Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Πανεπιστήμιο Πατρών

Διπλωματική Εργασία

cBox ένα αποκεντρωμένο σύστημα για την κατανομή πόρων που επιτρέπουν την επικοινωνία ανταλλαγής κίνησης σε ετερογενή περιβάλλοντα

Συγγραφέας
Νικόλαος Παλαγγιάς

Επιβλέπων
Καθ. Παύλος Σπυράκης

Συνεπιβλέπων
Δρ. Ιωάννης Χατζηγιαννάκης

Πάτρα 2012

Περιεχόμενα

1. Εισαγωγή

Το *cBox* είναι ένα σύστημα που περιέχει μια συλλογή από υπηρεσίες, βιβλιοθήκες και εφαρμογές, οι οποίες μπορούν να χρησιμοποιηθούν για την κοινή χρήση πόρων, όπως η συνδεσιμότητα στο διαδίκτυο, με άλλα μέλη ενός ad-hoc δικτύου με διαφάνεια και ασφάλεια. Οι δυνατότητες του συστήματος περιλαμβάνουν δικτύωση με ανοχή στις καθυστερήσεις, χρήση cache για τις αιτήσεις που προορίζονται για το διαδίκτυο, και πλήρως αποκεντρωμένη λειτουργία. Ο σχεδιασμός του συστήματος εκμεταλλεύεται καθιερωμένες τεχνολογίες όπως το ZeroConf και το mDNS επιτρέποντας έτσι στις συσκευές να ανακαλύπτουν συμβατές υπηρεσίες. Επίσης, χρησιμοποιεί μια μεγάλη ποικιλία προϋπαρχόντων τεχνολογιών και συσκευών για να διευκολυνθεί η επικοινωνία.

1.1. Κίνητρο και σημασία

Τα τελευταία 20 χρόνια, έχει γίνει πλέον σαφές, ότι ο κόσμος έχει εισέλθει στην εποχή της πληροφορίας και πως όλοι θα πρέπει να έχουν την δυνατότητα να συμμετάσχουν σε αυτή. Ξεκινώντας από την Φινλανδία, η πρόσβαση στο διαδίκτυο, έχει αρχίσει να καθιερώνεται νομικά ως βασικό ανθρώπινο δικαίωμα. Δεν είναι λίγες οι χώρες στις οποίες το διαδίκτυο θεωρείται βασική υποδομή λειτουργίας ενός τόπου όπως οι αυτοκινητόδρομοι. Ωστόσο υπάρχουν περιπτώσεις χωρών, όπου δεν υπάρχει συνδεσιμότητα στο διαδίκτυο, είτε λόγω έλλειψης υποδομών είτε λόγω φυσικών καταστροφών. Επιπρόσθετα, η πρόσβαση στο διαδίκτυο σε κάποιες περιοχές λογοκρίνεται, με αποτέλεσμα την περιορισμένη πρόσβαση είτε ακόμη και την πλήρη αποκοπή από τους παρόχους διαδικτύου. Ακόμη, μεγάλος αριθμός χρηστών ανησυχεί για τα προσωπικά τους δεδομένα και αναγνωρίζουν την ανάγκη διασφάλισης του απορρήτου των δεδομένων. Σε μερικές περιπτώσεις η ανταλλαγή δεδομένων μέσω απευθείας καναλιών δεν προτιμάται από τον χρήστη διότι η ασφάλεια της κυρίαρχης υποδομής έχει διακινδυνευθεί. Σε αυτές τις περιπτώσεις, θα πρέπει να βρεθούν νέοι τρόποι δρομολόγησης των πακέτων, εκμεταλλεύοντας διάφορα κανάλια επικοινωνίας.

Στις 2 Φεβρουαρίου 2010 ο Eden Moglen, καθηγητής Νομικής στο Πανεπιστήμιο της Columbia, ανακοίνωσε στο ISOC ένα νέο project με το όνομα *FreedomBox*. Το *FreedomBox* είναι ένα community project για την ανάπτυξη, τον σχεδιασμό αλλά και την προώθηση, προσωπικών εξυπηρετητών στους οποίους εκτελείται ελεύθερο λογισμικό για κατανεμημένα κοινωνικά δίκτυα, ηλεκτρονικό ταχυδρομείο και οπτικοακουστική επικοινωνία. Είναι ένας προσωπικός εξυπηρετητής στον οποίο εκτελείται ένα ελεύθερο λειτουργικό σύστημα με ελεύθερες εφαρμογές που είναι σχεδιασμένες να δημιουργούν και να διαφυλάττουν το προσωπικό απόρρητο. Οι προγραμματιστές έχουν ως σκοπό τη δημιουργία και διαφύλαξη του προσωπικού απορρήτου παρέχοντας μια ασφαλή πλατφόρμα για την ανάπτυξη κοινωνικών δικτύων. Αυτό μπορεί να γίνει αναπτύσσοντας λογισμικό που έχει τη δυνατότητα να εκτελείται σε plug computers τα οποία μπορούν με ευκολία να τοποθετηθούν σε μεμονωμένες κατοικίες ή γραφεία. Προωθώντας μια αποκεντρωμένη ανάπτυξη υλικού, τα FreedomBox θα παρέχουν ιδιωτικότητα στην καθημερινή ζωή και ασφαλή επικοινωνία για ανθρώπους που επιθυμούν να διαφυλάξουν την ελευθερία τους σε καταπιεστικά καθεστώτα.

1.2. Στόχος

Οι βασικοί στόχοι του cBox είναι:

- ***Ασφαλής Κοινωνική Δικτύωση:*** αντικατάσταση των κοινωνικών δικτύων με κεντρικοποιημένες υπηρεσίες που σέβονται την ιδιωτικότητα.

- ***Λήψη Ασφαλούς Αντίγραφου Ασφαλείας:*** αυτόματη λήψη κρυπτρογραφημνένων αντίγραφων ασφαλείας στα γειτονικά FreedomBoxes.

- ***Δίκτυο Προστασίας της Ουδετερότητας:*** σε περιπτώσεις περιορισμού της σύνδεσης του διαδικτύου από τον πάροχο, εύρεση εναλλακτικό δρομολογίου για την αποφυγή των περιορισμών.

- ***Ασφαλής και Ανώνυμη Δημοσιοποίηση:*** αποστολή ανώνυμων μηνυμάτων εκτός του δικτύου των FreedomBoxes.

- ***Ασφάλεια Οικειακού Δικτύου:*** πραγματική προστασία από κακόβουλους εισβολείς.

- ***Κρυπτογραφημένη Ηλεκτρονική Αλληλογραφία:*** με δυνατότητα κρυπτογράφησης και αποκρυπτογράφησης.

- ***Ιδιωτική Φωνητική Επικοινωνία:*** δυνατότητα πραγματοποίησης κρυπτογραφημένης κλήσης voice-over-Internet μεταξύ των FreedomBoxes.

Βασιζόμενοι σε αυτούς τους φθηνούς, μικρούς και χαμηλής κατανάλωσης εξυπηρετητές αποφασίστηκε, η κατασκευή ενός συστήματος που θα περιλαμβάνει την πλειονότητα των παραπάνω στόχων, το *cBox*. Ένα σύστημα το οποίο σε περιοχές όπου η πρόσβαση στο διαδίκτυο είναι περιορισμένη λόγω φυσικών καταστροφών αλλά και εξαιτίας πολιτικών χειραγώγησης από εταιρείες παροχής διαδικτύου, θα μπορεί να παρέχει συνδεσιμότητα στο διαδίκτυο στις περιοχές αυτές. Επιπρόσθετα, με το ενσωματωμένο τείχος προστασίας, θα μπορεί να παρέχει σε κάθε χρήστη προστασία από κακόβουλους εισβολείς. Ακόμη, με την κρυπτογράφηση της πληροφορίας που θα ανταλλάσσουν οι συσκευές, θα διασφαλίζεται το προσωπικό απόρρητο. Κάθε συσκευή θα περιέχει έναν proxy server, ο οποίος θα παρέχει ανωνυμία στον χρήστη και στις πληροφορίες που αποστέλλει αυξάνοντας και την ταχύτητα και απόδοση του διαδικτύο με τη χρήση cache.

1.3. Συνεισφορά

Το cBox είναι ένα σύστημα λογισμικού το οποίο σχεδιάστηκε και αναπτύχθηκε για την υλοποίηση των παραπάνων στόχων μέσω ενός αξιόπιστου, ασφαλούς αλλά και αποδοτικού τρόπου. Στηριζόμενο στην κατασκευή αλλά και συντήρηση ενός ομότιμου δικτύου μεταξύ έμπιστων κόμβων που είναι πρόθυμοι να συμμετάσχουν, το λογισμικό cBox μπορεί να εκτελεστεί σε μεγάλη ποικιλία συσκευών δίχως εξειδικευμένες απαιτήσεις υλικού, για παράδειγμα συσκευές βασισμένες στο λειτουργικό σύστημα Android, προσωπικούς υπολογιστές, ή μικρές χαμηλής κατανάλωσης συσκευές όπως BeagleBoard [BeagleBoard, 2011], Dreamplug [GlobalScale, 2011]. Ανθεκτικά πρωτόκολλα δρομολόγησης, διασφαλίζουν την υψηλή αξιοπιστία εσωτερικά του δικτύου, όπου κόμβοι μπορούν να προσθεθούν ή να αφαιρεθούν και να λειτουργήσουν ως Internet Gateways.

Το cBox αναπτύσσει διαφορετικούς προσαρμογείς για να διευκολύνει την επικοινωνία βασιζόμενο στο υλικό στον οποίο εκτελείται. Όταν εκτελείται σε μια συσκευή η οποία εμπεριέχει έναν προσαρμογέα Bluetooth, το λογισμικό του cBox, πραγματοποιεί οποιαδήποτε επικοινωνία χρησιμοποιώντας τη συγκεκριμένη διεπαφή επικοινωνίας. Αν το λογισμικό του cBox εκτελείται σε μια πιο ισχυρή συσκευή η οποία υποστηρίζει διεπαφές όπως το WiFi, το Ethernet και το Bluetooth, τότε το λογισμικό θα χρησιμοποιήσει και τις τρεις διεπαφές καταλλήλως. Οι υπηρεσίες και οι συνδέσεις που παρέχονται από τους κόμβους που συμμετέχουν, εξαρτώνται πλήρως από τις προτιμήσεις των χρηστών και είναι προσαρμόσιμες.

1.4. Οργάνωση της Εργασίας

Το παρών κείμενο αποτελεί τεχνική αναφορά του έργου ***cBox***. Στη συνέχεια γίνεται αναφορά του τρόπου οργάνωσης του κειμένου της εργασίας.

Στο κεφάλαιο 2 γίνεται παρουσίαση των τεχνολογιών του Παγκόσμιου Ιστού που χρησιμοποιήθηκαν για την ανάπτυξη του λογισμικού που εμπεριέχεται στο ***cBox*****,** έχοντας ως βασικό στόχο την ταχύτητα, την σταθερότητα και την ασφάλεια.

Στο κεφάλαιο 3 αναλύεται η αρχιτεκτονική του συστήματος ***cBox*** επεξηγώντας με λεπτομέρεια τα επίπεδα που συνιστούν την εφαρμογή που εκτελείται στο ***cBox***.

Στο κεφάλαιο 4 αναφέρονται οδηγίες προς τους προγραμματιστές που συντάχθηκαν κατά την διάρκεια της ανάπτυξης λογισμικού και έχουν ως μοναδικό στόχο την διευκόλυνση πρός τρίτους για την επέκταση του λογισμικού που περιλαμβάνεται στο ***cBox***.

Στο κεφάλαιο 5 συντάσσεται ο επίλογος της εργασίας, στον οποίο περιγράφεται η μελλοντική εργασία που μπορεί να πραγματοποιηθεί, τα εργαλεία που χρησιμοποιήθηκαν, ο ιστόπος στον οποίο μπορεί να βρεθεί ο πηγαίος κώδικας αλλα και οι ευχαριστίες.

Τέλος στο κεφάλαιο 6 γίνεται αναφορά της βιβλιογραφίας που χρησιμοποιήθηκε για να καταστήσει δυνατή την υλοποίηση του παρόντος έργου.

2. Σχετικές Τεχνολογίες Παγκόσμιου Ιστού

2.1. Twitter API

Το *Twitter* αποτελεί μια υπηρεσία κοινωνικής δίκτυωσης και μια υπηρεσία micro blogging. Δίνει την δυνατότητα στους χρήστες του να αποστέλλουν και να λαμβάνουν μηνύματα κειμένου, μήκους έως 140 χαρακτήρων γνωστά και ως “tweets”. Δημιουργήθηκε τον Μάρτιο του 2006 από τον Jack Dorsey και ξεκίνησε να λειτουργεί τον Ιούλιο του 2006. Η υπηρεσία έλαβε ταχύτατα μεγάλη δημοσιότητα, έχοντας πάνω από 140 εκατομμύρια ενεργούς χρήστες το 2012 και πάνω από 340 εκατομμύρια tweets καθημερινά. Οι μη εγγεγραμμένοι χρήστες έχουν την δυνατότητα ανάγνωσης των tweets ενώ οι εγγεγραμμένοι χρήστες έχουν την δυνατότητα αποστολής tweets μέσω της διεπαφής που παρέχεται από την ιστοσελίδα ή μέσω εφαρμογών τρίτων. Η ιστοσελίδα του Twitter είναι μια από τις δέκα με την μεγαλύτερη επισκεψιμότητα στο διαδίκτυο. Το *Twitter* έχει αναφερθεί ως ένας πολύ σημαντικός παράγοντας στην Arab Spring και σε άλλες πολιτικές διαμαρτυρίες. Το γεγονός αυτό, αποτέλεσε ένα πολύ σημαντικό κριτήριο, για το οποίο επιλέχθηκε το Twitter ως πρώτη εφαρμογή για το *cBox*, αν ληφθεί υπόψιν η φιλοσοφία που βρίσκεται πίσω από το FreedomBox και τον δημιουργό του.

2.1.1. Επισκόπιση

Το 2006, μαζί με την εκκίνηση της λειτουργίας του Twitter, δόθηκε και το *Twitter API* (Application Programming Interface), το οποίο δίνει τη δυνατότητα σε τρίτους προγραμματιστές, να αναπτύσσουν εφαρμογές που χρησιμοποιούν τις λειτουργίες του Twitter. Τα τελευταία δύο χρόνια ωστόσο, έχει δοθεί ιδιαίτερη προσοχή στην ανάπτυξη του *Twitter API*, ώστε να δοθεί στους προγραμματιστές μια σταθερή και αξιόπιστη προγραμματιστική διεπαφή.

Το Twitter παρέχει:

- ***Twitter for Websites***: αποτελεί ένα πακέτο προϊόντων που δίνει τη δυνατότητα σε ιστοσελίδες να ενσωματώσουν εύκολα και γρήγορα το Twitter. Μια πολύ γνωστή και συχνά χρησιμοποιούμενη λειτουργία αποτελεί το Twitter Button και το Follow Button που βρίσκονται στην πλειοψηφία των ιστοσελίδων σήμερα.

- ***Search API***: αποτελεί ένα API το οποίο σχεδιάστηκε για προϊόντα, τα οποία δίνουν την δυνατότητα στον χρήστη να θέσει ερωτήματα (queries) στα δεδομένα του Twitter. Ένα παράδειγμα χρησιμοποίησης αυτού του API συνιστά, η εύρεση των tweets που περιέχουν το ερώτημα που έθεσε ο χρήστης, η εύρεση των tweets που πραγματοποιήθηκαν από κάποιο συγκεκριμένο χρήστη.

- ***REST API***: αποτελεί ένα API το οποίο δίνει τη δυνατότητα στον προγραμματιστή να έχει πρόσβαση στον πυρήνα του Twitter. Ο πυρήνας του Twitter παρέχει λειτουργίες όπως είναι το Homeline (παρουσίαση των tweets ενος χρήστη), η αποστολή Status update (Αποστολή tweet), εύρεση πληροφοριών ενός χρήστη κ.α.

- ***Streaming API***: αποτελεί ένα API το οποίο εκτελεί πραγματικού χρόνου δειγματοληψία του Twitter Firehose. Απευθύνεται κυρίως στους προγραμματιστές που έχουν ανάγκη την εντατική εξόρυξη δεδομένων για λόγους έρευνας ή στατιστικής ανάλυσης.

2.1.2. Θέματα που αφορούν τους προγραμματιστές

Για την ανάπτυξη μιας εφαρμογής η οποία, χρησιμοποιεί στην υλοποίηση της το Twitter API θα πρέπει να ληφθούν κάποια στοιχεία υπόψιν. Παρακάτω αναφέρονται τα βασικότερα στοιχεία, τα οποία ήταν απαιτούμενα και για την υλοποίηση της εφαρμογής του cBox.

- ***Το Twitter API αποτελείται από τρία APIs***. Όπως αναφέρθηκε παραπάνω αποτελείται από τρία APIs, ωστόστο ο μεγαλύτερος αριθμός των εφαρμογών χρησιμοποιεί το REST API για την υλοποίηση όλων των βασικών λειτουργιών της εφαμογής. Για την εκτέλεση ερωτημάτων στα δεδομένα του Twitter χρησιμοποιήθηκε το Search API.

- ***Το Twitter API είναι απόλυτα βασισμένο σε HTTP***. Θα πρέπει να δοθεί ιδιαίτερη προσοχή στην απάντηση που αποστέλλεται από το Twitter, για την ανάγνωση των κωδικών λαθών.
  - Για την πραγματοποίηση αιτημάτων στο Twitter, τα οποία έχουν σκοπό την λήψη δεδομένων, αποστέλλεται ένα ***HTTP GET*** αίτημα.
  - Τα αιτήματα που έχουν σκοπό την υποβολή, την αλλαγή και την καταστροφή δεδομένων, χρησιμοποιούν ένα ***HTTP POST*** αίτημα.
  - Για την καταστροφή δεδομένων είναι δυνατή η αποστολή ενός ***HTTP DELETE*** αίτημα.

- ***Το Twitter API είναι RESTful***. Το Twitter API συμμορφώνεται με τις αρχές σχεδίασης του Representational State Transfer (REST). Μέχρι τη στιγμή του εγγράφεται το εν λόγω κείμενο, οι μορφότυποι δεδομένων που υποστηρίζονται είναι XML, JSON, RSS, Atom.

- ***Οι παράμετροι έχουν ορισμένους κανόνες***.
  - Οι παράμετροι θα πρέπει να μετατραπούν στην κωδικοποίηση UTF-8 και να είναι URL κωδικοποιημένα εφόσον το API είναι βασισμένο εξ' ολοκλήρου σε HTTP.
  - Οι παράμετροι της σελίδας ξεκινούν πάντα με 1.

- ***Υπάρχουν όρια σελιδοποίησης***. Όταν πραγματοποιείται ένα αίτημα στο Twitter, για την αύξηση της απόδοσης και της μείωση του φόρτου στους εξυπηρετητές, έχει εφαρμοστεί η λογική της σελιδοποίησης. Ο προγραμματιστής ορίζει πόσα αποτελέσματα θέλει να του επιστραφούν και συνέχεια μπορεί με κάποιο συμβάν να ζητήσει να του επιστραφεί η επόμενη σελίδα. Με τον τρόπο αυτό, μειώνεται η κίνηση στο δίκτυο πράγμα πολύ σημαντικό ειδικά σε εφαρμογές κινητών τηλεφώνων. Στο ***REST API*** το θεωρητικό όριο βρίσκεται στα 3200 statuses ανά σελίδα και στο ***Search API*** το θεωρητικό όριο βρίσκεται στα 1500 statuses ανά σελίδα.

2.1.3. Βιβλιοθήκες για το Twitter API σε διάφορες γλώσσες προγραμματισμού

Η βιβλιοθήκη στις γλώσσες προγραμματισμού αποτελεί ένα πολύ σημαντικό εργαλείο, που βοηθά τους προγραμματιστές ώστε να επαναχρησιμοποιούν τμήματα κώδικα που έχουν ήδη δημιουργήσει και χρησιμοποιήσει στο παρελθόν. Το γεγονός αυτό βοηθά έτσι ώστε να μην χρειάζεται να ξανανακαλύψει τον τροχό, να μειώσει τον χρόνο ανάπτυξης του λογισμικού, να μειώσει το κόστος της εφαρμογής κ.α. Έτσι από τη στιγμή που δόθηκε από το Twitter το συγκεκριμένο API, πολλοί προγραμματιστές έσπευσαν να δημιουργήσουν βιβλιοθήκες για διάφορες γλώσσες προγραμματισμού οι οποίες αξιοποιούν το API αυτό. Ένα μικρό δείγμα αναφέρεται παρακάτω με τις βιβλιοθήκες που υπάρχουν σε γλώσσες προγραμματισμού:

C++			
• kQOAuth	• libOAuth	• QTweetLib	• Twitcurl

Java		
• Scribe	• Twitter4J	• Twitter API ME

.NET			
• LINQ to Twitter	• Hammock	• Twitterizer	• TweetSharp

Objective – C / Cocoa			
• RSOAuth Engine	• MGTwitterEngine	• MPOAuth	• ShareKit

PHP			
• Oauth-php	• TmhOAuth	• TwitterOAuth	• Twitter async

Python			
• Tweepy	• Twython	• Python Twitter	• Twitty Twister

2.1.4. Θέματα ασφάλειας στο Twitter API

Με το πέρασμα του χρόνου η δημοσιότητα του Twitter αυξήθηκε εκθετικά, με αποτέλεσμα να προκύψουν προβλήματα ασφάλειας. Όντας τόσο δημοφιλές, το Twitter άρχισε να τραβά την προσοχή διαφόρων κακοπροαίρετων hackers, οι οποίοι προσπάθησαν να εκμεταλλευτούν τα δεδομένα αλλά και τις δυνατότητες που παρέχονταν από το Twitter API. Έτσι το Twitter αναγκάστηκε να αυξήσει την ασφάλεια στο API που παρέχει, ώστε να μειωθούν και αν είναι δυνατόν να εξαλειφθούν τα συμβάντα όπου κακοπροαίρετοι χρήστες να έχουν πρόσβαση σε λογαριασμούς που δεν έχουν το δικαίωμα.

Ο πρώτος τρόπος που χρησιμοποιήθηκε για την σύνδεση σε ένα λογαριασμό, ήταν η εισαγωγή ενός ονόματος χρήστη και ενός κωδικού, ούτως ώστε να είναι δυνατή η ταυτοποίηση του χρήστη. Για την διευκόλυνση του χρήστη, συνήθως παρέχεται η δυνατότητα αποθήκευσης των διαπιστευτηρίων από την εφαρμογή, ώστε να μην χρειάζεται ο χρήστης να τα εισάγει κάθε φορά που θέλει να εισέλθει στην εφαρμογή. Ωστόσο ένα αρχικό πρόβλημα προέκυψε όταν η εφαρμογή αποθήκευε τα διαπιστευτήρια και κακόβουλοι χρήστες τα ανακτούσαν με αποτέλεσμα να αποκτούν πρόσβαση σε λογαριασμούς άλλων χρηστών. Μια λύση που προτάθηκε, ήταν η κρυπτογράφηση των διαπιστευτηρίων κατά την αποθήκευση τους από την εφαρμογή. Με το πέρασμα του χρόνου και την εύρεση νέων ασφαλέστερων μεθόδων, η διαδικασία αυτή σύνδεσης έχει ξεπεραστεί (deprecated), ωστόσο υποστηρίζεται ακόμα.

Ο πιο κοινός και αποδεκτός τρόπος επικοινωνίας με το Twitter αποτελεί η χρήση του API μέσω SSL. Η τεχνολογία SSL συνιστά ένα ασφαλές πρωτόκολλο που επιτρέπει την αλληλεπίδραση με το Twitter χωρίς το φόβο παρεμβολής τρίτων. Για τις εφαρμογές χρήστη θα πρέπει να γίνεται χρήση του SSL για ασφαλείς συναλλαγές μεταξύ του χρήστη και του Twitter. Ειδικότερα θα πρέπει να γίνεται χρήση του SSL σε συνδυασμό με το OAuth. Η μόνη περίπτωση κατά την οποία είναι πιθανό να μην υπάρξει κάποιο όφελος από την χρήση του SSL είναι όταν υπάρχει επικοινωνία υπηρεσία με υπηρεσία. Αν η υπηρεσία που επικοινωνεί με το Twitter είναι της εμπιστοσύνης του προγραμματιστή τότε είναι δυνατή η παράλειψη του SSL.

2.1.5. Twitter4J

Όπως αναφέρθηκε παραπάνω, για την πλειοψηφία των χρησιμοποιούμενων γλωσσών προγραμματισμού έχουν υλοποιηθεί διάφορες βιβλιοθήκες που εκμεταλλεύονται το Twitter API και παρέχουν μια διεπαφή για την ευκολότερη ανάπτυξη εφαρμογών που λαμβάνουν υπόψιν τους το συγκεκριμένο API. Η ανάπτυξη εφαρμογών στο Android γίνεται με τη βοήθεια της γλώσσας προγραμματισμού JAVA. Έτσι θα έπρεπε να γίνει επιλογή της βιβλιοθήκης μέσα από αυτές που προαναφέρθηκαν. Η δημοφιλέστερη βιβλιοθήκη στη γλώσσα προγραμματισμού JAVA είναι η βιβλιοθήκη *Twitter4J* η οποία υλοποιήθηκε από τον Ιάπωνα προγραμματιστή Yusuke Yamamoto.

