



Università degli Studi dell'Aquila  
Facoltà di Ingegneria

---

Tesi di Laurea Specialistica in Ingegneria Informatica e Automatica

# Realizzazione di un prototipo della versione Cloud SaaS della suite IBM BigFix: Automazione del deployment e del testing

**Relatore interno:**

Prof. Serafino Cicerone

**Laureando:**

Beniamino Negrini

**Correlatore:**

Prof. Dott. Marco Secchi

**Relatore esterno:**

Dott. Bernardo Pastorelli

---

Anno Accademico 2016/2017

*Dedica a piè pagina*

# Indice

<b>1</b>	<b>CapitoloEsempio</b>	<b>4</b>
<b>2</b>	<b>Introduzione</b>	<b>5</b>
<b>3</b>	<b>IBM BigFix</b>	<b>6</b>
3.1	BigFix . . . . .	6
3.1.1	Architettura . . . . .	6
3.1.2	BigFix Platform . . . . .	9
3.1.3	BigFix Applications . . . . .	9
3.1.4	Fixlets . . . . .	9
<b>A</b>	<b>Tecnologie Utilizzate</b>	<b>11</b>
A.1	Linguaggi di programmazione . . . . .	11
A.2	Linguaggi di Markup e Stile . . . . .	11
A.3	Framework . . . . .	11
A.4	Ambiente di Sviluppo . . . . .	12
A.4.1	Eclipse . . . . .	12
A.4.2	Piattaforma Web . . . . .	12
A.4.3	Browser Testing . . . . .	12

Capitolo

1

CapitoloEsempio

Capitolo introduttivo di prova

- item di prova 1
- item di prova 2

Paragrafo    Paragrafo di prova

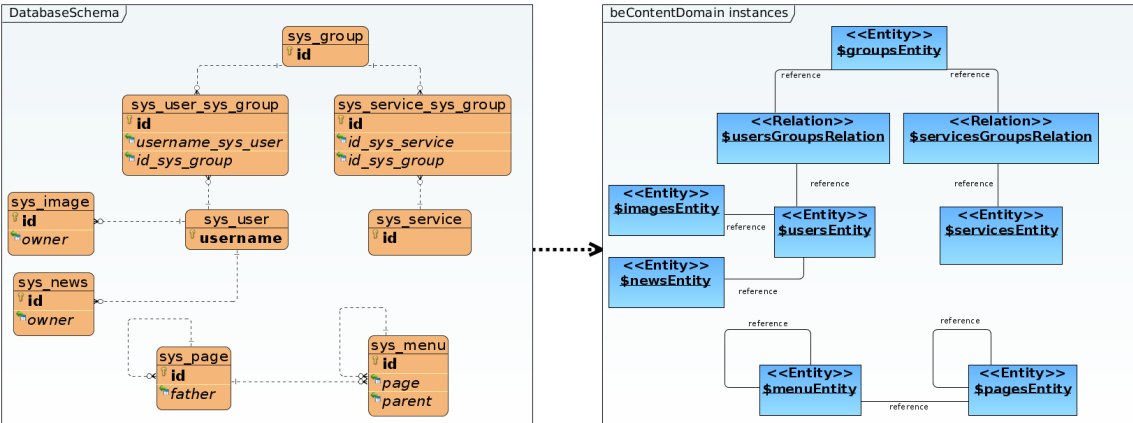


Figura 1.1: Immagine di prova

# Capitolo 2

## Introduzione

E' sempre più evidente che il cloud computing è il futuro del software. La rivoluzione consta nella distribuzione dei servizi di calcolo e alla virtualizzazione delle risorse. Viene data così all'utente la sensazione di un utilizzo centralizzato. Tutto ciò si è reso realizzabile dal momento in cui l'accesso alla rete è divenuto possibile da sempre più dispositivi con velocità di connessione sempre maggiore.

la tematica del cloud computing è stata centrale nel mio lavoro di tesi presso l'azienda IBM (International Business Machines Corporation), nella sua sede di Roma. Ho partecipato attivamente alla realizzazione di un prototipo software, ossia la versione SaaS della suite IBM BigFix. BigFix è una suite di prodotti dedicati alle aziende che risolvono problematiche di Endpoint Security e di compliance di dispositivi interni a determinate politiche aziendali. Tramite questi prodotti si ottiene pieno controllo su tutti i dispositivi aziendali, si rilevano eventuali attacchi, si distribuiscono aggiornamenti e patch, fino anche alla gestione energetica.

