



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΛΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σύγχρονες τάσεις στην Ανάπτυξη Εφαρμογών για Κινητές
Συσκευές: Μια κριτική προσέγγιση»

ΚΩΝΣΤΑΝΤΙΝΟΣ-ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΦΙΛΟΣ-ΤΣΙΤΣΩΝΗΣ

Επιβλέπων: Ιάκωβος Σ. Βενιέρης

Καθηγητής Ε.Μ.Π

Συνεπίβλεψη: Δρ. Σ. Καπελλάκη, ΕΔΙΠ Ε.Μ.Π.

Αθήνα, Μάρτιος 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΛΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σύγχρονες τάσεις στην Ανάπτυξη Εφαρμογών για Κινητές

Συσκευές: Μια κριτική προσέγγιση»

ΚΩΝΣΤΑΝΤΙΝΟΣ-ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΦΙΛΟΣ-ΤΣΙΤΣΩΝΗΣ

Επιβλέπων: Ιάκωβος Σ. Βενιέρης

Καθηγητής Ε.Μ.Π

Συνεπίβλεψη: Δρ. Σ. Καπελλάκη, ΕΔΙΠ Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10^η
Μαρτίου 2020.

.....

.....

.....

Ι. Βενιέρης

Θ. Κακλαμάνη

Γ. Ματσόπουλος

Καθηγητής Ε.Μ.Π.

Καθηγήτρια Ε.Μ.Π.

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020

**Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10^η
Μαρτίου 2020.**

.....

.....

.....

Ι. Βενιέρης

Θ. Κακλαμάνη

Γ. Ματσόπουλος

Καθηγητής Ε.Μ.Π.

Καθηγήτρια Ε.Μ.Π.

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020

..... Κωνσταντίνος-Γεώργιος Χριστόφιλος-Τσιτσώνης Διπλωματούχος Ηλεκτρολόγος
Μηχανικός & Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © Κωνσταντίνος-Γεώργιος Χριστόφιλος-Τσιτσώνης, 2020. Με επιφύλαξη παντός δικαιώματος. All rights reserved. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η συγκεκριμένη διπλωματική εργασία έχει ως κύριο αντικείμενο την επισκόπηση των πλατφόρμων ανάπτυξης εφαρμογών για κινητά τηλέφωνα. Στο πρώτο κεφάλαιο παρουσιάζουμε μία ανασκόπηση των εφαρμογών κινητών τερματικών και τις επιμέρους κατηγορίες τους. Ταυτόχρονα, δίνουμε έμφαση στα πλεονεκτήματα και στα μειονεκτήματα των native εφαρμογών και γίνεται μία σύγκριση μεταξύ των υβριδικών και των native εφαρμογών.

Στο δεύτερο κεφάλαιο περνάμε σε μία συγκριτική παρουσίαση των δύο λειτουργικών-κολοσσών της σύγχρονης εποχής-του Android και του Iphone Operating System. Στο συγκεκριμένο κεφάλαιο δίνουμε περισσότερη βαρύτητα στις native mobile apps.

Το τρίτο κεφάλαιο έχει ως βασικό του θέμα της διαδικτυακές εφαρμογές καθώς και τα κυριότερα εργαλεία ανάπτυξής τους. Εξάλλου, οι web εφαρμογές μονοπώλουν σήμερα την αγορά των mobile apps.

Το τέταρτο κεφάλαιο αποτελεί μία αναλυτική έκθεση των υβριδικών εφαρμογών για κινητές συσκευές και η εργασία μας ολοκληρώνεται με μία σειρά από συμπεράσματα και μελλοντικές επεκτάσεις.

Λέξεις Κλειδιά

Κινητά τηλέφωνα, Android, IOS, υβριδικές, native, διαδικτυακές εφαρμογές

Abstract

This thesis focuses on the overview of mobile application development platforms. In the first chapter we present an overview of mobile terminal applications and their individual categories. At the same time, we emphasize the advantages and disadvantages of native applications and a comparison is made between hybrid and native applications.

In the second chapter we go through a comparative presentation of the two modern-day operating giants - the Android and the Iphone Operating System. In this chapter we give more weight to native mobile apps.

The third chapter focuses on web applications as well as their main development tools. In addition, web applications are currently monopolizing the market for mobile apps. Chapter 4 is an in-depth exposition of hybrid mobile applications and our work concludes with a series of conclusions and future extensions.

Λέξεις Κλειδιά

Mobile phones, Android, IOS, hybrid, native, web applications

Ευχαριστίες

Σε αυτό το σημείο που ολοκληρώνονται οι σπουδές μου θα ήθελα αρχικά να ευχαριστήσω τους γονείς μου που με έχουν στηρίξει σε κάθε σημείο της ζωής μου με κάθε τρόπο. Επιπλέον, ευχαριστώ τον καθηγητή μου από τη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου κ. Ιάκωβο Βενιέρη γιατί στάθηκε δίπλα μου και στα μαθήματα που με εξέτασε καθώς και στην εκπόνηση της διπλωματικής μου εργασίας με περίσσεια ευαισθησία και κατανόηση. Η συγκεκριμένη εργασία δεν θα ολοκληρωνόταν χωρίς τη βοήθεια της Δρ. κας Σοφίας Καπελλάκη. Τόσο η καθοδήγησή που μου παρείχε όσο και οι τελικές διορθώσεις του κειμένου υπήρξαν καταλυτικής σημασίας για την ολοκλήρωση της παρούσας.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή - Ανασκόπηση Εφαρμογών Κινητών Τερματικών και Κατηγορίες (Native/Hybrid/Web).....	σελ.10
1.1 Γενικά.....	σελ.10
1.2 Πλεονεκτήματα-Μειονεκτήματα Native εφαρμογών (Native applications)...	σελ.11
1.3 Σύγκριση μεταξύ Εγγενών και Υβριδικών Εφαρμογών.....	σελ.16
ΚΕΦΑΛΑΙΟ 2: Native Mobile Apps (Android, iOS).....	σελ.18
2.1 Ανάπτυξη Android εφαρμογών με Java (Android Studio).....	σελ.20
2.2 Ανάπτυξη iOS εφαρμογών με Swift και Objective C.....	σελ.29
2.3 Δομικά Στοιχεία μιας εφαρμογής.....	σελ.32
ΚΕΦΑΛΑΙΟ 3: Web Mobile Apps.....	σελ.34
3.1 HTML5.....	σελ.34
3.2 CSS3.....	σελ.36
3.3 JavaScript.....	σελ.37
3.4 Responsive Web Design Techniques.....	σελ.39
3.5 BootStrap.....	σελ.41
ΚΕΦΑΛΑΙΟ 4: Hybrid Mobile Apps.....	σελ.46
4.1 Phone Gap.....	σελ.46
4.2 Apache Cordova.....	σελ.58
4.3 Ionic.....	σελ.50
4.4 Πολυεπίπεδες πλατφόρμες Ανάπτυξης (Cross-Platforms development).....	σελ.52
4.5 Υποστήριξη εφαρμογών κινητού υπολογισμού από τεχνολογίες νέφους.....	σελ.58
4.6 Ενημέρωση Εφαρμογών.....	σελ.61
ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα-Επεκτάσεις.....	σελ.68
Βιβλιογραφία-Αναφορές.....	σελ.75

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή - Ανασκόπηση Εφαρμογών Κινητών Τερματικών και Κατηγορίες (Native/Hybrid/Web)

1.1 Γενικά

Τα smartphones έχουν εισέλθει δυναμικά στην αγορά παρουσιάζοντας μία σταθερά ανοδική πορεία πωλήσεων με εξαιρετικές προοπτικές περαιτέρω ανάπτυξης. Αναπόσπαστο κομμάτι των smartphones αποτελούν οι εφαρμογές κινητών (mobile applications) οι οποίες μπορούν να καλύψουν μία πληθώρα αναγκών. Πολλές εταιρίες και πολλοί προγραμματιστές, έχουν ήδη αναπτύξει τις δικές τους εφαρμογές για κινητές συσκευές, προσφέροντας υπηρεσίες προστιθέμενης αξίας στους καταναλωτές, καθώς μία mobile εφαρμογή μπορεί να εξυπηρετήσει προσωπικές, επαγγελματικές, ενδοεπιχειρηματικές ανάγκες.

Υπάρχουν τρία είδη εφαρμογών για κινητές συσκευές, οι native εφαρμογές, οι διαδικτυακές εφαρμογές (Web applications) και οι Υβριδικές εφαρμογές (Hybrid applications). Κάθε μια έχει τα δικά της μειονεκτήματα και πλεονεκτήματα.

Οι native εφαρμογές για smartphones τρέχουν αποκλειστικά στη συσκευή. Κάθε native εφαρμογή προορίζεται και είναι σχεδιασμένη να τρέχει σε ένα συγκεκριμένο λειτουργικό σύστημα κινητών όπως το Android και το iOS. Αποτελείται από ένα εκτελέσιμο αρχείο το οποίο αποθηκεύεται στη συσκευή (.apk για Android, .app για Iphone). Η εφαρμογή αυτή μπορεί να υπάρχει προεγκαταστημένη ως μέρος του λειτουργικού συστήματος αλλά και να γίνει διαθέσιμη με την εγκατάσταση της από τον χρήστη μέσω του ηλεκτρονικού καταστήματος της εταιρείας, της οποίας ανήκει το λειτουργικό σύστημα που χρησιμοποιεί η εκάστοτε συσκευή (πχ το Google Play του Android, App Store της Apple). Μετά την εγκατάσταση συνήθως δημιουργείται ένα εικονίδιο στη επιφάνεια εργασίας με το οποίο ο χρήστης μπορεί να εκκινήσει την εφαρμογή κάθε φορά που θέλει να την χρησιμοποιήσει. Κατά την πρώτη εκτέλεση η εφαρμογή θα συνδεθεί άμεσα με το λειτουργικό σύστημα, χωρίς κανένα ενδιάμεσο στρώμα λογισμικού. Με αυτόν τον τρόπο μπορεί να έχει πρόσβαση στις λειτουργίες των εφαρμογών που συνοδεύουν το λειτουργικό σύστημα, αποκτώντας πλήρη έλεγχο της συσκευής. Οι Native εφαρμογές αναπτύσσονται συνήθως χρησιμοποιώντας ένα IDE. Τα IDE's όπως αναφέρθηκε παραπάνω, παρέχουν εργαλεία για τη δημιουργία αποσφαλμάτωσης, τη διαχείριση του σχεδίου, τον έλεγχο της έκδοσης, και άλλα

εργαλεία που χρειάζονται οι επαγγελματίες προγραμματιστές. Οι Web εφαρμογές για smartphones είναι διαδικτυακές εφαρμογές ειδικά σχεδιασμένες για χρήση σε smartphone ή tablet και ανοίγουν μέσω του web browser της συσκευής, με την επίσκεψή στην αντίστοιχη ηλεκτρονική διεύθυνση (URL) της εφαρμογής. Γενικά η πρόσβαση στα web applications για smartphones γίνεται μέσω του browser. Μπορούν να τρέξουν σε οποιαδήποτε πλατφόρμα που υποστηρίζει browser όπως android, iOS, Symbian ή BlackBerry. Ονομάζονται εφαρμογές ανεξάρτητης πλατφόρμας καθώς εκτελούνται στον browser της συσκευής, χωρίς να απαιτείται εγκατάσταση σε κάθε συσκευή ξεχωριστά, αποφεύγοντας με αυτόν τον τρόπο τα διάφορα ηλεκτρονικά καταστήματα (marketplaces) των παρόχων των λειτουργικών συστημάτων και τις διαφορετικές απαιτήσεις που θα πρέπει να πληρεί μια εφαρμογή για να γίνει διαθέσιμη μέσω αυτών [1].

1.2 Πλεονεκτήματα-Μειονεκτήματα Native εφαρμογών (Native applications)

Για τη δημιουργία μιας native εφαρμογής ο προγραμματιστής θα πρέπει να γράψει τον πηγαίο κώδικα, στην γλώσσα προγραμματισμού που υποστηρίζει το λειτουργικό σύστημα της συσκευής, και να συμπεριλάβει οτιδήποτε άλλο είναι απαραίτητο για την λειτουργία της εφαρμογής, όπως για παράδειγμα εικόνες, ηχητικά αρχεία κλπ.

Χρησιμοποιώντας εργαλεία που παρέχονται από την εταιρεία που δημιούργησε το λειτουργικό σύστημα, τα παραπάνω αρχεία μεταγλωττίζονται και παράγεται ένα αρχείο το οποίο αποτελεί την εφαρμογή που θα αποθηκευτεί στη συσκευή. Τα εργαλεία αυτά, σε συνδυασμό με επιπρόσθετες ευκολίες αποτελούν το περιβάλλον ανάπτυξης εφαρμογών (SDK) που προορίζονται για το εκάστοτε λειτουργικό σύστημα. Η διαδικασία ανάπτυξης μιας native εφαρμογής παρουσιάζει διαφορές μεταξύ των διαφορετικών λειτουργικών συστημάτων, όμως τα περιβάλλοντα ανάπτυξης (SDK's) διαφέρουν μεταξύ τους. Έχουν δημιουργηθεί για να εξυπηρετούν τις ανάγκες ενός συγκεκριμένου λειτουργικού συστήματος και παρέχουν διαφορετικά εργαλεία. Έτσι προκύπτει ότι βασικότερο μειονέκτημα των native εφαρμογών είναι ότι ο κώδικας που γράφτηκε για ένα συγκεκριμένο λειτουργικό σύστημα δεν μπορεί να χρησιμοποιηθεί σε κάποιο άλλο, που σημαίνει πως πρέπει να γραφτεί διαφορετικός κώδικας για την ίδια εφαρμογή ανάλογα με το λειτουργικό σύστημα. Επίσης πρέπει να ληφθεί υπόψη και οι αλλαγές που μπορεί να

υπάρχουν σε hardware χαρακτηριστικά πχ. retina display, ή εντελώς διαφορετικά hardware setup σε android συσκευές. Αν μια εφαρμογή στοχεύει διάφορα λειτουργικά συστήματα, αυξάνει την πολυπλοκότητα ανάπτυξης και συντήρησής της καθώς πρέπει να ξαναγραφτεί ο κώδικας για το εκάστοτε λειτουργικό σύστημα. Οι προγραμματιστές μπορούν να επιλέξουν να εξυπηρετήσουν μόνο ένα τμήμα της αγοράς (πλατφόρμα) με την ανάπτυξη εφαρμογών για μόνο ένα υποσύνολο των λειτουργικών συστημάτων, και κατά συνέπεια ένα υποσύνολο των μοντέλων των συσκευών. Ακόμη και αν έχουν κάνει την επιλογή τους όμως, μπορεί να υποχρεωθούν να φιλοξενήσουν διαφορετικές εκδοχές για την ίδια πλατφόρμα (π.χ. iOS 6.x, iOS 5.x, και τις διάφορες εκδόσεις του Android), αλλά και μια μεγάλη ποικιλία αναλύσεων οθόνης και αναλογίας εικόνας. Η ανάπτυξη εφαρμογών για τα μοντέλα συσκευών που αντιπροσωπεύουν την πλειοψηφία των συσκευών έχει γίνει μια δαπανηρή και χρονοβόρα διαδικασία. Αυτός ο κατακερματισμός, επηρεάζει σοβαρά κάθε οικοσύστημα, καθιστώντας πιο δύσκολο για τους μικρούς προγραμματιστές να ανταγωνιστούν, δεδομένου ότι είναι απίθανο να έχουν τους πόρους για να υποστηρίξουν την αυξανόμενη λίστα των μοντέλων συσκευών που χρησιμοποιούνται σήμερα. Μπορούν επίσης να είναι σε μειονεκτική θέση σε οικονομίες κλίμακας για την προώθηση των εφαρμογών τους, εάν δεν είναι διαθέσιμες ή δεν λειτουργούν σωστά στα περισσότερα μοντέλα συσκευών.

Τέλος ένα μικρό μειονέκτημα μπορεί να θεωρηθεί και το γεγονός πως για να γίνει update η εφαρμογή, πρέπει να ξανά-ανέβει στο εκάστοτε store (αφού εγκριθεί κτλ), να την ξανά-κατεβάσει και να την ξανά-κάνει install ο εκάστοτε χρήστης (στην περίπτωση της web app το update είναι πολύ πιο άμεσο). Τα πλεονεκτήματα των Native εφαρμογών όμως είναι πολλά και είναι ο λόγος που συνεχίζουν να υλοποιούνται και να είναι η βασικότερη μορφή εφαρμογής σε κινητές συσκευές. Ένα από τα βασικότερα πλεονεκτήματα είναι ότι η κάθε εφαρμογή έχει φτιαχτεί ειδικά για την συγκεκριμένη συσκευή και λειτουργικό, πράγμα που σημαίνει πως εκμεταλλεύεται πολύ καλά όλα τα advanced χαρακτηριστικά της εκάστοτε συσκευής και αλληλεπιδρά με το λειτουργικό σύστημα κάνοντας χρήση των διαθέσιμων διεπαφών με τη χρήση των διεπαφών προγραμματισμού εφαρμογών (API, Application Programming Interface) που παρέχει το κάθε λειτουργικό σύστημα για την ανάπτυξη και λειτουργία των εφαρμογών. Οι διεπαφές προγραμματισμού παρέχουν τη δυνατότητα σε μια native εφαρμογή να αλληλεπιδρά με την οθόνη αφής,

το πληκτρολόγιο, να συνδέεται σε δίκτυα, να επεξεργάζεται ηχητικά αρχεία που λήφθηκαν μέσω του μικροφώνου, να παράγει ήχους μέσω των ηχείων και να ελέγχει την κάμερα για λήψη φωτογραφιών ή βίντεο, το GPS, την πυξίδα, το γυροσκόπιο, κτλ.. Αυτά είναι μερικά από τα παραδείγματα από τις δυνατότητες που παρέχουν τα παραπάνω εργαλεία, αφού στο σύνολό τους παρέχουν πρόσβαση σε οποιοδήποτε χαρακτηριστικό της συσκευής. Σε συνδυασμό με την πρόσβαση στα τεχνικά χαρακτηριστικά της συσκευής, το λειτουργικό σύστημα παρέχει μεθόδους αλληλεπίδρασης και με υπηρεσίες υψηλότερου επιπέδου, οι οποίες είναι απαραίτητες για την παροχή υψηλού επιπέδου εμπειρίας στο χρήστη. Οι διεπαφές προγραμματισμού υψηλού επιπέδου προσφέρουν μια σειρά από υπηρεσίες όπως η πλοήγηση στο διαδίκτυο, η πρόσβαση στο ημερολόγιο, τις επαφές, το φωτογραφικό υλικό και φυσικά η δυνατότητα τηλεφωνικών κλήσεων, αποστολής και λήψης γραπτών μηνυμάτων. Οι υπηρεσίες αυτές συνήθως υλοποιούνται από εφαρμογές που είναι προεγκατεστημένες σχεδόν σε όλα τα λειτουργικά συστήματα. Το σημαντικό είναι ότι σε μια native εφαρμογή γίνεται να συνδυαστούν πολλές από τις παραπάνω λειτουργίες, καθώς με προγραμματιστικό τρόπο δίνεται η δυνατότητα πρόσβασης της εφαρμογής σε αυτές τις υπηρεσίες.

Οι προγραμματιστικές διεπαφές που απαιτούνται για την πλήρη αξιοποίηση της κινητής συσκευής και κατ' επέκταση του λειτουργικού συστήματος, είναι συγκεκριμένες και στενά συνδεδεμένες με το λειτουργικό σύστημα για το οποίο δημιουργήθηκαν. Το γεγονός αυτό προσθέτει πολυπλοκότητα και κόστος στην ανάπτυξη μιας native εφαρμογής με παρουσία σε διάφορα λειτουργικά συστήματα, αφού για κάθε λειτουργικό σύστημα απαιτείται η ανάπτυξη της ίδιας εφαρμογής χρησιμοποιώντας διαφορετικές προγραμματιστικές διεπαφές. Παρόλα αυτά ο ρόλος τους είναι ιδιαίτερα σημαντικός, αφού είναι εκείνες που καθιστούν εφικτή την ανάπτυξη ιδιαίτερα πολύπλοκων εφαρμογών. Επίσης στις native εφαρμογές είναι διαθέσιμο ένα σύνολο εργαλείων γραφικής διεπαφής. Το κάθε λειτουργικό σύστημα παρέχει ένα σύνολο από τα βασικά γραφικά συστατικά όπως κουμπιά, μενού, tab bars, ειδοποιήσεις και πολλά ακόμη. Εφαρμογές που κάνουν χρήση αυτών των γραφικών συστατικών κληρονομούν την εμφάνιση του συγκεκριμένου λειτουργικού συστήματος στο οποίο εγκαθίσταται, με αποτέλεσμα η εμπειρία που εισπράττει ο χρήστης να είναι μέσα στα πλαίσια που έχει συνηθίσει. Αυτό αυξάνει τον βαθμό δυσκολίας ανάπτυξης των εφαρμογών γιατί πλέον ο προγραμματιστής πρέπει να

λάβει υπ' όψιν του κατά την σχεδίαση της διεπαφής χρήστη την μεγάλη ποικιλία μεγεθών και αναλύσεων που μπορεί η εφαρμογή του να συναντήσει, η τεχνικές πλοήγησης είναι διαφορετικές οι εμφάνιση των στοιχείων είναι διαφορετική αλλά το πιο σημαντικό, οι συνήθειες των χρηστών είναι διαφορετικές. Ποτέ δεν πρέπει, ένας προγραμματιστής όταν φτιάχνει μια εφαρμογή για παράδειγμα στο περιβάλλον Android η οποία υπάρχει ήδη για κάποια άλλη πλατφόρμα, να προσπαθεί να μεταφέρει ακριβώς την διεπαφή χρήστη. Είναι λάθος τακτική η προσπάθεια αναπαραγωγής μιας διεπαφής από την μια πλατφόρμα στην άλλη. Οι χρήστες πρέπει να νιώθουν «σαν στο σπίτι τους» όταν χρησιμοποιούν μια εφαρμογή στην συσκευή τους, έτσι αυτή πρέπει να έχει εμφάνιση (το ονομαζόμενο look & feel) που συμβαδίζει με την γενικότερη φιλοσοφία του Android και της εκάστοτε πλατφόρμας. Για παράδειγμα, δεν θέλουμε μια εφαρμογή για Android να θυμίζει το iOS της Apple! Άλλωστε η φιλοσοφία των εφαρμογών διαφέρει κατά πολύ στις δυο πλατφόρμες. Έτσι κάθε λειτουργικό σύστημα αποτελείται από τα δικά του ιδιαίτερα γραφικά εργαλεία. Ακόμη και μεταξύ διαφορετικών εκδόσεων του ίδιου λειτουργικού παρατηρούνται διαφορές. Η γραφική διεπαφή είναι ένα πολύ σημαντικό κομμάτι για την επιτυχία του λειτουργικού συστήματος με αποτέλεσμα να υπάρχουν συνέχεια βελτιώσεις, άρα και διαφοροποιήσεις. Η διαφοροποίηση που παρατηρείται ανάμεσα σε αυτά τα εργαλεία καθιστούν απαραίτητη την εξοικείωση του σχεδιαστή, της εφαρμογής, με τα εργαλεία και τα συστατικά που παρέχονται [2].

Στα συν των Native εφαρμογών συγκαταλέγεται ότι το user interface και η απόκρισή είναι πολύ πιο γρήγορα και συνήθως πιο εντυπωσιακά από τα αντίστοιχα μιας web εφαρμογής. Ένα άλλο σημαντικό πλεονεκτήματα είναι πως δεν χρειάζεται σύνδεση στο internet για να χρησιμοποιηθεί η εφαρμογή (εκτός φυσικά και αν αυτό είναι requirement της ίδιας της εφαρμογής).

Στην περιγραφή των native εφαρμογών έγινε αισθητό το μέγεθος κατακερματισμού των λειτουργικών συστημάτων που υπάρχει στο πεδίο της ανάπτυξης εφαρμογών για κινητές συσκευές. Λύση σε αυτό το πρόβλημα προσπαθεί να δώσει η κατηγορία των διαδικτυακών εφαρμογών παρέχοντας την δυνατότητα ανάπτυξης εφαρμογών που θα εκτελούνται σε διαφορετικά λειτουργικά συστήματα και συσκευές χρησιμοποιώντας αποκλειστικά τεχνολογίες διαδικτύου (π.χ. HTML, HTML 5, CSS και Javascript, Adobe Flash, κλπ). Με τον τρόπο αυτό οι προγραμματιστές αποφεύγουν το τεράστιο εμπόδιο της εκμάθησης νέων γλωσσών

προγραμματισμού για κάθε διαφορετικό λειτουργικό σύστημα για το οποίο δημιουργούν μια εφαρμογή. Το γεγονός ότι το πεδίο των διαδικτυακών εφαρμογών διαθέτει ήδη αρκετά εργαλεία ανάπτυξης δε σημαίνει ότι αυτά μπορούν να χρησιμοποιηθούν στο χώρο των κινητών συσκευών χωρίς περαιτέρω εξέλιξη. Οι κινητές συσκευές, όπως περιγράφεται παραπάνω, έχουν ιδιαίτερα χαρακτηριστικά που καθιστούσαν τις υπάρχουσες τεχνολογίες ανεπαρκείς. Η πραγματική αφετηρία των διαδικτυακών εφαρμογών ήταν η προσέλευση της HTML5 και η συνεχής εξέλιξη των περιηγητών οι οποίοι πλέον μπορούν να εκμεταλλευτούν αυτές τις νέες δυνατότητες.

Το κυριότερο μειονέκτημα των διαδικτυακών εφαρμογών είναι ότι αντίθετα με τις native εφαρμογές, οι οποίες έχουν ως περιβάλλον εκτέλεσης το λειτουργικό σύστημα και άρα συνδέονται άμεσα με αυτό, οι διαδικτυακές εφαρμογές εκτελούνται στον περιηγητή ιστού (browser). Ο περιηγητής ιστού είναι και αυτός με την σειρά του, μία native εφαρμογή, άρα έχει άμεση πρόσβαση στα APIs του λειτουργικού συστήματος, όμως μόνο μερικά από αυτά είναι διαθέσιμα στις εφαρμογές που εκτελούνται σε αυτόν. Ενώ οι native εφαρμογές έχουν πλήρη πρόσβαση στη συσκευή στην οποία εκτελούνται, οι web εφαρμογές έχουν περιορισμένη πρόσβαση σε πολλά χαρακτηριστικά της συσκευής και σε κάποιες περιπτώσεις η πρόσβαση απουσιάζει τελείως. Η απουσία πρόσβασης οφείλεται σε δύο κυρίως λόγους. Ο πρώτος είναι η ασφάλεια της συσκευής. Οι native εφαρμογές είναι διαθέσιμες μέσω διαφόρων marketplaces τα οποία πριν διαθέσουν μια εφαρμογή στους χρήστες, εξετάζουν τις λειτουργίες της και ελέγχουν εάν διαθέτει χαρακτηριστικά που θα μπορούσαν να είναι επιβλαβή για την συσκευή ή για τα ευαίσθητα προσωπικά δεδομένα του χρήστη. Στις web εφαρμογές αυτός ο έλεγχος απουσιάζει αφού είναι διαθέσιμες μέσω οποιουδήποτε εξυπηρετή (server). Ο δεύτερος λόγος είναι η απουσία υλοποίησης διαφόρων προγραμματιστικών διεπαφών, έχοντας ως αποτέλεσμα την απουσία πρόσβασης σε διάφορες λειτουργίες της συσκευής. Ένα άλλο αρνητικό χαρακτηριστικό που πρέπει να αναφερθεί είναι ότι μια διαδικτυακή εφαρμογή μπορεί να είναι όσο αποδοτική της επιτρέπει ο περιηγητής. Όπως αναφέρθηκε νωρίτερα, οι διαδικτυακή προσέγγιση χρησιμοποιεί τον περιηγητή ιστού ως περιβάλλον εκτέλεσης. Το γεγονός αυτό επηρεάζει αρνητικά την απόδοση αφού η απόδοση της εφαρμογής δεν εξαρτάται μόνο από την επεξεργαστική ισχύ της συσκευής στην οποία εκτελείται, αλλά και από τον ίδιο τον περιηγητή.

