

APPENDICE 2 AL CAPITOLATO TECNICO

Descrizione architettuale del sistema SPRING



INDICE

1	SCOPO DEL DOCUMENTO	3
1.1	RIUSO DELLA SOLUZIONE	3
2	IL SISTEMA SPRING	4
2.1	PORTABILITÀ PER PIATTAFORMA	4
2.2	PORTABILITÀ PER CARATTERISTICHE FUNZIONALI	4
3	ARCHITETTURA INFORMATICA DI SPRING	6
3.1	SISTEMA GESTIONALE	6
3.1.1	Front end	6
	• STRUTS	7
3.1.2	Back end	7
3.2	PRESENTATION	8
3.2.1	Controller	8
3.2.2	View	8
3.2.3	JavaScript	9
3.2.4	AJAX	9
3.3	BUSINESS	9
3.3.1	Transaction Manager	9
3.3.2	Security Manager	9
3.3.3	Model	10
3.3.4	Dominio	10
3.3.5	Servizi funzionali	10
3.4	DATA ACCESS.....	10
3.4.1	DAO	10
3.5	SERVICE	11
3.5.1	Database di sistema.....	11
3.5.2	Service Locator	11
3.5.3	Tasker	11
3.5.4	Scheduler	12
3.6	B2B	12
3.6.1	Comunicazione timbrature.....	12
3.7	DATA	13
3.7.1	Database	13
3.7.2	Base dati SPRING	13
3.8	MOTORE DI WORKFLOW	14
4	ARCHITETTURA SISTEMISTICA DI SPRING	15
4.1	CONFIGURAZIONE STAND ALONE	15
4.2	CONFIGURAZIONE MEF	15



1 SCOPO DEL DOCUMENTO

Il presente documento descrive l'architettura del sistema SPRING, ovvero definisce la struttura della soluzione tecnica studiata per incontrare i requisiti e le esigenze funzionali del sistema di rilevazione presenze del personale.

SPRING ha come scopo principale quello di fornire a tutti gli uffici del Ministero dell'Economia e delle Finanze, a quelli di un Dipartimento del Ministero per lo Sviluppo Economico e a quelli della Corte dei conti uno strumento di supporto per la gestione dei processi amministrativi riguardanti il personale.

Il sistema è diffuso a tutti gli uffici (foglie terminali dell'organigramma) delle Amministrazioni sopra indicate, compresi gli uffici periferici (Ragionerie Territoriali dello Stato e Commissioni Tributarie), per un totale di circa 2.500 utenze di cui attive almeno 2.000. Le anagrafiche del personale gestite sono circa 20.000.

Le postazioni degli uffici centrali del MEF sono connesse al SIAP tramite rete LAN; tutte le altre postazioni sono connesse tramite rete geografica (SPC, Sistema Pubblico di Connettività).

1.1 RIUSO DELLA SOLUZIONE

Il sistema SPRING è stato adottato anche da altre importanti realtà della PA in ottica di riuso del SW e contenimento dei costi di sviluppo:

- Corte dei conti;
- DPS del MiSE.

La Corte dei conti ha installato il sistema presso il proprio CED, realizzando tre ambienti completi, di esercizio, manutenzione e collaudo.

Nel caso del DPS- MiSE, invece, il sistema è ospitato sui server del MEF e prevede due ambienti, esercizio e collaudo; le postazioni degli uffici del DPS sono connesse tramite la rete Infranet del Sistema Pubblico di Connettività.

Si precisa, tuttavia, che la configurazione sistemistica descritta nel seguito del documento fa diretto riferimento all'istanza del MEF. Le istanze Cdc e DPS, pur eseguendo la stessa versione del SW sono configurate in maniera leggermente diversa visto il numero molto ridotto di utenti rispetto al MEF.

Consip si riserva di variare tali ambienti, di aggiornare i prodotti software, come pure di definire nuove modalità di utilizzo dei sistemi.



