



**Politecnico  
di Torino**

**Politecnico di Torino**

Corso di Laurea in  
**Ingegneria della Produzione Industriale e dell'Innovazione Tecnologica**

A.a. 2023/2024

Sessione di Laurea Ottobre 2024

# **Evoluzione delle Architetture dei Sistemi Informativi**

Scenari e Modelli di Interconnessione di Tecnologie Innovative  
in Contesti di Trasformazione Digitale

Relatrice:

Maria Pia Cavatorta

Candidato:

Luca Panero

## Ringraziamenti

Al termine di questo percorso accademico, desidero esprimere la mia sincera gratitudine a tutti coloro che mi hanno sostenuto ed accompagnato durante questi anni.

Innanzitutto, un ringraziamento va alla mia relatrice, Prof.ssa Maria Pia Cavatorta, per i preziosi consigli e per il supporto durante la stesura dell'elaborato.

Ringrazio sentitamente anche la mia tutor di stage, Cécile, per avermi seguito durante il mio periodo di tirocinio, avermi sostenuto nei periodi di difficoltà e riposto grande fiducia per prendere in mano il progetto e lasciarmi tracciare la direzione.

Un ringraziamento va inoltre ai miei superiori, Antoine e Louis, per avermi dato l'opportunità di partecipare a progetti sfidanti e per il loro costante incoraggiamento e fiducia nelle mie capacità.

Desidero anche ringraziare i miei compagni "stagiaires" e colleghi, Sami, Mathis, Hadi, Ryan, Léo, Jean, Paul e Khaled "tonton", con i quali ho condiviso momenti di lavoro intensi e di arricchimento reciproco, ma che sono stati anche complici perfetti per staccare la spina. Con voi ho trovato l'equilibrio tra l'impegno e il divertimento, rendendo l'esperienza di tirocinio ancora più memorabile.

Non posso che esprimere la mia profonda gratitudine alla mia famiglia e ai miei genitori, grazie per il sostegno incondizionato. Nonostante le vostre differenze, mi avete sempre trasmesso valori complementari e mi avete insegnato a vedere il mondo da prospettive diverse, aiutandomi a trarre il meglio da entrambi. Le vostre lezioni di vita mi hanno spinto ad allargare i miei orizzonti e ad aprire la mente, rendendomi la persona che sono oggi.

Un pensiero speciale va ai miei amici di una vita, Alessandro, Giorgio e Guglielmo, che mi sono sempre stati al fianco nonostante la distanza, e con i quali ho condiviso ricordi speciali e con i quali sono cresciuto.

Grazie a tutti i miei compagni di corso, con cui ho condiviso tante esperienze e soddisfazioni. Ad Andrea e Simone, compagni di *campussate*, *ripassoni* e weekend lontani da casa. A Luca e Léo, coinquilini parigini, con i quali mi sono immerso nella ville lumière. A Marta e Sofia, coinquiline nizzarde, con le quali ho condiviso tanti ricordi e tanti alti e bassi.

Grazie a tutti per aver fatto parte di questa importante fase della mia vita e per essermi state vicino, anche da lontano.

## Sommario

Ringraziamenti.....	2
Tabella degli acronimi.....	6
1. Introduzione.....	7
2. Wavestone: storia, il settore della consulenza e le problematiche attuali.....	9
2.1. “The Positive Way”: una cultura aziendale orientata verso un impatto responsabile a 360°	10
2.2. Problematiche attuali.....	11
2.3. La trasformazione digitale: un nuovo paradigma industriale.....	12
2.3.1. Strategie e percorsi per la trasformazione digitale.....	15
2.3.2. Il ruolo dei Sistemi Informativi nel contesto di trasformazione digitale.....	17
2.3.3. Fallimento nella digitalizzazione ed il ruolo della consulenza.....	19
3. L’esperienza di tirocinio.....	23
4. I sistemi informativi aziendali e il ruolo dei MES.....	24
4.1. Un modello di rappresentazione dei SI: la piramide di Anthony.....	27
4.2. Le architetture dei sistemi informativi e il concetto di urbanizzazione.....	29
4.3. Standard e frameworks: ISA-95, IIRA e RAMI 4.0.....	34
4.3.1. ISA-95 (International Society of Automation) e il modello di Purdue.....	36
4.3.2. IIRA (Industrial Internet Reference Architecture).....	37
4.3.3. RAMI 4.0 (Reference Architecture Model Industry 4.0).....	38
4.4. I MES e il loro ruolo nell’ecosistema informativo aziendale.....	39
4.4.1. Le 11 funzionalità dei MES secondo l’ISA-95.....	40
4.4.2. Sfide e problematiche dei MES nel settore industriale.....	41
4.5. La convergenza IT/OT: un cambio di paradigma nella concezione delle architetture dei SI	43
5. Stato dell’arte dei sistemi informativi.....	46
5.1. Mappatura funzionale: le funzionalità corrispondono alle esigenze aziendali.....	46
5.1.1. Criteri per l’assegnazione delle funzionalità di un sistema informativo.....	47
5.2. Mappatura applicativa: softwares, interfacce e basi di dati.....	47
5.2.1. Le comunicazioni tra sistemi informatici in un SI tramite API.....	48
5.2.2. Lo stoccaggio e gestione delle informazioni: le basi di dati.....	52
5.2.3. Mappatura applicativa di un SI industriale e modello di Purdue.....	55
5.3. Mappatura tecnica: le fondamenta dei sistemi informativi.....	55
5.3.1. Servers.....	56
5.3.2. Rete.....	57
5.3.3. Meccanismi di sicurezza.....	58

5.3.4.	Mappatura tecnica di un SI industriale.....	59
6.	Le evoluzioni dei sistemi informativi .....	60
6.1.	Micro-servizi: un cambiamento nell’approccio dello sviluppo delle applicazioni.....	61
6.1.1.	La containerizzazione: Kubernetes e Docker.....	62
6.1.2.	Il data management nelle architetture a microservizi.....	64
6.1.3.	Service Mesh: una gestione avanzata del traffico e della sicurezza in un ambiente distribuito .....	67
6.2.	Un passaggio verso le architetture ad eventi .....	69
6.2.1.	MQTT: un protocollo di comunicazione specifico per il settore manifatturiero.....	71
6.2.2.	L’estensione Sparkplug B per la standardizzazione di MQTT.....	75
6.2.3.	L’UNS (Unified NameSpace): un’unica sorgente di verità nei sistemi informativi ...	77
6.3.	Apache Kafka e le piattaforme di streaming dei dati .....	79
6.4.	MQTT e Kafka: differenze e use case per un utilizzo complementare .....	81
6.5.	Le piattaforme CPS a supporto della sicurezza informatica dei SI .....	83
6.6.	Le piattaforme IIoT a confronto con i sistemi informatici monolitici più diffusi .....	84
6.7.	I data hubs: una soluzione innovativa per la gestione e centralizzazione del dato.....	85
7.	TOM generalizzato di integrazione delle ultime evoluzioni.....	87
7.1.	L’implementazione agile attraverso DevOps e DevSecOps.....	87
7.2.	La gestione del legacy: una sfida cruciale nel processo di trasformazione digitale .....	89
7.3.	Scenari progressivi di integrazione delle evoluzioni.....	89
7.3.1.	L’adozione dei microservizi “cloud-native”.....	90
7.3.2.	L’implementazione di un service mesh per la gestione del traffico .....	91
7.3.3.	Un architettura ibrida per la convergenza IT/OT .....	92
7.3.4.	Passaggio definitivo alle architetture ad eventi.....	93
8.	Conclusione.....	94
9.	Bibliografia .....	96
10.	Sitografia.....	97
11.	Appendice .....	99
11.1.	Mappatura funzionale di un’azienda nel settore manifatturiero .....	99
11.2.	Mappatura funzionale-applicativa della ripartizione delle principali funzionalità dei software di un SI .....	100
11.3.	Mappatura applicativa con soluzioni generaliste e modello di Purdue .....	101
11.4.	Mappatura tecnica.....	102
11.5.	Architettura a microservizi con Kubernetes e Docker.....	103
11.6.	Infrastruttura integrata MQTT-Kafka per le comunicazioni in un’architettura EDA....	104
11.7.	Tappa 1: adozione dei microservizi cloud-native .....	105

11.8.	Tappa 2: integrazione di un Service Mesh per la gestione del traffico est-ovest .....	106
11.9.	Tappa 3: passaggio verso un'architettura ibrida con l'integrazione di un broker MQTT nell'OT	107
11.10.	Tappa 4: passaggio definitivo verso un'architettura EDA .....	108

## Indice delle Figure

Figura 1	Digital Intensity Index in Europa nel 2023. Fonte: Eurostat.....	12
Figura 2	Spesa per tecnologie e servizi di trasformazione digitale a livello mondiale dal 2017 al 2027 (Fonte: Statista).....	13
Figura 3	Obiettivi di trasformazione tecnologica delle aziende nel 2024 e nel 2025. (Fonte: Statista) .....	15
Figura 4	Gerarchia di un sistema azienda .....	25
Figura 5	Il processo di produzione di informazioni.....	25
Figura 6	Rappresentazione della piramide di Anthony.....	27
Figura 7	Assi di integrazione di un sistema informativo .....	30
Figura 8	Modello di Purdue .....	37
Figura 9	Architettura di riferimento RAMI 4.0. Fonte: Reference Architectural Model Industrie 4.0 (RAMI 4.0).....	39
Figura 10	Storia ed evoluzione della convergenza IT/OT .....	44
Figura 11	Rappresentazione del funzionamento di una comunicazione tra due client tramite API ..	49
Figura 12	Il data management con un Data Warehouse.....	54
Figura 13	Il data management con un Data Lake .....	55
Figura 14	Esempio Saga Pattern .....	66
Figura 15	Architettura service mesh Istio .....	68
Figura 16	Modello pubblicazione-sottoscrizione.....	69
Figura 17	Infrastruttura logica industriale con MQTT e Sparkplug B .....	76
Figura 18	Esempio struttura UNS.....	78
Figura 19	Infrastruttura di una piattaforma di streaming di eventi .....	80
Figura 20	Struttura e componenti di un Data Hub .....	85
Figura 21	Ciclo di vita DevOps. (Fonte: italiancoders).....	88
Figura 22	Percorso di trasformazione delle architetture logiche dei SI.....	90

## Indice delle Tabelle

Tabella 1	Confronto dei principali protocolli utilizzati dagli API .....	52
Tabella 2	Confronto MQTT-HTTP dei volumi (bytes) per numero di messaggi inviati .....	74
Tabella 3	Confronto tempo medio di risposta MQTT-HTTP.....	75
Tabella 4	Confronto performance MQTT in base al livello di servizio (QoS).....	75
Tabella 5	Confronto MQTT e Apache Kafka.....	82

## Tabella degli acronimi

<b>SI</b>	Sistema informativo
<b>MES</b>	Manufacturing Execution System
<b>ERP</b>	Enterprise Resource Planning
<b>PLM</b>	Product Lifecycle Management
<b>SCADA</b>	Supervisory Control And Data Acquisition
<b>LIMS</b>	Laboratory Information Management System
<b>EMS</b>	Energy Management System
<b>QMS</b>	Quality Management System
<b>CMMS</b>	Computerized Maintenance Management System
<b>WMS</b>	Warehouse Management System
<b>TMS</b>	Transport Management System
<b>CRM</b>	Customer Relationship Management
<b>PLC</b>	Programmable Logic Controller
<b>IoT</b>	Internet of Things
<b>IIoT</b>	Industrial Internet of Things
<b>TOM</b>	Target Operating Model
<b>ICT</b>	Information and Communication Technologies
<b>DMZ</b>	DeMilitarized Zone
<b>API</b>	Application Programming Interface
<b>SOAP</b>	Simple Object Access Protocol
<b>REST</b>	Representational State Transfer
<b>RPC</b>	Remote Procedure Call
<b>DBMS</b>	DataBase Management Systems
<b>ICS</b>	Industrial Control System
<b>ESG</b>	Environmental, Social and Governance

# 1. Introduzione

La trasformazione digitale è uno dei fenomeni più significativi del nostro tempo, ridefinendo profondamente il modo in cui le aziende operano e competono in un mercato sempre più globalizzato e dinamico. Nell'era dell'Industria 4.0, le tecnologie digitali non sono più una semplice opzione, ma una necessità strategica per garantire l'efficienza, l'innovazione e la sostenibilità delle imprese. La digitalizzazione non riguarda solo l'adozione di nuove tecnologie, ma implica un cambiamento radicale nei processi aziendali, nella gestione delle informazioni e nell'interazione con il mercato (Lin, 2024).

L'obiettivo di questo elaborato è descrivere brevemente l'esperienza di tirocinio svolta presso Wavestone, società di consulenza operante nel settore della trasformazione digitale in Europa, e il progetto di ricerca svolto sull'evoluzione dell'interconnessione dei MES in un sistema informativo aziendale. Questa analisi è stata svolta durante il periodo di tirocinio con l'obiettivo di approfondire un argomento di forte interesse per una società come Wavestone, per forgiare delle convinzioni, creare framework di integrazione e distribuzione di un sistema informativo e, infine, per supportare eventuali offerte commerciali. Il progetto era inizialmente focalizzato sui MES e le loro interazioni con i componenti di un SI; tuttavia, data la vastità della problematica e dalla possibilità di adattare i framework realizzati ad altre tipologie di sistemi informatici, si è deciso di allargare il progetto alle architetture informative in generale, approfondendo maggiormente il contesto di telecomunicazioni nel settore industriale.

L'elaborato è strutturato come segue:

- Nel primo capitolo viene presentata brevemente Wavestone, la sua storia, il settore della consulenza in trasformazione digitale e le problematiche attuali affrontate in questo settore.
- Nel secondo capitolo viene descritto il tirocinio, le attività svolte e il progetto di stage che verrà approfondito nei capitoli successivi
- Il terzo capitolo si propone di inquadrare i sistemi informativi aziendali al giorno d'oggi, esplorando la centralità della gestione delle informazioni e dei sistemi informatici, per andare infine a definire i MES, le loro funzionalità e il loro ruolo nell'ecosistema aziendale. Vengono, inoltre, introdotti il concetto di architettura dei

sistemi informativi, i differenti livelli individuati nella letteratura e negli standard internazionali, e, infine, i suoi componenti.

- Il quarto capitolo propone un'analisi dello stato dell'arte dei sistemi informativi su tre assi principali: funzionale, logico e tecnico. Sulla base di questa analisi vengono definite le principali problematiche dei modelli attualmente più diffusi.
- Il quinto capitolo si propone di approfondire le ultime evoluzioni proposte dal mercato in questo settore e compararle con le tecnologie attualmente più diffuse.
- Infine, il sesto capitolo, presenta quattro scenari di integrazione delle tecnologie individuate nel quinto capitolo. Vengono, quindi, proposti dei framework di integrazione graduale per passare dalle architetture attuali a soluzioni più evolute, scalabili e flessibili.
- L'elaborato si conclude con una riflessione critica sui risultati ottenuti e i vantaggi e le problematiche che le aziende possono trarre e incontrare in caso di utilizzo dei modelli presentati.

In sintesi, l'elaborato offre una panoramica completa e integrata della teoria e della pratica sulla gestione dei sistemi informativi, con alcuni approfondimenti sulla data governance e sulle metodologie di integrazione e distribuzione dei sistemi informatici.



## 2. Wavestone: storia, il settore della consulenza e le problematiche attuali

Wavestone è una società di consulenza internazionale, specializzata nella trasformazione digitale e nel supporto strategico alle organizzazioni in ambito IT, cybersecurity e manageriale. Nasce nel 1990 come Solucom per poi fondersi successivamente con le attività europee di Kurt Salmon dando vita a Wavestone. La sua storia è, in maniera generale, caratterizzata dalla combinazione di esperienze e competenze di diverse società di consulenza, che hanno unito le loro forze per creare un'entità unica e robusta nel panorama globale della consulenza.

- Solucom, si è rapidamente affermata come un attore chiave nel settore della consulenza in tecnologie dell'informazione e della comunicazione. Con una crescita costante e l'acquisizione di altre società di consulenza, Solucom ha ampliato il suo raggio d'azione e le sue competenze, diventando un partner di fiducia per numerose aziende di primo piano.
- Kurt Salmon ha radici che risalgono al 1935 e si è costruita una reputazione solida come consulente strategico per settori specifici, tra cui retail, healthcare e consumer goods. La fusione delle attività europee di Kurt Salmon con Solucom ha portato alla nascita di Wavestone, una realtà che combina la profonda conoscenza tecnologica di Solucom con l'expertise strategica e settoriale di Kurt Salmon.
- Q-PERIOR, la cui acquisizione di Wavestone vede luce nel maggio del 2024 è una società operante nelle regioni est dell'Europa con competenze complementari ed una cultura condivisa.

Nel corso degli anni, Wavestone ha continuato a espandersi sia organicamente che attraverso acquisizioni mirate, rafforzando la sua presenza internazionale e ampliando il proprio portafoglio di servizi.

Wavestone opera su scala globale, con una presenza consolidata in 17 paesi in Europa, Nord America e Asia e più di 5500 collaboratori. La società serve una vasta gamma di settori industriali, tra cui l'automotive, i servizi finanziari, il settore pubblico, l'energia, il retail e le telecomunicazioni. La capacità di Wavestone di operare in molteplici settori deriva dalla sua struttura organizzativa, che si basa su team specializzati in diverse discipline e mercati.

Nel campo della trasformazione digitale, offre consulenza su un'ampia gamma di tematiche, tra cui cybersecurity, l'ottimizzazione dei processi aziendali, la gestione dei rischi, la governance IT, e l'implementazione di tecnologie emergenti come l'intelligenza artificiale e l'Internet of Things (IoT).

Una delle forze di Wavestone è l'indipendenza da società esterne, permettendo di adottare una metodologia pragmatica ed orientata ai risultati, piuttosto che orientata alla vendita di determinati prodotti.

## 2.1. “The Positive Way”: una cultura aziendale orientata verso un impatto responsabile a 360°

Wavestone è unita nell'aver un impatto positivo, dal punto di vista sociale e ambientale. “The Positive Way”, il nome della propria cultura aziendale, si fonda su cinque valori fondamentali:

- **Innovazione Responsabile:** Wavestone crede fermamente nell'innovazione come motore di crescita e progresso, ma con un approccio responsabile. Il "Positive Way" sottolinea l'importanza di adottare soluzioni innovative che non solo migliorino le performance aziendali, ma che abbiano anche un impatto positivo sulla società e sull'ambiente. Questo significa che ogni progetto e ogni iniziativa devono essere valutati non solo in termini di fattibilità economica, ma anche in termini di sostenibilità e responsabilità sociale.
- **Cultura dell'Eccellenza e del Lavoro di Squadra:** Wavestone promuove un ambiente in cui l'eccellenza professionale e la collaborazione sono al centro di ogni progetto. Il "Positive Way" si manifesta attraverso un impegno costante nel fornire consulenza di alta qualità, lavorando a stretto contatto con i clienti per comprendere appieno le loro esigenze e superare le loro aspettative. Il lavoro di squadra è fondamentale, e ogni membro di un team è incoraggiato a contribuire con le proprie competenze e idee per raggiungere gli obiettivi comuni.
- **Benessere e Sviluppo delle Persone:** Al centro del "Positive Way" c'è anche un forte impegno verso il benessere e lo sviluppo dei dipendenti. Wavestone promuove una cultura aziendale che valorizza il benessere fisico e mentale, offrendo un ambiente di lavoro inclusivo e stimolante. L'azienda investe significativamente nella

formazione e nello sviluppo professionale dei propri collaboratori, fornendo opportunità di crescita e carriera in linea con le aspirazioni personali e professionali.

- **Sostenibilità e Responsabilità Sociale:** Wavestone è consapevole del ruolo cruciale che le imprese hanno nel promuovere uno sviluppo sostenibile. Il "Positive Way" si traduce in azioni concrete volte a minimizzare l'impatto ambientale delle attività aziendali e a promuovere pratiche sostenibili. Questo impegno si estende anche alla responsabilità sociale, con iniziative che supportano comunità locali e globali, e progetti che mirano a risolvere problemi sociali attraverso l'uso della tecnologia e dell'innovazione.
- **Orientamento al Cliente:** Il "Positive Way" pone il cliente al centro di tutte le attività. L'azienda si impegna a costruire relazioni di lungo termine basate sulla fiducia, trasparenza e risultati concreti. Questo approccio si riflette nell'attenzione meticolosa ai dettagli e nella personalizzazione delle soluzioni per soddisfare le esigenze specifiche di ogni cliente.

Questa particolare cultura aziendale, in sintesi, è una base concreta che lega insieme l'intera organizzazione.

## 2.2. Problematiche attuali

Nel contesto della consulenza e della trasformazione digitale, Wavestone si confronta con diverse problematiche che attualmente impattano significativamente le aziende a livello globale:

1. **Aumento della Concorrenza:** In un mercato sempre più globalizzato, le aziende devono affrontare una concorrenza crescente non solo a livello locale, ma anche internazionale. La capacità di innovare e di adattarsi rapidamente ai cambiamenti del mercato è essenziale per mantenere una posizione di vantaggio competitivo.
2. **Digitalizzazione:** La digitalizzazione rappresenta sia un'opportunità che una sfida. L'adozione di nuove tecnologie richiede investimenti significativi e una gestione efficace del cambiamento, ma offre anche il potenziale per trasformare radicalmente i modelli di business e migliorare le performance aziendali. Secondo Eurostat circa il 58% delle aziende nel 2023 hanno raggiunto almeno un livello base di intensità digitale. Il "livello di intensità digitale" è misurato dal Digital Intensity Index (DII),

implicando l'utilizzo di almeno 4 di 12 tecnologie selezionate come IA, social media o il cloud computing. In Figura 1 viene rappresentato il DDI in Europa nel 2023.

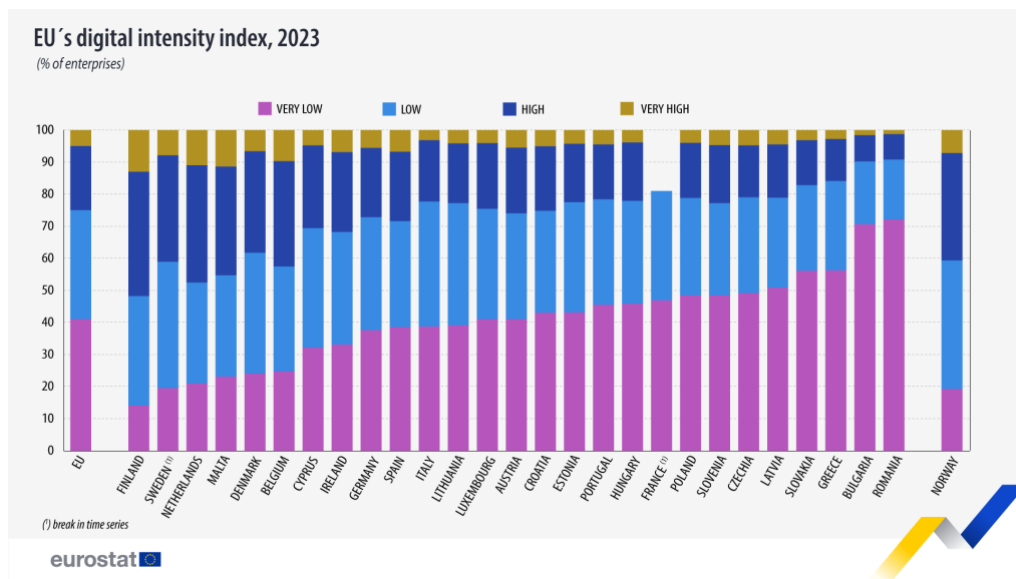


Figura 1 Digital Intensity Index in Europa nel 2023. Fonte: Eurostat

3. **Cambiamento Climatico:** Le crescenti preoccupazioni legate al cambiamento climatico spingono le aziende a adottare pratiche più sostenibili e a ridurre la loro impronta ecologica. Questo comporta l'integrazione di politiche ambientali nei processi produttivi e la ricerca di soluzioni innovative per affrontare le sfide ambientali.

### 2.3. La trasformazione digitale: un nuovo paradigma industriale

Nel contesto attuale, la trasformazione digitale rappresenta uno dei principali fattori di cambiamento per le aziende di ogni settore. Non è solo un cambiamento dal punto di vista tecnologico (Lin, 2024) ma una ridefinizione integrale dei modelli di business, dei processi organizzativi e delle interazioni con i clienti; in altre parole, un cambiamento radicale del modo in cui le imprese operano, competono e creano valore.

Secondo una definizione di una pubblicazione dell'Harvard Business Review "la trasformazione digitale consiste nell'uso della tecnologia per migliorare radicalmente le prestazioni o il raggiungimento di risultati in un'organizzazione." (George Westerman et al., 2014). (Jeanne W. Ross et al., 2017) descrivono la trasformazione digitale come "un percorso evolutivo che richiede alle organizzazioni di sviluppare nuove competenze digitali, di ripensare i propri modelli di business e di implementare tecnologie emergenti con visione

strategica”. Queste affermazioni mettono in evidenza come la trasformazione digitale non sia semplicemente una questione tecnologica, ma una strategia che permea tutti gli aspetti dell'organizzazione, dalle operazioni quotidiane alla visione a lungo termine.

Questo processo di trasformazione è stato accelerato dalla pandemia COVID-19 che ha aumentato notevolmente il ritmo della trasformazione digitale nelle organizzazioni di tutto il mondo nel 2020. Anche se la pandemia è finita, il lavoro da casa nelle organizzazioni a livello globale non solo è rimasto, ma è anche aumentato, incrementando ulteriormente la spinta alla trasformazione digitale. Altre cause che contribuiscono sono la domanda dei clienti e la necessità di essere alla pari con i concorrenti. Nel complesso, l'utilizzo delle tecnologie per la trasformazione digitale rende le organizzazioni più agili nel rispondere ai cambiamenti del mercato e migliora l'innovazione, rendendole così più resilienti.

È prevedibile, infatti, una crescita esponenziale dei budget allocati alla trasformazione digitale nei prossimi anni, fino a toccare i 3.9 miliardi di dollari nel 2027 (Figura 2).

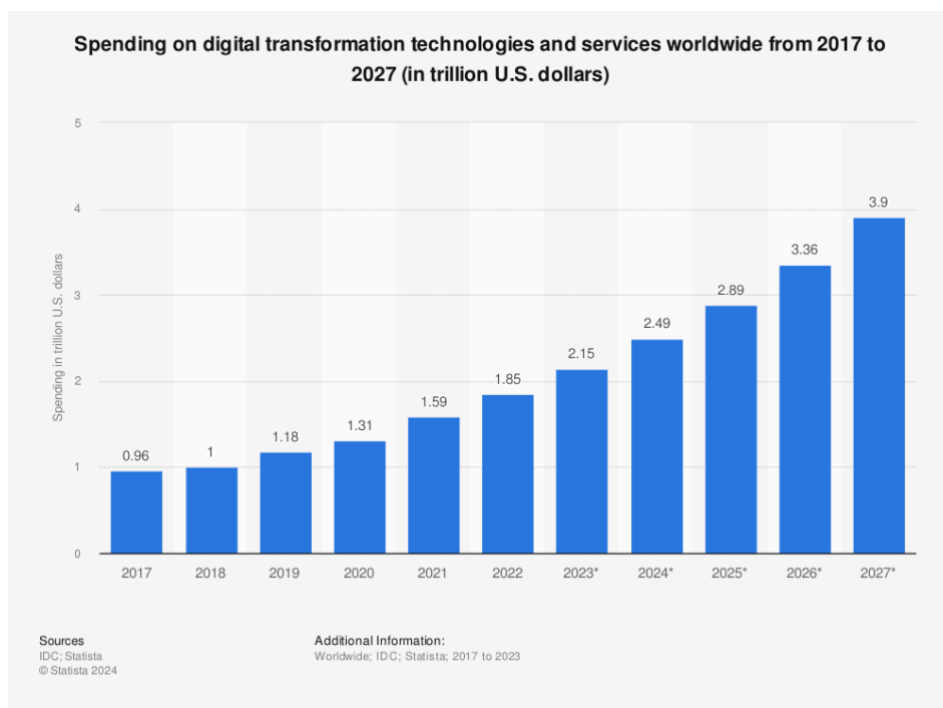


Figura 2 Spesa per tecnologie e servizi di trasformazione digitale a livello mondiale dal 2017 al 2027 (Fonte: Statista)

Questi dati si riferiscono all'adozione di tecnologie digitali per trasformare i processi e i servizi aziendali da non digitali a digitali, come ad esempio l'adozione di piattaforme cloud, l'utilizzo di soluzioni digitali per la comunicazione e la collaborazione, nonché l'automazione dei processi.

La digitalizzazione industriale (Industria 4.0) introduce concetti chiave come l'Internet delle Cose (IoT), l'intelligenza artificiale (AI), la robotica avanzata, la stampa 3D, e la realtà aumentata. Queste tecnologie non solo ottimizzano i processi produttivi, ma creano anche un ambiente interconnesso in cui dati e informazioni fluiscono senza soluzione di continuità tra diverse funzioni aziendali, consentendo decisioni più informate e tempestive.

La trasformazione digitale risulta quindi essere come passaggio obbligatorio per lo sviluppo di un'azienda e per il mantenimento della competitività in un mercato. Le ragioni per affrontare questo percorso sono infatti molteplici:

- Un aumento della concorrenza e quindi una necessità a differenziarsi
- Cambiamenti rapidi e repentini dei mercati, causati da un aumento esponenziale del Time-to-Market dei nuovi prodotti digitali. Vi è quindi una necessità di flessibilità e agilità nei processi produttivi, nella catena di approvvigionamento e nell'R&D
- Necessità di focalizzarsi sugli impatti ambientali e sugli obiettivi ESG
- Un aumento della customizzazione che porta ad una maggiore diversificazione e necessità un passaggio da una visione focalizzata sulla centralità del prodotto a una visione focalizzata sul cliente e i suoi bisogni
- Impatto delle nuove tecnologie come l'intelligenza artificiale, il machine learning, l'Internet of Things (IoT), la blockchain e il cloud computing, necessarie per rimanere competitivi
- Aumento delle minacce cyber e la crescente complessità delle normative, le aziende sono spinte a migliorare la loro infrastruttura di sicurezza e a garantire la conformità alle leggi
- La crescente quantità di dati generati dalle operazioni aziendali e dalle interazioni con i clienti ha reso la gestione dei dati una priorità strategica.

Secondo un questionario condotto da Statista nel 2024 (Figura 3) i principali obiettivi in termini di trasformazione digitale nel 2024 e nel 2025, sono incentrati sul dare la priorità agli obiettivi ESG (48%), aumentare le proprie capacità in cybersicurezza (45%) ed espandere il proprio mercato, raggiungendo nuovi segmenti e aree geografiche (43%).

