

Allegato II al Capitolato Tecnico -

CIG 80796512EE

Descrizione dell'ambiente attuale

Architettura di sistema FTWEB

Sommario

1. Premessa.....	3
La descrizione dell'Architettura di sistema illustra l'ambiente tecnologico di FTWEB in termini di:..3	
• disegno dell'architettura sistemistica della soluzione (HW);	3
• dettaglio delle macchine utilizzate (HW).....	3
• architettura logica della soluzione.....	3
L'obiettivo è quello di illustrare le caratteristiche di massima della soluzione tecnologica attualmente utilizzata.	3
2. Sistema Hardware.....	4
2.1 Tabella di dettaglio	4
2.2 Disegno dell'architettura di sistema	5
3. Architettura logica della soluzione	6
3.1 Disegno dell'architettura logica della soluzione	6
3.1.1 Strato di Presentazione	7
3.1.2 Business Logic e persistenza dei dati	8
3.1.3 Integration Logic	9
3.1.4 Moduli aggiuntivi	Errore. Il segnalibro non è definito.

1. Premessa

La descrizione dell'Architettura di sistema illustra l'ambiente tecnologico di FTWEB in termini di:

- disegno dell'architettura sistemistica della soluzione (HW);
- dettaglio delle macchine utilizzate (HW).
- architettura logica della soluzione

L'obiettivo è quello di illustrare le caratteristiche di massima della soluzione tecnologica attualmente utilizzata.

2. Sistema Hardware

2.1 Tabella di dettaglio

In questo paragrafo è descritto in forma tabellare l'hardware utilizzato per la realizzazione del prodotto.

		Ruolo	Cluster	OS	Software installati
1	Bilanciatore SW APACHE	HttpServer: Load Balancing, Reverse Proxy, Caching Services	NO	Centos 7	Apache 2.4.6
2	Application Server - WildFly 1	Application Server JEE #1: Business Logic Services (CLUSTER active-active)	SI	Centos 7	Wildfly v.13, java 1.8.0_172,
3	Application Server - WildFly 2	Application Server JEE #2: Business Logic Services (CLUSTER active-active)	SI	Centos 7	Wildfly v.13, java 1.8.0_172,
4	Application Server BI - Knowage Server	Business Intelligence Services	NO	Centos 7	java 1.8.0_172
5	AURIGA	Document Management Services	SAAS		
6	Identity Management System - WSO2 (SSO)	Identity Store, Identity Management, Access Management, SSO	NO	Centos 7	Wso2 5.6.0, java 1.8.0_172
7	PgPool	Cluster PostgreSQL Load Balancing/Connection pooling	NO	Centos 7	
8	Relational Database - PostgreSQL 1	Relational DataBase Server Business Logic	SI	Centos 7	postgresql v.9.2.24
9	Relational Database - PostgreSQL 2	Relational DataBase Server Business Logic	SI	Centos 7	postgresql v.9.2.24
10	DB Knowage	DB per Knowage	NO	Centos 7	

2.2 Disegno dell'architettura di sistema

Si illustra di seguito l'architettura del sistema.

