Πολυτεχνείο Κρήτης

Διπλωματική Εργασία

Παρακολούθηση Ατμοσφαιρικών Συνθηκών με Χρήση Αισθητήρων στο Υπολογιστικό Νέφος

Cloud Intelligent Monitoring Atmosphere Technology

Λάμπρος Μαστοράκης

Εργαστήριο Ευφυών Συστημάτων
–
Τμήμα Ηλεκτρονικών Μηχανικών
και Μηχανικών Υπολογιστών

Επιτροπή Διπλωματικής Εργασίας

Καθηγητής Ευριπίδης Πετράκης (επιβλέπων)
Καθηγητής Κώστας Καλαϊτζάκης
Επιστημονικός Συνεργάτης Στέλιος Σωτηριάδης

Οκτώβριος 2013

Περίληψη

Το CLIMATE (Cloud Intelligent Monitoring Atmosphere Technology) αποτελεί ένα διαδικτυακό σύστημα παρακολούθησης ατμοσφαιρικών συνθηκών με χρήση αισθητήρων στο «Νέφος» (Cloud). Μελετάει φυσικά μεγέθη της ατμόσφαιρας του περιβάλλοντος χώρου και λαμβάνει μετρήσεις για τις διακυμάνσεις τους. Σκοπός του συστήματος είναι να επεξεργάζεται τα δεδομένα των μετρήσεων προκειμένου να τα προσφέρει στους χρήστες ως χρήσιμη πληροφορία. Η λήψη και η επεξεργασία των δεδομένων, η παραμετροποίηση του συστήματος καθώς και η πιστοποίηση των χρηστών γίνεται αποκλειστικά συνδυάζοντας παρεχόμενες υπηρεσίες γενικού σκοπού και υπολογιστικούς πόρους που προσπελαύνονται διαδικτυακά και παρέχονται από υποδομές Νέφους. Το σύστημα μπορεί να εξυπηρετήσει ανάγκες χρηστών και οργανισμών προσφέροντας τη δυνατότητα, μέσω μιας διαδικτυακής διεπαφής, να παρακολουθούν τις διακυμάνσεις των φυσικών μεγεθών της ατμόσφαιρας σε συγκεκριμένους χώρους, να ενημερώνονται για τις μεταβολές των τιμών αυτών σε πραγματικό χρόνο και να έχουν πρόσβαση στο ιστορικό των μεταβαλλόμενων τιμών.

Abstract

CLIMATE (Cloud Intelligent Monitoring Atmosphere Technology) is a Cloud based system for monitoring atmosphere conditions using environmental sensors. The application studies physical atmosphere attributes and takes measurements of their variation. The purpose of this system is to edit the data retrieved in order to offer them as useful information to users. The retrieval and the editing of data, the system configuration as also the users' authorization are implemented combining Generic Enablers, services and resources provided through internet by Cloud infrastructures. CLIMATE services the needs of users and organizations to monitor the environmental conditions in specific places, to be informed about the changes of these conditions instantly and to have full access to the measurements' record offering a web application with a graphical user interface.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Πετράκη για τη σημαντική καθοδήγηση που μου προσέφερε καθ' όλη τη διάρκεια ενασχόλησής μου με τη διπλωματική εργασία. Επίσης όλους τους συναδέλφους από το εργαστήριο, εργαζομένους, προπτυχιακούς και μεταπτυχιακούς, και κυρίως τον Κώστα, τον Αλέξανδρο, το Στέλιο και το Σπύρο για τη καθοριστική βοήθειά τους. Τέλος την οικογένειά μου για την εμπιστοσύνη που μου δείχνει και φυσικά την Ιωάννα-Νεφέλη που συνεχίζει να με εμπνέει.

Η διπλωματική εργασία είναι αφιερωμένη στους γονείς μου που στήριξαν και συνεχίζουν να στηρίζουν τις επιλογές μου.

Περιεχόμενα

1. Εισαγωγή

1.1 Εφαρμογές με αισθητήρες και η ιδέα του Διαδικτύου των Πραγμάτων (IoT)

1.2 Βασικά χαρακτηριστικά των αισθητήρων και των αισθητήρων παρακολούθησης του περιβάλλοντος (Environmental Monitoring Sensors)

1.3 Αισθητήρες και Υπολογιστικό Νέφος (Cloud Computing)

1.3.1 Βασικά μειονεκτήματα προγενέστερων υποδομών και τεχνολογιών του Υπολογιστικού Νέφους

1.3.2 Τα άμεσα πλεονεκτήματα της τεχνολογίας του Νέφους

1.4 Περιγραφή του προβλήματος

1.5 Στόχοι της δουλειάς μας και περιγραφή της λύσης

1.5.1 Το μοντέλο των αισθητήρων

1.5.2 Οι υπηρεσίες που θα χρησιμοποιήσουμε

1.5.3 Η δομή της εργασίας

2. Υπάρχουσα τεχνολογία

2.1 Ο ορισμός του Υπολογιστικού Νέφους

2.2 Η ιστορία ανάπτυξης του Υπολογιστικού Νέφους

2.3 Πέντε ιστορικοί σταθμοί για το Υπολογιστικό Νέφος

2.4 Πως ωφελείται άμεσα ο χρήστης-καταναλωτής του Υπολογιστικού Νέφους;

2.5.1 Μοντέλα παροχής υπηρεσιών Νέφους

2.5.2 Υλοποιήσεις υποδομών Νέφους

2.6 Βασικά λειτουργικά χαρακτηριστικά του Υπολογιστικού Νέφους

2.7 Εικονικοποίηση Πόρων (Virtualization) και η σχέση του με το Υπολογιστικό Νέφος

2.7.1 Η τεχνολογία της Εικονικοποίησης Πόρων

2.7.2 Η τεχνολογία του Επιμελητή Εικονικών Μηχανών (Virtual Machine Monitor – Hypervisor)

2.8 Γιατί να επιλέξει κάποιος το Υπολογιστικό Νέφος;

2.9 Υπηρεσιο-κεντρική Αρχιτεκτονική (Service Oriented Architecture - SOA) και Υπολογιστικό Νέφος

2.9.1 Η λογική της Υπηρεσιο-κεντρικής Αρχιτεκτονικής

2.9.2 Η σχέση της Υπηρεσιο-κεντρικής Αρχιτεκτονικής με το Υπολογιστικό Νέφος

2.9.3 Τα βασικά πλεονεκτήματα της Υπηρεσιο-κεντρικής Αρχιτεκτονικής

2.10 REST-API (Representational State Transfer - Application Programming Interface)

2.10.1 Η αρχιτεκτονική του REST

2.10.2 Συνοπτικά ο τρόπος λειτουργίας του REST-API και τπυ πρωτοκόλλου HTTP

2.11 Προσφερόμενες υπηρεσίες Γενικού Σκοπού(Generic Enablers) και Ειδικού Σκοπού(Specific Enablers)

2.12 Το περιβάλλον του FI-WARE

3. CLIMATE
Cloud Intelligent Monitoring Atmosphere Technology – Παρακολούθηση ατμοσφαιρικών συνθηκών με χρήση αισθητήρων στο Νέφος

3.1 Η λειτουργικότητα του CLIMATE

3.2 Τα χαρακτηριστικά που μπορεί ο χρήστης να παρακολουθεί

3.3 Η αρχιτεκτονική του CLIMATE

3.3.1 Παραγωγοί και Καταναλωτές (Producers & Consumers)

3.3.2 Διεπαφή Συστήματος και Διεπαφή Χρηστών (Front-End & Back-End)

3.4 Οι υπηρεσίες, οι μηχανιστμοί και οι Generic Enablers

3.4.1 Netatmo authorization GE

3.4.2 Μηχανισμός συλλογής δεδομένων από τους αισθητήρες (Sensor Data Collector)

3.4.3 Υπηρεσία Σύνδεσης (Connectivity Service)

3.4.4 Orion Context Broker GE

3.4.5 JSON Storage GE

3.4.6 Keyrock Identity Management GE

3.4.7 History DB GE

3.4.8 Η Λογική Συστήματος (Application Logic) του CLIMATE

3.5 Το γραφικό περιβάλλον του συστήματος (GUI) – Ενδεικτική χρήση του CLIMATE

4 Υλοποίηση

4.1.1 Οι αισθητήρες

4.1.2 Η εγκατάσταση των αισθητήρων

4.1.3 Δημιουργία εφαρμογής στην υποδομή Νέφους του Netatmo

4.2.1 Παραμετροποίηση της υποδομής Νέφους του εργαστηρίου (Intellicloud)

4.2.2 Τοπολογία Δικτύου (Network Topology)

4.3.1 Δημιουργία μιας Εικονικής Μηχανής(Virtual Machine – VM)

4.3.2 Παραμετροποίηση της Εικονικής Μηχανής

4.4.1 Δημιουργία μιας Εικονικής Μηχανής του Orion Context Broker GE

4.4.2 Δημιουργία οντότητας(Entity) στον Context Broker GE

4.5.1 Δημιουργία μιας εικονικής μηχανής του JSON Storage GE

4.5.2 Παραμετροποίηση της Εικονικής Μηχανής του JSON Storage GE

4.6.1 Λήψη των μετρήσεων από το Νέφος του Netatmo – Μηχανισμός συλλογής δεδομένων

4.6.2 Ανανέωση της οντότητας στον Context Broker GE

4.7 Παραμετροποίηση της Εικονικής Μηχανής του Keyrock – Identity Management GE

4.8 Ο μηχανισμός ταυτοποίησης των χρηστών

4.9 Ο μηχανισμός δημιουργίας συνδρομής στον Orion Context Broker GE

4.10 Ο μηχανισμός με τον οποίο αποθηκεύονται οι απαντήσεις του Orion Context Broker GE

4.11 Ο μηχανισμός της ανανέωσης και της διαγραφής μιας συνδρομής στον Orion Context Broker GE

4.12 Ο μηχανισμός με τον οποίο γίνεται η παρακολούθηση των τιμών

4.13 Ο μηχανισμός αποσύνδεσης του χρήστη

4.14 Ο μηχανισμός με τον οποίο διατηρείται η συνεδρία του χρήστη

4.15 Περιπτώσεις χρήσης

5.Συμπεράσματα και μελλοντική δουλειά

5.1 Συμπεράσματα

5.2 Μελλοντική δουλειά

5.2.1 Δυναμική προσθαφαίρεση αισθητήρων

5.2.1 Ενημέρωση και έλεγχος συσκευών με βάση τα κατώφλια των τιμών που μελετάμε

6. Βιβλιογραφία

1. Εισαγωγή

1.1 Εφαρμογές με αισθητήρες και η ιδέα του Διαδικτύου των Πραγμάτων (IoT)

Τα τελευταία χρόνια μία μεγάλη ποικιλία από αισθητήρες έχουν μπει στην καθημερινότητας μας και πλέον βρίσκουν ευρεία εφαρμογή, τόσο στον τομέα της παραγωγής όσο και στην προσωπική μας ζωή. Οι αισθητήρες αυτοί χρησιμοποιούνται προκειμένου να συλλέξουν πληροφορίες από το περιβάλλον και την ανθρώπινη δραστηριότητα και έπειτα μέσω του διαδικτύου κατάλληλα συστήματα να αποθηκεύσουν και να επεξεργαστούν τα δεδομένα που παράγουν. Οι αισθητήρες συναντώνται πλέον παντού από τη στιγμή που βρίσκονται ενσωματωμένοι σε μια σειρά από συσκευές καθημερινής χρήσης όπως τα κινητά και τα αυτοκίνητα καθώς και σε εφαρμογές που συναντούμε στην ιατρική, τη βιομηχανία και την ρομποτική. Οι μετρήσεις που παίρνουν είναι κάθε είδους, από την κίνηση του ανθρωπίνου σώματος και την καρδιακή λειτουργία μέχρι την ατμοσφαιρική πίεση και την επιτάχυνση.

Οι αισθητήρες αυτοί έρχονται τα τελευταία χρόνια να θεμελιώσουν το "νέο" αναδυόμενο Διαδίκτυο (Internet), το λεγόμενο και Διαδίκτυο των πραγμάτων (Internet of Things - IoT)[1]. Η τάση είναι ότι πλέον όλα μας τα εργαλεία, όχι μόνο οι υπολογιστές, αλλά οι προσωπικές μας συσκευές, τα οχήματά μας ακόμα και ενεργειακά δίκτυα, μπορούν μέσω του Διαδικτύου και των αισθητήρων να επικοινωνήσουν μεταξύ τους και να θέσουν έτσι μια νέα μορφή οργάνωσης της πληροφορίας, της παραγωγής υπηρεσιών και προϊόντων. Η ιδέα του Διαδικτύου των Πραγμάτων βασίζεται σε δύο μείζονες τεχνολογικές εξελίξεις. Την ψηφιοποίηση όλο και περισσότερων μηχανών και την παραγωγή μεγάλου όγκου δεδομένων[1].

1.2 Βασικά χαρακτηριστικά των αισθητήρων και των αισθητήρων παρακολούθησης του περιβάλλοντος (Environmental Monitoring Sensors)

Έχει νόημα από τη στιγμή που θα ασχοληθούμε με δεδομένα τα οποία συλλέγονται από αισθητήρες να γνωρίζουμε ότι μεγάλη σημασία για τα αποτελέσματα και τα συμπεράσματα τα οποία εξάγουμε έχουν τα χαρακτηριστικά κάθε αισθητήρα τα οποία καθορίζονται από το είδος του αισθητήρα και τον κατασκευαστή. Ορισμένα από τα χαρακτηριστικά αυτά είναι: το Εύρος, δηλαδή τα όρια που μια συσκευή λειτουργεί αξιόπιστα, το Σφάλμα, δηλαδή η διαφορά μεταξύ στην μετρούμενη τιμή και την πραγματική, η Απόκριση, δηλαδή ο χρόνος που απαιτείται για να λάβει την

[1] http://www-01.ibm.com/software/info/internet-of-things/

τελική τιμή η έξοδος κ.α.[2].

Στο δικό μας σύστημα, το CLIMATE, θα χρησιμοποιήσουμε αισθητήρες για την παρακολούθηση των περιβαλλοντικών συνθηκών (Enviromental Monitoring Sensors). Οι αισθητήρες περιβάλλοντος εν γένει χρησιμοποιούνται για τη παρακολούθηση των συνθηκών του περιβάλλοντος είτε σε κλειστούς χώρους, όπως χώροι εργασίας και αποθηκευτικοί χώροι είτε σε ανοιχτούς χώρους όπως το εσωτερικό μιας πόλης ή γεωργικές εκτάσεις. Τέτοιου είδους αισθητήρες είναι απαραίτητοι για την δημιουργία συστημάτων παρακολούθησης και μελέτης των καιρικών μεταβολών και κυρίως της ποιότητας του περιβάλλοντος χώρου στον οποίο ζει και εργάζεται ο άνθρωπος.

Για την υλοποίηση του CLIMATE θα χρησιμοποιήσουμε τους αισθητήρες της Netatmo και συγκεκριμένα τους "Netatmo – Urban Weather Station"[2]. Οι αισθητήρες αυτοί καταγράφουν την ποιότητα του αέρα και συγκεκριμένα τη θερμοκρασία, την υγρασία, τη βαρομετρική πίεση, το διοξείδιο του άνθρακα και την ένταση του ήχου.

1.3 Αισθητήρες και Υπολογιστικό Νέφος (Cloud Computing)

Οι αισθητήρες αποτελούν συσκευές που ανιχνεύουν τη μεταβολή ενός ή περισσοτέρων φυσικών μεγεθών και παράγουν με βάση αυτό μια μετρήσιμη έξοδο. Πλέον η πλειοψηφία των αισθητήρων που συναντάμε ως εισόδους στα υπολογιστικά συστήματα, παράγουν ως έξοδο πληροφορίες δεδομένα τα οποία περιγράφονται από γνωστά «σχήματα» όπως τα JSON, XML κ.α. τα οποία επιτρέπουν την εύκολη ανάλυση και επεξεργασία τους.

Το Υπολογιστικό Νέφος αποτελεί ένα νέο μοντέλο ανάπτυξης εφαρμογών και πληροφοριακών συστημάτων το οποίο επιτρέπει τη χρήση δικτύων, αποθηκευτικού χώρου, επεξεργαστικής ισχύς, υπηρεσιών κ.α. από ένα «κοινό ταμείο» υπολογιστικών πόρων μέσω του Διαδικτύου. Οι υπηρεσίες και οι υπολογιστικοί πόροι διατίθενται από έναν ή περισσότερους παρόχους και οι καταναλωτές κάνουν χρήση αυτών των υπηρεσιών και των υπολογιστικών πόρων ανάλογα με τις ανάγκες τους.

1.3.1 Βασικά μειονεκτήματα προγενέστερων υποδομών και τεχνολογιών του Υπολογιστικού Νέφους

Το Νέφος αποτελεί πλέον ίσως το ιδανικό περιβάλλον προκειμένου να αναπτυχθούν εφαρμογές για αισθητήρες. Αυτό γιατί τα

[2] https://www.netatmo.com/en-US/product/weather-station

συστήματα που είχαν αναπτυχθεί κατά καιρούς για τη συλλογή και την επεξεργασία δεδομένων από αισθητήρες είχαν κάποια βασικά μειονεκτήματα:

- Τα περισσότερα συστήματα και εφαρμογές ήταν αναπτυγμένα πάνω σε έναν συγκεκριμένο μοντέλο αισθητήρα(sensor-specific). Ο χρήστης δεν είχε τη δυνατότητα να χρησιμοποιήσει διαφορετικού τύπου αισθητήρες ή ακόμα να προσθαφαιρέσει αισθητήρες στο σύστημά του, δημιουργώντας έτσι ένα σύστημα για κάθε ένα συγκεκριμένο τύπο αισθητήρα.

- Η λειτουργία των αισθητήρων και τα δεδομένα τα οποία παράγουν δεν ήταν προσβάσιμα μέσω Διαδικτύου.

- Εφόσον οι εφαρμογές ήταν αναπτυγμένες σε τοπικές υποδομές, αυτό είχε ως αποτέλεσμα είτε τον περιορισμό των υπολογιστικών πόρων, είτε της επεξεργαστικής ισχύος, είτε του χώρου αποθήκευσης των δεδομένων. Υπήρχε περιορισμός στο να έχουν πρόσβαση στο σύστημα πολλοί χρήστες. Είτε προσθέτοντας αισθητήρες για νέες μετρήσεις, είτε απλά έχοντας πρόσβαση στις ήδη υπάρχουσες βάσεις δεδομένων. Αυτό είχε ως βασικό μειονέκτημα των περιορισμό των δεδομένων που μπορούν να συλλεχθούν και σαφώς τον περιορισμό των χρηστών-και κυρίως των απομακρυσμένων χρηστών-που μπορούσαν να εξυπηρετηθούν[3].

1.3.2 Τα άμεσα πλεονεκτήματα της τεχνολογίας του Νέφους

Το Υπολογιστικό Νέφος έρχεται να δώσει μια λύση σε αυτά τα προβλήματα. Η λογική του Νέφους βασίζεται στη δυνατότητα του να προσφέρει μεγάλα αποθέματα πόρων, υπολογιστής ισχύος και αποθηκευτικών χώρων προσφέροντάς τα ως διαδικτυακές υπηρεσίες με σκοπό τη συλλογή, την αποθήκευση και την επεξεργασία του μεγάλου όγκου δεδομένων που παράγονται από τους αισθητήρες.

Το Υπολογιστικό Νέφος ως φιλοσοφία ανάπτυξης υπολογιστικών συστημάτων, ήρθε να δώσει κάποιες νέες δυνατότητες στο χώρο της πληροφορικής. Η ανεξαρτησία από ποια τοποθεσία προσφέρονται οι υπηρεσίες, η ελαστικότητα στην παροχή, τη δέσμευση και την αποδέσμευση πόρων καθώς και η μετρήσιμη προσφορά υπηρεσιών αναδεικνύουν το Νέφος ως το ιδανικό περιβάλλον για την ανάπτυξη συστημάτων και εφαρμογών στο Διαδίκτυο.

Θα μπορούσαμε εδώ να "κωδικοποιήσουμε" τα χαρακτηριστικά που κάνουν το Νέφος να διαφέρει από οποιαδήποτε άλλη αρχιτεκτονική[4]:

- Αυτο-εξυπυπηρέτηση ανάλογα με τις απαιτήσεις και τη ζήτηση
- Πρόσβαση μέσω διαδικτύου
- Κοινό ταμείο πόρων
- Ελαστικότητα και γρήγορη δέσμευση - αποδέσμευση πόρων
- Μετρήσιμη παροχή υπηρεσιών

1.4 Περιγραφή του προβλήματος

Αφορμή για τη διπλωματική εργασία αποτέλεσε η ιδέα για σχεδίαση ενός συστήματος παρακολούθησης της ποιότητας του αέρα είτε σε εσωτερικούς είτε εξωτερικούς χώρους, κάνοντας αποκλειστικά χρήση υπηρεσιών και πόρων που βρίσκονται στο Νέφος.

Ένα σύστημα το οποίο θα δίνει τη δυνατότητα παρακολούθησης της ποιότητας του αέρα θα μπορούσε να χρησιμοποιηθεί είτε από οργανισμούς-επιχειρήσεις που τους ενδιαφέρει να μελετάνε και να διατηρούν την ποιότητα του περιβάλλοντος αέρα σε συγκεκριμένα επίπεδα είτε για εμπορικούς σκοπούς.

Συγκεκριμένα θα μπορούσε να βρει εφαρμογή σε νοσοκομεία και κλινικές, στις οποίες η ποιότητα των περιβαλλοντικών συνθηκών που αναρρώνει ο ασθενής παίζουν ιδιαίτερο ρόλο. Επίσης σε σχολεία και χώρους δουλειάς, όπου επίσης οι συνθήκες μέσα στις οποίες φοιτούν και δουλεύουν οι άνθρωποι επηρεάζει τη συγκέντρωση και την αποδοτικότητά τους. Αντίστοιχα εμπορικές εφαρμογές που θα εξυπηρετούσε το σύστημά μας είναι η παρακολούθηση των συνθηκών εντός αποθηκευτικών χώρων ή αξιολόγηση τουριστικών περιοχών, χώρων αναψυχής κ.α. με βάση τις περιβαλλοντικές συνθήκες που επικρατούν εκεί.

Το σύστημα σκοπεύουμε να εξυπηρετεί δύο ειδών χρήστες. Τους διαχειριστές και τους τελικούς χρήστες-καταναλωτές. Οι διαχειριστές έχουν τη δυνατότητα εγκατάστασης νέων αισθητήρων στο σύστημά μας και ορισμό συνεπώς νέων χώρων παρακολούθησης με τα αντίστοιχα χαρακτηριστικά τους. Οι τελικοί χρήστες-καταναλωτές έχουν το δικαίωμα εγγραφής στο σύστημα, δημιουργίας προσωπικής συνδρομής μέσω της οποίας περιγράφουν ποια χαρακτηριστικά ενδιαφέρονται να παρακολουθούν με σκοπό τη λήψη δεδομένων που παράγουν οι αισθητήρες.

Συνεπώς οι απαιτήσεις του συστήματός μας θα μπορούσαν να κωδικοποιηθούν ως εξής:

- Ο χρήστης-διαχειριστής έχει τη δυνατότητα να συνδέει πάνω στο σύστημα αισθητήρες παρακολούθησης της ποιότητας του περιβάλλοντος χώρου.

- Ο χρήστης-καταναλωτής έχει πρόσβαση στις πληροφορίες-δεδομένα που παράγουν αυτοί οι αισθητήρες.

- Ο χρήστης-διαχειριστής δεν έχει περιορισμό στον αριθμό των αισθητήρων που θα χρησιμοποιήσει ως εισόδους.

- Ο χρήστης-καταναλωτής έχει τη δυνατότητα παρακολούθησης των αλλαγών που συμβαίνουν στα χαρακτηριστικά της ποιότητας του αέρα σε πραγματικό χρόνο και έχει πρόσβαση στο ιστορικό των μετρήσεων, από όλους τους αισθητήρες.

- Ο χρήστης-καταναλωτής παρακολουθεί μόνο τα δεδομένα που τον ενδιαφέρουν καθώς επίσης ενημερώνεται για τα νέα δεδομένα μόνο όταν αλλάζουν οι τιμές τους. Επίσης έχει και τη δυνατότητα επαναπροσαρμογής των προσωπικών του ρυθμίσεων ανά πάσα στιγμή.

- Ο χρήστης έχει πρόσβαση σε όλα τα παραπάνω αποκλειστικά από το διαδίκτυο.

- Το σύστημα προσφέρει μηχανισμό ταυτοποίησης και εγγραφής νέων χρηστών.

- Το σύστημά έχει τη δυνατότητα αποθήκευσης του μεγάλου όγκου των πληροφοριών που παράγουν οι αισθητήρες και διατήρησης πλήρους ιστορικού των δεδομένων για όλους τους αισθητήρες.

- Ο χρήστης έχει πρόσβαση σε γραφήματα με τις διακυμάνσεις των τιμών των χαρακτηριστικών που έχει επιλέξει να παρακολουθεί.

1.5 Στόχοι της δουλειάς μας και περιγραφή της λύσης

1.5.1 Το μοντέλο των αισθητήρων

Για την υλοποίηση του συστήματός θα χρησιμοποιήσουμε τους αισθητήρες παρακολούθησης της ποιότητας του περιβάλλοντος χώρου "Netatmo - Urban Weather Station". Ο 1ος αισθητήρας μπορεί και παίρνει μετρήσεις για θερμοκρασία, υγρασία, διοξείδιο του άνθρακα (CO2), πίεση και θόρυβο. Ο 2ος παίρνει

μετρήσεις μόνο για θερμοκρασία και υγρασία. Οι δύο αυτοί αισθητήρες μπορούν να λαμβάνουν τιμές από 2 διαφορετικές τοποθεσίες.

1.5.2 Οι υπηρεσίες που θα χρησιμοποιήσουμε

Το σύστημα υλοποιήθηκε στο Νέφος χρησιμοποιώντας την υπολογιστική υποδομή του εργαστηρίου-Intellicloud[3] και τις υπηρεσίες IaaS που προσφέρει. Επίσης κάνοντας χρήση υπηρεσιών της υποδομής PaaS που προσφέρει το FI-WARE καθώς επίσης και υπηρεσιών που προσφέρει η υποδομή Νέφους της εταιριας Netatmo.

Το FI-WARE[4] αποτελεί μια ανοιχτή υποδομή ανάπτυξης και παροχής υπηρεσιών Νέφους βασισμένη στη λογική του Διαδικτύου του Μέλλοντος (Future Internet) με σκοπό την προώθηση της καινοτομίας και της επιχειρηματικότητας μέσω οικονομικά αποδοτικών λύσεων.

