



Luca Cabibbo
Architettura
dei Sistemi
Software

Architettura orientata ai servizi

dispensa asw530

ottobre 2023

*I have not failed.
I've just found 10,000 ways that won't work.*

Thomas Edison



- Riferimenti

- ❑ Luca Cabibbo. **Architettura del Software: Strutture e Qualità**. Edizioni Efestò, 2021.
 - Capitolo 31, **Architettura orientata ai servizi**
- ❑ Erl, T. **SOA: Principles of Service Design**. Prentice Hall, 2008.
- ❑ Bell, M. **Service-Oriented Modeling: Service Analysis, Design, and Architecture**. Wiley, 2008.
- ❑ Papazoglou, M.P. **Web Services: Principles and Technology**. Pearson, 2008.
- ❑ Manes, A.T. **SOA is Dead; Long Live Services**. Application Platform Strategies Blog, Burton Group. 2009.



- Obiettivi e argomenti

□ Obiettivi

- presentare e discutere l'architettura orientata ai servizi (SOA)

□ Argomenti

- introduzione
- architettura orientata ai servizi
- architettura orientata ai servizi e obiettivi di business
- Enterprise Service Bus
- discussione



* Introduzione

- L'architettura a servizi ha, in generale, l'obiettivo di sostenere la costruzione di sistemi software in grado di soddisfare gli obiettivi di business, correnti e futuri, delle organizzazioni
 - esistono diversi tipi specifici di "architetture a servizi"
 - l'*architettura orientata ai servizi (Service-Oriented Architecture o SOA)* è un importante caso specifico di "architettura a servizi"



Introduzione alla SOA

- SOA è uno stile architeturale (sorto alla fine degli anni '90) che persegue diversi obiettivi di qualità
 - qualità tecnologiche
 - interoperabilità, sicurezza, affidabilità, disponibilità, ...
 - qualità di business
 - per sostenere gli obiettivi di business, correnti e futuri, delle organizzazioni
 - per consentire a un'organizzazione di offrire i propri processi di business (processi aziendali) in modo più efficiente, più efficace e più flessibile

- ci concentriamo soprattutto sugli aspetti di business



* Architettura orientata ai servizi

- L'architettura orientata ai servizi è basata sulla nozione di servizio
 - un **servizio** incapsula una funzionalità di business, per consentirne l'accesso come servizio software
 - un servizio può essere usato per costruire più applicazioni e processi di business
 - l'**architettura orientata ai servizi** (**Service-Oriented Architecture** o **SOA**) è uno stile architeturale per la costruzione di una molteplicità di sistemi o applicazioni, sulla base della composizione di un insieme di servizi
 - e non solo per costruire un singolo sistema



Architettura a componenti e SOA

- Confrontiamo l'architettura a componenti e la SOA
 - architettura a componenti
 - un componente implementa delle funzionalità
 - i componenti sono tutti realizzati con un'unica tecnologia
 - i componenti sono progettati per realizzare un'applicazione (per un'organizzazione) mediante una composizione (statica)
 - la portata è una singola applicazione software
 - sostiene le qualità tecnologiche del software



Architettura a componenti e SOA

- Confrontiamo l'architettura a componenti e la SOA
 - architettura orientata ai servizi
 - un servizio implementa delle funzionalità di business
 - i servizi sono interoperabili e possono essere composti in modo flessibile
 - i servizi sono progettati e usati per realizzare più applicazioni (anche per più organizzazioni) come composizione (dinamica e flessibile)
 - la portata sono tutte le applicazioni e i sistemi software di un'organizzazione
 - sostiene gli obiettivi di business di un'organizzazione



- Approccio della SOA

- La SOA è basata sul riuso e la composizione flessibile del patrimonio software esistente di un'organizzazione
 - si inizia con la decomposizione delle funzionalità di business dell'organizzazione, già implementate nei suoi sistemi software
 - queste funzionalità vengono incapsulate e offerte come servizi – interoperabili e componibili
 - è anche possibile consumare i servizi di altre organizzazioni – e offrire i propri servizi ad altre organizzazioni
 - ogni servizio può essere usato (e riusato) in più modi diversi, e combinato (e ricombinato) in modo flessibile con altri servizi, per realizzare una molteplicità di funzioni di business
 - questo sostiene agilità, innovazione, efficienza ed efficacia delle organizzazioni



Approccio della SOA

- Alcune idee alla base della SOA
 - per perseguire i propri obiettivi di business, un'organizzazione eroga diversi processi di business (business process o BP)
 - ogni BP consiste di un insieme di attività o compiti (servizi di business), da svolgere in modo coordinato
 - la SOA ambisce a sostenere i BP di un'organizzazione (con i loro obiettivi), insieme ad una loro evoluzione flessibile
 - la SOA persegue un allineamento tra l'organizzazione del software e la strutturazione dei BP e dei servizi di business dell'organizzazione – in modo che possano cambiare insieme
 - infatti, il business di un'organizzazione può essere flessibile solo quanto la sua infrastruttura tecnologica (IT)



