

SBNCloud

Technology Landscape

Versione: 2.0

Data: 10-Febbraio-2020

INDICE

1	SCOPO DEL DOCUMENTO	4
1.1	ACRONIMI E DEFINIZIONI	4
2	PRINCIPI ARCHITETTURALI.....	5
3	ARCHITETTURA APPLICATIVA.....	6
3.1	ISBNCLLOUD FRONT END APPS	6
3.2	DIGITAL PORTAL SERVICES	7
3.3	API MANAGEMENT	8
3.4	IDENTITY & ACCESS MANAGEMENT	8
3.5	LIBRARY MICROSERVICES	8
3.6	BUSINESS PROCESS MANAGEMENT	9
3.7	EVENTS STREAMING PLATFORM	9
3.8	DATA INTEGRATION SERVICES	10
3.9	ENTERPRISE INDEXING PLATFORM	10
3.10	DATABASE MANAGEMENT SYSTEMS.....	10
3.11	BI & ANALYTICS.....	11
3.12	LOGGING & MONITORING	11
3.13	CONTAINER MANAGEMENT PLATFORM	11
4	SOFTWARE APPLICATIVO	12
4.1	LIFERAY 7.X CE	13
4.2	CLIENT SIDE FRAMEWORKS.....	15
4.2.1	Angular	15
4.2.2	Bootstrap	15
4.3	WSO2.....	15
4.3.1	WSO2 API Manager.....	16
4.3.2	WSO2 Identity Server	17
4.4	ZEEBE.....	18
4.5	SPRING BOOT.....	19
4.6	APACHE KAFKA	19
4.7	APACHE NIFI.....	20
4.8	APACHE SOLR	21
4.9	POSTGRESQL.....	22
4.10	MONGODB	22
4.11	REDASH.....	22
4.12	EFK (ELASTICSEARCH, FLUENTD, KIBANA)	23
4.13	RANCHER.....	24
5	IPOTESI DIMENSIONAMENTO.....	25

5.1	ANALISI VOLUMI DELL'AS-IS	25
5.2	ASSUNZIONI	26
5.3	SCHEMA LOGICO DELLA SOLUZIONE.....	27
5.4	IPOTESI DIMENSIONAMENTO MINIMALE.....	27
5.5	IPOTESI DIMENSIONAMENTO A REGIME.....	28

1 SCOPO DEL DOCUMENTO

Scopo del presente documento è fornire una visione organica dell'architettura applicativa e tecnologica del progetto denominato di seguito SBNCLOUD.

A tal fine verrà presentata l'architettura di riferimento orientata ai micro-servizi e le macro-componenti applicative che la costituiscono.

1.1 ACRONIMI E DEFINIZIONI

Acronimo/Termine	Definizione
IAM	Identity and Access Management
BPM	Business Process Management
SSO	Single Sign On
CaaS	Container as a Service
PaaS	Platform as a Service
B2C	Business to customer

Tabella 1. Acronimi e definizioni

2 PRINCIPI ARCHITETTURALI

Il disegno dell'architettura target sarà progettato sulla base dei seguenti driver tecnologici:

- Utilizzo di soluzioni prettamente Open Source (piattaforme, prodotti, librerie ecc.) con un livello di affidabilità e funzionalità enterprise coerenti con il contesto applicativo e funzionale che si sta affrontando.
- Adozione di un'architettura orientata ai micro-servizi, gestita e monitorata da una piattaforma per l'orchestrazione di Container in modalità PaaS.
- Accesso ai principali servizi, erogati verso i Poli/Biblioteche in modalità SaaS, sia mediante web browser che esposizione di opportune API RESTful in modo da abbracciare un'ottica multicanale.

Più nel dettaglio le caratteristiche dell'architettura proposta garantiranno i seguenti principi:

Robustezza: tutte le componenti software saranno ridondate in modo da essere tolleranti a failure applicativi o infrastrutturali. In tutti i casi in cui sarà possibile si sceglierà un processo di comunicazione asincrono, così da separare la persistenza dell'informazione dai servizi che devono trattarla.

Infatti, la modalità asincrona garantisce che a fronte di un failure di una delle componenti coinvolte dalla comunicazione, l'elaborazione dei dati riparta in modo automatico, senza necessità di ritrasmissioni o di attività manuali.

Manutenibilità: tutte le eccezioni rispetto al normale flusso applicativo verranno raccolte in un componente centralizzato per rendere efficace l'analisi del problema. La logica di separazione a grana fine delle componenti applicative sulle specifiche responsabilità permetterà di agire in modo più rapido nell'identificazione dei problemi e la loro risoluzione.

Portabilità: le componenti software saranno sviluppate scegliendo tecnologie che non creano lock-in su particolari hardware vendor o su particolari sistemi operativi.

Scalabilità: La progettazione applicativa sarà orientata a produrre servizi stateless e componenti in modalità shared nothing per limitare il più possibile le interdipendenze tra le varie componenti. La riduzione dell'interdipendenza tra le componenti è un potente semplificatore per progettare sistemi che sappiano scalare orizzontalmente. Per la componente dati, in tutti i casi in cui non è necessario orientarsi sui tradizionali database relazionali verrà implementata sfruttando le caratteristiche native dei prodotti afferenti al mondo NoSQL che permettono, in modo nativo, di sfruttare le caratteristiche di replica del dato e di scalabilità orizzontale.

Verificabilità: per finalità di audit ogni operazione topica che viene svolta dal sistema verrà tracciata e conservata, in questo modo sarà sempre possibile verificare che il sistema o gli utenti del sistema abbiano agito in modo conforme alle specifiche date.

Multi-tenancy: le applicazioni realizzate saranno in grado di servire contemporaneamente più clienti, i quali potranno accedere alla medesima istanza applicativa in esecuzione su risorse virtuali condivise. L'isolamento dei dati e degli utenti avverrà a livello applicativo e di gestione dei dati, utilizzando gli opportuni meccanismi di autenticazione, autorizzazione e sicurezza.

3 ARCHITETTURA APPLICATIVA

L'architettura di sistema di alto livello con le sue componenti principali è illustrata nel diagramma sottostante.

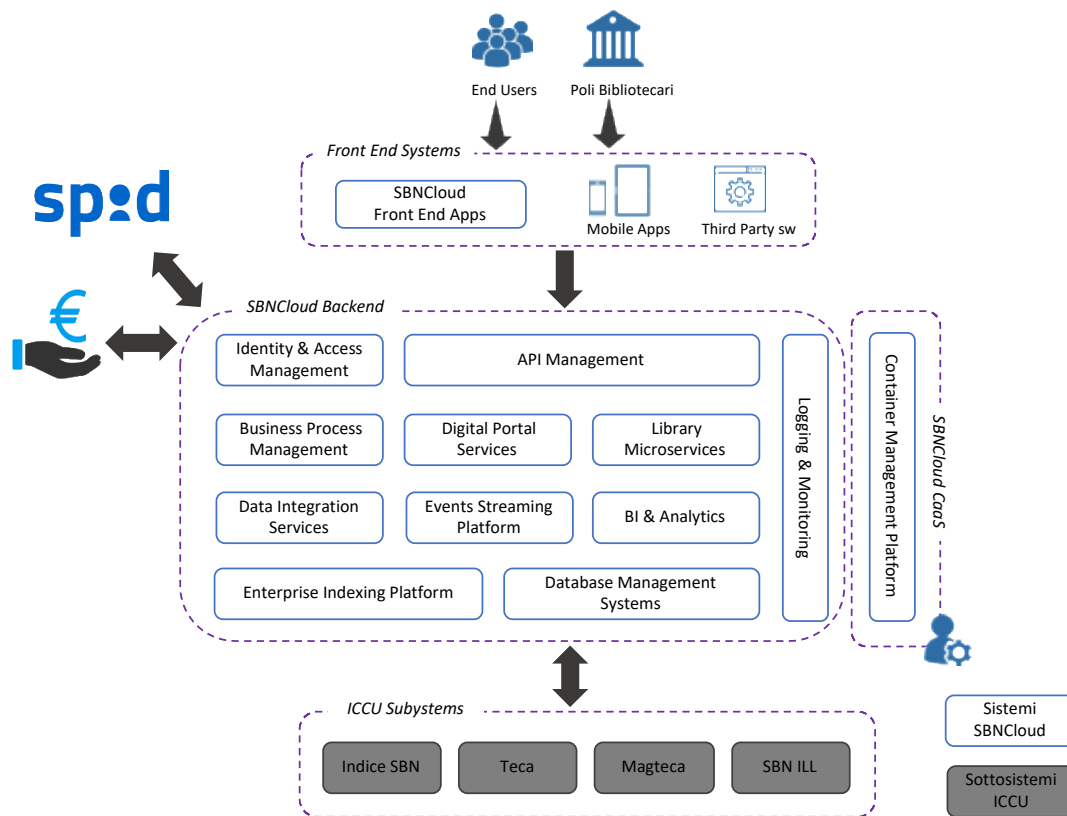


Figura 1: Architettura Applicativa

A seguire saranno descritti i singoli componenti ed il loro ruolo all'interno di SBNCLOUD.

