



# Fondamenti di Sistemi informativi

per il Settore dell'Informazione<sup>1</sup>

Cinzia Cappiello, Mariagrazia Fugini, Paul Grefen,  
Barbara Pernici, Pierluigi Plebani, Monica Vitali

18 settembre 2018

<sup>1</sup>Quest'opera è rilasciata nei termini della licenza Creative Commons Attribuzione – Condividi Allo Stesso Modo 3.0 Italia il cui testo è disponibile alla pagina Internet <http://creativecommons.org/licenses/by-sa/3.0/it/>



# Indice

|   |           |
|---|-----------|
| <b>I Definizioni, tecnologie e progettazione</b>              | <b>9</b>  |
| <b>1 Introduzione</b>   | <b>11</b> |
| 1.1 Dati e informazioni . . . . .                             | 12        |
| 1.2 Le organizzazioni . . . . .                               | 15        |
| 1.2.1 Risorse . . . . .                                       | 15        |
| 1.2.2 Processi . . . . .                                      | 16        |
| 1.3 Definizione e ruolo di sistema informativo . . . . .      | 19        |
| 1.4 Classificazione dei sistemi informativi . . . . .         | 20        |
| 1.4.1 Sistemi OLTP e OLAP . . . . .                           | 22        |
| 1.5 Organizzazione, IT e componenti di un SI . . . . .        | 25        |
| 1.6 Presentazione della struttura del libro . . . . .         | 28        |
| 1.7 Domande . . . . .   | 29        |
| <b>2 La progettazione di sistemi informativi</b>              | <b>31</b> |
| 2.1 Enterprise Architecture . . . . .                         | 31        |
| 2.2 Il processo di gestione del sistema informativo . . . . . | 37        |
| 2.3 Pianificazione . . . . .                                  | 39        |
| 2.3.1 Pianificazione strategica . . . . .                     | 40        |
| 2.3.2 Studio di fattibilità . . . . .                         | 42        |
| 2.4 Ciclo di vita di sviluppo del SI . . . . .                | 46        |
| 2.5 Domande . . . . .   | 48        |
| <b>II Un approccio architetturale ai sistemi informativi</b>  | <b>49</b> |
| <b>3 Approccio BOAT</b>                                       | <b>51</b> |
| 3.1 Descrizione . . . . .                                     | 51        |
| 3.2 Esempi . . . . .  | 54        |
| 3.2.1 Perfect Office Solutions House (POSH) . . . . .         | 54        |
| 3.2.2 TalkThroughUs (TTU) . . . . .                           | 55        |
| <b>4 Prospettiva di business</b>                              | <b>59</b> |
| 4.1 Descrizione . . . . .                                     | 59        |
| 4.2 Dimensioni dell'interazione . . . . .                     | 59        |

|          |   |            |
|----------|---|------------|
| 4.2.1    | Partecipanti (party) . . . . .  | 60         |
| 4.2.2    | Oggetti dell'interazione . . . . .  | 61         |
| 4.2.3    | Orizzonte temporale . . . . .   | 61         |
| 4.3      | Business driver . . . . .   | 62         |
| 4.3.1    | Tipi di driver . . . . .  | 63         |
| 4.4      | Driver e loro attuazione . . . . .  | 66         |
| 4.4.1    | Driver operazionalizzati e direzioni di business . . . . .                | 66         |
| 4.4.2    | Direzioni di business . . . . .   | 67         |
| 4.5      | Partecipanti e scenari . . . . .  | 71         |
| 4.6      | Tabella per il modello di business . . . . .                              | 72         |
| 4.7      | Esempi . . . . .  | 72         |
| 4.7.1    | POSH . . . . .  | 72         |
| 4.7.2    | TTU . . . . .   | 74         |
| <b>5</b> | <b>Prospettiva organizzativa</b>  | <b>77</b>  |
| 5.1      | Descrizione . . . . .   | 77         |
| 5.2      | Aspetti organizzativi inter-organizzazione . . . . .                      | 77         |
| 5.2.1    | Parti all'interno di uno scenario di e-business . . . . .                 | 78         |
| 5.2.2    | Raffinamento di intermediari e di canali . . . . .                        | 80         |
| 5.3      | Relazioni organizzative intra-strutture . . . . .                         | 82         |
| 5.3.1    | Front end e back end . . . . .  | 82         |
| 5.3.2    | Funzionalità front end e canali . . . . .                                 | 82         |
| 5.3.3    | Il concetto di mid office . . . . .                                       | 85         |
| 5.3.4    | Dalla struttura organizzativa ai processi inter-organizzazione . . . . .  | 86         |
| 5.4      | Modellazione dei processi . . . . .                                       | 89         |
| 5.4.1    | Definizioni . . . . .   | 89         |
| 5.4.2    | Modellazione di processo in BPMN . . . . .                                | 90         |
| 5.5      | Identificazione dei processi . . . . .                                    | 95         |
| 5.5.1    | Check-list settoriali . . . . .   | 96         |
| 5.5.2    | Catena del valore di Porter . . . . .                                     | 97         |
| 5.5.3    | Approccio Analitico . . . . .   | 98         |
| 5.6      | Operazionalizzazione dei business driver . . . . .                        | 99         |
| 5.7      | Esempi . . . . .  | 100        |
| 5.7.1    | POSH . . . . .  | 100        |
| 5.7.2    | TTU . . . . .   | 103        |
| <b>6</b> | <b>Architetture funzionali</b>  | <b>105</b> |
| 6.1      | Descrizione . . . . .   | 105        |
| 6.2      | Livelli di aggregazione e astrazione . . . . .                            | 106        |
| 6.2.1    | La dimensione aggregazione . . . . .                                      | 106        |
| 6.3      | La dimensione astrazione . . . . .  | 107        |
| 6.3.1    | Architetture di riferimento, standard e istanze di architetture . . . . . | 108        |
| 6.4      | Architetture Market-level . . . . .                                       | 109        |
| 6.5      | Architettura Party-level . . . . .  | 112        |

|            |  |            |
|------------|--|------------|
| 6.5.1      | Struttura delle architetture Party-level . . . . .                   | 112        |
| 6.5.2      | Architetture Party-level e Enterprise Architecture . . . . .         | 114        |
| 6.6        | Architettura System-level . . . . .                                  | 115        |
| 6.6.1      | La struttura delle architetture System-level . . . . .               | 116        |
| 6.7        | Relazioni tra architetture organizzative e architetture funzionali . | 117        |
| 6.8        | Esempi . . . . .   | 118        |
| 6.8.1      | POSH . . . . .   | 120        |
| 6.8.2      | TTU . . . . .  | 120        |
| <b>7</b>   | <b>Introduzione agli aspetti tecnologici</b>                         | <b>123</b> |
| 7.1        | Descrizione . . . . .  | 123        |
| 7.2        | La tecnologia a livello applicativo . . . . .                        | 123        |
| 7.3        | La tecnologia a livello di piattaforma . . . . .                     | 125        |
| 7.4        | La tecnologia a livello di architettura fisica . . . . .             | 126        |
| 7.5        | Domande sulla Parte II . . . . .                                     | 127        |
| <b>III</b> | <b>Aspetti tecnologici</b>   | <b>129</b> |
| <b>8</b>   | <b>Tecnologie a livello applicativo</b>                              | <b>131</b> |
| 8.1        | Componenti funzionali . . . . .                                      | 131        |
| 8.2        | ERP . . . . .  | 132        |
| 8.2.1      | Storia degli ERP . . . . .   | 132        |
| 8.2.2      | Proprietà degli ERP . . . . .  | 133        |
| 8.2.3      | Funzionalità di un ERP . . . . .                                     | 134        |
| 8.2.4      | Aspetti architetturali . . . . .                                     | 135        |
| 8.2.5      | Implementazione di un ERP . . . . .                                  | 137        |
| 8.2.6      | Benefici degli ERP . . . . .   | 138        |
| 8.3        | CRM . . . . .  | 138        |
| 8.3.1      | CRM operativo . . . . .  | 141        |
| 8.3.2      | CRM analitico . . . . .  | 144        |
| 8.3.3      | CRM collaborativo . . . . .  | 144        |
| 8.3.4      | Implementazione del CRM . . . . .                                    | 145        |
| 8.4        | Data Warehousing . . . . .   | 145        |
| 8.4.1      | Caratteristiche dei dati . . . . .                                   | 147        |
| 8.4.2      | Data warehouse . . . . .   | 149        |
| 8.4.3      | Architettura del data warehouse . . . . .                            | 150        |
| 8.4.4      | Modello concettuale del data warehouse . . . . .                     | 153        |
| 8.4.5      | Modelli logici del data warehouse . . . . .                          | 154        |
| 8.4.6      | Operazioni sul data warehouse . . . . .                              | 157        |
| 8.4.7      | Ciclo di vita del data warehouse . . . . .                           | 158        |
| 8.5        | Data Mining . . . . .  | 159        |
| 8.5.1      | Regole Associate . . . . .   | 161        |
| 8.5.2      | Classificazione . . . . .  | 162        |
| 8.5.3      | Clustering . . . . .   | 166        |
| 8.6        | Esempi su CRM . . . . .  | 168        |

|  |            |
|--|------------|
| 8.7 Selezione del software . . . . .                               | 169        |
| 8.8 Domande . . . . .  | 172        |
| <b>9 Tecnologie a livello di piattaforma</b>                       | <b>173</b> |
| 9.1 Architetture di integrazione . . . . .                         | 173        |
| 9.1.1 Integrazione dei dati . . . . .                              | 174        |
| 9.1.2 Integrazione dei processi . . . . .                          | 175        |
| 9.2 Evoluzione dei moduli applicativi . . . . .                    | 176        |
| 9.3 Piattaforme di integrazione . . . . .                          | 179        |
| 9.3.1 Architettura di integrazione punto-a-punto . . . . .         | 179        |
| 9.3.2 Architettura di integrazione hub-and-spoke . . . . .         | 179        |
| 9.3.3 Workflow component e workflow management system . .          | 181        |
| 9.4 Modelli a servizi . . . . .                                    | 183        |
| 9.4.1 Composizione e orchestrazione di servizi . . . . .           | 186        |
| 9.4.2 Selezione dinamica dei servizi di business . . . . .         | 186        |
| 9.4.3 Integrazione dei servizi . . . . .                           | 187        |
| 9.5 Domande . . . . .  | 189        |
| <b>10 Tecnologie a livello di architettura fisica</b>              | <b>191</b> |
| 10.1 Descrizione . . . . .   | 191        |
| 10.1.1 Evoluzione delle architetture fisiche . . . . .             | 191        |
| 10.2 Layer applicativi . . . . .                                   | 193        |
| 10.3 Applicazioni distribuite . . . . .                            | 194        |
| 10.3.1 Architetture per componenti applicativi: ERP . . . .        | 199        |
| 10.4 Scalabilità . . . . .   | 200        |
| 10.4.1 Server Farm . . . . .                                       | 201        |
| 10.5 Virtualizzazione . . . . .                                    | 205        |
| 10.6 Cloud Computing . . . . .                                     | 207        |
| 10.6.1 Caratteristiche del Cloud Computing . . . . .               | 207        |
| 10.6.2 Modelli di servizio . . . . .                               | 208        |
| 10.6.3 Modelli di deployment . . . . .                             | 209        |
| 10.6.4 Componenti architetturali BOAT in Cloud . . . . .           | 209        |
| 10.7 Opzioni di gestione dei SI . . . . .                          | 210        |
| 10.8 Domande . . . . .   | 213        |
| <b>11 Sicurezza dei sistemi informativi</b>                        | <b>215</b> |
| 11.1 Proprietà di sicurezza . . . . .                              | 215        |
| 11.2 Minacce, violazioni, vulnerabilità, attacchi . . . . .        | 216        |
| 11.3 Crittografia per la protezione dei dati . . . . .             | 219        |
| 11.3.1 Crittografia simmetrica . . . . .                           | 220        |
| 11.3.2 Crittografia asimmetrica . . . . .                          | 222        |
| 11.3.3 Integrità e funzione di hash . . . . .                      | 224        |
| 11.3.4 Firma digitale . . . . .                                    | 225        |
| 11.3.5 Aspetti di gestione delle chiavi e certificati digitali . . | 226        |
| 11.4 Gestione degli utenti e controllo degli accessi . . . . .     | 229        |
| 11.4.1 Autenticazione e autorizzazione . . . . .                   | 229        |

|  |            |
|--|------------|
| 11.4.2 Sistemi di controllo di accesso ai dati . . . . .                                   | 231        |
| 11.5 Meccanismi di sicurezza infrastrutturali . . . . .                                    | 240        |
| 11.5.1 Firewall . . . . .  | 240        |
| 11.5.2 Intrusion Detection Systems - IDS . . . . .   | 244        |
| 11.6 Domande . . . . .   | 247        |
| <b>12 ArchiMate</b>  | <b>249</b> |
| 12.1 Introduzione ad ArchiMate . . . . .   | 249        |
| 12.2 Elementi principali del modello . . . . .   | 251        |
| 12.2.1 Componenti . . . . .  | 251        |
| 12.2.2 Relazioni . . . . .   | 252        |
| 12.2.3 Nodi . . . . .  | 253        |
| 12.2.4 Business layer . . . . .  | 253        |
| 12.2.5 Application layer . . . . .   | 255        |
| 12.2.6 Livello tecnologico . . . . .   | 257        |
| 12.3 Esempi . . . . .  | 258        |
| 12.3.1 Esempio di rappresentazione di un sistema di composizione di un documento . . . . . | 258        |
| 12.3.2 Esempio di servizio assicurativo . . . . .  | 259        |
| 12.4 Rappresentazione dell'infrastruttura tecnologica . . . . .                            | 261        |
| 12.4.1 Componenti per l'infrastruttura tecnologica . . . . .                               | 262        |
| 12.4.2 Relazioni derivate . . . . .  | 263        |
| 12.4.3 Gruppi . . . . .  | 263        |
| 12.5 Rappresentare le scelte architetturali in ArchiMate . . . . .                         | 265        |
| 12.5.1 Architetture N-tier . . . . .   | 265        |
| 12.5.2 Virtualizzazione . . . . .  | 268        |
| <b>IV Scenari</b>  | <b>271</b> |
| <b>13 Sistemi informativi e nuove tecnologie</b>   | <b>273</b> |
| 13.1 Innovazioni tecnologiche . . . . .  | 273        |
| 13.2 Industria 4.0 . . . . .   | 275        |
| 13.3 Sistemi informativi per la pubblica amministrazione . . . . .                         | 279        |
| <b>V Bibliografia e indice analitico</b>   | <b>283</b> |



## **Parte I**

# **Definizioni, tecnologie e progettazione**



# Capitolo 1

## Introduzione

Nelle attuali realtà aziendali, ma anche organizzative e sociali in genere, caratterizzate da una quantità grande, e sempre crescente, di dati e informazioni, si assiste a una crescente interdipendenza tra il sistema informativo aziendale (sistema che ne gestisce le informazioni) e la capacità di operare dell'azienda stessa. Le modifiche di strategia, regole e processi aziendali (Business Process) richiedono sempre di più cambiamenti nell'hardware, nel software, nelle basi di dati, nelle reti di comunicazioni e a livello di interfacce utente. Spesso, ciò che l'organizzazione può fare oggi dipende dalle funzionalità rese disponibili dal suo sistema informativo. Internet e il World Wide Web (Web nel seguito) sono diventati uno dei fondamenti su cui le organizzazioni basano i loro sistemi informativi per costruire un'economia centrata sull'informazione. Fra i vari metodi per rispondere alle sfide di un'economia globale vi è quello di acquisire sempre più informazioni rilevanti per l'organizzazione oltre a valorizzare quelle già in possesso. Le piattaforme e le applicazioni messe a disposizione nel mondo Web sono una buona risposta in quanto permettono di raccogliere, gestire in modo correlato e distribuire in vari formati una grande quantità di informazioni provenienti da sorgenti informative eterogenee. Spesso tali sorgenti per loro stessa natura sono distribuite concettualmente e fisicamente, come accade per esempio in organizzazioni multinazionali, nelle pubbliche amministrazioni, e in ambienti industriali, commerciali, finanziari, di ricerca. In tali ambienti, l'accesso immediato ed efficace ai dati importanti, estratti dalla grande mole di dati disponibili, diventa critico e strategico. Le fusioni aziendali, le acquisizioni, e le partnership costituiscono un'altra forte motivazione per integrare sia le informazioni sia i processi che le gestiscono.

In questo contesto, i sistemi informativi possono definirsi come strumenti supportati dall'Information Technology (IT) e composti da applicazioni software, sistemi di gestione dati, interfacce utente (oggigiorno in prevalenza di tipo Web) e reti di comunicazione atti a supportare le organizzazioni nella gestione dell'informazione. I sistemi informativi presentano vari aspetti di complessità sia progettuale sia di controllo e di gestione. Tali aspetti sono dovuti alla presenza di varie fonti informative, di basi di dati eterogenee e differenziate per modelli

e tecnologie, di reti di comunicazione basate su vari protocolli e sono inoltre dovuti alla presenza di processi e attori aziendali distribuiti e interconnessi nella cooperazione tra entità organizzative diverse. Inoltre, il sistema informativo 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.

Per affrontare tali complessità, oggi la chiave di successo consiste nel progettare e realizzare sistemi informativi distribuiti, ovvero costituiti da applicazioni e dati risiedenti su vari nodi elaborativi, che permettono a diverse organizzazioni di lavorare in sinergia per gestire il carico applicativo e l'informazione in rete in maniera interconnessa, ma al tempo stesso mantenendo un certo livello di autonomia tra le parti. Questa visione si contrappone con quella che propone i sistemi informativi centralizzati, spesso basati su mainframe, che fino a pochi decenni fa costituivano la soluzione principale e che, nonostante la loro obsolescenza, sono ancora oggi parte del patrimonio hardware e software di diverse organizzazioni, rendendo quindi la prospettata sinergia con l'esterno molto difficile da ottenere. 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.

Governare la complessità che si viene a creare non è un compito semplice. Per questo motivo, sono stati proposti metodologie, metodi, tecniche e strumenti di supporto alla progettazione e realizzazione di un sistema informativo come sistema in grado di fornire ad un'organizzazione che vuole definirsi in ‘rete’ metodi adatti per governare sia i processi organizzativi interni che quelli per cui è necessaria un’interazione con altre organizzazioni.

## 1.1 Dati e informazioni

Prima di addentrarsi nella definizione di un sistema informativo e dei suoi componenti, è utile chiarire il significato di alcuni termini che solitamente sono considerati sinonimi, ma che in realtà nell’ambito dei sistemi informativi hanno una specifica, forte caratterizzazione. Infatti il termine ‘dato’ e il termine ‘informazione’ sono a torto considerati sinonimi. Al fine di illustrare le differenze, si prenderà a riferimento la cosiddetta piramide della conoscenza, detta anche piramide DIKW (Data Information Knowledge Wisdom), mostrata in Figura 1.1.

Si può notare come il dato rappresenti la base di tutto. La parola *dato* infatti deve essere interpretata come il participio passato del verbo “dare”. 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. Ad esempio, se si vuole rappresentare la numerosità di una classe di

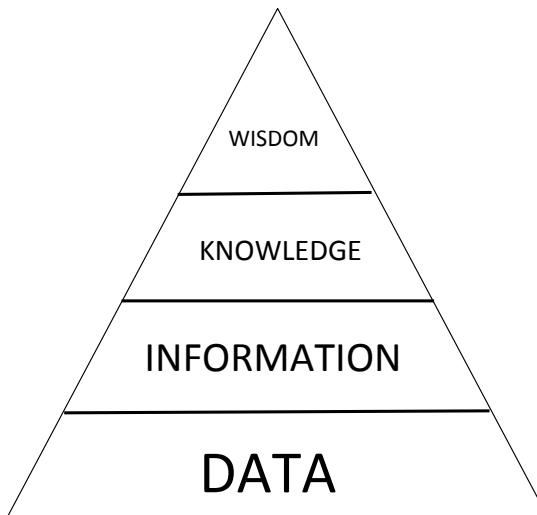


Figura 1.1: Piramide della conoscenza (fonte Wikipedia)

studenti, il tipo di dato deve essere necessariamente numerico o, per essere più precisi, deve essere un intero naturale, poiché non è corretto rappresentare questa porzione di realtà usando numeri negativi, tanto meno utilizzando numeri con decimali. Oltre al tipo, un dato può essere caratterizzato dalla sua unità di misura. Ad esempio, volendo rappresentare la temperatura in un'aula, oltre ad associare al dato un tipo numerico - questa volta un numero reale - va altresì indicata la scala di misura (per esempio, gradi Celsius o gradi Fahrenheit). Siccome i dati raccolti possono essere numerosi, la disciplina delle *basi di dati* definisce metodi e strumenti per organizzarli all'interno di strutture più o meno complesse e più o meno flessibili in grado di rendere agevole e ottimale non solo la loro memorizzazione, ma anche il loro recupero. Infatti, lo scopo di una base di dati è rappresentare una porzione della realtà attraverso la memorizzazione di un insieme di dati che riflettono tale realtà. Questo è suggerito dal primo modello utilizzato per la progettazione di una base di dati e, cioè, il modello concettuale, il cui scopo è appunto rappresentare ad alto livello la realtà per poter poi, attraverso trasformazioni tra modelli, riuscire a codificarle all'interno di un sistema informatico, ovvero l'insieme di componenti hardware, software e di telecomunicazione, detto anche sistema IT che supporta il funzionamento del sistema informativo (vedremo la definizione di sistema informativo, informatico e sistema IT più avanti nel volume).

Seguendo lo schema suggerito dalla piramide della conoscenza, al di sopra del dato si posiziona l'*informazione*. Come detto in precedenza, molto spesso questi due termini sono spesso usati come sinonimi. In realtà, 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. Infatti, un dato preso singolarmente, senza alcun contesto di riferimento, spesso resta inutile. Per

esempio, disporre di un dato di temperatura senza sapere se questa temperatura si riferisce a un ambiente interno, ad esempio un'aula, oppure ad un ambiente esterno non ha alcun senso. Per questo motivo, 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. Per esempio, è utile associare al numero di studenti l'anno accademico cui si fa riferimento e alla temperatura la data di rilevazione e l'aula in cui la rilevazione è avvenuta. L'informazione si può quindi definire come l'output di interrogazioni rivolte ad un insieme di dati (presumibilmente organizzate in una base di dati), come ad esempio “quanti gradi ci sono oggi in aula magna?” oppure “qual è la media degli studenti iscritti negli ultimi cinque anni a ingegneria meccanica?”. Continuando a risalire la piramide, sopra l'informazione si trova la *conoscenza* (*knowledge*) che è ottenibile integrando l'informazione con l'esperienza. Sapere infatti che a metà dicembre nell'aula magna vi sono 32 gradi suggerisce che vi sia qualche anomalia in quanto tale valore si discosta da quanto l'esperienza suggerisce, ovvero dalla media di temperature solitamente osservate in quel periodo. Mentre l'informazione permette di rappresentare la realtà in modo completo e utile, la conoscenza guida eventuali decisioni che possono influire sulla situazione attuale. Va sottolineato come la conoscenza viene acquisita nel tempo, attraverso l'esperienza da un soggetto, e solitamente non può essere trasferita ad altri soggetti. Al contrario, si possono trasferire informazioni o, al massimo, esperienza con il problema però di riuscire a formalizzare quest'ultima. Ad esempio, che intorno temporale va considerato per il calcolo della serie storica, rispetto alla misura della temperatura osservata? Solo lo stesso giorno? Una settimana, un mese? Infine, vi è la *saggezza* (*wisdom*) che rappresenta un'estensione della classica piramide della conoscenza (il cui nome suggerisce che al vertice ci sia, appunto, la 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. Tornando all'esempio, l'aver capito - attraverso il confronto tra la temperatura attuale dell'aula magna e la serie storica delle temperature - che vi è una anomalia, può essere motivo per spegnere il riscaldamento in quanto tale azione ha influenza sulla temperatura.

L'utilizzo della piramide, quale figura geometrica per la rappresentazione di questi concetti non è casuale. Innanzitutto, offre una stratificazione a livelli, evidenziando come i livelli superiori sono costruiti a partire dai livelli inferiori. In secondo luogo, l'area ricoperta da ogni livello diminuisce con l'aumento di livello. Questo vuole rappresentare il livello di *sintesi*: in basso abbiamo i dati che sono molto numerosi e a un livello di granularità fine, mentre al lato opposto abbiamo la saggezza rappresentata con un numero minore di elementi e quindi molto sintetica e a granularità grossa.

Dal punto di vista della gestione dei dati in un sistema informatico (ovvero la piattaforma IT), al crescere del livello diminuisce la capacità di gestione i 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

L'uso di sistemi informativi richiede la comprensione dell'organizzazione, della sua gestione e del ruolo dell'IT in azienda. Un sistema informativo crea infatti valore per l'azienda in termini di soluzioni organizzative e gestionali alle sfide poste dall'ambiente in cui opera. Senza entrare nel dettaglio di materie quali *teoria delle organizzazioni* o *organizzazione aziendale*, per i quali si rimanda a libri di testo specifici, 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. Fra le diverse classificazioni, si ricorda quella di Ricciardi, citata in [7] in cui le risorse sono classificate come:

- 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.

Le risorse hanno ciascuna un proprio ciclo di vita che generalmente comprende le fasi di Pianificazione, Acquisizione, Gestione e Manutenzione. Esempi di risorse sono: il personale, il denaro, i macchinari, e ovviamente i dati e informazioni che rappresentano l'elemento basilare di un sistema informativo.

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. L'informazione può anche essere, per alcune aziende, un prodotto. L'informazione ha le seguenti caratteristiche che la distinguono da qualsiasi altra risorsa aziendale: è intangibile, non deperibile (non viene distrutta con l'uso), auto-rigenerante (il suo utilizzo porta alla generazione di nuova informazione). Per esempio, nelle comunicazioni mobili, l'informazione serve per ottenere informazioni sugli utenti, proporre nuove tariffe, ottenere feedback su iniziative di marketing, aggiornare

le tariffe. In una linea di produzione, occorre memorizzare informazioni sui lavori effettuati (per esempio, tempo richiesto, percentuali di pezzi danneggiati), identificare problemi o migliorare l'esecuzione del processo. Gestire l'informazione (e questo vedremo essere l'obiettivo di un sistema informativo) si traduce quindi in innumerevoli attività quali:

- Creare informazione.
- Acquisire informazione.
- Elaborare informazione.
- Archiviare informazione.
- Trasmettere informazione.
- Presentare informazione.

Queste attività non richiedono necessariamente strumenti IT. L'informazione può essere gestita in modo implicito, ovvero basato su esperienza e competenze dei singoli, quando si tratta di attività difficilmente replicabili, oppure in modo esplicito, ma non supportato da IT (gestione manuale). Oppure ancora in modo esplicito e supportato da IT, ovvero in modo che consente di organizzare e reperire l'informazione in modo efficiente e facilmente replicabile.

### 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). I processi sono classificabili secondo diversi modelli di classificazione:

- la piramide di Anthony.
- la catena del valore di Porter.

**Classificazione dei processi secondo la piramide di Anthony.** Per dare una classificazione dei processi e delle applicazioni, si può far riferimento al modello gerarchico dell'organizzazione dato dalla *Piramide di Anthony* [2] mostrato in Figura 1.2. Nel modello di Anthony si distinguono tre livelli, legati alla struttura dell'organizzazione: un *livello operativo*, detto anche operazionale, in cui si considerano le attività di tipo operativo dell'azienda, un *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, e un *livello di pianificazione strategica*, dove si collocano le attività legate alla scelta degli obiettivi aziendali e alla definizione delle politiche aziendali.

Esempi di processi presso una pubblica amministrazione, quale un'amministrazione comunale, sono i seguenti (suddivisi per livello):



Figura 1.2: Modello a Piramide di Anthony

- *Operativo*: contabilizzazione dei pagamenti dei cittadini, interventi di manutenzione delle strade.
- *Controllo*: controllo dei pagamenti, solleciti, confronti mensili tra entrate previste ed effettive, monitoraggio inquinamento.
- *Strategico*: verifica dei costi e dei ricavi relativi ai servizi sociali, definizione di nuove tariffe, piani regolatori.

Presso una banca, esempi di processi ai tre livelli sono i seguenti:

- *Operativo*: gestione movimenti dei conti correnti, prelievi bancomat (ATM - Automatic Teller Machine).
- *Controllo*: revisione degli scoperti, concessione di un mutuo.
- *Strategico*: verifica dell'andamento di un servizio, decisione di aprire nuovi servizi.

Presso una azienda, esempi di processi sono i seguenti:

- *Operativo*: registrazione costi delle commesse, gestione magazzino.
- *Controllo*: controllo scostamenti settimanali tra preventivo e consuntivo.
- *Strategico*: scelta delle aree di mercato più convenienti.



Figura 1.3: Modello di Porter

**Classificazione dei processi secondo il modello di Porter.** Un secondo tipo di classificazione molto adottato in letteratura si basa sulla *catena del valore* (*value chain*) proposto da Porter [32] (Figura 1.3) che considera esclusivamente i processi di livello operazionale. Il modello value chain considera i processi e le *funzionalità* ad essi collegate come lo strumento a disposizione di un'organizzazione per dare valore ai prodotti e ai servizi rilasciati all'esterno. Sebbene il modello sia stato proposto per classificare i processi in aziende manifatturiere, i principi su cui si basa possono essere applicati anche ad altre organizzazioni.

Secondo questa classificazione, le funzionalità sono suddivise in *attività primarie*<sup>1</sup>, 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 di Figura 1.3. L'attività di *logistica in entrata* abilita il trasporto di materiali usati come input per la produzione. L'attività di *operations* esegue la produzione dei prodotti secondo quello che è la natura dell'azienda. L'attività di logistica di uscita abilita il trasporto delle merci prodotte ai clienti destinatari delle merci. Infine, l'attività di *marketing e vendite* punta ad attrarre i clienti e ad organizzare le vendite, mentre l'attività *servizi* è responsabile dei servizi post-vendita per i clienti.

I processi secondari con le relative funzioni sono mostrati come strati orizzontali nella parte superiore della Figura 1.3. La funzione *approvvigionamento* (*procurement*) si occupa di ordinare le merci e i servizi che sono necessari all'interno delle altre funzioni (in particolare, per la funzione di produzione operations). Come suggerisce il nome, la funzione *sviluppo della tecnologia* (*technology development*) si concentra sullo sviluppo di nuove tecnologie utilizzate nelle funzioni operations o nelle merci prodotte. La funzione *gestione delle risorse*

<sup>1</sup>si noti che in questo caso i termini attività e processo sono usati come sinonimi. In realtà, come si discuterà più avanti nel corso i due termini hanno significati distinti.

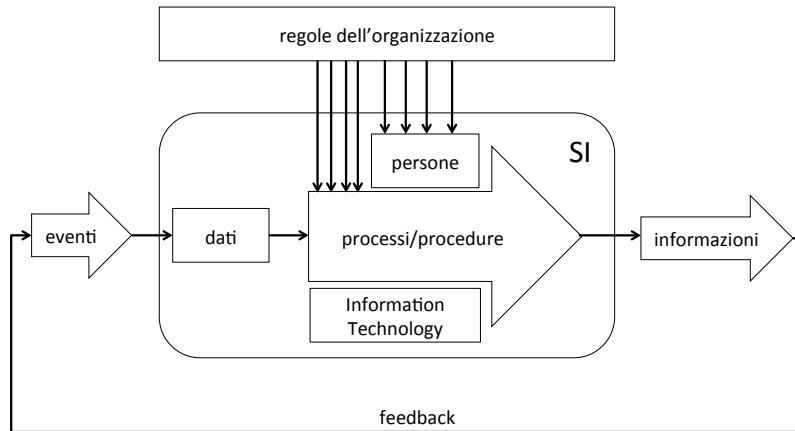


Figura 1.4: Rappresentazione di un sistema informativo (rielaborazione da [31])

umane (*Human Resource - HR - management*) gestisce il personale dell'organizzazione. Infine, la funzione *Attività infrastrutturali* è responsabile di fornire l'infrastruttura generale dell'organizzazione, ad esempio la parte amministrativa finanziaria.

### 1.3 Definizione e ruolo di sistema informativo

Dopo aver chiarito cosa è l'informazione e come le informazioni sono fondamentali per un'organizzazione, possiamo definire in modo preciso cosa è un sistema informativo. Per fare ciò ci ispiriamo alla definizione data in [31] che parte dal fatto che un sistema informativo è innanzitutto un *sistema* (si veda Figura 1.4). È quindi un insieme di procedure, metodi e strumenti dedicati allo svolgimento di alcune funzioni per il raggiungimento di un dato risultato. Nel caso specifico dei sistemi informativi, tale sistema si alimenta di eventi, rappresentati tramite dati, che accadono in un'organizzazione e li trasforma in informazioni, le quali devono essere opportunamente modellate e rappresentate a seconda di chi poi le utilizzerà. Eventi possono essere, in un'azienda manifatturiera, l'arrivo di un ordine, la consegna di un prodotto finito così come la segnalazione di esaurimento di un prodotto a magazzino. Questi eventi possono attivare i processi organizzativi (per esempio, l'avvio della produzione dell'ordine, la notifica al corriere per la consegna o l'invio dell'ordine di acquisto per il materiale esaurito) che, come visto in precedenza, rappresentano la base di un'organizzazione. Al fine di supportare questa trasformazione:

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

All'interno di un sistema informativo troviamo le basi di dati che 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. Ortagonalmente a questi due aspetti, 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, come evidenziato nel precedente paragrafo, un'organizzazione è costituita da processi e ruoli differenti ognuno dei quali può richiedere dati e informazioni a diverso livello di dettaglio. Per questo motivo, 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.

Un sistema informativo è quindi qualcosa di complesso che erroneamente è spesso associato al solo aspetto tecnologico che, in realtà ne rappresenta una porzione detta *sistema informatico*. In particolare, 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. Il sistema IT cambia nel tempo con l'evolvere dell'azienda e delle tecnologie. Quindi, i punti chiave del sistema informativo diventano la gestione della *risorsa informazione* secondo regole e obiettivi aziendali (business rules) con l'utilizzo delle tecnologie per una (o più) organizzazioni. Il ruolo dell'informazione è fondamentale in qualsiasi tipo di azienda, da quella manifatturiera a quella di servizi. L'informazione non è uno strumento accessorio, ma un bene strategico che assume importanza esattamente come altre risorse dell'azienda quali le risorse umane, materiali e finanziarie.

La progettazione e lo sviluppo di sistemi informativi include quindi la progettazione di vari aspetti: dati, processi, interazione utente. Lo sviluppo secondo la separazione in aree di interesse del progetto (dati, logica di business e interfaccia utente) è uno dei modi che permettono di gestire la complessità del progetto.

## 1.4 Classificazione dei sistemi informativi

Data la complessità del dominio applicativo di interesse di un sistema informativo (si pensi ai vari processi aziendali per esempio, in un'azienda manifatturiera), non è pensabile avere un singolo sistema informativo in grado di coprire tutti i processi e gestire tutte le risorse organizzative. Per questo motivo, esistono diverse tipologie di sistemi informativi (che possono essere integrati tra di loro) che si focalizzano su qualche (tipo di) processo, o che sono in grado di gestire determinate risorse (per esempio, i documenti, o i dati provenienti da sensori). Una prima classificazione di questi sistemi informativi è possibile prendendo a riferimento la piramide di Anthony e permette di distinguere i sistemi informativi in: sistemi informativi *operazionali* e i sistemi informativi *decisionali* (detti anche Management Information Systems o, da alcuni autori, sistemi informazionali).

- La finalità dei *sistemi operazionali* è di svolgere operazioni di base, quotidiane, di normale amministrazione, quali la gestione di transazioni e di

lavoro di ufficio, la contabilità, in generale la elaborazione di quelle che sono dette le situazioni aziendali [29]. Le parti fondamentali di un sistema informativo operazionale sono la base di dati operazionale (che contiene in forma strutturata l'intera informazione operativa, ovvero a livello operativo nella piramide di Anthony) e le funzioni operative.

- I *sistemi decisionali o informazionali* sono sistemi informativi 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. Con l'aumentare della quantità dei dati disponibili, è anche aumentata la richiesta di estrazione di informazioni strategiche. I sistemi informativi rispondono a questa esigenza con strumenti che si sono evoluti dagli originari strumenti di reportistica o strumenti tipo i fogli elettronici. Oggi, esistono funzionalità sofisticate di estrazione e analisi dati a fini strategici, per cui sono utilizzati anche termini quali 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) - sistemi di supporto alle decisioni, ovvero strumenti di supporto all'estrazione di informazioni per il processo decisionale quali i browser OLAP (OnLine Analytical Processing, si veda la sezione 1.4.1). Più recentemente per indicare in generale funzionalità di questo tipo viene adottato anche il termine Business Intelligence.

La piramide di Anthony offre anche la possibilità di comprendere la natura dei dati gestiti dai diversi sistemi informativi. Infatti, anche i dati sono classificabili a livello operativo/controllo/strategico. Esempi di dati operativi sono gli importi dei versamenti, le ore di presenza dei dipendenti; esempi di dati a livello di controllo sono i saldi mensili, la quantità di lavoro mensile di ciascun reparto; i dati strategici sono i dati macroeconomici, gli indicatori generali, o i dati di budget.

Come mostrato in Figura 1.5, i sistemi informativi operazionali trattano alti volumi di dati che rappresentano in dettaglio gli eventi che accadono nell'organizzazione. Tali dati sono ben strutturati (solitamente memorizzati in basi di dati) e provenienti dall'interno. Spostandosi ai sistemi informativi direzionali i dati sono sempre più aggregati poiché, nelle decisioni prese a questi livelli la sintesi è un requisito necessario. Al tempo stesso anche il formato potrebbe non essere più ben strutturato (vi sono anche documenti scritti in linguaggio naturale) e la loro provenienza è anche dall'esterno.

Vale la pena notare che il grado di strutturazione delle informazioni ha un impatto sul grado di automazione nelle decisioni. Come rappresentato in Figura 1.6, quando un sistema informativo tratta dati ben strutturati e anche il processo decisionale è ben definito (per esempio, vi è un algoritmo) allora il sistema stesso può autonomamente prendere decisioni (per esempio, riordino automatico del magazzino). Quando però il processo di decisione non può più

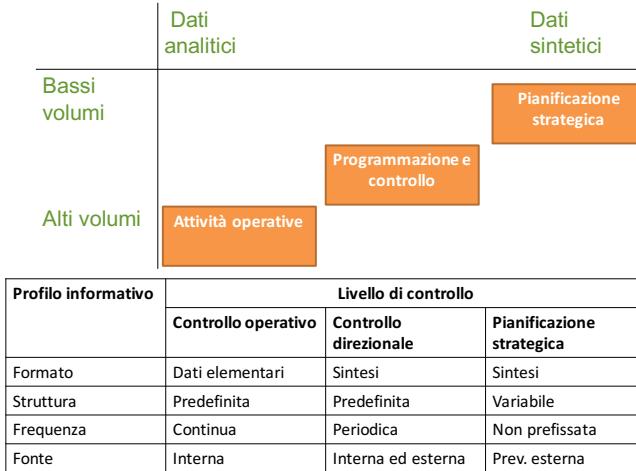


Figura 1.5: Profilo informativo dei sistemi informativi [7]

essere rappresentato in modo formale e la variabile umana gioca ancora un ruolo, il sistema informativo diventa un supporto al decisore. Questo significa che la chiusura dell’anello è rappresentata dall’utente umano. Infine, indipendentemente dalla natura del processo decisionale, avere dati poco strutturati porta a limitare l’impatto nell’uso dei sistemi informativi.

Rispetto a questo punto, val la pena notare, che negli ultimi anni si è avuta una forte spinta nell’adozione di tecniche di Intelligenza Artificiale (IA) all’interno dei sistemi informativi. Questo ha permesso di sviluppare sistemi in grado di rendere possibili decisioni anche in caso di dati non strutturati e processi non ben definiti in quanto, l’IA punta a sostituire l’uomo nelle decisioni.

#### 1.4.1 Sistemi OLTP e OLAP

I sistemi informativi possono essere definiti sinteticamente come un insieme di applicazioni che interagiscono con basi di dati. Le funzioni registrano eventi e/o consultano la base di dati che memorizza le informazioni permanenti del sistema.

Con il termine *transazione* in un sistema informativo ci si riferisce a una categoria molto ampia di concetti che comprende:

- Scambi o contratti fra impresa e ambiente esterno o fra unità della stessa azienda con uno specifico obiettivo (per esempio, ricezione degli ordini dei clienti, invio di ordini ai fornitori ecc.).
- trasformazioni, ossia attività operative che contribuiscono alla produzione di beni e/o all’erogazione di servizi.

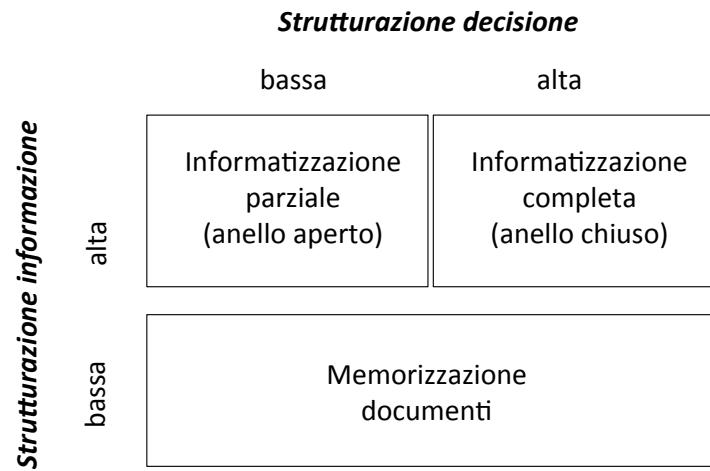


Figura 1.6: Profilo decisionale dei sistemi informativi [7]

- Movimentazioni di oggetti fisici all'interno dell'azienda (per esempio, assemblaggio di una scheda elettronica) o fra l'azienda e il mondo esterno (per esempio, spedizione di un pezzo dal magazzino al cliente).
- Certificazione di eventi, come ad esempio l'inserimento di un nuovo prodotto a catalogo o il pagamento di una fattura.
- A livello della base di dati, un insieme di operazioni elementari che modificano lo stato della base di dati, lasciando i dati in uno stato consistente (per esempio, esecuzione di un bonifico bancario, che comprende varie letture e scritture sulle basi di dati bancarie che alla fine devono terminare correttamente con un 'eseguito' o 'non eseguito'). Questo tipo di transazioni nell'ambito delle tecnologie dei sistemi di gestione di basi di dati (DBMS - DataBase Management System) vengono definite transazioni ACID (Atomiche, Consistenti, Isolate, Durevoli).

Le transazioni sono spesso certificate dalla creazione di un documento, cartaceo o elettronico, che ne testimonia l'avvenuta esecuzione.

I *sistemi OLTP* (*On-line Transaction Processing*) sono quelli che trattano operazioni caratterizzate da un gran numero di transazioni brevi e on-line<sup>2</sup> (che usano operazioni di INSERT, UPDATE, DELETE). I sistemi OLTP 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, per esempio, misurando l'efficienza in base al numero di transazioni al secondo

<sup>2</sup>Si noti che il termine on-line in questo ambito non indica una transazione visibile su Web, ma una transazione i cui effetti sono immediatamente riscontrabili sul sistema. Questo termine evidenzia l'evoluzione rispetto ai sistemi usati in precedenza detti di tipo *batch*, che rendevano visibili gli effetti della transazione solo a fronte di un aggiornamento che avveniva a intervalli regolari (solitamente notturno).

|                 | <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                                 |

Figura 1.7: OLTP e OLAP

(throughput). Nei sistemi di gestione di dati OLTP sono memorizzati dati dettagliati e attuali. Lo schema usato per mantenere le basi di dati transazionali è in genere un modello basato sulle relazioni (generalmente in terza forma normale - 3NF). Le query accedono a record individuali (per esempio, come nell'accesso in modalità Update alla Email di una base di dati aziendale). I sistemi OLTP sono idonei alla gestione di processi a livello operativo e di controllo.

I *sistemi di gestione di dati OLAP (On-line Analytical Processing)* invece trattano dati storici (o d'archivio). Il paradigma OLAP è caratterizzato da poche transazioni e interrogazioni (query) complesse che richiedono aggregazione di dati. I sistemi OLAP hanno come misura di efficienza e di efficacia il tempo di risposta. Le applicazioni OLAP sono usate principalmente nelle tecniche di Data Warehousing. Un sistema OLAP memorizza dati in formato aggregato, memorizza dati storici, archiviati secondo schemi multi-dimensional (generalmente schemi detti a stella). Spesso, le interrogazioni accedono a grandi quantità di dati per rispondere a quesiti complessi come, per esempio, quale è stato il profitto netto realizzato dall'azienda in una certa area geografica nello scorso anno.

I sistemi OLAP sono utilizzati per l'elaborazione di dati orientata al supporto decisionale, quindi sono adeguati a funzionalità collocate a livello di pianificazione e strategico della piramide di Anthony. Essi permettono infatti 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.

Come si vedrà nei capitoli successivi, le tecnologie OLAP sono classificate in base alle strutture utilizzate (MOLAP - Multidimensional OLAP, ROLAP - Relational OLAP o HOLAP - Hybrid OLAP).

In Figura 1.7 è mostrato un confronto fra i sistemi OLTP e OLAP basato sui criteri descritti nel testo.

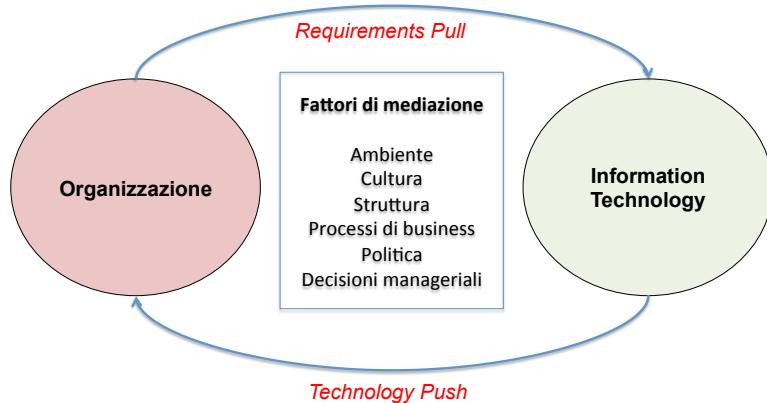


Figura 1.8: Interazione tra organizzazione e IT (rielaborazione da [24])

## 1.5 Organizzazione, IT e componenti di un SI

Tra l'organizzazione e l'IT sussiste una relazione bidirezionale, come mostrato in Figura 1.8. Questa relazione bidirezionale complessa è mediata da vari fattori, fra i quali non ultimo le decisioni prese — o non prese — dai dirigenti. Altri fattori di mediazione sono la cultura organizzativa, la struttura sociale e politica, i processi di business e l'ambiente.

Un aspetto importante in questa relazione tra aspetti tecnologici è che le decisioni prese a livello strategico o di controllo possono avere un impatto sulle scelte tecnologiche (*requirements pull*), mentre in altri casi i cambiamenti degli obiettivi aziendali, delle scelte organizzative e delle attività da svolgere possono essere dovuti a nuove tecnologie che offrono nuove opportunità nel settore di attività dell'azienda o per un suo ampliamento (*technology push*).

La piramide di Anthony è un riferimento anche per considerare aspetti di allineamento tra struttura organizzativa e sistema informativo. Come mostrato in Figura 1.9, la struttura gerarchica permette di rappresentare i flussi di controllo fra livello operativo e livello di controllo. Qui si vede che il livello operativo, nel caso in cui insorga un problema in una unità organizzativa (o in un processo, se si vuole considerare il processo svolto da una unità operativa e dalle persone/utenti coinvolti), genera un'eccezione che altera il normale flusso operativo. Per gestire l'eccezione si possono richiedere informazioni da altri processi/persone (restando al livello operativo) oppure chiedere l'intervento di un livello gerarchico più elevato, spostandosi nel livello di controllo di Anthony. Importante è raggiungere il punto dove risiedono i diritti e le competenze giusti per risolvere le eccezioni. Ciò genera globalmente, anche tramite la delega del controllo, vari flussi informativi relativi alla struttura organizzativa verticale (eccezioni) e orizzontali (necessità di informazioni da altri processi/persone).

La relazione tra le diverse esigenze organizzative è una relazione complessa. Il sistema informativo, come illustrato in Figura 1.4, è un insieme di compo-

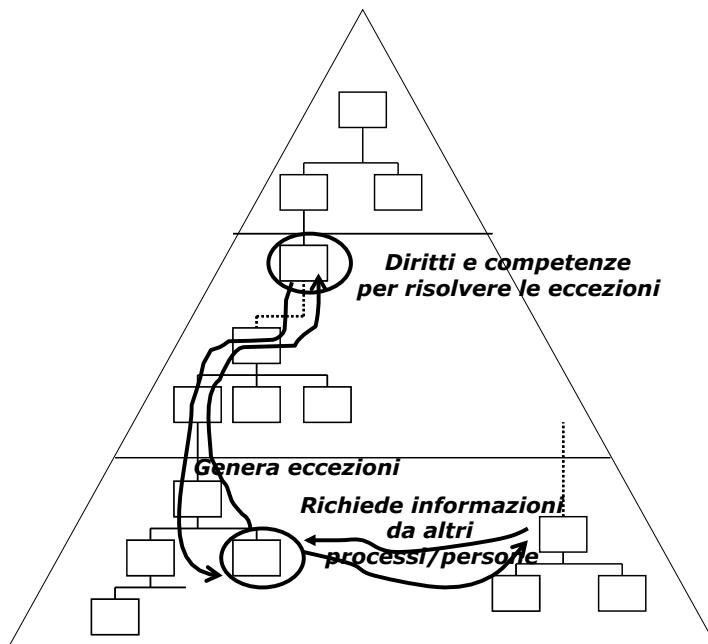


Figura 1.9: Flussi informativi e struttura organizzativa

nenti e di aree che vengono supportate/messe in cooperazione utilizzando un sistema informatico: la tecnologia IT usata dall'azienda (hardware, software, le infrastrutture di rete) per gestire l'informazione (dati), le persone, le procedure, l'organizzazione. La progettazione, lo sviluppo e la gestione di sistemi informativi include anche discipline non tecniche che trattano dell'accettazione dell'IT da parte degli utenti (psicologia), del valore aziendale dell'IT, della strategia di business. Si tratta quindi di una disciplina intersetoriale.

Nella Parte II del libro vengono illustrati più in dettaglio i rapporti tra aspetti relativi agli obiettivi aziendali (aspetti di business), aspetti organizzativi e aspetti tecnologici. 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. Il concetto di EA viene illustrato più in dettaglio nella Sezione 2.1 e i modelli per la sua rappresentazione verranno illustrati nella Parte II del presente testo.

Costruire un sistema informativo basato sull'IT può essere un processo difficile, lungo, complesso e costoso. Progettare un sistema informativo basato su IT significa pensare alla sua architettura, al modello dei processi aziendali, all'interfaccia navigazionale, per esempio basata su Web, alla sua gestione e controllo da parte del management, alla sua interazione con strumenti tipo social network e strumenti mobili per la raccolta e diffusione dell'informazione, e così via.

I componenti essenziali di un sistema informativo basato su IT sono da una parte la base di dati (componente di archiviazione dati), dall'altra le applicazioni che consentono di eseguire i processi di business e l'interfaccia utente. La progettazione e lo sviluppo di sistemi informativi include quindi la progettazione di vari aspetti: dati, processi, interazione utente; lo sviluppo secondo questa separazione in aree di interesse del progetto (dati, logica di business e interfaccia utente) è uno dei modi che permette di gestire la complessità del progetto.

Utilizzare metodologie di progetto ben formalizzate e che compartmentalizzano i vari aspetti può risultare molto utile, separando i vari Layer (livelli logici) per i dati, il business e i servizi (i dettagli verranno dati nel seguito), di presentazione (interfaccia utente) e dell'infrastruttura tecnologica. Devono essere progettare anche le interazioni con i sistemi esterni (per esempio, il sistema informativo di un fornitore, o bancario, per i pagamenti), con le varie sorgenti di dati (per esempio, provenienti da un sistema di monitoraggio sulla catena di produzione), con vari servizi (di fornitura dati, per esempio da un servizio meteo). Infine, vi sono sistemi comuni ai vari livelli (detti componenti Cross-Cutting o trasversali) quali la sicurezza, la gestione operativa e la comunicazione.

Si possono poi considerare i componenti fisici, quali i server, le basi di dati, le aree Intranet, Extranet, le zone protette (da Firewall) o zone demilitarizzate (aree neutre in cui si colloca in genere il server Web), configurazioni di server farm e ricorso a servizi in Cloud.

Come già detto, 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. Il sistema IT (o sistema informatico) cambia nel tempo con l'evolvere dell'azienda e delle tecnologie.

Le attività di gestione dell'informazione non richiedono tutte necessariamente strumenti IT, perché l'informazione può essere gestita anche tramite archivi cartacei, informali e i processi aziendali potranno avere parti automatizzate, in tutto o in parte, e parti svolte manualmente dagli operatori, in particolare per quanto riguarda i processi decisionali. Una delle sfide da affrontare nella progettazione e nella gestione dei sistemi informativi è quella di mantenere l'*allineamento* tra i processi e i dati aziendali e quelli rappresentati nel sistema informativo. Come illustrato in Figura 1.8, nel rapporto complesso tra organizzazione e Information Technology a supporto delle attività aziendale, esistono diversi fattori di mediazione, anche ambientali e culturali, che possono essere all'origine di scelte che nel tempo devono rimanere allineate, sia che il cambiamento abbia origini organizzative o esterne (*requirements pull*), sia che il cambiamento abbia alla base la disponibilità di nuove tecnologie (*technology push*).

Nel presente testo si studieranno in particolare le tecniche, i metodi e i modelli che consentono di progettare e mantenere allineati i processi e la gestione dei dati nei SI basati su Information Technology.

## 1.6 Presentazione della struttura del libro

Il presente libro è strutturato in quattro parti.

Nella prima parte, che comprende il presente capitolo introduttivo seguito dal Capitolo 2 che illustra le principali fasi di progettazione di un sistema informativo, vengono introdotti i principali concetti necessari alla comprensione di un sistema informativo. In particolare, nel Capitolo 2 viene illustrata la centralità del concetto di Enterprise Architecture come elemento su cui basare la progettazione e l'evoluzione nel tempo del sistema informativo aziendale. Nel capitolo sono illustrate anche le principali fasi di progettazione, concentrandosi sulle principali scelte che vengono effettuate nella pianificazione dei progetti di sistemi informativi.

Nella Parte II, viene presentato un approccio architetturale alla progettazione di sistemi informativi, denominato BOAT a partire dalle prospettive considerate: Business, Organization, Architecture, Technology. Nella seconda parte del libro vengono illustrati in dettaglio l'approccio BOAT in generale nel Capitolo 3 e gli elementi forniti per rappresentare la prospettiva di Business e quella Organizzativa, rispettivamente nei Capitoli 4 e 5. All'interno di quest'ultima viene approfondita la modellazione dei processi utilizzando la notazione BPMN e le tecniche per l'identificazione dei processi. Sulla base degli elementi definiti nelle prospettive di Business e Organizzativa nel Capitolo 6 viene definita l'Architettura Funzionale del SI, che viene descritta con un modello che consente di rappresentarla a diversi livelli di aggregazione e di astrazione. La presentazione di BOAT si conclude discutendo nel Capitolo 7 i principali aspetti tecnologici rilevanti per definire la prospettiva tecnologica in una Enterprise Architecture, introducendo il concetto di livello anche per la prospettiva tecnologica, definendo elementi tecnologici a livello applicativo, a livello di piattaforma e a livello di architettura fisica.

Nella Parte III del libro vengono illustrati i principali componenti tecnologici di un sistema informativo. A livello applicativo, nel Capitolo 8 vengono illustrati i sistemi ERP e CRM, utilizzati principalmente a livello operativo e i tipi di componenti utilizzati a livello decisionale, quali datawarehouse e data mining. Per quanto riguarda il livello di piattaforma, nel Capitolo 9 il libro si concentra sui componenti che abilitano l'interazione tra sistemi informativi e tra i loro componenti, discutendo le architetture e le piattaforme di integrazione e i modelli a servizi. Per quanto riguarda l'architettura fisica, nel Capitolo 10 si introducono i diversi tipi di architetture di sistemi, discutendo la loro configurazione e virtualizzazione e le caratteristiche dei sistemi basati su Cloud computing. Il capitolo si conclude discutendo le diverse opzioni di gestione delle architetture fisiche. Un aspetto molto importante nei sistemi informativi è la sicurezza, a cui viene dedicato il Capitolo 11. Vengono presentate le proprietà di sicurezza dei sistemi, l'utilizzo della crittografia per garantire la riservatezza e l'autenticità delle informazioni e i principali meccanismi per il controllo di accesso ai dati. Vengono infine presentati i principali meccanismi di sicurezza infrastrutturali. Sono infine presentati nel Capitolo 12 il modello e la notazione

ArchiMate e in particolare gli elementi che consentono di rappresentare tramite ArchiMate l'infrastruttura tecnologica e i suoi componenti.

Il libro si conclude nella Parte IV con una presentazione di alcuni scenari applicativi per lo sviluppo di sistemi informativi che integrano nuove tecnologie nel Capitolo 13.

## 1.7 Domande

- Illustrare come il modello della Piramide di Anthony viene utilizzato per caratterizzare le tipologie di sistemi informativi e le loro relazioni.
- Confrontare gli elementi rappresentati dal modello della Piramide di Anthony e dal modello di Porter.
- Illustrare che cosa si intende per Risorsa in un SI.
- Illustrare le differenze tra OLTP e OLAP.
- Individuare alcuni processi e dati a livello operativo, di controllo e strategico in una catena di supermercati.



## Capitolo 2

# La progettazione di sistemi informativi

### 2.1 Enterprise Architecture

Inizialmente, i sistemi informativi erano poco complessi e sostanzialmente sistemi chiusi, ovvero non connessi con sistemi di altre aziende. Questi sistemi supportavano un numero limitato di funzionalità a livello aziendale, quali la produzione o la contabilità ed erano organizzati a silos: costituiti da una serie di applicazioni indipendenti specializzate nel supporto e/o esecuzione di una specifica funzionalità. Con il passare del tempo, la complessità dei sistemi informativi è aumentata. Si tratta, infatti, di sistemi composti da diverse applicazioni integrate tra di loro per supportare sia processi interni dell'azienda (intraziendali) sia processi che prevedono la collaborazione con altre aziende (interaziendali). Inoltre, il sistema informativo di un'azienda è spesso il risultato di una crescita incrementale e sempre in evoluzione. Infatti, il sistema informativo si deve adattare sia ai cambiamenti dettati dal business che può cambiare strategia o esigenze sia ai cambiamenti dettati all'evoluzione tecnologica.

Quindi, come già introdotto nel Capitolo 1 e illustrato nella Figura 1.8, un progetto legato all'evoluzione di un sistema informativo può derivare da spinte tecnologiche (*technology push*) e spinte di business (*requirements pull*) che vanno conciliate e integrate per lo sviluppo organico del sistema informativo (Figura 2.1).

Le spinte di business possono derivare dal bisogno di nuove funzionalità oppure dall'esigenza di migliorare la qualità delle funzionalità offerte. Le spinte tecnologiche invece possono essere causate da una necessaria evoluzione tecnologica, che spesso richiede anche un adeguamento della struttura del sistema informativo. Si noti come le due spinte al cambiamento poi si possano influenzare a vicenda: la richiesta di nuove funzionalità necessariamente porta all'evoluzione tecnologica mentre l'evoluzione tecnologica porta spesso alla possibilità di erogare nuovi servizi e funzionalità.

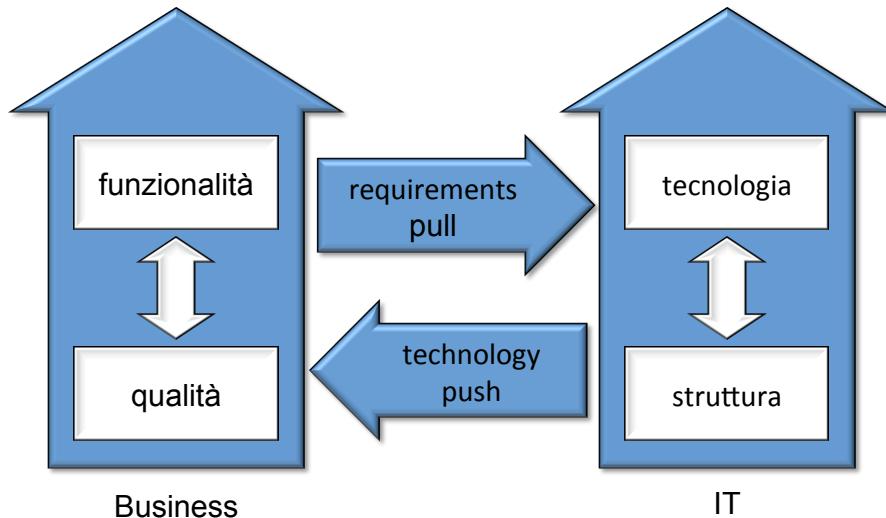


Figura 2.1: Il bisogno di una Enterprise Architecture

Per mantenere gestibili le relazioni tra il lato business e quello tecnologico è necessario progettare strutture di natura astratta, in modo da risultare indipendenti da scelte specifiche, tecnologiche o di business. Per definire la struttura del sistema informativo si ricorre al concetto di *Architettura*. Come definito nello standard ISO/IEC 42010[21], l’architettura di un sistema è data dall’insieme componenti, delle relazioni fra gli stessi e dei principi che definiscono la loro progettazione e evoluzione nel tempo. Quindi l’architettura fornisce tre diverse viste: (i) la vista statica relativa alla descrizione dei componenti, (ii) la vista dinamica relativa alla descrizione delle relazioni tra componenti e infine (iii) la vista delle regole relativa alle procedure che supportano la progettazione e evoluzione del sistema nel tempo. L’architettura fornisce anche diverse prospettive agli stakeholder del sistema. Si definiscono *stakeholder* coloro che hanno interesse al sistema considerato, per esempio relativamente all’intero ciclo di vita del sistema, perché ne sono proprietari, fruitori, finanziatori, utenti. In genere, possono aver bisogno di analizzare l’architettura da diversi punti di vista. Si pensi ad esempio all’architettura di una città formata da diversi componenti quali edifici, strade, parchi, ecc. Gli addetti degli uffici del comune della città potrebbero aver bisogno di una veduta aerea (cioè, una prospettiva globale) della città per vedere l’architettura nel suo complesso al fine di programmare interventi di sviluppo e crescita dell’area urbana. Una prospettiva locale, focalizzata sui dintorni di un singolo edificio potrebbe invece essere utile al proprietario di quell’edificio per analizzare la possibilità di espansione dell’edificio e chiedere permessi di ampliamento. In entrambi i casi, i due stakeholder sono interessati alla architettura della città per diversi obiettivi e quindi necessitano di analizzare l’architettura da due diverse prospettive. La definizione di una

architettura deve permettere dunque di soddisfare le diverse esigenze degli stakeholder dando la possibilità di analizzare il sistema da diversi punti di vista e con diversi livelli di granularità. E' per questo che ci si avvale di framework architettonici che stabiliscono le convenzioni, i principi e i metodi per descrivere le architetture in diversi domini e per diversi stakeholder.

Se proviamo ad applicare questo concetto generale di architettura a un'organizzazione e al suo sistema informativo arriviamo alla definizione di Enterprise Architecture. In [6] l'*Enterprise Architecture* (EA) viene definita come il mezzo per analizzare e descrivere una organizzazione nel suo stato attuale e futuro partendo da una prospettiva integrata di strategia, business e tecnologica. L'EA fornisce, infatti, una panoramica (mappa) dei processi, sistemi, tecnologia, strutture e capacità di una organizzazione. Questa mappa fornisce un contesto strategico per supportare l'evoluzione continua del sistema informativo dettata dalle spinte di business o tecnologiche. L'EA può essere vista cioè come una matrice (blueprint) di alto livello del sistema, che serve per capire la sua struttura interna come supporto alla sua progettazione, riprogettazione, configurazione e manutenzione.

Questo non è molto diverso dell'architettura (tecnica) di un edificio complesso, un ponte oppure una metropolitana. Il fatto che un'architettura sia una matrice di alto livello significa che l'enfasi è su una rappresentazione astratta, concettuale, e non sui singoli dettagli tecnici. Anche per questo è confrontabile con altri tipi di architetture: l'architettura di un edificio complesso o di un ponte, che non è specificata in termini di singoli colonne, mattoni, muri, ecc., ma con disegni astratti della costruzione da realizzare.

Alla fine degli anni Novanta, John Zachman definì fondamentale l'uso di una architettura per descrivere il sistema informativo aziendale in quanto la tecnologia iniziava a consentire la distribuzione di applicazioni e dati su macchine diverse: a suo parere la decentralizzazione senza una struttura ben precisa avrebbe generato il caos. Propose così il framework per definire l'Enterprise Architecture [42, 36]: una tassonomia per organizzare documenti, specifiche e modelli esistenti in uno schema generale e mostrare le relazioni che esistono tra essi.

Il *framework di Zachman* è organizzato in righe e colonne (si veda la Figura 2.2). Le colonne definiscono gli *Aspetti* da analizzare; in particolare vengono definite le seguenti sei colonne:

- Dati (cosa): si considerano i dati di cui l'organizzazione ha bisogno per operare.
- Funzioni (come): si analizzano le funzionalità eseguite dall'azienda per mandare avanti il proprio core business.
- Rete (dove): è relativa alla distribuzione geografica dell'organizzazione.
- Persone (chi): la gente coinvolta nell'esecuzione delle diverse funzionalità aziendali.
- Tempo (quando): relativo agli eventi significativi per il business.

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

Figura 2.2: Il framework per l'Enterprise Architecture di Zachman (elaborazione di [42])

- Motivazione (perché): si considerano gli obiettivi dell'azienda.

Da notare che nel framework di Zachman non viene definito alcun ordinamento tra le colonne.

Le righe invece mettono in luce i diversi *Punti di vista* che possono interessare diversi stakeholder. In particolare Zachman individua i seguenti cinque punti di vista:

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

Per ogni riga vengono definiti i vincoli sul sistema nel livello considerato. Da notare che questi vincoli sono additivi, nel senso che i vincoli a un livello inferiore si aggiungono a quelli dei livelli superiori.

Le celle del framework di Zachman create dall'intersezione tra righe e colonne 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. Se si ripercorre la colonna relativa ai dati, per esempio, si può vedere come lo scopo viene descritto attraverso la lista delle cose importanti da modellare in un particolare contesto. Il modello concettuale della base di dati da realizzare viene fornito tramite diagrammi noti in letteratura (es. modello Entità-Relazione ER). La progettazione concettuale viene poi seguita dalla progettazione logica che traduce il diagramma definito a livello concettuale nella rappresentazione dei dati adeguata alla tecnologia prescelta. La realizzazione effettiva della base di dati viene preceduta da un modello fisico e dalla definizione puntuale dei dati da inserire nel sistema. L'implementazione effettiva, infine, genera la funzionalità organizzativa costituita dal database specifico (per esempio, la base di dati Ordini). Ogni colonna, letta dall'alto verso il basso, specifica come diversi diagrammi e modelli vengono raffinati per passare dalle indicazioni del business all'effettiva realizzazione del sistema.

Il framework di Zachman è uno strumento efficace per descrivere il sistema informativo dell'organizzazione nel suo complesso. Oltre allo scopo descrittivo, la descrizione dell'EA può essere utilizzata anche a scopo prescrittivo al fine di guidare l'evoluzione del sistema.

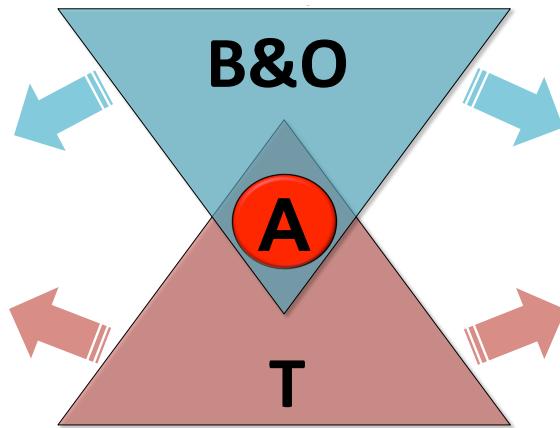


Figura 2.3: L’architettura funzionale (A) come elemento di pivot tra aspetti di business(B)/organizzativi(O) e aspetti tecnologici (T) in una EA

Nel presente testo utilizziamo per la progettazione dell’architettura del sistema come EA l’approccio BOAT proposto in [19], in cui gli aspetti da specificare sono i seguenti:

- Aspetti di business (B), legati agli obiettivi aziendali.
- Aspetti organizzativi (O), legati alla struttura dell’organizzazione.
- Aspetti architetturali (A), che descrivono l’*architettura funzionale* del sistema.
- Aspetti tecnologici (T), in cui vengono delineate le scelte tecnologiche di tipo generale (ad esempio, le piattaforme per lo scambio di dati tra i moduli, oppure le funzionalità necessarie per gestire aspetti di sicurezza).

Mentre i dettagli sulla rappresentazione di questi elementi sono discussi nelle parti successive del libro, si sottolinea in questa sezione il ruolo di collegamento svolto dall’architettura funzionale (A), che consente di creare le corrispondenze tra gli aspetti di business e organizzativi e gli aspetti tecnologici, come illustrato nella Figura 2.3.

Utilizziamo la seguente definizione di architettura funzionale di un sistema informativo [19]:

L’*architettura funzionale* (A) di un sistema informativo specifica la struttura di quel sistema in termini di componenti software che supportano specifiche funzioni e in termini di interfacce che consentono l’interazione tra quei componenti.

L’insieme dei quattro aspetti, B/O/A/T, definisce l’EA, specificando gli aspetti vincolanti per la realizzazione dei sistemi informativi, lo sviluppo di nuovi componenti o evoluzioni del sistema.



Figura 2.4: Processo di gestione della Enterprise Architecture

Un altro aspetto da sottolineare a questo punto è che anche l'EA deve essere gestita, in particolare, con riferimento alle continue evoluzioni del sistema.

Innanzitutto, durante un progetto, è necessario sviluppare l'EA futura ('*to-be*') in relazione a quella in essere (architettura presente, o '*as is*'). Inoltre, durante lo sviluppo di singole fasi di progetto, sarà necessario avere cura che l'EA prevista e quella effettivamente realizzata rimangano allineate nel tempo (Figura 2.4).

Per ottenere questi due obiettivi sono necessarie sia attività di governance (gestione strategica del sistema) sia di gestione nell'ambito dei singoli progetti.

Le fasi di progettazione dell'architettura di riferimento e il suo ruolo centrale nelle fasi di sviluppo dei sistemi sono illustrate nel seguito.

## 2.2 Il processo di gestione del sistema informativo

Spesso nella realizzazione di una applicazione, la progettazione assume un carattere non sistematico e privo di una chiara metodologia. Questo approccio è tipico nei casi in cui si vuole creare un sistema in tempi molto rapidi, ma può essere pericoloso in quanto la costruzione del sistema procederà per aggiustamenti successivi sulla base di richieste degli utenti o dei committenti e può risultare pericoloso se si vuole realizzare un'interazione con i sistemi informativi aziendali controllata, sicura e con una gestione coerente delle applicazioni. I problemi che si possono creare in sistemi pianificati e progettati in modo non sistematico vanno dal rischio di non disponibilità del sistema dovuto a interruzioni impreviste (crash), all'impossibilità di manutenere il sistema, all'incapacità di soddisfare nuove esigenze o di gestire la crescita (scalabilità).

Per gestire la complessità derivante dai vari aspetti legati alla progettazione e gestione di sistemi informativi, la soluzione è quella di seguire un processo sistematico. In particolare, le attività di pianificazione, progettazione, sviluppo e manutenzione di un sistema informativo rientrano in quello che si definisce **processo di gestione del sistema informativo** [35, 31, 3]. Tale processo comprende non solo problematiche tecnologiche, ma deve tenere in considera-

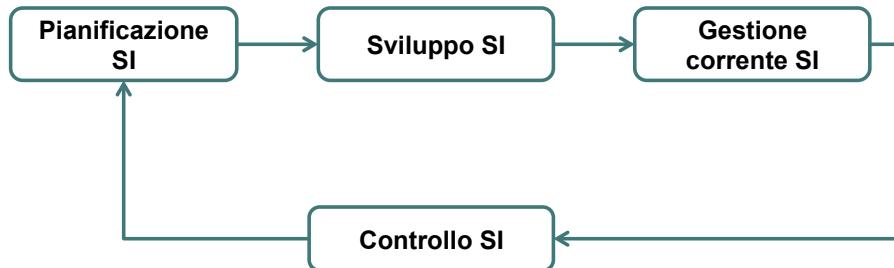


Figura 2.5: Gestione del sistema informativo (rielaborazione da [31])

zione gli elementi non tecnologici del sistema informativo, quali l'impatto che i cambiamenti sul sistema informativo hanno sull'intera organizzazione e sulle strategie organizzative.

L'esigenza di progettare e gestire in modo strutturato un sistema informativo nel suo funzionamento e nella sua evoluzione, considerando non solo gli aspetti tecnologici, ma anche quelli organizzativi, è sorta sin dai primi tempi in cui i sistemi informativi si sono affacciati nelle aziende. Si pensi ad esempio al Business System Planning (BSP) proposto da IBM negli anni '70, fino ad arrivare a framework più completi quali TOGAF<sup>1</sup>, il cui scopo è progettare e gestire una Enterprise Architecture .

Indipendentemente dal metodo utilizzato, il processo di gestione è anch'esso un processo, definito come un insieme di attività interrelate svolte per il raggiungimento di un obiettivo: per ottenere una buona progettazione e gestione di un sistema informativo, è utile seguire un insieme di fasi operative definite e organizzate in modo strutturato. In linea generale, al di là dello specifico metodo adottato, la gestione del sistema informativo può essere rappresentata da quattro fasi principali, organizzate in un ciclo, in un'ottica di miglioramento continuo, così come illustrato in Figura 2.5:

- *Pianificazione*: fase in cui si delineano le linee guida strategiche, il ruolo delle componenti organizzative e le istruzioni operative per la realizzazione.
- *Sviluppo*: raccolta e analisi dei requisiti e sviluppo del software e definizione dell'architettura hardware.
- *Gestione corrente*: interventi di routine eseguiti per il mantenimento del sistema.
- *Controllo*: operazioni effettuate in modo periodico per valutare l'adeguatezza del sistema informativo.

La prima cosa da notare è che la gestione del sistema informativo non si limita al solo sviluppo del software, ma comprende altre attività fondamentali

<sup>1</sup><http://www.opengroup.org/subjectareas/enterprise/togaf>

mirate a capire cosa sia necessario fare e capire come farlo. Inoltre, una volta realizzato, il sistema informativo viene continuamente controllato per analizzare il bisogno di cambiamenti. I cambiamenti che possono causare l'evoluzione del sistema informativo possono avvenire per cause:

- *Esogene all'organizzazione*: ad esempio, variazioni di leggi, oppure del mercato in cui l'azienda opera, che impone innovazioni tecnologiche per poter competere con gli altri player del settore.
- *Endogene all'organizzazione*: fattori interni, che possono riguardare l'esigenza di un rinnovamento tecnologico necessario a causa dell'obsolescenza tecnologica oppure la necessità di nuove tecnologie dettata da scelte strategiche (per esempio, la decisione di vendere nuovi prodotti/offrire servizi on line).

Diverse teorie affermano che il cambiamento è necessario perché l'azienda continui a detenere una posizione competitiva all'interno del proprio settore di riferimento. Infatti, vi sono metodologie (per esempio, ITIL<sup>2</sup>) che aiutano le aziende nell'identificazione delle fasi e dei passi per la gestione del cambiamento.

La gestione del cambiamento dipende dalla complessità e dall'impatto dell'intervento. Infatti, gli interventi sul sistema informativo possono essere:

1. Interventi di routine locali ad alcuni elementi del sistema: in questa categoria si annoverano interventi di manutenzione hardware o aggiornamento del software.
2. Interventi occasionali e complessi: ci si riferisce ad interventi che hanno un impatto elevato sull'architettura esistente del sistema informativo. Esempi di questi tipo di interventi sono la migrazione ad una nuova tecnologia (per esempio sistemi in cloud o mobili, o basati su sensori e tecnologie di sensori *wearable*) o l'aggiunta di nuove funzionalità.

È da notare che gli interventi occasionali e complessi al sistema informativo hanno connotazione e conseguenze radicalmente diverse nel caso si consideri una nuova organizzazione o una organizzazione esistente. Nel primo caso non si hanno sistemi preesistenti e non si ha alcun vincolo nella progettazione del sistema informativo dettato dall'esigenza di integrarsi con un sistema già utilizzato. Nel secondo caso, la progettazione di nuove funzionalità richiede l'analisi del sistema esistente e la considerazione di vincoli architetturali.

Nel presente testo si studieranno principalmente i passi di Pianificazione e sviluppo dell'EA.

## 2.3 Pianificazione

Obiettivo della pianificazione è predisporre una proposta di progetto che il committente possa approvare. I ruoli coinvolti in questa fase sono: (i) il committente

---

<sup>2</sup><http://www.itil-officialsite.com/>

che è colui che approva/finanzia il progetto, (ii) il/i soggetto/i interni o esterni (es. consulenti) all'azienda che predispongono il piano, (iii) i referenti dell'azienda (stakeholder) che sono coloro che sono interessati al progetto (es. azionisti, operatori, utilizzatori esterni, clienti o terze parti interessate alla catena del valore).

L'attività di pianificazione comprende due attività principali:

- *Pianificazione strategica*: in questa fase si identificano gli obiettivi che richiedono interventi di natura informatica.
- *Studio di fattibilità*: una volta appurato il bisogno di un intervento di natura informatica, si identificano le alternative progettuali.

### 2.3.1 Pianificazione strategica

La fase di *pianificazione strategica* si basa sull'acquisizione della conoscenza sulla situazione attuale dell'impresa e sull'identificazione delle aree e/o processi che necessitano interventi informatici. Si tratta di un'attività complessa e delicata che mira a realizzare il tanto ambito allineamento tra IT e business, secondo cui la strategia dei sistemi informativi dovrebbe essere in linea con gli obiettivi di business. Le attività principali della pianificazione strategica sono:

- L'analisi delle opportunità di sviluppo del sistema informativo: analizzando tecnologie, strategie e processi si delineano i possibili sviluppi del sistema IT.
- L'analisi dei fabbisogni informativi: un'analisi più dettagliata che porta alla modellazione delle informazioni e agli scambi informativi necessari per il funzionamento dell'organizzazione.

Nel corso degli anni, sono stati messi a punto modelli specifici per la gestione di queste due attività. Tra i vari strumenti, la matrice di Porter e Millar (si veda Figura 2.6) può essere un aiuto per capire l'importanza di un sistema informativo all'interno della propria organizzazione considerando l'intensità informativa del prodotto (o servizio) erogato e dei processi adottati per fornire tali prodotti (o servizi). Aziende tradizionali come quelle nel campo edile si posizionano all'interno del III quadrante, considerando che i loro processi e i loro prodotti non hanno un elevato contenuto tecnologico, in senso informatico. Al contrario, banche ed assicurazioni trovano nell'IT un supporto imprescindibile per eseguire i propri processi, per il fatto che i prodotti che offrono sono principalmente informazione. L'industria dell'automobile, così come tutte quelle organizzazioni che possono sfruttare l'IT principalmente per produrre prodotti tradizionali automatizzando le proprie linee produttive, trovano strategico l'uso dell'IT all'interno dei processi.

Quello che sta avvenendo negli ultimi anni, e che generalmente cade sotto il cappello di *digitalizzazione*, è il tentativo di capire le opportunità offerte dalla tecnologia al fine di spostare la posizione aziendale verso il I quadrante. Un

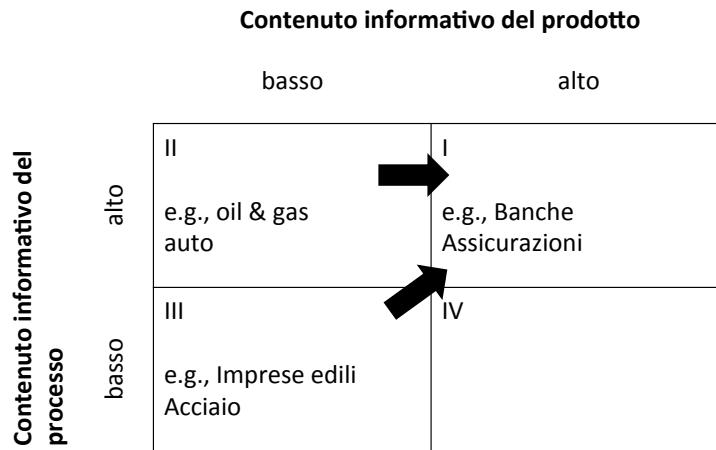


Figura 2.6: Matrice intensità informativa di Porter e Millar

esempio è dato dal mondo automobilistico. La creazione di automobili completamente gestite via software arricchisce il contenuto informativo del prodotto “auto” portando numerosi vantaggi alla clientela (per esempio, si pensi al controllo manutentivo preventivo) ed aprendo le porte ad ulteriori innovazioni (ad esempio, self-driving car). Lo stesso fenomeno sta avvenendo anche nel campo edilizio. Standard di rappresentazione degli oggetti facenti parte di un edificio e codifica precisa di tutte le fasi del ciclo di vita di un edificio - dalla progettazione alla manutenzione - è l’obiettivo del Building Information Modeling (BIM). Grazie a ciò si sta cercando di strutturare meglio e di digitalizzare il processo che porta alla costruzione di edifici complessi. Al tempo stesso, anche il manufatto, sia esso un edificio civile, residenziale o anche un ponte, è sempre più un oggetto digitalizzato attraverso l’installazione di sensoristica e dispositivi di attuazione al fine di migliorare il confort degli abitanti o rilevare possibili difetti strutturali in tempo reale e con costi minori.

Questo tipo di evoluzione sottolinea come il paradigma che ha sempre visto l’IT come una variabile dipendente nelle scelte organizzative deve cambiare, considerando l’IT una variabile indipendente anche a livello strategico. L’IT è stato spesso infatti spesso visto solo come un costo; le innovazioni all’interno di una organizzazione sono sempre state guidate da scelte strategiche che hanno puntato in primis a ridisegnare processi o prodotti, e solo a seguito di tali decisioni ci si è occupati di capire quale tecnologia fosse in grado di aiutare la realizzazione di tali decisioni. Oggi, l’IT deve essere visto come parte integrante nelle decisioni strategiche e concorrere alla definizione degli obiettivi e non subirli.



Figura 2.7: Fasi che compongono uno studio di fattibilità (rielaborazione da [31])

### 2.3.2 Studio di fattibilità

Lo *studio di fattibilità* precisa le caratteristiche di un potenziale intervento al sistema informativo. Di conseguenza si concentra sulla definizione di progetti informatici. Le fasi che compongono uno studio di fattibilità sono rappresentate in Figura 2.7.

#### Definizione degli obiettivi e specifiche funzionali

La *Definizione degli obiettivi e specifiche funzionali* mira in primo luogo a definire le aree organizzative su cui bisogna lavorare e a delineare i miglioramenti da apportare per ogni area considerata. Una volta definito in questo modo l'ambito in cui si dovranno definire i progetti informatici, si passa alla stesura dei requisiti funzionali: si descrivono ad alto livello le funzionalità che il sistema dovrebbe fornire. Infine, questa prima attività deve specificare i vincoli da considerare nella stesura dei progetti. Tali vincoli sono vincoli previsionali su specifiche non funzionali<sup>3</sup>. In particolare, ci si concentra su tre variabili: costo, tempo e qualità. Da notare che solitamente queste variabili non possono essere ottimizzate contemporaneamente, ma solo a coppie. Infatti, si potranno realizzare progetti a basso costo in breve tempo, ma sarà necessario rinunciare alla componente qualità. Oppure si può optare per progetti economici ad alta qualità, ma realizzabili in tempi lunghi o ancora si può pensare a progetti informatici costosi in grado di garantire qualità alta e tempi ridotti.

#### Progettazione delle soluzioni

La *Progettazione delle soluzioni* si compone di tre fasi (Figura 2.8):

<sup>3</sup>Relative al modo in cui le funzionalità dovranno essere fornite.

