

# Stato dell'arte

---

Versione **0.1**  
Data 15 December 2011  
Autori **Rino Goller, Maurizio Barazzuol, Katiuscia Reversi, Carlo Mirabassi, Cristiano Carlevaro.**

Abstract In questo capitolo viene trattata un'analisi dei sistemi operativi atta a descrivere le tecnologie disponibili e di quelle che poi sono state utilizzate per la realizzazione del progetto.

---

## Indice

INTRODUZIONE.....	4
APPLICAZIONI E PIATTAFORME MOBILI.....	5
PANORAMICA SISTEMI OPERATIVI MOBILI.....	6
SYMBIAN.....	7
Versioni di Symbian:.....	8
PARTICOLARITÀ.....	8
APPLICAZIONI.....	9
IOS.....	10
iPhone OS 3.....	10
OS 3.1.....	11
iOS 4.....	11
ANDROID.....	13
WINDOWS MOBILE.....	15
Servizi di comunicazione e networking.....	16
Moduli e Componenti di Windows CE.....	16
PALM OS.....	18
RESEARCH IN MOTION (RIM).....	20
BlackBerry Enterprise Server.....	20
DISPOSITIVI DI COMUNICAZIONE MOBILE.....	22
ANALISI DELLE TECNOLOGIE DISPONIBILI.....	23

---

<b>Web Service.....</b>	<b>23</b>
<b>Touch Screen.....</b>	<b>23</b>
<b>3G.....</b>	<b>23</b>
<b>Wi-Fi.....</b>	<b>24</b>
<b>Java.....</b>	<b>24</b>
<b>JavaFX.....</b>	<b>24</b>
<b>SDK.....</b>	<b>25</b>
<b>JSP.....</b>	<b>25</b>
<b>XML.....</b>	<b>26</b>
<b>CLIENT SIDE.....</b>	<b>27</b>
<b>SERVER SIDE.....</b>	<b>28</b>
<b>COMUNICAZIONE CLIENT-SERVER.....</b>	<b>29</b>
<b>JSON.....</b>	<b>29</b>
<b>SISTEMI LEGACY CON CUI INTEGRARSI.....</b>	<b>30</b>
<b>SCELTA PRELIMINARE DI TECNOLOGIE.....</b>	<b>31</b>
<b>CONCLUSIONE.....</b>	<b>32</b>

## Introduzione

Oggi giorno l'utilizzo del proprio cellulare per usufruire di servizi di vario tipo è attività sempre più comune. Nello specifico la possibilità di disporre di una piattaforma in grado di supportare applicazioni mirate a semplificare la vita quotidiana degli utenti è molto interessante.

Un framework ed un sistema software di base sul quale appoggiarsi e tramite il quale distribuire servizi potrebbe essere una soluzione.

Il target alla base di questo lavoro è quello di sviluppare un sistema che eroghi servizi web che permettano all'utilizzatore finale di non perdere tempo prezioso. La nostra idea è quella di creare una piattaforma dalla quale erogare i nostri servizi.

L'idea è semplice: fare in modo che qualsiasi utente in possesso di uno smartphone sia in grado di connettersi alla nostra piattaforma sulla quale potrà trovare diversi servizi.

La nostra piattaforma consiste in un sistema software su cui sono sviluppati ed eseguiti i nostri servizi.

L'analisi dello stato dell'arte verrà suddivisa in varie parti:

- panoramica sui sistemi operativi mobili con riferimento alle analisi di mercato.
- Descrizione dei vari sistemi operativi: Symbian, iOS, Android, Windows Mobile, Palm OS, RIM Blackberry, con un'analisi dei pro e dei contro in relazione all'obiettivo del progetto.
- Analisi delle tecnologie disponibili.
- Scelta preliminare delle tecnologie

## Applicazioni e piattaforme mobili

Il mercato dei sistemi operativi per i dispositivi mobili non è particolarmente vasto, ma è particolarmente florido in termini di applicazioni e linguaggi supportati. Ciò implica una concorrenza spietata essenzialmente mirata a fornire all'utente un dispositivo che sia il più possibile user-friendly, che permetta una fruizione immediata di un gran numero di applicazioni.

I sistemi operativi attualmente disponibili sono: Symbian (Symbian Foundation), Research In Motion (RIM, Blackberry), iOS (Apple), Windows CE (Microsoft), Android (Google), Palm (Palm Inc.).

Company	Units (2010)	Market Share % (2010)	Units (2009)	Market Share % (2009)
Symbian	111576,7	37,6	80878,3	46,9
R.I.M.	47451,6	16	34346,6	19,9
iOS	46598,3	15,7	24889,7	14,4
Microsoft	12378,2	4,2	15031	8,7
Android	67224,5	22,7	6798,4	3,9
Other Oss	11417,4	3,8	10432,1	6,1
<b>Total</b>	296646,6	100	172376,1	100

Le ultime statistiche indicano un calo vertiginoso del monopolio di Sybian a favore di RIM e iOS, che con il lancio di iPhone ha iniziato ad accaparrarsi sempre più punti percentuali dell'intero mercato.

Anche Google stando alle ultime statistiche ha iniziato ad espandersi sempre di più grazie al suo sistema operativo: Android. Dai dati presenti in tabella possiamo vedere come l'utilizzo di Android sia aumentato passando da pochi punti percentuali a poco meno di un quarto del mercato mondiale.

Successivamente verranno analizzate le diverse piattaforme con l'intento di evidenziare pro e contro di ogni sistema operativo in relazione all'obiettivo del progetto. Verranno presentate anche delle immagini esplicative delle architetture dei sistemi operativi analizzati reperite on line sui siti dei produttori e degli sviluppatori.

Lo scopo di questa analisi è cercare di capire quali sistemi operativi sono più adatti alla realizzazione del nostro progetto, evidenziando le caratteristiche sia a livello di architettura, sia a livello di componentistica hardware tali da poter permettere uno sviluppo che sia realmente multiplatforma e per trovare un compromesso, in termini di design e programmazione, per poter garantire una scalabilità e compatibilità il più ampie possibili.

Considerando il documento relativo ai requisiti per lo sviluppo del progetto si può arrivare a definire una serie di punti cardine che non sono altro che una serie di funzionalità che il dispositivo utente deve avere.

Esse sono: capacità di creare una connessione ad internet o, in alternativa, la predisposizione a connettersi ad una rete WiFi (meglio se entrambe), uno schermo di di dimensioni non inferiori ai 150x150px e un browser nel quale sia stato attivato il javascript.

Di seguito sono stati elencati i sistemi operativi che abbiamo considerato e analizzato come potenziali piattaforme per le quali sviluppare la nostra applicazione.