Οι υπηρεσίες πάνω στις οποίες θα βασιστούμε είναι οι εξής:

- **Netatmo Authorization GE**
  Υπηρεσία η οποία προσφέρεται από την υποδομής Νέφους της εταιρίας Netatmo

- **Keyrock-Identity Management GE**
  Υπηρεσία η οποία διατίθεται μέσω της πλατφόρμας του FI-WARE και φιλοξενείται στην υποδομή του FI-LAB[5]

- **Orion-Publish/Subscribe Context Broker GE**
  Υπηρεσία η οποία διατίθεται μέσω της πλατφόρμας του FI-WARE και φιλοξενείται στην υποδομή του Intellicloud

- **History DB GE**
  Υπηρεσία η οποία αναπτύχθηκε στο εργαστήριο μας και φιλοξενείται στην υποδομή του Intellicloud

- **JSON Storage GE**
  Υπηρεσία η οποία αναπτύχθηκε στο εργαστήριο μας και φιλοξενείται στην υποδομή του Intellicloud

[3] http://cloud.intellicloud.tuc.gr/
[4] http://www.fi-ware.org/
[5] http://www.fi-ware.org/lab/

1.5.3 Η δομή της εργασίας

Στο 2° κεφάλαιο θα γίνει αναφορά στις βασικές τεχνολογίες και τα εργαλεία του Υπολογιστικού Νέφους. Τόσο στις πιο διαδεδομένες μορφές του και στα μοντέλα παροχής υπηρεσιών όσο και στα βασικά λειτουργικά χαρακτηριστικά του και στον τρόπο με το οποίο ωφελούνται από αυτό οι σύγχρονες αρχιτεκτονικές ανάπτυξης πληροφοριακών συστημάτων. Επίσης θα γίνει αναφορά στο μοντέλο της αρχιτεκτονικής πάνω το οποίο θα βασιστούμε για την ανάπτυξη του CLIMATE, δηλάδη της υπηρεσιο-κεντρικής αρχιτεκτονικής (Service Orianted Architecture), στα βασικά εργαλεία επικοινωνίας και διεπαφής μεταξύ των διαδικτυακών υπηρεσιών (REST, API, HTTP κ.α.), στις παρεχόμενες υπηρεσίες Γενικού και Ειδικού Σκοπού(Generic & Specific Enablers) και στην πλατφόρμα του FI-WARE.

Στο 3° κεφάλαιο θα περιγράψουμε εκτενώς το CLIMATE. Θα μιλήσουμε για τη λειτουργικότητα του συστήματος, την αρχιτεκτονική του, τις υπηρεσίες που χρησιμοποιήθηκαν για τη σύνθεσή του, τον τρόπο με τον οποίο επικοινωνούν οι υπηρεσίες μεταξύ τους και ανταλλάσσουν πληροφορίες. Επίσης θα υπάρξει αναλυτική περιγραφή των υπηρεσιών αυτών όπως επίσης και του γραφικού περιβάλλοντος (GUI) μέσω του οποίου επικοινωνεί ο χρήστης με το σύστημα.

Στο 4° κεφάλαιο θα αναπτύξουμε τα βήματα με τα οποία υλοποιήσαμε το CLIMATE. Θα κάνουμε αναφορά στον τρόπο με τον οποίο παραμετροποιήσαμε τις Υποδομές Νέφους που χρησιμοποιούμε και τις αντίστοιχες υπηρεσίες και θα περιγράψουμε τον τρόπο με τον οποίο υλοποιήσαμε την επικοινωνιία μεταξύ τους. Τέλος στο κεφάλαιο αυτό, θα αναφέρουμε τρεις περιπτώσεις χρήσης και εφαρμογής του CLIMATE.

Στο 5° κεφάλαιο θα παραθέσουμε τα συμπεράσματα που προέκυψαν από την ενασχόλησή μας με την υλοποίηση του CLIMATE καθώς και προτάσεις για μελλοντική δουλειά.

Στο 6° κεφάλαιο θα αναφέρουμε τις πηγές και τους αντίστοιχους συνδέσμους τους, στις οποίες στηριχθήκαμε για τη υλοποίηση του συστήματος.

2. Υπάρχουσα τεχνολογία

2.1 Ο ορισμός του Υπολογιστικού Νέφους

Το Υπολογιστικό Νέφος (Cloud Computing) είναι ένα μοντέλο που επιτρέπει την εύκολη πρόσβαση μέσω διαδικτύου τη στιγμή που ζητείται σε ένα "κοινό ταμείο" από παραμετροποιήσιμους υπολογιστικούς πόρους όπως δίκτυα, αποθηκευτικός χώρος, εφαρμογές και υπηρεσίες οι οποίοι μπορούν πολύ εύκολα να παρακολουθούνται με πολύ μικρή παρέμβαση της διαχείρισης, ή αλληλεπίδρασης από τον πάροχο των υπηρεσιών. Αυτό το τεχνολογικό μοντέλο που ακούει στο όνομα Υπολογιστικό Νέφος έχει ως κεντρική ιδέα τη ευέλικτη διάθεση υπολογιστικών πόρων και απαρτίζεται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα παροχής της υπηρεσίας και τέσσερα μοντέλα υλοποίησης[5].

2.2 Η ιστορία ανάπτυξης του Υπολογιστικού Νέφους

Κατά το παρελθόν, στη βιομηχανία της πληροφορικής δημιουργήθηκε η ανάγκη ανάπτυξης "υπερ-υπολογιστικών" συστημάτων που θα εξυπηρετούσαν από άποψη πόρων τις υπολογιστικές απαιτήσεις της εποχής. Μια προσέγγιση του προβλήματος ήταν η υλοποίηση υπερ-υπολογιστικών μονάδων τύπου Blue Gene[6] με εκατοντάδες και πλέον επεξεργαστές. Μια δεύτερη προσέγγιση ήταν η δημιουργία συστάδων υπολογιστών (Clusters), δηλαδή ένα δίκτυο μικρών χαμηλού κόστους υπολογιστών οι οποίοι δουλεύουν όλοι μαζί ως ένας υπερ-υπολογιστής.

Οι συστάδες υπολογιστών χρησιμοποιήθηκαν ευρέως στη βιομηχανία και υιοθετήθηκαν από πολλές εταιρίες πληροφορικής. Η συστάδα εμφάνισε ένα στοιχείο του σύγχρονου Υπολογιστικού Νέφους ως προς τον τρόπο που αντιλαμβάνεται ο χρήστης τους πόρους που καταναλώνει και έχει στη διάθεσή του. Ο χρήστης δεν ενδιαφέρεται ποια μονάδα επεξεργασίας χρησιμοποιεί για να τρέξει το πρόγραμμα του αλλά η συστάδα υπολογιστών δίνει την εγγύηση ότι ο κώδικας θα τρέξει στην καλύτερη δυνατή διαθέσιμη μονάδα εκείνη τη στιγμή[6].

Στις αρχές της δεκαετίας του 90' ήρθε στην επιφάνεια μια νέα ιδέα που ονομάστηκε Υπολογιστικό Πλέγμα (Grid), ως εξέλιξη της συστάδας υπολογιστών, στο οποίο οι χρήστες μπορούσαν να συνδεθούν και να εκμεταλλευτούν μια υπηρεσία, της οποίας η χρήση από άποψη πόρων και χρόνου μπορούσε να μετρηθεί. Η τεχνολογία του Υπολογιστικού Πλέγματος επέκτεινε τις τεχνικές της συστάδας υπολογιστών, όπου πολλές διασυνδεδεμένες ανεξάρτητες συστάδες υπολογιστών μπορούσαν να λειτουργήσουν ως

[6] http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/bluegene/

πλέγμα και ακόμα ως αυτόνομο σύστημα.

Κάποια προβλήματα που είχε το Πλέγμα Υπολογιστών, όπως η διαχείριση της αποθήκευσης, η επίβλεψη της ασφάλειας και η μετακίνηση δεδομένων ήταν ζητήματα που έπρεπε το Πλέγμα Υπολογιστών να επιλύσει για να προχωρήσει ως ιδέα. Ειδικά η ασφάλεια των δεδομένων και η εμπιστευτικότητα ήταν το κυρίαρχο ζήτημα. Πηγαίνοντας το Πλέγμα Υπολογιστών ένα βήμα πιο μπροστά στην παροχή υπηρεσίας, κυρίως στο ζήτημα της εικονικοποίησης των παρεχόμενων πόρων και της σχετικής αυτονομίας που προσφέρει στις Εικονικές Μηχανές και τις διατιθέμενες εφαρμογές, είναι το Υπολογιστικός Νέφος το οποίο ενσωματώνει ιδέες από το Υπολογιστικό Πλέγμα και τις ολοκληρώνει σε υπηρεσίες[7].

2.3 Πέντε ιστορικοί σταθμοί για το Υπολογιστικό Νέφος

Πέντε βασικά γεγονότα-σταθμοί που εδραίωσαν το Υπολογιστικό Νέφος στην παγκόσμια αγορά πληροφορικής ήταν:

-Έναρξη των Amazon Web Services τον Ιούλιο του 2002[7]
-Έναρξη του S3 (Simple Storage Service) το Μάρτιο του 2006[8]
-Έναρξη του EC2 (Elastic Compute Cloud) τον Αύγουστο του 2006[9]
-Έναρξη του Google App Engine τον Απρίλιο του 2008[10]
-Έναρξη του Windows Azure Beta το Νοέμβριο του 2009[11]
[8]

2.4 Πως ωφελείται άμεσα ο χρήστης-καταναλωτής του Υπολογιστικού Νέφους;

- Η ψευδαίσθηση των άπειρων υπολογιστικών πόρων που μπορούν να είναι διαθέσιμοι κατά τη ζήτηση τους, εξαλείφει την ανάγκη για τους χρήστες του Υπολογιστικού Νέφους να προγραμματίζουν τις ανάγκες τους από άποψη υπολογιστών πόρων για μεγάλα χρονικά διαστήματα. Αν για ένα χρήστη ή μια επιχείρηση δημιουργηθεί μελλοντικά η ανάγκη χρήσης περισσότερων υπολογιστικών πόρων και ισχυρότερης υπολογιστκής ισχύος τότε άμεσα μπορούν να έχουν πρόσβαση σε αυτά.

- Μια επιχείρηση δε χρειάζεται να δεσμεύσει εκ των προτέρων

[7] http://aws.amazon.com/
[8] http://aws.amazon.com/s3/
[9] http://aws.amazon.com/ec2/
[10] https://cloud.google.com/appengine/
[11] https://www.remoteapp.windowsazure.com/

κεφάλαιο για την αγορά μεγάλης υπολογιστικής υποδομής την οποία στα πρώτα βήματά της δε θα την αξιοποιήσει. Αυτό είναι εφικτό επειδή μπορεί να ξεκινήσει με μια μικρή επένδυση σε υποδομή και παράλληλα καθώς μεγαλώνουν οι ανάγκες της να κάνει χρήση των πόρων που προσφέρονται από το Νέφος.

- Η δυνατότητα της πληρωμής ανά χρήση υπολογιστικής ισχύος αλλά και αποθηκευτικού χώρου ανά τακτά χρονικά διαστήματα (π.χ. επεξεργαστές ανά ώρα ή αποθηκευτικός χώρος ανά μέρα) και η απελευθέρωση τους όταν πλέον δεν είναι χρήσιμα. Με αυτόν τον τρόπο οι καταναλωτές των πόρων πληρώνουν με βάση το πόσο καταναλώνουν. Έπειτα το κεφάλαιο αυτό επενδύεται από τον πάροχο για τη συντήρηση και την αναβάθμιση των υπηρεσιών όπως γίνεται και στην αγορά της ηλεκτροδότησης.

2.5.1 Μοντέλα παροχής υπηρεσιών Νέφους

- **Υπολογιστκή 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 αποθηκευτικά μέσα. Με αυτή τη μορφή, οι χρήστες μπορούν να δημιουργήσουν τη δική τους Υπολογιστική Υποδομή πάνω στην οποία είναι οι ίδιοι υπεύθυνοι στο να εγκαταστήσουν, να διατηρήσουν και να τρέξουν τις δικές τους εφαρμογές. 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 μέσω του διαδικτύου. Οι εξυπηρετητές, τα αποθηκευτικά συστήματα, τα 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 EC2[12], Rackspace[13] 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, δοκιμή,

[12] http://aws.amazon.com/ec2/
[13] http://www.rackspace.com/managed-cloud/

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 Engine[14], Windows Azure[15], Amazon Elastic Beanstalk[16] κ.α.

- **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[17], Yahoo Mail[18], Dropbox[19] κ.α.[9]

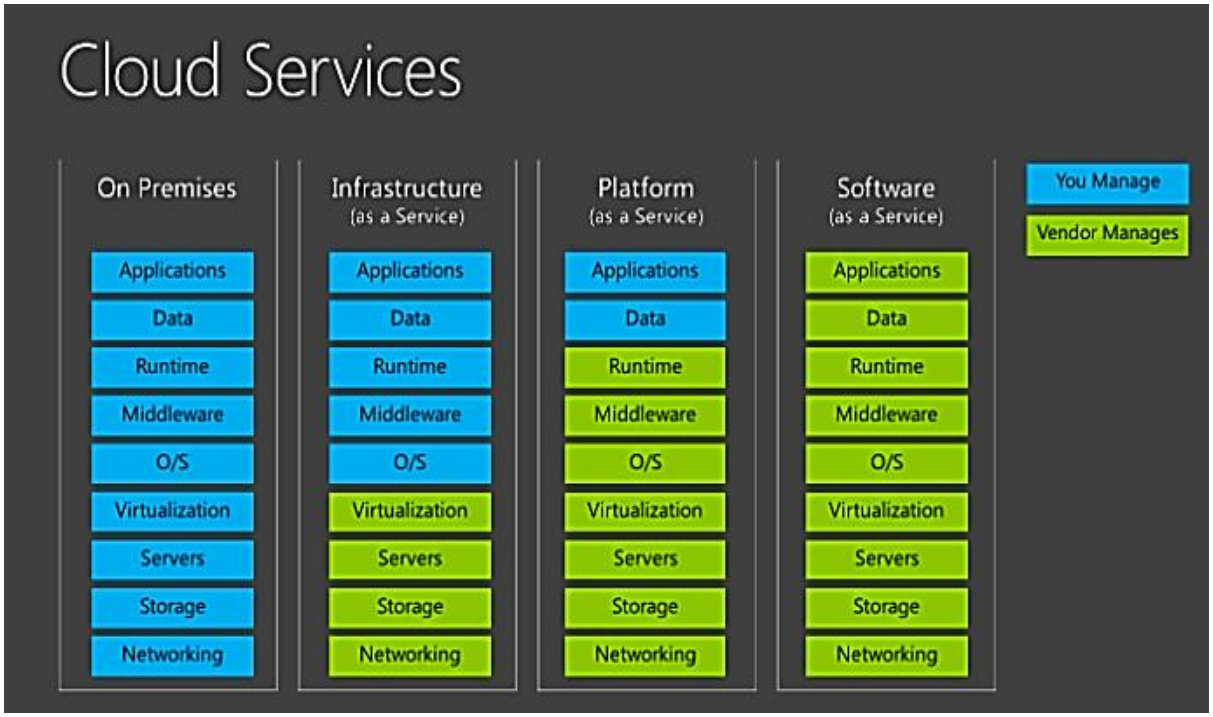
Όπως βλέπουμε λοιπόν και από το 'Σχήμα 1' και στα τρία μοντέλα παροχής υπηρεσιών τα δίκτυα, ο αποθηκευτικός χώρος, οι εξυπηρετητές και ο μηχανισμός της εικονικοποίησης διαχειρίζονται από τον πάροχο της υπηρεσίας.

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.

Και στο μοντέλο Λογισμικό ως Υπηρεσία ο καταναλωτής περιορίζεται μόνο στις υπηρεσίες που του προσφέρει μια εφαρμογή.

[14] https://cloud.google.com/appengine/
[15] https://www.remoteapp.windowsazure.com
[16] http://aws.amazon.com/elasticbeanstalk/
[17] http://www.google.com/work/apps/business/
[18] https://mail.yahoo.com/
[19] https://www.dropbox.com



Μοντέλα παροχής υπηρεσιών Νέφους – IaaS,PaaS,SaaS
Σχήμα 1

2.5.2 Υλοποιήσεις υποδομών Νέφους

Τα **Ιδιωτικά Σύννεφα (Private Cloud)** δίνουν στους χρήστες άμεση πρόσβαση σε υπολογιστικούς πόρους οι οποίοι φιλοξενούνται στην υποδομή ενός συγκεκριμένου οργανισμού. Οι χρήστες μπορούν να ελέγξουν από μόνοι τους και να τροποποιήσουν το μέγεθος των πόρων που παίρνουν, συχνά μέσω μιας διεπαφής διαδικτυακής υπηρεσίας. Ωστόσο, επειδή έχει εφαρμοστεί εντός ενός ήδη υπάρχοντος υπολογιστικού κέντρου (data center) κάποιου οργανισμού, το Ιδιωτικό Νέφος υπόκειται στους περιορισμούς ασφάλειας του οργανισμού, φυσικούς, ηλεκτρονικούς, και επομένως παρέχει μια μεγαλύτερη ασφάλεια σε ευαίσθητα δεδομένα και κώδικα.

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 από κάποιον από τους αναφερθέντες οργανισμούς, ή την εποπτεία του να την έχει ένας τρίτος οργανισμός ή επιχείρηση.

Τα **Δημόσια ή Κοινόχρηστα Σύννεφα (Public Cloud)** παρέχουν πρόσβαση σε υπολογιστικούς πόρους στο γενικό κοινό μέσω του διαδικτύου. Ο πάροχος του επιτρέπει στους καταναλωτές να έχουν υπό τον έλεγχο τους, τους πόρους που έχουν ζητήσει, και αυτό γίνεται πιο συχνά μέσω μιας διαδικτυακής διεπαφής. Το ενοίκιο ουσιαστικά του καταναλωτή είναι σύμφωνα με την ανάγκη του "pay as you go"(πληρώνεις όσο χρησιμοποιείς.) Τα Δημόσια

Σύννεφα προσφέρουν πρόσβαση σε μεγάλα "ταμεία" υπολογιστικών πόρων τα οποία μάλιστα είναι αρκετά επεκτάσιμα χωρίς να απαιτούν κάποια επένδυση κεφαλαίου για την ανάπτυξη των υποδομών του υπολογιστικού κέντρου (data center).

Τα **Υβριδικά Σύννεφα (Hybrid Clouds)** συνδυάζουν τους πόρους (π.χ. μηχανήματα, δίκτυο, αποθήκευση κλπ.) που προέρχονται από διαφορετικές υλοποιήσεις υποδομών Νέφους(για παράδειγμα συνδυασμός Ιδιωτικού Νέφους για την αποθήκευση δεδομένων και Δημόσιου Νέφους για την ενοικίαση επεξεργαστική ισχύος)[10].

2.6 Βασικά λειτουργικά χαρακτηριστικά του Υπολογιστικού Νέφους

- **Υπηρεσίες κατ' απαίτηση (On-Demand Service)**

Ο καταναλωτής μπορεί να ζητήσει μονομερώς τις υπολογιστικές δυνατότητες, όπως ο χρόνος που θα χρησιμοποιήσει στον εξυπηρετητή (server) και το μέγεθος του αποθηκευτικού χώρου που θα χρησιμοποιήσει μέσω δικτύου αυτόματα χωρίς να απαιτείται καμία ανθρώπινη αλληλεπίδραση με τον πάροχο της εκάστοτε υπηρεσίας.

- **Προσπέλαση διαδικτυακά από παντού (Ubiquitous Network Access)**

Οι Δυνατότητες αυτές είναι προσπελάσιμες από παντού δια μέσω δικτύου. Υπάρχουσες τεχνολογίες και μηχανισμοί(διεπαφές API, πρωτόκολλα HTTP, προγράμματα περιήγησης κ.α.) λειτουργούν καταλυτικά ώστε να επιτρέπουν τη χρήση και την πρόσβασή τους από πολλές και ετερόκλητες πλατφόρμες χρήστη (π.χ. κινητά τηλέφωνα, φορητούς υπολογιστές, PDA κ.α.).

- **Κοινό ταμείο υπολογιστικών πόρων ανεξαρτήτου τοποθεσίας (Location Independent Resource Pooling)**

Οι υπολογιστικοί πόροι του παρόχου συγκεντρώνονται σε ένα κοινό σύνολο χρησιμοποιώντας ένα μοντέλο πολλών ενοικιαστών, με διαφορετικούς φυσικούς και εικονικούς πόρους οι οποίοι αποδίδονται πολλές φορές δυναμικά μετά από την απαίτηση του χρήστη. Ο καταναλωτής γενικά δεν έχει κανένα έλεγχο και γνώση για την ακριβή τοποθέτηση του παρεχόμενου πόρου, αλλά μπορεί να προσδιορίσει σε ένα πιο αφηρημένο επίπεδο την τοποθεσία όπως η χώρα η πόλη ή το συγκεκριμένο υπολογιστικό κέντρο. Παραδείγματα τέτοιων πόρων είναι ο αποθηκευτικός χώρος, επεξεργασία, μνήμη, εύρος ζώνης δικτύου, και οι εικονικές μηχανές.

- **Ταχεία ελαστικότητα**

Οι δυνατότητες αυτές μπορούν να παρακολουθηθούν εύκολα, και να αυξηθεί η να μειωθεί το μέγεθος τους σχεδόν άμεσα. Στον καταναλωτή – τελικό χρήστη οι δυνατότητες αυτές που είναι

διαθέσιμες να παρακολουθήσει συχνά μοιάζουν να είναι άπειρες και μπορούν να αγοραστούν – αποκτηθούν σε οποιαδήποτε ποσότητα, οποιαδήποτε στιγμή.

- **Μετρήσιμη υπηρεσία**

Τα συστήματα Νέφους αυτόματα ελέγχουν και βελτιστοποιούν τη χρήση των υπολογιστικών πόρων χρησιμοποιώντας κάποια μετρητικά συστήματα σε κάποιο από τα επίπεδα της αφαίρεσης που εισάγουν, κατάλληλα για την συγκεκριμένη παρεχόμενη υπηρεσία (αποθηκευτικού χώρου, υπολογιστικής ισχύος, εύρους ζώνης, ενεργού αριθμού χρηστών κλπ.). Η χρήση των πόρων μπορεί να παρακολουθηθεί, ελεγχθεί και να αναφερθεί ότι παρέχει διαφάνεια και για τις δύο πλευρές, τελικού χρήστη – καταναλωτή και παρόχου της χρησιμοποιούμενης υπηρεσίας[11].

2.7 Εικονικοποίηση Πόρων (Virtualization) και η σχέση του με το Υπολογιστικό Νέφος

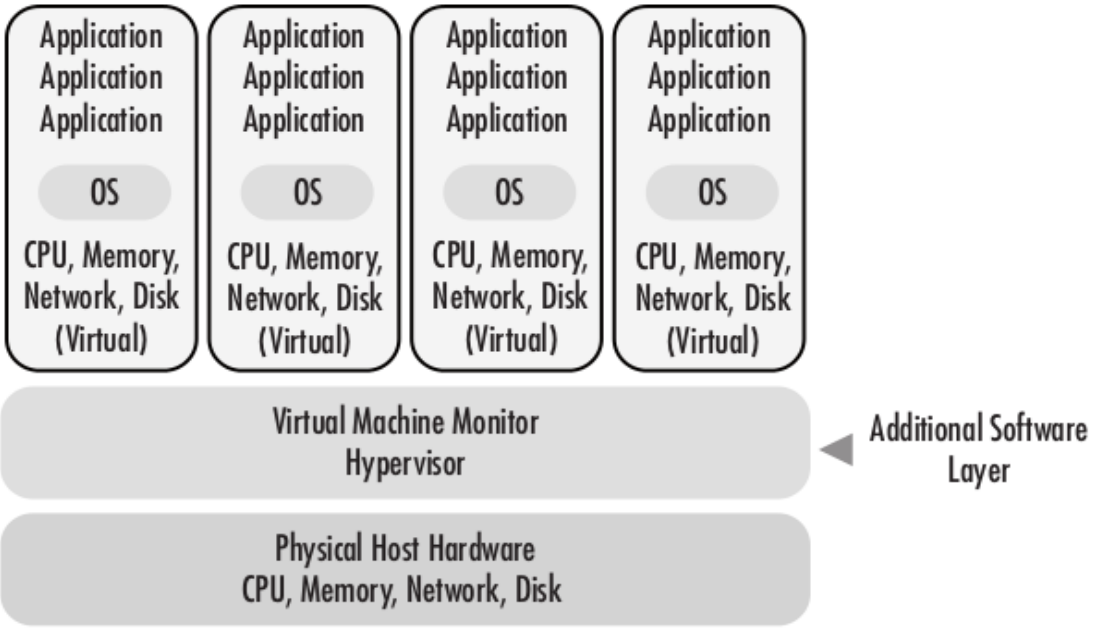
2.7.1 Η τεχνολογία της Εικονικοποίησης Πόρων

Η Εικονικοποίηση αποτελεί την τεχνολογία που διαχειρίζεται την πρόσβαση στους πόρους. Ουσιαστικά περιγράφει ευρέως την τμηματοποίηση των πόρων της υποκείμενης φυσικής υποδομής (CPUs, RAM κλπ.) ώστε να καλυφθούν τα αιτήματα υπηρεσίας προς τη συγκεκριμένη υποδομή. Η τεχνική της Εικονικοποίησης εφαρμόζεται σε όλα τα επίπεδα των υπολογιστικών υποδομών – δίκτυα, αποθήκευση, εξυπηρετητές κτλ. Η τεχνολογιία της εικονικοποίησης των πόρων ή αλλιώς εικονικών υποδομών δημιουργεί ένα επίπεδο ανεξαρτησίας ανάμεσα στις φυσικές υποδομές που λαμβάνουν χώρα οι διαδικασίες εκτέλεσης και στις εφαρμογές που εξυπηρετούνται από αυτές τις υποδομές καταναλώνοντας πόρους[12].

Το όφελος της εικονικοποίησης επαφίεται στο γεγονός ότι οι εικονικές υποδομές παρέχουν στους διαχειριστές του συστήματος τη δυνατότητα και το πλεονέκτημα της διαχείρισης των πόρων από ένα "κοινό ταμείο" όλης της υποδομής γεγονός που τους επιτρέπει αφενός την γρήγορη κάλυψη των αναγκών των καταναλωτών και την καλύτερη οργάνωση των επενδύσεων για νέες υποδομές. Συν τοις άλλοις η χρήση αυτή των εικονικών υποδομών δεν προκαλεί καμία διαταραχή στην εμπειρία του χρήστη-καταναλωτή καθώς δεν μπορεί να παρατηρήσει παρά μόνο πολύ μικρές μεταβολές στη λειτουργία του συστήματος.

Συγκεκριμένα στις υποδομές Νέφους, η τεχνική της εικονικοποίησης επιτρέπει σε πολλαπλές Εικονικές Μηχανές με τα αντίστοιχα λειτουργικά τους συστήματα να τρέχουν ταυτόχρονα στην ίδια υπολογιστική υποδομή και να προσφέρουν ουσιαστικά

διαφορετικές υπηρεσίες. Φαίνεται σαν κάθε λειτουργικό σύστημα να μοιράζεται τους διαθέσιμους πόρους από μία κοινή φυσική υπολογιστική υποδομή[13].



Η σχέση μεταξύ Φυσικής Υποδομής – Hypervisors – Εικονικών Μηχανών
Σχήμα 2

2.7.2 Η τεχνολογία του Επιμελητή Εικονικών Μηχανών (Virtual Machine Monitor – Hypervisor)

Όπως βλέπουμε στο 'Σχήμα 2' το πρόσθετο στρώμα λογισμικού το οποίο μεσολαβεί μεταξύ της Φυσικής Υποδομής και των Εικονικών Μηχανών αναφέρεται ως Επιμελητής Εικονικών Μηχανών – Hypervisor (VMM-Hypervisor). Ουσιαστικά μπορούμε να φανταστούμε τον Επιμελητή Εικονικών Μηχανών ως ένα Λειτουργικό Σύστημα το οποίο εξυπηρετεί τις αιτήσεις που κάνουν οι Εικονικές Μηχανές προς τη Φυσική Υποδομή για τη χρήση συγκεκριμένων υπολογιστικών πόρων. Ο Επιμελητής Εικονικών Μηχανών είναι απαραίτητος στο βαθμό που το μοντέλο του Υπολογιστικού Νέφους απαιτεί την ύπαρξη ενός λογισμικού το οποίο βρίσκεται κοντά στην φυσική υποδομή με στόχο την υποστήριξη διαφορετικών μεταξύ τους λειτουργικών συστημάτων.