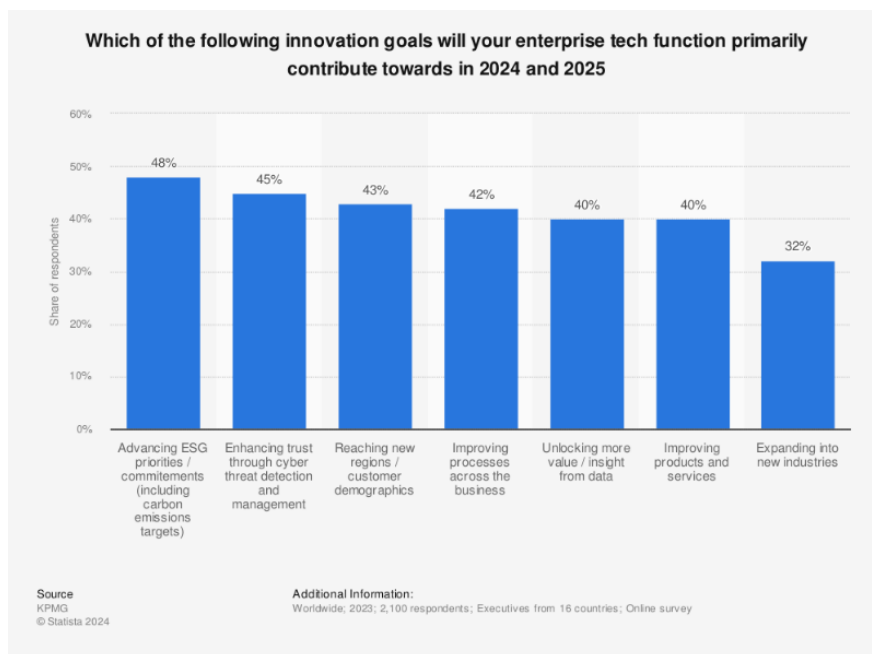


Figura 3 Obiettivi di trasformazione tecnologica delle aziende nel 2024 e nel 2025. (Fonte: Statista)

### 2.3.1. Strategie e percorsi per la trasformazione digitale

I percorsi di trasformazione digitale possono essere differenti e concentrarsi su specifiche aree:

- **Trasformazione dei processi:** implica una revisione e adattamento dei processi e dei flussi di lavoro fondamentali, che in maniera generale sono consolidati da lungo tempo, per allinearsi a nuovi obiettivi aziendali, alle dinamiche della concorrenza e alle crescenti esigenze di personalizzazione dei clienti, frequentemente attraverso l'automazione e digitalizzazione dei processi.
- **Trasformazione del business model:** implica un riadattamento dei principi fondamentali della creazione di valore in un settore specifico alle mutevoli dinamiche di mercato e alle tecnologie emergenti.
- **Trasformazione organizzativa e culturale:** una trasformazione digitale di successo deve trovare le fondamenta nella cultura e nei valori aziendali. Un indebolimento della fiducia interna può influire negativamente su produttività, iniziativa, creatività

e benessere del personale. Un'adozione lenta o resistente, ossia con una grande componente di reticenza al cambiamento dalla parte del personale, delle nuove tecnologie digitali può compromettere il raggiungimento degli obiettivi, la competitività, il fatturato e il valore del marchio. Una trasformazione efficace richiede una collaborazione e comunicazione trasparente dall'alto verso il basso (*change management*) spiegando chiaramente l'impatto e i benefici a lungo termine della trasformazione digitale.

La letteratura identifica uno specifico framework per la trasformazione digitale basata su cinque pilastri operativi fondamentali (Elia et al., 2024):

- **Processi:** riguardano l'insieme delle attività e delle aree organizzative. Permette di definire e strutturare le operazioni necessarie per raggiungere gli obiettivi prefissati e generare valore, attraverso processi di mappatura dei flussi di lavoro esistenti, l'identificazione di aree di miglioramento e l'implementazione di cambiamenti operativi coerenti con gli obiettivi strategici dell'organizzazione. La trasformazione dei processi può coinvolgere l'automazione di attività ripetitive, l'ottimizzazione della catena di approvvigionamento e l'integrazione di nuove tecnologie per migliorare l'efficienza e la produttività complessa.
- **Persone:** sia interne che esterne all'organizzazione che partecipano all'attività di trasformazione. Questo pilastro è critico poiché le persone possono essere sia facilitatori che ostacoli per il successo della trasformazione, a seconda del loro livello di coinvolgimento, impegno, prontezza nelle competenze digitali, autonomia e senso di responsabilità. La gestione delle persone in questo contesto include la formazione per colmare il divario di competenze digitali, l'*empowerment* dei dipendenti attraverso l'assegnazione di nuove responsabilità, e la promozione di una cultura organizzativa che favorisca l'innovazione e l'adozione del cambiamento. Inoltre, il coinvolgimento attivo di tutti gli *stakeholders*, compresi clienti e fornitori, è essenziale per garantire che la trasformazione digitale sia percepita come un'opportunità di crescita piuttosto che come una minaccia.
- **Piattaforme:** comprende le tecnologie digitali che supportano o abilitano direttamente o indirettamente il processo di trasformazione. Questo include i dispositivi digitali, l'infrastruttura tecnologica e i sistemi di gestione dei dati necessari per facilitare l'adozione di nuove modalità operative. Le piattaforme



possono includere software di gestione aziendale, sistemi di analisi dei dati, infrastrutture cloud, e strumenti di collaborazione online. L'importanza di questo pilastro risiede nella sua capacità di integrare nuove tecnologie con quelle esistenti, garantendo che i processi digitali siano scalabili, sicuri e in grado di sostenere le esigenze future dell'organizzazione. Inoltre, la scelta delle piattaforme giuste può determinare il successo o il fallimento di un'iniziativa di trasformazione digitale, poiché influisce direttamente sulla capacità dell'organizzazione di adattarsi rapidamente ai cambiamenti del mercato e alle nuove opportunità.

- **Partners:** comprende i fornitori esterni esperti o i collaboratori che apportano competenze specialistiche, soluzioni tecnologiche e strumenti metodologici necessari per implementare con successo l'iniziativa di trasformazione digitale. I partner possono essere aziende di consulenza, fornitori di tecnologia, o startup innovative che offrono soluzioni complementari. La collaborazione con partner esterni è fondamentale per acquisire competenze e risorse che potrebbero non essere disponibili internamente all'organizzazione, riducendo così i rischi e accelerando il processo di trasformazione. Inoltre, i partner possono offrire nuove prospettive e best practices che arricchiscono il progetto, assicurando che l'organizzazione sia all'avanguardia nelle sue strategie digitali.

Questi quattro pilastri convergono, quindi, per creare un progetto di inquadramento dell'iniziativa di trasformazione digitale.

### 2.3.2. Il ruolo dei Sistemi Informativi nel contesto di trasformazione digitale

I sistemi informativi e l'informatica rappresentano la spina dorsale della trasformazione digitale. Essi forniscono l'infrastruttura necessaria per raccogliere, elaborare e analizzare grandi quantità di dati, che sono il fattore fondante delle decisioni aziendali moderne. La gestione efficiente delle informazioni è essenziale per sfruttare appieno il potenziale delle tecnologie digitali, trasformando i dati grezzi in conoscenze utili e azioni concrete.

Come sottolineato da Thomas H. Davenport & Jeanne G. Harris, 2007, "la capacità di raccogliere, analizzare e agire sui dati è ciò che separa le aziende leader dai loro concorrenti." Questa citazione mette in luce il valore strategico dei dati e dell'analisi nella trasformazione

digitale, evidenziando il ruolo cruciale che i sistemi informativi e l'informatica giocano nel garantire il successo delle iniziative digitali.

Nel contesto industriale, l'integrazione dei Manufacturing Execution Systems (MES) con i sistemi informativi aziendali è un esempio paradigmatico dell'importanza dell'informatica nella trasformazione digitale. I MES sono sistemi critici che monitorano e controllano le operazioni di produzione in tempo reale, fornendo informazioni dettagliate sullo stato dei processi, sulla qualità del prodotto e sull'efficienza delle risorse. Quando interconnessi con gli altri sistemi aziendali, i MES consentono una visibilità completa delle operazioni e una gestione più agile e flessibile della produzione.

In un mondo sempre più interconnesso e guidato dai dati, i sistemi informativi e l'informatica sono al centro della trasformazione digitale, permettendo alle aziende di migliorare le proprie prestazioni e di innovare con successo.

Le tecnologie fondamentali sulle quali la trasformazione digitale al giorno d'oggi si appoggia sono le seguenti:

- **Ecosistemi informativi:** creazione di ecosistemi altamente interconnessi da API e altre tecnologie avanzate.
- **Cloud computing:** accesso a un insieme di risorse informatiche condivise e distribuite su piattaforme pubbliche o private che consentono agli utenti di accedere ai servizi e alle applicazioni da qualsiasi luogo e dispositivo connesso.
- **Internet degli oggetti (IoT):** un ecosistema di oggetti fisici (sensori, software e altre tecnologie) interconnessi che possono raccogliere, trasmettere e scambiare dati tra di loro e con altri sistemi. L'IoT permette, quindi, di monitorare e controllare questi oggetti da remoto, migliorando l'efficienza operativa, automatizzando processi e creando nuove opportunità per servizi personalizzati e innovazioni digitali.
- **Intelligenza artificiale (AI) e *Machine Learning*:** le applicazioni di IA generativa possono rispondere alle richieste del servizio clienti, fornire contenuti su richiesta ed eseguire altre attività in modo automatico e senza l'intervento umano, liberando i dipendenti per lavori di maggior valore.
- **L'automazione:** nel contesto industriale, la robotica dei processi, per eseguire attività ripetitive.

- **DevOps e DevSecOps:** metodologie di organizzazione dei progetti digitali con l'obiettivo di accelerare e aumentare la flessibilità e agilità di questi ultimi combinando e automatizzando il lavoro dei team di sviluppo del software e delle operazioni IT. Queste due metodologie verranno approfondite nel capitolo 7.1.
- **Digital twin:** creazione di copie digitali di prodotti, ambienti di lavoro o linee di produzione per migliorare le prestazioni, permettere una supervisione dematerializzata e migliorare l'efficienza o l'efficacia. Ad esempio, un produttore può creare un gemello digitale della propria officina per trovare il modo di migliorare la posizione dei macchinari per aumentare la produzione o ridurre i problemi di sicurezza.

Nel contesto delle trasformazioni digitali in campo industriale, il concetto di industria 4.0 è ricorrente e centrale. Secondo un recente studio di Wavestone, circa il 60% delle aziende intervistate nel 2023 ha già avviato iniziative di trasformazione digitale e circa il 45% sta già utilizzando l'IoT per monitorare in tempo reale le operazioni produttive (Industry 4.0 Barometer, Wavestone).

### 2.3.3. Fallimento nella digitalizzazione ed il ruolo della consulenza

La trasformazione digitale è una sfida complessa e, come sostenuto da McKinsey o da Harvard Business Review, circa 70% delle iniziative di trasformazioni digitali falliscono o non raggiungono il pieno potenziale. Le ragioni principali di questi fallimenti sono spesso radicate in una serie di fattori interconnessi.

In primo luogo, una definizione inadeguata degli obiettivi (*Target Setting*): una parte significativa del valore potenziale delle trasformazioni viene perso già nelle fasi iniziali, quando gli obiettivi non vengono stabiliti in modo abbastanza ambizioso o realistico. In molti casi, le aziende non riescono a sfruttare appieno il potenziale della trasformazione perché partono con aspettative troppo basse o non basate su un'analisi approfondita. Il ruolo della consulenza in questi casi è la facilitazione nella definizione di obiettivi realistici e ambiziosi, conducendo un'analisi approfondita delle opportunità e dei rischi. Questo processo include valutazioni del mercato, *benchmarking* competitivo e proiezioni finanziarie che permettono all'azienda di fissare target che massimizzano il potenziale della trasformazione. Da un punto di vista più tecnico e generale, nel caso della trasformazione degli ecosistemi informativi aziendali, la definizione di *Target Operating Models*

rappresentano una visione dettagliata e strategica di come un'organizzazione intende operare per raggiungere i propri obiettivi di business. Un TOM definisce la struttura, i processi, le risorse umane, la tecnologia e le metriche necessarie per ottimizzare le operazioni aziendali. Serve come guida per trasformare il business, allineando le capacità operative con la strategia aziendale e assicurando che l'organizzazione possa operare in modo efficace e sostenibile.

La perdita di valore e di considerazione del cambiamento in atto durante l'implementazione è la seconda ragione del fallimento di molte trasformazioni digitali. La fase di implementazione è identificata come il momento in cui si verifica la maggiore perdita di valore. Le aziende spesso non riescono a mantenere la disciplina necessaria per gestire un cambiamento su larga scala, portando a inefficienze e a un'erosione dei benefici attesi. Inoltre, molte trasformazioni si bloccano o rallentano, non riuscendo a raggiungere i risultati previsti entro i tempi stabiliti. In queste fasi, possono risultare utili l'implementazione di metodologie *Agile* per monitorare i progressi gradualmente con vari *Sprint* di lavoro e risolvere rapidamente i problemi, garantendo che la soluzione venga integrata senza interruzioni sui processi esistenti.

In terzo luogo, il mancato consolidamento dei cambiamenti: anche dopo la conclusione formale della trasformazione, il rischio di regressione è alto se le nuove pratiche e processi non vengono integrati ed assimilati nella cultura aziendale e nelle operazioni quotidiane. Le aziende che non riescono a istituzionalizzare i cambiamenti vedono spesso i risultati positivi svanire nel tempo, con un ritorno alle vecchie abitudini. L'adozione di piattaforme digitali aziendali potrebbe aiutare a mantenere il livello di attenzione verso le ultime trasformazioni alto e a introdurre programmi di formazione continua per i dipendenti, inclusi corsi su nuove tecnologie e workshop di *change management*, per garantire l'allineamento culturale e operativo con la nuova strategia digitale.

Un'altra causa di fallimento è la scarsa comunicazione e il mancato coinvolgimento del personale a tutti i livelli dell'organizzazione. I dipendenti spesso non comprendono come i cambiamenti li riguardino personalmente, il che porta a una resistenza passiva o attiva alla trasformazione. Le trasformazioni di maggior successo, al contrario, coinvolgono attivamente i dipendenti attraverso canali di comunicazione diretti e faccia a faccia, come *briefing* e *town hall*. Durante una trasformazione digitale in una grande azienda di servizi

finanziari, la società di consulenza propone servizi di *change management*, ossia l'introduzione di un approccio strutturato per guidare le persone, i team e le organizzazioni attraverso il processo di cambiamento. Questi servizi permettono di coinvolgere la pianificazione, l'implementazione e il monitoraggio delle transizioni all'interno di un'azienda, assicurando che i cambiamenti siano adottati con successo e che gli obiettivi organizzativi siano raggiunti. Il *change management* si concentra sul supporto e sull'accompagnamento delle persone coinvolte, minimizzando la resistenza e migliorando l'accettazione delle nuove pratiche, tecnologie o strategie.

Infine, le trasformazioni che falliscono spesso non riescono a destinare i migliori talenti ai progetti più importanti. Un'allocazione inadeguata delle risorse umane riduce notevolmente le possibilità di successo, poiché le iniziative chiave mancano della leadership e delle competenze necessarie per essere portate a termine con successo. Le aziende di successo, invece, sono quelle che assegnano i loro talenti migliori alle iniziative più critiche, garantendo così la massima attenzione e il massimo sforzo sui progetti che hanno il maggior potenziale di generare valore.

In sintesi, il ruolo della consulenza nella trasformazione digitale è di fondamentale importanza. Le società di consulenza fungono da catalizzatori per l'adozione di nuove tecnologie e modelli operativi, supportando le aziende nel navigare attraverso le complessità del cambiamento digitale. La consulenza offre alle organizzazioni competenze specialistiche, analisi strategiche e soluzioni su misura, che permettono di affrontare le sfide legate alla digitalizzazione con un approccio mirato e sostenibile.

Secondo un rapporto di McKinsey & Company (2018), "le aziende che adottano un approccio strutturato alla trasformazione digitale hanno una probabilità di successo significativamente più alta rispetto a quelle che intraprendono iniziative digitali frammentarie." Questa evidenza empirica sottolinea l'importanza di un piano di trasformazione ben definito e integrato, che le società di consulenza sono in grado di sviluppare e implementare grazie alla loro esperienza multidisciplinare.

La consulenza in questo campo non si limita a fornire soluzioni tecnologiche, ma coinvolge anche la gestione del cambiamento culturale e organizzativo. La digitalizzazione, infatti, richiede un'evoluzione della mentalità aziendale e delle competenze del personale. Le società di consulenza supportano le aziende nel creare una cultura dell'innovazione, promuovendo

la formazione continua e l'adozione di pratiche agili che sono essenziali per mantenere la competitività in un mercato in rapida evoluzione.

### 3. L'esperienza di tirocinio

Il paragrafo seguente ha l'obiettivo di contestualizzare l'esperienza di tirocinio svolta presso Wavestone SA, in qualità di consulente tirocinante all'interno del dipartimento "Manufacture, eNergy & Utilities", nello specifico in collaborazione con il team specializzato in "Operations & Industria 4.0".

Il tirocinio è stato suddiviso in tre missioni principali:

- Un progetto di tirocinio sull'evoluzione dell'interconnessione dei MES con l'ecosistema dei sistemi informativi aziendali. L'obiettivo di questo progetto è l'analisi dei sistemi informativi aziendali attuali, le evoluzioni di questi ultimi, il ruolo dei MES in questo ecosistema e come questi ultimi vengono integrati. In seguito ad un'analisi della letteratura e delle fonti disponibili su internet, sono stati sviluppati framework per la rappresentazione delle architetture dei sistemi informativi *to-be* e delle metodologie di integrazione.
- Un progetto di trasformazione digitale della divisione "Ingegneria e Nuovi Progetti Nucleari" di un'azienda operante nel settore energetico in Francia. Nello specifico, il tirocinante è stato integrato al team "Controllo della produzione" che si occupa della supervisione dei materiali da costruzione delle centrali nucleari e dell'ispezione dei fornitori. Per questo specifico team sono stati implementati due software, un MES ed un PLM, con l'obiettivo di digitalizzare e standardizzare la totalità dei processi, centralizzare le informazioni e i dati, e semplificare le procedure di ispezione, dematerializzando dove possibile.

La fase attuale del progetto è incentrata su attività di re-engineering, creazione di supporti di formazione specifici per i differenti team (ingegneria dei materiali, pianificazione, supervisione e ispezione) e animazione di sessioni di formazione e ateliers.

- Missioni interne in parallelo quali il supporto per la redazione di proposizioni commerciali, la realizzazione di benchmark ed analisi, la realizzazione di formazioni...

I seguenti paragrafi dell'elaborato si concentreranno quindi sul progetto di approfondimento delle tematiche attorno ai sistemi informativi e alla loro architettura.

## 4. I sistemi informativi aziendali e il ruolo dei MES

La gestione delle informazioni all'interno di un'azienda rappresenta, quindi, un aspetto cruciale per il successo organizzativo e competitivo in un contesto economico sempre più complesso e interconnesso. Le informazioni, infatti, sono considerate una risorsa strategica (Marco Tagliavini et al., 2008), al pari del capitale finanziario e delle risorse umane, poiché costituiscono la base su cui si fondano le decisioni aziendali, la pianificazione strategica e l'operatività quotidiana. La capacità di raccogliere, elaborare, memorizzare e distribuire informazioni in modo efficiente ed efficace può determinare la capacità di un'azienda di rispondere rapidamente ai cambiamenti del mercato, di innovare e di mantenere un vantaggio competitivo.

In questo contesto, un sistema informativo aziendale (Enterprise Information Systems - EIS) svolge un ruolo fondamentale: “gestisce, acquisisce, elabora, conserva, produce e scambia le informazioni di interesse di una azienda, ossia quelle informazioni utilizzate per il perseguimento di obiettivi strategici” (Marco Tagliavini et al., 2008). Questi sistemi si inseriscono quindi in un sistema organizzativo: un insieme di risorse e regole per lo svolgimento coordinato delle attività al fine del perseguimento degli scopi.

Si possono avere differenti tipologie di gestione dell'informazione:

- **Gestione implicita:** basata su criteri basati sull'esperienza, non formalizzati ed oggettivi
- **Gestione esplicita non supportata da ICT:** basata su criteri formalizzati e condivisi ma con una gestione manuale
- **Gestione esplicita supportata da ICT:** basata su criteri formalizzati e condivisi e con una gestione informatizzata

Come rappresentato in Figura 4, un sistema informatico è quindi un sottoinsieme dei sistemi informativi:





Figura 4 Gerarchia di un sistema azienda. Fonte: Marco Tagliavini et al. (2008). Sistemi per la gestione dell'informazione

Un sistema informativo è sia un generatore di informazioni sulla base di eventi, sia un consumatore di questi ultimi (Figura 5). Gli eventi possono essere interni o esterni all'azienda.



Figura 5 Il processo di produzione di informazioni. Fonte: Marco Tagliavini et al. (2008). Sistemi per la gestione dell'informazione

L'approccio allo sviluppo di un sistema informativo e quello allo sviluppo di un'organizzazione sono strettamente interdipendenti, poiché i sistemi informativi non sono solo strumenti tecnici, ma rappresentano anche una parte integrale e strategica dell'organizzazione stessa. Questa interdipendenza può essere spiegata attraverso diversi aspetti chiave:

- **Allineamento strategico:** il successo di un sistema informativo dipende fortemente dall'allineamento con la strategia aziendale. Un sistema informativo efficace deve supportare gli obiettivi strategici dell'organizzazione, facilitando l'esecuzione della sua missione e visione. Questo significa che lo sviluppo di un SI non può essere separato dal processo di pianificazione strategica dell'azienda. Le decisioni riguardanti le funzionalità del sistema, la sua architettura e le tecnologie da adottare

devono riflettere le priorità strategiche dell'organizzazione, come l'espansione in nuovi mercati, l'ottimizzazione dei costi o il miglioramento della *customer experience*.

- **Supporto ai processi aziendali:** i sistemi informativi sono progettati per automatizzare, migliorare e integrare i processi aziendali. Pertanto, lo sviluppo di un SI deve essere basato su una profonda comprensione dei processi organizzativi esistenti e su come questi possono essere ottimizzati o reingegnerizzati. Questo richiede un approccio collaborativo tra gli sviluppatori del sistema e i responsabili dei processi aziendali, garantendo che il sistema supporti adeguatamente le attività operative e decisionali. Inoltre, l'introduzione di un nuovo componente del sistema informativo può spesso portare alla necessità di ripensare e ridisegnare i processi aziendali stessi, in un ciclo di miglioramento continuo.
- **Cultura organizzativa e gestione del cambiamento:** lo sviluppo di un sistema informativo comporta inevitabilmente cambiamenti nel modo in cui le persone lavorano all'interno dell'organizzazione. La cultura organizzativa, che riflette i valori, le abitudini e le norme condivise all'interno dell'azienda, gioca un ruolo cruciale nell'adozione e nell'efficacia del sistema stesso. Un SI che non tiene conto della cultura organizzativa può incontrare resistenze e fallire nell'essere accettato dagli utenti. Di conseguenza, lo sviluppo di quest'ultimo deve essere accompagnato da iniziative di gestione del cambiamento che coinvolgano la formazione degli utenti, la comunicazione chiara dei benefici attesi e il coinvolgimento attivo dei dipendenti nei processi decisionali.
- **Integrazione tecnologica e organizzativa:** i sistemi informativi non esistono isolati; devono essere integrati con altre tecnologie e sistemi presenti all'interno dell'organizzazione. Questa integrazione richiede un'attenta pianificazione sia a livello tecnico che organizzativo. Dal punto di vista tecnico, deve essere garantita la compatibilità con i sistemi esistenti e la possibilità di scalare il sistema in futuro. Dal punto di vista organizzativo, l'integrazione richiede la collaborazione tra diverse funzioni aziendali e una *governance* chiara per gestire le interfacce tra i vari sistemi e processi.
- **Risorse e capacità organizzative:** l'efficacia di un sistema informativo dipende anche dalle risorse e dalle capacità dell'organizzazione, inclusi fattori come il capitale

umano, le competenze tecniche e la capacità di gestione dei progetti. Lo sviluppo di un SI richiede competenze specifiche in ambito tecnologico e gestionale, e la disponibilità di queste risorse può influenzare le decisioni relative all'approccio di sviluppo del sistema, come la scelta tra sviluppo interno, outsourcing o adozione di soluzioni cloud.

#### 4.1. Un modello di rappresentazione dei SI: la piramide di Anthony

La rappresentazione delle informazioni scambiate in un sistema azienda può essere fatta attraverso differenti modelli. Occorre, tuttavia, andare a classificare le informazioni scambiate su criteri specifici.

La piramide di Anthony, proposta da Robert N. Anthony negli anni '60, è un modello di rappresentazione di un sistema azienda che descrive i livelli di gestione e di controllo all'interno di un'organizzazione, e la loro interazione con i sistemi informativi. La piramide (Figura 6) suddivide le attività manageriali in tre livelli principali: gestione strategica, gestione tattica e gestione operativa. Ogni livello della piramide corrisponde a un tipo specifico di attività decisionale e di necessità informativa.

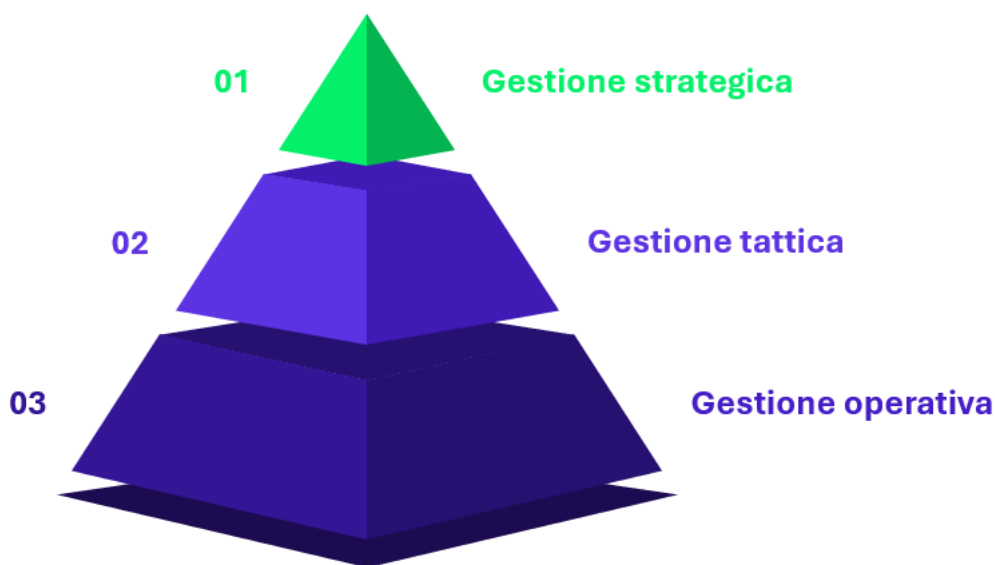


Figura 6 Rappresentazione della piramide di Anthony

- **Gestione Strategica** (Pianificazione strategica): questo livello è occupato dal *top management*, ovvero dagli alti dirigenti e dai responsabili delle decisioni strategiche

a lungo termine, della scelta delle risorse per il loro conseguimento e della definizione delle politiche di comportamento aziendale. In questo livello le informazioni devono essere sintetiche, aggregate e che possano supportare decisioni strategiche a livello quantitativo. Inoltre, vi è una prevalenza di informazioni esterne e non omogenee.

- **Gestione Tattica** (Programmazione e controllo): questo livello include i responsabili delle funzioni aziendali che si occupano della traduzione delle strategie definite dal top management in piani operativi concreti (tattiche), andando a programmare le risorse disponibili al fine di utilizzarle in maniera efficace e controllare il conseguimento degli obiettivi programmati. Le informazioni utilizzate sono dettagliate e specifiche su vari processi aziendali, e devono permettere di monitorare e controllare le performance e apportare eventuali correzioni ai piani operativi. Vi è, quindi, una prevalenza quindi di informazioni interne, dati omogenei e consuntivi ed elaborazioni ripetitive e coerenti nel tempo.
- **Gestione Operativa** (Attività operative): al livello più basso della piramide si trovano i manager operativi, responsabili della gestione quotidiana delle attività aziendali. Questi includono supervisor, capi reparto e altri responsabili delle operazioni quotidiane. Una gestione di questo tipo richiede informazioni dettagliate e in tempo reale per garantire che le operazioni aziendali siano svolte in modo efficiente e conforme ai piani tattici. I sistemi informativi a questo livello, come i sistemi di transazione e i sistemi di controllo operativo, sono fondamentali per monitorare l'efficienza delle operazioni, la qualità della produzione, e la gestione delle risorse. Vi è, quindi, una prevalenza di dati esatti, analitici ed esigenze informative in tempo reale.

Al fine di classificare ogni attività nel corretto livello della piramide si possono individuare differenti caratteristiche (Marco Tagliavini et al., 2008):

- Orizzonte temporale di riferimento
- Orientamento all'esterno
- Discrezionalità
- Ripetitività
- Prevedibilità

- Ruoli organizzativi coinvolti

La piramide di Anthony, quindi, funge da guida per la progettazione e l'implementazione di sistemi informativi che siano allineati con le diverse esigenze decisionali dell'organizzazione, assicurando che ogni livello manageriale disponga delle informazioni appropriate per il suo ruolo specifico.

## 4.2. Le architetture dei sistemi informativi e il concetto di urbanizzazione

Le architetture dei sistemi informativi rappresentano il modo in cui i componenti di un sistema informativo sono organizzati e interconnessi per supportare i processi aziendali e fornire le funzionalità necessarie all'organizzazione. Le architetture dei SI possono essere viste come un quadro strutturale che definisce le modalità di interazione tra hardware, software, dati, reti e persone. Al momento dell'integrazione di un nuovo componente di un sistema informativo si possono individuare tre assi principali (Figura 6) dove si verificano le interazioni con gli altri componenti:

- **Integrazione orizzontale** (Catena del valore): avviene tra le diverse strutture lungo la catena di fornitura, dal fornitore fino al cliente. Un'azienda integrata orizzontalmente si concentra su attività strettamente correlate alle sue competenze e, inoltre, costruisce partnership per supportare l'intera catena del valore. Questo implica quindi un'interazione e flussi di informazioni tra diverse aree funzionali quali gli approvvigionamenti, i trasporti, la produzione, la distribuzione e la logistica e infine il marketing e le vendite. Questa tipologia di integrazione può essere osservata a differenti livelli: ad esempio, nel caso di interazioni tra differenti stabilimenti produttivi si osserva comunque un'integrazione orizzontale. In questo contesto, i dati relativi alla produzione, come ritardi imprevisti, guasti inattesi e livelli di inventario, devono essere trasmessi tra le diverse strutture produttive con un ritardo minimo. Se necessario, gli ordini di produzione possono essere automaticamente riassegnati tra gli stabilimenti. A livelli macro, a un ritardo di consegna di un materiale, uno stabilimento produttivo può rispondere dinamicamente e in tempo reale con una modifica dei piani di produzione, andando ad aumentare l'efficienza di produzione e riducendo l'incidenza degli imprevisti.

- **Integrazione verticale** (piramide di Anthony): avviene tra i differenti livelli dell'azienda individuati nella piramide di Anthony (operativo, tattico e strategico). Ha come obiettivo principale di integrare ogni livello logico di un'organizzazione dal piano di produzione al top management. Questo permette di prendere le decisioni strategiche e tattiche con tempi di reazione ridotti e sulla base di dati che fluiscono in maniera trasparente e diretta tra i vari livelli. Nel caso di un'interazione tra stabilimento produttivo e top management, informazioni quali gli stati di produzione e le previsioni di consegna di un determinato prodotto possono fluire fino al top management per analisi delle performance per stabilimento, e, viceversa, gli ordini di fabbricazione sulla base delle previsioni di vendita possono fluire fino ai siti produttivi.
- **Integrazione trasversale** (Ciclo di vita): si riferisce alla connessione tra i differenti componenti di un SI per quanto riguarda la gestione di un prodotto attraverso tutte le sue fasi del ciclo di vita, dalla progettazione e sviluppo, fino ai processi di servizio al cliente post-vendita e riciclaggio in certi casi specifici. Si possono avere interazioni come uno scambio di informazioni tecniche su un prodotto tra il dipartimento di *R&D* e uno stabilimento produttivo.

In Figura 7, vengono rappresentati i tre assi da un punto di vista applicativo, ossia andando a rappresentare esempi di software aziendali e la loro collocazione nel grafico.

