

RIASSUNTO S.I. (tecnology)

ANGELO ANTONA

●CAPITOLI 1 & 2●

Principali della conoscenza → partendo dal basso verso l'alto, si hanno i seguenti livelli: **dati** = minime che rappresentano una determinata porzione di realtà, **informazione** = interpretazione di dati (si può pensare come "risposta ad una query sui dati"), **conoscenza** = capacità critica sull'informazione (se si è un tassista mi viene detto in un'aula in piena inverno, la conoscenza è la capacità di capire che in tale informazione c'è un'anomalia), **competenza** = capacità di integrare le decisioni critiche in base all'informazione (rispetto alla conoscenza applicata alle informazioni).

Classificazione dei processi: → i processi si possono classificare secondo due criteri.

- **Gerarchia di Anthony** → 5 livelli: processi operazionali, processi di controllo e processi decisionali.
 • **Gerarchia di Porter** → 5 occupazioni del solo livello operativo. In particolare divide i processi in processi primari (logistica in entrata, operazioni, logistica in uscita, gestione marketing, servizi) e processi secondari (gestione risorse umane, risorse finanziarie e sviluppo, ...).

Enterprise Architecture → l'enterprise architecture è una descrizione complessiva dell'organizzazione che comprende aspetti di business, aspetti organizzativi ed aspetti architetturali.
Inoltre l'EA di un'organizzazione fornisce una vista globale di tutti i suoi stakeholders dell'impresa. Un'accezione di comprensione dell'EA molto diffusa è il **processo di armonizzazione** che si compie con cui:
1. si una matrice nella quale o la colonne non hanno un ordinamento imposto (cioè, come, quando, perché), le righe rappresentano le attività strumentali che si possono o che non si fanno, che quindi hanno un ordine imposto (belle contigue = colore che si occupi della sintassi, quindi, belle contigue = il proprietario dell'impresa, belle logice = proprietà, belle logice = di dettaglio = programmatore).

Ripasso di gestione di un SI → quando si progetta un SI e bene man, ordine e coerenza i massimi segni il fondamentale standard di gestione di un SI, che si esprime in tre passi: a) pianificazione: fun di elezione del sistema e dei requisiti che dovrà realizzare; b) sviluppo: realizzazione concreta del SI, operazioni: operazioni di manutenzione ordinaria; c) controllo: standard periodici di verifica di rispetto la progettazione di SI nel dettaglio.

Progettazione di un SI $\rightarrow$ la progettazione di un SI si struttura in due operazioni principali:

La **priorizzazione strategica** → la **priorizzazione strategica** è la fase di progettazione in cui si ricomincia la cura dell'organizzazione che garantisce nel subire un intervento di informazioni. Uno degli strumenti usati a tale scopo è la **tabella di Porter e Miller**, ovvero una tabella in cui l'asse X rappresenta il **contenuto informativo del prodotto**, l'asse Y rappresenta il **contenuto informativo del processo**. Si cerca di individuare i punti del sistema da informatizzare al fine di far "caden" l'idea da un primo quadrante.

2) Studio di fattibilità → lo studio di fattibilità precisa le caratteristiche di un progetto interno. Si sviluppa nei seguenti passi:

Definizione degli obiettivi e dei requisiti funzionali → consiste nel definire più precisamente quali sono gli aspetti dell'organizzazione in cui operano, o decisioni della struttura funzionale, cioè dei requisiti che i nuovi prodotti dovranno soddisfare, o si delimitano i requisiti non funzionali in termini di qualità-costo-tempo.

3) Protezione degli interessi → si capisce di 3 for: ① identificazione degli interessi → si identifica con le parti coinvolte e si valuta quindi la convenienza di alcune tra molte; ② valutazione delle alternative → si valuta le alternative tenendo in un punto di vista tecnico ed in particolare si valutano le motivazioni, la disponibilità, la sicurezza e la sostenibilità della tecnologia, nonché la sua compatibilità con l'attuale sistema legislativo in presenza dell'organizzazione; ③ valutazione degli impatti organizzativi → oltre a fattori tecnici è necessario analizzare anche la compatibilità e flessibilità del personale in azienda: potrebbe essere necessario riorganizzare nella struttura gerarchica dell'azienda. Inoltre la resistenza e il cambiamento da parte del personale potrebbe far cambiare le implementazioni meno di quanto sperato.

3) Voluzione di convenienza economica → si valuta quale delle soluzioni che hanno superato i punti precedenti è più conveniente economicamente in termini di ricavi-costi, dove i costi sono quelli sostenuti per attirare l'investimento sul SE mentre i ricavi sono i ricavi differenziali riguardanti l'investimento stesso.

Ciclo di vita e sviluppo di un SI → l'approccio di sviluppo tradizionale è il cosiddetto **waterfall**, ovvero un approccio in cui lo sviluppo si realizza il SI per intero e si confronta con il committente solo alla fine della realizzazione. Questo approccio ha come svantaggio il fatto che il "controllo" da parte del committente avviene solo alla fine della realizzazione del software e nel caso vi sia un disallineamento tra il SI richiesto e le aspettative del committente, sarà necessario un dispendio di risorse (tempo e soldi) molto alto per risolvere il problema. Per ovviare a tali svantaggi, oggi si tende ad usare un approccio iterativo in cui vengono realizzati diversi prototipi in maniera incrementale, ponendo all'attenzione del committente che dovrà indicargli quando la direzione che si vuole che il SI segua.

● CAPITOLI BOAT ●

Prospettive di business → la prospettiva di business ha come obiettivo quella di formalizzare le motivazioni di business che spingono ad effettuare un intervento sul SI. In particolare è necessario specificare i seguenti elementi:

- **Partici** → sono i partecipanti dello scenario di business: hanno esseri, o governamenti (G), o cittadini (C), o business (B).
- **Oggetti** → sono i prodotti che vengono scambiati nell'interazione. Possono essere: o prodotti fisici, o prodotti digitali, prodotti ibridi, o prodotti finanziari, o servizi.
- **Orizzonti temporali** → si tratta della durata del rapporto. Si possono avere i seguenti orizzonti temporali: **statico** = i due partecipanti hanno un rapporto fisso, i pagamenti avvengono periodicamente e non per ogni singolo ordine. **Semi-dinamico** = è possibile cambiare partner di tanto in tanto, ma si è comunque legati ad un contratto e i pagamenti vengono effettuati periodicamente e non per ogni singolo ordine. **Dinamico** = il rapporto tra i due partecipanti è garantito solo all'interno del singolo ordine, i pagamenti avvengono per il singolo ordine. **Ultra-dinamico** = si può cambiare partner anche all'interno del singolo ordine se le condizioni lo richiedano.
- **Driver** → sono le motivazioni che spingono l'organizzazione ad effettuare l'intervento sul SI. Possono riguardare:
 - **efficienza** → è la capacità di raggiungere un obiettivo e si misura con il KPI (Key Performance Indicator) dato da $\frac{\text{costo del processo}}{\text{output}}$. Si distinguono due driver di efficienza: o **reach (estensione)** = aumentare l'estensione del business da un punto di vista temporale, spaziale o di canali, o **richness (ricchezza)** = migliorare il rapporto e l'intensità di comunicazione con i clienti.
 - **efficienza** → è la capacità di ottimizzare il rapporto $\frac{\text{output}}{\text{input}}$. Consiste molto spesso nell'automatizzazione dei processi.