<b>SISTEMI OPERATIVI</b>	<b>VERSIONI</b>
SYMBIAN PLATFORM	<ul style="list-style-type: none"> <li>• Symbian^2</li> <li>• Symbian^3</li> <li>• Symbian^4</li> </ul>
RESEARCH IN MOTION	<ul style="list-style-type: none"> <li>• 4.5 Branch [4.5.0]</li> <li>• 4.6 Branch [4.6.0 – 4.6.1]</li> <li>• 4.7 Branch [4.7.0 – 4.7.1]</li> </ul>
APPLE iOS	<ul style="list-style-type: none"> <li>• 3.0</li> <li>• 3.1</li> <li>• 3.2</li> <li>• 4.0</li> </ul>
MICROSOFT WINDOWS MOBILE	<ul style="list-style-type: none"> <li>• Windows Mobile</li> <li>• Windows Phone 7</li> </ul>
GOOGLE ANDROID	<ul style="list-style-type: none"> <li>• 1.5</li> <li>• 1.6</li> <li>• 2.0</li> <li>• 2.1</li> <li>• 3.0</li> </ul>
PALM OS	<ul style="list-style-type: none"> <li>• 1.0 Branch [1.0.2 – 1.0.3 – 1.0.4] <ul style="list-style-type: none"> <li>• 1.1 Branch [1.1.0]</li> </ul> </li> <li>• 1.2 Branch [1.2.0 – 1.2.1]</li> <li>• 1.3 Branch [1.3.1 – 1.3.5 – 1.3.5.1]</li> <li>• 1.4 Branch [1.4.0 – 1.4.1.1]</li> </ul>

## Symbian

È uno dei più diffusi sistemi operativi. Viene montato di serie sui telefoni Nokia.

Nell'analisi seguente vediamo le particolarità di questo sistema operativo:

- Multithreading
- Multitasking
- Supporto EDGE, 3G, data streaming, DVB-H, Open GL ES
- Supporto Ipv6
- Supporto Bluetooth 2.0
- Rete Wi-Fi
- Funzionalità VoIP
- Touchscreen
- Acceleratore grafico 3D
- Supporto per la visualizzazione e la modifica di file di testo, fogli di calcolo, presentazioni e file pdf.

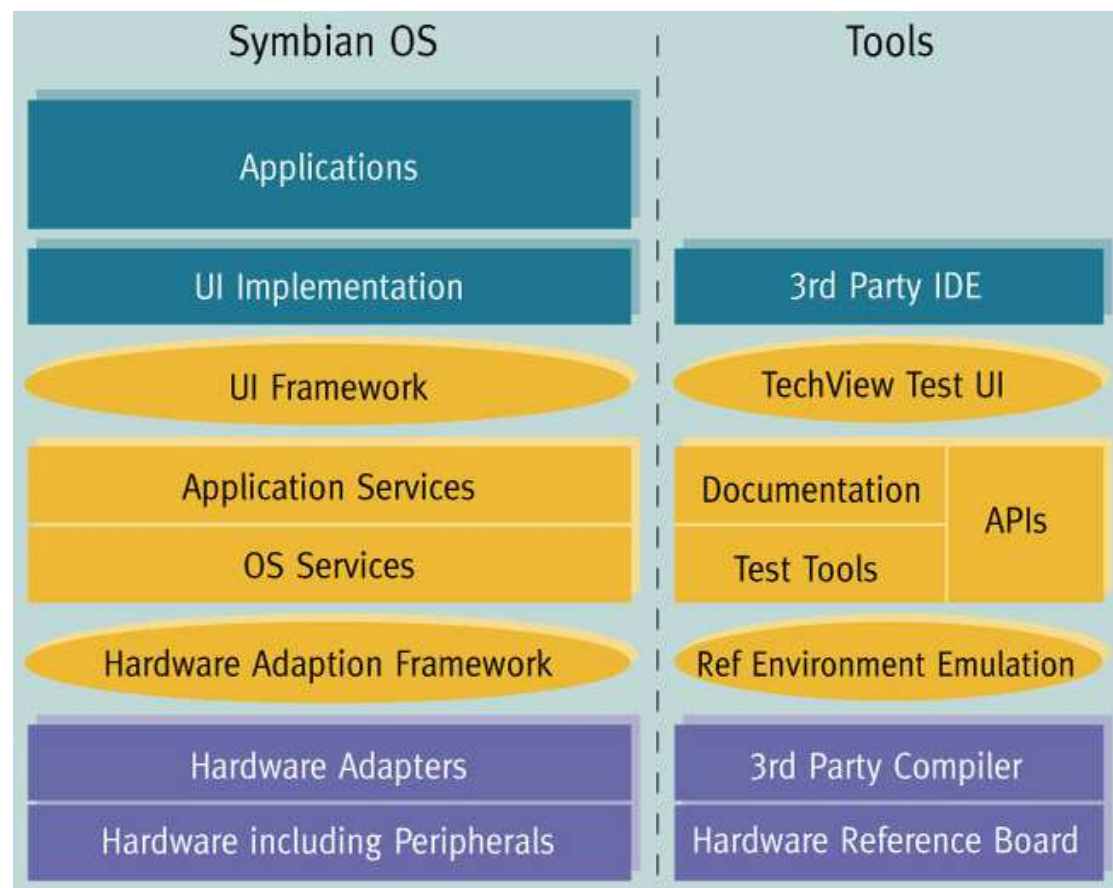


Illustrazione 1: Architettura Symbian – <http://www.mokabyte.it>

## iOS

E' il sistema operativo di Apple. Per creare un applicativo per questa piattaforma è necessario attenersi ad una serie di passi base senza contare il fatto che è necessario scrivere le applicazioni in un linguaggio che Apple ha creato ad hoc per le sue app, e cioè Xcode.

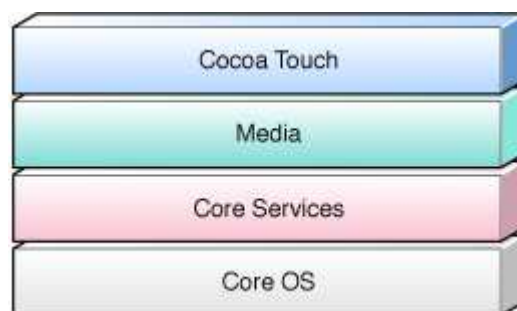
Questo tipo di linguaggio si rifà alla programmazione ad oggetti, ma per poter sviluppare con l'SDK apposita è necessario dotarsi di un sistema Apple di ultima generazione (Mac OS Leopard o Snow Leopard) e acquistare le licenze necessarie.

Successivamente lo sviluppatore deve inviare la propria App in maniera tale da farla valutare come idonea o meno per l'App Store.

Nel caso in cui il team di valutazione dia esito positivo allora nell'App Store comparirà una nuova applicazione da scaricare.

Nonostante i suoi innegabili pregi in termini di velocità e potenza è stato necessario, a fronte dell'analisi qui sopra, escludere tutto ciò che riguardava iPhone, iPad, iPod Touch, ecc.. poiché per lo sviluppo di applicazioni per questi dispositivi è necessario dotarsi di un ambiente di sviluppo particolare e soprattutto non free come invece è Netbeans.

