υπηρεσιο-κεντρικής” αρχιτεκτονικής. Η υπηρεσιο-κεντρική αρχιτεκτονική ως αφηρημένη έννοια βασίζεται στη λογική ότι οποιοδήποτε μεγάλο πρόβλημα μπορεί να επιμεριστεί και να διαχειριστεί καλύτερα εάν το σπάσουμε σε μικρότερα προβλήματα τα οποία το συνθέτουν. ίδιο ισχύει και για συστήματα τα οποία βασίζονται σε υπηρεσίες ιστού(web services) και τα οποία ακολουθούν αρχιτεκτονική προσανατολισμένη στην ανάπτυξη τύπου SoA.[9]

Η αρχιτεκτονική SoA, δηλαδή η αρχιτεκτονική με “βάση τις υπηρεσίες”, αποτελεί την πιο σύγχρονη αρχιτεκτονική προσέγγιση όσον αφορά το σχεδιασμό και την ανάπτυξη σύνθετων πληροφοριακών συστημάτων στο Νέφος. Σαφώς οι υπηρεσίες μιας συλλογής πάνω στην οποία αναπτύσσεται ένα πληροφοριακό σύστημα πρέπει να επικοινωνούν μεταξύ τους με συγκεκριμένα πρωτόκολλα και να ακολουθούν κάποιους βασικούς κανόνες τους οποίους θα δούμε πιο κάτω.

2.2.4.2 Υπηρεσίες Γενικού σκοπού(Generic Enablers) και ειδικού Σκοπού (Specific Enablers)

Οι Generic Enablers δεν είναι τίποτα άλλο από υπηρεσίες κοινού σκοπού οι οποίες διανέμονται μέσω διαδικτύου με τη μορφή SaaS από παρόχους Υπολογιστικού Νέφους για την δημιουργία εφαρμογών. Η αρχιτεκτονική στην οποία υπακούουν τέτοιου είδους υπηρεσίες και συμβαδίζουν με τους κανόνες της, είναι η REST και ακολουθείται από ένα API το οποίο επιτρέπει την πρόσβαση σε αυτές. Τα βασικά χαρακτηριστικά ενός Generic Enabler είναι η απλότητά του και η εύκολη χρήση του οποιαδήποτε στιγμή ζητηθεί από το χρήστη. Τέτοιες επαναχρησιμοποιήσιμες υπηρεσίες είναι πολύ σημαντικές σ’ ένα Υπολογιστικό Νέφος καθώς διευκολύνουν τους προγραμματιστές στην ανάπτυξη πιο σύνθετων συστημάτων.

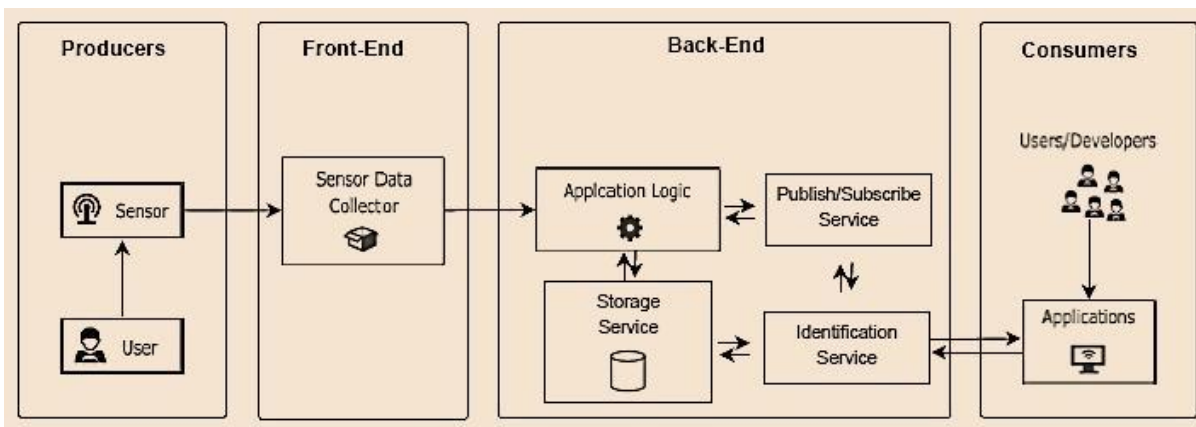
Η επικοινωνία και η σύνδεση παραπάνω του ενός Generic Enabler δημιουργεί έναν Specific Enabler που έχει ως στόχο την υλοποίηση μιας πιο σύνθετης λειτουργίας.. Σε

περίπτωση, που ο προγραμματιστής του Specific Enabler θελήσει να εντάξει μια νέα υπηρεσία, αυτό γίνεται πολύ εύκολα χωρίς να χρειαστεί να επέμβει στα επιμέρους κομμάτια/Generic Enablers που απαρτίζουν τον SE. Σημαντικό επίσης πλεονέκτημα κρίνεται και η αντικατάσταση μιας επιμέρους υπηρεσίας/Generic Enabler αν για παράδειγμα έχει διατεθεί μια καινούρια βελτιωμένη έκδοση της υπηρεσίας προς αντικατάσταση. Με αυτόν τον τρόπο ο SE παραμένει ενημερωμένος χωρίς να χρειαστεί να αλλάξει όλο το σύστημα εξολοκλήρου.

Σημαντικό γεγονός είναι ότι ο χρήστης δεν έχει επίγνωση για το πώς έχουν δημιουργηθεί αυτά τα μαύρα κουτιά(GEs SEs) αλλά μπορεί να τα χρησιμοποιήσει μόνο για τις υπηρεσίες που του προσφέρουν. Τέτοιου είδους «μαύρα κουτιά» διαθέτουν διάφοροι πάροχοι όπως η Google, Apple, Amazon Fiware δωρεάν ή με χρέωση για επιπλέον λειτουργικότητα.

2.2.4.3 Το Ειδικό Σχήμα Υπηρεσιοκεντρικής Αρχιτεκτονικής για τις ανάγκες της Εφαρμογής.

Το Νέφος αποτελεί το ιδανικό περιβάλλον για την ανάπτυξη υπηρεσιών(services) που μπορούν να διατεθούν για την ανάπτυξη πιο σύνθετων υπηρεσιών και ολοκληρωμένων υπολογιστικών συστημάτων. Όπως προαναφέραμε το Νέφος παρέχει ανεξαρτησία μεταξύ φυσικών πόρων και λειτουργικών συστημάτων-εφαρμογών ανεπτυγμένων πάνω σε αυτά. Η ανάπτυξη υπηρεσιών και εφαρμογών στηρίζεται στην ιδέα της ύπαρξης μικρών, γενικών και επαναχρησιμοποιήσιμων τμημάτων κώδικα που υλοποιούν βασικές λειτουργίες, γνωστά και ως Generic Enablers, τα οποία είναι προσβάσιμα από οποιοδήποτε υπολογιστικό σύστημα αρκεί να ακολουθείται ένα συγκεκριμένο πρωτόκολλο επικοινωνίας.



Σχήμα 5-Πρότυπη απεικόνιση της Υπηρεσιο-κεντρικής Αρχιτεκτονικής

Η λογική της υπηρεσιο-κεντρικής αρχιτεκτονικής θέλει διαφορετικούς παρόχους υπηρεσιών Νέφους να αναπτύσσουν εφαρμογές και υπηρεσίες, ο καθένας ακολουθώντας τη δικιά του υλοποίηση και τα δικά του μέσα (λειτουργικό σύστημα, γλώσσα προγραμματισμού, φυσικούς πόρους κλπ.) και οι υπηρεσίες αυτές να προσφέρονται μέσω διαδικτύου σε οργανισμούς που θέλουν να αναπτύξουν πιο σύνθετα υπολογιστικά συστήματα, ικανοποιώντας τις ανάγκες τους από τις ήδη υπάρχουσες υπηρεσίες που διατίθενται. Μια πρότυπη απεικόνιση Υπηρεσιο-κεντρικής Αρχιτεκτονικής η οποία ομαδοποιεί τις παρεχόμενες υπηρεσίες και τους GEs σε τέσσερα βασικά τμήματα (Παραγωγούς, Διεπαφή Χρήστη, Διεπαφή συστήματος και Καταναλωτές) εμφανίζεται στο Σχήμα '5'. Αριστερά είναι οι «Παραγωγοί» (Producers), στους οποίους περιλαμβάνονται οι αισθητήρες που παράγουν την πληροφορία. Αυτοί οι αισθητήρες μπορεί να αλληλεπιδρούν με χρήστες όπως συμβαίνει στους ιατρικούς αισθητήρες. Στο τμήμα «Διεπαφή Χρήστη» (Front-End) περιλαμβάνεται η υπηρεσία που παίζει το ρόλο μιας πύλης δικτύου ανάμεσα στα δεδομένα που στέλνει ο αισθητήρας και τα δεδομένα που διαχειρίζεται η εφαρμογή μέσω της «Διεπαφής Διαχείρισης Συστήματος» (Back-End). Στην «Διεπαφή Διαχείρισης Συστήματος» περιλαμβάνονται όλες εκείνες οι υπηρεσίες γενικού σκοπού για την ταυτοποίηση του χρήστη, δημιουργία συνδρομής, αποθήκευση των δεδομένων της εφαρμογής. Όλες αυτές οι υπηρεσίες επικοινωνούν με την Λογική Συστήματος (Application Logic), η οποία κάνει χρήση κανόνων, ελέγχων και συνθηκών για την μεταφορά της πληροφορίας στις επιμέρους υπηρεσίες. Τέλος, στο τμήμα «Καταναλωτές» περιλαμβάνονται είτε τελικοί χρήστες, είτε άλλες εφαρμογές που έχουν την δυνατότητα να αλληλοεπιδρούν με κάθε ενότητα του συστήματος.

2.2.4.4 Πρωτόκολλο επικοινωνίας HTTP και REST API

Το **HTTP** [10] (Hypertext Transfer Protocol) πρωτόκολλο επικοινωνίας αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client). Οι τέσσερις βασικότερες μέθοδοι που ορίζονται από το πρωτόκολλο αυτό είναι τύπου CRUD (Create – Read – Update – Delete) και αναφέρονται παρακάτω:

- **GET**, όταν ο πελάτης ζητάει ένα συγκεκριμένο πόρο (resource) από τον διακομιστή.
- **POST**, όταν ο πελάτης στέλνει δεδομένα στον διακομιστή π.χ. για εισαγωγή σε μια βάση δεδομένων.
- **PUT**, όταν ο πελάτης στέλνει δεδομένα για ένα ήδη υπάρχοντα πόρο (resource), ώστε να ανανεωθεί η τιμή του.
- **DELETE**, όταν ο πελάτης θέλει να διαγράψει έναν συγκεκριμένο πόρο από τον διακομιστή.

Αυτά όλα συμβαίνουν από την πλευρά του πελάτη όταν στέλνει αίτημα στον διακομιστή για μία συγκεκριμένη μέθοδο. Από την πλευρά του διακομιστή σε κάθε αίτημα που καλείται να επεξεργαστεί, αποκρίνεται στον πελάτη μ' έναν κωδικό κατάστασης ώστε να γνωρίζει αν για παράδειγμα βρέθηκε ο συγκεκριμένος πόρος που ζήτησε. Αυτοί οι κωδικοί αριθμοί χωρίζονται σε 5 μεγάλες κατηγορίες:

- Informational 1XX
- Successful 2XX
- Redirection 3XX
- Client Error 4XX
- Server Error 5XX

Το **REST** [11] είναι μια αρχιτεκτονική, η οποία αποτελείται από κανόνες και περιορισμούς για την δημιουργία λογισμικού και υπηρεσιών που φιλοξενούνται στο διαδίκτυο. Ο συγκεκριμένες υπηρεσίες επικοινωνούν μέσω του πρωτοκόλλου HTTP (Hypertext Transfer Protocol) με την χρήση των μεθόδων GET,POST,PUT,DELETE κ.α. που χρησιμοποιούνται για αιτήσεις από πελάτες σε εξυπηρετητές στο διαδίκτυο. Σε αυτό το μοντέλο, πελάτης-εξυπηρετητής, είναι βασισμένη η αρχιτεκτονική REST. Με την χρήση των μεθόδων οι πελάτες ξεκινούν αιτήματα (request) προ τους εξυπηρετητές, οι οποίοι με τη σειρά τους, επιστρέφουν τις ανάλογες απαντήσεις (response) κωδικοποιημένες συνήθως σε μορφή JSON ή XML.

Ένα **API** (Application Programming Interface) αποτελείται από ένα σύνολο οδηγιών και προτύπων προγραμματισμού για πρόσβαση σε υπηρεσίες που βρίσκονται στο διαδίκτυο. Οι προμηθευτές των υπηρεσιών προσφέρουν τα λεγόμενα APIs στο κοινό, έτσι ώστε οι προγραμματιστές να μπορούν να σχεδιάσουν εφαρμογές που να βασίζονται σε αυτά. Επιπλέον, τα APIs επιτρέπουν σε ένα πρόγραμμα να αλληλοεπιδρά με ένα άλλο, χωρίς να παρεμβάλλεται ο χρήστης. Για παράδειγμα, όταν μια διαδικτυακή εφαρμογή προσφέρει στον χρήστη την δυνατότητα εισαγωγής (login) του με λογαριασμό του Facebook, τότε η διαδικτυακή εφαρμογή κάνει χρήση του API που προσφέρει η εταιρεία Facebook για την ταυτοποίηση χρηστών. Με άλλα λόγια αναθέτουμε σε μια εξωτερική υπηρεσία (Facebook) την ταυτοποίηση των χρηστών της εφαρμογής μας για παροχή επιπλέον ασφάλειας, αξιοπιστίας και ευκολίας στον χρήστη.

2.2.2.5 JSON (JavaScript Object Notation) και XML (Extensible Markup Language)

Στην παρούσα εργασία ιδιαίτερο ενδιαφέρον έχει η δυναμική διαχείριση, αποθήκευση και μεταφορά ευαίσθητων δεδομένων. Για τον λόγο αυτόν ήταν αναγκαία η χρήση κατάλληλων τεχνολογιών. Παρακάτω αναφέρουμε συνοπτικά τις 2 τεχνολογίες-γλώσσες προγραμματισμού οι οποίες βοήθησαν στην διεκπεραίωση της εφαρμογής, καθώς και μια σύντομη σύγκρισή τους.

- **XML** σε μια πρόταση είναι «μια ομάδα προτύπων για την ανταλλαγή και δημοσίευση πληροφοριών με ένα δομημένο τρόπο». Η έμφαση στην δομή δεν μπορεί να υποτιμηθεί. Η XML αποτελεί μια γλώσσα που χρησιμοποιείται στην περιγραφή και τον χειρισμό δομημένων εγγράφων. Τα XML έγγραφα δεν περιορίζονται σε βιβλία και άρθρα ή ακόμη και ιστοσελίδες και μπορεί να περιλαμβάνουν αντικείμενα σε μια εφαρμογή πελάτη/εξυπηρετητή. Η XML προσφέρει την ίδια δενδροειδή δομή σε όλες τις παραπάνω εφαρμογές. Η XML δεν υποδεικνύει ή επιβάλλει τις λεπτομέρειες αυτής της δομής.
- **JSON** είναι ένα ανοιχτό πρότυπο σχεδιασμένο για δεδομένα αναγνωρίσιμα από άνθρωπο. Προέρχεται από την γλώσσα JavaScript και πιο συγκεκριμένα από την αναπαράσταση απλών τύπων δεδομένων και συσχετισμένων(associative) πινάκων. Παρά τη σχέση της με την JavaScript είναι ανεξάρτητη από την γλώσσα προγραμματισμού. Χρησιμοποιείται για την αποθήκευση και μεταφορά των δεδομένων όπως ακριβώς και η XML

Το ερώτημα είναι ποιο πρότυπο είναι επικρατέστερο στην ανταλλαγή δεδομένων. Αρχικά ή κύρια χρήση του προτύπου JSON είναι για την ανταλλαγή δεδομένων μεταξύ browsers και Servers , και η χρήση του έξω από το περιβάλλον του web είναι αρκετά σπάνια. Το πρότυπο XML χρησιμοποιείται και αυτό αποτελεσματικά για την παράδοση δεδομένων μεταξύ browser και server αλλά έχει επιπλέον χρήση για την εξαγωγή χρήσιμης πληροφορίας από μία βάση δεδομένων(database).[12]

Τα δύο επικρατέστερα πλεονεκτήματα του JSON σε σχέση με τον XML είναι η ταχύτητα και ευκολία όπου το ίδιο αναλύεται(parsing) , καθώς και το απλό σε ανθρώπινο μάτι συντακτικό του. Επιπλέον παρέχει μεγάλη ευκολία ανάκτησης δεδομένων από ένα αντικείμενο JavaScript. Στον αντίποδα της παραπάνω θέσης όταν μας ενδιαφέρει η αναζήτηση(querying) και οι αλλαγές προτύπων(format changes) τότε το XML είναι καθαρός νικητής.

2.3 Το Πρωτόκολλο Bluetooth Low Energy (BLE)

2.3.1 Ιστορικά στοιχεία και σχέση με το IOT

Το Bluetooth low energy είναι ένα πρότυπο δικτύωσης συσκευών χαμηλής κατανάλωσης και χαμηλού χρόνου αναμονής, σε δίκτυα εμβέλειας (εως και 50 μέτρα). Λειτουργεί στην ελεύθερη ζώνη συχνοτήτων 2.4 GHz , και χρησιμοποιεί ένα πομποδέκτη συχνότητας για την αντιμετώπιση παρεμβολών εξασθένισης.

Η τεχνολογία του Bluetooth Low Energy πρωτοεμφανίστηκε το 2006 με την ονομασία Wibree από την εταιρία παραγωγής προϊόντων τηλεπικοινωνίας Nokia. Οι ερευνητές της Nokia, ήδη από το 2001, είχαν ξεκινήσει μια προσπάθεια να καθορίσουν περιπτώσεις χρήσης, επαρκείς και νέες, για τις οποίες οι υπάρχουσες τεχνολογίες ασύρματης μετάδοσης δεν επαρκούσαν ή ήταν δύσχρηστες. Η προσπάθεια αυτή κατέληξε το 2004 σε ένα project με την ονομασία Bluetooth Low End Extension, το οποίο χρησιμοποιούσε στοιχεία από το πρότυπο του Bluetooth, με τελικό σκοπό τη δημιουργία μιας ασύρματης τεχνολογίας με τη μικρότερη δυνατή κατανάλωση ενέργειας. Αποτέλεσμα όλων αυτών ήταν το Wibree. Λιγότερο από ένα χρόνο μετά την κυκλοφορία του, όμως, η Nokia, η οποία είναι και ιδρυτικό μέλος του Bluetooth SIG, αποφάσισε να δώσει την τεχνολογία του Wibree στο SIG, προκειμένου αυτή να περιληφθεί στο μελλοντικό πρότυπο του Bluetooth, χρησιμοποιώντας το ήδη γνωστό εμπορικό σήμα και υπό τον έλεγχο του SIG. Έτσι, το 2010, με την κυκλοφορία της έκδοσης 4.0 του Bluetooth, το Bluetooth Low Energy εμφανίστηκε ως μία πολύ χαμηλής κατανάλωσης και χαμηλού κόστους ασύρματη τεχνολογία, με την εμπορική ονομασία Bluetooth Smart.

Εκτός από το Bluetooth Smart, υπήρχαν ήδη στην αγορά και ανταγωνιστικές αντίστοιχες τεχνολογίες, όπως το ANT ή το Zigbee. Ασχέτως των δυνατοτήτων και περιορισμών της κάθε τεχνολογίας, το Bluetooth Smart έχει το μεγάλο πλεονέκτημα ότι υποστηρίζεται από ένα οργανισμό όπως το SIG, ο οποίος έχει μεγάλο αριθμό μελών, και στον

οποίο συμμετέχουν όλοι οι μεγάλοι κατασκευαστές software και hardware, ενώ χρησιμοποιεί το ήδη ευρέως διαδεδομένο και δημοφιλές Bluetooth brand name. Ένα σημαντικό γεγονός για το Bluetooth Smart ήταν η υποστήριξή του από το iPhone 4S, που κυκλοφόρησε τον Οκτώβριο του 2011, το πρώτο smartphone με υποστήριξη της έκδοσης 4.0 του Bluetooth. Έτσι, άρχισαν να εμφανίζονται στην αγορά και τα πρώτα Bluetooth Smart gadgets, ενώ όλο και περισσότερα smartphones υποστήριζαν το νέο πρωτόκολλο. Στο λειτουργικό σύστημα Android, η επίσημη υποστήριξη ξεκίνησε σχετικά πρόσφατα με την έκδοση 4.3, ενώ κάποιοι κατασκευαστές παρείχαν ήδη υποστήριξη με δικές τους βιβλιοθήκες. Με την κυκλοφορία των Windows 8, η Microsoft παρέχει επίσης υποστήριξη για τη νέα τεχνολογία, ενώ, πλέον, πολλοί σταθεροί ή φορητοί υπολογιστές πωλούνται με ενσωματωμένο Bluetooth 4.0 hardware. Οι προοπτικές για την εξέλιξη του Bluetooth Smart είναι πολύ ευοίωνες και αναλυτές εκτιμούν ότι το πλήθος των συσκευών με υποστήριξη του προτύπου αναμένεται να εκτοξευθεί τα επόμενα χρόνια.

Όπως έγινε φανερό, η κυκλοφορία του Bluetooth Smart συνέπεσε με την άνθιση της αγοράς των smartphones και tablets, ενώ καινούρια προϊόντα μπορούσαν πλέον να δημιουργηθούν, με βάση τη νέα τεχνολογία, τα οποία δεν ήταν πριν εφικτά. Έτσι, εμφανίζονται έννοιες όπως φορητές συσκευές (wearable's), ή appcessory, ένας νεολογισμός που προκύπτει από το συνδυασμό των λέξεων app, που αναφέρεται σε μια εφαρμογή που τρέχει σε κάποιο

smartphone ή tablet, και accessory. Ουσιαστικά, πρόκειται για μικρού μεγέθους συσκευές, οι οποίες, λόγω χαμηλής κατανάλωσης, μπορούν να λειτουργούν με μια μπαταρία ακόμα και για

χρόνια. Οι συσκευές αυτές, από μόνες τους, δεν έχουν μεγάλη χρησιμότητα (π.χ. δεν έχουν User interface), όμως, σε συνδυασμό με μια εφαρμογή σε κάποιο smartphone, αποκτούν χρηστικότητα που δε θα ήταν αλλιώς δυνατή. Μάλιστα, πολλές διαφορετικές συσκευές μπορούν να συνδυαστούν σε νέες περιπτώσεις χρήσης, δίνοντας έτσι ακόμα μεγαλύτερη ευχρηστία σε διάφορα και διαφορετικά μεταξύ σενάρια χρήσης.

Οι δυνατότητες βέβαια δε σταματούν εδώ. Υπάρχουν ήδη εφαρμογές στον τομέα της υγείας, όπως θερμομέτρα, πιεσόμετρα και μετρητές γλυκόζης με Bluetooth Smart. Επίσης, η δυνατότητα αποστολής εντολών προς τις συσκευές, αλλά και από τις συσκευές, ανοίγει νέες προοπτικές σε τομείς όπως το Internet of Things. Με τη βοήθεια του BLE, μπορούμε να έχουμε πολλές τέτοιες συσκευές, μικρού μεγέθους και κόστους και με ελάχιστη κατανάλωση ενέργειας, οι οποίες, συνδυαζόμενες και ελεγχόμενες από κάποιο κεντρικό σημείο, παρέχουν στον χρήστη δυνατότητες ελέγχου, τοπικά ή απομακρυσμένα, και, κυρίως, αυτοματισμού ενεργειών.

2.3.2 Βασικά χαρακτηριστικά και σχεδιαστικές επιλογές του BLE

Το Bluetooth Low Energy σχεδιάστηκε εξ αρχής με γνώμονα τη μείωση της κατανάλωσης ενέργειας. Ένας άλλος παράγοντας που ελήφθη υπόψη ήταν το χαμηλό κόστος, αφού στόχος της τεχνολογίας ήταν να χρησιμοποιηθεί σε φτηνά προϊόντα μαζικής παραγωγής. Αν το κόστος ήταν απαγορευτικό για τον τελικό χρήστη, η χρησιμότητα της τεχνολογίας θα ήταν περιορισμένη. Το βασικό όμως χαρακτηριστικό και ο βασικός στόχος ήταν η ελάχιστη δυνατή κατανάλωση ενέργειας. Εδώ πρέπει να αναφέρουμε ότι και το Bluetooth Classic θεωρούνταν τεχνολογία χαμηλής κατανάλωσης ενέργειας, όμως, με το BLE, αυτό το χαρακτηριστικό πέρασε σε νέο επίπεδο. Εκεί, δηλαδή, που μια συσκευή Bluetooth Classic μπορεί να λειτουργεί με μπαταρία για μέρες ή εβδομάδες, μια συσκευή BLE μπορεί να λειτουργεί με μικρότερη μπαταρία για μήνες ή χρόνια.

Όπως και στην περίπτωση του Bluetooth Classic, το πρότυπο ορίζεται σε ολόκληρη τη στοίβα πρωτοκόλλων που πρέπει να υλοποιείται από τις συμβατές συσκευές. Με τον τρόπο υπάρχει μεγαλύτερος έλεγχος πάνω στις συμβατές συσκευές, κάτι το οποίο βοηθά στη διαλειτουργικότητα(interoperability) των επιλογών του σχεδιαστή, ενώ, στην περίπτωση του BLE, σημαίνει ότι όλα τα στρώματα του πρωτοκόλλου μπορούν να βελτιστοποιηθούν ως προς την κατανάλωση ενέργειας. Θεμιτή είναι η αναφορά ορισμένων χαρακτηριστικών του BLE.

- **Απλότητα:** Σε αντίθεση με το Bluetooth Classic, που είναι ένα αρκετά πολύπλοκο πρωτόκολλο επικοινωνίας, το BLE έχει απλοποιήσει πολλές από τις διαδικασίες. Η απλότητα αυτή σημαίνει χαμηλή κατανάλωση ενέργειας, αφού απαιτείται εκτέλεση λιγότερων εντολών από το micro-processor της συσκευής, αλλά και χαμηλότερο κόστος, αφού η δημιουργία του απαραίτητου υλικο-λογισμικού(firmware) είναι ευκολότερη και το ίδιο το firmware μικρότερο σε μέγεθος, απαιτώντας λιγότερη μνήμη.
- **Εξισορρόπηση υπολογιστικής ισχύος :** Πρωταρχικό ρόλο σε αυτό το χαρακτηριστικό του BLE είναι η εγγενής ασυμμετρία που παρουσιάζει το πρότυπο όσο αναφορά τους χρήστες ομότιμης ομάδας(peers).Μια φυσική ζεύξη δημιουργείται μεταξύ δύο συσκευών κυρίαρχου(master) και υπηρέτη(slave). Πολλές από τις διαδικασίες που απαιτούνται από το πρωτόκολλο εκτελούνται στον master. Η λογική είναι ότι ο master είναι συνήθως μια συσκευή τύπου smartphone ή tablet, η οποία έχει μεγάλη υπολογιστική ισχύ και λειτουργεί με επαναφορτιζόμενη μπαταρία υψηλής χωρητικότητας. Αντίθετα, ο slave, είναι συνήθως μια συσκευή μικρού μεγέθους και υπολογιστικής ισχύος, η οποία λειτουργεί με μια τυπική μπαταρία του εμπορίου.
- **Οικονομία σε κατανάλωση ενέργειας:** Οι συσκευές BLE λειτουργούν ακόμα και για χρόνια με μπαταρίες τύπου coin cell.
- **Ισχυρή ασφάλεια :**Πλήρης κρυπτογράφηση AES - 128²³ χρησιμοποιώντας CCM²⁴ για την παροχή ισχυρής κρυπτογράφησης και έλεγχου ταυτότητας των πακέτων δεδομένων

2.3.3 Πρωτόκολλο Χαρακτηριστικών (Attribute Protocol) και Γενικό Προφίλ Χαρακτηριστικών (Generic Attribute Profile Protocol)

Το Attribute Protocol (ATT) και το Generic Attribute Profile (GATT) είναι τα σημαντικότερα τμήματα της στοίβας πρωτοκόλλων του Bluetooth Low Energy, σε ό,τι αφορά

τους developers BLE εφαρμογών. Είναι τα τμήματα εκείνα, πάνω στα οποία είναι χτισμένα τα APIs στις περισσότερες πλατφόρμες που υποστηρίζουν το BLE, και, ως εκ τούτου, τα τμήματα με τα οποία έρχονται σε άμεση επαφή οι developers. Τα ATT και GATT αποτελούν το μοντέλο δεδομένων του BLE, πάνω στο οποίο μπορούν να δημιουργηθούν οι διάφορες εφαρμογές και περιπτώσεις χρήσης. Το ATT ορίζει τις βασικές έννοιες και τα δομικά στοιχεία του μοντέλου δεδομένων. Πάνω σε αυτά το GATT χτίζει μια ιεραρχία από δομές δεδομένων, δίνοντας δομή και σημασιολογία στην επίπεδη οργάνωση του ATT.

