

Service-Oriented Architecture

Dispensa ASW 455
ottobre 2014

*Un sistema ha successo
quando si trova all'intersezione naturale
tra tecnologia, politica ed economia.*

A.D. Wheelon



- Fonti

- [Papazoglou] Papazoglou, Web Services – Principles and Technology, 2008
- [POSA4] Pattern-Oriented Software Architecture – A Pattern Language for Distributed Computing, 2007
- [SAP] Chapter 13, Architectural Tactics and Patterns
- [Marks&Bell] Service-Oriented Architecture – a planning and implementation guide for business and technology, 2006
- [Erl 2008] SOA – Principles of Service Design, 2008
- [SEI 2005] O'Brien, Bass, Merson, Quality Attributes and Service-Oriented Architectures, Technical note CMU/SEI-2005-TN-014
- [SEI 2007] Bianco, Kotermanski, Merson, Evaluating a Service-Oriented Architecture, Technical report CMU/SEI-2007-TR-015
- [SEI 2011] Bianco, Lewis, Merson, Simanta, Architecting Service-Oriented Systems, Technical note CMU/SEI-2011-TN-008



* Introduzione

- La tecnologia (middleware) a **servizi**
 - è una tecnologia per l'integrazione di applicazioni distribuite – volta a risolvere problemi pragmatici di interoperabilità, basata su standard accettati dalla maggior parte dei produttori di software
 - i Web Services rappresentano la tecnologia “dominante” in questa arena

- L'**architettura orientata ai servizi (SOA)**
 - fornisce il contesto metodologico (e di business) in cui utilizzare al meglio le tecnologie basate su servizi



Servizi e architettura orientate ai servizi

- L'architettura orientata ai servizi è basata sui servizi come costruito (tipo di componente software) per sostenere sviluppo e composizione di applicazioni distribuite
 - un **servizio** ha l'obiettivo di incapsulare una ben precisa funzionalità di business (logica applicativa), per renderla disponibile e accessibile come servizio software da parte di client software sul web – ciascun servizio può essere usato per costruire diverse applicazioni e processi di business
 - **SOA (Service-Oriented Architecture, ovvero architettura orientata ai servizi)** è uno stile architetturale per la costruzione di una molteplicità di sistemi o applicazioni sulla base della composizione di un insieme di servizi – e non semplicemente per la costruzione di un singolo sistema come composizione di un insieme di servizi



Relazione tra WS e SOA

- Alcuni confondono WS e SOA – tuttavia, i due concetti sono ben distinti
 - i WS sono una *tecnologia per l'implementazione* di servizi
 - basata su standard specifici
 - una tecnologia abilitante alla realizzazione di soluzioni SOA
 - una tecnologia è necessaria per realizzare una SOA – ma non è sufficiente
 - SOA è un *approccio per progettare* sistemi
 - con lo scopo specifico di favorire la condivisione e l'integrazione di servizi
 - l'adozione di una SOA è resa più semplice dalla tecnologia dei WS



Web Services (in sintesi)

- Un *Web Service*
 - è un modulo o componente software, auto-contenuto e auto-descrittivo, accessibile mediante Internet, in modo indipendente dalla piattaforma
 - rappresenta un servizio, ovvero ha lo scopo di svolgere un compito, risolvere un problema, o condurre transazioni per conto di un utente o applicazione
- I Web Services
 - possono essere messi in corrispondenza e composti – favorendo l'integrazione di servizi, per creare processi di business completi, con un costo di sviluppo ridotto
 - questa possibilità è basata sulle capacità di descrivere servizi (WSDL), scoprire servizi (UDDI), invocare servizi (SOAP), comporre servizi (BPEL), definire livelli di qualità dei servizi (WS-*)