Τα λειτουργικά αυτά συστήματα συναντώνται στη βιβλιογρφία ως Φιλοξενούμενα Λειτουργικά Συστήματα (Guest Operating Systems). Συνεπώς θα μπορούσαμε να κάνουμε τον παραλληλισμό ότι αυτά τα Φιλοξενούμενα Λειτουργικά Συστήματα εκτελούνται ως εφαρμογές με συγκεκριμένες απαιτήσεις πάνω στο Λειτουργικό Σύστημα «Οικοδεσπότη» που είναι ο Επιμελητής των Εικονικών Μηχανών.

Κωδικοποιώντας τα τρία στάδια του 'Σχήματος 2' το οποίο

ουσιαστικά αποτελεί το μοντέλο με βάση το οποίο γίνονται οι διεργασίες σε μια Υποδομή Νέφους:

- Οι Εικονικές Μηχανές αποτελούν ένα σύνολο από εικονικούς πόρους(εικονική επεξεργαστική ισχύ, εικονική RAM, εικονικός αποθηκευτικός χώρος κ.α.), ένα Φιλοξενούμενο Λειτουργικό Σύστημα και τις εφαρμογές που εκτελούνται πάνω σε αυτό το Φιλοξενούμενο Λειτουργικό Σύστημα.

- Ο Επιμελητής Εικονικών Μηχανών ο οποίος διανέμει του πόρους στις Εικονικές Μηχανές ανάλογα με τις απαιτήσεις τους.

- Η υποδομή η οποία αποτελεί το σύνολο των Φυσικών Πόρων που διαθέτει η υποδομή μας.

Το εργαστήριό μας, το Intellicloud, κάνει χρήση της ελεύθερης και ανοιχτού κώδικα πλατφόρμας για την ανάπτυξη υποδομών Νέφους Openstack. Το Openstack[20] αποτελεί το λογισμικό το οποίο διαθέτει υπηρεσίες Νέφους της δομής «Υποδομή ως Υπηρεσία» IaaS και κομμάτι του οποίου αποτελεί ο Επιμελητής Εικονικών Μηχανών τύπου KVM Hypervisor[21][14].

2.8 Γιατί να επιλέξει κάποιος το Υπολογιστικό Νέφος;

Πέραν των τεχνικών ζητημάτων και των βασικών λειτουργικών χαρακτηριστικών του Υπολογιστικού Νέφους, χρειάζεται να δούμε πως ο καταναλωτής αυτού του νέου μοντέλου εξυπηρετείται και βοηθιέται ώστε να λύσει προβλήματα και μειονεκτήματα που αντιμετόπιζε με τη χρήση παλαιότερων υποδομών.

- **Αυτοεξυπηρέτηση στην τροφοδοσία των πόρων**

Επιτρέπει στους χρήστες να χρησιμοποιήσουν τα δικά τους σύνολα από υπολογιστικούς πόρους σύμφωνα με τις ανάγκες τους, χωρίς τις καθυστερήσεις και τις επιπλοκές που έχει συνήθως η αγορά νέων υποδομών. Η τεχνολογία αυτή η ανάπτυξη και τροποποίησή της επιτρέπει τη βελτίωση της εμπειρίας του τελικού χρήστη του Νέφους ενώ ταυτόχρονα μπορεί να γίνεται επίβλεψη, διαχείριση ή και επέκταση της υποκείμενης υποδομής.

- **Επεκτασιμότητα**

Σπάει τη κλειστή σχέση που υπήρχε ανάμεσα στις κυμαινόμενες ανάγκες μεμονωμένων χρηστών και τους τυπικούς περιορισμούς υποδομών και αυτό το κάνει επειδή μπορεί να καλύψει άμεσα ανάγκες για αύξηση υπολογιστικών πόρων αλλά και να τους απελευθερώσει όταν αυτοί δεν είναι αναγκαίοι.

[20] http://www.openstack.org/
[21] http://whatis.techtarget.com/definition/KVM-hypervisor

- **Αξιοπιστία και ανοχή στα σφάλματα**

Είναι εύκολο να δοθεί έμφαση σε συγκεκριμένα κομμάτια της υποδομής ώστε αυτά να βελτιωθούν έως ότου να καλύπτουν τις ανάγκες σε αξιοπιστία. Επίσης αυτά τα επίπεδα αξιοπιστίας μπορούν να τροποποιηθούν αλλά και να καλυφθούν χωρίς καμία μεσολάβηση του τελικού χρήστη.

- **Βελτιστοποίηση και Ενοποίηση**

Μεγιστοποιεί τη χρήση και αυξάνει την αποτελεσματικότητα της υπάρχουσας υποδομής και των πόρων της. Επεκτείνει τον κύκλο ζωής της υποδομής και μειώνει τα έξοδα κεφαλαίου για την ανανέωση της.

- **Ποιότητα προσφερόμενων υπηρεσιών**

Επιτρέπει στους διαχειριστές να επανεκτιμούν δυναμικά το επίπεδο προσφοράς παρεχόμενων υπηρεσιών και πόρων (service level agreement - SLA) που σχετίζεται με τους χρήστες και τις ομάδες αυτών για τους υπάρχοντες πόρους. Επιτρέπει στον εκάστοτε οργανισμό να αντιδρά γρήγορα σε κυμαινόμενες συνθήκες χωρίς να παρέχει στους χρήστες περιττές πληροφορίες και ανάμειξη στο ζήτημα παροχής πόρων.

- **Καλά προσδιοριμένη διεπαφή API**

Η χρήση ενός καλώς προσδιορισμένου και σταθερού API "απαγκιστρώνει" τους πελάτες από συγκεκριμένους προμηθευτές και βελτιώνει τη διαλειτουργικότητα (interoperability) με έναν ολοένα και αναπτυσσόμενο αριθμό από εργαλεία, αλλά και παρόχους υπηρεσιών Νέφους.

- **Διαθεσιμότητα ανάλογα με τις ανάγκες**

Ευθυγραμμίζει τη σπατάλη πόρων με την πραγματική χρήση τους επιτρέποντας έτσι στους εκάστοτε οργανισμούς να πληρώνουν μόνο για τους πόρους τους οποίους πραγματικά χρειάζονται, και όταν τους χρειάζονται.

2.9 Υπηρεσιο-κεντρική Αρχιτεκτονική (Service Oriented Architecture - SOA) και Υπολογιστικό Νέφος

2.9.1 Η λογική της Υπηρεσιο-κεντρικής Αρχιτεκτονικής

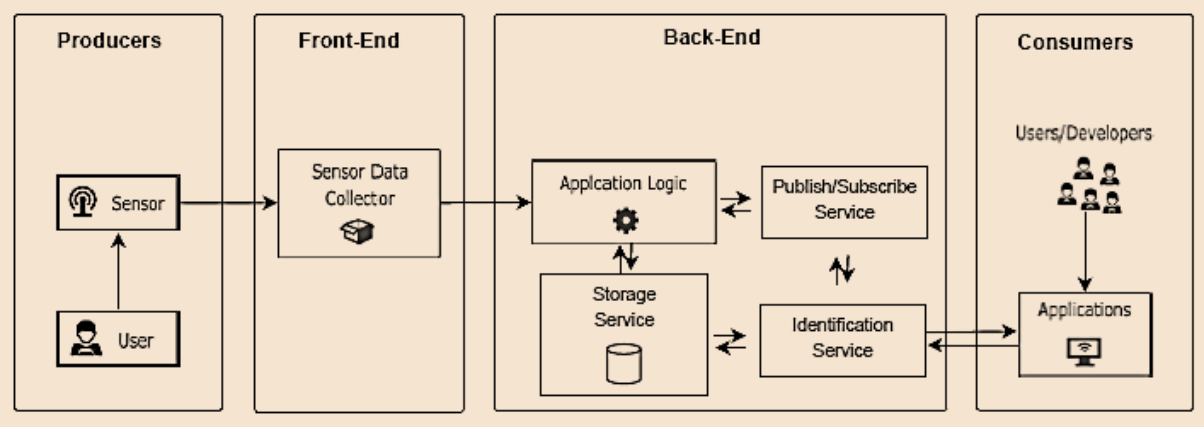
Το σύστημα το οποίο θα αναπτύξουμε θα βασίζεται στην αρχιτεκτονική ανάπτυξης εφαρμογών "Service Oriented Architecture" ή αλλιώς "υπηρεσιο-κεντρικής" αρχιτεκτονικής. Η υπηρεσιο-κεντρική αρχιτεκτονική ως αφηρημένη έννοια βασίζεται στη λογική ότι οποιοδήποτε μεγάλο πρόβλημα μπορεί να επιμεριστεί και να διαχειριστεί καλύτερα εάν το σπάσουμε σε μικρότερα προβλήματα τα οποία το συνθέτουν. Κατά συνέπεια το

ίδιο ισχύει και για συστήματα τα οποία βασίζονται σε υπηρεσίες ιστού(web services) και τα οποία ακολουθούν αρχιτεκτονική προσανατολισμένη στην ανάπτυξη τύπου SoA.

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[15].

2.9.2 Η σχέση της Υπηρεσιο-κεντρικής Αρχιτεκτονικής με το Υπολογιστικό Νέφος

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 λογισμικού που υλοποιούν βασικές λειτουργίες, γνωστά και ως Generic Enablers, τα οποία είναι προσβάσιμα από οποιοδήποτε υπολογιστικό σύστημα αρκεί να ακολουθείται ένα συγκεκριμένο πρωτόκολλο επικοινωνίας.



Πρότυπη απεικόνιση της Υπηρεσιο-κεντρικής Αρχιτεκτονικής
Σχήμα 3

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 Συστήματος, Διεπαφή Χρηστών και Καταναλωτές) εμφανίζεται στο 'Σχήμα 3'.

2.9.3 Τα βασικά πλεονεκτήματα της Υπηρεσιο-κεντρικής Αρχιτεκτονικής

- Μια διαδικτυακή υπηρεσία είναι ανεξάρτητη τόσο από το λειτουργικό σύστημα όσο και από τη σχετική φυσική υπολογιστική υποδομή.

- Σε ένα υπάρχον λογισμικό σύστημα η ένταξη και η ενσωμάτωση μιας νέας υπηρεσίας δεν απαιτεί αλλαγές στον μηχανισμό λειτουργίας του συστήματος.

- Μεγάλη διαθεσιμότητα και δωρεάν προσφορά τέτοιων υπηρεσιών η χρήση των οποίων μπορεί να είναι άμεση.

- Οι υπηρεσίες αυτές δίνουν τη δυνατότητα διόρθωσής τους και επέκτασής τους από τους κατασκευαστές χωρίς ο καταναλωτής-χρήστης της υπηρεσίας να αντιλαμβάνεται κάποια αλλαγή στο σύστημά του.

- Η παραμετροποίηση αυτών των υπηρεσιών δεν δεσμεύεται αυστηρά στα πλαίσια του κατασκευαστή, αλλά διατίθενται συχνά στην κοινότητα προς βελτίωση.

- Οι υπηρεσίες αυτές ακολουθούν ένα συγκεκριμένο μοντέλο επικοινωνίας, συνεπώς ο χρήστης-καταναλωτής δεν έχει παρά να αποστέλλει μέσω του συστήματός του δεδομένα και να δέχεται απαντήσεις από την επεξεργασία αυτών των δεδομένων.

- Οι υπηρεσίες αυτές δεν είναι απαραίτητες-δεσμευτικές για το σύστημα στο οποίο έχουν ενταχθεί. Μπορούν ανά πάσα ώρα και στιγμή να αντικατασταθούν από άλλες που προσφέρουν με καλύτερους όρους την ίδια εξυπηρέτηση.

- Οι πόροι που καταναλώνονται για την υπηρεσία αυτή(RAM, CPU κλπ) είναι πόροι που διατίθενται από τον πάροχο της υπηρεσίας, συνεπώς νέες υπηρεσίες δεν επιβαρύνουν σε καμία περίπτωση τις απαιτήσεις του συστήματός μας[16].

2.10 REST-API
(Representational State Transfer - Application Programming Interface)

2.10.1 Η αρχιτεκτονική του REST

Οι υπηρεσίες διαδικτύου που αναφέραμε παραπάνω και φυσικά οι GEs χρησιμοποιούν αρχιτεκτονική επικοινωνίας REST. Το REST είναι είδος αρχιτεκτονικής που αποτελείται από ένα σύνολο συγκεκριμένων κανόνων και περιορισμών που εφαρμόζονται στις διαδικτυακές υπηρεσίες, τόσο στο είδος των δεδομένων όσο και στο είδος των συνδέσεων. Η λογική του REST αγνοεί το περιεχόμενο της υπηρεσίας και τον τρόπο με τον οποίο αναπτύχθηκε. Επικεντρώνεται στην αλληλεπίδραση μεταξύ των υπηρεσιών καθώς και στην ερμηνεία των δεδομένων που ανταλλάσσονται ώστε να λειτουργούν συντονισμένα μέσα σε ένα αναπτυγμένο υπολογιστικό σύστημα[17].

Η αρχιτεκτονική για την υλοποίηση ενός REST-API ορίζει περιορισμούς σε τρία βασικά σύνολα. Στην βιβλιογραφία αναφέρονται ως Συστατικά (Components), Μηχανισμοί Σύνδεσης (Connectors) και Δεδομένα (Data). Δηλαδή στις υπηρεσίες και τους GEs, στις συνδέσεις και στα δεδομένα.

- Ως **Συστατικό** ορίζεται εκείνη η μονάδα λογισμικού που παρέχει επεξεργασία δεδομένων μέσω της διεπαφής της. Για παράδειγμα ένας GE.

- Ως **Μηχανισμός Σύνδεσης** ορίζεται ο μηχανισμός που επιτρέπει την επικοινωνία και τη συνεργασία μεταξύ των Συστατικών ώστε να ανταλλάσσουν δεδομένα και πληροφορίες.

- Ως **Δεδομένα** ορίζονται εκείνα τα στοιχεία πληροφορίας που μεταφέρονται ή λαμβάνονται από ένα Συστατικό μέσω του Μηχανισμού Σύνδεσης[18].

Συνεπώς η αρχιτεκτονική REST έρχεται να εξυπηρετήσει τον καταναλωτή της υπηρεσίας για το λόγο ότι μπορεί να αλληλεπιδρά αποτελεσματικά με τον προμηθευτή χωρίς να γνωρίζει και να τον ενδιαφέρει ο διακομιστής και οι πόροι που φιλοξενούνται εκεί. Ο μοναδικός περιορισμός αυτής της αλληλεπίδρασης είναι ότι η μεταξύ τους διαδικτυακή επικοινωνία πρέπει να γίνεται αυστηρά στα πλαίσια του πρωτοκόλλου HTTP.

2.10.2 Συνοπτικά ο τρόπος λειτουργίας του REST-API και του πρωτοκόλλου HTTP

- Το REST-API αναπαριστά τον τρόπο επικοινωνίας και μεταφοράς δεδομένων μεταξύ των υπηρεσιών στο οποίο στηρίζεται η επικοινωνία μέσα στο διαδίκτυο.

- Το REST από την πλευρά του είναι βασισμένο πάνω στο πρωτόκολλο HTTP, ένα πρωτόκολλο που μας δίνει τη δυνατότητα να στέλνουμε και να λαμβάνουμε πληροφορίες σε κείμενο(documents).

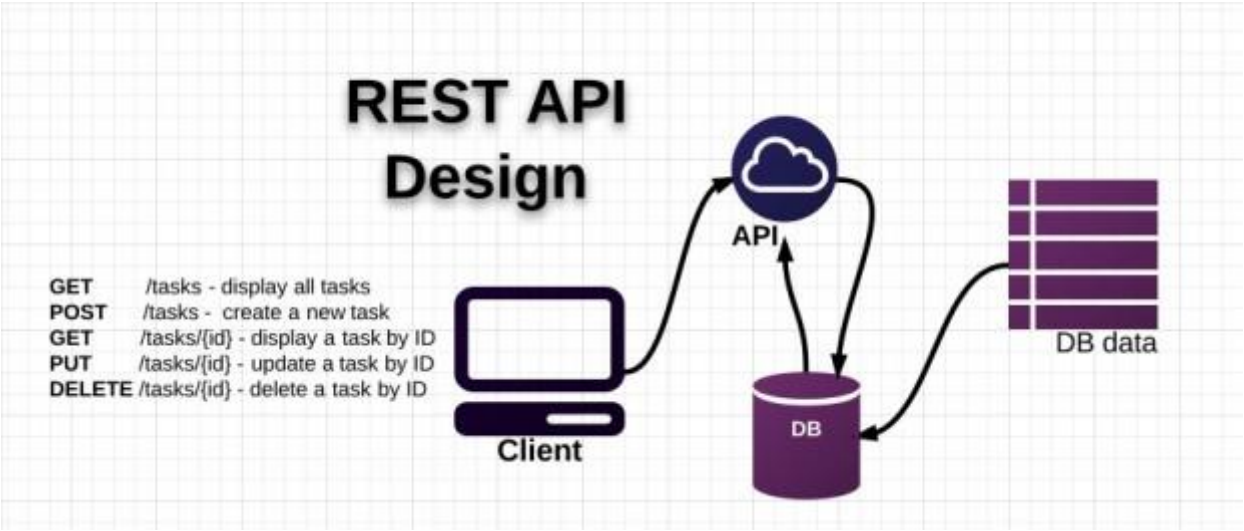
- Ο πελάτης κάνει την αίτηση και ο εξυπηρετητής απαντάει. (Client: request - Server: response)

- Το πρωτόκολλο HTTP είναι βασισμένο στο κείμενο και τα μηνύματα που μεταφέρει έχουν κυρίως σώμα και επικεφαλίδα. (Text based HTTP – Header and Body)

- Η επικεφαλίδα περιέχει πληροφορίες και οδηγίες για το μήνυμα που θέλουμε να στείλουμε όπως για παράδειγμα ο τρόπος κωδικοποίησης των δεδομένων. Το κυρίως σώμα έχει τα δεδομένα τα οποία θέλουμε να μεταφέρουμε μέσω του δικτύου σύμφωνα με τις οδηγίες πάντα της επικεφαλίδας.

- Κάθε αίτηση, γνωστή και ως API-CALL, ορίζει μια συγκεκριμένη μέθοδο HTTP μέσα στην επικεφαλίδα. Οι μέθοδοι ουσιαστικά λένε στον εξυπηρετητή το τι θα κάνει τα δεδομένα που παρέχονται. Οι δύο δημοφιλέστερες μέθοδοι είναι το GET και το POST, υπάρχουν επίσης η PUT και η DELETE.

- Οι αιτήσεις και οι απαντήσεις από το server κυρίως κωδικοποιούνται είτε σε JSON είτε σε XML. Τα δύο κυρίαρχα πρωτόκολλα δηλαδή κωδικοποίησης δεδομένων[19].



Παράδειγμα αρχιτεκτονικής REST-API
Σχήμα 4

Το 'Σχήμα 4' αναπαριστά ένα παράδειγμα επικοινωνίας ενός Πελάτη με τον Πάροχο μιας υπηρεσίας μέσω της αρχιτεκτονικής REST-API. Όπως περιγράψαμε παραπάνω, ο Πελάτης κάνει αιτήσεις χρησιμοποιώντας μεθόδους βασισμένες στο πρωτόκολλο HTTP και σύμφωνα με το API το οποίο διαθέτει ο πάροχος, ώστε να ενημερώσει τα δεδομένα μιας βάσης δεδομένων ή αντίστοιχα να λάβει από αυτή πληροφορίες.

2.11 Προσφερόμενες υπηρεσίες Γενικού Σκοπού(Generic Enablers) και Ειδικού Σκοπού(Specific Enablers)

Ένας **Generic Enabler** αποτελεί ένα σύνολο από υπηρεσίες κοινού σκοπού που είναι σχεδιασμένο στα πλαίσια της αρχιτεκτονικής REST και συνοδεύεται από ένα API το οποίο αποτελεί τη διεπαφή επικοινωνίας. Οι αιτήσεις σύμφωνα με τους κανόνες του API προς τον GE ονομάζονται εν γένει κλήσεις API (API-CALLS) [20].

Για παράδειγμα οι υπηρεσίες οι οποίες θα μπορούσε να φιλοξενεί ένας GE είναι τα εξής:
-Load object
-Get object
-Delete object
-Merge object

Το συγκεκριμένο παράδειγμα δείχνει ότι ο συγκεκριμένος GE προσφέρει τη δυνατότητα στο χρήστη να ανακαλέσει από τη βάση δεδομένων ένα αντικείμενο, να κάνει λήψη αυτού του αντικειμένου μέσω διαδικτύου ο χρήστης, να το σβήσει από τη βάση δεδομένων ή να το συγχωνεύσει με κάποιο άλλο αντικείμενο.

Ένας **Specific Enabler** χρησιμοποιεί υπηρεσίες από έναν ή περισσότερους GEs προκειμένου να υλοποιήσει μια πιο σύνθετη λειτουργία[20].

Για παράδειγμα ένας SE βασισμένος στο API του προηγούμενου GE που αναφέραμε μπορεί να είναι ο εξής:
-Load object (object 1)
-Load object (object 2)
-Merge objects (object 1+ object 2)
-Export merged objects

Στο συγκεκριμένο παράδειγμα ο SE προσφέρει τη δυνατότητα φόρτωσης δύο αντικειμένων από τη βάση δεδομένων, συγχόνευσή τους και εξαγωγή τους ως ένα ενιαίο αντικείμενο.

2.12 Το περιβάλλον του FI-WARE

Το FI-WARE αποτελεί μια υποδομή ανάπτυξης και παροχής υπηρεσιών Νέφους βασισμένη στη λογική του Διαδικτύου του Μέλλοντος[22] (Future Internet). Η υποδομή αυτή έχει εξελιχθεί σε μια κοινότητα στα πλαίσια της οποίας αναπτύσσονται και βελτιώνονται GEs και SEs οι οποίοι διατίθενται προς χρήση δωρεάν για την υλοποίηση μεγαλύτερων υπολογιστικών συστημάτων Νέφους.

Στο σύστημα το οποίο θέλουμε να υλοποιήσουμε, θα κάνουμε χρήση δύο υπηρεσιών-GEs οι οποίες παρέχονται από το παραπάνω περιβάλλον Νέφους του FI-WARE.

Το FI-WARE διαθέτει ένα κατάλογο από υπηρεσίες GEs με "ανοιχτά"-ελεύθερα προς χρήση APIs που επιτρέπουν στους προγραμματιστές να αποδεσμεύονται σε σημαντικό βαθμό από εταιρίες-προμηθευτές εμπορεύσιμου λογισμικού. Θέτει έτσι ένα ισχυρό θεμέλιο για το μέλλον του διαδικτύου, καλλιεργώντας ένα ιδανικό και βιώσιμο περιβάλλον για τους παρόχους υπηρεσιών και τους τελικούς χρήστες.[21]

Το μεγάλο προτέρημα του FI-WARE είναι όπως προαναφέραμε ο δωρεάν προς αξιοποίηση κατάλογός τους. Σε αυτό τον κατάλογο μπορεί ο οποιοσδήποτε κατασκευαστής να προσφέρει ένα νέο GE τον οποίο η κοινότητα έχει το δικαίωμα να τον βελτιώσει. Επίσης μπορεί ο οποιοσδήποτε προγραμματιστής να αξιολογήσει τους GEs που προσφέρονται και να επιλέξει τα κατάλληλα APIs που τον εξυπηρετούν για να χρησιμοποιήσει στο σύστημά του.

[22] http://en.wikipedia.org/wiki/Future_Internet

Οι δημοφιλέστεροι ίσως Generic Enablers που διατίθενται μέσω του καταλόγου του FI-WARE και κάποιοι από τους οποίους θα χρησιμοποιηθούν στο σύστημά μας είναι οι εξής:
Complex Event Processing(CEP)
Identity Management-Keyrock
Publish/Subscribe Context Broker-Orion[22]

3. CLIMATE
Cloud Intelligent Monitoring Atmosphere Technology – Παρακολούθηση ατμοσφαιρικών συνθηκών με χρήση αισθητήρων στο Νέφος

Σε αυτό το κεφάλαιο θα γίνει αναλυτική περιγραφή του CLIMATE, του συστήματος που έχουμε υλοποιήσει εστιάζοντας κυρίως στην αρχιτεκτονική πάνω στην οποία είναι βασισμένο, στις λειτουργίες που παρέχει και στη χρησιμότητά του. Το κεφάλαιο παρέχει μια πλήρη εικόνα του τρόπου με τον οποίο λειτουργεί το CLIMATE και πως διαχειρίζεται τις πληροφορίες που παρέχει. Η υλοποίηση του συστήματος περιγράφεται αναλυτικά στο κεφάλαιο 4.

3.1 Η λειτουργικότητα του CLIMATE

Το CLIMATE αποτελεί ένα σύστημα το οποίο δίνει τη δυνατότητα στους χρήστες που έχουν πρόσβαση σε αυτό να παρακολουθούν την ποιότητα του αέρα στον περιβάλλοντα χώρο σε μια σειρά από χώρους-δωμάτια μέσω μιας εφαρμογής στο διαδίκτυο. Η συγκεκριμένη εφαρμογή στοχεύει στο να μην χρειάζεται ο χρήστης να κάνει εγκατάσταση κάποιου λογισμικού προκειμένου να έχει πρόσβαση στην υπηρεσία που του προσφέρουμε, αλλά μέσω του περιηγητή (web browser) να μπορεί να έχει πρόσβαση στο σύνολο των εργαλείων που έχουμε αναπτύξει.

Οι χρήστες λοιπόν που θέλουν να κάνουν χρήστη του CLIMATE αρχικά καλούνται να δημιουργήσουν έναν προσωπικό λογαριασμό. Αυτό γιατί αφενός θέλουμε να υπάρχει μηχανισμός πιστοποίησης χρηστών και έγκρισής τους για τη χρήση της διαδικτυακής εφαρμογής (web application), αφετέρου θέλουμε το σύστημα να αποθηκεύει τις ρυθμίσεις του κάθε χρήστη, τις επιλογές και τις παραμετροποιήσεις που έχει κάνει.

Η εφαρμογή δίνει τη δυνατότητα στους χρήστες να επιλέγουν αυτοί ποιους χώρους θέλουν να παρακολουθούν καθώς επίσης και για ποια χαρακτηριστικά του κάθε χώρου θέλουν να παίρνουν πληροφορίες (μετρήσεις). Η επιλογή αυτή γίνεται μέσω της προσωπικής συνδρομής (subscription) του κάθε χρήστη. Η συνδρομή αυτή προφανώς διατηρείται αφού ο χρήστης κάνει Log Out και εισέλθει ξανά από οποιωνδήποτε υπολογιστή. Ο χρήστης φυσικά έχει την επιλογή της διαγραφής της συνδρομής του καθώς επίσης και της ανανέωσής της.