²³ https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

²⁴ https://en.wikipedia.org/wiki/CCM_mode

2.2.3.1 Πρωτόκολλο Χαρακτηριστικών(Attribute Protocol)

Το Attribute Protocol (ATT) ορίζει μια βασική δομή, το attribute, και τις διαδικασίες με τις οποίες μία συσκευή μπορεί να ανακαλύψει τα attributes, που βρίσκονται αποθηκευμένα σε μια απομακρυσμένη συσκευή, και, στη συνέχεια, να τα διαβάσει, να τα μεταβάλει ή να δεχτεί ειδοποιήσεις σχετικά με αυτά. Οι δύο αυτές συσκευές αποτελούν και τους δύο ρόλους που ορίζονται στο πρωτόκολλο, δηλαδή τους server και client. Ο ATT server είναι η συσκευή που περιέχει τα attributes, ενώ ο ATT client είναι η συσκευή που επικοινωνεί με το server, χρησιμοποιώντας το Attribute Protocol, προκειμένου να εκτελέσει τις παραπάνω λειτουργίες.

Ένα attribute έχει την εξής δομή

Handle	Type	Value	Permission
(16-bits)	(128 bit)	(0...512 bytes)	

- **Attribute Value:** Η τιμή ενός attribute είναι ένας πίνακας από bytes, που μπορεί να έχει σταθερό ή μεταβλητό μέγεθος, με μέγιστο τα 512 bytes. Τα δεδομένα που αποθηκεύονται στην τιμή του attribute είναι αδιάφορα για το πρωτόκολλο, και ορίζονται από ανώτερα επίπεδα, συνήθως σε συνάρτηση με έναν μοναδικά αποκωδικοποιήσιμο αριθμό(UUID).
- **Attribute handle :** Είναι μία τιμή 16 bit, η οποία χρησιμεύει ως αναγνωριστικό για το

συγκεκριμένο attribute. Κάθε attribute έχει το δικό του μοναδικό handle. Τα attribute handles χρησιμοποιούνται σε όλες τις λειτουργίες του πρωτοκόλλου, όπου απαιτείται η επιλογή κάποιου attribute. Κατά μία έννοια λειτουργούν ως διευθύνσεις των attributes. Το μικρό μέγεθος των handles βοηθάει στη μείωση του κόστους επικοινωνίας

- **Attribute Type:** Είναι τύπος ενός attribute καθορίζεται με τη βοήθεια ενός Universally Unique Identifier (UUID). Τα UUID είναι αναγνωριστικοί αριθμοί των 128 bit (16 bytes), οι οποίοι παράγονται με τέτοιο τρόπο, ώστε είναι δύσκολο να δημιουργηθούν όμοια UUIDs, για αυτό και θεωρούνται μοναδικοί. Χρησιμοποιούνται σε πολλά

πρωτόκολλα και περιπτώσεις χρήσης. Συνήθως αναπαρίστανται, με τη λεγόμενη κανονική μορφή, ως 32 δεκαεξαδικά ψηφία, χωρισμένα με παύλες σε 5 ομάδες (8-4-4-4-12). Για παράδειγμα 11223344-5566-7788-99aa-bbccddeeff00.

- **Permissions:** Τα permissions ενός attribute ορίζουν τις επιτρεπτές ενέργειες που μπορούν να γίνουν πάνω σε αυτό. Τα attribute permissions είναι ένας συνδυασμός από permissions πρόσβασης (access), κρυπτογράφησης (encryption), πιστοποίησης αυθεντικότητας (authentication) και εξουσιοδότησης (authorization).

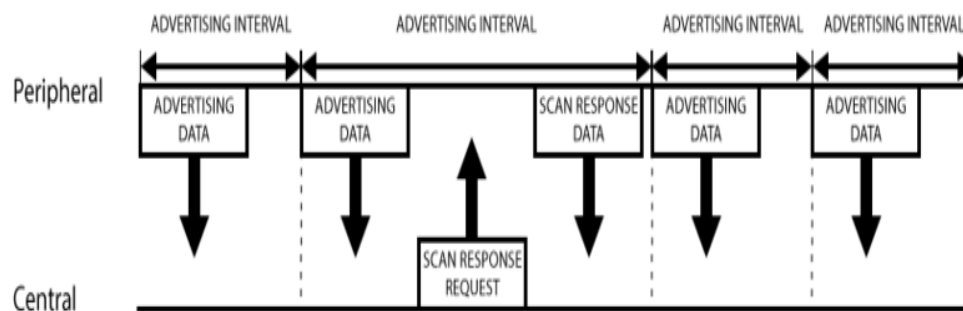
2.3.3.2 Γενικό προφίλ πρόσβασης (Generic Access Profile)

Το Generic Access Profile (GAP) ορίζει ρόλους, μέσω των οποίων δύο, συμβατές με το πρότυπο, συσκευές μπορούν ή και να μην μπορούν να ανακαλύψουν η μία την άλλη, να συνδεθούν και να ανταλλάξουν δεδομένα. Το GAP ορίζει διάφορους ρόλους για τις συσκευές όμως οι πιο σημαντικοί για την εκπλήρωση της παρούσας εργασίας ήταν οι Κεντρικές συσκευές (Central devices) καθώς και οι περιφερειακές συσκευές (Peripheral devices).

- **Peripheral devices:** Είναι μικρές συσκευές-αισθητήρες, χαμηλής κατανάλωσης που μπορούν να συνδεθούν σε μία μεγάλης ισχύος περιφερειακή συσκευή. Η συσκευή αυτή απαιτείται να έχει κυκλώματα αποστολής και λήψης δεδομένων. Τέτοιες συσκευές είναι αισθητήρες τύπου ανίχνευσης καρδιακών παλμών, οξυγόνου στο αίμα και πολλές ακόμα.
- **Central devices:** Είναι η κεντρικές συσκευές μεγαλύτερης επεξεργαστικής ισχύος και μνήμης. Προφανώς και τα Central devices πρέπει να διαθέτουν και αυτά απαραίτητα κυκλώματα αποστολής και λήψης δεδομένων. Τέτοιου είδους συσκευές είναι συνήθως τα tablets και smartphones.

Για να πραγματοποιηθεί μια επιτυχής επικοινωνία μεταξύ μιας κεντρικής συσκευής με μια περιφερειακή συσκευή θα πρέπει αρχικά να δημιουργηθεί μια σύζευξη μεταξύ τους. Για τον λόγο αυτό οι περιφερειακές συσκευές πραγματοποιούν διαδικασίες «**διαφήμισης**»(advertising) και «**ανίχνευσης απόκρισης πληροφορίας**»(Scan Response Data). ούτως ώστε να μπορέσουν να ενημερώσουν την κεντρική συσκευή για βασικές πληροφορίες ή οποία με την σειρά της ενημερώνεται μέσω ενός πακέτου αιτήματος(Scan Response Data Request) προς την περιφερειακή συσκευή. Και οι δύο διαδικασίες είναι ταυτόσημες καθώς αποστέλλουν κάθε φορά πακέτα πληροφορίας μεγέθους 31bytes. Παρόλα αυτά η ύπαρξη της διαδικασίας advertising είναι

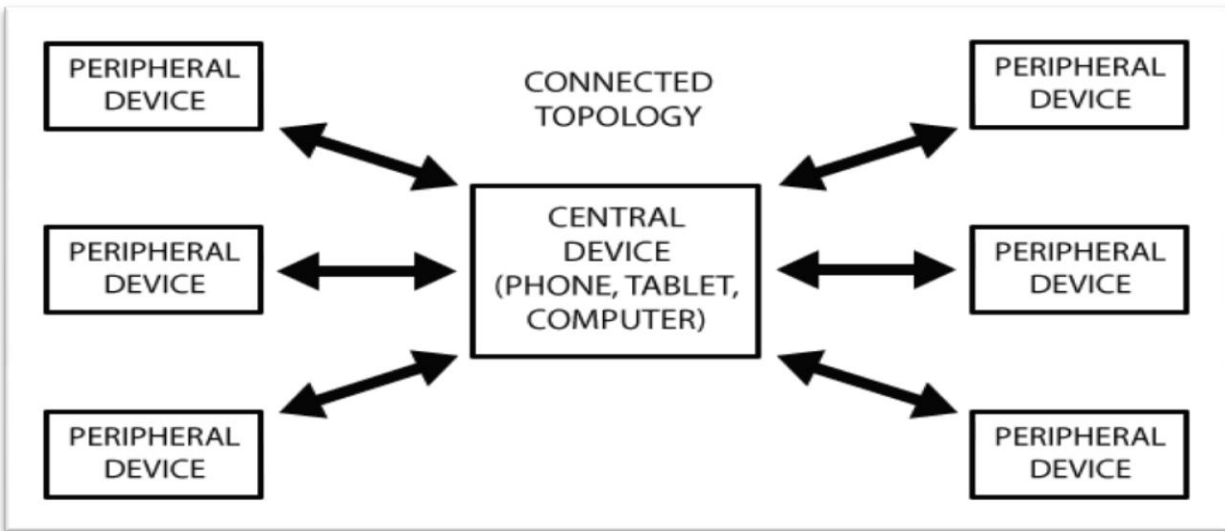
υποχρεωτική καθώς ενεργοποιείται συνεχώς με την περιοδική αποστολή πακέτων προς την κεντρική συσκευή, ώστε να γίνει αντιληπτή η ύπαρξη των περιφερειακών συσκευών ως προς την κεντρική συσκευή. Η διαδικασία Scan Response Data έρχεται σε δεύτερο στάδιο προσφέροντας επιπλέον χρήσιμη πληροφορία στην κεντρική συσκευή όπως συμβολοσειρές για το μοναδικό όνομα της συσκευής ή το όνομα του κατασκευαστή της κτλ. ροή λειτουργίας τον παραπάνω φαίνεται και στο Σχήμα 6



Σχήμα 6- Ροή διαδικασίας advertising μεταξύ Κεντρικής και περιφερειακής συσκευής

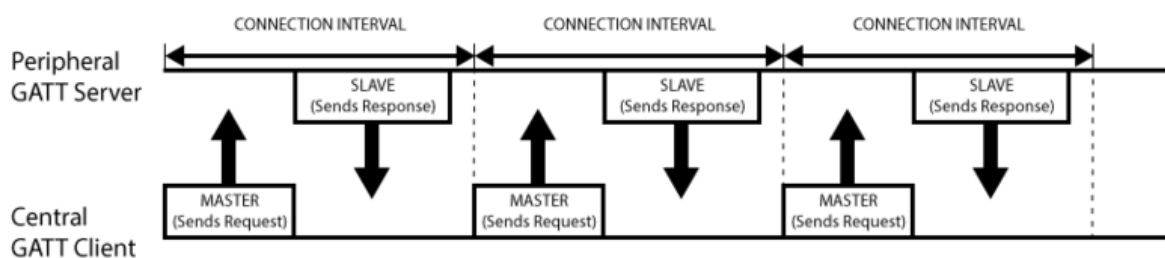
2.2.3.3 Γενικό Προφίλ Χαρακτηριστικών (Generic Attribute Profile)

Το Generic Attribute Profile ορίζει τον τρόπο με τον οποίο δύο συσκευές η οποίες ακολουθώντας το πρωτόκολλο BLE μεταφέρουν δεδομένα μεταξύ τους αφού πλέον έχουν ήδη συζευχθεί μεταξύ τους. Δηλαδή έχει περάσει η προαναφερόμενη διαδικασία του GAP όπου η περιφερειακή συσκευή «διαφημίζει» τα πακέτα της ως προς την κεντρική συσκευή. Όπως στο ATT, έτσι και στο GATT, έχουμε δύο ρόλους, τον server και τον client. Ο ρόλος κάθε peer στο GATT είναι ο ίδιος με αυτόν στο ATT. Δηλαδή, ο GATT server είναι ATT server και ο GATT client είναι ATT client. Το πιο σημαντικό γεγονός στην περίπτωση του GAAT είναι ότι μια περιφερειακή BLE Συσκευή μπορεί να συνδεθεί μόνο σε ένα περιφερειακό χωρίς να ισχύει το αντίστροφο. Μια περιφερειακή συσκευή μόλις συνδεθεί με μια κεντρική συσκευή τότε σταματάει να «διαφημίζει» τα πακέτα της προς άλλες κεντρικές συσκευές μέχρι να διακοπεί η σύνδεσή της. Ωστόσο μόλις επιτευχθεί η σύνδεση μεταξύ των συσκευών τότε η επικοινωνία είναι αμφίδρομη καθώς η κεντρική συσκευή μπορεί να ζητήσει και να στείλει δεδομένα (read-write) στις περιφερειακές συσκευές με τις οποίες έχει συνδεθεί.[13]



Σχήμα 7-Τοπολογία σύνδεσης GAAT

Πλέον οι περιφερειακές συσκευές είναι GAAT servers και οι κεντρικές συσκευές GATT clients . Καθώς ανά τακτά χρονικά διαστήματα (Interval) γίνεται ανταλλαγή πακέτων μεταξύ τους.

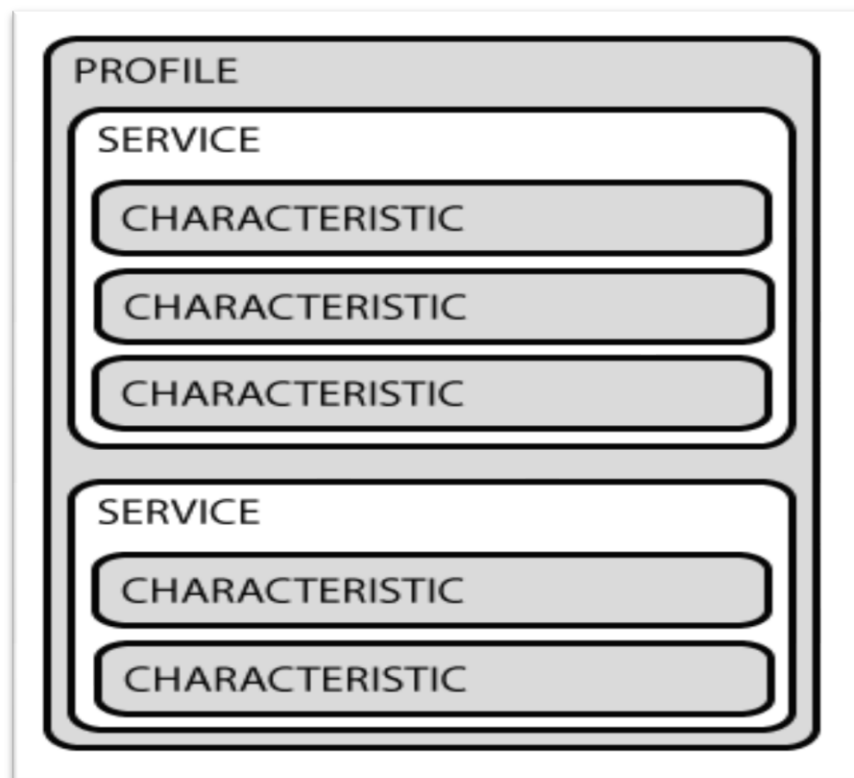


Σχήμα 8- Ροή διαδικασίας GAAT συναλλαγής πακέτων(GAAT Server-Client)

Το GAAP χρησιμοποιεί τα attributes, που παρέχονται από το ATT, οργανώνοντας τα σε μια ιεραρχία από δομές δεδομένων. Στη ρίζα της ιεραρχίας του GATT βρίσκεται το προφίλ (profile). Ένα προφίλ αποτελείται από μία ή περισσότερες υπηρεσίες (services). Κάθε υπηρεσία

αποτελείται από ένα ή περισσότερα χαρακτηριστικά (characteristics), ενώ μπορεί να ορίζει σχέσεις με άλλες υπηρεσίες. Κάθε χαρακτηριστικό περιέχει μία τιμή, ενώ μπορεί, επίσης, να περιέχει και μετα-δεδομένα σχετικά με την τιμή του, με τη μορφή περιγραφών (descriptors). Οι υπηρεσίες, τα χαρακτηριστικά και οι περιγραφείς αποθηκεύονται όλα σε attributes στον GATT server.

Αν δούμε την ιεραρχία του GATT με βάση το αντικειμενοστραφές μοντέλο, μία υπηρεσία είναι παρόμοια με μία κλάση, ενώ τα χαρακτηριστικά είναι τα δεδομένα που ενθυλακώνονται από αυτή. [14]



Σχήμα 9 Μοντέλο δεδομένων GAAT

- **Προφίλ(Profile):** Ουσιαστικά το προφίλ δεν υπάρχει στην ίδια την BLE συσκευή, και αναφορά του γίνεται για μια αφαιρετική έννοια πατρικής δομής όπου εμπεριέχει υπηρεσίες. Για παράδειγμα το προφίλ καρδιακών παλμών(Heart

Rate Profile ²⁵) συνδυάζει την υπηρεσία του Υπηρεσίας καρδιακών παλμών(Heart Rate Service) και την υπηρεσία πληροφορίας συσκευής(Device Information Service).Μια λίστα με τα ολοκληρωμένα GAAT-based μπορούν να προβληθούν στον σύνδεσμο <http://adafru.it/dih>.

- **Υπηρεσίες(Services)** : Οι υπηρεσίες χρησιμοποιούνται για να χωρίσουν την πληροφορία σε λογικά κομμάτια, και να τα κομμάτια αυτά να εξειδικεύονται σε χαρακτηριστικά της συσκευής. Κάθε υπηρεσία χαρακτηρίζεται από ένα μοναδικό αριθμό UUID το οποίο είναι 16 bit για τις επίσημες υπηρεσίες της SIG και 128 bit για υπηρεσίες που ορίζουν άλλοι προγραμματιστές. Μια υπηρεσία μπορεί να περιέχει από ένα η και περισσότερα χαρακτηριστικά. Για παράδειγμα η υπηρεσία καρδιακού ρυθμού(Heart Rate Service ²⁶) περιέχει τη τιμή καρδιακών παλμών(heart rate measurement), την τοποθεσία στο σώμα όπου έχει τοποθετηθεί ο αισθητήρας(Body sensor location-προκαθορισμένο),και το κεντρικό σημείο ελέγχου καρδιακού παλμού (heart rate control point-δεν υποστηρίζεται πάντα).
- **Χαρακτηριστικά(Characteristics)**: Ίσως το σημαντικότερο τμήμα της δομής GAAT.Τα χαρακτηριστικά και αυτά χαρακτηρίζονται από μοναδικά UUID τα οποία μπορεί να είναι 16 bit ή 128 bit και περιέχουν τιμές και πολλές φορές περιγραφής τιμών. Για παράδειγμα το χαρακτηριστικό Heart Rate Measurement το οποίο κυρίως θα μας απασχολήσει την εργασίας μας ξεκινάει με μια τιμή 8 bit(UINT8 ή UINT16) όπου δίνει την τιμή μέτρηση σε bpm και χρησιμοποιεί ένα UUID τύπου **0x2A37** .

²⁵ <http://adafru.it/ddy>

²⁶ <http://adafru.it/ddB>

2.4 Λογισμικό Android

2.4.1 Το λογισμικό Android και ιστορικά στοιχεία

Ένα από τα πιο διαδεδομένα λειτουργικά συστήματα σήμερα είναι το Android. Το Android αποτελεί ένα λειτουργικό σύστημα που συναντάται κυρίως σε ενσωματωμένα συστήματα, όπως είναι τα κινητά τηλέφωνα, τα tablet αλλά και άλλες κινητές συσκευές.



Σχήμα 10-Λογότυπο Android

Το Android αρχικά αναπτύχθηκε από τη Google και αργότερα από την Open Handset Alliance. Ο πυρήνας του Android είναι βασισμένος στον πολύ γνωστό και αρκετά δοκιμασμένο πυρήνα του λειτουργικού Linux. Κληρονομώντας όλα τα χαρακτηριστικά ασφαλείας του Linux και όλες τις διαχειριστικές τεχνικές μνήμης και επεξεργαστή που αυτό διαθέτει, το Android, καθίσταται ένα αρκετά αξιόπιστο λειτουργικό σύστημα.[15] Επιπλέον λόγω του ότι το Android είναι λογισμικό ανοιχτού κώδικα, ο καθένας μπορεί να το προσαρμόσει στις δικές του ανάγκες και στο δικό του υλικό. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από τη Google. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας σύμπραξης εταιρειών-κολοσσών στον τομέα της τεχνολογίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Από τις 30 Απριλίου 2009, μέσω κατάλληλων διεπαφών, και μέσω του Application Framework μπορούν με τη σειρά τους να παρέχουν επιπρόσθετε λειτουργίες-υπηρεσίες προς άλλες εφαρμογές, εφόσον κάτι τέτοιο φυσικά δεν περιορίζεται από τις πολιτικές ασφάλειας του Application Framework. Μερικές από τις πιο βασικές οντότητες που περιλαμβάνονται στο πλαίσιο του Application Framework είναι:

1)View system: Επιτρέπει την χρήση λιστών, πλαισίων, πεδίου κειμένου, κουμπιών κτλ.

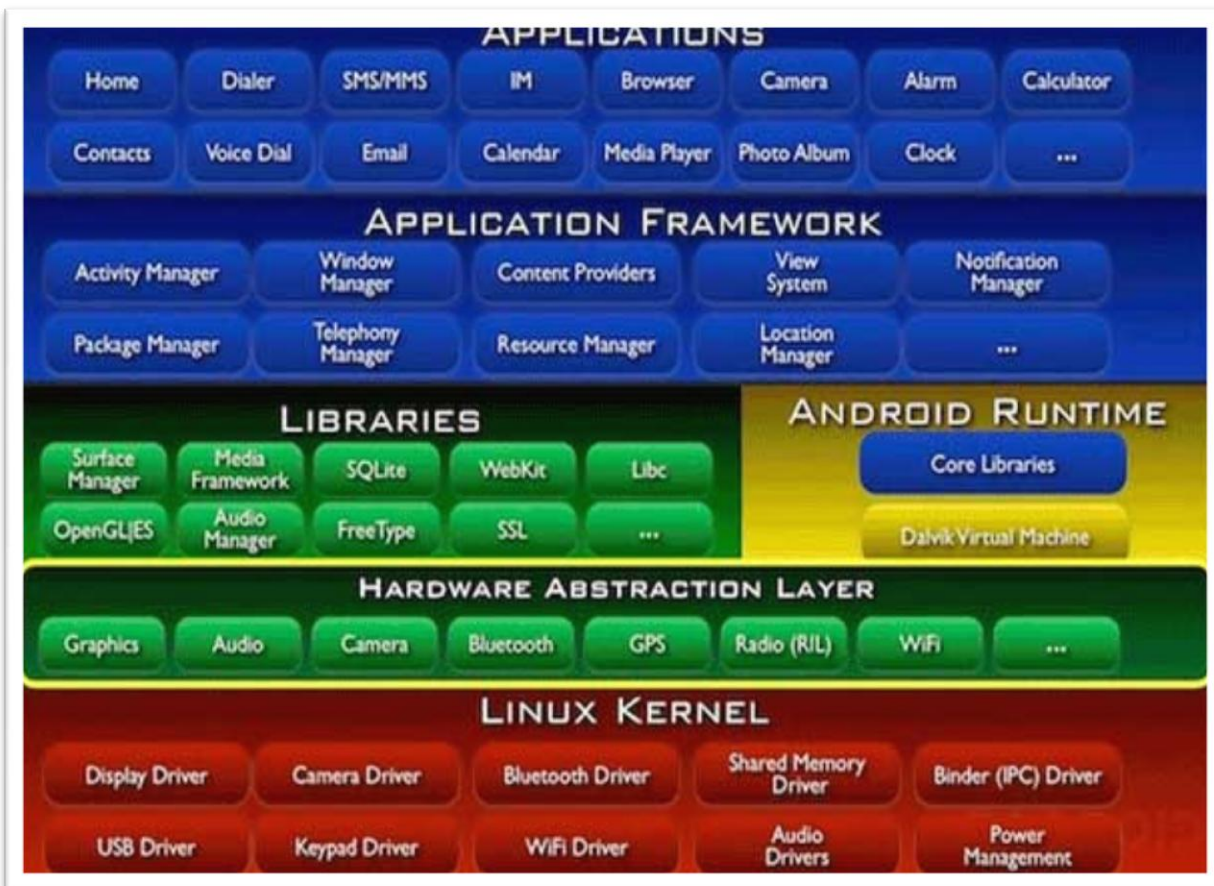
άλλων

2)Content Provider: Επιτρέπει στις εφαρμογές την πρόσβαση σε δεδομένα εφαρμογών ή τον διαμερισμό των δικό τους δεδομένων , όπως οι επαφές.

3)Resource Manager : Παρέχει την πρόσβαση σε πόρους όπως γραφικά και σε αρχεία σχετικά με την διάταξη των στοιχείων του γραφικού περιβάλλοντος

4)Activity manager: Διαχειρίζεται το κύκλο ζωής μιας δραστηριότητα(όπως θα αναλύσουμε παρακάτω) και παρέχει την δυνατότητα μετάβασης στις προγενέστερες κατάστασης τους.

- **Applications:** Στο ανώτερο από όλα επίπεδο Applications βρίσκονται οι εφαρμογές που θα χρησιμοποιήσουν τελικά οι χρήστες με διαφάνεια ως προς το τι συμβαίνει πίσω από αυτές ή το τι απαιτείται για την εκτέλεση τους από το λειτουργικό σύστημα. Όλες οι εφαρμογές είναι γραμμένες σε Java και μπορούν να τρέχουν πολλές ταυτόχρονα χωρίς να επηρεάζει η μια την άλλη.



Σχήμα 11 – Γενική αρχιτεκτονική λογισμικού Android

2.4.3 Βασικά συστατικά για την δημιουργία εφαρμογής σε Android

Το android αποτελείται από διάφορους τύπους συστατικών .Τα πιο σημαντικά κομμάτια συστατικών που χρησιμοποιήθηκαν στην παρούσα εργασία είναι **οι Δραστηριότητες (Activities)** οι **Υπηρεσίες (Services)** και των δεκτών εκπομπής (**Broadcast Receivers**)?

2.4.3.1 Δραστηριότητες (Activities)

Ένα activity αντιπροσωπεύει μία οθόνη της εφαρμογής, η οποία παρουσιάζει στον χρήστη ένα user interface (UI). Είναι το κομμάτι της εφαρμογής, με το οποίο έρχεται σε άμεση επαφή ο χρήστης, πάνω στο οποίο μπορεί να εκτελεί ενέργειες και να βλέπει τα αποτελέσματά τους. Το UI του activity ορίζεται, συνήθως, με τη βοήθεια ενός αρχείου XML. Μπορεί να οριστεί και προγραμματιστικά, αλλά ο πρώτος τρόπος είναι προτιμότερος, αφού αποσυνδέει τον κώδικα της εφαρμογής από τη μορφή του UI .

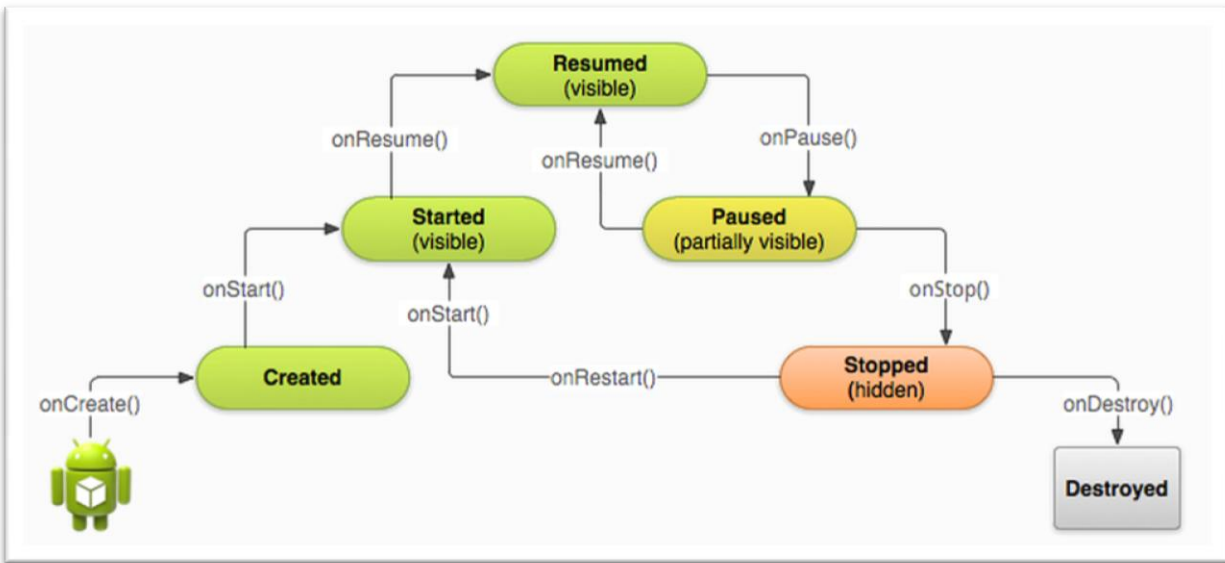
Ένα activity μπορεί να βρίσκεται σε μία από έξι καταστάσεις:

- Created: Το activity μόλις δημιουργήθηκε και είναι έτοιμος να προχωρήσει σε νέες καταστάσεις.
- Started :Έπειτα και μόνο από την δημιουργία του Activity ξεκινάει την λειτουργία του
- Resumed (Running): Το activity βρίσκεται στο προσκήνιο και έχει το user focus.
- Paused: Ένα άλλο activity βρίσκεται στο προσκήνιο και έχει το user focus, αλλά αυτό το activity είναι ακόμα ορατό.
- Stopped: Το activity βρίσκεται στο παρασκήνιο και δεν είναι ορατό στον χρήστη.
- Destroyed : Το activity καταστράφηκε

Κατά τη διάρκεια του κύκλου ζωής του activity, το σύστημα καλεί μια σειρά από callback μεθόδους πάνω σε αυτό. Οι κυριότερες είναι οι εξής:

onCreate()	Καλείται κατά τη δημιουργία του activity. Ακολουθείται πάντα από την onStart.
onStart()	Καλείται αμέσως πριν το activity γίνει ορατό στον χρήστη.
onPause()	Καλείται όταν κάποιο άλλο activity πρόκειται να περάσει στο προσκήνιο.
onResume()	Καλείται όταν το activity είναι πλέον ορατό, προτού ο χρήστης αρχίσει να αλληλεπιδρά με αυτό. Ακολουθείται πάντα από την onPause.
onStop()	Καλείται όταν το activity δεν είναι πλέον ορατό στον χρήστη.
onRestart()	Καλείται πριν την onStart, όταν το activity έχει γίνει Stopped.
onDestroy()	Καλείται πριν την καταστροφή του activity. Μετά την κλήση των onPause, onStop ή onDestroy, το activity θεωρείται killable και το σύστημα μπορεί να αποφασίσει να τερματίσει τη διεργασία του. Ο κύκλος ζωής ενός activity και οι αντίστοιχες μέθοδοι παρουσιάζονται στην εικόνα της διπλανής σελίδας

Πίνακας 1 – Μέθοδοι διαχείρισης ζωής μιας δραστηριότητας



Σχήμα 12-Κύκλος ζωής μιας δραστηριότητας.

