



# Specifiche di Architettura Tecnica

<b>Modello documento</b>
SA_ModelloSpecificheArchitettura_v01.3.dot



## SOMMARIO

<b>1 APPROVAZIONI .....</b>	<b>3</b>
<b>2 LISTA DI DISTRIBUZIONE .....</b>	<b>3</b>
<b>3 STORIA DELLE MODIFICHE .....</b>	<b>3</b>
<b>4 RIFERIMENTI .....</b>	<b>3</b>
<b>5 COPYRIGHT .....</b>	<b>3</b>
<b>6 ACRONIMI .....</b>	<b>3</b>
<b>7 CONTESTO .....</b>	<b>4</b>
<b>8 OBIETTIVI .....</b>	<b>4</b>
8.1 INTRODUZIONE .....	4
<b>9 ARCHITETTURA DEL SISTEMA.....</b>	<b>4</b>
9.1 ARCHITETTURA APPLICATIVA .....	5
9.1.1 Scenario di riuso .....	7
9.1.2 Scenario di adesione .....	7
9.1.3 Scenario di riuso .....	8
9.1.4 Scenario di adesione .....	8
9.2 MODELLO FISICO.....	10
9.2.1 Schema fisico minimo .....	10
9.2.2 Schema fisico in alta affidabilità .....	11

## INDICE DELLE FIGURE

<b>FIGURA 1 - INTERAZIONE FRA SISTEMI E MACRO-COMPONENTI .....</b>	<b>5</b>
<b>FIGURA 2 - COMPONENTI DI SISTEMA E LIVELLI LOGICI.....</b>	<b>6</b>
<b>FIGURA 3 – RIUSO: SCHEMA LOGICO-APPLICATIVO .....</b>	<b>8</b>
<b>FIGURA 4 - ADESIONE: SCHEMA LOGICO-APPLICATIVO .....</b>	<b>9</b>
<b>FIGURA 5 - SCHEMA FISICO MINIMO .....</b>	<b>10</b>
<b>FIGURA 6 - SCHEMA FISICO IN ALTA AFFIDABILITÀ .....</b>	<b>11</b>



## 1 APPROVAZIONI

Attività	Nominativo	Azienda	Tel.	e-Mail
Redazione	Francesco Manente	Engineering		francesco.manente@eng.it
Verifica				
Approvazione				

## 2 LISTA DI DISTRIBUZIONE

Nominativo	Azienda	Tel.	e-Mail	Tipo
Antonino Mola	Regione del Veneto		antonino.mola@regione.veneto.it	CC

Tipo: CC=Copia Controllata, PC=Per conoscenza

## 3 STORIA DELLE MODIFICHE

Versione	Data	Descrizione
1.0	18/06/2012	Prima emissione
1.1	18/01/2013	Aggiornamento
2.0	30/04/2015	Aggiunti riferimenti alla compatibilità del software con il rilascio in cluster

## 4 RIFERIMENTI

N.	Titolo	Autore	Versione	Data
[1]	System Design Federa	Antonio Merighi, Massimiliano Pianciamore	1.0	22/04/2009
[2]	Specifiche funzionali del Portale dei Pagamenti	Luigi Bellio, Francesco Manente	1.0	18/06/2012
[3]	Specifiche Tecniche del Nodo dei Pagamenti-SPC ad uso sperimentazione	Alberto Carletti, Stefano Ercoli, Mauro Bracalari	1.2	06/03/2012

## 5 COPYRIGHT

Questo documento appartiene alla Regione del Veneto. I contenuti del medesimo – testi, tabelle, immagini, etc. – sono protetti ai sensi della normativa in tema di opere dell'ingegno. Tutti i diritti sono riservati. Il presente documento potrà essere utilizzato per la realizzazione di progetti regionali liberamente ed esclusivamente nel rispetto delle regole (standard) stabilite dalla Regione del Veneto. Ogni altro utilizzo, compresa la copia, distribuzione, riproduzione, traduzione in altra lingua, potrà avvenire unicamente previo consenso scritto da parte di Regione del Veneto. In nessun caso, comunque, il documento potrà essere utilizzato per fini di lucro o per trarne una qualche utilità.