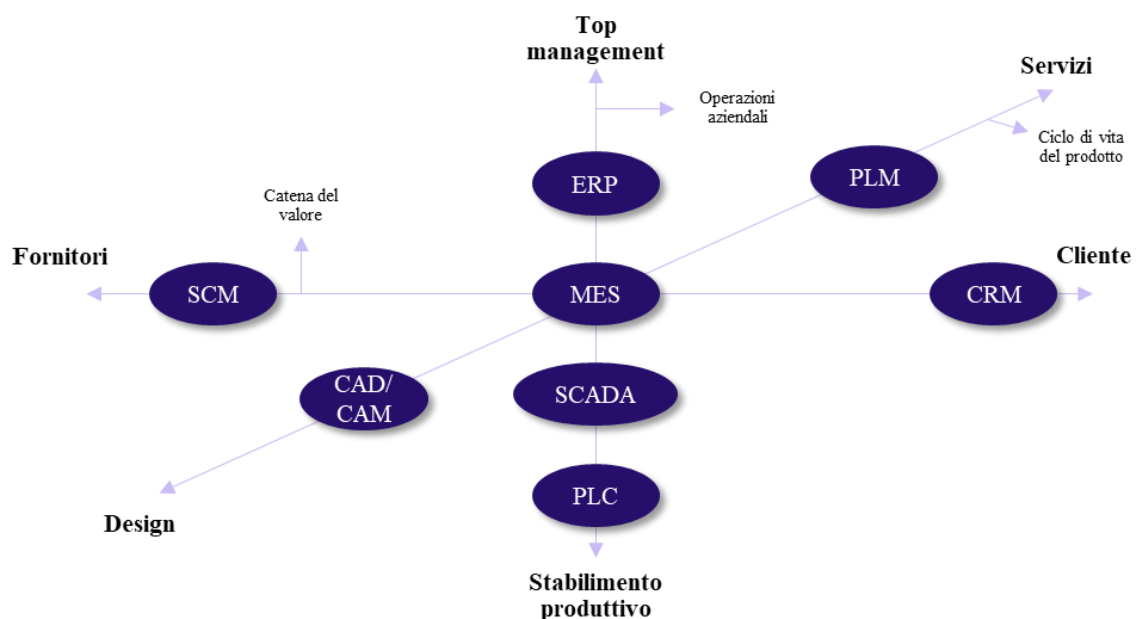


Figura 7 Assi di integrazione di un sistema informativo

Questi software possono essere raggruppati in tre categorie principali (Marco Tagliavini et al., 2008):

- **Executive Information Systems (EIS):** si occupano del supporto alla presa di decisione (ERP) e della gestione del rapporto con il cliente (CRM). Vengono utilizzati per andare a risolvere problemi semi-strutturati attraverso tecniche quali analisi *what-if* o ricerche obiettivo. Vengono inoltre integrati sistemi di intelligenza artificiale o machine learning per analisi previsionali e predittive.
- **Management Information Systems (MIS):** in questa categoria si possono trovare sistemi per la gestione della produzione e della logistica (MES, SCM, TMS), gestione della conoscenza (CAD, CAE, CAM).
- **Electronic Data Processing (EDP):** sistemi a supporto delle attività operative. Si possono trovare sistemi come gli SCADA per la supervisione e controllo delle linee di produzione o i TPS per il supporto ad attività ripetute periodicamente, applicate su grandi moli di dati.

Le architetture di un sistema informativo possono essere mappate su quattro livelli:

- **Di processi:** a questo livello di dettaglio vengono rappresentati i vari processi aziendali, i team e il personale coinvolto. Possono essere degli organigrammi come una rappresentazione delle attività e delle loro interazioni, in termini informativi. In altre parole, una mappa dei processi aziendali è una rappresentazione visiva o un diagramma che mostra le varie capacità di un'organizzazione e il loro contributo agli obiettivi complessivi dell'azienda, evidenziando le interazioni tra di esse. Essa fornisce una visione strutturata delle capacità organizzative, permettendo agli stakeholder di comprendere come le diverse funzioni aziendali collaborano per generare valore. Queste tipologie di mappe hanno generalmente una struttura gerarchica, in cui le capacità di alto livello sono suddivise in sotto-capacità, più dettagliate e specifiche, secondo il livello di precisione desiderato. L'obiettivo di queste mappe sono la classificazione e la prioritizzazione dei processi e l'identificazione dell'interazione tra questi ultimi.
- **Funzionale:** a questo livello si va a definire quali sono gli elementi funzionali del sistema informativo che vanno a supportare i processi del livello superiore. In base al settore aziendale rappresentato vengono costituite delle "isole funzionali" che

vanno a raggruppare funzionalità che similarità applicative. In secondo luogo, si vanno a classificare le tipologie di informazioni scambiate tra le differenti isole: si possono identificare informazioni sul personale, di processo o, ancora, sui prodotti.

- **Applicativa** (logica): dal punto di vista logico, il sistema informativo è formato da componenti informatici che implementano le funzionalità necessarie a supportare il livello superiore. Questa mappa è quindi composta da softwares, basi di dati, middlewares, ossia i softwares che si occupano dell'orchestrazione dei flussi di dati, protocolli di comunicazione e formato dei dati scambiati. Spesso la mappatura applicativa e funzionale viene fusa per andare a visualizzare istantaneamente la ripartizione delle funzionalità per software.
- **Tecnica**: in questa mappa viene infine rappresentato l'ecosistema fisico di un sistema informativo; in altre parole, vengono rappresentati gli elementi fisici all'interno dei quali vengono installati i componenti del livello applicativo. I componenti principali sono le infrastrutture informatiche, quali i server, i data center, i sensori e attuatori ma anche gli elementi di rete come la segmentazione del network, i router e gli switch.

In sintesi, i sistemi informativi rappresentano l'infrastruttura fondamentale necessaria per supportare le operazioni aziendali, gestire i dati e facilitare i processi decisionali del top management. Tuttavia, l'avvento di nuove tecnologie e l'espansione delle attività aziendali, ha portato molte organizzazioni ad affrontare delle problematiche di inefficienza della comunicazione, asimmetrie, frammentazioni ed aumento della complessità dei loro SI. La fluidità dei processi aziendali è quindi ostacolata, rallentando i processi decisionali e aumentando i costi di manutenzione e gestione. Tra le problematiche più ricorrenti si possono individuare:

- **Frammentazione ed isolamento (silos)**: costituisce un metodo di funzionamento con scarsa comunicazione o nulla tra i differenti teams a causa di una frammentazione delle informazioni e delle applicazioni in silos indipendenti. Ogni funzione aziendale ha i propri specifici processi e utilizza differenti strumenti e piattaforme che non comunicano tra di loro in maniera efficace. Questo isolamento genera un'assenza di visione trasversale, inefficienze operative e difficoltà nella condivisione delle informazioni.



- **Aumento della complessità e mancanza di scalabilità:** con l'evolversi e l'espandersi delle organizzazioni si assiste ad un aumento della complessità dei sistemi informativi che può andare ad appesantire l'architettura e renderla più difficile da mantenere e capace a adattarsi ai cambiamenti aziendali, in caso di adozione di nuove tecnologie o funzioni.
- **Incoerenza, ridondanza e duplicazione dei dati:** la frammentazione delle funzioni aziendali e la mancanza di comunicazione tra i team portano inoltre ad una ridondanza dei dati che vengono duplicati su più sistemi, applicazioni e basi di dati, Questo comporta un aumento del rischio di incongruenze ed errori, ma anche un aumento della carica di lavoro del personale che ha più difficoltà a reperire informazioni affidabili ed aggiornate.
- **Problematiche di sicurezza e conformità:** la digitalizzazione della maggior parte dei processi, ha come inconveniente l'aumento dell'esposizione ad attacchi cyber.
- **Cloud computing:** l'adozione di piattaforme di cloud ha portato le aziende a rivedere completamente l'approccio alle architetture dei loro sistemi informatici. L'aggiunta di uno strato (cloud) nell'architettura necessita la riprogettazione delle connessioni e dei flussi con i sistemi che prima erano gestiti internamente e fisicamente (*on-premise*).

Queste problematiche vengono quindi affrontate andando a mappare il sistema informativo sui quattro livelli sopra descritti. Inoltre, l'interesse a mappare un sistema informativo è spesso collegato al concetto di urbanizzazione, ossia il processo di riprogettazione e organizzazione di un sistema in maniera strutturata, coerente e che permetta un'evoluzione e crescita armoniosa a lungo termine, ossia che assicuri una scalabilità adeguata agli obiettivi strategici di un'azienda. In altre parole, la gestione dei flussi e dei componenti di un SI vengono trasformati in modo analogo a come si pianifica e gestisce lo sviluppo urbano.

In questo contesto, l'"urbanizzazione" permette quindi di andare a stabilire un piano strutturale che permetta al sistema informativo di crescere e adattarsi alle nuove esigenze aziendali senza creare inefficienze, ridondanze o complessità eccessive. Si possono andare a identificare alcuni criteri fondamentali da tenere in conto durante il processo di urbanizzazione:

- **Pianificazione strutturale:** consiste nell'andare a definire un TOM, ossia un piano a lungo termine in cui vengono identificate le componenti fondamentali che si vogliono integrare e/o mantenere e le loro interazioni.
- **Modularità e scalabilità:** ossia la capacità di adattarsi e ad evolvere nel corso del tempo
- **Integrazione e interoperabilità:** la capacità del sistema e dei suoi componenti a comunicare con altre piattaforme o applicazioni
- **Governance:** un insieme di regole e politiche che disciplinano la gestione e direzione di un sistema

### 4.3. Standard e frameworks: ISA-95, IIRA e RAMI 4.0

Nella gestione dei sistemi aziendali, l'adozione di standard e frameworks è essenziale per garantire efficienza, sicurezza, interoperabilità e allineamento strategico tra le diverse funzioni aziendali. Questi strumenti offrono linee guida, best practice e modelli che facilitano la progettazione, l'implementazione e la gestione dei sistemi informativi, promuovendo l'integrazione e l'ottimizzazione dei processi aziendali.

In altre parole, garantiscono un linguaggio comune e una metodologia riconosciuta a livello internazionale per gestire e integrare sistemi aziendali. Alcuni degli standard più rilevanti nel settore manifatturiero e dell'industria 4.0 includono:

- **ISA-95:** uno standard utilizzato per integrare i sistemi di controllo della produzione (OT) con i sistemi aziendali (IT), favorendo l'interoperabilità tra questi due mondi. Inoltre, si concentra sull'integrazione di sistemi informatici come i MES, il cui ruolo principale è appunto di collegare i due livelli OT e IT.
- **ISO/IEC 27001:** ampiamente utilizzato per la gestione della sicurezza delle informazioni, definendo i requisiti per stabilire, implementare, mantenere e migliorare un sistema di gestione della sicurezza delle informazioni (ISMS).
- **ISO/IEC 20000:** Riconosciuto come lo standard per la gestione dei servizi IT, fornisce le linee guida per la gestione di servizi IT che siano efficienti, allineati con gli obiettivi aziendali e in grado di migliorare la qualità e la continuità dei servizi forniti.

- **ISO 9001:** Riguarda la gestione della qualità aziendale. Anche se non specifico per i sistemi informativi, questo standard è fondamentale per garantire che i processi aziendali siano ben documentati, gestiti e migliorati costantemente.

In secondo luogo, vi sono dei framework specifici, ossia dei modelli organizzativi utilizzati per gestire, sviluppare e ottimizzare alcuni sistemi aziendali. I principali sono:

- **IIRA:** fornisce linee guida architetturali per integrare sistemi eterogenei, promuovendo interoperabilità, sicurezza e scalabilità nei contesti industriali.
- **RAMI 4.0:** descrive la connessione tra macchinari, processi e sistemi informatici, facilitando l'implementazione di soluzioni di Industria 4.0, dall'integrazione orizzontale e verticale, alla gestione del ciclo di vita dei prodotti.
- **ITIL (Information Technology Infrastructure Library):** largamente utilizzato per la gestione dei servizi IT, offre un insieme di pratiche consigliate per allineare i servizi IT con le esigenze di business, migliorando la qualità, la continuità e il valore dei servizi offerti.
- **COBIT (Control Objectives for Information and Related Technologies):** specifico per la data governance e la gestione dell'IT, consente alle organizzazioni di massimizzare il valore delle proprie risorse IT, riducendo al contempo i rischi associati e garantendo la conformità alle normative.
- **TOGAF (The Open Group Architecture Framework):** propone una metodologia per progettare, pianificare, implementare e gestire l'architettura aziendale in un'organizzazione, favorendo l'allineamento tra business e IT.
- **CMMI (Capability Maturity Model Integration):** È un modello di miglioramento dei processi che aiuta le organizzazioni a ottimizzare le loro operazioni, migliorando la capacità di gestire progetti complessi e sviluppare software di alta qualità.

Gli standard e i framework sono vantaggiosi per le aziende in quanto promuovono una gestione coerente e formalizzata dei sistemi informativi, dai livelli IT a quelli OT, andando a ridurre eventuali rischi operativi e migliorandone le performance. Inoltre, in alcuni settori, le normative internazionali possono andare a richiedere l'adozione e l'implementazione di standard riconosciuti. Infine, standard e frameworks permettono alle organizzazioni di gestire le complessità dei sistemi, supportando una maggiore agilità ai cambiamenti e migliorando i processi decisionali.

### 4.3.1. ISA-95 (International Society of Automation) e il modello di Purdue

Come anticipato, l'ISA-95 è uno standard internazionale particolarmente utilizzato nel settore manifatturiero e industriale per standardizzare e migliorare l'integrazione tra sistemi di automazione e sistemi informativi aziendali. Gli obiettivi principali di questo standard sono:

- Fornire un linguaggio comune per comunicare tra sistemi di automazione, principalmente dedicati alle operazioni in stabilimento, e sistemi gestionali.
- Ridurre i costi e i rischi dati dall'evoluzione e aumento della complessità del sistema informativo aziendale.
- Aumentare la flessibilità e l'interoperabilità per facilitare l'implementazione di nuovi sistemi o l'aggiornamento di sistemi esistenti (*legacy*).
- Creare una gerarchia chiara per la gestione dei dati e delle informazioni, andando a segmentare il sistema in differenti livelli funzionali isolati.
- Migliorare l'efficienza operativa attraverso una suddivisione logica dei compiti.

Il modello architetturale dell'ISA-95 si basa sul modello di Purdue (Figura 8), ossia una struttura di riferimento per la gestione e l'integrazione dei sistemi di automazione industriale che suddivide un SI in cinque livelli distinti:

- **Livello 0 – Dispositivi fisici:** Include sensori e attuatori che interagiscono direttamente con l'ambiente fisico.
- **Livello 1 – Controllo basico:** Comprende sistemi di controllo come PLC o RTU che regolano il funzionamento delle macchine e i processi operativi.
- **Livello 2 – Controllo della supervisione:** Utilizza sistemi SCADA o DCS per monitorare e supervisionare i processi.
- **Livello 3 – Gestione delle operazioni:** Comprende i sistemi MES che coordinano e gestiscono la produzione, ottimizzando i processi e garantendo un'esecuzione efficiente, ma anche altri sistemi di gestione delle capacità o delle performance produttive.
- **Livello 3.5 - DMZ:** consente una segmentazione tra l'OT e l'IT da un punto di vista informatico, riducendo i rischi legati ad attacchi informatici o accessi non autorizzati.

In altre parole, sfrutta server intermedi, proxy, o gateway, per filtrare le richieste e impedire accessi diretti alla rete OT e, viceversa, alla rete IT.

- **Livello 4 – Gestione aziendale:** Gestione strategica attraverso sistemi ERP, che includono la pianificazione delle risorse, la logistica, la finanza e le risorse umane.

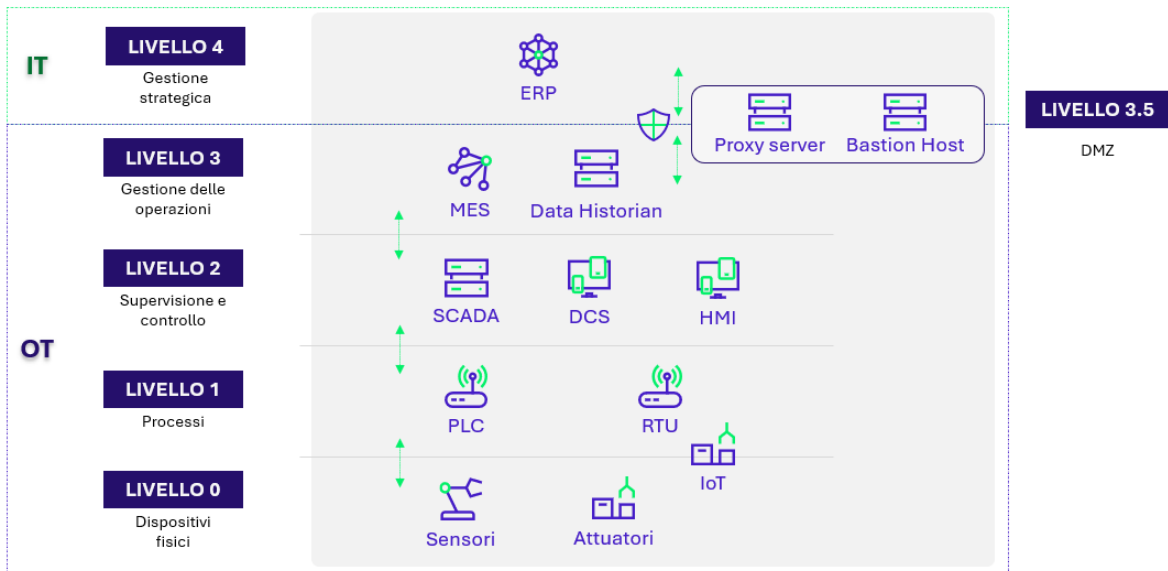


Figura 8 Modello di Purdue

Nella pratica, l'ISA-95 utilizza i concetti del modello di Purdue per definire standard comuni per lo scambio di dati tra livelli, garantendo che le informazioni fluiscano senza soluzione di continuità dalle operazioni di fabbrica fino alla pianificazione aziendale. È utilizzato quindi, principalmente, per organizzare i livelli di controllo e gestione all'interno di un'azienda, dal livello fisico delle macchine fino al livello di gestione aziendale. Definisce le risorse gestite, dai materiali al personale fino alle attrezzature, macchinari e informazioni, e specifica i flussi di dati scambiati tra i differenti sistemi definendo un modello standardizzato per garantire coerenza, uniformità e interoperabilità.

#### 4.3.2. IIRA (Industrial Internet Reference Architecture)

L'IIRA è un framework sviluppato dall'Industrial Internet Consortium (IIC) per fornire una guida alla progettazione di sistemi industriali interconnessi e smart nell'era dell'Industrial Internet of Things (IIoT). IIRA “specifica gli aspetti architettonici, i costrutti e gli approcci per aiutare lo sviluppo, la documentazione e la comunicazione dei sistemi IIoT. L'architettura di riferimento utilizza un vocabolario comune e un framework basato su standard per descrivere i punti di vista di business, utilizzo, funzionalità e implementazione”(The

Industrial Internet Reference Architecture). Viene quindi promossa un'interoperabilità tra i componenti di un ecosistema aziendale e garantita la sicurezza, la scalabilità e la resilienza richiesta dai sistemi industriali moderni che vanno ad integrare dispositivi intelligenti.

I principi e la struttura dell'architettura di riferimento riprendono le informazioni sopra citate sull'architettura dei sistemi informativi in generale andando ad individuare quattro “punti di vista”:

- **Aziendale:** specifica gli interessi e gli obiettivi degli stakeholder aziendali, la loro visione, i loro valori e i loro obiettivi nello stabilire un sistema IIoT nel contesto aziendale e normativo. Vengono quindi definiti i requisiti di business ed i benefici dell'implementazione di un'architettura di questo tipo.
- **D'uso:** viene rappresentato come una sequenza di attività che forniscono le funzionalità ed i servizi che il sistema deve offrire al fine di raggiungere le capacità fondamentali richieste dal modello organizzativo.
- **Funzionale:** si focalizza sui componenti funzionali di un sistema IIoT, andando a definire la loro struttura, le interfacce e i flussi e interazioni tra di essi e con gli elementi esterni dell'ambiente.
- **Implementativa:** va a rappresentare le tecnologie necessarie per implementare i componenti individuati nella vista funzionale andando a coordinare tramite le attività della vista d'uso ed andando a supportare le capacità della vista aziendale.

IIRA è cruciale per l'implementazione di ecosistemi industriali interconnessi, permettendo alle aziende di integrare e gestire in modo sicuro e scalabile, macchinari intelligenti e sistemi complessi.

#### 4.3.3. RAMI 4.0 (Reference Architecture Model Industry 4.0)

Il RAMI 4.0 è un modello di riferimento architetturale sviluppato dalla “Platform Industrie 4.0” per supportare l'implementazione dell'Industria 4.0.

La struttura del modello è tridimensionale e composta da 6 livelli principali (Figura 9): i processi aziendali dell'organizzazione; le funzioni degli *asset*; i dati richiesti; l'accesso alle informazioni; la transizione da processi reali e fisici al digitale, e, infine, il movimento di beni reali come oggetti fisici.

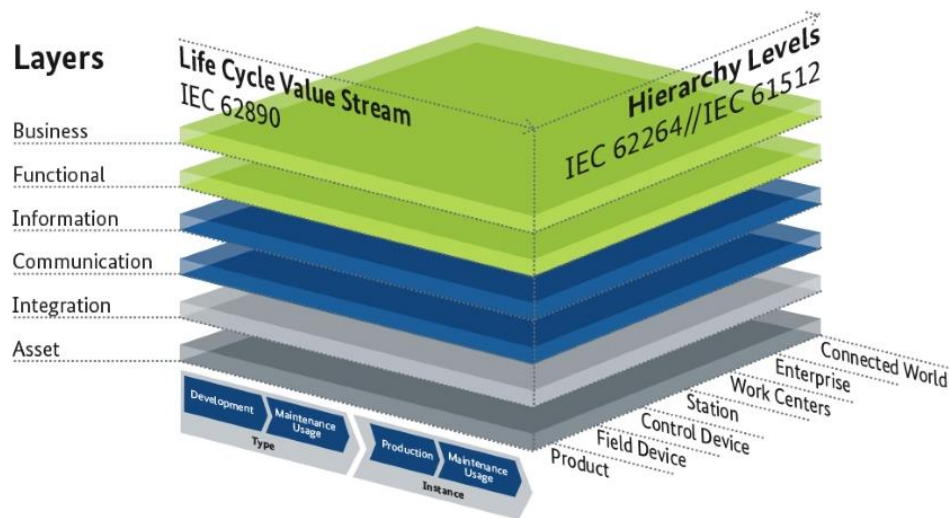


Figura 9 Architettura di riferimento RAMI 4.0. Fonte: Reference Architectural Model Industrie 4.0 (RAMI 4.0)

Vi sono, inoltre due ulteriori dimensioni:

- **Gerarchia della produzione:** suddivide i livelli di produzione, dal prodotto stesso fino ai sistemi gestionali, secondo lo standard IEC 62264/ISA-95.
- **Ciclo di vita del prodotto:** integra l'intero ciclo di vita del prodotto, dalla fase di progettazione, produzione e manutenzione fino alla dismissione.

RAMI 4.0 è fondamentale per le aziende che desiderano implementare sistemi produttivi digitalizzati e interconnessi, promuovendo la scalabilità e l'integrazione delle tecnologie emergenti nell'Industria 4.0.

#### 4.4. I MES e il loro ruolo nell'ecosistema informativo aziendale

La problematica dell'elaborato si fonda sulle difficoltà legate ai MES, al loro ruolo e le loro modalità di interconnessione in un sistema informativo. I capitoli successivi si concentreranno quindi, nella descrizione di questa tipologia di software e dell'individuazione ed analisi delle complessità in seguito alle ultime evoluzioni, per poi andare a fare una panoramica a livello più generale sugli altri componenti dei sistemi

informativi. Questo permette di definire le basi concettuali per presentare, in seguito, i framework ed i modelli di trasformazione dei SI individuati.

Come anticipato precedentemente, i MES sono dei software fondamentali per la gestione e l'ottimizzazione delle attività industriali. Con l'avvento dell'industria 4.0 il loro ruolo è diventato essenziale per la digitalizzazione e l'automazione delle aziende del settore manifatturiero, grazie alla loro capacità di fornire una visione centralizzata e in tempo reale dell'intero ciclo produttivo, andando ad integrare direttamente i dispositivi intelligenti sul terreno. Inoltre, offrono la possibilità di integrarsi con altri sistemi dell'ecosistema informativo aziendale, creando quindi un collegamento diretto tra gli stabilimenti produttivi e le altre funzioni.

#### 4.4.1. Le 11 funzionalità dei MES secondo l'ISA-95

L'ISA-95 identifica 11 funzionalità principali:

- **Raccolta e acquisizione dati:** fornisce interfacce per la raccolta dei dati in tempo reale sulle apparecchiature dell'azienda a livello di controllo (PLC, SCADA, ecc.), o tramite lettura manuale da parte degli operatori.
- **Pianificazione e schedulazione:** la programmazione delle operazioni definisce la sequenza di operazioni ritenuta ottimale. La gestione degli ordini di fabbricazione gestisce il flusso di produzione dei lotti e assicura che le tempistiche vengano rispettate.
- **Gestione dei processi:** assicura il controllo della produzione con la correzione e il miglioramento delle attività tramite un approccio SPC (Statistical Process Control). Permette, inoltre, di controllare e monitorare i parametri di produzione, la sequenza delle attività svolte e il personale, da un punto di vista, ad esempio, di competenze richieste per una particolare operazione. A questo modulo funzionale possono essere accostati sistemi che offrono la capacità di fornire *digital twins*, ossia, rappresentazioni digitali in tempo reale dell'ambiente di produzione.
- **Controllo della qualità:** garantisce la registrazione e la tracciabilità delle informazioni relative allo sviluppo del prodotto, il monitoraggio delle azioni correttive e la capitalizzazione delle conoscenze.
- **Gestione dei materiali:** comporta la gestione in tempo reale delle risorse produttive, tra cui il personale, le attrezzature, i materiali e qualsiasi altra risorsa necessaria per



la produzione. Per ottimizzarne l'uso e promuovere una buona pratica di produzione, le risorse possono o devono essere riservate e rese disponibili nella giusta quantità, qualità e al momento giusto.

- **Gestione della manutenzione:** controlla le attività di manutenzione curativa e preventiva, nonché le attività di miglioramento. La gestione della manutenzione delle apparecchiature è anche un prerequisito per la certificazione ISO 9001.
- **Analisi e monitoraggio delle performance:** monitora i vari indicatori di performance delle operazioni di produzione (tasso di utilizzo, tempo di ciclo, OEE, MTTF, MTBF, tasso di resa dei materiali, tasso di produttività, elementi per il calcolo della stima dei costi, ecc.). Fornisce tutti i report, le *dashboards* e gli indicatori necessari per gestire l'attività e supportare la presa di decisioni in breve tempo.
- **Gestione delle risorse umane:** consente di monitorare la disponibilità e l'assegnazione del personale per funzione e per disponibilità, registrando i compiti e il tempo impiegato. Può integrare, ad esempio, la gestione e il controllo degli accessi a un sito produttivo o ad una particolare area di uno stabilimento.
- **Tracciabilità e genealogia del prodotto:** traccia i prodotti in tempo reale per mantenere uno storico completo dei componenti utilizzati e delle condizioni di produzione di ogni prodotto finito.
- **Documentazione e conformità:** consente la gestione dei documenti relativi a prodotti, processi, progettazione e OF, e talvolta a condizioni di lavoro e certificazioni. Può fornire, ad esempio, i documenti sulla fornitura di gamme di produzione, sull'organizzazione, su procedure, istruzioni, avvisi tecnici, modalità operative, ecc.
- **Monitoraggio dei flussi e delle operazioni su prodotti e lotti:** monitora le gamme di produzione, tenendo conto degli imprevisti di produzione e dei percorsi delle varie linee o macchine. Questa funzione consente di ottimizzare le scorte e di tenere conto delle attività non pianificate (manutenzione, attesa, ecc.).

#### 4.4.2. Sfide e problematiche dei MES nel settore industriale

L'implementazione ed integrazione di un MES in un sistema informativo può presentare molteplici problematiche, legate a fattori tecnici, organizzativi e operativi:

- **Ruolo e sovrapposizione delle funzionalità:** in primo luogo, il ruolo dei MES viene spesso frainteso dalle aziende in quanto propone delle funzionalità che possono essere confuse o che si possono sovrapporre a quelle degli ERP o degli SCADA, ad esempio. Questo può portare a confusione riguardo alle responsabilità del sistema, a una duplicazione dei processi e dei dati, e una maggiore complessità nella gestione di questi ultimi. La chiara definizione dei confini funzionali e l'integrazione efficace dei sistemi sono fondamentali per evitare inefficienze e garantire che ogni sistema possa operare al massimo delle sue capacità senza sovrapporsi inutilmente agli altri.
- **Industria 4.0:** l'avvento dell'Industria 4.0 ha posto nuove sfide ai MES, spingendoli a integrarsi con tecnologie avanzate come l'IoT, l'intelligenza artificiale (AI), e il machine learning per supportare una produzione più intelligente e automatizzata. Questa transizione richiede che i MES non solo raccolgano e analizzino grandi volumi di dati in tempo reale, ma che siano anche capaci di adattarsi e rispondere in modo proattivo ai cambiamenti dinamici nel processo produttivo.
- **Modularità:** inizialmente intesi come dei blocchi “monolitici”, i MES, come altri sistemi informatici, sono di fronte ad una situazione che presenta sia opportunità che sfide. I sistemi modulari offrono flessibilità e permettono alle aziende di adattare i MES alle specifiche esigenze di produzione, aggiungendo o rimuovendo moduli a seconda delle necessità. Tuttavia, questo passaggio richiede un'attenta gestione del design del sistema per garantire che tutti i moduli siano compatibili e che l'integrazione tra di essi sia fluida. La modularità può anche introdurre complessità nell'integrazione dei moduli, specialmente se provengono da diversi fornitori. Il concetto di microservizi che pone le sue basi sulla modularità applicativa, verrà approfondito nel capitolo 6.1.
- **Interconnessione:** l'interconnessione dei MES con altri sistemi, sia interni che esterni, è fondamentale per una gestione efficace della produzione. Tuttavia, la molteplicità di standard e protocolli di comunicazione può rendere questa interconnessione una sfida significativa. Problematiche di questo tipo possono portare a interruzioni nel flusso di dati, ritardi nella produzione e potenziali errori operativi. È essenziale stabilire protocolli chiari e affidabili per la comunicazione e l'interfaccia dei sistemi per superare queste difficoltà. Data l'introduzione di tecnologie sempre più complesse e interconnesse nel livello OT e a causa delle

limitazioni di rete negli stabilimenti, la sfida aggiuntiva è data dalle performance di latenza e di rapidità di trasferimento dei messaggi.

#### 4.5. La convergenza IT/OT: un cambio di paradigma nella concezione delle architetture dei SI

Come anticipato nei paragrafi precedenti, la convergenza IT/OT è un fenomeno emergente che riflette l'evoluzione tecnologica e le sfide correlate alla segmentazione estrema, come delineato dal modello di Purdue. Tuttavia, con l'avanzare delle tecnologie e l'integrazione sempre più profonda tra IT e OT, le rigide distinzioni dei livelli rappresentati nel modello di Purdue sono state messe in discussione. Inoltre, l'obsolescenza dei sistemi industriali è uno dei punti di "cedimento" del sistema. Infatti, negli ultimi anni si possono riscontrare sempre più casi di attacchi informatici che prendono di mira i componenti OT per risalire alle informazioni confidenziali e più critiche dell'IT. In sintesi, le problematiche principali sono le seguenti:

- **Sicurezza informatica:** la sovraesposizione ad eventuali attacchi informatici e i differenti criteri di prioritizzazione della sicurezza dei dati tra IT e OT, portano a punti di cedimento sempre più numerosi nei SI. Le politiche di data governance di questi due livelli, infatti, si differenziano per obiettivi e parametri di controllo: il primo si focalizza sulla confidenzialità dei dati, mentre il secondo sulla disponibilità et integrità dei dati.
- **Differenze organizzative e culturali:** storicamente, i team IT e OT hanno operato in silos, ognuno con il proprio insieme di pratiche e tecnologie. Superare questa divisione richiede una visione unificata e un approccio collaborativo allo sviluppo della strategia.
- **Gestione dei dati:** IT e OT si differenziano anche per i volumi e la varietà di dati trattati.
- **Complessità tecnologiche:** come evidenziato in precedenza, i sistemi IT sono generalmente più avanzati in termini tecnologici in confronto ai sistemi OT in cui l'obsolescenza è una problematica principale.

