KeySuite 6.x

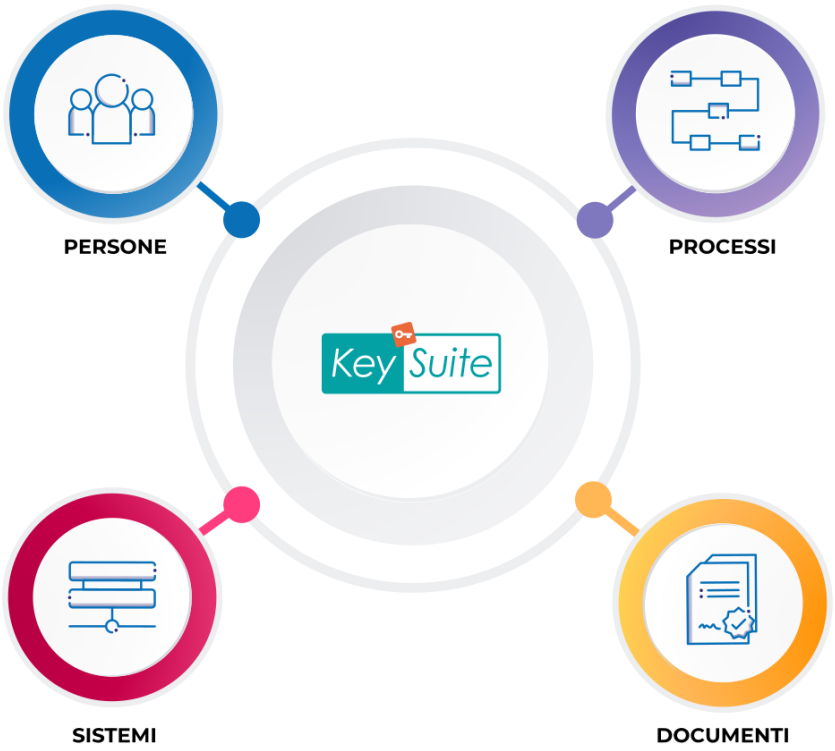
Architettura Applicativa

Sommario:

1. Introduzione 3
2. Tecnologie software 4
3. Architettura 5
   - 3.1. Servizi 6
     - 3.1.1. Desktop 6
     - 3.1.2. Docer 7
     - 3.1.3. BPM 7
     - 3.1.4. Portal 7
     - 3.1.5. Payment 8
     - 3.1.6. Form 8
     - 3.1.7. Flow Designer 8
   - 3.2. Security 8
     - 3.2.1. Identificazione e Autorizzazione 8
     - 3.2.2. Autenticazione e SSO 9
     - 3.2.3. OWASP Compliance 9
   - 3.3. Multi-Tenancy 10
4. Opzioni di Deployment 11
   - 4.1. Cloud Deployment 11
     - 4.1.1. Public Cloud 11
     - 4.1.2. On Premise Cloud 12
     - 4.1.3. Managed Cloud 12
   - 4.2. Database 12

1. Introduzione

KeySuite è la piattaforma, disponibile sia "on premise" sia in Cloud, nata per affiancare PA e organizzazioni in genere nel processo di semplificazione e digitalizzazione dei procedimenti amministrativi e organizzativi.



La piattaforma fornisce strumenti per la configurazione e la realizzazione di soluzioni applicative personalizzate che permettono la modellazione, l'esecuzione e il monitoraggio dei processi di business, con particolare riferimento a quei processi in cui è richiesta un'efficace comunicazione e interazione tra le persone, la produzione di documentazione e l'interazione con sistemi esterni.

I principali servizi e applicazioni *"out of the box"* sono:

- **Sistema per l'orchestrazione dei processi** (Business Process Management, **BPM**) un servizio che fornisce una soluzione completa per l'automazione, la gestione e il monitoraggio dei processi.
- **Sistema di gestione documentale** (Document Management System, **DMS**) opportunamente esteso per rispettare i dettami di una corretta gestione archivistica dei documenti nel rispetto della normativa.
- **Console di Back-Office**, applicazione web a supporto di amministratori e operatori degli uffici preposti che fornisce, in un contesto completamente personalizzabile, tutti gli strumenti per una completa interazione con i diversi procedimenti amministrativi e i documenti.