## 6 ACRONIMI

Termine	Descrizione
AS	Application Server
PA	Pubblica Amministrazione
SOA	Service Oriented Architecture
UI	User Interface
WS	Web Service



## 7 CONTESTO

Questo documento descrive, nell'ambito del Progetto MyGov, la soluzione denominata Portale dei Pagamenti che implementa le specifiche e le funzionalità previste dai documenti tecnici del "NODO DEI PAGAMENTI-SPC" di DigitPA [3] oltre a consentire l'eventuale integrazione diretta di altri Prestatori di Servizi di Pagamento.

## 8 OBIETTIVI

Il principale obiettivo del progetto è di realizzare un portale che permetta ad un cittadino di consultare la propria posizione debitoria nei confronti di un ente e di effettuare un pagamento, previsto o meno nella sua posizione (pagamento spontaneo).

Un secondo obiettivo è quello di fornire una soluzione tecnologicamente flessibile, standard e riusabile che possa essere adottata in contesti diversi. Ovvero deve essere possibile rilasciare l'intera soluzione in riuso oppure la sola componente dei servizi di back-end (nodo dei pagamenti).

La soluzione proposta risponde ai seguenti requisiti:

- Front-end (interfaccia utente web) accessibile secondo la Legge Stanca
- Integrazione con il sistema di autenticazione federata Federa ([1])
- Predisposizione multiente

Per gli aspetti funzionali si fa riferimento al documento di Specifiche Funzionali del Portale dei Pagamenti ([2]).

### 8.1 Introduzione

La soluzione illustrata nel presente documento viene denominata "Piattaforma dei Pagamenti della PA" inteso come l'insieme delle componenti di back-end e/o di front-end utilizzabili dal singolo ente o dall'intermediario.

In questo specifico contesto la Regione del Veneto intende realizzare il suddetto sistema all'interno dei propri sistemi informativi, ponendosi quindi come intermediario dei servizi di pagamento fra le singole amministrazioni e il Nodo dei Pagamenti SPC.

Tuttavia la soluzione può essere utilizzata in riuso completo con entrambe le componenti così come può essere utilizzata in riuso parziale per le componenti di back-end da un ente terzo.