Ο χρήστης ο οποίος τελικά έχει κάνει συνδρομή και έχει ουσιαστικά επιλέξει χώρους και χαρακτηριστικά της ατμόσφαιρας, ενημερώνεται ζωντανά μόνο όταν υπάρξει αλλαγή σε κάποιο από τα χαρακτηριστικά τα οποία παρακολουθεί. Επίσης ενημερώνεται με ειδική σήμανση αν μια τιμή κάποιου χαρακτηριστικού ξεπεράσει

ένα κατώφλι το οποίο έχουμε ορίσει εμείς, ουσιαστικά για το αν μια τιμή βρίσκεται σε επικίνδυνα επίπεδα. Τέλος έχει τη δυνατότητα να παρακολουθήσει το ιστορικό των τελευταίων τιμών μέσω γραφημάτων για όποιο χαρακτηριστικό θέλει αυτός.

3.2 Τα χαρακτηριστικά που μπορεί ο χρήστης να παρακολουθεί

Τα χαρακτηριστικά τα οποία μπορεί ο χρήστης να παρακολουθεί εξαρτώνται από τον αριθμό και το μοντέλο των αισθητήρων. Οι αισθητήρες που χρησιμοποιούμε στο σύστημά μας για τη μέτρηση των χαρακτηριστικών του περιβάλλοντος αέρα είναι οι "Netatmo – Weather Station". Το σύστημά μας χρησιμοποιεί δύο διαφορετικούς αισθητήρες (modules), οι οποίοι ουσιαστικά μας χρησιμεύουν για να παίρνουμε ταυτόχρονα μετρήσεις από δύο διαφορετικά σημεία.

Ο πρώτος αισθητήρας (Module 1) μπορεί και λαμβάνει πληροφορίες για τη Θερμοκρασία, την Υγρασία, τη Βαρομετρική πίεση, το Διοξείδιο του άνθρακα και το Θόρυβο. Ο δεύτερος αισθητήρας (Module 2) λαμβάνει πληροφορίες μόνο για τη Θερμοκρασία και την Υγρασία.

Και οι δύο αισθητήρες, έχουν οριστεί από τον κατασκευαστή να λαμβάνουν μετρήσεις ανά 300 δευτερόλεπτα-5 λεπτά και τα τεχνικά τους χαρακτηριστικά είναι τα εξής:

1ος Αισθητήρας

Χαρακτηριστικό	Εμβέλεια	Ακρίβεια
Θερμοκρασία	0°C έως 50°C	+- 0.3°C
Υγρασία	0 έως 100%	+- 3%
Βαρομετρική Πίεση	260 έως 1160 mbar	+-1 mbar
Διοξείδιο του Άνθρακα(CO2)	0 to 5000 ppm	+- 50 ppm
Θόρυβος	35 dB to 120 dB	+- 0 dB

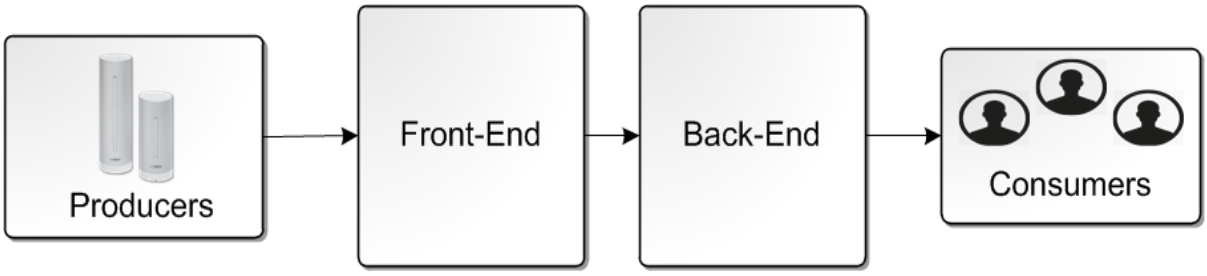
2ος Αισθητήρας

Χαρακτηριστικό	Εμβέλεια	Ακρίβεια
Θερμοκρασία	-40°C έως 65°C	+- 0.3°C
Υγρασία	0 έως 100%	+- 3%

3.3 Η αρχιτεκτονική του CLIMATE

Η υλοποίηση του συστήματός μας βασίζεται πάνω στη λογική της Υπηρεσιο-κεντρικής Αρχιτεκτονικής (Service Oriented Architecture - SoA). 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.
Η παραπάνω ιδέα της Υπηρεσιο-κεντρικής Αρχιτεκτονικής εξειδικεύεται στην υλοποίηση του συστήματός μας με την παρακάτω σχεδίαση του 'Σχήματος 5' η οποία συγκεντρώνει όλες τις υπηρεσίες του CLIMATE σε 4 βασικά τμήματα:[24]

- Παραγωγοί (Producers)
- Διεπαφή Συστήματος (Front-End)
- Διεπαφή Χρηστών (Back-End)
- Καταναλωτές (Consumers)



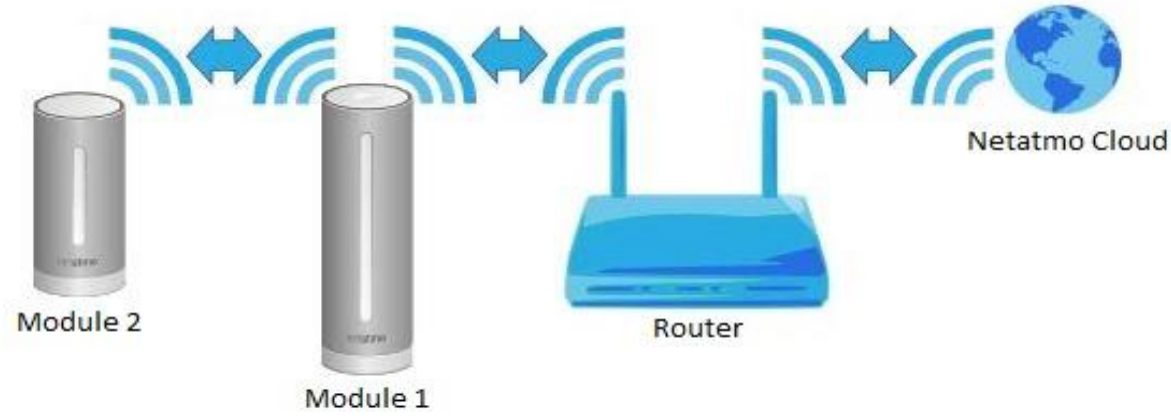
Γενική αρχιτεκτονική συστήματος
Σχήμα 5

3.3.1 Παραγωγοί και Καταναλωτές (Producers & Consumers)

Πρόσβαση στο σύστημά και δικαίωμα να το αλλάζουν (πχ. να προσθέτουν νέους αισθητήρες) έχουν μόνο οι παραγωγοί και οι καταναλωτές πληροφορίας. Οι παραγωγοί πληροφορίας είναι εκείνοι οι οποίοι θέτουν νέους αισθητήρες στο σύστημά μας ορίζοντας ουσιαστικά νέους χώρους τους οποίους μπορούμε να μελετήσουμε και να παράξουμε πληροφορία. Οι καταναλωτές, είναι ουσιαστικά οι χρήστες του CLIMATE στους οποίους απευθυνόμαστε. Είναι όσοι τους ενδιαφέρει να λάβουν τις πληροφορίες που παράγουν οι αισθητήρες και το σύστημά μας[25].

Παραγωγοί
Σε αυτό το στάδιο λειτουργίας γίνονται οι μετρήσεις της ποιότητας του περιβάλλοντος αέρα από τους αισθητήρες "Netatmo – Urban Weather Station". Οι δύο αυτοί αισθητήρες επικοινωνούν μεταξύ τους ασύρματα προκειμένου ο 2ος αισθητήρας να στέλνει δεδομένα που λαμβάνει στο 1ο αισθητήρα και αυτός με τη σειρά του μέσω διαδικτύου να τα στέλνει στη βάση δεδομένων του Netatmo όπου και αποθηκεύονται. Ο 1ος αισθητήρας συνδέεται ασύρματα στο τοπικό δίκτυο που του ορίζουμε εμείς. Το 'Σχήμα

6’ περιγράφει τον τρόπο επικοινωνίας με βάση τον οποίο υλοποιείται το 1° τμήμα της αρχιτεκτονικής μας, αυτό των Παραγωγών (Producers).



Διασύνδεση αισθητήρων – Τμήμα Παραγωγών
Σχήμα 6

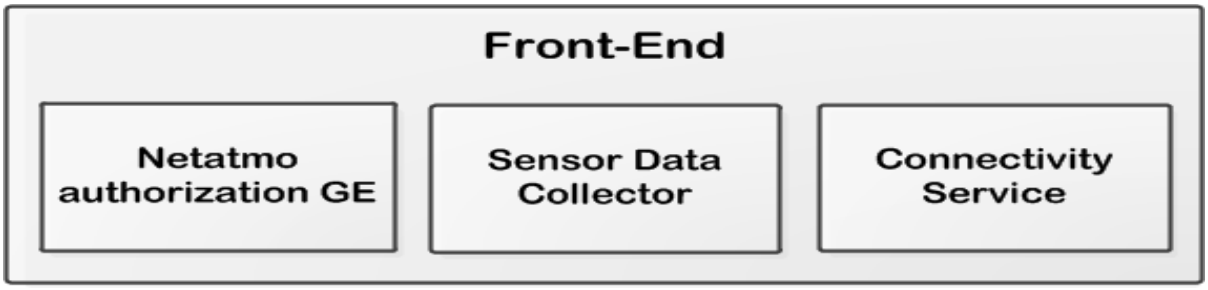
Καταναλωτές (Consumers)
Σε αυτό το τμήμα βρίσκονται οι χρήστες οι οποίοι θέλουν να έχουν πρόσβαση στις πληροφορίες που παράγει το σύστημά μας, καθώς επίσης και η εφαρμογή διεπαφής μέσω της οποίας έχει πρόσβαση ο χρήστης όπως φαίνεται στο ‘Σχήμα 7’. Σε αυτό το κομμάτι υλοποίησης αναπτύσσεται και το γραφικό περιβάλλον (graphical user interface – GUI) το οποίο διευκολύνει την πρόσβαση του χρήστη στις παρεχόμενες πληροφορίες και υπηρεσίες, διαθέτοντας του ένα σύνολο από εργαλεία.

Οι παρεχόμενες υπηρεσίες όπως έχουμε ήδη αναφέρει, είναι η δυνατότητα παρακολούθησης των κυμαινόμενων τιμών των χώρων και των χαρακτηριστικών που έχει ο χρήστης επιλέξει κάνοντας συνδρομή καθώς επίσης και η πρόσβαση σε ιστορικό μετρήσεων μέσω γραφημάτων.

Χρήστες και Διεπαφή
Σχήμα 7

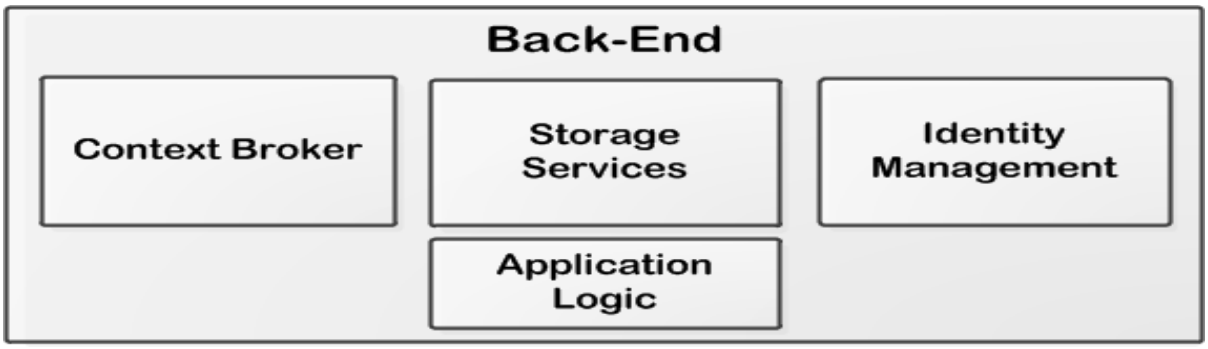
3.3.2 Διεπαφή Συστήματος & Διεπαφή Χρηστών (Front-End & Back-End)

Διεπαφή Συστήματος (Front-End)
Στη **Διεπαφή Συστήματος** βρίσκεται ο μηχανισμός συλλογής δεδομένων από το Νέφος του Netatmo. Οι υπηρεσίες που χρησιμοποιούμε όπως βλέπουμε και από το 'Σχήμα 8' είναι ο GE ταυτοποίησης που χρησιμοποιεί το Νέφος του Netatmo, ένας μηχανισμός με τον οποίο λαμβάνουμε δεδομένα από το Νέφος του Netatmo δηλαδή ο Sensor Data Collector και ο μηχανισμός ο οποίος στέλνει τα δεδομένα στις υπηρεσίες της Διεπαφής Χρηστών δηλαδή το Connectivity Service.



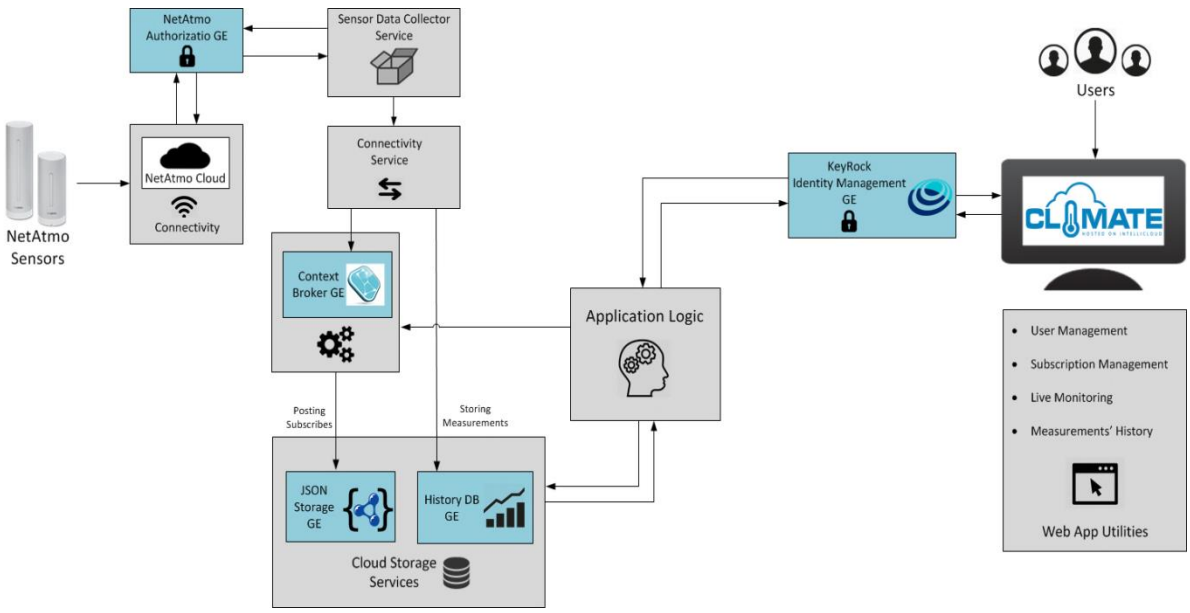
Υπηρεσίες στη Διεπαφή Συστήματος
Σχήμα 8

Διεπαφή Χρηστών (Back-End)
Στη Διεπαφή Χρηστών, όπως εμφανίζεται και στο 'Σχήμα 9', βρίσκεται ο μηχανισμός στον οποίο αποθηκεύουμε τα δεδομένα που λαμβάνουμε από τους αισθητήρες και από το Νέφος του Netatmo δηλαδή οι υπηρεσίες αποθήκευσης (Storage Services), ο μηχανισμός ταυτοποίησης χρηστών για την είσοδο στην εφαρμογή μας, δηλαδή ο Keyrock IDM GE, ο μηχανισμός για την συνδρομή, δηλαδή ο Context Broker GE καθώς επίσης και η Λογική Συστήματος του Climate.



Υπηρεσίες στη Διεπαφή χρηστών
Σχήμα 9

Το 'Σχήμα 10' δείχνει με αναλυτικό τρόπο την αρχιτεκτονική του συστήματος. Παραθέτει τους GEs που παρέχονται από το FI-WARE, τις υπηρεσίες που υλοποιήσαμε εμείς, καθώς επίσης και τη σχέση όλων αυτών μεταξύ τους.



Αρχιτεκτονική συστήματος CLIMATE
Σχήμα 10

Η βασική ιδέα της υλοποίησης είναι η εξής:

- Οι δύο αισθητήρες μας ανά 300 δευτερόλεπτα λαμβάνουν μετρήσεις από δύο διαφορετικούς χώρους και τις αποστέλλουν αυτόματα στη βάση δεδομένων του Νέφους του Netatmo.

- Ο μηχανισμός λήψης δεδομένων από το Νέφος του Netatmo, δηλαδή ο Sensor Data Collector ανά 20 δευτερόλεπτα μέσω κλήσεων API ενημερώνεται από το Netatmo Authorization GE για το αν υπάρχουν νέα δεδομένα από τους αισθητήρες μας. Αν υπάρχουν νέα δεδομένα τότε τα λαμβάνει υπό την μορφή JSON String.

- Στη συνέχεια ο μηχανισμός Connectivity Service παίρνει τα αποκωδικοποιημένα δεδομένα και τα προωθεί ώστε να ενημερώσει την οντότητα (Entity) του Context Broker GE και τη βάση δεδομένων που κρατάμε το ιστορικό των μετρήσεών μας, το Histroy DB GE. Η οντότητα αποτελεί ένα πληροφοριακό σχήμα της μορφής JSON String το οποίο διατηρεί πληροφορίες για τις μετρήσεις, τον τύπο των αισθητήρων, την ώρα λήψης των δεδομένων κ.α.

- Ο χρήστης μέσω του συστήματος διεπαφής της διαδικτυακής εφαρμογής που έχουμε αναπτύξει εγγράφεται στο σύστημά ή κάνει Log in αν έχει ήδη εγγραφεί.

- Η εγγραφή νέων χρηστών και η διαχείριση εισόδου των ήδη εγγεγραμμένων χρηστών στο σύστημά μας γίνεται από τον GE με όνομα Keyrock Identity Management GE. Η επικοινωνία με τον Identity Management GE για τη λήψη άδειας πρόσβασης στο

σύστημα για περιορισμένο χρονικό διάστημα (token), μοναδικού κωδικού που περιγράφει την ταυτότητα του κάθε χρήστη (User ID) κλπ. γίνεται μέσω κλήσεων API.

- Αφού αποκτήσει πρόσβαση στο σύστημά μας ένας χρήστης αυτομάτως γνωρίζουμε την ταυτότητά του. Είναι ο χαρακτηριστικός κωδικός για την αναγνώριση ενός χρήστη.

- Ο χρήστης πλέον μπορεί κάνει συνδρομή στον Context Broker GE ή να κάνει αίτηση να ανακαλέσει δεδομένα από το Histroy DB GE. Τέτοια δεδομένα για παράδειγμα είναι η Θερμοκρασία, η Υγρασία κ.α.

- Η συνδρομή του χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό συνδρομής (Subscripiton ID) και οι απαντήσεις που δίνει ο Context Broker GE με βάση την συνδρομή του χρήστη αποστέλλονται στον JSON Storage GE.

- Ο JSON Storage GE αποθηκεύει τα δεδομένα τα οποία εξάγει ο Context Broker GE. Τα δεδομένα αυτά αποτελούν απαντήσεις στις συνδρομές των χρηστών προς τον Context Broker GE και είναι οι μετρήσεις που έχει επιλέξει ο κάθε χρήστης να παρακολουθεί. Ο JSON Storage GE επιστρέφει ένα αναγνωριστικό αποθήκευσης (Storage ID) το οποίο αντιστοιχίζεται με το αναγνωριστικό συνδρομής (subscription ID). Και τα δύο αυτά αναγνωριστικά είναι μοναδικά για κάθε χρήστη.

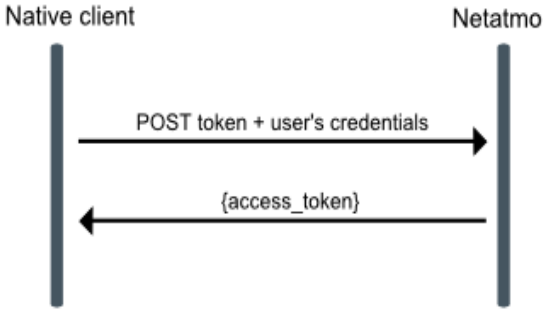
- Το τμήμα της αρχιτεκτονικής που ονομάζουμε Λογική Συστήματος (Application Logic) διαχειρίζεται τις πληροφορίες που χαρακτηρίζουν έναν χρήστη, δηλαδή το User ID, το Subcription ID και το Storage ID και με βάση αυτές τις πληροφορίες κάνει τις απαραίτητες κλήσεις API στους GEs και τις υπηρεσίες του συστήματός μας. Αναλυτικότερα, το τμήμα Λογική Συστήματος αναγνωρίζει την είσοδο ενός χρήστη στο σύστημα επικοινωνώντας με τον Keyrock Identity Management GE με το που εισάγει κάποιος χρήστης τα στοιχεία του. Μέσω αυτής της επικοινωνίας ενδιαφέρεται να λάβει μόνο το User ID του χρήστη. Έπειτα αναγνωρίζει το συγκεκριμένο User ID με ποιο Subsription ID και Storage ID αντιστοιχεί προκειμένου να δώσει τη δυνατότητα στο χρήστη να διαχειριστεί την συνδρομή του και να λάβει τα δεδομένα των μετρήσεων που επιθυμεί.

- Η παρακολούθηση των τιμών από το χρήστη γίνεται μέσω επικοινωνίας της Λογικής Συστήματος και του JSON Storage GE με κλήσεις API. Η Λογική Συστήματος γνωρίζει ποιο Storage ID αντιστοιχεί στο συγκεκριμένο χρήστη και συνεπώς καλεί τον JSON Storage GE να επιστρέψει τα αντίστοιχα δεδομένα.

3.4 Οι υπηρεσίες, οι μηχανισμοί και οι Generic Enablers

3.4.1 Netatmo authorization GE[23]

Την υπηρεσία την παρέχει η εταιρία Netatmo και μέσω αυτής μπορούν οι εξουσιοδοτημένοι (από την εταιρεία) χρήστες ή οι προγραμματιστές εφαρμογών να λαμβάνουν πληροφορίες και δεδομένα για τους αισθητήρες που τους ανήκουν. Συγκεκριμένα η αρχιτεκτονική του συγκεκριμένου GE χρησιμοποιεί το πρωτόκολλο ταυτοποίησης OAuth 2.0[24]. Συνεπώς ο προγραμματιστής εφαρμογών, όπως φαίνεται στο 'Σχήμα 11', χρειάζεται να κάνει αίτηση για τη λήψη μιας άδειας χρήσης (token) την 1η φορά που θα κάνει κλήση API στον GE και ανά 10800 δευτερόλεπτα αίτηση για την ανανέωση της άδειας. Με αυτό τον τρόπο παρέχεται συνεχής πρόσβαση στη βάση δεδομένων του Netatmo.



Πιστοποίηση χρηστών στο Νέφος του Netatmo.
Σχήμα 11

Το συγκεκριμένο GE παρέχει παράλληλα ένα RESTful API το οποίο υποστηρίζει μια σειρά από μεθόδους για την πρόσβαση στα δεδομένα μας. Συγκεκριμένα χρησιμοποιούμε τις μεθόδους:

GETUSER
Η συγκεκριμένη μέθοδος επιστρέφει πληροφορίες για ένα συγκεκριμένο χρήστη, όπως τα στοιχεία της εφαρμογής του, τη λίστα με τους αισθητήρες του κ.α.

DEVICELIST
Η μέθοδος αυτή επιστρέφει αναλυτικά τα στοιχεία των συσκευών ενός συγκεκριμένου χρήστη, τα μοντέλα και το είδος των αισθητήρων που χρησιμοποιεί, τα ID των συσκευών του κ.α.

GETMEASURE
Τέλος η συγκεκριμένη μέθοδος είναι αυτή η οποία χρησιμοποιείται περισσότερο στο σύστημά μας και ουσιαστικά επιστρέφει τις τελευταίες τιμές των μετρήσεων που έχουν λάβει

[23] https://dev.netatmo.com/doc/authentication
[24] http://oauth.net/2/

κάποιοι συγκεκριμένοι αισθητήρες.[26]

3.4.2 Μηχανισμός συλλογής δεδομένων από τους αισθητήρες (Sensor Data Collector)

Ο Sensor Data Collector είναι ένας μηχανισμός τον οποίο υλοποιήσαμε εμείς και ουσιαστικά είναι ο κώδικας ο οποίος επικοινωνεί με το RESTful API της εταιρίας όπως είδαμε πιο πάνω. Μέσα σε αυτό το μηχανισμό γίνεται η λήψη του token, η ανανέωσή του, η λήψη των δεδομένων του χρήστη καθώς και των συσκευών του και κυρίως η αίτηση για τη λήψη των μετρήσεων. Το token αποτελεί έναν μοναδικό κωδικό ο οποίος παρέχεται από το Netatmo authorization GE και ο οποίος είναι απαραίτητος να περιλαμβάνεται ως συνισταμένη μέσα στις μεθόδους με τις οποίες κάνουμε κλήσεις API. Αυτή η άδεια-κωδικός λήγει ανά τακτά χρονικά διαστήματα και είναι απαραίτητη η ανανέωσή του ή η έκδωση ενός νέου.

Ο Sensor Data Collector εκτελείται αυτόματα ανά 20 δευτερόλεπτα και ελέγχει ουσιαστικά για το αν υπάρχουν νέες τιμές από τους αισθητήρες προκειμένου να καλέσει τον επόμενο μηχανισμό ο οποίος είναι το Connectivity Service. Η λήψη όλων των δεδομένων από το Cloud του Netatmo είναι της μορφής JSON String.

Ο Sensor Data Collector έχει αναπτυχθεί πάνω σε Εικονική Μηχανή (VM) στην υποδομή Νέφους του εργαστηρίου μας.

3.4.3 Υπηρεσία Σύνδεσης (Connectivity Service

Ο μηχανισμός αυτός όπως είδαμε πιο πάνω είναι εκείνος ο οποίος ενημερώνει τις υπηρεσίες αποθήκευσης δεδομένων της υποδομής μας, τα Cloud Storage Services καθώς επίσης και τον Context Broker GE.

Αρχικά αποκωδικοποιεί το JSON String το οποίο έχει λάβει ο Sensor Data Collector με τις τελευταίες μετρήσεις των αισθητήρων και στη συνέχεια δημιουργεί τις αντίστοιχες μεταβλητές και το κατάλληλο JSON String για να ενημερώσει τον Context Broker GE και το History DB GE. Η επικοινωνία της συγκεκριμένης υπηρεσίας με τον Context Broker GE και me το History DB GE γίνεται μέσω κλήσεων API που θα περιγράψουμε έπειτα.
Το Connectivity Service έχει αναπτυχθεί πάνω σε Εικονική Μηχανή στην υποδομή Νέφους του εργαστηρίου μας.

3.4.4 Orion Context Broker GE[25]

Ο Orion Context Broker GE αποτελεί τη σημαντικότερη υπηρεσία μέσα στο σύστημά μας. Αποτελεί ίσως τον πιο ενημερωμένο GE της κατηγορίας υπηρεσιών Publish/Subscribe και διαθέτει διεπαφή επικοινωνίας για κλήσεις API NGSI10[26].

Ο συγκεκριμένος GE υποστηρίζει τις εξής λειτουργίες:

- Δημιουργία και ενημέρωση μιας οντότητας (Entity) της μορφής JSON String που διατηρεί τις πιο πρόσφατες τιμές των μετρήσεων που έχουν ληφθεί από τους αισθητήρες. Για παράδειγμα οι τιμές της θερμοκρασίας μέσα σε μια σειρά από δωμάτια.