SOA e qualità

- SOA è uno stile architeturale che (come ogni altro stile) si propone di perseguire un certo numero di obiettivi di qualità
 - SOA (come altri stili architeturali) sostiene attributi di qualità “tecnologica”
 - in primo luogo l’interoperabilità – ma anche sicurezza, affidabilità, disponibilità, ...
 - inoltre, SOA (diversamente da altri stili architeturali) cerca anche di sostenere obiettivi di “business”, correnti e futuri, delle organizzazioni
 - agilità di business – integrazione dei processi di business all’interno dell’organizzazione – integrazione dei processi di business con partner, fornitori e clienti – possibilità di monitorare/governare l’efficacia dei miglioramenti nei processi di business – ...
 - discutiamo questi due aspetti separatamente



* SOA come stile architeturale [SAP]

- L’architettura orientata ai servizi è uno stile architeturale per la costruzione di sistemi o applicazioni sulla base della composizione di un insieme di servizi
 - in effetti, diversi autori hanno descritto le SOA come stile architeturale
 - tuttavia, autori diversi hanno posto l’attenzione su aspetti differenti delle SOA – giungendo così a descrizioni differenti
 - viene ora illustrata la descrizione dello stile architeturale SOA secondo [SAP] – che enfatizza il sostegno agli attributi di qualità “tecnologica”
 - più avanti discuteremo il sostegno agli attributi di qualità di “business”



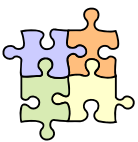
SOA come stile architetturale [SAP]

□ Contesto

- ci sono diversi servizi – offerti da fornitori di servizi e consumati da consumatori di servizi
- i consumatori di servizi desiderano comprendere e usare questi servizi – senza nessuna conoscenza dettagliata delle loro implementazioni

□ Problema

- si vuole sostenere l'interoperabilità di componenti e servizi distribuiti – scritti con linguaggi di programmazione diversi, in esecuzione su piattaforme differenti, forniti da diverse organizzazioni e distribuiti su Internet
- si vuole poter localizzare questi servizi – per poi combinarli (anche dinamicamente) per realizzare interazioni significative
- inoltre si vogliono ottenere livelli di prestazioni, sicurezza e affidabilità accettabili



SOA come stile architetturale [SAP]

□ Soluzione

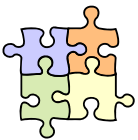
- organizza il sistema in termini di un insieme di componenti distribuiti che forniscono e/o consumano **servizi**
- i componenti fornitori di servizi e i componenti consumatori di servizi possono usare linguaggi di programmazione e piattaforme differenti
- i servizi sono in larga misura indipendenti – sono rilasciati indipendentemente, e spesso appartengono a sistemi e organizzazioni differenti
- i componenti hanno interfacce che descrivono i servizi che forniscono, o che richiedono ad altri componenti
- è anche possibile specificare attributi di qualità dei servizi – e fornire garanzie sui servizi tramite SLA
- i componenti effettuano le loro computazioni richiedendo servizi gli uni agli altri



SOA come stile architetturale [SAP]

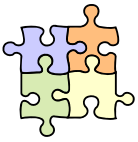
□ Alcune ulteriori osservazioni

- i componenti software che forniscono e consumano servizi possono assumere forme molto differenti tra loro – ad es., una transazione CICS in esecuzione su un mainframe oppure codice JavaScript in esecuzione su un browser web
- oltre a questi componenti, una SOA può usare altri elementi che forniscono servizi infrastrutturali – ad esempio
 - un registry dei servizi
 - un server per la composizione e l'orchestrazione di servizi
 - più in generale, un enterprise service bus, per l'invocazione di servizi
- il collegamento tra servizi può avvenire sulla base di diversi tipi di connettori
 - ad esempio, SOAP, REST oppure l'uso di messaggi asincroni



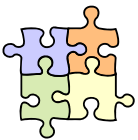
Discussione

- Questa breve presentazione dell'architettura orientata ai servizi – proposta da [SAP] – cattura *alcune* problematiche fondamentali affrontate dalle SOA
 - in particolare, consente di inquadrare soprattutto gli aspetti delle *tecnologie a servizi*
 - tuttavia, questa descrizione non coglie numerosi degli aspetti (concreti e complessi) che vanno di solito affrontati nei problemi di interoperabilità e di integrazione – come il contesto di business o le linee guida per la progettazione dei servizi e delle loro interfacce