Il progetto che abbiamo realizzato però è pensato per essere visualizzato ed utilizzato anche su iPhone e iPad. Infatti è sufficiente che nel browser di default (Safari) venga abilitato (qualora sia possibile disabilitarlo) il codice javascript. In questa maniera utilizzando una web app siamo riusciti ad aggirare i costi e le limitazioni imposte da Apple per la creazione e la distribuzione di servizi sulle loro piattaforme.



*Illustrazione 2: Architettura Kernel iOS*  
– <http://developer.apple.com>



## Android

É un sistema operativo open source per dispositivi mobili basato sul sistema operativo linux.

Usa il database SQLite, la libreria dedicata SGL per la grafica bidimensionale e supporta lo standard OpenGL ES 2.0 per la grafica tridimensionale. Le applicazioni vengono eseguite tramite la Dalvik virtual machine, una Java Virtual Machine adattata per l'uso su dispositivi mobili. Android è fornito di una serie di applicazioni preinstallate: un browser basato su Webkit, una rubrica e un calendario.

Considerando le caratteristiche di questo sistema operativo possiamo notare come sia compatibile con molte delle tecnologie attualmente presenti:

- **Adattabilità:** La piattaforma è adattabile a tutte le maggiori librerie grafiche VGA in 2D e le librerie grafiche in 3D basate su specifiche OpenGL ES 1.0.
- **Storage:** Il software per database SQLite è usato per lo storage dei dati.
- **Connettività:** Android supporta una vasta gamma di tecnologie per la connettività, come GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, e Wi-Fi.
- **Messaggistica:** Android supporta una vasta gamma di tecnologie per la messaggistica, come SMS, MMS, ed XMPP (basato su Smack di Jive Software).
- **Supporto Multimediale:** Android supporta tutti i maggiori formati di file audio/video/immagini, come: MPEG-4, H.264, MP3, AAC, AMR, JPEG, PNG e GIF.
- **Web Browser:** Il browser integrato in Android è basato sul framework open source WebKit (lo stesso di Safari, per intenderci).
- **Supporto hardware aggiuntivo:** Android permette di utilizzare videocamere, fotocamere, touchscreen, GPS, accelerometri ed altri componenti.
- **Sviluppo:** Android include un emulatore di dispositivi, strumenti per debugging, memoria, performance e un plug-in per l'IDE Eclipse.



Illustrazione 3: Architettura Kernel Android – <http://www.androidworld.it>

## Windows Mobile

È un sistema operativo dedicato ai sistemi embedded. Per questo sistema operativo Microsoft ha deciso di eliminare tutte le funzionalità non necessarie a vantaggio dell'utente che, anche in possesso di un dispositivo con hardware poco performante, riesce a trarre il massimo delle prestazioni del proprio smartphone.

Nel sistema operativo sono state integrate diverse tecnologie di comunicazione e networking, come:

- Bluetooth, 802.11 (802.1x, Extensible Authentication Protocol and 802.11 a configurazione automatica)
- Media Sense
- Funzionalità server come Remote Access Service (RAS)/ Point-to-Point Tunneling Protocol (PPTP)
- Server di tipo File Transfer Protocol (FTP), RTC Client API, Winsock 2.2, e Object Exchange Protocol (OBEX).
- Pila TCP/IP
- Network Driver Interface Specification 5.1.

Al kernel sono demandati la gestione della memoria, lo scheduling dei processi e alcune funzionalità che consentono la gestione dei file. Al kernel, inoltre, è

affidata la gestione della memoria virtuale, del multitasking e del multithreading.

La gestione dell'interfaccia grafica e dell'accesso all'hardware del dispositivo dedicato alla grafica è compito del modulo Gwes (Graphics, Windowing and Events Subsystem).

Gwes implementa le capacità del sistema di gestire le finestre, i dialoghi e i controlli (bottoni, caselle di testo, etc). Inoltre, Gwes è responsabile per la visualizzazione di grafica e di testo sul display grafico del dispositivo e della gestione della grafica avanzata. Il software applicativo e lo stesso Windows CE utilizzano le funzionalità dell' MGDI (Multiplatform Graphics Device Interface) del Gwes intensamente per la visualizzazione degli oggetti dell'interfaccia come menu, barre di scorrimento e icone.

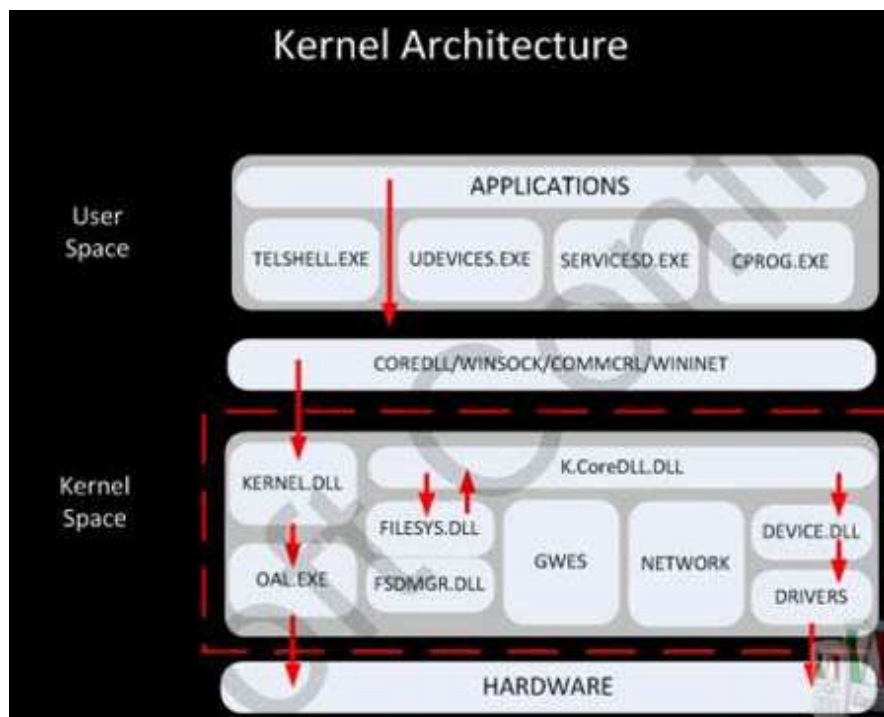


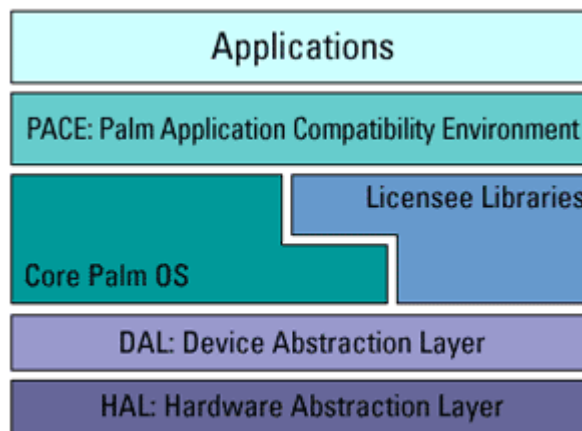
Illustrazione 4: Architettura Kernel Windows Mobile – <http://www.mobiletechworld.com>

