

# Riassunto Sistemi Informativi

per il Settore dell'Informazione

Anna Mettifogo

7 Ottobre 2024

Riassunto tratto dal libro "Fondamenti di Sistemi informativi per il Settore dell'Informazione - 2° edizione" di Cappiello, Fugini, Grefen, Pernici, Plebani, Vitali

# Indice

<b>I Definizioni, tecnologie e progettazione</b>	<b>3</b>
<b>1 Definizioni, tecnologie e progettazione</b>	<b>4</b>
1.1 Dati e Informazioni . . . . .	4
1.2 Le Organizzazioni . . . . .	5
1.2.1 Risorse . . . . .	5
1.2.2 Processi . . . . .	5
1.3 Definizione e ruolo di sistema informativo . . . . .	7
1.4 Classificazione dei sistemi informativi . . . . .	8
1.4.1 Sistemi OLTP e OLAP . . . . .	8
1.5 Organizzazione, IT e componenti di un SI . . . . .	9
 <b>II Enterprise Modelling</b>	 <b>10</b>
<b>2 Enterprise Architecture</b>	<b>10</b>
2.1 Framework per Enterprise Architecture . . . . .	10
<b>3 Modellare una EA con ArchiMate</b>	<b>12</b>
3.1 Introduzione ad ArchiMate . . . . .	12
3.2 Elementi principali del modello . . . . .	12
3.2.1 Componenti . . . . .	12
3.2.2 Relazioni . . . . .	12
<b>4 Modellazione dei processi</b>	<b>14</b>
4.1 Modellazione dei processi in BPMN . . . . .	14
4.1.1 Definizioni . . . . .	14
4.1.2 Business Process Modeling Notation . . . . .	14
4.2 Identificazione dei processi . . . . .	15
4.2.1 Check-list settoriali . . . . .	15
4.2.2 Catena del valore di Porter . . . . .	16
4.2.3 Approccio Analitico . . . . .	17
 <b>III Aspetti Tecnologici</b>	 <b>18</b>
<b>5 Introduzione agli elementi tecnologici</b>	<b>18</b>
5.1 Introduzione . . . . .	18
5.1.1 La tecnologia a livello applicativo . . . . .	18
5.1.2 La tecnologia a livello di piattaforma . . . . .	19
5.1.3 La tecnologia a livello di architettura fisica . . . . .	20
5.2 Componenti applicativi e tecnologici in ArchiMate . . . . .	20
<b>6 Livello applicativo: ERP e CRM</b>	<b>21</b>
6.1 Evoluzione dei Componenti Applicativi . . . . .	21

6.2	Componenti Funzionali . . . . .	22
6.3	ERP . . . . .	22
6.3.1	Storia degli ERP . . . . .	22
6.3.2	Proprietà degli ERP . . . . .	22
6.3.3	Funzionalità di un ERP . . . . .	23
6.3.4	Aspetti architetturali . . . . .	23
6.3.5	Implementazione di un ERP . . . . .	24
6.3.6	Benefici degli ERP . . . . .	24
6.4	CRM . . . . .	24
6.4.1	CRM operativo . . . . .	25
6.4.2	CRM analitico . . . . .	26
6.4.3	CRM collaborativo . . . . .	26
<b>7</b>	<b>Livello applicativo: Analisi dei dati</b>	<b>27</b>
7.1	Data Warehousing . . . . .	27
7.1.1	Caratteristiche dei dati . . . . .	27
7.1.2	Data warehouse . . . . .	28
7.1.3	Architettura del data warehouse . . . . .	29
7.1.4	Modello concettuale e Dimensional Fact Model (DFM) . . . . .	30
7.1.5	Modelli logici del data warehouse . . . . .	31
7.1.6	Operazioni sul data warehouse . . . . .	34
7.1.7	Ciclo di vita del data warehouse . . . . .	35
7.2	Data Mining . . . . .	35
7.2.1	Caratterizzazione e Discriminazione . . . . .	35
7.2.2	Regole Associative . . . . .	36
7.2.3	Classificazione . . . . .	36
7.2.4	Clustering . . . . .	36
7.3	Process mining . . . . .	37
7.3.1	Log di Eventi . . . . .	37
7.3.2	Process Discovery . . . . .	37
7.3.3	Controllo di conformità . . . . .	37
7.3.4	Miglioramento dei processi . . . . .	38
<b>8</b>	<b>Tecnologie a livello di piattaforma</b>	<b>39</b>
8.1	Architetture di integrazione . . . . .	39
8.2	Livelli applicativi e tecniche di integrazione . . . . .	39
8.3	Modelli di integrazione delle applicazioni . . . . .	39
8.3.1	Architettura di integrazione punto-a-punto . . . . .	40
8.3.2	Architettura di integrazione hub-and-spoke . . . . .	40
8.3.3	Integrazione con Business Process Management System . . . . .	40
8.3.4	Modelli a servizi . . . . .	40
8.3.5	Composizione e orchestrazione di servizi . . . . .	40
<b>9</b>	<b>Tecnologie a livello di architettura fisica</b>	<b>41</b>
9.1	Descrizione . . . . .	41
9.1.1	Evoluzione delle architetture fisiche . . . . .	41

9.2	Applicazioni distribuite . . . . .	41
9.3	Scalabilità . . . . .	42
9.3.1	Server Farm . . . . .	42
9.4	Virtualizzazione . . . . .	43
9.5	Cloud Computing . . . . .	44
9.5.1	Caratteristiche principali . . . . .	44
9.5.2	Modelli di servizio . . . . .	44
9.5.3	Modelli di deployment . . . . .	45
<b>10</b>	<b>Sicurezza dei sistemi informativi</b>	<b>46</b>
10.1	Proprietà di sicurezza . . . . .	46
10.2	Minacce, violazioni, vulnerabilità, attacchi . . . . .	46
10.3	Crittografia per la protezione dei dati . . . . .	48
10.3.1	Crittografia simmetrica . . . . .	48
10.3.2	Crittografia asimmetrica . . . . .	48
10.3.3	Integrità e funzione di hash . . . . .	48
10.3.4	Firma digitale . . . . .	49
10.3.5	Aspetti di gestione delle chiavi e certificati digitali . . . . .	49
10.4	Gestione degli utenti e controllo degli accessi . . . . .	49
10.4.1	Autenticazione e autorizzazione . . . . .	49
10.4.2	Autenticazione a due fasi . . . . .	50
10.4.3	Sistemi di controllo di accesso ai dati . . . . .	50
10.5	Zero Trust . . . . .	51
10.5.1	Il contesto . . . . .	52
10.5.2	L'approccio . . . . .	52
10.6	Meccanismi di sicurezza infrastrutturali . . . . .	53
10.6.1	Firewall . . . . .	53
10.6.2	Intrusion Detection Systems - IDS . . . . .	54
10.7	Evoluzione delle Minacce e delle Tecniche di Sicurezza . . . . .	54
10.7.1	Evoluzione delle Minacce . . . . .	55
10.7.2	Evoluzione della Crittografia . . . . .	55
<b>IV</b>	<b>La Progettazione</b>	<b>56</b>
<b>11</b>	<b>La progettazione di un SI</b>	<b>56</b>
11.1	Il processo di gestione del sistema informativo . . . . .	56
11.2	Pianificazione . . . . .	56
11.2.1	Pianificazione strategica . . . . .	57
11.2.2	Studio di fattibilità . . . . .	57
11.3	Ciclo di vita di sviluppo del SI . . . . .	58

# Capitolo 1

## 1 Definizioni, tecnologie e progettazione

I sistemi informativi possono definirsi come strumenti supportati dall'*Information Technology (IT)* e composti da applicazioni software, sistemi di gestione dati, interfacce utente (principalmente di tipo Web) e reti di comunicazione atti a supportare le organizzazioni nella gestione dell'informazione.

Inoltre, è strettamente correlato alle regole di business, alla struttura organizzativa e agli attori dell'ambiente aziendale e come tale va progettato e gestito, rendendo allineate le tecnologie e l'organizzazione.

La progettazione e realizzazione di un sistema informativo è quindi una materia complessa che da un lato deve affrontare le sfide future che vedono un'esplosione di dati a disposizione (*Big Data*) così come una enorme capacità di elaborazione accessibile in modo veloce e a basso costo (basata su *Cloud Computing*) e dall'altro lato deve permettere l'integrazione tra sistemi eterogenei.

### 1.1 Dati e Informazioni

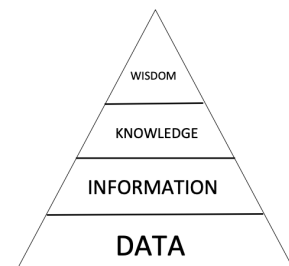
Prendiamo in riferimento la cosiddetta *Piramide della Conoscenza*, detta anche *Piramide DIKW (Data Information Knowledge Wisdom)*.

Il **dato** rappresenta la base di tutto.

Con il termine "dato" intendiamo un fatto, una misura, quindi un elemento che modella o descrive una porzione della realtà che si vuole rappresentare.

Associato ad un dato vi è sempre il suo tipo che specifica il dominio di valori che tale dato può assumere.

Oltre al tipo, un dato può essere caratterizzato dalla sua unità di misura.



Al di sopra del dato si posiziona l'**informazione**. È costruita a partire dai dati e, in particolare, l'informazione può essere definita come l'interpretazione di un singolo o un insieme di dati.

Affinché un dato sia utile deve essere combinato con altri dati al fine di definire un contesto che permetta di caratterizzare meglio la realtà che si sta rappresentando.

L'informazione si può quindi definire come l'output di interrogazioni rivolte ad un insieme di dati.

Sopra l'informazione si trova la **conoscenza** (*knowledge*) che è ottenibile integrando l'informazione con l'esperienza.

Infine, vi è la **saggezza** che rappresenta un'estensione della classica piramide della conoscenza e che può essere definita come l'esperienza applicata alla conoscenza per guidare un soggetto ad intraprendere l'azione più adatta in un determinato momento.

L'utilizzo della piramide non è casuale: offre una stratificazione a livelli, l'area ricoperta da ogni livello diminuisce con l'aumento di livello → questo vuole rappresentare il livello di sintesi. Al crescere del livello diminuisce la capacità di gestione dei contenuti della piramide della conoscenza con strumenti automatici. Mentre i dati vengono gestiti tramite sistemi di gestione di basi di dati (alta automatizzazione), le informazioni possono essere descritte e astratte per supportare le decisioni con strumenti automatici e semiautomatici, mentre la conoscenza presenta aspetti che si prestano solo parzialmente ad un supporto automatizzato.

## 1.2 Le Organizzazioni

Nell'ambito dei sistemi informativi è sufficiente focalizzarsi sui concetti di *risorse* e *processi*.

### 1.2.1 Risorse

**Risorsa** è tutto ciò con cui un'organizzazione opera, sia materiale che immateriale, per perseguire i suoi obiettivi. Si classificano in due tipi:

- Risorse Esterne
  - Ambiente sociale ed economico
  - Mercato
  - Clienti
- Risorse Interne
  - Risorse di scambio: prodotti (beni o servizi)
  - Risorse di struttura: strumenti finanziari, persone (risorse umane), infrastrutture
  - Risorse di gestione: norme, organigrammi, deleghe, piani, informazioni

Ciascuna risorsa ha un proprio ciclo di vita che generalmente comprende le fasi di Pianificazione, Acquisizione, Gestione e Manutenzione.

L'**informazione** viene usata per la comunicazione, per il supporto ai processi e per il supporto alle decisioni (*decision making*), per il quale si ha bisogno di informazioni sulle alternative strategiche o di controllo possibili. Per alcune aziende, l'informazione può essere un prodotto.

**Caratteristiche dell'informazione:** intangibile, non deperibile (non viene distrutta con l'uso), auto-rigenerante (il suo utilizzo porta alla generazione di nuova informazione).

In una linea di produzione, occorre memorizzare informazioni sui lavori effettuati, identificare problemi o migliorare l'esecuzione del processo.

### 1.2.2 Processi

**Processo** è l'insieme di attività che l'organizzazione nel suo complesso svolge per gestire il ciclo di vita di una risorsa, o di un gruppo omogeneo di risorse, per raggiungere un

risultato definito e misurabile (prodotto/servizio).

### Classificazione dei processi con la Piramide di Anthony

Si distinguono tre livelli, legati alla struttura dell'organizzazione:

- *livello operativo*, detto anche *operazionale* → in cui si considerano le attività di tipo operativo dell'azienda
- *livello di programmazione e controllo*, detto anche di *controllo direzionale* → in cui si considerano le attività tattiche quali la programmazione delle risorse disponibili e il controllo sul conseguimento degli obiettivi programmati
- *livello di pianificazione strategica* → dove si collocano le attività legate alla scelta degli obiettivi aziendali e alla definizione delle politiche aziendali.



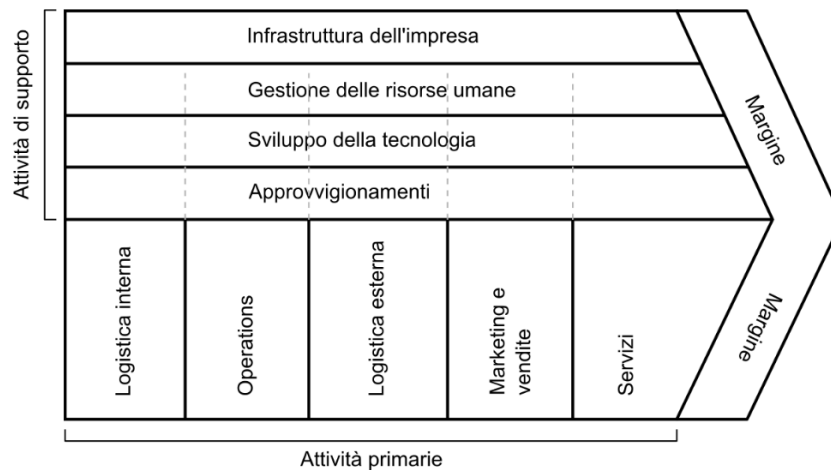
### Classificazione dei processi secondo il Modello di Porter

Si basa sulla *catena del valore* (*value chain*) proposto da Porter, che considera esclusivamente i processi di livello operativo.

Le funzionalità sono suddivise in attività primarie, che sono orientate direttamente agli obiettivi principali dell'organizzazione e normalmente caratterizzano le aziende nei loro settori di riferimento e le attività secondarie o di supporto, che assicurano il corretto funzionamento dei processi primari.

Ogni attività consuma risorse e produce risorse e il corretto svolgimento di tali attività ha impatto positivo sul profitto.

Le attività primarie sono mostrate come colonne verticali nella parte inferiore. I processi secondari con le relative funzioni sono mostrati come strati orizzontali nella parte superiore



### 1.3 Definizione e ruolo di sistema informativo

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

Le basi di dati permettono di memorizzare i dati in grado di rappresentare gli eventi di interesse che accadono all'interno di una organizzazione.

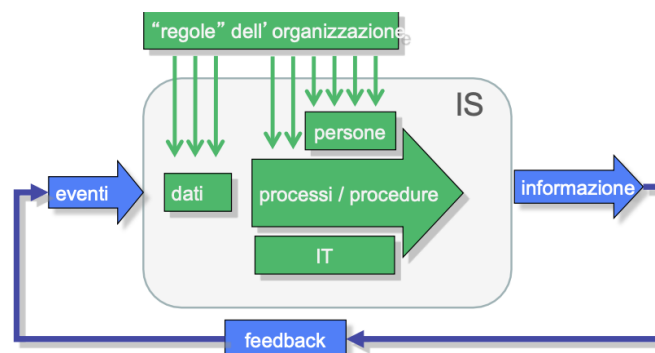
I metodi e gli strumenti di *Business Process Management (BPM)* offrono invece un insieme di tecniche per ottenere sinergia tra il processo organizzativo e la tecnologia a supporto del processo.

La progettazione dei sistemi informativi si pone l'obiettivo di capire e modellare, partendo dai bisogni organizzativi, come i dati e i processi possano essere supportati dalla tecnologia considerando che un'organizzazione è costituita da processi e ruoli differenti ognuno dei quali può richiedere dati e informazioni a diverso livello di dettaglio.

→ la gestione dei dati può prevedere l'uso di strumenti di analisi, quali il *data warehousing* e *data mining*, che si prestano a supportare più i livelli tattici e strategici che non i livelli operativi.

Il *sistema informatico* è un componente del sistema informativo, ovvero quello che è stato identificato come IT, che elabora, archivia e gestisce lo scambio di informazioni.

La sua progettazione e sviluppo include quindi le progettazioni di vari aspetti: dati, processi, interazione utente.