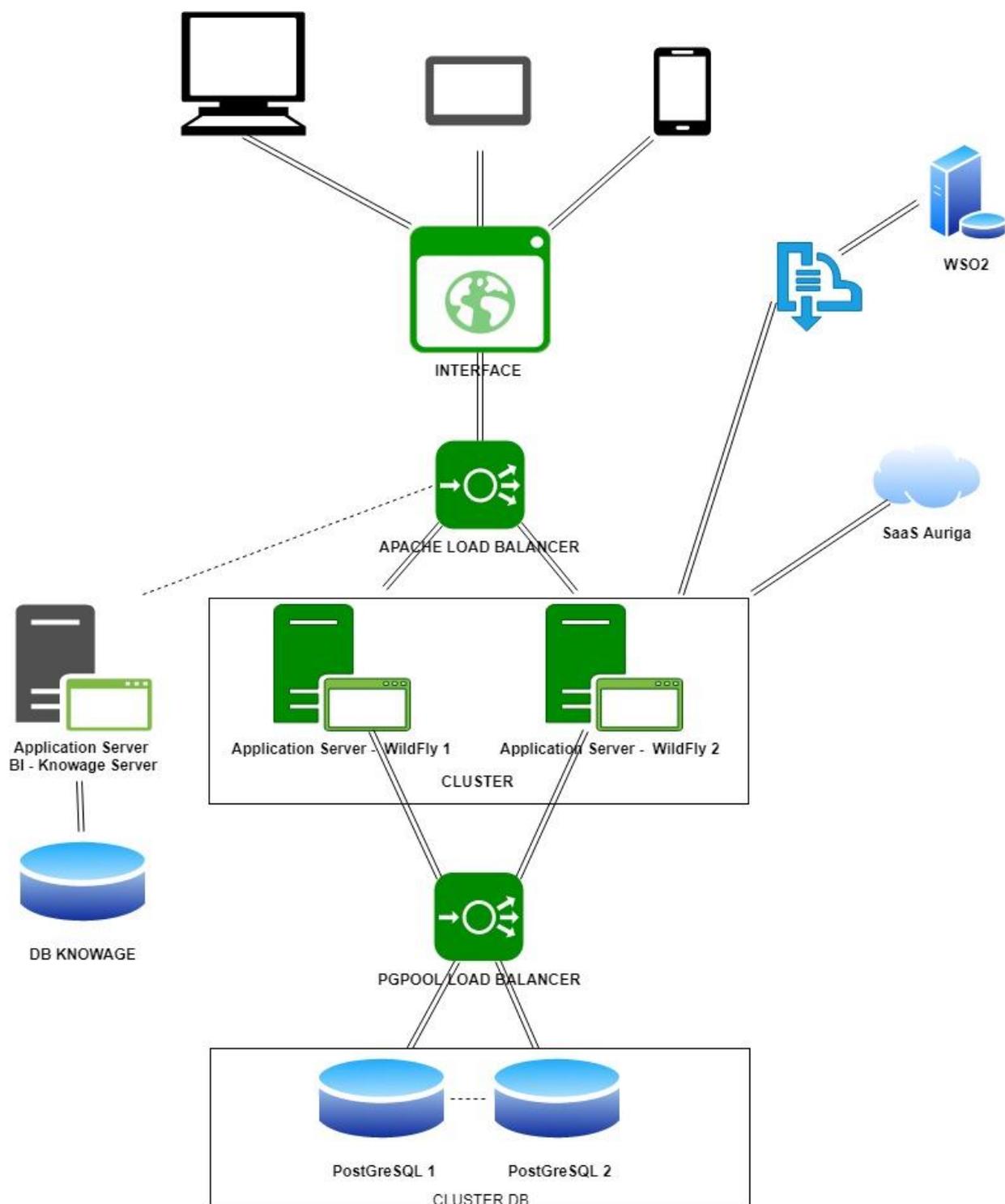


Figura 1 - Architettura del sistema

3. Architettura logica della soluzione

Si riporta in questo capitolo lo schema raffigurante l'architettura logica della soluzione.

3.1 Disegno dell'architettura logica della soluzione

Al fine di semplificare la lettura e la comprensione dell'architettura, gli strati di Logica saranno raffigurati in modalità "black box".