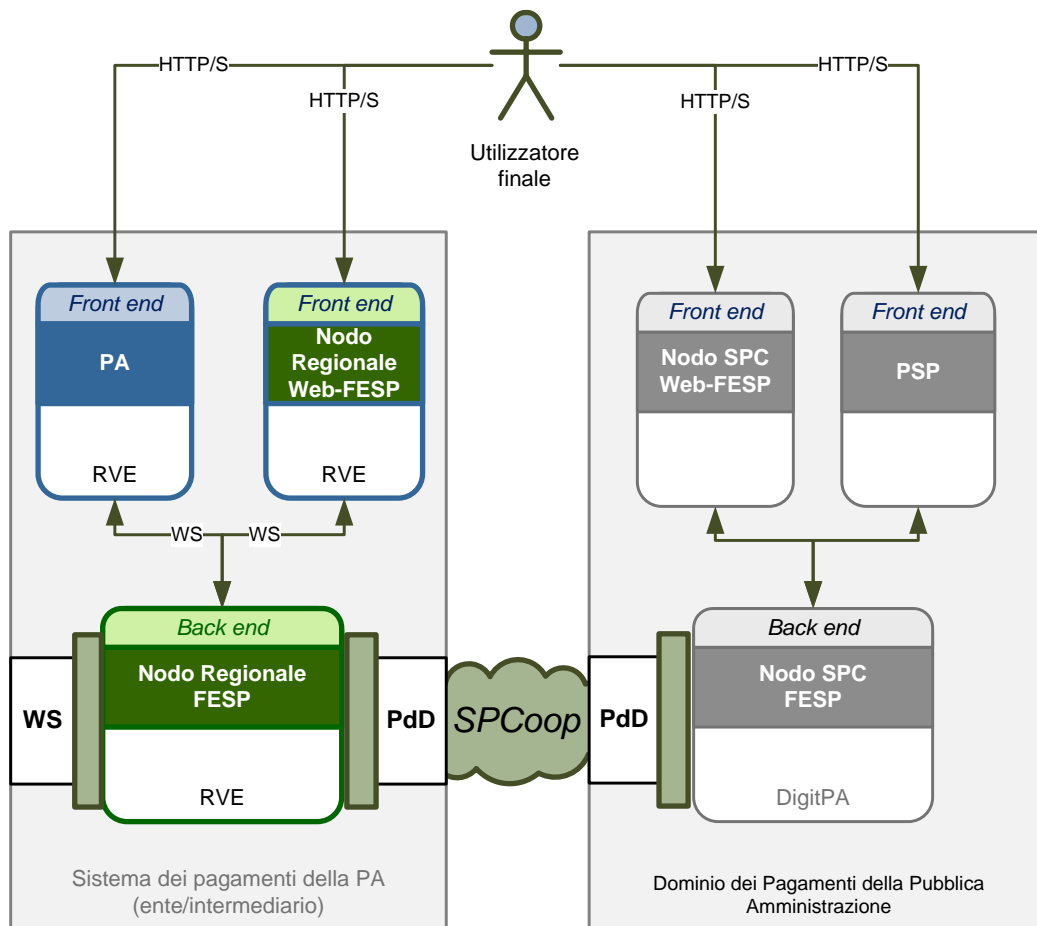
## 9 ARCHITETTURA DEL SISTEMA

La "Piattaforma dei Pagamenti della PA" si sviluppa su due livelli principali:

- Front-end (o Portale dei Pagamenti): rappresenta lo strato di "portale" ovvero l'interfaccia utente (UI web) che il cittadino o l'operatore dell'ente utilizzano per le funzionalità ad essi preposte. In questo documento le illustrazioni si riferiscono a implementazioni di UI web ma possono essere intese anche come UI su device *mobile*
- Back-end (o Nodo dei Pagamenti): rappresenta lo strato di servizi che implementano i servizi necessari al pagamento e/o alla gestione delle posizioni debitorie. Questo livello si pone come intermediario nei confronti del Nodo dei Pagamenti SPC

### 9.1 ARCHITETTURA APPLICATIVA

Si riporta un'illustrazione di interazione fra i sistemi con una rappresentazione delle macro-componenti.



**Figura 1 - Interazione fra sistemi e macro-componenti**

Il sistema riportato in figura come "Piattaforma dei Pagamenti della PA" (lato sx) rappresenta il portale dei pagamenti del singolo ente che voglia utilizzare il Nodo dei Pagamenti della DigitPA.

Tuttavia, come già esplicitato, in questo contesto progettuale in cui questo sistema viene implementato nei sistemi informativi regionali la Regione assume il ruolo di intermediario fra l'ente e il nodo della DigitPA.

Per ulteriori approfondimenti sui ruoli dei diversi sistemi si rimanda al documento [3].



Di seguito si riporta un'ulteriore illustrazione con i componenti di sistema e della loro separazione nei due livelli precedentemente descritti.