Approccio della SOA

□ Alcune idee alla base della SOA

- nei sistemi software tradizionali, ogni BP è realizzato mediante una o più applicazioni separate
 - le attività e i compiti di quel processo hanno spesso implementazioni separate – che non vengono riusate in più BP o applicazioni
 - i BP sono difficili da implementare, da gestire e da far evolvere
- nella SOA, queste attività e compiti vengono invece rappresentati mediante dei servizi software, che possono essere utilizzati in più BP e applicazioni
 - questi servizi vengono spesso realizzati esponendo delle funzionalità già esistenti mediante un'interfaccia a servizi
 - è più semplice far evolvere i BP esistenti e definire nuovi BP



Approccio della SOA

□ Alcune idee alla base della SOA

- ogni servizio viene progettato per poter essere utilizzato in più applicazioni e BP
 - ogni applicazione o BP che ha bisogno di quella funzionalità, può invocare il servizio corrispondente
- per sostenere questa possibilità di riuso, i servizi devono essere progettati per essere debolmente accoppiati, interoperabili e componibili
 - ovvero, devono soddisfare i principi di progettazione generali per i servizi che abbiamo discusso in un capitolo precedente

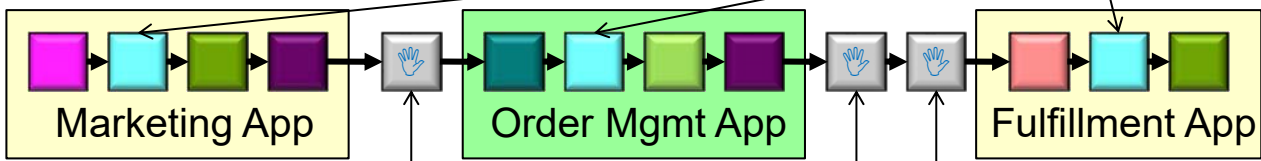


- Un esempio – punto di partenza



□ Un processo di business tradizionale

funzioni ripetute
in più applicazioni



una funzione di business

compiti eseguiti manualmente

- processo realizzato in più applicazioni separate
- funzioni di business accoppiate alle applicazioni
- alcune funzioni sono replicate in più applicazioni – con interfacce proprietarie, il riuso è difficile
- i passi manuali complicano la situazione
- difficile monitorare il processo di business
- difficile cambiare il processo di business



Un esempio

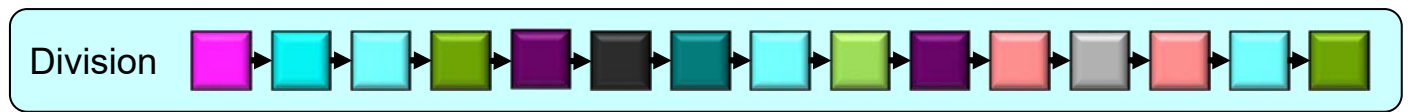


□ Questo esempio mostrerà come la SOA

- consente il riuso dei sistemi informatici esistenti
- fornisce la possibilità a più sistemi informatici di lavorare insieme
 - tecnologicamente – mediante un insieme di standard per l'interoperabilità
- sostiene flessibilità nel cambiamento/evoluzione dei processi di business
- sostiene in particolare un allineamento tra business e tecnologia, consentendo all'uno di cambiare insieme all'altro



Il (nuovo) punto di partenza



- il processo di business viene (ri)definito come composizione di un insieme di servizi
 - le funzionalità individuali del processo esistente – di solito realizzate mediante un'implementazione a componenti – vengono incapsulate e offerte come servizi
 - poi il processo di business viene (ri)definito come composizione di questi servizi

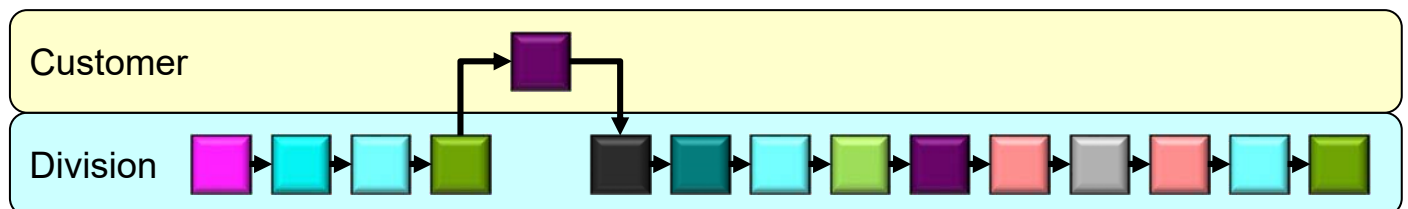
15

Architettura orientata ai servizi

Luca Cabibbo ASW



Interazione diretta con il cliente



- il processo può essere poi migliorato in molti modi
 - ad es., si può consentire al cliente di acquistare direttamente dal web
 - un partner commerciale (cliente) può fare ordini B2B mediante un servizio web
 - i diversi clienti sono serviti meglio