- **Portale dei Servizi On-Line**, applicazione web facile da usare, veloce e diretta, completamente personalizzabile, rivolta a cittadini, imprese ed utenti esterni all'organizzazione, per accedere al catalogo dei servizi e interagire con gli uffici di Enti pubblici o privati tramite sportelli virtuali polifunzionali.
- **Flow Designer**, applicazione web based molto intuitiva che consente di modellare un qualunque processo/procedimento evidenziandone la struttura e la scomposizione in attività elementari. Il designer permette inoltre, attraverso l'integrazione del servizio **Form Manager**, di disegnare i form accessibili e previsti nel procedimento. Queste caratteristiche consentono di utilizzare la terminologia comune alla PA o all'organizzazione, rendendo il disegno dei processi facilmente leggibile e riusabile, ma al tempo stesso conforme alla notazione standard BPMN 2.0, lo standard definito da OMG (Object Management Group) con cui descrivere graficamente e funzionalmente i processi di business per facilitarne la comprensione e l'implementazione a tutti gli stakeholder (business analysts, business administrators, developers).

In aggiunta KeySuite è anche un **Web Framework** general-purpose attraverso cui implementare nuove applicazioni web che eventualmente interagiscono con servizi e basi dati legacy dell'organizzazione, o integrare e personalizzare, sia da un punto di vista funzionale che estetico, applicazioni web di terze parti, al fine di presentare una ambiente uniforme e centralizzato per l'accesso a tutte le funzionalità di business.

2. Tecnologie software

Tutta la piattaforma KeySuite è basata su tecnologie software **open source** consolidate e standardizzate, aggiornabili indipendentemente a seguito di eventuali patch di sicurezza o correzione di bug. L'ambiente di runtime del software è nella maggior parte dei casi e per tutti i servizi proprietari **Java OpenJDK 11** e **Spring Boot 2.2.4**.

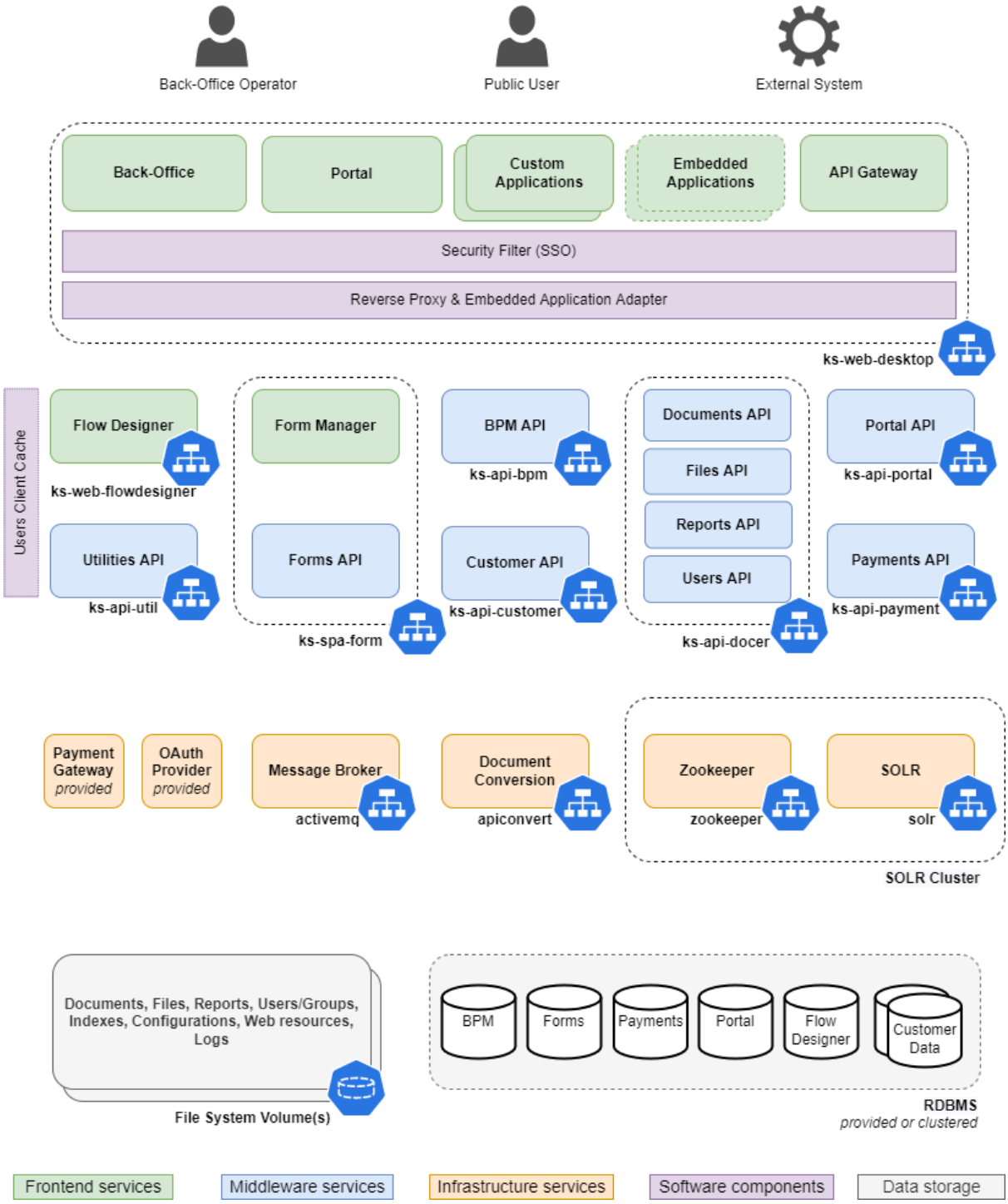
I servizi core, BPM e DMS, sono basati rispettivamente sulle tecnologie **JBPM 7** e **Solr Cloud 8**, e rappresentano i maggiori punti di forza della piattaforma.

