SBDIO I4.0
Big Data
for Industry

Rapporto di analisi e fattibilità e potenzialità di un servizio automatico di “business consulting” per servizi B2B focalizzati su smart product

Code	O5.1
Date	30/04/2020
Type	Confidential
Participants	UNIFE
Authors	Marco Alberti (UNIFE), Marco Govoni (UNIFE), Cesare Stefanelli (UNIFE), Mauro Tortonesi (UNIFE)
Corresponding Authors	Mauro Tortonesi

SBDIO I4.0
Big Data
for Industry

Sommario

1. Smart Product e Servitizzazione

Industry 4.0 è un processo estremamente innovativo nato ormai diversi anni fa, per portare le nuove tecnologie ICT all'interno delle industrie. Tradizionalmente si riferisce a impianti intelligenti dove tutti i componenti/macchine, eterogenei tra loro, sono interconnessi a tal punto da proporre nuovi scenari per la gestione degli shop floor. Inoltre, grazie all'enorme evoluzione dell'Internet Of Things (IoT), l'area degli **Smart Product** ha iniziato a essere di forte interesse per tutte le aziende coinvolte nel processo di sviluppo; questo trend comporta l'ampliamento dei confini di un impianto industriale anche nel mondo esterno, per macchinari smart distribuiti ovunque al di fuori di un impianto.

Il concetto di Smart Product abilita la realizzazione di una gamma di servizi totalmente nuova e innovativa, dal controllo remoto dei dispositivi fino a sofisticate funzioni adattabili al contesto operativo. In questo scenario, uno dei servizi più promettenti è sicuramente l'e-Maintenance, dove le capacità di gestione remota dei macchinari consentono al personale tecnico di interagire con i propri dispositivi, identificando il loro stato attuale e quello passato, ma anche per la modifica delle configurazioni e l'aggiornamento dei componenti software in esecuzione.

Questa innovazione consente innanzitutto di risolvere i problemi basilari che possono capitare durante il funzionamento dei macchinari, ma anche di gestire le anomalie più critiche, identificando le componenti problematiche e programmando l'intervento mirato direttamente on-site. Non solo, perché l'adozione delle nuove tecniche di Predictive Maintenance favorisce la rilevazione preventiva di queste anomalie, ad esempio tramite identificazione automatica del degrado di performance relative alle smart product in uso.

Lo scenario degli Smart Product apre quindi la strada a nuove opportunità e modelli di business, che devono essere sempre più allineati ai bisogni dei vendor e soprattutto dei loro customer. Ad esempio, si può rendere più intelligente l'assistenza post-vendita per abbassare i costi di gestione, considerare nuove forme di contratti come il *pay-per-use* o la *servitizzazione*. Un esempio sono gli smart product per la produzione di consumabili, come le capsule da caffè: in questo scenario i vendor possono proporre servizi smart nelle *supply chain*, per indicare automaticamente ai consumers quando effettuare il *refill* del loro stock. Questo può avvenire non solo sulla base delle capsule effettivamente consumate, ma anche tramite l'analisi aggregata e per-product dei vari pattern di consumo, insieme ad informazioni aggiuntive sul contesto: ad es. le previsioni meteorologiche indicano un vicino abbassamento delle temperature, e questo solitamente comporta un incremento del consumo delle capsule di caffè.

All'interno del progetto di ricerca SBDIOI40, l'O5.1 ha l'ambizione di realizzare strumenti automatici di "business consulting" tramite l'implementazione di nuovi servizi B2B di tipo "out" – ovverosia servizi pensati per un contesto applicativo esterno rispetto a quello della fabbrica intelligente. I servizi "out", tramite le funzioni di analytics di SBDIOI40, possono supportare gli smart product - macchine automatiche di nuova generazione che operano al di fuori della fabbrica. I servizi consentiranno di monitorare e controllare smart product da remoto abilitando l'e-Maintenance e la gestione business-oriented in tempo reale. La diagnosi remota dello smart product facilita l'intervento dei tecnici e l'e-Maintenance permette di ridurre notevolmente i tempi di inattività delle macchine dovuti ad avarie, con significativi vantaggi economici. I servizi gestionali business-oriented permettono di effettuare ragionamenti incrociando dati ottenuti dalle macchine, dal sistema informativo aziendale (CRM, ERP, ecc.) e da fonti esterne (open data/social/crowdsourcing). Il fine è dimostrare fattibilità e potenzialità di un servizio automatico di "business consulting" per individuare modalità operative più convenienti e/o nuovi scenari di mercato interessanti

1.1 Da Smart Factory a Smart Product

Solitamente i concetti tipici di Industry 4.0 si riferiscono agli scenari di una smart factory, dove le macchine sono considerate una parte delle linee di produzione all'interno dello shop floor. Molto spesso gli use case di Industry 4.0 coinvolgono grosse macchine industriali, smart, Internet-enabled e costose - che vengono distribuite e gestite direttamente on-site dal personale tecnico di riferimento. Tuttavia, questi approcci possono essere facilmente trasportati verso gli scenari di smart product, dove i macchinari sono tipicamente più contenuti ed economici; essi vengono distribuiti al di fuori della smart factory, spesso in località isolate e senza la presenza di personale tecnico per la loro gestione. Questi scenari sono sempre più abilitati dall'evoluzione dell'IoT, tramite la mercificazione dei sensori, delle unità di elaborazione e dei potenti stack software/hardware a disposizione degli sviluppatori. L'IoT rende quindi possibile il controllo degli oggetti fisici, solitamente gestiti da personale tecnico, come entità completamente digitali connesse ad Internet.

Lo scenario delle macchine Carpigiani vede smart product di piccole-medie dimensioni (se comparate alle linee di produzione in-factory) distribuite ed eventualmente riposizionate tra le varie gelaterie dei loro customers in tutto il mondo. Seppur di piccole-medie dimensioni, queste macchine eseguono processi tutt'altro che banali: gli ingredienti devono essere mescolati insieme, pastorizzati, omogeneizzati, raffreddati e pompati all'interno di congelatori a superficie raschiata, dove il 50% circa di acqua è congelata e l'aria è incorporata nel prodotto. Dopo la fase di raffreddamento possono essere eventualmente inseriti anche i variegati ed altri prodotti "di contorno", per poi riempire i contenitori a valle con il prodotto finale. I componenti meccanici ed elettronici sono concepiti

e sviluppati per il particolare scopo della macchina, e richiedono l'intervento di un tecnico specializzato durante le procedure di maintenance.

Queste tipologie di smart product possono essere in generale estremamente complesse, con sistemi high-duty dove il ciclo di vita è misurato nell'arco di diversi anni (di solito > 10 anni); ciò richiede interventi di manutenzione on-site, molto costosi sia dal punto di vista del costo del personale, che dal punto di vista delle risorse economiche perdute durante i downtime dei macchinari. In netto contrasto agli scenari di smart factory, si può caratterizzare l'ecosistema degli smart product con tre aspetti fondamentali:

- scala globale: un grande numero di macchine distribuite per il mondo, e la loro ricollocazione non è a carico del manutentore/vendor ma bensì del customer specifico che può facilmente spostarle sulla base delle sue esigenze.
- macchine incustodite: l'utilizzatore della macchina può usufruire di una serie limitata di tutte le funzioni disponibili, ad es. tramite Human Machine Interface (HMI). Inoltre, le limitate capacità tecniche dei proprietari richiedono interventi frequenti da parte dei tecnici della manutenzione, anche per semplici riconfigurazioni.
- deployment distribuito: solitamente ogni proprietario (es. gelataio) usa un numero limitato di macchinari, e questi non sono comunque strettamente connessi tra loro. Non è quindi necessaria una loro coordinazione rispetto a quello che invece avviene nelle linee di produzione/assemblaggio.

Lo scenario degli smart product offre ai vendor opportunità interessanti. Implementando infrastrutture di monitoraggio e controllo delle macchine remote, i vendor possono abilitare la manutenzione predittiva, in grado di rilevare automaticamente i degradi che potrebbero portare a malfunzionamenti in futuro, ad esempio prevedendo la vita utile residua di ingranaggi e cuscinetti rotanti o rilevando un deterioramento efficienza dei sistemi di refrigerazione. Ciò consentirebbe ai vendor di ottimizzare la pianificazione degli interventi di assistenza in loco secondo obiettivi multipli. Ad esempio, l'utilizzo dei componenti della macchina fino al punto "limite" di rottura consente di ridurre al minimo i costi di riparazione e i tempi di fermo macchina per preservare il flusso di entrate dei clienti. Considerando il caso specifico delle macchine per gelato Carpigiani, Teorema consente di: rilevare anomalie nella diminuzione di efficienza per i sistemi di refrigerazione, generalmente dovute a perdite di gas refrigerante o al degrado della pasta termica; monitorare i degradi del sistema di battitura, generalmente dovuti a sollecitazioni meccaniche; e programmare gli interventi in modo conveniente, massimizzando al contempo l'utilizzo di tecnici esperti e riducendo al minimo i tempi di fermo macchina. Di conseguenza, i venditori possono ridurre considerevolmente i loro costi di assistenza post-vendita e persino adottare nuove strategie commerciali, inclusi prezzi bassi per i contratti di assistenza post-vendita e modelli di business pay-per-use (verso la servitizzazione). Questa evoluzione consente ai proprietari di spostarsi verso scenari di value chain più complessi, innovativi e stimolanti. L'obiettivo principale è considerare i prodotti per tutta la loro vita, a partire

prima della produzione, ad esempio considerando i processi di progettazione / ingegneria e provisioning dei componenti, fino ai servizi post-vendita, considerando i servizi a valore aggiunto, come la manutenzione elettronica e persino la disattivazione.

1.2 RAMI 4.0

Lo scenario di fabbrica intelligente Industry 4.0 è attualmente sotto grande considerazione da parte delle grandi aziende e dei governi centrali / regionali spinti dall'aspettativa di un rivoluzionario cambiamento di prospettiva nel settore manifatturiero, dalla fornitura di nuovi servizi, dalla fornitura di nuove opportunità commerciali e dalla creazione di nuovi posti di lavoro. Prevediamo una duplice evoluzione. Da un lato, la tendenza accelerata verso l'automazione basata sulle TIC potrebbe limitare l'impiego del personale operativo nelle linee di produzione. D'altra parte, spingerà per il reclutamento di nuovi professionisti con competenze interdisciplinari non solo nei sistemi integrati ed elettronici, ma anche nelle reti di computer e nelle tecnologie Web.

A parte l'impatto sulla fabbrica intelligente, sosteniamo che l'IoT penetrerà in modo significativo anche nello scenario degli elettrodomestici intelligenti, sebbene non vi sia ancora un ampio consenso sul fatto che le soluzioni IoT possano raggiungere tutti gli obiettivi previsti. Notiamo che la mancanza di un'architettura standard e di un vocabolario per descrivere chiaramente le soluzioni di elettrodomestici intelligenti proposte e già sviluppate limita notevolmente la capacità di fecondazione interaziendale, già limitata dalla privacy e dalla concorrenza delle imprese. In effetti, descrivendo ogni prodotto e servizio con una terminologia diversa, sia in termini di architettura generale che di singoli componenti, le imprese pionieristiche non riescono a presentare in modo appropriato i risultati raggiunti, generando una grande eterogeneità e approfondendo la frammentazione e lo scompiglio nello sforzo di comunicazione.

Per spingere verso un approccio coordinato e omogeneo verso il paradigma della fabbrica intelligente, i gruppi di standardizzazione e i governi stanno proponendo nuove architetture e terminologie unificanti che considerano le questioni commerciali e tecnologiche. L'obiettivo principale è fornire una piattaforma comune per identificare meglio i principali aspetti critici e confrontare facilmente diverse soluzioni. In questo contesto, RAMI 4.0 è recentemente emerso come il modello più promettente. Rispetto ad altri standard emergenti che si concentrano principalmente sul modo in cui vengono gestiti gli elettrodomestici intelligenti (e i relativi dati), come l'Industrial Internet Reference Architecture (IIRA), RAMI 4.0 è più adatto a modellare il più ampio scenario della catena del valore intelligente, gestendo adeguatamente anche l'intero ciclo di vita, lo sviluppo e la manutenzione di dispositivi intelligenti. Infatti, RAMI 4.0 è stato concepito in modo nativo per affrontare scenari di Industry 4.0 complessi: considera la gestione dei dispositivi remoti solo come una delle sue sotto indicazioni,

fornendo considerazioni aggiuntive e pertinenti ed estendendolo lungo il cosiddetto ciclo di vita e flusso di valore e le direzioni della gerarchia.

Il modello finale è un'architettura di riferimento tridimensionale composta da:

- **Livelli**: questa dimensione verticale fornisce una segmentazione tradizionale dell'architettura IT, adottando un design a strati per facilitare lo sviluppo di nuove soluzioni. Questa dimensione parte dalle risorse (oggetti fisici) e dai suoi livelli di integrazione per digitalizzarle mentre il livello di comunicazione superiore fornisce l'accesso remoto. Quindi, il livello Informazioni raccoglie e gestisce i dati per alimentare il livello Funzionale offrendo funzionalità di alto livello per monitorare / controllare gli oggetti fisici sottostanti, e infine il livello aziendale di livello superiore presenta i processi relativi a tutte le organizzazioni;

- **Livelli di gerarchia**: questa dimensione orizzontale fornisce descrizioni funzionali per componente che descrivono il modo in cui il ciclo di vita del prodotto (intelligente) può interagire a livello incrociato con qualsiasi altro componente tipico della fabbrica. Ciò spazia dai dispositivi di campo e di controllo alle stazioni, ai centri di lavoro e all'azienda nel suo complesso. L'obiettivo finale è massimizzare la flessibilità del sistema estendendo l'ambiente di fabbrica al mondo esterno;

- **Ciclo di vita e flusso di valore**: questa dimensione orizzontale si concentra su interi prodotti, a partire dalla loro progettazione e sviluppo fino alla loro produzione, distribuzione e manutenzione effettive. Identifica chiaramente e differenzia la fase Tipo, per sviluppare e mantenere un nuovo prodotto, e la fase Istanza, per produrre e mantenere singole istanze del prodotto stesso una volta che è stato completamente progettato e sviluppato, comprendendo quindi anche i processi di e-Maintenance. Questa prospettiva è molto significativa per l'intera produzione di un piccolo lotto di prodotti. Vale la pena notare che il ciclo di vita e il flusso di valore di diversi attori possono interagire e interagire tra loro. Ad esempio, l'output della fase Instance di un fornitore che vende componenti di macchine può rappresentare l'input della fase Tipo di un'impresa che sta attualmente progettando e sviluppando una nuova macchina.