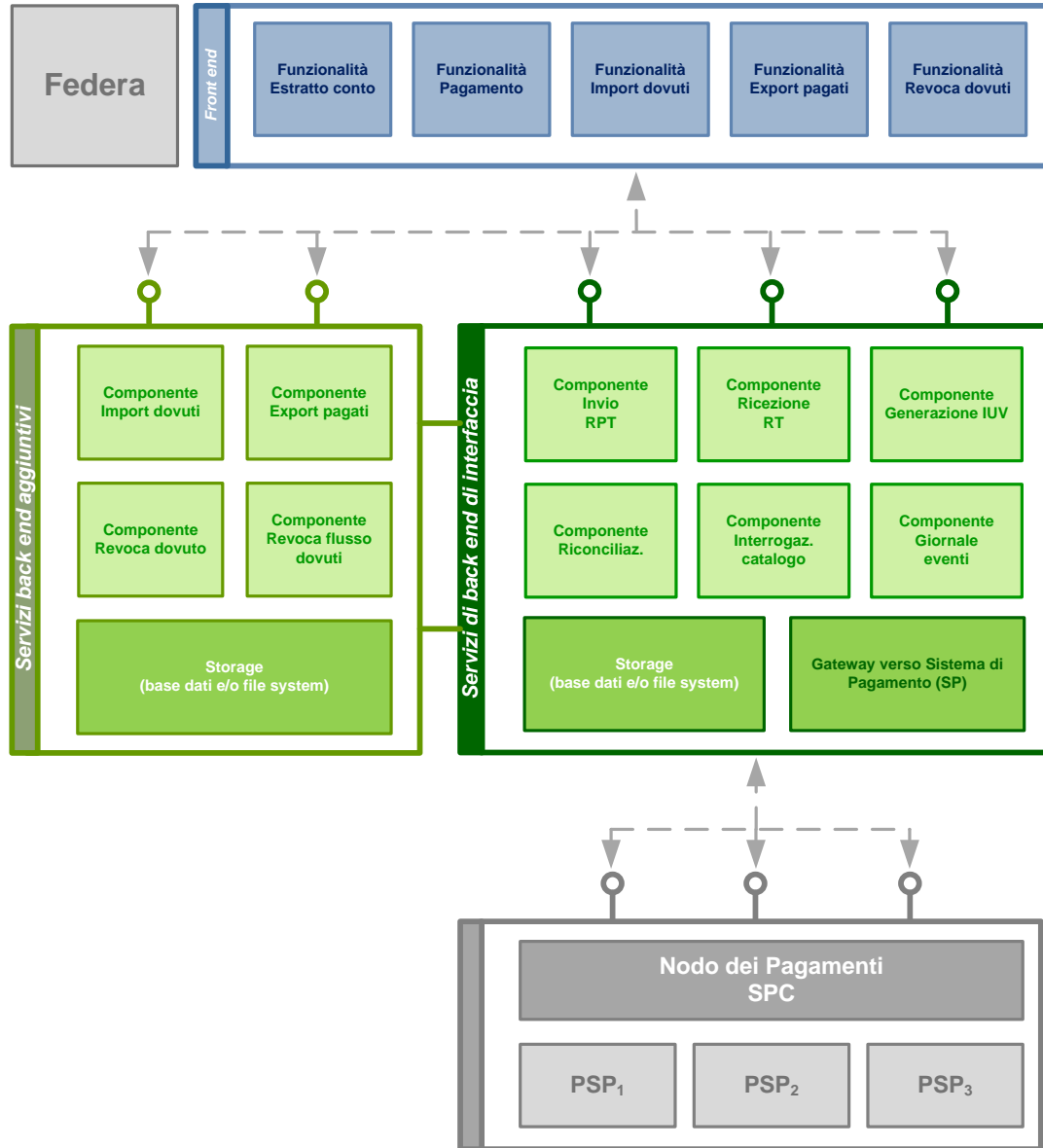


Figura 2 - Componenti di sistema e livelli logici



### 9.1.1 Scenario di riuso

In questo scenario è possibile riutilizzare nei diversi livelli (back-end e front-end) insiemi di componenti diversi, ovvero:

- **Servizi di back-end di interfaccia:** insieme di servizi necessari alla piattaforma per realizzare i processi di pagamento attraverso il Nodo SPC
- **Servizi di back-end aggiuntivi:** insieme di servizi opzionali e abilitanti per le funzionalità di front-end richieste
- Livello di **front-end:** opzionale nel caso in cui l'ente che riusa non disponga di una propria interfaccia utente web

Nel seguito del documento si farà quindi riferimento a queste due modalità di riuso:

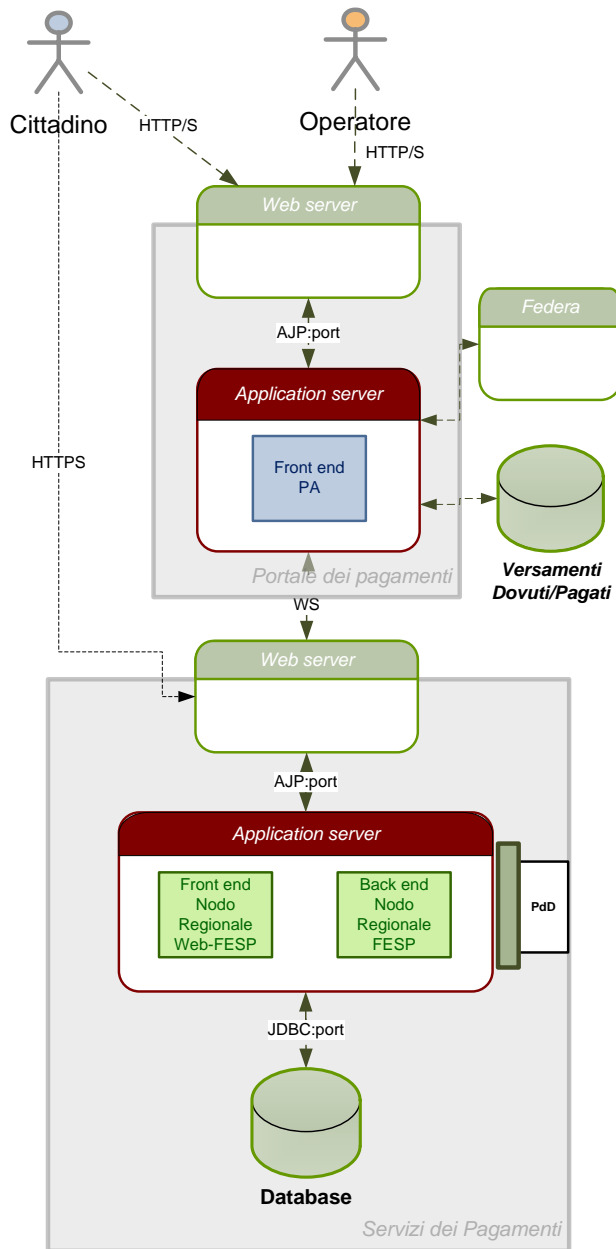
- A. riuso parziale:** inteso come riuso del solo livello di "Servizi di back-end di interfaccia"
- B. riuso completo:** inteso come riuso di entrambi i livelli di front-end e di back-end completi, ovvero "Front-end" + "Servizi di back-end aggiuntivi" + "Servizi di back-end di interfaccia"

