



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΕΓΚΑΤΑΣΤΑΣΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΗΣ

Υλοποίηση Εφαρμογής με τη χρησιμοποίηση της Ευρωπαϊκής Πλατφόρμας Ενδιάμεσου Λογισμικού webinos

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΡΩΜΑΝΟΥ - ΔΗΜΟΚΡΙΤΟΥ ΤΣΟΥΡΟΠΛΗ

Επιβλέπων : Δημήτριος Ασκούνης

Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2013



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΕΓΚΑΤΑΣΤΑΣΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΗΣ

Υλοποίηση Εφαρμογής με τη χρησιμοποίηση της Ευρωπαϊκής Πλατφόρμας Ενδιάμεσου Λογισμικού webinos

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΡΩΜΑΝΟΥ - ΔΗΜΟΚΡΙΤΟΥ ΤΣΟΥΡΟΠΛΗ

Επιβλέπων : Δημήτριος Ασκούνης

Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

2013.

.....
Δ. Ασκούνης
Επ. Καθηγητής Ε.Μ.Π

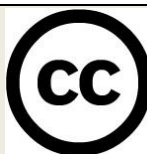
.....
Ι. Ψαρράς
Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2013

.....

Ρωμανός - Δημόκριτος Τσουροπλής

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.



Εργασία με Ανοιχτή Άδεια

Αυτή η εργασία χορηγείται με άδεια Creative Commons Αναφορά Δημιουργού 3.0 Ελλάδα. Για χρήση αυτής της εργασίας είναι απαραίτητη η αναφορά τόσο στον συγγραφέα (Τσουροπλής Ρωμανός - Δημόκριτος) όσο και στη Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Η συγκεκριμένη άδεια επιτρέπει την ελεύθερη χρήση αυτής της εργασίας για Διανομή (αναπαραγωγή, διανομή, παρουσίαση στο κοινό του έργου) και Διασκευή (για να τροποποιήσετε το έργο).

Για περισσότερες πληροφορίες σχετικά με την άδεια που συνοδεύει την παρούσα εργασία, παρακαλώ επισκεφθείτε τον ακόλουθο σύνδεσμο:
<http://creativecommons.org/licenses/by/3.0/gr/>

Ο συγγραφέας προτείνει και υποστηρίζει τη διάδοση και καθιέρωση χρήσης ανοιχτών αδειών.

ΠΡΟΛΟΓΟΣ

Η διπλωματική αυτή εργασία εκπονήθηκε στα πλαίσια των ερευνητικών δραστηριοτήτων του Τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων, της Σχολής Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, του Εθνικού Μετσόβιου Πολυτεχνείου, κατά το ακαδημαϊκό έτος 2012-2013.

Θα ήθελα να ευχαριστήσω τους Καθηγητές κ. Δημήτριο Ασκούνη και κ. Ιωάννη Ψαρρά για την ευκαιρία που μου δώσανε να ασχοληθώ ενεργά στο Εργαστήριο Συστημάτων Απόφασης και Διοίκησης και να μπορέσω να δω σε βάθος κάποια από τα πιο ενδιαφέροντα και καινοτόμα Ευρωπαϊκά Προγράμματα, ένα εκ των οποίων είναι και το webinos, πάνω στο οποίο βασίστηκε αυτή η διπλωματική εργασία.

Θα ήθελα επίσης να ευχαριστήσω θερμά τον υπεύθυνο της παρούσης διπλωματικής, υποψήφιο διδάκτορα κ. Χρήστο Ντάνο, τους ερευνητές κ. Χρήστο Μπότσικα, κ. Ανδρέα Μπότσικα και τον κ. Σπύρο Μουζακίτη για την πολύτιμη βοήθειά τους σε κάθε φάση της εκπόνησης αυτού του έργου καθώς και για την καθοδήγησή και τις συμβουλές τους που οδήγησαν στην ολοκλήρωση της εργασίας.

Τέλος, θα επιθυμούσα να αφιερώσω την παρούσα εργασία στην οικογένεια μου ως ελάχιστη ανταπόδοση για την ανεκτίμητη υποστήριξη που μου παρείχε καθ' όλη την ακαδημαϊκή μου πορεία.

ΠΕΡΙΛΗΨΗ

Ένα κοινό πρόβλημα της σύγχρονης εποχής είναι το γεγονός ότι οι πολλαπλές πλατφόρμες και τα διαφορετικά μοντέλα συσκευών με ανομοιογενή χαρακτηριστικά δεν ευνοούν την προτυποποίηση μίας μοναδικής μεθόδου ανάπτυξης λογισμικού η οποία να δύναται να αξιοποιηθεί με κοινό τρόπο από όλες αυτές τις συσκευές.

Η ανάγκη που απορρέει αφορά στον προγραμματισμό διαφορετικών τελικών εφαρμογών για την επίτευξη στόχευσης ευρύτερου κοινού. Διαφορετικές εμπειρίες χρηστών κατά την αλληλεπίδραση ανθρώπου – μηχανής για μία συγκεκριμένη εφαρμογή καθώς και προβλήματα διασύνδεσης των διαφόρων συσκευών είναι μερικές περαιτέρω ανάγκες.

Το webinos, ένα σύνολο επεκτάσεων του web runtime, ήρθε να συνδράμει στην επίλυση των παραπάνω αναγκών καθώς παρέχει τη δυνατότητα δημιουργίας εφαρμογών με συνέπεια, ασφάλεια, ενοποιημένα σε ένα σύνολο συσκευών.

Σκοπός της διπλωματικής αυτής εργασίας είναι η δημιουργία πρωτότυπης εφαρμογής χρησιμοποιώντας το πλαίσιο webinos με βάση το API το οποίο παρέχεται με τη μορφή του ανοιχτού κώδικα από τους δημιουργούς του. Για την εφαρμογή επιλέχθηκε η υλοποίηση ψυχαγωγικής μορφής παιχνιδιού τράπουλας, γνωστό ως roker, κατά το οποίο δημιουργείται ασφαλής διασύνδεση πολλαπλών διαφορετικών συσκευών μέσω της οποίας οι παίκτες μπορούν να ανταλλάσσουν πληροφορίες χωρίς καμμία απολύτως εξάρτηση από το υλικό το οποίο διαθέτουν.

Λέξεις – κλειδιά: webinos, πολλαπλές πλατφόρμες, αλληλεπίδραση ανθρώπου – μηχανής, πρωτότυπη εφαρμογή, ασφαλής διασύνδεση

Η σελίδα αυτή είναι σκόπιμα λευκή.

ABSTRACT

A common problem of the modern society is the fact that the multiple platforms and large variety of different device models with the plurality of various characteristics do not favor the prototyping of a single method to develop software that is capable of being utilized in a common way in a variety of devices.

As a result we have the need of programming different application programs so that we can achieve a larger audience target. Different end-user experiences during the human-machine interaction for a particular application, problems interconnecting the variety of devices are only some of the extra needs.

Webinos, a set of web runtime extensions, has come to give a solution to the above problems by giving the ability of creating applications with consistency, security, unified for the vast majority of devices.

The goal of this thesis is the creation of a prototype application by using the webinos framework based on the API provided in the form of open source code by its creators. The implementation of a recreational deck game, known as poker, has been chosen for this application in which a safe connectivity of multiple different devices is created. Through this connectivity the players can exchange information without having any dependency on the hardware they have.

Keywords: webinos, multiple platforms, human-machine interaction, prototype application, safe interconnection

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας Περιεχομένων

Σύνοψη Εικόνων	14
Σύνοψη Πινάκων	15
1 Εισαγωγή.....	16
1.1 Εισαγωγή στη λογική πλαισίου	17
1.2 Τεχνική Ορολογία	20
1.3 Οδηγός Επομένων Κεφαλαίων.....	22
2 Webinos	23
2.1 Εισαγωγή - Ιστορικά	24
2.2 Χαρακτηριστικά	26
2.3 Τρόπος Λειτουργίας	28
2.3.1 Προσωπική Ζώνη (Personal Zone).....	28
2.3.2 Κόμβος Προσωπικής Ζώνης (Personal Zone Hub – PZH)	29
2.3.3 Πληρεξούσιο Προσωπικής Ζώνης (Personal Zone Proxy - PZP).....	31
2.3.4 Τοπικές Συνδέσεις	34
2.3.5 Συγχρονισμός.....	35
2.3.6 Πολιτικές Συστήματος	36
2.3.7 Το webinos στη συσκευή.....	36
2.3.8 Υπηρεσίες του webinos	37
2.3.9 Σύνδεση, Προστασία της ιδιοκτησίας και Ασφάλεια.....	38
2.3.10 Επεκτασιμότητα	39
2.3.11 Εφαρμογές webinos	39
2.3.12 Καινοτομίες webinos.....	40
2.4 Τα APIs	43
3 Εφαρμογή Poker	47
3.1 Εισαγωγή - Αφορμή.....	48
3.2 Χαρακτηριστικά	51

3.3 SVG.....	63
3.4 Τα APIs που χρησιμοποιήθηκαν.....	66
3.5 Τρόπος Λειτουργίας με στοιχεία κώδικα	68
3.6 Παράδειγμα Ενδεικτικής Λειτουργίας	75
4 Επίλογος.....	80
4.1 Σύνοψη και Συμπεράσματα	81
4.2 Μελλοντικές επεκτάσεις	83
5 Αναφορές - Βιβλιογραφία	85
5.1 Αναφορές.....	86
5.2 Βιβλιογραφία.....	88

Σύνοψη Εικόνων

Εικόνα 1 – Προσωπική Ζώνη	29
Εικόνα 2 – Επικοινωνία ΡΖΗ - ΡΖΡ	32
Εικόνα 3 – Σύνδεση συσκευών ΡΖΡ μέσω TLS με το ΡΖΗ.....	33
Εικόνα 4 – Γενική εποπτική εικόνα της τεχνολογίας πλαισίου webinos	42
Εικόνα 5 - Use Case Scenario – Εγκατάσταση Εφαρμογής	52
Εικόνα 6 - Use Case Scenario - Παιχνίδι.....	53
Εικόνα 7 – UML Class Diagram	55
Εικόνα 8 – Αρχικοποίηση Εφαρμογής.....	75
Εικόνα 9 – Δημιουργία Τραπεζιού	76
Εικόνα 10 – Εισαγωγή 1 ^{ου} παίκτη.....	76
Εικόνα 11 – Εισαγωγή 2 ^{ου} παίκτη.....	77
Εικόνα 12 – Νέος γύρος παιχνιδιού	77
Εικόνα 13 – Άνοιγμα 3 πρώτων καρτών.....	78
Εικόνα 14 – Ολοκλήρωση παρτίδας.....	78
Εικόνα 15 – Κοινές πληροφορίες τραπεζιού στις συσκευές.....	79

Σύνοψη Πινάκων

Πίνακας 1 – Πιθανότητες συνδυασμών στο παιχνίδι Πόκερ Texas Hold'em	49
Πίνακας 2 – Classes - Device	56
Πίνακας 3 – Classes - Player	56
Πίνακας 4 – Classes - Table	57
Πίνακας 5 – Classes - NewCardImage	58
Πίνακας 6 – Classes - Game.....	59
Πίνακας 7 – Classes - WebinosConnector	60

1 Εισαγωγή

- 1.1** Εισαγωγή στη λογική πλαισίου
- 1.2** Τεχνική Ορολογία
- 1.3** Οδηγός Επομένων Κεφαλαίων

1.1 Εισαγωγή στη λογική πλαισίου

Οι άνθρωποι δημιούργησαν τις συνθήκες του κόσμου πάνω στον οποίο ζούμε σήμερα χρησιμοποιώντας την επικοινωνία ως κινητήριο δύναμη για την διαμόρφωση κοινωνιών. Με την διαμόρφωση των κοινωνιών αυτών καταφέραμε να θριαμβεύσουμε και να καινοτομήσουμε σε όλους τους τομείς με κύριο σκοπό την βελτίωση του βιοτικού μας επιπέδου.

Για να γίνει εφικτή αυτή η επικοινωνία μεταξύ των ανθρώπων χρειαζόμαστε ένα κοινό πλαίσιο. Αυτό μπορεί να είναι μία γλώσσα, μία κοινή αντίληψη του κόσμου, ένας κοινός τρόπος αντιμετώπισης συνθηκών και καταστάσεων και φυσικά συνδυασμός όλων των παραπάνω. Γνωρίζουμε ότι αλλιώς αντιλαμβάνεται την ίδια κατάσταση ένας άνθρωπος ο οποίος έχει κάποια βιώματα σε σχέση με έναν άλλον άνθρωπο με διαφορετικό υπόβαθρο. Για να μπορέσουν λοιπόν αυτοί οι άνθρωποι να επικοινωνήσουν, χρειάζονται κάποια κοινά ορίσματα και έναν κοινά αντιληπτό τρόπο να μοιραστούν τις σκέψεις και τις εμπειρίες τους.

Ως πλαισίωση μιας συζήτησης νοούνται φυσικά και κοινά στοιχεία ή γνωρίσματα τα οποία μπορεί να μην αποφέρουν άμεσα πληροφορία αλλά να συμβάλλουν έμμεσα σε μια παρόμοια νόηση του περιβάλλοντος με βάση την οποία να διαμορφώνεται ο χαρακτήρας και η κοσμοθεωρία ή κάτι άλλο σε σχέση ειδικά με την προκειμένη συζήτηση στους συμμετέχοντες.

Με βάση τα παραπάνω αντιλαμβανόμαστε ότι η χρήση ενός κοινού πλαισίου είναι καθοριστικός παράγοντας για την αξιοποίηση δεδομένων από διαφορετικούς ανθρώπους. Ενώ στις καθημερινές τους επαφές οι άνθρωποι έχουν αυτή την δυνατότητα σε έναν μεγάλο βαθμό, δεν μπορούμε να πούμε το ίδιο για τη διεπαφή ανθρώπου – μηχανής. Η επικοινωνία αυτή περιορίζεται σαφώς από έναν φτωχό τρόπο εισαγωγής δεδομένων από τους χρήστες με βάση πάντα ένα καθορισμένο πλαίσιο.

Τις τελευταίες δεκαετίες (από τα τέλη του 1980) λέγεται ότι ζούμε στην εποχή της πληροφορίας, γνωστή και ως εποχή των υπολογιστών, όπου οι άνθρωποι μεταφέρουν πληροφορίες ελεύθερα με μεγαλύτερο ποσοστό μετάδοσης και έχουν άμεση πρόσβαση σε γνώσεις οι οποίες θα ήταν δύσκολο έως αδύνατο να βρεθούν

παλιότερα. Το Διαδίκτυο είναι αυτό που υπήρξε ο μεγαλύτερος κοινωνός σε αυτό το γεγονός. Με βάση στοιχεία μελετών προκύπτει ότι σε λιγότερο από τέσσερα χρόνια η παγκόσμια κίνηση δεδομένων μέσω IP θα έχει τριπλασιαστεί ¹. Καθώς περνάνε τα χρόνια, οι συσκευές αναλογικά με τους ανθρώπους αυξάνονται ραγδαία και είμαστε μάρτυρες της μετάβασης από τα κατανεμημένα, στα κινητά σύγχρονα υπολογιστικά συστήματα. Τα στατιστικά στοιχεία της πρόσβασης στο Internet μέσω κινητών συσκευών είναι αποκαλυπτικά ². Οι κινητές συσκευές περιλαμβάνουν αλλά δεν περιορίζονται σε φορητούς υπολογιστές (laptops), υπολογιστές παλάμης (tablets), έξυπνα κινητά τηλέφωνα (smartphones) και φορητές συσκευές, όπως ρολόγια.

Τροχοπέδη στην ομαλή επικοινωνία μας με μία μηχανή είναι η τεράστια διαφοροποίηση στις συσκευές. Υπάρχει μάλιστα πληθώρα ξεχωριστών πλατφόρμων ειδικά διαμορφωμένων και εξειδικευμένων για την εκάστοτε συσκευή. Με βάση αυτό το πλαίσιο λοιπόν, η πληροφορία την οποία εισάγουμε σε μία συσκευή μπορεί να μην δύναται να προσπελαστεί από άλλη συσκευή και ακόμη πιο πιθανά να εξάγονται διαφορετικά αποτελέσματα κατά την επεξεργασία των ίδιων δεδομένων από διαφορετικού τύπου συσκευές. Όπως γίνεται αντιληπτό, η επικοινωνία μεταξύ των χρηστών διαφορετικών συσκευών δυσχεραίνεται και οδηγεί σε κακή και ανόμοια ερμηνεία των αποτελεσμάτων τις περισσότερες φορές.

Οι υπολογιστικές συσκευές δεν συνδέονται μόνο στο Internet, αλλά σχηματίζουν και τοπικά δίκτυα ή ασύρματα δίκτυα με κινητούς κόμβους. Οι κόμβοι αυτών των δικτύων αποτελούνται από όλων των ειδών τις συσκευές οι οποίες έχουν την δυνατότητα τέτοιας διασύνδεσης. Τέτοιες είναι οι κινητές συσκευές οι οποίες συνεχώς αλλάζουν θέση και περιβάλλον λειτουργίας. Οι συσκευές αυτές λοιπόν συνδέονται σε μία μεταβαλλόμενη δικτυακή υποδομή, στην οποία βρίσκονται κατανεμημένες υπηρεσίες, διαθέσιμες προς τους χρήστες.

Για να επιτύχουν οι συσκευές την αλληλεπίδραση με το φυσικό περιβάλλον και την προσαρμογή σε μεταβαλλόμενες συνθήκες, χρειάζεται να έχουν επίγνωση του περιβάλλοντος στο οποίο λειτουργούν και των παραγόντων που επηρεάζουν τη λειτουργία τους. Το σύνολο των παραγόντων αυτών όπως για παράδειγμα είναι ο χρόνος, η τοποθεσία, οι καιρικές συνθήκες, τα δίκτυα που υπάρχουν σε εμβέλεια και πολλές ακόμη μεταβλητές, αποτελούν το λεγόμενο context. Με βάση το context τα υπολογιστικά συστήματα επιτυγχάνουν την παροχή περισσότερων υπηρεσιών προς τους χρήστες με πιο αποδοτικό τρόπο.

Η αντίληψη αυτή του περιβάλλοντος από τις εφαρμογές ονομάζεται context-awareness και είναι ευρέως διαδεδομένη σχεδόν σε όλες τις συσκευές της σύγχρονης εποχής. Το πρόβλημα λοιπόν πλέον καθορίζεται ως ανάγκη εξεύρεσης μιας κοινής πλατφόρμας πάνω στην οποία να μπορούν εύκολα και ευέλικτα να δημιουργηθούν εφαρμογές οι οποίες να χρησιμοποιούν ομοιόμορφα το context μέσα στο οποίο βρίσκονται. Χρειαζόμαστε επομένως ένα ενδιάμεσο λογισμικό – πλαίσιο το οποίο να παρέχει την παραπάνω δυνατότητα.

Έχοντας το ενδιάμεσο λογισμικό, χρειαζόμαστε μία εφαρμογή για να μπορέσουμε να ανακαλύψουμε στην πράξη την αξία ενός τέτοιου κοινού πλαισίου. Λήφθηκε υπόψη ότι ένα ψυχαγωγικό, ευρέως γνωστό παιχνίδι θα είχε μεγαλύτερη απήχηση και συνεπώς μεγαλύτερη πρακτική αξία. Έτσι, αποφασίστηκε ότι το παιχνίδι τράπουλας πόκερ είναι μία από τις καλύτερες λύσεις για να δοκιμάσουμε τις δυνατότητες μιας εφαρμογής ενδιάμεσου λογισμικού που να πληροί της παραπάνω συνθήκες.

1.2 Τεχνική Ορολογία

Το webinos είναι ένα περιβάλλον επίγνωσης πλαισίου για καταναλωμένες εφαρμογές πολλαπλών πλατφόρμων. Χρηματοδοτείται από την ΕΕ και έχει στόχο να προσφέρει μια πλατφόρμα για εφαρμογές διαδικτύου σε κινητά, προσωπικούς υπολογιστές, οικιακά συστήματα πολυμέσων και συσκευές αυτοκινήτου.

Ο όρος context-awareness αναφέρεται στην ικανότητα μιας εφαρμογής να έχει επίγνωση του πλαισίου μέσα στο οποίο λειτουργεί και να μπορεί να το εκμεταλλεύεται καταλλήλως.

PZ είναι η προσωπική ζώνη όπως ορίζεται στην πλατφόρμα ενδιάμεσου λογισμικού webinos. Προσφέρει σε έναν χρήστη τη βάση ώστε να μπορεί να διευθύνει και να χρησιμοποιεί όλες τις συσκευές του εισάγοντας τις σε αυτή την προσωπική ζώνη.

Το PZH, συντομογραφία για το Personal Zone Hubs ορίζεται από το webinos ως μία υπηρεσία η οποία είναι προσβάσιμη από το Internet διαρκώς. Υπάρχει μία αντιστοιχία ένα προς ένα με τις προσωπικές ζώνες και κάθε PZH ανήκει σε έναν μοναδικό χρήστη.

PZP είναι τα Personal Zone Proxies, σε ελεύθερη μετάφραση τα πληρεξούσια προσωπικών ζωνών. Το πληρεξούσιο τρέχει τοπικά σε μία συσκευή και φιλοξενεί υπηρεσίες και εφαρμογές του webinos. Είναι υπεύθυνο για τις λειτουργικότητες μίας συσκευής.

HTML (HyperText Markup Language) ³ είναι η βασική γλώσσα δημιουργία διαδικτυακών ιστοσελίδων και άλλων πληροφοριών που μπορούν να εμφανιστούν σε φυλλομετρητή (browser). Αποτελείται από HTML αντικείμενα τα οποία ενσωματώνονται μέσα σε γωνιώδεις αγκύλες (angle brackets).

Τα CSS αρχεία (Cascading Style Sheets) ⁴ ενσωματώνουν μία γλώσσα η οποία χρησιμοποιείται για σκοπούς εμφάνισης ενός κειμένου γραμμένου σε HTML, XHTML, XML στους φυλλομετρητές. Καθορίζουν στοιχεία όπως το σχέδιο, τα χρώματα και τις γραμματοσειρές που χρησιμοποιούνται.