Il motore JBPM è un workflow engine scritto in Java per la gestione dei processi di business rilasciato da Red Hat. Il prodotto permette di modellare, eseguire e monitorare i processi durante l'intero ciclo di vita.

Come repository e motore di indicizzazione documentale si è scelto di utilizzare la tecnologia open source Solr Cloud in quanto presenta un'architettura estremamente flessibile che opportunamente configurata e personalizzata, gestisce indicizzazione, persistenza, sicurezza applicata alla ricerca, modelli flessibili e personalizzabili di definizione dei metadati dei documenti.

3. Architettura

L'architettura della soluzione KeySuite 6 è una architettura **cloud-native** a **microservizi** portabile, implementata nel rispetto del massimo disaccoppiamento tra le diverse aree funzionali disponibili: ogni servizio della piattaforma svolge il proprio specifico ruolo rendendo disponibile le sue funzionalità alle altre applicazioni/servizi della piattaforma.



Nel diagramma è riportata l'architettura logica e di deployment della soluzione, che descrive i principali componenti tra i quali i servizi, distinti per layer di appartenenza:

- **Frontend Services**

Sono i servizi che implementano l'interfaccia di accesso al sistema attraverso

  - **Applicazioni web** dotate di una interfaccia (UI) per l'interazione con gli utenti del sistema
  - **Rest API** che espongono all'esterno i servizi middleware per le applicazioni web di tipo SPA (Single Page application) o per l'integrazione di sistemi esterni di terze parti con la piattaforma KeySuite.

- **Middleware Services**

Sono i servizi core della soluzione che implementano la **domain business logic**. Ogni servizio è completamente indipendente e dedicato ad uno specifico dominio di business, ha una propria base dati eventuale, espone una Rest API e collabora con gli altri servizi necessari alla implementazione delle funzionalità di business.

- **Infrastructure Services**

Sono i servizi infrastrutturali che implementano aspetti di **cross-domain logic** e di **integrazione a basso livello** quali l'indicizzazione di documenti, la conversione di documenti, l'infrastruttura per lo scambio di messaggi, servizi esterni di pagamento (e.g. PMPay, PagoPA) e provider esterni di autenticazione OAuth (e.g. Keycloak, Google).

Gli stessi **servizi logici** individuati dai singoli blocchi nel diagramma sono fattorizzati, nella maggior parte dei casi in rapporto uno a uno, in **servizi deployabili**, coincidenti con l'applicazione eseguibile a runtime, ossia un *service* in ambito Kubernetes o un *processo Java* eseguito su una VM o host fisico, a seconda della modalità di deployment scelta.

3.1. Servizi

Di seguito è riportata una breve descrizione dei servizi principali e delle loro funzionalità.

