

Architetture orientate ai servizi

Gianluca Pironato

Introduzione

- ▶ **SOA (Service-Oriented Architecture)** è un modello architetturale per gestire la complessità dei sistemi distribuiti.
- ▶ Si basa sulla visione di una applicazione come un insieme di servizi indipendenti e interoperabili, che comunicano tramite protocolli standard.
- ▶ Un esempio semplice, fino a 03:07.

Attori delle SOA

- ▶ **Fornitori di servizi:** entità che offrono unità funzionali autonome per eseguire compiti specifici, definendone i termini d'uso e iscrivendoli in un registro dei servizi.
- ▶ **Bus di servizi (Enterprise Service Buses):** il canale di comunicazione tra i servizi, per gestire trasformazione dei dati e scambio di messaggi, basato sull'uso di protocolli standard.
- ▶ **Registro dei servizi:** un catalogo che descrive i servizi disponibili e le loro funzionalità per permettere agli utilizzatori di scoprirne l'esistenza.
- ▶ **Utilizzatori:** applicazioni che richiedono e sfruttano i servizi.

Bus di servizi (ESB)

- ▶ È un componente centrale che facilita la comunicazione tra i servizi in un'architettura SOA. Per usare N servizi non serviranno N interfacce, ma solo l'ESB che fa da singolo punto d'ingresso a quello che è il registro dei servizi.
- ▶ **Funzionalità:**
 - ▶ Gestione del routing dei messaggi tra servizi.
 - ▶ Trasformazione dei dati tra formati diversi.
 - ▶ Gestione degli errori e della sicurezza.
- ▶ Centralizza la logica di integrazione (non devo usare più interfacce) e riduce la complessità nelle interazioni tra servizi.
- ▶ Introduce però un punto di fallimento centrale e richiede un'attenta gestione e configurazione.

Principi fondamentali delle SOA

- ▶ **Modularità:** ogni servizio è autonomo e svolge una funzione specifica.
- ▶ **Riutilizzabilità:** i servizi possono essere utilizzati da diverse applicazioni.
- ▶ **Interoperabilità:** utilizzo di protocolli standard per garantire la comunicazione tra sistemi diversi.
- ▶ **Scalabilità:** capacità di aggiungere nuovi servizi senza modificare quelli esistenti.

Vantaggi delle SOA

- ▶ **Flessibilità:** adattamento rapido ai cambiamenti dei requisiti aziendali (stesso meccanismo alla base del Cloud: è estremamente semplice disporre di risorse temporanee).
- ▶ **Manutenzione semplificata:** i servizi sono sviluppati in modo indipendente e possono essere aggiornati singolarmente.
- ▶ **Integrazione:** facile connessione tra sistemi eterogenei.
- ▶ **Riutilizzo:** riduzione dei costi di sviluppo grazie alla riutilizzabilità dei servizi.

Sfide delle SOA

- ▶ **Complessità:** progettazione e implementazione richiedono competenze specifiche.
- ▶ **Sicurezza:** gestione dei dati sensibili tra servizi distribuiti.
- ▶ **Prestazioni:** latenza introdotta dalla comunicazione tra servizi.
- ▶ **Governance:** necessità di gestire versioni, dipendenze e ciclo di vita dei servizi.

Esempio di applicazione delle SOA

- ▶ **Settore bancario:**

- ▶ Servizio per la gestione dei conti correnti.
- ▶ Servizio per il calcolo dei tassi di interesse.
- ▶ Servizio per l'elaborazione dei pagamenti.

- ▶ I servizi possono essere utilizzati da diverse applicazioni (app mobile, sito web, sportelli ATM).
- ▶ Ciò è possibile perché la banca in questione adotta un'architettura orientata ai servizi (altrimenti, ogni applicazione dovrebbe implementare in modo inutilmente ripetitivo le stesse funzionalità).

Alcune differenze tra SOA e Microservices

L'architettura a Microservices riguarda più che altro le scelte di sviluppo che un team può adottare nell'implementazione di una singola applicazione.

- ▶ **Granularità dei servizi:**

- ▶ SOA: servizi di dimensioni maggiori, spesso orientati ai processi aziendali.
- ▶ Microservices: servizi più piccoli e focalizzati su singole funzionalità.

- ▶ **Tecnologia di comunicazione:**

- ▶ SOA: utilizza un ESB per la comunicazione.
- ▶ Microservices: comunicazione leggera tramite API RESTful.

Conclusioni

- ▶ Le SOA sono da pensare come soluzioni adatte a sistemi di grande complessità, altrimenti difficili da gestire.
- ▶ Le SOA rendono possibile la creazione di applicazioni distribuite integrando servizi sviluppati autonomamente.
- ▶ Nonostante le sfide, sono ampiamente utilizzate in settori come finanza, sanità e commercio elettronico e molte aziende si sono riconvertite interamente a questo modello (es. AirBnb).

- ▶ Architetture orientate ai servizi, <https://www.redhat.com/it/topics/cloud-native-apps/what-is-service-oriented-architecture>