- **Interoperabilità:** parallelamente all'eterogeneità dei sistemi IT e OT, si ha anche delle differenze in termini di protocolli di comunicazione che ostacolano l'interoperabilità tra i vari componenti del SI.

Le recenti innovazioni tecnologiche come il cloud computing, l'Industrial Internet of Things (IIoT) e le reti 5G hanno quindi spinto le organizzazioni a ripensare questo modello. L'architettura originale del modello di Purdue era ideale per isolare efficacemente i sistemi OT dai potenziali rischi di sicurezza provenienti da reti IT più aperte. Tuttavia, la crescente necessità di flussi di dati bidirezionali per supportare l'automazione avanzata e la raccolta di dati in tempo reale, ha reso questa separazione meno pratica e talvolta controproducente. La convergenza IT/OT ha quindi iniziato a enfatizzare l'integrazione sicura ed efficiente tra queste due sfere, spostando il focus da una separazione rigida a un'integrazione che favorisce la resilienza e la reattività senza compromettere la sicurezza.

La DMZ, una volta pensata come un punto di separazione forte, è ora vista come un punto di controllo critico per moderare i flussi di dati e garantire la sicurezza mentre si facilita la necessaria interazione tra IT e OT. In Figura 10 viene rappresentata la storia e l'evoluzione dei due livelli fino a giungere alla loro convergenza.

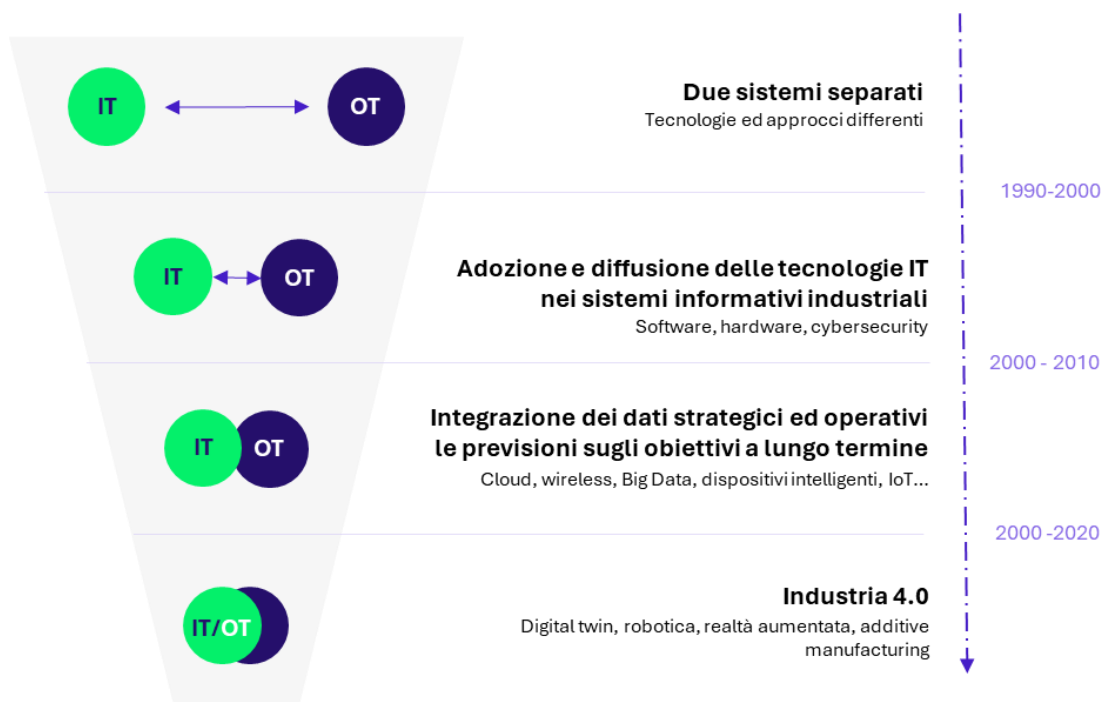


Figura 10 Storia ed evoluzione della convergenza IT/OT

Nonostante le sfide, il modello di Purdue continua a essere un riferimento utile per l'analisi e la pianificazione della sicurezza cibernetica nelle architetture informative, fornendo un quadro per comprendere e mitigare i rischi in un ambiente industriale sempre più connesso e automatizzato.

Nei successivi paragrafi viene presentato uno stato dell'arte delle architetture dei sistemi informativi, andando a suddividere l'analisi su tre punti di vista principali: funzionale, applicativo e tecnico. L'obiettivo è quello di individuare i componenti critici e responsabili delle problematiche sopra descritte. Viene, in seguito, proposta un'analisi delle principali evoluzioni in campo tecnologico e di come queste ultime impattino i sistemi informativi attuali e si propongano di risolvere le problematiche individuate. L'elaborato si conclude con la presentazione di quattro scenari di architetture applicabili gradualmente per fare evolvere il SI e andare a migliorare le performance, la flessibilità e la scalabilità.

## 5. Stato dell'arte dei sistemi informativi

Lo stato dell'arte dei sistemi informativi è stato realizzato andando a scomporre le architetture SI più ricorrenti nei quattro livelli ed andandoli ad analizzare da un punto di vista funzionale, applicativo e tecnico. Le questioni poste per ciascuno dei livelli sono le seguenti:

- **Q1:** *I softwares di esecuzione proposti sul mercato al giorno d'oggi come rispondono ai bisogni e alle funzionalità richieste dalle aziende nell'industria manifatturiera?*
- **Q2:** *Quali sono le soluzioni attualmente più utilizzate per interfacciare e interconnettere questi softwares all'interno di un sistema informativo industriale?*
- **Q3:** *Quali sono le infrastrutture attualmente utilizzate in un SI industriale?*

Per ognuno dei livelli e delle domande corrispondenti sono stati individuati ed analizzati i componenti e il loro funzionamento, andando, infine, a realizzare delle mappature.

### 5.1. Mappatura funzionale: le funzionalità corrispondono alle esigenze aziendali

Il livello funzionale, come suggerito da alcuni framework e standard approfonditi precedentemente, è un'astrazione del SI che permette di andare a strutturare e recensire le funzioni di un ecosistema informativo aziendale in blocchi, e le loro dipendenze. L'obiettivo è quindi quello di supportare il livello superiore, dei processi. Si possono realizzare mappature a differenti livelli di precisione. In questo caso, è stata realizzata una mappatura andando a prendere come riferimento il livello di granularità delle funzionalità definite dallo standard ISA-95 per i MES. Generalmente, le mappature di questo tipo vengono realizzate in maniera gerarchica, identificando delle funzionalità di alto livello per poi decomporle in sotto-capacità più dettagliate. In Appendice 1, viene rappresentata la mappatura funzionale di un'azienda nel settore manifatturiero, realizzata identificando le funzionalità lungo la catena del valore, dal fornitore al cliente. Queste funzionalità possono essere ritrovate nella maggior parte dei processi aziendali di organizzazioni operanti in un settore industriale. Si può, inoltre, osservare come alcune isole funzionali siano trasversali e impattanti a più livelli come la gestione dell'inventario, tutte le operazioni di logistica e trasporto, di manutenzione e qualità. L'isola funzionale strategica, di pianificazione e supporto anch'essa è trasversale

ma dal punto di vista verticale: va, infatti, intesa a livello più alto nella gerarchia funzionale e con dipendenze e interazioni con tutte le isole sottostanti.

Nella maggior parte dei casi le mappature funzionali vengono realizzate all'inizio di una missione di trasformazione, digitale o no, di un'organizzazione per osservare la ripartizione delle attività e individuare visualmente eventuali problematiche di sovrapposizione e duplicazione delle operazioni.

Nel caso delle trasformazioni digitali viene spesso orientata verso una visione applicativa, andando quindi ad assegnare ogni funzionalità ad uno specifico software. Gli obiettivi sono i medesimi. In Appendice 2, viene rappresentato un esempio di mappatura funzionale-applicativa focalizzata sui MES e le funzionalità condivise con altri sistemi informatici simili. Si può osservare come gran parte delle funzionalità possono essere coperte da altri SI, rendendo complicata la scelta della soluzione da adottare dalle aziende intraprendenti un processo di trasformazione digitale.

### 5.1.1. Criteri per l'assegnazione delle funzionalità di un sistema informativo

Molteplici criteri possono intervenire nella scelta di assegnazione delle funzionalità ai sistemi informatici già presenti o integrabili:

- **Costi:** includono i costi di acquisizione di un nuovo software, la sua integrazione, la manutenzione, la riprogettazione del legacy e l'interfacciamento
- **Compatibilità e interfacciamento:** con gli altri software del SI, la migrazione e deduplicazione dei dati e, infine, la vulnerabilità della soluzione
- **Performance:** l'efficacia della soluzione, la sua scalabilità, la capacità a rispondere ai bisogni funzionali e l'ergonomia e intuitività.

## 5.2. Mappatura applicativa: softwares, interfacce e basi di dati

Il livello applicativo descrive il sistema informativo aziendale attraverso i softwares, le loro interazioni – flussi e protocolli – e i dati scambiati. L'obiettivo della mappatura di questo livello è valutare le dipendenze inter-applicative, le caratteristiche dei flussi e identificare eventuali rischi di incompatibilità dei sistemi e di sicurezza.

Riprendendo la mappatura in Appendice 2 è possibile suddividere i sistemi in base al loro perimetro funzionale in tre categorie:

- Le soluzioni generaliste, ricoprenti almeno sei funzionalità. Ritroviamo principalmente gli ERP, MES, PLM e SCADA.
- Le soluzioni specializzate, ricoprenti meno di sei funzionalità. Ritroviamo LIMS, EMS, QMS, GMAO, WMS, TMS, CRM.
- Le soluzioni innovative, che vanno a fornire funzionalità specifiche attraverso l'utilizzo di tecnologie quali l'IA, la realtà aumentata, cobotica o digital twin.

La scelta di implementazione di soluzioni appartenenti a una di queste tre categorie risiede in molteplici riflessioni. Le soluzioni generaliste garantiscono una soluzione unica che permette di limitare le interazioni con gli altri sistemi e quindi la perdita di informazioni durante la loro trasmissione. Permette, tuttavia, una limitata personalizzazione e aumenta il rischio di compromettere l'intero sistema in caso di guasto.

Le soluzioni specializzate sono indirizzate principalmente a settori di nicchia o a processi necessitanti soluzioni personalizzate e specifiche per il loro settore. Vengono offerte funzionalità supplementari che rispondono in maniera precisa ai bisogni del cliente. Essi necessitano, tuttavia, una gestione del cambiamento più impattante e una attenzione particolare da porre sulla gestione delle connessioni, della compatibilità e dei dati.

Le soluzioni innovanti permettono di arricchire in termini tecnologici l'azienda e aumentare notevolmente le performance, andando, inoltre, a diminuire le dipendenze dalle altre tecnologie. Vi è, d'altro canto, una maggiore complessità di integrazione e in termini di gestione del dato.

### 5.2.1. Le comunicazioni tra sistemi informatici in un SI tramite API

I software in un SI per comunicare a vicenda utilizzano delle tecnologie middleware, ossia delle soluzioni che connettono due componenti o applicazioni per permettere lo scambio di dati. Esistono differenti tipologie di middleware, ma i più utilizzati sono gli API. Essi definiscono un insieme di protocolli che permettono di stabilire come i componenti di un SI devono interagire. Nella maggior parte dei casi i fornitori e sviluppatori di software integrano nell'applicazione un API manager proprietario. È possibile, inoltre, aggiungere in un sistema uno strato supplementare con un API gateway centrale per gestire le comunicazioni tra API



manager integrati a softwares sviluppati da differenti aziende e, quindi, con standard e linguaggi differenti.

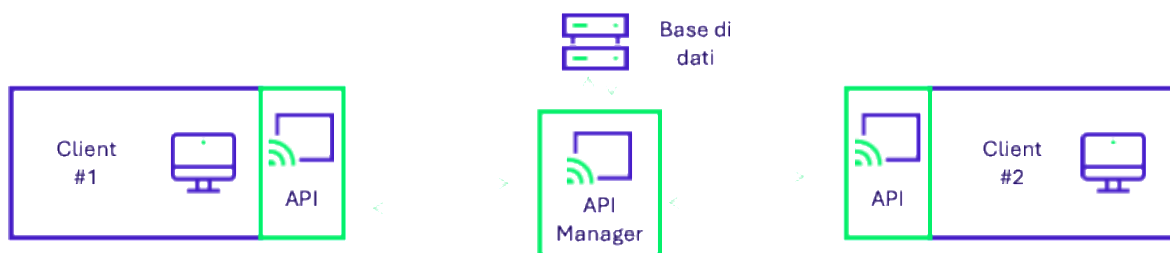


Figura 11 Rappresentazione del funzionamento di una comunicazione tra due client tramite API

Un API gateway è costituito di vari sottocomponenti con funzioni specifiche:

- Un componente responsabile dell'invio delle *query* agli API managers e di ricevere le risposte
- Un componente responsabile dell'amministrazione, ossia la verifica del mittente della *query*, del destinatario e degli accessi autorizzati
- Un componente di stoccaggio che gestisce i flussi e permette di fare un'analisi delle performance di gestione del traffico

Si possono, inoltre, classificare gli API in quattro categorie principali:

- **Pubblici:** accessibili e aperti, permettendone l'utilizzo da parte di qualsiasi utilizzatore. Un esempio è l'utilizzo della funzione per effettuare un acquisto su internet, come la connessione a PayPal.
- **Privati:** necessitano di una licenza per essere utilizzati e sono sfruttati all'interno di un ecosistema aziendale. Un esempio è la piattaforma di Netflix che sfrutta degli API per connettere i dispositivi alla biblioteca di film interna.
- **Condivisi:** sviluppati da alcune organizzazioni e condivisi con i loro partners. Un esempio è Skyscanner che per poter avere la visibilità del prezzo dei voli delle altre compagnie utilizza degli API condivisi.
- **Composti:** permettono di raggruppare più richieste attraverso diversi API. Ad esempio, al momento dell'acquisto di un oggetto su un sito internet, il fatto di creare un ordine, abbassare le quantità del carrello, cambiare lo status dell'ordine ed effettuare il pagamento, sono tutte *query* API raggruppate.

Gli API possono sfruttare differenti tipologie di protocolli per comunicare. Un protocollo di comunicazione è un insieme di regole stabilite con l'obiettivo di creare una comunicazione trasparente tra due dispositivi. Essi definiscono quindi una sintassi, una codifica digitale e delle procedure di verifica e correzione degli errori o di gestione delle perdite di dati durante la trasmissione.

Il modello OSI è un modello architetturale informatico che definisce sette livelli di astrazione delle telecomunicazioni in un sistema informativo. In altre parole, l'OSI fornisce uno standard che consente a sistemi di comunicazione eterogenei di comunicare tramite protocolli standard.

Ciascun livello del modello ha un compito preciso e comunica direttamente con i livelli adiacenti.

1. **Applicazione:** Si riferisce al tipo specifico di applicazione e ai suoi metodi di comunicazione standardizzati.
2. **Presentazione:** Riguarda principalmente la sintassi dei dati stessi che le applicazioni possono inviare e utilizzare.
3. **Sessione:** Responsabile del coordinamento di rete tra due applicazioni separate durante una sessione di comunicazione.
4. **Trasporto:** Garantisce che i pacchetti di dati arrivino nel giusto ordine, senza perdite o errori, o che possano essere recuperati senza problemi, se necessario. I protocolli TCP e UDP sono i più utilizzati.
5. **Rete:** Instradamento, trasferimento e indirizzamento su una rete dispersa o su più reti collegate di nodi o macchine.
6. **Collegamento dati:** Gestisce i frame di dati, che sono segnali digitali incapsulati in pacchetti di dati, e monitora i flussi di dati e gli errori.
7. **Fisico:** Si riferisce alla trasmissione di segnali digitali ed elettronici attraverso vari canali fisici.

Di seguito vengono elencati e confrontati i principali protocolli di comunicazione utilizzati dagli API.

**SOAP.** Un API SOAP utilizza dei protocolli di comunicazione che consentono lo scambio di informazioni strutturate tra sistemi distribuiti, basandosi principalmente su XML per la

formattazione dei dati. È molto rigido e orientato ai messaggi, il che significa che ogni richiesta e risposta segue una struttura predefinita. SOAP è spesso utilizzato in contesti aziendali che richiedono funzionalità avanzate come sicurezza (tramite WS-Security), affidabilità e transazioni. Supporta operazioni attraverso differenti protocolli di rete quali HTTP, SMTP e TCP. Tuttavia, l'uso di XML rende i messaggi pesanti e il protocollo stesso è considerato complesso da implementare. Attualmente sono i protocolli più diffusi nei sistemi legacy.

**REST.** Gli API REST è uno stile architetturale che sfrutta i metodi standard del protocollo HTTP, come GET, POST, PUT e DELETE, per la comunicazione tra client e server. I dati possono essere formattati in vari modi, come JSON o XML, ma JSON è il formato più diffuso per la sua leggerezza. Tuttavia, non include nativamente funzionalità come la sicurezza avanzata o la gestione delle transazioni, il che può richiedere integrazioni esterne per determinati casi d'uso.

**GraphQL.** È un linguaggio di query sviluppato da Facebook, progettato per superare i limiti delle API tradizionali. Invece di avere endpoint che restituiscono set di dati fissi, GraphQL permette ai client di specificare esattamente quali dati desiderano ottenere, il che riduce notevolmente la quantità di dati inutili trasmessi e migliora l'efficienza. Questo è particolarmente utile per le applicazioni mobili o front-end che necessitano di caricare solo specifici sottoinsiemi di informazioni, riducendo il sovraccarico e migliorandone l'efficienza. Sebbene offra una grande flessibilità, la sua potenza richiede una gestione attenta per evitare *query* troppo complesse o costose dal punto di vista computazionale.

**gRPC.** Framework di comunicazione open-source sviluppato da Google, che consente l'interazione tra client e server tramite chiamate di procedura remota (RPC). Utilizza il protocollo HTTP/2 per garantire prestazioni elevate e comunicazioni bidirezionali, supportando lo streaming dei dati in tempo reale. I dati vengono serializzati utilizzando Protocol Buffers, un formato binario altamente efficiente, che permette una trasmissione più rapida e compatta rispetto a JSON o XML. gRPC è ideale per l'esecuzione di operazioni su server distanti.

**Apache Thrift.** È un framework open-source originariamente sviluppato da Facebook per la creazione di servizi scalabili e performanti in ambienti con sistemi distribuiti. Thrift consente di definire interfacce di servizio e genera automaticamente il codice per diversi linguaggi di

programmazione, rendendo più semplice la comunicazione tra sistemi eterogenei. Supporta chiamate di procedura remota (RPC) ed è ottimizzato per le prestazioni di comunicazione, risultando efficiente in contesti di microservizi o in sistemi in cui è necessario interfacciare più linguaggi. Tuttavia, la sua configurazione e gestione può risultare più complessa rispetto alle altre soluzioni. In Tabella 1 viene sintetizzato il confronto tra i differenti protocolli sopra presentati.

	<b>SOAP</b>	<b>REST</b>	<b>gRPC</b>	<b>GraphQL</b>	<b>Apache Thrift</b>
<i>Protocolli</i>	Tutti	HTTPS	HTTPS	HTTPS	Tutti
<i>Formato di dati</i>	XML	XML, JSON, testo, HTML	XML, JSON	XML, JSON, testo, HTML	Binario
<i>Sicurezza</i>	Possibile integrazione del funzionamento in HTTPS	HTTPS	Differenti possibilità in base al formato dei dati	HTTPS	HTTPS
<i>Vantaggi</i>	Altamente strutturato	Performance e flessibilità	Performance e trasferimento di dati tra server distanti	Minimizzazione del numero di transazioni	Compatibilità con un grande numero di linguaggi
<i>Svantaggi</i>	Messaggi voluminosi e difficoltà di dimensionamento	Query a struttura fissa e impossibilità di filtrarle	Complessità dell'architettura	Complessità del backend e struttura	Scarsa diffusione e documentazione
<i>Use case</i>	Legacy e API privati	Applicazioni moderne e API pubblici	Applicazioni che eseguono azioni su server distanti	Transazioni con più fonti di dati e necessità di filtraggio delle tabelle	Servizi multilinguaggio, servizi scalabili e affidabili

Tabella 1 Confronto dei principali protocolli utilizzati dagli API

### 5.2.2. Lo stoccaggio e gestione delle informazioni: le basi di dati

Una base di dati è un insieme organizzato di dati strutturati, semi-strutturati e non strutturati, memorizzati in modo da poter essere facilmente gestiti, recuperati e aggiornati. Le basi di dati sono progettate per consentire la memorizzazione di grandi quantità di informazioni e

per facilitare l'accesso rapido e l'aggiornamento efficiente di tali informazioni. Vengono gestite da sistemi di gestione di basi di dati (DBMS) che permettono operazioni come l'inserimento, la modifica, la cancellazione e la consultazione dei dati. I database possono essere relazionali (strutturati in tabelle) o non relazionali (come quelli basati su documenti, grafi o chiavi-valori). È possibile individuare due principali tipologie di basi di dati:

- **Data Warehouse:** base di dati relazionale che raggruppa dei dati strutturati da sorgenti differenti.
- **Data Lake:** base di dati utilizzata per lo stoccaggio in un grande “lago” di dati centralizzato per gestire dati strutturati e non strutturati nella loro forma grezza.
- **Data LakeHouse:** una particolare base di dati ibrida che sfrutta le particolarità e caratteristiche di entrambe le soluzioni sopra citate.

**Data Warehouse.** Esse sfruttano dei processi di ETL (Extraction, Transformation, Loading) per combinare i dati in una grande base di riferimento centrale. Utilizzano, inoltre, delle regole operative per “pulire” e organizzare i dati grezzi e prepararli per lo stoccaggio, l’analisi e il Machine Learning. Un Data Warehouse è composto da differenti *Data Marts*. Un Data Mart è un sottoinsieme di un Data Warehouse che contiene un insieme specifico per supportare le esigenze analitiche e decisionali di un particolare dominio funzionale o dipartimento aziendale. I Data Marts sono progettati per fornire informazioni più mirate e accessibili agli utenti finali, facilitando le analisi rapide e le decisioni strategiche in un determinato settore. A differenza del Data Warehouse, che raccoglie e gestisce dati da più fonti per tutta l'azienda, un Data Mart si concentra su un ambito specifico per soddisfare le esigenze di gruppi di utenti particolari. In Figura 12 viene rappresentata la gestione dei dati nel caso di una base di dati di tipo Warehouse.

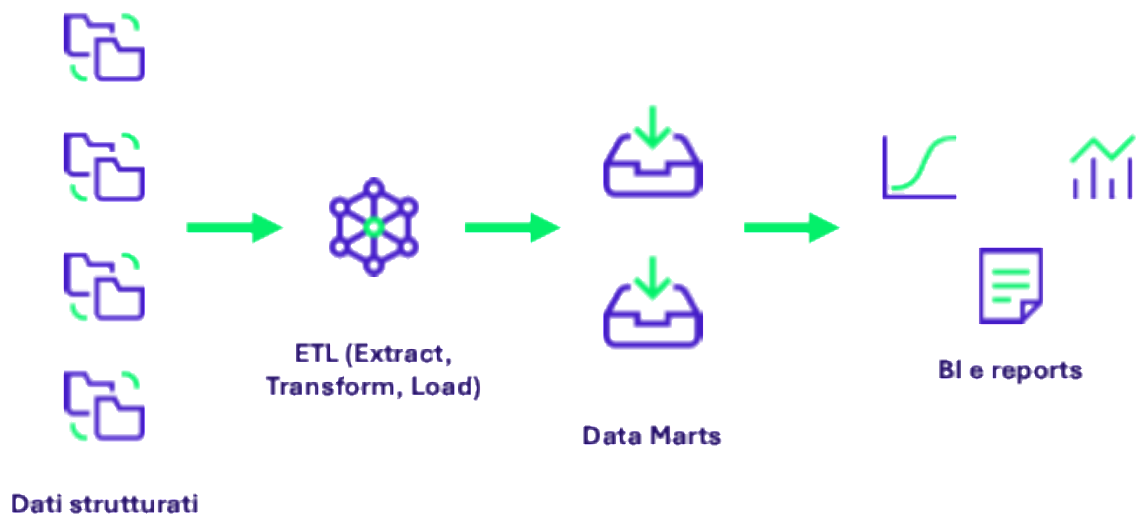


Figura 12 Il data management con un Data Warehouse

Il funzionamento di queste tipologie di basi di dati può essere sintetizzato in quattro tappe:

- **Estrazione:** l'ETL estrae i dati dalle varie sorgenti
- **Pulizia:** la base di dati pulisce i dati che ingerisce e procede alle deduplicazione e riformattazione necessarie per organizzare i dati in maniera coerente e strutturata
- **Trasformazione:** adattamento dei modelli di dati ai casi d'uso.
- **Aggiornamento:** aggiornamento continuo dei dati attraverso le sorgenti dei dati alle quali è connessa. I dati storici sono arricchiti in maniera continuativa con l'integrazione di nuovi dati.

**Data Lake.** A differenza dei data Warehouse, esse stoccano dati di tutte le tipologie e volumi. Per fare ciò sfruttano tecnologie ELT (Extract, Load, Transform) che copiano o esportano i dati grezzi dalle sorgenti verso le zone di trasferimento per poi caricarle nel lago di dati. La tappa di trasformazione viene fatta per ultima attraverso l'utilizzo di "query" SQL per elaborare i dati prima dell'analisi. In Figura 13 viene rappresentata la gestione dei dati nel caso di una base di dati di tipo Lake. È possibile, inoltre, aggiungere un software ETL che prende il Data Lake come sorgente dei dati per aggiungere un livello di base di dati di tipo Warehouse successivamente. Questo permette di sfruttare i vantaggi di entrambe le tipologie di DBMS e creare delle strutture di tipo LakeHouse.

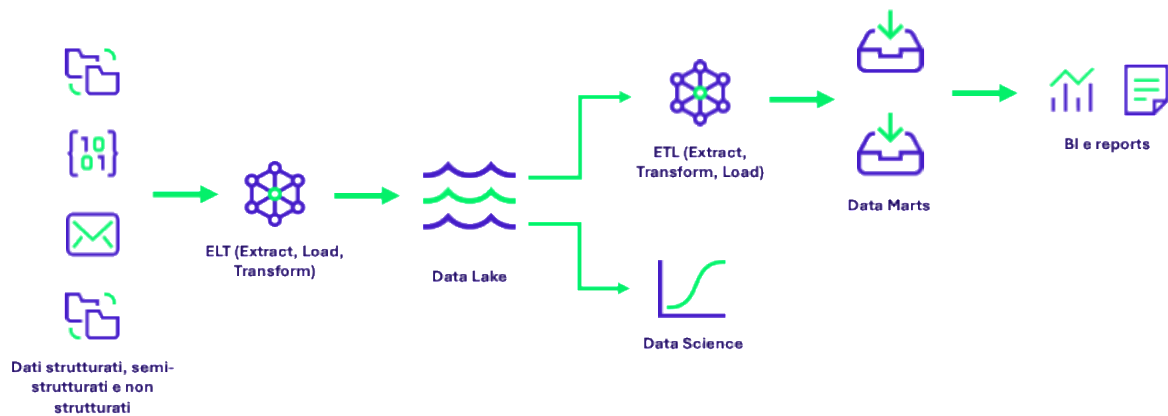


Figura 13 Il data management con un Data Lake

### 5.2.3. Mappatura applicativa di un SI industriale e modello di Purdue

Dopo aver definito e descritto i differenti componenti dal punto di vista applicativo è possibile andare a rappresentare il modello di Purdue sulla base della mappatura funzionale realizzata al paragrafo precedente. In Appendice 3 viene rappresentata la mappatura applicativa di un SI composto da soluzioni generaliste per visualizzare le coperture funzionali di ciascuno dei sistemi utilizzati. È stato ipotizzato l'utilizzo di un MES, un PLM, un ERP ed uno SCADA, accostati da delle tecnologie di BI e di gestione dei progetti, e una base di dati di tipo Warehouse. Sono state, inoltre, rappresentate le possibili interazioni e flussi tra i differenti sistemi.

I MES interagiscono con l'ERP per raccogliere tutte le informazioni sulle vendite, risorse disponibili e acquisti e inviare gli stati in tempo reale di produzione, con il PLM per i dati tecnici e procedure di fabbricazione e con lo SCADA per inviare gli ordini di fabbricazione e raccogliere i dati sulle linee di produzione. Tutte le informazioni vengono inviate alla base di dati per ingestione e stoccaggio.

### 5.3. Mappatura tecnica: le fondamenta dei sistemi informativi

Il livello tecnico è una rappresentazione di un SI che permette di descrivere l'infrastruttura sulla quale gli elementi applicativi sono implementati, ossia gli elementi fisici. In altre parole, la mappatura tecnica rappresenta la ripartizione geografica delle attrezzature di rete nei differenti siti dell'organizzazione. Si ritrovano quindi:

- **Infrastruttura materiale:** servers, dispositivi di stoccaggio, sensori e attuatori
- **Infrastruttura logica:** sistemi informatici di esecuzione, middlewares e HMI

- **Dispositivi di sicurezza:** Meccanismi e protocolli di sicurezza, proxy servers, portali di sicurezza, firewalls, DMZ, ZTL
- **Dispositivi di rete:** LAN, MAN, WAN, protocolli di comunicazione

Gli obiettivi sono la presentazione delle caratteristiche tecniche dei sistemi fisici, identificare la presenza di punti singoli di guasto da un punto di vista di sicurezza informatica, e rappresentare la rete, la sua segmentazione e i suoi componenti.

### 5.3.1. Servers

I servers sono l'elemento chiave in una architettura di sistemi informativi in quanto ospitano fisicamente i componenti del livello applicativo. Un server è un sistema informatico che fornisce servizi, risorse o dati ad altri computer, chiamati *client*, all'interno di una rete. I server rispondono quindi alle richieste dei client, gestendo l'elaborazione delle informazioni, la memorizzazione e la distribuzione dei dati o l'esecuzione di applicazioni, attraverso un modello detto client-server. I server possono essere sia dispositivi hardware specializzati, sia software configurati per gestire specifiche funzionalità in una rete. Si possono distinguere i server fisici, computer dotati di risorse hardware dedicate, come CPU, RAM, storage e schede di rete, e server virtuali (Virtual Machines) create all'interno di un server fisico tramite software di virtualizzazione come VMware o Hyper-V. Un singolo server fisico può ospitare molteplici server virtuali, ognuno con il proprio sistema operativo e applicazioni, condividendo però le risorse hardware del server fisico.

I server possono svolgere differenti funzioni in un sistema informativo:

- Server di **applicazioni**: utilizzati per ospitare ed eseguire applicazioni software e fornire accesso a funzionalità di back-end come database, autenticazione e servizi aziendali complessi.
- Server di **basi di dati**: ospitano uno o più database e gestiscono le richieste di accesso ai dati, eseguendo query SQL e fornendo i risultati ai client o ad altri server. Vengono utilizzati per l'archiviazione, la gestione e il recupero dei dati in modo efficiente.
- Server di **file**: ospitano e distribuiscono file che possono essere condivisi da un gran numero di client o utenti. Vengono utilizzati per centralizzare l'archiviazione dei file, consentendo ai client di accedere, modificare e condividere questi ultimi da una posizione comune.