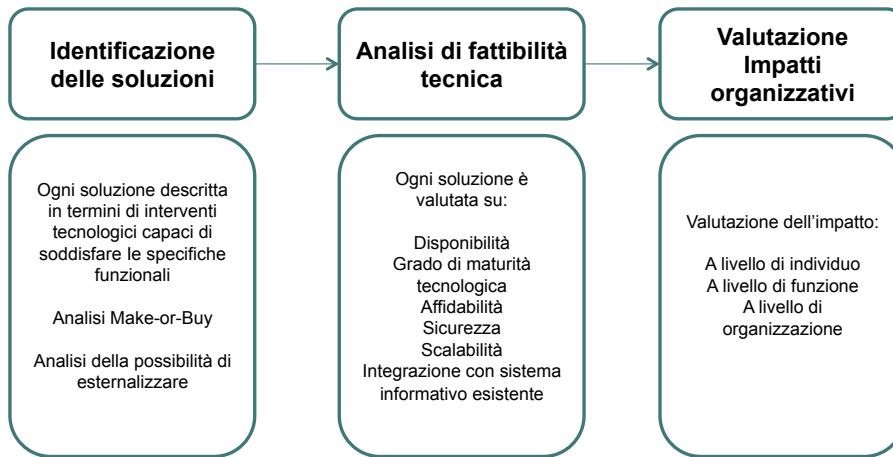


Figura 2.8: Fasi della progettazione delle soluzioni (rielaborazione da [31])

1. *Identificazione delle soluzioni*: sulla base delle specifiche funzionali e dei vincoli definiti nella fase precedente si definiscono le soluzioni informatiche in grado di soddisfarli. Ogni soluzione viene descritta specificando nel dettaglio gli interventi tecnologici.

L'analisi delle soluzioni porta solitamente a svolgere due tipi di analisi: analisi Make-or-Buy e analisi della possibilità di esternalizzare (outsourcing).

L'analisi **Make-or-Buy** prende in considerazione il fatto che sul mercato per alcuni interventi esistono dei prodotti software già pronti: i cosiddetti COTS (Commercial Off-The-Shelf o Commercially available Off-The-Shelf). L'adozione di questi software (alternativa *Buy*) si pone in alternativa alla possibilità di progettare e realizzare un prodotto su misura (alternativa *Make*).

L'alternativa **Buy** (acquisto) ha come vantaggi il fatto di essere più economica dell'alternativa Make e di richiedere solitamente un minor tempo di implementazione.

L'alternativa **Make** (sviluppo) porta alla realizzazione di un prodotto più flessibile e vicino alle esigenze aziendali. Esiste anche la possibilità di optare per un approccio ibrido in cui parti della soluzioni vengono acquisite secondo la modalità Make e altre secondo la modalità Buy al fine di massimizzare i vantaggi dei due approcci e avere progetti contenuti dal punto di vista di tempo e costi, ma in grado di soddisfare appieno le esigenze aziendali.

In Figura 2.9 vengono riassunti i principali e svantaggi delle soluzioni Make e Buy.



Figura 2.9: Confronto tra soluzioni Make e Buy

Una volta definito come verrà realizzato il sistema si può passare all’analisi che interessa la sua *gestione*. La gestione può essere interna (in-house) o esterna all’azienda (outsourcing). **IT outsourcing** è il termine utilizzato per indicare la situazione in cui la realizzazione o la gestione dei servizi informatici viene affidata a un fornitore esterno.

L’outsourcing porta molti vantaggi soprattutto ad aziende di dimensioni limitate che non hanno al loro interno competenze informatiche. In particolare porta a (i) un miglioramento della qualità del servizio dettato dalle competenze e professionalità del fornitore esterno, (ii) la possibilità per l’azienda di focalizzarsi sulle proprie attività di core business, (iii) una diminuzione dei costi IT e (iv) una maggiore propensione all’aggiornamento ed evoluzione tecnologica.

I benefici principali sono la possibilità di utilizzare competenze non disponibili all’interno dell’azienda e il risparmio potenziale di costi. Gli svantaggi principali riguardano la perdita di competenze interne e la dipendenza dal fornitore di servizi.

In Figura 2.10 vengono riassunti i principali e svantaggi dell’outsourcing.

2. *Analisi di fattibilità tecnica:* in questa fase vengono descritte le possibili soluzioni per rispondere alle esigenze definite nelle fasi precedenti.

Ogni soluzione deve essere valutata al fine di stabilire la sua adeguatezza da un punto di vista tecnologico. La prima valutazione riguarda le caratteristiche tecniche. In particolare si valutano dimensioni di qualità architetturali quali:

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

Figura 2.10: Vantaggi e svantaggi dell'outsourcing

- (a) Disponibilità: valuta se la soluzione proposta è effettivamente realizzabile (o acquisibile sul mercato) nei tempi previsti.
- (b) Grado di maturità tecnologica: prendendo in considerazione la tecnologia che si vuole utilizzare si valuta in che stadio del ciclo di vita si trova.
- (c) Affidabilità: si considera qual è l'esposizione ai guasti della soluzione identificata. Infatti, l'affidabilità si traduce nella proprietà di un sistema di garantire una certa continuità nella sua operatività.
- (d) Sicurezza: si valuta l'impatto che la soluzione identificata ha sulla sicurezza del sistema informativo e se richiede l'adozione di nuovi meccanismi di sicurezza quali la crittografia, per esempio.
- (e) Scalabilità: si valuta la propensione all'evoluzione del sistema. In particolare, ci si chiede quanto costerebbe in termini di risorse (tempo e denaro) ampliare il sistema in un periodo successivo a quello considerato.

Una volta analizzate le caratteristiche tecniche si deve valutare come la soluzione individuata si integra con il sistema informativo esistente (se presente). Spesso, infatti, il sistema informativo esistente (detto anche *sistema legacy*) pone vincoli architettonici sulle soluzioni, specialmente se tra il sistema informativo esistente e la soluzione considerata si devono gestire flussi informativi. Se i due sistemi sono eterogenei e tra loro non compatibili, si deve tenere in considerazione che oltre al costo di sviluppo della soluzione ci saranno costi aggiuntivi derivanti dall'integrazione dei sistemi. I costi sono il più delle volte ingenti e i tempi di sviluppo non trascurabili. L'analisi di fattibilità tecnica si conclude con la valutazione

dell'allineamento della soluzione con gli obiettivi specificati e si misura la sua adeguatezza e la sua importanza strategica.

3. *Valutazione degli impatti organizzativi*: l'introduzione di una nuova soluzione IT ha solitamente diversi impatti all'interno di un'azienda. È necessario valutare questo impatto per valutare eventuali limiti nella realizzabilità del sistema. L'impatto organizzativo può essere valutato a livello di individuo, a livello di funzione e a livello di organizzazione. Per quanto riguarda l'individuo, è importante tenere in considerazione la reazione degli utilizzatori del nuovo sistema. Infatti, una resistenza al cambiamento o scarse competenze possono compromettere il successo del progetto. A livello di funzione si valuta se la nuova soluzione porta al bisogno di ridisegnare ruoli o a ridefinire mansioni. Infine a livello organizzativo si valuta se la soluzione implica la necessità di collaborare con altre realtà dando vita a partnership di varia natura e livelli.

#### **Valutazione di convenienza economica**

Per quanto riguarda la *valutazione della convenienza economica*, le soluzioni proposte oltre ad essere fattibili tecnicamente devono anche essere convenienti da un punto di vista economico. In altre parole, i benefici associati alla realizzazione della soluzione devono essere maggiori dei costi.

I *costi* devono essere contabilizzati considerando tutte le risorse associate alla soluzione: hardware, software e risorse umane.

Ogni costo deve essere correlato con la fase del ciclo di vita della soluzione in cui si manifesta e l'unità funzionale a cui viene attribuito.

I *benefici* in genere si quantificano considerando solo i benefici tangibili e diretti, quelli cioè facilmente monetizzabili. Detti benefici si concretizzano spesso in una riduzione dei costi operativi (per esempio, minor tempo di esecuzione di attività automatizzate), in una riduzione degli oneri finanziari (per esempio, minori giacenze di magazzino), o in un incremento dei ricavi (per esempio, aumento delle vendite).

L'analisi dettagliata di questi aspetti non viene trattata nell'ambito del presente testo. Per eventuali approfondimenti si fa riferimento al testo [24].

## **2.4 Ciclo di vita di sviluppo del SI**

Lo studio di fattibilità porta alla scelta della soluzione considerando diverse opzioni valide e fattibili sia da un punto di vista tecnico che organizzativo, valutando le alternative anche in base alla convenienza economica.

Una volta scelta la soluzione, si procede alla realizzazione dell'intervento con l'attività di sviluppo. La fase di sviluppo è stata oggetto di diversi studi sia nell'ambito dei sistemi informativi sia nell'ambito dell'ingegneria del software.

Nel caso di un processo di sviluppo con alternativa Make, il ciclo di sviluppo è quello rappresentato in Figura 2.11 e consiste in un ciclo retroazionato in cui

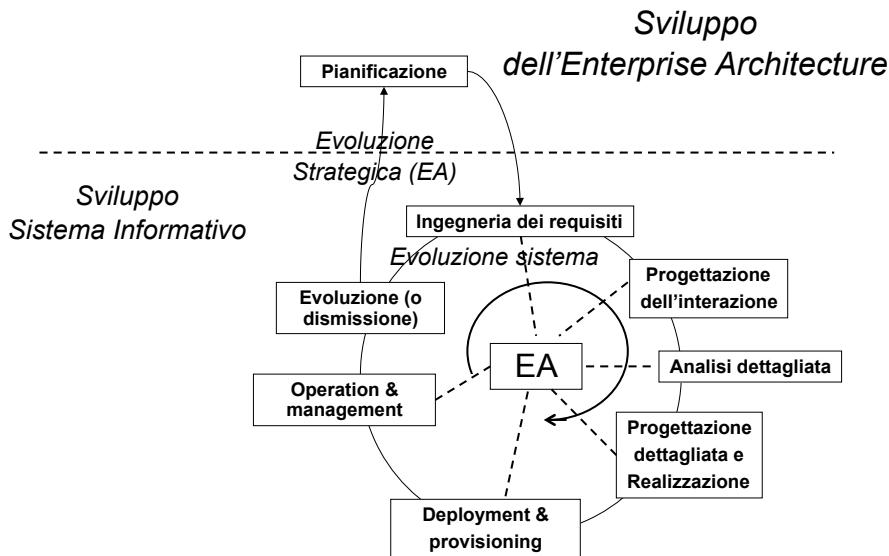


Figura 2.11: Il ciclo di vita dello sviluppo di un sistema informativo

si hanno le seguenti attività: **Pianificazione** di dettaglio della soluzione prescelta, **Raccolta e Analisi dei requisiti** al fine di definire le funzionalità che il sistema deve fornire, **Analisi** delle funzionalità da fornire (**Progettazione dell'interfaccia** e **Analisi dettagliata**), **Progettazione dettagliata e Realizzazione** delle suddette funzionalità. La retroazione consiste in attività di valutazione svolte nella fase di messa in opera (**Deployment & Provisioning**) e gestione del sistema (**Operations & Management**) mirate a garantire l'allineamento del progetto e dell'implementazione ai requisiti iniziali e agli obiettivi di progetto.

Tutte queste fasi compongono il ciclo di vita di sviluppo del software (SDLC, Software Development Life Cycle) e la disciplina dell'ingegneria del software si occupa, tra i suoi vari aspetti, di fornire metodi e strumenti per supportare al meglio tale ciclo di vita. Storicamente l'approccio detto *a cascata*, o *waterfall*, prevedeva l'esecuzione delle varie attività che compongono il ciclo di vita in una stretta sequenza. Questo però porta a verificare l'aderenza tra quanto sviluppato e le aspettative del committente solo al termine del lavoro. Questo implica che eventuali divergenze possono essere comprese solo al termine dello sviluppo, con pesanti conseguenze economiche.

L'approccio *iterativo* punta a risolvere questo tipo di problema prevedendo lo sviluppo incrementale della soluzione. Partendo da prototipi iniziali con poche funzionalità seguite da ulteriori release sempre più complete, il committente viene coinvolto in modo continuo durante il ciclo di vita. Questo permette di verificare l'aderenza tra requisiti e prodotto più frequentemente e risolvere eventuali divergenze prima che diventino troppo ampie.

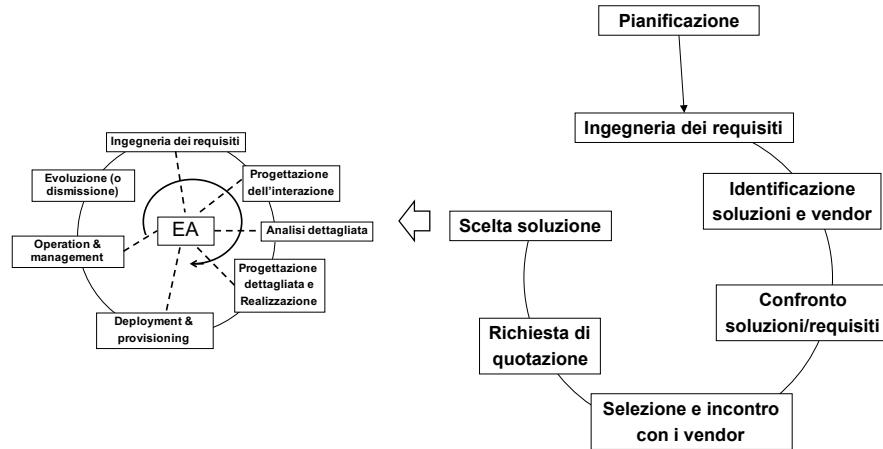


Figura 2.12: Ciclo di sviluppo alternativa Buy

Negli ultimi anni stanno prendendo sempre più piede gli approcci *agili* che prevedono il rilascio incrementale di diverse release, ma con minore strutturazione delle varie attività.

Vista la disponibilità di numerosi software COTS per le organizzazioni, è opportuno notare che il processo di sviluppo di un'alternativa Buy ha un ciclo di sviluppo leggermente diverso da quello dell'alternativa Make (vedere Figura 2.12). Il ciclo di sviluppo e implementazione viene preceduto dal ciclo di acquisto, in cui si ha la selezione della soluzione idonea ai requisiti aziendali. Il ciclo di sviluppo è poi a supporto della personalizzazione, implementazione e integrazione della soluzione scelta.

Lasciando a testi di ingegneria del software la trattazione in dettaglio dei cicli di vita dello sviluppo software, in questo testo si forniscono dettagli relativi alla pianificazione del sistema informativo e allo sviluppo dell'EA.

## 2.5 Domande

- Illustrare i modelli per la segmentazione dei sistemi informativi.
- Descrivere le fasi del processo di sviluppo.
- Illustrare le differenze tra requisiti funzionali e non funzionali.
- Discutere i criteri che vengono utilizzati per decidere tra la selezione di componenti software (approccio buy) e la realizzazione di un nuovo software ad hoc (approccio make).

## **Parte II**

# **Un approccio architetturale ai sistemi informativi**



# Capitolo 3

## Approccio BOAT

Come descritto nei capitoli precedenti, l’Enterprise Architecture (EA) è diventata un elemento fondamentale per la pianificazione e progettazione dei sistemi informativi aziendali. In questo libro verrà illustrato in dettaglio il framework BOAT (Business, Organization, Architecture, Technology) che definisce passi, metodi e strumenti in grado di descrivere il sistema informativo aziendale nel suo complesso da diverse prospettive. Queste descrizioni possono avere due diversi scopi: descrittivo e prescrittivo. Nel primo caso, esse mirano a rappresentare il sistema informativo in essere, mentre nel secondo caso hanno l’obiettivo di pianificare l’adozione di nuove parti e componenti del sistema informativo.

### 3.1 Descrizione

Utilizzando come riferimento la terminologia introdotta dal framework di Zachman descritto nel capitolo precedente, BOAT [19] permette di analizzare il sistema informativo dalle seguenti prospettive:

- *Business*: descrive gli obiettivi di business - spiega l’esistenza di uno scenario.
- *Organizzazione*: descrive come le organizzazioni sono strutturate e connesse per raggiungere gli obiettivi definiti dal business.
- *Architettura*: specifica in modo dettagliato i sistemi informativi utilizzati per supportare le organizzazioni.
- *Tecnologia*: descrive la realizzazione tecnologica dei sistemi la cui struttura è descritta dall’architettura.

Gli aspetti analizzati dal framework BOAT sono le *organizzazioni* coinvolte (quella analizzata e le organizzazioni con cui si interfaccia), gli *oggetti* da gestire (cioè, prodotti e servizi) e il tempo che regola gli eventi significativi da rappresentare e gestire. Il framework BOAT può essere rappresentato come in Figura

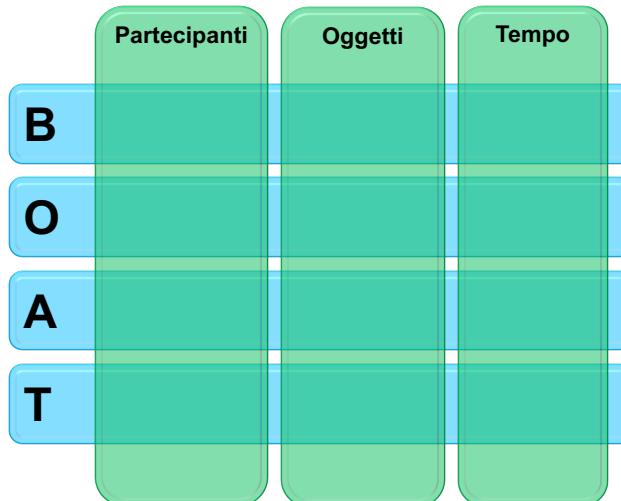


Figura 3.1: Il framework BOAT

3.1 è assimilato al framework di Zachman in quanto le prospettive analizzate sono equiparabili rispettivamente ai primi quattro livelli definiti da Zachman: contestuale, concettuale, logico e fisico. Gli aspetti analizzati in BOAT invece corrispondono rispettivamente ai seguenti aspetti nel framework di Zachman: Persone (chi) per i Partecipanti, Dati (cosa) per gli Oggetti, e Tempo (quando).

I diversi aspetti considerati in BOAT vengono analizzati secondo le varie prospettive seguendo una metodologia e con l'ausilio di diversi diagrammi. In particolare, è opportuno sottolineare che per le prospettive Architettura e Tecnologia si delineano due componenti di diversa natura:

*Architettura* Un gruppo di componenti statici e dinamici (e con logica mista fra questi due tipi) che garantiscono l'allineamento fra strategia aziendale e attività operative.

*Infrastruttura tecnologica* Un insieme di tecnologie hardware, software e di rete che possono supportare le operazioni aziendali permettendo l'uso di risorse e servizi.

Gli aspetti relativi ai componenti tecnologici a livello applicativo, alle piattaforme software e alle infrastrutture fisiche sono discussi nella Parte III del volume.

Nel seguito l'approccio BOAT viene presentato in sequenza, partendo dagli aspetti di business fino agli aspetti architetturali, funzionali e tecnologici. E' però da sottolineare che in genere la progettazione avviene in modo iterativo, sviluppando l'EA in più iterazioni, come illustrato in Figura 3.2.

La Figura 3.3 evidenzia come nei progetti il punto di partenza può cambiare a seconda del tipo di progetto. Secondo molte teorie, la gestione del cambiamento (*change management*) è un elemento fondamentale per mantenere posizioni

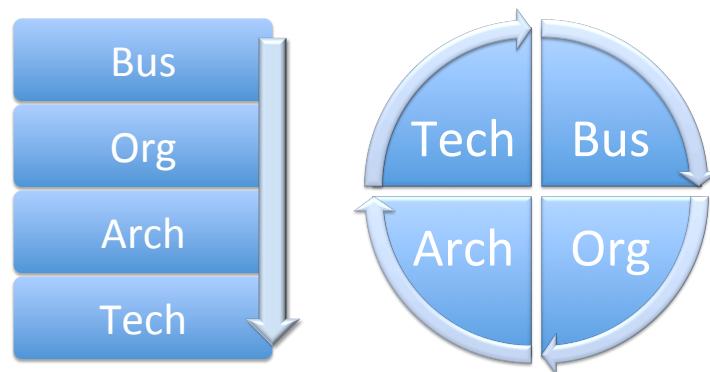


Figura 3.2: Sequenza e iterazioni della modellazione dei vari aspetti

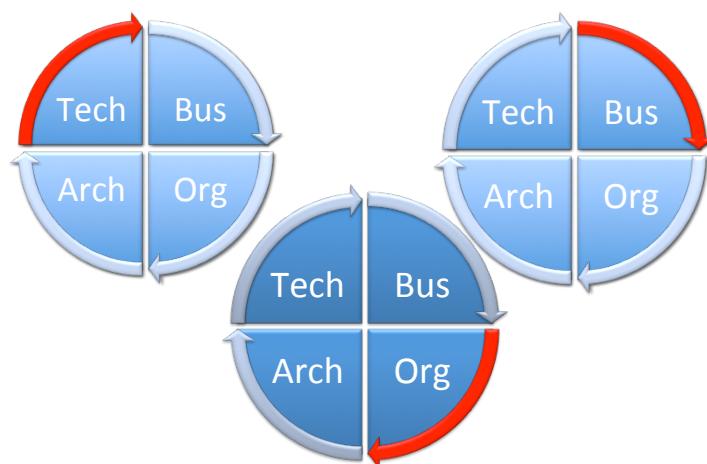


Figura 3.3: Tipi di iterazioni della modellazione dei vari aspetti

di successo sul mercato. I cambiamenti che causano l'evoluzione del sistema informativo possono avvenire per cause diverse:

*Esogene all'organizzazione*: per esempio, cambiano le leggi, o il mercato impone innovazioni tecnologiche.

*Endogene all'organizzazione*: per esempio, obsolescenza delle macchine, o nuove strategie.

Come già discusso nel Capitolo 1 queste cause possono portare alla ridefinizione del sistema informativo agendo su leve e prospettive diverse: le modifiche legislative richiederanno modifiche di aspetti organizzativi, mentre una nuova strategia richiederà di ripensare l'azienda a partire dagli aspetti di business. Un cambiamento dovuto al desiderio di adottare una nuova tecnologia dovrà necessariamente portare a una riflessione su quali sono gli aspetti di business che giustificano l'adozione di quella tecnologia, seguita da una riorganizzazione a livello organizzativo.

Da notare che, qualunque sia il punto di partenza, la completa ridefinizione del sistema informativo avverrà analizzando il sistema dalle altre prospettive, seguendo sempre lo stesso ordine (è definito modello a ruota e lo si utilizza procedendo in senso orario).

La presentazione di BOAT viene ora illustrata tramite alcuni esempi.

## 3.2 Esempi

In questa sezione verranno discussi i punti di partenza nello sviluppo dell'EA. Nelle sezioni successive, questi esempi verranno sviluppati secondo l'approccio e le notazioni proposte in BOAT.

### 3.2.1 Perfect Office Solutions House (POSH)

Perfect Office Solutions House (POSH) è una catena di negozi che vende soluzioni d'ufficio complete, che consistono di materiale come piccoli oggetti tipo o taccuini, fino a forniture e arredi completi, con complemento di servizi professionali e consulenza. POSH è fornitore sia di singoli consumatori che di aziende, focalizzandosi sul mercato 'high-end' – non competono sul prezzo ma sulla qualità. POSH è sul mercato da molti decenni. POSH ha 5 grandi negozi e 3 filiali sparsi nel paese. I negozi più grandi dispongono di prodotti fisicamente accessibili. Le filiali lavorano su cataloghi. L'ufficio centrale POSH e il magazzino principale sono co-locati con uno dei negozi grandi. POSH ha un volume d'affari pari a circa \$850 milioni all'anno e impiega circa 750 persone nel suo staff. Attualmente, vende sia nei propri negozi e filiali sia tramite venditori selezionati (terze parti verso cui POSH vende all'ingrosso). Osservando che l'e-business può portare nuove opportunità, POSH ha deciso di lanciare un nuovo piano aziendale a livello di azienda per aprirsi a nuove opportunità. Le direzioni individuate sono due:

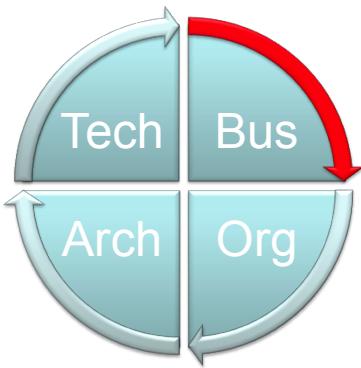


Figura 3.4: Modello a ruota di BOAT per POSH

1. Vendere direttamente a tutti i clienti nazionali senza l'intervento dei venditori terze parti, così da incrementare i margini di profitto o riuscire a fare offerte più competitive ad aziende più grandi abbassando i prezzi.
2. Offrire più servizi, come consulenze d'ufficio e gestione magazzino (in modo che i clienti possano esaminare che cosa hanno comprato esattamente da POSH e adattare i nuovi acquisti).

#### *Punto di partenza per POSH*

I progettisti dell'e-business di POSH riconoscono che entrare nell'e-business in rete non è un'azione che avviene velocemente e in un passo solo, ma richiede un'evoluzione costante. Anche se il mercato della fornitura d'ufficio forse non è fra i più dinamici, è relativamente nuovo nel campo dell'e-business e pertanto richiede notevole attenzione ai cambiamenti. Dato che POSH si rende conto che il progetto del proprio e-business dovrà evolvere in futuro, l'organizzazione sposa il modello a ruota di BOAT. Per il caso di POSH, lo sviluppo dell'e-business è chiaramente guidato dalla richiesta di nuove opportunità di business. Gli elementi di BOAT seguono i requisiti di business. Sicuramente, le nuove tecnologie dell'informazione non sono il driver di nuove opportunità di POSH - quelle esistenti offrono già più di quanto è necessario. Pertanto, l'aspetto B (Business) è il trigger dello sviluppo (come mostrato in Figura 3.4).

### 3.2.2 TalkThroughUs (TTU)

TalkThroughUs (TTU) è una nuova azienda che offre servizi di traduzione nel contesto dell'economia globale. Visto che le aziende tendono sempre più a fare business oltre i confini nazionali, le barriere linguistiche diventano un problema sempre più rilevante. Anche se molti parlano inglese nel mondo, non si tratta ancora di una lingua comunemente conosciuta in alcuni Paesi – e là dove è conosciuta, a volte la padronanza della lingua inglese è ancora troppo bassa per permettere di concludere affari agevolmente. Questo vale per la comunicazione

scritta e orale. Pertanto, TTU vuole fornire servizi di traduzione e interpretariato in internet. Chiama questo il concetto *IT<sup>2</sup>*: traduzione e interpretariato per mezzo della tecnologia dell'informazione. TTU cerca di offrire servizi linguistici fra qualsiasi coppia delle lingue più rilevanti nel mondo.

TTU è una azienda relativamente nuova e ancora in fase di espansione. Ha un ufficio centrale combinato con la divisione Research & Development (R&D, Ricerca e Sviluppo) che si occupa sia dello sviluppo di nuovi business model sia dell'introduzione delle nuove tecnologie dell'informazione nell'operatività di TTU. Oltre all'ufficio centrale, TTU ha attualmente 5 uffici internazionali nel mondo (progetta di espandere questo numero) e impiega 150 persone, di cui circa i due terzi operanti nella sede centrale. In aggiunta allo staff, TTU ha una rete di più di 1000 interpreti e traduttori free-lance nel mondo. Il volume di affari di TTU è cresciuto rapidamente da quando l'azienda ha iniziato; attualmente tale volume si aggira sui \$72 Milioni all'anno. I servizi di interpretariato di TTU permettono ai clienti di avere a disposizione intepreti di TTU che supportano gli incontri d'affari via rete digitale fra parti che non parlano una lingua comune a livello adeguato. Tali servizi si basano su interpreti ingaggiati in modo 'live' in incontri elettronici usando sistemi di conferenze digitali – basati sia su telefono sia video-conferenze. Per queste ultime, TTU fa affidamento su piattaforme di cloud computing. Poichè TTU si rivolge a un mercato in cui la privacy delle discussioni è un requisito fondamentale, la confidenzialità è un requisito importantissimo nei servizi forniti.

I servizi di traduzione di TTU permettono ai clienti l'invio di documenti in una certa lingua e la loro traduzione in altra lingua. Tali servizi si basano preferibilmente su scambi elettronici dei documenti: i clienti mandano documenti in input in formato digitale e nello stesso formato li ricevono indietro tradotti da TTU. TTU punta a tempi di molto brevi di traduzione dei documenti elettronici: i documenti brevi tipicamente vengono tradotti in un'ora. Per andare incontro ad esigenze più tradizionali e a documenti di tipo storico, TTU offre anche la traduzione di documenti fisici. Questi servizi di traduzione possono essere ordinati via internet. Naturalmente, il trasporto dei documenti fisici (via posta) comporta tempi di gestione molto più lunghi rispetto ai tempi dei documenti elettronici. TTU garantisce tempi di risposta veloci, anche in questo caso. Per documenti fisici urgenti, ad alta priorità, TTU usa una terza parte, un corriere di alta qualità per il trasporto fisico.

#### *Punto di partenza per TTU*

TTU capisce che muoversi nell'ambito dell'e-business in rete comporta un'innovazione costante. Il settore delle traduzioni aziendali in rete è un nuovo dominio di business, che richiede di adattarsi costantemente alle evoluzioni del mercato - che possono arrivare dalle esigenze clienti e dagli sviluppi portati avanti dalla concorrenza. Inoltre, il business model si avvale di tecnologie della comunicazione evolute, specie quando si tratta di traduzioni in tempo reale (certamente quando è richiesta la comunicazione audio/video o addirittura la telepresenza). Poichè questa tecnologia è in veloce sviluppo, comporta possibili cambiamenti al business model o all'organizzazione di TTU. In altre parole: il

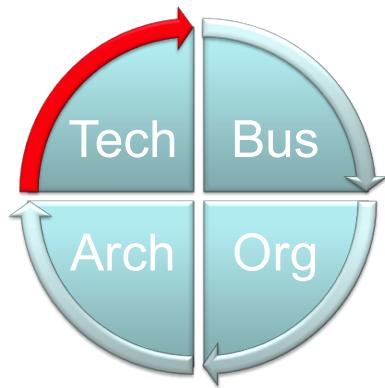


Figura 3.5: Modello a ruota BOAT per TTU

dominio di business di TTU è influenzato da forti componenti di ‘technology push’.

TTU adotta il modello BOAT come segue. L’aspetto T (Tecnologia) è il trigger dello sviluppo dell’e-business (come illustrato in Figura 3.5): la possibilità tecnica di fornire servizi di interpretariato e traduzione online porta all’identificazione di nuove opportunità di business. Le nuove tecnologie della comunicazione avranno grande influenza sulla natura e il modo di fornire i servizi, ovvero, le nuove opportunità di business e i nuovi obiettivi di business sono definiti in base alla disponibilità di nuove tecnologie.



# Capitolo 4

# Prospettiva di business

## 4.1 Descrizione

In questo capitolo si illustra come analizzare e modellare la prospettiva di Business nell'approccio BOAT [19].

Il risultato che si vuole ottenere è costruire un *modello di business* (nel seguito anche *business model*), che fornisce le motivazioni per la realizzazione di un intervento sul sistema informativo.

Come già discusso, la prospettiva di business può essere il primo passo nella pianificazione di un intervento sul sistema informativo (requirement pull), in alternativa all'analisi degli obiettivi di business quando il punto di partenza del progetto è una spinta tecnologica (technology push). L'aspetto di business viene presentato per primo per dare un ordine alla trattazione, ma va ricordato che la costruzione del modello di business può partire anche da una spinta tecnologica, oppure da una proposta di modifica organizzativa, che deve essere giustificata nell'ambito degli obiettivi generali dell'azienda.

Il modello di business viene quindi costruito identificando uno *scenario* per l'intervento sulla base dei business driver, che inserisce nella EA le modalità previste per l'interazione tra partecipanti. Si introduce nel seguito il concetto di *driver* a livello business, con una discussione dei principali tipi di driver: efficacia ed efficienza.

## 4.2 Dimensioni dell'interazione

Esaminiamo innanzitutto i principali componenti dell'interazione tra i partecipanti descritti all'interno di un modello di business e analizziamo le loro caratteristiche. Questi componenti sono alla base della definizione degli attori (partecipanti) che interagiscono tra loro tramite il sistema informativo, focalizzandosi sull'interazione dell'azienda con parti esterne. I componenti definiscono le modalità di interazione previste (ad esempio, un cliente interagisce con l'azienda per acquistare un libro on-line).

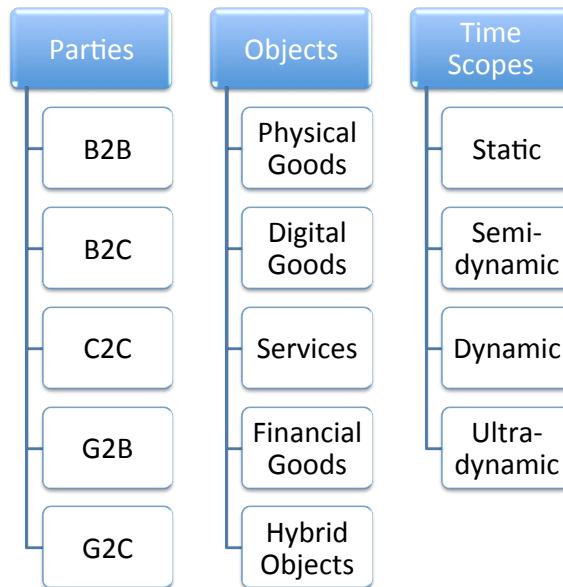


Figura 4.1: Schema dei valori nelle dimensioni della classificazione

Nella prossima sezione vengono esaminati alcuni casi di attualizzazione dei driver che prendono in esame situazioni differenti dal punto di vista delle interazioni tra i partecipanti.

Le dimensioni prese in esame in questa sezione sono: partecipanti, oggetti dell'interazione, orizzonte temporale.

#### 4.2.1 Partecipanti (party)

Le classificazioni proposte in letteratura sono svariate e differiscono in base alle diverse possibili tipologie di utenti. In Figura 4.1, vengono proposte alcune classificazioni basate sui seguenti termini:

- *Business (B)*: rappresenta aziende che forniscono informazioni e servizi in rete; in genere sono dotate di un sistema informativo a supporto dello svolgimento delle proprie attività. Esempi sono un'azienda manifatturiera, un operatore telefonico.
- *Government (G)*: rappresentano organizzazioni (amministrazioni pubbliche) che forniscono servizi di tipo pubblico, per esempio basati su politiche del territorio (alcuni esempi: Comune, Regione, Agenzia delle Entrate).
- *Consumatore oppure Cittadino (C)*: rappresentano utilizzatori dei servizi forniti da altri partecipanti.

Si parla di applicazioni *B2B* (Business to Business) se l'applicazione è rivolta alla cooperazione tra imprese mentre si parla di applicazioni *B2C* (Business to

Consumer) per applicazioni (tipicamente di commercio elettronico) rivolte al cliente.

Nel caso in cui gli enti che erogano il servizio siano Pubbliche Amministrazioni, si hanno le varianti *G2C* (Government to Citizen) o *G2G* (Government to Government) per interazioni tra pubbliche amministrazioni. Le interazioni *G2B* rappresentano servizi forniti dalle amministrazioni pubbliche alle aziende, in cui è previsto uno scambio di informazioni sistematico.

Un caso particolare sono le interazioni *C2C*, in cui il sistema favorisce o supporta l'interazione diretta tra clienti o cittadini.

#### 4.2.2 Oggetti dell'interazione

Una seconda dimensione dell'interazione riguarda la tipologia di oggetti scambiati tra le parti nell'interazione. Come indicato in Figura 4.1, si identificano le seguenti tipologie di scambio:

- *Prodotti fisici*: nell'interazione si scambiano oggetti fisici (ad esempio, un libro, un vestito, ecc.).
- *Prodotti digitali*: vengono scambiati prodotti quali documenti elettronici, estratti conto on-line, e-book, file musicali.
- *Servizi*: vengono forniti servizi, quali un servizio telefonico (voice-over-IP), una prenotazione di un volo, e così via.
- *Prodotti finanziari*: vengono scambiati tra le parti prodotti finanziari, quali ad esempio azioni di una società quotata in borsa.
- *Oggetti ibridi*: lo scambio consiste in oggetti di natura mista, tra quelli visti sopra, per esempio, uno stock di azioni e la corrispondente scheda di prodotto sotto forma di documento on-line.

#### 4.2.3 Orizzonte temporale

La terza dimensione di analisi considerata riguarda il tempo e in particolare l'orizzonte temporale dell'interazione.

Esso indica la stabilità della relazione tra le parti. Più che indicare una durata, viene indicato se le transazioni che vengono effettuate, qualunque sia la loro durata, sono tra partecipanti che si pongono in relazione fissa tra loro oppure se questa relazione è variabile nel tempo. La Figura 4.1 riporta le seguenti tipologie di orizzonti temporali per l'interazione:

- *Statico*: i partecipanti stabiliscono una relazione stabile nel tempo, solitamente regolata da un contratto che ne stabilisce le modalità e condizioni. Si tratta di fornitori di prodotti o servizi abituali per l'organizzazione, oppure clienti abituali, che ordinano sistematicamente gli oggetti all'organizzazione. In genere, i pagamenti non vengono effettuati per la singola transazione, ma periodicamente sulla base della quantità di oggetti scambiati.

- *Semi-dinamico*: la relazione è stabile, ma sono possibili cambiamenti di tanto in tanto. In questo caso si ha la modifica del fornitore abituale, ma le forniture sono comunque trattate in gruppo e non singolarmente per quanto riguarda ordini e pagamenti.
- *Dinamico*: la relazione in questo caso si considera limitata al singolo ordine da un cliente al fornitore. Per ordini diversi si considerano possibili fornitori diversi.
- *Ultra-dinamico*: la relazione può variare all'interno del singolo ordine. Ad esempio, se si prevede un ritardo nella consegna, si cambia fornitore per avere la sicurezza di avere l'oggetto entro la scadenza desiderata.

L'unità di base per la definizione dell'orizzonte temporale è quindi la transazione tra le parti e l'orizzonte temporale è relativo a quante transazioni vengono effettuate tra gli stessi partecipanti.

### 4.3 Business driver

I *business driver* (*fattori chiave per l'azienda*) sono quelli che motivano il progetto del sistema informativo o la sua evoluzione identificando uno o più obiettivi generali per il progetto.

I business driver vengono distinti in due categorie: l'*efficacia*, obiettivi di tipo strategico per l'azienda, legati alle sue attività, e l'*efficienza*, che ugualmente può assumere un valore strategico in alcuni settori di attività, ed è legata ad un migliore uso delle risorse per svolgere le attività stesse.

Un aspetto comune a entrambi i driver è l'esigenza di rendere questi obiettivi misurabili, tramite *indicatori* (raggruppabili in Key Performance Indicator - KPI, ovvero indici che ne monitorano l'andamento), che vengono definiti per poter valutare a posteriori se i miglioramenti previsti sono stati raggiunti e per identificare eventuali criticità.

Nel definire gli obiettivi, si farà riferimento al modello presentato nelle Figure 4.2 e 4.3, caratterizzato da tre insiemi di variabili: input, output effettivo e output atteso.

Gli input fanno riferimento alle risorse necessarie per svolgere le attività nell'azienda. Come già illustrato nel Capitolo 1, le risorse possono essere di varia natura: materiali (materie prime), risorse umane, e immateriali (ad esempio, informazioni, servizi, o risorse finanziarie). Quando si parla di output, si intende il risultato da raggiungere. Nel caso di un'azienda con processi produttivi, l'output misura la quantità di beni prodotta. Per aziende di servizi, come ad esempio una compagnia assicurativa, l'output possono essere le pratiche di rimborso gestite o le nuove polizze stipulate; per un'azienda telefonica, i contratti stipulati o il numero di utenti serviti. L'output effettivo misura la quantità effettivamente ottenuta, mentre l'output atteso è una misura previsionale, che indica l'obiettivo da raggiungere.

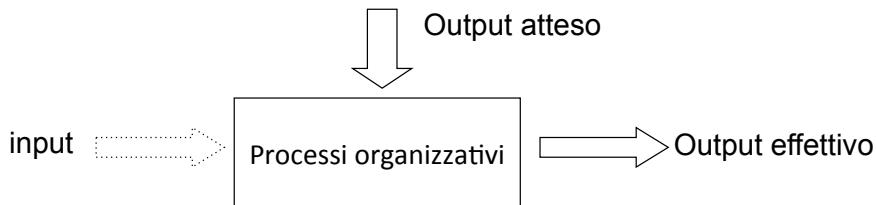


Figura 4.2: Efficacia

### 4.3.1 Tipi di driver

#### Efficacia

La definizione di efficacia misura la capacità dell'azienda di raggiungere obiettivi strategici e può essere definita come (Figura 4.2):

$$\text{Efficacia} = \text{Output-effettivo} / \text{Output-atteso}$$

dove l'output atteso definisce un obiettivo strategico da raggiungere e l'output effettivo ne misura il raggiungimento.

Esempi di indicatori di efficacia sono i seguenti:

- Aumentare del 5% la quota di mercato entro la fine dell'anno.
- Raggiungere nuove categorie di clienti; vanno definiti opportuni indicatori e KPI, che consentano di valutare il suo raggiungimento. Ad esempio, segmentando i clienti per fasce d'età, si può voler raggiungere l'obiettivo di avere clienti di una fascia d'età che non utilizzavano il prodotto in precedenza; oppure raggiungere clienti in nuove località, e anche in questo caso andrà definito un indicatore per misurare il risultato, quale il numero di nuove località, oppure il numero di clienti in nuove località.
- Ridurre il numero di reclami.

Nella valutazione dell'efficacia distinguiamo le dimensioni di *estensione* (*reach*) e le dimensioni di *ricchezza* (*richness*).

**Estensione (reach)** Con l'estensione si cattura il raggiungimento di un obiettivo che allarga l'ambito di attività dell'azienda, da un punto di vista spaziale, temporale, oppure nei canali di comunicazione con i clienti.

Vengono definite le seguenti dimensioni di *estensione* (*reach*):

- *Geografica*: Ad esempio da locale a nazionale.
- *Temporale*: ad esempio, aumentare l'orario o il numero di giorni di apertura, fino ad arrivare a una copertura di 24 ore per sette giorni alla settimana (24/7).

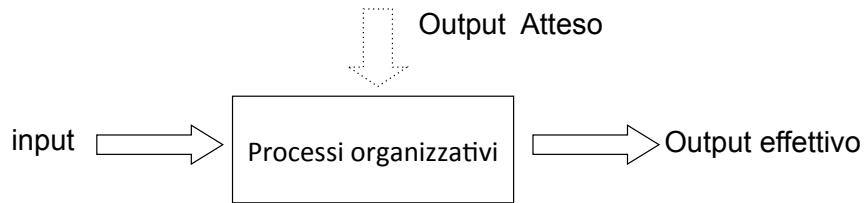


Figura 4.3: Efficienza

- *Modalità (canali)*: la multicanalità è un aspetto che caratterizza l'interazione per molte categorie di servizi. L'interazione può avvenire con canali tradizionali (negozi, sportelli), ma anche con diversi tipi di canali on-line, come Web, email, instant messaging, social media (ad esempio, FaceBook). Un'estensione può essere l'affiancamento dei canali on-line di tipo tradizionale (web, email) con i social media, caratterizzati da tempi e modalità di gestione molto differenti, in quanto richiedono un presidio costante del canale per rispondere alle richieste dei clienti in tempi brevi.

**Ricchezza (richness)** Con la ricchezza, si misura l'intensità della comunicazione, ovvero quanto sono intense le relazioni dell'azienda con le parti con cui interagisce (per es., clienti e fornitori). Per aumentare la dimensione ricchezza, si può agire su fattori quali la frequenza delle interazioni, il livello di interattività, il dettaglio, la tipologia dei dati utilizzati (immagini, video, ...), e la personalizzazione delle interazioni.

### Efficienza

La definizione di efficienza fa riferimento al rapporto tra output effettivo e input (si veda Figura 4.3):

$$\text{Efficienza} = \text{Output-effettivo} / \text{Input}$$

dove l'Input è misurato in termini delle risorse necessarie per ottenere l'output.

Anche l'efficienza può essere vista come obiettivo strategico. Un aumento di efficienza si può ottenere in due modi: ottenendo lo stesso risultato con risorse ridotte, oppure aumentando l'output a parità di risorse. Esempi di aumento di efficienza sono la riduzione dei costi o dei tempi di produzione a parità di risorse impiegate.

Possiamo esaminare meglio l'efficienza considerando un'organizzazione come una struttura a tre livelli basandoci sul modello di Anthony visto nell'introduzione. In un'organizzazione tradizionale, la dimensione dei livelli in termini di personale coinvolto (o *effort*) è mostrata a sinistra in Figura 4.4: il livello operazionale è il più ampio e quello strategico il più piccolo, considerando la forma a piramide. Aumentare di molto l'efficienza spesso si traduce nell'automatizzare

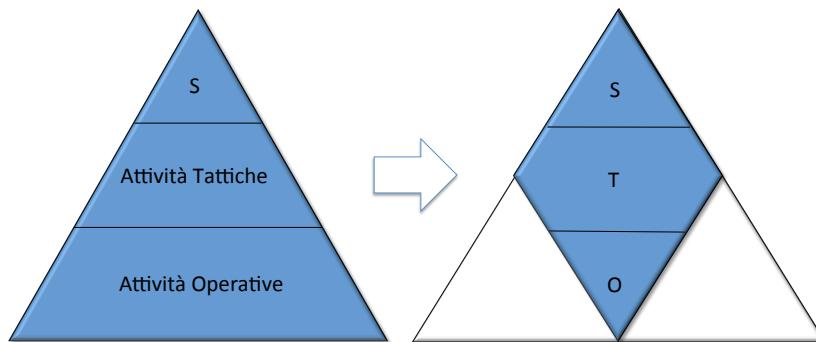


Figura 4.4: Efficienza e impatti sulla struttura organizzativa

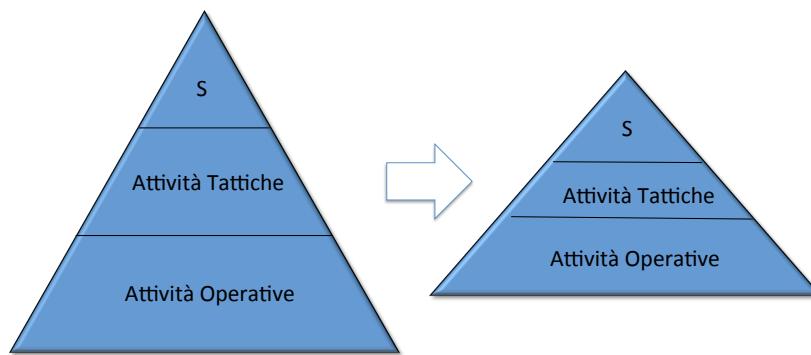


Figura 4.5: Efficienza e impatti sulla struttura organizzativa - disintermediazione tra livelli

le funzioni a livello operazionale: i compiti routinari vengono completamente automatizzati e l'intervento umano va principalmente nella gestione delle eccezioni, quindi la dimensione del livello operazionale viene notevolmente ridotta. Questo può cambiare la forma della piramide in una forma a diamante come mostrato a destra nella figura, in cui il livello tattico è il più ampio. In questo caso, un compito importante del livello tattico è quello di configurare continuamente i parametri del sistema automatizzato che esegue i processi operazionali, così che questi funzionino nel modo più autonomo possibile.

Un aumento di efficienza può avere come effetto anche una riduzione dei livelli intermedi nell'organizzazione (Figura 4.5), con la riduzione dei livelli di controllo necessari, in quanto favorisce la velocità e l'analisi dei flussi di informazione tra i diversi livelli dell'organizzazione. Ne risulteranno semplificate le attività di controllo svolte a livello operativo e le attività di gestione svolte a livello manageriale di controllo e pianificazione, che vengono semplificate dalla disponibilità di dati aggiornati ed aggregati per prendere le decisioni e pianificare le attività.

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

Figura 4.6: Principali motivazioni (driver) a livello business e direzioni di business

## 4.4 Driver e loro attuazione

### 4.4.1 Driver operazionalizzati e direzioni di business

In Figura 4.6, vengono riassunti i principali business driver presi in esame e per ciascuno vengono definiti alcuni possibili corrispondenti driver effettivi, o operazionalizzati (*Operationalized driver*) e le direzioni di business (*Business direction*).

Per operazionalizzare l'**estensione**, possiamo aumentare la disponibilità delle applicazioni di e-business e aumentare l'accessibilità delle applicazioni. Disponibilità significa essere on-line nei momenti richiesti (cosa che spesso vuol dire sempre), ma anche avere disponibili le giuste risorse (compresa l'informazione), cioè essere pronti al momento giusto. Questo porta a una direzione di business che etichettiamo come business *on-time e on-line*. L'accessibilità nell'e-business significa permettere l'uso dei giusti canali nelle giuste circostanze, ovvero, avere tanti canali di collaborazione disponibili in varie situazioni. Etichettiamo questo con il termine direzione *multicanale*.

La **ricchezza** può essere operazionalizzata in vari modi. Ne sceglio due, di grande rilevanza pratica nell'e-business in rete. Il primo modo consiste nell'uso di informazione ricca sui clienti per creare un alto livello di *customer intimacy*, per legare un cliente a uno scenario di e-business nell'ambito di un mercato competitivo. Chiamiamo questa direzione *gestione aumentata della relazione cliente* (enhanced CRM - Customer Relationship Management). Il secondo modo consiste nel fornire al cliente una transizione fluida tra la parte fisica e quella digitale del business (sia durante le transazioni singole sia durante una sequenza di transazioni di e-business nel tempo) e chiamiamo questo *transizionalità*. Per ottenere la transizionalità, un'organizzazione ha bisogno di integrare strettamente

mente le proprie operazioni fisiche e digitali, spesso etichettate come *bricks and clicks integrati* (che sceglieremo come etichetta per la direzione di business).

**L'efficienza**, come terzo business driver, può essere operazionalizzata secondo due dimensioni: costo e tempo. Mettiamo in relazione il driver operazionalizzato dell'efficienza di *costo* con una direzione di business che mira a minimizzare i costi delle operazioni di business applicando l'automazione al più alto livello possibile, ovvero quello dell'automazione completa. Il driver operazionalizzato dell'efficienza di *tempo* fa riferimento alla direzione di business che mira a ridurre i tempi di throughput delle operazioni di business (esecuzione delle transazioni): direzione di business detta di compressione temporale (time-compressed).

La Figura 4.6 illustra una vista delle sei direzioni di business identificate. Nel seguito, discutiamo ciascuna direzione in modo più dettagliato. Per le stesse direzioni, vedremo che la caratteristica dell'e-business moderno ha grande impatto sul modo in cui quella direzione può essere realizzata: lo scopo di una direzione di business può andare al di là della singola organizzazione della rete di e-business.

#### 4.4.2 Direzioni di business

##### Direzione on-time e on-line

La ‘old economy’ si basa su canali di comunicazione lenti e asincroni, quali i servizi postali tradizionali. Ciò significa che l’interazione diretta fra parti è molto limitata e i margini temporali sono inseriti nei processi per tener conto di questo. Nell’e-business in rete, possiamo avere interazione diretta fra partner usando canali elettronici come internet. Questo apre la strada al business on-line, ovvero, business in cui le parti interagiscono usando strumenti digitali. Il business on-line permette una sincronizzazione più stretta fra partner, permettendo di aver processi on-time – anche detti processi real-time. In vari domini, l’aspetto real-time è molto importante, per esempio nelle transazioni di borsa. Di conseguenza, le strutture di business possono essere costruite con accoppiamento molto più stretto (portando al business ‘just-in-time’) oppure si possono abilitare scenari con valori più bassi per la dimensione tempo.

Nei primi tempi dell’e-business (fine del secolo scorso), essere ‘davvero on-time e on-line’ era un obiettivo primario e, a volte, un punto decisivo per il successo nel collocare sul mercato uno scenario. L’uso di un’interfaccia web non implica necessariamente la capacità di gestire transazioni completamente on-line: l’interfaccia web spesso era solo una facciata rispetto alla gestione manuale collocata dietro di essa. Questo significa che la modalità ‘davvero on-line’ non si è ottenuta e, se la gestione manuale seguiva il tipico orario d’ufficio, non si era ottenuto nemmeno il ‘davvero on-time’. Oggi, è sempre più considerato come scenario standard che le transazioni siano completamente eseguite on-line e in real-time. Un esempio è l’ambito turistico, in cui le prenotazioni dei viaggi aerei e degli alberghi sono diventate completamente interattive secondo una transizione avvenuta in modo quasi automatico. Nella transizione da business ‘tradizionali’ a e-business in rete, essere on-time e on-line pone una

nuova sfida: il raggiungimento di tali obiettivi non è determinato da una singola organizzazione centrale, ma dall'intera rete che fornisce un oggetto complesso di e-business a un cliente. Se un partecipante della rete non è on-time e on-line, l'intera rete non lo è dal punto di vista del cliente (perché l'oggetto non può essere consegnato). In altri termini: per la direzione di business, una rete è forte tanto quanto lo è il suo partecipante più debole.

### **Business multi-canale**

Introdurre l'e-business spesso vuol dire mettere in atto molti canali comunicativi tra i partecipanti. Ciò è senz'altro vero nel caso dell'approccio 'bricks and clicks' (discusso più avanti nella sezione), in cui i canali di comunicazione tradizionali (per esempio, posta, telefono) vengono realizzati su canali digitali (per esempio, un sito web). Ma anche nell'e-business puro, possono esistere vari canali, per esempio e-mail, moduli (form) basati sul web, sistemi di risposta con voce digitale, e così via. Dal punto di vista del business, vi sono due aspetti fondamentali quando si usano canali multipli in parallelo: la flessibilità e la sincronizzazione. La flessibilità dovrebbe permettere ai partner di cambiare il canale su cui stanno operando senza disturbare le relazioni di business tattiche, o perfino cambiare il canale nell'ambito di un singolo ordine di business senza alterare i processi operazionali. La sincronizzazione va considerata nel progetto dell'attività di business per essere sicuri che le transazioni siano condotte su più di un canale in parallelo, continuando a fornire risultati congruenti. Ci riferiamo alla messa in esecuzione dell'e-business considerando la multicanalità in fase di progettazione.

La progettazione multi-canale può essere necessaria per vari motivi. Un motivo è l'esistenza di vari gruppi di clienti, ciascuno con le proprie preferenze riguardo a un canale primario. Lo si osserva per esempio nelle aziende con ordini gestiti via mail, in cui i clienti senior possono preferire canali di consegna tradizionali mentre i clienti giovani possono preferire canali di consegna digitali. Un'altra ragione è l'uso di vari tipi di strumenti di comunicazione. Vediamo per esempio l'uso di vari tipi di dispositivi mobili usati a seconda dell'effettivo contesto fisico di un utente - e un utente può cambiare canale 'al volo' al mutare del contesto in cui si trova ad operare.

### **Gestione avanzata della relazione con il cliente**

Nella 'old economy', la natura dei mezzi di comunicazione tipicamente realizza comunicazioni poco frequenti e a larga granularità fra i partner. Il costo di comunicazioni off-line, manuali e della gestione dei dati impedisce la gestione di informazioni dettagliate e aggiornate. Questo rende la relazione con il cliente (già precedentemente definita come Customer Relationship Management - CRM) piuttosto limitata. Negli scenari di e-business, l'interazione fra partner può essere a basso costo, veloce e frequente. La raccolta dati on-line permette di gestire e usare informazioni molto dettagliate e aggiornate sui partner. Questo vale sia per i partner singoli sia per gruppi o classi di partner. Questo tipo di gestione

dei dati cliente si basa sulle tecnologie dei ‘big data’, i cui sviluppi spalancano la porta a forme completamente nuove di CRM e pertanto a nuove forme di business che si basano su una relazione stretta e di alto livello con il cliente (detta customer intimacy). Esempi esistono nelle applicazioni di vendite elettroniche che permettono di comporre dinamicamente le offerte ai clienti, sulla base del loro comportamento attuale e passato. Si noti che nella vendita tradizionale, tipicamente, vengono registrate solo le transazioni (vendite), arrivando ad ottenere una immagine ampia e generale del comportamento cliente. Nelle vendite elettroniche, è invece possibile registrare in modo molto dettagliato il comportamento completo dei clienti, per esempio, la loro storia di browsing dei cataloghi, anche quando le azioni del cliente non portano a nessuna transazione. Il comportamento dei clienti può essere confrontato con quello di clienti simili per suggerire o addirittura predire un comportamento futuro. Ovviamente, la privacy dei clienti B2C in ambiente digitale [1] è un aspetto importante in questo contesto.

### ‘Bricks and clicks’ integrato

Molte organizzazioni hanno già stabilizzato un loro comportamento, o footprint, nella ‘old economy’. Riconoscere che l’e-business in rete offre nuove possibilità non vuol dire necessariamente abbandonare completamente questa presenza già stabilizzata, perché ciò vorrebbe dire una perdita sostanziosa di una base di clienti. In questo contesto, il ‘vecchio’ business è spesso chiamato bricks, mentre il ‘nuovo’ business è detto clicks [25]. È necessario integrare ‘vecchio’ e ‘nuovo’ business, ottenendo quanto è detto ‘bricks and clicks’ integrato.

In questa integrazione, vi sono due aspetti importantissimi. Primo, deve esserci sinergia fra i bricks e i clicks, così che uno offra valore aggiunto all’altro (per esempio, offrendo servizi aggiuntivi ai partner). Secondo, la demarcazione fra bricks and clicks deve essere flessibile, così che le attività sia del cliente sia dell’azienda possono transitare dall’uno all’altro senza disturbare le operazioni di business. Generalmente, lo spostamento va da bricks a clicks, ma non necessariamente (se uno scenario clicks fallisce, deve essere disponibile uno scenario bricks di backup negli stessi segmenti di business). Una direzione con bricks and clicks integrati è accoppiata al progetto di business multi-canale. In questo caso, parte dei canali fa fronte alla parte bricks e parte si rivolge ai clicks.

Un’area in cui il ‘bricks and clicks’ integrato è stato di grande importanza per il successo aziendale è quello delle società di vendita per corrispondenza. Esse hanno iniziato in modo totalmente bricks, con comunicazioni basate su cataloghi cartacei e formulari per gli ordini. Si sono poi spostate nell’ambito clicks introducendo negozi web con cataloghi e modalità d’ordine in rete. Dato che queste società spesso si rivolgono a vari segmenti di clientela con vari tassi di adozione di internet, mantenere in parallelo l’operatività ‘bricks and clicks’ è essenziale. Alcune parti del business, come la restituzione delle merci [10], mantengono elementi bricks e quindi richiedono opportune integrazioni.

### **Business completamente automatizzato**

Nell'e-business in generale, la tecnologia dell'informazione è usata per automatizzare le funzioni che, nelle configurazioni tradizionali, sono eseguite dalle persone. Quindi, la quantità di intervento umano nei processi è ridotto, tipicamente per ridurre i costi. In casi estremi di e-business in rete, si può tentare di arrivare ad operazioni (al livello operazionale) che non richiedono alcun intervento umano – così, lo sforzo umano è richiesto solo per questioni gestionali (ai livelli tattico e strategico). Chiamiamo questa direzione business completamente automatizzato, versione estrema dell'aumento di efficienza discusso in Sezione 4.3, che effettivamente ‘cancella’ il livello di controllo operazionale, come mostrato in Figura 4.4. L'automazione completa in uno scenario e-business è ottenibile solo se non vi sono operazioni di manipolazione di oggetti fisici, ma solo di oggetti digitali o servizi, oppure se la gestione fisica è completamente data in outsourcing a fornitori di servizi che non sono considerati parte dello scenario di e-business. Spesso, tuttavia, non è possibile ottenere una completa automazione dei processi operazionali: può essere necessario ricorrere agli operatori umani per gestire le eccezioni, ovvero casi non programmati. Ma per oggetti digitali o per servizi altamente standardizzati, l'automazione completa è ottenibile, in linea di principio. Si noti che anche l'operatività e la manutenzione delle strutture di calcolo può essere data in outsourcing (per esempio, ricorrendo al Cloud Computing, come discusso nel Capitolo 3), in modo che l'organizzazione possa focalizzarsi completamente sulle attività tattiche e strategiche.

### **Business compresso temporalmente**

Integrando elementi di business ‘on-time’ e ‘on-line’ e le direzioni di business completamente automatizzato, si può ottenere la direzione del business compresso temporalmente. Questo significa che si possono utilizzare modelli di e-business basati su transazioni che vengono eseguite in una frazione ridotta di tempo rispetto ai modelli di business tradizionali. Per esempio, le transazioni tradizionali, che tipicamente richiedono giorni per essere completate, possono essere compresse in transazioni di e-business che vengono portate a termine nell'ordine dei secondi. Lo stesso accade per processi multi-transazione. La compressione temporale è permessa dal fatto che le reazioni dei partner nell'esecuzione possono essere praticamente istantanee (poiché essi sono sempre on-line e possono automatizzare i compiti decisionali). Questa direzione permette di ottenere modelli di business con elevati livelli di comportamento ‘just-in-time’ e con partnership a breve durata. Esempi pratici di scenari di e-business a compressione temporale estrema si trovano nei mercati finanziari per azioni e valute, dove i partecipanti che agiscono velocemente traggono vantaggi rispetto a chi si muove più lentamente. Esempi estremi di scenari a compressione temporale si trovano spesso in business che operano solo con oggetti digitali (si veda la Sezione 4.2). Ciò nasce dal fatto che gestire oggetti fisici richiede il tempo imposto dai vincoli fisici di gestione degli oggetti. Per esempio, le merci devono essere trasportate su distanze fisiche, cosa che richiede un tempo

che ha limiti fisici di compressione. Le direzioni di compressione del tempo e della automazione completa possono presentarsi anche contemporaneamente in uno scenario di e-business. Tuttavia, non si tratta degli stessi aspetti, perché la compressione temporale mira alla riduzione dei tempi di esecuzione, mentre l'automazione completa mira alla riduzione dei costi (si veda Figura 4.6). Come con la direzione on-time e on-line, l'aspetto rete è molto importante per la direzione compressione temporale: di solito, non ha molto senso comprimere i tempi di esecuzione di una sola organizzazione se le altre rimangono ancorate alla loro vecchia velocità di esecuzione (a meno, naturalmente, che quella organizzazione non sia un collo di bottiglia).

## 4.5 Partecipanti e scenari

Le relazioni tra organizzazioni sono complesse e la filiera di gestione della produzione di un prodotto o di un servizio ha in genere numerosi partecipanti. Gli obiettivi dei partecipanti possono essere diversi, così come possono differire anche l'organizzazione interna e l'utilizzo della tecnologia dell'informazione.

Per ottenere una descrizione ben strutturata, vi è l'esigenza di separare i diversi punti di vista relativi alle interazioni.

Si definiranno quindi diversi *scenari* di interazione tra i partecipanti, ciascuno focalizzato al raggiungimento di determinati obiettivi. In questo modo, sarà possibile prendere in esame separatamente le interazioni tra i diversi partecipanti.

Gli scenari sono definiti come rappresentato in modo astratto nella Figura 4.7. Essi coinvolgono un numero limitato di partecipanti, solitamente le due parti, oppure le due parti e un intermediario che facilita la loro relazione (ad esempio, un'istituzione bancaria per fornire servizi di pagamento tra le parti). Ad esempio, in Figura 4.7, se l'azienda coinvolta nella progettazione è l'azienda indicata con B, questa è coinvolta negli Scenari 1 e 2. Lo scenario 3 non è di nostro interesse, perché riguarda solo l'interazione tra le parti D ed E. Nello Scenario 1, l'attenzione è focalizzata sull'interazione tra A e B, mentre nello Scenario 2 l'interazione è tra B, C e D. Ad esempio, B può svolgere il ruolo di fornitore, D quello di consumatore, mentre C quello di intermediario.

Queste strutture di aziende in rete e le loro interazioni posso variare dinamicamente. Un riferimento può essere la collocazione dell'azienda nella catena del valore, ottenuta collegando tra loro i modelli dei processi aziendali delle singole aziende secondo l'approccio di Porter, in cui ogni elemento indica la posizione dell'azienda come consumatore e come fornitore nella catena di creazione del valore (ad esempio, si veda Figura 4.8). Nell'esempio, dove l'azienda in esame è indicata in colore più scuro, l'azienda è fornitrice di un consumatore che a sua volta è fornitrice di un ulteriore consumatore e così via. Tali strutture si possono anche aggregare e separare dinamicamente fornendosi reciprocamente servizi ai fini di raggiungere l'obiettivo definito.

In Figura 4.9, si evidenziano inoltre due situazioni possibili riguardo i flussi informativi lungo la catena del valore: nel primo caso (figura in alto), i flussi

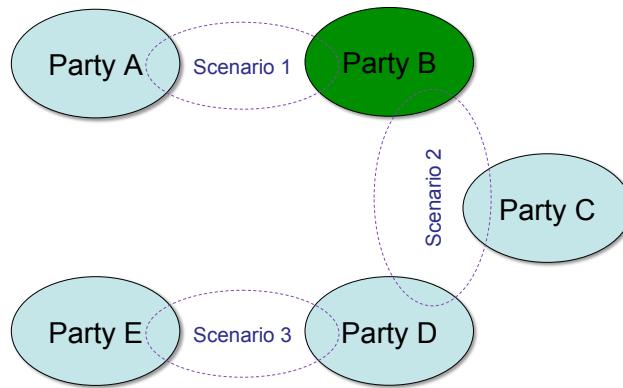


Figura 4.7: Partecipanti e scenari



Figura 4.8: Esempio di catena del valore

informativi seguono la produzione dei prodotti, dalle fasi iniziali di acquisizione dei materiali alla realizzazione del prodotto finito. Nel secondo caso (figura in basso), la produzione e tutte le attività collegate ad essa necessarie vengono attivate sulla base delle richieste dei clienti, fornendo oggetti personalizzati.

## 4.6 Tabella per il modello di business

Concludiamo la trattazione degli aspetti di business riportando in Figura 4.10 una tabella riassuntiva che rappresenta il *modello di business* che utilizziamo per rappresentare le dimensioni dell’interazione tra i partecipanti e i principali driver e le direzioni di business identificate per il loro raggiungimento.

## 4.7 Esempi

### 4.7.1 POSH

Classifichiamo lo scenario POSH usando le dimensioni introdotte in questo capitolo (si veda la Figura 4.11).

Considerando la dimensione relativa ai partecipanti dell’e-business in rete, classifichiamo POSH come sia B2B sia B2C: POSH vende sia a clienti singoli che ad aziende. Non includiamo B2G nella nostra analisi dello scenario: an-

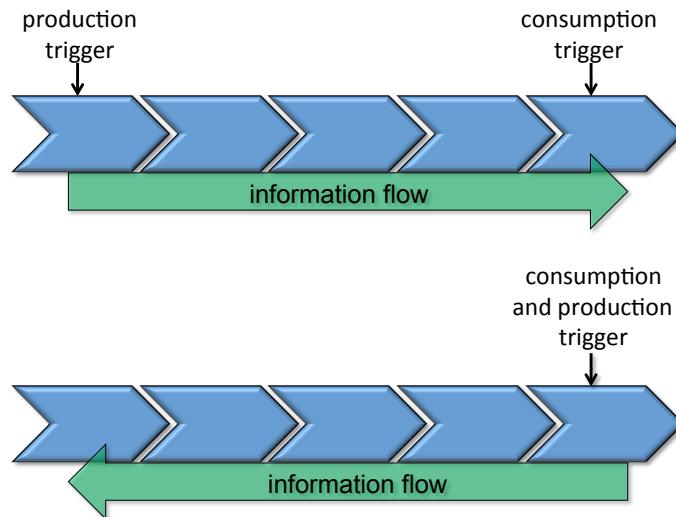


Figura 4.9: Supply e demand chain

che se POSH vende arredi a organizzazioni governative, queste ultime sono considerabili organizzazioni come le altre nelle transazioni verso POSH.

Gli oggetti che POSH tratta sono principalmente beni fisici, più precisamente beni discreti (forniture di ufficio, materiale e arredi). Ma POSH fornisce anche servizi attorno a questi beni fisici, per cui abbiamo un caso anche di oggetti ibridi. I servizi includono per esempio la manutenzione di strumenti e arredi. Nello scenario POSH, vediamo anche beni finanziari, quando i clienti pagano i beni acquistati - ma questi beni sono di marginale importanza per la nostra analisi.

Nella dimensione relativa al time scope, POSH lavora in scenari semi-dinamici e dinamici. Il time scope semi-dinamico è relativo a collaborazioni a progetto in cui un certo numero di ordini correlati vengono collocati dallo stesso cliente – qui, una sequenza di transazioni viene eseguita nel contesto di una relazione di business a lunga durata. Il time scope dinamico è applicabile agli acquisiti individuali, spesso da parte di clienti non-business del segmento B2C.

#### *Direzioni (driver)*

POSH ha due nuovi business goal:

1. Vendere a tutti i clienti direttamente, senza intervento di rivenditori terze parti.
2. Offrire uno spettro più ampio di servizi di consulenza ai clienti.

Questi business goal possono essere messi in relazione ai business driver come li abbiamo discussi nella Sezione 4.3. Il *Business goal 1* è il caso tipico di

## Business Model: <specify name>

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

Figura 4.10: Tabella per descrivere il modello di business

| Case study: POSH |                                    |  |
|------------------|------------------------------------|--|
| Parties          | B2B<br>B2C                         |  |
| Objects          | Physical objects                   |  |
| Time Scope       | Semi-dynamic<br>Dynamic            |  |
| Drivers          | Increasing reach                   | mainly geographical,<br>temporal secondary |
| Directions       | Enriched CRM<br>On-time and online |  |

Figura 4.11: POSH - aspetti di business

incremento di estensione geografica. Quando si pensa a incrementare l'estensione, POSH capisce che dovrebbe anche accrescere l'estensione temporale, perché i proprietari di piccole aziende possono voler concludere affari con POSH al di fuori delle normali ore di ufficio. Il *Business goal 2* è il caso di incremento della ricchezza: POSH vuole avere più interazione con i clienti per legarli a sé in modo più forte.

In termini di nuove direzioni di business, il CRM aumentato è applicabile a POSH per aumentare la customer intimacy. Essere on-time e on-line è una direzione di business (come conseguenza dell'aumento nella ricchezza temporale).

### 4.7.2 TTU

Classifichiamo ora lo scenario TTU usando le dimensioni di classificazioni di questo capitolo (si veda la Figura 4.12).

| Case study: TTU   |   |                                    |
|-------------------|---|------------------------------------|
| <b>Parties</b>    | B2B   | B2C possible too, but not main aim |
| <b>Objects</b>    | Digital services                                |                                    |
| <b>Time Scope</b> | Dynamic   |                                    |
| <b>Drivers</b>    | Increasing reach                                | geographical + temporal + modal    |
| <b>Directions</b> | On-time and on-line<br>Time-compressed business |                                    |

Figura 4.12: TTU - aspetti di business

Nella dimensione ‘partecipanti’, TTU è un tipico caso B2B: si tratta di una organizzazione che interagisce con altre organizzazioni. Può a volte lavorare anche per i clienti singoli (per esempio, persone che vogliono la traduzione di documenti privati importanti), ma questi aspetti non sono la base del modello di business.

Nella dimensione ‘oggetti’, TTU vende servizi digitali. Si potrebbe discutere che TTU vende contenuti digitali (documenti tradotti). Produrre nuovi contenuti di documenti però non è l’attività principale, che invece consiste nel trasformare i contenuti esistenti, palesemente una funzionalità servizio. I beni finanziari hanno un ruolo importante nei pagamenti dei servizi forniti da TTU, ma in modo marginale.

Per la dimensione ‘time scope’, TTU lavora in modo semi-dinamico e dinamico. Poiché i clienti devono registrarsi a TTU per usare i loro servizi, la relazione di e-business è semi-dinamica, ma è anche possibile usare TTU per attività individuali, creando caratteristiche dinamiche. Quando TTU fornisce servizi di interpretariato in real-time, possono anche apparire aspetti ultra-dinamici: le sessioni di interpretariato possono essere concluse (e instradate a un altro fornitore di servizi) prima di essere completate. Però, TTU non considera questo un aspetto base del business e quindi non adatta l’organizzazione a trattare tale aspetto.

#### *Direzioni (driver)*

TTU, come nuovo partecipante nel mercato della traduzione e interpretariato, ha una visione altamente basata sui principi dell’e-business in rete. Il driver principale di TTU è aumentare l’estensione: tramite i canali di e-business può raggiungere più potenziali clienti di un servizio tradizionale (offrendo sia la traduzione di documenti sia servizi di interpretariato ‘fisici’). TTU mira ad aumentare l’estensione geografica, temporale e modale. L’estensione modale mira più nettamente a fornire servizi on-line, real-time, in sessioni di video conferenza.

Vi sono due direzioni principali per TTU. Una direzione è il business on-time e on-line: la forza dell’approccio TTU sta nel fatto di poter essere contattata via internet quando i servizi realmente servono. La seconda è il business compresso

temporalmente: con le comunicazioni e la maggior parte del trasferimento documenti via canali elettronici, i tempi di throughput dei processi che richiedono traduzione o interpretariato vengono abbreviati in modo sostanziale.

# Capitolo 5

# Prospettiva organizzativa

## 5.1 Descrizione

Nel capitolo precedente, abbiamo discusso la prospettiva di Business (B) del framework BOAT. In questo capitolo, continuiamo la discussione passando alla prospettiva organizzativa - Organization (O) con lo scopo di fornire una visione strutturata degli aspetti organizzativi di cui tener conto nello sviluppo di sistemi informativi in rete. Per un approfondimento degli aspetti organizzativi in contesti più generali, si può far riferimento, per esempio, a [28, 22].

Di seguito, seguendo un approccio top-down, prima discutiamo le strutture a livello di reti di business fra organizzazioni (dette *strutture inter-organizzative*); poi a livello della singola organizzazione (dette *strutture intra-organizzative*). Successivamente, passiamo a trattare sistematicamente le funzioni di business in tali strutture e illustriamo come i processi di business possono essere posizionati nel contesto di tali strutture e funzioni organizzative, sia a livello di rete sia a livello di singola organizzazione.

## 5.2 Aspetti organizzativi inter-organizzazione

Raggiungere una buona comprensione delle strutture organizzative è importante per ogni dominio di business (si veda per esempio, [28]). Nell'e-business in rete questo è particolarmente vero: si ha a che fare con scenari complessi in cui è essenziale avere una chiara visione della struttura organizzativa. La natura dinamica di molti scenari di e-business aumenta la necessità di chiarezza: si deve capire che cosa è soggetto a cambiamenti e che cosa non lo è. In questa sezione, esaminiamo la struttura delle collaborazioni di e-business a livello di rete in modo incrementale e top-down, iniziando con una fotografia di alto livello che poi raffiniamo in due passi. Nella Sezione 5.3, continuiamo il raffinamento dentro i confini dell'organizzazione, ovvero a livello di *singola organizzazione*.

In Figura 5.1 sono illustrati i principali componenti degli aspetti organizzativi discussi in questa sezione:

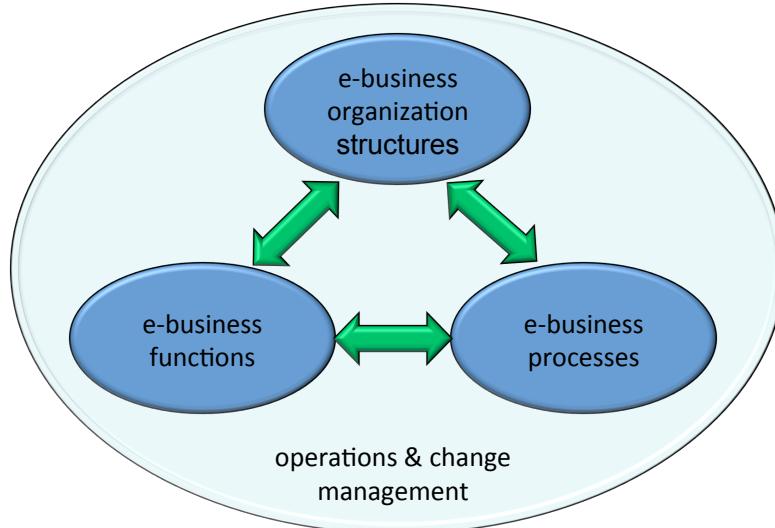


Figura 5.1: Ingredienti degli aspetti organizzativi

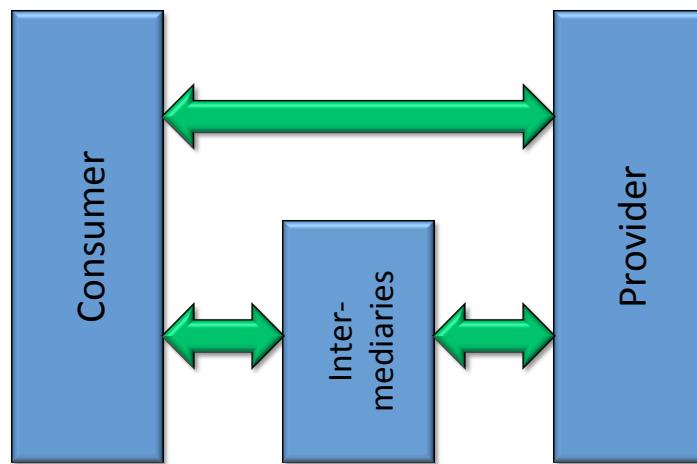
- Strutture dell'organizzazione e-business: vengono identificati i partecipanti (*parti*) che interagiscono con l'organizzazione.
- Funzioni di e-business: descrive le funzioni organizzative, derivate dal modello di Porter illustrato nell'introduzione.
- Processi di e-business: identifica e modella i processi, come sequenze possibili di interazioni tra le parti.

### 5.2.1 Parti all'interno di uno scenario di e-business

Inizialmente, ci concentriamo su scenari di e-business inter-organizzativi in cui due o più parti autonome collaborano in un mercato di e-business. Pertanto, la struttura più semplice di uno scenario è il mercato visto come una scatola nera, come mostrato in Figura 5.2. Naturalmente, questa struttura organizzativa non fornisce molta informazione ed è quindi etichettata come *livello 0*. Diventerà più interessante quando la si apre, raffinandola o espandendola.

Quando si raffina un diagramma a livello mercato, si identificano tre ruoli per le parti che vi collaborano:

- *consumatore*: una parte che richiede un oggetto di e-business (un prodotto o servizio, come discusso nella Prospettiva di Business nel Capitolo precedente).
- *fornitore*: una parte che offre un oggetto di e-business (un prodotto o servizio).



- *intermediari*: una (o più) parte(i) con ruolo ausiliario nel trasferimento di oggetti di e-business dal fornitore al consumatore.

Mettendo questi tre ruoli delle parti in relazione fra loro in un diagramma che raffini quello di Figura 5.2, otteniamo la Figura 5.3 che mostra un'organizzazione a *livello 1*. Ognuna delle tre parti può essere di tipo diverso, come discusso nella Sezione 4.2, ovvero, Business (Azienda), Consumatore/Cittadino, e “Government” (Pubblica Amministrazione).

La struttura organizzativa di Figura 5.3 contiene la collaborazione fra un singolo consumatore e un singolo fornitore. In generale, come descritto nella Sezione 4.5, vi sono varie parti coinvolte nella catena del valore di uno scenario, con una catena per i fornitori, una per i fornitori dei fornitori e così via. Il risultato è una struttura a rete, simile a quella mostrata in Figura 5.4.

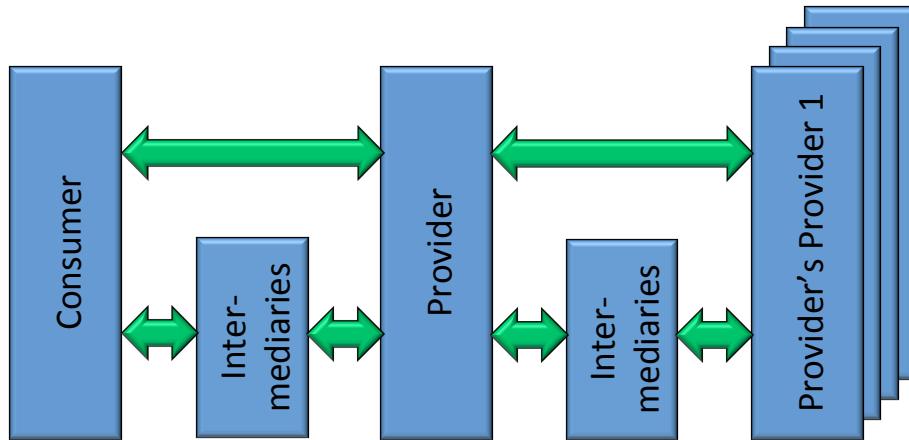


Figura 5.4: Struttura organizzativa, Livello 1 esteso

Le strutture sulla parte destra della figura sono simili a quelle sulla sinistra: c'è collaborazione fra un fornitore e un consumatore (a destra il fornitore è il consumatore del proprio fornitore). In altri termini, nella figura, ritroviamo ripetuto lo stesso pattern di struttura organizzativa. Ciò significa che, raggiungendo la comprensione del pattern di base, si possono comprendere vari tipi di strutture organizzative. Per questo, ci concentriamo sul pattern di base di Figura 5.3 per analizzare più in dettaglio le strutture organizzative.

Si noti che *consumatore* e *fornitore* sono ruoli – un'organizzazione può essere fornitore in una relazione di e-business, e al contempo consumatore in un'altra relazione (come mostrato in Figura 5.4). Si noti anche che il ruolo di intermediario è in qualche modo relativo a uno scenario di e-business. Per esempio, quando si analizza uno scenario di vendita on line (e-retailing), un'organizzazione che gestisce le carte di credito è un intermediario. In uno scenario di pagamenti, tuttavia, la stessa organizzazione agisce da fornitore. Tipicamente, un intermediario è caratterizzato dal fatto che non produce né consuma oggetti primari in uno scenario. La struttura in Figura 5.3 è chiara, ma contiene poca informazione relativa alla struttura per essere di supporto alle fasi di analisi o progetto dettagliato degli scenari. Perciò, nel seguito raffiniamo ulteriormente questa figura.

### 5.2.2 Raffinamento di intermediari e di canali

In Figura 5.3, non si fa distinzione fra diversi tipi di intermediari. Tuttavia, spesso uno scenario richiede vari tipi di intermediari, come per esempio:

- Un broker, che aiuta le parti a trovarsi ed identificarsi sul mercato (per esempio, con meccanismi simili alle pagine gialle elettroniche).

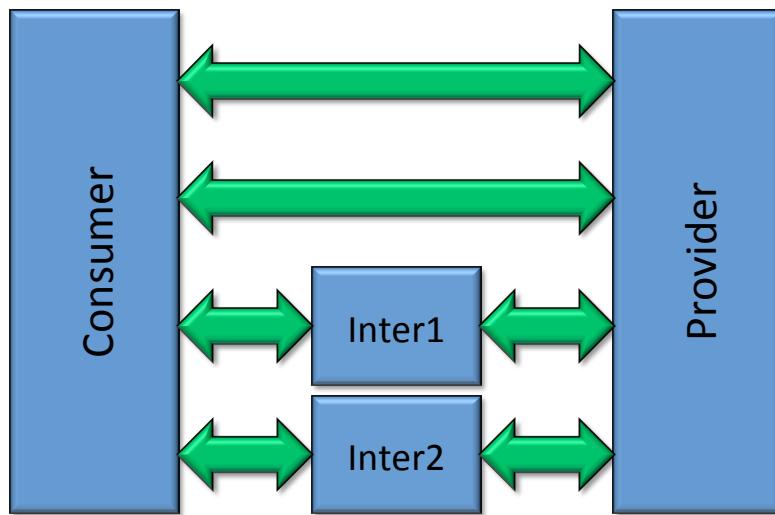


Figura 5.5: Struttura organizzativa, Livello 2

- Un intermediario finanziario, che gestisce l'esecuzione dei pagamenti fra parti.
- Un intermediario di trasporto, che gestisce il trasporto di oggetti fisici fra parti.

L'insieme degli intermediari necessari dipende dalla classificazione dello scenario: se esso non comprende scambio di oggetti (Sezione 4.5), può non servire un intermediario di trasporto. In uno scenario di puro scambio, non si rende necessario un intermediario finanziario. Altri scenari, tuttavia, possono richiedere altri intermediari, per esempio per analizzare la reputazione delle parti o per stipulare formalmente contratti elettronici fra le parti. In Figura 5.5 è mostrata la situazione multi-intermediario in una struttura organizzativa di *livello 2*. Per semplicità, appaiono solo due intermediari, etichettati come *Inter1* e *Inter2*; in pratica il numero di intermediari è arbitrario.

La seconda estensione di Figura 5.5 consiste nell'identificazione di canali comunicativi multipli fra consumatore e fornitore lungo cui fluisce l'informazione o gli oggetti di e-business. Canali multipli possono ciascuno servire una funzione di business diversa (come si spiega in seguito) oppure, servire la stessa funzione di business in vari modi (per esempio, per mezzo di vari canali comunicativi, come telefono o internet). Il secondo caso è detto multi-canale (strettamente correlato al progetto multi-canale discusso in Sezione 4.3). La figura mostra due canali per semplicità; il numero può essere arbitrario.