3.1.1. Desktop

Il servizio **Desktop** (*ks-web-desktop*) implementa alla base un web server e relativo web framework in grado di esporre all'esterno un numero arbitrario di applicazioni web distinte.
Built-In con la piattaforma KeySuite sono disponibili le due applicazioni **Back-Office** e **Portal**, già precedentemente descritte.
Back-Office e Portal, sono a loro volta ampiamente personalizzabili ed estendibili con nuove pagine e funzionalità per soddisfare le esigenze dell'organizzazione.

Il meccanismo di accesso alle diverse applicazioni è basato su un avanzato meccanismo di mapping della url pubblica, sia a livello di host che a livello di path, in modo da poter esporre applicazioni su url completamente differenti o contesti differenti della stessa url, il tutto in maniera trasparente all'utente che accede.
Il servizio mette inoltre a disposizione un altrettanto avanzato meccanismo di *reverse proxy* che implementa un **Api Gateway** per l'accesso a tutti i servizi middleware della piattaforma o che può essere usato per proxare applicazioni legacy e di terze parti.

Proprio in quest'ultimo caso è anche possibile manipolare a runtime le pagine ospitate per inserirle (**Embedded Applications**) nella struttura dell'applicazione ospite in maniera armonica sia dal punto di vista estetico, preservando ad esempio i classici header e footer dell'applicazione principale, che funzionale, condividendo lo stesso meccanismo di autenticazione (SSO, descritto in seguito). In questo modo le organizzazioni hanno la possibilità di costruire *portali applicativi* che centralizzano ed uniformano l'accesso di tutte le applicazioni.

3.1.2. Docer

Il servizio **Docer** (*ks-api-docer*) implementa un **sistema di gestione documentale** arricchito da funzionalità archivistiche.
Esso offre una completa gestione dei documenti, del titolario di classificazione, dei fascicoli, delle cartelle e di eventuali anagrafiche. Per ogni entità è possibile gestire le operazioni basilari (CRUD), impostare i permessi di accesso (ACL), effettuare ricerche avanzate e, tramite funzioni specifiche, determinare relazioni (classificazione, fascicolazione, profilazione, etc.).
Alcune operazioni archivistiche sono eventualmente integrate con sistemi esterni di registrazione particolare (protocollo, contabilità, verticali di registro, etc.).
È possibile associare ad un documento o ad un fascicolo una specifica *tipologia* che permette di associare all'entità dei metadati specifici.

3.1.3. BPM

Il servizio **BPM** (*ks-api-bpm*), insieme al sistema documentale, rappresenta il cuore della piattaforma KeySuite. Implementa il **motore di processo** che orchestra tutte le attività durante l'esecuzione di una istanza di processo.
Il servizio è inoltre responsabile anche di tutta la gestione "offline" dei processi permettendo il deploy di nuovi processi e la loro configurazione.

3.1.4. Portal

Il servizio **Portal** (*ks-api-portal*), è un piccolo servizio dedicato per la gestione e la fruizione di un **catalogo di procedimenti amministrativi** da parte degli utenti pubblici (e.g. i cittadini). Permette di gestire un catalogo di procedimenti raggruppato per aree tematiche e categorie, gestire la scheda di dettaglio, vincoli di apertura e chiusura del servizio, e altri parametri di configurazione. Permette inoltre di gestire le bozze non ancora inviate, lo storico dei procedimenti avviati e tutti gli aspetti di monitoraggio e interazione con il processo da parte del cittadino.

3.1.5. Payment

Il servizio **Payment** (*ks-api-payment*) è un servizio dedicato ai pagamenti. Si tratta di un servizio “adapter” che uniforma e standardizza l’accesso a vari **payment gateway** di terze parti (e.g. **PagoPA**, **PmPay**, etc.), pluggabili e configurabili all’interno della soluzione KeySuite.

3.1.6. Form

Il servizio **Form** (*ks-spa-form*) gestisce in maniera completamente autonoma i form deployati con la soluzione. Esso si compone sia di una applicazione web, dedicata a Business Analyst e Developer, per la definizione e modellazione visuale del form (*design time*), sia di una API Rest per l’accesso al repository e soprattutto per il rendering a *runtime* dei form, che solitamente sono poi ospitati (*embedded*) nelle pagine delle applicazioni Desktop (e.g. Portal, Back-Office, etc.).