## Palm OS

Sviluppato dalla società PalmSource Inc. è stato acquisito insieme alla società dalla ACCESS Systems che ha convertito il PalmOS in un layer di applicazioni, API e interfaccia utente da far girare su un kernel Linux ottimizzato per dispositivi mobili. Presenta, nella sua ultima versione, una serie di funzionalità particolarmente utili:

- multithreading
- sistema di crittografia a 128 bit
- supporto per schermi ad alta risoluzione
- supporto multimediale
- supporto delle funzionalità bluetooth

Anche nel caso di Palm è sufficiente che la versione presente sul dispositivo disponga di un browser all'interno del quale Javascript sia attivo e che il dispositivo sia in grado di connettersi in maniera autonoma o quanto meno disponga di un antenna wifi che consenta la connessione ad una rete senza fili.



*Illustrazione 5: Architettura Kernel Palm OS*  
– <http://www.apogeeonline.com>

## Research In Motion (RIM)

BlackBerry OS è programmato per svolgere più compiti alla volta. Esso utilizza alcuni dispositivi di input particolari, come la rotella di scorrimento, la trackball e il trackpad. Il sistema operativo supporta il Mobile Information Device Profile (MIDP) 1.0 e il Wireless Application Protocol (WAP) 1.2.

Java MIDP viene utilizzato per i dispositivi mobili. Una sincronizzazione wireless completa è consentita dal sistema operativo BlackBerry con il calendario, attività, contatti, lo scambio e-mail e note. Queste funzioni sono svolte da un software, BlackBerry Enterprise server (BES), che è una parte del sistema operativo. Un'alternativa alla BES è il BlackBerry Internet Service (BIS). La modalità *Enterprise* si attiva portando all'interno della propria azienda un apposito server che gestisce il collegamento fra il server di posta interno dell'azienda e la rete BlackBerry.

Altra caratteristica importante è il fatto che il software presente sul Blackberry sia sviluppato in Java e quindi particolarmente interessante per lo sviluppatore che vuole realizzare un'applicazione. Il fatto che sia scritto in Java permette di utilizzare ambienti di sviluppo disponibili gratuitamente sul web.

Il Blackberry a differenza dei dispositivi trattati precedentemente presenta delle caratteristiche in termini di connettività molto innovative, ma ha per contro delle limitazioni a livello di browser integrato abbastanza grossolane.

Abbiamo infatti riscontrato che nelle versioni meno recenti il browser del dispositivo non visualizza mai correttamente il layout della pagina, rendendo necessario un refresh della stessa.

La cosa può dipendere da fattori indipendenti dalla pagina, poiché è un problema di interpretazione del browser. Ad ogni modo il sistema funziona regolarmente nonostante questo problema di visualizzazione.

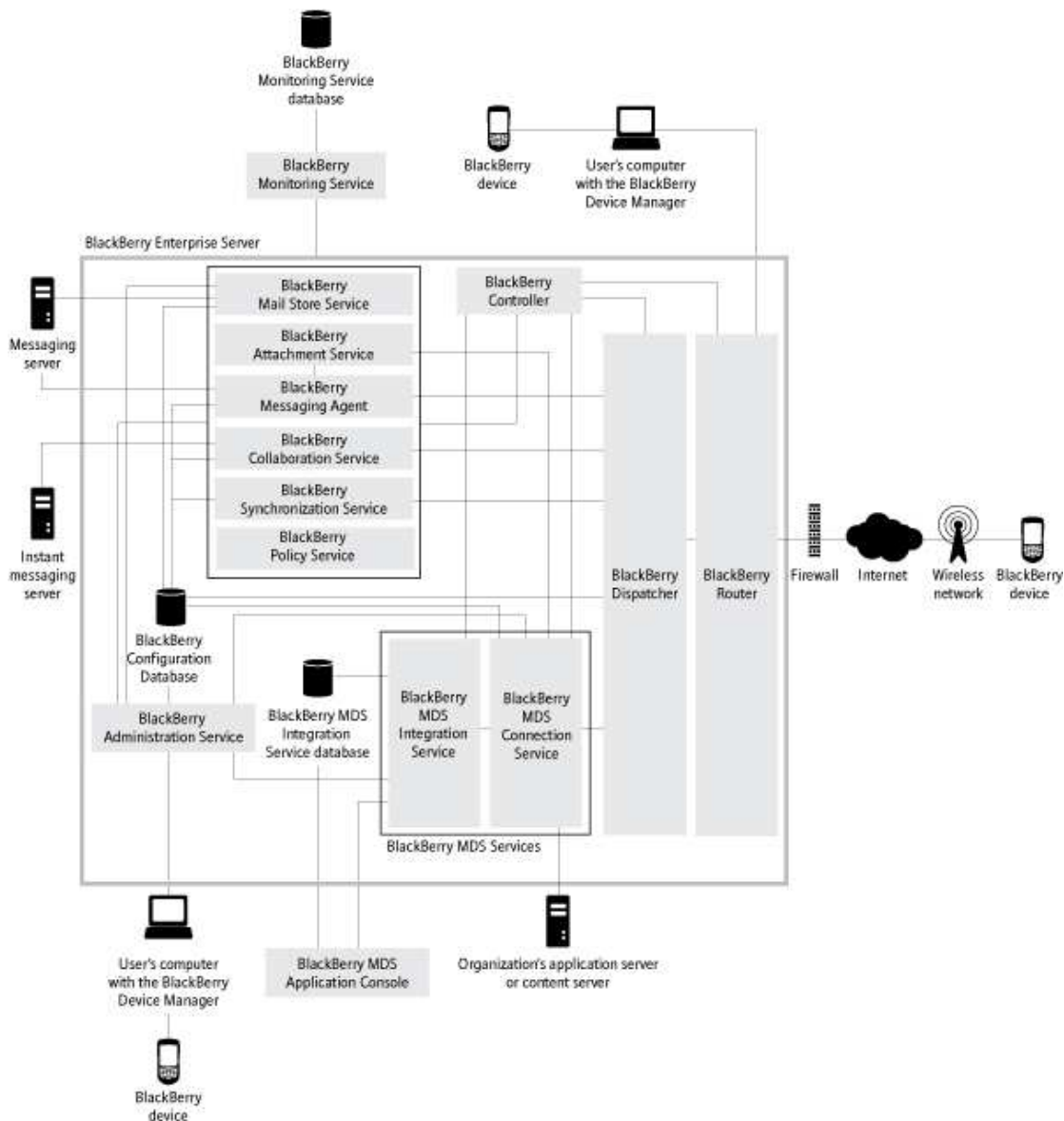


Illustrazione 6: Kernel Blackberry OS - <http://docs.blackberry.com>

## Dispositivi di comunicazione mobile

Il design e le interfacce, sempre più user friendly, hanno permesso di creare dispositivi interattivi e sempre più performanti. I brand che, sopra tutti, hanno il monopolio dei dispositivi mobili (quali telefoni cellulari, PDA, smartphone, ecc..) sono essenzialmente cinque e vengono elencati di seguito:

1. Android OS di serie su alcuni modelli HTC, Samsung.
2. Blackberry OS di serie su tutti i modelli Blackberry tra i quali ricordiamo: Bold, Storm, Storm II.
3. Windows Mobile di serie su alcuni modelli HTC quali HD Mini, Pro Touch e Pro Touch II, HD7 e altri ancora.
4. Apple iOS3 e iOS4 di serie sui modelli di iPhone 3, 3GS e 4.
5. Symbian di serie sui Nokia.

Dopo aver compiuto un'indagine e un'analisi sulle molteplici piattaforme presenti sul mercato si è deciso di prendere a titolo di campione un dispositivo mobile per ogni sistema operativo in maniera tale poter testare il sistema sui vari dispositivi.

L'idea di procedere in questa direzione deriva essenzialmente dal fatto che si vuole creare un sistema che sia multiplatforma e scalabile, un sistema che sia in grado di essere supportato da svariati dispositivi (e nella fattispecie da svariati sistemi operativi) senza impedimenti di tipo software e/o hardware. Con questo obiettivo è iniziata quindi una fase di testing mirata a valutare la compatibilità dei vari ambienti di sviluppo Open Source con tutto ciò che riguardasse i dispositivi.

Sono state riscontrate notevoli limitazioni imposte dai marchi proprietari dei dispositivi in merito allo sviluppo Open Source di software e applicazioni.

Prima tra tutti Apple.

Il brand di Cupertino infatti ha posto dei limiti notevoli e invalicabili a meno che lo sviluppatore non si doti di Computer Desktop o Laptop Apple con Sistema operativo non antecedente il MacOSX e non paghi il canone per acquistare la licenza da sviluppatore.

Questo primo ostacolo è stato poi aggirato stabilendo di concentrare le attenzioni su quei dispositivi il cui ambiente di sviluppo fosse integrabile tramite plug-in di ambienti di sviluppo più completi (NetBeans oppure Eclipse). Si è scelto di supportare quindi, in accordo con la filosofia del progetto, e cioè dello sviluppo di software open source, di tenere presenti solo i modelli di

telefoni cellulari e smartphone con sistema operativo Android, Windows Mobile, Blackberry e Symbian.



## Analisi delle tecnologie disponibili

Da una prima analisi dei sistemi operativi presenti sui dispositivi mobili abbiamo ricavato una lista di tecnologie che possono tornare utili nella realizzazione del progetto.

Di fatto la nostra idea è quella di creare una piattaforma in grado di erogare servizi web, quindi è stato necessario capire come e in che maniera combinare queste tecnologie a disposizione in maniera da garantire scalabilità e portabilità.

### Web Service.

E' per definizione un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori su di una medesima rete; caratteristica fondamentale di un Web Service è quella di offrire un'interfaccia software (descritta in un formato automaticamente elaborabile quale, ad esempio, il Web Services Description Language) utilizzando la quale altri sistemi possono interagire con il Web Service stesso attivando le operazioni descritte nell'interfaccia tramite appositi "messaggi" inclusi in una "busta" (la più famosa è SOAP): tali messaggi sono, solitamente, trasportati tramite il protocollo HTTP e formattati secondo lo standard XML.

Proprio grazie all'utilizzo di standard basati su XML, tramite un'architettura basata sui Web Service (chiamata, con terminologia inglese, Service Oriented Architecture, SOA) applicazioni software scritte in diversi linguaggi di programmazione e implementate su diverse piattaforme hardware possono quindi essere utilizzate, tramite le interfacce che queste "espongono" pubblicamente e mediante l'utilizzo delle funzioni che sono in grado di effettuare (i "servizi" che mettono a disposizione) per lo scambio di informazioni e l'effettuazione di operazioni complesse (quali, ad esempio, la realizzazione di processi di business che coinvolgono più aree di una medesima azienda) sia su reti aziendali come anche su Internet: la possibilità dell'interoperabilità fra diversi linguaggi di programmazione (ad esempio, tra Java e Python) e diversi sistemi operativi (come Windows e Linux) è resa possibile dall'uso di standard aperti.

### Touch Screen.

Senza dubbio una delle caratteristiche tecniche che più attirano l'utenza.

La possibilità di eliminare tasti veri, interagendo con il dispositivo e con le sue applicazioni semplicemente toccando lo schermo ha comportato una massiccia migrazione degli utenti verso dispositivi con questa caratteristica.

Infatti l'idea di poter contare su dispositivi con display molto grandi che fungessero all'occorrenza anche da tastiere ha rivoluzionato il modo di

intendere il telefono cellulare come dispositivo atto essenzialmente a fare telefonate e/o scrivere sms.

Dal classico touch screen si è passati poi con Apple al sistema Multi Touch che ha permesso una volta di più di confermare la versatilità e la funzionalità dei dispositivi con tale caratteristica. Il Multi Touch aggiunge -oltre al fatto di poter schiacciare bottoni "virtuali" sullo schermo- di poter contare su più punti di contatto contemporaneo sullo schermo.

In questa maniera non ci si limita più a toccare in un punto lo schermo del dispositivo per far eseguire una determinata operazione al software, bensì ci sono combinazioni e movimenti precisi con i quali è possibile effettuare operazioni che dovrebbero essere eseguite tramite un menu e quindi con tasti di navigazione.

### 3G.

Nell'ambito della telefonia cellulare, il termine 3G (acronimo di 3rd Generation) indica le tecnologie e gli standard di terza generazione.

I servizi abilitati dalle tecnologie di terza generazione consentono il trasferimento sia di dati "voce" (telefonate digitali) che di dati "non-voce" (ad esempio, download da internet, invio e ricezione di email ed instant messaging anche se la killer application utilizzata come traino dal marketing degli operatori 3G per l'acquisizione di nuova clientela è la videochiamata).

Le iniziali prospettive di forte crescita del mercato hanno portato, soprattutto in Europa, ed in Italia in particolare, a far sì che le aste tra gli operatori di telefonia mobile per la concessione delle licenze per l'utilizzo della banda siano state viste dai governi nazionali come possibili fonti di introito ed abbiano effettivamente richiesto agli operatori aggiudicatisi le gare grossi investimenti (nell'ordine di milioni di euro); questo fatto – unito alla necessità degli ulteriori notevoli investimenti necessari all'implementazione delle reti di telecomunicazioni 3G – ha portato diversi operatori a trovarsi in difficoltà finanziarie e, come conseguenza, ha ritardato l'effettiva operatività dei servizi di terza generazione in molti paesi.