### 5.3 Relazioni organizzative intra-strutture

Nella sezione precedente, abbiamo raffinato un modello organizzativo di un e-business a livello mercato, però mantenendo opaca la struttura organizzativa interna delle parti che collaborano – sono rimaste scatole nere. In questa sezione, raffiniamo ulteriormente la struttura organizzativa aprendone la struttura interna ed esplorando le relazioni organizzative intra-strutture.

#### 5.3.1 Front end e back end

In Figura 5.5 le funzioni di business sia del consumatore che del fornitore sono mostrate come scatole nere, senza dettagli interni. Per progettare o analizzare in dettaglio una struttura organizzativa, apriamo questa scatola nera. Cominciamo spiegando che cosa si trova dentro la scatola nera.

L’interazione fra parti in un e-business in rete cambia di frequente a seguito del cambio sia dei business model sia delle tecnologie – ricordiamo che operano forze congiunte di *pull* dei requisiti e di *push* tecnologico come discusso nel Capitolo 1. Invece, le attività centrali delle organizzazioni tipicamente non cambiano così di frequente. Prendiamo per esempio una azienda telefonica che vende servizi su internet. L’interazione con i clienti, per esempio sotto forma di pacchetti di abbonamento offerti, cambia di continuo. Invece, le attività centrali (core), che riguardano l’infrastruttura per le chiamate e la relativa fatturazione, cambiano molto lentamente. Il fatto del diverso ritmo di cambiamento fra funzioni di business orientate internamente ed esternamente implica che dobbiamo disaccoppiarle. Senza un disaccoppiamento chiaro, non possiamo cambiarne una senza impattare sull’altra. In altre parole, ci serve realizzare una separazione dei compiti a livello di struttura intra-organizzativa. Il disaccoppiamento si vede in Figura 5.6 (la struttura organizzativa a *livello 3*). La funzionalità di business centrale che ha uno scopo intra-organizzativo (cioè, non esposto al mondo esterno) è detta *back end*. La funzionalità di business in contatto con le parti esterne (quindi con scopo inter-organizzativo) è detta *front end*. Nelle organizzazioni di tipo amministrativo, entrambi i tipi di funzionalità sono spesso dette rispettivamente *back office* e *front office*.

Disaccoppiare front end e back end significa che in situazioni reali va definito molto chiaramente che cosa fa parte del back end, che cosa del front end e come le due parti interoperano.

#### 5.3.2 Funzionalità front end e canali

Il passo finale di raffinamento da fare a questo punto è l’identificazione delle singole funzionalità nel front end. Una parte (con ruolo di consumatore o fornitore) ha varie funzionalità di business distinte che le servono per collaborare con le parti esterne, come pubblicizzare gli oggetti che offre su un certo mercato, negoziare i prezzi o le condizioni di consegna, acquistare oggetti e pagare gli oggetti acquistati (e in genere poche altre funzionalità). Poiché la flessibilità all’interno delle funzionalità è importantissima per seguire o iniziare gli sviluppi

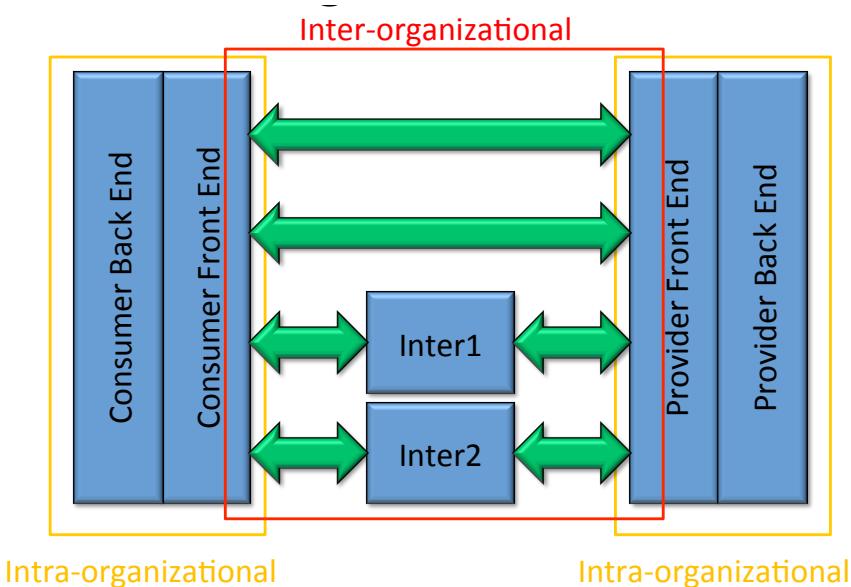


Figura 5.6: Struttura organizzativa, Livello 3

di mercato, è utile allocare le funzionalità a blocchi organizzativi diversi. Questi blocchi di base sono chiamati *unità organizzative* o moduli organizzativi. Ciò è mostrato in modo astratto in Figura 5.7, una elaborazione della Figura 5.6, che risulta in una struttura organizzativa a *livello 4*. I moduli  $CF_x$  rappresentano i moduli front end dei consumatori; i moduli  $PF_x$  rappresentano i moduli front end del fornitore. Il numero di moduli front end dipende dallo scenario.

A scopi illustrativi, la figura astratta di Figura 5.7 viene resa più concreta in Figura 5.8 riempiendo i moduli front end e gli intermediari con un esempio. Vedremo ulteriori esempi quando si discuteranno i casi alla fine di questo capitolo.

È anche possibile modularizzare le funzionalità back end. Non abbiamo mostrato la modularizzazione qui, poiché non è specifica per l'e-business negli aspetti O di BOAT: i business tradizionali richiedono una organizzazione modulare anche del back end. Quando ci si sposta a livello di architecture (A) di BOAT (si veda il prossimo capitolo), questo aspetto diventa più interessante.

In caso di e-business di tipo B2C, la parte Consumer in genere ha funzionalità molto limitate. Se è così, la distinzione tra funzionalità di front end e di back end non è molto utile. Alcuni moduli funzionali (per esempio, solo acquisto e pagamento, come in Figura 5.9 con Consumer a sinistra) possono essere sufficienti – o perfino un singolo modulo che rappresenti tutte le attività del consumatore.

Se un'organizzazione usa canali multipli per la stessa funzione di business (multi-channeling), è conveniente avere un modulo organizzativo per ogni canale. Questo aumenta ulteriormente il livello di flessibilità dell'organizzazione.

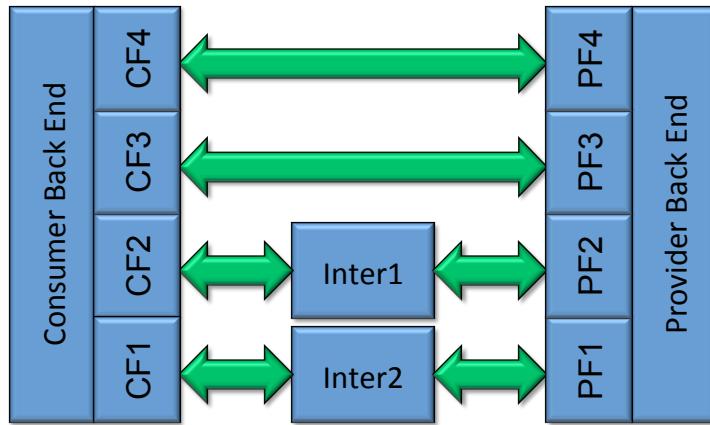


Figura 5.7: Struttura organizzativa, Livello 4

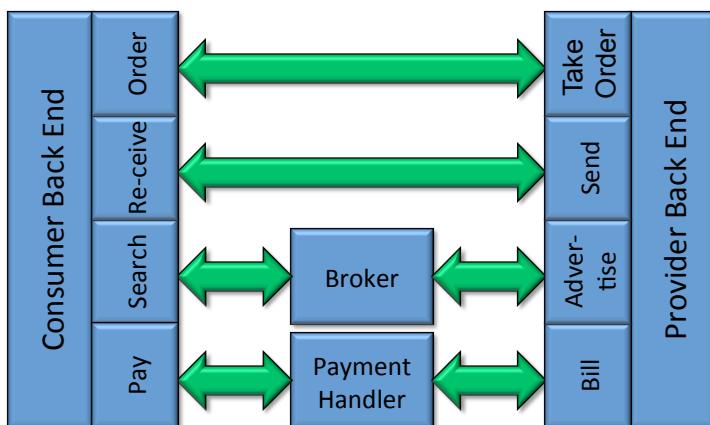


Figura 5.8: Struttura organizzativa, Livello 4 - esempio B2B

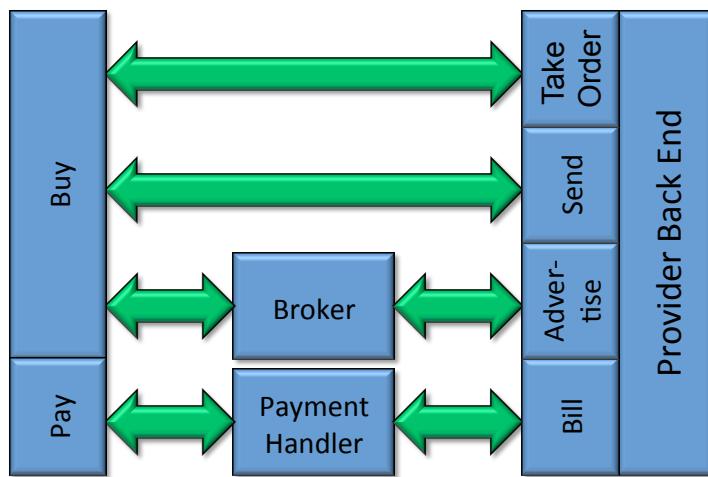


Figura 5.9: Struttura organizzativa, Livello 4 - esempio B2C

(ovviamente a spese di un numero più elevato di moduli e quindi di una maggiore complessità). Se l'organizzazione fornitrice di Figura 5.9 accetta ordini via Web (W) e via email (E), la sua struttura organizzativa può riflettere questo fatto, come mostrato in Figura 5.10. Si noti che questa struttura richiede la sincronizzazione fra moduli che supportano la stessa funzione di business per mantenere congruenti i processi di business, cosa che aggiunge complessità. Questo è il caso in cui bisogna scegliere esplicitamente in merito al trade-off tra flessibilità e complessità.

### 5.3.3 Il concetto di mid office

In alcune organizzazioni, gli approcci alla progettazione combinano i moduli di front end e di back end nel *mid office*. Questa funzione è un “sistema avanzato di alternanza” fra moduli back end e moduli front end, che permette di collegare le funzioni di business che vengono eseguite. Lo scopo del mid office è quello di incrementare la flessibilità. Le sue funzionalità comprendono, per esempio, l'integrazione flessibile di prodotti base forniti dal back end dell'organizzazione in prodotti composti attraverso vari canali situati nel front end. Un esempio nel dominio finanziario è il seguente: i prodotti bancari di base (conti correnti o prestiti) amministrati nel back end possono essere combinati in modo flessibile nel mid office in pacchetti finanziari offerti ai consumatori tramite il front end. La struttura organizzativa risultante è mostrata in Figura 5.11 (basata su quella di Figura 5.7). Qui abbiamo dettagliato i vari back end del consumatore (moduli *CBx*) e del fornitrice (moduli *PBx*) per mostrare le funzioni di integrazione svolte dal mid office.

Per brevità, non useremo il concetto di mid office nel volume. Se necessario, le discussioni che seguono possono essere facilmente estese a includere questo

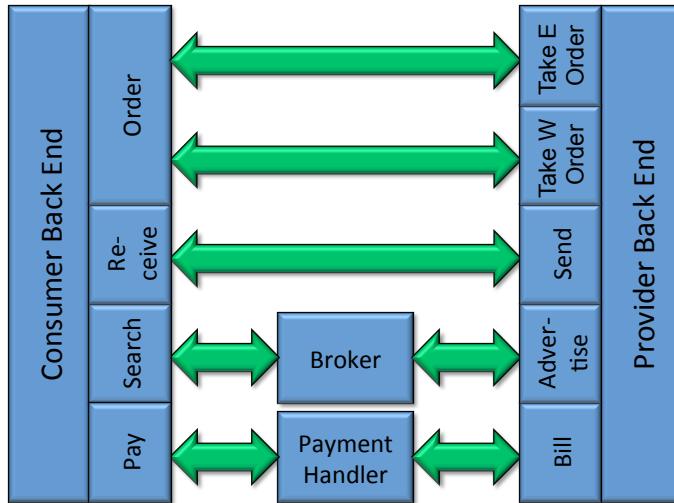


Figura 5.10: Struttura organizzativa, Livello 4 - esempio multicanale

concetto.

### 5.3.4 Dalla struttura organizzativa ai processi inter-organizzazione

La struttura organizzativa di livello 4 è caratterizzata dalla rappresentazione delle interazioni tra le parti e dalle funzionalità fornite da fornitori e consumatori. Come già discusso nell'introduzione, nella progettazione siamo però interessati non solo a descrivere la struttura delle interazioni (network), ma anche i processi, e cioè le possibili sequenze di interazione.

I *processi inter-organizzazione*, detti anche processi in rete [15], processi cross-organizzazione [20] o processi collaborativi [26], rappresentano le possibili sequenze di interazione tra le parti.

Nel seguito (si veda la Sezione 5.5) si parlerà anche della rappresentazione dei *processi intra-organizzazione*, ovvero di questi processi eseguiti completamente (si dice anche messi in opera) all'interno dei confini di una singola organizzazione e della relazione di quest'ultimi con i processi inter-organizzazione.

I processi inter-organizzazione sono eseguiti in collaborazione fra due o più parti – quindi sono interessanti dal punto di vista dell'e-business in rete e sono usati per realizzare la collaborazione nei modelli, come discusso nel capitolo precedente. Si rende un processo esplicitamente inter-organizzazione allocando i passi alle diversi parti di uno scenario – o, messo in termini semplici: si indica chi fa che cosa.

Riprendendo il semplice processo di acquisto on line (e-buying) descritto nella struttura organizzativa di Figura 5.8, possiamo indicare l'ordine previsto

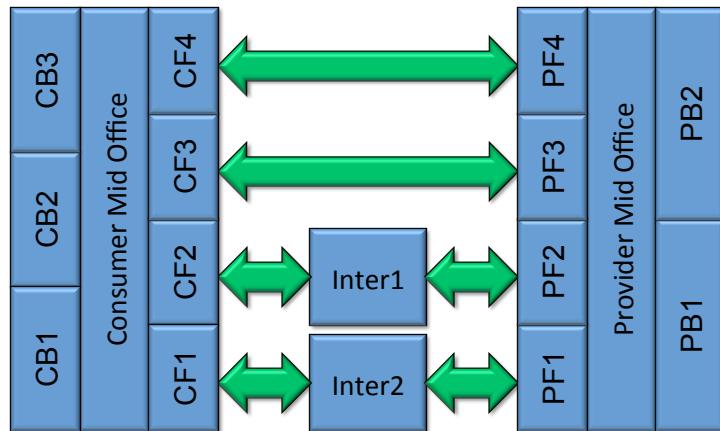


Figura 5.11: Struttura organizzativa, Livello 4 - esempio con mid office

per i flussi informativi tra le parti aggiungendo la direzione delle interazioni ottenendo un diagramma ibrido organizzazione/processo come quello in Figura 5.12. Questo diagramma offre una visione di alto livello di come un processo progredisce attraverso le funzionalità delle organizzazioni coinvolte. Il consumatore invia un ordine tramite la funzione Order al fornitore, che la riceve con Take Order. Successivamente, il fornitore invia le merci, ricevute con Receive dal consumatore, e il conto, inviato con la funzione Bill e ricevuto e pagato tramite la funzione Pay, tramite l'interazione con il Payment Handler che funge da intermediario.

Si noti come nella Figura 5.12 siano coinvolte nel processo solo sei delle funzioni indicate nel diagramma. Infatti l'interazione tra fornitore e consumatore tramite un broker viene qui considerata un processo separato da quello della singola vendita, rappresentata nel diagramma, in quanto si svolgerà in tempi e modi diversi.

#### Tipi di flussi di controllo inter-organizzazione

Esistono tre tipi di processi inter-organizzazione a seconda della distribuzione del flusso di controllo all'interno del processo, ovvero dell'allocazione delle responsabilità nel flusso.

Il caso più semplice è quello di flusso di controllo unilaterale, in cui una parte ha il controllo completo dell'intero processo, tipico del caso in cui il fornitore gestisce il processo e il consumatore non ha un proprio back end. Questo significa che il fornitore prende tutte le decisioni di instradamento e scheduling delle attività del processo. Le altre parti in collaborazione partecipano al processo solo eseguendo compiti specifici su richiesta del controllore. Questi compiti individuali possono essere eseguiti per esempio usando interfacce basate su web verso la parte controllore.

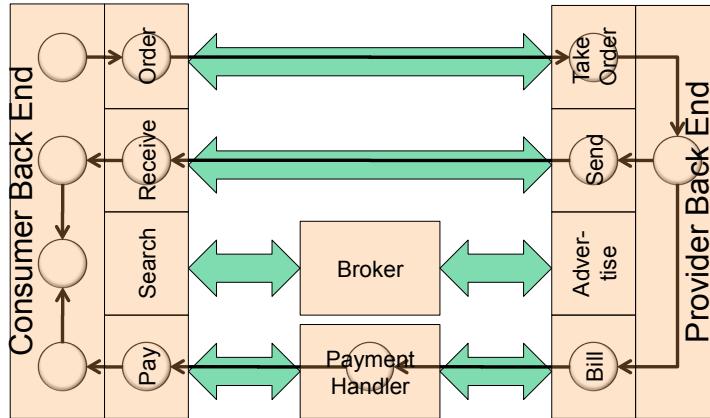


Figura 5.12: Processo in rete mappato sulla struttura organizzativa

Più complesso è il caso di flusso di controllo bilaterale: le due parti controllano il flusso in modo collaborativo. Il processo è diviso in due porzioni, di cui ciascuna parte è responsabile. Quindi le due parti devono sincronizzare gli stati delle loro porzioni del processo e scambiano dati di processo per mantenere l'intero processo in esecuzione corretta.

Il caso del flusso di controllo multilaterale è una generalizzazione del caso di flusso di controllo bilaterale. Qui la situazione è simile, ma le parti coinvolte nel prendere il controllo del processo sono molte. Questo tipo di controllo può avvenire in scenari di aziende virtuali, in cui tante organizzazioni eseguono un processo complesso in modo collaborativo [17, 27]. Un esempio astratto di flusso di controllo multilaterale è mostrato in Figura 5.13. Qui, cinque organizzazioni (indicate da ellissi) eseguono insieme un processo inter-organizzazione con attività (indicate da piccoli cerchi) eseguite da ciascuna organizzazione. Il processo inizia presso l'organizzazione A e termina presso l'organizzazione E; contiene sia collegamenti di flusso di controllo intra-organizzazione (archi pieni dentro le ellissi) sia collegamenti di flusso di controllo inter-organizzazione (archi tratteggiati fra le ellissi). Le cinque organizzazioni insieme sono responsabili del flusso di controllo dell'intero processo, ovvero il flusso di controllo è distribuito tra loro.

Il flusso di controllo multilaterale rende la sincronizzazione più complessa. Quando la topologia della collaborazione contiene cicli (ovvero quando le organizzazioni sono ‘connesse’ tramite più di un cammino) le parti potrebbero perfino eseguire attività incongruenti se i processi non sono ben progettati.

Nelle classi del flusso di controllo bilaterale e multilaterale, le organizzazioni partecipanti devono esporre i dettagli dei loro processi alle altre organizzazioni per sincronizzare i loro processi (a livello di *vista pubblica* del processo), ma non devono necessariamente esporre tutti i dettagli dei processi interni (ovvero di *vista privata* del processo). Infatti, i processi possono contenere dettagli confi-

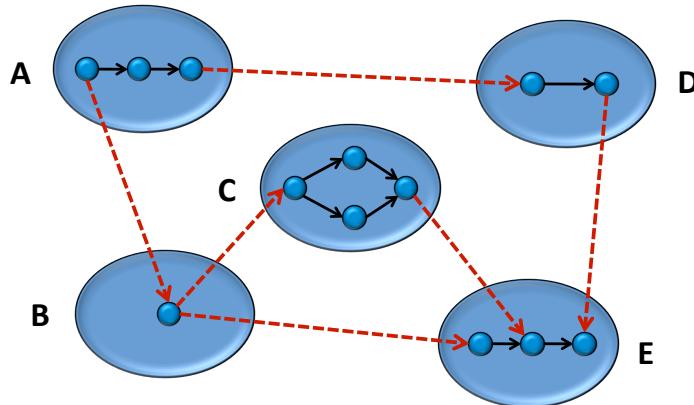


Figura 5.13: Interazione tra più organizzazioni

denziali (per esempio, per ragioni di competitività) o semplicemente irrilevanti per le altre parti. Quindi, le organizzazioni in genere espongono astrazioni dei loro processi interni in termini di processi esterni a livello inter-organizzazione [16].

## 5.4 Modellazione dei processi

### 5.4.1 Definizioni

Come abbiamo visto, le strutture organizzative descrivono l'organizzazione delle funzioni di business delle organizzazioni che partecipano allo scenario, ma non descrivono come le varie funzioni di business vengono attivate nel tempo durante la collaborazione. È evidente da quanto sopra illustrato che è necessario modellare i processi in modo più preciso di quanto illustrato finora, per mettere in evidenza la struttura delle interazioni, le sequenze dei messaggi e le principali alternative. Per rappresentare questi aspetti, dobbiamo introdurre la nozione di processo aziendale (business process o processo di business) e della sua modellazione.

In questa sezione, prima descriviamo i principi base e la natura del concetto di processo di business. Poi mostriamo come specificare i processi di business introducendo la notazione BPMN (Business Process Modeling Notation) nella Sezione 5.4.2. Visto che trattiamo di collaborazione fra organizzazioni, i processi in questo contesto si estendono su tante organizzazioni. Discutiamo questo aspetto nella prossima sezione; successivamente, descriviamo i processi nell'e-business. Concludiamo questa sezione discutendo le differenze fra processi statici e dinamici. Il concetto di processo dinamico permette di pensare ai processi come intrecciati su varie organizzazioni che stabiliscono le collaborazioni ‘al volo’ mentre fanno business.

Un *processo di business* (d'ora in poi semplicemente *processo*) è un insieme di *attività* eseguite secondo un ordine specifico per ottenere un certo *obiettivo di business*. Un processo può affrontare obiettivi primari, come vendere un prodotto, oppure anche obiettivi secondari, come realizzare una analisi finanziaria mensile di un'azienda, attività che costituisce un processo secondario.

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 come visto in Sezione 5.3. Nella descrizione di un processo, 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*. Le attività di un processo sono parte delle funzioni di businesss, come discusso in Sezione 5.3. Gli attori che eseguono attività spesso sono esseri umani, ma possono anche essere macchine, le quali, nel contesto e-business, sono tipicamente sistemi informativi automatizzati. Si pensi ad esempio alla direzione (driver) volta a raggiungere processi completamente automatizzati, discussa in Sezione 4.4.

Un processo viene specificato secondo un *modello di processo* che specifica l'ordine dei passi. I passi possono essere attività elementari o sottoprocessi. Le attività elementari sono unità atomiche che non vanno ulteriormente raffinate nella specifica di processo. I sottoprocessi sono processi a un livello più basso di aggregazione, usati per spezzare la complessità di un modello o riusare parti di un modello per altri obiettivi di business diversi.

L'ordinamento dei passi in un modello può essere semplice (per esempio, una sequenza) o complesso (contenente cammini alternativi, cammini paralleli e iterazioni). L'ordinamento dei passi nel tempo è detto *flusso di controllo*. Quando si parla di processi, ci si focalizza sul flusso di controllo e poco sui dettagli interni dei singoli passi.

Un modello di processo specifica la struttura di un certo tipo di processo, ovvero tutte le possibili esecuzioni del processo. Per esempio, il modello di un processo di vendite specifica tutti i modi in cui devono essere eseguite le vendite. Una singola esecuzione di un processo secondo un certo modello è detto *istanza del modello*, o *caso del processo*. Eseguire una vendita specifica, quindi, è una istanza del processo vendite.

#### 5.4.2 Modellazione di processo in BPMN

Per facilitare l'interazione tra diverse organizzazioni o anche la gestione interna dei processi, questi, una volta identificati, devono essere modellati in modo da definire come una determinata attività deve svolgersi. 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 (sia clienti che fornitori).

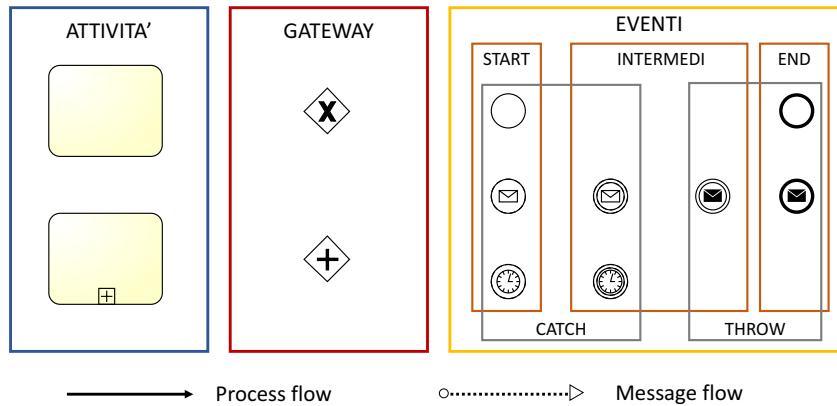
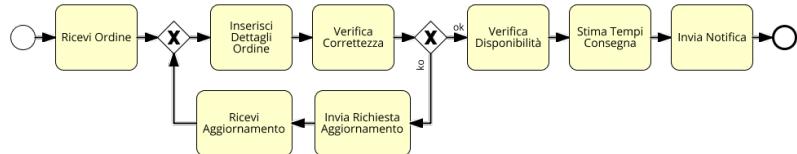


Figura 5.14: Gli elementi principali della notazione BPMN divisi nella categorie di Attività, Gateway ed Eventi.

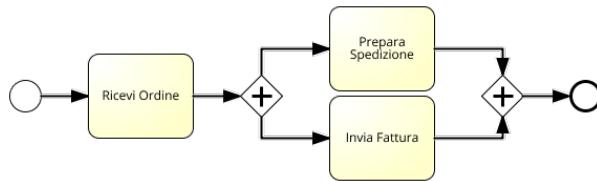
Una delle più diffuse notazioni per modellare i processi è la Business Process Modeling Notation (BPMN). Gli elementi principali che costituiscono un processo sono:

- *Attività*: le unità di lavoro del processo. Una attività ha un sotto-obiettivo e una durata non nulla e contribuisce all’obiettivo del processo intero. Nella notazione BPMN un’attività 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 (in base ad una condizione viene svolto un insieme di attività invece che un altro), o biforcazioni nel caso in cui più attività possono essere eseguite contemporaneamente. Nella notazione BPMN un gateway è rappresentato con un rombo al cui interno è disegnato un simbolo che ne definisce la tipologia.
- *Eventi*: gli 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. Nella notazione BPMN gli eventi sono modellati con un cerchio al cui interno è disegnato un simbolo che ne definisce la tipologia.

Uno specchietto degli elementi base di BPMN raggruppati per categoria è mostrato in Figura 5.14. Gli elementi di un processo sono collegati tra loro tramite il *Process flow*. Le attività possono essere semplici oppure possono essere



(a) Processo di ordine che usa il gateway XOR



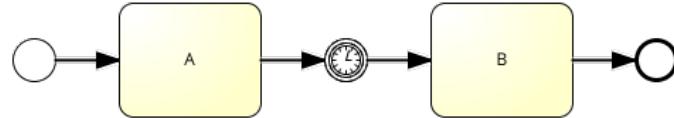
(b) Processo di ordine che usa il gateway AND

Figura 5.15: Esempio di utilizzo dei gateway XOR e AND

un raggruppamento di tante attività che contribuiscono a realizzare un sottobiettivo del processo. In questo caso, si parla di *sotto-processi*, rappresentati come un’attività con un “+” in basso al centro, come mostrato in figura.

Esistono anche diversi tipi di gateway. Nella Figura 5.14 sono mostrati rispettivamente un gateway XOR (rappresentato con una “X”) e un gateway AND (identificato da un “+”). Il gateway XOR permette di scegliere uno solo tra due flussi alternativi, in base ad una condizione sui dati del processo. In alternativa il gateway può essere usato anche per ricongiungere più flussi all’interno di un processo (appena un’attività su uno dei flussi entranti nel gateway è completata, il processo procede). Un esempio di processo che usa il gateway XOR è mostrato in Figura 5.15a, in cui è modellato un processo di gestione di un ordine che, se i dati inseriti non sono corretti, ne richiede l’aggiornamento. Il gateway AND invece 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. Quando usato per ricongiungere più rami, il suo comportamento è quello di sincronizzare rami del processo aspettando che tutte le attività che lo precedono sui rami in ingresso siano completate per proseguire. Nell’esempio di Figura 5.15b, viene mostrato un processo in cui, ricevuto un ordine da un cliente, la fattura e la spedizione sono preparate contemporaneamente, e il processo termina solo quando entrambe le attività sono completate.

Infine, gli eventi possono essere raggruppati secondo diverse categorie. Gli eventi possono trovarsi all’inizio, alla fine, o all’interno del flusso di un processo. In base alla loro posizione si classificano in *eventi start* (rappresentati con un cerchio singolo sono le condizioni che fanno iniziare il processo), *eventi end* (rappresentati con un cerchio singolo ma spesso sono le condizioni che fanno terminare un processo) ed *eventi intermedi* (rappresentati con un doppio cer-



(a) Processo con evento timer intermedio



(b) Processo con evento messaggio throw intermedio

Figura 5.16: Esempio di utilizzo di eventi in un processo

chio. Ogni processo inizia e termina sempre con un evento. All'interno di un evento, è possibile trovare un simbolo che ne identifica il tipo. Tra gli eventi più usati troviamo ad esempio il *timer* che indica un qualsiasi evento di natura temporale (ogni lunedì, dopo due ore, scaduto il tempo previsto, ecc.). Un altro evento molto usato è l'*evento messaggio*, rappresentato con una busta, che invece rappresenta uno scambio di informazioni tra l'interno e l'esterno dell'organizzazione, quindi con i partner. Gli eventi possono anche essere suddivisi in base al fatto che siano prodotti dal processo o che siano invece catturati dal processo ma prodotti all'esterno dello stesso. Nel primo caso si parla di *eventi throw*, nel secondo di *eventi catch*. I due tipi di eventi si distinguono nella notazione perché i primi sono contrassegnati da un'icona interna colorata di nero. Ad esempio l'invio di un messaggio è un evento throw rappresentato con una busta nera, mentre lo scadere di un timer è un evento catch. Per loro natura gli eventi start sono sempre catch, mentre gli eventi end sono sempre throw. Gli eventi intermedi possono essere sia catch che throw. In Figura 5.16a il processo inizia con un evento start generico e, terminata l'attività A attende il verificarsi di un evento timer per iniziare l'attività B e poi terminare con un end generico. In modo simile, in Figura 5.16b dopo l'attività A il processo invia un messaggio e poi prosegue.

BPMN permette inoltre di modellare l'interazione tra più organizzazioni, che avviene tramite lo scambio di messaggi. Ogni organizzazione ha il suo processo che dovrà coordinarsi tramite eventi o attività di tipo messaggio con il processo del partner. I messaggi scambiati sono indicati esplicitamente nel modello, con l'uso del *Message flow*. Il message flow può partire da un evento throw di un messaggio ed arrivare a un evento catch, oppure partire e/o arrivare ad elementi di tipo attività che generano o ricevono messaggi.

Un esempio è mostrato in Figura 5.17, dove sono modellati i processi di

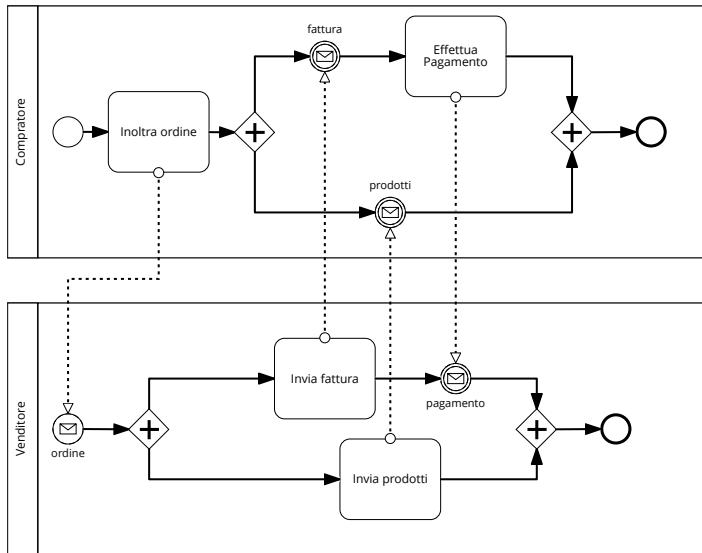


Figura 5.17: Esempio di processo venditore-acquirente

un compratore (in alto) e un venditore (in basso). Questa figura rappresenta una possibile descrizione dell’interazione tra organizzazioni descritta nel modello organizzativo di livello 4 di Figura 5.8 (per una prima discussione supponiamo un’interazione diretta senza intermediario per i pagamenti).

La stessa interazione, ma con un diverso processo per il rivenditore è mostrato in Figura 5.18. In questo caso il rivenditore applica come business rule l’invio dei prodotti solo dopo ricezione del pagamento. Come si vede, la relazione tra gli attori può non cambiare se il processo interno di uno dei due cambia. In altri casi però una modifica potrebbe avere effetti anche sul processo del partner cambiando il modo in cui le attività dei due attori devono coordinarsi.

È anche per questo motivo che la rappresentazione dei processi deve fornire anche una vista pubblica del processo, per consentire di verificare che le interazioni previste siano consistenti. Ad esempio, nel caso in esame, se il venditore aspettasse il pagamento prima di fornire la merce e il compratore aspettasse la merce prima di pagare si verificherebbe una classica situazione di blocco e il processo non potrebbe procedere.

Mostriamo infine la rappresentazione del processo corrispondente al diagramma organizzativo di livello 4 di Figura 5.8 con la partecipazione dell’intermediario Payment Handler nella Figura 5.19.

In questo paragrafo sono stati introdotti in breve i concetti principali di BPMN. Per una trattazione completa ed esempi è possibile riferirsi a [8] o a [41] per una trattazione più approfondita sulla modellazione dei processi. Per la realizzazione dei diagrammi è stato utilizzato Signavio, un tool online disponibile

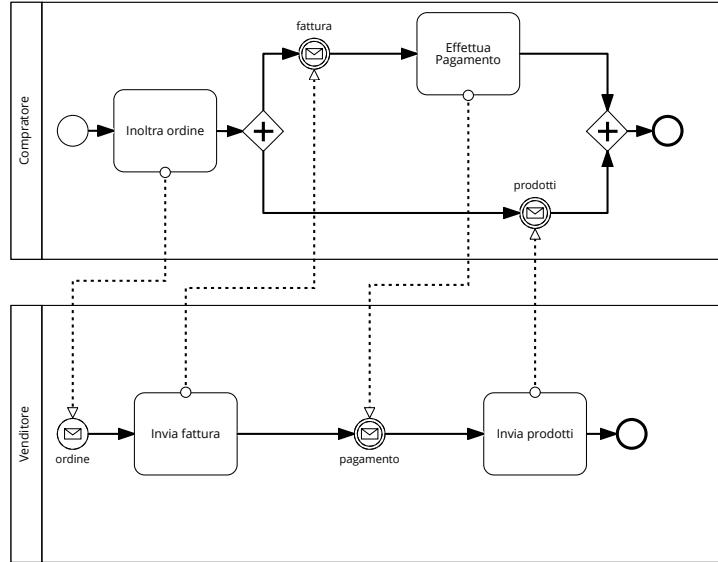


Figura 5.18: Esempio di processo venditore-acquirente modificato

con licenza accademica [34].

## 5.5 Identificazione dei processi

Nella Sezione 5.2, abbiamo discusso come i moduli organizzativi (unità) implementano le funzioni di business nelle organizzazioni. Questa implementazione consiste nella definizione in dettaglio dei processi aziendali, definiti come l'insieme delle attività attraverso cui le risorse di un'organizzazione possono realizzare gli obiettivi aziendali trasformando un input in un output, creando in questo modo valore. Un processo aziendale è quindi focalizzato su un output (un servizio o un prodotto) e descrive tutte le attività e i ruoli coinvolti per ottenerlo. Per questo motivo un processo può coinvolgere diverse unità all'interno di un'organizzazione e diversi ruoli, che collaborano tra loro al fine di realizzare un obiettivo comune. Il valore risultante dal processo dipende dal buon funzionamento delle singole attività e dalla qualità del loro coordinamento. Quindi, sebbene un processo sia finalizzato alla realizzazione di un obiettivo aziendale (funzionalità offerta dall'organizzazione agli attori esterni con cui interagisce), esso definisce tutte le attività che portano a tale risultato. Queste attività quindi comprendono anche tutte le funzioni di back-end che contribuiscono alla realizzazione dell'obiettivo e che non sono direttamente esposte agli attori esterni.

Identificare e descrivere i processi di un'organizzazione è un'attività molto complessa e costosa. Innanzitutto i processi possono essere descritti a diversi livelli di dettaglio. Ogni processo può infatti essere scomposto gerarchicamente

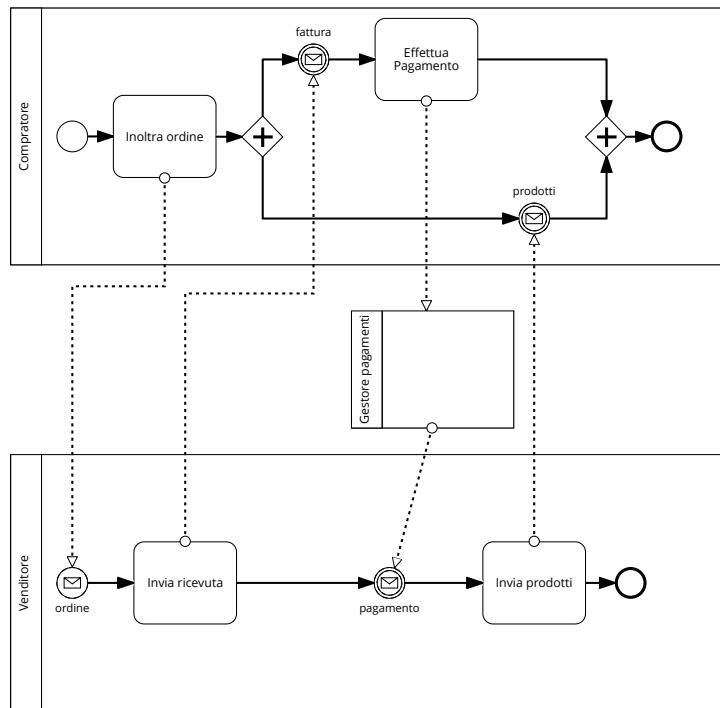


Figura 5.19: Esempio di processo venditore-acquirente con intermediario

in fasi, attività e operazioni. In base alle proprie necessità, un’organizzazione può stabilire il livello di dettaglio per la descrizione dei propri processi. Solitamente, le piccole aziende non hanno interesse nella descrizione dei processi a un maggiore livello di dettaglio (le operazioni), in quanto tale attività di analisi potrebbe risultare eccessivamente costosa in rapporto ai benefici ottenuti. Stabilito il livello di dettaglio desiderato, è necessario identificare quali sono i processi all’interno di un’organizzazione, passo necessario per procedere poi alla loro modellazione.

Nel seguito discutiamo alcuni approcci che è possibile utilizzare per l’identificazione dei processi.

### 5.5.1 Check-list settoriali

Un metodo molto semplice per supportare l’identificazione dei processi è quello che si basa sulla somiglianza di organizzazioni che operano nello stesso settore. Il metodo 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. Questo approccio viene chiamato check-list proprio perché consiste nella selezione dei processi da considerare da una lista di riferimento. Un esempio

di framework che fornisce una classificazione dei processi aziendali è il Process Classification Framework fornito dall'American Productivity & Quality Center (APQC)<sup>1</sup>.

Questo approccio, sebbene molto semplice, ha comunque varie limitazioni. Mentre il metodo risulta efficace per settori altamente regolamentati dove i processi delle diverse organizzazioni sono effettivamente molto simili, esso è poco applicabile quando il settore è poco regolamentato. In questo caso, infatti, la sovrapposizione completa tra la check-list e i processi aziendali non è garantita. Un'altra limitazione di questo metodo è dovuta alla necessità di effettuare un'analisi approfondita dei processi aziendali prima di applicare il metodo, in modo da poter effettuare il confronto. Questa operazione è ovviamente dispendiosa e complessa.

### 5.5.2 Catena del valore di Porter

Nel capitolo introduttivo abbiamo visto come i processi possono essere classificati facendo riferimento alla catena del valore di Porter. Tale modello suddivide i processi in processi primari e processi di supporto e definisce cinque categorie di processi primari e quattro categorie di processi di supporto. Tale classificazione può essere utilizzata anche come punto di partenza per l'identificazione dei processi di un'organizzazione.

Con riferimento alla definizione degli aspetti organizzativi e al modello delle funzioni di business secondo l'approccio BOAT, i processi identificati dal modello di Porter possono essere messi in relazione con gli elementi del modello degli aspetti organizzativi BOAT, distinguendo tra attività di front end e attività di back end. Un esempio di funzioni di front end ottenute dal modello di Porter come base per l'identificazione strutturale delle funzioni di front end dell'organizzazione è mostrato in Figura 5.20. Nella colonna di sinistra della tabella, troviamo le funzioni di business identificate, nella colonna centrale, abbiamo dettagliato e specializzato queste funzioni.

In modo simile è possibile procedere per identificare i processi di back end, come mostrato in Figura 5.21.

Anche questo approccio presenta delle limitazioni. Innanzitutto, l'approccio di Porter si focalizza su attività suddivise in base alle unità organizzative (marketing, logistica, produzione, ecc.), perdendo quindi la visione interfunzionale e collaborativa tipica dei processi. Inoltre, le attività identificate da Porter tramite le nove categorie sono un numero limitato delle attività dell'organizzazione analizzate ad un livello di aggregazione dei processi molto elevato. A partire da questo modello, è quindi difficile, definire i processi ad un livello di dettaglio superiore, qualora questa analisi sia di interesse per l'organizzazione. Infine, il modello non è un modello generale che possa essere di riferimento per qualunque tipo di organizzazione. Il modello è infatti stato pensato per organizzazioni produttive ‘tradizionali’, dove la produzione fisica e il trasporto sono le attività più importanti. Pertanto, non è immediato applicare efficacemente il modello

---

<sup>1</sup><https://www.apqc.org/pcf>

| Porter function      | Networked e-business function    | Comment                   |
|----------------------|----------------------------------|---------------------------|
| <i>provider side</i> |                                  |                           |
| Marketing & Sales    | advertising                      | usually via intermediary  |
|                      | negotiating                      |                           |
|                      | contracting                      | possibly via intermediary |
|                      | selling                          |                           |
|                      | billing                          |                           |
| Outbound Logistics   | sending goods                    | possibly via intermediary |
|                      | status information provisioning  |                           |
| Service              | goods information provisioning   |                           |
|                      | update provisioning              | in case of digital goods  |
|                      | value added service provisioning |                           |
| <i>consumer side</i> |                                  |                           |
| Procurement          | searching                        | usually via intermediary  |
|                      | negotiating                      |                           |
|                      | contracting                      | possibly via intermediary |
|                      | buying                           |                           |
|                      | paying                           | usually via intermediary  |
| Inbound Logistics    | receiving goods                  | possibly via intermediary |

Figura 5.20: Esempio di catalogo di funzioni di front end

ad organizzazioni orientate ai servizi o alle piccole e medie imprese in cui alcune delle attività identificate da Porter potrebbero essere del tutto assenti.

### 5.5.3 Approccio Analitico

L'approccio analitico all'identificazione dei processi 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 (fornitori, clienti, intermediari) e ricostruisce il processo a ritroso identificando prima l'output a loro fornito dall'organizzazione, quindi le attività che portano a generare questo output, 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. Ad esempio è possibile partire dalla definizione delle interfacce verso l'esterno che la nostra organizzazione espone e ricostruire i processi che permettono di realizzare tali funzionalità.

Questo procedimento permette di eseguire l'identificazione dei processi primari, quelli cioè che portano alla generazione di un output in termini di prodotti o servizi. Per descrivere completamente l'organizzazione è quindi necessario completare questo procedimento con l'identificazione dei processi di supporto,

| Porter function           | Networked e-business function | Comment                 |
|---------------------------|-------------------------------|-------------------------|
| Operations                | production                    | may be digital objects  |
|                           | stock keeping                 | n.a. if digital objects |
| Procurement               | procurement                   |                         |
| Technology Development    | product configuration         |                         |
|                           | catalog management            |                         |
| Human Resource Management | personnel selection           |                         |
|                           | knowledge innovation          |                         |
| Firm Infrastructure       | financial administration      | linked with e-payments  |
|                           | IS management                 |                         |

Figura 5.21: Funzioni di back end

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. Alcuni tool di BPM, partendo dalla rappresentazione di un processo secondo una notazione standard, possono tradurla in procedure software in modo automatico o semi-automatico.

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

## 5.6 Operazionalizzazione dei business driver

Nel Capitolo 4, abbiamo discusso l'aspetto business (B), considerando le classi di ingredienti che contribuiscono ai modelli di business. Come visto in questo capitolo, l'aspetto organizzazione (O) è una operazionalizzazione dell'aspetto B. Come tale, dovremmo essere in grado di identificare un chiaro mapping tra gli elementi dell'aspetto B e gli elementi dell'aspetto O in uno scenario specifico.

Nel seguito, discutiamo il mapping B verso O senza pretese di completezza, in quanto un mapping completo da B a O è molto complesso e dipendente dallo scenario. Invece, discutiamo con chiarezza poche relazioni importanti, così che il lettore sia in grado di analizzare le altre relazioni da solo.

Nella Sezione 4.3 abbiamo visto i concetti di reach, richness e efficienza come ingredienti base dei modelli: i miglioramenti relativi a questi concetti costituiscono i business driver. In questa sezione, analizziamo l'operazionalizzazione dei concetti di reach e richness.

Il concetto di reach fa un chiaro riferimento alla comunicazione fra organizzazioni in uno scenario. Pertanto, possiamo operazionalizzare il concetto in termini di canali di comunicazione fra parti che collaborano (come introdotto in Figura 5.7). La reach geografica, per esempio, può essere aumentata solo se vengono installati canali di comunicazione con le giuste caratteristiche fra le organizzazioni, ovvero canali che aiutano a colmare le distanze. La reach modale

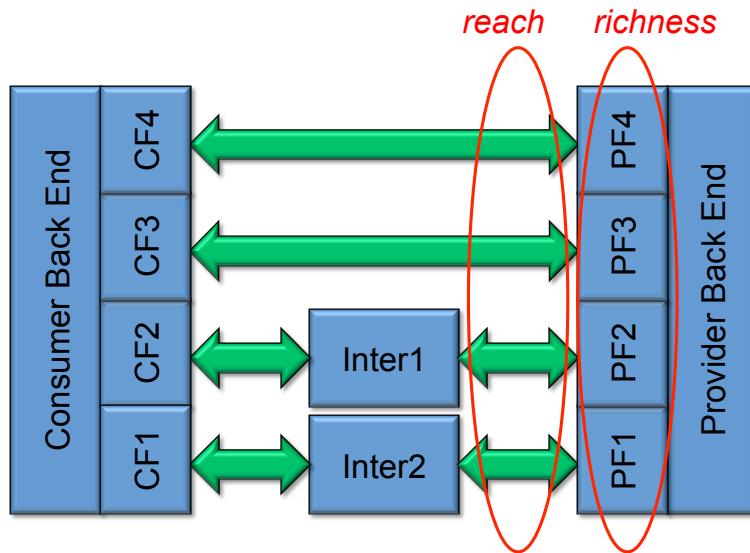


Figura 5.22: Reach e richness nella struttura organizzativa

può essere incrementata usando la multi-canalità, come discusso in precedenza in questo capitolo.

Il livello di richness che può essere offerto da un’organizzazione è pesantemente correlato alle funzionalità dei blocchi di base del front office organizzativo, che troviamo nella struttura organizzativa (si veda ancora la Figura 5.7): i blocchi del front office ‘implementano’ l’interazione coi partner. Per fornire richness migliore, le funzionalità di questi blocchi devono essere espanso, oppure vanno introdotti nuovi blocchi nella struttura organizzativa. Per esempio, nell’aspetto O, può essere inserito un nuovo blocco funzionale di pubblicità interattiva per aumentare la richness in B.

Ricapitoliamo le osservazioni esposte tramite la Figura 5.22, indicando dove sono situate la reach e la richness nella struttura organizzativa di Figura 5.7.

## 5.7 Esempi

Riprendiamo i casi di studio e discutiamone ora gli aspetti organizzativi. Per brevità, non elaboriamo le architetture di POSH e TTU in dettaglio (servirebbe un intero libro solo per questo). Invece, mostriamo alcuni modelli architetturali ai tre livelli di aggregazione per illustrare i concetti discussi in questo capitolo.

### 5.7.1 POSH

POSH vende due tipi di prodotti, gestiti in modi diversi. I piccoli oggetti di ufficio sono consegnati ai clienti dal magazzino interno tramite un partner logistico,

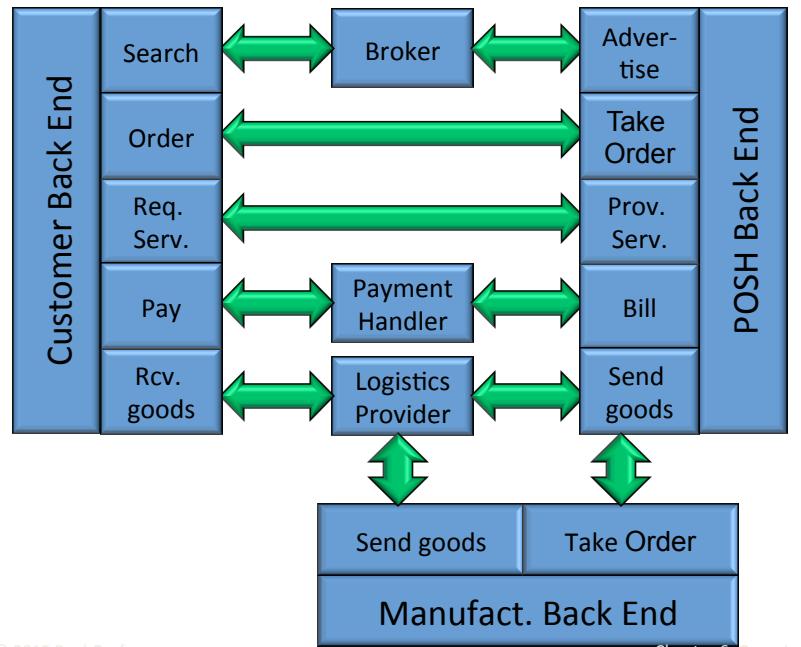


Figura 5.23: Esempio di struttura organizzativa di POSH

gli arredamenti di grandi dimensioni sono consegnate da (e, se necessario, prodotte da) un produttore di arredi con cui POSH coopera. Questo significa che lo scenario POSH è a tre parti: fornitore (POSH), consumatore e produttore.

Prima, elaboriamo la struttura organizzativa dello scenario POSH; poi esaminiamo il processo di vendita.

### Struttura organizzativa

La struttura organizzativa dello scenario POSH è mostrata in Figura 5.23. A destra in alto vediamo la struttura dello stesso POSH. A sinistra, vediamo la struttura del consumatore. Il caso è quello di un consumatore business (B2B). Il caso di un consumatore cliente singolo (B2C) è una semplificazione di questo caso, poiché un cliente singolo tipicamente non ha moduli organizzativi a sé: l'intera struttura organizzativa consiste di un modulo. In basso, vediamo l'organizzazione del produttore di arredi, per la parte rilevante per lo scenario POSH (ovviamente, questa parte avrà una struttura organizzativa più complessa). Vi sono tre parti intermediarie: un broker tramite cui POSH fa pubblicità, un gestore pagamenti che gestisce la fatturazione e i pagamenti per conto di POSH, e un partner fornitore di servizi di logistica che si occupa delle consegne – è lo stesso fornitore ad occuparsi sia di piccole sia di grandi merci.

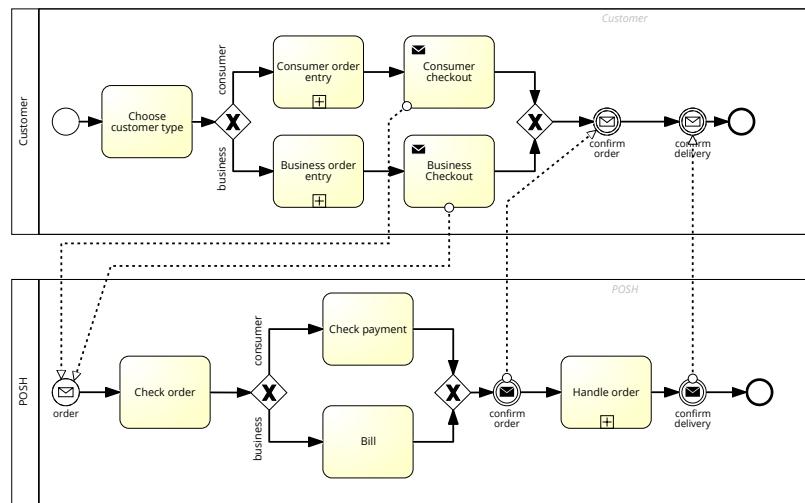


Figura 5.24: Esempio POSH

Si noti che è una scelta di progetto modellare il produttore di arredi come una terza parte. Un'alternativa sarebbe considerarlo come intermediario. La scelta è fatta basandosi sull'osservazione che il produttore produce oggetti di e-business che vengono smerciati e di conseguenza prende il ruolo principale nella catena del valore. Come tale, non è solo un fornitore di servizi 'addizionali' che ha un ruolo minore nella catena del valore.

### Funzioni di business e processi di business

Nel processo di vendite, POSH distingue fra consumatore cliente singolo (C) e client business (B). I consumatori B hanno a disposizione un catalogo più sostanzioso e avanzato per gli ordini, una procedura di checkout con più opzioni di consegna e ricevono la fatturazione offline per pagamenti differiti, mentre il consumatore C paga direttamente (per esempio, usando la carta di credito). POSH decide di realizzare i processi di vendita, marketing & vendite e outbound logistic tramite le funzioni di Figura 5.20. Siccome un'istanza (caso) del processo di vendita coincide con una transazione di vendita, la funzione pubblicità non è compresa nel processo di e-sales ma ha il proprio processo. La negoziazione e la contrattazione non sono considerate applicabili a questo dominio di business – anche se POSH considera di includere la contrattualizzazione in futuro per coprire gli ordini B2B, i quali comportano rischi finanziari più elevati.

La specifica di alto livello del processo appare in Figura 5.24. I clienti devono scegliere nel momento in cui contattano POSH. Sulla base della scelta, vi è uno split nel processo fra consumatore singolo (C) e business (B). B deve effettuare il login in un account pre-esistente per identificarsi. Entrambi usano sottoprocessi per inserire l'ordine. I sottoprocessi hanno ulteriori elaborazioni simili agli esempi visti prima. Dopo il pagamento e la fatturazione, i casi B2C e B2B seguono gli stessi passi. Il passo di gestione ordine è un sottoprocesso di una certa complessità – dentro questo passo, c'è la distinzione fra piccole merci consegnate dal magazzino POSH e grandi merci consegnate dal produttore di mobili.

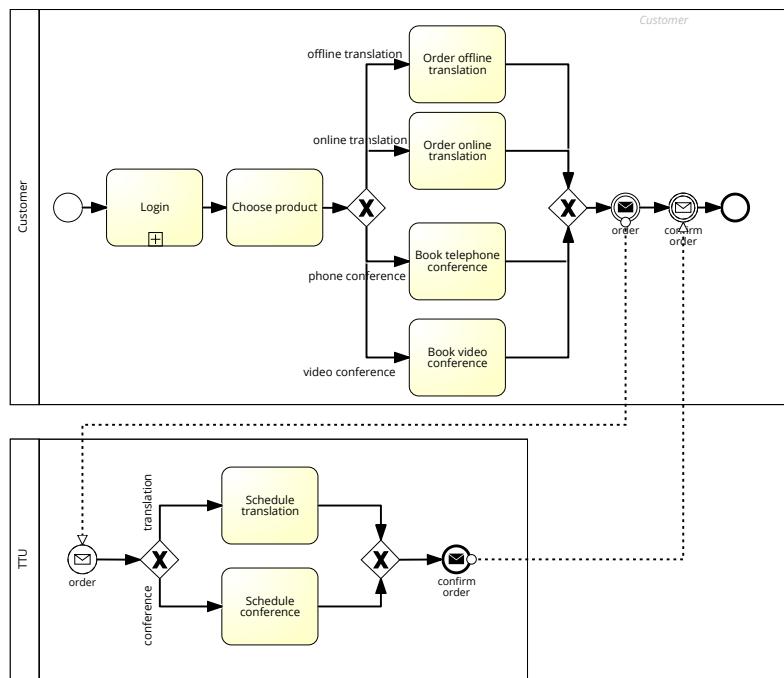


Figura 5.25: Esempio TTU

### 5.7.2 TTU

TTU fornisce vari servizi (come oggetti e-business): traduzione di documenti e interpretariato real-time. Nei primi servizi, fornisce traduzioni offline di documenti fisici e traduzioni online di documenti digitali. Nell'interpretariato, offre servizi di conferenze telefoniche e in video-conferenza. Di seguito, vediamo come questo ha effetto sulle funzioni di TTU a livello di business e di processi. Poi discutiamo come TTU gestisce i cambiamenti.

#### Funzioni di business e processi di business

Per sottolineare le differenze fra i servizi, TTU ha un processo di e-sales orientato al prodotto (diversamente da POSH, che ha una struttura di processo orientata al cliente).

Il processo e-sales di TTU è mostrato in Figura 5.25. Siccome TTU lavora solo con consumatori registrati (tipicamente B), questi devono per prima cosa effettuare il log in. Poi scelgono il prodotto da ordinare e, in base alla scelta, il processo procede.

Diversamente da POSH, la consegna degli oggetti di e-business, la fatturazione e il pagamento non fanno parte del processo di TTU. Questa è una scelta di progetto, la cui ragione sta nel fatto che TTU preferisce fatturare dopo la

consegna del servizio (perché la cifra dipende dallo sforzo nella consegna) e non considera la consegna come parte del processo di vendita.

### Gestione del cambiamento

TTU riconosce che l'innovazione è di fondamentale importanza per la sua esistenza. TTU non impiega traduttori o interpreti – usa freelancer assunti in modo just-in-time. Questo vuol dire che TTU stessa offre servizi digitali: da una parte, mette in contatto organizzazioni che richiedono servizi linguistici e individui che possono fornirli; dall'altra parte, rende disponibili infrastrutture elettroniche che supportano la consegna di questi servizi. Quindi, il business model può facilmente essere copiato dai concorrenti, se rimane troppo statico.

Il facilitatore dell'innovazione in TTU è il dipartimento R&D (Research and Development - Ricerca e Sviluppo). TTU è intenzionato a mantenere questo dipartimento allineato con le operazioni di business. Per questa ragione, TTU ha deciso di costituire una squadra detta di Innovation & Change Management (ICM) formata da membri chiave dell'organizzazione – dal management, ai dipartimenti R&D e operations. Il gruppo ICM opera in base a incontri regolari per discutere possibili nuovi sviluppi in un'atmosfera aperta. I punti in agenda sono sempre tre:

1. Nuovi requisiti per i servizi TTU indicati dai consumatori esistenti.
2. Nuovi sviluppi tecnologici nella traduzione di documenti e nelle conferenze audio e video.
3. Nuove opportunità di business, ovvero, nuovi possibili servizi e nuovi gruppi di consumatori.

# Capitolo 6

## Architetture funzionali

### 6.1 Descrizione

L'architettura è il blueprint (schema costruttivo) della struttura di un sistema complesso:

*“Un’architettura software definisce una struttura che organizza gli elementi software e le risorse di un sistema software. Gli elementi software e le risorse sono rappresentati da sottosistemi. Data una certa architettura software, questi sottosistemi hanno specifiche responsabilità e relazioni verso altri sottosistemi”* (Mathias Weske) [41].

*“L’architettura funzionale di un sistema informativo specifica la struttura del sistema in termini di componenti software funzionali che supportano una specifica funzione e le interfacce che supportano le interazioni fra quei componenti”* (Paul Grefen) [19].

Le architetture nel nostro contesto sono blueprint per il progetto di sistemi informativi di e-business, ovvero per i sistemi basati su IT che supportano funzioni e processi, come discusso in precedenza. In altri termini, le architetture sono progetti delle strutture di sistemi di e-business – non così diversi dalle architetture nel mondo delle costruzioni dove esse sono progetti delle strutture di edifici. Nell'e-business, le architetture costituiscono l'interfaccia fra elementi non-IT (nella prospettiva B e O di BOAT framework) e gli elementi correlati all'IT nella prospettiva T di BOAT, che vedremo nel prossimo capitolo).

Si noti che non discutiamo l'architettura nel più ampio contesto dell'ingegneria del software. Per questi aspetti, facciamo riferimento a testi di Ingegneria del Software, per esempio [5, 37].

Prima, discutiamo la natura e i concetti principali delle architetture dei sistemi informativi, di cui le architetture dell'e-business sono una sottoclasse.

Nelle prossime sezioni, discutiamo i tre livelli di aggregazione (dalla struttura dei sistemi che coinvolgono organizzazioni multiple alla struttura dei sistemi che supportano parte di una singola organizzazione): il livello market (Market-level), il livello Party (Party-level) e il livello sistema (System-level).

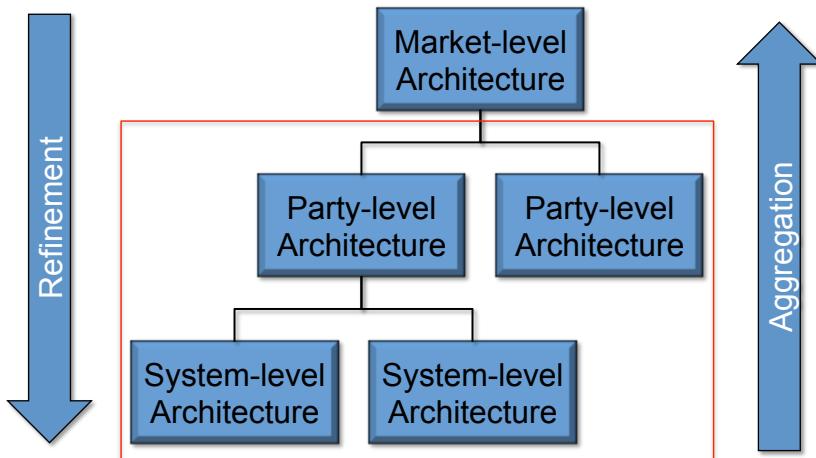


Figura 6.1: Livello di aggregazione delle architetture

## 6.2 Livelli di aggregazione e astrazione

Si possono descrivere le architetture dei sistemi informativi a vari livelli - da strutture semplici, di alto livello a strutture dettagliate e di basso livello. Per essere sicuri di che cosa descriviamo, dobbiamo capire la natura di questi livelli. Sotto, spieghiamo due dimensioni importanti lungo le quali descriviamo i livelli architetturali: aggregazione e astrazione.

### 6.2.1 La dimensione aggregazione

Le architetture possono essere specificate a vari livelli di dettaglio detti anche livelli di aggregazione. I livelli di aggregazione sono posizionati lungo la dimensione aggregazione delle architetture. Spostandoci in basso lungo questa dimensione (diminuendo il livello di aggregazione), facciamo zoom in nelle architetture, analizzando strutture più piccole a maggior livello di dettaglio. Spostandoci in su (aumentando il livello di aggregazione), facciamo zoom out, analizzando strutture più grandi a un livello di dettaglio minore. Questo modo di procedere è analogo a quanto già visto per l'aspetto Organizzazione, nel raffinamento dal livello 0 al livello 4.

In questo capitolo, discutiamo le architetture dei sistemi informativi a tre livelli di aggregazione:

- *Architetture Market-level*: descrivono la struttura dei sistemi di e-business al livello in cui vari partecipanti, o parti, si inseriscono nel mercato dell'e-business. Si tratta di architetture inter-organizzative, poiché si concentrano sulle relazioni fra i sistemi dei partecipanti in un mercato. Descrivono quindi le interazioni tra i sistemi dei diversi partecipanti.

- *Architetture Party-level:* descrivono le strutture a livello di organizzazioni singole. Come descriviamo nei sistemi di e-business, le interfacce verso il mondo esterno (mercato e-business in cui l'organizzazione opera) sono importanti, ma il mondo esterno non è descritto. Quindi, le architetture Party-level sono intra-organizzative.
- *Le architetture System-level* descrivono le strutture dei sottosistemi di un partecipante all'e-business. Un sistema individuale può essere un componente funzionale in un sistema Party-level che supporta una funzione specifica (come identificata nella prospettiva O di BOAT), o un sistema general-purpose che supporta varie funzioni.

Le architetture Market-level possono essere raffinate in architetture Party-level, a loro volta raffinabili in architetture System-level. Viceversa, è possibile aggregare le architetture System-level in Party-level, aggregabili in Market-level. Mostriamo entrambe le relazioni in Figura 6.1. Il numero di architetture Party-level relative ad un'architettura Market-level dipende dal numero di partecipanti nel mercato (mostriamo solo due architetture Party-level in figura, per semplicità). Il numero di architetture System-level relative ad un'architettura Party-level dipende dal numero di sistemi usati dalla parte corrispondente (mostriamo ancora due architetture in figura per semplicità). Il numero di sistemi dipende dalla complessità della funzionalità supportata dal partecipante e da quella offerta dai sistemi: più complessa la funzionalità supportata da una parte, maggiore il numero dei sistemi coinvolti, mentre sistemi con ampie funzionalità implicano meno sistemi.

Si noti che parti multiple simili in un mercato possono condividere la stessa architettura Party-level: ciò significa che hanno organizzato la struttura dei loro sistemi in modo simile. Ma possono anche avere diverse architetture Party-level, pur conformandosi alla stessa architettura Market-level.

### 6.3 La dimensione astrazione

Le architetture di sistema informativo (e quindi le architetture e-business) possono essere descritte a vari livelli di astrazione: in termini astratti ma anche in termini molto concreti - e in modi a metà fra questi estremi. I livelli di astrazione si posizionano lungo la dimensione di astrazione. Descrivendo un'architettura in termini astratti, i componenti sono specificati in termini generici, tipicamente etichettando la loro funzionalità (o classe di funzionalità). Un'etichetta può essere Sistema di Pagamento o sistema Enterprise Resource Planning (ERP). Quando un'architettura è descritta in termini molto concreti, i componenti sono descritti con molta precisione, generalmente etichettando la funzionalità e il tipo esatto del componente in termini di vendor del software e versione del software.

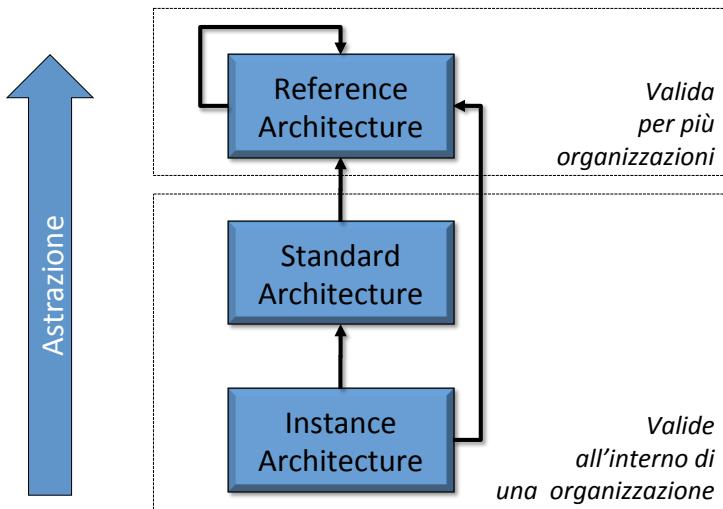


Figura 6.2: Livelli di astrazione nelle architetture

### 6.3.1 Architetture di riferimento, standard e istanze di architetture

Parlando di architetture di sistemi informativi, è bene distinguere fra vari tipi di architetture, a seconda della loro origine e del loro scopo. Abbiamo tre tipi di architetture a diversi livelli di astrazione:

- *architettura istanza*: architettura di uno specifico sistema informativo in uno specifico contesto. In altre parole, una specifica istanza di un tipo di sistema. Un esempio è l'architettura Party-level di una organizzazione. Un'architettura istanza non è disegnata esplicitamente per contenere strutture riusabili in vari contesti.
- *architettura standard*: architettura definita come standard per una classe di sistemi informativi in uno specifico contesto organizzativo, tipicamente una grande azienda o un consorzio di aziende. Si può definire un'architettura standard, per esempio, per tutti i sistemi di procurement appartenenti a una holding di aziende. Come tale, un'architettura standard può essere la base per la definizione di un'architettura istanza di un sistema specifico in una certa classe e nel contesto organizzativo dell'architettura standard.
- *architettura di riferimento*: architettura definita come uno standard per una classe di sistemi per varie organizzazioni, ovvero, senza uno specifico contesto organizzativo.

Un'architettura di riferimento è spesso definita da una parte indipendente, come un ente di standardizzazione o una organizzazione governativa. Può essere la base per definire un'architettura standard, o per definire

direttamente un'architettura istanza. Può anche essere la base per altre architetture di riferimento, tipicamente più elaborate.

I tre tipi di architetture e le loro relazioni sono mostrate in Figura 6.2, in cui gli archi fra classi indicano relazioni ‘possono essere basate su’ fra architetture.

Poiché l'aspetto interoperabilità fra sistemi è importantissimo, le architetture di riferimento e standard sono molto rilevanti. L'uso di un'architettura standardizzata è la base per assicurare che i sistemi basati su tali architetture andranno bene insieme ('fit together'), cioè saranno in grado di comunicare e collaborare nei processi automatizzati - entro i confini dell'organizzazione ma, molto più importante, anche attraverso i confini delle organizzazioni.

Nella prospettiva architettonica di BOAT, le architetture di riferimento devono essere indipendenti dalla tecnologia: ci si dovrebbe focalizzare prima di tutto su quali funzionalità vanno inserite, poi su come ogni funzionalità è effettivamente realizzata adottando una certa tecnologia.

Molte architetture di riferimento attualmente usate nell'interazione di e-business sono invece dipendenti dalla tecnologia. In altre parole, non si occupano di aspetti generali, concettuali dell'interazione di e-business come funzionalità astratte, ma mettono in evidenza la descrizione della struttura di una specifica soluzione tecnologica. Quindi, queste architetture appartengono alla prospettiva tecnologica (T) di BOAT.

## 6.4 Architetture Market-level

Mentre nel modello organizzativo rappresentiamo le funzionalità, nel modello di architettura funzionale rappresentiamo i componenti concettuali di un sistema.

Un'architettura *Market-level* descrive la struttura di un intero sistema informativo e si focalizza sui messaggi scambiati tra i partecipanti (un primo esempio è illustrato nella Figura 6.3).

Nella figura, l'architettura rappresenta uno scenario di e-business con due parti in collaborazione (A a sinistra e B a destra) e un intermediario (parte I in basso alla figura).

Questa figura corrisponde al livello 2 delle strutture organizzative dell'aspetto organizzativo di BOAT. Vediamo ora come raffinare questa rappresentazione in un'architettura funzionale.

Un'architettura *Market-level* descrive come interagiscono i sistemi software delle organizzazioni che prendono parte allo scenario di e-business. Usiamo la seguente definizione (una specializzazione della definizione generale di architettura di sistema informativo discussa nella sezione precedente):

“L'architettura *Market-level* di un sistema e-business per uno scenario definisce la struttura di quel sistema a livello inter-organizzativo in termini di componenti software funzionali che supportano funzioni specifiche delle principali parti coinvolte nello scenario e di interfacce di alto livello che supportano le interazioni fra quei componenti.”

Vari termini in questa definizione meritano alcune spiegazioni. Un'architettura *Market-level* ha carattere inter-organizzativo. Ciò significa che essa descrive

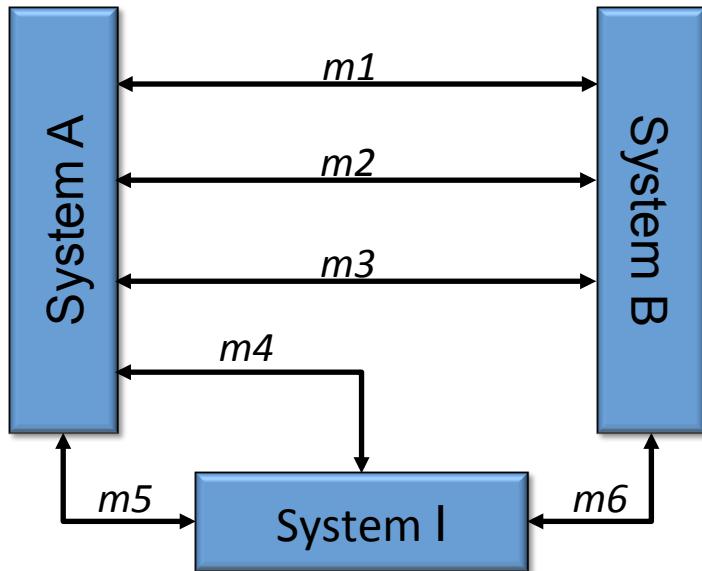


Figura 6.3: Architettura a livello di mercato ad alta aggregazione

i sistemi informativi di tutti i partner partecipanti a uno scenario, non un singolo partner. Come visto per la prospettiva O di BOAT, nel caso semplice ciò significa due parti principali e un certo numero di intermediari. In uno scenario più interconnesso a rete, sono anche possibili scenari multi-partite, cosa che porta ad architetture Market-level che coprono i sistemi di tante parti coinvolte. L'enfasi è il supporto alla collaborazione fra le parti. Un'architettura Market-level contiene componenti software funzionali, cioè sistemi software che supportano specifiche funzioni di business. Contiene anche interfacce ad alto livello, cioè i collegamenti fra questi moduli software. L'enfasi è sulle interfacce fra componenti che appartengono a parti diverse, perché sono proprio queste interfacce ad implementare la collaborazione fra parti (ovvero, le interazioni Market-level).

Gli archi sono etichettati con i tipi di messaggi scambiati (etichette da m1 a m6 in figura 6.3), per esempio messaggi pubblicitari, ordini di acquisto o pagamenti. Si noti che un'interfaccia spesso scambia più di un tipo di messaggio (per una interfaccia bidirezionale, in genere almeno due). Pertanto, ogni etichetta di messaggio rappresenta un insieme di tipi di messaggi. Le corrispondenze tra le etichette e i nomi associati ai diversi tipi di messaggi (più messaggi per ciascuna etichetta) vengono rappresentate in una tabella riassuntiva o nel dizionario dei dati. Vedremo esempi più elaborati parlando dei casi.

Visto che un'architettura Market-level descrive le interazioni fra i sistemi delle principali parti che prendono parte a uno scenario e-business, la sua struttura è determinata dalla topologia di queste parti (la rete di business) in collaborazione. Esiste una forte congruenza (somiglianza strutturale) fra modelli

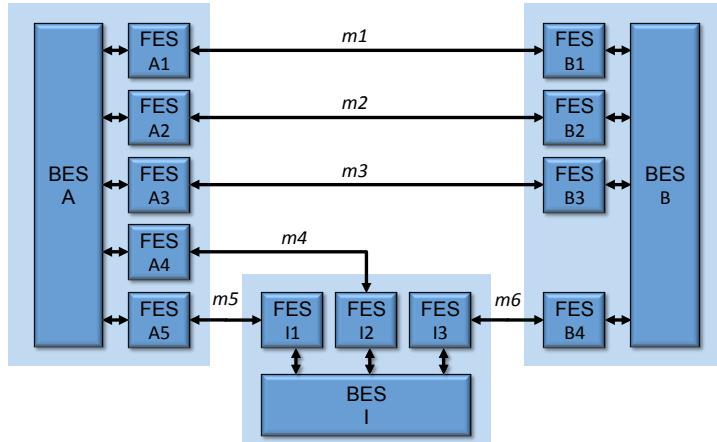


Figura 6.4: Architettura Market-level (a livello di mercato)

di alto livello nella prospettiva O di BOAT e le architetture Market-level della prospettiva A, poiché entrambe contengono interazioni con le stesse reti di organizzazioni: la prospettiva O modella le interazioni fra organizzazioni, la prospettiva A le interazioni fra i sistemi di quelle organizzazioni.

La Figura 6.4 mostra un'architettura Market-level semplice, astratta e vi aggiunge altri elementi che compongono l'architettura per ogni parte. Le aree ombrigate nella figura indicano i confini delle parti. I moduli etichettati con *BES* (Back End System) indicano i componenti back end corrispondenti al modello organizzativo di BOAT. Analogamente, i moduli etichettati con *FES* (Front End System) indicano i componenti front end. Mostreremo alcuni esempi concreti di architetture alla fine del capitolo, considerando i casi di studio.

Come discusso in dettaglio più avanti in Sezione 6.7, c'è distinzione fra funzionalità FE (Front End) al livello organizzativo, e FES (Front End System) a livello di architettura funzionale. Il mapping fra i due non è necessariamente uno a uno.

Un'architettura Market-level si focalizza sulle relazioni fra partecipanti, mostrando i sistemi e le interfacce che implementano quelle relazioni. Nella figura, i sistemi sono riquadri, le interfacce sono archi tra le parti.

Rispetto ai sistemi nell'architettura, vediamo la separazione fra sistemi front end e back end delle tre parti. Questa separazione è analoga a quella tra funzioni di front end e back end nel modello organizzativo di BOAT. Le interfacce tra sistemi front end e back end di un'organizzazione vengono elaborate nelle architetture Party-level, come vedremo nella prossima sezione.

I sistemi front end supportano le funzioni front end (per esempio, pubblicità, acquisto o pagamento). Poiché l'interazione fra sistemi delle parti è modellata nell'architettura Market-level, i sistemi front end vengono identificati uno per uno individualmente.

Si noti che non necessariamente c'è un mapping uno-a-uno tra funzioni front

end in O e sistemi front end in A: un sistema può supportare varie funzioni, o una funzione richiedere sistemi multipli. Si noti anche che la multicanalità fra due parti può avere impatto sull'architettura: ogni canale può dare esito a una coppia di sistemi front end con un'interfaccia.

I sistemi back end sono mostrati come ‘black box’, perché la loro struttura interna non è rilevante per le interfacce a Market level: una parte ‘vede’ solo i sistemi front-end dei suoi collaboratori. I sistemi back end vengono elaborati nelle architetture Party-level (come vedremo nella prossima sezione).

## 6.5 Architettura Party-level

Ora facciamo un passo in profondità nella gerarchia di aggregazione (come in Figura 6.1) e parliamo di architetture Party-level. Prima discutiamo il concetto di architettura Party-level; poi illustriamo la struttura delle architetture Party-level. Concludiamo la sezione con la spiegazione della relazione fra architetture Party-level e Enterprise Architecture.

Un'architettura Party-level descrive la struttura dei sistemi di ogni partecipante (anche di un intermediario) in uno scenario. Usiamo la seguente definizione (ancora una specializzazione della definizione generale di architettura discussa in Sezione 6.1):

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

Mentre un'architettura Market-level è inter-organizzativa, un'architettura Party-level è intra-organizzativa. Come tale, un'architettura Party-level è un raffinamento di una porzione di un'architettura Market-level: la porzione associata ad uno specifico partecipante.

In un'architettura Party-level, diamo più dettagli che nel Market-level tramite i seguenti raffinamenti:

- Dettaglio dei sistemi di back end (che sono black box nell'architettura Market-level).
- Interfacce tra sistemi back end.
- Dettaglio delle interfacce tra sistemi front end e back end.

### 6.5.1 Struttura delle architetture Party-level

Come mostrato in Figura 6.5, i componenti delle architetture Party-level sono:

- *Componenti generici*: i principali componenti funzionali.

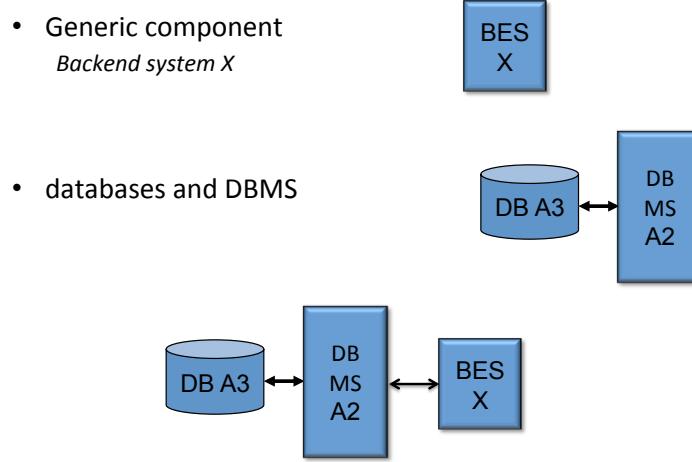


Figura 6.5: Elementi dell’architettura Party level

- *Database e DBMS (sistemi piattaforma)*: denotano componenti per la memorizzazione di dati e i componenti necessari per l’accesso ai dati, considerati qui come componenti funzionali, cioè sistemi che forniscono la funzionalità di gestione dei dati, astrazione dei componenti tecnologici specifici.
- *Connettori*: entità di collegamento fra i componenti. Un esempio appare nella parte inferiore della Figura 6.5, che connette un componente generico, un DBMS e un componente di memorizzazione. Le connessioni con i DBMS e i database non vengono etichettate, in quanto le etichette poste sui database sono rappresentative del loro contenuto informativo.

La Figura 6.6 mostra un esempio di architettura Party-level astratta (vedremo un esempio concreto nei casi a fine capitolo). Questa architettura è il raffinamento del partecipante A dell’architettura Market-level di Figura 6.4. Nella figura, il confine del partecipante A è indicato dall’area ombreggiata.

Nell’architettura Party-level, mostriamo gli stessi sistemi front-end dell’architettura Market-level corrispondente per il partecipante che stiamo esaminando (da FES A1 a FES A5). Gli altri partecipanti dell’architettura Market-level sono *entità esterne*, disegnate come rettangoli arrotondati e interfacce esterne ombreggiate verso di loro, corrispondenti a quelli della architettura Market-level. Si noti che i messaggi delle interfacce esterne sono coerenti con quelli a livello di architettura Market-level. Il resto dell’architettura Party-level è un’elaborazione del sistema back end a black box nella architettura Market-level del partecipante considerato.

Nell’esempio, mostriamo tre sistemi di back end (da BES A1 a BES A3) e due sistemi piattaforma: il DBMS A1 e il DBMS A2. I sistemi back end

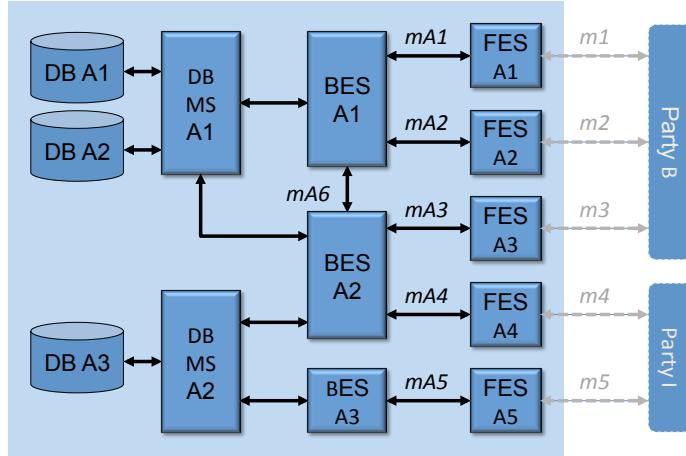


Figura 6.6: Architettura Party-level (a livello di partecipante)

supportano i processi back end, come l'amministrazione finanziaria, la gestione prodotti e la gestione del magazzino. I DBMS gestiscono i dati per i sistemi back end, dati che risiedono nei database da DB A1 a DB A3.