Questo approccio consente di caratterizzare meglio la catena del valore, rendendo più chiare le numerose e diverse interazioni tra le fasi successive della vita di un prodotto all'interno di un'impresa, migliorandone così la gestione. Inoltre, sottolinea le dipendenze da imprese esterne, ad esempio evidenziando aspetti potenzialmente critici come una forte dipendenza da un determinato prodotto di un venditore. In conclusione, RAMI 4.0 include le definizioni di base e gli elementi costitutivi che consentono di modellare lo scenario della catena del valore intelligente, basato sul ciclo di vita e sul flusso di valori. Consente alle aziende di identificare e delineare meglio i molti aspetti coinvolgendo la progettazione, lo sviluppo e l'implementazione di smart apparecchi, facilitando la

loro descrizione e piena comprensione tra il personale (interno ed esterno) incaricato di interagire con essi. Allo stesso tempo, notiamo che la maggior parte dei casi d'uso di RAMI 4.0 si concentra sulla fabbrica intelligente e in generale sulla manutenzione di prodotti intelligenti di grandi dimensioni e costosi. Riteniamo invece che sia necessario applicare questi standard anche negli scenari di smart product e gestirli lungo l'intero ciclo di vita, come proponiamo nel seguenti sezioni.

1.3 La soluzione Carpigiani

Carpigiani è una delle industrie leader mondiali nel mercato delle macchine per gelateria: vende in 110 paesi in tutto il mondo, essendo le sue macchine installate in gelaterie e ristoranti. Le macchine funzionano senza alcun supporto tecnico in loco e, nonostante l'elevata qualità costruttiva, si imbattono in costi di manutenzione non trascurabili durante la loro lunga vita prevista (da 10 a 20 anni), a causa delle elevate operazioni a cui sono sottoposti. Carpigiani ha abbracciato pervasivamente la rivoluzione dell'Industria 4.0, aderendo alla filosofia manifatturiera discussa nella prima parte di questo documento. La soluzione Teorema integra le macchine per il gelato in una catena del valore intelligente che coinvolge molti stakeholder diversi, che vanno dai proprietari e dagli operai delle gelaterie ai clienti finali, posizionando Carpigiani all'avanguardia nella tendenza degli elettrodomestici intelligenti.

Teorema include cinque servizi intelligenti principali descritti di seguito. Il servizio di sicurezza rappresenta un aiuto fondamentale per i proprietari di gelaterie. Monitorando e registrando le temperature dei gelati prodotti da ciascuna macchina e fornendo report dettagliati e (soft) in tempo reale, Teorema consente di migliorare e certificare la sicurezza dei consumatori di gelati e di individuare tempestivamente le problematiche legate all'approvvigionamento alimentare. In questo modo, i proprietari di macchine, vale a dire i produttori di gelati, possono certificare che i processi di produzione del gelato non "rovinano la catena del freddo" e seguono invece le rigorose norme di sicurezza alimentare HACCP (Hazard Analysis and Critical Control Points). Il servizio di ottimizzazione dell'efficienza monitora automaticamente i modelli di utilizzo delle macchine per gelateria e suggerisce processi più efficienti. Ad esempio, a periodi di tempo prestabiliti è necessario svuotare completamente la macchina per eseguire un ciclo di sanificazione, eliminando gli ingredienti rimasti per il gelato. Quando l'operatore deve ricaricare la macchina, Teorema può segnalare l'esatta quantità di ingredienti necessari per servire il gelato fino al successivo ciclo di sanificazione della macchina, riducendo così al minimo lo spreco di cibo. Si noti che la macchina consiglia l'operatore sulla quantità di ingredienti da aggiungere anche in base alle recenti tendenze del consumo di gelati e alle previsioni del tempo (ad esempio, in caso di consumo di gelati di pioggia in genere diminuisce). Il servizio di ottimizzazione energetica analizza tutti i dettagli delle esigenze di produzione di più macchine per gelato nello stesso negozio e può consigliare come risparmiare energia.

Raccomanderà sempre l'uso più efficiente delle macchine, in base alle reali esigenze aziendali. Ad esempio, è possibile limitare il picco del consumo energetico pianificando automaticamente le attività di lavorazione del gelato (cicli di pastorizzazione) quando altre apparecchiature nella stessa gelateria sono in standby. Il sistema indica se una singola macchina è pronta per la produzione o deve essere passata alla conservazione in base allo stato di altre macchine e tiene traccia del consumo giornaliero di energia. Esperienze con l'adozione anticipata del servizio di ottimizzazione energetica, nonché simulazioni estese e accurate delle operazioni della macchina per il gelato in una vasta gamma di scenari realistici, indicano che potrebbe essere in grado di ridurre il picco di consumo energetico di una tipica gelateria fino a 20 %.

Il servizio di manutenzione intelligente consente ai tecnici Carpigiani (produttore) di attuare strategie di manutenzione predittiva attraverso una profonda conoscenza delle condizioni operative delle macchine sul campo. Teorema consente di monitorare a distanza lo stato sinottico delle macchine per gelato e consente la visualizzazione e l'analisi in tempo reale dei dati raccolti tramite un'interfaccia Web. Il servizio di e-Maintenance Carpigiani mira a rilevare componenti difettosi sfruttando i dati forniti dal prodotto collegato per generare automaticamente rapporti di allarme che descrivono condizioni anomale o il completamento con successo di processi come la pastorizzazione e i registri operativi ("eventi" nella terminologia di Teorema) per monitorare lo stato attuale delle macchine. Le funzioni di manutenzione remota consentono di ridurre significativamente i costi di assistenza post vendita e i tempi di riparazione. In effetti, riconfigurare da remoto i parametri di lavoro (o persino aggiornare il firmware) è spesso un modo molto efficace per risolvere i problemi esposti dalle macchine per il gelato o per prolungarne la durata. Ad esempio, riducendo la velocità di rotazione del battitore o aumentando i tempi per il processo di pastorizzazione consente una pianificazione intelligente degli interventi di assistenza in loco che ne minimizza i costi. Ciò non solo comporta costi operativi inferiori per i clienti (proprietari di macchine), ma consente anche di esplorare nuovi modelli di business orientati ai servizi, più allineati alle esigenze dei clienti. Inoltre, il magazzino integrato Carpigiani consente una rapida verifica della disponibilità del componente di ricambio richiesto ed eventualmente un rapido riordino tramite l'integrazione con ERP fornitore per garantire il rapido recupero e consegna del pezzo di ricambio. A questo punto il tecnico specializzato in gelateria può programmare l'intervento in loco per riparare la macchina. Il servizio di supporto alle imprese include una dashboard aziendale che consente ai clienti di controllare lo stato delle loro macchine e controllare la redditività delle loro gelaterie o ristoranti. Per i clienti che gestiscono diversi negozi, Teorema consente confronti location-aware tra le metriche relative alle prestazioni aziendali correlate, per valutare se in un negozio specifico qualsiasi pratica di cattiva gestione della macchina sta riducendo la redditività, ad esempio ritardi nella pulizia della macchina causano tempi di inattività inutili. Inoltre, la maggiore efficienza nei servizi di assistenza post-vendita introdotta da Teorema ha permesso di iniziare a offrire contratti pay-per-use, in cui i clienti noleggiano macchine da gelato (controllate a distanza) invece di acquistarle. Infine, notiamo anche che Teorema ha permesso a Carpigiani di passare a una pratica di fabbricazione di personalizzazione di massa.

Mentre Carpigiani non è mai stata una società di produzione di massa, con l'adozione di Teorema, l'azienda ha iniziato a spedire macchine per gelato solo con un software di base e ad installare software specifico per il cliente nel momento in cui le macchine sono installate nei locali del cliente. L'adozione delle politiche di personalizzazione di massa abilitate a Teorema ha consentito una significativa riduzione dei tempi di consegna e dei costi di personalizzazione, aprendo la strada a nuovi servizi come la messa a punto a grana fine e su richiesta dei processi di produzione del gelato implementati dalle macchine, facilitando i clienti nello sviluppo le loro ricette di gelato.

Concludiamo questa sezione tracciando alcune prime lezioni apprese sulla base dell'esperienza dei Carpigiani. Innanzitutto, l'adozione dei principi dell'Industria 4.0 in macchine relativamente economiche e di piccole dimensioni può migliorare scenari intelligenti arricchendo nuovi servizi con una pletora di informazioni tradizionalmente disponibili solo per il reparto specifico che sono state generate in modo basato su silos. In effetti, l'integrazione orizzontale e verticale, come suggerito da RAMI 4.0, sono estremamente importanti. Adozione dell'integrazione orizzontale inter-area, L'industria 4.0 può sfruttare appieno tutte le informazioni prodotte dagli elettrodomestici intelligenti, creando un nuovo valore spingendo verso la fecondazione incrociata di diversi dipartimenti. Tramite l'integrazione verticale, invece, Industry 4.0 può alimentare diversi livelli organizzativi con informazioni generate da altri livelli, consentendo di migliorare il processo decisionale dei dirigenti di livello superiore in base alle informazioni aggiornate sulla produzione e agli ordini in entrata. Ad esempio, se una linea di produzione in un'officina è attualmente poco performante rispetto al suo obiettivo, i dirigenti operativi possono essere prontamente informati per consentire di indagare su possibili problemi. Di conseguenza, è possibile rilevare (e adottare le opportune contromisure) nel caso in cui la produttività sia inferiore alle aspettative non appena si verifichi un problema (quindi non in attesa di rapporti settimanali forniti dai supervisori della linea di produzione) o riprogrammare più facilmente la produzione della macchina in caso è necessario per soddisfare un ordine imprevedibile di alta priorità.

1.4 Teorema: Architettura e Implementazione

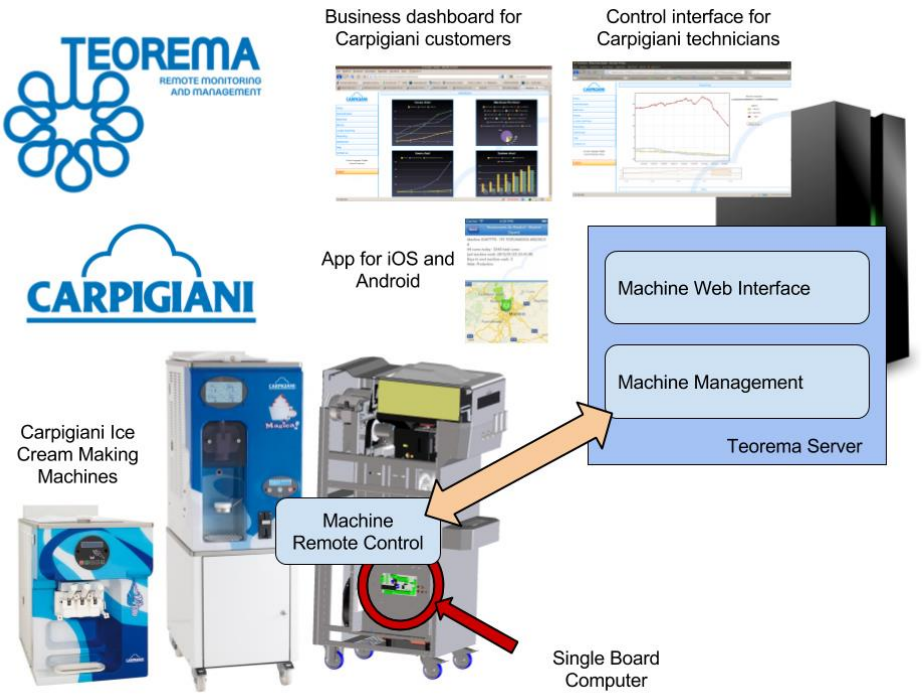

Fig. 1. Smart Product management solution Teorema di Carpigiani.

Teorema è la piattaforma completa e integrata (hardware e software) nel cuore della soluzione smart product Carpigiani per macchine per gelateria. Come illustrato in Fig. 1, Teorema ha un'architettura distribuita composta da diversi componenti software, in esecuzione su macchine per il gelato o in ambiente cloud ibrido Carpigiani (con componenti in esecuzione su un cloud privato o su cloud Amazon AWS, a seconda delle preferenze del cliente). Teorema lo è basato su tre componenti principali: Machine Remote Control (MRC) che si comporta come client e funziona su un computer a scheda singola installato su macchine per gelato, e due componenti sul lato server, ovvero Machine Management (MM) e Machine Web Interface (MWI) fungendo rispettivamente da punto di accesso per l'invio di informazioni e come interfaccia grafica multi-ruolo.

