

Ατομική Διπλωματική Εργασία

**ΕΡΓΑΛΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΣΤΗΜΑΤΩΝ ΕΞΥΠΝΩΝ
ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ**

Στέφανος Τάραντο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΛΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΣΤΗΜΑΤΩΝ ΕΞΥΠΝΩΝ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ

Στέφανος Τάραντο

Επιβλέπων Καθηγητής

Δημήτρης Ζεϊναλιπούρ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2014

Ευχαριστίες

Αρχικά, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή μου κ. Δημήτρη Ζεϊναλιπούρ για όλη τη βοήθεια και την καθοδήγηση που μου παρείχε κατά τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Θα ήθελα επίσης να τον ευχαριστήσω για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα καινοτόμο θέμα, καθώς και νέες τεχνολογίες οι οποίες θα βοηθήσουν στην μετέπειτα πορεία μου.

Επιπλέον θα ήθελα να ευχαριστήσω τον μεταπτυχιακό φοιτητή του τμήματος πληροφορικής Γιώργο Λάρκου για την βοήθεια και την στήριξη που μου παρείχε κατά την διάρκεια της διπλωματικής μου εργασίας, καθώς και όλους όσους ήταν στο πλευρό μου κατά την διάρκεια των σπουδών μου.

Αναγνώριση

Επίδειξη του συστήματος SmartLab και των επεκτάσεων του, που περιγράφονται σε αυτήν την διπλωματική εργασία έχουν παρουσιαστεί στο ακόλουθο συνέδριο:

- "Demonstration Abstract: Sensor Mockup Experiments with SmartLab", Georgios Larkou, Marios Mintzis, **Stefano Taranto**, Andreas Konstantinidis, Panayiotis G. Andreou, Demetrios Zeinalipour-Yazti "Proceedings of the 13th international conference on Information processing in sensor networks" (IPSN '14), Berlin, Germany IEEE Press, Pages: 339--340, Berlin, Germany, April 15-17, ISBN: 978-1-4799-3146-0, 2014

Περίληψη

Λόγο της αύξησης του αριθμού των έξυπνων κινητών συσκευών τα τελευταία χρόνια παρουσιάστηκαν νέες ανάγκες, όπως για παράδειγμα ο έλεγχος πραγματικών συσκευών καθώς και η εύκολη και γρήγορη διαχείριση τους. Η παρούσα διπλωματική εργασία εκπονήθηκε με σκοπό την επέκταση του συστήματος SmartLab, μέσω εργαλείων και υποσυστημάτων που θα επιτρέψουν στο σύστημα να είναι πιο επεκτάσιμο, καθώς και την πρόσθεση νέων λειτουργιών. Επιπλέον ο σχεδιασμός και η υλοποίηση μιας νέας αρχιτεκτονικής, με βάση TCP υποδοχές, που ονομάζεται Socket Programming Bridge, η οποία απαλλάσσει το σύστημα από περιορισμούς που υπήρχαν, λόγω του εργαλείου ADB, επιτρέποντας έτσι την ανάπτυξη του συστήματος σε μια αστική κλίμακα. Ένας άλλος στόχος της εργασίας που έγινε είναι η λειτουργία του συστήματος με περισσότερη σταθερότητα αντικαθιστώντας το εργαλείο ADB με το νέο πρωτόκολλο που υλοποιήθηκε. Μέσω της εργαλειοθήκης που έχει υλοποιηθεί στο σύστημα, σε Django/Python, παρέχονται εργαλεία μέσω των οποίων ο διαχειριστής του συστήματος μπορεί να συντηρεί τις συσκευές απομακρυσμένα με μια απλή και εύκολη διεπαφή. Μια άλλη συνεισφορά της εργασίας που έχει γίνει, είναι το νέο υποσύστημα που επεκτείνει το εργαλείο AppInventor, μέσω του οποίου οι χρήστες του συστήματος μπορούν να αναπτύξουν εφαρμογές για συσκευές Android εύκολα, μέσω ενός γραφικού περιβάλλοντος, και να τις εκτελούν απευθείας στις συσκευές που έχουν κρατημένες στο σύστημα. Μια άλλη λειτουργία που επιτρέπεται από το σύστημα είναι η εκτέλεση εντολών κελύφους στις συσκευές, μέσω του υποσυστήματος Remote Shells που έχει υλοποιηθεί στην εργασία αυτή.

Για την αξιολόγηση των νέων υποσυστημάτων που έχουν υλοποιηθεί, έχουν γίνει πειράματα, για κάθε υποσύστημα ξεχωριστά για τις δύο αρχιτεκτονικές, επιτρέποντας έτσι την σύγκριση της αρχιτεκτονικής ADB, με την αρχιτεκτονική SPB. Τα πειράματα έχουν γίνει βάση του χρόνου απόκρισης κάθε υποσυστήματος, καθώς και την χρησιμοποίηση του επεξεργαστή. Τα πειραματικά αποτελέσματα έδειξαν πως η νέα αρχιτεκτονική έχει καλύτερους χρόνους απόκρισης καθώς και καλύτερη χρησιμοποίηση του επεξεργαστή.

Περιεχόμενα

Ευχαριστίες	i
Αναγνώριση	ii
Περίληψη	iii
Περιεχόμενα	iv
Κεφάλαιο 1: Εισαγωγή	1
1.1 Εισαγωγή	1
1.2 Παρακίνηση και συνεισφορά	2
1.3 Περίγραμμα εργασίας	5
Κεφάλαιο 2: Σχετική Εργασία	6
2.1 Εισαγωγή	6
2.2 Λύσεις απομακρυσμένης παρακολούθησης	7
2.3 Κλίνες δοκιμών ασύρματων δικτύων αισθητήρων	7
2.4 Κλίνες δοκιμών έξυπνων συσκευών	8
2.5 Ανθρωποκεντρικές κλίνες δοκιμών	10
Κεφάλαιο 3: Αρχιτεκτονική SmartLab	12
3.1 Εισαγωγή	12
3.2 Διεπαφή χρήστη και επίπεδο δεδομένων	12
3.3 Επίπεδο εξυπηρετητή συσκευών	15
3.4 Επίπεδο υλικού	17
3.5 Αρχές και μεθοδολογία σχεδιασμού	19
Κεφάλαιο 4: Υλοποίηση Πρωτοκόλλου SPB	20
4.1 Εισαγωγή	20
4.2 Εργαλείο ADB και οι περιορισμοί του	21
4.3 Ανασκόπηση του SPB πρωτοκόλλου	23
4.4 Υλοποίηση εξυπηρετητή SPB	25
4.5 Υλοποίηση πελάτη SPB	32
Κεφάλαιο 5: Εργαλεία Διαχείρισης SmartLab	36
5.1 Εισαγωγή	36

5.2	Ανασκόπηση και περιγραφή εργαλείων διαχείρισης	36
5.3	Υλοποίηση	40
Κεφάλαιο 6: Γραφικό Περιβάλλον Προγραμματισμού		43
6.1	Εισαγωγή	43
6.2	Ανασκόπηση Εργαλείου AppInventor	43
6.3	Επέκταση Εργαλείου AppInventor	45
Κεφάλαιο 7: Πειραματικά Αποτελέσματα		48
7.1	Εισαγωγή	48
7.2	Πειραματική μεθοδολογία	49
7.3	Πειραματικά αποτελέσματα για χρόνο απόκρισης	49
7.4	Πειραματικά αποτελέσματα για χρήση επεξεργαστή	53
Κεφάλαιο 8: Συμπεράσματα και μελλοντικές επεκτάσεις		56
8.1	Συμπεράσματα	56
8.2	Μελλοντικές επεκτάσεις	57
Βιβλιογραφία		58
Παράρτημα Α.		A-1

Κεφάλαιο 1

Εισαγωγή

1.1	Εισαγωγή	1
1.2	Παρακίνηση και συνεισφορά	2
1.3	Περίγραμμα εργασίας	5

1.1 Εισαγωγή

Το 2012 ήταν η αρχή μιας εποχής, στην οποία ο αριθμός των έξυπνων κινητών συσκευών ξεπέρασε για πρώτη φορά στη ιστορία τον αριθμό όλων των τύπων προσωπικών υπολογιστών μαζί. Το λειτουργικό Android, σήμερα κατέχει ένα μεγάλο ποσοστό της αγοράς σε λειτουργικά έξυπνων συσκευών, με περισσότερες από 1,000,000 εφαρμογές. Ο επαναπρογραμματισμός έξυπνων συσκευών καθώς και η χρήση τους για έλεγχο εφαρμογών και συλλογή δεδομένων σε μεγάλη κλίμακα, είναι μια χρονοβόρα και κουραστική διαδικασία η οποία έχει σημαντικές υλικοτεχνικές προκλήσεις. Το σύστημα SmartLab [1], μια ολοκληρωμένη αρχιτεκτονική για διαχείριση ομάδων έξυπνων συσκευών Android, προσφέρει με εύκολο και γρήγορο τρόπο διαχείριση ομάδων έξυπνων συσκευών.

Το σύστημα προσφέρει έλεγχο σε πραγματικές συσκευές πάνω στις συσκευές, για παράδειγμα στο λειτουργικό σύστημα, στην δικτύωση, στην αποθηκευτική μονάδα [2], καθώς επίσης και σε σενάρια τα οποία απαιτούν αισθητήρες και γεωγραφική θέση [3],[4]. Το σύστημα SmartLab έχει χρησιμοποιηθεί σε διάφορες εργασίες και πειράματα, λόγω της εύκολης και γρήγορης διαχείρισης πολλών συσκευών. Το σύστημα SmartLab αποτελείται από περισσότερες από 40 έξυπνες συσκευές Android, η οποίες είναι συνδεδεμένες στο

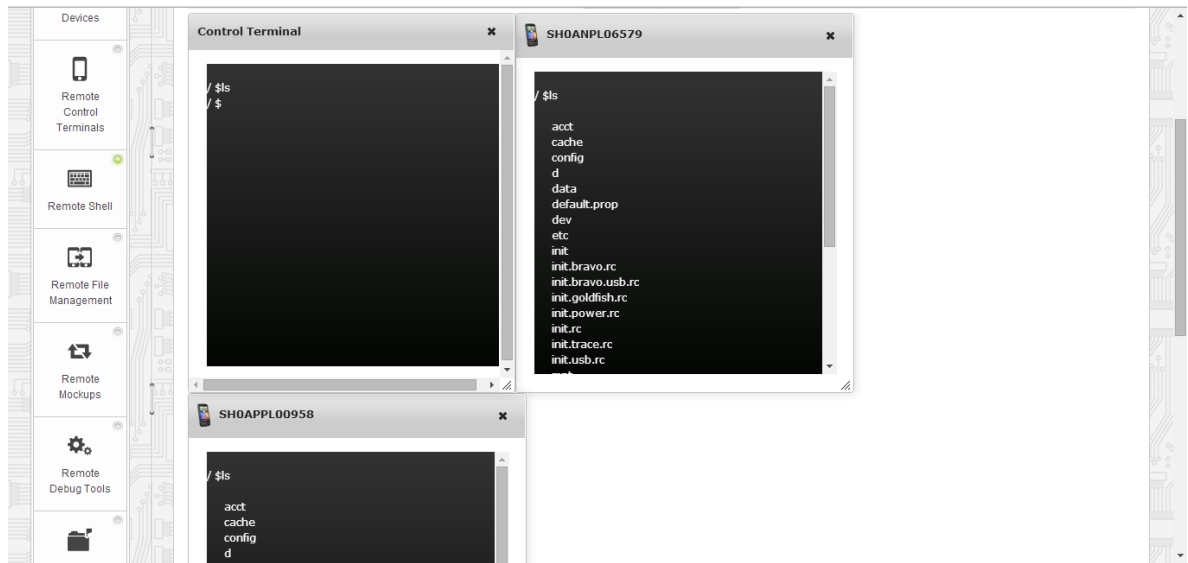


Σχήμα 1.1 : Υποσύστημα συσκευών συνδεδεμένες τοπικά στο νέφος του SmartLab

σύστημα είτε εικονικά, είτε ενσύρματα ή ασύρματα σε ιδιωτικό νέφος (Σχήμα 1.1). Μέσω μιας εύκολης δικτυακής διεπαφής, οι χρήστες του συστήματος μπορούν να φορτώσουν και να εγκαταστήσουν εφαρμογές σε πολλές συσκευές παράλληλα, να παρακολουθούν καθώς και να αλληλοεπιδρούν με τις οθόνες των συσκευών, να μεταφέρουν αρχεία, να εκτελέσουν εντολές κελύφους, καθώς και να δίνουν στις συσκευές GPS ή sensor mockups.

1.2 Παρακίνηση και συνεισφορά

Σε αυτή την διπλωματική εργασία, σκοπός ήταν η επέκταση του συστήματος SmartLab έτσι ώστε να επεκταθούν ορισμένα από τα ήδη υπάρχον υποσυστήματα, να υλοποιηθούν νέα υποσυστήματα, καθώς και να μετατραπούν έτσι ώστε το σύστημα να μπορεί να



Σχήμα 1.2: Remote Shells subsystem (User Interface)

επεκταθεί σε μεγαλύτερη κλίμακα. Η συνεισφορά της εργασίας αυτής χωρίζεται σε 4 βασικές ενότητες:

- **Υλοποίηση Remote shells (ADB):** Λόγω του ότι το υποσύστημα Remote Shells που προϋπήρχε, δεν υποστήριζε την φύλαξη κατάστασης σε μια περίοδο λειτουργίας, καθώς και η διεπαφή του υποσυστήματος δεν ήταν διαισθητική, έχει υλοποιηθεί ένα νέο υποσύστημα Remote shells. Στο υποσύστημα αυτό υποστηρίζεται η φύλαξη κατάστασης μεταξύ διάφορων εντολών, που επιτρέπει την πλοήγηση σε καταλόγους και την πιο εύκολη χρήση του, αφού προσομοιώνει ένα πραγματικό κέλυφος. Επίσης η διεπαφή του κελύφους έχει σχεδιαστεί, έτσι ώστε να μοιάζει με πραγματικό κέλυφος bash, αφού υποστηρίζεται και η πλοήγηση των εντολών που εκτελέστηκαν στο ιστορικό του χρήστη. Επιπλέον το υποσύστημα δεν χρησιμοποιεί πλέον AJAX αιτήσεις, αλλά Websockets.
- **Επέκταση εργαλείου AppInventor:** Το σύστημα SmartLab, εκτός από τις λειτουργίες που προσφέρει, οι οποίες αναφέρθηκαν πιο πάνω, έχει την δυνατότητα

να προσφέρει τις συσκευές του και για εκπαιδευτικούς λόγους. Το σύστημα δηλαδή μπορεί να προσφέρει στους χρήστες που ενδιαφέρονται στην ανάπτυξη εφαρμογών, οι οποίοι δεν έχουν συσκευές Android, την ευκαιρία να αναπτύξουν εφαρμογές, χρησιμοποιώντας τις συσκευές του συστήματος. Για τον λόγο αυτό, έχει γίνει η επέκταση του εργαλείου AppInventor. Μέσω του εργαλείου αυτού, οι χρήστες μπορούν εύκολα, μέσω του γραφικού περιβάλλοντος που παρέχει το εργαλείο, να υλοποιούν εφαρμογές για έξυπνες συσκευές Android. Η επέκταση του εργαλείου, έχει ως αποτέλεσμα την πρόσθεση μιας νέας λειτουργίας, έτσι ώστε μετά την υλοποίηση της εφαρμογής, οι χρήστες με το πάτημα ενός κουμπιού, να μπορούν να εκτελέσουν την εφαρμογή τους στις συσκευές που έχουν κρατημένες στο σύστημα.

- **Υλοποίηση εργαλείων διαχείρισης:** Λόγω του μεγάλου αριθμού συσκευών που παρέχει το σύστημα στους χρήστες, θα πρέπει ο διαχειριστής των συσκευών, να μπορεί εύκολα και γρήγορα να συντηρεί τις συσκευές. Λόγω του ότι η διαδικασία αυτή απαιτεί από τον διαχειριστή να βρίσκεται τοπικά στον εξυπηρετητή που είναι συνδεδεμένες οι συσκευές, καθώς και να κάνει τις λειτουργίες συντήρησης στις συσκευές μια προς μια, υπάρχει η ανάγκη δημιουργίας μιας εργαλειοθήκης η οποία ξεπερνά τους περιορισμούς αυτούς. Η εργαλειοθήκη αυτή προσφέρει απομακρυσμένη συντήρηση των συσκευών, καθώς και παράλληλη συντήρηση πολλών συσκευών.
- **Σχεδιασμός και υλοποίηση αρχιτεκτονικής SPB:** Η μέχρι τώρα αρχιτεκτονική του συστήματος χρησιμοποιούσε το εργαλείο ADB, ώστε να μπορεί ο εξυπηρετητής συσκευών να επικοινωνεί με τις συσκευές. Το εργαλείο ADB, περιορίζει το σύστημα στις λειτουργίες που παρέχει, καθώς και την επεκτασιμότητα του. Για τον λόγο αυτό υπάρχει η ανάγκη σχεδιασμού και υλοποίησης μιας νέας αρχιτεκτονικής, η οποία να μπορεί να προσομοιώνει τις λειτουργίες που παρέχει το εργαλείο ADB, χωρίς όμως να περιορίζει το σύστημα. Η αρχιτεκτονική που έχει σχεδιαστεί και υλοποιηθεί στην εργασία αυτή, ονομάζεται Socket Programming

Bridge (SPB), η οποία για την επικοινωνία μεταξύ εξυπηρετητή-συσκευών, χρησιμοποιεί TCP υποδοχές, προσομοιώνοντας όμως εξ ολοκλήρου της λειτουργίες του εργαλείου ADB. Με την αρχιτεκτονική SPB, το σύστημα δεν περιορίζεται, έτσι μπορεί να επεκταθεί και σε εξωτερικούς χώρους, αφού οι συσκευές συνδέονται είτε με WIFI είτε με 3G.

1.3 Περίγραμμα εργασίας

Στο πρώτο κεφάλαιο παρουσιάζονται ο σκοπός της εν λόγω διπλωματικής εργασίας, η παρακίνηση, καθώς και οι συνεισφορά που έχει γίνει. Στο δεύτερο κεφάλαιο θα παρουσιαστεί η σχετική δουλειά που υπάρχει καθώς και το βασικό υπόβαθρό της εργασίας αυτής. Στο τρίτο κεφάλαιο θα παρουσιαστούν συνοπτικές πληροφορίες για τα υποσυστήματα του συστήματος SmartLab, καθώς και βασικά στοιχεία των δύο αρχιτεκτονικών (SPB και ADB), για όλα τα επίπεδα λειτουργίας τους, δηλαδή από την διεπαφή χρήστη, μέχρι και το επίπεδο των συσκευών. Στο τέταρτο κεφάλαιο θα παρουσιαστεί εκτενώς το εργαλείο ADB και τους περιορισμούς του συστήματος με βάση την αρχιτεκτονική ADB. Στη συνέχεια θα γίνει μια ανασκόπηση του πρωτοκόλλου SPB, και θα επεξηγηθεί η υλοποίηση του, από πλευράς εξυπηρετητή αλλά και από πλευράς πελάτη Android. Στο πέμπτο κεφάλαιο θα παρουσιαστεί η εργασία που έγινε για την υλοποίηση της επέκτασης του εργαλείου AppInventor και την ενσωμάτωση του στο σύστημα SmartLab. Στο κεφάλαιο έξι παρουσιάζεται η εργαλειοθήκη συντήρησης του συστήματος, καθώς και τα πλεονεκτήματα που παρέχει στον διαχειριστή συσκευών του συστήματος. Στο έβδομο κεφάλαιο θα παρουσιαστεί η πειραματική μελέτη που έγινε, για την σύγκριση των δύο αρχιτεκτονικών (SPB και ADB), για κάθε υποσύστημα ξεχωριστά, με βάση δύο κριτήρια, τον χρόνο απόκρισης κάθε υποσυστήματος, καθώς και την χρησιμοποίηση του επεξεργαστή της συσκευής. Τέλος στο όγδοο κεφάλαιο θα παρουσιαστούν τα τελικά συμπεράσματα και τις επιπλέον λειτουργίες που μπορούν να γίνουν.