Η JavaScript ⁵ είναι μια γλώσσα προγραμματισμού η οποία επιτρέπει την εκτέλεση εντολών στην πλευρά του χρήστη για αλληλεπιδράσεις μαζί του, χρησιμοποιώντας τον φυλλομετρητή, επικοινωνώντας ασύγχρονα και αλλάζοντας δυναμικά τα δεδομένα περιεχομένου που προβάλλονται στον χρήστη.

Το SVG (Scalable Vector Graphics) ⁶ είναι μια μορφή διανυσματικής απεικόνισης βασισμένη σε XML (Extensible Markup Language) για διαστάσιμα γραφικά με υποστήριξη αλληλεπίδρασης και κίνησης των στοιχείων του.

1.3 Οδηγός Επομένων Κεφαλαίων

Στα επόμενα κεφάλαια αναφέρεται πιο συγκεκριμένα το πλαίσιο μέσα στο οποίο δημιουργήθηκε, υλοποιήθηκε και λειτούργησε η εφαρμογή webinos poker με τη χρήση του ενδιάμεσου λογισμικού webinos.

Στο 2^ο κεφάλαιο αναλύεται με σύντομο τρόπο η πλατφόρμα ενδιάμεσου λογισμικού webinos. Ο αναγνώστης εισάγεται στην φιλοσοφία και το πλαίσιο ανάπτυξης της πλατφόρμας. Αναφέρονται τα χαρακτηριστικά και επεξηγείται ο τρόπος λειτουργίας της. Στο τέλος του κεφαλαίου διακρίνουμε τα APIs (Application Programming Interface – Διεπαφή Προγραμματισμού Εφαρμογών) τα οποία προσφέρονται δημόσια υπό την μορφή του ανοιχτού κώδικα για το πλαίσιο εφαρμογών webinos.

Στο 3^ο κεφάλαιο γίνεται εισαγωγή στην ίδια την εφαρμογή και στους λόγους που οδήγησαν στην επιλογή της συγκεκριμένης εφαρμογής ως θέμα της παρούσης διπλωματικής. Επεξηγούνται τα χαρακτηριστικά και ο τρόπος σύλληψης και συγγραφής του κώδικα όπου ακολουθήθηκαν συγκεκριμένες μοντελοποιημένες μέθοδοι ώστε να παραχθεί το βέλτιστο δυνατό αποτέλεσμα. Ιδιαίτερη μνεία γίνεται στο τεχνολογία SVG που χρησιμοποιήθηκε και οι λόγοι που οδήγησαν σε αυτή την επιλογή. Αναφέρονται τα APIs που αξιοποιήθηκαν από την πλατφόρμα ενδιάμεσου λογισμικού webinos. Παρουσιάζεται ο τρόπος λειτουργίας της εφαρμογής με ενδεικτικά στοιχεία κώδικα και τέλος παρέχεται ένα παράδειγμα λειτουργίας της εφαρμογής ώστε να γίνει αντιληπτή η ευκολία και η καινοτομία της εφαρμογής με χρήση του πλαισίου webinos.

Τέλος, στο 4^ο κεφάλαιο βρίσκεται ο επίλογος με την σύνοψη και τα συμπεράσματα της διπλωματικής εργασίας και πιθανές μελλοντικές επεκτάσεις της εφαρμογής που θα την έκαναν ακόμη καλύτερη.

2 **Webinos**

- 2.1** Εισαγωγή - Ιστορικά
- 2.2** Χαρακτηριστικά
- 2.3** Τρόπος Λειτουργίας
- 2.4** Τα APIs

2.1 Εισαγωγή - Ιστορικά

Το webinos είναι ένα έργο χρηματοδοτούμενο από την ΕΕ με κύριο σκοπό να παραδώσει μια πλατφόρμα για δικτυακές εφαρμογές σε κινητές συσκευές, προσωπικούς υπολογιστές, οικιακές συσκευές πολυμέσων (τηλεόραση) και ενσωματωμένων σε αυτοκίνητα συστημάτων. Ξεκίνησε τον Σεπτέμβριο του 2010 και είχε προγραμματισμένη διάρκεια τριών ετών μέχρι τον Αύγουστο του 2013. Παραπάνω από είκοσι συνεργάτες σε όλη την Ευρώπη συνετέλεσαν στο έργο. Μεταξύ των συνεργατών βρίσκονται ακαδημαϊκά ινστιτούτα, εταιρείες ανάπτυξης λογισμικού, εταιρείες βιομηχανικής έρευνας, αυτοκινητοβιομηχανίες και άλλοι. Το πρόγραμμα λειτουργεί υπό το [EU FP7 ICT Programme](#) με προϋπολογισμό 14 εκατομμυρίων ευρώ, 10 από τα οποία είναι χορηγία της Ευρωπαϊκής Ένωσης. Ο πλήρης τίτλος του έργου είναι webinos (Secure Web Operating System Application Delivery Environment).

Στόχος του webinos είναι να προσφέρει μια πλατφόρμα η οποία θα υποστηρίζει και θα παρέχει ένα ομογενοποιημένο πλαίσιο για πολλαπλές συσκευές, πολλαπλούς χρήστες και πολλαπλά λειτουργικά περιβάλλοντα μέσα σε ένα ασφαλές δίκτυο. Υπάρχουν λογισμικά με παρόμοια χαρακτηριστικά τα οποία προσπαθούν να γεφυρώσουν το χάσμα των διαφορετικών πλαισίων, όπως το PhoneGap, το Bondi, και άλλα. Τα περισσότερα όμως από αυτά, αν όχι όλα, παρουσιάζουν ελλείψεις σε ένα ή και περισσότερα στοιχεία τα οποία έρχεται να καλύψει το webinos.

Το όραμα του webinos είναι να μπορέσει να προχωρήσει την χρήση των υπολογιστικών συστημάτων κατά ένα λογικό βήμα, να επιφέρει την διάχυτη υπολογιστική (ή αλλιώς πανταχού υπολογιστική παρουσία, ubiquity). Για να το κάνει αυτό, θα πρέπει η εκάστοτε συσκευή γνωρίζοντας το πλαίσιο γύρω της να έχει τη δυνατότητα λήψης αντίστοιχων αποφάσεων με βάση αυτό.

Ο πηγαίος κώδικας του πλαισίου webinos διατίθεται υπό την μορφή του ανοιχτού κώδικα (open source), στην υπηρεσία Github ⁷, μια υπηρεσία φιλοξενίας κώδικα για έργα ανάπτυξης λογισμικού με χρήση του συστήματος ελέγχου αναθεωρήσεων (revision control system) git ⁸, καταμερισμένο σε διάφορες δημόσιες αποθήκες (public repositories) για την διευκόλυνση του προγραμματιστή, όλες άμεσα διαθέσιμες στη διεύθυνση <https://github.com/webinos>. Επιπλέον οδηγίες για

την εγκατάσταση και χρησιμοποίηση του παραπάνω κώδικα βρίσκονται στο portal της επίσημης σελίδας της πλατφόρμας ενδιάμεσου λογισμικού webinos, στο <https://developer.webinos.org/>.

2.2 Χαρακτηριστικά

Για να μπορέσει να πετύχει το webinos το όραμα του, δηλαδή δικτυακές εφαρμογές οι οποίες υλοποιούνται σε ένα εύρος συνδεδεμένων συσκευών με τη δυνατότητα απρόσκοπτης αλληλεπίδρασης μεταξύ τους, παρέχει μια πλατφόρμα ανοιχτού κώδικα (open source) με πρακτική εφαρμογή και προσιτή σε όλους.

Τα κύρια χαρακτηριστικά τα οποία παρουσιάζει το webinos για να επιτύχει το παραπάνω όραμα στην πλατφόρμα του ανοιχτού κώδικα είναι:

- Βασίζεται στα επιτεύγματα της διαδικτυακής κοινότητας και επεκτείνει το open source web runtime environment.
- Προσφέρει ένα κοινό σύνολο διεπαφών προγραμματισμού εφαρμογών (APIs) ώστε να επιτρέπει τη χρήση του με ανοιχτό και ασφαλή τρόπο σε ένα σύνολο λειτουργικών, συσκευών και πιθανών δικτύων.
- Οι εφαρμογές του μπορούν να χρησιμοποιηθούν σε ένα σύνολο πλαισίων σε όλες τις συσκευές ενός χρήστη.
- Οι εφαρμογές μπορούν να θέσουν σε λειτουργία και να χρησιμοποιήσουν όλες τις δυνατότητες και τους πόρους που διαθέτει το υλικό (hardware) μιας συγκεκριμένης συσκευής
- Οι εφαρμογές μπορούν να έχουν ασφαλή πρόσβαση σε ιδιωτικά και μη ιδιωτικά δεδομένα και άλλες υπηρεσίες στο cloud και στα κοινωνικά δίκτυα, όπως και σε στοιχεία στο τερματικό του χρήστη.
- Οι εφαρμογές χτίζονται μία φορά από τους παρόχους λογισμικού και μπορούν να παραταχθούν οπουδήποτε.
- Οι κατασκευαστές και οι συντηρητές δικτύων διαθέτουν μια πλατφόρμα υπηρεσιών η οποία συμμορφώνεται στους κανονισμούς του ανοιχτού λογισμικού (open source) και των ανοιχτών προτύπων (open standards), τα οποία ισχύουν για τις συσκευές που καλύπτονται από το webinos, δηλαδή τα κινητά, τους προσωπικούς υπολογιστές, τα οικιακά πολυμέσα και τα συστήματα εντός αυτοκινήτων.

Οι επιλεγμένες γλώσσες που χρησιμοποιούνται στο webinos υποστηρίζουν πλήρως τις διαδικτυακές τεχνολογίες, συγκεκριμένα πρόκειται για τις HTML, CSS και

JavaScript. Το webinar καλείται να διευθετήσει συγκεκριμένες προκλήσεις όπως είναι οι πρόβλεψη και προσαρμογή της ασφάλειας σε ένα μεγάλο εύρος συσκευών, υπηρεσιών και δικτύων. Εξίσου σημαντική είναι η ασφάλεια των προσωπικών δεδομένων των χρηστών ανεξαρτήτως της συσκευής ή της εφαρμογής που χρησιμοποιούν.

Οι προδιαγραφές και η ανάπτυξη του webinar λειτούργησε με βάση 18-μηνιαίες περιόδους οι οποίες ακολουθούσαν την λογική:

1. Ανάλυση απαιτήσεων
2. Συλλογή προδιαγραφών
3. Ανάπτυξη πλατφόρμας εφαρμογής
4. Έλεγχος και αξιολόγηση

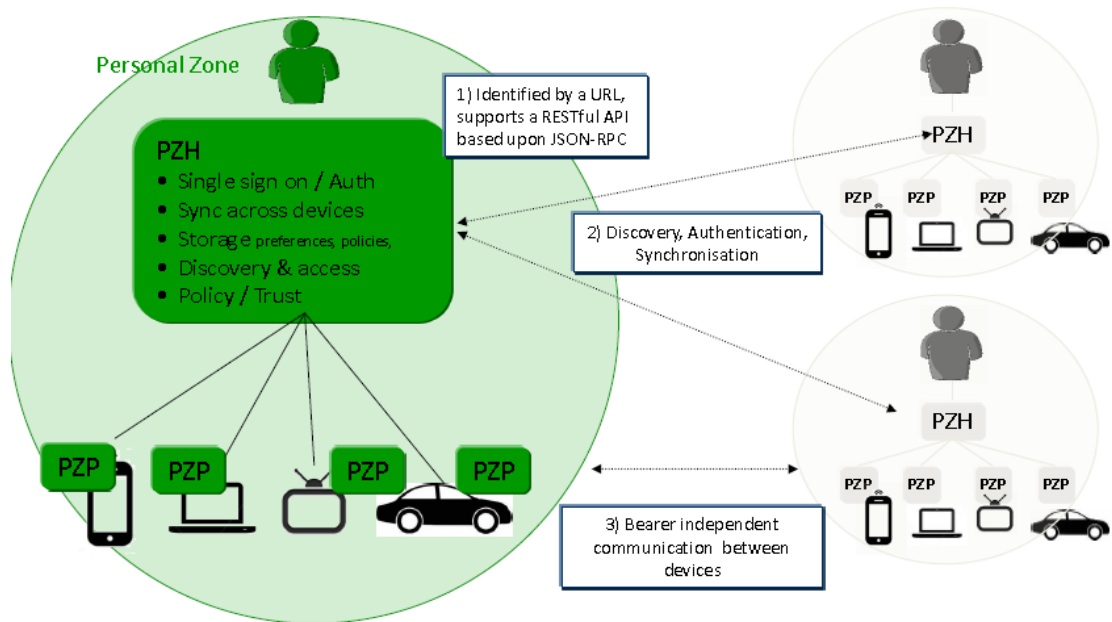
2.3 Τρόπος Λειτουργίας

Το webinos λειτουργεί με βάση κάποιες συμβάσεις κατά τις οποίες διαχωρίζει τον χώρο σε προσωπικές ζώνες ως μια μορφή επικαλύμματος δικτύου (overlay network). Η προσωπική ζώνη παρέχει τη βάση που χρειαζόμαστε για να διαχειριστούμε τις συσκευές του χρήστη μαζί με τις υπηρεσίες που τρέχουν σε αυτές. Μέσα σε αυτές τις υπηρεσίες περιλαμβάνονται οι προσωπικές υπηρεσίες του χρήστη στο cloud.

2.3.1 Προσωπική Ζώνη (Personal Zone)

Ο κάθε χρήστης μπορεί να έχει περισσότερες από μία συσκευές ενεργοποιημένες στο webinos. Το σύνολο αυτών των συσκευών αποτελεί την Προσωπική του Ζώνη (Personal Zone). Η Προσωπική Ζώνη δρα ως ένα ξεχωριστό δίκτυο σε σχέση με τα υποκείμενα φυσικά δίκτυα και πρωτόκολλα. Σκοπός της είναι να παρέχει εύκολη πρόσβαση σε τοπικές και απομακρυσμένες υπηρεσίες, απλοποιώντας το έργο ενός προγραμματιστή εφαρμογών. Επίσης πρέπει να επιτρέπει την εύκολη ανίχνευση συσκευών και υπηρεσιών και να παρέχει επικοινωνιακά μονοπάτια βασισμένα σε έμπιστες σχέσεις, αποσυνδεδεμένα από τις υποκείμενες τεχνολογίες.

Η Προσωπική Ζώνη υλοποιείται πάνω σε μία διανεμημένη αρχή η οποία αποτελείται από έναν Κόμβο Προσωπικής Ζώνης (Personal Zone Hub – PZH) και πολλαπλά Πληρεξούσια Προσωπικής Ζώνης (Personal Zone Proxy – PZP). Στην Εικόνα 1 βλέπουμε την βασική αντίληψη της Προσωπικής Ζώνης.



Εικόνα 1 – Προσωπική Ζώνη

Μέσα στην Προσωπική Ζώνη, κάθε συσκευή η οποία διαθέτει PZP συνδέεται άμεσα με ένα PZH.

2.3.2 Κόμβος Προσωπικής Ζώνης (Personal Zone Hub – PZH)

Ο Κόμβος Προσωπικής Ζώνης (PZH) χρησιμοποιείται για τη δημιουργία μίας Προσωπικής Ζώνης. Είναι το κύριο στοιχείο πάνω στο οποίο συγχρονίζονται και ταυτοποιούνται όλες οι συσκευές ενός χρήστη. Σκοπός του πέρα από τη δημιουργία της Ζώνης είναι να επιτρέπει με ασφαλή τρόπο στις συσκευές να επικοινωνούν μεταξύ τους καθώς και την ανίχνευση και πρόσβαση σε συσκευές άλλων Προσωπικών Ζωνών.

Πρακτικά η υλοποίηση βασίζεται σε μία τοποθέτηση του PZH στο σύννεφο, υποθέτοντας ότι αυτή είναι ασφαλής, ακόμα και αν αυτή γίνεται σε εξυπηρετητή

τρίτου. Οι επιτρεπτές λειτουργίες του PZH είναι μέσω της διεπαφής να προσθέτει μία συσκευή στην PZH , να ανακαλεί πιστοποιητικά για κάθε ταυτοποιημένη συσκευή και να ελέγχει την κατάσταση ταυτοποίησής της.

Για να επιτρέψει εξωτερική παρέμβαση σε μία Προσωπική Ζώνη, το webinos καθορίζει τον Κόμβο Προσωπικής Ζώνης ως μια υπηρεσία η οποία είναι προσιτή μέσω του Διαδικτύου, το οποίο είναι διαθέσιμο οποιαδήποτε στιγμή, αντίθετα με τις προσωπικές συσκευές οι οποίες μπορούν να απενεργοποιηθούν ή να βρίσκονται εκτός εύρους επικοινωνίας. Υπάρχει μια 1-1 αντιστοιχία μεταξύ των Προσωπικών Ζωνών και των Κόμβων Προσωπικών Ζωνών. Κάθε PZH ανήκει σε έναν μοναδικό χρήστη και ταυτοποιείται μέσω συγκεκριμένης URL (uniform resource locator). Για παράδειγμα όταν ο ιδιοκτήτης χρησιμοποιεί έναν δημόσιο υπολογιστή, το PZH του επιτρέπει να αποκτήσει πρόσβαση στις συσκευές της Προσωπικής του Ζώνης και τις υπηρεσίες αυτής κατά τη διάρκεια περιόδου περιήγησης του. Επιτρέπει επιπρόσθετα πρόσβαση σε άλλους χρήστες οι οποίοι έχουν λάβει το δικαίωμα από τον ιδιοκτήτη για αυτή την ενέργεια.

Το PZH παρέχει την υποστήριξη ανακάλυψης άλλων PZH με βάση τα όνομα, ψευδώνυμο ή διεύθυνση e-mail των άλλων χρηστών. Υπάρχει βέβαια πάντα η δυνατότητα να περιοριστεί η εξεύρεση του PZH από άλλους. Αυτό καθορίζεται από το κάθε χρήστη.

Ακολουθούν οι κρίσιμες λειτουργίες τις οποίες παρέχει το PZH:

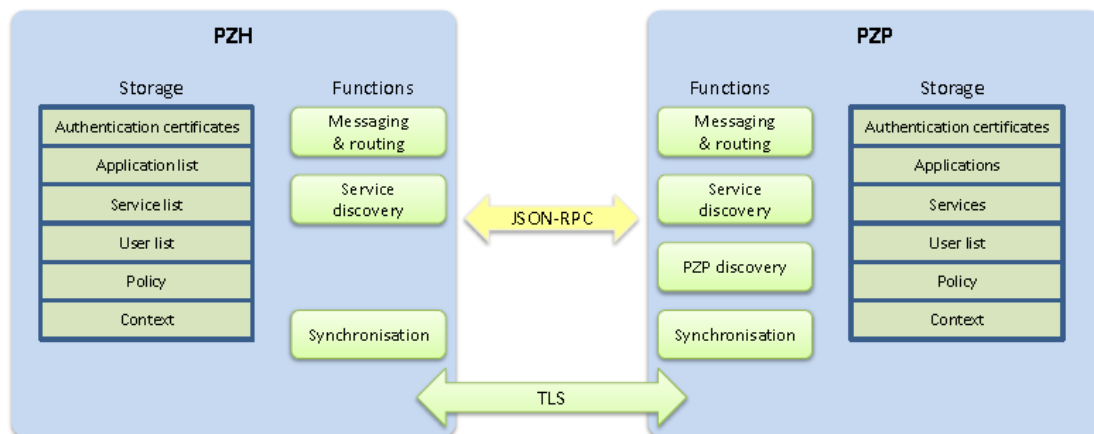
- Μια καθορισμένη οντότητα στην οποία μπορούν να σταλούν και να μεταδοθούν όλα τα μηνύματα – σαν μια προσωπική ταχυδρομική θυρίδα
- Πιστοποιητικά αναγνώρισης PZP/PZH από άλλους χρήστες οι οποίοι θεωρούνται έμπιστοι
- Αναγνωριστικά και πιστοποιητικά εφαρμογών τα οποία έχουν πρόσβαση στη Ζώνη
- Το ίδιο για υπηρεσίες
- Το ίδιο για συνδεδεμένες συσκευές
- Όλοι οι κανόνες πολιτικής που χρησιμοποιούνται
- Όλα τα σχετικά δεδομένα πλαισίου
- Ταυτοποίηση χρήστη
- Δημιουργία ασφαλούς συνεδρίας PZP για την μετάδοση μηνυμάτων και συγχρονισμό

- Υπηρεσία δημιουργίας συνεδρίας για ασφαλή μετάδοση μηνυμάτων μεταξύ των εφαρμογών και των υπηρεσιών
- Ασφαλή κοινωνική δικτύωση χρησιμοποιώντας τα πιστοποιητικά που έχουν ανταλλαχθεί μεταξύ εμπιστων χρηστών
- Πιθανή υπηρεσία σύνδεσης σε άλλες δικτυακές υπηρεσίες χρησιμοποιώντας το PZH ως ασφαλή κόμβο
- Συγχρονισμός πλαισίου: Το PZH ενεργεί ως κύριο αποθετήριο για όλα τα δεδομένα πλαισίου

2.3.3 Πληρεξούσιο Προσωπικής Ζώνης (Personal Zone Proxy - PZP)