## 1.4 Classificazione dei sistemi informativi

Esistono diverse tipologie di sistemi informativi che si focalizzano su qualche (tipo di) processo, o che sono in grado di gestire determinate risorse.

La piramide di Anthony permette di distinguere in sistemi informativi *operazionali* o *decisionali* (*Management Information Systems*):

- **Sistemi operazionali** → svolgono operazioni di base, quotidiane. Le parti fondamentali di un sistema informativo operativo sono la base di dati operativa e le funzioni operative.
- **Sistemi decisionali o informativi** → servono a supporto delle attività decisionali e strategiche aziendali, in risposta all'esigenza di sfruttare il patrimonio dei dati per identificare le informazioni utili al processo decisionale.

Esistono funzionalità sofisticate di estrazione e analisi dati a fini strategici

- *data warehouse* → base di dati informazionale, che raccoglie in un unico sistema i dati di interesse per l'azienda
- *data warehouse* → base di dati informazionale, che raccoglie in un unico sistema i dati di interesse per l'azienda
- *data warehousing* → insieme di attività che realizza la definizione, costruzione e mantenimento struttura delle informazioni nel data warehouse
- *Decision Support System (DSS)* → strumenti di supporto all'estrazione di informazioni per il processo decisionale quali i browser *OLAP (OnLine Analytical Processing)*

Queste funzionalità vengono generalmente chiamate *Business Intelligence*.

Anche i dati sono classificabili a livello operativo/controllo/strategico.

### 1.4.1 Sistemi OLTP e OLAP

I **sistemi OLTP** (*On-line Transaction Processing*) trattano operazioni caratterizzate da un gran numero di transazioni brevi e on-line (= una transazione i cui effetti sono immediatamente riscontrabili sul sistema).

Sono focalizzati sulla gestione molto rapida delle query, sul mantenere l'integrità dei dati in ambienti multi-accesso e sul garantire l'efficienza ed efficacia del sistema; i dati sono memorizzati dettagliati e attuali.

Sono idonei alla gestione di processi a livello operativo e di controllo.

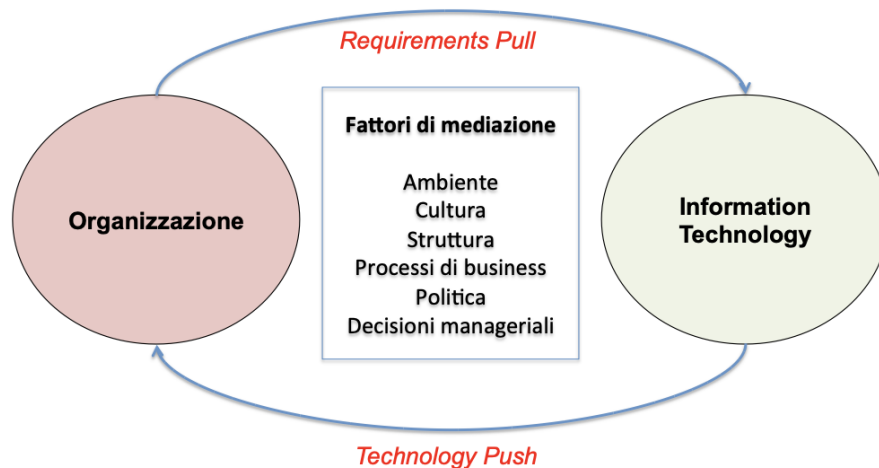
I **sistemi OLAP** (*On-line Analytical Processing*) invece trattano dati storici (o d'archivio). È caratterizzato da poche transazioni e interrogazioni (query) complesse che richiedono aggregazione di dati. Hanno come misura di efficienza e di efficacia *il tempo di risposta*. Permettono di realizzare operazioni complesse e casuali, in cui ogni singola operazione può coinvolgere molti dati aggregati, storici, e in genere anche non attuali, e per cui le proprietà ACID non sono rilevanti perché le operazioni sono di sola lettura.

	<b>OLTP</b>	<b>OLAP</b>
Utente	Impiegato (molti)	Dirigente (pochi)
Funzione	Operazioni giornaliere	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

## 1.5 Organizzazione, IT e componenti di un SI

Tra l'organizzazione e l'IT sussiste una relazione bidirezionale mediata da vari fattori. Le decisioni prese a livello strategico o di controllo possono avere un impatto sulle scelte tecnologiche (*requirements pull*), mentre i cambiamenti degli obiettivi aziendali ecc, possono essere dovuti a nuove tecnologie che offrono nuove opportunità nel settore di attività dell'azienda o per un suo ampliamento (*technology push*).

Un aspetto fondamentale per mantenere coerenti i vari componenti del sistema è la definizione di una architettura complessiva (*Enterprise Architecture - EA*) alla base del sistema (o dei sistemi) informativi aziendali.



# Capitolo 2

## 2 Enterprise Architecture

### 2.1 Framework per Enterprise Architecture

Per definire la struttura del sistema informativo si ricorre al concetto di Architettura → è data dall'insieme delle componenti, delle relazioni fra componenti e dei principi che definiscono la loro progettazione ed evoluzione nel tempo. Fornisce 3 diverse viste:

- *vista statica*: relativa alla descrizione dei componenti.
- *vista dinamica*: relativa alla descrizione delle relazioni tra componenti.
- *vista delle regole*: relativa alle procedure che supportano la progettazione ed evoluzione del sistema nel tempo.

Fornisce anche diverse prospettive agli *stakeholder* del sistema (aka gli interessati). Deve permettere di soddisfare le diverse esigenze degli stakeholder dando la possibilità di analizzare il sistema da diversi pov e con diversi livelli di granularità.

Arriviamo alla definizione di *Enterprise Architecture (EA)* → il mezzo per analizzare e descrivere un'organizzazione nel suo stato attuale e futuro partendo da una prospettiva integrata di strategia, business e tecnologia.

Fornisce una panoramica dei processi, dei sistemi, della tecnologia e delle capacità di una organizzazione. Può essere vista come una matrice (*blueprint*) di alto livello del sistema, utile per capirne la struttura interna, come supporto alla sua progettazione, riprogettazione, configurazione e manutenzione.

A fine anni 90, *John Zachman* definì fondamentale l'uso di un'architettura per descrivere il sistema informativo aziendale, visto che la tecnologia iniziava a consentire la distribuzione di applicazioni e dati su macchine diverse.

Il *framework di Zachman* è organizzato in righe e colonne. Le colonne definiscono gli *Aspetti* da analizzare; vengono definite le seguenti sei colonne:

- *Dati (cosa)*: i dati di cui l'organizzazione ha bisogno per operare.
- *Funzioni (come)*: le funzionalità eseguite dall'azienda.
- *Rete (dove)*: relativa alla distribuzione geografica dell'organizzazione.
- *Persone (chi)*: coloro coinvolti nell'esecuzione delle funzionalità aziendali.
- *Tempo (quando)*: relativo agli eventi significativi per il business.
- *Motivazione (perché)*: gli obiettivi dell'azienda.

Non viene definito alcun ordinamento tra le colonne.

Le righe invece mettono in luce i diversi *Punti di vista* che possono interessare diversi stakeholder:

- *Scopo (contestuale)*: descrive il sistema in termini di dimensioni, forma, relazioni e obiettivi di base.
- *Modello dell'azienda (modello concettuale)*: interessa il proprietario ed è una prospettiva orientata alla progettazione del business.
- *Modello del sistema (modello logico)*: per il progettista, fornisce una specifica dettagliata attraverso la definizione del modello del sistema.
- *Modello tecnologico (modello fisico)*: pensato per i realizzatori chi e deve tradurre il progetto logico in un progetto fisico.
- *Rappresentazione dettagliata*: raccoglie le specifiche da dare ai programmatori per la realizzazione effettiva del sistema.

Le celle definiscono le *Viste* (denominate *prospettive* da Zachman) che identificano i modelli e diagrammi in grado di fornire le informazioni su un particolare aspetto secondo un certo punto di vista.

Nel presente testo utilizziamo per la progettazione dell'architettura del sistema come EA utilizzando l'approccio ArchiMate proposto in [5], in cui gli aspetti da specificare sono i seguenti:

- Aspetti di business, legati agli obiettivi aziendali.
- Aspetti applicativi, in cui si identificano le funzionalità del sistema.
- Aspetti infrastrutturali, in cui vengono delineate le scelte tecnologiche di tipo generale

	What Cosa? DATI	How Come? FUNZIONI	Where Dove? RETE	Who Chi? PERSONE	When Quando? TEMPO	Why Perché? MOTIVAZIONE
Contestuale AMBITO Pianificatore	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 Owner	Modello Entità-Relazioni	Modello dei processi	Rete logistica	Modello della struttura organizzativa	Modello eventi	Relazioni tra obiettivi
Logico Modello sistema Progettista	Modello dei dati	Diagramma dei processi	Architettura del sistema distribuito	Diagramma ruoli e relazioni	Diagramma eventi	Diagramma regole
Fisico Modello tecnologico Costruttore	Progettazione dei dati	Specifiche delle funzioni	Architettura del sistema	Interfaccia uomo-macchina	Specifiche eventi	Specifiche regole
Dettaglio Componenti Sottocostrutture	Es. Definizione dei dati	Es. Applicazione (codice)	Es. Architettura di rete	Es. Architettura sicurezza	Dettagli eventi	Dettagli regole

# Capitolo 3

## 3 Modellare una EA con ArchiMate

### 3.1 Introduzione ad ArchiMate

I 3 livelli considerati vengono denominati in ArchiMate come *Livello di Business* (*Business layer*), *Livello Applicativo* (*Application layer*) e *Livello Tecnologico* (*Technology layer*).

La classificazione si articola su 2 dimensioni principali:

- **Livelli (*layer*)**: un costrutto può appartenere ad uno dei tre livelli presentati in precedenza: Business Layer, Application Layer, Technology Layer.
- **Aspetti**: un costrutto può rappresentare un elemento passivo, comportamentale o attivo.

### 3.2 Elementi principali del modello

Gli elementi del modello sono i componenti e le relazioni per collegare i componenti.

#### 3.2.1 Componenti

Sono gli elementi che vengono utilizzati per descrivere la struttura del sistema. È con un rettangolo.

- rettangoli semplici = elementi attivi
- rettangoli con angoli arrotondati = comportamenti
- rettangoli con una banda superiore aggiuntiva = elementi passivi

Solitamente il simbolo grafico è rappresentato nel rettangolo in alto a destra.

#### 3.2.2 Relazioni

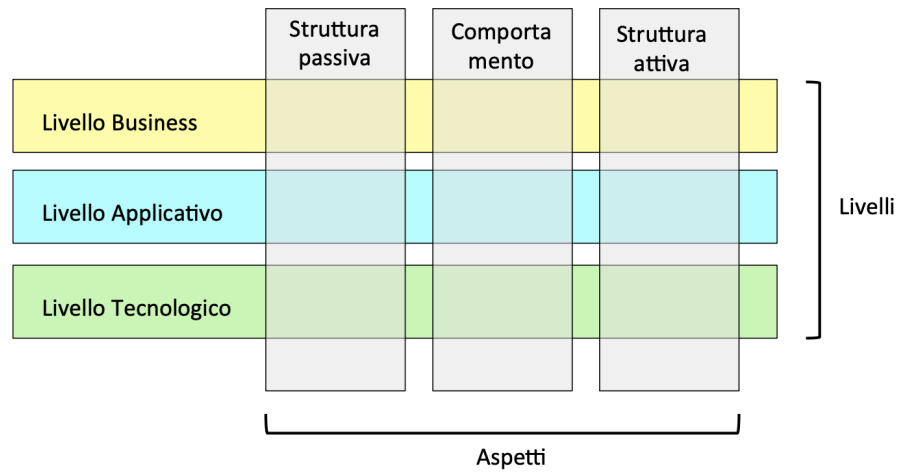
Le relazioni definiscono i possibili collegamenti tra gli elementi. Le tipologie sono le stesse per tutti i livelli e si distinguono in relazioni “*strutturali*”, di “*dipendenza*”, “*dinamiche*” e “*altre*”.

- relazioni strutturali → usate per rappresentare la struttura di un elemento
- relazioni di dipendenza → indicano quando un elemento dipende da un altro
- relazioni dinamiche → rappresentano dipendenze comportamentali tra gli elementi
- relazioni altre → che non ricadono nelle precedenti

Le principali relazioni:

- **Accesso** (*Access*): lega un componente comportamentale a un componente passivo o oggetto. È una relazione di dipendenza.

- *Composizione (Composition)*: lega un elemento a suoi componenti, i quali non possono esistere indipendentemente. È una relazione strutturale.
- *Realizzazione (Realization)*: indica un elemento ha un ruolo importante nella creazione, ottenimento o funzionamento di un elemento più astratto. È una relazione strutturale.
- *Assegnamento (Assignment)*: associa un elemento a quello che lo esegue. È una relazione strutturale.
- *Serve a (Serving)*: indica una relazione tra un servizio e un altro elemento che lo utilizza. È una relazione di dipendenza.



# Capitolo 4

## 4 Modellazione dei processi

### 4.1 Modellazione dei processi in BPMN

#### 4.1.1 Definizioni

Il concetto di processo è centrale nella rappresentazione dei sistemi informativi, in quanto consente di collegare le attività svolte per un certo scopo e di comprendere i flussi di dati.

Un *processo di business* (aka processo) è un insieme di attività eseguite secondo un ordine specifico per ottenere un certo obiettivo di business.

Le attività di un processo implementano i passi necessari a raggiungere un certo obiettivo e sono eseguite da **attori** (anche detti *agenti*) dell'organizzazione che sono localizzati in unità organizzative.

Gli attori vengono concettualizzati in **ruoli**.

Un ruolo descrive le capacità richieste a un attore per eseguire una certa attività così da poter scegliere un certo attore quando il processo entra in esecuzione.

→ Selezionare un attore per rivestire un ruolo è detto *risoluzione di ruolo*.

Un processo viene specificato secondo un modello di processo che specifica l'*ordine dei passi*. I passi possono essere *attività elementari o sottoprocessi*.

L'ordinamento dei passi in un modello può essere *semplice* o *complesso*.

L'ordinamento dei passi nel tempo è detto **flusso di controllo**.

Una singola esecuzione di un processo secondo un certo modello è detto *istanza del modello*.

#### 4.1.2 Business Process Modeling Notation

Modellare un processo permette di definire l'ordine preciso in cui le diverse attività che lo compongono devono svolgersi e anche le eventuali relazioni esistenti tra i processi interni all'organizzazione e i processi esterni di eventuali partner.

Una delle più diffuse notazioni per modellare i processi è la *Business Process Modeling Notation (BPMN)*.

Gli elementi principali che costituiscono un processo sono:

- **Attività**: le unità di lavoro del processo, è rappresentata con un rettangolo al cui interno è scritto il nome dell'attività.
- **Gateway**: sono usati per modellare flussi alternativi all'interno di un processo, come nodi di scelta o biforcazioni nel caso in cui più attività possono essere eseguite contemporaneamente. In BPMN è rappresentato con un rombo al cui interno è disegnato un simbolo che ne definisce la tipologia.

- **Eventi:** sono informazioni relative all'ambiente in cui il processo viene eseguito che possono essere generate dal processo o semplicemente ricevute durante la sua esecuzione, e possono provenire dall'interno o dall'esterno dell'organizzazione. In BPMN sono un cerchio al cui interno è disegnato un simbolo che ne definisce la tipologia.

Gli elementi di un processo sono collegati tra loro tramite il *Process Flow*.

I **sotto-processi** sono rappresentati come un'attività con un “+” in basso al centro.

Esistono anche diversi tipi di gateway:

- un **gateway XOR** (rappresentato con una “X”) → permette di scegliere uno solo tra due flussi alternativi.
- un **gateway AND** (identificato da un “+”) → permette di attivare tanti rami in parallelo e quindi di non definire un ordine preciso per l'esecuzione delle attività che si trovano su questi rami.

Infine, gli eventi possono essere raggruppati secondo diverse categorie:

- **eventi start** → rappresentati con un cerchio singolo sono le condizioni che fanno iniziare il processo
- **eventi intermedi** → rappresentati con un doppio cerchio
- **eventi end** → rappresentati con un cerchio singolo ma spesso sono le condizioni che fanno terminare un processo

Tra gli eventi più usati troviamo il *timer* che indica un qualsiasi evento di natura temporale e l'evento *messaggio*, rappresentato con una busta, che invece rappresenta uno scambio di informazioni tra l'interno e l'esterno dell'organizzazione.