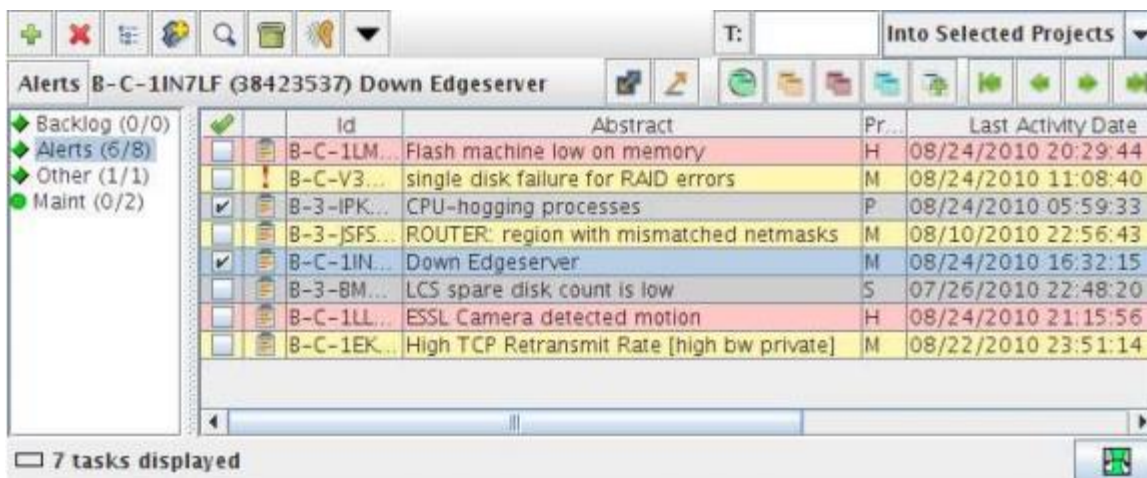
Κεφάλαιο 2

Σχετική Εργασία

2.1	Εισαγωγή	6
2.2	Λύσεις απομακρυσμένης παρακολούθησης	7
2.3	Κλίνες δοκιμών ασύρματων δικτύων αισθητήρων	7
2.4	Κλίνες δοκιμών έξυπνων συσκευών	8
2.5	Ανθρωποκεντρικές κλίνες δοκιμών	10

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστεί μια σύντομη επισκόπηση της σχετικής εργασίας. Το σύστημα SmartLab έχει εμπνευστεί από τα συστήματα PlanetLab [5] και Emulab [6], τα οποία πρωτοπόρησαν σε παγκόσμια ερευνητικά δίκτυα. Επίσης από το σύστημα MoteLab [7], το οποίο πρωτοπόρησε σε ερευνητικά δίκτυα αισθητήρων καθώς και το Amazon Elastic Compute Cloud (EC2). Καμία από τις παραπάνω εργασίες, δεν έχει επικεντρωθεί σε έξυπνες συσκευές, επομένως οι κλίνες δοκιμών των πιο πάνω εργασιών έχουν εντελώς διαφορετικές αρχιτεκτονικές και επιθυμίες. Στα πιο κάτω υποκεφάλαια, θα γίνει μια ανασκόπηση των κλινών δοκιμών που σχετίζονται με το SmartLab.



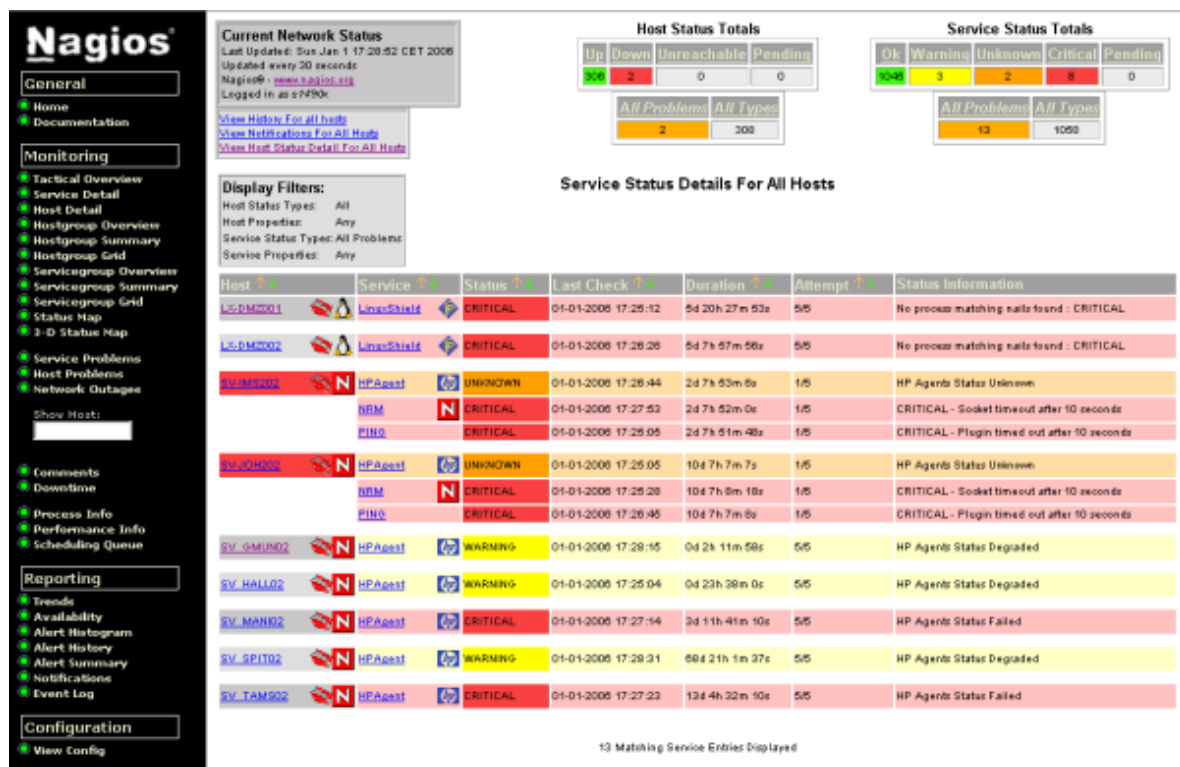
Σχήμα 2.1: Akamai Query System

2.2 Λύσεις απομακρυσμένης παρακολούθησης

Υπάρχουν επί του παρόντος αρκετές λύσεις για απομακρυσμένη παρακολούθηση. Μια από αυτές είναι το σύστημα Nagios (Σχήμα 2.2), ένα ανοικτού κώδικα σύστημα απομακρυσμένης παρακολούθησης, το Akamai Query System [8] (Σχήμα 2.1), STORM [9] και RedAlert [10]. Τα πιο πάνω συστήματα τείνουν κυρίως να παρέχουν λύσεις για εφαρμογές διαδικτύου και για εξυπηρετητές. Επιπλέον, κανένα από τα παραπάνω συστήματα δεν παρέχει εργαλεία που σχετίζονται με την ρύθμιση και την διαχείριση ομάδων από έξυπνες συσκευές. Το σύστημα SmartLab έχει ως στόχο να παρέχει απομακρυσμένη παρακολούθηση ειδικά για νέφος έξυπνων συσκευών.

2.3 Κλίνες δοκιμών ασύρματων δικτύων αισθητήρων

Το σύστημα MoteLab [7] (Σχήμα 2.3) είναι ένα διαδικτυακό σύστημα αισθητήρων το οποίο έχει αναπτυχθεί από το Πανεπιστήμιο Harvard, που έχει πρωτοπορήσει στην έρευνα δικτύων αισθητήρων. Το σύστημα CitySense [11] το οποίο είναι διάδοχο του συστήματος



Σχήμα 2.2: Nagios System

MoteLab, επιτρέπει σε αστική κλίμακα την ανάπτυξη δικτύων αισθητήρων. Το Mobiscrope [12] είναι μια ομοσπονδία κατανεμημένων κινητών αισθητήρων, σε ένα σύστημα ανίχνευσης που πετυχαίνει μεγάλης πυκνότητας κάλυψη σε ευρεία περιοχή, μέσω κινητικότητας. Η εταιρία Microsoft έχει κάνει αρκετές προσπάθειες στο να κτίσει δίκτυα αισθητήρων με κινητές συσκευές [13], όμως καμία από τις προσπάθειες αυτές δεν έχει επικεντρωθεί σε έξυπνες συσκευές, ούτε σε χαρακτηριστικά τους, όπως για παράδειγμα καταγραφή οθόνης και διαδραστικότητα.

2.4 Κλίνες δοκιμών έξυπνων συσκευών

Υπάρχουν επί του παρόντος αρκετές εμπορικές πλατφόρμες που επιτρέπουν απομακρυσμένη πρόσβαση σε πραγματικές έξυπνες συσκευές, συμπεριλαμβανομένων των

Mote ID	Room (click for map)	Network jack	IP Addr	Program Port	Firmware Version	Status	Status last updated
1	MD219	2058-2	192.168.1.101	10001	gwa-v2	OK	2 Jan 2014 18:15:14
2	MD219	2058-2	192.168.1.101	10002	gwa-v2	OK	2 Jan 2014 18:15:14
3	MD110	1069-1	192.168.1.102	10001	gwa-v2	OK	2 Jan 2014 18:15:14
4	MD110	1069-1	192.168.1.102	10002	gwa-v2	OK	2 Jan 2014 18:15:14
5	MD111	1060-1	192.168.1.103	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
6	MD111	1060-1	192.168.1.103	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
7	MD113	1005-1	192.168.1.104	10001	N/A	REMOVED	22 Jul 2009 18:17:00
8	MD113	1005-1	192.168.1.104	10002	N/A	REMOVED	22 Jul 2009 18:17:00
9	MD113	1010-1	192.168.1.105	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
10	MD113	1010-1	192.168.1.105	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
11	MD114	1262-1	192.168.1.106	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
12	MD114	1262-1	192.168.1.106	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
13	MD115	1017-1	192.168.1.107	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
14	MD115	1017-1	192.168.1.107	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
15	MD116	1276D1-1	192.168.1.108	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
16	MD116	1276D1-1	192.168.1.108	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
17	MD117	1033-1	192.168.1.109	10001	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
18	MD117	1033-1	192.168.1.109	10002	gwa-v2	DISABLED (cannot ping)	2 Jan 2014 18:15:16
19	MD117	1026-1	192.168.1.110	10001	gwa-v2	OK	2 Jan 2014 18:15:14

Σχήμα 2.3: MoteLab System

συστημάτων Samsung Remote Test Lab [14], PerfectoMobile [15] (Σχήμα 2.4), Device Anyware [16] και AT&T ARO. Οι πλατφόρμες αυτές διαφέρουν από το σύστημα SmartLab στους πιο κάτω τρόπους:

- Επικεντρώνονται σε δοκιμές εφαρμογών σε διαφορετικές συσκευές
- Δεν είναι ανοικτά, επομένως δεν προσφέρουν τρόπους για το πώς να υλοποιηθούν και να εκτελεστούν αποτελεσματικά εφαρμογές για έξυπνες συσκευές σε μεγάλη κλίμακα
- Δεν υποστηρίζουν ευρύ φάσμα από λειτουργίες που προσφέρονται από το σύστημα SmartLab, όπως για παράδειγμα αισθητήρες, mockups και αυτοματοποίηση.

Τα εργαστήρια Sandia National Laboratories έχουν αναπτύξει και ξεκινήσει το σύστημα MegaDroid [17], ένα σύστημα 520 κόμβων αξίας \$500K που χρησιμοποιεί 300,000 εικονικές συσκευές Android. Η βασικός σκοπός του συστήματος MegaDroid, είναι να επιτρέψει σε ερευνητές να προσομιώσουν πραγματικούς χρήστες. Το σύστημα αυτό επικεντρώνεται μόνο σε εικονικές συσκευές, αντίθετα με το σύστημα SmartLab, το οποίο επικεντρώνεται και σε εικονικές αλλά και σε πραγματικές συσκευές.



Σχήμα 2.4: PerfectoMobile System

2.5 Ανθρωποκεντρικές κλίνες δοκιμών

Υπάρχει ακόμη μια μεγάλη κατηγορία συστημάτων η οποία επικεντρώνεται σε ευκαιριακές και συμμετοχικές κλίνες δοκιμών σε έξυπνες συσκευές, με πραγματικούς χρήστες, για παράδειγμα τα συστήματα PRISM [18], CrowdLab [19] και PhoneLab [20, 21], τα συστήματα αυτά όμως είναι συνήθως συμπληρωματικά και έχουν διαφορετικά ζητούμενα από αυτά του συστήματος SmartLab. Το σύστημα PhoneLab, το οποίο είναι μια συμμετοχική πλατφόρμα δοκιμών, που αποτελείται από φοιτητές και καθηγητές στο University of Buffalo, δεν επιτρέπει σε προγραμματιστές εφαρμογών να έχουν πρόσβαση στις οθόνες των συσκευών, να μεταφέρουν αρχεία ούτε και την αποσφαλμάτωση εφαρμογών, αλλά επιτρέπει την εκκίνηση εργασιών καταγραφής δεδομένων με τρόπο εκτός σύνδεσης. Το PhoneLab στοχεύει στην συλλογή δεδομένων αντιθέτως με τα σενάρια τα οποία προσφέρει το σύστημα SmartLab, όπως για παράδειγμα αποσφαλμάτωση και εγκατάσταση. Επιπλέον το σύστημα PhoneLab, είναι πιο περιοριστικό αφού εργασίες που υποβάλλονται πρέπει να εγκριθούν μέσα από μια διαδικασία, διότι οι εφαρμογές που αναπτύσσονται, εκτελούνται πάνω σε πραγματικές συσκευές, πραγματικών ανθρώπων.

Τέλος, Carat Project του UC Berkeley, επιτρέπει διάγνωση συλλογικής ενέργειας και συστάσεις για την καλύτερη διαχείριση της ενέργειας έξυπνων συσκευών από περισσότερες από μισό εκατομμύριο συσκευές. Το σύστημα SmartLab είναι συμπληρωματικό στις πιο πάνω έρευνες αφού προσφέρει ιδέες και αναφορές για ποικίλα υποσυστήματα που μπορούν να εκμεταλλευθούν τα συστήματα αυτά.

Κεφάλαιο 3

Αρχιτεκτονική SmartLab

3.1	Εισαγωγή	12
3.2	Διεπαφή χρήστη και επίπεδο δεδομένων	12
3.3	Επίπεδο εξυπηρετητή συσκευών	15
3.4	Επίπεδο υλικού	17
3.5	Αρχές και μεθοδολογία σχεδιασμού	19

3.1 Εισαγωγή

Στο κεφάλαιο αυτό, αρχικά θα δούμε την αρχιτεκτονική του συστήματος με SPB και πιο συγκεκριμένα την διεπαφή χρήστη, το επίπεδο δεδομένων, το επίπεδο του εξυπηρετητή των συσκευών και τέλος το επίπεδο του υλικού, δηλαδή των συσκευών. Κατά την διάρκεια αυτού του κεφαλαίου θα βλέπουμε της μετατροπές από την αρχιτεκτονική με βάση ADB σε SPB.

3.2 Διεπαφή χρήστη και επίπεδο δεδομένων

Το σύστημα SmartLab, υλοποιεί διάφορα υποσυστήματα για την επικοινωνία χρήστη-συσκευών χρησιμοποιώντας είτε συνδέσεις με βάση Websockets είτε κλήσεις AJAX. Οι πιο κάτω λειτουργίες ήταν βασισμένες στο εργαλείο ADB για την λειτουργία τους, ενώ με την νέα αρχιτεκτονική χρησιμοποιείται το πρωτόκολλο SPB(Socket Programming Bridge). Πιο συγκεκριμένα το σύστημα SmartLab υποστηρίζει τα εξής:

- Remote Control Terminals (RCT), ένα υποσύστημα με βάση Websockets, το οποίο προσφέρει απομακρυσμένες οθόνες των συσκευών καθώς επίσης μιμείται κλικ και χειρονομίες στην οθόνη των συσκευών.
- Remote File Management (RFM), ένα υποσύστημα με βάση AJAX κλήσεις, που επιτρέπει στους χρήστες να στέλνουν ή να παραλαμβάνουν αρχείο από και προς τις συσκευές.
- Remote Shells (RS), ένα υποσύστημα με βάση Websockets, το οποίο επιτρέπει ένα ευρύ φάσμα εντολών UNIX οι οποίες εκτελούνται στις συσκευές σε λειτουργικό Android.
- Remote Scripting Environment (RSE), ένα υποσύστημα με βάση Websockets, το οποίο επιτρέπει την καταγραφή κλικ του χρήστη και την εκτέλεση τους για επαναλαμβανόμενες διαδικασίες.
- Remote Debug Tools (RDT), ένα υποσύστημα με βάση Websockets, το οποίο προσφέρει στο χρήστη πληροφορίες αποσφαλμάτωσης.
- Remote Mockups (RM), ένα υποσύστημα με βάση Websockets, το οποίο δίνει δεδομένα GPS ή αισθητήρων στις συσκευές σε XML αρχεία, σε περίπτωση που χρειάζεται να γίνουν πειράματα με τέτοιες μετρήσεις.

Websockets

Προκειμένου να επιτύχουμε μια γρήγορη και αξιόπιστη σύνδεση μεταξύ της διεπαφής χρήστη και του εξυπηρετητή, το σύστημα SmartLab χρησιμοποιεί HTML5 / Websockets, έτσι ώστε να υπάρχει μια πλήρως αμφίδρομη επικοινωνία πάνω από μια TCP υποδοχή από τον περιηγητή. Παραδοσιακά οι περιηγητές ιστού δεν υποστήριζαν TCP υποδοχές, για θέματα ασφαλείας και οι αιτήσεις γίνονταν μέσω HTTP αιτήσεων. Τα Websockets παρέχουν υψηλού ρυθμού HTTP αλληλεπιδράσεις, οι οποίες είναι απαραίτητες σε κάποια από τα υποσυστήματα του SmartLab. Τα Websockets αποτελούνται από δύο βασικά μέρη:

- HTTP χειραψία, κατά την οποία κάποια κλειδιά πρωτοκόλλου σε επίπεδο εφαρμογής ανταλλάσσονται.