2.4.3.2 Υπηρεσίες (Services)

Ένα service είναι ένα στοιχείο μιας εφαρμογής, που χρησιμοποιείται, κυρίως, για την εκτέλεση λειτουργιών, που απαιτούν μεγάλο χρονικό διάστημα, στο παρασκήνιο. Σε αντίθεση με το activity, ένα service δεν παρέχει UI. Για την εκτέλεση χρονοβόρων ενεργειών, μπορούν επίσης να χρησιμοποιηθούν και Java threads. Όμως, σε αντίθεση με το service, ένα thread είναι συνδεδεμένο με το αντικείμενο που το δημιουργήσε (π.χ. ένα activity), οπότε ο κύκλος ζωής του δεν μπορεί να ξεπεράσει αυτόν του πατρικού αντικειμένου. Ένα service εμφανίζεται με δύο μορφές, ενώ μπορεί να βρίσκεται ταυτόχρονα και στις δύο:

- **Started:** Ένα component μιας εφαρμογής μπορεί να ξεκινήσει το service, καλώντας τη μέθοδο `startService()`. Από εκεί και πέρα, το service μπορεί να τρέχει για απεριόριστο χρονικό διάστημα, ακόμα και αν το αρχικό component καταστραφεί. Το Service συνεχίζει να εκτελείται, μέχρι κάποιο άλλο component να το σταματήσει, είτε

να σταματήσει το ίδιο τον εαυτό του, αφού ολοκληρώσει την εργασία για την οποία ξεκίνησε.

- **Bound:** Ένα component μπορεί να συνδεθεί (bind) με ένα service, καλώντας τη μέθοδο `bindService ()`. Ένα bound service παρέχει ένα interface, μέσω του οποίου, τα components που συνδέονται με αυτό, μπορούν να αλληλεπιδράσουν μαζί του. Τα components μπορεί να ανήκουν στην ίδια ή σε κάποια άλλη εφαρμογή. Στην τελευταία περίπτωση, έχουμε interprocess communication (IPC). Στην περίπτωση που πρόκειται για την ίδια εφαρμογή, το service object μπορεί να χρησιμοποιηθεί απευθείας από τα άλλα components. Ένα bound service τερματίζεται, όταν και το τελευταίο component αποσυνδέεται από αυτό, εκτός αν είναι επίσης και started.

2.4.3.3 Δέκτες εκπομπής(Broadcast Receivers)

Ένας broadcast receiver είναι ένα στοιχείο μιας εφαρμογής, που ως σκοπό έχει την αποδοχή system-wide broadcast μηνυμάτων, που στέλνονται, είτε από το ίδιο το λειτουργικό σύστημα, είτε από άλλες εφαρμογές. Τα μηνύματα αυτά συνδέονται, συνήθως, με κάποιο γεγονός, όπως, για παράδειγμα, ότι η οθόνη έκλεισε, ότι το επίπεδο της μπαταρίας είναι χαμηλό, ότι κάποιο download που εκτελούνταν στο παρασκήνιο ολοκληρώθηκε. Ένας broadcast receiver δεν έχει user interface. Συνήθως, η λειτουργία του είναι να εκτελεί κάποιες απλές ενέργειες, σχετικές με το γεγονός που συνέβη, ή να ενεργοποιεί άλλα components, προκειμένου να εκτελέσουν πιο σύνθετες ενέργειες.

2.4.4 Επιπλέον χρήσιμα συστατικά Λογισμικού Android για την δημιουργία της εφαρμογής μας

1. **SQLite Database:** Η SQLite είναι μία μηχανή βάσης δεδομένων, ειδικά σχεδιασμένη για ενσωματωμένα συστήματα. Δεν έχει ξεχωριστή διεργασία server, αλλά, αντίθετα, εκτελείται μέσα στη διεργασία της εφαρμογής που τη χρησιμοποιεί. Η πλήρης βάση δεδομένων είναι αποθηκευμένη σε ένα μόνο αρχείο. Το Android παρέχει κατάλληλο API για τη δημιουργία βάσεων δεδομένων, χρησιμοποιώντας την SQLite, καθώς και API για την εκτέλεση ερωτημάτων και εντολών SQL πάνω σε αυτές.

Οι περισσότερες μηχανές βάσης δεδομένων χρησιμοποιούν στατικούς τύπους για τα δεδομένα που αποθηκεύουν. Η SQLite, αντίθετα, χρησιμοποιεί ένα πιο γενικό σύστημα δυναμικών τύπων. Ο τύπος ενός αντικείμενου στην SQLite δεν εξαρτάται από τη στήλη της βάσης δεδομένων στην οποία βρίσκεται, αλλά είναι συσχετισμένος με το ίδιο το αντικείμενο. Μπορούν δηλαδή αντικείμενα που βρίσκονται στην ίδια στήλη να έχουν διαφορετικούς τύπους. Η SQLite ορίζει πέντε Storage Classes για τα δεδομένα της:

- NULL
- INTEGER: signed integer (1, 2, 3, 4, 6, ή 8 bytes)
- REAL: floating point (8-byte IEEE floating point number)
- TEXT: text string (no data length)

Με τη βοήθεια αυτών μπορούν να αποθηκευτούν και οι υπόλοιποι τύποι, που ορίζονται στην SQL, αλλά δεν υπάρχουν στην SQLite. Για παράδειγμα μια ημερομηνία μπορεί να αποθηκευτεί ως ένας ακέραιος αριθμός (π.χ. Milliseconds από 1-1-1970 με αποτέλεσμα την ακριβή ανάκτηση χρήσιμης πληροφορίας μέσω αιτημάτων (queries).

2. **Assets Folder:** Φάκελος ο οποίος μπορεί να χρησιμοποιηθεί στην εισαγωγή «raw» data αρχείων όπως πχ XML τα οποία είναι χρήσιμα στην ομαλή λειτουργία της εφαρμογής μας καθώς μπορεί εύκολα να ανανεώνεται κάθε φορά ως μορφή ενημέρωσης, για την επεκτασιμότητα της εφαρμογής.

3 Intelligate(Intelligent Gateway) και γενικός σχεδιασμός του συστήματος



Σχήμα 12 – Λογότυπο εφαρμογής Intelligate

Στο εν λόγω κεφάλαιο θα πραγματοποιηθεί μια εμπεριστατωμένη αναφορά περιγραφής των δομικών στοιχείων της εφαρμογής **Intelligate(Intelligent Gateway)** , εστιάζοντας κυρίως στο σενάριο χρήσης το οποίο καλύπτει, τον τρόπο της επίλυσης του, καθώς στην περιγραφή της γενικής αρχιτεκτονικής του συστήματος που έχουμε υλοποιήσει με βάση το μοντέλο της υπηρεσιοκεντρικής αρχιτεκτονικής που αναφέραμε στο 2^ο κεφάλαιο. Στο επόμενο κεφάλαιο θα γίνει αναλυτική αναφορά στα επιμέρους κομμάτια της γενικής αρχιτεκτονικής καθώς και εκτενέστερη ανάλυση της υλοποίησης των μηχανισμών που προσφέρει η εφαρμογή μας.

3.1 Σενάριο Χρήσης και λειτουργικότητα της εφαρμογής

Το σύστημα μας αναπτύχθηκε με σκοπό να λύσει καθημερινά προβλήματα στον χώρο της υγείας και να κάνει το έργο των γιατρών πιο εύκολο, δίνοντας τους τη δυνατότητα απομακρυσμένης παρακολούθησης-βοήθειας προς τους ασθενείς.

Το σενάριο χρήσης μας αντιμετωπίζει το πρόβλημα της μείωσης του κόστους για αναπηρίες που παράγονται από καρδιακές παθήσεις και προάγει την ευκολότερη διαχείριση της αποκατάστασης του ασθενούς επηρεάζοντας έτσι τον παθόντα , την οικογένεια του και κατ'

επέκταση ολόκληρη την κοινωνία. Το αντίκτυπο του παραπάνω προβλήματος για τους ασθενείς είναι ότι δεν είναι σε θέση οι ίδιοι να ελέγχουν την κατάστασή από μόνοι τους καθώς τους οδηγεί συνεχώς σε τακτικές συναντήσεις με το εκάστοτε γιατρό επιμελητή τους ή ακόμη και στην μακράς διαρκείας νοσηλεία τους σε ένα νοσοκομείο αυξάνοντας έτσι το κόστος νοσηλείας(περισσότερη απασχόληση προσωπικού) και θεραπείας(μηχανήματα-όργανα) για το ίδιο το οργανισμό υγείας αλλά και την ταλαιπωρία του ίδιου του πάσχοντα. Το άγχος ,ο φόβος και η αβεβαιότητα είναι κυρίαρχα συναισθήματα σε ευαίσθητες κοινωνικά ομάδες όπως άτομα τρίτης ηλικία-άτομα με αναπηρίες και καρδιοπάθειες , με αποτέλεσμα να μην μπορούν να απολαύσουν καθημερινές δραστηριότητες και να είναι σε συνεχή εξάρτηση με τον εκάστοτε πάροχο ιατρικής περίθαλψής τους.

Μια επιτυχημένη λύση βελτιώνει τα αποτελέσματα όσο αναφορά

- Αποκατάσταση της καρδιοπάθειας του ασθενούς αυξάνοντας έτσι την αυτοπεποίθηση του στον καθημερινό τρόπο ζωής του ,καθώς ο ίδιος έχει πάντα τον γιατρό σε ενημέρωση.
- Μειώνει το κόστος σε θέματα νοσηλείας και μηχανημάτων αλλά και μηχανοργάνωσης. Έτσι νοσηλευτές δεν θα έχουν τόσο μεγάλο φόρτο εργασίας , αλλά θα φροντίζουν τους ίδιους τους ασθενείς μόνο σε περιπτώσεις όπου ο γιατρός θα ορίσει κρίσιμες. Η κρισιμότητα της κατάστασης του ασθενούς μπορεί να αναγνωρίζεται από τον γιατρό σε πραγματικό χρόνο ή από την ανάλυση του ιστορικού της ζωτικής του πληροφορίας.
- Δραματική μείωση χρόνου παρακολούθησης ασθενούς από τον γιατρό του.

Η εφαρμογή Intelligate προορίζεται για γιατρούς νοσηλευτές και νοσοκομεία που διεξάγουν προγράμματα καρδιακής αποκατάστασης. Η Intelligate είναι μια νέφους βασισμένη(cloud based) εφαρμογή σε κινητή πλατφόρμα(κινητό ή tablet) λογισμικού Android η οποία δίνει την δυνατότητα απομακρυσμένης παρακολούθησης καρδιακού παλμού και ποσοστού οξυγόνου στο αίμα σε πραγματικό χρόνο με την χρήση κατάλληλων αισθητήρων.

Ουσιαστικά η εφαρμογή Intelligate δεν αποτελεί ένα απλό διάυλο επικοινωνίας μεταξύ αισθητήρα και νέφους αλλά διαθέτει την επιπλέον εξυπνάδα να μπορεί να επεξεργάζεται πλέον τα δεδομένα που παράγει και να τα στέλνει σε συγκεκριμένες χρονικές στιγμές που ορίζονται από την κρισιμότητα μια κατάστασης ή από την επιλογή του ίδιου του χρήστη. Θα γίνει αναλυτική περιγραφή του τρόπου αποστολής δεδομένων στο νέφος σε επόμενη ενότητα του ίδιου κεφαλαίου. Επιπλέον ο γιατρός μπορεί να ορίσει και να ανανεώνει κανόνες οι οποίοι θα χαρακτηρίζουν τα κατώφλια των τιμών που παράγουν οι αισθητήρες, κρίνοντας με αυτόν τρόπο την κατάσταση του ασθενούς σε κρίσιμη η μη(Πχ ο καρδιακός παλμός ξεπέρασε τα 120 bpm). Η Intelligate έχει επιπλέον μια λειτουργία που τη καθιστά ιδιαίτερη, καθώς η κινητή συσκευή του ασθενούς μπορεί να μην βρίσκεται συνεχώς εντός δικτύου , και με αυτόν τον τρόπο θα πρέπει να διασφαλιστεί ότι τα δεδομένα αυτά δεν θα πάνε χαμένα και ότι θα αποσταλούν με την πρώτη ευκαιρία στο νέφος. Μέσω της Intelligate ο γιατρός μπορεί εύκολα να επιβλέπει τα ζωτικά

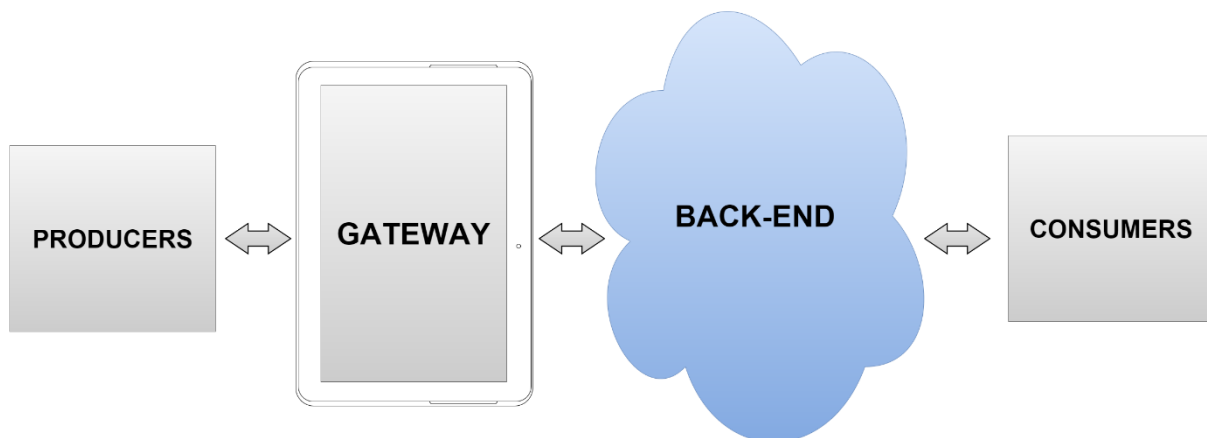
δεδομένα οποιουδήποτε ασθενούς και να ενημερώνεται άμεσα για την κατάστασή του σε πραγματικό χρόνο. Επιπλέον υπάρχει η δυνατότητα η δυνατότητα προβολής του πλήρους ιστορικού μετρήσεων και κατάστασης κάθε ασθενούς από τον ίδιο τον γιατρό. Τέλος ο επιβλέπων γιατρός μπορεί να ενημερώνει με ασφαλή τρόπο τον ασθενή του οποιαδήποτε χρονική στιγμή με την αποστολή μηνυμάτων όπου θα ορίζουν κατάλληλες ενέργειες του πάσχοντα με στόχο την περιστασιακή επίλυση του προβλήματος.

3.2 Γενική Αρχιτεκτονική

Η υλοποίηση του συστήματός μας βασίζεται πάνω στη λογική της Υπηρεσιο-κεντρικής Αρχιτεκτονικής (Service Oriented Architecture - SoA) όπως αναλύθηκε και στο δεύτερο κεφάλαιο. Δηλαδή στην αρχιτεκτονική που θέλει το σύστημά μας να αποτελείται από μικρότερα υπο-συστήματα ή αλλιώς υπηρεσίες. Οι υπηρεσίες αυτές με τον κατάλληλο προγραμματισμό, μπορούν να επικοινωνούν μεταξύ τους και να παράγουν πληροφορία η οποία μπορεί να αξιοποιηθεί κατάλληλα. Τα τμήματα αυτής της αρχιτεκτονικής είναι:

- Παραγωγοί (Producers)
- Πύλη εφαρμογή (Application Gateway)
- Διεπαφή διαχείρισης Συστήματος νέφους (Back-End)
- Καταναλωτές (Consumers)

και φαίνονται ξεκάθαρα στο παρακάτω σχήμα.

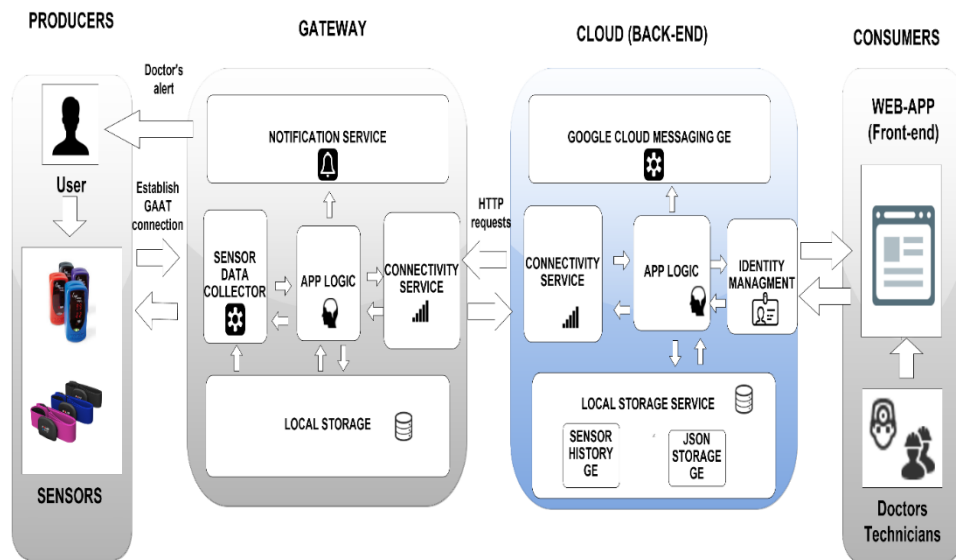


Σχήμα 13 – Γενική αρχιτεκτονική συστήματος

- **Producers:** Διάφοροι χρήστες έχουν στην διάθεση τους αισθητήρες οι οποίοι παράγουν πληροφορία για τα ζωτικά τους δεδομένα όπως για παράδειγμα μέτρηση του καρδιακού ρυθμού (heart rate). Σε αυτό το στάδιο λειτουργίας στην δική μας περίπτωση οι αισθητήρες είναι τύπου Bluetooth low energy και στην περίπτωση τους κάθε φορά που ενεργοποιούνται ξεκινάει η διαδικασία του advertising προκειμένου να συνδεθούν με μια κεντρική συσκευή.
- **Gateway:** Ίσως το σημαντικότερο κομμάτι της διπλωματικής μας εργασίας καθώς αποτελεί την βασική ιδέα της εφαρμογής. Όπως φαίνεται και στο σχήμα το κομμάτι αυτό στην δική μας περίπτωση είναι μια Android Εφαρμογή η οποία χρησιμοποιεί της ιδιότητες μιας πύλης δικτύου(gateway) μέσω ενός κινητού η tablet για να μεταφέρει τα δεδομένα από τους αισθητήρες στο νέφος. Οι χρήστες μπορούν να αλληλεπιδρούν με την εφαρμογή όπως αυτό θα αναλυθεί σε επόμενες ενότητες.
- **Διεπαφή Συστήματος Νέφους:** Στον εν λόγω λειτουργικό κομμάτι , τα δεδομένα έχουν ληφθεί από το Front-end και είναι πλέον στο νέφος. Τα δεδομένα αυτά επεξεργάζονται και αποθηκεύονται κατάλληλα με βάση το σενάριο χρήσης της εφαρμογής μας με στόχο την εξυπηρέτηση των καταναλωτών.
- **Καταναλωτές:** Το τελευταίο υποσύστημα της γενικής αρχιτεκτονικής μας Σε αυτό το τμήμα βρίσκονται οι χρήστες οι οποίοι θέλουν να έχουν πρόσβαση στις πληροφορίες που παράγει το σύστημά μας, καθώς επίσης και η εφαρμογή διεπαφής μέσω της οποίας έχει πρόσβαση ο χρήστης. Σε αυτό το κομμάτι υλοποίησης αναπτύσσεται και το γραφικό περιβάλλον(graphical user interface – GUI) το οποίο διευκολύνει την πρόσβαση του χρήστη στις παρεχόμενες πληροφορίες και υπηρεσίες, διαθέτοντας του ένα σύνολο από εργαλεία.

3.3 Η αρχιτεκτονική του συστήματος Intelligate

Η προηγούμενη ενότητα ανέφερε την γενική αρχιτεκτονική του συστήματος στην οποία στηρίχτηκε το σύστημά μας .Πλέον είμαστε σε θέση να δείξουμε την ολοκληρωμένη αρχιτεκτονική του Intelligate και να γίνει αναφορά για τα συγκεκριμένα εσωτερικά κομμάτια κάθε υπο-συστήματος χωριστά όπου θα αναλυθούν στις επόμενες υπο-ενότητες.



Σχήμα 14 – Αρχιτεκτονική συστήματος Intelligate

3.3.1 Παραγωγοί (Producers)

Όπως προαναφέραμε σε προηγούμενη ενότητα, διάφοροι χρήστες έχουν στην διάθεση αισθητήρες οι οποίοι παράγουν πληροφορία για τα ζωτικά τους δεδομένα. Η ενότητα αυτή διαχωρίζεται ουσιαστικά σε 2 μέρη.

- Χρήστες ασθενείς
- Αισθητήρες

Στους χρήστες-ασθενείς όπου χρησιμοποιούν τους αισθητήρες και έχουν στην διάθεσή τους την κινητή συσκευή (tablet ή κινητό) με το οποίο μπορούν ουσιαστικά μόνο να βλέπουν τα δεδομένα τα οποία παράγουν οι ίδιοι οι αισθητήρες. Επιπλέον οι ίδιοι οι χρήστες έχουν την δυνατότητα μέσω της Διεπαφή χρήστων (front end) της εφαρμογής Intelligate να επιλέξουν κατ'επιλογή (on demand) την αποστολή των δεδομένων που οι αισθητήρες που παράγουν στον γιατρό τους. Τέλος μπορούν να λαμβάνουν ενημερώσεις από τον γιατρό τους όταν ο ίδιος αποφασίσει ότι μια κατάσταση είναι κρίσιμη.

Το 2^ο μέρος της ενότητας Producers είναι προφανώς οι αισθητήρες-περιφερειακά οι οποίοι παράγουν δεδομένα ζωτικής σημασίας. Στην δική μας περίπτωση χρησιμοποιήθηκαν 2 ειδών αισθητήρες όπου αναφέρονται με τα βασικά χαρακτηριστικά τους παρακάτω.

1. Polar H7 Heart Rate Sensor



Σχήμα 15-Αισθητήρας Polar H7

Ο polar H7 είναι Συνδυαστικός πομπός που καταγράφει σε ζωντανή μετάδοση και με ακρίβεια την καρδιακή συχνότητα σε εφαρμογές ευρείας κλίμακας. Ο αισθητήρας καρδιακού ρυθμού προσαρτάται γύρω από το στήθος με έναν ελαστικό ιμάντα . Ανιχνεύει χτύπο της καρδιάς , το ηλεκτρικό σήμα της καρδιάς , στην ακρίβεια και την αξιοπιστία του ηλεκτροκαρδιογραφήματος. Στη συνέχεια δίνει μια αναφορά χρονισμού για περιστατικών ρυθμό της καρδιάς και μεταδίδει την πληροφορία στον υπολογιστή κατάρτισης. Ο υπολογιστής κατάρτισης διαβάζει αυτό το σήμα από τον αισθητήρα καρδιακού ρυθμού και υπολογίζει τον αριθμό των καρδιακών παλμών ανά λεπτό [16].

2. Nonin 3230 Bluetooth Smart Pulse oximeter



Σχήμα 16-Αισθητήρας Nonin Nonix 3230

Με την αυξημένη ανάγκη για απομακρυσμένη διαχείριση της νόσου , το Onyx II, μοντέλο 3230 με ασύρματη τεχνολογία Bluetooth Smart παρέχει μια λύση παρακολούθησης οξυμετρίας. Η Ασύρματη τεχνολογία Bluetooth Smart σε συσκευές παρακολούθησης της υγείας επιτρέπει στους ασθενείς , σε συνδυασμό με τους κλινικούς γιατρούς , να παρακολουθούν πιο εύκολα ζωτικά δεδομένα όπως το εύρος τιμών κορεσμού οξυγόνου και καρδιακών παλμών ανά λεπτό. Ως αποτέλεσμα , οι ασθενείς θα μπορούν να εκτελέσουν καθημερινές δραστηριότητες και να στείλουν τους ζωτικής σημασίας δεδομένα ασύρματα μέσω συσκευών επικοινωνίας, όπως κινητά τηλέφωνα και υπολογιστές , κ.λπ.[17]

3.3.2 Έξυπνη Πύλη διαδικτύου(Intelligate)

Το εν λόγω λειτουργικό κομμάτι αποτελεί τον βασικό πυλώνα της διπλωματικής μας εργασίας. Ουσιαστικά η ενότητα Gateway αποτελείται από μια έξυπνη εφαρμογή η οποία έχει την δυνατότητα να συλλέγει κατάλληλα τα δεδομένα που παράγουν οι αισθητήρες ανα χρονική στιγμή , να επεξεργάζεται-αποθηκεύει τοπικά αυτά τα δεδομένα και να τα στέλνει στο νέφος(cloud-back end) με έξυπνο τρόπο με βάση το πρότυπο σχήμα που έχει ορίσει ο γιατρός σε κάθε ασθενή. Η εφαρμογή επιπλέον διαθέτει μια διαπαφή χρήστη(User Interface) με την οποία ο γιατρός μπορεί να εγγράψει ένα καινούργιο ασθενή στο σύστημα, να ορίσει τους αισθητήρες με τους κανόνες τους για κάθε χρήστη. Το User – interface δίνει την δυνατότητα στον χρήστη-ασθενή επιπλέον να μπορεί να ελέγχει ανα πάσα στιγμή τις μετρήσεις των BLE αισθητήρων καθώς και να τροποποιήσει τον τρόπο αποστολής των δεδομένων με τρεις διαφορετικούς τρόπους όπου θα αναλυθούν αργότερα. Τέλος το User Interface δίνει την δυνατότητα στον χρήστη να ειδοποιείται από τον γιατρό του με κατάλληλα μηνύματα ανα πάσα χρονική στιγμή. Σύμφωνα με το «σχήμα 14» όπου παρουσιάζεται ουσιαστικά η αρχιτεκτονική του συστήματος μας η ενότητα Gateway αποτελείται από 5 συγκεκριμένες ενότητες.

- **Sensor Data Collector:** Η συγκεκριμένη υπο-ενότητα έχει σαν στόχο την συλλογή των δεδομένων από τους αισθητήρες και την προώθηση αυτών των δεδομένων προς τις άλλες υπο-ενότητες του Intelligate.
- **Local Storage Service:** Σε αυτήν την ενότητα γίνεται η τοπική αποθήκευση συγκεκριμένων τύπων δεδομένων. Χαρακτηριστικά το Local Storage Service περιλαμβάνει το XML σχήμα όλων των αισθητήρων που υποστηρίζονται από το συγκεκριμένο app, καθώς και αναγνωριστικά στοιχεία του ίδιου του ασθενούς(email) με τους κανόνες που τον διέπουν κωδικοποιημένα κατάλληλα. Τέλος περιλαμβάνει μέσους όρους μετρήσεων που παράγουν οι αισθητήρες σε συγκεκριμένα χρονικά διαστήματα σε

περίπτωση όπου η κινητή συσκευή βρεθεί εκτός δικτύου. Με αυτόν τον τρόπο υπάρχει η διασφάλιση δεν υπάρχει χαμένη πληροφορία.