- Server **web**: ospitano software e dati che gli utenti richiedono da Internet o da una intranet.
- Server **proxy**: un sistema o router che fornisce un gateway tra un client e un server, focalizzandosi sulla messa in sicurezza della comunicazione.

In base alle necessità aziendali e alle possibilità fisiche di ospitare servers, esistono differenti soluzioni per organizzare l'infrastruttura tecnica:

- **On-premise**: le infrastrutture IT sono fisicamente situate in un datacenter aziendale e gestite internamente. In questo modo, l'azienda ha un controllo totale su ogni aspetto tecnico dai server e lo stoccaggio dei dati, alla rete e la sicurezza informatica. L'azienda acquista quindi una licenza e si occupa personalmente della loro implementazione nel proprio ecosistema informatico.
- **IaaS** (Infrastructure as a Service): il fornitore di servizi e applicazioni offre e gestisce l'infrastruttura fisica che viene virtualizzata per il cliente che invece gestisce direttamente le risorse virtualizzate.
- **PaaS** (Platform as a Service): il fornitore ospita l'hardware e il software sulla propria infrastruttura e fornisce all'utente una piattaforma.
- **SaaS** (Software as a Service): il software è distribuito e gestito dal fornitore come servizio cloud accessibile tramite internet. Applicazioni di questo tipo sono chiamate *on-shelf*, esse sono, infatti, pronte per l'uso e a essere utilizzate immediatamente attraverso un browser web o altre interfacce. Il fornitore si occupa di tutti gli aggiornamenti, la sicurezza e la manutenzione.

### 5.3.2. Rete

Una rete è definita come il collegamento di almeno due sistemi informatici mediante una connessione via cavo o wireless. Esistono diversi tipi di rete, a seconda delle loro dimensioni, del modo in cui sono segmentate, delle relazioni tra le apparecchiature, ecc.

Nella gestione della rete si possono individuare principalmente due componenti: i routers e gli switch. Il primo è un dispositivo che permette la connessione di più reti attraverso l'invio di pacchetti di dati. Esso opera al livello tre del modello OSI (livello di rete) e utilizza gli indirizzi IP per determinare il percorso del traffico. Gli switch permettono, invece, di gestire il traffico all'interno di una rete locale e inoltra i dati tra i dispositivi della suddetta rete. Esso

opera al livello 2 del modello OSI (livello di collegamento dati) e utilizza gli indirizzi MAC per inviare i dati al dispositivo corretto.

Esistono diverse categorie e classificazioni di reti:

- **LAN** (Local Area Network): essa ricopre una piccola area geografica e spesso raggruppa i client e i server di uno stesso dipartimento o funzione aziendale. La LAN viene tipicamente utilizzata per condividere risorse internamente come file, stampanti e connessioni internet.
- **MAN** (Metropolitan Area Network): ricopre un'area geografica più ampia delle LAN ed è progettata per collegare reti locali diverse ma vicine. Essa può essere utilizzata per andare a collegare, ad esempio, un magazzino e il sito produttivo di un'azienda nella stessa città.
- **WAN** (Wide Area Network): quest'ultima rete permette di andare a collegare le reti locali e metropolitane distanti tra loro geograficamente.

### 5.3.3. Meccanismi di sicurezza

Da un punto di vista dei meccanismi di sicurezza del livello tecnico, si vanno a individuare le segmentazioni della rete, firewalls, DMZ e VPN.

Una rete informatica può essere suddivisa in più *subnet* o segmenti più piccoli e distinti, ma anche isolati dagli altri fisicamente attraverso degli switch o routers o virtualmente attraverso dei firewalls. Un'azienda ha interesse a segmentare una rete al fine di migliorare la sicurezza e le prestazioni, e per avere controllo sulla gestione del traffico, andando a limitare e gestire gli accessi tra e verso i segmenti della stessa.

Una Demilitarized Zone è un componente fondamentale nella segmentazione di un sistema informativo. Come anticipato, essa si aggiunge come livello supplementare nel modello di Purdue per andare a bloccare i flussi diretti tra due reti, nella maggior parte dei casi tra la rete OT e quella IT. All'interno di una DMZ possiamo trovare i seguenti dispositivi:

- Uno o più **firewalls**
- Un server **SMTP** (Simple Mail Transfer Protocol) ossia un server utilizzato per garantire la sicurezza dell'invio e la ricezione di messaggi per posta elettronica su internet.

- Un server **DNS** (Domain Name System) che traduce i nomi di dominio leggibili dall'uomo in indirizzi IP leggibili dalla macchina
- Un server **WWW** (World Wide Web) che risponde alle richieste provenienti dal World Wide Web su una rete pubblica (Internet) o privata, utilizzando principalmente il protocollo HTTP
- Un server proxy

Negli ultimi anni, si è sempre più diffuso un approccio alla sicurezza informatica del tipo *Zero Trust* che parte dal presupposto che ogni utente, dispositivo o sistema, sia a priori inaffidabile. Questo modello modifica la gestione degli accessi e delle autenticazioni ai livelli della DMZ e dei firewalls andando a verificare continuamente l'identità dell'utente ad ogni suo tentativo di collegamento alle risorse del SI, indipendentemente dalla sua posizione di origine.

#### 5.3.4. Mappatura tecnica di un SI industriale

La mappatura in Appendice 4 è stata realizzata mantenendo il modello di Purdue e andando a rappresentare i server fisici dove le applicazioni sono ospitate, la segmentazione di un ipotetico sito produttivo e i differenti meccanismi di sicurezza.

Nel livello dei dispositivi industriale è stata ipotizzata una segmentazione per ogni area produttiva andando a rappresentare le differenti HMI, sensori, attuatori e PLC. Per ogni segmento è stato aggiunto uno switch. Ogni area produttiva è collegata al segmento degli operatori di fabbrica operanti sullo SCADA, attraverso un router. Il segmento dell'infrastruttura amministrativa della rete industriale (ICS) si occupa di andare a gestire il traffico e gli accessi dei livelli 0,1,2. Un *Data Historian* raccoglie e stocca tutti i dati di produzione. Nei livelli 3 e 4 vengono rappresentati i server delle applicazioni IT, della base di dati e dei file, ma anche un'eventuale piattaforma cloud, il router WAN e i collegamenti internet tramite VPN.

## 6. Le evoluzioni dei sistemi informativi

In seguito alla realizzazione dello stato dell'arte, sono state individuate numerose problematiche e piste di approfondimento per tutti e tre i livelli dell'architettura. È stato quindi deciso di focalizzarsi sul livello logico per andare ad approfondire quali sono le modalità di sviluppo innovative delle applicazioni industriali e come queste ultime saranno interconnesse nel futuro. Al fine di procedere con la costruzione degli scenari evolutivi, è necessario, quindi, andare ad approfondire alcuni concetti e tecnologie innovative attualmente disponibili sul mercato.

In primo luogo, da un punto di vista applicativo è sempre più diffuso il concetto di micro-servizi, ossia un approccio architettonico in cui le applicazioni software sono costituite da diversi componenti interconnessi, che rispondono a funzionalità specifiche. Questo approccio innovativo è stato sviluppato in parallelo al concetto di *event-driven*, un modello di architettura informatica il cui stato è definito dagli eventi che si verificano al suo interno. Si ha, quindi, un discostamento dai modelli classici *data-driven* che presuppongono una connessione asincrona tramite API.

Le architetture EDA (Event-driven Architecture) presuppongono l'utilizzo di event brokers, una particolare tipologia di middlewares che alla creazione di un evento o, in altre parole, di un cambiamento di stato di un dispositivo, attiva una risposta a tutti i dispositivi collegati a quello di partenza. Nel settore industriale gli event brokers più utilizzati sono basati su protocolli MQTT e su piattaforme Apache Kafka.

Per quanto riguarda la gestione e lo stoccaggio dei dati vi è una sempre maggiore richiesta di soluzioni che integrino o supportino l'utilizzo di sistemi di IA e Machine Learning. I Data Hub, soprattutto nel campo industriale, permettono di integrare allo stesso tempo dei Data Lake per supportare le analisi predittive e dei Data Warehouse per tutto ciò che riguarda le analisi BI. Essi, inoltre, supportano ed integrano dispositivi di ETL, ELT e di data governance.

Queste evoluzioni chiave sono approfondite nei seguenti paragrafi per poi essere inserite in quattro scenari di architetture logiche.

## 6.1. Micro-servizi: un cambiamento nell'approccio dello sviluppo delle applicazioni

La crescente esigenza di agilità e flessibilità e l'integrazione di nuove tecnologie nei processi produttivi hanno evidenziato la necessità di una maggiore interoperabilità, centralizzazione, modularizzazione e indipendenza. Nelle architetture monolitiche tradizionali, tutti i componenti di un'applicazione software sono strettamente collegati e distribuiti come un'unica entità. È il caso delle applicazioni “generaliste” inserite nella mappatura logica dello stato dell'arte. Un'architettura a microservizi, invece, suddivide l'applicazione in piccoli servizi che possono essere distribuiti indipendentemente l'uno dall'altro e comunicare tra loro attraverso una rete.

Lo sviluppo a microservizi è caratterizzato da un approccio modulare in cui le applicazioni vengono create da una serie di servizi distinti che possono essere distribuiti e aggiornati in modo indipendente, anziché da un unico corpo monolitico di codice. I microservizi sono quindi caratterizzati da una grande autonomia, in quanto ogni servizio è sviluppato, distribuito, eseguito e ridimensionato indipendentemente e senza influenzare il funzionamento degli altri componenti. Essi sono, inoltre, altamente specializzati in quanto sono progettati per specifici bisogni e specifiche funzionalità. Nel corso del tempo i microservizi possono evolversi e aumentare in complessità, offrendo, tuttavia, la possibilità di essere scomposti in servizi più piccoli per mantenere i vantaggi di questa specializzazione.

In sintesi, i vantaggi dei microservizi sono i seguenti:

- Flessibilità e agilità nello sviluppo, concezione e manutenzione
- Libertà tecnologica grazie alla possibilità di utilizzare differenti linguaggi di programmazione e strumenti per ogni servizio. Ogni team ha quindi la libertà di scegliere gli strumenti migliori per risolvere i loro problemi specifici.
- Resilienza in caso di errori o guasti. Ogni servizio essendo indipendente non impatta il funzionamento degli altri.
- Scalabilità e semplicità di distribuzione. Essendo ogni team autonomo e indipendente, viene facilitata l'integrazione continua di nuovi componenti e funzionalità ma anche di accelerazione del time-to-market.

Possono, tuttavia, presentarsi degli inconvenienti in caso di gestione e integrazione non ottimale dei microservizi. È possibile trovarsi in situazioni di eccessiva dipendenza tra microservizi, riportando quindi a una situazione simile alle architetture ad applicazioni monolitiche. Un'architettura a microservizi non può, inoltre, essere intesa come lo standard migliore per lo sviluppo delle applicazioni in quanto secondo i casi d'uso, potrebbe essere ottimale optare per una applicazione monolitica, come nel caso di aziende con budget relativamente bassi, necessità di implementazione rapida e di funzionalità con grande dipendenza le une dalle altre.

### 6.1.1. La containerizzazione: Kubernetes e Docker

Questa modernizzazione dell'infrastruttura comporta il passaggio ad applicazioni cloud-native distribuite tramite tecnologie a container.

Un container è un'unità di software in cui il codice dell'applicazione è impacchettato con tutti i file necessari per l'esecuzione. Le applicazioni containerizzate sono più facili da spostare da un ambiente all'altro, pur mantenendo tutte le loro funzioni.

Sul mercato esistono diverse soluzioni per lo sviluppo delle applicazioni a microservizi e si possono suddividere in tre categorie:

- Piattaforme per la containerizzazione ed esecuzione dei servizi quali Docker
- Piattaforme per l'orchestrazione dei servizi e per la creazione di cluster, quali Kubernetes
- Piattaforme integranti le due soluzioni sopra descritte e offrenti funzionalità supplementari come Amazon Elastic Kubernetes/Container Service e Microsoft Azure

Un'architettura a microservizi ideale che sfrutta entrambe le tecnologie open-source Docker e Kubernetes, viene descritta di seguito e rappresentata in Appendice 5.

Docker è una piattaforma di containerizzazione e *runtime* che aiuta gli sviluppatori a costruire, distribuire ed eseguire container. Essa permette di creare, gestire e distribuire container in modo efficiente, garantendo che i microservizi siano eseguiti in un ambiente isolato e uniforme, indipendentemente dal sistema operativo sottostante.

Kubernetes è una piattaforma che orchestra i sistemi di runtime dei container su un cluster di risorse in rete. In altre parole, gestisce i cluster di container e fornisce strumenti per il bilanciamento del carico, l'esecuzione delle applicazioni e l'interconnessione tra client e servizio.

Questa piattaforma organizza quindi l'architettura in cluster ossia raggruppamenti di servizi e di un orchestratore. Ogni cluster è composto da un piano di controllo che gestisce i flussi di dati, i carichi di lavoro e pianifica l'esecuzione di ogni container, e di un piano di dati composto a sua volta da *node workers*.

I componenti chiave di questo ambiente sono i *pod* ossia un gruppo di uno o più container, con risorse di rete e di storage condivise e una specifica per l'esecuzione dei container. I contenuti di un *pod* sono sempre co-localizzati e co-schedulati ed eseguiti in un contesto condiviso. In altre parole, se un'applicazione richiede più container che lavorano insieme, questi vengono posizionati all'interno dello stesso *pod*. Si parla, inoltre, di *ReplicaSet* che garantisce che un determinato numero di repliche di un *pod* sia sempre in esecuzione nel cluster al fine di fornire ridondanza, scalabilità e disponibilità di servizio.

Il piano di controllo è costituito da:

- Un **server API** responsabile della gestione delle richieste esterne ed interne. Funziona, quindi, come interfaccia di gestione principale delle interazioni tra node master e node workers ma anche tra differenti clusters. Questo server è *stateless*, permette infatti di memorizzare nell'*etcd* le informazioni sullo stato del cluster.
- L'**etcd** è il key-value-store che si occupa di stoccare tutte le informazioni critiche di configurazione e di stato del cluster.
- Un **pianificatore** responsabile di assegnare i servizi su nodi specifici in base a flussi di lavoro automatizzati e a condizioni definite dall'utente. Questo componente decide quindi su quale nodo far eseguire un *pod* in base a determinati criteri e vincoli, come risorse disponibili (CPU, memoria), requisiti di tolleranza ai guasti, politiche di affinità ecc.
- Un **controllore cloud** è un componente aggiuntivo responsabile di collegare un cluster Kubernetes con l'API di un provider di cloud e di gestire risorse cloud specifiche, come nodi, bilanciatori di carico, volumi di storage e indirizzi IP pubblici.

- Un **controllore** monitora lo stato di un cluster e assicura che lo stato attuale del cluster corrisponda a quello desiderato. Ha quindi il compito di assicurare che siano presenti il numero desiderato di repliche di un pod e invia istruzioni per spostare il cluster verso lo stato desiderato dall'operatore del cluster.

Ciascun node worker è costituito da:

- Uno o più **Pods**
- Un **container runtime engine**, responsabile dell'esecuzione dei container
- Un **kubelet** responsabile di garantire che i contenitori specificati nella configurazione del pod siano in esecuzione su un nodo specifico e di gestire il loro ciclo di vita. Esegue le azioni comandate dal piano di controllo.
- Un **kube-proxy** è il componente che gestisce tutte le comunicazioni di rete all'esterno e all'interno del cluster, inoltrando il traffico o rispondendo al livello di filtraggio dei pacchetti del sistema operativo.

In sintesi un'architettura di questo tipo offre un approccio moderno e scalabile per sviluppare, distribuire e gestire applicazioni complesse, garantendo flessibilità, resilienza e miglior gestione delle risorse.

### 6.1.2. Il data management nelle architetture a microservizi

Nelle architetture a microservizi esistono diversi modelli di gestione dei dati:

- Un database per servizio è un modello che offre una grande scalabilità e permette di sfruttare appieno i vantaggi della decentralizzazione dei microservizi
- Un database condiviso che accede ai dati usando transazioni ACID (Atomicità, Coerenza, Isolamento e Durabilità) locali. Tuttavia, questo modello è contrario alla natura dei microservizi ed è quindi ideale per le applicazioni legacy e per le applicazioni semplificate in cui i carichi di lavoro in lettura sono elevati rispetto a quelli in scrittura.

Il modello con un database per servizio prevede che ogni microservizio abbia il proprio database, ma questo modello non soddisfa le proprietà ACID. Una transazione di tipo ACID permette di garantire l'affidabilità e la coerenza nella gestione di una base di dati e deve soddisfare le seguenti condizioni:



- **Atomicità:** garantisce che una transazione sia eseguita nella sua interezza o che non venga eseguita affatto. Questo impedisce l'eventuale perdita e corruzione dei dati, nel caso, ad esempio, di interruzione a metà del flusso.
- **Consistenza:** assicura che le transazioni non alterino la consistenza della base di dati. In altre parole, garantiscono che dopo la loro esecuzione la base di dati rispetti tutte le regole di integrità referenziale, di dati ed eventuali vincoli applicati.
- **Isolamento:** garantisce che le transazioni vengano eseguite in maniera sequenziale e senza che esse interferiscano l'una con l'altra nel caso di esecuzione in simultanea.
- **Durabilità:** garantisce che l'effetto di una transazione permanga nella base di dati anche in caso di guasto del sistema.

In casi come questi si utilizza un modello di progettazione delle transazioni specifico, chiamato *Saga Pattern*.

***Saga Pattern.*** Una saga è una sequenza di transazioni che aggiorna ogni servizio e pubblica un messaggio o un evento per attivare la fase successiva della transazione. Se una fase fallisce, la saga esegue transazioni di compensazione che annullano le transazioni precedenti. In Figura 14 viene rappresentato un esempio di applicazione con tre microservizi: uno per la gestione degli ordini di fabbricazione, uno per la gestione dei processi e uno per l'acquisizione dei dati della linea di produzione. Supponendo che il servizio "Programmazione" debba ottenere le informazioni sullo stato della linea dal servizio "Acquisizione dei dati" e i dati di performance della linea dal servizio "Gestione dei processi", esso attiverà una saga di transazioni (pattern) per raccogliere le suddette informazioni.



Figura 14 Esempio Saga Pattern

In una saga pattern possono essere eseguite differenti tipologie di transazioni:

- Le **transazioni compensate** sono transazioni che possono essere potenzialmente annullate elaborando un'altra transazione con effetto opposto.
- Una **transazione pivot**, ossia il punto di decisione di una saga: se viene convalidata, la saga continua fino al suo completamento.
- Le **transazioni ritentabili** sono transazioni che seguono la transazione pivot e sono garantite per il successo.

Una saga pattern può essere implementata in due modalità principali: coreografica e orchestrata.

Nel primo caso, ogni microservizio esegue la sua transazione e, una volta completata, emette un evento per attivare la transazione della saga nel microservizio successivo. Questo garantisce un'autonomia e indipendenza ad ogni microservizio e si adatta a sistemi molto distribuiti con alta disponibilità (repliche). Tuttavia, può risultare complicato tracciare e gestire il flusso e il traffico di eventi.

Nel secondo caso, viene integrato un orchestratore centrale con l'obiettivo di gestire il flusso delle transazioni, inviando comandi ai vari microservizi per far eseguire le transazioni locali in maniera sequenziale, secondo la saga. Si ha quindi un maggiore controllo sul traffico, permettendo di avere una visione completa della saga e del flusso di transazioni, tuttavia,

l'orchestratore diventa un punto singolo di guasto o un collo di bottiglia nel caso di un numero eccessivo di transazioni.

### 6.1.3. Service Mesh: una gestione avanzata del traffico e della sicurezza in un ambiente distribuito

Quando un'organizzazione crea sempre più microservizi, la complessità aumenta. I team spesso costruiscono in linguaggi diversi e distribuiscono in ambienti multipli, e le organizzazioni sono generalmente indipendenti nel modo in cui gestiscono i loro servizi. Questa complessità può andare a impattare e ridurre l'agilità di un approccio a microservizi.

Nel caso degli API tradizionali, ogni microservizio gestisce internamente la logica del bilanciamento del carico, la gestione degli errori e gli aspetti legati alla rete. Inoltre, la crittografia e l'autenticazione sono implementate e mantenute in ogni singolo microservizio. Con l'aumentare del numero di servizi e dei flussi può risultare inoltre complesso e frammentario il monitoraggio delle comunicazioni.

I *Service Mesh* introducono un'infrastruttura di rete trasparente che gestisce tali interazioni, rendendo le comunicazioni tra i microservizi sicure, osservabili, rapide e gestibili. Essi si basano su *proxy sidecars*, distribuiti su ogni microservizio, che intercettano tutto il traffico in entrata e in uscita dal servizio, e su un piano di controllo attraverso il quale gli sviluppatori possono configurare e aggiungere politiche di sicurezza e comunicazione, indipendentemente dalla tecnologia utilizzata.

Nel caso di un'architettura che utilizza il *saga pattern* per la gestione delle transazioni, il *service mesh* semplifica il processo di coordinazione ed esecuzione delle transazioni:

- permette di controllare e reindirizzare il traffico per implementare le transazioni compensative,
- fornisce gli strumenti per il tracciamento, al fine di monitorare l'esecuzione di una *saga*
- implementa politiche di *circuit breaker*, *retry policy* e *timeouts* per aumentare la resilienza delle interazioni

In Figura 15 è rappresentato un esempio di architettura di service mesh tipo Istio, una piattaforma open-source per la gestione delle comunicazioni tra microservizi. Il piano di controllo è costituito da tre elementi principali:

- Un *pilot* per la gestione della configurazione dei proxy come il *routing*, bilanciamento del carico, gestione della rete, individuazione dei servizi, resilienza.
- Una *citadel* che gestisce l'autenticazione tramite meccanismi mTLS, il controllo degli accessi e la crittografia per garantire la sicurezza delle comunicazioni.
- Un *galley*, responsabile dell'ingestione, della convalida e della distribuzione della configurazione di Istio attraverso il pilot ai proxy sidecar.

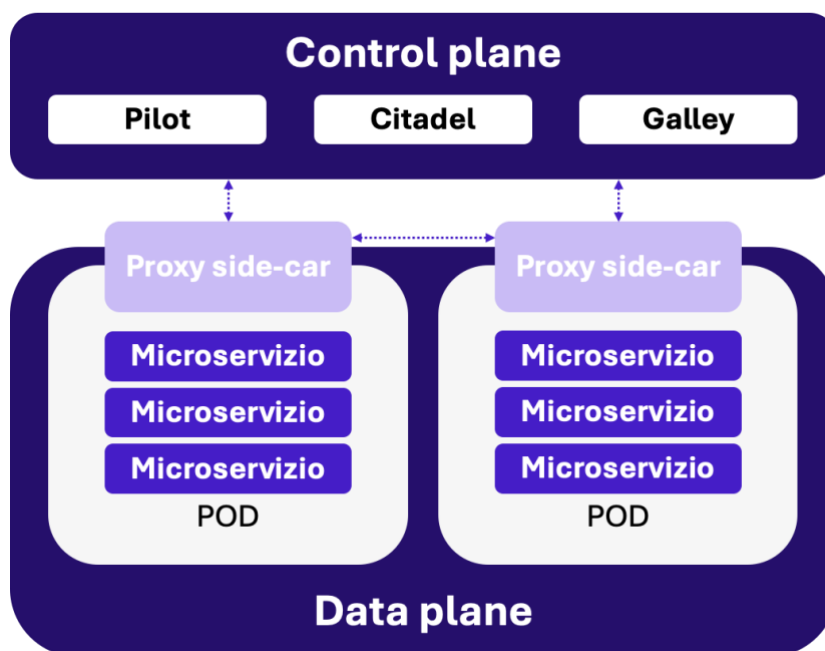


Figura 15 Architettura service mesh Istio

A differenza degli API, che operano principalmente al livello 7 (livello applicazione) del modello OSI, i service mesh operano sia al livello 4 (livello di trasporto) che al livello 7. Essi, infatti, gestiscono in modo più granulare il traffico di rete, andando a monitorare e controllare le connessioni tra i servizi su protocolli come TCP e UDP ma offrendo anche la possibilità di gestire protocolli applicativi come HTTP e TLS.

In sintesi, si può affermare che i service mesh offrono funzionalità più complete in confronto agli API che necessitano di estensioni supplementari per la sicurezza, il bilanciamento del

carico e l'osservabilità, e, inoltre, sono ottimali per la gestione delle comunicazioni in un'architettura a microservizi.

## 6.2. Un passaggio verso le architetture ad eventi

La crescita e lo sviluppo del Cloud, il passaggio all'approccio di sviluppo delle applicazioni in microservizi e la diffusione di metodologie agili con team che lavorano su componenti su piccola scala, hanno portato a un'evoluzione delle architetture applicative verso architetture di microservizi "Cloud-native". Tuttavia, questa frammentazione porta a un aumento della combinazione di invocazioni tra questi servizi (traffico est-ovest). Risulta quindi opportuno domandarsi se un modello di invocazione delle transazioni sincronizzato tramite API sia tutt'ora ottimale o se sia più conveniente passare verso un modello asincrono detto "di eventi". L'architettura basata sugli eventi è un approccio in cui le azioni in un sistema sono attivate da eventi specifici, come azioni dell'utente o cambiamenti di stato. I componenti del sistema interagiscono in modo non vincolato. Questa tipologia di architettura si discosta dal concetto classico e più diffuso di architettura *data-driven* dove si ha una grande dipendenza dai dati condivisi. Queste ultime necessitano di un'orchestrazione minuziosa, a differenza delle architetture *event-driven* che sono più reattive e asincrone. Un'architettura EDA consente inoltre una maggiore flessibilità nella gestione degli eventi e processi paralleli.

Generalmente le architetture EDA si basano su dei modelli pubblicazione-sottoscrizione o pub/sub (Figura 16), in cui i produttori di eventi, detti publishers (pubblicatori), pubblicano i messaggi su un canale detto *topic*. Tutti i client abbonati a un determinato *topic*, detti *subscribers* (sottoscrittori), ricevono automaticamente i messaggi corrispondenti da parte dell'event broker (Gartner, 2023).



Figura 16 Modello pubblicazione-sottoscrizione

Si possono avere cinque differenti tipologie di orchestratori di eventi (Quix, n.d.):

- **Bus di eventi:** segue il modello pub/sub e viene utilizzato quando esiste una relazione uno-a-uno tra il mittente e il destinatario del messaggio. Ogni messaggio viene inviato a un singolo destinatario e viene consumato una sola volta. Un approccio di questo tipo, *point-to-point*, viene utilizzato quando un messaggio deve essere elaborato una sola volta.
- **Pub/sub classico:** un servizio pub/sub trasmette lo stesso messaggio a tutti gli abbonati a un argomento specifico, utilizzando un modello *push* in cui i consumatori elaborano un messaggio non appena arriva. La consegna dei messaggi è gestita da sottoscrizioni decentralizzate agli argomenti piuttosto che da regole centralizzate, e in genere non c'è un filtro incorporato basato sul contenuto dei messaggi.
- **Servizio di coda di messaggi:** Anche i servizi di accodamento di messaggi si concentrano sulla consegna affidabile dei messaggi, ma il loro modello di consegna dei messaggi è di tipo *pull*. Ciò significa che i consumatori devono richiedere nuovi messaggi a intervalli regolari. Questo ha il vantaggio di consentire ai servizi di consumare i messaggi coerentemente al proprio ritmo di lavoro, anche in caso di temporanee disconnessioni o fallimenti del sistema.
- **Piattaforme di streaming di eventi:** piattaforme di questo tipo possono gestire volumi molto elevati di eventi in tempo reale, consentendo al sistema di consumare e analizzare questi eventi simultaneamente.
- **Piattaforme di processazione di eventi:** simili alle piattaforme di streaming di eventi, permettono di elaborare gli eventi in maniera complessa, consentendo di definire regole, triggers e operazioni da eseguire quando si verificano determinati eventi o sequenze di eventi.

Le architetture event-driven sono quindi ampiamente utilizzate in una varietà di settori in cui viene richiesta una reattività in tempo reale, flessibilità, scalabilità e la capacità a trattare ed elaborare grandi volumi di eventi. I casi d'uso principali possono essere:

- Interfacce grafiche di uno user come i click di un bottone o i movimenti di un mouse
- La creazione di pipelines di dati in tempo reali per alimentare delle basi di dati di tipo Warehouse per le analisi BI
- Funzionalità in tempo reale come i servizi di notifica, nel caso in cui si verifichino eventi specifici, come la ricezione di un messaggio

- La gestione delle comunicazioni dei dispositivi IoT che generano un flusso continuo di eventi che devono essere elaborati in tempo reale
- La manutenzione predittiva, generando un flusso costante di eventi alle condizioni operative di un macchinario

Per ognuna delle tipologie di orchestratori di eventi esistono differenti soluzioni sul mercato. Il documento si focalizzerà sulle due attualmente più diffuse: MQTT, un servizio di coda di messaggi per l'industria 4.0 e Apache Kafka, una piattaforma di streaming di eventi.

### 6.2.1. MQTT: un protocollo di comunicazione specifico per il settore manifatturiero

MQTT (MQ Telemetry Transport) è un protocollo standard OASIS di trasporto di messaggistica pub/sub client-server (Andrew Banks et al., 2019), progettato per le connessioni con siti remoti che hanno dispositivi con vincoli di risorse o larghezza di banda di rete limitata. MQTT, infatti, è stato originariamente sviluppato per monitorare le pipelines di petrolio e gas su reti satellitari, ricercando una semplicità di implementazione, una consegna di dati con differenti qualità di servizio (QoS), leggerezza ed efficienza in termini di larghezza di banda e un'osservabilità sulla sessione di connessione.

Queste caratteristiche lo rendono ideale per l'uso in molte situazioni, compresi gli ambienti con vincoli, come la comunicazione in contesti Machine to Machine (M2M) e Internet of Things (IoT), dove è richiesto un ingombro di codice ridotto e/o la larghezza di banda della rete molto limitata (Andrew Banks et al., 2019).

Dal 2014 il protocollo è stato standardizzato da OASIS rendendolo aperto e neutrale a fornitori.

MQTT è descritto come "leggero", in altre parole è progettato per essere semplice da implementare e utilizzare, efficiente e non impattante dal punto di vista delle risorse utilizzato. Esso, infatti, utilizza il formato binario proprio per ridurre le dimensioni dei messaggi e aumentare l'efficienza della comunicazione. Viene inoltre minimizzata la quantità di dati da trasmettere e riduce la potenza di elaborazione necessaria per interpretare i messaggi.

In secondo luogo, MQTT è estremamente facile da implementare in quanto rispetto ad altri protocolli ha un'impronta di codice ridotta con un basso *overhead* e un conseguente utilizzo più efficiente delle risorse di rete. L'*overhead* di un messaggio indica il costo in termini di tempo e di bytes per generare un dato o trasmettere una determinata quantità di dati tra due componenti di una rete.

MQTT implementa tre livelli di servizio (Quality of Service, QoS) al fine di offrire differenti livelli di affidabilità per la consegna dei messaggi:

- **QoS 0:** permette una consegna al più una volta, non garantendo quindi la consegna del messaggio e adattandosi quindi a use cases in cui vengono scambiati grandi volumi di dati e la perdita di dati non è critica.
- **QoS 1:** permette la consegna almeno una volta, ma in caso di errori viene consegnato più volte. Il broker attende la conferma di ricezione del messaggio per un tempo stabilito prima di inviare nuovamente il messaggio.
- **QoS 2:** permette la consegna esattamente una volta, garantendo che i messaggi vengano confermati e reinviati fino a corretta ricezione da parte del destinatario. Questo livello di servizio è più impattante in termini di performance, di risorse utilizzate e di traffico di rete rispetto ai livelli inferiori.