- Φάση μεταφοράς δεδομένων, κατά την οποία δεδομένα αποστέλλονται και παραλαμβάνονται σε πλήρως αμφίδρομο τρόπο.

Προκειμένου να υποστηρίζονται τα Websockets, στους περισσότερους περιηγητές, χρησιμοποιείται το ανοιχτό λογισμικό kanaka websocketify plugin. Το συγκεκριμένο λογισμικό αναλαμβάνει την αρχική χειραγώγηση, καθώς επίσης λειτουργεί σε SWF εάν ο περιηγητής δεν υποστηρίζει HTML5, έτσι υποστηρίζοντας όλους τους περιηγητές ιστού.

Σύστημα αρχείων

Το σύστημα SmartLab χρησιμοποιεί ext4 τοπικό σύστημα αρχείων στον εξυπηρετητή ιστού. Μετά την εγγραφή ενός χρήστη δημιουργείται ένας κατάλογος για τον συγκεκριμένο χρήστη με καθορισμένο μέγεθος. Το σύστημα αρχείων είναι τοποθετημένο με SSHFS, σε όλους τους εξυπηρετητές. Με τον τρόπο αυτό υπάρχει ένας ενιαίος κατάλογος για κάθε χρήστη. Αναφορικά με την συνδεσιμότητα μεταξύ των συστημάτων αρχείων των συσκευών, υπάρχουν δύο επιλογές:

- Τοποθέτηση του καταλόγου των χρηστών στις συσκευές μέσω SSHFS
- Αντιγραφή των δεδομένων από/προς τις συσκευές μέσω ADB.

Η πρώτη επιλογή είναι καλύτερη ως προς θέματα απόδοσης, όμως είναι διαθέσιμη μόνο σε εκδόσεις Android 4.0 ICS, ενώ η δεύτερη επιλογή είναι πιο καθολική προσέγγιση.

SSHFS & MySQL

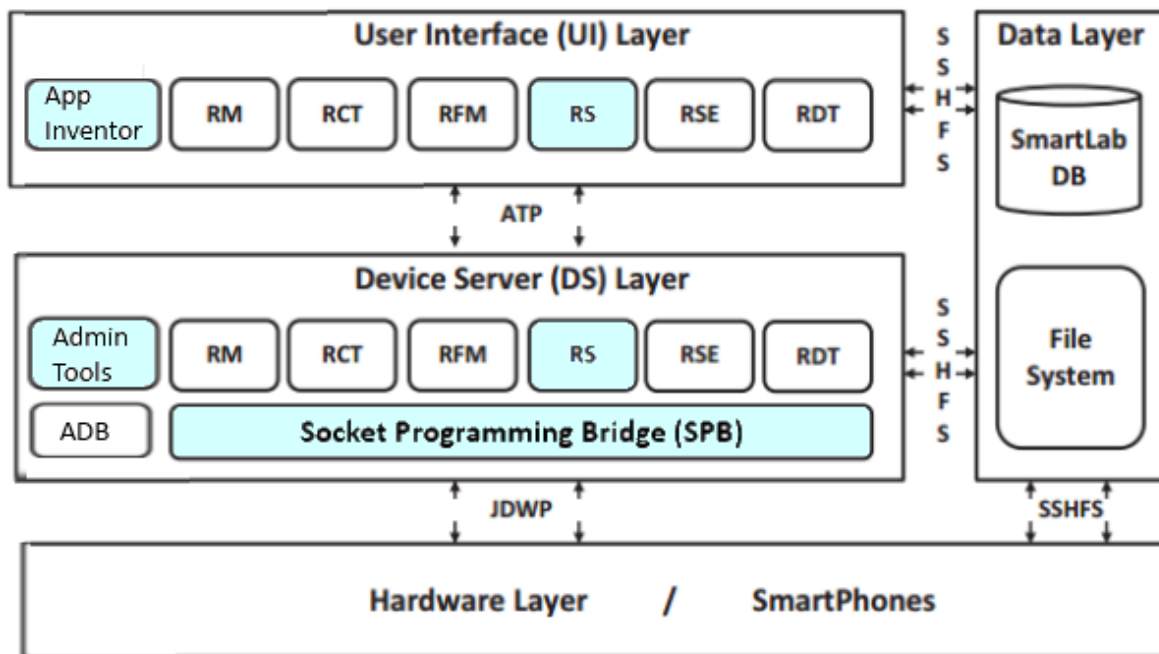
Η επικοινωνία μεταξύ του συστήματος αρχείων του διακομιστή ιστού και του συστήματος αρχείων του εξυπηρετητή γίνεται μέσω του SSHFS πρωτοκόλλου. Ο διακομιστής ιστού χρησιμοποιεί επίσης βάση δεδομένων MySQL 5.5, η οποία χρησιμοποιείται για φύλαξη δεδομένων που σχετίζονται με τους χρήστες, τις συσκευές και τους απομακρυσμένους εξυπηρετητές του SmartLab.

3.3 Επίπεδο εξυπηρετητή συσκευών

Ο εξυπηρετητής συσκευών είναι ο εξυπηρετητής ο οποίος συνδέει τις απομακρυσμένες και τις εικονικές συσκευές με την διεπαφή χρήστη. Αυτή την στιγμή χρησιμοποιείται λειτουργικό σύστημα CentOS 6.3 x64 με 4x2.4 GHz εικονικούς επεξεργαστές, 8GB RAM, 80GB σκληρό δίσκο. Οι αιτήσεις από την διεπαφή χρήστη γίνονται μέσω Websockets προς τον εξυπηρετητή συσκευών.

ATP

Το ATP (ADB Tunnel Protocol) είναι ένα πρωτόκολλο το οποίο τρέχει πάνω από τα Websockets, έτσι ώστε να μπορεί να επικοινωνεί ο εξυπηρετητής συσκευών με την διεπαφή χρήστη και το αντίθετο. Οι αιτήσεις ATP (από την διεπαφή χρήστη προς τον εξυπηρετητή συσκευών) στην αρχιτεκτονική με βάση το εργαλείο ADB μεταφράζονταν σε καλέσματα της βιβλιοθήκης DDMLIB για μεταφορά αρχείων, καταγραφή οθονών, κλπ., καθώς επίσης και για το υποσύστημα των κλικ. Με την αρχιτεκτονική με βάση το SPB πρωτόκολλο, οι αιτήσεις αυτές μεταφράζονται σε μηνύματα του πρωτοκόλλου για την προσομοίωση των λειτουργιών της βιβλιοθήκης DDMLIB (Σχήμα 3.1).



Σχήμα 3.1 : Συστατικά Αρχιτεκτονικής SmartLab και υποσυστήματα τα οποία επικοινωνούν μέσω του πρωτοκόλλου SPB

Device Plug-n-Play

Η σύνδεση και η αποσύνδεση των συσκευών από τον εξυπηρετητή συσκευών θα πρέπει να ενημερώνει την διεπαφή χρήστη με την κατάσταση της κάθε συσκευής. Επομένως με την αρχιτεκτονική ADB μέσω του ADB εργαλείου, εντολές SQL αποστέλλονται στην MySQL βάση δεδομένων και ενημερώνει την κατάσταση των συσκευών όταν αλλάζει. Με την αρχιτεκτονική SPB, οι εντολές SQL αποστέλλονται, με την διαφορά ότι η σύνδεση των συσκευών δεν ενημερώνεται μέσω του ADB, αλλά μέσω μιας TCP σύνδεσης μαζί με το μήνυμα που περιέχει την μοναδική ταυτότητα της συσκευής.

Περιορισμοί εξυπηρετητή συσκευών

Κάθε εξυπηρετητής συσκευών με την αρχιτεκτονική βάση ADB μπορεί να υποστηρίξει μέχρι και 16 εικονικές συσκευές και θεωρητικά μέχρι και 127 πραγματικές, λόγω των περιορισμών του εξυπηρετητή ADB που θα παρουσιαστεί στο επόμενο κεφάλαιο. Προκειμένου να ξεπεραστούν προβλήματα αστάθειας που υπάρχουν στις πραγματικές συσκευές, κάθε εξυπηρετητής συσκευών συνδέεται με είτε 16 εικονικές συσκευές είτε με 16 πραγματικές. Αυτό είναι συμβατό με την αρχιτεκτονική του datacenter το οποίο μπορεί να υποστηρίξει πολλούς εικονικούς εξυπηρετητές συσκευών. Με πολλούς εικονικούς εξυπηρετητές συσκευών, το σύστημα προσφέρει γραμμική επεκτασιμότητα. Με την αρχιτεκτονική SPB, λόγω του ότι δεν υπάρχει πλέον το εργαλείο ADB, δεν υπάρχουν περιορισμοί στον αριθμό συσκευών που υποστηρίζει ο κάθε εξυπηρετητής, επομένως ούτε και στην επεκτασιμότητα του συστήματος.

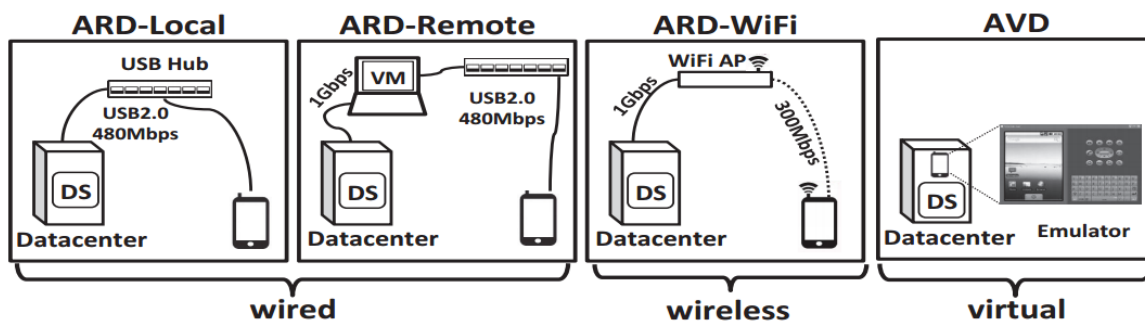
Διαχείριση εξυπηρετητή συσκευών

Επιπλέον από τον εξυπηρετητή JAVA, κάθε εξυπηρετητής συσκευών χρησιμοποιεί Apache καθώς και PHP. Ο τοπικός διακομιστής ιστού είναι υπεύθυνος να προσφέρει μια συλλογή από εργαλεία διαχείρισης, τα οποία επιτρέπουν συντήρηση

3.4 Επίπεδο υλικού

Υλικό και Λειτουργικό

Το υλικό του συστήματος SmartLab αποτελείται από τις συσκευές Android καθώς και το κέντρο δεδομένων. Πιο συγκεκριμένα το SmartLab χρησιμοποιεί περισσότερες από 40 συσκευές Android (π.χ. HTC Desire, Desire S, Hero Sensation, Google Nexus S, Nokia N900 κλπ.). Η πλειοψηφία των συσκευών είχε λειτουργικό Android 2.1-2.3. Οι συσκευές αυτές έγιναν rooted καθώς και αναβαθμίστηκαν σε λειτουργικό Android 4.0.4,



Σχήμα 3.2 : Λεπτομέρειες Σύνδεσης συσκευών με εξυπηρετητή που υποστηρίζονται στην αρχιτεκτονική ADB

χρησιμοποιώντας προσαρμοσμένο ROM. Οι αναβαθμισμένες συσκευές επιτρέπουν πολύ μεγαλύτερη ευελιξία.

Συνδέσεις ADB

Το σύστημα προσφέρει πολλούς τρόπους σύνδεσης (Σχήμα 3.2). Οι πλειοψηφία των συσκευών είναι συνδεδεμένες με τον εξυπηρετητή μέσω ARD-Local τρόπου, χρησιμοποιώντας USB κόμβους. Στο σύστημα επιπλέον συσκευές είναι συνδεδεμένες από το εργαστήριο μέσω ARD-Remote τρόπου. Με τον τρόπο αυτό σύνδεσης, υπάρχουν οι δυνατότητες έτσι ώστε το σύστημα να επεκταθεί και εκτός του τμήματος. Τέλος, μερικές από τις συσκευές εντός του τμήματος είναι συνδεδεμένες με ARD-WIFI τρόπο. Η επικοινωνία μεταξύ των συσκευών και του κάθε εξυπηρετητή συσκευών είναι ασφαλισμένη, εφόσον κάθε συσκευή είναι συνδεδεμένη μέσω του ιδιωτικού δικτύου μειώνοντας έτσι τις πιθανότητες για κακόβουλες επιθέσεις.

Συνδέσεις SPB

Με την αρχιτεκτονική SPB, το σύστημα επιτρέπει στις συσκευές να συνδέονται στους εξυπηρετητές μέσω TCP σύνδεσης, και όχι μέσω του εργαλείου ADB. Επομένως ο τρόπος σύνδεσης των συσκευών είναι ο ARD-WIFI για όλες τις συσκευές (Σχήμα 3.3). Εφόσον

πλέον οι συσκευές είναι συνδεδεμένες με είτε WIFI είτε 3G, αυτό σημαίνει πως δεν περιορίζονται από USB καλώδια, αλλά μπορούν να μεταφέρονται. Με την αρχιτεκτονική ADB, το πρόβλημα με τις συσκευές που είναι συνδεδεμένες με ARD-WIFI είναι πως υπάρχει πρόβλημα ασφαλείας εάν οι συσκευές δεν βρίσκονται σε ιδιωτικό δίκτυο, αφού οποιοσδήποτε μπορεί να συνδεθεί στις συσκευές και εκτελέσει κακόβουλες επιθέσεις. Με την αρχιτεκτονική SPB, το πρόβλημα αυτό δεν υπάρχει, αφού δεν είναι ο εξυπηρετητής που συνδέεται στις συσκευές, αλλά οι συσκευές σε αυτόν, επομένως δεν μπορεί να γίνει κακόβουλη επίθεση στις συσκευές, σε όποιο δίκτυο κι αν βρίσκονται.

3.5 Αρχές και μεθοδολογία σχεδιασμού

Η αρχιτεκτονική του SmartLab επικεντρώνεται σε μερικούς στόχους, όπως δομημένα στοιχεία, κλιμάκωση και επεκτασιμότητα. Ο σχεδιασμός του συστήματος έγινε με ένα «άπληστο» τρόπο από κάτω προς τα πάνω. Σε κάθε επίπεδο, κάθε εναλλακτική επιλογή λήφθηκε υπόψη, αλλά υλοποιήθηκε η πιο αποδοτική λύση. Για κάθε επίπεδο έγινε προσεχτική μελέτη, έτσι ώστε να υπάρχει ευελιξία και να μπορεί γρήγορα να εξελιχθεί το σύστημα.

Κεφάλαιο 4

Υλοποίηση Πρωτοκόλλου SPB

4.1	Εισαγωγή	20
4.2	Εργαλείο ADB και οι περιορισμοί του	21
4.3	Ανασκόπηση του SPB πρωτοκόλλου	23
4.4	Υλοποίηση εξυπηρετητή SPB	25
4.5	Υλοποίηση πελάτη SPB	32

4.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά θα μιλήσουμε γενικά για το εργαλείο ADB και στην συνέχεια για τους περιορισμούς που έχει στο σύστημα SmartLab. Στη συνέχεια θα εξηγήσουμε για την νέα αρχιτεκτονική του συστήματος, SPB, η οποία προσπερνά του περιορισμούς της παλαιότερης αρχιτεκτονικής και τον τρόπο λειτουργίας του νέου πρωτοκόλλου επικοινωνίας. Στο υποκεφάλαιο 4.4 θα μιλήσουμε πιο αναλυτικά για την υλοποίηση του εξυπηρετητή SPB και το πώς υλοποιεί τα υποσυστήματα. Στο τελευταίο μέρος αυτού του κεφαλαίου θα δούμε την υλοποίηση του πελάτη SPB, σε λειτουργικό σύστημα Android, και την διεπαφή που παρέχει.

 Android Development Tools		
/tools		
SDK Tools		Debuggers /Emulators android, ddms, emulator
		Testers/Stress Testers monkey, monkey runner
		Profilers dmtracedump, systrace, traceview, hprof-conv
		Graphical/Layout Optimizers hierarchyviewer, draw9patch, layout-opt
		.apk tools proguard, zipalign
		Miscellaneous mkshcard, sqlite3
/platform-tools		
Platform Tools		Debuggers adb
		Shell tools shell, bmgr, logcat
		Miscellaneous aidl, aapt, dexdump, dx

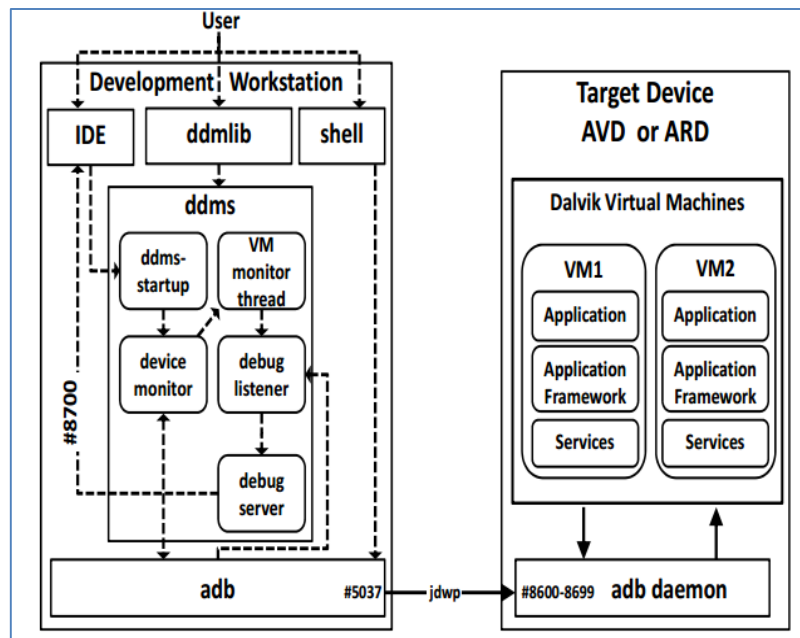
Σχήμα 4.1 : Android Development Tools

4.2 Εργαλείο ADB και οι περιορισμοί του

Στην ενότητα αυτή θα εξηγήσουμε σε βάθος το Android Debug Bridge (ADB), το οποίο είναι υπεύθυνο για όλη την επικοινωνία μεταξύ του εξυπηρετητή και των συσκευών στην προηγούμενη αρχιτεκτονική του συστήματος, καθώς και τους περιορισμούς που είχε στο σύστημα.

Η εντολή ADB είναι μέρος των εργαλείων πλατφόρμας, τα οποία παρέχονται από τα εργαλεία ανάπτυξης λογισμικού Android και επιτρέπουν ανάπτυξη, εγκατάσταση και έλεγχο των εφαρμογών σε πραγματικές ή εικονικές συσκευές. Τα εργαλεία αυτά χωρίζονται σε δύο κατηγορίες (βλέπε σχήμα 4.1), τα SDK εργαλεία, τα οποία είναι ανεξάρτητα πλατφόρμας, και τα εργαλεία πλατφόρμας, τα οποία είναι προσαρμοσμένα να υποστηρίζουν χαρακτηριστικά της τελευταίας πλατφόρμας Android.