- Άμεση και λειτουργική ανανέωση αυτών των τιμών. Για παράδειγμα ανανέωση της τιμής της θερμοκρασία για όλα τα δωμάτια για κάποια συγκεκριμένα μόνο.

- Συνδρομή για τη ενημέρωση των τιμών σε μια σειρά από χαρακτηριστικά. Για παράδειγμα συνδρομή για την παρακολούθηση της θερμοκρασίας στα δωμάτια 1 και 3.

- Ενημέρωση όταν υπάρξει αλλαγή σε κάποια από τις τιμές που μελετάμε ή ανά κάποιο συγκεκριμένο χρονικό διάστημα. Για παράδειγμα ενημέρωση του χρήστη όταν αλλάξει η θερμοκρασία είτε στο 1° είτε στο 3° δωμάτιο.

Ο Context Broker GE δέχεται αιτήσεις για δημιουργία μιας οντότητας (Entity), για ενημέρωση των τιμών των χαρακτηριστικών αυτής της οντότητας, για διόρθωση της οντότητας και για διαγραφή κάποιων χαρακτηριστικών μέσω API κλήσεων με τις μεθόδους πρωτοκόλλου HTTP POST και DELETE. Τα δεδομένα που δέχεται τα διατηρεί σε μια βάση δεδομένων τύπου Mongo Db[27].

Επίσης ο Context Broker GE δίνει απαντήσεις σε μια συγκεκριμένη διεύθυνση που του ορίζουμε εμείς, ανά τακτά χρονικά διαστήματα ή ανά μεταβολή κάποιας τιμής κάνοντας χρήστη της μεθόδου POST.

Ο Context Broker GE που χρησιμοποιούμε αποτελεί Εικονική Μηχανή η οποία τρέχει στην υποδομή Νέφους του εργαστηρίου μας[27].

---

[25] http://catalogue.fi-ware.org/enablers/publishsubscribe-context-broker-orion-context-broker
[26] https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI-10_Open_RESTful_API_Specification
[27] http://www.mongodb.org/

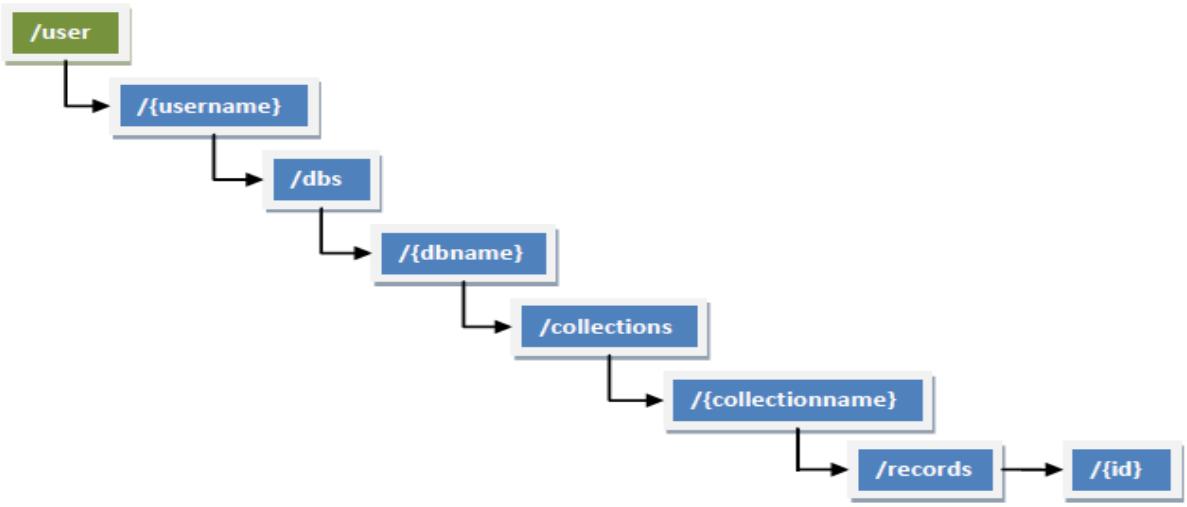
3.4.5 JSON Storage GE

Ο συγκεκριμένος GE αποτελεί υλοποίηση του εργαστηρίου μας, του Intellicloud και η Εικονική Μηχανή του συγκεκριμένου GE που χρησιμοποιούμε τρέχει στην υποδομή Νέφους του εργαστηρίου μας. Ο JSON Storage GE είναι ένα σύστημα το οποίο έχει ως σκοπό να αποθηκεύει JSON Strings κάνοντας χρήση της Mongo DB. Ο συγκεκριμένος GE κάνει χρήση αυτού του τύπου βάσης για το λόγο ότι η Mongo DB επιτρέπει την αποθήκευση «σχημάτων» χωρίς να ακολουθεί τη μεθοδολογία αποθήκευσης των παραδοσιακών βάσεων δεδομένων (πίνακες, κελιά, κλπ.) Η πρόσβαση και σε αυτό τον GE γίνεται μέσα από ένα RESTful API.

Η αρχιτεκτονική της συγκεκριμένης υπηρεσίας βασίζεται σε τέσσερα τμήματα:

- Χρήστες (Users)
- Βάσεις Δεδομένων (Databases)
- Συλλογές (Collections)
- Εγγραφές (Records)

Η σχεδίαση του JSON Storage GE βασίζεται στη λογική εγγραφής χρηστών και δέσμευσης προσωπικού χώρου και δεδομένων. Συγκεκριμένα οι χρήστες διαθέτουν προσωπικές βάσεις δεδομένων, οι οποίες βάσεις δεδομένων περιέχουν συλλογές και τα δεδομένα εντός των συλλογών ονομάζονται εγγραφές. Η μεταξύ τους σχέση περιγράφεται από το 'Σχήμα 12':



Η σχεδίαση του JSON Storage GE
Σχήμα 12

Ο JSON Storage GE επιτρέπει την είσοδο και την ταυτοποίηση χρηστών, τη δημιουργία βάσεων δεδομένων για τον κάθε χρήστη, συλλογών εντός των βάσεων δεδομένων οι οποίες συλλογές αποτελούνται από εγγραφές με μοναδικό ID. Επίσης επιτρέπει την ανανέωση και την διαγραφή των παραπάνω στοιχείων.

Στη δικιά μας υλοποίηση, έχουμε δημιουργήσει ένα χρήστη, μία βάση δεδομένων, μία συλλογή και μέσα εκεί δημιουργούμε και διαγράφουμε εγγραφές. Όπως είδαμε παραπάνω η κάθε εγγραφή αποτελεί ένα σχήμα της μορφής JSON δηλαδή τα δεδομένα που επιστρέφονται από τον Orion Context Broker GE ως απαντήσεις στους χρήστες[28].

3.4.6 Keyrock Identity Management (IDM) GE[28]

Ο συγκεκριμένος GE αποτελεί τη γέφυρα που συνδέει το επίπεδο εισόδου του χρήστη στο σύστημά μας με το επίπεδο χρήσης της εφαρμογής. Αναλαμβάνει πλήρως το μηχανισμό με τον οποίο εγγράφονται νέοι χρήστες και το μηχανισμό που διαχειρίζεται την είσοδο των ήδη εγγεγραμμένων χρηστών στο CLIMATE.

Η εικονική μηχανή την οποία χρησιμοποιούμε για το συγκεκριμένο GE είναι ένα μια κοινή-δημόσια εικονική μηχανή (Public Instance) η οποία τρέχει στην υποδομή Νέφους του FI-LAB[29]. Προκειμένου να το χρησιμοποιήσουμε όπως θα δούμε και πιο κάτω χρειάζεται να κάνουμε μια εσωτερική εφαρμογή σε αυτή τη δημόσια εικονική μηχανή προκειμένου να ρυθμίσουμε τον τρόπο με τον οποίο το θα επικοινωνεί με το σύστημά μας και θα διαχειρίζεται του χρήστες μας.

Η επικοινωνία επίσης με το συγκεκριμένο GE γίνεται μέσω ενός RESTful API. Ο κάθε χρήστης ο οποίος επιθυμεί να εισαχθεί στο σύστημά μας, συνδέεται μέσω του συγκεκριμένου GE ο οποίος του επιστρέφει ένα κωδικό άδειας χρήσης (token). Το token ανανεώνεται συχνά και δίνει τη δυνατότητα στο χρήστη να έχει μόνιμη πρόσβαση στα εργαλεία του CLIMATE. Το χαρακτηριστικό του κάθε χρήστη είναι το User ID το οποίο μας το επιστρέφει ο IDM GE και είναι απαραίτητο για τη λειτουργία της Λογικής Συστήματος[29].

3.4.7 History DB GE

Η συγκεκριμένη υπηρεσία είναι ένας Generic Enabler τον οποίο υλοποιήσαμε εμείς πάνω στην προσωπική μας Εικονική Μηχανή στο οποίο φιλοξενείται το CLIMATE. Η συγκεκριμένη υπηρεσία εξυπηρετεί την ανάγκη να κρατάμε το ιστορικό των μετρήσεων που λαμβάνουν οι αισθητήρες μας καθώς και τον ακριβή χρόνο που λήφθηκε η κάθε μέτρηση. Ο συγκεκριμένος GE επικοινωνεί και λαμβάνει κατευθείαν δεδομένα από το Sensor Data Collector χωρίς να μεσολαβεί ο Context Broker GE σε αντίθεση με τον JSON

[28] http://catalogue.fi-ware.org/enablers/identity-management-keyrock
[29] http://www.fi-ware.org/lab/

Storage GE. Τα δεδομένα τα οποία λαμβάνει ο συγκεκριμένος GE τα αποθηκεύει σε μια βάση δεδομένων τύπου My-SQL.

3.4.8 Η Λογική Συστήματος (Application Logic) του CLIMATE

Η Λογική Συστήματος αποτελεί το σύνολο εκείνων των διεργασιών που έχουμε αναπτύξει προκειμένου οι GEs και οι υπηρεσίες που έχουμε χρησιμοποιήσει να μπορούν να επικοινωνούν μεταξύ τους αποτελεσματικά. Ορίζει τον τρόπο με τον οποίο επικοινωνούν οι υπηρεσίες και οι GEs ώστε η πληροφορία που παράγεται από το τμήμα των Παραγωγών να αξιοποιείται και να φθάνει στο τμήμα των Καταναλωτών.

Τρία βασικά στοιχεία τα οποία είναι απαραίτητα προκειμένου η Λογική Συστήματος να αξιοποιεί την παραγόμενη πληροφορία και να την παρέχει στους χρήστες αποτελεσματικά:

- **User ID**
  Το ID που μας επιστρέφει ο IDM GE κατά την εισαγωγή ενός χρήστη στο σύστημά μας. Το User ID παράγεται αυτόματα από τον IDM GE κατά την εγγραφή ενός νέου χρήστη και είναι μοναδικό.

- **Subscription ID**
  Το ID που αντιστοιχεί σε μια νέα συνδρομή. Παράγεται από τον Context Broker GE αυτόματα κατά την αίτηση για μια νέα συνδρομή. Είναι επίσης μοναδικό.

- **Storage ID**
  Το ID που αντιστοιχεί για κάθε εγγραφή-σχήμα στον JSON Storage GE. Παράγεται αυτόματα κατά την αίτηση για την εισαγωγή νέων δεδομένων. Τα δεδομένα που αντιστοιχούν σε κάθε Storage ID αποτελούν απάντηση όπως έχουμε ήδη αναφέρει σε κάποια συνδρομή. Άρα υπάρχει στενή αντιστοιχία μεταξύ Subscription ID και Storage ID.

Οι λειτουργίες που υλοποιεί η Λογική Συστήματος:

- Διαβάζει το User ID κατά την είσοδο ενός χρήστη στο σύστημά μας.

- Ελέγχει αν ο χρήστης έχει δημιουργήσει την προσωπική του συνδρομή στον Context Broker GE.

- Αν όχι τον στέλνει στη φόρμα για τη δημιουργία της συνδρομής.

- Δημιουργεί το κατάλληλο JSON και με κλήση API κάνει αίτηση

στον Context Broker GE για τη νέα συνδρομή.

- Ο Context Broker GE επιστρέφει το Subscription ID και η Λογική Συστήματος δημιουργεί μια εγγραφή-σχήμα στο JSON Storage GE.

- Ο JSON Storage GE επιστρέφει το Storage ID.

- Πλέον η Λογική Συστήματος από τη στιγμή που έχει στη διάθεσή το User ID, το Subscripiton ID και το Storage ID στέλνει το χρήστη στη σελίδα που μπορεί να παρακολουθεί τα χαρακτηριστικά τα οποία τον ενδιαφέρουν.

- Αν έχει δημιουργήσει συνδρομή τον στέλνει κατευθείαν στη σελίδα για τη παρακολούθηση των χαρακτηριστικών τα οποία έχει δηλώσει ότι τον αφορούν.

- Η Λογική Συστήματος με την ίδια λογική διαχειρίζεται τη διαγραφή και την ανανέωση μιας συνδρομής.

- Διατηρεί Συνεδρία Χρήστη (User Seesion). Μεταβλητές που διατηρούν για μία συνεδρία τις προσωπικές επιλογές του χρήστη.

- Εάν ο χρήστης επιλέξει να αποχωρήσει από το σύστημα, τον στέλνει στον IDM GE που διαχειρίζεται την είσοδο και την έξοδο των χρηστών προκειμένου να κάνει Log Out.

Θα μπορούσαμε να πούμε ότι η Λογική Συστήματος (Application Logic) είναι ο πυρήνας της Διεπαφής Χρηστών (Back-End) 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.

3.5 Το γραφικό περιβάλλον του συστήματός (GUI) – Ενδεικτική χρήση του CLIMATE

Το γραφικό περιβάλλον της εφαρμογής μας έχει σχεδιαστεί έτσι ώστε να εξυπηρετεί με τον καταλληλότερο τρόπο τους χρήστες που επιλέγουν να το χρησιμοποιήσουν και συγκεκριμένα να δίνει τη δυνατότητα σε χρήστες που δεν έχουν οικειότητα με τη χρήση πολύπλοκων υπολογιστικών συστημάτων, να εξοικειώνονται με το CLIMATE όσο το δυνατόν γρηγορότερα.

Προκειμένου να δημιουργήσουμε ένα φιλικό και εξυπηρετικό προς

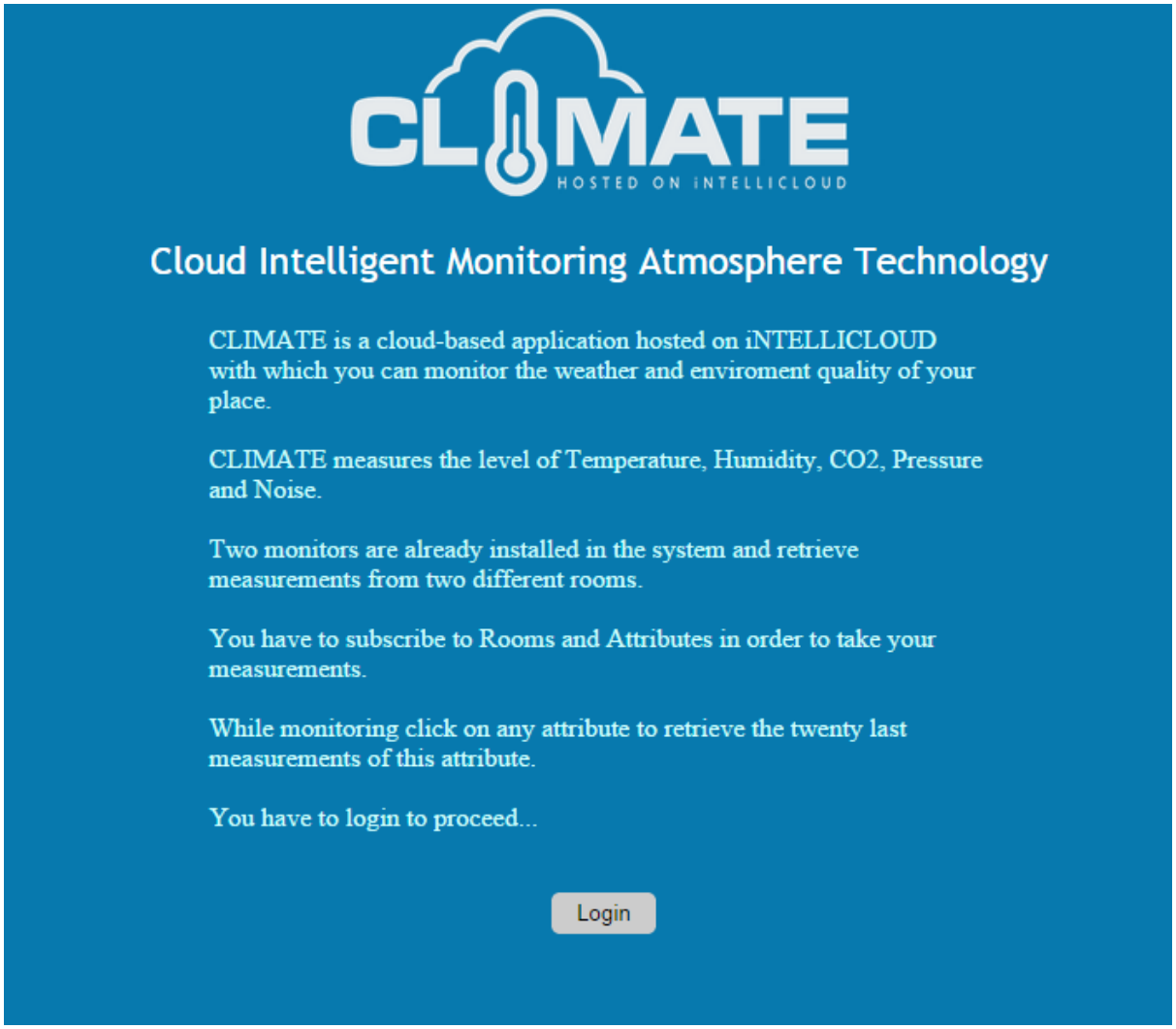
το χρήστη γραφικό περιβάλλον κάναμε χρήση των εργαλείων HTML[30] και CSS[31] καθώς επίσης σχεδιάσαμε και εικόνες και κουμπιά προκειμένου ο χρήστης να επιλέγει και να δηλώνει μέσω αυτών αισθητήρες και χαρακτηριστικά για να παρακολουθεί[30,31].

Συγκεκριμένα με την HTML δημιουργήσαμε τις φόρμες μέσω των οποίων ο χρήστης επιλέγει και καταχωρεί τις επιλογές του. Αντίστοιχα με τη CSS καταφέραμε να σχεδιάσουμε και να εισάγουμε τα αντίστοιχα χρώματα, εικόνες, στοιχεία προκειμένου το γραφικό περιβάλλον να είναι όσο το δυνατόν πιο εύχρηστο και καλαίσθητο.

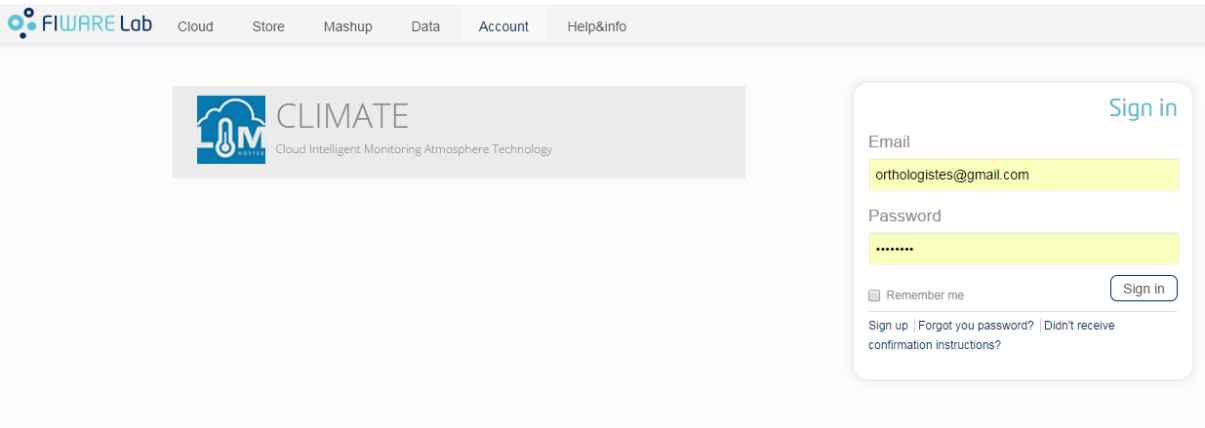
Σε αυτό το σημείο θα παραθέσουμε ένα παράδειγμα χρήσης. Το παράδειγμα ακολουθεί τα παρακάτω βήματα και λογική:

- Ο χρήστης μπαίνει στην αρχική σελίδα της εφαρμογής μας όπως φαίνεται στο 'Σχήμα 13' στην οποία υπάρχει περιγραφή του CLIMATE και του ζητείται να κάνει Login προκειμένου να προχωρήσει.
- Εισάγει τα στοιχεία του όπως φαίνεται στο 'Σχήμα 14'.
- Του εμφανίζεται η φόρμα του 'Σχήματος 15' για να επιλέξει τα δωμάτια στα οποία θέλει να κάνει συνδρομή. Συγκεκριμένα στο παράδειγμά μας επιλέγουμε και τα δύο δωμάτια που έχουμε στη διάθεσή μας και πατάμε «συνδρομή».
- Εν συνεχεία εμφανίζεται η φόρμα του 'Σχήματος 16' προκειμένου ο χρήστης να επιλέξει τα διαθέσιμα χαρακτηριστικά κάθε δωματίου. Στο παράδειγμά μας επιλέγουμε όλα τα χαρακτηριστικά και στα δύο δωμάτια και πατάμε «συνδρομή».
- Όπως βλέπουμε από το 'Σχήμα 17' το σύστημά μας ενημερώνει το χρήστη ότι προσπαθεί να λάβει τις μετρήσεις με βάση τις επιλογές που έχει κάνει.
- Έπειτα εμφανίζεται η οθόνη με τις λειτουργίες παρακολούθησης (monitoring) όπως περιγράφεται στο 'Σχήμα 18'. Εκεί ο χρήστης παρακολουθεί τις τιμές σε όλα τα χαρακτηριστικά που έχε επιλέξει και επίσης ενημερώνεται για τη στιγμή που έγινε η τελευταία μέτρηση. Παρατηρούμε ότι η θερμοκρασία στο 1° δωμάτιο είναι με κόκκινο χρώμα, αυτό σημαίνει ότι βρίσκεται σε σχετικά υψηλά επίπεδα. Επίσης παρατηρούμε τις επιλογές για διαγραφή της συνδρομής ή ανανέωσής του.
- Στη συνέχεια ο χρήστης επιλέγει το χαρακτηριστικό για το διοξείδιο του άνθρακα στο 1° δωμάτιο.
- Το CLIMATE του εμφανίζει ιστορικό των 20 τελευταίων μετρήσεων αυτού του χαρακτηριστικού όπως βλέπουμε και από το 'Σχήμα 19'.
- Ο χρήστης επιστρέφει στην οθόνη με τις λειτουργίες παρακολούθησης και επιλέγει Log out.
- Κάνει Log out στο 'Σχήμα 20'.

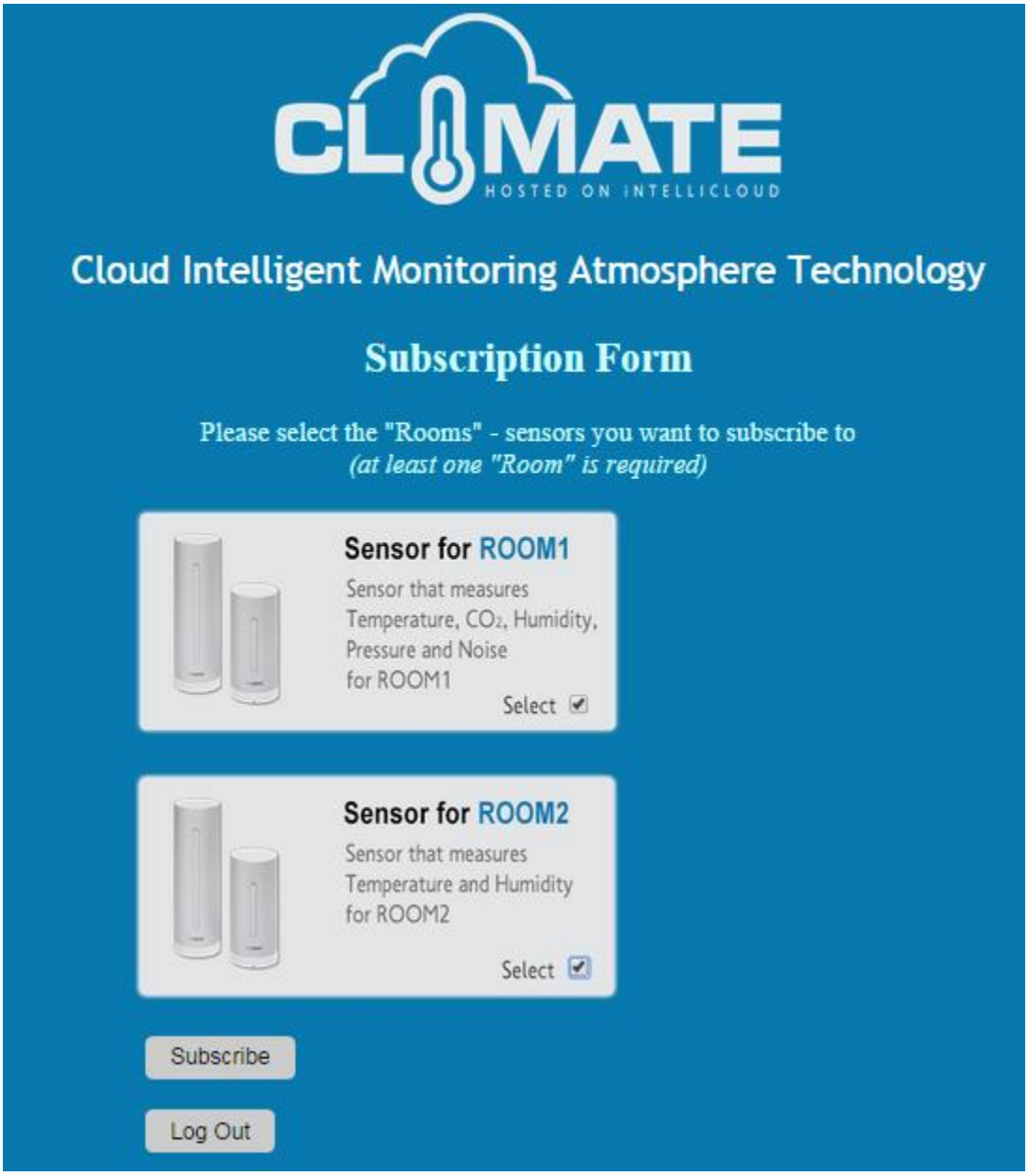
[30] http://el.wikipedia.org/wiki/HTML
[31] http://el.wikipedia.org/wiki/CSS



