

# RIASSUNTO DI SISTEMI INFORMATIVI

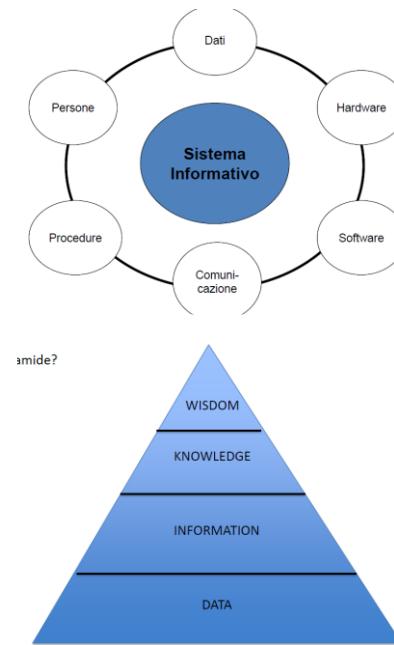
2020

# Capitolo 1 – Introduzione

Un sistema informativo è uno strumento supportato dalla Information Technology (IT) e composto da applicazioni e sistemi di gestione dei dati, interfacce utente e reti di comunicazione, il cui scopo è la gestione dell'informazioni per i fini di una organizzazione.

## Piramide della conoscenza

- **DATO:** un fatto, una misura, con un dominio e una unità di misura
- **INFORMAZIONE:** costruita dall'aggregazione di più dati tramite la loro interpretazione, è il risultato di una "interrogazione" sui dati
- **CONOSCENZA (knowledge):** esperienza integrata all'informazione per interpretare in modo utile l'informazione a fini decisionali
- **SAGGEZZA (wisdom):** ulteriore applicazione dell'esperienza alla conoscenza al fine di consentire di prendere la decisione corretta nel momento adatto



## Risorse

Una **risorsa** è tutto ciò con cui opera una organizzazione.

Le risorse possono essere:

- Interne (Prodotti: beni e servizi, Strumenti finanziari, Persone (risorse umane), Infrastrutture, Norme, Deleghe, Piani, Informazioni, ...)
- Esterne (Ambiente sociale ed economico, Mercato, Clienti)

In particolare, l'**Informazione** è una risorsa ed è impiegata per la comunicazione, il supporto ai processi e alle decisioni, può anche essere un prodotto. L'informazione va gestita.

## Processi

Un **Processo** è l'insieme di attività che un'organizzazione svolge per gestire il ciclo di vita di una risorsa, al fine di raggiungere un risultato definito.

### PIRAMIDE DI ANTHONY:

Classifica i processi secondo i tre livelli della struttura di una organizzazione.

- **LIVELLO OPERATIVO:** attività di tipo operativo di una azienda, come contabilizzazione dei pagamenti o interventi di manutenzione
- **LIVELLO DI PROGRAMMAZIONE E CONTROLLO:** attività tattiche di programmazione delle risorse e controllo sul conseguimento degli obiettivi programmati. Ad esempio, il controllo dei pagamenti, confronti tra entrate e uscite, monitoraggio attività.
- **LIVELLO DI PIANIFICAZIONE STRATEGICA:** attività legate alla scelta degli obiettivi aziendali e alla definizione delle politiche aziendali. Ad esempio, la definizione di nuove tariffe per i clienti, decisioni riguardo l'apertura di nuovi servizi, scelta delle aree di mercato in cui investire.



(questa classificazione vale anche per i dati utilizzati)

### MODELLO DI PORTER:

Classificazione dei soli processi di livello operazionale a seconda delle loro funzionalità e utilità nei processi stessi.

- **ATTIVITA' PRIMARIE:** ciò di cui si occupa una azienda, orientate agli obiettivi di una organizzazione. Sono svolte sequenzialmente.
- **ATTIVITA' SECONDARIE:** sono attività di supporto a quelle primarie, che ne garantiscono il corretto svolgimento. Sono svolte in parallelo a tutte le primarie.



## Definizione e ruolo di un sistema informativo

Un **Sistema Informativo** è un sistema, quindi un insieme di procedure, metodi, strumenti, dedicati alla raccolta di eventi, visti come dati, alla loro trasformazione in informazione, e alla gestione dell'informazione stessa secondo le regole aziendali, al fine del perseguitamento degli obiettivi dell'organizzazione.

"Un sistema informativo è definito come l'insieme dei mezzi, della conoscenza organizzativa e delle competenze tecniche per gestire la risorsa informazione"

L'informazione è utilizzata dal sistema in modi diversi, a seconda dei bisogni, in supporto dei processi, ruoli e obiettivi differenti all'interno dell'organizzazione.

Un **Sistema Informatico** è un componente del sistema informativo che si occupa dell'elaborazione, archiviazione e scambio delle informazioni. Esso cambia nel tempo con l'evolversi dei bisogni dell'azienda e delle tecnologie.

Data la vasta possibilità di applicazione dei sistemi informativi, in una organizzazione sono presenti più sistemi informativi, che si dividono in due categorie:

- **SISTEMI OPERAZIONALI:**
  - svolgono le funzioni di base che comprendono gestione di transazioni, ordinario lavoro d'ufficio e contabilità.
  - Le parti fondamentali di questo tipo di sistemi informativi sono le funzioni operative e la base di dati operazionale.
  - Lavorano su alti volumi di dati che rappresentano in dettaglio gli eventi, sono ben strutturati in basi di dati e provengono dall'interno.
- **SISTEMI DECISIONALI (informazionali):**
  - Svolgono funzioni di Business Intelligence, quindi sono a supporto delle attività decisionali e strategiche.
  - Comprendono sistemi quali il Data warehousing e Decision Support System.
  - Utilizzano dati aggregati e sintetizzati, con formato non necessariamente strutturato e provenienza anche dall'esterno.

NB: Quando sono trattati dati ben strutturati, come avviene sempre nel caso dei sistemi operazionali, anche il processo decisionale da seguire è en definito, quindi è più facile una automazione delle decisioni.

Una **transazione** può essere: uno scambio o un contratto, una attività operativa, una movimentazione di una risorsa, la certificazione di un evento, un'operazione all'interno di una base di dati. In genere le transizioni sono certificate da un documento (cartaceo o elettronico) che registra l'esecuzione.

I sistemi informativi si possono classificare in base al numero e tipo di transizioni che eseguono:

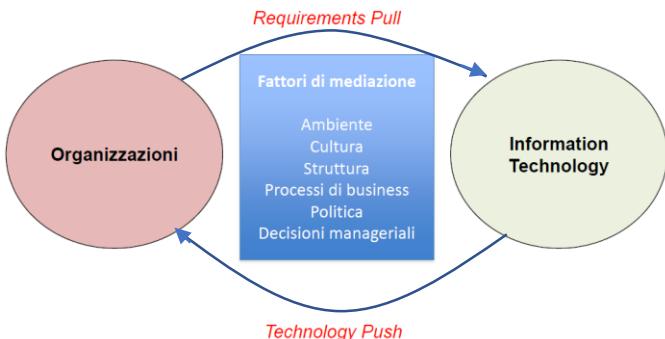
- **SISTEMI OLTP (On-Line Transaction Processing):**
  - svolgono operazioni che richiedono un gran numero di transizioni brevi e On-Line, cioè che hanno un impatto immediato sul sistema da cui sono visibili.
  - Gestiscono molto rapidamente le query mantenendo degli standard di efficienza e assicurano l'integrità dei dati.
  - I dati sono dettagliati e attuali.
  - Sono molto adatti alla gestione di processi a livello operativo e di controllo.
- **Sistemi OLAP (On-Line Analytical Processing):**
  - Svolgono operazioni basate su poche transazioni effettuate poco frequentemente.
  - Le query sono molto complesse e effettuate principalmente in Data Warehouse.
  - I dati utilizzati sono salvati in modo aggregato e hanno provenienza storica.
  - Sono adatti al supporto delle decisioni a livello di pianificazione e strategico.

	OLTP	OLAP
Utente	Impiegato (molti)	Dirigente (pochi)
Funzione	Operazione giornaliera	Supporto alle decisioni
Progettazione	Orientata all'applicazione	Orientata al soggetto
Dati	Correnti, aggiornati, dettagliati, relazionali, omogenei	Storici, aggregati, multidimensionali, eterogenei
Uso	Ripetitivo	Casuale
Accesso	Read-write, indicizzato	Read, sequenziale
Unità di lavoro	Transazione breve	Interrogazione complessa
Metrica	Throughput	Tempo di risposta

## Organizzazione e IT

**Requirements pull:** le decisioni prese a livello strategico o di controllo influenzano le scelte tecnologiche, che seguono le necessità dell'azienda.

**Technology push:** le nuove tecnologie offrono nuove opportunità nel settore di attività dell'azienda e ciò crea nuovi obiettivi aziendali e influenza le scelte organizzative



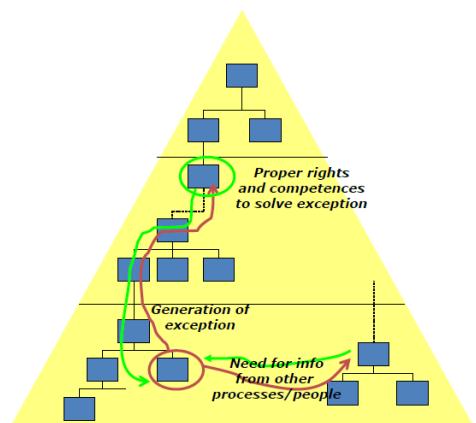
## Flussi informativi

Se sorge un'eccezione, si cercano informazioni al livello a cui ci si trova, e in caso non si riesca si passa al livello gerarchico superiore, fino a che non si trova il punto dove risiedono i diritti e le competenze necessari.

Ci sono quindi flussi informativi verticali e orizzontali.

I componenti del sistema informativo sono messi in comunicazione tra loro mediante il sistema informatico.

E' importante considerare aspetti quali l'accettazione dell'IT da parte degli utenti e il valore aziendale dell'IT.



# Capitolo 2 – Progettazione di SI

## Enterprise Architecture

Inizialmente i sistemi informativi erano chiusi, cioè non connessi con sistemi di altre aziende. Ora sono sistemi più complessi, composti da diversi moduli che interagiscono tra loro per supportare processi interni alla azienda (intraziendiali) e che coinvolgono collaborazione tra aziende (interaziendiali).

E' conveniente utilizzare strutture astratte per rappresentare l'interazione tra i diversi componenti di un sistema informativo, anche perché questo è in continua evoluzione a causa delle spinte di business (requirements pull) e dell'evoluzione tecnologica (technology push), in modo che non esista dipendenza dalle scelte specifiche, tecnologiche o di business.

Una **Enterprise Architecture (EA)** è una descrizione della organizzazione nel suo stato attuale e futuro in una prospettiva che integra strategia, business e tecnologia. E' una descrizione di alto livello del sistema, astratta e di tipo concettuale, che non comprende i singoli dettagli tecnici.

L'EA deve essere gestita tenendo conto delle eventuali evoluzioni del sistema, quindi si sviluppa l'EA futura (to be) in relazione a quella attuale (as is).



Si definiscono **Stakeholder** tutti i soggetti che hanno interesse nel sistema considerato, per una sola fase per il suo intero ciclo di vita, in quanto proprietari, fruitori finanziatori, utenti, eccetera.

## FRAMEWORK DI ZACHMAN:

E' uno strumento utile alla descrizione, organizzazione e prescrizione in fase di realizzazione o evoluzione, di un sistema informativo.

	What Cosa? DATI	How Come? FUNZIONI	Where Dove? RETE	Who Chi? PERSONE	When Quando? TEMPO	Why Perché? MOTIVAZIONE
Contestuale AMBITO <i>Pianificatore</i>	Liste elementi importanti per l'impresa	Lista processi eseguiti dall'impresa	Lista località in cui opera l'azienda	Lista unità organizzative importanti per l'azienda	Lista eventi	Lista obiettivi, strategie
Concettuale Modello Impresa <i>Owner</i>	Modello Entità- Relazion	Modello dei processi	Rete logistica	Modello della struttura organizzativa	Modello eventi	Relazioni tra obiettivi
Logico Modello sistema <i>Progettista</i>	Modello dei dati	Diagramma dei processi	Architettura del sistema distribuito	Diagramma ruoli e relazioni	Diagramma eventi	Diagramma regole
Fisico Modello tecnologico <i>Costruttore</i>	Progettazione dei dati	Specifiche delle funzioni	Architettura del sistema	Interfaccia uomo- macchina	Specifiche eventi	Specifiche regole
Dettaglio Componenti <i>Sottocontraente</i>	Es. Definizione dei dati	Es. Applicazione (codice)	Es. Architettura di rete	Es. Architettura sicurezza	Dettagli eventi	Dettagli regole

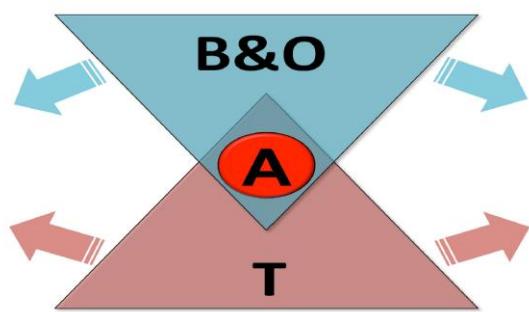
Le colonne rappresentano gli aspetti da analizzare nella descrizione di una architettura.

Per ogni riga, in cui sono ordinati i punti di vista che possono interessare ai diversi stakeholder, vengono definiti i vincoli sul sistema nel livello considerato, che sono additivi, cioè quelli a livello inferiore si aggiungono a quelli superiori.

## Approccio BOAT

Per la progettazione della EA un approccio possibile è quello BOAT in cui gli aspetti da specificare sono:

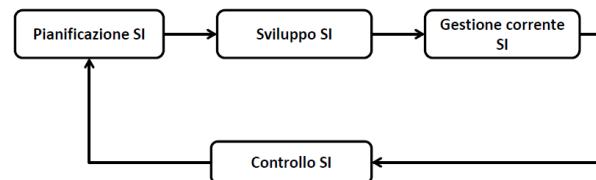
- Aspetti di **BUSINESS**: gli obiettivi aziendali.
- Aspetti di **ORGANIZZAZIONE**: la struttura dell'organizzazione.
- Aspetti di **ARCHITETTURA**: l'**architettura funzionale** del sistema, cioè la sua struttura in termini di componenti software che supportano specifiche funzionalità e in termini di interfacce che ne consentono l'interazione.
- Aspetti di **TECNOLOGIA**: le scelte tecnologiche effettuate.



## Gestione del sistema informativo

Il processo di gestione di un sistema informativo è rappresentato in quattro fasi principali:

- **PIANIFICAZIONE**: in questa fase si delineano le linee guida strategiche e il ruolo delle componenti organizzative e le istruzioni operative per la realizzazione.
- **SVILUPPO**: raccolta e analisi dei requisiti e definizione dell'architettura.
- **GESTIONE CORRENTE**: gli interventi di routine e occasionali più complessi per il mantenimento del sistema.
- **CONTROLLO**: operazioni periodiche di valutazione dell'adeguatezza del sistema informativo.



## Pianificazione

La fase di pianificazione ha l'obiettivo di creare una proposta di progetto che il committente approvi.

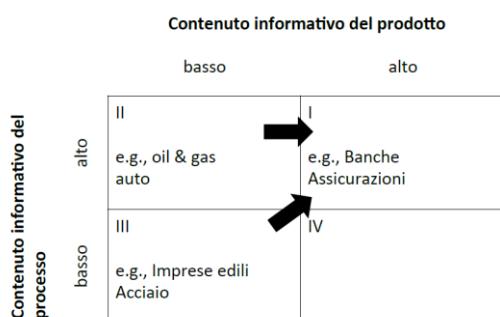
Si suddivide in due attività principali:

- **Pianificazione strategica** in cui sono identificati gli obiettivi, cioè dove sono necessari interventi.
- **Studio di fattibilità** in cui sono analizzate le alternative progettuali con lo scopo di scegliere la migliore secondo diversi criteri.

### PIANIFICAZIONE STRATEGICA:

Si deve innanzitutto comprendere la situazione attuale dell'azienda e le aree di essa che necessitano di interventi di natura informatica. Poi ci sono due attività principali:

- L'analisi delle opportunità di sviluppo del sistema informativo in termini di tecnologie, strategie e processi per delineare possibili sviluppi e modifiche del sistema.
- L'analisi dei fabbisogni informativi in cui si modellano le informazioni e gli scambi informativi necessari all'azienda per migliorarne il funzionamento.



Esistono modelli specifici il cui scopo è la gestione di queste due attività. La matrice di Porter e Millar è uno strumento utile a comprendere l'importanza del sistema informativo nella propria organizzazione a seconda delle attività di cui questa si occupa.

La digitalizzazione è il tentativo di sfruttare le opportunità offerte dalla tecnologia, spostando la posizione della propria azienda nella matrice verso il primo quadrante.

## STUDIO DI FATTIBILITÀ:

Si tratta di uno studio che precisa le caratteristiche di un intervento sul sistema informativo.

È composto da tre fasi:



Nella fase di “Definizione degli obiettivi e specifiche funzionali” sono definite le aree organizzative in cui sono necessari interventi e per ognuna si delineano i miglioramenti che è opportuno apportare, sono quindi descritte ad alto livello le funzionalità che il sistema evoluto dovrebbe fornire. Infine, per ogni intervento sono specificati dei vincoli in termini di costo, tempo e qualità.

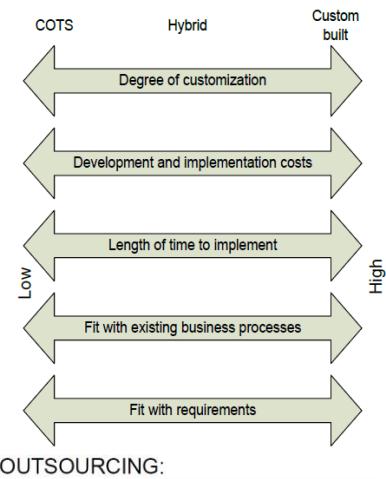
La fase di “Progettazione delle soluzioni” è essa stessa articolata in altrettante tre fasi:

- **Identificazione delle soluzioni:** ogni possibile soluzione è descritta specificando nel dettaglio gli interventi tecnologici. Dopo di che ogni soluzione è sottoposta a due tipi di analisi.

L’analisi **Make-or-Buy** stabilisce se sia più conveniente l’acquisto di un prodotto software già pronto, quindi un **COTS** (Commercial Off-The-Shelf), o la realizzazione di un prodotto su misura. È possibile e spesso opportuno optare per un approccio ibrido tra le due soluzioni. Entrambe hanno comunque vantaggi e svantaggi (vedi immagine).

L’analisi seguente riguarda la gestione del sistema: esso può essere infatti gestito internamente (**in-house**) o esternamente(**outsourcing**). L’outsourcing porta diversi vantaggi, soprattutto alle aziende più piccole, come miglioramento della qualità del servizio in quanto svolto da altre aziende specializzate, la possibilità dell’azienda di focalizzarsi sulle proprie attività principali, e una maggiore propensione all’aggiornamento tecnologico gestito dall’azienda esterna. Gli svantaggi consistono nella perdita di competenze interne e nella dipendenza dal fornitore del servizio.

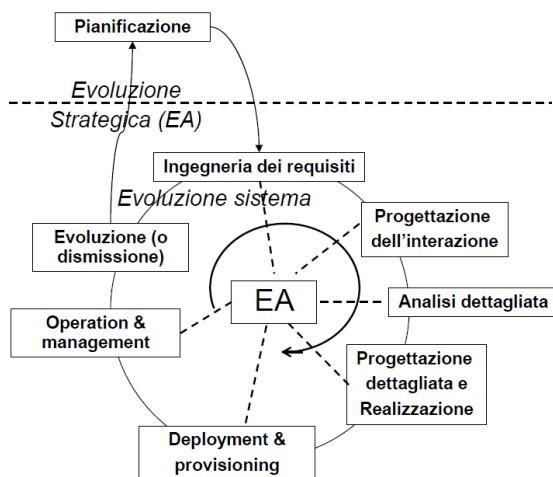
- **Analisi di fattibilità tecnica:** ogni soluzione è valutata in base alla sua adeguatezza da un punto di vista tecnologico, cioè si valuta come essa si integra con il sistema informativo già esistente (sistema legacy) che pone vincoli architettonici sulle soluzioni. In particolare, se i sistemi sono eterogenei si devono considerare costi aggiuntivi per la loro integrazione.
- **Valutazione degli impatti organizzativi:** si valutano gli impatti dell’introduzione della nuova soluzione all’interno dell’azienda: la reazione degli utilizzatori del nuovo sistema, resistenti al cambiamento o con competenze troppo scarse per utilizzarlo, la necessità di dover ridisegnare ruoli o ridefinire mansioni, la necessità di collaborazione con altre realtà.