Prospettiva organizzativa → la prospettiva organizzativa ha come scopo quello di descrivere gli aspetti organizzativi dello scenario. Sono previsti diversi livelli di dettaglio: **livello 0** = mercato visto come una *single player* mena; **livello 1** = si specificano le due parti principali in gioco ma si trattano gli intermediari come un'unica entità; **livello 2** = si specificano gli intermediari; **livello 3** = si separano front-end e back-end; **livello 4** = si specificano i singoli moduli di front-end.

È possibile inoltre che sia presente un livello intermedio tra front e back-end, detto **mid office**. Per quanto riguarda i processi con cui le organizzazioni interagiscono, vi sono tre tipi di flussi: **inter-organizzativi**, ovvero: o flussi **unilaterali** = c'è presente un'organizzazione coordinatrice del processo e le altre entità seguono la direttiva di tale organizzazione, o flussi **bilaterali** = due organizzazioni gestiscono in maniera condivisa il processo.

Prospettiva architetturale → nella prospettiva architetturale si schemano le componenti concettuali del SI. Come per la prospettiva organizzativa, nella prospettiva architetturale si definiscono diversi livelli di dettaglio: o **market-level** = si specificano i partecipanti, i loro moduli di FES (front-end System) e i messaggi che tali moduli si scambiano per la cooperazione inter-organizzativa; o **party-level** = si localizza l'attenzione su un singolo partecipante, specificandone i moduli di BES (Back-end System), i messaggi tra i moduli di BES e quelli tra i moduli di BES e FES; o **System-level** = si pone l'attenzione su un singolo componente del processo.

È possibile creare architetture di 1a tipologia. **Architetture di estensione** = architetture specifiche per una singola organizzazione in un determinato contesto, o **architetture standard** = architetture adottate ad un insieme di aziende, per esempio facenti parte di un gruppo. **Architetture di riferimento** = architetture generali, adottate a qualsiasi azienda che sia della tipologia adottata e nel contesto adottato. Un'architettura di riferimento può essere usata per creare un'architettura standard, che a sua volta può essere usata per creare un'architettura di estensione.

Nota: un'architettura party-level è un "crap" delle Enterprise Architectures (altisonante).

Prospettiva tecnologica → riguarda la tecnologia messa a supporto del SI. Ne parleremo nelle prossime pagine.

●CAPITOLO 8●

ERP → l'ERP è un modulo software che ha come scopo quello di automatizzare e gestire le attività operative dell'azienda. Si differenzia dai precedenti sistemi a Silos, i quali presentavano problematiche di integrazione e comunicazione con altri moduli, l'ERP è caratterizzato da una base dati unica (unicità dell'informazione), possibilità di acquisire le varie funzionalità del sistema separatamente (modularità) e imposizione all'azienda di best practices (possibilità, non è il sistema che si adatta all'azienda ma quest'ultima ad adeguarsi alle regole dell'ERP).

La struttura dell'ERP è suddivisa in:

- ERP core → è la componente principale dell'ERP ed è la prima ad essere stata progettata. Essa include moduli: interionali (gestione risorse umane, gestione amministrativa, ...), moduli esterni (dipendono dal settore del mercato in cui l'azienda opera) e moduli decisionali (pianificazione strategica, controllo budgetario).
- moduli aggiuntivi → sono moduli sviluppati successivamente ed integrabili all'ERP core. Alcuni dei moduli più diffusi sono: PLM (Product Life Cycle Management) è un modulo che traccia il prodotto dalla sua ideazione alla produzione conclusa, SCM (Supply Chain Management) è un modulo che gestisce la scorta in magazzino, CRM (Customer Relationship Management) che gestisce il rapporto col cliente.

Architetture dell'ERP può essere implementato in tre modi:

- architettura client-server → gli utenti si collegano al sistema tramite dei client interni all'azienda, i quali sono connessi al server principale.
- architettura web based → gli utenti possono connettersi al sistema utilizzando i loro dispositivi online dall'esterno dell'azienda.
- architettura SaaS (Software as a Service) → l'ERP viene fornito sotto forma di servizio dal vendor, abbastanza a costi inferiori e di conseguenza al cliente.

CRM → il CRM (Customer Relationship Management) è un modulo, spesso integrato come complemento dell'ERP, che mira alla gestione dell'interazione tra l'azienda e il cliente. La differenza di questo con il ERP, non tutte le aziende possono trarre vantaggio dall'adozione del CRM. In particolare, i vantaggi sono professionali ed umano di clienti che l'azienda possiede e alla dimensione finanziaria del rapporto col cliente. Il CRM consta di tre parti principali:

- CRM operativo → è il modulo di front-end del CRM e si occupa dell'interazione vera e propria con i clienti. Tale modulo è costituito a sua volta da seguenti sotto-moduli:

MODULO GESTIONE MARKETING

- modulo generazione lista clienti → una metodologia molto usata per inviare campagne di marketing o offrire minuti a categorie specifiche di clienti è quella di generare delle liste di clienti rubrici per segmentazione. La segmentazione viene effettuata dal CRM analitico, di cui si parlerà in seguito.
- modulo gestione campagne → è il modulo che supporta l'operatore nella definizione di una nuova campagna di marketing, fino all'analisi dei risultati aggiuntivi.
- modulo cross-selling e up-selling → con cross-selling si indica l'intenzione da parte dell'impresa di vendere più prodotti possibili al cliente. Con up-selling si indica l'intenzione di far scegliere al cliente l'oggetto più costoso.

MODULO AUTOMAZIONE VENDITE

- modulo gestione vendite → il modulo di gestione vendite assiste l'operatore in tutto il processo di vendita, dalla prima interazione col cliente fino alla chiusura del contratto. Sempre tra le "responsabilità" di questo modulo vi è quella di fornire un'interfaccia web sul portale web dell'azienda, che consente al cliente di compilare l'acquisto online.
- modulo gestione contratti → questo modulo si occupa di determinare quali tra i clienti dell'azienda sono adatti a ricevere una determinata proposta/campagna di marketing.
- modulo gestione opportunità → è molto simile al modulo di gestione contatti, ma a differenza di quest'ultimo la sua analisi si rivolge all'utente che non sono ancora clienti dell'azienda.