3.1 ISBNCLOUD FRONT END APPS

Rappresenta l'insieme delle applicazioni web di Front End che interagiscono direttamente con gli utenti mediante interfacce grafiche dedicate. Nella soluzione sono stati identificati quattro differenti Portali/Siti web:

1. SBNCLOUD Marketplace: portale di e-commerce che consente la vendita dei servizi offerti da SBNCLOUD e ne gestisce le richieste di convenzionamento da parte dei singoli poli o gruppi di biblioteche.
2. SBNCLOUD OPAC: espone il catalogo ed il patrimonio informatizzato delle biblioteche ai lettori consentendo ai singoli poli/biblioteche la possibilità di organizzare e supervisionare la produzione di contenuti editoriali della propria istanza pubblica di OPAC.
3. SBNCLOUD Mini-Sito Polo/Biblioteca: sito istituzionale accessibile senza registrazione dall'utente di front end.

-
4. SBNCloud LMS: portale destinato al personale di biblioteca che espone i servizi necessari alla gestione del catalogo, del patrimonio fisico e digitale, delle acquisizioni, della circolazione di materiale bibliografico e di tutte le attività connesse alla conduzione di una biblioteca.
 5. SBNCloud Administration: insieme di applicazioni web dedicate al personale ICCU incaricato di amministrare e governare la piattaforma SBNCloud.

I portali e siti web saranno costituiti da un insieme organizzato di applicazioni web "client-side" modulari ed indipendenti, realizzate secondo il principio delle Single Page Application (SPA). Le SPA consumeranno i servizi API messi a disposizione dal componente di API Management rendendo fruibile il contenuto sia sui canali Web che Mobile (seguendo un approccio *mobile first*).

Alternativamente sarà possibile per le singole biblioteche aderenti, strutturando un processo di gestione della registrazione dei poli bibliotecari al sistema, realizzare le proprie interfacce o direttamente attraverso la piattaforma di Digital Portal Services, scegliendo da un elenco finito di templates/modelli di sito, oppure realizzando applicazioni custom in modalità SPA e attingendo quindi ai contenuti specifici tramite endpoint ReST dedicati.

3.2 DIGITAL PORTAL SERVICES

Il componente di Digital Portal Services rappresenta il sistema di backend di Web Content Management, che oltre ad erogare contenuti tramite un frontend tradizionale, attraverso una piattaforma moderna che implementa un'architettura CMS decoupled, espone al front end i servizi necessari a:

- Mantenere separato il codice di programmazione, la grafica ed i contenuti;
- Permettere di tenere distinta l'attività di sviluppo del sito dall'attività di mantenimento;
- Valorizzare la natura comunicativa del sito;
- Consentire l'aggiornamento del sito via web;
- Utilizzare degli editor visuali;
- Creare in automatico la navigazione e la mappa del sito;
- Gestire le fasi di pubblicazione editoriale (workflow);
- Organizzare e supervisionare la produzione di contenuti;
- Gestire gli utenti, tramite opportuna integrazione con il componente IAM;
- Rendere disponibile un motore di ricerca integrato;
- Gestire i portali in lingue diverse;
- Aiutare a creare e mantenere siti accessibili;
- Supportare il lavoro collaborativo.

In questo modo sarà possibile fruire delle features della piattaforma di content management classiche per la realizzazione di una interfaccia base di esposizione dei contenuti, come pure di realizzare interfacce custom per i vari poli, sfruttando le caratteristiche decoupled della piattaforma che vedono i contenuti erogabili anche attraverso degli endpoint REST.

L'amministratore del sistema di Web Content Management gestirà, tramite un pannello di controllo, i contenuti da inserire o modificare. Ad esempio, si potranno gestire articoli, link, immagini, forum, materiale scaricabile, la struttura stessa delle pagine e anche differenti versioni dello stesso sito.

Una singola istanza applicativa sarà in grado di servire contemporaneamente più clienti, i quali avranno accesso alla medesima istanza in esecuzione su risorse condivise. L'isolamento dei dati e degli utenti per singoli Istituti/Poli avverrà a livello applicativo e di gestione dei dati, utilizzando gli opportuni meccanismi di autenticazione, autorizzazione e sicurezza in ottica Multi-Tenancy.

3.3 API MANAGEMENT

La componente di API Management gestisce il ciclo di vita delle API della piattaforma SBNCloud, consentendo pertanto di:

- Esporre i micro-servizi in API immediatamente fruibili come servizi di Business
- Proteggere, attraverso un Gateway, i micro-servizi esposti tramite API per evitare abusi
- Disporre un accesso sicuro per le identità valide alle sole API autorizzate
- Ottimizzare le performance dei sistemi e gestire il ciclo di vita delle API

Grazie all'utilizzo di un API Manager potranno, eventualmente, essere messe a disposizione delle API "pubbliche" per lo sviluppo e/o integrazione di applicazioni di terze parti in linea con il nuovo modello di interoperabilità promosso dall'AgID.

3.4 IDENTITY & ACCESS MANAGEMENT

Il componente di Identity & Access Management (IAM) offre un insieme completo di funzionalità per la gestione dell'intero ciclo di vita delle identità digitali consentendo l'accesso ai servizi ed alle risorse di SBNCloud solo agli utenti correttamente autenticati sulla base delle policy di sicurezza definite da ICCU.

Attraverso questo componente sarà possibile:

- Automatizzare le procedure di onboarding e offboarding mantenendo il controllo necessario sugli utenti e sul loro accesso alle applicazioni web.
- Fornire un'esperienza di Single Sign On (SSO) durante la navigazione dell'utente nelle molteplici applicazioni web della piattaforma.
- Gestire gli utenti e offrire servizi self-service sulle identità digitali (quali la creazione ed il ripristino delle password, gli aggiornamenti del profilo e le richieste di accesso).
- Consentire ai cittadini l'accesso ai servizi attraverso la propria identità SPID attraverso una federazione con il sistema SPID.

3.5 LIBRARY MICROSERVICES

I servizi di business e le logiche core della piattaforma di Library Management saranno realizzati con applicazioni a micro-servizi. Ciascun servizio sarà eseguibile in modo indipendente, permettendo di aumentare significativamente le prestazioni del sistema nel suo complesso e migliorandone, allo stesso tempo, anche la fruibilità.

Ogni servizio identificherà una particolare funzione di business, esposta mediante API RESTful, completa di una propria base dati, processi di calcolo e controlli sui dati.

La cooperazione tra micro-servizi avverrà principalmente in modo asincrono, seguendo un pattern di comunicazione a eventi, o attraverso l'invocazione sincrona di API RESTful.

Lo sviluppo di architetture a micro-servizi presenta vantaggi di rilievo, ad esempio:

- Il micro-servizio è visto come una blackbox e l'esposizione mediante API consente di astrarsi dal dettaglio implementativo (linguaggio o tecnologia utilizzata);
- Ogni micro-servizio può essere rilasciato indipendentemente dagli altri, consentendo una frequenza di rilascio in produzione più elevata;
- L'aumento del livello di automazione dei test è condotto in isolamento sui diversi servizi;
- Il fault di un micro-servizio non si ripercuote sull'intero sistema riducendo il disservizio verso clienti/partner.

Nella soluzione si distinguono quattro tipologie di micro-servizi:

- Core Services: realizzano i servizi di business core per la gestione delle biblioteche.
- Integration Services: servizi di integrazione a supporto della cooperazione applicativa tra applicazioni esterne/interne.
- Servizi Trasversali: sono i componenti che hanno una dimensione trasversale grazie alle funzionalità di base offerte come la profilazione di dettaglio degli utenti, i servizi di logging e auditing, i servizi di messaggistica etc.
- Data Services: servizi che consentono l'accesso online diretto in consultazione e scrittura a determinate aree della base dati garantendone un adeguato livello di sicurezza.

3.6 BUSINESS PROCESS MANAGEMENT

La componente di Business Process Management (BPM) permette di eseguire workflow complessi e long-running consentendo di mappare i processi di business così da gestirli attraverso una sequenza ordinata di task destinati ad utenti e servizi abilitati.

Il BPM garantirà inoltre l'aderenza dei processi allo standard BPMN 2.0 (Business Process Model and Notation).

3.7 EVENTS STREAMING PLATFORM

Il componente di Events Streaming Platform realizza una piattaforma di streaming distribuita che permetterà lo scambio di eventi tra i micro-servizi in near-real-time secondo il pattern Publish/Subscribe con i seguenti benefici:

- Disaccoppiamento: il mittente (publisher) non deve necessariamente conoscere i destinatari dell'informazione (subscriber).
- Maggiore affidabilità: mittente e ricevente non devono necessariamente essere disponibili allo stesso istante per comunicare.

-
- Scalabilità efficiente: aumentando i membri di un gruppo di consumer è possibile processare più rapidamente un flusso di dati.
 - Riduzione processi batch: una piattaforma di streaming promuove lo scambio immediato dell'informazione in near-real-time evitando l'esecuzione di job giornalieri per lo spostamento massivo di record.
 - Facile integrazione: i nuovi micro-servizi potranno sottoscrivere ai feed di interesse senza impattare gli altri micro-servizi.

3.8 DATA INTEGRATION SERVICES

Componente dedicato all'esecuzione schedulata di processi ETL (Extract Transform Load) per l'estrazione e l'acquisizione massiva dei dati tra diversi sistemi applicando al contempo regole per garantirne la qualità e l'aderenza ai formati standard.