OUTSOURCING:	
Vantaggi	Svantaggi
Rimanere focalizzati sul core business	Perdita di controllo strategico dell’ IT
Disponibilità di competenze altamente specializzate (solitamente non presenti nell’azienda)	Alta dipendenza dall’azienda esterna
Riduzione dei costi	Gap tra la visione dell’azienda e la visione dell’azienda esterna
	Bisogno di gestire le relazioni con l’azienda esterna
	Bisogno di monitorare i livelli di servizio (SLA)

## Sviluppo

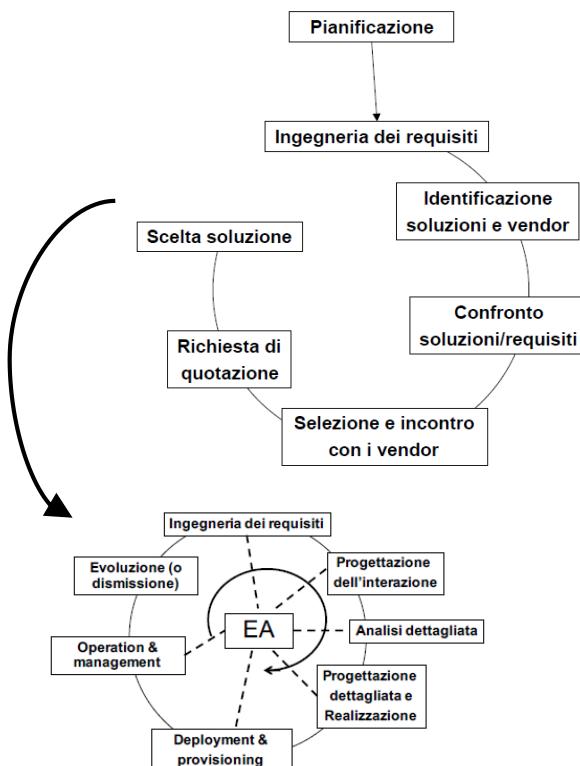
Se si è scelta una alternativa **Make**, lo sviluppo segue un percorso costituito da un ciclo retroazionato composto da diverse attività.



Queste fasi compongono il **ciclo di vita di sviluppo del software**.

Lo sviluppo "a cascata" non è più utilizzato perché lo svolgimento sequenziale delle attività ha innumerevoli svantaggi tra cui la possibilità di verifica dell'aderenza tra prodotto e aspettative del committente solo alla fine del processo di sviluppo. E' quindi preferito uno sviluppo "iterativo" basato sullo sviluppo incrementale della soluzione mediante affinamento e aggiunta di funzionalità a prototipi da analizzare.

Se si è scelta una alternativa **Buy**, il ciclo di sviluppo e implementazione segue il ciclo di acquisto in cui si persegue la selezione della soluzione idonea ai requisiti aziendali.



## Capitolo 3 – Approccio BOAT

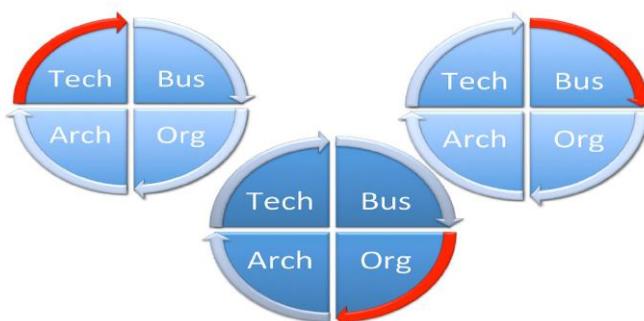
### Approccio BOAT

L'approccio BOAT permette di analizzare il sistema informatico secondo le seguenti prospettive:

- BUSINESS: gli obiettivi di business
- ORGANIZATION: come le organizzazioni sono strutturate e connesse per raggiungere gli obiettivi stabiliti dal business
- ARCHITECTURE: specifica in modo dettagliato i sistemi informativi di supporto alle organizzazioni
- TECHNOLOGY: descrive la realizzazione tecnologica dei sistemi

Gli aspetti analizzati sono le organizzazioni coinvolte, gli oggetti da gestire e il tempo che regola gli eventi significativi da rappresentare.

In genere la progettazione dell'EA avviene in modo iterativo, scegliendo il punto di partenza a seconda del particolare progetto che si sta affrontando, la situazione attuale e gli obiettivi.



# Capitolo 4 – Prospettiva di Business (B)

## Business Model

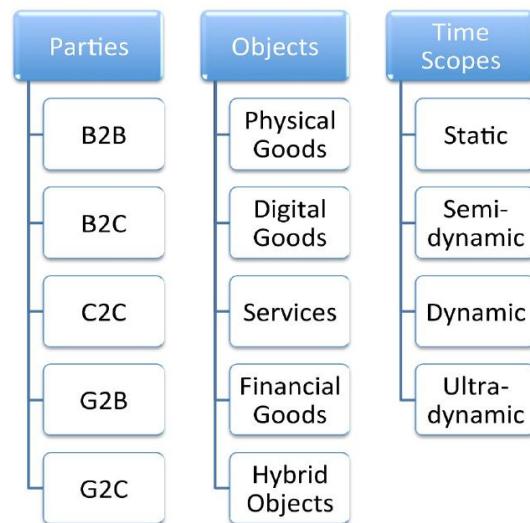
Il risultato dell'analisi di una prospettiva di business è un **Business Model**, che fornisce le motivazioni per effettuare un intervento sul sistema informativo.

Il Business model è costruito identificando uno **scenario** per l'intervento, sulla base dei **business driver**.

## Partecipanti (Party)

I partecipanti ad uno scenario di business sono classificati a seconda del loro tipo:

- BUSINESS (B): aziende che forniscono informazioni, beni o servizi
- GOVERNMENT (G): organizzazioni e amministrazioni pubbliche che forniscono servizi di tipo pubblico
- CONSUMER (C): utilizzatori dei servizi forniti dagli altri partecipanti



## Oggetti

Sono gli oggetti scambiati tra i partecipanti nelle iterazioni all'interno dello scenario. Possono essere:

- Prodotti fisici
- Prodotti digitali
- Servizi
- Prodotti finanziari
- Oggetti ibridi

## Orizzonte temporale

Indica il tipo di relazione nel tempo tra i due partecipanti ad una interazione. L'orizzonte temporale è di diversi tipi:

- STATICO: i partecipanti hanno una relazione stabile nel tempo, generalmente regolata da un contratto.
- SEMI-DINAMICO: i partecipanti hanno una relazione stabile ma che occasionalmente subisce dei cambiamenti.
- DINAMICO: la relazione tra i partecipanti è limitata ad una singola iterazione, come un solo ordine da un cliente a un fornitore.
- ULTRA-DINAMICO: la relazione può variare all'interno della singola iterazione, per esempio se si stabilisse che in caso di ritardo dell'ordine si cambierebbe fornitore

## Business Driver

I **Business Driver** sono i fattori chiave per l'azienda, cioè i motivi che spingono al progetto o all'evoluzione del sistema informativo identificando uno o molteplici obiettivi.

Gli obiettivi devono essere misurabili, per questo si utilizzano degli indicatori, detti **Key Performance Indicator (KPI)**, che hanno sempre come input riferimenti a risorse da utilizzare e due tipi di output: quello *effettivo* che si riferisce a quanto è stato raggiunto concretamente il risultato e quello *atteso* che si riferisce a quanto sarebbe l'obiettivo da raggiungere idealmente.

I Business driver sono di due tipi: l'efficacia e l'efficienza.

## Efficacia

L'**efficacia** è la misura della capacità dell'azienda di raggiungere obiettivi strategici.

$$\text{Efficacia} = \frac{\text{Output effettivo}}{\text{Output atteso}}$$



Nella valutazione dell'efficacia si distingue tra *estensione* e *ricchezza*.

## ESTENSIONE (REACH):

Ci si interessa al raggiungimento di un obiettivo che allarga l'ambito di attività di una azienda.

Ci sono diversi tipi di estensione:

- Geografica: aumentare la presenza a livello territoriale dell'azienda
- Temporale: aumentare la copertura nel tempo, ad esempio aumentare gli orari di apertura
- Modalità (canali): aumentare i modi diversi con cui l'azienda può stabilire interazioni, in generale con i clienti. Ad esempio, l'affiancamento ai canali online tradizionali come web e mail dei social media.

## RICCHEZZA (RICHNESS):

Ci si interessa all'intensità delle relazioni dell'azienda con le parti con cui interagisce, in termini di frequenza dettaglio, tipologia e metodi delle interazioni.

### Efficienza

L'**efficienza** è la misura del rapporto tra output ottenuto e input.

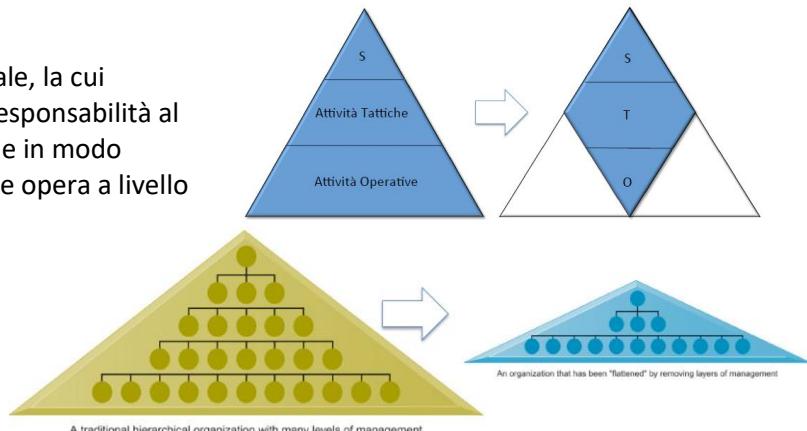
$$\text{Efficienza} = \frac{\text{Output effettivo}}{\text{Input}}$$



Ci sono due modi per ottenere un aumento di efficienza: ridurre le risorse in input o aumentando l'output a parità di risorse utilizzate per ottenerlo.

Spesso l'aumento dell'efficienza è ottenuto tramite l'automatizzazione delle funzioni a livello operazionale, la cui dimensione viene ridotta. Ciò conferisce maggiore responsabilità al livello tattico che deve configurare frequentemente e in modo opportuno i parametri del sistema automatizzato che opera a livello operativo.

Un aumento di efficienza inoltre può portare ad una riduzione dei livelli intermedi dell'organizzazione, in quanto favorisce la velocità e l'analisi dei flussi di informazione tra i vari livelli dell'organizzazione.



## Operationalizzazione dei Driver

L'estensione (Reach) si può operazionalizzare aumentando la disponibilità, quindi aumentare la presenza in termini di tempo (On-time) e avere sempre le risorse utili (On-line), e la accessibilità, quindi avere più canali disponibili a seconda delle evenienze.

La ricchezza (Richness) ha due modi principali di essere migliorata: utilizzando informazioni per creare "customer intimacy" utile a legare il cliente al proprio scenario di business (Customer Relationship Management – CRM), oppure migliorando la "transizionalità", ovvero fornendo al cliente una transizione fluida tra parte fisica e digitale del business (Bricks and clicks integrati).

Business driver	Operationalized driver	Business direction
Reach	Availability	On-time and online business
	Accessibility	Multi-channel business
Richness	Customer intimacy	Enhanced customer relationship management (CRM)
	Transizionality	Integrated bricks & clicks
Efficiency	Cost efficiency	Completely automated business
	Time efficiency	Time-compressed business

L'efficienza si può operazionalizzare secondo le due dimensioni del costo e del tempo, minimizzandoli il più possibile.

## Direzioni di Business

### ON-TIME E ON-LINE:

Nel business on-line, in cui le parti interagiscono con strumenti digitali, si crea una sincronizzazione più stretta fra le parti e i processi sono on-time, o real-time. Oggi questo costituisce lo scenario standard.

### BUSINESS MULTI-CANALE:

L'e-business prevede l'esistenza di molti canali utilizzabili per la comunicazione tra i partecipanti. La flessibilità deve permettere ad essi di poter cambiare il canale su cui stanno operando senza disturbare le relazioni di business e la sincronizzazione deve assicurarsi che le transazioni avvengano sui canali in parallelo senza perdita di informazioni e portando a risultati congruenti.

## GESTIONE AVANZATA DELLA RELAZIONE CON IL CLIENTE:

La raccolta dei dati online è a basso costo e permette di gestire informazioni molto dettagliate e aggiornate sui partner, grazie alle tecnologie dei “Big Data”. Il business tramite CRM può basarsi su una relazione stretta e di alto livello con il cliente (customer intimacy).

## ‘BRICKS AND CLICKS’ INTEGRATO:

È un modello di business basato sull’integrazione del “vecchio” business (Bricks) e del nuovo (clicks) in modo che la sinergia tra i due offra valore a entrambi. Le attività del cliente e dell’azienda devono poter passare da uno all’altro senza impattare le operazioni di business in corso.

## BUSINESS COMPLETAMENTE AUTOMATIZZATO:

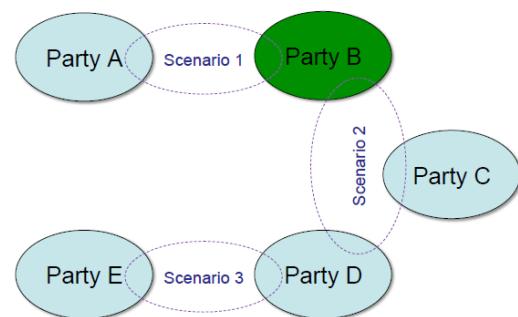
La tecnologia dell’informazione è utilizzata nell’e-business per automatizzare funzioni tradizionalmente affidate a persone, questo porta tipicamente ad una diminuzione dei costi.

## BUSINESS COMPRESSO TEMPORALMENTE:

Si cerca di ottenere un modello di business basato su transizioni eseguite in una frazione di tempo ridotta rispetto ai modelli tradizionali. Ciò avviene integrando business on-line e on-time. E’ importante che tutte le organizzazioni che collaborano o partecipano a scenari comprimano i loro tempi di esecuzione, altrimenti si creano colli di bottiglia.

## Partecipanti e scenari

Per ottenere una descrizione strutturata delle relazioni tra organizzazioni e del loro funzionamento, è necessario separare i diversi punti di vista relativi alle interazioni. Sono quindi definiti diversi **scenari** di interazione tra i partecipanti, ciascuno con i propri obiettivi.



## Tabella per descrivere il modello di business

Category	Value(s)	Remarks
Parties	<specify party type(s)>	<optionally state remarks>
Objects	<specify object(s)>	<optionally state remarks>
Time Scope	<specify time scope(s)>	<optionally state remarks>
Drivers	<specify business driver(s)>	<optionally state remarks>
Directions	<specify business direction(s)>	<optionally state remarks>

# Capitolo 5 – Prospettiva Organizzativa (O)

## Aspetti organizzativi

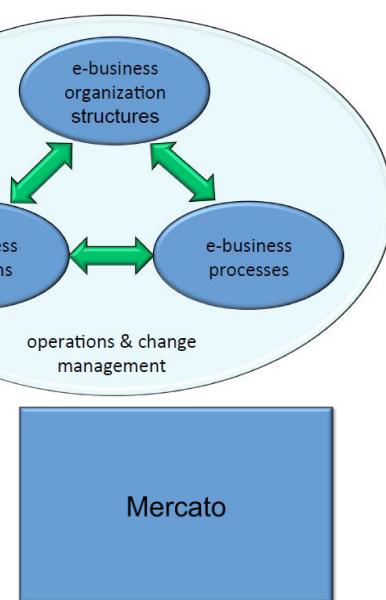
La prospettiva organizzativa è analizzata a partire dalle strutture inter-organizzative, passando poi al livello della singola organizzazione di interesse, quindi alle strutture intra-organizzative, con un approccio top-down.

I principali componenti degli aspetti organizzativi sono:

- Strutture dell'organizzazione e-business: i partecipanti che interagiscono con l'organizzazione
- Funzioni di e-business: le funzioni organizzative derivate dal modello di Porter
- Processi di e-business: modellazione dei processi come sequenze possibili di interazione tra le parti

## Livello 0

La struttura più semplice di uno scenario è il mercato stesso visto come una scatola nera, questa struttura organizzativa è etichettata come livello 0.



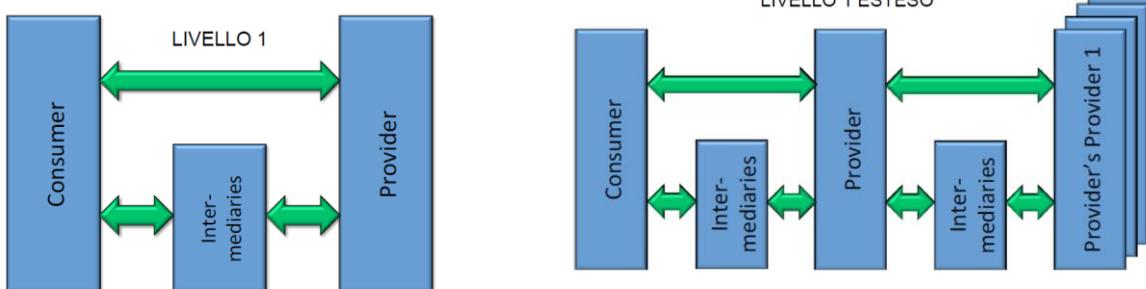
## Livello 1

Nel livello 1 avviene un primo raffinamento e si identificano tre ruoli principali:

- CONSUMATORE: la parte che richiede un oggetto di e-business
- FORNITORE: la parte che offre un oggetto di e-business
- INTERMEDIARIO: una (o più) parte che ha un ruolo ausiliario nel trasferimento di oggetti dal fornitore al consumatore

Il fornitore può essere esso stesso consumatore in relazione ad un altro fornitore, quindi i ruoli sono solo relativi allo scenario particolare di e-business analizzato. Anche gli intermediari sono ruoli relativi al solo scenario analizzato.

Questa criticità è analizzata nel Livello 1 esteso.

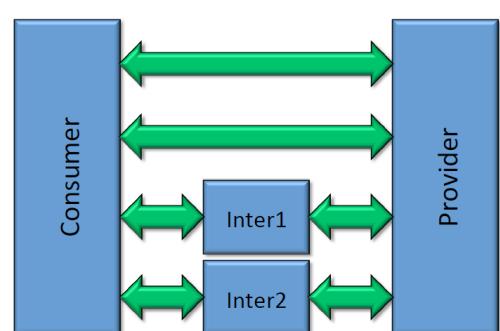


## Livello 2

Al livello 2 avviene un raffinamento di intermediari e di canali. Infatti, esistono diversi tipi di intermediari, e spesso sono richiesti in contemporanea:

- BROKER: rende possibile alle parti di trovarsi e identificarsi sul mercato
- INTERMEDIARIO FINANZIARIO: si occupa della gestione dei pagamenti
- INTERMEDIARIO DI TRASPORTO: gestisce il trasporto di oggetti fisici tra le parti

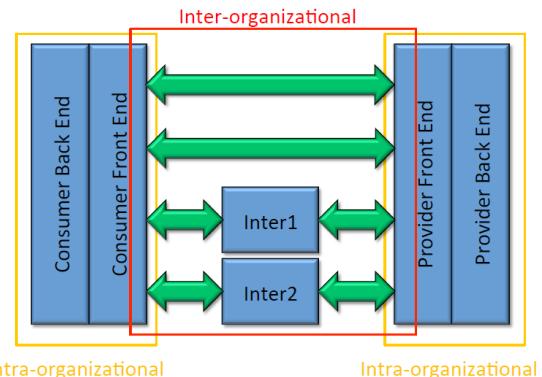
Il numero di intermediari è arbitrario. Inoltre, a questo livello sono rappresentati canali multipli che servono una funzione di business diversa.



## Livello 3

Il livello 3 evidenzia il disaccoppiamento tra le funzioni di business orientate internamente e esternamente. Infatti, esse sono divise tra:

- **BACK END:** Funzionalità di business centrale che ha scopo intra-organizzativo e non è esposta all'esterno. In genere queste funzionalità cambiano di rado.
- **FRONT END:** Funzionalità di business in contatto con le parti esterne con scopo inter-organizzativo. Queste funzionalità sono soggette a frequenti cambiamenti dovuti all'evoluzione dei business model e delle tecnologie.

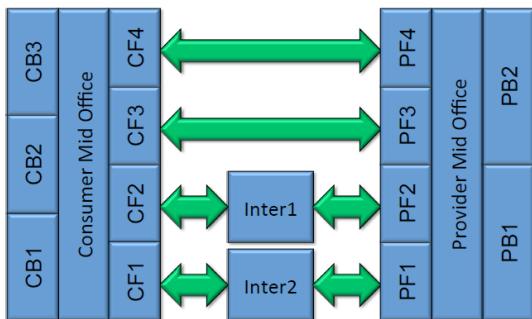


## Livello 4

Al livello 4 sono identificate le singole funzionalità nel front end, allocandole in blocchi organizzativi diversi, detti *unità organizzative* o *moduli organizzativi*.

In caso di Business B2C, la parte Consumer ha in genere funzionalità molto limitate, conviene quindi evitare la distinzione tra funzionalità di back end e di front end.

Se una organizzazione utilizza canali multipli per la stessa funzione di business, è utile avere un modulo organizzativo per ognuno di questi canali.



## Processi inter-organizzazione

Un **processo di business** è un insieme di attività svolte in un ordine specifico per ottenere un certo *obiettivo di business*. Queste attività sono eseguite da *attori*, che nella descrizione del processo sono

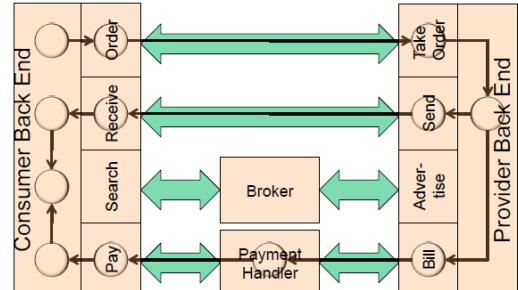
concettualizzati nei loro *ruoli*, che descrivono le capacità richieste a un attore per svolgere l'attività necessaria. Un processo è sempre specificato secondo un *modello di processo* che specifica l'ordine dei passi (attività elementari o sottoprocessi), mentre l'ordine nel tempo dei passi è detto *flusso di controllo* e può non corrispondere al modello.