Ulteriori informazioni sulle funzioni di Teorema possono essere recuperate dal sito Web dedicato ai servizi Carpigiani 2, con la brochure aziendale e un video 3 che descrive il sistema intelligente offerto. Inoltre, su Google Play 4 e Apple App Store 5 è disponibile un'app mobile dedicata con le credenziali per un account dimostrativo predefinito disponibile insieme a un video

1.4.1 Implementazione di Teorema

Il software installato sulla macchina, come nella figura 2, è costituito da due componenti guidati da eventi, simultanei e in continuo coordinamento in esecuzione su un computer a scheda singola (SBC) basato su ARM. L'applicazione Supervisor si occupa delle comunicazioni ad alta frequenza e di basso livello con i principali componenti hardware della macchina, consentendo a Unified Teorema Interface (UTI) di concentrarsi su funzioni di livello superiore con un approccio agnostico hardware. Il supervisore è inoltre responsabile della gestione degli aggiornamenti del software, dell'applicazione dei protocolli di sicurezza operativa in ogni momento e della gestione della configurazione di parametri di funzionamento della macchina. UTI invece fornisce un HMI all'avanguardia che integra 3 moduli importanti: gestione dei processi, monitoraggio energetico e controllo remoto. La gestione dei processi esegue un controllo completo del processo di gestione dei gelati, implementando l'esecuzione di ricette personalizzate, applicando i protocolli di sicurezza alimentare e monitorando e ottimizzando l'efficienza del processo. Il monitoraggio energetico misura il consumo di energia, crea un modello di utilizzo della macchina e suggerisce possibili ottimizzazioni. Controllo remoto, fornisce funzioni di monitoraggio e controllo remoto e può opzionalmente funzionare come componente autonomo. Ciò rende Teorema facile da installare su macchine più vecchie, eseguendo il modulo di controllo remoto sopra un kit di monitoraggio remoto collegato all'interfaccia di diagnostica RS-485 della macchina. Concentrandosi sul lato server, dettagliamo le principali interazioni e componenti interni di MM e MWI in Fig. 3. MM riceve i dati di manutenzione dalle macchine per la produzione di gelati e li memorizza in un DBMS dedicato; è anche responsabile del coordinamento degli aggiornamenti software. Sfruttando i dati raccolti, MWI è costituito da sei componenti che forniscono tutte le principali funzioni della piattaforma Teorema, implementate su tre componenti principali: Business Logic, Role-Based Authentication Control (RBAC) e Integrazione CRM / ERP. Configuration Management consente di attivare (o pianificare) gli aggiornamenti software e la riconfigurazione dei parametri di lavoro per una singola macchina o un gruppo di macchine, attraverso un'interfaccia facile da usare. La visualizzazione fornisce funzioni di visualizzazione comparativa per i dati raccolti dalle macchine, consentendo il facile rilevamento di anomalie. Business Dashboard fornisce un'interfaccia alla piattaforma Teorema specificamente dedicata ai clienti Carpigiani, che consente loro di monitorare le entrate e l'efficienza della loro macchina per la produzione di gelati e di intervenire quando necessario. Il reporting fornisce funzioni di gestione a grana fine multi-ruolo su PDF o e-mail. L'API ReST esporta tutte le principali funzioni della piattaforma attraverso un'interfaccia programmabile, che consente di accedervi da remoto, ad esempio tramite app mobili.

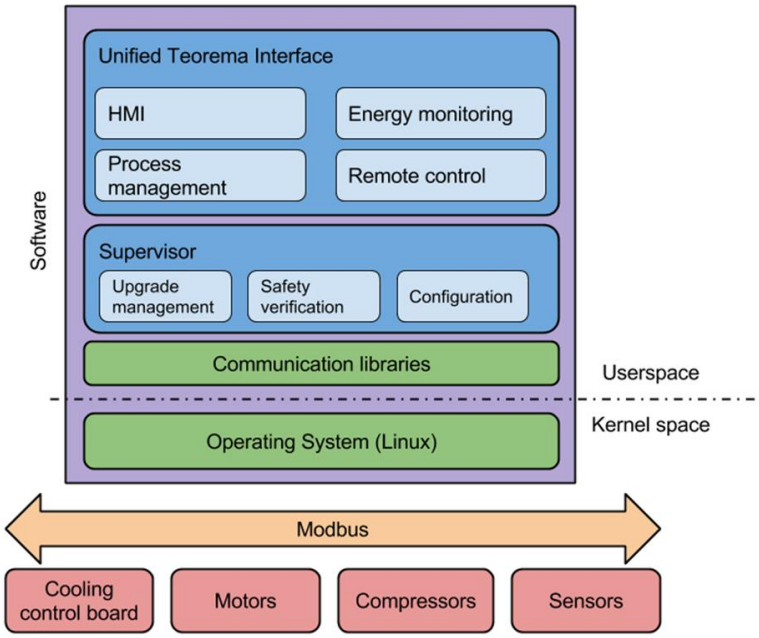

Fig. 2. Architettura Software delle macchine Carpigiani

Infine, MWI include Analytics che è probabilmente il componente più importante dell'intera piattaforma Teorema. Di fatto, ha il compito di elaborare i dati raccolti dalle macchine per la produzione di gelati, di aggregarli per acquisire conoscenze sui processi delle macchine (ad esempio il trattamento di pastorizzazione) e di costruire modelli comportamentali per ogni macchina. Inoltre, Analytics offre funzioni di rilevamento e previsione automatiche degli errori, implementate su sofisticate soluzioni Big Data che eseguono algoritmi di diagnostica e rilevamento delle anomalie in un cloud privato. Più specificamente, Analytics implementa il monitoraggio in tempo reale (soft) basato sulla soglia delle condizioni operative delle macchine utilizzando i dati grezzi raccolti dalle macchine, insieme all'analisi delle serie temporali, ad esempio un vicino più vicino (1-NN) e Dynamic Time Warping (DTW).

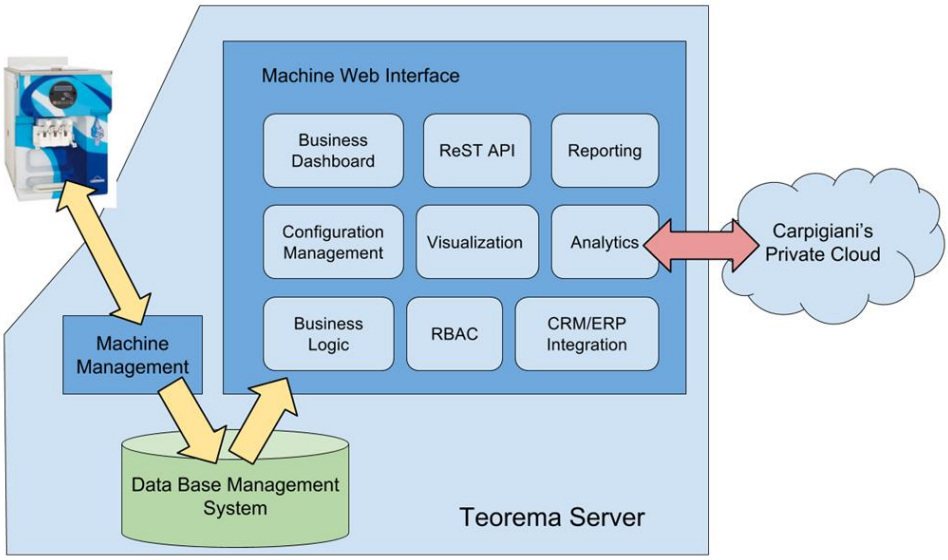

Fig. 3. Architettura server-side di Teorema

2. Requisiti e obiettivi

La piattaforma Teorema è sicuramente il fiore all'occhiello dell'infrastruttura IT di Carpigiani. Nata dodici anni fa, è caratterizzata da un'architettura innovativa per quel periodo, ma nel corso degli anni ha mostrato diversi limiti. Il sistema non è pensato per integrare uno strumento di esplorazione dei dati orientato al personale non tecnico. Inoltre le esigenze aziendali stanno aumentando notevolmente a causa dell'evoluzione del mercato circostante, legato principalmente al fenomeno in continua crescita dei Big Data. La visualizzazione dei dati è a bassa densità e può risultare controintuitiva. È infatti uno strumento non pensato per la facile creazione e customizzazione di dashboard, che invece sarebbero particolarmente utili per confrontare più informazioni contemporaneamente e individuare comportamenti anomali. Su Teorema è possibile visualizzare solamente lo stato di una macchina alla volta non avendo di conseguenza una visione d'insieme. Questo costituisce un limite troppo significativo in quanto restringe il campo visivo e a livello pratico, rallenta le operazioni di monitoraggio. Inoltre proprio a causa della forte eterogeneità delle macchine Carpigiani, il modello dei dati non è standardizzato e ciò lo rende troppo "rigido" per poter effettuare delle analisi ad ampio spettro.

2.1 Obiettivi

È diventato indispensabile poter intrecciare sorgenti dati differenti, interne ed esterne, con lo scopo di inferire informazioni nuove, legate alle preferenze dei consumatori. Un esempio potrebbe essere studiare il numero di coni prodotti o l'uptime delle macchine, in parallelo con i dati meteorologici delle varie regioni. In generale,

l'obiettivo principale è quello di avere una visualizzazione dei dati trasversale ad alta densità e intuitiva, in modo tale da possedere una visione globale delle macchine, anche in confronto tra loro. Per la realizzazione di nuovi servizi gestionali business-oriented possiamo ragionare a step. Il primo passo fondamentale diventa il passaggio da un Data Lake contenente i dati "grezzi" (CRM e MySql), a un Data Warehouse con dati "spacchettati". Successivamente, il secondo obiettivo diventa quello di configurare e utilizzare un applicativo "già pronto" per costruire dashboard customizzate per le esigenze aziendali. Attualmente i report vengono realizzati ad hoc unicamente dal team che si occupa di Teorema e vengono forniti agli altri enti aziendali in base alle specifiche esigenze.

L'analisi dei dati viene sviluppata dai programmatori ogni volta che nasce una necessità. Questo meccanismo va quindi superato cercando di costruire un servizio automatico di Business Consulting, che porta il dato a essere modellato automaticamente e successivamente visualizzato in aggregazioni dinamiche e intelligenti. Le dashboard sono perfette per questo scopo. Sono intuitive, colorate, interattive, permettono (o dovrebbero permettere) di poter essere lette in piena autonomia da ogni rappresentante aziendale, rendendo indipendente il business nel reperimento delle informazioni desiderate. Gli scopi "secondari" sono facilitare la consultazione dello storico dei dati, in modo da ottenere un consolidamento dell'architettura di analisi e una retroazione su tutto il processo di raccolta dei dati stessi.

2.2. Descrizione del progetto

La prima fase consiste nel trovare l'architettura più adatta per raccogliere, in maniera intelligente, la grossa mole di dati raccolti negli ultimi dodici anni grazie a Teorema. La modernizzazione del sistema di raccolta necessita lo studio approfondito del dominio Big Data e dello stato dell'arte delle aziende nel settore dell'Industria 4.0, una ricerca ad ampio spettro delle tecnologie presenti sul mercato e infine un'analisi dettagliata delle best practices per il deployment del sistema. Nella seconda fase si dovrà scegliere l'applicativo più adatto nella costruzione di un sistema B2B di Business Consulting.

I requisiti funzionali che si devono ricercare per tale applicativo sono i seguenti:

- Data Management Tools
- Data Discovery Applications
- Reporting Tools

Il primo punto si può suddividere in Data quality management ed Extract, Transform and Load (ETL). Insieme di funzionalità che aiutano l'azienda a mantenere i dati puliti, standardizzati e privi di errori. La gestione della

qualità dei dati garantisce che le analisi successive siano corrette e possano portare a miglioramenti all'interno dell'azienda. La standardizzazione è particolarmente importante perché fornisce la possibilità di raccogliere dati da fonti eterogenee, trasformarli e caricarli nel sistema di destinazione. Poiché i dati primari sono spesso organizzati utilizzando schemi o formati diversi, gli analisti possono utilizzare gli strumenti di ETL per normalizzarli al fine di operare un'analisi utile.

Il secondo aspetto comprende tutte le pratiche che permettono di effettuare operazioni avanzate sul dato, dette di drill-down e slice and dice. Si parla delle applicazioni di Data Mining, Machine Learning e Predictive Analytics, che servono a ordinare grandi quantità di dati per identificare modelli nuovi e pattern sconosciuti oltre ad analizzare i dati attuali e quelli storici per fare previsioni sui rischi e sulle opportunità future. Come ultima, ma non per questo meno importante, la capacità di presentare i dati e trasmettere facilmente i risultati dell'analisi. In risposta, i fornitori di software hanno lavorato per mascherare la complessità di queste applicazioni e concentrarsi sempre di più sull'esperienza utente.

Gli strumenti più importanti in questo senso sono le visualizzazioni e le dashboard. Le prime consistono in un'unica rappresentazione grafica, che può essere di varie tipologie (grafico a torta, istogramma, grafico a linee, ecc.). La forza di una visualizzazione è quella di focalizzare un particolare dato, mostrandone il comportamento. Le dashboard invece sono un insieme di visualizzazioni e lo scopo è di evidenziare gli indicatori chiave di prestazione (KPI), dando una visione d'insieme.

I requisiti non funzionali che devono contraddistinguere un applicativo di Business Consulting sono quindi la capacità di orientarsi verso un'interfaccia sempre di più user-friendly e sicuramente quello di possedere al suo interno un sistema di autenticazione e autorizzazione per mantenere i dati al sicuro. Altro aspetto fondamentale è la crescita esponenziale dei dati che avverrà durante il tempo, perciò diventa molto importante implementare una soluzione che sia facilmente scalabile verticalmente, ma soprattutto orizzontalmente.

3. Data Warehouse

I dati sono cambiati. Volumi più grandi, necessità di un'elaborazione più rapida e nuovi tipi di dati comportano nuovi scenari ad analizzare a fondo. Nel contesto dei Big Data, molte organizzazioni hanno iniziato a ricercare nuove soluzioni come tecnologie NoSQL, per aumentare le capacità dei database relazionali tradizionali e per risolvere i nuovi problemi legati alle performance e alla scalabilità.