MODULO SERVIZI AI CLIENTI

- modulo di call-center → è il modulo che gestisce le chiamate di richiesta di supporto da parte dei clienti. Offre servizi come IVR (Interazione Response Voice) che consente ai clienti di auto-inserire la propria chiamata all'operatore più opportuno.
- modulo di Web-based Self-service → è un portale web che permette agli utenti di poter autonomamente risolvere i problemi comuni e facili da risolvere riguardanti il prodotto/servizio da loro acquistato.
- modulo di Call-Scripting → consente di consegnare informazioni personalizzate rilevanti dalle interazioni che l'azienda ha col cliente. In tal modo, in conversazioni successive, gli operatori possono accedere a tali informazioni per migliorare l'esperienza utente. Tale modulo può inoltre suggerire all'operatore una serie di domande da fare al cliente con lo scopo di individuare con precisione il problema e di consigliare la soluzione.

- CRM analitico → il CRM analitico è dati memorizzati nel DB del CRM gestiva e ne estrae informazioni utili agli scopi di reporting, ovvero categorizzare i clienti e compararne i dati, analisi, ovvero la segmentazione dei clienti in categorie, predicting, cioè predire i bisogni dei clienti.

Data Warehousing → è una warehouse, nessuna dell'esistenza di dimensioni i dati per usare a scopo decisionale e strategico. In particolare sono importanti le seguenti dimensioni: DATA WAREHOUSING: insieme di operazioni utili a popolare e a mantenere la base di dati decisionale, DATA WAREHOUSE: base di dati decisionale.

L'architettura di un DW si può dividere in:

- al livello più basso vi sono le fonti delle quali vengono estratti i dati di interesse (spesso tali fonti sono i DB operazionali).
- i dati estratti danno origine alle operazioni di trasformazione del flusso di rendere tali dati coerenti e compatibili tra loro. Per effettuare tali operazioni è possibile che si faccia uso di un DB intermedio detto Staging area.
- vi è poi il Data Warehouse vero e proprio, nel quale vengono caricati i dati già trasformati. Tale contenitore di dati può raggiungere dimensioni molto elevate e quindi può essere inteso e interpretato a tutti i dati presenti nel DW complessivi. Per questo motivo spesso si usano dei Data Marts, ovvero dei DW concepiti concettuali solo un sottoinsieme di dati del DW centrale.

Un altro aspetto molto importante da analizzare è che i DW sono di tipo OLAP (Online Analytical Processing) e per così dire proprietà OLAP (olap, consistent, isolate, durable) basati sul modello FASHI (fast, analytical, multidimensional, interactive). Per risolvere tali problemi il sistema deve essere composto da un modello Multidimensionale. Tale tipologia di modello è rappresentata da un ipercubo. Tale ipercubo è composto da:

- fatti → sono le unità elementari dell'ipergrafo e rappresentano l'evento di interesse per l'analisi (es. vendita).
 - misure → sono le caratteristiche del fatto che sono interessanti per l'analisi (es. prezzo di vendita, percentuale di sconto applicato).
 - dimensioni → sono le coordinate dell'ipergrafo che identificano il fatto (es. data di vendita, filiale nella quale si è svolta l'attività, la vendita).
- Tale modello può essere rappresentato tramite il DFM (dimensional fact model) in cui il fatto è rappresentato come un rettangolo contenuto nella misura del fatto mentre le dimensioni sono rappresentate come filari disposti in ordine gerarchico e connessi al fatto principale. È inoltre possibile inviare dagli attributi operativi contenuti nel cubo una struttura non necessariamente corrispondente.

Come detto, è necessario affluire dati alle operazioni ai dati prima di poterli inserire nel dato warehouse. Tali operazioni si riassumono nell'ETL:

- **Extrazione** → consiste nell'estrazione dei dati dalle fonti. Tale estrazione può essere completa (tutti i dati vengono prelevati) o incrementale (s. estraggono solo i dati aggiunti o modificati dopo l'ultima estrazione).
- **Trasformazione** → i dati estratti devono essere trasformati secondo i passaggi seguenti: 1) pulizia dei dati → vengono risolti problemi di mancanza di dati/mancata consistenza di dati; 2) Ricomposizione → si aggregano tutti i dati che fanno riferimento allo stesso fatto; 3) Standardizzazione → si affiniscono genericamente le operazioni di comparazione/consolidamento di campi o l'omogeneizzazione delle unità di misura al fine di rendere i dati compatibili; 4) Rimozione di duplicati.

Parlando ora di implementazione concreta del DWH, vi sono tre scelte possibili:

- usare un modello **HOLAP** → consiste nel riprendere fedelmente il modello multidimensionale nel concetto. I vantaggi di tale approccio consistono nella facilità di risposta del sistema alle interrogazioni. Tale schema ha però lo svantaggio di richiedere calcoli specifici sulle basi di dati multidimensionali per essere usato. Inoltre si ha un enorme spreco di spazio poiché ogni dato necessario all'analisi lo spazio di ogni fatto dell'intersezione e quindi allora non si è in grado di usare un modello **ROLAP** → è il modello Rolap, ottimizzato per la logica multidimensionale delle classiche tabelle relazionali. Vi sono due modi per effettuare ciò:
 - **diagonale a celle** → il modello multidimensionale viene completamente esplicito. Vi è una tabella contenente le misure e le tabelle contenenti le dimensioni...
 - **In tal caso le interrogazioni del sistema si riducono a delle semplici join a tutte le tabelle.**
 - **diagonale a blocchi di misure** → mantengono l'aggettamento del modello multidimensionale in tabelle relazionali; in tal caso si conservano ancora alcune informazioni generali.

Boni generali:

Il modello **ROLAP** risolve i problemi dello spreco di spazio e quello della necessità di conservare spaziosi per l'uso del sistema, ma perde con sé lo svantaggio della **lentezza** nel rispondere a query molto complesse.

- usare un modello **HOLAP** → il modello HOLAP consiste nell'usare un modello **ROLAP** per il DWH centrale (di grandi dimensioni: risposta lenta) mentre si usa un modello **ROLAP** per i dati-marginali (di piccole dimensioni, più scomposti la velocità di risposta che il rapporto di spazio).
- Le operazioni che si possono effettuare su un DWH sono: **roll-up** = salire di livello gerarchico in uno o più dimensioni dell'**cube**; **drill-down** = scendere di livello gerarchico in una o più dimensioni; **slice** = fare una singola dimensione così da ottenere una "fetta di spicco"; **dice** = fare più dimensioni così da restringere il campo di analisi.

Data Mining → un'altra delle tecniche per esplorare dei dati: operative informazioni usabili a livello decisionale e strategico e il dato-mining

Le tecniche di dato-mining si possono suddividere in "tecniche di apprendimento supervisionato" e "non supervisionato". Un'altra distinzione può inoltre essere effettuata tra le tecniche predittive e descrittive. Analizziamo le tecniche trattate nel caso.

Regole associative → si tratta di algoritmi che mirano a trovare regole utili ed interessanti connesse da rapporti causali all'interno dei dati operazionali. Tali regole vengono spesso rappresentate come espressioni del tipo $A \Rightarrow B$, la "velocità" di una regola associativa si misura in base a **Confidenza** = $P(B|A)$ e **Supporto** = $P(A \cup B)$. Le regole associative vengono spesso usate per analizzare il consumo di clienti di siti di e-commerce al fine di identificare prodotti acquistati spesso insieme.