I **processi intra-organizzazione** rappresentano i processi messi in opera all'interno dei confini di una singola organizzazione.

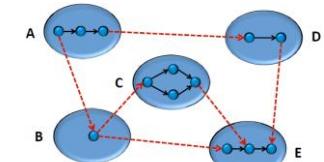
I **processi inter-organizzazione**, o processi di rete, invece rappresentano le possibili sequenze di interazione tra le parti, allocando i diversi passi di un processo alle diverse parti di uno scenario.

I processi inter-organizzazione sono di tre tipi, a seconda della gestione del flusso di controllo:

- Flusso di controllo UNILATERALE: una parte ha il controllo completo dell'intero processo, mentre le altre parti partecipano eseguendo compiti specifici su richiesta del controllore.
- Flusso di controllo BILATERALE: le due parti controllano il flusso in modo collaborativo dividendosi il processo in due porzioni, di cui la rispettiva parte è responsabile.
- Flusso di controllo MULTILATERALE: le molteplici parti (più di due) controllano ognuna la sua parte di flusso di controllo in modo collaborativo.



Nelle classi del flusso di controllo bilaterale e multilaterale, le organizzazioni devono esporre i dettagli dei propri processi utili agli altri partecipanti (vista pubblica), in modo da sincronizzare i processi. Non devono necessariamente esporre tutti i dettagli dei processi interni (vista privata), ma in genere solo astrazioni dei processi interni in termini di processi esterni.



# Capitolo 6 – Architetture funzionali (A)

## Architetture e livelli di astrazione

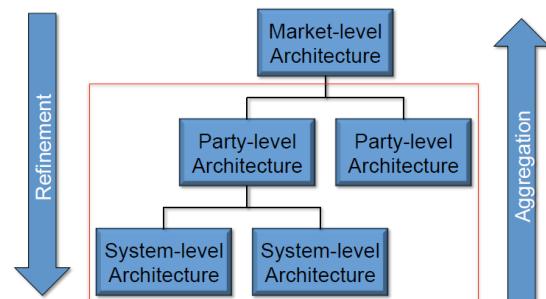
L'architettura è il blueprint della struttura di un sistema complesso, in particolare: "L'**architettura funzionale** di un sistema informativo specifica la struttura del sistema in termini di componenti software funzionali che supportano una specifica funzione e le interfacce che supportano le interazioni fra quei componenti".

Le architetture sono progetti delle strutture di sistemi di e-business, focalizzate sui loro componenti concettuali, e fanno da interfaccia tra elementi non-IT, cioè B e O di BOAT, e gli elementi IT, cioè T di BOAT.

Le architetture sono specificate a diversi livelli di dettaglio o di aggregazione, maggiore è l'aggregazione e minore è il dettaglio dell'analisi, e viceversa.

I livelli di aggregazione sono:

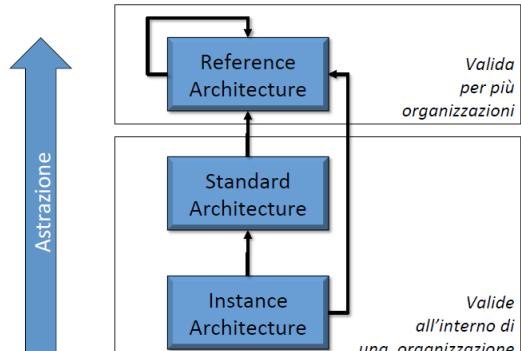
- **ARCHITETTURA MARKET-LEVEL**: la struttura dei sistemi di e-business al livello in cui i vari partecipanti si inseriscono nello scenario, concentrandosi sulle relazioni e interazioni tra i sistemi dei partecipanti.
- **ARCHITETTURA PARTY-LEVEL**: la struttura dei sistemi a livello di organizzazioni singole.
- **ARCHITETTURA SYSTEM-LEVEL**: le strutture in dettaglio dei sottosistemi di un partecipante all'e-business.



La descrizione può essere anche basata su livelli di astrazione: si parte da livelli più astratti, descrivendo i componenti in termini generici con focus sulla loro funzionalità, per arrivare a descrizioni in termini molto concreti.

I livelli di astrazione sono:

- **ARCHITETTURA ISTANZA**: architettura di uno specifico sistema informativo in uno specifico contesto (per esempio una architettura party-level).
- **ARCHITETTURA STANDARD**: architettura definita come standard per una classe di sistemi informativi in uno specifico contesto organizzativo, da utilizzarsi come base per la definizione di architetture istanze di sistemi specifici.
- **ARCHITETTURA DI RIFERIMENTO**: architettura definita come uno standard per classi di sistemi per varie organizzazioni, indipendente dalla tecnologia da adottare, spesso definita da enti di standardizzazione o organizzazioni governative.



L'utilizzo di architetture standardizzate è utile ad assicurare che vari sistemi basati su tali architetture siano compatibili e vadano bene assieme (fit together).

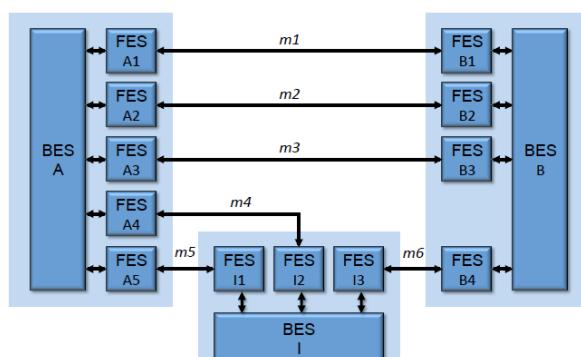
## Architetture Market-level

La architettura Market-level descrive la struttura di un intero sistema informativo e si focalizza sui messaggi scambiati tra i partecipanti, in particolare sulle interazioni dei sistemi software delle organizzazioni che prendono parte allo scenario di e-business. Questa architettura ha carattere inter-organizzativo, quindi descrive i sistemi informativi di tutti i partecipanti.

Sono mostrati i componenti software funzionali, cioè componenti software che supportano specifiche funzioni di business, e i collegamenti tra di essi.

I messaggi che sono scambiati tra i componenti sono esplicitati tramite il contenuto scambiato.

I sistemi di back end sono mostrati come black box perché la loro struttura non è rilevante per le interfacce viste dalle organizzazioni collaboranti.



Message Set	Contents Exchanged
m1	Provider search request, Provider search result
m2	Provider profile, Provider offer
m3	Order, Order confirmal on
m4	Payment order
m5	Payment nol fical on
m6	Service request, Service informal on
m7	Delivery nol fical on, Delivery confirmal on
m8	Shipment request, Shipment confirmal on
m9	Shipment request, Shipment confirmal on
m10	Product on order, Product on confirmal on

## Architetture Party-level

L'architettura Party-level descrive la struttura dei sistemi di ogni partecipante di uno scenario.

“L'architettura Party-level di un sistema di e-business di un partecipante in uno scenario definisce la struttura di quel sistema a livello intra-organizzativo di quel partecipante in termini di: i) componenti funzionali software che supportano funzioni specifiche di quel partecipante e ii) interfacce di alto livello che supportano le interazioni fra quei componenti, come pure le interazioni con altri partecipanti dello scenario.”

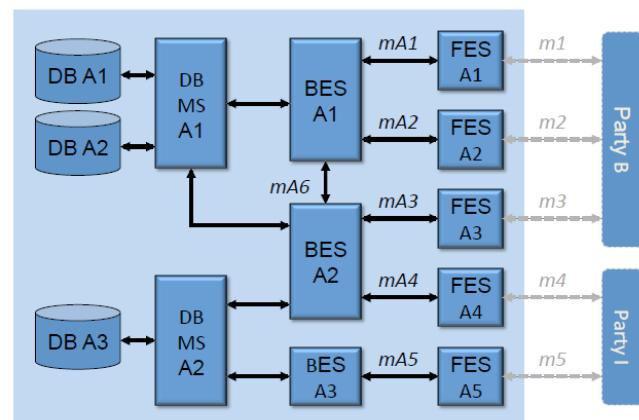
Preso un partecipante allo scenario nella architettura Market-level, sono raffinati il suo back end, completo di tutte le interfacce tra i sistemi che ne fanno parte, comprese quelle tra sistemi di front end e back end.

I componenti delle architetture Party-level sono:

- **COMPONENTI GENERICI**: componenti funzionali.
- **Database e DBMS**: considerati come sistemi che forniscono la funzionalità di gestione dei dati.
- **CONNETTORI**: entità di collegamento tra i componenti.

Il front end è lo stesso dell'architettura Market-level e gli altri partecipanti dell'architettura Market-level sono rappresentati come entità esterne. Le interfacce tra sistemi di front end e back end sono etichettate con i tipi di messaggi.

L'architettura Party level mostra il sottoinsieme della EA complessiva che è coinvolto nello scenario analizzato.



## Architetture System-level

Una architettura System-level è un raffinamento di uno specifico sistema componente, back end o front end, dell'architettura Party-level.

“L'architettura System-level definisce la struttura di uno specifico sistema informativo di uno specifico partecipante in uno scenario in termini di componenti software che supportano specifiche sotto-funzioni di una funzione di quel partecipante e le interfacce che supportano le interazioni fra quei componenti.”

I sistemi piattaforma sono in genere sistemi standard COTS.

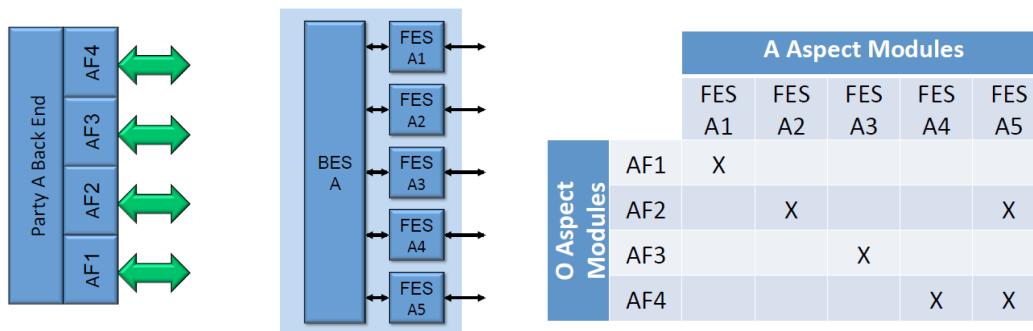
I sistemi con cui il sistema in esame comunica sono mostrati come entità esterne, collegati tramite interfacce esplicitate che corrispondono a quelle presenti al Party level.

Le interfacce tra i sistemi sono specificate e etichettate con insiemi di messaggi.

## Architetture organizzative e funzionali

A livello di business, è necessario ci sia allineamento tra i requisiti a livello di business e i sistemi informativi realizzati, quindi si cerca di mantenere grande somiglianza tra le strutture organizzative (O) e le strutture dei sistemi dell'architettura (A).

Gli elementi rilevanti nella prospettiva organizzativa devono essere visibili anche nella descrizione dell'architettura, e le dipendenze evidenziate chiaramente, per esempio con l'ausilio di una matrice di corrispondenza. Nel caso più semplice esiste una corrispondenza uno-a-uno tra le descrizioni delle funzionalità ad alto livello in O e gli elementi architetturali in A.



# Capitolo 7 – Intro agli aspetti tecnologici (T)

## Tecnologia a livello applicativo

Questo livello di tecnologia comprende le applicazioni software utilizzabili a supporto delle funzionalità aziendali, sia nei sistemi operazionali che in quelli informazionali. L'insieme dei componenti a livello applicativo di una azienda è definito **portafoglio applicativo**.

Le tipologie di applicazioni sono:

- **DW – Data Warehouse**: applicazioni di archivio di dati.
- **BI – Business Intelligence**: applicazioni per la raccolta e analisi di informazioni utili per studiare la situazione aziendale.
- **EB – E-Business**: applicazioni utili a svolgere attività di business che comportano comunicazione, collaborazione tra imprese e esecuzione di transazioni aziendali.
- **CRM – Customer Relationship Management**: applicazioni utili alla gestione delle interazioni con i clienti tramite l'analisi di interazioni stesse e di offerte innovative mirate alla fidelizzazione della clientela.
- **ERP – Enterprise Resource Planning**: suite software utile al supporto delle principali attività di una azienda.
- **APS – Advanced Planning and Scheduling**: Applicazioni utilizzate in ambito manifatturiero per la gestione di materie prime e capacità produttiva.
- **MES – Manufacturing Execution Systems**: sistemi che si occupano di tracciare e documentare il processo produttivo.

Questi moduli corrispondono a componenti funzionali nelle architetture Party-level e possono essere a loro volta composti da più moduli funzionali.

## Tecnologia a livello di piattaforma

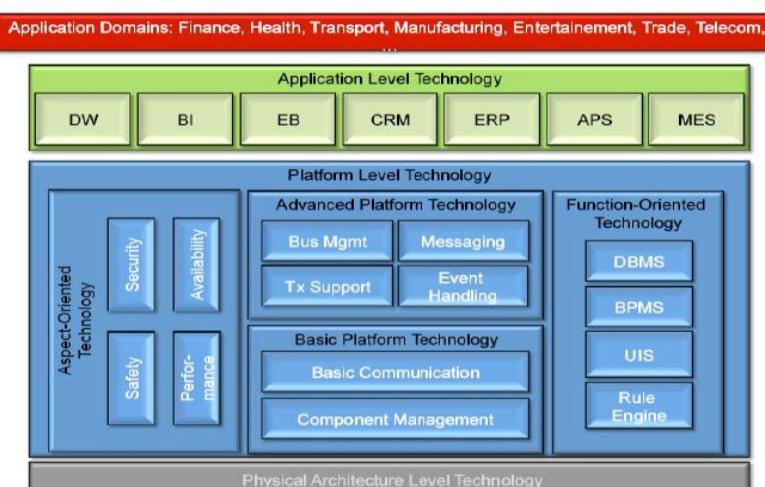
Sono tecnologie utili allo sviluppo delle applicazioni, in grado gestire diversi aspetti e classificate come segue:

- TECNOLOGIE ORIENTATE ALLE FUNZIONALITA':
  - **DBMS – DataBase Management Systems**
  - **BPMS – Business Process Management Systems**: utili a modellare e controllare i flussi di attività.
  - **UIS – User Interface Systems**
  - **Rule Engine**: sistemi software per testare e eseguire a runtime le regole di business.
- TECNOLOGIE ORIENTATE AGLI ASPETTI NON FUNZIONALI: si occupano di aspetti non funzionali come la sicurezza e le prestazioni.
- TECNOLOGIE DI BASE: per la gestione della comunicazione tra applicativi e componenti.
- ADVANCED PLATFORM TECHNOLOGY: si occupano di aspetti avanzati come il bus management, il messaging e la gestione di eventi.

## Tecnologia a livello di architettura fisica

Comprendono i paradigmi con cui si allocano le applicazioni su macchine fisiche, considerando anche soluzioni di outsourcing.

Una volta definito chi gestisce l'infrastruttura, le risorse sono rese disponibili come **servizi** tramite virtualizzazione. La struttura di comunicazione tra servizi è fornita dal **ESB – Enterprise Service Bus** che si occupa del corretto instradamento dei messaggi scambiati tra i servizi senza conoscere la loro collocazione fisica su una piattaforma.



# Capitolo 8 – Tecnologie a livello applicativo

## Componenti funzionali

Il sistema informativo è composto da diverse applicazioni che sono dei casi modulari, ognuno dei quali spesso rappresenta una attività all'interno dell'azienda.

La modularità ha numerosi vantaggi: per primo una elevata manutenibilità dell'architettura, poi il fatto che cambiare un modulo non ha impatto sul resto dell'architettura.

Di solito i moduli sono COTS per la maggior parte, mentre solo alcune funzionalità sono realizzate appositamente per l'architettura progettata.

Le principali applicazioni che sono solitamente acquisite in modalità Buy dalle aziende sono ERP e CRM.

## ERP

L'ERP (Enterprise Resource Planning) è una suite software che contiene moduli a supporto del sistema operazionale e quindi delle transazioni interne all'organizzazione finalizzate alla gestione dell'impresa.

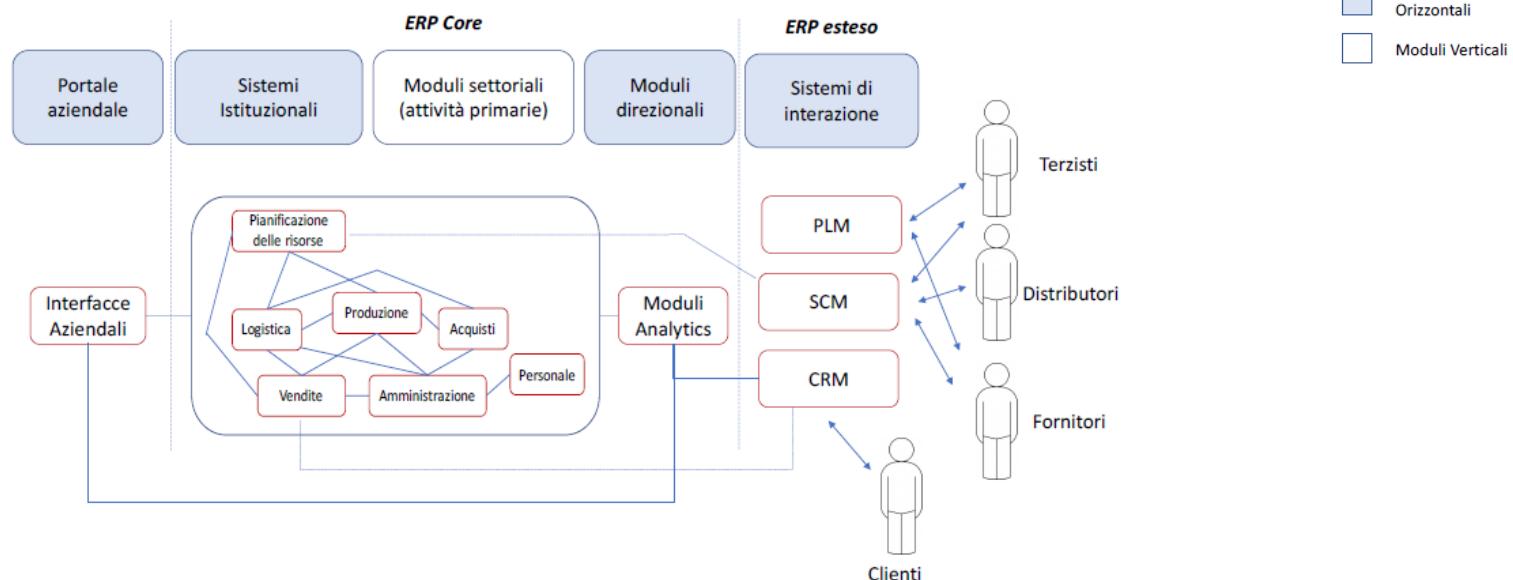
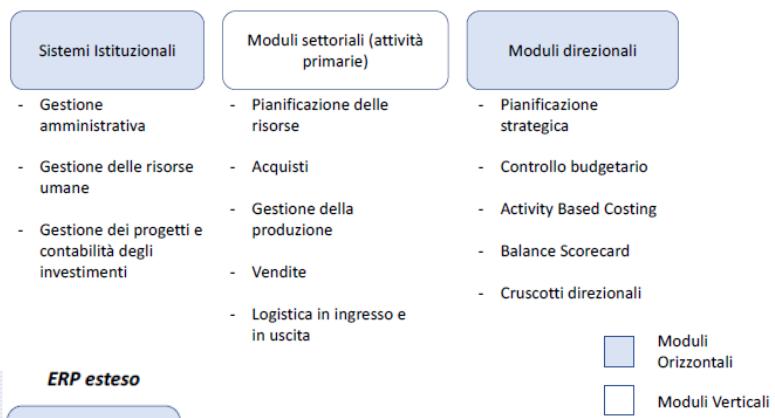
Gli ERP hanno tre proprietà fondamentali:

- UNICITA' DELL'INFORMAZIONE: esiste una sola rappresentazione logica dei dati valida per ogni modulo all'interno dell'ERP. Infatti gli ERP utilizzano una base di dati unica che offre le proprie funzionalità tramite interfacce.
- MODULARITA': ha benefici su flessibilità e scalabilità dell'applicazione, oltre ad abilitare nuove strategie di acquisizione dei COTS.
- PRESCRITTIVITA': l'ERP incorpora la logica di funzionamento dell'impresa e le business rules, ciò aiuta a normare i processi obbligando le attività ad adattarsi alle regole del software che sono scelte.

I moduli si possono classificare in tre categorie:

moduli istituzionali, moduli settoriali (attività primarie) e moduli direzionali.

Solo i moduli settoriali sono specifici di un particolare contesto aziendale (moduli verticali), mentre i sistemi istituzionali e i moduli direzionali sono intersettoriali per le loro funzionalità (moduli orizzontali).