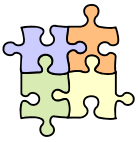
* SOA e obiettivi di business I

- Le architetture software sono il ponte tra gli obiettivi di business di un'organizzazione e i loro sistemi software
 - le tecnologie e gli stili architetture studiati finora (ad es., l'architettura a componenti) si pongono l'obiettivo di sostenere attributi di qualità "tecnologica" – prestazioni, scalabilità, sicurezza, ...
 - nell'architettura orientata ai servizi – grazie anche alla maggior maturità delle tecnologie sottostanti – l'attenzione si sposta anche e soprattutto sugli aspetti e gli obiettivi di "business", *correnti e futuri*, delle organizzazioni



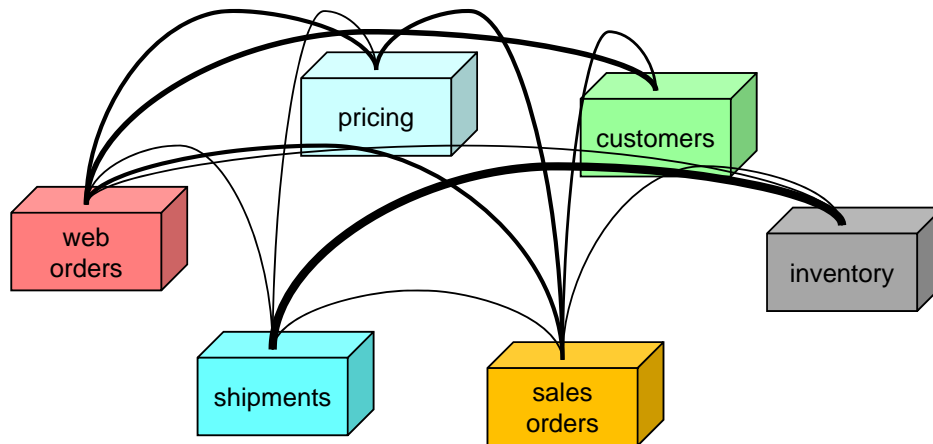
Contesto

- Il contesto in cui si muovono oggi le grandi organizzazioni
 - alta competitività e alta incertezza
 - integrazione globale – di informazioni e processi
 - le organizzazioni richiedono agilità
 - per offrire i propri servizi (in senso aziendale) in modo più efficiente/efficace
 - per offrire servizi innovativi, per offrire servizi a nuovi mercati
 - un'organizzazione deve poter cambiare rapidamente – e poter continuare a cambiare in modo flessibile – i propri processi di business (processi aziendali)
 - inoltre, il ruolo delle tecnologie informatiche (IT) è tale che il business di un'organizzazione può essere flessibile solo quanto la sua infrastruttura IT
- Lo stile SOA si propone di far sì che l'IT sostenga – anziché ostacolare – l'agilità di business delle organizzazioni

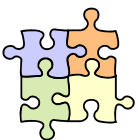


Limiti delle architetture a componenti

- Le architetture a componenti offrono numerosi vantaggi
 - ad esempio, poter definire componenti che incapsulano funzionalità di business