2 IL SISTEMA SPRING

2.1 PORTABILITÀ PER PIATTAFORMA

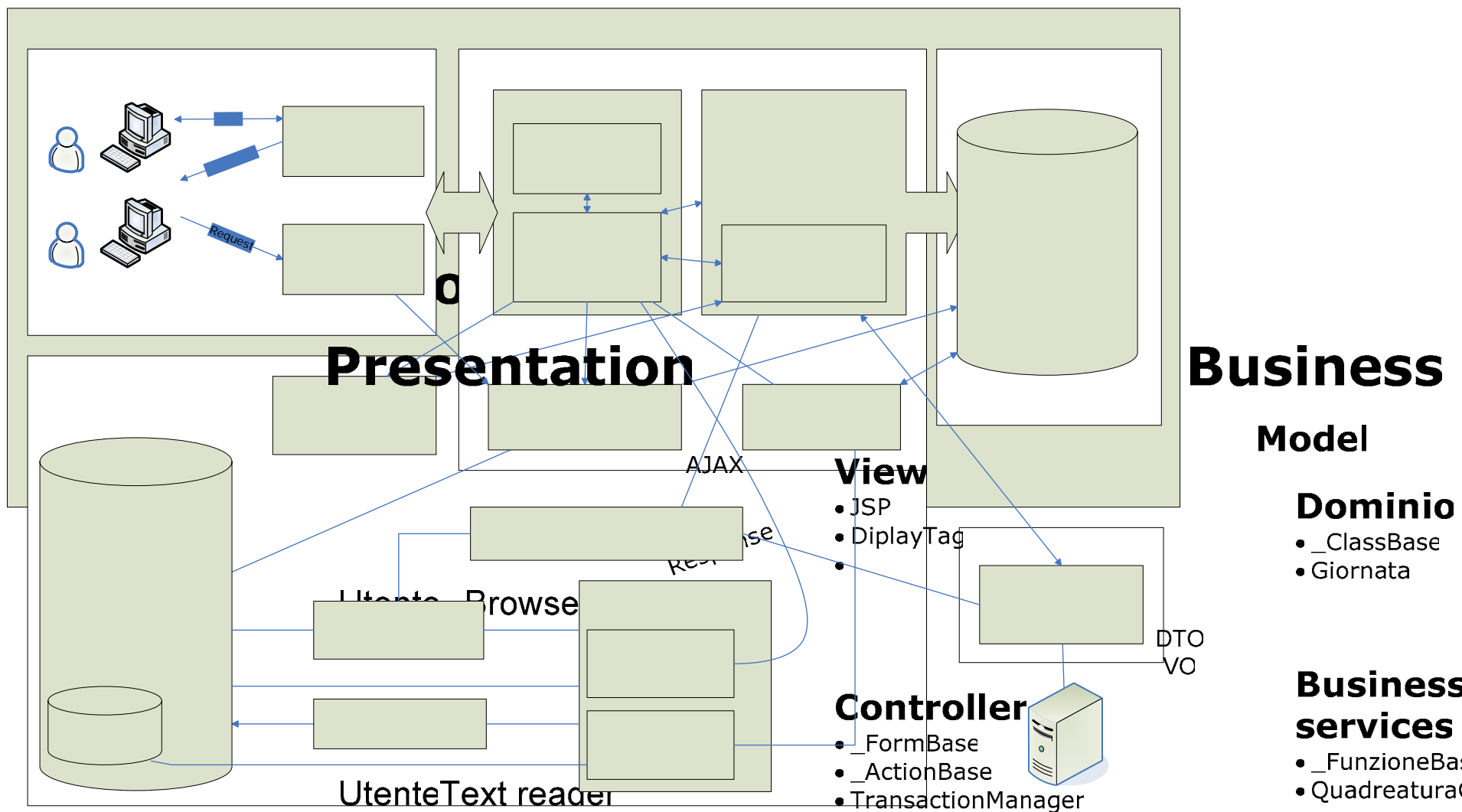
Per assecondare l'esigenza della portabilità si sceglie la tecnologia J2EE che è da anni quella meno soggetta a vincoli, supportata da tutti i SO server sia di livello main stream che free, implementata da parecchi vendor di application server anche questi sia di fascia alta che gratuiti.

2.2 PORTABILITÀ PER CARATTERISTICHE FUNZIONALI

Per perseguire le necessità caratteristiche di ogni Amministrazione, si punta su un sistema interamente configurabile le cui impostazioni sono gestibili attraverso il sistema stesso da personale non tecnico.