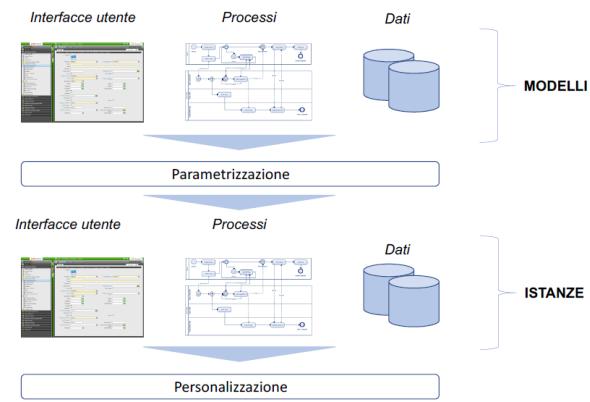
L'ERP era originariamente composto dall'ERP core, poi è nato l'**ERP esteso**, che comprende moduli per l'interazione con i clienti come il PLM (Product Lifecycle Management) che si occupa di gestire le informazioni riguardanti i prodotti durante tutto il loro ciclo di vita, l'SCM (Supply Chain Management) che è a supporto delle decisioni di acquisto, e il CRM (Customer Relationship Management) che supporta l'interazione con il cliente dal contatto al post-vendita.

Inizialmente, negli anni 90', gli ERP erano basati sul paradigma Client-Server, poi con la diffusione del Web sono comparsi i Web-based ERP, contenenti applicazioni fruibili tramite browser, più economici grazie all'esternalizzazione. Altra possibilità è data dal Cloud Computing: molti ERP comprendono moduli in modalità SaaS (Software as a Service).

Gli ERP sono per la maggior parte applicativi COTS e in commercio si trovano suite diverse per diversi settori. Gli ERP supportano tutte le attività operative, ma la pianificazione e progettazione della rete sono escluse.

Pur essendo verticalizzati su un settore specifico, avendo le aziende necessità diverse, gli ERP sono configurabili tramite il settaggio di opportuni parametri per adattarsi all'azienda che li utilizza, disattivando o attivando funzionalità o selezionandone una opportuna versione.

I vantaggi conferiti dall'adozione di un ERP sono molteplici: ci sono vantaggi in termini di efficienza (risparmio su tempi e costi delle attività operative) ed efficacia (dati dall'unicità dei dati, dalla capacità di gestione delle informazioni e di standardizzazione delle piattaforme IT) di una azienda.



## Benefici

- **Efficienza operativa**
  - Minori costi di staff (BPR)
  - Minori scorte
  - Minori costi logistici
  - Minori costi di approvvigionamento
  - Maggiore produttività e flessibilità
- **Efficacia operativa**
  - Maggiore tasso di evasione ordini
  - Migliorata capacità di risposta al cliente
  - Minor tempo nel prendere decisioni
  - Miglior servizio e migliore informazione al cliente
- **IT**
  - Standardizzazione delle piattaforme IT
- **Valore dell'informazione**
  - Condivisione globale della informazione
  - Maggiore capacità decisionale e velocità di adeguamento alle variazioni

## CRM

Il CRM (Customer Relationship Management) è una suite software che supporta le organizzazioni nelle interazioni con i clienti. Il suo principale utilizzo è approfondire i bisogni dei clienti e i loro comportamenti al fine di migliorare l'utilizzo dei canali di comunicazione con gli stessi.

Non tutte le aziende possono trarre vantaggio dall'adozione di un CRM, ma la sua utilità cresce con il grado di interazione con la clientela del core business dell'azienda.

I punti chiave di un CRM sono il supporto alla relazione con il cliente attraverso molteplici canali finalizzato al contatto in entrata e in uscita, il supporto post-vendita, la gestione della lealtà e l'analisi del comportamento del cliente. Questo sistema aiuta nel miglioramento degli obiettivi aziendali, a caratterizzare i clienti, nel progetto delle campagne di marketing. Viste queste sue funzioni, il CRM è una strategia vera e propria adottata dall'organizzazione.

L'acquisizione di un CRM funziona nello stesso modo di quella di un ERP, esistono infatti molti COTS con funzionalità e modalità diverse.

I benefici dell'adozione di un CRM consistono nell'incremento della soddisfazione del cliente, miglioramento dell'efficienza dei call center, supporto all'acquisizione di nuovi clienti, semplificazione dei processi di marketing e di vendita e miglioramento dell'efficienza degli agenti di vendita.

## Vantaggi e svantaggi

### Vantaggi

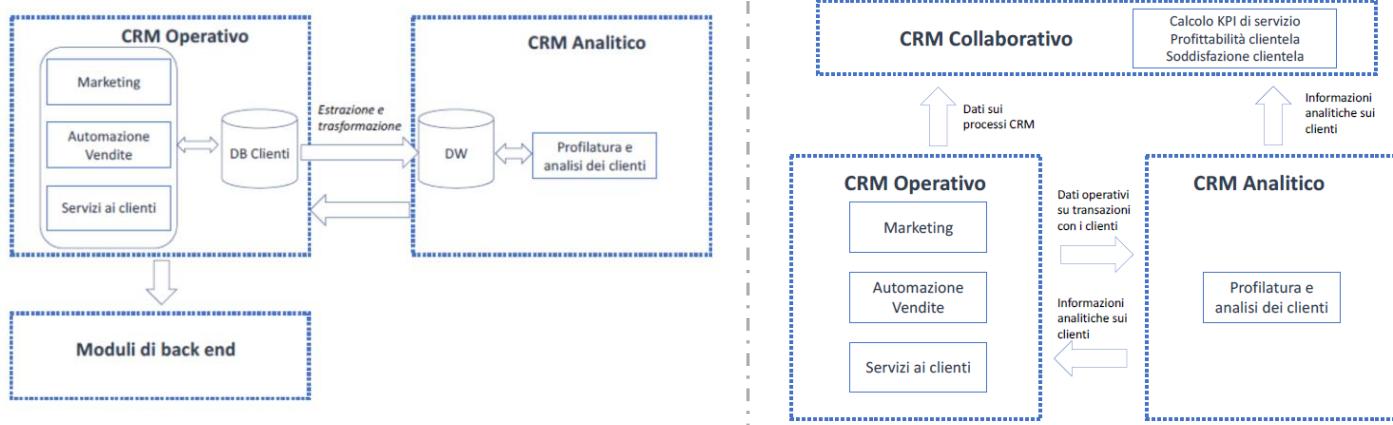
- Aumenta la soddisfazione dei clienti
- I servizi sono offerti in modo migliore e più veloce
- Le informazioni sui clienti sono integrate
- Migliora il valore del brand

### Svantaggi

- Investimento significativo



Il CRM ha due componenti: il **CRM operativo**, che è il modulo front end a supporto delle interazioni giornaliere, e registra tutte le interazioni tra azienda e clienti in un'unica base di dati, e il **CRM analitico**, che è il modulo di back end che si occupa dell'analisi dei dati dei clienti, basandosi sui dati estratti da un data warehouse costruito a partire dalle informazioni estratte dalla base di dati del CRM operativo. I dati del CRM sono instradati anche verso i sistemi di back end che ne necessitano. Esiste anche un terzo modulo, il **CRM collaborativo**, che si prepone di calcolare indicatori, come profitabilità e soddisfazione dei clienti, e raccogliere altri dati da condividere con diversi settori dell'azienda.



## CRM operativo

Si occupa dell'interazione clienti-azienda ed è composto da diversi componenti per supportare tre attività principali:

### MARKETING:

La strategia delle aziende si basa sempre più spesso sulla fidelizzazione dei clienti già esistenti, quindi sul vendere più prodotti a clienti già acquisiti. Ciò è realizzato tramite la progettazione di campagne di marketing, attuata tramite l'acquisizione e l'analisi dei dati. Un dipartimento di marketing dispone in genere dei seguenti sistemi:

- Generazione liste clienti: liste di clienti adatti a ricevere tipi di comunicazioni di marketing.
- Gestione campagne: moduli utili all'automatizzazione di attività e processi di marketing guidando gli utenti nella definizione, pianificazione e analisi dei risultati delle campagne di marketing.
- Cross-selling e Upselling: moduli che aiutano il dipartimento ad attuare strategie di Cross-selling (vendere il maggior numero di prodotti possibile a un cliente) e di Upselling (vendere oggetti di maggior valore possibile a un cliente)

### AUTOMAZIONE VENDITE:

I moduli per il supporto alle vendite possono gestire tutti i canali e si possono dividere in:

- Gestione vendite: offre supporto a tutte le fasi di un processo di vendita, dalla selezione dei contatti al supporto dell'agente di vendita. Inoltre, sono registrate tutte le operazioni permettendo un'analisi dell'operato dell'agente.
- Gestione contatti: si occupa della gestione dei dati del cliente e identifica potenziali clienti per le vendite future, mantiene informazioni utili per migliorare il rapporto con il cliente.
- Gestione opportunità: cerca nuovi clienti o organizzazioni per le vendite future, facendo risparmiare tempo ad agenti e ufficio marketing.

### SERVIZI AI CLIENTI:

I moduli per il servizio ai clienti si occupano di curare le relazioni con i clienti dopo che la vendita è stata effettuata. Questi moduli sono composti da sottomoduli:

- Contact Center: si occupano della gestione delle chiamate inbound e outbound. Il CRM registra le interazioni con il cliente e fornisce funzionalità per il riconoscimento del cliente e la gestione della chiamata.
- Web based self-service: permette ai clienti di trovare sul web soluzioni ad eventuali problemi.
- Call scripting: modulo che fornisce agli operatori una base di dati contenente le soluzioni per risolvere i problemi sollevati dai clienti.

## CRM analitico

Il CRM analitico utilizza i dati raccolti dal CRM operativo per analizzare le preferenze e il comportamento dei clienti, estrarne pattern significativi e supportare al meglio il processo decisionale. Ciò è realizzato utilizzando strumenti quali il data warehouse e sistemi di data mining. Le funzionalità principali sono:

- Reporting: capire chi sono i clienti.
- Analysis: segmentazione dei clienti per categorie.
- Predicting: predire le azioni e i desideri dei clienti.

I risultati delle analisi, oltre ad essere utilizzati a livello decisionale, sono inviati al CRM operativo che ne fa uso nelle modalità già discusse.

## Data Warehousing

Ai livelli più alti della piramide di Anthony, c'è la necessità di accedere in modo veloce ed efficace a dati aggregati in tempo reale, per eseguire interrogazioni complesse al fine di estrarre informazioni di interesse da diverse fonti, interne ed esterne. Gli strumenti utili a questa mansione rientrano nella BI, Business Intelligence.

Sono utilizzati dei sistemi di tipo OLAP, detti **data warehouse** (DW), con le seguenti caratteristiche:

- Gli utenti sono manager di alto e basso livello, che accedono ai dati in lettura.
- Il sistema è **orientato alle entità**, cioè considera i principali oggetti di analisi come vendite ed ordini.
- I dati sono sia storici che attuali, completi di un'etichetta temporale. Si dice che è un sistema **variabile nel tempo**.
- I dati provengono dalle basi di dati operazioni e da fonti esterne, e sono prelevati disomogenei e resi consistenti e integrati per avere una visione completa. Si dice che è un sistema **integrato**.
- Il sistema è **persistente**: una volta inseriti i dati non vengono solitamente modificati.
- I dati sono aggregati e organizzati secondo strutture che ne facilitano l'analisi.
- La struttura è il più semplice possibile e deve mettere in relazione tutti e soli i dati di interesse.
- Le fonti di dati devono essere integrate ed aggiornate.
- Gli strumenti di analisi devono avere brevi tempi di risposta.

Per rappresentare i dati si ricorre al *modello multidimensionale*, in cui l'informazione è rappresentata tramite il concetto di *iper cubo*, formato da un numero variabile di dimensioni N, in cui ogni dimensione rappresenta una dimensione di analisi per la base di dati.

Gli elementi costitutivi di una base di dati multidimensionale sono:

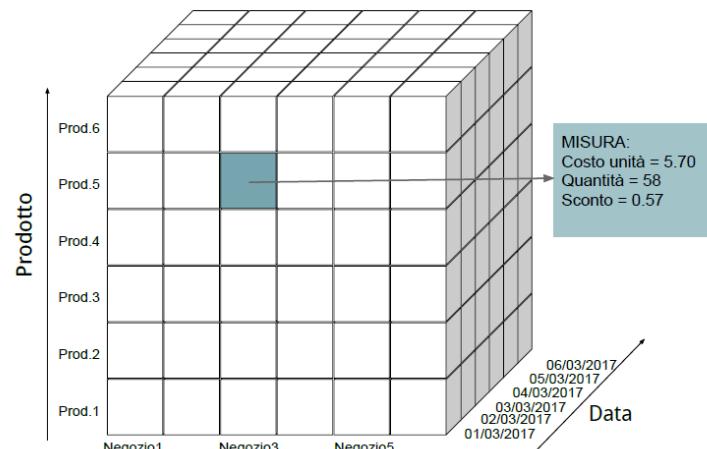
- **FATTO**: l'elemento dell'iper cubo ottenuto specificando un valore per ogni dimensione.
- **DIMENSIONE**: Insieme delle coordinate degli elementi, che corrispondono alle dimensioni di analisi.
- **MISURA**: valore quantitativo del fatto elementare.

Le dimensioni possono essere numerose e organizzate in gerarchie, con livelli di granularità che variano da livello a livello. Una dimensione può essere manipolata cambiando il livello di granularità, selezionando il livello di dettaglio di interesse per l'analisi.

Un data warehouse, in quanto un sistema OLAP, gode delle proprietà FASMI: deve avere una visione multidimensionale dei dati (Multidimensional), che devono contenere tutte le informazioni di interesse (Informational), su cui permette di effettuare analisi complesse (Analytical) a più utenti, ma con diversi permessi di accesso ai dati (Shared), rispondendo alle loro richieste in tempo ridotto (Fast).

La costruzione dei data warehouse ha un approccio iterativo, sono infatti costruiti ipercubi a partire dai fatti di maggiore interesse e successivamente si aggiungono e popolano altri ipercubi.

La creazione è un'operazione onerosa, ma anche il mantenimento e il continuo aggiornamento dei dati richiede un discreto sforzo.

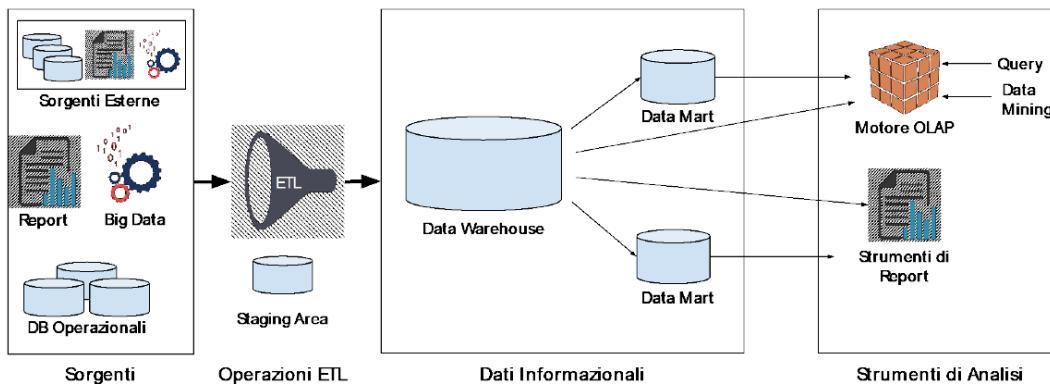


Prodotto	—	Tipo	—	Categoria
ID Prodotto		Pesce		Alimentari
		Carne		Detersivi
		Bagnoschiuma		Bibite
		Biscotti		

## Architettura del data warehouse

Un DW è composto da più basi di dati organizzate in modo gerarchico:

- SORGENTI: Basi di dati da cui sono estratti i dati che popoleranno il data warehouse.
- STAGING AREA: Area in cui transitano i dati per essere sottoposti alle operazioni dette **ETL** (Extraction, Transformation, Loading) che consistono nell'estrazione dei dati dalle sorgenti, nella loro organizzazione secondo la struttura multidimensionale desiderata e nel loro caricamento nel DW. Questa area può non essere presente, se le operazioni ETL sono svolte altrove.
- IL DATA WAREHOUSE vero e proprio.
- DATA MART: basi di dati che, se presenti, si trovano a valle del DW, sono data warehouse tematici dalle dimensioni ridotte atti a contenere informazioni che sono in genere di interesse a solo parte degli utenti, che quindi non servono siano nel data warehouse principale. Queste permettono l'implementazione di una architettura distribuita, al contrario se non presenti si ha una architettura centralizzata.



### OPERAZIONI ETL:

L'**estrazione** dei dati definisce quali siano i dati che devono essere estratti e come questo processo debba avvenire. Essa può essere *statica*, se vengono considerati tutti i dati presenti nelle sorgenti, o *incrementale*, se vengono considerati solo i dati nuovi o modificati nel periodo di tempo che intercorre dall'ultima estrazione avvenuta.

Dopo l'estrazione i dati subiscono una **trasformazione**, in quanto derivano da fonti eterogenee e devono dattarsi ad un unico formato adatto al data warehouse. Vengono svolte le seguenti operazioni:

- PULIZIA DEI DATI (data cleaning): correzione degli errori presenti nei dati, come la mancanza di valori o la presenza di valori inammissibili, o l'inconsistenza tra valori presenti in campi che hanno vincoli.
- RICONCILIAZIONE: operazione che mira a mettere in relazione i dati relativi al medesimo oggetto.
- STANDARDIZZAZIONE DEI FORMATI: operazioni di standardizzazione per rendere omogenei i dati.
- RICERCA E ELIMINAZIONE DEI DUPLICATI: si riducono ad una sola istanza le informazioni ridondanti.

L'ultima fase è il **caricamento** dei dati nel data warehouse ciò avviene seguendo il modello multidimensionale.

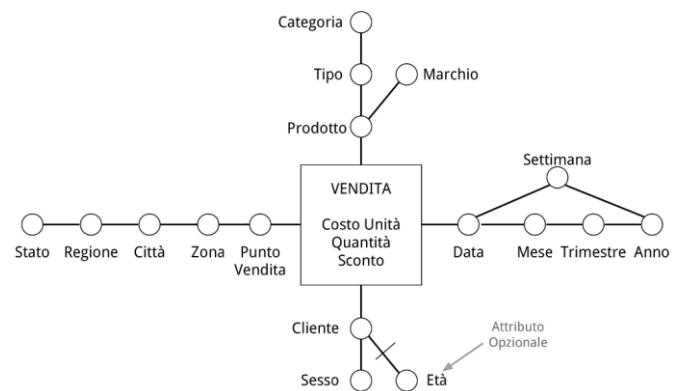
Le funzionalità ETL sono supportate e documentate da **metadati** che interessano:

- Struttura del DW
- Metadati operazionali: si riferiscono alla storia dei dati
- Metadati per mappare i dati operazionali ai dati caricati nel DW: informazioni sulle sorgenti e sul loro contenuto
- Statistiche d'uso

## Modello concettuale del data warehouse

Il modello più utilizzato per la rappresentazione dei sistemi multidimensionali è il **Dimensional Fact Model** (DFM). In esso il fatto è rappresentato come un rettangolo contenente le misure corrispondenti, mentre le dimensioni sono rappresentate come cerchi etichettati e collegati al fatto, e possono essere attributi del fatto oppure gerarchie ad albero. Alcuni attributi delle gerarchie possono essere opzionali e ciò è indicato con una barra sulla linea corrispondente all'attributo.

Il data warehouse non è composto da un solo ipercubo, ma da molteplici, ognuno che modella le dimensioni di interesse per una analisi. Per ognuno di questi ipercubi è modellato un DFM.



## Modelli logici del data warehouse

Il modello concettuale deve essere tradotto in modello logico, scegliendo quale DBMS utilizzare per gestire l'informazione multidimensionale.

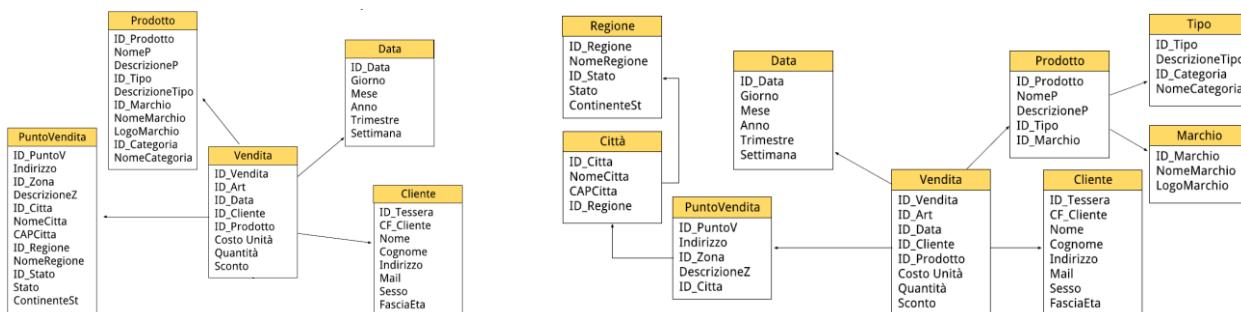
**MODELLI MOLAP:** sta per *Multidimensional OLAP* e traduce il modello concettuale in modo esatto in una base di dati. Ciò consente di sfruttare al meglio tutti i vantaggi di una base di dati multidimensionale e si ottengono interrogazioni efficienti e veloci. Tuttavia, l'utilizzo di queste basi di dati, che sono poco diffuse, richiede conoscenze specifiche che l'utente medio non possiede. Inoltre, questo modello è molto dispendioso dal punto di vista dell'allocazione della memoria, in quanto per ogni possibile combinazione di valori delle dimensioni è allocata una cella di memoria per il fatto corrispondente, anche se questo non si è verificato e la cella rimane vuota.

**MODELLI ROLAP:** sta per *Relational OLAP* e traduce il modello concettuale in un modello relazionale, su cui è possibile svolgere interrogazioni con linguaggi basati su SQL. Questo modello può essere utilizzato da utenti che non possiedono conoscenze specifiche, e richiede molta meno memoria di quella che richiederebbe un sistema basato su un modello MOLAP. Esistono però diversi svantaggi dovuti al fatto che non si aderisce al modello concettuale dei dati e le interrogazioni sono lente e macchinose.