1.3 Σύγκριση μεταξύ Εγγενών και Υβριδικών Εφαρμογών

Η άμεση πρόσβαση στα APIs των λειτουργικών συστημάτων επιτρέπουν την ανάπτυξη εγγενών εφαρμογών που εκμεταλλεύονται στο έπακρο τις δυνατότητες των συσκευών. Αυτό σημαίνει ότι, εκτός από τη δυνατότητα χρήσης κάθε υποσυστήματος και λειτουργικότητας της πλατφόρμας, οι εγγενείς εφαρμογές έχουν τη βέλτιστη απόκριση και ταχύτητα, καθώς δεν διαμεσολαβεί κάποιο επιπλέον αφαιρετικό επίπεδο για την υλοποίηση. Επιπλέον η χρήση των έτοιμων γραφικών διεπαφών που παρέχονται από κάθε πλατφόρμα, σε συνδυασμό με τις σχεδιαστικές οδηγίες που προτείνονται σε κάθε νέα έκδοση των λειτουργικών συστημάτων, προσφέρουν βελτιωμένη χρηστικότητα και εμπειρία χρήσης (UX User Experience)[3].

Η δυσκολία υλοποίησης είναι το μεγαλύτερο μειονέκτημα των εγγενών εφαρμογών. Το πρόβλημα δεν είναι μόνο σχετικό με το κόστος και το χρόνο. Το θέμα είναι ότι συνήθως πρέπει να αναπτυχθεί ακριβώς η ίδια εφαρμογή, με την ίδια λειτουργικότητα και τις ίδιες δυνατότητες από διαφορετικές ομάδες. Το πρόβλημα δηλαδή είναι η επικοινωνία των ομάδων στην ανάπτυξη κοινής εφαρμογής σε διαφορετικές αρχιτεκτονικές, με διαφορετικές γλώσσες προγραμματισμού και διαφορετικά εργαλεία. Γι' αυτό άλλωστε και οργανισμοί με πλούσιο ανθρώπινο δυναμικό και τεράστιο χρηματικό αποθεματικό προσπαθούν να βρουν λύσεις στην ανάπτυξη εφαρμογών για τις δύο πλατφόρμες με κοινό κώδικα.

Αυτό ακριβώς είναι και το πλεονέκτημα των μεταγλωττισμένων ανά πλατφόρμα εφαρμογών. Με αυτή την τεχνική στην ουσία αναπτύσσεται μία εφαρμογή που εκτελείται και στα δύο συστήματα. Ο τρόπος που αυτό γίνεται διαφέρει σε κάθε προγραμματιστικό εργαλείο, όπως και το ποσοστό του κώδικα που μπορεί να επαναχρησιμοποιηθεί ή το πόσο κοντά στην “εγγενή” όψη είναι το τελικό αποτέλεσμα. Σίγουρο όμως είναι ότι αυτές οι τεχνικές έχουν μεγάλη ανάπτυξη τα τελευταία χρόνια, ωριμάζουν, βελτιώνονται, τεκμηριώνονται καλύτερα και χρησιμοποιούνται όλο και περισσότερο.

Μεγάλη ανάπτυξη επίσης έχουν και οι διαδικτυακές εφαρμογές. Αυτό οφείλεται κυρίως στην αλματώδη ανάπτυξη της πλατφόρμας στην οποία στηρίζονται, δηλαδή στους φυλλομετρητές ιστού. Επιπλέον νέες τεχνολογίες όπως οι εφαρμογές PWA (Progressive Web Apps) προσπαθούν να μετριάσουν τα μειονεκτήματα των web εφαρμογών προσομοιώνοντας εν μέρη την εμφάνιση των εφαρμογών και

βελτιώνοντας την ταχύτητα και την απόκριση τους ακόμα και με μικρές ταχύτητες διαδικτύου και χρησιμοποιώντας δυνατότητες που παλαιότερα δεν ήταν δυνατές όπως ειδοποιήσεις και λειτουργία εκτός σύνδεσης.

Σε κάθε περίπτωση όμως οι web εφαρμογές στερούνται δύο σημαντικών παραγόντων. Το πρώτο είναι η δυνατότητα ανακαλυψιμότητας (Discoverability) που προσφέρουν τα app stores και δεύτερο ότι δεν έχουν πρόσβαση σε όλα τα χαρακτηριστικά των συσκευών. Οι υβριδικές εφαρμογές από την πλευρά τους εγκαθίστανται μέσω των app stores, έχουν πρόσβαση στα υποσυστήματα και στους αισθητήρες των συσκευών αλλά στερούνται την ομαλότητα, την ομοιομορφία και γενικά την εμπειρία χρήσης που προσφέρουν οι καλογραμμένες εγγενείς εφαρμογές.

Βλέπουμε λοιπόν ότι όλες οι τεχνικές έχουν τα πλεονεκτήματα και τα μειονεκτήματά τους. Στη δική μας περίπτωση, που είναι η ανάπτυξη μιας απλής εφαρμογής χωρίς περίπλοκο περιβάλλον χρήστη και χωρίς ιδιαίτερες απαιτήσεις από τα υποσυστήματα των συσκευών, προτιμήθηκε η ανάπτυξη με βάση το υβριδικό μοντέλο που προσφέρει εύκολη και γρήγορη υλοποίηση με τα επιπλέον θετικά της εγκατάστασης μέσω των app stores και τη χρήση ορισμένων πρόσθετων δυνατοτήτων της συσκευής όπως το GPS μέσω πρόσθετου (plugin) [4].



Εικόνα 1: Τα 3 βασικά λειτουργικά Συστήματα για Mobile Συσκευές (πηγή: nectarbits.com)

ΚΕΦΑΛΑΙΟ 2: Native Mobile Apps (Android, iOS)

Native εφαρμογές καλούνται οι εφαρμογές οι οποίες έχουν δημιουργηθεί για έναν δεδομένο τύπο συσκευής/λειτουργικού συστήματος χρησιμοποιώντας την προκαθορισμένη από τον κατασκευαστή γλώσσα προγραμματισμού και τα εργαλεία δημιουργίας εφαρμογών (SDK – software development kit), τα οποία ο καθένας παρέχει, δηλαδή Objective-C και xCode για εφαρμογές iOS, Java και Eclipse για εφαρμογές Android, C# και Visual Studio για εφαρμογές των Windows Phone κ.α. Η πρόσβαση σε αυτές γίνεται πατώντας πάνω στο αντίστοιχο εικονίδιο στην αρχική «οθόνη» της συσκευής. Ο χρήστης μπορεί να κατεβάσει τις εφαρμογές αυτές από το ηλεκτρονικό «κατάστημα» εφαρμογών, το οποίο έχει δημιουργήσει η κάθε εταιρία. Το γεγονός ότι οι native εφαρμογές γράφονται με τη γλώσσα που είναι η προκαθορισμένη για την εκάστοτε συσκευή, τις κάνει να είναι πολύ γρήγορες και αξιόπιστες, ενώ ταυτόχρονα μπορούν να εκμεταλλευτούν πλήρως όλες τις δυνατότητες που προσφέρει η συσκευή, όπως η φωτογραφική μηχανή, το GPS, η πυξίδα, το επιταχυνσιόμετρο, η λίστα επαφών ή να ενημερώνει τον χρήστη για διάφορα γεγονότα, όταν η εφαρμογή είναι κλειστή με push notifications κ.α. Δημιουργώντας εφαρμογές με τη γλώσσα, τα εργαλεία και ακολουθώντας την φιλοσοφία στην ανάπτυξη εφαρμογών της κάθε πλατφόρμας, οι εφαρμογές είναι εύχρηστες, γρήγορες, ευχάριστες για τον χρήστη και ταιριαστές με το περιβάλλον της κάθε συσκευής. Η native ανάπτυξη εφαρμογών έχει και τα αρνητικά της βέβαια. Κατ' αρχάς η εκάστοτε εφαρμογή μπορεί να λειτουργήσει μόνο στην πλατφόρμα για την οποία έχουν δημιουργηθεί, παραδείγματος χάριν, μια iOS εφαρμογή σε Objective-C μπορεί να λειτουργήσει μόνο σε iPhone, iPad και σε άλλα προϊόντα της Apple που έχουν το λειτουργικό iOS. Αυτό συνεπάγεται ότι στην περίπτωση που είναι αναγκαίο η εφαρμογή να παίξει και σε άλλη πλατφόρμα - ως συνέχεια του προηγούμενου παραδείγματος - απαιτείται η εφαρμογή να «παίξει» και σε συσκευές με λειτουργικό Android, πρέπει η εφαρμογή να σχεδιαστεί εκ νέου με τις προδιαγραφές της νέας πλατφόρμας - στην προκειμένη του Android - και να ξαναγραφεί στη native γλώσσα προγραμματισμού, που για το παράδειγμά μας είναι η Java, με αποτέλεσμα το κόστος εξέλιξης της εφαρμογής να γίνεται πολύ υψηλό. Επιπλέον οι εφαρμογές αυτές, προκειμένου να γίνουν διαθέσιμες στο κοινό μέσα από τους ηλεκτρονικούς χώρους πώλησης/διανομής εφαρμογών της κάθε εταιρίας, πρέπει πρώτα να εγκριθούν από αυτές με σκοπό οι εφαρμογές να τηρούν τις νόρμες που η κάθε εταιρία ορίζει.

Οι Native εφαρμογές βρίσκονται στη συσκευή και είναι προσπελάσιμες μέσω εικονιδίων στην αρχική οθόνη της συσκευής. Οι Native εφαρμογές εγκαθίστανται μέσω ενός κατάστηματος εφαρμογών (όπως το Google Play ή το App Store της Apple). Έχουν αναπτυχθεί ειδικά για μία πλατφόρμα και μπορούν να επωφεληθούν πλήρως από όλες τις λειτουργίες της συσκευής καθώς μπορούν να χρησιμοποιήσουν την κάμερα, το GPS, το επιταχυνσιόμετρο, την πυξίδα, τη λίστα των επαφών, και ούτω καθεξής. Επιπλέον, οι native apps μπορούν να χρησιμοποιήσουν το σύστημα ειδοποίησης της συσκευής και έχουν τη δυνατότητα να λειτουργήσουν χωρίς σύνδεση. Δεν υπάρχουν περιορισμοί κατα τη διάρκεια του προγραμματισμού, ενώ ακόμα έχουν υψηλότερη ταχύτητα από τις άλλες εφαρμογές. Οι πιθανότητες για σφάλματα είναι μικρότερες αλλά και τα ετήσια updates είναι σπάνια και δεν διαρκούν αρκετό χρονικό διάστημα όπως σε άλλα είδη εφαρμογών. Τέλος, πωλούνται σε πολλές αγορές όπως iTunes, Android Market κ.λπ [5].

Οι Native εφαρμογές λειτουργούν στις κινητές συσκευές και έχουν αναπτυχθεί για συγκεκριμένο τύπο συσκευής με συγκεκριμένα χαρακτηριστικά με συγκεκριμένη πλατφόρμα και συγκεκριμένο framework το οποίο συνήθως υποδεικνύεται από τον κατασκευαστή. Παραδείγματος χάρη Objective C και Xcode για το Apple IOS, Java και Android Studio ή Eclipse για το Google Android και C# και Visual Studio για το Microsoft Windows Phone. Οι εφαρμογές αυτές μπορεί να είναι προ εγκατεστημένες μαζί με το λειτουργικό σύστημα της συσκευής ή να διανέμονται μέσω από το αντίστοιχο application store του λειτουργικού.

Αρνητικό για τις εφαρμογές αυτές αποτελεί το γεγονός πως απαιτείται απαραίτητα να υλοποιηθεί κώδικας σε δύσκολες γλώσσες όπως (Objective,Java) κάτι που είναι χρονοβόρο και επίζημιο ιδιαίτερα απαιτείται η εφαρμογή να έχει συμβατότητα με πολλαπλές φορητές συσκευές και πλατφόρμες. Υψηλό κόστος απαιτεί και η συντήρησή τους καθώς και μία πιθανή ενημέρωση ιδιαίτερα όταν χρησιμοποιείται η εφαρμογή από πολλές πλατφόρμες. Η διαδικασία να πάρει το app έγκριση στο app store μπορεί να αποδειχθεί μακρά και κουραστική για τον δημιουργό του και δεν είναι δεδομένο ότι θα στεφθεί με επιτυχία. Επίσης, δεν υπάρχει καμία εγγύηση ότι η εφαρμογή γίνεται αμέσως δημοφιλής στους χρήστες. Οι χρήστες των διαφορετικών κινητών μπορούν να χρησιμοποιούν διαφορετικές εκδόσεις της εφαρμογής κάτι που καθιστά δύσκολο για τον δημιουργό να συνεχίσει να προσφέρει

την υποστήριξή της. Τέλος, ο δημιουργημένος κώδικας δεν είναι σε θέση να χρησιμοποιηθεί ταυτόχρονα σε Android και iPhone συσκευές.

2.1 Ανάπτυξη Android εφαρμογών με Java (Android Studio)

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικροσυσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για την Sun, έκανε ήδη πειραματισμούς πάνω στη C++ και είχε παρουσιάσει κατά καιρούς κάποιες πειραματικές γλώσσες ως πρότυπα για το νέο εργαλείο που αναζητούσαν στη Sun. Τελικά, μετά από λίγο καιρό κατέληξαν σε μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσά Oak. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα. Η Oak ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με την C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστραφή (object oriented) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα, οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα Oak ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που εκτός των άλλων ήταν το όνομα της αγαπημένης ποικιλίας καφέ για τους δημιουργούς της. Η επίσημη εμφάνιση της Java, αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Ο πρώτος μεταγλωττιστής (compiler) της ήταν γραμμένος στη γλώσσα C από τον James Gosling. Το 1994, ο A.Van.Hoff ξαναγράφει τον μεταγλωττιστή της γλώσσας σε Java, ενώ το Δεκέμβριο του 1995 πρώτες οι IBM, Borland, Mitsubishi Electronics, Sybase και Symantec ανακοινώνουν σχέδια να χρησιμοποιήσουν τη Java για την δημιουργία λογισμικού. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006, η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα όσον αφορά το μεταγλωττιστή (javac) και το πακέτο ανάπτυξης (JDK, Java Development Kit). Στις 27 Απριλίου 2010, η εταιρία λογισμικού Oracle Corporation ανακοίνωσε ότι μετά από πολύμηνες

συζητήσεις ήρθε σε συμφωνία για την εξαγορά της Sun Microsystems και των τεχνολογιών (πνευματικά δικαιώματα/ πατέντες) που η δεύτερη είχε στην κατοχή της ή δημιουργήσει. Η συγκεκριμένη συμφωνία θεωρείται σημαντική για το μέλλον της Java και του γενικότερου οικοσυστήματος τεχνολογιών γύρω από αυτή μιας και ο έμμεσος έλεγχος της τεχνολογίας και η εξέλιξη της περνάει σε άλλα χέρια [6].

Σύμφωνα με την περιγραφή που έχει δοθεί από την εταιρεία Sun για τη γλώσσα προγραμματισμού Java, η τελευταία είναι μια απλή, αντικειμενοστραφής, καταναμημένη, υψηλής απόδοσης, συμπαγής, ασφαλής, ανεξάρτητης αρχιτεκτονικής, μεταφέρσιμη, υψηλής απόδοσης και ιδιαίτερα δυναμική. Η Java αποτελεί μια απλή γλώσσα καθώς παρουσιάζει αρκετές ομοιότητες με τις γλώσσες C και C++ οι οποίες είναι ευρέως διαδομένες, διευκολύνοντας έτσι την εκμάθησή της, ενώ ταυτόχρονα περιέχει λίγες προγραμματιστικές δομές και ιδιαίτερα καλά ορισμένη σημασιολογία. Δεδομένου ότι είναι αντικειμενοστραφής γλώσσα εστιάζει στον ορισμό αντικειμένων και των αντίστοιχων λειτουργιών τους. Στη java η έννοια της κλάσης, η οποία περιγράφει μια συλλογή δεδομένων και τις λειτουργίες που αυτά επιδέχονται, διαθέτει ιδιαίτερη σημασία. Κάθε μια από τις κλάσεις προέρχεται από μια άλλη μέσω της κληρονομικότητας μεταξύ των κλάσεων (inheritance), ορίζοντας κατ' αυτό τον τρόπο μια ιεραρχία κλάσεων στη κορυφή της οποίας υφίσταται η αρχική κύρια κλάση που είναι το αντικείμενο (object). Τα αντικείμενα μιας κλάσης χρησιμοποιούνται σε ένα πρόγραμμα Java και δημιουργούνται κατά τη διάρκεια εκτέλεσης του προγράμματος. Ακόμα, η Java χαρακτηρίζεται ως καταναμημένη γλώσσα καθώς επιτρέπει την επικοινωνία με αντικείμενα τα οποία βρίσκονται σε απομακρυσμένες θέσεις στο δίκτυο, ενώ ταυτόχρονα επιτρέπει την επικοινωνία με άλλες εφαρμογές μέσω διαδικτυακών συνδέσεων. Εκτός από καταναμημένη, θεωρείται και ερμηνευόμενη. Αυτό συμβαίνει καθώς ο μεταγλωττιστής της Java δεν παράγει έναν τελικό κώδικα για ένα συγκεκριμένο υπολογιστή. Αντίθετα, παράγει κάτι ενδιάμεσο σε μορφή bytes το οποίο ονομάζεται bytecode. Αυτός ο κώδικας περνά στη συνέχεια από τον διερμηνέα της Java. Αυτή η διαδικασία γίνεται προκειμένου ο κώδικας να έχει τη δυνατότητα να εκτελεστεί σε πολλά διαφορετικά περιβάλλοντα υπολογιστών, εφόσον βέβαια ο εν λόγω διερμηνέας είναι διαθέσιμος σε αυτά. Η Java θεωρείται ως μια συμπαγής γλώσσα προγραμματισμού. Αυτό συμβαίνει καθώς διαθέτει ένα ισχυρό σύστημα τύπων, το οποίο επιτρέπει εκτενείς ελέγχους κατά τη διάρκεια μετάφρασης των προγραμμάτων. Με τον τρόπο αυτό, η Java συμβάλλει στην ανάπτυξη

αξιόπιστου και συμπαγούς λογισμικού. Εκτός των άλλων η Java χαρακτηρίζεται ως ασφαλής γλώσσα. Αυτό συμβαίνει καθώς τα προγράμματα που έχουν αναπτυχθεί σε αυτή, μπορούν να χρησιμοποιηθούν από μεγάλο εύρος χρηστών καθένας από τους οποίους διαθέτει διαφορετικά δικαιώματα πρόσβασης, με αποτέλεσμα να μην είναι δυνατό να υπάρξουν ανεπιθύμητες παρενέργειες στο σύστημα. Το πλεονέκτημα αυτό, προσδίδεται από τη χρήση ενός ενδιάμεσου κώδικα (bytecode verifier) ο οποίος εντοπίζει τυχόν περίεργες ενέργειες οι οποίες είναι δυνατό να επηρεάσουν αρνητικά το περιβάλλον εργασίας του χρήστη. Συνεχίζοντας, η Java αποτελεί μια γλώσσα ανεξάρτητης αρχιτεκτονικής καθώς διαθέτει έναν ενδιάμεσο κώδικα, ο οποίος δεν αναφέρεται σε ένα συγκεκριμένο τύπο υπολογιστή αλλά αντίθετα μεταφράζεται κατάλληλα με τη βοήθεια του διερμηνέα. Επίσης, εξαιτίας του παραπάνω χαρακτηριστικού της java, δηλαδή της ανεξαρτησίας της αρχιτεκτονικής, τα προγράμματα σε java καθίστανται μεταφέρσιμα καθώς προσαρμόζονται σε διάφορους τύπους υπολογιστών διατηρώντας αμετάβλητους τους αρχικούς τύπους δεδομένων. Αυτό σημαίνει ότι ανεξάρτητα από την αρχιτεκτονική που ακολουθούν όμοια προγράμματα, θα δώσουν ίδια αποτελέσματα για ίδιες αρχικές τιμές των παραμέτρων τους.

Η java αποτελεί μία γλώσσα με αρκετά υψηλή απόδοση. Παρόλα αυτά, δεν είναι δυνατό να φτάσει την απόδοση των γλωσσών προγραμματισμού C και C++ καθώς οι τελευταίες υλοποιούνται απευθείας σε μεταγλωττιστές. Ωστόσο, έχουν κατασκευαστεί μεταγλωττιστές της τελευταίας στιγμής (just-in-time compilers) οι οποίοι εμπεριέχουν διερμηνείς και βελτιώνουν ιδιαίτερα την απόδοση εκτέλεσης των προγραμμάτων Java. Ακόμη, υποστηρίζει πολλαπλά νήματα εκτέλεσης. Αυτό σημαίνει ότι δίνει τη δυνατότητα ταυτόχρονης εκτέλεσης πολλών διεργασιών, γεγονός που είναι ιδιαίτερα αποδοτικό. Έτσι, λοιπόν, φαίνεται ότι η Java αποτελεί μια δυναμική γλώσσα, η οποία έχει σχεδιαστεί προκειμένου να προσαρμόζεται σε ένα δυναμικά εξελισσόμενο περιβάλλον. Αυτό σημαίνει ότι οι κλάσεις που πρέπει να εκτελεστούν στο πρόγραμμα, είναι δυνατό να βρίσκονται σε κάποιο άλλο μέρος του δικτύου και να μεταφερθούν δυναμικά προκειμένου να εκτελεστούν τοπικά. Είναι φανερό, επομένως, ότι οι κλάσεις δεν είναι απαραίτητο να ενσωματωθούν στο πρόγραμμα κατά τη διάρκεια της μετάφρασής τους. Τέλος, η Java υποστηρίζει τη μεταφορά του εκτελέσιμου περιεχομένου σε εφαρμογές πολυμέσων. Αυτό σημαίνει, ότι υπάρχει πλέον η δυνατότητα εκτέλεσης των προγραμμάτων στο περιβάλλον του

χρήστη με αποτέλεσμα να αυξάνονται οι δυνατότητες των χρηστών, γεγονός που ενισχύει ακόμα περισσότερο τη δυναμικότητα της γλώσσας [7].

Το Android [8] είναι ένα λειτουργικό σύστημα για έξυπνες συσκευές, το οποίο βασίζεται στον πυρήνα του Linux και πλέον εξελίσσεται από τη Google. Αν και αρχικά είχε σχεδιαστεί μόνο για smartphones και tablets, πλέον χρησιμοποιείται και σε παιχνίδια - μηχανές, ψηφιακές κάμερες, ηλεκτρονικούς υπολογιστές (όπως ο HP Slate 21) και άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα για φορητές συσκευές και μάλιστα οι συσκευές που χρησιμοποιούν android είναι περισσότερες από τις συσκευές που χρησιμοποιούν iOS, OSX και Windows μαζί. Ο πηγαίος κώδικας διατίθεται από την Google με άδειες ανοιχτού λογισμικού. Παρ' όλα αυτά, οι περισσότερες συσκευές αποτελούνται από έναν συνδυασμό ελεύθερου λογισμικού, καθώς και εξειδικευμένου για την συσκευή λογισμικού. Αποτελεί έναν ιδιαίτερα δημοφιλή τρόπο στις εταιρίες τεχνολογίας που χρειάζονται ένα φθινό και αξιόπιστο λειτουργικό σύστημα για τις συσκευές τους. Ο ελεύθερος χαρακτήρας του android έχει ενθαρρύνει τους προγραμματιστές, τους λάτρεις της τεχνολογίας και φυσικά τις εταιρίες να ασχοληθούν με τη δημιουργία εφαρμογών για συσκευές android, οι οποίες είναι διαθέσιμες στο κοινό μέσω του Google Play Store. Η πλειοψηφία αυτών δημιουργείται με τη γλώσσα προγραμματισμού Java χρησιμοποιώντας το Android SDK (Software Development Kit), το οποίο εκτός από βιβλιοθήκες περιέχει και debugger (εργαλείο για την εύρεση σφαλμάτων) και εξομοιωτή. Επίσης δημιουργία εφαρμογών μπορεί να γίνει με C και C++ χρησιμοποιώντας το Native Development Kit, καθώς και με το Google Inventor, το οποίο είναι ένα γραφικό περιβάλλον δημιουργίας εφαρμογών και απευθύνεται σε αρχαίους.

Η εταιρία Android δημιουργήθηκε στο Palo Alto της California το 2003 με σκοπό να δημιουργήσει ένα λειτουργικό σύστημα για “έξυπνες” συσκευές. Αρχικά στόχευαν στην δημιουργία ένα λειτουργικού συστήματος για ψηφιακές φωτογραφικές μηχανές, μέχρι που είδαν ότι η αγορά για αυτές δεν ήταν αρκετά μεγάλη, οπότε και στράφηκαν στα κινητά τηλέφωνα έχοντας σαν ανταγωνισμό το Symbian και τα Windows Phones. Το 2005 η Google εξαγόρασε την Android και ξεκίνησε τη δημιουργία μιας πλατφόρμας για “έξυπνες” συσκευές βασισμένη στον πυρήνα του Linux και σύντομα συνεργάστηκε με πλήθος κατασκευαστών hardware. Στις 5 Νοεμβρίου του 2007 η Google μαζί με κατασκευαστές κινητών τηλεφώνων,

παρόχους κινητής τηλεφωνίας και κατασκευαστές ολοκληρωμένων κυκλωμάτων παρουσίασαν το android, ενώ στις 22 Οκτωβρίου του 2008 παρουσιάστηκε η πρώτη συσκευή με λειτουργικό android το HTC Dream. Το 2010 η google σε συνεργασία με την HTC δημιούργησε το πρώτο smartphone της σειράς Nexus, το Nexus One, η οποία συνεχίστηκε σε συνεργασία με την LG, με τις συσκευές, Nexus 4 , Nexus 5 και Nexus 6 και ταυτόχρονα σε συνεργασία με την Asus δημιούργησε tablet.