Il primo paese ad introdurre la tecnologia 3G su scala commerciale è stato il Giappone: nel 2005 circa il 40% delle utenze erano esclusivamente su reti 3G e la transizione della maggior parte delle utenze da reti 2G a 3G dovrebbe essere conclusa dal 2006.

L'esperienza giapponese ha inoltre mostrato che la videotelefonia è lungi dall'essere la killer-application delle reti 3G ed occupa solo una frazione dei servizi offerti e fruiti su tali reti; tra i servizi di maggiore interesse per gli utenti vi è, ad esempio, il downloading di file musicali, l'utilizzo di servizi video a contenuto generato dall'utente, quali videoblog e moblog, e di servizi di incontri in video, denominati video dating.

### Wi-Fi.

Le cosiddette Reti Wireless (senza fili) sono uno dei tanti canali tramite i quali si possono utilizzare i servizi messi a disposizione.

Inoltre la possibilità di appoggiarsi a reti wi-fi (ove disponibili) permette di utilizzare un trasferimento dati molto più rapido e soprattutto ad un costo notevolmente contenuto -se non addirittura inesistente.

### Java.

La piattaforma Java è largamente utilizzata soprattutto per il fatto che è open source. Permette lo sviluppo di software compatibile con gran parte di dispositivi mobili e con i sistemi operativi montati.

Lo standard Java viene distribuito in varie versioni. Si parte da quella base che è JavaSE (Standard Edition per l'appunto) che offre funzionalità atte a fornire allo sviluppatore un base solida per lo sviluppo di software.

Alla SE segue la versione più completa JavaEE (acronimo per Enterprise Edition) che è costituita da un insieme di specifiche che definiscono le caratteristiche e le interfacce di un insieme di tecnologie pensate per la realizzazione di applicazioni di tipo enterprise e mission critical.

La Sun Microsystems mette a disposizione anche la versione di Java che pensata appositamente per i dispositivi mobili, quindi con proprietà hardware e prestazioni nettamente inferiori ad un computer Desktop o Laptop. Nella fattispecie la J2ME (nota anche come Java ME, cioè Micro Edition) è un runtime e una collezione di API per lo sviluppo di software dedicato a dispositivi a risorse limitate come i PDA, telefoni cellulari e simili. J2ME è la tecnologia più diffusa per lo sviluppo di giochi e utilities per i cellulari. Come le altre edizioni di Java, J2ME è una piattaforma portatile. Ciò implica che il suo funzionamento può essere emulato con un Personal Computer, cosa che semplifica notevolmente l'attività di sviluppo e di collaudo.

### JavaFX.

JavaFX è una famiglia di software applicativi, basati sulla Piattaforma Java, per la creazione di rich Internet applications, applicazioni web che hanno tutte le caratteristiche e funzionalità delle comuni applicazioni per computer. Con JavaFX è possibile realizzare delle applicazioni per computer, cellulari, dispositivi portatili di vario genere, televisori e altri tipi di piattaforme. JavaFX include, oltre ad una crescente libreria di funzionalità grafiche, un vero e proprio linguaggio di programmazione indipendente da Java, chiamato "JavaFX script", un linguaggio di scripting dichiarativo e staticamente tipizzato fortemente orientato alla programmazione grafica; questo rende la programmazione di rich Internet application, e di applicazioni grafiche in genere, particolarmente agevolata. In aggiunta al pacchetto JavaFX SDK, che include compilatore, utilità per l'esecuzione, e tutta la libreria JavaFX

necessaria per lo sviluppo, sono stati rilasciati alcuni strumenti che rendono lo sviluppo in JavaFX molto più agile:

- Un componente aggiuntivo per NetBeans che integra tutte le fasi di sviluppo JavaFX in un unico IDE
- JavaFX Production Suite: Utilità varie per facilitare il passaggio da programmi di grafica (Adobe Photoshop e Adobe Illustrator) e da formati come SVG, a codice JavaFX.

A Partire da Java 7, la libreria di JavaFX è distribuita con il Java Runtime Environment.

### **AMBIENTE DI SVILUPPO.**

Per lo sviluppo delle applicazioni e del server sono state utilizzate diverse SDK (Software Development Kit). Nella fattispecie si è scelto di utilizzare come ambiente di sviluppo Netbeans 6.9.1 che permette di programmare in linguaggio Java in modo compatibile per ogni sua piattaforma (SE, EE, ME, FX). Il vantaggio derivante dall'utilizzo di un ambiente di sviluppo come Netbenas è il fatto di poter installare plug in specifici per ogni tipologia di dispositivo in maniera tale da non dover cambiare l'ambiente di sviluppo qualora fosse cambiato il dispositivo.

Per questo progetto si è scelto di supportare vari dispositivi con Sistemi Operativi differenti cioè Blackberry e Android.

Entrambi necessitano di SDK specifiche, ma è anche possibile con qualche accorgimento e tramite il download di alcuni plugin di poter creare applicazioni compatibili con Android anche dall'interfaccia di Netbeans. Naturalmente -dal momento che l'idea di base è quella di fare utilizzo di software open source- si è dovuto escludere la piattaforma di Apple poiché implicava costi elevati. È stato quindi necessario quindi escludere dai dispositivi supportati l'iPhone, l'iPad e l'iPod Touch.

Gli unici problemi derivati dall'utilizzo delle specifiche SDK per ogni sistema operativo sono state riscontrate nello sviluppo per Blackberry. Infatti RIM (Research In Motion) proprietaria di Blackberry ha apportato delle modifiche alle classi Java più comuni. Questo ha comportato una brusca virata dal momento che l'idea di un sistema multipiattaforma, scalabile e compatibile con ogni dispositivo non era più possibile poiché è stato necessario creare client specifici per ogni dispositivo.

Altre complicazioni sono derivate dalla caompatibilità delle varie SDK con i sistemi operativi sui quali sono state installate. Si è riscontrato infatti che su Microsoft Seven gli emulatori di blackberry erano poco performanti e poco stabili, compromettendo così la fase di sviluppo e di test.

E' stato necessario quindi adottare la soluzione seguente: ricreare un ambiente windows XP su una macchina virtuale che permettesse di lavorare più tranquillamente.