Τα εργαλεία πλατφόρμας παρέχουν επίσης εργαλεία κελύφους στα οποία μπορεί να υπάρξει πρόσβαση μέσω του ADB, όπως το εργαλείο bmgr, το οποίο επιτρέπει



Σχήμα 4.2 : Android Debug Bridge connection (ADB)

αλληλεπίδραση με το backup manager της συσκευής, το εργαλείο logcat, το οποίο παρέχει μηχανισμό για την συλλογή πληροφοριών για σκοπούς αποσφαλμάτωσης.

Οι εφαρμογές σε λειτουργικό Android μπορούν να αναπτυχθούν σε οποιοδήποτε συμβατό περιβάλλον ανάπτυξης (π.χ. Eclipse) και ο κώδικας τους είναι γραμμένος σε κώδικα JAVA με το Android SDK. Στη συνέχεια μετατρέπονται σε Dalvik-compatible εκτελέσιμα (.dex) αρχεία χρησιμοποιώντας το εργαλείο πλατφόρμας dx, συρρικνώνονται με τη βοήθεια του εργαλείου proguard και εγκαθίστανται στην συσκευή μέσω της εντολής adb install, του εργαλείου ADB. Όπως φαίνεται στο σχήμα 4.2, κάθε εφαρμογή είναι ενθυλακωμένη στη δική της διεργασία και εκτελείται στη δική της εικονική μηχανή Dalvik. Επιπλέον, κάθε εικονική μηχανή Dalvik αφήνει ανοικτή μια θύρα για σκοπούς αποσφαλμάτωσης από τοπικούς ή απομακρυσμένους σταθμούς εργασίας, μέσω της ADB υπηρεσίας (adbd). Ο adbd τρέχει σε κάθε συσκευή και εξυπηρετεί στην σύνδεση της συσκευής με τον εξυπηρετητή.

Περιορισμοί του ADB στο σύστημα SmartLab

- Οι συσκευές χρειάζεται να είναι συνδεδεμένες με τον εξυπηρετητή του συστήματος, μέσω USB καλωδίων, έτσι ώστε να μπορεί να υπάρξει σύνδεση μέσω του εργαλείου ADB. Αυτό δεν επιτρέπει στις συσκευές να μετακινούνται, το οποίο είναι ένας σημαντικός περιορισμός, αφού δεν μπορούν να εκτελεστούν πειράματα που απαιτούν μετακίνηση στο χώρο (π.χ. γεοτοποθέτησης).
- Τα υποσυστήματα του SmartLab, λόγω του ότι επικοινωνούν με τις συσκευές μέσω του εργαλείου ADB, έχουν το μειονέκτημα ότι περιορίζονται από τους χρόνους απόκρισης του εργαλείου.
- Λόγο του εργαλείου ADB, όσο αυξάνεται ο αριθμός των συσκευών που είναι συνδεδεμένες με τον εξυπηρετητή, οι χρόνοι απόκρισης του κάθε υποσυστήματος επηρεάζονται αρνητικά.
- Λόγο της χρήσης του εργαλείου ADB, η συμπεριφορά των συσκευών πολλές φορές είναι ασταθής, και χρειάζεται συχνή παρακολούθηση του συστήματος.
- Με τη χρήση του WIFI ADB, υπήρχαν προβλήματα ασφαλείας, εφόσον δεν συνδέονταν οι συσκευές στον εξυπηρετητή, αλλά ο εξυπηρετητής, σε αυτές, επομένως θα μπορούσε κάποιος κακόβουλα να συνδεθεί σε αυτές.

4.3 Ανασκόπηση του SPB πρωτοκόλλου

Στην ενότητα αυτή θα εξηγήσουμε σε βάθος το *Socket Programming Bridge (SPB)*, το οποίο είναι υπεύθυνο για όλη την επικοινωνία μεταξύ του εξυπηρετητή και των συσκευών στην καινούργια αρχιτεκτονική του συστήματος.

Το πρωτόκολλο SPB, έχει ως στόχο την αντικατάσταση της επικοινωνίας όλων των υποσυστημάτων του SmartLab αντί να γίνεται μέσω του εργαλείου ADB μέσω USB καλωδίων, να γίνεται μέσω δικτύου, αφού θα ξεπεραστούν πολλοί από τους περιορισμούς που υπάρχουν.

Η ιδέα πίσω από τον σχεδιασμό του πρωτοκόλλου αυτού, είναι η προσομοίωση μερικών βασικών λειτουργιών που παρέχει το εργαλείο ADB, μέσω μιας διεπαφής σχεδόν όμοιας με αυτή του ADB. Ο λόγος που οι δύο διεπαφές πρέπει να είναι όμοιες, είναι για λόγους ευκολίας υλοποίησης, αφού θα πρέπει να αλλαχθούν όλα τα υποσυστήματα.

Οι λειτουργίες που προσωμειώνει το πρωτόκολλο SPB είναι οι λειτουργίες οι οποίες είναι απαραίτητες για την σωστή λειτουργία των υποσυστημάτων του SmartLab.

Διεπαφή πρωτοκόλλου SPB

Η διεπαφή του πρωτοκόλλου έχει ως εξής:

- **Shell (command, deviceID):** Η εντολή shell παίρνοντας δύο παραμέτρους, την εντολή για εκτέλεση, καθώς και την ταυτότητα της συσκευής, εκτελεί την συγκεκριμένη εντολή στην συσκευή και επιστρέφει την έξοδο της.
- **Click (deviceID, type, xs, ys, xe, ye):** Η εντολή click εκτελεί το συγκεκριμένο κλικ παίρνοντας ως παραμέτρους, την ταυτότητα της συσκευής, τον τύπο του κλικ (οι τύποι των κλικ επεξηγούνται στο τέλος του τμήματος 4.5) και τις διαστάσεις στις οποίες πρέπει να εκτελεστεί το κλικ.
- **Screenshot (deviceID):** Η εντολή screenshot, παίρνοντας ως παράμετρο την ταυτότητα της συσκευής, επιστρέφει μια ροή οθονών από τη συσκευή στον εξυπηρετητή.
- **PushFile/PullFile (deviceID, file):** Οι εντολές αυτές παίρνοντας ως παραμέτρους την ταυτότητα της συσκευής, καθώς και το αρχείο, μεταφέρει το αρχείο από τον εξυπηρετητή στην συσκευή και από την συσκευή στο εξυπηρετητή αντίστοιχα.

Μηνύματα πρωτοκόλλου SPB

Ο τρόπος με τον οποίο το πρωτόκολλο επικοινωνεί από τον εξυπηρετητή στις συσκευές, είναι μέσω μηνυμάτων που αποτελούν το πρωτόκολλο αυτό. Βάση των μηνυμάτων που παραλαμβάνονται στις συσκευές, καθορισμένες λειτουργίες εκτελούνται, έτσι ώστε να επιτευχθεί η κάθε λειτουργία για το αντίστοιχο υποσύστημα.

Το πρωτόκολλο από πλευράς πελάτη έχει ως εξής:

1. Η συσκευή συνδέεται στον εξυπηρετητή αποστέλλοντας μαζί και την μοναδική της ταυτότητα
2. Η συσκευή αναμένει μήνυμα λειτουργίας
3. Όταν λάβει μήνυμα λειτουργίας εκτελεί την λειτουργία που αντιστοιχεί στο συγκεκριμένο μήνυμα, και απαντά στον εξυπηρετητή με την έξοδο της λειτουργίας αυτής
4. Επέστρεψε στο βήμα 2

Εάν για οποιοδήποτε λόγο η συσκευή αποσυνδεθεί από τον εξυπηρετητή, τότε προσπαθεί να ξανασυνδεθεί μετά από κάποιο χρονικό διάστημα.

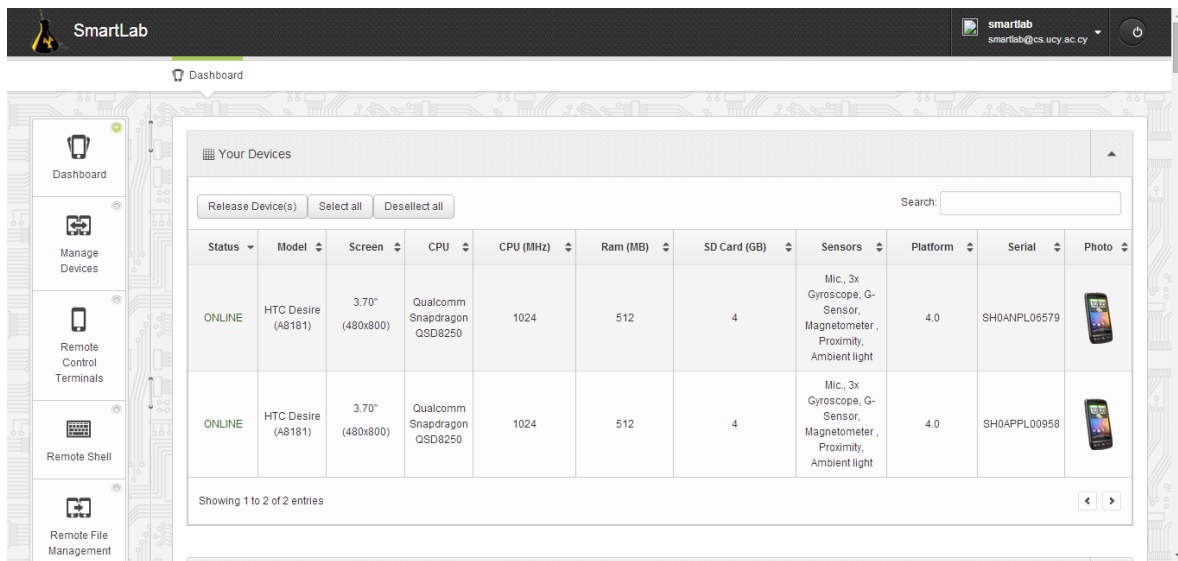
Το πρωτόκολλο από πλευράς εξυπηρετητή έχει ως εξής:

1. Ανάμενε μέχρι μια συσκευή να συνδεθεί
2. Ενημέρωσε τον πίνακα με την ταυτότητα και την υποδοχή για αυτή την συσκευή και ενημέρωσε την βάση δεδομένων, ότι η συγκεκριμένη συσκευή είναι διαθέσιμη
3. Ανάλογα με το μήνυμα που λάβεις από τον διακομιστή ιστού, στείλε το κατάλληλο μήνυμα στις ανάλογες συσκευές και ανάμενε αποτέλεσμα, αν υπάρχει
4. Επέστρεψε στο βήμα 3

Τα μηνύματα λειτουργίας είναι της μορφής `command: [options]`, όπου `command`, είναι η αντίστοιχη εντολή για κάθε υποσύστημα που θα πρέπει να εκτελεστεί.

4.4 Υλοποίηση εξυπηρετητή SPB

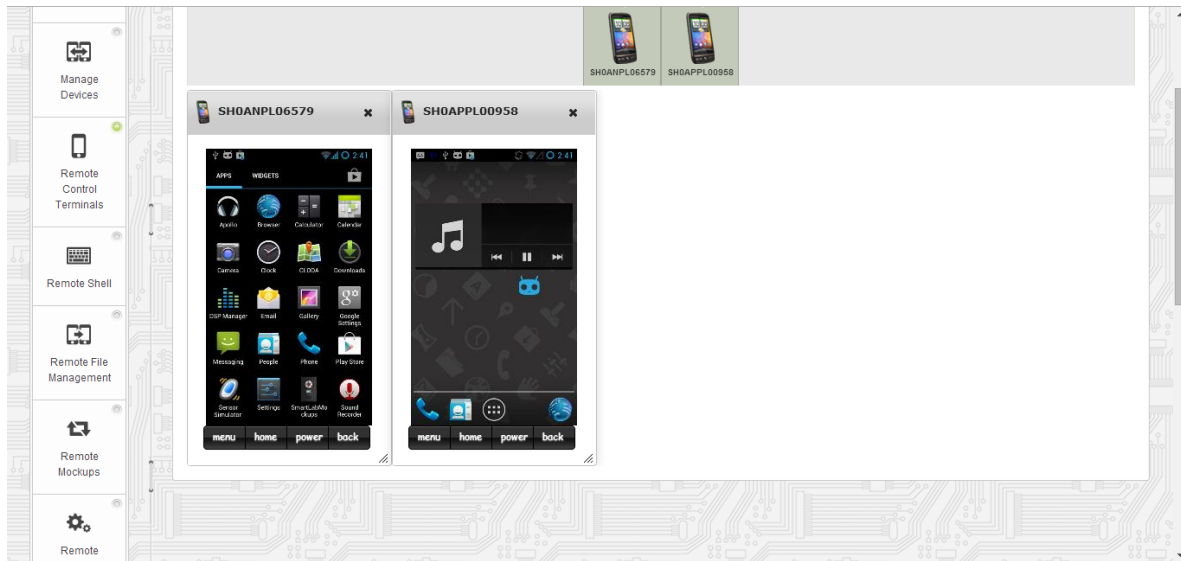
Στο προϋπάρχον σύστημα, εφόσον ο εξυπηρετητής ήταν γραμμένος σε γλώσσα προγραμματισμού JAVA , για λόγους ευκολίας και συμβατότητας παραμένει ως έχει. Σκοπός στην υλοποίηση του εξυπηρετητή ήταν να μπορεί εύκολα να αλλάξει βάσει της



Σχήμα 4.3 : Κρατημένες συσκευές χρήστη στο σύστημα (Dashboard)

καινούργιας αρχιτεκτονικής. Γι' αυτό τον λόγο η υλοποίηση είναι μια ξεχωριστή διεπαφή, η οποία θα αντικαταστήσει τις λειτουργίες του Android Debug Bridge (ADB). Η υλοποίηση αυτή προσομοιώνει τις λειτουργίες τις οποίες παρέχει ο ADB, για παράδειγμα την αποστολή αρχείων από την συσκευή προς τον εξυπηρετητή, καθώς και από τον εξυπηρετητή προς την συσκευή, την εγκατάσταση εφαρμογών, αποστολή στιγμιότυπων της οθόνης της συσκευής, την αποστολή εντολών στο κέλυφος της συσκευής καθώς και την προσομοίωση των κλικ στην συσκευή. Η συγκεκριμένη υλοποίηση δεν προσομοιώνει μόνο τις λειτουργίες αυτές, αλλά και την διεπαφή του ADB, επιτρέποντας έτσι την εναλλαγή του εξυπηρετητή από τον ADB στο SPB με πολύ μικρές αλλαγές. Ο εξυπηρετητής είναι πλέον συνδεδεμένος με τις συσκευές μέσω TCP συνδέσεων αντί με USB, και πλέον δεν υπάρχουν περιορισμοί στις λειτουργίες του όπως πριν.

Ο εξυπηρετητής είναι υπεύθυνος για την σύνδεσή του διακομιστή ιστού και των κινητών συσκευών.



Σχήμα 4.4: Remote control terminals subsystem (User Interface)

Επομένως ο εξυπηρετητής επικοινωνεί με των διακομιστή ιστού μέσω web-socket σύνδεσης , και με τις συσκευές μέσω TCP. Βάση των λειτουργιών που επιθυμεί ο χρήστης από την διεπαφή του διακομιστή ιστού ο εξυπηρετητής εκτελεί την αντίστοιχη συμπεριφορά, η οποία θα επικοινωνήσει με τις συσκευές του χρήστη βάση του πρωτοκόλλου SPB που έχει περιγραφηκε στο προηγούμενο κεφάλαιο.

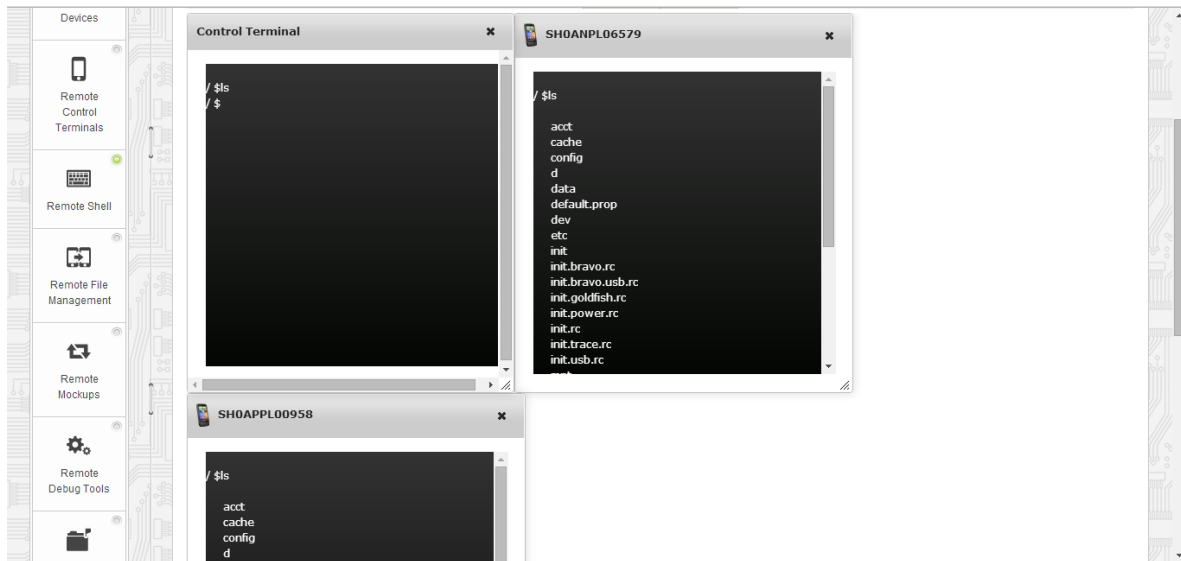
Ο εξυπηρετητής για να γνωρίζει σε πιά συσκευή πρέπει να απαντά, κρατά μια συσχέτιση μεταξύ των μοναδικών ταυτοτήτων των συσκευών μαζί με την TCP υποδοχή της.

Βασικές διαφορές από την προηγούμενη υλοποίηση του εξυπηρετητή στις διάφορες υπολειτουργίες ακολουθούν πιο κάτω.

Remote Control Terminals:

Στην προηγούμενη υλοποίηση ο εξυπηρετητής ζητούσε κατευθείαν οθόνη τις συσκευής η οποία είχε μέγεθος 1.5 MB, όπου στην συνέχεια την συμπίεζε και τις άλλαζε το μέγεθος έτσι ώστε να ανταποκρίνεται στο μέγεθος της πραγματικής οθόνης της συσκευής.

Η πιο πάνω διαδικασία ήταν πολύ χρονοβόρα, αφού θα έπρεπε κάθε φορά για κάθε συσκευή και κάθε οθόνη να μεταφέρεται 1.5MB, με αποτέλεσμα όταν χρησιμοποιούνται πολλές συσκευές ο χρόνος απόκρισης του υποσυστήματος να γίνεται πολύ μεγάλος.