Η ιστορία εκδόσεων του λειτουργικού συστήματος των κινητών ξεκίνησε με την κυκλοφορία του Android beta το Νοέμβριο του 2007. Η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance (OHA), και έχουν γίνει μια σειρά από ενημερώσεις στην λειτουργία του συστήματος από την αρχική κυκλοφορία του. Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα. Στη συνέχεια, παρατίθεται πίνακας, ο οποίος παρουσιάζει τις διάφορες εκδόσεις του λογισμικού που έχουν κυκλοφορήσει, με τις αντίστοιχες ονομασίες τους και τις ημερομηνίες κυκλοφορίας τους στην αγορά

Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
N/A	1.0	23 Σεπτεμβρίου 2008	1
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5–7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9–10
Honeycomb	3.0 – 3.2.6	22 Φεβρουάριου 2011	11–13
IceCream	4.0 – 4.0.4	18 Οκτωβρίου	14–15

Sandwich		2011	
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16–18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19–20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21–22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 - 7.1.1	22 Αυγούστου 2016	24
Oreo	8.0	21 Αυγούστου 2017	26

Πίνακας 1: Οι εκδόσεις του Android (πηγή: official Android website)

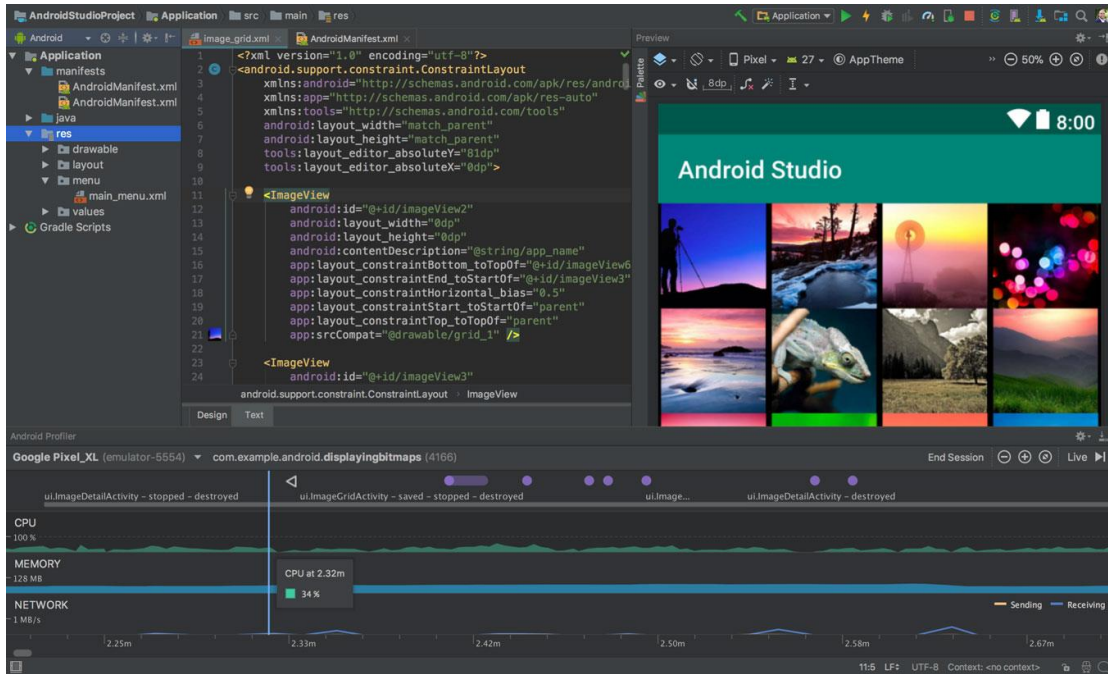
Το περιβάλλον ανάδρασης του Android βασίζεται στην άμεση χειραγώγηση, καθώς χρησιμοποιεί ως εντολές εισόδου στο λειτουργικό σύστημα, εντολές αφής, οι οποίες αντιστοιχούν σε πραγματική αλληλεπίδραση με την οθόνη του κινητού τηλεφώνου. Αυτές οι δράσεις είναι κινήσεις όπως το σύρσιμο, πάτημα (στιγμιαίο ή διάρκειας), τσίμπημα (κανονικό και αντίστροφο) προκειμένου να μετακινηθούν και να τοποθετηθούν επί της οθόνης διάφορα αντικείμενα. Κάθε τέτοια κίνηση εντοπίζεται άμεσα από το σύστημα δίνοντας στον χρήστη την εντύπωση ενός πλήρως διαδραστικού ρευστού περιβάλλοντος με πολλές παράλληλες δυνατότητες. Το hardware της συσκευής, διαθέτει επίσης σε ορισμένες περιπτώσεις επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες εγγύτητας, τα οποία χρησιμοποιούνται από ορισμένες εφαρμογές προκειμένου να ανταποκριθούν στις πρόσθετες ενέργειες του χρήστη, όπως είναι για παράδειγμα η ρύθμιση της οθόνης κατά τον κατακόρυφο και τον οριζόντιο προσανατολισμό, ανάλογα με τον τρόπο που ο χρήστης κρατάει τη συσκευή, ή προσομοιώνοντας τον έλεγχο του τιμονιού σε άλλες εφαρμογές (όπως παιχνίδια κτλ.) Η αρχική οθόνη των Android, έχει την ίδια λογική με την επιφάνεια εργασίας των υπολογιστών. Πρόκειται για ένα κόμβο ο οποίος περιλαμβάνει την πλοήγηση στο περιβάλλον του Android. Στο homescreen, όπως ονομάζεται, βρίσκονται εικονίδια που με το πάτημά τους εκκινούν τις εφαρμογές που επιθυμεί ο χρήστης να του είναι εύκολα προσβάσιμες. Επίσης, υπάρχουν widgets, τα οποία αποτελούν γραφικά βοηθήματα για γρήγορη ενημέρωση σε θέματα όπως η ώρα, ο καιρός και άλλα. Η αρχική οθόνη μπορεί να αποτελείται από περισσότερες της μίας «σελίδες», οι οποίες είναι δυνατό να εμφανιστούν στον χρήστη με ένα απλό

«σύρσιμο» στην οθόνη αφής είτε προς τα μπροστά είτε προς τα πίσω. Ασφαλώς αξίζει να σημειωθεί ότι οι «σελίδες» οι οποίες είναι δυνατό να εμφανιστούν στην αρχική οθόνη του συστήματος, προσαρμόζονται στις προτιμήσεις του χρήστη προκειμένου να ικανοποιηθούν με το βέλτιστο τρόπο οι ανάγκες του από το σύστημα. Επίσης, στο πάνω μέρος της οθόνης, εμφανίζεται μια γραμμή κατάστασης, η οποία περιέχει πληροφορίες σχετικά με τη συσκευή και τη συνδεσιμότητα της. Αυτή η γραμμή κατάστασης μπορεί να "τραβηχτεί" προς τα κάτω για να αποκαλύψει ενημερώσεις (notifications) που αφορούν τον χρήστη από τις εφαρμογές που χρησιμοποιεί, καθώς και έναν μικρό πίνακα ελέγχου για αλλαγές στην ένταση του ήχου, στη φωτεινότητα της οθόνης και σε άλλα στοιχεία της συσκευής. Είναι προφανές, ότι οι εφαρμογές (applications) αποτελούν ένα σημαντικό τμήμα των Android συσκευών. Οι εφαρμογές αυτές, επεκτείνουν τη λειτουργικότητα των συσκευών και έχουν γραφτεί κατά κύριο λόγο σε γλώσσα προγραμματισμού Java, χρησιμοποιώντας το kit ανάπτυξης λογισμικού Android (SDK). Το SDK αποτελείται από μια πλήρη σειρά εργαλείων ανάπτυξης, εντός των οποίων περιλαμβάνεται ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες λογισμικού, μια συσκευή η οποία εξομοιώνει το περιβάλλον του Android και που βασίζεται στο QEMU, τεκμηρίωση, δείγματα κώδικα και βοηθητικό υλικό, που περιγράφει τον τρόπο λειτουργίας του αναφερόμενου λογισμικού ανάπτυξης. Αρχικά, η Google υποστήριξε το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Eclipse με τη χρήση των Android Development Tools (ADT) plugin ενώ στη συνέχεια, τον Δεκέμβριο 2014 η Google έφερε στη κυκλοφορία το Android Studio, το οποίο είναι βασισμένο στο IntelliJ IDEA, ως κύριο IDE για την ανάπτυξη εφαρμογών Android. Το Android έχει μια μεγάλη ποικιλία εφαρμογών, οι οποίες μπορούν να αποκτηθούν από τους χρήστες, με ένα απλό κατέβασμα και εγκατάσταση του αρχείου .APK της εφαρμογής στη συσκευή, είτε με τη λήψη τους χρησιμοποιώντας ένα από τα υπάρχοντα online καταστήματα του διαδικτύου. Το Google Play Store είναι το πρωταρχικό κατάστημα εφαρμογών, το οποίο είναι δυνατό να εγκατασταθεί σε συσκευές Android οι οποίες συμμορφώνονται με τις απαιτήσεις συμβατότητας της Google και άδεια χρήσης του λογισμικού της Google Mobile Services. Το Google Play Store επιτρέπει στους χρήστες να αναζητήσουν, να κατεβάσουν και να ενημερώσουν τις εφαρμογές που δημοσιεύθηκαν από την Google ή ακόμα και από άλλους προγραμματιστές.

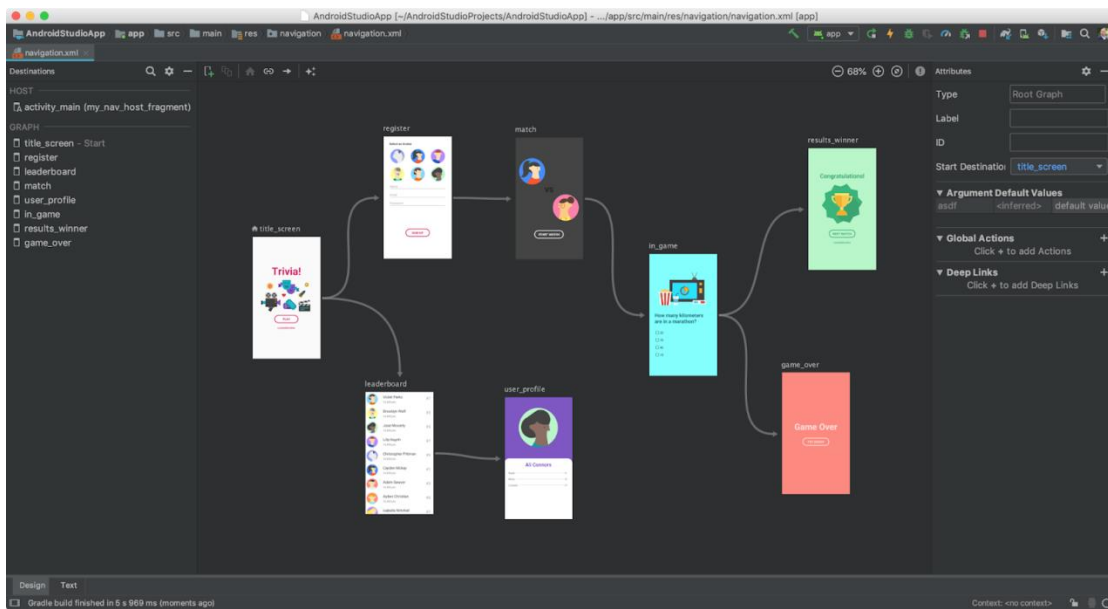
Το Android Studio [9] είναι το επίσημο περιβάλλον ανάπτυξης για εφαρμογές Android. Πρωτοεμφανίστηκε 16 Μαΐου 2003 σε συνέδριο της Google και είναι βασισμένο στο λογισμικό JetBrains-Intelli J IDEA ενώ είναι κατασκευασμένο σε Java. Επίσης εμπεριέχει το Android SDK το οποίο περιλαμβάνει τις βιβλιοθήκες που χρειάζονται για την ανάπτυξη των εφαρμογών.

Το Android Studio προσφέρει διάφορες υπηρεσίες στους χρήστες, κάποιες από αυτές είναι οι εξής:

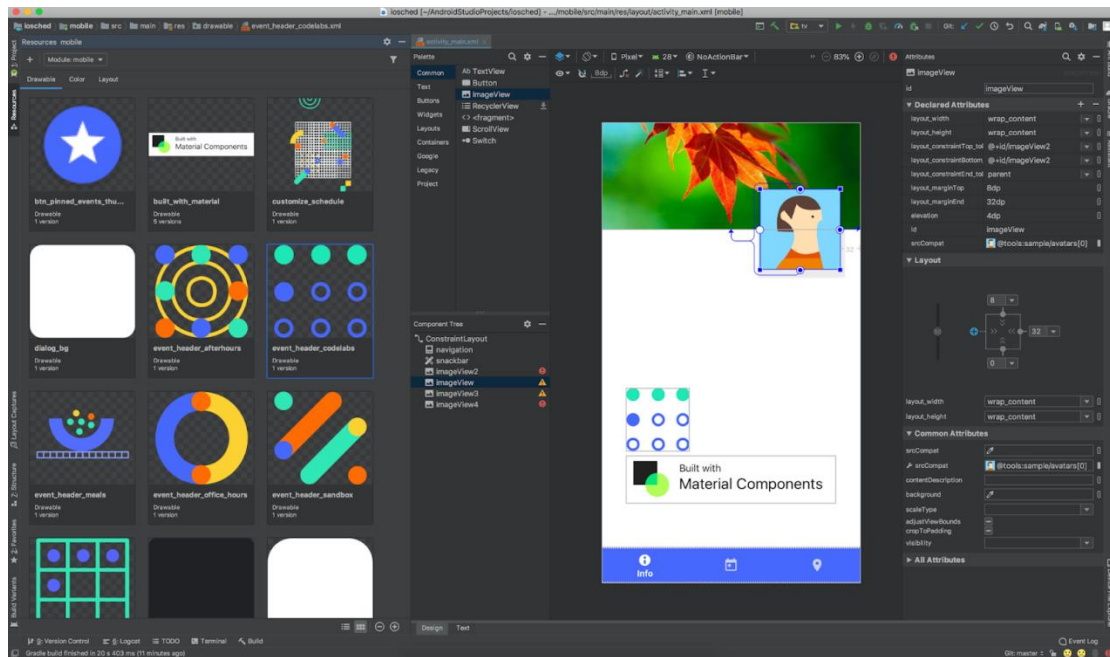
- Ένα ευέλικτο Cradle-based σύστημα κατασκευής.
- Ένα γρήγορο και με πολλές λειτουργίες εξομοιωτή (emulator).
- Δυνατότητα ανάπτυξης εφαρμογών για όλες τις συσκευές Android (Smartphones, Smart watches , Smart TV's).
- Όταν γίνονται αλλαγές στον κώδικα το Android Studio επιτρέπει την άμεση εκτέλεση του χωρίς να δημιουργεί καινούριο εκτελέσιμο APK (Τα αρχεία .apk είναι τα εκτελέσιμα αρχεία εγκατάστασης εφαρμογών Android).
- Περιέχει πρότυπα block κώδικα και επιτρέπει την σύνδεση με το Git Hub προκειμένου να χρησιμοποιηθεί κώδικας που βρίσκεται ανεβασμένος on-line.
- Περιέχει εργαλεία ελέγχου της απόδοσης , της χρηστικότητας και και έλεγχο συμβατότητας των εφαρμογών.
- Υποστήριξη C++
- Υποστήριξη Google Cloud Platform.



Εικόνα 2: Η πλατφόρμα Android Studio (πηγή: Android Developers)



Εικόνα 3: Η πλατφόρμα Android Studio και η διασύνδεσή της με κινητές συσκευές (πηγή: Android Developers)



Εικόνα 4: Το Android Studio και η εμφάνιση της αρχικής οθόνης (πηγή: androidstudio.googleblog.com)

2.2 Ανάπτυξη iOS εφαρμογών με Swift και Objective C

Το λειτουργικό σύστημα iOS είναι ένα λογισμικό, το οποίο έχει δημιουργηθεί από την εταιρία Apple και ενώ ξεκίνησε να χρησιμοποιείται στα κινητά της τηλέφωνα πλέον χρησιμοποιείται από την πλειοψηφία των συσκευών, τις οποίες παράγει η εταιρία, με εξαίρεση τους ηλεκτρονικούς υπολογιστές. Στην κατά πολλούς θεωρούμενη ως την καλύτερη του παρουσίαση, ο διευθύνων σύμβουλος της apple Steve Jobs στις 9/1/2007 παρουσίασε το iPhone, μια συσκευή που άλλαξε ριζικά τον τρόπο με τον οποίο ο κόσμος βλέπει τα smartphones 13 χρόνια μετά. Πλέον το iOS συγκαταλέγεται στα παλαιότερα λειτουργικά συστήματα έξυπνων συσκευών, τα οποία βρίσκονται στη γραμμή παραγωγής. Παρόλα αυτά δεν είναι παρωχημένο και ούτε με χαμηλές επιδόσεις. Αντιθέτως η Apple με συνεχή εξέλιξη έχει φτιάξει ένα πολύ γρήγορο, εύχρηστο και αξιόπιστο λειτουργικό σύστημα. Το iOS προσφέρει μία ευχάριστη και εύκολη εμπειρία στους απλούς χρήστες ως προς την χρήση και ταυτόχρονα μια ισχυρή πλατφόρμα για τους προγραμματιστές. Στην αρχική παρουσίαση του iPhone η apple ανακοίνωσε ότι το iPhone θα “έτρεχε” τον ίδιο πυρήνα λογισμικού UNIX, τον οποίο έτρεχαν και οι τότε υπολογιστές της εταιρίας. Ωστόσο κάτι τέτοιο δε συνέβη ποτέ, καθώς ήταν εξαρχής προφανές ότι με το

λειτουργικό σύστημα των υπολογιστών θα μοιραζόταν μόνο ορισμένα χαρακτηριστικά [10].

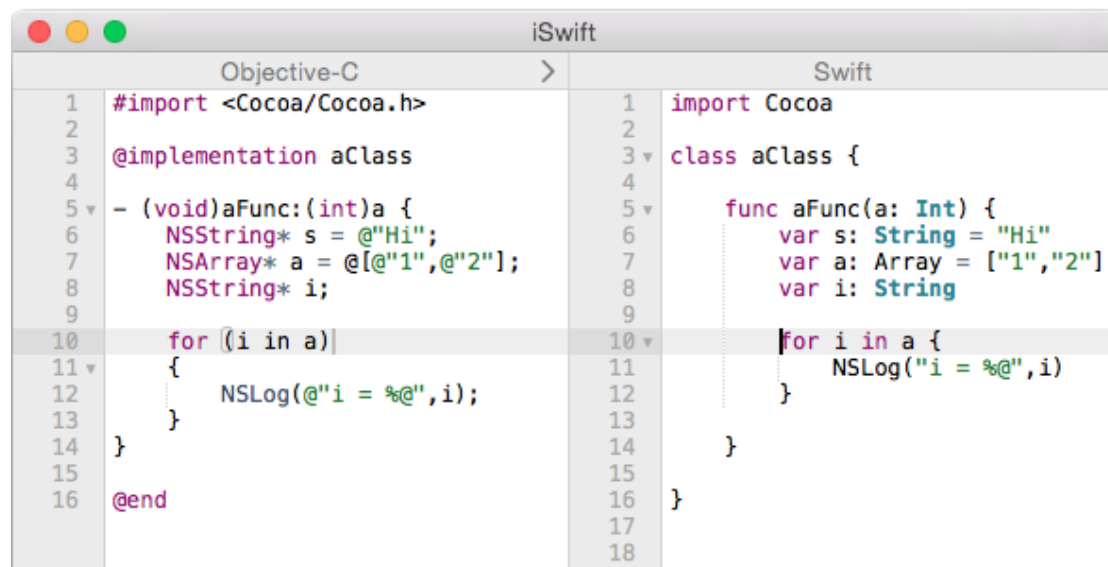
Η πρώτη έκδοση του iOS γνωστή και ως iPhone OS, ενώ έδινε πολύ ευχάριστη, ομαλή, (smooth) και γρήγορη εμπειρία στο χρήστη και ενώ ήταν αρκετά πίσω από τον ανταγωνισμό - εφόσον δεν υποστήριζε βασικές λειτουργίες, όπως 3G, MMS - δεν έδινε πρόσβαση στον χρήστη στο σύστημα αρχείων της συσκευής, καθώς επίσης και την δυνατότητα να εγκαταστήσει εφαρμογές από τρίτους κατασκευαστές και όχι μόνο αυτές, που είχε δημιουργήσει η ίδια η Apple. Η τελευταία δυνατότητα - η οποία μας ενδιαφέρει κατά κύριο λόγο, εφόσον αποτελεί σκοπό αυτής τη διπλωματικής εργασίας - δόθηκε στους χρήστες στην επόμενη έκδοση του λειτουργικού συστήματος, μιας και αυτοί μπορούσαν να τις κατεβάσουν από το App Store. Σε αντίθεση με τον ανταγωνισμό, η εγκατάσταση εφαρμογών έγινε άμεση, γρήγορη και ασφαλής, αφού γινόταν αποκλειστικά από το App Store και όχι από το διαδίκτυο χωρίς κανένα έλεγχο, όπως γινόταν μέχρι στιγμής. Η άλλη πρωτοτυπία της Apple ήταν το iOS SDK (Software Development Kit), με το οποίο έδινε στους προγραμματιστές την ευκαιρία να εκμεταλλευτούν της δυνατότητες του iPhone στο έπακρο και σαν αποτέλεσμα να δημιουργηθούν εκπληκτικές εφαρμογές. Έξι χρόνια περαιτέρω αναβαθμίσεων έχουν οδηγήσει στο iOS, ένα λειτουργικό, που παρέχει στο χρήστη όλες τις δυνατότητες για μια εκπληκτική εμπειρία.

Η γλώσσα Objective-C είναι μία αντικειμενοστραφής γλώσσα γενικού σκοπού και αποτελεί μαζί με τη γλώσσα C++ την αντικειμενοστραφή εξέλιξη της γλώσσας C. Είναι η γλώσσα που χρησιμοποιείται ως επί το πλείστον από την Apple για τα λειτουργικά συστήματα iOS και OSX, καθώς και από τα αντίστοιχα APIs (Application Programming Interface), το cocoa touch και το cocoa. Πέραν του προγραμματισμού στα συστήματα της Apple, η γλώσσα Objective-C είναι γενικότερα δημοφιλής στους προγραμματιστικούς κύκλους, καθώς σαν επέκταση της C συνδέεται άμεσα με τα περιβάλλοντα UNIX και Linux, τα οποία είναι επίσης γραμμένα σε C. Παρ' όλα αυτά δεν τυγχάνει της αποδοχής της C++, καθώς η τελευταία είναι πιο γρήγορη και πιο ολοκληρωμένη από άποψη υποστηριζόμενων βιβλιοθηκών. Το 1981 οι Brad Cox και Tom Love στην ανάγκη να λύσουν το πρόβλημα της επαναχρησιμοποίησης κώδικα στο software design και στον προγραμματισμό, δημιούργησαν την Objective-C σαν εξέλιξη της C, εφόσον ήταν βαρύνουσας σημασίας η συμβατότητα μαζί της. Ο Cox ξεκίνησε να φτιάχνει έναν

προεπεξεργαστή, για να προσθέσει στη C μερικές από τις λειτουργίες της γλώσσας Smalltalk, και σύντομα είχε μία λειτουργική επέκταση για τη C, που της προσέθετε τις δυνατότητες του αντικειμενοστραφούς προγραμματισμού. Το 1982 για πρώτη φορά ο Love την χρησιμοποίησε για εμπορικούς σκοπούς. Το 1988 η εταιρία NeXT πήρε τα δικαιώματα χρήσης της Objective-C από τη StepStone (η εταιρία που είχε τα δικαιώματα της Objective-C) και επέκτεινε τον gcc compiler έτσι, ώστε να την υποστηρίζει. Έπειτα δημιούργησε τις βιβλιοθήκες AppKit και Foundation Kit, πάνω στις οποίες βασίστηκε το NeXTstep user interface (διεπαφή με τους χρήστες), καθώς και το NeXTstep Interface Builder (εργαλείο για την κατασκευή διεπαφών). Και ενώ οι σταθμοί εργασίας της NeXT απέτυχαν εμπορικά, τα προγραμματιστικά εργαλεία της χρησιμοποιούνταν ευρέως, με αποτέλεσμα η NeXT να σταματήσει την ενασχόλησή της με το hardware και να αφοσιωθεί στο software πουλώντας το NeXTstep (και το OpenStep) ως πλατφόρμες προγραμματισμού. Τέλος, για να συμφωνεί με τους κανόνες του GNU General Public License (project, το οποίο επιτρέπει σε όλους την ελεύθερη χρήση του λογισμικού), η Objective-C συμπεριελήφθη στην υλοποίηση του gcc compiler. Το 1996, αφού η Apple εξαγόρασε την NeXT, χρησιμοποίησε το Openstep για την δημιουργία του νέου της λειτουργικού συστήματος, του MAC OS X. Αυτό αποτελούνταν από την Objective-C, το προγραμματιστικό εργαλείο της NeXT “Project Builder”, που σήμερα είναι γνωστό και ως xCode, και από το εργαλείο δημιουργίας διεπαφών με το χρήστη “Interface Builder”. Ακόμα και σήμερα τα περισσότερα αντικείμενα του Cocoa API, τα οποία χρησιμοποιούνται για τον προγραμματισμό συσκευών της Apple, βασίζονται στα αντικείμενα του Openstep.

Η γλώσσα Swift είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία δημιουργήθηκε από την Apple για την ανάπτυξη εφαρμογών σε iOS και OSX. Υποστηρίζει τα framework Cocoa και Cocoa Touch της Apple, καθώς και όλες τις εφαρμογές για προϊόντα της Apple, που έχουν γραφεί σε Objective-C. Επιπλέον επιτρέπει στο ίδιο πρόγραμμα να τρέχει Objective-C, C, C++ και Swift. Η Swift είναι επί της ουσίας μία ανακατασκευή της Objective-C, χρησιμοποιώντας σύγχρονες τεχνικές και σύνταξη. Το 2014 παρουσιάστηκε στο παγκόσμιο συνέδριο προγραμματιστών της Apple (Worldwide Developers Conference). Η εξέλιξή της βέβαια έχει αρχίσει από το 2010 από πλήθος προγραμματιστών με επικεφαλής τον

Chris Lattner, οι οποίοι δημιούργησαν μια γλώσσα με στοιχεία από τις δημοφιλέστερες γλώσσες προγραμματισμού.



```
Objective-C > Swift
1 #import <Cocoa/Cocoa.h> 1 import Cocoa
2 2
3 @implementation aClass 3 class aClass {
4 4
5 - (void)aFunc:(int)a { 5 func aFunc(a: Int) {
6     NSString* s = @"Hi"; 6     var s: String = "Hi"
7     NSArray* a = @[@"1",@"2"]; 7     var a: Array = ["1","2"]
8     NSString* i; 8     var i: String
9 9
10 for (i in a) 10 for i in a {
11 { 11     NSLog("i = %@", i)
12     NSLog(@"i = %@", i); 12 }
13 } 13
14 } 14 }
15 15
16 @end 16 }
17 17
18 18
```

Εικόνα 5: Το περιβάλλον της Swift-Objective C (πηγή: iswift.org)

2.3 Δομικά Στοιχεία μιας εφαρμογής

Μια εφαρμογή αποτελείται από συνδεδεμένα διακριτά στοιχεία τα όποια περιγράφονται από ένα δηλωτικό αρχείο που περιγράφει την δομή τους και την αλληλεπίδραση τους. Τα επόμενα δομικά στοιχεία είναι οι βασικοί πυλώνες των εφαρμογών [11]:

- Activity Το επίπεδο παρουσίασης της εφαρμογής. Κάθε οθόνη της εφαρμογής είναι μία επέκταση της κλάσης Activity. Κάθε Activity χρησιμοποιεί στιγμότυπα της κλάσης View για να παρουσιάσει δεδομένα στην οθόνη σχηματίζοντας διάφορες γραφικές διεπαφές με το χρήστη καθώς και για να αλληλοεπιδράσει με τις ενέργειες του.
- Services Οι αθέατοι εργάτες της εφαρμογής. Τα Services τρέχουν στο παρασκήνιο ανανεώνοντας τα δεδομένα και τις Activities και ενεργοποιώντας διάφορες ειδοποιήσεις. Χρησιμοποιούνται για να κάνουν κάποια εργασία όταν η εφαρμογή δεν είναι ορατή ή ενεργοποιημένη.
- Content Providers Κοινόχρηστα αποθηκευμένα δεδομένα. Χρησιμοποιούνται για την διαχείριση και τον διαμοιρασμό των βάσεων δεδομένων της εφαρμογής. Μπορείς

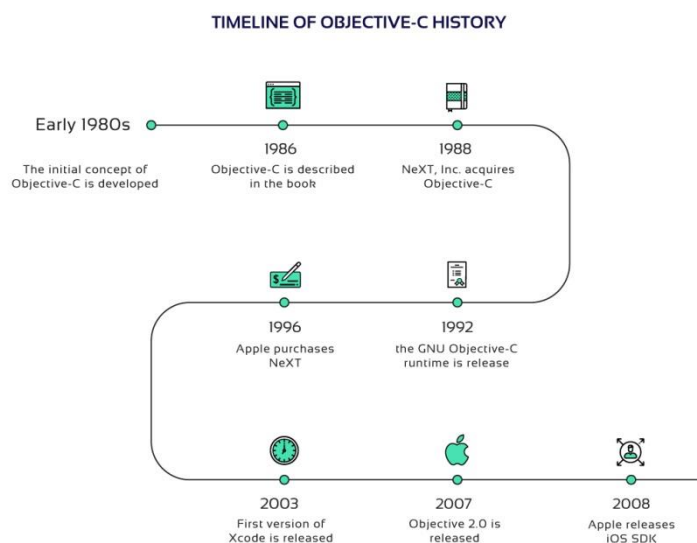
μέσω ενός Provider να αφήσεις ανοιχτή την πρόσβαση στα δεδομένα σου από μία άλλη εφαρμογή ή να έχεις πρόσβαση στη βάση δεδομένων μίας άλλης εφαρμογής παίρνοντας δεδομένα από αυτή.

