

# Unità 7

## Capire l'infrastruttura digitale

### Livello 3 - approfondimento

#### Temi trattati all'interno dell'Unità

---

- Architetture Digitali.
- Architetture Web
- Service Oriented Architecture (SOA).
- Cloud Computing

#### Sommario

---

<b>ARCHITETTURE: C'ERA UNA VOLTA...</b>	1
<b>ARCHITETTURE PER I SERVIZI</b>	3
<b>SOA E INNOVAZIONE</b>	3
<b>DALLA SOA AL CLOUD</b>	3
<b>CLOUD: I SERVIZI</b>	4
<b>CLOUD COMPUTING: VANTAGGI E SVANTAGGI</b>	6
<b>BIBLIOGRAFIA</b>	7

#### ARCHITETTURE: C'ERA UNA VOLTA...

In origine il software applicativo, cioè lo strato superiore dell'architettura software era costituito quasi esclusivamente da software sviluppato su specifiche o almeno sulla base dei requisiti comunicati dall'azienda stessa.

A partire dagli anni '90 si è andato sempre più affermando l'acquisto di prodotti applicativi che supportano in modo integrato i processi gestionali delle aziende, riferendosi spesso a riconosciute "best practices".

Tali prodotti hanno preso il nome di prodotti **ERP (Enterprise Resource Planning)** e hanno costituito e ancora costituiscono la componente dominante dell'architettura software delle aziende medio-grandi, ma sono molto diffusi, nelle versioni più semplici, anche nelle aziende più piccole.

Tali prodotti presentano una ampia gamma di parametri in modo da potersi adattare alle diverse situazioni aziendali. L'assegnazione di tali parametri, che richiede le competenze di un analista di processi più che di un programmatore, prende il nome di "customizzazione" del prodotto.

Per far fronte a situazioni dove l'assegnazione dei parametri non è sufficiente a soddisfare le esigenze di supporto ai processi di una certa azienda, i prodotti ERP offrono un linguaggio di programmazione con il quale si possono arricchire le funzioni generali del prodotto con funzionalità "ad hoc".

Questa attività che richiede competenze combinate di analisi di processo e di programmazione, prende il nome di "personalizzazione del prodotto".

E' ovvio che i prodotti ERP diventano parte rilevante dell'architettura delle aziende che li adottano e un investimento, fra licenze, customizzazione ed eventuali personalizzazioni, assai cospicuo, oltre a costituire una forte dipendenza dal fornitore che adegua progressivamente il prodotto alle nuove normative o aggiungendo ulteriori funzionalità.

Pertanto, come per ogni prodotto strategico acquisibile sul mercato è necessario salvaguardare l'investimento considerando l'affidabilità del fornitore e la rilevanza del prodotto sul mercato, oltre naturalmente alla valutazione dell'aderenza dei processi supportati rispetto agli obiettivi dell'azienda.

Per **architettura** digitale si intende l'insieme dei criteri in base ai quali è progettato e realizzato un sistema informatico o un dispositivo che ne fa parte.

**L'architettura software** è l'organizzazione fondamentale di un sistema, definita dai suoi componenti, dalle relazioni reciproche fra i componenti e tra questi ultimi e l'ambiente.

Descrivere l'architettura software significa elencarne le parti costituenti ed illustrarne i rapporti interfunzionali.

Possiamo semplificare gli elementi di base che compongono un'applicazione informatica in questo modo:

- **Dati**  
Sono i dati da trattare (dati sugli utenti, sul servizio offerto...).
- **Logica**  
È la logica applicativa (*business logic*), ovvero le funzionalità che l'applicazione deve offrire, il tipo di trattamento dei dati previsto, quali operazioni si possono fare...
- **Interfaccia**  
Può essere una interfaccia web (ad esempio un browser), o una interfaccia *stand-alone* (installata sulla macchina su cui è presente l'applicativo software al completo).

Questi elementi vengono gestiti nelle differenti architetture. Le **Architetture monoliti** erano popolari ai tempi dei *mainframe* (il *mainframe* o sistema centrale è una tipologia di computer caratterizzata da prestazioni di elaborazione di alto livello di tipo centralizzato), erano praticamente composti dal codice indivisibile che controllava l'intera logica dell'applicazione, dalla gestione (e memorizzazione) dei dati all'interfaccia utente.