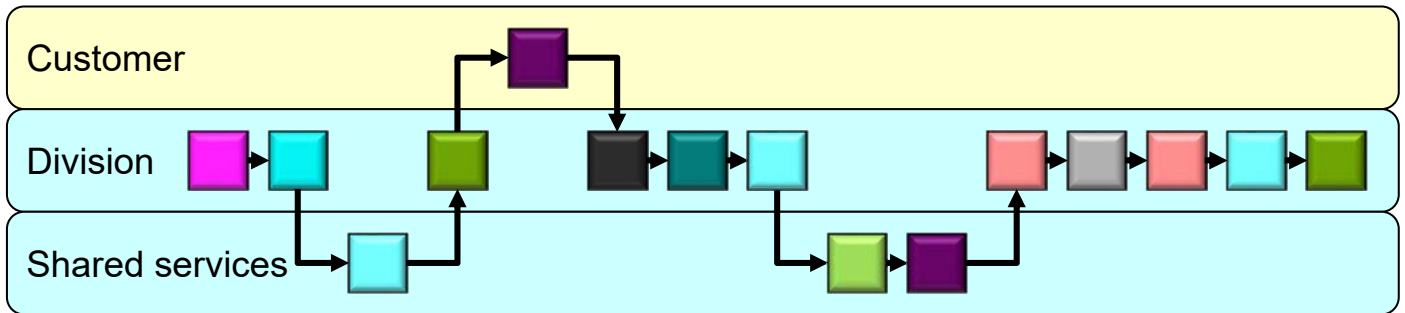
16

Architettura orientata ai servizi

Luca Cabibbo ASW



Condivisione di servizi



- i servizi di business comuni possono essere condivisi dall'intera organizzazione – e consolidati
- riduzione delle ridondanze – dei relativi costi di sviluppo e di gestione – possibilità di ottenere economie di scala

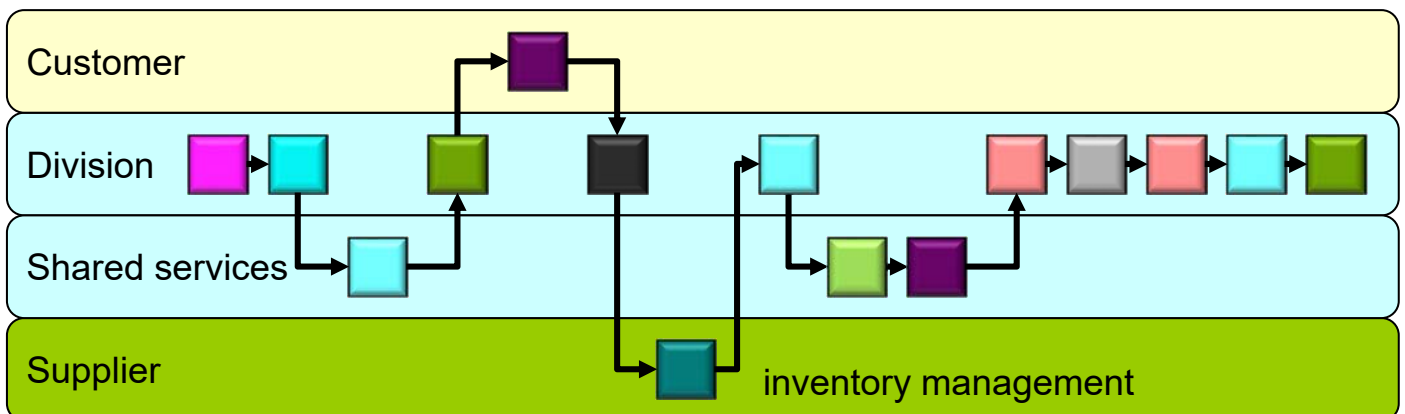
17

Architettura orientata ai servizi

Luca Cabibbo ASW



Inventario gestito dal fornitore



- possibilità di delegare funzionalità a partner commerciali (fornitori) mediante interazioni B2B
- riduzione dei costi e servizio migliore