→ Widgets Οπτικά στοιχεία της εφαρμογής που μπορούν να προστεθούν στη βασική οθόνη. Χρησιμοποιώντας τα μπορούμε να φτιάξουμε δυναμικά ,δια δραστικά στοιχεία της εφαρμογής για την βασική οθόνη του χρήστη.

→ Notifications Πλαίσιο ειδοποιήσεις του χρήστη. Μέσω των ειδοποιήσεων μπορεί να ενημερώσεις το χρήστη χωρίς να χάνει την προσοχή από την τρέχουσα εφαρμογή και χωρίς να την διακόπτεις.

→ Intents ένα πλαίσιο που αποστέλλει ένα μήνυμα στο επίπεδο της δια - εφαρμογής. Χρησιμοποιώντας μια Intent μπορείς να εκπέμψεις μηνύματα σε όλο το σύστημα ή σε μία συγκεκριμένη Activity προκαλώντας έτσι κάποια άλλη ενέργεια.

→ Broadcast Receivers είναι οι καταναλωτές των Intents. Δημιουργώντας και χρησιμοποιώντας ένα Broadcast receiver η εφαρμογή μπορεί να περιμένει για κάποιο Intent που πληροί τα συμφωνηθέντα κριτήρια και να προβεί στις απαραίτητες ενέργειες. Ειδικό για το σχεδιασμό εφαρμογών οδηγούμενων από γεγονότα(event-driven apps).



Εικόνα 6: Η ιστορική πορεία της Objective C (πηγή: lightIT)

ΚΕΦΑΛΑΙΟ 3: Web Mobile Apps

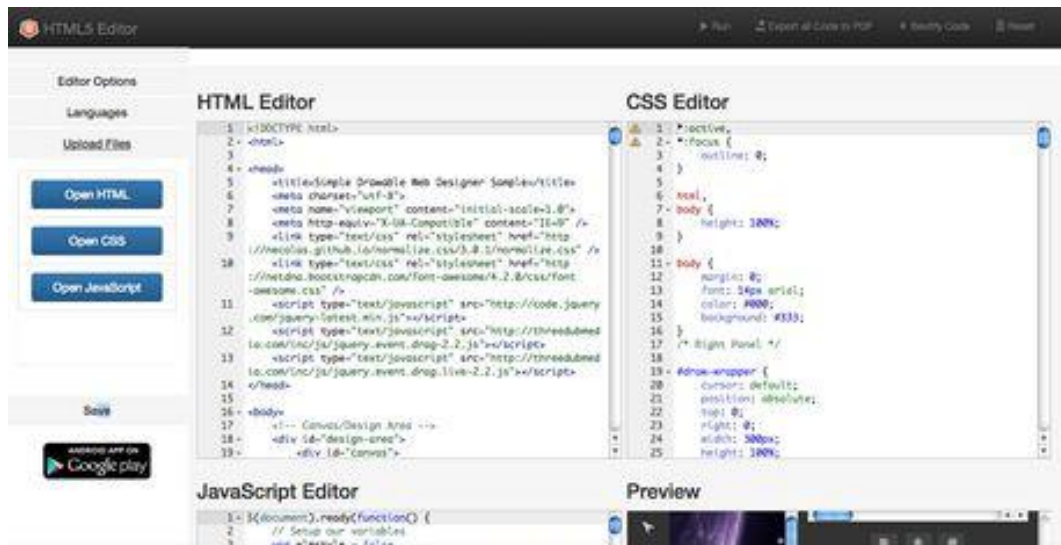
Σύμφωνα με τον οργανισμό διεθνούς Κοινοπραξίας του Παγκόσμιου Ιστού W3C(World Wide Consortium), τα web standards είναι τα πρότυπα και οι ολοκληρωμένες τεχνικές προδιαγραφές που καθορίζουν και περιγράφουν όλες τις πτυχές του World Wide Web. Τα τελευταία χρόνια, ο όρος αυτός φέρεται να υιοθετεί πλέον ένα σύνολο βέλτιστων τυποποιημένων πρακτικών για την κατασκευή ιστοσελίδων, καθώς και τη φιλοσοφία του web design και development όπου περιλαμβάνονται τέτοιου είδους μέθοδοι. Επίσης, τα πρότυπα αυτά περιλαμβάνουν πολλές αλληλοεξαρτώμενες προδιαγραφές, πάνω στις οποίες καθοδηγείται ολόκληρο το διαδίκτυο, έχοντας στο στόχαστρο την διαλειτουργικότητα, την ευχρηστία και την προσβασιμότητα. Αυτοί οι τρεις παράγοντες είναι και το κλειδί για την αρμονική συνεργασία και επικοινωνία διαφορετικών διεπαφών και συσκευών με σκοπό την διαχείριση των πληροφοριών κάτω από τις ίδιες προδιαγραφές.

3.1 HTML5

Η HTML5 [12] είναι η πιο πρόσφατη έκδοση στην σειρά των εκδόσεων στην ιστορία της και είναι πλέον η γλώσσα αιχμής στο World Wide Web. Πήρε τον τίτλο της W3C Recommendation στις 28-10-2014. Όπως είναι αναμενόμενο, έχει αναπτυχθεί από την κοινοπραξία αλλά και από το επονομαζόμενο Web Hypertext Application Technology Working Group(WHATWG). Η νέα αυτή έκδοση διαφέρει σε σχέση με τις παλαιότερες εκδόσεις σε πολλούς σημαντικούς τομείς. Είναι χρήσιμο να αναφερθούν κάποια βασικά πράγματα σχετικά με την προέλευση της HTML,έτσι ώστε να γίνει κατανοητή η HTML5. Η HTML ξεκίνησε στις αρχές της δεκαετίας του 1990 ως ένα σύντομο έγγραφο που παρέθετε κάποια στοιχεία τα οποία χρησιμοποιούνταν για την κατασκευή ιστοσελίδων. Πολλά από τα στοιχεία αυτά αφορούσαν την περιγραφή περιεχομένου ιστοσελίδων όπως οι επικεφαλίδες, οι παράγραφοι και οι λίστες. Ο αριθμός έκδοσης της HTML μεγάλωνε καθώς η γλώσσα εξελισσόταν με την εισαγωγή νέων στοιχείων και προσαρμογών στους κανόνες της. Η τελευταία έκδοση είναι η HTML5. Η HTML5 αποτελεί φυσική εξέλιξη των προηγούμενων εκδόσεων της HTML και προσπαθεί να καλύψει τις ανάγκες των υπάρχοντων και των μελλοντικών ιστότοπων. Κληρονομεί τη συντριπτική πλειονότητα των λειτουργιών από τους προκατόχους της, γεγονός που σημαίνει ότι αν κάποιος έχει γράψει κώδικα HTML πριν από την έλευση της HTML5, γνωρίζει

ήδη αρκετά για την HTML5. Αυτό σημαίνει επίσης ότι μεγάλο μέρος της HTML5 λειτουργεί τόσο σε παλιούς όσο και σε νέους φυλλομετρητές. Η προς τα πίσω συμβατότητα αποτελεί βασική αρχή στον σχεδιασμό της HTML5. Η HTML (αρχικοποίηση του αγγλικού HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<div>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, ενώ μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML. Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

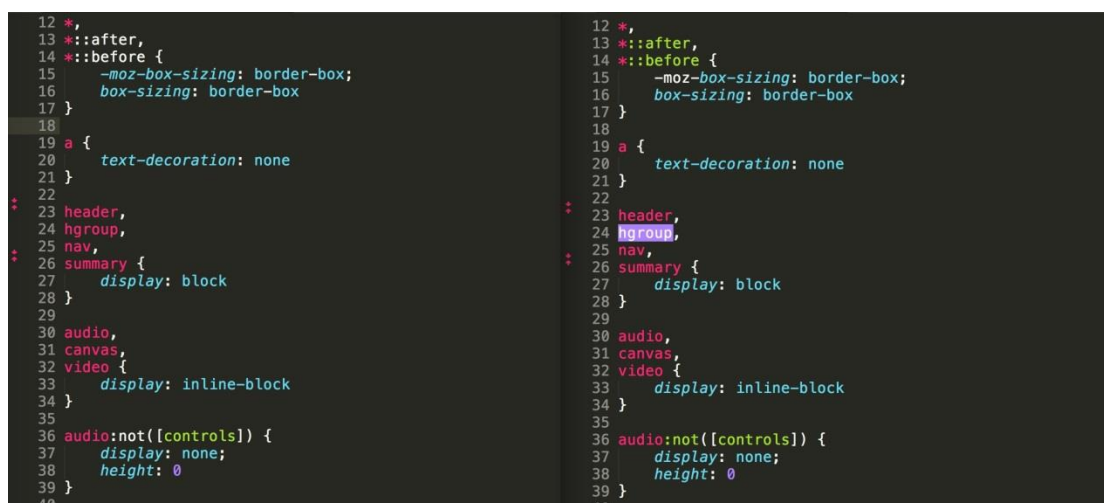


Εικόνα 7: Το περιβάλλον της HTML5

3.2 CSS3

Ενώ η HTML καθορίζει το νόημα του περιεχομένου και προσδίδει στις ιστοσελίδες τη βασική τους δομή, τα επάλληλα φύλλα στυλ (Cascading Style Sheets, CSS) καθορίζουν την εμφάνιση. Το φύλλο στυλ (style sheet) είναι απλώς ένα αρχείο κειμένου το οποίο περιλαμβάνει έναν ή περισσότερους κανόνες που προσδιορίζουν – μέσω ιδιοτήτων (properties) και τιμών (values) – με ποιον τρόπο θα πρέπει να απεικονιστούν τα συγκεκριμένα στοιχεία των ιστοσελίδων. Υπάρχουν ιδιότητες CSS για τον έλεγχο της βασικής μορφοποίησης (όπως είναι το μέγεθος της γραμματοσειράς και το χρώμα), των ιδιοτήτων της διάταξης της σελίδας (όπως είναι η θέση και η ροή του κειμένου) και του χειρισμού της εκτύπωσης (όπως είναι ο καθορισμός του σημείου αλλαγής σελίδας όταν την τυπώνει ο χρήστης). Η τεχνολογία CSS διαθέτει επίσης πολλές δυναμικές ιδιότητες που επιτρέπουν σε αντικείμενα να εμφανίζονται και να εξαφανίζονται, και οι οποίες είναι χρήσιμες για τη δημιουργία πτυσσόμενων μενού και άλλων αλληλεπιδραστικών στοιχείων. Αυτός ο διαχωρισμός της μορφοποίησης και περιεχομένου δίνει τη δυνατότητα να παρουσιαστεί η ίδια σελίδα σήμανσης σε διαφορετικές μορφές για την απόδοση διαφορετικών μεθόδων, όπως στην οθόνη, σε έντυπη μορφή, από τη φωνή (όταν διαβάζεται από μια μηχανή αναζήτησης που βασίζεται σε ομιλία ή αναγνώστη οθόνης) και στις συσκευές που βασίζονται σε Braille. Μπορεί επίσης να χρησιμοποιηθεί για να εμφανίσει την ιστοσελίδα διαφορετικά, ανάλογα με το μέγεθος της οθόνης ή η συσκευή στην οποία προβάλλεται το αντικείμενο. Αν και ο συντάκτης

μιας ιστοσελίδας, συνήθως, συνδέει ένα CSS αρχείο μέσα στο αρχείο σήμανσης, οι αναγνώστες για να καθορίσουν ένα διαφορετικό φύλλο στυλ, όπως ένα CSS αρχείο που είναι αποθηκευμένο στο δικό τους υπολογιστή, πρέπει να παρακάμψουν αυτό που έχει καθορίσει ο συγγραφέας. Αν ο συγγραφέας ή ο αναγνώστης δεν συνδέσουν το έγγραφο σε ένα φύλλο στυλ, θα εφαρμόζεται το προεπιλεγμένο στυλ του προγράμματος περιήγησης. Ένα άλλο πλεονέκτημα των CSS είναι ότι οι αισθητικές αλλαγές στο γραφικό σχεδιασμό ενός εγγράφου (ή εκατοντάδες έγγραφα) μπορούν να εφαρμοστούν γρήγορα και εύκολα, τροποποιώντας μερικές γραμμές σε ένα αρχείο, και όχι από μια διαδικασία επίπονη (και κατά συνέπεια ακριβή) και όχι πάνω από το έγγραφο γραμμή προς γραμμή για να αλλάξουν την σήμανση [13].



```
12 *,
13 ::after,
14 ::before {
15   -moz-box-sizing: border-box;
16   box-sizing: border-box
17 }
18
19 a {
20   text-decoration: none
21 }
22
23 header,
24 hgroup,
25 nav,
26 summary {
27   display: block
28 }
29
30 audio,
31 canvas,
32 video {
33   display: inline-block
34 }
35
36 audio:not([controls]) {
37   display: none;
38   height: 0
39 }
40
```

Εικόνα 8: Το περιβάλλον του CSS3 (πηγή: packagecontrol.io)

3.3 JavaScript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototypebased), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι

βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές και συναρτησιακό στυλ προγραμματισμού. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού) [14].

```
tsconfig.json
1
2 "compilerOptions": {
3   "target": "es5",
4   "module": "commonjs",
5   "noImplicitAny": true,
6   "removeComments": true,
7   "preserveConstEnums": true,
8   "sourceMap": true
9 },
10 "files": [
11   "./core/js/class.typescript.ts"
12 ]
13

Ts class.typescript.ts
1 class Typescript {
2   foo = 2;
3   bar = 'string';
4
5   constructor(foo: number, bar: string) {
6     this.foo = foo;
7     this.bar = bar;
8   }
9
10  method() {
11    alert('foo: ' + this.foo
12          + ' - bar: ' + this.bar);
13  }
14
15  let typo = new Typescript(5, 'test');
16  typo.method();
17
18
19

JS class.typescript.js
1 var Typescript = (function () {
2   function Typescript(foo, bar) {
3     this.foo = 2;
4     this.bar = 'string';
5     this.foo = foo;
6     this.bar = bar;
7   }
8   Typescript.prototype.method = function () {
9     alert('foo: ' + this.foo
10           + ' - bar: ' + this.bar);
11   };
12   return Typescript;
13 }());
14 var typo = new Typescript(5, 'test');
15 typo.method();
16 /** sourceMappingURL=class.typescript.js.map

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Samets-MBP:tnado-seo-cms worldwantweb$ tsc
Samets-MBP:tnado-seo-cms worldwantweb$ tsc
Samets-MBP:tnado-seo-cms worldwantweb$ tsc
Samets-MBP:tnado-seo-cms worldwantweb$ []
```

Εικόνα 9: Το περιβάλλον του Javascript (πηγή:tnado.com)

3.4 Responsive Web Design Techniques

Όλη αυτή η εξέλιξη της τεχνολογίας των τελευταίων χρόνων αλλά και η τάση των χρηστών να χρησιμοποιούν ολοένα και περισσότερο τις κινητές τους συσκευές δημιούργησε την ανάγκη για την εφαρμογή της τεχνικής που ονομάζεται Responsive Web Design (RWD) για τον σχεδιασμό των ιστοσελίδων. Σύμφωνα με αυτή την τακτική μια ιστοσελίδα σχεδιάζεται με τέτοιο τρόπο έτσι ώστε το περιεχόμενό της, να διαμορφώνεται και να προσαρμόζεται ανάλογα με την συσκευή και το μέγεθος της οθόνης που καλείται να εμφανιστεί η σελίδα.

Την έννοια responsive site τη γνωρίσαμε πρώτη φορά από ένα άρθρο του Ethan Marcotte στο alistapart.com το 2010. Οι Web developers και Web designers μέχρι τότε ήταν συνηθισμένοι στο να δημιουργούν ιστοσελίδες και διάφορα designs (σχέδια) σύμφωνα με το εκάστοτε format (σχήμα, διάταξη) και τις διαστάσεις του μέσου που θα τα περιελάμβαναν [15].

Συμβαδίζοντας όμως με την εξέλιξη της τεχνολογίας παρατηρούμε πως υπάρχει πλέον στην αγορά μεγάλη ποικιλία από συσκευές οι οποίες συνδέονται στο διαδίκτυο, μικρές, μεγάλες, επιτραπέζιες, υπολογιστές, τηλέφωνα, tablets κλπ. Των οποίων οι διαστάσεις ποικίλουν. Έτσι ο Marcotte θέλησε να τους δείξει ότι πρέπει να δημιουργήσουν μία νέα εποχή στο web design, το Responsive web design και κατ'

επέκταση το Responsive Site. Το Responsive Site έχει στόχο την ευελιξία όσον αφορά την μετακίνηση του σχεδιασμού στο ευρύ φάσμα των συσκευών που υπάρχουν σήμερα. Είναι, δηλαδή ένα web design το οποίο θα προσφέρει στον χρήστη καλύτερες συνθήκες πλοήγησης.

Το Responsive Web Design (RWD) πήρε την ονομασία του από την λέξη "Respond" και σημαίνει ότι μια ιστοσελίδα προσαρμόζεται δυναμικά και «on the fly» (δηλαδή άμεσα) σε κάθε διάσταση και προσανατολισμό (orientation) οθόνης. Αυτό σημαίνει ότι όταν κάποιος πλοηγηθεί σε μια ιστοσελίδα η οποία είναι Responsive, χρησιμοποιώντας για παράδειγμα ένα smartphone, η ιστοσελίδα θα διαμορφωθεί με τέτοιο τρόπο ώστε να μην χρειάζεται ο χρήστης να κάνει μεγέθυνση ή να κάνει πλάγιο σκρολ για να διαβάσει με ευκολία το περιεχόμενό της. Το Flexibility είναι το βασικό χαρακτηριστικό ενός Responsive Web Design. Τα πάντα θα πρέπει να είναι flexible: layouts, image sizes, text block κτλ. Αυτό επιτυγχάνεται με ένα σύνολο από τεχνικές οι οποίες μπορούν να δώσουν στο σχεδιασμό ενός responsive site το flexibility που χρειάζεται.

Στις μέρες μας ο ορισμός του Responsive Web Design έχει διευρυνθεί αρκετά, αλλά εξακολουθεί να αναγνωρίζεται παραδοσιακά από τις παρακάτω βασικές τεχνικές:

1. Fluid Grid Layout

Με τον όρο fluid grids αναφερόμαστε στην ιδιότητα που σε μια ιστοσελίδα όταν αποκτά ελαστικότητα και προσαρμοστικότητα σε οθόνες διάφορων αναλύσεων και μεγεθών. Αυτό σημαίνει ότι τα μέρη που απαρτίζουν μια σελίδα «ρέουν» μέσα σε αυτή ώστε να προσαρμοστούν στο περιβάλλον του χρήστη. Στον σχεδιασμό ενός ιστότοπου χρησιμοποιούμε grids (οριζόντιες και κάθετες γραμμές) στο layout μας, ώστε η διάταξη ή η οργάνωση του περιεχομένου να προσαρμόζεται μέσα σε αυτό και όχι να έχουν κάποιο σταθερό πλάτος. Με την χρήση των grids δεν χρειάζεται να κάνουμε πολλές παρεμβάσεις στον σχεδιασμό αλλά και στον κώδικα, καθώς τα στοιχεία της σελίδας προσαρμόζονται μέσα στις διαστάσεις του parent container ανάλογα με την ανάλυση της οθόνης ή/ και το orientation της συσκευής (portrait ή landscape).

2. Flexible Images

Με τον ορισμό `flexible images`, εννοούμε οποιοδήποτε γραφικό στοιχείο ή `media` (`images`, `icons`, `video` κτλ) μπαίνει μέσα στη σελίδα και θέλουμε να προσαρμόζεται στην οθόνη του χρήστη αναλογικά. Μια τεχνική που προτιμάται στο `responsive web design`. Συνήθως αναφερόμαστε σε φωτογραφίες που μπορούν να τοποθετηθούν στα πλαίσια εντός του `grid` και μπορούν να προσαρμοστούν στο `maxwidth: 100%`. Αυτός ο τρόπος μειώνει το μέγεθος των φωτογραφιών και κατά συνέπεια το μέγεθος της σελίδας καθώς φορτώνουμε μια φορά την φωτογραφία και μετά την προσαρμόζουμε στα διάφορα μεγέθη.

3. Media queries

Είναι μια `CSS` τεχνική που ενσωματώθηκε στην `version CSS3` που επιτρέπει στο περιεχόμενο μιας σελίδας να προσαρμόζεται σε διαφορετικές αναλύσεις της οθόνης. Πρόκειται στην ουσία για έναν έλεγχο που γίνεται βάσει συγκεκριμένων κριτηρίων (όπως το πλάτος ή το ύψος της οθόνης της συσκευής μας) για πάρει τελικά το περιεχόμενο της σελίδας μας την μορφή την οποία θέλουμε. Με την `CSS3` ελέγχουμε ουσιαστικά πως θα τοποθετηθούν τα `styles` στον κώδικα με `media attributes`. Διαφορετικά `styles` μπορούν να εφαρμοστούν με διαφορετικές ιδιότητες χρησιμοποιώντας `breakpoints` ανάλογα με τις διαστάσεις της οθόνης (`width/ height`) αλλά και με το προσανατολισμό της οθόνης (`portrait/ landscape`) κτλ.

Τα παραπάνω είναι οι βασικές τεχνικές που χρησιμοποιούμε στο `Responsive Web Design`. Ωστόσο, υπάρχουν και άλλες τεχνικές που είναι συνυφασμένες με το `Responsive Design`. Ένα παράδειγμα είναι η `mobile-first` προσέγγιση όπου σε αυτή την περίπτωση κάνουμε τον σχεδιασμό με βάση το `mobile` και μετα πηγαίνουμε σε άλλες συσκευές όπως `tablet`, `laptop` και `υπολογιστές`.

3.5 Bootstrap

Το `Bootstrap` είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει `HTML` και `CSS` για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις `JavaScript`. Είναι το πιο δημοφιλές πρόγραμμα στο `GitHub` και έχει χρησιμοποιηθεί από τη `NASA` και το `MSNBC`, μεταξύ άλλων.

Το Bootstrap αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. Πριν το Bootstrap, διάφορες βιβλιοθήκες χρησιμοποιήθηκαν για την ανάπτυξη της διεπαφής, η οποία οδήγησε σε αντιφάσεις και υψηλή φορολογική επιβάρυνση συντήρησης. Σύμφωνα με τον Twitter developer Mark Otto, για την αντιμετώπιση αυτών των προκλήσεων : "... πήρα μαζί μία μικρή ομάδα προγραμματιστών για να σχεδιάσει και να κατασκευάσει ένα νέο εργαλείο εσωτερικής κατανάλωσης και τότε βρήκα μια ευκαιρία να κάνουμε κάτι περισσότερο. Μέσω αυτής της διαδικασίας, είδαμε τους εαυτούς μας να χτίζουν κάτι πολύ πιο σημαντικό από οποιοδήποτε άλλο εσωτερικό εργαλείο. Μήνες αργότερα, καταλήξαμε σε μια πρώιμη έκδοση του Bootstrap ως έναν τρόπο για να καταγράψουμε και να μοιραστούμε κοινά πρότυπα σχεδιασμού και περιουσιακών στοιχείων εντός της εταιρείας».

Η πρώτη εγκατάσταση υπό πραγματικές συνθήκες συνέβη κατά τη διάρκεια της πρώτης Hackweek του Twitter». Ο Mark Otto έδειξε σε κάποιους συναδέλφους πώς να επιταχύνουν την ανάπτυξη του έργου τους με τη βοήθεια της εργαλειοθήκης. Ως αποτέλεσμα, δεκάδες ομάδες έχουν μετακινηθεί στο πλαίσιο.

Τον Αύγουστο του 2011 κυκλοφόρησε Twitter Bootstrap ως λογισμικό ανοιχτού κώδικα. Τον Φεβρουάριο του 2012, ήταν το πιο δημοφιλές έργο ανάπτυξης στο GitHub.

Το Bootstrap έχει σχετικά ελλιπή υποστήριξη για HTML5 και CSS, αλλά είναι συμβατό με όλους τους φυλλομετρητές (browsers). Βασικές πληροφορίες συμβατότητας των ιστοσελίδων είναι διαθέσιμες για όλες τις συσκευές και τα προγράμματα περιήγησης. Υπάρχει μια έννοια της μερικής συμβατότητας που κάνει τα βασικά στοιχεία μιας ιστοσελίδας που διατίθενται για όλες τις συσκευές και τα προγράμματα περιήγησης. Για παράδειγμα, οι ιδιότητες πάνω στις οποίες θεσπίστηκε το CSS3 για στρογγυλεμένες γωνίες, κλίσεις και σκιές, χρησιμοποιούνται από το Bootstrap παρά την έλλειψη υποστήριξης από μεγάλα προγράμματα περιήγησης στο Web. Αυτά επεκτείνουν τη λειτουργικότητα του πακέτου εργαλείων, αλλά δεν απαιτούνται για τη χρήση του.

Από την έκδοση 2.0 υποστηρίζει επίσης responsive design. Αυτό σημαίνει ότι η διάταξη των ιστοσελίδων προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (PC, tablet, κινητό τηλέφωνο). Το

Bootstrap είναι ανοικτού κώδικα και είναι διαθέσιμο στο GitHub. Οι προγραμματιστές ενθαρρύνονται να συμμετέχουν στο έργο και να κάνουν τη δική τους συνεισφορά στην πλατφόρμα.