- **Notification Service:** Το Notification Service έχει τον έλεγχο της εφαρμογής για πιθανές αλλαγές καταστάσεων, με στόχο την άμεση ενημέρωση του χρήστη. Παραδειγματικά μιλώντας όταν υπάρχουν δεδομένα που δεν έχουν αποσταλεί στο νέφος με το που συνδεθεί ο αισθητήρας με την συσκευή εμφανίζεται ένα σχετικό μήνυμα «Υπάρχει διαθέσιμη πληροφορία που δεν έχει αποσταλεί στο νέφος, θέλετε να συνεχίσετε με την αποστολή των δεδομένων;», αφήνοντας έτσι τον χρήστη να αποφασίσει αν θέλει ή όχι να μεταφέρει τα δεδομένα του στο νέφος. Ένα επιπλέον παράδειγμα είναι η ενημέρωση του ασθενούς με ηχητική σήμανση(alert) όταν ο γιατρός στείλει ένα μήνυμα στην συσκευή του χρήστη-ασθενή.
- **Connectivity Service (Gateway):** Αποτελεί τον δίαυλο επικοινωνίας μεταξύ Gateway και Cloud (Back end). Τα δεδομένα κωδικοποιούνται κατάλληλα με την μορφή JSON string και αποστέλλονται στο νέφος με βάση το πρωτόκολλο HTTP. Η Διαχείριση των απάντησεων από τον διακομιστή server είναι ευθύνη της συγκεκριμένης υπο-ενότητας (Πχ Σφάλμα 400 λανθασμένη αίτηση).
- **Application Logic (Gateway):** Η Λογική Συστήματος αποτελεί το σύνολο εκείνων των διεργασιών που έχουμε αναπτύξει προκειμένου οι υπηρεσίες που έχουμε χρησιμοποιήσει να μπορούν να επικοινωνούν μεταξύ τους αποτελεσματικά. Ορίζει τον τρόπο με τον οποίο επικοινωνούν οι υπηρεσίες ώστε η πληροφορία που παράγεται από το τμήμα των Παραγωγών να αξιοποιείται και να φθάνει στο τμήμα των Καταναλωτών. Θα μπορούσαμε να πούμε ότι η Λογική Συστήματος (Application Logic) είναι ο πυρήνας της εφαρμογής που συντονίζει όλες τις υπηρεσίες και τις λειτουργίες που μεσολαβούν μεταξύ της λήψης των δεδομένων και του γραφικού περιβάλλοντος της εφαρμογής στην οποία έχει πρόσβαση ο χρήστης.

3.3.3 Διεπαφή διαχείρισης Συστήματος νέφους(Cloud back-end)

Το τμήμα «Διεπαφή διαχείρισης Συστήματος νέφους», αποτελείται από τις υπηρεσίες ειδικού σκοπού που χρησιμοποιήσαμε για την επεξεργασία και την αποθήκευση των δεδομένων όλων των αισθητήρων που αλληλεπιδρούν με το σύστημά μας και στην δική μας περίπτωση υλοποιήθηκε με στόχο την επιβεβαίωση ορθής λειτουργίας της εφαρμογής Intelligate. Πιο συγκεκριμένα, οι υπηρεσίες γενικού σκοπού(Generic Enablers) είναι ο Google Cloud Messaging και JSON Storage GE, οι οποίες είναι υπεύθυνες για την διαχείριση των μηνυμάτων του γιατρού προς τον χρήστη-ασθενή και την αποθήκευση πληροφορίας και δεδομένων αντίστοιχα.

Επιπλέον, το τμήμα αυτό περιέχει τον μηχανισμό ταυτοποίησης των χρηστών που εισέρχονται στην εφαρμογή, δηλαδή τον KeyRock Identity Management GE. Τέλος η λογική συστήματος (Application Logic) είναι το σημαντικότερο τμήμα που υλοποιήσαμε και στοχεύει στην μεταφορά της πληροφορίας στα κατάλληλα επιμέρους τμήματα (Αποθήκευση, Ταυτοποίηση, Διαχείριση πληροφορίας) ώστε το σύστημα να είναι λειτουργικό. Σύμφωνα με το «σχήμα 14» όπου παρουσιάζεται ουσιαστικά η αρχιτεκτονική του συστήματος μας η ενότητα Gateway αποτελείται από 5 συγκεκριμένες ενότητες.

- **Connectivity Service (Back end):** Αποτελεί τον δίαυλο επικοινωνίας μεταξύ Gateway και Cloud (Back end) όπως και στην προηγούμενη περίπτωση. Τα δεδομένα από-κωδικοποιούνται κατάλληλα με την μορφή JSON string και αποστέλλονται στις επιπλέον ΥΠΟ ενότητες της διεπαφής διαχείρισης Συστήματος νέφους.
- **Application Logic(Back end) :** Μπορεί να θεωρηθεί το σημαντικότερο υπο-τμήμα της διεπαφής διαχείρισης συστήματος νέφους. Στο τμήμα αυτό περιέχονται έλεγχοι, κανονισμοί και οι απαραίτητες κλήσεις API ώστε το σύστημα να είναι λειτουργικό. Για να πραγματοποιηθεί οποιαδήποτε λειτουργία στο back-end, τα δεδομένα περνούν από αυτό το τμήμα για να σταλούν σε κάποιον Generic Enabler ή να εμφανιστούν ως «απάντηση» (response) στον χρήστη-καταναλωτή.
- **Storage Service:** Χρησιμοποιείται στο σύστημα μας για την αποθήκευση πληροφορίας σχετικά με τους χρήστες-ασθενείς όπως αναγνωριστικά της συσκευή τους, κανόνων των αισθητήρων, τους Αισθητήρες και επιπλέον αποθηκεύεται το ιστορικό δεδομένων που έχει σταλεί από κάποιον αισθητήρα σε συνδυασμό με την ημερομηνία μέτρησης και την κατάσταση του ασθενούς.
- **Google cloud Messaging GE:** Αποτελεί μια υπηρεσία γενικού σκοπού όπου πλέον ο χρήστης καταναλωτής-γιατρός μπορεί να αποστείλει με ασύγχρονο τρόπο μηνύματα προς τον χρήστη-ασθενή χρησιμοποιώντας σαν παράμετρο ένα αναγνωριστικό id της συσκευής του χρήστη-ασθενή.
- **Identity management:** Έχει σαν στόχο την ταυτοποίηση χρηστών-καταναλωτών που αλληλεπιδρούν με την web εφαρμογή με στόχο την ασφάλεια του συστήματος.

3.3.4 Καταναλωτές (Consumers)

Το τελευταίο υποσύστημα της γενικής αρχιτεκτονικής μας Σε αυτό το τμήμα βρίσκονται οι χρήστες οι οποίοι θέλουν να έχουν πρόσβαση στις πληροφορίες που παράγει το σύστημά μας,

καθώς επίσης και η εφαρμογή διεπαφής μέσω της οποίας έχει πρόσβαση ο χρήστης. Σε αυτό το κομμάτι υλοποίησης αναπτύσσεται και το γραφικό περιβάλλον (graphical user interface – GUI) το οποίο διευκολύνει την πρόσβαση του χρήστη στις παρεχόμενες πληροφορίες και υπηρεσίες, διαθέτοντας του ένα σύνολο από εργαλεία. Σε αυτήν την τελευταία υπο-ενότητα πρωταρχικό ρόλο έχουν.

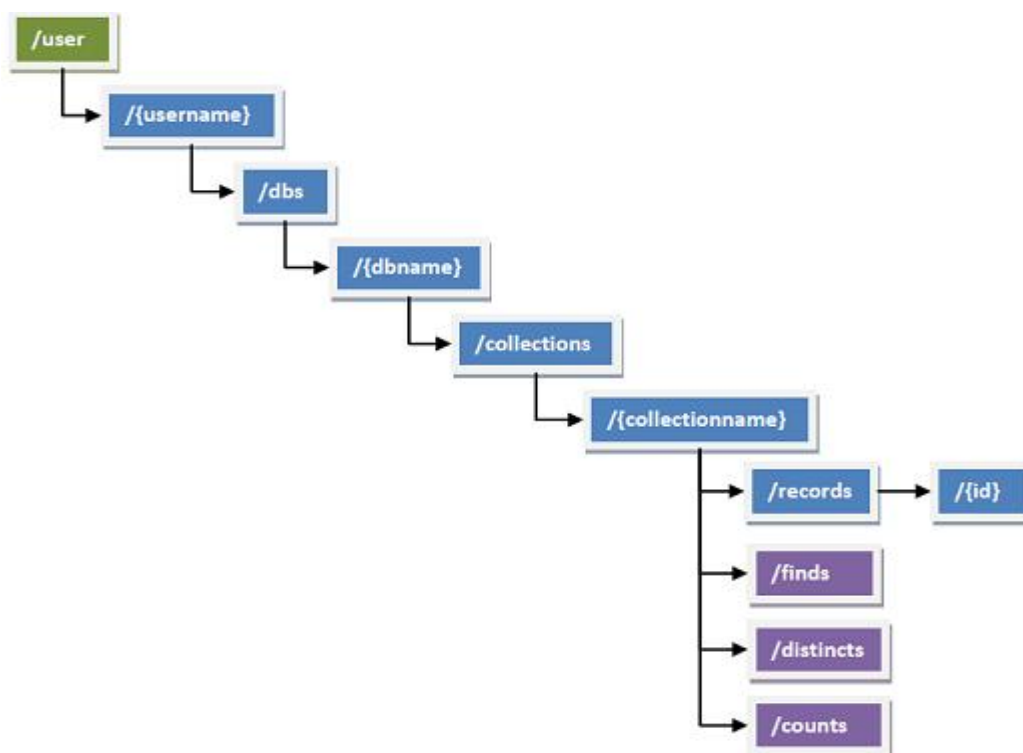
- **Χρήστες καταναλωτές:** Που αποτελούνται από ειδικούς τεχνικούς όπου ορίζουν νέα σχήματα για καινούργιους αισθητήρες που μπορεί να έχουν στην διαθεσή τους οι οργανισμοί υγειονομικής περίθαλψης. Καθώς και τους γιατρούς οι οποίοι μπορεί να θέτουν του ειδικούς κανόνες για κάθε χρήστη-ασθενή ξεχωριστά καθώς και να αποστέλουν μηνύματα ανα τακτά χρονικά διαστήματα. Οι γιατροί επιπλέον έχουν πρόσβαση στο ιστορικό μετρήσεων κάθε ασθενούς και επιπλέον μπορούν να παρακολουθούν τα αποτελέσματα που παράγουν οι αισθητήρες σε πραγματικό χρόνο (real time) ελέγχοντας έτσι με αποτελεσματικό τρόπο την κατάσταση του κάθε ασθενούς ξεχωριστά
- **Web application:** Διαδικτυακή εφαρμογή με Διεπαφή χρήστη (User Interface) που επιτρέπει στους χρήστες καταναλωτές να μπορούν να αλληλεπιδρούν με αποτελεσματικό τρόπο με τους μηχανισμούς που προσφέρει η εφαρμογή.

3.4 Υπηρεσίες γενικού Σκοπού για της ανάγκες του συστήματος Intelligate

3.4.1 JSON storage GE και ο τρόπος λειτουργίας του

Ο JSON Storage GE είναι η υπηρεσία που υποστηρίζει την αποθήκευση πληροφορίας σε μορφή JSON μέσω της πιο αφηρημένου τύπου βάσης Mongo DB [18]. Η υπηρεσία συνοδεύεται από API σχεδιασμένο με βάση την αρχιτεκτονική REST. Οι βασικές λειτουργίες που υποστηρίζει η υπηρεσία είναι η δημιουργία, ανανέωση και διαγραφή χρηστών όπως επίσης και την δημιουργία βάσεων δεδομένων, συλλογών και εγγραφών. Ο χρήστης δεσμεύει χώρο, έτσι

ώστε να αποθηκεύει δεδομένα, δίνοντας όνομα χρήστη και κωδικό πρόσβασης, χωρίς να επιτρέπεται η προσπέλαση προσωπικών δεδομένων από τρίτους. Ο κάθε χρήστης δημιουργεί βάσεις δεδομένων που περιέχουν συλλογές και αυτές με τη σειρά τους περιλαμβάνουν εγγραφές. Η Εικόνα 17 παρουσιάζει την αναλυτική δομή της υπηρεσίας.



ΣΧΗΜΑ 17-Δομή οργάνωσης JSON-storage GE

Η σχεδίαση και η υλοποίηση της συγκεκριμένης υπηρεσίας παρουσιάζει αρκετά πλεονεκτήματα σε σύγκριση με τους μέχρι τώρα γνωστούς τύπους βάσεων δεδομένων. Αναλυτικότερα, ο JSON Storage GE αποτελεί μια πιο ευέλικτη λύση για αποθήκευση δεδομένων καθώς δέχεται μεγαλύτερο όγκο δεδομένων ανταποκρινόμενος ικανοποιητικά και στην μεγάλη ταχύτητα με την οποία αυτά στέλνονται. Η ασφάλεια που παρέχει στο χρήστη για την προστασία των δεδομένων του συμπεριλαμβάνεται σε ένα ακόμα από τα πλεονεκτήματα της υπηρεσίας. Οι βασικές μέθοδοι που υποστηρίζει η υπηρεσία μέσω της διεπαφής παρουσιάζονται στον Πίνακα 1

Μέθοδοι	Λειτουργίες
POST	Εισαγωγή δεδομένων
GET	Ανάκτηση δεδομένων
PUT	Ανανέωση δεδομένων
DELETE	Διαγραφή δεδομένων

3.4.2 Google Cloud messaging GE και ο τρόπος λειτουργίας του

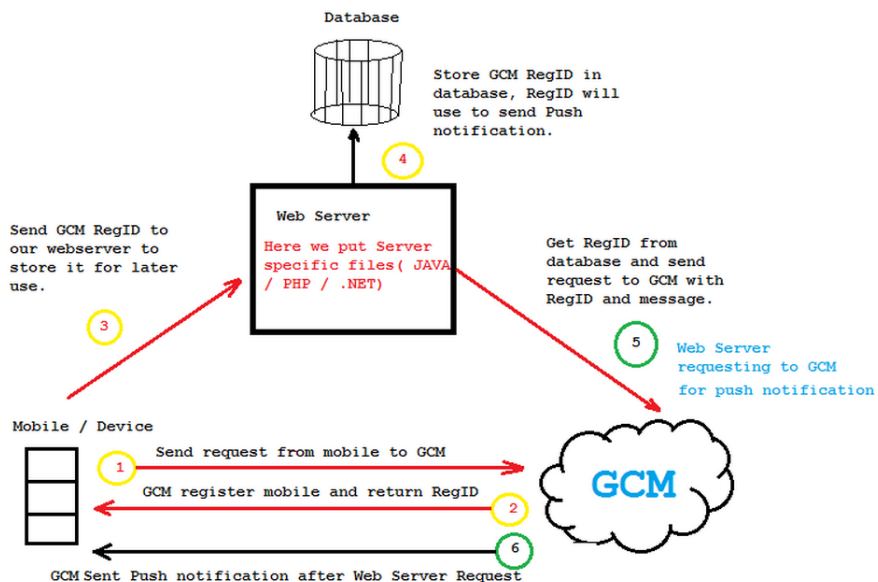
Το Google cloud messaging GE[19] είναι μια δωρεάν υπηρεσία που βοηθά τους προγραμματιστές να στέλνουν μηνύματα σε πολλές πλατφόρμες: Android, iOS και Chrome. Για παράδειγμα, ένας διακομιστής (Server) μπορεί να στείλει μηνύματα απευθείας σε μεμονωμένες συσκευές ή ομάδες συσκευών. Επίσης, η εφαρμογή σε μια συσκευή μπορεί να στείλει μηνύματα απευθείας σε έναν διακομιστή και σε συσκευές που ανήκουν στην ίδια ομάδα.

Μια υλοποίηση GCM περιλαμβάνει έναν διακομιστή σύνδεσης της Google (google connection Server) έναν διακομιστή χρήστη (User-server) και μια εφαρμογή πελάτη (Client app).



Σχήμα 18-Γενική αρχιτεκτονική Google cloud messaging

Όλες οι ενότητες αυτές επικοινωνούν μεταξύ τους με πρωτόκολλο HTTP με τον παρακάτω τρόπο.



Σχήμα 19-Τρόπος μετάδοσης πληροφορίας Google cloud messaging

Τα βήματα έχουν ως εξής

- 1) Το κινητό ή tablet λογισμικού android που περιλαμβάνει την εφαρμογή στέλνει ένα HTTP request στον GCM server με βάση τα credentials όπου έχει ορίσει η Google μέσω του AppEngine (Paas) για κάθε χρήστη όπως θα δούμε σε επόμενο κεφάλαιο της υλοποίησης του συστήματος.
- 2) Ο Google Connection server απαντάει στο μήνυμα της συσκευής και της δίνει ένα μοναδικό id εγγραφής (Registration ID) το οποίο την χαρακτηρίζει.
- 3) Έπειτα το client Application (Intelligence στην δική μας περίπτωση) στέλνει το Registration ID στον δικό μας Server για να το χρησιμοποιήσει αργότερα.
- 4) Ο Server που διαθέτουμε αποθηκεύει το registration id της συσκευής σε μια τοπική βάση δεδομένων μέχρι ο χρήστης να αποφασίσει να στείλει ένα μήνυμα. Στην δική μας περίπτωση θα την αποθηκεύσει στον JSON storage GE.
- 5) Με την χρήση του Registration ID μαζί με τον μήνυμα κείμενο , μπορεί γίνει μια αίτηση προς τον GCM server για να μεριμνήσει να στείλει σε μορφή ενημέρωσης το μήνυμα προς το Client App.
- 6) Ο GCM server στέλνει το μήνυμα προς το Client Up με διασφαλισμένο τρόπο καθώς μπορεί να κρατήσει το μήνυμα μέχρι η συσκευή να βρεθεί εντός δικτύου.

3.4.3 Identity Management-KeyRock

Η υπηρεσία γενικού σκοπού KeyRock IDM [20] καλύπτει κάποιες πτυχές που αφορούν την πρόσβαση των χρηστών στα δίκτυα, υπηρεσίες και εφαρμογές. Επιπλέον, η συγκεκριμένη υπηρεσία χρησιμοποιείται για την έγκριση ξένων υπηρεσιών να αποκτούν πρόσβαση σε προσωπικά δεδομένα που είναι αποθηκευμένα σε ένα ασφαλές περιβάλλον. Ο KeyRock περιλαμβάνει Διεπαφή REST API για την χρήση του από δημιουργούς εφαρμογών.

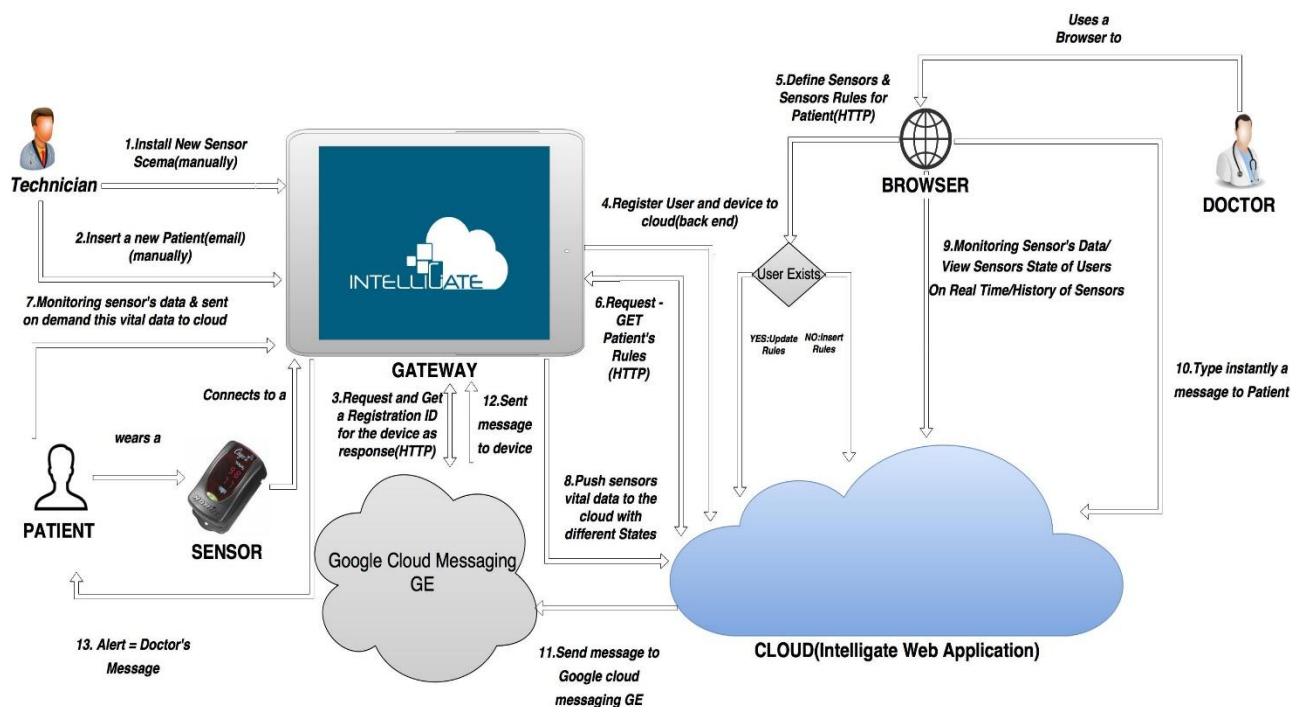
Στην περίπτωση της δικιάς μας υπηρεσίας χρησιμοποιούμε τον IDM για την ταυτοποίηση των χρηστών μας, οι οποίοι αρχικά πρέπει να είναι εγγεγραμμένοι στην υπηρεσία

FIWARE. Ο IDM γνωρίζουμε από τις προδιαγραφές του ότι παρέχει ασφάλεια όσων αφορά τα προσωπικά δεδομένα των χρηστών, γι' αυτόν τον λόγο τον χρησιμοποιούμε. Γνωρίζουμε ότι κάνει χρήση του πρωτοκόλλου επικοινωνίας https, το οποίο ενδείκνυται για την ασφαλή μετάδοση προσωπικών δεδομένων (username, password) μέσω του διαδικτύου.

Ο συγκεκριμένος GE είναι εντός της αρχιτεκτονικής μας, αλλά δεν υλοποιήθηκε καθώς στόχος της διπλωματικής εργασίας ήταν να επικεντρωθεί στην ορθή λειτουργία της εφαρμογής Intelligate.

3.5 Ανάλυση ροής δεδομένων και ενεργειών συστήματος Intelligate

Για να γίνει πιο κατανοητή η χρήση του συστήματος του Intelligate αναγκαίο είναι να παρουσιαστεί με κάποιον τρόπο το σύνολο των ενεργειών του κάθε χρήστη (Τεχνικό προσωπικό, γιατροί και ασθενείς) καθώς και την μεταφορά των αναγκαίων δεδομένων σε μια λογική χρονική σειρά. Με άλλα λόγια πρέπει να δημιουργηθεί ένα διάγραμμα ροής της πληροφορίας και των ενεργειών που απασχολούν κάθε χρήστη συνολικά. Το πιο ιδανικό μοντέλο για να παρουσιαστεί σε συνοπτικό τρόπο όλη η ροή πληροφορίας του συστήματος που αναπτύξαμε είναι με το παρακάτω διάγραμμα ροής εργασιών (workflow diagram).



Σχήμα 19-Τρόπος ροής εργασιών συστήματος Intelligate

Προφανώς η όψη ενός απλού διαγράμματος δεν αρκεί και για αυτόν τον λόγο πρέπει να εξεξηγηθεί κάθε μια ενέργεια ξεχωριστά.

Βήμα 1^ο – Εισαγωγή ενός νέου αισθητήρα στο σύστημα Intelligate

Στο 1^ο βήμα φαίνεται ένας χρήστης ο οποίος είναι και ο διαχειριστής του Συστήματος Intelligate ο οποίος έχει πρόσβαση στα τεχνικά χαρακτηριστικά του συστήματος. Ο Διαχειριστής έχει πάντα την τελευταία έκδοση της εφαρμογής και έχει την δυνατότητα να τροποποιεί μόνο ένα XML (Ενότητα 2.2.2.5) αρχείο το οποίο βρίσκεται στα Asset Folders (Ενότητα 2.4.4). Ουσιαστικά το XML αυτό αρχείο διαθέτει όλη την χρήσιμη πληροφορία για τους αισθητήρες ούτως ώστε να μπορούν να αναγνωριστούν από την πύλη διαδικτύου (Gateway) μέσω της εφαρμογής μας (Intelligate). Με τον τρόπο αυτό ορίζοντας κάθε νέο XML σχήμα για κάθε αισθητήρα έχουμε μια δυναμική υποστήριξη αισθητήρων από την συσκευής μας , με τον τρόπο που θα αναλύσουμε στο επόμενο κεφάλαιο. Επιπλέον ο γιατρός είναι και υπεύθυνος για την εγκατάσταση της εφαρμογής στο κινητό του χρήστη.

Βήμα 2^ο – Εισαγωγή ενός νέου χρήστη στο σύστημα του Intelligate

Στο 2^ο βήμα φαίνεται πως ο ίδιος ο διαχειριστής του συστήματος και μόνο αυτός μέσω Authentication (δεν έχει υλοποιηθεί) , μπορεί να εισάγει νέους χρήστες-ασθενείς στο σύστημα Intelligate χρησιμοποιώντας ως χαρακτηριστικό ID το email του χρήστη. Το προφίλ του χρήστη αποθηκεύεται τοπικά της συσκευής αλλά προωθείται και προς το νέφος όπως θα δούμε σε επόμενο βήμα

Βήμα 3^ο- Αίτημα προς τον Google cloud messaging Generic Enabler για την παροχή αναγνωριστικού ID της συσκευής

Όπως είναι αναμενόμενο το σύστημά μας εξυπηρετεί διάφορους χρήστες-ασθενείς όπου ο κάθενας από αυτούς διαθέτουν διαφορετικά devices. Με κάποιον τρόπο θα πρέπει όλα αυτά τα devices να διαχωρίζονται μεταξύ τους και να αντιστοιχούν στους κατάλληλους κατόχους τους στο σύστημά μας. Για τον λόγο αυτό υπάρχει Ο GCM GE (ενότητα 3.4.2) ο οποίος με βάση ενός μηχανισμού αίτησης από την εφαρμογή μας , ο GCM , μας προσφέρει σαν απάντηση ένα αναγνωριστικό και μοναδικό ID το οποίο ονομάζεται Αναγνωριστικό Εγγραφής Συσκευής (Registration ID ή RegID).

Βήμα 4^ο- Εγγραφή Χρήστη και Συσκευής στο νέφος

Όπως είδαμε και από τα προηγούμενα βήματα πλέον η εφαρμογή έχει στην μνήμη της(Local storage) όλα τα δεδομένα που τις χρειάζονται για να αντιστοιχίσει έναν ασθενή σε μια συγκεκριμένη συσκευή. Αυτά τα δεδομένα με ένα απλό POST-REST API call(Ενότητα 2.2.2.4) προωθούνται προς το νέφος και αποθηκεύονται κατάλληλα στο JSON storage GE (Ενότητα 3.4.1).

Βήμα 5^ο-Καθορισμός αισθητήρων και κανόνων ανα χρήστη

Στο συγκεκριμένο βήμα ο ιατρός με την χρήση ενός browser έχει την δυνατότητα να ορίσει τους αισθητήρες για κάθε χρήστη ασθενή , καθώς και τους κανόνες που τους διέπουν συμπληρώνοντας απλά μια φόρμα , όπου θέτει το αναγνωριστικό και μοναδικό όνομα ενός αισθητήρα, την οικογένεια του αισθητήρα(π.χ. NoninNonix) καθώς και τις τιμές κατωφλίων κάθε μέτρησης που παράγουν οι BLE αισθητήρες μαζί με τις μονάδες μέτρησής τους. Οι κανόνες ουσιαστικά αποτελούν αυτά τα κατώφλια ,καθώς προσφέρουν μια μέγιστη επιτρεπτή τιμή και μια ελάχιστη επιτρεπτή τιμή όπου στο ενδιάμεσο αυτόν των τιμών προσδιορίζεται η ομαλή κατάσταση ενός ασθενή. Παραδειγματικά μιλώντας ένας χρήστης μπορεί να έχει ως μέγιστο κατώφλι τα 120 bpm καρδιακού ρυθμού. Αν η τιμή αυτή ξεπεραστεί τότε ο χρήστης σημαίνει ότι παραβιάζει αυτόν τον κανόνα και πλέον η κατάσταση του θεωρείται κρίσιμη. Στο βήμα αυτό αν ο καθορισμών αισθητήρων-κανόνων ήδη υπάρχει για κάποιον χρήστη-ασθενή τότε απλά ανανεώνεται.

Βήμα 6^ο-Ανάκτηση πληροφορίας κανόνων στην κινητή συσκευή

Προφανώς στο προηγούμενο βήμα οι κανόνες που έχουν οριστεί από τον γιατρό ενός ασθενούς βρίσκονται στο Local Storage της διεπαφής συστήματος του νέφους. Για τον λόγω αυτό θα πρέπει με κάποιον τρόπο αυτή η χρήσιμη πληροφορία να μεταφερθεί και στο κινητό του χρήστη. Η εφαρμογή Intelligate μέσω ενός απλού αιτήματος http-POST με παράμετρους το αναγνωριστικό του χρήστη και το κατάλληλο κωδικό ταυτοποίησης(Basic authentication) μπορεί να ανακτήσει του κανόνες και τους αισθητήρες που αφορούν τον συγκεκριμένο χρήστη ασθενή και να τις αποθηκεύσει κατάλληλα στην υπο-ενότητα Local Storage του Gateway για περαιτέρω χρήση.

Βήμα 7^ο-Αλληλεπίδραση του χρήστη ασθενή με την εφαρμογή Intelligate

Ο χρήστης ασθενής πλέον δεν έχει να κάνει και πολλές ενέργειες καθώς φορώντας του φορητούς αισθητήρες ,ανοίγοντας την εφαρμογή και συνδέοντας την με τον συγκεκριμένο αισθητήρα, μπορεί να παρακολουθήσει σε πραγματικό χρόνο τις τιμές των μετρήσεων που παράγουν οι αισθητήρες καθώς να επέμβει στον τρόπο λειτουργίας αποστολής των μετρήσεων (on demand) , άμα ο ίδιος δεν νιώθει καλά. Πλην αυτών των 2 ενεργειών ο χρήστης ασθενής δεν μπορεί να προβεί σε καμία άλλη ενέργεια καθώς το σύστημα υλοποιεί με αυτόματο και δυναμικό τρόπο κάθε επιμέρους λειτουργία.