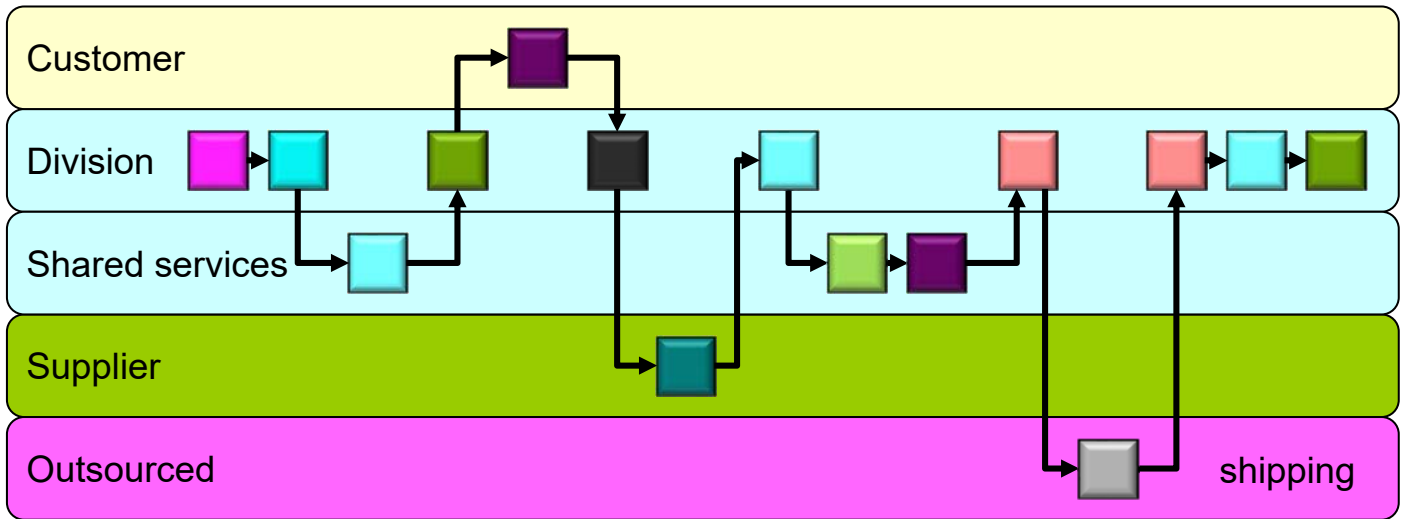
18

Architettura orientata ai servizi

Luca Cabibbo ASW



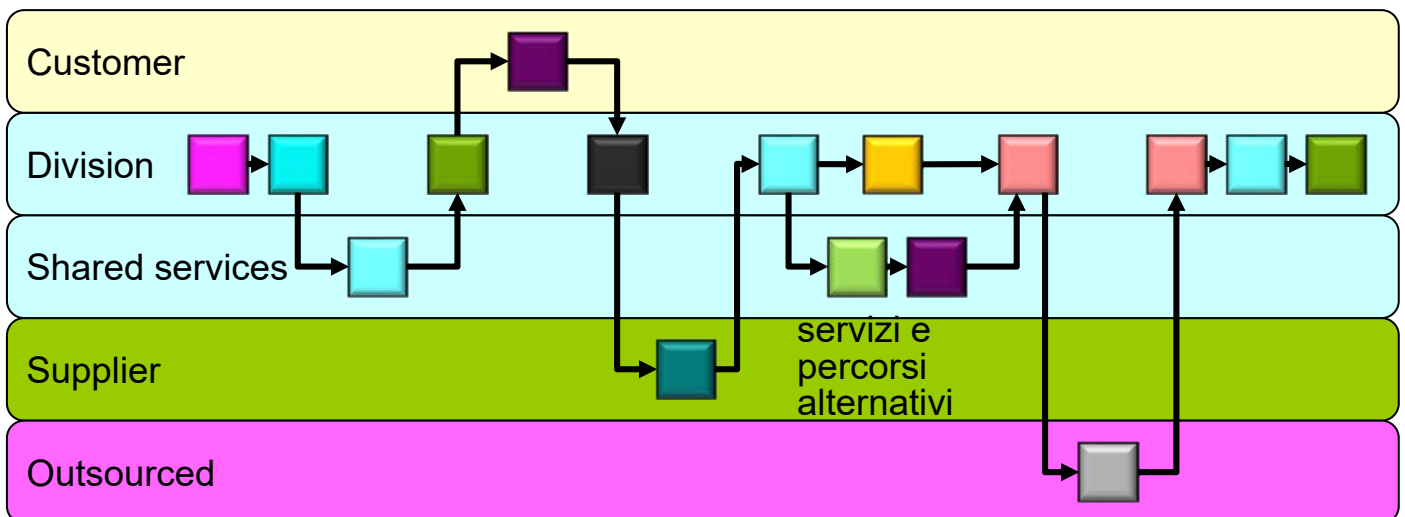
Outsourcing



- possibilità di dare in outsourcing funzionalità relative a competenze non fondamentali
- riduzione di costi e delle infrastrutture di spedizione



Miglioramento del processo



- possibilità di modificare il processo e le regole di gestione dello stesso
- la definizione e la ridefinizione di un processo è vista più come un'attività di **assemblaggio** di servizi/compiti che non come un'attività di **sviluppo**