**MODELLI HOLAP:** sta per *Hybrid OLAP* e traduce il modello concettuale in modo ibrido, solitamente utilizzando una base di dati di tipo relazionale, limitando lo spazio utilizzato e rendendo accessibili le interazioni a più utenti. Solitamente quando si utilizzano questi modelli per le data warehouse, invece i data mart tematici, avendo dimensioni limitate, sono implementati con basi di dati multidimensionali.

Per mappare una base di dati multidimensionale in uno schema relazionale bisogna identificare le tabelle che compongono lo schema, ed esistono due approcci principali:

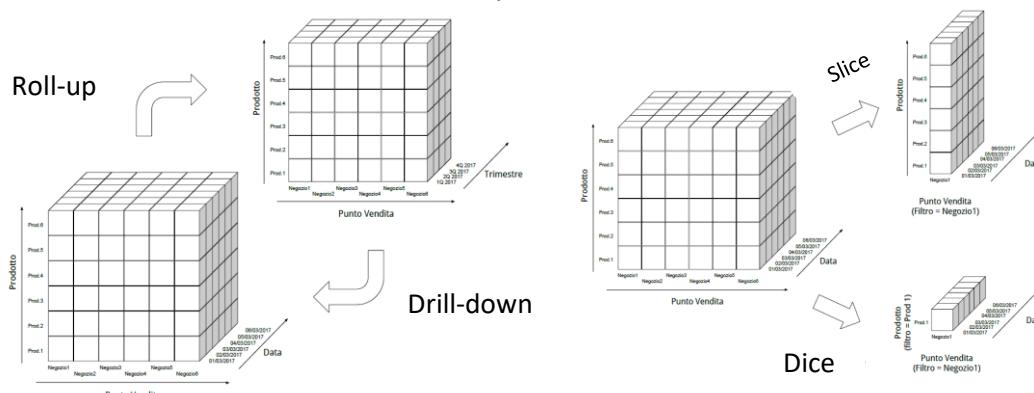
- Nello **schema a stella** sono utilizzate due tipologie di tabelle: le tabelle dei fatti, che contengono tutti gli attributi corrispondenti alle misure del fatto, quindi tutti i fatti specifici, e le tabelle delle dimensioni, che contengono per ogni dimensione tutti gli attributi relativi alla gerarchia corrispondente. Viene persa la gerarchia delle dimensioni, che viene appiattita in un'unica tabella.
- Nello **schema a fiocco di neve** ad ogni dimensione sono associate più tabelle con le opportune relazioni, così da conservare le dipendenze funzionali più rilevanti per gli utenti.



## Operazioni sul data warehouse

I data warehouse prevedono un insieme di operazioni utili all'analisi dei dati.

- Drill-down:** si ottengono dati più dettagliati scendendo lungo la gerarchia di una dimensione, quindi passando ad un livello di aggregazione minore.
- Roll-up:** operazione inversa al drill-down, in cui si passa ad aggregazione maggiore, diminuendo il livello di dettaglio.
- Slice:** si focalizza l'analisi su una porzione dei dati selezionando un valore per una delle dimensioni di analisi.
- Dice:** si identificano precisi insiemi di coordinate a qualsiasi livello gerarchico sia utile, per qualsiasi dimensione desiderata, ottenendo un ipercubo contenente un insieme ridotto dei dati.

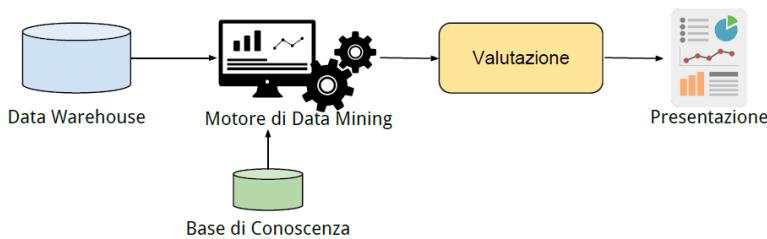


## Data mining

Il **data mining** è l'attività volta a riconoscere ed estrarre in modo automatico informazioni nascoste da basi di dati di grandi dimensioni. In alcune fasi è richiesta l'interazione con l'utente.

Il data mining è un processo composto da diversi passi successivi, che possono essere iterati più volte per raffinare l'operazione ottenuta. Questi passi sono:

1. Creazione del data warehouse da utilizzare
  - 1) Selezione dei dati ritenuti di interesse
  - 2) Pulizia dei dati (cleaning)
  - 3) Integrazione dei dati provenienti da fonti diverse
  - 4) Trasformazione e strutturazione dei dati ottenuti
2. Data mining
  - 5) Ricerca di condizioni notevoli all'interno dei dati
  - 6) Valutazione dei risultati e loro riduzione ai soli ritenuti rilevanti
  - 7) Presentazione dei dati finali all'utente



Ci sono diverse classificazioni delle tecniche di data mining: esse possono essere tecniche in apprendimento *supervisionato* o *non supervisionato*, oppure si può distinguere tra tecniche *predittive*, che analizzano i dati del passato per supportare predizioni, o *descrittive*, che li analizzano e li categorizzano.

Esistono diverse tecniche di data mining, tra cui le regole associative, la classificazione e il clustering.

### REGOLE ASSOCIATIVE:

Le regole associative (espresse solitamente nella forma  $A \Rightarrow B$ ) descrivono relazioni che intercorrono tra diversi attributi della base di dati e sono utilizzate per identificare correlazioni tra elementi, per esempio nella basket analysis si può identificare i prodotti che un utente è propenso a comprare assieme.

Spesso sono utilizzate per la pianificazione di campagne promozionali efficaci e per l'attivazione di azioni preventive di manutenzione degli impianti.

### CLASSIFICAZIONE:

È una tecnica che assegna in modo automatico gli elementi della base di dati a classi predefinite. Si parte da un insieme di elementi per i quali è nota la classe di appartenenza, e si costruisce un modello con lo scopo di identificare quali sono le caratteristiche che permettono di assegnare un elemento ad una classe. L'insieme degli elementi di cui si conosce a priori la classe utilizzati per costruire il modello è chiamato **training set**, mentre l'insieme degli elementi usati per testare i risultati della classificazione effettuata è chiamato **test set**. Un modello è migliore a seconda di quanto si dimostra in grado di classificare in modo corretto gli elementi del test set. I classificatori sono di tipi diversi e si basano su tecniche di diversa natura (matematica, alberi di decisione, reti neurali, etc) e si distinguono in base a un insieme di caratteristiche: l'accuratezza, la velocità, la scalabilità, la robustezza e l'interpretabilità.

Un modello di classificazione è quello basato sugli *alberi di decisione*. In una prima fase, detta *fase di build*, avviene la costruzione dell'albero, inizialmente a partire da una base di dati storica. Successivamente, nella *fase di pruning*, l'albero è modificato attraverso raffinamenti algoritmici successivi che comportano la scelta della radice, della profondità, e la partizione del dataset.

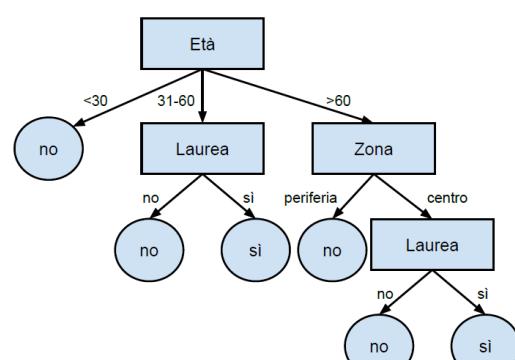
Per la valutazione del risultato ottenuto tramite il test set, si utilizzano diversi parametri calcolati a partire dal numero di falsi e veri positivi e falsi e veri negativi:

$$\text{Precision} = \frac{\text{VeriPositivi}}{\text{VeriPositivi} + \text{FalsiPositivi}}$$

La Precision indica con che accuratezza un albero di decisione inserisce gli individui in una classe, ha buoni valori a partire dall'80%.

$$\text{Recall} = \frac{\text{VeriPositivi}}{\text{VeriPositivi} + \text{FalsiNegativi}}$$

La Recall indica quanti elementi l'algoritmo ha assegnato ad una classe rispetto a quelli che idealmente la costituiscono, quindi il "richiamo" nell'area degli elementi selezionati.



## CLUSTERING:

Come la classificazione, il clustering è una tecnica che permette di raggruppare in classi un insieme di elementi in base alle loro caratteristiche. A differenza della classificazione, non richiede però la conoscenza a priori delle classi di appartenenza dell'insieme di training, ma ricerca le classificazioni e talvolta il numero di classi (cluster) in modo automatico basandosi su similarità nei valori. Ad un cluster non è assegnata una categoria chiara, in quanto l'algoritmo raggruppa gli elementi in base ad affinità ignorando il loro significato. L'algoritmo più diffuso di clustering è il *k-means*.

Gli algoritmi di clustering si classificano in base alla modalità di assegnamento degli elementi ai cluster. Un algoritmo di clustering è *esclusivo* se assegna ogni elemento del dataset ad un solo cluster, altrimenti si dice *sovraposto*. I metodi *fuzzy* sono metodi sovrapposti in cui l'associazione degli elementi ad ogni classe è associata ad un peso che varia tra 0 e 1. Infine, un metodo è *completo* se ogni elemento appartiene ad almeno una classe, altrimenti è detto *parziale*.

Un esempio dell'utilizzo da parte dei CRM di metodi di classificazione come il clustering è la segmentazione dei clienti in gruppi di clienti simili al fine di individuare le strategie di marketing adottare.

## Selezione del software

Quando si acquista un software COTS è necessario selezionare il pacchetto adeguato alle esigenze dell'azienda. Fondamentale è quindi la fase di raccolta e analisi dei requisiti (Ingegneria dei requisiti) funzionali e non funzionali.

Gli indicatori usati per differenziare le soluzioni di mercato sono:

### INDICATORI FUNZIONALI:

- **Grado di completezza** (copertura funzionale): rapporto tra la somma delle funzionalità fornite e il numero di funzionalità richieste.
- **Personalizzabilità**: quanto la soluzione può essere adattata alle esigenze specifiche dell'azienda tramite parametrizzazione e personalizzazione.

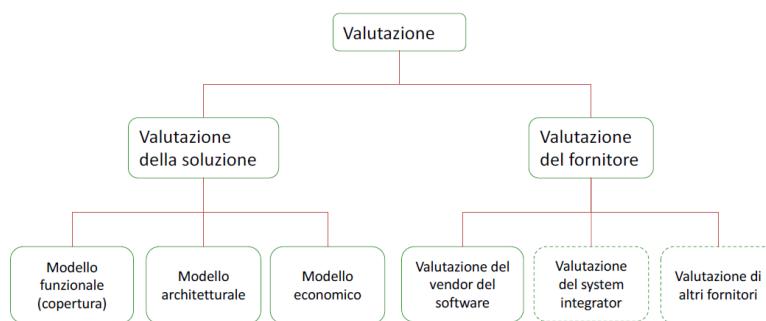
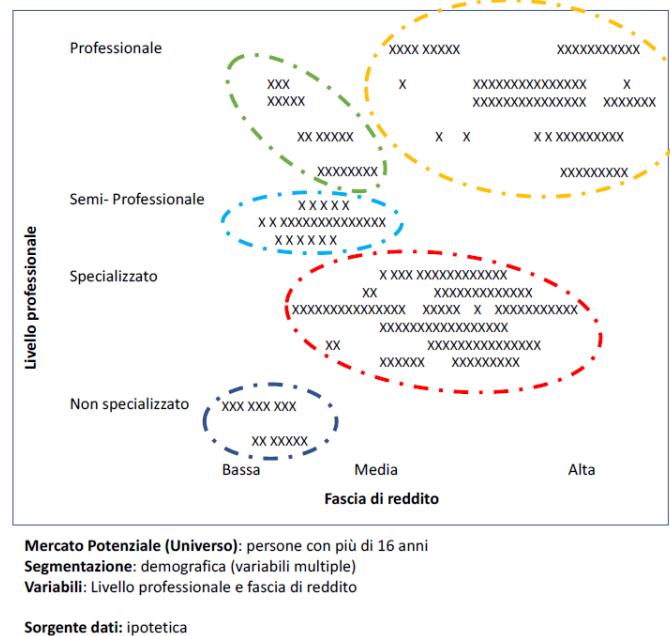
### INDICATORI ARCHITETTURALI:

- **Scalabilità dell'infrastruttura**: il sistema deve essere scalabile nel tempo, in modo da mantenere inalterate le prestazioni, o poterle migliorare, con l'aumento del carico transazionale. Questo indice è valutabile considerando il numero di stazioni gestibili dal software, il numero di livelli dell'architettura o il metodo di gestione della base di dati.
- **Grado di interoperabilità della soluzione**: quanto l'operativo è un'entità aperta verso l'esterno.
- **Livello di sicurezza del sistema**: sicurezza del sistema in termini di regolazione degli accessi e in termini di protocolli adottati.

Una volta appurato che il prodotto soddisfi gli indicatori funzionali e non funzionali richiesti, si valuta l'affidabilità dei fornitori dei pacchetti. La **credibilità** e la **capacità** del produttore (vendor) si valutano considerando l'esperienza, la presenza a livello territoriale, la reputazione e l'attenzione all'evoluzione e aggiornamento del prodotto nel tempo.

Ultimo step è la valutazione economica, dato un budget infatti è necessario considerare i **costi della soluzione** per verificare la fattibilità dell'investimento. I costi possono essere classificati come *investimenti di progetto*, legati ai costi delle licenze, dell'hardware, delle risorse umane e i costi di change management. Bisogna poi considerare i **costi di gestione** che comprendono manutenzione, costi infrastrutturali, costi di personalizzazione e di aggiornamento del software.

Quando sono verificati tutti questi indicatori, le soluzioni non scartate vengono analizzate in dettaglio attraverso demo e incontri con i fornitori.



# Capitolo 9 – Tecnologia a livello di piattaforma

## Architetture di integrazione

In un sistema informativo, ci sono diversi moduli applicativi, collegati tra loro da flussi informativi che seguono le business rules. A causa della loro natura eterogenea, è necessario implementare meccanismi di integrazione per garantire uno scambio efficace di informazioni tra i moduli. Le **architetture di integrazione** sono soluzioni tecniche e organizzative volte a garantire il coordinamento delle attività intra-organizzative e inter-organizzative.

Per quanto riguarda l'**integrazione dei dati**, i problemi che riguardano i flussi informativi nei sistemi OLTP e OLAP sono di formato e di semantica dei dati.

Il *formato* descrive come i dati sono salvati e presentati, i problemi riguardano:

- Rappresentazione: la stessa informazione può essere rappresentata in modi diversi in moduli differenti.
- Struttura: nella stessa rappresentazione, i campi possono essere ordinati in modo diverso.
- Presentazione: lo stesso dato può essere mostrato all'utente in formati diversi.

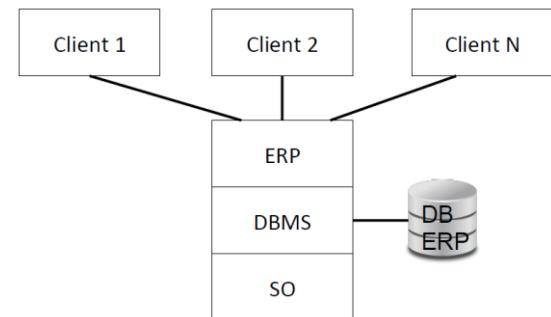
La *semantica* riguarda il significato attribuito all'informazione, per esempio in sistemi diversi per lo stesso tipo di dati ci si può riferire a diverse unità di misura, e di ciò bisogna tener conto nell'interazione tra i sistemi.

Invece, l'**integrazione dei processi** richiede di definire quali siano le informazioni scambiate tra moduli applicativi distinti, in un contesto intra-organizzativo o inter-organizzativo, e le modalità di interazione tra processi: servono meccanismi di sincronizzazione, per la gestione di messaggi ed eventi, interfacce, e stabilire l'ordine dei messaggi.

## Evoluzione dei moduli applicativi

Gli elementi di una architettura funzionale sono composti, a livello di piattaforma, da un insieme di applicazioni, che forniscono le funzionalità previste, e da un sistema di gestione dei dati, solitamente gestiti tramite DBMS, che a loro volta necessitano di funzioni del sistema operativo (SO).

Le organizzazioni moderne utilizzano un gran numero di applicazioni che interagiscono tra loro e condividono dati, quindi una modifica dei dati da parte di una applicazione deve riflettersi sulle altre. Sarebbe possibile effettuare un allineamento periodico dei dati replicati, ma ciò non è ideale, quindi il problema dell'integrazione è risolto con la condivisione tra le applicazioni di un'unica base di dati e con la definizione intrinseca dei processi nella logica dei singoli moduli.



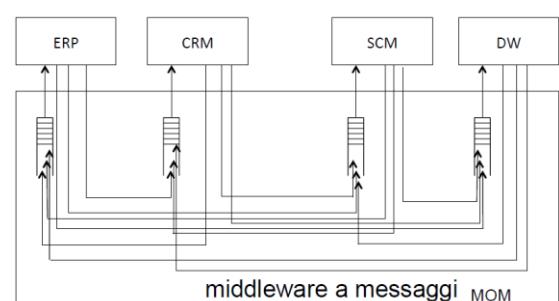
## Piattaforme di integrazione

Vi sono diverse architetture di integrazione che consentono una condivisione efficiente dei dati tra applicazioni.

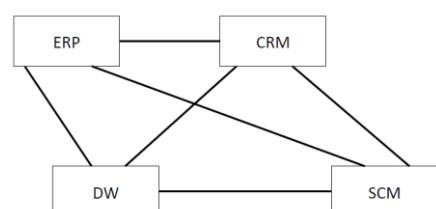
### ARCHITETTURE DI INTEGRAZIONE PUNTO-A-PUNTO:

Si realizza una interfaccia diretta tra i sistemi che si devono scambiare i dati. Le applicazioni di integrazione forniscono meccanismi di trasformazione dei dati e gestiscono lo scambio dei messaggi tra i sistemi.

Le due soluzioni tecnologiche sono quelle che prevedono o il collegamento diretto tra le applicazioni o la presenza di un **middleware a messaggi** (Message-Oriented Middleware, MOM) che semplifica lo scambio di messaggi tra applicazioni, ma rimane una soluzione punto-a-punto.



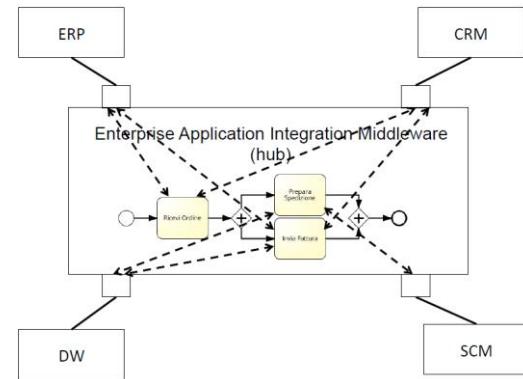
Questa tipologia di soluzioni comporta diversi svantaggi. La logica di integrazione è cablata all'interno di ogni sistema, rendendo complicata la sostituzione dei componenti, che sia per un aggiornamento o per una modifica del sistema, quindi si ha un grande sforzo di manutenzione. Inoltre, è necessario implementare un numero di interfacce dell'ordine di  $N^2$ , quindi oltre a comportare alti costi, questa soluzione non è scalabile, perché la modifica di una applicazione comporta la modifica di  $N - 1$  interfacce.



## ARCHITETTURE DI INTEGRAZIONE HUB-AND-SPOKE:

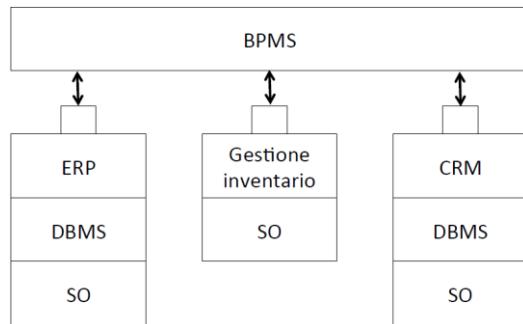
Il paradigma **hub-and-spoke** è basato sulla presenza di un *hub* centrale, al quale vengono collegati i sistemi tramite degli *spoke*, che fungono anche da adattatori. Un messaggio viene inviato all'hub centrale, che si occupa del suo instradamento verso la sua destinazione.

Il numero di interfacce da implementare si riduce a  $N$ , e la sostituzione di un componente comporta solo quella del suo spoke. La soluzione è quindi scalabile, ma mantiene una certa complessità nell'implementazione degli adapter e dell'hub, contenenti la logica applicativa e sistemi opzionali.



## WORKFLOW COMPONENT E WORKFLOW MANAGEMENT SYSTEM:

Un **Business Process Management System** (BPMS), detto anche **Workflow component**, consente di gestire lo stato di avanzamento dei processi e i flussi informativi tra applicazioni. Esso consente di separare le applicazioni dalla loro logica di composizione all'interno del modulo applicativo, rendendo facili le modifiche del modulo applicativo stesso. Un BPMS può far parte di una suite ERP e rendere possibile il coordinamento tra le funzionalità offerte da moduli funzionali distinti. Il componente di **Workflow Management System** (WFMS), anche esso un BPMS, è utile a fornire la logica di integrazione quando si devono integrare sistemi forniti da produttori diversi. Questi sistemi consentono l'integrazione, ma non forniscono interfacce ai sistemi, possono quindi essere implementati in un hub di un sistema hub-and-spoke.