Βήμα 8^ο- Προώθηση δεδομένων στο νέφος

Τα δεδομένα που προωθούνται προς το νέφος είναι οι μετρήσεις που παράγουν οι αισθητήρες(μαζί με το χαρακτηριστικό όνομα του αισθητήρα),η χρονική στιγμή της μέτρησής τους η κατάσταση του χρήστη κάθε χρονική στιγμή με βάση τους κανόνες που τον διέπουν, καθώς και ο τρόπος μετάδοσης.

- **Αναγνωριστικό Όνομα Συσκευής(Device Name ID):** Αποτελεί το αναγνωριστικό όνομα του αισθητήρα.
- **Μετρήσεις(Measures):**Αφορούν το όνομα-τιμή-μονάδα μέτρησης των μετρήσεων πχ(heart rate 80 bpm.Επιπλέον αφορούν την κατάσταση του ασθενούς με βάση κάθε μέτρηση του αισθητήρα. Οι καταστάσεις είναι 2 ειδών.
 - 1)**Ομαλή κατάσταση:** Οι τιμές μετρήσεων που παράγουν οι αισθητήρες κυμαίνονται εντός του πεδίου μέγιστου και ελαχίστου κατωφλίου.
 - 2) **Κίνδυνος :** Οι τιμές μετρήσεων που παράγουν οι αισθητήρες έχουν παραβιάσει κάποιον από τους κανόνες.
- **Χρόνος μέτρησης:** Αφορά την ημερομηνία σε format (Χρόνος-μήνας-ημέρα) και χρόνου (ώρα: λεπτά: δευτερόλεπτα).
- **Τρόπος μετάδοσης πληροφορίας :** Χωρισμένη σε 3 διαφορετικούς τρόπους

1)ON TIME INTERVAL:Η πληροφορία στέλνεται ανα τακτά χρονικά διαστήματα μεγέθους 10 δευτερολέπτων υπολογίζοντας τους μέσους όρους τιμών μετρήσεων του ενδιάμεσου διαστήματος.

2)ON DEMAND : Άμα ο χρήστης νιώθει αδιάθετος τότε στέλνει απευθείας τις μετρήσεις που παράγουν οι αισθητήρες μέσω ενός button , και πλέον το Intelligate λειτουργεί ως απλό Gateway.Δηλαδή κάθε μέτρηση των αισθητήρων μεταφέρεται στο

νέφος την στιγμή που παράγεται επιτρέποντας έτσι την παρακολούθηση των μετρήσεων σε πραγματικό χρόνο.

3)VIOLATION_RULE : Σε αυτήν την περίπτωση έχουμε παραβίαση κανόνα. Δηλαδή η τιμή μέτρησης ξεπέρασε μια τιμή κανόνα. Το Intelligate ενεργοποιεί την αυτοματοποιημένη μέθοδο αποστολής των μετρήσεων ανα δευτερόλεπτο μέχρι την στιγμή όπου οι μετρήσεις επαναφερθούν στο επιτρεπτό εύρος τιμών

Βήμα 9^ο Παρακολούθηση μετρήσεων σε πραγματικό χρόνο στο νέφος

Μια από τις πιο σημαντικές λειτουργίες του συστήματος είναι το γεγονός ότι ο γιατρός έχει την δυνατότητα μέσω ενός browser να παρακολουθεί σε πραγματικό χρόνο τις μετρήσεις που παράγουν οι αισθητήρες για κάθε διαφορετικό ασθενή. Επιπλέον ο γιατρός έχει πρόσβαση στο ιστορικό μετρήσεων κάθε ασθενούς ανα ημέρα μέτρησης για πιο αναλυτική μελέτη της διακύμανσης των ζωτικών δεδομένων. Τέλος ο γιατρός μπορεί να επιβλέπει την κατάσταση κάθε ασθενούς ανεξάρτητα από τον ποιο ασθενή έχει επιλέξει να παρακολουθεί δίνοντας στο σύστημα μας την δυνατότητα άμεσης επέμβασης του , σε μια κρίσιμη στιγμή.

Βήμα 11^ο -12^ο – 13^ο Αποστολή γραπτού μηνύματος από τον γιατρό και εμφάνιση του μηνύματος με την μορφή ενημέρωσης στον ασθενή

Τέλος ο ιατρός άμα κρίνει ότι μια κατάσταση για έναν ασθενή μπορεί να θεωρείται κρίσιμη, τότε έχει την δυνατότητα αποστολής μηνύματος προς την συσκευή του συγκεκριμένου χρήστη με την βοήθεια του Google Cloud messaging GE ο οποίος θα μεριμνήσει να στείλει (ακόμα και σε περίπτωση απουσίας δικτύου) μήνυμα στην συσκευή με την μορφή ενημέρωσης. Ο χρήστης ενημερώνεται με ηχητικό και παλμικό(δόνηση) μήνυμα και μπορεί να διαβάσει το μήνυμα κείμενο του γιατρού και να προβεί στις κατάλληλες ενέργειες που τον συμβουλεύει ο γιατρός του.

4 Υλοποίηση του συστήματος

Σε αυτό το κεφάλαιο πλέον προχωράμε στην περιγραφή της υλοποίησης του συστήματος. Γίνεται ειδική ανάλυση στα επιμέρους κομμάτια της γενικής αρχιτεκτονικής. Πλέον κάθε υπηρεσία ειδικού σκοπού(Generic enabler) και υπηρεσιών που έχει υλοποιηθεί ή παραμετροποιηθεί αναλύεται ως ξεχωριστό κομμάτι βάση της υλοποίησης και λειτουργικότητας που της δώσαμε. Έπειτα γίνεται η πλήρης αναφορά του τρόπου υλοποίησης των μηχανισμών που προσφέρει η εφαρμογή σε συνδυασμό με το γραφικό περιβάλλον(User interface) της εφαρμογής σε επίπεδο συσκευής και διαδικτύου. Στο εν λόγω κεφάλαιο δεν θα προβάλουμε κώδικα καθώς θα υπάρχει σύντομα η δυνατότητα ανάκτησης και μελέτης του μαζί με όλα τα σχόλια που έχουν ενσωματωθεί πάνω σε αυτόν.

4.1 Υλοποίηση ενότητας Έξυπνης πύλης Διαδικτύου

Στο σημείο αυτό θα γίνει αναφορά όλων των υπο ενοτήτων της αρχιτεκτονικής συστήματος Intelligate σε επίπεδο υλοποίησης προβάλλοντας αρχικά το γενικό σχήμα ροής πληροφορίας και ανάλυση του σχήματος.

4.1.1 Εισαγωγή σχήματος αισθητήρα στο σύστημα Intelligate

Στο προηγούμενο κεφάλαιο αναφέραμε , μέσω του διαγράμματος ροής εργασιών(Σχήμα 19) ,ότι το αρχικό βήμα είναι η εισαγωγή ενός νέου αισθητήρα στο σύστημα Intelligate.Ο αρμόδιος τεχνικός ορίζει ένα συγκεκριμένο XML σχήμα το οποίο θα δώσει την δυνατότητα αναγνώρισης των τιμών των χαρακτηριστικών(characteristic values) όπου παράγουν οι αισθητήρες βάση του γενικού προφίλ χαρακτηριστικών(Generic attribute profile-GAAT).Πιο συγκεκριμένα η τιμή του χαρακτηριστικού βρίσκεται σε μια συγκεκριμένη διεύθυνση τύπου χαρακτηριστικού (Attribute type) και χαρακτηρίζεται από ένα συγκεκριμένο και μοναδικό UUID.Για να μπορέσει να αποκωδικοποιηθεί κατάλληλα η χρήσιμη πληροφορία τιμής του χαρακτηριστικού θα πρέπει να γίνει ένας κατάλληλος συνδυασμός των τιμών των

χαρακτηριστικών μέσω μιας συνάρτησης την οποία έχει ορίσει ο ίδιος ο κατασκευαστής της συσκευής.

Με την εισαγωγή ενός XML σχήματος με τα κατάλληλα tags δίνεται η δυνατότητα δυναμικής εισαγωγής νέων σχημάτων αισθητήρων για την αναγνώριση των τιμών των χαρακτηριστικών τους όπως η μέτρηση του καρδιακού παλμού. Η περιγραφή του XML σχήματος περιγράφεται σύμφωνα με τα παρακάτω tags σε δεντρική μορφή.

- **<sensor>** : Αυτό το tag περιλαμβάνει όλες τις BLE συσκευές-αισθητήρες που υποστηρίζει η συσκευή.
- **<device>** : Ένας συγκεκριμένος αισθητήρας όπου χαρακτηρίζεται από την οικογένεια του , device family (πχ Polar H7)
- **<values>**:Αναφέρονται στο όνομα της μέτρησης καθώς και στις μονάδες μέτρησης της. Πχ η συσκευή Polar H7 παράγει μέτρηση καρδιακού παλμού με μονάδα μέτρησης τα bpm.
- **<format>**:Αποτελεί το τύπο (format) που είναι ορισμένες οι τιμές που παράγουν οι αισθητήρες.(Πχ FORMAT_UINT8- ακέραιος 8 bit).
- **<position> & <multi>** :Πολλές φορές οι κατασκευαστές των BLE αισθητήρων ορίζουν μια πιο περίπλοκη συνάρτηση ώστε να μπορεί να αποκωδικοποιηθεί το χαρακτηριστικό το οποίο μας ενδιαφέρει. Πιο συγκεκριμένα η τιμή μπορεί να αναγνωρίζεται από έναν συγκεκριμένο λογικό συνδυασμό διάφορων τιμών ενός χαρακτηριστικού. Τα συγκεκριμένα tags βοηθούν στην δυναμική διαχείριση του συνδυασμού τιμών χαρακτηριστικών προσδιορίζοντας την θέση της τιμής εντός του χαρακτηριστικού(<position>) και πολλαπλασιάζοντας το κατάλληλα με μια ακέραια τιμή(<multi>) .

Για να γίνει πιο κατανοητή η χρήση του XML σχήματος παραθέτουμε το ήδη υπάρχον XML σχήμα υποστήριξης των αισθητήρων Polar H7 και NoninNonix 3250.

```

<? xml version="1.0" encoding="UTF-8" ?>
<sensors>
  <device name="NoninOnix">
    <uuid id="0aad7ea0-0d60-11e2-8e3c-0002a5d5c51b">
      <values name="SPO2" type="%">
        <format>FORMAT_UINT8</format>
        <position>7</position>
        <multi>1</multi>
      </values>
      <values name="pulse_Rate" type="bpm" picture="1">
        <format>FORMAT_UINT8</format>
        <position>[8,9]</position>
        <multi>[256,1]</multi>
      </values>
    </uuid>
  </device>

  <device name="Polar H7">
    <uuid id="00002a37-0000-1000-8000-00805f9b34fb">
      <values name="pulse_Rate" type="bpm">
        <format>FORMAT_UINT8</format>
        <position>1</position>
        <multi>1</multi>
      </values>
    </uuid>
  </device>
</sensors>

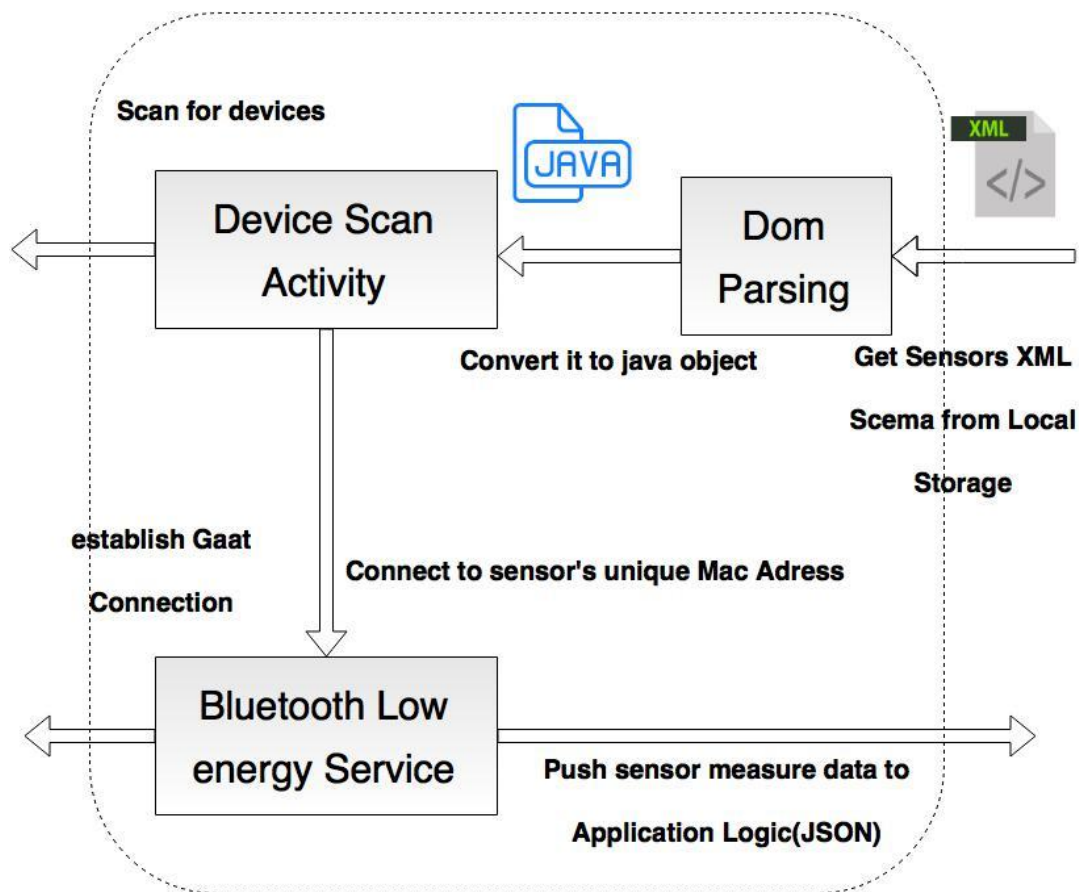
```

Έχουμε 2 συσκευές , όπου η πρώτη είναι η NoninNonix με ένα UUID χαρακτηριστικού η οποία παράγει το εύρος ποσοστού οξυγόνου στο αίμα (SPO2) καθώς και τους καρδιακούς παλμούς(pulse_Rate) τις μονάδες μέτρησής τους καθώς και το format. Αντίστοιχα η δεύτερη συσκευή είναι η Polar H7 ή οποία παράγει μόνο τον καρδιακό ρυθμό ως τιμή χαρακτηριστικού σε ένα διαφορετικό UUID.Για την πρώτη συσκευή ο τρόπος αποκωδικοποίησης της τιμής του καρδιακού ρυθμού είναι να συνδυάσουμε προσθέτοντας την τιμή του χαρακτηριστικού στην 8^η θέση πολλαπλασιασμένη με 256 με την τιμή του χαρακτηριστικού στην 9 θέση (πολλαπλασιασμένημε1).

$$SpO2 = \text{Format_UINT8} (256 * \text{CharacteristicPosition} (8) + \text{CharacteristicPosition} (9))$$

Το XML σχήμα κάθε αισθητήρα βρίσκεται εντός του Asset Folder και μπορεί να ανανεώνεται τακτικά από τον διαχειριστή με αποτέλεσμα την επεκτασιμότητα της εφαρμογής, χωρίς να χρειάζεται να γράφεται εκ νέου κώδικας εντός του project.

4.1.2 Sensor Data Collector



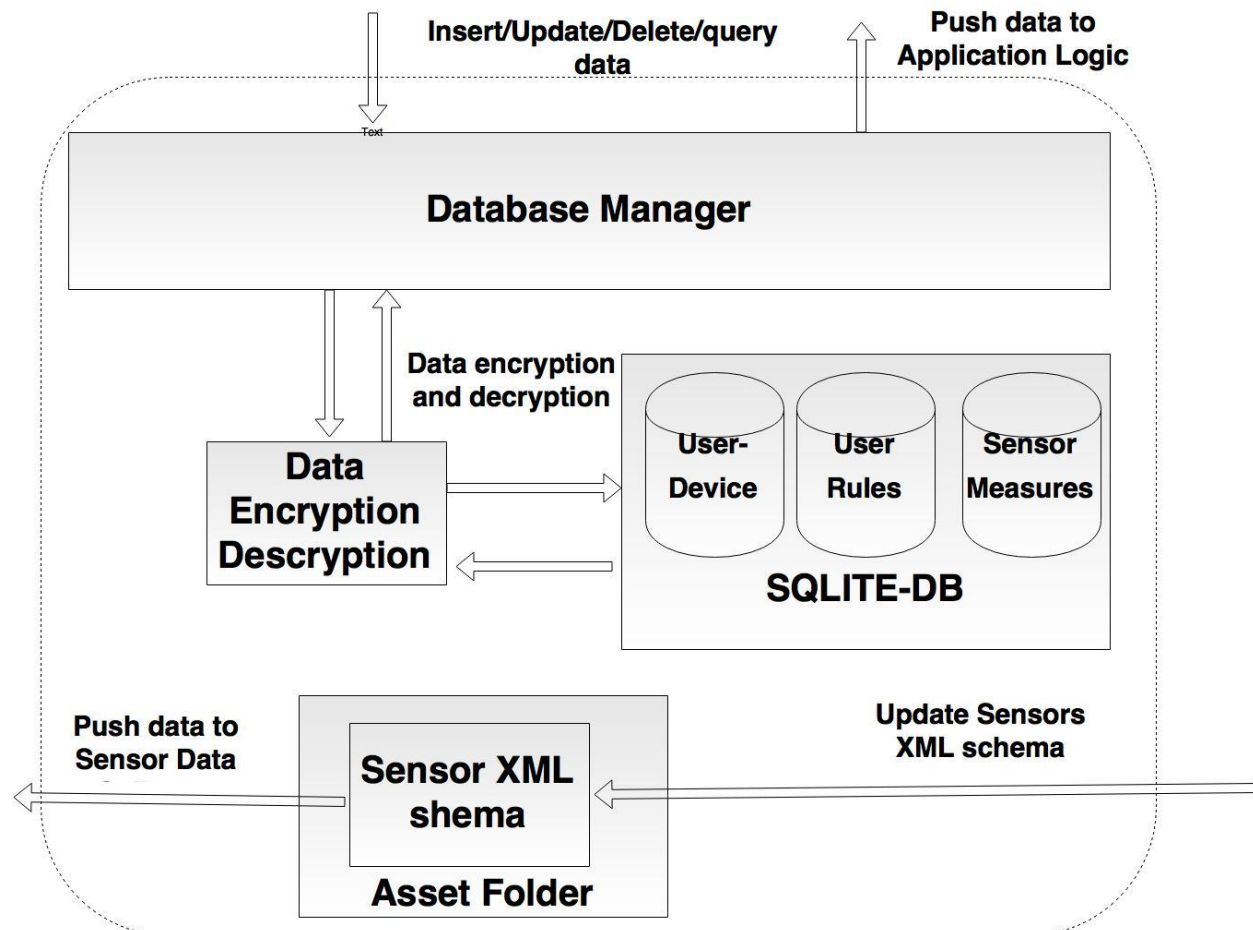
Σχήμα 20-Sensor Data Collector

Με την ενεργοποίηση της εφαρμογής και ενώ ο χρήστη ασθενής έχει φορέσει τους wearable αισθητήρες η δραστηριότητα DeviceScanActivity δέχεται και αποθηκεύει σε κατάλληλη δομή δεδομένων το XML σχήμα των αισθητήρων που υποστηρίζει η συσκευή αφού έχει γίνει ήδη η μετατροπή του σε αντικείμενο Java(Java Object),μέσω της λειτουργίας Dom Parsing[21] , με στόχο την αναγνώριση τιμών των χαρακτηριστικών(Characterists value) που παράγουν οι αισθητήρες κάθε χρονική στιγμή. Έπειτα η δραστηριότητα DeviceScanActivity ενεργοποιεί τον μηχανισμό ανίχνευσης περιφερειακών συσκευών(scanning) για να εντοπίσει ποιες περιφερειακές συσκευές διαφημίζουν τον εαυτό τους(advertising).Η διαδικασία scanning διαρκεί συνολικά 10 δευτερόλεπτα.Η δραστηριότητα DeviceScanActivity προσφέρει στον χρήστη ένα Γραφικό περιβάλλον(User Interface) έτσι ώστε να μπορεί να επιτευχθεί η σύνδεση της κεντρικής συσκευής(tablet) με μια συγκεκριμένη περιφερειακή συσκευή(αισθητήρες) μέσω μιας μοναδικής διεύθυνσης MAC address.Ο χρήστης έχει την δυνατότητα επιπλέον να εκκινήσει(και να σταματήσει) ξανά την διαδικασία scanning όσες φορές αυτός θέλει μέχρι να ανιχνευθεί η επιθυμητή συσκευή στην οποία ο ίδιος θέλει να συνδεθεί.

Με το πού επιτευχθεί η σύνδεση ενεργοποιείται αυτόματα η υπηρεσία BluetoothLowEnergyService η οποία με την σειρά της εγκαθιστά μια σύνδεση μεταξύ περιφερειακής και κεντρικής συσκευής τύπου (client-Server) για την αναγνώριση του Generic Access profile(GAAP) της περιφερειακής συσκευής. Μέσω της κατάλληλης δομής δεδομένων που έχει υποθηκευμένο το σχήμα των αισθητήρων σε μορφή αντικειμένων JAVA γίνεται σύγκριση του μοναδικού UUID των χαρακτηριστικών με αποτέλεσμα την αναγνώριση των τιμών-χαρακτηριστικών που παράγει ο συγκεκριμένος αισθητήρας από την κεντρική συσκευή. Έπειτα ακολουθεί η κατάλληλη αποκωδικοποίηση της χρήσιμης πληροφορίας που περιλαμβάνει την αναγνώριση της μέτρησης (Measure Name), την τιμή της(Measure Value) καθώς και την μονάδα μέτρησης της(Measure Unit).

Οι αισθητήρες παράγουν δεδομένα ανα δευτερόλεπτο ,και με την χρήση ενός Broadcast Receiver ,η πληροφορία τους προωθείται προς την ενότητα Application Logic της εφαρμογής αφού μετασχηματιστούν κατάλληλα σε μορφή JSON string μέχρι η συσκευή να αποσυνδεθεί από το σώμα του χρήστη ασθενή (Disconnection).

4.1.3 Local Storage



Σχήμα 21-Local Storage

Η υλοποίηση του Local Storage στην εφαρμογή μας αποτελεί μια από τις σημαντικότερες λειτουργίες που προσφέρει η εφαρμογή Intelligate στο σύστημά μας και διαχωρίζει την συσκευή μας από ένα απλό Gateway. Μέσω κατάλληλων αποθηκευτικών μέσων επιτρέπεται η εισαγωγή, ανανέωση διαγραφή καθώς και ο κατάλληλος ορισμός αιτημάτων έτσι ώστε να έχουμε μια αποτελεσματική ροή πληροφορίας στην εφαρμογή μας. Αρχικά η πληροφορία περνάει από κατάλληλο encryption κατά την εισαγωγή της και encryption κατά την εξαγωγή της καθώς οι μετρήσεις ζωτικής σημασίας θεωρούνται ευαίσθητες για κάθε χρήστη και δεν πρέπει να αποσπαστούν από την βάση δεδομένων με τις κανονικές τους τιμές. Η χρήσιμη πληροφορία αποθηκεύεται εσωτερικά της συσκευής μέσω βάσης δεδομένων SQLite σε διάφορα ανεξάρτητα tables μεταξύ τους. Επιπλέον στην ενότητα Local Storage περιλαμβάνεται και το αρχείο XML γενικής περιγραφής των αισθητήρων που υποστηρίζονται από την εφαρμογή μέσα σε έναν φάκελο Asset Folder ο οποίος μπορεί να ανανεώνεται κάθε χρονική στιγμή.

Για να γίνει πιο κατανοητή ή χρήση του Local Storage θα πρέπει να γίνει η περιγραφή των επιμέρους εσωτερικών κομματιών του.

Database manager

Αποτελεί το διαχειριστή του component Local Storage. Σε αυτό το σημείο ορίζονται οι κατάλληλοι πίνακες (tables) για κάθε διαφορετικό τύπου δεδομένων που θέλουμε να αποθηκεύσουμε στην βάση μας. Επιπλέον περιλαμβάνει το σύνολο των αιτημάτων(queries) για την ορθή και συγκεκριμένη ανάκτηση πληροφορίας από την βάση δεδομένων. Μερικά από αυτά τα αιτήματα είναι η addSensorMeasure() η οποία αποθηκεύει την μέτρηση ενός αισθητήρα , deleteSensorMeasure() διαγραφή μετρήσεων ενός συγκεκριμένου αισθητήρα, ReturnSensors() επιστροφή των χαρακτηριστικών id αισθητήρων όπου έχουν αποθηκευμένες μετρήσεις , καθώς και αρκετές ακόμα βοηθητικές λειτουργίες που μπορούν εύκολα να αναγνωστούν από τον κώδικα της εργασίας μας.

Data encryption και Decryption

Το συγκεκριμένο κομμάτι πάρθηκε αυτούσιο από το Specific Enabler του Fiware Local Storage SE. Πρόκειται για ένα component το οποίο δέχεται σαν είσοδο συγκεκριμένου είδος πληροφορία κειμένου , και αυτό το κωδικοποιεί- αποκωδικοποιεί κατάλληλα βάση του με βάση το AES(Advanced Encryption standard)[23]. Το συγκεκριμένο κομμάτι

δέχεται ένα key μαζί με το plain text όπου το κωδικοποιεί και το αποκωδικοποιεί κατάλληλα με βάση με βάση το AES. Οι 2 μέθοδοι που χρησιμοποιούνται έχουν την παρακάτω μορφή

- Aes128_encrypt (key, cleartext) : Κωδικοποίηση με ορισμένο key και cleartext και μεταφορά στο κατάλληλο table της βάσης δεδομένων.
- Aes128_decrypt (key, crypto): Αποκωδικοποίηση του ορισμένου text και την κρυπτογραφημένης πληροφορίας και προώθησή της στο Application logic.

SQLITE-DB

Αποτελεί την βάση δεδομένων και διαθέτει τα ξεχωριστά και διάφορα tables.

1)User-Device: Περιλαμβάνει το αναγνωριστικό id του ασθενούς χρήστη καθώς και της αντιστοιχισμένης συσκευής του(email και Device Registration ID).

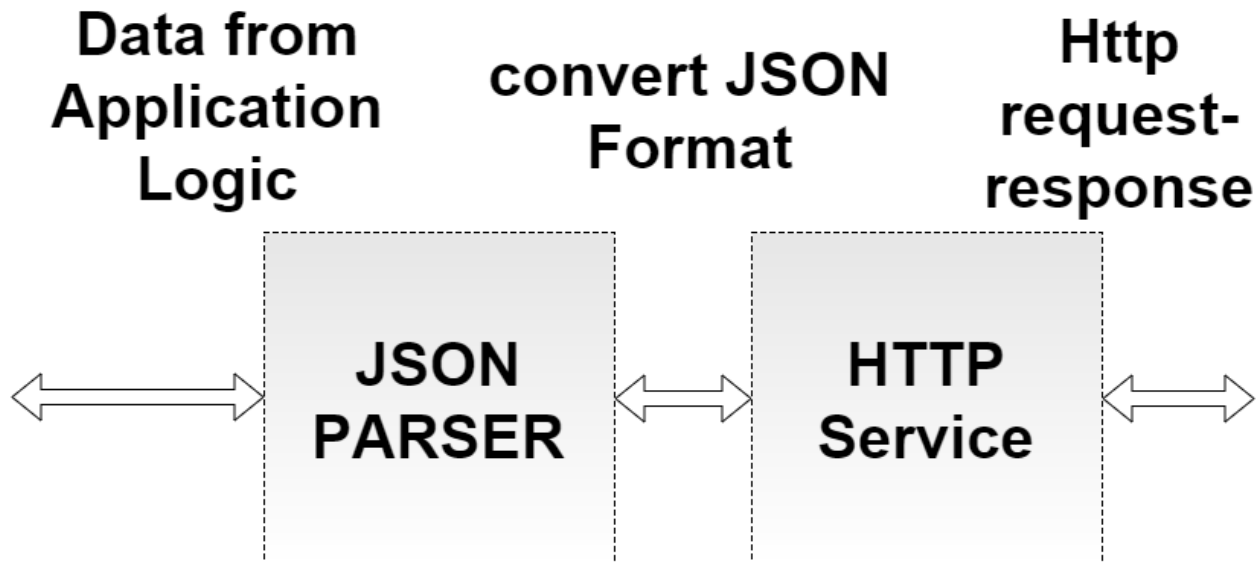
2)User-Rules: Περιλαμβάνει το αναγνωριστικό id χρήστη(email) και αισθητήρων(Device Name ID) μαζί με τα επιτρεπτά όρια τιμών που ορίζουν τους κανόνες ανά χρήστη(max measure – min measure).