Το Bootstrap [16] είναι σπονδυλωτό και αποτελείται ουσιαστικά από μια σειρά stylesheets που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα στυλ που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους. Προσαρμογές είναι δυνατές σε περιορισμένη έκταση μέσω ενός κεντρικού στυλ διαμόρφωσης. Η χρήση γλώσσας στυλ επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς (operators), ένθετους επιλογείς, γνωστά και ως μείγματα mixin. Από την έκδοση 2.0, η διαμόρφωση του Bootstrap έχει επίσης μία ειδική επιλογή "Προσαρμογή " στο documentation. Επιπλέον, ο σχεδιαστής του έργου επιλέγει σε μια φόρμα τα επιθυμητά συστατικά και τα προσαρμόζει, εάν είναι αναγκαίο, σε τιμές διαφόρων εναλλακτικών λύσεων για τις ανάγκες του. Στη συνέχεια δημιουργείται ένα πακέτο που περιλαμβάνει ήδη το προχτισμένο CSS στυλ.

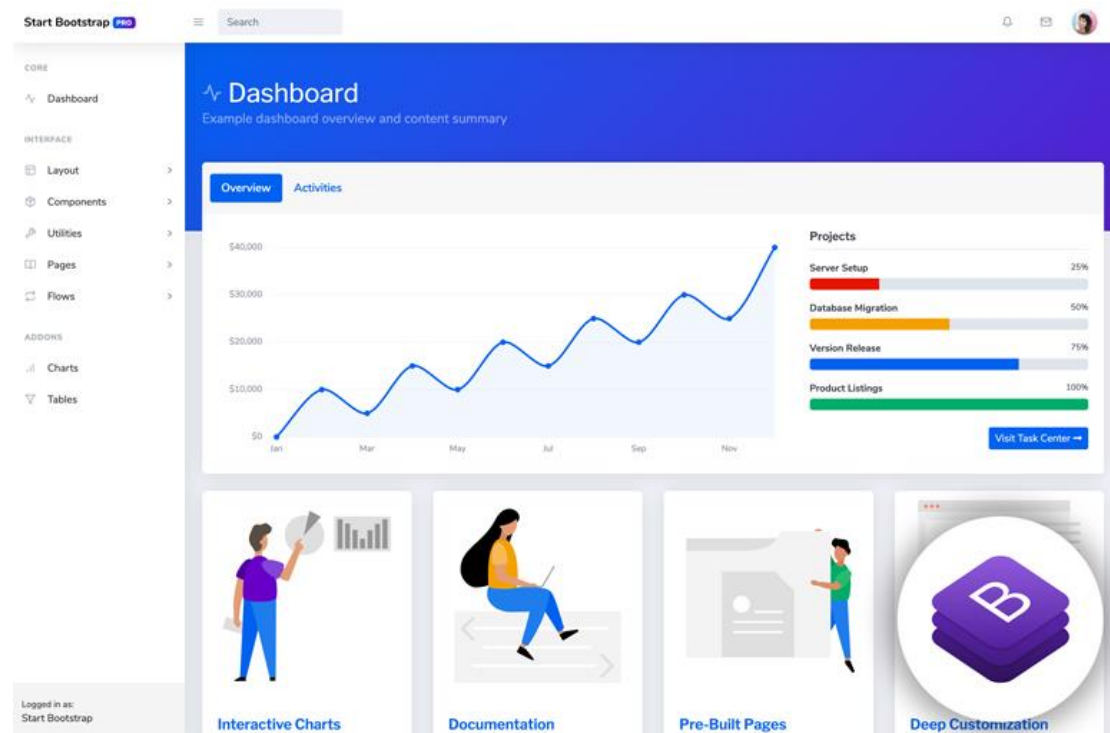
Το Bootstrap έρχεται σταθερό με 940 pixel πλάτος. Εναλλακτικά, ο προγραμματιστής μπορεί να χρησιμοποιήσει μία μεταβλητού πλάτους διάταξη. Και για τις δύο περιπτώσεις, η εργαλειοθήκη έχει τέσσερις παραλλαγές για χρήση σε διάφορους τύπους συσκευών. Κάθε παραλλαγή ρυθμίζει το πλάτος των στηλών.

Εκτός από τα βασικά HTML στοιχεία, το Bootstrap περιέχει και άλλα στοιχεία περιβάλλοντος που χρησιμοποιούνται συχνά. Αυτά περιλαμβάνουν κουμπιά με προηγμένα χαρακτηριστικά (π.χ. ομαδοποίηση κουμπιών ή drop -down επιλογή, οριζόντιες και κάθετες καρτέλες, πλοήγηση, σελιδοποίηση, κ.λπ.), ετικέτες, προηγμένες τυπογραφικές δυνατότητες, εικονίδια, προειδοποιητικά μηνύματα και γραμμές προόδου. Το Bootstrap έρχεται με πολλά συστατικά JavaScript σε μια μορφή jQuery plugin. Παρέχουν πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρousel. Μπορούν επίσης να επεκτείνουν τη λειτουργικότητα ορισμένων υφιστάμενων στοιχείων της διασύνδεσης, όπως για παράδειγμα μια αυτόματη πλήρη λειτουργία για πεδία εισαγωγής. Στην έκδοση 2.0, υποστηρίζονται τα ακόλουθα JavaScript plugins: Modal, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel και Typeahead. Διατίθεται επίσης μια

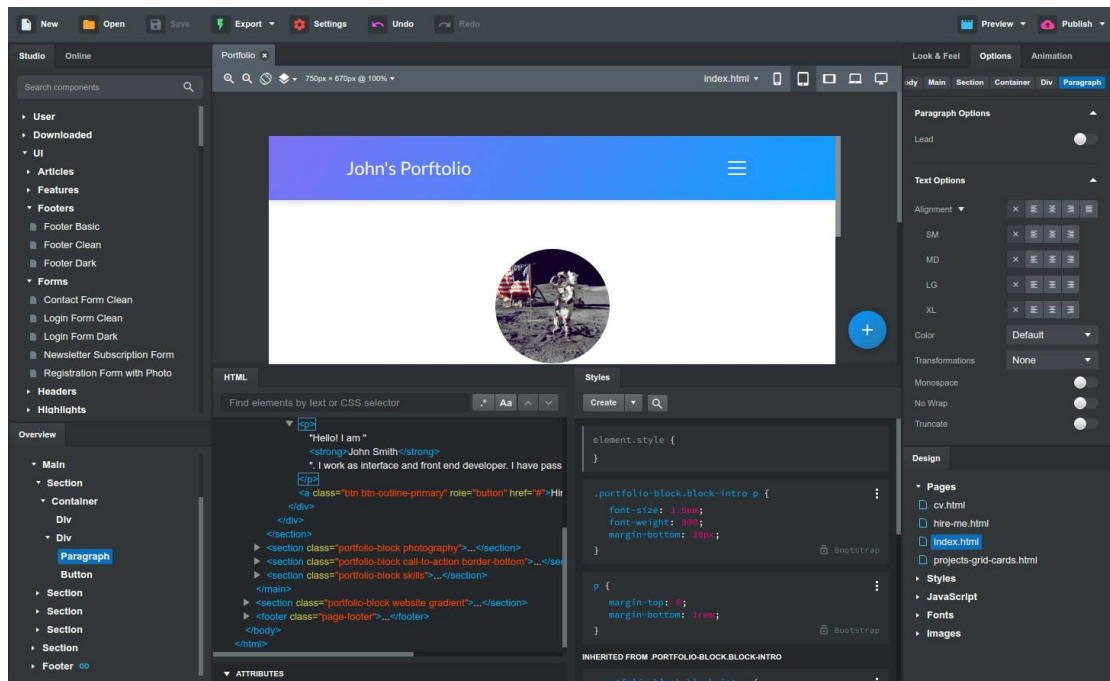
υλοποίηση του Bootstrap Twitter που χρησιμοποιεί το Dojo Toolkit. Λέγεται Dojo Bootstrap και είναι στα plugins του Bootstrap Twitter. Χρησιμοποιεί 100% κώδικα Dojo και έχει υποστήριξη για AMD (Asynchronous Module Definition).

Για να χρησιμοποιηθεί το Bootstrap σε μια σελίδα HTML, ο σχεδιαστής του έργου κάνει λήψη του στυλ CSS Bootstrap και περιλαμβάνει μια σύνδεση στο αρχείο HTML.

Αν ο προγραμματιστής θέλει να χρησιμοποιήσει τα στοιχεία JavaScript, θα πρέπει να αναφέρονται μαζί με τη βιβλιοθήκη jQuery στο HTML έγγραφο. Το ακόλουθο παράδειγμα δείχνει πώς λειτουργεί αυτό. Ο κώδικας HTML ορίζει μια απλή μορφή αναζήτησης και μια λίστα των αποτελεσμάτων σε μορφή πίνακα. Η σελίδα αποτελείται από στοιχεία πληροφοριών HTML 5 και CSS σύμφωνα με την τεκμηρίωση του Bootstrap.



Εικόνα 10: Το περιβάλλον του Bootstrap (πηγή: StartBootstrap)



Εικόνα 11: Το περιβάλλον του Bootstrap (πηγή: bootstrapstudio.io)

ΚΕΦΑΛΑΙΟ 4: Hybrid Mobile Apps

4.1 Phone Gap

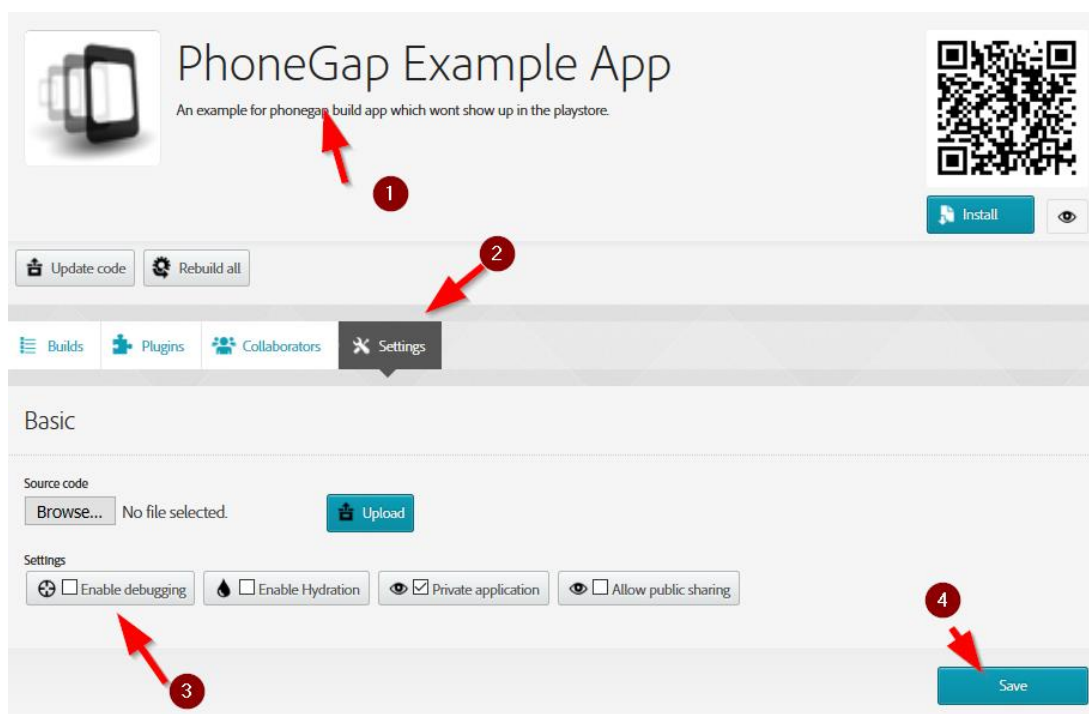
Η μεγάλη ανάπτυξη στον τομέα των κινητών τηλεφώνων και συγκεκριμένα των έξυπνων τηλεφώνων (smartphones) που κινείται με ιλιγγιώδεις ταχύτητες τα τελευταία χρόνια, δημιούργησε μια τεράστια ζήτηση για ενσωμάτωση νέων χαρακτηριστικών και νέων δυνατοτήτων στα κινητά τηλέφωνα. Αυτό είχε σαν αποτέλεσμα πολλές εταιρείες να αρχίσουν να αναπτύσσουν εφαρμογές για κινητά τηλέφωνα, προκειμένου να διεκδικήσουν μερίδιο στη νέα αυτή αγορά για να κερδίσουν νέους πελάτες, αλλά και να αυξήσουν την ικανοποίηση ενδεχομένως των πελατών που ήδη είχαν, από την πιο «παραδοσιακή» μέχρι τώρα παρουσία τους.

Τα έξυπνα κινητά τηλέφωνα, όπως και οι υπολογιστές, έχουν ένα λειτουργικό σύστημα, προκειμένου να λειτουργήσουν, και πάνω στο οποίο αναπτύσσονται και εγκαθίστανται οι εκάστοτε εφαρμογές. Σε αντίθεση όμως με τους υπολογιστές, όπου το μεγαλύτερο μέρος των χρηστών χρησιμοποιούν ηλεκτρονικού υπολογιστές με το λειτουργικό σύστημα Windows, και ακολουθούν σε πιο περιορισμένη έκταση η Apple με το MacOS και το Linux, στα κινητά ο ανταγωνισμός προς το παρόν είναι πολύ πιο έντονος, όπου πέρα από τις αρκετές διαφορετικές εταιρείες κατασκευής τηλεφώνων, συναντάμε και ποικίλα λειτουργικά συστήματα, όπως το Android της Google, το iOS της Apple, το BlackBerry, το Windows Mobile της Microsoft, το WebOS της Hewlett Packard, και το Symbian.

Η ύπαρξη τόσων πολλών διαφορετικών λειτουργικών συστημάτων, όπου το κάθε ένα έχει συγκεκριμένες απαιτήσεις αναφορικά με τις εφαρμογές που θα γραφούν για να λειτουργήσουν πάνω σε αυτό, αλλά και την απαίτηση για ειδικά εργαλεία ανάπτυξης, έχει σαν αποτέλεσμα, πολλές φορές να χρειάζεται η ανάπτυξη της ίδιας εγγενούς εφαρμογής (native application) να γίνεται από την αρχή, καθώς υπάρχει ένα χάσμα (gap), ανάμεσα στα λειτουργικά αυτά συστήματα και στις απαιτήσεις τους.

Το χάσμα αυτό που υπάρχει στα λειτουργικά συστήματα που χρησιμοποιούνται στα κινητά τηλέφωνα έρχεται να καλύψει το PhoneGap. Το Phonegap, είναι μια «πλατφόρμα» ανοιχτού κώδικα, όπου ενσωματώνει ανοιχτά πρότυπα και τεχνολογίες διαδικτύου προκειμένου να γεφυρώσει το κενό που υπάρχει ανάμεσα στα λειτουργικά συστήματα των διαφόρων εταιρειών.

Ο προγραμματιστής που χρησιμοποιεί το Phonegap, χρειάζεται μόνο να γράψει τον κώδικά του, με τη χρήση HTML5, CSS και JavaScript. Στη συνέχεια, από το πακέτο ανοιχτού κώδικα του Phonegap, χρησιμοποιεί το κατάλληλο αρχείο που προορίζεται για το συγκεκριμένο λειτουργικό σύστημα το οποίο στοχεύει ο προγραμματιστής, και με τα ειδικά εργαλεία ανάπτυξης, που διαθέτουν οι εταιρείες των αντίστοιχων λειτουργικών συστημάτων γίνεται η δημιουργία (compile) της εκτελέσιμης εφαρμογής. Προκειμένου να δημιουργηθεί ένα εκτελέσιμο αρχείο για λειτουργικό σύστημα άλλης εταιρείας, ο προγραμματιστής το μόνο που έχει να κάνει, είναι να ενσωματώσει το αντίστοιχο αρχείο από το πακέτο του Phonegap, και με το αντίστοιχο επίσης εργαλείο ανάπτυξης, να δημιουργήσει (compile) την εφαρμογή, χωρίς την ανάγκη να ξαναγράψει από την αρχή το πρόγραμμα, εφόσον ο πηγαίος κώδικας (HTML5, CSS, JavaScript) παραμένει ίδιος. Όσον αφορά κομμάτια τα οποία απαιτούν native υλοποίηση, όπως στην περίπτωση μας το AR περιβάλλον, το phonegap δίνει τη δυνατότητα ενσωμάτωσης native κώδικα στην υβριδική εφαρμογή με την μορφή “επεκτάσεων” (plugins) [17].



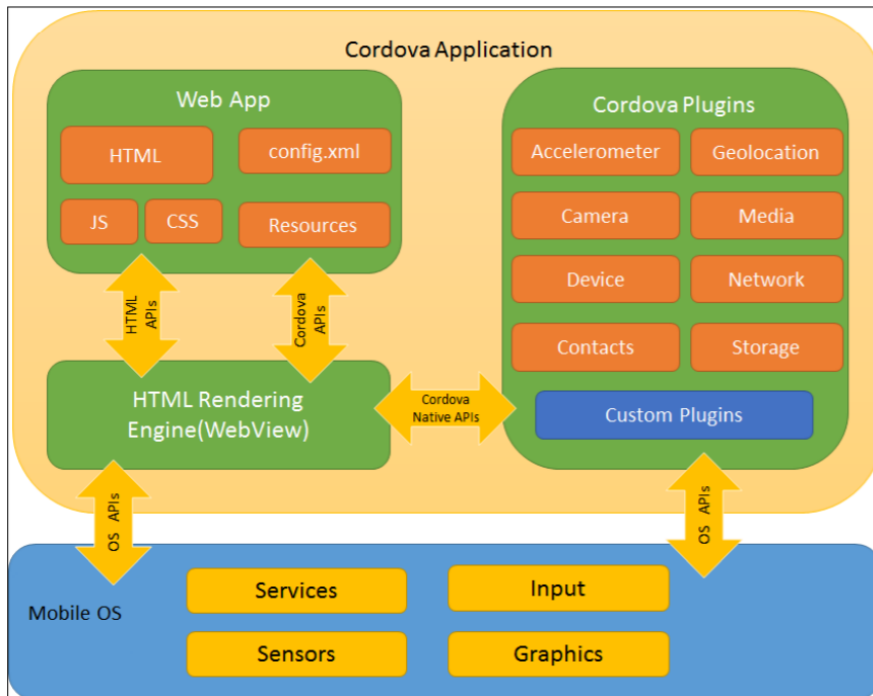
Εικόνα 12: Το περιβάλλον του PhoneGap(πηγή:pointDeveloper.com)

4.2 Apache Cordova

Το Apache Cordova είναι ένα framework ανοιχτού κώδικα για ανάπτυξη εφαρμογών σε κινητά. Επιτρέπει την ανάπτυξη εφαρμογών ανεξαρτήτου πλατφόρμας (cross-platform development) με χρήση των τεχνολογιών προγραμματισμού διαδικτύου όπως HTML5, CSS3 και Javascript. Οι εφαρμογές που δημιουργούνται εκτελούνται μέσα σε wrappers που στοχεύουν στην επιθυμητή πλατφόρμα (android, ios κτλ) και βασίζονται στις αντίστοιχες Διεπαφές Προγραμματισμού Εφαρμογών (Application Programming Interface, API) οι οποίες δίνουν πρόσβαση σε όλες τις δυνατότητες της συσκευής σχετικά με τους αισθητήρες, την κάμερα, τα δεδομένα κτλ. Η χρήση του Apache Cordova συνίσταται όταν:

- Ένας προγραμματιστής εφαρμογών για κινητά θέλει να επεκτείνει τη χρήση της εφαρμογής του σε πέρα από μία πλατφόρμα χωρίς να πρέπει να υλοποιήσει την εφαρμογή με χρήση της γλώσσας που έχει η κάθε πλατφόρμα.
- Ένας προγραμματιστής εφαρμογών διαδικτύου θέλει να προωθήσει τη διαδικτυακή εφαρμογή του σε διαφορετικά app store portals.
- Ένας προγραμματιστής εφαρμογών για κινητά ενδιαφέρεται να συνδυάσει κάποια κομμάτια από μια Native εφαρμογή με ένα WebView (Ειδικό Παράθυρο Φυλλομετρητή, Special Browser Window) και να έχει πρόσβαση σε APIs στο επίπεδο της συσκευής ή θέλει να δημιουργήσει μια διεπιφάνεια μεταξύ Native και WebView αντικειμένων.

Στην παρακάτω εικόνα φαίνεται η βασική αρχιτεκτονική μιας εφαρμογής η οποία έχει υλοποιηθεί με το Apache Cordova [18]:



Εικόνα 13: Αρχιτεκτονική Εφαρμογής Σχεδιασμένης με Apache Cordova

Θετικά

- Ανεξάρτητα από την εμπειρία της πλατφόρμας και της γλώσσας από την πλευρά του διακομιστή, ένας σημαντικός αριθμός προγραμματιστών έχουν εμπειρία με τις HTML, JavaScript και CSS. Το Apache Cordova επιτρέπει στους προγραμματιστές αυτούς να αξιοποιήσουν άμεσα τις υπάρχουσες δεξιότητές τους. Η αξία αυτή δεν μπορεί να υπερεκτιμηθεί, καθώς μειώνει την κατάρτιση και επιτρέπει quick-to-market στάση σε επιχειρήσεις έτοιμες να το θιοθετήσουν.
- Οι εφαρμογές του Cordova εγκαθίστανται όπως μία εγγενής εφαρμογή, και είναι σε θέση να αξιοποιήσουν την γνωστοποιησιμότητα των καταστημάτων εφαρμογών.
- Το Cordova ακολουθεί μια αρχιτεκτονική plugin, κάτι το οποίο σημαίνει ότι η πρόσβαση στο εγγενές API της συσκευής μπορεί να επεκταθεί με ένα ενωτικό τρόπο. Υπάρχουν πολλά plugins για να επιλέξεις – επιτρέποντας στους προγραμματιστές να επικεντρωθούν στις δεξιότητες του ιστού που ήδη έχουν (αυτό είναι επίσης και αδυναμία, όπως θα δούμε παρακάτω).
- Το Cordova είναι εργαλείο ανοικτού κώδικα και δωρεάν, έτσι δεν υπάρχουν έξοδα αδειοδότησης (επίσης πιθανό μειονέκτημα όπως αναφέρεται παρακάτω).

Αρνητικά

- Φυσικά το να προσφέρεις κάτι δωρεάν δεν είναι συνταγή επιτυχίας. Στην πραγματικότητα, η εμφάνιση του PhoneGap Build και Icenium είναι σαφείς αποδείξεις ότι το δωρεάν Apache Cordova είναι θλιβερά ελλιπής. Η δύναμη του να είσαι λογισμικό ανοιχτού κώδικα – και να αξιοποιείς τα ταλέντα ενός ευρέως φάσματος συνεισφερόντων – είναι ευχή και κατάρα μαζί. Αν χρειαστεί αν επεκτείνεις την εφαρμογή σου με ένα προσαρμοσμένο Cordova/PhoneGap plugin, οι πιθανότητες είναι υπέρ του να βρεις κάποιον. Ωστόσο, μπορεί να είναι ξεπερασμένο και να μην υποστηρίζει τις πλατφόρμες που στοχεύεις.
- Η αρχιτεκτονική plugin δουλεύει σωστά αν μπορείς να βρεις τα plugin που χρειάζεσαι ή αν οι προγραμματιστές σου είναι ικανοί να γράψουν το δικό τους plugin εκεί που χρειάζεται. Ωστόσο, οι πιθανότητες είναι ότι έχετε επιλέξει το Cordova, εν μέρει, για να αποφευχθεί η ανάγκη για εξειδικευμένες δεξιότητες εγγενούς προγραμματισμού.
- Η απόδοση των εφαρμογών του Cordova/PhoneGap έχουν συχνά επικριθεί. Το εγγενές περιβάλλον εργασίας πάντα θα ξεπερνάει μια υβριδική λύση, αλλά βελτιώσεις στο υλικό της συσκευής και υλοποιήσεις WebView έχουν μειώσει τη διαφορά. Οι προγραμματιστές θα πρέπει να δώσουν πολύ έμφαση στην απόδοση, που σημαίνει ότι οι γνώσεις τους πάνω σε εργαλεία δημιουργίας προφίλ όπως και ποιό πλαίσιο διεπαφής χρήστη ιστού είναι πιο φιλικό προς τα κινητά είναι σημαντικές γνώσεις.

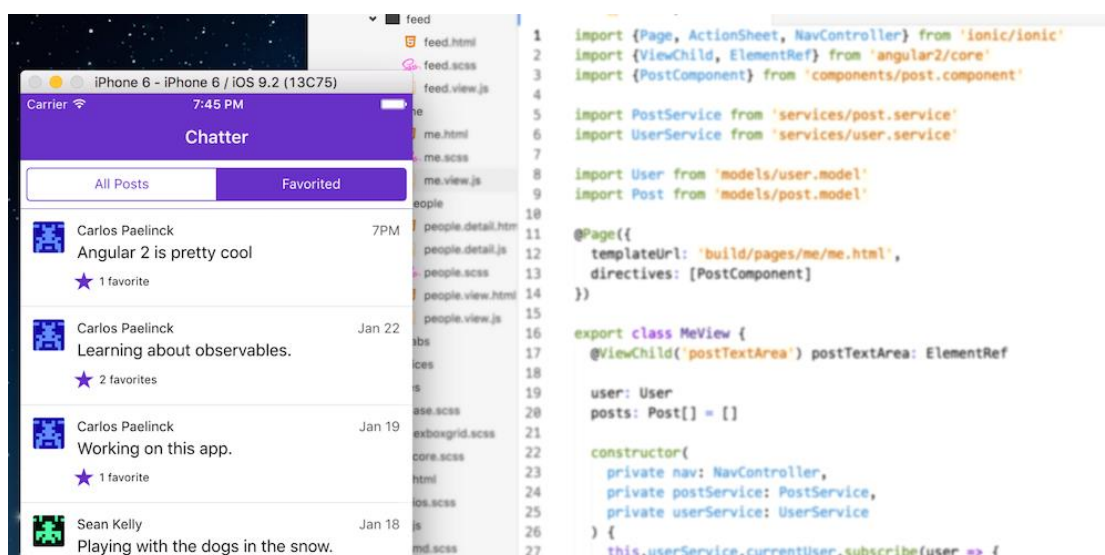
4.3 Ionic

Το Ionic είναι ένα πλαίσιο ανάπτυξης ανεξαρτήτου πλατφόρμας εφαρμογών κινητού υπολογισμού. Αποτελεί ένα εργαλείο ανοιχτού κώδικα και δημιουργήθηκε το 2013, βασισμένο σε HTML5, CSS και JavaScript και το πλαίσιο ελέγχου AngularJS. Οι εφαρμογές αυτές βασίζόμενες σε τεχνολογίες διαδικτύου, μετατρέπονται σε native εφαρμογές και διανέμονται στα ηλεκτρονικά καταστήματα εφαρμογών προς εγκατάσταση στις συσκευές μέσω του Apache Cordova. Εν συνεχεία, από τη δεύτερη έκδοση του μέχρι και την τρίτη που είναι η τελευταία διαθέσιμη αυτή τη στιγμή, γνωστό πλέον ως Ionic 2, βασίζεται σε νεότερες εκδόσεις του πλαισίου Angular και

πλέον χρησιμοποιείται η TypeScript. Στο παρακάτω διάγραμμα παρουσιάζονται οι τεχνολογίες που χρησιμοποιούνται σε κάθε στάδιο ανάπτυξης μια Ionic εφαρμογής.