Gli eventi possono anche essere suddivisi tra:

- **eventi throw** → prodotti dal processo
- **eventi catch** → prodotti all'esterno e catturati dal processo

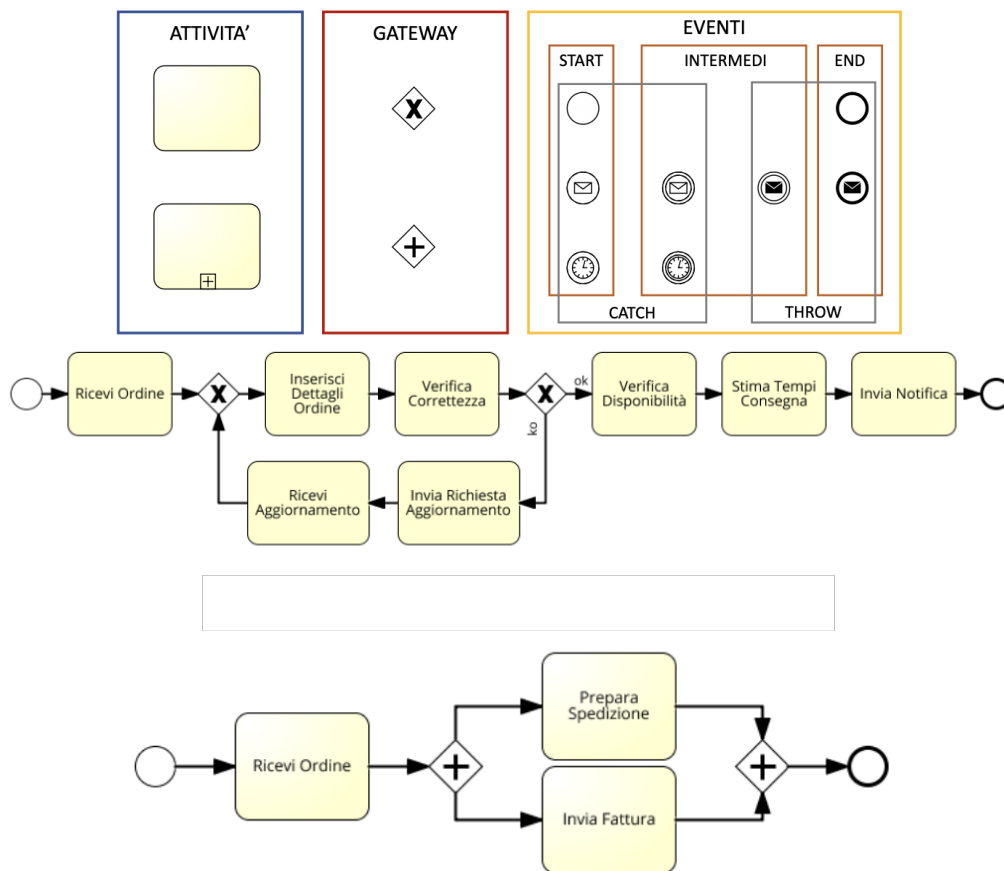
## 4.2 Identificazione dei processi

Discutiamo alcuni approcci che si possono utilizzare per l'identificazione dei processi.

### 4.2.1 Check-list settoriali

Si basa sulla somiglianza di organizzazioni che operano nello stesso settore. Prende come riferimento una lista di processi che sono stati identificati come tipici per il settore in cui opera l'organizzazione a partire dalla quale è possibile stabilire quali di questi processi sono presenti nella propria organizzazione.





Viene chiamato check-list proprio perché consiste nella selezione dei processi da considerare da una lista di riferimento.

Ha comunque varie limitazioni:

- risulta efficace per settori molto regolamentati, dove i processi sono molto simili, ma è poco applicabile in casi di poca regolamentazione.
- serve fare un'analisi approfondita dei processi aziendali, così da poter fare il confronto

#### 4.2.2 Catena del valore di Porter

La classificazione di Porter può essere utilizzata anche come punto di partenza per l'identificazione dei processi di un'organizzazione.

Ha delle limitazioni:

- si focalizza su attività suddivise in base alle unità organizzative, perdendo la visione interfunzionale e collaborativa tipica dei processi.
- le attività identificate da Porter tramite le 9 categorie sono un numero limitato delle attività dell'organizzazione analizzate ad un livello di aggregazione dei processi molto elevato.

- non è un modello generale che possa essere di riferimento per qualunque tipo di organizzazione
- non è immediato applicare efficacemente il modello ad organizzazioni orientate ai servizi o alle piccole e medie imprese in cui alcune delle attività identificate da Porter potrebbero essere del tutto assenti.

#### **4.2.3 Approccio Analitico**

È basato su un'analisi di tipo *top-down*. L'analisi parte dall'identificazione dei destinatari, cioè gli attori esterni con cui l'organizzazione interagisce e ricostruisce il processo a ritroso identificando prima l'output a loro fornito dall'organizzazione, i ruoli coinvolti e infine gli input del processo in termini sia di input fisici che di informazioni necessarie ai vari ruoli coinvolti per svolgere le loro attività.

È quindi necessario completare questo procedimento con l'identificazione dei processi di supporto, quelli cioè il cui focus non è rivolto all'esterno dell'organizzazione, ma alla sua gestione interna. I processi così identificati possono essere rappresentati utilizzando tecniche di modellazione dei processi.

L'approccio analitico è il più preciso tra quelli illustrati, ma anche il più dispendioso in termini di tempo e costi.

# Capitolo 5

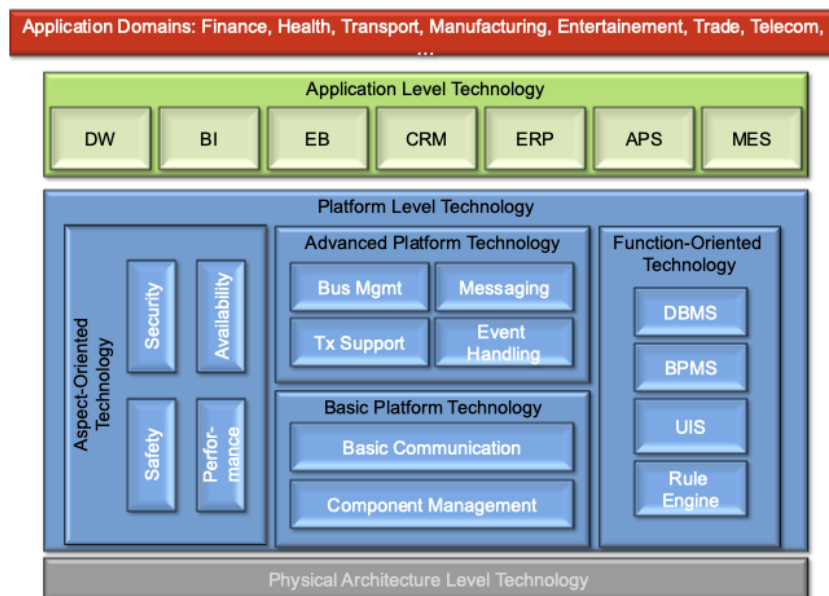
## 5 Introduzione agli elementi tecnologici

### 5.1 Introduzione

I componenti tecnologici dei sistemi informativi possono essere classificati secondo le categorie identificate in tabella.

Le tecnologie si dividono in 3 livelli: (i) *livello applicativo*, (ii) *livello di piattaforma* e (iii) *livello di architettura fisica*.

Ogni dominio applicativo è caratterizzato da obiettivi e esigenze diverse che inevitabilmente portano a scelte e realizzazioni differenti.



#### 5.1.1 La tecnologia a livello applicativo

Comprende tutte le applicazioni software che è possibile adottare a supporto delle diverse funzionalità aziendali nei sistemi operazionali e informativi.

L'insieme dei componenti a livello applicativo di un'azienda viene definito **portafoglio applicativo**.

Il portafoglio viene costruito con alcune di queste tipologie di applicazioni:

- **DW - Data Warehouse:** archivio contenente i dati di un'organizzazione progettati secondo un modello multidimensionale in modo tale da agevolare analisi a supporto delle scelte manageriali.
- **BI - Business Intelligence:** moduli per la raccolta e l'analisi di informazioni utili alla valutazione dell'attuale situazione aziendale e alla previsione di eventi futuri.

- ***EB - E-Business***: si riferisce alle applicazioni di supporto a funzionalità di business mediante strumenti digitali per la comunicazione, per la collaborazione tra imprese e per l'esecuzione di transazioni aziendali.
- ***CRM - Customer Relationship Management***: insieme di moduli a supporto di una strategia per la gestione delle interazioni con clienti potenziali e esistenti. Sono preposti alla memorizzazione e all'analisi delle interazioni con i clienti, e all'offerta di servizi innovativi mirati alla fidelizzazione della clientela.
- ***ERP - Enterprise Resource Planning***: suite software composta da moduli che sono volti a supportare principalmente le attività operative di un'azienda.
- ***APS - Advanced Planning and Scheduling***: insieme di applicazioni usate in ambito manifatturiero per gestire bene materie prime e capacità produttiva.
- ***MES - Manufacturing Execution Systems***: sistemi usati in ambito manifatturiero per tracciare e documentare il processo produttivo dalla trasformazione delle materie prime ai prodotti finiti

### 5.1.2 La tecnologia a livello di piattaforma

Le tecnologie a questo livello si dividono in 4 tipologie: (i) *Tecnologie orientate alle funzionalità*, (ii) *Tecnologie orientate agli aspetti non funzionali*, (iii) *Tecnologie di base* e (iv) *Tecnologie avanzate*.

Tra le *tecnologie orientate alle funzionalità* troviamo:

- ***DBMS - Database Management Systems***: sistemi per gestire, controllare e ottimizzare l'accesso e la manipolazione dei dati.
- ***BPMS - Business Process Management Systems***: sistemi per modellare, automatizzare, eseguire, controllare e ottimizzare i flussi di attività di business.
- ***UIS - User Interface Systems***: insieme di servizi capaci di supportare l'interazione da parte degli utenti e gestire interfacce diverse in modo adattivo e flessibile, con funzionalità di interazione avanzate e di gestione di elementi grafici e multimediali.
- ***Rule Engine***: sistema software che permette di definire, testare, eseguire a runtime le regole di business in modo indipendente dalle applicazioni.

Le *tecnologie orientate agli aspetti non funzionali* includono invece tecnologie in grado di garantire alcune caratteristiche non funzionali legate alle applicazioni, quali l'efficienza, la sicurezza, la correttezza, le prestazioni del sistema e la safety rispetto ai rischi fisici e ambientali.

Le *tecnologie di base* invece comprendono tecnologie per gestire la comunicazione tra applicativi e i componenti.

Tra le *tecnologie avanzate* invece citiamo i sistemi di messaging, di gestione di eventi e delle transazioni.

### 5.1.3 La tecnologia a livello di architettura fisica

I sistemi informativi attuali si basano su sistemi IT e sono passati da architetture centralizzate a distribuite per meglio adattarsi alle esigenze moderne.

Un aspetto chiave è la virtualizzazione delle risorse, che permette di utilizzare server virtuali (VM) invece di server fisici, rendendo i sistemi più flessibili e dinamici.

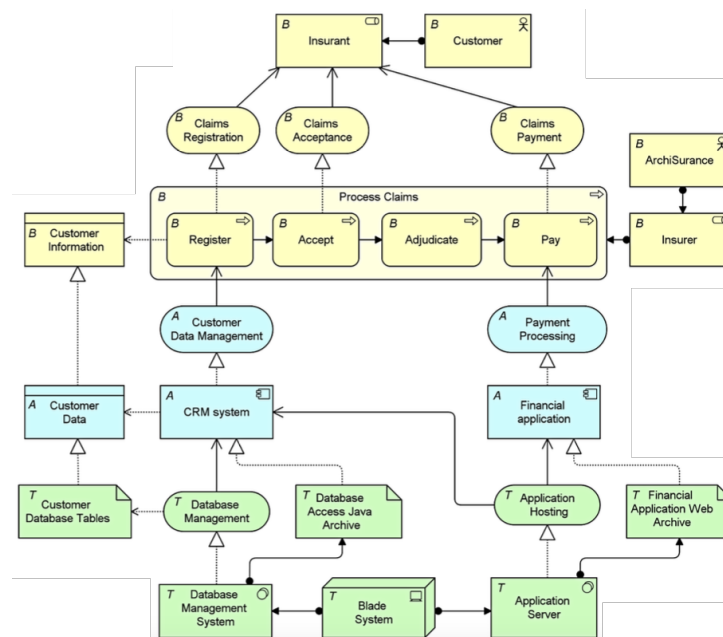
Questa flessibilità consente di allocare risorse virtuali in modo adattivo in base alle necessità di computazione. L'uso di tecnologie come il Cloud Computing facilita la gestione dell'infrastruttura e permette soluzioni di outsourcing.

## 5.2 Componenti applicativi e tecnologici in ArchiMate

La figura sottostante illustra un esempio di integrazione tra servizi di business e componenti applicative e tecnologiche usando ArchiMate. A livello di business, tre servizi principali vengono forniti all'assicurato: registrazione della richiesta, accettazione e pagamento del risarcimento. Il processo Process Claims supporta queste attività con i sottoprocessi Register, Accept e Pay, assegnati all'azienda ArchiSurance.

A livello applicativo, due servizi principali, Customer Data Management e Payment Processing, supportano i processi Register e Pay. Customer Data Management è gestito da un sistema CRM che utilizza un Database Management System (DBMS) per accedere ai dati dei clienti. Il DBMS opera su un nodo fisico chiamato Blade System, che costituisce l'architettura fisica del sistema.

Payment Processing è gestito da un'applicazione finanziaria e un sistema di Application Hosting supporta sia il CRM che la Financial Application, realizzati tramite un Application Server che funziona anch'esso sul nodo Blade System.



# Capitolo 6

## 6 Livello applicativo: ERP e CRM

Ora analizziamo più nel dettaglio le tecnologie a livello applicativo introdotte nel cap5.

### 6.1 Evoluzione dei Componenti Applicativi

All'inizio si usava il linguaggio macchina con la codifica di alcune funzionalità di base.

Negli anni '70, i SO cominciano a permettere di separare la logica applicativa dalla programmazione delle interfacce con l'hardware.

Negli anni '80 nascono i DBMS, che permettono di condividere dati tra applicazioni diverse, evitando la ridondanza e disaccoppiando la struttura fisica dei dati dal modo in cui l'applicazione vi accede.

Alla fine nascono le interfacce grafiche per semplificare l'interazione dell'utente con l'app, sostituendo le precedenti interfacce testuali. Un componente applicativo può essere scomposto in elementi diversi, sia dal pov funzionale che architetturale.

Le app si dividono per 3 livelli logici, detti *layer*:

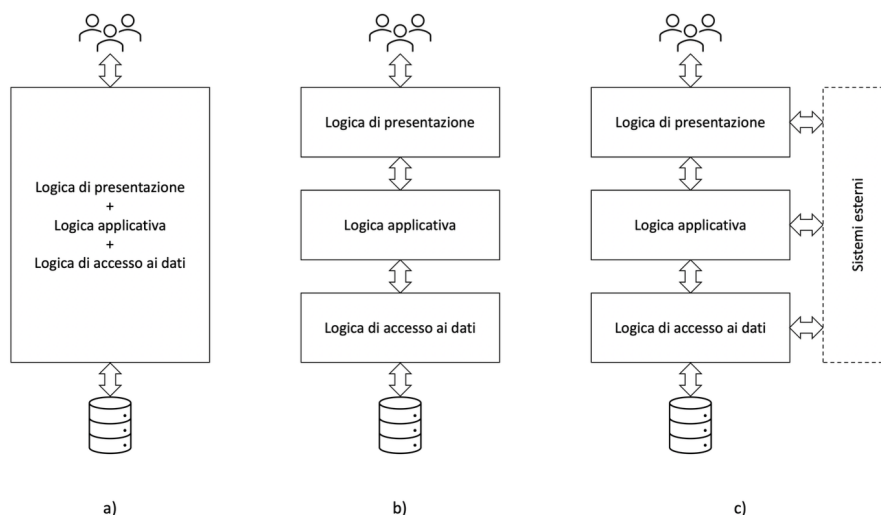
- **Livello presentazione (P)** → gestisce la logica di presentazione e l'interazione con l'utente: contiene le modalità di interfacciamento (grafico) e di rendering delle informazioni. Anche detto *front end*, è accessibile tramite una GUI (Graphical User Interface) o User Interface (UI). È solitamente costituito dal web.
- **Livello della logica applicativa / logica di business (A)** → gestisce le funzioni da mettere a disposizione dell'utente.
- **Livello di logica di accesso ai dati (D)** → gestisce l'informazione eventualmente con accesso ai database, a sistemi legacy o anche a risorse informative disponibili in rete.

Back End = livelli di logica applicativa e di accesso ai dati → accessibili tramite API.