Algoritmi di classificazione → sono tecniche predittive supervisionate. Il loro scopo è suddividere un insieme di elementi dati in input in delle categorie definite a priori. Per costruire un insieme di dati la cui classe di appartenenza è nota a priori e si costruisce un algoritmo in grado di riconoscere le caratteristiche che distinguono gli elementi in una o nell'altra classe. Una volta raggiunto l'obiettivo, si viene l'algoritmo attraverso un test su (insieme di elementi di cui è nota la classe di appartenenza) e si ne calcolano le misure (Velocità di risposta) e **Ricordo** (Velocità di risposta).

Si possono implementare gli algoritmi di classificazione in molti modi. Uno dei più noti è tramite un albero di decisione, ovvero un albero in cui ogni nodo rappresenta un attributo di classificazione. Da scelto dell'attributo viene effettuato un test che il sotto-insieme tra due livelli successivi sia molto alto molto basso la differenza di entropia tra i nodi che si trovano allo stesso livello sia elevatissima nulla. Il processo di classificazione termina quando l'agente di un ulteriore attributo non apporta un contributo entropico rilevante.

Algoritmi di clustering → gli algoritmi di clustering sono tecniche descrittive non supervisionate. Tali tecniche permettono di suddividere gli elementi di un dato-set in insiemi senza conoscere a priori le classi che si ottengono. La regola che vi sono molti algoritmi di clustering sono le seguenti: 1) elementi appartenenti alla stessa cluster hanno distanza minima, 2) elementi appartenenti a cluster diversi hanno distanza massima. Affinché sia possibile calcolare la distanza tra gli oggetti è necessario che gli attributi siano rappresentati numericamente e che sia definita una funzione di distanza. La distanza tra due elementi.

Gli algoritmi di clustering si distinguono in: **algoritmi esclusivi** (un elemento può appartenere ad un solo cluster); **algoritmi inclusivi** (un elemento può appartenere a più cluster); **hard** (ogni elemento appartiene a un solo cluster, con un grado di appartenenza espresso da un valore numerico); **algoritmi fuzzy** (tutti gli elementi appartengono almeno a un cluster) e **algoritmi ibridi** (è possibile che un elemento non appartenga a nessun cluster).

Le tecniche di clustering si valgono un bene a: **flexibilità** = capacità di analizzare elementi categorici oltre che numerici; **robustezza** = capacità di classificazione nonostante anche a seguito di variazioni negli attributi del dato-set; **efficienza** = velocità con cui si ottengono i cluster completi.

Uno dei più famosi algoritmi di clustering è l'algoritmo **K-means**, il cui passaggio sono i seguenti: 1) Si sceglie casualmente K (numero noto a priori) centri tra gli elementi del dato-set; 2) Si assegnano tutti gli altri elementi al centro più vicino; 3) Si ricalciano i centri; 4) Si ri-assegnano gli elementi al nuovo centro più vicino... L'algoritmo termina quando si giunge a convergenza.

Process Mining → il process mining è un insieme di operazioni e tecniche che hanno il fine di estrarre informazioni dai processi aziendali. L'informazione viene estratta attraverso l'analisi di log, cioè delle registrazioni delle operazioni avvenute durante l'esecuzione dei processi. Le operazioni di process mining consistono di 3 passaggi:

- 1) **Descrizione** → consiste nel ricostruire il flusso effettivo dei processi attraverso i log.
- 2) **Conformanza** → si confronta il flusso ricostruito con quello previsto dalle politiche aziendali.
- 3) **Enhancement** → a partire dai risultati si ricavano informazioni sui possibili interventi al SI al fine di correggere/migliorare/ottimizzare i processi.

●CAPITOLO 9●

Architettura di integrazione → sappiamo che i sistemi informativi contengono più applicazioni che determinano il funzionamento del sistema informatico stesso. Tali applicazioni necessitano di comunicazione, ovvero necessitano di scambiarsi dati, i quali dovranno quindi essere integrati. Inoltre anche i singoli moduli sono a loro volta composti da dei sotto-moduli, che dovranno anch'essi essere integrati tra loro. Per mettere in pratica tutto ciò sono nati i seguenti tipi di integrazioni:

- Integrazione dei dati** → come visto per i DW, i dati riferiti ad un medesimo oggetto possono non essere compatibili tra loro a causa di differenze sintattiche o semantiche. Sono quindi necessarie integrazioni di dati in maniera simile a quanto avviene nelle operazioni di ETL dei DW.
- Integrazione dei processi** → nella comunicazione tra due moduli di un medesimo applicativo non basta integrare i dati; è altresì necessario che i due moduli condividano l'ordine di esecuzione dei messaggi, ovvero è necessario che i loro processi siano integrati.
- Integrazione degli applicativi** → quando si parla di integrare applicativi di natura diversa, vi sono principalmente 3 strade percorribili:
 - **integrazione punto a punto**: ogni applicativo ha un collegamento diretto con ogni altro modulo applicativo. Gli svantaggi principali di tale approccio sono due: 1) la logica di integrazione è integrata nei moduli stessi, il che rende molto complesso eventuali modifiche; 2) sono necessarie N² interfacce e il cambio di un componente è necessario modificare N-1 interfacce. Questo implica che la soluzione non è scalabile.
 - **integrazione hub/spoke**: tale soluzione consiste nel racchiudere la logica di integrazione in un dispositivo "intermediario", ovvero l'hub, al quale si collegano i moduli applicativi tramite degli spoke, che fungono anche da router. Il vantaggio di tale soluzione è racchiusa nella maggior scalabilità data che in tal caso sono necessarie N interfacce e che a seguito della modifica di un componente basta modificare la relativa interfaccia. Permane però la difficoltà nella definizione e nella modifica della regola di integrazione all'interno dello spoke.
 - **integrazione tramite BPS** → quando è molto difficile descrivere la regola di integrazione all'interno dell'hub diventa comodo adottare come soluzione un BPS, ovvero un componente applicativo che consente di essere la regola di integrazione a livello applicativo, ma separando tale logica dai moduli applicativi veri e propri così da facilitare eventuali modifiche e mantenendo una buona scalabilità.