Da un punto di vista di persistenza, ossia della possibilità di un client di riconnettersi al broker in caso di guasto o disconnessione imprevista e recuperare messaggi precedenti, vi sono tre differenti possibilità di implementazione:

- **Non persistenza:** per default, i messaggi non sono stoccati nel server e sono persi in caso di guasto.
- **Coda persistente:** in questa modalità, il server stocca i messaggi finché non sono consegnati al client. Nel caso in cui il client non è disponibile, i messaggi sono messi in coda fino a successiva connessione.
- **Persistenza con conferma di ricezione:** in questa modalità i messaggi sono stoccati fino a che il client conferma la corretta ricezione di un messaggio.

In maniera analoga ai livelli di servizio, all'aumentare del livello di persistenza, si va ad impattare le performance del sistema.



Vi sono, infine, differenti possibilità in termini di sicurezza e protezione dagli attacchi informatici. Di default, MQTT utilizza il protocollo di trasporto TCP, non criptato. È possibile, tuttavia, implementare dei meccanismi di autenticazione come OAuth2.0 o delle politiche di autorizzazione per i client al fine di limitare l'abbonamento a certi topic critici.

I messaggi MQTT sono composti da diverse parti che consentono di trasportare un contenuto e le informazioni necessarie per la sua consegna.

Il *fixed header* contiene informazioni sull'interpretazione del messaggio, definisce il livello di servizio e di persistenza desiderato.

Il *variable header* è presente solo in alcuni casi per fornire informazioni aggiuntive per la gestione del messaggio, come ad esempio il topic associato.

Il *payload* è il contenuto effettivo del messaggio costituito dai veri e propri dati.

Un esempio possibile in un contesto industriale potrebbe essere la rilevazione della temperatura in una cella di produzione. In questo caso si avrà un topic di questo tipo:

*stabilimento X / linea Y / temperatura*

Il payload sarà la temperatura rilevata.

Il messaggio sarà strutturato nella seguente maniera:

- Fixed header:
  - Tipologia di messaggio: PUBLISH
  - QoS: 1 (almeno una volta)
  - Flag retain: *true*. Il broker conserva il messaggio per i nuovi abbonati al topic.
- Variable header: *stabilimento X / linea Y / temperatura*
- Payload: 23.5 °C

#### 6.2.1.1. Confronto MQTT e HTTP

È possibile confrontare i protocolli MQTT e HTTP attraverso differenti misure di performance. È necessario, tuttavia, introdurre i principali comandi per ognuno di essi:

- Un MQTT client può eseguire cinque comandi principali:
  - CONNECT, per instaurare una connessione,
  - SUBSCRIBE per sottoscrivere un topic,

- PUBLISH, per pubblicare un evento,
- UNSUBSCRIBE per disiscriversi a un topic,
- DISCONNECT per disconnettersi da un event broker.
- I client HTTP, invece, possono invocare richieste di tipo GET, PUT, DELETE e POST. Non si ha quindi un’instaurazione di una connessione continua come nel caso di MQTT.

Da un punto di vista del volume degli overhead in termini di bytes, deve essere di almeno 2 bytes e il cui contenuto può essere formattato in codice binario, mentre il secondo ha un overhead di almeno 8 bytes, il cui header è in formato testo. Prendendo come esempio uno scambio di messaggi tra dispositivi IoT si può comparare il volume in bytes dell’overhead di messaggio TCP a seconda del numero di messaggi inviati. In Tabella 2 vengono forniti alcuni esempi:

	<b>Bytes MQTT</b>	<b>Bytes HTTP</b>
<i>Stabilimento connessione</i>	5572	2261
<i>Disconnessione</i>	376 (opzionale)	0
<i>Per ogni messaggio pubblicato</i>	388	3285
<i>Somma per 1 messaggio</i>	6336	5546
<i>Somma per 10 messaggi</i>	9829	55460
<i>Somma per 100 messaggi</i>	44748	554600

*Tabella 2 Confronto MQTT-HTTP dei volumi (bytes) per numero di messaggi inviati*

MQTT al completamento della connessione TCP invia dei pacchetti di messaggi *CONNECT* per stabilire e mantenere la connessione. Per questa ragione, l’instaurazione di una connessione tra un client e un broker risulta più pesante in termini di bytes per MQTT.

L’equivalente dell’“overhead” di MQTT per HTTP deve essere incluso in ogni richiesta, ragione per la quale l’overhead per messaggio è molto più grande.

Non essendoci una connessione continua tra i client con HTTP, l’overhead di disconnessione è sempre pari a zero mentre per MQTT può risultare pari a 0, in quanto non obbligatorio inviare il pacchetto DISCONNECT per chiudere la connessione TCP.

In secondo luogo, si può andare a misurare le performance in termini di tempo medio di risposta (ms) per una singola connessione TCP. In Tabella 3 viene fatto un confronto in base al numero di messaggi in un ciclo di connessione:

<b>Numero di messaggi in un ciclo di connessione</b>	<b>Risposta media in ms per messaggio MQTT (QoS 1)</b>	<b>Risposta media per messaggio HTTP</b>
1	113	289
10	47	289
100	43	289

*Tabella 3 Confronto tempo medio di risposta MQTT-HTTP*

A differenza di HTTP, non essendo necessario inviare nuovamente l'overhead di configurazione in MQTT, il tempo medio di risposta viene ridotto progressivamente all'aumentare del numero di messaggi inviati.

L'esempio descritto è rappresentativo nel caso di connessioni con QoS (Quality of Service) 1. Si può quindi ipotizzare un miglioramento delle performance per il livello QoS 0.

In Tabella 4 è rappresentato il confronto in termini di throughput e tempi di risposta per QoS 1 e QoS 0:

<b>Numero di messaggi</b>	<b>QoS MQTT</b>	<b>Messaggi / s</b>	<b>Tempo medio di risposta (ms)</b>
10000	1	1234	0,81
10000	0	18416	0,054

*Tabella 4 Confronto performance MQTT in base al livello di servizio (QoS)*

Questi tre confronti sono stati effettuati attraverso un broker MQTT HiveMQ e una connessione a un dispositivo IoT fittizio.

### 6.2.2. L'estensione Sparkplug B per la standardizzazione di MQTT

Essendo MQTT molto aperto, viene lasciata grande libertà agli sviluppatori nella sua implementazione: ad esempio, non definisce la denominazione dei topic o la struttura dei messaggi. Questo può portare a incoerenze in caso di collaborazione tra soggetti diversi che non applicano le stesse regole di comunicazione. Sparkplug B è una specifica open-source che formalizza la struttura degli argomenti e il loro contenuto al fine di migliorare l'interoperabilità di MQTT. Essa si basa su diversi principi chiave di progettazione (Sparkplug Specification Project Team, n.d.):

- **Definizione** di un topic *namespace*: in modo tale da fornire una semantica che consenta la scoperta automatica e la comunicazione bidirezionale tra i client MQTT di un sistema. Il concetto di namespace verrà sviluppato nel capitolo successivo.
- Gestione delle **condizioni delle attrezzature**: permette di sfruttare le potenzialità di MQTT sulle capacità di “Continuous Session Awareness” per seguire lo stato e le condizioni dei dispositivi industriali.
- **Strutturazione del messaggio per l’industria manifatturiera**: vengono definiti i meccanismi di codifica del payload sulla base del namespace definito in precedenza.

In Figura 17 viene rappresentata l’infrastruttura logica industriale con MQTT e Sparkplug B.

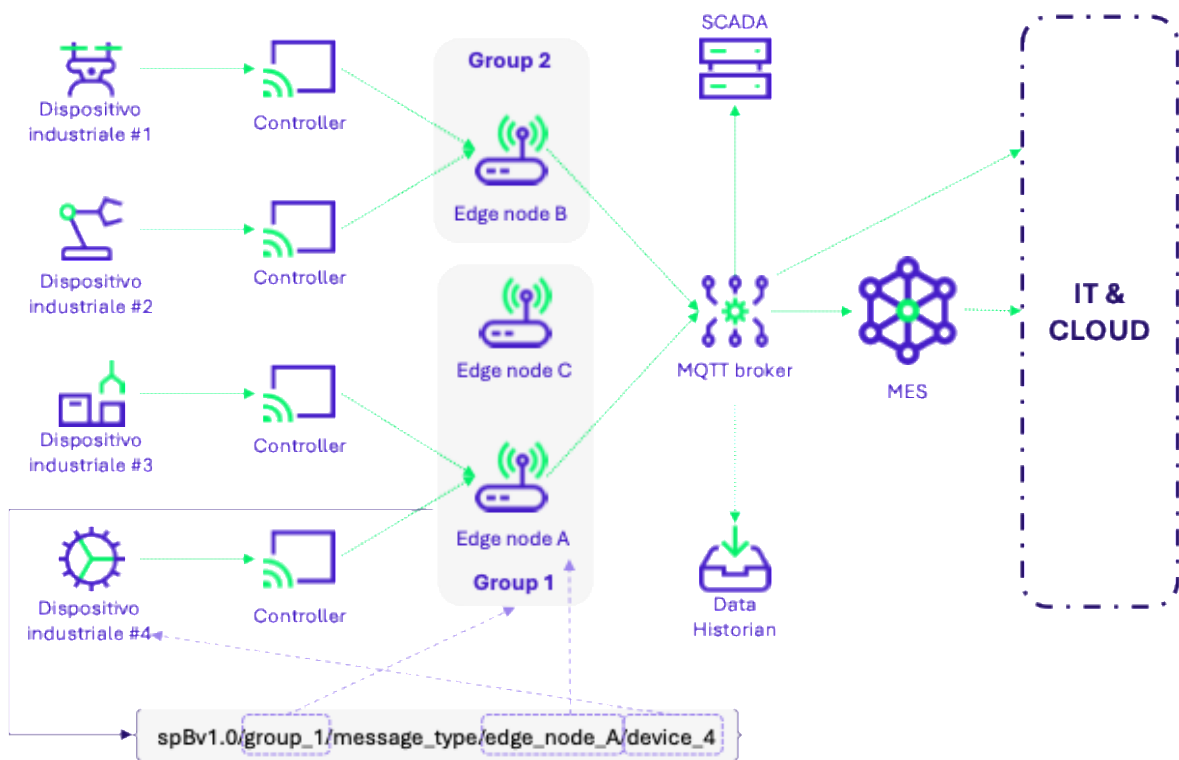


Figura 17 Infrastruttura logica industriale con MQTT e Sparkplug B

L’MQTT broker è posto al centro dell’infrastruttura e funge da intermediario tra i client che pubblicano messaggi e i client che hanno effettuato sottoscrizioni. Esso deve essere compatibile con i requisiti delle infrastrutture Sparkplug, che aggiungono lo schema di gestione dello stato e della connessione. Inoltre, esso deve essere dimensionato per adattarsi ai volumi di traffico. Possono essere, infatti, inseriti, più brokers a seconda delle use case.

Situati tra i dispositivi e il broker, gli *edge nodes* catturano le informazioni dai dispositivi IoT e non (PLC, RTU, Sensori...) e le trasferiscono al broker, agendo come client del broker MQTT e filtrando e amplificando il segnale.

Gli edge nodes sono concentrati in gruppi dagli architetti di sistema in base alla loro applicazione o in base alla loro distribuzione fisica nello stabilimento.

In Figura 18 viene anche rappresentato un esempio di messaggio. Il protocollo, infatti, definisce topic specifici e un formato per i messaggi che facilita la comunicazione standardizzata tra dispositivi industriali.

- **spBv1.0** indica la versione di Sparkplug utilizzata
- **group\_id** specifica il gruppo di appartenenza dell'edge node che ha inviato il messaggio
- **message\_type** indica il tipo di messaggio. Esso può essere:
  - *NBIRTH* per le connessioni al broker MQTT
  - *NDEATH* per le disconnessioni
  - *DBIRTH* per la segnalazione di una connessione di un dispositivo
  - *DDEATH* per la segnalazione di una disconnessione di un dispositivo
  - *DATA* per l'aggiornamento dei dati provenienti da un dispositivo
- **edge\_node\_id** specifica l'identificativo del'edge node
- **device\_id** specifica l'identificativo del dispositivo

### 6.2.3. L'UNS (Unified NameSpace): un'unica sorgente di verità nei sistemi informativi

Come citato in precedenza, uno *Unified NameSpace* è un concetto fondamentale nelle architetture industriali moderne e sempre più diffuso e introdotto da protocolli come Sparkplug B. Esso mira a centralizzare tutti i dati e gli eventi rilevanti di un'azienda in una struttura semantica unificata e standardizzata. L'obiettivo di questo approccio è quello di avere una visione unica e coerente delle informazioni nella totalità dell'azienda centralizzando e standardizzando l'accesso ai dati provenienti da varie fonti, per facilitare l'integrazione, la collaborazione e il processo decisionale.

In Figura 18 viene rappresentato un esempio di struttura UNS sulla base dello standard Sparkplug B in un settore industriale:

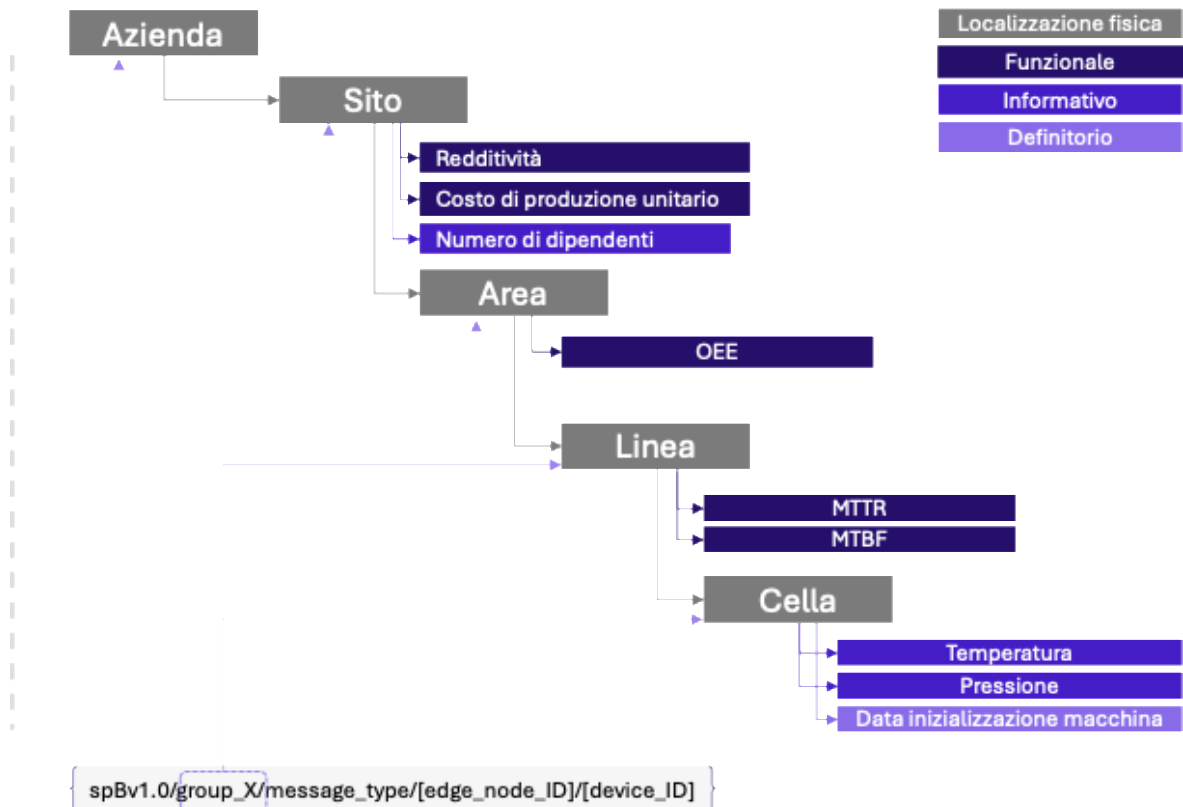


Figura 18 Esempio struttura UNS

Esistono 4 tipologie di spazi semantici:

- **Funzionale:** si riferisce ai parametri dei dati in un sistema industriale in base alla loro funzione. Ciascun parametro è raggruppato in base al compito specifico che svolge e non in base alla sua posizione fisica o alla rete. Nell'esempio in figura i dati come OEE (Overall Equipment Efficiency) o l'MTTR (Mean Time To Repair) saranno raggruppati in un namespace industriale mentre dati come la redditività di un sito produttivo o il costo di produzione industriale sono raggruppati in namespace finanziari/economici.
- **Informativo:** organizza i dati in base al loro contenuto informativo. Questi namespaces sono principalmente utilizzati per il consumo da parte di database, che, seguendo l'esempio in figura, raggruppano tutti i dati di temperatura o pressione in singoli gruppi. In questo modo viene semplificata la presa di decisione e l'analisi preventiva.
- **Definitorio:** organizza i parametri in base ai loro attributi, come il tipo, una condizione o una dimensione. Questi attributi persistono nel tempo e permettono di

visualizzare coerentemente in un'unica posizione dei parametri per confronti efficaci e semplicemente realizzabili.

- **Ad hoc:** si tratta di uno spazio temporaneo per l'organizzazione dei dati in fase di sviluppo dell'UNS. In alcuni casi può essere opportuno creare degli spazi ad hoc per la risoluzione di problematiche specifiche, andando a raggruppare dei parametri eterogenei.

In sintesi, i vantaggi dell'implementazione di un UNS sono la visibilità centralizzata e standardizzazione di tutti i dati, la facile integrazione di sistemi ed applicazioni attraverso un'interfaccia comune per l'accesso ai dati, e la scalabilità grazie una migliore capacità a adattarsi al cambiamento.

Al fine di implementare uno spazio semantico è necessario avere team di data governance specifici per la gestione e aggiornamento di ciascuno spazio.

### 6.3. Apache Kafka e le piattaforme di streaming dei dati

Un'alternativa alla gestione delle architetture di eventi è l'implementazione di una piattaforma di streaming. Come descritto in precedenza, si basano sul servizio pub/sub, fornendo funzionalità aggiuntive: le piattaforme di streaming di eventi come Kafka possono raccogliere, elaborare e distribuire grandi quantità di eventi in tempo reale. Come nel caso di MQTT esse sono composte da produttori e consumatori di eventi disaccoppiati e da una piattaforma centrale con funzionalità supplementari. I consumatori allo stesso modo possono sottoscrivere a specifici flussi di eventi, detti *stream*, che vengono processati in modo indipendente e asincrono.

In Figura 19 è rappresentata l'infrastruttura di una piattaforma di streaming di eventi dei suoi componenti principali. I concetti di client produttore, consumatore e di topic sono già stati introdotti precedentemente. Kafka a differenza di MQTT crea dei cluster di broker, ossia dei raggruppamenti fisici per garantire scalabilità, un alto livello di disponibilità, tolleranza ai guasti e il bilanciamento del carico. L'obiettivo è quindi di permettere la persistenza e la replicazione dei dati.

I consumatori di eventi sono raggruppati in *consumer groups* che collaborano per elaborare eventi da un singolo topic, permettendo di suddividere il carico di lavoro e migliorando la scalabilità del sistema.

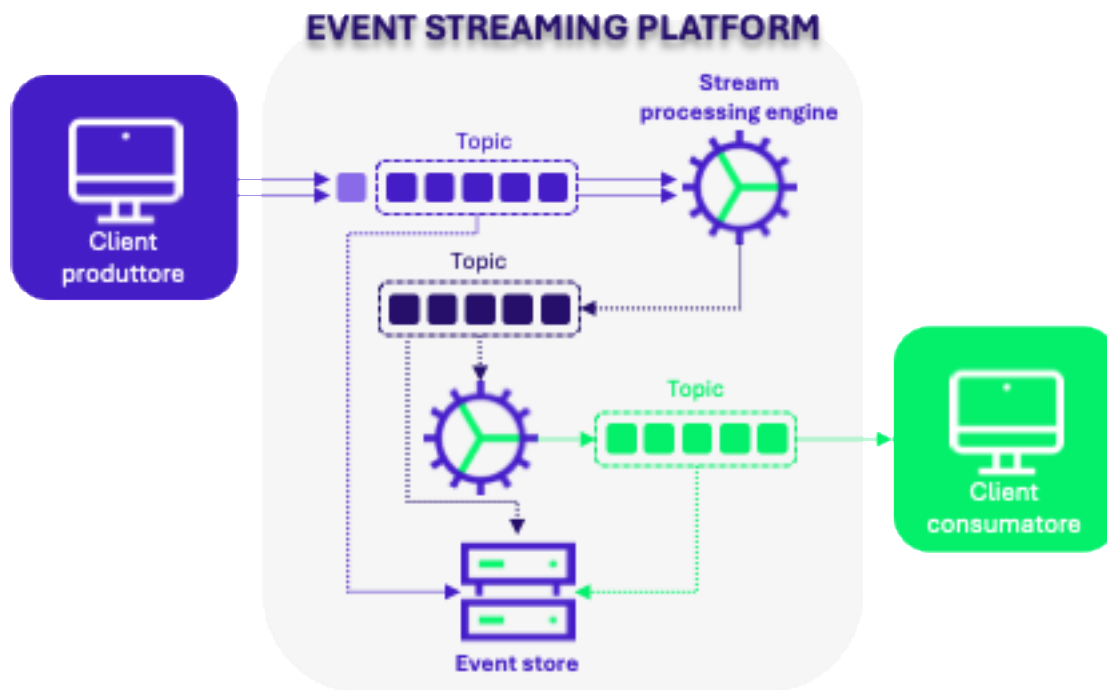


Figura 19 Infrastruttura di una piattaforma di streaming di eventi

A differenza di MQTT, la gestione e creazione dei topic viene fatta internamente alla piattaforma con la possibilità di creare code. Il motore di elaborazione del flusso permette di applicare trasformazioni, aggregazioni o regole di trattamento dei flussi di eventi in tempo reale.

Con Apache Kafka vengono infatti create delle *partizioni*. Ogni partizione è una porzione del flusso di eventi associato a un topic, secondo le regole definite dal motore di elaborazione del flusso, che consente di elaborare parallelamente degli eventi e migliorare la scalabilità. Per aumentare ulteriormente il livello di disponibilità, è possibile creare delle repliche delle partizioni su più cluster. Questo assicura inoltre che i dati non vengano persi in caso di guasto di un componente. Per ogni partizione viene definita una replica *leader* che gestisce le richieste per la partizione, mentre le replica *follower* copiano i dati direttamente da quella



*leader*. Nel caso in cui quest'ultimo si guasti, una delle repliche follower assume il ruolo di leader.

All'interno della piattaforma è presente, inoltre, un archivio di eventi, progettato per conservare le registrazioni immutabili dei cambiamenti di stato di un sistema (Quix, n.d.). Gli eventi vengono memorizzati in ordine cronologico e, a differenza delle basi di dati tradizionali, comporta la ricostruzione di uno stato attraverso la riproduzione di eventi piuttosto che il mantenimento e l'aggiornamento dello stesso.

I vantaggi principali di Apache Kafka sono la possibilità di processare dei grandi volumi di dati attraverso un alto throughput. Grazie all'infrastruttura altamente distribuita, attraverso i cluster e le partizioni di topic, assicura una grande affidabilità e tolleranza ai guasti, consentendo di mantenere i flussi di dati in continuo anche se un componente si blocca. Per le stesse ragioni, Kafka assicura una grande scalabilità e interoperabilità con sistemi differenti, oltre a una comunicazione in tempo reale.

#### 6.4. MQTT e Kafka: differenze e casi d'uso per un utilizzo complementare

Apparentemente MQTT e Kafka possono essere considerati come due alternative alla gestione delle comunicazioni in un'architettura EDA. Infatti, entrambi permettono l'invio di dati basandosi su un modello pub/sub con un event broker, si inseriscono in un ambiente

altamente distribuito come i microservizi o i dispositivi industriali e sono altamente affidabili garantendo alte performance di trasferimento dei dati e livelli di disponibilità.

Nella Tabella 5 sono riassunte le principali differenze tra MQTT e Kafka:

	<b>MQTT</b>	<b>Apache Kafka</b>
<i>Throughput e latenza</i>	Adatto a ambienti industriali dove la consegna rapida dei messaggi è critica	Alti livelli di throughput e adatto ad ambienti che processano alti volumi di dati
<i>Stoccaggio dei dati</i>	Limitate opzioni di stoccaggio a lungo termine	Possibilità di inserire degli archivi di eventi di dimensioni configurabili
<i>Tolleranza ai guasti</i>	Differenti meccanismi di gestione dei livelli di servizio ma limitati in termini di replicazione a lungo termine	Grande persistenza del dato e possibilità di riprodurre gli eventi memorizzati
<i>Implementazione</i>	Configurazione relativamente semplice e standardizzata	Configurazione altamente complessa a causa della necessità di impostare i cluster e pianificare i flussi di partizioni e repliche
<i>Casi d'uso</i>	Ideato per gli ambienti industriali e per le comunicazioni tra dispositivi IoT	Progettata specificatamente per lo streaming ed elaborazione di grandi quantità di dati in tempo reale come nel caso di analisi, aggregazione ed elaborazione di big data tramite IA e ML

Tabella 5 Confronto MQTT e Apache Kafka

Nel complesso si può osservare come entrambi offrano le stesse funzionalità generali ma a livelli differenti, rendendo ciascuno dei due adatto a casi d'uso differenti. Risulta quindi ipotizzabile un utilizzo simultaneo andando a sfruttare la complementarità delle funzionalità di ciascuno.

In Appendice 6 viene rappresentato un esempio di infrastruttura integrata con MQTT e Apache Kafka. Il primo, per le sue caratteristiche, viene inserito nei livelli OT per la comunicazione con i dispositivi industriali mentre il secondo nei livelli IT andando a comunicare con le applicazioni aziendali di livello più alto.

I dispositivi IoT pubblicano i dati tramite gli *edge nodes* su un broker MQTT che definisce i topic di primo livello nei namespace industriali. Un bridge MQTT-Kafka viene utilizzato per trasferire i dati dai broker MQTT a Kafka. Quest'ultimo possiede un connettore nativo, Kafka Connect, per semplificare questo passaggio. Allo stesso modo alcuni MQTT brokers

come HiveMQ possiedono funzionalità di integrazione con Apache Kafka. Una volta i dati pubblicati su Kafka, il motore di elaborazione dei flussi, analizza e aggrega i topic andando a inserirli nei namespace aziendali, memorizza le partizioni e le distribuisce ai consumer groups.

Attraverso l'utilizzo di un modello di questo tipo si può facilmente ottenere un flusso di dati bidirezionale che consente un'elaborazione dei dati industriale in tempo reale per facilitare e velocizzare la presa di decisione ai livelli più alti.

## 6.5. Le piattaforme CPS a supporto della sicurezza informatica dei SI

Fino ad ora si è parlato di efficienza, performance delle comunicazioni e scalabilità, tuttavia, una dei più grandi punti di attenzione nello sviluppo e manutenzione delle architetture dei SI è la sicurezza, specie ai livelli OT. Secondo uno studio di McKinsey, l'investimento nella sicurezza dei dispositivi fisici è aumentato del 12% annuo a causa delle crescenti preoccupazioni legate alla sicurezza informatica. I protocolli e le soluzioni approfondite nei capitoli precedenti hanno delle funzionalità integrate di criptaggio e autenticazione ma non vi è la possibilità di avere un'osservabilità centralizzata delle condizioni di sicurezza di tutti i componenti dell'ambiente industriale.

Le piattaforme CPS (Cyber-Physical Systems) sono definite secondo Gartner come “sistemi ingegnerizzati che orchestrano rilevamento, calcolo, controllo, rete e analisi per interagire con il mondo fisico (compresi gli esseri umani).”. In altre parole, sono prodotti e servizi che utilizzano la conoscenza dei protocolli utilizzati, i pacchetti di rete operativi/produttivi o i metadati del traffico e il comportamento degli asset dei processi fisici per scoprire, classificare, mappare e proteggere i componenti fisici in ambienti di industriali o critici al di fuori degli ambienti IT aziendali.

Nel dettaglio gli attributi di queste piattaforme includono:

- Visibilità e categorizzazione degli asset
- Mappatura dettagliata degli asset
- Supporto per protocolli industriali proprietari
- Diagrammi di rete dettagliati e flussi di dati
- Informazioni sulle vulnerabilità

- Gestione delle informazioni sulle minacce
- Integrazione con strumenti di sicurezza IT per la reattività in caso di rischi e pericoli emergenti

Il loro funzionamento si fonda su due componenti principali: i sensori per la rilevazione delle condizioni degli asset e una piattaforma per la mappatura di questi ultimi e la gestione delle politiche di sicurezza.

## 6.6. Le piattaforme IIoT a confronto con i sistemi informatici monolitici più diffusi

Un altro componente fondamentale nelle architetture dei SI moderne sono le piattaforme IIoT. Esse offrono la possibilità di semplificare l'integrazione dell'OT con l'IT comunicando con dei gateway IIoT per accedere direttamente alle fonti di dati (dispositivi IoT). L'obiettivo è di migliorare il processo decisionale, la visibilità operativa e il controllo degli impianti. Fornendo possibilità di elaborazione degli eventi, distribuisce i carichi di lavoro (ad esempio, interfaccia ML, analisi dei flussi) sull'edge gateway. Il gateway esegue questi carichi di lavoro e trasmette i dati alla piattaforma IIoT.

Apparentemente questa tipologia di piattaforma fornisce funzionalità simili a quelle dei MES, tuttavia differiscono in termini di approccio e focus operativo. Questi ultimi da un punto di vista di applicazione operano su un livello più generale per gestire e controllare i processi di fabbricazione, mentre le seconde si focalizzano sulla connettività industriale e sull'efficienza e rapidità di trasferimento e trattamento dei dati. Da un punto di vista di analisi dei dati le piattaforme IIoT possono integrare possibilità di analisi avanzata tramite IA e ML per la manutenzione predittiva mentre i MES si focalizzano sul monitoraggio e la reperistica dei dati operativi. Infine, per quanto riguarda l'elaborazione dei dati, le piattaforme IIoT si basano su un approccio di edge computing, che avvicina le applicazioni aziendali alle fonti di dati andando a trattare questi ultimi localmente per poi inviare le informazioni più pertinenti una volta elaborate verso i livelli superiori.

Le piattaforme IIoT, in sintesi, offrono maggiore flessibilità e capacità analitiche rispetto ai MES, grazie alla loro capacità di integrare una vasta gamma di dispositivi, di supportare l'elaborazione in tempo reale e di utilizzare tecnologie di intelligenza artificiale per l'analisi predittiva. Tuttavia, i MES rimangono centrali per la gestione dei processi di produzione, fornendo strumenti specifici per l'ottimizzazione delle linee di produzione e la gestione degli

ordini. E quindi ipotizzabile un utilizzo in parallelo per sfruttare appieno le funzionalità e la complementarità di entrambi.

## 6.7. I data hubs: una soluzione innovativa per la gestione e centralizzazione del dato

Da un punto di vista di gestione dei dati, vi è un interesse sempre maggiore per integrare i Data Lake specializzati nella gestione di grandi volumi di dati e ottimizzati per le implementazioni di soluzioni di IA e ML a dei Data Warehouse, più adatti alla reportistica. Inoltre, l'evoluzione delle architetture informatiche ha reso sempre più eterogenei i sistemi complicando la data governance. Infatti, secondo Gartner, circa il 90% delle aziende sperimenta difficoltà con l'integrazione dei dati provenienti da fonti disparate.

Per questi motivi i Data Hub sono delle infrastrutture moderne sempre più diffuse che consentono la raccolta, gestione, integrazione e distribuzione di grandi volumi di dati provenienti da fonti eterogenee. L'interconnessione e l'interscambio di dati tra sistemi diversi viene facilitato, promuovendo la governance, la sicurezza e la scalabilità dei dati.

