Disegno Architettura Tecnica ADOC2

Cliente:	**Comune di Bari**
Progetto:	**AVB – Piattaforma documentale, Protocollo Informatico e Gestione Atti**
Codice Commessa/e:	
Redatto da:	**GdL**
Verificato da:	**GdL**
Approvato da:	
Data redazione:	**14/02/2019**
Data verifica:	**16/01/2020**
Data approvazione:	
Versione:	**1.1**
Nome documento:	**AVB_Disegno Architettura Tecnica ADOC2_v.1.1.docx**

VERSIONI

VERS.	Motivo	Modifiche	Data Approvazione	Approvatore
1.0	Prima Emissione			
1.1	Integrazione CAS	**§ 4**		

Indice

1 Contesto

Il presente documento ha lo scopo di illustrare l'architettura generale di ADOC2 ed i requisiti e relativo dimensionamento minimale per l'ambiente di esercizio per la gestione della piattaforma ADOC2 in relazione ai razionali indicati dal Cliente di seguito riportati:

COMUNE	Delibere		Ordinanze		Determine		ANNO	
	2017	2018	2017	2018	2017	2018	2017	2018
Ruvo di Puglia	565	511	134	121	1.533	1.386	27.998	27.241
Mola di Bari	45	110	110	77	1.628	1.364	28.936	29.620
Casamassima	401	249	86	74	1.855	1.637	25.650	22.599
Adelfia	0	0	0	1	1.383	1.275	0	0
Turi	0	0	0	1	1.256	1.007	20.646	19.105
Noicattaro	0	0	0	0	0	0	25.684	23.534
Giovinazzo	0	0	0	0	0	0	23.309	19.864
Cassano delle Murge	1	24	1	1	1.130	956	17.820	15.104
Cellamare	0	0	0	0	0	0	5.942	5.326
Bari	Altro sistema	Altro sistema	Altro sistema	Altro sistema	Altro sistema	Altro sistema	277.715	323.649

2 Rappresentazione architetturale

2.1 Introduzione al Sistema

La soluzione architetturale proposta per la componente ADOC2 si basa su è un'architettura multi-tier. Gli aspetti innovativi della Piattaforma sono:

- Alto livello di affidabilità, con particolare riferimento agli standard di qualità e sicurezza;
- Alto livello di Interoperabilità con i sistemi esterni;
- Architettura della piattaforma altamente scalabile in grado di rispondere facilmente alle esigenze più estreme di capability: ciascun layer è sviluppato con differenti tecnologie JAVA 2 di livello enterprise (ver 1.8).

Il core di servizi di document management è integrato e potenziato da una componente di business process management, concepita con una flessibilità tale per cui:

- offre servizi appoggiati al suo motore di BPM interno (Activiti di Alfresco) per guidare e/o automatizzare l'iter di processi/procedimenti;
- se il processo/procedimento è già guidato da un workflow esterno di un'applicazione da integrare con le funzionalità di document management, offre tutti i servizi che servono a legare il procedimento al fascicolo/documenti che gli competono e a recepire gli avanzamenti guidati dal workflow esterno.

La soluzione prevede un'architettura modulare, logicamente organizzata in quattro livelli, che presenta le seguenti caratteristiche:

- affidabilità: totale ridondanza ai guasti HW e SW di ogni singolo componente;
- scalabilità: l'architettura è progettata per gestire l'elaborazione di grandi volumi di dati;
- flessibilità: la soluzione è facilmente integrabile e customizzabile;
- storage replicato: il dato posto in conservazione è sempre memorizzato in almeno due infrastrutture storage.

Architettura Fisica Le componenti di seguito descritte sono relative sia agli ambienti di collaudo sia a quelli di produzione che sono separati e indipendenti. L'architettura è logicamente organizzata in quattro livelli.