3.1.7. Flow Designer

Il servizio **Flow Designer** (*ks-web-flowdesigner*) è a disposizione di utenti specifici (Business Analyst, Developers) per permettere la definizione di processi seguendo la notazione BPMN2.

il Flow Designer, tipicamente deployato solo in ambienti di sviluppo, è completamente integrato con il BPM server ed il Form Manager permettendo un *testing* a *design time* dei processi definiti e delle form associate all’avvio di istanze e lo svolgimento di *human task*.

3.2. Security

3.2.1. Identificazione e Autorizzazione

Tutti i componenti della KeySuite gestiscono meccanismi di applicazione delle autorizzazioni indipendenti e consoni ai dati gestiti, ma condividono un unico meccanismo di identificazione dell’utente e dell’individuazione dei ruoli applicativi da utilizzare per autorizzare le operazioni sui dati.

In particolare il componente Desktop è dotato di diverse opzioni di login che permettono in modo esplicito (inserendo username/password) o implicito (tramite integrazione con un Identity provider ad esempio) di identificare l’utente (determinare quindi la *username* con cui impersonificare tutte le operazioni da compiere applicativamente).

Il Desktop è poi dotato di un meccanismo di *reverse proxy* delle chiamate che permette di filtrare le invocazioni di API verso gli altri componenti arricchendole di un token JWT contenenti le informazioni fondamentale di identificazione (*username*, *tenant,* ed eventuale *sub-tenant* con cui solitamente si intendono Ente ed AOO).

Gli utenti, i metadati ed i gruppi/ruoli ad essi associati sono persistiti nel sistema documentale e messi a disposizione degli altri servizi. Un servizio (documentale, bpm, etc.) potrà quindi utilizzare tali informazioni per autorizzare le operazioni sui dati.

C'è una profonda differenza nella KeySuite tra gruppi e ruoli associati agli utenti.
I primi (i gruppi) ricalcano tipicamente la struttura organizzativa di un Ente, e sono l'ente stesso, le AOO in esso contenuto e tutte le UO (Unità Organizzative) censite all'interno di ogni AOO.
I secondi (i ruoli) fanno parte dell'impianto del sistema, censiti in base ai ruoli applicativi dei vari componenti e possono avere un significato **locale** (applicato solo rispetto ad un ramo della struttura organizzativa) oppure **globale** (applicato indipendentemente dalla struttura organizzativa).

Utenti, gruppi e ruoli vanno tenuti sincronizzati con i dati organizzativi, utilizzando diversi meccanismi disponibili di importazione e sincronizzazione.

3.2.2. Autenticazione e SSO

KeySuite offre un meccanismo interno di autenticazione basato su password, è tuttavia ragionevole delegare ad un sistema esterno uno o più aspetti dell'autenticazione e/o dell'autorizzazione.

Sono disponibili due meccanismi principali di SSO.
Il primo, molto semplice, è delegare la verifica della password ad un sistema LDAP esterno.
In questo caso è sempre necessario adottare altri meccanismi per sincronizzare la base dati degli utenti e dei gruppi associati, leggendo utenti e gruppi da un file CSV, da una tabella DB o da LDAP.
L'altro meccanismo, più generico, è quello di integrare un sistema compliant con **OIDC** (**OpenID Connect**) in modo da ricevere da esso le informazioni di identificazioni ed eventualmente anche quelle di autorizzazione, mappando gruppi e ruoli dell'identity provider con quelli censiti nella piattaforma KeySuite.
Esiste inoltre la funzionalità di sincronizzare in tempo reale gli utenti a seguito dell'autenticazione esterna (su Keycloak, WSO2, Google, Facebook od ogni altro provider OIDC).