## Modelli e servizi

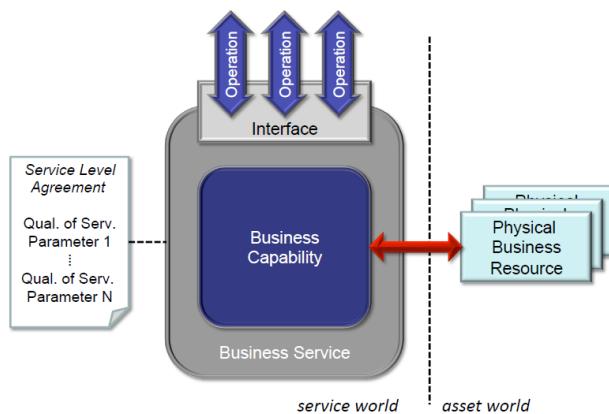
Un paradigma di progettazione software è quello realizzato mediante le **architetture orientate ai servizi** (Service Oriented Architecture – SOA), in cui le funzionalità sono realizzate in modo modulare tramite servizi interrogabili in remoto tramite http. Un **servizio** è la realizzazione di un modulo di una funzionalità di business, contenuta e indipendente dalle altre, che può essere invocato dagli utenti del servizio stesso, che ne conoscono solo l'interfaccia, quindi le operazioni attraverso cui il servizio può essere utilizzato.

Per assicurarsi che un servizio sia in grado di fornire una certa funzionalità richiesta, si stipula con i consumatori un accordo di qualità, chiamato **Service Level Agreement** (SLA), che definisce le caratteristiche non funzionali che il servizio deve garantire, in termini di parametri di *qualità del servizio* (QoS). Ogni parametro specifica una o più metriche che possono essere utilizzate per monitorare l'esecuzione del servizio e per accertarsi esso fornisca una risposta valida.

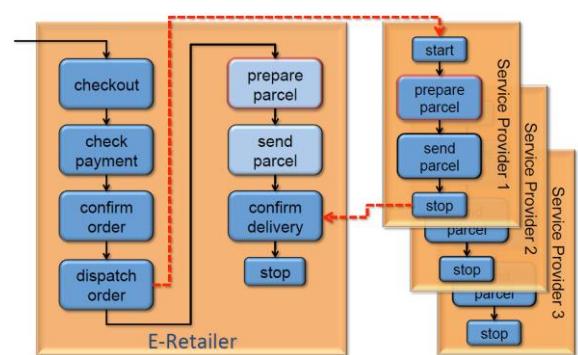
Nelle SOA, mediante lo scambio di messaggi in internet con HTML e codifica XML, ciascun *Service Provider* pubblica le informazioni sui servizi offerti in un **Service Registry**, descrivendo i servizi in termini di operazioni e messaggi scambiati. L'utente interessato ad una funzionalità deve quindi interrogare il Service Registry per conoscere i servizi disponibili che la realizzino, e dalla descrizione ottenuta saprà che messaggi scambiare, quali protocolli utilizzare e come accedere al servizio.

Nello stile architettonico denominato *REST – Representation State Transfer*, si parla di *RESTful services* quando le richieste vengono effettuate direttamente a una risorsa URI (cioè tramite il suo Uniform Resource Identifier), e si parla di sistemi *REST web based* quando la risposta viene fornita in un formato predefinito (come XML o HTTP).

È possibile comporre servizi in uno più complesso, in modo da utilizzarli in modo congiunto per fornire tutte le funzionalità richieste da una applicazione. Questo è possibile facendo ricorso a tecnologie standard che consentono la comunicazione e la cooperazione tra componenti mediante la rete internet. In particolare, una **service orchestration** determina la sequenza con cui è necessario invocare i diversi servizi.

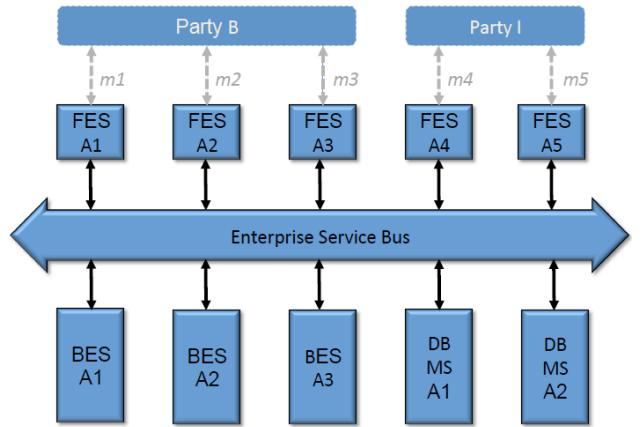


SLA Business Service	Manage Shopping Cart	
Response Time	Maximum	0.25 sec
Availability	Average	0.10 sec
Cost	Minimum	99.99 %
	Average	€ 0.02



Per ottenere flessibilità, si utilizza una rete dinamica di servizi di business, nella quale si creano servizi completi componendo in modo dinamico i servizi base forniti dalle varie organizzazioni. Gli elementi che compongono un servizio sono selezionati al momento di necessità, utilizzando il service registry. Sul mercato esistono dei **Service Broker**, organizzazioni che si occupano di trovare il migliore allineamento tra servizi offerti e richiesti. La responsabilità della sincronizzazione tra i servizi richiesti può essere affidata ad un solo partecipante, detto orchestratore, ma può anche essere distribuita tra i vari partecipanti, si parla allora di una coreografia di servizi, gestita in modalità peer-to-peer.

Ciascun componente di un servizio complesso definisce una propria interfaccia, che specifica quali servizi offre al sistema e quali messaggi devono essere scambiati durante l'utilizzo. L'integrazione dei componenti può essere realizzata utilizzando infrastrutture di integrazione basate sui servizi, tipiche dei sistemi a **Enterprise Service Bus (ESB)**. I componenti di automazione di workflow possono essere utilizzati per realizzare la logica applicativa di integrazione, tutti con accesso all'unico bus di integrazione.



# Capitolo 10 – Tecnologie a livello di architettura fisica

L'architettura fisica di un sistema si adatta alle esigenze informative dell'organizzazione. Ci sono due macrocategorie di architetture a livello di architettura fisica:

- **SISTEMI INFORMATICI CENTRALIZZATI:** i dati e le applicazioni risiedono in un unico nodo elaborativo, ad essi si accede attraverso nodi limitati a periferiche di input e output.
- **ARCHITETTURE DISTRIBUITE:** architetture in cui o le applicazioni (elaborazione distribuita), o il patrimonio informativo (base di dati distribuita), risiedono in più nodi elaborativi distinti.

È possibile strutturare le applicazioni secondo tre livelli logici, chiamati **layer**, che trovano corrispondenza negli strati software nei quali vengono partizionate le applicazioni, e le cui combinazioni di disposizioni determinano il tipo di architettura fisica:

- **Livello di presentazione (P):** gestisce la logica di presentazione all'utente, costituisce il front end dell'applicazione.
- **Livello di logica applicativa (A):** o logica di business, si occupa delle funzioni da mettere a disposizione all'utente.
- **Livello di logica di accesso ai dati (D):** si occupa della gestione dell'informazione, eventualmente con accesso ai database. Insieme al livello A costituisce il back end dell'applicazione.

## Applicazioni distribuite

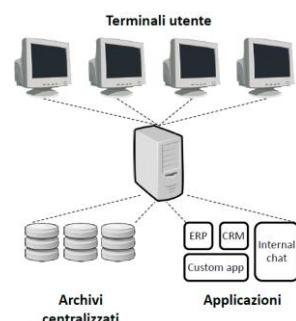
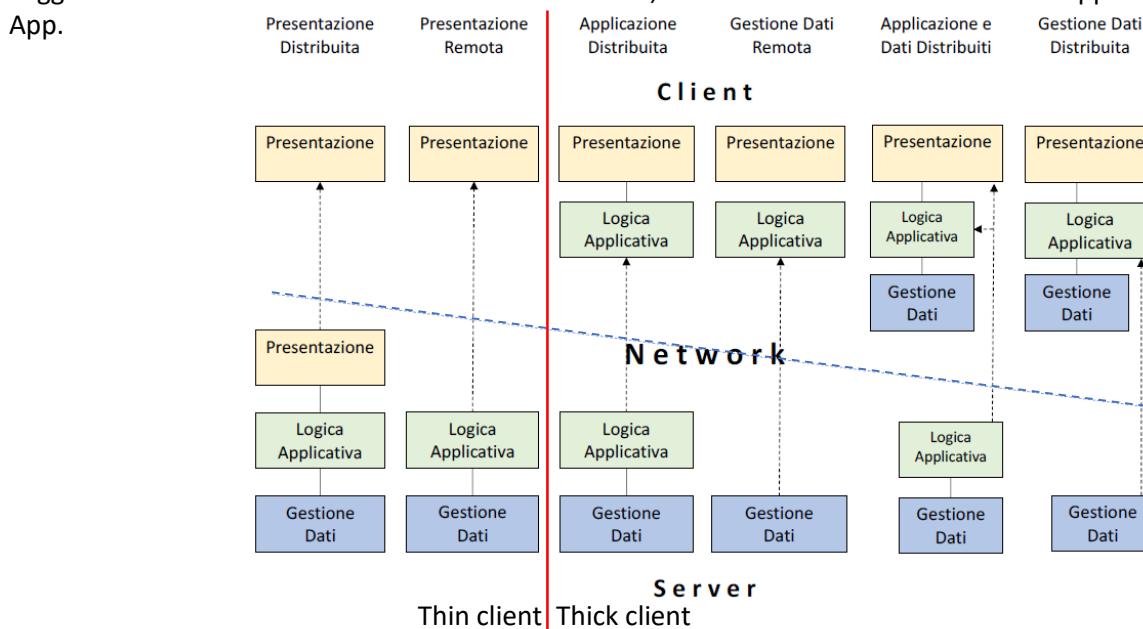
Nei sistemi distribuiti, i livelli logici (layers) vengono installati su livelli hardware, chiamati **tier**. A seconda del numero di livelli si possono avere diversi tipi di architetture:

### ARCHITETTURA SINGLE TIERED:

I tre livelli software sono assegnati ad un'unica macchina, come nei sistemi basati su mainframe. È un approccio ormai obsoleto ma ancora valido, infatti i sistemi mainframe garantiscono alte prestazioni, affidabilità e sicurezza, facilità nella manutenzione. Per contro, una architettura di questo tipo è poco flessibile.

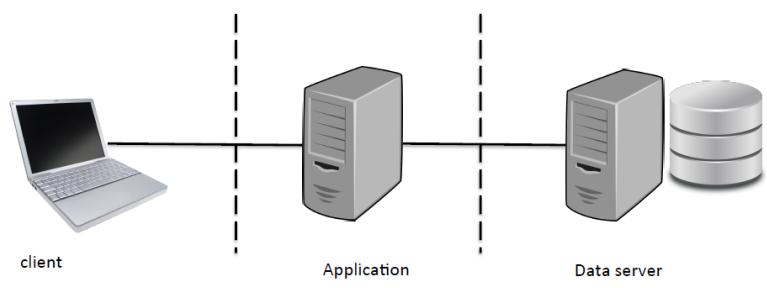
### ARCHITETTURA TWO-TIERED:

Questa architettura prevede due livelli di distribuzione, configurabili secondo sei differenti allocazioni dei layer logici. Si parla di configurazione **thin client** se il nodo client gestisce solo la logica di presentazione, mentre si parla di **thick client** se esso, oltre alla logica di presentazione, gestisce anche parte o l'intera logica applicativa. Una configurazione thick client ha il vantaggio di non richiedere una connessione stabile tra client e server, ma solo nei momenti di necessità, tuttavia ci sono diversi svantaggi che nascono dalla necessità di amministrare diverse macchine client, tra cui anche l'elevato costo degli aggiornamenti. Attualmente si sono diffondendo sistemi di supporto per l'aggiornamento automatico del software a lato client, ciò facilita la diffusione di moduli applicativi su client quali



### ARCHITETTURA THREE-TIERED:

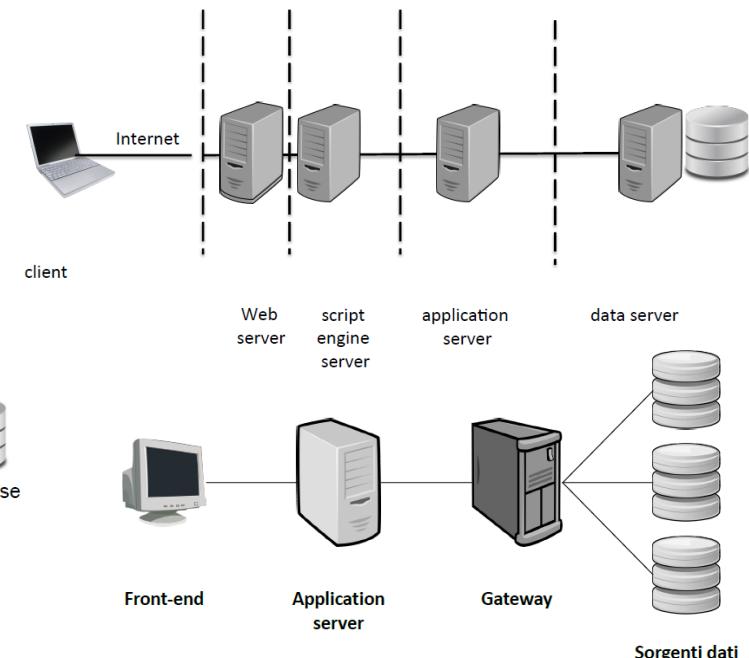
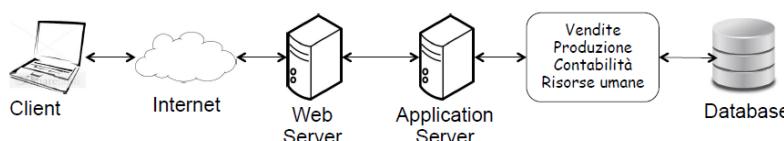
I tre livelli applicativi sono suddivisi fra altrettante macchine dedicate: una stazione di lavoro utente, un server intermedio (middle tier) e un server di gestione dati. Questa architettura fornisce alle infrastrutture maggiori caratteristiche di scalabilità e flessibilità. Le configurazioni thick client con tre tier non sono molto indicate, è meglio centralizzare la logica applicativa su un server intermedio da condividere tra i client, in modo che basti garantire la sua robustezza ed efficienza. Il server intermedio a middle tier, che va dimensionato in modo da garantire un livello di prestazioni per un certo numero massimo prefissato di utenti contemporaneamente, consente inoltre di ridurre il carico del DBMS server, con cui mantiene una connessione permanente.



### ARCHITETTURE N-TIERED:

Sono architetture ad  $n$  livelli, che puntano ad ottimizzare il carico lato server distribuendolo su più nodi, disponendo una o più macchine dedicate a diverse funzionalità.

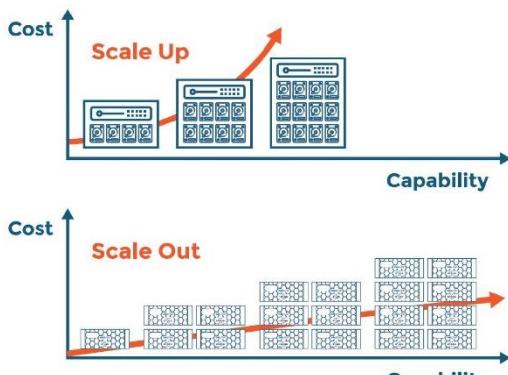
I sistemi informativi basati su web spesso si basano su 5 tier. Il numero di tier può aumentare se vengono introdotti server dedicati al supporto dello strato di comunicazione, ovvero del middleware.



### Scalabilità

La **scalabilità** di una architettura è la capacità dell'infrastruttura di soddisfare richieste crescenti da parte degli utenti con aggiornamenti adeguati. Ci sono due tipi di scalabilità:

- Scalabilità VERTICALE (scale-up): si ottiene un aumento di prestazioni mantenendo inalterato il numero di nodi ma aumentandone la capacità elaborativa. Non sempre ad un aumento di capacità elaborativa corrisponde un proporzionale aumento delle prestazioni.
- Scalabilità ORIZZONTALE (scale-out): l'aumento delle prestazioni avviene attraverso la replicazione di nodi che costituiscono l'eventuale collo di bottiglia. I processi sono così distribuiti tra i nodi replica, alleggerendo il carico dei nodi preesistenti. L'upgrade del sistema impatta sulla struttura dei tier: l'aggiunta di nodi replica richiede l'introduzione di un sistema di *load balancing*, che si occupa di distribuire il carico di elaborazione tra i nuovi nodi. Si utilizza il *principio del downsizing*: a parità di potenza di calcolo complessiva installata, server di fascia bassa hanno costi inferiori a server di fascia alta o mainframe.



In caso di sovrardimensionamento, l'upgrade potrebbe portare al sostenimento di costi non necessari, mentre un sottodimensionamento potrebbe rendere l'upgrade non sufficiente in poco tempo. Il problema del dimensionamento è importante soprattutto in casi in cui il numero degli utenti è una variabile poco predibile, come nel caso di applicazioni Web.

In ambito Web non si parla di scalabilità ma di **elasticità** del sistema, cioè la proprietà del sistema non solo di aumentare le proprie risorse a fronte di picchi di richieste, ma anche di diminuirle quando le richieste diminuiscono.

## Server Farm

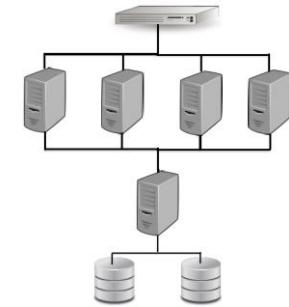
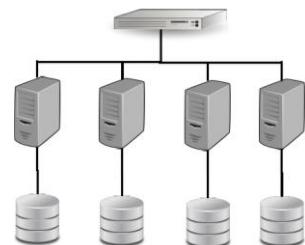
Una server farm è un'organizzazione molto scalabile dei tier fisici, ottenuta seguendo l'approccio scale-out, in un insieme di elaboratori che condividono il carico elaborativo, le applicazioni e, a seconda delle configurazioni, i dati. Le server farm possono essere realizzate secondo due principi progettuali: il *cloning* (clonazione) e il *Partitioning* (partizionamento).

### CLONAZIONE:

Sui server vengono installate le stesse applicazioni software ed i medesimi dati, e le richieste sono instradate usando un sistema di load-balancing. Un *RACS* (Reliable Array of Cloned Services) è un insieme di cloni che implementano tutte le stesse funzionalità e sono dedicati allo svolgimento del medesimo servizio. C'è una elevata tolleranza ai guasti, infatti se un clone si guasta, il servizio può continuare ad essere erogato da un altro nodo.

Esistono due configurazioni possibili per i RACS:

- **Shared-nothing:** i dati sono replicati su ogni clone in un disco locale ad ognuno di essi. Un aggiornamento dei dati deve essere applicato ad ognuno di essi, quindi questa configurazione è poco performante per servizi in cui sono svolte molte operazioni di write, al contrario ha prestazioni migliori per servizi di tipo read-only.
- **Shared-disk:** chiamata anche *cluster*, in questa configurazione i cloni condividono un server di memorizzazione che gestisce i dischi fissi.

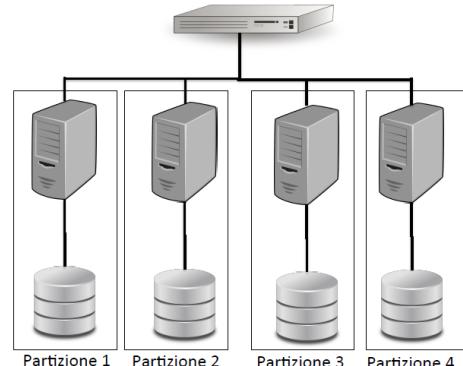


### PARTIZIONAMENTO:

Con il partizionamento. L'hardware è duplicato, per ripartire l'esecuzione di una applicazione tra i nodi, partizionando anche i dati, quindi ogni nodo svolge una specifica funzione. Il partizionamento è trasparente alle applicazioni e ogni richiesta viene inviata e gestita dalla partizione che dispone delle risorse per farlo.

**Graceful degradation** (degrado parziale): in caso di malfunzionamento non tutto il sistema risulta inaccessibile, ma solo alcune funzionalità o dati corrispondenti al partizionamento guasto.

Per risolvere il problema di indisponibilità di alcune funzionalità applicative, spesso si clonano i singoli server che costituiscono la partizione, creando dei **pack**. Si parla allora di *RAPS* (Reliable Array of Partitioned Service).



## Virtualizzazione

La **virtualizzazione** è la tecnologia che permette di astrarre le componenti fisiche di un elaboratore al fine di renderle disponibili in modalità software. A differenza di una componente fisica, con delle componenti virtualizzate chiedere l'esecuzione di un'operazione significa eseguire una chiamata ad una applicazione la quale assegnerà secondo il suo funzionamento questa operazione alla componente fisica opportuna di cui dispone.

## Cloud computing

Il Cloud computing è un paradigma per la realizzazione di architetture applicative, basate sulla virtualizzazione, che possono accedere alle risorse richieste *on demand*, cioè solo quando servono effettivamente, da qualunque dispositivo.

I diversi vantaggi del cloud computing sono: la riduzione dei costi, la scalabilità che consente alta elasticità, un utilizzo migliorato delle risorse in situazioni di carico elevato.