Disegno Architettura Tecnica ADOC2

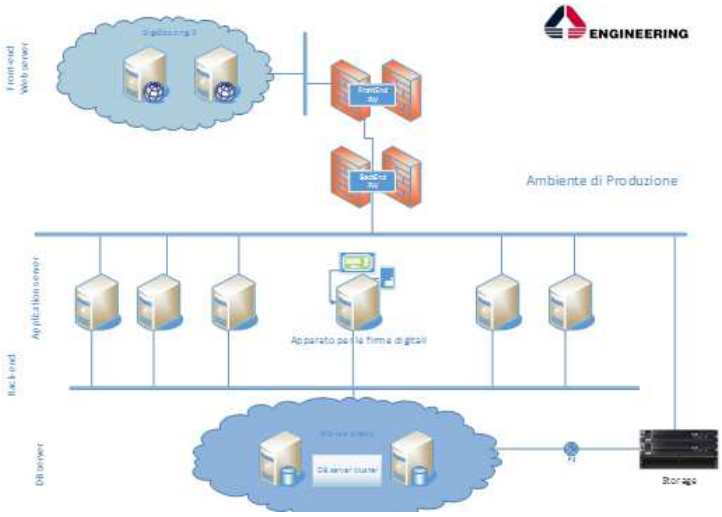


Figure 1 - Architettura Fisica della piattaforma

2.2 Disegno architetturale

ADOC2 è una soluzione modulare, in cui i singoli moduli, salvo quelli trasversali, possono essere attivati singolarmente, nel seguito sono descritte le caratteristiche tecnologiche più significative. I singoli moduli possono essere attivati separatamente. Da un punto di vista tecnologico l'architettura di ADOC2 è illustrata nella figura seguente.

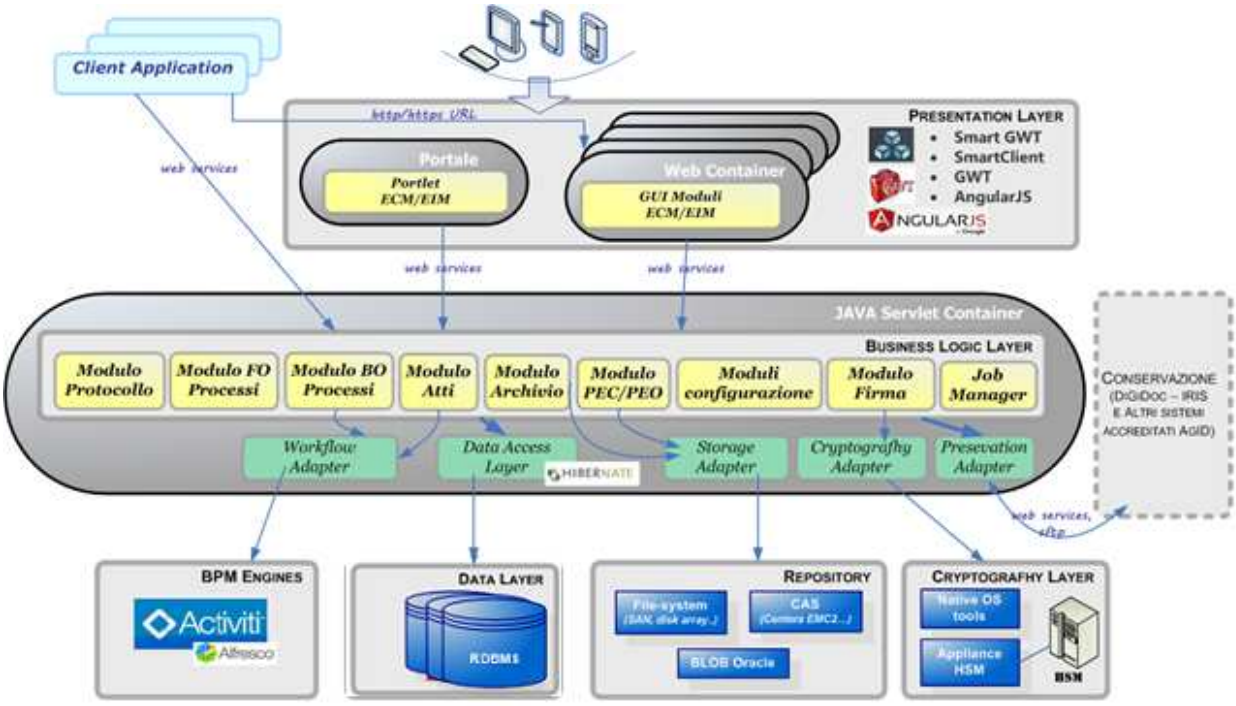


Figure 2 - Architettura della piattaforma ADOC2

I livelli sono descritti nel seguito:

Presentation Layer (PL) È la parte dell'architettura destinata all'interazione con l'utente, che avviene tramite qualsiasi browser web di ultima generazione. È realizzato attraverso i framework **Smart Client** e **SmartGWT** che sono stati prescelti tra molti altri framework **AJAX** per le seguenti caratteristiche:

- consentono uno sviluppo rapido di GUI web agili, potenti e di facile utilizzo, in piena aderenza al paradigma WEB 2.0 e alle specifiche di accessibilità WAI-ARIA;
- provvedono meccanismi per costruire velocemente delle GUI come rappresentazioni a video di DataSource che possono leggere i dati a partire da un'ampia varietà di sorgenti (XML, JSON, dati di un RDBMS, array locali) e per mantenere la sincronizzazione tra quanto mostrato nelle GUI e le sorgenti dei DataSource;
- la portabilità sui diversi browser è garantita al meglio dal fatto che la programmazione avviene interamente in java, senza l'utilizzo di JavaScript (che limita portabilità sui diversi browser);
- consente un equilibrio ottimale tra l'esigenza di un client veloce e leggero, che possa girare anche su macchine senza particolari requisiti di memoria e CPU, e quella di non sovraccaricare troppo la parte server;
- la presentation è interamente controllata attraverso XML (ad esempio per le label dei campi delle GUI) e CSS (per font, stili, sfondi ecc) che possono essere caricati/modificati a runtime, prendendone uno piuttosto che un altro a seconda dello specifico client o su richiesta dell'utente.

Da un punto di vista del dispiegamento il presentation layer di ADOC2 è costituito da:

- un'interfaccia di aggregazione dei moduli (scrivania utente, di amministrazione e configurazione ecc) – realizzata come web-application – dalla quale accedere alle funzionalità web dei vari moduli;
- un insieme di portlet (es. scrivania virtuale dell'utente; to-do list dei processi su cui si hanno attività assegnate; iter di un processo di lavoro; dettaglio di una scheda documento; contenuti di un'unità di aggregazione o contenitore fisico di documenti, lista dei documenti da firmare, …) che, una volta dispiegate in un web container, possono essere richiamate dall'interno di applicazioni terze richiamandole direttamente come url http/https (previo trust tra chiamante e modulo di ADOC2 erogatore).

Il presentation layer contiene solo le logiche di presentazione dei dati e dei documenti digitali, non quelle di archiviazione, lavorazione e reperimento degli stessi che sono provviste dal business logic layer con il quale il presentation layer comunica quasi esclusivamente attraverso servizi RESTful (più di rado servizi SOAP).

Business Logic Layer (BLL) Il Business Logic Layer è il layer in cui sono implementati tutti i servizi di gestione e fruizione del repository documentale e di processo (nonché dei dati del contesto organizzativo, procedurale, archivistico e tecnologico in cui si colloca il repository) che ADOC2 mette a disposizione.

Si articola in una serie di moduli, ciascuno implementato come una o più librerie jar, che comunicano con i sistemi/applicativi esterni e con il presentation layer esclusivamente tramite servizi RESTful e SOAP.

Inoltre, contiene il modulo dedicato alla gestione dei job dell'infrastruttura, tra i quali:

- processi schedulati di acquisizione flussi dati e documenti da file-system, ftps e varie altre sorgenti;
- processi schedulati di invio di flussi dati;
- grosse estrazioni di dati dal repository e produzione schedulata di stampe e report;

- processi di scarico e invio "massivo" di e-mail

Tutti i Web Service sono realizzati utilizzando JAX-WS (Java API for XML Web Services) che è lo standard java per realizzare web service secondo le raccomandazioni WS-I Organization, ovvero come interfacce standard indipendenti dalle piattaforme e dai linguaggi di programmazione utilizzati. Inoltre i web service adottano ***MTOM*** – Message Transmission Optimization Mechanism, raccomandato dal W3C – per trasferire e ricevere i file archiviati e da archiviare, consentendo una trasmissione estremamente efficiente anche per file di grosse dimensioni. Questo layer non comunica direttamente né con il database né con gli storage in cui è fisicamente distribuito l'archivio dei documenti digitali né infine con i servizi di crittografia – firma e timbro – e con i motori di workflow integrati nell'infrastruttura, in quanto le comunicazioni con tali componenti di back-end sono mediate rispettivamente da Data Access Layer, Storage Adapter, Cryptografhy Adapter e Workflow Adapter: ciò serve a creare un livello di disaccoppiamento e quindi l'indipendenza del business logic layer dalla specifica natura/implementazione dei componenti di back-end utilizzati.