Le API si mostrano come un elenco di signature di metodi, ma anche come metodi che ricevono in ingresso una query SQL e ne restituiscono il risultato.

I livelli logici quindi possono essere contenuti all'interno di uno stesso software artefact nel caso di *applicazioni monolitiche*, oppure scomposte su diversi software artefact che comunicano tra di loro tramite opportune interfacce nel caso di *applicazioni distribuite*.

Nel secondo caso è possibile distinguere tra **sistemi chiusi**, in cui i singoli livelli logici sono progettati per comunicare soltanto con i livelli adiacenti, e **sistemi aperti** in cui ogni livello logico può anche comunicare con sistemi esterni.



a) Applicazioni monolitiche b) Applicazioni distribuite a sistema chiuso c) Applicazioni distribuite a sistema aperto

## 6.2 Componenti Funzionali

Le moderne applicazioni aziendali sono composte da *moduli*, ciascuno progettato per gestire una funzione specifica, come contabilità, produzione, o risorse umane.

Questa modularità consente di adattare le applicazioni alle esigenze aziendali in modo flessibile e scalabile.

→ **Modularità:** I sistemi sono sempre più costruiti utilizzando moduli *COTS* (*Commercial Off-The-Shelf*), che possono essere integrati tra loro per formare un sistema completo. Questi moduli preconfigurati migliorano la manutenibilità, riducono i tempi di sviluppo e permettono una maggiore adattabilità alle esigenze aziendali.

## 6.3 ERP

Gli ERP sono suite software che integrano i processi aziendali principali, come produzione, logistica, vendite, contabilità e risorse umane. L'ERP gestisce e automatizza i flussi di informazioni tra i vari settori dell'azienda.

### 6.3.1 Storia degli ERP

L'ERP ha avuto origine negli anni '90 come evoluzione dei sistemi *CIM* (*Computer Integrated Manufacturing*) e *MRP* (*Material Requirements Planning*).

L'obiettivo era quello di creare un sistema integrato che permettesse di gestire tutti i processi aziendali in modo centralizzato.

### 6.3.2 Proprietà degli ERP

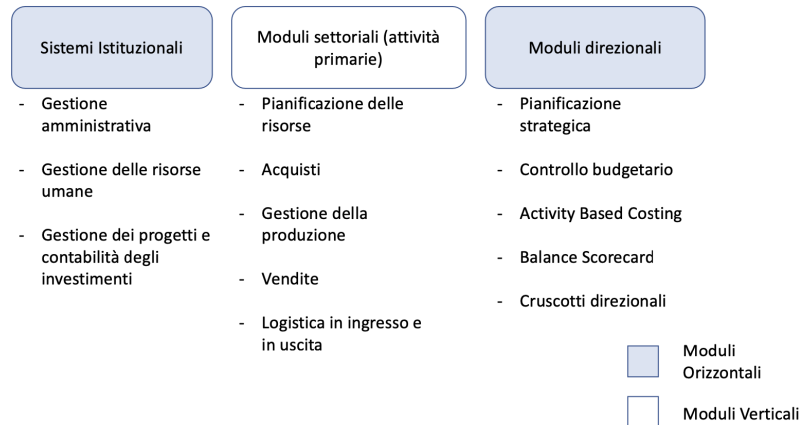
1. **Unicità dell'informazione:** I dati vengono inseriti una sola volta e condivisi tra i diversi moduli del sistema.
2. **Modularità:** L'ERP è composto da moduli che possono essere attivati in base alle esigenze aziendali.

3. **Prescrittività:** L'implementazione di un ERP spesso richiede che l'azienda si adatti ai processi previsti dal software, ma l'ERP stesso può essere adattato ai processi aziendali.

### 6.3.3 Funzionalità di un ERP

Gli ERP coprono una vasta gamma di processi aziendali, tra cui:

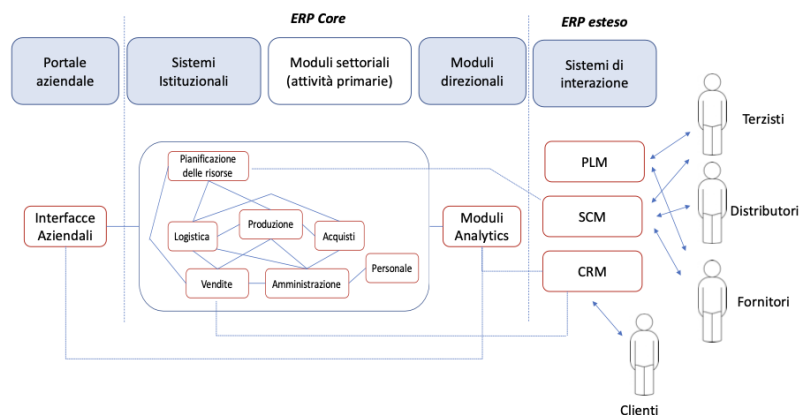
1. **Gestione della produzione:** Pianificazione e controllo del processo produttivo.
2. **Logistica:** Gestione degli acquisti e delle scorte.
3. **Contabilità:** Automazione della contabilità generale e della gestione finanziaria.
4. **Risorse umane:** Gestione dei dipendenti e delle buste paga.



### 6.3.4 Aspetti architetturali

Gli ERP sono passati da un'architettura client-server ad architetture web-based, che permettono una maggiore flessibilità, scalabilità e distribuzione del sistema.

Gli ERP moderni utilizzano anche tecnologie di analisi avanzata e di cloud computing, consentendo accesso ai dati da qualsiasi luogo e una maggiore efficienza.





### 6.3.5 Implementazione di un ERP

L'implementazione di un ERP richiede un'attenta pianificazione e personalizzazione. Solitamente, l'ERP viene acquistato come pacchetto software preconfigurato (*COTS*) e adattato alle esigenze specifiche dell'azienda.

Questo processo può portare a una riduzione dei costi, grazie all'automazione dei processi aziendali, e a un miglioramento dell'efficienza operativa.

### 6.3.6 Benefici degli ERP

Tra i principali benefici troviamo:

- **Automazione dei processi:** riducono la necessità di interventi manuali, aumentando l'efficienza.
- **Unificazione dei dati:** tutte le informazioni aziendali sono centralizzate e accessibili da tutti i moduli, riducendo la duplicazione dei dati.
- **Standardizzazione dei processi:** favoriscono l'adozione di processi aziendali standardizzati, migliorando l'efficacia complessiva.

## 6.4 CRM

Il *CRM (Customer Relationship Management)* è un sistema software che supporta le organizzazioni nella gestione delle interazioni con i clienti, mettendo il cliente al centro della strategia aziendale. Serve a comprendere meglio i bisogni e i comportamenti dei clienti, migliorando l'utilizzo dei canali di comunicazione e creando relazioni personalizzate e durature. ***L'obiettivo principale è aumentare la soddisfazione dei clienti e il valore reciproco tra impresa e cliente.***

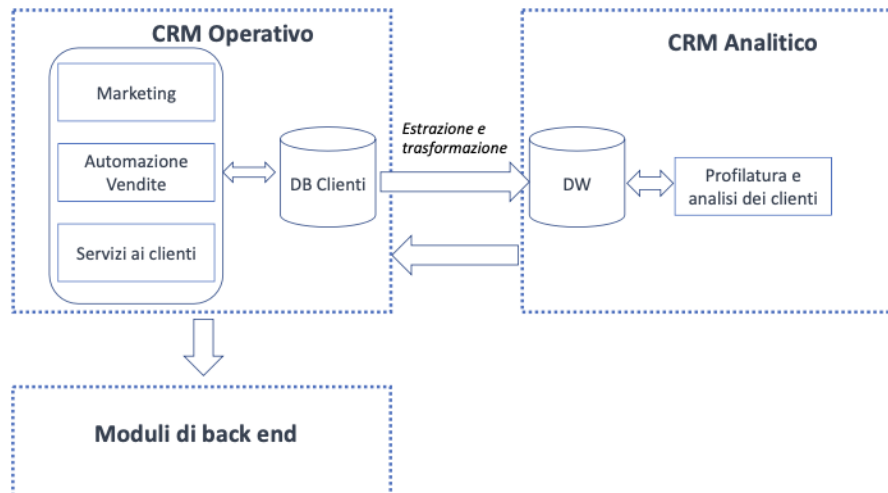
Il CRM viene spesso considerato *un'estensione dell'ERP*, ma è più utile in settori con interazioni frequenti, contratti di lunga durata, e possibilità di instaurare rapporti di lealtà, soprattutto in presenza di multicanalità (fisico, web, telefono). Oltre a supportare le vendite, fornisce anche funzioni post-vendita e di gestione della lealtà.

Il CRM è una *strategia aziendale* che, oltre ai moduli software, si concentra sul miglioramento degli obiettivi aziendali tramite l'analisi e la personalizzazione del marketing.

Esistono due componenti principali: il **CRM operativo**, che gestisce le interazioni quotidiane con i clienti, e il **CRM analitico**, che si occupa dell'analisi dei dati raccolti.

Il **CRM collaborativo** facilita la condivisione di informazioni tra i vari attori aziendali.

Il CRM operativo registra tutte le interazioni con i clienti e alimenta un data warehouse, che viene utilizzato dal CRM analitico per eseguire analisi approfondite. Queste interazioni possono generare eventi da gestire tramite sistemi di back-end, come l'ERP, per la gestione delle richieste di vendita.



Esempio di architettura di massima per la gestione dei dati in un CRM

#### 6.4.1 CRM operativo

Il *CRM operativo* gestisce le interazioni quotidiane tra l'azienda e i clienti. Si occupa della gestione di tutte le comunicazioni sui vari canali, come telefonate, email e visite in negozio, centralizzando le informazioni in una base di dati unificata.

È suddiviso in diversi moduli che supportano attività fondamentali come vendite, marketing e assistenza clienti:

- **Marketing:** le aziende utilizzano il CRM per sviluppare campagne di marketing personalizzate e fidelizzare i clienti esistenti, piuttosto che focalizzarsi esclusivamente sull'acquisizione di nuovi clienti. Il CRM consente di segmentare i clienti sulla base dei dati raccolti (ad esempio, cronologia degli acquisti o interazioni con l'azienda), permettendo così di ottimizzare le strategie di marketing e aumentare la soddisfazione del cliente.
  - *Generazione liste clienti:* crea liste di clienti target basate su dati rilevanti, come interazioni precedenti o comportamenti di acquisto.
  - *Gestione campagne:* automatizza le attività di marketing, migliorando l'efficacia e riducendo gli errori umani, aumentando la produttività e consolidando le relazioni con i clienti.
  - *Cross-selling e Upselling:* aiuta a vendere prodotti aggiuntivi (cross-selling) o a proporre prodotti più costosi (upselling) in base alle preferenze del cliente.
- **Automazione Vendite:** include moduli come il *Sales Force Automation (SFA)*, che supporta gli agenti di vendita in tutte le fasi del processo di vendita, dalla gestione dei contatti alla creazione di offerte e ordini. Questi moduli permettono di automatizzare i processi di vendita, migliorando la gestione e il monitoraggio delle performance.
  - *Gestione vendite:* automatizza o offre un supporto a tutte le fasi di un processo di vendita.

- *Gestione contatti*: si occupa della gestione dati dei clienti e identifica eventuali clienti per le vendite future. Tiene anche in considerazione info aggiuntive che possono contribuire a migliorare il rapporto con il cliente.
- *Gestione opportunità*: vuole trovare nuovi clienti o organizzazioni per vendite future.
- **Servizi ai clienti**: Il CRM operativo include strumenti per il supporto post-vendita. Un buon servizio clienti, specialmente nel post-vendita, è essenziale per fidelizzare il cliente. I moduli più comuni sono:
  - *Contact Center*: gestisce le chiamate in entrata e in uscita, utilizzando funzionalità avanzate come *l'Interactive Voice Response (IVR)* e la distribuzione automatica delle chiamate agli operatori.
  - *Web-based Self-Service*: permette ai clienti di cercare soluzioni ai propri problemi attraverso un portale web, riducendo i costi del supporto diretto.
  - *Call Scripting*: fornisce agli operatori del call center risposte predefinite e suggerimenti per gestire le chiamate in modo più efficiente e risolvere i problemi dei clienti.

#### 6.4.2 CRM analitico

Il CRM analitico utilizza i dati raccolti dal CRM operativo per analizzare il comportamento e le preferenze dei clienti, al fine di individuare schemi significativi che possano migliorare la strategia aziendale.

Questo tipo di CRM utilizza tecniche di *data mining* e *data warehousing* per segmentare i clienti e fornire previsioni utili.

Le principali finalità del CRM analitico sono:

- **Reporting**: Fornisce informazioni dettagliate sui clienti, permettendo all'azienda di comprendere meglio chi sono e quali sono le loro caratteristiche.
- **Analisi**: Segmenta i clienti in base a diversi criteri, come clienti attivi, dormienti o inattivi, e permette di indirizzare campagne e azioni mirate.
- **Previsioni**: Consente di prevedere i comportamenti futuri dei clienti, come le probabilità di acquisto o abbandono del servizio, permettendo all'azienda di ottimizzare le proprie strategie.

Il CRM analitico invia i risultati delle analisi al CRM operativo per supportare decisioni come la segmentazione delle campagne di marketing o suggerimenti di vendita personalizzati.

#### 6.4.3 CRM collaborativo

Vuole calcolare alcuni indici rilevanti per l'intera azienda, quali la **profittabilità** e la **soddisfazione dei clienti** e di **condividere** tali informazioni per valutare l'impatto dell'utilizzo di CRM.

# Capitolo 7

## 7 Livello applicativo: Analisi dei dati

### 7.1 Data Warehousing

I dati sono fondamentali non solo operativamente, ma anche **per decisioni tattiche e strategiche all'interno di un'organizzazione**. Con l'aumento della complessità delle analisi richieste, come la correlazione tra variabili (es. residenza di un cliente e uso di determinati servizi), è essenziale accedere a dati aggregati e strumenti in grado di supportare queste analisi, racchiusi nell'ambito della Business Intelligence (BI).

Inizialmente, il supporto decisionale era basato su **report statici**, prodotti regolarmente ma non interattivi, con limiti in termini di aggiornamento e dati disponibili, poiché si basavano solo su fonti interne. Successivamente, i fogli di calcolo, come Excel, hanno introdotto maggiore flessibilità permettendo all'utente di personalizzare le analisi. Tuttavia, questi strumenti presentano difficoltà nell'integrazione e mancano di controllo di qualità e integrità dei dati.

Per rispondere alle esigenze di analisi avanzate ai livelli superiori della piramide di Anthony, è emerso il concetto di data warehouse. Un **data warehouse** raccoglie dati **sia interni sia esterni**, organizzati in una struttura ottimizzata per le analisi, con strumenti capaci di fornire risposte rapide e con dati aggiornati e integrati.

#### 7.1.1 Caratteristiche dei dati

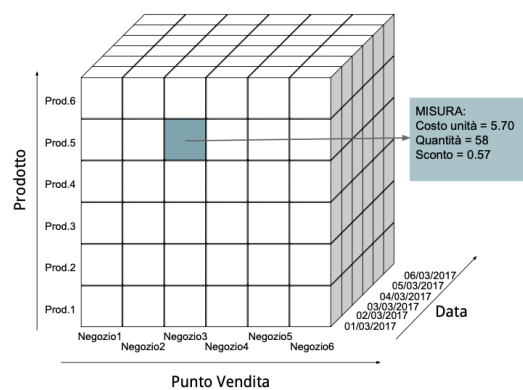
Un data warehouse è un **sistema OLAP** (Online Analytical Processing) progettato per supportare attività strategiche e decisionali, a differenza dei DBMS tradizionali basati su OLTP (On-line Transaction Processing), che gestiscono transazioni brevi e frequenti. I dati analitici si distinguono dai dati operazionali per vari aspetti:

- **Obiettivo:** i dati sono utilizzati per prendere decisioni sugli sviluppi futuri, ma anche per identificare problematiche organizzative o possibilità di crescita.
- **Utenti:** gli utilizzatori dei dati sono i manager di alto e medio livello che utilizzano l'informazione per migliorare i risultati dell'organizzazione
- **Orizzonte temporale:** mentre i dati operazionali si basano principalmente sul presente, i dati analitici utilizzano dati storici relazionandoli ai dati attuali per identificare problematiche, tendenze e periodicità
- **Livello di dettaglio:** i dati analitici sono solitamente dati aggregati a partire dai dati operazionali o da altre fonti secondo diversi livelli di granularità
- **Accesso:** diversamente dagli utenti dei dati operazionali, gli utenti dei dati analitici accedono all'informazione solo in lettura

Per gestire questi dati, si usa il **modello multidimensionale**, rappresentato come un **ipercubo** con dimensioni di analisi (es. tempo, prodotto, punto vendita) e misure quantitative (es. quantità venduta). Le dimensioni possono essere organizzate in gerarchie per consentire analisi a diversi livelli di granularità, aggregando i dati per categorie specifiche.