Το Ionic [19] παρέχει όλες τις δυνατότητες και τα χαρακτηριστικά που διατίθενται στις εγγενείς εφαρμογές κινητού υπολογισμού. Οι χρήστες μπορούν να δημιουργήσουν μια εφαρμογή, να την παραμετροποιήσουν για κινητές συσκευές ανά λειτουργικό σύστημα, και να την αναπτύξουν με τη χρήση του Cordova. Το πλαίσιο ανάπτυξης Ionic εστιάζει στην κατασκευή εφαρμογών που καλύπτουν τα πρότυπα του διαδικτύου και προορίζονται για κάθε σύγχρονη συσκευή. Συνεπώς, η ανάπτυξη λογισμικού βασισμένο στο Ionic υποστηρίζει ένα μεγάλο πλήθος πλατφορμών και εκδόσεων. Σε αυτές συγκαταλέγεται το Android για εκδόσεις από 4.1 και πάνω, το iOS, που υποστηρίζεται από iOS 7 και πάνω, το Blackberry και την πλατφόρμα Windows για εκδόσεις νεότερες των Windows 10.

Η πλατφόρμα Ionic προσφέρει μια σειρά ισχυρών, υβριδικών, backend υπηρεσιών και εργαλείων κινητής υποστήριξης που καθιστούν εύκολη και γρήγορη την κλιμάκωση υψηλής απόδοσης υβριδικών εφαρμογών, με ταχείς ρυθμούς. Οι υπηρεσίες αυτές στηρίζονται στο γεγονός ότι η εφαρμογή κινητού υπολογισμού είναι κατασκευασμένη με τεχνολογίες ιστού σε αντίθεση με τις υπηρεσίες άλλων πλατφορμών. Παρακάτω παρατίθενται οι υπηρεσίες που παρέχει το Ionic (με έκδοση 1.6.5. ή νεότερη), αξιοποιώντας τις δυνατότητες που προσφέρει το Ionic CLI (Command Line Interface)



Εικόνα 14: Ανάπτυξη Εφαρμογών με το Ionic (πηγή: Modus Create)

4.4 Πολυεπίπεδες πλατφόρμες Ανάπτυξης (Cross-Platforms development)

Η ορολογία ανεξάρτητο πλατφόρμας ή πολυ-πλατφόρμα δέχεται ως ελεύθερη μετάφραση cross-platform ή multi-platform είναι ένα χαρακτηριστικό το οποίο αναφέρεται στο λογισμικό που εφαρμόζεται και διαλειτουργεί σε πολλαπλά λειτουργικά συστήματα ή πλατφόρμες υλικού. Στη πράξη επιτρέπει στους προγραμματιστές με μια συγκεκριμένη γλώσσα προγραμματισμού να αναπτύξουν λογισμικό γράφοντας το πρόγραμμα μια μόνο φορά, δέχοντας μικρή έως καθόλου τροποποίηση, τρέχοντας σε όλα τα συστήματα.

Ο κινητήριος παράγοντας πίσω από τις τεχνολογίες cross-platform είναι ότι θα είμαστε σε θέση να χρησιμοποιούμε αυτές τις τεχνολογίες για να εντοπίζουν περισσότερες πλατφόρμες, μειώνοντας την ανάπτυξη του πηγαίου κώδικα. Ένα από τα πλεονεκτήματα που έχουμε με τη χρησιμοποίηση cross-platform εφαρμογών είναι ότι μας παρέχουν πιο εύκολη υλοποίησή με τις γλώσσες HTML και JavaScript παρά με την ανάπτυξη με Java. Αυτό συμβαίνει λόγω της ευκολίας της χρήσης των εργαλείων ανάπτυξης και την εξοικείωση των γλωσσών, μειώνουν τα τεχνικά εμπόδια που μπορεί να εμποδίζουν την έγκριση της φυσικής ανάπτυξης. Παρόλα αυτά απαιτούν μεγάλη εξειδίκευση (και σε διαφορετικές τεχνολογίες) για την χρήση τους.

| Το πλαίσιο cross-platform [20] είναι το κατάλληλο εργαλείο για έναν προγραμματιστή εφαρμογών για κινητές συσκευές, το οποίο μπορεί να τον οδηγήσει στην επιτυχία, διότι του επιτρέπει σε μια πολύ ανταγωνιστική αγορά που θα πρέπει να είναι όσο το δυνατόν πιο αποτελεσματικοί οι μέθοδοι παραγωγής, να προγραμματίσει μια εφαρμογή η οποία θα είναι δυνατό να πραγματοποιηθεί το ίδιο σε όλες τις πλατφόρμες, εξοικονομώντας χρόνο και πόρους. Χωρίς το cross-platform ο χρόνος που απαιτούνταν ήταν αρκετός, αφού έπρεπε να προγραμματιστεί μια εφαρμογή για μια μόνο πλατφόρμα.

| Μια cross-platform στρατηγική μπορεί να εφαρμοστεί μόνο με ένα crossplatform πλαίσιο, όπου κάθε εφαρμογή χρειάζεται να προγραμματιστεί μια μόνο φορά και στη συνέχεια μπορεί να διανεμηθεί σε όλες τις πλατφόρμες, με αποτέλεσμα να πραγματοποιείται πλήρη πρόσβαση στην αγορά κινητών συσκευών.

| Η ανάπτυξη με cross-platform σε κινητές εφαρμογές επιτρέπει την εύκολη ενσωμάτωση με το περιβάλλον cloud. Η ολοκλήρωση της ανταλλαγής για παράδειγμα, κωδικοποιείται σε cross-platform περιβάλλον και θα εκτελεστεί σε πολλαπλές πλατφόρμες κινητών.

| Κάθε πλατφόρμα έχει το δικό της ξεχωριστό πλαίσιο λειτουργίας και η καθεμία ορισμένη ευελιξία σε σχέση με τις υπόλοιπες. Επίσης καθίσταται υποχρεωτική η ανάγκη για τους προγραμματιστές να χρησιμοποιούν περιοριστικά εργαλεία και τεχνολογίες στην εκάστοτε εφαρμογή. Αυτό θέτει σε κίνδυνο τις προτιμήσεις των χρηστών και σε μειονεκτική θέση τους προγραμματιστές γιατί μπορεί να μην είναι σε θέση να αξιοποιήσουν τη μοναδική λειτουργικότητα και τα εργαλεία αντίστοιχης native πλατφόρμας κατά την ανάπτυξη μιας εφαρμογής για πολλαπλές πλατφόρμες.

Τα περισσότερα πλαίσια ανάπτυξης χρησιμοποιούν ένα υποσύνολο της JavaScript. Σε περίπτωση που δεσμευτούν σε ένα πλαίσιο και χρειαστεί να μετακομίσουν σε ένα διαφορετικό πλαίσιο, θα μπορούσε ενδεχομένως να αποδειχθεί δύσκολο να κάνει αυτή τη μετάβαση.

Υπάρχουν πολλά είδη ανάπτυξης εφαρμογών για κινητές συσκευές, τα τρία πιο βασικά εξ' αυτών είναι οι Native (εγγενείς) εφαρμογές, οι εφαρμογές Web και οι υβριδικές εφαρμογές. Οι Native εφαρμογές εκτελούνται εξ ολοκλήρου σε κινητή συσκευή και ασχολούνται άμεσα και αποτελεσματικά με το υλικό της συσκευής. Οι Web εφαρμογές με τη σειρά τους τρέχουν σε Web browsers και οι περισσότεροι από αυτούς εκτελούνται σε έναν απομακρυσμένο διακομιστή, βέβαια τρέχουν σε πολλές πλατφόρμες, αλλά με μικρότερη απόδοση. Οι υβριδικές εφαρμογές είναι ένα μείγμα της εγγενούς και της Web εφαρμογής, δεδομένου ότι οι προγραμματιστές είναι δύσκολο να γράψουν το κώδικα για κάθε πλατφόρμα και είναι αρκετά χρονοβόρο, στρέφονται στα πλαίσια cross-platform που διευκολύνουν το θέμα αυτό με την επίτευξη της λύσης «γράψω-μια-φορά αναπτύσσω-παντού» («write-once deploy-everywhere»)

- Appcelerator Titanium

Η πλατφόρμα ανάπτυξης Appcelerator είναι μια ιδιωτική εταιρεία ανάπτυξης εφαρμογών για κινητές συσκευές με έδρα το Mountain View, της Καλιφόρνια η οποία εισάγαγε την πλατφόρμα Titanium το Δεκέμβριο του 2008. Το Appcelerator

ιδρύθηκε από τους Jeff Haynie και Nolan Wright το 2006 με την ονομασία Hakano, με κύρια δραστηριότητα τις web εφαρμογές. Μετέπειτα τον Οκτώβριο του 2007 μετονομάστηκε σε Appcelerator με ανθρώπινο δυναμικό 160 εργαζομένους. Το Titanium είναι ένα open-source πλαίσιο που επιτρέπει τη δημιουργία εγγενών εφαρμογών (native apps) σε σταθερούς υπολογιστές, tablet, κινητά τηλέφωνα μέσω γλωσσών προγραμματισμού για το διαδίκτυο όπως HTML, PHP, JavaScript, Python.

Το Titanium αποτελείται από ένα SDK δηλαδή μια εργαλειοθήκη ανάπτυξης λογισμικού που παρέχει τα κατάλληλα εργαλεία, μεταγλωττιστές και API για τη δημιουργία εφαρμογών. Το Titanium είναι διαθέσιμο για Mac, Linux και Windows. Για την ανάπτυξη των εφαρμογών για iPhone θα πρέπει να εκτελείτε σε λειτουργικό Mac με το iPhone SDK. Η ανάπτυξη για το Android απαιτεί το Android SDK. Τέλος, μας δίνεται και η δυνατότητα αναβάθμισης στην Titanium έκδοση Enterprise η οποία μας προσφέρει καλύτερη υποστήριξη και επιπλέον υπηρεσίες.

- Corona SDK

Η πλατφόρμα ανάπτυξης Corona SDK δημιουργήθηκε από τον Walter Luh, ο ιδρυτής της Corona Labs, ο οποίος μετά την αναχώρηση του από την Adobe το 2007, ήταν ο κύριος αρχιτέκτονας του Flash Lite. Τον Ιούνιο του 2009 κυκλοφορεί το πρώτο Corona SDK beta δωρεάν.

Λίγο καιρό αργότερα τον Απρίλιο του 2010 κυκλοφορεί για πρώτη φορά το Corona SDK 2.0 beta έχοντας ένα νέο χαρακτηριστικό, την υποστήριξη cross-platform σε iPhone, iPad, Android συσκευές με αποτέλεσμα να επιτρέπει στους προγραμματιστές λογισμικού να χτίσουν κινητές εφαρμογές.

- Intel XDK

Μια νέα λύση για την ανάπτυξη της HTML5, CSS, JavaScript είναι το Intel XDK. Είναι ένα cross-platform εργαλείο ανάπτυξης που παρέχει μια απλοποιημένη ροή εργασίας και επιτρέπει στους προγραμματιστές εύκολα και γρήγορα να σχεδιάσουν, να χτίσουν και να διορθώσουν web και υβριδικές εφαρμογές σε διάφορα καταστήματα app.

Μερικά από τα κύρια χαρακτηριστικά του XDK είναι ότι σου δίνει τη δυνατότητα να κάνεις δωρεάν download και το πρώτο πράγμα που παρατηρεί κανείς,

είναι το πόσο εύκολα μπορείς να κάνεις εγκατάσταση σε σχέση με τις υπόλοιπες πλατφόρμες. Το XDK περιλαμβάνει επίσης ένα γραφικό επεξεργαστή, ο οποίος λείπει σε μεγάλο βαθμό από όλες τις κινητές πλατφόρμες HTML5.

Το Intel XDK έχει τη δυνατότητα να αναπτυχθεί σε οποιαδήποτε πλατφόρμα, επειδή η μεταγλώττιση γίνεται στο cloud. Άλλες πλατφόρμες προσφέρουν μια παρόμοια υπηρεσία για την πολλαπλή ανάπτυξη της πλατφόρμας, αλλά είναι περιορισμένη. Επίσης υποστηρίζει πλαίσια όπως το Bootstrap και jQuery Mobile. Αυτά τα στοιχεία διεπαφής χρήστη μας επιτρέπουν να οικοδομήσουμε εύκολα και γρήγορα το περιβάλλον εργασίας της εφαρμογής μας. Ένα μειονέκτημα που έχει το XDK είναι ότι δεν μπορούμε να ενημερώσουμε το πρόγραμμα επεξεργασίας της βάσης που είναι ενσωματωμένη με το XDK. Μόλις δημιουργήσουμε ένα app πρέπει να βρούμε ένα τρόπο για να δοκιμάσουμε τη λειτουργικότητα του, αυτό θα πραγματοποιηθεί με τη βοήθεια ενός εξομοιωτή για κινητές συσκευές και έτσι θα έχουμε τη δυνατότητα να δοκιμάσουμε σε πραγματικές συσκευές και τις επιδόσεις του προφίλ χωρίς να χρειάζεται κάποια περαιτέρω εγκατάσταση.

- Sencha Touch 2

Το Sencha Touch 2 είναι ένα υψηλής απόδοσης πλαίσιο ανοιχτού κώδικα για κινητές εφαρμογές cross-platform. Το αποτέλεσμα της εφαρμογής είναι υβριδικό και είναι γραμμένο σε γλώσσες Web: HTML5, CSS3 και JavaScript. Στόχος του πλαισίου είναι να επιτρέπει στους προγραμματιστές να δημιουργούν εύκολα και γρήγορα επίγειες κινητές εφαρμογές με HTML5 σε υποστηριζόμενες πλατφόρμες που λειτουργούν με Android, Windows, iOS, BlackBerry, Kindle και Bada. Για την ανάπτυξη εφαρμογών απαιτείται ένας web server ο οποίος θα τρέχει σε τοπικό επίπεδο. Επίσης το συγκεκριμένο πλαίσιο εκμεταλλεύεται την επιτάχυνση υλικού ώστε να παραδώσει μια απίστευτη εμπειρία σε app.

- Xamarin

Το Xamarin επιτρέπει την ανάπτυξη cross-platform εφαρμογών, το οποίο είναι εξ' ολοκλήρου γραμμένο σε C#. Οι εφαρμογές μοιράζονται τον ίδιο κώδικα για iOS, Android, Windows, Mac κ.ά. και παράγει μια εγγενή εφαρμογή ή ολοκληρωμένη εφαρμογή σε .NET που παρέχει το IDE (MonoDevelop IDE και Xamarin plug-in για το Visual Studio).

Οτιδήποτε μπορούμε να κάνουμε σε Objective-C ή Java, μπορούμε να το κάνουμε σε C# για όλες τις διαθέσιμες πλατφόρμες, παράδειγμα δεν χρειάζεται να εγκαταστήσουμε το Java SDK για την ανάπτυξη σε Android, διότι θα χρησιμοποιήσουμε το εργαλείο Visual Studio που μας παρέχει όλες τις απαραίτητες λειτουργίες για την ανάπτυξη εφαρμογών. Το Xamarin έχει πάνω από 500.000 προγραμματιστές σε περισσότερες από 120 χώρες σε όλο τον κόσμο.

- Unity3D

Το Unity 3D είναι μια cross-platform 3D μηχανή δημιουργίας παιχνιδιού που αναπτύχθηκε από την Unity Technologies, συμπεριλαμβανομένου και ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Μπορούμε να το χρησιμοποιήσουμε για την ανάπτυξη παιχνιδιών σε desktop, ιστοσελίδες, κονσόλες και κινητές συσκευές. Η πρώτη έκδοση που ανακοινώθηκε το 2005 ήταν μόνο για Mac OS, στο παγκόσμιο συνέδριο για developers της Apple, μετέπειτα επεκτάθηκε και είναι διαθέσιμο σε περισσότερες από δέκα πλατφόρμες.

Η πλήρης επαγγελματική σουίτα, το Unity Pro είναι διαθέσιμο από \$1500 το χρόνο ή \$75 το μήνα σε άτομα για προσωπική χρήση ή εταιρείες για επαγγελματική χρήση. Στις 3 Μαρτίου του 2015 με την απελευθέρωση του Unity 5.0, η Unity Technologies έκανε την πλήρη λειτουργία να διατίθεται δωρεάν, συμπεριλαμβανομένων όλων των χαρακτηριστικών και την υποστήριξη. Τέλος, το Unity 3D εργαλείο παρέχει ένα οπτικό βοήθημα που μειώνει την ποσότητα του χρόνου για την ολοκλήρωση του κώδικα και έτσι οι προγραμματιστές μπορούν απλά να σύρουν τους κώδικες από το ένα αντικείμενο στο άλλο. Υποστηρίζει τις περισσότερες πλατφόρμες με μόνη εξαίρεση τους χρήστες των iPhone που τους περιορίζει να το κατεβάσουν για μια δοκιμαστική περίοδο των 30 ημερών.

- Qt

Το Qt είναι μια συνδρομητική υπηρεσία για την ανάπτυξη εφαρμογών ανεξαρτήτου πλατφόρμας. Πριν την αγορά του, παρέχεται η δυνατότητα δωρεάν δοκιμής, με τους χρήστες να μπορούν να επωφεληθούν αυτής ώστε να καταλήξουν στην καλύτερη απόφαση επιλογής τεχνολογιών. Στην υπηρεσία αυτή χρησιμοποιείται η γλώσσα C++ για την ανάπτυξη λογισμικού. Το πλαίσιο ανάπτυξης Qt δίνει τη δυνατότητα προσομοίωσης εκτέλεσης της εφαρμογής σε διάφορες συσκευές κινητού

υπολογισμού. Ένα άλλο θετικό στοιχείο αυτού του εργαλείου ανάπτυξης ανεξαρτήτου πλατφόρμας είναι η ταχύτητα. Τα εργαλεία που προσφέρει βοηθούν στη διάθεση της εφαρμογής στα ηλεκτρονικά καταστήματα πολύ πιο γρήγορα. Συνολικά, είναι πραγματικά εύκολο στη χρήση, μια και δεν έχει περίπλοκη διεπαφή όπως κάποια άλλα εργαλεία ανάπτυξης. Το πλήθος των περιβαλλόντων ανάπτυξης εφαρμογών ανεξαρτήτου πλατφόρμας αυξάνεται συνεχώς και τα περισσότερα από αυτά διαφέρουν μεταξύ τους σε πολλά σημεία.

- Alpha Anywhere

Το Alpha Anywhere, για παράδειγμα, είναι ένα από αυτά και παρέχει στον χρήστη τη δυνατότητα να επιλέξει εκείνος την γλώσσα προγραμματισμού που θα χρησιμοποιήσει, επιλέγοντας ανάμεσα σε Xbasic, NTO#, VB.NET, JavaScript ή οποιαδήποτε γλώσσα υποστηρίζεται από .NET, όπως C++ και Cobra. Ένα ακόμη τέτοιο πλαίσιο ανάπτυξης είναι το 5app, το οποίο διαφέρει σε μεγάλο βαθμό από τα άλλα εργαλεία, λόγω της σχεδίασής του για επιχειρήσεις που αναπτύσσουν εφαρμογές για τους υπαλλήλους τους. Το 5app δίνει έμφαση στην ασφάλεια των δεδομένων και χρησιμοποιεί JavaScript και HTML5 για την ανάπτυξη των εφαρμογών. Κάποια ακόμη από τα εργαλεία ανάπτυξης εφαρμογών ανεξαρτήτου πλατφόρμας είναι τα RhoMobile Suite, Cocos2d, Unity3d, Webix, Native Script, Monocross, Adobe, Codename One και Convertigo. Στον πίνακα που ακολουθεί παρουσιάζεται μία συγκριτική μελέτη για τις πλατφόρμες ανάπτυξης [21], [22], [23].

Εργαλεία CrossPlatform	Υποστηριζόμενες πλατφόρμες	Γλώσσα Προγραμματισμού	Περιβάλλον Ανάπτυξης	Άδεια
PhoneGap	Android, iOS, webOS, Windows Phone, Symbian, Blackberry and Bada	HTML5, CSS3 and JavaScript.	Eclipse, and XCode	Open- Source
Appcelerator Titanium	iOS, Android and BlackBerry.	HTML, CSS and JavaScript.	Eclipse-based Titanium Studio.	Open- Source
Xamarin	iOS, Android, and Windows Phone	C#	MonoDevelop IDE	30 days trial.
Unity 3D	iOS, Android, and Windows Phone	C#, JavaScript and Boo	Maya, 3DS Max.	30 days trial.

Corona SDK	iOS, Android, Kindle Fire and NOOK.	HTML5.	Lua scripting language.	Free unlimited trial, paid to publish.
Flash Professional	iOS, Android, and BlackBerry	JavaScript Flash Language (JSFL) and ActionScript	AIR 3.2 SDK	30 days trial.
Rhodes	Android, BlackBerry, iOS, Windows Phone and Symbian	HTML and Ruby.	RhoStudio IDE - Eclipse, Visual Studio, Netbeans, IntelliJ, and Textemate.	Open-Source

Συγκριτικός Πίνακας για CrossPlatform Εργαλεία

4.5 Υποστήριξη εφαρμογών κινητού υπολογισμού από τεχνολογίες νέφους

Το υπολογιστικό νέφος σύμφωνα με το US National Institute for Standards and Technology είναι «ένα μοντέλο που δίνει τη δυνατότητα της συνεχούς, εύκολης και υψηλών απαιτήσεων πρόσβασης σε μια κοινόχρηστη συλλογή ρυθμιζόμενων υπολογιστικών πόρων, οι οποίοι τροφοδοτούνται και απελευθερώνονται με ελάχιστη προσπάθεια διαχείρισης και αλληλεπίδρασης παροχής υπηρεσιών»[23]. Χρησιμοποιώντας απλούς όρους, θα μπορούσαμε να πούμε ότι τεχνολογία υπολογιστικού νέφους, ή νέφος ή αλλιώς cloud computing, αποτελεί οποιοδήποτε λογισμικό χρησιμοποιεί ο χρήστης, το οποίο όμως δεν τρέχει στον υπολογιστή του, αλλά τρέχει στο διαδίκτυο.

Οι υπηρεσίες του υπολογιστικού νέφους παρέχουν στο χρήστη τη δυνατότητα αποθήκευσης δεδομένων, όπως μουσική, βίντεο, έγγραφα, εικόνες κ.α., καθώς και την πρόσβαση σε εφαρμογές αξιοποιώντας απλά το διαδίκτυο. Ως κοντινότερο παράδειγμα χρήσης θα μπορούσε να θεωρηθεί η πρόσβαση στο ημερολόγιο, ο καθημερινός έλεγχος, η αποστολή αλλά και η λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου, όπου όλα αυτά τα δεδομένα είναι αποθηκευμένα σε διακομιστές και η πρόσβαση τους γίνεται απομακρυσμένα χάρη στο νέφος. Γνωστές εφαρμογές ηλεκτρονικού ταχυδρομείου που αξιοποιούν τις δυνατότητες αυτές είναι το Gmail, το Yahoo ή το Hotmail. Μία ακόμα αξιοσημείωτη υπηρεσία του νέφους είναι η

αποθήκευση δεδομένων (μουσική, βίντεο, έγγραφα, εικόνες κ.α.) αλλά και η κοινοποίηση τους μέσω εφαρμογών όπως το Google Drive, το OneDrive, το Dropbox, το Microsoft Skydrive, το Amazon και το iCloud.

Η εξοικονόμηση χρημάτων είναι ένα από τα κυριότερα πλεονεκτήματα του υπολογιστικού νέφους για τους χρήστες. Αξιοποιώντας τις εφαρμογές του νέφους, δεν είναι απαραίτητη η αγορά λογισμικών ή ακόμα και η αγορά και συντήρηση ακριβών εξυπηρετητών και εγκαταστάσεων αποθήκευσης δεδομένων. Έτσι εξοικονομούνται χρήματα, χώρος γραφείων και έξοδα για εσωτερικό προσωπικό υποστήριξης των παρεχόμενων υπηρεσιών πληροφορικής. Από αυτό επωφελούνται εξατομικευμένοι χρήστες διαδικτύου, αλλά κυρίως οι επιχειρήσεις ή οργανισμοί. Από έρευνες έχει προκύψει ότι το 81% των επιχειρήσεων που χρησιμοποιούν ήδη το υπολογιστικό νέφος ανέφεραν μείωση των δαπανών τους ως 30%.

Η χρήση του υπολογιστικού νέφους χαρακτηρίζεται από μεγάλη αξιοπιστία σχετικά με την απώλεια δεδομένων. Τα δεδομένα είναι αποθηκευμένα σε διακομιστές σε διάφορα κέντρα δεδομένων (data centers) σε όλο τον πλανήτη, κάτι που ελαχιστοποιεί την πιθανότητα να χαθούν. Δεδομένου ότι όλα τα δεδομένα σας είναι αποθηκευμένα στο σύννεφο η δημιουργία αντιγράφων και η ανάκτησή τους είναι σχετικά πολύ πιο εύκολη από ότι σε μια φυσική συσκευή.

Ωστόσο, όπως ακριβώς και στην απευθείας σύνδεση, υπάρχει πάντα ο κίνδυνος κάποιος να προσπαθήσει να αποκτήσει πρόσβαση στα προσωπικά δεδομένα ενός χρήστη, έτσι είναι σημαντική η επιλογή ενός ισχυρού κωδικού πρόσβασης και η προσοχή σε τυχόν ρυθμίσεις απορρήτου κατά την χρήση μιας υπηρεσίας. Οι εταιρείες παροχής υπηρεσιών νέφους εγγυώνται ότι υπάρχει ασφάλεια των δεδομένων που ανεβάζει κάποιος στο νέφος, χάρη στην κρυπτογράφηση που πραγματοποιείται. Στις περισσότερες εταιρείες, όμως, η κρυπτογράφηση γίνεται μετά το ανέβασμα των δεδομένων στο νέφος, πράγμα που δεν είναι ιδιαίτερα ασφαλές.

Τα κέντρα δεδομένων και οι εξυπηρετητές, ωστόσο, ευθύνονται για το 2% της παγκόσμιας ρύπανσης του περιβάλλοντος, λόγω της ανάγκης μεγάλων ποσοτήτων ενέργειας για να λειτουργούν όλο το 24ωρο. Οι χώρες στις οποίες φιλοξενούνται συγκροτήματα κέντρων δεδομένων έχουν αυστηρή νομοθεσία και υποχρεώνουν τις εταιρείες να χρησιμοποιούν εναλλακτικές μορφές ενέργειας (ήλιος, αέρας, νερό κ.α.).

Κάποιες εταιρείες σχεδιάζουν κέντρα δεδομένων σε περιοχές με πολύ κρύο έτσι ώστε να μην χρειάζονται πολλή ενέργεια.

Το υπολογιστικό νέφος κατηγοριοποιείται ανάλογα με τον τύπο της υπηρεσίας που προσφέρει σε μοντέλα παροχής υπηρεσιών. Το πρώτο είναι αυτό που χρησιμοποιείται πιο συχνά και ονομάζεται Λογισμικό ως Υπηρεσία (Software as a Service - SaaS). Τα μοντέλα Πλατφόρμα ως Υπηρεσία (Platform as a Service - PaaS) και Backend ως Υπηρεσία (Backend as a Service - BaaS ή και Mobile Backend as a Service - MBaaS) μας παρέχουν ένα περιβάλλον εργασίας για ανάπτυξη εφαρμογών. Τέλος, οι Υποδομές ως Υπηρεσία (Infrastructure as a Service - IaaS) μας παρέχουν τους πόρους που χρειαζόμαστε μέσω του ίντερνετ, προκειμένου να στήσουμε ένα δικό μας εικονικό διακομιστή, μια ιστοσελίδα, ή μια πλήρη online υπηρεσία.