Attraverso questo componente sarà possibile ad esempio filtrare, trasformare, ordinare, aggregare, ripulire, deduplicare e convalidare le informazioni bibliografiche contenute in un file fornito da un sistema esterno prima di salvarle nella base dati.

3.9 ENTERPRISE INDEXING PLATFORM

Componente che realizza una piattaforma di ricerca ed indicizzazione di un unico Catalogo bibliografico condiviso tra diversi Poli/Biblioteche e sincronizzato con il servizio Indice SBN.

L'accesso online alle notizie bibliografiche contenute nel Catalogo da parte dei Front End (o di altri micro-servizi) avverrà attraverso API RESTful.

L'aggiornamento del Catalogo potrà avvenire in real-time e/o attraverso l'esecuzione di processi batch.

Il processo di indicizzazione consentirà l'esecuzione di interrogazioni bibliografiche complesse da parte dei Front End offrendo, ad esempio, funzionalità per la ricerca a faccette, il raggruppamento dei risultati, l'hit highlighting, lo spell-checking, il query suggestion, il re-ranking dell'interrogazione, la ricerca di tipo geospaziale e la ricerca multilingua.

3.10 DATABASE MANAGEMENT SYSTEMS

La componente di Database Management System (DBMS) permetterà di gestire e conservare le basi dati applicative, operazionali e tecniche della piattaforma SBNCloud.

Data la natura variegata dei formati utilizzati e le diverse necessità tecnologiche proposte nell'architettura di SBNCloud saranno previsti sia database RDBMS che NoSQL a garanzia delle performance complessive della piattaforma.

3.11 BI & ANALYTICS

La componente di Business Intelligence & Analytics sarà orientata principalmente ai Business Analyst di ICCU e offrirà funzionalità per l'esplorazione grafica dei dati, la progettazione di dashboard interattive in modalità self-service e l'interpretazione dei dati di business allo scopo di monitorarne l'andamento attraverso gli indicatori di performance preconfigurati (KPI).

3.12 LOGGING & MONITORING

Componente dedicata al collezionamento, conservazione e storicizzazione dei log applicativi prodotti dalla piattaforma SBNCloud allo scopo di diagnosticare eventuali malfunzionamenti e prevenire fault di sistema.

Le informazioni di interesse estratte dai log saranno accessibili dal personale ICCU abilitato attraverso un'unica interfaccia grafica centralizzata. Inoltre, un servizio di alerting configurabile consentirà di preallertare il personale al verificarsi di un evento critico (ad es. l'indisponibilità di un determinato servizio).

3.13 CONTAINER MANAGEMENT PLATFORM

Piattaforma di Cloud Computing per la gestione e l'orchestrazione dei container che ospiteranno i micro-servizi della piattaforma SBNCloud.

Consente di gestire l'intero ciclo di vita di un micro-servizio facilitandone lo sviluppo, il deploy, la manutenzione e l'integrazione con altri micro-servizi e offrendo inoltre strumenti dedicati per il monitoraggio, il bilanciamento del carico di lavoro e l'auto-scaling.

4 SOFTWARE APPLICATIVO

L'architettura tecnologica target è stata definita a valle di un processo di valutazione dei prodotti candidati per ciascun componente, descritto nel capitolo precedente, sulla base dei requisiti architetturali e dell'esperienza di Almaviva maturata su progettualità simili.

La tabella seguente mostra l'elenco del software e dei framework applicativi individuati con le relative versioni (le versioni aggiornate alla data di stesura del documento, sono da ritenersi indicative e potrebbero subire variazioni in base ad eventuali future evolutive).

Layer	Componente	Software	Docker container
Front End	SBNCloud Front End Apps (Web Applications)	Single Page Applications realizzate con framework Angular e Bootstrap ospitate su Web Sercer Apache.	Si
Front End/Back End	SBNCloud Front End Apps (Siti Web/Portale); Digital Portal Services	Liferay 7.x CE (Community Edition) ospitato su application server Tomcat	No
Backend	API Management	WSO2 API Manager	Si
	Identity & Access Management	WSO2 Identity Server	Si
	Library Microservices	Spring Boot 2.x + Spring Data	Si
	Business Process Management	Zeebe	Si
	Events Streaming Platform	Apache Kafka	Si
	Data Integration Services	Apache NiFi	Si
	Enterprise Search Platform	Apache SolR	No
	Database Management System	PostgreSQL 11.x; MongoDB 4.x	No
	BI & Analytics	Redash	Si
	Logging & Monitoring	EFK (Elasticsearch, Fluentd, Kibana)	Si (tranne Elasticsearch)
User Authentication	OpenLdap	No	
Container As A Service	Container Management Platform	Rancher 2.x	

Tabella 2. Elenco Software Applicativo

L'immagine sottostante illustra inoltre un diagramma che sovrappone i prodotti con i componenti architetturali descritti nel capitolo precedente.

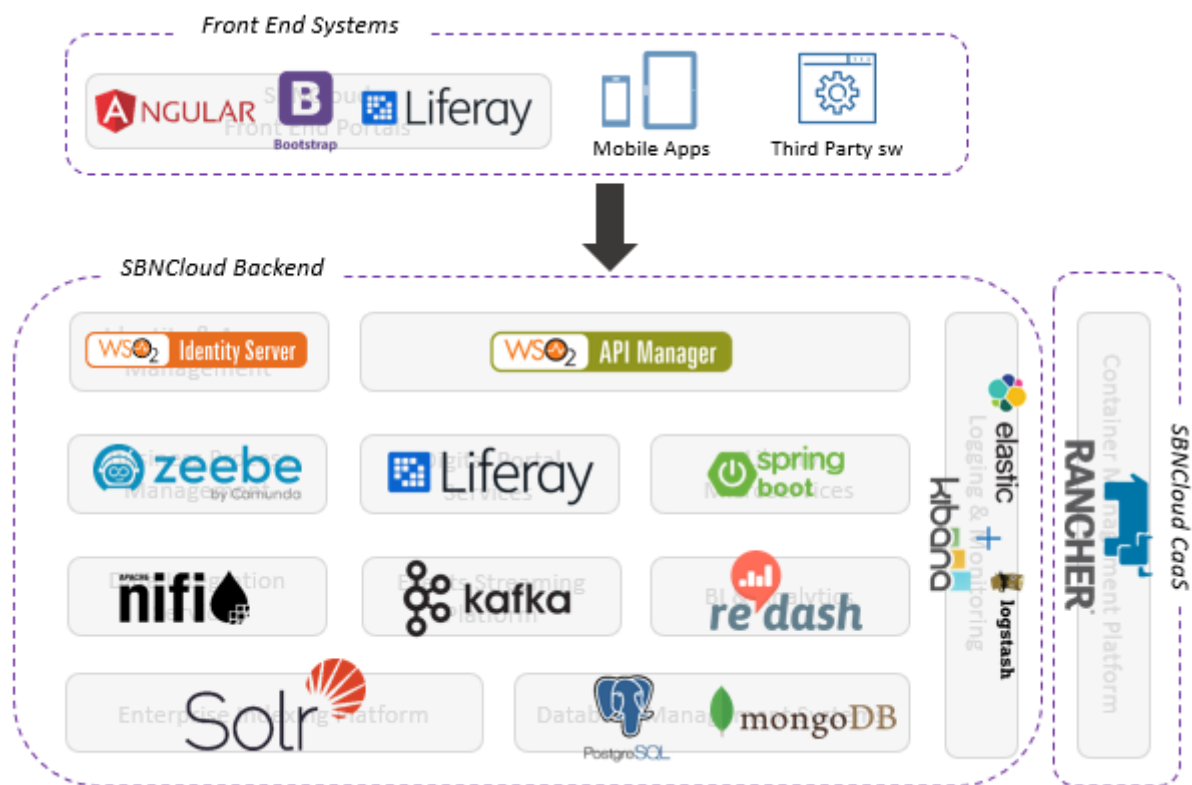


Figura 2: Software Applicativo

4.1 LIFERAY 7.X CE

Sviluppato in Java e distribuito con il modello open source, Liferay CE è una piattaforma completa di Digital Experience ottimizzata per la creazione e la gestione di portali web. Liferay consente la realizzazione di moduli web riusabili (ad es. Portlet) che possono essere composte per formare il corpo delle pagine web che costituiscono i siti web di un portale.

Le peculiarità del prodotto possono essere sintetizzate nei punti seguenti:

- **Solidità:** Liferay costituisce una solida piattaforma che consente di costruire rapidamente siti web fruibili sia da desktop e mobile.
- **Flessibilità ed estendibilità:** permette lo sviluppo di nuove applicazioni e personalizzazioni in base alle esigenze.
- **Sicurezza:** fornisce elevati livelli di sicurezza applicativa e di infrastruttura compatibilmente con le normative vigenti in materia di sicurezza e privacy.
- **Suite di componenti:** fornisce tutte le funzionalità standard necessarie in un sito.
- **Funzioni redazionali:** fornisce capacità di gestione documentale/redazionale avanzata.
- **Integrazioni:** è corredato da un set di plugin per l'integrazione con altri sistemi.
- **Supporto multilingua:** supporta la localizzazione in tutte le lingue internazionali con più di 30 lingue già supportate al momento.