Gli elementi costitutivi di una base di dati multidimensionale sono:

- **Fatto**: elemento dell'ipercubo ottenuto specificando un valore per ogni dimensione
- **Dimensione**: coordinate di ciascun elem. che corrispondono a dimensioni di analisi
- **Misura**: valore quantitativo del fatto elementare



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

Esempio di gerarchia per la dimensione prodotto.

Il livello di granularità cresce da destra verso sinistra.

### 7.1.2 Data warehouse

Un **data warehouse** è un **sistema di archiviazione di dati storici, sintetici e integrati rilevanti per un'organizzazione**, progettato per supportare processi decisionali complessi.

Il data warehousing comprende tutte le operazioni di popolazione, manutenzione e interrogazione della base dati direzionale. Essendo un sistema OLAP (Online Analytical Processing), è ottimizzato per eseguire poche query complesse, richiedendo l'aggregazione di grandi volumi di dati strutturati in schemi multidimensionali (spesso a stella). La sua efficienza è misurata dal **tempo di risposta per rispondere a domande strategiche**, come il profitto netto in specifiche aree e periodi.

I sistemi OLAP supportano operazioni complesse e casuali, con grandi volumi di dati aggregati e storici, senza richiedere le proprietà ACID (Atomicità, Coerenza, Isolamento, Durabilità) poiché sono operazioni di sola lettura.

#### Caratteristiche FASMI di un sistema OLAP:

- **Fast**: risposte rapide alle query
- **Analytical**: supporto per analisi complesse
- **Shared**: accesso condiviso per più utenti con diversi permessi

- *Multidimensional*: visione multidimensionale dei dati
- *Informational*: disponibilità di tutte le informazioni rilevanti.

#### **Caratteristiche essenziali del data warehouse:**

- Orientato alle entità: focalizzato su entità principali come vendite, ordini, prodotti
- Integrato: dati consolidati da fonti interne ed esterne
- Variabile nel tempo: dati storici con etichette temporali
- Persistente: i dati sono archiviati in sola lettura, senza modifiche

### **7.1.3 Architettura del data warehouse**

Il **data warehousing** è un insieme di attività che definiscono, costruiscono e mantengono un data warehouse, una base di dati che raccoglie, integra e struttura le informazioni aziendali. Il data warehouse si popola attraverso il processo ETL (Extraction, Transformation, Loading), che estrae i dati dalle sorgenti (come basi di dati operative o dati esterni), li trasforma nella struttura desiderata e li carica nel data warehouse. L'architettura del data warehouse può essere a due o tre livelli, a seconda della presenza o meno di una staging area (una base dati intermedia).

A valle del data warehouse possono esserci i data mart, che sono piccoli data warehouse tematici contenenti un sottoinsieme delle informazioni. I data mart consentono di definire viste personalizzate per specifici gruppi di utenti, riducendo le dimensioni o modificando la granularità dei dati.

Un'architettura centralizzata utilizza un solo data warehouse, mentre una distribuita impiega più data mart. Un esempio pratico di data warehousing è una catena di supermercati che analizza dati provenienti da diverse sorgenti (vendite, scorte, campagne pubblicitarie, meteo) per ottimizzare la gestione e la pianificazione. I dati vengono elaborati nel data warehouse, ma specifici settori possono utilizzare i data mart per estrarre report personalizzati.

#### **ETL (Extraction, Transformation, Loading)**

Il processo ETL si occupa di estrarre, trasformare e caricare i dati nel data warehouse. Le fasi principali sono le seguenti:

1. **Estrazione**: i dati vengono estratti dalle sorgenti in due modalità:
  - **Statica**: vengono estratti tutti i dati disponibili nelle sorgenti.
  - **Incrementale**: vengono estratti solo i dati nuovi o modificati dall'ultimo aggiornamento.
2. **Trasformazione**: i dati estratti vengono preparati per l'inserimento nel data warehouse attraverso diverse operazioni, tra cui:

- **Pulizia dei dati (Data cleaning):** correzione di errori, eliminazione di dati mancanti, valori inammissibili o incongruenti.
  - **Riconciliazione:** integrazione dei dati da diverse fonti, mettendo in relazione i dati riferiti allo stesso "oggetto".
  - **Standardizzazione dei formati:** omogeneizzazione dei dati, che include:
    - Congiunzione o separazione di campi: combinazione di più campi in uno o viceversa.
    - Standardizzazione dei codici di classificazione: allineamento delle convenzioni diverse sotto una codifica comune.
    - Standardizzazione del formato dei dati: omogeneizzazione dei formati di rappresentazione dei dati.
  - **Ricerca e eliminazione dei duplicati:** identificazione e rimozione delle informazioni duplicate.
3. **Caricamento:** i dati trasformati vengono infine caricati nel data warehouse seguendo il modello multidimensionale

Il processo ETL è fondamentale per garantire che i dati siano corretti, coerenti e pronti per l'analisi.

## METADATI

Le funzionalità degli strumenti ETL sono supportate e documentate dai metadati, ossia dai *dati relativi ai dati*. I metadati sono una parte fondamentale dell'architettura di un data warehouse e sono solitamente raccolti in un repository che include diverse informazioni, tra cui:

1. **Struttura del data warehouse:** I metadati descrivono la struttura del data warehouse, inclusi schemi, viste, dimensioni, gerarchie, e la scomposizione in data mart con la relativa localizzazione.
2. **Metadati operazionali:** Questi metadati si riferiscono alla storia dei dati, documentando la fonte di origine e le trasformazioni subite.
3. **Metadati per mappare i dati operazionali ai dati nel data warehouse:** Forniscono informazioni sulle sorgenti dei dati, il loro contenuto, le regole di trasformazione e di aggiornamento del data warehouse.
4. **Statistiche d'uso:** I metadati descrivono anche come e con che frequenza viene utilizzato il data warehouse.

### 7.1.4 Modello concettuale e Dimensional Fact Model (DFM)

Come precedentemente discusso, i data warehouse si basano sul modello multidimensionale, nel quale i fatti, ossia gli elementi di interesse per l'analisi, sono rappresentati tramite un insieme di misure, definite dalle dimensioni che fungono da coordinate del fatto.

Esistono diversi modelli concettuali per rappresentare i sistemi multidimensionali, e il più utilizzato è il ***Dimensional Fact Model*** (DFM). Nel DFM, il fatto è rappresentato come un rettangolo contenente le misure corrispondenti. Le dimensioni sono invece rappresentate come cerchi etichettati collegati al fatto. Le dimensioni possono essere:

- **Semplici attributi del fatto**
- **Gerarchie:** rappresentate come alberi, che hanno come radici le dimensioni di base

Alcuni attributi delle gerarchie possono essere opzionali, indicati nel DFM tramite una barra sulla linea corrispondente all'attributo.

**Esempio di DFM:** Nel caso di una catena di supermercati, il fatto potrebbe essere rappresentato dalla *vendita*, con misure quali la quantità venduta, il valore unitario e l'eventuale sconto. Le dimensioni del fatto possono includere: la data della vendita, il luogo, il prodotto venduto, il cliente.

Ogni dimensione è associata a una gerarchia, che permette di effettuare analisi più o meno dettagliate sui fatti. Ad esempio, è possibile analizzare le vendite totali di un prodotto in un periodo specifico, raggruppando i punti vendita di Milano per sesso dei clienti. L'aggregazione avviene lungo le gerarchie delle dimensioni considerate. In questo esempio, l'attributo *età* del cliente è opzionale.

Il DFM rappresenta un ***ipercubo relativo*** alle vendite per la catena di supermercati. Tuttavia, un data warehouse non è composto da un solo ipercubo, ma da molteplici ipercubi, ognuno relativo a un diverso fatto di interesse. Ad esempio, un data warehouse di un'azienda che vuole analizzare vendite, efficienza del supporto clienti e ottimizzazione delle forniture avrà tre ipercubi distinti: vendite, assistenza clienti, forniture.

Per ogni ipercubo sarà modellato un DFM che definisce il fatto, le misure, le dimensioni e le gerarchie di interesse per ciascun tipo di analisi.

### 7.1.5 Modelli logici del data warehouse

Una volta definito il modello concettuale del data warehouse, il passo successivo è tradurlo in un modello logico. Questa fase richiede la scelta di un DBMS adeguato per gestire e interrogare le informazioni multidimensionali, considerando le esigenze specifiche dell'azienda.

## MODELLI MOLAP

Il modello MOLAP (Multidimensional OLAP) rappresenta una delle scelte più dirette per implementare un data warehouse multidimensionale. Questo modello traduce il modello concettuale in una base di dati che consente interrogazioni veloci tramite motori specializzati, ottimizzando le prestazioni nelle operazioni di aggregazione dei dati. I principali vantaggi di un sistema MOLAP sono i seguenti:

- **Traduzione esatta del modello concettuale:** il modello MOLAP riflette fedelmente la struttura multidimensionale concepita nel modello concettuale.



- **Interrogazioni veloci:** la gestione dei dati attraverso una base di dati multidimensionale consente operazioni rapide, particolarmente adatte per analisi complesse.
- **Ottimizzazione delle performance:** i motori MOLAP sono progettati per trattare e manipolare grandi volumi di dati multidimensionali in modo altamente efficiente.

Tuttavia, non mancano alcuni svantaggi legati all'uso di MOLAP:

- **Conoscenze specifiche richieste:** le basi di dati multidimensionali sono meno diffuse rispetto ai tradizionali database relazionali, e richiedono competenze avanzate che potrebbero non essere comuni tra gli utenti.
- **Modelli proprietari:** MOLAP utilizza linguaggi e modelli proprietari, per cui non esistono standard universalmente adottati, rendendo l'integrazione con altre tecnologie più complicata.
- **Allocazione inefficiente dello spazio:** il modello MOLAP si basa su un concetto di ipercubo in cui ogni possibile combinazione delle dimensioni necessita di una cella di memoria, anche se la cella rimane vuota quando non si verifica il fatto associato.

## MODELLI ROLAP

Il modello ROLAP (Relational OLAP) è un'alternativa al modello MOLAP, in cui il modello concettuale multidimensionale viene tradotto in un formato relazionale. I dati vengono rappresentati utilizzando tabelle e interrogazioni SQL standard. Le operazioni di JOIN sulle tabelle descrivono il fatto e le sue dimensioni. Tra i vantaggi principali di ROLAP troviamo:

- **Facilità di utilizzo:** gli utenti già esperti in SQL possono interagire facilmente con il sistema senza dover acquisire competenze specifiche nel trattamento dei dati multidimensionali.
- **Efficienza nello spazio:** rispetto al modello MOLAP, ROLAP occupa meno spazio poiché ogni fatto viene rappresentato come una singola riga nelle tabelle, senza dover riservare celle vuote per tutte le possibili combinazioni delle dimensioni.
- **Uso di tecnologie note:** essendo basato su un modello relazionale, ROLAP sfrutta tecnologie ampiamente utilizzate per la gestione di basi di dati.

Tuttavia, ROLAP presenta anche degli svantaggi:

- **Difficoltà di adesione al modello concettuale:** la rappresentazione relazionale dei dati è meno aderente al modello multidimensionale, il che può risultare in una gestione meno intuitiva dei dati.
- **Lentezza nelle interrogazioni:** le operazioni di JOIN necessarie per rispondere alle interrogazioni sui dati multidimensionali possono risultare lente e poco efficienti rispetto ai sistemi MOLAP.

## MODELLI HOLAP

Il modello HOLAP (Hybrid OLAP) cerca di combinare i punti di forza di MOLAP e ROLAP, adottando una soluzione ibrida. In questo modello, una base di dati relazionale viene utilizzata per il data warehouse, mentre i data mart tematici, contenenti dimensioni limitate, vengono gestiti tramite una base di dati multidimensionale. I principali vantaggi di HOLAP includono:

- **Efficienza nello spazio:** la parte relazionale del modello riduce l'occupazione di spazio nel data warehouse, mantenendo comunque prestazioni elevate per le interrogazioni.
- **Accesso tramite tecnologie standard:** consente l'interazione con strumenti SQL conosciuti, riducendo la necessità di acquisire nuove competenze.
- **Ottimizzazione delle performance per i data mart:** l'uso di MOLAP per i data mart garantisce interrogazioni più rapide e precise per dimensioni limitate.

Tuttavia, l'approccio HOLAP comporta una maggiore complessità nella gestione della parte ibrida del sistema, richiedendo un'integrazione efficace tra le due componenti.

## SCHEMI MULTIDIMENSIONALI SU BASI DI DATI RELAZIONALI

Per mappare un modello multidimensionale su un database relazionale, è necessario utilizzare schemi relazionali che descrivano la struttura dei dati. Due degli approcci più comuni per questo scopo sono lo *schema a stella* e lo *schema a fiocco di neve*.

### SCHEMA A STELLA

Nello schema a stella, si utilizzano due tipologie principali di tabelle:

- **Tabella dei fatti:** contiene gli attributi relativi alle misure del fatto e ogni riga rappresenta un fatto specifico.
- **Tabelle delle dimensioni:** descrivono le dimensioni associate al fatto e contengono gli attributi relativi alla gerarchia delle dimensioni.

Questo approccio semplifica le interrogazioni, che vengono realizzate tramite operazioni di JOIN tra la tabella dei fatti e le tabelle delle dimensioni. Tuttavia, l'integrazione delle gerarchie dimensionale risulta semplificata e appiattita in un'unica tabella.

### SCHEMA A FIOCCO DI NEVE

Lo schema a fiocco di neve rappresenta un'alternativa più complessa, in cui ogni dimensione è associata a più tabelle, con relazioni tra di esse per conservare la struttura gerarchica e le dipendenze funzionali. Questo approccio consente di mantenere meglio la struttura multidimensionale, ma può risultare più complesso da gestire rispetto allo schema a stella.

## CONCLUSIONI

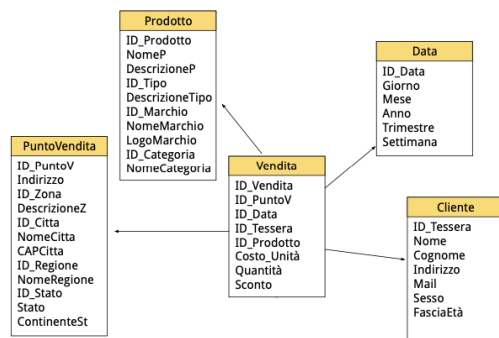
La scelta tra MOLAP, ROLAP e HOLAP dipende dalle specifiche esigenze dell'azienda,

dalle prestazioni desiderate e dalla familiarità degli utenti con le tecnologie coinvolte. Gli schemi a stella e a fiocco di neve rappresentano due diverse modalità per mappare un modello multidimensionale su una base di dati relazionale, ognuna con i propri vantaggi e svantaggi legati alla semplicità e alla complessità del modello.

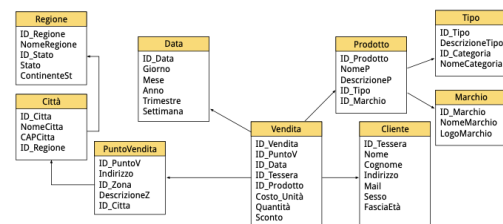
### 7.1.6 Operazioni sul data warehouse

Il data warehousing include tecniche per analizzare i dati attraverso l'interazione con l'ipercubo dei fatti, che consente agli utenti di esplorare informazioni dettagliate in base alle proprie esigenze. Le principali tecniche, applicabili tramite operatori OLAP, sono: *drill down*, *roll up*, *slice* e *dice*.

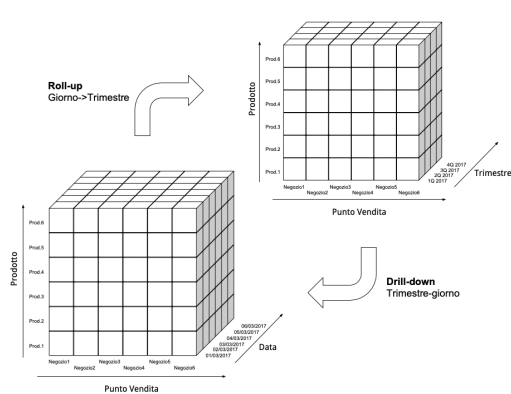
- Il ***drill-down*** consente di approfondire i dati, passando da un livello aggregato a uno più dettagliato lungo una gerarchia (es. da analisi trimestrale a giornaliera delle vendite).
- Il ***roll-up*** riduce il dettaglio, passando a un livello più aggregato (es. da analisi mensile a annuale).
- Il ***slice*** isola una porzione dei dati fissando il valore di una dimensione, creando una "fetta" dell'ipercubo.
- Il ***dice*** permette di selezionare un insieme di coordinate a vari livelli gerarchici, ottenendo un ipercubo ridotto con dati specifici (es. vendite di un prodotto in un negozio).