SOA (Software Oriented Architecture) → un altro approccio utilizzabile per "costruire un sistema informativo tramite la composizione di moduli" è quello di implementare tali moduli acquisendoli come web-based services da dei service provider. Un servizio, per essere tale, deve garantire la funzionalità del corrispondente modulo in maniera indipendente dagli altri moduli. Inoltre, al fine di garantire che si vuole che il modulo sottinteso siano rispettati al servizio viene associato un SLA (Service Level Agreement), ovvero un documento che esprime in termini di QOS (Quality Of Service) quali requisiti sottintende il sistema. Prendendo ora dal punto di vista del service provider, al fine di rendere disponibile il servizio egli dovrà inserirlo in un Service Registry, che contiene a SLA di vari servizi e ne indica i comandi da fare (si ricorre all'interfaccia per verificare della funzionalità del modulo). Tali comandi sono spesso espressi nei protocolli basati su HTTP (protocollo SOAP). Come detto, un sistema informativo è composto da molti moduli che comunicano tra loro e anche nel caso di architettura orientata ai servizi tale approccio non cambia. È quindi necessario ordinare le invocazioni di tali servizi affinché quelle che vengono chiamate "orchestrazione di servizi". Tale orchestrazione avviene in maniera simile a quanto occorre per l'integrazione di moduli, con la differenza (ed il vantaggio) che in questo caso è possibile cambiare dinamicamente il provider del servizio. L'orchestrazione di servizi può avvenire in due modalità: "orchestrazione centralizzata" in cui una singola entità ha l'onere di sincronizzare i servizi e fornire la funzionalità del modulo completo, o "orchestrazione distribuita", nella quale i provider sincronizzano i loro servizi in modalità peer-to-peer (in tal caso si parla di "composizione di servizi"). All'interno di un sistema informativo, la sincronizzazione tra i servizi si può effettuare tramite un **Enterprise Service BUS**, che può anche essere combinato con dei workflow component come BPS.

Note: per facilitare la scelta dei servizi nel service registry si può per ogni servizio associare a dei **Service Banner**, che hanno come compito quello di creare di ritorno il massimo allineamento tra i servizi richiesti dal sistema e quelli offerti dal servizio completo, composto da vari "servizi elementari" presenti nel Service Registry. **Note:** nel sistema architetturale REST (Representational State Transfer) si parla di **resourceful services** quando le richieste vengono inviate direttamente ad una risorsa URI, o esse usano bandi quando la risposta vengono fornita in un linguaggio standard, solitamente basato su HTML.

●CAPITOLO 10●

Architettura distribuita su più livelli → la applicazione alla base di un SI possono essere molto complesse, tuttavia è possibile schematizzare la struttura in 3 livelli: 1) livello di presentazione → è il livello che gestisce l'interazione con l'utente. Tale interazione avviene tramite GUI. 2) Livello di logica applicativa → è il livello che si occupa di fornire le funzioni da rendere disponibili all'utente. 3) Livello di logica di accesso ai dati → è in pratica il DBMS. Gli utenti dei livelli presentati si relazionano con il mondo esterno attraverso delle API, ovvero interfacce non proprio per il loro uso bensì invocabili da altri programmi. I tre livelli all'usarsi possono essere distribuiti in modo diverso sul hardware. (livelli applicativi: layer; livelli hardware: tier). Si possono avere:

- **Architettura single-tier** → tutti e i 3 livelli logici sono installati su un singolo mac elaborativo. I vantaggi di tale approccio consistono nella semplicità di implementazione e manutenzione. I problemi di tale implementazione sono invece la necessità di avere molti con client proprio di calcolo e la bassa scalabilità.

- **Architettura two-tier** → i layer applicativi vengono installati su due tier fisici. Si creano quindi diverse combinazioni possibili, raggruppabili in due categorie: 1) **thin-client** → la logica applicativa è quella di accesso ai dati sono contenute nel modo di "base end" mentre i terminali degli utenti possiedono solo il livello di presentazione. È una buona scelta di configurazione quando si hanno client con poca potenza di calcolo. Tuttavia il buon funzionamento del sistema è fortemente dipendente dalla rete che collega i nodi. 2) **thick-client** → i client contengono parte della logica applicativa. Questo rende il sistema meno dipendente dall'ordinamento della rete, ma comporta la necessità di installare client più potenti e incrementa la difficoltà di upgrade del sistema.

- **Architettura three-tier** → si aumenta il numero di tier a 3. Anche in questo caso si hanno diverse configurazioni possibili ed anche in questo caso si possono distinguere in configurazioni **thin** / **thick** client a seconda del numero di layer presenti sul client. L'architettura a tre tier permette di elaborare un modo tier, il client ed il DBMS. Tale modo potrebbe per esempio contenere la logica applicativa, che viene condivisa da tutti i client. In tal modo si alleggerisce il carico di richieste al DBMS, dato che questo dovrà essere sottoposto ad un'unica interminata connessione col singolo nodo di logica applicativa.

- **Architettura n-tier** → sono architetture in cui la logica applicativa viene suddivisa su più di 3 livelli fisici. I SI basati sul web formano spesso parte di questa categoria. In particolare si usano 5 tier: uno per la logica di accesso ai dati, uno per la logica applicativa e tre per il layer di presentazione (uno per la logica, uno per il client e uno per il web server e l'ultimo per la logica engine).

Scalabilità → la scalabilità è la capacità di un sistema di aumentare/diminuire la propria risorse in base all'effettiva necessità del momento. Questa proprietà, nel caso di sistemi sul web è detta "elasticità". La scalabilità può essere: 1) **orizzontale** → aumentare il numero di nodi elaborativi, particolare aumento la potenza dei singoli nodi. Gli aspetti per migliorare la scalabilità di un sistema sono a seguito:

- **server farm** → le server farm sono di centri elaborativi costituiti da molte unità computazionali su cui viene suddiviso il lavoro di calcolo. Il partizionamento può essere effettuato in modo:

- **clonazione** → si installano molti nodi su quali vengono installati le stesse applicazioni. Si presume che i dati siano condivisi tra tutti i nodi o duplicati su ogni nodo si parla di clonazione **shared-disk** o **shared-nothing** da configurazioni **shared-nothing** sono ottimali per applicazioni **read-only**, ma le loro implementazioni e mantenimento per applicazioni **write-intensive** può essere molto oneroso. Tuttavia ha i dati contenuti nei diversi nodi. I vantaggi di un'implementazione a clonazione sono la riduzione di punti e la complessità nell'espansione della server farm dato che non è necessaria modificare la struttura dei tier a seguito dell'acquisto di un nuovo nodo elaborativo. L'insieme di client è detto **RMS**.

- **partizionamento** → si suddivide il carico elaborativo sui diversi nodi predefiniti le funzioni offerte su diversi nodi: ogni richiesta viene indirizzata al server responsabile. In tal modo la quantità di query è minore rispetto all'approccio di clonazione, ma si ha una maggiore scalabilità.

Virtualizzazione → questa si usa un apposito software (chiamato **server** presentato ed chiamato di RMS). In tal modo si combinano i vantaggi della due implementazioni. La virtualizzazione è una tecnologia che consente di distinguere la risorsa fisica della macchina dall'elaborazione virtuale della stessa. Ciò avviene tramite un **hypervisor**, cioè un componente software che "usa" delle risorse virtuali (VRM) che sono le risorse fisiche del sistema. L'insieme di tali risorse è detto "virtual machine". Se chiamata alla CPU da parte di un'applicazione, viene colata su una VM si risolve in realtà in una richiesta alla CPU, risolve che poi a sua volta inoltra alla CPU fisica. Se da un lato tale approccio comporta un dispendio nella prestazioni dell'elaborazione, l'installazione di molte virtual machines su un singolo nodo fisico le applicazioni girano quindi "ovvero" l'impressione di possedere la intera risorsa della macchina per la propria esecuzione, quando invece stanno facendo eseguire su una VM installata su un nodo fisico contenente alla VM.