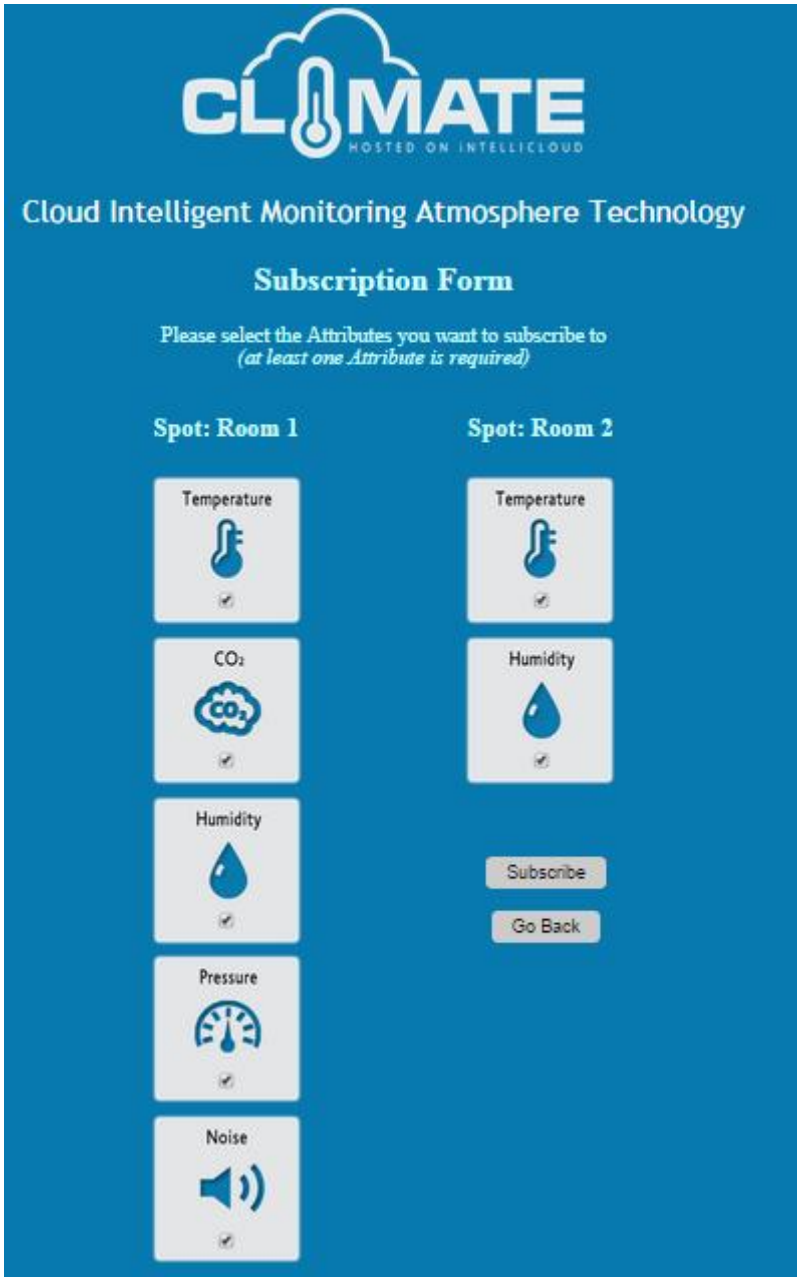
Αρχική σελίδα του CLIMATE
Σχήμα 13



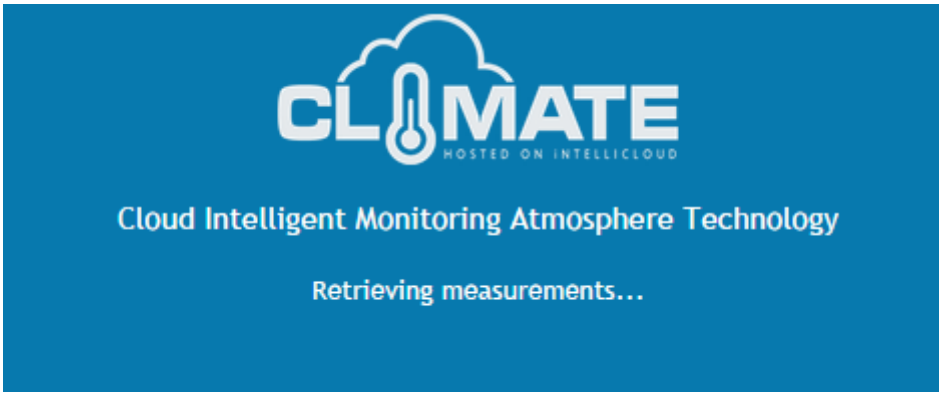
Φόρμα εισαγωγής στοιχείων του χρήστη για Login
Σχήμα 14



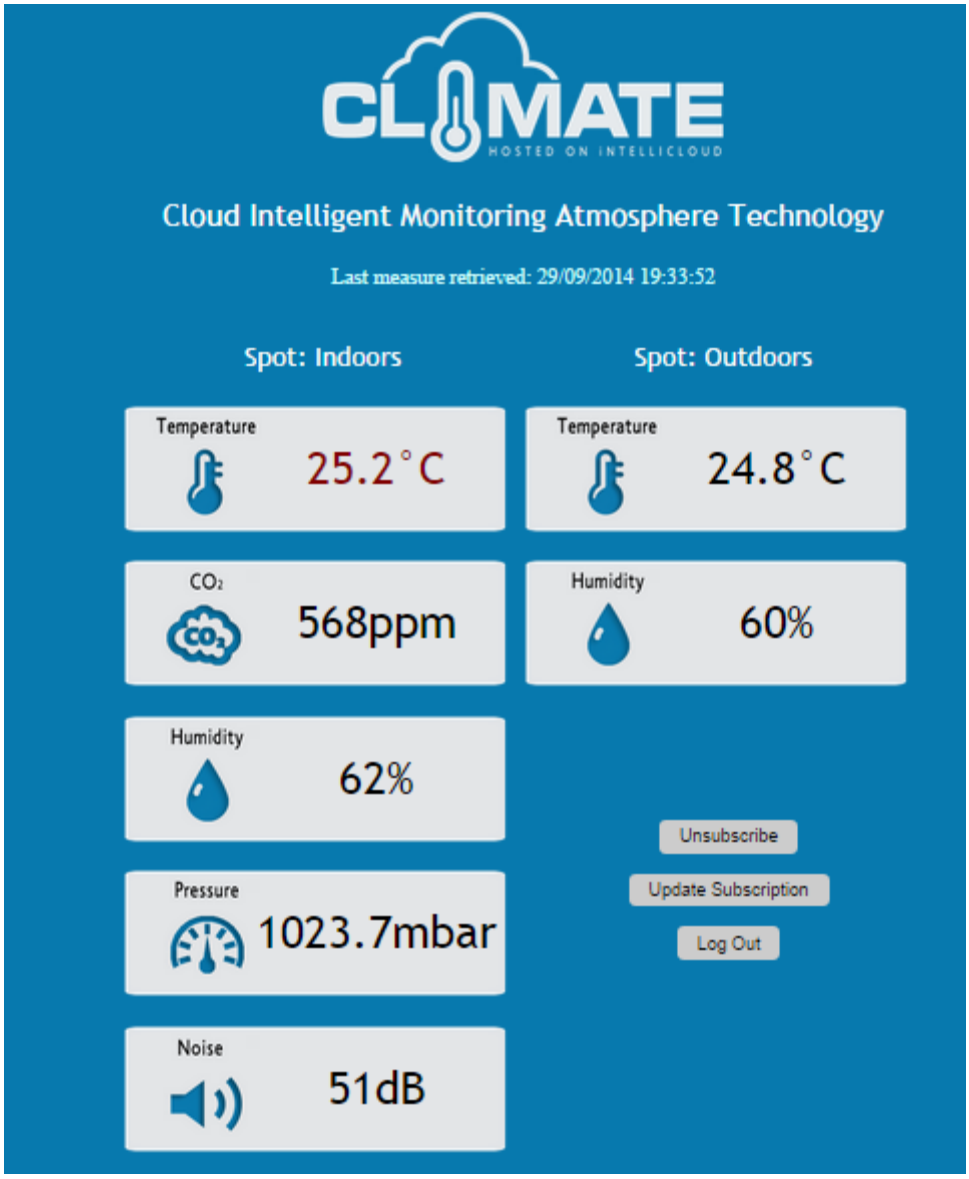
Επιλογή αισθητήρων και δωματίων
Σχήμα 15



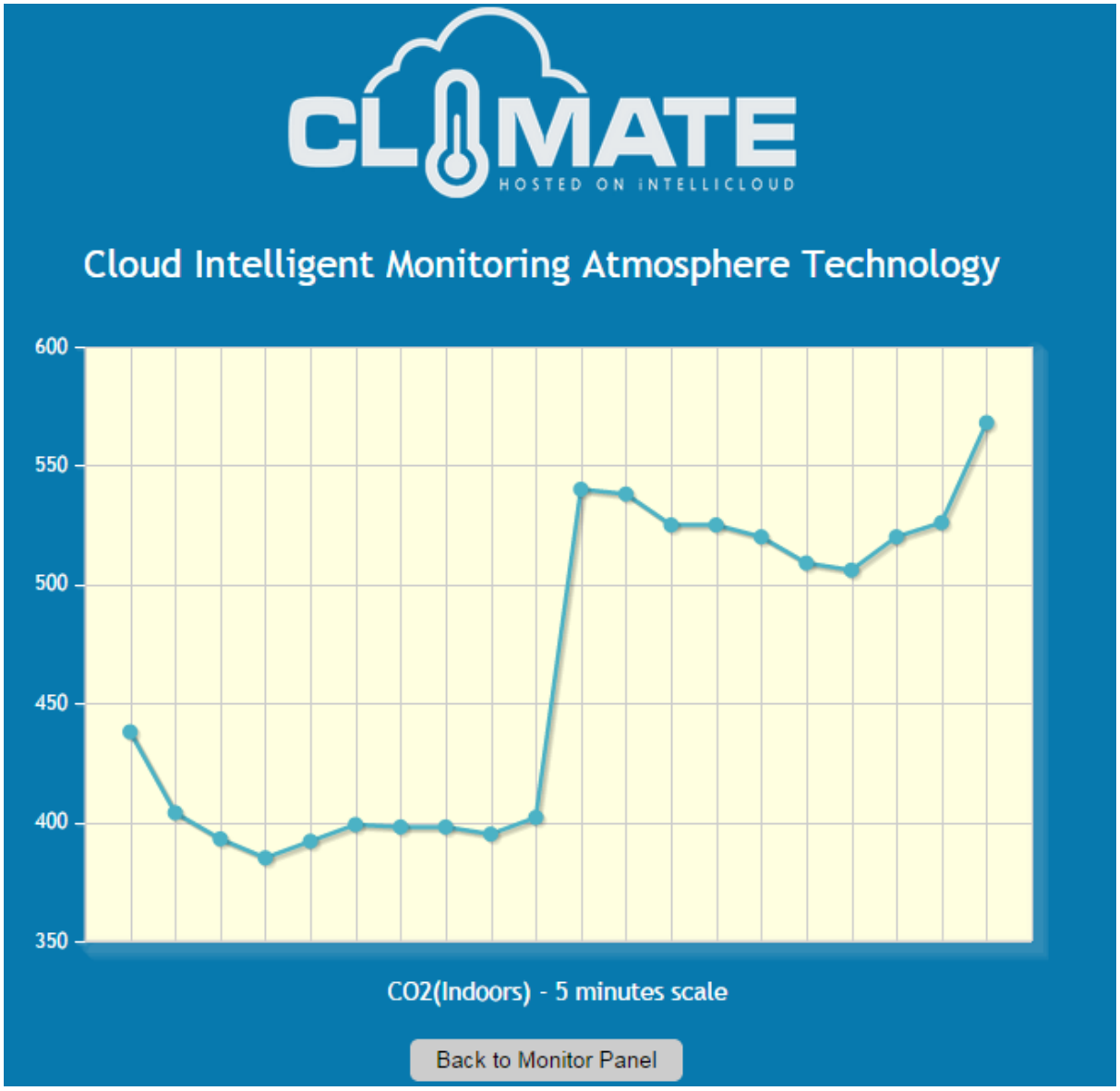
Επιλογή χαρακτηριστικών
Σχήμα 16



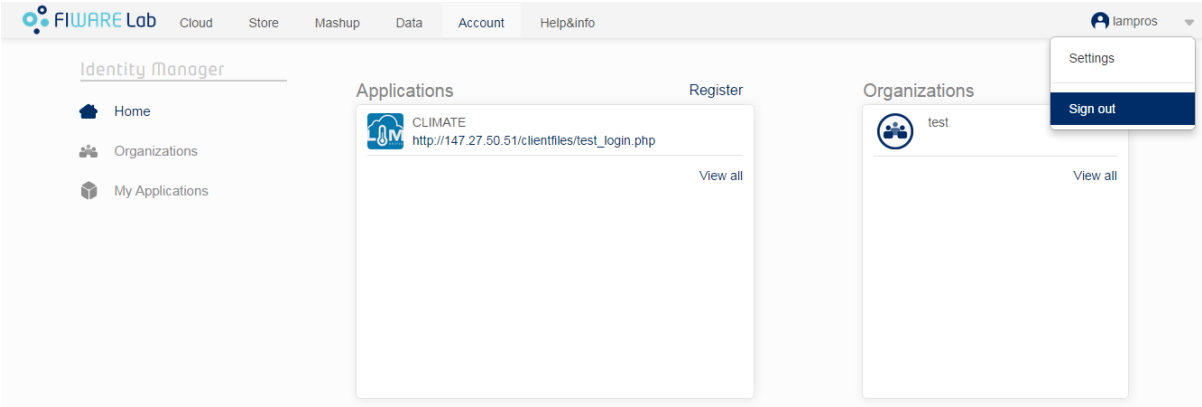
Μήνυμα ότι λαμβάνονται οι μετρήσεις που έχουν ζητηθεί
Σχήμα 17



Οθόνη παρακολούθησης
Σχήμα 18



Γράφημα του ιστορικού μετρήσεων του CO2
Σχήμα 19



Φόρμα εξόδου του χρήστη από το CLIMATE – Logout
Σχήμα 20

4 Υλοποίηση

Σε αυτό το κεφάλαιο θα περιγράψουμε αναλυτικά τη μεθοδολογία που ακολουθήσαμε στην ανάπτυξη του συστήματος. Θα περιγράψουμε τα εργαλεία που χρησιμοποιήσαμε, τον τρόπο με τον οποίο κάναμε τις υπηρεσίες και τους GEs να επικοινωνούν μεταξύ τους και φυσικά τη μεθοδολογία ανάπτυξης της Λογικής του Συστήματος (Application Logic).

4.1.1 Οι αισθητήρες

Οι αισθητήρες που χρησιμοποιούμε για τη μέτρηση των χαρακτηριστικών του περιβάλλοντος αέρα είναι οι "Netatmo – Weather Station". Θα χρησιμοποιήσουμε δύο διαφορετικούς αισθητήες (Modules), οι οποίοι ουσιαστικά θα μας χρησιμεύσουν για να παίρνουμε ταυτόχρονα μετρήσεις από δύο διαφορετικά σημεία. Ο πρώτος αισθητήρας (Module 1) μπορεί και λαμβάνει πληροφορίες για τη Θερμοκρασία, την Υγρασία, τη Βαρομετρική πίεση, το Διοξείδιο του άνθρακα και το Θόρυβο. Ο δεύτερος αισθητήρας (Module 2) λαμβάνει πληροφορίες μόνο για τη Θερμοκρασία και την Υγρασία. Και οι δύο αισθητήρες, έχουν οριστεί από τον κατασκευαστή να λαμβάνουν μετρήσεις ανά 300 δευτερόλεπτα-5 λεπτά.

4.1.2 Η εγκατάσταση των αισθητήρων

Οι αισθητήρες λαμβάνουν δεδομένα και τα στέλνουν αυτόματα στην υπηρεσία αποθήκευσης δεδομένων του κατασκευαστή-της εταιρίας Netatmo. Οι προδιαγραφές των αισθητήρων απαιτούν ρύθμιση του 1ου αισθητήρα ώστε να συνδέεται στο τοπικό ασύρματο δίκτυο και αντίστοιχες ρυθμίσεις στο 2ο αισθητήρα προκειμένου να επικοινωνεί ασύρματα με τον 1ο. Ουσιαστικά η αποστολή των δεδομένων γίνεται μόνο από τον 1ο αισθητήρα, δηλαδή το Module 1. Ο 1ος αισθητήρας είναι αυτοενισχυόμενος και για να λειτουργήσει χρειάζεται να είναι συνδεδεμένος στη μπρίζα ενώ ο δεύτερος λειτουργεί με μπαταρίες.

Αρχικά θα κάνουμε τους αισθητήρες και συγκεκριμένα 1ο να επικοινωνεί με το τοπικό ασύρματο δίκτυο.

- Προκειμένου να συνδέσουμε ένα νέο αισθητήρα, πρέπει να έχουμε ένα λογαριασμό στο Netatmo, οπότε δημιουργούμε έναν στη διεύθυνση 'auth.netatmo.com/access/login'.

- Στη συνέχεια αφού έχουμε κάνει λογαριασμό, πρέπει από τη διεύθυνση 'my.netatmo.com/app/station' να κατεβάσουμε ένα εκτελέσιμο αρχείο με όνομα "Netatmo Weather Station setup

wizard” προκειμένου να ρυθμίσουμε firmware του Module 1 προκειμένου να επικοινωνεί με το τοπικό ασύρματο δίκτυο.

Πλέον οι αισθητήρες μας μπορούν και στέλνουν δεδομένα στο Cloud του Netatmo. Μέσω της σελίδας ‘my.netatmo.com/app/station’ βλέπουμε την κατάσταση τους και κάποια χαρακτηριστικά της λειτουργίας τους. Επίσης επειδή πειραματικά θέσαμε τον 1° αισθητήρα να παρακολουθεί την ποιότητα του αέρα σε έναν εσωτερικό χώρο και το 2° αισθητήρα σε έναν εξωτερικό χώρο, ορίσαμε ονόματα για τα δύο αυτά σημεία, Indoor και Outdoor αντίστοιχα.

Ο Διαχειριστής (administrator) του συστήματος μπορεί να προσθέσει ένα νέο αισθητήρα ακολουθώντας την ίδια διαδικασία.

4.1.3 Δημιουργία εφαρμογής στην υποδομή Νέφους του Netatmo

Όπως έγινε γνωστό από την αρχιτεκτονική του συστήματός μας στα προηγούμενα κεφάλαια θέλουμε να υλοποιήσουμε εκείνο τον μηχανισμό ο οποίος θα λαμβάνει τα δεδομένα από το Νέφος του Netatmo. Ουσιαστικά την τελευταία χρονικά μέτρηση και από τους δύο αισθητήρες. Ο μηχανισμός αναφέρεται ως Sensor Data Collector.

Η επικοινωνία με το Νέφος του Netatmo γίνεται μέσω ενός RESTful API που υποστηρίζει και μηχανισμό πιστοποίησης χρηστών (authentication) για ασφαλή επικοινωνία μεταξύ του προγραμματιστή εφαρμογών και του Νέφους του Netatmo. Προκειμένου να κάνουμε χρήση του API, χρειάζεται να δημιουργήσουμε μια εφαρμογή στο εσωτερικό του Νέφους του Netatmo. Εκεί ορίζουμε σε ποιους από τους ήδη εγκατεστημένους αισθητήρες θέλουμε να έχουμε πρόσβαση και να λαμβάνουμε τα δεδομένα τους. Η εφαρμογή αυτή είναι απαραίτητη για την ταυτοποίηση των χρηστών.

Στη σελίδα λοιπόν ‘https://dev.netatmo.com/dev/’ δημιουργούμε μια νέα εφαρμογή με όνομα “intelliatmo” και αντιστοιχούμε σε αυτό τους αισθητήρες που θέλουμε να χρησιμοποιήσουμε στο σύστημά μας. Με το που δημιουργήσουμε την εφαρμογή μας, το Νέφος του Netatmo μας επιστρέφει το Client ID και το Client SECRET όπως βλέπουμε στο ‘Σχήμα 21’. Ουσιαστικά το ID και τον κωδικό της εφαρμογής μας, δύο απαραίτητα στοιχεία, προκειμένου όπως θα δούμε πιο κάτω να τα χρησιμοποιήσουμε για το ταυτοποίηση.

Client id	536402811b7759de4278eb42
Client secret	zbNlmtojTOhJKLGEfxh57wLg3VydM

Client ID & Client Secret
Σχήμα 21

Από πλευράς εγκατάστασης των αισθητήρων και παραμετροποίησης του Νέφους του Netatmo, είμαστε έτοιμοι να χρησιμοποιήσουμε το RESTful API της εταιρίας.

4.2.1 Παραμετροποίηση της υποδομής Νέφους του εργαστηρίου (Intellicloud)

Όπως αναφέραμε και σε προηγούμενα κεφάλαια, το Intellicloud χρησιμοποιεί ως Επιμελητή Εικονικών Μηχανών (Hypervisor) μια υπηρεσία του Openstack. Μέσω λοιπόν του 'Dashboard'[32], δηλαδή της διεπαφής που προσφέρει, θα δημιουργήσουμε αρχικά τη δική μας Εικονική Μηχανή, πάνω στην οποία και θα υλοποιήσουμε τη Λογική Συστήματος του CLIMATE. Η πρόσβαση στο Dashboard γίνεται μέσω της σελίδας 'http://cloud.intellicloud.tuc.gr/horizon' κάνοντας χρήση του Username και του Password που μας έχει προμηθεύσει ο διαχειριστής του Intellicloud.

4.2.2 Τοπολογία Δικτύου (Network Topology)

Αρχικά θα πρέπει να ορίσουμε το προσωπικό μας εσωτερικό Εικονικό Δίκτυο (virtual network) καθώς και το προσωπικό μας Εικονικό Δρομολογητή (virtual router). Δημιουργούμε το δικό μας δίκτυο με όνομα lampros_net και με εσωτερική διεύθυνση 64.64.1.0-64 όπως βλέπουμε και στο 'Σχήμα 22'.

Name	Subnets Associated	Shared	Status	Admin State
lampros_net	**lampros_subnet** 64.64.1.0/24	No	ACTIVE	UP

Εικονικό Δίκτυο
Σχήμα 22

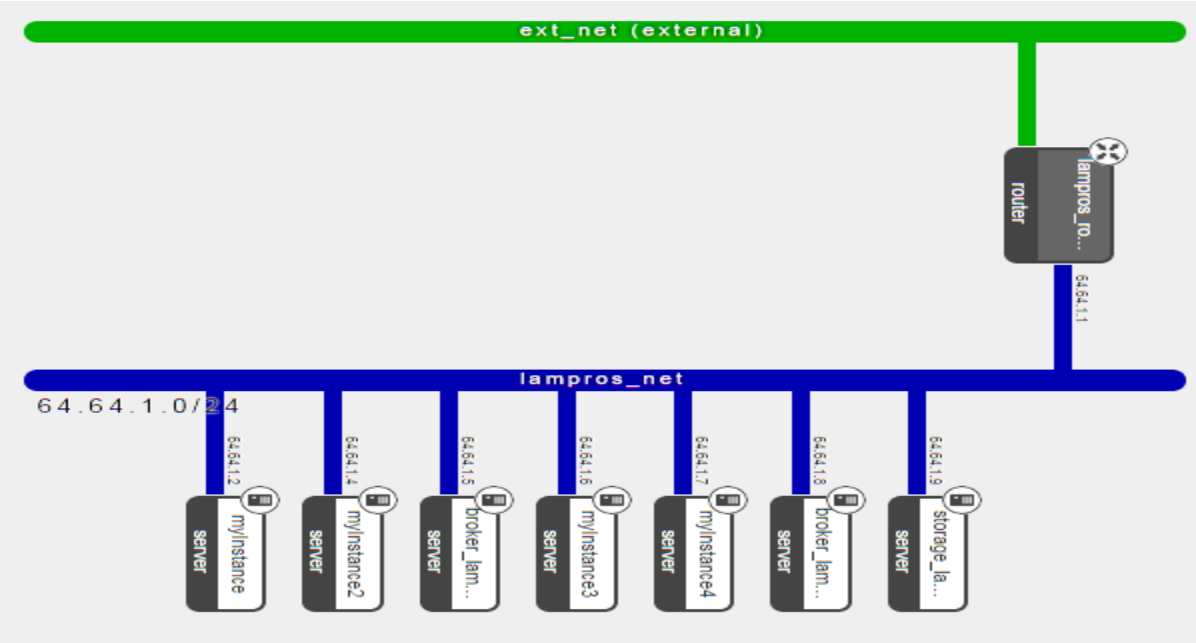
Αντίστοιχα και το προσωπικό μας εικονικό δρομολογητή με όνομα lampros_router όπως βλέπουμε στο 'Σχήμα 23'.

[32] https://www.openstack.org/software/openstack-dashboard/

	Name	Status	External Network
	lampros_router	Active	ext_net

Εικονικός Δρομολογητής
Σχήμα 23

Πλέον η τοπολογία του εσωτερικού μας προσωπικού δικτύου είναι έτοιμη και περιγράφεται από το 'Σχήμα 24'.



Network Topology
Σχήμα 24

Αφού δημιουργήσαμε την τοπολογία του δικτύου μας πρέπει να ορίσουμε κάποιους κανόνες, ουσιαστικά να ορίσουμε τα πρωτόκολλα και τις πόρτες μέσω των οποίων το δίκτυό μας θα επικοινωνεί με εξωτερικά δίκτυα. Οι πόρτες είναι επιλεγμένες σύμφωνα τις απαιτήσεις των εικονικών μηχανών και των GEs που χρησιμοποιούμε και περιγράφονται από το 'Σχήμα 25'.

	IP Protocol	From Port	To Port	Source
	Any	-	-	default
	Any	-	-	default
	TCP	1028	1028	0.0.0.0/0 (CIDR)
	TCP	80	80	0.0.0.0/0 (CIDR)
	TCP	3000	3000	0.0.0.0/0 (CIDR)
	TCP	1026	1026	0.0.0.0/0 (CIDR)
	ICMP	-1	-1	0.0.0.0/0 (CIDR)
	TCP	22	22	0.0.0.0/0 (CIDR)
	TCP	3306	3306	0.0.0.0/0 (CIDR)

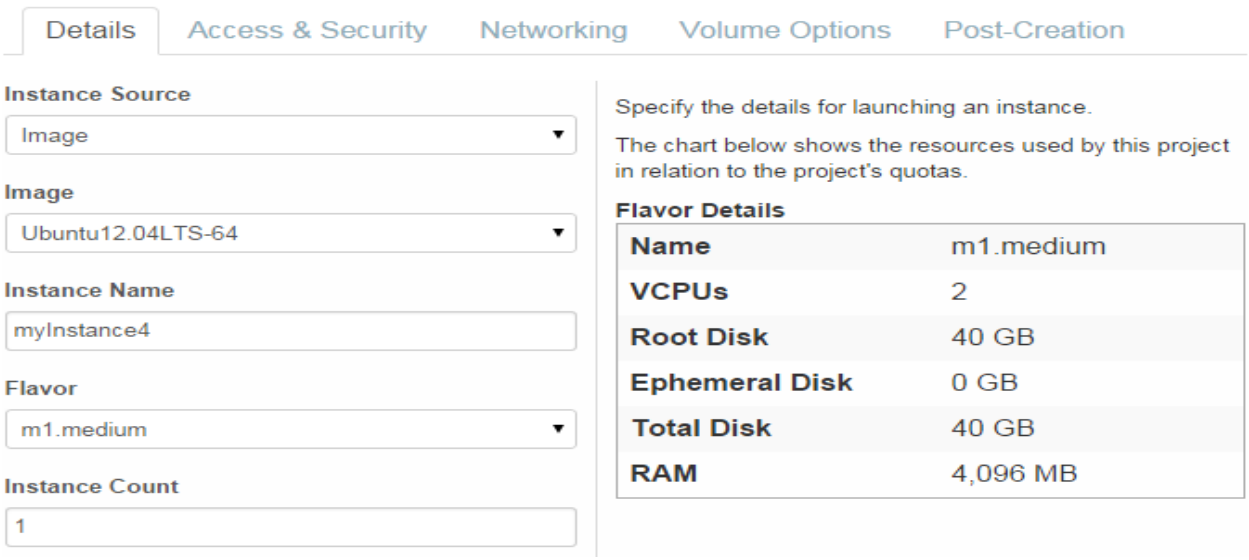
Κανόνες δικτύου
Σχήμα 25

Τώρα είμαστε σε θέση να δημιουργήσουμε πάνω σε αυτό το δίκτυο τις εικονικές μηχανές μας, καθώς επίσης και να θέσουμε σε λειτουργία τις υπηρεσίες και τους GEs που θα χρησιμοποιήσουμε.