Σχήμα 4.5: Remote Shells subsystem (User Interface)

Στην καινούργια υλοποίηση, ο εξυπηρετητής ζητά μέσω του πρωτοκόλλου πιο πάνω, μόνο μια φορά στην αρχή, να αρχίσει την διαδικασία των οθονών. Στη συνέχεια μια ροή από οθόνες ξεκινά να αποστέλλεται στον εξυπηρετητή. Με αυτό τον τρόπο δεν χρειάζεται κάθε φορά να γίνεται αίτηση για οθόνη, γλυτώνοντας έτσι χρόνο. Ο εξυπηρετητής παραλαμβάνει την οθόνη ήδη συμπιεσμένη και στις σωστές διαστάσεις, επομένως απλά την προωθεί στον διακομιστή ιστού. Με αυτό τον τρόπο η οθόνη που μεταφέρεται μέσω του δικτύου είναι μεγέθους 20-40 KB, πετυχαίνοντας έτσι πολύ καλύτερους χρόνους απόκρισης. Ένα άλλο πλεονέκτημα αυτής της προσέγγισης είναι πως όσο μεγαλώνει ο αριθμός των συσκευών, ο χρόνος απόκρισης δεν μειώνεται καθόλου, αφού πλέον δεν υπάρχει το κρίσιμο σημείο στον ADB, και παραλαμβάνονται οι οθόνες από διαφορετικές συσκευές παράλληλα.

Remote Shells:

Στην παλιά υλοποίηση ο εξυπηρετητής πριν εκτελέσει μια εντολή ζητούσε μέσω του ADB SHELL να εκτελεστεί η εντολή στην συσκευή, λαμβάνοντας την έξοδο από την διεργασία αυτή.

Με την καινούργια υλοποίηση μέσω του πρωτοκόλλου, ο εξυπηρετητής ζητά από την συσκευή να εκτελέσει την συγκεκριμένη εντολή κατευθείαν στο κέλυφος, και λαμβάνοντας την έξοδο της εντολής μέσω της TCP σύνδεσης.

Clicks:

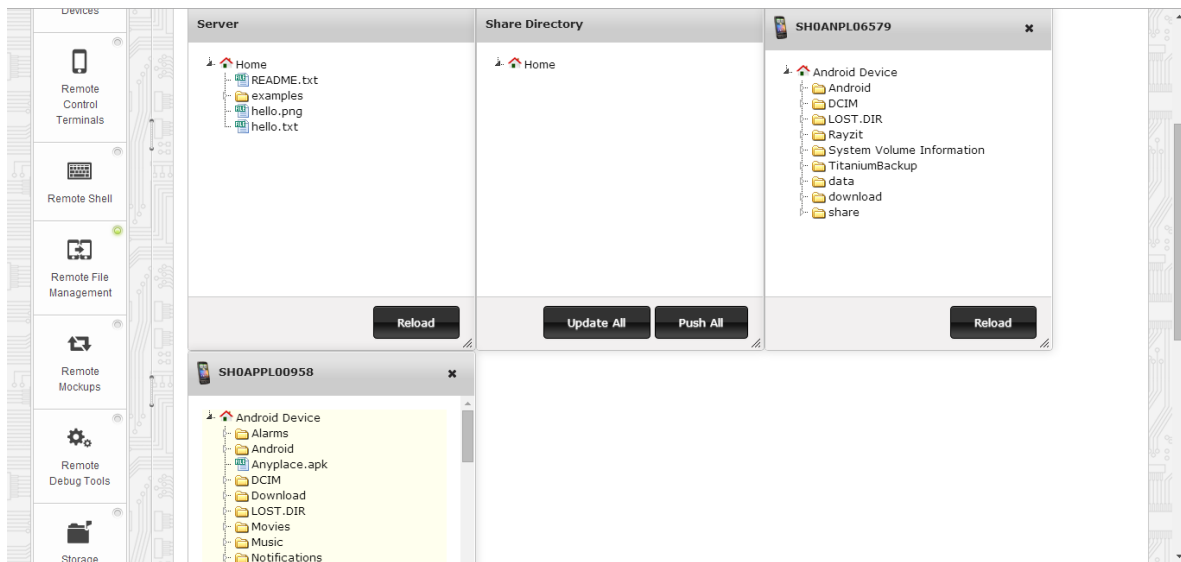
Στην προηγούμενη υλοποίηση η διαδικασία των κλικ γινόταν με την βοήθεια του ADB καθώς και του Monkey Runner. Τα εργαλεία αυτά εκτός του ότι δεν επέτρεπαν εντελώς παράλληλα την εκτέλεση των κλικ στις συσκευές, η διαδικασία γινόταν μέσω ενός PHP εξυπηρετητή ο οποίος ήταν αποκλειστικά για τον σκοπό αυτό μαζί με την μεταφορά αρχείων. Δηλαδή ο διακομιστής ιστού, μετέφερε κατευθείαν τα κλικ στον PHP εξυπηρετητή μέσω του οποίου γίνονταν τα κλικ στις συσκευές με την βοήθεια των δύο πιο πάνω εργαλείων.

Στην νέα υλοποίηση η διαδικασία δεν χρησιμοποιεί PHP εξυπηρετητή, ADB ή Monkey Runner. Ο διακομιστής ιστού επικοινωνεί κατευθείαν με τον εξυπηρετητή αποστέλλοντας τον συγκεκριμένο κλικ. Στην συνέχεια ο εξυπηρετητή μέσω του πρωτοκόλλου επικοινωνεί με τις συσκευές. Επομένως από πλευράς εξυπηρετητή δεν χρησιμοποιούνται εξωτερικά εργαλεία, ούτε επιπλέον εξυπηρετητές.

Application install:

Η εγκατάσταση εφαρμογών στις συσκευές γινόταν μέσω της εντολής adb install του ADB.

Η εγκατάσταση εφαρμογών στην νέα υλοποίηση γίνεται με την βοήθεια του υποσυστήματος κελύφους.



Σχήμα 4.6: File push/pull subsystem (User Interface)

File push/pull:

Στην παλιά υλοποίηση η διαδικασία της μεταφοράς αρχείων γινόταν με την βοήθεια ενός εξωτερικού PHP εξυπηρετητή και την βοήθεια του εργαλείου ADB. Δηλαδή μέσω του διακομιστή ιστού ο χρήστης επέλεγε τα αρχεία για μεταφορά τα οποία μεταφέρονταν από τον PHP εξυπηρετητή μέσω του ADB στις συσκευές.

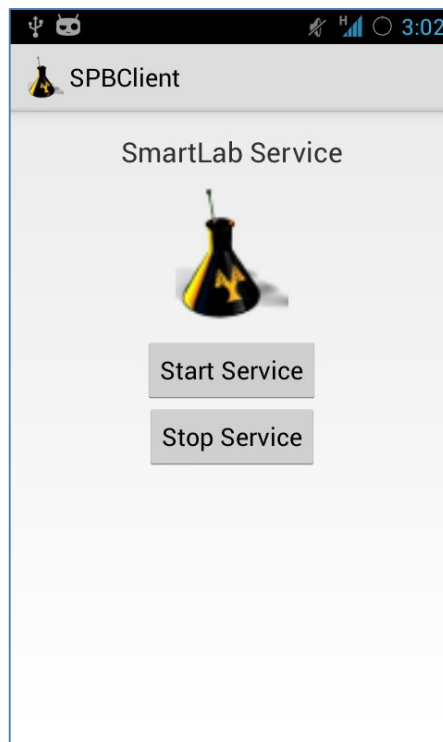
Στην νέα υλοποίηση, μέσω του διακομιστή ιστού ο χρήστης επιλέγει τα αρχεία που θα μεταφερθούν, και στην συνέχεια μέσω του εξυπηρετητή η μεταφορά γίνεται μέσω του πρωτοκόλλου με TCP σύνδεση.

Βασικά πλεονεκτήματα νέας υλοποίησης:

- Στην νέα υλοποίηση οι συσκευές δεν χρειάζεται να είναι συνδεδεμένες με τον εξυπηρετητή, μέσω του ADB, αλλά μπορούν να είναι οπουδήποτε υπάρχει διαδίκτυο, είτε WIFI είτε 3G. Αυτό επιτρέπει στις συσκευές να μπορούν να μετακινούνται, επομένως θα μπορούσε ο οποιοσδήποτε να προσφέρει την συσκευή του στο SmartLab. Ένα άλλο πλεονέκτημα στο ότι οι συσκευές μπορούν να

μεταφέρονται, είναι πως θα μπορούσαν να τρέχουν οποιαδήποτε πειράματα, για παράδειγμα γεοτοποθέτησης στις συγκεκριμένες συσκευές.

- Στην νέα υλοποίηση, λόγο του ότι δεν υπάρχει πλέον ο ADB, δεν υπάρχουν προβλήματα με τον χρόνο απόκρισης, αφού τα υποσυστήματα μπορούν να τρέχουν εντελώς παράλληλα.
- Στην νέα υλοποίηση με την αύξηση του αριθμού των συσκευών δεν υπάρχουν προβλήματα με τον χρόνο απόκρισης, αφού πλέον τα δεδομένα περνούν μέσω δικτύου και όχι μέσω του ADB.
- Λόγο της χρήσης του ADB στην προηγούμενη υλοποίηση υπήρχαν περιορισμοί στον αριθμό συνδεδεμένων συσκευών ανά πάσα στιγμή, καθώς και ασταθής συμπεριφορά των συσκευών.
- Με τη χρήση του WIFI ADB, υπήρχαν προβλήματα ασφαλείας, εφόσον δεν συνδέονταν οι συσκευές στον εξυπηρετητή, αλλά ο εξυπηρετητής, σε αυτές, επομένως θα μπορούσε κάποιος κακόβουλα να συνδεθεί σε αυτές.

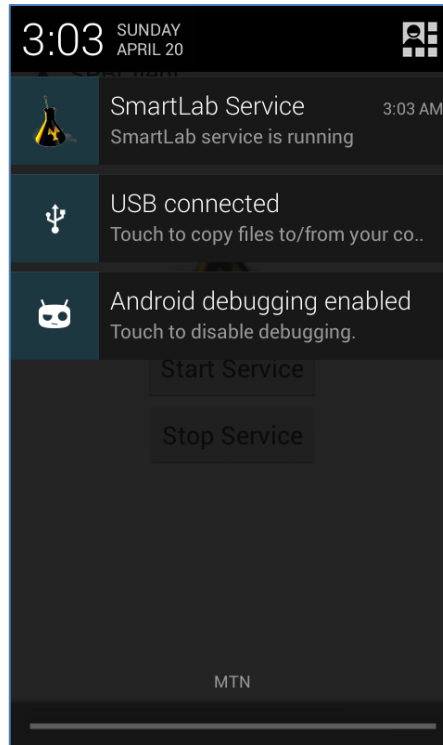


Σχήμα 4.7: Start Screen of the SmartLab client

4.5 Υλοποίηση πελάτη SPB

Λόγο του ότι οι πλειοψηφία των συσκευών είναι σε λειτουργικό Android, ο πελάτης είναι γραμμένος σε android, έτσι καθιστώντας λειτουργήσιμο το νέο πρωτόκολλο στην πλειοψηφία των συσκευών. Στην παλιά υλοποίηση δεν υπήρχε πελάτης στις συσκευές, αλλά χρησιμοποιήθηκε ο ADB, για την λειτουργικότητα των συσκευών. Εφόσον τα βασικά προβλήματα του συστήματος οφείλονταν στον ADB, θα έπρεπε να υλοποιηθεί ένας πελάτης ο οποίος θα προσομειώνει τις προηγούμενες λειτουργίες των συσκευών.

Στην συγκεκριμένη υλοποίηση ο πελάτης αποτελείται από μια απλή διεπαφή χρήστη, η οποία επιτρέπει την εκκίνηση της υπηρεσίας παρασκηνίου, που είναι υπεύθυνη για την λειτουργία του πρωτοκόλλου (Σχήμα 4.7). Μετά την εκκίνηση της υπηρεσίας μια ειδοποίηση εμφανίζεται στη γραμμή ειδοποιήσεων, έτσι ώστε ο χρήστης να γνωρίζει πως η υπηρεσία είναι σε λειτουργία (Σχήμα 4.8).



Σχήμα 4.8: Notification bar with SmartLab service running

Η συγκεκριμένη υπηρεσία εκτελεί την πιο κάτω διαδικασία:

- Σύνδεση με τον εξυπηρετητή και αποστολή της μοναδικής ταυτότητας της.
- Έναρξη τοπικού εξυπηρετητή για εκτέλεση των κλικ καθώς και μιας διεργασίας κελύφους, έτσι ώστε να μπορεί ο χρήστης να εκτελεί εντολές φυλάγοντας καταστάσεις μεταξύ εκτέλεσης διαφορετικών εντολών.
- Αναμονή για ποια διαδικασία να εκτελέσει βάση του πρωτοκόλλου.

Βασικές λειτουργίες για τις διάφορες διαδικασίες ακολουθούν πιο κάτω.

Remote Control Terminals:

Η συσκευή μόλις λάβει μήνυμα για την διαδικασία οθονών ξεκινά την πιο κάτω εκτέλεση:

- Ξεκινά το εργαλείο screencap, έτσι ώστε να πάρει ένα στιγμιότυπο οθόνης
- Συμπίεζεται η οθόνη έτσι ώστε να έχει μικρό μέγεθος, 20-40KB, χωρίς να μειώνεται πολύ η ποιότητα της εικόνας.
- Η εικόνα αποστέλλεται στον εξυπηρετητή.

Η υλοποίηση για καλύτερο χρόνο απόκρισης, υπάρχει, όπου αντί η διαδικασία να γίνεται από ένα μόνο νήμα, πολλά νήματα φυλάσσουν τις εικόνες, και ένα άλλο τις αποστέλλει στον εξυπηρετητή.

Η υλοποίηση αυτή λόγω συσκευών που δεν είναι πολυπύρηνες δεν έχει δοκιμαστεί πειραματικά.

Remote Shells:

Η συσκευή μόλις λάβει μήνυμα για την διαδικασία κελύφους ξεκινά την πιο κάτω εκτέλεση:

- Βάση του μηνύματος, διαβάζει την εντολή που πρέπει να εκτελεστεί.
- Η συγκεκριμένη εντολή αποστέλλεται στην διεργασία κελύφους και εκτελείται.
- Το αποτέλεσμα της εντολής διαβάζεται από το κέλυφος και επιστρέφεται στον εξυπηρετητή.

Clicks:

Η συσκευή μόλις λάβει μήνυμα για την διαδικασία των κλικ ξεκινά την πιο κάτω εκτέλεση:

- Βάση του μηνύματος, διαβάζει το είδος κλικ που πρέπει να εκτελεστεί (power,back,home,menu,click,drag).
- Η συγκεκριμένη εντολή αποστέλλεται στον τοπικό εξυπηρετητή που είναι υπεύθυνος για τα κλικ.
- Ο τοπικός εξυπηρετητής με την βοήθεια του εργαλείου Uiautomator εκτελεί το κλικ.

File push/pull:

Η συσκευή μόλις λάβει μήνυμα για την διαδικασία της μεταφοράς αρχείου ξεκινά την πιο κάτω εκτέλεση:

- Βάση του μηνύματος, διαβάζει το αρχείο που πρέπει να αποσταλεί είτε από την συσκευή στον εξυπηρετητή, είτε από τον εξυπηρετητή προς την συσκευή.
- Το συγκεκριμένο αρχείο αποστέλλεται και φυλάγεται.



Σχήμα 4.9: Είδη κλικ που υποστηρίζονται από το υποσύστημα

Είδη κλικ:

- **Power:** Το συγκεκριμένο κλικ είναι υπεύθυνο για το πάτημα του κουμπιού power. Δηλαδή εάν η οθόνη της συσκευής είναι απενεργοποιημένη την ενεργοποιεί, και το αντίθετο.
- **Back:** Το συγκεκριμένο κλικ είναι υπεύθυνο για το πάτημα του κουμπιού back.
- **Home:** Το συγκεκριμένο κλικ είναι υπεύθυνο για το πάτημα του κουμπιού home.
- **Menu:** Το συγκεκριμένο κλικ είναι υπεύθυνο για το πάτημα του κουμπιού menu.
- **Click:** Το συγκεκριμένο κλικ είναι υπεύθυνο για το πάτημα στην οθόνη της συσκευής σε συντεταγμένες x, y .
- **Drag:** Το συγκεκριμένο κλικ είναι υπεύθυνο για συρόμενο πάτημα της οθόνης από συντεταγμένες $x1, y1$ σε συντεταγμένες $x2, y2$.

(Βλέπε Σχήμα 4.7 για τα πιο πάνω κουμπιά)

Κεφάλαιο 5

Εργαλεία Διαχείρισης SmartLab

5.1	Εισαγωγή	36
5.2	Ανασκόπηση και περιγραφή εργαλείων διαχείρισης	36
5.3	Υλοποίηση	40

5.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά θα επεξηγηθούν τα εργαλεία διαχείρισης, η αναγκαιότητα των εργαλείων αυτών στο σύστημα SmartLab καθώς και οι περιορισμοί τους. Στη συνέχεια θα παρουσιαστούν λεπτομέρειες της υλοποίησης των εργαλείων αυτών, όπως για παράδειγμα τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση τόσο της λειτουργίας των εργαλείων καθώς και την διεπαφή χρήστη αυτών των εργαλείων.