3.2.3. OWASP Compliance

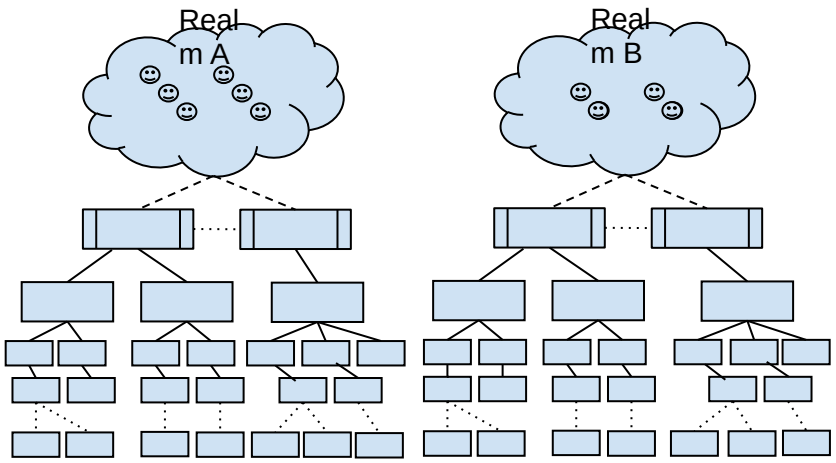
Come illustrato, in KeySuite, ogni interazione con l'esterno avviene attraverso il servizio Desktop.
Ogni servizio è dotato della propria sicurezza applicativa e tutte le interazioni avvengono tramite chiamate HTTP alle API Rest.
Il servizio Desktop è stato sottoposto a rigidi controlli di sicurezza superando i diversi livelli di compliance con i requisiti OWASP, garantendo un'alta affidabilità della soluzione anche quando esposta su internet.

3.3. Multi-Tenancy

KeySuite in tutti i suoi servizi gestisce i dati in **tenant** indipendenti e mutualmente isolati permettendo una installazione unica per gestire più organizzazioni (più Enti).
La base dati degli attori (utenti, gruppi e ruoli) relativa ad un tenant può essere condivisa eventualmente tra più tenant (tra più enti), introducendo il concetto di “realm”, che rappresenta quindi un contenitore indipendente di attori condivisibile tra più tenant.

Sotto un tenant (un Ente associato quindi ad un realm) si sviluppa la struttura organizzativa composta da una lista di AOO e sotto ogni AOO si sviluppa un albero di UO.

All’interno del realm sono censiti gli utenti che sono quindi eventualmente condivisi tra più tenant.



Tipicamente un applicativo (ad esempio un portale o un backoffice) è **attestato** tramite configurazione ad uno specifico Ente o AOO ed ogni azione compiuta dall’utente è quindi circoscritta al tenant identificato in fase di login.

4. Opzioni di Deployment

4.1. Cloud Deployment

La scelta di elezione, e dunque raccomandata, è il deployment in cloud, inteso come cluster basato su tecnologia **Kubernetes**.
Questa modalità di deployment permette infatti di massimizzare i benefici dell'architettura a microservizi che costituisce la piattaforma KeySuite.
Il vantaggio principale rispetto al deployment su VM (esposto nel paragrafo successivo) è la **flessibilità e la dinamicità di allocazione delle risorse** insita nel cluster che permette di dimensionare il deployment con la granularità del singolo servizio, sia in termini di risorse hardware che in termini di numero di repliche, al fine di sostenere il massimo carico di lavoro previsto e nel contempo ottimizzare il consumo di risorse.

Inoltre la natura stessa del cluster garantisce di per sé la continuità del servizio a fronte di fault di un nodo (**High Availability**) ed il corretto bilanciamento del carico (**Load Balancing**) senza particolari accorgimenti se non la corretta configurazione dei servizi e delle repliche di ciascuno di essi.

Infine, un altro vantaggio diretto che scaturisce dal deploy in cluster è la possibilità di **gestire e monitorare i servizi da un'unica console centralizzata** per accedere a log, statistiche e metriche di utilizzo ed individuare e risolvere in maniera più efficiente eventuali problematiche legate al software o alle performance.

Nell'ambito cloud proponiamo fondamentalmente tre approcci: **public**, **on premise** e **managed**.

4.1.1. Public Cloud

Il **cloud pubblico** è la modalità più comune di deployment. *Sia l'infrastruttura che il provisioning del cluster Kubernetes sono gestiti da un provider di terze parti attraverso il modello **KaaS*** (Kubernetes as a Service). I principali provider e ambienti supportati sono **Amazon EKS**, **Google GKE**, **Azure AKS**.

Questo approccio è consigliato quando il cliente non dispone già di un cluster o di una infrastruttura proprietaria, né di un reparto IT con le competenze necessarie per la sua gestione.
Ha dunque il vantaggio di esonerare il cliente dalla gestione dell'infrastruttura con conseguente riduzione di costi e tempi di messa in esercizio, e soprattutto ha il vantaggio di poter scalare in linea teorica senza limiti, acquistando le risorse necessarie e pagando solo per l'uso effettivo.