In questo senso la realizzazione è stata concepita fin dall'origine come autosufficiente, senza nulla presumere sull'ambiente informatico e amministrativo che la ospita. Quindi questa infrastruttura realizza ad esempio anche funzionalità quali anagrafe personale e gestione utenze. Questa base si definisce "Configurazione autonoma".

La prima customizzazione consiste nella "Configurazione MEF" che prevede in termini di configurazione funzionale l'integrazione con i sistemi esistenti nel MEF. Ad esempio integra la gestione dell'anagrafe personale di HR e la gestione delle utenze di LDAP SSO istanziate presso il MEF.





3 ARCHITETTURA INFORMATICA DI SPRING

Il sistema SPRING, per la parte gestionale, si compone di tre livelli: *Presentation*, *Business Logic* e *Data layer*, trattati nei prossimi paragrafi.

La collaborazione con altri sistemi avviene tramite Web Services ed è trattata nel paragrafo B2B.

La sezione Service tratta l'insieme di SW a supporto del sistema.

3.1 SISTEMA GESTIONALE

Concettualmente l'infrastruttura si articola in:

- un frontend che interfaccia il sistema agli utenti;
- un backend che contiene:
 - logica funzionale interna,
 - logica di integrazione della configurazione,
 - interfacciamento con altri sistemi come le timbrature,
 - batch processing di stampe e calcoli;
- un database relazionale per la persistenza dei dati.

Questa struttura concettuale è realizzata in architettura three tier che prevede la suddivisione del sistema in tre diversi moduli dedicati rispettivamente all'interfaccia utente (presentation), alla logica funzionale (business logic) ed alla gestione dei dati persistenti (data layer), con lo scopo di disaccoppiare i livelli applicativi rendendoli indipendenti, flessibili e manutenibili.

3.1.1 Front end

Il presentation layer si occupa della costruzione dinamica dell'interfaccia utente.

Il primo requisito prevede che siano prodotte interfacce web con html accessibile.

Su numeri di migliaia di utenti contemporanei l'approccio alla produzione delle pagine è vincolato dalla scalabilità. Dopo aver vagliato diverse opportunità la scelta è ricaduta sul framework open *Struts* di Jakarta. Oltre all'estrema scalabilità, *Struts* offre diversi vantaggi:

- implementa il l'approccio architetturale MVC, realizza quindi una buona separazione fra interfaccia e logica applicativa;
- assegna a differenti strutture dei compiti specifici permettendo di incapsulare la complessità dell'implementazione e di modulare la realizzazione: JSP per la visualizzazione delle informazioni, Form Bean per il model e Servlet Action con *struts-config.xml* per governare il flusso di navigazione;
- propone una serie di tag library per la prospettazione delle informazioni e permette una verifica formale preventiva dei dati immessi dall'utente;
- permette la modifica del layout con impatti ridotti o nulli sul back end ed agevola, fra l'altro, ricicli su forma e accessibilità dell'interfaccia.

Nel presentation layer sarà presente una struttura deputata alla gestione della sicurezza, collegata ad una omologa struttura del business layer, che risponde in modo centralizzato alle richieste dell'interfaccia legate all'abilitazione delle funzioni fino a specifici campi in base all'utente collegato.



Saranno, inoltre, presenti delle classi, suddivise per funzione, che assicurano le abilitazioni dell'interfaccia rispetto alle specifiche regole semantiche dell'informazione.

- **STRUTS**

Jakarta Struts è un MVC web application framework, ovvero è un framework per lo sviluppo di applicazioni web J2EE basato sul pattern Model-View-Controller il quale consente di disaccoppiare i diversi componenti dell'applicazione in base al loro ruolo nell'architettura per ottenere vantaggi in termini di riusabilità e manutenibilità.

In questo modello i ruoli di presentation, control e business logic vengono affidati a componenti diversi e sono tra di loro disaccoppiati, con evidenti vantaggi in termini di riusabilità, manutenibilità, estensibilità e modularità. In un'applicazione costruita secondo il pattern MVC si possono quindi individuare tre livelli logici ben distinti che molto schematicamente svolgono i seguenti compiti:

- *controller*: determina il modo in cui l'applicazione risponde agli input dell'utente; esamina le richieste dei client, estrae i parametri della richiesta e li convalida, si interfaccia con lo strato di business logic dell'applicazione; sceglie la successiva vista da fornire all'utente al termine dell'elaborazione.
- *model*: contiene i dati visualizzati dalle viste; è ciò che viene elaborato e successivamente presentato all'utente;
- *view*: visualizza all'utente i dati contenuti nel model; è la rappresentazione dello stato corrente del Model.