Si trovano tipicamente numerosi sistemi aziendali che elaborano e producono dati: strumenti che alimentano il sito Web rivolto al cliente, di monitoraggio dell'inventario nei magazzini, ERP, CRM e simili. In Carpigiani Teorema stesso fa parte di questa categoria, detta dei sistemi OLTP. Solitamente ciascuno di questi ha il suo database, ma ciò non permette di effettuare analisi su larga scala. Un Data Warehouse, al contrario, è un database di grandi dimensioni indipendente. I dati devono essere estratti utilizzando un flusso continuo di aggiornamenti o eseguendo un dump dei dati periodico, poi modellati e quindi caricati nel Data Warehouse. Questo processo di acquisizione dei dati fa parte di quella serie di operazioni generalmente denominate come Extract-Transform-Load (ETL) ed è illustrato nella Figura 8.

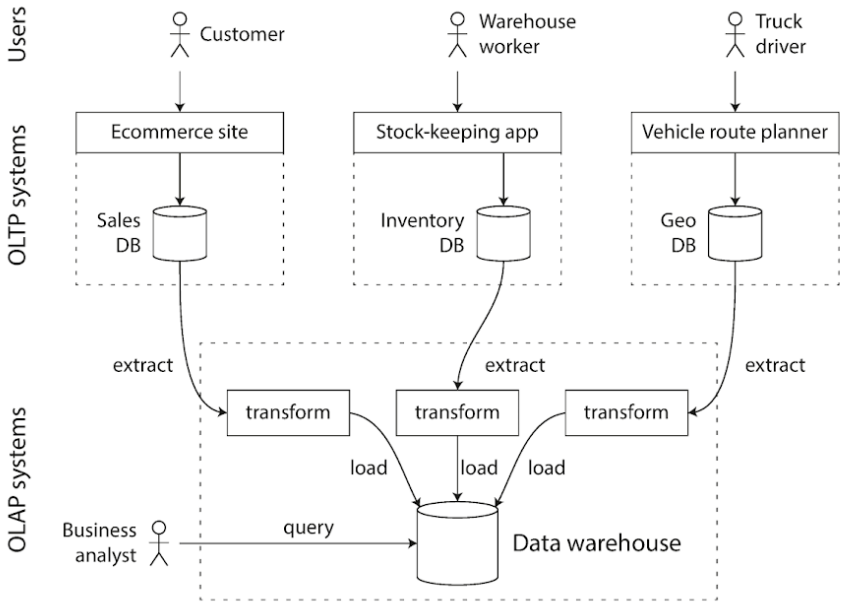

Fig. 8. Operazioni di Extract-Transform-Load (ETL)

3.1 Database relazionale vs. NoSQL

È importante notare che NoSQL non si riferisce a una precisa tecnologia/strumento. È un termine generico per indicare diverse tecnologie, ognuna delle quali affronta un problema specifico. Le principali tecnologie NoSQL impiegate in ambito Big Data sono le seguenti:

- Database orientati ai documenti
- Database orientati alle colonne
- Database chiave-valore
- Database orientato ai grafi

Nel contesto di questo studio ci si concentra sulla prima categoria, essendo una delle più performanti. Di seguito una breve descrizione degli altri.

Database orientati alle colonne: quando si arriva ad avere trilioni di righe e petabyte di dati nelle “tabelle dei fatti”, archiviarli e interrogarli in modo efficiente diventa un problema difficile. Una tipica query sul Data Warehouse accede contemporaneamente solo a quattro o cinque delle colonne presenti, pertanto le query "SELECT *" sono raramente necessarie per l'analisi. L'idea alla base dell'archiviazione orientata alle colonne è semplice: non archiviare tutti i valori della riga insieme, ma memorizzare tutti i valori di ciascuna colonna insieme. Se ogni colonna è memorizzata in un file separato, una query deve solo leggere e analizzare quelle colonne utilizzate in quella query, il che può risparmiare molto lavoro.

Database chiave/valore: un database chiave-valore immagazzina i dati come un insieme di coppie di chiave-valore, dove una chiave rappresenta un identificatore univoco. Sia chiavi che valori possono essere di diversi tipi, da un oggetto semplice ad articolati oggetti composti. I database chiave-valore sono altamente partizionabili e consentono un dimensionamento orizzontale a livelli che altri tipi di database non possono consentire. Ciò offre una notevole flessibilità e segue più da vicino concetti moderni come la programmazione orientata agli oggetti, il che può portare a grandi guadagni di prestazioni in determinati carichi di lavoro.

Database orientato ai grafi: se l'applicazione ha principalmente relazioni one-to-many, tipiche dei dati strutturati ad albero o nessuna relazione tra record, il modello orientato ai documenti è più appropriato. In caso contrario, il modello relazionale può gestire casi semplici di relazioni molti-a-molti, ma quando le connessioni all'interno dei dati diventano più complesse, diventa più naturale iniziare a modellare i dati come un grafico. Un grafico è costituito da due tipi di oggetti: vertici (noti anche come nodi o entità) e bordi (noti anche come relazioni o archi).

3.2 Document Database – JSON

È importante ricordare che i database orientati ai documenti in generale risolvono una serie di problemi legati ai database relazionali tradizionali:

- Schema rigoroso: con un database relazionale, se si dispone di dati fortemente eterogenei e semi-strutturati, si è costretti a creare un gruppo di colonne di dati "varie" casuali e inserire i dati come un blocco descrittivo e ciò impedisce di poter operare analisi su queste informazioni

- Scalabilità difficile: Con un database relazionale, se i tuoi dati sono così grandi da non poter essere inseriti facilmente in un singolo server, non esiste in general un meccanismo per ripartizionare questi dati su più nodi

- Migrazioni difficili: Con un database relazionale, cambiare la struttura del database può essere una grande sfida (specialmente quando i dati crescono).

JSON (JavaScript Object Notation) è il formato per lo scambio di dati più popolare sul web. Per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare ed analizzare la sua sintassi. Si basa su un sottoinsieme del Linguaggio di Programmazione JavaScript. Tuttavia, JSON è un formato di testo completamente indipendente dal linguaggio di programmazione. Questa caratteristica fa di JSON un linguaggio ideale per lo scambio di dati e soprattutto come codifica del dato all'interno di un database orientato ai documenti.

JSON è basato su due strutture:

- Un insieme di coppie nome/valore. In diversi linguaggi, questo è realizzato come un oggetto, un record, uno struct, un dizionario, una tabella hash, un elenco di chiavi o un array associativo.

- Un elenco ordinato di valori. Nella maggior parte dei linguaggi questo si realizza con un array.

Questo tipo di formato diventa necessario nel momento in cui il dato risulta molto eterogeneo. Troviamo questa situazione in Carpigiani, perché nei dati troviamo una parte delle informazioni estremamente variabile in base al tipo di macchina e modello.

3.3 Strumenti

Prima di tutto è stata fatta una comparazione fra i sistemi di storage più importanti compatibili con i dati in formato JSON, MongoDB e PostgreSql, per decidere quale fosse la soluzione più adatta a per lo storage dei dati "spacchettati".

3.3.1 PostgreSQL

PostgreSQL è una tecnologia open source flessibile, affidabile e performante e può essere una soluzione di data warehousing semplice, efficiente e a basso costo. Un database PostgreSQL rappresenta la perfetta combinazione tra le tecnologie dei database SQL (relazionali) e NoSQL (schema-less). Nasce come database object-relational in grado di offrire, proprio per la natura intrinseca di un linguaggio orientato agli oggetti, una forte estensibilità. Supporta classi, oggetti, metodi e tipi di dato custom. Grazie a questa caratteristica di fatto, è in grado di sfruttare al massimo le potenzialità dei due modelli, sia offrendo potenti funzionalità di query SQL (legate alle proprietà transazionali ACID), sia integrando al suo interno il modello dei dati JSON e JSONB. Quest'ultimo è salvato in un formato binario decomposto, che rende più lente le operazioni di input a causa del sovraccarico di conversione aggiunto, ma allo stesso tempo velocizza enormemente le elaborazioni e inoltre supporta le indicizzazioni, le quali favoriscono ulteriormente l'aumento del throughput. Pertanto, utilizzando un protocollo HTTP/JSON, molti strumenti possono utilizzare PostgreSQL per servizi di analytics e data mining, grazie alla forte compatibilità con le API REST (Figura 9).

PostgreSQL protegge i dati e coordina gli accessi concorrenti multiutente attraverso le transazioni. Il Transaction Model utilizzato da PostgreSQL si basa sul modello MVCC (Multi-Version Concurrency Control). Tale modello provvede a una migliore performance rispetto a quella riscontrabile in altre tecnologie che coordinano la multiutenza attraverso i sistemi di bloccaggio a livello di tabella, di pagina o di riga.

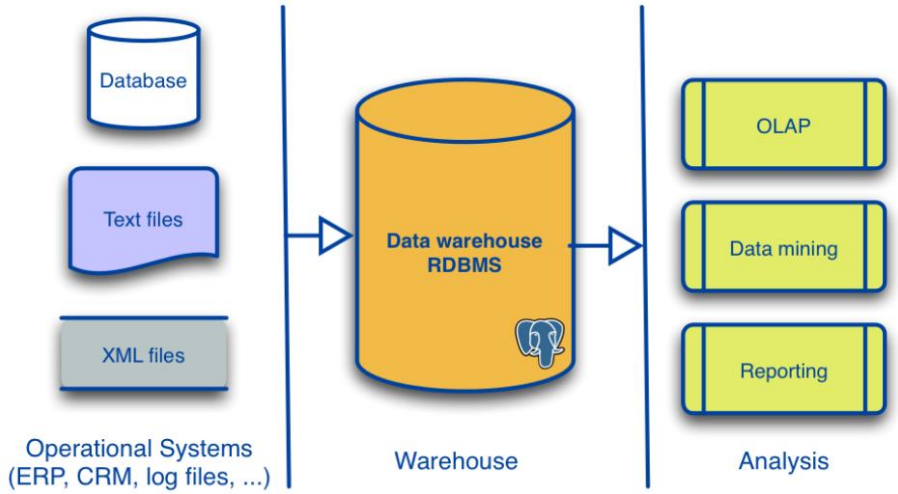

Fig. 9. Architettura Data Warehouse.

3.3.2 MongoDB

MongoDB è un sistema NoSQL open source ed è stata l'implementazione più visibile e accessibile nel periodo in cui gli sviluppatori si sono orientati verso questa tipologia di database. A differenza di PostgreSQL, MongoDB è completamente orientato ai documenti e ciò lo rende molto scalabile e affidabile. Infatti esso possiede sistemi built-in per lo sharding e per la replication. Come mostrato in Figura 10, lo sharding è l'approccio per scalare orizzontalmente, che permette di dividere i dati e memorizzare partizioni di essi su macchine differenti. Questo consente di immagazzinare più dati e gestire più carico di lavoro senza richiedere l'utilizzo di server più grandi e potenti. Il formato del documento è il BSON, molto simile al JSONB utilizzato da PostgreSQL. BSON estende il modello JSON per fornire tipi di dati addizionali,campi ordinati e per essere efficiente nella codifica e decodifica con diversi linguaggi di programmazione.

MongoDB nasce nel 2009, ed inizialmente ha ricevuto non poche critiche a causa di numerosi bug e performance sensibilmente inferiori rispetto ai sistemi relazionali. Le prime problematiche erano compensate dalla notevole semplicità che contraddistingue il linguaggio di query e dalla maggiore versatilità rispetto ad altre soluzioni NoSQL. Infatti la flessibilità del modello a documenti unito a un linguaggio di interrogazione semplice ma potente, permette a MongoDB di essere usato in un maggior numero di scenari applicativi, pur rimanendo nella fascia di mercato specializzata, tipica del panorama NoSQL.

Dalla versione 1.8 a quella attuale, MongoDB ha effettuato però notevoli passi avanti, migliorando le performance delle query, diminuendo i tempi di latenza ed aggiungendo funzionalità orientate al mondo Big Data. Ora è addirittura possibile effettuare transazioni multi-documento, effettuare trasformazioni all'interno

del database e infine costruire delle pipeline tipiche di un processo di ETL. Un'altra caratteristica di MongoDB è il supporto di indici secondari generici, permettendo una varietà di query veloci e fornendo capacità di indexing uniche, composite, geospaziali e full text.

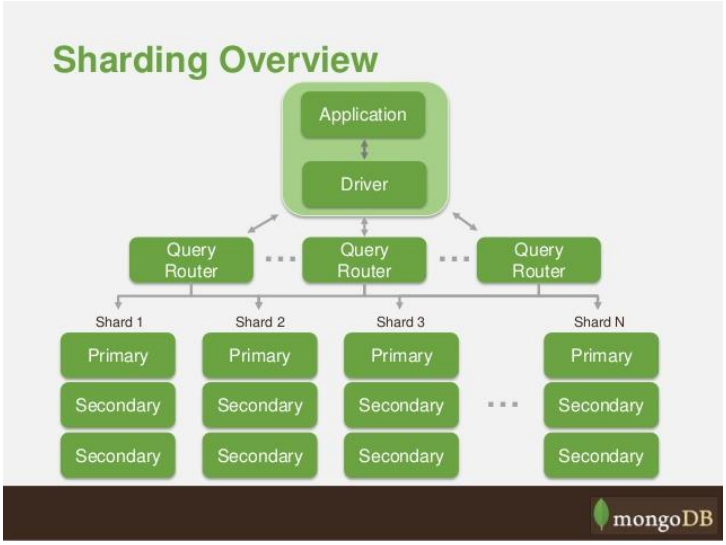

Fig. 10. Sharding built-in di MongoDB

3.3 Comparazione database

La Tabella 3-1 mostra le principali caratteristiche analizzate per ciascuna delle due tecnologie. Prima di tutto vanno sottolineati gli aspetti più importanti da tenere in considerazione tra quelli presenti al suo interno. Tali aspetti sono guidati dalle caratteristiche fondamentali riguardanti i Big Data cioè Volume, Velocità e Varietà.