3)Sensor-Measures: Σε περίπτωση απουσίας δικτύου η υπηρεσία του Local Storage ενεργοποιείται για να αποθηκευτούν τοπικά οι μετρήσεις των αισθητήρων που δεν στάλθηκαν στο νέφος. Συγκεκριμένα το table Sensor Measures περιλαμβάνει τις μέσες τιμές κάθε αισθητήρα ανά δέκα δευτερόλεπτα περιέχοντας πεδία ημερομηνίας(Date) χρονικής έναρξης μέτρησης(Starting Measure) και λήξης(Ending Measure) .

Asset Folder

Το XML σχήμα κάθε αισθητήρα βρίσκεται εντός του Asset Folder και μπορεί να ανανεώνεται τακτικά από τον διαχειριστή με αποτέλεσμα την επεκτασιμότητα της εφαρμογής, χωρίς να χρειάζεται να γράφεται εκ νέου κώδικας εντός του project.

4.1.4 Connectivity Service



Σχήμα 22-Connectivity Service

Το Connectivity Service ουσιαστικά αποτελεί το δίαυλο επικοινωνίας μεταξύ της κεντρικής συσκευής (Gateway) με την διεπαφή συστήματος του cloud. Τα δεδομένα τα οποία έρχονται από την ενότητα λογικής συστήματος, μετασχηματίζονται κατάλληλα με την μορφή JSON string μέσω της ΥΠΟ ενότητας JSON Parser και έπειτα με τα κατάλληλα HTTP calls που διαθέτει η ασύγχρονη υπηρεσία HTTP Service αποστέλλονται με ασφάλεια στο νέφος. Το Http Service επιπλέον κάνει έλεγχο των απαντήσεων του server(υπο-ενότητα 2.2.4.4) ενημερώνοντας το σύστημα της εφαρμογής κατάλληλα. Οι λειτουργίες του Connectivity Service σε επίπεδο συσκευής είναι οι παρακάτω

- **Ανάκτηση Device Registration ID από τον Google Cloud messaging Server:** Γίνεται μέσω των κατάλληλων εργαλείων που προσφέρει η βιβλιοθήκη google cloud messaging.

Αποστολή email και RegID στο νέφος: Με ένα από POST- στην διεύθυνση

<http://147.27.50.154/Intelligence/GCM/InsertDeviceReg.php> με τα κατάλληλα headers ο χρήστης καθώς και η συσκευή του αποθηκεύονται κατάλληλα στο νέφος και συγκεκριμένα στον JSON Storage GE. Το περιεχόμενο του POST είναι σε JSON format και έχει την παρακάτω δομή.

```
{
  "name": "user1@gmail.com",
  "reg_id": "APA91bE1U9LUy7J2mfwvCRUMPOosVpbN2kybRJ_Dk_Upz7eUu6G1gcSkMI87t
QwZz8pznTNNZCMgsMCfwrL7umcdF-Y5BNWJHtr_1ARF8rngYOKmC1yI-
W0H1QAsyluhltJ5PX-yaEmi"
```


}

- **Λήψη κανόνων ανα χρήστη:** Με ένα απλό POST RESTFULL API CALL στην διεύθυνση: `http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/finds` ανακτώνται οι κατάλληλοι κανόνες αισθητήρων που διέπουν τους χρήστες ασθενείς. Τα δεδομένα αυτά προωθούνται προς το Application Logic το οποίο με την σειρά του τα μεταφέρει για αποθήκευση προς τον Local Storage για περαιτέρω χρήση.
- **Αποστολή Μετρήσεων Αισθητήρων :** Πραγματοποιείται η αποστολή των δεδομένων ανάλογα με τον τρόπο αποστολής.

1)OnTimeInterval : Αποστέλλεται ο μέσος όρος μετρήσεων ανα 10 δευτερόλεπτα

2)On demand : Αποστέλλεται η τιμή που παράγει ο αισθητήρας εκείνη την χρονική στιγμή(on real time) μέχρι ο χρήστης αποσυνδέσει τον αισθητήρα από πάνω του.

3)Violation Rule Αποστέλλεται η τιμή που παράγει ο αισθητήρας εκείνη την χρονική στιγμή(on real time) μέχρι να σταματήσει να γίνεται παραβίαση κανόνα για έναν συγκεκριμένο ασθενή.

Το μήνυμα κειμένου του αιτήματος έχει JSON format. Ένα χαρακτηριστικό παράδειγμα.

```

{
  "nameID": "user1@gmail.com",
  "SensorId": "Nonin3230_501860219",
  "State": "ON_TIME_INTERVAL",
  "Date": "2015-10-13",
  "start_time": "11:57:47",
  "end_time": "11:57:57",
  "Measures": [
    {
      "SensorMeasure": "pulse_Rate",
      "MeasureValue": 86,
      "MeasureUnit": "bpm",
      "Condition": "OK"
    },
    {
      "SensorMeasure": "SPO2",
      "MeasureValue": 98,
      "MeasureUnit": "%",
      "Condition": "OK"
    }
  ]
}

```

Επιπλέον είναι η δυνατή η αποστολή συνολικών μετρήσεων σε μορφή πίνακα JSON σε περίπτωση όπου υπάρχουν αποθηκευμένες τιμές μετρήσεων των αισθητήρων που δεν έχουν αποσταλεί στο νέφος.

```
[
  {
    "nameID": "user1@gmail.com",
    "SensorId": "Nonin3230_501860219",
    "State": "ON_TIME_INTERVAL",
    "Data": "2015-10-21",
    "start_time": "12:50:00",
    "end_time": "12:50:10",
    "Measures": [
      {
        "SensorMeasure": "pulse_Rate", "MeasureValue": 82.6, "MeasureUnit": "bpm", "Condition": "OK"
      },
      {
        "SensorMeasure": "SPO2", "MeasureValue": 98, "MeasureUnit": "%", "Condition": "OK"
      }
    ]
  },
  {
    "nameID": "user1@gmail.com",
    "SensorId": "Nonin3230_501860219",
    "State": "ON_TIME_INTERVAL",
    "Data": "2015-10-21",
    "start_time": "12:50:11",
    "end_time": "12:50:21",
    "Measures": [
      {
        "SensorMeasure": "pulse_Rate", "MeasureValue": 88.5, "MeasureUnit": "bpm", "Condition": "OK"
      },
      {
        "SensorMeasure": "SPO2",
        "MeasureValue": 98,
        "MeasureUnit": "%",
        "Condition": "OK"
      }
    ]
  }
]
```

Η ενεργοποίηση ή όχι του Connectivity Service γίνεται από την Υπο ενότητα Application Logic ή οποία ελέγχει συνεχώς την διαθεσιμότητα του δικτύου. Η ίδια ενότητα υπάρχει και από μεριάς νέφους η οποία μετατρέπει τα JSON αντικείμενα σε κατάλληλες δομές δεδομένων.

4.1.5 Notification Service

Η υπο-ενότητα της γενικής αρχιτεκτονικής , Notification Service , αποτελεί μια υπηρεσία η οποία ενημερώνει τους χρήστες ασθενείς με τα κατάλληλα ειδοποιητικά μηνύματα Toast messages καθώς και alerts για σημαντικές αλλαγές που συμβαίνουν στην ροή λειτουργίας της εφαρμογής. Οι λειτουργίες αυτές είναι διάφορες και οι πιο βασικές από αυτές είναι:

- **Ενημέρωση ενεργοποίησης Bluetooth συσκευής:** Για να γίνει η ομαλή λειτουργία ης εφαρμογής Intelligate θα πρέπει να είναι ενεργοποιημένο το Bluetooth της συσκευής ώστε να γίνουν ανιχνεύσιμες οι περιφερειακές συσκευές. Σε περίπτωση όπου το bluetooth της συσκευής δεν είναι ενεργοποιημένο , τότε το Notification Service ενημερώνει τον χρήστη κατάλληλα για την ενεργοποίηση του.
- **Αλλαγή καταστάσεων:** Ανάλογα με την κάθε περίπτωση της εφαρμογής εκτελούνται διαφορετικές διεργασίες. Παραδειγματικά μιλώντας σε περίπτωση απουσία δικτύου το Notification Service Ειδοποιεί τον χρήστη ασθενή κατάλληλα ότι δεν υπάρχει δίκτυο και ενεργοποιείται η υπηρεσία του Local Storage για την αποθήκευση των χαμένων μετρήσεων που παράγουν οι αισθητήρες.
- **Εμφάνιση αναγκάων απαντήσεων διακομιστή:** Σε ορισμένες περιπτώσεις όπου εκτελούμε ένα HTTP request το Notification Service εμφανίζει τον κωδικό και το μήνυμα της απάντησης του διακομιστή(Server).
- **Ενημέρωση χρήστη ασθενή για το μήνυμα του γιατρού σε μορφή ειδοποίησης(alert):** Ο Γιατρός μπορεί να τυπώσει και να αποστείλει ένα μήνυμα κείμενο στην συσκευή του χρήστη μέσω του Google cloud messaging Server.Μέσω του Notification Service γίνεται ηχητική και παλμική ειδοποίηση. Ο χρήστης με την σειρά του μπορεί να επιλέξει αυτήν την ενημέρωση για να προβληθεί το μήνυμα κειμένου του γιατρού σε μια νέα δραστηριότητα(Activity).

Το Notification Service για ορισμένες λειτουργίες του 'όπως η τελευταία είναι συνεχώς ενεργοποιημένο ακόμα και αν η εφαρμογή δεν εκτελείται , δίνοντας έτσι την δυνατότητα συνεχούς χρόνου επικοινωνίας του νέφους με την κεντρική συσκευή.

4.1.6 Application Logic(Intelligate)

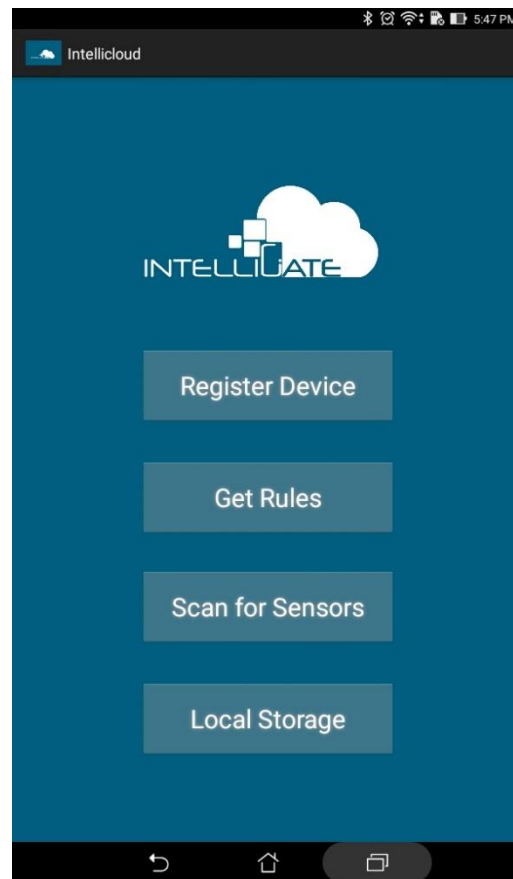
Η ενότητα Application Logic αποτελεί κομμάτι κεντρικής διαχείρισης όλων των υπηρεσιών που αλληλεπιδρούν με την κυρίως εφαρμογή. Είναι ο εγkéφαλος της εφαρμογής που δίνει εντολές έναρξης και λήξης σε κάθε διαφορετική υπηρεσία. Επιπλέον διαθέτει στο χρήστη και το γραφικό περιβάλλον της εφαρμογής Intelligate όπως θα φανεί στην επόμενη ΥΠΟ ενότητα. Επιπλέον λειτουργίες του Application Logic εκτός από την διαδικασία ενεργοποίησης κάθε ξεχωριστής υπηρεσίας και προσφορά του UI είναι οι εξής.

- **Μηχανισμός Διαχείρισης κανόνων(Rule management) :** Ο μηχανισμός αυτός καλύπτει τις λογικές συγκρίσεις των μετρήσεων που παράγουν οι αισθητήρες με βάση τους κανόνες που έχουν ληφθεί από νέφος.
- **Μηχανισμός Επεξεργασίας μετρήσεων :** Σε αυτό το στάδιο γίνεται κατάλληλη επεξεργασία των μετρήσεων του αισθητήρα όπως υπολογισμός του μέσου όρου τιμών ανα τακτά χρονικά διαστήματα, εισαγωγή χρονικών χαρακτηριστικών μέτρησης(Ημερομηνία-ώρα-λεπτά-δευτερόλεπτα) καθώς και την κατάσταση του ασθενούς κάθε χρονική στιγμή λειτουργίας της εφαρμογής.
- **Μηχανισμός Έλεγχου διαθεσιμότητας Internet:** Μέσω της μεθόδου `IsConnected()` γίνεται έλεγχος της διαθεσιμότητας του Internet κάθε χρονική στιγμή.
- **Μηχανισμός Αλλαγής καταστάσεων αποστολής:** Με κριτήριο της επιλογής του χρήστη ή την κατάσταση του γίνεται η αλλαγή τρόπου μετάδοσης πληροφορίας με βάση τις 3 διαφορετικές καταστάσεις που αναλύθηκαν στην υπο-ενότητα 3.5 (ON_TIME-INTERVAL,ON_DEMANT,RULE_VIOLATION).
- **Συγχρονισμός συστήματος:** Όσο η συσκευές είναι συνδεδεμένες στην εφαρμογή Intelligate η λογική διαχείρισης συστήματος με την βοήθεια ενός χρονικού μετρητή, κάνει τις κατάλληλες διεργασίες με βάση την αρχική σχεδίαση του συστήματος. Για παράδειγμα στον τρόπο μετάδοσης

ON_TIME_INTERVAL το Application Logic ενεργοποιεί το Connectivity Service το 10 ο δευτερόλεπτο νέας μέτρησης αποστέλλοντας του τον μέσο όρο μετρήσεων όλων των προηγούμενων τιμών.

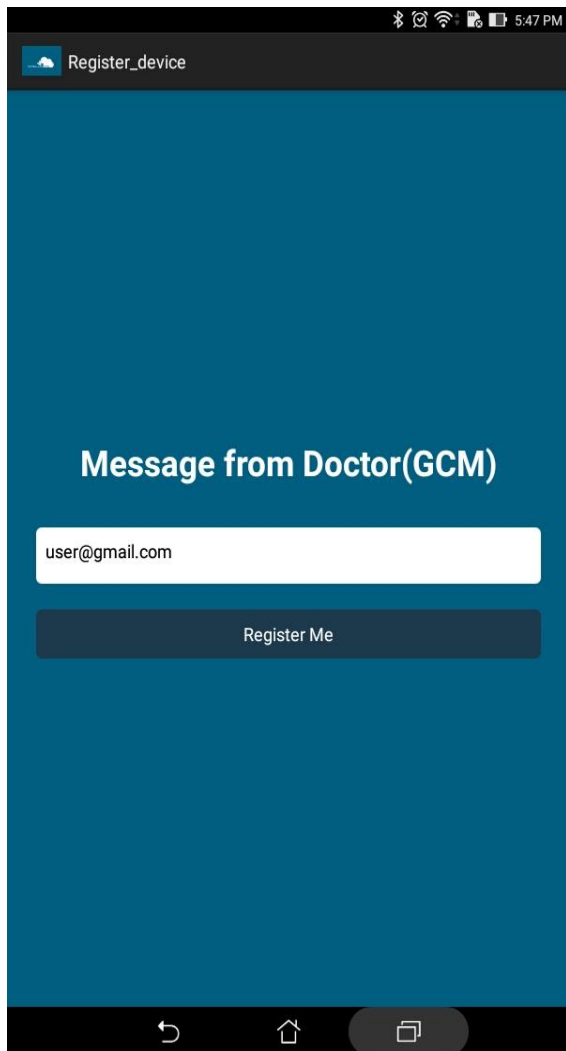
4.1.7 Intelligate Application και γραφικό περιβάλλον (User Interface)

Η ενότητα λογικής εφαρμογής δίνει την δυνατότητα αλληλεπίδρασης του χρήστη με την εφαρμογή Intelligate παρακάτω θα δείξουμε ορισμένα βασικά γραφικά στιγμιότυπα της εφαρμογής σε επίπεδο κινητής πλατφόρμας με βάση την ανάλυση ροής δεδομένων και ενεργειών του συστήματος(3.5).

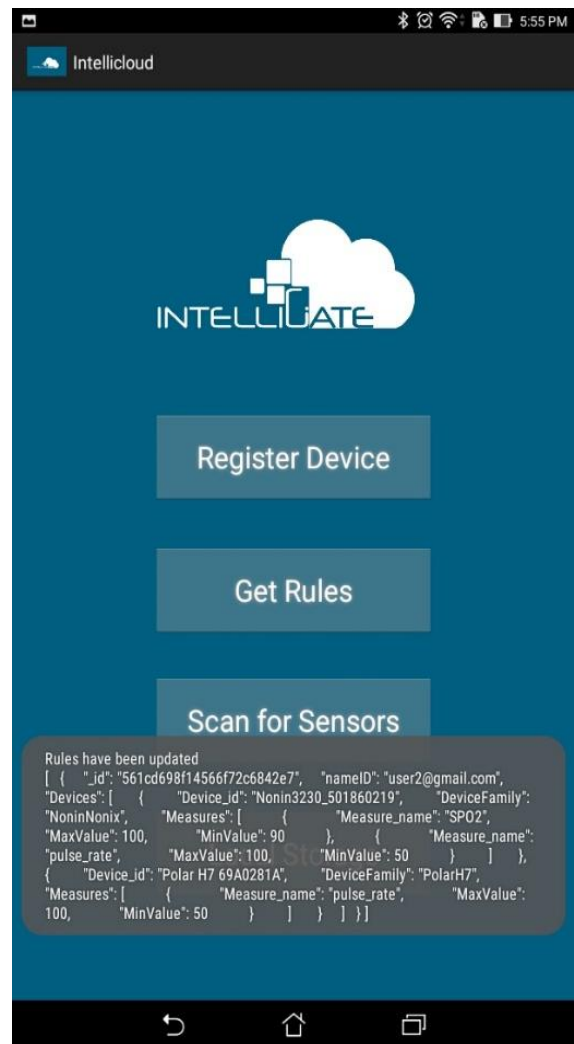


Σχήμα 23-Η αρχική δραστηριότητα-menu της εφαρμογής.

- Register Device: Εισαγωγή ενός νέου χρήστη μαζί με την συσκευή του στο back-end cloud.
- Get Rules : Λήψη κανόνων από το back-end cloud στην συσκευή
- Scan For Sensors : Σύνδεση με έναν αισθητήρα
- Local Storage: Προβολή τιμών που δεν έχουν αποσταλεί στο νέφος.

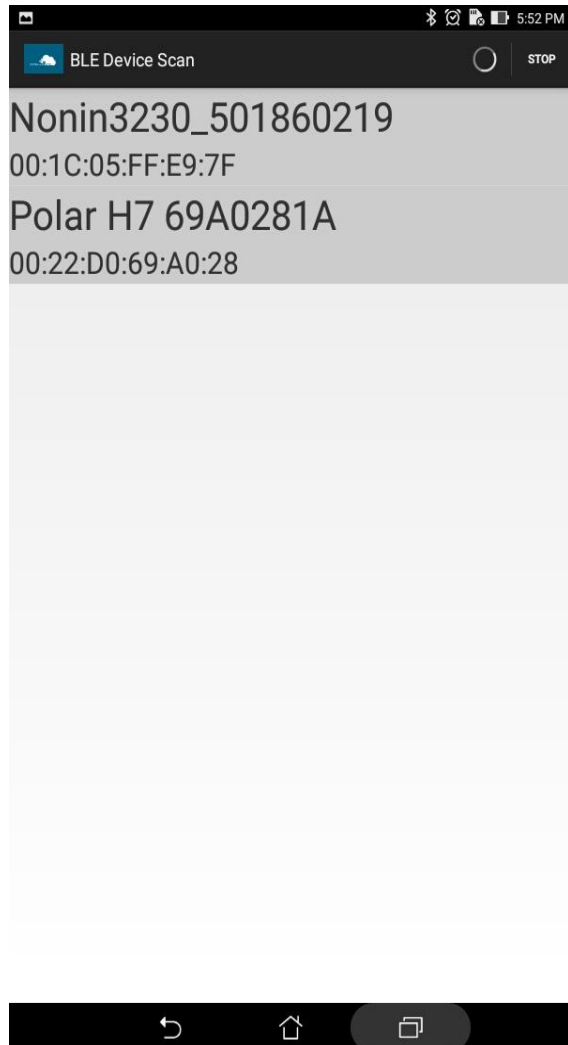


Σχήμα 23
Register Device

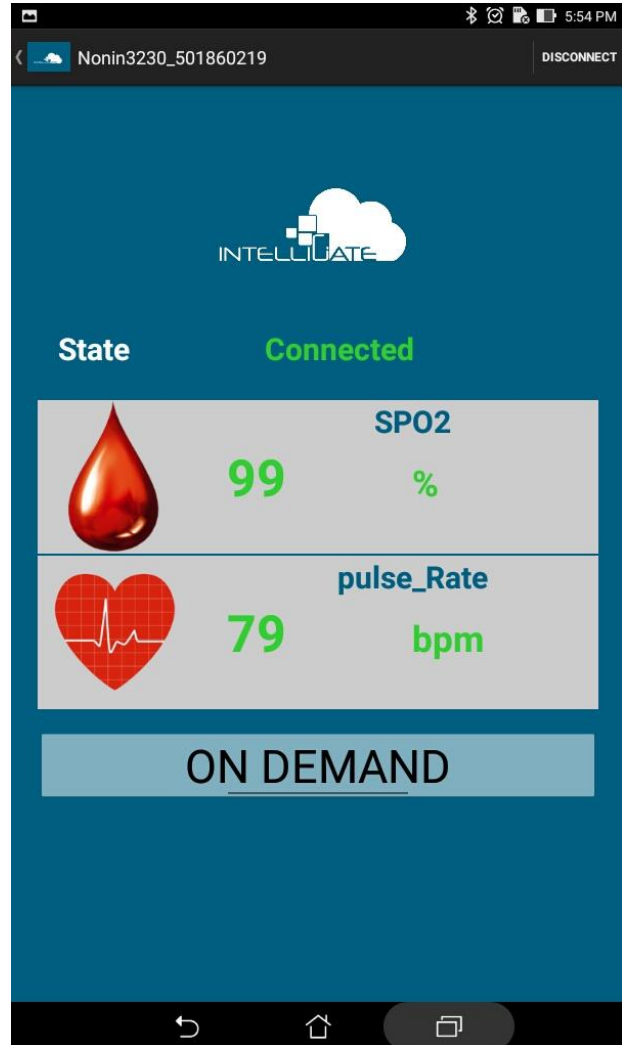


Σχήμα 24-
Get

Rules(+Notification)



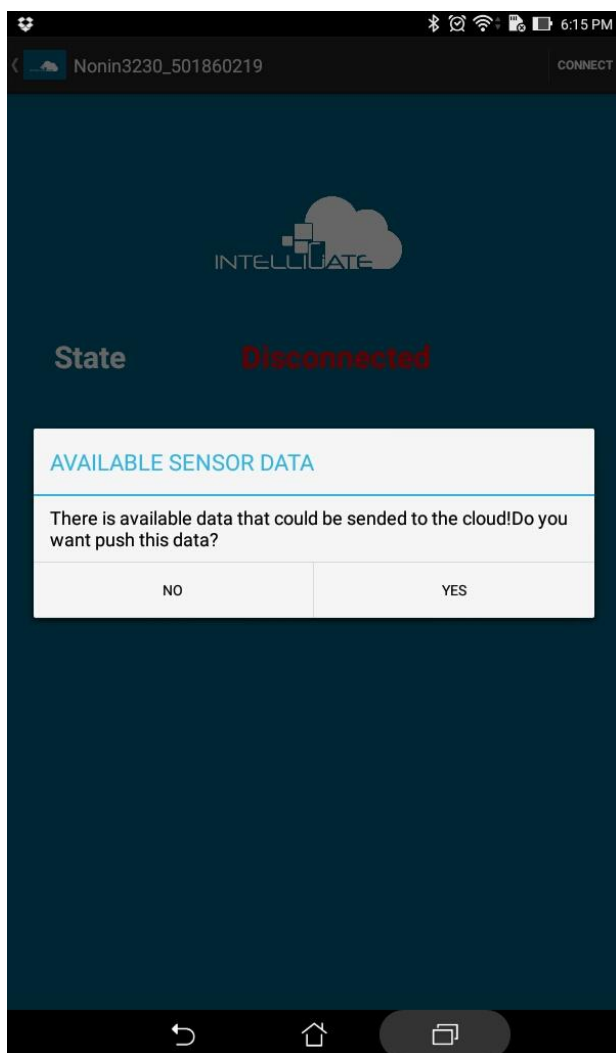
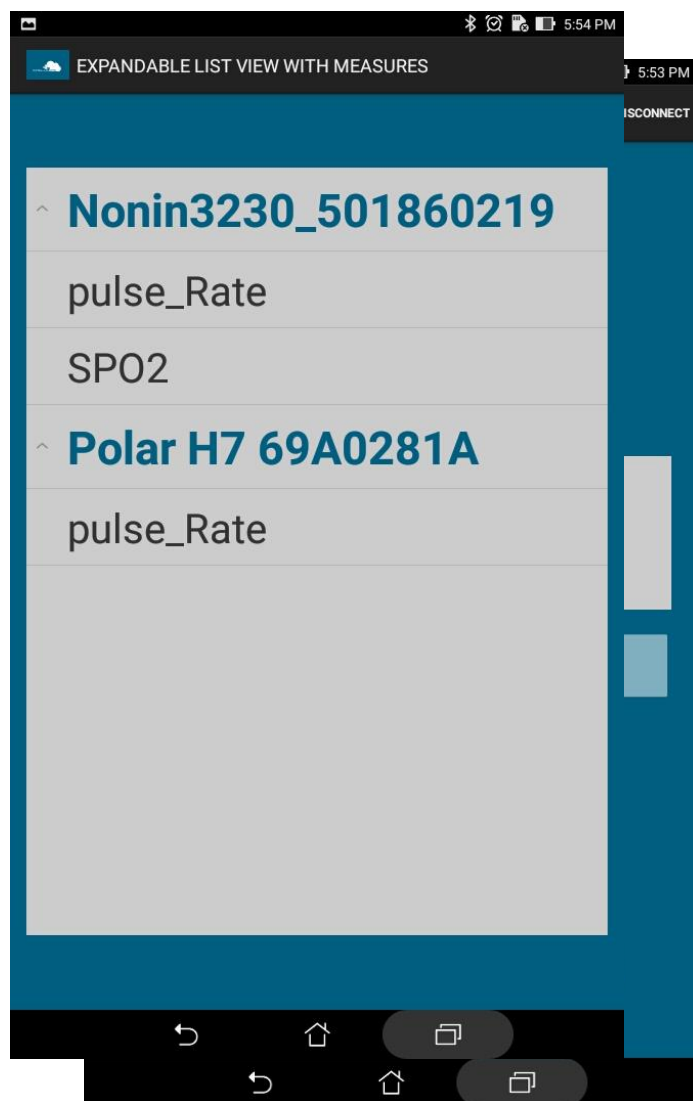
Σχήμα 25-Scan For Sensors