3.1.2 Back end

Nel backend SPRING si collocano i moduli di:

- business logic che ingloba ed assicura il rispetto delle regole funzionali del sistema;
- data access per la comunicazione con la base dati;
- job server per la schedulazione e la gestione dei batch di calcolo, stampe ed ETL;
- web service per l'interfacciamento non ETL con altri sistemi.

La business logic sarà costituita da una struttura di classi che compongono il modello del dominio funzionale, il cui stato costituisce l'informazione ed i cui metodi realizzano l'applicazione delle regole.

Il modulo data access si occupa della gestione della base dati, è finalizzato al disaccoppiamento fra db e business; sarà impiegato per la tracciatura dell'attività utente.

SPRING prevede alcuni batch di calcolo caratterizzati da un uso intensivo e massivo del database. Per evitare una sovrapposizione con il gestionale che possa deprimere le performance della componente on-line, si disegna un modulo concettualmente distinto per queste funzionalità a cui saranno demandati anche la gestione dei report e della schedulazione e programmazione attività. In base all'esigenze d'esercizio questo modulo potrà essere dislocato su hardware distinto da quello che ospita il web server.

Per le comunicazioni B2B si prevede un ulteriore modulo per la pubblicazione di servizi su cui al momento è previsto il solo servizio di acquisizione timbrature. La forma di Web Services (non PDD) garantisce la comunicazione attraverso internet, necessaria alle piccole Amministrazioni che intendano ostare il sistema presso strutture esterne alla intranet.



3.2 PRESENTATION

Il layer Presentation gestisce il colloquio con l'utente ed è realizzato con due componenti:

- il Controller che si occupa della lettura ed interpretazione dell'input dell'utente;
- il View che si occupa della presentazione delle informazioni richieste all'utente.

Queste componenti comunicano con l'utente attraverso il Web Browser tramite il protocollo HTTP.

L'HTML prodotto trasmesso all'utente rispetta le regole di accessibilità previste dalla legge Stanca.

Per agevolare l'uso del programma sono impiegate delle tecniche, supportate dalla maggior parte dei browser, che sfruttano JavaScript (un linguaggio lato browser) ed AJAX (HTTP 1.1).

Le funzioni utente hanno generalmente un approccio transazionale tutto o niente, e persistono i dati indicati dall'utente in blocchi informativi che lasciano il sistema in uno stato di coerenza funzionale. Il modulo gestionale non è session safe, ovvero in situazioni di inserimento intermedie non confermate dall'utente con esplicito salvataggio, vanno perse in caso di eccezioni di sistema.

3.2.1 Controller

L'input dell'utente è gestito dal Controller che interpreta, smista e lavora le chiamate HTTP.

Il controller di Struts:

- legge i dati presenti in maschera e li inserisce in apposite proprietà dei vari FormBean;
 - innesca l'Action che esegue le attività preliminari di validazione ed interpretazione dell'attività utente;
 - verifica i diritti dell'utente di eseguire la richiesta tramite il Security Manager;
 - l'Action alternativamente:
 - reindirizza il controllo del flusso applicativo verso un altro modulo tramite un Forward;
- oppure
- procede con la gestione della chiamata richiedendo servizi business tramite Business Delegate e Data Transfer Object.

3.2.2 View

Il View si occupa della costruzione dell'interfaccia utente. La produzione di HTML è realizzata tramite JSP popolato dalle Tag Libraries di Struts che reindirizzano il contenuto dei View Object.

Per la produzione di liste è utilizzata la Tag Library Display Tag che gestisce, fra l'altro, paginazione ed ordinamento ed offre possibilità reportistiche nei formati PDF, CSV ed Excel.

Le pagine web prodotte sono rese accessibili, nel rispetto della legge Stanca, attraverso opportuna integrazione custom dell'HTML generato dalla Tag Libraries.



3.2.3 JavaScript

Il JavaScript (un linguaggio di scripting supportato dalla maggior parte dei browser) è impiegato per migliorare l'usabilità dell'interfaccia utente.