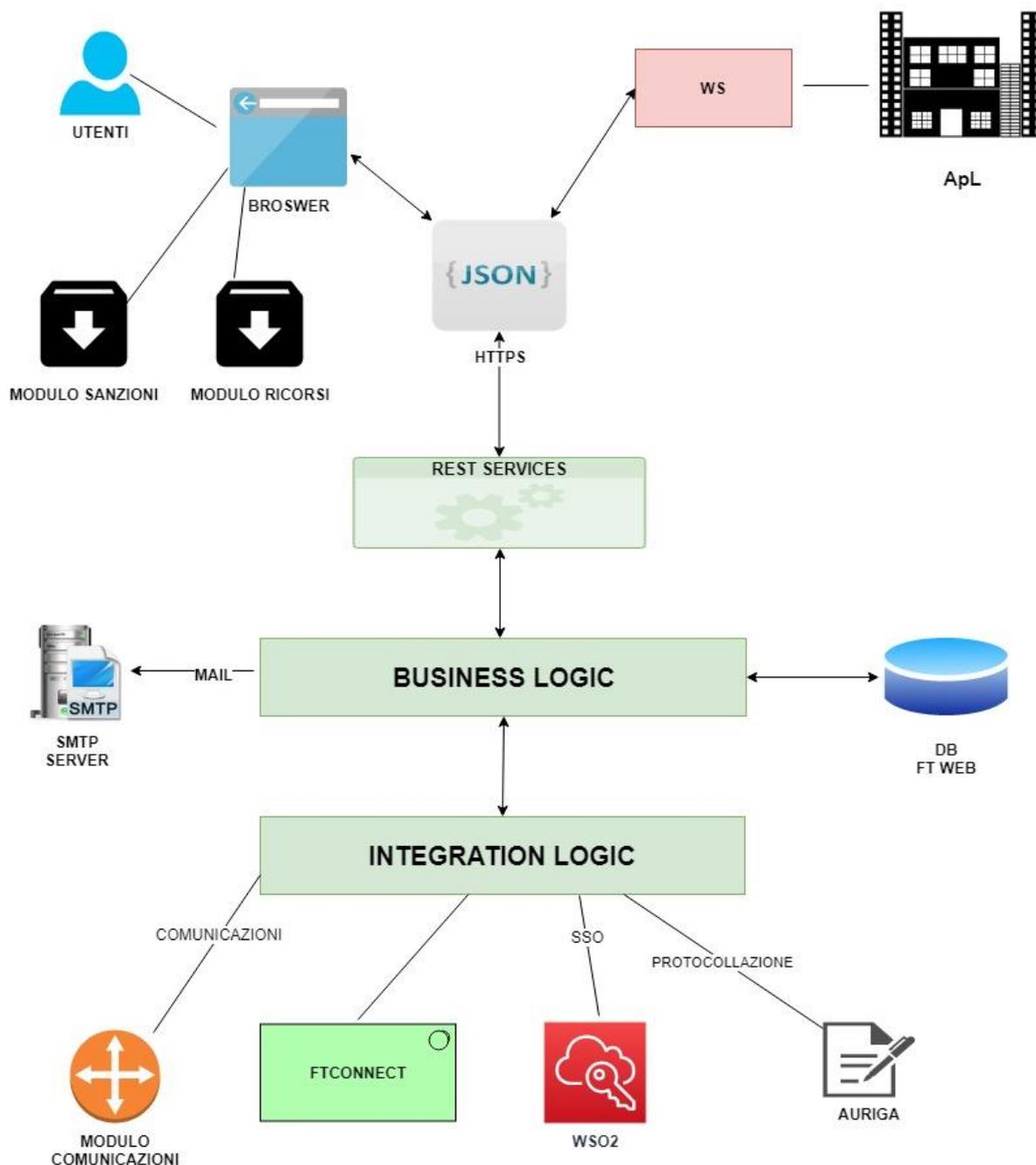


Figura 2 – Architettura logica della soluzione

3.1.1 Strato di Presentazione

Le interfacce che costituiscono il *front-end* atto a presentare i servizi applicativi e dunque destinate all'interazione con gli utenti finali sono dotate di caratteristiche di responsività (*responsive web design – RWD*) e piena compatibilità con i browser moderni, utilizzano servizi RESTful per disaccoppiare in modo completo la logica di presentazione da quella di accesso ai dati. Requisiti garantiti dall'utilizzo del framework di front-end (Angular) lato client.

Il sistema FTWeb utilizza il *web server* Apache, in ragione dello *stack* architetturale prescelto.

3.1.1.1 Servizi REST

Il sistema espone dei servizi REST che utilizzano strutture dati in formato JSON (stringhe di informazioni) per l'interscambio dei dati nelle sessioni di browsing verso gli utenti e per l'interlocuzione con i servizi offerti in modo diretto alle ApL.

Il JSON in input viene prodotto:

- da client REST, per quanto riguarda i servizi di interazione con applicativi esterni (Web services delle ApL, ecc..)
- dal framework Angular 6, per le sessioni di navigazione ed utilizzo della piattaforma via browser sui client degli utenti

L'utilizzo di Angular permette alle applicazioni di essere eseguite interamente dal web browser, previo download da parte del web server. L'elaborazione è dunque interamente *client-side*, e si evidenzia dunque il conseguente risparmio in termini prestazionali poichè non risulta necessario reinviare la pagina web al *web server Apache* implementato, in funzione di ogni richiesta di azione da parte dell'utente. Il codice generato da Angular è correttamente gestito da tutti i principali web browser moderni.

Angular inoltre, in combinazione con il toolkit open source Bootstrap, garantisce la full responsivity degli applicativi: il design del sito web, in funzione delle geometrie e delle soluzioni grafiche adottate, si adatta alle dimensioni del dispositivo utilizzato

A partire dai dati immessi in input dagli utenti e/o servizi client, viene quindi generato un JSON.

Il JSON prodotto contiene l'intero flusso informativo oggetto di elaborazione.

Complementarmente, i dati in output offerti dal server per soddisfare le richieste dei client vengono elaborati a partire dai dati forniti dal server.

Il *parser* JAX-RS dedicato si occupa infatti di generare la stringa JSON che verrà resa intellegibile in interfaccia ad uso degli utenti.

3.1.1.2 *Sistemi di controllo degli input*

Al fine di gestire i campi di input offerti nelle interfacce ad uso degli utenti del nuovo sistema FTWEB, sono stati implementati dei sistemi di controllo atti a mitigare potenziali rischi di utilizzo malevolo del sistema.

Indipendentemente dal client chiamante, viene effettuata una prima chiamata ad un apposito servizio utile a profilare l'utente ed a generare il relativo token di sicurezza; tale token dovrà essere inviato dal client al server ad ogni interlocuzione successiva, al fine di garantire l'identità del chiamante.