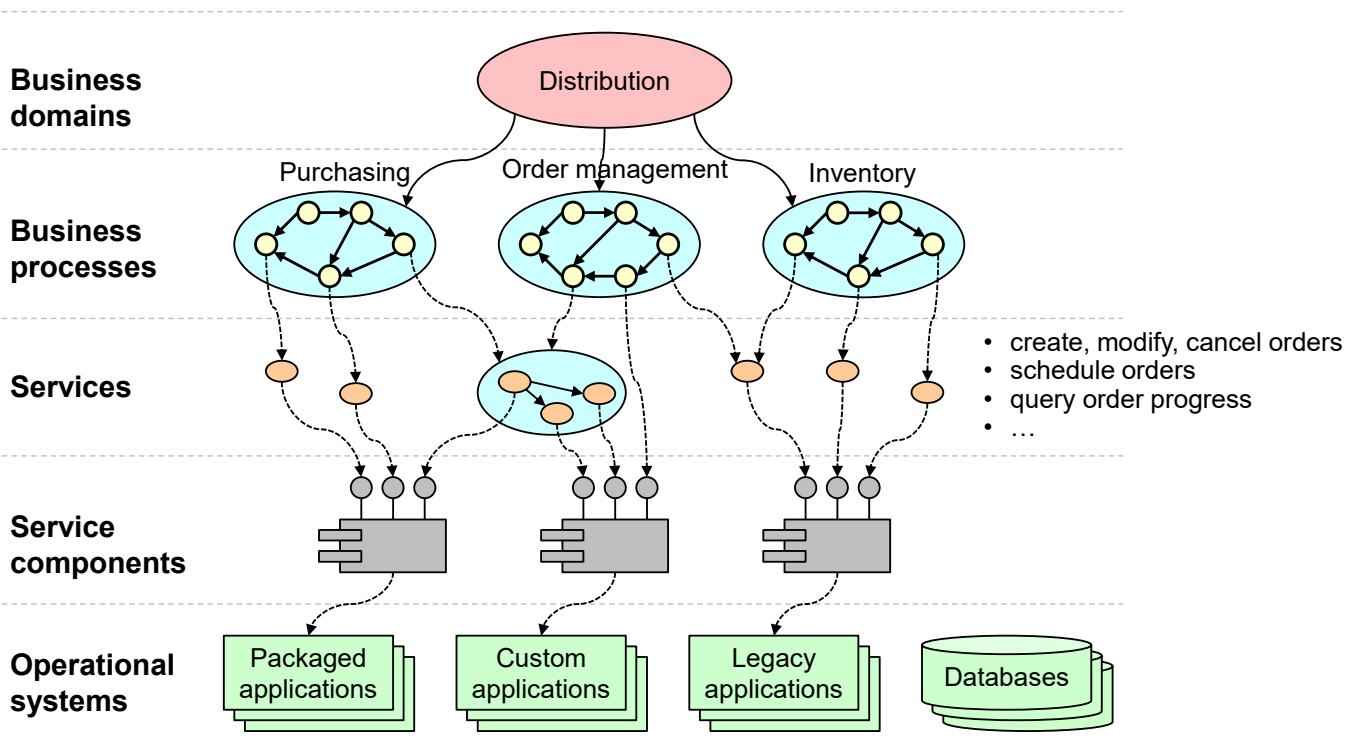


- Architettura a strati della SOA

- La SOA è organizzata complessivamente come un'architettura a strati, centrata attorno a due strati principali
 - **servizi di business (business service)**
 - ogni servizio implementa una funzionalità di business discreta
 - applicazioni o **processi di business (business process)**
 - ogni BP viene definito come composizione di un insieme di servizi di business condivisi
- questo abilita una definizione e una modifica "rapida" dei BP
 - la composizione viene di solito effettuata come un'attività di **assemblaggio**, e solo raramente come un'attività di **sviluppo**



Architettura a strati della SOA





Architettura a strati della SOA

- Organizzazione a strati della SOA
 - i **domini di business** collaborano per perseguire gli obiettivi di business di alto livello dell'organizzazione
 - i **processi di business** rappresentano i processi aziendali dell'organizzazione, nell'ambito dei rispettivi domini di business
 - i **servizi di business** rappresentano le attività e i compiti di business dell'organizzazione, che supportano uno o più processi di business dell'organizzazione
 - i **componenti** realizzano le funzionalità richieste, che vengono poi incapsulate ed esposte come servizi
 - i **sistemi operazionali** sono i sistemi legacy dell'organizzazione, le cui funzionalità vengono accedute dai componenti



Architettura a strati della SOA

- Organizzazione a strati della SOA
 - esempi di **domini di business** di un'organizzazione
 - produzione, distribuzione, marketing e risorse umane
 - esempi di **processi di business** – nel contesto del dominio della produzione
 - acquisizione di prodotti, gestione di ordini e inventario
 - esempi di **servizi di business** – nel contesto del processo di business della gestione di ordini
 - creazione, modifica e cancellazione di ordini, gestione di pagamenti e spedizioni, conoscere lo stato di un ordine e analisi degli ordini



Architettura a strati della SOA

□ Conseguenze della SOA

- abilita il riuso di funzionalità di business già implementate da sistemi legacy
- che vengono esposte come servizi, affinché possano essere condivise e consumate in modo uniforme
- per essere integrate per comporre dei processi di business in modo flessibile
- per supportare gli obiettivi di business dell'organizzazione
- in questo, rivestono un ruolo fondamentale
 - l'interoperabilità e la componibilità dei servizi
 - la soddisfazione dei principi di progettazione generali per i servizi



* SOA e obiettivi di business

- L'architettura orientata ai servizi enfatizza gli aspetti e gli **obiettivi di business, correnti e futuri**, delle organizzazioni
 - questa sezione discute prima gli obiettivi di business della SOA, e poi come essi vengono perseguiti