Le interfacce tra sistemi front end e back end e fra sistemi back end sono etichettate con i tipi di messaggi (che possono essere insiemi di tipi di messaggi, come nell'architettura Market-level). Si noti la scelta di includere il nome del partecipante (o un'abbreviazione) nelle etichette, per evitare confusione con le etichette nel Market-level (per esempio, quando il numero di interfacce Market-level aumenta).

### 6.5.2 Architetture Party-level e Enterprise Architecture

Un'architettura funzionale rappresenta i componenti funzionali relativi a uno scenario cui un'organizzazione partecipa. Quindi, vi è una relazione fra architettura aziendale e architettura(e) di business di un'organizzazione.

Tuttavia, un'organizzazione può essere coinvolta in scenari multipli (come succede anche in scenari tradizionali multipli). Pertanto, un'architettura Party-level è un sottoinsieme di un'architettura funzionale complessiva, parte dell'EA, adattata a uno specifico scenario.

Tipicamente, se un'organizzazione è coinvolta in vari scenari, riusa parte dei propri sistemi nell'ambito dei diversi scenari. Di conseguenza, le architetture Party-level per i vari scenari si sovrappongono nell'EA, ovvero, hanno funzionalità comuni condivise in termini di sistemi e di interfacce identificate nell'architettura. Ciò è illustrato in Figura 6.7 per una organizzazione coinvolta in quattro diversi scenari: Procurement, Vendite, HRM, Finance, che avranno alcuni componenti in comune tra tutti i sistemi o una parte di essi.

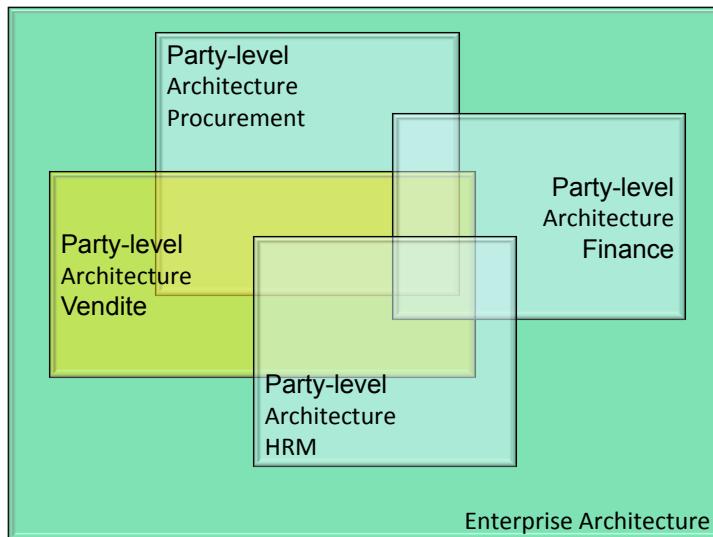


Figura 6.7: Architettura aziendale

## 6.6 Architettura System-level

Ora ci spostiamo a un livello più profondo nella dimensione di aggregazione delle architetture (si veda Figura 6.1) e arriviamo alle architetture System-level per i sistemi informativi. Una *architettura system-level* è un raffinamento di un componente in un'architettura Party-level, come discusso nella sezione precedente.

Un'architettura System-level descrive la struttura di uno specifico sistema informativo di un partecipante dello scenario. Usiamo la seguente definizione (come le definizioni precedenti, una specializzazione della definizione generale di architettura di sistema informativo discussa nella Sezione 6.1):

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

Ovviamente, un'architettura System-level è intra-organizzativa, poiché è un raffinamento di parte di un'architettura Party-level intra-organizzativa. Un'architettura System-level è la specifica della struttura di un sistema front end o di un sistema back end identificato in un'architettura Party-level. I sistemi piattaforma sono per lo più sistemi standard acquisiti 'off-the-shelf'. La loro architettura interna è in genere poco interessante dal punto di vista dell'e-business in rete.

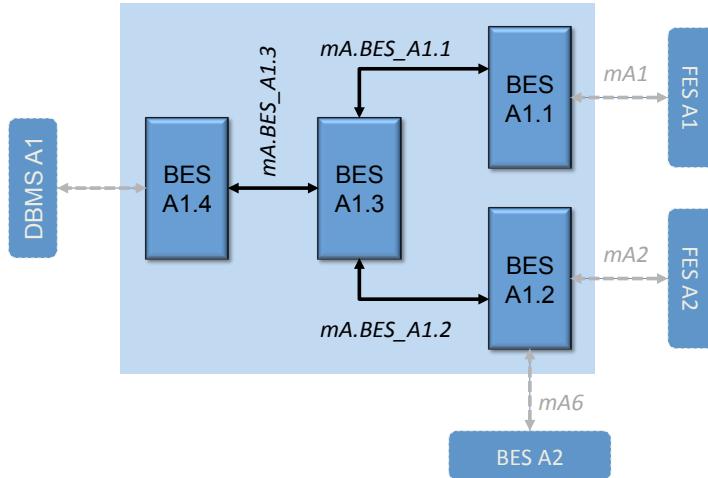


Figura 6.8: Architettura a livello di sistema

### 6.6.1 La struttura delle architetture System-level

La struttura di un'architettura System-level dipende notevolmente dalla natura del sistema descritto. Le architetture di sistemi semplici consistono di pochi componenti, quelle di sistemi complessi possono essere formate da molti componenti. La Figura 6.8 mostra un esempio di architettura a livello astratto System-level (vedremo esempi nei casi a fine capitolo). L'architettura è l'elaborazione del sistema BES A1 dell'architettura Party-level di Figura 6.6 (i confini di questo sistema sono mostrati nell'area ombreggiata della figura). L'architettura System-level mostra che *BES A1* internamente consiste di quattro moduli, etichettati da *BES A1.1* a *BES A1.4*.

Come mostra la figura, i sistemi con cui il sistema in esame comunica sono mostrati come entità esterne, indicati con rettangoli con angoli arrotondati. Le interfacce con questi sistemi devono essere congruenti con quelle specificate nell'architettura Party-level corrispondente (sia per quanto riguarda la struttura dell'interfaccia che l'insieme dei messaggi). Le interfacce tra i componenti interni del sistema in considerazione sono specificate ed etichettate con insiemi di messaggi. Questi sono mA.BES\_A1.1 - mA.BES\_A1.3 nell'esempio - con l'etichetta del modulo inclusa nel messaggio per evitare fraintendimenti al livello Party.

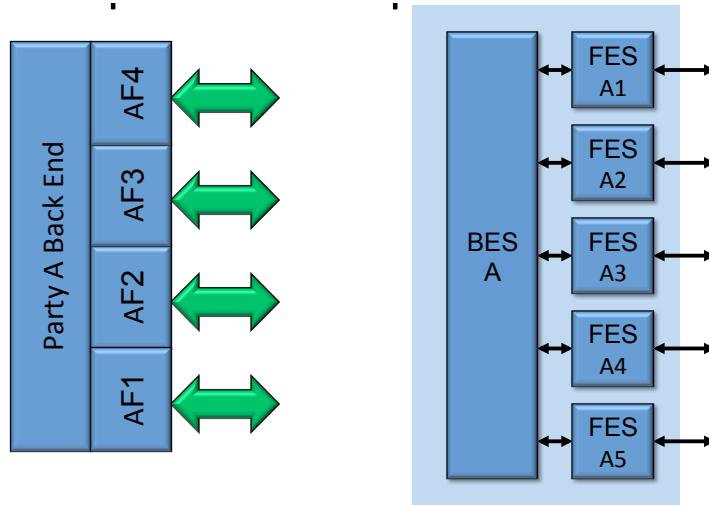


Figura 6.9: Relazione tra aspetti organizzativi e aspetti funzionali

## 6.7 Relazioni tra architetture organizzative e architetture funzionali

Come abbiamo illustrato nel capitolo, le architetture ad alto livello descritte nella prospettiva A di BOAT hanno varie somiglianze a livello funzionale con le strutture organizzative della prospettiva O. È possibile trovare le corrispondenze tra gli elementi appartenenti alle due prospettive. In questa sezione discutiamo due principi che possono guidare nell'identificazione di queste corrispondenze.

Nell'e-business, è essenziale che vi sia un forte allineamento tra i requisiti a livello di business e i sistemi informativi realizzati, così da ottenere applicazioni di e-business che soddisfino sia i requisiti funzionali sia quelli non funzionali. Questo allineamento può essere ottenuto cercando di trovare un isomorfismo tra le strutture organizzative definite nella prospettiva O e le strutture dei sistemi informativi definiti nella prospettiva A.

Gli elementi rilevanti identificati nella prospettiva organizzativa devono essere visibili anche nella prospettica architettonale, e le dipendenze esistenti tra le due prospettive devono essere rappresentate in modo chiaro.

Nel caso più semplice, esiste una relazione uno-a-uno tra le descrizioni delle funzionalità ad alto livello in O e gli elementi architetturali in A: ogni modulo nella prospettiva O corrisponde in questo caso ad un modulo nella prospettiva A. Spesso però la realtà è un po' più complessa. Ad esempio, nella situazione mostrata in Figura 6.9, a sinistra vediamo la struttura organizzativa a livello 4 di un partecipante A, mentre a destra vediamo l'architettura Market-level della stessa organizzazione. Il numero di moduli di front end a livello organizzativo (quattro) è diverso dal numero di moduli dell'architettura funzionale (cinque). Non è quindi possibile individuare una corrispondenza uno-a-uno.

|                     |     | A Aspect Modules |           |           |           |           |
|---------------------|-----|------------------|-----------|-----------|-----------|-----------|
|                     |     | FES<br>A1        | FES<br>A2 | FES<br>A3 | FES<br>A4 | FES<br>A5 |
| O Aspect<br>Modules | AF1 | X                |           |           |           |           |
|                     | AF2 |                  | X         |           |           | X         |
|                     | AF3 |                  |           | X         |           |           |
|                     | AF4 |                  |           |           | X         | X         |

Figura 6.10: Matrice di corrispondenza tra aspetti organizzativi e aspetti funzionali

È fortemente consigliato descrivere la corrispondenza in modo esplicito in una situazione concreta. Questo renderà esplicite le dipendenze ad alto livello nel momento in cui si verificheranno cambiamenti. La descrizione può essere espressa nella forma di una matrice di corrispondenza, come mostrato ad esempio in Figura 6.10 per la situazione descritta in Figura 6.9. La matrice specifica quali moduli della prospettiva O dipendono da quali moduli della prospettiva A per il supporto automatizzato. Nella matrice, ogni modulo organizzativo dipende da un proprio modulo architettonale dedicato; vi sono poi due moduli organizzativi che dipendono anche da un secondo modulo condiviso a livello di architettura.

## 6.8 Esempi

Nel seguito vengono presentati alcune possibili architetture per i casi di studio presi in esame. Le architetture presentate, in particolare per il caso POSH, sono possibili alternative architettoniche, alcune delle quali verranno discusse in dettaglio nei prossimi capitoli. Si noti che oltre ai diagrammi che rappresentano le architetture funzionali, per i messaggi viene fornita una tabella che, per ciascun tipo messaggio, illustra il contenuto dei flussi di informazioni scambiate tra le parti, che possono contenere anche più messaggi. Infatti, l'etichetta posta sui flussi dei diagrammi delle architetture funzionali rappresenta in genere più di un messaggio che è necessario scambiare nella comunicazione per ottenere la comunicazione voluta. Spesso, l'interazione prende la forma di richiesta/risposta, e l'arco viene indicato nel diagramma come arco bidirezionale.

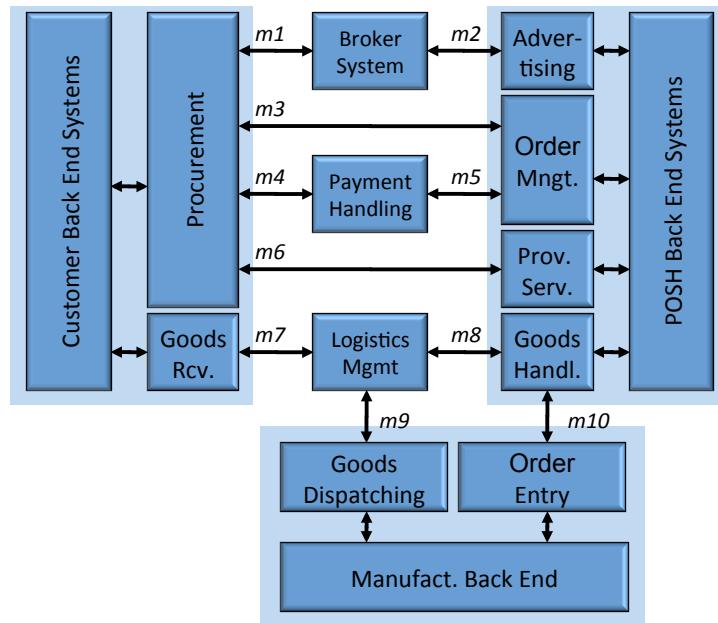


Figura 6.11: Architettura a livello di mercato per POSH

| Message Set | Contents Exchanged                              |
|-------------|---|
| <b>m1</b>   | Provider search request, Provider search result |
| <b>m2</b>   | Provider profile, Provider offer                |
| <b>m3</b>   | Order, Order confirmation                       |
| <b>m4</b>   | Payment order                                   |
| <b>m5</b>   | Payment notification                            |
| <b>m6</b>   | Service request, Service information            |
| <b>m7</b>   | Delivery notification, Delivery confirmation    |
| <b>m8</b>   | Shipment request, Shipment confirmation         |
| <b>m9</b>   | Shipment request, Shipment confirmation         |
| <b>m10</b>  | Production order, Production confirmation       |

Figura 6.12: Descrizione dei messaggi scambiati da POSH

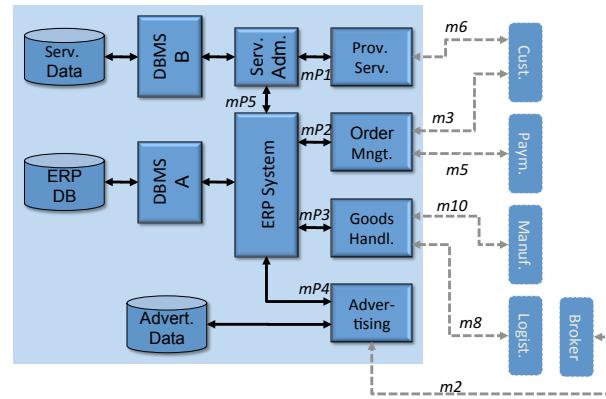


Figura 6.13: Architettura Party-level per POSH con sistema ERP

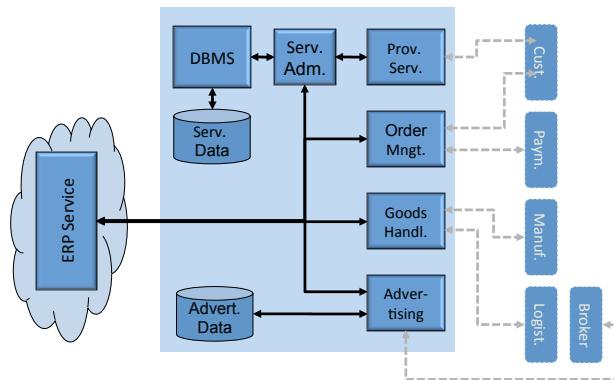


Figura 6.14: Architettura Party-level per POSH con servizio ERP

### 6.8.1 POSH

Si presentano ora alcune soluzioni alternative architetturali nelle Figure 6.13 (con ERP) e 6.14 (con ERP come servizio esterno).

### 6.8.2 TTU

Un'architettura per TTU è illustrata nella Figura 6.16.

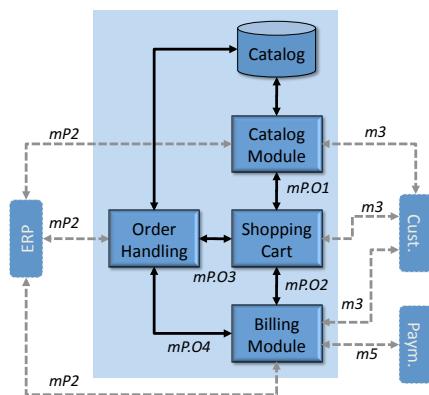


Figura 6.15: Architettura System-level per POSH

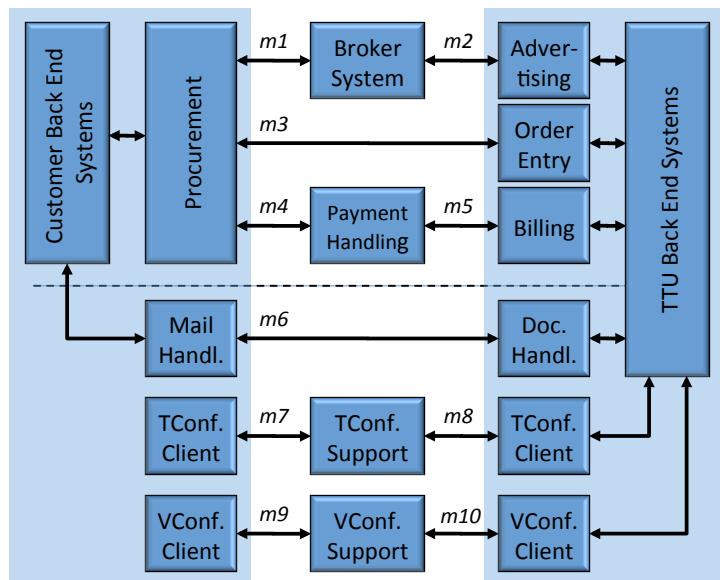


Figura 6.16: Architettura a livello di mercato per TTU



## Capitolo 7

# Introduzione agli aspetti tecnologici

### 7.1 Descrizione

I componenti tecnologici dei sistemi informativi possono essere classificati secondo le categorie identificate in Figura 7.1.

Considerando l'architettura dei sistemi informativi, la Figura 7.1 può essere vista come la mappa delle principali tecnologie utilizzabili per l'effettiva realizzazione dei sistemi stessi. Le tecnologie si dividono in tre livelli: (i) livello applicativo, (ii) livello di piattaforma e (iii) livello di architettura fisica. La selezione effettiva delle tecnologie da utilizzare ad ogni livello viene anche guidata dal dominio applicativo considerato (*Application Domains* in Figura 7.1). Ogni dominio applicativo infatti è caratterizzato da obiettivi e esigenze diverse che inevitabilmente portano a scelte e realizzazioni differenti.

### 7.2 La tecnologia a livello applicativo

La tecnologia a livello applicativo comprende tutte le applicazioni software che è possibile adottare a supporto delle diverse funzionalità aziendali nei sistemi operazionali e informazionali. L'insieme dei componenti a livello applicativo di un'azienda viene definito *portafoglio applicativo*. Il portafoglio viene costruito solitamente selezionando alcune delle seguenti 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:** insieme di moduli per la raccolta e l'analisi di informazioni utili alla valutazione dell'attuale situazione aziendale e alla previsione di eventi futuri.

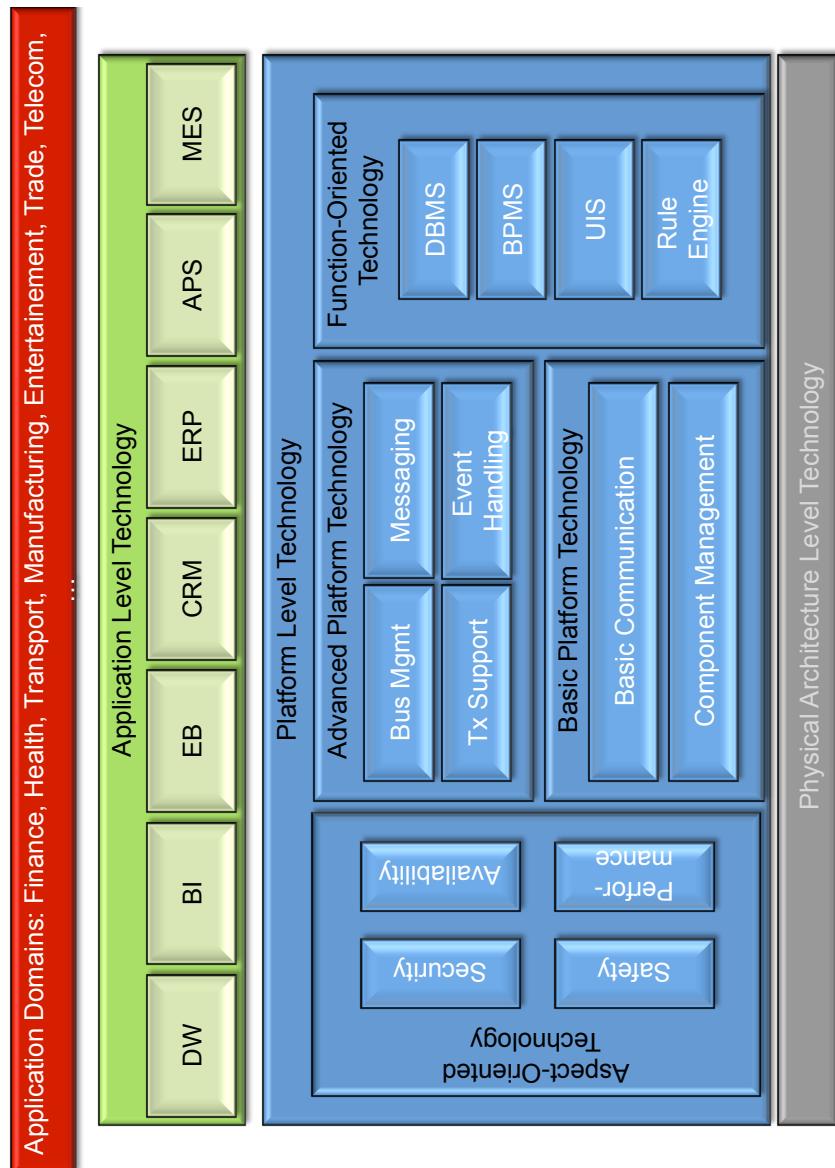


Figura 7.1: Componenti tecnologici

- **EB - E-Business:** questo termine 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. Detti moduli 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. Per esempio in un'azienda manifatturiera, i moduli ERP sono in grado di supportare tutte le attività primarie e di supporto costituenti la catena del valore di Porter.
- **APS - Advanced Planning and Scheduling:** insieme di applicazioni usate in ambito manifatturiero per gestire in modo ottimale 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.

Nel decidere quali tecnologie adottare a livello applicativo, oltre alla tipologia bisogna considerare anche l'approccio migliore per la realizzazione del sistema scegliendo anche tra le strategie Make-or-Buy (già discusse nel Capitolo 2). Per funzionalità ormai largamente in uso da varie organizzazioni in campi diversi, e quindi giunte ad un alto livello di standardizzazione, spesso si utilizzano soluzioni di tipo Buy , che permettono di adottare moduli software COTS (Commercial Off-The-Shelf), come tipicamente avviene per le suite ERP, basate su procedure consolidate e personalizzabili in base alle esigenze aziendali. Tali moduli dovranno poi essere integrati con moduli realizzati in modalità Make per rispondere ad esigenze specifiche dell'organizzazione.

I moduli applicativi sopra elencati corrispondono a componenti funzionali nelle architetture Party-level, e possono a loro volta essere composti da più moduli funzionali di dettaglio che possono essere dettagliati nelle architetture di livello System-level.

### 7.3 La tecnologia a livello di piattaforma

L'implementazione delle applicazioni richiede l'utilizzo di diverse tecnologie infrastrutturali di piattaforma, in grado di gestire aspetti quali l'accesso ai dati, la comunicazione tra applicativi, la sicurezza e altri aspetti non funzionali. Possiamo dividere le tecnologie a questo livello in quattro macro-categorie: (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 (quindi i processi) di business.
- **UIS - User Interface Systems:** insieme di servizi capaci di supportare l'interazione da parte degli utenti e gestire interfacce di tipo diverso in modo adattivo e flessibile, anche 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. In Figura 7.1 in particolare si evidenziano sistemi per garantire la sicurezza delle applicazioni e delle comunicazioni, la disponibilità delle funzionalità erogate, 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. Per quanto riguarda i sistemi per la gestione dei componenti, questi consentono di tenere traccia dei componenti e i servizi in uso nell'impresa per facilitarne il riuso e la tracciabilità.

Ultima categoria di tecnologie sono quelle che si occupano di aspetti avanzati (**Advanced platform technology** in Figura 7.1). Tra queste ad esempio citiamo i sistemi di bus management, di messaging, di gestione di eventi e delle transazioni (Tx support) in grado di supportare l'integrazione e la comunicazione tra applicazioni.

## 7.4 La tecnologia a livello di architettura fisica

Come discusso nell'introduzione, i sistemi informativi attuali si basano sui sistemi informatici (sistemi IT) che ne costituiscono la componente automatizzata. Le architetture dei sistemi informatici si sono sviluppate ed evolute nel corso degli anni passando da schemi centralizzati a modelli distribuiti, maggiormente rispondenti alle necessità di decentralizzazione e di cooperazione delle moderne organizzazioni. Le tecnologie a livello di architettura fisica comprendono i paradigmi con cui è possibile allocare le applicazioni sulle macchine fisiche, specificando eventuali soluzioni di outsourcing come il Cloud Computing. Dal punto di vista architettonico è infatti importante definire chi gestisce l'infrastruttura

che viene fornita. È possibile ragionare in termini di gestione dell'infrastruttura fisica oppure a livello di piattaforma o anche di applicazioni. La tendenza recente è la virtualizzazione delle risorse, rendendole cioè disponibili come *servizi*. In questo caso, non si avrà a disposizione un server fisico, ma si parlerà di server virtuali (Virtual Machine - VM), che avranno caratteristiche analoghe ai corrispondenti server fisici. La virtualizzazione delle risorse fisiche rende più flessibile la gestione del sistema, distribuendo le risorse virtuali sulle risorse fisiche effettivamente disponibili in base alle necessità di calcolo.

Un altro concetto importante da introdurre nel caso delle infrastrutture a servizi è quello di *Enterprise Service Bus* (ESB). L'ESB fornisce una infrastruttura di comunicazione tra servizi, consentendo di istruire opportunamente i messaggi scambiati tra servizi senza conoscerne la collocazione fisica o su una piattaforma. Il concetto di ESB è utilizzato nei diagrammi delle architetture funzionali dell'EA per indicare una infrastruttura di comunicazione tra componenti, non necessariamente legato alla scelta di una specifica infrastruttura tecnologica. In Figura 7.1 vediamo il Bus management come componente a livello di piattaforma che abilita la virtualizzazione, unitamente agli altri meccanismi di supporto allo scambio dei messaggi e di gestione di eventi, appoggiandosi sull'infrastruttura fisica disponibile.

Dettagli sulle tecnologie relative alle architetture fisiche verranno descritti nei capitoli successivi.

## 7.5 Domande sulla Parte II

- Parlare di Business Processes: principi, linguaggi di descrizione, esempi.
- Secondo l'approccio BOAT, discutere le differenze tra la progettazione degli aspetti relativi all'organizzazione e quelli relativi alle architetture funzionali, e discutere in particolare la rappresentazione dell'interazione tra le parti.
- Illustrare il modello di Porter e discutere le relazioni con gli aspetti di modellazione organizzativa di un sistema informativo.
- *Minicaso - modelli di business.* Su un proprio caso di studio definire uno scenario con il modello di business associato, utilizzando le tre dimensioni definite nel modello. Indicare per quale dimensione è probabile che il valore cambi in un futuro prossimo. Illustrate la vostra analisi a un compagno (e ascoltate la sua).



# **Parte III**

# **Aspetti tecnologici**



## Capitolo 8

# Tecnologie a livello applicativo

### 8.1 Componenti funzionali

Come introdotto nel precedente capitolo, il sistema informativo è un sistema complesso composto da diverse applicazioni che sono nella maggior parte dei casi *modulari*: il modulo è definito come un componente software autonomo associato a una o più funzionalità, quindi con obiettivi specifici. L'architettura è così partizionata in moduli funzionali, ognuno dei quali copre un determinato ruolo al suo interno. Spesso il partizionamento in moduli ricopre la divisione funzionale delle varie attività in azienda (si consideri, ad esempio, a livello operativo le funzioni identificate dalla catena del valore di Porter).

La modularità è una caratteristica fondamentale in quanto migliora la manutenibilità dell'architettura. Soprattutto in sistemi aperti, ovvero strutturati per cooperare con altri sistemi in rete, in cui la interconnessione con altre imprese favorisce il cambiamento e la dinamicità delle organizzazioni, le architetture interne sono sottoposte spesso a cambiamenti. La modularità consente di cambiare moduli lasciandone inalterati altri e quindi consente di fare cambiamenti senza impattare sull'intera architettura, ma solo sul/sui modulo/i interessati. Inoltre, i sistemi informativi delle organizzazioni moderne sono talmente complessi da rendere quasi impossibile, o comunque poco conveniente, la loro realizzazione partendo da zero. È per questo che nella maggior parte dei casi che è meglio optare per moduli già esistenti presenti sul mercato (i componenti *COTS*) e realizzare solo alcune funzionalità. Con questo approccio davanti alla classica decisione Make-or-Buy si predilige, per lo meno per funzionalità standard, sempre più spesso Buy rispetto al Make.

Due delle applicazioni che in azienda vengono maggiormente acquisite con modalità “Buy” sono *ERP* (*Enterprise Resource Planning*) e *CRM* (*Customer Relationship Management*), descritte nelle prossime sezioni. Secondo quanto

descritto nel Capitolo 7, ERP e CRM sono componenti tecnologici a livello applicativo.

## 8.2 ERP

L'ERP (Enterprise Resource Planning) è una suite software che offre moduli a supporto del sistema operazionale e quindi delle transazioni interne all'organizzazione finalizzate alla gestione dell'impresa. Un sistema ERP offre funzionalità che coprono interamente la catena del valore di Porter.

### 8.2.1 Storia degli ERP

Per comprendere la nascita dei sistemi ERP bisogna risalire alla nascita dei sistemi CIM (Computer Integrated Manufacturing), diffusi nelle organizzazioni negli anni '80, che si focalizzavano sulle attività di produzione nelle aziende manifatturiere. Infatti, a partire dal monitoraggio delle attività, dall'integrazione e valutazione dei dati estratti dalle singole macchine, dei dati relativi all'organizzazione di celle, aree produttive e stabilimenti, e altro, il CIM ha l'obiettivo di fornire una visione integrata del sistema produttivo, per supportarne i diversi settori (progettazione, ingegnerizzazione, produzione, controllo della qualità, pianificazione della produzione e marketing). Questo principalmente per supportare decisioni relative all'ottimizzazione dell'efficienza dei processi.

A fianco dei sistemi CIM, negli anni '80 nascono anche i sistemi MRP (Materials Requirements Planning), con le seguenti funzionalità principali: (i) programmare il fabbisogno dei materiali automatizzando la distinta base e, partendo dagli obiettivi di produzione (ad esempio, il numero di prodotti da produrre), definire in modo automatico la lista di materie prime da acquistare, (ii) programmare le lavorazioni e (iii) controllare le giacenze. I sistemi MRP vengono presto inglobati dai sistemi di MRP II (Manufacturing Resource Planning). Si tratta di sistemi modulari che mirano a sostituire qualsiasi altro sistema settoriale per la gestione delle materie prime e delle lavorazioni. Infatti, gli MRP II si occupano della pianificazione e gestione di tutte le risorse e i processi aziendali presenti a livello operativo: i materiali, la produzione, il personale, la manutenzione, le risorse finanziarie e amministrative.

L'ERP nasce negli anni novanta per espandere ulteriormente i sistemi MRP II. La prima versione dell'ERP mette in collegamento diretto le aree di gestione contabile con l'area di gestione logistica (magazzini ed approvvigionamento). Successivamente, inizia l'implementazione delle relazioni interne anche con le altre aree del livello operazionale (per esempio, aree di vendita, distribuzione, produzione, manutenzione impianti, gestione dei progetti, ecc.).

Dai primi anni del 2000, i maggiori fornitori di soluzioni ERP iniziano a creare soluzioni informatiche verticali per i vari settori merceologici delle aziende; iniziano così a nascere specializzazioni degli applicativi per il settore automobilistico, per il settore delle vendite al dettaglio, per il settore logistico, per il settore meccanico e per il settore edile.

### 8.2.2 Proprietà degli ERP

Oltre alle funzionalità offerte dai primi sistemi informatici, bisogna ricordare che fino ai primi anni novanta i sistemi informativi nelle aziende seguono un ciclo di progettazione e di sviluppo detto *a silos*: le applicazioni sono isole autonome di supporto a una ben definita funzione aziendale. Tali isole contengono tutte le componenti software (base di dati, logica applicativa e presentazione) necessarie a svolgere il compito per cui sono state realizzate. Questo approccio funziona bene fino a che le applicazioni non devono interagire. Il bisogno di integrare o semplicemente far comunicare le applicazioni mette, infatti, in luce i punti deboli dell'approccio, quali: (i) il bisogno di un notevole sforzo informatico per gestire i flussi di informativi tra applicazioni, (ii) la ridondanza dei dati, che provoca disallineamento tra gli stessi favorendo errori e incongruenze, (iii) la difficoltà nell'avere una visione di insieme dei dati presenti nei diversi sottosistemi.

Questi problemi finiscono con il favorire la diffusione degli ERP, che si contraddistinguono per tre proprietà fondamentali:

- Unicità dell'informazione.
- Modularità.
- Prescrittività.

L'*unicità dell'informazione* si riferisce alla presenza di una unica rappresentazione logica dei dati valida per ogni modulo all'interno dell'ERP. Tutti i moduli vedono gli stessi valori: si risolvono in questo modo i problemi di ridondanza e incongruenza dei dati.

Per quanto riguarda la *modularità*, la suite è composta da moduli autosufficienti che possono essere acquisiti e integrati in modo incrementale. Ogni modulo è composto da funzioni che generalmente supportano attività elementari (per esempio, il ricevimento di materiali). Oltre a benefici sulla flessibilità e scalabilità dell'applicazione, la modularità abilita nuove strategie di acquisizione dei COTS. Infatti, è possibile comprare moduli da uno stesso fornitore (Strategia *One Stop Shopping*) o da fornitori diversi (Strategia *Best of Breed*). Nel primo caso, è garantita l'interoperabilità tra moduli. Nel secondo caso l'interoperabilità non è garantita, ma si ha l'opportunità di acquisire i moduli da fornitori diversi, caratterizzati da un elevato tasso di specializzazione, e quindi di migliore qualità, per svolgere le diverse funzionalità. Chiaramente, in questo ultimo caso è richiesto un notevole sforzo di integrazione.

La terza caratteristica degli ERP, la *prescrittività*, è legata al fatto che questi prodotti incorporano la logica di "funzionamento" dell'impresa, codificata anche nelle business rules. Ad esempio, si possono normare i processi di produzione e logistica, prescrivendo che un materiale non può entrare senza ordine o che un materiale può essere ordinato se e solo se è stato richiesto da un ente aziendale autorizzato. In questo modo è l'azienda che si deve adattare al software e non viceversa. Questo potrebbe rappresentare un vantaggio per le aziende che adottano l'ERP in quanto solitamente la logica incorporata è ottimale poiché viene progettata ispirandosi a best practice o a processi ben noti e ottimizzati.

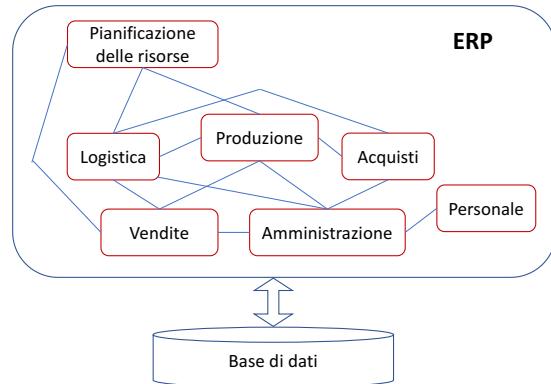


Figura 8.1: Organizzazione di un sistema ERP (rielaborazione da [30])

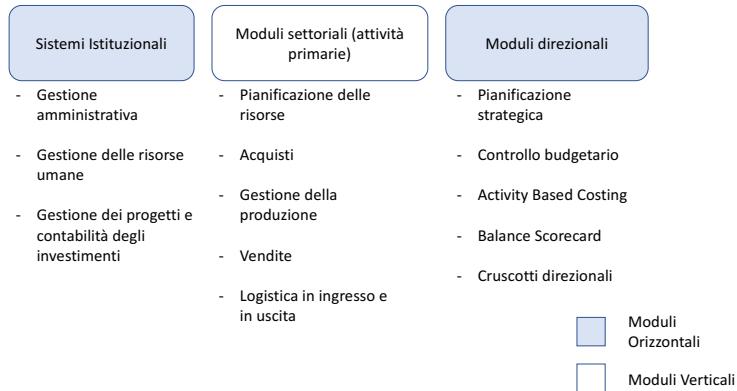


Figura 8.2: Suite ERP

### 8.2.3 Funzionalità di un ERP

La Figura 8.1 riporta una rappresentazione schematica dell'organizzazione di un sistema ERP. I moduli rappresentati sono alcuni tra le funzionalità principali implementate dalle best practice. I moduli si possono poi classificare in tre categorie: sistemi istituzionali, moduli settoriali (attività primarie) e moduli direzionali (si veda la Figura 8.2). Moduli istituzionali e moduli direzionali sono moduli orizzontali, ovvero intersettoriali. I moduli settoriali sono verticali, ovvero specifici di un determinato contesto aziendale. I moduli direzionali forniscono elaborazioni dei dati volte a dare una visione d'insieme delle attività svolte a livello operativo, quindi con elementi quali i cruscotti, la reportistica e analisi economiche o di business intelligence avanzate.

La Figura 8.2 rappresenta un ERP detto *core*, ovvero principale. Successivamente, l'ERP core è stato esteso per andare incontro a nuove esigenze soprattutto nell'interazione con i clienti e i fornitori. Sono stati definiti nuovi

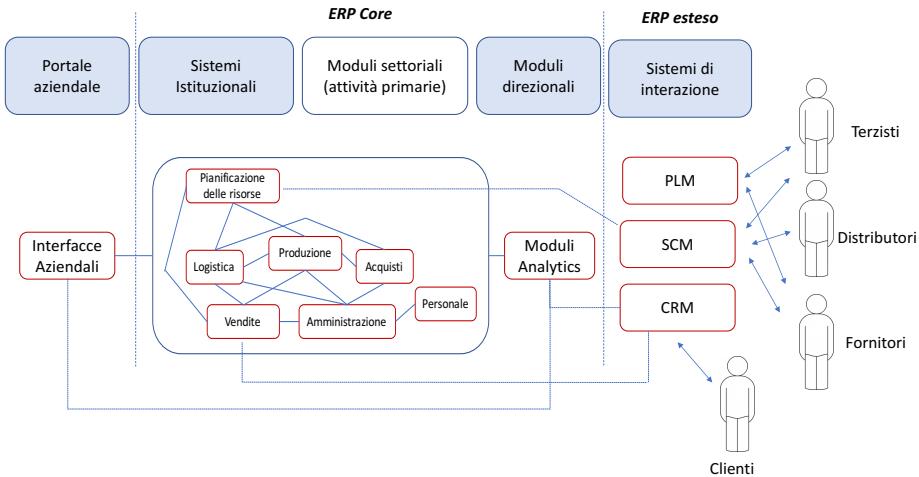


Figura 8.3: ERP esteso

applicativi, dando origine all'ERP *esteso*, come mostrato in Figura 8.3. Il PLM (Product Lifecycle Management) è un applicativo in grado di gestire le informazioni dei prodotti seguendoli nel loro intero ciclo di vita: dall'ideazione del prodotto alla sua realizzazione. SCM (Supply Chain Management) è un software per la gestione della supply chain che fornisce le informazioni a supporto delle decisioni di acquisto e scheduling della produzione. In particolare, queste informazioni aiutano le aziende a pianificare la loro supply chain (supply chain planning) e a eseguire i passi della supply chain (supply chain execution). Il CRM (Customer Relationship Management) supporta l'interazione con il cliente, dal contatto al post-vendita, con l'obiettivo di fornire servizi avanzati per fidelizzarlo e aumentarne la soddisfazione.

Si noti che questi moduli di estensione possono fare parte della suite ERP come modulo di estensione, ma potrebbero in alternativa anche essere acquisiti separatamente da altri fornitori, e quindi essere nell'architettura moduli esterni con cui il sistema ERP deve colloquiare e quindi essere integrato.

L'adozione di un ERP all'interno di un'azienda si può rappresentare in un'architettura BOAT Party-level come rappresentato in Figura 8.4. Si può osservare come l'ERP utilizzi una base di dati unica e offra le proprie funzionalità tramite delle interfacce che permettono agli utenti (interni o esterni all'azienda) di specificare le proprie richieste.

#### 8.2.4 Aspetti architetturali

Dal punto di vista architettonico, si possono individuare tre approcci principali dettati dall'evoluzione delle tecnologie nel tempo (si veda Figura 8.5). Negli anni '90 l'ERP era installato nelle organizzazioni utilizzando principalmente il paradigma client-server. In questo modo, tutti i moduli funzionali venivano

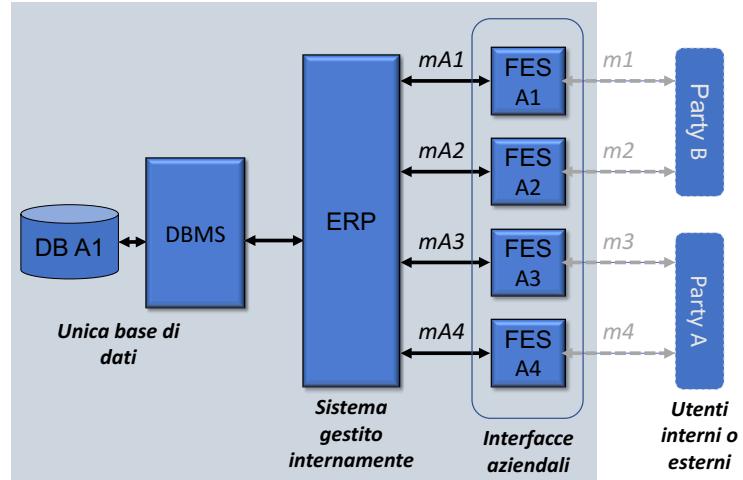


Figura 8.4: Rappresentazione ERP con architettura Party-level

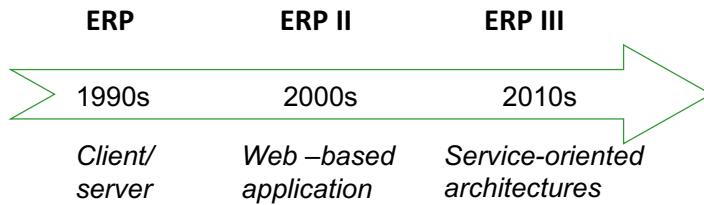


Figura 8.5: Evoluzione architetturale degli ERP

installati su uno o più server e gli utenti accedevano alle diverse funzionalità tramite client. Con la diffusione del Web, i vendor ERP hanno iniziato ad offrire *Web-based ERP*: ERP sviluppati come applicazioni Web e quindi fruibili attraverso l'utilizzo di un browser. Si passa in questo modo da architetture a due livelli ad architetture a tre o più livelli. Questa innovazione viene riconosciuta come il fattore che ha permesso alle aziende di accedere al proprio ERP anche in remoto. Questo offre la possibilità agli utenti di accedere alle funzionalità del sistema ERP anche fuori dall'organizzazione. Altra innovazione degli ERP si ha in concomitanza con la diffusione del Cloud Computing: molti vendor ERP offrono la possibilità di acquisire i moduli in modalità SaaS (Software as a Service). Questi dettagli architetturali verranno descritti e approfonditi nei prossimi capitoli.

In ogni caso, l'evoluzione architetturale degli ERP ha portato una diminuzione dei costi legati alla sua acquisizione e gestione. Infatti, gli ERP Web-based hanno offerto la possibilità di esternalizzare la gestione del sistema informativo

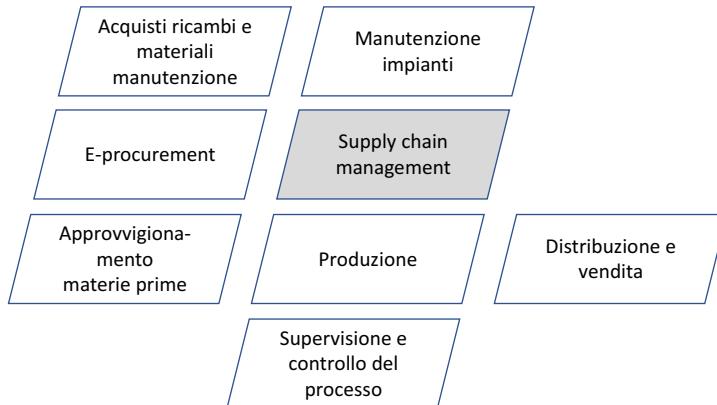


Figura 8.6: Moduli principali offerti per aziende chimiche e siderurgiche (rielaborazione da [7])

abbassandone sensibilmente i costi. Tali costi sono stati ridotti ulteriormente dai SaaS ERP in quanto (i) l'adozione della logica a servizi per lo sviluppo degli ERP riduce i costi per i vendor e di conseguenza anche i prezzi di vendita; (ii) la tariffazione è fatta a consumo e di solito il costo totale risulta minore del tradizionale costo di acquisizione (costo di licenza e implementazione).

### 8.2.5 Implementazione di un ERP

Come detto all'inizio gli ERP sono applicativi COTS. In quanto tali, non vengono progettati e realizzati per una singola impresa, ma per un insieme di aziende appartenenti a un determinato settore merceologico (o dominio applicativo) o accomunate da alcune caratteristiche. Ci saranno quindi in commercio suite diverse per diversi settori. Le suite pensate per aziende manifatturiere offrono moduli che ricalcano le funzionalità della catena del valore di Porter. Però bisogna considerare che ad esempio le industrie chimiche e siderurgiche seguono un modello leggermente diverso e pertanto anche la suite software sviluppata sarà composta da moduli diversi. In Figura 8.6 si ha una segmentazione possibile dei moduli ERP per questo tipo di aziende. Si noti che il Supply Chain Management non è incluso nell'ERP in quanto è di solito una sua estensione. La Figura 8.7 illustra invece una suite software per le aziende telefoniche e di servizi energetici. Un ERP supporta quasi tutte le attività operative: solo la pianificazione e progettazione della rete sono escluse.

In ogni caso, anche se gli ERP sono verticalizzati su un settore specifico, le aziende all'interno del settore hanno spesso esigenze diverse. Per questo motivo, una volta installati non sono immediatamente utilizzabili da un'azienda ma devono essere configurati in modo corretto per soddisfare il più possibile i requisiti aziendali (vedere processo di implementazione in Figura 8.8). La prima opera-

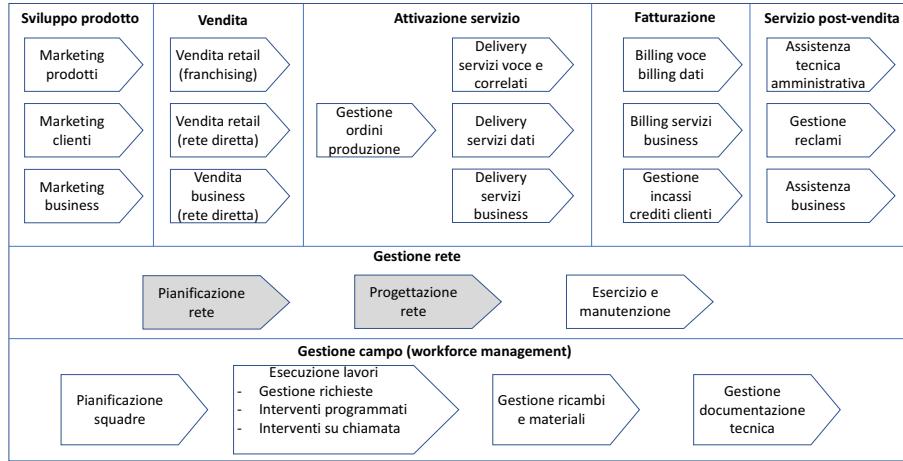


Figura 8.7: Moduli principali offerti per aziende telefoniche e di servizi energetici (rielaborazione da [7])

zione svolta in questa direzione è la parametrizzazione: attraverso il settaggio di alcuni parametri si possono attivare e disattivare delle funzionalità o selezionare una specifica versione di una funzione. In questo modo si costruisce una istanza dell'applicazione. Se l'azienda vuole apportare ulteriori cambiamenti al prodotto deve procedere con una vera e propria personalizzazione che consiste in apportare cambiamenti al codice. Questa fase può essere assimilata a un progetto software volto ad apportare modifiche ad un programma esistente (vedi anche la Sezione 2.4).

### 8.2.6 Benefici degli ERP

L'adozione di un ERP ha effetti sui processi gestionali, ma anche sul modo con cui l'azienda si pone sul mercato. In altre parole, questo intervento porta vantaggi in termini di efficienza e di efficacia. Per quanto riguarda i vantaggi di efficienza, essi sono correlati al risparmio di tempi e costi sulle attività operative ottenuti tramite l'automazione e l'integrazione di procedure. I vantaggi in termini di efficacia sono invece legati all'unicità dei dati, alla capacità di elaborare e veicolare le informazioni e alla standardizzazione delle piattaforme IT: esiste infatti la possibilità di avere una maggior disponibilità, accessibilità e tempestività dei dati. Questo consente di avere i dati necessari per supportare le decisioni a livello tattico e strategico.

## 8.3 CRM

Il CRM (Customer Relationship Management) è una suite software che supporta le organizzazioni nelle interazioni con i clienti. Il CRM pone al centro della

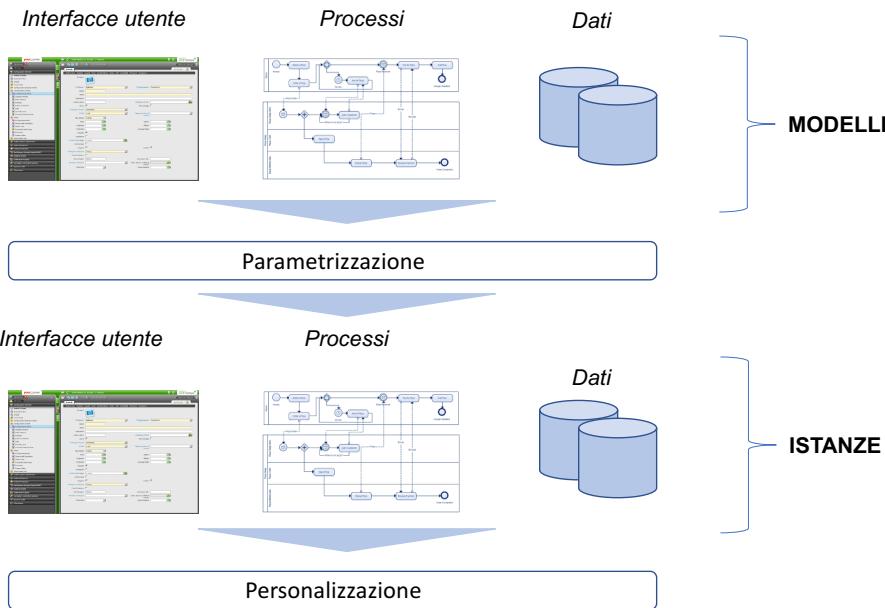


Figura 8.8: Implementazione di una suite ERP

strategia aziendale il cliente. Il CRM viene di solito utilizzato per capire meglio i bisogni e i comportamenti dei clienti e per migliorare l'utilizzo dei canali di interazione con gli stessi. In particolare, il CRM ha l'obiettivo di fidelizzare i clienti costruendo relazioni personalizzate di lungo periodo capaci di (i) aumentare la soddisfazione dei clienti e (ii) aumentare il valore dell'impresa per il cliente e del cliente per l'impresa.

Il CRM viene spesso inquadrato come estensione dell'ERP in quanto fornisce funzionalità aggiuntive per la gestione del cliente a partire dal contatto con lo stesso fino ad arrivare ai servizi post-vendita. Al contrario dell'ERP però solo alcune aziende possono trarre benefici dalla sua implementazione. Infatti, le opportunità create dal CRM sono maggiori in settori in cui si ha una maggiore intensità della relazione con la clientela (frequenza della interazione, durata del rapporto contrattuale), si ha una certa numerosità della clientela, si può instaurare un rapporto di lealtà con il cliente e si ha una multicanalità del rapporto. Quest'ultima si riferisce al fatto che un cliente può accedere ai servizi offerti da una organizzazione utilizzando canali diversi (per esempio, canale fisico, Web, telefono). In breve, si può affermare che i sistemi CRM sono forniti con maggiori funzioni rispetto ai sistemi di logistica commerciale delle suite ERP. I punti chiave del CRM sono il supporto alla relazione con il cliente attraverso molteplici canali (multicanalità) finalizzato al contatto in entrata (inbound) e in uscita (outbound), il supporto post vendita e gestione della lealtà (service) e l'analisi del comportamento del cliente.

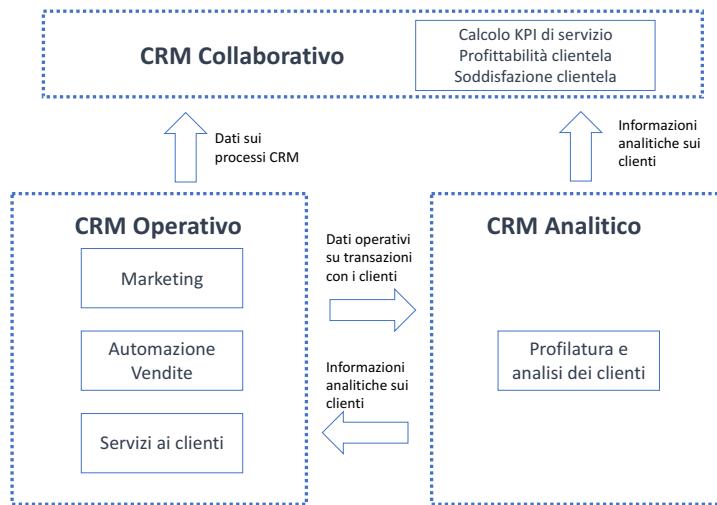


Figura 8.9: Componenti di un Enterprise CRM

È importante capire che il CRM non è solo un'applicazione, ma una vera e propria strategia che l'organizzazione deve perseguire a livello aziendale. Nonostante il CRM sia composto da diversi moduli software e da varie tecnologie, esso è da considerare come sistema a supporto di un processo dedicato al miglioramento degli obiettivi aziendali. Implementare un CRM aiuta a caratterizzare meglio i clienti e a progettare campagne di marketing personalizzate cercando di aumentare i ritorni.

I due componenti principali del CRM sono il CRM operativo e il CRM analitico. Il *CRM operativo* (o operazionale) è il modulo di front end che supporta le interazioni giornaliere con i clienti mentre il *CRM analitico* è inquadrato come modulo di back end che si occupa principalmente dell'analisi dei dati dei clienti. La Figura 8.9 illustra i componenti del CRM e le comunicazioni tra moduli. In figura si vede che il CRM include anche un terzo modulo, il *CRM collaborativo* che è preposto al calcolo di alcuni indicatori che verranno condivisi con diversi attori all'interno dell'azienda (per esempio, manager vendite e marketing, responsabile agenti di vendita).

La Figura 8.10 raffina la figura precedente focalizzandosi solo sull'interazione tra CRM operativo e CRM analitico e mettendo soprattutto in luce i componenti necessari allo scambio e alla gestione dei dati. Il CRM operativo registra tutte le interazioni tra il cliente e l'azienda sui vari canali in una unica base di dati. Il CRM analitico per compiere tutte le analisi si basa su un data warehouse costruito a partire dalle informazioni estratte dalla base di dati del CRM operativo. Il funzionamento del data warehouse verrà illustrato nella Sezione 8.4. La figura evidenzia anche il fatto che i dati del CRM operativo vanno anche verso i sistemi di back end. Infatti, molte volte le interazioni con il cliente generano un evento che poi deve essere gestito dai sistemi di back end: ad esempio una

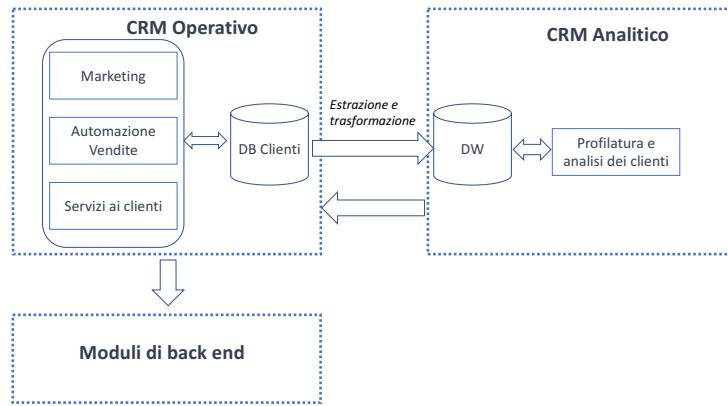


Figura 8.10: Architettura di massima per la gestione dei dati in un CRM

| Marketing                 | Vendite              | Servizi ai clienti     |
|---------------------------|----------------------|------------------------|
| Generazione liste clienti | Gestione vendite     | Contact center         |
| Gestione campagne         | Gestione contatti    | Web based self service |
| Cross-selling e Upselling | Gestione opportunità | Call scripting         |

Figura 8.11: Componenti del CRM operativo (rielaborazione da [4])

richiesta di vendita deve essere poi gestita da un ERP. Nelle prossime sezioni illustriamo la composizione dei moduli di CRM presentata in Figura 8.9.

### 8.3.1 CRM operativo

Il CRM operativo si occupa dell'interazione dei clienti con l'azienda e in particolare, come descritto in Figura 8.9 è costituito da diversi componenti per supportare le attività di vendita, di marketing e di supporto al cliente. Detti componenti si avvalgono di altri moduli come descritto in Figura 8.11.

**Marketing** La strategia delle aziende non si focalizza più sull'estendere la clientela e vendere i propri prodotti a più clienti possibili, ma sul fidelizzare il più possibile i clienti esistenti e quindi vendere più prodotti a clienti già acquisiti. Questo è realizzabile solo tramite una buona progettazione di campagne di marketing orientate al singolo cliente o a piccoli gruppi di clienti. È da sottolineare che il successo di una campagna di marketing è di solito proporzionale alla capacità dell'azienda di acquisire e analizzare dati. I sistemi che un dipartimento di marketing può pensare di implementare per aumentare la soddisfazione dei clienti sono:

- *Generazione liste clienti:* produce liste di clienti adatti a ricevere una comunicazione di marketing. Dette liste sono compilate tramite una seg-

mentazione dei clienti fatta considerando dati relativi alle interazioni del cliente con l'azienda (es. visite al sito, questionari, chiamate al call center). La segmentazione viene fornita dal CRM analitico.

- *Gestione campagne*: questo modulo è anche detto Marketing Automation e si riferisce a moduli progettati per automatizzare attività e processi di marketing. Nel dettaglio, guida gli utenti nella definizione, pianificazione, programmazione e analisi dei risultati delle campagne di marketing. I benefici che l'azienda può trarre dall'implementazione di questo modulo sono notevoli: (i) miglioramento dell'efficienza e riduzione degli errori, (ii) miglioramento del tasso di conversione dei contatti (la percentuale dei contatti che viene trasformata in clientela), (iii) aumento della produttività ed efficacia di marketing, (iv) consolidamento delle relazioni con la clientela.
- *Cross-selling e Upselling*: con il termine cross-selling ci si riferisce al vendere un maggior numero prodotti o servizi a un cliente, mentre con il termine upselling si indica la strategia con cui si cerca di aumentare il valore delle vendite a un cliente. Per esempio, un ristorante fa cross-selling se cerca di vendere un piatto in più al proprio cliente mentre fa upselling se cerca di vendere un piatto più costoso invece di quello scelto. Questo modulo aiuta il dipartimento di marketing ad attuare queste strategie offrendo tutti i dati sui clienti ed analisi appropriate.

**Automazione Vendite** Il reparto vendite è stato il primo ad utilizzare il CRM con uno dei primi componenti ad essere sviluppati: il *Sales Force Automation (SFA)*. SFA è un modulo che viene utilizzato dagli agenti di vendita e li supporta lungo tutto il processo di vendita: identificazione del cliente, gestione del contatto, definizione dell'offerta al cliente e registrazione dell'acquisto. Ora i moduli per il supporto alle vendite riescono a gestire tutti i canali e si dividono in:

- *Gestione vendite*: il modulo di gestione vendite automatizza o offre un supporto a tutte le fasi di un processo di vendita illustrato in Figura 8.12. In particolare, il modulo si occupa della selezione dei contatti, supporta l'agente di vendita nella stesura dell'offerta e del contratto. Inoltre, registrando tutte le operazioni svolte dagli agenti, consente di analizzare l'operato dell'agente. Sul canale Web fornisce tutte le funzionalità per fare un acquisto on line: gestione del catalogo, gestione del carrello, conclusione dell'ordine e pagamento.
- *Gestione contatti*: si occupa della gestione dati dei clienti e identifica eventuali clienti per le vendite future. Il modulo di gestione contatti tiene anche in considerazione informazioni aggiuntive che possono contribuire a migliorare il rapporto con il cliente. Ad esempio, predisporre il call center con funzioni per identificare il chiamante e recuperare note su interazioni precedenti può essere utile per accogliere il cliente salutandolo per nome

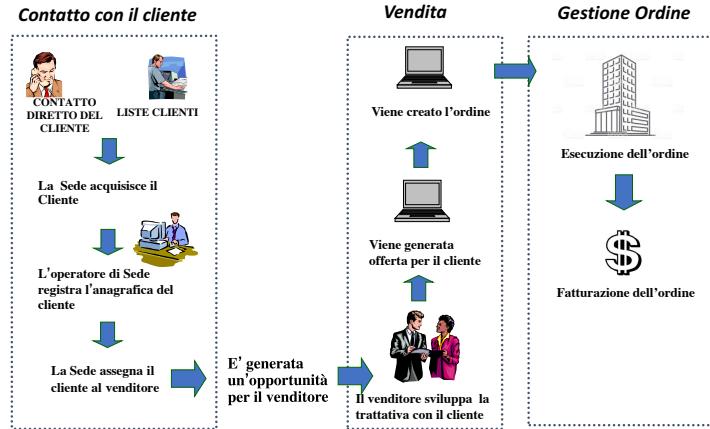


Figura 8.12: Passi principali di un processo di vendita

e personalizzare la conversazione sulla base delle comunicazioni già avvenute in modo tale da far sentire importante il cliente e da fidelizzarlo ulteriormente.

- *Gestione opportunità*: questo modulo ha l'obiettivo di trovare nuovi clienti o organizzazioni per vendite future. Le sue funzionalità sono altresì preposte a calcolare la probabilità di una vendita associata a un cliente facendo risparmiare tempo ad agenti e ufficio marketing sull'analisi e il contatto con clienti non profitevoli. La principale differenza tra il modulo di gestione contatti e quello di gestione opportunità sta nel fatto che il primo si occupa di persone ed organizzazioni che sono già clienti mentre il secondo si focalizza su contatti nuovi.

**Servizi ai clienti** Vendite e marketing sono le unità aziendali che interagiscono maggiormente con i clienti prima di una vendita. Per questo, i moduli di supporto a queste attività sono considerati tra i più importanti del CRM. Purtroppo, molte organizzazioni non considerano l'importanza di continuare a curare le relazioni con i clienti anche dopo che la vendita è stata effettuata. Curare le relazioni nella parte post-vendita risulta spesso ancor più importante per consolidare i rapporti con i clienti e fidelizzarli. Per questo motivo, il CRM offre anche un modulo di Servizi ai clienti (Customer service) che si compone dei seguenti sottomoduli:

- *Contact center*: i contact center (o call center) sono responsabili della gestione delle chiamate inbound (i clienti chiamano l'azienda) e delle chiamate outbound (l'azienda chiama i clienti). Il CRM registra tutte le interazioni del cliente con il call center e fornisce funzionalità per il riconoscimento del cliente e per la gestione della chiamata. In particolare, funzionalità

comuni adottate dai call center sono (i) la distribuzione automatica delle chiamate, (ii) IVR (Interactive Voice Response) con cui gli utenti prelevano o forniscono informazioni attraverso l'uso della tastiera telefonica, (iii) composizione automatica predittiva che automaticamente chiama i clienti e seleziona il giusto operatore per la gestione della telefonata.

- *Web based self service*: permette ai clienti di usare il Web per trovare soluzione ai loro problemi o per contattare l'azienda (per esempio, form o chat on line).
- *Call scripting*: questo modulo gestisce la conoscenza fornendo agli operatori la possibilità di accedere a una base di dati contenente le soluzioni per risolvere i problemi sollevati dai clienti. Il sistema può anche offrire una lista di domande che l'operatore può porre al cliente per determinare i veri problemi e le soluzioni.

### 8.3.2 CRM analitico

Il CRM operativo registra tutte le interazioni con il cliente in un'unica base di dati. Il CRM analitico usa questi dati per analizzare le preferenze e il comportamento dei clienti per estrarne pattern significativi. Queste analisi sono rese possibili dall'utilizzo di data warehouse e sistemi di data mining che verranno spiegati nel dettaglio nelle Sezioni 8.4 e 8.5. Questi sistemi aggregano, analizzano e diffondono velocemente informazioni all'interno dell'organizzazione per supportare al meglio il processo decisionale. Il CRM analitico ha tre finalità principali: (i) *Reporting*, che aiuta a capire chi sono i clienti, le loro caratteristiche e preferenze, (ii) *Analysis*, che aiuta a segmentare i clienti in categorie (per esempio, clienti attivi, dormienti e inattivi), (iii) *Predicting*, che aiuta l'organizzazione a predire le azioni e i desideri dei clienti. In questo modo l'azienda può identificare nuove opportunità e offrire un servizio di qualità superiore da vari punti di vista (efficienza, attenzione ai requisiti, tempestività di risposta, ecc.).

Il CRM analitico invia i risultati delle analisi al CRM operativo. Infatti, per esempio supporta il modulo di Marketing fornendo la segmentazione della clientela e identificando i clienti potenzialmente interessati alla specifica campagna di marketing. Supporta poi le vendite suggerendo ai clienti prodotti o servizi a cui potrebbero essere interessati sulla base delle loro preferenze e del loro storico di acquisti.

### 8.3.3 CRM collaborativo

Il CRM collaborativo si pone l'obiettivo di calcolare alcuni indici rilevanti per l'intera azienda, quali la profitabilità e la soddisfazione dei clienti e di condividere tali informazioni e altri dati estratti dal CRM analitico con alcuni attori all'interno dell'organizzazione. Tali indicatori mirano a valutare l'impatto che l'adozione del CRM ha avuto sull'organizzazione (per esempio, maggiori

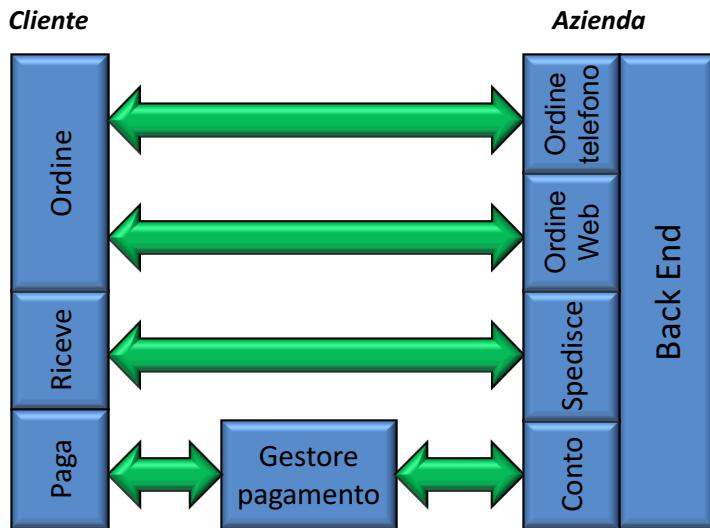


Figura 8.13: Esempio di vendita multicanale

ritorni) e per questo possono essere anche utilizzati per controllare se l'investimento ha portato ai risultati sperati che erano stati individuati come driver dell'investimento stesso.

### 8.3.4 Implementazione del CRM

Per il CRM valgono le stesse considerazioni fatte per l'ERP: è una suite che può essere acquisita in modalità Make o Buy (in quanto disponibili come COTS) e possono essere gestiti On Premises o in Outsourcing (Application Service Provider o SaaS). I benefici principali derivanti dall'adozione del CRM sono: incremento della soddisfazione del cliente, miglioramento dell'efficienza del call center, supporto all'acquisizione di nuovi clienti, semplificazione dei processi di marketing e di vendita e miglioramento dell'efficienza degli agenti di vendita.

Ipotizzando di avere un'azienda che vende prodotti via telefono o via Web come rappresentato in Figura 8.13, la presenza di un CRM può essere messa in luce in un diagramma Party-level come rappresentato in Figura 8.14.

## 8.4 Data Warehousing

I dati sono una risorsa importante non solo a livello operativo, ma anche a livello tattico e strategico per supportare i processi decisionali all'interno di un'organizzazione. Nasce quindi la necessità di accedere a dati aggregati in tempo reale e di eseguire interrogazioni complesse, come l'analisi della correlazione tra diverse variabili (ad esempio, la relazione esistente tra il luogo di residenza di un cliente e la sua predisposizione ad usufruire di una certa categoria di servizi).

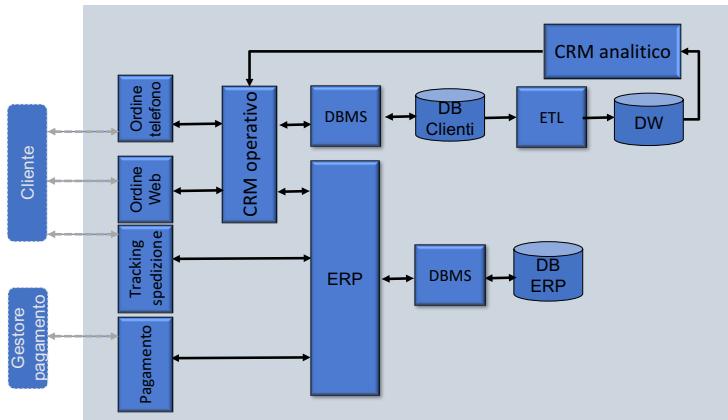


Figura 8.14: Esempio di diagramma Party-level in presenza di CRM

Quando ci si sposta dal livello operazionale ai livelli più alti della piramide di Anthony, cambiano sia la tipologia di dati di interesse sia il modo in cui questi vengono usati. Perché le attività tattiche e strategiche abbiano successo, è necessario fornire strumenti in grado di estrarre le informazioni di interesse da diverse fonti (interne ed esterne) e in modo veloce ed efficace. Tali strumenti rientrano all'interno della cosiddetta *Business Intelligence (BI)*.

I primi strumenti di supporto alle decisioni sono stati i report. I report contengono dati analitici, ottenuti dall'analisi della base di dati operazionale, che vengono prodotti seguendo un formato prefissato a scadenze regolari. Questo strumento soffre di diverse limitazioni, in quanto i dati contenuti sono dati statici con cui l'utente non può interagire per ottenere dettagli o ulteriori informazioni su alcuni aspetti non previsti. Inoltre, la produzione dei report può risultare complessa e richiedere un tempo non trascurabile, rendendo impossibile accedere ad informazioni aggiornate. Infine, i report danno una visione limitata degli aspetti di interesse per le decisioni strategiche in quanto si basano solo sui dati interni dell'organizzazione che vengono estratti e aggregati dalla base di dati operazionale.

Uno strumento più evoluto e per vari aspetti più dinamico è costituito dai fogli di calcolo (es. Excel). In questo caso, le analisi possono essere definite direttamente dall'utente finale che può cambiarle a suo piacimento ed eseguirle nel momento in cui le informazioni sono necessarie. Nonostante questi vantaggi rispetto ai report, i fogli di calcolo risultano comunque limitativi a causa dei procedimenti complessi necessari per esportare i dati dalla base di dati ed importarli nel foglio elettronico. Inoltre, usando un tale strumento, ogni utente può creare uno strumento personalizzato che, nonostante i vantaggi, non ha il beneficio di un controllo di integrità e di correttezza delle analisi che è invece tipico di strumenti condivisi.

Uno strumento in grado di supportare le attività ai livelli più alti della

piramide di Anthony è invece una base di dati con caratteristiche diverse dalle classiche basi di dati operazionali:

- La base di dati può comprendere dati delle basi di dati operazionali e dati di fonti esterne.
- I dati devono essere organizzati secondo una struttura che faciliti le analisi.
- La struttura della base di dati deve essere semplice e mettere in relazione tutti e solo i dati di interesse.
- Le fonti di dati devono essere integrate e aggiornate.
- Devono essere disponibili strumenti di analisi caratterizzati da brevi tempi di risposta.

Per rispondere a queste esigenze nasce il concetto di *data warehouse* [29].

#### 8.4.1 Caratteristiche dei dati

Un data warehouse è un sistema di tipo OLAP. Questo lo distingue dai DBMS tradizionali che invece si poggiano su tecnologie di tipo OLTP.

I dati analitici usati a supporto delle attività strategiche e decisionali si differenziano dai dati operazionali per diverse caratteristiche:

- Obiettivo: i dati devono essere utilizzati per prendere decisioni sugli sviluppi futuri dell'organizzazione, ma anche per identificare problematiche organizzative o possibilità di crescita per l'organizzazione.
- Utenti: gli utilizzatori dei dati sono i manager di alto e basso 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: contrariamente ai dati operazionali, gli utenti dei dati analitici accedono all'informazione solo in lettura, e non in scrittura.

Date queste caratteristiche, il classico modello relazionale non è soddisfacente per realizzare una base di dati analitica.

Per rappresentare i dati di un sistema OLAP si ricorre al *modello multidimensionale*. In tale modello l'informazione è rappresentata tramite il concetto di *ipercubo*, formato da un numero variabile di *dimensioni* N, in cui ogni dimensione è una dimensione di analisi per la base di dati. Ogni elemento è un

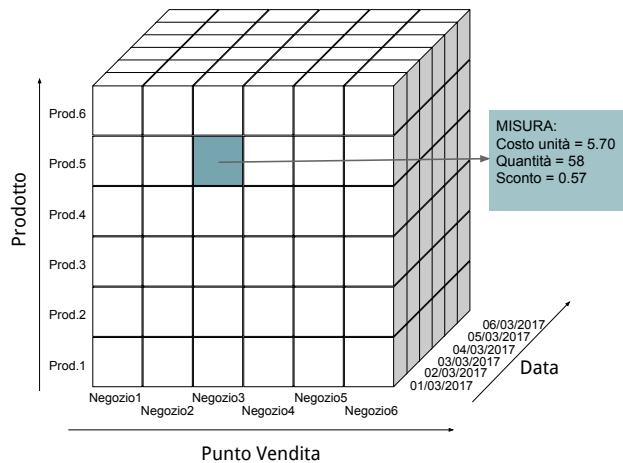


Figura 8.15: Ipercubo a tre dimensioni che rappresenta gli aspetti rilevanti per l'analisi delle vendite di una catena di supermercati

oggetto della base di dati, cioè la registrazione di un'informazione, il cui valore è definito dalle coordinate rappresentate dalle dimensioni di analisi. Gli elementi costitutivi di una base di dati multidimensionale sono:

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

Un esempio di ipercubo tridimensionale per l'analisi delle vendite di una catena di supermercati è mostrato in Figura 8.15, in cui le dimensioni di analisi sono il tempo, il prodotto e il punto vendita, mentre il fatto è costituito dalla vendita e ha come misure la quantità venduta, il costo unitario e lo sconto applicato. Ogni dimensione rappresenta una delle coordinate del fatto e ha un dominio di valori discreti che permettono di definire l'insieme dei fatti di interesse. Nel caso in cui i valori di una dimensione non siano discreti, questi devono essere resi tali applicando ad esempio delle soglie per definire gli intervalli di valori considerati. Le dimensioni possono essere numerose e organizzate in gerarchie, dove il livello di granularità varia da livello a livello. Le gerarchie delle dimensioni sono organizzate in modo che esista una dipendenza funzionale tra un livello a più alta granularità e un livello a più bassa granularità, tale che tra di essi esista una relazione del tipo 1...N. Nell'esempio di Figura 8.16, i prodotti sono suddivisi in più categorie. Ogni categoria comprende uno o più tipi di prodotti e ogni tipo comprende più prodotti specifici. Lungo una linea gerarchica le relazioni determinano aggregazioni diverse per la dimensione considerata.

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

Figura 8.16: Esempio di gerarchia per la dimensione prodotto. Il livello di granularità cresce da destra verso sinistra.

Ritornando quindi all'ipercubo di Figura 8.15, la dimensione prodotto può essere manipolata cambiando il livello di granularità, ad esempio considerando non i singoli prodotti ma raggruppandoli per categoria (o per tipo). In questo caso, il contenuto del fatto sarà l'aggregazione delle informazioni per tutti i prodotti appartenenti alla categoria considerata. Simili operazioni sulle altre dimensioni, le loro granularità o i loro valori permettono di selezionare il livello di dettaglio di interesse per l'analisi.

#### 8.4.2 Data warehouse

Il data warehouse è una base di dati che raccoglie dati sintetici, integrati e strutturati di interesse per un'organizzazione. In senso più esteso, si può definire invece il data warehousing come l'insieme di tutte le operazioni atte alla popolazione della base di dati, al suo mantenimento, e tutte le tecniche di interrogazione che permettono di estrarre l'informazione dal data base direzionale. Il data warehouse è un sistema di tipo OLAP (On Line Analytical Processing). Il paradigma OLAP è caratterizzato da poche transazioni e interrogazioni (query) complesse che richiedono aggregazione di grandi quantità di dati. I sistemi OLAP hanno come misura di efficienza e di efficacia il tempo di risposta. Un sistema OLAP memorizza dati in formato aggregato, memorizza dati storici, archiviati secondo schemi multi-dimensional (generalmente schemi detti a stella). Spesso, le interrogazioni accedono a grandi quantità di dati per rispondere a quesiti complessi come, per esempio, quale è stato il profitto netto realizzato dall'azienda in una certa area geografica nello scorso anno.

I sistemi OLAP sono utilizzati per l'elaborazione di dati orientata al supporto decisionale, quindi sono adeguati a funzionalità collocate a livello di pianificazione e strategico della piramide di Anthony. Essi permettono infatti 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.

Un sistema OLAP gode invece delle proprietà FASMI (Fast, Analytical, Shared, Multidimensional, Informational). Deve quindi avere una visione multidimensionale dei dati (*multidimensional*) che devono contenere tutte le informazioni di interesse (*informational*) su cui permette di effettuare analisi complesse

(*analytical*) a più utenti ma con diversi permessi di accesso ai dati (*shared*) rispondendo alle loro richieste in un tempo ridotto (*fast*).

Inoltre, il data warehouse deve avere le seguenti caratteristiche:

- **Orientato alle entità (o oggetti):** considera le principali entità di analisi. Per esempio le vendite, gli ordini, i prodotti possono essere delle entità di analisi.
- **Integrato:** i dati considerati vengono prelevati sia da fonti interne che da fonti esterne all'azienda. Partendo da dati disomogenei essi devono essere resi consistenti e integrati in modo da avere una visione completa e chiara delle entità analizzate.
- **Variabile nel tempo:** Il data warehouse memorizza dati storici. Al fine di analizzare la storicità degli eventi tutti i dati sono associati a un'etichetta temporale che ne identifica il periodo di riferimento.
- **Persistente:** una volta inseriti in un data warehouse, i dati non vengono solitamente modificati. I dati sono archiviati in modalità di sola lettura.

#### 8.4.3 Architettura del data warehouse

Il data warehousing è l'insieme di attività che portano alla definizione, costruzione e mantenimento delle informazioni nel data warehouse, che può essere definito come la base di dati che raccoglie tutti i dati di interesse per l'azienda, sintetizzandoli, integrandoli e strutturandoli. All'interno del data warehouse (DW) è possibile individuare diverse basi di dati organizzate in modo gerarchico. Al primo livello abbiamo le *sorgenti*, cioè le fonti da cui i dati che popoleranno il data warehouse devono essere estratti. Queste sorgenti comprendono sicuramente la base di dati operazionale dell'organizzazione, ma anche altre basi di dati o sorgenti esterne all'organizzazione stessa che possono essere strutturate secondo varie forme (basi di dati, report, big data, ecc.). Tutte queste sorgenti vengono sottoposte ad un insieme di operazioni dette *ETL* (Extraction, Transformation e Loading) che permettono di estrarre dalle sorgenti i dati di interesse, di trasformarli secondo la struttura multidimensionale desiderata, e caricarli all'interno del data warehouse. Questa fase di popolamento può far uso di una base di dati intermedia tra le sorgenti e il data warehouse vero e proprio, chiamato *staging area*. Il data warehouse è invece la base di dati multidimensionale contenente tutte le informazioni di interesse accessibili mediante la valorizzazione delle dimensioni di analisi. In base alla presenza o meno della staging area, l'architettura del DW può essere a due (senza staging area) o tre (con staging area) livelli. A valle del data warehouse ci può essere un altro tipo di base di dati chiamata *data mart*. I data mart sono piccoli data warehouse tematici contenente un estratto dell'informazione contenuta nel data warehouse completo. Quest'ultimo può infatti avere dimensioni molto elevate che difficilmente sono di interesse per tutti gli utenti del sistema. Utilizzando i data mart è possibile definire viste del data warehouse che contengono un sotto-insieme delle

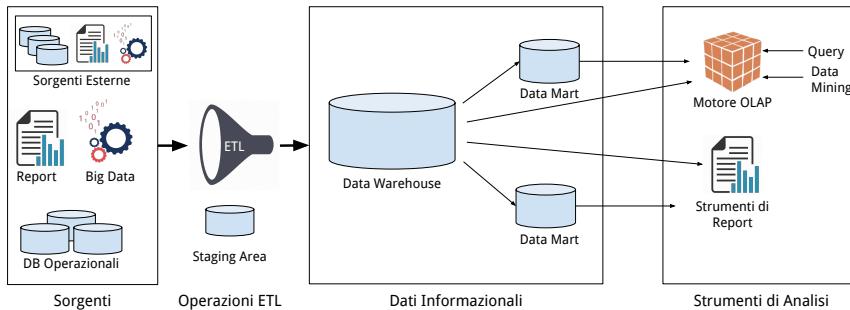


Figura 8.17: Architettura del data warehouse a tre livelli

informazioni. Questo sotto-insieme può essere una riduzione delle dimensioni di analisi (si prendono in esame solo quelle di interesse per la categoria di utenti a cui il data mart è indirizzato), del livello di granularità delle dimensioni, oppure dell'orizzonte temporale selezionato. Un'architettura che fa uso del solo DW è un'architettura centralizzata, in cui tutti i dati sono contenuti nella sola base di dati multidimensionale centralizzata. Al contrario, l'uso dei data mart permette l'uso di un'architettura distribuita in cui le diverse basi di dati tematiche possono essere posizionate su nodi diversi.

Per fare un esempio, possiamo riferirci ad una catena di supermercati che utilizza il data warehousing per migliorare la propria azienda. Nell'esempio le sorgenti di informazioni possono essere varie. Prima di tutto, ci sono le basi di dati operazionali, nel caso specifico la base di dati contenente i dati delle vendite e quella delle scorte di magazzino. Possono inoltre essere di interesse per le analisi anche le informazioni relative alle campagne pubblicitarie condotte dalla catena. Altre sorgenti che possono influenzare le decisioni possono invece provenire dall'esterno come i dati meteorologici (la vendita di alcuni prodotti potrebbe essere influenzata anche dalle condizioni atmosferiche), e dati provenienti da agenzie specializzate nell'analisi delle vendite, come Nielsen. Queste informazioni vengono trasformate e ristrutturate utilizzando le trasformazioni ETL per essere poi immagazzinate nel data warehouse aziendale. Dalla interazione con il data warehouse possono essere prese decisioni che sono influenzate da tutte le sorgenti analizzate, come ad esempio la pianificazione dei rifornimenti. Altri settori, come la gestione dei clienti o la pianificazione logistica, usano invece un sottoinsieme dei dati contenuti nel data warehouse e per questo motivo si può pensare alla creazione di data mart tematici che possono essere interrogati tramite un motore OLAP e da cui è possibile estrarre report in modo automatizzato.