### 9.1.2 Scenario di adesione

Nello scenario di adesione, ovvero come nel contesto di un intermediario multiente verso altre PA (come nel Portale dei Pagamenti della Regione del Veneto), è possibile riutilizzare completamente l'intera piattaforma e fruire di tutte le funzionalità descritte approfonditamente nel documento di Specifiche Funzionali ([2]).

### 9.1.3 Scenario di riuso

Si riporta lo schema logico-applicativo nel caso di riuso completo o parziale così come descritto nel § 9.1.1, punti A e B.

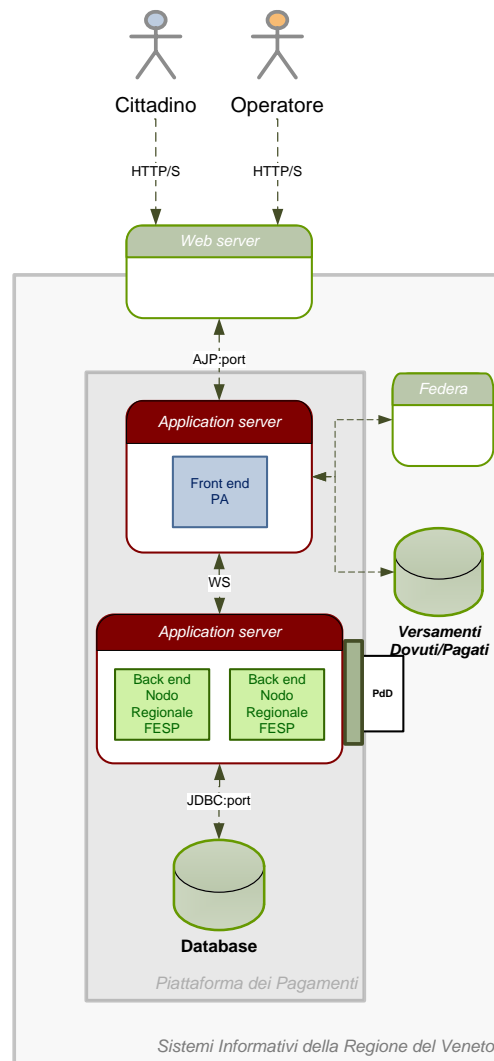


**Figura 3 – Riuso: schema logico-applicativo**

### 9.1.4 Scenario di adesione

Si riporta lo schema logico-applicativo nel caso di adesione inteso come il contesto progettuale e architetturale della Regione Veneto, dispiegato all'interno dei suoi sistemi informativi.





**Figura 4 - Adesione: schema logico-applicativo**

Nel primo schema (scenario di riuso) si è voluto evidenziare come la piattaforma dei pagamenti possa essere integrabile con sistemi di autenticazione e autorizzazione generici, previa realizzazione degli adapter necessari.

Nel secondo schema invece (scenario in adesione) si è voluto calare nel contesto architetturale della Regione del Veneto il sistema di autenticazione, riportando la soluzione adottata nei sistemi informativi ovvero Federa ([1]).

La tecnologia scelta per l'implementazione della piattaforma è Java EE intesa non solo nell'utilizzo del linguaggio ma anche nell'adozione delle sue soluzioni architetturali.