Σχήμα 26-Connected to device

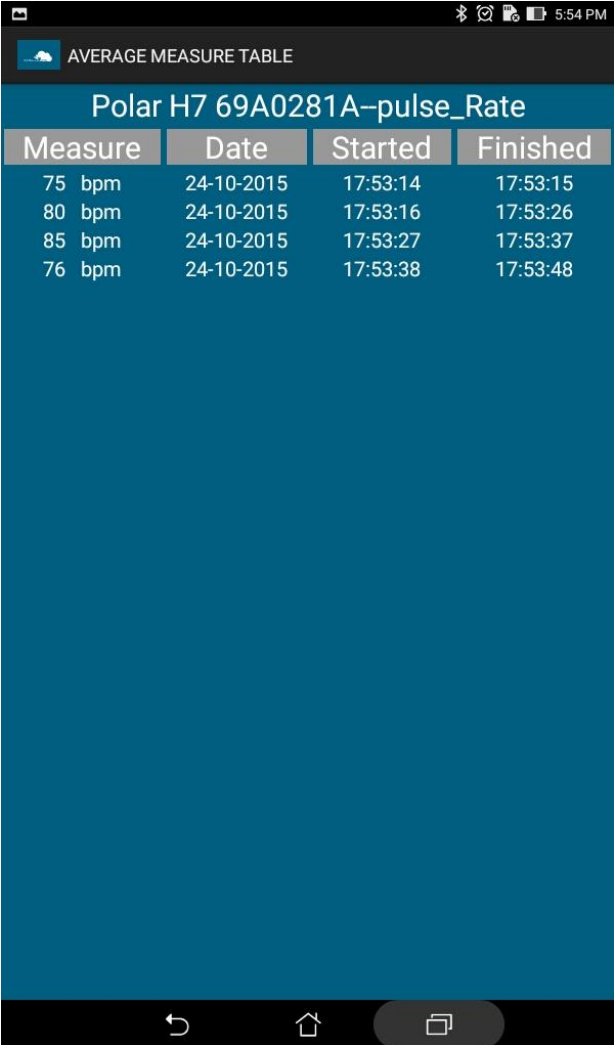
Σχήμα 27-Περίπτωση απουσίας δικτύου

Σχήμα 28-Ειδοποίηση αποστολής
χαμένης πληροφορίας στο
νέφος



Σχήμα 29-30 Επιλογή προβολής ιστορικού

Χαμένης πληροφορίας ανα αισθητήρα-μέτρηση



AVERAGE MEASURE TABLE

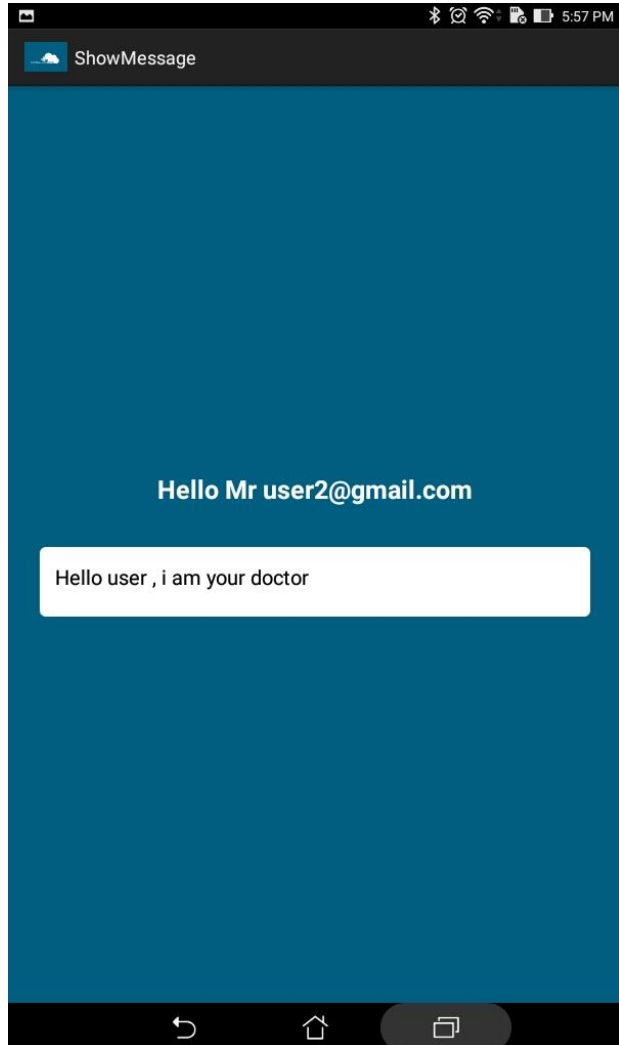
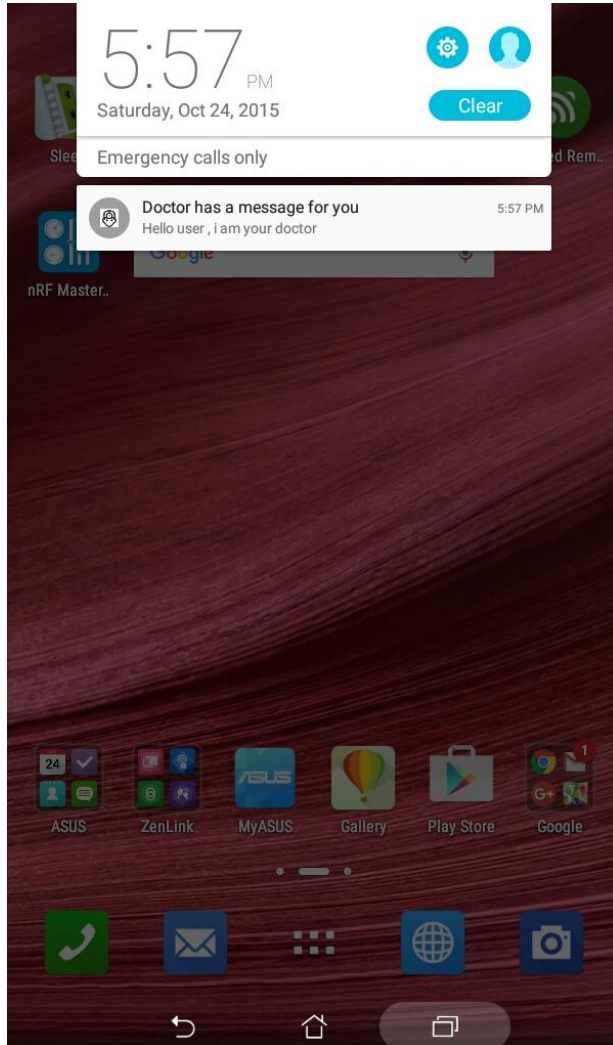
Polar H7 69A0281A-pulse_Rate

Measure	Date	Started	Finished
75 bpm	24-10-2015	17:53:14	17:53:15
80 bpm	24-10-2015	17:53:16	17:53:26
85 bpm	24-10-2015	17:53:27	17:53:37
76 bpm	24-10-2015	17:53:38	17:53:48

Σχήμα 31-Alert γιατρού

Σχήμα 32-Εμφάνιση μηνύματος
γιατρού

4.2 Υλοποίηση ενότητας Διεπαφή Διαχείρισης Συστήματος Νέφους



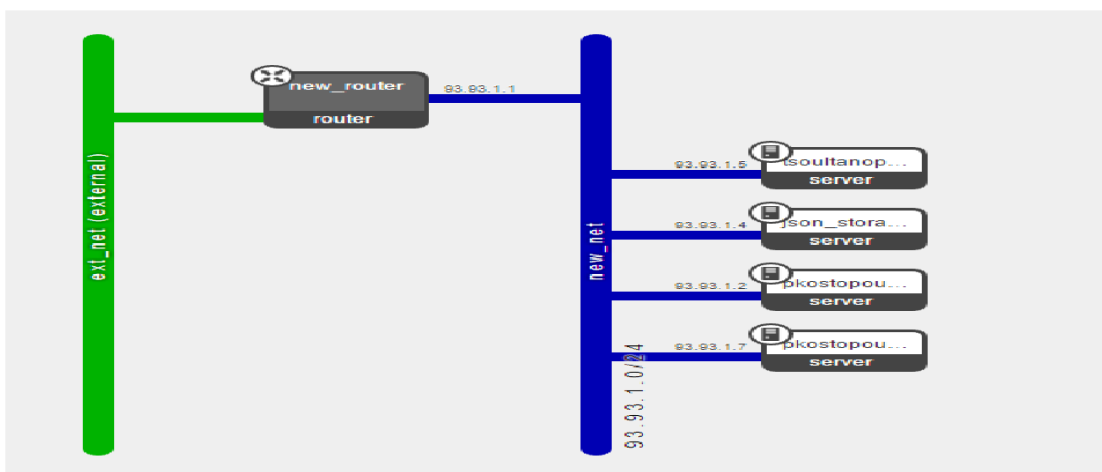
Στο σημείο αυτό θα γίνει αναφορά στην δημιουργία και παραμετροποίηση μιας εικονικής μηχανής , καθώς και τις τροποποιήσεις που υπέστησαν οι Υπηρεσίες γενικού σκοπού για την εξυπηρέτηση των αναγκών της εργασίας μας. Τέλος θα εξηγηθούν οι βασικοί μηχανισμοί της λογικής συστήματος(Back-end). Ξανατονίζουμε ότι η συγκεκριμένη ενότητα υλοποιήθηκε για την επιβεβαίωση της ορθής λειτουργίας της android εφαρμογής Intelligate.

4.2.1 Παραμετροποίηση εικονικών μηχανών

Η υποδομή Νέφους Intellicloud μέσω των υπηρεσιών σε μορφή IaaS που προσφέρει μας έδωσε τη δυνατότητα να φιλοξενήσουμε το σύστημα μας στο Νέφος, δημιουργώντας εικονικές μηχανές. Αρχικά, για να αποκτήσουμε πρόσβαση στη συγκεκριμένη ιδιωτική υποδομή μας δόθηκαν τα κατάλληλα αναγνωριστικά (όνομα χρήστη και κωδικός χρήστη) από τον διαχειριστή της υποδομής Νέφους. Για την δημιουργία μιας εικονικής μηχανής ήταν απαραίτητο να ακολουθηθούν κάποια συγκεκριμένα βήματα, όπως αυτά περιγράφονται παρακάτω.

1. Δημιουργία Τοπολογίας Δικτύου

Αρχικά, η δημιουργία τοπολογίας δικτύου πραγματοποιήθηκε με τον ορισμό ενός εσωτερικού εικονικού δικτύου (virtual network) και ενός εικονικού δρομολογητή (virtual router) έτσι ώστε να υπάρχει επικοινωνία με εξωτερικά δίκτυα. Η επικοινωνία αυτή είναι βασισμένη σε κανόνες και πρωτόκολλα τα οποία εμείς ορίσαμε σύμφωνα με τις ανάγκες και τις απαιτήσεις των υπηρεσιών που αλληλεπιδρούν με το σύστημά μας.



2. Δημιουργία Στιγμιότυπου (Instance)

Με την δημιουργία στιγμιότυπου δεσμεύουμε μια Εικόνα (Image) μέσα στο οποίο περιλαμβάνεται η λογική της εφαρμογής μας. Την Εικόνα την παραμετροποιήσαμε σύμφωνα με τις ανάγκες μας και την υπολογιστική ισχύ που χρειαζόμασταν. Ειδικότερα, επιλέξαμε λειτουργικό σύστημα Ubuntu 12.04 LTS-64 bit, υπολογιστική ισχύ small με μνήμη RAM 2GB, σκληρό δίσκο 20GB (Σχήμα 13). Στη συνέχεια, για την πρόσβαση στο

διαδίκτυο δώσαμε εξωτερική και εσωτερική διεύθυνση (IP) ώστε να ολοκληρωθεί το στιγμιότυπο (Σχήμα 14).

<input type="checkbox"/>	Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
<input type="checkbox"/>	tsoultanopoulos	93.93.1.5 147.27.50.154	m1.small 2GB RAM 1 VCPU 20GB Disk	tsoultanopoulos_key	Active	None	Running	Create Snapshot More ▾
<input type="checkbox"/>	json_storage_vm	93.93.1.4 147.27.50.153	m1.small 2GB RAM 1 VCPU 20GB Disk	tsoultanopoulos_key	Active	None	Running	Create Snapshot More ▾

3. Εργαλεία ανάπτυξης εφαρμογής και Εικονική Μηχανή

Αφού δημιουργήσαμε την εικονική μηχανή, εγκαταστήσαμε τα εργαλεία που ήταν απαραίτητα για την υλοποίηση της λογικής του συστήματος. Για την εγκατάσταση και την πρόσβαση στην εικόνα χρησιμοποιήθηκε το εργαλείο Putty(ssh protocol) και FilleZila αντίστοιχα.

Αρχικά, χρησιμοποιήσαμε το λογισμικό Apache ώστε η εικονική μηχανή να έχει τη δυνατότητα προσφοράς υπηρεσιών στον διακομιστή (server). Επιπλέον, εγκαταστάθηκε η βιβλιοθήκη της PHP που μας ήταν χρήσιμη στην επικοινωνία με τον διακομιστή και περιείχε τα εργαλεία με τα οποία πραγματοποιούνται οι κλήσεις στις υπηρεσίες. Επιπλέον εγκαταστάθηκαν οι βιβλιοθήκες CURL, για την δημιουργία HTTP «αιτημάτων» (requests) στην PHP και η βιβλιοθήκη APC για την χρήση μνήμης cache στις περιπτώσεις που θέλουμε η υπηρεσία μας να ανταποκρίνεται πιο γρήγορα. Τέλος, στην εικονική μηχανή που είναι εγκατεστημένη η υπηρεσία μας κάναμε εισαγωγή του Php admin για πιο εύκολη διαχείριση της βάσης δεδομένων μας.

4.2.2 Παραμετροποίηση JSON STORAGE GE

Η υπηρεσία JSON Storage GE παρέχεται από την υποδομή Νέφους Intellicloud σε μορφή ανενεργούς εικόνας (snapshot). Η υπηρεσία είναι έτοιμη προς χρήση δημιουργώντας μια εικόνα (image) και δίνοντάς της μια εξωτερική διεύθυνση (floating IP 147.27.50.33:3000).

Στην περίπτωση του JSON Storage GE είναι αναγκαίο να γίνει μια αρχική παραμετροποίηση πριν ξεκινήσει η λειτουργία του συστήματος Intelligate, ώστε ο SE να είναι πλήρως λειτουργικός στην αποθήκευση των δεδομένων. Όπως αναφέρθηκε σε προηγούμενη υποενότητα, ο JSON Storage GE δίνει την δυνατότητα δημιουργίας χρηστών οι οποίοι μπορούν να διατηρούν τις δικές τους βάσεις δεδομένων, συλλογές και εγγραφές. Στον παρακάτω πίνακα παρουσιάζονται αναλυτικά οι βάσεις δεδομένων και οι αντίστοιχες συλλογές που χρειάστηκε να δημιουργηθούν με βάση το σχήμα του JSON storage (Ενότητα 3.4.1 Σχήμα 17).

Διεύθυνση	Αποτέλεσμα
http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/	Δημιουργία Collection USERS για την αποθήκευση χρηστών-ασθενών μαζί με την συσκευή τους.
http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/	Δημιουργία Collection Sensors Rules για την αποθήκευση κανόνων-αισθητήρων ανά χρήστη.

Σε κάθε κλήση προς την υπηρεσία χρησιμοποιήσαμε ως επικεφαλίδες (Header) “Content-type : application/json” και “Authorization : Basic {base64(username:password)}” που είναι μια κωδικοποίηση για την ταυτοποίηση των στοιχείων του χρήστη. Οι βασικοί μέθοδοι που (REST API)χρησιμοποιήθηκαν στις παραπάνω συλλογές είναι οι εξής

Μέθοδοι	Διευθύνσεις	Σώμα(JSON)	Αποτέλεσμα
POST	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/records	{ "name": "...", "RegID": "..." }	Εισαγωγή νέου χρήστη ασθενή με την συσκευή του.
PUT	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/records	{ "name": "...", "RegID": "..." }	Ανανέωση νέου χρήστη ασθενή με την συσκευή του.
POST	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/finds	{ "name": "...", } }	Εύρεση και ανάκτηση στοιχείων συγκεκριμένου ασθενή.
DELETE	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/{user_id}	-----	Διαγραφή χρήστη-συσκευής.
GET	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/USERS/records	-----	Ανάκτηση όλων των γραμμένων στο σύστημα ασθενών.

POST	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/records	{ "nameID": "user1@gmail.com", "Devices": [{ "Device_id": "Nonin3230_501860219", "DeviceFamily": "NoninNonix", "Measures": [{ "Measure_name": "SPO2", "MaxValue": 100, "MinValue": 90 }, { "Measure_name": "pulse_rate", "MaxValue": 100, "MinValue": 50 }] }] }	Εισαγωγή κανόνα στο σύστημα
PUT	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/records	>> >>	Ανανέωση κανόνων
POST	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/finds	{ "name": "...", }	Εύρεση κανόνων χρήστη
DELETE	http://147.27.50.33:3000/users/thodoris/dbs/intellicloud/collections/SensorRules/{SensorRuleid}		Διαγραφή κανόνων χρήστη

4.2.3 Παραμετροποίηση Google Cloud Messaging GE

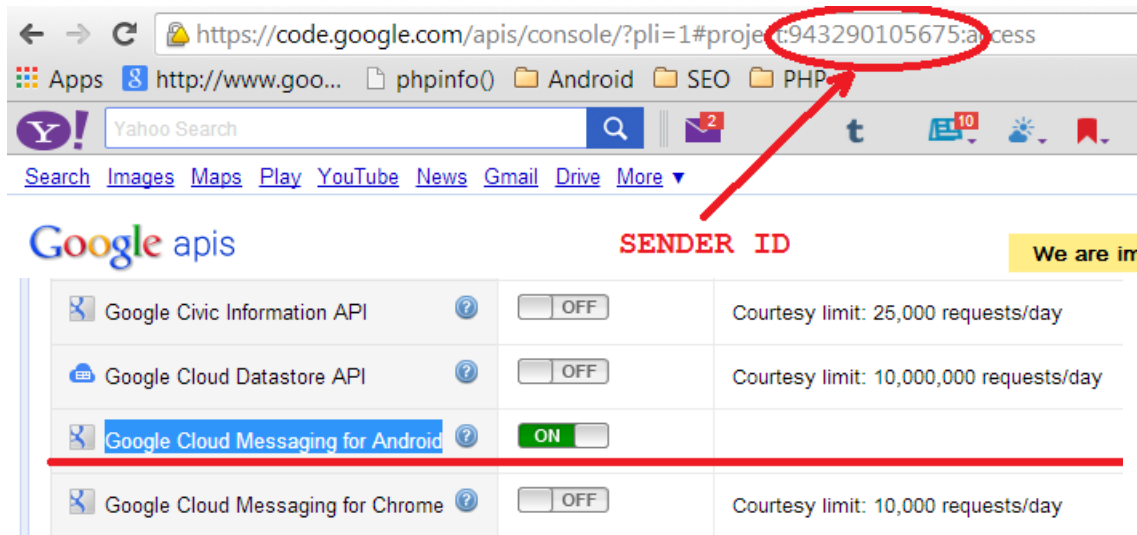
Όπως είπαμε το Google cloud messaging GE[19] είναι μια δωρεάν υπηρεσία που βοηθά τους προγραμματιστές να στέλνουν μηνύματα σε πολλές πλατφόρμες. Για την προσαρμογή του συγκεκριμένου GE στις δικές μας ανάγκες ακολουθήθηκαν τα παρακάτω βήματα.

Βήμα 1^ο Ενεργοποίηση του GCM από της πλατφόρμα PAAS της GOOGLE και λήψη κωδικού αποστολής(senderID) και κωδικού API(API key).

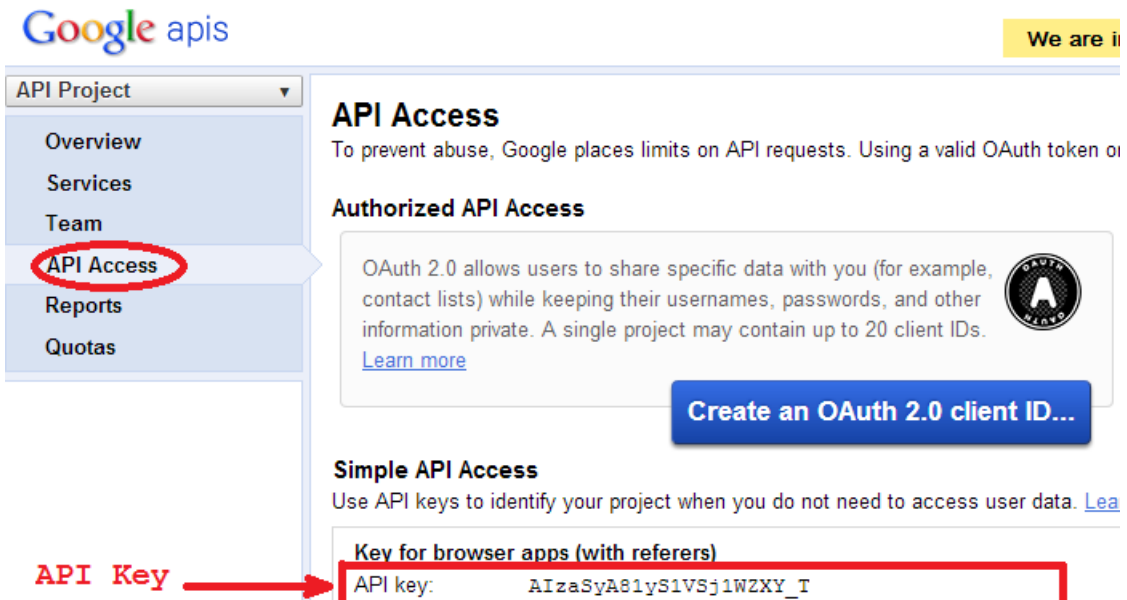
1. Ανοίγουμε την σελίδα <https://console.developers.google.com/project/intellegate-7dd89?pli=1> που ουσιαστικά είναι η Google APis Console page
2. Αν δεν έχουμε ένα API project , τότε δημιουργούμε ένα καινούργιο.



3. Έπειτα ενεργοποιούμε την υπηρεσία google cloud messaging , και λαμβανουμε το SenderID που μας προσφέρει μέσω του url .



4. Επιλέγουμε την ενότητα API access , και στο κάτω μέρος της συσκευής φαίνεται ο κωδικός API key όπου θα χρησιμοποιηθεί από τον διακομιστή μας για να στείλει μέσω του GCM μηνύματα σε κάθε διαφορετική συσκευή.



Βήμα 2^ο Δημιουργία κώδικα από μεριάς διακομιστή(server side code) για αποθήκευση του registration id της συσκευής , και την προώθηση μηνυμάτων στην συσκευή.

Σε αυτό το βήμα έχουμε υποθέσει ότι έχει γραφτεί ο κατάλληλος κώδικας(σύμφωνα με την προηγούμενη υπο ενότητα) από μέρους συσκευής. Το παραπάνω βήμα υλοποιείται με την δημιουργία ορισμένων php αρχείων που βοηθούν σε όλη αυτήν την λειτουργία.

1. InsertDeviceReg.php : Λαμβάνει ένα JSON String από την συσκευή και το εισάγει αυτούσιο στη συλλογή USERS του JSON STORAGE. Το Json string περιλαμβάνει το όνομα χρήστη καθώς και το αναγνωριστικό της συσκευής (Registration ID).

```
{
  "name": "user1@gmail.com",
  "reg_id": "APA91bE1U9LUy7J2mfwvCRUMPOosVpbN2kybRJ_Dk_Upz7eUu6G1gcSkMI87t
  QwZz8pZvTNNZCMgsMCfwrL7umcdF-Y5BNWJHtr_1ARF8rngYOKmC1yI-
  W0H1QAsyluhlTJ5PX-yaEmi"
}
```
2. GCM/index.php: Περιλαμβάνει ένα κατάλληλο text_area για να μπορέσει ο γιατρός να εισάγει το μήνυμά του, και σε αυτό το php αρχείο είναι επιπλέον αποθηκευμένο το API key. Το μήνυμά κείμενο του γιατρού «φεύγει» (tag “m” από το παρακάτω πίνακα) από τον διακομιστή προς την συσκευή μας βάση του Registration ID της συσκευής. (tag “to”

από το παρακάτω πίνακα). Έτσι με την παρακάτω κλήση επιτυγχάνεται η αποστολή ενός μηνύματος από το server μας στην συσκευή.

Μέθοδος	Διεύθυνση	Authorization	Σώμα(JSON)	Αποτέλεσμα
POST	https://android.googleapis.com/gcm/send	key=AlzaSyCxmcmPtZolt2gaZZW-p3g-y_Do8AXtw	{ "data": { "m": "Hello User I am your doctor" }, "to" : "APA91bGRsygK3GGgFgSIFnIFE4X8AgZcOg9aQdjLSyYRFamkpYfSGl4RoybseFPg2FuOKU4XcJgj21VEzUU EeqeWoF5VfNZE9BWwKqcv-9i1Os3U9R9uVmJQHz2DisK1SjwpczegBZRc" }	Αποστολή μηνύματος από Server Σε κινητή συσκευή μέσω GCM GE

4.2.4 Sensor History GE

Ο JSON storage GE αποτελεί μια ιδανική υπηρεσία γενικού σκοπού , για την αποθήκευση JSON strings. Όμως σε ορισμένες περιπτώσεις χρειαζόμαστε μια βάση δεδομένων η οποία θα μπορεί να υποστηρίξει διάφορων τύπων λογικών αιτημάτων όπως «Δώσε μου την τελευταία μέτρηση της τελευταίας ημέρας». Κάτι τέτοιο δεν είναι τόσο εύκολο να υλοποιηθεί στο JSON storage GE και για τον λόγο αυτόν δημιουργήσαμε μια καινούργια υπηρεσία γενικού σκοπού , η οποία έπειτα από ένα POST request από την συσκευή μετατρέπει την JSON πληροφορία σε κατάλληλη μορφή δομής δεδομένων και την εισάγει σε μια SQL βάση δεδομένων. Με τον τρόπο αυτό οι μετρήσεις που παράγουν οι αισθητήρες είναι αποθηκευμένες σε ένα table SensorsHistory που αποτελούν το ιστορικό κάθε αισθητήρα για κάθε ασθενή ξεχωριστά ανα διαφορετική ημέρα.

Τα πεδία του table SensorsHistory είναι τα εξής

- **ID (int):** To Primary key, Auto increment κάθε γραμμής-εγγραφής.

- **Email (text):** Το αναγνωριστικό ID κάθε χρήστη-ασθενούς.
- **SensorID (text):** Το αναγνωριστικό ID κάθε συσκευής.
- **Measures(long text):** Ουσιαστικά το JSON string μετρήσεων που παράγουν οι αισθητήρες(π.χ)

```
[{"SensorMeasure":"pulse_Rate","MeasureValue":96,"MeasureUnit":"bpm","Condition":"OK"}, {"SensorMeasure":"SPO2","MeasureValue":98,"MeasureUnit":"%","Condition":"OK"}]
```
- **STATE(text):** Ο τρόπος που μεταδόθηκε η πληροφορία (ON_TIME_INTERVAL,ON_DEMAND,VIOLATION_RULE)
- **Date(Date) :**Ημερομηνία σε format Χρόνος-Μήνας-Ημέρα
- **Start_time (time):** Χρόνος εκκίνησης εγγραφής μέτρησης σε format ωρα: λεπτά: δευτερόλεπτα.
- **End_time (time):** Χρόνος λήξης εγγραφής μέτρησης σε format ωρα: λεπτά: δευτερόλεπτα.

4.2.5 Connectivity Service

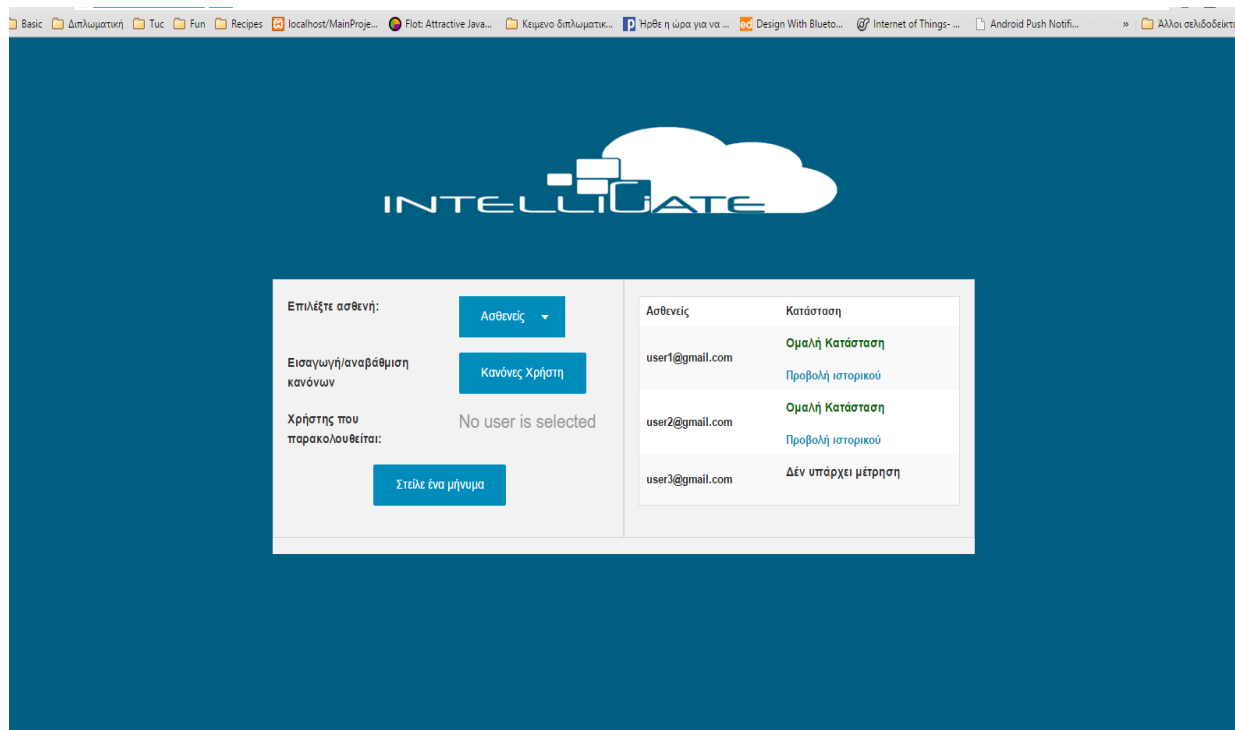
Ίδια ακριβώς λειτουργία με το Connectivity Service της ενότητας Intelligate. Υπάρχει ένα συγκεκριμένο κομμάτι μετατροπής δεδομένων σε JSON(και το αντίστροφο) , καθώς και προώθησης πληροφορίας από το νέφος προς την ενότητα Intelligate.

4.2.6 Application Logic (cloud) και μηχανισμοί.

Όπως έχουμε αναφέρει, το τμήμα της λογικής του συστήματος κινεί τα νήματα της εφαρμογής μας. Μέχρι τώρα έχει γίνει η περιγραφή λειτουργίας του σε κάθε ξεχωριστή υπηρεσία γενικού σκοπού. Σε αυτό το σημείο θα παρουσιάσουμε τα τεχνικά χαρακτηριστικά, την λειτουργία και τους μηχανισμούς. Η λογική του συστήματος υλοποιήθηκε σε γλώσσα PHP και με τη χρήση της βιβλιοθήκης Curl έτσι ώστε να εκτελούνται οι κλήσεις API. Στην ενότητα του Application Logic θα συμπεριλάβουμε και την υλοποίηση του Γραφικού περιβάλλοντος που υλοποιήθηκε με την βοήθεια του framework zurb-foundation²⁷. Παρακάτω θα αναφέρουμε τους μηχανισμούς υποστήριξης που

²⁷ <http://foundation.zurb.com/>

προσφέρονται από το web-app Intelligate η οποία είναι προσβάσιμη στο link <http://147.27.50.154/Intelligate/> .