### **ETL**

Particolare attenzione va dedicata agli strumenti a supporto del processo ETL. Questi, come detto in precedenza, permettono di estrarre, trasformare e caricare

i dati del data warehouse. Per quanto riguarda l'*estrazione*, essa definisce quali dati devono essere estratti e come devono essere trattati (ad esempio, se devono essere aggregati alla fonte o estratti al massimo livello di dettaglio). Inoltre, l'estrazione può essere: (a) *Statica* se vengono considerati tutti i dati presenti nelle sorgenti o (b) *Incrementale* se vengono presi in considerazione solo i dati prodotti o modificati dalle sorgenti nell'intervallo di tempo intercorso dall'ultimo aggiornamento del data warehouse. A seguito dell'estrazione, i dati necessitano di alcune *trasformazioni* in quanto derivando da fonti eterogenee possono essere rappresentati in formati diversi e di conseguenza potrebbero risultare non facilmente integrabili. Le principali operazioni che vengono eseguite sono:

- *Pulizia dei dati* (data cleaning): I dati estratti dalle sorgenti molte volte contengono errori che devono essere corretti prima del loro inserimento nel data warehouse. Questa operazione viene solitamente fatta per prima nella fase di trasformazione, ma in alcuni casi può essere fatta anche dopo o durante. I principali problemi relativi alla qualità dei dati che vengono risolti in questa fase sono la mancanza di dati, la presenza di valori inammissibili e l'inconsistenza tra valori presenti in campi diversi ma che sono legati tra loro da particolari regole o che hanno lo stesso significato.
- *Riconciliazione*: l'integrazione di dati diversi deve essere preceduta da questa operazione che mira a mettere in relazione i dati relativi allo stesso "oggetto".
- *Standardizzazione dei formati*: i dati devono essere resi omogenei con delle operazioni di standardizzazione. Dette operazioni si riferiscono a (i) Congiunzione o spezzamento di campi: informazioni presenti in più campi vengono ricondotte a un campo solo o l'informazione presente in un campo viene riportata in campi separati; (ii) Standardizzazione di codici di classificazione: convenzioni diverse per indicare lo stesso criterio vengono allineate a una stessa codifica; (iii) Standardizzazione del formato dei dati: dati rappresentati con formati diversi vengono resi omogenei.
- *Ricerca e eliminazione dei duplicati*: in caso di presenza di informazioni duplicate, questa operazione permette di identificarle e di ridurle ad una singola istanza.

Il *caricamento* si occupa invece di trasferire i dati nel data warehouse seguendo il modello multidimensionale introdotto nella Sezione 8.4.1.

### Metadati

Le funzionalità svolte dagli strumenti ETL sono supportate e documentate da metadati (dati relativi ai dati). I metadati sono una parte importante dell'architettura di un data warehouse e di solito sono raccolti in un repository che include:

- *Struttura del DW*: i metadati descrivono la struttura del DW (schemi, viste, dimensioni, gerarchie, la scomposizione in data mart e relativa localizzazione).
- *Metadati operazionali*: si riferiscono alla storia dei dati e documentano la fonte di origine e le trasformazioni subite.
- *Metadati per mappare i dati operazionali ai dati caricati nel DW*: forniscono informazioni sulle sorgenti e sul loro contenuto, le regole di trasformazione e di aggiornamento del DW.
- *Statistiche d'uso*: i metadati descrivono anche come e con che frequenza viene utilizzato il DW.

#### 8.4.4 Modello concettuale del data warehouse

Come detto in precedenza, i data warehouse si basano sul modello multidimensionale dei dati nel quale vengono rappresentati i fatti, che sono gli elementi di interesse dell'analisi tramite un insieme di misure, definite tramite le dimensioni che sono le coordinate del fatto.

Esistono diversi modelli concettuali per rappresentare i sistemi multidimensionali. Quello 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 stesso. Le dimensioni possono essere semplici attributi del fatto oppure gerarchie, rappresentate come alberi, che hanno come radici le dimensioni di base. Alcuni attributi delle gerarchie possono essere opzionali e questo viene indicato nel DFM tramite una barra sulla linea corrispondente all'attributo.

Un esempio di DFM rappresentante il fatto vendita per la catena di supermercati citato prima è riportato in Figura 8.18. Il fatto in questo caso è rappresentato dalla vendita che ha come misure la quantità venduta, il valore unitario e l'eventuale sconto corrispondente. Il fatto è determinato da quattro dimensioni: la data in cui la vendita è stata effettuata, il luogo in cui si è verificato il fatto, il prodotto oggetto della vendita e il cliente. Ad ognuna di queste dimensioni è associata una gerarchia, attraverso la quale è possibile effettuare analisi più o meno dettagliate sui fatti registrati nel data warehouse. Ad esempio è possibile analizzare la quantità totale di vendite relative ad uno specifico prodotto per tutto il periodo di analisi per i punti vendita relativi alla città di Milano raggruppati in base al sesso dei clienti. Queste informazioni risultano dall'aggregazione lungo le varie gerarchie delle dimensioni considerate. Nel DFM modellato, l'attributo età per il cliente è opzionale.

Il DFM mostrato come esempio corrisponde all'ipercubo relativo alle vendite per la catena di supermercati. Il data warehouse però non sarà composto da un solo ipercubo, ma da tanti ipercubi, ognuno in grado di modellare le dimensioni relative ad un fatto di interesse per l'analisi. Ad esempio, il data warehouse di un'azienda interessata ad effettuare analisi relative alle vendite, all'efficienza del supporto clienti e all'ottimizzazione dell'interazione con i fornitori avrà tre

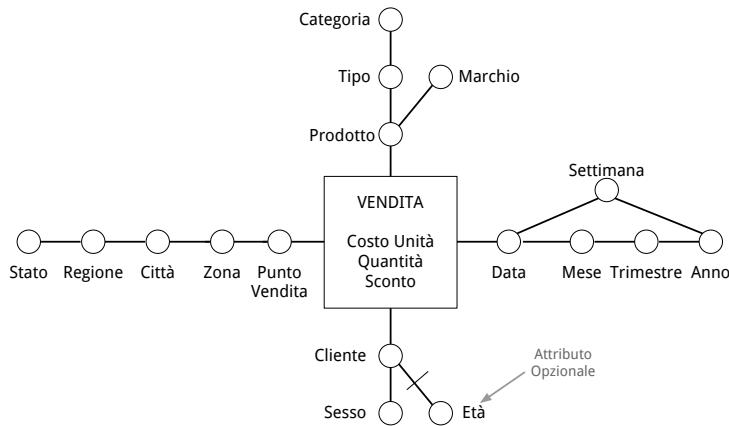


Figura 8.18: Dimensional Fact Model per le vendite di un supermercato

diversi ipercubi nel proprio data warehouse, uno per ogni analisi di interesse: vendite, assistenza clienti, forniture. Per ognuno di questi ipercubi sarà quindi modellato un DFM, con la definizione del fatto, delle misure, delle dimensioni e delle loro gerarchie di interesse per i singoli fatti in analisi.

#### 8.4.5 Modelli logici del data warehouse

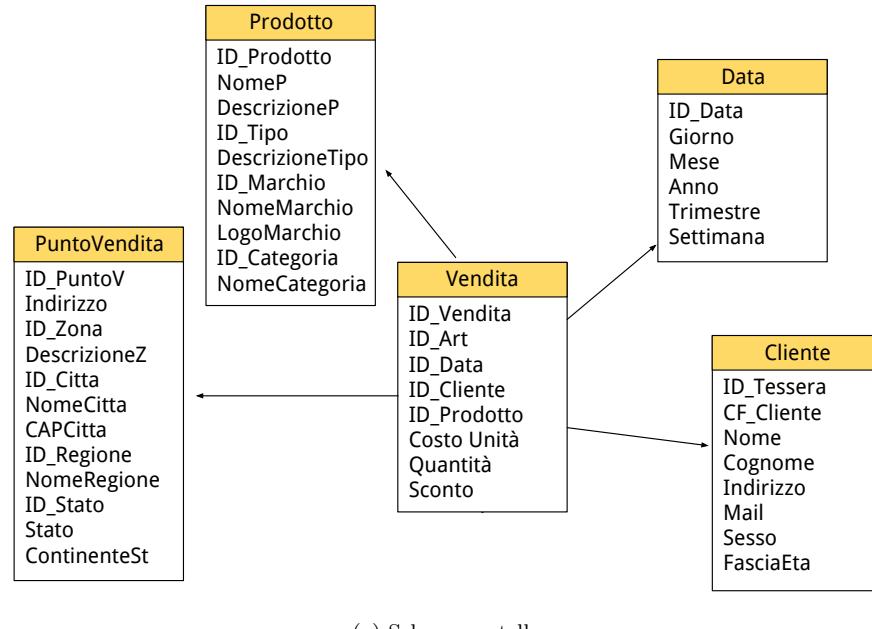
Una volta definito il modello concettuale della base di dati multidimensionale, questa deve essere implementata traducendo il modello concettuale in un modello logico. Si deve quindi scegliere che DBMS utilizzare per gestire l'informazione multidimensionale.

**Modelli MOLAP** La scelta più diretta potrebbe essere quella di utilizzare un modello logico di tipo MOLAP (Multidimensional OLAP). Il modello MOLAP traduce il modello concettuale in una base di dati multidimensionale che può essere interrogato utilizzando motori multidimensionali progettati appositamente per trattare questa tipologia di dati. Questo tipo di modello ha il vantaggio di tradurre in modo esatto il modello concettuale e di sfruttare tutti i vantaggi di una base di dati multidimensionale, rendendo le interrogazioni efficienti e veloci. D'altro canto però, le basi di dati multidimensionali sono molto meno diffuse delle classiche basi di dati relazionali e utilizzarle richiede conoscenze specifiche che l'utente medio non possiede. Inoltre, si tratta per lo più di modelli e linguaggi proprietari per i quali non è definito alcun standard diffuso. Infine, il modello multidimensionale si basa sul concetto di ipercubo, in cui per ogni possibile combinazione dei valori delle dimensioni deve essere allocata una cella di memoria per salvare il fatto corrispondente, anche se questo in realtà non si è verificato e la cella rimane vuota.

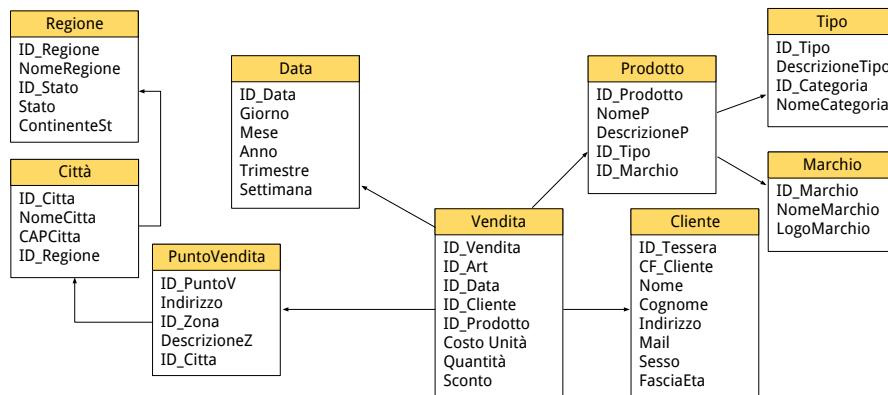
**Modelli ROLAP** Per ovviare a questi problemi è possibile utilizzare in alternativa al modello MOLAP un modello concettuale di tipo ROLAP (Relational OLAP). I modelli ROLAP traducono il modello concettuale multidimensionale in un modello relazionale che rappresenta quindi i dati mediante l'uso di tabelle e permette di svolgere le interrogazioni utilizzando linguaggi di interrogazione noti basati su SQL ed effettuando operazioni di JOIN sulle tabelle descriventi il fatto e le sue dimensioni. I vantaggi principali di questo modello sono una conseguenza della conoscenza degli strumenti di interrogazione su basi di dati relazionali che permettono quindi agli utenti di interagire con il data warehouse senza conoscenze specifiche. Inoltre, essendo basata sul modello relazionale, questa rappresentazione richiede spazio occupato ridotto rispetto a quello necessario per il modello MOLAP, poiché ogni fatto corrisponde ad una riga della base di dati e non è necessario riservare spazio anche per fatti che non sono stati registrati. Questo modello ha però alcuni svantaggi dovuti alla difficoltà di aderire al modello concettuale dei dati e alla lentezza e macchinosità delle interrogazioni su un motore relazionale che poco si presta a gestire dati multidimensionali.

**Modelli HOLAP** Il modello HOLAP (Hybrid OLAP) traduce il modello multidimensionale in modo ibrido, solitamente utilizzando una base di dati relazionale per il data warehouse, limitando l'occupazione di spazio e permettendo l'interazione con strumenti conosciuti. Le basi di dati multidimensionali sono invece usate per i data mart tematici, in cui le dimensioni sono comunque limitate per la loro natura, ma che hanno in questo modo il vantaggio di interrogazioni più efficienti.

**Schemi multidimensionali su basi di dati relazionali** Al fine di mappare una base di dati multidimensionale in uno schema relazionale è necessario identificare le tabelle che comporranno questo schema. In letteratura esistono due approcci principali per questo scopo: lo *schema a stella* e lo *schema a fiocco di neve*. Nello schema a stella, vengono utilizzate due tipologie di tabelle: la tabella dei fatti e le tabelle delle dimensioni. La prima contiene tutti gli attributi corrispondenti alle misure del fatto e ogni riga di questa tabella corrisponderà quindi ad un fatto specifico. Le tabelle delle dimensioni contengono invece, per ogni dimensione associata al fatto nello schema multidimensionale, tutti gli attributi relativi alla gerarchia corrispondente alla dimensione. Un'interrogazione si tradurrà quindi in un JOIN tra tutte le tabelle coinvolte sulla base del fatto di interesse. In questo modo però viene perso il concetto di gerarchia delle dimensioni, che viene appiattita all'interno di un'unica tabella. Un approccio alternativo prevede di usare uno schema a fiocco di neve, in cui ad ogni dimensione vengono associate più tabelle con le opportune relazioni, che permettono così di conservare alcune delle dipendenze funzionali più rilevanti per gli utenti. Un confronto tra i due schemi è mostrato in Figura 8.19 in cui sono mostrati lo schema a stella e lo schema a fiocco di neve relativi al modello multidimensionale il cui modello concettuale è mostrato in Figura 8.18.



(a) Schema a stella



(b) Schema a fiocco di neve

Figura 8.19: Confronto tra schema a stella e schema a fiocco di neve per rappresentare il fatto Vendite di una catena di supermercati

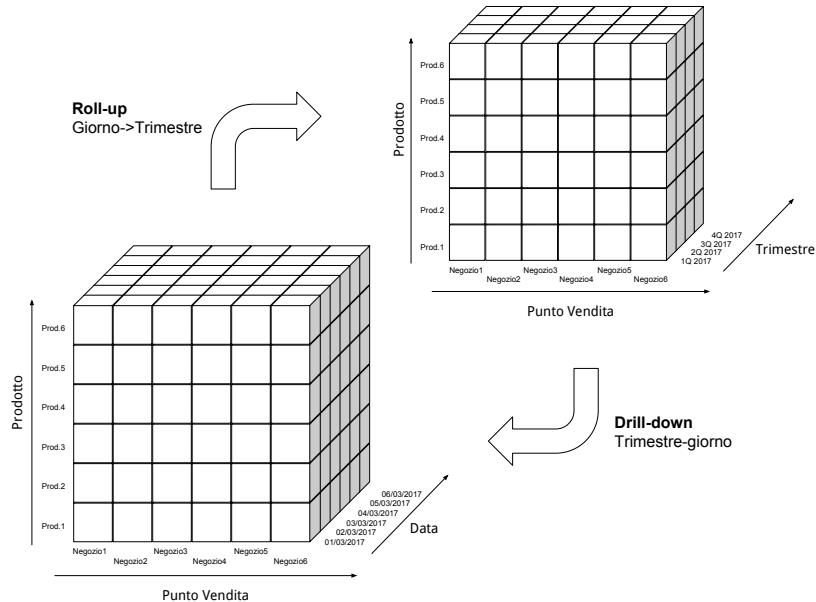


Figura 8.20: Esempio di roll-up e drill-down con aggregazione/disaggregazione sulla dimensione *Data*

#### 8.4.6 Operazioni sul data warehouse

Il data warehousing comprende, oltre ad un modello dei dati direzionali, anche un insieme di tecniche che possono essere utilizzate per analizzare i dati. Queste tecniche devono permettere ad utenti anche poco esperti di interagire con l'ipercubo dei fatti per ottenere informazioni più o meno dettagliate sulla base delle esigenze, focalizzandosi sulle dimensioni di interesse. L'interazione parte solitamente da un'ipotesi formulata da un utente che porta all'estrazione del sottoinsieme dei dati di interesse che può essere approfondita con interazioni successive. Ogni interazione consiste nell'applicazione di un operatore OLAP. I principali operatori OLAP sono: drill down, roll up, slice e dice.

Il **drill-down** è l'operazione che permette di ottenere dati più dettagliati scendendo lungo una gerarchia di una dimensione e quindi passando da un livello di aggregazione più alto ad uno più basso. Considerando il DFM di Figura 8.18, un'operazione di drill down potrebbe far passare da un'analisi per trimestre ad un'analisi per singolo giorno delle vendite (v. Figura 8.20). L'operazione opposta al drill down è quella di **roll-up**, in cui il livello di dettaglio si riduce passando ad un livello di granularità inferiore lungo una dimensione di analisi. Un esempio può essere quello di passare da un'analisi mensile delle vendite ad una annuale in cui i valori per mese sono tra loro aggregati.

L'operatore **slice** permette di focalizzare l'analisi su una porzione dei dati fissando il valore per una delle dimensioni di analisi. Ad esempio, è possibile

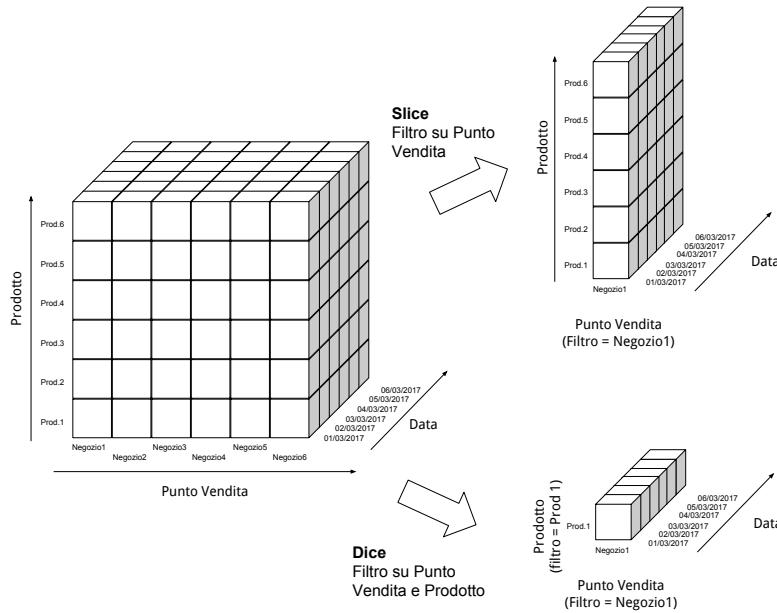


Figura 8.21: Esempio di slice e dice

focalizzarsi sulle vendite relative ad un singolo punto vendita, mantenendo però tutti i dettagli relativi alle altre dimensioni di analisi. Il risultato di questa operazione è una fetta dell'iper cubo originale, identificata dalla dimensione fissata (v. Figura 8.21). Si noti che, contrariamente al roll-up presentato precedentemente, lo slice seleziona un singolo negozio e non aggrega rispetto al negozio. Un'altra operazione di riduzione è quella di **dice** in cui però ad essere identificate sono un insieme di coordinate a qualsiasi livello gerarchico per qualsiasi dimensione desiderata. Il risultato in questo caso è sempre un iper cubo ma con un insieme ridotto dei dati. Ad esempio è possibile analizzare i dati delle vendite relativi a un prodotto (ad esempio *prodotto1*) in un determinato punto vendita (ad esempio *negozi1*).

#### 8.4.7 Ciclo di vita del data warehouse

Come detto nella Sezione 8.4.4, il data warehouse di un'organizzazione sarà composto da tanti ipercubi, ognuno contenente le informazioni rilevanti per un fatto di interesse per la gestione direttiva dell'organizzazione.

La costruzione del DW segue quindi un approccio iterativo. Si parte identificando il fatto di maggiore interesse e si modella e popola quindi il primo iper cubo. Successivamente, uno alla volta, altri ipercubi di interesse vengono aggiunti e a loro volta popolati. Questo approccio ha un duplice vantaggio: da una parte rende possibile individuare i fatti di interesse in momenti successivi, rispondendo a nuove esigenze dell'organizzazione; d'altro canto, limita l'inve-

stimento iniziale sviluppando la base di dati multidimensionale un pezzo alla volta.

La fase di popolamento del DW è una fase complessa che richiede l'esecuzione di diverse operazioni. Come già detto, per popolare il DW si fa uso degli strumenti di ETL (Extraction, Transformation, e Loading) (Sezione 8.4.3) in cui l'ultima fase consiste nel *caricamento* dei dati nel DW. I dati vengono inseriti partendo dal livello di aggregazione massimo e scendendo poi ai livelli più dettagliati.

Durante l'inserimento è importante considerare anche le problematiche relative all'aggiornamento dei dati che può riguardare i fatti o le dimensioni. Aggiornare un fatto significa inserire un valore non esistente per una combinazione di dimensioni, o modificare un valore esistente aggiornandolo. Aggiornare una dimensione può invece portare ad una situazione di incongruenza. Ad esempio, se la dimensione indirizzo per un cliente varia nel tempo, sarà necessario prendere una delle seguenti decisioni: (1) aggiornare il valore indirizzo anche per i fatti pre-esistenti in modo che un cliente sia sempre associato allo stesso indirizzo; (2) mantenere valido il valore precedente senza eseguire nessun aggiornamento; (3) adottare una soluzione ibrida in cui si conserva traccia dei diversi indirizzi relativi al cliente relazionati ai relativi intervalli temporali. L'ultima soluzione è quella più complessa da gestire.

Il ciclo di vita di un data warehouse mette in luce anche i suoi limiti. Infatti, è importante sottolineare come il processo ETL possa richiedere, per la sua esecuzione, anche diverso tempo. Non è infatti da considerarsi un'attività necessariamente automatica. Specie nella fase di trasformazione, dove è necessario risolvere alcune discordanze tra dati provenienti da fonti diverse, potrebbe rendersi necessario anche l'intervento umano. Pertanto, non solo la creazione di un data warehouse è un'operazione onerosa, ma anche il suo mantenimento e il continuo aggiornamento dei dati richiede un discreto sforzo.

Questa complessità produce, come effetto, un aumento della distanza tra l'aggiornamento dei dati nelle basi di dati sorgenti - che avviene molto di frequente - e il momento in cui questi aggiornamenti sono visibili in un data warehouse. Per ridurre questa distanza, negli ultimi anni, sono stati proposti approcci alternativi ai data warehouse che spesso sfruttano algoritmi e tecniche proprie dei big data che, pur non richiedendo una fase di progettazione così dettagliata come nel caso del data warehouse, sono in grado di produrre risultati di analisi in tempo reale.

## 8.5 Data Mining

Il data warehousing opera a supporto dei processi deduttivi dell'utente, fornendo un insieme di strumenti atti all'interazione in tempo reale con la base di dati multidimensionale e alla navigazione dei dati. Tale interazione permette all'utente di svolgere analisi sui dati alla ricerca di specifiche informazioni considerate di interesse da quell'utente specifico. Questo approccio all'analisi dei dati direzionali all'interno di un'organizzazione può risultare insufficiente, ed

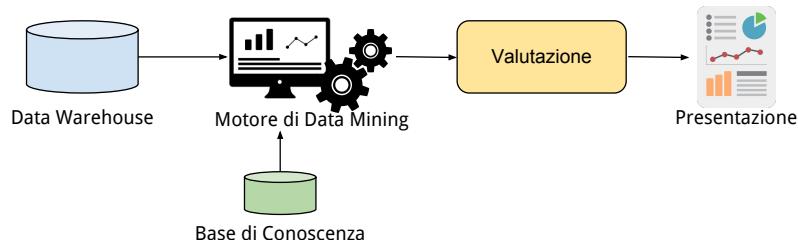


Figura 8.22: Architettura del Data Mining a partire dal data warehouse

ulteriori analisi volte ad estrarre in modo automatico l'informazione nascosta all'interno del data warehouse possono arricchire l'insieme di conoscenze che possono guidare le strategie aziendali.

Come definito in [29], il *data mining* è l'attività volta a riconoscere ed estrarre in modo automatico informazioni nascoste da basi di dati di grandi dimensioni, come ad esempio il data warehouse. Il processo di data mining è composto da diversi passi che possono essere iterati varie volte al fine di raffinare l'informazione ottenuta. Questo processo, seppure per buona parte automatico, richiede l'interazione con l'utente in diverse fasi. L'utente può infatti guidare il processo iterativo indicando le direzioni di indagine di maggiore interesse. Inoltre, anche i risultati finali offerti dalle analisi di data mining devono essere validati e selezionati dagli utenti, in quanto parte dell'informazione risultante potrebbe rivelarsi essere di scarso interesse.

In generale, il processo di data mining è composto dai seguenti passi:

1. Selezione dei dati ritenuti di interesse ai fini dell'analisi.
2. Pulizia dei dati (cleaning) al fine di eliminare incongruenze ed errori.
3. Integrazione dei dati provenienti da fonti diverse al fine di renderli congruenti.
4. Trasformazione e strutturazione dei dati risultanti dalle fasi precedenti.
5. Analisi di data mining volta alla ricerca di condizioni notevoli all'interno dei dati analizzati.
6. Valutazione dei risultati e loro riduzione ai soli ritenuti rilevanti.
7. Presentazione dei risultati finali all'utente.

I passi da 1 a 4 corrispondono ai passi necessari per la creazione del data warehouse, mentre i passi da 5 a 7 sono tipici del solo data mining. L'architettura del data mining è mostrata in Figura 8.22.

Le funzioni di data mining possono essere classificate in diverse categorie [29, 40] brevemente discusse in questo paragrafo. In primo luogo, 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*. Le tecniche di *apprendimento supervisionato* associano ad ogni record considerato una variabile target che può essere ad esempio un’etichetta di classificazione di quell’insieme di dati. L’operazione di associazione tra il record e l’etichetta di classificazione è detta “labelling”. La variabile target costituisce l’output desiderato a partire dall’analisi del record che sarebbe l’input al sistema. Nell’*apprendimento non supervisionato* le tecniche non considerano una variabile target e di conseguenza mirano a identificare regolarità o regole ricorrenti significative.

Un’altra classificazione può essere fatta considerando gli obiettivi di analisi in base a cui si possono distinguere tecniche *predittive* e *descrittive*. Le *tecniche descrittive* analizzano i dati del passato e li categorizzano, filtrano e individuano pattern applicando funzioni matematiche o statistiche. Le *tecniche predittive* invece analizzano i dati del passato per supportare predizioni e decisioni in grado di anticipare eventi.

Considerando le funzionalità fornite possiamo individuare diverse tecniche di data mining. Quelle descritte in questo testo sono le regole associative, la classificazione e il clustering. In base alle categorie descritte, le regole associative e il clustering sono due tecniche descrittive a apprendimento non supervisionato, mentre la classificazione è una tecnica predittiva ad apprendimento supervisionato.

### 8.5.1 Regole Associative

Le regole associative descrivono le relazioni che intercorrono tra diversi attributi della base di dati, al fine di identificare condizioni che si verificano spesso contemporaneamente. Sono solitamente espresse con la formula  $A \Rightarrow B$  dove A è detto antecedente e B conseguente. Le regole associative possono essere utilizzate ad esempio per la market basket analysis, cioè per identificare prodotti che un utente compra spesso insieme. Un esempio è la seguente regola:

$$\text{compra}(X, \text{cereali}) \Rightarrow \text{compra}(X, \text{latte})$$

La validità di una regola associativa può essere valutata sulla base di due valori: il supporto e la confidenza. La *confidenza* misura la certezza della relazione trovata, e quindi quante volte nel data set analizzato, se si verifica la condizione espressa nell’antecedente, si verifica anche quella del conseguente. In termini probabilistici, questa relazione si può esprimere come la probabilità condizionata  $P(B|A) = P(A, B)/P(A)$ , e quindi tramite il rapporto tra il numero di volte che A e B si verificano insieme (un acquisto include sia latte sia cereali) sul numero di volte che si osserva A (un acquisto include i cereali). Il *supporto* indica invece il numero di volte per cui la regola è verificata sul totale delle registrazioni nel dataset ed è quindi uguale a  $P(A, B)$ .

Come esempio, consideriamo di analizzare i dati delle vendite di prodotti cercando di trovare relazioni tra la vendita di prodotti diversi. Consideriamo il seguente estratto della base di dati:

$$t_1 = \{p_1, p_2, p_3\}$$

$$t_2 = \{p_1, p_4\}$$

$$t_3 = \{p_1, p_3\}$$

$$t_4 = \{p_2, p_5, p_6\}$$

dove  $t_i$  rappresenta un timestamp e  $p_i$  rappresenta la vendita di un prodotto. Se volessimo estrarre le regole associative che hanno sia supporto S che confidenza C superiore a 0,5 otterremmo:

$$p_1 \Rightarrow p_3 \text{ con}$$

$$S = P(p_1, p_3) = \frac{2}{4} = 0,5 \text{ e}$$

$$C = P(p_3|p_1) = \frac{P(p_1, p_3)}{P(p_1)} = \frac{\frac{2}{4}}{\frac{3}{4}} = 0,66$$

$$p_3 \Rightarrow p_1 \text{ con}$$

$$S = P(p_1, p_3) = \frac{2}{4} = 0,5 \text{ e}$$

$$C = P(p_1|p_3) = \frac{P(p_1, p_3)}{P(p_3)} = \frac{\frac{2}{4}}{\frac{2}{4}} = 1$$

Entrambe le relazioni, essendo speculari, si verificano due volte in  $t_1$  e  $t_3$  su un totale di quattro casi analizzati, quindi il supporto  $P(A, B)$  in entrambi i casi sarà del 50%. La confidenza invece dipende anche dal numero di volte che il primo termine si verifica nei casi analizzati. Mentre  $p_1$  si verifica tre volte su quattro (di cui una volta indipendentemente dal verificarsi di  $p_3$ ),  $p_3$  si verifica solo due volte e sempre in congiunzione con  $p_1$ . Per questo motivo, la confidenza delle due regole, calcolata come  $P(A, B)/P(A)$  è differente.

Possibili applicazioni delle regole associative sono la pianificazione di campagne promozionali efficaci (suggerimento di prodotti in base ad acquisti precedenti dei clienti) e l'attivazione di azioni preventive di manutenzione degli impianti. Un esempio del primo caso è osservabile sul sito di acquisti on line Amazon, dove la visualizzazione di un prodotto è sempre accompagnata da un elenco di altri prodotti frequentemente acquistati insieme al prodotto osservato.

### 8.5.2 Classificazione

La classificazione permette di assegnare in modo automatico gli elementi della base di dati a classi predefinite. La classificazione è una procedura guidata, in cui si parte da un insieme di elementi annotato, cioè per i quali è nota la classe di appartenenza, e si costruisce un modello in grado di identificare quali sono le caratteristiche che permettano di assegnare un elemento per cui la classe non è nota ad una delle classi analizzate. Questo modello viene poi testato con un insieme di dati per cui la classe è nota per vedere se il classificatore è in grado di individuare la categoria corretta. L'insieme degli elementi annotati usati per costruire il modello è solitamente chiamato *training set*, mentre l'insieme degli elementi usati per testare i risultati della classificazione è chiamato *test set*. La qualità del modello viene misurata con la sua capacità di classificare in modo corretto gli elementi che compongono il test set. Una volta ottenuto un modello soddisfacente è possibile usarlo su dati non classificati per effettuare una pre-

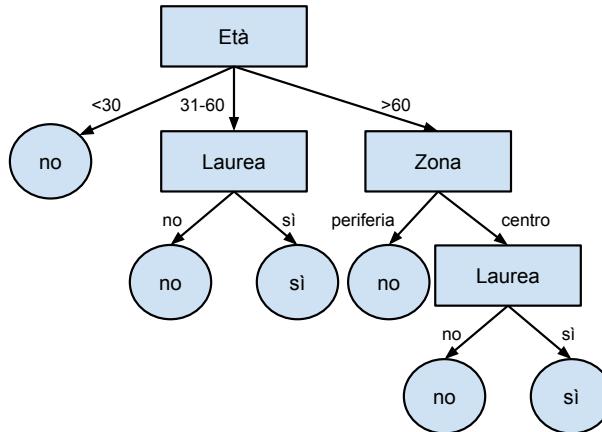


Figura 8.23: Albero di decisione per la predizione dell'acquisto di un televisore OLED

dizione sulla classe di appartenenza. Possibili applicazioni della classificazione sono la predizione della qualità dei fornitori o dei prodotti, e la predizione della propensione all'acquisto dei clienti.

Esistono diversi tipi di classificatori basati su diverse tecniche tra cui funzioni matematiche, reti Bayesiane, alberi di decisione e reti neurali. Essi si distinguono in base a un insieme di caratteristiche: (i) Accuratezza - percentuale elementi classificati correttamente; (ii) Velocità - tempo di costruzione del modello e tempo di classificazione; (iii) Scalabilità - capacità del classificatore di costruire un modello con un crescente numero di variabili; (iv) Robustezza - capacità di classificazione in caso di elementi mancanti; (v) Interpretabilità, ovvero la facilità nell'interpretare i risultati del modello.

Come esempio di modello di classificazione consideriamo gli *alberi di decisione*, in cui i nodi sono gli attributi del soggetto da classificare, gli archi i valori assumibili dall'attributo, le foglie sono le classi a cui gli elementi possono essere assegnati. Un esempio di albero di decisione è mostrato in Figura 8.23, in cui l'obiettivo è quello di predire se un utente acquisterà o meno un televisore OLED. Gli attributi considerati sono l'età del cliente (suddivisa per fasce d'età), il suo livello di istruzione (possesso della laurea) e la sua residenza (centro città o periferia). L'albero, costruito sulla base dei dati relativi ad acquisti precedenti di altri utenti, può essere utilizzato per decidere se indirizzare una pubblicità specifica per il prodotto sulla base delle caratteristiche dei clienti.

La costruzione di un albero di decisione si svolge tramite raffinamenti successivi. Partendo da un training set etichettato, si seleziona un attributo come nodo radice (nella figura l'età) e si suddivide il dataset sulla base dei diversi valori assunti da tale attributo (in questo caso discretizzati su tre gruppi: minore o pari a 30 anni, tra 31 e 60 anni, e maggiore di 60 anni). 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 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 (nell'esempio “compro” o “non compro”). 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.

La selezione degli attributi che permetta di arrivare ad un modello efficace con un numero ridotto di passi è quindi una fase molto delicata. L'esempio mostrato in figura considera un **albero generale**, in cui ogni attributo può assumere un numero qualsiasi (ma discreto) di valori. Questa condizione non è però generale e un attributo può assumere un numero fisso di valori (se solo due valori si parla di **albero binario**). Maggiore è il numero di valori, più complessa è la costruzione del modello. Nel caso di attributi continui, è possibile applicare delle soglie per ricondurli ad attributi discreti. Inoltre, il passaggio da un livello al successivo può dipendere dalla valutazione di un solo attributo (**albero univariato**) o dalla combinazione dei valori di più attributi (**albero multivariato**).

Come detto in precedenza, il modello ottenuto può essere valutato in base alla sua capacità di classificare gli elementi del test-set. Sulla base delle scelte effettuate (ordine degli attributi selezionati ad ogni passo) per un dataset possono essere costruiti diversi modelli di alberi di decisione. La valutazione permette di selezionare quello che meglio è capace di rappresentare il dataset considerato.

A tal proposito è fondamentale il ruolo del *test set*. Infatti, per ogni suo elemento, attraverso l'operazione di labelling, si conosce la sua corretta classificazione. Riprendendo l'esempio dell'acquisto del televisore, per ogni individuo è indicato se ha intenzione di acquistare o meno il televisore. Un albero di decisione risulta quindi perfetto quando è in grado di classificare ogni elemento in una classe identica a quella per cui è stato etichettato. Riprendendo l'esempio di Figura 8.23, per ogni elemento del test set con meno di 30 anni non deve essere previsto l'acquisto del televisore, altrimenti sarebbe in contrasto con quanto indicato in fase di labelling.

Nella realtà esistono sempre delle discrepanze, e queste sono calcolate partendo dall'analisi dei cosiddetti *falsi positivi* e *falsi negativi* (v. Figura 8.24). Un falso negativo si verifica quando un individuo non viene classificato come appartenente ad una classe ma in realtà dovrebbe appartenervi. Ad esempio, predico attraverso l'albero che un utente non è associato alla classe di acquirenti che comprerà il televisore, quando in realtà nel test set questo individuo è etichettato come un individuo che compra il televisore. In modo duale, un falso positivo si verifica se un individuo viene associato alla classe degli acquirenti di un televisore ma in realtà l'individuo non comprerà il televisore.

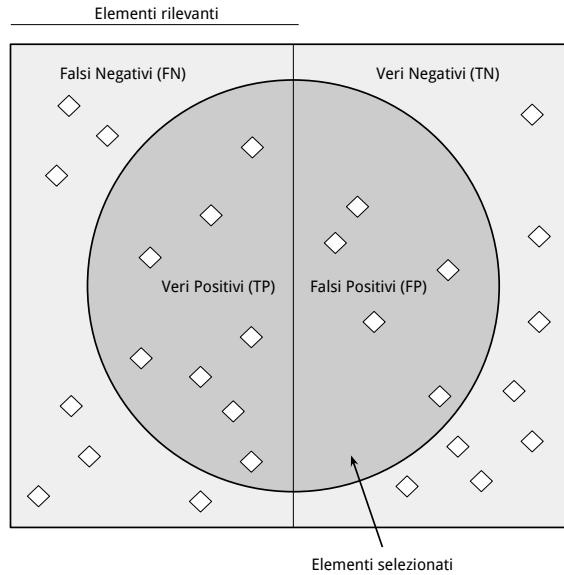


Figura 8.24: Illustrazione dei possibili risultati della classificazione: falsi positivi, falsi negativi, veri positivi e veri negativi

In modo analogo, si possono definire anche i cosiddetti veri positivi e veri negativi che, contrariamente ai casi precedenti, indicano quando correttamente un individuo è inserito, o non inserito, in una classe.

Calcolati quindi il numero totale di: (i) falsi positivi (FP), (ii) falsi negativi (FN), (iii) veri positivi (TP), (iv) e veri negativi (TN) per l'intero test set, si definiscono *Precision* e *Recall*. La Precision è la percentuale di elementi classificati correttamente tra tutti quelli classificati come appartenenti ad una classe:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Formulando la definizione in modo intuitivo, la Precision indica con che accuratezza un albero di decisione inserisce gli individui in una classe. Ad esempio, una Precision pari al 66% indica che nei due terzi dei casi, un individuo che viene identificato come potenziale acquirente del televisore effettivamente lo diventi.

La Recall indica quanti elementi l'algoritmo di classificazione è stato in grado di associare alla classe rispetto a quelli che idealmente costituiscono la classe:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Riferendoci all'esempio visto sopra, una Recall pari al 50% indica che gli elementi che l'albero di decisione ha inserito nel gruppo dei potenziali acquirenti

del televisore costituisce solo la metà dell'effettivo insieme di persone che davvero acquisterà il televisore.

### 8.5.3 Clustering

Il clustering, come la classificazione, permette di raggruppare un insieme di elementi in base alle loro caratteristiche, assegnando loro una classe di appartenenza. La principale differenza tra i due approcci consiste nel fatto che il clustering non richiede di conoscere a priori le classi di appartenenza dell'insieme di training, ma trova le classificazioni e talvolta anche il loro numero di classi in modo completamente automatico sulla base delle similarità dei valori degli attributi degli elementi del dataset. Il risultato del clustering quindi è un insieme di cluster (raggruppamenti) composti da elementi tali che siano verificate le seguenti condizioni: (i) la somiglianza tra gli elementi appartenenti allo stesso cluster è massima; (ii) la somiglianza tra elementi di cluster diversi è minima.

I metodi di clustering sono applicati in diversi campi, come le scienze sociali, il riconoscimento delle immagini e le analisi di marketing. Esse si basano su una capacità del cervello umano di raggruppare gli elementi sulla base di affinità. Ad esempio, un insieme di clienti potrebbe essere raggruppato grazie al clustering in base ai loro comportamenti di acquisto dei prodotti.

Per permettere di calcolare la distanza tra gli elementi di un insieme, gli attributi che descrivono gli elementi devono essere numerici. I principali requisiti con cui gli algoritmi di clustering sono valutati sono: (i) la flessibilità - capacità di analizzare elementi categorici oltre che numerici; (ii) la robustezza - stabilità del risultato in presenza di leggere variazioni nel valore degli attributi; (iii) efficienza - il tempo necessario per ottenere la suddivisione del dataset in cluster.

Il più diffuso algoritmo di clustering è l'algoritmo *k-means*, una tecnica non supervisionata. Esso richiede (i) la conoscenza a priori del numero  $k$  di cluster da identificare, (ii) una funzione di somiglianza che indica la vicinanza tra due elementi. La funzione di somiglianza dipende, solitamente attraverso somme pesate, dagli attributi che definiscono un individuo dell'insieme che si sta analizzando. Pensando a clienti di un supermercato, la loro somiglianza potrebbe dipendere dall'età anagrafica, dalla zona geografica di provenienza, e dalla professione. Calcolando la differenza tra i valori di queste dimensioni e considerando l'importanza di tali differenze attraverso somme pesate, è possibile quantificare quanto un individuo è simile a un altro. La funzione di somiglianza è applicata con un approccio iterativo all'insieme per cui si vogliono identificare i 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).

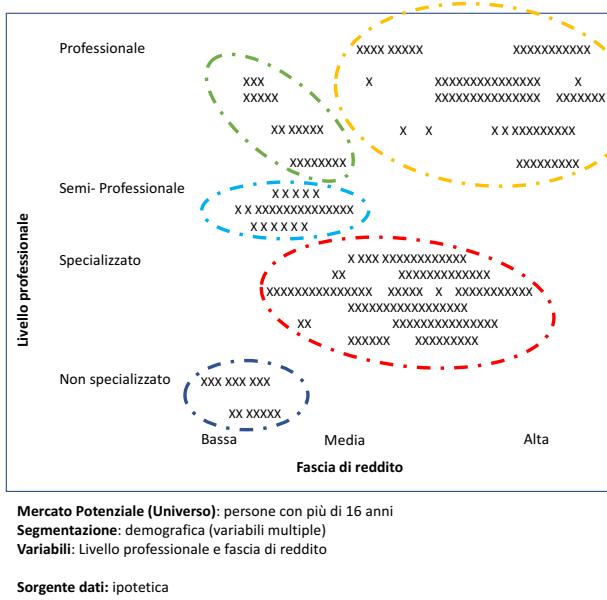


Figura 8.25: Esempio di clustering ( $k=5$ ) utilizzato per la segmentazione del mercato

Un esempio di clustering è mostrato in Figura 8.25 dove il clustering è utilizzato per la segmentazione di mercato, cioè per la suddivisione dei clienti in un insieme di “gruppi sociali” che costituiscono i segmenti di mercato, sulla base di caratteristiche geografiche, socio-demografiche o comportamentali. Nell'esempio, i clienti vengono raggruppati sulla base di due attributi: la loro fascia di reddito (bassa, media, alta) e il loro livello professionale (non specializzato, specializzato, semi-professionale e professionale). A partire dalla distribuzione degli attributi, vengono identificati cinque cluster, ognuno identificante un segmento. È importante notare che ai cluster non è assegnata una categoria chiara, dal momento che esse non sono definite, ma ogni insieme rappresenta elementi che condividono valori simili per gli attributi.

Gli algoritmi di clustering si classificano sulla base dell’assegnazione degli elementi ai cluster. In particolare un algoritmo è **esclusivo** se assegna ogni elemento del dataset ad un solo cluster, mentre è **sovraposto** se uno stesso elemento può essere assegnato a più classi. I metodi di tipo **fuzzy** sono metodi sovrapposti in cui l’assegnazione di un elemento ad ogni classe è associata ad un peso che varia tra 0 e 1. I metodi si classificano inoltre in **completi** se ogni elemento appartiene ad almeno una classe e **parziali** se alcuni elementi possono non essere assegnati a nessuna classe.

## 8.6 Esempi su CRM

Le tecniche di data mining vengono usate dal CRM analitico per analizzare le azioni e il comportamento dei clienti e trovare pattern significativi. Ad esempio le regole associative vengono utilizzate per analizzare gli acquisti dei diversi clienti e trovare i prodotti che vengono comprati insieme più spesso. In particolare le regole  $X \Rightarrow Y$  possono essere utilizzate per consigliare Y a tutti i clienti che comprano X. Ad esempio, consideriamo i servizi di raccomandazione offerti da Amazon: nel momento in cui andiamo a visitare la pagina di un libro, il sito, sotto la sezione “Chi ha acquistato questo articolo ha acquistato anche” ci suggerisce altri libri che altri utenti hanno comprato insieme al libro da noi considerato. Questo è il risultato di regole di associazione che associano il libro da noi considerato (X) ad altri libri spesso comprati insieme (Y). Le regole di associazione permettono alle aziende di essere proattive e anticipare i desideri del cliente.

I metodi di classificazione (ad esempio gli alberi di decisione) ci permettono, a partire da un insieme di oggetti, le loro caratteristiche e la loro classificazione, di trovare il metodo e i criteri attraverso cui associare gli oggetti alle classi in maniera automatica. Queste tecniche hanno trovato spesso impiego nel settore assicurativo: dati i clienti, la loro storia e la fascia di rischio sono stati definiti metodi per classificare i nuovi clienti in probabili fasce di rischio e determinare un premio ad esse commisurato. Criteri di classificazione risultano essere in questo settore l’età, l’area di residenza e il tipo di macchina assicurata. Oppure una azienda che classifica manualmente, in base alla storia, i clienti in diverse categorie quali:

- *Platinum*: utenti affidabili, non sensibili al prezzo, disposti a provare nuovi prodotti, fedeli.
- *Gold*: utenti che fanno molti acquisti, ma che cercano sconti, acquistano da diversi fornitori e non sono fedeli.
- *Iron*: utenti che generano basso volume di affari o che non sono costanti negli acquisti.
- *Lead*: utenti esigenti che richiedono un’attenzione speciale, ma non mostrano fedeltà.

Applicando i metodi di classificazione, si possono trovare le caratteristiche dei clienti da usare come criteri di classificazione per attribuire a priori la categoria e capire quali strategie di marketing adottare.

Il clustering serve per segmentare i clienti e formare gruppi di clienti simili. Consideriamo un provider telefonico che vuole segmentare la sua base di clienti per categorie di utilizzo dei servizi. Se i clienti possono essere classificati per utilizzo, l’azienda può offrire abbonamenti più interessanti ai propri clienti. Le variabili sono date dall’utilizzo dei seguenti servizi: Multiple lines, Voice mail, Internet, Visualization Caller ID, Call waiting, Call forwarding, 3-way calling,

Electronic billing. L'adozione di tecniche di clustering porta all'identificazione di tre cluster:

- *Cluster 1*: clienti che spendono molto e acquistano molti servizi.
- *Cluster 2*: clienti che tendono a spendere moderatamente per calling services.
- *Cluster 3*: clienti che spendono poco e che acquistano pochi servizi.

La segmentazione fornisce informazioni per le campagne di marketing e per l'offerta di servizi avanzati pensati per una specifica categoria di clienti.

## 8.7 Selezione del software

Come discusso nella parte relativa alla pianificazione dei sistemi informativi, alcuni software di tipo COTS possono essere acquisiti in modalità Buy. In questo caso la difficoltà dell'azienda sta principalmente nella selezione del pacchetto più adeguato alle sue esigenze. In questo capitolo vogliamo discutere le variabili da considerare nel processo di selezione che si compone di diverse fasi come descritto nel Capitolo 2.4 (si veda Figura 2.12).

Il primo passo verso la selezione del software è l'attività di raccolta e analisi dei requisiti (ingegneria dei requisiti). Come per ogni attività di progettazione, è necessario definire i requisiti funzionali e non funzionali. I primi si concretizzeranno in una lista di funzionalità che il software dovrà fornire mentre quelli non funzionali definiranno le proprietà e i vincoli architetturali.

I requisiti guidano la ricerca dei pacchetti disponibili sul mercato che devono essere valutati tramite una serie di indicatori che consentano di descriverli e classificarli con sufficiente livello di dettaglio. Gli indicatori capaci di differenziare le diverse soluzioni di mercato sono:

- *Indicatori funzionali*: si tratta delle funzionalità supportate dai diversi moduli che compongono il sistema software in esame.
- *Indicatori architetturali*: aspetti tecnologici critici al momento della scelta del sistema software e soprattutto per la sua scalabilità nel tempo.
- *Costi della soluzione*.

Oltre alle caratteristiche della soluzione, è necessario anche valutare l'azienda che la fornisce. Quindi la valutazione della soluzione deve essere associata ad una valutazione del fornitore del pacchetto e di altri eventuali fornitori associati che potrebbero intervenire nella fase di installazione (ad esempio, per l'integrazione con sistemi legacy). La Figura 8.26 sintetizza i vari aspetti di valutazione da prendere in considerazione.

Per quanto riguarda gli **Indicatori funzionali** si deve dare in primo luogo importanza al *grado di completezza (copertura funzionale)*. Il grado di completezza è una metrica che misura la completezza funzionale dei sistemi software

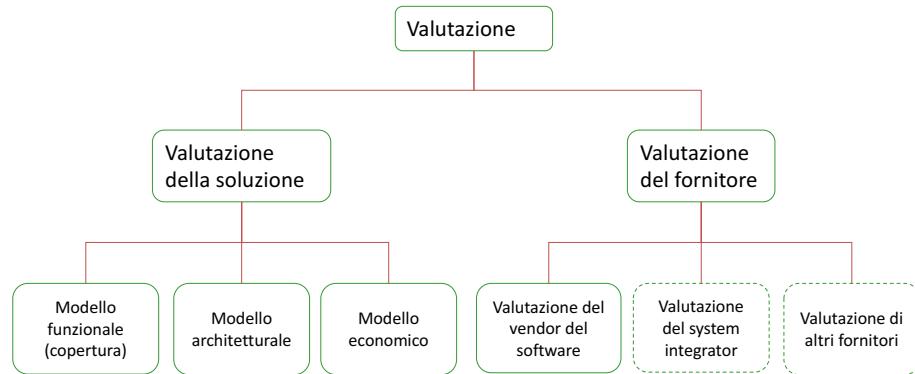


Figura 8.26: Albero di valutazione

come rapporto tra la somma delle funzionalità fornite dal pacchetto e il numero totale di funzionalità richieste dall'utente. Ad esempio, in un caso come quello illustrato in Figura 8.27, si può notare come pacchetti diversi possano fornire la stessa funzionalità in modi diversi: nella selezione bisogna preferire la soluzione che più si avvicina alle esigenze dell'azienda e quindi quella che ha il grado di copertura maggiore.

| Processi                                 | Sistema attuale   | Obiettivi   | Vendor 1                            | Vendor 2                             | Vendor 3  |
|--|---|---|-------------------------------------|--------------------------------------|---|
| <b>Gestione degli ordini del cliente</b> | Solo registrazione su sistema locale; ordini non visibili dal cliente | Portale per cliente<br>Stato ordine visibile<br>Simulare la schedulazione | Portale per cliente                 | Simulazione della schedulazione      | Registrazione degli ordini  |
| <b>Schedulazione produzione</b>          | Manuale   | Schedulazione a capacità finita   | Manuale                             | Schedulazione a capacità finita      | Schedulazione a capacità finita, ottimizzazione delle risorse. Facilmente configurabile |
| <b>Produzione</b>                        | Registrazione delle bolle   | Rilevazione strumentale e manuale dell'avanzamento                        | Rilevazione con terminali portatili | Rilevazione manuale dell'avanzamento | Rilevazione strumentale e manuale dell'avanzamento                                      |

Figura 8.27: Esempio di confronto di copertura

Un altro indicatore funzionale rilevante è la *Personalizzabilità* della soluzione che indica se la soluzione può essere adattata alle esigenze dell'azienda tramite parametrizzazione e personalizzazione o solo tramite parametrizzazione. Alcune aziende (poche, per fortuna) non offrono infatti la possibilità di alterare il codice in quanto ciò richiederebbe la riprogrammazione del modulo in base alle richieste dell'utente.

Gli **Indicatori architetturali** possono essere diversi e dipendono dalla natura del pacchetto. Gli indicatori più diffusi sono:

- *La scalabilità dell'infrastruttura:* la metodologia dovrà fornire delle indicazioni a supporto della scelta di un sistema scalabile nel tempo, in modo tale da mantenere inalterate le prestazioni (o avere la facoltà di migliorarle) al crescere dal carico transazionale e delle funzionalità del sistema informativo. L'indice di scalabilità può essere valutato considerando il numero di stazioni gestibili dal software, il numero di livelli dell'architettura o il modo di gestire la base di dati (v. Fig. 8.28).
- *Il grado di interoperabilità della soluzione:* la metodologia dovrà aiutare a comprendere quanto l'applicativo possa essere considerato una entità aperta verso il mondo esterno.
- *Il livello di sicurezza del sistema:* questa variabile misura la sicurezza assicurata dal sistema, sia in termini di regolazione degli accessi sia a livello di protocolli (per esempio, protocolli crittografici) utilizzati per lo scambio di dati riservati.

| Variabile   | Dimensione   | Metrica per il software                     | Valori Possibili                       |
|-------------|--------------|---|--|
| Scalabilità | Estensionale | Numero di stazioni gestibili dal software   | $\in \mathbb{N}$                       |
|             | Intensionale | Numero di moduli installabili separatamente | $\in \mathbb{N}$                       |
|             |              | Gestione distribuzione dati                 | 0 = Non distribuita<br>1 = Distribuita |
|             |              | Gestione replicazione dei dati              | 0 = Gestita<br>1 = Non gestita         |
|             |              | Numero di livelli                           | $\in [2, \mathbb{N}]$                  |

Figura 8.28: Esempio di indicatori architetturali

Una volta identificate le applicazioni che soddisfano sia requisiti funzionali che non funzionali, si valuta l'affidabilità dei fornitori dei pacchetti. La **credibilità** e la **capacità** del produttore (vendor) si valutano considerando l'esperienza maturata, la presenza territoriale nazionale ed internazionale, la reputazione organizzativa e l'attenzione all'evoluzione del prodotto nel tempo. Sono dunque da preferire i vendor noti che hanno una certa copertura sul territorio e una consolidata reputazione, costruita anche sulla base delle aziende clienti. Non è da sottovalutare poi la frequenza di aggiornamento del prodotto che denota una certa attenzione allo stesso e al suo aggiornamento e evoluzione. Analogamente vengono valutati i system integrator (chi realizza integrazione di pacchetti software) e gli eventuali altri fornitori.

L'elenco dei pacchetti idonei viene solitamente ridotto a seguito della valutazione economica. Dato un certo budget, è necessario considerare i **costi della**

**soluzione** per verificare la fattibilità dell'investimento. I costi della soluzioni non comprendono solo i costi delle licenze ma devono essere considerati altri fattori. I costi classificabili come *investimenti di progetto* sono legati ai costi delle licenze (che dipendono dalle funzionalità e dal numero di postazioni), ai costi hardware, al costo delle risorse umane e ai costi di change management. Inoltre bisogna considerare i *costi di gestione* che considerano i canoni di manutenzione, costi infrastrutturali, costi di personalizzazione e aggiornamento del software.

Per quanto riguarda gli ERP, la composizione numerica media dell'investimento è in genere suddivisa come segue: consulenza (30%), hardware (25%) sviluppo software (15%), prodotti software (15%) e formazione del personale (15%). Bisogna, inoltre, considerare un costo stimato intorno al 15% annuo dell'investimento iniziale per il supporto operativo, assistenza e manutenzione.

Le soluzioni che superano anche la valutazione economica vengono analizzate nel dettaglio attraverso demo e incontri con i fornitori. Gli incontri sono fondamentali per chiarire i requisiti utente, capire al meglio il prodotto e prendere la decisione finale.

## 8.8 Domande

- Illustrare le caratteristiche generali dei sistemi ERP e discutere le alternative per il loro inserimento in un diagramma architettonico secondo l'approccio BOAT.
- Che cosa si intende per sistema ERP? Descriverne principi di funzionamento, vantaggi d'uso e architettura funzionale di massima.
- Illustrare l'architettura funzionale dei sistemi ERP e discutere i vantaggi e gli svantaggi degli approcci make e buy.
- Illustrare il ruolo dei dati in un sistema ERP.
- Illustrare le caratteristiche dei sistemi ERP e la loro architettura funzionale.
- Discutere i criteri che vengono utilizzati per decidere tra la selezione di componenti software (approccio buy) e la realizzazione di un nuovo software ad hoc (approccio make).
- Illustrare l'architettura e le principali proprietà di un Data Warehouse. Esemplicare alcune problematiche di estrazione e caricamento dei dati.
- Discutere i criteri che vengono utilizzati per decidere tra la selezione di componenti software e la realizzazione di un nuovo software ad hoc (strategie di sourcing).

## Capitolo 9

# Tecnologie a livello di piattaforma

### 9.1 Architetture di integrazione

Come descritto nei capitoli precedenti, un sistema informativo ha come elementi principali i dati e le applicazioni che li gestiscono. Il modo in cui le applicazioni gestiscono i dati è definito dalle regole aziendali (business rules) sono spesso rappresentate anch'esse come processi, che indicano l'insieme delle attività da svolgere per ottenere un determinato obiettivo.

Un sistema informativo può quindi essere visto come un insieme di moduli applicativi che realizzano le funzionalità che il sistema deve supportare. Come visto nella prospettiva architettonica (A) di BOAT, i moduli applicativi possono implementare componenti dell'architettura di back end (BES) o front end (FES). Tali moduli non sono indipendenti, ma comunicano tra di loro tramite scambio di messaggi.

I moduli applicativi utilizzati in un'azienda sono collegati tra loro da flussi informativi, utili per lo scambio di informazioni e per segnalare eventi, quali le eccezioni che si verificano nella gestione delle attività e che richiedono un intervento di tipo decisionale al di là delle normali regole di funzionamento dei processi legati alle attività. Nei diagrammi architettonici, questi collegamenti sono rappresentati in modo astratto da collegamenti tra i moduli dell'architettura. Come illustrato nella vista architettonica, infatti, le interazioni possono essere rappresentate a livello di Party-level e poi raffinate in architetture System-level.

Nel sistema informativo di un'organizzazione, esistono vari *moduli applicativi* il cui obiettivo è di gestire un insieme di attività strettamente correlate tra loro, oppure di gestire il ciclo di vita di un certo tipo di risorse. Esempi di insiemi di attività tra loro correlate sono le attività legate alla gestione dei clienti (Customer Relationship Management - CRM) tramite call center o tramite sportelli fisici e virtuali. Altro esempio sono le attività relative al marketing o alla catena di approvvigionamento (Supply Chain Management - SCM). Esempi di sistemi

focalizzati alla gestione delle risorse nell'azienda sono quelli relativi alla gestione delle risorse umane (Human Resources - HR) oppure alla gestione dei flussi finanziari.

Come visto nella descrizione delle architetture funzionali, i vari moduli applicativi che compongono un sistema si scambiano informazioni (indicate come messaggi nei diagrammi di rappresentazione delle architetture funzionali). La gestione delle attività avviene attivando diversi moduli software, i quali devono operare in modo armonico. Inoltre, l'organizzazione deve cooperare con i sistemi informativi di aziende esterne (clienti e fornitori, partner, ecc.). A causa dell'eterogeneità dei moduli, è necessario implementare *meccanismi di integrazione* per garantire uno scambio efficace di informazione tra moduli.

Le architetture di integrazione [41] sono soluzioni tecniche e organizzative volte a garantire il coordinamento delle attività intra-organizzative e inter-organizzative. Oggetto dell'integrazione sono sia i *dati* usati e creati dai diversi moduli del portafoglio applicativo sia i *processi* che coinvolgono diverse attività dell'organizzazione.

### 9.1.1 Integrazione dei dati

Come discusso nel caso del data warehouse e degli strumenti ETL, dati provenienti da sorgenti diverse possono avere caratteristiche che non li rendono fra loro compatibili. Per questo motivo, essi devono essere trasformati perché non è certo che i moduli applicativi, pur trattando gli stessi dati, li utilizzino rappresentati secondo lo stesso formato. Per quanto riguarda i flussi informativi, sia nei sistemi OLTP sia in quelli OLAP, i problemi di integrazione riguardano sia il *formato* sia la *semantica* dei dati.

Consideriamo innanzitutto il *formato* dei dati, che descrive come i dati vengono salvati e presentati. I problemi di integrazione relativi al formato dei dati riguardano:

- *Rappresentazione*: la stessa informazione può essere rappresentata diversamente da moduli diversi. Prendiamo come esempio un indirizzo: esso può essere rappresentato come un record <via/piazza, numero civico, CAP, città>, oppure come semplice stringa di testo contenente gli stessi elementi. La rappresentazione degli indirizzi può variare di Paese in Paese, ma anche all'interno dello stesso Paese (si pensi ad esempio alle varie denominazioni esistenti in Italia per indicare una via, quali campo, calle, ecc.). Pertanto, la rappresentazione mediante un'unica stringa consente di gestire gli indirizzi in modo più generale. Tuttavia, in questo caso, si pone il problema del riconoscimento dei vari campi del record all'interno dell'applicazione (ad esempio, di separare CAP e città).
- *Struttura*: si ha la stessa rappresentazione, ma i campi sono ordinati in modo diverso. Ad esempio, il record “indirizzo” potrebbe essere composto dagli stessi campi, ma in ordine diverso <CAP, città, via/piazza, numero civico>, oppure strutturato gerarchicamente, ad esempio <<CAP, città, >, <via/piazza, numero civico>>.

- *Presentazione:* lo stesso dato può essere presentato all'utente con formati diversi. Ad esempio, un campo “cognome” di 15 caratteri potrebbe essere non sufficientemente lungo per rappresentare tutti i possibili cognomi. Quindi, moduli applicativi diversi potranno presentare il cognome con campi di 15, 20, 30 caratteri. La corrispondenza tra le stringhe di sistemi informativi diversi non è garantita se il cognome supera la lunghezza minima prevista.

Un altro problema di integrazione si presenta se si esamina la *semantica dei dati*, che riguarda il significato attribuito all'informazione. Ad esempio, se ipotizziamo di gestire ordini e abbiamo un campo “prezzo”, possiamo associare ad esso un valore numerico. Nel caso di un sistema informativo italiano, questo valore può rappresentare un prezzo in euro. Tale interpretazione però non ha carattere generale. Infatti, se dobbiamo far colloquiare i sistemi informativi di un'azienda in Italia e un'azienda in Gran Bretagna, oltre al valore andrà anche indicata la valuta. Andranno inoltre fatte alcune ipotesi sulla modalità di conversione tra una valuta e l'altra: la conversione andrà fatta al momento dell'ordine o al momento del pagamento? da chi verrà fornito il tasso di cambio? Il problema sopra citato si potrà presentare non solo per quanto riguarda le valute, ma anche per le unità di misura considerate: sto ordinando 10 chili oppure 10 libbre di un prodotto?

L'integrazione tra sistemi informativi, sia tra aziende, come pure all'interno di aziende che operano in ambito internazionale, presenterà spesso problemi di questo tipo.

### 9.1.2 Integrazione dei processi

L'integrazione dei processi all'interno di un'organizzazione o tra organizzazioni richiede di definire quali siano le informazioni scambiate tra moduli applicativi distinti e le modalità di interazione tra processi: punti di sincronizzazione, messaggi ed eventi, interfacce, ordine dei messaggi.

Si consideri l'esempio in Figura 9.1 che riguarda l'interazione tra il sistema informativo di un cliente e quello di un fornitore. Un primo momento di sincronizzazione è la creazione dell'ordine: è il messaggio inviato dal cliente che crea nel fornitore un'istanza del processo di gestione di quell'ordine. Seguono altri messaggi di interazione legati al pagamento e alla spedizione della merce, che portano alla sincronizzazione dei due processi tramite scambi di informazioni. Non sarà solo necessario che le informazioni siano scambiate secondo un formato e una semantica compatibili, ma anche che l'ordine dei messaggi previsto da entrambi gli attori sia lo stesso. Infatti, qualora il cliente aspettasse la merce prima di effettuare il pagamento e nel frattempo il fornitore aspettasse il pagamento prima di inviare la merce, i due sistemi si scambierebbero gli stessi messaggi, ma si creerebbe una situazione di blocco critico che non consentirebbe di portare a termine l'esecuzione del processo.

Il caso discusso riguarda l'interazione tra aziende diverse; si verificano casi simili anche all'interno di una singola azienda, se l'azienda per gestire servizi e

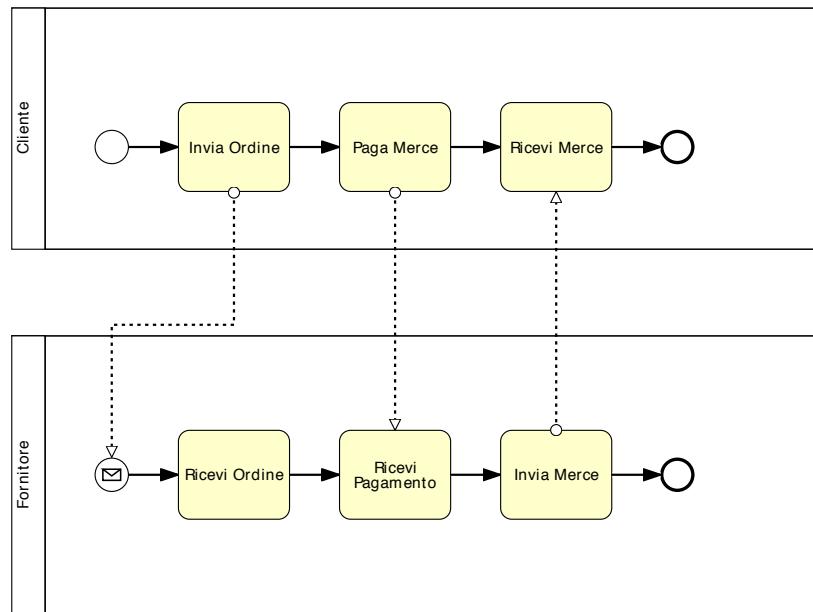


Figura 9.1: Integrazione dei processi

funzionalità diverse utilizza sistemi informativi o componenti architetturali diversi. Anche in tal caso è necessario avere meccanismi di condivisione o scambio di informazioni e di sincronizzazione dei processi.

## 9.2 Evoluzione dei moduli applicativi

Le architetture di integrazione hanno avuto un'evoluzione che ha seguito l'evoluzione tecnologica dell'informatica, sia dal punto di vista dei componenti hardware (disponibilità di sistemi dipartimentali e poi personali, prima desktop e poi portatili e smartphone), delle reti, locali e geografiche sia dal punto di vista delle piattaforme software (middleware) per la gestione dell'interconnessione tra sistemi. Le prime applicazioni venivano sviluppate senza alcun supporto esterno, utilizzando linguaggi vicini al linguaggio macchina (assembler o simili) e con codifica di funzionalità di base. Negli anni '70, i sistemi operativi cominciano a permettere di separare la logica applicativa dalla programmazione delle interfacce con l'hardware. Ogni applicazione è indipendente e usa dati dedicati salvati su file. 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. Nascono infine le interfacce grafiche per semplificare l'interazione dell'utente con l'applicazione, sostituendo le precedenti interfacce testuali.

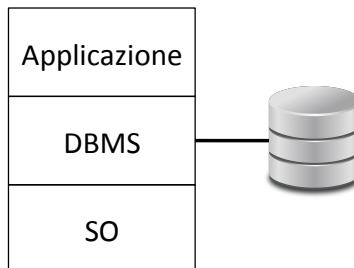


Figura 9.2: Elementi di un modulo applicativo

I singoli moduli applicativi possono quindi essere scomposti in elementi diversi, sia dal punto di vista funzionale che architettonico. Nella Figura 9.2, gli elementi dell'architettura funzionale di un sistema informativo sono composti a livello di piattaforma da un insieme di applicazioni, che consentono di fornire le funzionalità previste, e di un sistema per la gestione dei dati, solitamente gestiti tramite un DBMS (come illustrato nelle parlando di architetture funzionali). Le applicazioni e il DBMS utilizzano a loro volta le funzioni del Sistema Operativo (SO).

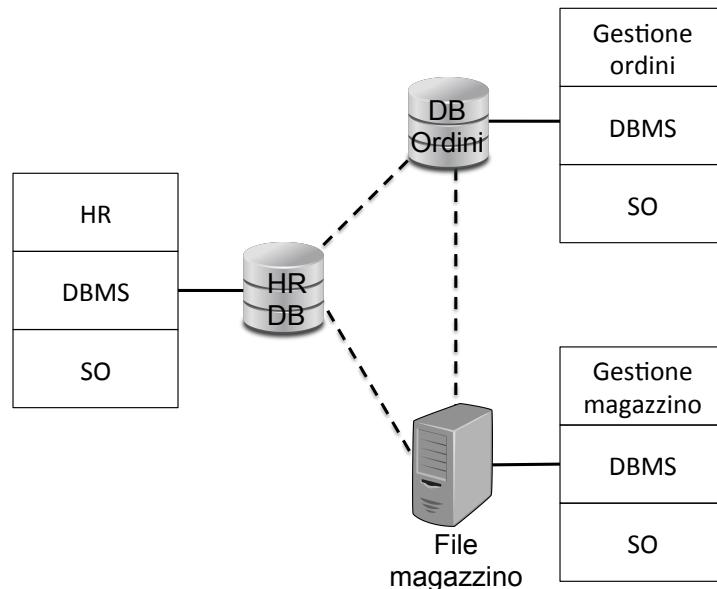


Figura 9.3: Moduli applicativi con dati replicati

Le prime applicazioni sono state sviluppate con una logica a silos, in cui ogni applicazione era vista come un componente indipendente. Le organizzazioni moderne usano un numero considerevole di applicazioni che interagiscono tra

loro e condividono dati. Una modifica nei dati da parte di un'applicazione deve riflettersi sulle altre. Nella Figura 9.3, si mostra il caso in cui i sistemi sono tra loro indipendenti, ma contengono dati replicati.

Ad esempio il sistema di gestione ordini e il sistema di gestione del magazzino contengono entrambi informazioni sulle merci ordinate. In un caso, le informazioni vengono viste dalla prospettiva della pratica amministrativa dell'ordine; dall'altro, si gestiscono informazioni relative alla consegna della merce. Lo scambio di informazioni tra questi settori è necessario, ad esempio per verificare che le merci pagate vengano consegnate e che quanto consegnato corrisponda a quanto è stato ordinato.

La soluzione illustrata in Figura 9.3 presenta diversi svantaggi, perché i dati replicati non sono sempre allineati, in quanto l'allineamento dei dati in genere viene fatto periodicamente. Ciò può portare all'incongruenza dei dati contenuti nelle diverse basi di dati e/o all'utilizzo di dati obsoleti.

Alcune suite software molto diffuse, come ad esempio la suite ERP, nascono per superare queste problematiche risolvendo il problema dell'integrazione con la condivisione di un'unica base di dati e la definizione intrinseca dei processi nella logica dei singoli moduli. L'onere dell'integrazione è quindi a livello di progetto di sistema e, vista la complessità del progetto e del sistema che ne deriva, si sono diffuse soluzioni che forniscono funzionalità tipiche dei sistemi informativi sviluppate in modo standard, basate su best practice, come nel caso dei sistemi ERP, la cui architettura descritta a livello di piattaforma, già discussa nella Sezione 8.2.4, è illustrata in Figura 9.4.

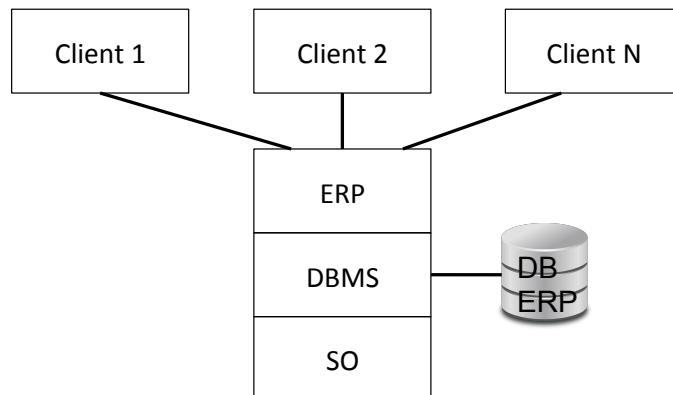


Figura 9.4: Architettura ERP

Come già visto, lo svantaggio di questi sistemi è una certa mancanza di flessibilità nella definizione dei processi aziendali. Inoltre, si verifica spesso il caso in cui l'azienda decide di utilizzare diversi tipi di sistemi con queste caratteristiche, ad esempio un ERP e un CRM, che devono essere integrati sia tra loro sia con altri componenti sviluppati internamente all'organizzazione. Pertanto, il tema dell'integrazione si pone nuovamente, ad un altro livello.

## 9.3 Piattaforme di integrazione

Le applicazioni che costituiscono il sistema informativo di un'organizzazione sono sviluppate indipendentemente, ognuna con i propri dati. I dati di applicazioni diverse hanno anche struttura e semantica differente, pur riferendosi allo stesso oggetto. Vi sono diverse architetture di integrazione per garantire una condivisione dei dati efficiente. Nel seguito si presentano le principali soluzioni, secondo l'approccio presentato in [41] a cui si rimanda per eventuali approfondimenti, facendo presente che tutti i tipi di integrazione discussi possono tuttora presentarsi nei singoli progetti di integrazione e che le scelte di integrazione saranno dettate da diversi fattori, sia tecnologici, ma anche organizzativi e progettuali.

### 9.3.1 Architettura di integrazione punto-a-punto

Una possibile soluzione al problema dell'integrazione tra applicazioni è la realizzazione di un'*architettura punto-a-punto*. Essa consiste nella realizzazione di un'interfaccia diretta tra i sistemi che si devono scambiare i dati. Esistono diverse soluzioni tecnologiche, i due casi principali vengono rappresentati in Figura 9.5 e in Figura 9.6.

In entrambi i casi, i sistemi vengono collegati tramite applicazioni di integrazione che diventano parte del sistema informativo e che, se necessario, forniscono anche i necessari meccanismi di trasformazione tra i dati per renderli compatibili tra più sistemi. Nella Figura 9.6, l'interazione è supportata da una piattaforma *middleware a messaggi* (*Message-Oriented Middleware - MOM*), che si occupa di gestire lo scambio dei messaggi tra i sistemi, gestendo aspetti generali quali la gestione delle code di messaggi, l'instradamento, la sicurezza.

Lo svantaggio di soluzioni di questo tipo è che la logica di integrazione viene cablata all'interno di ciascun sistema e diventa molto difficile sostituire uno dei componenti, sia per un aggiornamento sia per cambiare componente. Infatti, in questa soluzione, il numero di interfacce da implementare è dell'ordine  $N^2$ , dove  $N$  indica il numero di applicazioni coinvolte. La soluzione quindi non è scalabile, in quanto la modifica di un'applicazione comporta la modifica di  $N-1$  interfacce.

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

Si sono studiate soluzioni per evitare di collegare i sistemi direttamente come nel caso delle soluzioni punto-a-punto. Il paradigma *hub and spoke* (Figura 9.7) è basato su un hub centrale, a cui vengono collegati i sistemi tramite degli spoke, che fungono anche da adattatori. In questo modo i collegamenti non sono di tipo diretto. Il sistema che invia il messaggio non deve definire esplicitamente quale sistema riceverà il messaggio. Il messaggio invece viene inviato al sistema hub di integrazione. È chiaro che in questo modo i componenti possono essere sostituiti più facilmente. D'altra parte la logica applicativa di integrazione rimane comunque nascosta nell'implementazione dell'hub. In alcuni casi, gli

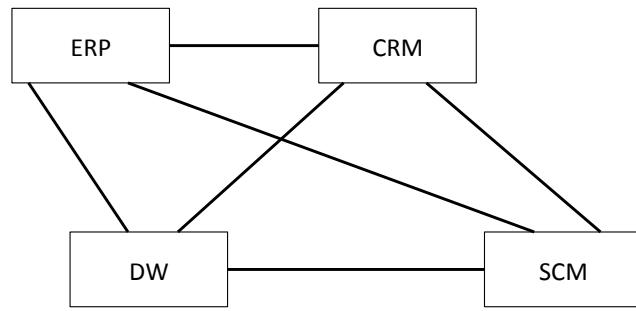


Figura 9.5: Integrazione punto-a-punto

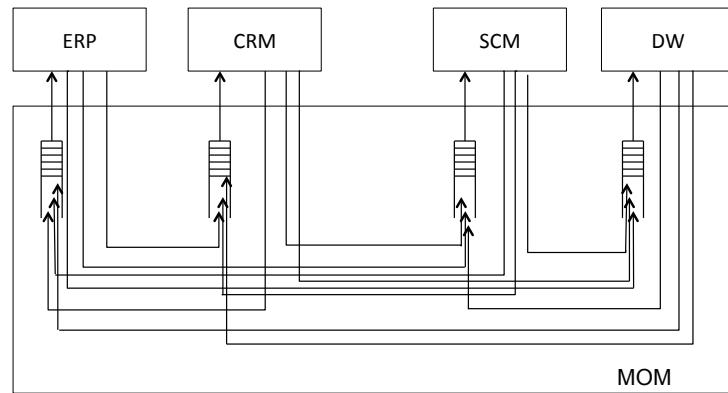


Figura 9.6: Integrazione con middleware a messaggi (rielaborazione da [41])

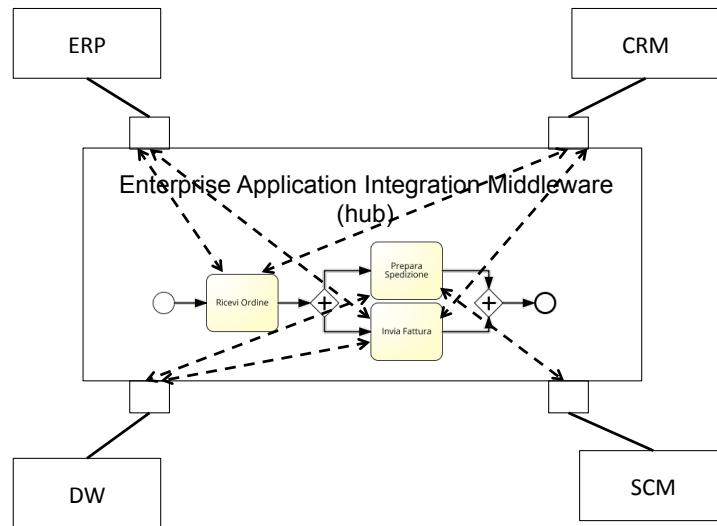


Figura 9.7: Integrazione con architettura hub and spoke (rielaborazione da [41])

hub contengono sistemi di automazione dei processi, che definiscono l'integrazione tramite processi di integrazione (come mostrato nella figura), facilitando la visibilità della logica applicativa.

Oppure, analogamente, la logica di integrazione può essere supportata da sistemi di middleware (Figura 9.8), in cui è possibile definire le regole di scambio dei messaggi tra i sistemi, ad esempio tramite meccanismi di publish and subscribe. Queste sono regole di tipo generale, quindi non definite per il singolo messaggio da istruire. Anche in questo caso, la logica di integrazione può essere poco visibile, perché nascosta nell'implementazione delle regole.

In questa soluzione, il numero di interfacce da implementare si riduce a N, cioè un'interfaccia per applicazione. Questo rende la soluzione più scalabile, ma la complessità rimane nell'implementazione degli adapter e dell'hub. Inoltre, la logica applicativa è nascosta nelle regole, ed eventuali modifiche risultano complesse.

### 9.3.3 Workflow component e workflow management system

Nell'architettura di integrazione hub and spoke, la realizzazione dell'integrazione comprende anche la definizione dell'ordine di esecuzione delle operazioni in modo implicito nelle regole. Qualora le applicazioni siano numerose e sia necessario coordinare le funzionalità fornite sulla base di processi formalizzati e soggetti a possibili cambiamenti nel tempo, è possibile utilizzare componenti a livello di piattaforma applicativa per dare un supporto all'automatizzazione dei processi, separata dalla logica applicativa delle applicazioni e non inserita al loro interno.

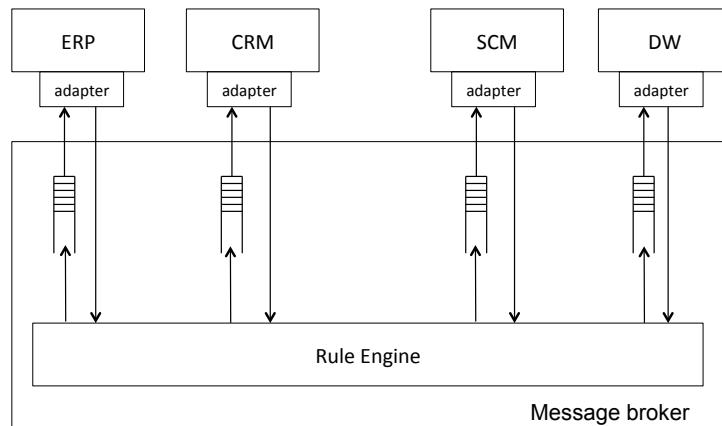


Figura 9.8: Integrazione con message broker (rielaborazione da [41])

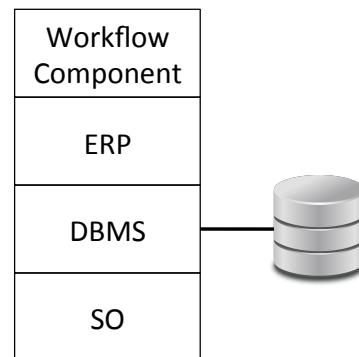


Figura 9.9: Integrazione mediante Workflow component

Un Business Process Management System (BPMS), detto anche Workflow component, consente, sulla base di una descrizione formalizzata dei processi, di gestire lo stato di avanzamento e i flussi informativi tra applicazioni. Il Workflow component è utile per semplificare le modifiche del modulo applicativo dovute a modifiche nei processi dell'organizzazione o alla loro riorganizzazione, poiché consente di separare le applicazioni dalla loro logica di composizione all'interno del modulo applicativo.

È possibile definire processi di integrazione ad hoc tramite un componente di automazione dei processi, che consente di gestire lo scambio di messaggi tra le varie funzionalità in modo integrato. Ad esempio, un BPMS (Workflow component) può fare parte di una suite ERP (nella Figura 9.9 si evidenzia il Workflow Component in un sistema ERP), rendendo possibile coordinare funzionalità fornite da moduli funzionali diversi, per gestire nuovi processi in modi non previsti all'interno della piattaforma.

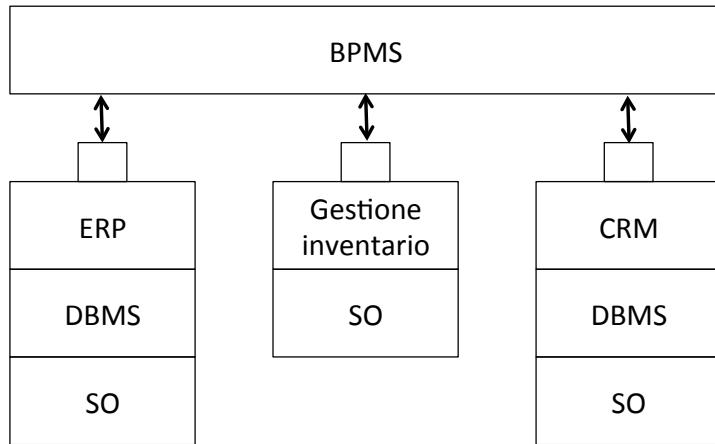


Figura 9.10: Integrazione mediante BPMS

Quando invece si tratta di integrare sistemi forniti da produttori diversi, si può ricorrere a un sistema di gestione di workflow come componente esterno, che implementa la logica di orchestrazione dello scambio dei messaggi al di fuori dei singoli componenti (Figura 9.10). In questo caso, il componente, denominato Workflow Management System (WFMS o Business Process Management System - BPMS), fornisce l'infrastruttura di integrazione, che è basata sull'esecuzione di una logica di integrazione predefinita in uno schema di processo (ad esempio, definita utilizzando BPMN).

I sistemi sopra illustrati consentono l'integrazione, ma non risolvono il problema dell'interfacciamento dei sistemi. In ogni caso, se si integrano sistemi diversi, è necessario fornire gli elementi adattatori che consentano ai sistemi di colloquiare tra loro.

## 9.4 Modelli a servizi

Un paradigma alternativo di progettazione software è quello realizzato tramite le architetture orientate ai servizi (Service Oriented Architecture - SOA). In questo paradigma, le funzionalità sono realizzate in modo modulare tramite servizi interrogravibili in remoto tramite protocollo internet HTTP.