L'attività dell'utente è agevolata tramite degli script che realizzano attività quali la validazione dei dati, la messaggistica, l'autocompletamento delle informazioni, la selezione automatica di valori e pulsanti. Tali script sono eseguiti sul browser evitando il colloquio request response HTTP con il server, rendendo più rapida la risposta applicativa.

Tutte le funzionalità applicative client sono implementate anche lato server in modo di conseguire le medesime funzionalità nel caso di browser che non supportano o che hanno disabilitato JavaScript.

L'impiego di JavaScript è armonizzato con l'accessibilità e consente l'uso di Text Reader ed altri dispositivi in tecnologia assistiva.

3.2.4 AJAX

Nell'ottica dell'usabilità l'impiego di AJAX (tecnologia XML HTTP1.1 JS) permette la gestione interattiva tipica dei programmi client/server evitando le lungaggini del ciclo di richiesta HTTP.

AJAX è impiegato in situazioni specifiche su funzioni che hanno particolari criticità su performance ed usabilità.

AJAX è impiegato in modo adeguato all'accessibilità e non preclude il funzionamento applicativo su browser che non lo supportano.

3.3 BUSINESS

3.3.1 Transaction Manager

Il Transaction Manager offre dei servizi per il controllo di una transazione, espone funzionalità quali:

- inizializzazione di una transazione;
- reperimento e rilascio di connessione al data base;
- conferma ed annullamento della transazione.

Il TM è utilizzato dai Function o direttamente dai Business Delegate.

3.3.2 Security Manager

Il Security Manager espone servizi informativi che indicano la possibilità di utilizzo degli oggetti funzionali da parte dell'utente o di un client più in generale.

Il SM fornisce:

- l'informazione di autenticazione dell'utente;
- l'elenco delle Responsabilità assegnate all'utente;
- i filtri di visibilità del partizionamento dati legato ad una Responsabilità;
- il menù e l'elenco delle funzioni logiche abilitate all'utente/Responsabilità.



3.3.3 Model

Il modello logico di SPRING si colloca nel Business layer e contiene la struttura informativa e le regole funzionali.

3.3.4 Dominio

Nel dominio sono presenti le classi che rappresentano il modello informativo. Le classi di tipo POJO estendono una classe base `_Dominio`. Ogni classe del dominio è associata ad un omologo oggetto del Dao che ne incapsula la logica di persistenza.

- `_Dominio (POJO Base)` ← Giornata
- `_Business (use TM)` ← Quadratura Giornata.

3.3.5 Servizi funzionali

I Servizi funzionali contengono le funzioni per la manipolazione dei dati che garantiscono la coerenza logica delle informazioni. Ogni metodo al termine dell'esecuzione lascia il sistema in uno stato di coerenza funzionale. Ogni SF conosce ed utilizza gli oggetti di dominio.

I SF estendono la classe base `_Funzione` che offre il servizio di supporto transazione. Per ogni SF esiste un corrispondente oggetto Dao, SF usa più oggetti di Dominio, richiama i servizi di persistenza Dao e coordina la transazione.

3.4 DATA ACCESS

Per l'accesso ai dati sono impiegate le tecnologie JDBC, SQLJ, Hibernate e EJB3.

3.4.1 DAO

Il Data Access Object disaccoppia i livelli Business e Data, esponendo servizi business di persistenza ed incapsulando le modalità di accesso ai dati.

La classe `_DAO` integra il supporto per connessione `ConnectionManager` e transazione `TransationManager`. Per ogni oggetto di dominio esiste un corrispondente DAO (che estende il DAO base) che ne realizza la persistenza.

Per ciascun Servizio funzionale esiste un corrispondente DAO che opera la manipolazione dati opportuna.

- `_Daobase (use CM)` ← `GiornataDao` ← `QuadGiorDao`

I DAO accedono direttamente al database via JDBC o tramite SQLJ e non contengono logica di Business. Pertanto, il DAO è l'unico oggetto Java coinvolto nel caso fosse necessario migrare verso diverse tipologie di database.



3.5 SERVICE

I servizi tecnici sono finalizzati alla gestione informatica del sistema e sono concettualmente raggruppati in un insieme eterogeneo detto Service.