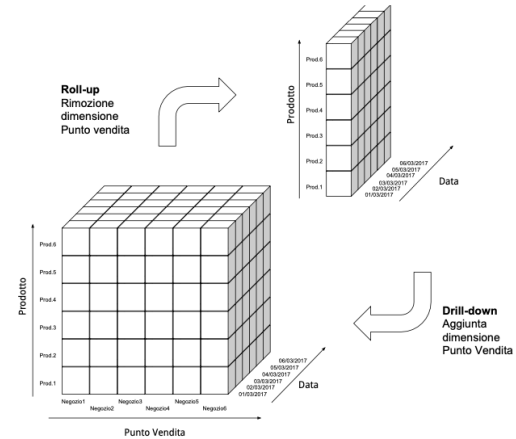
schema a stella



schema a fiocco di neve



esempio di *roll-up* e *drill-down* con aggregazione/disaggregazione sulla dimensione *Data*



esempio di *roll-up* e *drill-down* con eliminazione di dimensione *Punto vendita*

### 7.1.7 Ciclo di vita del data warehouse

Il data warehouse di un'organizzazione è composto da più ipercubi, ognuno relativo a un fatto di interesse. La costruzione segue un approccio iterativo: si inizia con il primo ipercubo e successivamente se ne aggiungono altri. La fase di popolamento utilizza strumenti ETL, ma è complessa e richiede attenzione agli aggiornamenti dei dati. Gli aggiornamenti dei fatti e delle dimensioni, come gli indirizzi dei clienti, possono causare incongruenze. Il processo ETL non è completamente automatizzato e richiede tempo, specialmente nella fase di trasformazione dei dati, rendendo il mantenimento del data warehouse oneroso. Per ridurre la distanza tra i dati sorgenti e il data warehouse, sono stati sviluppati approcci alternativi basati su big data, che permettono analisi in tempo reale senza la necessità di una progettazione dettagliata.

## 7.2 Data Mining

Questa sezione introduce il **Data Mining**, una famiglia di tecniche utilizzate per estrarre informazioni utili da grandi insiemi di dati. Diversamente dal Data Warehousing, che si concentra sulla creazione di una base di dati organizzata per l'analisi, il Data Mining mira a scoprire schemi nascosti e a costruire modelli predittivi.

I sistemi informativi decisionali hanno la necessità di accedere a dati aggregati ed eseguire interrogazioni complesse, come l'analisi della correlazione tra diverse variabili. Per garantire il successo delle attività tattiche e strategiche, è fondamentale disporre di strumenti in grado di estrarre le informazioni di interesse da diverse fonti (interne ed esterne) in modo veloce ed efficace.

Le tecniche di Data Mining possono essere classificate in diverse categorie, considerando la possibilità di istruire gli algoritmi con esempi ideali, si possono distinguere le tecniche di data mining in tecniche in apprendimento supervisionato e non supervisionato.

### 7.2.1 Caratterizzazione e Discriminazione

Gli algoritmi di caratterizzazione e discriminazione hanno l'obiettivo di estrarre informazioni rilevanti che possano descrivere in modo preciso e sintetico i dati contenuti nella

base dati analizzata.

Un esempio di questa tecnica è la *generalizzazione*, che riduce il livello di dettaglio per ridurre la numerosità degli elementi da analizzare. Questa riduzione può essere effettuata aggregando gli elementi, trovando caratteristiche comuni all'interno del dataset.

La riduzione può essere effettuata utilizzando tecniche di aggregazione come quelle usate nel data warehouse (*roll up*), oppure, nel caso in cui la gerarchia degli attributi non sia conosciuta, eliminando gli attributi che hanno una variabilità troppo elevata.

### 7.2.2 Regole Associative

Le regole associative mirano a identificare relazioni tra elementi frequenti in un set di dati.

Le regole di associazione si basano su due metriche: **supporto** e **confidenza**.

Il **supporto** di una regola misura la frequenza con cui una combinazione di elementi appare nel set di dati.

La **confidenza** di una regola, invece, indica la probabilità che un elemento sia presente dato che un altro elemento è già presente.

Ad esempio, una regola associativa potrebbe indicare che l'80% dei clienti che acquistano il pane acquistano anche il latte (confidenza dell'80%) e che questa combinazione di prodotti si presenta nel 10% delle transazioni (supporto del 10%).

### 7.2.3 Classificazione

La classificazione assegna elementi a categorie predefinite. Un esempio è l'uso degli *alberi di decisione*, modelli ad albero che utilizzano una serie di regole per classificare gli elementi in base ai valori dei loro attributi.

La costruzione di un albero di decisione si svolge tramite ragionamenti successivi. Partendo da un training set etichettato, si seleziona un attributo come nodo radice e si suddivide il dataset sulla base dei diversi valori assunti da tale attributo. La scelta dell'attributo è dettata da valutazioni legate all'entropia che misura la quantità di informazione estraibile dall'attributo con l'obiettivo di avere un salto di entropia (per esempio, guadagno informativo) elevato tra i livelli dell'albero e entropia minima, idealmente pari a zero, per le foglie dell'albero.

Per ogni sotto-classe si seleziona un ulteriore attributo e si riesegue la stessa operazione. L'obiettivo è quello di ottenere nelle foglie un insieme di casi che appartengono tutti alla stessa classe di assegnazione. **Il procedimento si ferma quando si è raggiunta la massima profondità stabilita per l'albero**, o quando la sotto-classe ottenuta ha una variabilità molto bassa per cui un'ulteriore suddivisione non permette di avere un insieme più omogeneo.

### 7.2.4 Clustering

Il *clustering* raggruppa elementi simili in cluster in base alla loro somiglianza. La funzione di somiglianza determina quanto due elementi sono simili in base ai valori dei loro attributi.

Un esempio di algoritmo di clustering è **k-means**, che utilizza un approccio iterativo per assegnare elementi a cluster in base alla loro distanza dai centroidi dei cluster.

L'approccio inizia con l'**identificazione della posizione dei centroidi**, uno per ogni cluster, rappresentanti il punto con coordinate corrispondenti al valore medio degli attributi per gli elementi di un insieme. Inizialmente i centroidi vengono posizionati in modo **casuale**. Gli elementi del dataset vengono quindi assegnati ad un cluster sulla base della loro **distanza** da ogni centroide. Per ogni insieme individuato, viene ricalcolata la posizione dei centroidi e si ripete l'assegnamento degli elementi. L'algoritmo si ferma quando **il sistema converge** (i centroidi hanno una posizione stabile e gli elementi assegnati ai set non cambiano insieme da un'iterazione all'altra).

## 7.3 Process mining

### 7.3.1 Log di Eventi

I log di eventi sono la base del *Process Mining*. Questi log registrano le attività svolte all'interno di un processo, catturando informazioni come:

- Nome dell'attività.
- Timestamp (data e ora di inizio, e talvolta di fine).
- Stato dell'attività (avvio, completamento, sospensione, ecc.).
- Risorse coinvolte (personale, sistema, responsabile, ecc.).
- Informazioni aggiuntive utili per l'analisi (ad esempio, il cliente interessato).

Un esempio di log di eventi è mostrato nella Figura 7.14 del testo, dove viene registrata anche l'informazione relativa al cliente.

### 7.3.2 Process Discovery

Il *Process Discovery* si occupa di generare un modello di processo a partire dai log di eventi, senza ricorrere a informazioni preesistenti. Gli algoritmi di Process Mining analizzano le tracce (sequenze di attività per una specifica istanza del processo) presenti nei log, identificando:

- Attività.
- Relazioni tra attività.
- Possibili diramazioni.

Le relazioni tra attività possono essere di diverso tipo: successione diretta, causalità, parallelismo e scelta. Queste relazioni possono essere rappresentate tramite i gateway della notazione BPMN, come mostrato nella Figura 7.15.

### 7.3.3 Controllo di conformità

Il controllo di conformità ha l'obiettivo di verificare se l'esecuzione reale di un processo rispetta il modello definito. Il Process Mining aiuta a identificare eventuali deviazioni, come:

- Esecuzione di attività non previste.
- Esecuzione di attività in un ordine diverso da quello definito.

Un esempio di dashboard di analisi di un processo è mostrato nella Figura 7.16, realizzata con il tool Apromore. La dashboard evidenzia la frequenza e i tempi di esecuzione delle attività, oltre a metriche generali sul processo.

#### **7.3.4 Miglioramento dei processi**

Il Process Mining consente di migliorare un modello di processo esistente sfruttando le informazioni contenute nei log di eventi. L'analisi permette di individuare:

- Inefficienze.
- Colli di bottiglia.
- Aree di miglioramento.

# Capitolo 8

## 8 Tecnologie a livello di piattaforma

### 8.1 Architetture di integrazione

Questa sezione evidenzia l'importanza di collegare i diversi moduli applicativi in un sistema informativo per garantire un flusso di informazioni efficiente e completo. I sistemi informativi moderni sono spesso costituiti da una molteplicità di applicazioni, sviluppate in momenti diversi e con tecnologie diverse, che necessitano di comunicare tra loro per supportare i processi aziendali.

- L'integrazione è un aspetto critico per la crescita e evoluzione dei sistemi informativi.
- Spesso le organizzazioni si confrontano con la problematica delle applicazioni a silos: applicazioni che non dialogano tra loro, creando inefficienze, ridondanze e ostacolando la collaborazione tra le diverse aree aziendali.
- L'integrazione dei sistemi informativi porta numerosi benefici:
  - Migliore gestione dei dati.
  - Eliminazione delle ridondanze.
  - Maggiore efficienza.
  - Migliore collaborazione tra le funzioni aziendali.

### 8.2 Livelli applicativi e tecniche di integrazione

Diverse tecniche di integrazione possono essere applicate ai vari livelli di un'applicazione.

- **Integrazione a livello dati:** questa tecnica prevede la creazione di un'unica sorgente di dati condivisa, accessibile da tutte le applicazioni.
- **Integrazione a livello di applicazione:** questo approccio si basa sullo scambio di messaggi tra le applicazioni. Le applicazioni possono comunicare tra loro utilizzando protocolli standard o API.
- **Integrazione a livello di presentazione:** questo metodo unifica l'interfaccia utente per diverse applicazioni. Gli utenti possono accedere a diverse funzionalità da un'unica interfaccia, semplificando l'utilizzo del sistema.

### 8.3 Modelli di integrazione delle applicazioni

Il testo presenta una serie di modelli di integrazione frequentemente adottati, ognuno con i suoi vantaggi e svantaggi:



### 8.3.1 Architettura di integrazione punto-a-punto

In questo modello, le applicazioni si connettono direttamente tra loro.

- *Vantaggio*: Semplicità di implementazione per sistemi con poche applicazioni.
- *Svantaggio*: Non scalabile per sistemi complessi con molte applicazioni, in quanto il numero di connessioni cresce esponenzialmente con il numero di applicazioni.

### 8.3.2 Architettura di integrazione hub-and-spoke

In questo modello, un hub centrale funge da intermediario per la comunicazione tra le applicazioni.

- *Vantaggio*: Maggiore scalabilità rispetto al modello punto-a-punto, semplificando la gestione delle connessioni.
- *Svantaggio*: L'hub può diventare un punto critico di guasto, compromettendo l'intero sistema in caso di malfunzionamenti.

### 8.3.3 Integrazione con Business Process Management System

Un BPMS coordina l'interazione tra le applicazioni in base a processi aziendali formalizzati.

- *Vantaggio*: Permette di modellare e automatizzare i processi aziendali, migliorando l'efficienza e la visibilità dei processi stessi.
- *Svantaggio*: Richiede un'attenta analisi e formalizzazione dei processi aziendali, oltre all'adozione di un BPMS che può essere complesso da implementare e gestire.

### 8.3.4 Modelli a servizi

Un ESB funge da bus di comunicazione centrale per la gestione dei messaggi tra le applicazioni.

- *Vantaggio*: Offre un'infrastruttura di comunicazione flessibile e scalabile, semplificando l'integrazione di diverse applicazioni.
- *Svantaggio*: Può diventare complesso da gestire in sistemi di grandi dimensioni.

### 8.3.5 Composizione e orchestrazione di servizi

Questo modello si basa sul paradigma Service-Oriented Architecture (SOA), in cui le applicazioni espongono le proprie funzionalità come servizi.

- *Vantaggio*: I servizi possono essere riutilizzati e combinati per creare nuove applicazioni e processi aziendali.
- *Svantaggio*: Richiede una progettazione attenta dei servizi e può essere complesso da implementare e gestire.

# Capitolo 9

## 9 Tecnologie a livello di architettura fisica

### 9.1 Descrizione

Questo capitolo si concentra sull'organizzazione dell'hardware e del software per la gestione delle informazioni, analizzando le diverse architetture dei sistemi informativi.

#### 9.1.1 Evoluzione delle architetture fisiche

Il termine *architettura* in ambito IT si riferisce alle scelte tecniche e organizzative che influenzano lo sviluppo e l'utilizzo delle risorse tecnologiche. Esistono due macro-categorie di architetture a livello fisico: **centralizzate** e **distribuite**.

#### ARCHITETTURE CENTRALIZZATE

- Prevedono un unico nodo di elaborazione (mainframe) su cui risiede tutto il software e i dati.
- I terminali degli utenti si collegano al mainframe per accedere alle risorse.
- Erano la soluzione tipica dei primi sistemi informativi a causa del costo elevato di hardware e reti.

#### ARCHITETTURE DISTRIBUITE

- Si sono diffuse con l'avvento di tecnologie hardware e software più economiche e flessibili.
- *Caratteristiche:*
  - Le applicazioni sono suddivise in moduli eseguiti su nodi diversi (elaborazione distribuita).
  - I dati sono distribuiti su più nodi di elaborazione (base di dati distribuita).
- *Vantaggi:*
  - Maggiore flessibilità.
  - Migliore scalabilità.
  - Maggiore tolleranza ai guasti.

### 9.2 Applicazioni distribuite

Le architetture distribuite si distinguono in:

- **Single-Tiered:** un unico nodo gestisce tutti i componenti.

- **Two-Tiered:** due livelli di distribuzione, tipiche dei primi sistemi client/server.
  - *Figura 9.3:* Illustra le sei diverse configurazioni possibili.
- **Three-Tiered:** tre livelli di distribuzione (presentazione, logica applicativa e gestione dati).
  - *Figura 9.4:* Mostra le tredici configurazioni possibili.
  - *Esempio:* La Figura 9.5 mostra un'architettura Three-Tiered.
- **N-Tiered:** i livelli applicativi sono distribuiti su un numero  $n$  di tier.
  - *Esempio:* La Figura 9.7 mostra un sistema 4-tier.

## 9.3 Scalabilità

La scalabilità di un'architettura IT si riferisce alla sua capacità di adattarsi a richieste crescenti. Esistono due tipi di scalabilità:

- **Verticale (scale-up):** si aumenta la capacità elaborativa dei nodi esistenti.
  - *Vantaggi:* Semplicità di implementazione.
  - *Svantaggi:* Limiti fisici nell'espansione, aumento delle prestazioni non sempre proporzionale all'investimento.
- **Orizzontale (scale-out):** si aggiungono nuovi nodi al sistema.
  - *Vantaggi:* Scalabilità a costi contenuti grazie al principio del downsizing.
  - *Svantaggi:* Maggiore complessità di gestione, necessità di un sistema di load balancing.

Il dimensionamento corretto di un sistema è cruciale per evitare sprechi di risorse (sovradimensionamento) o prestazioni insufficienti (sottodimensionamento).

### 9.3.1 Server Farm

Seguendo l'approccio *scale-out*, i tier fisici possono essere realizzati come **server farm**, un insieme di elaboratori che condividono il carico e le applicazioni, visibile come una singola risorsa. Le server farm sono molto scalabili, permettendo l'aggiunta di macchine per soddisfare richieste crescenti a costi contenuti. Esse possono essere progettate seguendo due principi:

- **Clonazione (Cloning):** installazione delle stesse applicazioni e dati su ogni server. Le richieste sono distribuite tramite un sistema di *load balancing*. Un sistema di cloni dedicato a un servizio si definisce **RACS** (Reliable Array of Cloned Services), offrendo sia *scalabilità* sia *fault tolerance*. Esistono due configurazioni:
  - **Shared-nothing:** ogni clone ha il proprio storage locale, rendendo complicata la gestione di servizi “write-intensive” ma adatta per applicazioni “read-only” (vedi Figura 9.8).