Nelle SOA, il concetto di servizio è un aspetto fondamentale della prospettiva organizzativa. Un servizio è la realizzazione di un modulo di una funzionalità di business che può essere invocato dagli utenti (consumatori) del servizio, sia internamente che esternamente all'organizzazione. Il servizio quindi definisce una funzionalità di business, con interfaccia di interazione con gli altri componenti ben definita.

In [41] i servizi sono definiti come task computazionali con un alto livello di disaccoppiamento che comunicano attraverso la rete, la quale gioca un ruolo

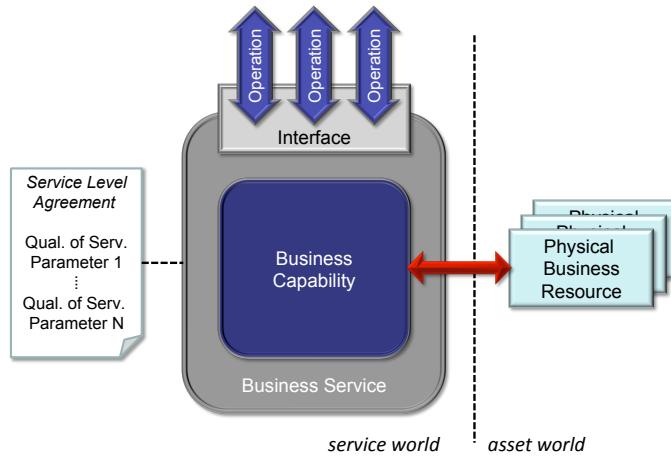


Figura 9.11: Il paradigma a servizi

| SLA<br>Business Service | Manage Shopping Cart |          |
|-------------------------|----------------------|----------|
| <b>Response Time</b>    | Maximum              | 0.25 sec |
|                         | Average              | 0.10 sec |
| <b>Availability</b>     | Minimum              | 99.99 %  |
| <b>Cost</b>             | Average              | € 0.02   |

Figura 9.12: Livelli di servizio

fondamentale nelle interazioni di tipo business-to-business. Per supportare i servizi, una SOA si focalizza su come i servizi sono descritti e organizzati per supportarne l'utilizzo in modo dinamico.

Un servizio, per sua natura, deve fornire una funzionalità che sia auto-contenuta e quindi indipendente rispetto agli altri servizi con cui si relaziona<sup>1</sup>. Inoltre, per i propri consumatori, un servizio è una black box di cui si conosce solo l'*interfaccia* per richiamarne le funzionalità. L'interfaccia definisce le *operazioni* attraverso cui il servizio può essere usato dai propri utenti. Le caratteristiche di un servizio sono mostrate in Figura 9.11 [19].

Per assicurarsi che un servizio sia in grado di fornire una certa funzionalità, ma anche che tale funzionalità sia fornita nel modo corretto, viene stabilito con i consumatori un accordo di qualità, chiamato *Service Level Agreement (SLA)*, che assume un'importanza fondamentale nel paradigma a servizi. Lo SLA definisce le caratteristiche non funzionali garantite dal servizio, in termini di parametri di *qualità del servizio (QoS)*. Ogni parametro di QoS specifica una o più metriche che possono essere utilizzate per monitorare l'esecuzione

<sup>1</sup><http://www.opengroup.org/standards/soa>

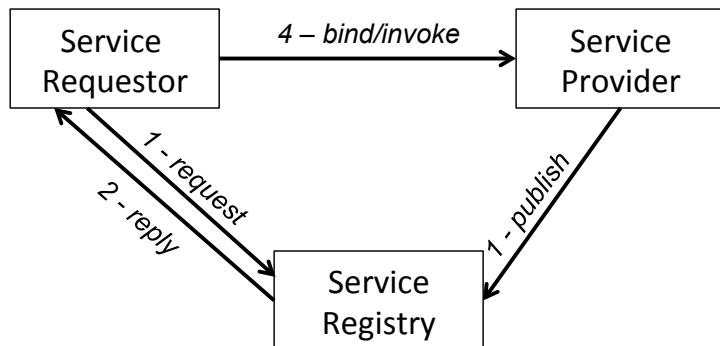


Figura 9.13: Service Oriented Architecture

del servizio e per accertarsi che il servizio risponda in modo valido. Esempi di parametri di QoS sono il tempo di risposta (tempo impiegato dal servizio interrogato per fornire una risposta), la disponibilità (la frazione di tempo in cui il servizio è disponibile) e il costo operativo per invocare il servizio. Un esempio di SLA è mostrato in forma di tabella in Figura 9.12, relativo ad un servizio di gestione di un carrello di acquisto. Nella tabella sono indicati tre parametri di qualità, per ognuno dei quali è possibile definire una o più metriche. Ad esempio, il tempo di risposta è misurato sia come massimo tempo trascorso dall'invio della richiesta sia come tempo medio.

Nella Figura 9.11 sono mostrate anche le risorse fisiche a disposizione del servizio (asset world). Queste risorse non sono visibili all'utilizzatore del servizio, che vede solo l'interfaccia e lo SLA e non le risorse fisiche utilizzate dal servizio, che possono anche essere variabili.

L'architettura a servizi SOA (Service Oriented Architecture) si è sviluppata anche grazie alla possibilità di scambiare messaggi su Internet tramite protocolli standard definiti sopra il protocollo HTTP e utilizzando la codifica XML (protocollo *SOAP*). Nella SOA (Figura 9.13), ciascun fornitore di servizio (*Service Provider*) pubblica le informazioni sui servizi offerti in un *Service Registry*, un registro dei servizi, descrivendo i servizi in termini di operazioni e messaggi scambiati. Tale descrizione è fatta usando il linguaggio standard *WSDL* (*Web Service Description Language*). Chi vuole interagire con il servizio (*Service Requestor*) dovrà quindi interrogare Service Registry per conoscere i servizi disponibili che realizzano le funzionalità richieste e dalla loro descrizione saprà che messaggi scambiare, quali protocolli utilizzare e come accedere al servizio. Il servizio viene quindi utilizzato direttamente da Service Requestor, collegandosi al servizio (bind) e invocando le operazioni fornite dal Service Provider (Figura 9.13). In questo modo vengono abilitate le interazioni di tipo dinamico e ultradinamico, in cui i partecipanti alle interazioni possono essere selezionati dinamicamente.

Oltre alla possibilità di invocare servizi utilizzando la descrizione dei servizi con il linguaggio WSDL e il protocollo SOAP, sono attualmente comuni altre

forme più leggere di realizzazione di sistemi a servizi, basate sull'uso diretto dell'invocazioni delle operazioni utilizzando ad esempio il protocollo HTTP.

Nello stile architetturale di realizzazione di servizi denominato *REST - REpresentational State Transfer*, si parla di *RESTful services*, quando le richieste vengono effettuate direttamente a una risorsa URI (tramite il suo Uniform Resource Identifier) e si parla di sistemi *REST web based*, in cui la risposta viene fornita in un formato predefinito (XML, HTML, JSON o altri formati). Quando viene utilizzato il protocollo HTTP, che è il più comune, le operazioni disponibili comprendono quelle predefinite dai metodi HTTP (GET, POST, PUT, DELETE, ecc.). Questo modo di invocare i servizi presenta vantaggi in termini di performance, modificabilità e semplicità. Rimangono i requisiti già indicati relativi alla necessità di definire i formati e le modalità di scambio di informazioni tra sistemi che consentano una effettiva integrazione.

#### 9.4.1 Composizione e orchestrazione di servizi

Per soddisfare le necessità dei clienti in uno scenario basato sui servizi, è necessario ricorrere alle funzionalità offerte da diversi servizi di business. Per questo motivo, servizi diversi possono essere usati in modo congiunto per fornire tutte le funzionalità richieste da una singola applicazione software. Ad esempio, per realizzare un'applicazione di e-shopping, è necessario interrogare un servizio per gestire il carrello, un servizio per la ricerca nel catalogo, un altro servizio per il pagamento, e così via. Per fornire la funzionalità richiesta, tutti questi servizi devono essere composti in un servizio di business più complesso. Le architetture di tipo SOA si concentrano infatti più che sulla modularizzazione del software, su come integrare componenti software indipendenti e distribuiti. Questo è possibile facendo ricorso a tecnologie e standard che permettono la comunicazione e cooperazione tra questi componenti tramite la rete internet.

Comporre servizi consiste nel gestire questi servizi come elementi di un'orchestrazione di servizi. Una *service orchestration* determina la sequenza con cui i singoli servizi sono invocati, in modo simile a quanto descritto nel caso dell'orchestrazione dei processi di business. Mentre un processo di business descrive l'ordine con cui le funzioni di business sono eseguite nel tempo, l'orchestrazione dei servizi descrive l'ordine con cui i servizi sono invocati, usando una rappresentazione simile a quella introdotta per i processi (per esempio, BPMN).

#### 9.4.2 Selezione dinamica dei servizi di business

Nella collaborazione tra organizzazioni in un contesto orientato ai servizi, la flessibilità delle collaborazioni è molto rilevante in quanto permette di rispondere alle richieste di clienti diversi che presentano requisiti fra loro molto differenti. La flessibilità è anche necessaria per reagire ai cambiamenti del mercato che, nel caso dei servizi, può essere molto dinamico.

La flessibilità può essere ottenuta usando una rete dinamica di servizi di business. In tale rete, un servizio completo viene creato componendo in modo dinamico i servizi forniti dalle varie organizzazioni della rete. Solitamente, un

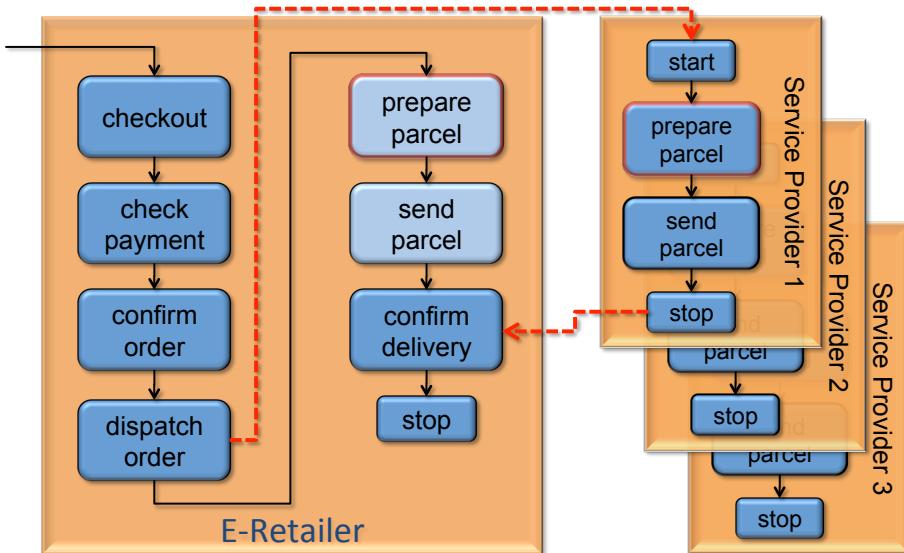


Figura 9.14: Invocazione dinamica di servizi

partecipante assume il ruolo di orchestratore, prendendosi l'onere di realizzare l'orchestrazione effettiva del servizio complesso. L'orchestrazione richiede infatti la responsabilità centralizzata per la sincronizzazione dei servizi coinvolti. Nel caso in cui la responsabilità della sincronizzazione sia distribuita tra i vari partecipanti, si parla di coreografia di servizi, in cui la sincronizzazione è gestita con la modalità peer-to-peer.

Come già introdotto e illustrato in Figura 9.11, in una rete dinamica di servizi, gli elementi che compongono il servizio complesso sono selezionati al momento, usando il service registry, in base alle funzionalità offerte e in base allo SLA. Per facilitare questa operazione, nel mercato possono essere presenti alcuni Service Broker, ovvero organizzazioni che si occupano di trovare il migliore allineamento tra servizi offerti e servizi richiesti. Ad esempio, come illustrato nella Figura 9.14, un sistema può invocare dinamicamente servizi forniti da diversi fornitori, scegliendo di volta in volta il fornitore da utilizzare, oppure sostituendo un fornitore con un altro quando necessario (ad esempio, nel caso in cui non si riceva una risposta a una richiesta entro un tempo prefissato).

#### 9.4.3 Integrazione dei servizi

Il paradigma a servizi facilita l'integrazione definendo a priori i possibili messaggi da scambiare. Ciascun componente definisce una propria interfaccia, che specifica quali servizi sono offerti dal sistema e quali messaggi debbano essere scambiati per utilizzarli, sia in ingresso sia in uscita. Le applicazioni di integrazione possono quindi essere costruite utilizzando tale paradigma come base.

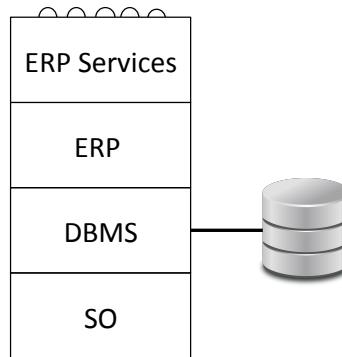


Figura 9.15: Funzionalità come servizi in un sistema ERP

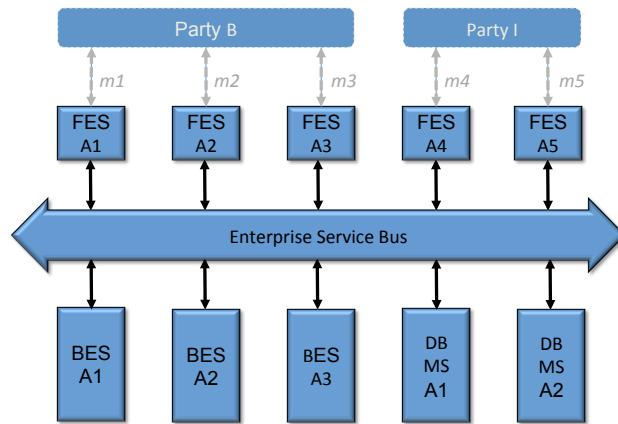


Figura 9.16: Architetture a servizi con ESB - Party-level

Si possono delineare diversi scenari. Innanzitutto i sistemi esistenti possono esporre proprie funzionalità in forma di servizi, come illustrato in Figura 9.15, approccio che ha caratterizzato l'evoluzione dei sistemi ERP negli anni 2000. L'integrazione dei componenti può essere realizzata utilizzando infrastrutture di integrazione basate sui servizi, tipiche dei sistemi a Enterprise Service Bus (ESB), che consentono di gestire l'integrazione di servizi in un'azienda, utilizzando servizi aggiuntivi di registry per la ricerca dei servizi da utilizzare e collegare. La Figura 9.16 illustra la notazione utilizzata per rappresentare un ESB in un'architettura Party-level a servizi, in cui i moduli dell'architettura funzionale non comunicano direttamente, ma tramite un Enterprise Service Bus (ESB).

I componenti di automazione di workflow possono infine essere utilizzati per realizzare la logica applicativa di integrazione, tutti con accesso ad un unico bus

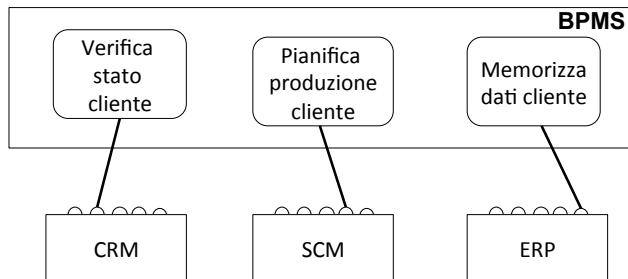


Figura 9.17: Composizione di servizi

di integrazione (Figura 9.17) .

I diversi paradigmi di integrazione possano convivere in un sistema informativo. Un processo di business può essere supportato da un sistema di workflow che, per svolgere alcune attività, utilizza servizi forniti dall'integrazione di altri componenti, sia utilizzando sistemi di integrazione di applicazioni basati su middleware (a volte indicati come Enterprise Application Integration nei sistemi informativi) sia mediante accesso diretto a servizi tramite la loro interfaccia.

I diversi livelli di rappresentazione di un sistema verranno discussi ulteriormente nel capitolo dedicato ad ArchiMate.

## 9.5 Domande

- Illustrare le architetture di integrazione dei sistemi informativi, discutendo l'evoluzione dall'integrazione point-to-point, discutendo vantaggi e svantaggi.
- Punti di forza delle architetture distribuite dei Sistemi Informativi: parametri di valutazione.
- Architetture di integrazione di componenti software: differenze tra approcci a messaggi e sistemi basati su servizi.
- Definire layer e tier e illustrare, discutendole, le possibili configurazioni a due livelli.
- Allocazione dei layer sui tier: discussione delle possibili configurazioni di una architettura a tre livelli.
- Spiegare brevemente quali sono gli approcci più usati per l'integrazione di applicazioni eterogenee in un sistema informativo aziendale.
- Illustrare le problematiche di integrazione dei dati e dei processi e descrivere una possibile soluzione tecnologica (o piattaforma di integrazione) tra quelle studiate.

- Architetture d'integrazione di componenti software: differenze tra approcci a messaggi e sistemi basati su servizi.

# Capitolo 10

## Tecnologie a livello di architettura fisica

### 10.1 Descrizione

In questo capitolo ci concentreremo sull'organizzazione dell'hardware e del software per gestire la risorsa informazione. Come vedremo, a seconda delle esigenze informative dell'organizzazione che si sta considerando, l'architettura del sistema informativo può cambiare sensibilmente e, in particolare, ci occuperemo di descrivere le possibili relazioni che possono intercorrere tra hardware e software. Questo capitolo presenta alcuni aspetti legati alle possibili alternative per la realizzazione di architetture fisiche su cui allocare i componenti software illustrati in precedenza.

#### 10.1.1 Evoluzione delle architetture fisiche

Nell'ambito dei sistemi informativi, il termine architettura indica l'insieme delle scelte tecniche e organizzative che influiscono sullo sviluppo e sull'utilizzo delle risorse tecnologiche di tipo IT.

In prima battuta, è utile considerare due macro-categorie di architetture a livello di architettura fisica: *centralizzate* e *distribuite*. I sistemi informativi centralizzati, basati su un'architettura centralizzata, prevedono la presenza di un solo nodo di elaborazione su cui è installato tutto il software necessario all'elaborazione dell'informazione. Questa architettura rappresenta la tipica soluzione dei primi sistemi informativi, dove le reti avevano diffusione limitata, e dove il costo per l'acquisto e la manutenzione di un singolo nodo era piuttosto elevato. Collegati al nodo centrale vi erano uno o più terminali: nodi di elaborazione con capacità molto limitata, spesso ridotta alla presenza di una periferica di input (ad esempio, tastiera) e una di output (ad esempio, schermo o stampante). Grazie a connessioni di rete dirette verso il nodo centrale, i terminali potevano

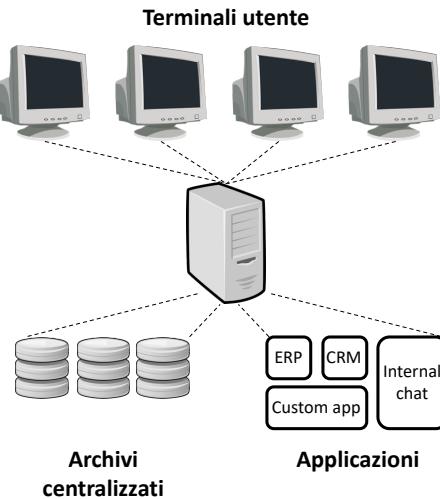


Figura 10.1: Sistema informatico centralizzato

interagire con il software installato, senza però aver la possibilità di eseguire alcuna applicazione in locale.

In particolare, si parla di *sistema informatico centralizzato* (o semplicemente *sistema centralizzato*) quando i dati e le applicazioni risiedono in un unico nodo elaborativo (Figura 10.1).

I sistemi centralizzati sono nati con l'informatica moderna negli anni '50 e si sono sviluppati negli anni '60 e '70 grazie all'evoluzione dei mainframe, all'introduzione dei sistemi operativi time-sharing, ed allo sviluppo dei sistemi di gestione di basi di dati centralizzati. La nascita e lo sviluppo negli anni '70 e '80 di nuove tecnologie più economiche, sia nell'hardware sia nelle strutture di gestione dati versatili e facili da usare (mini e micro elaboratori, personal computer, reti locali di comunicazione, architetture client/server, interfacce utente di tipo grafico, modelli evoluti per la strutturazione dei dati) ha portato alla crisi del modello centralizzato e ha promosso la realizzazione di sistemi basati su architetture distribuite (Figura 10.2) le quali, per essere definite tali, richiedono che almeno una delle seguenti due condizioni sia verificata:

- Le applicazioni, fra loro cooperanti, risiedono su più nodi elaborativi (elaborazione distribuita).
- Il patrimonio informativo, unitario, è ospitato su più nodi elaborativi (base di dati distribuita).

In termini generali, un sistema distribuito è costituito da: (i) uno o più nodi di elaborazione che, al contrario dei terminali del sistema centralizzato, possono eseguire anche applicazioni in locale, e (ii) una o più di applicazioni logicamente

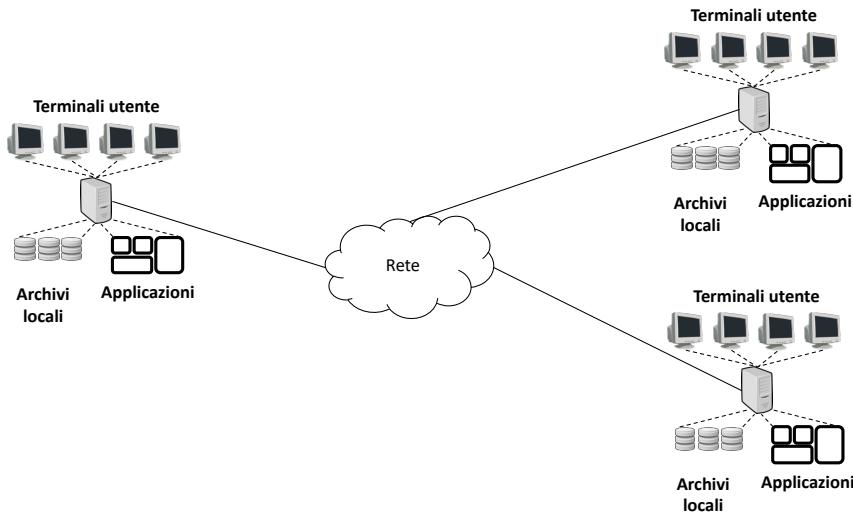


Figura 10.2: Sistema informatico distribuito

indipendenti che collaborano per il perseguimento di obiettivi comuni attraverso un'infrastruttura di comunicazione.

Ciò nonostante, nei primi anni '90 il modello distribuito è stato sottoposto a forte critica per le maggiori complessità progettuali e di gestione. Questo problema si è ulteriormente acuito quando internet è stato ampiamente utilizzato per permettere la comunicazione tra i vari nodi. Con l'obiettivo di risolvere questi limiti, sono state sviluppate architetture più complesse basate sulla distribuzione di livelli logici, legati alla struttura delle applicazioni, su livelli fisici, legati alla struttura fisica dei sistemi di elaborazione. Come illustrato nelle sezioni successive, tali architetture hanno tenuto conto della possibilità che un'applicazione - nel momento in cui è disponibile su internet - può trovarsi a dover rispondere a un ampio numero di utenti.

## 10.2 Layer applicativi

L'architettura di un'applicazione può essere molto complessa. È possibile però, in generale, strutturare le applicazioni secondo tre livelli logici, in letteratura spesso indicati come *layer*, che trovano corrispondenza in altrettanti strati software nei quali vengono partizionate le applicazioni:

- Il livello *presentazione* (*P*) si occupa di gestire la logica di presentazione, quindi le modalità di interazione con l'utente: contiene le modalità di interfacciamento (grafico) e di rendering delle informazioni. Questo livello è anche denominato a volte *front end* dell'applicazione ed è accessibile

dall'utente attraverso una GUI (Graphical User Interface) o, come più comunemente detto, una interfaccia grafica o User Interface (UI).

- Il livello della *logica applicativa* (*A*) o logica di business si occupa delle funzioni da mettere a disposizione all'utente.
- Il livello di logica di *accesso ai dati* (*D*) si occupa della gestione dell'informazione, eventualmente con accesso ai database, o a sistemi legacy (con *sistema legacy* si intende un sistema ereditato dal passato sviluppato con tecnologie di vecchia generazione, ad esempio un'applicazione COBOL, che tuttavia è ancora molto importante per l'organizzazione), o anche a risorse informative disponibili in rete.

I livelli di logica applicativa e di accesso ai dati costituiscono il *back end* dell'applicazione e, al contrario della logica di presentazione, sono accessibili attraverso metodi basati su API (Application Programming Interface). A differenza delle GUI, che sono pensate per un utente umano, le API sono rivolte ad altri software. In questo modo, è possibile integrare fra loro applicazioni diverse. Volendo fare un paragone con i tipici linguaggi di programmazione (e.g., C), un'API per la logica applicativa si mostra come un elenco di signature<sup>1</sup> di metodi. Riguardo invece al livello di accesso ai dati, le API possono anche essere sotto forma di metodi che ricevono in ingresso una query SQL e restituiscono il risultato di tale query.

In linea generale, sono quindi messe a disposizione le informazioni legate all'input della funzione che il programma chiamante dovrà fornire, e l'output della funzione che il programma chiamante dovrà essere in grado di interpretare. Come già detto, essendo ormai i sistemi messi a disposizione su internet, una API si mostra all'esterno sottoforma di metodi che possono essere chiamati, potenzialmente, da qualunque applicazione connessa (v. anche la Sezione 9.4 sulle interfacce a servizi).

### 10.3 Applicazioni distribuite

Nei sistemi distribuiti, i livelli logici software, o layer, – appena presentati – vengono installati su livelli hardware (detti *tier*), dove un livello rappresenta una macchina (o un insieme di macchine), ognuna con la propria capacità elaborativa. A seconda di come avviene la distribuzione di layer sui tier, un'applicazione può essere configurata in vari modi.

La prima architettura prevede un singolo livello di distribuzione ed è detta *Single Tiered*. In questo caso, i tre livelli software sono assegnati ad un'unica macchina. Questa configurazione è quella classica terminale-host che caratterizzava i sistemi basati su mainframe. Pur rappresentando un'architettura centralizzata, spesso associata ad un approccio ormai obsoleto, essa mantiene ancora una propria validità. Infatti, i sistemi mainframe, Single Tiered per eccellenza,

---

<sup>1</sup>la signature di un metodo è la sua firma composta dall'identificatore del metodo e dalla lista dei tipi dei parametri che riceve in input

garantiscono un alto livello di prestazioni, affidabilità e sicurezza. Oltre a ciò, la manutenzione dell'architettura risulta molto semplice essendo limitata ad un singolo nodo. Per contro, una architettura di questo tipo risulta poco flessibile. Infatti, l'applicazione installata è di tipo monolitico e un upgrade di uno dei livelli ha un impatto diretto sugli altri. Inoltre, la necessità di migrare verso interfacce utente di tipo grafico ha mostrato tutti i limiti di questa configurazione poiché richiederebbe che l'elaborazione grafica – attività abbastanza onerosa in termini di computazione – avvenisse, per ogni terminale, direttamente sul mainframe.

Una prima evoluzione dell'architettura prevede due livelli di distribuzione ed è detta *Two-Tiered*. Qui i livelli logici sono divisi tra due nodi di elaborazione. Questa configurazione è stata implementata nella prima generazione di sistemi client/server negli anni '80, ed è possibile identificare sei differenti configurazioni (Figura 10.3), in base alle modalità di allocazione dei tre layer logici: i layer di Presentazione (P), di Gestione della logica Applicativa (A) e di Gestione Dati (D). Le configurazioni differiscono in termini di: traffico di rete, prestazioni e compiti eseguiti dalla stazione di lavoro e comportano livelli diversi di flessibilità e di facilità di modifica. A seconda delle funzionalità assegnate al nodo client, si parla di configurazioni con *thin client* se il nodo client gestisce la sola logica di presentazione. Un esempio è dato da client in cui è presente un Browser Web (il Client Web) che non richiede significativa potenza di calcolo in quanto si limita a interpretare e mostrare pagine HTML. Questo tipo di soluzione, che ha il vantaggio di funzionare anche con risorse limitate, richiede una connessione stabile tra lato client e lato server per poter rispondere a tutte le richieste degli utenti.

Si parla invece di *thick client* (anche chiamati *fat client*) quando, oltre alla logica di presentazione, è presente anche parte (o l'intera) logica applicativa. Questo richiede che il nodo in questione sia sufficientemente potente per poter eseguire sia funzioni di visualizzazione che di calcolo. Per fare un esempio, si pensi alle app per smartphone. In questo caso, si è in presenza di un thick client in quanto l'app ha in locale tutti gli strumenti per soddisfare le richieste dell'utente e richiede l'interazione con il server solo per qualche specifica funzionalità.

Sebbene la presenza di thick client non richieda una connessione stabile tra client e server, ma limitata ai soli momenti in cui la funzionalità o i dati richiesti non risiedono sul nodo client, essi hanno introdotto alcuni svantaggi che hanno origine nella necessità di amministrare diverse macchine client anziché un singolo server, come avveniva per i sistemi centralizzati. Se un'applicazione viene aggiornata frequentemente, in configurazioni thick gli aggiornamenti devono essere effettuati su tutte le macchine client e ciò può comportare elevati costi di gestione.

Attualmente si sono diffusi sistemi di supporto all'aggiornamento automatico delle applicazioni lato client che rendono più gestibili la distribuzione e gli aggiornamenti del software e questo facilita la diffusione di moduli applicativi su client quali le App.

Riprendendo le varie configurazioni delle architetture Two-Tiered mostrate in Figura 10.3, le configurazioni su due livelli dette “Presentazione distribuita” e “Presentazione remota” corrispondono a thin client, mentre le restanti configurazioni a thick client.

Una terza architettura prevede tre livelli di distribuzione (*Three-Tiered*): i tre livelli applicativi sono suddivisi, in linea di principio, fra altrettante macchine dedicate, ovvero: una stazione di lavoro utente, un server intermedio (detto anche middle tier) e un server di gestione dei dati. La Figura 10.4 mostra la possibile ripartizione dei tre layer architetturali su tre tier:

- Le configurazioni da 1 a 5 riguardano situazioni *thick client*, maggiormente indicate per realtà interne all’organizzazione (sistemi Intranet).
- Le configurazioni da 6 a 13 riguardano situazioni *thin client*, adatte alla distribuzione di servizi a un insieme aperto di utenti (sistemi Internet);
- Le configurazioni 4, 5, 7, 8 e da 10 a 13 realizzano la separazione fisica fra utenti (*tier 1*) e dati (ospitati unicamente nel *tier 3*), tramite l’introduzione di una macchina intermedia (*tier 2*). Tali configurazioni sono quindi maggiormente sicure riguardo al controllo dell’accesso ai dati da parte degli utenti.

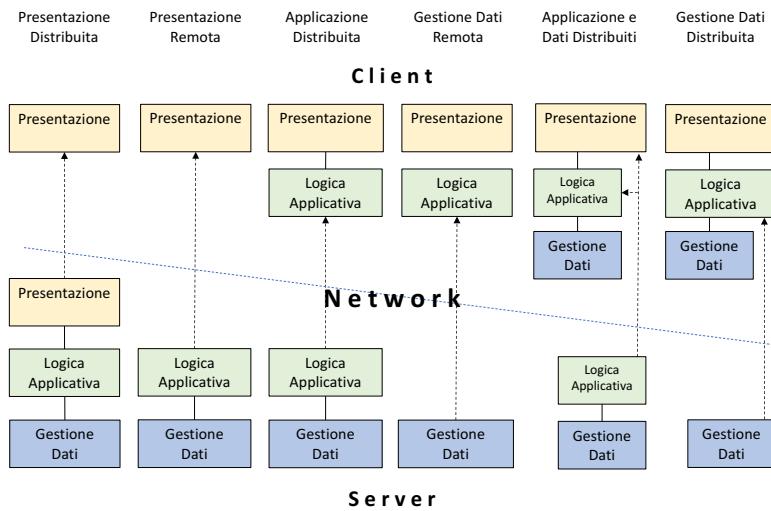


Figura 10.3: Classificazione delle architetture a 2 tier

| <b>Config.<br/>/Tier</b> | <b>1</b>  | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b>  | <b>10</b> | <b>11</b> | <b>12</b> | <b>13</b> |
|--------------------------|-----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
| <i>Tier<br/>3</i>        | D         | D        | D        | D        | D,A      | D        | D        | D,A      | D         | D         | D,A       | D,A       | D,A,<br>P |
| <i>Tier<br/>2</i>        | D         | D        | D,A      | A        | A        | D,A      | A        | A        | D,A,<br>P | A,P       | A,P       | P         | P         |
| <i>Tier<br/>1</i>        | D,A,<br>P | A,P      | A,P      | A,P      | A,P      | P        | P        | P        | P         | P         | P         | P         | P         |

Figura 10.4: Classificazione delle architetture a 3-tier

Un sistema a tre livelli (tier) conferisce alle infrastrutture IT maggiori caratteristiche di scalabilità e flessibilità. Le configurazioni thick a tre livelli non sono particolarmente indicate se la logica applicativa distribuita è particolarmente onerosa dal punto di vista della richiesta di potenza di calcolo (CPU o RAM). In queste situazioni, è più conveniente centralizzare la logica applicativa su un server intermedio. In questo modo è sufficiente garantire la robustezza e l'efficienza del server intermedio, che viene condiviso tra i diversi client. Inoltre, il server viene dimensionato in modo da garantire un prefissato livello di prestazioni per un certo numero di utenti contemporaneamente attivi, numero che è ragionevole assumere sia sempre inferiore al numero totale degli utenti che hanno accesso al sistema informativo. Il server intermedio consente inoltre di ridurre il carico al DBMS server. Il middle tier infatti mantiene connessioni permanenti con il DBMS. Inoltre, il numero di sessioni aperte tra il tier intermedio e DBMS è in generale inferiore al numero di connessioni aperte tra i client ed il tier intermedio.

Un esempio di sistema 3-tier (configurazione 7) è presentato nella Figura 10.5, in cui ciascun layer è allocato su un tier separato.

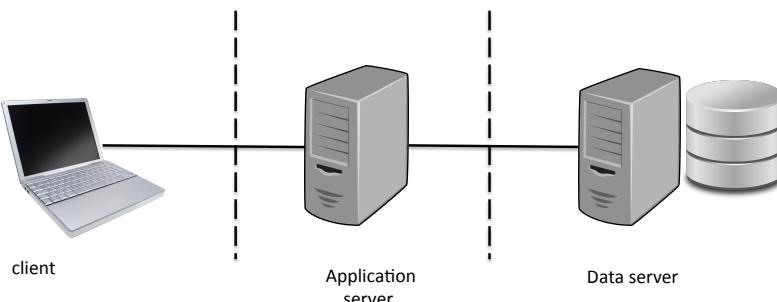


Figura 10.5: Esempio di architettura Three-Tiered (configurazione 7)

Mentre il passaggio tra architetture Single-Tiered a Two-Tiered ha permesso di alleggerire il carico del mainframe spostando parte della potenza di calcolo sui nodi client, che nel frattempo si erano diffusi grazie al calo dei costi, l'architettura Three-Tiered punta a ottimizzare il carico lato server distribuendolo su due nodi. Negli ultimi anni, questo tipo di tendenza si è sviluppata ulteriormente.

te proponendo architetture in cui i layer applicativi possono essere distribuiti su un numero  $n$  di tier (*n-tiered architecture*) generico, dove per diverse funzionalità sono previste diverse macchine dedicate: ad esempio, gateway server, authentication server, transaction server, database server, e così via.

I sistemi informativi basati su Web spesso si basano su 5 tier. In particolare, la presentazione viene realizzata su 3 tier: il browser dell'utente, il Web server che garantisce l'accesso alle pagine e agli oggetti statici e lo script engine che si occupa della generazione delle pagine dinamiche. I tier 4 e 5 corrispondono all'application server e al DBMS. Un esempio di architettura a 5 livelli per un sistema informativo basato su Web è presentato nella Figura 10.6.

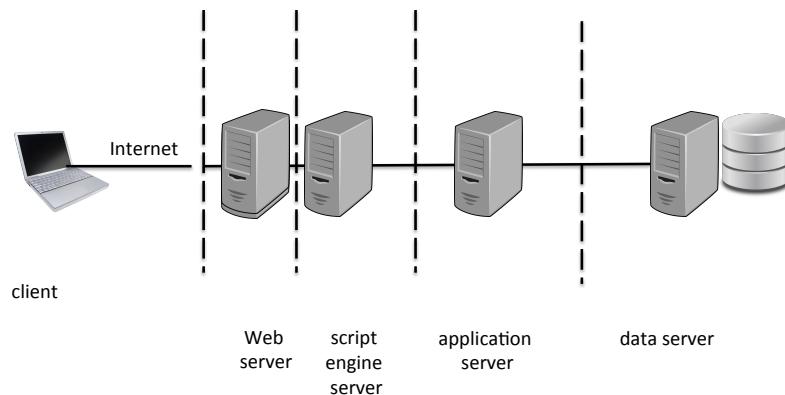


Figura 10.6: Esempio di sistema 5-tier per SI basati su Web

Il numero di tier inoltre può crescere se vengono introdotti server dedicati a supporto dello strato di comunicazione, ovvero del middleware. Per esempio, quando più sorgenti informative sono integrate tra loro e consultate in tempo reale, è possibile introdurre un database gateway tra le sorgenti informative e l'application server (si veda Figura 10.7). L'application server esegue le fun-

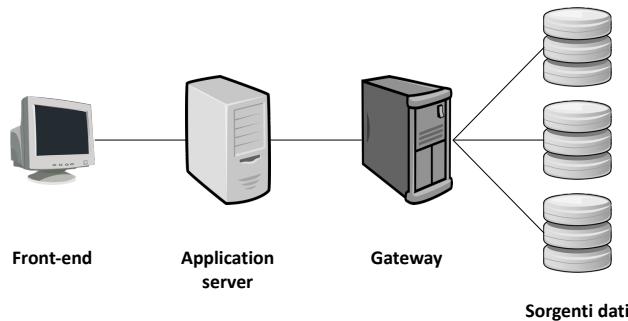


Figura 10.7: Esempio di sistema 4-tier

zionalità applicative basandosi su una visione unificata dei dati. Il database

gateway fornisce un accesso trasparente ai DBMS della rete e traduce le chiamate API dell'application server nella sintassi della sorgente dei dati remota. Ciò evita che lo sviluppatore debba conoscere i dettagli dei database remoti e delle relative modalità e linguaggi di accesso. Questa trasparenza della localizzazione comporta inoltre la trasparenza rispetto a problemi di *migrazione*: se una sorgente di dati viene spostata, o convertita in una nuova struttura, è necessario modificare soltanto il database gateway senza modificare l'application server o la logica di front end del sistema informativo. È necessario osservare che, anche se si introduce un maggior numero di tier fisici, gli strati logici (*layer*) software è pari a 3 in quanto, ad esempio, i database gateway vengono considerati solitamente strato di logica applicativa.

### 10.3.1 Architetture per componenti applicativi: ERP

Vediamo ora come le architetture viste sopra possono portare a realizzazioni diverse nel caso di utilizzo di un sistema ERP.

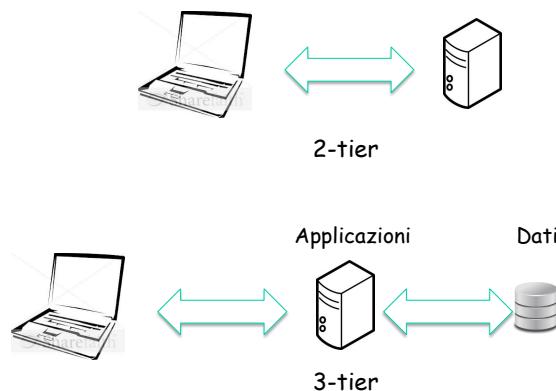


Figura 10.8: ERP in modalità client-server

Nella Figura 10.8 vediamo il caso in cui il sistema ERP viene utilizzato in modalità client-server, dove il client è il modulo dedicato con cui l'utente interagisce con il sistema ERP. Nell'architettura 2-tier, il client risiede su un tier e la logica applicativa e il database risiedono sul server, mentre in quella 3-tier il client interagisce con l'application server che a sua volta si interfaccia con il database server.

L'architettura in Figura 10.9 mostra invece il caso in cui l'utente utilizza un browser come client e l'interazione con il sistema ERP è effettuata tramite un web server.

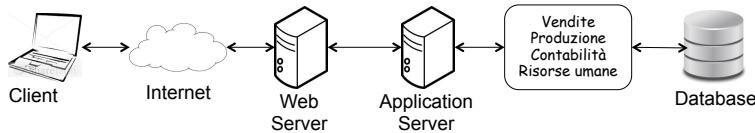


Figura 10.9: Architettura per ERP a 5 livelli con server web

## 10.4 Scalabilità

Con *scalabilità* di una architettura IT si intende la capacità dell'infrastruttura di soddisfare richieste crescenti da parte degli utenti (o di un maggiore numero di utenti) con aggiornamenti adeguati.

In linea generale possiamo identificare due tipi di scalabilità:

- *Scalabilità verticale* (scale-up): l'aumento di prestazioni si ottiene mantenendo inalterato il numero di nodi di elaborazione, ma aumentandone la loro capacità elaborativa. Ad esempio, un server viene reso più performante con aggiunta di CPU, RAM o capacità di memoria di massa. In questo caso, l'operazione di scaling è abbastanza semplice da realizzare in quanto non altera la distribuzione delle applicazioni. Per contro, non sempre ad un aumento della capacità elaborativa corrisponde un aumento proporzionale delle prestazioni (ad esempio, raddoppiare la CPU non significa automaticamente raddoppiare il numero di processi eseguibili), oltre al fatto che esistono limiti fisici oltre i quali non è possibile andare (ad esempio, il numero di slot disponibili per alloggiare le CPU e/o le memorie sono limitati). Questo porta a situazioni in cui l'architettura non è scalabile oltre un certo livello.
- *Scalabilità orizzontale* (scale-out): l'aumento di capacità avviene attraverso una replicazione dei nodi che costituiscono l'eventuale collo di bottiglia. In tal modo, i processi che inizialmente il singolo nodo non era in grado di gestire sono distribuiti tra i nodi replica. In questo caso, l'upgrade del sistema è più complesso in quanto impatta sulla struttura dei tier. Infatti, l'aggiunta di nodi di replica richiede l'introduzione di un sistema di *load balancing* che si occuperà poi di distribuire il carico di elaborazione tra tali nodi. Il vantaggio risiede nella possibilità di sfruttare il principio di downsizing, che permette di avere una scalabilità a costi contenuti. Proposto all'inizio degli anni '80 e considerato tuttora valido, il principio del downsizing afferma che server di fascia bassa hanno costi inferiori a server di fascia alta o mainframe, a parità di potenza di calcolo complessiva installata.

Indipendentemente dall'approccio seguito per rendere un sistema scalabile, è sempre difficile dimensionare correttamente il sistema. Nel caso della scalabilità verticale, ad esempio, dimensionare significa capire quanta memoria aggiuntiva

installare, mentre, nel caso della scalabilità orizzontale, capire quanti nodi di replica introdurre.

Il problema si manifesta sia in caso di sovradimensionamento che sottodimensionamento. In caso di sovradimensionamento, l'upgrade dell'architettura potrebbe portare ad esborsi economici non richiesti, in quanto le risorse aggiunte potranno non essere utilizzate visto il non corrispondente aumento degli utenti. Al contrario, un sottodimensionamento potrebbe rendere l'upgrade non sufficiente in poco tempo.

Soprattutto nel caso di applicazioni Web, il problema del dimensionamento non è di facile soluzione in quanto il numero degli utenti è una variabile dipendente dal tempo, spesso poco predicibile. Si immagini, per esempio, un sito di e-commerce specifico per l'Italia. Supponendo che durante il giorno il numero degli utenti si attesti su un numero costante (con qualche picco durante alcune ore del giorno), il dimensionamento del sistema considererà questo numero quale valore da prendere in esame per capire la richiesta di risorse. Questo, però, non solo porta ad uno spreco di risorse durante la notte (quando pochi utenti utilizzano il sistema) ma anche a un sottodimensionamento in particolari periodi dell'anno, ad esempio prima delle feste natalizie, dove il numero di utenti potrebbe crescere ulteriormente. A tal proposito, soprattutto in ambito Web, non si parla di scalabilità ma di *elasticità* del sistema: la proprietà di un sistema non solo di aumentare le proprie risorse a fronte di picchi di richieste, ma anche di diminuirle quando le richieste vengono meno. L'elasticità, che vedremo essere uno dei cardini delle soluzioni basate su Cloud Computing, potrebbe essere vista come una scalabilità di tipo orizzontale reversibile, dove sono previste sia l'aggiunta di nodi, sia la loro riduzione.

#### 10.4.1 Server Farm

Seguendo l'approccio di tipo scale-out, i tier fisici possono essere realizzati anche come *server farm*, ovvero come un insieme di elaboratori che condividono il carico elaborativo, le applicazioni e, a seconda delle configurazioni, i dati. Una server farm è vista dagli altri livelli come un'unica risorsa.

I sistemi basati su server farm risultano molto scalabili poiché è possibile far fronte ad un incremento dei requisiti introducendo ulteriori macchine senza essere soggetti ai vincoli di upgrade di una singola macchina e, per il principio del downsizing, con costi contenuti rispetto alle potenzialità, dal punto di vista del numero di utenti che si possono servire.

Le server farm possono essere realizzate secondo due principi progettuali: *Cloning* (clonazione) e *Partitioning* (partizionamento).

Si parla di *clonazione* quando sui server vengono installate le stesse applicazioni software ed i medesimi dati. Le richieste sono poi instradate ai vari elementi appartenenti all'insieme dei cloni usando un sistema di load-balancing. Un insieme di cloni dedicati allo svolgimento di un particolare servizio è detto *RACS* (*Reliable Array of Cloned Services*) e ogni clone implementa lo stesso insieme di funzionalità. La clonazione del servizio permette di ottenere vantaggi sia di scalabilità sia di tolleranza ai guasti (fault tolerance) in termini di disponibilità.

nibilità del servizio stesso. Infatti se un clone subisce un guasto, un altro nodo può continuare ad erogare il servizio bloccato. I RACS si possono presentare in due configurazioni: shared-nothing e shared-disk a seconda che si condivida lo storage o meno.

Nella prima configurazione (*shared-nothing*, si veda la Figura 10.10): i dati memorizzati sono replicati su ogni clone e risiedono in un disco fisso locale ad ogni clone. Quindi, un aggiornamento di dati deve essere applicato ad ognuno dei cloni. Di conseguenza, questa configurazione risulta problematica per i servizi “write-intensive”, mentre presenta prestazioni migliori per applicazioni di tipo “read-only”, quali l’accesso a pagine statiche o il download di file o immagini dai Web server.

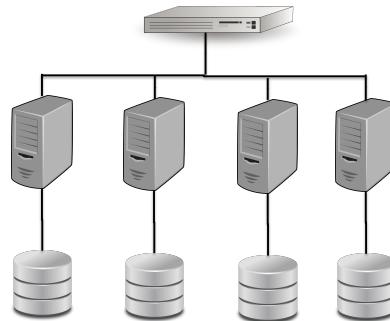


Figura 10.10: Configurazione shared-nothing

Nella configurazione *shared-disk*, spesso chiamata *cluster*, i cloni condividono un server di memorizzazione che gestisce i dischi fissi (Figura 10.11).

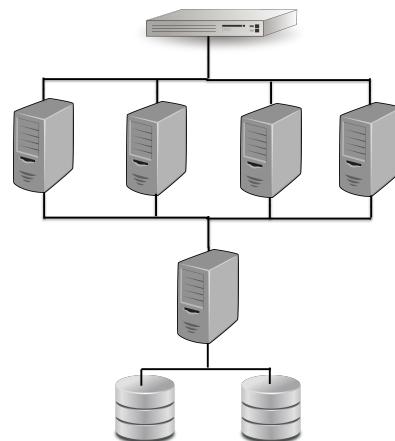


Figura 10.11: Configurazione shared-disk o cluster

La tecnica di *partizionamento* prevede la duplicazione dell'hardware, per ripartire l'esecuzione dell'applicazione tra i nodi (Figura 10.12), partizionando anche i dati. Ogni nodo svolge quindi una funzione specializzata; ad esempio il sistema Web di vendite di un'azienda può essere suddiviso per tipologie di clienti o per linee di prodotto ed ognuna è gestita da un nodo.

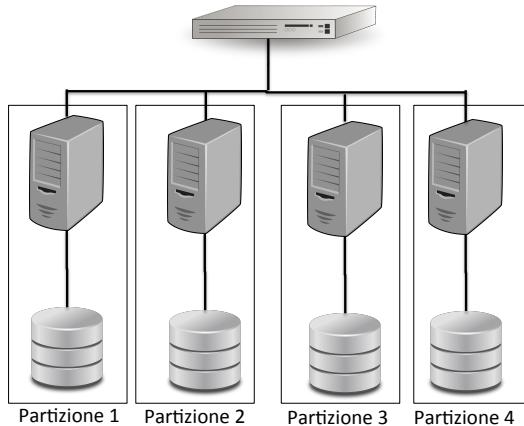


Figura 10.12: Partizionamento

Il *partizionamento* è completamente trasparente alle applicazioni e le richieste vengono inviate alla partizione che possiede i dati rilevanti. Ad esempio, se viene implementato un partizionamento per linee di prodotto (A, B, ...), le richieste di accesso alla linea di prodotto A verranno indirizzate al server che è in grado di accedere ai dati della linea di prodotto A. Questa tecnica però non migliora la disponibilità poiché i dati sono memorizzati su un singolo server e in caso di guasto la parte di servizio da esso gestita non risulta più accessibile. Questa caratteristica è nota come proprietà di *graceful degradation* (degrado parziale) dei sistemi distribuiti: a differenza dei sistemi centralizzati, in caso di malfunzionamento non tutto il sistema risulta inaccessibile, ma solo alcune funzionalità non risultano più disponibili. Per risolvere il problema di indisponibilità di alcune funzionalità applicative in caso di guasto, si impiega spesso la clonazione dei singoli server che costituiscono la partizione, creando in tale modo dei *pack*. Si parla allora di *RAPS* (*Reliable Array of Partitioned Service*), una soluzione che garantisce sia scalabilità che disponibilità del servizio (Figura 10.13).

Un esempio di architettura a 5 livelli in server farm per un sistema informativo basato su Web è presentato nella Figura 10.14. I web server e gli script engine sono in genere clonati, mentre l'application server è partizionato o partizionato e clonato, a seconda delle applicazioni.

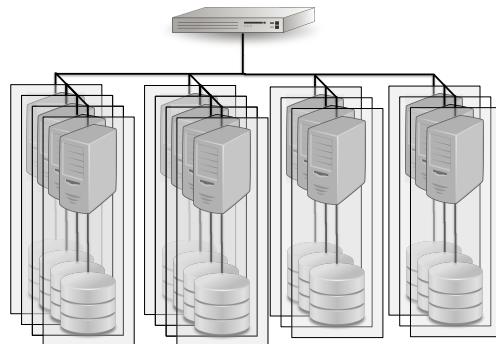


Figura 10.13: Architettura ridondata e partizionata

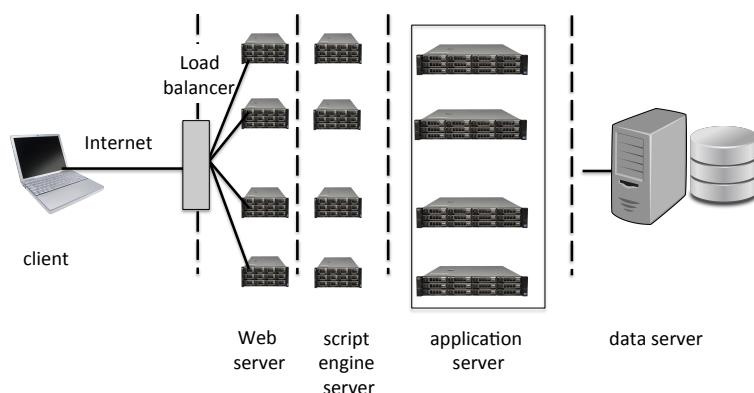


Figura 10.14: Esempio di sistema 5-tier per SI basati su web

## 10.5 Virtualizzazione

La realizzazione di server farm ha permesso, come descritto, di realizzare infrastrutture in grado di sopportare il carico computazionale richiesto nel momento in cui il numero di utenti, complice anche il fatto che ormai le applicazioni sono spesso pensate per essere accedute da Web, aumenti ad un valore tale da pregiudicare le performance dell'applicazione stessa. Va, però, sottolineato come la realizzazione di server farm risulti essere una attività spesso economicamente dispendiosa: sia dal punto di vista degli investimenti necessari per acquistare e installare nuove risorse, sia dal punto di vista dell'esercizio, in quanto la manutenzione e la gestione di tali risorse risulta essere proporzionale all'aumentare del numero di macchine da gestire. Ciò ha limitato quindi la possibilità di offrire applicazioni Web alle sole organizzazioni in grado di sopportare tali costi, creando grandi centri di calcolo (data center) e escludendo di fatto tutte le organizzazioni medio-piccole.

Al contempo, però ha preso nuovo slancio la tecnologia software che passa sotto il nome di *virtualizzazione*, sviluppata sin dagli anni '60, e che permette di astrarre le componenti fisiche di un elaboratore (ad esempio, CPU, memoria, disco) al fine di renderle disponibili in modalità software. Una CPU virtuale è un componente software che, agli occhi dell'utilizzatore, fornisce capacità di elaborazione come una CPU normale (fisica). La differenza sta nel fatto che chiedere l'esecuzione, ad esempio, di un calcolo ad una CPU virtuale significa eseguire una chiamata ad una applicazione (la CPU virtuale), la quale trasformerà la chiamata alla CPU fisica ad essa associata. Questa indirezione, se da un lato porta ad un degrado delle prestazioni, ha diversi vantaggi:

- È possibile disaccoppiare le risorse fisiche dalla applicazione che le usa. Quindi, mantenendo inalterata la CPU virtuale, una modifica della CPU fisica è totalmente trasparente all'applicazione.
- È possibile frazionare una risorsa fisica in diverse risorse virtuali. Ad esempio la stessa memoria fisica può essere suddivisa in  $n$  memorie virtuali, ognuna a disposizione di una singola applicazione.
- È possibile condividere la stessa risorsa fisica tra diverse risorse virtuali pur dando l'impressione all'applicazione di avere l'uso esclusivo di una parte delle risorse fisiche. Per esempio, la stessa CPU può essere acceduta contemporaneamente da diverse CPU virtuali. Dal punto di vista dell'applicazione, essa continua ad avere l'impressione che la CPU sia a suo uso esclusivo. In realtà, altre CPU virtuali, quindi altre applicazioni, vi accedono in modo concorrente. Ciò permette di ottimizzare l'uso delle risorse.

L'insieme delle risorse virtuali necessarie al funzionamento di una applicazione è detta *macchina virtuale (VM)* che, per essere eseguita, necessita di una macchina fisica in grado di fornire le risorse reali. La corrispondenza tra risorse reali e virtuali è gestita da uno strato software detto *hypervisor* o *virtual machine monitor* (vedi Figura 10.15) il quale si può installare direttamente sulla

macchina fisica (in questo caso si parla di installazione *bare metal*) oppure sopra il sistema operativo della macchina fisica (in questo caso il sistema operativo è detto sistema operativo dell'*host*). Tra gli hypervisor più utilizzati citiamo: XEN, VMWare, VirtualBox. L'obiettivo dell'hypervisor, indipendentemente dalla sua configurazione, è quello di rendere possibile l'esecuzione di una o più macchine virtuali offrendo alle applicazioni installate sopra di esse, le risorse virtuali a loro assegnate.

Essendo la macchina virtuale, dal punto di vista di un'applicazione, una macchina a tutti gli effetti, su di essa è possibile installare a sua volta un sistema operativo (detto sistema operativo *guest*) che può anche differire dal sistema operativo *host*. In questo modo, con un hypervisor in modalità *hosted*, su una macchina su cui è installato, ad esempio, Mac OS X quale sistema operativo, è possibile eseguire una macchina virtuale su cui è installato un sistema operativo quale Linux o Windows.

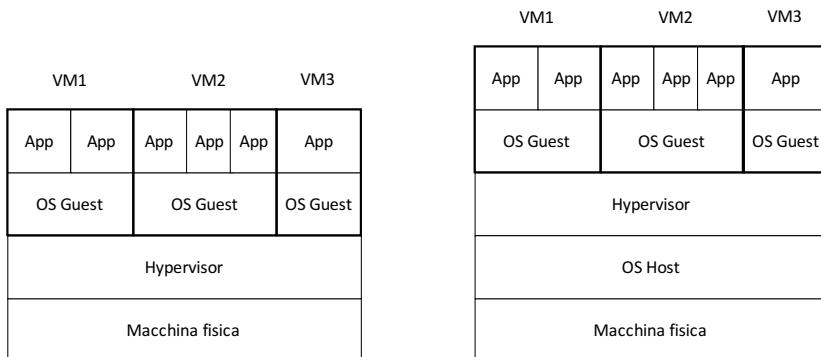


Figura 10.15: Hypervisor in modalità bare-metal (sinistra) e hosted (destra)

La virtualizzazione ha offerto, negli ultimi anni, la possibilità di migliorare e ottimizzare la gestione delle risorse soprattutto nelle server farm. Infatti, uno dei problemi delle server farm, e dei data center in generale, risiede nello spreco di risorse nel momento in cui le richieste, per una data applicazione, risultano – anche solo temporaneamente – al di sotto della capacità dell'architettura. Questo tipo di spreco è evidente nel momento in cui, ad ogni applicazione, è riservata una specifica macchina fisica. Infatti, statisticamente, l'utilizzo medio di un server è di circa 5-10% rispetto alle sue potenzialità. Al fine di ovviare a questo tipo di situazione, la virtualizzazione ha permesso di attuare politiche di *server consolidation*. Ciò significa che, data una macchina fisica, attraverso l'installazione di un hypervisor, su di essa sono installate diverse macchine virtuali corrispondenti a quelle che erano prima delle macchine fisiche permettendo, quindi, di avere diverse applicazioni installate e quindi usando al meglio le risorse fisiche a disposizione.

In questo scenario, la virtualizzazione offre uno strumento ulteriore. Infatti, non essendo una macchina virtuale legata ad una specifica macchina fisica, essa può migrare da macchina fisica a macchina fisica (VM migration). Al momento,

le tecniche di migrazione più utilizzate sono quelle *a freddo* che richiedono lo spegnimento della macchina, lo spostamento, e la sua successiva riaccensione. Al contempo, si stanno sviluppando soluzioni *a caldo* dove lo spostamento della macchina virtuale può avvenire anche durante l'esecuzione dell'applicazione.

## 10.6 Cloud Computing

Il cambio di paradigma nell'apertura dei sistemi attraverso i servizi e le SOA, unitamente alla flessibilità offerta dalla virtualizzazione, ha generato terreno fertile per il Cloud Computing. Questo viene definito dal NIST, l'istituto nazionale americano di Standard e Tecnologia, come: “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”.

Il Cloud Computing è quindi un paradigma per la realizzazione di architetture applicative che, poggiando su sistemi virtualizzati, possono accedere a tali risorse da qualunque dispositivo connesso alla rete da qualunque località. La particolarità di tali risorse è la possibilità di richiederle *on demand*, quindi solo quando servono. Sarà poi la piattaforma Cloud a offrire i servizi per metterle a disposizione quando necessario e a rilasciarle quando non più utili. Grazie alle sue caratteristiche, il Cloud Computing permette di gestire, riducendone i costi, applicazioni caratterizzate da un carico elevato di richieste con un'alta variabilità nel tempo e che gestiscono una quantità rilevante di dati.

Il Cloud Computing si basa sul concetto di *utility computing*, in cui le risorse informatiche vengono viste dagli utenti come una fornitura simile a quella elettrica. L'utente paga in base all'utilizzo che fa delle risorse e (come nel caso dell'elettricità) ha l'illusione che le risorse informatiche siano infinite. Questa illusione è possibile perché i gestori di piattaforme Cloud sfruttano al meglio le risorse disponibili combinando tra loro applicazioni con profili di carico computazionale molto diversi. È infatti altamente improbabile che in un'infrastruttura che ospita tante applicazioni eterogenee tutte le applicazioni abbiano un momento di picco di richieste contemporaneamente.

Entrando nel dettaglio, il paradigma Cloud Computing si compone di;

- Cinque caratteristiche essenziali.
- Tre modelli di fornitura del servizio Cloud.
- Quattro modelli di deployment.

Ognuno di questi elementi è descritto in maggior dettaglio nei prossimi tre paragrafi.

### 10.6.1 Caratteristiche del Cloud Computing

L'istituto nazionale americano di Standard e Tecnologia (NIST) definisce il Cloud Computing come un sistema caratterizzato dalle seguenti caratteristiche:

- *On-demand self-service*: gli utenti possono ottenere risorse come tempo di esecuzione o spazio di memorizzazione in modo autonomo.
- *Accesso alla rete onnipresente*: gli utenti possono usare reti standard e dispositivi internet per accedere alle risorse Cloud.
- *Accesso alle risorse indipendente dalla loro localizzazione*: le risorse sono suddivise in più server e l'utente di solito non sa con precisione dove queste sono localizzate.
- *Elasticity*: le risorse possono essere rapidamente assegnate, aumentate o ridotte per soddisfare le richieste degli utenti.
- *Monitoraggio sull'utilizzo*: il costo delle risorse Cloud è commisurato sulla base delle risorse effettivamente utilizzate.

### 10.6.2 Modelli di servizio

I tre modelli base di servizio (di erogazione) di un sistema basato su Cloud Computing sono:

**Infrastructure as a Service (IaaS)** Il gestore della piattaforma Cloud mette a disposizione degli utenti risorse di tipo fisico (server, ma anche storage, fino ai dispositivi di rete). Questi vengono visti dagli utilizzatori come elementi virtuali, che possono essere usati per ospitare le applicazioni degli utenti installate all'interno di macchine virtuali (VM) che definiscono la quantità di risorse fisiche richieste. Le risorse sono virtuali perché l'utente non ha consapevolezza della macchina fisica in cui le proprie VM sono ospitate. Sulle macchine virtuali, gli utenti possono installare il proprio software di piattaforma e le applicazioni. Essendo le VM controllate e gestite dagli utenti, essi avranno la possibilità di configurarle e quindi gestirle in modo flessibile, acquisendo e rilasciando dinamicamente risorse computazionali. Tra i fornitori di risorse Cloud secondo il modello IaaS si cita Amazon AWS EC2 (condivisione di risorse computazionali) e S3 (fornitura di risorse di storage) e Microsoft Azure Virtual Machines.

**Platform as a Service (PaaS)** Secondo questo modello di fornitura, il Cloud provider mette a disposizione degli utenti, oltre alle risorse fisiche, anche alcune tecnologie a livello di piattaforma. Ad esempio, il fornitore può fornire servizi di DBMS in Cloud. Vengono fornite delle API per utilizzare il servizio e l'utente si prende carico dell'implementazione delle proprie applicazioni. L'utente ha il controllo della sola applicazione ma può configurare alcuni parametri relativi alle tecnologie di piattaforma. Esempi di PaaS sono IBM Bluemix (ora IBM Cloud), Amazon AWS Simple DB e Microsoft Azure Cloud Services.

**Software as a Service (SaaS)** Quando si parla di SaaS, il Cloud provider, oltre all'infrastruttura e alle tecnologie di piattaforma, fornisce agli utenti veri e propri applicativi. Il cliente non ha quindi nessun controllo sul servizio, ma può

configurare alcuni parametri relativi all'applicazione. Questi sistemi permettono di accedere alla stessa applicazione da diversi dispositivi condividendone i dati, e permettono inoltre la condivisione anche con altri utenti. Esempi molto diffusi di SaaS sono Gmail, Google Drive e Dropbox.

**everything as a Service (\*aaS)** Generalizzando il paradigma della virtualizzazione, molti altri componenti a livello applicativo, business e fisico vengono forniti come servizi. Si parla ad esempio di DaaS (Data as a Service), BPaaS (Business Process as a Service), e così via.

### 10.6.3 Modelli di deployment

Vengono definiti quattro diversi tipi di deployment:

- *Private Cloud*: L'infrastruttura è fornita ad uso esclusivo di una singola organizzazione e la stessa può essere fornita e gestita dall'organizzazione stessa o da altre organizzazioni. Questo modello di deployment gode di un alto livello di sicurezza e personalizzazione. Il suo impiego però comporta costi maggiori e una minore scalabilità.
- *Community Cloud*: L'infrastruttura è fornita ad uso di una comunità di organizzazioni che condividono un obiettivo o un'esigenza comune. Anche in questo caso, l'infrastruttura può essere fornita e gestita dall'organizzazione stessa o da altre organizzazioni. Il livello di sicurezza e personalizzazione è ancora abbastanza elevato e i problemi relativi ai costi e alla scalabilità, seppur presenti sono in parte mitigati.
- *Public Cloud*: l'infrastruttura è fornita ad uso pubblico da un fornitore privato o pubblico. Il servizio può essere offerto sia gratuitamente che a pagamento. In questo caso, i costi per usare il servizio sono molto ridotti e l'architettura è molto scalabile, ma il livello di sicurezza e personalizzazione è sensibilmente inferiore rispetto ai modelli precedenti.
- *Hybrid Cloud*: L'infrastruttura è realizzata componendo due o più infrastrutture Cloud dei modelli precedenti (public, private, community) che collaborano pur rimanendo entità distinte per fornire un servizio ai clienti. Questo modello gode dei benefici dovuti all'uso di modelli di deployment multipli. Per poter collaborare, le infrastrutture Cloud coinvolte devono utilizzare tecnologie standard in modo condiviso.

### 10.6.4 Componenti architetturali BOAT in Cloud

La Figura 10.16 illustra un esempio di componenti definiti a livello di piattaforma (BES, DBMS) allocati in Cloud, con accesso tramite un Enterprise Service Bus (ESB).

L'ERP come tutte le applicazioni aziendali può essere gestito internamente (on-premises) o esternamente (in outsourcing).

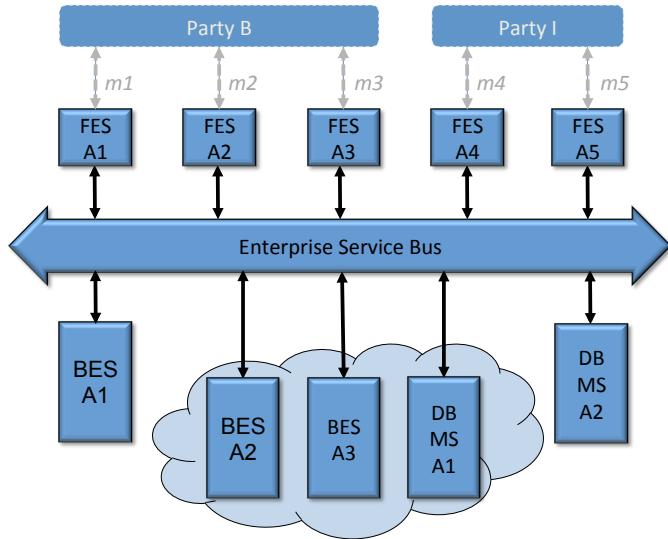


Figura 10.16: Architetture a servizi in Cloud

Il Cloud offre anche la fruizione di ERP SaaS, in cui il fornitore ERP mette a disposizione una versione del proprio software in modalità SaaS (in BOAT rappresentabile come in Figura 10.17 contrapposta alla gestione on premises già vista in Figura 9.4).

## 10.7 Opzioni di gestione dei SI

Come discusso in precedenza nella Sezione 3.3, in fase di pianificazione è necessario operare scelte sia sulle modalità di acquisizione e sviluppo dei componenti applicativi (scelta Make or Buy) sia sulla gestione dell'infrastruttura fisica, che può essere interna o esterna all'organizzazione (in-house, outsourcing).

Focalizzandoci sulle opportunità di gestione dell'infrastruttura fisica, nella Figura 10.18 vengono elencati e descritti i possibili livelli di servizi di outsourcing distinguendo diverse opzioni:

- Livello 0: le organizzazioni non hanno nessun componente gestito fuori dall'azienda e si occupano della gestione dei propri sistemi *In house*.
- Livello 1: denominato anche *servizi condivisi*, questo livello si riferisce al caso in cui l'organizzazione è supportata da una azienda IT che lavora in modo dedicato per l'organizzazione stessa. Quindi si può dire che l'organizzazione gestisce ancora internamente i propri sistemi. Questo è una opzione molto utilizzata dalle banche in cui le diverse filiali sono supportate da una azienda IT che offre servizi in modo esclusivo a tutto il gruppo.

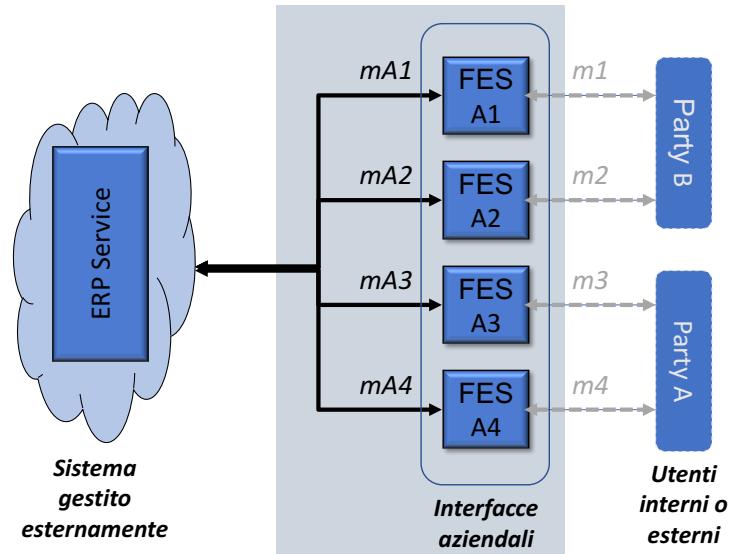


Figura 10.17: Rappresentazione ERP gestito in Cloud (SaaS ERP)

- Livello 2: le organizzazioni che non hanno competenze IT possono anche scegliere come opzione quella del *Supporto esterno* in cui la gestione IT si svolge internamente all'azienda, ma con il supporto di un'organizzazione esterna (per esempio, una società di consulenza).
- Livello 3: a questo livello, alcune componenti del sistema iniziano a spostarsi fuori dall'organizzazione. L'opzione *Consorzio* permette alle organizzazioni di condividere l'uso di alcune risorse per dividere i costi dell'investimento. Di solito questa opzione mette in condivisione strutture molto costose quali call center o data center.
- Livello 4: l' *Outsourcing selettivo* è un outsourcing parziale in cui si trasferisce all'esterno dell'organizzazione la gestione di alcune componenti del sistema informativo. Spesso, le aziende decidono di spostare all'esterno solo applicazioni che non risultano critiche per il core business aziendale (per esempio, Outsourcing di analisi di mercato, strumenti di simulazione o di elaborazioni complesse).
- Livello 5: l' *Outsourcing completo* riguarda invece lo spostamento dell'intera gestione del sistema informativo all'esterno dell'azienda.

L'adozione delle diverse opzioni dipende spesso dalla grandezza dell'organizzazione e dalla presenza di competenze IT all'interno della stessa. Per esempio, le aziende di grandi dimensioni gestiscono i propri sistemi utilizzando le opzioni presenti ai livelli 0, 1, 2 e 3. Le organizzazioni di medie dimensioni utilizza-

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

Figura 10.18: Opzioni di gestione dei sistemi informativi

no anche l'Outsourcing selettivo e in alcuni casi anche quello completo, mentre quelle di piccole dimensioni utilizzano prevalentemente l'Outsourcing completo.

Le organizzazioni che forniscono servizi di Outsourcing vengono denominate *Service Provider* e tradizionalmente, dal punto di vista dell'infrastruttura fisica, il loro supporto esterno può essere ulteriormente distinto in:

- *Housing*: concessione in locazione ad un utente di uno spazio fisico (in rack), dove inserire il server, di proprietà del cliente. Tipicamente i server vengono ospitati in Webfarm o Data Center in cui si garantisce un'attenta gestione degli aspetti hardware, software ed infrastrutturali.
- *Hosting*: fornitura di servizi in rete (ad esempio per siti web, ospitati presso server di un provider).

Se il Service Provider offre i propri servizi utilizzando un'infrastruttura Cloud, allora le aziende possono considerare come opzioni di outsourcing i paradigmi già discussi nella sezione precedente:

- *Infrastructure as a Service (IaaS)*: l'infrastruttura fisica è fornita come servizio da un Provider su Cloud.
- *Platform as a Service (PaaS)*: la piattaforma tecnologica è fornita come servizio da un Provider su Cloud.
- *Software as a Service (SaaS)*: l'accesso a una intera applicazione è fornito come servizio da un Provider su Cloud.

Quando una infrastruttura viene data in Outsourcing o vengono utilizzati servizi esterni, è importante per l'organizzazione avere le necessarie garan-

zie relative alle prestazioni offerte dal Service Provider. Per questo motivo è necessario:

- Definire livelli di servizio (SLA – Service Level Agreement).
- Definire KPI (Key Performance Indicator) per il servizio/processo esternalizzato.
- Definire in dettaglio i *meccanismi per la gestione del servizio*:
  - Descrizione del servizio.
  - Fornitura del servizio.
  - Definizione livelli di servizio.
  - Monitoraggio.
  - Gestione dei problemi.
  - Reporting.
  - Documentazione.

## 10.8 Domande

- Illustrare le tecniche di clonazione e partizionamento.
- Illustrare le caratteristiche del Cloud computing.
- Opportunità SaaS per le aziende: vantaggi e svantaggi.



## Capitolo 11

# Sicurezza dei sistemi informativi

Esaminato dalla prospettiva della sicurezza, ovvero della protezione di sistemi, processi e dati dall'uso non consentito, un sistema informativo è visto come un insieme di entità (aree organizzative, gruppi di utenti o di risorse, applicativi e così via) che collaborano per perseguire obiettivi interni (per esempio, di tipo amministrativo) ed esterni (per esempio, fornitura di servizi a clienti e fornitori). In tale ottica, un sistema informativo è composto da entità che cooperano su basi organizzative utilizzando piattaforme tecnologiche che richiedono alle entità cooperanti di conoscersi direttamente o tramite una entità terza certificatrice.

In questo scenario, la sicurezza nei sistemi informativi è l'insieme delle misure organizzative e tecnologiche tese ad assicurare ad un *utente autorizzato* l'accesso a tutti e soli i servizi e le risorse previsti per quello specifico utente, nei tempi e secondo modalità predeterminate (per esempio, in sola lettura). Il capitolo illustra gli *attacchi* al sistema e le tecniche di *protezione*. In particolare, il capitolo illustra le tecniche alla base della sicurezza, principalmente *tecniche crittografiche*, i sistemi per il controllo dell'accesso alle basi di dati, e le componenti architettonicali in grado di gestire attacchi con opportune contromisure.

### 11.1 Proprietà di sicurezza

Una definizione più formale di sicurezza dei sistemi informativi è quella fornita da ISO [11] che la considera come l'insieme delle misure atte a proteggere i requisiti che si desidera il sistema soddisfi. Tali requisiti sono definiti secondo le *proprietà di sicurezza* descritte di seguito:

- *Integrità*: il sistema deve impedire l'alterazione diretta o indiretta delle informazioni, sia da parte di utenti e processi non autorizzati sia a seguito di eventi accidentali.

- *Autenticità*: il destinatario delle informazioni deve poter verificare l'identità del mittente e le informazioni devono essere integre. L'autore dell'informazione non potrà negare la paternità dell'informazione (non ripudio).

L'autenticità ha come presupposto l'*Autenticazione*: ogni *agente* (entità in grado di intraprendere azioni nel sistema, quindi utenti, programmi, macchine e così via) deve essere identificato prima di poter interagire con il sistema. L'autenticazione può anche essere mutua (anche il browser, ad esempio, deve poter autenticare il server con cui interagisce, prima di iniziare una sessione di scambio di informazioni). L'autenticazione più semplice è quella che richiede di inserire una password.

- *Riservatezza (o confidenzialità)*: nessun utente deve poter ottenere o dedurre dal sistema informazioni che non è autorizzato a conoscere. Si parla di *privacy* quando la riservatezza riguarda i dati sugli individui.
- *Disponibilità*: il sistema deve rendere disponibili a ciascun utente abilitato le informazioni e le risorse cui ha diritto di accedere, nei tempi e nei modi previsti.

La sicurezza deve essere considerata come un requisito non funzionale per il sistema informativo. Per considerarla tale, è necessario integrarne gli aspetti in tutte le fasi della progettazione e realizzazione di un sistema informativo, dalla pianificazione fino alla codifica della soluzione software e al deployment, al fine di valutare l'effettiva protezione messa in atto dal sistema. In questo modo, il sistema non solo sarà in grado di identificare e/o evitare tentativi di intrusione esterni o di abusi interni (*protezione attiva*), ma anche di rispondere efficacemente ad un attacco limitandone i danni (*protezione reattiva*).

## 11.2 Minacce, violazioni, vulnerabilità, attacchi

Le *minacce* a un sistema informativo si distinguono in:

- *Fisiche*: che danneggiano gli impianti e le infrastrutture. Si tratta, per esempio, di furti, danneggiamenti intenzionali, eventi accidentali (rottura impianto di condizionamento) o disastri naturali (inondazioni, incendi, terremoti).
- *Logiche*: che sottraggono o alterano informazioni e risorse. Esempi sono la sottrazione di dati (numeri di carta di credito memorizzati), la creazione di punti di accesso nascosti (backdoor) al sistema e in generale l'uso illegale delle risorse di calcolo e di comunicazione.
- *Accidentali*: ci si riferisce ad errori di configurazione del software e della rete, a malfunzionamenti di programmi, errori di data entry e così via.

Considerando che in questo testo ci si focalizza prevalentemente sulle minacce logiche, è da evidenziare che le *violazioni*, ovvero le conseguenze delle minacce, commesse ai danni dei sistemi informativi vengono perpetrare sfruttandone la vulnerabilità e possono comportare svariate situazioni non desiderate o illegittime, tra le quali, oltre all'accesso non autorizzato a risorse del sistema, anche la mancata possibilità di utilizzare le risorse cui si ha accesso (negazione del servizio, detta *Denial of Service - DoS*).

Una vulnerabilità presuppone l'esistenza di uno o più punti deboli (*exploit*) che rappresentano un potenziale fattore di rischio poiché consentono a chiunque di introdursi all'interno dello stesso e danneggiarlo. Un exploit è un programma, o più specificamente una tecnica che, facendo leva sulle vulnerabilità di un sistema, di un programma o di un protocollo, cerca di causarne un comportamento anomalo ed imprevisto. In genere, gli exploit hanno un campo di applicazione limitato poiché sono strettamente legati alla presenza di un determinato sistema operativo e/o servizio di rete o di particolari configurazioni. La maggior parte delle *vulnerabilità* di un sistema deriva da una scarsa qualità di progetto, da bug nella programmazione o nel testing, o da inadeguati controlli fisici e controlli degli accessi logici (per esempio, tramite password) ai siti e/o alle macchine e alle reti.

Nell'ambito dei sistemi informativi, gli *attacchi* di sicurezza sono una qualsiasi azione, messa in atto da individui o organizzazioni, che colpisce le basi di dati, le infrastrutture, le reti di comunicazioni e/o dispositivi elettronici personali tramite atti malevoli finalizzati al furto, all'alterazione o alla distruzione di elementi specifici (definiti target dell'attacco), violando le regole di accesso ai sistemi. Ciò diventa particolarmente rilevante in sistemi definiti critici (per esempio, sistemi governativi, sanitari, militari, ecc).

Tra gli attacchi più frequenti a livello di rete menzioniamo:

- *Sniffing*: intercettazione di messaggi scambiati tra due agenti e quindi accesso non autorizzato ad informazioni.
- *Spoofing*: invio di pacchetti con indirizzo di sorgente diverso dal proprio allo scopo di non farsi identificare o di impersonare un altro agente rubandone l'identità.
- *Hijacking*: anche detto Man-in-the-middle, combina tecniche di sniffing con tecniche di spoofing al fine di deviare le comunicazioni verso un agente malevolo che è così in grado di intercettare e modificare il traffico in modo invisibile ai sistemi compromessi.
- *Flooding*: intasamento della rete con quantità di traffico non desiderato che porta al blocco o all'irraggiungibilità di un servizio (Denial of Service - DoS).

A livello applicativo invece troviamo attacchi quali:

- *Malware*: software creato con lo scopo intenzionale di causare danni più o meno gravi. Si elencano qui i tipi di malware principali. (i) *Virus* – un

frammento di software presente che, una volta eseguito, è in grado di infettare dei file in modo da riprodursi facendo copie migliorate e più potenti di se stesso. (ii) *Worm* – una particolare categoria di malware in grado di replicarsi automaticamente; a differenza di un virus, un worm è di per sé un programma eseguibile e non necessita di infettare altri programmi per diffondersi. (iii) *Cavalli di Troia (Trojan)* – devono il loro nome al fatto che le loro funzionalità sono nascoste all'interno di un programma apparentemente utile; l'utente stesso, installando ed eseguendo un certo programma, può inconsapevolmente installare ed eseguire anche il codice nascosto. Quindi, a differenza dei virus o dei worm, i Trojan non si diffondono autonomamente. Spesso i Trojan sono nascosti in programmi scaricabili da internet. (iv) *Ransomware* – un tipo di malware che agisce sul dispositivo che infetta limitandone l'accesso fino a quando non viene pagato un riscatto (“ransom”).

- *Backdoor (porta sul retro)*: il progettista può lasciare in un programma una “porta” segreta per superare in parte le procedure di sicurezza attivate in un sistema informatico. Queste porte sono il più delle volte lasciate intenzionalmente aperte dai gestori del sistema informatico per agevolare le operazioni di manutenzione dell'infrastruttura informatica. Le backdoor possono essere però utilizzate anche per manomettere il sistema. Attraverso queste porte, infatti, un utente esterno può prendere il controllo remoto della macchina senza alcuna autorizzazione.
- *Spyware*: software usati per raccogliere informazioni dal sistema su cui sono installati e per trasmetterle a un destinatario interessato. Questo programma viene utilizzato per raccogliere generalmente informazioni riguardanti le attività online di un utente senza il suo consenso. Il termine spyware è spesso usato anche per definire un'ampia gamma di malware dalle funzioni più diverse, quali l'invio di pubblicità non richiesta (spam), la modifica della pagina iniziale o della lista dei Preferiti del browser, attività illegali quali la redirezione su falsi siti di e-commerce o l'installazione di dialer che chiamano numeri a tariffazione speciale.

Infine, è necessario considerare anche da *chi* difendersi. Bisogna esaminare sia gli attacchi provenienti da *terze parti* (persone esterne all'organizzazione che si desidera proteggere), ma anche da *utenti interni* al sistema, che utilizzano i propri diritti per accedere a informazioni non di loro competenza. I punti di accesso ad una rete di comunicazione sono potenziali punti di attacco. Inoltre gli utenti, sia gli utenti finali sia i gestori di sistema o di rete, che hanno il controllo di macchine e router e i privilegi di cancellazione/alterazione del traffico di rete, sono utenti interni che vanno controllati. Spesso l'attacco avviene da parte di personale esterno non autorizzato, ma ancora più spesso da utenti legittimi.

Dopo questa panoramica relativa alle proprietà di sicurezza da garantire per un sistema informativo e agli attacchi più comuni, il resto del capitolo illustrerà alcune delle tecniche di base utilizzate per contrastare tali attacchi al fine di garantire le citate proprietà.

Lo schema generale preso a riferimento prevede la presenza di un *soggetto* (sia esso una persona o un applicativo) che ha necessità di accedere ad un *oggetto* (sia esso una base di dati, un applicativo, un servizio in rete). È quindi necessario garantire l'autenticazione dei soggetti, il controllo che il soggetto abbia diritto ad accedere all'oggetto richiesto e la necessità di impedire a terze parti di inserirsi nella comunicazione tra soggetto e oggetto.

Tra le tecniche fondamentali, sicuramente la crittografia riveste un ruolo di importanza primaria. Grazie alla crittografia è infatti possibile non solo proteggere la confidenzialità delle trasmissioni (proprietà di riservatezza), ma anche fornire soluzioni per l'autenticità e l'integrità. Pertanto, sono di seguito illustrati i principi di base che governano gli algoritmi solitamente utilizzati. Successivamente, sono illustrate le tecniche di autenticazione possibili che, come detto in precedenza, possono anche basarsi — ma non solo — sulla crittografia. Infine sono affrontati i problemi di controllo degli accessi, sia relativamente alle basi di dati, sia per gli aspetti di rete (esaminando firewall e IDS).