Data Access Layer (DAL) Il Data Access Layer è lo strato dell'architettura in cui sono implementate tutte le logiche di accesso ai dati del RDBMS. L'implementazione di questo strato è realizzata tramite il framework di Object Relational Mapping (ORM) Hibernate, che si occupa di mappare le strutture del database negli oggetti java – Hibernate DAO – che sono a loro volta rimappati negli oggetti di dominio – Domain DAO – utilizzati dal Business Logic Layer. Il database non deve necessariamente essere dedicato e può essere condiviso con altre applicazioni.

Back-End Component Adapters (BCA) *Storage Adapter e Cryptografhy Adapter* sono concepiti per garantire al meglio l'indipendenza della soluzione dalla specifica implementazione dei componenti di back-end preposti rispettivamente all'archiviazione dei documenti digitali del repository e ad offrire i servizi di firma e timbro digitale e di workflow management: questo è un elemento importante tenuto conto della continua e rapida evoluzione delle soluzioni offerte per tali componenti dal mondo IT nonché dei cambi normativi in materia di firma digitale. Inoltre, gli adapter sono implementati in modo tale che è possibile attivare – tramite configurazioni - e supportare simultaneamente più istanze e tipologie di componenti di back-end, il tutto in maniera assolutamente trasparente per i moduli del business logic layer. All'interno degli adapter si trovano i "connettori" specializzati per interfacciare un particolare tipo di componente di back-end. I connettori inclusi nella soluzione proposta sono i seguenti:

- nello ***Storage Adapter*** quelli per: file-system; ECM Documentum; Alfresco; SharePoint; CAS Centera di EMC2;
- nel ***Cryptografhy Adapter*** quelli per apparati (HSM) e servizi di firma remota di mercato (Infocert, Aruba, Medas); servizi di firma, generazione e apposizione del timbro digitale, lettura barcode realizzati con componenti interamente open-source;
- nel ***Workflow Adapter*** quelli per Activiti, motore di BPM open-source incluso nella suite Alfresco;
- nel ***Preservation Adapter*** quelli per il versamento in conservazione e l'esibizione verso e dai seguenti sistemi di conservazione accreditati da AgID, al momento la soluzione ADOC2 ha connettori in uso per il sistema di conservazione del Polo Archivistico Regione Emilia Romagna, per DocFly di Aruba e per LegalDoc di Infocert.

2.3 Infrastruttura tecnologica, framework e linguaggi utilizzati

Nel presente paragrafo vengono riassunti gli standard tecnologici e le componenti open-source su cui è basata l'architettura applicativa.

Nome	Utilizzo
J2EE container:	Tomcat 8 o WebSphere Application Server 7
RDBMS:	Oracle Enterprise Edition 12.x, eventualmente in configurazione RAC o PostgreSQL

Tabella 1 – Standard tecnologici di riferimento

Nome	Utilizzo
Java Runtime:	Oracle JSE7, Oracle Java Platform Standard Edition 8
JDK e JRE Compatibility level:	JEE7, Java Platform Enterprise Edition 8.0.

Tabella 2 - Infrastruttura tecnologica e linguaggi utilizzati

Disegno Architettura Tecnica ADOC2

Nome	Utilizzo
JAX-WS (Java API for XML Web Services)	Per realizzare web-service – SOAP e REST – e client che utilizzano XML per comunicare
Jersey	Per realizzare web-service REST e i relativi client (è l'implementazione di riferimento della specifica JAX-RS)
Smart GWT	Framework di base per la realizzazione delle web UI: si occupa del rendering del framework ajax GWT appoggiandosi alle librerie Smartclient
Spring	Per realizzare i componenti come applicazioni java enterprise facilmente configurabili e altamente riusabili
Java Runtime:	JSE7, Java Platform Standard Edition 8
Hibernate	ORM utilizzato per l'accesso al database

