

WEB SERVICES

Capitolo 1

Service Oriented Computing(SOC) utilizza servizi come la base di supporto allo sviluppo rapido, a basso costo e di facile composizione di applicazioni distribuite. I Servizi riflettono un nuovo approccio alla programmazione “orientato ai servizi” basato sull'idea di composizione di applicazioni scoperte, usate e disponibili sulla rete invece di costruire nuove applicazioni o usando applicazioni disponibili per completare alcuni task. I servizi hanno varie funzionalità che variano da semplici richieste a eseguire sofisticati processi di business che richiedono relazioni peer-to-peer tra servizi, consumatori e fornitori. Tuttavia i servizi sono molto spesso costruiti per essere indipendenti dal contesto nel quale vengono usati. A livello di middleware, un basso accoppiamento richiede che l'approccio orientato ai servizi sia indipendente dalle specifiche tecnologie e dal sistema operativo. In particolare i servizi e la composizione di servizi non si appoggiano a nessun linguaggio di programmazione, utilizzando standard (basati su XML) e protocolli che descrivono interfacce basati su standard aperti. Con l'uso di questi standard, agli sviluppatori è data l'opportunità di accedere ai sistemi e alle applicazioni distribuite sulla rete basate su cosa fanno e non come lo fanno. Le basi su cui si fonda SOC è che un'applicazione non può essere pensata come un singolo processo eseguito all'interno di una singola organizzazione. Il valore di un'applicazione non è più misurata dalle sue funzionalità ma dalla capacità di integrarsi con i suoi ambienti circostanti. Un nuovo tipo di applicazione può basarsi solamente su un insieme di servizi interattivi offrendo interfacce ben definite ai loro potenziali utenti. A queste applicazioni è solito riferirsi come **applicazioni composite**. Il modello del servizio permette di una chiara distinzione tra **service provider** (organizzazioni che provvedono a fornire servizi e assistenza a questi ultimi), **service clients** (organizzazioni che usano tali servizi), **service aggregators** (organizzazioni che consolidano servizi multipli in un unico servizio). I service provider, che provvedono a fornire il servizio, utilizzano infrastrutture distribuite per l'integrazione tra applicazioni enterprise. I client di un servizio possono essere anche altre applicazioni che si trovano dentro o fuori dal contesto enterprise. I servizi devono essere:

- **Tecnologicamente neutrali:** devono essere usati attraverso il numero più basso possibile di standard tecnologici che sono disponibili sulla maggior parte degli ambienti.
- **Basso Accoppiamento:** non devono richiedere conoscenza di alcuna struttura interna o contesto a livello client o server.
- **Supporto alla trasparenza di locazione:** i servizi dovrebbero avere le loro definizioni e informazioni di locazione in un repository come UDDI ed essere accessibili da una varietà di client che può localizzare e usare il servizio.

I servizi possono essere implementati su una singola macchina o su un largo numero e varietà di dispositivi e distribuiti sulla LAN o WAN. Un particolare caso d'interesse è quando i servizi usano Internet e standard di Internet aperti. Il risultato è un **web service** che condivide caratteristiche di più servizi generali, ma richiedono una considerazione speciale perché usano un meccanismo di bassa sicurezza per l'interazione con il servizio. I web service costituiscono un

infrastruttura di computer distribuiti formati da differenti moduli che provano a comunicare tramite la rete formando virtualmente un sistema logico. I web service sono modulari e accessibili sulla rete. I Web services possono variare in funzionalità: da semplici richieste a complete applicazioni di business. I web service permettono agli sviluppatori di costruire applicazioni distribuite usando gli standard di internet e qualunque piattaforma o linguaggio di programmazione che sia richiesto.

2. Il concetto di software come servizio

I web service sono differenti dalle pagine web. Le seconde hanno come target gli utenti, mentre le prime sono sviluppate per essere utilizzate da altre applicazioni. Il concetto di software come servizio è rivoluzionario e apparso prima con il modello ASP. ASP sono un insieme di compagnie che impacchettano software insieme per creare servizi professionali. Un ASP è un entità di terze parti che fa deploy, hosta e gestisce accesso ad applicazioni impacchettate e consegna queste soluzioni software attraverso la rete e clienti multipli da un data center centrale. L'idea di base di ASP era di "affittare" le applicazioni ai consumatori. Anche se è stata la prima a introdurre il concetto di software come servizio, soffriva di molte limitazioni come l'incapacità di sviluppare applicazioni altamente interattive, incapace di provvedere a soluzioni altamente configurabili e non capace di integrare applicazioni. Il risultato era un'architettura monolitica, altamente fragile, specifica per un dato consumatore e non riutilizzabile. Oggi siamo nel mezzo di uno sviluppo significativo del software come servizio. La nuova architettura permette interazioni asincrone a basso accoppiamento basati sugli standard di XML con l'intento di rendere la comunicazione attraverso Internet più semplice. L'uso dei web service provvede ad una soluzione più flessibile. Quando confrontiamo i web service con le applicazioni web le distinguiamo per 4 caratteristiche:

- I web service agiscono come risorse alle altre applicazioni che richiedono queste risorse con o senza l'intervento umano. Questo significa che i web service possono essere chiamati da altri web service. Questo comporta alta flessibilità e adattabilità non disponibile nelle web application.
- I web service sono modulari e conoscono le funzionalità che fanno e quali input sono richiesti per produrre i suoi output. Può anche descrivere i suoi requisiti non funzionali: ad esempio l'area geografica che copre, il costo.
- I web service sono più visibili e gestibili delle applicazioni Web. Lo stato di un web service può essere monitorato e gestito in qualunque tempo usando tool esterni.
- I web service creano concorrenza. Se diversi web service hanno le stesse funzionalità allora diverse organizzazioni potrebbero fare delle offerte per usare il migliore.

3. Cosa sono i Web service?

I web service si basano su insieme di standard di internet aperti che permettono agli sviluppatori di implementare applicazioni distribuite. La cosa più importante è che i web service possono essere integrati in queste applicazioni distribuite per eseguire virtualmente qualunque tipo di task. I web

service possono comunicare con altri usando un vocabolario comune e un insieme delle sue capacità in accordo con la SAO (Service Oriented Architecture).

- Web Service sono moduli debolmente accoppiati: I servizi comunicano con gli altri dinamicamente e usano le tecnologie di internet, rendendo possibile costruire ponti tra sistemi che a volte richiedono complessi sforzi di sviluppo. I sistemi debolmente accoppiati richiedono un più semplice livello di coordinazione e permettono una riconfigurazione più semplice.
- I web service incapsulano semanticamente discrete funzionalità: un web service è un container che conosce tutto su di lui (metodi, parametri, interfacce, tipi di dati e protocolli che usa).
- Si può accedere ai web service programmaticamente: questo permette di cercare e aggiornare informazioni in tempo reale migliorando efficienza e accuratezza. (non hanno interfaccia grafica).
- I web service possono essere dinamicamente trovati e inclusi nelle applicazioni: i webservice possono essere assemblati al volo per svolgere una data funzionalità.
- I web service sono descritti in termini di un linguaggio descrittivo che provvede a caratteristiche funzionali e non.

Le tecnologie web service hanno introdotto un nuovo tipo di logica di business: "la logica globale di business", ovvero i servizi web possono essere usati una volta per svolgere operazioni usate da centinaia di migliaia di consumatori. In più i web service aggiungono sicurezza ovvero mentre i siti usano i firewall del computer per garantire sicurezza e privacy, i web service usano SOAP (Simple Object Access Protocol). SOAP combina HTTP con XML.

4. Web service tipi e caratteristiche

I web service si possono distinguere in due tipi:

1. Informational, o tipo I, supporta solo operazioni in entrata. Aspetta una richiesta la riceve e risponde. È molto comune e di solito è senza stato.
2. Complex, o tipo II, ha un meccanismo di coordinazione tra operazioni in entrata e in uscita e sono con stato.

Informational: Coinvolgono semplici richieste, espongono il proprio contenuto o quello di altre applicazioni a cui sono legate. Si possono suddividere in:

- Content Service: danno accesso al contenuto (esempio: meteo)
- un più complesso servizio di informazioni che provvede all'aggregazione di dati da diversi sistemi. (esempio servizi di viaggio)
- Information syndication service: sono valori aggiunti che pretendono di "inserirsi" in siti commerciali di vari tipi (Marketplace). Sono generalmente offerti da terze parti

Complex: utili alle applicazioni enterprise per comunicare con altre. Come le informational richiedono il supporto a SOAP, XML e UDDI ma richiedono anche emergenti standard:

- processi di business e messaggi in XML.
- Un registro per pubblicare e scoprire processi di business e protocolli di comunicazione.
- Collaborazioni tra partner.
- Terminologie di business standard.
- Un infrastruttura universale dei messaggi.

I web service possono essere suddivisi a seconda di come sono programmati:

- **Programmatic web service:** incapsulano un processo di business programmatico ed espone le funzionalità della logica di business. Sono scritte in linguaggi come Java, C++ e vengono eseguite attraverso un interfaccia WSDL.
- **Interactive web services:** espone le funzionalità nel browser. È un' applicazione che combina un web server, un application server e un database sottostante.