### 11.3 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 (o più) *chiavi*. Solo la segretezza della chiave (o di una delle chiavi) può garantire la riservatezza e l'autenticità dei messaggi. Inoltre la chiave deve essere scelta tra un vastissimo numero di combinazioni (spazio delle chiavi) e cambiata spesso (per esempio, session key). La sicurezza crittografica dipende dall'ampiezza dell'intervallo di possibili valori che può assumere una chiave: più sono i possibili valori più diventa difficile, per i non autorizzati, recuperare la chiave e decifrare un messaggio.

Esistono due classi di algoritmi:

- *Algoritmi a chiave simmetrica* (o a chiave segreta), a cui appartengono algoritmi di sostituzione, algoritmi di trasposizione e algoritmi derivanti dalla loro combinazione. Per questi algoritmi esiste una sola chiave, detta segreta o simmetrica, che deve essere condivisa in modo sicuro tra chi invia e chi riceve. Esempi per questa classe di algoritmi sono il DES (Data Encryption Standard), usato in realtà nella versione 3-DES (Triple-DES), e AES (Advanced Encryption Standard) che è divenuto l'algoritmo standard nel 2000.
- *Algoritmi a chiave asimmetrica* (o a chiave pubblica), che si basano su problemi matematicamente complessi e sul fatto che ogni soggetto possieda una coppia di chiavi (chiave privata e chiave pubblica). Per questa classe di algoritmi, i metodi più utilizzati sono l'RSA e il Diffie-Hellman.

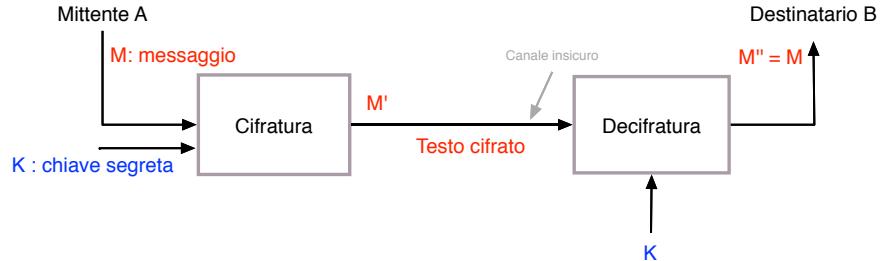


Figura 11.1: Crittografia simmetrica

### 11.3.1 Crittografia simmetrica

In Figura 11.1 è mostrato lo schema di funzionamento della crittografia simmetrica, dove  $K$  indica la chiave simmetrica (chiave segreta) condivisa fra mittente e destinatario. In caso di attacchi di tipo sniffing sul canale insicuro, i malintenzionati, non riuscendo a captare il testo in chiaro  $M$ , potrebbero solo raccogliere un gran numero di messaggi su cui fare crittoanalisi con l'intento di ricavare la chiave  $K$ . Per rendere inutili le crittoanalisi e garantire la sicurezza della comunicazione la chiave  $K$  deve cambiare spesso (possibilmente ad ogni sessione). Il problema è che  $A$  e  $B$  devono trovare un canale sicuro alternativo a quello usato per la trasmissione per scambiarsi di frequente la chiave  $K$  segreta.

Le due tecniche principali alla base della crittografia simmetrica sono la sostituzione e la trasposizione (o permutazione).

Nella Figura 11.2 sono illustrati due meccanismi di *sostituzione* fissa, con chiave numerica e con sostituzione monoalfabetica. In generale, sono necessarie mappature con testi più lunghi per rendere più difficile la ricostruzione del testo originario. Un algoritmo semplice a chiave simmetrica basato sul meccanismo di sostituzione fissa è il cifrario di Cesare. L'algoritmo di cifratura prevede che il messaggio venga cifrato sostituendo ogni lettera del messaggio con la lettera dell'alfabeto che si trova  $k$  posizioni più avanti, dove  $k$  costituisce la chiave da condividere tra mittente e destinatario. Ad esempio, se il mittente A vuole mandare un messaggio  $M = \text{"ciao"}$  e  $k = 4$ , il messaggio  $M'$  sarà uguale a "goes" (considerando l'alfabeto italiano).

Nella *trasposizione*, la chiave serve per indicare come permutare le lettere contenute nel testo in chiaro durante la cifratura. Nell'esempio riportato nella Figura 11.2, la chiave indica in quante colonne viene trascritto il testo in chiaro, che poi vengono ordinate secondo l'ordine lessicografico delle lettere della chiave. Per cui le prime lettere del testo cifrato corrispondono alla colonna che contiene la lettera A della chiave: "fine" e così via, seguendo l'ordine indicato per le colonne dalla chiave.

Come detto prima, sostituzioni e trasposizioni sono le operazioni base degli algoritmi di crittografia simmetrica, come mostrato in Figura 11.3, in cui il

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| <p>► <b>Sostituzione</b></p> <p>chiave = 4</p> <table style="margin-left: 20px; border-collapse: collapse;"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>Z</td></tr> <tr><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>V</td><td>U</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>A</td></tr> </table> <p>ciao -&gt; goes</p> <p>mapping fisso (sostituzione monoalfabetica)</p> <table style="margin-left: 20px; border-collapse: collapse;"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr> <tr><td>Q</td><td>W</td><td>E</td><td>R</td><td>T</td><td>Y</td><td>U</td><td>I</td><td>O</td><td>A</td><td>P</td><td>S</td><td>D</td><td>F</td><td>G</td><td>H</td><td>J</td><td>K</td><td>L</td><td>Z</td><td>X</td><td>C</td><td>V</td><td>B</td><td>N</td><td>M</td></tr> </table> <p>ciao -&gt; eoqg</p> | A | B | C | D | E | F | G | H | I | L | M | N | O | P | Q | R | S | T | U | V | Z | E | F | G | H | I | L | M | N | O | P | Q | R | S | T | V | U | W | X | Y | Z | A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Q | W | E | R | T | Y | U | I | O | A | P | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M |
| A   | B | C | D | E | F | G | H | I | L | M | N | O | P | Q | R | S | T | U | V | Z |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| E   | F | G | H | I | L | M | N | O | P | Q | R | S | T | V | U | W | X | Y | Z | A |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| A   | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Q   | W | E | R | T | Y | U | I | O | A | P | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <p>► <b>Trasposizione (permutazione)</b></p> <table style="margin-left: 20px; border-collapse: collapse;"> <tr><td>P</td><td>R</td><td>O</td><td>V</td><td> </td><td>A</td><td>T</td><td>I</td></tr> <tr><td>4</td><td>5</td><td>3</td><td>7</td><td>1</td><td>6</td><td>2</td><td></td></tr> <tr><td>t</td><td>r</td><td>a</td><td>s</td><td>f</td><td>e</td><td>r</td><td></td></tr> <tr><td>i</td><td>r</td><td>e</td><td>u</td><td>n</td><td>m</td><td>i</td><td></td></tr> <tr><td>l</td><td>l</td><td>o</td><td>n</td><td>e</td><td>a</td><td>b</td><td></td></tr> </table> <p>chiave</p> <p>testo in chiaro</p> <p>testo cifrato: fneribaeo...</p>   | P | R | O | V |   | A | T | I | 4 | 5 | 3 | 7 | 1 | 6 | 2 |   | t | r | a | s | f | e | r |   | i | r | e | u | n | m | i |   | l | l | o | n | e | a | b |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| P   | R | O | V |   | A | T | I |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4   | 5 | 3 | 7 | 1 | 6 | 2 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| t   | r | a | s | f | e | r |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| i   | r | e | u | n | m | i |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| l   | l | o | n | e | a | b |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Figura 11.2: Crittografia simmetrica: sostituzione e trasposizione

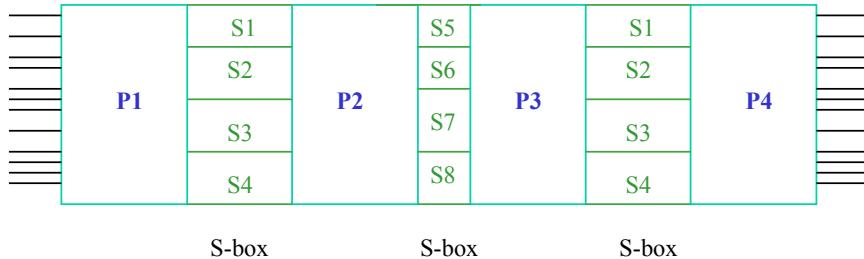


Figura 11.3: Crittografia simmetrica: composizione di cifrature

messaggio cifrato è ottenuto dal testo in chiaro da una serie di sostituzioni - blocchi S-box - e trasposizioni - blocchi P-box (Permutation box).

L'algoritmo standard è l'AES in cui il messaggio viene cifrato utilizzando una chiave che può essere lunga 128, 192 o 256 bit. Per la cifratura il messaggio viene spezzato in blocchi da 128 bit e ogni blocco viene cifrato tramite diverse fasi che comprendono combinazioni in xor dei bit del messaggio con i bit di chiave, sostituzioni, trasposizioni e combinazioni tra bit in fasi intermedie.

I punti critici della crittografia simmetrica sono legati alla gestione della chiave: (i) la chiave deve essere cambiata spesso per evitare che venga scoperta, (ii) la chiave deve essere generata e distribuita con la massima sicurezza. Il maggior vantaggio della crittografia simmetrica è invece la velocità con cui vengono eseguite le operazioni di codifica e decodifica essendo anche possibile una sua implementazione in hardware con processori dedicati.

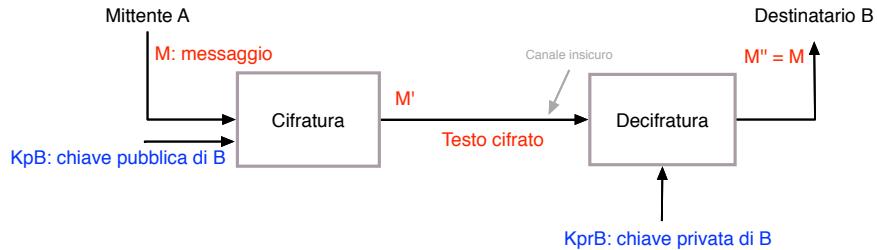


Figura 11.4: Crittografia asimmetrica (riservatezza)

### 11.3.2 Crittografia asimmetrica

La crittografia asimmetrica (o a chiave pubblica) richiede che ogni agente sia in possesso di una coppia di chiavi correlate: una chiave pubblica  $K_P$  e una corrispondente chiave privata  $K_{pr}$ . La correlazione tra le due chiavi riguarda il principio di un algoritmo di crittografia asimmetrica: tutto ciò che è cifrato con una delle due chiavi può essere decifrato con l'altra chiave della coppia. Quindi, supposto che la coppia di chiavi in possesso di A sia  $\{K_{pr}^A, K_p^A\}$  tutto ciò che viene cifrato con  $K_{pr}^A$  può essere decifrato solo con  $K_p^A$  e viceversa. Una proprietà importante delle coppie di chiavi è che non deve essere in alcun modo possibile dedurre una delle due chiavi a partire dalla conoscenza dell'altra.

La chiave pubblica può essere distribuita, mentre la chiave privata deve restare segreta e l'agente che la possiede deve conservarla in modo adeguato. Non essendo possibile, per la proprietà citata in precedenza, che data la chiave pubblica si possa risalire a quella privata, nonostante la prima sia a disposizione di chiunque su repository pubblici, la sicurezza della seconda non è comunque minata. Questo evita la necessità di trasmettere le chiavi segrete fra agenti in rete, i quali devono solo condividere le proprie chiavi pubbliche, direttamente o tramite un Servizio di Distribuzione Chiavi Pubbliche.

Con la crittografia asimmetrica, possiamo usare le due coppie di chiavi,  $\{K_{pr}^A, K_p^A\}$  e  $\{K_{pr}^B, K_p^B\}$ , in possesso di A e B, per garantire riservatezza e autenticità. In particolare, un primo modo per utilizzare gli algoritmi a chiave simmetrica è mostrato in Figura 11.4. In questo caso il mittente A cifra il messaggio con la chiave pubblica di B e, pertanto, il messaggio può essere decifrato solo con la chiave privata di B. Questo garantisce la proprietà di riservatezza in quanto B sarà l'unico in grado di decifrare il messaggio (essendo l'unico in possesso di  $K_{pr}^B$ ) e di conseguenza il mittente ha la garanzia che il messaggio non venga letto da agenti non autorizzati. Il punto debole di questo meccanismo è che tutti possono entrare in possesso della chiave pubblica di B e di conseguenza fingersi A all'interno della comunicazione, con problemi di attacchi di tipo Man-in-the-middle.

Il mittente A potrebbe anche cifrare il messaggio con la sua chiave privata. In questo caso il destinatario B deve decifrare il messaggio con la chiave pubblica di A (vedere Figura 11.5). In questo modo lo scambio è autenticato, in quanto

cifrando con la propria chiave privata il mittente A certifica la propria identità.

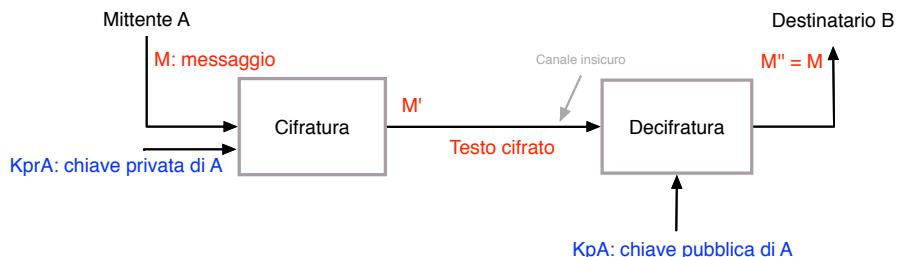


Figura 11.5: Crittografia asimmetrica (autenticità)

Il messaggio scritto da A è però facilmente leggibile anche da terzi in possesso della chiave pubblica di A. Per garantire sia la proprietà di autenticità sia la proprietà di riservatezza i due approcci descritti sopra sono da combinare con una doppia fase di cifratura come mostrato in Figura 11.6.

Il principio su cui si basa la crittografia asimmetrica è l'uso di una funzione  $f$  semplice, la cui inversa  $f^{-1}$  è complessa. Per esempio, il calcolo del prodotto di due numeri primi ( $f$ ) è semplice, mentre  $f^{-1}$  (calcolo di due numeri primi che, dato un numero di partenza, ne siano fattori) è complesso, specie se il numero da fattorizzare è molto grande ( $> 10^{100}$ ).

Lo schema della crittografia asimmetrica fu proposto da Diffie e Hellmann nel 1976, basandosi proprio sul prodotto di due numeri primi molto grandi. Anche un altro degli algoritmi più famosi, l'algoritmo RSA, è basato sulla difficoltà di trovare i fattori primi di numeri grandi e la costruzione della coppia di chiavi avviene secondo i seguenti passi:

1. si scelgono due numeri primi  $P, Q$  di valore elevato;
2. si calcola  $N = P \times Q$  e  $Z = (P - 1) \times (Q - 1)$ .  $N$ , come vedremo, sarà la componente comune delle due chiavi;
3. vanno ora scelti due valori  $E$  e  $D$  che comporranno il resto delle chiavi;
  - $E$  (con  $E < N$ ) va scelto tra i numeri che non hanno fattori comuni con  $Z$  ( $E, Z$  sono detti relativamente primi);
  - $D$  deve soddisfare la seguente equazione  $(E \times D) \bmod Z = 1$ , quindi  $(E \times D) - 1$  deve essere divisibile per  $Z$
4. la chiave pubblica  $K_p$  e la chiave privata  $K_{pr}$  sono quindi definite come:  $K_p = (E, N), K_{pr} = (D, N)$

Per fare un esempio (con numeri piccoli) supponiamo che i valori di partenza siano  $P = 5$  e  $Q = 7$ . Per definizione  $N = 35$ ,  $Z = 24$ . Un valore adatto per  $E$  è 5 in quanto  $E$  e  $Z$  non hanno fattori comuni ed  $E < Z$ . Infine, un valore valido di  $D$  è 29 in quanto  $5 \times 29 = 145$  e la divisione  $145/24$  ha resto pari a 1.

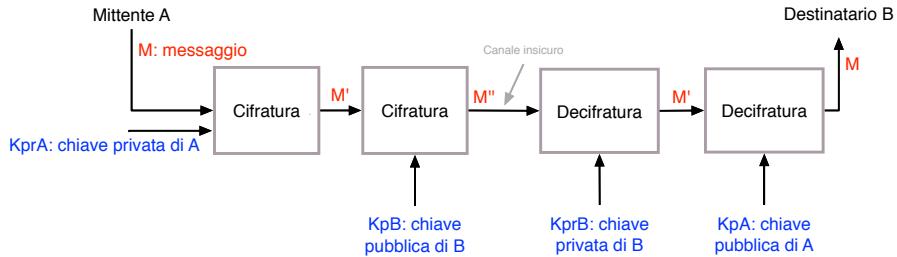


Figura 11.6: Crittografia asimmetrica (riservatezza e autenticità)

Dai dati ottenuti è possibile dire che  $K_p = (5, 35)$ ,  $K_{pr} = (29, 35)$ .

Avendo le chiavi a disposizione, detto  $M$  il messaggio da cifrare e  $C$  il messaggio cifrato:

- cifratura:  $C = M^E \text{mod} N$
- decifratura  $M = C^D \text{mod} N$

Lo svantaggio dell'algoritmo RSA, e degli algoritmi a chiave asimmetrica in generale, è la lentezza computazionale. Infatti, la necessità di calcolare l'esponenziale di numeri non piccoli richiede una certa capacità computazionale per messaggi molto lunghi. Per questo motivo, si usa la criptratura asimmetrica per crittografare le chiavi che devono essere scambiate tra gli agenti negli algoritmi a chiave privata, e per lo scambio di messaggi o testi si usa la criptratura simmetrica.

### 11.3.3 Integrità e funzione di hash

Gli algoritmi di crittografia simmetrica e asimmetrica garantiscono le proprietà di riservatezza e autenticità. Per riuscire a fornire anche una garanzia in termini di integrità ci si deve avvalere delle *funzioni di hash*. Una funzione di hash è una funzione che trasforma un messaggio  $M$  di lunghezza arbitraria in un output di lunghezza fissa chiamato *impronta digitale*, *hash* o *digest* del messaggio originale. Gli algoritmi di *hash* sono pubblici e ben noti. La funzione di hash viene utilizzata in quest'ambito in quanto caratterizzata dalle seguenti proprietà:

- Coerenza: a messaggi uguali deve essere associato lo stesso digest.
- Univocità: la probabilità che due messaggi diversi (anche di un solo bit) siano associati allo stesso digest deve essere quasi nulla.
- Non invertibilità: la funzione non deve essere invertibile. Deve essere impossibile risalire al messaggio partendo dal digest.

Le funzioni di hash generano così le impronte digitali (fingerprint) del messaggio, che possono costituire una prova dell'integrità del messaggio stesso, cioè

del fatto che non sia stato manipolato da agenti non autorizzati durante la comunicazione. Infatti, come mostrato in Figura 11.7, generalmente un mittente che vuole spedire un messaggio può calcolarne l'impronta digitale tramite una funzione di hash e inviare il messaggio concatenato all'impronta. Una volta ricevuto il messaggio, il destinatario deve disaccoppiare il messaggio dall'impronta e ricalcolare l'impronta del messaggio ricevuto, generando il digest “fresco”. Se l'impronta ricevuta e quella calcolata a partire dal messaggio ricevuto sono uguali allora si può dire che il messaggio è integro altrimenti il messaggio è stato manipolato da agenti non autorizzati durante qualche fase della trasmissione.

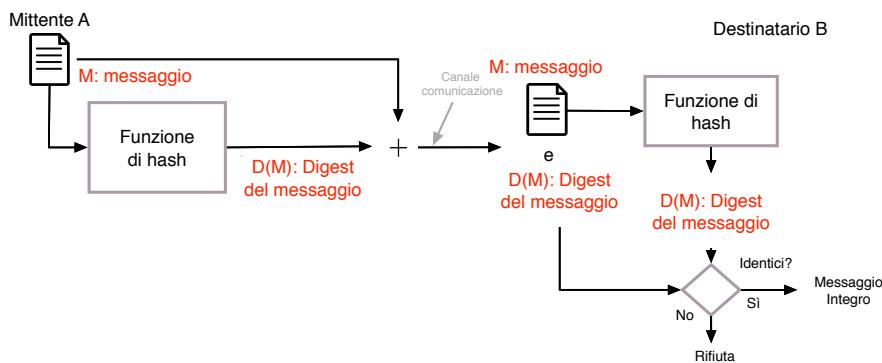


Figura 11.7: Utilizzo della funzione di hash

La Figura 11.7 mostra il principio che sta alla base dell'uso della funzione di hash, ma per un utilizzo sicuro della funzione bisogna anche proteggere l'impronta da attacchi esterni. Di conseguenza, l'impronta deve essere utilizzata in combinazione a sistemi a chiave pubblica come mostrato nella prossima sezione.

#### 11.3.4 Firma digitale

Obiettivo della creazione e apposizione a un dato/informazione di una *firma digitale* (o firma elettronica) è garantire l'autenticità dei dati e identificare con certezza il loro creatore [12]. Lo schema della firma digitale appare in Figura 11.8. La firma digitale è definita come il digest crittografato di un documento (o messaggio), ottenuto utilizzando la chiave privata del mittente del documento. La firma viene allegata al documento; poi, firma e documento vengono spediti. In alcuni casi, prima della spedizione, firma e documento vengono crittografati con la chiave pubblica del destinatario che deve usare la sua chiave privata per decifrare quanto ricevuto. In generale una volta che firma e documento vengono ricevuti, si eseguono due azioni in parallelo: (i) ricalcolo del digest del documento; (ii) decodifica della firma con la chiave pubblica del mittente e recupero del digest del documento originale. A questo punto si possono confrontare i due digest e capire se il documento è stato manipolato da agenti non autorizzati.

La firma digitale richiede un tempo di elaborazione lungo e in genere non viene usata in processi di rete che richiedono alte prestazioni, ma solo in soluzioni

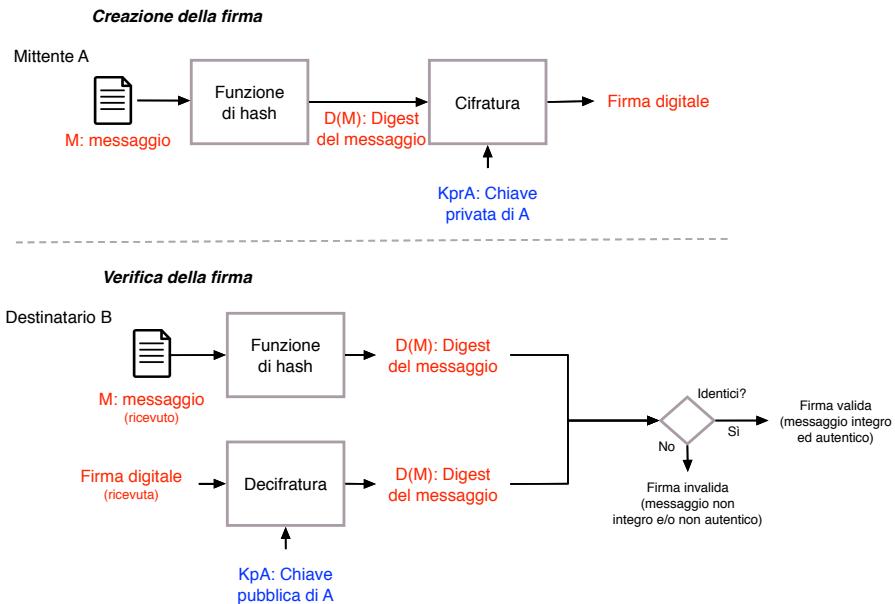


Figura 11.8: Firma digitale

applicative in cui è direttamente coinvolto un agente umano (usando protocolli SSL, TLS e S/MIME). Inoltre, la firma digitale, poiché dipende non solo dalla chiave privata del firmatario, ma anche dai dati tramite il digest, è diversa per ogni diverso insieme di dati che viene firmato. La firma apposta su un documento elettronico quindi non può essere semplicemente copiata ed apposta su un documento diverso. Infine, essa non solo identifica chi ha generato i dati, ma dimostra anche che i dati non sono stati modificati dopo essere stati firmati ed è quindi più forte della firma autografa (proprietà del *non ripudio*).

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

Per poter applicare gli algoritmi di crittografia, e talvolta anche quelli di digest, occorre che le parti interessate possiedano e condividano le chiavi. Questo aspetto è detto *gestione delle chiavi* (key management) e richiede la soluzione di vari sottoproblemi, quali la generazione e la conservazione delle chiavi e lo scambio delle chiavi (key exchange o key distribution).

La generazione delle chiavi deve essere fatta direttamente da chi effettuerà le operazioni crittografiche, in casi eccezionali (ambienti fidati ad alta sicurezza) è lecito permettere che le chiavi vengano generate da un'entità diversa da quella che ne farà uso. Le chiavi crittografiche dovrebbero essere stringhe casuali di bit e quindi per la loro generazione si può utilizzare un Random Number Generator (RNG). La qualità del RNG determina la qualità del sistema di sicurezza.

Per la conservazione delle chiavi crittografiche, specialmente quelle private, si possono utilizzare soluzioni software o hardware. Nel primo caso si può ricorrere alla memorizzazione delle chiavi in un file cifrato protetto da password. Per quanto riguarda le soluzioni hardware le chiavi possono essere memorizzate all'interno di dispositivi hardware removibili (es. hard disk esterni) oppure si può ricorrere a dispositivi "attivi" ossia in grado di memorizzare le chiavi e di svolgere autonomamente le operazioni crittografiche (ad esempio, smart-card, token USB o PC card).

Lo scambio di chiavi è particolarmente critico nella crittografia simmetrica in quanto la conoscenza della chiave comprometterebbe interamente la comunicazione. Una delle tecniche più usate per lo scambio di chiavi è la tecnica OOB (Out-Of-Band) in cui la chiave viene distribuita tramite un canale diverso da quello su cui transitano i dati. In alternativa, se il destinatario dei dati cifrati dispone di una coppia di chiavi asimmetriche, il mittente può inviargli la chiave segreta cifrandola con la chiave pubblica del destinatario. Infine, si può anche ricorrere a una terza parte (key distribution center).

Nel caso in cui la chiave da scambiare sia di tipo pubblico, non è importante la sua segretezza: il problema è associarvi correttamente l'identità del possessore. Le chiavi pubbliche vengono diffuse tramite una struttura dati denominata *Certificato a Chiave Pubblica* o PKC (Public Key Certificate) di cui esistono vari formati tra cui quello X.509v3.

I sistemi che supportano la generazione e gestione delle chiavi sono dette PKI (*Public Key Infrastructure*). Si tratta di infrastrutture che forniscono metodi e strumenti per realizzare le funzioni crittografiche di base per una certa comunità di utenti, quali:

- emissione di certificati a chiave pubblica, dopo aver svolto i dovuti controlli tecnici e procedurali;
- revoca dei certificati a chiave pubblica (per esempio, a seguito di furto della chiave privata associata alla chiave pubblica);
- distribuzione dei certificati a chiave pubblica e delle informazioni circa i certificati revocati.

Opzionalmente, una PKI può fornire anche supporto per la validazione di una firma digitale tramite funzioni di identificazione dell'ora in cui è stata apposta la firma (marca temporale), del ruolo ricoperto dall'individuo che ha firmato (certificazione del ruolo) e del motivo per cui è stata apposta la firma (politica di firma).

Lo schema di una PKI è mostrato in Figura 11.9: l'utente (visto come end entity dello scenario) richiede un certificato personale all'entità certificatrice (CA - Certification Authority) la quale svolge il ruolo di notaio, garantendo l'identità delle parti e svolgendo anche il ruolo di Autorità di Registrazione (Registration Authority - RA). In alternativa, la CA può ricorrere a una terza parte fidata per il ruolo di RA, in carico di identificare l'utente, tramite una apposita fase di acquisizione delle credenziali di fiducia. I certificati sono memorizzati in un

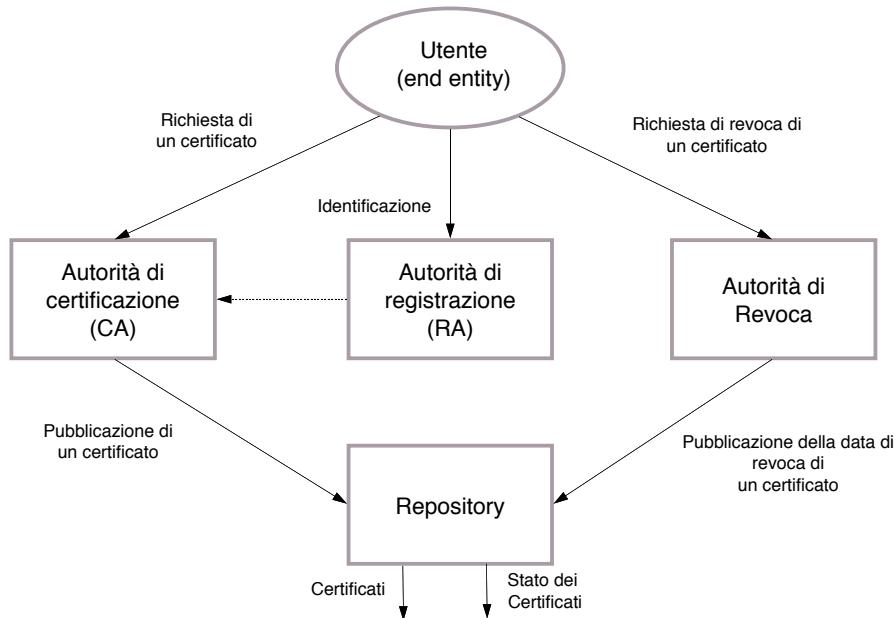


Figura 11.9: Attori e funzioni di una Public Key Infrastructure

Repository opportunamente condiviso nell'ambiente di cooperazione, cui si accede per verificare l'identità di una parte (per esempio, del Server di Internet Banking in cui si stanno inserendo dati sensibili). Tale Repository deve rilasciare anche informazioni sullo stato dei certificati, i quali hanno una scadenza o possono essere revocati su richiesta della CA o della RA. Un'Autorità di Revoca si occupa di revocare i certificati scaduti. È importante che la revoca di un certificato venga resa nota nell'ambiente con tempestività per evitare frodi relative a certificazioni scadute.

Un esempio di certificato è mostrato in Figura 11.10, dove i vari campi indicano anche l'algoritmo di cifratura impiegato (RSA), quello di hash (MD5), la lunghezza della chiave (1024 bit), le autorità di certificazione e rilascio (PoliMI), la validità e il titolare della firma.

Per garantire che il certificato non possa essere alterato, esso viene protetto con la firma digitale dell'Autorità di Certificazione che lo ha emesso. Quando si riceve una chiave pubblica attraverso un certificato, occorre controllare l'integrità e validità della firma: il digest calcolato sul certificato (ossia sulla chiave pubblica e sui dati ad essa associati) viene confrontato con il digest estratto dalla firma digitale apposta dalla CA che ha emesso il certificato.

Se questo confronto ha esito positivo, il certificato è sicuramente integro, ma resta da verificare se la CA che lo ha emesso è una CA fidata. Per questo motivo viene mantenuto un elenco delle chiavi pubbliche delle CA fidate.

La validità di un certificato digitale è garantita dalla CA. Per questo motivo,

| Campi:                | Esempio:                               |
|-----------------------|--|
| • version             | 1                                      |
| • serial number       | 2430                                   |
| • signature algorithm | RSA with MD5, 1024                     |
| • issuer              | C=IT, O=PoliMi                         |
| • validity            | 1/1/2006 - 31/12/2020                  |
| • subject             | C=IT, O=PoliMi<br>CN=MariagraziaFugini |
| • subjectpubkeyinfo   | RSA, 1024, xxxx...xxx                  |
| • digital signature   | yy.....yyy                             |

Figura 11.10: Esempio di struttura di un certificato digitale a chiave pubblica

un certificato digitale è un documento in cui sono riportate le informazioni su chi è il possessore della chiave pubblica, la chiave pubblica stessa, e il tutto viene firmato dalla CA per garantire che nessuno abbia alterato il certificato.

Esempi di utilizzo delle chiavi pubbliche e dei certificati sono i seguenti: scambio di messaggi e-mail, applicazioni proprietarie (per controllo remoto, o per realizzare moduli di sicurezza), controllo dell'accesso ad apparecchiature hardware (router, o hardware di rete), a livello applicativo (per esempio, per applicazioni di e-commerce o per realizzare siti Web sicuri) e per ottenere componenti software firmati, quindi trusted.

## 11.4 Gestione degli utenti e controllo degli accessi

La Figura 11.11 offre una panoramica della gestione sicura degli utenti tramite i vari meccanismi illustrati fin qui e nel seguito della sezione.

### 11.4.1 Autenticazione e autorizzazione

Preliminamente ad ogni controllo di sicurezza, è necessario *autenticare* gli agenti prima di permettere loro di aprire un canale di comunicazione o di accedere a informazioni situate in Intranet o in zone protette. I meccanismi di autenticazione permettono al sistema di riconoscere in modo certo un agente, richiedendogli credenziali che lo portano a un passo più avanzato nell'accesso al sistema, rispetto a una semplice registrazione (utilizzata per esempio, per fare browsing di informazioni). Con l'autenticazione, sono possibili filtri nelle modalità di accesso ai dati delle applicazioni e dal DBMS o altri componenti del sistema informativo.

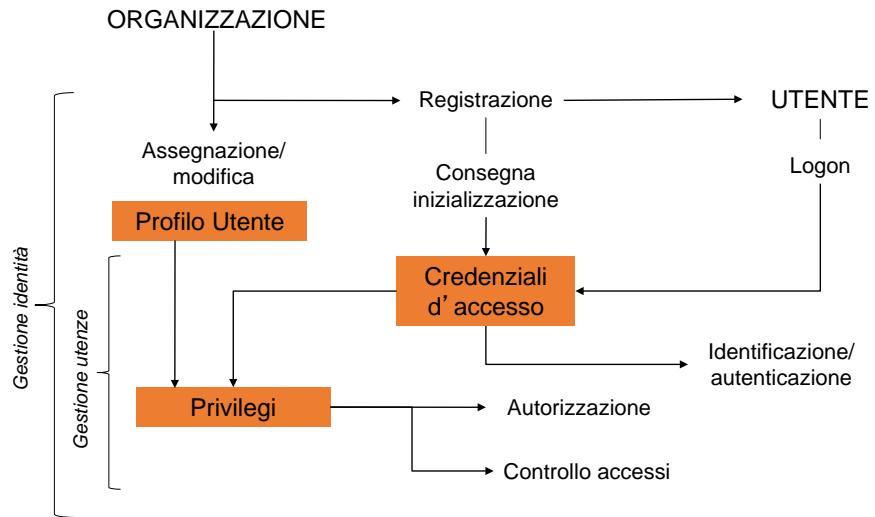


Figura 11.11: La gestione utenti con metodi di autenticazione e altri meccanismi di sicurezza

Un agente può essere identificato sfruttando una o più delle seguenti caratteristiche:

- SOMETHING YOU KNOW - SYK: un'informazione conosciuta (es. password).
- SOMETHING YOU HAVE - SYH: un oggetto posseduto (es. una carta magnetica).
- SOMETHING YOU ARE - SYA: una caratteristica fisica - nel caso di persone (es. impronta digitale, retina, DNA).

L'autenticazione semplice si basa su userid e password mentre l'autenticazione robusta si basa su meccanismi del tipo One-Time Password (OTP), su sistemi challenge-response, in cui la password non viene mai trasmessa ma solo usata per effettuare un calcolo che ne dimostra indirettamente la conoscenza: risposta = F(sfida, password), o mediante l'uso di dispositivi personali (token) come avviene nelle smart card, o infine su caratteristiche biometriche.

Mentre l'autenticazione permette di verificare la licenza di un identificativo (credenziali), associando un agente a una *identità* precisa, l'*autorizzazione* è la fase in cui si verifica se l'agente è autorizzato a svolgere la funzione richiesta, quali, per esempio funzioni del sistema operativo (per esempio, accesso ai file), funzioni applicative (business process, task) e programmi.

In particolare, il *controllo degli accessi* è l'insieme delle procedure che verificano se l'agente è autorizzato allo svolgimento di funzioni elementari (read,

write, execute). Può essere svolto dal sistema operativo, dal DBMS o da programmi/servizi specifici di sicurezza, limitando così i danni che possono essere causati da utenti illegittimi e/o programmi malevoli perché i servizi di sicurezza sono progettati specificamente per le risorse da proteggere.

Nella prossima sezione si approfondisce il controllo dell'accesso ai dati memorizzati nelle basi di dati.

### 11.4.2 Sistemi di controllo di accesso ai dati

Quando si parla di *DBMS sicuri* ci si riferisce alle politiche e ai meccanismi del DBMS per proteggere i dati da accessi illegittimi. Rispetto ai controlli di accesso messi in atto da un sistema operativo, per il DBMS gli aspetti importanti sono la granularità degli oggetti protetti, le correlazioni semantiche dei dati, l'esistenza di metadati (dizionari), che vanno anch'essi protetti, la molteplicità dei tipi di dati, il fatto che esistono oggetti statici e oggetti dinamici (per esempio le view) e il ciclo di vita dei dati (più lungo rispetto alla vita di un oggetto del sistema operativo). I meccanismi di sicurezza nei DBMS hanno come requisiti la necessità di trattare dati a diverse granularità, di filtrare diversi modi di accesso e tipologie di controllo (controlli di inferenza, di flusso, di accesso), di gestire l'autorizzazione dinamica (grant/revoke) e di garantire l'assenza di backdoor, complessivamente fornendo prestazioni ragionevoli [9].

In generale, il sistema del controllo degli accessi funziona secondo lo schema illustrato in Figura 11.12.

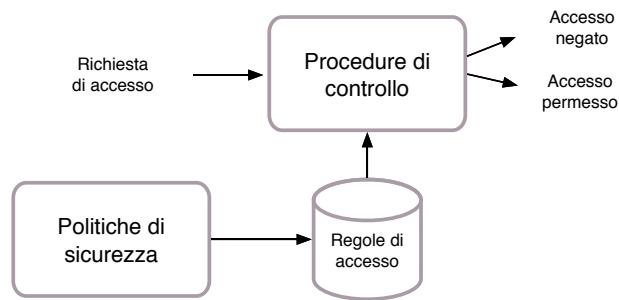


Figura 11.12: Sistema di controllo degli accessi

I controlli di accesso nei DBMS si basano su *regole di accesso* composte da:

- *Autorizzatori*: quando vale il concetto di ownership sulle risorse (ovvero chi crea un dato ne diventa owner - proprietario) si permette al possessore di una risorsa di gestirne gli accessi in modo selettivo, diventando così l'autorizzatore per quella risorsa. Più in generale, l'autorizzatore è per esempio il Data Base Administrator (DBA) o il security officer, che hanno funzione di controllo sui dati. Per basi di dati centralizzate, è utile che vi

sia un unico autorizzatore, mentre in basi di dati distribuite esistono vari autorizzatori organizzati per esempio in gerarchie.

- *Soggetti*: sono utenti (definiti dai loro user profile), amministratori e programmatore applicativi, gruppi o classi, transazioni o applicazioni. Se l'utente è in più gruppi e/o usa vari applicativi, il soggetto è identificato da una tupla del tipo:
  - (USER-ID, GROUP-ID) oppure
  - (USER-ID, APPL-ID) oppure
  - (USER-ID, ROLE-ID), che identifica l'utente tramite il suo ruolo aziendale (controllo dell'accesso role-based).
- *Oggetti*: la scelta della granularità degli oggetti protetti è cruciale per l'efficacia/efficienza dei controlli. Un oggetto protetto è per esempio una tabella, una colonna, una tupla, un elemento. Occorre proteggere anche gli schemi e i dizionari.
- *Diritti*: indica i modi di accesso (es. read, write, execute per i programmi, oppure CRUD - create, read, update, delete per i dati).

La definizione delle regole è dettata dalle *politiche di sicurezza*. In linea generale è possibile distinguere tra due approcci: a *sistema chiuso* e a *sistema aperto*. Le politiche di accesso in un sistema chiuso prevedono che gli accessi permessi sono tutti e soli quelli specificati dalle regole. Quindi tutto ciò che non è esplicitamente permesso è da considerarsi vietato. Al contrario, in un sistema aperto, gli accessi permessi sono tutti, tranne quelli che vengono espressamente vietati tramite regole. Pertanto, tutto ciò che non è esplicitamente vietato è da considerarsi permesso.

Le regole di accesso definite dagli autorizzatori sono basate su un *modello base a tre componenti*:

*(subject, object, right)*

Il modello è stato poi esteso al modello a quattro componenti: (subject, object, right, constraint), dove i constraint esprimono i vincoli di accesso dipendenti dal contenuto. I vincoli permettono di affrontare gli aspetti di semantica dei dati, perché permettono di esprimere filtri di accesso dipendenti dal significato aziendale dei dati - per esempio, l'attributo STIPENDIO può essere concesso in "lettura" al solo titolare dello stipendio stesso. Possono esprimersi anche altri vincoli che negano per esempio la possibilità di accesso combinato a dati "correlati" - che porterebbe a inferire dati per i quali non si ha diritto di lettura, ma che possono essere dedotti come combinazioni di dati acceduti lungo varie transazioni, apparentemente scorrelate (vincoli dipendenti dalla storia degli accessi).

Un esempio di tabella relazionale SYSAUTH utilizzata per esprimere le regole di accesso è mostrata in Figura 11.13.

| <b>subj</b> | <b>obj</b>             | <b>right</b> | <b>constraint</b>                |
|-------------|------------------------|--------------|----------------------------------|
| mgr         | EMP<br>NAME<br>ADDRESS | ALL<br>READ  | = mgr of NAME<br>NOT mgr of NAME |
| employee    | NAME<br>AGE<br>ADDRESS | READ,UPDATE  | employee=NAME                    |
|             | NAME<br>AGE<br>ADDRESS | READ         | mgr(NAME)=<br>mgr(employee)      |

Figura 11.13: Esempio di tabella delle regole di accesso per la relazione EMP

In figura, l'accesso a EMP (relazione con attributi NAME, ADDRESS, AGE) può essere acceduta in qualunque modalità (ALL) dai manager (mgr), in sola lettura (READ) negli attributi NAME e ADDRESS per gli impiegati di cui "mgr" (ovvero l'utente che si è autenticato al sistema con credenziali di ruolo MANAGER) non è manager. Il soggetto employee può accedere in READ e UPDATE ai propri dati e in sola lettura (READ) ai dati di impiegati gestiti da altri manager (per esempio, di altri dipartimenti).

La parte 'constraint' del modello a quattro componenti viene realizzata tramite *meccanismi* basati su Protection View sulla base di dati: il vincolo definisce una view dei dati legittimi per un utente definendo una view. Il modello fornisce controllo dell'accesso dipendente dal nome (per esempio, dal nome della tabella/attributo), controllo dell'accesso dipendente dal contenuto, come si è appena illustrato, ma ancora non fornisce controlli sull'amministrazione dell'insieme delle regole di accesso (chi le può cambiare, gestire, ecc.) né sulla delega dei privilegi di accesso. Servono modelli a cinque e più componenti per riuscire ad esprimere regole di accesso più granulari e legate alla semantica dei dati e alla dinamicità dei dati stessi.

Prima di illustrare gli altri modelli di regole, si illustrano le *politiche di accesso ai dati*, in particolare raggruppate in due categorie:

- Politiche discrezionali.

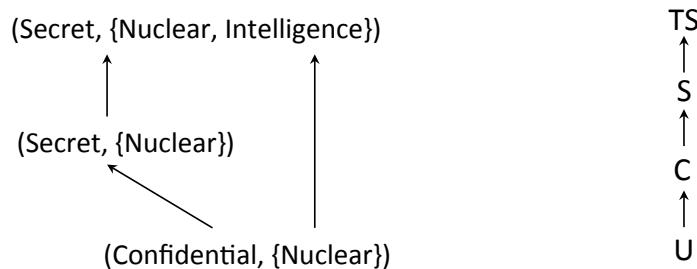


Figura 11.14: Security Class e livelli di sensitività (MAC)

- Politiche mandatorie.

In un sistema gestito in base a politiche discrezionali di accesso ai dati (*Discretionary Access Control - DAC*), è ‘discrezione’ degli utenti amministrare i dati che possiedono; infatti, nei sistemi DAC vale il concetto di *owner - proprietario* che può autorizzare altri utenti all’accesso, definire il tipo di accesso da concedere (grant) ad altri (lettura, scrittura, esecuzione, aggiunta dati - append), definire accessi selettivi per esempio, basati su nome, contenuto, parametri di sistema, storia, aggregazione di dati.

I sistemi gestiti in base a politiche mandatorie (*Mandatory Access Control - MAC*) sono adatti per database con dati sensibili (governativi, militari, sanitari, ...) dove DAC non è sufficiente perché sono archiviate informazioni vitali, con diversi livelli di sensitività, e dove sono necessari anche controlli sul flusso di dati e sul rilascio di dati e controlli sugli utenti. Caratteristica di questi sistemi sensibili è che essi subiscono attacchi sofisticati da parte di utenti determinati (per esempio, attacchi basati su Trojan Horse). Le politiche MAC classificano gli oggetti, assegnando loro un *livello di sensitività* e i soggetti, assegnando loro un *livello di clearance*.

Sono definite inoltre le classi di sicurezza (Security Class - SC), come:

$$SC = (\text{componente gerarchica, insieme-di-categorie})$$

dove la componente gerarchica individua il livello (tra livelli esiste un ordinamento (parziale) e le categorie individuano le aree di appartenenza dei dati (es., dipartimenti o aree applicative). Tra le SC è definita la relazione di dominanza (si veda per esempio [33] per i database di tipo MAC).

I meccanismi di sicurezza che implementano politiche MAC devono garantire che tutti i soggetti abbiano accesso solo ai dati per cui possiedono la clearance appropriata. Le regole sono base sono:

1. *No-Read-Up* (per operazioni di lettura) detta Simple Security property:  
Un soggetto  $S$  è autorizzato ad accedere in lettura a un oggetto  $O$  solo se  $L(S) \geq L(O)$ .

2. *No-Write-Down* (per operazioni di scrittura, che potrebbero degradare i dati, quando questi vengono scritti, per esempio, su un file classificato più basso del livello di clearance di chi scrive). Questa è detta *\*-property* (Star Property): un soggetto S è autorizzato ad accedere in scrittura a un oggetto O solo se  $L(S) \leq L(O)$ .

Queste regole, combinate, impediscono *flussi di dati* tra soggetti “high” e “low”.

Tipici livelli di classificazione (sensitività dati e clearance utenti) sono: Unclassified (U), Confidential (C), Secret (S), Top Secret (TS).

Esempi di categorie sono: 1. Nuclear, NATO, Intelligence, oppure 2. Produzione, Personale, Engineering.

In Figura 11.14 sono riportati esempi di confronti fra livelli e fra SC. Sulla sinistra, si vede la relazione “domina” fra SC, sulla destra è riportata la gerarchia dei livelli, da Unclassified fino a Top Secret.

L’implementazione di protezione in DBMS MAC per l’accesso controllato ai dati multi-livello usa una Trusted Computing Base, ovvero un insieme di componenti hardware e software fidati per la memorizzazione e gestione delle etichette di classificazione direttamente nel database, eventualmente in forma cifrata oppure in un database separato gestito da appositi filtri. Variano le prestazioni nei vari casi.

### DAC e DBMS

In Figura 11.15 si vede come si può realizzare il controllo di accesso basato sul nome della tabella EMP ivi definita e dei suoi attributi in base al ruolo utente.

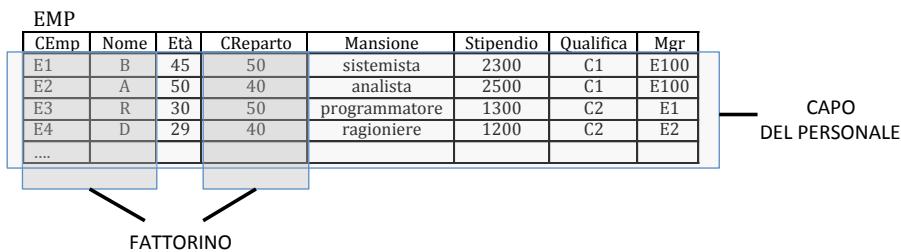


Figura 11.15: Accesso basato sul ruolo (DAC)

Le regole di accesso possono essere specificate nel profilo utente, oppure associate alla descrizione dell’oggetto. I meccanismi base per la specifica di autorizzazioni possono essere inseriti nel Query Language del DBMS o nei comandi di SO.

In Figura 11.16 è mostrato il controllo di accesso basato sul contenuto di EMP, ombreggiato in grigio. In questo caso una query definisce la vista di protezione D50 contenente i soli dati degli impiegati del reparto con CReparto=50, vista che è poi concessa (grant) in lettura all’utente “John”.

| EMP  |      |     |          |          |           |           |     |  |
|------|------|-----|----------|----------|-----------|-----------|-----|--|
| CEmp | Nome | Età | CReparto | Mansione | Stipendio | Qualifica | Mgr |  |
| E1   | B    | 50  |          |          |           |           |     |  |
| E3   | R    |     | 50       |          |           |           |     |  |
| ...  |      |     | 50       |          |           |           |     |  |

John può vedere solo CEmp, Nome, CReparto  
degli impiegati del reparto 50

Figura 11.16: Accesso basato sul contenuto (DAC)

Per esempio, in un DBMS relazionale possiamo avere:

```
GRANT INSERT, SELECT ON EMP TO John
```

La clausola GRANT attribuisce i diritti di effettuare operazioni di inserimento (INSERT) e interrogazione (SELECT) sulla tabella EMP all'utente John.

In alternativa, se viene definita una vista D50 sulla relazione EMP come in Figura 11.16 e si possono specificare i diritti d'accesso sulla vista come segue:

```
CREATE VIEW D50(CEmp, Nome, CReparto) AS
    SELECT CEmp, Nome, CReparto
    FROM EMP
    WHERE CReparto=50
```

```
GRANT SELECT ON D50 TO John
```

Altro meccanismo è quello della *query modification*, in cui la query viene modificata rispetto alla richiesta originale dell'utente, per consentire l'accesso solo ai dati per cui l'utente ha l'autorizzazione. La modifica aggiunge ulteriori restrizioni rispetto alla query originale per rispettare le regole definite per l'accesso ai dati.

Se viene definito il seguente diritto di accesso:

```
GRANT SELECT ON EMP TO John WHERE Stipendio<1500
```

Quando John formula la richiesta:

```
SELECT * FROM EMP
```

Il DBMS risponderà eseguendo la seguente interrogazione modificata per rispettare la regola di accesso definita:

```
SELECT *
FROM EMP
WHERE Stipendio < 1500
```

È ora possibile illustrare il modello a sei componenti in ambienti DAC:

$$(gtor, gtee, obj, right, constraint, prop\_rule)$$

dove *prop-rule* è la regola di propagazione dei privilegi, che specifica se un privilegio può essere concesso (e a quali condizioni) ad altri utenti in ambienti DAC. Si può avere: 1. nessuna propagazione, 2. propagazione incondizionata (Unbounded Propagation), 3. propagazione limitata (Bounded Propagation, per esempio per una catena limitata di utenti o per un massimo di tempo).

In tutti i casi, è da affrontare il problema della *revoca dei privilegi* concessi, che deve avvenire per i privilegi che un utente ha ricevuto per esempio in modo multiplo da utenti diversi (si rimanda a [9, 33] per dettagli sul problema della revoca).

I meccanismi DAC nei DBMS possono essere basati sui *ruoli*, per esempio, i ruoli utente in un'organizzazione (manager, capo del personale, impiegato, magazziniere, fattorino, ecc.). Tramite il concetto di ruolo, le autorizzazioni sono associate al ruolo anziché ai singoli utenti, agevolando così la scrittura e manutenzione dell'insieme delle regole di accesso, che poi vengono memorizzate su un archivio apposito di sistema per essere accedute al momento della decisione di concedere o meno un accesso. Si ottiene così flessibilità nella gestione di autorizzazioni e aderenza alla struttura organizzativa.

## MAC e DBMS

Per la gestione di database MAC si realizzano tavole relazionali multi-livello con diversa granularità di classificazione dei dati: livello di relazione, attributo, record (tupla), elemento (dato). Meccanismo fondamentale per la realizzazione è la *polinstanziazione*, ovvero l'istanziazione (esistenza di tante copie fisiche della stessa tabella) multipla di tavole fisicamente separate a seconda della loro classificazione di sicurezza. Innanzitutto, la TCB (Trusted Computing Base) esegue la mediazione di tutti gli accessi soggetti-oggetti in modo “trusted”, ovvero mediante componenti che non possono essere manomessi (concetto di Reference Monitor).

La polinstanziazione avviene mediante regole per la classificazione degli elementi nel database in cui una *Multi-Level Relation* (MLR) è definita come una relazione R:

$$R(A1, C1, \dots, An, Cn, TC)$$

in cui *TC* è l'attributo Tuple Classification, che assegna il livello alla riga (tupla) della tabella e  $C_i$  è la sensitività relativa all'attributo  $A_i$ .

L'accesso a una MLR avviene come segue:

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

Figura 11.17: Esempio di relazione multilivello con dati classificati con metodo MAC

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

Figura 11.18: Istanza di livello Top Secret

- Lettura: un soggetto può accedere in lettura alle istanze di una MLR al proprio livello o al di sotto.
- Scrittura: può generare polistanziazione se l'elemento da inserire esiste già con quel nome. In particolare, se il soggetto richiedente ha una classificazione più bassa o non comparabile rispetto a quella dell'elemento da scrivere (elemento non visibile), la polistanziazione evita l'inferenza illecita di dati, a partire da dati acceduti lecitamente. Se il soggetto richiedente ha una classificazione che domina quella dell'elemento da inserire, la polistanziazione evita la negazione di servizio.

Un esempio di relazione in cui i dati sono classificati secondo il metodo MAC è mostrato in Figura 11.17. Ogni attributo ha un corrispondente attributo che ne descrive la sensitività (CNomeAttributo). La prima tupla contiene dati secret e globalmente è etichettata Secret (S). Le altre due tuple sono ciascuna etichettata al massimo dei livelli ivi contenuti, ovvero Top Secret (TS), in quanto contengono almeno un attributo Top Secret.

Per la relazione illustrata nella Figura 11.17 vengono create le istanze corrispondenti alle diverse classi di accesso, in questo caso Top Secret (Figura 11.18) e Secret (Figura 11.19).

L'istanza di livello Top Secret contiene l'intero contenuto della tabella rappresentata in Figura 11.17 mentre nell'istanza Secret di Figura 11.19, che contiene l'istanziazione della tabella con livello di clearance Secret, non è presente un valore per lo stipendio di Ann.

Nelle relazioni multilivello il valore nullo può avere quindi due diversi significati: essere un effettivo valore nullo, oppure contenere valore di attributi che hanno una classificazione superiore a quella dell'istanza.

Poiché in alcuni casi dall'assenza di un valore è possibile inferire la presenza di altri valori (e quindi ad esempio in questo caso è possibile inferire la presenza

| Name | CName | Dept  | CDDept | Salary | CSalary | CTuple |
|------|-------|-------|--------|--------|---------|--------|
| BoB  | S     | Dept1 | S      | 10 K   | S       | S      |
| Ann  | S     | Dept2 | S      | -      | S       | S      |

Figura 11.19: Istanza di livello Secret (S)

| Name | CName | Dept  | CDDept | Salary | CSalary | CTuple |
|------|-------|-------|--------|--------|---------|--------|
| BoB  | S     | Dept1 | S      | 10 K   | S       | S      |
| Ann  | S     | Dept2 | S      | 20 K   | TS      | TS     |
| Ann  | S     | Dept2 | S      | 15 K   | S       | S      |
| Sam  | TS    | Dept2 | TS     | 20 K   | TS      | TS     |

Figura 11.20: Esempio di relazione multilivello con polistanziazione

di valori TS per Ann), nell’istanziazione Secret della tabella è possibile inserire valori fintizi per il valore TS, compatibili con il valore TS (ad esempio uno stipendio di 10K anziché il valore TS di 20K).

Un’alternativa è quella di avere una polinstanziazione nella tabella, che permette di avere righe con la stessa chiave primaria. In questo caso si possono anche avere due tuple con la stessa chiave primaria caratterizzate da diverso livello di sensitività come rappresentato in Figura 11.20. In questo caso l’istanziazione Top Secret fornisce per lo stipendio di Ann un valore di 20K, mentre quella Secret il valore 15K.

L’istanza Top Secret conterrà entrambe le tuple (polistanziazione) e avrà lo stesso contenuto della Figura 11.20.

### Esempio di attacco alle basi di dati: SQL Injection

Ad esempio, volendo usare un database per autenticare gli utenti tramite la seguente query:

```
SELECT username, password FROM users
WHERE username = 'inputUsername' AND password = 'inputPassword'
```

si ottiene che se la query restituisce una tupla allora l’utente è autenticato altrimenti no. Un problema consiste nel comprendere che cosa succede se l’utente inserisce come input uno Username valido e come Password la seguente stringa:

```
'UnaStringa' OR 'a'='a'
```

La query elaborata sarebbe:

```
SELECT username, password FROM users
WHERE username = 'inputUsername'
      AND password = 'UnaStringa' OR 'a'='a'
```

L’utente viene comunque autenticato perché la query restituisce sempre almeno una riga. Questo è un esempio di attacco SQL injection.

## 11.5 Meccanismi di sicurezza infrastrutturali

Gli strumenti in grado di fornire contromisure rispetto agli attacchi a un sistema informativo a livello di architettura fisica consistono principalmente nel realizzare una configurazione sicura del sistema operativo, dei client e dei server e nell’attuare, tramite adeguati *componenti architetturali*, politiche di autenticazione degli utenti, di controllo degli accessi, e di log delle attività.

In questa sezione si illustrano le componenti architetturali per la sicurezza, più precisamente firewall e/o proxy server, collocati in corrispondenza dei collegamenti con reti pubbliche esterne al sistema per filtrare gli accessi, e sistemi di Intrusion Detection in grado di monitorare gli eventi che avvengono nel sistema e sulla rete.

In particolare, un Firewall è un insieme di componenti hardware e/o software che collegano una rete fidata a reti insicure. Esso controlla, implementando opportune policy di sicurezza, il traffico in entrata e in uscita dalla rete sicura, con lo scopo di prevenire, rilevare e annullare eventuali attacchi e richieste non autorizzate. Un Intrusion Detection System (IDS) è un componente hardware e/o software che attua il processo di monitoraggio degli eventi di un sistema, o di una rete di calcolo, con obiettivo l’individuazione delle intrusioni, definite come tentativi di compromettere la confidenzialità, l’integrità o la disponibilità del sistema.

### 11.5.1 Firewall

Con il termine *firewall* si identifica in modo generico un insieme di componenti e di servizi finalizzati a controllare e limitare il traffico fra una rete da proteggere e l’esterno. Gli schemi architetturali secondo i quali le componenti di un firewall (principalmente router ed application gateway) possono essere disposte per proteggere una rete sono molteplici, e vanno valutati dal progettista sulla base di vari fattori, tra i quali sicurezza, flessibilità rispetto al controllo di nuovi protocolli ed applicazioni, prestazioni del sottosistema firewall e della rete nel suo complesso. Hanno ovviamente rilievo anche il costo di installazione e la manutenzione.

I firewall rappresentano la prima linea di difesa per la sicurezza della rete. Costituiscono una barriera tra le reti interne (Intranet e/o Extranet) sicure e le reti esterne che possono essere anche non affidabili, come Internet. Installare un firewall non significa essere sicuri: un firewall installato e non configurato è inutile. L’aumento del grado di sicurezza portato da un firewall dipende dalla correttezza delle regole specificate. Inoltre, l’adozione di un firewall deve seguire tre leggi principali:

- Il firewall deve essere l’unico punto di contatto tra la rete interna e l’esterno.

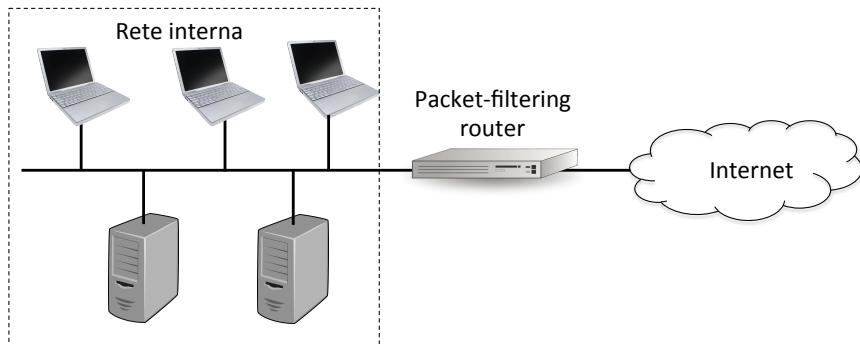


Figura 11.21: Architettura Firewall con Packet filtering

- Tutti i pacchetti devono essere bloccati, tranne quelli “autorizzati” (secondo la politica aziendale).
- Il firewall deve essere un sistema “sicuro” a sua volta.

I componenti di base che permettono il monitoraggio del traffico da parte di un firewall sono di due tipi: i) Screening router o Packet filtering e Packet-Inspection e ii) Application gateway. Esistono poi varie configurazioni per i firewall come la configurazione Dual-homed host, Bastion host, Screened host gateway, e diverse altre.

I firewall *Screening router* sono dispositivi in grado di bloccare o instradare i pacchetti in transito fra la rete interna e quella esterna, sulla base della loro intestazione (packet-filtering) o del loro contenuto (packet-inspection). Si tratta in ogni caso di dispositivi operanti a livello rete e/o trasporto e quindi trasparenti rispetto allo strato applicativo.

Anche i firewall sono soggetti ad attacchi che mirano a riconfigurare le regole di filtraggio dei pacchetti. Per la protezione dei firewall occorre che eventuali funzioni di configurazione remota siano disabilitate. Altri attacchi ai firewall sono tesi a modificare le tabelle interne di routing ed aggirare in tal modo eventuali altri firewall posti a difesa della rete interna. Questo tipo di attacchi possono ad esempio essere condotti attraverso falsi pacchetti Internet Control Message Protocol (ICMP) - protocollo di servizio per reti a pacchetto che trasmette informazioni riguardanti i malfunzionamenti, informazioni di controllo o messaggi tra i vari componenti di una rete di calcolatori; pertanto i firewall vanno configurati in modo da disabilitare le funzioni di dynamic routing.

Più in generale, i firewall basati sul *Packet filtering* valutano i pacchetti sulla base della loro intestazione (vedere Figura 11.21). Si tratta di dispositivi configurabili tramite un elenco ordinato di regole che permettono o negano il passaggio di un pacchetto in base alle informazioni contenute nell'header del messaggio (indirizzo IP e porta sorgente, indirizzo IP e porta destinazione, TCP flag, IP options e così via). Molti produttori di router applicano le regole di filtraggio solo sui pacchetti in uscita dal router, non su quelli in ingresso. In questo

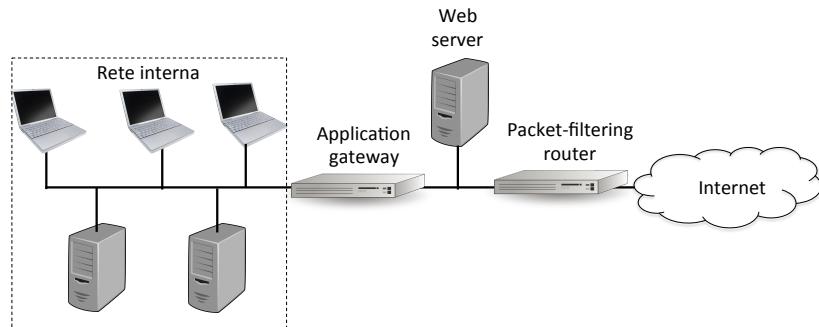


Figura 11.22: Architettura Firewall con Application Gateway

modo, tuttavia, pacchetti generati artificiosamente e aventi come indirizzo sorgente quello di un host della rete interna sarebbero instradati dal router sulla rete interna come se fossero effettivamente provenienti dall'host oggetto dello spoofing (tipo di attacco che impiega in vari modi la falsificazione dell'identità - spoof).

Verificando i pacchetti anche in ingresso, il router può rilevare che un pacchetto con indirizzo sorgente uguale a quello di un host interno non può provenire dalla rete esterna, e deve quindi essere bloccato. I problemi di questa soluzione risiedono nella capacità di individuare un insieme di regole corretto e completo rispetto alle esigenze della sicurezza aziendale in esame. Occorre inoltre individuare eventuali errori nella configurazione, aspetto che può risultare difficile per la mancanza di strumenti automatici di verifica, e per la incompletezza dei log generati dal sistema di sicurezza.

I firewall di tipo *Packet inspection*, in particolare, valutano i pacchetti sulla base del loro payload. Si tratta di dispositivi in grado di capire un insieme predefinito di protocolli che viaggiano incapsulati nel protocollo IP, e in alcuni casi controllare lo stato delle relative sessioni (*stateful inspection*). A fronte della loro complessità, offrono grande flessibilità nella definizione delle regole di filtraggio (dette *smart rule*) che, operando a livello di sessione nel protocollo TCP (session filtering), sono paragonabili a quelle offerte dagli application gateway. Rispetto a questi ultimi, però, non hanno funzionalità proxy (ovvero di sistema intermediario per le richieste da parte dei client verso risorse su altri server, disaccoppiando l'accesso al web dal browser). In particolare, non ricostruiscono i pacchetti, ma si limitano a lasciar transitare quelli che superano le regole di filtraggio.

I firewall di tipo *Application gateway* offrono attualmente il maggior grado di controllo del traffico fra una rete interna e la rete Internet. Sono in genere realizzati tramite un host opportunamente configurato con software specifico in grado di erogare, accanto ai servizi di packet-filtering e packet-inspection (visti negli screening router), anche i cosiddetti servizi di *application proxy*. Un application proxy è un processo che, eseguito sul gateway, si pone a livello di applicazione nella comunicazione fra componenti di una specifica applicazione.

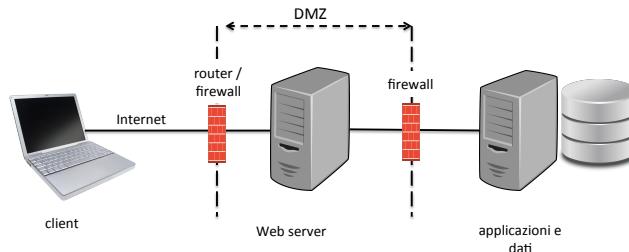


Figura 11.23: Firewall in una architettura 3-tier con server web in DMZ

per la quale è stato progettato (vedere Figura 11.22). Esempi di application proxy commercialmente disponibili sono quelli per il controllo del traffico telnet, FTP e HTTP. Per esempio, nel caso di applicazioni client-server, un application proxy comunica con il client simulando il server, e viceversa, comunica con il server simulando il client; quindi, i pacchetti non viaggiano direttamente fra client e server. Il flusso dei pacchetti provenienti dalle parti viene intercettato dal proxy, interpretato a livello applicativo, ricostruito sulla base della conoscenza del protocollo e instradato verso il comunicante.

Rispetto ai Packet-inspection router, gli Application gateway (o meglio, gli application proxy in esecuzione su di essi), presentano vantaggi e svantaggi. Ricostruendo il traffico, un application gateway minimizza le probabilità di successo di attacchi basati su debolezze di un server rispetto a particolari sequenze di messaggi non previste dal protocollo. La ricostruzione dei pacchetti in uscita richiede un tempo di analisi che rende un application gateway più lento di un packet-inspection router. Inoltre, operando a livello di applicazione, un application gateway è in grado di controllare un numero più ridotto di protocolli, e richiede la installazione e la configurazione di un nuovo proxy per ciascun nuovo protocollo da gestire.

Due esempi di architetture a tre e a cinque livelli per sistemi informativi basati su web in cui sono stati inseriti dei firewall sono riportati nelle Figure 11.23 e 11.24. La zona che si trova tra il firewall che riceve il traffico dalla rete esterna e il firewall a valle del web server che filtra il traffico verso la rete interna viene chiamata *DMZ (DeMilitarized Zone)*. La DMZ viene considerata comunque una zona esposta ad attacchi di sicurezza. La DMZ ha lo scopo di aggiungere un livello ulteriore di sicurezza alla rete esterna. Infatti, la rete esterna può accedere solo alle risorse esposte nella DMZ e il resto della rete dell'organizzazione è dietro il secondo firewall. La DMZ funziona come una sottorete isolata posta tra Internet e la rete privata. Il nome DMZ deriva dal termine zona demilitarizzata, un'area posta tra le nazioni in cui non sono messe in opera operazioni militari. In Figura 11.24 viene mostrato come possono essere creati diversi livelli di sicurezza per proteggere i diversi livelli della architettura.

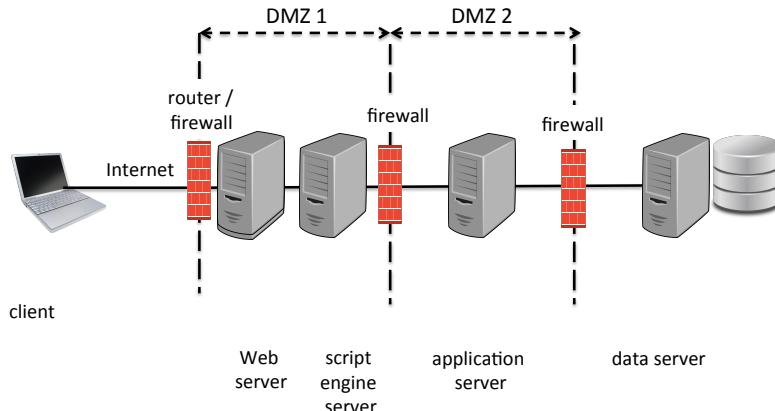


Figura 11.24: Architettura 5-tier con tre firewall

### 11.5.2 Intrusion Detection Systems - IDS

Con il termine *intrusione* si intende un insieme di azioni volte a compromettere una o più proprietà di sicurezza (principalmente, integrità, riservatezza, disponibilità) di un sistema o più in generale di risorse di calcolo o di rete. Si definisce *Intrusion Detection (ID)* il processo di monitoraggio degli eventi di un sistema, o di una rete, cercando tracce evidenti (es., log) di intrusioni [39]. *Intrusion prevention* è l'estensione del concetto di ID con aspetti di controllo dell'accesso volti a proteggere i sistemi da tentivi di sfruttamento malevolo di risorse (exploitation). *IDS*, infine, sono i sistemi che automatizzano il processo di Intrusion Detection, identificando intrusioni di utenti non autorizzati (crackers) e di utenti autorizzati (insider threaters).

Talvolta gli IDS interagiscono con i firewall. Per alcune intrusioni, ad esempio, un IDS può richiedere la chiusura delle connessioni attive, impedire qualsiasi altra connessione dall'indirizzo IP dell'aggressore e notificare all'amministratore che un probabile attacco è in atto.

Le principali assunzioni su cui si basano gli IDS sono:

- Le attività di sistema sono osservabili.
- Le attività normali e quelle intrusive hanno evidenze diverse.
- Da una prospettiva algoritmica, si usano Features (che catturano le evidenze di intrusione) e Modelli (che permettono di comporre tali evidenze).
- Da una prospettiva architettonica, sono necessari vari componenti: processori per i dati di audit, una base di conoscenza, un 'decision engine', e generatori di allarmi e risposte.

La Figura 11.25 mostra l'architettura di base di un IDS.

Le componenti funzionali di un IDS sono:

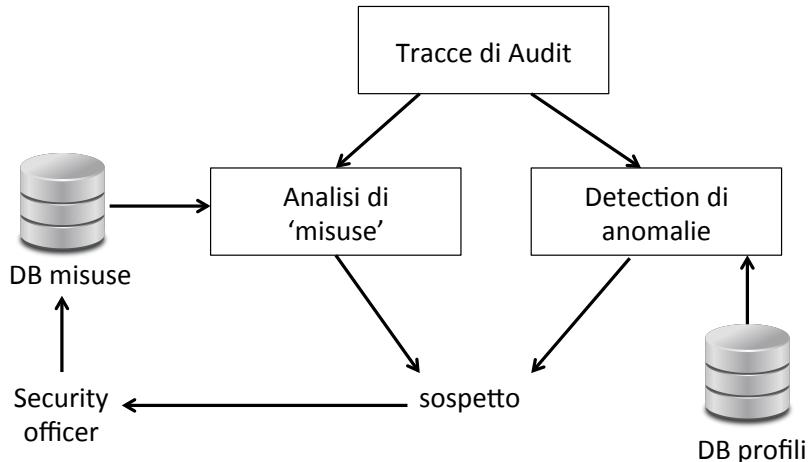


Figura 11.25: Architettura semplificata di un IDS

- un insieme Sensori (Information Source), che si occupano di eseguire la raccolta di elementi (per esempio, tracce di log) da cui è possibile rilevare le tracce di un'intrusione;
- algoritmi di analisi, che implementano l'ID sulla base degli eventi raccolti dai sensori, stabilendo se un attacco è effettivamente in atto o ha già avuto luogo;
- componenti Alerting and Response: componenti attivi che, quando un attacco è in atto, intraprendono azioni che vanno dal report degli allarmi ad azioni più complesse (chiusura di una porta TCP).

Una possibile *classificazione degli IDS* avviene in base alle Information Source.

1. Un *Network-based IDS* è un tipo di IDS che rileva gli attacchi analizzando il traffico di rete (per esempio, i pacchetti) per analizzare se qualche pacchetto malicioius è in transito con lo scopo di disattivare l'IDS stesso. I requisiti per tale IDS richiedono che i sensori siano distribuiti sulla rete.

I vantaggi risiedono nel fatto che sono sufficienti pochi sensori per monitorare una rete anche di grandi dimensioni e che l'introduzione dell'IDS non ha un impatto eccessivo sulla configurazione e sulle prestazioni della rete preesistente. Gli svantaggi sono la poca efficienza in situazioni di congestione di rete, il fatto che tali IDS non riescono ad analizzare protocolli cifrati e che non riescono a rilevare se un attacco ha avuto successo. Infine, si osserva che alcuni attacchi che creano pacchetti anomali (o malformati) possono evadere i controlli dell'IDS.

Fra i tool che realizzano gli IDS Network-based, vi è Ethereal come open source GUI-based packet viewer. Oppure si menziona tcpdump, un pacchetto “packet capture” ([www.tcpdump.org](http://www.tcpdump.org)) per codice con varie interfacce per sistemi operativi diversi e per vari tipi di sensori.

2. Un *Host-based IDS* è un software (molto simile ai software antivirus) da installare su ogni macchina da proteggere. Tale software opera sulle informazioni raccolte su singoli host a livello di sistema operativo oppure di applicazione. Esso è in grado di analizzare gli attacchi con grande precisione, determinando quale processo e quale utente è stato coinvolto. Questi IDS possono inferire l'esito degli attacchi. Quindi, riassumendo, i vantaggi consistono nel fatto che questi IDS sono in grado di rilevare attacchi non rilevabili dagli IDS network-based e possono essere utilizzati anche su dati cifrati. Possono inoltre rilevare Trojan Horse e attacchi che violano l'integrità del software. Come svantaggi, si menzionano la difficoltà di gestione di tali sistemi (specie la loro scalabilità) e il fatto che questi IDS possono essere resi inefficaci o disabilitati da alcuni attacchi di tipo Denial-of-Service. Sono anche alti gli impatti sulle prestazioni del sistema operativo.

Un'altra *classificazione degli IDS* avviene in base al tipo di analisi:

1. L'analisi di tipo *misuse detection* è volta ad individuare i comportamenti già conosciuti noti come attacchi (per esempio, tramite tecniche di pattern matching). Essi analizzano l'attività del sistema confrontandola con pattern detti *signature* di comportamento relativi ad attacchi noti. I vantaggi di queste soluzioni è che tali IDS rilevano gli attacchi senza generare un numero eccessivo di falsi allarmi e possono rilevare tempestivamente l'uso di un tool o di qualche tecnica che genera uno specifico attacco. Gli svantaggi risiedono nel fatto che sono rilevabili solo attacchi conosciuti e che le signature vanno aggiornate di frequente. Purtroppo, questi IDS spesso non sono in grado di identificare le varianti di attacchi noti e richiedono frequenti patch di manutenzione ed evoluzione per restare al passo con l'evoluzione dei pattern di attacco.
2. L'analisi di tipo *anomaly detection* invece individua deviazioni significative dal comportamento normale. Questi IDS cercano di identificare comportamenti anomali negli host o nella rete. Si basano sull'ipotesi che un sistema sotto attacco si comporta in modo diverso rispetto a quando funziona normalmente. Essi costruiscono profili che rappresentano il normale comportamento di utenti, rete e host, a partire da dati storici. Definiscono poi un insieme di indicatori aggregati utilizzati per stabilire la deviazione dal comportamento normale del sistema. Per ogni indicatore, vengono stabilite soglie di normalità, al di fuori delle quali l'IDS genera un allarme. Questi IDS possono rilevare attacchi senza conoscerne pienamente i dettagli e possono essere utilizzati per definire le signature per i sistemi misuse detection. Purtroppo, essi generano molti di falsi allarmi a causa

dell'imprevedibilità degli utenti di rete e richiedono un vasto training di apprendimento del normale profilo di comportamento. Non sono in grado di riportare dettagli sul tipo di attacco rilevato.

Infine, si definisce *Response* la reazione dell'IDS quando l'analisi ha rilevato un attacco. Si hanno due tipi di response:

- *Active response*: consiste nell'acquisizione di maggiori informazioni aumentando il livello di sensitività delle sorgenti di informazione; in alternativa l'IDS può modificare l'ambiente bloccando l'attacco in corso tramite azioni di reset delle connessioni, o di riconfigurazione di router e firewall.
- *Passive response*: l'IDS eleva una notifica all'amministratore di sistema riguardo all'attacco in corso, lasciando agli operatori le decisioni su quali azioni intraprendere.

## 11.6 Domande

- Definire le proprietà di sicurezza dei sistemi informativi
- Illustrare il meccanismo di generazione e verifica della firma digitale.
- Considerare il requisito di integrità e discutere le tecniche per prevenire possibili attacchi di violazione di tale proprietà.
- Illustrare i principali tipi di controllo degli accessi alle basi di dati.
- Descrivere le tecniche crittografiche: tipi di algoritmi, ruolo delle chiavi crittografiche, applicazioni della crittografia.
- Illustrare i principali tipi di controllo accessi alle basi di dati.
- Definire le proprietà di sicurezza e indicare quali possono essere garantite attraverso la crittografia e con quali tecniche.
- Illustrare le differenze tra crittografia simmetrica e asimmetrica.
- Illustrare il funzionamento della firma digitale.
- Scopo di un certificato digitale e relazione con una PKI (Public Key Infrastructure).
- Certificati digitali: illustrare le funzionalità, gli ambiti in cui vengono utilizzati e l'infrastruttura che ne permette l'emissione e la gestione.
- Caratteristiche degli algoritmi di hash e loro ruolo nella verifica/validazione di una firma digitale.
- Illustrare le funzionalità di firewall e IDS.



# Capitolo 12

# ArchiMate

## 12.1 Introduzione ad ArchiMate

Nel presente capitolo si illustra ArchiMate<sup>®</sup>, un linguaggio di modellazione di Enterprise Architecture proposto da The Open Group<sup>1</sup> [14, 38, 23].

Come già discusso in precedenza, ad esempio nel framework di Zachman e nei vari aspetti di BOAT, anche in ArchiMate la descrizione dell'architettura viene presentata secondo diversi punti di vista, che consentono di rappresentare un sistema con diagrammi diversi collegati tra loro.

Nella Figura 12.1 sono illustrati i livelli (layer) e gli aspetti principali di ArchiMate (ArchiMate core), che verranno discussi nel seguito:

- *Livelli (layer)*: Business Layer, Application Layer, Technology Layer. I layer di ArchiMate rappresentano l'architettura dell'impresa a livelli, in cui i livelli superiori sono basati su servizi forniti dai livelli inferiori. Il livello *Business* fornisce prodotti e servizi ai clienti esterni. Il livello *Applicativo* supporta il livello di business con servizi forniti da componenti applicativi. Il livello *Tecnologico* offre servizi infrastrutturali (di elaborazione, memorizzazione, comunicazione) necessari per eseguire le applicazioni, realizzati tramite dispositivi di elaborazione e comunicazione e da software di sistema.

In genere, nei diagrammi di modellazione si utilizzano per gli elementi rappresentati il giallo per il Business Layer, il blu per l'Application layer e il verde per il Technology Layer.

- *Aspetti*: struttura passiva, comportamento e struttura attiva. Per ciascun livello, si definiscono tre aspetti. Il *comportamento* rappresenta le azioni effettuate dal sistema; analogamente a quanto già visto nella Sezione 9.4.3 il comportamento può essere modellato a vari livelli nelle architetture di integrazione. La *struttura passiva* (o statica) definisce gli oggetti del

---

<sup>1</sup><http://www.opengroup.org/>

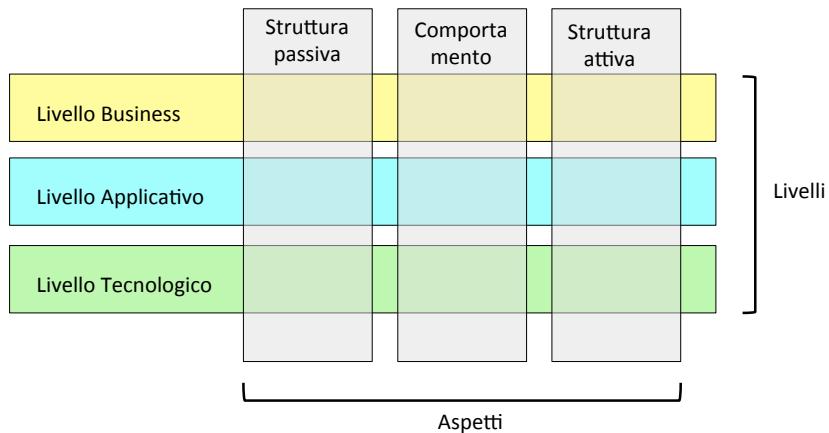


Figura 12.1: Aspetti e layer di ArchiMate: elementi core (fonte: The Open Group, 2016)

comportamento (ad esempio, un ordine, oppure un file eseguibile, a seconda del livello). La *struttura attiva* definisce gli elementi che forniscono il comportamento e sono quindi il soggetto dell’azione (ad esempio, attori a livello di business, dispositivi a livello infrastrutturale).

In questo capitolo viene illustrato l’approccio di ArchiMate in generale, ma l’attenzione verrà posta soprattutto sulla rappresentazione dell’infrastruttura tecnologica.

Infatti, se si considerano gli aspetti di BOAT (Business, Organizzazione, Architettura funzionale e Tecnologia), abbiamo visto alcuni modelli per rappresentare i primi tre. Vi è l’esigenza di una rappresentazione anche del quarto aspetto, quello dell’infrastruttura tecnologica, ed è indispensabile collegarlo agli altri aspetti, per la realizzazione di una EA completa.

Rispetto a BOAT, possiamo vedere le possibili corrispondenze tra le rappresentazioni viste in BOAT e quelle di ArchiMate nella Figura 12.2.

ArchiMate è un approccio orientato ai servizi: i componenti ai vari livelli (B,A,T) forniscono servizi ai componenti del livello superiore. L’approccio alla modellazione in ArchiMate basato sui servizi è illustrato nella Figura 12.3. Nella figura si vede come i servizi si alternano ai componenti che li realizzano: ad esempio i servizi applicativi sono realizzati da componenti applicativi. A loro volta, i componenti applicativi utilizzano servizi infrastrutturali. La relazione tra i servizi e i componenti che li realizzano viene chiamata *realisation relation*, mentre la relazione tra i componenti e i servizi necessari al loro funzionamento è definita come *serving relation*. Si vede quindi che il modello fornisce servizi di business ai clienti, realizzati da processi di business, che a loro volta utilizzano servizi applicativi, e così via.