Σχήμα 33-Αρχική σελίδα Intelligate




Στο σχήμα 31 φαίνεται η αρχική σελίδα Intelligate web application. Το συγκεκριμένο Interface επιτρέπει την αλληλεπίδραση της Web εφαρμογής με τον γιατρό. Όπως φαίνεται μέσω εργαλείων JavaScript (με βιβλιοθήκη query). Ο χρήστης-γιατρός μπορεί , να επιλέξει έναν συγκεκριμένο ασθενή για να τον παρακολουθεί σε πραγματικό χρόνο, να ορίσει κανόνες για τον συγκεκριμένο ασθενή , να παρακολουθήσει σε πραγματικό χρόνο την κατάσταση κάθε ασθενούς(δεξιά πίνακας), καθώς και να στείλει ένα μήνυμα. Τέλος μπορεί να δει το ιστορικό οποιουδήποτε από τους εγγεγραμμένους χρήστες ανα πάσα χρονική στιγμή.

Μηχανισμός Ορισμού κανόνων για κάθε χρήστη ασθενή


Προσθήκη Ανανέωση Κανόνων





Χρήστης: user1@gmail.com

Επιβεβαίωση

User   JSON  Properties


Email
user1@gmail.com





Devices 




Device 1   Device  JSON  Properties

DEVICE ID
Nonin3230_501860219

DEVICE FAMILY
NoninNonin

Measures 

Measure_name	MaxValue	MinValue	
SPO2	100	90	 
pulse_rate	100	50	 

 New  Load new  All

Σχήμα 33-Ορισμός κανόνων για κάθε χρήστη

Ο χρήστης γιατρός μπορεί να ορίσει με δυναμικό τρόπο , τους κανόνες ενός ασθενούς συμπληρώνοντας απλά μια φόρμα η οποία περιλαμβάνει το όνομα του χρήστη(email) και τους αισθητήρες που χαρακτηρίζονται από το όνομα τους , την οικογένεια τους καθώς και τις μετρήσεις τους μαζί με τα κατάλληλα κατώφλια. Η φόρμα αυτή είναι δυναμική που σημαίνει ότι μπορούμε να εισάγουμε όσες συσκευές θέλουμε για κάθε χρήστη , κάνοντας έτσι πιο αποδοτική την αντιστοίχη κανόνων-συσκευών με ασθενείς. Όταν ο γιατρός ορίσει τους κανόνες για έναν συγκεκριμένο χρήστη τότε πατάει επιβεβαίωση και τα δεδομένα αποστέλλονται στον JSON storage όπου και αποθηκεύονται μέχρι να ανακτηθούν από την κινητή συσκευή που τρέχει την εφαρμογή Intelligate.

Ο συγκεκριμένος μηχανισμός προσφέρει μια δυναμική φόρμα συμπλήρωσης η οποία δημιουργήθηκε με την βοήθεια μια βιβλιοθήκης JavaScript η οποία ονομάζεται json_editor²⁸. Με την βοήθεια της συγκεκριμένης βιβλιοθήκης ορίζουμε ένα Json schema το οποίο ουσιαστικά επιτρέπει την δυναμική προσθαφαίρεση στοιχείων με δυναμικό τρόπο. Το json schema αυτό φαίνεται παρακάτω.

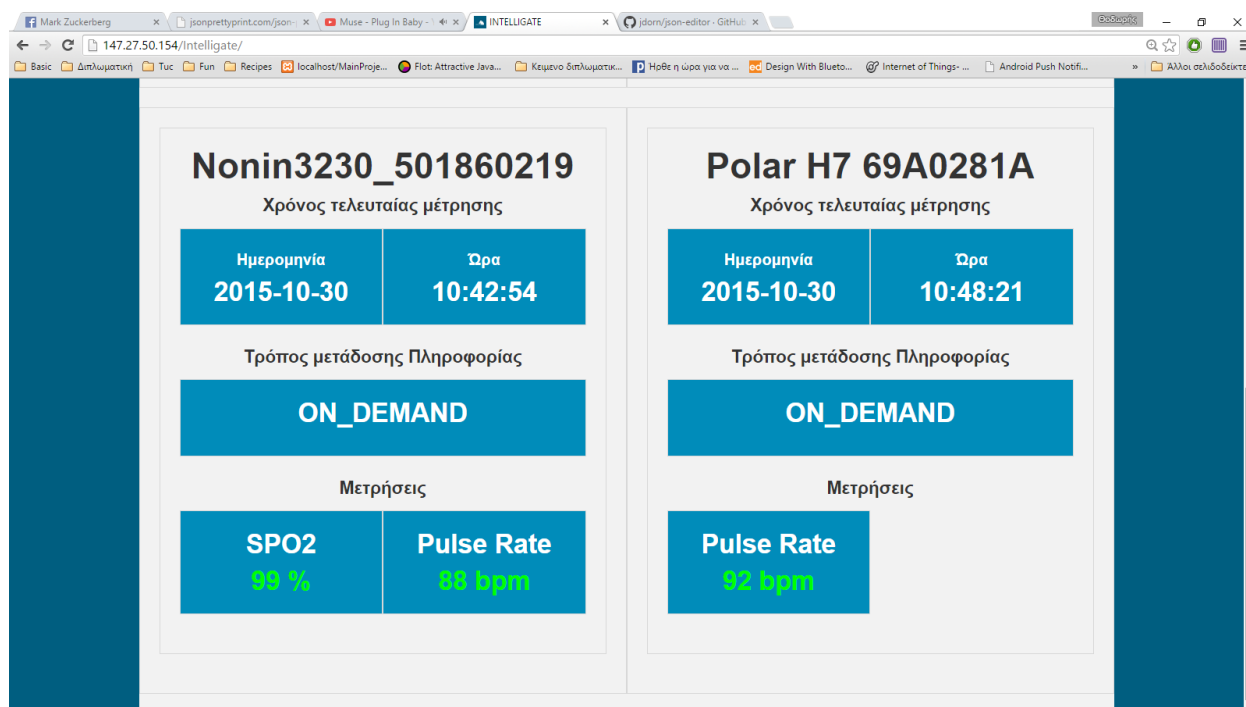
²⁸ <https://github.com/jdorn/json-editor>

```

{
  "type": "object",
  "title": "User",
  "properties": {
    "nameID": {
      "type": "string",
      "title": "Email",
      "format": "email",
      "required": "true"
    },
    "Devices": {
      "type": "array",
      "title": "Devices",
      "items": {
        "type": "object",
        "title": "Device",
        "properties": {
          "Device_id": {
            "type": "string",
            "title": "DEVICE ID"
          },
          "DeviceFamily": {
            "type": "string",
            "title": "DEVICE FAMILY"
          },
          "Measures": {
            "type": "array",
            "title": "Measures",
            "format": "table",
            "items": {
              "type": "object",
              "properties": {
                "Measure_name": {
                  "type": "string"
                },
                "MaxValue": {
                  "type": "integer"
                },
                "MinValue": {
                  "type": "integer"
                }
              }
            }
          }
        }
      }
    }
  }
}

```

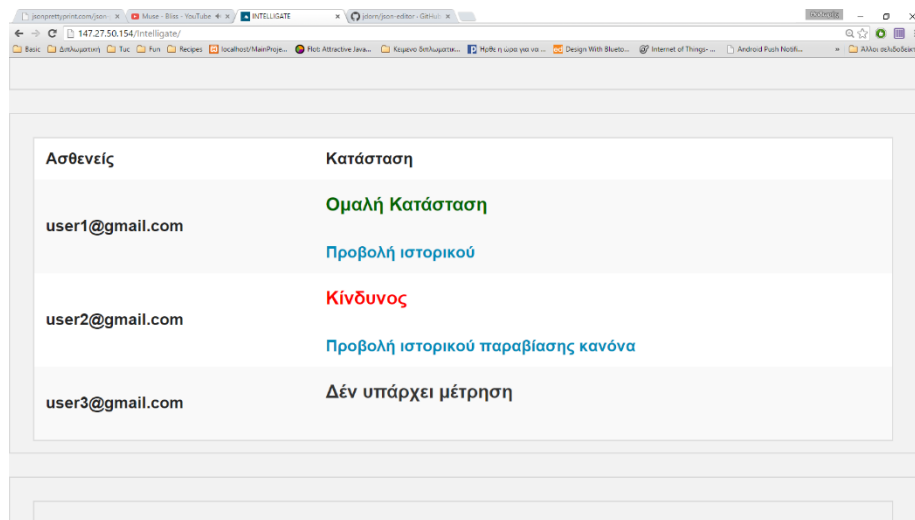
Μηχανισμός Παρακολούθησης τιμών ασθενούς



Σχήμα 34-Patient Monitoring

Ο χρήστης γιατρός αφού επιλέξει έναν συγκεκριμένο χρήστη ασθενή τότε έχει την δυνατότητα να παρακολουθεί τις μετρήσεις ζωτικής σημασίας του χρήστη ασθενή σε πραγματικό χρόνο. Με βοήθεια ασύγχρονων αιτημάτων τύπου Ajax-Jquery ανανεώνουμε κάθε φορά το συγκεκριμένο κομμάτι, λαμβάνοντας πάντα την τελευταία μέτρηση για κάθε χρήστη-αισθητήρα από την βάση δεδομένων που προσφέρει ο SensorHistoryGE δίνοντας την αίσθηση ότι όλα αυτά τα γεγονότα γίνονται σε πραγματικό χρόνο. Μέσω αυτού του μηχανισμού ο χρήστης γιατρός βλέπει το όνομα της συσκευής, τον χρόνο της τελευταίας μέτρησης(ώρα και Ημερομηνία), τον τρόπο μετάδοσης της πληροφορίας καθώς και τις Μετρήσεις που παράγει ο κάθε διαφορετικός αισθητήρας. Ο Μηχανισμός αυτός στηρίζεται στην ποικιλία των διαφορετικών αιτημάτων ανα δευτερόλεπτο ως προς την βάση δεδομένων που κρατάει το ιστορικό μετρήσεων των αισθητήρων και είναι όλοι φανεροί στο αρχείο db_helper.php.

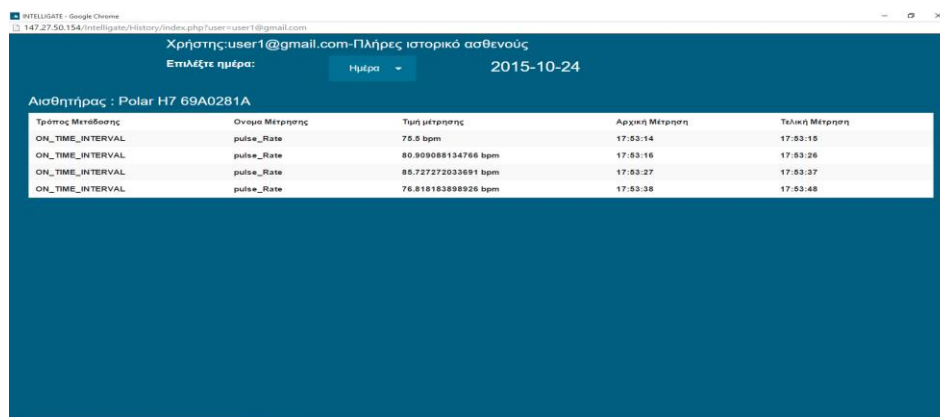
Μηχανισμός Ορισμού παρακολούθησης κατάστασης συνόλου ασθενών και του ιστορικού τους.



Ασθενείς	Κατάσταση
user1@gmail.com	Ομαλή Κατάσταση Προβολή ιστορικού
user2@gmail.com	Κίνδυνος Προβολή ιστορικού παραβίασης κανόνα
user3@gmail.com	Δέν υπάρχει μέτρηση

Σχήμα 35-Patients Condition

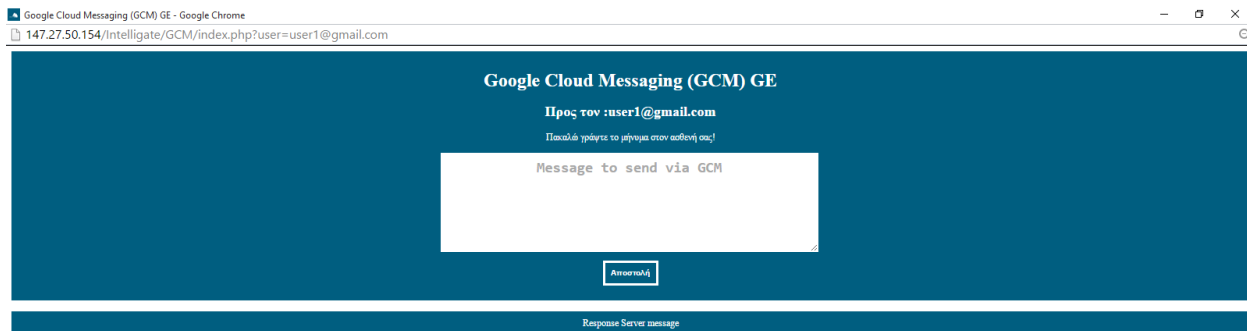
Ο χρήστης γιατρός ανεξάρτητα από ποιόν ασθενή έχει επιλέξει να παρακολουθεί έχει την δυνατότητα να βλέπει την κατάσταση του συνόλου των ασθενών ανα πάσα χρονική στιγμή προσφέροντας έτσι στο σύστημα την αμεσότητα της επέμβασής του. Και σε αυτό το σημείο μέσω του SensorsHistory GE γίνεται μια κλήση αιτήματος που επιστρέφει όλους τους εγγεγραμμένους ασθενείς μαζί με τις καταστάσεις τους σε μορφή πίνακα. Ο πίνακας αυτός ανανεώνεται κάθε δευτερόλεπτο με την χρήση ασύγχρονων αιτημάτων(AJAX) για κάθε εγγεγραμμένο ασθενή. Εκτός από την κατάσταση του ασθενή το συγκεκριμένο κομμάτι δίνει την δυνατότητα στον γιατρό να επιλέξει την προβολή του συνολικού ιστορικού του χρήστη για όλη την διάρκεια των μετρήσεων του όπως φαίνεται στο παρακάτω σχήμα.



Χρήστης:user1@gmail.com-Πλήρες ιστορικό ασθενούς				
Επιλέξτε ημέρα:		Ημέρα	2015-10-24	
Αισθητήρας : Polar H7 69A0281A				
Τρόπος Μετάδοσης	Όνομα Μέτρησης	Τιμή μέτρησης	Αρχική Μέτρηση	Τελική Μέτρηση
OH_TIME_INTERVAL	pulse_Rate	75.9 bpm	17:53:14	17:53:15
OH_TIME_INTERVAL	pulse_Rate	80.909088134766 bpm	17:53:16	17:53:26
OH_TIME_INTERVAL	pulse_Rate	85.727272033691 bpm	17:53:27	17:53:37
OH_TIME_INTERVAL	pulse_Rate	76.81818388928 bpm	17:53:38	17:53:48

Σχήμα 36-Patient History

Μηχανισμός αποστολής μηνύματος



Σχήμα 37-Sending a message

Ο ιατρός μπορεί άμα κρίνει ότι μια κατάσταση για έναν ασθενή μπορεί να θεωρείται κρίσιμη να στείλει ένα μήνυμα προς την συσκευή του συγκεκριμένου χρήστη με την βοήθεια του Google Cloud messaging GE ο οποίος θα μεριμνήσει να στείλει (ακόμα και σε περίπτωση απουσίας δικτύου) μήνυμα στην συσκευή με την μορφή ενημέρωσης. Το web application προσφέρει ένα κατάλληλο text area το οποίο η λειτουργία του έχει αναλυθεί στην υποενότητα 4.3.3 .Ο χρήστης γιατρός πληκτρολογώντας το κείμενο μήνυμα πατάει αποστολή και περιμένει μέχρι του επιστραφεί η κατάλληλη αναφορά αποστολής μηνύματος.

5 Συμπεράσματα και μελλοντικές επεκτάσεις

5.1 Συμπεράσματα

Ο αρχικός στόχος της εφαρμογής Intelligate ήταν να εκμεταλλευτεί την ιδέα του διαδικτύου των πραγμάτων όπου με την χρήση κατάλληλων φορητών αισθητήρων και την τεχνολογία που μας προσφέρουν υποδομές του υπολογιστικού νέφους να επιτευχθεί η απομακρυσμένη παρακολούθηση ασθενών από τους γιατρούς τους οι οποίοι πάσχουν από καρδιακές παθήσεις από όποιο σημείο και αν βρίσκονται με την χρήση απλά ενός φυλλομετρητή(από μεριάς γιατρού) και μιάς έξυπνης κινητής συσκευής(smartphone) από μεριά χρήστη ασθενούς. Ο δυναμικό τρόπους εισαγωγής νέων αισθητήρων δίνει την δυνατότητα στην εφαρμογή να μην υπάρχουν περιορισμοί στην προσθαφέραιση νέων αισθητήρων που παράγουν δεδομένα ζωτικής σημασίας. Επιπλέον ένα ακόμη χαρακτηριστικό το οποίο καθιστά την διπλωματική μας εργασία διαφορετική από άλλα συστήματα ίδιου τύπου είναι ο τρόπος αποστολής των δεδομένων στο νέφος καθώς αλλάζει ανάλογα με την περιστάσεις. Η διαδικτυακή μας εφαρμογή «τρέχει» σε ένα εικονικό μηχανήμα το οποίο παρέχεται μέσω της υποδομής Intelligate ή οποία μπορεί να μας προσφέρει επιπλέον υπολογιστικούς πόρους αν το σύστημα μας γίνει πιο πολύπλοκο. Η λειτουργία υπηρεσιών γενικού σκοπού(Generic Enablers) μας βοήθησε στην απλούστευση της υλοποίησης του συστήματος καθώς πολλές υπηρεσίες τις οποίες χρειαζόμασταν ήταν ήδη έτοιμες , και μέσω μιας απλής παραμετροποίησης μπορούσαν να εξυπηρετήσουν δικές μας ανάγκες.

Συγκεκριμένα σε επίπεδο της εφαρμογής Intelligate οι βασικές λειτουργίες που υποστηρίζονται είναι οι εξής.

- Δυναμική προσθήκη BLE φορητών αισθητήρων μέσω ενός XML σχήματος
- Εγγραφή ενός νέου χρήστη στο σύστημα.
- Δυνατότητα λήψης κανόνων-αισθητήρων που αφορούν ένα χρήστη ασθενή από το νέφος.
- Δυνατότητα παρακολούθησης των ζωτικών μετρήσεων που παράγουν οι BLE αισθητήρες από μεριά του χρήστη ασθενή.
- Έξυπνος τρόπος μετάδοσης δεδομένων προς το νέφος με 3 διαφορετικούς τρόπους(On time interval-on Demand-Violation Rule).

- Τοπική αποθήκευση δεδομένων ανα τακτά χρονικά διαστήματα , έπειτα από κατάλληλη κρυπτογράφηση, σε περίπτωση αδυναμίας μετάδοσης δεδομένων προς το νέφος σε περίπτωση απουσίας δικτύου.
- Ειδοποίηση και προβολή μηνύματος Γιατρού προς τον ασθενή μέσω κατάλληλου γραφικού περιβάλλοντος.

Σε επίπεδο της web εφαρμογής Intelligate (web application μαζί με διεπαφή συστήματος νέφους(back end cloud))

- Εισαγωγή/ανανέωση κανόνων-αισθητήρων που αφορούν έναν χρήστη.
- Παρακολούθηση ζωτικών δεδομένων που παράγουν οι αισθητήρες για κάθε χρήστη σε πραγματικό χρόνο.
- Παρακολούθηση της κατάστασης(Ομαλή ή κίνδυνος) όλων των ασθενών ανεξάρτητα από ποιόν ασθενή παρακολουθεί ο γιατρός σε πραγματικό χρόνο.
- Ιστορικό μετρήσεων των δεδομένων των αισθητήρων για κάθε χρήστη ασθενή ξεχωριστά.
- Δυνατότητα του γιατρού να αποστέλλει μήνυμα προς τον ασθενή οποιαδήποτε χρονική στιγμή.

Κυριότερα πλεονεκτήματα του Συστήματος Intelligate

- **Δια λειτουργικότητα :** Καθώς μέσω του ορισμού XML σχήματος αισθητήρων και της υλοποίησης ολόκληρου του συστήματος , επιτυγχάνεται η δυναμική πρόσθεση BLE αισθητήρων που υπακούν στο πρωτόκολλο επικοινωνίας Bluetooth Low energy.”
- **Επεκτασιμότητα:** Το σύστημά μας υπακούει στις βασικές αρχές τις Υπηρεσιοκεντρικής αρχιτεκτονικής, με αυτόν τον τρόπο οποιαδήποτε νέα λειτουργία θέλουμε να ενσωματώσουμε στο σύστημα μας , μπορεί εύκολα να ενσωματωθεί ως κομμάτι μιας ήδη υπάρχουσας υπηρεσίας ή ως ένα νέο ξεχωριστό κομμάτι.
- **Ελαστικότητα:** προσαρμογή των υπολογιστικών πόρων της υποδομής νέφους σε περίπτωση απαίτησης από την εφαρμογή.

- **Οικονομία σε διαδικτυακό φόρτο εργασίας:** Η εφαρμογή Intelligate μετατρέπει την κινητή συσκευή από μια απλή πύλη διαδικτύου σε μια έξυπνη πύλη διαδικτύου ή οποία στέλνει δεδομένα στο νέφος ανάλογα με την κρισιμότητα της κατάστασης.
- **Εύκολο στην χρήση:** Δεδομένης της ανάλυσης ροής δεδομένων και ενεργειών στο σύστημά μας (3.5) και την προσφορά ενός απλού γραφικού περιβάλλοντος σε επίπεδο κινητής πλατφόρμας και νέφους το σύστημα Intelligate είναι εύκολο ως προς την κατανόηση λειτουργίας και χρήσης του.

5.2 Μελλοντικές επεκτάσεις

Το σύστημα που αναπτύξαμε έχει την δυνατότητα με συγκεκριμένες επεκτάσεις, οι οποίες επισημαίνονται συνοπτικά παρακάτω να γίνει περισσότερο λειτουργικό και αποδοτικό ξεπερνώντας τους περιορισμούς της εφαρμογής μας.

- **Προσθήκη περισσότερων αισθητήρων τύπου BLE :** Μέσω της δυναμικής προσθήκη BLE αισθητήρων ως μελλοντική επέκταση είναι η εισαγωγή νέων σχημάτων αισθητήρων προσφέροντας νέες δυνατότητες στο σύστημά μας.
- **Προσθήκη αισθητήρων διαφορετικού τύπου πρωτοκόλλου:** Στην αγορά εργασίας υπάρχουν και αισθητήρες οι οποίες δεν έχουν την δυνατότητα άμεσης σύνδεσης τους στο διαδίκτυο όπως αισθητήρες με πρωτόκολλο επικοινωνίας Zig-Bee²⁹, Z-wave³⁰ όπου θα μπορούν να υποστηριχτούν στην εφαρμογή Intelligate.
- **Σύνδεση πολλών αισθητήρων ταυτόχρονα στην συσκευή:** Ένας περιορισμός της εφαρμογής μας ήταν ότι η κεντρική συσκευή(tablet) μπορούσε να υποστηρίξει και να συνδεθεί μόνο με μια περιφερειακή συσκευή(αισθητήρας).Ως μελλοντική επέκταση της εφαρμογής θα ήταν η ταυτόχρονη υποστήριξη πληθώρας περιφερειακών συσκευών από την κεντρική συσκευή με την συγχρονισμένη ή και ασύγχρονη μεταφορά των δεδομένων τους προς το νέφος.

²⁹ <http://www.zigbee.org/>

³⁰ <http://www.z-wave.com/>

- **Υποστήριξη σε περισσότερα λειτουργικά συστήματα κινητών συσκευών:** Επιπλέον ένα περιορισμός της εφαρμογής μας ήταν ότι λειτουργεί μόνο σε λογισμικό Android.Ως μελλοντική επέκταση της εφαρμογής μπορεί να υλοποιηθεί και σε κινητά διαφορετικό λογισμικού όπως IOS³¹,Windows Mobile³²,Blackberry³³ κτλ.
- **Υποστήριξη της αναγνώρισης θέση του ασθενούς μέσω GPS³⁴ της κινητής συσκευής:**Ο Γιατρός και μόνος αυτός , να έχει το δικαίωμα να βλέπει την θέση του ασθενή του σε περίπτωση όπου προκύψει κάποιο πρόβλημα.
- **Ολοκλήρωση διεπαφής συστήματος σε επίπεδο νέφους:** Όπως αναφέραμε και στο κυρίως μέρος της διπλωματικής μας εργασίας , η διεπαφή συστήματος σε επίπεδο νέφους μαζί με την web εφαρμογή αποτελεί κομμάτι ορθής λειτουργίας της εφαρμογής Intelligate σε επίπεδο συσκευής. Ως μελλοντική επέκταση είναι η ολοκλήρωση διεπαφής του συστήματος σε λειτουργίες που να καλύπτουν πιο αποδοτικά έναν οργανισμό υγειονομικής περίθαλψης(νοσοκομεία-νοσηλευτικά κέντρα).
- **Ασφάλεια στην ροή πληροφορίας των δεδομένων:** Παραπάνω λειτουργίες του συστήματος που αναφέρθηκαν, έχουν πραγματοποιηθεί αλλά η μεγαλύτερη πρόκληση στην τεχνολογίας νέφους , είναι η ασφάλεια των δεδομένων κατά την μεταφορά τους από την συσκευή στο νέφος(και αντίστροφα). Οι μετρήσεις ζωτικών δεδομένων θεωρούνται ευαίσθητες και δεν πρέπει να είναι προ βάσιμες από τον οποιοδήποτε. Πιο συγκεκριμένα το σύστημα μας αν δεν προσφέρει την ασφαλή μεταφορά των ζωτικών δεδομένων τότε ο ίδιος ο χρήστης μπορεί να πέσει θύμα διαφημιστικών εταιριών που θα προσφέρουν μια «καλύτερη στήριξη» στο πρόβλημα υγείας του. Σύμφωνα με τα παραπάνω θα πρέπει ως μελλοντική επέκταση να υλοποιηθεί η ασφαλής ανταλλαγή πληροφορίας μεταξύ νέφους και συσκευής.

³¹ <https://el.wikipedia.org/wiki/IOS>

³² <https://www.microsoft.com/el-gr/windows/phones>

³³ <http://us.blackberry.com/>

³⁴ https://el.wikipedia.org/wiki/Global_Positioning_System

Bibliography

- [1] Anon, 1999:Jan. *A taste of the future: the electronic tongue*, s.l.: CNN Interactive .
- [2] Wilson, C. B., 1999. [Online]
Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1129066/>
- [3] Brown, E., October 25, 2011. *NIST*. [Online]
Available at: <http://www.nist.gov/itl/csd/cloud-102511.cfm>
- [4] Armbrust, M., April 2010. *A View of Cloud Computing*. [Online]
Available at: <http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext>
- [5] Appcore, n.d. *Appcore*. [Ηλεκτρονικό]
Available at: <http://www.appcore.com/3-types-cloud-service-models/>
- [6] Appcore, n.d. *Appcore*. [Ηλεκτρονικό]
Available at: <http://www.appcore.com/types-cloud-computing-private-public-hybrid-clouds/>
- [7] Omar SEFRAOUI, Mohammed AISSAOUI & Mohsine ELEULDJ, n.d. *OpenStack: Toward an Open-Source Solution for*, s.l.: s.n.
- [8] G Panagopoulou
http://www.dpa.gr/pls/portal/docs/PAGE/APDPX/EUROPEAN_DP_DAY_GENERAL/2014_DP_DAY/OMILIES_2014_DP_DAY/PANAGOPOULOU_DIADIKTIO%20TON%20PRAGMATON.PDF

- [9] Wikipedia Service Oriented architecture
Available at: https://en.wikipedia.org/wiki/Service-oriented_architecture

- [10] Hypertext Trasfer Protocol Available at
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [11] Representational state tranfer
Available at: https://en.wikipedia.org/wiki/Representational_state_transfer
- [12] Posted at January 9, 2008 by Nicholas C. Zakas Is Json better Than XML?
Available at: <https://www.nczonline.net/blog/2008/01/09/is-json-better-than-xml/>
- [13] GAAT NetWork Topology Available at: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- [14] GAAT Available
at:<https://developer.bluetooth.org/TechnologyOverview/Pages/GATT.aspx>
- [15] Android history available at: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [16] http://www.polar.com/e_manuals/H7_Heart_Rate_Sensor/Polar_H7_Heart_Rate_Sensor_accessory_manual_English.pdf
- [17] <http://www.oxygenium.gr/media/1465/9560%20Brochure.pdf>
- [18] Interact Gesture Recognition to the Cloud Alexandros Preventis,Kostas Stravoskoufos,Stelios sotiriadis , Euripides Petrakis availble
at:<http://www.intelligence.tuc.gr/~petrakis/publications/ucc.pdf>
- [19] Google Cloud messaging available at : <https://developers.google.com/cloud-messaging/gcm>
- [20] <http://catalogue.fiware.org/enablers/identity-management-keyrock>
- [21] <http://developer.android.com/reference/javax/xml/parsers/DocumentBuilder.html>
- [22] Adnvanced Encryption available at :
- [23] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