Tabella 3 - Principali framework di sviluppo utilizzati

Nome	Utilizzo
JavaMail APIs	Comunica con le caselle e-mail tramite protocolli standard (smtp e IMAP), gestisce i messaggi di posta elettronica sulle caselle
Bouncy Castle Crypto APIs	Verifica firma digitale pkcs#7 e CAdES e apertura busta crittografica
iText	Verifica firma digitale PAdES e apposizione timbro digitale
XAdES4j	Verifica firma digitale XAdES
Aperture SDK	Per verifica formato file da archiviare/allegati alle e-mail
MimeUtils	Per verifica formato file da archiviare/allegati alle e-mail
Apache Tika	Per verifica formato file da archiviare/allegati alle e-mail
Apache Commons	Varie funzioni di utilità
Cron4j	Scheduling di processi
Lucene	Indicizzazione e ricerca full-text e semantica sui dati e file archiviati
OpenOffice	Reader - a supporto dell'indicizzatore – per i documenti nei formati delle suite MS Office e Open Office allegati alle e-mail. Installato come servizio viene anche utilizzato per: - verificare la presenza di macro nei documenti firmati digitalmente (la cui presenza renderebbe la sottoscrizione non valida in termini di legge); - convertire in pdf e pdf/A i documenti nei formati delle suite MS Office e Open Office (per consentirne la visualizzazione da web UI senza necessità di alcun reader installato sul client, neppure Acrobat Reader; prima di firmarli digitalmente e/o allegarli alle e-mail da inviare).
Tesseract-OCR	Per effettuare l'Optical Character Recognition (OCR) sui documenti immagine in modo da poterli poi indicizzare tramite Lucene
jPedal	Per convertire in tiff i pdf immagine, in modo da potervi effettuare l'OCR tramite Tesseract (quest'ultimo lavora solo sui tiff). Inoltre, per visualizzare e ricercare nei file pdf senza necessità di Acrobat Reader installato sul client
Apache Velocity	Per i template delle e-mail (ad esempio quelle di risposta automatica)

Tabella 4 - Librerie/componenti open-source o di terze parti utilizzati

2.4 Postazioni client

Il presente paragrafo dettaglia quali sono attualmente le caratteristiche e i requisiti di un client affinché possa utilizzare le GUI e più in generale le funzionalità web messe a disposizione dai moduli presenti in fornitura.

I sistemi operativi su cui è certificato il corretto funzionamento dei moduli in tutte le loro componenti sono:

- Windows 7, 8 e 10 (a 32 e 64 bit);
- Windows Server (2007, 2008, 2012);
- Linux Ubuntu (qualsiasi distribuzione delle più aggiornate, a 32 e 64 bit).

Nessuna componente di installazione o plug-in aggiuntivo oltre al browser è richiesto ai client per l'utilizzo del sistema. Nei casi in cui, su uno specifico client si debba interagire con delle periferiche o dei tool installati direttamente sul client quali, ad esempio, scanner, stampanti, dispositivi client di firma, è naturalmente richiesto che sul client siano stati installati i driver della periferica utilizzata per effettuare la specifica operazione tramite web GUI (scansione, editing on-line, stampa e firma client). Viceversa, non serve avere installato sul client alcun software di verifica e apposizione firma quale Dike, File Protector o analoghi.

In particolare, per l'utilizzo delle web GUI è necessario che sia disattivato il blocco dei pop-up e siano abilitati:

- javascript/scriplet;
- **plug-in Java** per esecuzione Java Web Start o applet, utilizzati solo laddove sia necessaria l'interazione con una periferica (scanner, dispositivo client di firma): per questi, a oggi, è **garantito il supporto da tutti i browser di ultima generazione**, anche quelli che hanno desupportato NPAPI (tecnologia richiesta per l'esecuzione dei moduli SW Java).

Per l'esecuzione di Java Web Start e applet è altresì richiesta la presenza di una JRE 1.7 o superiore sul client (è sempre garantita la portabilità sulla versione JRE più aggiornata).

Per quanto riguarda gli scanner interfacciabili da web GUI l'unico vincolo è che lo scanner utilizzi o abbia il supporto per i ***driver TWAIN***. Lo scanner può essere sia uno scanner di rete sia uno scanner collegato direttamente alla postazione utente. Sono stati utilizzati con successo dalle web GUI sia scanner standard di Fujtsu, HP, Canon, Epson e molte altre case, sia scanner professionali usati negli archivi o in ambiti tecnici (es. urbanistica, ambiente) per scansioni di formati e risoluzioni particolari (anche scanner planetari per scansioni fino al formato A0). Per quanto riguarda invece la firma digitale sono supportati tutti i dispositivi di firma client:

- smart-card
- token USB
- CNS provvista di certificato di firma
- Aruba KEY

oltre che servizi di firma remota (attraverso HSM in-house o erogati da Certification Authority).