## JSP.

JavaServer Pages, di solito indicato con l'acronimo JSP (letto anche talvolta come Java Scripting Preprocessor) è una tecnologia Java per lo sviluppo di applicazioni Web che forniscono contenuti dinamici in formato HTML o XML. Si basa su un insieme di speciali tag con cui possono essere invocate funzioni predefinite o codice Java (JSTL). In aggiunta, permette di creare librerie di nuovi tag che estendono l'insieme dei tag standard (JSP Custom Tag Library). Le librerie di tag JSP si possono considerare estensioni indipendenti dalla piattaforma delle funzionalità di un Web server. Nel contesto della piattaforma Java, la tecnologia JSP è correlata con quella dei servlet. All'atto della prima invocazione, le pagine JSP vengono infatti tradotte automaticamente da un compilatore JSP in servlet. Una pagina JSP può quindi essere vista come una rappresentazione ad alto livello di un servlet. Per via di questa dipendenza concettuale, anche l'uso della tecnologia JSP richiede la presenza, sul Web server, di un servlet container, oltre che di un server specifico JSP detto motore JSP (che include il compilatore JSP); in genere, servlet container e motore JSP sono integrati in un unico prodotto (per esempio, Tomcat svolge entrambe le funzioni).

JSP è una tecnologia alternativa rispetto a numerosi altri approcci alla generazione di pagine Web dinamiche, per esempio PHP, o ASP o la più tradizionale CGI. Differisce da queste tecnologie non tanto per il tipo di contenuti dinamici che si possono produrre, quanto per l'architettura interna del software che costituisce l'applicazione Web (e, di conseguenza, sui tempi di sviluppo, la portabilità, la modificabilità, le prestazioni, e altri aspetti di qualità del software).

## Comunicazione client-server

La comunicazione client-server avviene tramite protocollo http.

Si tratta infatti di un progetto web, dove i servizi sono dei Web Services che possono essere erogati sia da Metacortex sia creati ex novo da terzi poi resi disponibili sulla piattaforma creata.

Il primo step è la connessione ad internet. Nel caso di reti wifi ci si collega e si è immediatamente riconosciuti sulla rete con l'indirizzo IP assegnato dal server, mentre se ci si collega utilizzando il proprio dispositivo come modem allora l'instradamento sulla rete internet richiederà il tempo che mette il dispositivo ad instaurare una connessione verso internet sulle reti che supporta, ad esempio il 3G, 2G, GPRS, UMTS, ecc..

Come per le tecnologie disponibili di seguito abbiamo analizzato i vari linguaggi utili per la comunicazione tra il dispositivo utente e il server Metacortex.

### JSON.

JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati. Per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi. Si basa su un sottoinsieme del linguaggio di programmazione javascript.

JSON è un formato di testo completamente indipendente dal linguaggio di programmazione, ma utilizza convenzioni conosciute dai programmatori di linguaggi della famiglia del C, come C, C++, C#, Java, JavaScript, Perl, Python, e molti altri. Questa caratteristica fa di JSON un linguaggio ideale per lo scambio di dati.

JSON è basato su due strutture:

- Un insieme di coppie nome/valore. In diversi linguaggi, questo è realizzato come un oggetto, un record, uno struct, un dizionario, una tabella hash, un elenco di chiavi o un array associativo.
- Un elenco ordinato di valori. Nella maggior parte dei linguaggi questo si realizza con un array, un vettore, un elenco o una sequenza.

Queste sono strutture di dati universali. Virtualmente tutti i linguaggi di programmazione moderni li supportano in entrambe le forme. E' sensato che un formato di dati che è interscambiabile con linguaggi di programmazione debba essere basato su queste strutture.

### XML.

XML (sigla di eXtensible Markup Language) è un metalinguaggio di markup, ovvero un linguaggio marcatore che definisce un meccanismo sintattico che consente di estendere o controllare il significato di altri linguaggi marcatori.

Costituisce il tentativo di produrre una versione semplificata di SGML che consenta di definire in modo semplice nuovi linguaggi di markup da usare in ambito web. Il nome indica quindi che si tratta di un linguaggio marcatore (markup language) estensibile (eXtensible) in quanto permette di creare tag personalizzati.

### RSS.

È uno dei più popolari formati per la distribuzione di contenuti Web; è basato su XML, da cui ha ereditato la semplicità, l'estensibilità e la flessibilità. L'applicazione principale per cui è noto sono i *feed RSS*, che permettono di essere aggiornati su nuovi articoli o commenti pubblicati nei siti di interesse senza doverli visitare manualmente uno a uno.

RSS definisce una struttura adatta a contenere un insieme di notizie, ciascuna delle quali sarà composta da vari campi (nome autore, titolo, testo, riassunto, ...). Quando si pubblicano delle notizie in formato RSS, la struttura viene aggiornata con i nuovi dati; visto che il formato è predefinito, un qualunque lettore RSS potrà presentare in una maniera omogenea notizie provenienti dalle fonti più diverse.

### ATOM.

Con il termine Atom ci si riferisce a due standard distinti. Atom Syndication Format è un formato di documento basato su XML per la sottoscrizione di contenuti web, come blog o testate giornalistiche. Atom Publishing Protocol (AtomPub o APP) è un semplice protocollo basato su HTTP usato per la lettura, creazione e aggiornamento di risorse web.

Atom è basato sull'esperienza delle varie versioni del protocollo lanciato da Netscape, RSS. In passato atom è stato conosciuto (anche se solo per un breve periodo) come "Pie" e poi come "Echo".

La specifica completata di atom è stata presentata all'IETF per l'approvazione, nel giugno 2005, l'ultimo passo per diventare uno standard internet RFC. A luglio, il formato di pubblicazione Atom è stato dichiarato pronto per essere implementato. L'ultimo formato dei dati e dei protocolli di pubblicazione di Atom sono linkati nella home page del gruppo di lavoro.

Prima che Atom entrasse nello IETF, il gruppo ha prodotto "Atom 0.3", che supporta una moltitudine di strumenti per la sottoscrizione, sia per chi riceve la sottoscrizione, sia per chi la pubblica. Anche alcuni servizi di Google, come Blogger o Gmail, usano questo formato di pubblicazione.

Così come produce il formato di sottoscrizione, il Progetto Atom sta lavorando sul Protocollo di Pubblicazione Atom, con lo scopo di migliorare e standardizzare i meccanismi di pubblicazione già esistenti come le API Blogger e il protocollo server/client XML-RPC LiveJournal.

## Sistemi legacy con cui integrarsi

Un aspetto importante, nonché grande ambizione, del sistema è quello dell'integrazione, o meglio, della possibilità di integrazione della piattaforma studiata con dei sistemi anteriori.

La problematica della compatibilità quindi non si è presentata unicamente a livello client laddove il cellulare o smartphone considerato non fosse in grado di supportare la runtime creata, ma anche a livello del server e soprattutto in termini di Database.

Pensiamo ad esempio ad un'azienda medio grande con un Database specifico del proprio settore commerciale. Il DB è del tipo IBM AS400, quindi è possibile affermare che non si tratti di uno dei prodotti -in termini di DB- degli ultimi 5 anni. Supponiamo che la suddetta azienda decida di ampliare il settore commerciale e decida di destinare ad ogni membro del reparto un dispositivo con il quale vendere licenze e/o abbonamenti a determinati servizi.

Fa quindi richiesta di acquisto per la piattaforma.