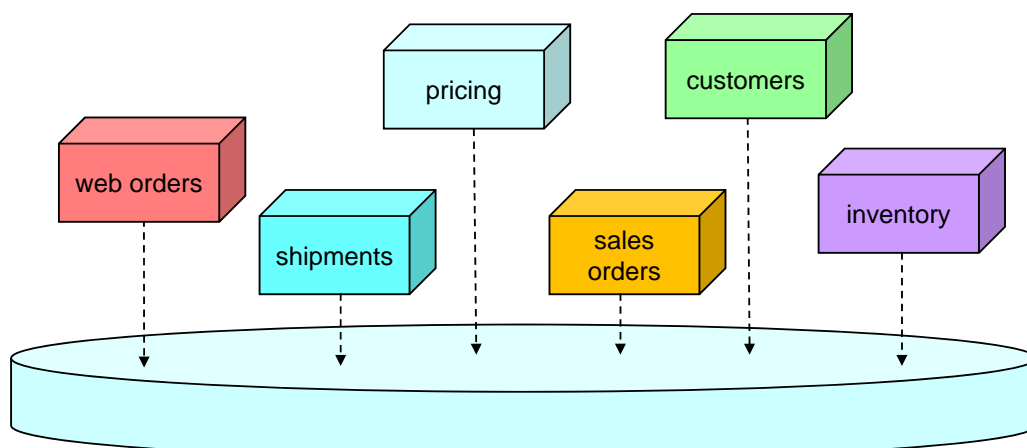


- tuttavia, nei contesti di integrazione, ciascun componente deve essere opportunamente collegato ad altri componenti, in modo *statico* – aumentando la complessità del sistema



Architettura a servizi

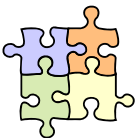
- In un'architettura a servizi
 - ciascun servizio incapsula una funzionalità di business
 - i servizi sono integrati e composti per formare applicazioni e sistemi
 - i servizi sono debolmente accoppiati – per rendere più facile, più flessibile e più agile la loro integrazione e composizione





SOA e innovazione

- L'innovazione può essere definita come il processo di effettuare un cambiamento per realizzare/fare qualcosa di nuovo
 - nei sistemi tradizionali, hardware, software e reti sono integrati in modo rigido – quindi effettuare cambiamenti è difficile
- SOA rende più semplici i cambiamenti e l'innovazione
 - in una SOA, l'IT è realizzato come un insieme di “componenti” che è facile assemblare e riconfigurare
 - ciascuno di questi “componenti” è un servizio di business – erogato dalla propria oppure anche da un'altra organizzazione
 - in una SOA, questi servizi possono essere assemblati come si vuole – sostenendo cambiamento e innovazione – risparmiando tempo e denaro



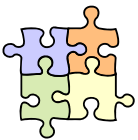
SOA è come ...

- SOA, per sostenere innovazione e cambiamento, è come ...
 - ... i mattoncini Lego www.youtube.com/watch?v=A3_QIYJRVvk
 - in una SOA, i tuoi sistemi informatici sono costruiti con componenti assemblati in modo modulare, che possono essere riconfigurati facilmente – come i mattoncini Lego
 - ciascun mattoncino rappresenta un servizio di business – come verificare il saldo di un conto corrente o il livello di inventario di un prodotto, o tracciare lo stato di consegna di una spedizione
 - una SOA è possibile comporre i tuoi sistemi informatici e processi di business come con i mattoncini Lego – è quindi possibile assemblare i tuoi servizi, in modo semplice e flessibile, per creare un processo adatto al mercato
 - se nel mercato c'è bisogno di un processo diverso, anziché iniziare da zero, tu puoi prendere gli stessi mattoncini, e riconfigurarli per fare qualcosa di diverso – risparmiando tempo e denaro



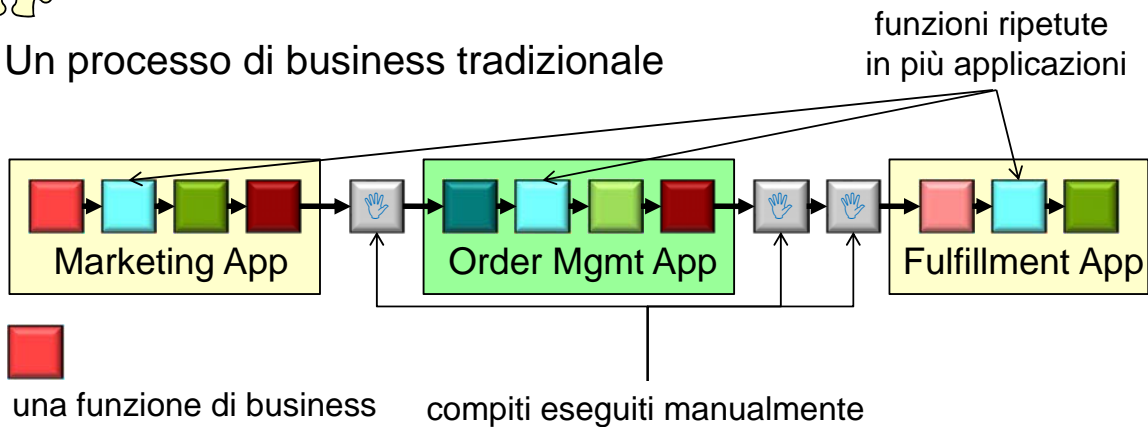
SOA è come ...