A partire dall'ultima release della serie 7.1 e con l'attuale 7.2 la piattaforma presenta moderne features che ne consentono un'implementazione come motore di contenuti disaccoppiato. Il concetto di "decoupled CMS" ha recentemente guadagnato molta popolarità consentendo agli sviluppatori front-end di accedere alle funzionalità di presentazione dei contenuti e ai servizi principali utilizzando gli strumenti con cui sono più a loro agio e le tecnologie front-end che preferiscono, impiegando per il backend motori di contenuti e flussi editoriali stabili testati ed acclarati da successi di mercato.

Liferay 7.x si propone come CMS disaccoppiato (decoupled), che offre la possibilità di combinare un approccio privo di interfacce di frontend con un'interfaccia utente completa, consentendo alle aziende di scegliere per ciascun progetto e che necessitano di ciò che si adatta meglio, se l'interfaccia tradizionale o interfacce custom che consumano i contenuti invocandoli tramite REST API.

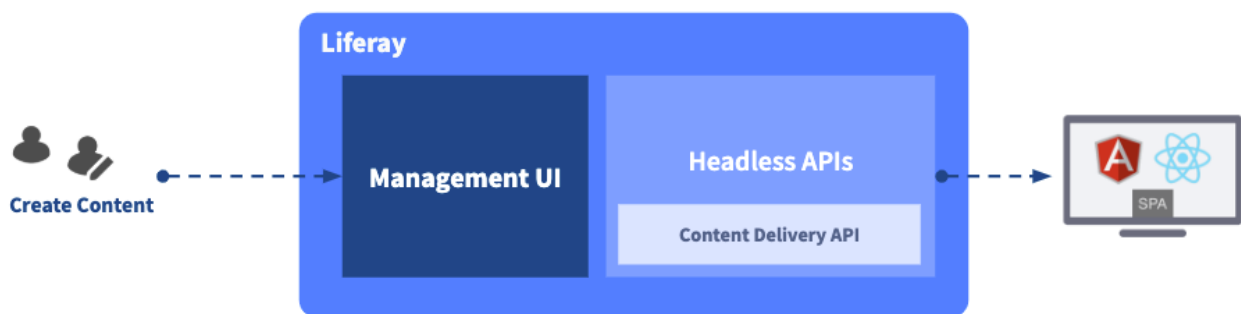


Figura 3: Liferay CE decoupled

Il sistema CMS è stato originariamente creato per i siti Web server side in mente e all'epoca era sufficiente. Tuttavia, poiché la tecnologia ha iniziato a passare da desktop e dispositivi mobili a dispositivi diversi, come smartwatch, dispositivi ad attivazione vocale e altre forme di IoT, è stato necessario creare nuovi modelli di presentazione dei contenuti, il che ha reso il caricamento degli stessi più oneroso sia per gli sviluppatori che per i professionisti del marketing.

Liferay disaccoppia la presentazione e il contenuto, offrendoti il meglio di un CMS tradizionale con il meglio di un'architettura headless in un'unica piattaforma, esponendo ricchissime interfacce REST che gli sviluppatori dei frontend possono implementare via API.

Le API REST di Liferay Portal seguono le specifiche OpenAPI e consentono alle applicazioni di frontend (come le Single Page Applications) di utilizzare i servizi Web RESTful senza essere legate all'interfaccia utente del portale Liferay. Questo fornisce molta libertà durante la progettazione e lo sviluppo delle applicazioni web.

L'approccio disaccoppiato al CMS consente di fruire di:

- 1) Una parte di gestione di siti e contenuti editoriali robusta e centralizzata, per n siti e touchpoint.
- 2) Una interfaccia di consumo dei contenuti editoriali standard direttamente erogata dalla piattaforma.
- 3) La possibilità di distribuire i contenuti editoriali in maniera innovativa e orientata alla massimizzazione della digital experience degli utenti finali, realizzando interfacce ad hoc specifiche per touchpoint.

Per garantire alta affidabilità e scalabilità al prodotto, si suggerisce una configurazione in cluster, con 4 nodi bilanciati.

4.2 CLIENT SIDE FRAMEWORKS

Sono le tecnologie che implementano l'interfaccia utente garantendone l'interazione in ottica B2C multicanale.

4.2.1 *Angular*

Angular è il framework client realizzato da Google con l'obiettivo di fornire un'esperienza utente più fluida e simile alle applicazioni desktop dei sistemi operativi tradizionali.

Le applicazioni Angular sono basate su HTML, CSS e TypeScript e realizzano una Single Page Application ovvero una applicazione web che può essere eseguita su una singola pagina web del browser.

Nell'ambito di SBNCloud saranno realizzate molteplici Single Page Application, una per ogni funzione di business, collegate tra loro attraverso link contenuti all'interno dei singoli portali web che agiranno da Consumer dei servizi applicativi esposti dall'API Manager.

4.2.2 *Bootstrap*

Per la realizzazione delle Single Page Application si farà uso anche del framework Bootstrap così da rendere web responsive la fruizione delle funzionalità di front-end. Si parla di responsiveness, e più in particolare di Responsive Web Design, quando lo stile CSS utilizzato nell'applicazione è pensato e sviluppato per rendere il contenuto della pagina Web fruibile a prescindere dalle dimensioni dello schermo del dispositivo su cui viene eseguito, sia esso un pc, un notebook, uno smartphone o un tablet.

La principale caratteristica di Bootstrap è quella di essere compatibile con tutti i browser più recenti e di facilitare allo sviluppatore front end il compito di scrivere codice poiché si adatta nativamente ad ogni tipo di dispositivo.

4.3 WSO2

WSO2 è una società open source fondata nel 2005. I prodotti sono rilasciati con versione Apache License Version 2 ma è possibile acquistare un supporto specialistico.

Tutti i prodotti WSO2 si sviluppano su un framework comune chiamato "OSGI – WSO2 Carbon" che fornisce funzionalità comuni a tutti i prodotti. Avendo la stessa base, i diversi prodotti sono facilmente integrabili tra loro; inoltre tra le funzionalità offerte dal Carbon, di interesse è la possibilità di effettuare modifiche e deploy a caldo delle soluzioni implementate: questa caratteristica conferisce ai sistemi la possibilità di lavorare in regime di business continuity, ovvero la possibilità di effettuare rilasci senza dover gestire tempi di disservizio in produzione.

Le soluzioni offerte forniscono inoltre la flessibilità di sviluppare applicazioni e servizi on-premises, su cloud pubblici o privati, o su ambienti ibridi e con la possibilità di migrare facilmente da una scelta all'altra.

Infine, i sistemi risultano agnostici rispetto all'infrastruttura di esecuzione: possono essere eseguiti su sistemi operativi differenti e/o architetture hardware differenti. Tutti i prodotti richiedono come unico prerequisito la presenza di una virtual machine JAVA.

Tra la gamma di prodotti offerti da WSO2, la selezione è stata effettuata sulle componenti che seguono.

4.3.1 WSO2 API Manager

In un'architettura Enterprise composta da più sistemi che erogano servizi la componente di API Manager consente di gestire ed orchestrare le richieste per accedere alle API da parte di applicazioni e partner, inviando le richieste al backend ed effettuando il Routing della risposta al sistema chiamante. La componente di Api Management permette inoltre di gestire un registro dei servizi, di analizzare e limitare l'uso di determinati servizi, di memorizzare il risultato di determinate chiamate a servizi esterni (per limitare le spese) ed è perfettamente integrato con il sistema WSO2 Identity and Access Management.

Si evidenzia che WSO2 API Manager è stato recentemente nominato Leader nel Forrester Wave™: "API Management Solutions, Q4 2018" a dimostrazione della qualità dei servizi offerti anche rispetto ad altri competitors commerciali.

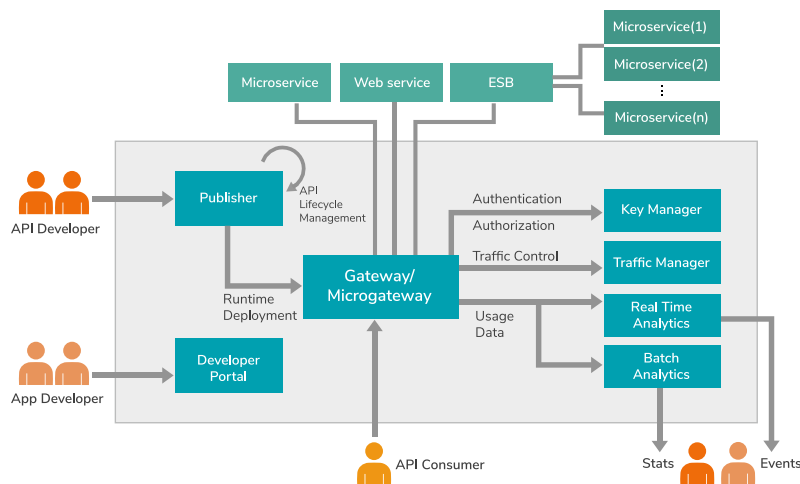


Figura 4: WSO2 API Manager

Si vanno ad identificare i seguenti elementi principali del prodotto:

- API Store: Questo componente fornisce uno spazio per i fruitori dell'API, consente di auto-registrarsi, scoprire, sottoscrivere e valutare nuove API. Viene inoltre data la possibilità ai developer di creare applicazioni logiche che inglobano più API e di applicare ad esse meccanismi di throttling.
- API Publisher: Questo componente consente ai fornitori di API di creare e pubblicare facilmente le loro API.
- API Gateway: Il componente di Gateway è la parte fondamentale del sistema, permette di gestire l'interazione con le API, in particolare andando a definire meccanismi di sicurezza per l'interazione con le api stesse. Si basa sul prodotto WSO2 ESB, noto per le ottime performance, in particolare è

rappresentato da un proxy che intercetta le chiamate verso le API applicando le policy di sicurezza e di throttling.