Il primo, importantissimo, è legato alla capacità di un Data Warehouse di scalare facilmente. MongoDB nasce per i Big Data ed è costruito per poter scalare orizzontalmente attraverso sistemi di sharding e replication built-in. PostgreSQL è in grado di scalare verticalmente, ma le nuove versioni integrano sistemi che, seppur complessi, permettono al sistema di applicare le stesse tecniche. Si ricorda che PostgreSQL non è nato per supportare i Big Data, ma si è orientato verso quella direzione negli anni e rappresenta una tecnologia adattissima per assumere il ruolo di Data Warehouse, essendo in grado di interagire con strumenti esterni per il Data Mining e per il Reporting.

La velocità è legata evidentemente alle performance. Secondo delle ricerche rese note sul sito della Conferenza di Percona sui database open source, di cui le Tabelle 3-2 e 3-3 mostrano un resoconto, PostgreSQL risulta essere assai più performante rispetto a MongoDB.

I Big Data possono essere strutturati o semi-strutturati. Per questo motivo la varietà è una caratteristica che i database moderni devono affrontare. Il formato JSON supportato da entrambe le soluzioni risulta essere molto adatto per gestire queste tipologie di dati. Il fatto che, sia MongoDB che PostgreSQL, abbiano puntano a una sua versione in formato binario per aumentare le prestazioni delle query, sta a indicare che le soluzioni moderne sono orientate in questa direzione.

Da non sottovalutare un ultimo aspetto molto importante: l'affidabilità. Nonostante si parli di Big Data, dove questo aspetto può venir meno, Carpigiani non si può permettere di memorizzare un dato che perde di consistenza. PostgreSQL, essendo un database nato con il modello relazionale, rispetta ancora le proprietà ACID e quindi da questo punto di vista risulta molto più affidabile. Da questa attenta valutazione si può trarre che PostgreSQL appare la soluzione più adatta per ricoprire il ruolo di Data Warehouse nel nuovo progetto Big Data Carpigiani.

Funzionalità	**MongoDB**	**PostgreSQL**
Nascita tecnologia open source	2009	1995
Schema	Dinamico (schema-less)	Dinamico e Relazionale
Supporto ai JSON Document	Sì: JSON e BSON	Sì: JSON e JSONB
Supporto a key/value Data	Sì	Sì (dal 2006)
Supporto a Relational Data / Normalized Form Storage	No	Sì
Transazioni atomiche	All'interno di un documento	Sì
Linguaggi web supportati	JavaScript, Python, Ruby, e altri…	JavaScript, Python, Ruby, e altri…
Supporto ai dati Geo-spaziali	Sì	Sì
Scalabilità	Orizzontale	Verticale
Sharding e Replication	Built-in: Facile	Complesso
Programmazione server side	No	Molto supportata
Query potenti	No	Sì
Join con chiave esterne	No	Sì
Affidabilità	Bassa	Alta
Latenza	Non stabile	Bassa

Domini applicativi	Big Data, con alta concorrenza e dove la consistenza dei dati non è richiesta	Operazioni transazionali, OLTP, Big Data (Data Warehouse), Data Mining, OLA, Reporting

Tabella 3-1 Confronto caratteristiche di MongoDB e PostgreSQL

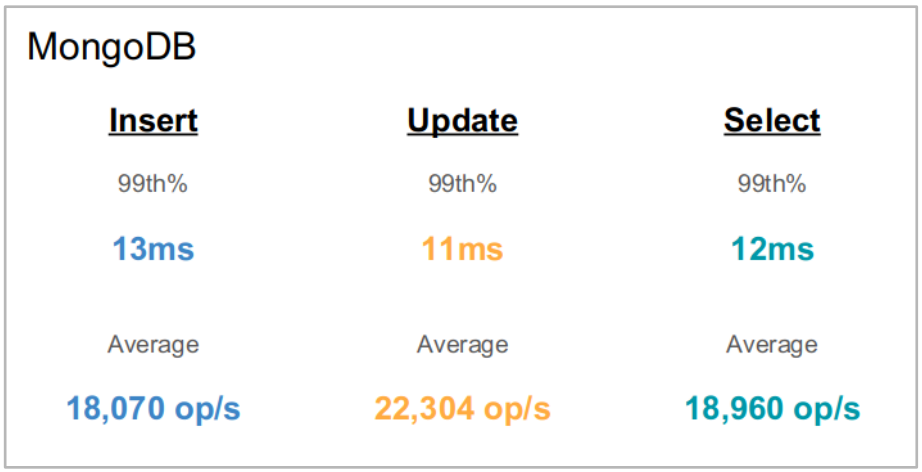

Tabella 3-2. Performance di MongoDB

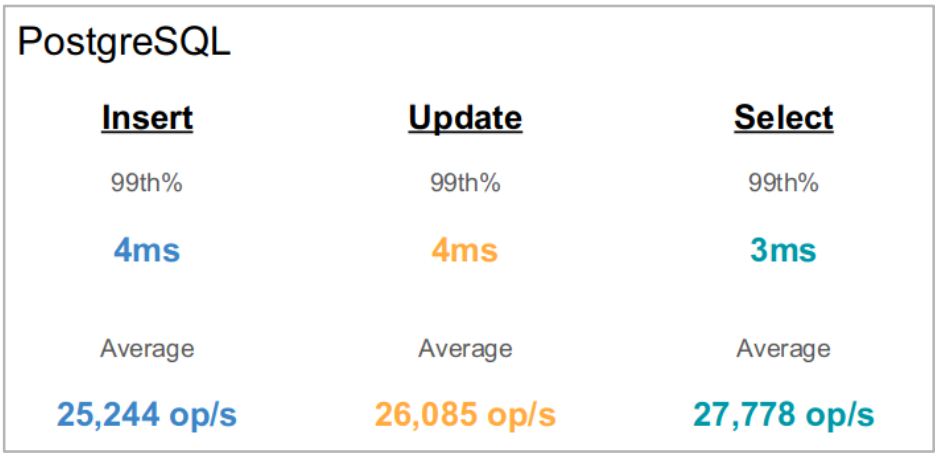

Tabella 3-3. Performance di PostgreSQL

4. Analytics B2B: piattaforme

Un applicativo di Business Consulting adatto nel nostro contesto deve garantire la possibilità di normalizzare il modello dei dati, effettuare delle aggregazioni e permettere una visualizzazione trasversale ad ampio spettro attraverso grafici e dashboard customizzabili.

4.1 Pentaho

Pentaho è un sistema di Business Intelligence progettato per aiutare le aziende a prendere decisioni basate sui dati, con una piattaforma per l'integrazione e l'analisi dei dati stessi. La piattaforma include estrazione, trasformazione e caricamento (ETL), analisi dei big data, visualizzazioni, dashboard, reporting, Data Mining e un sistema di analisi predittiva. La funzionalità di integrazione dei dati di Pentaho consente agli utenti di trovare, gestire e combinare dati da più fonti, incluso il supporto nativo per database analitici, Hadoop e NoSQL. Il software offre strumenti di analisi aziendale interattivi come analisi visive e dashboard, oltre a soluzioni di reporting flessibili. L'analisi predittiva offerta dal sistema include algoritmi di apprendimento automatico, strumenti per l'elaborazione dei dati e la capacità di importare modelli di terze parti.

4.1.1 Architettura di Pentaho

Pentaho è una piattaforma software che si compone di diversi tool i quali possono essere utilizzati in maniera totalmente indipendente, tanto che può essere considerata una vera e propria suite piuttosto che un singolo prodotto. Tale piattaforma è disponibile in 2 distribuzioni differenti:

- Community Edition (C.E.): una versione gratuita e open source che viene gestita interamente dalla community di Pentaho
- Enterprise Edition (E.E.): è la versione a pagamento, più completa della precedente a livello di funzionalità disponibili e di supporto tecnico

Il software è costruito su un'architettura aperta e può essere integrato con più sistemi non facenti parti della suite, lasciando così ampia libertà di utilizzo. La struttura definisce solo i livelli e non gli strumenti che devono essere utilizzati.

Si parla di una architettura modulare, come mostrato in Figura 11. Lo stack di Pentaho si suddivide in:

1. Presentation Layer (Thin Client): Pentaho User console è la principale interfaccia web-based che permette all'utente finale di visualizzare, creare e pianificare report dinamici e dashboard customizzate.

2. Business Intelligence Platform: il Server Applications supporta la Pentaho User Console e consiste in tutto il sistema di programmazione, gestione e sicurezza del dato. Il Server Enterprise Console supporta la Enterprise Console e include i servizi per la gestione del repository, della programmazione e della sicurezza e la configurazione del server. Il Data Integration Server viene invece utilizzato per eseguire i lavori di integrazione e trasformazione del dato e offre servizi come la pianificazione e la gestione dei contenuti (compresa la revisione e l'integrazione della sicurezza).

3. Data and Application Integration (Power Tools): corrisponde al sistema di ETL di Pentaho. Sono gli strumenti desktop di progettazione che permettono di modellare il dato e di creare un ricco reporting.

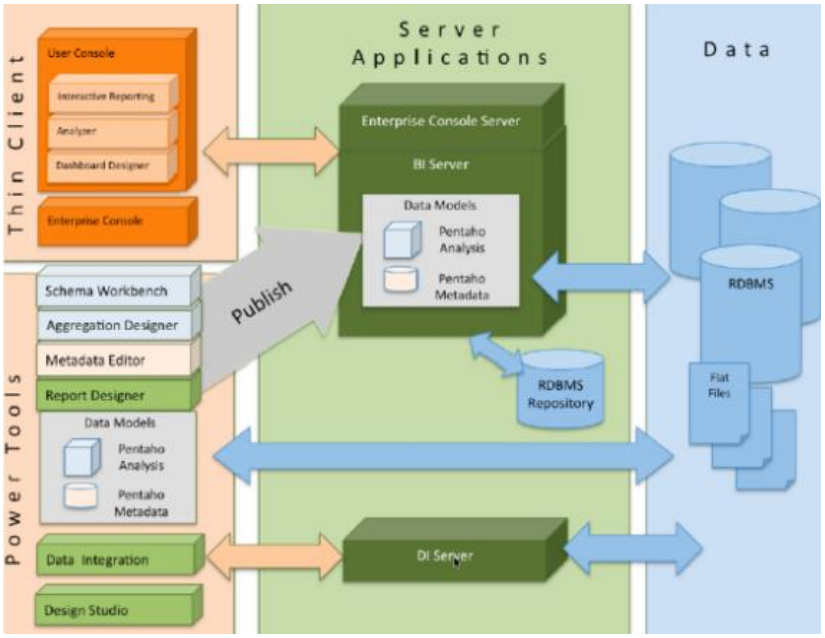

Fig. 11. Architettura di Pentaho.

4.2 KNIME

KNIME è una soluzione di Business Intelligence open source, molto simile a Pentaho, per l'analisi dei dati e per il reporting, che integra vari componenti per l'apprendimento automatico e il Data Mining. Un'interfaccia grafica permette all'utente il montaggio di nodi per la pre-elaborazione, la modellazione, l'analisi e la visualizzazione

dei dati. L'applicativo consiste proprio in un sistema di ETL con possibilità di realizzare visualizzazioni a partire dalle aggregazioni effettuate con i dati.

Nella piattaforma di analisi KNIME, le singole attività sono rappresentate da nodi. Ogni nodo viene visualizzato come una casella colorata con porte di input e output, nonché uno stato, come mostrato nella Figura 12. Gli input sono i dati che il nodo elabora e gli output sono i set di dati risultanti. Ogni nodo ha impostazioni specifiche, che si possono regolare in una finestra di configurazione. Quando questo avviene, lo stato del nodo cambia, mostrato da un semaforo sotto ciascun nodo. I nodi possono eseguire tutti i tipi di attività, tra cui lettura / scrittura di file, trasformazione di dati, modelli di formazione, creazione di visualizzazioni e così via. Oggi KNIME conta più di 2500 diversi nodi e nuove funzionalità sono in continuo sviluppo e spingono verso innovativi metodi di analisi. Una collezione di nodi interconnessi costituisce un flusso di lavoro (workflow).

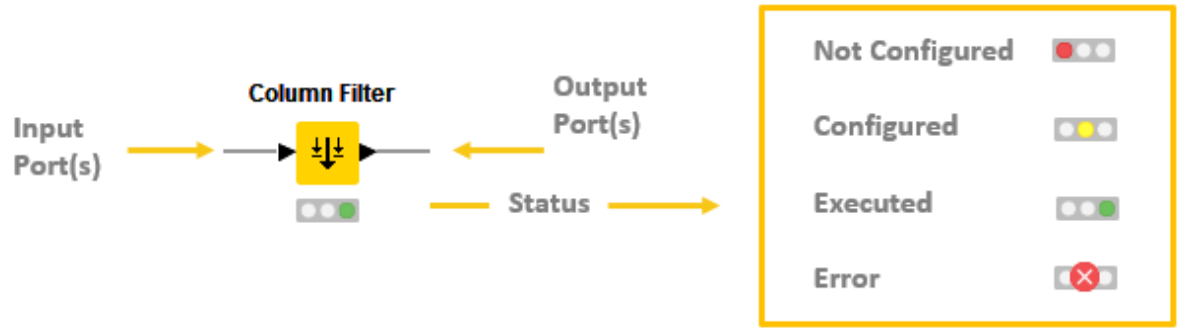

Fig. 12. Nodo di KNIME

4.2.1 Architettura di KNIME

Il cuore di KNIME è appunto la KNIME Analytics Platform, il tool grafico descritto in precedenza, che permette di elaborare operazioni di ETL avanzate e costruire dashboard interattive a partire dai dati. È possibile però aggiungere funzionalità alla piattaforma di analisi, installando estensioni o integrando software esterni. Le estensioni disponibili possono essere open source gratuite fornite da KNIME, estensioni gratuite fornite dalla community oppure estensioni commerciali, compresi i nuovi nodi tecnologici forniti dai partners di KNIME.