## 9.2 MODELLO FISICO

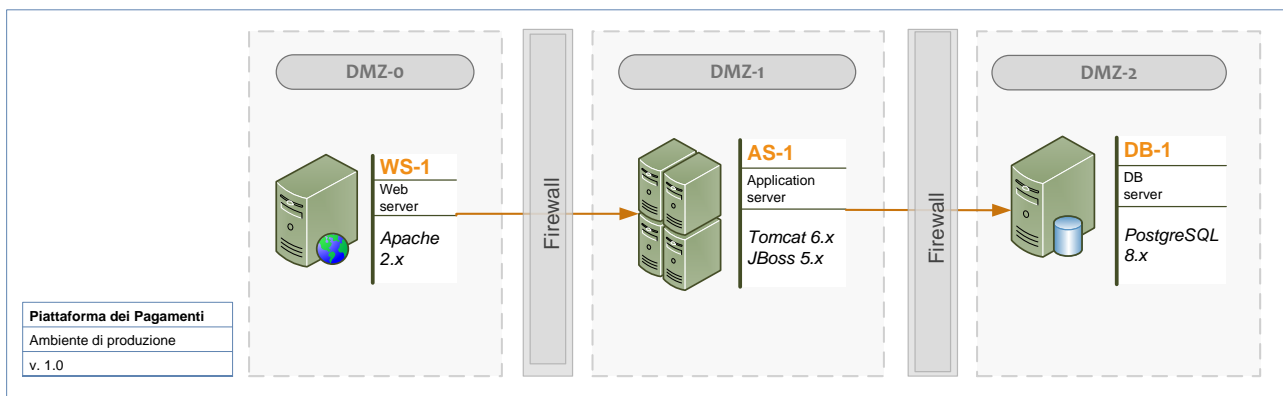
Di seguito si riportano gli schemi fisici dell'architettura in ambiente di produzione, nella configurazione minima e nella configurazione in alta affidabilità.

Le configurazioni sono ovviamente da adattare al contesto progettuale (requisiti/vincoli di affidabilità) e al contesto architetturale dei sistemi informativi di produzione.

### 9.2.1 Schema fisico minimo

Lo schema fisico minimo prevede una sola catena nello stack architetturale (*web server*, *application server*, *batch server* e *DB server*).

Può essere applicato in ambienti o contesti progettuali in cui non è richiesta alta affidabilità o elevati livelli di servizio.