Στον ιστότοπο της βιβλιοθήκης παρουσιάζονται απλά παραδείγματα για τη χρήση της βιβλιοθήκης. Η βιβλιοθήκη εκμεταλλεύεται, παρόλο που δεν είναι επίσημη, όλο το εύρος του Twitter API. Δίνει τη δυνατότητα στον προγραμματιστή, εύκολα, να εκτελέσει βασικές και ειδικές λειτουργίες. Σε ελάχιστες γραμμές κώδικα, δίνεται η δυνατότητα στον προγραμματιστή, να αποστείλει κάποιο tweet, να λάβει το Timeline, να στείλει/λάβει απευθείας μηνύματα, να πραγματοποιήσει αναζήτηση για κάποιο tweet, να πραγματοποιήσει έλεγχο σελιδοποίησης, να διαχειριστεί το πρωτόκολλο OAuth κ.α.

Όλες οι ενέργειες πραγματοποιούνται μέσω ενός στιγμιοτύπου της κλάσης Twitter. Αφού ο χρήστης συνδεθεί στο Twitter με την χρήση του πρωτοκόλλου OAuth, μπορεί να πραγματοποιήσει οποιαδήποτε λειτουργία παρέχεται από το Twitter API. Ιδιαίτερη προσοχή θα πρέπει να δοθεί κατά την εκτέλεση επερώτησης στα δεδομένα του Twitter. Στην περίπτωση αυτή, γίνεται χρήση του Search API, και χρησιμοποιούνται ειδικές δομές όπως η κλάση JQuery για να πραγματοποιηθεί η επερώτηση στα δεδομένα του Twitter.

Το *Twitter4J* περιέχει ειδικές δομές, χρησιμοποιώντας την σχεδιαστική μέθοδο Factory Pattern, οι οποίες κατασκευάζουν, αλλά και πραγματοποιούν τις HTTP ενέργειες που απαιτεί το Twitter API. Ο προγραμματιστής που χρησιμοποιεί τη συγκεκριμένη βιβλιοθήκη, δεν απαιτείται να κατασκευάσει προγραμματιστικά την αλληλουχία αλφαριθμητικών η οποία πραγματοποιεί τις HTTP ενέργειες, ούτε χρειάζεται να χειριστεί τις απαντήσεις που δέχεται από τους εξυπηρετητές. Στο στιγμιότυπο της κλάσης που δημιουργείται, εισάγονται τα διαπιστευτήρια, που έχουν αποδοθεί στον συγκεκριμένο λογαριασμό και στη συνέχεια πραγματοποιούνται οι ενέργειες που θέλει ο προγραμματιστής με ειδικές μεθόδους. Για την αποστολή ενός tweet για παράδειγμα, ο προγραμματιστής αφού έχει εισάγει τα διαπιστευτήρια, δίνει στην κατάλληλη μέθοδο ως όρισμα το tweet που επιθυμεί και καλεί τη μέθοδο αυτή. Η αποστολή του tweet καλύπτεται από τη βιβλιοθήκη. Η διαδικασία συνιστά τη δημιουργία του HTTP POST σύμφωνα με τα προσυμφωνημένα πρότυπα, η πραγματοποίηση του HTTP POST, η λήψη του HTTP response. Τελικά επιστρέφεται στον προγραμματιστή μια μεταβλητή κατάστασης και σε περίπτωση που υπήρξε κάποιο πρόβλημα προκαλείται μια εξαίρεση της βιβλιοθήκης. Ένα σημαντικό γεγονός είναι η απόκρυψη τους χειρισμού των HTTP ενεργειών που περιλαμβάνει και τον χειρισμό του OAuth.

Θα πρέπει να τονιστεί, ότι επειδή οι βιβλιοθήκες για το Twitter API είναι ανεπίσημες και έχουν αναπτυχθεί από τρίτους, δεν υπάρχει συμβατότητα των δεδομένων σε διαφορετικές γλώσσες προγραμματισμού και σε διαφορετικές βιβλιοθήκες. Αυτό σημαίνει πως, σε περίπτωση που επικοινωνούν δυο εφαρμογές που εμπεριέχουν βιβλιοθήκες για το Twitter API, ακόμα και της ίδια γλώσσας προγραμματισμού, δεν υπάρχει συμβατότητα μεταξύ των κλάσεων των διαφορετικών βιβλιοθηκών, παρόλο που είναι πιθανό να χρησιμοποιούν την ίδια ονοματολογία.

2.2. OAuth

Το OAuth αποτελεί ένα ανοιχτό πρότυπο για ταυτοποίηση χρηστών. Το πρότυπο αυτό επιτρέπει στους χρήστες να μοιράζουν τους προσωπικούς τους πόρους, που βρίσκονται αποθηκευμένοι σε ένα διαδικτυακό τόπο, με έναν άλλο διαδικτυακό τόπο χωρίς να χρειάζεται να παραδώσουν τα διαπιστευτήρια τους. Αυτό επιτρέπει σε ένα χρήστη, να παραχωρήσει σε μια third party εφαρμογή, τη δυνατότητα να έχει πρόσβαση στις πληροφορίες του μέσω ενός διαφορετικού παρόχου υπηρεσιών, χωρίς να απαιτείται η διαμοίραση των δικαιωμάτων ή των δεδομένων του σε πλήρη έκταση.

Στις 4 Δεκεμβρίου 2007 δημοσιεύτηκαν οι προδιαγραφές του OAuth Core 1.0 (γνωστό και ως RFC 5849), αναθεωρήθηκαν στις 24 Ιουνίου 2009 και οριστικοποιήθηκαν τον Απρίλιο του 2010. Το πρότυπο αυτό αποτελεί μια από γρηγορότερα αναπτυσσόμενες προδιαγραφές. Αποτελεί μια αναγκαία λύση για το πρόβλημα της ταυτοποίησης χρηστών στις web εφαρμογές που απαιτούν απομακρυσμένη πρόσβαση. Το OAuth 1.0 αντικαθίσταται από το καινούριο πρωτόκολλο OAuth 2.0 από το IETF.

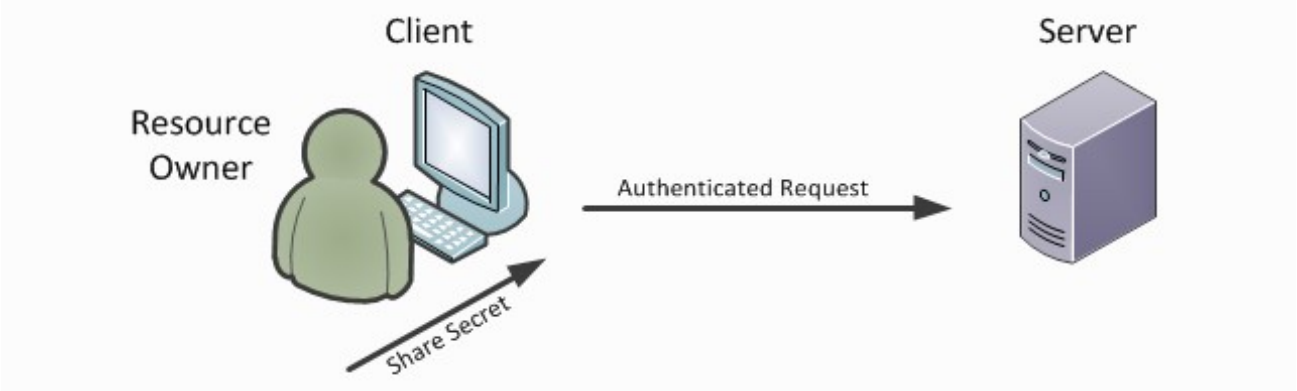
Καθώς το web μεγαλώνει, όλο και περισσότεροι ιστότοποι στηρίζονται σε κατανεμημένες υπηρεσίες και στο υπολογιστικό νέφος. Το πρόβλημα, έγκειται στο γεγονός ότι, οι εφαρμογές αυτές για να έχουν πρόσβαση στα δεδομένα των χρηστών στους ιστότοπούς τους, ζητούν από τον χρήστη το όνομα χρήστη και τον κωδικό πρόσβασης. Η μέθοδος αυτή απαιτεί την αποκάλυψη των διαπιστευτηρίων σε κάποιον τρίτο. Έχοντας κανείς υπόψιν του, την τάση των ανθρώπων να επαναχρησιμοποιούν τους ίδιους κωδικούς πρόσβασης σε διαφορετικές εφαρμογές στη ζωή τους, μπορεί κανείς εύκολα να συμπεράνει ότι η αποκάλυψη ενός κωδικού μπορεί να θέσει σε κίνδυνο πολλές εφαρμογές. Ακόμη η μέθοδος αυτή δίνει πλήρη προσβασιμότητα στον χρήστη σε μία εφαρμογή, χωρίς να τον περιορίζει σε ορισμένες λειτουργίες. Το πρωτόκολλο OAuth παρέχει στον χρήστη τη δυνατότητα να παραχωρήσει την πρόσβαση στους πόρους τους, σε third party εφαρμογές, χωρίς να απαιτείται η διαμοίραση των διαπιστευτηρίων τους. Ακόμη παρέχει τη δυνατότητα της περιορισμένης πρόσβασης, δηλαδή να είναι δυνατές μόνο μερικές λειτουργίες απο την third party εφαρμογή.

2.2.1. Ορολογία του OAuth

Το πρωτόκολλο OAuth ορίζει τρεις σαφείς και διαφορετικούς ρόλους: τον πελάτη (client), τον εξυπηρετητή (server) και τον ιδιοκτήτη των πόρων (resource owner). Αυτοί οι τρείς ρόλοι βρίσκονται σε κάθε OAuth συναλλαγή· σε μερικές περιπτώσεις ο πελάτης είναι και ιδιοκτήτης πόρων. Η πρωτότυπη έκδοση της προδιαγραφής χρησιμοποιούσε ένα διαφορετικό σύνολο όρων σχετικά με αυτούς τους ρόλους: καταναλωτής (consumer – client), πάροχος υπηρεσίας (service provider – server) και χρήστης (user – resource owner).

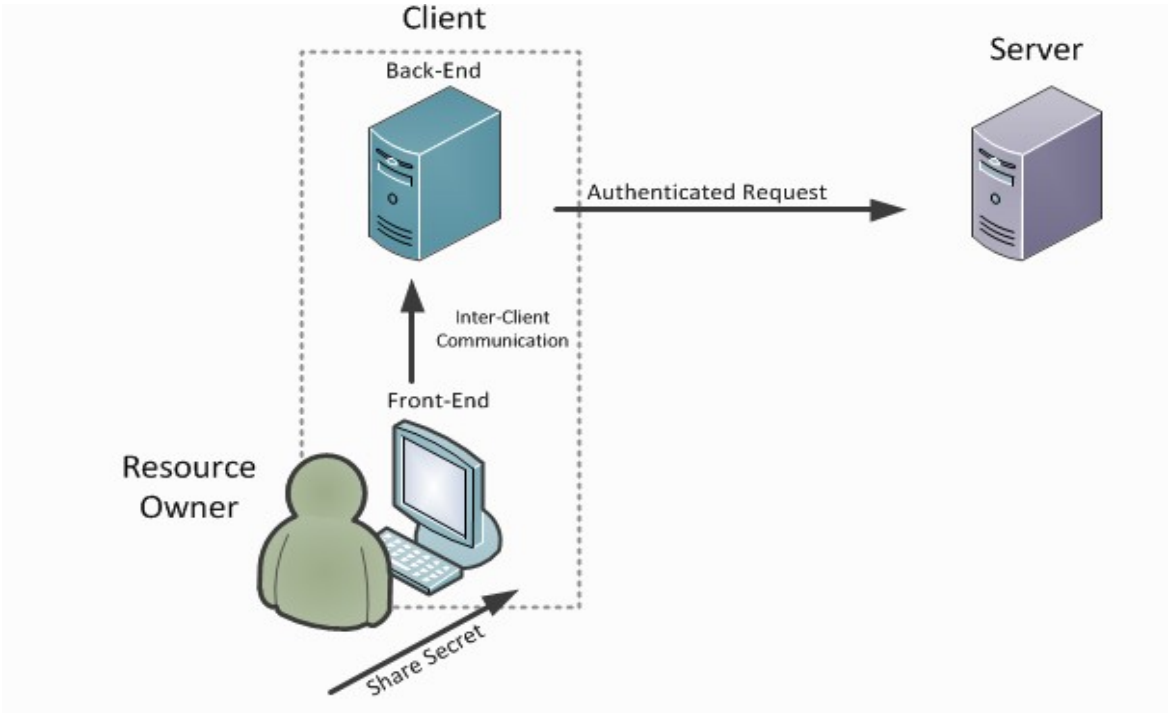
Στο παραδοσιακό πελάτης – εξυπηρετητής μοντέλο ταυτοποίησης χρήστη, ο πελάτης χρησιμοποιεί τα διαπιστευτήρια του για να έχει πρόσβαση στους πόρους του που φιλοξενούνται στον εξυπηρετητή. Σχετικά με τον εξυπηρετητή, το κοινό μυστικό που χρησιμοποιείται από τον πελάτη, ανήκει στον πελάτη. Ο εξυπηρετητής μπορεί μόνο να διακρίνει αν τα διαπιστευτήρια που του δίνονται από τον πελάτη είναι σωστά. Δεν μπορεί να διακρίνει αν κάποιος τρίτος χρησιμοποιεί τα διαπιστευτήρια ενός χρήστη.

Δεν είναι λίγες οι φορές όπου κάποιος τρίτος έχει στη διάθεση του τα διαπιστευτήρια ενός πελάτη και χρησιμοποιώντας τα, αποκτά πρόσβαση στα δεδομένα του πελάτη, με τον πελάτη να βρίσκεται σε πλήρη άγνοια. Όταν εμπλέκεται κάποιος τρίτος, τυπικά ένας χρήστης αλληλεπιδρά με τον πελάτη, ο πελάτης λειτουργεί εκ μέρους του χρήστη. Σε αυτές τις περιπτώσεις ο πελάτης, δεν έχει πρόσβαση στους πόρους του, αλλά στους πόρους του ιδιοκτήτη πόρων. Αντι να χρησιμοποιεί τα διαπιστευτήρια του πελάτη, ο πελάτης χρησιμοποιεί τα διαπιστευτήρια του ιδιοκτήτη πόρων για να πραγματοποιεί τα ερωτήματα, παριστάνοντας τον ιδιοκτήτη πόρων. Τα διαπιστευτήρια του χρήστη συνήθως αποτελούνται από ένα όνομα χρήστη και έναν κωδικό πρόσβασης, αλλά οι ιδιοκτήτες πόρων δεν περιορίζονται στους χρήστες, μπορεί να αποτελούν οποιαδήποτε οντότητα που ελέγχει τους πόρους του εξυπηρετητή.



Το μοντέλο γίνεται πιο λεπτομερές όταν ο πελάτης συνίσταται από μια web-based εφαρμογή. Σε αυτή την περίπτωση, ο πελάτης διαχωρίζεται μεταξύ ενός front-end συστατικό, που συνήθως εκτελείται μέσα σε έναν φυλλομετρητή web στο desktop του ιδιοκτήτη πόρων, και ένα back-end συστατικό που συνήθως εκτελείται στον εξυπηρετητή του πελάτη.

Ο ιδιοκτήτης πόρων αλληλεπιδρά με ένα από τα μέρη της εφαρμογής του πελάτη καθώς ο εξυπηρετητής λαμβάνει αιτήσεις από ένα άλλο μέρος. Ωστόσο, αδιαφορώντας για την εσωτερική αρχιτεκτονική που χρησιμοποιεί ο πελάτης, και πάλι λειτουργεί σαν μια οντότητα, από μέρους του ιδιοκτήτη πόρων.



Ο *προστατευόμενος πόρος* είναι ο πόρος που αποθηκεύεται στον εξυπηρετητή και ο οποίος απαιτεί ταυτοποίηση για να έχει κανείς πρόσβαση σε αυτόν. Οι προστατευόμενοι πόροι ανήκουν ή ελέγχονται από τον ιδιοκτήτη πόρων. Καθένας που ζητάει να έχει πρόσβαση σε έναν προστατευόμενο πόρο πρέπει να του επιτραπεί η είσοδος, μόνο από τον ιδιοκτήτη των πόρων.

Ο αριθμός των σκελών (legs) χρησιμοποιούνται για να περιγράψουν μια OAuth αίτηση. Ειδικότερα αναφέρεται στον αριθμό των τμημάτων που συμπεριλαμβάνονται. Για παράδειγμα αν ο πελάτης είναι και ο ιδιοκτήτης πόρων, τότε περιγράφεται ως 2-σκελών (2-legged).

Το OAuth χρησιμοποιεί τριών ειδών διαπιστευτήρια:

- *Κλειδί και Μυστικό Καταναλωτή (Consumer Key and Secret / Client Credentials):*

  Χρησιμοποιείται για την ταυτοποίηση του πελάτη. Επιτρέπει στον εξυπηρετητή να συλλέξει πληροφορίες για τους πελάτες χρησιμοποιώντας τις υπηρεσίες του, προσφέροντας σε μερικούς πελάτες ειδική μεταχείριση. Σε μερικές περιπτώσεις, δεν αποτελεί έμπιστο κριτήριο και χρησιμοποιείται μόνο για τη συλλογή πληροφοριών.

- *Τεκμήριο και Μυστικό Αίτησης (Request Token and Secret / Temporary Credentials):*

  Χρησιμοποιείται για την εξακρίβωση της αίτησης ταυτοποίησης. Δίνει τη δυνατότητα σε διαφορετικού είδους εφαρμογές να φιλοξενούνται, παρέχοντας ευελιξία και ασφάλεια.

- *Τεκμήριο και Μυστικό Πρόσβασης (Access Token and Secret / Token Credentials):*

  Χρησιμοποιείται στη θέση του ονόματος χρήστη και του κωδικού πρόσβασης. Αντι να χρειάζεται ο ιδιοκτήτης πόρων να διαμοιραστεί τα διαπιστευτήρια με τον πελάτη, επιτρέπει στον εξυπηρετητή να εκδόσει μια ειδική κλάση διαπιστευτηρίων στον πελάτη, τα οποία αναπαριστούν την παραχώρηση πρόσβασης στον πελάτη από τον ιδιοκτήτη πόρων. Ο πελάτης χρησιμοποιεί τα τεκμηριωμένα διαπιστευτήρια για να έχει πρόσβαση στους πόρους χωρίς να γνωρίζει τον κωδικό του ιδιοκτήτη πόρων. Ακόμη συμπεριλαμβάνει ένα αναγνωριστικό τεκμήριο, συνήθως μια τυχαία αλληλουχία αλφαριθμητικών που είναι μοναδική, δύσκολη να μαντευθεί, και η οποία έχει ζευγαρωθεί με ένα μυστικό για να προστατέψει το τεκμήριο από το να χρησιμοποιηθεί από μη εξουσιοδοτημένους χρήστες. Τέλος είναι περιορισμένα σε εύρος και διάρκεια και μπορούν να ανακαλεστούν ανα πάσα στιγμή.

2.2.2. Παράδειγμα OAuth

Η Άννα (ιδιοκτήτης πόρων) έχει πρόσφατα ανεβάσει κάποιες προσωπικές της φωτογραφίες από τις διακοπές της (προστατευμένοι πόροι) στον ιστότοπο διαμοίρασης φωτογραφιών “photos.tmp.gr” (εξυπηρετητής). Θέλει να χρησιμοποιήσει τον ιστότοπο “printer.tmp.gr” για να εκτυπώσει μία από αυτές τις φωτογραφίες. Η Άννα συνδέεται στον ιστότοπο “photos.tmp.gr” χρησιμοποιώντας τα διαπιστευτήρια της - όνομα χρήστη και κωδικό πρόσβασης. Ωστόσο δεν θέλει να κοινοποιήσει το όνομα χρήστη και τον κωδικό πρόσβασης στον ιστότοπο “printer.tmp.gr”, που απαιτούνται για να εκτυπώσει τη φωτογραφία που θέλει.

Ο ιστότοπος “printer.tmp.gr” έχει δηλώσει τον εαυτό του ως πελάτη στον ιστότοπο “photos.tmp.gr” και του έχει απαντήσει παρέχοντας του το κλειδί και το μυστικό του καταναλωτή:

Client Key	**Client Secret**
`dpf43f3p214k3103`	`kd94hf93k423kf44`

Αρχικά, πριν ο ιστότοπος “printer.tmp.gr” να ρωτήσει την Άννα αν μπορεί να έχει πρόσβαση στις φωτογραφίες της, θα πρέπει πρώτα να εγκαταστήσει ένα προσωρινό σύνολο διαπιστευτηρίων με τον ιστότοπο “photos.tmp.gr” για την αναγνώριση της αίτησης. Για να γίνει αυτό ο πελάτης στέλνει τον παρακάτω HTTPS POST στον εξυπηρετητή:

```
POST /initiate HTTP/1.1
Host: photos.tmp.gr
Authorization: OAuth realm=”Photos”,
    oauth_consumer_key=”dpf43f3p214k3103”,
    oauth_signature_method=”HMAC-SHA1”,
    oauth_timestamp=”137131200”,
    oauth_nonce=”wIjqoS”,
    oauth_callback=”http%3A%2F%2Fprinter.tmp.gr%2Fready”,
    oauth_signature=”74KNZJeDHnMBp0EMJ9ZHt%2FXKycU%3D”
```

Ο εξυπηρετητής επαληθεύει την αίτηση και απαντά με ένα σύνολο από προσωρινά διαπιστευτήρια στο σώμα μιας HTTP απάντησης:

```
HTTP/1.1 200 OK
Content-Type: application/x-www-form-urlencoded

oauth_token=hh5s93j4hdidpola&oauth_token_secret=hdhd0244k9j7ao03&
oauth_callback_confirmed=true
```

Ο πελάτης ανακατευθύνει το user-agent της Άννας στο σημείο ταυτοποίησης του ιδιοκτήτη πόρων του στον εξυπηρετητή για να λάβει την έγκριση της Άννας, για την πρόσβαση στις

φωτογραφίες της:

```
Https://photos.tmp.gr/authorize?oauth_token=hh5s93j4hdidpola
```

Ο εξυπηρετητής ζητά από την Άννα να συνδεθεί με το όνομα χρήστη και τον κωδικό πρόσβασης, αν επιτύχει, ζητά την έγκριση για την πρόσβαση του “printer.tmp.gr” στις προσωπικές της φωτογραφίες. Η Άννα αποδέχεται την αίτηση αυτή, ο user-agent της την ανακατευθύνει στο callback URI που παρέχεται από τον πελάτη για την προηγούμενη αίτηση:

```
Http://printer.tmp.gr/ready?
oauth_token=hh5s93j4hdidpola&oauth_verifier=hfdp7dh39dks9884
```

Η callback αίτηση ενημερώνει τον πελάτη ότι η Άννα ολοκλήρωσε την διαδικασία ταυτοποίησης. Ο πελάτης στη συνέχεια ζητά ένα σύνολο από τεκμηριωμένα διαπιστευτήρια χρησιμοποιώντας τα προσωρινά διαπιστευτήρια.

```
POST /initiate HTTP/1.1
Host: photos.tmp.gr
Authorization: OAuth realm=”Photos”,
    oauth_consumer_key=”dpf43f3p214k3103”,
    oauth_token=”hh5s93j4hdidpola”,
    oauth_signature_method=”HMAC-SHA1”,
    oauth_timestamp=”137131201”,
    oauth_nonce=”walat1h”,
    oauth_verifier=”hfdp7dh39dks9884”,
    oauth_signature=”74KNZJeDHnMBp0EMJ9ZHt%2FXKycU%3D”
```

Ο εξυπηρετητής επαληθεύει την αίτηση και απαντά με ένα σύνολο από τεκμηριωμένα διαπιστευτήρια στο σώμα μιας HTTP απάντησης:

```
HTTP/1.1 200 OK
Content-Type: application/x-www-form-urlencoded

oauth_token=nnch734d00s12jdk&oauth_token_secret=pfkkdhi9s13r4s00
```

Έχοντας ο πελάτης λάβει τα διαπιστευτήρια, έχει τη δυνατότητα να ζητήσει την προσωπική φωτογραφία:

```
GET /photos?file=ju.jpg&size=original HTTP/1.1
Host: photos.tmp.gr
Authorization: OAuth realm=”Photos”
   oauth_consumer_key=”dpf43f3p214k3103”,
   oauth_token=”nnch734d00s12jdk”,
   oauth_siganture_method=”HMAC-SHA1”,
   oauth_timestamp=”137131202”,
   oauth_nonce=”chapoH”,
   oauth_signature=”MdpQcU8iPSUjWoN%2FUDMsK2sui9I%3D”
```

Ο εξυπηρετητής στο “photos.tmp.gr” επαληθεύει την αίτηση και απαντά με την φωτογραφία που ζητήθηκε. Ο πελάτης “printer.tmp.gr” έχει τη δυνατότητα να συνεχίσει να έχει πρόσβαση στις προσωπικές φωτογραφίες της Άννας χρησιμοποιώντας το ίδιο σύνολο τεκμηριωμένων διαπιστευτηρίων για όσο διαρκεί η ταυτοποίηση της Άννας, ή μέχρι η Άννα να ανακαλέσει την άδεια πρόσβασης.