- API Key Manager: La gestione della sicurezza è affidata all'API Key Manager. Tale componente è un estratto del componente Identity Server di WSO2 e permette di andare a gestire la profilazione degli utenti e le chiavi di accesso definite dai protocolli OAuth2.

Quando l'API Gateway viene chiamato per gestire una API ingaggia il modulo di key management per verificare la validità dei token di autenticazione.

Il gateway API supporta l'autenticazione API con lo standard OAuth 2.0 (implicit, authorization code, client, SAML, IWA Grant Type), e consente alle organizzazioni IT di far rispettare i limiti di frequenza applicando i criteri di sicurezza preconfigurati.

Durante il ciclo di vita ogni API viene:

1. Sviluppata (Develop) ad esempio attraverso micro-servizi.
2. Pubblicata (Publish) ovvero registrata sull'API Manager.
3. Gestita (Manage) ovvero viene data la possibilità di associare alla pubblicazione delle api un meccanismo di bpm per poter avere un ciclo approvativo della pubblicazione.
4. Monitorata (monitor) ovvero viene monitorato il ciclo di vita delle API in termini di tempo di utilizzo o tipologia di utenti fruitori. Questo meccanismo di monitor può essere integrato con strumenti avanzati che permettono analisi statistiche approfondite dashboard di controllo funzionali e personalizzabili.

4.3.2 WSO2 Identity Server

Con la componente WSO2 Identity Server viene fornito un sistema sofisticato di gestione della sicurezza e delle identità digitali sia per le Web Application che per i micro-servizi attraverso funzionalità di single sign-on/sign-out (SSO), identity federation, strong authentication (autenticazione a più fattori), identity administration (funzionalità self service per la registrazione ed il ripristino delle credenziali), account management (gestione delle utenze con workflow autorizzativi), identity provisioning, fine-grained access control (supporta XACML) e API security attraverso l'integrazione con WSO2 API Manager (OAuth2, SAML2, JWT, Windows Authentication).

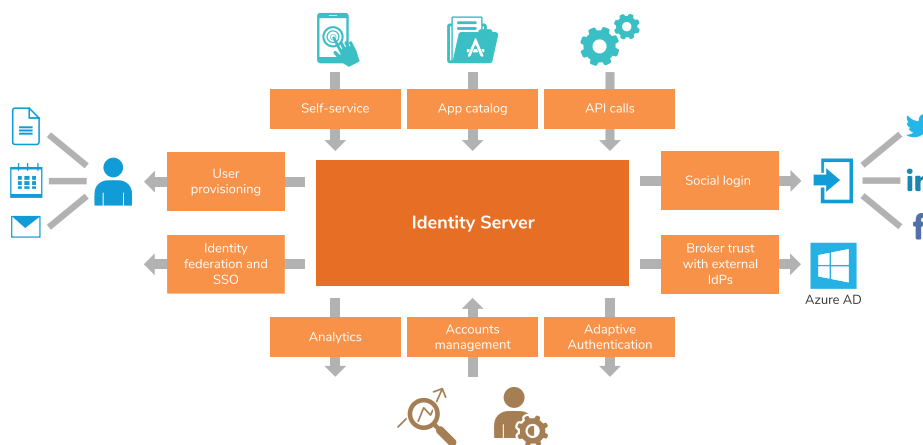


Figura 5: WSO2 Identity Server

Il prodotto offre inoltre un'interfaccia grafica per il monitoraggio, la reportistica e l'auditing così da mantenere sotto controllo l'intero ciclo di vita dell'Identità Digitale.

4.4 ZEEBE

Zeebe è un workflow engine per l'orchestrazione di micro-servizi Open Source sponsorizzato da Camunda. Nei flussi di lavoro orchestrati da Zeebe, ogni attività viene eseguita da un micro-servizio dedicato.

Zeebe è stato progettato per risolvere il problema dell'orchestrazione dei micro-servizi su larga scala e per raggiungere questo obiettivo fornisce:

- Scalabilità orizzontale e nessuna dipendenza da un database esterno; Zeebe scrive i dati direttamente nel filesystem sugli stessi server in cui è distribuito e può facilmente distribuire l'elaborazione su un cluster di macchine per offrire un throughput più elevato.
- Tolleranza agli errori tramite un meccanismo di replica facile da configurare, che consente a Zeebe di resistere a guasti hw o al sistema operativo senza perdita di dati e con tempi di fermo minimi.
- Un'architettura basata sui messaggi in cui tutti gli eventi rilevanti per il flusso di lavoro vengono scritti in un registro in modo immutabile. Questi eventi possono essere esportati su sistemi esterni per l'archiviazione a lungo termine per fornire, ad esempio, un sistema di monitoraggio dell'intero flusso di lavoro.
- Un modello di interazione publish-subscribe, che consente ai micro-servizi che si connettono a Zeebe di mantenere un elevato livello di controllo fornendo al contempo un meccanismo per gestire i picchi di lavoro (pattern back-pressure).
- Flussi di lavoro visivi modellati in BPMN 2.0 standard ISO in modo che gli stakeholder tecnici e non tecnici possano collaborare alla progettazione del flusso di lavoro in un linguaggio comune.
- Un modello client indipendente dal linguaggio, che consente di costruire un client Zeebe praticamente in qualsiasi linguaggio di programmazione.

Di seguito un semplice diagramma che mostra come Zeebe può essere utilizzato per la visibilità nei flussi di lavoro che si estendono su micro-servizi:

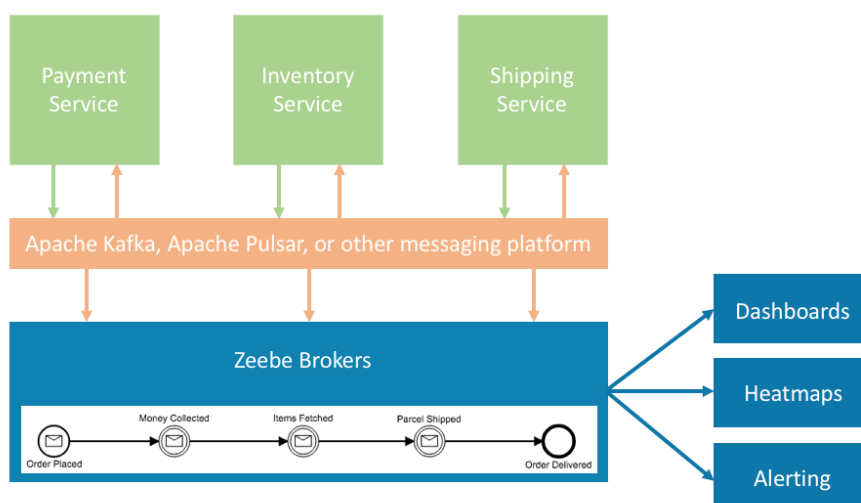


Figura 6: Zeebe: flusso di esempio

4.5 SPRING BOOT

Lo sviluppo dei singoli microservizi avverrà tramite il framework Spring Boot che consente di semplificare la realizzazione di applicazioni stand-alone che espongono le proprie funzionalità attraverso servizi RESTful attraverso la configurazione automatica della struttura, delle librerie e delle dipendenze di compilazione dell'applicazione stessa. In tal modo lo sviluppatore può focalizzarsi esclusivamente sulla creazione della logica di business.

L'accesso ai dati relazionali e non relazionali avverrà attraverso il framework Spring Data, facilmente integrabile in Spring Boot, che dispone nativamente dei moduli per MongoDB, SolR e PostgreSQL.

Ad ogni modo si valuterà, in base all'esigenza ed alla specificità del microservizio, se estendere il framework core (ad es. Spring Security, Spring Integration) e/o sfruttare librerie Open Source (garantite e sotto supporto per gli aggiornamenti di sicurezza).

L'accesso alle funzioni applicative avverrà mediante esposizione di API RESTful in JSON e tutte le richieste saranno gestite dall'API Gateway del prodotto WSO2 API Manager, indipendentemente dal richiedente (Single Page Applications, App Mobile o applicazioni di terze parti).

L'API Gateway esporrà tutte le API messe a disposizione dai microservizi di Back End in maniera sicura, facilmente fruibile e controllata. Sarà posizionato davanti ai servizi esposti, in modo tale che tutti i sistemi esterni debbano effettuare l'accesso a servizi e risorse attraverso questo componente.

Infatti, il Gateway, per ogni accesso al sistema, da parte di un'applicazione esterna, effettuerà i seguenti passi:

- Riceve le richieste per accedere alle API
- Attua le politiche di controllo di accessi, integrandosi se necessario anche con altre componenti
- Applica le regole di rate limiting e throttling
- Invia le richieste al backend dell'API (questo step può essere mediato da un servizio di integration)
- Effettua il routing della risposta al sistema chiamante.