Contesto

- Il contesto per le grandi organizzazioni
 - alta competitività e alta incertezza
 - integrazione globale, di informazioni e processi
 - per sopravvivere, è richiesta **agilità di business** – la capacità di
 - offrire i propri servizi e processi di business in modo più efficiente ed efficace
 - offrire processi di business innovativi, in nuovi mercati
 - più in generale, poter cambiare rapidamente e in modo flessibile i propri processi di business
 - il business di un'organizzazione è spesso flessibile solo quanto i suoi sistemi software
 - il software dovrebbe dunque sostenere l'agilità di business delle organizzazioni – e non ostacolarla



Approccio all'agilità di business

- L'approccio della SOA per sostenere l'agilità di business
 - allineare l'organizzazione del business con l'organizzazione del software, in modo che possano variare insieme
 - sostenere lo sviluppo e l'evoluzione agile dei processi di business dell'organizzazione
 - i processi di business sono realizzati come applicazioni composte
 - sulla base del riutilizzo di servizi software già esistenti all'interno dell'organizzazione
 - con la possibilità di fruire anche di servizi software esterni all'organizzazione



Obiettivi di business della SOA

- In modo più specifico, ecco i principali obiettivi di business affrontati dalla SOA
 - abilitare un'integrazione semplice e flessibile con i propri sistemi legacy
 - ottimizzare i propri processi di business, per aumentarne efficienza ed efficacia e ridurre i costi operativi
 - sostenere l'agilità di business, per gestire rapidamente il cambiamento dei processi di business



Dai principi agli obiettivi di business

- Per raggiungere questi obiettivi di business, è innanzitutto necessario che i servizi che costituiscono la SOA soddisfino i principi per la progettazione dei servizi di [Erl 2008]
 - i servizi condividono un contratto formale
 - i servizi realizzano un'astrazione della logica sottostante
 - i servizi sono debolmente accoppiati
 - i servizi sono riusabili
 - i servizi sono componibili
 - i servizi sono autonomi
 - i servizi sono stateless
 - i servizi possono essere scoperti
 - i servizi hanno un'interfaccia accessibile in rete
 - la locazione dei servizi è trasparente



- Discutiamo ora come l'architettura orientata ai servizi sostiene gli obiettivi di business appena presentati
 - agilità – capacità di offrire servizi innovativi e di adattarsi a opportunità e minacce competitive
 - ottimizzazione dei processi
 - integrazione con i sistemi legacy
 - l'ipotesi è che i servizi soddisfino i principi generali per la progettazione dei servizi



Agilità



- Agilità – essere capaci di offrire servizi innovativi e di adattarsi rapidamente a nuove opportunità e potenziali minacce in un contesto competitivo
 - l'agilità è sostenuta soprattutto dalla possibilità di comporre servizi, e di poterli combinare in modi nuovi e diversi
 - insieme al riutilizzo dei propri servizi interni, l'uso di servizi esterni e la possibilità di aggiungere nuovi servizi
 - è importante anche l'uso di servizi standardizzati, la conformità a standard e la definizione di servizi a grana (abbastanza) grossa
 - ad es., quando si presenta una nuova opportunità di business, gli sviluppatori possono rapidamente implementare un processo di business innovativo – assemblando una nuova applicazione dai servizi disponibili, e aggiungendo se richiesto dei nuovi servizi



Ottimizzazione dei processi



- Ottimizzare i propri processi di business – per aumentare la loro efficienza ed efficacia, e ridurre i costi operativi
 - l'ottimizzazione dei processi di business è un fattore critico di successo del business di un'organizzazione
 - questo obiettivo è sostenuto soprattutto dalla possibilità di comporre servizi e di usare sia servizi interni che servizi esterni
 - ad es., ogni singola funzionalità applicativa può essere offerta come singolo servizio nell'ambito dell'organizzazione, e poi consumato da tutte le applicazioni, indipendentemente dalle tecnologie con cui esse sono realizzate – è anche possibile vendere questi servizi ad altri – perseguendo economie di scala e riducendo i costi di gestione della diverse funzionalità



Integrazione con i sistemi legacy



- Abilitare un'integrazione semplice e flessibile con i propri sistemi legacy
 - molti sistemi legacy implementano un insieme ricco di funzionalità – che possono essere riusate come servizi in una SOA
 - se questo viene fatto usando tecniche e strumenti non invasivi, allora, allo stesso tempo
 - i sistemi legacy possono continuare a offrire le loro funzionalità
 - queste funzionalità possono essere fruite anche da altri consumatori di servizi
 - questa è in genere un'alternativa più economica che non reimplementare le funzionalità dei sistema legacy



* Enterprise Service Bus



- La SOA richiede un'opportuna infrastruttura runtime per implementare o gestire gli standard tecnologici che si vogliono adottare per sostenere interoperabilità e componibilità
 - l'infrastruttura minimale per i servizi web può essere fornita da un application server (AS) – che può normalmente fungere anche da contenitore di servizi web
 - è però necessaria anche un'infrastruttura per collegare servizi in esecuzione in AS diversi e realizzati con tecnologie eterogenee – per realizzare una federazione di application server
 - è necessaria un'infrastruttura di deployment complessa – che di solito viene chiamata un *Enterprise Service Bus*