2.3. Protocol Buffers

Τα *Protocol Buffers* αποτελούν μια μορφή serialization με μια γλώσσα περιγραφής διεπαφής που αναπτύσσεται από την Google. Η πρωτότυπη υλοποίηση της γλώσσας για τις γλώσσες προγραμματισμού C++, Java και Python είναι διαθέσιμη υπό την άδεια του ελεύθερου λογισμικού και ανοιχτού κώδικα. Πολλές υλοποιήσεις σε διάφορες άλλες γλώσσες προγραμματισμού υπάρχουν διαθέσιμες ή βρίσκονται υπο κατασκευή. Ο σκοπός σχεδίασης της μορφής αυτής ήταν η απλότητα και η αποδοτικότητα. Ειδικότερα σχεδιάστηκε για να είναι μικρότερη αλλά και ταχύτερη από τον προκάτοχο της, την XML.

Τα *Protocol Buffers* χρησιμοποιούνται ευρέως από τη Google για την αποθήκευση και την εναλλαξιμότητα όλων των τύπων πληροφορίας. Αποτελούν τη βάση για ένα custom remote procedure call (RPC) σύστημα το οποίο χρησιμοποιείται σχεδόν σε όλες τις διασυνδέσεις μεταξύ μηχανών στη Google.

Οι δομές δεδομένων που χρησιμοποιούνται στο πρωτόκολλο αυτό ονομάζονται μηνύματα (messages). Έτσι λοιπόν, οι δομές δεδομένων και οι υπηρεσίες που προβλέπονται από το πρωτόκολλο ορίζονται μέσα σε ένα Proto Definion αρχείο, το οποίο έχει την κατάληξη .proto. Το αρχείο αυτό αποτελεί τον πηγαίο κώδικα και το οποίο στη συνέχεια μεταγλωττίζεται με τον μεταγλωττιστή protoc. Η μεταγλώττιση του πηγαίου κώδικα, γεννά ένα νέο πηγαίο κώδικα ο οποίος αντιστοιχεί τους ορισμούς του μηνύματος. Ανάλογα με τα ορίσματα που θα δωθούν στον μεταγλωττιστεί θα παραχθεί ο αντίστοιχος πηγαίος κώδικας για την γλώσσα προγραμματισμού που επιλέχθηκε κατά τη μεταγλώττιση.

Τα *Protocol Buffers* γίνονται serialize σε δυαδική μορφή που αντιστοιχεί σε αυτή του καλωδίου, η οποία παρέχει πλεονεκτήματα όπως ότι είναι συμπαγής, έχει εμπρός και πίσω συμβατότητα, αλλά δεν είναι αυτο-περιγραφόμενη (απαιτείται το αρχείο προδιαγραφών για την ανάκτηση της πληροφορίας). Παρόλο που ο αρχικός σκοπός τους είναι να διευκολύνουν την επικοινωνία μέσω δικτύων, η απλότητα και η ταχύτητα, κάνουν τους *Protocol Buffers* μια εναλλακτική όπου απαιτείται η διαλειτουργικότητα με άλλες γλώσσες προγραμματισμού ή άλλα συστήματα.

2.3.1. Protocol Buffers στο cBox

Το σύστημα που υλοποιήθηκε απαιτεί την συνεργασία μεταξύ διαφορετικών γλωσσών προγραμματισμού όπως είναι η C++ (για την υλοποίηση της βιβλιοθήκης επικοινωνίας) και η JAVA (για την υλοποίηση του επιπέδου εφαρμογών). Έτσι η μορφή των *Protocol Buffers* αποτέλεσε μια ιδανική λύση για την υλοποίηση των μηνυμάτων που αποστέλλονται μεταξύ των κόμβων.

cBoxMessage		
• id:(uint32)	• address:(Addresses)	• data:(Data)

Το κάθε μήνυμα του cBox περιέχει τις τρείς μεταβλητές που αναφέρθηκαν παραπάνω.

- ***id***: Αποτελεί την ταυτότητα του μηνύματος
- ***address***: Περιέχει τις διευθύνσεις που αφορούν το επίπεδο NAL (Network Abstraction Layer).
- ***data***: Περιέχει πληροφορίες και δεδομένα για το επίπεδο εφαρμογής.

Στη συνέχεια θα αναλυθούν οι υπόλοιπες δομές που εμπεριέχονται στο αρχείο ορισμού του μηνύματος τύπου cBoxMessage. Θα πρέπει να αναφερθεί, ότι για μελλοντική επέκταση του μηνύματος έχουν εισαχθεί και δομές οι οποίες δεν χρησιμοποιούνται στην παρούσα έκδοση.

Addresses		
• source_ipv4:(string) • destination_ipv4:(string)	• source_ipv6:(string) • destination_ipv6:(string)	• source_nal:(string) • destination_nal:(string)

Στη δομή Addresses περιέχονται τα εξής στοιχεία:

- ***source_ipv4***: Αποτελεί την IPv4 διεύθυνση του κόμβου που στέλνει το μήνυμα.
- ***source_ipv6***: Αποτελεί την IPv6 διεύθυνση του κόμβου που στέλνει το μήνυμα.
- ***source_nal***: Αποτελεί την NAL διεύθυνση του κόμβου που στέλνει το μήνυμα. Η NAL διεύθυνση δεν έχει υλοποιηθεί, αλλά ορίστηκε για την περίπτωση της δημιουργίας εσωτερικού ανεξάρτητου δικτύου, με ειδικές διευθύνσεις.

- ***destination_ipv4***: Αποτελεί την IPv4 διεύθυνση του κόμβου για τον οποίο προορίζεται το μήνυμα.
- ***destination_ipv6***: Αποτελεί την IPv6 διεύθυνση του κόμβου για τον οποίο προορίζεται το μήνυμα.
- ***destination_nal***: Αποτελεί την NAL διεύθυνση του κόμβου για τον οποίο προορίζεται το μήνυμα. Η NAL διεύθυνση δεν έχει υλοποιηθεί, αλλά ορίστηκε για την περίπτωση της δημιουργίας εσωτερικού ανεξάρτητου δικτύου, με ειδικές διευθύνσεις.

Data			
• type: (Datatype)	• authentication :(string)	• data:(string)	• extended_data :(string)

Στη δομή Data περιέχονται τα εξής στοιχεία:

- ***type***: Αποτελεί τον τύπο των δεδομένων, για την διάκριση μεταξύ των εφαρμογών στο επίπεδο των εφαρμογών.
- ***authentication***: Περιέχει τα διαπιστευτήρια για την επικοινωνία με εφαρμογές που απαιτούν πιστοποίηση.
- ***data***: Περιέχει τα δεδομένα που θέλει να στείλει ο κόμβος.
- ***extended_data***: Περιέχει επιπλέον δεδομένα που υπάρχει περίπτωση να απαιτεί η εφαρμογή.

Datatype		
• TWITTER:(0)	• FLICKR:(1)	• RAW:(2)

Ο τύπος δεδομένων Datatype αποτελεί ένα enumeration με τις εξής οντότητες:

- ***TWITTER:*** Τα δεδομένα αναφέρονται στην εφαρμογή Twitter
- ***FLICKR:*** Τα δεδομένα αναφέρονται στην εφαρμογή Flickr
- ***RAW:*** Τα δεδομένα αναφέρονται σε raw δεδομένα.

SocketType		
• TCP:(0)	• UDP:(1)	• BROADCAST:(2)

Ο τύπος δεδομένων SocketType αποτελεί ένα enumeration με τις εξής οντότητες:

- ***TCP***: Απαιτώντας η αποστολή του μηνύματος να γίνει με τη χρήση TCP Socket.
- ***UDP***: Απαιτώντας η αποστολή του μηνύματος να γίνει με τη χρήση UDP Socket.
- ***BROADCAST***: Απαιτώντας η αποστολή του μηνύματος να γίνει με τη χρήση Broadcast Socket.

2.4. Base64Coder

Η κωδικοποίηση Base64 είναι ένα σύνολο από παρόμοια σχήματα κωδικοποίησης που

αναπαριστούν δυαδικά δεδομένα σε μια μορφή ASCII ακολουθίας αλφαριθμητικών μετατρέποντας το σε μια αναπαράσταση με βάση το 64. Ο όρος Base64 προέρχεται από μια συγκεκριμένη MIME κωδικοποίηση μεταφοράς περιεχομένου. Η κωδικοποίηση Base64 χρησιμοποιείται κυρίως σε περιπτώσεις κωδικοποίησης δυαδικών δεδομένων που πρέπει να αποθηκευθούν και να μεταφερθούν από μέσα που είναι σχεδιασμένα για δεδομένα κειμένου.

Η Base64Coder είναι μια γρήγορη και συμπαγής βιβλιοθήκη η οποία κωδικοποιεί και αποκωδικοποιεί στη μορφή Base64. Δεν υπάρχει κάποια standard βιβλιοθήκη στο JAVA SDK για την κωδικοποίηση Base64. Οι μη τεκμηριωμένες κλάσεις sun.misc.BASE64Encoder και sun.misc.BASE64Decoder δεν θα πρέπει να χρησιμοποιούνται. Για τον λόγο αυτό επιλέχθηκε η ανοιχτού κώδικα υλοποίηση της κωδικοποίησης Base64 στη γλώσσα προγραμματισμού JAVA με το όνομα Base64Coder.Στην υλοποίηση του cBox, η κωδικοποίηση αυτή ήταν απαραίτητη κατά την εισαγωγή δεδομένων στο cBoxMessage.

- Οι Protocol Buffers υποστηρίζουν την μεταφορά μόνο βασικών δομών των γλωσσών προγραμματισμού.

- Κατά τη δημιουργία την εφαρμογής, υπήρξε η ανάγκη μεταφοράς διαφόρων τύπων δεδομένων.

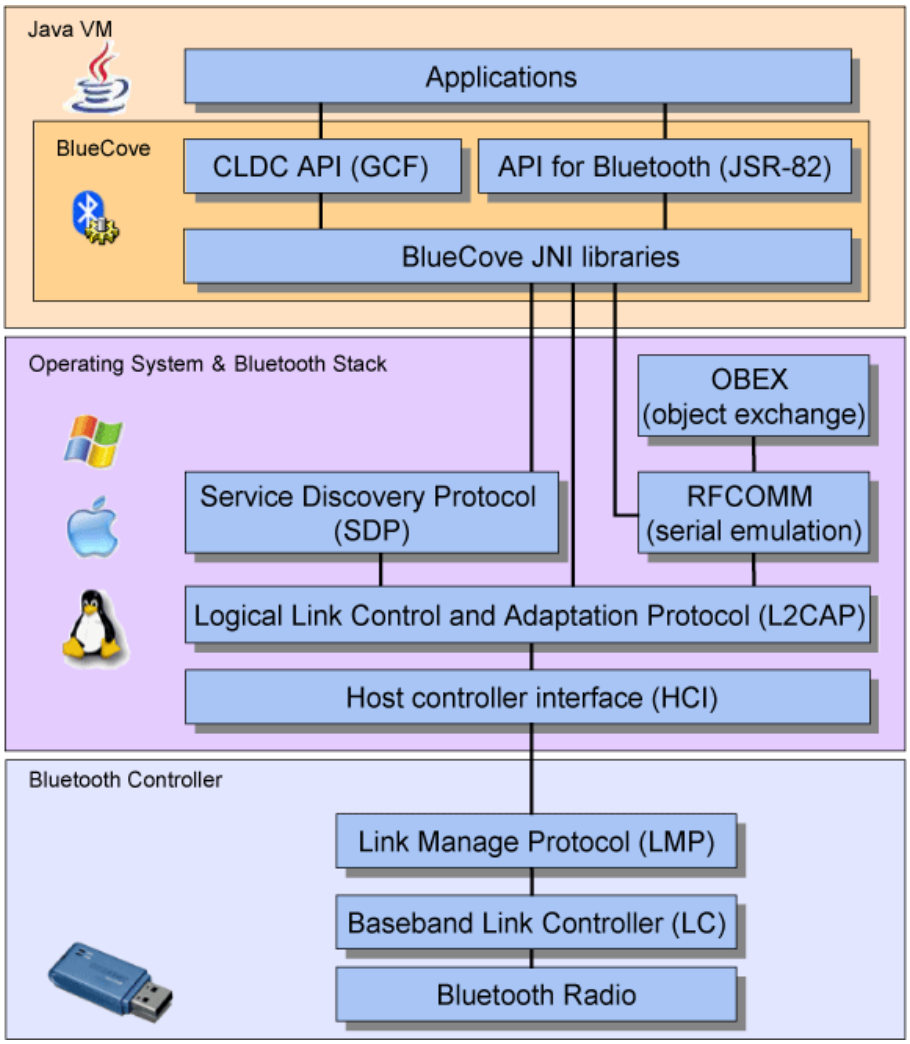
- Για την δυνατότητα μεταφοράς διαφόρων τύπων δεδομένων πραγματοποιείται η ακόλουθη διαδικασία:
  - Μετατροπή του τύπου δεδομένων σε δυαδικό τύπο δεδομένων.
  - Και στη συνέχεια μετατροπή αυτού μέσω του Base64Coder σε ακολουθία αλφαριθμητικών.

- Για την δυνατότητα λήψης διαφόρων τύπων δεδομένων πραγματοποιείται η ακόλουθη διαδικασία:
  - Μετατροπή της ακολουθίας αλφαριθμητικών μέσω του Base64Coder σε δυαδικό τύπο δεδομένων.
  - Και στη συνέχεια μετατροπή του δυαδικού τύπου δεδομένων στον αρχικό τύπο δεδομένων που αποστάλθηκε.

2.5. BlueCove

Η BlueCove είναι μια βιβλιοθήκη ανεπτυγμένη στη γλώσσα προγραμματισμου JAVA για τη διεπαφή του Bluetooth (JSR-82 υλοποίηση) και υποστηρίζει διεπαφές με την πλειονότητα των σύγχρονων λειτουργικών συστημάτων. Δίδεται διαθέσιμο στο κοινό υπό την άδεια χρήσης Apache License Version 2.0 η οποία επιτρέπει την χρήση της βιβλιοθήκης αυτής και σε εμπορικά προϊόντα. Το τμήμα της βιβλιοθήκης που αφορά το λειτουργικό σύστημα Linux και τη χρήση του BlueZ δίδεται υπό την άδεια GNU GPL. Στη συνέχεια αναφέρονται οι διεπαφές που υποστηρίζονται από τη βιβλιοθήκη:

- Mac OS X
- WIDCOMM (Windows Mobile)
- BlueSoleil
- Microsoft Bluetooth (Windows XP, Vista, Mobile)
- BlueZ (Linux)



Η BlueCove παρέχει JSR-82 JAVA διεπαφή για ακόλουθα Bluetooth Profiles:

- ***SDAP*** – Service Discovery Application Profile
- ***RFCOMM*** – Serial Cable Emulation Protocol
- ***L2CAP*** – Logical Link Control and Adaptation Protocol
- ***OBEX*** – Generic Object Exchange Profile (GOEP) πάνω από RFCOMM και TCP

2.6. Wiselib

Η *Wiselib* συνιστά μια βιβλιοθήκη αλγόριθμων για ενσωματωμένες συσκευές που βρίσκονται συνδεδεμένες μέσω ενός δικτύου. Περιέχει διάφορες κλάσεις αλγόριθμων που είναι δυνατό να μεταγλωττιστούν για διαφορετικές πλατφόρμες όπως iSense, Contiki και τον εξομοιωτή δικτύου αισθητήρων Shawn. Είναι εξολοκλήρου γραμμένη στη γλώσσα προγραμματισμού C++ και χρησιμοποιεί templates με τον ίδιο ακριβώς τρόπο όπως η Boost και η CGAL. Με τον τρόπο αυτό δίνει η δυνατότητα να γραφεί γενικός αλλά και ανεξάρτητος από την πλατφόρμα πηγαίος κώδικας που παράλληλα μεταγλωττίζεται αποδοτικά σε πολλές και διαφορετικές πλατφόρμες.

Η *Wiselib* παρέχει εύχρηστες διεπαφές με το λειτουργικό σύστημα, πράγμα το οποίο απλοποιεί τη διαδικασία ανάπτυξης λογισμικού και ελαχιστοποιεί την ανάγκη για χρήση λειτουργιών χαμηλού επιπέδου. Μόλις πραγματοποιηθεί η ανάπτυξη αλγόριθμου, δίνεται η δυνατότητα εκτέλεσης του σε περιβάλλον εξομοίωσης για λόγους αποσφαλμάτωσης. Αφού ο προγραμματιστής βεβαιωθεί για την ορθή εκτέλεση του αλγόριθμου μπορεί να τον μεταγλωττίσει για πραγματικές πλατφόρμες δίχως την ανάγκη αλλαγής του κώδικα που υλοποίησε.

Μετά την ανάπτυξη και αποσφαλμάτωση ενός αλγόριθμου δίνεται η δυνατότητα στον προγραμματιστή να ενσωματώσει τον κώδικα, κατευθείαν στην εφαρμογή στην οποία προορίζεται. Για παράδειγμα, αναπτύσσοντας μια εφαρμογή η οποία προορίζεται για συσκευές iSense η οποία συλλέγει δεδομένα από αισθητήρες, ένας αλγόριθμος δρομολόγησης μπορεί να χρησιμοποιηθεί για την δρομολόγηση των δεδομένων σε ένα κεντρικό σημείο – κόμβο. Ένα άλλο σενάριο αποτελούν οι εφαρμογές που βασίζονται μόνο στη *Wiselib*. Η *Wiselib* παρέχει μια μέθοδο application_main η οποία καλείται από τον κώδικα της *Wiselib,* και στην οποία μπορεί να εισαχθεί ο κώδικας που υλοποίησε ο προγραμματιστής. Ένα μεγάλο πλεονέκτημα αποδίδεται στο γεγονός ότι η εφαρμογή μπορεί να μεταγλωττιστεί για οποιαδήποτε πλατφόρμα υποστηρίζεται από την βιβλιοθήκη, χωρίς την ανάγκη αλλαγής του πηγαίου κώδικα. Το σημείο στο οποίο απαιτείται αλλαγή είναι ο τρόπος με τον οποίο γίνεται η μεταγλώττιση και ειδικότερα τα Makefiles.

Η βιβλιοθήκη *Wiselib* χωρίζεται σε δύο βασικά μέρη:

- ***Εξωτερική Διεπαφή (External Interface)***: ορίζει τη διεπαφή με το Λειτουργικό Σύστημα.

- ***Εσωτερική Διεπαφή (Internal Interface)***: ορίζει τις διαθέσιμες δομές δεδομένων και τους διαθέσιμους αλγόριθμους.

Παρατηρώντας άλλες βιβλιοθήκες που είναι γραμμένες στη γλώσσα προγραμματισμού C++ και χρησιμοποιούν templates (όπως η STL), η *Wiselib* έχει τα ίδια βασικά σχεδιαστικά μέρη:

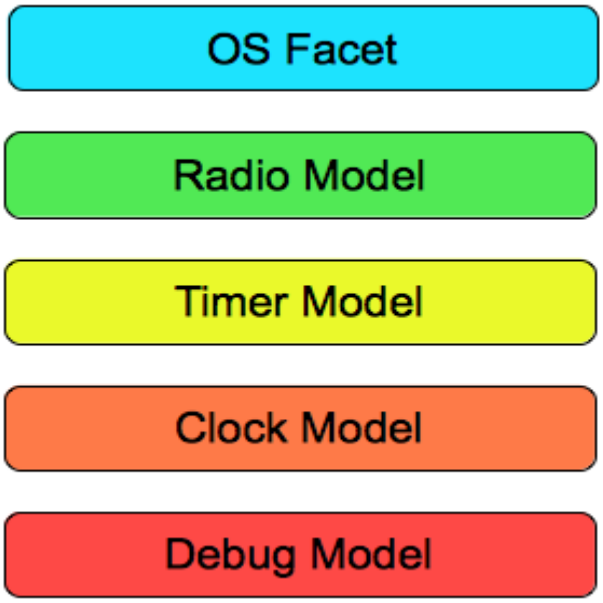
- ***Concept:*** αποτελεί μια περιγραφή διεπαφής και παρουσιάζει τους τύπους και τις μεθόδους που παρέχονται από μία κλάση

- ***Model***: αποτελεί την υλοποίηση ενός ή περισσοτέρων *Concepts*, υλοποιεί κάθε έναν τύπο και κάθε μία μέθοδο που ορίζεται από τη διεπαφή.

2.6.1. Το Android στη Wiselib

Στα External Interfaces της *Wiselib* έχουν υλοποιηθεί πολλά λειτουργικά συστήματα όπως το Contiki ή το iSense OS. Τα τελευταία χρόνια, ένα λειτουργικό σύστημα που βρίσκεται σε ραγδαία ανάπτυξη είναι το Android. Τη στιγμή που συγγράφεται αυτή η εργασία το *Android* βρίσκεται στην πλειονότητα των κινητών τηλεφωνικών συσκευών του κόσμου. Τα επόμενα χρόνια το συγκεκριμένο λειτουργικό σύστημα αναμένεται να λάβει ακόμα μεγαλύτερο μέρος της αγοράς. Το συγκεκριμένο λειτουργικό συστημα υποστηρίζεται από μεγάλες εταιρείες κινητής τηλεφωνίας και παράλληλα είναι ανοιχτού κώδικα λογισμικό το οποίο το καθιστά ακόμα πιο δημοφιλές. Έτσι αποφασίστηκε να εισαχθεί ειδική διεπαφή για το *Android* στη *Wiselib*.

Για την εισαγωγή του *Android* στη *Wiselib* απαιτείται η υλοποίηση ενός *External Interface*. Η διεπαφή αυτή παρέχει πρόσβαση στο λειτουργικό σύστημα της συσκευής. Κάθε λειτουργικό σύστημα στη Wiselib παρέχει ορισμένες λειτουργίες. Η υλοποίηση των λειτουργιών αυτών θα πρέπει να γίνει με όσο τον δυνατόν πιο γενικό τρόπο.



Android OS

- ***OS Facet***: παρέχει πληροφορίες για το λειτουργικό σύστημα. Πιο συγκεκριμένα, ορίζει κάποιες πληροφορίες για τα δεδομένα που μπορεί να επεξεργάζεται και περιέχει επίσης τα Models που έχουν υλοποιηθεί στο λειτουργικό αυτό σύστημα.

- ***Radio Model***: υλοποιεί την δυνατότητα αποστολής και λήψης μηνυμάτων. Στη διεπαφή αυτή, μπορεί να δοθεί ένα μήνυμα, και αυτή στη συνέχεια αναλαμβάνει να

το αποστείλλει στον προορισμό της. Επίσης έχει τη δυνατότητα να λάβει μηνύματα και στη συνέχεια είτε να τα προωθήσει είτε να τα εκτελέσει κάποια λειτουργία βάσει του μηνύματος που έλαβε.

- ***Timer Model***: υλοποιεί την δυνατότητα εισαγωγής χρονιστή. Ο χρονιστής περιέχει μια ουρά από γεγονότα που πρέπει να συμβούν σε ορισμένα χρονικά διαστήματα. Κατά την εκτέλεση μιας εφαρμογής γίνεται εγγραφή κάποιων συμβάντων τα οποία πρέπει να εκτελεστούν σε ορισμένα χρονικά διαστήματα. Μετά το πέρας του χρόνου που έχει οριστεί ένα συμβάν, εκτελείται η μέθοδος που έχει δοθεί στο συγκεκριμένο συμβάν.

- ***Clock Model***: υλοποιεί την δυνατότητα χρήσης εσωτερικού ρολογιού. Παρέχει κάποιες μεθόδους για τη λήψη της τωρινής ώρας, αλλά και κάποιες μεθόδους επερξεργασίας της ώρας, όπως λήψη seconds, milliseconds, microseconds.

- ***Debug Model***: υλοποιεί την δυνατότητα εκτύπωσης μηνυμάτων. Χρησιμοποιείται κυρίως για την εκτύπωση μηνυμάτων κατά την αποσφαλμάτωση μιας εφαρμογής ή ενός αλγόριθμου. Μπορεί να χρησιμοποιηθεί για την ενημέρωση του χρήστη κατά την εκτέλεση μιας εφαρμογής.

2.7. Java Native Interface (JNI)

Το *Java Native Interface (JNI)* αποτελεί ένα προγραμματιστικό framework το οποίο επιτρέπει σε κώδικα της γλώσσας προγραμματισμού JAVA ο οποίος εκτελείται σε ένα Java Virtual Machine (JVM) να καλέσει και να καλεστεί από native εφαρμογές και βιβλιοθήκες που είναι γραμμένες σε γλώσσες προγραμματισμού όπως η C, C++ και ASSEMBLY. Native εφαρμογές καλούνται αυτές οι οποίες έχουν υλοποιηθεί για συγκεκριμένο υλικό και λειτουργικό σύστημα και δεν μπορούν να εκτελεστούν σε διαφορετικές πλατφόρμες. Η εξάρτηση από το υλικό προκαλείται από την επιλογή της γλώσσας προγραμματισμού στην οποία θα υλοποιηθεί η εφαρμογή· η C, η C++ και η ASSEMBLY είναι γλώσσες προγραμματισμού που προκαλούν αυτή την εξάρτηση.