5.2 Ανασκόπηση και περιγραφή εργαλείων διαχείρισης

Μέχρι τώρα στο σύστημα SmartLab δεν υπήρχε τρόπος να γίνει συντήρηση των κινητών συσκευών με απομακρυσμένο τρόπο. Δηλαδή αν κάποια συσκευή έπρεπε για παράδειγμα να επανεκκινηθεί ή να επαναφερθεί σε κατάσταση στην οποία δούλευε κανονικά, θα

```

1  #!/bin/bash
2
3
4
5  declare -a devices
6  i=0
7
8  while read line
9  do
10     devices[i]=$line
11     i=$((i+1))
12 done < "devices.txt"
13
14
15 for device in "${devices[@]}"
16 do
17     wipe="adb -s $device shell wipe data"
18     restart="adb -s $device reboot"
19     $wipe &
20     $restart &
21
22 done
23
24
25 echo "Now will sleep"
26 sleep 180
27 echo "Back"
28 postwipe="monkeyrunner postwipe.py"
29 $postwipe
30 rm devices.txt

```

Σχήμα 5.1: Κώδικας που εκτελείται κατά την διάρκεια λειτουργίας του εργαλείου Wipe

έπρεπε να γίνει χειρωνακτικά η διαδικασία πάνω στην συσκευή. Επίσης για κάθε συσκευή η διαδικασία θα έπρεπε να γίνει ξεχωριστά αφού δεν θα υπήρχε τρόπος να γίνει για περισσότερες συσκευές ταυτόχρονα. Η ανάγκη αυτή υπάρχει στο σύστημα, αφού οι κινητές συσκευές χρειάζονται τακτικά παρακολούθηση και συντήρηση. Επίσης υπάρχει μεγαλύτερη ανάγκη για τέτοια εργαλεία, αφού μελλοντικός σκοπός του συστήματος είναι να επεκταθεί, έτσι ώστε να μπορούν οι συσκευές να μην βρίσκονται μόνο συνδεδεμένες στον εξυπηρετητή του SmartLab, αλλά να βρίσκονται σε διάφορους άλλους εξυπηρετητές

```

1  #!/bin/bash
2
3
4  declare -a devices
5  i=0
6
7  while read line
8  do
9      devices[i]=$line
10     i=$((i+1))
11 done < "devices.txt"
12
13 for device in "${devices[@]}"
14 do
15     adb="adb -s $device backup -apk -shared -all -f ./backup/$device.ab"
16     $adb &
17 done
18
19 monkeyRunner="monkeyrunner backup.py"
20 $monkeyRunner
21 rm devices.txt

```

Σχήμα 5.2: Κώδικας που εκτελείται κατά την διάρκεια λειτουργίας του εργαλείου Backup

από ενδιαφερόμενα άτομα. Λόγο λοιπόν της ανάγκης αυτής, έχουν υλοποιηθεί εργαλεία τα οποία επιτρέπουν συντήρηση των κινητών συσκευών, όπως για παράδειγμα επαναφορά της κατάστασης των συσκευών σε μια λειτουργήσιμη τους κατάσταση, δημιουργία αντιγράφου της κατάστασης τους, καθώς και εκκαθάριση της συσκευής και επαναφοράς της στην εργοστασιακή της κατάσταση, οι λειτουργίες και οι λεπτομέρειες υλοποίησης θα παρουσιαστούν με λεπτομέρεια στο επόμενο υποκεφάλαιο. Τα εργαλεία αυτά παρέχονται στον διαχειριστή των συσκευών, έτσι ώστε μόνο ο διαχειριστής να μπορεί να εφαρμόζει τα εργαλεία αυτά, και όχι οποιοσδήποτε χρήστης. Τα εργαλεία αυτά έχουν κάποιους περιορισμούς, αφού η λειτουργία τους βασίζεται στο εργαλείο ADB. Επομένως οι συσκευές για να μπορούν συντηρηθούν από τον διαχειριστή θα πρέπει να είναι συνδεδεμένες με τον εξυπηρετητή μέσω ADB σύνδεσης, η οποία θα μπορούσε να είναι είτε ενσύρματη (ARD-Local/ARD-Remote), είτε ασύρματη (WIFI-Remote).

```

1  #!/bin/bash
2
3  declare -a devices
4  i=0
5
6  file=$1
7
8  echo $file
9
10 while read line
11 do
12     devices[i]=$line
13     i=$((i+1))
14 done < "devices.txt"
15
16
17 for device in "${devices[@]}"
18 do
19     restore="adb -s $device restore $file"
20
21     $restore &
22
23
24 done
25
26 backup="monkeyrunner restore.py"
27 $backup
28
29 rm devices.txt
30

```

Σχήμα 5.3: Κώδικας που εκτελείται κατά την διάρκεια λειτουργίας του εργαλείου Restore

5.3 Υλοποίηση

Η εργαλειοθήκη αυτή έχει υλοποιηθεί με τα πιο κάτω:

- **Django/Python:** Για την υλοποίηση της διεπαφής χρήστη των εργαλείων, καθώς και την επικοινωνία με την βάση δεδομένων. Επίσης το εργαλείο Django, παρέχει μια διεπαφή, όπου τα εργαλεία καλούνται να εκτελεστούν με HTTP αιτήσεις.
- **MonkeyRunner:** Για την υλοποίηση των λειτουργιών που πρέπει να γίνουν στην συσκευή ώστε να ολοκληρωθεί η διαδικασία κάθε εργαλείου.
- **UNIX Shell Scripts:** Για την χρησιμοποίηση των λειτουργιών που παρέχει το εργαλείο ADB.

Η συγκεκριμένη εργαλειοθήκη αποτελείται από τρία βασικά εργαλεία:

- **Wipe Tool:** Το συγκεκριμένο εργαλείο διαγράφει όλα τα δεδομένα της συσκευής, δηλαδή εφαρμογές, αρχεία, ρυθμίσεις, και επαναφέρει την συσκευή στην εργοστασιακή της κατάσταση. Στο σχήμα 5.1 φαίνεται ο κώδικας του εργαλείου, το οποίο εκτελεί τις εξής λειτουργίες:
 - Διαβάζει τις ταυτότητες των συσκευών από ένα αρχείο, το οποίο έχει δημιουργηθεί από τον διακομιστή ιστού, και τις φυλάει σε ένα πίνακα.
 - Για κάθε συσκευή διαγράφει τα δεδομένα τις και γίνεται επανεκκίνηση με τις εντολές `adb -s [deviceID] shell wipe data` και `adb -s [deviceID] reboot` αντίστοιχα.
 - Αναμένει μέχρι οι συσκευές να ξανά εκκινήσουν κανονικά.
 - Εκτελεί το `monkeyrunner` πρόγραμμα `postwipe.py`, μέσω του οποίου θα συμπληρωθεί αυτόματα η διαδικασία μετά την διαγραφή δεδομένων.
- **Backup Tool:** Το εργαλείο αυτό δημιουργεί ένα αρχείο, το οποίο φυλάσσει την κατάσταση της συσκευής, δηλαδή τις εφαρμογές, τις ρυθμίσεις και τα αρχεία που

έχει η συσκευή την συγκεκριμένη στιγμή. Στο σχήμα 5.2 φαίνεται ο κώδικας του εργαλείου, το οποίο εκτελεί τις εξής λειτουργίες:

- Διαβάζει τις ταυτότητες των συσκευών από ένα αρχείο, το οποίο έχει δημιουργηθεί από τον διακομιστή ιστού, και τις φυλάει σε ένα πίνακα.
 - Για κάθε συσκευή δημιουργεί αρχείο επαναφοράς δεδομένων, με όνομα την ταυτότητα της κάθε συσκευής μέσω της εντολής `adb -s [deviceId] backup -apk -shared -all -f [filename]`.
 - Εκτελεί το monkeyrunner πρόγραμμα `backup.py`, μέσω του οποίου θα συμπληρωθεί αυτόματα η διαδικασία μετά την δημιουργία των αρχείων επαναφοράς δεδομένων.
- **Restore Tool:** Το εργαλείο αυτό επαναφέρει στην συσκευή μέσω ενός αρχείου, μια κατάσταση στην συσκευή. Το αρχείο αυτό μπορεί να δημιουργηθεί μέσω του Backup Tool. Στο σχήμα 5.3 φαίνεται ο κώδικας του εργαλείου, το οποίο εκτελεί τις εξής λειτουργίες:
 - Διαβάζει τις ταυτότητες των συσκευών από ένα αρχείο, το οποίο έχει δημιουργηθεί από τον διακομιστή ιστού, και τις φυλάει σε ένα πίνακα.
 - Για κάθε συσκευή επαναφέρει από το αρχείο επαναφοράς δεδομένων, με όνομα την πρώτη παράμετρο στο εργαλείο, μέσω της εντολής `adb -s [deviceId] restore [filename]`.
 - Εκτελεί το monkeyrunner πρόγραμμα `restore.py`, μέσω του οποίου θα συμπληρωθεί αυτόματα η διαδικασία μετά την επαναφορά του αρχείου επαναφοράς δεδομένων.

Για την χρήση των πιο πάνω εργαλείων, ο χρήστης αρχικά πρέπει να συνδεθεί στο σύστημα με τον λογαριασμό, τον οποίο είναι διαχειριστής στον συγκεκριμένο εξυπηρετητή. Στην συνέχεια μια διεπαφή με όλες τις συσκευές εμφανίζεται στον διαχειριστή (Σχήμα 5.4), μέσω της οποίας μπορεί να επιλέξει τις συσκευές, και να εκτελέσει το ανάλογο εργαλείο. Κατά την εκτέλεση του ανάλογου εργαλείου, γίνεται μια AJAX αίτηση στον εξυπηρετητή, μέσω της οποίας το ανάλογο πρόγραμμα κελύφους

Status	Model	Screen	CPU	CPU (MHz)	Ram (MB)	SD Card (GB)	Sensors	Platform	Serial	Photo
ONLINE	HTC Desire S (S510e)	3.70" (480x800)	Qualcomm Snapdragon MSM8255	1024	768	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	2.2	HT13YTJ26874	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0ANPL06499	
ONLINE	Google Nexus S	4.00" (480x800)	Samsung Exynos 9110	1024	1024	0	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	2.3	37320124DA8A00EC	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0APPL01895	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0APPL01511	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0APPL01201	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0APPL00803	
ONLINE	HTC Desire (A8181)	3.70" (480x800)	Qualcomm Snapdragon QSD8250	1024	512	4	Mic, 3x Gyroscope, G-Sensor, Magnetometer, Proximity, Ambient light	4.0	SH0APPL01700	
ONLINE	HTC Hero (A6262)	3.20" (320x480)	Qualcomm Snapdragon MSM7600A	512	288	2	Mic, Gyro sensor, G-Sensor, Magnetometer, Proximity, Ambient light sensor	2.2	HT99TL902572	

Σχήμα 5.4: Διεπαφή εργαλειοθήκης εργαλείων διαχείρισης

εκτελείται και στην συνέχεια το ανάλογο Monkeyrunner πρόγραμμα, έτσι ώστε ο οδηγός για την συγκεκριμένη λειτουργία, να εκτελεστεί αυτόματα από το σύστημα.

Λόγο του ότι το εργαλείο Monkeyrunner, δεν επιτρέπει πολλά προγράμματα που χρησιμοποιούν το εργαλείο αυτό, να τρέχουν παράλληλα, σε κάθε εκτέλεση φυλάσσονται σε ένα προσωρινό αρχείο οι μοναδικές ταυτότητες των συσκευών. Στην συνέχεια τα προγράμματα κελύφους διαβάζουν τις ταυτότητες των συσκευών από το αρχείο αυτό και περνά στο ανάλογο Monkeyrunner πρόγραμμα ως παράμετρο τις συσκευές. Επομένως με αυτό τον τρόπο επιτυγχάνεται εκτέλεση σε πολλές συσκευές, με τον περιορισμό ότι δεν τρέχουν εντελώς παράλληλα, άλλα με διοχέτευση.

Κεφάλαιο 6

Γραφικό Περιβάλλον Προγραμματισμού

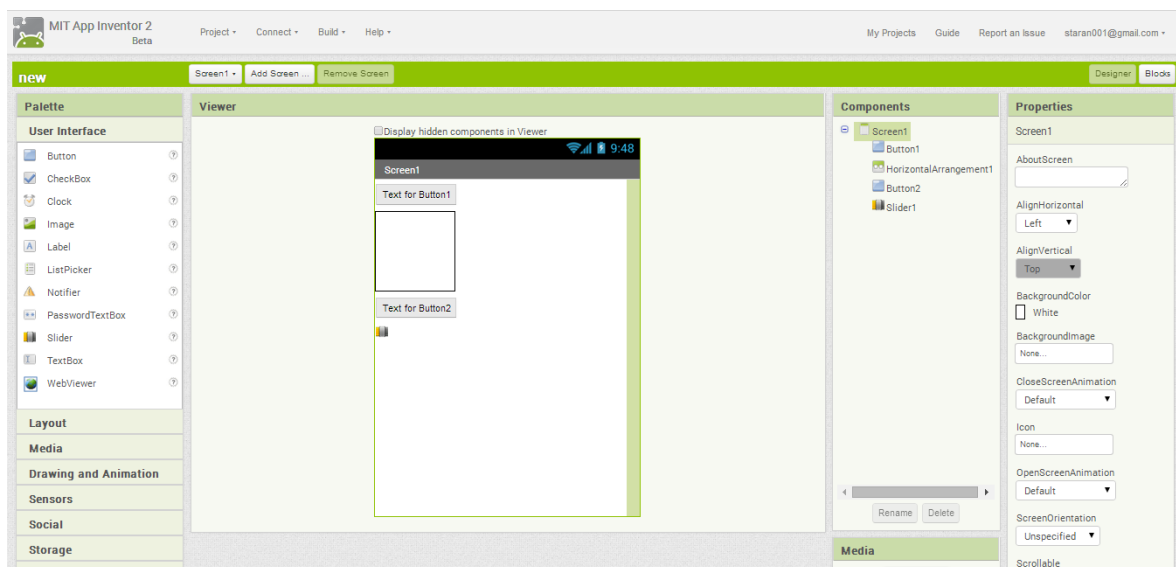
6.1	Εισαγωγή	43
6.2	Ανασκόπηση Εργαλείου AppInventor	43
6.3	Επέκταση Εργαλείου AppInventor	45

6.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά θα επεξηγηθεί το εργαλείο AppInventor καθώς και η χρησιμότητα του εργαλείου αυτού γενικά, αλλά και πιο συγκεκριμένα τον ρόλο του στο σύστημα SmartLab. Στη συνέχεια θα παρουσιαστεί ο τρόπος επέκτασης του εργαλείου αυτού, έτσι ώστε να ανταποκρίνεται στις λειτουργίες που χρειάζονται από το σύστημα και η ενσωμάτωση του σε αυτό.

6.2 Ανασκόπηση Εργαλείου AppInventor

Το εργαλείο AppInventor, είναι μια εφαρμογή ιστού, ανοικτού κώδικα, η οποία αρχικά ήταν διαθέσιμη από την Google, ενώ στη συνέχεια συντηρήθηκε από το MIT. Το εργαλείο αυτό δίνει την δυνατότητα σε ανθρώπους που δεν γνωρίζουν από προγραμματισμό να

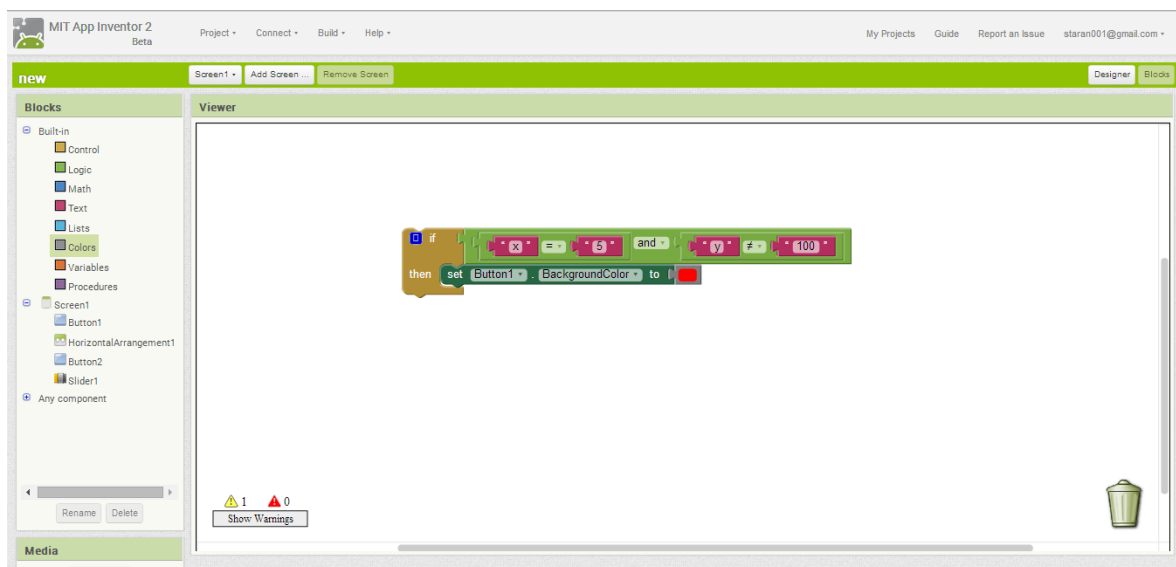


Σχήμα 6.1 : Γραφικό περιβάλλον επεξεργαστή διεπαφής AppInventor

μπορούν να δημιουργούν εφαρμογές Android. Η διαδικασία αυτή γίνεται μέσω γραφικού περιβάλλοντος το οποίο παρέχει στον χρήστη τις πιο κάτω δυνατότητες:

- Να διαμορφώσει την διεπαφή χρήστη της εφαρμογής, μέσω αντικειμένων που μπορεί να σύρει στην οθόνη εργασίας, όπως για παράδειγμα κουμπιά, πλαίσια κειμένου, διατάξεις οθόνης και πολλά άλλα (Σχήμα 6.1).
- Η λογική του προγράμματος γίνεται με παρόμοιο τρόπο, αφού παρέχονται μπλοκ, τα οποία προσομοιώνουν λειτουργίες λογικής ενός προγράμματος, όπως για παράδειγμα εντολές if, μαθηματικές πράξεις και άλλα εργαλεία λογικής ενός προγράμματος (Σχήμα 6.2).
- Για την μεταγλώττιση των προγραμμάτων αυτών χρησιμοποιείται μεταγλωττιστής με βάση την γλώσσα Scheme, έτσι ώστε να δημιουργηθεί το αρχείο apk για την εγκατάσταση της εφαρμογής στις συσκευές.

Το εργαλείο AppInventor έχει σημαντικό ρόλο στο σύστημα SmartLab, αφού πλατφόρμα μπορεί με αυτό τον τρόπο να βοηθήσει ανθρώπους οι οποίοι ενδιαφέρονται να μάθουν να αναπτύσσουν λογισμικό για συσκευές Android, να δημιουργήσουν εύκολα και γρήγορα εφαρμογές μέσω του συστήματος.