I *mainframe* erano adatti a eseguire pochi processi *stand-alone*, anziché diversi processi comunicanti. All'epoca non esistevano ancora le basi dati.

Le **Architetture client/server** Sono nate alla fine degli anni '70 e sono state gestite fino alla metà degli anni '80. Consistevano in server più piccoli ed economici dei *mainframe* e in *workstations* (stazioni di lavoro) e PC. Erano presenti le basi dati e le applicazioni erano suddivise in due parti:

- **Back-end (server):** gestione di database e dei compiti di manipolazione dei dati.
- **Front-end (client):** gestione interfaccia utente.

All'inizio degli anni '90 vengono introdotte le Architetture a tre livelli (three-tier) che gestivano in modo esplicito l'aspetto della logica riportandola a livello architeturale.

In questo caso i tre livelli sono:

- **livello 1** : gestione dei dati (DBMS, file XML, ....),
- **livello 2** : *business logic*, logica (processamento dei dati, ...),
- **livello 3** : interfaccia utente (presentazione dei dati, servizi).

Ogni livello ha i suoi obiettivi e vincoli e nessun livello interviene sugli altri. Inoltre non c'è comunicazione diretta tra livello 1 e livello 3:

L'evoluzione dell'architettura a tre livelli in architettura web-based avviene soprattutto con la presenza di Internet, queste applicazioni hanno:

- **Interfaccia utente:** gestita sul browser utente.
- **Logica applicativa:** gestita sul server, che si interfaccia al web attraverso il web server.
- **Livello dei dati:** su database, eventualmente localizzato su una macchina diversa da quella del web server per questioni di sicurezza.

## ARCHITETTURE PER I SERVIZI

L'ulteriore evoluzione delle architetture web-based ha permesso di gestire le esigenze funzionali e applicative non più con dei sistemi dedicati ma con servizi forniti in base alle necessità.

Un servizio è una funzionalità (di *business*) realizzata tramite un componente che rispetta un'interfaccia.

La **Service-Oriented Architecture (SOA)** è un'architettura che utilizza una logica orientata ai servizi per supportare le esigenze degli utenti decomponendo le singole applicazioni in funzioni elementari riutilizzabili.

Nell'architettura SOA è possibile modificare, in maniera relativamente semplice, le modalità di interazione tra i servizi e la combinazione con cui i servizi vengono utilizzati nel processo, così come diventa più agevole aggiungere nuovi servizi e modificare i processi per rispondere alle specifiche esigenze di business.

Il processo di business non è più vincolato a una specifica piattaforma o a un'applicazione, ma può essere considerato come un componente di un processo più ampio e quindi riutilizzabile o modificabile.

La SOA agevola l'interazione tra le diverse realtà permettendo alle attività di business di sviluppare processi efficienti, sia internamente sia esternamente, e aumentare di conseguenza la flessibilità e l'adattabilità.

## SOA E INNOVAZIONE

Oggi le organizzazioni si muovono in un contesto di alta competitività, incertezza, e integrazione globale di informazioni e processi.

Viene richiesta *agilità* per offrire i propri servizi in modo più efficiente ed efficace. Per offrire servizi innovativi e per operare in nuovi mercati un'organizzazione deve poter cambiare rapidamente e in modo flessibile i propri processi di business.

Inoltre, il ruolo delle tecnologie informatiche è tale che il business di un'organizzazione può essere flessibile solo quanto la sua infrastruttura IT.

Un'architettura basata sui servizi si propone di far sì che l'IT sostenga – anziché ostacolare – l'agilità di business delle organizzazioni.

L'innovazione può essere definita come il processo di cambiamento per realizzare/fare qualcosa di nuovo. Nei sistemi tradizionali hardware, software e reti sono integrati in modo rigido e la SOA rende più semplici i cambiamenti e l'innovazione.

Nella SOA l'IT è realizzato come un insieme di componenti che è facile assemblare e riconfigurare e ciascuno di questi componenti è un servizio di business erogato dalla propria o anche da un'altra organizzazione.