Το *JNI* επιτρέπει στον προγραμματιστή να καλέσει native πηγαίο κώδικα από την γλώσσα προγραμματισμού JAVA. Χρησιμοποιώντας το *JNI* ο προγραμματιστής θυσιάζει κάποια από τα βασικά πλεονεκτήματα που παρέχει η γλώσσα προγραμματισμού JAVA, την δυνατότητα εκτέλεσης σε όλες τις πλατφόρμες που εκτελούν το Java Runtime Environment, για να καλέσει κώδικα γραμμένο σε γλώσσα προγραμματισμού που είναι εξαρτημένος από το υλικό. Μια τέτοια σχεδιαστική επιλογή θα πρέπει να γίνει με πλήρη σύνεση.

Ωστόσο, επιλέγοντας τη χρήση *JNI* παρουσιάζονται δύο σημαντικά *πλεονεκτήματα*:

- Επαναχρησιμοποίηση native πηγαίου κώδικα.

- Αύξηση ταχύτητας εκτέλεσης κρίσιμων, σε θέματα απόδοσης, τμημάτων κώδικα.

Στην περίπτωση του cBox, αποφασίστηκε η εισαγωγή στην εφαρμογή της βιβλιοθήκης Wiselib. Η βιβλιοθήκη Wiselib είναι πλήρως γραμμένη στη γλώσσα προγραμματισμού C++. Εφόσον το επίπεδο εφαρμογών θα υλοποιούταν στη γλώσσα προγραμματισμού JAVA, για να είναι δυνατή η συμβατότητα με τη βιβλιοθήκη Wiselib, επιλέχθηκε η εισαγωγή JNI στην εφαρμογή. Η συμβατότητα με μια τέτοια βιβλιοθήκη παρέχει στους χρήστες του cBox πληθώρα πλεονεκτημάτων. Ένα από τα σημαντικότερα πλεονεκτήματα είναι το πλήθος των αλγόριθμων που έχουν υλοποιηθεί και ελεγχθεί για τη βιβλιοθήκη αυτή, και τους οποίους οποιοσδήποτε χρήστης μπορεί να εισάγει κατευθείαν στο cBox, χωρίς την παραμικρή μετατροπή του προϋπάρχοντα κώδικα. Ακόμη, εάν κάποιος χρήστης επιθυμεί να υλοποιήσει έναν αλγόριθμο δρομολόγησης πακέτων για το cBox, μπορεί αρχικά να τον υλοποιήσει για την βιβλιοθήκη Wiselib, να τον ελέγξει για την ορθή εκτέλεση του στο εξομοιωτή Shawn και στη συνέχεια αφού βεβαιωθεί για την ορθή του λειτουργία, μπορεί χωρίς καμία απολύτως μετατροπή να τον εισάγει στο cBox.

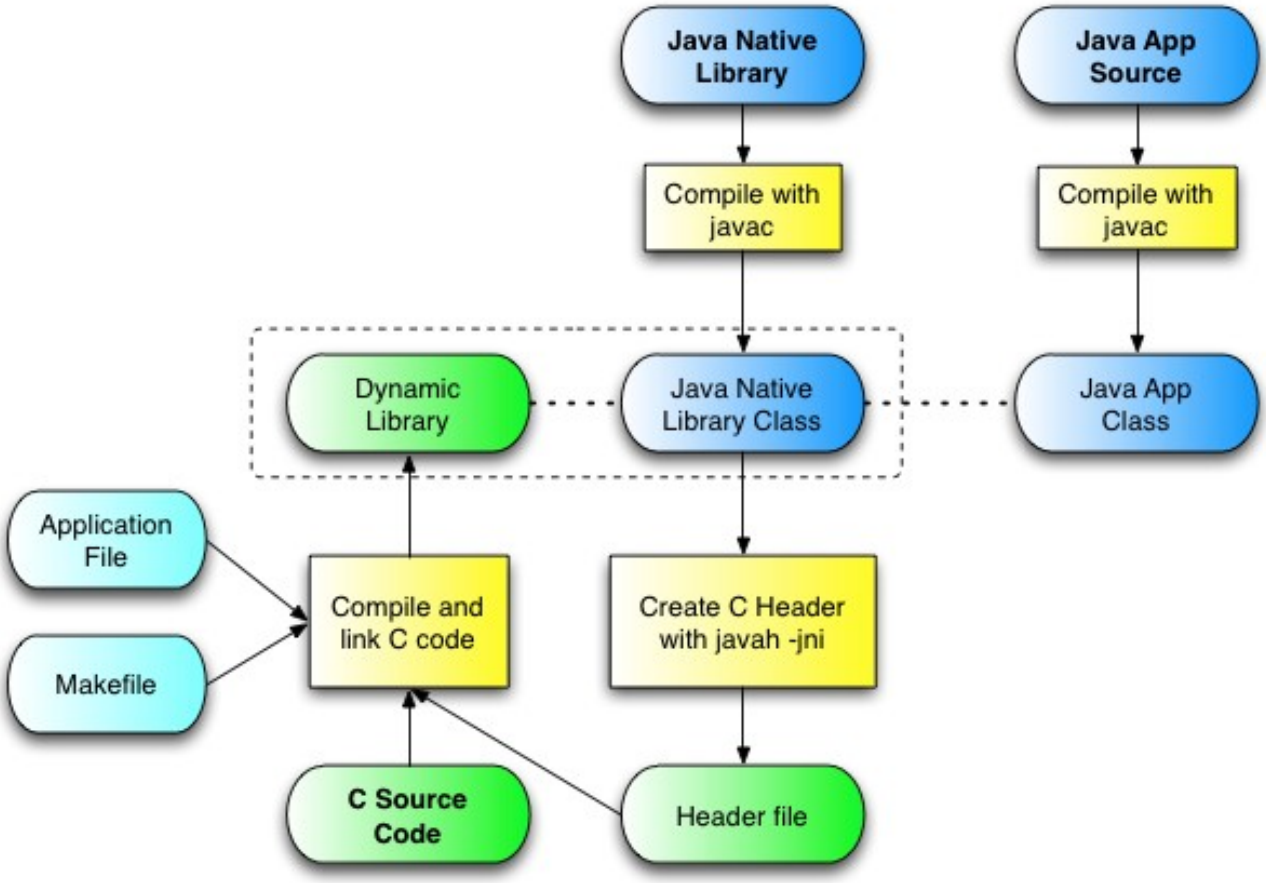
Ωστόσο το JNI παρέχει και την δυνατότητα εκτέλεσης πηγαίου κώδικα της γλώσσας προγραμματισμού JAVA από native πηγαίο κώδικα. Η επιλογή αυτή γίνεται κυρίως σε

περιπτώσεις όπου υπάρχει κάποια έτοιμη βιβλιοθήκη η οποία έχει υλοποιηθεί σε γλώσσα προγραμματισμού JAVA και αποτελεί άμεση ανάγκη η χρησιμοποίηση της από τον native κώδικα. Στην περίπτωση του cBox, η επιλογή αυτή χρησιμοποιήθηκε στο επίπεδο Network Abstraction Layer. Η επικοινωνία των cBox μεταξύ τους γίνεται με τη χρήση native κώδικα και ιδιαίτερα με Sockets. Επειδή όμως στο Android, όπου μέσω του Native Development Kit (NDK) γίνεται χρήση του *JNI* και η βιβλιοθήκη Bluez δεν υποστηρίζεται στο NDK, αναγκαστικά επιλέχθηκε η λύση της κλήσης βιβλιοθήκης σε γλώσσα προγραμματισμού JAVA η οποία υλοποιεί τη διεπαφή με το Bluetooth.

2.7.1. Android και JNI – Native Development Kit

Το λειτουργικό σύστημα *Android* είναι βασισμένο στο πολύ γνωστό και ευρέως χρησιμοποιούμενο Linux. Έτσι παρέχει σχεδόν πλήρη συμβατότητα με τις διάφορες συσκευές που βρίσκονται αυτή τη στιγμή στην αγορά. Ακόμη, η γλώσσα που χρησιμοποιείται ίσως περισσότερο σήμερα, είναι η γλώσσα προγραμματισμού JAVA λόγω της δυνατότητας εκτέλεσης σε διάφορες πλατφόρμες αλλά και της απλότητας κατά την περίοδο ανάπτυξης λογισμικού. Ωστόσο έχει ένα μεγάλο μειονέκτημα, την ταχύτητα εκτέλεσης. Έτσι δημιουργήθηκε το *JNI* το οποίο επιτρέπει στους προγραμματιστές να εκτελούν native κώδικα μέσω εφαρμογών JAVA.

Για να επεκταθεί η δυνατότητα αυτή, της εκτέλεσης native κώδικα μέσω JAVA εφαρμογών και στο Android, υλοποιήθηκε από την Google, το *Native Development Kit (NDK)*. Το *Android NDK* είναι ένα εργαλείο το οποίο εξαρτάται από το *Android SDK* και επιτρέπει στον προγραμματιστή να αναπτύξει τμήματα κώδικα που είναι κρίσιμα από θέμα απόδοσης σε μια εφαρμογή με την χρήση native κώδικα.



Παρέχει headers και βιβλιοθήκες που δίνουν πολλές δυνατότητες στον προγραμματιστή. Μερικές από αυτές είναι:

- Κατασκευή Activities.
- Χειρισμό της εισόδου από τον χρήστη.
- Χρησιμοποίηση των αισθητήρων που βρίσκονται στο υλικό της συσκευής.
- Πρόσβαση στους πόρους της εφαρμογής.

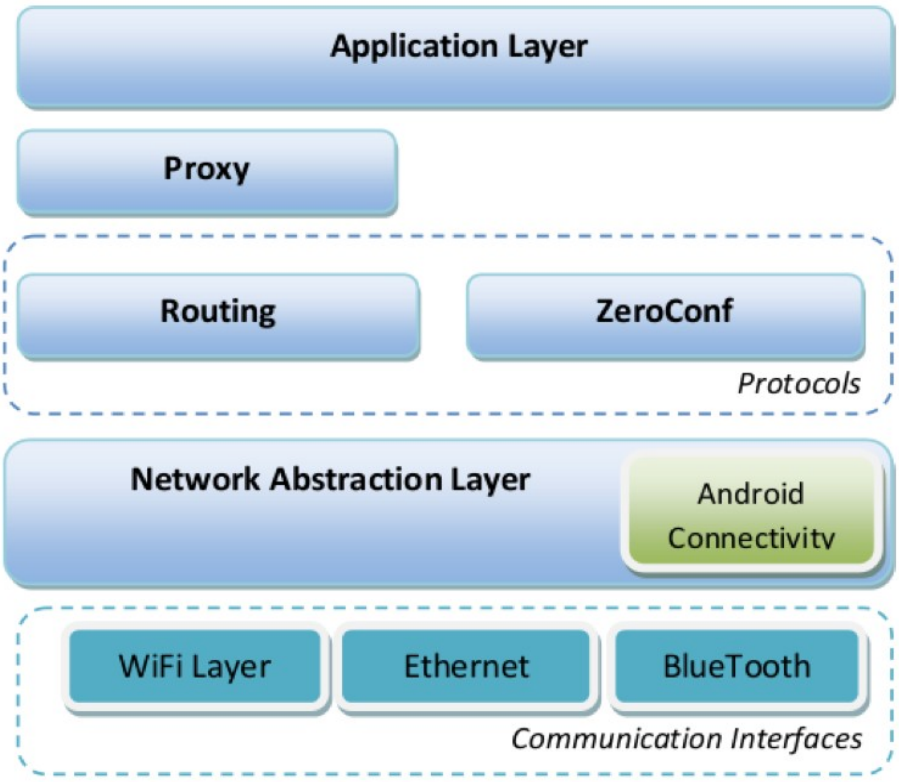
Κατά την ανάπτυξη εφαρμογών που χρησιμοποιούν το NDK, το εκτελέσιμο της εφαρμογής εισάγεται και πάλι σε αρχείο .apk και εκτελείται μέσω VM στην συσκευή. Αυτό συμβαίνει διότι, στην ουσία δημιουργείται μια JAVA εφαρμογή η οποία καλεί native κώδικα, οποίος πραγματοποιεί τις λειτουργίες που επιθυμεί ο προγραμματιστής.

Το NDK παρέχεται και για τα τρία μεγάλα και ευρέως χρησιμοποιούμενα λειτουργικά συστήματα: Windows, Mac OS X, Linux.

3. Αρχιτεκτονική του Συστήματος cBox

Για την δημιουργία ενός συστήματος όπως το cBox, το οποίο υλοποιεί πολλές και διαφορετικές λειτουργίες, απαιτείται ο διαχωρισμός του συστήματος σε διάφορα επίπεδα. Ο διαχωρισμός αυτός εξυπηρετεί την ευκολότερη ανάπτυξη του συστήματος, αλλά και συμβατότητα με νέα επιπρόσθετα στοιχεία. Η συμβατότητα αποτελεί ένα από τα βασικά πλεονεκτήματα του συστήματος cBox, το οποίο δίνει την δυνατότητα να εισαχθούν νέες λειτουργίες σε αυτό, χωρίς την παραμικρή μετατροπή στην υπάρχουσα εφαρμογή. Το σύστημα λογισμικού έχει διαχωριστεί σε διάφορα επίπεδα. Το κάθε επίπεδο παρέχει μια διεπαφή που είναι συμβατή και με τα επόμενα επίπεδα. Με τον τρόπο αυτό μπορεί εύκολα και χωρίς κόπο να πραγματοποιηθεί προσθαφαίρεση επιπέδων και στοιχείων της αρχιτεκτονικής, χωρίς να επηρεαστεί η λειτουργία του συστήματος. Το κάθε επίπεδο είναι ανεξάρτητο από τα υπόλοιπα, δηλαδή λειτουργούν αυτόνομα. Η διεπαφή που παρέχει το κάθε επίπεδο ωστόσο, είναι συμβατή με τα γειτονικά επίπεδα. Η αφαίρεση επιπέδων είναι δυνατή, με ανάλογες επιπτώσεις στη λειτουργικότητα του συστήματος.

Έτσι το σύστημα cBox διαχωρίστηκε σε διάφορα επίπεδα, τα οποία εξυπηρετούν εξειδικευμένες λειτουργίες.



- ***Communication Interfaces*** : Αποτελείται από τις διεπαφές που επιτρέπουν στις συσκευές να επικοινωνούν μεταξύ τους. Είναι το κατώτερο επίπεδο της εφαρμογής. Καμία από τις διεπαφές δεν είναι υποχρεωτική. Ωστόσο αν δεν υπάρχει καμία από τις τρεις διεπαφές, δεν είναι δυνατή η επικοινωνία της συσκευής με άλλα μέλη του cBox συστήματος.

- ***Network Abstraction Layer*** : Αποτελεί το επίπεδο αφαιρετικότητας για την επικοινωνία μεταξή των συσκευών. Λαμβάνει ένα μήνυμα από κάποιο από τα ανώτερα επίπεδα και αναλαμβάνει με πλήρη διαφάνεια να αποστείλλει το μήνυμα χρησιμοποιώντας όλες τις διαθέσιμες διεπαφές επικοινωνίας. Συνιστά αναπόσπαστο τμήμα του cBox, διότι διαφορετικά δεν είναι δυνατή η εκμεταλλευση των διεπαφών επικοινωνίας με άμεση συνέπεια την αδυναμία επικοινωνίας με άλλες συσκευές. Το επίπεδο είναι *υποχρεωτικό*.

- ***Protocols*** : Αποτελεί το επίπεδο όπου εισάγονται τα διάφορα πρωτόκολλα δρομολόγησης πακέτων αλλά και διαχείρισης ονοματολογίας των συσκευών στο εσωτερικό δίκτυο που δημιουργείται. Το επίπεδο αυτό είναι απαραίτητο για την χρήση διεπαφών WiFi και Ethernet, όπου η επικοινωνία γίνεται με την χρήση IP διευθύνσεων. Σε περίπτωση μη υλοποίησης του επιπέδου αυτού, η επικοινωνία είναι δυνατή μόνο με τη χρήση Bluetooth, Broadcast και καταχώρηση στατικών διευθύνσεων για τις άλλες δυο διεπαφές. Το επίπεδο είναι *προαιρετικό*.

- ***Proxy*** : Αποτελεί το επίπεδο στο οποίο εισάγεται ένας proxy server. Χρησιμοποιείται για την προσωρινή αποθήκευση αιτήσεων, μηνυμάτων και web πληροφοριών (όπως ιστοσελίδες) για την αύξηση της ταχύτητας και της αξιοπιστίας του δικτύου. Ακόμη δίνεται η δυνατότητα της ανωνυμίας του χρήστη που αποστέλλει πληροφορίες στο διαδίκτυο και γενικότερα στο δίκτυο. Το επίπεδο είναι *προαιρετικό*.

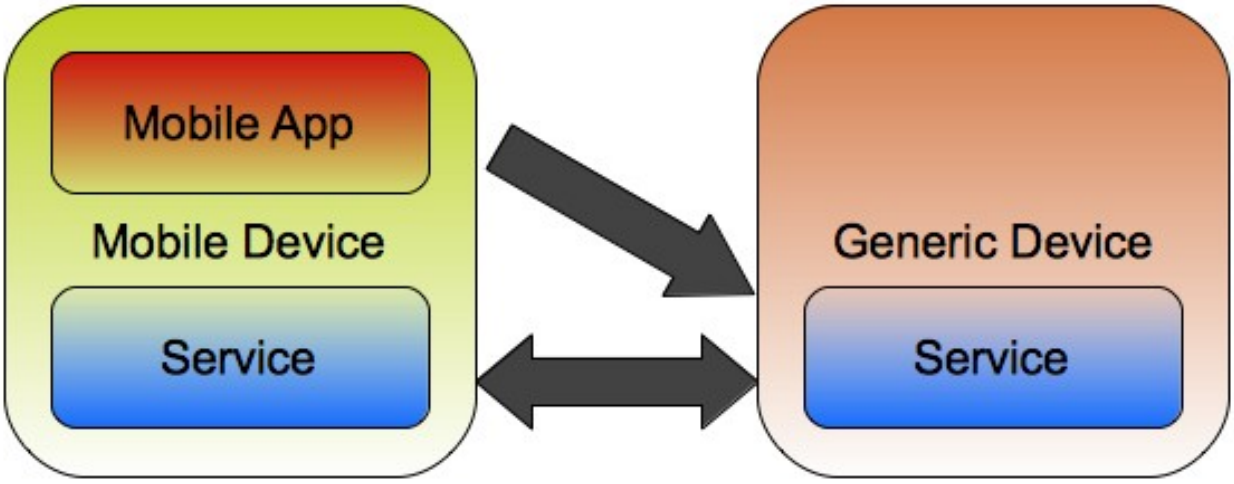
- ***Application Layer*** : Αποτελεί το επίπεδο εφαρμογών. Το επίπεδο αυτό εκμεταλλεύεται τις υπηρεσίες που του παρέχονται από τα χαμηλότερα επίπεδα και υλοποιεί τη διεπαφή με τον χρήστη και τις λειτουργίες της εφαρμογής. Το επίπεδο είναι *υποχρεωτικό*.

3.1. Application Layer

Το *Application Layer* είναι το επίπεδο εφαρμογών, το οποίο υλοποιεί την ίδια την εφαμοργή και αλληλεπιδρά με τον χρήστη. Στο επίπεδο αυτό είναι δυνατή η υλοποίηση οποιασδήποτε εφαρμογής, ωστόσο για την επικοινωνία με τις υπόλοιπες συσκευές που βρίσκονται στο δίκτυο γίνεται χρήση τον χαμηλότερων επιπέδων του συστήματος. Όπως είναι γνωστό, είναι δυνατή η εκτέλεση του cBox σε διαφορετικές πλατφόρμες. Για να επιτεχθεί η ιδιότητα αυτή το επίπεδο εφαρμογών στο οποίο στηρίζεται ολόκληρο το σύστημα, το επίπεδο αυτό υλοποιήθηκε με μια γλώσσα προγραμματισμού που είναι ανεξάρτητη της πλατφόρμας στην οποία εκτελείται. Η γλώσσα αυτή προγραμματισμού είναι η JAVA. Η γλώσσα αυτή δίνει τη δυνατότητα στην εφαρμογή να εκτελεστεί σε όλες τις πλατφόρμες που έχουν αναφερθεί. Ωστόσο για την προσθήκη λειτουργικότητας σε κάποιες πλατφόρμες πραγματοποιήθηκε μια διάκριση στο επίπεδο των εφαρμογών:

- Εφαρμογή για ***κινητές συσκευές*** (Android): Συνιστά το τμήμα του επιπέδου εφαρμογών το οποίο υλοποιεί μια εφαρμογή για κινητή συσκευή. Το συστατικό αυτό περιέχει την διεπαφή με τον χρήστη και διάφορες λειτουργίες που παρέχει η εφαρμογή.

- Εφαρμογή που εκτελείται ως ***υπηρεσία***. Συνιστά το τμήμα του επιπέδου εφαρμογών το οποίο εκτελείται στο παρασκήνιο. Ο χρήστης δεν αντιλαμβάνεται ότι υπάρχει το συστατικό αυτό. Είναι υπεύθυνο για την προώθηση μηνυμάτων που λαμβάνει από γειτονικές συσκευές στον τελικό προορισμό τους. Θα πρέπει να τονιστεί ότι και η εφαρμογή για κινητές συσκευές εκτελεί παράλληλα την υπηρεσία, για να μπορούν και οι κινητές συσκευές να προωθούν τυχόν μηνύματα που λαμβάνουν.



Η διάκριση που πραγματοποιήθηκε, έγινε με σκοπό την ευκολότερη κατανόηση της αρχιτεκτονικής του συγκεκριμένου επιπέδου. Το βασικότερο τμήμα του επιπέδου εφαρμογών είναι το τμήμα του εκτελείται ως υπηρεσία, αφού χρησιμοποιείται και στις κινητές συσκευές. Προς το παρόν, οι εφαρμογές που προορίζονται για οποιεσδήποτε συσκευές ***εκτός*** κινητών τηλεφώνων, εκτελούν ***μόνο*** την υπηρεσία. Θα πρέπει να αναφερθεί ωστόσο πως είναι δυνατή η παράλειψη της εφαρμογής για κινητή συσκευή με αποτέλεσμα την εγκατάσταση μόνο της υπηρεσίας στην κινητή συσκευή. Το γεγονός αυτο προϋποθέτει βέβαια τη μεταγλώττιση της εφαρμογής για την πλατφόρμα της κινητής συσκευής, ωστόσο έχει τη δυνατότητα της ανεξάρτητης εκτέλεσης σαν να ήταν οποιαδήποτε συσκευή. Η εφαρμογή κινητής συσκευής απλά προσθέτει επιπλέον λειτουργικότητα στον χρήστη.

3.1.1. Η Twitter εφαρμογή

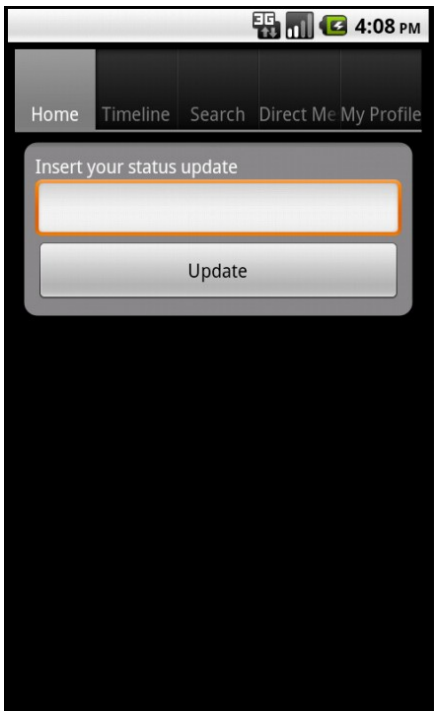
Έχοντας υπόψιν τους βασικούς σκοπούς δημιουργίας και ανάπτυξης του project FreedomBox, η χρήση των κοινωνικών δικτύων αποτελεί ένα σημαντικό γεγονός. Οι στόχοι αυτού του project αναφέρονταν σε κοινωνικά δίκτυα που σέβονται την ιδιωτικότητα του χρήστη. Λόγω του γεγονότος ότι μέχρι σήμερα δεν υπάρχει κάποιο αντίστοιχα ευρέως διαδεδομένο κοινωνικό δίκτυο, επιλέχθηκε ένα από τα δημοφιλέστερα στον κόσμο, το Twitter. Το κοινωνικό αυτό δίκτυο έχει χρησιμοποιηθεί σε περιόδους φυσικής καταστροφής είτε σε περιοχές όπου δεν υφίσταται η ελευθεροτυπία και η ελευθερία της έκφρασης για την μαζική οργάνωση λαών.

3.1.2. Λειτουργίες της Twitter εφαρμογής

Λαμβάνοντας έτσι υπόψιν τον βασικό σκοπό δημιουργίας ενός τέτοιου συστήματος όπως το cBox, μια εφαρμογή για το κοινωνικό δίκτυο Twitter αποτελεί ιδανική εκκινητήρια εφαρμογή για το σύστημα αυτό. Η εφαρμογή που δημιουργήθηκε για το cBox, παρέχει όλες τις βασικές λειτουργίες που θα πρέπει να είναι διαθέσιμες από μια Twitter εφαρμογή. Οι λειτουργίες αυτές είναι:

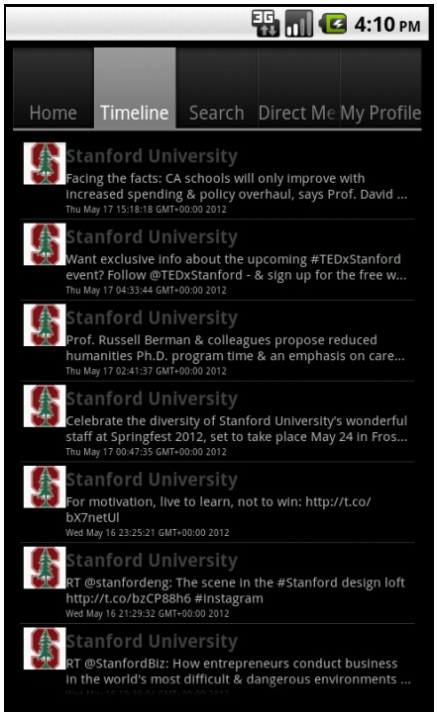
- ***Σύνδεση σε λογαριασμό*** με την χρήση του πρωτοκόλλου OAuth.