Nelle prossime sezioni si illustrano prima gli elementi principali di ArchiMate (vers. 3.0) nella Sezione 12.2, seguiti da alcuni esempi e successivamente nella

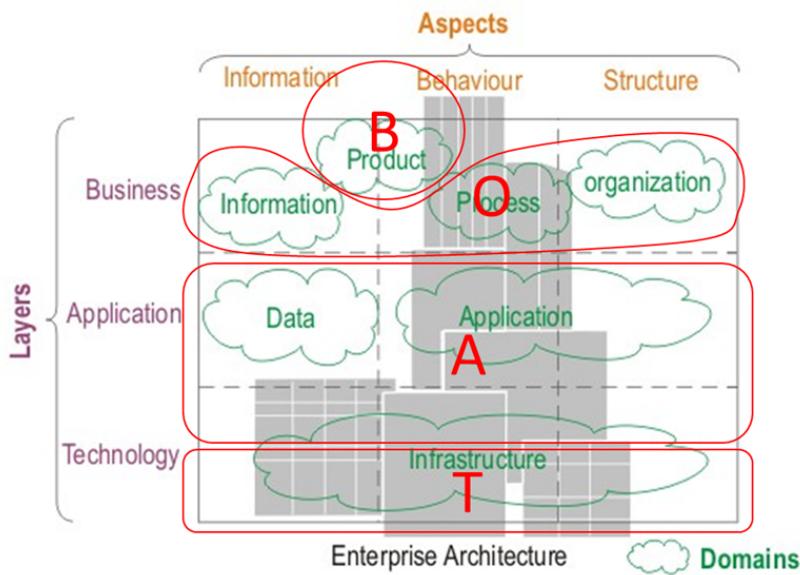


Figura 12.2: Rappresentazione degli aspetti di BOAT e ArchiMate

Sezione 12.4 si illustra in dettaglio la rappresentazione delle diverse possibili scelte tecnologiche presentate nei capitoli precedenti. Per approfondimenti si rimanda alle specifiche [38] e ai testi [23, 38].

## 12.2 Elementi principali del modello

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

### 12.2.1 Componenti

I componenti nel modello sono gli elementi che vengono utilizzati per descrivere la struttura del sistema. Per ciascun livello è definito un insieme di componenti che lo caratterizza, come illustrato nel seguito.

Graficamente, i componenti che corrispondono a elementi passivi (oggetti) e attivi (soggetti) vengono rappresentati con rettangoli, mentre i componenti che corrispondono a comportamenti sono disegnati con angoli arrotondati. I componenti che caratterizzano un livello sono in genere associati a un simbolo grafico rappresentato nel rettangolo in alto a destra e possono avere anche una rappresentazione grafica alternativa (nel presente testo si utilizzerà solo la prima rappresentazione, con rettangoli).

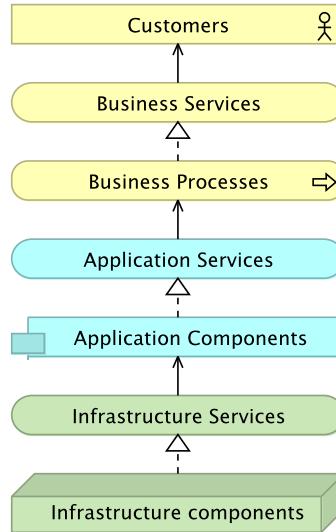


Figura 12.3: Servizi e ArchiMate (rielaborazione da [23])

### 12.2.2 Relazioni

Le relazioni definiscono i possibili collegamenti tra gli elementi e vengono illustrate nella Figura 12.4. In generale, le tipologie delle relazioni sono le stesse per tutti i livelli e si distinguono in relazioni strutturali e di dipendenza tra i componenti.



Figura 12.4: Relazioni

Le principali relazioni definite in ArchiMate sono le seguenti:

- *Accesso*: lega un componente comportamentale a un componente passivo o oggetto, ad esempio, una funzione applicativa che accede a un dato. È una relazione di dipendenza.
- *Composizione*: come in UML, lega un elemento a suoi componenti, i quali non possono esistere indipendentemente. È una relazione strutturale.

- *Realizzazione*: indica che l'elemento dal lato senza freccia crea l'elemento dall'altro lato, ad esempio un componente applicativo realizza un servizio corrispondente al livello di business. È una relazione strutturale.
- *Assegnamento*: 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.

Le relazioni strutturali e di dipendenza non sono tutte allo stesso livello e possono essere ordinate secondo la loro forza nel collegare gli elementi tra loro, come verrà illustrato in dettaglio nel seguito (Sezione 12.4.2).

### 12.2.3 Nodi

È possibile utilizzare un nodo (Figura 12.5) per modellare un'infrastruttura hardware e software in modo più o meno dettagliato.

Si può definire una relazione di nidificazione attraverso le relazioni di composizione, aggregazione, assegnazione e realizzazione. In questo modo, un nodo può modellare diversi livelli di astrazione.

Nella Figura 12.5, vengono fornite tre rappresentazioni a un diverso livello di astrazione di un componente infrastrutturale che contiene un server SQL, il sistema operativo e un software per la condivisione di file. Le tre rappresentazioni sono sostanzialmente equivalenti, in quanto rappresentano un database server. Nelle tre rappresentazioni, quello che cambia è il livello di dettaglio fornito dalla rappresentazione del database server nel mostrare i servizi forniti e i suoi componenti hardware e software e le loro relazioni.

Nella Figura 12.5a vengono illustrate le relazioni tra tutti i singoli elementi che compongono il database server, mentre nella Figura 12.5b vengono raggruppati in un nodo gli elementi corrispondenti al software SQL server e al sistema operativo sul device, mettendo ancora in evidenza i servizi tecnologici forniti e le relazioni tra questi, infine nella Figura 12.5c, tutti i componenti sono inseriti nel nodo, che verrà quindi considerato un unico elemento, a un livello di astrazione maggiore.

### 12.2.4 Business layer

Presentiamo ora gli elementi che caratterizzano i singoli livelli e le relazioni tra essi. La presentazione del Business Layer si basa sull'illustrazione di un tipico Business pattern (Figura 12.6).

Nella figura viene illustrato un Business service (si noti l'icona ellittica che caratterizza i servizi nella presentazione grafica degli elementi). A un business service viene associata la sua interfaccia, che corrisponde al canale attraverso cui il servizio viene fornito esternamente.

Il Business service è realizzato da un Business process (caratterizzato dall'icona freccia), che accede a un Business Object.

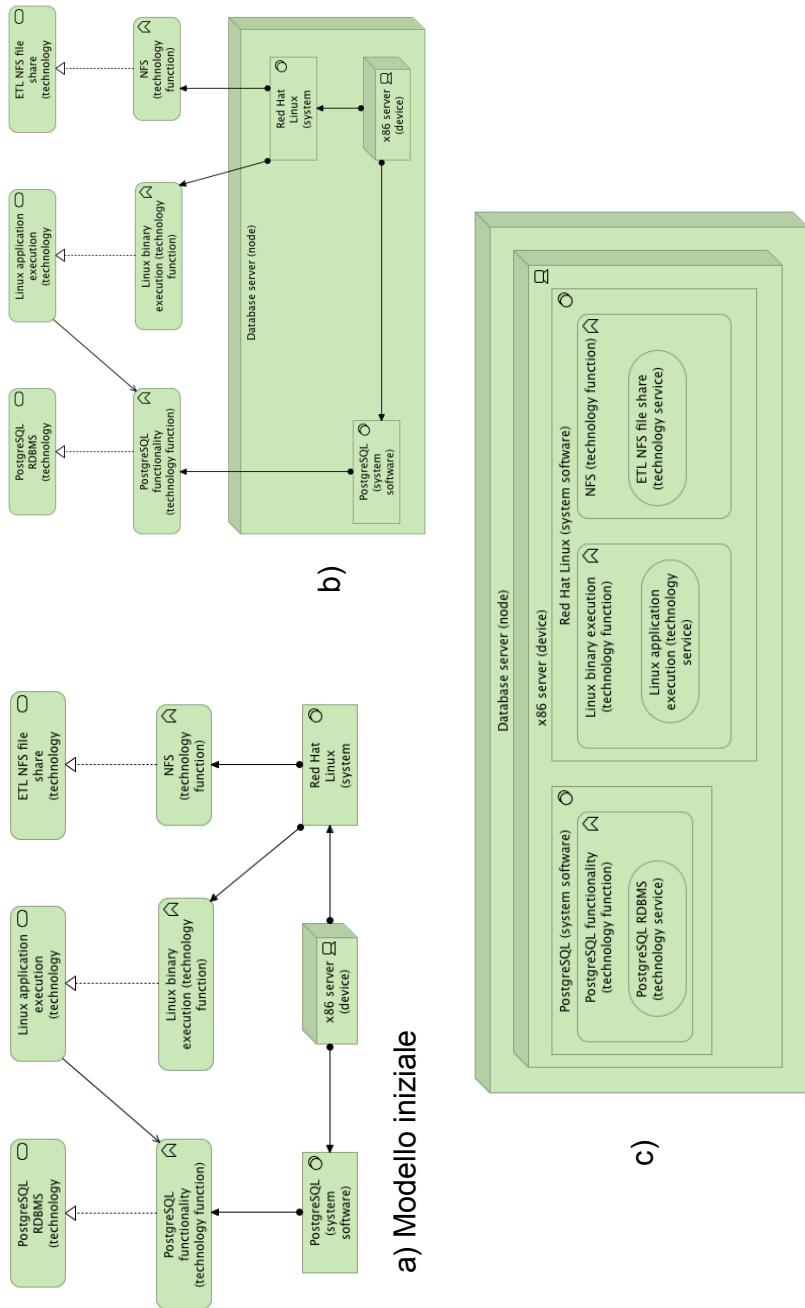


Figura 12.5: Esempi di nodi

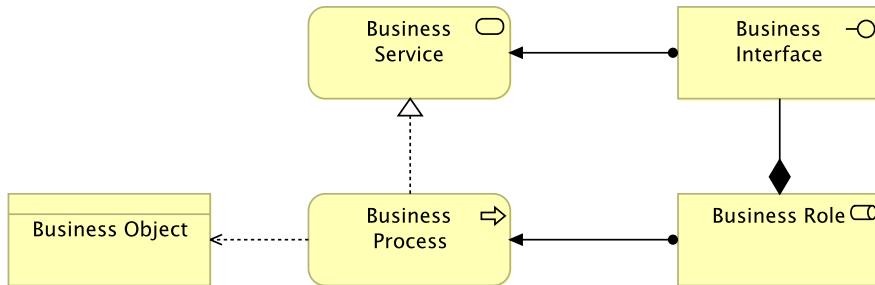


Figura 12.6: Business pattern

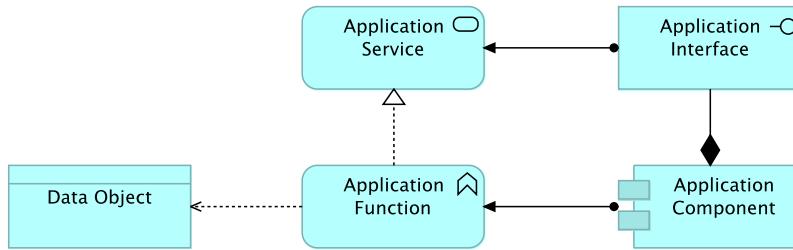


Figura 12.7: Application pattern

L'elemento attivo che esegue il processo è un **Business role** (caratterizzato da un'icona cilindro), assegnato al processo. L'interfaccia del **Business service** è un componente del **Business role**.

In Figura 12.12 viene illustrato come esempio a livello Business di ArchiMate un servizio di composizione di un documento (è un servizio generico, collegato a un **Business service** non rappresentato in figura). Nel business layer vediamo rappresentati un **Business Process** (*Write Answer*), con il **Business role** delegato alla sua esecuzione (oggetto attivo collegato con la relazione di Assegnamento), e cioè *Customer support*. Il **Business process** agisce su un **Business Object** (*Answer*), collegato ad esso con una relazione di Accesso.

### 12.2.5 Application layer

A livello applicativo, i componenti hanno relazioni simili a quelle illustrate sopra per il livello business. Come sopra, illustriamo una tipica configurazione a questo livello in Figura 12.7.

Nella figura viene illustrato un **Application service** con l'interfaccia ad esso associata.

L'**Application service** è realizzato da una **Application Function** (caratterizzato dall'icona freccia verso l'alto), che utilizza un **Data Object**.

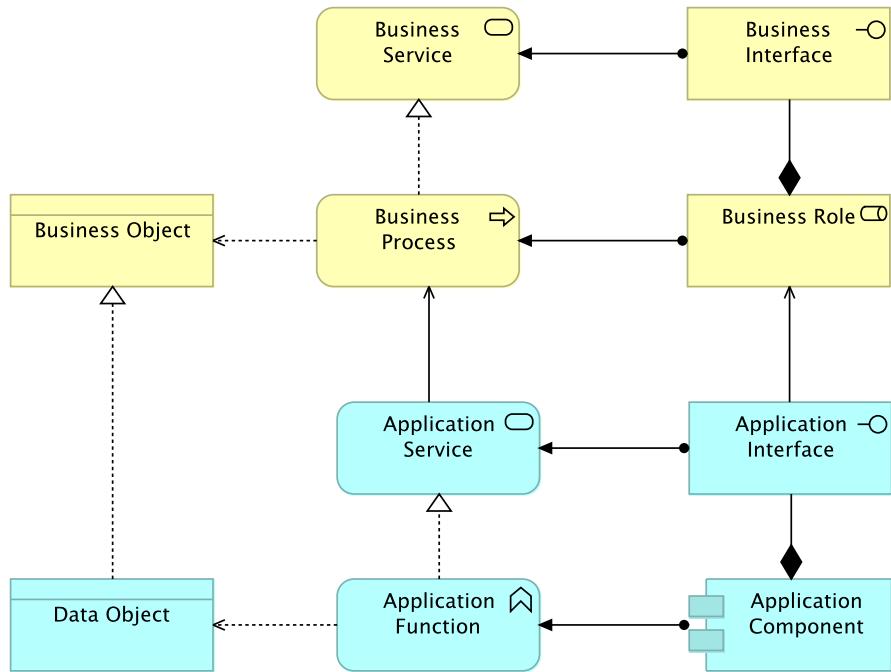


Figura 12.8: Collegamento di una applicazione tipica con una tipica funzionalità di business

L'elemento attivo che esegue l'Application Function è un Application component (caratterizzato dai due rettangoli a lato), assegnato alla funzione. L'interfaccia dell'Application service è un componente dell'Application Component.

In Figura 12.12, l'Application service è document creation, realizzato dalla funzione Editing di MS word (Application Function). Il componente applicativo assegnato alla funzione di editing è word processing [MS word], con la sua GUI (Application interface), che ne è una parte. il Data Object è l'oggetto passivo della funzione di editing, e cioè Word Document.

I due pattern illustrati vengono collegati nel modello quando un'applicazione viene utilizzata da un business pattern. Il collegamento viene rappresentato con una relazione *serves a* (serving), come illustrato nella Figura 12.8.

Riprendendo l'esempio di Figura 12.12, Document creation (Application Service) serve il Business Process Write Answer, e la GUI serve Customer support.

La relazione *serves a* può essere utilizzata anche per collegare applicazioni tra loro. Come visto nel capitolo dedicato alle architetture funzionali di BOAT, questo tipo di collegamenti si verifica tipicamente nel collegamento tra sistemi di back-end e nei collegamenti tra sistemi di back-end e sistemi di front-end. Nella Figura 12.9 viene mostrata la realizzazione di questo collegamento. Nella figura

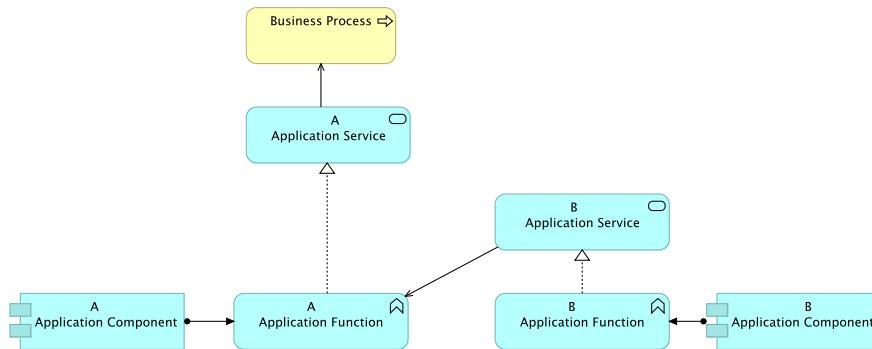


Figura 12.9: Collegamento tra due applicazioni

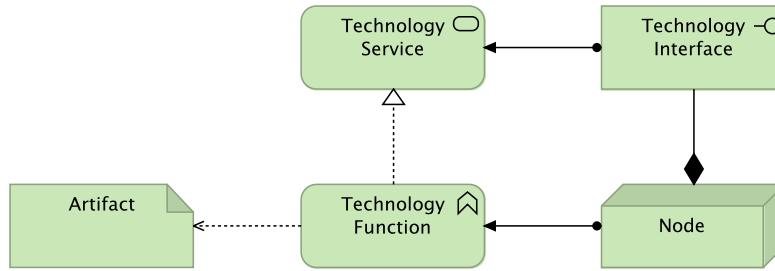


Figura 12.10: Infrastruttura di base

due funzioni applicative sono realizzate da due componenti applicativi, ciascuno dei quali è collegato alla funzione applicativa corrispondente. La *funzione applicativa B* viene utilizzata come servizio dalla *funzione applicativa A* come segue: la funzione applicativa B espone il servizio *application service B*, tramite la relazione *realizza*. La funzione applicativa A utilizza il servizio esposto dalla funzione applicativa B tramite la relazione *serve a* che collega l'*application service B* alla funzione applicativa A.

A sua volta la *funzione applicativa A* espone il servizio *application service A* tramite la relazione *realizza*, che viene utilizzato (relazione *serves a*) dal business process.

### 12.2.6 Livello tecnologico

Viene ora descritto il livello tecnologico. La Figura 12.10 mostra una configurazione tipica per l'infrastruttura tecnologica.

Come nei casi precedenti, abbiamo un servizio tecnologico con l'interfaccia ad esso associata e una funzione tecnologica che lo realizza.

L'oggetto in questo caso ha una rappresentazione grafica per indicare che è un artefatto (file di testo, eseguibile o altro). Un nodo, con la sua interfaccia tecnologica, è assegnato alla funzione tecnologica, nel senso che la potrà eseguire.

La Figura 12.11 mostra il collegamento tra livello applicativo e infrastruttura, illustrando come l'infrastruttura tecnologica fornisce i servizi necessari all'applicazione.

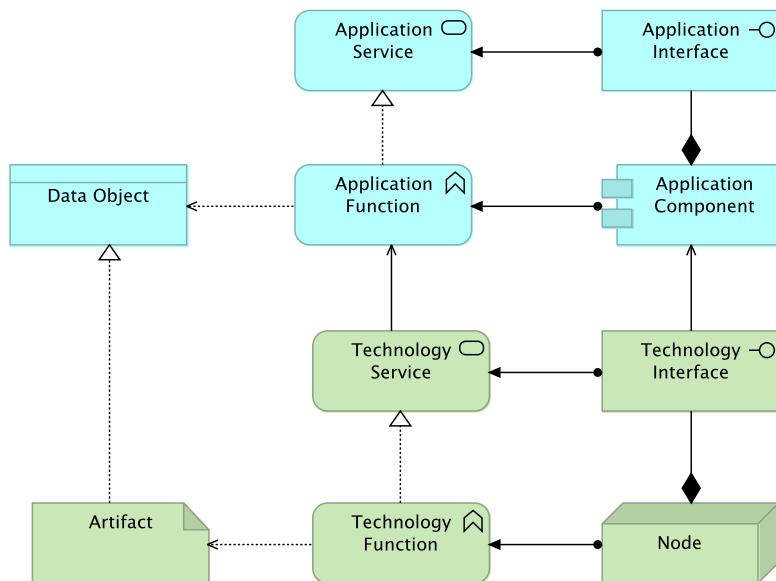


Figura 12.11: Collegamento tra livello applicativo e infrastruttura

## 12.3 Esempi

### 12.3.1 Esempio di rappresentazione di un sistema di composizione di un documento

Il modello in Figura 12.12 mostra la scrittura di una risposta a un cliente da parte di un ruolo di supporto ai clienti (Customer Support). Per funzionare, sono necessari due servizi infrastrutturali per eseguire l'applicazione e archiviare il documento. In questo esempio, tutto avviene su un PC standalone.

Abbiamo già discusso nelle sezioni precedenti i layer Business e Application di ArchiMate e le loro relazioni. Qui li vediamo collegati anche con l'infrastruttura tecnologica necessaria (per i dettagli vedere la Sezione 12.4). A livello tecnologico il documento Word viene realizzato dal file foo.doc, la funzione di editing utilizza il servizio tecnologico Windows application execution, realizzato dalla funzione tecnologica Windows binary execution, eseguita sul dispositivo

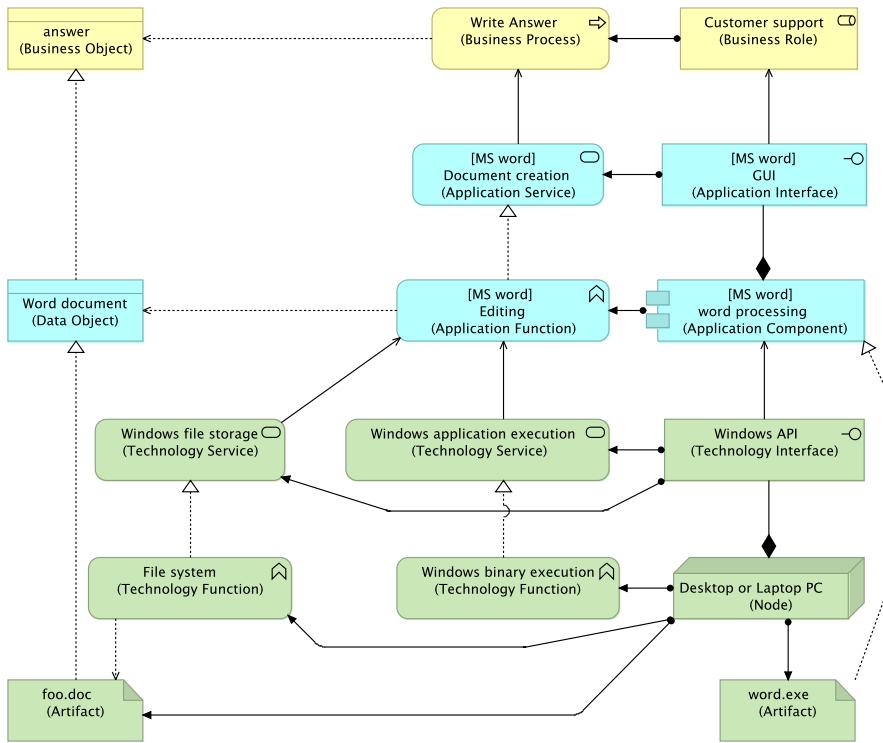


Figura 12.12: Esempio di servizio business di composizione di un documento [14]

etichettato Desktop or Laptop PC, a cui viene assegnato oltre alla funzione di esecuzione binaria anche l'eseguibile word.exe

### 12.3.2 Esempio di servizio assicurativo

Nella Figura 12.13 viene illustrata l'architettura per fornire il servizio di gestione di un rimborso ai clienti da parte di un'assicurazione.

Da notare la rappresentazione dei clienti e dell'assicurazione con l'icona persona, associata ai corrispondenti ruoli (la relazione associazione è qui rappresentata in modo simmetrico, secondo ArchiMate 2.0).

In questa figura è utile esaminare anche come i componenti applicativi (CRM system, Policy administration, Financial application) forniscono i servizi necessari per realizzare il processo di business.

L'infrastruttura tecnologica è fornita da un mainframe e da un sistema blade.

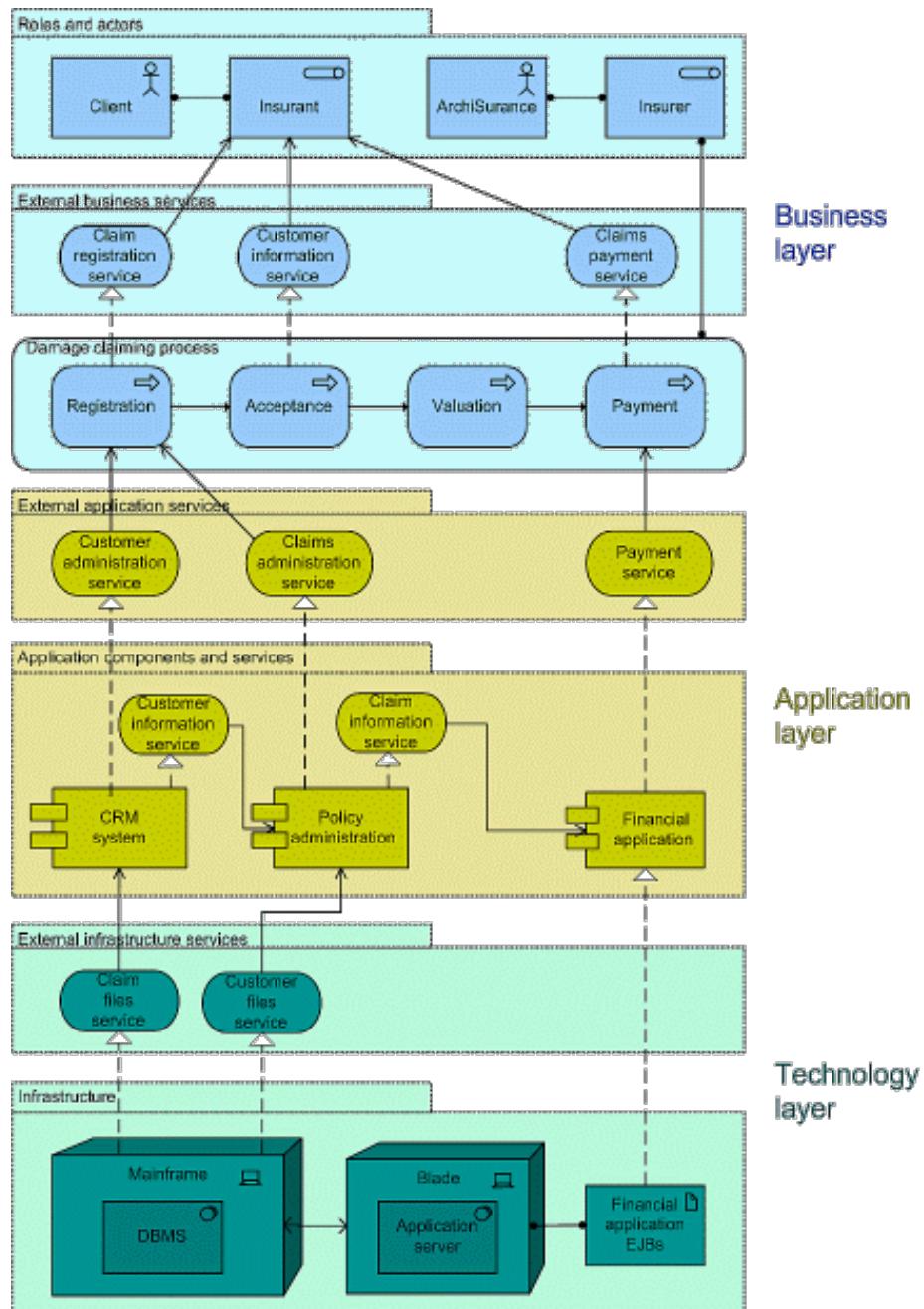


Figura 12.13: Esempio di servizio business di composizione di un documento  
(fonte: IBM Developers word)

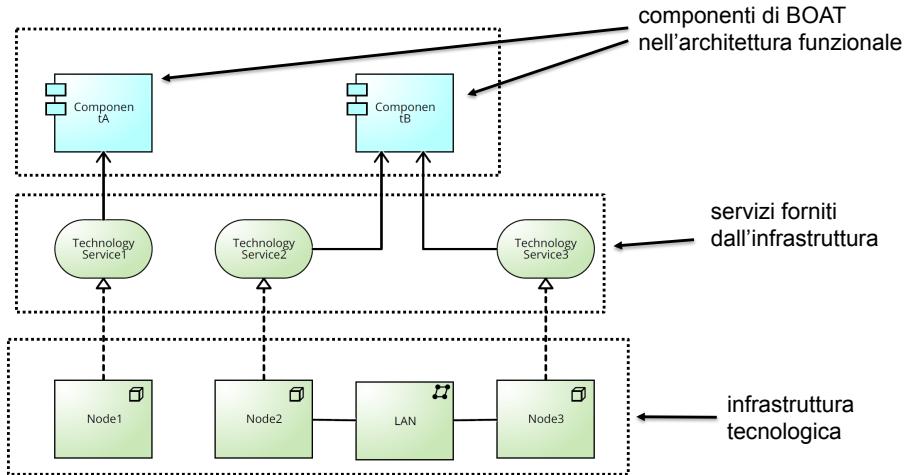


Figura 12.14: Relazione tra componenti applicativi e infrastruttura tecnologica

## 12.4 Rappresentazione dell'infrastruttura tecnologica

In questo capitolo l'obiettivo è quello di fornire uno strumento per rappresentare l'infrastruttura tecnologica. Di conseguenza verranno illustrati più in dettaglio i componenti infrastrutturali.

Nel seguito, rappresenteremo il collegamento tra i componenti applicativi tipici dell'architettura funzionale di BOAT e i componenti infrastrutturali come illustrato nella Figura 12.14.

Iniziamo discutendo le corrispondenze tra elementi nei diagrammi Party Level di BOAT e i diagrammi di ArchiMate. I componenti applicativi rappresentati in ArchiMate corrispondono agli *elementi di back end che compongono l'architettura Party level di BOAT*. Come già discusso, è possibile che vi siano collegamenti tra applicazioni (come illustrato in Figura 12.9), che corrispondono ai collegamenti tra BES in BOAT. Come già discusso nel capitolo sull'integrazione delle applicazioni, questi collegamenti possono essere diretti oppure utilizzare servizi infrastrutturali che li supportano. Nel seguito, per semplicità, non si discuteranno i dettagli di servizi infrastrutturali di questo tipo.

Per quanto riguarda gli *elementi di front end che compongono l'architettura Party-level di BOAT*, questi vengono rappresentati in ArchiMate come Application Services, a cui (se rappresentate) vengono assegnate le Application Interfaces, che sono componenti di un Application Component.

Nel seguito per semplicità ci si concentrerà sulla rappresentazione dell'infrastruttura tecnologica, per cui le interfacce non verranno indicate nei diagrammi seguenti.

Nella rappresentazione dell'architettura tecnologica in ArchiMate si vuole

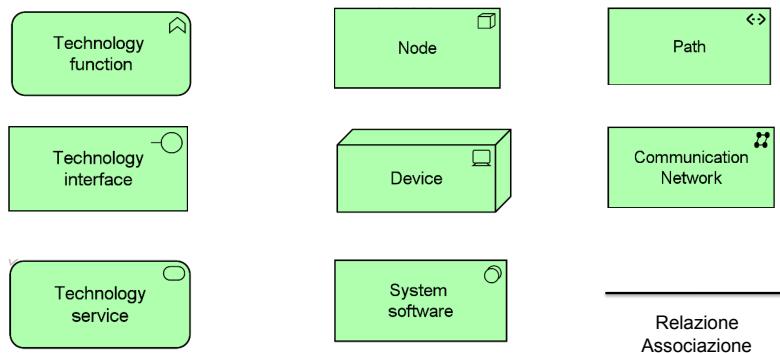


Figura 12.15: Componenti per modellare l'infrastruttura tecnologica

rappresentare quali servizi infrastrutturali saranno utilizzati dai componenti applicativi e la relativa realizzazione.

Si ricorda che i diagrammi Party-level consentono di rappresentare alcune scelte realizzative che hanno anche implicazioni infrastrutturali, da cui la parziale sovrapposizione della parte A di BOAT con il Layer Tecnologico di ArchiMate, mostrata in Figura 12.2. Nel seguito si esamineranno in particolare la rappresentazione in ArchiMate di servizi e infrastrutture gestiti da operatori esterni (Sezione 12.4.3) e la virtualizzazione (Sezione 12.5.2, già discussi nella parte A di BOAT).

Si introducono nella prossima sezione i componenti forniti da ArchiMate per la rappresentazione dell'infrastruttura tecnologica (Sezione 12.4.1) e in seguito si discute la rappresentazione delle scelte architetturali in ArchiMate (Sezione 12.5).

### 12.4.1 Componenti per l'infrastruttura tecnologica

I principali componenti utilizzati per rappresentare i dettagli dell'infrastruttura tecnologica sono i seguenti (Figura 12.15, si noti l'icona che caratterizza ciascun tipo di componente):

- *Node*: Una risorsa computazionale o fisica che ospita, manipola o interagisce con altre risorse computazionali o fisiche.
- *Technology Function*: Un insieme di comportamenti tecnologici che possono essere eseguiti da un nodo.
- *Technology Service*: Un comportamento tecnologico esposto esplicitamente come servizio.
- *Technology Interface*: Un punto di accesso da cui è possibile accedere ai servizi tecnologici offerti da un nodo.

- *Device*: Una risorsa IT fisica sulla quale i componenti software e gli artefatti (ad esempio, i file dei componenti software eseguibili) possono essere archiviati o distribuiti per l'esecuzione.
- *Software*: Software che fornisce o contribuisce a un ambiente di archiviazione, esecuzione e utilizzo di software o dati distribuiti al suo interno.
- *Path*: Un collegamento tra due o più nodi, attraverso il quale questi nodi possono scambiare dati o materiale.
- *Network*: Un insieme di strutture e funzionalità che collegano sistemi informatici o altri dispositivi elettronici per la trasmissione, l'instradamento e la ricezione di dati o comunicazioni basate su dati come voce e video.
- *Relazione Associazione*: si introduce qui una nuova relazione, per rappresentare il collegamento tra nodi e reti.

### 12.4.2 Relazioni derivate

Le relazioni strutturali e di dipendenza tra i componenti sono ordinate dalle più deboli alle più forti secondo quanto illustrato nella Figura 12.16. Nella figura non viene indicata la relazione Associazione, perché la regola vale per le relazioni strutturali e di dipendenza (association fa parte della categoria “other”) e quindi non viene utilizzata nelle relazioni derivate.

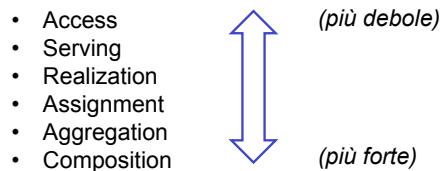


Figura 12.16: Ordinamento tra le relazioni strutturali e di dipendenza

**Regola (relazioni derivate)**: due relazioni strutturali o di dipendenza che si uniscono a un elemento intermedio possono essere combinate e sostituite dalla più debole delle due.

L'applicazione transitiva di questa proprietà consente di sostituire una catena di relazioni con la più debole della catena. Le relazioni derivate risultanti sono relazioni valide in ArchiMate. Un esempio di trasformazione tra modelli è mostrato nella Figura 12.17.

### 12.4.3 Gruppi

E' possibile raggruppare più elementi come mostrato nella Figura 12.18.

L'utilizzo del raggruppamento nel livello infrastrutturale serve soprattutto a indicare chi gestisce i vari elementi infrastrutturali che vengono rappresentati. Ad esempio, nella Figura 12.18 viene messo in evidenza che l'infrastruttura è

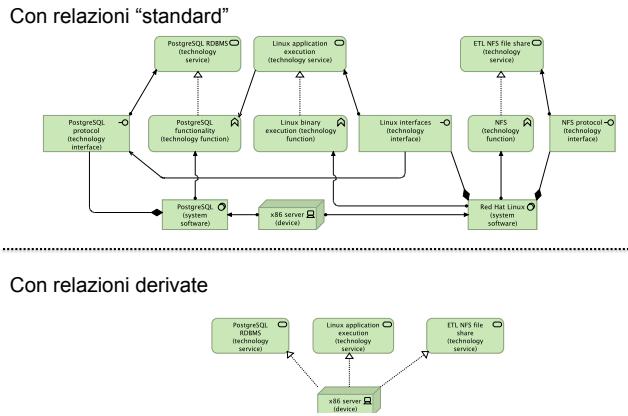


Figura 12.17: Esempio di relazioni derivate

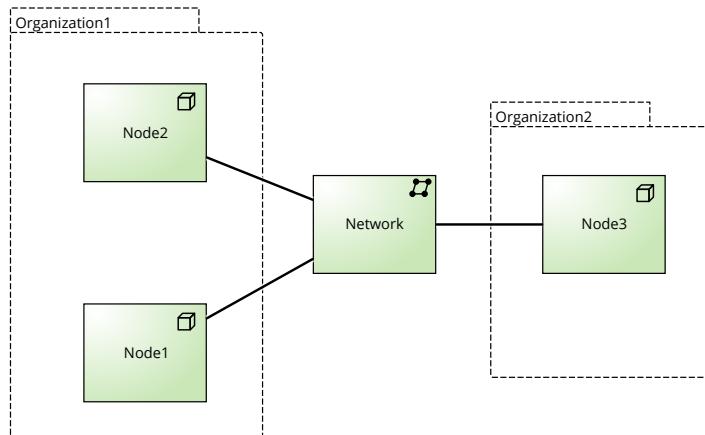


Figura 12.18: Esempi di gruppi

fornita da due organizzazioni diverse e che gli elementi sono collegati tra di loro tramite una rete.

## 12.5 Rappresentare le scelte architetturali in ArchiMate

In questa sezione illustriamo come rappresentare le scelte architetturali in ArchiMate, a partire da un diagramma Party level di BOAT. Abbiamo già discusso nella sezione precedente come, a partire da un diagramma Party level di BOAT, si rappresentano in ArchiMate i componenti BOAT dell'architettura funzionale come componenti applicativi e servizi forniti dall'infrastruttura. In questa sezione l'attenzione viene posta sulla rappresentazione degli aspetti specifici dell'architettura tecnologica e delle possibili alternative di realizzazione a livello tecnologico. Si discutono in particolare la rappresentazione di architetture a N-tier nella Sezione 12.5.1 e la rappresentazione di componenti infrastrutturali virtualizzati nella Sezione 12.5.2.

### 12.5.1 Architetture N-tier

Come visto nel Capitolo 11, i livelli logici (layer) possono essere associati in modi diversi ai livelli fisici. In questa sezione presenteremo la rappresentazione tramite ArchiMate di architetture a 2-tier e 3-tier, in diversi configurazioni, utilizzando i costrutti illustrati nelle sezioni precedenti per modellare l'infrastruttura tecnologica.

#### 2-tier, thick client

Prendiamo in considerazione un'architettura 2-tier, in cui il client è un thick client, contiene cioè l'applicazione, e la gestione dei dati viene effettuata sul server.

Come illustrato nella Figura 12.19, il componente applicativo è realizzato dal (software sul) client. La funzione applicativa è servita sia dall'esecuzione dell'applicazione nel client sia dal back end (gestione dati) sul server, per la gestione dei dati. L'artefatto Application distribution indica il software che viene distribuito sui client per l'esecuzione sia della logica di business che della presentazione dell'applicazione. I nodi client e server comunicano attraverso la rete.

La stessa architettura può essere rappresentata in modo più astratto come mostrato nella Figura 12.20, dove non si mettono in evidenza i componenti del componente applicativo.

#### 2-tier, thin client

Esaminiamo ora il caso di una architettura 2-tier con thin client.

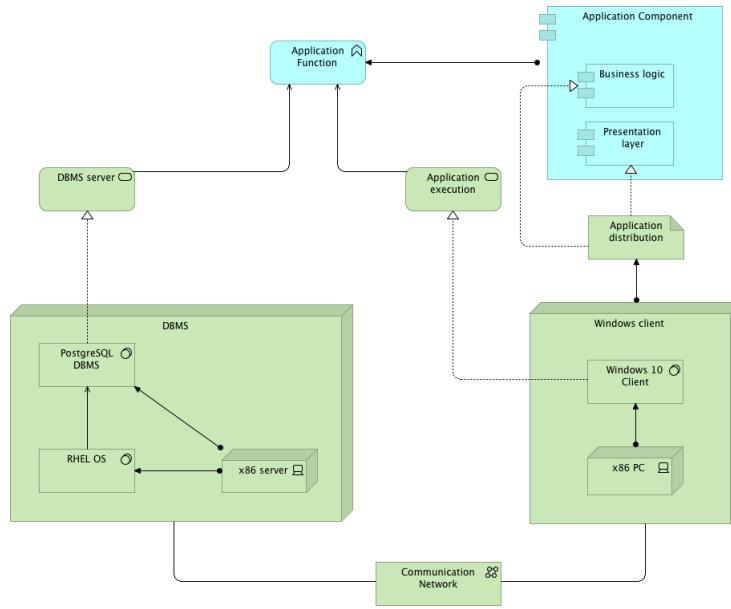


Figura 12.19: Thick client in architettura 2-tier

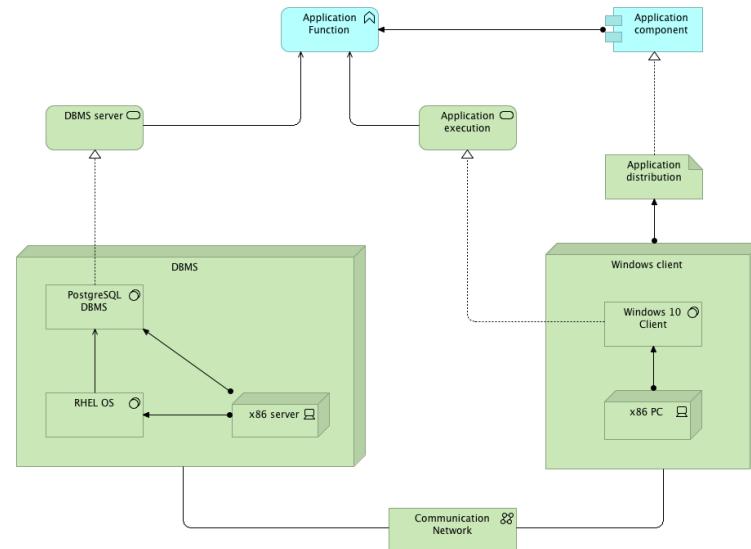


Figura 12.20: Thick client in architettura 2-tier, con componenti applicativi composti

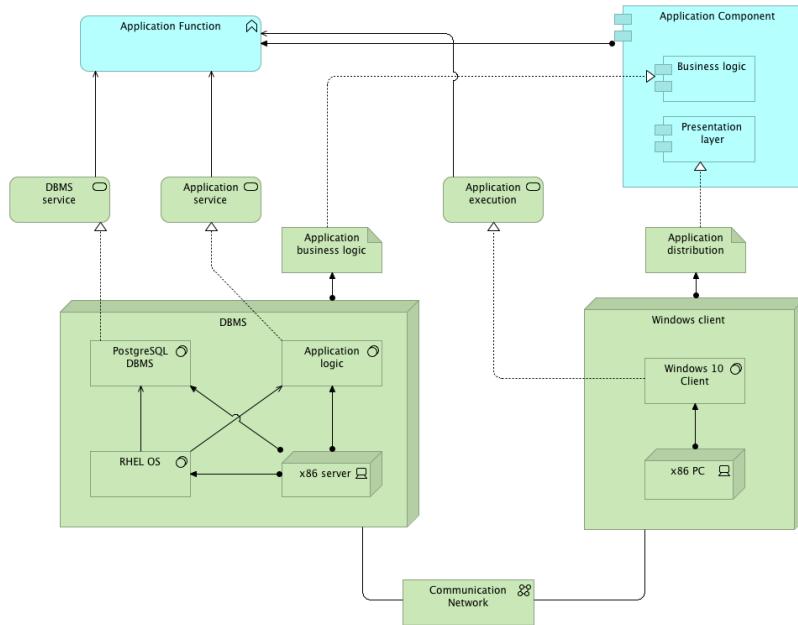


Figura 12.21: Thin client in architettura 2-tier

In questo caso, illustrato in Figura 12.21, il componente applicativo è realizzato dal (software sul) client, per il livello di presentazione e anche dal (software sul) server per la logica di business. La funzione applicativa viene quindi fornita sia dall'esecuzione dell'applicazione nel client (presentazione) sia dal backend (servizi di gestione dati e logica) sul server. I nodi client e server comunicano attraverso la rete.

### Architetture 3-tier

Prendiamo ora in esame un'architettura a tre livelli, nella configurazione 7 illustrata nella Sezione 10.3 (Figura 10.5), in cui ciascun livello logico è associato a un livello fisico.

In ArchiMate, questa configurazione viene modellata come mostrato nella Figura 12.22.

Il componente applicativo è realizzato dal (software sul) client, per il livello di presentazione, e anche dal (software sul) server applicativo per la logica di business. La funzione applicativa viene fornita sia dall'esecuzione dell'applicazione nel client sia dai servizi forniti dal server delle applicazioni e dal server di gestione dei dati.

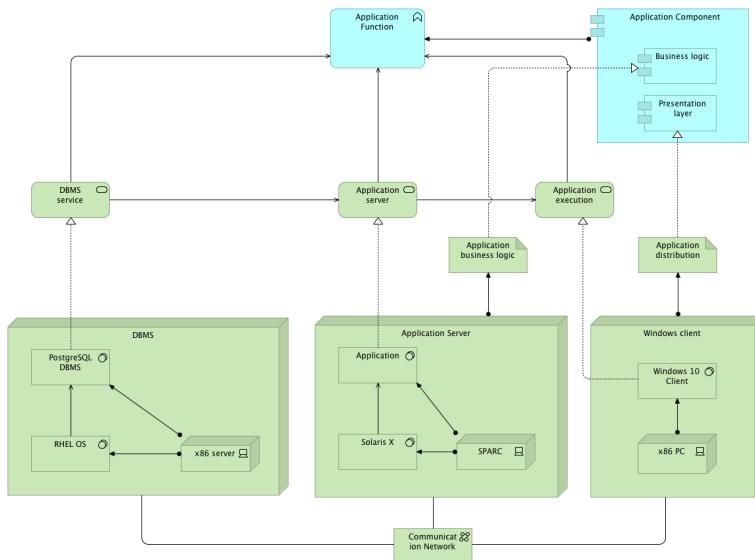


Figura 12.22: Architettura 3-tier, configurazione 7

### 12.5.2 Virtualizzazione

Concludiamo il capitolo illustrando alcune architetture con elementi virtualizzati.

Innanzitutto riprendiamo le due possibili tipologie di virtualizzazione, bare-metal e hosted.

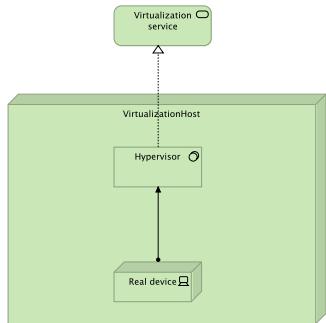
Come mostrato nella Figura 12.23, la differenza tra le due architetture è nella presenza del sistema operativo, mostrato come componente software, nel caso dell'architettura hosted.

Confrontiamo ora un'architettura tradizionale, con due server non virtualizzati, mostrata in Figura 12.24 e un'architettura con server virtualizzati come in Figura 12.25.

Nella Figura 12.24, con due server non virtualizzati, sono rappresentati due nodi, uno per il server che contiene il DBMS, e uno per il Web server. Ciascuno dei due nodi contiene un dispositivo fisico (nodo X86 server), assegnato all'esecuzione di due software. In entrambi i casi un componente software è il sistema operativo (rispettivamente RHEL OS e Ubuntu server). Il nodo con il DBMS contiene anche il componente software PostgreSQL DBMS e il nodo con il Web server contiene il componente software Apache web server.

Nella Figura 12.25 vengono invece rappresentati i due server virtualizzati, eseguiti su uno stesso dispositivo fisico (indicato come Real device in figura). Abbiamo in questo caso tre nodi: uno per l'host di virtualizzazione (VirtualizationHost) e due per i server virtualizzati. E' da notare che i server virtuali sono software di sistema e non device. I server virtuali utilizzano il servizio di vir-

- Bare metal



- Hosted

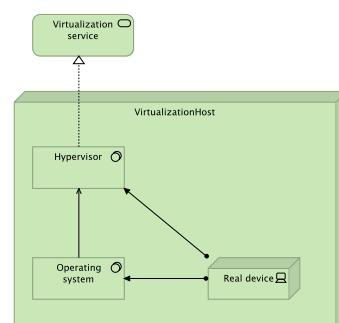


Figura 12.23: Architetture di virtualizzazione

tualizzazione fornito dall'hypervisor. I nodi virtuali sono componenti (software) del nodo fisico.

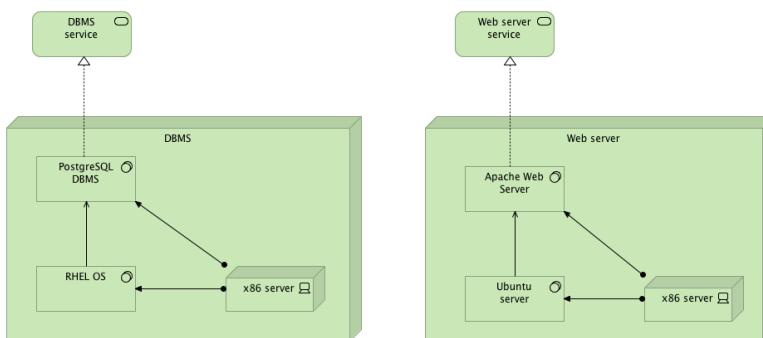


Figura 12.24: Due server, non virtualizzati

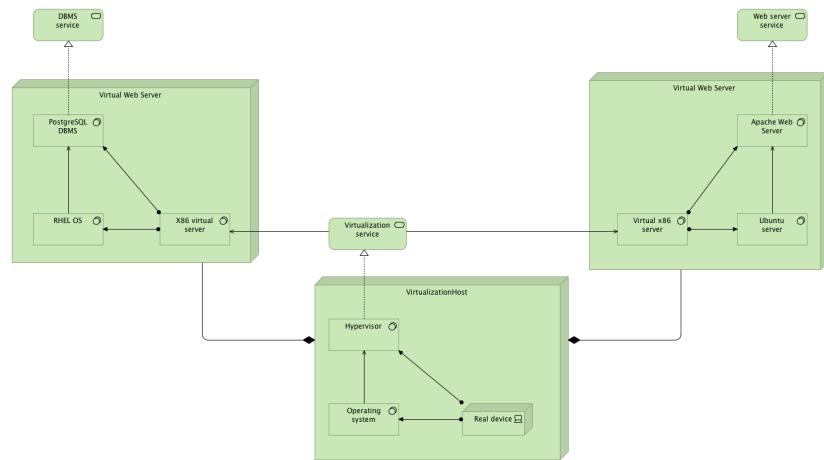


Figura 12.25: Due server virtualizzati

# **Parte IV**

# **Scenari**



## Capitolo 13

# Sistemi informativi e nuove tecnologie

### 13.1 Innovazioni tecnologiche

Come discusso nel capitolo introduttivo, i sistemi informativi sono sistemi complessi tramite i quali un'organizzazione può raggiungere i suoi obiettivi e migliorare i suoi prodotti o servizi rendendoli competitivi sul mercato. Un sistema informativo non è però isolato da quello che succede all'esterno dell'organizzazione e il modo in cui è progettato e le sue potenzialità sono influenzate dalle tecnologie impiegate per realizzarlo. L'emergere di nuove tecnologie e la loro adozione all'interno del sistema informativo di un'organizzazione può portare molti vantaggi e influenzare anche gli stessi obiettivi aziendali.

Negli ultimi anni stiamo assistendo all'affermazione di diverse tecnologie emergenti nel settore IT che stanno influenzando in modo significativo il futuro dei sistemi informativi. Queste tecnologie, conosciute con il nome di Big Five sono:

- Big data
- Internet of Things
- Cloud computing
- Social media
- Sistemi mobili

**Big Data.** I Big Data sono data set di dimensioni estremamente elevate per cui l'immagazzinamento e l'analisi dei dati in essi contenuti richiede l'uso di sistemi distribuiti. L'analisi dei dati, spesso risultato del monitoraggio di file di log di un sistema, di interazioni con i social media, possono essere analizzati per rivelare pattern, tendenze e associazioni che possono essere di grande valore per

migliorare il funzionamento del sistema (es. pattern di interazione degli utenti con un sistema informatico).

**Internet of Things (IoT).** Consiste nella interconnessione via internet di un insieme di dispositivi utilizzati in modo quotidiano (smartphone, sensori, smart-watch) che hanno la capacità di inviare e ricevere dati. I dispositivi, mediante una connessione internet, possono essere monitorati e collegati in remoto, scambiandosi dati tramite appositi protocolli. Esempi di dispositivi sono impianti per il monitoraggio cardiaco, microchip per il tracciamento degli animali in una fattoria, videocamere, automobili dotate di sensori. L'utilizzo di questa tecnologia è abilitante per la realizzazione di sistemi automatizzati con un intervento diretto da parte degli operatori sempre più limitato.

**Cloud Computing.** Il cloud computing è un paradigma che prevede l'uso di server remoti per immagazzinare, gestire ed elaborare le informazioni. Tramite questo modello le risorse possono essere condivise tra diversi utenti tramite internet, permettendo l'accesso da qualsiasi locazione. Grazie alla condivisione delle risorse, è possibile dare agli utenti che le utilizzano l'impressione che tali risorse siano infinite. Il cloud computing riduce anche i costi della computazione dato che le risorse IT sono condivise tra diversi utenti che pagano sulla base dell'utilizzo effettivo. L'adozione del cloud computing permette alle organizzazioni di eseguire le loro applicazioni in modo più veloce con costi di gestione e manutenzione ridotti. La quantità di risorse computazionali utilizzate dall'applicazione può cambiare in modo dinamico e trasparente per l'utente in modo da rispondere ad esigenze impreviste.

**Social Media.** I social media sono tecnologie volte alla facilitazione della creazione e condivisione di informazioni, idee e interessi con una comunità virtuale. L'utilizzo dei social media può aiutare le organizzazioni ad effettuare strategie di marketing focalizzate sulle tipologie di clienti. Grazie ai dati raccolti è infatti possibile effettuare una profilazione dei comportamenti di un utente o di una categoria di utenti e sfruttarli per strategie di marketing più efficaci.

**Sistemi Mobili.** I sistemi mobili (o mobile computing) sono una tecnologia che prevede l'interazione tra l'utente e un sistema computazionale che può essere trasportato durante il suo utilizzo permettendo la trasmissione di dati, contenuti vocali e video. Le caratteristiche di questi sistemi sono quindi la portabilità (sono sistemi che possono essere trasportati facilmente da un punto ad un altro in base alle esigenze), la connettività (i sistemi mobili sono continuamente connessi alla rete internet per permettere la collaborazione con altri utenti) e l'interattività (i diversi sistemi mobili sono connessi tra di loro per permettere la comunicazione e la collaborazione tra di essi tramite la trasmissione di dati). Esempi comuni di sistemi mobili sono i laptop, gli smartphone, le smart card e i dispositivi indossabili, come orologi, braccialetti e collane dotati di capacità computazionale e connettività.

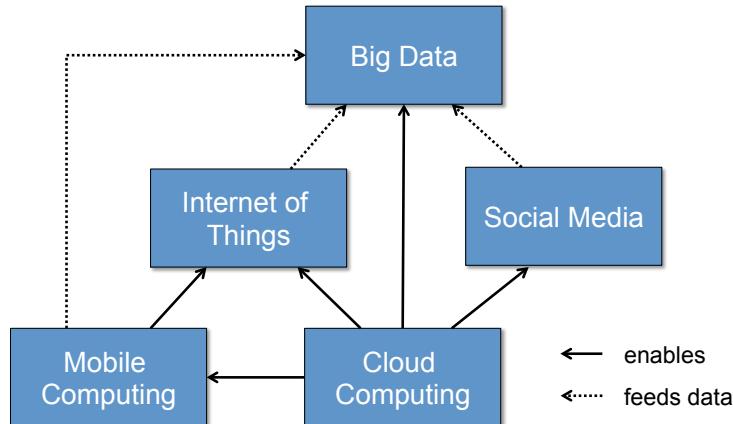


Figura 13.1: "Big five"

Come illustrato in Figura 13.1 queste tecnologie sono relazionate tra di loro per cui spesso l'adozione di una richiede o rende opportuna l'adozione dell'altra.

Nel capitolo vengono illustrati alcuni scenari applicativi e come le seguenti tecnologie emergenti nel settore hanno un impatto sullo sviluppo dei sistemi informativi

## 13.2 Industria 4.0

Industria 4.0 è vista come la quarta rivoluzione industriale. La prima ha migliorato l'efficienza grazie alla forza motrice del vapore. La seconda è caratterizzata dall'uso dell'elettricità e dalla produzione di massa (linee di assemblaggio), mentre la terza ha accelerato l'automazione utilizzando l'elettronica e l'IT. La quarta, in corso, sembra seguire un ritmo maggiormente evolutivo, piuttosto che dirompente. Si concentra sul rendere gli oggetti fisici integrati con continuità a una rete informativa. Internet riesce infatti a fondere gli aspetti di macchine intelligenti (smart), i sistemi di produzione e i processi creando una rete sofisticata di conoscenza aziendale. Il mondo reale si sta tramutando in un certo senso in un grande sistema informativo automatizzato.

Il termine Industria 4.0 va differenziato rispetto a concetti più circoscritti, quali "Internet of Things", "maker movement" o "Factory 4.0". Industria 4.0 pone l'enfasi sulla digitalizzazione congruente di vari settori industriali e sul collegamento fra tutte le unità produttive che partecipano a un'unità economica.

Considerando le caratteristiche di un ambiente industriale, si possono evidenziare gli aspetti descritti nel seguente.

**Sistemi cyber-fisici e mercato.** I sistemi IT oggi sono già il cuore dei sistemi produttivi. In Industria 4.0, i sistemi produttivi saranno ancora più connessi a tutti i sotto-sistemi, ai processi, agli oggetti interni ed esterni e alle reti di

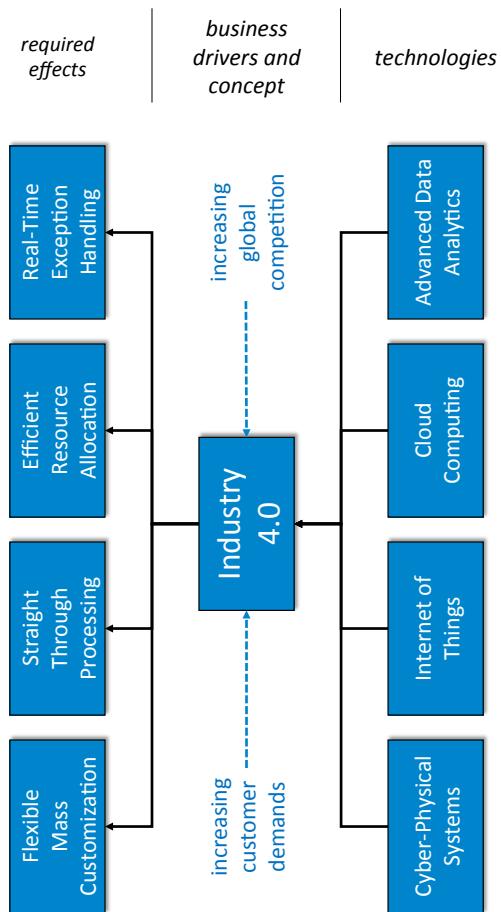


Figura 13.2: Industria 4.0 - Overview

fornitori e clienti. La complessità sarà molto più elevata e richiederà offerte di mercato più articolate. I sistemi IT saranno la base per realizzare sistemi informativi che comprendono macchinari, sistemi di memorizzazione e immagazzinamento di oggetti e informazioni, e materiali che si conformano a certi standard predefiniti e che nel complesso sono interrelati in quello che si chiama un sistema cyber-fisico (cyber-physical system - CPS). I sistemi di produzione vengono così controllati in tempo reale (real-time). Gli impianti e i sistemi informatici/informativi avranno interfacce chiaramente definite e simili. Usando le tecnologie, sarà possibile sostituire in modo flessibile le macchine lungo la catena del valore. Ciò rende possibile una produzione altamente efficiente in cui i processi produttivi possono essere cambiati con breve preavviso e in cui i tempi di inattività abbassati o azzerati.

**Robot e macchine smart.** In alcuni settori i robot già hanno sostituito il lavoro umano. I robot multi-purpose sviluppati dai vari attori di Industria 4.0 nel manifatturiero europeo sono raddoppiati dal 2004. A breve, i robot diventeranno intelligenti (smart), ovvero saranno capaci di adattarsi, comunicare e interagire fra loro e con l'ambiente. Ciò permetterà un ulteriore incremento di produttività portando a un grande cambiamento nella struttura dei costi, negli skill e nella configurazione dei siti di produzione. Gli smart robot non solo sostituiranno gli umani in workflow semplici, ma lavoreranno con gli umani interagendo sui vari task e usando interfacce smart basate su sensori. L'uso dei robot si sta ampliando a varie funzioni: produzione, logistica, gestione del lavoro di ufficio. Il controllo sulle attività può essere remoto. Se sorge un problema, la persona responsabile viene avvisata sul telefono mobile con collegamento a una web cam, per visionare il problema e vedere/dare istruzioni anche senza essere fisicamente presente.

**Big Data.** I dati sono ormai considerati il materiale grezzo del XXI secolo. In effetti, la quantità di dati a disposizione del business raddoppia ogni 1.2 anni. Un impianto del futuro produrrà una mole notevole di dati che vanno salvati, processati e analizzati, con un cambiamento radicale nell'approccio alla loro gestione. Un aspetto cruciale riguarda la cyber security relativa alla segretezza e integrità dei dati che è sentita come un aspetto che può impattare significativamente sulla competitività. Sono allo studio metodi innovativi per gestire i big data, per esempio tramite nuove generazioni di Cloud computing.

**Nuova qualità della connettività.** In Industria 4.0 deve esserci collegamento fra gli oggetti fisici e quelli digitali. Macchine, artefatti di lavoro, sistemi e persone scambieranno continuamente informazioni via Internet; quindi gli oggetti fisici saranno collegati alla loro immagine digitale. La produzione con macchine interconnesse diventa facilitata: una macchina viene immediatamente informata quando una parte è prodotta da un'altra macchina, e lo stesso succede per i trasportatori o i robot della logistica. Le macchine si adattano automaticamente alla fase produttiva e perfino i prodotti possono comunicare

fra loro quando sono pronti – tramite una rete Internet of Things – e chiedere a un mezzo di trasporto di essere prelevati, o inviare una e-mail al dipartimento ordini dicendo “Sono pronto a essere consegnato”. Gli impianti e gli uffici sono collegati così da organizzare lo scheduling della produzione ottimizzando la capacità degli impianti stessi.

**Efficienza energetica e decentralizzazione.** I cambiamenti climatici e la scarsità di risorse influenzano le tendenze di tutti i player di Industria 4.0. Le tendenze saranno verso la decentralizzazione degli impianti, creando il bisogno di tecnologie “carbon-neutral” nella produzione. Usare energie rinnovabili sta diventando più attraente finanziariamente per le aziende e in futuro vi saranno sempre più impianti che generano la potenza che loro serve, con implicazioni per i fornitori di infrastrutture. Si stanno anche studiando impianti nucleari decentralizzati di piccolo taglio per rifornire i grandi impianti ad alto consumo elettrico e fornire un risparmio energetico a doppia cifra.

**Industrializzazione virtuale.** Creare un nuovo impianto o un nuovo prodotto in un impianto esistente è molto difficile perché richiede ore di adattamento, prove, test pre-serie, un gruppo di persone di alto livello per il lancio e vari costi addizionali inattesi. Un giorno perso è una grande perdita finanziaria. Industria 4.0 si propone di usare impianti e prodotti virtuali che preparino la produzione fisica. Ogni processo prima viene simulato e verificato virtualmente; quando la soluzione finale è pronta viene realizzato il mapping fisico – ovvero tutti i software, i parametri, e le matrici numeriche vengono caricati sulle macchine fisiche che controllano la produzione. Si stima che bastino tre giornate per realizzare un componente automobilistico anziché i tre mesi necessari attualmente. Gli impianti virtuali possono essere progettati e visualizzati in 3D, mostrando anche come i lavoratori e le macchine interagiranno.

In Figura 13.2 appaiono i fattori chiave sopra illustrati, riportati a una struttura funzionale dell’azienda ispirata al framework BOAT. La colonna di sinistra mostra gli effetti che si vogliono ottenere tramite il concetto di Industria 4.0 concept (non-esaustivo). La colonna intermedia mostra i due driver principali legati al concetto di Industria 4.0, mentre la colonna di destra mostra le principali classi tecnologiche che lo abilitano.

La Figura 13.3 illustra un esempio di architettura per il sistema informativo in un caso industriale. Nell’architettura compaiono diversi moduli software che contribuiscono alla realizzazione del sistema: ERP (Enterprise Resource Planning), MES (Manufacturing Execution System) e PLMS (Product Lifecycle Management System).

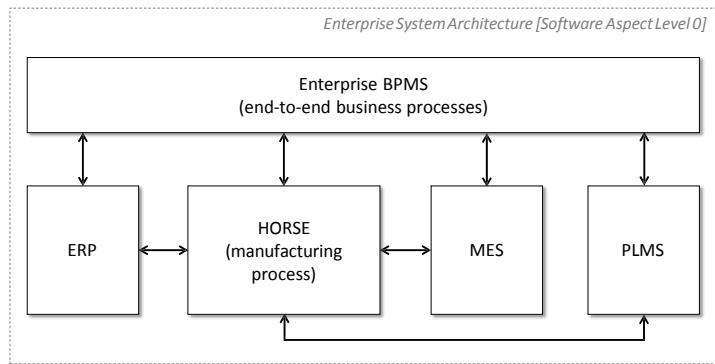


Figura 13.3: Industria 4.0 e architetture software [18]

### 13.3 Sistemi informativi per la pubblica amministrazione

I sistemi informativi per la pubblica amministrazione sono caratterizzati, tra l'altro, da:

- grande numero e varietà di attori sia pubblici sia privati che utilizzano/gestiscono il sistema informativo;
- vari sistemi informativi autonomi collegati per cooperare (per esempio, in ambito sanitario, sistemi regionali, comunali, ospedalieri, ecc.);
- regole di business e processi aziendali complessi e soggetti a normativa variabile.

Diventano sempre più importante il progetto e la gestione di sistemi informativi per le pubbliche amministrazioni volti a gestire per esempio la spesa sanitaria, le tasse e i contributi socio/assistenziali ecc., con la necessità di migliorare le prestazioni e la connessione di/fra enti responsabili dei servizi e con riduzione dei costi di erogazione dei servizi stessi. Anche questi sistemi stanno integrando varie componenti, tra cui le componenti per esempio di monitoraggio e domotica in ambito sanitario, che richiedono processi e sistemi di alto valore e basso costo, da affiancare a modalità di gestione che facilitino la condivisione di informazioni tra i diversi attori che erogano i servizi in ottica di comunità territoriale.

L'utilizzo dell'IT a supporto dei servizi pubblici è una leva strategica per l'integrazione di dati e processi in ambito servizi pubblici: l'utilizzo dell'IT si va gradualmente intensificando anche nel panorama della pubblica amministrazione italiana. Un modello molto utilizzato consiste nel declinare i modelli delle smart cities nel settore pubblico, attività che richiede lo sviluppo di un supporto pervasivo di tipo IT ai processi delle pubbliche amministrazioni e una



Figura 13.4: Il modello generale dei processi a supporto delle persone fragili in Attiv@bili

scelta di modelli e strumenti gestionali e tecnologici per il supporto della rete di attori e dei luoghi fisici a cui fa riferimento, per esempio, il cittadino lungo l'intero percorso di un processo qualunque di interazione con la pubblica amministrazione.

Un esempio di sistema informativo distribuito per la pubblica amministrazione è il sistema di Integrated Care realizzato da Regione Lombardia con il progetto Attiv@bili [13]. Il progetto definisce workflow, parametri di valutazione e linee guida per la definizione di piani assistenziali rivolti ai soggetti fragili, partendo dall'analisi dei servizi e dell'infrastruttura tecnologica utilizzata dagli Enti dei territori di riferimento. Il processo di assistenza alle persone fragili, può essere in sintetizzato nelle macro-fasi riportate in Figura 13.4.

Il sistema informativo di supporto a tale scenario proposto e mostrato in Figura 13.5 è basato sulla ripartizione logica della piattaforma Attiv@bili in due strati connessi: il supporto ai processi di assistenza e il supporto localizzato agli ambienti assistiti.

Lo strato Sistema di Orchestrione e Integrazione flessibile gestisce l'interoperabilità applicativa fra processi e i dati assistenziali. Si tratta di una piattaforma alla quale i vari attori coinvolti nel processo socio-assistenziale (ASL, Comune, Enti erogatori, assistiti e loro familiari) si possono interfacciare sia direttamente sia attraverso meccanismi di interoperabilità, con i sistemi software che utilizzano nello svolgimento delle loro specifiche attività.

Ciascun software si interfaccia al Sistema di Orchestrione come sistema fornitore di informazioni (Producer) o come sistema consumatore di informazioni (Consumer), nella forma di messaggi scambiati (in formato SOAP, tipico dei sistemi a servizi).

Dal punto di vista tecnico, il sistema adotta un modello di architettura a eventi EDA (Event-Driven Architecture, EDA) che implementa un paradigma “publish/subscribe” basato sullo scambio di messaggi. Il sistema di orchestrione ha il compito di ricevere le notifiche di evento provenienti da un Producer e instradarle sulla base delle regole di sottoscrizione alle quali i sistemi Consumer hanno precedentemente aderito. Questa soluzione, molto utilizzata nei sistemi informativi per la pubblica amministrazione, consente interazioni di tipo uno-a-

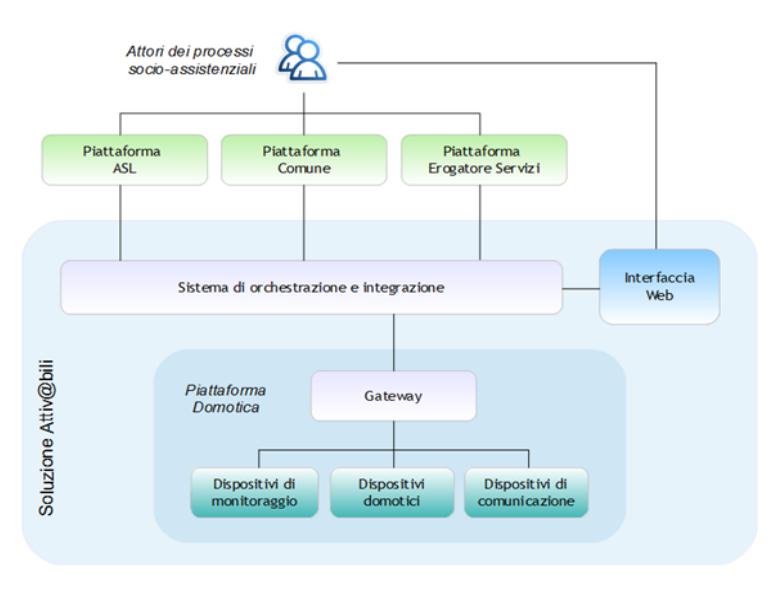


Figura 13.5: Il sistema informativo integrato di Attiv@bili

molti mediante l'invio di notifiche di evento e risulta adatto per coordinare più processi paralleli a cui partecipano diverse componenti applicative.

Rispetto ai sistemi software da interfacciare alla soluzione Attiv@bili, il sistema di orchestrazione non impone alcun vincolo architettonico o tecnologico: possono essere integrati pacchetti applicativi di tipo stand-alone o client-server o applicazioni web, realizzati con qualsiasi linguaggio di programmazione. Al rispettivo fornitore è soltanto richiesto di implementare un connettore (o adapter) per la comunicazione con il sistema di orchestrazione attraverso un Web Service, mediante l'invio e ricezione di messaggi con protocollo SOAP.

Lo strato Piattaforma domotica e di monitoraggio ambienti controllati consiste in un sistema di collegamento e integrazione tra i dispositivi collocati presso l'alloggio assistito, che esercita funzioni di controllo per i dati raccolti localmente dai dispositivi presso l'alloggio. Su tale strato si innestano i diversi dispositivi domotici, di monitoraggio e di comunicazione, che interagiscono con il soggetto assistito e con l'ambiente in cui risiede. A partire dai dati raccolti la soluzione prevede un intervento a supporto dell'assistito, che può essere automatizzato. La Piattaforma Domotica e di Monitoraggio si basa su un gateway unificato che agisce da connettore tra il sistema di orchestrazione e i dispositivi domotici collocati presso la residenza dell'assistito. La connessione del gateway a Internet può essere fornita sia da un servizio ADSL sia da una connessione mobile 3G/4G, a seconda della disponibilità presenti sul territorio. Per la connessione verso i dispositivi domotici, il gateway unificato supporta invece diverse tecnologie: USB, Ethernet, Bluetooth, Wi-Fi.



**Parte V**

**Bibliografia e indice  
analitico**



# Bibliografia

- [1] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati. *Digital Privacy: Theory, Technologies and Practices*. Auerbach Publications, 2008.
- [2] R N Anthony. *Planning and Control: a Framework for Analysis*. Cambridge MA: Harvard University Press, 1965.
- [3] D. Ardagna, M.G. Fugini, B. Pernici, and P. Plebani. *Sistemi informativi basati su web*. Franco Angeli, Serie Sistemi Informativi, vol. 6, 2006.
- [4] P. Baltzan and A. Phillips. *Business Driven Information Systems*. 2015.
- [5] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice (3rd Edition)*. Addison-Wesley Professional, 2012.
- [6] Scott A. Bernard. *An Introduction To Enterprise Architecture: Third Edition*. AuthorHouse, 2012.
- [7] Giampio Bracchi, Chiara Francalanci, and Gianmario Motta. *Sistemi Informativi d'Impresa*. McGraw Hill, 2010.
- [8] Cinzia Cappiello, Pierluigi Plebani, and Monica Vitali. Business Process Modeling Notation (BPMN), 2015.
- [9] Silvana Castano, Mariagrazia Fugini, Giancarlo Martella, and Pierangela Samarat. *Database security*. Addison-Wesley & ACM Press, 1995.
- [10] R. de Koster and J. Zuidema. Commercial returns in a mail order company: the Wehkamp case. In *S. Flapper, J. van Nunen, and L. Wassenhove; Managing Closed-Loop Supply Chains*, pages 97–106. Springer, 2005.
- [11] Lois Evans. Protecting information assets using ISO/IEC security standards. *Information Management*, 50(6):28, 2016.
- [12] Mariagrazia Fugini, Fabrizio Maio, and Pierluigi Plebani. *Sicurezza dei Sistemi Informatici*. Apogeo, 2001.

- [13] M.G. Fugini, J. Finocchi, M. Vitali, P. Locatelli, L. Gastaldi, G. Garavaglia, F. Cirilli, and S. Moraschi. Uso delle piattaforme ICT a supporto dei servizi socio-sanitari: il progetto Attiv@bili. *MECOSAN - Management ed economia sanitaria, Franco Angeli*, 98, 2016.
- [14] Gerben Wierda. *Mastering Archimate - Edition III*. R&A, 2017.
- [15] P. Grefen. Networked business process management. *International Journal of IT/Business Alignment and Governance*, 4(2):54–82, 2013.
- [16] P. Grefen, H. Ludwig, and S. Angelov. A three-level framework for process and data management of complex e-services. *International Journal of Cooperative Information Systems*, 12(4):487–531, 2003.
- [17] P. Grefen, N. Mehandjiev, G. Kouvas, G. Weichhart, and R. Eshuis. A dynamic business network process management in instant virtual enterprises. *Computers in Industry*, pages 86–103, 2009.
- [18] P. Grefen, I. Vanderfeesten, and G. Boultadakis (eds.). HORSE, Beta working paper, 2016.
- [19] Paul Grefen. *Beyond E-Business: Towards networked structures*. Routledge, 2016.
- [20] Y. Hoffner, H. Ludwig, C. Gülcü, and P. Grefen. Architecture for cross-organisational business processes. In *Proceedings 2nd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 2–11. IEEE, 2000.
- [21] ISO/IEC/IEEE. Iso/iec/ieee 42010:2011 - system and software engineering: architecture description. XXX, December 2011.
- [22] J. Jestion and J. Nelis. *Business Process Management, Second Edition: Practical Guidelines to Successful Implementations*. Butterworth-Heinemann, 2008.
- [23] Marc Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, 2016.
- [24] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm, 15th Edition*. Pearson, 2017.
- [25] A.M. Lindstrom. *Clicks, Bricks & Brands*. Kogan Page, 2001.
- [26] C. Liu, Q. Li, Y. Zhang, M. Indulska, and X. Zhao. Introduction to the first workshop on collaborative business processes (CBP 2007). In *Proceedings Business Process Management Workshops*, pages 181–184. Springer, 2008.
- [27] N. Mehandjiev and P. Grefen. *Dynamic Business Process Formation for Instant Virtual Enterprises*. Springer, 2010.

- [28] H. Mintzberg. *Structure in Fives: Designing Effective Organizations.* Prentice Hall, 1992.
- [29] Maurizio Pighin and Anna Marzona. *Sistemi informativi aziendali. Struttura e processi.* Pearson, 2011.
- [30] Maurizio Pighin and Anna Marzona. *Sistemi informativi aziendali: struttura e processi.* Pearson, 2011.
- [31] Federico Pigni, Aurelio Ravarini, and Donatella Sciuto. *Sistemi per la gestione dell'informazione.* Apogeo, 2009.
- [32] M. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance.* Free Press, 1985.
- [33] Pierangela Samarati and Ravi Sandhu. *Database Security X: Status and prospects.* Springer, 2016.
- [34] Signavio. BPM editor, <http://academic.signavio.com>.
- [35] Craig Van Slyke and France Belanger. *Information Systems for Business: An Experiential Approach.* John Wiley & Sons Inc, 2011.
- [36] J.F. Sowa and J.A. Zachman. Extending and formalizing the information systems architecture. *IBM Systems Journal*, 31(3):590–616, 1992.
- [37] R. Taylor, N. Medvidovic, and E.M. Dashofy. *Software Architecture: Foundations, Theory, and Practice.* Wiley, 2009.
- [38] The Open Group. *The ArchiMate Enterprise Architecture Modeling Language, 3.0.* The Open Group, 2016.
- [39] HA Madni Uppal, Memoona Javed, and MJ Arshad. An overview of intrusion detection system (IDS) along with its commonly used techniques and classifications. *International Journal of Computer Science and Telecommunications*, 5(2):20–24, 2014.
- [40] Carlo Vercellis. *Business Intelligence: modelli matematici e sistemi per le decisioni.* McGraw-Hill, 2006.
- [41] Mathias Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition.* Springer, 2012.
- [42] John A. Zachman. A framework for information systems architecture. *IBM Syst. J.*, 26(3):276–292, September 1987.

# Indice analitico

- Sales Force Automation (SFA), 142  
accesso ai dati - D, 194  
accuratezza, 163  
agente, 216  
alberi di decisione, 163  
albero di decisione binario, 164  
albero di decisione generale, 164  
albero di decisione multivariato, 164  
albero di decisione univariato, 164  
Algoritmi a chiave asimmetrica (o a chiave pubblica), 219  
Algoritmi a chiave simmetrica (o a chiave segreta), 219  
apprendimento non supervisionato, 161  
apprendimento supervisionato, 161  
APS - Advanced Planning and Scheduling, 125  
architettura di riferimento, 108  
architettura funzionale, 36  
architettura market-level, 106, 109  
architettura party-level, 107  
architettura System-level, 115  
architettura system-level, 107  
as-is, 37  
attacchi, 217  
autenticazione, 216, 229  
autenticità, 216
- B2B, 60  
B2C, 61  
benefici, 46  
BI, *vedi anche* Business Intelligence  
BPMN (Business Process Modeling Notation), 90, 91  
BPMS - Business Process Management Systems, 126
- business driver, 62, 66  
Business Intelligence, 123, 146  
business process, 11, 89, 90  
business rules, 20, 94, 133
- Certification Authority - CA, 227  
certificato a chiave pubblica - PKC, 227  
 ciclo di vita, 46  
 classificazione, 162  
 clonazione, 201  
 cloud computing, 207, 210  
 cluster, 166  
 clustering, 166  
 community cloud, 209  
 confidenza, 161  
 conoscenza, 14  
 costi, 46  
 COTS, 43, 48, 125, 131, 137, 145  
 CRM, 125, 135, 138, 168  
 Customer Relationship Management, *vedi anche* CRM
- data mart, 150  
data mining, 144, 160  
data warehouse, 123, 140, 144, 147, 149  
DBMS, 23, 126, 176  
Denial of Service - DoS, 217  
dice, 158  
Dimensional Fact Model (DFM), 153  
dimensioni di qualità, 44  
Discretionary Access Control (DAC), 234  
disponibilità, 216  
drill-down, 157  
DW, *vedi anche* data warehouse
- E-Business, 125

- EB, *vedi anche* E-Business  
efficacia, 62, 63  
    reach, 63  
    richness, 64  
efficienza, 62, 64  
Enterprise Architecture - EA, 26, 33, 38, 250  
Enterprise Resource Planning, *vedi anche* ERP  
Enterprise Service Bus (ESB), 188  
entropia, 163  
ERP, 125, 132  
ESB (Enterprise Service Bus), 127  
ETL, 151, 159, 174  
  
falso negativo, 164  
falso positivo, 164  
FASMI, 149  
fattibilità tecnica, 44  
firewall, 240  
firma digitale, 225  
framework di Zachman, 33  
funzione di somiglianza, 166  
funzioni di hash, 224  
  
gestione del sistema informativo, 37, 44  
gestione delle chiavi, 226  
  
HOLAP, 155  
Human Resource, 19  
hybrid cloud, 209  
  
IDS (intrusion Detection System, 244  
indicatori, 62  
Infrastructure as a Service, 208  
integrità, 215  
interpretabilità, 163  
  
k-means, 166  
Key Performance Indicator (KPI), 62  
  
layer, 193  
logica applicativa - A, 194  
  
macchina virtuale - VM, 205  
Make-or-Buy, 43, 125, 145  
  
Buy, 43, 48, 131, 169  
Make, 43  
Mandatory Access Control (MAC), 234  
Manufacturing Execution Systems, *vedi anche* MES  
MES, 125  
Message-Oriented Middleware - MOM, 179  
metadati, 152  
minacce, 216  
modelli di deployment del Cloud Computing, 209  
modello di business, 72  
MOLAP, 154  
  
obiettivi, 42  
oggetti dell'interazione, 61  
OLAP (On-line Analytical Processing), 24, 147, 149  
OLTP (On-line Transaction Processing), 23  
Orizzonte temporale, 61  
Outsourcing, 212  
outsourcing, 43, 44  
  
partecipanti (party), 60  
partizionamento, 203  
pianificazione strategica, 40  
Piramide di Anthony, 16  
Platform as a Service, 208  
PLM (Product Lifecycle Management), 135  
politiche di sicurezza, 232  
portafoglio applicativo, 123  
Porter  
    attività primarie, 18  
    attività secondarie, 18  
    value chain, 18  
Precision, 165  
presentazione - P, 193  
privacy, 216  
private cloud, 209  
processi inter-organizzazione, 86  
processi intra-organizzazione, 86  
processo, 16  
progettazione delle soluzioni, 42

- proprietà di sicurezza, 215
- public cloud, 209
- Public Key Infrastructure - PKI, 227
- punto-a-punto, 179
- QoS - Quality of Service, 184
- qualità
  - affidabilità, 45
  - disponibilità, 45
  - maturità tecnologica, 45
  - scalabilità, 37, 45
  - sicurezza, 45
- RACS - Reliable Array of Cloned Services, 201
- RAPS - Reliable Array of Partitioned Services, 203
- Recall, 165
- regole associative, 161
- requirements pull, 27, 31
- REST - Representational state transfer, 186
- reti Bayesiane, 163
- reti neurali, 163
- riservatezza, 216
- risorsa, 15
- robustezza, 163
- ROLAP, 155
- roll-up, 157
- Rule Engine, 126
- SaaS, 210
- scalabilità, 163
- scenario, 71
- schema a fiocco di neve, 155
- schema a stella, 155
- SCM (Supply Chain Management), 135, 137
- server farm, 201
- service orchestration, 186
- service registry, 185
- servizi, 127, 183
- shared-disk, 202
- shared-nothing, 202
- single tiered, 194
- sistema legacy, 45, 194
- SLA - Service Level Agreement, 184
- slice, 157
- SOA (Service Oriented Architecture), 185
- SOA - Service Oriented Architecture, 183
- SOAP - Service Oriented Architecture Protocol, 185
- Software as a Service, 208
- stakeholder, 32
- studio di fattibilità, 40, 42
- supporto, 161
- technology push, 27, 31
- test set, 162
- three tiered (3-Tiers), 196
- tier, 194
- to-be, 37
- training set, 162
- transazione, 22
- two tiered (2-Tiers), 195
- UIS - User Interface Systems, 126
- utility computing, 207
- velocità, 163
- vero negativo, 165
- vero positivo, 165
- violazioni, 217
- virtualizzazione, 205