La sfida da me raccolta è quindi proprio quella di portare tutto questo arsenale di strumenti nella leggerezza del cloud. Rendendolo disponibile, nel giro di pochi minuti, anche a chi è sempre stato intimorito dalla difficoltà di installazione di uno strumento così potente, ma allo stesso tempo complesso.

# Capitolo 3

## IBM BigFix

### 3.1 BigFix

I prodotti della suite IBM BigFix consentono di monitorare e gestire in tempo reale un elevato numero di dispositivi fisici e virtuali connessi (fino a 250.000). Questi possono essere sia fisici che virtuali, come ad esempio server, desktop, notebook, dispositivi mobili, tablet, POS, ATM, chioschi self-service. Gli utenti principali di questi prodotti sono gli amministratori di sistema. Tramite le applicazioni BigFix possono avere il pieno controllo sugli endpoint, come distribuire software, applicare delle patch, effettuare il deploy di sistemi operativi, proteggere da attacchi di rete e molto altro.

#### 3.1.1 Architettura

Un'architettura di BigFix è, per sua natura, molto articolata, poichè la necessità è quella di gestire un numero elevato ed eterogeneo di dispositivi. Essa si basa sul consolidato pattern stilistico Client/Server, ma con una struttura leggermente variata, prevedendo l'inserimento di un ulteriore layer frapposto tra client e server, i relay, i quali sono fondamentali per bilanciare il carico.

Ma partiamo subito con un esempio per avere un ponto di riferimento. Come possiamo notare, l'elemento fondamentale è il server, il quale ha lo scopo di racco-