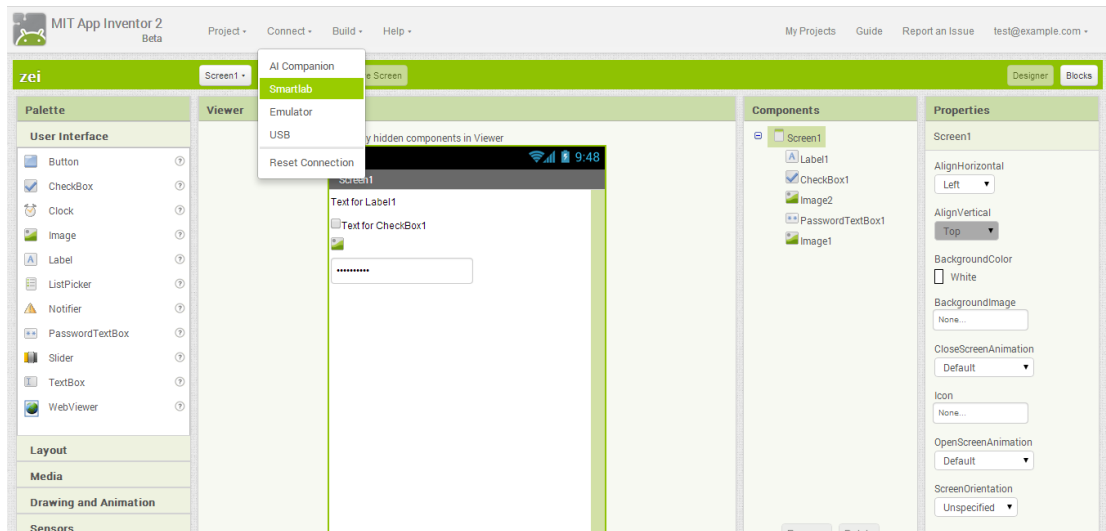


Σχήμα 6.2 : Γραφικό περιβάλλον επεξεργαστή λογικής AppInventor

6.3 Επέκταση Εργαλείου AppInventor

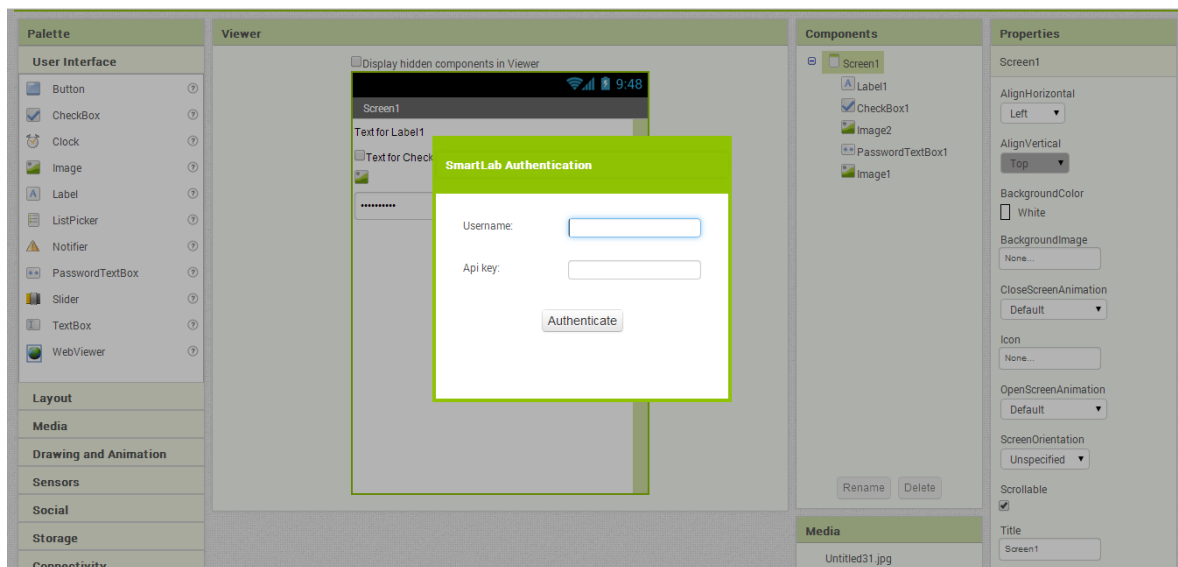
Το εργαλείο AppInventor, λόγω του ότι είναι ανοικτού κώδικα, μπορεί οποιοσδήποτε να κάνει αλλαγές ή να προσθέσει λειτουργίες. Η επέκταση που έγινε στο εργαλείο αυτό, είναι να μπορούν εφαρμογές που δημιουργούνται από το εργαλείο, κατευθείαν να εγκαθίστανται στις συσκευές που έχει κρατημένες ο χρήστης. Ο τρόπος με τον οποίο προστέθηκε η λειτουργία αυτή είναι μέσω των πιο κάτω βημάτων:

- Το εργαλείο με τις επιπρόσθετες λειτουργίες βρίσκεται πλέον ως μέρος της ιστοσελίδας του συστήματος SmartLab, έτσι μπορούν οι χρήστες του συστήματος να χρησιμοποιούν το εργαλείο μέσω του συστήματος.
- Στην διεπαφή του εργαλείου έχει προστεθεί μια νέα λειτουργία, δηλαδή κάτω από την επιλογή “Connect”, έχει προστεθεί μια νέα επιλογή “SmartLab” (Σχήμα 6.3), μέσω της οποίας ο χρήστης μεταφέρει το ark αρχείο στις συσκευές που έχει κρατημένες, αφού πρώτα συνδεθεί με το όνομα του και το api κλειδί που του παρέχεται από το σύστημα (Σχήμα 6.4).



Σχήμα 6.3 : Νέα λειτουργία εργαλείου AppInventor ώστε να συνδέεται με το σύστημα

- Η μεταφορά του apk αρχείου στις συσκευές γίνεται μέσω μιας HTTP αίτησης, η οποία στέλνει στον εξυπηρετητή του SmartLab τον σύνδεσμο με το apk αρχείο, καθώς και τα στοιχεία του χρήστη, έτσι ώστε να γίνει έλεγχος της ταυτότητας του εάν είναι έγκυρος ή όχι.



Σχήμα 6.4 : Παράθυρο καταχώρησης ονόματος χρήστη και API κλειδιού για την επικύρωση ταυτότητας του χρήστη

Κεφάλαιο 7

Πειραματικά Αποτελέσματα

7.1	Εισαγωγή	48
7.2	Πειραματική μεθοδολογία	49
7.3	Πειραματικά αποτελέσματα για χρόνο απόκρισης	49
7.4	Πειραματικά αποτελέσματα για χρήση επεξεργαστή	53

7.1 Εισαγωγή

Σαυτό το κεφάλαιο θα παρουσιαστούν τα πειράματα που έγιναν για κάθε υποσύστημα του συστήματος SmartLab συγκρίνοντας τις δύο υλοποιήσεις, με ADB και SPB. Αρχικά θα παρουσιαστεί η πειραματική μεθοδολογία που χρησιμοποιήθηκε για την αποτίμηση των πειραμάτων, στη συνέχεια πειραματικά αποτελέσματα σύγκρισης των δύο υλοποιήσεων βάση του χρόνου απόκρισης για κάθε υποσύστημα και τέλος πειραματικά αποτελέσματα βάση της χρήσης του επεξεργαστή.

Τα πειράματα που ακολουθούν έγιναν έτσι ώστε να καταλάβουμε αν η καινούργια αρχιτεκτονική χωρίς την χρήση του ADB, έχει καλύτερα αποτελέσματα από πλευράς χρόνου απόκρισης, καθώς και από πλευράς χρήσης του επεξεργαστή.

7.2 Πειραματική μεθοδολογία

Η πειραματική μελέτη έγινε για μία, δύο και τέσσερις συσκευές τύπου HTC Desire.

Ο λόγος που η πειραματική μελέτη έγινε σε διαφορετικό αριθμό συσκευών, είναι να δούμε πειραματικά εάν η νέα υλοποίηση είναι και πειραματικά καλύτερη από την παλιά, όπως αναμενόταν.

Κάθε πείραμα έχει εκτελεστεί για πέντε λεπτά, και έχει επαναληφθεί πέντε φορές, έτσι ώστε να έχουμε πιο ακριβή αποτελέσματα.

Για την συλλογή μετρήσεων στα πιο κάτω πειράματα έχει χρησιμοποιηθεί το εργαλείο htop, έτσι ώστε να μετρηθεί η χρήση του επεξεργαστή από τον android πελάτη, καθώς και αρχεία καταγραφής του συστήματος, έτσι ώστε να συλλεχθούν οι χρόνοι απόκρισης για κάθε πείραμα.

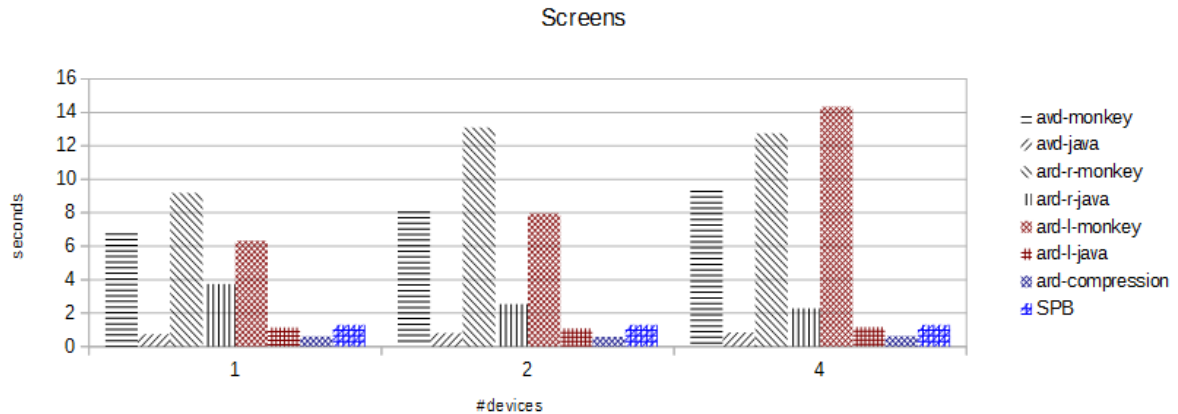
Λόγω του ότι ο χρόνος απόκρισης του υποσυστήματος Remote Shells είναι μηδαμινός και για τις δύο υλοποιήσεις παραλείπεται.

Επίσης στα πειραματικά αποτελέσματα για την χρήση του επεξεργαστή παρουσιάζονται μόνο τα δύο υποσυστήματα τα οποία η χρήση του επεξεργαστή που έχουν δεν είναι μηδαμινή.

7.3 Πειραματικά αποτελέσματα για χρόνο απόκρισης

Remote Control Terminals

Στο πιο πείραμα θα μελετηθεί ο χρόνος αποστολής οθονών από τις συσκευές προς τον εξυπηρετητή. Σκοπός του πιο κάτω πειράματος είναι η μελέτη των προσεγγίσεων, δηλαδή οι υλοποιήσεις με βάση το εργαλείο ADB σε σύγκριση με την υλοποίηση με βάση το SPB. Το πιο κάτω πείραμα έχει εκτελεστεί για μία συσκευή, δύο και τέσσερις, έτσι ώστε να μελετήσουμε την συμπεριφορά των δύο αρχιτεκτονικών για μεταβλητό αριθμό συσκευών. Κάθε πείραμα έχει εκτελεστεί για πέντε λεπτά, πέντε φορές, έτσι ώστε να έχουμε καλύτερα αποτελέσματα.

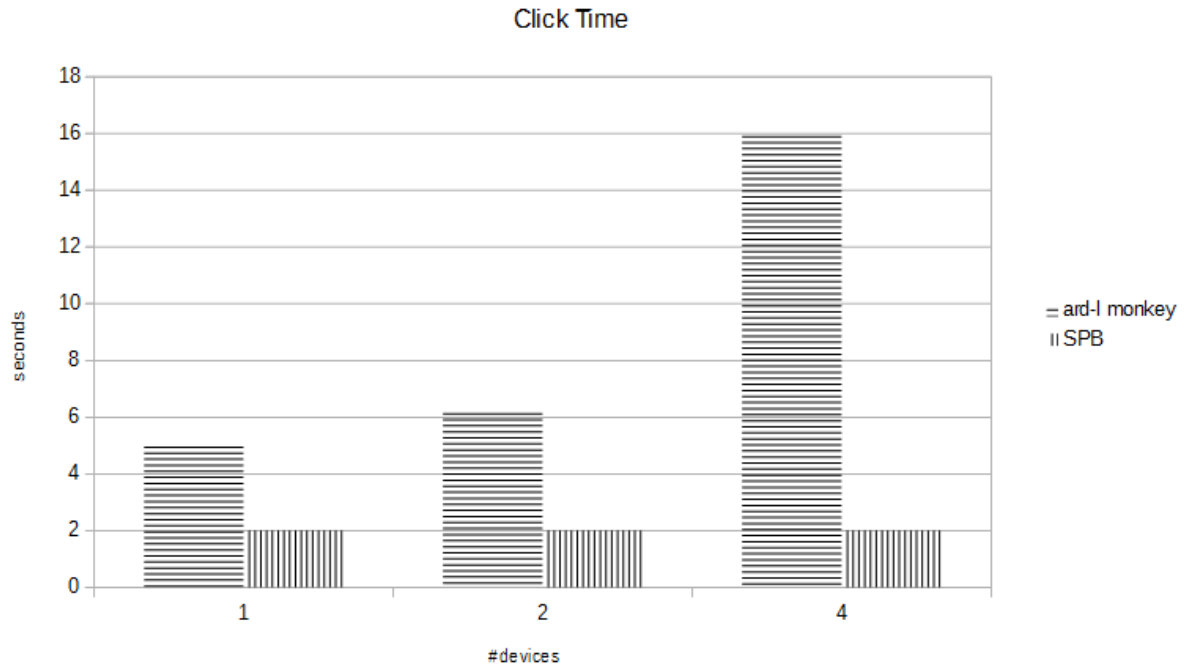


Σχήμα 7.1: Σύγκριση διάφορων υλοποιήσεων για το υποσύστημα Remote Control Terminals (RCT)

Στην πιο πάνω γραφική παράσταση (Σχήμα 7.1) μπορούμε να δούμε τον χρόνο σε δευτερόλεπτα της κάθε προηγούμενης υλοποίησης για το συγκεκριμένο υποσύστημα σε σύγκριση με το νέο (SPB). Ο χρόνος που παρουσιάζεται στο πιο πάνω σχήμα είναι για την αποστολή μίας οθόνης από τις συσκευές στον εξυπηρετητή. Στο σχήμα παρατηρούμε πως ο χρόνος για αποστολή μιας οθόνης αυξάνεται για τις περισσότερες υλοποιήσεις όσο αυξάνεται ο αριθμός των συσκευών αντιθέτως με την νέα. Επίσης ο χρόνος αποστολής είναι πολύ καλύτερος σε σύγκριση με την πλειοψηφία των υλοποιήσεων. Λόγο του ότι το εργαλείο ADB, επικοινωνεί με τις συσκευές μέσω καλωδίων USB, αναμένεται πως οι χρόνοι απόκρισης θα είναι καλύτεροι από την υλοποίηση SPB, η οποία χρησιμοποιεί το δίκτυο. Ο λόγος που η νέα υλοποίηση έχει καλύτερους χρόνους από την πλειοψηφία των άλλων υλοποιήσεων, είναι πως η κάθε οθόνη συμπιέζεται πριν σταλεί στις συσκευές, επομένως η χρήση του δικτύου είναι πολύ μικρότερη. Η σύγκριση η οποία μας ενδιαφέρει περισσότερο στο συγκεκριμένο πείραμα είναι μεταξύ της υλοποίησης ard-compression με SPB. Η υλοποίηση SPB έχει ελάχιστα χειρότερο χρόνο απόκρισης από την υλοποίηση ard-compression, όμως η χρησιμοποίησή του επεξεργαστή στη συγκεκριμένη υλοποίηση είναι πολύ χειρότερη όπως θα παρουσιαστεί σε πιο κάτω πείραμα.

Clicks

Στο πιο πείραμα θα μελετηθεί ο χρόνος αποστολής και εκτέλεσης ενός κλικ από τον εξυπηρετητή προς τις συσκευές. Σκοπός του πιο κάτω πειράματος είναι η μελέτη των δύο προσεγγίσεων, δηλαδή η υλοποίηση με βάση το εργαλείο ADB σε σύγκριση με την υλοποίηση με βάση το SPB. Το πιο κάτω πείραμα έχει εκτελεστεί για μία συσκευή, δύο και

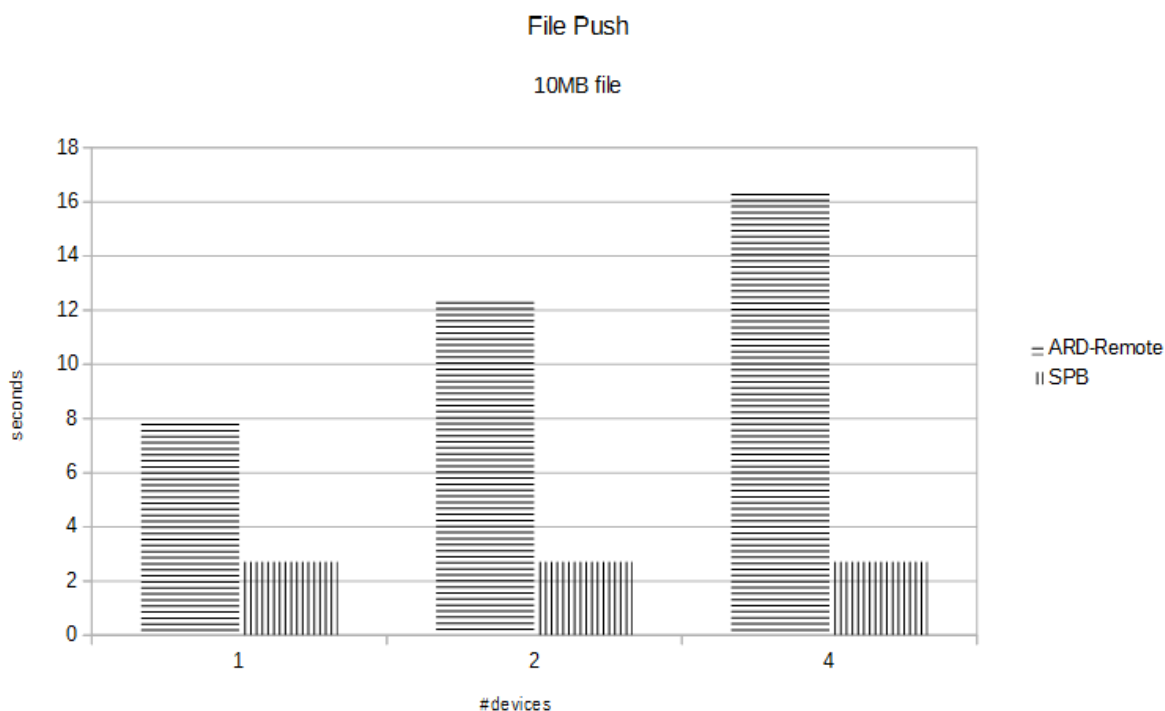


Σχήμα 7.2: Σύγκριση δύο υλοποιήσεων για το υποσύστημα των Click

τέσσερις, έτσι ώστε να μελετήσουμε την συμπεριφορά των δύο αρχιτεκτονικών για μεταβλητό αριθμό συσκευών. Κάθε πείραμα έχει εκτελεστεί για πέντε λεπτά, πέντε φορές, έτσι ώστε να έχουμε καλύτερα αποτελέσματα. Το κλικ που εκτελείτε στο πιο κάτω πείραμα είναι το power κλικ, το οποίο συμπεριφέρεται με τον ίδιο τρόπο όπως τα υπόλοιπα κλικ, επομένως δεν υπάρχει διαφορά στα αποτελέσματα των πειραμάτων όποιο είδος κλικ κι αν επιλέξουμε.

Στην πιο πάνω γραφική παράσταση (Σχήμα 7.2) μπορούμε να δούμε τον χρόνο σε δευτερόλεπτα της προηγούμενης υλοποίησης για το συγκεκριμένο υποσύστημα σε

σύγκριση με το νέο (SPB). Ο χρόνος που παρουσιάζεται στο πιο πάνω σχήμα είναι για την αποστολή ενός κλικ από τον εξυπηρετητή στις συσκευές. Στο σχήμα παρατηρούμε πως ο χρόνος για αποστολή ενός κλικ αυξάνεται στην παλιά υλοποίηση όσο αυξάνεται ο αριθμός των συσκευών αντιθέτως με την νέα που παραμένει σταθερός. Επίσης ο χρόνος αποστολής είναι πολύ καλύτερος σε σύγκριση με την παλαιότερη υλοποίηση.