3.5.1 Database di sistema

Nel database di sistema si collocano la configurazione sistemistica ed informatica del sistema, necessarie al perseguimento della completa configurabilità di SPRING, e tutte le informazioni non funzionali alla tematica applicativa.

Il DB di sistema contiene:

- *configurazione sistemistica*: determina su quali hardware sono disponibili quali servizi;
- *configurazione funzionale*: mantiene l'elenco delle funzioni logiche attive deploy-ate sul sistema e la relativa versione;
- *sicurezza*: contiene le utenze funzionali e tecniche con le relative autorizzazioni;
- *log attività*: mantiene il log delle attività che l'utente funzionale e di sistema svolge sul sistema;
- *tasks*: archivia i Task funzionali (calcolo e stampe) e di sistema (backup) con relative schedulazioni e status;
- *report repository*: archivio dei documenti prodotti a fronte di richieste utente non interattive (sincrone).

3.5.2 Service Locator

Il Service locator offre l'individuazione centralizzata dei servizi di sistema e ne incapsula la complessità legata alla configurazione sistemistica.

3.5.3 Tasker

Il Tasker è la coppia oggetto/servizio (Task/TaskManager) per la gestione delle attività utente e di sistema. Contiene tutte le informazioni necessarie alla passivazione/archiviazione ed esecuzione di una unità di lavoro.

Il TaskManager accetta richieste con modalità quali Immediato/Differito/Schedulato/Sincrono/Asincrono, opera la schedulazione e l'esecuzione del task restituendo alternativamente direttamente il risultato o il ticket per la verifica dello stato d'avanzamento e per il reperimento dell'eventuale prodotto (es. Report).

In base alla configurazione, il TaskManager sceglie su quale server/servizio elaborare il job, ad esempio sul server del gestionale i task immediati e su un batch server quelli differiti o schedulati.

Computation/ComputationManager estende il Tasker e gestisce le operazioni di calcolo richieste dall'utente; permette l'archiviazione, l'esecuzione, la passivazione ed il riavvio dopo un'esecuzione parziale. Permette, inoltre, la gestione del risultato di operazioni computazionali quali QuadraturaGiornata e QuadraturaMese.

Report/ReportManager gestisce la produzione di report e può restituire il documento prodotto direttamente al chiamante o archivarlo nel report repository per restituirlo in seguito. I report sono generati tramite le tecnologie JasperReports, con accesso diretto alla basedati, e DisplayTag con accesso mediato dal backend del gestionale.



3.5.4 Scheduler

Lo Scheduler è il sottosistema di programmazione di attività. Invocato dal TaskManager, lo Scheduler memorizza il Task per notificare in seguito allo stesso TaskManager l'avvenimento dell'evento.

3.6 B2B

La comunicazione con i sistemi di confine è gestita tramite Web Services con accesso diretto alla base dati.

3.6.1 Comunicazione timbrature

La collaborazione applicativa realizza la comunicazione delle timbrature da parte del sistema di rilevazione ingressi tramite due Web Service ed una procedura di calcolo.

- (a) Il sistema di Rilevazione Ingressi richiama il WS di accettazione timbrature e gli passa un elenco di record timbratura.
- (b) Il WS Timbrature:
 - 1. legge il messaggio di elenco timbrature;
 - 2. scrive il messaggio in una tabella di timbrature da lavorare, nel data base SPRING, in un campo XML;
 - 3. lancia in modo asincrono la procedura di acquisizione;
 - 4. restituisce il risultato al chiamante rilasciando un ticket di chiamata.
- (c) La procedura di acquisizione timbrature:
 - 1. interpreta il file delle timbrature;
 - 2. smista le timbrature corrette nell'acquisizione e scarta le altre.
- (d) Il sistema Rilevazione Ingressi richiama il WS Scarti con un ticket di chiamata.
- (e) Il WS Scarti risponde alternativamente con lo stato non definito dell'acquisizione, ovvero con l'elenco degli scarti.



3.7 DATA

Il data layer è deputato a gestire la persistenza dei dati.

Il livello data è disaccoppiato dal business dallo strato Data Access (DAO/EJB) che incapsula le istruzioni SQL per la manipolazione dati.

In caso di configurazioni che prevedano l'uso di un motore diverso da Oracle, l'intervento di adattamento è circoscritto allo strato DAO e non coinvolge gli altri moduli del sistema.