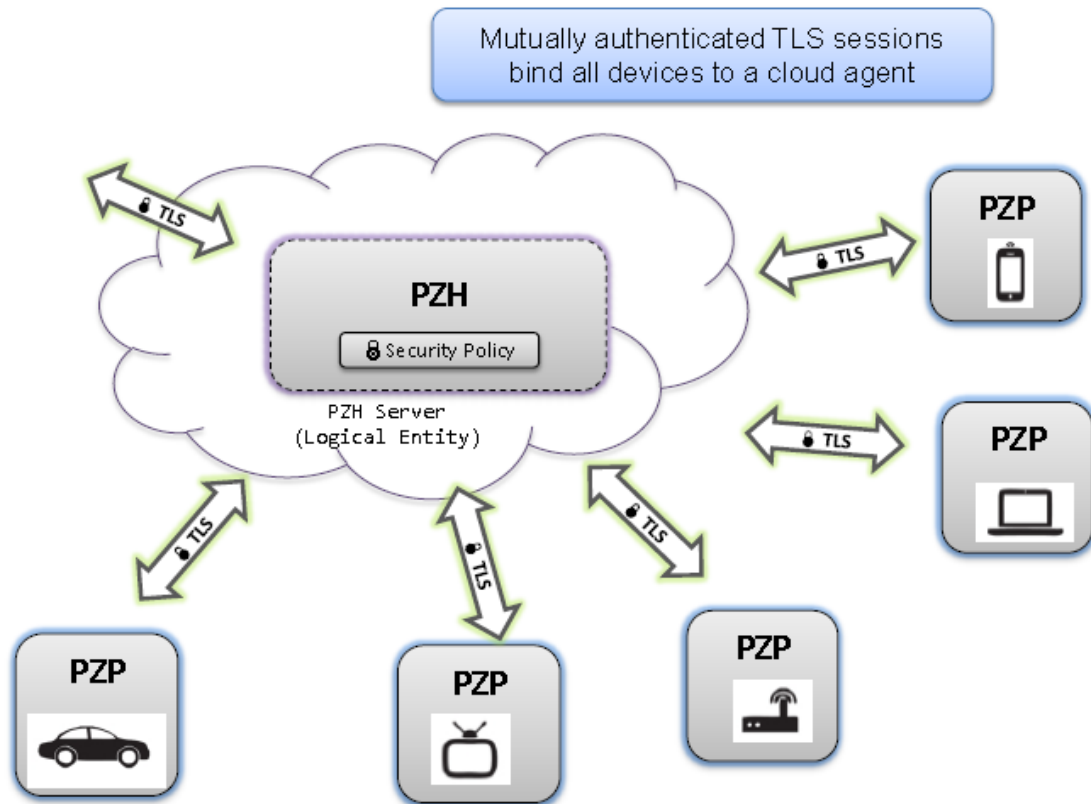
Ένα PZP τρέχει σε κάθε webinos ενεργοποιημένη συσκευή και είναι υπεύθυνο για την επικοινωνία της με το PZH. Αυτή επιτυγχάνεται μέσω ενός ασφαλούς TLS καναλιού που ταυτοποιεί αμοιβαία και τις δύο πλευρές, και σε περίπτωση σφάλματος στην επικοινωνία μπορεί να γίνει έμμεση ταυτοποίηση μέσω της σύνδεσης του με ένα ήδη ταυτοποιημένο PZP. Στην περίπτωση που ο PZH δεν είναι διαθέσιμος λόγω σφάλματος στη σύνδεση με το δίκτυο, είναι δουλειά του PZP να λειτουργήσει στη θέση του και να αποθηκεύσει τις απαραίτητες πληροφορίες και γεγονότα συγχρονίζοντας αυτές αργότερα με το PZH, όταν η σύνδεση είναι πλέον δυνατή. Επιπρόσθετα, το PZP ευθύνεται για κάθε ανίχνευση χρησιμοποιώντας φορείς τοπικού υλικού. Ο βασικός του ρόλος είναι η εξαγωγή των APIs της συσκευής ως υπηρεσίες με έναν απόλυτα ασφαλή τρόπο. Ακόμα το PZP συνδέεται στον webinos RunTime (WRT) με μία διαμορφωμένη εφαρμογή διαδικτυακής υποδοχής η οποία ονομάζεται ασφαλής διάυλος (secure channel). Το WRT είναι το μέρος όπου όλες οι ανεπτυγμένες εφαρμογές ανήκουν και τρέχουν. Τέλος, μετά τον ορισμό της εξωτερικής διεπαφής στο PZP, μπορούμε να χρησιμοποιήσουμε ένα ψεύδο-PZP για να κάνουμε δυνατή την πρόσβαση του webinos σε συσκευές, στις οποίες αυτό δεν μπορεί να εφαρμοστεί πλήρως, και με αυτό τον τρόπο να έχουμε πρόσβαση σε υπηρεσίες διαθέσιμες σε αυτές τις συσκευές.

Στην Εικόνα 2 παρουσιάζεται η αρχιτεκτονική των PZH & PZP και η ενδιάμεση συνομιλία τους με χρησιμοποίηση του JSON-RPC ⁹ το οποίο είναι θεμελιωμένο πάνω σε TLS (Transport Layer Security) ¹⁰ συνεδρίες.



Εικόνα 2 – Επικοινωνία PZH - PZP

Μια πιο γενική θεώρηση της μετάδοσης δεδομένων μεταξύ των PZH και PZP μπορεί να γίνει αντιληπτή μέσω της Εικόνας 3 όπου ξεχωρίζουμε ανάγκη ασφαλούς σύνδεσης μέσω των συνεδριών TLS.



Εικόνα 3 – Σύνδεση συσκευών PZP μέσω TLS με το PZH

Εικονικό Πληρεξούσιο Προσωπικής Ζώνης (virtual PZP)

Το εικονικό πληρεξούσιο προσωπικής ζώνης δημιουργήθηκε μεταγενέστερα των υπολοίπων συστατικών μίας προσωπικής ζώνης και ο ρόλος του αρχικά ήταν να «ελαφρύνει» της λειτουργίες του PZH. Αυτό διότι από την αρχή η ιδέα ήταν το ενδιαμέσο λογισμικό να μην επιβαρύνει σημαντικά την υπολογιστική μνήμη των συσκευών. Για το σκοπό αυτό η υλοποίηση του γίνεται μέσα σε έναν διακομιστή του διαδικτύου και όχι απαραίτητα εντός της προσωπικής ζώνης ενός χρήστη. Οι περαιτέρω λειτουργίες του δεν έχουν αποσαφηνιστεί πλήρως, μιας και είναι ακόμα σε αρχικό στάδιο η σχεδιάσή και η ανάπτυξή του. Η επικρατέστερη άποψη προβλέπει την δημιουργία διαφορετικών vPZP, τα οποία θα διαφοροποιούνται στο είδος των υπηρεσιών και λειτουργιών που το καθένα θα προσφέρει (π.χ. Context vPZP, Notification vPZP, Media vPZP και άλλα). Η κύρια ιδέα προβλέπει ακόμα την ομαδοποίηση των επιμέρους διαφορετικών vPZP, σε κοινούς χώρους του ίδιου διακομιστή (vPZP farms). Η μελέτη του θα μας απασχολήσει στη συνέχεια της

εργασίας, καθώς εντός του context vPZP τοποθετείται η Context βάση δεδομένων, όπου αποθηκεύονται όλα τα δεδομένα περιβάλλοντος, της προσωπικής ζώνης ενός χρήστη.

2.3.4 Τοπικές Συνδέσεις

Ένα από τα χαρακτηριστικά που παρέχει το webinos είναι μια ενιαία τοπική υπηρεσία σύνδεσης με βάση τη φυσική εγγύτητα, κάνοντας τις διαφορετικές τεχνολογίες διασύνδεσης διαφανή. Μια συσκευή με ενεργοποιημένο το webinos είναι σε θέση να βρει οποιαδήποτε από τις υπόλοιπες διασυνδεδεμένες συσκευές γύρω από αυτή. Το δίκτυο επικάλυψης επιτρέπει σε διαφορετικές εφαρμογές και υπηρεσίες να μιλήσουν μεταξύ τους πάνω από αυτές τις διαφορετικές τεχνολογίες διασύνδεσης, απομονώνοντας τον κωδικό της εφαρμογής από τις υπόλοιπες λεπτομέρειες. Το PZP επιτρέπει στις εφαρμογές να έχουν πρόσβαση σε υπηρεσίες άλλων τοπικών συσκευών ενεργοποιημένων στο webinos στην ίδια Προσωπική Ζώνη, καθώς και για την παροχή υπηρεσιών σε μη webinos συσκευές.

Οι συσκευές οι οποίες βρίσκονται στο ίδιο φυσικό χώρο μπορεί να ανήκουν σε διαφορετικούς χρήστες και με βάση αυτή την λογική ανήκουν σε διαφορετικές Προσωπικές Ζώνες. Αυτό σημαίνει ότι η σύνδεση μπορεί να επιτευχθεί μεταξύ διαφορετικών Προσωπικών Ζωνών και οι χρήστες μπορούν να μοιραστούν τις υπηρεσίες και τις εφαρμογές των συσκευών, μεταξύ τους.

2.3.5 Συγχρονισμός

Το webinos παρέχει λειτουργία συγχρονισμού με βάση το rsync [RSNC] . Συμπεριλαμβάνει την ανίχνευση και τη συγχώνευση των διαφορών , και ερώτηση προς το χρήστη για την επίλυση των συγκρούσεων , λαμβάνοντας υπόψη τις περιόδους χρήσης εκτός σύνδεσης (offline) . Η διαδικασία περιλαμβάνει μια σύγκριση των ρολογιών ως βάση για τη διόρθωση πριν προχωρήσει στη σύγκριση του χρόνου κάθε αλλαγής. Η προσέγγιση αυτή είναι εμπνευσμένη από το έργο για τον καταναμημένο έλεγχο αναθεώρησης. Ο συγχρονισμός πραγματοποιείται όταν μια συσκευή είναι εγγεγραμμένη στη Προσωπική Ζώνη ή όταν συμβαίνουν αλλαγές. Αυτό συμβαίνει επίσης σε συνδυασμό με την τοπική ανακάλυψη, ώστε να ενεργοποιήσει ένα κοινό μοντέλο του πλαισίου. Για τα δίκτυα που βασίζονται στην IP , multicast ανακοινώσεις και απαντήσεις ερωτημάτων μπορούν να παρατηρηθούν να ενημερώνουν μια τοπική μνήμη. Ο συγχρονισμός και η ασφαλή πρόσβαση στο πλαίσιο αποτελούν ένα σημαντικό μέρος της πλατφόρμας webinos. Οι φυλλομετρητές (browsers) υποστηρίζουν ήδη τους μηχανισμούς για την καταγραφή των προτιμήσεων και την τοπική αποθήκευση δεδομένων της εφαρμογής. Το webinos μπορεί να βασιστεί πάνω σε αυτά τα επιπρόσθετα αρχεία της βάσης δεδομένων που πραγματοποιήθηκε ως μέρος του προφίλ του προγράμματος περιήγησης, και είναι προσβάσιμο από αξιόπιστο κώδικα σε επεκτάσεις του προγράμματος περιήγησης . Το webinos μπορεί να χρησιμοποιήσει αρχεία JSON [11](#) για την ανταλλαγή μηνυμάτων συγχρονισμού .

Ο συγχρονισμός χρειάζεται για να λειτουργήσει ακόμη και όταν η συσκευή λειτουργεί με ένα υποσύνολο μιας Προσωπικής Ζώνης, σε περίπτωση απουσίας της πρόσβασης στο Διαδίκτυο. Αυτό στηρίζεται στη δυνατότητα συγχρονισμού των συσκευών σε ένα δίκτυο από ομότιμους χρήστες. Ο συγχρονισμός εξαρτάται από τη δυνατότητα να συγχωνεύσετε τις αλλαγές και τον εντοπισμό και την επίλυση αντιφατικών αλλαγών . Εάν το μοντέλο δεδομένων πλαισίου είναι ανεξάρτητο, τότε μία προσέγγιση είναι να συγχρονίσει απλώς την τελευταία αλλαγή σε ένα συγκεκριμένο μέρος του πλαισίου. Εάν το μοντέλο δεδομένων πλαισίου έχει αλληλεξαρτήσεις, το επικαιροποιημένο μοντέλο πρέπει να πληροί τους περιορισμούς ακεραιότητας .

2.3.6 Πολιτικές Συστήματος

Στο webinos οι πολιτικές είναι γραμμένες σε ένα αρχείο XML. Σε αυτό το αρχείο είναι πιθανό να οριστούν πολλαπλές δέσμες πολιτικών: κάθε σύνολο έχει έναν «συνδυαστικό» κανόνα που χρησιμοποιείται για να καθορίσει ποια πολιτική θα εφαρμοστεί. Ο στόχος της πολιτικής θα μπορούσε να αφορά χρήστες, εφαρμογές, συσκευές ή συνδυασμούς τους.

Η διαχείριση της πολιτικής είναι μια υπηρεσία που υλοποιείται μέσα στο πλαίσιο του PZP . Η υπηρεσία διαχείρισης της πολιτικής προστατεύει την ιδιοκτησία και έχει πρόσβαση στην επιβολή ελέγχου αιτημάτων ώστε να γίνεται χρηστή διαχείριση των προσωπικών δεδομένων του χρήστη και ελέγχει την πρόσβαση στις τοπικές δυνατότητες των συσκευών και τα χαρακτηριστικά τους. Αυτό γίνεται με το ταίριασμα των αιτήσεων βάσει των γραπτών πολιτικών, προκειμένου να καθοριστεί η πρόσβαση ή μη στους ζητούμενους πόρους.

2.3.7 Το webinos στη συσκευή

Μια συσκευή με εγκατεστημένο webinos, είναι μια συσκευή με ένα PZP που τρέχει σε αυτή και συνήθως ένα webinos Widget Runtime (WRT) περιβάλλον για την εκτέλεση webinos εφαρμογών. Οι οντότητες webinos εγκαθίστανται ως ενδιάμεσο λογισμικό στη συσκευή. Οι webinos εφαρμογές τρέχουν στον φυλλομετρητή του Διαδικτύου ή σαν ένα widget το οποίο ερμηνεύεται από το λειτουργικό σύστημα της συσκευής.

Ένα WRT είναι ένας ειδικός τύπος του προγράμματος περιήγησης. Θα πρέπει να είναι σε θέση να ερμηνεύει τις τελευταίες προδιαγραφές JavaScript, HTML και CSS. Είναι υπεύθυνος για την απόδοση των στοιχείων UI (user interface) της webinos εφαρμογής. Ως μέρος του WRT, ένα webinos αντικείμενο παρέχει το σύνολο των βασικών APIs ως μεθόδους και ιδιότητες για να καταστήσει δυνατή την χρήση των

συγκεκριμένων δυνατοτήτων της συσκευής, δίνοντας έτσι την δυνατότητα της εφαρμογής webinos να τρέξει σε διαφορετικές πλατφόρμες συσκευής. Μέσω αυτού του αντικειμένου, άλλοι προγραμματιστές είναι σε θέση να έχουν πρόσβαση στο λειτουργικότητα του πλαισίου webinos. Ένα WRT παρουσιάζει τις περιβαλλοντικές ιδιότητες και τα κρίσιμα γεγονότα στο PZP, έτσι ώστε να μπορεί να επεξεργάζεται την πολιτική ασφαλείας και τα συμφραζόμενα γεγονότα σωστά.

2.3.8 Υπηρεσίες του webinos

Για τις εφαρμογές που τη χρησιμοποιούν, η υπηρεσία webinos είναι ένα ειδικό webinos API. Παρέχει μια συλλογή από λειτουργίες και γεγονότα τα οποία είναι προσβάσιμα από την εφαρμογή webinos. Αυτές οι λειτουργίες και τα γεγονότα είναι πάντα διαθέσιμα στον προγραμματιστή εφαρμογών ως ένα σύνολο από συναρτήσεις στη γλώσσα JavaScript, ανεξαρτήτως του που βρίσκεται η υλοποίηση. Εκεί υπάρχουν οι ακόλουθοι τύποι υπηρεσιών webinos:

- APIs τα οποία μπορούν να προσπελαστούν απομακρυσμένα χρησιμοποιώντας το JSON-RPC. Το webinos API φιλοξενείται από ένα PZP και η πρόσβαση σε αυτό διαμεσολαβείται από έναν διαχειριστή πολιτικών στα PZP τόσο του καλούντος όσο και του παρόχου.

Μια εφαρμογή είναι μια υπηρεσία webinos εάν παρουσιάζει εξωτερικές υπηρεσίες ως JavaScript APIs. Η εφαρμογή φιλοξενείται στο PZP και άλλες εφαρμογές μπορούν να κάνουν χρήση της.

2.3.9 Σύνδεση, Προστασία της ιδιοκτησίας και Ασφάλεια

Η πλατφόρμα webinos παρέχει σε κάθε συσκευή ένα σύνολο από APIs για την πρόσβαση σε υπηρεσίες που εκτίθενται άμεσα από την Προσωπική Ζώνη. Ένα παράδειγμα είναι η μέθοδος που χρησιμοποιείται για την ανακάλυψη υπηρεσιών που ταιριάζουν με το συγκεκριμένο τύπο υπηρεσίας και το πλαίσιο των περιορισμών. Η μέθοδος είναι ασύγχρονη, και καταλήγει σε επανακλήσεις καθώς περιστατικά υπηρεσιών ανακαλύπτονται. Οι προγραμματιστές εφαρμογών μπορούν στη συνέχεια να παρέχουν ένα περιβάλλον εργασίας χρήστη (UI) για την επιλογή μεταξύ των εναλλακτικών λύσεων, όπου ο κατάλογος θα ενημερώνεται δυναμικά ενώ οι υπηρεσίες θα είναι διαθέσιμες ή θα παύουν να είναι διαθέσιμες. Η προσέγγιση αυτή επιτρέπει στους προγραμματιστές εφαρμογών να προσφέρουν στους χρήστες τα μέσα ώστε να λάβουν περισσότερες πληροφορίες σχετικά με κάθε μία από τις επιλογές, καθώς και να καταγράφουν τις προτιμήσεις χρήσης σε μελλοντικές καταστάσεις .

Η διαδικασία της σύνδεσης με μια υπηρεσία (που έχει πρώτα ανακαλυφθεί) περιλαμβάνει :

- αμοιβαίας επαλήθευσης ταυτότητας, όπου η ζώνη πιστοποιεί την υπηρεσία , και η υπηρεσία πιστοποιεί την ζώνη
- εξασφάλιση της επικοινωνίας μέσω της χρήσης κρυπτογραφικών πρωτοκόλλων, προστασία από υποκλοπές και επιθέσεων, πλαστών διευθύνσεων IP και πλαστογραφημένων εγγραφών DNS
- αναθεώρηση και ικανοποίηση του αιτήματος από την υπηρεσία για αυξημένα προνόμια.

Οι εφαρμογές (ή οι ενσωματωμένες υπηρεσίες) μπορούν να ζητήσουν αυξημένα προνόμια. Αυτό συνήθως αντιμετωπίζεται όταν η εφαρμογή εκτελείται για πρώτη φορά, και η απόφαση του χρήστη καταγράφεται για τις επόμενες χρήσεις. Έτσι, οι εφαρμογές θεωρούνται ως υπηρεσίες και ισχύουν οι αντίστοιχες πολιτικές. Το βασικό μοντέλο είναι αυτό της γνωστοποίησης και συγκατάθεσης. Το σχετικό UI παρέχεται από την πλατφόρμα webinos και όχι από τις εφαρμογές. Ένα επιπλέον UI παρέχεται ώστε να δώσει στους χρήστες τη δυνατότητα να επανεξετάσουν και να

ανακαλέσουν τις αποφάσεις τους. Η ίδια η συσκευή μπορεί να επιβάλει πολιτικές ασφάλειας, για παράδειγμα να θέσει μια «λευκή λίστα» (white list) με τις υπηρεσίες που μπορούν να έχουν ιδιαίτερα προνόμια.

2.3.10 Επεκτασιμότητα

Τα APIs του webinos έχουν σχεδιαστεί ώστε να προσδίδουν επεκτασιμότητα. Χρησιμοποιούν την αντικειμενοστρέφεια στην οποία είναι σύνηθες να περνάει ένα αντικείμενο ως μία νέα υπόσταση σε μία μέθοδο όπου το αντικείμενο υποστηρίζει μία ή περισσότερες διασυνδέσεις. Οι διασυνδέσεις αυτές ερμηνεύονται από άλλα στοιχεία, τα οποία είναι επίσης υπεύθυνα για την τεκμηρίωση των επεκτάσεων. Οι προγραμματιστές μπορούν να καλέσουν μια τυποποιημένη μέθοδο QueryInterface ώστε να μετατρέψουν ένα αντικείμενο σε ένα καθορισμένο περιβάλλον, όταν αυτό είναι αναγκαίο για να αποφευχθούν διενέξεις ονόματος.

Έχοντας ανακαλυφθεί και συνδεθεί μια υπηρεσία, εκτίθεται ως αντικείμενο στο περιβάλλον εκτέλεσης μιας ιστοσελίδας. Αυτό το αντικείμενο δρα ως τοπικό πληρεξούσιο (proxy) για την υπηρεσία, η οποία μπορεί να παρέχεται από μια απομακρυσμένη συσκευή. Το webinos επιτρέπει στους προγραμματιστές εφαρμογών να καταχωρήσουν μια απλή λειτουργία επανάκλησης, ή να μεταδώσουν ένα αντικείμενο το οποίο υποστηρίζει συγκεκριμένη διαπροσωπεία.

2.3.11 Εφαρμογές webinos

Μια webinos εφαρμογή φιλοξενείται από το PZP και τρέχει στη συσκευή, όπου η συσκευή θα μπορούσε να είναι προσπελάσιμη στο Internet, δηλαδή ένας Server.

Μια webinos εφαρμογή είναι συσκευασμένη, σύμφωνα με τις προδιαγραφές συσκευασίας, και εκτελείται στο πλαίσιο του WRT. Μια webinos εφαρμογή έχει πρόσβαση σε ευαίσθητα από πλευράς ασφάλειας ικανότητες, με τη μεσολάβηση του αρχείου XACML όπως ορίζεται από τον διαχειριστή της πολιτικής της συσκευής. Μια webinos εφαρμογή μπορεί να εκθέσει ορισμένες ή όλες τις ικανότητες της συσκευής ως μια webinos υπηρεσία. Ένας προγραμματιστής εφαρμογών αποκτά πρόσβαση στις δυνατότητες του webinos μέσω του JavaScript αντικειμένου του webinos.

Οι εφαρμογές webinos μπορούν να μεταφορτωθούν και να εγκατασταθούν σε συσκευές, ή μπορούν να φιλοξενοούνται από εξυπηρετητές, με συστατικά τα οποία φορτώνονται δυναμικά όταν χρειάζεται. Οι εφαρμογές μπορούν να κάνουν χρήση των υπηρεσιών, και με τη σειρά τους μπορούν να παρέχουν υπηρεσίες. Οι υπηρεσίες μπορεί να περιλαμβάνουν ένα UI που εκτίθεται ως μέρος μιας εφαρμογής, π.χ. μέσα σε ένα στοιχείο HTML iframe. Η ικανότητα να συνδυάζει και να εξατομικεύει υπηρεσίες χρησιμοποιείται για να υποστηρίξει «mashups». Οι εφαρμογές είναι ουσιαστικά υπηρεσίες που μπορούν να εγκατασταθούν ή να αποθηκευτούν ως σελιδοδείκτες.