4.6 APACHE KAFKA

Apache Kafka è una piattaforma di streaming distribuita che possiede tre funzionalità chiave:

- Pubblicare e sottoscrivere stream di record. Sotto questo aspetto è simile a una coda di messaggi o sistema di messaggistica aziendale.
- Memorizzare flussi di record e garantire Fault Tolerance.
- Elaborare flussi di record non appena essi si verificano.

Apache Kafka viene utilizzato per due ampie classi di applicazioni:

1. Realizzazione di pipeline di flussi di dati in tempo reale che muovono dati in modo affidabile tra sistemi o applicazioni.
2. Creazione di applicazioni di streaming in tempo reale che trasformano o reagiscono ai flussi di dati.

Kafka può essere installato in modalita cluster, su uno o più server. Il cluster memorizza i flussi di record in categorie denominate Topic. Ogni record appartenente ad un topic è caratterizzato da una chiave, un valore ed un timestamp.

Kafka ha quattro API principali:

1. L'API Producer consente a un'applicazione di pubblicare un flusso di record su uno o più Topic di Kafka.
2. L'API Consumer consente a un'applicazione di sottoscrivere uno o più Topic ed elaborare il flusso di record da essi prodotto.
3. L'API Streams consente a un'applicazione di agire come un processore di flusso, consumando un flusso di input da uno o più Topic e producendo un flusso di output su uno o più Topic di output, trasformando efficacemente i flussi di input in flussi di output.
4. L'API Connector consente di creare ed eseguire produttori o consumatori riutilizzabili che collegano i Topic di Kafka alle applicazioni o ai sistemi di dati esistenti. Ad esempio, un connettore di un database relazionale potrebbe acquisire ogni modifica su una tabella già esistente.

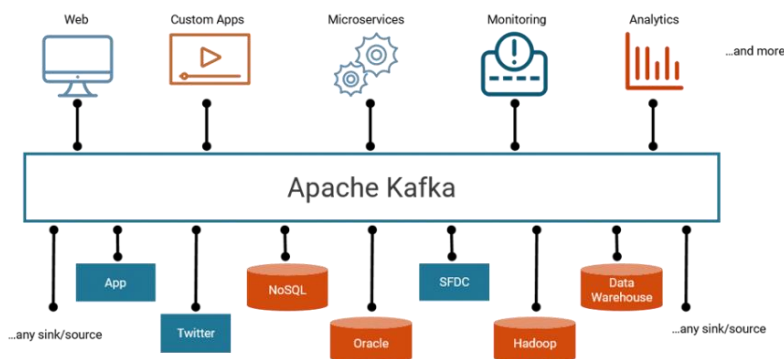


Figura 7: API di Apache Kafka

Apache Kafka gestisce la comunicazione tra client e server con un protocollo applicativo che si appoggia su una comunicazione TCP. Questo protocollo è versionato e mantiene la retrocompatibilità con la versione precedente. Il client Kafka è disponibile come libreria in diversi linguaggi, tra cui Java.

La peculiarità di Kafka è la resilienza attraverso le repliche distribuite. Queste repliche non influiscono in alcun modo sul throughput.

4.7 APACHE NIFI

NiFi è un prodotto creato per automatizzare i data flow tra i sistemi. Il termine data flow identifica il flusso di informazioni automatizzato e gestito tra i sistemi.

I concetti fondamentali della progettazione di NiFi sono strettamente correlati alla programmazione basata su flussi.

Ecco alcuni dei principali aspetti di NiFi:

- FlowFile: rappresenta ogni oggetto che si muove attraverso il sistema, per ognuno di essi NiFi tiene traccia di una mappa di stringhe di attributi, coppia chiave/valore e il suo contenuto associato.
- FlowFile Processors: i processori eseguono effettivamente il lavoro, hanno in carico le operazioni di instradamento, trasformazione o mediazione dei dati tra i sistemi. I processori hanno accesso agli attributi di un determinato FlowFile e al suo flusso di contenuti, possono operare su zero o più FlowFiles in una determinata unità di lavoro e possono eseguire il commit o il rollback.
- Connection: le connessioni forniscono il collegamento effettivo tra i processori. Funzionano come code e consentono a vari processi di interagire a velocità diverse. Queste code possono essere priorizzate dinamicamente.
- Flow Controller: il controller di flusso è a conoscenza di come i processi collegano e gestiscono i thread e le allocazioni degli stessi utilizzati da tutti i processi. Il controller di flusso funge da intermediario che facilita lo scambio di FlowFiles tra processori.
- Process Group: un gruppo di processi è un insieme specifico di processi e relative connessioni, che possono ricevere dati tramite porte di input e inviare dati tramite porte di output. In questo modo, i gruppi di processi consentono la creazione di componenti completamente nuovi semplicemente aggregando altri componenti.

NiFi è un prodotto molto versatile e ricco di componenti e connettori (tra cui Solr) che rende molto agevole la gestione dei flussi e il data ingestion.

Alcune delle caratteristiche più interessanti includono la gestione dei flussi, la facilità d'uso, la sicurezza, l'architettura scalabile. Inoltre, possiede una user interface da cui schedulare e gestire i flussi in maniera più agevole.

4.8 APACHE SOLR

Apache Solr è una piattaforma di ricerca open source del progetto Apache Lucene. La sua caratteristica principale è la capacità di ricerca full text, hit highlighting, faceted search, raggruppamento dinamico, integrazione con le basi di dati, gestione di documenti (come documenti word e pdf). Fornisce una ricerca distribuita e la replicazione dell'indice.

Tra le feature principali di Solr elenchiamo:

- Funzionalità avanzate di ricerca full-text
- Capacità di abbinamento tra cui frasi, caratteri jolly, join, raggruppamenti e molto altro su qualsiasi tipo di dati
- Ottimizzato per traffico ad alto volume
- Interfacce aperte basate su standard: XML, JSON e HTTP
- Interfaccia utente amministrativa integrata e reattiva per semplificare il controllo delle istanze Solr
- Monitoraggio facile
- Esposizione di metriche tramite JMX
- Altamente scalabile e Fault Tolerant
- Costruito e collaudato su Apache Zookeeper, Solr rende facile scalare sia in verticale e che in orizzontale. Solr fa repliche, distribuzione, ribilanciamento e gestisce la Fault Tolerance.
- Indicizzazione Near Real-Time per permettere di accedere ai contenuti in qualsiasi momento.

-
- Architettura plugin estensibile, Solr pubblica estensioni ben definite che semplificano l'integrazione di plug-in di query e indicizzazione. Ovviamente, dato che è open source con licenza Apache, è possibile modificare qualsiasi parte del codice.

4.9 POSTGRESQL

PostgreSQL è uno tra i più robusti database open source di tipo relazionale. Il suo ruolo all'interno della soluzione è quello di essere il database principale per lo storage di dati operazionali application-specific, che in linea di massima non saranno oggetto di analytics o di processing.

4.10 MONGODB

Come altri database documentali No-SQL, MongoDB supporta la progettazione di Schema dinamici, consentendo ai documenti di una raccolta di avere campi e strutture diversi. Il database utilizza un archivio di documenti e un formato di scambio dati chiamato BSON che fornisce una rappresentazione binaria di documenti simili a JSON. Lo sharding automatico consente di distribuire i dati di una raccolta su più sistemi per la scalare orizzontalmente all'aumentare del volume di dati.

Tra le caratteristiche distintive di MongoDB figurano:

- Scalabilità: MongoDB ha le capacità per scalare agilmente su cluster con oltre 1.000 nodi, offrendo milioni di operazioni al secondo su oltre 100 miliardi di documenti e petabyte di dati.
- Alta Affidabilità: la replicazione fornisce ridondanza e aumenta la disponibilità dei dati. Con più copie di dati su diversi server di MongoDB, le repliche forniscono un alto livello di Fault Tolerance rispetto ad un'architettura single-server. In alcuni casi, le repliche possono fornire una maggiore capacità di lettura in quanto i client possono inviare operazioni di lettura a server diversi. E' possibile inoltre conservare copie aggiuntive per scopi dedicati, come il ripristino di emergenza, la creazione di report e il backup.
- Consistenza: MongoDB è "strongly consistent" by design, quindi se si esegue una scrittura, sarà sempre possibile leggere il risultato della scrittura appena eseguita dalle letture successive. La consistenza è ottenuta attraverso il paradigma single-master adottato da MongoDB. E' possibile indebolire il modello di consistenza di MongoDB al fine di aumentare il throughput, abilitando la lettura dai nodi replica secondari, trasformando MongoDB in un sistema "eventually consistent".

4.11 REDASH

Redash è una web application di data visualization e dashboarding che consente di scrivere le query nella loro sintassi naturale per interrogare le basi dati. L'applicativo è ricco di funzionalità come auto-completamento in tempo reale, e molte scorciatoie da tastiera, consente inoltre di creare snippet per elementi usati di frequente così da poterli riutilizzare. I risultati vengono memorizzati nella cache per un breve tempo e vengono pianificati i tempi di aggiornamento automatico per i risultati dei dati da cui vengono estrapolati i grafici.

È possibile accedere a dati provenienti da molte sorgenti dati, tra le quali spiccano le sorgenti no sql: ElasticSearch, MongoDB, Cassandra e Hadoop Hive. Consente inoltre l'accesso a molte sorgenti SQL-Like (MySQL, Sql-Server, Postgres, Oracle).