In Figura 20 viene rappresentata l'infrastruttura e i componenti di un Data Hub.

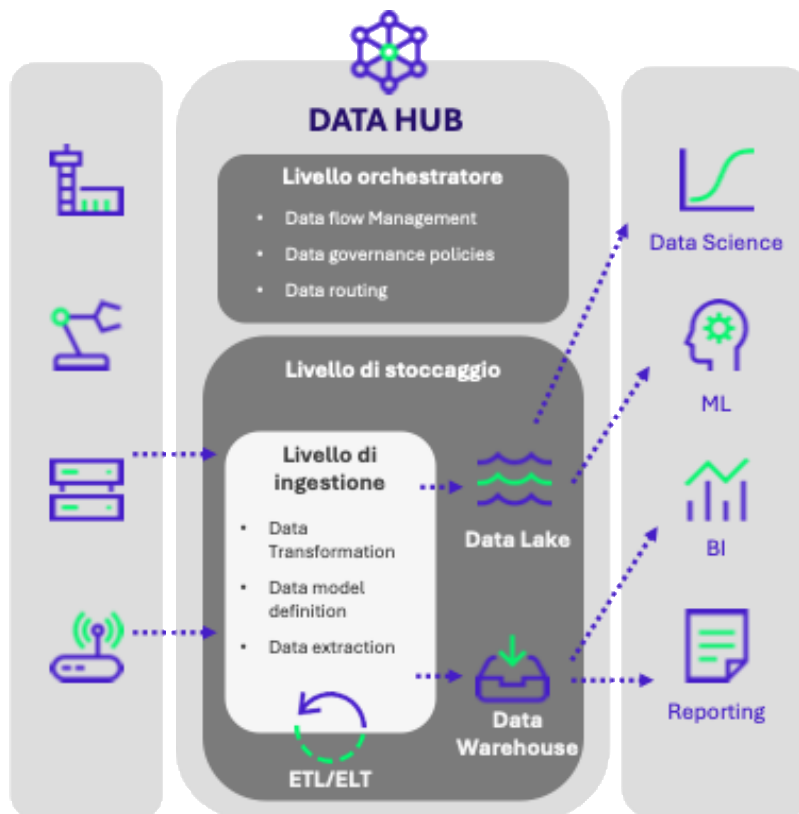


Figura 20 Struttura e componenti di un Data Hub

A differenza delle classiche basi di dati presentate nei capitoli precedenti, offre differenti funzionalità suddivise in tre livelli principali. Un primo livello di “ingestione” con funzionalità di ETL ed ELT a seconda del bisogno per estrarre i dati, strutturarli e trasformarli per poi inviarli verso il livello di stoccaggio. Quest’ultimo può essere rappresentato da un Data LakeHouse, che integra in un'unica soluzione le caratteristiche dei Data Lake e dei Data Warehouse. Questi due livelli sono supervisionati e controllati dal livello orchestratore che svolge funzionalità di data routing, governance e gestione dei flussi e del traffico.

## 7. TOM generalizzato di integrazione delle ultime evoluzioni

Una volta identificate e approfondite le principali evoluzioni nel campo del livello applicativo delle architetture dei SI è possibile andare a realizzare un TOM ideale, ossia una rappresentazione di una visione strutturata e strategica di come un'organizzazione intende evolversi e integrare nuove tecnologie e soluzioni informatiche. L'obiettivo è di andare ad allineare l'infrastruttura IT con gli obiettivi aziendali, garantendo che le risorse tecnologiche supportino in modo efficiente le priorità strategiche dell'organizzazione.

Un primo indicatore da osservare è l'efficienza operativa, vi deve essere un miglioramento della gestione delle risorse, una riduzione dei costi ed un'automatizzazione dei processi ripetitivi. In secondo luogo, l'infrastruttura ideata deve consentire che i sistemi e le piattaforme implementate siano sufficientemente flessibili per adattarsi ai cambiamenti delle esigenze aziendali ma anche delle evoluzioni tecnologiche. Infine, è necessario che vi sia un miglioramento della sicurezza informativa e una conformità con le normative.

### 7.1. L'implementazione agile attraverso DevOps e DevSecOps

Al fine di poter implementare correttamente, rapidamente e in maniera affidabile le infrastrutture ideate nel TOM è consigliato utilizzare degli approcci evoluti e agili come DevOps e DevSecOps.

Il primo è particolarmente diffuso nello sviluppo di software ed è spesso accostato da metodologie di gestione dei progetti di tipo *agile*. Esso si riferisce ad una metodologia di gestione collaborativa dei dati incentrato sulla promozione della comunicazione tra i team di sviluppo (Dev) e le operazioni IT (Ops). L'obiettivo è il miglioramento della velocità e qualità del rilascio, la riduzione del ciclo di vita di sviluppo e una maggiore efficienza operativa. Il suo ciclo di vita, rappresentato in Figura 21, è una serie di processi di sviluppo iterativi e automatizzati, o flussi di lavoro, eseguiti come parte di un ciclo di sviluppo più ampio, automatizzato e iterativo.

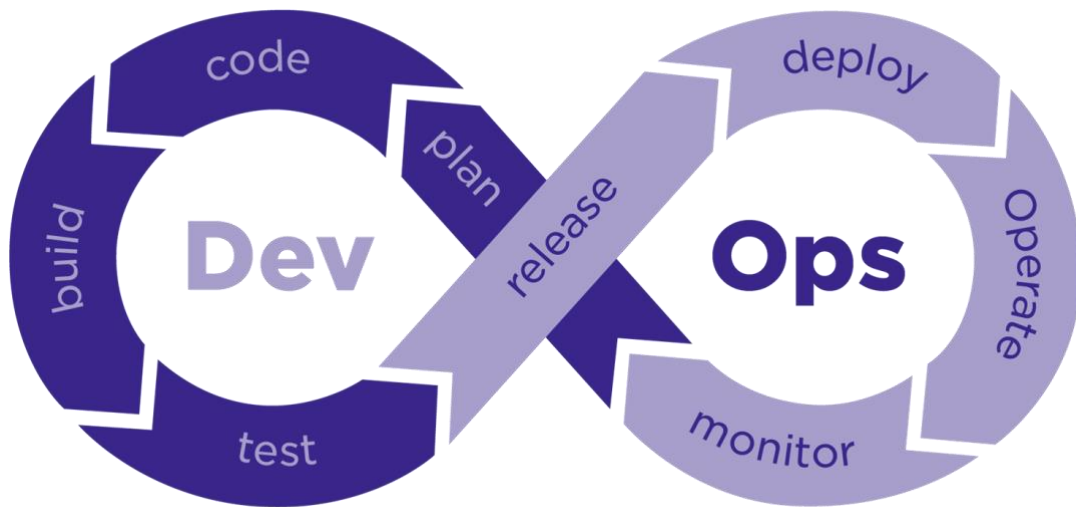


Figura 21 Ciclo di vita DevOps. (Fonte: italiancoders)

I principi chiave di DevOps sono i seguenti:

- **Integrazione continua** (Continuous Integration – CI): processo di automazione per gli sviluppatori che facilita il raggruppamento delle modifiche al codice in un unico repository centrale.
- **Implementazione continua** (Continuous Deployment – CD): estende il concetto di CI, includendo la distribuzione di piccole modifiche al codice in produzione come parte di un processo di routine e frequente.
- **Infrastruttura come codice** (Infrastructure as a Code – IaC): permette la gestione e la configurazione delle infrastrutture IT attraverso codice, attraverso file di configurazione testabili e versionabili.
- **Collaborazione e comunicazione continua**: promozione della collaborazione tra i differenti team di sviluppo attraverso strumenti e pratiche condivise.
- **Monitoraggio e feedback continui**: lo sviluppo delle applicazioni viene monitorato costantemente, attraverso test di performance continui e raccoglimento di feedback e eventuali bug.

La metodologia DevSecOps integra all'interno del ciclo di vita del modello di DevOps la sicurezza (Sec) fin dalle prime fasi di sviluppo, garantendo che ogni componente software venga testato e validato non solo dal punto di vista funzionale ma anche dal punto di vista della robustezza in caso di attacchi informatici. Vengono, infatti, automatizzati i processi di



analisi delle vulnerabilità nelle pipeline CI/CD, migliorando la copertura e la reattività alle minacce. Verifiche del tipo *penetration test* vengono eseguite regolarmente al fine di identificare si dalle prime fasi eventuali vulnerabilità.

## 7.2. La gestione del legacy: una sfida cruciale nel processo di trasformazione digitale

Un ulteriore fattore chiave nell'implementazione di una nuova visione di infrastruttura informatica è la gestione del legacy. Differenti soluzioni possono essere ipotizzate al fine di ridurre l'impatto da un punto di vista di cambiamento e accettazione delle nuove pratiche ma anche per semplificare e velocizzare l'integrazione delle nuove tecnologie. Secondo uno studio di Wavestone (2024), esistono 5 principali strategie per modernizzare i sistemi legacy sul mainframe. La sfida principale non è solo la migrazione dei sistemi in linea con la visione aziendale e i requisiti ma anche continuare a far funzionare i sistemi originali. Le strategie comprovate per una modernizzazione e una migrazione efficiente e sicura si trovano nelle "5 R":

- **Ritirare** (Retire): le applicazioni che non sono più in linea con la visione aziendale devono essere ritirate dall'architettura, riducendo i costi operativi e di manutenzione.
- **Rimpiazzare** (Replace): le applicazioni che sono ancora necessarie ma ormai obsolete devono essere rimpiazzate da nuove soluzioni che possono essere sviluppate internamente o acquistate *on-shelf* a seconda degli interessi aziendali. In questo caso la più grande sfida sta nell'integrazione e interconnessione delle soluzioni scelte e nella migrazione dei dati.
- **Rilocalizzare** (Relocate): può essere opportuno rilocalizzare alcune applicazioni in nuovi ambienti operativi, come su delle piattaforme cloud.
- **Rifattorizzare/ricostruire** (Refactor/Rebuild): alcune applicazioni che rimangono chiave per il funzionamento del SI possono essere modificate e riqualficate secondo i principi definiti in fase di redazione del TOM.
- **Mantenere** (Retain): può essere opportuno mantenere le applicazioni nel medio-lungo termine a seconda dei vincoli imposti.

## 7.3. Scenari progressivi di integrazione delle evoluzioni

Per concludere lo studio sono stati ipotizzati degli scenari di integrazione delle tecnologie descritte in maniera progressiva, partendo dalla mappatura logica individuata nello stato

dell'arte e andando gradualmente a adottare soluzioni a micro-servizi e un passaggio verso una gestione tramite service mesh in un primo momento, per poi dirigersi verso un cambiamento di approccio della visione di architettura con l'implementazione di sistemi come MQTT e Apache Kafka.

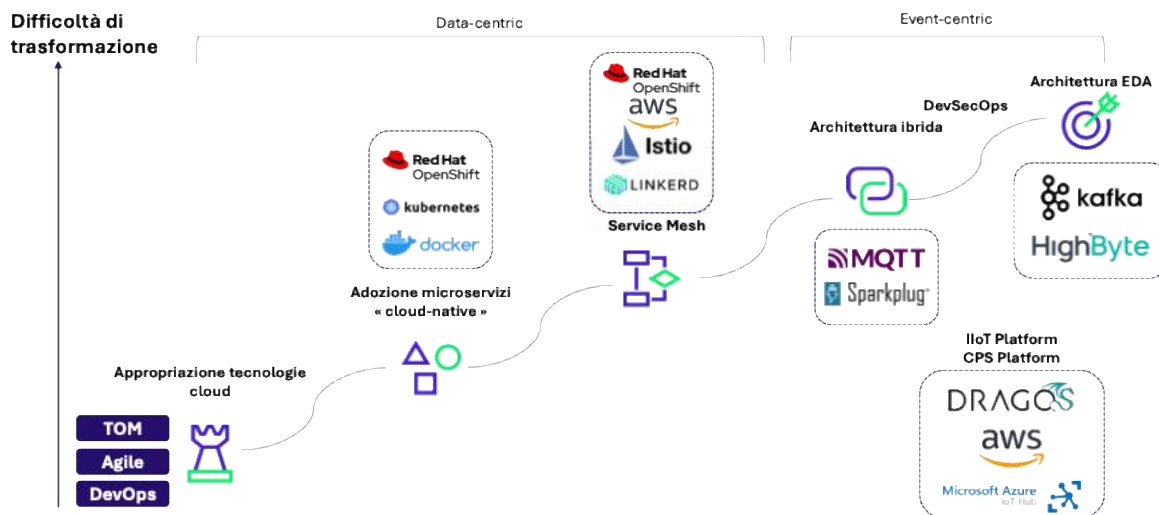


Figura 22 Percorso di trasformazione delle architetture logiche dei SI

In Figura 22 viene rappresentato il percorso di trasformazione ipotetico che un'azienda deve affrontare per modernizzare l'architettura logica del proprio SI. Vengono, inoltre, proposte alcune soluzioni disponibili sul mercato da poter integrare nel proprio ecosistema.

Di seguito vengono approfonditi i quattro scenari rappresentati in Appendice.

### 7.3.1. L'adozione dei microservizi “cloud-native”

In un primo momento è possibile andare a convertire o aggiungere dei microservizi cloud-native all'architettura SI. È stato ipotizzato di riprendere le stesse funzionalità individuate nello stato dell'arte e replicarle in raggruppamenti costituiti da Pod. Viene mantenuta una gestione della comunicazione tramite API e viene ipotizzata una gestione dei dati attraverso basi di dati distribuite per ciascun servizio. In Appendice 7 viene rappresentata la mappatura realizzata.

In sintesi, i vantaggi e le caratteristiche chiave sono le seguenti:

- **Autonomia:** i microservizi sono costituiti da singoli componenti di servizio accoppiati in modo lasco, che possono essere sviluppati, distribuiti, gestiti, modificati

e reimpiegati senza compromettere la funzione di altri servizi o l'integrità dell'applicazione.

- **Manutenibilità e testabilità:** le nuove funzionalità possono essere provate e annullate se non funzionano.
- **Organizzazione intorno alle funzionalità aziendali:** ogni servizio è progettato per risolvere un'esigenza aziendale specifica.
- **Facilità di distribuzione:** integrazione e consegna continua sulla base dell'approccio DevSecOps.
- **Libertà tecnologica:** ogni team può scegliere lo strumento migliore per risolvere problemi specifici.
- **Resilienza:** l'indipendenza di ogni servizio aumenta la resistenza dell'applicazione ai guasti.

I costi di implementazione per l'azienda sono relativamente alti a seconda della soluzione individuata sul mercato ed è necessaria un'impattante accompagnamento per la gestione del cambiamento.

### 7.3.2. L'implementazione di un service mesh per la gestione del traffico

In un secondo momento, è possibile ipotizzare l'inserimento di soluzione di service mesh per la gestione del traffico est-ovest. In Appendice 8 viene rappresentata la mappatura realizzata. Per questioni di semplicità di visualizzazione è stato inserito un solo *Control Plane* senza entrare nel dettaglio dei cluster che sono strettamente legati alle use case. Per le comunicazioni nord-sud è stata ipotizzata l'integrazione di un API gateway.

In sintesi, i vantaggi e le caratteristiche chiave sono le seguenti:

- **Centralizzazione delle transazioni:** consente agli sviluppatori di separare e gestire le comunicazioni da servizio a servizio in un livello di infrastruttura dedicato.
- **Osservabilità:** offre una visione chiara delle transazioni a livello di tutti i piani dati della rete per fornire l'osservabilità del comportamento del servizio in tempo reale.
- **Sicurezza interna:** crittografia delle comunicazioni e distribuzione delle politiche di sicurezza, comprese autenticazione e autorizzazione, dalla rete all'applicazione e ai singoli microservizi.

- **Specializzazione del traffico nord-sud ed est-ovest:** il servizio mesh è più adatto al traffico est-ovest (traffico interno all'azienda), mentre i gateway api sono più adatti al traffico nord-sud (connessione con i clienti esterni).
- **Gestione del traffico:** controllo preciso dell'instradamento delle richieste e del comportamento del traffico, distribuzione del traffico e bilanciamento del carico.

Questa seconda tappa risulta meno impattante per le aziende rispetto alla prima ma risulta, tuttavia, consigliabile da un punto di vista della sicurezza e delle performance nella gestione del traffico di rete.

### 7.3.3. Un'architettura ibrida per la convergenza IT/OT

In un terzo momento, si può ipotizzare un passaggio verso una tipologia di architettura ibrida. Ai livelli OT viene implementato un event broker MQTT per la gestione delle comunicazioni industriali, consentendo un primo passo verso le architetture EDA mentre al livello IT viene mantenuta la gestione tramite API gateways. In Appendice 9 viene rappresentata la mappatura realizzata.

In sintesi, i vantaggi e le caratteristiche chiave sono le seguenti:

- **Tecnologie specifiche per l'Industria 4.0:** il protocollo MQTT è stato progettato specificamente per ambienti con risorse limitate ed è ideale per comunicare con i dispositivi IoT industriali.
- **Prestazioni migliorate:** aumento del throughput, della velocità e del consumo di risorse.
- **Scalabilità:** progettata per adattarsi ai cambiamenti, sia che si tratti di aggiungere nuove fonti di dati, di modificare le strutture di dati esistenti o di cambiare le esigenze aziendali.
- **Interoperabilità e flessibilità:** facilita l'integrazione di vari sistemi e applicazioni fornendo un'interfaccia comune per l'accesso ai dati, riducendo così le incompatibilità e le ridondanze
- **Gestione in tempo reale:** l'architettura event-driven consente di attivare automaticamente le risposte quando si crea un evento specifico.

L'implementazione di una tecnologia di questo tipo può essere effettuata su anni a causa del forte impatto sul sistema informativo e del carattere dirompente del cambiamento di approccio nei confronti della gestione della sincronizzazione dei dati.

#### 7.3.4. Passaggio definitivo alle architetture ad eventi

Viene, infine, ipotizzato un passaggio definitivo verso un'architettura EDA attraverso l'inserimento nei livelli IT di una piattaforma di streaming di eventi come Apache Kafka. In Appendice 10 viene rappresentata la mappatura realizzata. Il funzionamento specifico dell'integrazione di Apache Kafka con MQTT segue i principi dettagliati nel capitolo 6.4. Viene, inoltre, ipotizzato abbandono della concezione di architettura definito da Purdue per passare verso un'architettura che applica i principi dell'Edge Computing (Gartner, 2024). È stata quindi suddivisa l'architettura in due livelli principali: l'Edge industriale con i dispositivi industriali IoT e no, che va a trattare localmente i dati e le informazioni per poi inviare solo quelle pertinente e già elaborate verso il cloud (Gartner, 2024).

Vengono, inoltre, inserite nell'architetture tecnologie come le piattaforme CPS e IIoT descritte nei capitoli precedenti per avere una visione completa delle evoluzioni.

In sintesi, i vantaggi e le caratteristiche chiave sono le seguenti:

- **Disaccoppiamento e scalabilità:** il disaccoppiamento consente di aggiungere o rimuovere produttori e consumatori senza influenzare il resto del sistema, il che migliora la scalabilità.
- **Protocolli open source:** il protocollo MQTT, lo standard Sparkplug e la piattaforma di streaming Kafka definiscono i formati dei messaggi e il comportamento dell'editore e dell'abbonato, utilizzando un formato di serializzazione dei dati ampiamente supportato (buffer del protocollo Google). Queste caratteristiche migliorano l'interoperabilità e l'indipendenza dal fornitore.
- **Flussi asincroni:** garantisce una conoscenza continua dello stato del sistema e dei flussi di eventi, eliminando la necessità per le applicazioni di interrogare i dispositivi per determinare lo stato della sessione e i valori dei parametri.

A causa dell'elevata complessità nell'integrazione delle tecnologie descritte e dell'elevato costo in termini economici e temporali, quest'ultimo scenario è quello di più complessa realizzazione.

## 8. Conclusion

Nel presente documento si è presentato il progetto di ricerca svolto durante l'attività di tirocinio presso Wavestone SA. Si è brevemente introdotta l'azienda, il settore della consulenza e le problematiche attuali per tracciare, successivamente, un percorso evolutivo delle architetture dei Sistemi Informativi (SI), partendo da un'introduzione ai sistemi informativi e una descrizione generale delle architetture dei SI articolata in quattro livelli strutturali (livello dei processi, livello funzionale, livello di applicazione e livello tecnico), delineando come queste componenti abbiano costituito le basi dei sistemi informativi tradizionali.

A seguire è stato presentato uno stato dell'arte degli ecosistemi informativi con una focalizzazione sui livelli funzionale, applicativo e tecnico. Ci si è concentrato sull'individuazione dei componenti principali per ciascun livello e delle sfide attuali affrontate dalle aziende da un punto di vista di gestione dei dati, delle comunicazioni e della modernizzazione del SI in generale. Una particolare importanza viene data alle sfide legate convergenza dell'IT e dell'OT nel campo della trasformazione digitale. L'integrazione delle tecnologie operative attraverso i dispositivi IoT nei livelli IT permette da un lato di unificare la gestione dei dati aziendali, migliorando la reattività, l'efficienza e la capacità decisionale, ma espone anche i sistemi OT a minacce informatiche, richiedendo una maggiore attenzione alla compatibilità e interoperabilità tra tecnologie tradizionalmente distinte. Infatti, secondo uno studio di McKinsey, il 60% delle aziende industriali che ha iniziato l'integrazione tra IT e OT ha riscontrato problemi significativi di sicurezza e governance, specialmente nei settori industriali.

Il documento ha in seguito presentato un'analisi approfondita delle innovazioni tecnologiche più rilevanti nel panorama attuale nell'ambito applicativo. È stato approfondito il concetto di architettura a microservizi, un'architettura software modulare, scalabile e sviluppata nativamente su cloud con l'obiettivo di sostituire i sistemi monolitici tradizionali. Sono state inoltre approfondite il service mesh, che consente una gestione avanzata della comunicazione tra microservizi, e le architetture event-driven, in cui i sistemi reagiscono agli eventi in tempo reale attraverso l'utilizzo di event brokers, garantendo una maggiore flessibilità e reattività.

L'analisi ha anche preso in esame tecnologie emergenti come MQTT, protocollo di messaggistica leggero ideale per le applicazioni industriali, e le piattaforme di streaming di

eventi (ad es. Apache Kafka), che permettono la gestione e l'elaborazione di grandi volumi di dati in tempo reale. Sono state esaminate ulteriori tecnologie complementari come le piattaforme CPS (Cyber-Physical Systems), che offrono una visibilità sulle condizioni di vulnerabilità informatica degli asset fisici, le piattaforme IIoT, e i Data Hub, fondamentali per l'integrazione dei dati provenienti da fonti eterogenee.

Sono stati, infine, presentati quattro scenari ipotetici di integrazione delle tecnologie descritte attraverso una mappatura delle architetture applicative, dimostrando come queste innovazioni possano essere combinate per migliorare la scalabilità, la resilienza e l'efficienza operativa.

L'evoluzione delle architetture dei SI riflette un cambiamento paradigmatico nell'approccio alle tecnologie aziendali. La transizione verso modelli più modulari, flessibili e scalabili, come i microservizi e le architetture event-driven, non solo migliora la capacità di gestire carichi di lavoro sempre più complessi, ma favorisce anche una maggiore integrazione tra il mondo fisico e quello digitale.

Il valore aggiunto delle soluzioni presentate è da una parte, la capacità di gestire i dati in tempo reale, dall'altra l'aumento dell'automazione e della capacità di reazione agli eventi industriali e operativi. Tuttavia, queste innovazioni pongono anche nuove sfide, in particolare in termini di sicurezza, governance e complessità architetturale, che richiedono un approccio equilibrato e metodologie avanzate per garantire una transizione sostenibile verso le architetture di nuova generazione.

Gli scenari proposti hanno come limiti principali alla loro adozione il carattere dirompente delle innovazioni implementate, che si discostano notevolmente dalle soluzioni monolitiche tradizionalmente diffuse negli ambienti informativi aziendali. In secondo luogo, una trasformazione digitale a questo livello comporta costi elevati sia in termini economici sia in termini temporali: un processo di conversione di un'architettura tradizionale verso un'architettura ad eventi deve essere reso parte di un progetto a visione a lungo termine con grandi investimenti iniziali anche da un punto di vista di gestione del cambiamento.

## 9. Bibliografia

- Andrew Banks, Ed Briggs, Ken Borgendale, & Rahul Gupta. (2019). *MQTT Version 5.0*.  
<https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html><https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf><http://docs.oasis-open.org/mqtt/mqtt/v5.0/cos01/mqtt-v5.0-cos01.docx>
- Elia, G., Solazzo, G., Lerro, A., Pigni, F., & Tucci, C. L. (2024). The digital transformation canvas: A conceptual framework for leading the digital transformation process. *Business Horizons*, 67(4), 381–398. <https://doi.org/10.1016/j.bushor.2024.03.007>
- Gartner. (2023). *Guide for event streaming processing*.
- Gartner. (2024). *Reference Architecture for Integrating OT and modern IT*.
- George Westerman, Didier Bonnet, & Andrew McAfee. (2014). Leading Digital: Turning Technology into Business Transformation. *HarvardBusinessReviewPress*.
- Industry 4.0 Barometer*. (n.d.).
- Jeanne W. Ross, Cynthia Beath, & Ina M. Sebastian. (2017). Designing and executing digital strategies. *MISQuarterlyExecutive*.
- Lin, J. (2024). Digital Transformation Strategies and Practices for Corporates. *Advances in Economics, Management and Political Sciences*, 96(1), 204–213. <https://doi.org/10.54254/2754-1169/96/2024MUR0086>
- Marco Tagliavini, Aurelio Ravarini, & Donatella Sciuto. (2008). *Sistemi per la gestione dell'informazione*.
- Quix. (n.d.). *Guide to the Event-Driven, Event Streaming Stack*.
- Sparkplug Specification Project Team. (n.d.). *Sparkplug 3.0.0 Sparkplug Specification Eclipse Sparkplug Contributors*.
- The Industrial Internet Reference Architecture*. (n.d.).
- Thomas H. Davenport, & Jeanne G. Harris. (2007). *Competing on Analytics: The New Science of Winning*.



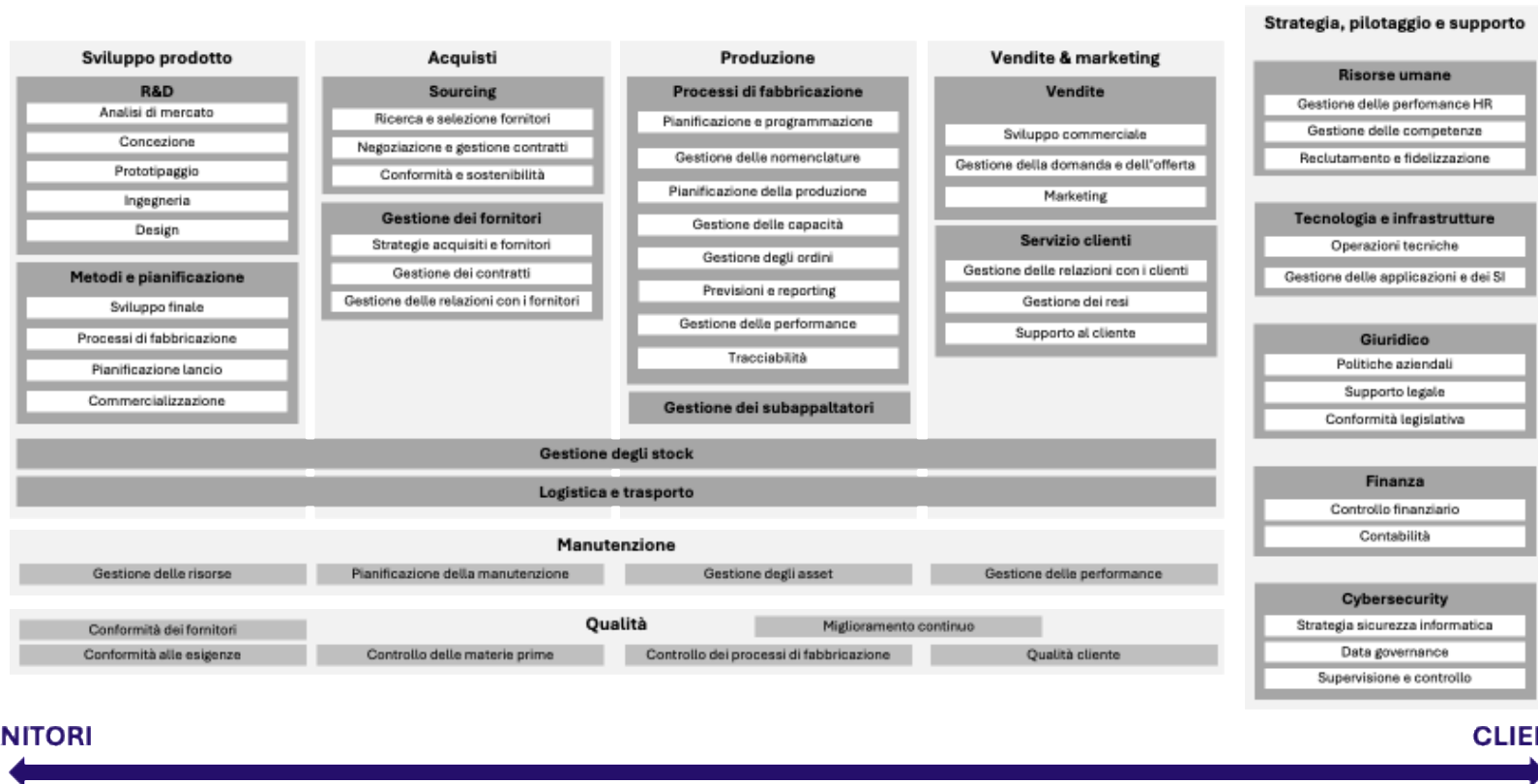
## 10. Sitografia

- <https://www.wavestone.com/en/>
- <https://www.statista.com/statistics/870924/worldwide-digital-transformation-market-size/>
- <https://www.sap.com/resources/what-is-digital-transformation>
- <https://www.statista.com/statistics/1446720/tech-innovation-goals-business/>
- [A mainframe time bomb? How companies seamlessly manage legacy integration | Wavestone](#)
- [Industry 4.0 Barometer - 2023 Edition | Wavestone](#)
- [Chapitre 5-Lurbanisation des SI.pdf \(slideshare.net\)](#)
- [What is a Manufacturing Execution System \(MES\)? | IBM](#)
- [A mainframe time bomb? How companies seamlessly manage legacy integration | Wavestone](#)
- <https://www.databricks.com/glossary/acid-transactions>
- <https://medium.com/@lupass93/zero-trust-architecture-on-kubernetes-with-istio-service-mesh-eade6c5a3c53>
- <https://www.hivemq.com/blog/mqtt-vs-http-protocols-in-iiot/>
- <https://blog.devops.dev/mastering-apache-kafka-a-comprehensive-guide-in-a-nutshell-for-seamless-data-streaming-dbc1c18cedd6>
- <https://www.hivemq.com/blog/mqtt-vs-kafka-real-time-bidirectional-data-processing/>
- <https://www.emqx.com/en/blog/mqtt-and-kafka>
- <https://www.kai-waehner.de/blog/2021/03/22/apache-kafka-mqtt-part-3-of-5-manufacturing-industrial-iiot-industry-4-0/>
- <https://www.kai-waehner.de/blog/2021/03/15/apache-kafka-mqtt-sparkplug-iiot-blog-series-part-1-of-5-overview-comparison>
- <https://www.ibm.com/topics/mes-system>
- <https://www.ptdigital.com/article/mes-vs-erp-how-digital-solutions-work-together>
- <https://www.ptdigital.com/manufacturing-execution-system-complete-guide#how-does-an-mes-work>
- <https://www.machinemetrics.com/blog/downsides-of-mes>
- <https://codeit.us/blog/mes-architecture>
- <https://aws.amazon.com/fr/blogs/industries/refactor-legacy-manufacturing-execution-systems-into-event-driven-microservices/>
- <https://kafka.apache.org/intro>
- <https://easyteam.fr/kafka-pour-les-debutants>
- <https://i-flow.io/en/ressources/kafka-as-a-central-message-broker-for-industry-4-0-vs-mqtt/>
- <https://www.clarify.io/learn/unified-namespace-uns>
- <https://www.hivemq.com/blog/what-is-unified-namespace-uns-iiot-industry-40/>
- <https://cisco.goffinet.org/ccna/fondamentaux/protocoles-modeles-communication/>
- <http://scadalliance.com>
- <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>
- <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>
- [https://testsigma.com/blog/different-types-of-apis-and-protocols-2022-updated/#What\\_is\\_API\\_vs\\_REST\\_API](https://testsigma.com/blog/different-types-of-apis-and-protocols-2022-updated/#What_is_API_vs_REST_API)
- <https://aws.amazon.com/fr/event-driven-architecture/>