- Αποστολή ***Status Update***.



- Με την χρήση αυτής της επιλογής ο χρήστης ο οποίος έχει συνδεθεί μέσω της εφαρμογής έχει τη δυνατότητα να αποστείλλει ένα Status Update – Tweet στον λογαριασμό του στο Twitter.
- Η αποστολή ενός tweet γίνεται μέσω της εισαγωγής του κειμένου που ο χρήστης επιθυμεί και στη συνέχεια για να πραγματοποιηθεί η αποστολή πρέπει ο χρήστης να πατήσει το κουμπί Update.
- Κατά την εισαγωγή κειμένου, η περιοχή στην οποία εγγράφεται το κείμενο προσαρμόζεται ανάλογα με το μέγεθος του κειμένου.
- Αποτελεί την ***εκκινητήρια οθόνη*** της εφαρμογής.

• Λήψη ***Timeline*** και εμφάνιση του σε λίστα.

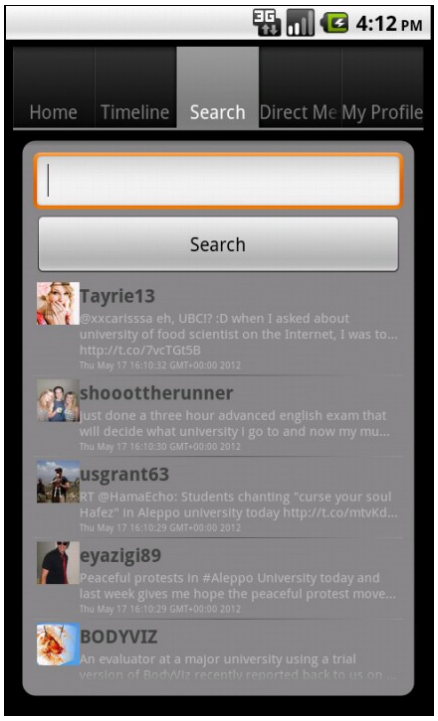


- Με την επιλογή της καρτέλας Timeline εμφανίζεται στην οθόνη το Timeline του συγκεκριμένου λογαριασμού, δηλαδή τα τελευταία tweets των χρηστών που ακολουθεί ο συγκεκριμένος λογαριασμός.
- Επειδή η εφαρμογή είναι πραγματικού χρόνου, κάθε φορά που επιλέγεται η συγκεκριμένη καρτέλα, λαμβάνονται από το Twitter, τα νεώτερα tweets.
- Κατά την λήψη των νεώτερων tweets των χρηστών που ακολουθεί ο συγκεκριμένος λογαριασμός, εμφανίζεται ένα παράθυρο και ενημερώνει τον χρήστη πως θα πρέπει να περιμένει για να ληφθούν τα tweets από το Twitter. Όσο το παράθυρο αναμονής βρίσκεται στο προσκήνιο, δεν είναι δυνατή η

πραγματοποίηση κάποιας άλλης λειτουργίας.

- Μόλις ληφθούν τα tweets εμφανίζονται σε μια λίστα κύλισης.
- Το κάθε στοιχείο της λίστας κύλισης, αποτελείται από:
  - Τη ***φωτογραφία*** του χρήστη που πραγματοποίησε το tweet.
  - Το ***όνομα χρήστη*** που πραγματοποίησε το tweet.
  - Το ***tweet*** που απέστειλε ο συγκεκριμένος χρήστης. Ο αριθμός των χαρακτήρων του tweet είναι προκαθορισμένος. Αν ξεπεραστεί τότε κατά την παρουσίαση του tweet εμφανίζονται τρείς τελείες, ενημερώνοντας τον χρήστη πως το tweet είναι μεγαλύτερου μήκους από αυτό που μπορεί να εμφανιστεί.
  - Την ***ημερομηνία*** και ***ώρα***, όπου στάλθηκε το συγκεκριμένο tweet στη μορφή: “Thu May 17 15:18:18 GMT+00:00 2012”

- ***Αναζήτηση*** στα δεδομένα του Twitter.



- Με την επιλογή της συγκεκριμένης καρτέλας εμφανίζεται μια μπάρα εισαγωγής δεδομένων και ένα κουμπί με τον τίτλο “Search”.
- Με την εισαγωγή ενός κριτηρίου, και την επιλογή του κουμπιού “Search” πραγματοποιείται αναζήτηση στα δεδομένα του Twitter, χρησιμοποιώντας το κριτήριο που δόθηκε από τον χρήστη.
- Επειδή η αναζήτηση πραγματοποιείται στα δεδομένα του Twitter απαιτείται

κάποιος χρόνος για την αποστολή του κριτηρίου αναζήτησης, την πραγματοποίηση της αναζήτησης στο Twitter και την λήψη του αποτελέσματος της αναζήτησης.

- Κατά την αναμονή για το αποτέλεσμα της αναζήτησης εμφανίζεται παράθυρο το οποίο ενημερώνει τον χρήστη πως θα πρέπει να περιμένει εως ότου λάβει τα αποτελέσματα της αναζήτησης.
- Τα αποτελέσματα της αναζήτησης είναι μια λίστα από tweets τα οποία εμφανίζονται στην οθόνη κάτω από το κουμπί “Search” σε μια λίστα κύλισης.
- Το κάθε στοιχείο της λίστας κύλισης αποτελείται από:
  - Τη ***φωτογραφία*** του χρήστη που πραγματοποίησε το tweet.
  - Το ***όνομα χρήστη*** που πραγματοποίησε το tweet.
  - Το ***tweet*** που απέστειλε ο συγκεκριμένος χρήστης. Ο αριθμός των χαρακτήρων του tweet είναι προκαθορισμένος. Αν ξεπεραστεί τότε κατά την παρουσίαση του tweet εμφανίζονται τρείς τελείες, ενημερώνοντας τον χρήστη πως το tweet είναι μεγαλύτερου μήκους από αυτό που μπορεί να εμφανιστεί.
  - Την ***ημερομηνία*** και ***ώρα***, όπου στάλθηκε το συγκεκριμένο tweet στη μορφή: “Thu May 17 15:18:18 GMT+00:00 2012”

- Αποστολή και Λήψη ***απευθείας μηνυμάτων***.
  - Με την επιλογή της συγκεκριμένης καρτέλας εμφανίζεται ένα παράθυρο αναμονής κατά το οποίο ενημερώνει τον χρήστη πως πρέπει να αναμείνει, εως ότου λάβει τα απευθείας μηνύματα που του έχουν αποσταλεί.
  - Με την επιλογή της συγκεκριμένης καρτέλας εμφανίζονται δύο πεδία εισαγωγής δεδομένων:
    - ***Όνομα χρήστη*** στον οποίο επιθυμεί ο λογαριασμός να αποστείλει το μήνυμα του.
    - Το ***μήνυμα*** το οποίο επιθυμεί ο λογαριασμός στον χρήστη.
  - Με την επιλογή της συγκεκριμένης καρτέλας εμφανίζονται προηγούμενα και νέα απευθείας μηνύματα τα οποία έχει λάβει ο λογαριασμός από άλλους χρήστες.
  - Τα απευθείας μηνύματα αφού ληφθούν από το Twitter, εμφανίζονται στην οθόνη μέσω μιας ***λίστας κύλισης***.
  - Το κάθε στοιχείο της λίστας κύλισης αποτελείται από:
    - Το ***όνομα χρήστη*** που απέστειλε το συγκεκριμένο απευθείας μήνυμα.
    - Το ***μήνυμα*** που απέστειλε.
    - Την ***ημερομηνία*** και ***ώρα***, όπου στάλθηκε το συγκεκριμένο tweet στη μορφή: “Thu May 17 15:18:18 GMT+00:00 2012”

- Για την αποστολή ενός απευθείας μηνύματος, αφού εισαχθούν τα πεδία ονόματος χρήστη και μηνύματος επιλέγεται το κουμπί “Send direct message”, το οποίο πραγματοποιεί την αποστολή του απευθείας μηνύματος.
- Μετά την λήψη ενός απευθείας μηνύματος ο λογαριασμός ενημερώνεται για το απευθείας μήνυμα μέσω ηλεκτρονικού ταχυδρομείου.

- Λήψη ***στατιστικών*** και ***πληροφορίες*** που αφορούν τον λογαριασμό του χρήστη που έχει συνδεθεί.

- Με την επιλογή της συγκεκριμένης καρτέλας εμφανίζεται ένα παράθυρο αναμονής κατά το οποίο ενημερώνει τον χρήστη πως πρέπει να αναμείνει έως ότου λάβει τις απαιτούμενες πληροφορίες από το Twitter για τον λογαριασμό αυτόν.
- Μόλις ληφθούν οι πληροφορίες για τον συνδεδεμένο λογαριασμό εμφανίζονται:
  - Η ***φωτογραφία*** που έχει εισάγει ο χρήστης στον λογαριασμό του.
  - Το ***όνομα χρήστη*** – ψευδώνυμο που χρησιμοποιεί ο χρήστης.
  - Ο ***αριθμός*** των ***χρηστών*** που ***ακολουθούν*** το συγκεκριμένο λογαριασμό.

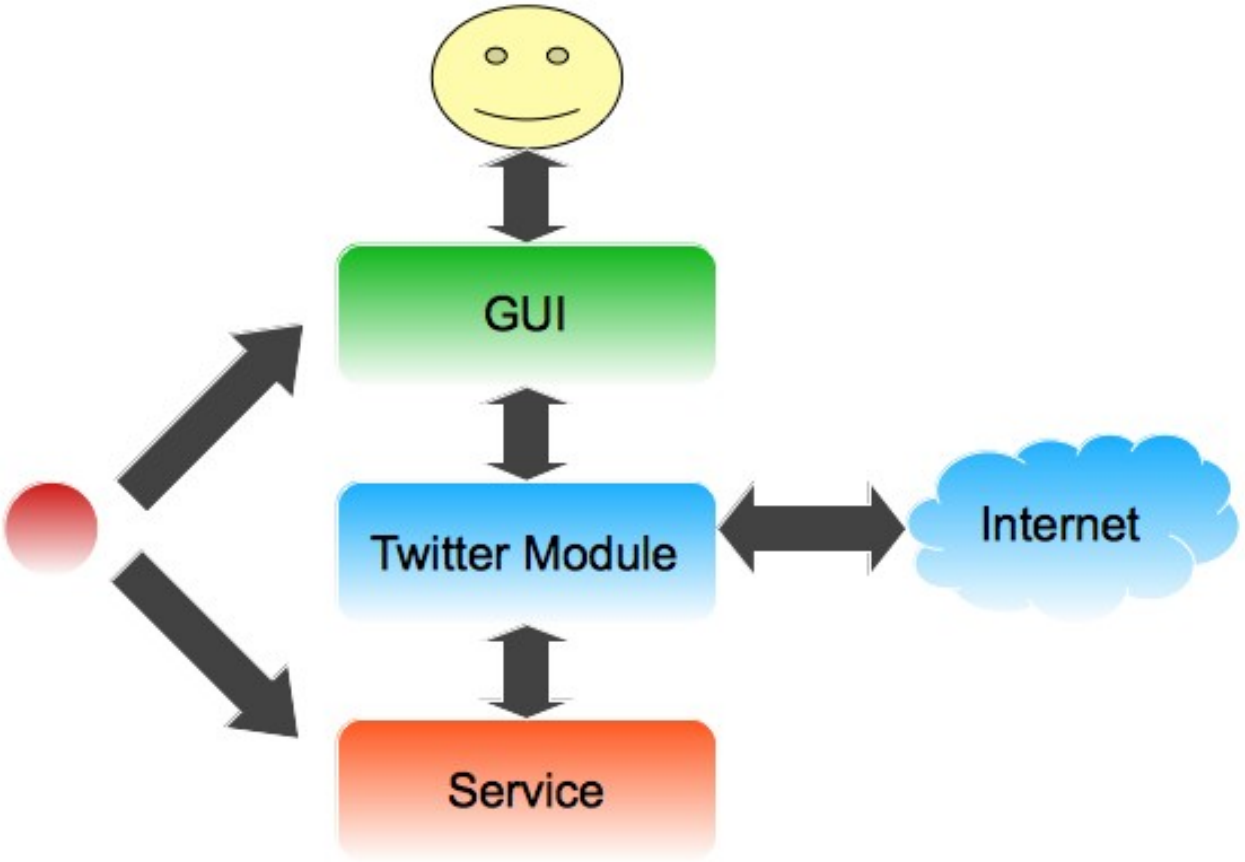
- Ο ***αριθμός*** των ***φίλων*** του συγκεκριμένου λογαριασμού.
- Ο ***αριθμός*** των ***αγαπημένων*** του συγκεκριμένου λογαριασμού.
- Ο ***αριθμός*** των ***tweets*** που έχει πραγματοποιήσει ο συγκεκριμένος λογαριασμός.

- Πλοήγηση στο μενού με την χρήση ***tabs***.

3.1.3. Σχεδιασμός της Twitter εφαρμογής

Η εφαρμογή για το Twitter υλοποιεί όλες τις βασικές λειτουργίες που απαιτούνται από έναν κοινό χρήστη του κοινωνικού δικτύου Twitter. Είναι μία εφαρμογή η οποία υλοποιεί το Graphic User Interface (GUI) δηλαδή την διεπαφή με τον χρήστη, την ορθή εκτέλεση των λειτουργιών που παρατέθηκαν, την επικοινία με το Twitter και την διεπαφή με το συστατικό Service για την επικοινωνία με τις υπόλοιπες συσκευές του δικτύου.

Για τον ευκολότερο αλλά και σαφέστερο τρόπο υλοποίησης της εφαρμογής χρησιμοποιήθηκε η μέθοδος ανάπτυξης λογισμικού Model View Controller (MVC), η οποία διαχωρίζει το γραφικό περιβάλλον (GUI) από την λειτουργικότητα της εφαρμογής αλλά και τα δεδομένα της.



Κατά την έναρξη της εκτέλεσης της εφαρμογής εκτελούνται παράλληλα:

- Η εφαρμογή του Twitter.

- Λαμβάνει τα αιτήματα και τις παραμέτρους αυτών από τον χρήστη μέσω του GUI.
- Χρησιμοποιώντας το Twitter Module εκτελεί τα αιτήματα που έλαβε από τον χρήστη και τα αποστέλλει στο Twitter.com.
- Λαμβάνει το αποτέλεσμα του αιτήματος του χρήστη από το Twitter.
- Εμφανίζει στον χρήστη τα αποτελέσματα της αίτησης του μέσω του GUI.
- Αν πραγματοποιήθηκε η αίτηση, λαμβάνει ένα μήνυμα στο ηλεκτρονικό του ταχυδρομείο.

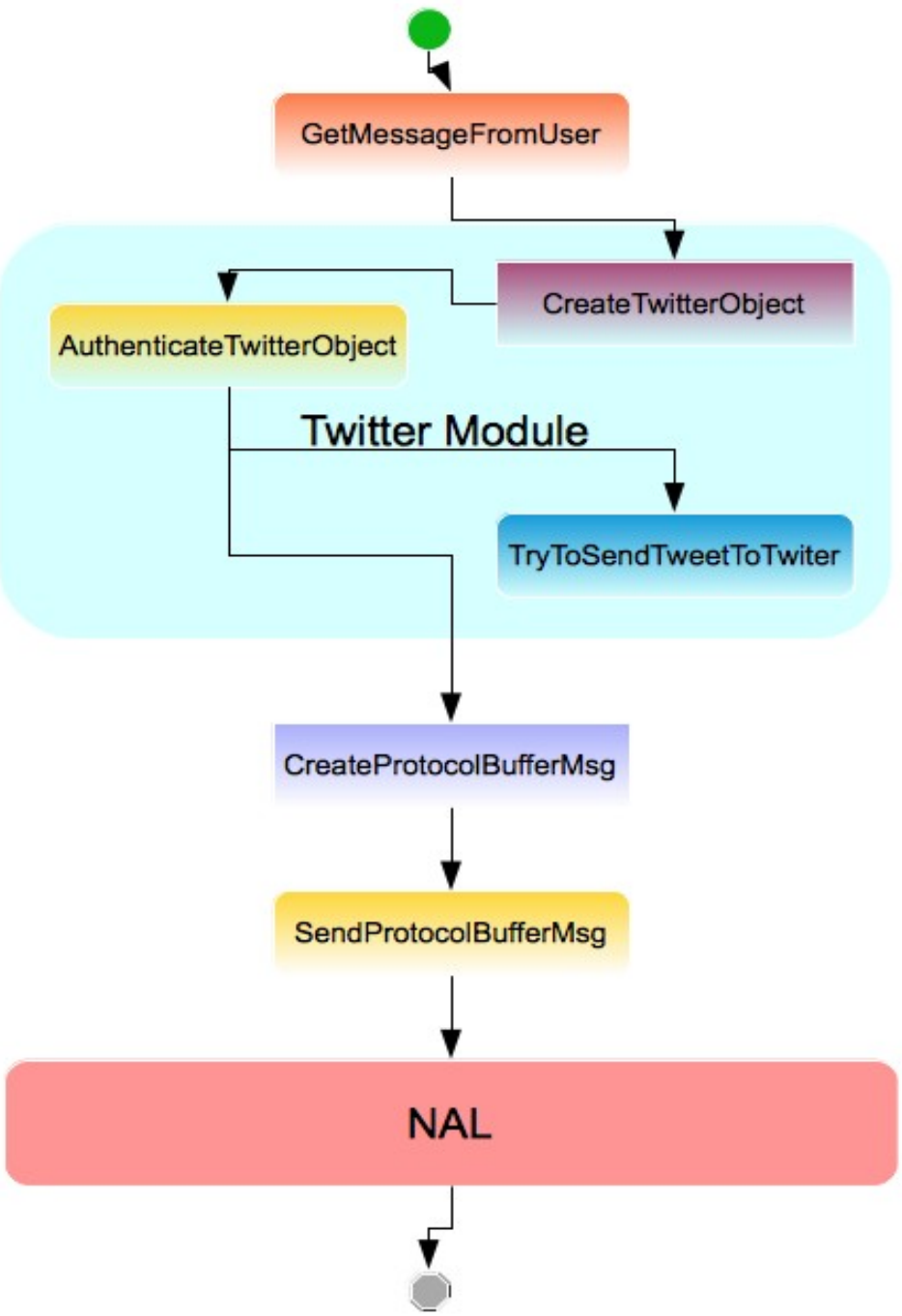
• Η υπηρεσία για την επικοινωνία των κόμβων του δικτύου.

- Περιμένει να λάβει κάποιο μήνυμα από γειτονικό κόμβο.
- Μόλις λάβει κάποιο μήνυμα το προωθεί στους γειτονικούς του κόμβους.
- Ελέγχει το είδος του μηνύματος.
- Αν το μήνυμα είναι τύπου Twitter, τότε αυτό αναλύεται και δίδεται στο Twitter Module.
- Το Twitter Module αναλαμβάνει να εκτελέσει την επικοινωνία με το Twitter εφόσον υπάρχει συνδεσιμότητα.
- Ο χρήστης της εφαρμογής δεν ενημερώνεται για τα αποτελέσματα της αίτησης.
- Αν πραγματοποιήθηκε η αίτηση, λαμβάνει ένα μήνυμα στο ηλεκτρονικό του ταχυδρομείο.

Η Twitter εφαρμογή είναι γραμμένη στη γλώσσα προγραμματισμού JAVA. Ωστόσο, έχει αναφερθεί πως παράλληλα με την Twitter εφαρμογή εκτελείται και η Υπηρεσία η οποία είναι υπεύθυνη για την προώθηση μηνυμάτων που λαμβάνονται από γειτονικές συσκευές. Η Υπηρεσία είναι επίσης γραμμένη στη γλώσσα προγραμματισμού JAVA. Και τα δυο συστατικά εμπεριέχουν ένα τμήμα όπου τους δίνεται η δυνατότητα επικοινωνίας με άλλες συσκευές οι οποίες εκτελούν το λογισμικό cBox. Για την επικοινωνία με τις συσκευές αυτές χρησιμοποιείται το Network Abstraction Layer (NAL) το οποίο είναι γραμμένο σε C++.

3.1.4. Ροή Αποστολής Μηνύματος στο Twitter

Για την αποστολή ενός μηνύματος στο Twitter ακολουθείται μια συγκεκριμένη διαδικασία. Η βασική ροή εργασιών που πραγματοποιείται για την αποστολή ενός μηνύματος στο Twitter παρουσιάζεται στη συνέχεια. Το πρώτο διάγραμμα ροής αναφέρεται στο cBox το οποίο πραγματοποιεί την δημιουργία και αποστολή του μηνύματος:



Η κύρια διαδικασία διαδικασία εκτέλεσης για την αποστολή ενός tweet περιλαμβάνει τα εξής βήματα:

1. Λήψη από το GUI του μηνύματος που επιθυμεί ο χρήστης να αποστείλλει.

2. Δημιουργία ενός Twitter object το οποίο χρησιμοποιείται για την εκτέλεση οποιασδήποτε λειτουργίας στο Twitter.

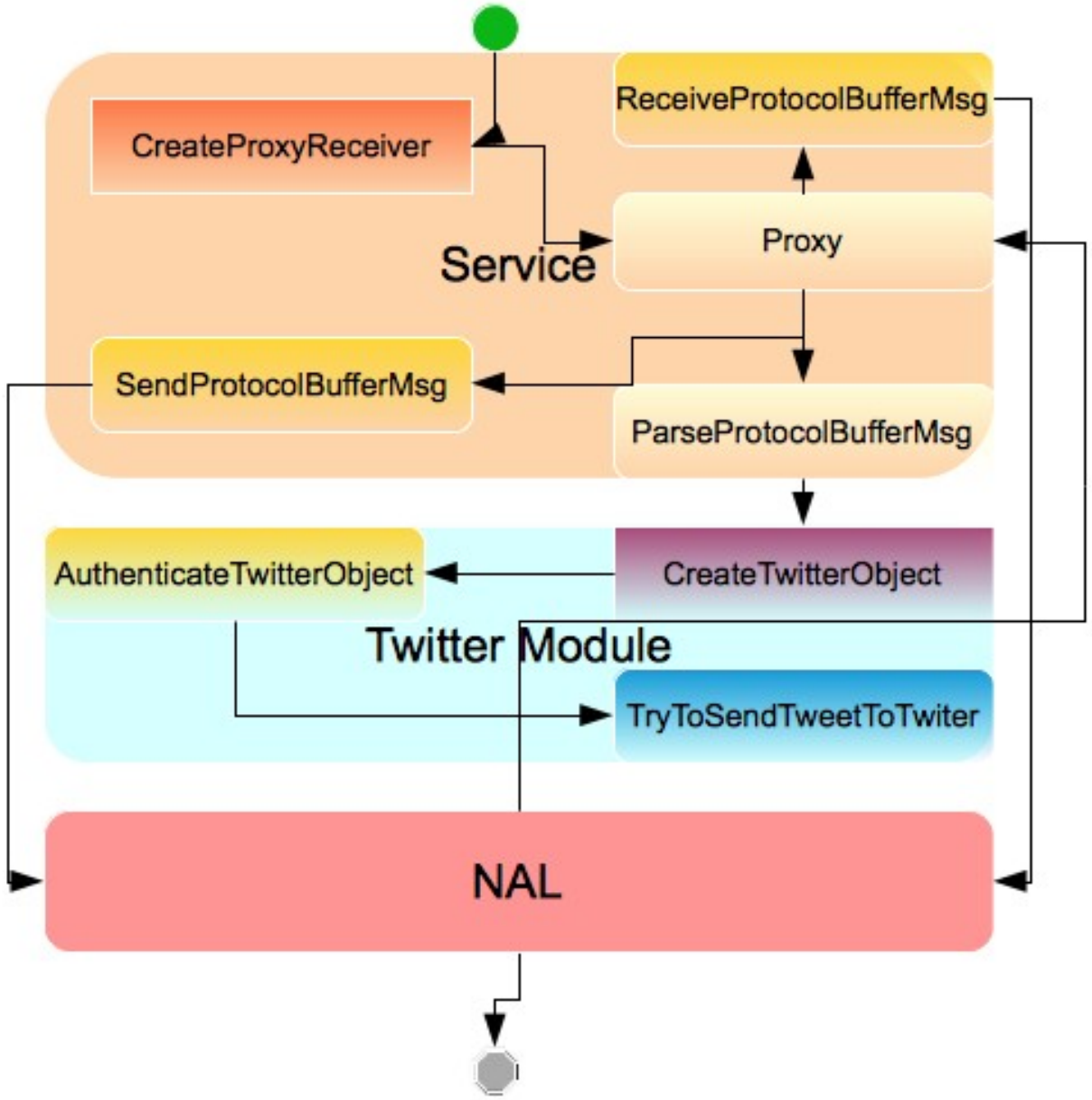
3. Εισαγωγή των OAuth διαπιστευτήριων στο Twitter object, τα οποία απαιτούνται για την δυνατότητα πραγματοποίησης πιστοποιημένων λειτουργιών με το Twitter.