Πιο αναλυτικά, με το μοντέλο SaaS (Software-as-a-Service) παρέχεται η δυνατότητα της μίσθωσης διαφόρων εφαρμογών. Η ουσία του μοντέλου αυτού, είναι ότι ο πελάτης δεν αγοράζει το λογισμικό αλλά πληρώνει για την υπηρεσία που του παρέχει. Ο πελάτης δεν διαθέτει κανένα λογισμικό στους διακομιστές του, αλλά χρησιμοποιεί τις εγκαταστάσεις της υπηρεσίας, με μόνο μέλημα την πληρωμή της υπηρεσίας, της συντήρηση του διακομιστή και τη διαβούλευση. Αυτή την περίοδο πολλές startups εταιρείες, επιλέγουν αυτό το επιχειρηματικό μοντέλο για την παροχή λύσεων λογισμικού στους καταναλωτές. Ένα πλεονέκτημα για τους τελικούς χρήστες είναι ότι δεν χρειάζεται να ενημερώνει ποτέ το λογισμικό, ενώ στο νέφος τρέχει πάντα την πιο πρόσφατη έκδοση. Όσον αφορά το μοντέλο PaaS (Platform as a service), είναι ένα περιβάλλον ανάπτυξης, εγκατάστασης και διανομής λογισμικού και παρέχεται ως υπηρεσία σε προγραμματιστές μέσω του διαδικτύου. Διευκολύνει την ανάπτυξη του λογισμικού και τον διαμοιρασμό του απαλλαγμένο από το κόστος και την πολυπλοκότητα της αγοράς και της διαχείρισης της αναγκαίας υποδομής, ενώ ταυτόχρονα παρέχει όλες τις διευκολύνσεις που απαιτούνται για την υποστήριξη του κύκλου ζωής του λογισμικού.

Με το μοντέλο PaaS οι προγραμματιστές μπορούν να δημιουργούν εφαρμογές διαδικτύου (web applications) χωρίς να χρειάζεται να εγκαταστήσουν κανένα «εργαλείο» τοπικά στον υπολογιστή τους ενώ μπορούν να διανείμουν τα λογισμικά που παράγουν χωρίς χρήση κανενός ιδιαίτερου εργαλείου διαχείρισης.

4.6 Ενημέρωση Εφαρμογών

Οι εφαρμογές κινητού υπολογισμού δημιουργήθηκαν, αρκετό διάστημα πριν το διαδίκτυο αποκτήσει δημοτικότητα. Ένας προγραμματιστής, για παράδειγμα, ο Larry Wall, δημιούργησε το 1987, το Perl, μία δημοφιλή γλώσσα προγραμματισμού για τον διακομιστή. Αυτό έγινε 7 χρόνια πριν το διαδίκτυο ξεκινήσει να κερδίζει δημοτικότητα, έξω από ακαδημαϊκούς και τεχνολογικούς κύκλους. Παρόλο που οι πρώτες βασικές εφαρμογές ήταν σχετικά απλές, η διαδικασία αναβάθμισής τους εκείνη την εποχή ήταν αρκετά πολύπλοκη και χρονοβόρα [21].

Οι ενημερώσεις των ευρύτερων λειτουργικών συστημάτων, των προγραμμάτων κτλ, είναι μια διαδικασία που ονομάζεται “patching”. Τα patches, σχεδιάζονται με τέτοιο τρόπο ώστε να ενημερώνουν το πρόγραμμα για το οποίο προορίζονται, να διορθώσουν πιθανά λάθη που υπάρχουν καθώς και να το αναβαθμίσουν. Η διαδικασία αυτή περιλαμβάνει επίσης, τη διόρθωση σφαλμάτων ασφαλείας, τη βελτίωση της λειτουργίας του προγράμματος (π.χ. να γίνει πιο φιλικό προς το χρήστη), καθώς και να αυξήσει τις επιδόσεις του εκάστοτε προγράμματος (μείωση κατανάλωσης πόρων ή χρόνου ανταπόκρισης κτλ). Μία τραγική ειρωνεία που αναδεικνύεται συνήθως με τις ενημερώσεις των προγραμμάτων, είναι η εμφάνιση επιπλέον προβλημάτων στη λειτουργία τους, κάτι που απαιτεί δημιουργία νέας έκδοσης και συνεπώς την ανάγκη νέας ενημέρωσης για την αντιμετώπιση των καινούργιων δυσλειτουργιών. Κάπως έτσι αναδείχθηκε και η ανάγκη του version control, ο έλεγχος δηλαδή μεταξύ των εκδόσεων με την ασφαλή επιστροφή σε παλαιότερη έκδοση που χαρακτηριζόταν από καλύτερη ευστάθεια.

Οι ενημερώσεις λογισμικού κλειστού κώδικα είναι ένα είδος ενημερώσεων που διανέμονται συνήθως ως εκτελέσιμα αρχεία και ο χρήστης καλείται απλά να τα εκτελέσει στον υπολογιστή του, χωρίς να του δίνεται η δυνατότητα αλλαγής ή αφαίρεσης μέρους του κώδικα. Στην αντίπερα όχθη, υπάρχει το ανοιχτό λογισμικό, που είναι ελεύθερο προς χρήση από οποιονδήποτε, χωρίς συνήθως να υπόκειται σε κάποιου είδους άδεια. Από τις λίγες υποχρεώσεις όσων θέλουν να δουλέψουν με τέτοιου είδους λογισμικά, είναι η δυνατότητα αλλαγών που γίνονται από κάποιον τρίτο, να είναι και αυτές ελεύθερες προς τροποποίηση, χρήση κτλ. Οι ενημερώσεις σε λογισμικά ανοιχτού κώδικα, γίνονται με διάφορους τρόπους. Από τους πιο γνωστούς είναι η δυνατότητα απόκτησης του πηγαίου κώδικα και η μετατροπή του σε

εκτελέσιμο αρχείο από τον εκάστοτε χρήστη, όπως επίσης και η άμεση απόκτηση ενός εκτελέσιμου αρχείου, το οποίο όμως διαθέτει ολόκληρο τον κώδικα που το συνθέτει, καθώς και οδηγίες για τη λειτουργία του. Οι δύο παραπάνω τρόποι, τροποποιούν το υπάρχον λογισμικό που κατέχει ο χρήστης, είτε με τη διόρθωση του δυαδικού αρχείου, συμπεριλαμβάνοντας σε αυτό τις διορθώσεις ή την πλήρη αντικατάσταση του. Παράδειγμα, στους πρώτους μικροεπεξεργαστές που κυκλοφόρησαν, το λειτουργικό σύστημα που είχε ο μικροεπεξεργαστής, υπήρχε εγκατεστημένο κάποιο εργαλείο το οποίο δεχόταν ενημερώσεις συνήθως από κάποιο αρχείο κειμένου (π.χ. text.txt κτλ), διάβαζε τα δεδομένα του αρχείου αυτού και στη συνέχεια εφάρμοζε τις διορθώσεις στο αντίστοιχο λογισμικό. Εδώ αξίζει να σημειωθεί ότι οι διορθώσεις και γενικότερα οι αλλαγές, μετατρέποντουσαν αρχικά σε δυαδική μορφή και ύστερα εφαρμοζόντουσαν στο εκτελέσιμο λογισμικό.

Με την ευρεία έλευση και χρήση του διαδικτύου, σιγά σιγά ο χρήστης, καθώς και όσοι παράγουν λογισμικά και διαθέτουν ενημερώσεις για αυτά, άρχισαν να χρησιμοποιούν το διαδίκτυο για την χρήση τους. Ο κατασκευαστής τοποθετούσε το λογισμικό ή τις ενημερώσεις του λογισμικού σε χώρο στον οποίο είχαν όλοι πρόσβαση μέσω του διαδικτύου και στη συνέχεια ο χρήστης το κατέβαζε τοπικά στον υπολογιστή του για να κάνει τις ενημερώσεις. Ο τρόπος λειτουργίας των ενημερώσεων όσο αφορά το επίπεδο της εγκατάστασής τους δεν διέφερε πολύ από την αρχική διαδικασία [22].

Η γνωριμία και εξοικείωση του κόσμου με το διαδίκτυο, έφερε ριζικές αλλαγές στην καθημερινότητα των χρηστών, καθώς και στην εργαλειακή αντιμετώπιση και χρήση του ίδιου του διαδικτύου. Πολύ γρήγορα οι εταιρείες παραγωγής λογισμικού, ανέπτυξαν την τεχνολογία υπολογιστικού νέφους. Πάνω σε αυτή την τεχνολογία, φάνηκε πολύ γρήγορα, ότι χρειάζονται περισσότερα εργαλεία/λογισμικά για την βέλτιστη εκμετάλλευσή της. Αυτό μας οδήγησε, στη δημιουργία λογισμικών τύπου Cloud Storage (App Store στο Android, iCloud στο iOS κτλ). Τα λογισμικά αυτά, λειτουργούν σαν “διαμεσολαβητές” τόσο για τον δημιουργό του λογισμικού όσο και για τον χρήστη. Ταυτόχρονα πρόσφερε πάρα πολλές διευκολύνσεις. Ο δημιουργός του λογισμικού το τοποθετεί στα διάφορα Cloud Storages που διαθέτουν διάφορες εταιρίες, στη συνέχεια ο χρήστης απλά βρίσκει το λογισμικό που τον ενδιαφέρει, έχοντας εγκατεστημένο στον υπολογιστή του, στο κινητό του κτλ το αντίστοιχο native application του παρόχου, το κατεβάζει

τοπικά στην συσκευή του, το εγκαθιστά και έτσι όλη η διαδικασία των ενημερώσεων αυτοματοποιείται. Η native εφαρμογή, όταν ο δημιουργός του λογισμικού δημοσιεύσει κάποια νέα έκδοση, το κατεβάζει τοπικά και το εγκαθιστά χωρίς να χρειάζεται κάποια ιδιαίτερη τεχνική γνώση. Η διαδικασία αυτή μπορεί να ρυθμιστεί ώστε να πραγματοποιείται αυτόματα η αποδοχή και η εκτέλεση της ενημέρωσης.

Η παραπάνω τεχνολογία αποτελεί ένα είδος επανάστασης αρχικά στη χρήση του διαδικτύου αλλά και στη λειτουργία και τη χρηστικότητα των ηλεκτρονικών υπολογιστών και όλων των ηλεκτρικών συσκευών που υποστηρίζουν την διασύνδεση με το διαδίκτυο. Επίσης, πολύ σημαντικό, έλυσε σε μεγάλο βαθμό, το πρόβλημα που υπήρχε παλιότερα στις ενημερώσεις ασφαλείας, που είτε ο χρήστης δύσκολα μπορούσε να τις βρει είτε δεν μπορούσε να τις εγκαταστήσει λόγω της τεχνικής εξοικείωσης που χρειαζόταν. Με την έλευση της τεχνολογίας υπολογιστικού νέφους οποιαδήποτε ενημέρωση ασφαλείας για το εκάστοτε λογισμικό γίνεται αυτόματα.

Θα γίνει αναφορά σε μερικά από τα πιο γνωστά ηλεκτρονικά καταστήματα εφαρμογών, όπως το App Store για iOS λογισμικά, το Google Play Store ή Android Market της Google για Android και το Windows Store της Microsoft για συσκευές Windows. Επίσης θα αναφερθεί ο τρόπος λειτουργίας των καταστημάτων, όπως η εγκατάσταση των προϊόντων (προγράμματα λογισμικού), ο τρόπος ενημέρωσής τους, καθώς και η πολιτική του κάθε καταστήματος, πως δηλαδή γίνεται αποδεκτή μια εφαρμογή στη “φιλοξενία” από το κατάστημα, το κόστος που έχει αυτή η διαδικασία κτλ [23].

Το Google Play Store, είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο αναπτύσσεται από την ίδια τη Google και λειτουργεί σε όλες τις συσκευές κινητού υπολογισμού (κινητά τηλέφωνα, tablets, κ.α.) με λειτουργικό σύστημα Android. Το λογισμικό που φιλοξενείται και είναι διαθέσιμο προς χρήση, είτε δωρεάν είτε επί πληρωμή, από τους χρήστες Android συσκευών, αναπτύσσεται μέσω του εργαλείου Android software development kit (SDK). Το Google Play, λειτουργεί επίσης και ως κέντρο δεδομένων (Media Center), δίνοντας τη δυνατότητα στους χρήστες να ακούσουν μουσική, να διαβάσουν περιοδικά και βιβλία ή να δουν ταινίες και τηλεοπτικά προγράμματα.

Το 2016 το Google Play Store μετρούσε πάνω από 82 λήψεις εφαρμογών, ενώ το Φλεβάρη του 2017 είχε στα αποθετήριά του πάνω από 2.7 εκατομμύρια

δημοσιευμένες εφαρμογές, σε πάνω από 145 χώρες στις οποίες οι χρήστες Android συσκευών μπορούν να έχουν πρόσβαση. Τα παραπάνω νούμερα, μας δίνουν μια μικρή εικόνα, για το τεράστιο μέγεθος αυτής της πλατφόρμας, τις δυνατότητες που δίνονται στον χρήστη, καθώς επίσης αποτελούν ζωντανή απόδειξη πως οι τεχνολογίες νέφους έχουν έρθει για να μείνουν [24].

Για να γίνει διαθέσιμη μία εφαρμογή, ένας προγραμματιστής ή μια εταιρία, χρειάζεται να πληρώσει μια φορά 25 δολάρια για να αποκτήσει έναν λογαριασμό Google Play Developer Console. Το λογισμικό που είναι πλέον διαθέσιμο από το Google Play Store, μπορεί να διατίθεται δωρεάν προς χρήση ή επί πληρωμή. Αναλυτικότερα:

1. Ο ιδιοκτήτης του λογισμικού έχει τη δυνατότητα να ελέγχει σε ποιες χώρες θα είναι διαθέσιμο το λογισμικό του, καθώς επίσης και το κόστος αγοράς του λογισμικού από τον τελικό χρήστη, με τη δυνατότητα να το διαφοροποιεί από χώρα σε χώρα.

2. Στην περίπτωση που το λογισμικό διατίθεται επί πληρωμή, ο ιδιοκτήτης, δικαιούται το 70% του κόστους και το υπόλοιπο 30% πάει σε διάφορα πάγια έξοδα καθώς και ένα ποσοστό από αυτό πηγαίνει στην Google.

Άλλες δυνατότητες που παρέχονται:

1. Ο ιδιοκτήτης μπορεί να κυκλοφορήσει κάποια δοκιμαστική έκδοση του προϊόντος και να την διαθέσει σε συγκεκριμένους χρήστες, ομάδες κτλ.

2. Ο ιδιοκτήτης μπορεί να διαθέσει τις αναβαθμίσεις του λογισμικού του σταδιακά, απευθυνόμενος πρώτα σε λίγους χρήστες, στη συνέχεια σε περισσότερους κ.ο.κ.

3. Οι χρήστες έχουν τη δυνατότητα να προπαραγγείλουν λογισμικό, όπως επίσης και ταινίες, μουσική, βιβλία κτλ.

4. Οι χρήστες έχουν τη δυνατότητα να ζητήσουν αποζημίωση μέχρι και 48 ώρες από τη στιγμή που αγόρασαν ένα λογισμικό.

Το Google Play Store, όπως και τα άλλα καταστήματα εφαρμογών, κάνουν ενημερώσεις στις συσκευές, βασισμένες στην τεχνολογία υπολογιστικού νέφους. Μόλις υπάρχει μια ενημέρωση για ένα λογισμικό, το Google Play Store, εμφανίζει μια ειδοποίηση στον χρήστη ότι υπάρχει η δυνατότητα ενημέρωσης κάποιας

εφαρμογής. Ο χρήστης από την πλευρά του, πρέπει να δεχτεί να γίνει εγκατάσταση αυτής της ενημέρωσης και μόλις γίνει η αποδοχή, η ενημέρωση του λογισμικού γίνεται αυτόματα, χωρίς να χρειάζεται κάποια παραπάνω διεργασία από τον χρήστη.

Το iOS App Store είναι η πλατφόρμα που αναπτύσσει η Apple και είναι προεγκατεστημένη σε όλες τις συσκευές κινητού υπολογισμού iOS λειτουργικού συστήματος. Το λογισμικό που φιλοξενείται και είναι διαθέσιμο προς χρήση από τους κατόχους των συσκευών αυτών αναπτύσσεται μέσω του εργαλείου iOS Software Development Kit (iOS SDK) και είναι διαθέσιμο είτε δωρεάν είτε επί πληρωμή, ανάλογα με τη χρήση που γίνεται στο λογισμικό ή την εξ' αρχής αγορά της εφαρμογής για την πλήρη λειτουργία της. Το App Store, κυκλοφόρησε τον Ιούλιο του 2008, έχοντας στα αποθετήριά του μισό εκατομμύριο εφαρμογές έτοιμες προς χρήση, με το νούμερο αυτό έως τον Ιανουάριο του 2020 να έχει εκτοξευθεί στις 2.2 εκατομμύρια εφαρμογές [25].

Για τη χρήση του iOS App Store ένας προγραμματιστής ή μία εταιρεία πληρώνει ετήσια συνδρομή, η οποία ανέρχεται στα 99 δολάρια και του δίνει τη δυνατότητα να προσφέρει στους χρήστες των iOS συσκευών τις εφαρμογές που έχει στην κατοχή του και φυσικά να μπορεί να τις διαχειριστεί. Όταν κάποιο λογισμικό αγοράζεται από κάποιον χρήστη, οι διάφοροι φόροι που υπάρχουν ανά χώρα καλύπτονται από την Apple. Οι φόροι κυμαίνονται μεταξύ 2% και 4% της τιμής του λογισμικού. Η Apple αμείβεται με το 30% των εσόδων από την πώληση ενός λογισμικού μέσα από το iOS App Store και ο δημιουργός του λογισμικού παίρνει το 70% του ποσού αυτού.

Οι τρόποι με τους οποίους το λογισμικό που είναι διαθέσιμο από το iOS App Store διατίθεται στον χρήστη είναι οι παρακάτω:

1. Το λογισμικό διατίθεται δωρεάν, αλλά κατά το περιεχόμενό του χαρακτηρίζεται από μεγάλο πλήθος διαφημίσεων που αποτελούν μία μορφή εσόδων.

2. Το λογισμικό διατίθεται εν μέρει δωρεάν. Έτσι ο χρήστης έχει τη δυνατότητα δωρεάν πρόσβασης και χρήσης κάποιων υπηρεσιών της εφαρμογής, ενώ κάποιες άλλες είναι απαραίτητο να αγοραστούν με το ποσό που έχει καθοριστεί για την καθεμία. Στην περίπτωση αυτή ο χρήστης επιλέγει ποιες και πόσες υπηρεσίες θα χρησιμοποιήσει ανάλογα με τις ανάγκες και τα ενδιαφέροντα του.

3. Το πλήρες λογισμικό διατίθεται αποκλειστικά με την αγορά του.

Οι ενημερώσεις στις εφαρμογές που διατίθενται μέσω του iOS App Store πραγματοποιούνται βασισμένες στην τεχνολογία υπολογιστικού νέφους, στην ίδια

λογική με το Google Play Store, που αναλύθηκε παραπάνω. Στην περίπτωση της Apple όμως, υπάρχει μια διαφορά σε σχέση με τις υπόλοιπες εταιρείες που προσφέρουν παρόμοιες υπηρεσίες. Το λογισμικό της, δεν λειτουργεί σε όλες τις συσκευές της Apple που τρέχουν με λειτουργικό macOS. Συγκεκριμένα, με την αγορά ενός προϊόντος της Apple, παρέχεται ενημέρωση σχετικά με τον χρόνο υποστήριξης του προϊόντος και μετά πέρασ αυτής της ημερομηνίας παύει η συσκευή να ενημερώνεται από το iOS App Store και να μπορεί να κατεβάζει τις νέες εφαρμογές. Με αυτόν τον τρόπο η Apple αναγκάζει τους χρήστες να πρέπει να προβούν σε αγορά καινούργιας συσκευής [26].

Το εργαλείο Testflight χρησιμοποιείται για τον έλεγχο δοκιμαστικών (beta) εκδόσεων εφαρμογών κινητού υπολογισμού, πριν την επίσημη κυκλοφορία τους. Το Testflight χρησιμοποιείται κυρίως από ομάδες ανάπτυξης λογισμικού για προϊόντα της εταιρείας Apple. Πέραν της αυτοματοποιημένης λειτουργίας Fastlane για τον έλεγχο της δοκιμαστικής έκδοσης, ο έλεγχος μπορεί να πραγματοποιηθεί ακολουθώντας μεμονωμένα βήματα μέσω της διαδικτυακής πύλης iTunes Connect, το οποίο επιτρέπει την εισαγωγή εσωτερικών και εξωτερικών ελέγχων. Οι εσωτερικοί αξιολογητές αποτελούν μέρος της ομάδας ανάπτυξης, επομένως είναι μόνο οι προγραμματιστές ή εκείνοι που έχουν ρόλους προγραμματιστών στην ομάδα του iTunes Connect (διαχειριστές). Ο χρόνος που απαιτείται για τον έλεγχο της εφαρμογής αποδίδεται κυρίως στη δημιουργία της εφαρμογής στο iTunes Connect και την υποβολή της εφαρμογής μέσω Xcode, η οποία μπορεί να διαρκέσει προκειμένου να εγκριθεί μέσω του αυτοματοποιημένου συστήματος επιθεώρησης εφαρμογών Apple. Επιπλέον, πρέπει να δηλωθούν ρητά κάποιες πληροφορίες σχετικά με τη δοκιμαστική εφαρμογή που υποβάλλεται, όπως μια περιγραφή και τι πρέπει να δοκιμαστεί.

Το TestFairy είναι μία από τις συνιστώμενες εναλλακτικές λύσεις του TestFlight. Λειτουργεί για iOS και Android και δεκάδες άλλες πλατφόρμες και παρέχει κάτι περισσότερο από απλή διανομή εφαρμογών. Το TestFairy παρέχει βίντεο που δείχνουν τι ακριβώς συνέβη στην εφαρμογή σας προτού προκληθεί κάτι. Η υπηρεσία παρέχει στους χρήστες τη δυνατότητα να χρησιμοποιήσουν την εφαρμογή στη συσκευή τους και να στείλουν τα σχόλιά τους στον προγραμματιστή. Αυτή η ανατροφοδότηση, μπορεί να ωθηθεί αυτόματα σε Slack ή JIRA, με ένα βίντεο που δείχνει τι ακριβώς έγινε με τον χρήστη.

Το Applivery παρέχει επίσης μία από τις πιο ολοκληρωμένες λύσεις, προσφέροντας ενσωμάτωση με Slack, JIRA και Google Analytics. Επιτρέπει επίσης τη μεταφόρτωση εφαρμογών αυτόματα, απευθείας από το διακομιστή στον οποίο πραγματοποιείται η ανάπτυξη λογισμικού, που είναι έτοιμη να χρησιμοποιεί Bitrise, Jenkins, Travis-CI ή οποιαδήποτε άλλη συνεχή πλατφόρμα ολοκλήρωσης, ανάπτυξης και παράδοσης. Επιπλέον, παρέχει ακόμη και ένα plugin Fastlane το οποίο αυτοματοποιεί όπως και στο TestFlight τη διαδικασία επιθεώρησης της εφαρμογής, ακριβώς όπως και οι ανταγωνιστές του. Το συγκεκριμένο σύστημα παρέχει ένα REST API το οποίο καλύπτει ένα μεγάλο εύρος δυνατοτήτων.

Το Windows Phone Store, με προηγούμενη ονομασία Windows Phone Marketplace είναι το κατάστημα εφαρμογών που ανέπτυξε η Microsoft για τις Windows συσκευές κινητού υπολογισμού. Ξεκίνησε με το Windows Phone 7 τον Οκτώβριο του 2010. Με την ανάπτυξη του Windows Phone 7.5, η Microsoft παρουσίασε το διαδικτυακό Marketplace που προσέφερε εγκατάσταση εφαρμογών χρησιμοποιώντας τεχνολογίες νέφους. Τον Αύγουστο του 2012, η Microsoft μετονομάστηκε σε Windows Phone Store. Το 2015 η Microsoft ανακοίνωσε ότι το Windows Phone Store θα καταργηθεί σταδιακά και θα αντικατασταθεί από το Windows Store, το οποίο θα λειτουργεί ως ενοποιημένο κατάστημα για όλες τις συσκευές που λειτουργούν με Windows. Αυτή η διαδικασία αναπτύχθηκε σταδιακά με τη βοήθεια της ιστοσελίδας των Windows, μια προσωρινή λύση πριν από το ενοποιημένο κατάστημα. Το Windows Store, η πλατφόρμα ψηφιακής διανομής για τα Microsoft Windows, ξεκίνησε ως κατάστημα εφαρμογών για τα Windows 8 και Windows Server 2012, ως το κύριο μέσο διανομής των εφαρμογών Universal Platform της πλατφόρμας. Με τα Windows 10, ωστόσο, η Microsoft συγχώνευσε τις άλλες πλατφόρμες διανομής (Windows Marketplace, Windows Phone Store, Xbox Video και Xbox Music) στο Windows Store, καθιστώντας το ενιαίο σημείο διανομής για εφαρμογές, ψηφιακά δεδομένα όπως βίντεο και μουσική και ηλεκτρονικό βιβλίο. Ωστόσο, το Σεπτέμβριο του 2017, οι υπηρεσίες του ηλεκτρονικού καταστήματος μουσικής αντικαταστάθηκε από την Spotify [27].

ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα-Επεκτάσεις

Η διαρκής αύξηση των χρηστών των κινητών συσκευών ωθεί τις εταιρείες και τους προγραμματιστές για την ανάπτυξη εφαρμογών για κινητές συσκευές. Τα εκτιμώμενα κέρδη από τις εφαρμογές για κινητές συσκευές υπολογίζονται περί τα 77δισ δολάρια. Η δημιουργία native apps για τις διαφορετικές πλατφόρμες κινητών εφαρμογών έχει ως αποτέλεσμα την σπατάλη προγραμματιστικού χρόνου και είναι δαπανηρή. Η εξέλιξη των τεχνολογιών Web έρχονται να δώσουν τα απαραίτητα εργαλεία για την δημιουργία Cross-Platform εφαρμογών [28].

Με την έλευση της τέταρτης βιομηχανικής επανάστασης, μια πληθώρα από τεχνολογίες αιχμής αναδύονται με αποτέλεσμα τη δημιουργία νέων ερευνητικών ευκαιριών και προκλήσεων. Καινοτόμες τεχνολογίες στο χώρο των τηλεπικοινωνιών και κινητών εφαρμογών αποτελούν οι αρχιτεκτονικές 5G και τα μοντέλα υπολογιστικών πόρων στο άκρο (edge paradigms). Οι αρχιτεκτονικές 5G θα επιτρέψουν μεταξύ άλλων, αποδοτική επικοινωνία μεταξύ συσκευών οι οποίες βρίσκονται σε κοντινή απόσταση (device-to-device proximity services) και ενεργειακή απόδοση κατά την εκτέλεση διεργασιών υψηλής υπολογιστικής ισχύος. Στα μοντέλα υπολογιστικών πόρων στο άκρο ανήκουν τεχνολογίες όπως Mobile Edge Computing (MEC), Mobile Cloud Computing (MCC) και Fog Computing. Δεδομένου ότι στη σύγχρονη κοινωνία οι κινητές συσκευές (smartphones, tablets) έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων με πολλά πλεονεκτήματα λόγω των εφαρμογών που εκτελούνται σε αυτές, λόγω του όγκου δεδομένων που διασχίζουν το δίκτυο που διασυνδέει κινητές συσκευές και τους υπολογιστικούς πόρους, πολλές προκλήσεις δημιουργούνται. Οι προκλήσεις δημιουργούνται λόγω όγκου δεδομένων αλλά και λόγω της αυξημένης διαδικτυακής κίνησης και των πολύπλοκων πρωτοκόλλων που χρησιμοποιούνται, τα οποία απαιτούν low-latency και high-bandwidth. Διαμέσω των Mobile Backend-as-Service (MbaaS) περιβαλλόντων τα οποία παρέχουν νεφοϋπολογιστικές υπηρεσίες, οι συγκεκριμένες εφαρμογές δύναται να λειτουργήσουν αποδοτικά, ωστόσο οι προκλήσεις μετριάζονται χωρίς να επιλύονται ολοκληρωτικά και με τη πάροδο του χρόνου, νέα προβλήματα θα αναδυθούν.