4.3.1 Δημιουργία μιας Εικονικής Μηχανής(Virtual Machine – VM)

Σε αυτό το σημείο θα δημιουργήσουμε μια εικονική μηχανή πάνω στην οποία θα προγραμματίσουμε τη Λογική Συστήματος του CLIMATE προκειμένου να επικοινωνούν κατάλληλα μεταξύ τους οι υπηρεσίες του συστήματός μας.

Η Εικόνα (image) η οποία περιέχει το λειτουργικό σύστημα που θα χρησιμοποιήσουμε για τη δημιουργία της εικονικής μηχανής μας, είναι το 'Ubuntu 12.04 LTS-64'. Επιλέγουμε το όνομα του στιγμιοτύπου (instance) καθώς και τους πόρους που θα δεσμεύσει. Συγκεκριμένα έχουμε στη διάθεσή μας δύο εικονικούς επεξεργαστές (virtual CPUs), 40GB αποθηκευτικό χώρο και 4GB RAM όπως βλέπουμε και από το 'Σχήμα 26'.



Λεπτομέριες της Εικονικής Μηχανής
Σχήμα 26

Αφού έχουμε δημιουργήσει την εικονική μηχανλη, μας απομένει να της δώσουμε μια Εξωτερική διεύθυνση (floating IP) προκειμένου να έχει πρόσβαση στο διαδίκτυο. Βλέπουμε πλέον ότι είναι σε λειτουργία και μάλιστα προσβάσιμη από το διαδίκτυο στη διεύθυνση 147.27.50.51 όπως μας δίχνει και το '*Σχήμα* 27'.

☐	myInstance4	64.64.1.7 147.27.50.51	m1.medium | 4GB RAM | 2 VCPU | 40GB Disk	LAMPROS_KEY2	Active	None	Running

Floating IP
Σχήμα 27

4.3.2 Παραμετροποίηση της Εικονικής Μηχανής

Σε αυτό το σημείο θα χρειαστεί να εγκαταστήσουμε στην εικονική μηχανή τις κατάλληλες βιβλιοθήκες και τα απαραίτητα εργαλεία για να δουλέψουμε.

Θα κάνουμε χρήση του λογισμικού Apache προκειμένου η εικονική μηχανή να έχει τη δυνατότητα προσφοράς υπηρεσιών διακομιστή (server), της βιβλιοθήκης PHP που μας επιτρέπει να προγραμματίζουμε σε γλώσσα PHP η οποία είναι απαραίτητη για την ανάπτυξη διαδικτυακών εφαρμογών και της βάσης δεδομένων MySQL για να εξυπηρετήσει τις ανάγκες αποθήκευσης.

Προκειμένου να έχουμε πρόσβαση στην εικονική μηχανή, θα χρησιμοποιήσουμε πρωτόκολλο επικοινωνίας SSH2[33] και θα κάνουμε χρήση του εργαλείου PuTTY[34].

[33] http://en.wikipedia.org/wiki/SSH2
[34] http://www.putty.org/

Από τη στιγμή που έχουμε πρόσβαση στην εικονική μηχανή εγκαθηστούμε τα εργαλεία που προαναφέραμε και συγκεκριμένα τις εξής εκδόσεις:

- Apache 2[35]
- PHP 5[36]
- MySQL 5.0[37]

Από τη στιγμή που έχουμε ολοκληρώσει την εγκατάσταση των απαραίτητων εργαλείων, δημιουργούμε μια βάση δεδομένων προκειμένου να αποθηκεύουμε κάποιες απαραίτητες πληροφορίες για τους χρήστες που θα καλεστούν να χρησιμοποιήσουν το σύστημά μας. Οι πληροφορίες αυτές όπως περιγράψαμε παραπάνω είναι η ταυτότητα του χρήστη, ο χαρακτηριστικός κωδικός μιας συνδρομής στον Orion Context Broker GE και ο χαρακτηριστικός κωδικός μιας συνδρομής στον JSON Storage GE, δηλδή το User ID, το Subscription ID και το Storage ID. Για να γίνει κατανοητό το σχήμα της συγκεκριμένης βάσης δεδομένων παραθέτουμε το παράδειγμα του 'Σχήματος 28'.

userid	subid	storid
2760	5413607198802d5c2180c0a9	541360746429e0323a131b80

User ID – Subscription ID – Storage ID
Σχήμα 28

4.4.1 Δημιουργία μιας Εικονικής Μηχανής του Orion Context Broker GE

Από τον κατάλογο των στιγμιοτύπων (snapshots) που διατίθονται στην υποδομή Νέφους του εργαστηρίου μας θα χρησιμοποιήσουμε το στιγμιότυπο με όνομα 'context_broker_TUC-0.11.0-1x86_64_medium' προκειμένου να εκκινήσουμε μια εικονική μηχανή του Context Broker.

Κάνουμε ακριβώς την ίδια διαδικασία με τη δημιουργία της προηγούμενης εικονικής μηχανής και στη συνέχεια της αποδίδουμε μια εξωτερική διεύθυνση προκειμένου να έχει πρόσβαση στο διαδίκτυο. Κατά την ολοκλήρωση της εικονικής μηχανής του Context Broker ΓΕ, η εξωτερική διεύθυνση και οι πόροι που έχουν δεσμευτεί φαίνονται στο 'Σχήμα 29'.

[35] http://httpd.apache.org/
[36] http://php.net/
[37] http://www.mysql.com/

broker_lampros2	64.64.1.8 147.27.50.48	m1.medium | 4GB RAM | 2 VCPU | 40GB Disk

Χαρακτηριστικά της εικονικής μηχανής
του Context Broker GE
Σχήμα 29

Προκειμένου να θέσουμε σε πλήρη λειτουργία τον Context Broker GE πρέπει να εκκινήσουμε τις υπηρεσίες αποθήκευσης(βάση δεδομένων) και συνδρομής(subscription). Ο Context Broker GE χρησιμοποιεί βάση δεδομένων Mongo DB για το λόγο που αναφέραμε και παραπάνω ότι οι οντότητες με τις πληροφορίες που χρησιμοποιεί είναι της μορφής σχήματος JSON.

Η εικονική μηχανή του Context Broker GE πλέον είναι διαθέσιμη στο διαδίκτυο στη διεύθυνση 147.27.50.48:1026 όπως και το API του στην πόρτα 1026.

4.4.2 Δημιουργία οντότητας(Entity) στον Context Broker GE

Όπως έχουμε αναφέρει και στα προηγούμενα κεφάλαια, ο Orion Context Broker GE διατηρεί μια οντότητα. Η οντότητα είναι ένα JSON String το οποίο έχει σαν τιμές τις τελευταίες μετρήσεις που έχουν λάβει οι δύο αισθητήρες μας για όλα τα χαρακτηριστικά τους. Σε αυτό το σημείο πρέπει δημιουργήσουμε την οντότητα με μορφή η οποία θα περιγράφει τα δωμάτια που μελετάμε, τα χαρακτηριστικά της ατμόσφαιρας και την κατάσταση των αισθητήρων μας.

Επικοινωνούμε μέσω της διεπαφής API που διαθέτει ο Context Broker GE και συγκεκριμένα καλώντας τη μέθοδο POST στη διεύθυνση 147.27.50.48:1026/NGSI10/updateContext και με τις επικεφαλίδες 'Content-Type: application/json' & 'Accept: application/json' που ορίζουν ότι τα δεδομένα τα οποία ανταλλάσσονται είναι της μορφής JSON String. Η οντότητα που δημιουργήσαμε διατίθεται πλέον για ενημέρωση ή για πιθανή διόρθωση.

4.5.1 Δημιουργία μιας εικονικής μηχανής του JSON Storage GE

Ομοίως με τον Context Broker χρησιμοποιούμε ένα στιγμιότυπο του συγκεκριμένου GE ο οποίος βρίσκεται στον κατάλογο των στιγμιοτύπων του Νέφους προκειμένου να δημιουργήσουμε μια εικονική μηχανή που θα προσφέρει τις υπηρεσίες του JSON Storage GE. Ακολουθώντας λοιπόν παρόμοια διαδικασία, η εικονική μηχανή μας δημιουργείται με τα χαρακτηριστικά που

περιγράφονται στο 'Σχήμα 30' και είναι διαθέσημη στο διαδίκτυο στην παρακάτω διεύθυνση.

storage_lampros	64.64.1.9 147.27.50.116	m1.medium | 4GB RAM | 2 VCPU | 40GB Disk

Χαρακτηριστικά της εικονικής μηχανής
Του Json Storage GE
Σχήμα 30

Προκειμένου να θέσουμε σε πλήρη λειτουργία τον JSON Storage GE πρέπει να εκκινήσουμε τις κατάλληλες υπηρεσίες, αποθήκευσης και διεπαφής. Ο JSON Storage GE χρησιμοποιεί βάση δεδομένων Mongo DB για το λόγο που αναφέραμε και παραπάνω ότι τα δεδομένα τα οποία αποθηκεύει είναι της μορφής σχήματος JSON.

Ο JSON Storage GE είναι έτοιμος προς χρήση και είναι διαθέσιμος στο διαδίκτυο στην πόρτα 3000 και συγκεκριμένα στη διεύθυνση 147.27.50.116.

4.5.2 Παραμετροποίηση της Εικονικής Μηχανής του JSON Storage GE

Προκειμένου να κάνουμε χρήση του JSON Storage GE χρειάζεται να δημιουργήσουμε χρήστη ο οποίος θα έχει τις δικές του βάσεις δεδομένων, τις δικές του συλλογές και μέσα σε αυτές τις συλλογές τις προσωπικές του εγγραφές. Η αρχιτεκτονική του συγκεκριμένου GE έχει περιγραφεί αναλυτικά στα προηγούμενα κεφάλαια. Συνεπώς δημιουργούμε ένα χρήστη κάνοντας κλήση API με τη μέθοδο POST στη διεύθυνση 147.27.50.116:3000/users συνοδευόμενη με την επικεφαλίδα 'Content-Type: application/json' προκειμένου να δηλώσουμε ότι τα δεδομένα τα οποία ανταλλάσσονται είναι της μορφής JSON.

Στη συνέχεια δημιουργούμε την προσωπικής μας βάση δεδομένων και συλλογή μέσα στην οποία θα αποθηκεύονται οι συνδρομές. Οι συνδρομές όπως περιγράψαμε πιο πάνω αποτελούν απαντήσεις που επιστέφει ο Context Broker GE στις συνδρομές(subscriptions) που κάνουν οι χρήστες προς αυτόν. Συνεπώς προκειμένου να αποθηκεύσουμε όλες τια απαντήσεις του Context Broker GE δε θα χρειαστούμε πάνω από μία βάση δεδομένων και μια συλλογή. Τα δεδομένα μεταξύ τους θα ξεχωρίζονται μόνο από το Storage ID που θα επιστρέφουν όπως είπαμε προηγούμενα.

Προκειμένου να δημιουργήσουμε αυτά τα οποία αναφέραμε, κάνουμε κλήση API με POST στον GE μας. Χρησιμοποιούμε μία επικεφαλίδα η οποία περιέχει το όνομα και τον κωδικό του χρήστη που δημιουργήσαμε με μορφή κωδικοποίησης base64 καθώς επίσης και

την επικεφαλίδα που ορίζει ότι τα δεδομένα που ανταλλάσσονται είναι της μορφής JSON String.

Η κλήση API περιέχει στοιχεία για το όνομα τη βάσης δεδομένων καθώς και τη συλλογής που θέλουμε να δημιουργήσουμε. Συνεπώς δημιουργούμε μια βάση δεδομένων με όνομα "myDB" και μέσα σε αυτή τη συλλογή με όνομα "user1" μέσα στην οποία θα καταγράφονται όλες οι εγγραφές του συγκεκριμένου GE.

4.6.1 Λήψη των μετρήσεων από το Νέφος του Netatmo – Μηχανισμός συλλογής δεδομένων

Σε αυτό το σημείο θα περιγράψουμε τον τρόπο με τον οποίο κάνουμε λήψη των δεδομένων μας από το Νέφος του Netatmo και τα μεταφέρουμε τα δεδομένα αυτή στον History DB GE και στον Context Broker GE. Χρησιμοποιούμε γλώσσα PHP για το λόγο που προαναφέραμε πιο πάνω, ότι αποτελεί ιδανικό εργαλείο για την ανάπτυξη διαδικτυακών εφαρμογών και τη βιβλιοθήκη CURL[38]. Η εκτέλεση του κώδικα για τη λήψη και συλλογή δεδομένων από το Νέφος του Netatmo γίνεται ανά 20 δευτερόλεπτα.

Με την πρώτη απόπειρα επικοινωνίας με τον Νέφος του Netatmo, λαμβάνουμε άδεια χρήσης (token) ώστε να θεωρηθούμε πιστοποιημένοι χρήστες την ασφαλή λήψη των μετρήσεων από τους αισθητήρες μας.

Στη συνέχεια διαβάσουμε τις συσκευές μας, δηλαδή της λίστα από τους προσωπικούς μας αισθητήρες και αποθηκεύουμε σε μια μεταβλητή τις τελευταίες μετρήσεις που έχουν λάβει. Ολοκληρώνεται με αυτό τον τρόπο το πρώτο σκέλος του μηχανισμού Sensor Data Collector.

4.6.2 Ανανέωση της οντότητας στον Context Broker GE

Αρχικά δημιουργούμε το JSON String με το οποίο θα ανανεώσουμε οντότητα μέσα στον GE. Το JSON String πρέπει να περιέχει τις τελευταίες μετρήσεις που λάβαμε πιο πάνω και να είναι της μορφής που αναγνωρίζει το Context Broker.

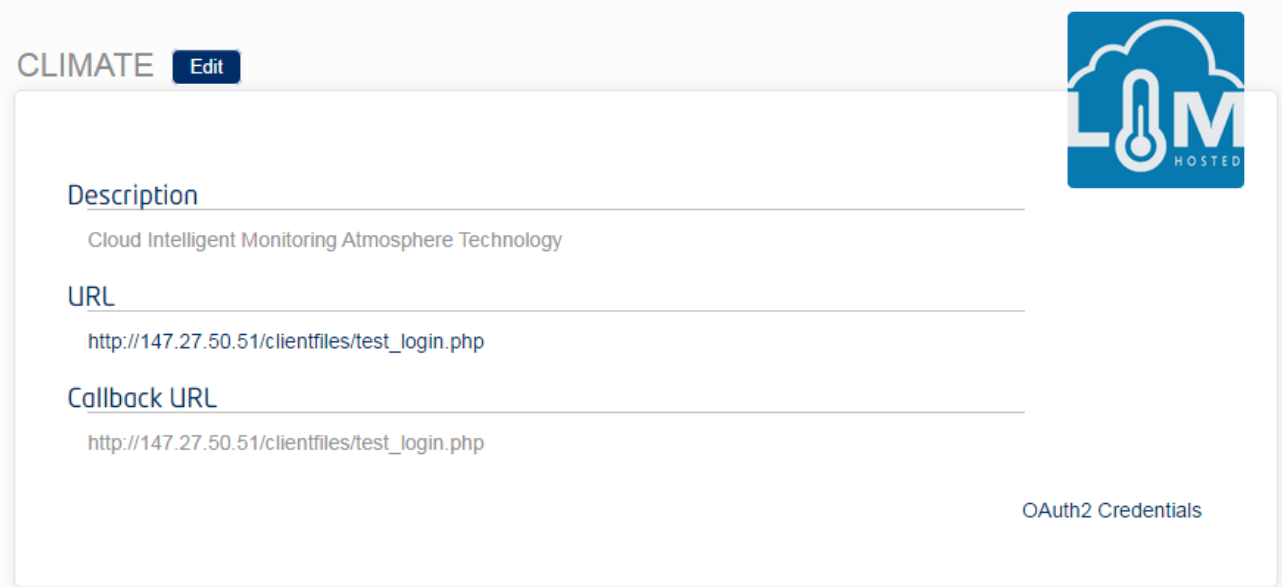
Στη συνέχεια με τη χρήση της βιβλιοθήκης CURL κάνουμε κλήση API με POST στον Context Broker GE και ανανεώνουμε τα δεδομένα της οντότητας έχοντας ως όρισμα τις τιμές που λάβαμε πιο πάνω.

[38] http://curl.haxx.se/

4.7 Παραμετροποίηση της Εικονικής Μηχανής του Keyrock – Identity Management GE

Όσο αφορά το συγκεκριμένο GE θα κάνουμε χρήση μιας δημόσιας εικονικής μηχανής (public instance – public VM)την οποία μας προμηθεύει στο FI-LAB και βρίσκεται συγκεκριμένα στη διεύθυνση https://account.lab.fi-ware.org.

Προκειμένου να χρησιμοποιήσουμε τη συγκεκριμένη δημόσια εικονική μηχανή πρέπει να ορίσουμε στο εσωτερικό του μια εφαρμογή βάσει της οποίας θα ελέγχονται και θα πιστοποιούνται οι χρήστες που έχουν έγκριση να χρησιμοποιήσουν το CLIMATE. Δημιουργούμε λοιπόν αυτή την εφαρμογή με όνομα CLIMATE όπως ακριβώς και το σύστημά μας.



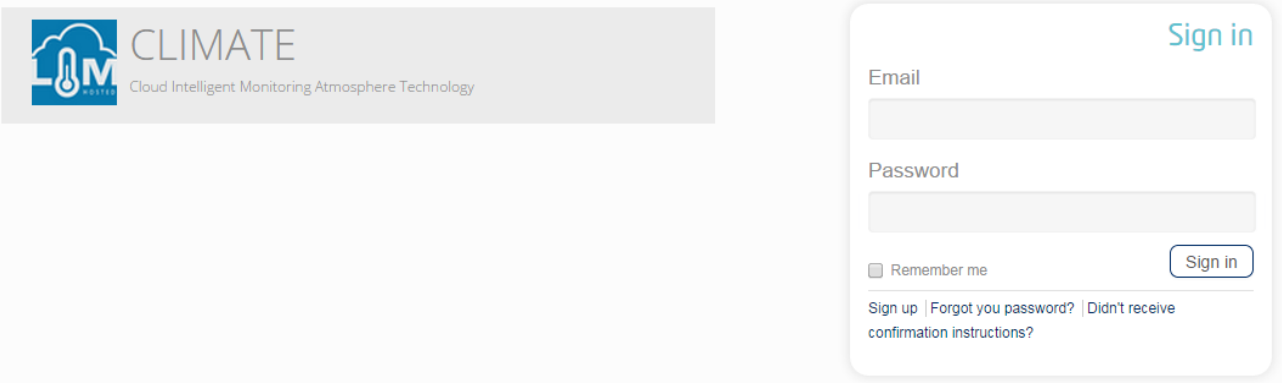
Παραμετροποίηση του Keyrock IDM GE
Σχήμα 31

Βλέπουμε από το 'Σχήμα 31' πως το πεδίο «URL» και το «Callback URL» αναγράφουν μια συγκεκριμένη διεύθυνση. Αυτή είναι η διεύθυνση στην οποία δρομολογούνται οι χρήστες όταν περάσουν τον μηχανισμό ταυτοποίησης του Keyrock IDM GE, δηλαδή η αρχική σελίδα του CLIMATE.

4.8 Ο μηχανισμός ταυτοποίησης των χρηστών

Η διεύθυνση στην οποία βρίσκεται η αρχικής σελίδα του CLIMATE είναι η: http://147.27.50.51/clientfiles/test_index.php. Στη συγκεκριμένη σελίδα υπάρχει περιγραφή του συστήματός μας και η η δυνατότητα εισόδου με την επιλογή Login για τους χρήστες που θελήσουν να κάνουν χρήση τη εφαρμογής. Αν ο χρήστης επιλέξει να εισέλθει στο σύστημά μας δρομολογείται στη δημόσια εικονική μηχανή του Keyrock IDM GE όπου καλείται να βάλει τα στοιχεία

του ή να κάνει νέα εγγραφή, όπως βλέπουμε και από το 'Σχήμα 32'.



Η φόρμα εισαγωγής των στοιχείων των χρηστών
Σχήμα 32

Αν τελικά πιστοποιηθεί από τον Keyrock IDM GE ο χρήστης δρομολογείται στη σελίδα http://147.27.50.51/clientfiles/test_login.php στην οποία με βάση τα στοιχεία του χρήστη επιχειρούμε να λάβουμε την άδεια παραμονής στην εφαρμογή (token) προκειμένου να γίνει έναρξη μιας συνεδρίας χρήσης του CLIMATE από το συγκεκριμένο χρήστη.

Αφού λάβουμε τελικά το token, ο Keyrock IDM GE επιστρέφει και το όνομα που έχει εισάγει ο χρήστης (username). Με βάση το όνομα του χρήστη, γίνεται έλεγχος για το αν αυτός ο χρήστης υπάρχει ήδη καταχωρηθεί στη βάση δεδομένων μας ή αν πρόκειται για έναν νέο χρήστη. Αν έχουμε να κάνουμε με ένα νέο χρήστη, τότε αυτός ο χρήστης εισάγεται στη βάση και τον δρομολογούμε στο αρχείο που θα κάνει την προσωπική του συνδρομή στον Context Broker GE. Αν πρόκειται για χρήστη ο οποίος ήδη έχει εγγραφεί παλαιότερα στο σύστημά μας, τότε ελέγχουμε αν ο χρήστης έχει προσωπική συνδρομή στον Context Broker GE. Αν έχει ήδη προσωπική συνδρομή, τότε δρομολογείται στο αρχείο που μπορεί να παρακολουθήσεις τα δεδομένα που τον απασχολούν. Αν δεν έχει προσωπική συνδρομή, τότε δρομολογείται στο αρχείο για να δηλώσει τη νέα του συνδρομή.

Με αυτόν τον τρόπο ολοκληρώνεται ο μηχανισμός ταυτοποίησης χρηστών. Το σύστημά μας είναι έτοιμο στο να δέχεται νέους χρήστες καθώς επίσης και να ταυτοποιεί τους παλαιότερους.

4.9 Ο μηχανισμός δημιουργίας συνδρομής στον Orion Context Broker GE

Όπως αναφέραμε και στα προηγούμενα κεφάλαια η λογική πάνω στην οποία είναι στημένο το σύστημα μας, θέλει τους χρήστες που χρησιμοποιούν το CLIMATE να έχουν δυνατότητα αφενός να

επιλέγουν τα δωμάτια τα οποία θέλουν να παρακολουθούν και αφετέρου τα χαρακτηριστικά της ατμόσφαιρας σε κάθε δωμάτιο ξεχωριστά.

Σε αυτό το σημείο θα δούμε αναλυτικά πως υλοποιείται η λειτουργία επιλογής των δωματίων. Αρχικά ελέγχουμε αν στη συνεδρία στην οποία βρίσκεται ο χρήστης έχουν ήδη επιλεχθεί αισθητήρες από προηγούμενη συνδρομή στον Context Broker GE. Αν ήδη έχει επιλεχθεί κάποιος αισθητήρας, αναιρείται αυτή η επιλογή προκειμένου να έχει τη δυνατότητα ο χρήστης να επιλέξει ξανά δωμάτια.

Αφού ο χρήστης επιλέξει τα δωμάτια τα οποία θα παρακολουθεί, οι επιλογές του αποστέλλονται με τη μέθοδο POST στε ένα άλλο αρχείο στο οποίο γίνεται η επιλογή των χαρακτηριστικών της ατμόσφαιρας.

Αφού διαβάσουμε τις επιλογές του χρήστη ως προς τα δωμάτια και τα χαρακτηριστικά της ατμόσφαιρας, συνθέτουμε το κατάλληλο JSON String προκειμένου να κάνουμε με βάση αυτό αίτηση στον Context Broker GE για μια νέα συνδρομή. Το JSON String πρέπει αφενός να περιέχει τις επιλογές του χρήστη και αφετέρου να είναι της μορφής-σχήματος που μπορεί να αναγνωρίσει ο GE. Στη συνέχεια με κλήση API και με τη μέθοδο POST όπως είδαμε και προηγούμενα κεφάλαια, δημιουργούμε τη νέα συνδρομή στον Context Broker GE. Ο Context Broker GE μας επιστρέφει το Subscription ID που ουσιαστικά ορίζει μοναδικά την συνδρομή που μόλις κάναμε. Με βάση αυτό το Subscription ID ανανεώνουμε τη βάση δεδομένων που διατηρεί τα στοιχεία των χρηστών προκειμένου να αντιστοιχήσουμε το User ID του χρήστη με το νέο Subscription ID του.

Με αυτό τον τρόπο ολοκληρώνεται η διαδικασία δημιουργίας συνδρομής χρηστών στον Context Broker GE.

4.10 Ο μηχανισμός με τον οποίο αποθηκεύονται οι απαντήσεις του Orion Context Broker GE

Όπως αναφέραμε και στα προηγούμενα κεφάλαια ο χρήστης έχει τη δυνατότητα να κάνει συνδρομή στον Context Broker GE έχοντας μια σειρά από επιλογές. Εδώ θα δούμε τον τρόπο με τον οποίο διαχειριζόμαστε τις απαντήσεις του Context Broker GE. Γενικά η απάντηση για κάθε συνδρομή στον Context Broker GE μπορεί να στέλνεται και σε διαφορετική διεύθυνση. Στη δικιά μας υλοποίηση όλες οι απαντήσεις του συγκεκριμένου GE στέλνονται στην ίδια διεύθυνση και εκεί τις διαχειριζόμαστε αναλόγως.

Το πρώτο πράγμα που κάνουμε με το που λάβουμε τα δεδομένα τα οποία επιστρέφει ο Context Broker GE είναι να διαβάσουμε αυτά

δεδομένα και να τα αποθηκεύσουμε μέσα σε μια μεταβλητή που θα διατηρεί τη μορφή JSON Sring. Στη συνέχεια λοιπόν αποκωδικοποιούμε τα δεδομένα αυτά από τη μορφή JSON String προκειμένου να πάρουμε πληροφορία για το Subsrcription ID, δηλαδή η απάντηση που μόλις λάβαμε σε ποια συνδρομή αντιστοιχεί. Στη συνέχεια ελέγχουμε μέσα στη βάση δεδομένων που διατηρεί του χρήστες, αν στο συγκεκριμένο Subscription ID αντιστοιχεί κάποιο Storage ID. Αν δεν αντιστοιχεί, σημαίνει ότι ο συγκεκριμένος χρήστης έκανε για πρώτη φορά συνδρομή στον Orion Context Broker GE. Συνεπώς κάνουμε κλήση API στο JSON Storage GE προκειμένου να δημιουργήσουμε ένα νέο Storage ID και να αποθηκεύσουμε στη συνέχεια εκεί τα δεδομένα του χρήστη ως μια νέα εγγραφή στην ήδη υπάρχουσα συλλογή του συγκεκριμένου GE.