Το webinos επιτρέπει την τοπική, client-side (από την πλευρά του χρήστη) προσαρμογή του UI της αίτησης με βάση τα χαρακτηριστικά της συσκευής. Η διαδικασία προσαρμογής παίρνει πολλά από αυτά τα χαρακτηριστικά υπόψη, με το πιο αξιοσημείωτο να είναι το μέγεθος της οθόνης και η ανάλυση, τον τύπο της συσκευής και τις λεπτομέρειες εισόδου. Για να καταστεί αυτό δυνατό , οι εφαρμογές θα δημιουργούνται με μια δηλωτική περιγραφή UI ότι είναι ανεξάρτητες από τη συσκευή. Αυτή η περιγραφή θα μείνει όσο πιο κοντά στις βασικές τεχνολογίες web είναι δυνατό. Κατά το χρόνο εκτέλεσης, η περιγραφή αυτή μετατρέπεται σε μια διάταξη HTML / CSS / JS κατάλληλη για μια web runtime .

2.3.12 Καινοτομίες webinos

Η ιδέα της προσωπική ζώνης ξεκίνησε και αναπτύχθηκε κατά τη φάση της σχεδίασης του webinos. Αυτό καθώς αποδείχτηκε ότι μπορεί να λειτουργήσει ως μία

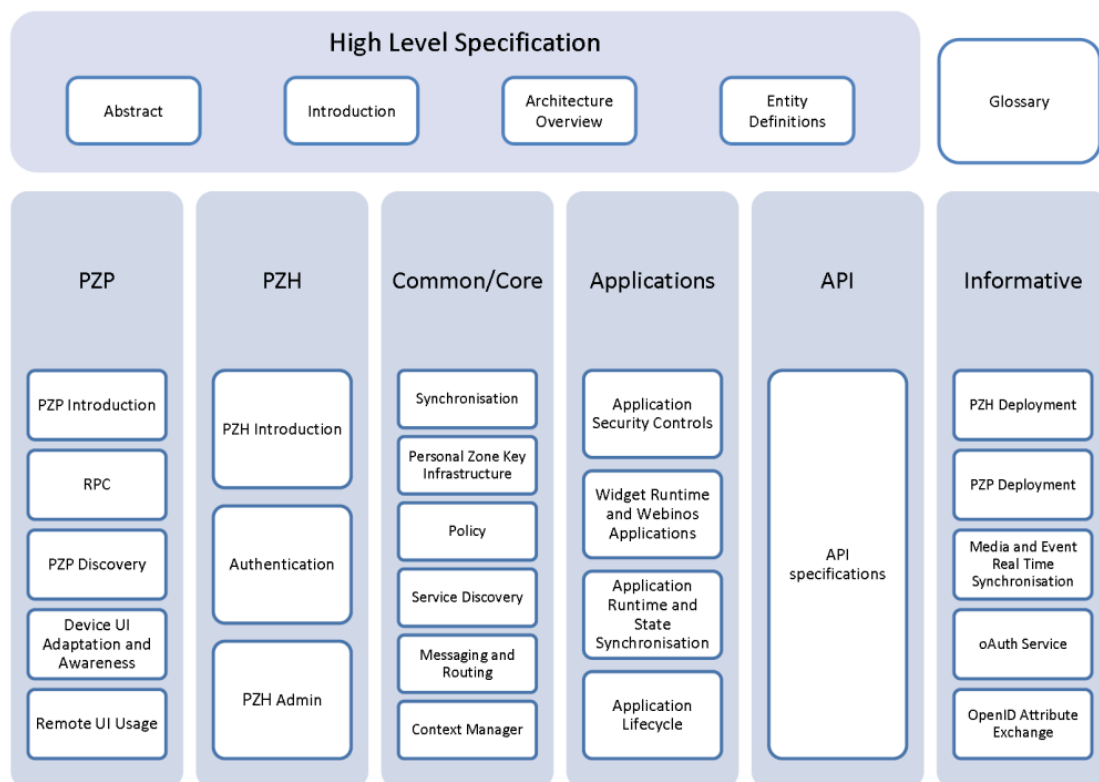
χρήσιμη παραδοχή που θα απλοποιεί τα πράγματα για τους σχεδιαστές εφαρμογών. Έτσι οι προγραμματιστές θα μπορούν πιο εύκολα να διαχειριστούν την πολυπλοκότητα που έχει ένα σύστημα εφαρμογών πολλαπλών συσκευών το οποίο προορίζεται για χρήση ανάμεσα σε διαφορετικές πλατφόρμες. Αρχικά, είχε υποτεθεί πως κάθε προσωπική ζώνη θα περιελάμβανε μία συσκευή και θα είχε μόνο έναν ιδιοκτήτη. Αποτέλεσμα της υπόθεσης αυτής ήταν οι προσωπικές ζώνες να θεωρούνται μικρά, απομονωμένα συστήματα συσκευών όπου μόνο ένας χρήστης ήταν παρών κάθε στιγμή. Αυτό βοήθησε στον αρχιτεκτονικό σχεδιασμό και στην ασφάλεια - η αυθεντικοποίηση μπορούσε να βασιστεί εν μέρει στις ταυτότητες των συσκευών - και έτσι προχώρησε το πρόγραμμα σημαντικά.

Έγινε όμως ξεκάθαρο πως πρόκειται για μία υπεραπλούστευση, αφού πολλές οικιακές συσκευές ανήκουν και χρησιμοποιούνται από διάφορους ανθρώπους από κοινού και δεν έχουν έναν μοναδικό ιδιοκτήτη (π.χ. ένας οικογενειακός προσωπικός υπολογιστής ή μία τηλεόραση). Αν γινόταν η υπόθεση πως αυτά είναι μέρη μίας ζώνης θα προκαλούνταν προβλήματα στις εφαρμογές και τις υπηρεσίες που θα απαιτούσαν χρησιμοποίηση ταυτοτήτων και αυθεντικοποιήσεων. Για τους παραπάνω λόγους, ο σχεδιασμός διαφοροποιήθηκε στη συνέχεια του προγράμματος, ώστε να επιτρέπει στις συσκευές να τρέχουν σε διαφορετικά πληρεξούσια (proxies). Δυστυχώς, η προσέγγιση αυτή απαιτεί τα πληρεξούσια που βρίσκονται στην ίδια συσκευή να είναι απομονωμένα το ένα από το άλλο σε διαφορετικούς λογαριασμούς χρηστών. Οι λογαριασμοί αυτοί έχουν δείξει πως έχουν χαμηλή χρηστικότητα. Για παράδειγμα σε μία οικιακή ρύθμιση, αυτό απαιτεί κάθε χρήστη να είναι συνδεδεμένος ώστε να μπορούν να τρέξουν και να δεχτούν τα πληρεξούσιά του απομακρυσμένες εντολές. Επίσης δεν είναι δυνατή έτσι, η λειτουργία του webinos σε συσκευές και συστήματα που δεν παρέχουν την δυνατότητα χρησιμοποίησής τους από πολλαπλούς λογαριασμούς. Έτσι οι προγραμματιστές του webinos κατέληξαν στη λύση των πολλαπλών πληρεξουσίων ανά συσκευή, η οποία είναι αποδεκτή προς το παρόν, αν και μελλοντικά απαιτεί περαιτέρω ανάπτυξη και μία καλύτερη υλοποίηση.

Από την παραπάνω συνειδητοποίηση φαίνεται ότι προέκυψαν συγκεκριμένες παραδοχές. Αρχικά, έγινε αντιληπτό ότι η έννοια του χρήστη και του λογαριασμού χρήστη αλλάζει και πολλές φορές η αοριστία μίας τέτοιας έννοιας δεν βοηθάει κατά την εύρεση τεχνικής λύσης, ειδικά όταν κλήθηκαν οι προγραμματιστές ανάπτυξης της πλατφόρμας ενδιάμεσου λογισμικού webinos να αντιμετωπίσουν διαφορετικούς τύπους συσκευών. Ακόμα, ότι η πρόωρη εισαγωγή μίας αόριστης έννοιας στο

σύστημα μπορεί να αποδειχτεί τόσο θετική όσο και αρνητική. Από τη μία μπορεί να βοηθήσει στην ανάπτυξη, αλλά από την άλλη μπορεί και να περιορίσει την ικανοποίηση των απαιτήσεων σε κάποιες από τις περιπτώσεις χρήσης, και αυτό με απρόβλεπτο τρόπο. Τέλος, έρευνες έχουν δείξει, πως οι προγραμματιστές που θέλουν να σχεδιάσουν μία εφαρμογή πάνω στο webinos, και οι οποίοι δεν είναι και τόσο εξοικειωμένοι με την πλατφόρμα, πιθανότατα να δυσκολευτούν ιδιαίτερα, δεδομένου πως η τεχνική υλοποίηση ενός πληρεξουσίου προσωπικής ζώνης και η κατάλληλη προσαρμογή του αποτελούν ένα δύσκολο πρόβλημα.

Τέλος, μια γενική εικόνα της αρχιτεκτονικής λειτουργίας του πλαισίου webinos γίνεται πιο εύκολα αντιληπτή μέσω της παρακάτω εικόνας όπου ξεχωρίζουμε τα PZP, PZH, κοινά στοιχεία, εφαρμογές και APIs. Κάθε μία από αυτές τις ενότητες ενσωματώνει συγκεκριμένα χαρακτηριστικά στοιχεία και μεθόδους συνδεσιμότητας με άλλες ενότητες.



Εικόνα 4 – Γενική εποπτική εικόνα της τεχνολογίας πλαισίου webinos

2.4 Τα APIs

Για την καλύτερη συμμόρφωση με τις τεχνικές προδιαγραφές της web κοινότητας αλλά και για την περαιτέρω διευκόλυνση των προγραμματιστών εφαρμογών όσον αφορά την πλατφόρμα ενδιάμεσου λογισμικού webinos, οι δυνατότητες των συσκευών αλλά και γενικά του πλαισίου παρέχονται με τη μορφή υπηρεσιών μέσω των APIs.

Η χρήση των κλήσεων απομακρυσμένης διαδικασίας (Remote Process Calls – RPCs) που θα πραγματοποιούνται από το webinos Runtime (WRT) στο εκτιθέμενο API άλλων συσκευών εντός της προσωπικής ζώνης, θα επιτρέπει στους προγραμματιστές να αυξάνουν το εύρος των λειτουργιών της εφαρμογής τους.

Μετά από έρευνες πάνω σε σενάρια, περιπτώσεις χρήσεων και απαιτήσεις συμφωνήθηκε ένα σύνολο από APIs τα οποία στη συνέχεια αναθεωρήθηκαν, συμπληρώθηκαν και καταργήθηκαν ώστε να φτάσουν στο πακέτο το οποίο παρέχεται αυτή τη στιγμή με πιθανές αναθεωρήσεις στο μέλλον.

Τα APIs όπως έχουν καθοριστεί μέχρι τον Σεπτέμβριο του 2012 και μπορούμε να μελετήσουμε περαιτέρω στην ιστοσελίδα του παραδοτέου <http://dev.webinos.org/deliverables/wp3/Deliverable34/> είναι εν συντομία τα κάτωθι:

➤ **The Generic Actuator API**

Το Generic Actuator API παρέχει εφαρμογές με ένα API για τον έλεγχο των ενεργοποιητών.

➤ **App2App Messaging API**

Η διαπροσωπεία για την ανταλλαγή μηνυμάτων μεταξύ των εφαρμογών, τοπικά ή / και εξ αποστάσεως.

➤ **AppState Synchronisation API**

Διαπροσωπεία για την ενεργοποίηση και διαχείριση του συγχρονισμού της εφαρμογής.

➤ **Authentication API**

Παρέχει πληροφορίες για τις αιτήσεις σχετικά με την τρέχουσα κατάσταση της ταυτότητας των χρηστών.

➤ **Contacts API**

Αυτό το API παρέχει πρόσβαση σε ένα ενιαίο βιβλίο διευθύνσεων του χρήστη.

➤ **Context API**

Το πλαίσιο αναφορά για την πλευρά του χρήστη (client-side).

➤ **Device Interaction API**

Το Device Interaction API επιτρέπει τον έλεγχο διαφόρων στοιχείων της συσκευής.

➤ **Device Status API**

Το webinos Device Status API επιτρέπει την λήψη πληροφοριών σχετικά με διάφορες "πτυχές" της συσκευής.

➤ **The Event Handling API**

Το webinos Event Handling API παρέχει τα μέσα για ανταλλαγή δεδομένων υπό την μορφή γεγονότων ανάμεσα σε οντότητες που διαθέτουν συγκεκριμένες διευθύνσεις, τοπικά ή απομακρυσμένα.

➤ **AppLauncher API**

Επιτρέπει την ενεργοποίηση των εφαρμογών του webinos που έχουν εγκατασταθεί τοπικά στη συσκευή.

➤ **MediaContent API**

Αυτό το API παρέχει πρόσβαση σε περιεχόμενο πολυμέσων και σχετικές πληροφορίες.

➤ **MediaPlay API**

Αυτό το API μπορεί να χρησιμοποιηθεί για την αναπαραγωγή αρχείων πολυμέσων σε ένα λογισμικό εξωτερικής συσκευής αναπαραγωγής πολυμέσων, αντί για αναπαραγωγή στο χρόνο εκτέλεσης του widget / browser.

➤ **Navigation API**

Το Navigation API του webinos παρέχει μηχανισμό για την αλληλεπίδραση με το ενσωματωμένο λογισμικό πλοήγησης.

➤ **NFC API**

Υποστήριξη του Near Field Communication (NFC).

➤ **The Web Notifications API**

Το Web Notifications API του webinos παρέχει τη δυνατότητα εμφάνισης ειδοποιήσεων προς τον χρήστη εκτός του πλαισίου μιας σελίδας.

➤ **Payment API**

Η διαπροσωπεία για τις λειτουργίες Πληρωμών.

➤ **The Remote UI API**

Το remoteUI API του webinos παρέχει τα μέσα για τη δημιουργία και πρόσβαση σε ένα UI σε μία απομακρυσμένη συσκευή χρησιμοποιώντας εντολές χειρισμού του DOM (Document Object Model) [12](#).

➤ **Secure Element API**

Το Secure Element API παρέχει στις διαδικτυακές εφαρμογές τη δυνατότητα να έχουν πρόσβαση σε ασφαλή δεδομένα.

➤ **The Generic Sensor API**

Το Generic Sensor API παρέχει στις εφαρμογές διαδικτύου ένα API για την πρόσβαση σε δεδομένα από τους αισθητήρες της ίδιας ή άλλης συσκευής.

➤ **Discovery API**

Το webinos Discovery API δίνει στις εφαρμογές τη δυνατότητα να ανακαλύπτουν υπηρεσίες χωρίς να υπάρχει προγενέστερη γνώση των υπηρεσιών.

➤ **TV Control API**

Διαπροσωπεία για τον έλεγχο και διαχείριση μιας τηλεόρασης.

➤ **Vehicle API**

Παρέχεται η πρόσβαση σε συγκεκριμένα δεδομένα αυτοκινήτου.

➤ **webinos core interface**

Κοινή διεπαφή μέσω της οποίας μπορούν οι εφαρμογές να έχουν πρόσβαση σε όλα τα webinos APIs και η οποία περιλαμβάνει επίσης πληροφορίες σχετικά με την Προσωπική Ζώνη του webinos.

➤ **webinos Widget API**

Καθορισμένες επεκτάσεις του webinos πάνω στο W3C Widget Interface [13](#).

3 Εφαρμογή Poker

- 3.1 Εισαγωγή - Αφορμή
- 3.2 Χαρακτηριστικά
- 3.3 SVG
- 3.4 Τα APIs που χρησιμοποιήθηκαν
- 3.5 Τρόπος Λειτουργίας με στοιχεία κώδικα
- 3.6 Παράδειγμα Ενδεικτικής Λειτουργίας

3.1 Εισαγωγή - Αφορμή

Η εφαρμογή webinos poker προέκυψε σαν ιδέα κατά την επαφή μου με την πλατφόρμα ενδιάμεσου λογισμικού webinos και την αναγνώριση της ανάγκης υλοποίησης εφαρμογών οι οποίες θα μπορούσαν να εκμεταλλευτούν πλήρως τις καινοτομίες αυτής της πλατφόρμας. Επιπλέον κίνητρο υπήρξε ο διαγωνισμός που διοργανώθηκε από το ίδιο το webinos consortium, το 1^ο webinos app challenge [14](#), στο οποίο η εφαρμογή διακρίθηκε στην 3^η θέση [15](#).

Το πόκερ είναι ένα από τα πιο διάσημα παιχνίδια τράπουλας με πολλούς παίκτες και φανατικούς σε όλο τον κόσμο. Στο παιχνίδι αυτό μπορούν να συμμετάσχουν από 2 έως 10 παίκτες σε ένα τραπέζι. Οι παραλλαγές του παιχνιδιού είναι πολλές αλλά το πιο δημοφιλές κι αυτό που χρησιμοποιήθηκε στην εφαρμογή είναι το Texas Hold'em [16](#) [17](#). Το εν λόγω παιχνίδι προσφέρεται για στρατηγική και μαθηματική ανάλυση. Πρόκειται για ένα παιχνίδι στοιχημάτων όπου οι παίκτες καλούνται όχι τόσο να κερδίσουν την εκάστοτε παρτίδα όσο να κάνουν τις μαθηματικά ορθότερες επιλογές. Πιθανοτικά, χρειάζεται βελτιστοποίηση των κερδών και ελαχιστοποίηση των απωλειών με βάση τα χαρτιά τα οποία έχουν ήδη παιχτεί και τα χαρτιά τα οποία μπορούν ακόμη να παιχτούν. Πριν ξεκινήσει ένα παιχνίδι έχει καθοριστεί το ελάχιστο ποσό πονταρίσματος.

Το παιχνίδι ξεκινάει μοιράζοντας 5 κλειστά φύλλα της τράπουλας πάνω στο τραπέζι και 2 σε κάθε παίκτη. Έχουμε 5 γύρους πονταρισμάτων. Στο 1^ο γύρο ισχύουν 2 υποχρεωτικά πονταρίσματα, τα γνωστά ως Blinds, 1 μικρό για τον πρώτο παίκτη στα αριστερά αυτού που μοιράζει και 1 μεγάλο για τον δεύτερο στα αριστερά. Οι υπόλοιποι παίκτες καλούνται να εξισώσουν το ποντάρισμα, να το αυξήσουν ή να πάνε πάσο. Εάν κάποιος πάει πάσο, χάνει κάθε δικαίωμα στο συγκεκριμένο γύρο παιχνιδιού και "καίγονται" τα φύλλα του. Ο γύρος πονταρισμάτων τελειώνει όταν όλοι οι εναπομείναντες παίκτες έχουν εξισώσει το ποντάρισμα τους. Το ποντάρισμα του καθενός τότε αθροίζεται στο κέντρο (pot) και γυρνάει 3 από τα κοινά φύλλα του τραπέζιου. Ακολουθεί ξανά γύρος πονταρισμάτων με τον ίδιο τρόπο. Γυρνάει 1 φύλλο του τραπέζιου, νέος γύρος πονταρισμάτων, άλλο 1 φύλλο αποκαλύπτεται και ακολουθεί ο τελικός γύρος πονταρισμάτων. Πλέον όσοι μένουν ακόμη στο τραπέζι ξέρουν όλους τους πιθανούς συνδυασμούς που μπορούν να γίνουν με τα κοινά

φύλλα και οι πιθανότητες έχουν μειωθεί αρκετά καθώς περισσότερα στοιχεία έχουν γίνει γνωστά.

Ο συνδυασμός που ψάχνουμε αποτελείται από 5 φύλλα συνολικά, τα καλύτερα από τα 5 κοινά και τα 2 που έχει ο κάθε παίκτης στο χέρι του. Οι συνδυασμοί ακολουθούν μια συγκεκριμένη ιεραρχία. Παραθέτουμε αυτή την ιεραρχία από τον πιο δυνατό συνδυασμό ως τον πιο απλό:

1. **Royal Flush** (Φλος Ρουαγιάλ) - Δέκα Βαλές Ντάμα Ρήγας Άσος του ίδιου χρώματος
2. **Straight Flush** (Φλος) - Πέντε συνεχόμενα φύλλα του ίδιου χρώματος
3. **Τέσσερα Όμοια** (Καρέ) - Τέσσερα όμοια φύλλα
4. **Full House** (Φουλ) - Τρία όμοια φύλλα και επιπλέον 2 όμοια φύλλα. Για παράδειγμα Πέντε, Πέντε, Πέντε και Δύο, Δύο
5. **Flush** (Χρώμα) - Πέντε οποιαδήποτε φύλλα του ίδιου χρώματος
6. **Straight** (Κέντα) - Πέντε συνεχόμενα φύλλα ανεξαρτήτου χρώματος
7. **Τρία Όμοια** - Τρία όμοια φύλλα
8. **Δύο ζευγάρια** - Δύο ζευγάρια όμοιων φύλλων, για παράδειγμα Δέκα, Δέκα και Πέντε, Πέντε
9. **Ένα ζευγάρι** - Δύο όμοια φύλλα

Προφανώς, εάν δυο παίκτες έχουν τον ίδιο συνδυασμό, δυνατότερος είναι αυτός με το υψηλότερο φύλλο. Υψηλότερο φύλλο θεωρείται ο Άσος και ακολουθούν Ρήγας, Ντάμα, Βαλές, Δέκα μέχρι Δύο. Αν για παράδειγμα έχουμε δυο παίκτες με Φουλ και οι δύο, ο ένας έχει: Πέντε, Πέντε, Πέντε και Τρία, Τρία και ο δεύτερος: Έξι, Έξι, Έξι και Δύο, Δύο κερδίζει ο δεύτερος. Αν τα φύλλα ήταν με την ακόλουθη μορφή: ο πρώτος: Τρία, Τρία, Τρία και Πέντε, Πέντε και ο δεύτερος: Δύο, Δύο, Δύο και Έξι, Έξι κερδίζει ο πρώτος.