Nella prospettiva della portabilità, in fase di sviluppo l'uso di caratteristiche proprie di Oracle non portabili su altri DB (ad esempio PL/SQL) è stato completamente evitato.

3.7.1 Database

Il motore RDBMS di SPRING per le configurazioni *stand alone* e MEF (ovvero integrata con il SIAP) è Oracle Database 10g. Il DB è utilizzato principalmente nelle sue funzionalità di archiviazione ed organizzazione dati.

3.7.2 Base dati SPRING

Il database SPRING in configurazione *stand alone* archivia su basi dati concettualmente distinte:

- tutte le informazioni tematiche del sistema di rilevazione presenze quali Timbrature e Giuridico insieme a quelle di struttura organizzativa dell'Amministrazione;
- tutti i dati di configurazione informatica come sicurezza e setup sistemistico.

Nella configurazione MEF sullo stesso database server:

- su uno schema SPRING sono ospitate le tabelle per la parte tematica;
- uno schema SPRINGSys ospita la configurazione sistemistica;
- uno schema SPRINGMef contiene le strutture dati ed oggetti (tabelle, viste, sinonimi, ecc.) per l'integrazione con il SIAP.



3.8 MOTORE DI WORKFLOW

Il porting di alcuni processi del sistema SIAP in SPRING, attualmente in corso, ha determinato la necessità di integrare SPRING con un motore di Workflow.

I processi sorgenti sono all'interno della suite delle Oracle Applications e sono attualmente in esercizio utilizzando il motore Oracle Workflow.

Nell'ambito di SPRING è stato invece deciso di utilizzare il motore Opensource "SPAGIC".

SPAGIC è una delle piattaforme di integrazione più completa tra quelle disponibili nel mondo open source, in quanto già dotata di connettori per i protocolli più diffusi, di servizi di middleware molto versatili e, soprattutto, di una console di monitoraggio che garantisce la possibilità di controllare ogni servizio e processo gestendo le ripartenze in caso di errore o di indisponibilità del canale di comunicazione.

Il modello di riferimento della piattaforma SPAGIC è riconducibile all'OSGi Universal Middleware (Open Service Gateway initiative) per l'implementazione di soluzioni XaaS altamente modulari in modo da consentire una modalità di sviluppo rivolta a fornire risposte concrete all'aumento delle complessità inerenti alle specifiche architetturali, alla riusabilità ed alla gestione del ciclo di vita di componenti di diversa natura.

I principali moduli che compongono la soluzione sono i seguenti:

- Ambiente IDE, tramite Spagic Studio, basato su Eclipse IDE, predispone gli strumenti di progettazione, sviluppo e deploy;
- Modulo di supporto all'attività utente ed eForm, tramite Spagic task list e PDF;
- Servizi e connettori, tramite Spagic OSGi Bundles;
- Modulo di gestione, tramite Spagic Monitor che rende disponibili i servizi di management e di gestione del ciclo di vita dei singoli moduli.



4 ARCHITETTURA SISTEMISTICA DI SPRING

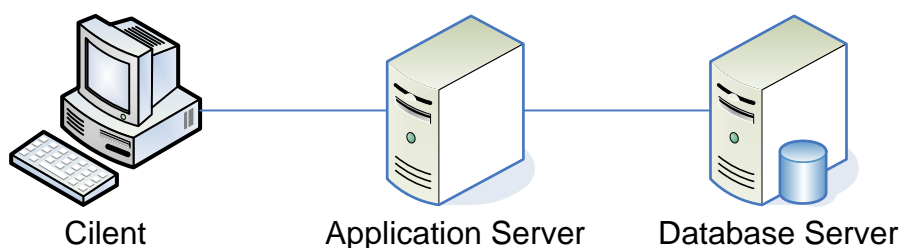
L'architettura del sistema prevede:

- un database relazionale per l'archiviazione delle informazioni;
- un application server, con supporto alle specifiche J2EE 1.4 e JDK 5, per la gestione del sistema e l'interfaccia utente;
- un job server per la gestione di elaborazioni batch di calcolo, stampa ed ETL;
- un web service per l'interfacciamento non ETL con altri sistemi.