Tutti i dati ricevuti in input dal sistema sono sottoposti a verifica di sicurezza da parte del sistema; nello specifico, vengono effettuate opportune operazioni di codifica al fine di evitare potenziali attacchi di tipo XSS – *cross site scripting/SQL Injection*; operazioni di successiva decodifica rendono comunque intellegibile l'informazione al momento dell'eventuale nuova presentazione in interfaccia.

3.1.2 **Business Logic e persistenza dei dati**

Lo strato logico di business logic si occupa di eseguire le operazioni funzionali sottese ai processi implementati e, complementariamente, di gestire la persistenza e la consistenza dei dati utilizzati nei flussi.

3.1.2.1 *Business Logic*

La Business Logic è ospitata ed eseguita interamente dagli *application server* WildFly JBOSS.

L'elaborazione è dunque *server side* poiché JBOSS funge da *middleware* in funzione delle richieste pervenute dal client (attraverso il proprio web browser) e della necessaria interfaccia con la parte dati

A livello di Business Logic vengono inoltre effettuate le chiamate al SMTP server per la canalizzazione delle notifiche email previste dai processi implementati in FTWeb. Tale funzione verrà comunque sostituita in ragione dell'implementazione del c.d. Modulo Comunicazioni, la cui gestione, come dettagliato nei paragrafi successivi, sarà demandata allo strato di Integration Logic.

3.1.2.2 *Persistenza dei dati*

Per quanto concerne la persistenza e la consistenza dei dati nei flussi gestiti dai server, l'obiettivo è demandato al *framework* Hibernate versione 5.3.1, implementante le Java Persistence API (JPA 2.1).

Esiste un unico utente che può accedere alla base dati, tale utente è configurato all'interno dell'application server wildfly e possiede tutte le grant necessarie per portare a termine le operazioni che è possibile eseguire da ftweb.

Attraverso il *pattern* architetturale DAO (*Data Access Object*), vengono quindi rese disponibili classi (con relativi metodi) che rappresentano ognuna N entità tabellari di *database*, al fine di stratificare e isolare l'accesso alle tabelle tramite query (poste all'interno dei metodi delle classi), e dunque al layer dati da parte della Business Logic. Questa scelta architetturale crea un elevato livello di astrazione e, di conseguenza, una più facile manutenibilità. I metodi del DAO, con le rispettive query, verranno così richiamati dalle classi della Business Logic, fornendo un *mapping* (1 DAO a N entità tabellari) delle classi Java nelle tabelle del database relazionale.

Complementarmente inoltre, la scelta architetturale permette di gestire il reperimento degli oggetti dal database, generando ed eseguendo automaticamente le query SQL necessarie al recupero del flusso informativo e la successiva nuova istanza dell'oggetto precedentemente mappato sul database.

Attraverso la generazione delle chiamate SQL da parte del framework, vengono rese non necessarie implementazioni di recupero e conversione "manuali" dei dati, e viene così garantita la massima portabilità in tutti i database SQL.

3.1.3 Integration Logic

Lo strato Integration Logic è lo strato applicativo al quale è demandata la preparazione e l'interlocuzione tra i servizi di FTWeb e i servizi esterni utilizzati dal sistema per completare i processi.

In questo strato sono gestite le chiamate dirette ai seguenti sistemi:

- sistema di protocollo/documentale Auriga
 - esposto in modalità SaaS
- sistema di autenticazione WSO2.

Le chiamate vengono effettuate attraverso l'utilizzo del protocollo SOAP (*Simple Object Access Protocol*).

L'utilizzo di SOAP permette dunque lo scambio di pacchetti informativi, in forma di "messaggi", per la gestione di richieste provenienti da sistemi necessitanti interazione a livello di processo.

A livello di struttura, si illustra di seguito un'esemplificazione di un messaggio SOAP

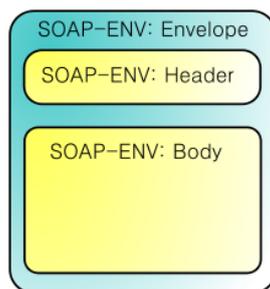


Figura 3 - Messaggio SOAP

3.1.3.1 Interazione con Auriga

L'interazione con Auriga permette al sistema FTWeb di ottenere nuovi protocolli per documenti da caricarsi ex novo a sistema, o di integrare un protocollo già esistente con ulteriore documentazione.

Lo *stream* dati, opportunamente configurato, permette quindi ai sistemi dialoganti di identificare il documento oggetto di elaborazione e di gestirlo conformemente alle specifiche necessità.