Οι πιθανότητες να συμβούν με μία τράπουλα οι παραπάνω συνδυασμοί εμφανίζονται στον ακόλουθο πίνακα:

Πίνακας 1 – Πιθανότητες συνδυασμών στο παιχνίδι Πόκερ Texas Hold'em

Συνδυασμοί	Ερμηνεία	Τρόποι	Πιθανότητες
Ζεύγος / Ζευγάρι	Δύο φύλλα ίδιας αξίας	1.098.240	35 προς 24
Δύο (ή διπλά) ζευγάρια	Δύο ζευγάρια	123.552	20 προς 1

Τριάδα	Τρία φύλλα ίδιας αξίας	54.912	46 προς 1
Κέντα	Πέντε συνεχόμενα	10.200	254 προς 1
Χρώμα	Πέντε ασυνεχή ίδιας σειράς	5.108	508 προς 1
Φουλ	Τρία όμοια και ένα ζευγάρι	3.744	693 προς 1
Καρέ	Τέσσερα ίδια	624	4.164 προς 1
Κέντα Χρώμα ή Φλος	Πέντε συνεχόμενα ίδιας σειράς	36	72.192 προς 1
Φλος Ρουαγιάλ	Φλος με τα πέντε μεγαλύτερης αξίας φύλλα (10-J-Q-K-A)	4	649.739 προς 1

Μετά τον τελευταίο γύρο πονταρισμάτων οι παίκτες που έχουν απομείνει και δεν έχουν πάει πάσο, αποκαλύπτουν τα φύλλα τους και προκύπτει έτσι ο νικητής με βάση τους παραπάνω συνδυασμούς. Το συνολικό ποσό (pot) το οποίο έχει συσσωρευτεί δίνεται στον νικητή και όσοι παίκτες επιθυμούν συνεχίζουν σε καινούριο γύρο παιχνιδιού.

Με γνώμονα αυτούς τους κανόνες και τους λόγους που εξηγήθηκαν στην αρχή αυτού του κεφαλαίου δημιουργήθηκε το παιχνίδι webinos poker το οποίο μπορούν να χρησιμοποιήσουν παίκτες χωρίς την ανάγκη φυσικής τράπουλας σε ένα ευχάριστο περιβάλλον, μέσα από πολλαπλές συσκευές, ακόμη και απομακρυσμένα, χωρίς κανένα απολύτως πρόβλημα.

3.2 Χαρακτηριστικά

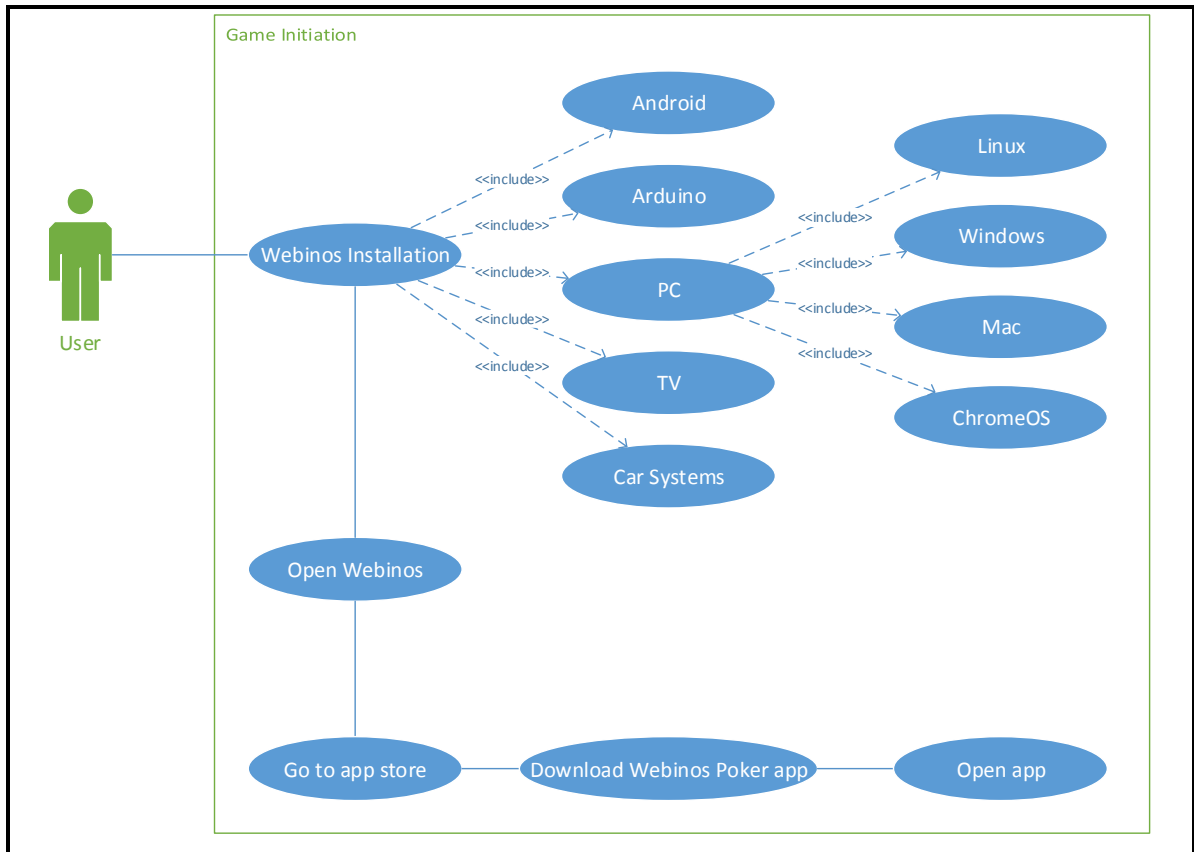
Για την εφαρμογή webinos poker χρησιμοποιήθηκαν μια σειρά από τεχνολογίες, αρχιτεκτονικές και συγκεκριμένη μοντελοποίηση ώστε να δημιουργηθεί η βέλτιστη δυνατή εμπειρία για τον χρήστη αλλά και τον προγραμματιστή που θα ήθελε να ασχοληθεί με τον παραχθέντα κώδικα.

Για την καλύτερη δυνατή τελική λύση κρίθηκε απαραίτητο να ακολουθηθεί συγκεκριμένη μηχανική απαιτήσεων κατά την οποία έγινε πρώτα η ανάλυση κι έπειτα η οριστικοποίηση των προδιαγραφών. Ακολουθήθηκε το μοντέλο του καταρράκτη (waterfall model) το οποίο χαρακτηρίζεται από σειριακά βήματα – φάσεις, με ανάδραση μεταξύ δύο γειτονικών βημάτων. Το μοντέλο του καταρράκτη βασίζεται στην δημιουργία προδιαγραφών σε κάθε βήμα. Έτσι ξεκινάμε παράγοντας αρχικά τις προδιαγραφές και απαιτήσεις του συστήματος.

Για να μπορέσει να γίνει αυτό χρησιμοποιήθηκε μοντελοποίηση των βασικών σεναρίων χρήσης (Use Case Diagrams) του προγράμματος, των διαγραμμάτων Ενεργειών (Activity Diagrams) και στη συνέχεια παράλληλα με τη συγγραφή του κώδικα έγινε αρχικοποίηση και συνεχής ενημέρωση διαγραμμάτων Κλάσεων (Class Diagrams).

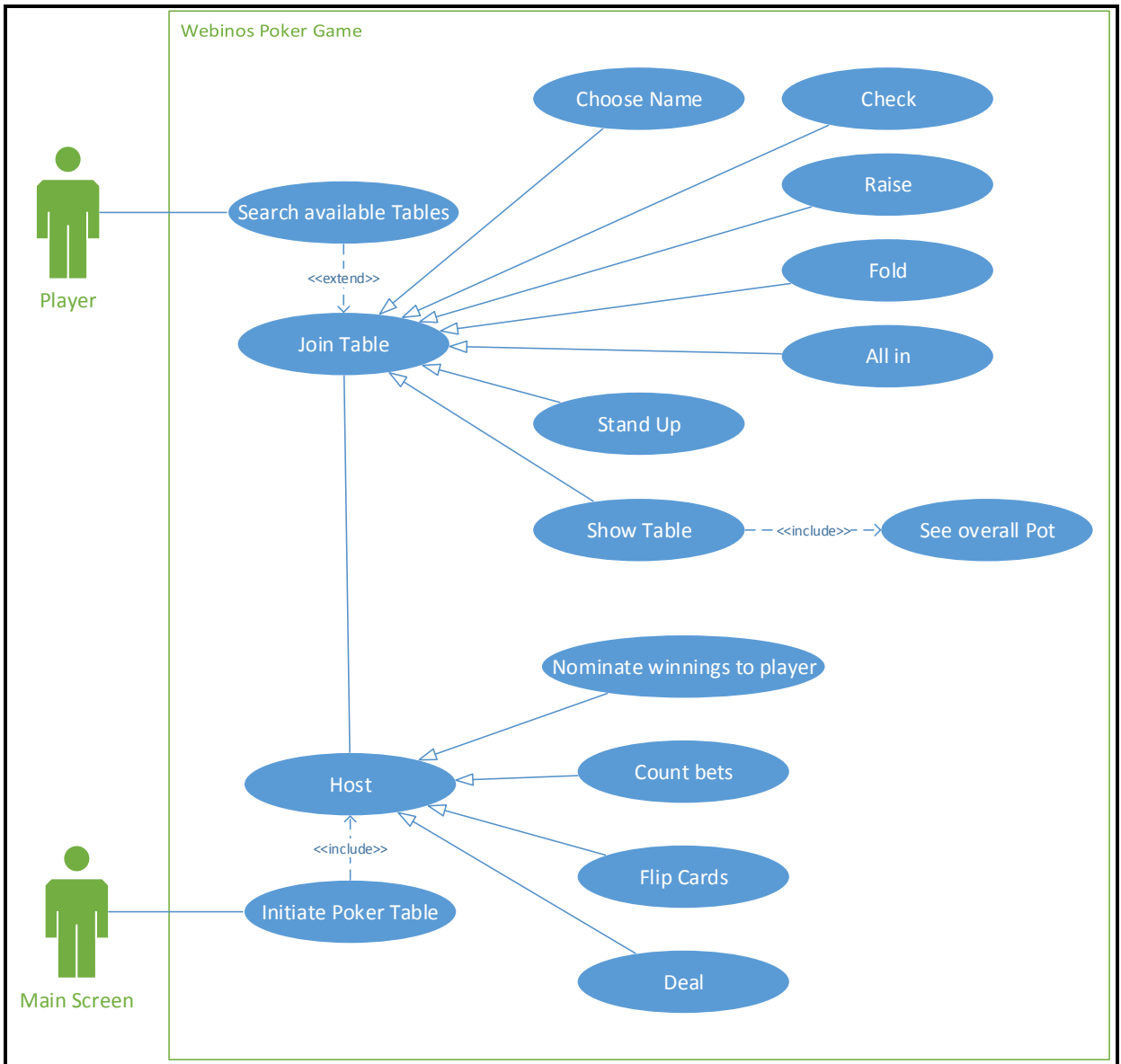
Και τα τρία παραπάνω είδη διαγραμμάτων αποτελούν αναπαραστήσεις των δύο υποκατηγοριών της Ενοποιημένης Γλώσσας Μοντελοποίησης (UML) [18](#). Η UML είναι μια προτυποποιημένη γενικού σκοπού γλώσσα μοντελοποίησης στον τομέα της τεχνολογίας λογισμικού. Η μοντελοποίηση βοηθάει πολύ κατά την διαδικασία της συγγραφής του κώδικα μιας εφαρμογής, ιδίως σε αρχικό επίπεδο, καθώς απαιτείται να εξαντλήσεις όλα τα πιθανά σενάρια χρήσης πριν ακόμη προχωρήσεις στην υλοποίηση της εφαρμογής και να αποκτήσεις μια βασική εικόνα των διαδικασιών του κώδικα σου κατά την υλοποίηση της.

Το Use Case Diagram το οποίο υλοποιήθηκε στην αρχή, παρουσιάζει τις πιθανές αλληλεπιδράσεις των χρηστών με την εφαρμογή. Υπάρχουν δύο βασικοί τύποι χρηστών, οι παίκτες με τις ιδιωτικές πληροφορίες τους και η κεντρική οθόνη παρουσίασης κοινών δεδομένων. Έτσι το use case diagram έχει ως εξής:



Εικόνα 5 - Use Case Scenario – Εγκατάσταση Εφαρμογής

Σε αυτό το διάγραμμα όπως γίνεται αντιληπτό παρουσιάζουμε το σενάριο εγκατάστασης της εφαρμογής μας με το προαπαιτούμενο βήμα της εγκατάστασης του πλαισίου webinos στη συσκευή του χρήστη. Οι συσκευές μπορεί να είναι κάποια κινητή έξυπνη συσκευή με λειτουργικό σύστημα android (smartphone, tablet), ένα σύστημα Arduino, προσωπικός υπολογιστής με κάλυψη για πιο ευρέως διαδεδομένα λειτουργικά συστήματα, έξυπνο σύστημα τηλεόρασης ή σύστημα αυτοκινήτου το οποίο να υποστηρίζει την παραπάνω διαδικασία. Τα ακόλουθα βήματα είναι με τη σειρά: Αρχικοποίηση του ενδιάμεσου λογισμικού webinos, πλοήγηση εντός της εφαρμογής στο κατάστημα τελικών εφαρμογών, επιλογή, κατέβαση και εγκατάσταση της εφαρμογής webinos Poker και άνοιγμα της εφαρμογής.



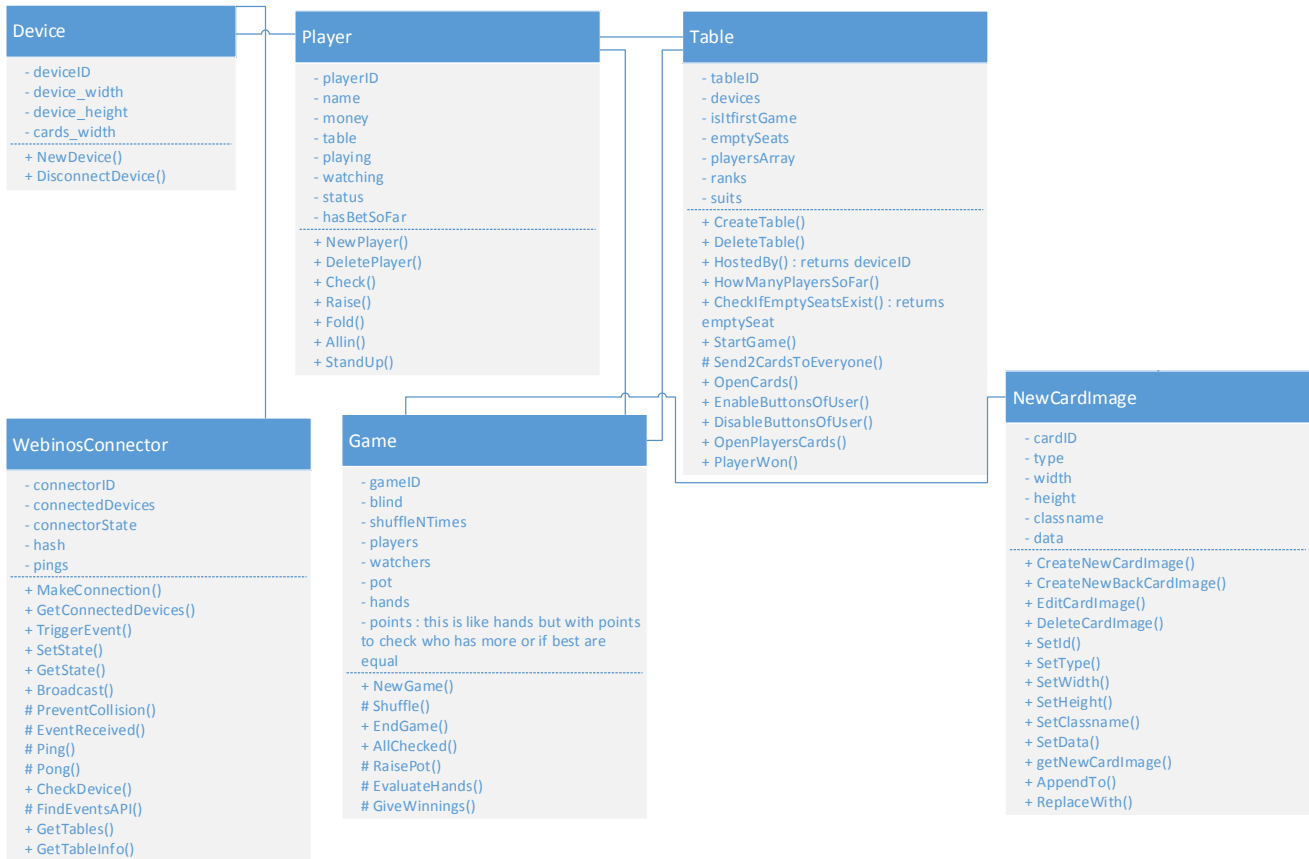
Εικόνα 6 - Use Case Scenario - Παιχνίδι

Απαραίτητη προϋπόθεση που υπονοείται εδώ είναι ότι έχουμε ακολουθήσει πλήρως την διαδικασία εγκατάστασης του πλαισίου webinos, όπως περιγράφεται παραπάνω.

Η λογική του παιχνιδιού όπως μπορεί να φανεί και στην Εικόνα 2 είναι ότι μία κεντρική οθόνη εκτελεί καθήκοντα διαμοιραστή της τράπουλας στους χρήστες που συνδέονται σε ένα συγκεκριμένο τραπέζι. Αυτό το κεντρικό τραπέζι διατηρεί την πληροφορία όσον αφορά όλη την τράπουλα και τα χαρτιά τα οποία κρατάει στο χέρι του ο κάθε παίκτης. Με αυτό τον τρόπο διατηρείται η ασφάλεια καθώς κανένας παίκτης δεν έχει τη δυνατότητα να γνωρίζει τα υπόλοιπα φύλλα, αυτή η γνώση υπάρχει μόνο στο κεντρικό τραπέζι το οποίο δεν είναι παίκτης.

Εν συνεχεία, αφού εξήχθησαν τα πρώτα σενάρια χρήσης της εφαρμογής έπρεπε να περάσουμε στον σχεδιασμό του κώδικα. Για να γίνει σωστή ενσωμάτωση του κώδικα πλαισίου με της εφαρμογής χρησιμοποιήθηκε κατά κύριο λόγο η γλώσσα προγραμματισμού JavaScript η οποία ευνοεί την χρήση αντικειμενοστραφούς προγραμματισμού. Στον αντικειμενοστραφή προγραμματισμό ένα σύστημα ορίζεται από μια συλλογή αντικειμένων τα οποία αλληλοεπιδρούν το ένα με το άλλο και με αντικείμενα εξωτερικά προς το σύστημα. Τα αντικείμενα στέλνουν μηνύματα, και αντιδρούν σε εισερχόμενα μηνύματα. Τα πλεονεκτήματα του αντικειμενοστραφούς προγραμματισμού είναι πολλαπλά και προτείνεται στον αναγνώστη να τα ανακαλύψει στις αντίστοιχες πηγές [19](#).

Προτού γραφτεί ο κώδικας κρίθηκε σκόπιμο να φτιαχτεί ένα μοντέλο UML Class Diagram όπου θα σκιαγραφούσαμε την γενική μορφή των συνιστωσών της εφαρμογής με πιο συγκεκριμένο τρόπο. Αυτή η UML μοντελοποίηση φαίνεται στην παρακάτω εικόνα:



Εικόνα 7 – UML Class Diagram

Το διάγραμμα αυτό βοήθησε πολύ στην αρχική σχεδίαση του κώδικα αν και δεν ήταν πλήρως δεσμευτικό. Πολλές αλλαγές έγιναν στην πορεία και η τελική μορφή του κώδικα είναι ελαφρώς αλλαγμένη δυναμικά.

Πιο συγκεκριμένα, τα στοιχεία της εφαρμογή αποτελούνταν από τους παρακάτω πίνακες χωρίς τις μεταξύ τους συσχετίσεις:

Πίνακας 2 – Classes - Device

Ονομασία	Device	
Μεταβλητές	Ιδιωτική	deviceId
	Ιδιωτική	deviceWidth
	Ιδιωτική	deviceHeight
	Ιδιωτική	cardsWidth
Μέθοδοι	Δημόσια	NewDevice()
	Δημόσια	DisconnectDevice()

Πίνακας 3 – Classes - Player

Ονομασία	Player	
Μεταβλητές	Ιδιωτική	playerID
	Ιδιωτική	name
	Ιδιωτική	money
	Ιδιωτική	table
	Ιδιωτική	playing
	Ιδιωτική	watching
	Ιδιωτική	status
	Ιδιωτική	hasBetSoFar
Μέθοδοι	Δημόσια	NewPlayer()
	Δημόσια	DeletePlayer()

	Δημόσια	Check()
	Δημόσια	Raise()
	Δημόσια	Fold()
	Δημόσια	Allin()
	Δημόσια	StandUp()

Πίνακας 4 – Classes - Table

Όνομασία	Table	
Μεταβλητές	Ιδιωτική	tableID
	Ιδιωτική	devices
	Ιδιωτική	isItFirstGame
	Ιδιωτική	emptySeats
	Ιδιωτική	playersArray
	Ιδιωτική	ranks
	Ιδιωτική	suits
Μέθοδοι	Δημόσια	CreateTable()
	Δημόσια	DeleteTable()
	Δημόσια	HostedBy()
	Δημόσια	HowManyPlayersSoFar()
	Δημόσια	CheckIfEmptySeatExists()
	Δημόσια	StartGame()

	Προστατευόμενη	Send2CardsToEveryone()
	Δημόσια	OpenCards()
	Δημόσια	EnableButtonsOfUser()
	Δημόσια	DisableButtonsOfUser()
	Δημόσια	OpenPlayersCards()
	Δημόσια	PlayerWon()

Πίνακας 5 – Classes - NewCardImage

Όνομασία	NewCardImage	
Μεταβλητές	Ιδιωτική	cardID
	Ιδιωτική	type
	Ιδιωτική	width
	Ιδιωτική	height
	Ιδιωτική	classname
	Ιδιωτική	data
Μέθοδοι	Δημόσια	CreateNewCardImage()
	Δημόσια	CreateNewBackCardImage ()
	Δημόσια	EditCardImage()
	Δημόσια	DeleteCardImage()
	Δημόσια	SetId()
	Δημόσια	SetType()

	Δημόσια	SetWidth()
	Δημόσια	SetHeight()
	Δημόσια	SetClassname()
	Δημόσια	SetData()
	Δημόσια	getNewCardImage()
	Δημόσια	AppendTo()

Πίνακας 6 – Classes - Game

Ονομασία	Game	
Μεταβλητές	Ιδιωτική	gameID
	Ιδιωτική	blind
	Ιδιωτική	shuffleNTimes
	Ιδιωτική	players
	Ιδιωτική	watchers
	Ιδιωτική	pot
	Ιδιωτική	hands
	Ιδιωτική	points
Μέθοδοι	Δημόσια	NewGame()
	Προστατευόμενη	Shuffle()
	Δημόσια	EndGame()
	Δημόσια	AllChecked()

	Προστατευόμενη	RaisePot()
	Προστατευόμενη	EvaluateHands()
	Προστατευόμενη	GiveWinnings()

Πίνακας 7 – Classes - WebinosConnector

Όνομασία	WebinosConnector	
Μεταβλητές	Ιδιωτική	connectorID
	Ιδιωτική	connectedDevices
	Ιδιωτική	connectorState
	Ιδιωτική	hash
	Ιδιωτική	pings
Μέθοδοι	Δημόσια	MakeConnection()
	Δημόσια	GetConnectedDevices()
	Δημόσια	TriggerEvent()
	Δημόσια	SetState()
	Δημόσια	GetState()
	Δημόσια	Broadcast()
	Προστατευόμενη	PreventCollision()
	Προστατευόμενη	EventReceived()
	Προστατευόμενη	Ping()
	Προστατευόμενη	Pong()

	Δημόσια	CheckDevice()
	Προστατευόμενη	FindEventsAPI()
	Δημόσια	GetTables()
	Δημόσια	GetTableInfo()

Μετά τον σχεδιασμό του κώδικα προχωρήσαμε στην υλοποίηση και ενσωμάτωση του. Αυτό το κομμάτι αποτέλεσε το μεγαλύτερο και δυσκολότερο σημείο κατά την συνολική υλοποίηση της εφαρμογής.