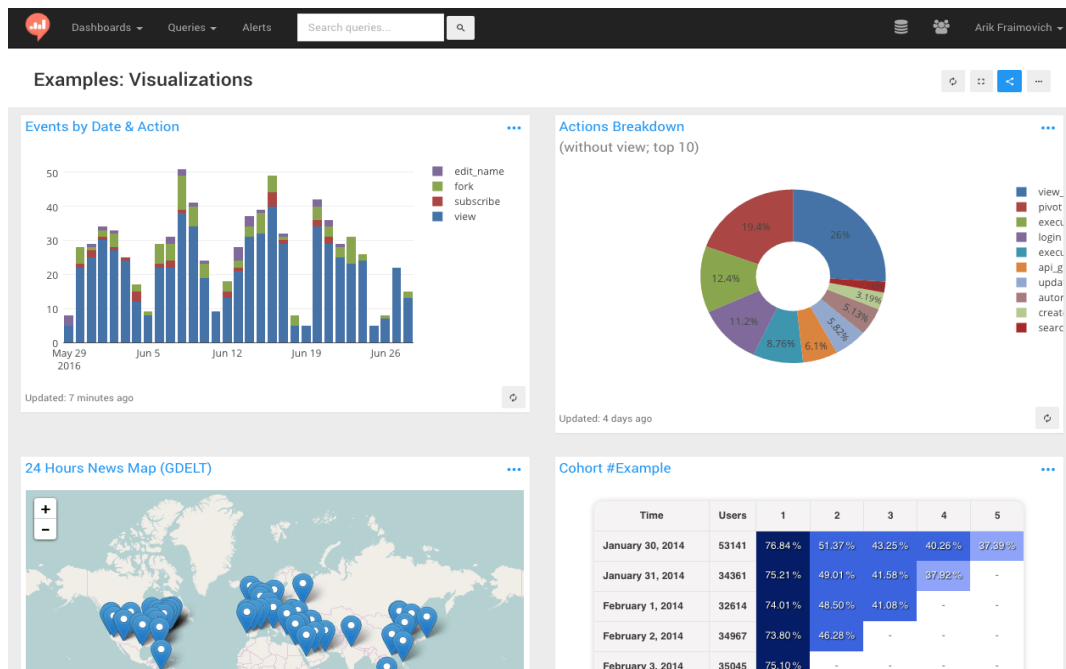


Figura 8: Dashboard di Redash

Le query e le visualizzazioni vengono create direttamente dalla web app ed è possibile utilizzare le query salvate per creare degli alert basati sul superamento di determinate soglie. Gli alert possono essere inviati sia tramite mail, sia tramite Slack o Hipchat.

Redash supporta autenticazione e ACL sia tramite login, sia tramite SAML/LDAP/OAUTH.

Le visualizzazioni prodotte possono essere annidate all'interno di altre, è inoltre possibile esportare i grafici in formato png.

Redash mette inoltre a disposizione delle API REST per effettuare sostanzialmente tutte le operazioni che possono essere effettuate da User Interface, rendendo possibile l'integrazione con soluzioni di reportistica custom.

4.12 EFK (ELASTICSEARCH, FLUENTD, KIBANA)

Kibana è lo strumento di data visualization dello stack EFK (elasticsearch/fluentd/kibana).

Viene distribuito come una web application attraverso la quale è possibile sia esplorare I dati, sia scrivere e salvare delle query (in modo visuale o attraverso la sintassi di Lucene). Le query salvate (visualization) possono essere annidate tramite l'utilizzo di iframe o aggregate in dashboard, anch'esse esportabili.

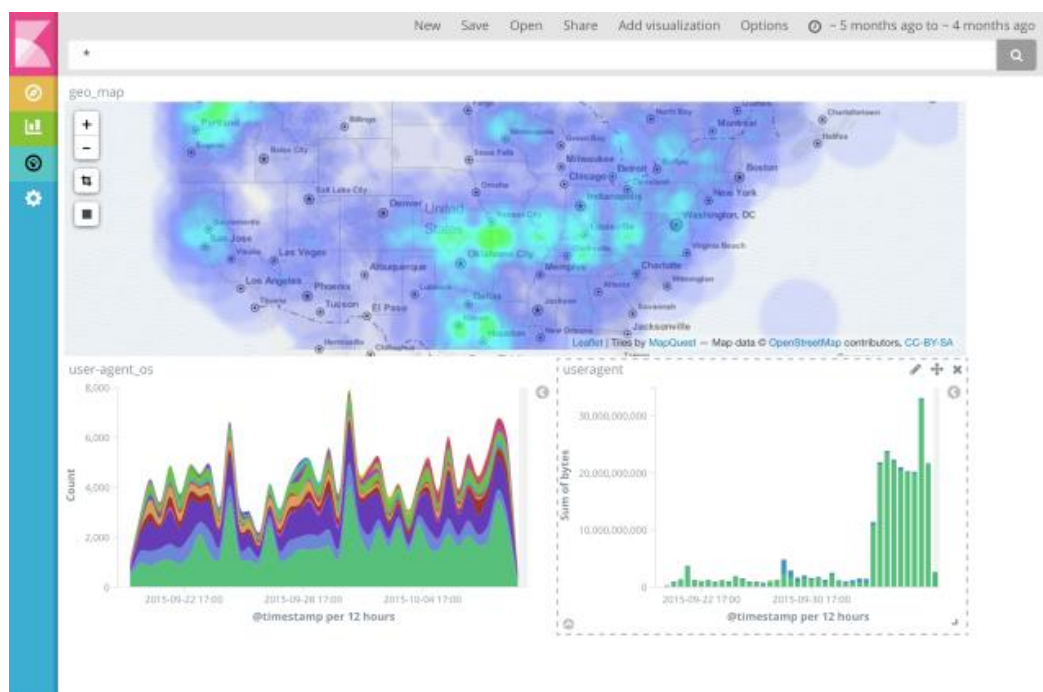


Figura 9: Dashboard di Kibana

Funzionalità quali export dei grafici, alerting e autenticazione sono disponibili solo attraverso *x-pack*, il pacchetto dei plugin a pagamento.

Kibana è in grado di accedere ai dati presenti solo ed esclusivamente su di un cluster di elasticsearch.

Kibana è estendibile attraverso un Sistema di plugin (nodejs/angular) attraverso i quali è possibile aggiungere nuovi tipi di visualizzazioni, nuove app (come ad esempio Timelion introdotta nella versione 5) o integrarsi nel ciclo di vita del prodotto, per aggiungere ad esempio autenticazione custom.

4.13 RANCHER

Kubernetes è un sistema di orchestrazione di contenitori open source per automatizzare la distribuzione, la scalabilità e la gestione di applicazioni containerizzate che aggiunge a tutti i vantaggi offerti dalle piattaforme di container, fra le quali docker, servizi per l'automatizzazione della configurazione, della gestione del failover e della scalabilità.

Alla soluzione si affianca Rancher, un Container Management Platform, che consente di semplificare la gestione e configurazione di un ambiente containerizzato offrendo, se necessario, un supporto esteso per le aziende.

Inoltre, è in grado di gestire in modo del tutto trasversale servizi cloud esposti dai maggiori cloud provider quali GKE (Google container Engine), EKS (Elastic Container Engine) e AKS (Azure Container services) o importare cluster kubernetes esistenti.

5 IPOTESI DIMENSIONAMENTO

Nei paragrafi successivi sono riportate le ipotesi di dimensionamento di un'infrastruttura virtualizzata generica, che può essere utilizzata come base di partenza per la selezione della specifica piattaforma tecnologia target da utilizzare in fase di effettiva implementazione.

5.1 ANALISI VOLUMI DELL'AS-IS

In questo paragrafo vengono riportate le informazioni sui volumi gestiti dalle attuali distribuzioni SBNWeb presenti nei poli, utilizzate per stimare il dimensionamento riportato nei paragrafi successivi.

Descrizione	Valore	Nota
numero di poli	26	+78 poli prospect
numero di biblioteche	1428	+4500 biblioteche prospect
numero di utenti totali	utenti professionali: 5.000 utenti lettori: >100.000	Per gli utenti lettori non abbiamo una stima precisa dato che non tutti i poli utilizzano SBNWeb per erogare servizi. In SBNCLOUD e per i poli geograficamente affini (es. poli con base su Roma (BVE, IEI, RMR ecc)) ci aspettiamo un certo grado di sovrapposizione
numero documenti/copie	polo grande: 2.500.000/4.000.000 polo medio: 1.000.000/2.000.000 polo piccolo: 300.000/500.000	I documenti sono entità descritte. Per ogni entità esistono da 0 a N copie. In SBNCLOUD molti documenti ora descritti su più poli saranno accorpati, ma non le copie che sono segregate per biblioteca
dimensione database	polo grande: 25 GB polo medio: 10 GB polo piccolo: 3 GB	Ci sono molte variabilità che dipendono dalla natura del materiale descritto, ma su alcuni poli da noi gestiti si evidenzia un'occupazione media per tipologia di entità: doc: descrizione/legami: 4KB + inventari(copie): 4KB utenti lettori: 2KB utenti professionali: 5KB richieste servizio: 2KB
numero utenti contemporanei	polo grande: 100 polo medio: 60 polo piccolo: 20	media per polo.
interazioni con indice centrale	polo grande: 9000 polo medio: 4000 polo piccolo: 1500	per polo <u>al giorno</u>
numero di poli per tipologia	polo grande: 6 polo medio: 5 polo piccolo: 15	attuale composizione dei 26 poli attuali
dimensione indice solr per tipologia	polo grande: 10 polo medio: 5 polo piccolo: 1	attuale dimensione degli indici solr relativi alle installazioni opac presenti nei 26 poli
stima dimensione totale indice solr	100 GB	stima della dimensione totale degli indici solr della componente opac nelle attuali installazioni presenti
stima dimensione totale db	250 GB	stima della dimensione totale dei db di SBNWeb attualmente presenti nei 26 poli.
dimens.storage (SBNWeb)	3120 GB	dimensione degli storage per i 26 poli
dimens.storage (Opac)	1200 GB	dimensione degli storage previsti per le 10 installazioni opac

5.2 ASSUNZIONI

Di seguito sono riportate le assunzioni utilizzate per stimare il dimensionamento.