La virtualizzazione ha trovato largo uso nelle **Server farm**, dove è stato usato per effettuare "emulazione", cioè installazione di più macchine virtuali sulle singole macchine fisiche. Inoltre in questo modo, non avendo la macchina virtuale legata alla specifica macchina fisica, è possibile migrare la VM da un nodo all'altro.

Cloud Computing → grazie alla virtualizzazione è stata possibile la diffusione del Cloud Computing. Le server farm infatti hanno costi di creazione e gestione molto elevati e solo le organizzazioni più grandi hanno potuto sostenere le alte organizzazioni hanno deciso di offrire di servizi, prima che mettessero loro a disposizione le risorse richieste. Il cloud computing è caratterizzato dalle seguenti caratteristiche principali:

- **on-demand self-service** → l'utente deve poter accedere al servizio in qualunque momento attraverso la rete pubblica.
- **on-demand self-service** → il cliente deve poter richiedere autonomamente il servizio in qualsiasi momento.
- **indipendenza della localizzazione** → l'utente deve poter accedere ai dati ovunque essi si trovino.
- **multitenancy del consumo** → il cliente paga per un servizio proporzionale ai suoi consumi.
- **elasticità** → il sistema deve essere in grado di allargare/ridurre la risorsa ad un utente a seconda delle necessità in maniera dinamica.

I servizi offerti dal provider di cloud computing possono essere divisi e categorizzati in:

- **infrastructure as a service (IaaS)** → il provider fornisce solo le risorse fisiche tramite cloud. Su di esse l'utente può in autonomia installare della VM nelle quali può girare i propri applicativi.
- **platform as a service (PaaS)** → il provider fornisce sia le risorse fisiche che la tecnologia di piattaforma (es. DBMS). L'utente ha ancora qualche possibilità di personalizzazione, ma le sue azioni di modifica si concentrano più che altro a livello applicativo.
- **software as a service (SaaS)** → il provider fornisce vari o propri applicativi e cui l'utente non ha possibilità di personalizzazione.

Esistono poi diverse modalità di deployment (distribuzione):

- **private cloud** → l'infrastructure cloud è ad uso esclusivo di una singola organizzazione. Tale approccio ha come vantaggi la possibilità di personalizzazione e la sicurezza, ma comporta input costi e una bassa scalabilità.
- **community cloud** → simile al private cloud, ma l'infrastructure cloud è condivisa da un insieme di organizzazioni accomunate tra loro. Tale approccio è caratteristico da livelli di sicurezza elevati e da scalabilità migliorata rispetto al private cloud.
- **public cloud** → l'infrastructure cloud è pubblico (grazie ad un auto private o sociali). In tale approccio i livelli di sicurezza sono drasticamente più bassi, ma le loro grandi i costi.
- **hybrid cloud** → l'infrastructure cloud è formato dall'unione di due delle precedenti opzioni.

Prima di una scelta progettuale di una organizzazione che deve decidere in che modo gestire il proprio SI, tale scelta può scegliere tra le seguenti opzioni: 1) **Dispersa**: la gestione avviene all'interno dell'organizzazione stessa, 2) la gestione viene effettuata da una società "esterna" che fornisce servizi esclusivamente all'organizzazione, 3) **capitale dell'analisi**: la gestione avviene internamente all'azienda ma col supporto di una società di consulenza. 3.a) alcune componenti dell'azienda (spesso le più critiche/meno critiche) vengono esternalizzate. 3.b) alcune componenti dell'azienda (spesso le più critiche/meno critiche) vengono esternalizzate. 3.c) alcune componenti dell'azienda (spesso le più critiche/meno critiche) vengono esternalizzate. 3.d) alcune componenti dell'azienda (spesso le più critiche/meno critiche) vengono esternalizzate.

Un provider può fornire uno spazio fisico (es. slot) in cui il cliente può allocare le proprie risorse fisiche di sua proprietà (hardware) oppure fornire direttamente le risorse virtuali. In entrambi i casi è importante che il cliente e il provider concordino dei livelli di prestazioni garantite attraverso la compilazione di un SLA (Service Level Agreement) e la definizione di KPI (Key Performance Indicator).

●CAPITOLO 1●

Sicurezza e minacce → la sicurezza nei sistemi informatici consiste nell'assicurarci che ogni intento nel sistema possa accadere a tutti e solo a tutti per cui gli obiettivi di sicurezza sono: **Autenticazione** → un sistema sa sempre due utenti della seguente proprietà: **Identità** → i dati non devono essere modificati non autorizzati, **Autenticazione** → nel sistema ogni singolo che riceva un messaggio deve poter conoscere l'identità del mittente. Il concetto di autenticazione si basa sul concetto di autenticazione: ogni soggetto che voglia interagire col sistema deve prima farsi riconoscere. **Integrità** → non devono esserci vie false accessi ad informazioni non autorizzate. Quando la proprietà di riservatezza si riferisce ai dati del sistema prende il nome di privacy e protezione.

Disponibilità → il sistema deve sempre essere disponibile nei tempi e nei modi predefiniti.

La sicurezza di un sistema può essere minata da: **Attacchi** (fori, leggi o accidenti). Quelli trattati nel corso sono gli attacchi **leggi**. Gli attacchi **leggi** possono essere:

- Attacchi a livello di rete** → in questa categoria ricadono attacchi come: **Sniffing** (furto di pacchetti al fine di intercettare i messaggi in una comunicazione), **spoofing** = imito di messaggi con l'intento di mettere in pericolo al fine di impersonare un altro soggetto, **man-in-the-middle** = combinazione di tecniche di sniffing e spoofing al fine di deviare l'intera comunicazione verso un soggetto terzo, **flooding** = imito di molte macchine di pacchetti al fine di intorpidire il sistema e renderlo non disponibile (denial of service).
- Attacchi a livello applicativo** → in questa categoria si annoverano attacchi come: **virus** = pezzi di codice combinati all'interno di programmi che se eseguiti sono in grado di infettare altri file, duplicarsi e creare copie infinite di se stessi, **worm** = sono simili ai virus, ma a differenza di questi i worm non hanno bisogno di allegarsi a un file. I worm si diffondono, ripuliti sono programmi opportunamente creati con intenti furbi e malevoli; **ransomware** = software che impediscono l'accesso al sistema fino al pagamento di un riscatto (ransom). Tutti i software elencati ricadono nell'appellativo di "malware".

Vi sono poi altri attacchi a livello applicativo come per esempio la **bruteforce** = vulnerabilità basata su tentativi ripetuti di forzare la password e la **replay attack** = registrazione del sistema o **spoofing** = software che hanno come obiettivo quello di nascondere informazioni riservate ad un sistema terzo non autorizzato.

Crittografia → la crittografia è un sistema di cifratura che si basa su due elementi fondamentali: **la chiave** e un **algoritmo di cifratura**.