Σχήμα 7.3: Σύγκριση δύο υλοποιήσεων για το υποσύστημα αρχείων

File push/pull

Στο πιο πείραμα θα μελετηθεί ο χρόνος αποστολής ενός αρχείου μεγέθους 10MB από τον εξυπηρετητή προς τις συσκευές. Σκοπός του πιο κάτω πειράματος είναι η μελέτη των δύο προσεγγίσεων, δηλαδή η υλοποίηση με βάση το εργαλείο ADB σε σύγκριση με την υλοποίηση με βάση το SPB. Το πιο κάτω πείραμα έχει εκτελεστεί για μία συσκευή, δύο και τέσσερις, έτσι ώστε να μελετήσουμε την συμπεριφορά των δύο αρχιτεκτονικών για

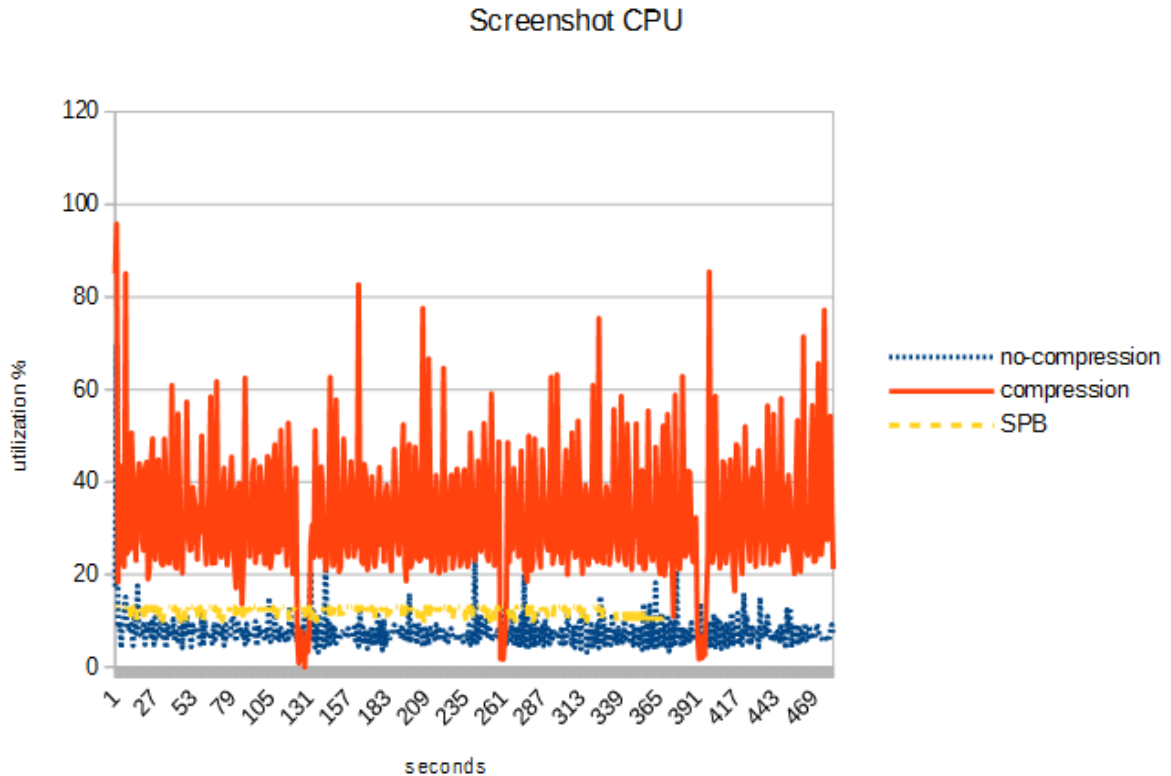
μεταβλητό αριθμό συσκευών. Κάθε πείραμα έχει εκτελεστεί για πέντε λεπτά, πέντε φορές, έτσι ώστε να έχουμε καλύτερα αποτελέσματα.

Στην πιο πάνω γραφική παράσταση (Σχήμα 7.3) μπορούμε να δούμε τον χρόνο σε δευτερόλεπτα της προηγούμενης υλοποίησης για το συγκεκριμένο υποσύστημα σε σύγκριση με το νέο (SPB). Στο σχήμα παρατηρούμε πως ο χρόνος για αποστολή ενός αρχείου προς τις συσκευές αυξάνεται όσο αυξάνεται ο αριθμός των συσκευών στην υλοποίηση με βάση το εργαλείο ADB, αντιθέτως με την νέα που παραμένει σταθερός ανεξαρτήτως του αριθμού των συσκευών. Παρατηρούμε επίσης πως ο χρόνος αποστολής του αρχείου είναι πολύ μικρότερος στην υλοποίηση με βάση το SPB παρά στην παλαιότερη υλοποίηση.

7.4 Πειραματικά αποτελέσματα για χρήση επεξεργαστή

Remote Control Terminals

Στο πιο πείραμα θα μελετηθεί η χρήση του επεξεργαστή για το υποσύστημα οθονών. Σκοπός του πιο κάτω πειράματος είναι η μελέτη τριών προσεγγίσεων, δηλαδή η υλοποίηση με βάση το εργαλείο ADB χωρίς συμπίεση σε σύγκριση με την υλοποίηση με βάση το SPB το οποίο περιέχει συμπίεση, καθώς και υλοποίηση με βάση το εργαλείο ADB με συμπίεση. Κάθε πείραμα έχει εκτελεστεί για 100 οθόνες, πέντε φορές, έτσι ώστε να έχουμε καλύτερα αποτελέσματα.

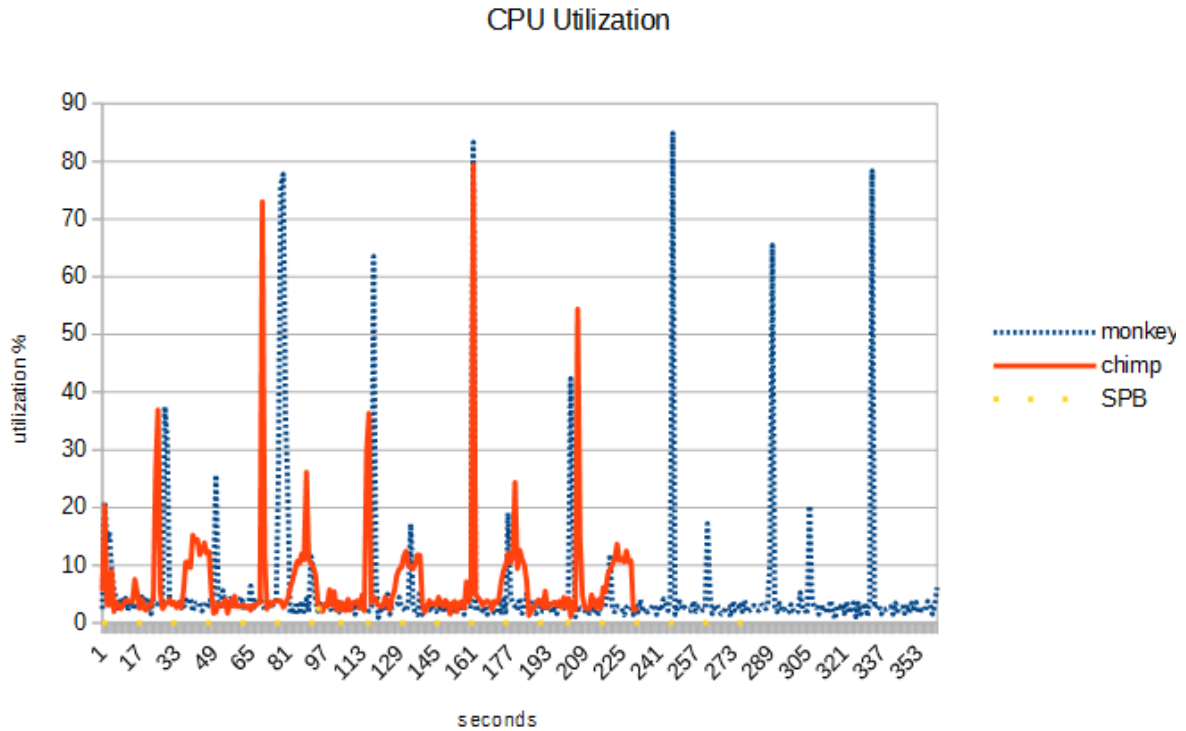


Σχήμα 7.4: Σύγκριση τριών υλοποιήσεων για το υποσύστημα οθονών (χρήση επεξεργαστή)

Στην πιο πάνω γραφική παράσταση (Σχήμα 7.4) μπορούμε να δούμε το ποσοστό χρήσης του επεξεργαστή των προηγούμενων δύο υλοποιήσεων για το συγκεκριμένο υποσύστημα σε σύγκριση με το νέο (SPB). Στο σχήμα παρατηρούμε πως το ποσοστό χρήσης για την υλοποίηση με συμπίεση είναι πολύ περισσότερος από τις άλλες δύο υλοποιήσεις. Η νέα υλοποίηση (SPB), έχει μεγαλύτερη χρήση του επεξεργαστή, εφόσον και σε αυτή γίνεται μερική συμπίεση, όμως πολύ καλύτερη από την υλοποίηση με συμπίεση που χρησιμοποιεί το ADB.

Clicks

Στο πιο πείραμα θα μελετηθεί η χρήση του επεξεργαστή για το υποσύστημα των κλικ. Σκοπός του πιο κάτω πειράματος είναι η μελέτη τριών προσεγγίσεων, δηλαδή οι δύο



Σχήμα 7.5: Σύγκριση τριών υλοποιήσεων για το υποσύστημα των Click (χρήση επεξεργαστή)

υλοποιήσεις με βάση το εργαλείο ADB σε σύγκριση με την υλοποίηση με βάση το SPB το οποίο χρησιμοποιεί τοπικό εξυπηρετητή στην συσκευή. Κάθε πείραμα έχει εκτελεστεί για 100 κλικ, πέντε φορές, έτσι ώστε να έχουμε καλύτερα αποτελέσματα.

Στην πιο πάνω γραφική παράσταση (Σχήμα 7.5) μπορούμε να δούμε το ποσοστό χρήσης του επεξεργαστή των προηγούμενων δύο υλοποιήσεων για το συγκεκριμένο υποσύστημα σε σύγκριση με το νέο (SPB). Στο σχήμα παρατηρούμε πως το ποσοστό χρήσης για την νέα υλοποίηση είναι μηδαμινός.

Κεφάλαιο 8

Συμπεράσματα και μελλοντικές επεκτάσεις

8.1	Συμπεράσματα	56
8.2	Μελλοντικές επεκτάσεις	57

8.1 Συμπεράσματα

Σε αυτή την διπλωματική εργασία, σκοπός ήταν η επέκταση του συστήματος SmartLab έτσι ώστε να επεκταθούν ορισμένα από τα ήδη υπάρχον υποσυστήματα, να υλοποιηθούν νέα υποσυστήματα, καθώς και να μετατραπούν έτσι ώστε το σύστημα να μπορεί να επεκταθεί σε μεγαλύτερη κλίμακα. Ο στόχος της εργασίας αυτής έχει επιτευχθεί με τις πιο κάτω υλοποιήσεις:

- Υλοποίηση Remote shells (ADB)
- Επέκταση εργαλείου AppInventor
- Υλοποίηση εργαλείων διαχείρισης
- Σχεδιασμός και υλοποίηση αρχιτεκτονικής SPB

Τα πιο πάνω υποσυστήματα και εργαλεία έχουν προσφέρει στο σύστημα την δυνατότητα να προσφέρει στους χρήστες, μέσω μιας εύκολης και διαδραστικής διεπαφής, νέες λειτουργίες, καθώς και εύκολη διαχείριση των συσκευών από τον διαχειριστή. Η πιο βασική μελέτη που έχει γίνει στην εργασία αυτή, είναι ο σχεδιασμός και η υλοποίηση του πρωτοκόλλου SPB, μέσω του οποίου το σύστημα ξεπερνά κάθε περιορισμό που προκαλεί το εργαλείο ADB. Μέσα από την πειραματική μελέτη που έγινε στην εργασία αυτή, μέσα

από την οποία έγινε σύγκριση των δύο αρχιτεκτονικών του συστήματος (αρχιτεκτονική ADB με αρχιτεκτονική SPB), τα αποτελέσματα έδειξαν πως η νέα αρχιτεκτονική (SPB), έχει καλύτερους χρόνους απόκρισης καθώς και καλύτερη χρησιμοποίηση του επεξεργαστή για την πλειοψηφία των υποσυστημάτων. Επίσης εκτός από πλευράς απόδοσης, η αρχιτεκτονική SPB προσφέρει στο σύστημα την δυνατότητα επέκτασης του σε αστική κλίμακα, αφού πλέον οι συσκευές δεν έχουν προβλήματα ασφαλείας σε εξωτερικά δίκτυα.

8.2 Μελλοντικές επεκτάσεις

Τα υποσυστήματα που υλοποιήθηκαν, καθώς και η αρχιτεκτονική SPB θα μπορούσαν να επεκταθούν με τους πιο κάτω τρόπους:

- Λόγο της γρήγορης ανάπτυξης σε τεχνολογίες των έξυπνων συσκευών, συνεχώς παρέχονται νέοι τρόποι για την υλοποίηση των υποσυστημάτων. Για τον λόγο αυτό μια καλή επέκταση θα ήταν η παρακολούθηση και η μελέτη των νέων τρόπων λειτουργίας των υποσυστημάτων αυτών και η πειραματική μελέτη από θέματα απόδοσης.
- Η επέκταση του πρωτοκόλλου SPB έτσι ώστε να μπορεί το σύστημα να παρέχει εξυπηρετητές εκτός της δικής του υποδομής, μεγαλώνοντας έτσι όχι μόνο των αριθμό συσκευών που παρέχονται, αλλά και των αριθμό χρηστών του.
- Η επέκταση του πελάτη σε Android, έτσι ώστε να παρέχεται με ασφάλεια η δυνατότητα σε οποιοδήποτε χρήστη που είναι κάτοχος έξυπνης συσκευής, να μπορεί να προσφέρει την δική του συσκευή στο σύστημα, χωρίς όμως να γίνεται εισβολή στην ιδιωτικότητα του χρήστη. Με τον τρόπο αυτό θα παρέχονται συσκευές οι οποίες βρίσκονται οπουδήποτε.

Βιβλιογραφία

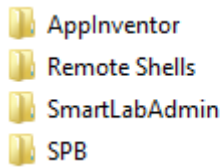
- [1] A. Konstantinidis, C. Costa, G. Larkou, and D. Zeinalipour-Yazti, “Demo: a programming cloud of smartphones”, In ACM Mobisys’12.
- [2] H. Kim, N. Agrawal, and C. Ungureanu. “Revisiting storage for smartphones,” In USENIX FAST’12. (Best Paper Award).
- [3] D. Zeinalipour-Yazti, C. Laoudias, C. Costa, M. Vlachos, M. I. Andreou, and D. Gunopulos, “Crowdsourced Trajectory Similarity with Smartphones”, In IEEE Trans. on Know. and Data Engr., Vol. 25, pp. 1240-1253, 2013.
- [4] C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti, and C. G. Panayiotou, “The airplace indoor positioning platform for android smartphones,” In IEEE MDM’12 (Best Demo Award).
- [5] L. Peterson et. al., “A Blueprint for Introducing Disruptive Technology into the Internet,” In ACM HotNets’02.
- [6] D. Johnson, T. Stack, R. Fish, D.M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau “MobileEmulab: A Robotic Wireless and Sensor Network Testbed”, In IEEE INFOCOM’06.
- [7] G. Werner-Allen, P. Swieskowski, M. Welsh, “MoteLab: a wireless sensor network testbed,” In ACM/IEEE IPSN’05.
- [8] J. Cohen, T. Repantis, S. McDermott, S. Smith, and J. Wein. “Keeping Track of 70,000+ Servers: The Akamai Query System”, In USENIX LISA’10.
- [9] M. Dehus, and D. Grunwald. “STORM: Simple Tool for Resource Management”, In USENIX LISA’08.
- [10] E. Sorenson, Explosive Networking, and Strata Rose Chalup, VirtualNet, “RedAlert: A Scalable System for Application Monitoring”, In USENIX LISA’99.
- [11] R. Narayana-Murty, G. Mainland, I. Rose, A.R. Chowdhury, A. Gosain, J. Bers, and M. Welsh. “CitySense: An Urban-Scale Wireless Sensor Network and Testbed,” In IEEE HST’08.
- [12] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, “Mobiscopes for Human Spaces”, In IEEE Per-Com’07.

- [13] A. Kansal, M. Goraczko, and F. Zhao. “Building a sensor network of mobile phones”, In ACM/IEEE IPSN’07.
- [14] Samsung, Remote Test Lab: <http://goo.gl/SW3t6>.
- [15] Perfecto Mobile, <http://goo.gl/DSlP9>.
- [16] Keynote Systems Inc., Device Anywhere: <http://goo.gl/mCxft>.
- [17] T. Verry, “MegaDroid simulates network of 300,000 Android smartphones”, <http://goo.gl/jMaS8>.
- [18] T. Das, P. Mohan, V.N. Padmanabhan, R. Ramjee, and A. Sharma, “PRISM: Platform for Remote Sensing using Smartphones”, In ACM MobiSys’10.
- [19] E. Cuervo, P. Gilbert, B. Wu, and L.P. Cox, “CrowdLab: An Architecture for Volunteer Mobile Testbeds”, In COMSNETS’11.
- [20] G. Challen, M. Demirbas, S.Y. Ko, and T. Kosar “PhoneLab: A Large-Scale Participatory Smartphone Testbed”, NSF PECS Workshop, Jan. 27-28, 2011.
- [21] G. Challen et. al. “PhoneLab: A Large-Scale Participatory Smartphone Testbed”, In USENIX NSDI’12 (poster).

Παράρτημα Α.

Πηγαίος Κώδικας

Ακολουθεί ο πηγαίος κώδικας των τεσσάρων συνεισφορών που βρίσκεται στο CD που συνοδεύει την παρούσα διπλωματική εργασία κάτω από τον φάκελο src που έχει την εξής μορφή:



Στο φάκελο:

- **AppInventor:** Βρίσκεται η υλοποίηση της επέκτασης του εργαλείου AppInventor. Τα αρχεία που περιλαμβάνονται είναι μόνο αυτά τα οποία έχουν αλλαχθεί έτσι ώστε να επιτευχθεί η ενσωμάτωση με το σύστημα.
- **Remote Shells:**
 - **Client side:** Βρίσκεται η υλοποίηση του πελάτη ιστού του υποσυστήματος Remote Shells (RS).
 - **Server side:** Βρίσκεται η υλοποίηση του εξυπηρετητή του υποσυστήματος Remote Shells (RS)
- **SmartLabAdmin:** Βρίσκεται η υλοποίηση της εργαλειοθήκης διαχείρισης συσκευών σε Django framework.
- **SPB:**
 - **ClickServer:** Βρίσκεται η υλοποίηση του εξυπηρετητή που είναι υπεύθυνος για τα κλικ στη συσκευή.
 - **SPBClient:** Βρίσκεται η υλοποίηση του πελάτη SPB σε Android ο οποίος είναι υπεύθυνος για την σύνδεση με τον εξυπηρετητή SPB.
 - **SPBServer:** Βρίσκεται η υλοποίηση του εξυπηρετητή SPB.