Κατά τον προγραμματισμό του κώδικα πάρθηκαν δυναμικά κάποιες αποφάσεις για την μεγιστοποίηση της ποιότητας αλλά και την βελτίωση της τελικής χρήσης της εφαρμογής. Ένα παράδειγμα είναι η επιλογή του προγραμματισμού των εικονικών μερών της εφαρμογής στη γλώσσα SVG και όχι χρησιμοποιώντας κάποια άλλη μορφή όπως jpeg, png, κλπ.

Μετά το προγραμματισμό της εφαρμογής (αλλά και κατά τη διάρκεια του, σε μικρότερο βαθμό) πραγματοποιήθηκε έλεγχος της με βάση όλα τα πιθανά σενάρια ώστε να είναι σίγουρο ότι έχει δημιουργηθεί η καλύτερη δυνατή λύση για αυτό που σχεδιάστηκε αρχικά.

Μετά το κομμάτι του ελέγχου ακολούθησε φυσικά η συντήρηση του κώδικα για αποσφαλμάτωση όπου χρειαζόταν και η πιθανή δημιουργία νέων στοιχείων επιπρόσθετων στην αρχική σχεδίαση των σεναρίων χρήσης για την εφαρμογή. Με βάση όλα τα παραπάνω δημιουργείται η πεποίθηση ότι έγινε από την πλευρά του προγραμματιστή η απαραίτητη προετοιμασία για την καλύτερη αντιμετώπιση όλων των προκλήσεων που μπορεί να εμφανίζονταν σε σχέση με τον προγραμματισμό της εφαρμογής.

Ο πηγαίος κώδικας παρέχεται με τη μορφή του ανοιχτού κώδικα (open source) ²⁰ ελεύθερα στο διαδίκτυο, στο [Github](#), μια υπηρεσία φιλοξενίας κώδικα για έργα ανάπτυξης λογισμικού το οποίο χρησιμοποιεί το σύστημα ελέγχου αναθεωρήσεων (revision control system) git, ώστε ο οποιοσδήποτε το επιθυμεί να κατανοήσει εις βάθος την τεχνολογία αλλά και να έχει την δυνατότητα παραμετροποίησης του

παρεχόμενου αρχικού κώδικα. Ο κώδικας της εφαρμογής βρίσκεται στη διεύθυνση <https://github.com/romdim/Webinos-Poker>.

3.3 SVG

Η γενικότερη αισθητική και εμπειρία του χρήστη – παίκτη (UX – User Experience) ήταν μία βασικότερη παράμετρος κατά τον σχεδιασμό και υλοποίηση της εφαρμογής webinos poker. Με γνώμονα αυτό λοιπόν πάρθηκε η απόφαση να χρησιμοποιηθεί η τεχνολογία SVG (Scalable Vector Graphics) για την απεικόνιση των γραφικών στοιχείων. Η απόφαση αυτή προέκυψε μετά από πολλή σκέψη επάνω στις απαιτήσεις της εφαρμογής, τις δυνατότητες και τα μειονεκτήματα των διάφορων τεχνολογιών.

Η τεχνολογία SVG είναι ένας τύπος απεικόνισης δισδιάστατων στοιχείων με διανύσματα σε XML (Extensible Markup Language) μορφή. Το γεγονός ότι η μορφή του κώδικα των SVG αρχείων στηρίζεται στο XML, το κάνει ιδανικό για διαδικτυακές εφαρμογές και συνάδει με τα πρότυπα της κοινότητας. Αυτή του η ιδιότητα σημαίνει ότι έχει τη δυνατότητα αναζήτησης εντός του αρχείου (searchability), τη δυνατότητα δημιουργίας δεικτών για γρηγορότερο εντοπισμό στοιχείων (indexing), μπορεί να παραμετροποιηθεί με μεγάλη ευκολία και εάν χρειαστεί να συμπιεστεί. Τα SVG αρχεία, ως XML που είναι, μπορούν να αναγνωστούν και να αλλαχτούν εύκολα και απλά σε έναν συντάκτη κειμένου.

Όλοι οι μοντέρνοι φυλλομετρητές ιστού όπως είναι οι Mozilla Firefox, Google Chrome, Internet Explorer, Opera και Safari υποστηρίζουν την τεχνολογία SVG και μπορούν να αποδώσουν τον κώδικα άμεσα όπως κάνουν με τα αρχεία HTML και CSS. Όπως αναφέρθηκε νωρίτερα το SVG αποτελείται από διανύσματα τα οποία συνθέτουν μία εικόνα. Το γεγονός αυτό είναι το πιο σημαντικό πλεονέκτημα αυτής της τεχνολογίας καθώς επιτρέπουν στους φυλλομετρητές να προσαρμόσουν το μέγεθος της εικόνας σύμφωνα με τα δεδομένα που έχουν χωρίς να χάνουν καθόλου την ποιότητα της εικόνας σε αντίθεση με όλους τους υπόλοιπους τύπους αρχείων.

Στην δική μας περίπτωση όπου δημιουργείται εφαρμογή κατάλληλη για όλες τις συσκευές επιτυγχάνουμε τη μέγιστη δυνατή απόδοση και βέλτιστη εμπειρία για τον χρήστη με δεδομένο ότι ανεξάρτητα τη συσκευή του μπορούμε να απεικονίσουμε με μεγάλη ακρίβεια όλες τις κάρτες, τις μάρκες και το τραπέζι της εφαρμογής χωρίς να παρουσιάζεται η γνωστή παραμόρφωση των εικονοστοιχείων (pixels).

Αντίστοιχα υποστηρίζονται όλες οι κινήσεις των εικόνων (animation – translation) όπως ορίζονται στη γλώσσα CSS (Cascading Style Sheets) και πιο συγκεκριμένα με την προτυποποίηση που διατυπώθηκε από το W3C (World Wide Web Consortium) για την έκδοση 3 του CSS. Αντίστοιχα πρόκειται για τη μόνη ανοιχτού λογισμικού γλώσσα αυτή τη στιγμή μαζί με το στοιχείο HTML Canvas του HTML 5 με τη δυνατότητα δυναμικών αλλαγών κατά την αποτύπωση στον φυλλομετρητή.

Οι εναλλακτικές λύσεις για την αποτύπωση εικόνων είναι οι τεχνολογίες:

- JPEG
- TIFF
- PSD
- PDD
- BMP
- PICT
- PNG
- GIF

Το jpeg format (Joint Photographic Experts Group) συνηθίζεται στο διαδίκτυο καθώς χρησιμοποιεί καλές μεθόδους συμπίεσης και μπορεί να δημιουργήσει πολύ μικρά σε μέγεθος αρχεία. Παλιότερα ήταν ευρέως χρησιμοποιούμενο ακριβώς για τον λόγο ότι η ταχύτητα συνδεσιμότητας των χρηστών ήταν περιορισμένη. Βασικά μειονεκτήματα είναι ότι δεν είναι κατάλληλο για εικόνες με κείμενο, μεγάλα κομμάτια χρώματος ή απλά σχήματα γιατί δημιουργείται φαινόμενο θολώματος της εικόνας και τα χρώματα μπορεί να αλλάξουν.

Τα tiff αρχεία (Tagged Image File Format) είναι bitmap εικόνες χωρίς απώλεια ποιότητας και υποστήριξη διαφάνειας αλλά μεγάλα σε μέγεθος και άκαμπτα. Αντίστοιχα με τα tiff αρχεία είναι τα bmp (bitmap image file), τα οποία όμως φτάνουν ακόμη μεγαλύτερο μέγεθος. Τα pict είναι bitmap αρχεία τα οποία χρησιμοποιούνταν στους υπολογιστές Mac και πλέον δεν βρίσκονται συχνά σε χρήση.

Τα psd (Photoshop Document), pdd (Alternative file extension for Portable Document Format) αρχεία είναι παράγωγα του εργαλείου Photoshop όπου διατηρούνται οι βασικές ιδιότητες της εικόνας.

Οι png εικόνες (Portable Network Graphics) είναι ευρέως διαδεδομένες στο

διαδίκτυο διότι πρόκειται για μικρού μεγέθους, χωρίς απώλεια στην ποιότητα και με υποστήριξη διαφάνειας εικόνες. Αποτελούν την τη γραφική αντικατάσταση των gif αρχείων για το διαδίκτυο.

Τα gif αρχεία (Graphics Interchange Format) χρησιμοποιούνται για δημιουργία απλών γραφικών με περιορισμένη χρωματική παλέτα, 256 χρωμάτων μόνο. Πρόκειται για μικρού μεγέθους εικόνες, οι οποίες αποτυπώνονται γρήγορα στους φυλλομετρητές. Είναι πολύ συνηθισμένη μάλιστα η χρήση της μορφής αυτής για δημιουργία κινούμενων εικόνων.

Όλοι οι προαναφερθέντες τύποι αρχείων έχουν το βασικό μειονέκτημα της παραμόρφωσης κατά την αλλαγή μεγέθους μιας εικόνας σε αντίθεση με το επιλεγμένο format.

Για να απεικονιστούν λοιπόν οι κάρτες της τράπουλας, από τον άσσο μέχρι τον ρήγα, τα τέσσερα διαφορετικά χρώματα και οι πίσω πλευρές της τράπουλας χρησιμοποιήθηκε η Βιβλιοθήκη Γραφικών “Vectorized Playing Cards 1.3”, ένα ελεύθερο λογισμικό με άδεια επαναχρησιμοποίησης. Τα αρχεία των εικόνων υπέστησαν επεξεργασία και συμπίεση ώστε να ταιριάζουν περισσότερο με την λογική της εφαρμογής.

3.4 Τα APIs που χρησιμοποιήθηκαν

Η εφαρμογή webinos Poker χρησιμοποίησε συγκεκριμένες υπηρεσίες API από την πλατφόρμα ενδιάμεσου λογισμικού webinos.

Το webinos core interface είναι η κεντρική διαπροσωπεία που καλείται από όλα τα υπόλοιπα APIs με βάση την αντικειμενοστρέφεια και υλοποιεί βασικές μεθόδους. Πρόκειται για ένα κομμάτι του καθολικού αντικειμένου του παραθύρου. Παρέχει επίσης σημαντικές πληροφορίες της προσωπικής ζώνης που αφορούν την συνδεσιμότητα μιας συσκευής, το όνομα του rzh, ένα φιλικό όνομα για το rzh και ένα αναγνωριστικό εφαρμογής.

Χρησιμοποιήθηκε το webinos Discovery API, το οποίο έχει ως κύριο σκοπό την παροχή εφαρμογών για την εξεύρεση τοπικών ή απομακρυσμένων υπηρεσιών για τις οποίες δεν υπάρχει πρότερη γνώση. Αυτές οι υπηρεσίες μπορεί να βρίσκονται μέσα στην ίδια τη συσκευή, σε οντότητες οι οποίες συνδέονται άμεσα με τη συσκευή, σε οντότητες που υπάρχουν στο ίδιο IP δίκτυο ή υπηρεσίες που είναι εγγεγραμμένες μέσα σε μία προσωπική ζώνη από άλλες έμπιστες υπηρεσίες.

Η πιο χρήσιμη όμως υπηρεσία με την οποία λειτούργησε κατά κύριο λόγο η εφαρμογή είναι το Events Handling API, το οποίο μας δίνει τη δυνατότητα να ανταλλάσσουμε δεδομένα κατά την εφαρμογή κάποιον συγκεκριμένων γεγονότων.

Για να γίνει το παραπάνω αντιληπτό μπορούμε να φέρουμε ως παράδειγμα την περίπτωση όπου ένας χρήστης – παίκτης αποφασίζει κατά τη διάρκεια ενός γύρου στοιχημάτων μέσα σε ένα παιχνίδι με περισσότερους παίκτες που κάθονται σε ένα τραπέζι, ότι θέλει να αυξήσει το ποσό στοιχήματος κατά ένα συγκεκριμένο νούμερο. Αυτό σημαίνει πρακτικά ότι μόλις έχει έρθει η σειρά του παίκτη πατιέται το αντίστοιχο κουμπί με τον μετρητή να δείχνει το αντίστοιχο νούμερο. Ως συνέπεια αυτού του γεγονότος στέλνεται ένα μήνυμα σε όλες τις συσκευές και αυτή που πρέπει, δηλαδή το τραπέζι εν προκειμένω αξιοποιεί την πληροφορία αυξάνοντας το απαιτούμενο ποσό στοιχήματος για τους υπόλοιπους παίκτες, ενημερώνεται η μεταβλητή με την πληροφορία για το πότε τελειώνει ο συγκεκριμένος γύρος στοιχημάτων (πριν από τον τωρινό παίκτη δηλαδή) και περνάμε άμεσα στον επόμενο παίκτη. Εδώ στέλνεται ένα νέο μήνυμα από το τραπέζι πλέον με το όνομα

του παίκτη που συνεχίζει την παρτίδα, το ελάχιστο ποσό στοιχήματος και ό,τι άλλη πληροφορία χρειαζόμαστε. Με βάση αυτό το broadcast πληροφορίας η συσκευή του πρώτου παίκτη απενεργοποιεί τα εμφανιζόμενα κουμπιά καθώς πέρασε η σειρά του και ενεργοποιούνται τα κουμπιά του επόμενου στη σειρά. Οι υπόλοιποι παίκτες αγνοούν το μήνυμα ή κρατάνε απλά την πληροφορία ότι πέρασε το παιχνίδι στον επόμενο παίκτη.

Η παραπάνω μετάδοση πληροφοριών είναι ζωτική για την εφαρμογή και δεν θα μπορούσε να είναι δυνατή χωρίς τη χρήση ενός κοινού πλαισίου ανταλλαγής τέτοιων μηνυμάτων όπως είναι το webinos.

3.5 Τρόπος Λειτουργίας με στοιχεία κώδικα

Κατά την εγκατάσταση της εφαρμογής στην πλατφόρμα ενδιάμεσου λογισμικού webinos μίας συσκευής παρέχεται όλος ο κώδικας που χρειάζεται σε αυτή. Μία συσκευή θα μπορούσε να λειτουργήσει ταυτόχρονα και ως χρήστης – παίκτης αλλά και ως τραπέζι όπου θα διατίθεται όλη η πληροφορία σχετικά με το παιχνίδι.

Με τη μορφή που περιγράφηκε στο προηγούμενο κεφάλαιο για την ανταλλαγή των δεδομένων, διατηρούμε ασφαλή την πληροφορία ενός παιχνιδιού και εξασφαλίζεται το αδιάβλητο. Μία συσκευή είναι επιφορτισμένη με τη δημιουργία του παιχνιδιού, σε αυτή την περίπτωση η συσκευή με τον ρόλο του τραπέζιου, και επικοινωνεί με όλες τις υπόλοιπες συσκευές μοιράζοντας τις κάρτες που πρέπει στην αντίστοιχη συσκευή. Προσπάθεια υποκλοπής δεν γίνεται δυνατή με αυτό τον τρόπο.

Σε όλες τις συσκευές υπάρχει το αρχείο `webinosConnector.js`, το οποίο παρέχει όλη την αναγκαία πληροφορία για τη σύνδεση της εφαρμογής με το webinos και την παροχή μεθόδων ικανών να χρησιμοποιήσουν ορθά τις υπηρεσίες του. Τέτοιες μέθοδοι είναι για παράδειγμα η εξεύρεση άλλων συσκευών στο `p2p`, η σύνδεση με το Events Handling API του webinos, οι listeners των γεγονότων όπως περιγράφηκαν ωρίτερα και οι μέθοδοι αποστολής των αντίστοιχων μηνυμάτων.

Ταυτόχρονα έχουμε ένα μικρό αρχείο που δέχεται από το DOM το πλάτος και ύψος της συσκευής ώστε να μπορέσει να χρησιμοποιήσει τα δεδομένα αυτά για την καλύτερη γραφική αναπαράσταση του τραπέζιου και των καρτών, τα οποία όπως αναλύθηκε λόγω των διανυσματικών γραφικών μπορούν να προσαρμοστούν στο ανάλογο μέγεθος που κρίνεται απαραίτητο ανά συσκευή.

Όλα τα γραφικά στοιχεία βρίσκονται σε έναν υποφάκελο, αρχειοθετημένα ανάλογα με την κάρτα που την οποία αντιπροσωπεύουν, σε SVG αρχεία το καθένα. Αυτά τα αρχεία καλούνται μόνο όταν τύχει να χρειαστούμε την αντίστοιχη κάρτα για το παιχνίδι και υπόκεινται στις διάφορες αλλαγές και μετατροπές όπως αυτές ορίζονται στο CSS αρχείο το οποίο περιέχει όλες τις πληροφορίες για αυτή τη λειτουργία αλλά και για την περαιτέρω ωραιοποίηση και απεικόνιση των

αντικειμένων που εμφανίζονται στην εφαρμογή.

Κατά την εκκίνηση της εφαρμογής σε μία συσκευή αναγνωρίζονται οι υπόλοιπες συσκευές της προσωπικής ζώνης και η εφαρμογή μας ανιχνεύει τον χώρο για τραπέζια όπου περιμένουν παίκτες ή διεξάγονται ήδη παιχνίδια. Έτσι εμφανίζεται μια οθόνη με την επιλογή να καθίσουμε ή να παρακολουθήσουμε ένα παιχνίδι ή αλλιώς εναλλακτικά να φιλοξενήσουμε εμείς στη δική μας συσκευή ένα τραπέζι και συνεπώς ένα παιχνίδι.

Αυτό μπορεί να φανεί στο παρακάτω κομμάτι κώδικα:

```
// If there is at least one table created then we ask for it
function com_requestTables() {
    tablesArray = [];
    communication.broadcast('Request Tables from the first host', {});
}

// Only the first host is responsible to send back all tables. (so as to
reduce broadcast signals..)
communication.listen('Request Tables from the first host',
    function (data) {
        if (tablesArray.length > 0)
            if (tableID == tablesArray[0]) {
                communication.broadcast('Give Tables', { allTables:
tablesArray });
            }
    }
);

communication.listen('Give Tables',
    function (data) {
        tablesArray = data.data.allTables;
        gui_refreshTableListGUI();
    }
);
```

Όλα τα γραφικά στοιχεία αλλά και ολόκληρες οι σελίδες δημιουργούνται δυναμικά μέσω JavaScript (και πιο συγκεκριμένα jQuery) σε κάθε συσκευή. Αυτό σημαίνει ότι κώδικας όπως ο παρακάτω:

```

// First Screen
function gui_initialization()
{
    $('body').html('<h1>POKER GAME</h1> <h2>SELECT YOUR TABLE</h2> <div
id="tables"></div> <div id="buttons"> <button onclick="hostANewTable()">Host A
New Table</button> <button onclick="gui_initialization()">Refresh</button>
</div>');

    // Request all Hosted Tables.
    com_requestTables();
}

```

Δημιουργεί κώδικα html μέσω JavaScript με βάση το τι επιστρέφουν οι διάφορες συναρτήσεις. Οπότε αν για παράδειγμα υπάρχουν ήδη δύο τραπέζια αρχικοποιημένα η εφαρμογή χτίζει τον κώδικα ώστε να εμφανίζονται άμεσα αυτά τα δύο τραπέζια στον νεοεισαχθέντα χρήστη.

Μόλις αποφασίσουμε να καθίσουμε σε ένα υπάρχον τραπέζι μας ζητείται να δώσουμε ένα όνομα ως αναγνωριστικό για το παιχνίδι και να πατήσουμε το κουμπί Play. Εδώ δημιουργείται η οντότητα του παίκτη όπου τίθεται και η κατάσταση του ως playing, watching, idle ή offline. Μέσα από μία συγκεκριμένη διαδικασία ο χρήστης ενημερώνεται για τις διάφορες πληροφορίες του τραπέζιού όπως είναι οι υπόλοιποι παίκτες, ποιος παρακολουθεί, πιθανός γύρος παιχνιδιού ο οποίος τρέχει εκείνη τη στιγμή με στοιχεία σαν το ποσό του στοιχήματος ως τώρα, οι 5 κάρτες που υπάρχουν στο τραπέζι και το σημείο στο οποίο βρίσκεται το παιχνίδι, για παράδειγμα ποιος έχει σειρά να παίξει επόμενος.