Esistono due tipi di crittografia:

- Crittografia simmetrica** → nella crittografia simmetrica i due soggetti della scambio condividono una singola chiave di cifratura che, usando l'algoritmo concordato, usano per cifrare/decifrare i messaggi della comunicazione. Cio vuol dire che il punto critico della crittografia simmetrica è la riservatezza della chiave. Gli algoritmi di crittografia simmetrica fanno uso delle seguenti operazioni:

- associare** → consiste nel associare la lettera del messaggio con altre lettere secondo un algoritmo ben preciso. Un esempio molto semplice di associazione è il cifrario di Cesare. Tale algoritmo consiste nel sostituire ogni lettera del messaggio con la lettera che si trova K posizioni avanti nell'alfabeto (K è chiamata **cifra**).
- permutare** → dato una chiave K = parola, di lunghezza n, la lunghezza della parola indica in quale colonna suddividere le lettere del messaggio, l'ordine.

La crittografia delle lettere associando la parola chiave insieme in che ordine mettere tali colonne.

I punti negativi della crittografia simmetrica consistono nel fatto che tale crittografia può essere implementata con hardware specifico, rendendo molto veloci le fasi di cifratura e decifratura da eseguirsi. La crittografia simmetrica riguarda la gestione della chiave e la sua conservazione.

- Crittografia asimmetrica** → nella crittografia asimmetrica ogni soggetto possiede due chiavi: una pubblica e una privata. Gli algoritmi alla base della crittografia asimmetrica si dividono in funzioni di chiave pubblica e funzioni inverse. È difficile da un punto di vista computazionale (es. prodotto fra numeri primi molto grandi). È necessario che valga la seguente condizione: **D** tutto ciò che viene cifrato con la chiave pubblica può essere decifrato solo con la rispettiva chiave privata o viceversa. **D** non due sono possibili: o una o l'altra, ma una chiave pubblica della conversazione dell'altro.

La crittografia asimmetrica può garantire la proprietà di autenticità e riservatezza nel seguente modo: quando si manda un messaggio, si utilizza una **chiave privata**.

- si cifra il messaggio con la **chiave privata** del mittente in modo che il ricevente, decifrando il messaggio con la chiave pubblica del mittente, non abbia dubbi sull'identità del mittente (proprietà di autenticità).
- si cifra il messaggio con la **chiave pubblica** del ricevente. In questo modo si può decifrare il messaggio solo se si possiede la chiave del ricevente e quindi solo se si è il ricevente (proprietà di riservatezza).

Un algoritmo di crittatura molto diffuso è il **RSA** in cui i messaggi sono i numeri. Si scelgono due numeri primi X e Y e si calcola $N = X \cdot Y$ e $Z = (X-1)(Y-1)$. Una volta stabilito ciò si calcola $D = \text{numero} < N$ che non ha fattori comuni con N . E tale che $(E \cdot D) - 1$ sia divisibile per Z . In tal modo otteniamo che $K_{pr} = (D \cdot N)$ e $K_{pub} = (E \cdot N)$. Le operazioni effettuate saranno quindi cifratura $= H^E \cdot INI$, decifratura $= C^D \cdot INI$, con H messaggio in chiaro e C messaggio cifrato.

Le operazioni degli algoritmi a chiave asimmetrica consistono nel fatto che la cifratura e la decifratura sono per algoritmi per questo motivo spesso si usa la crittografia simmetrica per creare un canale di comunicazione sicuro sul quale inviare una chiave di crittografia simmetrica per poi continuare la conversazione con la crittografia simmetrica.

Funzione di Hash e impronta digitale → come visto, la crittatura è in grado di garantire la proprietà di autenticità e riservatezza. È necessario però garantire anche la proprietà di integrità. A tale scopo si usa la funzione di hash, cioè una funzione che, preso in input un messaggio di lunghezza variabile, dà in output una stringa di lunghezza fissa. Tale output prende il nome di **impronta digitale**. E i messaggi che sono rispettati le seguenti condizioni: **D** dato due input diversi, due output sono differenti, cioè la funzione di hash genera due impronte digitali, quindi **D** non due in alcun modo sono possibili, rendendo il messaggio protetto dal digito.

Il funzionamento dell'operazione dell'impronta digitale è il seguente: **D** l'input è un dato, il digito fornisce la funzione di hash e viene il messaggio, **D** il decrittamento sopra il messaggio del digito, si calcola il digito e confronta il digito fuori con l'impronta ricevuta: se sono uguali il messaggio è integro.

Nel caso si vuole garantire anche la proprietà di autenticità e riservatezza è possibile passare con una **funzione di hash** a crittatura il digito, ottenendo la **Firma Digitale**.

Gestione delle chiavi → la gestione (creazione, distribuzione e conservazione) delle chiavi richiede di risolvere vari problemi. Per prima cosa, idealmente le chiavi dovrebbero essere create e conservate da chi le farà poi uso, in un ambiente sicuro. **Random Number Generator (RNG)**. Per quanto riguarda le chiavi di crittografia simmetrica, questa operazione di generazione delle chiavi avviene in un canale di comunicazione sicuro, creato per esempio attraverso una **funzione di hash** asimmetrica. Gli algoritmi di crittografia asimmetrica non hanno invece questa necessità poiché il fatto che la chiave pubblica sia nota a tutti non è un problema. Tuttavia il problema è associare alla chiave pubblica il corrispondente soggetto. A tale scopo sono stati creati i **PKC (Public Key Certificate)** che racchiudono le informazioni riguardanti la chiave pubblica. Tali certificati vengono creati da una **PKI (Public Key Infrastructure)**, ossia un'entità che fornisce l'infrastruttura per generare e distribuire chiavi. Questo non è un utente necessario di una nuova chiave, ma è richiesto da una **Certificate Authority**, che svolge anche il ruolo di **Registration Authority** e registra l'associazione tra la chiave e il suo proprietario.

Il ruolo della **Registration Authority** è di corrispondere alla **PKI** in un archivio centrale, nel momento di creare la chiave. Nella realtà delle **PKI** (Public Key Infrastructure) della chiave (e a seguito del suo firma) la **CA** o la **RA** possono richiedere ad una **Registration Authority** al fine di ricevere il **KPC** relativo alla chiave richiesta o compromessa.

Autenticazione → nel caso un utente voglia accedere ad una qualunque parte di un sistema, le prime operazioni da effettuare sono l'autenticazione e l'autorizzazione. L'autenticazione consiste nel riconoscimento dell'identità dell'utente e può essere effettuata tramite **SVM (Something you know)** come una password, **SVH (Something you have)** come un token, **SVH (Something you are)** come un'impronta digitale. L'autenticazione "classica" prende il via da un campione di una password associato, utilizzato per derivare un maggior livello di sicurezza spesso si usano algoritmi di **OTP (one time password)**.