Ebbene, se la piattaforma sviluppata non fosse in grado di integrarsi con sistemi legacy del tipo dell'AS400 è ragionevole pensare che l'azienda citata nell'esempio non avrebbe mai pensato di rimuovere un sistema solido come quello di IBM per passare ad uno più recente, comportando così costi elevati, formazione del personale su un diverso sistema e successivamente anche i costi della nuova piattaforma.

È stato quindi necessario considerare tutti gli aspetti tecnici tali da poter garantire compatibilità "retroattiva" atta a garantire all'utente finale un sistema versatile e che comportasse pochi step e pochi upgrade del sistema già esistente.



## Scelta preliminare di tecnologie

L'idea di base che ha ispirato la realizzazione di questo sistema è quella di creare un sistema multiplatforma che sia in grado di erogare servizi web.

L'idea di multiplatforma si può riassumere in maniera esaustiva in pochi concetti: si crea una parte server che gestisce tutte le informazioni e la navigazione e poi si crea un applicativo che va installato lato client. La peculiarità di questo applicativo sta nel fatto che deve, o meglio dovrebbe, essere compatibile con tutti i dispositivi mobili (posto che rientrino nella categoria degli smartphone e che soddisfino alcune caratteristiche a livello hardware e di capacità di calcolo).

In questa maniera non è necessario ricordarsi il modello del dispositivo oppure consultare liste di dispositivi supportati. In generale, riassumendo, la missione è: fare in modo che la maggior parte degli smartphone, e più nel dettaglio qualsiasi sistema operativo, permetta l'installazione dell'applicativo sviluppato. La necessità di sviluppare un applicativo destinato ad essere installato e successivamente utilizzato su un dispositivo mobile ha portato alla luce la problematica della scelta del ambiente di sviluppo da adottare.

In un primo momento la scelta è ricaduta sulla distribuzione J2ME (Sun Microsystem) che si è rivelata inadatta allo scopo in quanto poco performante in ambito grafico. Si è scelto quindi di utilizzare un ambiente di sviluppo che sopperisse alle mancanze di J2ME. La scelta è ricaduta su JavaFX che si è rivelata essere la più indicata in quanto, stando alle dichiarazioni degli sviluppatori della Oracle, è supportato da tutti i sistemi operativi mobili (ad eccezione di Apple iOS 4).

JavaFX in un primo momento è risultato essere un ambiente di sviluppo versatile perché specifico per la creazione di interfacce grafiche e per il design di applicazioni per dispositivi mobili quali cellulari di nuova generazione, smartphone e P.D.A. .

Lo sviluppo di una piccola porzione di codice ha permesso di testare la bontà di JavaFX. Il codice implementato realizzava una pagina di accesso dove fosse necessaria l'interazione con l'utente. Questa pagina è stata utilizzata come test per esaminare come i sistemi operativi mobili supportassero veramente il codice scritto con JavaFX.

Com'è prevedibile, però, i vari brand hanno posto parecchi limiti in termini di "libertà" per gli sviluppatori di applicazioni multiplatforma. Infatti si è dovuto constatare che tutte le applicazioni sono sviluppate ad hoc per ogni sistema operativo poiché ognuno di essi e più in generale ogni brand presente sul mercato adotta le sue policy per lo sviluppo di applicativi.

Nella fattispecie si è dovuto virare bruscamente da quello che era l'obiettivo iniziale e "ripensare" l'architettura cercando di mantenere l'idea di multiplatforma con la quale si era partiti poiché l'unico sistema operativo mobile in grado di supportare la virtual machine di JavaFX è risultato essere

Windows Mobile. Android e RIM anche se compatibili con software sviluppato in Java, non sono ancora compatibili con codice scritto con JavaFX. Questo ha comportato un abbandono la con la conseguente necessità di adottare una nuova strategia progettuale.

A questo proposito è stato pensato -lato client- di utilizzare un framework comune a tutte le piattaforme che utilizzasse le classi base del linguaggio java (comune a tutti i dispositivi) e successivamente di realizzare un'applicazione specifica per ogni piattaforma che poggiasse su questo framework magari integrandolo con gli oggetti già implementati nel layout grafico specifico per ogni modello di smartphone.

Anche questo approccio però non ha dato i risultati sperati poiché gli ambienti di sviluppo necessari per progettare applicazioni per sistemi operativi quali Android e per smartphone quali il Blackberry sono completamente slegati tra di loro nonostante le applicazioni si scrivano in linguaggio Java. Infatti tutte le classi normalmente utilizzate sono state cambiate in maniera tale da garantire all'utilizzatore di -ad esempio- Blackberry sempre la stessa interfaccia grafica.

La soluzione si è presentata sotto forma di web app che permettesse di aggirare i problemi di compatibilità e che permettesse uno sviluppo unico e compatibile con la grande maggioranza dei browser presenti negli attuali dispositivi.

In questa maniera le limitazioni imposte da Apple e i problemi derivanti dalla poca funzionalità degli ambienti di sviluppo per BlackBerry sono state sistematicamente eliminate poiché sul dispositivo vero e proprio non viene installato nulla.

## Conclusione

La necessità di avere dispositivi reali sui quali e con i quali testare il progetto ha portato alla redazione di un elenco di dispositivi veri e propri da usare come mezzi di testing che fossero comunque in grado di instaurare una connessione e dotati di browser integrato.

I modelli che abbiamo scelto sono:

- Blackberry Bold 9300.
- Blackberry Curve 8520.
- Nokia N97 Mini.
- Samsung Galaxy Mini.
- HTC HD T 5555.
- Nokia C5-00.
- Nokia N70.

Questi telefoni soddisfano tutte le caratteristiche che sono state messe insieme grazie all'analisi dei sistemi operativi mobili attualmente disponibili e all'analisi delle tecnologie disponibili.

Infatti tutti possono instaurare una connessione ad internet su rete 3G e soprattutto hanno un browser interno che permette di accedere al sito per poter scaricare l'applicativo.

Altro fattore importante: tutti i dispositivi supportano l'installazione di applicazioni e i sistemi operativi montati sono compatibili con applicativi sviluppati in java.

In conclusione questa analisi dello stato dell'arte e di tutto ciò che attualmente è disponibile sul mercato dei dispositivi mobili ha permesso di evidenziare pro e contro di diverse tipologie di smartphone, tenendo conto di ciò che è necessario (sia in termini hardware che software) affinché la portabilità e la scalabilità siano effettive.

Infine, nonostante non ci sia possibile testare ogni dispositivo mobile, smartphone, PDA presente sul mercato, la scelta di questi dispositivi è un campione abbastanza significativo delle tipologie di telefono (eccezion fatta per alcuni dispositivi pensati ad hoc per un utilizzo particolare) che oggi vengono utilizzati di più, e, quindi, un risultato positivo in termini di compatibilità tra l'applicativo e il telefono rappresenta un dato importante nella realizzazione di questo progetto.