4.1 CONFIGURAZIONE STAND ALONE

La configurazione stand alone prevede:

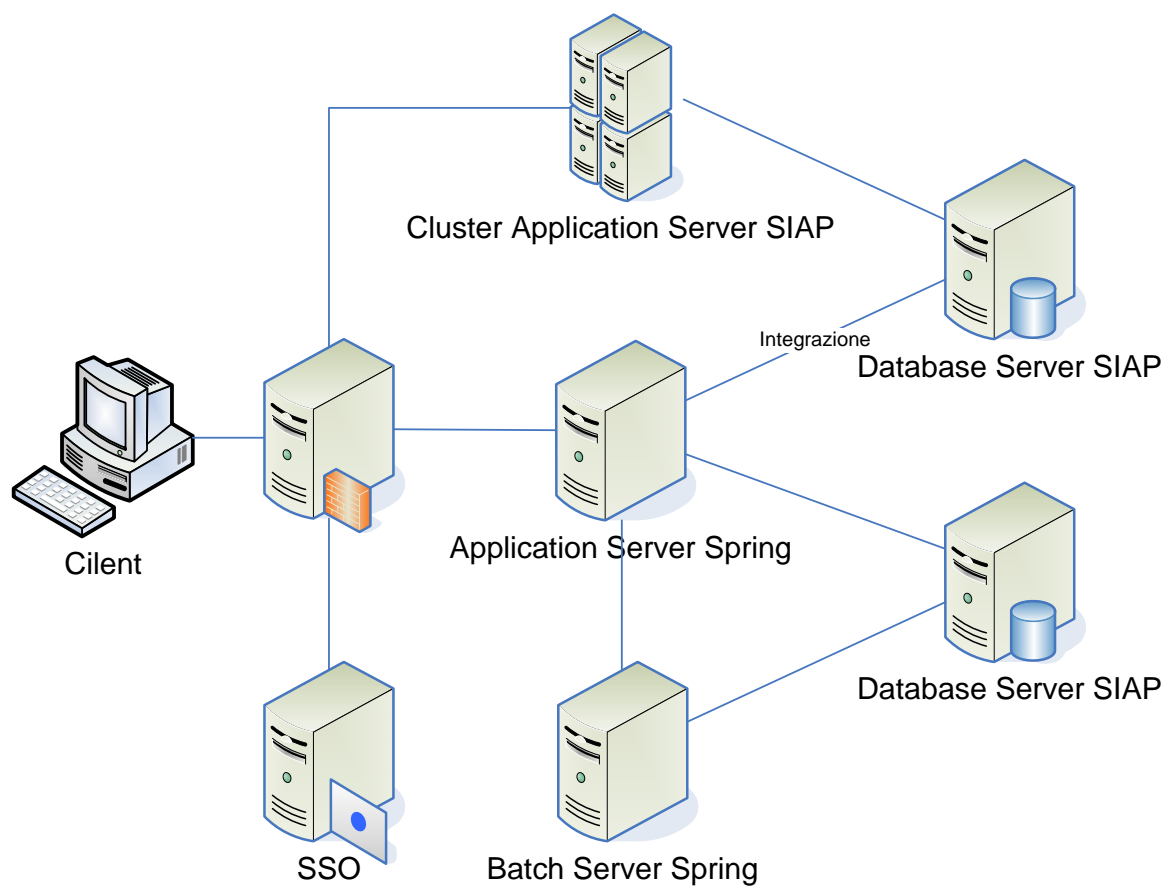
- un Application Server Oracle AS 10G v10.1.3 ;
- un Database Server Oracle DB 10G.



4.2 CONFIGURAZIONE MEF

La configurazione MEF prevede:

- un Application Server Oracle AS 10G v10.1.3 per il gestionale SPRING ed i WebServices;
- un Application Server Oracle AS 10G v10.1.3 per l'esecuzione batch di procedure di calcolo e report;
- un SSO Server per l'autenticazione degli utenti;
- un Database Server Oracle DB 10G che ospita la base informativa della rilevazione presenze di SPRING e la base dati di sistema;
- una serie di Application Server e Database che ospitano il resto del sistema SIAP/HR.



Sui due AS è installata la stessa versione del software SPRING, le attività di elaborazione sono ripartite dinamicamente fra i due server in base alle loro caratteristiche (es. on-line/batch) secondo quanto indicato nel DB di configurazione sistemistica.