Il cloud computing si basa sul concetto di **utility computing**: l'utente paga in base all'utilizzo che fa delle risorse, senza preoccuparsi di quante siano: il compito di sfruttare le risorse effettivamente disponibili al meglio è lasciato ai gestori delle piattaforme cloud.

Il cloud computing è un sistema con le seguenti caratteristiche:

- **On-demand self-service:** gli utenti possono ottenere risorse come tempo di esecuzione o spazio di memorizzazione in modo autonomo.
- **Accesso alla rete omnipresente:** gli utenti possono accedere attraverso qualunque dispositivo internet.
- **Accesso alle risorse indipendente dalla loro localizzazione.**
- **Elasticity:** le risorse possono essere rapidamente assegnate per rispondere alle richieste degli utenti.
- **Monitoraggio sull'utilizzo:** il costo delle risorse è commisurato della base delle risorse effettivamente utilizzate.

Esistono tre **modelli di servizio** (di erogazione), di base, per un sistema basato su cloud computing:

- **Infrastructure as a Service** (IaaS): Il gestore della piattaforma di cloud computing mette a disposizione degli utenti risorse di tipo fisico, che vengono viste dagli utilizzatori come elementi virtuali per ospitare le proprie applicazioni all'interno di macchine virtuali.
- **Platform as a Service** (PaaS): il provider mette a disposizione degli utenti sia risorse di tipo fisico, che tecnologie a livello di piattaforma, un esempio è un DBMS in Cloud. Vengono fornite delle API per utilizzare il servizio e l'utente si prende carico di implementare le proprie applicazioni per accedervi.
- **Software as a Service** (SaaS): oltre all'infrastruttura fisica e alle tecnologie di piattaforma, all'utilizzatore sono forniti veri e propri applicativi per interfacciarsi con il servizio. L'utente non ha controllo sul servizio, ma può comunque configurare alcuni parametri delle applicazioni. Esempi di SaaS sono Gmail e Dropbox.
- **Everything as a Service** (\*aaS): molti altri componenti a livello di business, applicativo e fisico vengono forniti come servizi. Si parla di Data as a Service (DaaS), Business Process as a Service (BPaaS), e così via.

Esistono anche quattro **modelli di deployment**:

- PRIVATE CLOUD: uso esclusivo di una organizzazione, con alto livello di sicurezza e personalizzazione.
- COMMUNITY CLOUD: l'infrastruttura è fornita ad una comunità di organizzazioni cooperanti, il livello di sicurezza e personalizzazione è ancora abbastanza alto.
- PUBLIC CLOUD: infrastruttura fornita ad uso pubblico, con costi ridotti e grande scalabilità dell'architettura, ma livelli di sicurezza e personalizzazione inferiori.
- HYBRID CLOUD: combinazione di modelli precedenti, gode dei benefici dovuti all'uso di modelli di deployment multipli

## Opzioni di gestione del SI

L'infrastruttura che ospita i componenti applicativi può essere in-house o in outsourcing. Per quanto riguarda l'outsourcing esistono diversi livelli di servizio:

- LIVELLO 0: tutto gestito in-house.
- LIVELLO 1 (servizi condivisi): sistemi gestiti internamente all'organizzazione, se ne occupa una azienda IT che lavora in modo dedicato all'organizzazione stessa.
- LIVELLO 2 (supporto esterno): sistemi ancora interni all'organizzazione ma gestiti da una azienda IT esterna.
- LIVELLO 3 (consorzio): alcune componenti del sistema sono gestite all'esterno, a volte condivise con altre organizzazioni per ridurne i costi.
- LIVELLO 4 (Outsourcing selettivo): si trasferiscono all'esterno alcune risorse selezionate, in generale applicazioni non critiche per il core business aziendale.
- LIVELLO 5 (Outsourcing completo): tutto gestito in outsourcing.

Tipologia	Livello	Caratteristiche
In house	0	Servizi sono gestiti internamente
Servizi condivisi (Insourcing)	1	Un'azienda (unità) fornisce servizi IT a tutte le unità
Supporto esterno	2	I servizi sono gestiti internamente, ma con il supporto di compagnie specializzate e società di consulenza
Consorzio	3	Un gruppo di compagnie con bisogni simili creano un consorzio specializzato per alcuni servizi (e.g., call center, data center)
Outsourcing selettivo	4	Alcuni servizi, solitamente non core, sono gestiti da aziende esterne
Outsourcing completo	5	Tutto l'IT è gestito da una ditta esterna. Se l'infrastruttura appartiene al cliente il servizio è chiamato <i>facility management</i>

Le organizzazioni che forniscono servizi di outsourcing sono denominate **Service Provider** e il loro supporto, dal punto di vista dell'architettura fisica, può essere distinto in:

- **Housing**: concessione di uno spazio fisico ad un utente dove inserire risorse fisiche di proprietà del cliente stesso. Tipicamente i server sono ospitati in Webfarm o Data Center.
- **Hosting**: fornitura di servizi in rete, come per siti web ospitati presso server del provider.

Quando una infrastruttura viene data in outsourcing, è importante avere garanzie riguardo le prestazioni offerte dal Service Provider, si devono quindi definire livelli di servizio, KPI (Key Performance Indicators) e definire nei dettagli i meccanismi per la gestione del servizio.

# Capitolo 11 – Sicurezza dei sistemi informativi

## Proprietà di sicurezza

La sicurezza nei sistemi informativi è l'insieme delle misure organizzative e tecnologiche tese ad assicurare ad un utente autorizzato l'accesso a tutti e soli i servizi e le risorse previsti per quello specifico utente. Ciò avviene tramite la protezione di requisiti definiti secondo le proprietà di sicurezza:

- **Integrità**: il sistema deve impedire l'alterazione diretta o indiretta delle informazioni.
- **Autenticità**: il destinatario delle informazioni deve poter verificare l'identità del mittente e le informazioni devono essere integre, inoltre l'autore dell'informazione non deve poter negare l'autenticità dell'informazione. Per assicurare l'autenticità è necessario avere da presupposto l'**Autenticazione**, cioè che ogni agente sia identificato prima di poter interagire con il sistema.
- **Riservatezza (Secrecy)**: nessun utente deve ottenere accesso dal sistema ad informazioni che non è autorizzato a conoscere. La privacy dei dati personali è comportata dalla Secrecy.
- **Disponibilità**: il sistema deve rendere disponibili le informazioni agli utenti abilitati che hanno diritto ad accedervi.

La sicurezza è un requisito non funzionale di un sistema che deve essere pensata in fase di design nel sistema, non aggiunta a posteriori.

Il sistema deve essere in grado di combattere tentativi di intrusione o abusi provenienti dall'esterno (*protezione attiva*), ma anche si rispondere ad un attacco limitandone i danni (*protezione reattiva*).

## Minacce e attacchi

Le **minacce** per un sistema si distinguono in:

- **FISICHE**: danneggiano impianti e infrastrutture.
- **LOGICHE**: sottraggono o alterano informazioni e risorse.
- **ACCIDENTALI**: errori di configurazione del software o della rete, malfunzionamenti o errori di data entry.

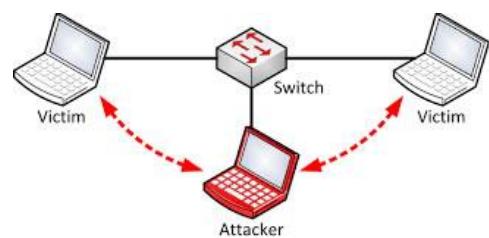
Una **violazione** è la conseguenza di una minaccia, che può comportare situazioni illegittime come l'accesso non autorizzato a risorse o dati del sistema, o come la mancata possibilità di utilizzare le risorse di cui si ha accesso, detta *Denial of Service – DoS*.

Le violazioni avvengono sfruttando le **vulnerabilità** del sistema, che presuppongono l'esistenza di dei punti deboli, o *exploit*, cioè tecniche che facendo leva sulle vulnerabilità cercano di causare un comportamento anomalo del sistema.

Gli **attacchi** di sicurezza sono una qualunque azione che colpisce basi di dati, infrastrutture, reti di comunicazioni e/o dispositivi tramite atti malevoli finalizzati al furto, all'alterazione o alla distruzione di elementi specifici, violando le regole di accesso ai sistemi.

Gli attacchi più frequenti a livello di rete sono:

- **Sniffing**: intercettazione di messaggi scambiati tra due agenti, è un attacco alla riservatezza.
- **Spoofing**: invio di pacchetti con indirizzo di sorgente diverso dal proprio allo scopo di non farsi identificare o di impersonare un altro agente.
- **Hijacking**: anche detto *Man-in-the-middle*, consiste nella deviazione delle comunicazioni verso un agente malevolo che può intercettare e modificare il traffico in modo invisibile ai sistemi compromessi.
- **Flooding**: intasamento della rete con quantità di traffico non desiderato che porta al blocco o all'irraggiungibilità di un servizio (DoS).



Gli attacchi più frequenti a livello applicativo sono:

- **Malware:** software creato allo scopo di creare danni, si dividono in:
  - **Virus:** software che quando eseguito infetta i file e si riproduce-
  - **Worm:** programma eseguibile che si replica in automatico, senza infettare altri programmi.
  - **Trojan horse:** codice nascosto in programmi che conferisce funzionalità malevole.
  - **Ransomware:** limita l'accesso al dispositivo infettato fino al pagamento di un riscatto (ransom).
- **Backdoor:** porta segreta lasciata aperta in un programma che consente di superare in parte le procedure di sicurezza, spesso lasciate intenzionalmente per essere utilizzate in fase di manutenzione. In particolare, una *botola malevola* è una backdoor da cui fuoriesce dell'informazione. Per evitare le backdoor bisogna affidarsi solo a trusted providers.
- **Spyware:** software utilizzati per raccogliere informazioni sul sistema su cui sono installati e trasmetterle a un destinatario interessato.

È importante considerare che gli attacchi possono provenire sia da utenti esterni che da utenti interni al sistema, che utilizzano i propri diritti per accedere a informazioni non di loro competenza. Il 90% degli attacchi avviene da parte di utenti legittimi.

## Crittografia

La crittografia consiste nella codifica dell'informazione mediante una cifratura. I meccanismi di crittografia sono composti da un algoritmo, detto **funzione critografica**, e da una o più **chiavi** segrete, che devono essere difficili da individuare e cambiate spesso.

Esistono due tipi di algoritmi:

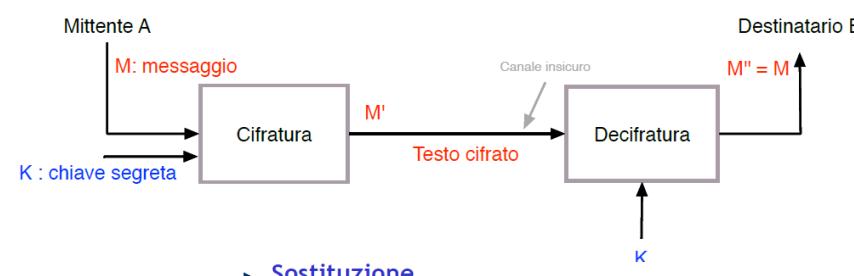
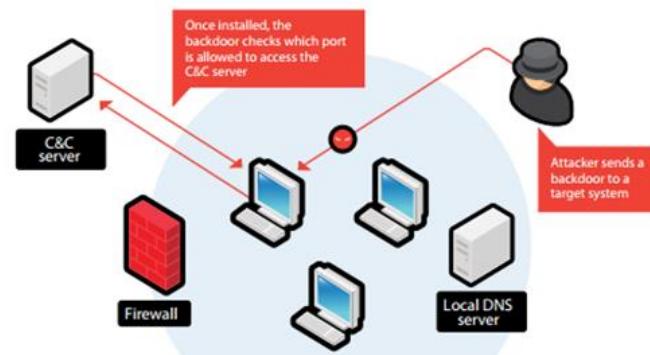
- ALGORITMI A CHIAVE SIMMETRICA (o *a chiave segreta*): esiste una sola chiave condivisa tra chi invia e chi riceve.
- ALGORITMI A CHIAVE ASIMMETRICA (o *a chiave pubblica*): ogni soggetto possiede una coppia di chiavi, una chiave privata e una chiave pubblica.

### CRITTOGRAFIA SIMMETRICA:

Esiste una chiave segreta che è condivisa tra mittente e destinatario ed è utile a decifrare l'informazione scambiata. Gli attacchi di sniffing possono cercare di raccogliere un gran numero di messaggi al fine di individuare la chiave, per questo è utile cambiare spesso la chiave, probabilmente ad ogni sessione. Per lo scambio di chiave i due partecipanti alla sessione devono individuare un canale alternativo e sicuro.

Le due tecniche principali per la crittografia simmetrica sono la sostituzione e la trasposizione (o *permutazione*). Nella tecnica di **sostituzione**, la chiave è utilizzata per sostituire le lettere con altre scelte in base alla chiave; ne è un esempio il cifrario di Cesare in cui ogni lettera è sostituita con quella che si trova  $k$  posizioni più avanti nell'alfabeto, dove  $k$  è la chiave. Nella tecnica di **trasposizione**, la chiave serve per indicare come permutare le lettere contenute nel testo in chiaro durante la cifratura, secondo diverse modalità in base all'algoritmo scelto.

Queste due tecniche sono combinate diverse volte dagli algoritmi di crittografia effettuando una serie di sostituzioni (S-box) e trasposizioni (P-box) alternate. Il processo è effettuabile anche a livello hardware operando sui bit di un messaggio.



► **Sostituzione**  
chiave = 4

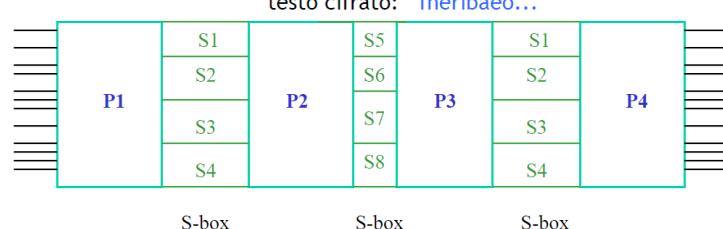
A B C D E F G H I L M N O P Q R S T U V Z  
E F G H I L M N O P Q R S T U V Z A B C D  
ciao -> goes

mapping fisso (sostituzione monoalfabetica)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
Q W E R T Y U I O P A S D F G H J K L Z X C V B N M  
ciao -> eoqg

### ► Trasposizione (permutazione)

P R O V A T I	chiave
4 5 3 7 1 6 2	
t r a s f e r	
i r e u n m i	
l l o n e a b	
testo cifrato: fneribao...	



## CRITTOGRAFIA ASIMMETRICA:

Ogni agente è in possesso di una coppia di chiavi correlate: una chiave pubblica e una chiave privata.

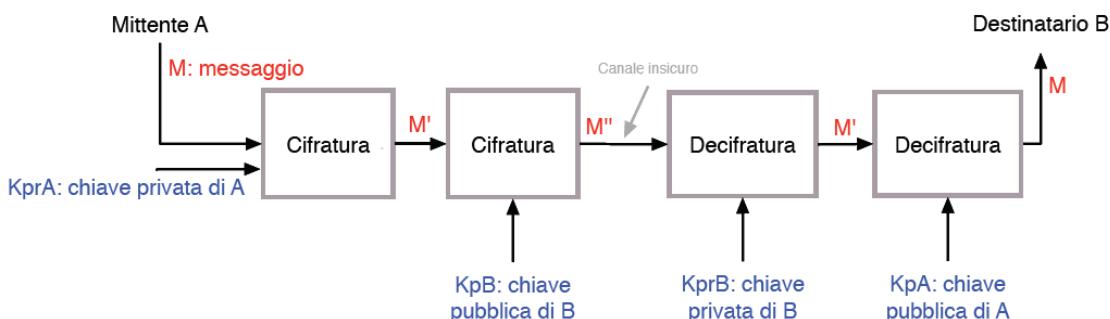
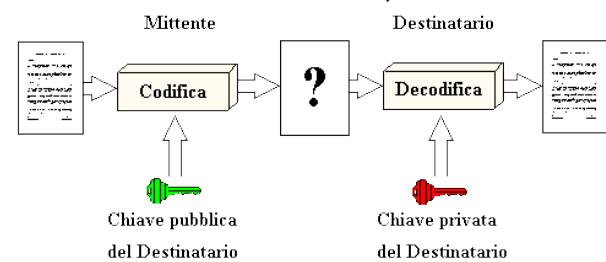
L'algoritmo si basa sul seguente principio: ciò che è cifrato con una chiave è decifrabile solo con l'altra chiave della coppia.

Non deve essere in alcun modo possibile dedurre una delle chiavi di una coppia a partire dall'altra, quindi accertato ciò è possibile che la chiave pubblica sia distribuita, mentre quella privata deve restare segreta. Non è necessario uno scambio di chiavi segrete, come avviene per la crittografia simmetrica, ma solo la distribuzione di quelle pubbliche, che può avvenire attraverso qualunque canale o tramite un servizio di distribuzione di chiavi pubbliche.

Il mittente può cifrare il messaggio con la chiave pubblica del destinatario che, una volta ricevuto il messaggio, è l'unico a poterlo decifrare utilizzando la sua chiave privata, che solo lui possiede. È garantita la riservatezza, essendo che solo il destinatario può leggere il messaggio, ma questo sistema ha il punto debole di essere facilmente soggetto ad attacchi di tipo man-in-the-middle, poiché è facile per terze parti fingersi il mittente all'interno della comunicazione, essendo la chiave di cifratura pubblica.

Se invece il mittente utilizzasse la propria chiave privata per cifrare il messaggio, nessuno potrebbe impersonarlo, ma il messaggio sarebbe facilmente leggibile da terze parti con accesso alla chiave pubblica che serve al destinatario per decifrarlo.

Per garantire sia la riservatezza che l'autenticità, i due approcci sono eseguiti in modo combinato con una doppia fase di cifratura. Il mittente allora cifra il suo messaggio con la propria chiave privata, e poi con la chiave pubblica del destinatario. Quest'ultimo, ricevuto il messaggio, prima lo decifra con la sua chiave privata, poi con quella pubblica del mittente.



Lo svantaggio degli algoritmi a chiave asimmetrica è la loro lentezza computazionale, in quanto comportano calcoli di esponenziali non piccoli, che richiedono grande capacità computazionale. Spesso allora si utilizza la crittografia asimmetrica solo per lo scambio della chiave segreta da utilizzare nella crittografia simmetrica, a cui è delegato il resto del testo scambiato.

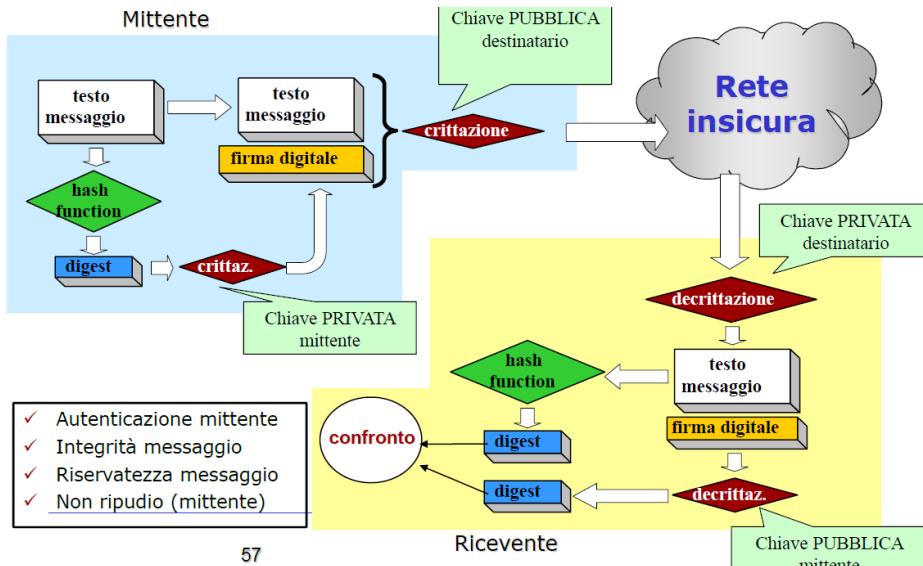
## Integrità e firma digitale con funzioni di hash

Una **funzione di hash** è una funzione che trasforma un messaggio in un output di lunghezza fissa chiamato *impronta digitale*, o *hash*, o *digest* del messaggio originale. La funzione di hash ha le seguenti proprietà:

- **COERENZA:** a messaggi uguali corrisponde lo stesso digest.
- **UNIVOCITÀ:** la probabilità che due messaggi diversi siano associati allo stesso digest deve tendere a zero.
- **NON IRREVERTIBILITÀ:** la funzione non deve essere invertibile.

Le funzioni di hash creano dei digest che possono essere utilizzati come prova dell'integrità del messaggio stesso: se il mittente crea il digest del messaggio e lo invia assieme ad esso, il destinatario può ricostruire il digest dal messaggio e confrontarlo con quello ricevuto, per capire se il messaggio è stato manipolato.

**La firma digitale** è il digest crittografico di un documento, ottenuto utilizzando la chiave privata del mittente del documento. Quindi la firma digitale è un digest, inviato utilizzando crittografia asimmetrica con doppia fase di cifratura, utilizzato per garantire l'integrità e l'autenticità del messaggio. Per via dei lunghi tempi di elaborazione richiesti, la firma digitale è utilizzata solo in alcuni contesti.