- SOA, per sostenere innovazione e cambiamento, è come ...
 - ... le note musicali
 - ciascun servizio è come una nota musicale
 - questi servizi possono essere composti in modo semplice e flessibile – così come è facile creare un nuovo motivo musicale utilizzando sempre le stesse note
 - ... il guardaroba
 - ciascun servizio è come un capo d'abbigliamento
 - questi servizi possono essere integrati in modo flessibile – così come è facile vestirsi in modo adatto a ciascuna occasione (e magari anche diverso) utilizzando i capi d'abbigliamento nel tuo guardaroba



Un esempio - punto di partenza

- Un processo di business tradizionale

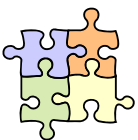


- processo realizzato in più applicazioni separate
- funzioni di business accoppiate alle applicazioni
- alcune funzioni sono replicate in più applicazioni – con interfacce proprietarie, il riuso è difficile
- i passi manuali complicano la situazione
- difficile monitorare il processo di business
- difficile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

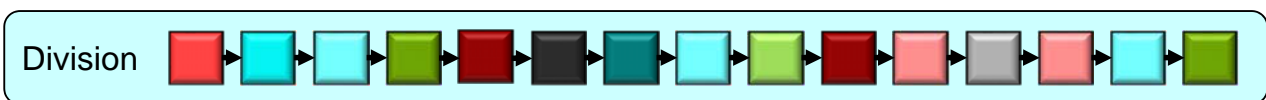


Un esempio

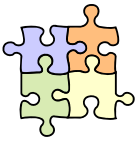
- Il seguito dell'esempio mostra come una SOA
 - consente il riutilizzo dei sistemi informatici esistenti
 - fornisce la possibilità a più sistemi informatici di lavorare insieme
 - tecnologicamente – mediante un insieme di standard per l'interoperabilità
 - sostiene la flessibilità nel cambiamento/evoluzione dei processi di business
 - sostiene in particolare un allineamento tra business e tecnologia, consentendo all'uno di cambiare insieme all'altro
 - il business di un'organizzazione può essere flessibile solo quanto la sua infrastruttura tecnologica (IT) – se l'IT non può cambiare, non può cambiare nemmeno il business



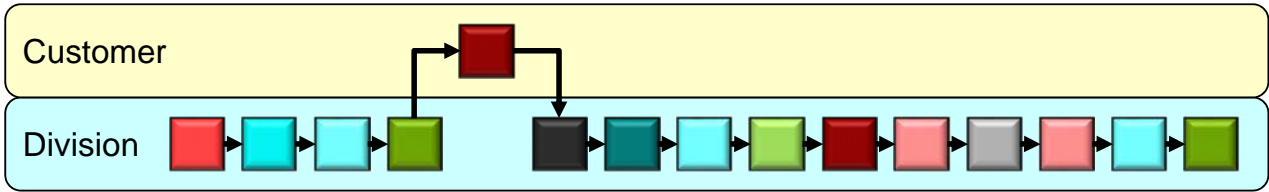
Il (nuovo) punto di partenza



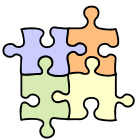
- il processo di business viene definito come composizione di un insieme di servizi
 - le funzionalità individuali del processo esistente – realizzate mediante un'implementazione a componenti – vengono incapsulate e offerte come servizi
 - poi il processo di business viene (ri)definito come composizione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