Αφού δεσμεύσουμε το καινούριο Storage ID το καταχωρούμε στη βάση δεδομένων των χρηστών προκειμένου πλέον να έχουμε μια πλήρη αντιστοιχία μεταξύ User ID, Subscription ID και Storage ID. Αντίστοιχα αν υπάρχει τελικά στη βάση κάποιο Storage ID, κάνουμε κλήση στον JSON Storage GE, προκειμένου να τον ενημερώσουμε για τα νέα δεδομένα που λάβαμε από το Context Broker GE

Με αυτό τον τρόπο ολοκληρώνεται η υλοποίηση του μηχανισμού αποθήκευσης των δεδομένων που επιστρέφει ο Context Broker GE.

4.11 Ο μηχανισμός της ανανέωσης και της διαγραφής μιας συνδρομής στον Orion Context Broker GE

Όπως αναφέραμε και στα προηγούμενα κεφάλαια, το σύστημά μας προσφέρει και τη δυνατότητα ανανέωσης καθώς και διαγραφής της συνδρομής που έχει κάνει ο χρήστης. Για να γίνει κάτι τέτοιο χρειάζεται αρχικά να ενημερωθούμε για το User ID του χρήστη που βρίσκεται αυτή τη στιγμή μέσα στη συνεδρία.

Στη συνέχεια κάνουμε αναζήτηση μέσα στη βάση δεδομένων των χρηστών προκειμένου να βρούμε το αντίστοιχο Subscription ID και Storage ID του χρήστη. Με το που βρούμε αυτά τα δύο στοιχεία, κάνουμε κλήση API στον Context Broker GE και στον JSON Storage GE προκειμένου να ενημερώσουμε τον πρώτο ότι θέλουμε να διαγράψουμε την συνδρομής και το δεύτερο ότι θέλουμε να σβήσει τα δεδομένα που κρατάει για το συγκεκριμένο χρήστη ως εγγραφή.

Τέλος, αφού έχουμε ενημερώσει τους δύο αυτούς GEs ενημερώνουμε και τη βάση των χρηστών διαγράφοντας το Subscription ID και Storage ID του συγκεκριμένου χρήστη. Αφού έχουμε υλοποιήσει αυτά τα βήματα στέλνουμε το χρήστη στο αρχείο που συντάσεται η μια νέα συνδρομή προς τον Context Broker GE. Ο μηχανισμός με

τον οποίο υλοποιείται η διαγραφή μιας συνδρομή γίνεται ακριβώς με τον ίδιο τρόπο όπως η ανανέωση της συνδρομής μόνο που τώρα ο χρήστης δε δρομολογείται στο αρχείο για τη δημιουργία νέας συνδρομής.

4.12 Ο μηχανισμός με τον οποίο γίνεται η παρακολούθηση των τιμών

Αφού έχει εγγραφεί ένας χρήστης, έχει κάνει συνδρομή στον Context Broker GE και έχει επιλέξει ουσιαστικά τους αισθητήρες και τα χαρακτηριστικά που τον ενδιαφέρουν, δεν έχει παρά να φτάσει στο σημείο τελικά της παρακολούθησης των κυμαινόμενων τιμών.

Η παρακολούθηση των τιμών γίνεται μέσα από ένα αρχείο-σελίδα το οποίο και ανανεώνεται ανά 20 δευτερόλεπτα προκειμένου να κάνει έλεγχο για τυχόν αλλαγές των τιμών. Στο συγκεκριμένο αρχείο, αρχικά διαβάζουμε το User ID του χρήστη που βρίσκεται εντός της συνεδρίας προκειμένου να ενημερωθούμε ποιον χρήστη καλούμαστε να εξυπηρετήσουμε.

Η όλη λογική όπως έχουμε ήδη περιγράψει θέλει στη διαδικασία της παρακολούθησης των δεδομένων να ενημερωνόμαστε από τον JSON Storage GE με βάση το χαρακτηριστικό Storage ID του χρήστη. Συνεπώς στη συνέχεια κάνουμε αναζήτηση μέσα στη βάση των χρηστών προκειμένου να βρούμε το Storage ID του συγκεκριμένου χρήστη. Αν δεν βρεθεί σημαίνει ότι ο χρήστης δεν έχει κάνει ακόμα κάποια προσωπική συνδρομή και συνεπώς του επιστρέφουμε το αντίστοιχο μήνυμα.

Αν βρεθεί το Storage ID, κάνουμε κλήση API στο JSON Storage GE με βάση το συγκεκριμένο Storage ID, προκειμένου να πάρουμε το JSON String με τα δεδομένα και τις μετρήσεις που αφορούν το συγκεκριμένο χρήστη. Στη συνέχεια αποκωδικοποιούμε τα δεδομένα που μόλις λάβαμε προκειμένου να μπορούμε να τα επεξεργαστούμε και εμφανίζουμε στο χρήστη τα δεδομένα που τον ενδιαφέρουν καθώς επίσης και την ώρα που έγινε η τελευταία μέτρηση.

Επίσης, κατά τη διαδικασία της παρακολούθησης των τιμών, ο χρήστης έχει τη δυνατότητα να εμφανίζει για όποιο χαρακτηριστικό επιθυμεί ένα γράφημα των τελευταίων 20 τιμών του. Οι τιμές αυτές βρίσκονται μέσα στη βάση δεδομένων που διατηρεί το ιστορικό των μετρήσεων και λαμβάνονται με κλήσεις API στον History DB GE που έχουμε υλοποιήσει για ακριβώς αυτό το σκοπό. Για τη σχεδίαση των γραφημάτων χρησιμοποιούμε τη βιβλιοθήκη PHP Chart Lite[39].

[39] http://phpchart.org/

Τέλος καθώς ο χρήστης παρακολουθεί τις τιμές που τον ενδιαφέρουν έχει και τη δυνατότητα να ανανεώσει την συνδρομή του στον Context Broker GE ώστε να επιλέξει άλλους αισθητήρες, άλλα δωμάτια και άλλα χαρακτηριστικά να παρακολουθεί. Με αυτό τον τρόπο ολοκληρώνεται ο μηχανισμός παρακολούθησης στο σύστημά μας.

4.13 Ο μηχανισμός αποσύνδεσης του χρήστη

Μέσα στο σύστημά μας υπάρχει η επιλογή αποσύνδεσης του χρήστη με τη μορφή κουμπιού Log Out. Κατά την επιλογή του "log out" καταστρέφεται αυτόματα η συνεδρία του συγκεκριμένου χρήστη και δρομολογείται ο χρήστης σε μια φόρμα που παρέχεται από τη δημόσια εικονική μηχανή του Keyrock IDM GE προκειμένου να αποσυνδεθεί από το CLIMATE.

4.14 Ο μηχανισμός με τον οποίο διατηρείται η συνεδρία του χρήστη

Όπως έχουμε αναφέρει και πιο πριν τα δεδομένα του χρήστη και οι επιλογές του αποθηκεύονται μέσα σε μια συνεδρία προκειμένου να αποφεύγεται σε άτομα να κάνουν χρήση των αρχείων του συστήματός μας χωρίς να είναι πιστοποιημένοι χρήστες. Συνεπώς σε κάθε αρχείο ελέγχεται για το αν υπάρχει ενεργή συνεδρία (active session) του συγκεκριμένου χρήστη. Αν δεν υπάρχει, ο χρήστης στέλνεται στην αρχική σελίδα της εφαρμογής ώστε να κάνει εγγραφή ή σύνδεση αν έχει ήδη λογαριασμό.

4.15 Περιπτώσεις χρήσης

Σε αυτό το σημείο θα παραθέσουμε τρεις περιπτώσεις προκειμένου να δείξουμε τη λειτουργικότητα του συστήματος που έχουμε υλοποιήσει και την χρησιμότητα που μπορεί να έχει σε σενάρια τη καθημερινής ζωής.

1η Περίπτωση

Η πρώτη περίπτωση στην οποία θα μπορούσε το σύστημά μας να έχει εφαρμογή είναι στην ανάγκη ενός νοσοκομείου να παρακολουθεί συνεχώς την ποιότητα του αέρα μέσα στα δωμάτια νοσηλείας. Είναι γνωστό ότι σε ειδικές και βαριές περιπτώσεις ασθενών η υγρασία, οι υψηλές θερμοκρασίες, το διοξείδιο του άνθρακα και ο θόρυβος μπορούν να επιβαρύνουν την υγεία τους ή να αναστείλουν την ανάρρωσή τους.

Για παράδειγμα, ο υπεύθυνος ελέγχου μιας πτέρυγας ενός

νοσοκομείου θα μπορούσε να κάνει συνδρομή στο χώρο της εντατικής θεραπείας που η θερμοκρασία δεν πρέπει να ξεπερνάει τους 20° με 22° βαθμούς και στις αίθουσες που νοσηλεύονται ασθενείς με άσμα όπου η υγρασία και το διοξείδιο του άνθρακα πρέπει να είναι σε πολύ χαμηλά επίπεδα.

2η Περίπτωση

Ένα δεύτερο και ίσως πιο εμπορικό σενάριο θα ήταν η δημιουργία ενός «χάρτη» που θα απεικονίζει τουριστικές περιοχές ή ακόμα περιοχές στο εσωτερικό μιας πόλης τις οποίες θα τις περιγράφει με βάση την ποιότητα του περιβάλλοντος που επικρατεί εκεί. Ουσιαστικά θα μπορεί να υπάρχει μια συνολική εποπτεία μιας περιοχής για τους μέσους όρους του θορύβου που έχει, της υγρασίας, της βαρομετρικής πίεσης κλπ.

Αυτό το σενάριο για παράδειγμα μπορεί να επηρεάσει την επιλογή ενός ζευγαριού για το που θα περάσουν τις καλοκαιρινές τους διακοπές. Κάποιοι θα απέφευγαν να κλείσουν δωμάτιο σε μια περιοχή με υψηλούς μέσους όρους θορύβου για το λόγο ότι μπορεί να είναι ηλικιωμένοι ή με υψηλό δείκτη υγρασίας στην περίπτωση που αντιμετωπίζουν αρθριτικά προβλήματα.

Το ίδιο σενάριο θα μπορούσε να επηρεάσει και την επιλογή για την αγορά ενός σπιτιού σε κάποιο σημείο της πόλης, τις τιμές των οικοπέδων σε μια ευρύτερη περιοχή, την επιλογή για την έναρξη λειτουργίας ενός εστιατορίου κ.α.

3η Περίπτωση

Μια τρίτη περίπτωση στην οποία θα μπορούσε να έχει χρήση το CLIMATE θα ήταν στο εσωτερικό μιας αποθήκης. Υπάρχουν εταιρίες που καλούνται να αποθηκεύουν για μεγάλο χρονικό διάστημα ευαίσθητα προϊόντα όπως για παράδειγμα τρόφιμα. Αυτές οι εταιρίες πρέπει να έχουν λεπτομερή εικόνα για την κατάσταση κυρίως της θερμοκρασίας και τη υγρασίας που επικρατούν στο εσωτερικό του χώρου.

Σε μια μελλοντική δουλειά, θα μπορούσε ίσως το σύστημά μας αυτόματα να επικοινωνεί στη βάση του διαδικτύου των πραγμάτων (IoT) με άλλες συσκευές, να τις θέτει σε λειτουργία προκειμένου αυτές να βελτιώνουν την κατάσταση της θερμοκρασίας και της υγρασίας και να την προσαρμόζουν στα επιθυμητά επίπεδα.

5.Συμπεράσματα και μελλοντική δουλειά

5.1 Συμπεράσματα

Η λειτουργία του CLIMATE πληρεί τους στόχους που είχαμε θέσει στο 1° κεφάλαιο τόσο στο ζήτημα διαχείρισης χρηστών όσο και στο ζήτημα δημιουργίας συνδρομής και παρακολούθησης χώρων, αισθητήρων, τιμών κλπ.

Σημαντικό στοιχείο ήταν η επιλογή χρήσης των Generic Enablers και των υπηρεσιών Νέφους κυρίως στο ότι μείωσε σε πολύ μεγάλο βαθμό την πολυπλοκότητα του κώδικά μας και της Λογικής Συστήματος. Αυτό γιατί τρεις απαραίτητες λειτουργίες, αυτές της ταυτοποίησης χρστών, της συνδρομής σε δωμάτια-χαρακτηριστικά περιβάλλοντος αέρα και της αποθήκευσης σχημάτων JSON θα έπρεπε να τις είχαμε αναπτύξει εμείς εξ ολοκλήρου. Επίσης η συντήρηση καθώς και η αναβάθμιση των υπηρεσιών αυτών στο μέλλον δε θα επηρεάσει τη δουλειά μας, αλλά πιθανώς να βελτιώσει την απόδοση του συστήματός μας.

Ο μόνος στόχος ο οποίος δεν επιτύχαμε και επαφίεται στο να ολοκληρωθεί σε μελλοντική δουλειά, είναι το σύστημα μας να έχει τη δυνατότητα δυναμικής πρόσθεσης και αφαίρεσης αισθητήρων. Αυτή τη στιγμή λειτουργεί εξαιρετικά μόνο για συγκεκριμένο – γνωστό αριθμό αισθητήρων.

Όπως περιγράψαμε στην υλοποίηση του συστήματός μας, η εγκατάσταση νέων αισθητήρων προβλέπει τη δήλωσή τους στο Νέφος του Netatmo και παράλληλα την αντιστοίχησή τους στον προσωπικό μας λογαριασμό στο Νέφος του Netatmo με βάση τον οποίο έχουμε τη δυνατότητα να λαμβάνουμε μετρήσεις για μια σειρά από αισθητήρες που μας ανοίκουν.

Η υλοποίησή μας προς το παρόν καλύπτει μέχρι και τη δυνατότητα που προσφέρει η υπηρεσία του Sensor Data Collector. Δηλαδή αν εγκατασταθούν νέοι αισθητήρες και αντιστοιχιστούν στο λογαριασμό μας στο Νέφους του Netatmo, τότε αυτόματα το σύστημά μας τους αναγνωρίζει και μπορεί να λαμβάνει τα δεδομένα τα οποία παράγουν.

Τα προβλήματα που αντιμετωπίσαμε όμως προκειμένου να δημιουργήσουμε ένα σύστημα που πράγματι θα παρείχε τη δυνατότητα δυναμικής προσθαφαίρεσης αισθητήρων ήταν συγκεκριμένα τα εξής:

1. Ο Context Broker GE όπως παραθέτουμε στη βιβλιογραφία χρειάζεται την αρχικοποίηση μιας οντότητας καθιστώντας δύσκολη την παραμετροποίησή της. Απαιτεί να ορίσεις εξαρχής ένα μοντέλο με βάση το οποίο θα δουλέψεις, θα ανανεώσεις τις τιμές του και εν τέλει να κάνεις συνδρομές επ αυτού.

Για να το κάνουμε πιο συγκεκριμένο, στη δικιά μας περίπτωση ο Context Broker GE απαιτεί να γνωρίζει εκ των προτέρων πόσα δωμάτια παρακολουθούμε, ουσιαστικά πόσους αισθητήρες έχουμε στη διάθεσή μας και ποια χαρακτηριστικά έχει ο κάθε αισθητήρας. Δηλαδή η περιγραφή του μοντέλου «δωμάτια-αισθητήρες-χαρακτηριστικά» πρέπει να είναι συγκεκριμένο.

2. Αν υποθέσουμε ότι καταφέρνουμε να προσπεράσουμε αυτό το πρόβλημα, αντιμετωπίζουμε ένα ακόμα πρόβλημα στις συνδρομές και το πως αυτές περιγράφονται μέσω της αίτησης στο API του Context Broker GE. Ο Context Broker GE δέχεται συνδρομές με βάση τα δωμάτια και τα χαρακτηριστικά. Παρόλα αυτά δεν ακολουθεί μια λογική σειρά που λέει πως αρχικά δηλώνεις ποια δωμάτια θες να παρακολουθείς και μετά ποια χαρακτηριστικά σε κάθε δωμάτιο. Αλλά ακολουθεί μια αντίληψη που θέλει να δηλώνεις ξεχωριστά δωμάτια και ξεχωριστά χαρακτηριστικά. Τα χαρακτηριστικά δηλαδή είναι ανεξάρτητα των δωματίων και το αντίστροφο.

   Για να γίνουμε πιο συγκεκριμένοι, υποθέτουμε πως έχουμε τρία δωμάτια στα οποία μπορούμε να παρακολουθούμε σε κάθε ένα από αυτά τη θερμοκρασία, την υγρασία και το CO2. Στην συνδρομή δεν μπορούμε να δηλώσουμε ότι θέλουμε να παρακολουθούμε από τα δύο πρώτα δωμάτια τη θερμοκρασία και την υγρασία και από το τρίτο μόνο τη θερμοκρασία. Από τη στιγμή που υπάρχει η υγρασία και στο τρίτο, τότε θα μας την επιστρέψει. Αυτό γίνεται γιατί η συνδρομή κοιτάει πρώτα ποια χαρακτηριστικά θέλει ο χρήστης να παρακολουθήσει και μετά πάει και τα ψάχνει στα δωμάτια που έχει δηλώσει ο χρήστης.

   Συνεπώς εάν προσθέτουμε αυθαίρετα αισθητήρες των οποίων τα χαρακτηριστικά έχουν κοινά ονόματα, δηλαδή έτσι όπως ονοματίζονται από τα αποτελέσματα που επιστρέφει το API του Netatmo, τότε δε θα μπορούμε να δημιουργήσουμε συγκεκριμένες και πολύπλοκες συνδρομές.

3. Το βασικότερο ίσως όλων είναι ότι προς το παρόν ο Context Broker GE δεν υποστηρίζει τη λειτουργία της διαγραφής ενός χαρακτηριστικού ή πόσο μάλλον ενός αισθητήρα-δωματίου. Παρόλο που αναφέρεται στη βιβλιογραφία ως δυνατότητα στο API που προσφέρει, στην πραγματικότητα δεν ολοκληρώνει ποτέ μια αίτηση διαγραφής. Στο σύστημά μας για παράδειγμά δε θα είχε κάποιο νόημα να μπορείς να θέτεις νέους αισθητήρες χωρίς να έχει τη δυνατότητα διαγραφής τους.

5.2 Μελλοντική δουλειά

5.2.1 Δυναμική προσθαφαίρεση αισθητήρων

Ως μελλοντική δουλειά θα μπορούσε αρχικά να γίνει η προσπάθεια υπέρβασης των προβλημάτων που αναφέραμε με σκοπό το CLIMATE να αποτελέσει ένα σύστημα το οποίο θα προσφέρει τη δυνατότητα δυναμικής προσθαφαίρεσης αισθητήρων.

Μια πρώτη ιδέα θα ήταν κατά τη λήψη των δεδομένων από το Νέφος του Netatmo να ονοματίζαμε ξεχωριστά τα κοινά χαρακτηριστικά του κάθε αισθητήρα. Για παράδειγμα αν έχουμε τρεις αισθητήρες που και οι τρεις μπορούν και μετρούν τη θερμοκρασία, τότε η ονοματολογία κάθε χαρακτηριστικού θα πρέπει να φέρει και την ταυτότητα του αισθητήρα από τον οποίο έγινε η μέτρηση, για παράδειγμα “Temperature1”, “Temperature2”, “Temperature3”. Και με βάση αυτά τα ονόματα κάθε φορά να δημιουργούσαμε μια νέα οντότητα μέσα στον Context Broker GE.

Αυτό θα διόρθωνε το δεύτερο πρόβλημα που παραθέσαμε, αφού δε θα μπορούσαν να υπάρχουν χαρακτηριστικά με κοινά ονόματα.

Ένα επόμενο βήμα, από τη στιγμή που δεν έχουμε τη δυνατότητα παραμετροποίησης της οντότητας του Context Broker GE, θα ήταν όταν κάθε φορά που προστίθεται ένας αισθητήρας ουσιαστικά να δημιουργούμε μια οντότητα από την αρχή. Να ξεχνάμε την παλιά και πλέον να κάνουμε νέες αιτήσεις για συνδρομή μόνο στη καινούρια. Κάτι τέτοιο σαφώς θα δημιουργούμε πολλές αλλαγές στη Λογική Συστήματος η οποία διαχειρίζεται και συντάσσει τις συνδρομές προς τον Context Broker GE. Όσο αφορά τη δυνατότητα της διαγραφής, η διόρθωση αυτού του προβλήματος επαφίεται στην έκδοση νέων εκδόσεων ή αναβαθμίσεων του Context Broker GE.

Όλα τα παραπάνω σαφώς προϋποθέτουν τη χρήση ίδιων ακριβώς αισθητήρων με αυτούς που χρησιμοποιήσαμε και εμείς για την υλοποίηση του συστήματός μας, τους “Netatmo – Urban Weather Station”.

5.2.1 Ενημέρωση και έλεγχος συσκευών με βάση τα κατώφλια των τιμών που μελετάμε

Μια δεύτερη προσπάθεια ως μελλοντική δουλειά θα ήταν η ενημέρωση του χρήστη όταν η τιμή κάποιου χαρακτηριστικού που μελετάει περάσει ένα συγκεκριμένο κατώφλι που ο ίδιος έχει ορίσει.

Για παράδειγμα, μπορεί κάποιος να ενδιαφέρεται να ενημερωθεί μέσω mail ή ακόμα με sms στο κινητό του αν η θερμοκρασία ή η

υγρασία μέσα στην αποθήκη που διατηρεί κάποια τρόφιμα περάσει κάποια συγκεκριμένη τιμή.

Επίσης θα μπορούσε ακριβώς στη ίδια λογική το σύστημά μας να επηρεάζει τη λειτουργία κάποιων συσκευών προκειμένου να διορθώσει την ποιότητα του αέρα μέσα σε ένα χώρο. Για παράδειγμα στη βάση του Διαδικτύου των Πραγμάτων, θα μπορούσε κάποιος χρήστης να ορίσει ότι όταν η υγρασία μέσα στο διαμέρισμα τον οποίο κατοικεί περάσει μια συγκεκριμένη τιμή, τότε να ενεργοποιείται ένας αφυγραντήρας μέχρις ότου η υγρασία πέσει σε κάποια χαμηλή τιμή. Αυτό ίσως προϋποθέτει συσκευές που συνδέονται στο διαδίκτυο και μπορούν να διαχειριστούν μέσω αυτού.

6. Βιβλιογραφεία

[1] Internet of Things(IoT) by Ivy Wigmore 2014
http://whatis.techtarget.com/definition/Internet-of-Things

[2] "Sensors for Measurement and Control" by Peter Elgar
https://docs.google.com/document/d/19DqwKX7UEslqiIvtvcM9RacX1QGv6g6F74wkMehmAE8/edit

[3] "BENEFITS AND DRAWBACKS OF CLOUD-BASEDVERSUS TRADITIONAL ERP SYSTEMS" by Jiaqi Duan, Parwiz Faker, Alexander Fesak and Tim Stuart
http://www.academia.edu/2777755/Benefits_and_Drawbacks_of_Cloud-Based_versus_Traditional_ERP_Systems

[4] "The definition of Cloud Computing" by National Institute of Standards and Technology
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[5] "NIST Working Definition of Cloud Computing" by Peter Mell and Tim Grance
http://www.cubegeek.com/2009/06/the-nist-cloud-definition.html
[6] http://www.keystonesandrivets.com/kar/2008/06/Cloud-computing.html by Paul Wallis 2008
[7] "Cloud Interchangeability - Redefining Expectations" by André Monteiro, Joaquim Sousa Pinto, Claudio Teixeira and Tiago Batista
http://www.researchgate.net/publication/220865624_Cloud_Interchangeability_-_Redefining_Expectations

[8] "5 Key Events in the history of Cloud Computing" by Kaushik Raghupathi
http://architects.dzone.com/news/5-key-events-history-Cloud?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+zones%2Fdotnet+113

[9] Cloud Computing - The New Frontier of Internet Computing by George Pallis
http://cgi.di.uoa.gr/~ad/MDE556/Papers/palis-ic10.pdf

[10] "The definition of Cloud Computing" by National Institute of Standards and Technology
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[11] "Cloud Computing discussion – Perspectives on Cloud Computing" by Dave Duden

http://www.nlc.org/documents/Utility%20Navigation/About%20NLC/SML/NLC-RISC/RISC-Cloud-Technology-2011-Staff-Conference.pdf

[12] “The Best Damn Server Virtualization Book Period” by Rogier Dittner and David Rule
http://books.google.gr/books?id=L4iSishz58EC&pg=PR1&lpg=PR1&dq=The+Best+Damn+Server+Virtualization+Book+Period+Kris+Buytaert&source=bl&ots=ZruPULrmlf&sig=jVZsbBGGzaNcnmPRrOJoTVASjRA&hl=el&sa=X&ei=fmUsVPu4DajnywOKjIHQCg&ved=0CDQQ6AEwAw#v=onepage&q=The%20Best%20Damn%20Server%20Virtualization%20Book%20Period%20Kris%20Buytaert&f=false

[13] “New approach to virtualizing x86s” by Chris Barclay 2006
http://www.networkworld.com/news/tech/2006/102306techupdate.html

[14] “A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project” by Charles David Graziano
http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=3243&context=etd

[15] “Service-Oriented Architecture (SOA): Concepts, Technology, and Design” by Thomas Erl
http://www.weibnc.com/wp-content/uploads/brkpdfs/Service-Oriented-Architecture-SOA-Concepts-Technology-and-Design-by-Thomas-Erl-Independent-View-Of-Soa.pdf

[16] “Cloud Computing and SOA” by Geoffrey Raines
http://www.mitre.org/sites/default/files/pdf/09_0743.pdf

[17] “WebSphere Business Monitor” - IBM
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.monitor.install.doc/data/rest_services.html

[18] “REST Architectural Elements and Constraints”
http://mrbool.com/rest-architectural-elements-and-constraints/29339

[19] “Representational State Transfer”
http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[20] http://catalogue.fi-ware.org/

[21] “Interact: Gesture Recognition in the Cloud” by Alexandros Preventis, Kostas Stravoskoufos, Stelios Sotiriadis and Euripides G.M. Petrakis

[22] http://www.fi-ware.org/about/

[23] “SPECIFICATIONS OF WEATHER STATION”
SPECIFICATIONS WEATHER STATION

[24] “Cloud Computing Architecture” by Jonathan Strickland
http://computer.howstuffworks.com/cloud-computing/cloud-computing1.htm

[25] “An Architecture Description Language for Dynamic Sensor-Based Applications” by Humberto Cervantes, Didier Donsez and Lionel Touseau
http://membres-liglab.imag.fr/PPerso/membres/donsez/pub/publi/ccnc08-wadl.pdf

[26] “BUILD WITH NETATMO - CREATE YOUR OWN PROJECT”
https://dev.netatmo.com/doc

[27] “Publish/Subscribe Context Broker - Orion Context Broker”
http://catalogue.fi-ware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[28] “JSON Storage GE – User and Programmer Guide” by Alexandros Preventis

[29] “Identity Management – KeyRock”
http://catalogue.fi-ware.org/enablers/identity-management-keyrock

[30] “Html Tutorial”
http://www.w3schools.com/html/

[31] “Learn to Code Advanced HTML & CSS – Responsive Web Design”
http://learn.shayhowe.com/advanced-html-css/responsive-web-design/