Όλες αυτές οι διαδικασίες υλοποιούνται όπως έχει ήδη αναφερθεί με τη λογική των μηνυμάτων μεταξύ των συσκευών και με μία σειρά μεθόδων, ιδιωτικών ή δημόσιων, μέσα στον κώδικα που στηρίζεται στην αντικειμενοστρέφεια. Ένα παράδειγμα τέτοιας μεθόδου όταν μπαίνει ένας νέος χρήστης σε ένα τραπέζι είναι και η παρακάτω όπου ελέγχεται σε έναν πίνακα εάν υπάρχει άδεια θέση ώστε να μπορέσει ο χρήστης να παίξει:

```

// If there is a free spot return its place.
function checkIfThereIsFreeSpot() {
    if (emptySeatsArray.length > 0)
        return true;
    return false;
}

```

Τη στιγμή που θα συμπληρωθούν τουλάχιστον 2 θέσεις μπορεί να ξεκινήσει το παιχνίδι. Στην οντότητα του παιχνιδιού όπου και λειτουργεί η κύρια λογική της εφαρμογής ανακατεύεται με απρόβλεπτο τρόπο η τράπουλα N φορές (1 και μόνο είναι αρκετή, αλλά επιλέχθηκε η προσομοίωση τυπικών παιχνιδιών τράπουλας, οπότε γίνονται 4 ανακατέματα), δεσμεύονται 5 κάρτες για το τραπέζι και μοιράζονται από 2 κάρτες σε κάθε παίκτη, δεσμεύονται τα δύο απαιτούμενα πρώτα πονταρίσματα, το μικρό και μεγάλο blind και ενεργοποιούνται τα κουμπιά του επόμενου σε σειρά παίκτη. Τα κουμπιά όλων των υπολοίπων είναι αρχικοποιημένα σε ανενεργή κατάσταση. Ενδεικτικά ο κώδικας που ενεργοποιεί τα κουμπιά του επόμενου παίκτη είναι:

```
function com_enableThisDevicesButtonsGUI(playerNo) {
    communication.broadcast('Enable your buttons please sir', { table:
tableID, device: playerNo, pot: pot });
}

communication.listen('Enable your buttons please sir',
function (data) {
    if (tableID == data.data.table) {
        if (deviceID == data.data.device) {
            if (money == 0)
                check();
            else {
                buttonsEnabled = true;
                gui_refreshTwoCardsGUI();
            }
        }
    }

    for (var i = 0; i < playersArray.length; i++)
        playersNamesArray[playersArray[i]] =
playersNamesArray[playersArray[i]].replace("&rarr;", "");
        playersNamesArray[data.data.device] = '&rarr;';
        playersNamesArray.concat(playersNamesArray[data.data.device]);
        gui_refreshNamesListGUI();
    }
});
```

Οι επιλογές που έχει ο παίκτης σε αυτή τη φάση είναι να:

1. Τα δει (check)
2. Ανεβάσει το στοίχημα (raise)
3. Παίξει με ό,τι έχει (all in)

4. Πάει πάσο (fold)
5. Σηκωθεί (stand up)

Η τελευταία επιλογή ισοδυναμεί με το να πάει πάσο και να φύγει από το τραπέζι την ίδια στιγμή χάνοντας όποιο ποσό είχε στοιχηματίσει μέχρι αυτό το σημείο.

Ενδεικτικά πάλι παρατίθεται ο κώδικας που αντιστοιχεί στην επιλογή check του παίκτη:

```
// Once Check Button is pressed some or one cards are flipped and the game goes on.
this.check = function () {
    buttonsEnabled = false;

    var betBefore = playerHasThrownThatMuchOnTable[deviceID];

    if (deviceID == playersArray[0]) {
        playerHasThrownThatMuchOnTable[deviceID] =
playerHasThrownThatMuchOnTable[playersArray[playersArray.length - 1]];
    }
    else {
        playerHasThrownThatMuchOnTable[deviceID] =
playerHasThrownThatMuchOnTable[playersArray[deviceID - 1]];
    }

    var differenceBetweenBets = playerHasThrownThatMuchOnTable[deviceID] -
betBefore;

    allPlayersMoney[deviceID] -= differenceBetweenBets;

    com_setPlayersBetGUI('check');
}
```

Στο τέλος ενός γύρου παιχνιδιού θα πρέπει να αποτιμηθεί κάπως ποιος είναι νικητής με βάση τα χαρτιά που κρατάει στα χέρια του, έναντι όλων των υπολοίπων που συμμετείχαν στο παιχνίδι μέχρι τέλους. Σημειώνουμε ότι υπάρχει η δυνατότητα να νικήσουν την παρτίδα παραπάνω από ένας παίκτες και να μοιραστούν τα κέρδη. Η αποτίμηση των καρτών που κρατάει ο κάθε παίκτης στα χέρια του γίνεται με συγκεκριμένες μεθόδους που έχουν υλοποιηθεί στο αρχείο poker.js και ακολουθούν τους κανονισμούς του παιχνιδιού όπως αυτοί καθορίστηκαν στο υποκεφάλαιο 3.1 – Εισαγωγή – Αφορμή. Ακολουθούν ένα παράδειγμα όπου η εφαρμογή αποτιμά τις εικόνες της τράπουλας σε νούμερο και ο έλεγχος για το «χέρι» του παίκτη για την περίπτωση που έχει πετύχει 5 κάρτες στη σειρά (straight).


```

this.changeRanksToNumbers = function()
{
    for (var i=0; i<7; i++)
    {
        switch (handRank[i])
        {
            case "J":
                handRank[i] = 11;
                break;
            case "Q":
                handRank[i] = 12;
                break;
            case "K":
                handRank[i] = 13;
                break;
            case "A":
                handRank[i] = 14;
                handRank[-1] = 1;
                handSuit[-1] = handSuit[i];
                break;
            default:
                handRank[i] = parseInt(handRank[i]);
        }
    }
};

```

```

// Player has Straight?
function Straight()
{
    var _handRank = handRank;
    var max = 0;

    Order(_handRank);

    this.isThereStraight = function()
    {
        var straight = 1;
        if (_handRank.length == 8)
            var start = 0, temp = _handRank[-1];
        else
            var start = 1, temp = _handRank[0];
        for (var i=start; i<7; i++)
        {
            if (_handRank[i] == temp + 1)
            {
                straight++;
                temp++;
            }
            else if (straight > 4)
            {
                max = temp;
                return true;
            }
            else
            {
                straight = 1;
            }
        }
    }
};

```

```

        temp = _handRank[i];
    }
}
return false;
};
this.setScore = function() { return SetScore("Straight", 5, max, 0);
};
}

```

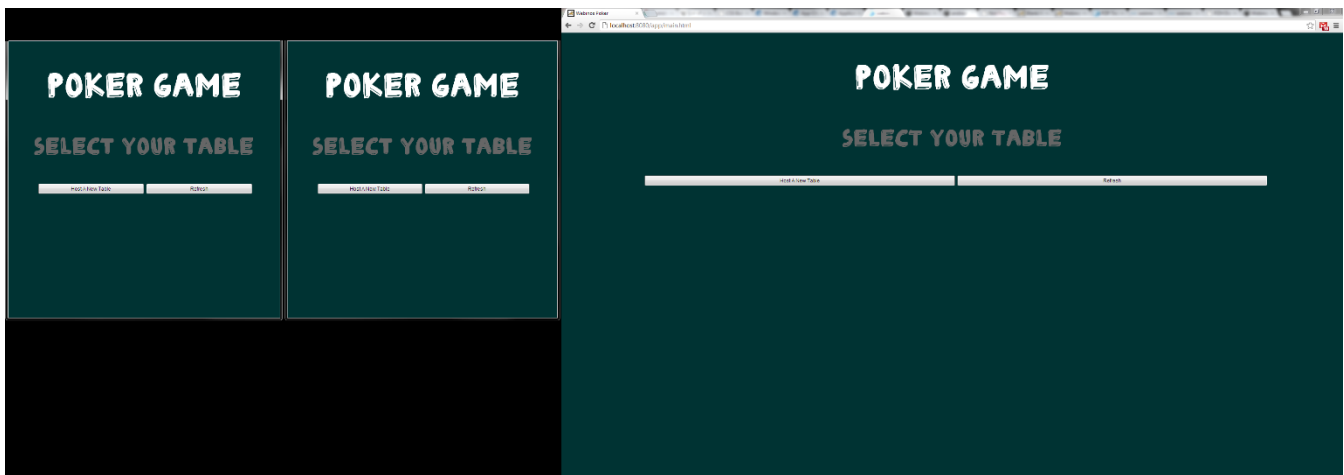
Τέλος, ελέγχονται τα χαρτιά όλων των παικτών, ταξινομούνται και απονέμεται το ποσό του στοιχήματος μαζί με το αντίστοιχο μήνυμα στον ή στους παίκτες που κέρδισαν. Έπειτα ένας νέο γύρος ξεκινάει μεταφέροντας τη σειρά παιχνίματος κατά έναν παίκτη δεξιά. Η ίδια διαδικασία μπαίνει σε εφαρμογή.

Να σημειωθεί ότι με τον τρόπο που έχει υλοποιηθεί η εφαρμογή είναι εφικτό πολύ εύκολα να προσαρμοστεί σε οποιοδήποτε άλλο παιχνίδι τράπουλας κυρίως βάση της αντικειμενοστρέφειας που διακρίνει τον κώδικα. Αυτό εξάγεται από το γεγονός ότι το αντικείμενο που συνιστά τις κάρτες ή το τραπέζι είναι ήδη πλήρως καθορισμένα. Τα μόνα στοιχεία που χρειάζονται αλλαγή είναι οι αρχικές μεταβλητές όπως το πόσοι χρήστες μπορούν να παίξουν ταυτόχρονα στο νέο παιχνίδι που θέλουμε να στήσουμε, πόσες κάρτες μοιράζονται σε αυτούς τους παίκτες και κυρίως οι κανόνες που διέπουν αυτό το νέο παιχνίδι. Σε κάποιες περιπτώσεις όπως για παράδειγμα στην πόκα, οι κανόνες που χρησιμοποιούνται μοιάζουν πολύ με τους δικούς μας, έτσι θα μπορούσαμε να επαναχρησιμοποιήσουμε με μεγάλη ευκολία ένα αξιόλογο σε μέγεθος κομμάτι του κώδικα και να έχουμε άμεσα μία καινούρια εφαρμογή.

3.6 Παράδειγμα Ενδεικτικής Λειτουργίας

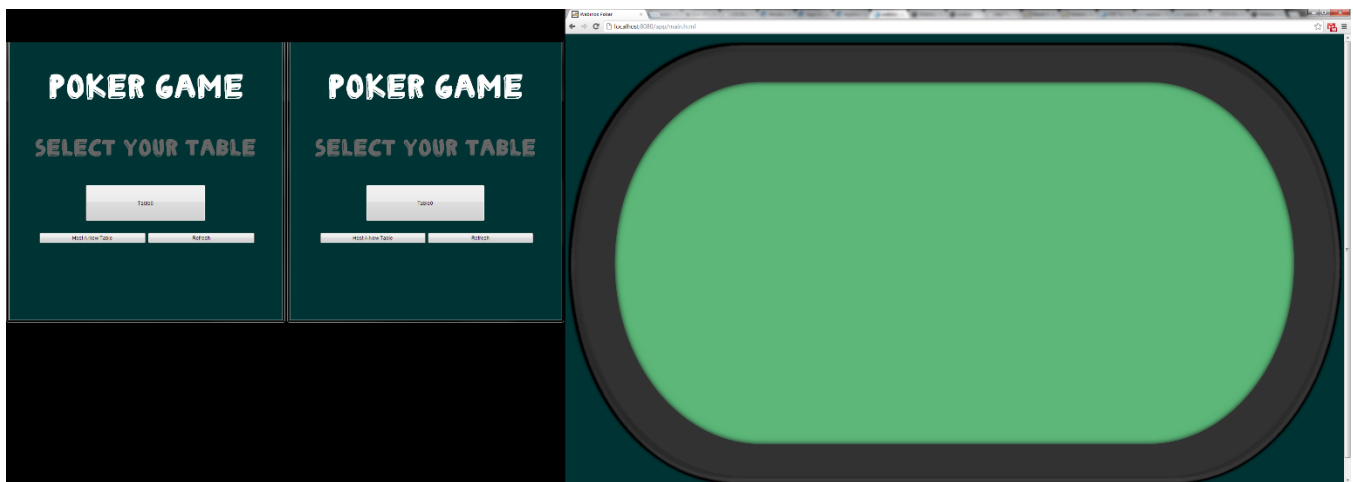
Σε αυτό το υποκεφάλαιο παρατίθεται ένα παράδειγμα ενδεικτικής λειτουργίας της εφαρμογής όπως αυτό καταγράφηκε και αποτυπώθηκε με εικόνες του συστήματος σε όλα τα ενδιάμεσα βήματα.

Διαθέτουμε τρεις συσκευές, μία μεγάλη οθόνη που χρησιμοποιείται ως η κεντρική λογική και απεικόνιση των κοινών καρτών του παιχνιδιού, το τραπέζι και δύο μικρότερες σε έξυπνα κινητά τηλέφωνα με τα οποία συμμετέχουμε στο ίδιο παιχνίδι. Συνεπώς υπάρχουν δύο παίκτες. Σε όλες τις εικόνες που παραθέτουμε εμφανίζονται και οι τρεις οθόνες παράλληλα για να γίνει πιο εύκολα αντιληπτή η αλληλεπίδραση μεταξύ των συσκευών. Πρώτα αρχικοποιούμε την εφαρμογή και στις τρεις συσκευές:



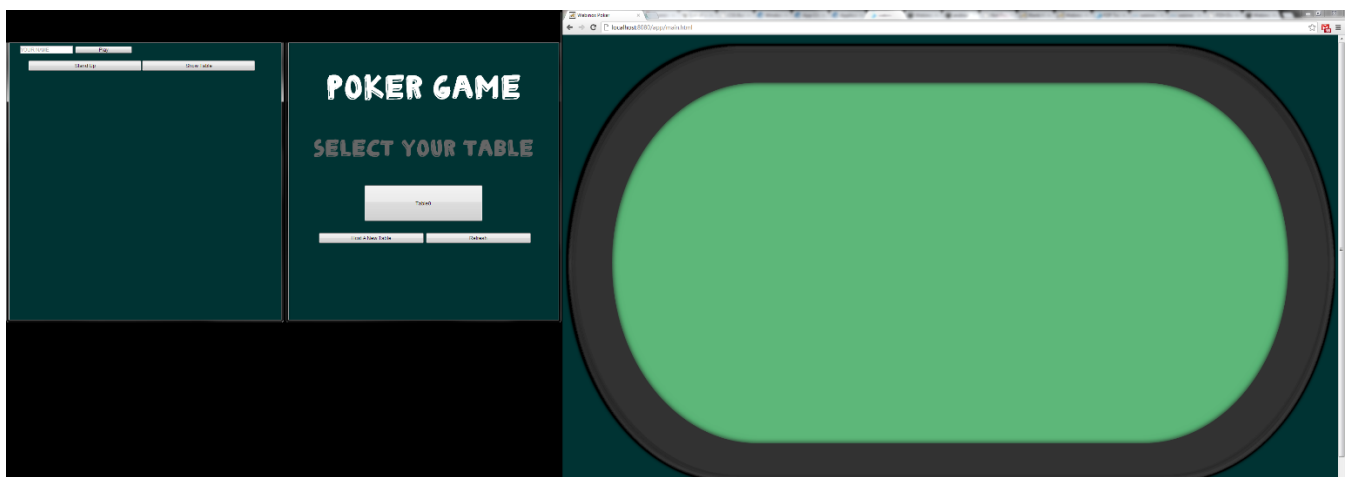
Εικόνα 8 – Αρχικοποίηση Εφαρμογής

Στη συνέχεια δημιουργούμε ένα κοινό τραπέζι στη μεγάλη οθόνη η οποία αμέσως εμφανίζεται ως επιλογή στις μικρότερες συσκευές:

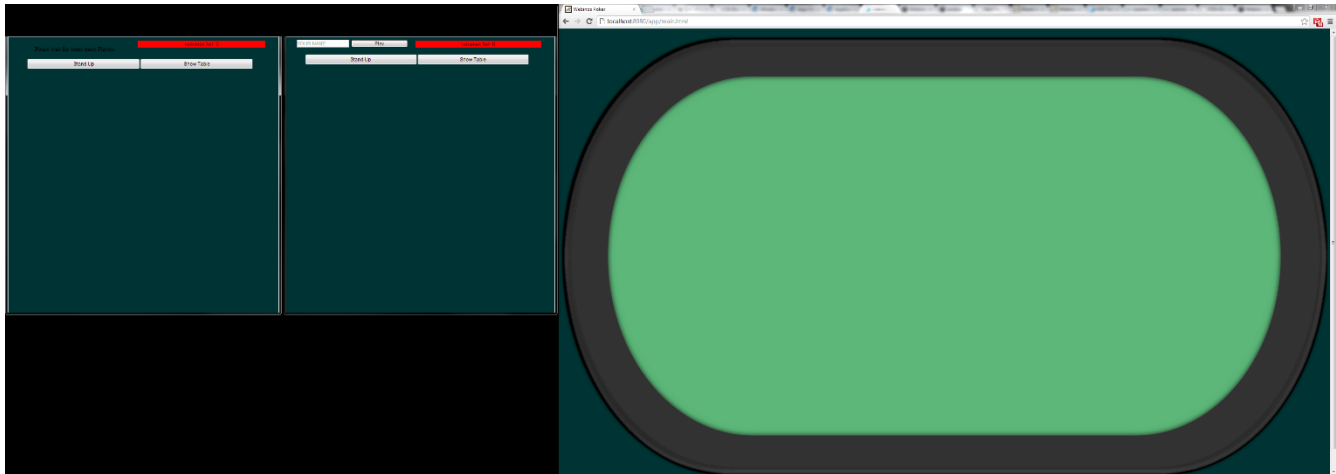


Εικόνα 9 – Δημιουργία Τραπεζιού

Επιλέγουμε το εν λόγω τραπέζι και με τις δύο συσκευές όπου μας ζητείται μια ονομασία – αναγνωριστικό για το παιχνίδι. Η πληροφορία μεταδίδεται άμεσα σε όλους τους συντελεστές.

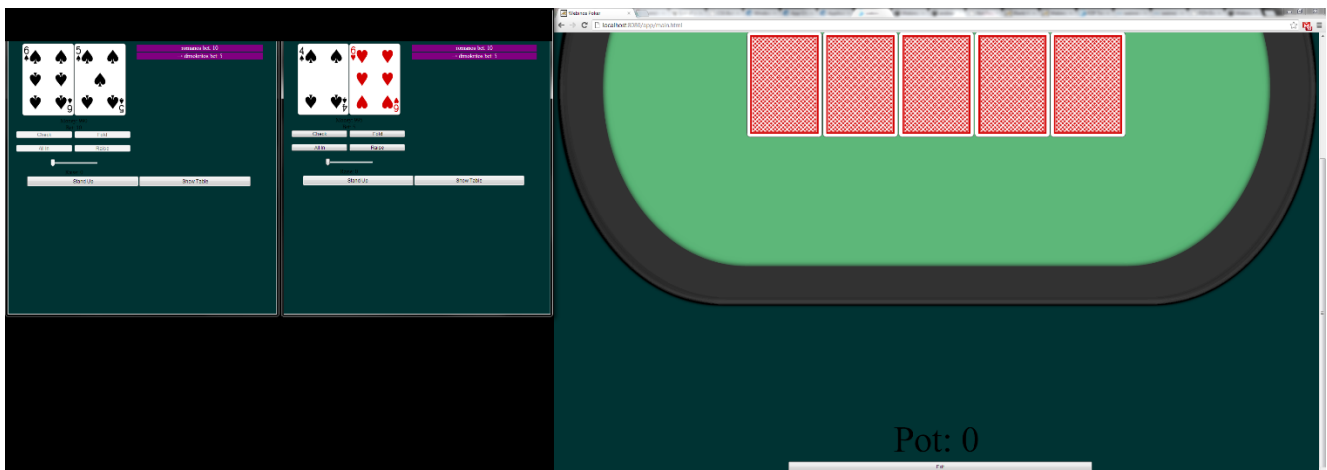


Εικόνα 10 – Εισαγωγή 1^{ου} παίκτη



Εικόνα 11 – Εισαγωγή 2^{ου} παίκτη

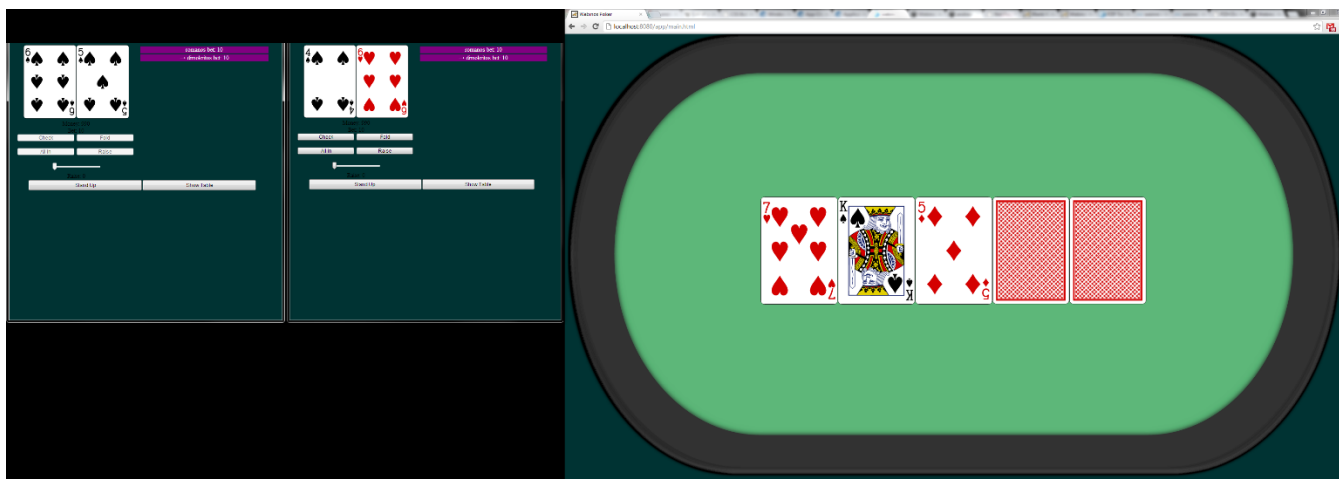
Με την εισαγωγή και του 2^{ου} παίκτη η παρτίδα ξεκινάει. Διαθέτουμε το ποσό των 1000 μονάδων σε κάθε παίκτη. Μοιράζονται 5 φύλλα στο τραπέζι και 2 στον κάθε παίκτη. Ταυτόχρονα εμφανίζονται στις 2 συσκευές τα ονόματα όλων των παικτών μαζί με ένα βελάκι το οποίο συμβολίζει τον επόμενο παίκτη στη σειρά.