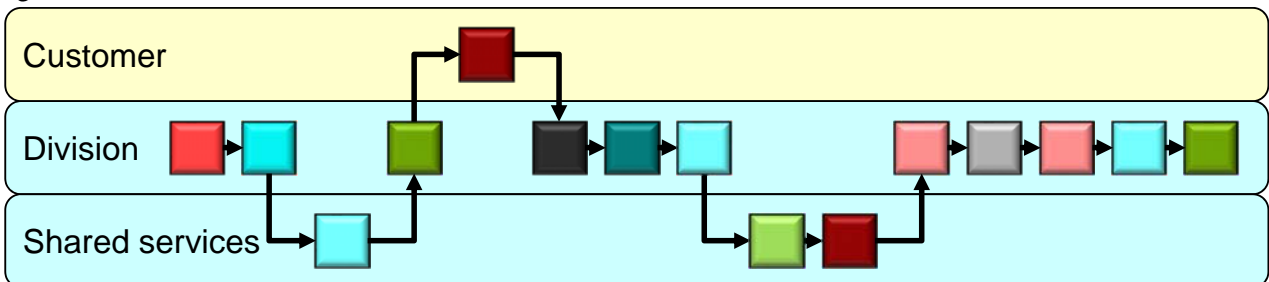
Flessibilità - interazione diretta con il cliente



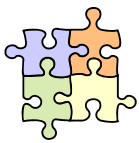
- il processo può essere poi migliorato in molti modi
 - ad esempio, si può consentire al cliente di acquistare direttamente dal web
 - un partner commerciale (cliente) può fare ordini B2B mediante un web service
 - i vari clienti sono serviti meglio



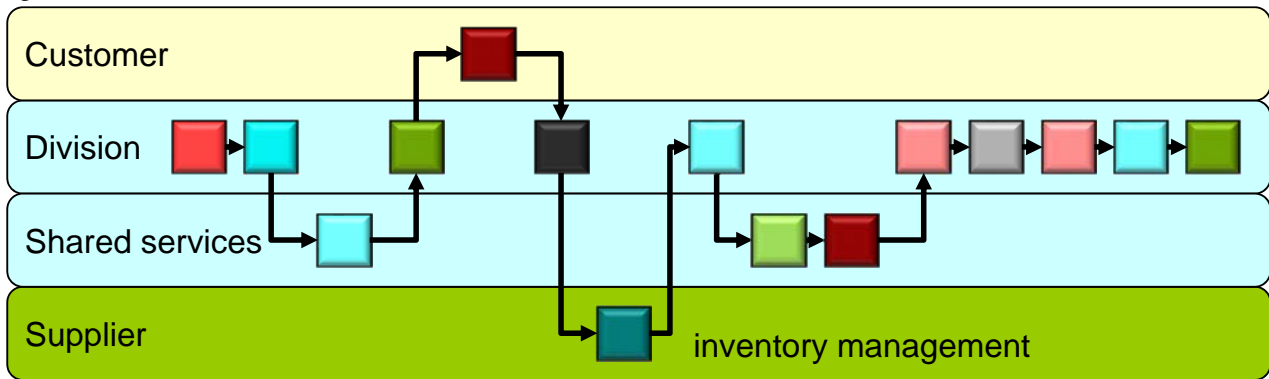
Flessibilità - condivisione di servizi



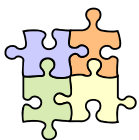
- i servizi di business comuni possono essere condivisi dall'intera organizzazione – e consolidati
- riduzione delle ridondanze – dei relativi costi di sviluppo e di gestione – possibilità di ottenere economie di scala