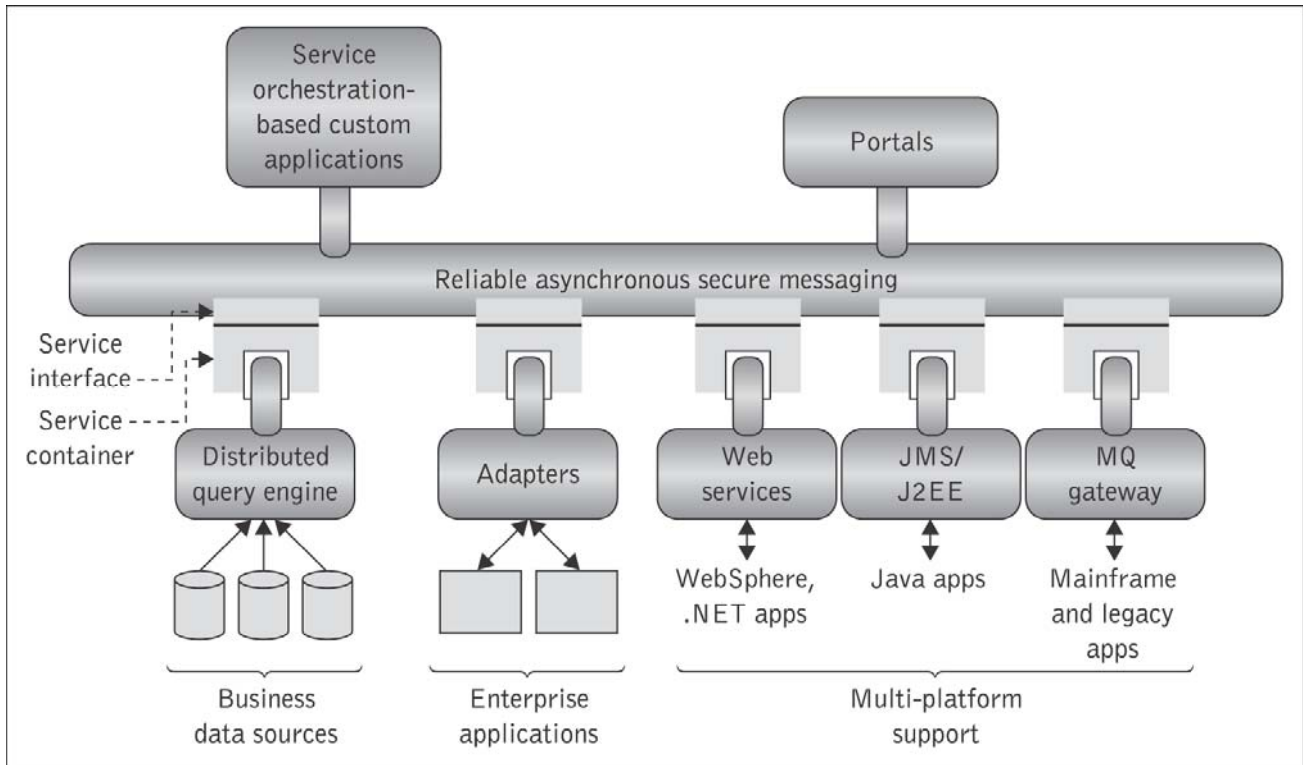
Enterprise Service Bus



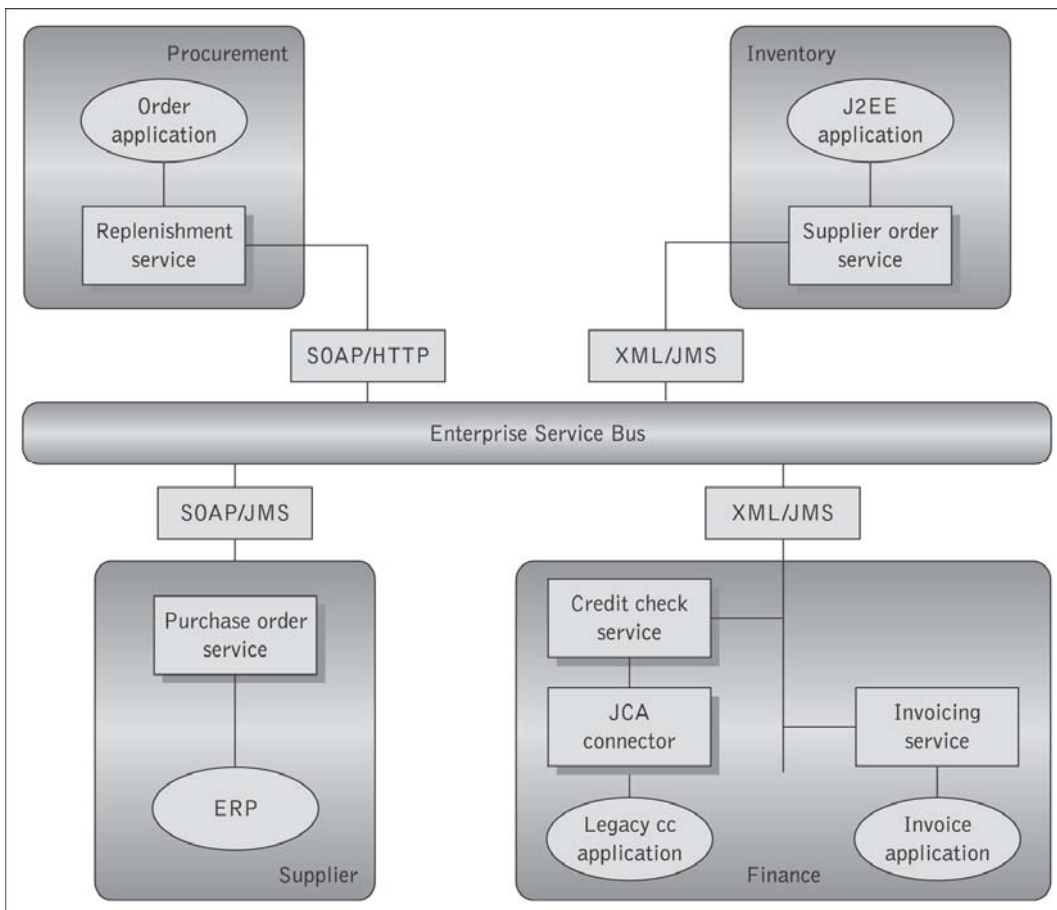
- Un *Enterprise Service Bus (ESB)*
 - ha l'obiettivo di connettere un insieme di sistemi software separati ed eterogenei – per gestire in modo coerente le interazioni e lo scambio di messaggi tra fornitori e consumatori di servizi
 - si occupa di
 - convertire messaggi tra protocolli e tecnologie differenti
 - trasformare i dati contenuti nei messaggi
 - gestire sicurezza, transazioni e affidabilità – con riferimento agli standard utilizzati nello scambio di messaggi
 - supportare la scoperta di servizi
 - gestire la composizione e l'orchestrazione di servizi



ESB – connessione di diverse tecnologie



ESB – connessione di servizi remoti





* Discussione

□ Per riassumere

- i **servizi** sono la risorsa architeturale primaria della SOA
- l'**architettura orientata ai servizi** è uno stile architeturale
 - per organizzare un insieme di servizi distribuiti – controllati da più organizzazioni – e consentirne una fruizione flessibile
 - per costruire una molteplicità di sistemi e applicazioni sulla base della composizione flessibile di questi servizi
 - per consentire alle organizzazioni di sviluppare, interconnettere e mantenere applicazioni e servizi di tipo enterprise in modo efficiente ed economico
 - per far sì che un'organizzazione possa strutturare le informazioni e i processi di business in modo da sostenere i propri obiettivi di business, in un contesto competitivo



- Critica alla SOA

- **SOA is Dead; Long Live Services** (un articolo del 2009) ha presentato questa opinione (motivata anche dalla recessione economica di quegli anni): “la SOA è morta”
 - alcune organizzazioni sono riuscite ad applicare la SOA con successo