La SOA utilizza lo stesso principio dei mattoncini da costruzione:

- in una SOA i sistemi informatici sono costruiti con componenti assemblati in modo modulare che possono essere riconfigurati facilmente;
- ciascun mattoncino è un servizio di business;
- in una SOA è possibile assemblare i servizi in modo semplice e flessibile per creare un processo adatto al mercato;
- se poi nel mercato c'è bisogno di un processo diverso, anziché ripartire da zero, si utilizzano gli stessi mattoncini riconfigurandoli per fare qualcosa di diverso in un nuovo processo.

La SOA è come il guardaroba: ciascun servizio è come un capo d'abbigliamento e questi servizi possono essere integrati in modo flessibile.

## DALLA SOA AL CLOUD

Da un punto di vista architetturale, il cloud condivide col SOA l'orientamento ai servizi. Esistono però differenze tra i due concetti: nella SOA, il servizio è un principio di progettazione e architettura del sistema.

Il *Cloud Computing* invece colloca il concetto di servizio ad un livello di generalità e astrazione maggiore.

Per il Cloud il servizio è il rapporto tra l'utilizzatore e la generica risorsa computazionale di interesse, in questo contesto il software costituisce solo un sottoinsieme del dominio complessivo del servizio fornito.

Con il Cloud il principio di *service-oriented-computing* evolve verso l'idea di *computing-as-a-service*: applicazioni, piattaforme, funzionalità di rete, dispositivi di calcolo e di memorizzazione vengono offerti come servizi.

La metafora della nuvola (**Cloud**) è usata spesso nei diagrammi di rete (rappresenta una rete a scatola nera).

Una nuvola è infatti, per sua natura, opaca e, di solito, è considerata molto grande e distante così come nel *cloud computing* la realizzazione interna è opaca, i confini sono spesso sfumati.

Poi, le nuvole si possono sovrapporre e le nuvole possono cambiare dinamicamente forma, così come le dinamiche del *cloud computing*.

Il *Cloud Computing* è il risultato della stretta interconnessione tra ricerca scientifica e tecnologia informatica sviluppatasi negli ultimi trent'anni.

La ricerca scientifica su scala mondiale richiedeva e richiede grandi quantità di risorse di calcolo e grandi quantità di dati geograficamente distribuiti.

In questo modo nasce il *Grid Computing*, ovvero l'infrastruttura basata su Internet per la realizzazione di *e-Science* a livello globale. Le risorse vengono condivise e non ci si limita solo allo scambio di file ma si ha anche accesso diretto a computer, software e hardware. In questo modo aumentano le risorse computazionali a tutto tondo.

## Il Sistema GRID

La più grande Grid europea è quella del CERN di Ginevra in Svizzera, che viene utilizzata per la condivisione delle risorse di calcolo per il processo di trasformazione dell'acceleratore di particelle europeo.

Il Grid collega il CERN a undici centri situati in Europa, Canada, Asia con l'obiettivo di registrare i dati dell'acceleratore di particelle.

La necessità non solo dei ricercatori scientifici, ma anche del mondo del business di condividere delle risorse distribuite, ha guardato alla creazione e alla condivisione di potenze di calcolo e di risorse passando dal concetto di Grid alla *Utility Computing*, e la *Utility Computing* ha generato la possibilità di fornire servizi a consumo (*Information Utilities*).

## CLOUD: I SERVIZI

Nel Cloud sono distinguibili 3 livelli di servizi a consumo:

### IaaS (Infrastructure as a Service)

Rappresenta il primo passo nel Cloud Computing e consiste nell'utilizzare l'infrastruttura messa a disposizione dal fornitore per eseguire la propria applicazione, a fronte di un pagamento in base all'utilizzo dell'infrastruttura stessa. Solitamente, questo tipo di servizio offre la parte relativa al *networking*, allo *storage* dei dati, ai server fisici e ai software di virtualizzazione.

### PaaS (Platform as a Service)

Rappresenta il passo immediatamente successivo all'IaaS, in quanto il provider fornisce non solo l'infrastruttura ma anche il sistema operativo, i *middleware* (cioè i software intermediari) e l'ambiente necessario per eseguire l'applicazione dell'utente.