Autorizzazione → lo scopo successivo all'autenticazione è l'autorizzazione, cioè la verifica che l'utente abbia effettivamente i diritti di accesso ai dati per i quali sta facendo richiesta. Per comprendere come tale verifica venga effettuata, è necessario introdurre i sistemi di controllo di accesso ai dati nei DBMS. Il controllo dell'accesso viene effettuato attraverso regole formate dai seguenti elementi: **1. autorizzatore** = chi ha accesso ai permessi sulla risorsa, **2. soggetto** = chi ha richiesto l'accesso alla risorsa, **3. oggetto** = la risorsa alla quale si vuole accedere, **4. diritto** = le operazioni che il soggetto è autorizzato a compiere sull'oggetto.

Del punto di vista della regola di accesso ai dati, i sistemi sono classificabili in sistemi aperti: tutto ciò che non è espressamente vietato è consentito, e sistemi chiusi: tutto ciò che non è espressamente consentito è vietato.

Le regole di accesso sono di solito attese la tripla (soggetto, oggetto, diritto, connessi, proprietario).

Politica di accesso ai dati → abbiamo visto in che modo il sistema verifica i diritti di accesso di soggetti sugli oggetti, vediamo ora in base a quali politiche tali processi vengono "deviati" ai membri dell'organizzazione. Esistono principalmente due tipi di politiche di accesso:

- **DAC (Discretionary Access Control)** → è un sistema in cui "regia" il Discretionary Access Control, il creatore di un dato ne è anche l'owner e può dare a chi conosce i privilegi di accesso a tale risorsa. Spesso si stabiliscono i privilegi in ruoli, così che sia più facile gestire i permessi di accesso. Il DAC è adatto a sistemi in cui la sicurezza non è un punto critico. Da un punto di vista pratico il DAC può essere applicato in un DBMS implementando la regola di accesso a livello del SO o nei comandi SQL. Per esempio è possibile che il sistema integri la query SQL dell'utente con ulteriori restrizioni che facciano sì che la risposta alla query contenga solo le informazioni a cui l'utente può accedere legittimamente.
- **MAC (Mandatory Access Control)** → nei sistemi con sicurezza critica (es. sistemi militari) è necessario adottare una politica di accesso ai dati di tipo MAC. In tale approccio i soggetti vengono suddivisi per clearance, e gli oggetti vengono suddivisi per sensibilità. Valgono le seguenti due regole: 1) un utente può leggere solo dati che hanno un livello di sensibilità minore del livello di clearance del soggetto stesso e può scrivere solo dati che hanno un livello di sensibilità $\geq$ del livello di clearance del soggetto stesso. Vengono inoltre definite delle classi gerarchiche.
- Applicazioni di sistemi MAC nei DBMS avvengono nel seguente modo: a) l'autenticazione viene affidata ad una TCB (trusted computing base). Si può dire, più alla polimerizzazione della tabella, una propria classificazione di sicurezza. Si definiscono allora delle HLR (Multi Level Relations) caratterizzate da un TE (intorno al livello della riga). L'utente può accedere alla HLR se il suo livello $\geq$ TE della riga.

Mechanismi di sicurezza intrasistemi → si tratta di disposizioni in grado di analizzare i pacchetti in ingresso/uscita dalla rete o di analizzare il comportamento online per rilevare eventuali irregolarità o violazioni. I meccanismi di sicurezza intrasistemi sono principalmente due.

• **Firewall** → firewall sono disposizioni posti nel punto di connessione tra una rete filtrata da una rete esterna. Il compito dei firewall è quello di bloccare eventi quali pacchetti malvoli. Esistono principalmente due tipi di firewall a seconda che il controllo venga a livello di rete o a livello applicativo:

- Screening router → sono dispositivi che effettuano il filtraggio a livello di rete. Vi sono due tipi di screening router: packet filtering → eseguono il filtraggio attraverso la consultazione del header dei pacchetti verificando che sono rispettati dalle regole imposte (es.: indirizzo mittente compatibile con indirizzo desiderato), packet inspection → eseguono il filtraggio analizzando anche il contenuto del messaggio anziché in grado di comprendere una grande varietà di protocolli che vengono incorporati nel protocollo IP.

- Application gateway → sono dispositivi che agiscono a livello applicativo. In particolare eseguono due compiti principali: 1) oltre alla analisi di packet filtering e packet inspection eseguono anche una analisi a livello applicativo, specificando a: 1) i protocolli e i messaggi, 2) il server, il ruolo di application proxy, cioè agisce da intermediario tra client e server, seguendo il ruolo dell'uno o dell'altro a seconda della fase della comunicazione.

Anche zone protette e screening: di comune delle soluzioni si ha che gli screening router sono più veloci in termini di performance, ma soffrono di una elevata difficoltà nella definizione delle regole di filtraggio. Gli Application gateway invece offrono la miglior flessibilità per quanto riguarda la definizione delle regole di filtraggio, ma presentano performance inferiori.

Nella protezione della rete, stesso si usa una configurazione con doppio firewall. La zona interposta tra i due firewall è detta **Demilitarized Zone**: nel caso di un attacco, sarà questa la zona esposta ai tentativi malvoli.

IDS (Intrusion Detection System) → sono sistemi che analizzano il comportamento del sistema al fine di rilevare eventuali anomalie nel suo funzionamento. Le componenti architetturali di un IDS sono: sensori in grado di rilevare gli eventi nel sistema; algoritmi in grado di prendere decisioni sulla base degli eventi registrati; componenti di alerting/registri che interpretano le azioni conseguenti alla rilevazione di un'anomalia. Gli IDS si possono classificare in base a:

• **Modalità di registrazione degli eventi:**

- Network Based IDS → sono sistemi che si basano sulla presenza di sensori sparsi nella rete dell'organizzazione e che analizzano il traffico di rete. I vantaggi di tale approccio sono la semplicità di implementazione e la scarsa impatto sulle performance, i costi consistono nella complessa gestione di rilevazione e mitigazione.
- Host Based IDS → richiedono di essere installati su ogni host della rete ed operano in maniera simile ad un antivirus. Gli Host Based IDS analizzano i pacchetti a livello applicativo e, nel caso del rilevamento di un attacco, sono in grado di fornire informazioni molto dettagliate su quest'ultimo. Il loro svantaggio di tale implementazione è l'elevato impatto sulle performance.

• **Modalità di rilevazione degli attacchi:**

- misuse detection IDS → in tale tipologia di IDS gli attacchi vengono individuati sulla base di un repository di intrusioni note. Il problema di questi IDS è che non sono in grado di rilevare attacchi non noti e che producono una numerosa variazione di falsi positivi. Inoltre il massiccio aggiornamento di frequente il repository degli attacchi noti.
- anomaly detection IDS → in tale tipo di IDS si assume che "modelli di normale funzionamento del sistema" sulla base di dati storici e si impara da nuove di commistibilità. Se il comportamento del sistema esce dalle soglie prestabilite, l'IDS notifica una potenziale intrusione. Questo tipo di IDS può riconoscere attacchi non comuni noti, ma ha il difetto di notificare molti falsi positivi.

Le azioni da un potenziale attacco:

- IDS passivi → inviano soltanto una notifica all'amministratore del sistema.
- IDS attivi → oltre a notificare l'intrusione, eseguono azioni come bloccare le connessioni con l'attaccante o reimpaginare il firewall.