Per quanto riguarda le stampanti di etichette, **la stampa su etichetta e la generazione dell'eventuale barcode non utilizzano SW proprietari delle stampanti**, sfruttando esclusivamente SW open source sia per la generazione e apposizione di barcode o timbri bidimensionali sia per la comunicazione con la stampante che avviene con le medesime modalità utilizzate verso una stampante generica. Le dimensioni e l'orientamento delle etichette si settano tramite configurazioni di sistema e/o sul client e senza alcuna necessità di intervento di adeguamento software. Tra le stampanti utilizzate per la stampa etichette le più utilizzate sono:

- Dymo LabelWriter
- Zebra serie GC, GK e TLP (sia termiche che a trasferimento termico)

In particolare, la stampante usata da un client per la stampa delle etichette si può selezionare da interfaccia di ADOC2 tra le stampanti disponibili per il client (sia collegate al PC che condivise): non deve quindi coincidere con la stampante predefinita del client e anzi è **possibile salvare** la **scelta della stampante** da utilizzare per le **etichette in una «preference» specifica** legata all'**utente**, diversa da quella della stampante usata per le altre stampe effettuabili dal sistema.

2.5 Descrizione integrazioni

La fornitura proposta nasce come un **catalogo di servizi** - web-service SOAP e RESTfull - **da integrare nel «layer» di interoperabilità del Cliente** che adotta ADOC2 come soluzione documentale. Le interfacce web sono disaccoppiate dai servizi che sono molteplici e sono stati definiti e ottimizzati nell'arco di una pluriennale e variegata esperienza di utilizzo da parte di applicazioni/sistemi esterni.

Nelle figura seguente sono illustrati i punti di contatto principali tra applicazioni/sistemi esterni e la soluzione proposta, e tra la soluzione e i sistemi verso cui si candida a diventare l'unico intermediario.

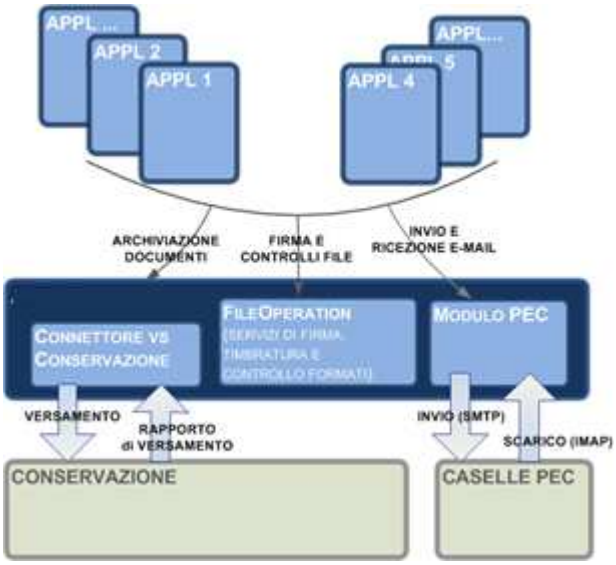
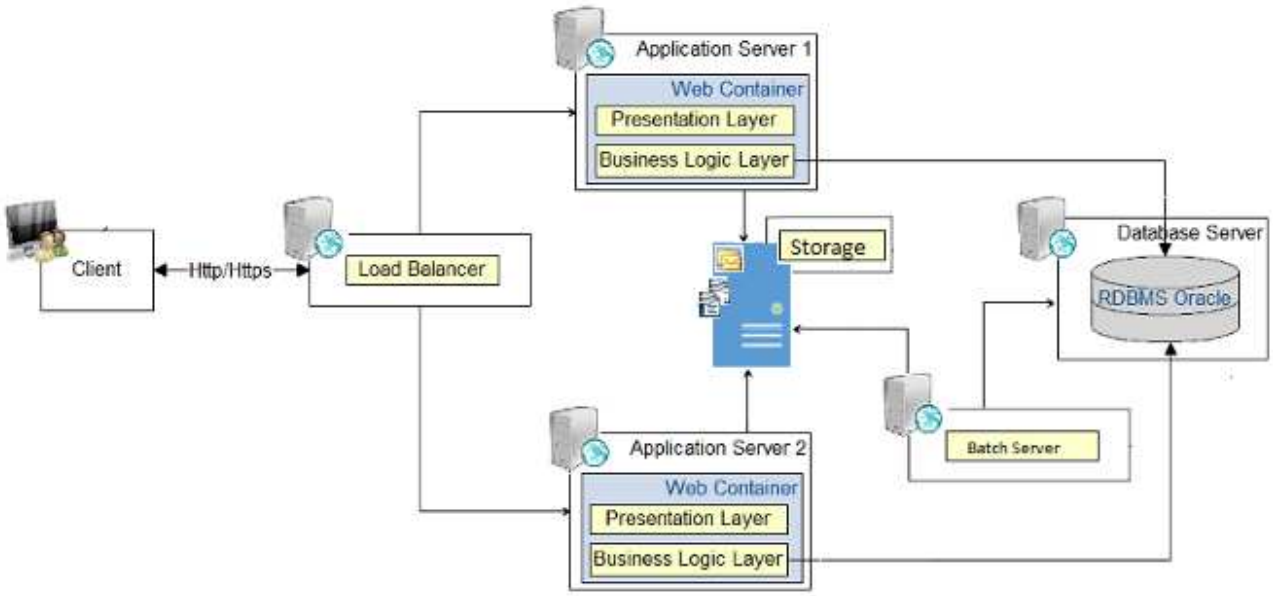