Gli svantaggi potrebbero risiedere nei costi di servizio che sono per necessità più elevati rispetto a qualsiasi soluzione on premise a regime.

4.1.2. On Premise Cloud

Con **cloud on premise** si intende nello specifico *il provisioning e la manutenzione del cluster Kubernetes sui sistemi del cliente, da parte del dipartimento IT dello stesso*. Per quanto riguarda l'infrastruttura, questa è dunque normalmente anch'essa gestita su datacenter proprietario o opzionalmente acquistata da un provider di terze parti (IaaS).

A meno che il cliente non gestisca già una tale infrastruttura ed un cluster Kubernetes proprietari, dove sono ad esempio deployate altre applicazioni legacy dell'organizzazione, la soluzione on premise non è mai consigliata per ovvi motivi legati ai costi e alle competenze necessarie all'implementazione e manutenzione di infrastruttura e cluster Kubernetes.
Dal punto di vista della scalabilità abbiamo poi necessariamente una minore flessibilità dove un'eventuale necessità di maggiori risorse comporta solitamente l'acquisto e la gestione di nuovo hardware, fisico o virtuale che sia.

Resta però un'opzione praticabile quando norme e regolamenti stringenti impongono che i dati e/o i servizi aziendali gestiti debbano risiedere nel perimetro dell'azienda, o che non siano nella disponibilità di terzi, specialmente se residenti in paesi con norme sulla privacy più lasche di quelle comunitarie.
In questi casi, comunque, il provisioning e la manutenzione di un cluster ad hoc per il cliente è di norma al di fuori dei servizi da noi erogati.

4.1.3. Managed Cloud

Con **cloud gestito** intendiamo una *soluzione che prevede il deploy di una istanza della piattaforma KeySuite dedicata al cliente in un cluster Kubernetes gestito dal fornitore*.

Il servizio proposto è basato su una infrastruttura cloud pubblica (IaaS) presso un provider di fiducia sulla quale i nostri tecnici gestiscono provisioning e manutenzione di un cluster Kubernetes opportunamente configurato e dimensionato per ospitare le installazioni KeySuite dei nostri clienti.
Il cloud gestito è inoltre affiancato da procedure di deployment standardizzate e strumenti di monitoraggio ottimizzati per la piattaforma KeySuite.

Dunque come nell'opzione cloud pubblico il cliente è esonerato dalla gestione dell'infrastruttura e del cluster Kubernetes e analogamente ad un cloud on premise ha la possibilità di usufruire di una infrastruttura semidedicata, riuscendo nel contempo, a fronte di un corretto dimensionamento delle risorse, di ottimizzare i costi di esercizio.

4.2. Database

Nell'architettura KeySuite la base dati è vista come componente infrastrutturale indipendente sia dal punto di vista logico che di deployment. Il database è infatti il servizio che spesso il cliente possiede e gestisce auto-

nomamente all'interno della propria infrastruttura o che necessita di requisiti di ridondanza e clustering complessi da implementare e manutenere.
Abbiamo dunque le seguenti opzioni di deployment.

- **On Premise**

Quando il database è deployato nell'infrastruttura del cliente e gestito dal proprio dipartimento IT. In questo caso solitamente anche il resto dei servizi è deployato nell'infrastruttura interna per ovviare a problemi di connessione e sicurezza degli accessi alla base dati. In alternativa infatti deve essere concessa la connessione (SSL) a determinati indirizzi IP pubblici e statici.

- **Cloud**

Quando il cliente non possiede una infrastruttura interna e/o sono necessari requisiti di ridondanza e alta disponibilità. In questo caso il database è sempre raggiungibile dall'esterno per definizione.

È importante notare che per una installazione ottimale di KeySuite sono necessarie due istanze di database (database engine) separate, una per gestire i dati di runtime, ed un'altra per gestire il dato di storico. In questo modo la base dati di runtime può rimanere più leggera e garantire performance migliori.

KeySuite è compatibile con i seguenti RDBMS:

- MySql 5.7
- SQL Server 2017+
- PostgreSQL 9+
- Oracle 12c+