Come mostrato in Figura 13, la parte open source offerta da KNIME non comprende il lato Server. Il KNIME Server aggiunge le seguenti funzionalità (Figura 14):

- Collaboration: permette di esportare/importare facilmente workflow in un team di lavoro esteso, grazie a un repository condiviso
- Deployment: facilita la gestione degli ambienti di test e production e offre un'interfaccia di accesso sul browser.
- Security: integra un sistema di autenticazione e autorizzazione avanzato
- Scheduling: permette di automatizzare l'elaborazione dei vari workflows sviluppati
- API REST: fornisce delle API di alto livello che facilitano l'integrazione dei workflows prodotti con applicazioni esterne, attraverso chiamate da remoto
- Management: semplifica la gestione della sicurezza, degli utenti e del versioning

Il deploy del server può essere realizzato on-premises oppure sul cloud, supportato da AWS o da Microsoft Azure. In questo caso ovviamente si hanno dei vantaggi in termini di scalabilità e di costi di manutenibilità del sistema.

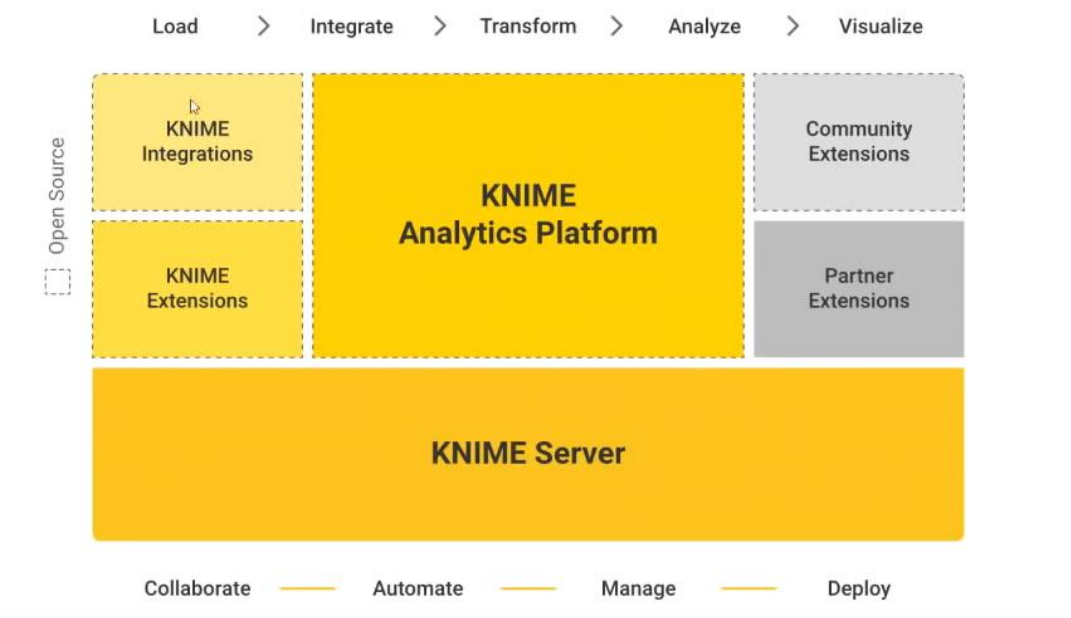

Fig. 13. Architettura di KNIME

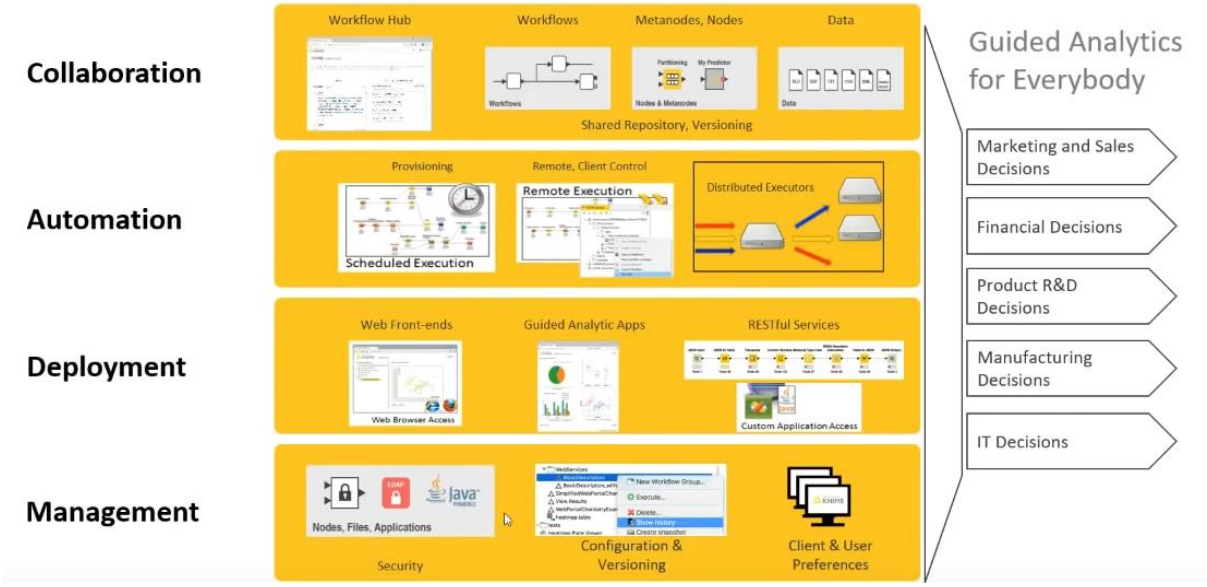

Fig. 14. Funzionalità del Server KNIME

4.2.2 KNIME Report Designer

L'estensione KNIME Report Designer integra BIRT nella piattaforma di analisi KNIME e consente di creare report basati sui risultati dei flussi di lavoro. BIRT (Basic Intelligence Reporting Tool) è un software open source utilizzato per i report. L'estensione fornisce un template di report per ciascun flusso di lavoro e per ciascun nodo, in modo da identificare le tabelle di dati che dovrebbero essere disponibili all'interno del report. La creazione di un report consiste in due attività separate:

- Preparare i dati che si desidera utilizzare nel report. Questo passaggio viene eseguito in un flusso di lavoro KNIME.
- Progettare, modellare e modificare la presentazione di questi dati con l'editor del template.

Pertanto l'editor del flusso di lavoro KNIME mostra in modo completo tutta la pre-elaborazione che viene eseguita sui dati, mentre l'editor del template rivela come i dati verranno presentati nel report.

4.3 Elastic

Elastic è una search company che offre sul mercato un prodotto, disponibile sia sul cloud che in versione on-premises, il quale può essere considerato un vero e proprio ecosistema informatico, nato per rispondere a una suite di casi d'uso legati al mondo dei Big Data. Questa soluzione viene denominata Elastic stack ELK. L'acronimo deriva appunto dalle iniziali dei moduli principali: Elasticsearch, Logstash e Kibana.

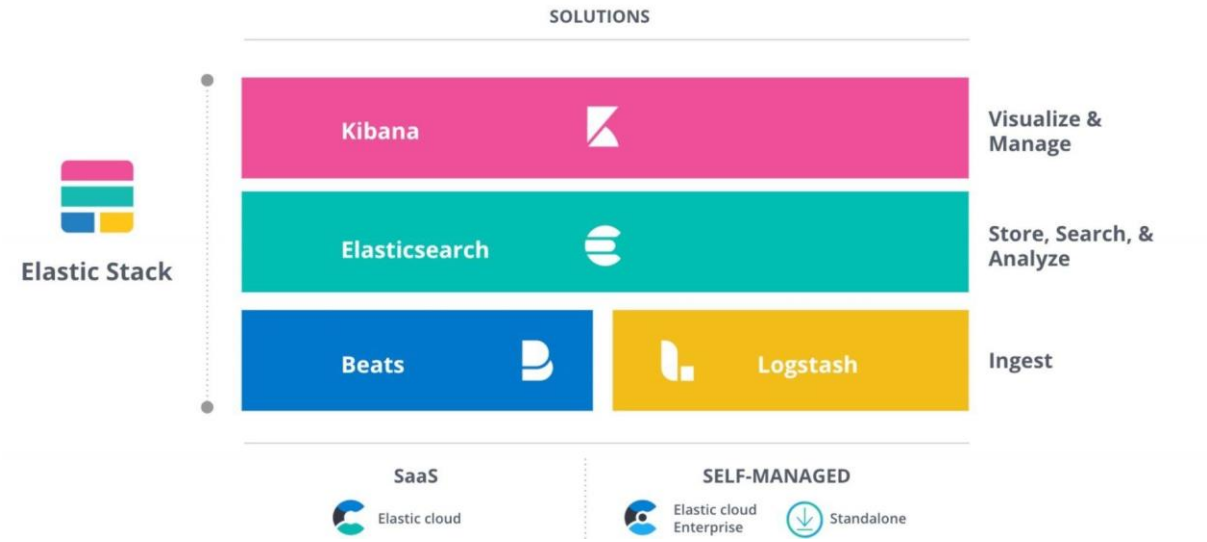

Fig. 15. Elastic Stack ELK

Come possiamo notare dalla Figura 15, in realtà lo stack si compone di quattro moduli:

- Beats e Logstash: si occupano dell'ingestione dei dati
- Elasticsearch: motore di ricerca in cui vengono indicizzati e memorizzati i dati
- Kibana: consente di visualizzare graficamente i dati contenuti su Elasticsearch e di gestire l'intero stack

Lo stack Elastic completo è disponibile sia in versione Cloud, con un'interfaccia di gestione molto pratica e veloce, sia scaricabile sulle proprie macchine server e quindi utilizzabile on-premises. Ci sono due distribuzioni gratuite, una open source gestita da AWS che però fornisce un set di funzionalità limitato e uno scarso supporto tecnico

e la versione Basic, interamente gestita da Elastic, che permette di sfruttare molte delle funzionalità che interessano per questo progetto. Elastic Basic non è interamente open source, in quanto è sotto la licenza Elastic 2.0 che impone dei limiti nella modifica e ridistribuzione del codice. Non è possibile infatti creare un'applicazione a partire dal codice sorgente e rivenderla ai propri clienti. Si descrivono ora i moduli software che compongono la piattaforma. Si è deciso di approfondire la versione Basic, perché il set di funzionalità garantito era sufficiente. Inoltre durante la fase di studio sono state aggiunte nuove funzionalità gratuite per superare la concorrenza del servizio gestito da Amazon, dettaglio che auspicava un forte interessamente da parte di Elastic alla proprio versione base.

4.3.1 Elasticsearch

Il cuore di Elastic Stack è un motore open source distribuito per la ricerca e l'analisi real-time dei dati, basato su Apache Lucene[1]. È una piattaforma di ricerca near Real time e ciò indica che c'è una leggera latenza dal momento in cui si indicizza un documento fino al momento in cui diventa ricercabile. La sua flessibilità e la sua potenza sono particolamente adatte a rispondere alle esigenze di scalabilità orizzontale. Elasticsearch combina le funzionalità della ricerca full-text (come faceting, autocomplete, ecc.) con una potenza e una velocità che gli permettono di muoversi con agilità su grandi quantità di dati strutturati o non strutturati. Elasticsearch comunica con un protocollo HTTP/JSON, che permette di indicizzare, memorizzare ed analizzare dati in formato JSON e fornisce API di alto livello molto potenti per semplificare queste procedure e l'interrogazione dei dati stessi. Sempre attraverso le suddette API è possibile interfacciarsi anche con applicativi esterni alla piattaforma.

Originariamente le soluzioni software proposte da Elastic erano tutte legate alla ricerca full-text. Si parla di Enterprise Search per facilitare la ricerca all'interno di una gestione documentale in ambito aziendale, Site Search in ambito Web e infine App Search per la ricerca generica all'interno di un'applicazione. Di fatto, Elasticsearch è la prima tecnologia a proporre un sistema che permetta di eseguire ricerche full-text in modo perfettamente parallelo, scalabile e real-time. Per questa caratteristica e per il suo primato nel settore, Elasticsearch è il secondo motore di ricerca più popolare nel mondo. Tra i suoi utilizzatori più famosi ricordiamo Mozilla, GitHub, CERN e Netflix. I successivi sviluppi hanno portato l'azienda a orientarsi verso nuovi use cases legati al mondo del monitoraggio delle applicazioni (APM), dei log e della sensoristica IoT. Le ultime release, che contengono il sistema SIEM (Security Information & Event Management), si sono sp[1]ostate verso il mondo della

[1] **Apache Lucene** è una libreria software per motori di ricerca gratuita e open source, originariamente scritta in Java. Sebbene adatto a qualsiasi applicazione che richieda l'indicizzazione e la capacità di Nella ricerca *full-text*,

sicurezza. Infatti permettono di studiare lo storico dei dati legati alla sicurezza di un sistema informatico, intrecciandoli con i dati ottenuti attraverso il monitoraggio in tempo reale.