Figure 3 - Punti di contatto principali tra applicazioni/sistemi esterni e ADOC2

Come già detto il sistema proposto ha un impianto a servizi (**architettura SOA**) ma, oltre all'integrazione tramite servizi, offre altre modalità di integrazione, per garantire la massima apertura verso qualsiasi applicazione/sistema produca o gestisca documentazione.

3 Ambiente di Produzione

L'infrastruttura hardware di ADOC2 dell'ambiente di produzione è rappresentata dal disegno:



e descritta nel seguito.

- 1 web server (WS) che costituisce il sottosistema di front-end: ospita un web server Apache HTTP Server 2.0 che deve effettuare il bilanciamento SW verso i 2 nodi Application Server. La macchina in questione è quella che ha come IP XX.XX.XX.XX ed il sistema operativo installato è CentOS Linux versione 7. Per quanto riguarda lo smistamento delle request, il metodo utilizzato è di tipo *bybusyness*. In questo modo viene tenuta traccia del numero di request assegnate a ciascun server e la nuova request viene assegnata automaticamente al server con il numero più basso di sessioni attive. Così viene garantito il fatto che, la lunghezza della coda rimanga uniforme e che una eventuale nuova request sia sempre smistata al server che ha la probabilità più alta di soddisfarla il più rapidamente possibile.

- 2 Application Server (AS1 e AS2), in bilanciamento attivo/attivo (realizzato attraverso Apache http Server del sottosistema di front-end) che ospitano i J2EE container – Tomcat 8 - in cui sono dispiegati i contesti applicativi di ADOC2, sia quelli relativi al presentation layer (GUI) che quelli relativi al business logic layer (servizi). Le macchine in questione sono quelle che hanno come IP: XX.XX.XX.XX (AS1) e XX.XX.XX.XX (AS2). Su entrambe le macchine è installato il sistema operativo CentOs Linux versione 7.

- 1 batch server (BS) dedicato ai processi batch di indicizzazione ed eventuale OCR della documentazione elettronica archiviata nel sistema stesso; su tale server sono attivati i batch di integrazione con i sistemi esterni. La macchina in questione è quella che ha IP: XX.XX.XX.XX ed il sistema operativo installato è CentOS Linux versione 7.

- 1 database server (DB), accessibile via JDBC sia da AS1 e AS2 sia da BS. La macchina in questione ha IP: XX.XX.XX.XX ed il sistema operativo installato è Windows Server 2012 R2.