4. Προσπάθεια αποστολής του tweet στο Twitter διαμέσου του ταυτοποιημένου Twitter object.

5. Δημιουργία ενος cBoxMessage (Protocol Buffers) τύπου TWITTER, το οποίο θα περιέχει το μήνυμα και τα OAuth διαπιστευτήρια.

6. Αποστολή του cBoxMessage στις γειτονικές συσκευές που εκτελούν το cBox μέσα από το NAL.

Το επόμενο διάγραμμα ροής αναφέρεται στο συστατικό της Υπηρεσίας. Το συστατικό αυτό εκτελείται σε κάθε εφαρμογή cBox, και είναι υπεύθυνο για την διαχείριση μηνυμάτων που λαμβάνονται από γειτονικές συσκευές που εκτελούν το cBox.



Η κύρια διαδικασία διαδικασία εκτέλεσης για την λήψη ενός tweet από την Υπηρεσία περιλαμβάνει τα εξής βήματα:

1. Δημιουργία ενός Proxy ο οποίος είναι υπεύθυνος για την λήψη αλλά και την διαχείριση ενός μηνύματος.

2. Ο Proxy εκτελεί μια συνάρτηση από τον NAL η οποία περιμένει να λάβει κάποιο μήνυμα.

3. Μόλις ο Proxy λάβει ένα cBoxMessage από το NAL, τότε εκτελεί μια λειτουργία για τον εντοπισμό του τύπου μηνύματος που έλαβε.

4. Αν το μήνυμα είναι τύπου TWITTER δημιουργεί ένα Twitter Object το οποίο χρησιμοποιείται για την εκτέλεση οποιασδήποτε λειτουργίας στο Twitter.

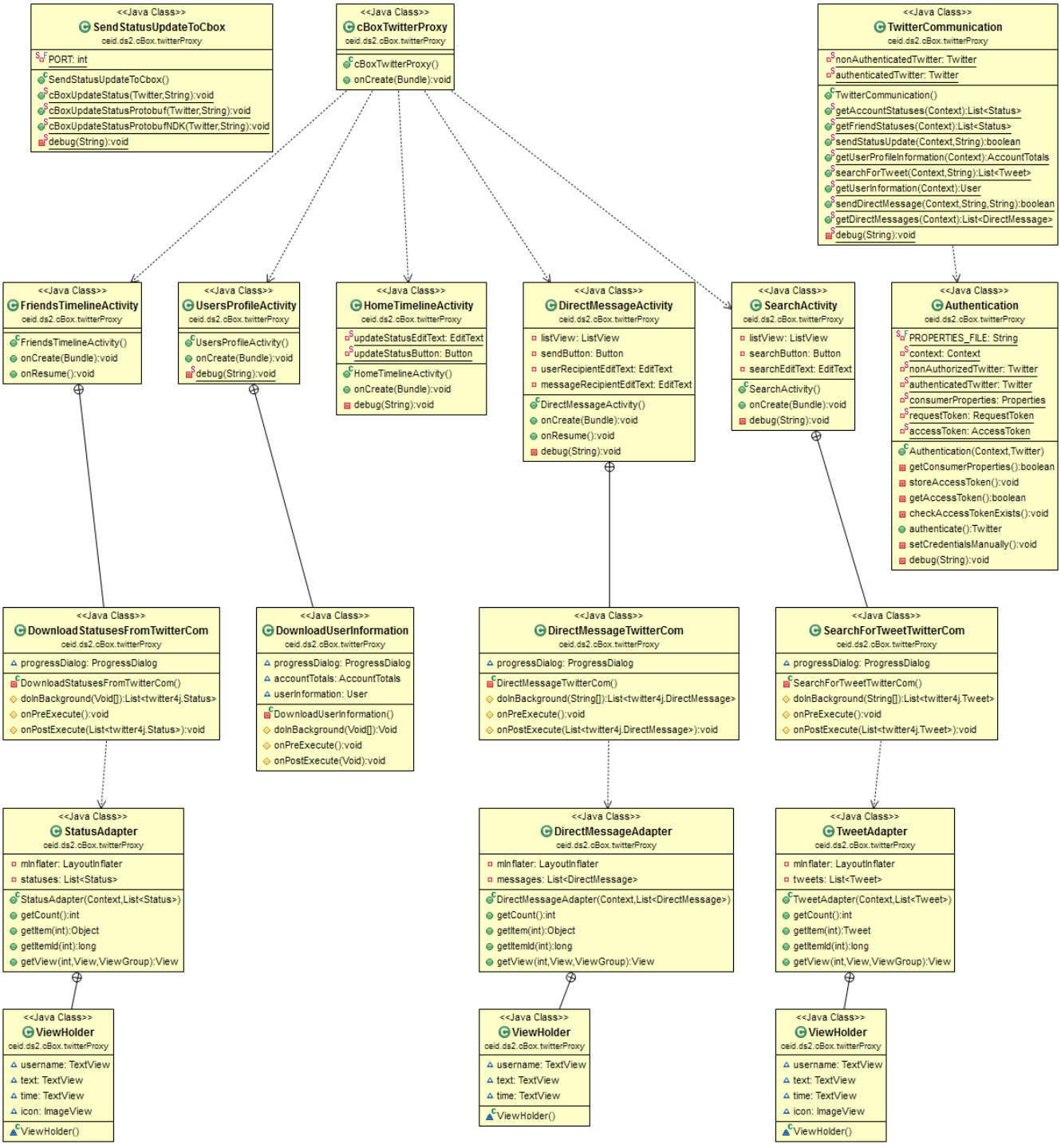
5. Εισαγωγή των OAuth διαπιστευτήρια στο Twitter object, τα οποία απαιτούνται για την δυνατότητα πραγματοποίησης πιστοποιημένων λειτουργιών με το Twitter.

6. Προσπάθεια αποστολής του tweet στο Twitter διαμέσου του ταυτοποιημένου Twitter object.

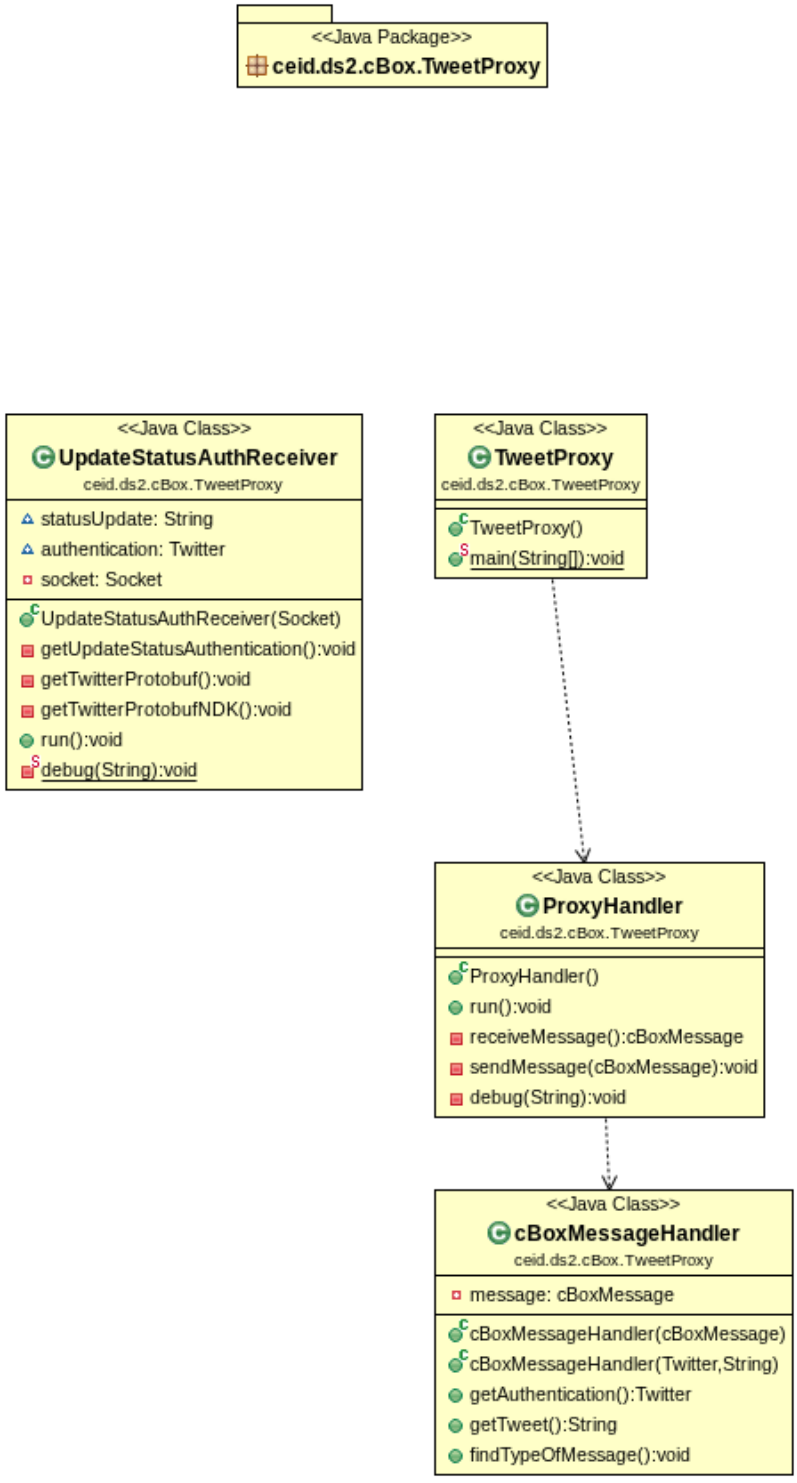
7. Τέλος αποστέλλει το cBoxMessage στις γειτονικές συσκευές μέσω του NAL.

3.1.5. Διάγραμμα Κλάσεων του Twitter Application

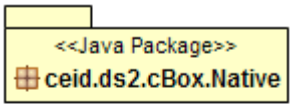
Το ***Twitter Application*** συνιστά την εφαρμογή, η οποία εκτελείται σε συσκευές με λειτουργικό σύστημα Android. Η εφαρμογή παρέχει ένα περιβάλλον διεπαφής για τον χρήστη με το Twitter. Ο τρόπος υλοποίησης της εφαρμογής αυτής παρουσιάζεται παρακάτω μέσω του διαγράμματος κλάσεων UML.

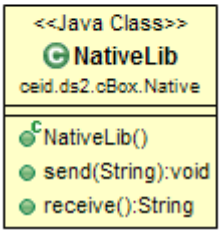


Βασικό τμήμα της εφαρμογής cBox είναι η ***Υπηρεσία***, η οποία προωθεί τα μηνύματα προς τις γειτονικές και όχι μόνο συσκευές. Η Υπηρεσία εκτελεί κάποια νήματα παράλληλα ώστε να έχει τη δυνατότητα να χειρίζεται πολλά εισερχόμενα μηνύματα. Στη συνέχεια παρουσιάζεται το διάγραμμα κλάσεων του συστατικού Υπηρεσία.



Για την επικοινωνία με τα χαμηλότερα επίπεδα απαιτείται η χρήση της βιβλιοθήκης JNI. Η κλήση των χαμηλότερων επιπέδων μέσω JNI στην ιεραρχία της εφαρμογής γίνεται με την χρήση του παρακάτω πακέτου:





3.2. Network Abstraction Layer (NAL)

Το *Network Abstraction Layer* αποτελεί το χαμηλότερο επίπεδο της εφαρμογής του cBox. Είναι το επίπεδο το οποίο διαχειρίζεται την αποστολή και την λήψη μηνυμάτων μεταξύ δυο συσκευών που εκτελούν το λογισμικό του cBox. Μεταβιβάζοντας ένα πακέτο τύπου cBoxMessage στο επίπεδο αυτό, ο προγραμματιστής δεν χρειάζεται να διαχειριστεί κάποια από τις διεπαφές που είναι διαθέσιμες στη συγκεκριμένη συσκευή. Το *NAL* είναι υπεύθυνο για την επιλογή των διαθέσιμων διεπαφών επικοινωνίας και μέσω αυτών να γίνει η αποστολή ή λήψη ενός μηνύματος cBoxMessage.

Η επικοινωνία μεταξύ των συσκευών που εκτελούν το λογισμικό του cBox πραγματοποιείται μέσω των ακόλουθων διεπαφών επικοινωνίας:

- ***Ethernet***
- ***WiFi***
- ***Bluetooth***

Το επίπεδο αυτό της εφαρμογής cBox είναι γραμμένο στη γλώσσα προγραμματισμού C++. Η χρήση του επιπέδου *NAL* πραγματοποιείται μέσω της διεπαφής που δίδεται από τη βιβλιοθήκη *Wiselib*. Ωστόσο για να είναι δυνατή η χρήση του επιπέδου *NAL* σε διάφορες πλατφόρμες, αφού το cBox υποστηρίζεται από διάφορα Λειτουργικά Συστήματα, θα πρέπει να γίνεται η κατάλληλη μεταγλώττιση της βιβλιοθήκης που περιλαμβάνει το *NAL*.

Κατά την χρήση του επιπέδου αυτού, υπάρχει πλήρης διαφάνεια κατά την επικοινωνία των κόμβων. Αυτό σημαίνει ότι οποιαδήποτε ενέργεια πραγματοποιείται για την αποστολή ή λήψη ενός μηνύματος μεταξύ των κόμβων του δικτύου δεν γίνεται αντιληπτή ούτε από τον χρήστη αλλά ούτε και από τον προγραμματιστή. Στον προγραμματιστή δίδεται μια βιβλιοθήκη η οποία υλοποιεί το παρόν επίπεδο, δίνοντας του τη δυνατότητα να την χρησιμοποιήσει χωρίς την ανάγκη γνώσης της ακριβούς υλοποίησης της.

Στον προγραμματιστή παρέχεται ένα Radio Model το οποίο παρέχει την απαραίτητη αφαίρεση στην επικοινωνία των κόμβων. Με τον τρόπο αυτό του δίνεται η δυνατότητα για:

- ***Ενεργοποίηση της διεπαφής***. Κατά την πραγματοποίηση αυτής της λειτουργίας κάθε διαθέσιμη διεπαφή επικοινωνίας Ethernet – WiFi – Bluetooth που υπάρχει διαθέσιμη στη συγκεκριμένη συσκευή ενεργοποιείται στο cBox. Με τον όρο ενεργοποίηση υπονοείται η δυνατότητα αποστολής και λήψης μηνυμάτων μέσω αυτής της διεπαφής.

- ***Αποστολή ενός μηνύματος μέσω της ενεργοποιημένης διεπαφής***. Κατά την

πραγματοποίηση αυτής της λειτουργίας, αποστέλλεται το δοθέν μήνυμα χρησιμοποιώντας κάθε διεπαφή που έχει ενεργοποιηθεί με την παραπάνω ενέργεια.

- ***Λήψης ενός μηνύματος μέσω της ενεργοποιημένης διεπαφής.*** Κατά την πραγματοποίηση αυτής της λειτουργίας, λαμβάνεται ένα μήνυμα από κάποια από τις ενεργοποιημένες διεπαφές που έχει ενεργοποιηθεί.

- ***Απενεργοποίηση της διεπαφής.*** Κατά την πραγματοποίηση αυτής της λειτουργίας κάθε ενεργοποιημένη διεπαφή επικοινωνίας απενεργοποιείται, με αποτέλεσμα να τερματιστεί η χρήση της από την εφαρμογή cBox.

Ο τρόπος με τον οποίο πραγματοποιούνται οι παραπάνω ενέργειες δεν είναι ορατός στον προγραμματιστή όμως μπορεί με απλά προγραμματιστικά βήματα να χρησιμοποιήσει τη συγκεκριμένη διεπαφή για να επικοινωνήσει με τις υπόλοιπες συσκευές του δικτύου. Το επίπεδο αυτό χρησιμοποιεί τα πρότυπα που ορίζονται από την βιβλιοθήκη Wiselib σχετικά με τα Models, τα Concepts και γενικότερα τον ορισμό των μεθόδων που παρέχουν τις απαραίτητες λειτουργίες.

3.2.1. Ροή αποστολής και λήψης μηνύματος

Η ροή αποστολής και λήψης ενός μηνύματος προς και από γειτονικές συσκευές που εκτελούν την εφαρμογή του cBox περιλαμβάνει κάποια βασικά βήματα. Ως γνωστόν το επίπεδο αυτό δεν είναι εμφανές στον χρήστη αλλά ούτε και στον προγραμματιστή.

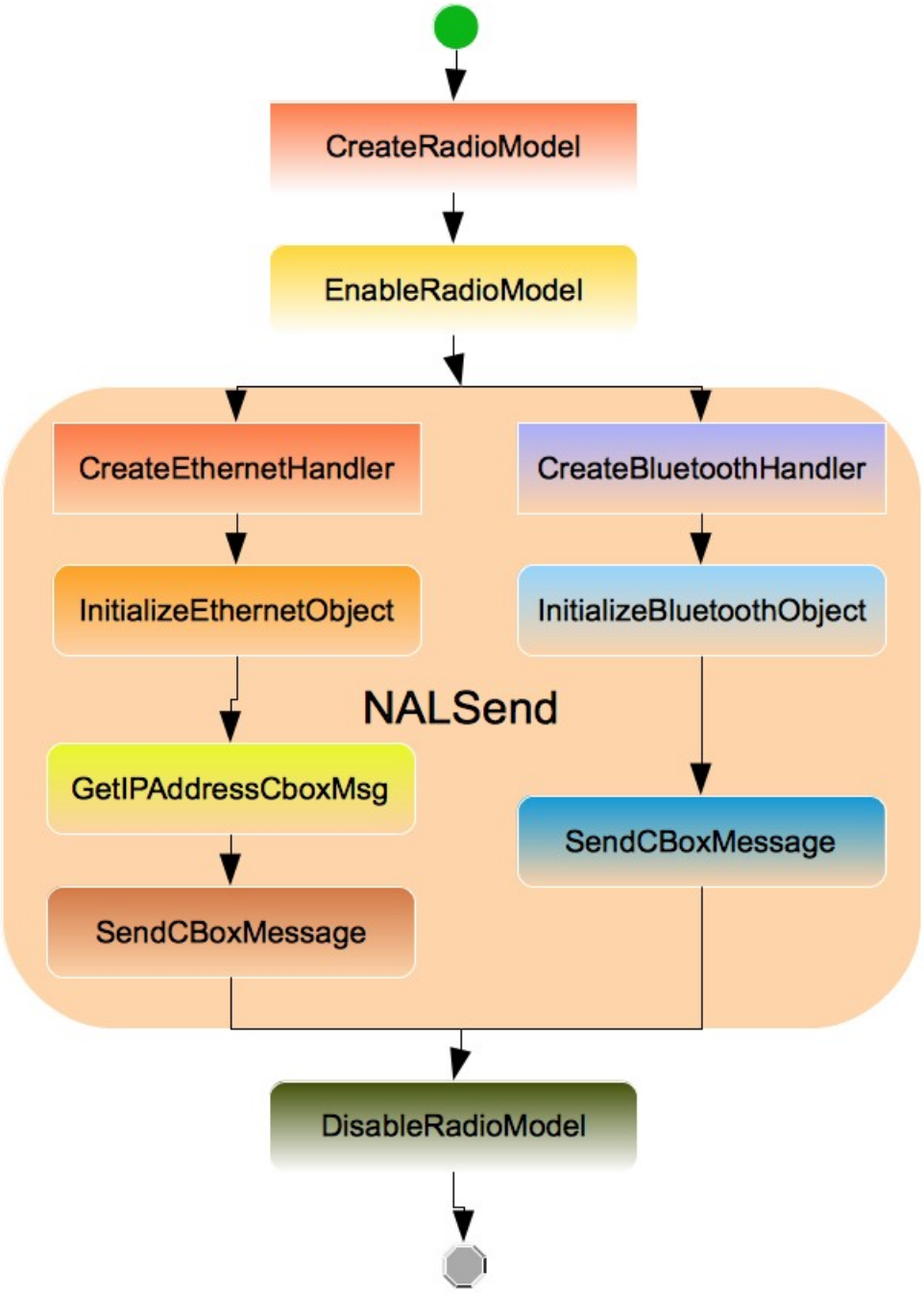


Diagram NAL Send

Τα κύρια σημεία κατά την αποστολή ενός μηνύματος μέσω του επιπέδου NAL είναι τα εξής:

1. ***Δημιουργία*** του Radio Model το οποίο είναι περιέχει όλη την λειτουργικότητα του επιπέδου αυτού.

2. ***Ενεργοποίηση*** του Radio Model.

3. ***Αποστολή*** του μηνύματος τύπου cBoxMessage.

    1. Αποστολή του μηνύματος μέσω Ethernet – WiFi.
        1. Δημιουργία του χειριστή των διεπαφών Ethernet – WiFi.
        2. Αρχικοποίηση του χειριστή.
        3. Λήψη της διευθύνσης στην οποία θα αποσταλλεί το μήνυμα.
        4. Αποστολή του μηνύματος.

    2. Αποστολή του μηνύματος μέσω Bluetooth.
        1. Δημιουργία του χειριστή της διεπαφής Bluetooth.
        2. Αρχικοποίηση του χειριστή.
        3. Αποστολή του μηνύματος.

4. ***Απενεργοποίηση*** του Radio Model.

Ακολουθεί η ροή λήψης ενός μηνύματος στο επίπεδο NAL.

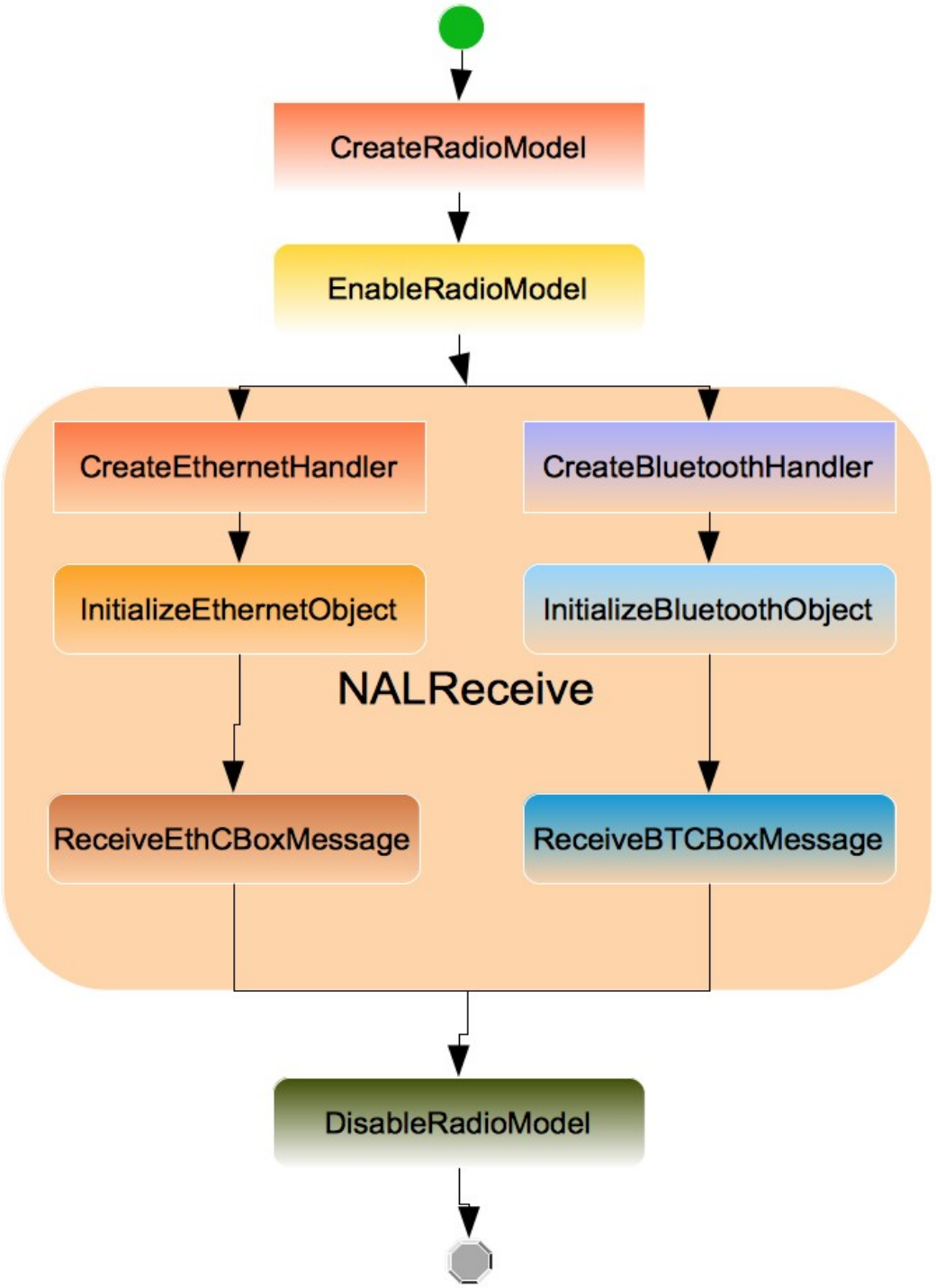


Diagram NAL Receive

Τα κύρια σημεία κατά την λήψη ενός μηνύματος μέσω του επιπέδου NAL είναι τα εξής:

1. ***Δημιουργία*** του Radio Model το οποίο είναι περιέχει όλη την λειτουργικότητα του επιπέδου αυτού.