- **Shared-disk:** i cloni condividono un unico storage (cluster), garantendo disponibilità senza ridondanza di dati (vedi Figura 9.9).
- **Partizionamento (Partitioning):** suddivisione delle applicazioni e dei dati tra nodi specializzati (vedi Figura 9.10). Le richieste sono inviate alla partizione rilevante, come nel caso di un sistema Web aziendale segmentato per prodotto. Il partizionamento presenta la proprietà di *graceful degradation*: in caso di guasto, solo alcune funzionalità sono inaccessibili. Utilizzando la clonazione all'interno delle partizioni, si ottiene una struttura detta **RAPS** (Reliable Array of Partitioned Services), che garantisce sia scalabilità che disponibilità (vedi Figura 9.11).

Un esempio di architettura *five-tiered* in server farm per un sistema informativo Web è presentato in Figura 9.12. Qui, i web server e gli script engine sono clonati, mentre l'application server può essere partizionato o partizionato e clonato.

## 9.4 Virtualizzazione

### Concetto di Virtualizzazione:

- **Virtualizzazione:** Consente di creare risorse virtuali come CPU, memoria e dischi, indipendenti dall'hardware fisico.
- **Macchina Virtuale (VM):** Include tutte le risorse virtuali necessarie per far funzionare un'applicazione. Può eseguire sistemi operativi guest diversi dall'host.
- **Hypervisor:** Uno strato software che gestisce la corrispondenza tra risorse fisiche e virtuali. Esistono due tipi principali:
  - **Bare-metal:** Installato direttamente sull'hardware fisico.
  - **Hosted:** Funziona sopra un sistema operativo host.

### Vantaggi della Virtualizzazione:

1. **Disaccoppiamento delle risorse:** L'hardware fisico può essere modificato senza influenzare le applicazioni virtuali.
2. **Frazionamento e condivisione:** Una singola risorsa fisica può essere suddivisa in più risorse virtuali o condivisa tra più applicazioni.
3. **Ottimizzazione delle risorse:** Migliora l'efficienza sfruttando al meglio le capacità dei server.

### Applicazioni nella gestione di server:

- La virtualizzazione permette politiche di *server consolidation*, riducendo il numero di macchine fisiche necessarie.
- Contribuisce a ridurre gli sprechi di risorse quando il carico delle applicazioni è inferiore alla capacità disponibile.

### Migrazione di Macchine Virtuali (VM):

- **Migrazione a freddo:** La VM viene spenta, trasferita e riaccesa.
- **Migrazione a caldo:** La VM viene trasferita durante l'esecuzione, minimizzando i tempi di inattività.

**Conclusione:** La virtualizzazione è stata fondamentale per migliorare la gestione delle risorse nei data center e per l'evoluzione verso il *cloud computing*, che viene approfondito nella sezione successiva.

## 9.5 Cloud Computing

Il **cloud computing** è un paradigma per l'architettura delle applicazioni, basato sull'accesso on-demand a risorse condivise (reti, server, storage, applicazioni, servizi), utilizzando infrastrutture virtualizzate. Consente l'accesso alle risorse via Internet da qualsiasi dispositivo, ottimizzando costi e scalabilità.

### 9.5.1 Caratteristiche principali

Il cloud computing si basa su cinque caratteristiche fondamentali:

1. **On-demand self-service:** Gli utenti possono accedere alle risorse in autonomia, senza intervento manuale del provider.
2. **Broad network access:** Le risorse sono accessibili tramite rete e compatibili con vari dispositivi (PC, smartphone, tablet).
3. **Resource pooling:** Le risorse fisiche e virtuali vengono condivise in modo dinamico tra gli utenti.
4. **Rapid elasticity:** Capacità di scalare le risorse rapidamente in base alle esigenze, garantendo elasticità.
5. **Measured service:** Utilizzo delle risorse misurato e controllato, spesso con modelli di pagamento a consumo.

### 9.5.2 Modelli di servizio

Esistono tre principali modelli di servizio:

1. **Infrastructure as a Service (IaaS):** Fornisce risorse virtuali (server, storage, rete) su cui gli utenti possono installare e configurare software.
2. **Platform as a Service (PaaS):** Fornisce piattaforme integrate per lo sviluppo e l'esecuzione di applicazioni, senza gestire l'infrastruttura sottostante.
3. **Software as a Service (SaaS):** Consente agli utenti di accedere a software e applicazioni già configurate tramite Internet.

### 9.5.3 Modelli di deployment

Sono disponibili quattro modelli di distribuzione:

1. **Cloud pubblico:** Le risorse sono gestite da provider esterni e accessibili a più utenti.
2. **Cloud privato:** Utilizzato da un'unica organizzazione, garantendo maggior controllo e sicurezza.
3. **Cloud ibrido:** Combina cloud pubblico e privato, sfruttandone i vantaggi in modo complementare.
4. **Cloud comunitario:** Condiviso da un gruppo di organizzazioni con esigenze simili, come enti governativi o aziende dello stesso settore.

**Conclusione:** Il cloud computing, grazie a modelli flessibili e a un'efficiente gestione delle risorse, è diventato una soluzione chiave per aziende e organizzazioni, migliorando scalabilità, flessibilità e accessibilità, pur richiedendo attenzione a sicurezza e privacy.

# Capitolo 10

## 10 Sicurezza dei sistemi informativi

### 10.1 Proprietà di sicurezza

Le proprietà fondamentali della sicurezza nei sistemi informativi sono:

**1. Integrità:**

- Garantisce che i dati non siano alterati in modo non autorizzato.
- Protegge da modifiche intenzionali, accidentali o derivanti da errori di sistema.
- Comprende anche la verifica della correttezza dei dati durante il trasferimento.

**2. Autenticità:**

- Permette di identificare l'origine delle informazioni.
- Fornisce prove che i dati ricevuti provengano dalla fonte dichiarata.
- Utilizza strumenti come certificati digitali e protocolli di autenticazione.

**3. Riservatezza:**

- Evita che persone non autorizzate possano accedere alle informazioni.
- Include la protezione di dati sensibili e delle comunicazioni tramite crittografia.
- Spesso richiede strumenti di controllo degli accessi.

**4. Disponibilità:**

- Assicura che le risorse siano utilizzabili da utenti autorizzati nei momenti richiesti.
- Include la protezione contro attacchi di tipo DoS (Denial of Service).
- Garantisce il funzionamento continuo tramite ridondanza e soluzioni di disaster recovery.

### 10.2 Minacce, violazioni, vulnerabilità, attacchi

Esploriamo i rischi e le problematiche che compromettono la sicurezza dei sistemi informativi, analizzando le tipologie di minacce, le vulnerabilità che possono essere sfruttate e i tipi di attacchi.

#### TIPOLOGIE DI MINACCE

**• Fisiche:**

- Furti, sabotaggi, danni intenzionali.
- Eventi naturali come incendi, alluvioni o terremoti che compromettono i sistemi.

- **Logiche:**

- Exploit di vulnerabilità software, accessi non autorizzati.
- Malware, backdoor o utilizzo illecito di risorse di rete.

- **Accidentali:**

- Errori umani come configurazioni errate o cancellazioni accidentali.
- Malfunzionamenti hardware o software.

## **VULNERABILITÀ**

- Le vulnerabilità rappresentano debolezze nei sistemi che possono essere sfruttate dagli attaccanti.

- **Esempi comuni:**

- **Password deboli:** Facili da indovinare o scoprire con attacchi di forza bruta.
- **Software non aggiornato:** Spesso contiene bug o falle di sicurezza.
- **Configurazioni errate:** Impostazioni non sicure nei sistemi o nelle reti.
- **Mancanza di firewall o antivirus:** Lascia i sistemi esposti.

## **TIPI DI ATTACCHI**

- **Phishing:**

- Tecnica di inganno per sottrarre informazioni sensibili come credenziali e dati bancari.
- Solitamente veicolata tramite email o messaggi falsi.

- **Malware:**

- Software malevolo come virus, worm, trojan e ransomware.
- Può danneggiare i sistemi o cifrare i dati per richiedere riscatti.

- **Sniffing:**

- Intercettazione di dati trasmessi in rete non protetta.
- Permette di catturare password, messaggi e altre informazioni sensibili.

- **Spoofing:**

- Tecnica per falsificare l'identità (ad esempio indirizzi IP o email) per ingannare utenti o sistemi.

- **DoS e DDoS:**

- Attacchi Denial of Service e Distributed Denial of Service che sovraccaricano le risorse di un sistema, rendendolo inaccessibile.



## 10.3 Crittografia per la protezione dei dati

Crittografare significa codificare l'informazione, trasformando il messaggio originale in un messaggio cifrato. I meccanismi di crittografia sono composti da un algoritmo (funzione crittografica) e da una chiave (o più chiavi). La sicurezza dipende dalla lunghezza e dalla segretezza della chiave, che deve essere cambiata frequentemente. Esistono due classi di algoritmi:

- **Algoritmi a chiave simmetrica:** Utilizzano una chiave segreta condivisa tra mittente e destinatario, come DES e AES.
- **Algoritmi a chiave asimmetrica:** Utilizzano una coppia di chiavi, pubblica e privata, come RSA, Diffie-Hellman e El-Gamal.

### 10.3.1 Crittografia simmetrica

La crittografia simmetrica utilizza una chiave segreta condivisa ( $K$ ) tra il mittente e il destinatario. La chiave deve essere cambiata frequentemente per garantire la sicurezza. Le operazioni principali sono:

- **Sostituzione:** Ogni carattere del testo in chiaro viene sostituito con un altro.
- **Trasposizione:** Le lettere del testo in chiaro vengono permutate secondo una chiave.

Esempi di algoritmi di crittografia simmetrica includono il **Cifrario di Cesare** e l'**AES**. L'AES è uno degli algoritmi più sicuri e utilizza chiavi di lunghezza 128, 192 o 256 bit. La crittografia simmetrica è veloce, ma la gestione della chiave è un problema critico.

### 10.3.2 Crittografia asimmetrica

La crittografia asimmetrica utilizza due chiavi correlate: una **chiave pubblica** ( $K_P$ ) e una **chiave privata** ( $K_{Pr}$ ). La chiave pubblica può essere distribuita pubblicamente, mentre la chiave privata deve rimanere segreta. Il meccanismo consente di garantire la riservatezza e l'autenticità dei messaggi. Gli algoritmi più noti sono **RSA**, **Diffie-Hellman**, e **El-Gamal**.

Un esempio di utilizzo pratico è l'uso della crittografia asimmetrica per lo **scambio sicuro di chiavi** per la crittografia simmetrica. Un altro uso è la cifratura di un messaggio con la chiave pubblica di  $B$ , in modo che solo  $B$ , che possiede la chiave privata corrispondente, possa decifrarlo.

### 10.3.3 Integrità e funzione di hash

Le **funzioni di hash** sono utilizzate per garantire l'integrità dei dati. Una funzione di hash trasforma un messaggio di lunghezza arbitraria in un output di lunghezza fissa chiamato **digest** o **impronta digitale**. Le funzioni di hash devono essere:

- **Deterministiche:** La stessa input deve produrre lo stesso output.
- **Uniche:** La probabilità che due messaggi diversi abbiano lo stesso digest deve essere quasi nulla.

- **Non invertibili:** Non è possibile risalire al messaggio originale a partire dal digest.

Le funzioni di hash sono fondamentali per verificare che i dati non siano stati alterati durante la trasmissione. Un esempio di funzione di hash è **SHA-256**.

#### 10.3.4 Firma digitale

La **firma digitale** è un metodo per garantire l'autenticità e l'integrità di un messaggio. La firma digitale è il digest cifrato di un documento, ottenuto usando la chiave privata del mittente. Il destinatario può verificare la firma usando la chiave pubblica del mittente. Questo processo garantisce che il mittente sia chi dice di essere e che il messaggio non sia stato alterato.

Le firme digitali vengono utilizzate in protocolli di sicurezza come **SSL**, **TLS**, e **S/MIME**. La firma digitale è più sicura di una firma autografa e non può essere facilmente falsificata.

#### 10.3.5 Aspetti di gestione delle chiavi e certificati digitali

La **gestione delle chiavi** è un aspetto fondamentale della crittografia. Le chiavi devono essere generate, distribuite e conservate in modo sicuro. Le **Infrastrutture di Chiavi Pubbliche** (PKI) gestiscono la distribuzione e la revoca dei certificati digitali.

Un **certificato digitale** è un documento che associa una chiave pubblica a un'entità e viene firmato da un'Autorità di Certificazione (CA). I certificati digitali vengono utilizzati per garantire che una chiave pubblica appartenga effettivamente al possessore dichiarato. La gestione delle chiavi pubbliche è realizzata tramite sistemi come la PKI, che fornisce servizi come l'emissione di certificati, la revoca e la validazione delle firme digitali.

### 10.4 Gestione degli utenti e controllo degli accessi

#### 10.4.1 Autenticazione e autorizzazione

Prima di ogni controllo di sicurezza, è necessario **autenticare** gli utenti per garantire che solo gli agenti autorizzati possano accedere a risorse protette. L'autenticazione è il processo attraverso il quale un sistema verifica l'identità di un utente, richiedendo credenziali specifiche, come una password, un oggetto fisico o caratteristiche biometriche.

Gli agenti possono essere identificati utilizzando uno o più dei seguenti fattori:

- **Something You Know (SYK):** Un'informazione conosciuta dall'utente (es. password).
- **Something You Have (SYH):** Un oggetto posseduto dall'utente (es. una carta magnetica).
- **Something You Are (SYA):** Una caratteristica fisica dell'utente (es. impronta digitale, retina).

L'autenticazione può essere **semplice** (userid e password) o **robusta** (One-Time Password, sistemi challenge-response, dispositivi personali come smart card, o biometria).

L'autenticazione robusta è più sicura rispetto a quella semplice, poiché non trasmette mai direttamente la password, ma la utilizza solo per generare un valore che conferma la conoscenza della stessa.

#### 10.4.2 Autenticazione a due fasi

L'**autenticazione a più fattori** (MFA) è una misura di sicurezza che richiede due o più elementi per verificare l'identità dell'utente, riducendo il rischio di attacchi basati su password. Poiché i metodi di autenticazione tradizionali (come la password) sono vulnerabili ad attacchi come **smudge attacks** (estrazione di password da impronte su dispositivi touchscreen), **brute force** e **shoulder surfing** (osservazione diretta delle vittime), l'MFA è diventata una soluzione più sicura.

L'autenticazione a **due fattori** (2FA) è la più comune e richiede almeno due dei seguenti fattori:

- **Something You Know (SYK)**: Es. password.
- **Something You Have (SYH)**: Es. un codice inviato tramite SMS o generato da un'app di autenticazione.
- **Something You Are (SYA)**: Es. impronta digitale o riconoscimento facciale.

La 2FA è sicura perché il secondo fattore (come il codice OTP) viene generato in modo pseudocasuale, con una durata limitata (solitamente 30 secondi). Questo rende l'OTP difficile da intercettare o riprodurre.

Un esempio di 2FA è l'**SPID** (Sistema Pubblico di Identità Digitale) che utilizza la combinazione di password, un'app (come PosteID) e un PIN. In alcuni casi, l'autenticazione può arrivare fino a **tre fattori** (3FA), ma la 2FA è quella più utilizzata, consigliata da molti servizi online come Amazon, Google, PayPal e Facebook.

L'uso della 2FA, pur essendo opzionale nella maggior parte dei casi, è altamente raccomandato per garantire la sicurezza dell'account.

#### 10.4.3 Sistemi di controllo di accesso ai dati

La **funzione di hash** è un componente essenziale per garantire l'**integrità** dei dati, assicurando che le informazioni non siano state alterate in modo non autorizzato. Le funzioni di hash trasformano un messaggio di lunghezza variabile in un valore di lunghezza fissa chiamato **digest** o **impronta digitale** del messaggio. Questo valore serve come una rappresentazione unica e compatta dei dati originali.

#### FUNZIONE DI HASH

- Una funzione di hash è un algoritmo che prende un input (di qualsiasi dimensione) e produce un output di lunghezza fissa. L'output è il "digest", che rappresenta il contenuto originale del messaggio.

- Le funzioni di hash sono utilizzate per garantire che i dati non siano stati alterati durante il trasferimento. Se anche un singolo bit del messaggio originale cambia, il digest risultante cambierà drasticamente, rendendo facile rilevare qualsiasi modifica.