Εικόνα 12 – Νέος γύρος παιχνιδιού

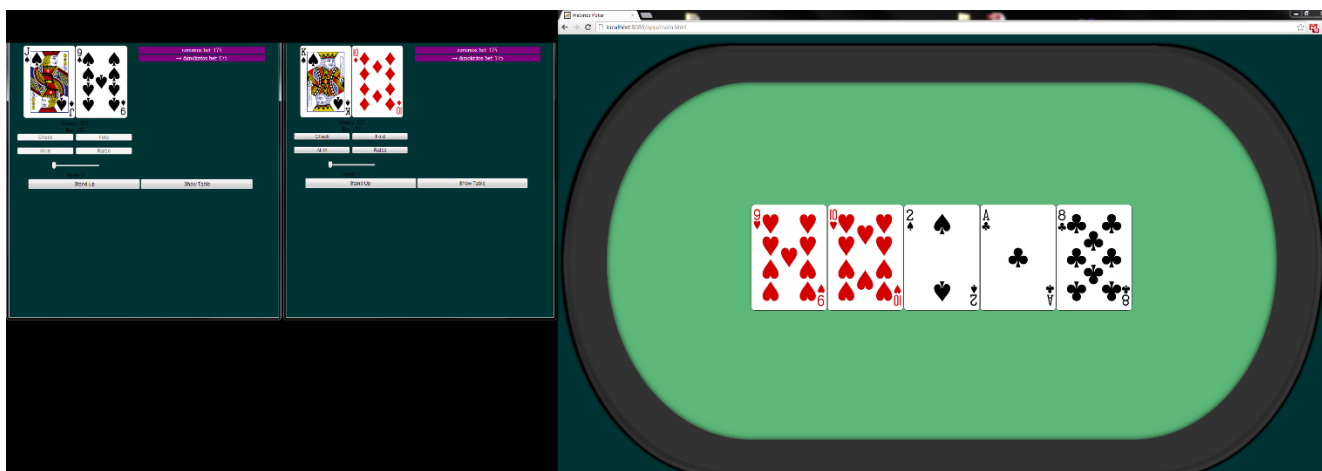
Μετά από τον πρώτο γύρο ίσων στοιχημάτων από όλους τους παίκτες προχωράμε στη φάση όπου ανοίγουν οι τρεις πρώτες κάρτες του τραπέζιού. Τώρα

είναι πιο ξεκάθαρο για τους παίκτες αν υπάρχει κάποιος συνδυασμός που να τους πλησιάζει πιθανοτικά προς τη νίκη.



Εικόνα 13 – Άνοιγμα 3 πρώτων καρτών

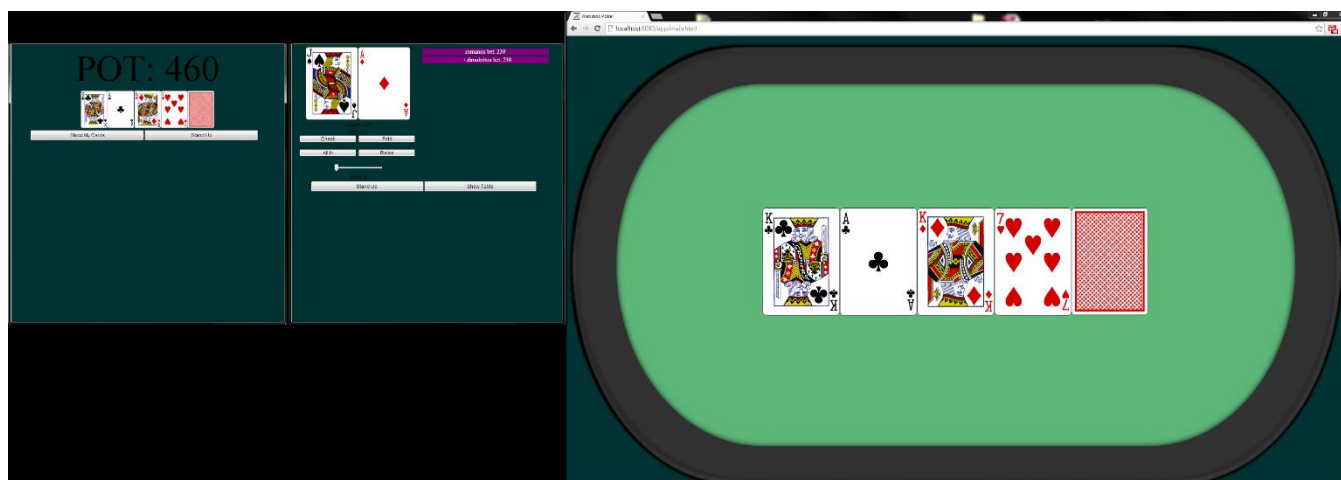
Η παρτίδα συνεχίζεται στον ίδιο ρυθμό με ανέβασμα του στοιχήματος κατά το δοκούν. Στο τέλος βρισκόμαστε με τις 5 κάρτες του τραπέζιού ανοιχτές και συνεπώς γνωρίζουμε όλους τους πιθανούς συνδυασμούς. Εδώ οι παίκτες έχουν το τελευταίο ποντάρισμα πριν αναγκαστούν να ανοίξουν τα χαρτιά τους και να ανακηρυχθεί ο νικητής.



Εικόνα 14 – Ολοκλήρωση παρτίδας

Εδώ βλέπουμε ότι ο ένας παίκτης έχει στα χέρια του 2, 8, δύο 9άρια, 10, βαλέ και άσσο. Με βάση αυτά τα φύλλα κρατάει τα δύο 9άρια, το 10, τον βαλέ και τον άσσο ως το πιο δυνατό χέρι. Ο δεύτερος παίκτης έχει δύο 10άρια αντί για 9άρια και έναν ρήγα αντί για βαλέ. Συνεπώς ο δεύτερος παίκτης είναι και ο νικητής αυτού του γύρου κερδίζοντας 150 μονάδες.

Στην επόμενη οθόνη φαίνεται η επιλογή που έχει η εφαρμογή σε κάθε συσκευή να εμφανίζει την πληροφορία του κεντρικού τραπέζιού για την περίπτωση όπου ένας από τους παίκτες δεν βρίσκεται στον ίδιο χώρο με τους υπόλοιπους και δεν μπορεί να δει συνεπώς την βασική οθόνη.



Εικόνα 15 – Κοινές πληροφορίες τραπέζιού στις συσκευές

4 **Επίλογος**

- 4.1 **Σύνοψη και Συμπεράσματα**
- 4.2 **Μελλοντικές επεκτάσεις**

4.1 Σύνοψη και Συμπεράσματα

Συνοψίζοντας, για την παρούσα διπλωματική εργασία δημιουργήθηκε εφαρμογή με χρήση της ευρωπαϊκής πλατφόρμας ενδιαμέσου λογισμικού webinos. Για την εφαρμογή επιλέχθηκε το ψυχαγωγικό παιχνίδι τράπουλας poker.

Το πρόβλημα συνίσταται από το γεγονός ότι στις μέρες μας υπάρχει δυσκολία εξεύρεσης ενός κοινού πλαισίου αναφοράς για την μετάδοση πληροφορίας, μηνυμάτων και λειτουργικότητας μεταξύ της ευρείας γκάμας των συσκευών που κυκλοφορούν. Το webinos αν και δεν είναι το μόνο που προσπαθεί να λύσει αυτό το πρόβλημα, χρησιμοποιεί τον βέλτιστο τρόπο με τις περισσότερες επιλογές ώστε να φέρει την αλλαγή σε αυτό το θέμα.

Η πλατφόρμα webinos επιτρέπει την χρησιμοποίηση ενός κοινού πλαισίου για τον προγραμματισμό και την παράταξη εφαρμογών σε μια πληθώρα συσκευών με διαφορετικά λειτουργικά συστήματα. Ενδεικτικά αναφέρονται έξυπνα κινητά με λειτουργικό android, μονάδες οικιακής ψυχαγωγίας, προσωπικοί υπολογιστές, αυτοκίνητα με αντίστοιχα συστήματα. Το webinos προσφέρει τη δυνατότητα αξιοποίησης των υπηρεσιών που έχει μία συσκευή μέσα από ένα ενοποιημένο σύστημα API.

Παράλληλα με τα παραπάνω το πλαίσιο webinos επιτρέπει την αποστολή μηνυμάτων και πληροφορίας μεταξύ των εφαρμογών διαφορετικών συσκευών μέσα σε ένα κοινό χώρο τον οποίο ονομάζει προσωπική ζώνη. Μέσα σε αυτή τη ζώνη υπάρχουν συσκευές έμπιστες για τον χρήστη οι οποίες μπορούν να αξιοποιηθούν πλήρως από τον χρήστη. Η μετάδοση της πληροφορίας γίνεται με ασφαλή τρόπο.

Για την εφαρμογή webinos poker χρησιμοποιήθηκαν μία σειρά από APIs του webinos, τα οποία παρέχουν τις απαραίτητες λειτουργικότητες των συσκευών στην εφαρμογή ώστε να γίνει ένα παιχνίδι poker εφικτό.

Το παιχνίδι στηρίζεται στους κανονισμούς όπως ορίζονται για το poker texas hold'em. Με βάση αυτούς τους κανονισμούς χτίστηκε με αντικειμενοστραφή τρόπο ο κώδικας, χρησιμοποιώντας τις γλώσσες HTML, CSS, JavaScript και την τεχνολογία SVG, αρχεία βασισμένα στην γλώσσα XML με τα οποία μπορούμε να σχεδιάσουμε

εύκολα και απλά εικόνες με χρήση διανυσμάτων, κάτι που ευνοεί την μεγέθυνση ή σμίκρυνση της εικόνας όσο θέλουμε χωρίς καμμία απολύτως παραμόρφωση, κοινό πρόβλημα για όλες τις άλλες μορφές απεικονίσεων. Ιδανική λύση για εφαρμογή η οποία τρέχει σε αρκετές συσκευές διαφορετικών μεγεθών και αναλύσεων η καθεμία.

Κατά τη διάρκεια του παιχνιδιού ανταλλάσσονται ασφαλή μηνύματα μεταξύ των συσκευών – παικτών και μιας κεντρικής οθόνης όπου παρουσιάζονται οι κοινές κάρτες – πληροφορίες. Οι παίκτες μπορούν να διασκεδάσουν συμμετέχοντας σε ένα παιχνίδι χρησιμοποιώντας το κινητό, τηλεόραση, υπολογιστή ή όποια άλλη υποστηριζόμενη συσκευή επιθυμούν.

Η δημιουργία εφαρμογής με τη χρήση της πλατφόρμας webinos ταυτόχρονα με την ίδια την ανάπτυξη της πλατφόρμας από τις συνεργαζόμενες εταιρείες και πανεπιστήμια δεν έκανε καθόλου εύκολη την ενσωμάτωση του κώδικα του πλαισίου στον κώδικα μας με συχνά σφάλματα. Έχοντας διαρκή συνεννόηση με την ομάδα ανάπτυξης, τα προβλήματα που παρουσιάστηκαν λύθηκαν. Η πλατφόρμα είναι πλέον στο τελικό στάδιο ανάπτυξης της και συνεπώς σε πιο σταθερή μορφή.

4.2 Μελλοντικές επεκτάσεις

Η εφαρμογή webinos roker, παρόλο που βρίσκεται σε λειτουργία και μπορεί να χρησιμοποιηθεί χωρίς κανένα ιδιαίτερο πρόβλημα, θα δεχόταν σίγουρα πολλές και αξιόλογες διορθώσεις αλλά και επεκτάσεις.

Σκόπιμο κρίνεται να γίνει ο κώδικας ακόμη πιο γενικός και αντικειμενοστραφής για την περαιτέρω απλούστευση των διαδικασιών και την εύκολη επαναχρησιμοποίησή του όπου χρειαστεί. Ένα χρόνο μετά την αρχική έκδοση της εφαρμογής εμφανίζονται αρκετές παρατυπίες στον κώδικα οι οποίες με τη χρησιμοποίηση της επιπλέον κερδισμένης εμπειρίας, η εφαρμογή μπορεί να γίνεται ακόμη καλύτερη.

Βασικό στοιχείο επέκτασης θα ήταν να χρησιμοποιηθούν περισσότερα APIs του webinos και ως εκ τούτου μεγαλύτερη λειτουργικότητα από τις συσκευές. Ένα παράδειγμα θα ήταν η χρησιμοποίηση του Device Status API το οποίο παρέχει πληροφορία όπως είναι η κατάσταση της μπαταρίας ή το μέγεθος της οθόνης που θα μπορούσε να αξιοποιηθεί παραδείγματος χάριν για την εμφάνιση αντίστοιχων μηνυμάτων προς τον χρήστη. Το MediaPlayer API θα μπορούσε να χρησιμοποιηθεί για την οπτικοακουστική επιβράβευση του νικητή ενός γύρου παιχνιδιού. Αντίστοιχα, θα μπορούσαν να εξερευνηθούν μία σειρά από άλλες αντίστοιχες αλλαγές με βάση τις προσφερόμενες υπηρεσίες του webinos.

Το ίδιο το παιχνίδι θα μπορούσε να αλλάξει σε πολλούς τομείς ώστε να γίνει πιο αρεστό. Αισθητικά, με καλύτερο υλικό και μεταβάσεις μεταξύ των βημάτων. Πιο επεξηγηματικά στάδια και υλοποίηση βοήθειας για την εφαρμογή. Χρηστικά, καλό θα ήταν να υπάρχει δυνατότητα αποθήκευσης της κατάστασης ενός παιχνιδιού και επαναφορά του μια άλλη στιγμή. Πολύ σημαντική αλλαγή θα ήταν η χρησιμοποίηση επιπρόσθετων αμοιβών σε στάδια – ορόσημα για κάποιον παίκτη, το λεγόμενο gamification μιας εφαρμογής.

Μια επέκταση η οποία είναι σχετικά εύκολη στην υλοποίηση και θα προσέθετε μεγάλη αξία στην εφαρμογή, είναι η υλοποίηση περισσότερων παιχνιδιών τράπουλας με βάση και πάλι την πλατφόρμα webinos. Αυτό θα μπορούσε να γίνει χρησιμοποιώντας την ήδη υλοποιημένη εφαρμογή, είτε ως προσθήκη στην

εφαρμογή, είτε ως τελείως διαφορετικό κομμάτι στην πλατφόρμα. Για παράδειγμα, θα μπορούσαμε να υλοποιήσουμε παιχνίδια [21](#) όπως τα: 21, πόκα, αγωνία, πρέφα και άλλα πολλά, σε ένα σετ διασκέδασης.

Μία ακόμη επιπλέον επέκταση που θα ήταν ενδιαφέρουσα και από αλγοριθμική άποψη, είναι η αξιολόγηση των διαφορετικών χειρών ενός χρήστη και η εξαγωγή πιθανοτικών δεδομένων με βάση τα οποία θα μπορούσαμε να παρέχουμε βοήθεια για την απόφαση που μπορεί να πάρει ο παίκτης όσον αφορά τα πονταρίσματα.

Τέλος, η εισαγωγή υπολογιστικής ευφυΐας (Artificial Intelligence) στο παιχνίδι για αύξηση ίσως του ενδιαφέροντος ή για την περίπτωση ενός χρήστη – παίκτη θα ήταν επίσης πολύ καλή.

5 Αναφορές - Βιβλιογραφία

5.1 Αναφορές

5.2 Βιβλιογραφία

5.1 Αναφορές

- [1] [Σε λιγότερο από τέσσερα χρόνια η παγκόσμια κίνηση δεδομένων μέσω IP θα έχει τριπλασιαστεί](#), Δημοσίευση στο Πρώτο Θέμα στις 3 Ιουνίου 2013.
- [2] [Η αύξηση της πρόσβασης στο internet μέσω κινητών συσκευών επιταχύνεται σε όλο τον κόσμο!](#) Wedia Digital Agency
- [3] <http://www.w3.org/html/> & <http://en.wikipedia.org/wiki/HTML>
- [4] <http://www.w3.org/Style/CSS/> & http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [5] <http://en.wikipedia.org/wiki/JavaScript>
- [6] <http://www.w3.org/Graphics/SVG/> & http://en.wikipedia.org/wiki/Scalable_Vector_Graphics
- [7] <https://github.com/>
- [8] <http://git-scm.com/> & [http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software))
- [9] <http://json-rpc.org/>
- [10] http://en.wikipedia.org/wiki/Transport_Layer_Security
- [11] http://en.wikipedia.org/wiki/Transport_Layer_Security
- [12] http://en.wikipedia.org/wiki/Document_Object_Model
- [13] <http://www.w3.org/TR/widgets-apis/>
- [14] 1^{ος} Διαγωνισμός Εφαρμογών από το webinos.
<http://www.webinos.org/1stappchallenge/>
- [15] Νικητές 1^{ου} Διαγωνισμού Εφαρμογών webinos cross-screen challenge και επιπλέον υλικό (2012, Νοέμβριος). <http://www.webinos.org/blog/2012/11/14/app-challenge-finalist-apps-demo-videos/>

- [16] <http://el.wikipedia.org/wiki/Πόκερ>
- [17] http://el.wikipedia.org/wiki/Τέξας_χόλντεμ
- [18] http://en.wikipedia.org/wiki/Unified_Modeling_Language
- [19] http://en.wikipedia.org/wiki/Object-oriented_programming
- [20] http://en.wikipedia.org/wiki/Open_source
- [21] http://en.wikipedia.org/wiki/Playing_card

5.2 Βιβλιογραφία

- [1] Vergori, P., Ntanos, C., Gavelli, M., and Askounis, D. 2013. The webinos Architecture: A Developer's Point of View. In *Mobile Computing, Applications, and Services. Fourth International Conference, Mobicase 2012, Seattle, Wa, USA, October 2012. Revised Selected Papers*, D. Uhler, K. Mehta and J. L. Wong, Eds. Springer-Verlag New York Inc
- [2] Ntanos, C., Botsikas, C., Rovis, G., Kakavas, P., & Askounis, D. (2013). A context awareness framework for cross-platform distributed applications. *Journal of Systems and Software*.
- [3] Fuhrhop, C., Lyle, J., & Faily, S. (2012, April). The webinos project. In *Proceedings of the 21st international conference companion on World Wide Web* (pp. 259-262). ACM.
- [4] Lyle, J., Faily, S., Fléchais, I., Paul, A., Göker, A., Myrhaug, H., ... & Martin, A. (2012, January). On the design and development of webinos: a distributed mobile application middleware. In *Distributed Applications and Interoperable Systems* (pp. 140-147). Springer Berlin Heidelberg.
- [5] Desruelle, H., Lyle, J., Isenberg, S., & Gielen, F. (2012). On the challenges of building a web-based ubiquitous application platform.
- [6] webinos, 2013. webinos Device apis - Working Repository. <http://dev.webinos.org/specifications/api/>
- [7] webinos system specifications. (2013). <http://www.webinos.org/content/html/D033/index.html>
- [8] Su, T., Lyle, J., Atzeni, A., Faily, S., Virji, H., Ntanos, C., & Botsikas, C. (2013). Continuous Integration for Web-Based Software Infrastructures: Lessons Learned on the webinos Project. In *Hardware and Software: Verification and Testing* (pp. 145-150). Springer International Publishing.
- [9] Ροβής, Γ. (2013). Μοντελοποίηση Δεδομένων Περιβάλλοντος Προσωπικών Πληροφοριακών Συστημάτων. Διπλωματική Εργασία. <http://artemis->

new.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/6577/1/DT2013-0023.pdf

[10] Μπόμπα, Χ., Ντάνος, Χ. (2013). Ένα περιβάλλον επίγνωσης πλαισίου για κατανομημένες εφαρμογές πολλαπλών πλατφορμών.

[11] Faily, S., Flechais, I., Lyle, J., & Martin, A. (2012). On the design and development of webinos: a distributed mobile application middleware.

[12] webinos, 2013. Webinos Home Page. <http://www.webinos.org/>

[13] Sklansky, D. B., & Jones, D. A. (1995). U.S. Patent No. 5,382,025. Washington, DC: U.S. Patent and Trademark Office. (Method for playing a poker game)

[14] Gilpin, A., & Sandholm, T. (2006, July). A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. In Proceedings of the National Conference on Artificial Intelligence (Vol. 21, No. 2, p. 1007). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[15] Οι κανόνες και η ιστορία του παιχνιδιού Poker στη wikipedia. (2013, Δεκέμβριος). <http://en.wikipedia.org/wiki/Poker>

[16] Billings, D., Burch, N., Davidson, A., Holte, R., Schaeffer, J., Schauenberg, T., & Szafron, D. (2003, August). Approximating game-theoretic optimal strategies for full-scale poker. In IJCAI (pp. 661-668).

[17] Εποχή της πληροφορίας στη Wikipedia. (2013, Μάιος). http://el.wikipedia.org/wiki/%CE%95%CF%80%CE%BF%CF%87%CE%AE_%CF%84%CE%B7%CF%82_%CF%80%CE%BB%CE%B7%CF%81%CE%BF%CF%86%CE%BF%CF%81%CE%AF%CE%B1%CF%82

[18] Pfleeger, S., (2003). Τεχνολογία Λογισμικού: Θεωρία και Πράξη, Α Τόμος. Εκδ. Κλειδάριθμος, ISBN 960-209-620-9.

[19] Sommerville, I., (2008, Μάιος). Βασικές Αρχές Τεχνολογίας Λογισμικού. Εκδ. Κλειδάριθμος, 8η έκδοση.

[20] UML Class Diagram in Wikipedia. (2013, December). http://en.wikipedia.org/wiki/Class_diagram

[21] UML Use Case Diagram in Wikipedia. (2013, November). http://en.wikipedia.org/wiki/Use_Case_Diagram