2. ***Ενεργοποίηση*** του Radio Model.

3. ***Λήψη*** του μηνύματος τύπου cBoxMessage.

    1. Λήψη μηνύματος μέσω των διεπαφών Ethernet – WiFi.
        1. Δημιουργία του χειριστή των διεπαφών Ethernet – WiFi.
        2. Αρχικοποίηση του χειριστή.
        3. Λήψη του μηνύματος.

    2. Λήψη του μηνύματος μέσω της διεπαφής Bluetooth.
        1. Δημιουργία του χειριστή της διεπαφής Bluetooth.
        2. Αρχικοποίηση του χειριστή.
        3. Λήψη του μηνύματος.

4. ***Απενεργοποίηση*** του Radio Model.

3.2.2. Ενσωμάτωση του NAL στο NDK

Το NAL (Network Abstraction Layer) αποτελεί το χαμηλότερο επίπεδο της εφαμοργής cBox. Είναι το επίπεδο της εφαρμογής το οποίο είναι υπευθυνο για την επικοινωνία των συσκευών που εκτελούν το cBox. Η επικοινωνία αυτή υλοποιήθηκε στη γλώσσα προγραμματισμού C++. Το γεγονός αυτό οδηγεί στο συμπέρασμα ότι θα πρέπει να γίνει ενσωμάτωση του NAL στο NDK ώστε να είναι δυνατή η εκτέλεση της εφαρμογής cBox σε Android συσκευές. Για την ενσωμάτωση του NAL στο Android θα πρέπει να ελεγχθεί η συμβατότητα λειτουργίας του επιπέδου στο NDK.

- ***Ethernet***: Το NDK ***παρέχει*** την δυνατότητα χρήσης της τεχνολογίας των Sockets, άρα υπάρχει συμβατότητα με το επίπεδο NAL.

- ***WiFi***: Το NDK ***παρέχει*** την δυνατότητα χρήσης της τεχνολογίας των Sockets, άρα υπάρχει συμβατότητα με το επίπεδο NAL.

- ***Bluetooth***: Το NDK ***δεν παρέχει*** την δυνατότητα χρήσης της βιβλιοθήκης BlueZ την στιγμή που εγγράφεται η συγκεκριμένη εργασία, άρα ***δεν υπάρχει συμβατότητα*** με το επίπεδο NAL.

Για να είναι λοιπόν ***δυνατή η χρήση του Bluetooth*** μέσω του NDK έπρεπε να βρεθεί μια λύση για τον συγκεκριμένο σχεδιασμό του cBox. Για την εύρεση της λύσης έπρεπε να ληφθούν υπόψιν:

- Το επίπεδο NAL έχει γραφτεί στη γλώσσα προγραμματισμού C++ ώστε να είναι δυνατή η ενσωμάτωση της Wiselib.

- Ο σχεδιασμός της εφαρμογής επιτρέπει την επικονωνία ***μόνο*** μέσω του επιπέδου NAL. Δεν είναι δυνατή η μεταφορά της επικοινωνίας μέσω Bluetooth σε άλλο επίπεδο της εφαρμογής.

- Αδυναμία χρήσης της διεπαφής του Bluetooth μέσω του NDK.

- Δυνατότητα χρήσης της διεπαφής παρέχεται μόνο μέσω των διεπαφών της γλώσσας προγραμματισμού JAVA.

- Η πιο απλή και αξιόπιστη λύση για τη χρήση της διεπαφής του Bluetooth είναι η αξιοποίηση της βιβλιοθήκης ***BlueCove***.

- Η βιβλιοθήκη ***BlueCove*** ωστόσο είναι γραμμένη στη γλώσσα προγραμματισμού JAVA.

- Το NDK δίνει τη δυνατότητα να πραγματοποιείται από τη γλώσσα προγραμματισμού C++ κλήση πηγαίου κώδικα ο οποίος είναι γραμμένος στη γλώσσα προγραμματισμού JAVA.

Λαμβάνοντας υπόψιν τους παραπάνω λόγους εκπονήθηκε μια λύση, η οποία εξαλείφει τα προβλήματα που προέκυψαν. Η λύση αυτή είναι η μοναδική για την οποία ***δεν απαιτείται:***

- Αλλαγή του σχεδιασμού της εφαρμογής.
  - Αλλαγή της αρχιτεκτονικής των επιπέδων της εφαρμογής.
  - Αφαίρεση της συμβατότητας με τη βιβλιοθήκη Wiselib.
  - Αλλαγή της γλώσσας προγραμματισμού στην οποία υλοποιείται το επίπεδο NAL.

- Αφαίρεση της λειτουργικότητας της διεπαφής του Bluetooth από την εφαρμογή cBox.

- Αφαίρεση της δυνατότητας χρήσης της εφαρμογής στο Android.

Η λύση η οποία ικανοποιεί τις παραπάνω απαιτήσεις είναι η εκμετάλλευση της δυνατότητας κλήσης της πηγαίου κώδικα ο οποίος είναι γραμμένος στη γλώσσα προγραμματισμού JAVA σε συνδυασμό με την αξιόπιστη λύση της βιβλιοθήκης BlueCove.

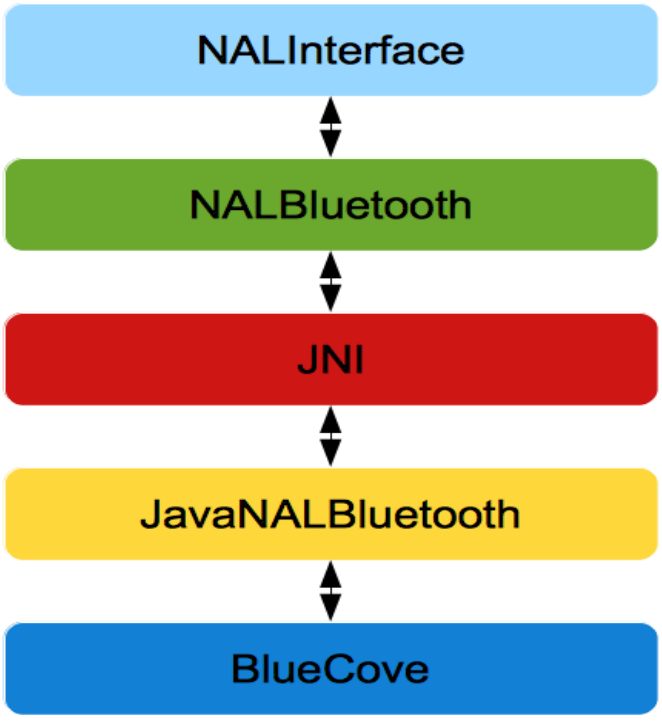


Diagram of Bluetooth Architecture

Έτσι δημιουργήθηκε μια υποτυπώδης βιβλιοθήκης η οποία παρέχει δύο μεθόδους:

- ***Send***. Παρέχει τη δυνατότητα αποστολής ενός μηνύματος μεσω Bluetooth.
- ***Receive***. Παρέχει τη δυνατότητα λήψης ενός μηνύματος μεσω Bluetooth.

Η βιβλιοθήκη αυτή είναι γραμμένη στη γλώσσα προγραμματισμού JAVA. Εκμεταλλεύεται πλήρως την αξιόπιστη, σταθερή και συμβατή με πολλά συστήματα βιβλιοθήκη BlueCove για την αλληλεπίδραση με τη διεπαφή του Bluetooth. Η κλήση της γίνεται με τη χρήση της τεχνολογίας JNI που παρέχεται από το NDK.

Ενσωματώνοντας στο επίπεδο NAL την βιβλιοθήκη αυτή που δημιουργήθηκε, προστέθηκε η ***δυνατότητα χρήσης*** και της διεπαφής του ***Bluetooth***. Μαζί με την βιβλιοθήκη παρέχεται και μια διεπαφή η οποία είναι γραμμένη στη γλώσσα προγραμματισμού C++, η οποία επιτρέπει την άμεση ενσωμάτωση της στο επίπεδο NAL. Η διεπαφή πραγματοποιεί τις απαραίτητες ενέργειες για την κλήση της JAVA βιβλιοθήκης μέσω του NAL.

3.2.3. Ροή αποστολής & λήψης μηνύματος μέσω Bluetooth

Λόγω του περιορισμού χρήσης του Bluetooth μέσω NDK, προτάθηκε μια λύση που εξαλείφει τον περιορισμό αυτόν. Η λύση αυτή ωστόσο απαιτεί ιδιαίτερο χειρισμό αλλά και σχεδιασμό. Για τον λόγω αυτό γίνεται ειδική αναφορά για τον τρόπο αποστολής αλλά και λήψης μηνύματος μέσω Bluetooth.

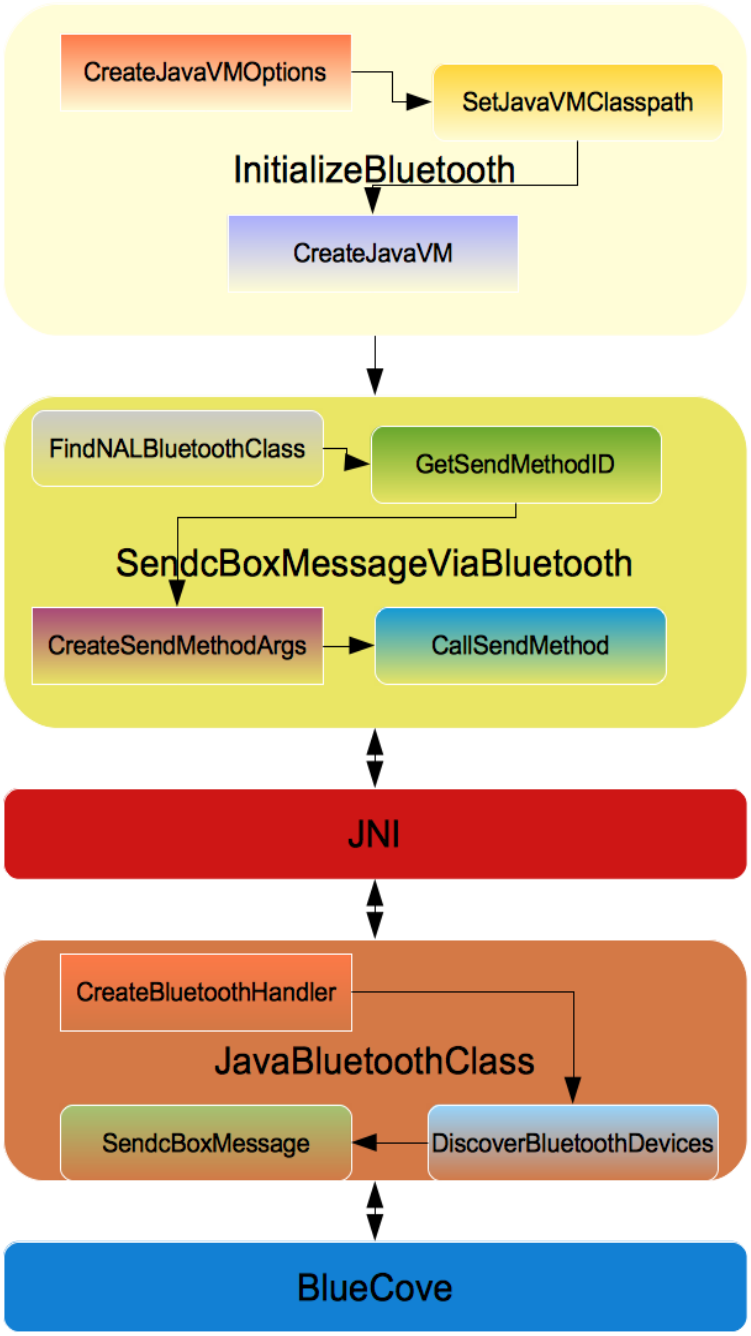


Diagram NALBluetooth Send

Τα κύρια βήματα που πραγματοποιούνται κατά την αποστολή ενός μηνύματος cBoxMessage μέσω Bluetooth:

1. Για να γίνει χρήση του Bluetooth μέσα από το NAL, θα πρέπει αρχικά να γίνει κλήση του μεσολαβητή ανάμεσα στη γλώσσα προγραμματισμού C++ και τη γλώσσα προγραμματισμού JAVA.

    1. Ο μεσολαβητής αυτός ορίζει τις παραμέτρους που θα δωθούν στο JAVA VM που πρέπει να δημιουργηθεί.

    2. Στις παραμέτρους αυτές περιλαμβάνεται το Classpath στο οποίο έχουν αποθηκευθεί:
        - η JAVA βιβλιοθήκη που χειρίζεται τη βιβλιοθήκη BlueCove
        - η βιβλιοθήκη BlueCove
        - η GPL έκδοση της βιβλιοθήκης BlueCove

    3. Αφού οριστούν οι παράμετροι, δημιουργείται το JAVA VM το οποίο απαιτείται για την εκτέλεση του JAVA κώδικα.

2. Στη συνέχεια χρησιμοποιώντας τον μεσολαβητή καλείται η συνάρτηση η οποία είναι υπεύθυνη για την αποστολή του μηνύματος. Στη συνάρτηση που είναι υπεύθυνη για την αποστολή του μηνύματος:

    1. Αρχικά εντοπίζεται η JAVA κλάση στην οποία ανήκει η μέθοδος η οποία πρέπει να καλεστεί.

    2. Στη συνέχεια εντοπίζεται η μέθοδος που πρέπει να καλεστεί.

    3. Ακόμη καθορίζεται η παράμετρος που πρέπει να αποσταλλεί. Πιο συγκεκριμένα ορίζεται το μήνυμα που πρέπει να αποσταλλεί.

    4. Τέλος καλείται η JAVA μέθοδος μέσω του JNI με παράμετρο το μήνυμα που πρέπει να σταλεί.

3. Τέλος καλείται η JAVA μέθοδος η οποία αποστέλλει ένα δοθέν μήνυμα.

    1. Η μέθοδος αυτή εντοπίζει τις γειτονικές συσκευές οι οποίες έχουν την διεπαφή του Bluetooth τους σε Discoverable Mode.

    2. Προσπαθεί να συνδεθεί στις συσκευές αυτές.

    3. Αν πραγματοποιηθεί επιτυχής σύνδεση, αποστέλλει το μήνυμα.

Στη συνέχεια παρουσιάζεται αναλυτικά η ροή εκτέλεσης για την λήψη ενός μηνύματος cBoxMessage μέσω Bluetooth.

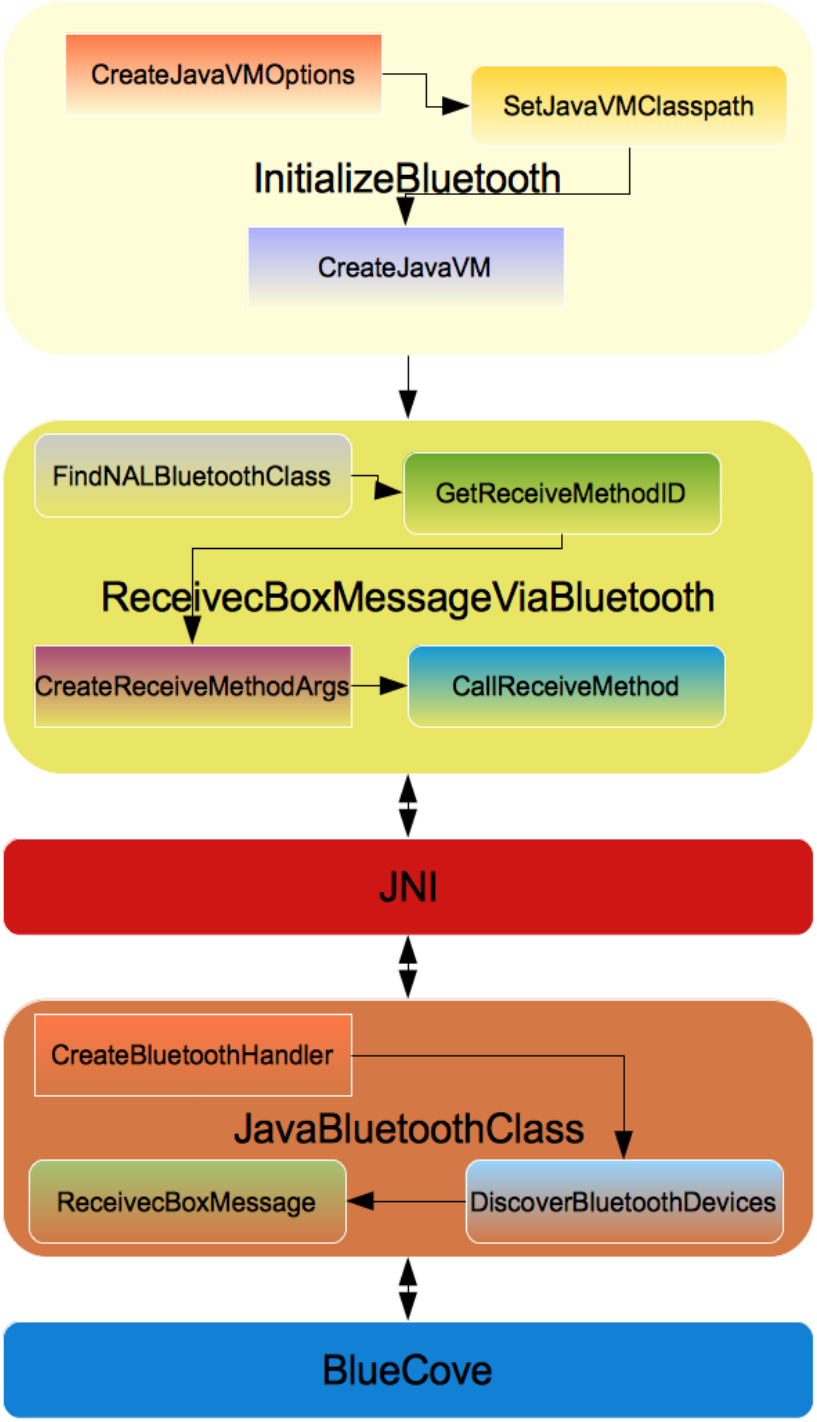


Diagram NALBluetooth Receive

Τα κύρια βήματα που πραγματοποιούνται κατά την λήψη ενός μηνύματος cBoxMessage μέσω Bluetooth:

4. Για να γίνει χρήση του Bluetooth μέσα από το NAL, θα πρέπει αρχικά να γίνει κλήση του μεσολαβητή ανάμεσα στη γλώσσα προγραμματισμού C++ και τη γλώσσα προγραμματισμού JAVA.

    1. Ο μεσολαβητής αυτός ορίζει τις παραμέτρους που θα δωθούν στο JAVA VM που πρέπει να δημιουργηθεί.

    2. Στις παραμέτρους αυτές περιλαμβάνεται το Classpath στο οποίο έχουν αποθηκευθεί:
        - η JAVA βιβλιοθήκη που χειρίζεται τη βιβλιοθήκη BlueCove
        - η βιβλιοθήκη BlueCove
        - η GPL έκδοση της βιβλιοθήκης BlueCove

    3. Αφού οριστούν οι παράμετροι, δημιουργείται το JAVA VM το οποίο απαιτείται για την εκτέλεση του JAVA κώδικα.

5. Στη συνέχεια χρησιμοποιώντας τον μεσολαβητή καλείται η συνάρτηση η οποία είναι υπεύθυνη για την λήψη του μηνύματος. Στη συνάρτηση που είναι υπεύθυνη για την λήψη του μηνύματος:

    1. Αρχικά εντοπίζεται η JAVA κλάση στην οποία ανήκει η μέθοδος η οποία πρέπει να καλεστεί.

    2. Στη συνέχεια εντοπίζεται η μέθοδος που πρέπει να καλεστεί.

    3. Τέλος καλείται η JAVA μέθοδος μέσω του JNI.

6. Τέλος καλείται η JAVA μέθοδος η οποία λαμβάνει ένα εισερχόμενο μήνυμα.

    1. Η μέθοδος αυτή δίνει τη δυνατότητα στις γειτονικές συσκευές οι οποίες βρίσκονται σε εμβέλεια να αποστείλουν στην συσκευή αυτή ένα μήνυμα.

2. Περιμένει κάποια εισερχόμενη σύνδεση.

3. Μόλις πραγματοποιήσει κάποια σύνδεση, τότε δέχεται το μήνυμα που της αποστέλλει η γειτονική συσκευή.

3.2.4. Ροή αποστολής & λήψης μηνύματος μέσω Ethernet – WiFi

Οι διεπαφές Ethernet και WiFi χρησιμοποιούν το ίδιο τμήμα κώδικα για την επικοινωνία με άλλες συσκευές. Και οι δύο διεπαφές απαιτούν κάποιο πρωτόκολλο δρομολόγησης για να είναι δυνατή η επικοινωνία τους με άλλες συσκευές. Επίσης το NDK δίνει τη δυνατότητα στο NAL να εκτελέσει το συγκεκριμένο τμήμα χωρίς την εμφάνιση προβλημάτων παρόμοιων με αυτών που παρουσιάστηκαν στη διεπαφή Bluetooth.

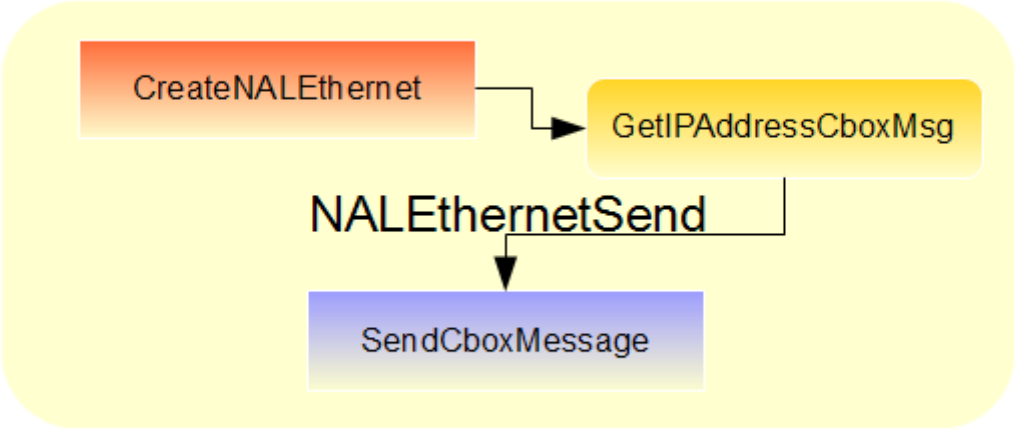


Diagram NALEthernet Send

Στη συνέχεια παρουσιάζονται τα βασικά βήματα που ακολουθούνται για την αποστολή ενός μηνύματος τύπου cBoxMessage:

1. Δημιουργία του χειριστή της διεπαφής Ethernet – WiFi.

2. Λήψη της διεύθυνσης στην οποία θα αποσταλλεί στο μήνυμα.

3. Αποστολή του μηνύματος τύπου cBoxMessage στη διευθυνση που αναφέρει.