- 1 SAN o altro storage in alta affidabilità (mirroring) con 500 GB, espandibile, per l'archiviazione dei documenti elettronici e delle e-mail gestiti da ADOC2, accessibile in lettura e scrittura sia da AS1 e

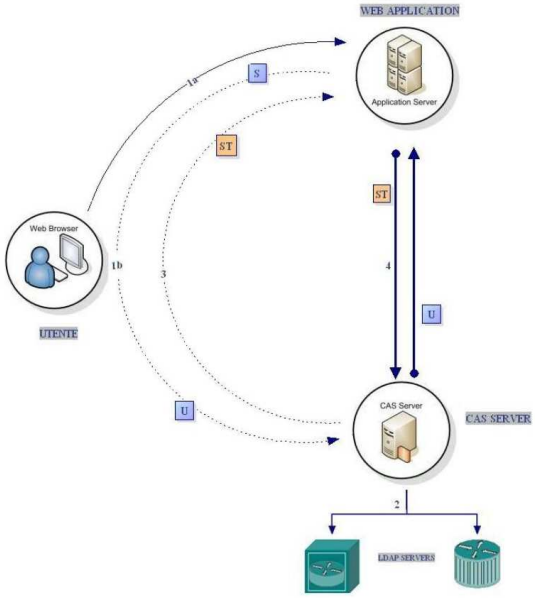
AS2 che da BS. Il suo dimensionamento iniziale dipende dall'attuale storage gestito presso il Cliente.

Riepilogando, l'architettura hardware dell'ambiente di produzione ADOC2 è costituita dalle seguenti macchine:

Server	Indirizzo IP	SO	RAM	CPU	Disco	Software
Protocollo_WEB Web Server load balancer	XX.XX.XX.XX	CentOS Linux versione 7	4 GB	2 Dual Core	60 GB	Web Server Apache Server HTTP
Protocollo_APP2-AS1 Application Server (front end e servizi)	XX.XX.XX.XX	CentOS Linux versione 7	16 GB	2 Dual Core	60 GB	apache-tomcat-8.5.31 jdk1.8.0_191 OpenOffice 4.1.6
Protocollo_APP3-AS2 Application Server (front end e servizi)	XX.XX.XX.XX	CentOS Linux versione 7	16 GB	2 Dual Core	60 GB	apache-tomcat-8.5.31 jdk1.8.0_191 OpenOffice 4.1.6
Protocollo_BATCH-Batch-BS (Batch Server)	XX.XX.XX.XX	CentOS Linux versione 7	4 GB	2 Dual Core	60 GB	jdk1.8.0_191
ADOC2-DB (Database Server)	XX.XX.XX.XX	Windows Server 2102 R2	8 GB	2 Dual Core	80 GB	

4 Integrazione ADOC2 con CAS

L'obiettivo dell'integrazione di ADOC2 con il servizio di autenticazione centralizzata, CAS, è far sì che l'utente si possa autenticare un'unica volta su un portale predefinito con le proprie credenziali ed essere ri-direzionato verso il servizio da lui richiesto.



La modalità di funzionamento del server CAS utilizzata è quella diretta. Questo schema di funzionamento viene attivato nel momento in cui un utente, tramite browser, cerca di accedere ad un servizio applicativo la cui autenticazione è gestita dal CAS. Il server applicativo rimanda l'utente alla pagina di login del server CAS tramite connessione sicura HTTPS, passando come parametro aggiuntivo l'URL dell'applicazione richiesta. Il CAS chiede all' utente che gli vengano fornite le sue credenziali, tramite la relativa form di autenticazione.

Se l'autenticazione non ha avuto successo verrà visualizzato un opportuno messaggio contenente l'errore ottenuto, come nome utente o password incorretti, e l'utente non avrà accesso al servizio. In caso invece di autenticazione avvenuta, l'utente viene ridiretto automaticamente all'applicazione richiesta, appendendo però all'URL di redirezione un ticket, chiamato service ticket (ST), sotto forma di una lunga stringa alfanumerica. Questo ticket serve per indicare che l'utente ha eseguito correttamente l'autenticazione, e richiesto quella specifica applicazione.

Infatti, il server CAS nel momento in cui verifica le credenziali dell'utente, prende anche nota del servizio da lui richiesto, così da associare al ticket stesso, il nome dell'utente e il servizio richiesto. Il service ticket è utilizzabile una sola volta, ed invalidato non appena utilizzato. A questo punto l'applicazione richiamata dall'utente, verifica se il service ticket passatogli dal browser utente sia corretto, tramite una comunicazione

protetta HTTPS con il server CAS. Il server CAS controlla che il ticket inviatogli dall'applicazione sia valido e associato allo stesso servizio richiesto, dopodiché se la validazione ha successo, viene restituito al chiamante il nome utente. A questo punto l'applicazione è sicura dell'identità dichiarata dell'utente, e può procedere nella propria comunicazione con esso, chiudendo di fatto il ciclo di autenticazione.