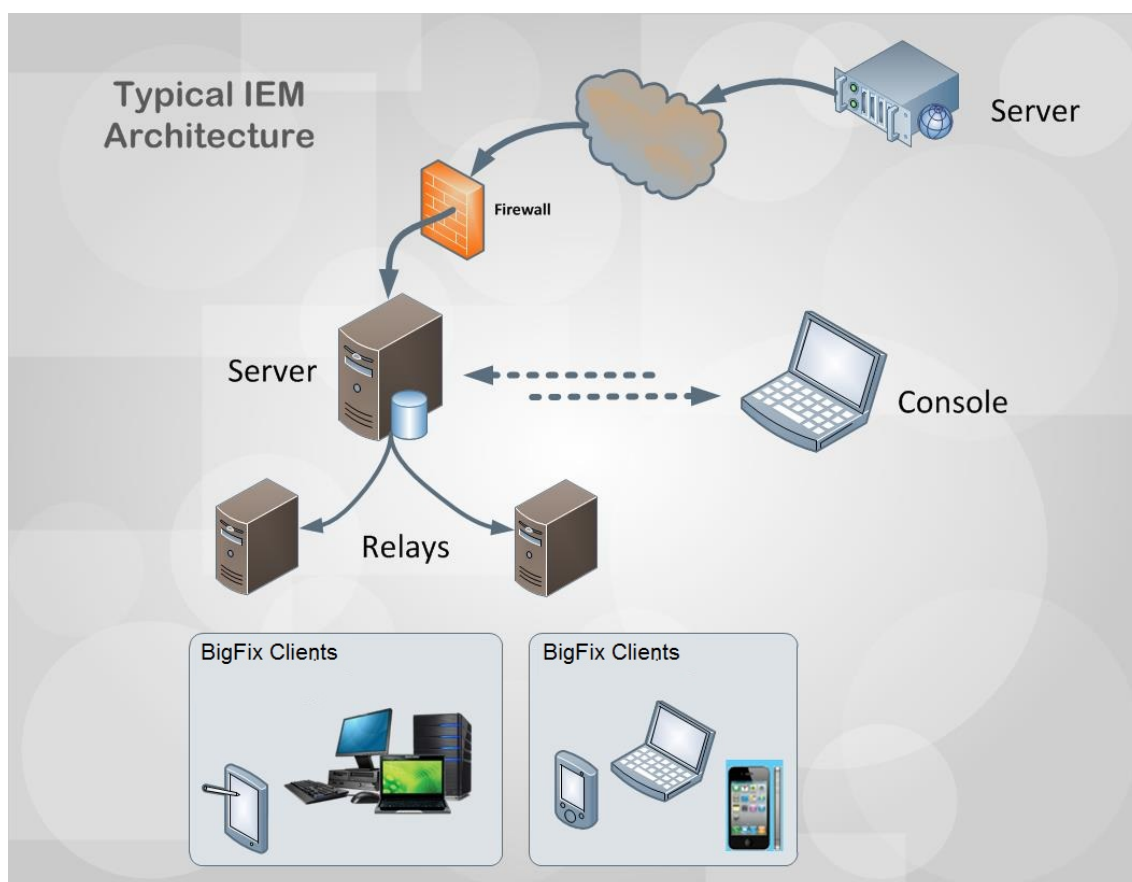


Figura 3.1: Un'architettura BigFix di esempio

gliere dei particolari messaggi, le Fixlet. Questi messaggi possono essere visualizzati dall'operatore che lavora sulla console e inoltrati quindi, a questo punto ai relay. E' competenza dei relay poi interagire con i singoli client e assicurarsi l'esecuzione delle Fixlet. Le Fixlet, infatti, altro non sono che delle azioni che devono essere necessariamente compiute dai client. Andiamo ad analizzare le singole componenti dell'architettura.

**Servers** Il server coordina tutto il flusso di informazioni e si preoccupa di salvare le informazioni sul database. Al tempo stesso però, lascia agli Agent il compito di effettuare analisi ed eseguire azioni specifiche. Ciò consente di liberare il Server da un pesantissima computazione. Per questo motivo il Server stesso può gestire un altissimo numero di client.

**Relays** I Relay si comportano come una cache tra i Client e il Server e sono di numero variabile in base al numero di Client. Aiutano il Server a gestire i dispositivi anche se funzionalmente non sono altro che Client che sono stati promossi a Relay, aggiungendo a loro dei servizi. A questo punto i Client non si interfaceranno mai con il Server, alleggerendone così notevolmente il workload. Possono, ad esempio, più Client richiedere un download al Relay, il quale effettuerà un'unica richiesta al Server

**Agents** Un Agent è installato su ogni Client facente parte dell'architettura di BigFix. essi hanno il compito di raccogliere le Fixlet, tramite le quali sono in grado, ad esempio, di individuare e correggere le exposure di sicurezza, configurazioni scorrette e altre vulnerabilità. Fa dei continui check per confrontare lo stato del dispositivo con le policy stabilite. Appena scopre che il dispositivo è fuori dalla compliance, viene informato il Server ed agisce subito per porre rimedio e, al termine, informa nuovamente sull'esito dell'operazione.

**Web Reports** I Web Reports sono il componente che consentono ad utenti autorizzati di monitorare tutti i dispositivi di BigFix. Si può in questo modo tenere traccia di vulnerabilità, azioni richieste e molto altro.

**Consoles** La Console permette agli amministratori di interagire con tutti i Client dell'ambiente BigFix. Gli utenti possono così distribuire velocemente patch e configurazioni.

BigFix, da un punto di vista logico, si suddivide in due grandi macro-componenti, la Platform e le Applications. La prima svolge la funzione di layer sulla quale vengono sviluppate tutte le funzionalità dello strato di applications. Questa suddivisione consente una chiara suddivisione delle competenze da parte di progettisti, sviluppatori, tester e assistenti dei clienti. Il team della platform si concentra quindi nel fornire una solida infrastruttura al team delle applications, il quale svilupperà i singoli strumenti al servizio dell'utente.