Gli elementi chiave che costituiscono la piattaforma si possono riassumere nei seguenti termini:

- Cluster: è una raccolta di uno o più nodi (server) che tiene insieme tutti i dati e fornisce funzionalità di indicizzazione e ricerca federate su tutti i nodi. All'interno di un cluster è possibile aggiungere e rimuovere nodi a piacimento senza alterarne in alcun modo il funzionamento. Questo rende di fatto Elasticsearch uno strumento dalle potenzialità enormi, permettendo al sistema di scalare orizzontalmente in maniera quasi automatica.

- Nodo: è un singolo server che fa parte del cluster, memorizza i dati e partecipa alle funzionalità di indicizzazione e ricerca del cluster. Proprio come un cluster, un nodo è identificato da un nome che per impostazione predefinita è un IDentificatore Universale Univoco (UUID) casuale assegnato all'avvio. I nodi possono assumere ruoli differenti a secondi del contesto e delle esigenze architetturali:

    - Master: sono i nodi standard, che quindi fanno tutto, ma in un sistema complesso si occupano della gestione del cluster

    - Data: mantengono i dati indicizzati e gestiscono le operazioni legate a esse; si possono differenziare in nodi Hot e Warm per poter discriminare i dati più o meno significativi e gestire il loro ciclo di vita

    - Ingest: sono nodi specifici che sostituiscono il ruolo di Logstash, quindi permettono la connessione diretta a sorgenti dati differenti per la raccolta all'interno del cluster

    - Coordinating: in un sistema complesso con più nodi e cluster, ogni nodo deve gestire le operazioni di distribuzione; effettuano il routing, gestiscono l'indicizzazione e la ricerca dei dati distribuita

    - Alerting: sono i nodi che eseguono i job di alerting

    - Machine Learning: sono i nodi che eseguono i job di Machine Learning.

Data la natura distribuita di Elasticsearch, la sua architettura pone al centro della complessità i Nodi e la loro organizzazione. Elasticsearch utilizza il paradigma Master-Slave per gestire la comunicazione tra i nodi. Quando

---

Lucene è riconosciuta per la sua utilità nell'implementazione di motori di ricerca su Internet e nella ricerca locale in un unico sito.

una nuova istanza di Elasticsearch viene avviata, un nodo master viene scelto tra quelli disponibili attraverso una politica di elezione. Il nodo master è responsabile sia dell'organizzazione dei dati all'interno del cluster, sia dell'aggiunta e rimozione di un nodo attivo o malfunzionante. Se il nodo master smette di funzionare, Elasticsearch ne eleggerà uno nuovo.

- Indice: è una raccolta di documenti con caratteristiche in qualche modo simili. Ad esempio, è possibile avere un indice per i dati dei clienti, un altro indice per un catalogo prodotti e ancora un altro indice per i dati dell'ordine. Un indice è identificato da un nome (che deve essere tutto in minuscolo) e questo nome viene utilizzato per fare riferimento all'indice quando si eseguono operazioni di indicizzazione, ricerca, aggiornamento ed eliminazione dei documenti in esso contenuti. In realtà un indice è soltanto un namespace logico che punta a uno o più shard.

- Shard: è un'unità di lavoro di basso livello che contiene una porzione di dati dell'indice. Ogni shard è una singola istanza Lucene, dunque è un sistema di ricerca autonomo. Gli shard costituiscono lo strumento con cui Elasticsearch distribuisce i dati all'interno del cluster, ed è per questo che un indice può essere distribuito su più nodi. Esistono due tipi di shard: primario e replica. Le repliche sono delle semplici copie di uno shard primario, che permettono al cluster di continuare a funzionare anche in caso di fault o sostituzione di un nodo. Se un nodo qualsiasi smette di funzionare, le repliche vengono promosse a shard primario.

- Documento: è un'unità di informazioni di base che può essere indicizzata e che possiede un ID univoco. Ad esempio, è possibile avere un documento per un singolo cliente, un altro documento per un singolo prodotto e un altro per un singolo ordine. Questo documento è espresso in JSON.

4.3.2 Logstash e i Beats

Logstash e beats rappresentano insieme un modulo fondamentale dello stack, che è quello della raccolta, indicizzazione, normalizzazione e modellazione dei dati.

I Beats sono "trasportatori di dati" (Data shipper) leggeri che collezionano e trasportano svariate tipologie di dati verso Elasticsearch. I beats sono stati modularizzati e sono già predisposti a raccogliere dati da diverse tipologie di fonti. Alcuni esempi sono i filebeat (legge i file di log), i winlogbeat (legge gli eventi di un sistema Windows), i packetbeat (legge i dati relativi al traffico di rete), i metricbeat (legge metriche di sistema e applicative) e molti altri. Sono componenti che sfruttano poca memoria e che quindi sono adatti per essere installati direttamente sull'host da cui vengono raccolti i dati.

Logstash invece è modulo molto più complesso e che sfrutta molte risorse e per questo necessita di essere installato sul server e può anche essere scalato orizzontalmente. Ha la funzione di raccogliere ed elaborare dati e log da qualsiasi fonte. Grazie a questo componente è possibile la normalizzazione e la variazione di schemi e formati. Rispetto ai Beats svolge funzioni più avanzate sui dati, permettendo di fare azioni come parsing ed enrichment. Rappresenta il componente ETL dello stack Elastic.

Al suo interno è possibile configurare una o più pipeline per il percorso dei dati. La pipeline (Figura 16) è costituita da una parte di Input che si connette alla fonte del dato, poi da una parte centrale, Filter, che permette di effettuare numerose operazioni sul dato (filtraggio, modellazione, normalizzazione) e allo stesso tempo interrogare altre fonti esterne (come database di metadati) per arricchirlo e infine una parte di Output, che consente di indicizzare il dato e caricarlo su un sistema di storage, che può essere Elasticsearch o un altro database. Per ognuno di questi tre elementi esistono dei plugin che permettono di realizzare le sopra citate operazioni. I plugin sono scritti per la maggior parte in Java ed è possibile svilupparne di nuovi per le esigenze che sopraggiungono, ma questi devono essere open source. Esistono numerosi plugin sviluppati dagli utenti della piattaforma. Logstash è dotata di una sistema di caching che permette di memorizzare le query effettuate e quindi velocizzare e alleggerire il processo di arricchimento del dato. È possibile anche configurare una coda persistente per evitare di perdere dati all'interno della pipeline in caso di crash dell'applicativo.

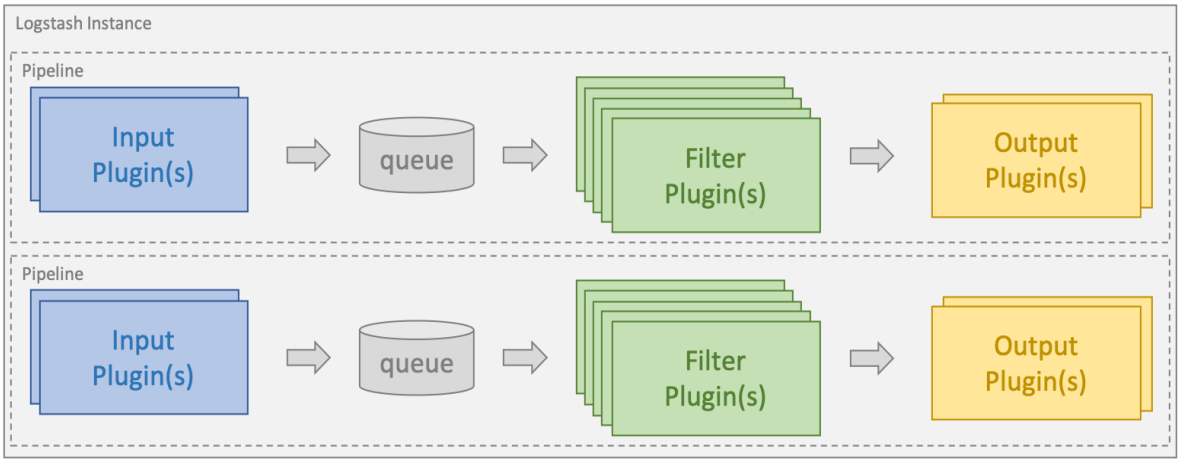

Fig. 16. Architettura di Logstash

4.3.3 Kibana

Kibana è un'interfaccia web estensibile per la presentazione visiva dei dati raccolti, quindi costituisce il lato front-end dello stack. È uno strumento della suite di Elastic che fornisce un'interazione dinamica con tutti i dati grazie alle potenti API e a un sistema di discovery avanzato. Usando aggregazioni e filtri predefiniti, Kibana permette la creazione semplificata di dashboard, grafici e tabelle, istogrammi e mappe termiche. Esso è dotato di potenti funzionalità geospaziali che consentono di sovrapporre senza soluzione di continuità le informazioni geografiche sopra i dati e visualizzare i risultati sulle mappe. Allo stesso tempo facilita la gestione dell'intero stack tramite un'interfaccia di controllo. Kibana produce layout di visualizzazione che permettono di catturare a colpo d'occhio l'intero valore dei dati facilitandone la visione complessiva e il monitoraggio. È possibile trascinare dinamicamente le finestre temporali, ingrandire e rimpicciolire specifici sottoinsiemi di dati ed eseguire il drill down sui report per estrarre informazioni fruibili dai dati. Kibana inoltre permette di creare degli "spazi" ad hoc per team di lavoro o per tipologia di dati da visualizzare. Questa funzionalità può essere molto utile per discriminare l'utente che andrà a utilizzare l'interfaccia web.

Il primo elemento è il sistema Discover. Una panoramica dettagliata sui dati che, come mostrato in Figura 17, fornisce le seguenti funzionalità:

- Mostra i dati in formato JSON oppure "spacchettati" in formato tabellare
- Fornisce un'idea di quanti dati vengono raccolti in un arco temporale specifico
- Permette di effettuare filtri sui campi e creare tabelle che inquadrano valori di interesse selezionati
- Mostra una statistica dei valori posseduti per ogni campo

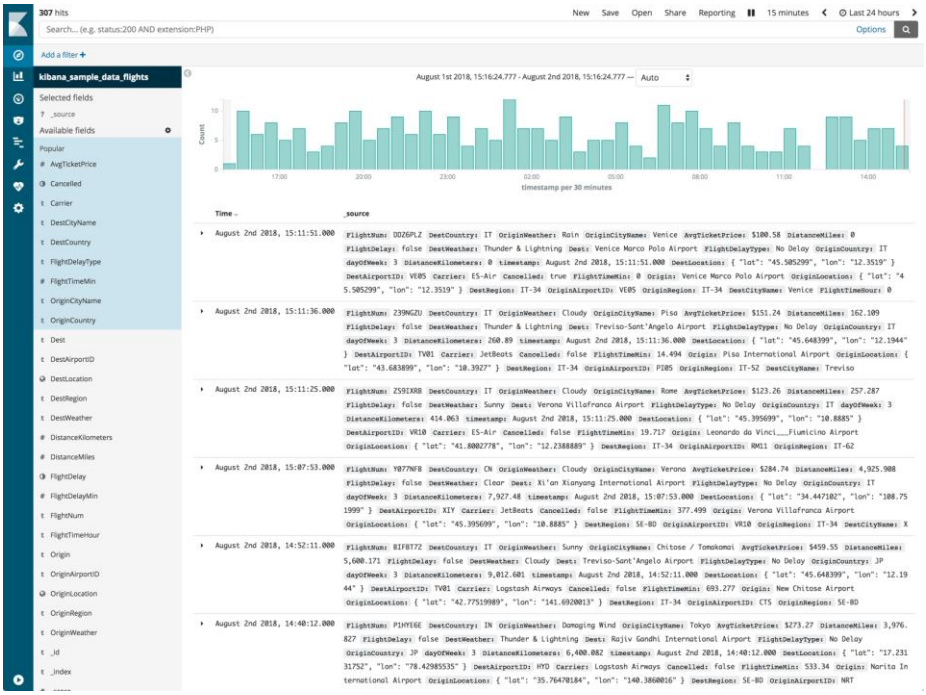

Fig. 17. Discover di Kibana

L'interfaccia di Management fornisce una panoramica generale che mostra lo stato di Elasticsearch e Kibana (nelle versioni a pagamento è possibile configurare le pipeline di Logstash direttamente su Kibana).

Lo spazio legato a Elasticsearch permette di gestire gli indici creati, compreso le politiche legate al loro ciclo di vita, in più fornisce un'interfaccia per gestire i documenti salvati, gli shard e i cluster. Per quanto riguarda Kibana, invece, da qui è possibile creare l'index pattern, gestire i vari campi del documento che Elasticsearch riconosce: gestione dei tipi, creazioni di script che modellano il dato, ecc…Infine mostra una panoramica di tutti gli oggetti salvati (visualizzazioni, dashboard, reports, ecc..), con la possibilità di gestire i vari spazi creati per i team di lavoro.

La piattaforma web possiede una console che permette di interagire a pieno con le API REST del protocollo per creare indici, interrogare e modellare i dati. È possibile inoltre monitorare lo stato dello stack e visualizzare lo storico delle richieste e delle risposte.

4.4 Considerazioni Architetturali e Tecnologiche

Le considerazioni sono riassunte in Tabella 4-1. Come si può notare tutte le soluzioni analizzate sono caratterizzate da un sistema in grado di fornire funzionalità avanzate di ETL, Data Integration e Data Mining. Sicuramente Pentaho e Elastic puntano molto sul lato ETL, infatti facilitano molto l'elaborazione delle pipeline che lavorano sul dato, senza chiedere agli sviluppatori particolari competenze su linguaggi di querying specifici (per esempio SQL). Questo perché viene tutto reso trasparente dalla piattaforma, che rende disponibili pannelli di configurazione per i vari nodi che caratterizzano la pipeline. L'interfaccia grafica di KNIME, in particolare, è molto user-friendly. Come descritto in precedenza, il componente dello stack ELK che svolge il ruolo di ETL è Logstash. Il componente non fornisce un'interfaccia grafica per la sua configurazione, e per essere utilizzato richiede lo studio di un linguaggio specifico. Allo stesso tempo però, grazie alla presenza dei plugin (programmabili in Java), Logstash ha potenzialità enormi di estendibilità e adattabilità al contesto applicativo. La presenza stessa di una community che facilita la diffusione di plugin riutilizzabili è un grossissimo vantaggio.