Τα πλεονεκτήματα ανάπτυξης εφαρμογών με την χρήση cross platforms είναι πολλά. Κατά την άποψή μας θεωρούμε ότι η ανάπτυξη τέτοιου είδους εφαρμογών, θα μονοπωλήσουν την Βιομηχανία ανάπτυξης τα επόμενα χρόνια.

Συνοπτικά παραθέτουμε τους λόγους :

Με την ίδια υλοποίηση εξασφαλίζεται συμβατότητα με διαφορετικά λειτουργικά συστήματα

- Χρήση τεχνολογιών “open source” και εργαλείων σε σχέση με την υλοποίηση

- Δυναμικά πλαίσια ανάπτυξης όπως ionic html5 που παρουσιάζουν πολλαπλά πλεονεκτήματα

- Χρήση javascript και πλαισίων ανάπτυξης εφαρμογών, όπως το Angular.js, της εταιρείας Google Corporation.

- Χαμηλότερο λειτουργικό κόστος για τις εταιρείες ανάπτυξης , λόγω του ότι με μία υλοποίηση καλύπτονται διάφορα λειτουργικά συστήματα. Κατ’ αυτό τον τρόπο υπάρχει οικονομία κλίμακος στις επιχειρήσεις ανάπτυξης λογισμικού εφαρμογών, κάτι το οποίο λειτουργεί πολλαπλασιαστικά και για την τελική τιμή διάθεσης των εφαρμογών / υπηρεσιών στην αγορά αλλά και στην ανάπτυξη υγιών επιχειρηματικών μοντέλων με έμφαση στην εξειδίκευση και την καινοτομία.

- Οι εταιρείες πάροχοι λειτουργικών συστημάτων (Apple, Google, MicroSoft) έχουν δηλώσει την πρόθεσή τους για την ανάπτυξη του συγκεκριμένου μοντέλου και επενδύουν εκατομμύρια δολάρια για την επίτευξη του συγκεκριμένου σκοπού. Δεν είναι τυχαίο ότι η Google αναπτύσσει το πλαίσιο ανάπτυξης εφαρμογών Angular.js και δηλώνει ότι αποτελεί στρατηγική και τον πυλώνα ανάπτυξης της εταιρείας τα επόμενα χρόνια.

- Το κόστος των κινητών συσκευών έχει αρχίσει και φθίνει, με αποτέλεσμα την μεγαλύτερη διείσδυση στην αγορά αλλά και την αντικατάσταση των φορητών υπολογιστών από κινητές συσκευές (μεγάλες οθόνες με υψηλή ανάλυση, μεγάλη διάρκεια ζωής μπαταρίας, εξελιγμένοι επεξεργαστές, κ.α.)

Είναι κατανοητό ότι οι υβριδικές εφαρμογές αποτελούν το μοντέλο ανάπτυξης των εφαρμογών σε κινητές συσκευές. Αυτό θα είναι και το σκηνικό τα επόμενα χρόνια εισάγοντας την χρήση του διαδικτύου πραγμάτων (internet of things) το οποίο απαιτεί εκτός των άλλων διαχείριση μεγάλων ποσοτήτων δεδομένων (big data) αλλά και εξειδικευμένες καινοτόμες εφαρμογές κινητών συσκευών, κάτι το οποίο επιτυγχάνεται με την χρήση cross platforms [29].

Περαιτέρω έρευνα μπορεί να επικεντρωθεί στην διαχείριση μεγάλων ποσοτήτων δεδομένων (big data) από υβριδικές εφαρμογές, καθώς και στην αρχιτεκτονική ωθήσεων ειδοποιήσεων (push notifications).

Μέχρι το τέλος του 2020 εκτιμάται ότι το 84% του παγκόσμιου πληθυσμού θα χρησιμοποιεί κινητή τεχνολογία. Η ανάπτυξη και η δημιουργία των κατάλληλων επιχειρηματικών εφαρμογών μπορεί να βοηθήσει τις εταιρείες να γίνουν πιο αποδοτικές και κερδοφόρες.

Ωστόσο, η κρίση δεξιοτήτων που μαστίζει την ανάπτυξη εφαρμογών για smartphones θα μπορούσε να θέσει σε κίνδυνο αυτόν τον πολύπαθο κλάδο, καθώς η ζήτηση ξεπερνά τις δυνατότητες σε πολλές επιχειρήσεις που δραστηριοποιείται στο κομμάτι αυτό.

Μία από τις πιο δυναμικές εταιρείες στον χώρο της πληροφορικής, η γνωστή Robert Half, τόνισε ότι η αυξημένη ζήτηση για smartphone apps έχουν ωθήσει τις εταιρείες να βρουν υποψηφίους με δεξιότητες ανάπτυξης εφαρμογών για κινητά. Λέει ότι οι υποψήφιοι με εξαιρετικό ταλέντο είναι σπάνιοι εξαιτίας της μεγάλης ποικιλίας των απαιτούμενων δεξιοτήτων, όχι μόνο των τεχνικών δεξιοτήτων, αλλά και της επιχειρηματικής έμπνευσης, διότι δεν πρέπει να ξεχνάμε πως ο εργαζόμενος είναι ένα πολυεργαλείο που όταν δουλεύει αποδοτικά φέρνει χρήματα στις επιχειρήσεις [30].

Είναι δεδομένο ότι οι προγραμματιστές εφαρμογών για κινητά πρέπει να έχουν εμπειρία στην ανάπτυξη εφαρμογών σε όλες τις μεγάλες πλατφόρμες (iOS, Android, Windows Phone), εμπειρία κωδικοποίησης με Mobile Frameworks όπως Xcode, Android SDK και / ή Windows Phone App Studio και εμπειρία σε γλώσσες προγραμματισμού ανάπτυξης ιστοσελίδων και τα πλαίσια εφαρμογών για κινητά.

Αλλά η διαχείριση του έργου και οι δεξιότητες επικοινωνίας είναι επίσης απαραίτητες. Οι προγραμματιστές πρέπει να συνεργάζονται και να επικοινωνούν με συναδέλφους και τους ηγέτες στην επιχείρηση. Αυτός ο συνδυασμός δεξιοτήτων αποδεικνύεται αδιευκρίνιστος για πολλές επιχειρήσεις και, μαζί με τις άλλες προκλήσεις που σχετίζονται με την ανάπτυξη επιχειρηματικών εφαρμογών, πολλές εταιρείες έχουν αναβάλει μαζικές παραγγελίες με το φόβο πως δεν θα ανταπεξέλθουν και θα εκτεθούν ανεπανόρθωτα [31].

Αλλά το μέγα ερώτημα είναι ποιος θα πρέπει να φτιάξει αυτές της ζωτικής σημασίας εφαρμογές που κάνουν χρήση οι πελάτες και ενσωματώνουν την υπάρχουσα σειρά επιχειρηματικών κινήσεων, στέλνοντας την ένα βήμα πιο μπροστά.

Μια επιλογή είναι να γίνει πρόσληψη εξωτερικών προγραμματιστών, αλλά αυτό μπορεί να είναι δαπανηρό και μπορεί να είναι δύσκολο για αυτούς τους ανθρώπους να κατανοήσουν πραγματικά την επιχείρηση από μέσα με τον ίδιο τρόπο όπως μια εσωτερική ομάδα.

Μια πιο αποδοτική λύση για την αντιμετώπιση της έλλειψης δεξιοτήτων που δεν χρησιμοποιούν εργαλεία ανάπτυξης περίπλοκου κώδικα είναι οι πλατφόρμες ανάπτυξης εφαρμογών ταχείας δημιουργίας που απευθύνονται για επαγγελματικές λύσεις. Μπορούν να μετατρέψουν τη διαδικασία ανάπτυξης εφαρμογών και να επιτρέψουν στις επιχειρήσεις να επωφεληθούν από την έκρηξη της ζήτησης για εφαρμογές.

Χρησιμοποιώντας το εσωτερικό προσωπικό ή τους γνωστούς σας ως «προγραμματιστές», αυτές οι πλατφόρμες επιτρέπουν στις εταιρείες να ξεπεράσουν το χάσμα δεξιοτήτων ανάπτυξης για smartphone apps και να επιτρέψουν σε μη προγραμματιστές να δημιουργήσουν εφαρμογές για κινητά γρήγορα και εύκολα.

Τα εργαλεία ανάπτυξης χαμηλού κώδικα επιτρέπουν στις επιχειρήσεις να χρησιμοποιούν το υπάρχον προσωπικό τους, παρά τους εκπαιδευμένους προγραμματιστές, για να δημιουργούν εξελιγμένες και λειτουργικές εφαρμογές που μπορούν να αυτοματοποιήσουν τις επιχειρηματικές διαδικασίες και να επιτρέψουν στις εταιρείες να ξεκινήσουν ψηφιακές υπηρεσίες αμέσως. Η έλλειψη εσωτερικών δεξιοτήτων ανάπτυξης εφαρμογών για κινητά δεν αποτελεί πλέον εμπόδιο για το mobile app development [33].

Κάθε εταιρεία ανάπτυξης mobile app στοχεύει στο να είναι προσιτή, ενδιαφέρουσα και user-friendly. Οι developers των mobile apps χρησιμοποιούν πολλές τεχνολογίες που μπορούν να χρησιμοποιηθούν ευρέως είτε για μία συγκεκριμένη πλατφόρμα είτε για ανάπτυξη πολλών εφαρμογών. Οι κύριες γλώσσες προγραμματισμού που χρησιμοποιούνται από τους developers είναι C ++, Swift, Java, PHP και HTML5. Αυτές οι γλώσσες υπολογιστών και προγραμματισμού έχουν βελτιώσει την εμπειρία των developer.

Στα μελλοντικά επαγγέλματα θα είναι απαραίτητες κάποιες κορυφαίες τεχνολογίες, για αυτό το λόγο οι εταιρίες mobile app εργάζονται πάνω σε αυτές τις εξελίξεις.

Μία τέτοια τεχνολογία είναι η τεχνητή νοημοσύνη. Η Τεχνητή Νοημοσύνη είναι η προσομοίωση της ανθρώπινης νοημοσύνης που επεξεργάζεται μηχανές. Αυτές οι διαδικασίες περιλαμβάνουν τρία σημαντικά βήματα που είναι: μάθηση, λογική και διόρθωση. Επίσης, αποτελείται από διάφορους τύπους δεδομένων που έχουν ευθυγραμμισμένους αλγόριθμους που περιλαμβάνουν ορισμένα υποσύνολα. Αυτά τα υποσύνολα περιλαμβάνουν την επιστήμη των δεδομένων, τη μηχανική μάθηση και τη βαθιά εκμάθηση. Σήμερα, η ΑΙ έχει γίνει αναπόσπαστο μέρος των περισσότερων

εταιρειών ανάπτυξης mobile app. Εκτός από τα χαρακτηριστικά γνωρίσματα όπως η εκμάθηση της οθόνης, η λογική και ο σχεδιασμός, το ΑΙ παρέχει επίσης δυνατότητες όπως η αντίληψη και η δυνατότητα χειρισμού.

Όπως αναφέρουν οι ειδικοί, οι αλγόριθμοι τεχνητής νοημοσύνης βρίσκονται σχεδόν παντού γύρω μας: από την αγορά ενός προϊόντος, την παρακολούθηση μίας ταινίας και τους ψηφιακούς προσωπικούς βοηθούς, μέχρι τα ρομπότ, τα αυτόνομα οχήματα και τις διαγνώσεις ασθενειών και παθήσεων.

Ειδικότερα όπως αναφέρεται στη Λευκή Βίβλο "Mobile AI and the Future of Intelligent Devices" που συνέταξε η εταιρία αναλύσεων International Data Corporation (IDC) και χρηματοδότησε η Huawei, η τεχνητή νοημοσύνη "θα γίνει ένα σύμβολο, όπως ακριβώς ήταν και η ατμομηχανή στην έναρξη της Βιομηχανικής Επανάστασης". Η πλειοψηφία του κόσμου θα είναι χρήστες έξυπνων κινητών (smartphone). Πέραν κλάδων όπως οι κατασκευές, το λιανεμπόριο και η υγεία, η τεχνητή νοημοσύνη θα επηρεάσει σημαντικά και το μέλλον των προσωπικών υπολογιστών. Σύμφωνα με τη Λευκή Βίβλο της IDC, η πλατφόρμα που θα επωφεληθεί περισσότερο από τη διείσδυση της τεχνητής νοημοσύνης είναι τα έξυπνα κινητά (smartphones). Αναλυτικότερα οι εκτιμήσεις της IDC θέλουν τους χρήστες smartphones να αποτελούν το 2021 το 60% του παγκόσμιου πληθυσμού.

Η νέα γενιά εφαρμογών και υπηρεσιών που βρίσκεται στα σκαριά, θα προκύψει με την εκμετάλλευση των δυνατοτήτων της τεχνητής νοημοσύνης. Οι εφαρμογές τεχνητής νοημοσύνης θα επηρεάσουν πληθώρα τομέων. Οι αλλαγές στο λιανεμπόριο θα είναι σημαντικές. Ένας πιθανός συνδυασμός συστημάτων, λογισμικού αναγνώρισης προσώπου και μηχανικής μάθησης (machine learning) που προκύπτει βάσει τεχνητής νοημοσύνης, θα μπορούσε να δώσει τη δυνατότητα στα καταστήματα να αναγνωρίζουν τους πελάτες τους, να τους προσφέρουν συμβουλές και ιδέες κατά τις αγορές τους και να τους διευκολύνουν κατά την πληρωμή και την παραλαβή των παραγγελιών τους με πλήρως αυτοματοποιημένες, αυτόνομες λειτουργίες χωρίς την παραμικρή ανθρώπινη παρέμβαση. Η νέα αυτή γενιά εφαρμογών θα είναι σε θέση να δώσει πνοή στο ηλεκτρονικό εμπόριο, ενισχύοντας εντυπωσιακά την εμπειρία του καταναλωτή.

Στον τομέα της υγείας, η τεχνητή νοημοσύνη σε συνδυασμό με τη μηχανική μάθηση μπορεί να βοηθήσει τους γιατρούς στην ανάλυση μεγάλου όγκου δεδομένων

όπως π.χ. το ιατρικό ιστορικό ενός ασθενούς, επιτρέποντας έτσι ακριβέστερες διαγνώσεις με μηδενικό ποσοστό σφάλματος και ακόμα μικρότερο ρίσκο.

Έτσι, θα οδηγηθούμε σε συστήματα παρακολούθησης υγείας που θα είναι σε θέση όχι απλά να καταγράφουν δεδομένα όπως για παράδειγμα τα βήματα κάποιου ημερησίως αλλά να προειδοποιούν για ενδεχόμενους κινδύνους όπως μυϊκούς τραυματισμούς. Η τεχνητή νοημοσύνη θα βελτιώσει σημαντικά τις εφαρμογές προσωπικών βοηθών, ενισχύοντας έτσι τη συνολική εμπειρία χρήσης των smartphones. Οι ψηφιακοί προσωπικοί βοηθοί θα έχουν τη δυνατότητα, επί παραδείγματι, να αναγνωρίζουν τον τίτλο της ταινίας που φαίνεται σε μία φωτογραφία μίας υπαίθριας διαφήμισης, να βρίσκουν τους κινηματογράφους που παίζεται και να κλείνουν εισιτήρια βάσει του ημερολογίου του χρήστη, ζητώντας από τον τελευταίο απλά μία έγκριση [34].

Μία ακόμα τέτοια τεχνολογία είναι η εφαρμογή Cloud. Η εφαρμογή Cloud είναι ένα software πρόγραμμα που είναι πλήρως cloud-based και επιτρέπει να εργάζονται πολλά άτομα ταυτόχρονα. Η εφαρμογή Cloud όχι μόνο εξοικονομεί χρόνο και κόστος αλλά επίσης βελτιώνει την ανάπτυξη του προϊόντος.

Επιπλέον, το Internet of Things αναφέρεται στη διασύνδεση καθημερινών αντικειμένων μέσω του internet. Το πλεονέκτημα του IoT είναι ότι δεν απαιτεί καμία φυσική αλληλεπίδραση. Στη σημερινή εποχή, το Internet γίνεται ολοένα και πιο δημοφιλές και απαιτείται από διάφορες βιομηχανίες όπως το βιομηχανικό Internet, ο αυτοματισμός στο σπίτι, η υγειονομική περίθαλψη, η γεωργία κ.λπ. Δεδομένου ότι το IoT συνδέει τα πάντα σε ένα ενιαίο σύστημα, καθίσταται ευκολότερο για τις κινητές εφαρμογές να αναπτύσσονται, να λειτουργούν και να παρέχουν ακόμα και πολύπλοκες υπηρεσίες.

Μία ακόμα τεχνολογία είναι η Blockchain που είναι μια λίστα με blocks ή records και συνδέονται μεταξύ τους μέσω κρυπτογράφησης. Η τεχνολογία Blockchain έκανε πολλές εφαρμογές πολύ πιο εύκολες, όπως supply chain, digital currency transaction, Blockchain consultant κ.λπ. Επίσης μειώνει τους διαμεσολαβητές, διαφυλάσσει τη διαφάνεια και διασφαλίζει τα δεδομένα των χρηστών, λόγω των οποίων χρησιμοποιείται σε διάφορους τομείς όπως η ψηφοφορία, η ανταλλαγή νομισμάτων, οι τράπεζες, η διαχείριση συμβάσεων κ.λπ.

Τέλος, η τεχνολογία AR ή Augmented Reality προσθέτει ψηφιακά στοιχεία σε μια ζωντανή προβολή, χρησιμοποιώντας κυρίως μια κάμερα ή ένα smartphone, σε αντίθεση με το VR ή την εικονική πραγματικότητα που δημιουργούν μια εμπειρία

που σας κλείνει τελείως από τον εξωτερικό κόσμο. Η τεχνολογία AR / VR χρησιμοποιείται ευρέως σε διάφορες εφαρμογές όπως η εκπαίδευση, η μηχανική, η ιατρική, η αρχιτεκτονική κλπ. Δεδομένου ότι αυτές οι πραγματικότητες εμπλέκουν τους χρήστες τους μέσω φορητών τεχνολογιών και εφαρμογών, αυτές οι τεχνολογίες θα βρίσκονται σε μεγάλη ζήτηση σύντομα. Δεδομένου ότι η τεχνολογία μετατοπίζεται στη νέα εποχή του 4D κόσμου, οι τεχνολογίες AR και VR κατάφεραν να συγκεντρώσουν την προσοχή όχι μόνο των εταιρειών ανάπτυξης εφαρμογών για κινητά, αλλά και των επιχειρηματιών και των επενδυτών.

Συμπερασματικά, μπορεί να ειπωθεί ότι όλες αυτές οι τεχνολογίες χρησιμοποιούνται ευρέως από τους developers των mobile app για να απομακρύνουν την πολυπλοκότητα του συστήματός τους και να παρέχουν μια καλύτερη εμπειρία στους χρήστες τους.

Βιβλιογραφία-Αναφορές

- [1] Wikipedia, “Mobile App”, [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Mobile_app [Πρόσβαση Οκτώβριος 2019].
- [2] Ken Yarmosh (2016), “How Often Should You Update Your App?”, [Ηλεκτρονικό]. Available: <https://savvyapps.com/blog/how-often-should-you-update-your-app> [Πρόσβαση Οκτώβριος 2019].
- [3] Ian Blair, Mobile App Development Blog, “10 Best Cross Platform Mobile Development Tools”, [Ηλεκτρονικό]. Available: <https://buildfire.com/best-mobile-cross-platformdevelopment-tools/> [Πρόσβαση Φεβρουάριος 2020].
- [4] “Apache Cordova Artwork”, [Ηλεκτρονικό]. Available: <https://cordova.apache.org/artwork/> [Πρόσβαση Φεβρουάριος 2020].
- [5] Open Source Guide, “PhoneGap/Cordova”, [Ηλεκτρονικό]. Available: <http://www.opensource-guide.com/en/Solutions/Development-and-intermediate-layers/Cross-platformmobile-frameworks/Phonogap-cordova> [Πρόσβαση Φεβρουάριος 2020].
- [6] Apache Cordova, “Architectural overview of Cordova platform”, [Ηλεκτρονικό]. Available: <https://cordova.apache.org/docs/en/latest/guide/overview/> [Πρόσβαση Φεβρουάριος 2020].
- [7] UniScoop, “Appcelerator Titanium Review”, [Ηλεκτρονικό]. Available: <http://uniscoop.com/good-bad-appcelerator-titanium/> [Πρόσβαση Φεβρουάριος 2020].
- [8] Appcelerator Titanium Tutorial (2018), [Ηλεκτρονικό]. Available: http://randikastitaniumtutorial.blogspot.gr/2013/04/introduction-to-appcelerator-titanium_10.html [Πρόσβαση Φεβρουάριος 2020].
- [9] Appcelerator Wiki (2017), “Titanium Mobile Overview – Documentations and Guides”, [Ηλεκτρονικό]. Available: <https://wiki.appcelerator.org/pages/viewpage.action?pageId=27164787> [Πρόσβαση Φεβρουάριος 2020].
- [10] Mobile World Live (2019), “Microsoft linked to dev tool provider Xamarin”, [Ηλεκτρονικό]. Available: <https://www.mobileworldlive.com/apps/news-apps/microsoftlinked-tool-provider-xamarin-reports/> [Πρόσβαση Φεβρουάριος 2020].
- [11] Business Wire (2017), “Xamarin and Samsung Collaborate to Bring Advanced Security to Enterprise Applications”, [Ηλεκτρονικό]. Available: <https://www.businesswire.com/news/home/20140317005266/en/Xamarin->

SamsungCollaborate-Bring-Advanced-Security-Enterprise [Πρόσβαση Φεβρουάριος 2020].

[12] Cygnet Infotech (2015), “PhoneGap or Titanium or Xamarin - Which Cross-Platform Framework Should You Choose?”, [Ηλεκτρονικό]. Available: <http://www.cygnetinfotech.com/blog/phonegap-or-titanium-or-xamarin-which-cross-platform-should-youchoose> [Πρόσβαση Φεβρουάριος 2020].

[13] Cogito Solutions, “Sencha Ext JS”, [Ηλεκτρονικό]. Available: <http://www.cogitosolutions.com/hk/node/434> [Πρόσβαση Φεβρουάριος 2020].

[14] Ext JS – JavaScript Framework for web apps, “Sencha Ext JS”, [Ηλεκτρονικό]. Available: <https://www.sencha.com/products/extjs/> [Πρόσβαση Φεβρουάριος 2020].

[15] Learning Corona SDK with the DevilSquid, “Corona Level Editor”, [Ηλεκτρονικό]. Available: <https://devilsquid.com/corona-level-editor/> [Πρόσβαση Φεβρουάριος 2018].

[16] Wikipedia, “Corona (software)”, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Corona_\(software\)](https://en.wikipedia.org/wiki/Corona_(software)) [Πρόσβαση Φεβρουάριος 2020].

[17] AppFutura (2018), “Ionic Framework: free SDK for developing hybrid mobile apps”, [Ηλεκτρονικό]. Available: <https://www.appfutura.com/blog/ionic-framework-free-sdkfor-developing-hybrid-mobile-apps/> [Πρόσβαση Φεβρουάριος 2020].

[18] Ionic Framework, [Ηλεκτρονικό]. Available: <https://forum.ionicframework.com/t/isthere-a-graphic-diagram-available-that-shows-how-ionic-2-and-all-related-technologiesconnect/88375> [Πρόσβαση Φεβρουάριος 2018].

[19] Wikipedia, “Ionic (mobile app framework)”, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)) [Πρόσβαση Φεβρουάριος 2020].

[20] The Qt Company, [Ηλεκτρονικό]. Available: <https://www1.qt.io/company/> [Πρόσβαση Φεβρουάριος 2020].

[21] Qt, “Cross-platform software development for embedded and Desktop”, [Ηλεκτρονικό]. Available: <https://www.qt.io/> [Πρόσβαση Φεβρουάριος 2020].

[22] Business Wire (2014), “Qt Creator IDE”, [Ηλεκτρονικό]. Available: <https://www.businesswire.com/news/home/20140225005937/en/Digia-announces-majorupdate-Qt-Enterprise-Embedded> [Πρόσβαση Φεβρουάριος 2020].

[23] Κοντόπουλος Ιάσων, Ματσαρίδης Παναγιώτης, (2018), “Cloud Computing”, Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών Πολυτεχνική Σχολή Κοζάνης Πανεπιστήμιο Δυτικής Μακεδονίας [Ηλεκτρονικό]. Διαθέσιμο:

- https://arch.icte.uowm.gr/docs/CloudComputing_Parallel_Distributed_Dasygenis_Matsari_dhs_Kontopoulous.pdf [Πρόσβαση Φεβρουάριος 2020].
- [24] Sarah Jacobsson Purewal (2015), “How to free up Google Drive space”, [Ηλεκτρονικό]. Available: <https://www.cnet.com/how-to/how-to-free-up-google-drive-space/> [Πρόσβαση Φεβρουάριος 2020].
- [25] Microsoft OneDrive Plans, [Ηλεκτρονικό]. Available: <https://onedrive.live.com/about/enUS/plans/> [Πρόσβαση Φεβρουάριος 2020].
- [26] Dropbox, [Ηλεκτρονικό]. Available: <https://www.dropbox.com/help/billing/cost> [Πρόσβαση Φεβρουάριος 2020].
- [27] Cloud Storage Security Comparison (2019), [Ηλεκτρονικό]. Available: <https://tresorit.com/cloud-storage-comparison> [Πρόσβαση Φεβρουάριος 2020].
- [28] The Balance, “Review of Cloud Storage Service MEGA”, [Ηλεκτρονικό]. Available: <https://www.thebalance.com/mega-review-1356576> [Πρόσβαση Φεβρουάριος 2020].
- [29] Ευαγγελία Κοκκίνου (2019), “Εφαρμογές νέφους (cloud computing)”, [Ηλεκτρονικό]. Available: <https://www.slideshare.net/kokkinoulitsa/cloud-computing-71824176> [Πρόσβαση Φεβρουάριος 2018].
- [30] Wikipedia, “Cloud computing”, [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Cloud_computing [Πρόσβαση Φεβρουάριος 2020].
- [31] Shshil B., Leena J., Sandeep J.; (2010); "Cloud Computing: A study of infrastructure as a service (IaaS)"; International Journal of Engineering and Information Technology; Waves publishers [Πρόσβαση Φεβρουάριος 2020].
- [32] Krishnan Subramanian (2013), “MBaaS and PaaS: Different Dance On The Same Data”, [Ηλεκτρονικό]. Available: <http://allthingsplatforms.com/platforms/mbaas-and-paasdifferent-dance-on-the-same-data/> [Πρόσβαση Φεβρουάριος 2020].
- [33] Cristian Spoiala (2015), “Cloud offering: Comparison between IaaS, PaaS, SaaS, BaaS”, [Ηλεκτρονικό]. Available: <https://assist-software.net/blog/cloud-offering-comparisonbetween-iaas-paas-saas-baas> [Πρόσβαση Φεβρουάριος 2020].