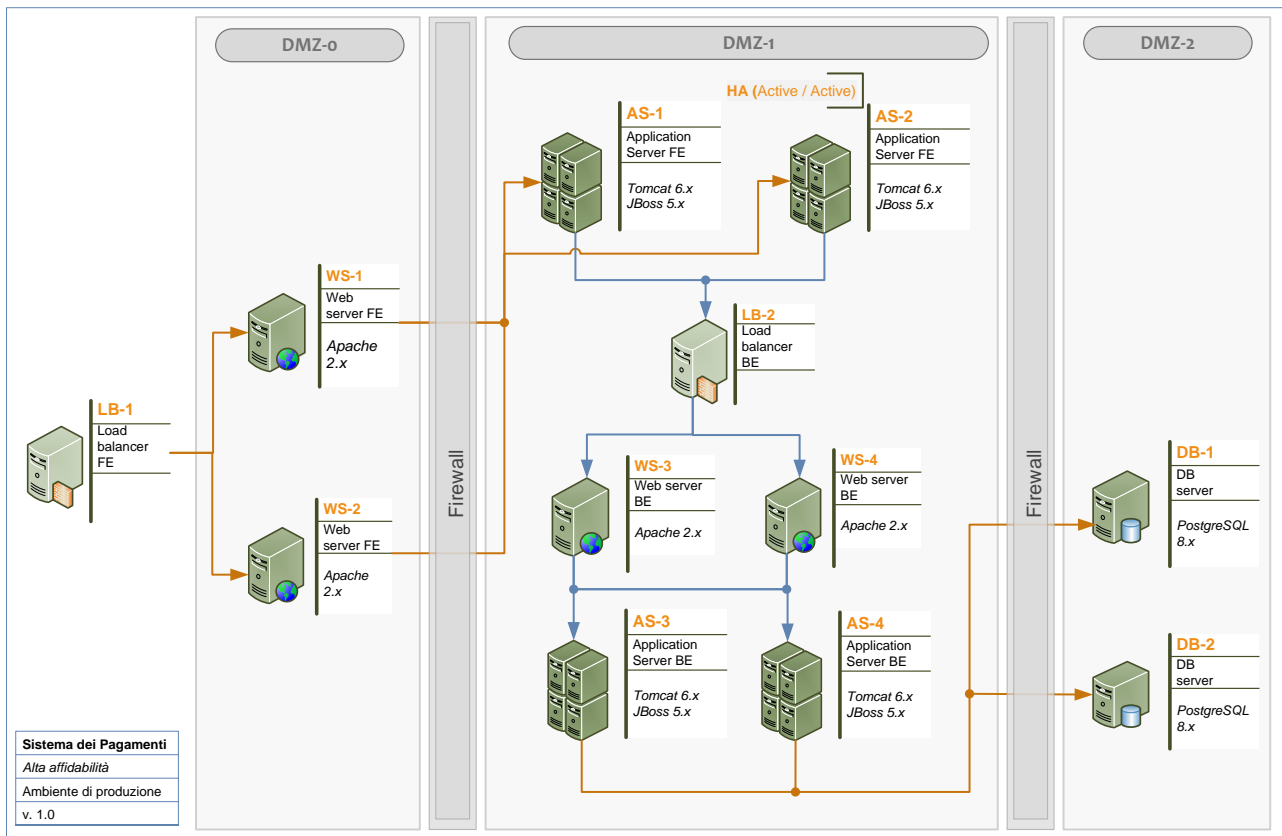


**Figura 5 - Schema fisico minimo**

### 9.2.2 Schema fisico in alta affidabilità

Lo schema fisico in alta affidabilità (failover) prevede due catene nello stack architetturale ridondate (*web server*, *application server* e *DB server*).

Può essere applicato in ambienti o contesti progettuali in cui è richiesta l'alta affidabilità, elevate prestazioni e stringenti livelli di servizio.



**Figura 6 - Schema fisico in Alta Affidabilità**

Lo schema prevede una catena completamente ridondata su due nodi e configurata in alta affidabilità ovvero se un utente è collegato al portale e improvvisamente vi fosse un *failure* del nodo su cui ha la sessione, questa viene replicata sull'altro server e l'utente prosegue la sua operatività in maniera trasparente.

Il *load balancer* distribuisce il carico sui due *web server*, garantendo che in caso di *failure* di uno dei due le richieste vengano ridirezionate sul *web server* attivo.

Entrambi i *web server* possono inoltrare le richieste ad entrambi gli *application server* di front-end e ciascuno dei *web server* è configurato in modo che se vi è un *failure* di un *application server*, ancora una volta le richieste vengano ridirezionate sull'altro.

Gli *application server* di front-end sono configurati in alta affidabilità con replica di sessione in modo da garantire ad un utente collegato la navigazione sul portale in maniera continua e trasparente. L'applicazione di front-end memorizza in sessione solo oggetti serializzabili e non fa uso di *caching* di secondo livello (non c'è quindi necessità di replica della cache).



Gli *application server* di front-end utilizzano i servizi erogati dagli *application server* di back-end. Anche questi ultimi, per gestire il failover, hanno la medesima catena ridondata e incrociata, così nel caso in cui un *web server* o un *application server* non rispondesse le richieste vengono girate sull'omologo server attivo.

Gli *application server* di back-end non sono necessariamente configurati in alta affidabilità con replica di sessione in quanto non strettamente necessario. Infatti i servizi erogati sono tutti *stateless* per cui non esiste una sessione da garantire. L'applicazione di back-end non fa uso di caching di secondo livello (non c'è quindi necessità di replica della cache).

Ancora una volta anche i *web server* di back-end sono raggiunti dagli *application server* di front-end tramite un *load balancer*, che ne ripartisce il carico e gestisce il failover.

I batch ospitati dai *batch server* non fanno uso di caching di secondo livello (non c'è quindi necessità di replica della cache). Inoltre i job vengono presi in carico dai vari batch sincronizzandosi su cambi di stato memorizzati nel database (non c'è quindi necessità di gestire esternamente ai batch una sincronizzazione fra questi anche in presenza di istanze multiple dello stesso batch).

Il failover, infine, è gestito ovviamente anche a livello di DB server garantendo così un'istanza

d

i

D

B

s

e

m

p

r

e

a

t

t

i

v