Un altro elemento in comune, sempre a favore di tali soluzioni, è la forte compatibilità con i sistemi legati al mondo del Big data. Infatti tutte e tre gli applicativi possono connettersi ai maggiori database sul mercato (compreso PostgreSQL), ai framework di calcolo Big Data come Hadoop, Spark oppure alle piattaforme di streaming e messaggistica più importanti come Kafka, RabbitMQ.

È stato evidenziato che la possibilità di creare dashboard in base alle esigenze è un elemento strategico necessario per qualsiasi azienda. Carpigiani necessita di una strumento simile per superare i limiti evidenziati su Teorema. Elastic è in grado di fornire questa funzionalità ed è sicuramente la migliore tra le tre soluzioni. A livello di grafica è decisamente la più moderna e innovativa, ma soprattutto garantisce una libertà di customizzazione eccezionale. Inoltre i grafici all'interno della dashboard sono tutti collegati, cioè variano dinamicamente (e in contemporanea) nel momento in cui vengono applicati dei filtri o quando si va a cliccare su un particolare elemento di una visualizzazione. Questo favorisce enormemente le operazioni di drill down sul dato, cioè la possibilità di conoscere le motivazioni che portano a certi comportamenti. KNIME permette di costruire dashboard interattive molto potenti. La configurazione delle varie visualizzazione, sulla piattaforma desktop risulta però alquanto complessa. Queste operazioni vengono semplificate sul portale web, il quale però è compreso solamente nel pacchetto commerciale. Pentaho dà la possibilità di creare delle dashboard, ma nella versione open source queste non sono customizzabili liberamente, ma al contrario vanno costruire a partire da modelli prefissati. La grafica di Pentaho inoltre risulta obsoleta e poco intuitiva.

Un limite di Elastic è sicuramente dato dal reporting. La versione Basic permette solo di condividere dei file CSV come risultato. Il reporting avanzato è compreso nei pacchetti con licenza commerciale. In realtà, grazie alle API REST, questo problema può essere risolto utilizzando un applicativo esterno. Pentaho dispone di un particolare

tool per effettuare queste operazioni, invece KNIME utilizza BIRT che è un software open source esterno. Entrambi permettono di creare un PDF, con la possibilità di esportarlo.

Un tema importante è quello del deployment. Pentaho e KNIME, oltre alla versione on-premises, offrono una versione cloud appoggiandosi sui massimi vendor, AWS e Microsoft Azure. Per quanto riguarda Elastic il discorso è più complesso. Esistono due distribuzioni cloud, una fornita da AWS e una direttamente da Elastic. Le due soluzioni sono in forte competizioni, soprattutto nell'ultimo anno, tanto che Elastic ha deciso aggiungere funzionalità per combattere la concorrenza. In particolare, il supporto tecnico di Elastic è molto più competente e a stretto contatto con il cliente e questo è un punto a favore per l'azienda. La piattaforma cloud inoltre facilita molto la gestione: operazioni come l'aumento dello storage, dei nodi, della velocità sono a portata di un click con il mouse. Infine i costi della soluzione cloud di Elastic sono sicuramente più contenuti rispetto a quelli di Pentaho e KNIME.

Elastic ha infine un grandissimo vantaggio costituito dal modulo centrale, cioè il sistema di storage Elasticsearch. Avere la possibilità di memorizzare i dati al proprio interno, nel proprio formato JSON, aumenta notevolmente le performance delle elaborazioni e offre al sistema funzionalità fondamentali. Elasticsearch infatti, come la maggior parte dei database NoSQL, permette di scalare orizzontalmente in modo molto semplice ed intuitivo. Di fatto è possibile aggiungere e rimuovere nodi a caldo senza danneggiare in alcun modo l'integrità del sistema. Attraverso le tecniche incorporate di sharding e replicazione garantisce e gestisce automaticamente un'alta affidabilità del sistema senza dover investire particolarmente nelle strutture. Anche la ricerca full-text costituisce un vantaggio, perché permette di effettuare operazioni di filtraggio in tempi molto rapidi. Così come le aggregazioni sono facilitate da un set di API estremamente potenti, specifiche per questo tipo di operazioni, raggiungendo livelli di complessità di molto superiori a quelle di un comune database relazionale.

A partire dalle considerazioni appena descritte, si può intuire che Pentaho è un'ottima tecnologia, ma che non fornisce la possibilità di customizzare dashboard e ciò rappresenta una funzionalità troppo importante al fine di raggiungere gli obiettivi imposti da Carpigiani per questo progetto. Pentaho è certamente uno strumento molto potente, compatibile con le principali tecniche di calcolo Big Data (per esempio Hadoop), ma principalmente è costruito per elaborare i dati in modalità batch e non stream. Analizzando quindi le altre due tecnologie, Pentaho è stata la prima esclusa.

KNIME ed Elastic risultano molto adatte in termini di prestazioni e funzionalità offerte. La prima soluzione può essere preferita per quanto riguarda il tool grafico di ETL, sicuramente più avanzato rispetto al corrispondente Logstash, perché semplifica il lavoro dello sviluppatore. Tuttavia, le funzionalità server fanno parte di un pacchetto commerciale troppo costoso. Elastic, invece, fornisce quasi tutto ciò di cui Carpigiani ha bisogno senza costi aggiuntivi. Inoltre la parte front-end possiede una grafica molto più moderna e intuitiva e dà una forte libertà nella customizzazione delle proprie dashboard.

Si è visto come il modulo Elasticsearch inoltre fornisca dei vantaggi enormi in termini di affidabilità e prestazioni grazie al motore Lucene. Un aspetto da non sottovalutare è quello di un possibile investimento futuro, sia per una migrazione verso il cloud, sia per quanto riguarda le potenzialità che possono dare le tecniche di Machine Learning. I costi di Elastic sono inferiori a quelli di KNIME.

Da queste considerazioni si evince che lo stack Elastic risulta essere il più convincente per diversi fattori come l'architettura molto scalabile, i costi ridotti, la grafica avanzata.

Soluzione	Pro	Contro
Pentaho	• Data integration, ETL e Data Mining avanzati • Reporting completo • Compatibilità con diversi sistemi (storage, messaggistica, Big Data) • Non serve una conoscenza del linguaggio SQL	• Customizzazione dashboard limitata nella versione community • Grafica obsoleta • Licenza molto costosa (sia cloud che on-premises)
KNIME	• Data integration, ETL avanzati • Compatibilità con diversi sistemi (storage, messaggistica, Big Data) • Non serve una conoscenza del linguaggio SQL	• Versione cloud molto costosa • Funzionalità server solo nella versione commerciale, molto costosa
Elastic	• Data integration, ETL, Machine Learning avanzati • Sistema affidabile e scalabile • Dashboard customizzabili • Forum molto utilizzato e Documentazione precisa • Grafica innovativa • Interfaccia potente e intuitiva • API REST potenti • Supporto sul cloud avanzato • Compatibilità con diversi sistemi (storage, messaggistica, Big Data)	• Reporting, Machine Learning e Alerting a pagamento • Licenza on-premises costosa

Tabella 4-1. Confronto delle piattaforme di Analysis

4.5. Architettura Finale e Considerazioni Deployment

La Figura 18 mostra la nuova architettura, nella quale possiamo individuare tre passaggi fondamentali.

Il primo passaggio è costituito dal **Cleaner**, un modulo software realizzato internamente a Carpigiani che permette di spacchettare i dati dal formato "grezzo" nel nuovo formato **JSONB**, caricandoli poi sul Data Warehouse **PostgreSQL**. Il secondo passaggio è rappresentato dal cambio di Data Model, dove i dati vengono prelevati in formato JSONB attraverso **Logstash** - che si occupa di arricchirli, applicare filtri e modifiche prima di caricarli su **Elasticsearch**. Il terzo ed ultimo passaggio si verifica ogniqualvolta l'utente crea su Kibana una nuova visualizzazione o dashboard, mostrando i risultati dei parametri scelti con eventuali operazioni sugli andamenti e/o aggregazioni.

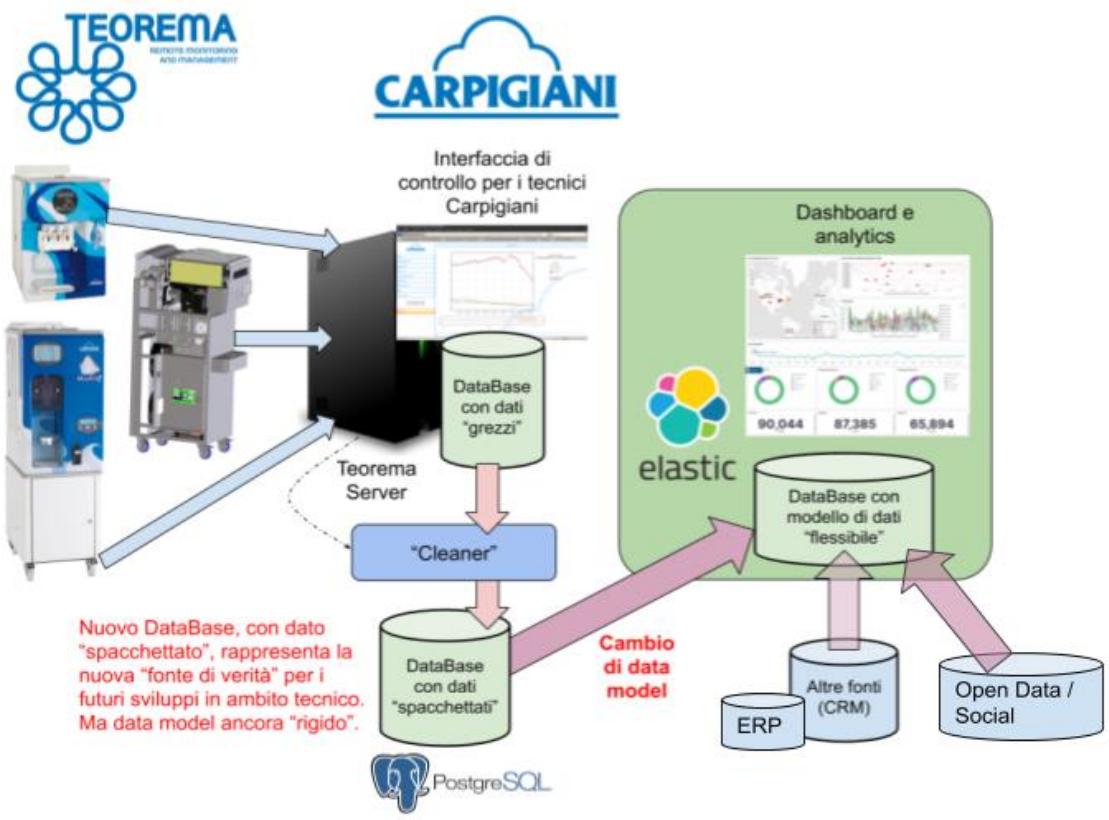

Fig.18. Architettura della piattaforma di analisi Big Data estesa al fine di supportare la realizzazione di servizi "out".

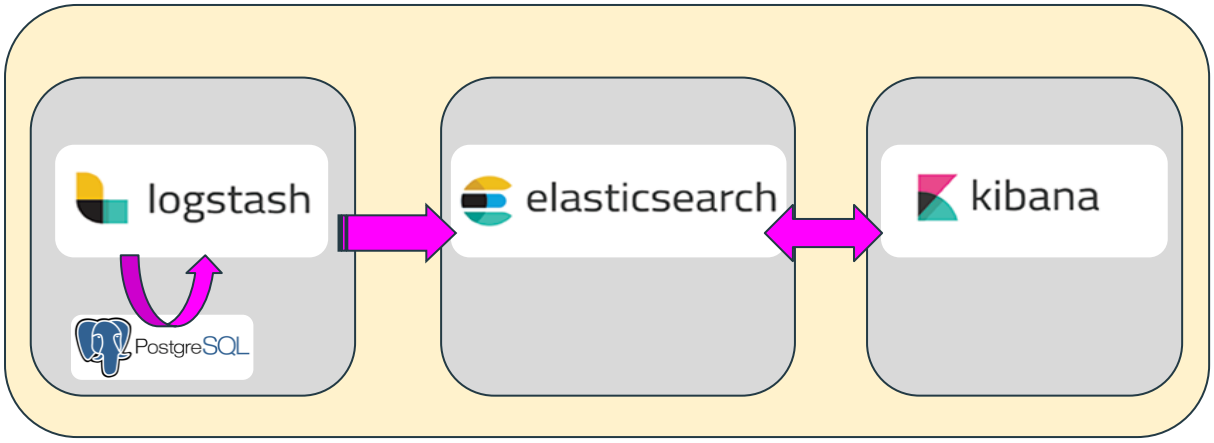

Fig. 19. Deploy dei componenti su piattaforma OpenStack.

Nell'ambito di questo progetto è stato definito un modello di deployment che comprende i componenti PostgreSQL (Data WareHouse) e lo stack ELK (LogStash+Elasticsearch+Kibana) - non si considera il modulo Cleaner, di competenza Carpigiani.

Tali componenti verranno installati in un'architettura distribuita sull'infrastruttura OpenStack gestita dal partner di progetto T3Lab. Questa prevede l'impiego di differenti macchine virtuali (VM) sulle quali saranno installati tutti i componenti della nuova architettura (Figura 19).