N.	Assunzione
1	L'architettura dell'infrastruttura sarà consolidata durante la fase di progettazione di dettaglio, di conseguenza il dimensionamento potrà essere soggetto a ricicli successivi in base alla tecnologia scelta (VMWare, Nutanix, AWS, Azure, ...)
2	Il dimensionamento non include l'area di backup e l'area di monitoraggio applicativo
3	Non sono inclusi nel presente dimensionamento gli apparati di rete fisici necessari a stabilire un canale di comunicazione adeguato con i sistemi esterni (switch, gateway, routers, firewall, ecc.).
4	Non è incluso nel dimensionamento un sistema IDPS antintrusione in grado di monitorare e respingere, attraverso l'utilizzo di tecniche di sicurezza attiva, attacchi esterni provenienti da Internet.
5	Si assume l'adozione di antivirus centralizzato sulle componenti esposte a maggior rischio.
6	Si assume che tutte le componenti hardware siano ridondate per evitare Single Points Of Failure.
7	Il dimensionamento non include i server di posta elettronica (si prevede siano utilizzati eventuali installazioni preesistenti)
8	Nel dimensionamento non sono state prese in considerazione la business continuity e il disaster recovery.

5.3 SCHEMA LOGICO DELLA SOLUZIONE

Nella figura seguente vengono riportate le tipologie di macchine virtuali ipotizzate nel dimensionamento raggruppate per servizio. La soluzione prevede che i moduli WSO2, Zeebe, Redash, Kafka, Nifi ed i Microservizi Spring Boot siano ospitati come container docker all'interno del servizio Rancher, mentre il motore CMS viene erogato in HA con un cluster e uno stack di risorse dedicato. In questa soluzione è stato ipotizzato un servizio OpenLdap dedicato che potrebbe eventualmente essere sostituito da un analogo servizio preesistente nell'infrastruttura di origine.

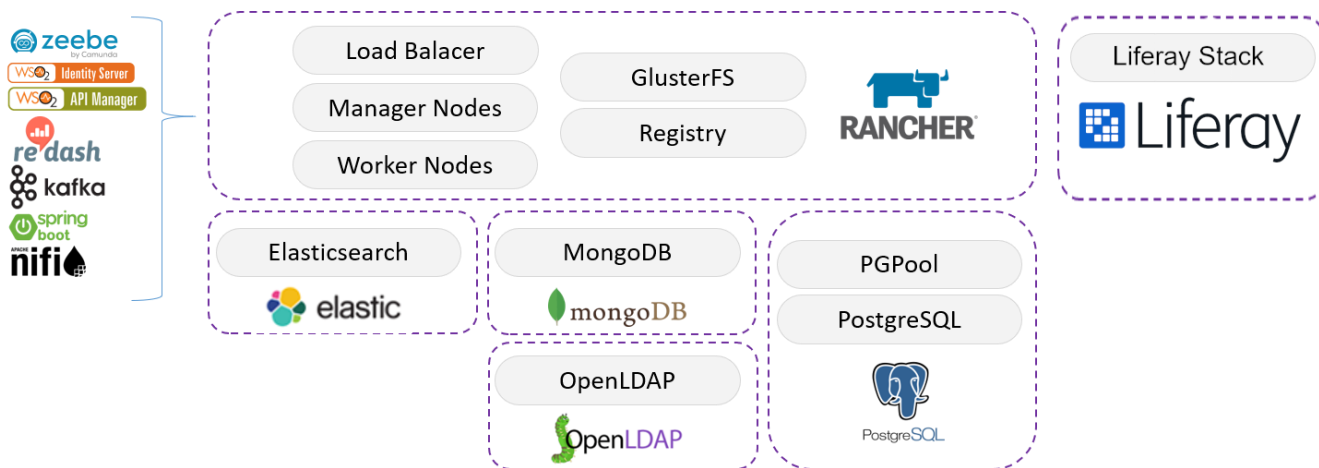


Figura 10: Diagramma Logico per Raggruppamenti Risorse

5.4 IPOTESI DIMENSIONAMENTO MINIMALE

In questa tabella viene riportata l'ipotesi per il dimensionamento minimale dell'infrastruttura.

Le risorse logiche sono indicate per singola macchina.

Ruolo	Num.	CPU (vCPU)	RAM (GB)	Storage (GB)	Software	Note
DNS	1	n.a.	n.a.	n.a.	Server DNS	In base al servizio DNS disponibile presso l'Istituto
Liferay Load Balancer	2	4	8	40	Ubuntu Apache	
Liferay Web Server	2	2	4	140	Ubuntu Tomcat Liferay CE	
Liferay DB Postgresql	2	8	16	300	Ubuntu Pgpool II Postgresql	
Liferay File Storage	1	---	---	300	---	NAS o NFS share

Ruolo	Num.	CPU (vCPU)	RAM (GB)	Storage (GB)	Software	Note
Rancher Load Balancer	2	4	8	100	Ubuntu ProxyHA	
Rancher Manager Nodes	3	4	8	150	Ubuntu Rancher Manager Node	
Rancher Worker Nodes	3	16	32	250	Ubuntu Rancher Worker Node	Di seguito i servizi ospitati su Rancher: WSO2, Zeebe, Kafka, Web Server (SPA di front-end), Redash, Fluentd, Dibana, Nifi, micro-servizi (Spring Boot)
Rancher GlusterFS	3	4	8	300	Ubuntu GlusterFS	
Rancher Registry	2	4	8	100	Ubuntu Docker Registry	
PostgreSQL (PGPOOL)	2	2	6	100	Ubuntu Pgpool II	
PostgreSQL (Server)	2	8	16	300	Ubuntu PostgreSQL	
MongoDB	3	8	32	300	Ubuntu MongoDB	
Elasticsearch	3	8	32	300	Ubuntu Elastic	Un solo cluster è utilizzato per l'indicizzazione sia dei Log applicativi che delle pagine web e dei contenuti di Liferay.
OpenLDAP	2	4	8	100	Ubuntu ZooKeeper OpenLDAP	Si ipotizza di utilizzare macchine dedicate o un server preesistente per memorizzare le credenziali utente
Solr	3	8	32	300	Ubuntu ZooKeeper Solr	
TOTALE	35	216	580	7460		

5.5 IPOTESI DIMENSIONAMENTO A REGIME

In questa tabella viene riportata l'ipotesi per il dimensionamento a regime dell'infrastruttura, è stata considerata un'adesione del 100% degli attuali poli piu' un ulteriore incremento del 30% di nuovi poli.

Le risorse logiche sono indicate per singola macchina.

Ruolo	Num.	CPU (vCPU)	RAM (GB)	Storage (GB)	Software	Note
DNS	1	n.a.	n.a.	n.a.	Server DNS	In base al servizio DNS disponibile presso l'Istituto
Liferay Load Balancer	2	4	8	100	Ubuntu Apache	
Liferay Web Server	4	8	16	300	Ubuntu Tomcat Liferay CE	
Liferay DB Postgresql	2	8	16	800	Ubuntu Pgpool II Postgresql	
Liferay File Storage	1	---	---	800	---	NAS o NFS share in alta affidabilità
Liferay Elasticsearch	3	8	32	300	Ubuntu Elastic	Cluster dedicato per l'indicizzazione di siti web e contenuti del portale Liferay. <i>Nota: a seguito di verifiche in fase di progettazione potrebbe essere sufficiente un solo cluster Elasticsearch condiviso con la componente di log management.</i>
Rancher Registry	2	4	8	100	Ubuntu Docker Registry	
PostgreSQL (PGPOOL)	2	2	6	100	Ubuntu Pgpool II	
PostgreSQL (Server)	4	8	16	500	Ubuntu PostgreSQL	
MongoDB	5	8	32	300	Ubuntu MongoDB	
Elasticsearch	5	8	32	300	Ubuntu Elastic	Cluster Elasticsearch utilizzato dalla componente di log management per l'indicizzazione dei file di log.
OpenLDAP	2	4	8	100	Ubuntu ZooKeeper OpenLDAP	Si ipotizza di utilizzare macchine dedicate o un server preesistente nell'infrastruttura ICCU per memorizzare le credenziali utente
Solr	5	8	64	300	Ubuntu ZooKeeper Solr	
TOTALE	38	252	956	11800		