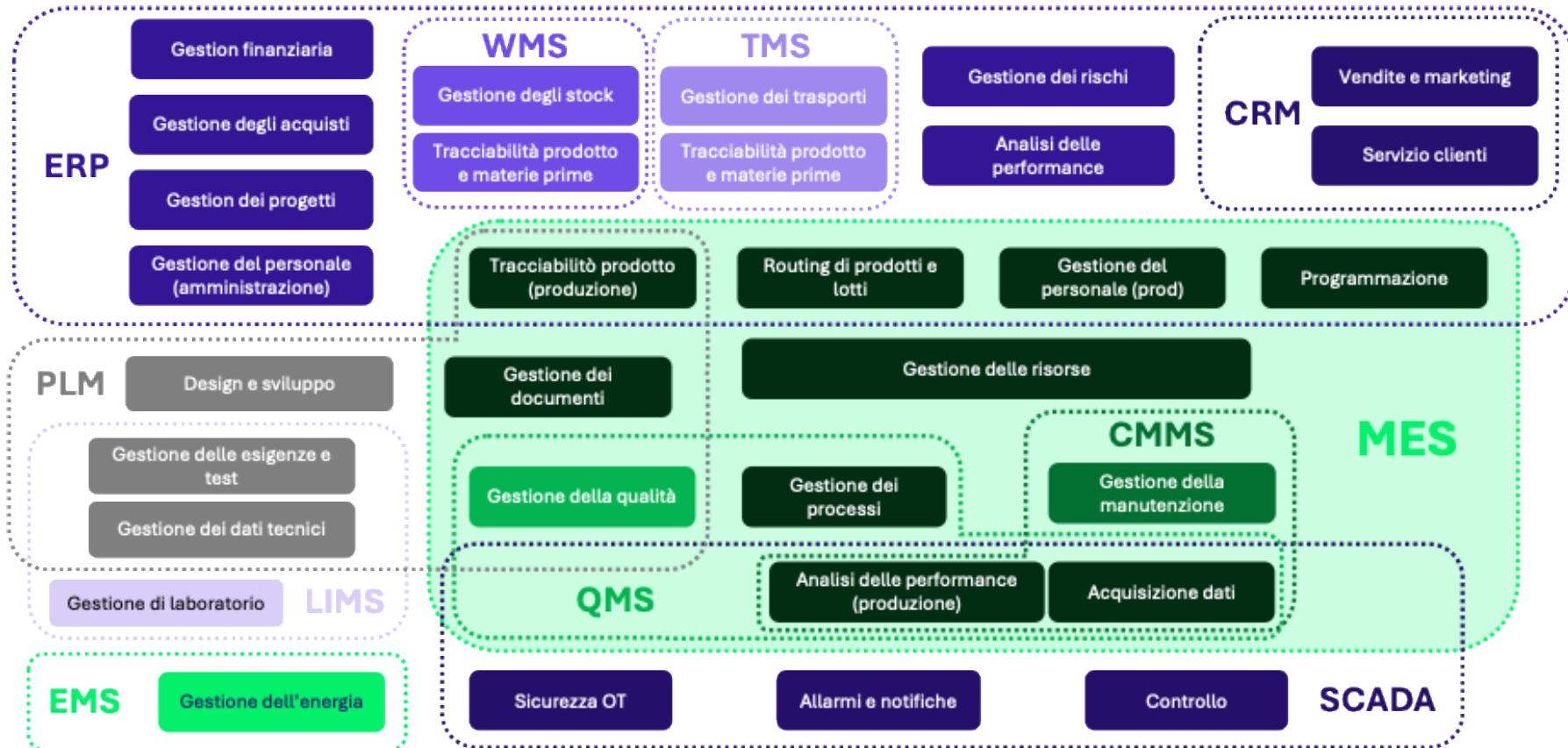
- <https://www.ibm.com/blogs/ibm-france/2023/10/24/point-de-vue-sur-les-architectures-microservices-event-driven/>
- <https://www.confluent.io/learn/event-driven-architecture/>
- [https://www.confluent.io/learn/microservices/?session\\_ref=https://www.google.com/](https://www.confluent.io/learn/microservices/?session_ref=https://www.google.com/)
- <https://www.linkedin.com/pulse/choosing-between-data-event-driven-architecture-bruno-fonzi/>
- <https://www.linkedin.com/pulse/event-driven-architecture-rocky-bhatia-k1nuf/>
- <https://www.qlik.com/us/data-lake>

# 11. Appendice

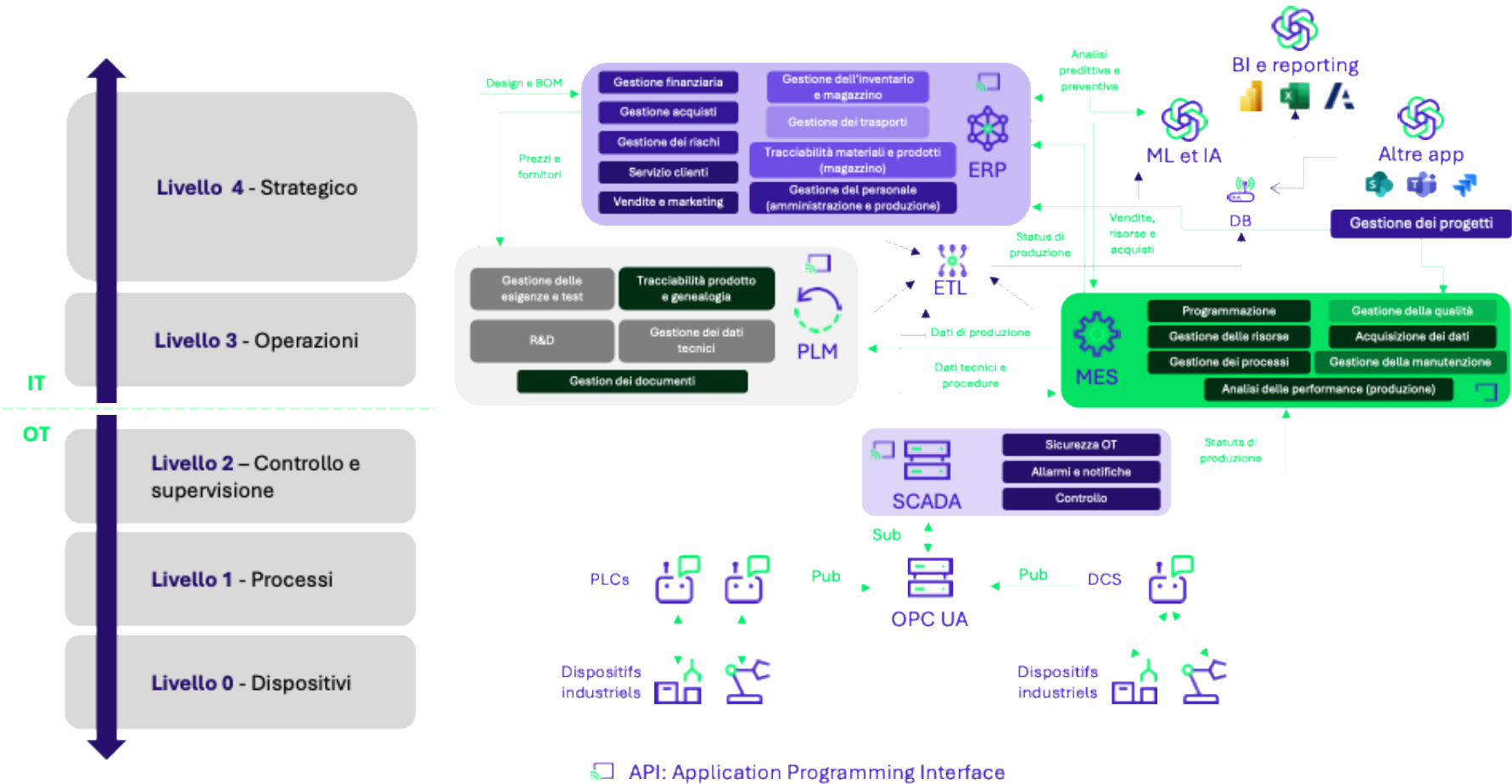
## 11.1. Mappatura funzionale di un'azienda nel settore manifatturiero



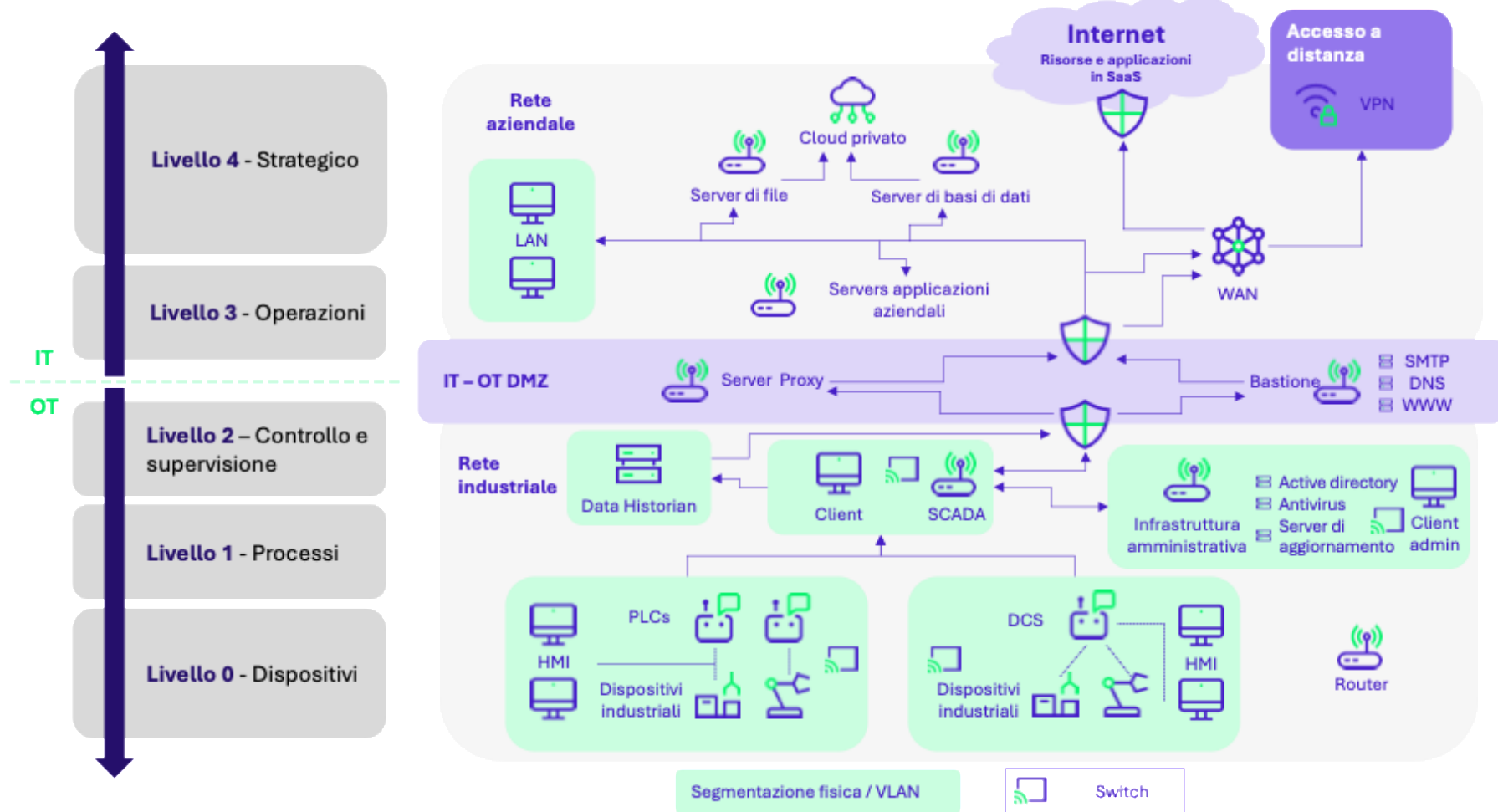
## 11.2. Mappatura funzionale-applicativa della ripartizione delle principali funzionalità dei software di un SI



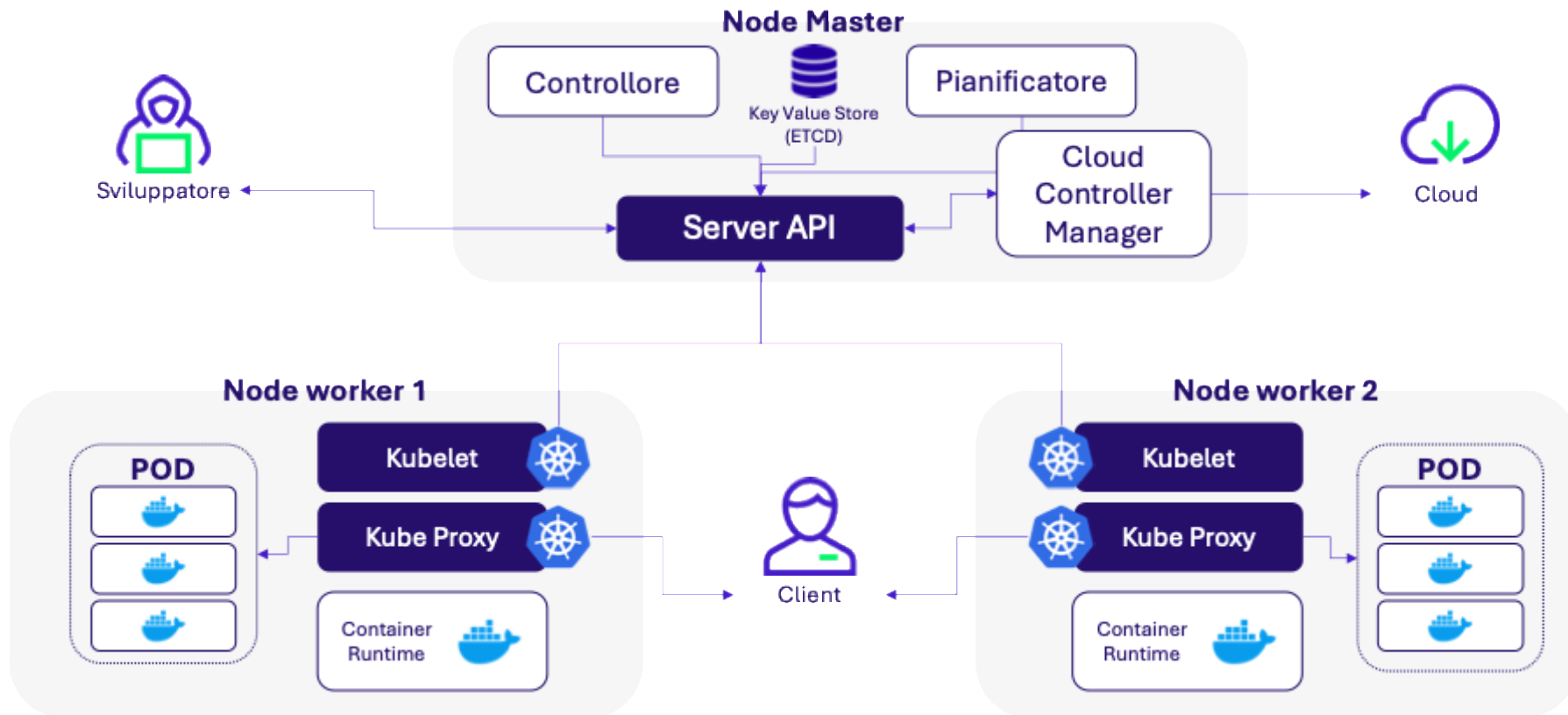
### 11.3. Mappatura applicativa con soluzioni generaliste e modello di Purdue



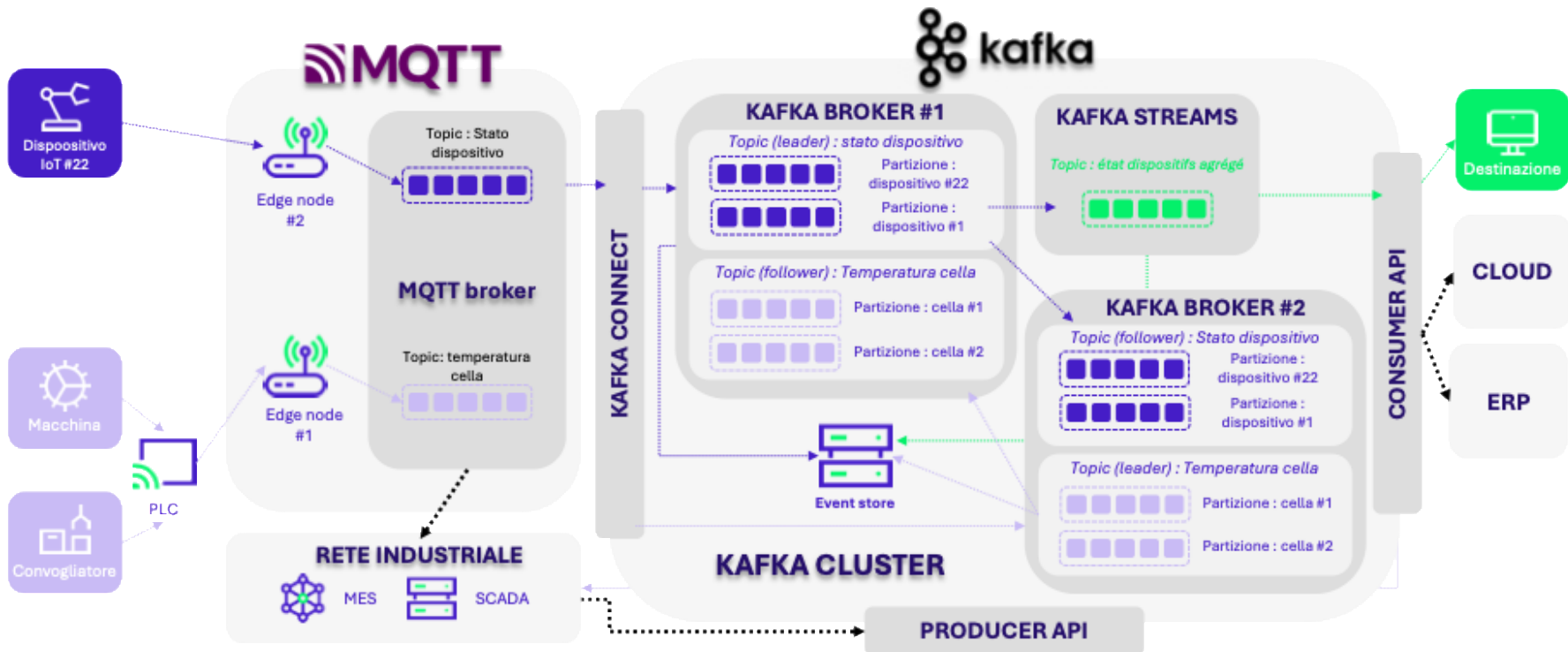
# 11.4. Mappatura tecnica



## 11.5. Architettura a microservizi con Kubernetes e Docker

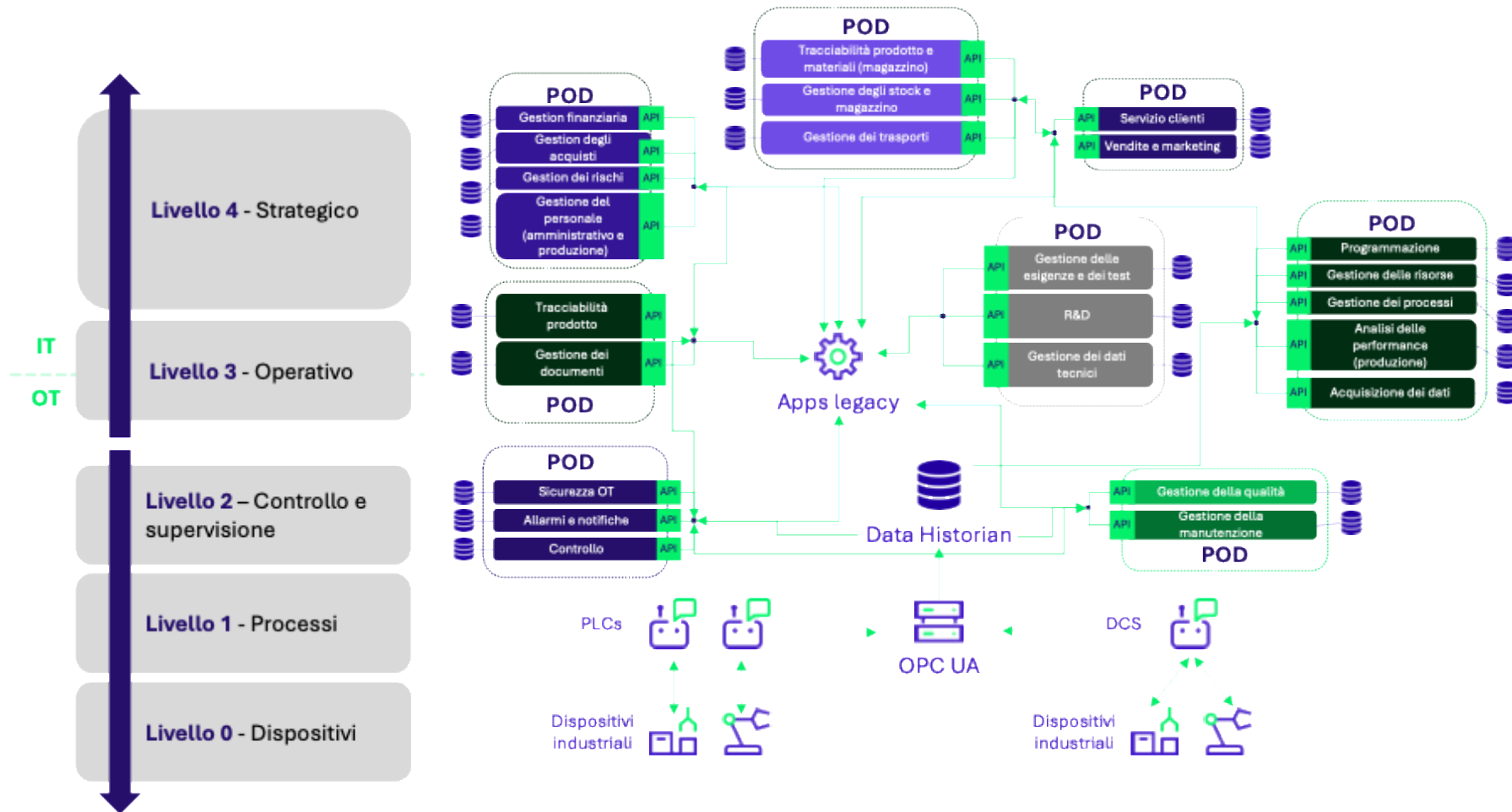


## 11.6. Infrastruttura integrata MQTT-Kafka per le comunicazioni in un'architettura EDA

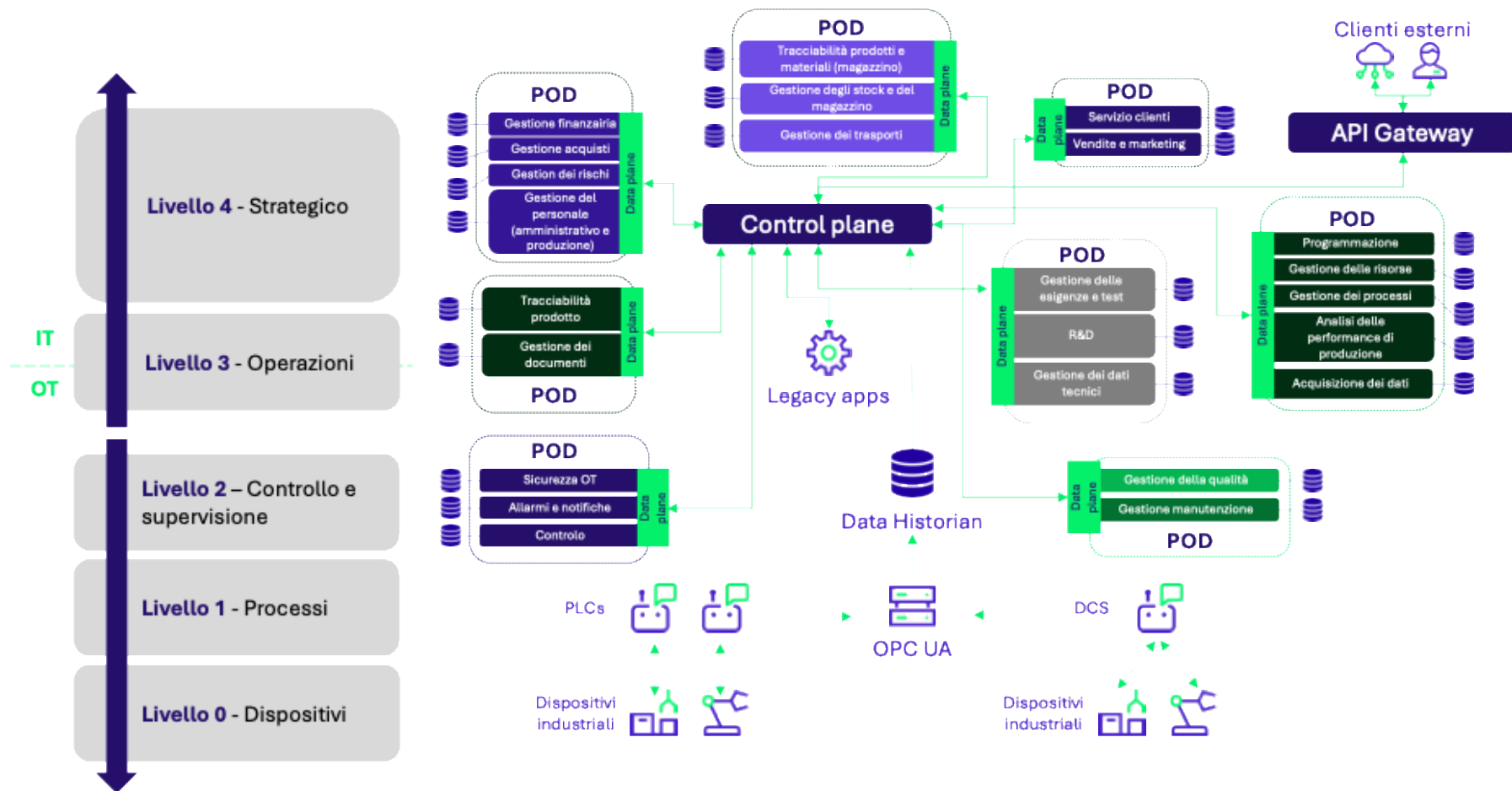




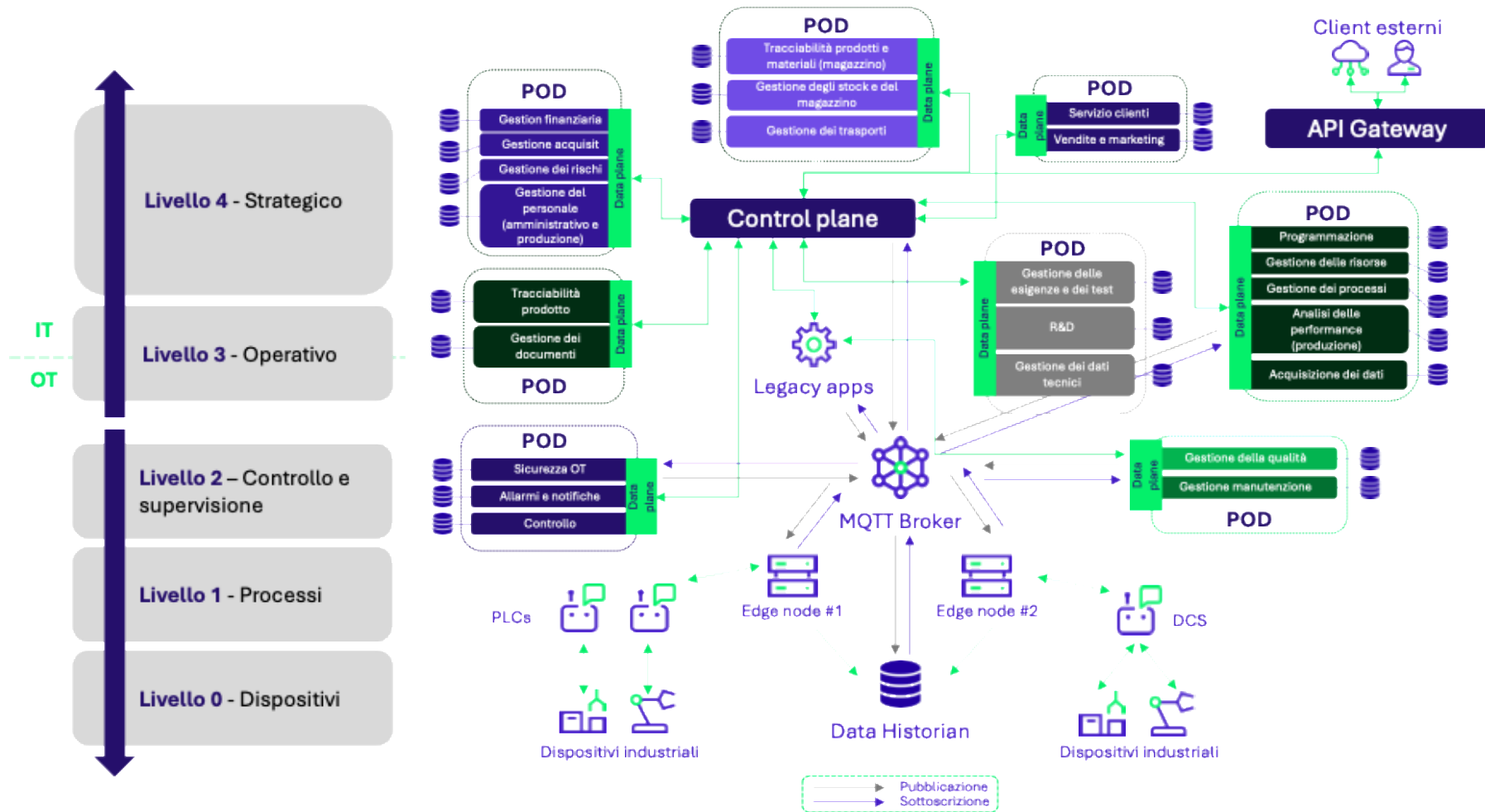
## 11.7. Tappa 1: adozione dei microservizi cloud-native



### 11.8. Tappa 2: integrazione di un Service Mesh per la gestione del traffico est-ovest



### 11.9. Tappa 3: passaggio verso un'architettura ibrida con l'integrazione di un broker MQTT nell'OT



## 11.10. Tappa 4: passaggio definitivo verso un'architettura EDA