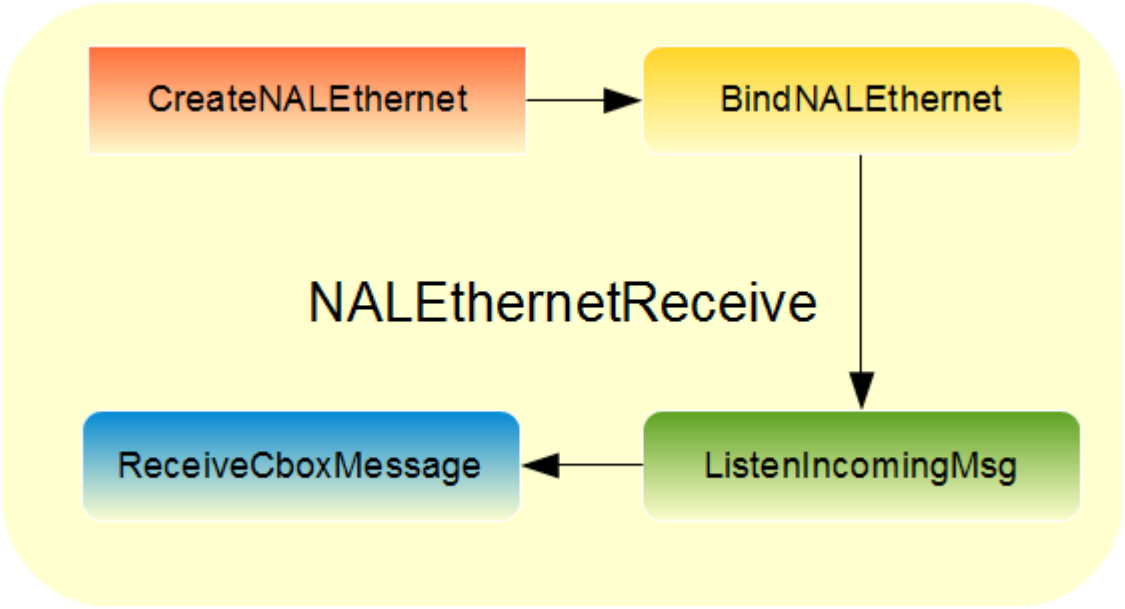


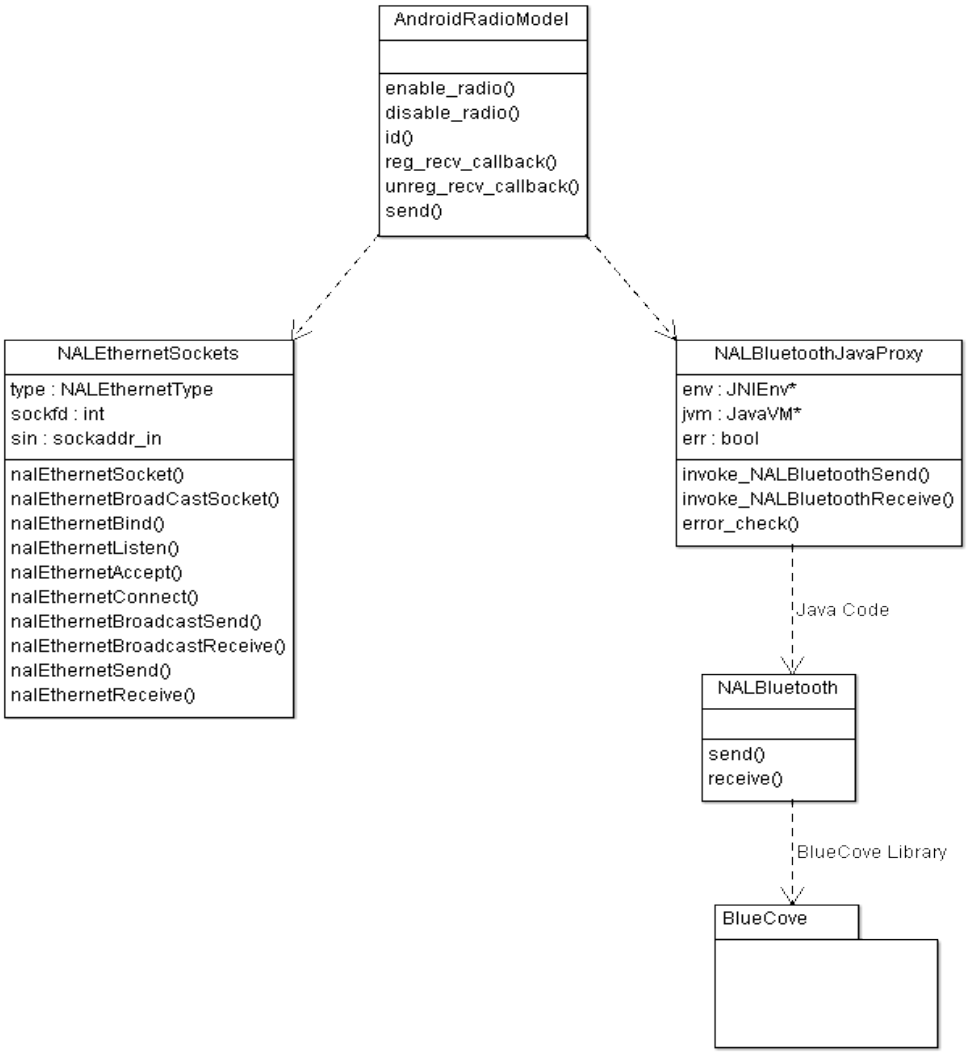
Diagram NALEthernet Receive

Στη συνέχεια αναφέρονται τα βασικά σημεία κατά την λήψη ενός μηνύματος τύπου cBoxMessage:

1. Δημιουργία του χειριστή διεπαφής Ethernet – WiFi.
2. Αρχικοποίηση του σε συγκεκριμένη port (9876).
3. Αναμονή για λήψη ενός μηνύματος τύπου cBoxMessage.

3.2.5. Διάγραμμα Κλάσεων του NAL

Το NAL αποτελεί το χαμηλότερο επίπεδο της εφαρμογής cBox και χρησιμοποιείται για την επικοινωνία των συσκευών που εκτελούν το λογισμικό αυτό. Λαμβάνοντας υπόψιν τα εμπόδια κατά τη διαδικασία σχεδιασμού του συγκεκριμένου επιπέδου, στη συνέχεια παρατίθεται το διάγραμμα κλάσεων του συγκεκριμένου επιπέδου.



4. Οδηγίες προς τον προγραμματιστή

Το λογισμικό του cBox έχει αναπτυχθεί χρησιμοποιώντας διάφορα συστατικά. Τα συστατικά αυτά βοηθούν τον προγραμματιστή στην ευκολότερη αλλά και ορθότερη υλοποίηση της εφαρμογής. Ωστόσο για την ευκολότερη κατανόηση του πηγαίου κώδικα που έχει αναπτυχθεί αλλά και για την δυνατότητα επέκτασης της εφαρμογής δίνονται οι παρακάτω οδηγίες για την ανάπτυξη των συστατικών της εφαρμογής.

4.1.1. Εγκατάσταση NDK

Για την εγκατάσταση του NDK θα πρέπει ο προγραμματιστής να έχει καταβιβάσει και να έχει εγκαταστήσει πρωτίστως το Software Development Kit (SDK). Στη συνέχεια:

1. Καταβίβαση του NDK από τον σύνδεσμο "https://developer.android.com/sdk/ndk/index.html" για το κατάλληλο λειτουργικό σύστημα.

2. Αποσυμπίεση του αρχείου σε κάποιο φάκελο, κατά προτίμηση να μην υπάρχει κενό στη διαδρομή του φακέλου.

4.1.2. Δημιουργία Android Εφαρμογής με χρήση NDK

Για την δημιουργία μιας εφαρμογής Android η οποία θα χρησιμοποιεί το NDK θα πρέπει να ακολουθηθεί η παρακάτω διαδικασία. Η διαδικασία που θα αναφερθεί προϋποθέτει την χρήση του εργαλείου ανάπτυξης Eclipse IDE. Στην εφαρμογή που περιγράφεται παρακάτω, γίνεται κλήση μιας συνάρτησης σε native κώδικα η οποία προσθέτει δυο αριθμούς.

1. Δημιουργία ενός Android Project

2. Δημιουργία μιας JAVA κλάσης που να περιέχει τους ορισμούς για τις native μεθόδους.

```
package gr.example.demo;

public class NativeLib {
  static {
    System.loadLibrary("ndk_demo");
  }

  /**
   * Adds two integers, returning their sum
   */
  public native int add( int v1, int v2 );
}
```

3. Μέσα στο φάκελο /bin, εκτέλεση της εντολής κέλυφους : “javah -jni gr.example.demo.Nativelib” για να παραχθεί το header file των native συναρτήσεων.

4. Μετακίνηση του αρχείου gr_example_demo_Nativelib.h που παράχθηκε στον φάκελο <project>/jni (Αν δεν υπάρχει θα πρέπει να δημιουργηθεί).

5. Μέσα στο φάκελο <project>/jni θα πρέπει να δημιουργηθεί και το πηγαίο αρχείο, με τον κώδικα των συναρτήσεων, στο οποίο θα πρέπει να γίνει include το header file που δημιουργήθηκε παραπάνω.

```
#include "gr_example_demo_Nativelib .h"

JNIEXPORT jint JNICALL Java_gr_example_demo_Nativelib_add
  (JNIEnv * env, jobject obj, jint value1, jint value2) {
                return (value1 + value2);
}
```

6. Δημιουργία ενός αρχείου Android.mk μέσα στον φάκελο <project>/jni.

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE    := ndk_demo
LOCAL_SRC_FILES := ndk_demo.c
include $(BUILD_SHARED_LIBRARY)
```

7. Δημιουργία ενός αρχείου Application.mk μέσα στον φάκελο <project>/jni.

```
APP_PROJECT_PATH := $(call my-dir)/project
APP_MODULES      := ndk_demo
```

8. Δημιουργία μιάς κλάσης JAVA από την οποία θα γίνει η κλήση του native κώδικα.

```
public class NDKDemo extends Activity {
  NativeLib nativeLib;
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    nativeLib = new NativeLib();
    int sum = nativeLib.add(1,2);
  }
}
```

9. Με εντολές κέλυφους, μετακίνηση στον φάκελο <project>/jni, κλήση του εργαλείου ndk-build. Δηλαδή “cd <project>/jni” και στη συνέχεια “<ndk-path>/ndk-build”, για να παραχθεί η native βιβλιοθήκη.

10. Εκτέλεση του project.

4.1.3. Οδηγίες ανάπτυξης εφαρμογής με Twitter4J

Το Twitter4J αποτελεί μια βιβλιοθήκη η οποία δίνει τη δυνατότητα στον προγραμματιστή να αξιοποιήσει με εύκολο και γρήγορο τρόπο τις δυνατότητες που δίνει το Twitter API. Για την ανάπτυξη εφαρμογής που χρησιμοποιεί αυτή τη βιβλιοθήκη στο εργαλείο Eclipse IDE θα πρέπει να πραγματοποιηθούν τα εξής βήματα:

1. Καταβίβαση της βιβλιοθήκης από τον ιστότοπο "http://twitter4j.org/en/twitter4j-2.2.5.zip". Αν η εφαρμογή προορίζεται για το λειτουργικό σύστημα Android, έχει αναπτυχθεί ειδική, πιο ελαφριά έκδοση της βιβλιοθήκης στον ιστότοπο "http://twitter4j.org/en/twitter4j-android-2.2.5.zip".

2. Αποσυμπίεση του αρχείου μέσα στο φάκελο του Project.

3. Δεξί κλικ σε κάθε βιβλιοθήκη (.jar)

4. Επιλογή "Build Path"

5. Επιλογή "Add to Build Path"

4.1.4. Παράδειγματα χρήσης του Twitter4J

Το Twitter4J συνιστά μια βιβλιοθήκη αξιοποίησης του Twitter API και διευκόλυνσης του προγραμματιστή που επιθυμεί να το ενσωματώσει στην εφαρμογή του. Η βιβλιοθήκη αυτή, αποκρύπτει τον χειρισμό των HTTP ενεργειών και την δημιουργία των κατάλληλων μορφών ώστε να είναι συμβατο με το Twitter API. Στη συνέχεια παρουσιάζονται κάποια παραδείγματα χρήσης της βιβλιοθήκης Twitter4J.

- Αποστολή Tweet

  Στην περίπτωση που ο προγραμματιστής θέλει να αποστείλει ένα tweet που να γράφει “Hello”

```
// The factory instance is re-useable and thread safe.
Twitter twitter = new TwitterFactory().getInstance();
Status status = twitter.updateStatus(“Hello”);
System.out.println(“Return status: “ + status.getText());
```

- Λήψη του Timeline

  Στην περίπτωση που ο προγραμματιστής θέλει να λάβει το Timeline

```
// The factory instance is re-useable and thread safe.
Twitter twitter = new TwitterFactory().getInstance();
List<Status> statuses = twitter.getFriendsTimeline();
for (Status status : statuses) {
    System.out.println(status.getUser().getName() + “:” + status.getText());
}
```

- Αποστολή και λήψη απευθείας μηνυμάτων

```
// The factory instance is re-useable and thread safe.
Twitter sender = new TwitterFactory().getInstance();
DirectMessage message = sender.sendDirectMessage(recipientId, “Hello”);
System.out.println(“@” + message.getRecipient() + “:” + message.getText());
```

- Αναζήτηση Tweets

```
// The factory instance is re-useable and thread safe.
Twitter twitter = new TwitterFactory().getInstance();
Query query = new Query(“Universities”);
QueryResult result = twitter.search(query);
for (Tweet tweet : result.getTweets()) {
   System.out.println(“@” + tweet.getFromUser() + “:” + tweet.getText());
}
```

4.1.5. Οδηγίες ανάπτυξης Protocol Buffers στο Eclipse IDE

Το Eclipse IDE αποτελεί ίσως το δημοφιλέστερο εργαλείο στην κατηγορία του για ανάπτυξη εφαρμογών, κυρίως στη γλώσσα προγραμματισμού JAVA. Έτσι λοιπόν για την ευκολότερη ανάπτυξη εφαρμογών που χρησιμοποιούν τα *Protocol Buffers* έχουν αναπτυχθεί κάποια πρόσθετα τα οποία εισάγονται στο Eclipse. Παρακάτω παρατίθονται οδηγίες για την προετοιμασία του Eclipse ώστε να είναι δυνατή η ανάπτυξη λογισμικού με *Protocol Buffers*.

1. Καταβίβαση και εγκατάσταση του πρόσθετου (plugin) " protoclipse " για το Eclipse. Στο Eclipse εκτελούμε:
   1. Επιλογή "Help"
   2. Επιλογή "Install New Software"
   3. Εισαγωγή του συνδέσμου "http://protoclipse.googlecode.com/svn/trunk/site/"
   4. Επιλογή "Add" και μετά "Next" εως ώτου ολοκληρωθεί η εγκατάσταση.

2. Καταβίβαση από τον ιστότοπο (Αναλόγως βέβαια και η έκδοση των Protocol Buffers) "grepcode.com/snapshot/repo1.maven.org/maven2/com.google.protobuf-java/2.4.1" της βιβλιοθήκης "protobuf-java-2.4.1.jar"

3. Προσθήκη της βιβλιοθήκης "protobuf-java-2.4.1.jar" στο Build Path
   1. Αντιγραφή και επικόλληση μέσα στο Project της βιβλιοθήκης
   2. Ανανέωση του Project
   3. Δεξί κλικ πάνω στη βιβλιοθήκη
   4. Επιλογή "Build Path"
   5. Επιλογή "Add to Build Path"

4. Ορισμός της διαδρομής που αντιστοιχεί στον μεταγλωττιστή protoc
   1. Επιλογή "Window"
   2. Επιλογή "Preferences"
   3. Επιλογή "Protobuf"
   4. Ορισμός της διαδρομής που αντιστοιχεί στον μεταγλωττιστή protoc

5. Δημιουργία ενός αρχείου μέσα στο src/ με κατάληξη .proto που περιέχει τον ορισμό του μηνύματος του Protocol Buffer.

6. Προσθήκη αυτόματου μεταγλωττιστή στο Project

1. Δεξί κλικ πάνω στο Project
2. Επιλογή “Add Google Protobuf Compiler Nature”

4.1.6. Οδηγίες ανάπτυξης BlueCove στο Eclipse IDE

Το BlueCove αποτελεί μια Java βιβλιοθήκη για Bluetooth (JSR-82 υλοποίηση) και παρέχει διεπαφή για διάφορα λειτουργικά συστήματα. Για την χρήση της βιβλιοθήκης σε Linux συστήματα απαιτείται η καταβίβαση και της GPL έκδοσης της βιβλιοθήκης BlueCove-GPL.

1. Καταβίβαση της βιβλιοθήκης από τον σύνδεσμο “http://code.google.com/p/bluecove/downloads/list”.

2. Μετακίνηση του αρχείου .jar στον φάκελο του project.

3. Ανανέωση του project στο Eclipse IDE.

4. Δεξί κλικ σε κάθε βιβλιοθήκη (.jar).

5. Επιλογή “Build Path”.

6. Επιλογή “Add to Build Path”.

4.1.7. Οδηγίες ανάπτυξης Base64Coder στο Eclipse IDE

Το Base64Coder αποτελεί μια Java βιβλιοθήκη που παρέχει ένα γρήγορο και συμπαγή τρόπο κωδικοποίησης και αποκωδικοποίησης σε Base64. Για την χρήση της βιβλιοθήκης απαιτούνται τα παρακάτω βήματα.

1. Καταβίβαση της βιβλιοθήκης από τον σύνδεσμο "http://www.source-code.biz/base64coder/java/base64coder.zip".

2. Αποσυμπίεση του αρχείου.

3. Μετακίνηση του αρχείου .jar στον φάκελο του project.

4. Ανανέωση του project στο Eclipse IDE.

5. Δεξί κλικ σε κάθε βιβλιοθήκη (.jar).

6. Επιλογή "Build Path".

7. Επιλογή "Add to Build Path".

5. Επίλογος

Το cBox συνιστά ένα ολοκληρωμένο λογισμικό, το οποίο δίνει τη δυνατότητα σε συσκευές που χρησιμοποιούν διαφορετικές πλατφόρμες να επικοινωνούν με αξιόπιστο τρόπο. Το σύστημα αυτό επιτυγχάνει την ανεξάρτητη επικοινωνία των κόμβων του δικτύου χωρίς την απαραίτητη χρήση του διαδικτύου ή άλλων παραπλήσιων δομών δικτύου. Με την χρήση των διεπαφών επικοινωνίας που βρίσκονται διαθέσιμες σε κάθε συσκευή παρέχεται στους χρήστες του η λειτουργικότητα της ανταλλαγής μηνυμάτων χωρίς την εξάρτηση άλλων δικτύων. Έτσι προσφέρεται ένας νέος εναλλακτικός τρόπος επικοινωνίας, ο οποίος έχοντας αναπτυχθεί με ανοιχτό κώδικα παρέχει ασφάλεια, ιδιωτικότητα και καμία εξάρτηση από τρίτους.

Το cBox σύστημα αποτελεί έναν προσωπικό εξυπηρετητή ο οποίος μπορεί να διαχειριστεί θέματα όπως η συνδεσιμότητα με νέες συσκευές, η δρομολόγηση πακέτων, η ανωνυμία στο δίκτυο, το web caching και την διεπαφή με τον χρήστη. Αποτελείται από επίπεδα τα οποία συντελούν στην ευκολότερη ανάπτυξη αλλά και τροποποίηση του λογισμικού, δίνοντας έτσι τη δυνατότητα σε προγραμματιστές να το χρησιμοποιήσουν για έλεγχο εκτέλεσης κάποιου αλγόριθμου σε πραγματικό χρόνο. Είναι ένα σύστημα το οποίο λόγω της συμβατότητας του μπορεί να χρησιμοποιηθεί στην πλειονότητα των συσκευών σήμερα.

5.1. Μελλοντική Εργασία

Το cBox αποτελεί μια πολλά υποσχόμενη εφαρμογή η οποία μπορεί να χρησιμοποιηθεί σε ετερογενή δίκτυα για την επικονωνία μεταξύ των κόμβων. Η αρχική ανάπτυξη του βασίστηκε στην αξιοπιστία παράδοσης των μηνυμάτων που έχουν αποσταλλεί. Έτσι ο σχεδιασμός του λογισμικού έγινε με βάση το παραπάνω κριτήριο. Ως μελλοντική εργασία θα ήταν δυνατό να τεθούν θέματα όπως:

- ***Απόδοση***: για την βελτίωση της απόδοσης κατά την επικοινωνία των συσκευών απαιτείται η χρήση αλγόριθμων δρομολόγησης πακέτων. Οι αλγόριθμοι αυτοί, ανάλογα με τις πολυπλοκότητες που επιτυγχάνουν έχουν την δυνατότητα να αυξήσουν την απόδοση του δικτύου.
- ***Εφαρμογές***: το cBox αναπτύχθηκε έχοντας ως αρχική εφαρμογή την διεπαφή με το Twitter. Ωστόσο η ανάπτυξη της εφαρμογής cBox έχει πραγματοποιηθεί με τέτοιο τρόπο ώστε να είναι δυνατή η ενσωμάτωση και άλλων εφαρμογών όπως το Flickr ή ακόμη και νέων εφαρμογών.
- ***ZeroConf – Routing***: τα επίπεδα αυτά κατά την ανάπτυξη της πρότυπης εφαρμογής του cBox παραλείφθηκαν. Ωστόσο αποτελούν δύο πεδία με ευρή φάσμα έρευνας και τα οποία δίνουν την δυνατότητα στον προγραμματιστή να προσθέσει έναν αλγόριθμο δρομολόγησης και να εισάγει μια λύση για το πρόβλημα του namespace.

- ***iOS***: δίνεται η δυνατότητα επέκτασης της εφαρμογής σε νέες platformes όπως το iOS. Μέχρι σήμερα υπάρχει μια υλοποίηση για το NAL επίπεδο η οποία έχει εισαχθεί και στην δοκιμαστική έκδοση της βιβλιοθήκης Wiselib. Η υλοποίηση αυτή μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογής η οποία θα είναι συμβατή με το cBox.
- ***Proxy***: ένας υποτυπώδης Proxy Server έχει υλοποιηθεί και έχει ενσωματωθεί στην εφαρμογή του cBox ο οποίος διατήρεί τα μηνύματα έως ώτου τα αποστείλει στον προορισμό τους. Ωστόσο είναι δυνατή τη ενσωμάτωση ενός ολοκληρωμένου Proxy Server όπως για παράδειγμα το Squid, το οποίο θα παρέχει στον χρήστη νέες δυνατότητες όπως Web Caching και Anonymity.

5.2. Πηγαίος Κώδικας

Ο πηγαίος κώδικας που έχει αναπτυχθεί στα πλαίσια του έργου cBox βρίσκεται αναρτημένος στον ιστότοπο Github. Ο σύνδεσμος στον οποίο μπορεί να βρεθεί ο πηγαίος κώδικας είναι:

https://github.com/CEID-DS/cbox/demo

5.3. Εργαλεία

Για την ανάπτυξη του λογισμικού και για την συγγραφή της διπλωματικής εργασίας χρησιμοποιήθηκαν τα ακόλουθα εργαλεία:

- IDE Eclipse:
  - JAVA Developement Kit
  - JAVA Runtime Environment
  - CDT (C/C++ Development Tooling)
  - Android Software Developement Kit
  - Android Native Development Kit
  - Protobuf
  - ObjectAid UML
- ArgoUML
- StarUML
- OpenOffice Writer
- OpenOffice Draw

5.4. Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε την περίοδο Νοέμβριος 2011 – Ιούνιος 2012 στα πλαίσια του Προπτυχιακού προγράμματος σπουδών του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών. Μία πρώιμη έκδοση του συστήματος cBox, με υποσύνολο των λειτουργιών, μελετήθηκε στα πλαίσια του μαθήματος Κατανεμημένα Συστήματα 2 του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών. Πριν από την παρουσίαση αυτής της διπλωματικής εργασίας θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν στην πραγματοποίηση της και ιδιαίτερα:

Τον Καθηγητή κ. Παύλο Σπυράκη για την ανάθεση αυτού του τόσο ενδιαφέροντος θέματος και για την ευκαιρία που μου δόθηκε να κάνω πράξη τα ερευνητικά μου ενδιαφέροντα. Τον κ. Ιωάννη Χατζηγιαννάκη για την καθοδήγηση του καθ' όλη τη διάρκεια της έρευνας - υλοποίησης του λογισμικού που εμπεριέχεται στη διπλωματική εργασία και για τις υποδείξεις που μου παρείχε καθώς και για την κριτική που πραγματοποίησε στο κείμενο της εργασίας.

Τον κ. Μιχάλη Μουντράκη CFO της εταιρείας UIT Athens για τις πολύτιμες και καθοριστικές συμβουλές, για την ψυχολογική στήριξη, για την άμεση ανταπόκριση του σε κάθε μου ερώτημα κατά την διάρκεια της εκπόνησης της εργασίας. Τον κ. Σταύρο Βαϊτση CEO της εταιρείας UIT Athens για τις διαφωτιστικότατες συζητήσεις που πραγματοποιήσαμε.

Τους φίλους μου που με βοήθησαν με ανταλλαγές απόψεων, το ενδιαφέρον τους και για τη σημαντική τους βοήθεια σε όλα τα στάδια της εργασίας. Ειδικότερα θα ήθελα να ευχαριστήσω τους Γεώργιο Σαχπατζίδη, Βάιο Τσιτσώνη, Γεώργιο Σμέρο.

Την οικογένεια μου και ειδικότερα τους γονείς μου και την αδελφή μου για τη συνεχή και άοκνη υποστήριξή τους σε κάθε βήμα της ζωής μου. Απλά σας ευχαριστώ.

6. Βιβλιογραφία

1. The FreedomBox Foundation, http://freedomboxfoundation.org/

2. BragleBoard [2011] - 'Low cost, high performance, low power ompa3 based platform'. http://beagleboard.org/

3. GlobalScale [2011] - 'Developer kit for the marvell pxa510 highly integrated soc', http://globalscaletechnologies.com/

4. Android – Linux-based operating system for mobile devices such as smartphones and tablet computers, http://www.android.com/

5. Twitter API – Create applications that integrate Twitter, https://dev.twitter.com

6. OAuth – An open protocol to allow secure API authorization in a simple and standard method from desktop and web applications, http://oauth.net/ - http://hueniverse.com/oauth

7. Google Protocol Buffers – Language-neutral, platform-neutral, extensible mechanism for serializing structured data, http://code.google.com/p/protobuf

8. Wiselib – A generic algorithms library for heterogeneous, distributed, embedded systems, http://github.com/ibr-alg/wiselib

9. Java Native Interface - Programmer's Guide and Specification, http://java.sun.com/docs/books/jni/

10. Socket Programming - Beej's Guide to Networking Programming Using Internet Sockets, http://beej.us/guide/bgnet/

11. BlueCove - Java Libray for Bluetooth (JSR - 82 implementation), http://bluecove.org/

12. Base64Coder – An open-source Base64 encoder/decoder in Java, http://www.source-code.biz/base64coder/java/