### 3.1.2 BigFix Platform

La Platform è una tecnologia multi-layer scritta in linguaggio C++ che agisce come colonna portante di tutta l'infrastruttura di BigFix. La Platform svolge infatti funzioni fondamentali, spesso utilizzate anche da altre applicazioni dei layer superiori. La Platform divide le responsabilità.

**Lifecycle Management** Include il controllo remoto, la distribuzione dei software e il deploy di sistemi operativi.

**Patch Management** Consente l'applicazione e la gestione delle patch.

**Core Protection** E' il cuore di tutta la routine di Security. Troviamo qui delle funzionalità anti-malware, firewall e protezione da qualsiasi variante di virus.

**Inventory** Raccoglie informazioni peculiari a riguardo dei software installati su ogni dispositivo. E' in grado di effettuare analisi sugli utilizzi fornendo una base per la decisione sulle licenze da acquistare.

**Server Automation** Fornisce un Hypervisor per le Virtual Machines. Esso controlla infatti eventuali malfunzionamenti degli ambienti virtuali.

### 3.1.3 BigFix Applications

Tutti i prodotti applicativi che fanno parte di questo componente consentono di gestire in maniera semplice tutte le operazioni inerenti alla security. A differenza della Platform sono implementate in linguaggio Java ed hanno funzionalità atomiche tra di loro. Sono l'interfaccia principale con il quale interagisce l'amministratore aziendale.

### 3.1.4 Fixlets

Le Fixlet sono il metodo attraverso il quale si svolgono tutte le operazioni come distribuzione di software, installazioni di patch e configurazioni. Esse sono dei mes-

saggi inoltrati ai client di BigFix e utilizzano un linguaggio di query specifico, il Relevance.

### **Il linguaggio Relevance**

Con una Fixlet si può anche ispezionare un desiderato aspetto di un client. Il linguaggio di Relevance consente di interrogare il client identificandone caratteristiche dell'hardware o del software tramite gli Inspectors. Una necessità può essere infatti quella di applicare una Fixlet solamente a dei client con determinate caratteristiche hardware/software oppure che si trovano in stati ben definiti. Si può in questo modo, facilmente identificare il corretto sottoinsieme di client ai quali è destinata una nuova Fixlet ed applicarla solo ad essi.

## Tecnologie Utilizzate

### A.1 Linguaggi di programmazione

- PHP 5.4.7  
<http://www.php.net/>;
- Javascript  
<http://www.w3.org/standards/webdesign/script>;

### A.2 Linguaggi di Markup e Stile

- HTML4/HTML5;
- CSS/CSS3;

### A.3 Framework

- Smarty Template Engine  
<http://www.smarty.net/>;
- JQuery  
<http://jquery.com/>;

- JQueryUI  
<http://jqueryui.com/>;
- beContent  
<http://www.becontent.org/>;

## **A.4 Ambiente di Sviluppo**

### **A.4.1 Eclipse**

Per Eclipse sono state utilizzate due versioni differenti, la 4.2.2 in ambiente Windows e la 3.8.0 in ambiente Ubuntu/Linux

<http://www.eclipse.org/>

Inoltre è stato utilizzato il pacchetto

- PHP Development Tools 3.1.1  
<http://projects.eclipse.org/projects/tools.pdt>;

### **A.4.2 Piattaforma Web**

#### **XAMPP**

<http://www.apachefriends.org>

- Apache Web Server ver. 2.4.3  
<http://httpd.apache.org/>;
- MySql Database Management System ver. 5.5.27  
<http://dev.mysql.com/>;

### **A.4.3 Browser Testing**

#### **Mozilla Firefox**

- Firebug ver 1.11.2  
<http://getfirebug.com/>

- Plug-In Validator ver. 0.0.6  
<https://addons.mozilla.org/it/firefox/addon/validator/>;
- Plug-In Google Page Speed ver. 2.0.2.3  
<https://developers.google.com/speed/pagespeed/?hl=it-IT>;

### **Google Chrome**

- Strumenti per gli sviluppatori integrati

### **Responsive Testing**

- Viewport Resizer- Responsive Design Bookmarklet  
<http://lab.maltewassermann.com/viewport-resizer/> ;



*Dedica a fine pagina*