### SaaS (Software as a Service)

In questo caso non viene eseguita un'applicazione proprietaria del cliente, ma il cliente stesso paga il diritto di utilizzo di un'applicazione messa a disposizione dal provider, senza preoccuparsi di come essa venga realizzata e gestita nel Cloud. L'unica preoccupazione del cliente in questo caso, oltre ovviamente alla scelta della corretta applicazione che soddisfi le sue necessità, è quella di gestire il numero di licenze richieste in funzione del numero di utenti.

Il *Cloud Computing* prevede inoltre quattro modelli di distribuzione (*deployment*)

#### 1. Public Cloud

L'infrastruttura di cloud è resa disponibile al pubblico ed è posseduta e gestita da un'organizzazione che vende servizi cloud.

Caratteristiche:

- servizi e infrastrutture sono fornite esternamente attraverso Internet,
- efficienza di condivisione delle risorse,
- rischi di vulnerabilità.
- 

Un cloud pubblico risulta la scelta più logica qualora:

- il proprio carico di lavoro standard per applicazioni venga usato da molte

persone, ad esempio l'email,

- si renda necessario testare e sviluppare un codice di applicazioni,
- si sia entrati in possesso di applicazioni SaaS (*Software as a Service*) da un venditore che ha una strategia di sicurezza ben implementata,
- si necessiti di capacità incrementale (abilitazione all'aggiunta di capacità del computer per periodi di picco),
- si stiano svolgendo progetti collaborativi,
- si stia svolgendo un progetto di sviluppo di software *ad hoc*, utilizzando una Piattaforma come Servizio di offerta del cloud (PaaS).

## 2. Private Cloud

L'infrastruttura di cloud è solo per una singola organizzazione e può essere gestita dall'organizzazione stessa o da terze parti (nota: secondo alcune definizioni questa però non è considerata propriamente una forma di *cloud computing*).

Caratteristiche:

- servizi e infrastrutture mantenute in una rete privata,
- livello di sicurezza e controllo alto,
- costi di acquisto e mantenimento del software e infrastrutture,
- meno vantaggi sui costi.

Un cloud privato risulta la scelta migliore se:

- il business consiste nella gestione dei propri dati e dei propri applicativi, e pertanto il controllo e la sicurezza sono fondamentali;
- il proprio business è parte di un'industria che deve conformarsi a sicurezza stretta e problematiche di privacy dei dati;
- la propria azienda è sufficientemente grande da riuscire a far girare un cloud data center di nuova generazione che sia efficiente.

## 3. Hybrid Cloud

L'infrastruttura di cloud è la composizione di due o più cloud che rimangono però come entità separate, ma che sono legate insieme per sostenere la portabilità di dati e/o delle applicazioni.

Caratteristiche:

- utilizzo di *public cloud* e *private cloud* per la gestione delle risorse,
- gestione di *multiple cloud* come un'unica entità.

possibili problematiche:

- rendere possibile comunicazione tra più cloud,
- garantire sicurezza di comunicazione.

## 4. Community Cloud

L'infrastruttura di cloud è condivisa da più organizzazioni e sostiene gli interessi di questa comunità di organizzazioni, può essere gestita dalla comunità stessa o da terzi.

## **CLOUD COMPUTING: VANTAGGI E SVANTAGGI**

Per utilizzare le architetture e i servizi cloud è utile capire quali sono i vantaggi e gli svantaggi che possiamo avere come utenti per poter poi scegliere, secondo i bisogni, se usare o meno le potenzialità di questi servizi.

### **Vantaggi per l'utente**

- ✓ Nessun software da scaricare e installare sul proprio computer.
- ✓ Non si deve acquistare una licenza software. Ci si abbona al servizio, che spesso è gratuito.
- ✓ Nessuna necessità di upgrade del software quando vengono aggiunte nuove funzioni o eliminati problemi esistenti.
- ✓ Si può accedere ai propri documenti da ogni computer connesso in rete.
- ✓ È possibile condividere i documenti con altri utenti, senza bisogno di un proprio server.
- ✓ Nessun problema in caso di *crash* del proprio computer: i documenti sono al sicuro sul server.