## UTILIZZO DELLE FUNZIONI DI HASH

- **Verifica dell'integrità:** Un'applicazione o un sistema può calcolare il digest di un file o di un messaggio e confrontarlo con un valore precedentemente memorizzato. Se i due digest corrispondono, i dati sono intatti; se sono diversi, è indicato che i dati sono stati modificati.
- **Indirizzamento unico dei dati:** Le funzioni di hash sono anche utilizzate per generare identificatori unici per i dati, come nei sistemi di gestione dei file o nei blockchain.

## ALGORITMI DI HASH COMUNI

- **SHA-256 (Secure Hash Algorithm 256-bit):** Una funzione di hash sicura, ampiamente utilizzata nei protocolli di sicurezza e nelle criptovalute, che produce un digest di 256 bit.
- **MD5 (Message Digest Algorithm 5):** Una funzione di hash obsoleta e vulnerabile a collisioni, che non è più sicura per la protezione dei dati sensibili.

## SICUREZZA DELLE FUNZIONI DI HASH

Una funzione di hash sicura deve avere alcune proprietà, tra cui:

- **Determinismo:** La stessa input deve produrre sempre lo stesso output.
- **Efficienza:** Deve essere rapida da calcolare.
- **Resistenza alle collisioni:** Due input distinti non devono mai produrre lo stesso digest.
- **Preimage resistance:** Non deve essere possibile risalire dall'output al dato originale.
- **Second preimage resistance:** Non deve essere possibile trovare un secondo input che generi lo stesso digest.

## 10.5 Zero Trust

Di fronte a cyber attacchi basati su metodologie in continua evoluzione, come quelli basati sull'intelligenza artificiale, è possibile adottare un approccio Zero Trust (ZT), che consente di gestire lo smart working, accentuato dalla pandemia di Covid-19. Il modello ZT si basa sul principio che l'accesso a sistemi e applicazioni deve essere consentito solo se

strettamente necessario e senza fiducia implicita. Ogni accesso deve essere continuamente verificato, indipendentemente da dove provenga.

L'approccio ZT è stato sviluppato nel 2010 per proteggere le risorse aziendali in ambienti cloud, distribuiti e remoti. Con il lavoro e le risorse sempre più virtualizzati, il concetto tradizionale di "perimetro di rete" è obsoleto. Oggi, gli utenti e le applicazioni possono trovarsi anche al di fuori di quello che una volta era considerato il perimetro protetto, rendendo necessaria un'ulteriore protezione.

### 10.5.1 Il contesto

Nel *The Global Risk Report 2024* del World Economic Forum, la cyber security è identificata come uno dei principali fattori di rischio, insieme ai cambiamenti climatici e alle tensioni geopolitiche. Gli attacchi informatici sono evoluti: le aziende criminali sono ora strutturate con risorse, capacità di sviluppare prodotti e team dedicati, come qualsiasi altro business.

L'emergenza pandemica ha reso evidente la necessità di adottare modelli di sicurezza come ZT, in particolare per affrontare le vulnerabilità del lavoro remoto. Le soluzioni tradizionali come le **VPN** si sono rivelate rischiose, poiché si basano su un modello di fiducia implicita riguardo chi accede alla rete. Inoltre, la sicurezza degli endpoint è essenziale per proteggere da minacce provenienti da dispositivi che non sono sempre aggiornati o controllati.

### 10.5.2 L'approccio

Il modello Zero Trust (ZT) si basa su tre principi fondamentali:

1. **Tutte le entità sono considerate non attendibili.**
2. **L'accesso è basato sul privilegio minimo.**
3. **Il sistema effettua un monitoraggio costante della sicurezza.**

Il sistema di sicurezza verifica l'identità di utenti, dispositivi e applicazioni tramite metodi di autenticazione e autorizzazione, sia che si trovino all'interno che all'esterno della rete. Ogni accesso o trasferimento di dati viene costantemente monitorato.

L'architettura ZT, conosciuta anche come "protezione senza perimetro", si compone di vari principi e componenti integrati, tra cui:

- **Visibilità e controllo degli ambienti** on-premise/cloud e dei dispositivi IoT.
- **Controllo dei flussi di rete** tra tutte le risorse.
- **Verifica continua dell'identità e accesso al cloud.**
- **Segmentazione della rete e delle applicazioni.**
- **Autenticazione multifattore (MFA).**

- **Policy di accesso granulari**, come il controllo dell'accesso a singoli record nel database.
- **Minimizzazione dell'uso di VPN e firewall.**

Inoltre, il modello ZT prevede l'uso di tecniche di *deception*, simili a quelle utilizzate nello spionaggio e controspionaggio, per individuare precocemente le minacce e prevenire attacchi.

Le principali implementazioni di Zero Trust includono:

- **Zero Trust Architecture**
- **Zero Trust Network Access (ZTNA)**
- **Zero Trust Secure Web Gateway (SWG)**

In sintesi, il modello Zero Trust offre una soluzione avanzata per proteggere le risorse aziendali contro minacce sempre più sofisticate, particolarmente in un contesto di lavoro remoto e applicazioni cloud.

## 10.6 Meccanismi di sicurezza infrastrutturali

I **meccanismi di sicurezza infrastrutturali** sono strumenti progettati per proteggere i sistemi informativi da attacchi a livello di architettura fisica. Le misure includono la configurazione sicura di sistemi operativi, client e server, l'implementazione di politiche di autenticazione, controllo degli accessi, e la registrazione delle attività tramite log. Le principali componenti architetturali sono i **firewall**, i **proxy server** e i **sistemi di Intrusion Detection (IDS)**.

### 10.6.1 Firewall

Il **firewall** è una componente hardware o software che protegge una rete sicura da reti insicure, filtrando il traffico in entrata e in uscita. Le sue funzioni principali includono:

- Prevenire, rilevare e annullare gli attacchi.
- Implementare politiche di sicurezza per controllare l'accesso alla rete.

Le principali configurazioni di firewall includono:

- **Screening router:** Blocca i pacchetti in transito basandosi sull'intestazione o sul contenuto.
- **Application gateway:** Fornisce un controllo più dettagliato del traffico, funzionando come un proxy che intermedia le comunicazioni tra client e server.

Il firewall deve seguire alcune leggi fondamentali:

- Deve essere l'unico punto di contatto tra la rete interna e l'esterno.
- Tutti i pacchetti devono essere bloccati, tranne quelli autorizzati.
- Deve essere sicuro anche dal punto di vista delle sue configurazioni.

### 10.6.2 Intrusion Detection Systems - IDS

Un **Intrusion Detection System (IDS)** monitora gli eventi di un sistema o di una rete per individuare intrusioni, che sono tentativi di compromettere la sicurezza (confidenzialità, integrità, disponibilità). Gli IDS sono in grado di rilevare e rispondere a tentativi di attacco, potendo lavorare in due modi:

- **Detection:** Individuare intrusioni da parte di utenti non autorizzati o da attacchi interni (insider threat).
- **Prevention:** Interrompere o bloccare attacchi in corso.

Gli IDS si suddividono in:

- **Network-based IDS:** Monitorano il traffico di rete per rilevare attacchi, come pacchetti malformati o sospetti. Sono vantaggiosi per monitorare ampie reti, ma non riescono a rilevare attacchi cifrati.
- **Host-based IDS:** Installati su singoli dispositivi, rilevano attacchi a livello di sistema operativo o applicazione, con maggiore precisione, ma sono più complessi da gestire.

Esistono due tipi principali di analisi degli IDS:

- **Misuse detection:** Rileva attacchi noti confrontando il comportamento del sistema con modelli predefiniti (signature-based).
- **Anomaly detection:** Rileva comportamenti anomali confrontando l'attività corrente con un profilo di comportamento normale (behavior-based).

Gli IDS possono rispondere a un attacco in due modi:

- **Active response:** Prendere misure immediate, come bloccare il traffico o resettare le connessioni.
- **Passive response:** Notificare agli amministratori dell'attacco in corso, lasciando a loro la decisione su come procedere.

## 10.7 Evoluzione delle Minacce e delle Tecniche di Sicurezza

L'evoluzione delle minacce alla sicurezza informatica è strettamente legata alla crescita del **cyberspazio**, l'ambiente in cui si svolgono operazioni su Internet. La diminuzione dei costi di accesso alla rete e lo sviluppo della banda larga hanno portato a una continua espansione di Internet, rendendo i sistemi informatici più vulnerabili agli attacchi da parte di criminali, hacker e terroristi. Questi attacchi possono essere globali e avvenire quasi istantaneamente, con impatti devastanti come la frode e il furto di segreti commerciali.

### 10.7.1 Evoluzione delle Minacce

Le minacce informatiche stanno evolvendo a causa di vari fattori, tra cui:

- La maggiore pervasività dei dispositivi.
- La riduzione dei costi per compiere attacchi informatici.
- La diminuzione della competenza necessaria per effettuare attacchi complessi.

Le piccole e medie imprese sono particolarmente vulnerabili, dovendo affrontare costi più alti per la protezione e difficoltà maggiori nel rispondere ai danni economici e reputazionali causati da attacchi. Esiste una **asimmetria informativa**, dove le aziende più piccole sono svantaggiate rispetto agli aggressori.

### 10.7.2 Evoluzione della Crittografia

Gli algoritmi crittografici si stanno adattando alle nuove minacce. L'algoritmo **RSA**, che è alla base della crittografia asimmetrica, sta incontrando difficoltà a causa dei progressi nell'informatica quantistica. Ad esempio, l'algoritmo di **Shor** permette di fattorizzare numeri grandi in modo molto più efficiente, minacciando la sicurezza di RSA.

A causa di ciò, molte organizzazioni stanno migrando verso crittografia **post-quantistica** (PQC), che è progettata per resistere agli attacchi da parte dei computer quantistici. Alcuni algoritmi simmetrici e funzioni di hash sono ancora sicuri contro i computer quantistici. Per garantire la sicurezza in un'era quantistica, si prevede che la lunghezza delle chiavi nei cifrari simmetrici debba essere raddoppiata.

Alcuni algoritmi candidati per la crittografia post-quantistica includono:

- Crittografia basata su **reticolo**.
- Crittografia basata su **hash**.
- Crittografia basata su **codice**.
- Crittografia **polinomiale multivariata**.

La standardizzazione della crittografia post-quantistica è in corso, con l'obiettivo di proteggere le informazioni dai rischi legati ai progressi dell'informatica quantistica, mantenendo la compatibilità con le infrastrutture esistenti.



# Capitolo 11

## 11 La progettazione di un SI

### 11.1 Il processo di gestione del sistema informativo

Un progetto è un insieme di attività pianificate per raggiungere un obiettivo specifico entro tempi, costi e risorse definiti. I sistemi informativi richiedono una gestione rigorosa per affrontare la complessità tecnica, organizzativa ed economica.

#### CARATTERISTICHE DEI PROGETTI DI SVILUPPO

- **Temporalità:** ogni progetto ha un inizio e una fine definiti.
- **Unicità:** ogni progetto è diverso, pur avendo obbiettivi comuni con altri progetti.
- **Risorse limitate:** budget, personale e infrastrutture sono vincolati.
- **Incidenza dei rischi:** l'incertezza è elevata, richiedendo piani di gestione specifici.

#### RUOLO DELLE ORGANIZZAZIONI NEI PROGETTI

- La **sponsorizzazione** da parte della dirigenza è cruciale per il successo del progetto.
- Le organizzazioni devono bilanciare innovazione tecnologica e adattamento ai cambiamenti del mercato.

### 11.2 Pianificazione

La pianificazione mira a predisporre una proposta progettuale chiara e dettagliata, approvabile dal committente. Questa fase è essenziale per garantire che il progetto sia allineato con gli obiettivi organizzativi e strategici.

#### RUOLI COINVOLTI

- **Committente:** approva e finanzia il progetto.
- **Team interno o consulenti esterni:** analizzano opportunità e sviluppano il piano.
- **Stakeholder:** rappresentano le parti interessate, come azionisti, operatori e clienti.

#### ATTIVITÀ PRINCIPALI

La pianificazione si articola in due attività principali:

1. Pianificazione Strategica.
2. Studio di Fattibilità.

### 11.2.1 Pianificazione strategica

La pianificazione strategica si occupa di identificare obiettivi e opportunità in ambito IT, con particolare attenzione all'allineamento tra strategia aziendale e sistemi informativi.

#### ATTIVITÀ PRINCIPALI

##### 1. Analisi delle Opportunità:

- Esaminare processi, tecnologie e strategie per individuare potenziali sviluppi del sistema IT.

##### 2. Analisi dei Fabbisogni Informativi:

- Modello delle informazioni necessarie per supportare l'organizzazione.
- Definizione degli scambi informativi chiave.

#### INDICATORI E BUSINESS INFORMATIVI

- **Efficacia:** Misura il raggiungimento degli obiettivi strategici. La formula è:

$$\text{Efficacia} = \frac{\text{Output Effettivo}}{\text{Output Atteso}}$$

Esempi:

- Incrementare la quota di mercato.
- Ridurre il tempo per sviluppare un prodotto.

- **Efficienza:** Valuta l'ottimizzazione delle risorse. La formula è:

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

Esempi:

- Riduzione dei costi.
- Miglioramento della produttività.

### 11.2.2 Studio di fattibilità

Lo studio di fattibilità si concentra sull'analisi delle soluzioni progettuali alternative, considerando vincoli e benefici.

#### FASI DELLO STUDIO DI FATTIBILITÀ

##### 1. Definizione degli Obiettivi e Specifiche Funzionali:

- Identificare le aree organizzative da migliorare.
- Definire requisiti funzionali (funzionalità richieste) e non funzionali (prestazioni, vincoli tecnici).

## 2. Progettazione delle Soluzioni:

- Identificazione di possibili soluzioni IT.
- Analisi Make-or-Buy: sviluppare software internamente (Make) o acquistare soluzioni esistenti (Buy).

## 3. Valutazione di Convenienza Economica:

- Valutazione dei costi (licenze, hardware, personale).
- Benefici economici: riduzione dei costi operativi, aumento dei ricavi.

## OUTPUT DELLO STUDIO DI FATTIBILITÀ

Un'analisi dettagliata che include:

- Obiettivi chiari e misurabili.
- Scelte progettuali con relativi vincoli tecnici e organizzativi.
- Stime di costi e benefici.

## 11.3 Ciclo di vita di sviluppo del SI

Lo studio di fattibilità conduce alla scelta della soluzione considerando fattori tecnici, organizzativi ed economici. La realizzazione del sistema informativo segue due approcci principali:

- **Make:** sviluppo interno di una soluzione.
- **Buy:** acquisizione di componenti già disponibili sul mercato.

L'*Enterprise Architecture* (EA) gioca un ruolo chiave nella pianificazione e sviluppo, offrendo una struttura di riferimento per tutte le fasi del ciclo di vita del sistema.

## CICLO DI SVILUPPO CON STRATEGIA *MAKE*

### Fasi del Ciclo

1. Pianificazione dettagliata della soluzione.
2. Raccolta e analisi dei requisiti per identificare le funzionalità necessarie.
3. Progettazione dell'interfaccia e analisi dettagliata.
4. Progettazione tecnica e realizzazione.
5. Messa in opera (*Deployment & Provisioning*) e gestione operativa (*Operations & Management*).

## Modelli di sviluppo

- **Waterfall (a cascata):**
  - Le attività sono eseguite in sequenza.
  - Le verifiche si fanno alla fine, causando potenziali ritardi e costi elevati.
- **Iterativo:**
  - Sviluppo incrementale tramite prototipi e release successive.
  - Il committente partecipa durante tutto il ciclo di vita, riducendo le divergenze.
- **Agile:**
  - Sviluppo rapido e iterativo con minor strutturazione, adatto a scenari dinamici.

## CICLO DI SVILUPPO CON STRATEGIA *BUY*

### Fasi del Ciclo

Il processo di sviluppo è preceduto dal **ciclo di acquisto**, che include:

- Raccolta e analisi dei requisiti (funzionali e non funzionali).
- Identificazione e valutazione delle soluzioni disponibili sul mercato.
- Selezione del pacchetto software più adeguato.

### Criteri di Valutazione

- **Indicatori funzionali:** completezza funzionale e personalizzabilità del software.
- **Indicatori architetturali:** scalabilità, interoperabilità e sicurezza.
- **Costi:** licenze, hardware, risorse umane e manutenzione.

### Valutazione del Fornitore

- Esperienza, reputazione, frequenza di aggiornamento del prodotto.
- Affidabilità dei *system integrator* e dei fornitori associati.

### Valutazione Economica

- Suddivisione dei costi (es. consulenza, hardware, sviluppo).
- Calcolo del costo totale di proprietà (TCO) e dei benefici tangibili.

### Selezione Finale

- Analisi dettagliata tramite demo e incontri con i fornitori.
- Chiarezza dei requisiti e adattabilità della soluzione.