 - tuttavia, per molte organizzazioni, il tentativo di applicare la SOA è stato un esperimento fallimentare – non ha portato a nessun beneficio reale
 - il problema però non è nei servizi o nella SOA in sé, ma nel modo in cui la SOA è stata applicata
 - dunque, la SOA è morta
 - i servizi sono però una buona idea – magari da utilizzare nel contesto di altri stili di “architetture a servizi”



Critica alla SOA

- Alcune osservazioni importanti da questo articolo
 - Successful SOA (i.e., application re-architecture) requires disruption to the status quo.
 - SOA is not simply a matter of deploying new technology and building service interfaces to existing applications; it requires redesign of the application portfolio. And it requires a massive shift in the way IT operates.
 - The small select group of organizations that has seen spectacular gains from SOA did so by treating it as an agent of transformation. In each of these success stories, SOA was just one aspect of the transformation effort.
 - And here's the secret to success: SOA needs to be part of something bigger. If it isn't, then you need to ask yourself why you've been doing it.



Critica alla SOA

- Alcune conclusioni
 - l'utilizzo di una tecnologia a servizi, da sola, non garantisce che il risultato sia una (buona) architettura orientata ai servizi
 - lo sforzo richiesto per la realizzazione di una buona SOA è molto alto
 - perché richiede un ripensamento complessivo non solo del software, ma anche dell'intera organizzazione
 - non c'è consenso su come applicare *bene* lo stile SOA
 - la SOA può fornire agilità – tuttavia, questo non significa che è uno stile architettonico adatto a team di sviluppo agili
 - gli obiettivi della SOA sono molto ambiziosi
 - ma un'organizzazione richiede davvero un'architettura orientata ai servizi "completa"?