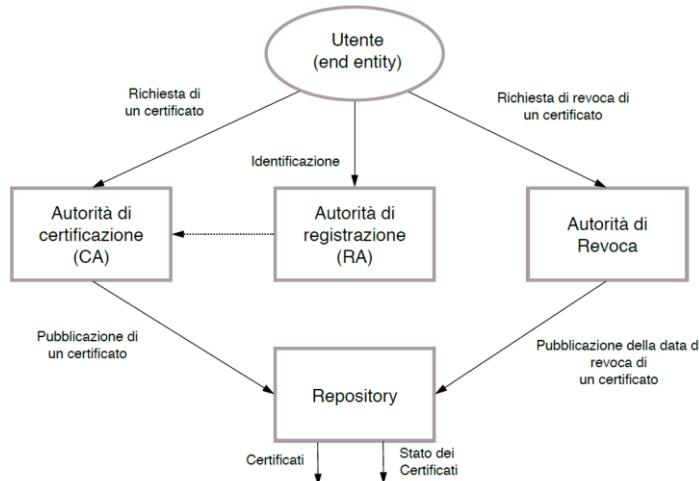
57

### Gestione delle chiavi e certificati digitali

La generazione delle chiavi è un'operazione generalmente effettuata da chi effettuerà le operazioni crittografiche, anche se in casi eccezionali è affidata a un'entità diversa, e la qualità della chiave generata dipende dalla qualità del RNG che viene utilizzato. Le chiavi sono conservate sia con soluzioni software che hardware, come file protetti da password o dispositivi rimovibili che le contengano.

Lo scambio di chiavi è un'operazione critica. Spesso la chiave segreta della crittografia simmetrica viene distribuita OOB, cioè out-of-band, su un canale diverso da quello su cui transitano i dati.

Le chiavi pubbliche vengono distribuite mediante una struttura dati chiamata certificato a chiave pubblica, o **Public Key Certificate** (PKC). Della generazione e gestione delle chiavi si occupano le **Public Key Infrastructure** (PKI), che in particolare si occupano di emissione, revoca e distribuzione dei certificati a chiave pubblica.



### Gestione utenti e controllo accessi

Prima di aprire un canale di sicurezza o accedere a informazioni, è essenziale autenticare in modo certo gli agenti. Grazie all'autenticazione, è possibile effettuare un filtraggio delle modalità di accesso ai vari componenti del sistema, concedendo o no determinate autorizzazioni a un utente o a un ruolo, ricoperto da un gruppo di utenti.

Un agente può essere autenticato utilizzando una delle seguenti caratteristiche:

- SOMETHING YOU KNOW (come una password)
- SOMETHING YOU HAVE (come una carta magnetica)
- SOMETHING YOU ARE (come un'impronta digitale)

Più di queste caratteristiche sono richieste per l'accesso, più robusta è l'autenticazione. Una autenticazione semplice si basa su userid e password, mentre un'autenticazione robusta si basa su meccanismi di tipo One-Time Password (OTP), su sistemi challenge-response, in cui la password non viene trasmessa per verificarne la correttezza ma utilizzata per effettuare un calcolo che ne dimostra indirettamente la conoscenza, o mediante l'uso di token come smart card, o basandosi su caratteristiche biometriche.

Il **controllo degli accessi** è l'insieme delle procedure che verificano se l'agente è autorizzato allo svolgimento delle funzioni elementari (read, write, execute).



## Sistemi di controllo di accesso ai dati

Si parla di **DBMS sicuri** con riferimento alle politiche e ai meccanismi dei DBMS per proteggere i dati da accessi illegittimi. Questi meccanismi devono essere in grado di trattare i dati a diverse granularità, filtrare diversi modi di accesso e tipologie di controllo, gestire l'autorizzazione dinamica (grant/revoke), e di garantire l'assenza di backdoor. Le regole di accesso nei DBMS sono composte da:

- **AUTORIZZATORI**: il proprietario (owner) di una risorsa gestisce gli accessi in modo selettivo, diventando l'autorizzatore per quella risorsa.
- **SOGGETTI**: gli utenti, definiti dai loro *user profile*.
- **OGGETTI**: le risorse protette, la scelta della loro granularità è essenziale.
- **DIRITTI**: i diversi modi di accesso da consentire, per esempio *read, write, create, update, delete*, e così via.

Le **politiche di sicurezza** dettano la definizione delle regole in un sistema, che può essere **chiuso**, se nessuno detiene dei permessi e questi vanno concessi, o **aperto**, se tutti gli utenti hanno tutti i permessi e questi vengono tolti a seconda del tipo di utente.

Le regole di accesso definite dagli autorizzatori sono basate sul modello:

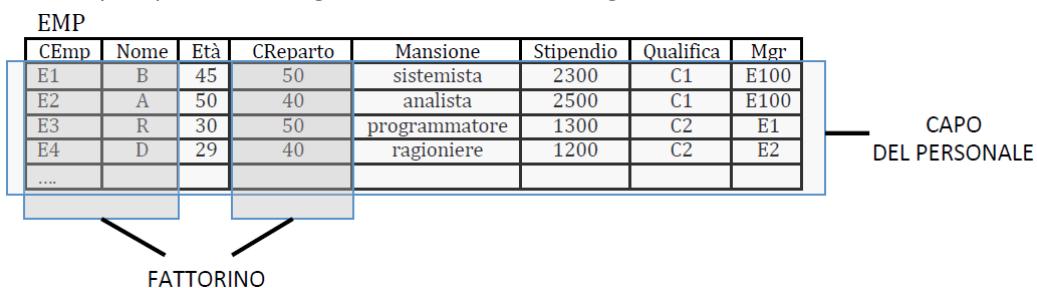
*(subject, object, right, constraint)*

Dove i constraint esprimono i vincoli di accesso dipendenti dal contenuto, cioè delle view dei dati che sono le uniche concesse all'utente.

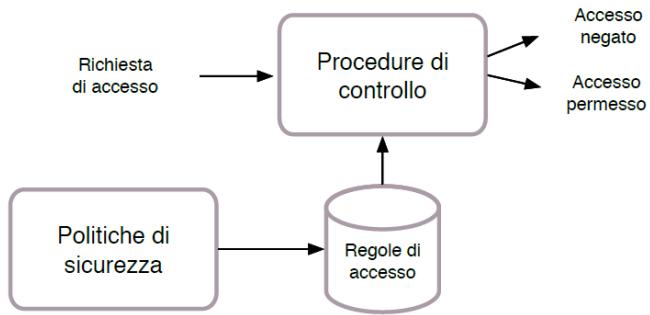
Le politiche di accesso ai dati si dividono in politiche discrezionali (Discretionary Access Control – DAC) e in politiche manderarie (Mandatory Access Control – MAC).

### POLITICHE DISCREZIONALI:

Gli utenti amministrano i dati che possiedono, e possono autorizzare gli altri utenti all'accesso. Le regole di accesso possono essere specificate nel profilo utente, oppure associate alla descrizione dell'oggetto protetto. I comandi per la specifica delle autorizzazioni sono inseriti o nel Query language del DBMS, per esempio con la clausola *Grant*, o direttamente nel SO. Un utente che non dispone di un permesso può vedersi negata la possibilità di eseguire una interrogazione, oppure con il meccanismo di *query modification* la sua interrogazione produrrà solo i risultati che è autorizzata a produrre per quell'utente, ignorando le richieste illegali.



In ambiente DAC si può utilizzare il modello a sei componenti: *(gtor, gtee, obj, right, constraint, prop\_rule)* dove *prop\_rule* è la regola di propagazione dei privilegi, che specifica se, e a quali condizioni, un permesso può essere propagato da chi lo detiene, ma non è owner, ad altri utenti.



## POLITICHE MANDATORIE:

Adatte a database sensibili, gli oggetti sono classificati assegnando loro dei livelli di **sensitività** e i soggetti sono classificati con dei livelli di **clearance**. Sono inoltre definite delle **security class** (SC) che categorizzano in modo gerarchico le aree di appartenenza dei dati. Le regole di base delle politiche MAC si devono assicurare che i soggetti possano accedere solo ai dati per cui dispongono della clearance appropriata, in particolare le due regole base sono:

- **No-read-up**: un soggetto può accedere in lettura solo a oggetti che hanno livello di sensitività minore o uguale al livello di clearance del soggetto.
- **No-write-down**: un soggetto è autorizzato ad accedere in scrittura ad un oggetto solo se il livello di clearance del soggetto è minore o uguale al livello di sensitività dell'oggetto.

Tipici livelli di classificazione sono: Unclassified (U), Confidential (C), Secret (S), Top Secret (TS).

Name	CName	Dept	CDDept	Salary	CSalary	CTuple
BoB	S	Dept1	S	10 K	S	S
Ann	S	Dept2	S	20 K	TS	TS
Sam	TS	Dept2	TS	20 K	TS	TS

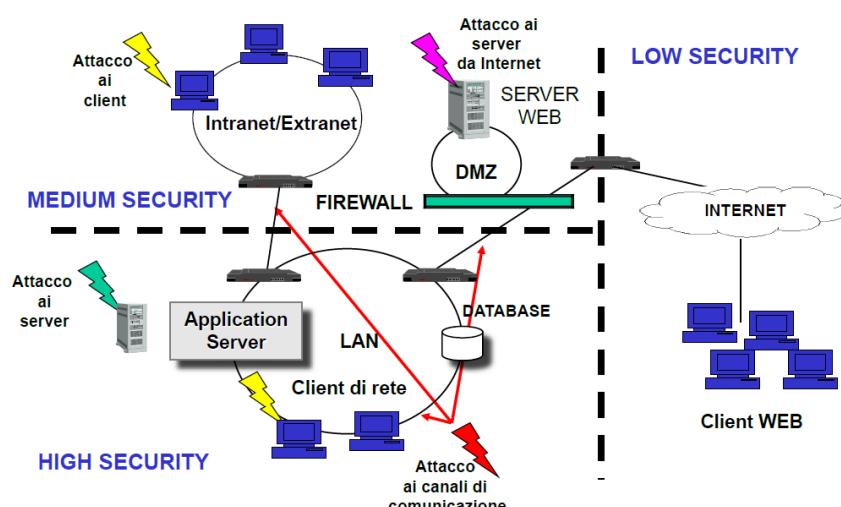
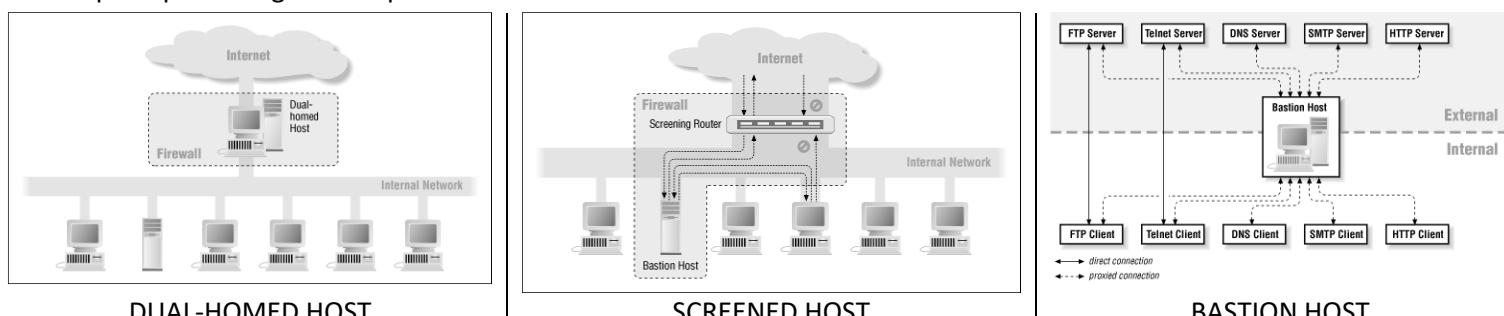
I DBMS con politiche MAC sono implementati utilizzando una *trusted computing base* (TCB), cioè un insieme di componenti hardware e software fidati per la memorizzazione e gestione delle etichette di classificazione. Per la realizzazione di questi tipi di database, si utilizzano tabelle relazionali multilivello instanziate in modo separato a seconda della loro classificazione di sicurezza, e la TCB esegue la mediazione di tutti gli accessi in modo "trusted", cioè mediante componenti che non possono essere manomessi.

## Firewall

Tra le configurazioni a livello di architettura fisica utili alla protezione da attacchi c'è il **firewall**, che è genericamente un insieme di componenti e servizi finalizzati a controllare e limitare il traffico tra una rete da proteggere e le reti esterne. Le tre leggi principali che deve seguire un firewall sono:

- Il firewall deve essere l'unico punto di contatto tra la rete interna e l'esterno
- Tutti i pacchetti, tranne quelli autorizzati, devono essere bloccati
- Il firewall deve essere un sistema sicuro a sua volta

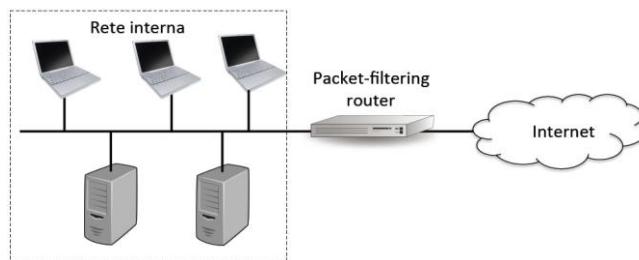
L'efficacia di un firewall dipende dalla correttezza delle regole specificate e dal tipo di configurazione messa in atto. Le principali configurazioni possibili sono:



I componenti base che permettono il monitoraggio del traffico sono gli *screening router* e gli *application gateway*.

#### SCREENING ROUTER:

Dispositivi che bloccano o instradano i pacchetti in transito tra rete esterna e interna sulla base della loro intestazione (packet filtering) o del loro contenuto (packet inspection). I firewall vanno configurati in modo da disabilitare funzioni di configurazione remota, per la sicurezza del firewall stesso, e le funzioni di dynamic routing, per il suo corretto funzionamento.

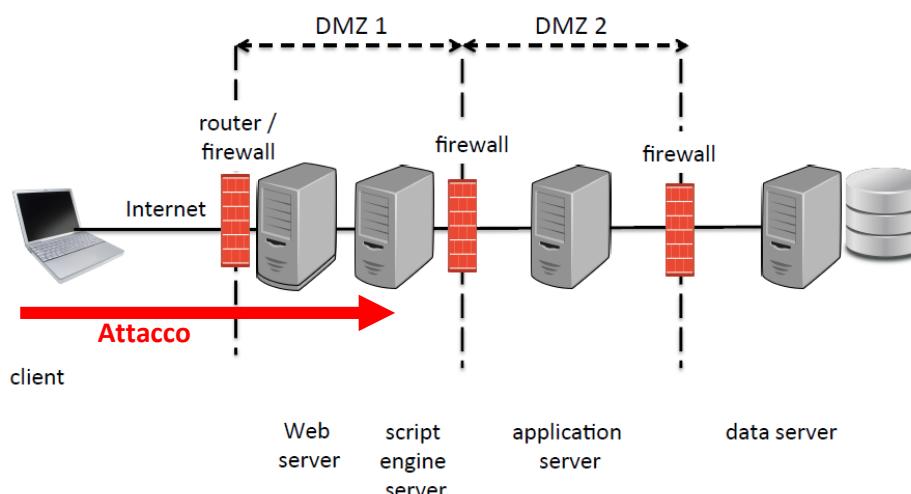
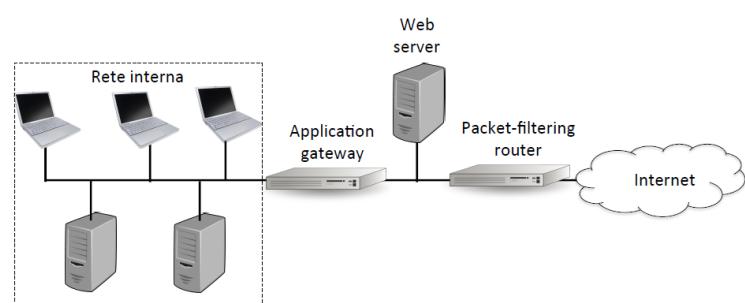


I firewall basati su **packet filtering** valutano i pacchetti in base alla loro intestazione, utilizzando un elenco di regole che permettono o negano il passaggio di un pacchetto in base alle informazioni contenute nel suo header. Molti produttori di router applicano le regole di filtraggio solo sui pacchetti in uscita dal router, non su quelli in ingresso, il che non impedisce attacchi di spoofing, anche comportanti l'impersonificazione di un agente della rete interna protetta.

I firewall di tipo **packet inspection** valutano i pacchetti sulla base del loro payload, offrendo una grande flessibilità nella definizione delle regole di filtraggio (smart rule) che quando operano a livello di sessione nel protocollo TCP sono paragonabili a quelle degli application gateway, ma senza funzionalità proxy.

#### APPLICATION GATEWAY:

I firewall di questo tipo sono realizzati mediante un host apposito configurato con software specifico che eroga, oltre a servizi di packet filtering e inspection, anche servizi di **application proxy**. Questi ultimi consistono in processi che si pongono tra client e server comunicanti, simulando le due comunicazioni e controllando il traffico. In particolare, il flusso proveniente dalle parti viene intercettato, simulato a livello applicativo, ricostruito sulla base della conoscenza del protocollo utilizzato e instradato verso il comunicante. Ricostruendo il traffico, questa tecnologia minimizza la probabilità di successo degli attacchi basati su vulnerabilità, ma la ricostruzione dei pacchetti richiede un tempo di analisi che comporta lentezza nella comunicazione. Inoltre, poiché lavora a livello di applicazione, un application gateway è in grado di controllare un numero ridotto di protocolli, per cui è necessaria la configurazione di un nuovo proxy per ogni nuovo protocollo da gestire.



Per aumentare il livello di sicurezza, si possono aumentare i livelli delle architetture, così che un attacco debba passare attraverso più livelli. In particolare, si può predisporre una zona che si trova tra la rete interna e quella esterna, chiamata **Demilitarized Zone (DMZ)**. La rete esterna può accedere solo alle risorse esposte nella DMZ, mentre il resto delle risorse è nascosto dietro altri firewall.

## Intrusion Detection Systems – IDS

Con **Intrusion Detection** (ID) si intende il processo di monitoraggio degli eventi di un sistema con il fine di individuare tracce evidenti di intrusioni, cioè quegli insiemi di azioni volte a compromettere una o più proprietà di sicurezza del sistema. L'**Intrusion prevention** è l'estensione dell'ID con aspetti di controllo dell'accesso volti a proteggere il sistema da tentativi di sfruttamento malevolo. Un IDS è un sistema che automatizza l'ID, identificando intrusioni di utenti autorizzati (*insider threaters*) e non autorizzati (*crackers*).

Una volta rilevata un'intrusione, un IDS può richiedere la chiusura delle connessioni, impedire altre connessioni con l'indirizzo IP sospetto, e notificare all'amministratore la presenza di un possibile attacco.

Gli IDS si basano sui seguenti presupposti: le attività di sistema devono essere osservabili, ed è possibile distinguere tramite le loro diverse evidenze le attività normali da quelle intrusive; ciò è realizzato mediante algoritmi basati su Features e Modelli, mentre, dal punto di vista architetturale, mediante vari componenti tra cui processori per i dati audit, una base di conoscenza, un decision engine, e generatori di allarmi.

I componenti funzionali di un IDS sono:

- Un insieme di sensori (Information Source), che raccolgono gli elementi da cui è possibile rilevare le tracce di un'intrusione.
- Algoritmi di analisi, che implementano l'ID sulla base degli eventi raccolti dai sensori.
- Componenti Alerting e Response, cioè componenti che quando un attacco è in atto intraprendono azioni come, ad esempio, il report degli allarmi e azioni di difesa come la chiusura di porte.

Gli IDS si possono classificare in base al tipo di sensori:

- **NETWORK-BASED IDS**: rileva gli attacchi analizzando il traffico di rete, con sensori distribuiti nella rete stessa. Sono necessari pochi sensori per monitorare anche grandi reti, che hanno basso impatto sulle prestazioni della rete, ma in caso di congestione questi sistemi perdono di efficienza. Inoltre, sono aggirabili tramite l'utilizzo di pacchetti anomali da parte di alcuni particolari attacchi.
- **HOST-BASED IDS**: software che cerca evidenze di intrusioni analizzando informazione raccolte dal singolo host su cui è installato. Rilevano attacchi che un network-based IDS potrebbe non rilevare, ma sono poco scalabili e possono essere disabilitati da attacchi di tipo Dos.

Si possono classificare in base ai tipi di analisi che effettuano:

- **MISUSE DETECTION**: analisi volta a individuare comportamenti già conosciuti e noti come attacchi, tramite il confronto dei pattern di comportamento, detti *signature*, trovati con quelli noti degli attacchi. È una modalità molto affidabile e rilevano tempestivamente anomalie, con un numero minimo di falsi allarmi, ma sono rilevabili solo attacchi conosciuti e le signature vanno aggiornate di frequente.
- **ANOMALY DETECTION**: si individuano deviazioni significative dal comportamento normale, sull'ipotesi che un sistema sotto attacco si comporti diversamente da come si comporta normalmente. Costruiscono infatti dei profili del comportamento dell'utente, sulla base di dati storici, da utilizzare per i confronti. Purtroppo, generano molti falsi allarmi e richiedono un vasto training, benché possano rilevare attacchi sconosciuti.

È anche possibile una classificazione in base ai tipi di Response:

- **ACTIVE RESPONSE**: l'IDS blocca l'attacco in corso tramite azioni di reset delle connessioni o riconfigurazione di router e firewall, oppure raccoglie informazioni aumentando il livello di sensitività delle sorgenti di informazione.
- **PASSIVE RESPONSE**: l'IDS si limita a inviare una notifica dell'attacco in corso all'amministratore.