### **Svantaggi per l'utente**

- ✓ È necessaria una connessione internet con banda adeguata.
- ✓ *Data security*: tutti i dati risiedono su un server remoto, di cui non si ha il controllo (anche se si possono effettuare regolari *backup*).
- ✓ *Data privacy*: i dati sono visibili e utilizzabili da terzi.
- ✓ Minor completezza funzionale: attualmente, le funzioni fornite dalle applicazioni cloud sono meno complete di quelle fornite dai software tradizionali.
- ✓ Non è agevole lavorare *offline*, per ricevere e trasmettere le modifiche occorre essere connessi.

Allo stesso modo, dal punto di vista di organizzazioni e aziende che intendano utilizzare servizi cloud per il proprio business, è utile considerare i vantaggi e gli svantaggi forniti da queste architetture.

### **Vantaggi per le aziende**

- ✓ Eliminazione/riduzione delle infrastrutture software/hardware e dei costi del personale per la gestione dei sistemi.
- ✓ Riduzione/eliminazione dei costi fissi.
- ✓ Scalabilità dei servizi: tutto quello che serve, quando serve, solo se serve.
- ✓ Costi *pay-per-use*.

### **Svantaggi per le aziende**

- ✓ I dati risiedono su un server remoto, di cui si ha scarso controllo; ne conseguono possibili problemi d'integrità dei dati.
- ✓ Servizi standardizzati che non sempre soddisfano a pieno le esigenze aziendali.
- ✓ Non è agevole lavorare *offline*, per ricevere e trasmettere le modifiche occorre essere connessi.

Per ultimo è utile analizzare gli aspetti legati alla sicurezza nel cloud.

Il *Cloud Computing* comporta la migrazione dei dati dai sistemi locali dell'utente ai sistemi remoti del fornitore, il quale svolge un ruolo centrale nell'adozione delle misure necessarie a garantire la sicurezza dei dati.

Spostare la gestione della sicurezza a un fornitore terzo di servizi non preserva le imprese da rischi possibili:

- L'utente perde il controllo diretto ed esclusivo sui dati.
- La riservatezza e la disponibilità delle informazioni dipendono dai meccanismi di sicurezza adottati dal *service provider*.
- Il servizio può essere il risultato di una catena di servizi acquisiti presso altri *service provider*, diversi dal fornitore con cui l'utente stipula il contratto di servizio.
- L'utente non conosce sempre chi, dei vari gestori di servizi intermedi, può accedere a determinati dati.

- La conservazione dei dati in luoghi geografici differenti ha riflessi immediati su:
  - la normativa applicabile in caso di contenzioso tra l'utente e il fornitore,
  - le disposizioni nazionali che disciplinano il trattamento, l'archiviazione e la sicurezza dei dati.
- L'adozione da parte del fornitore di tecnologie proprie può rendere complessa per l'utente:
  - la transizione di dati e documenti da un sistema cloud ad un altro,
  - lo scambio di informazioni con soggetti che utilizzano servizi cloud di fornitori differenti,
  - portabilità e l'interoperabilità dei dati a rischio.

Il fornitore dei servizi cloud, in base al tipo di servizio offerto, si assume la responsabilità di preservare la riservatezza dei dati, preservare l'integrità dei dati, preservare la disponibilità dei dati.

## **BIBLIOGRAFIA**

TITOLO	AUTORE	EDIZIONI	ANNO
Introduzione alle service oriented architecture (SOA)	Nestore Fantuzzi	Hoepli	2008
Cloud Computing e IT as a Service	Reportec	Reportec	2012
Cloud computing - Aspettative, problemi, progetti e risultati di aziende passate al modello «as a service»	Antonella Ferrari Emanuela Zanleone	Franco Angeli	2011
Vivere nella nuvola. Oltre 200 servizi e programmi gratuiti alla portata di tutti per	Gianluigi Bonanomi	Hoepli	2013
Cloud computing. Architettura, infrastrutture, applicazioni	George Reese	Tecniche Nuove	2010