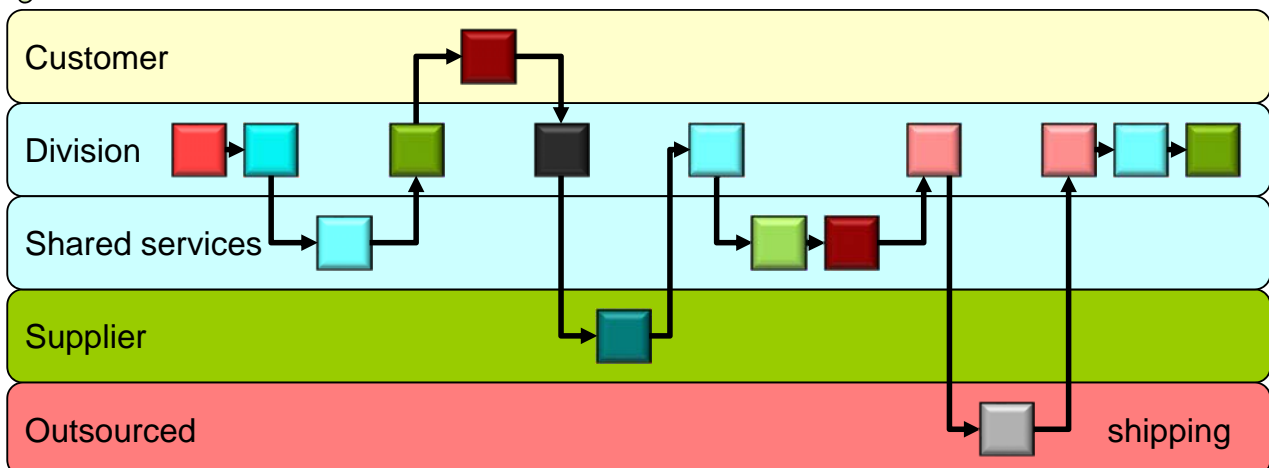
Flessibilità - inventario gestito dal fornitore



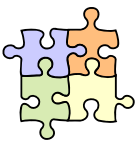
- possibilità di delegare funzionalità a partner commerciali (fornitori) mediante interazioni B2B
- riduzione dei costi e servizio migliore



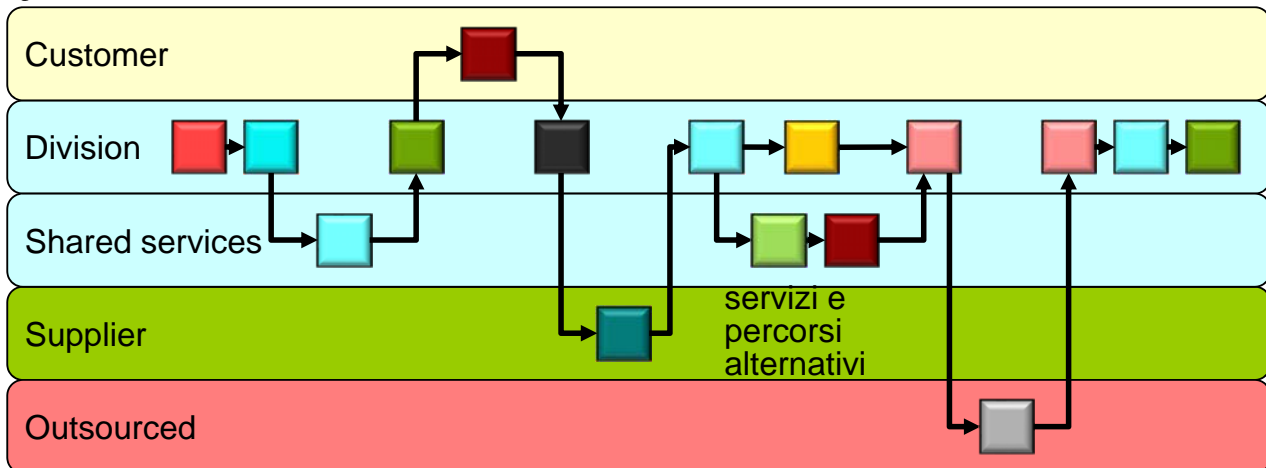
Flessibilità - outsourcing



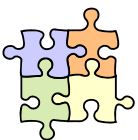
- possibilità di dare in outsourcing funzionalità relative a competenze non fondamentali
- riduzione di costi e delle infrastrutture di spedizione



Flessibilità - miglioramento del processo

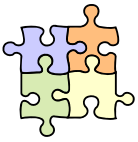


- possibilità di modificare il processo e le regole di gestione dello stesso
- la definizione e la ridefinizione di un processo è vista più come un'attività di *assemblaggio* di servizi/compiti che non come un'attività di *sviluppo*



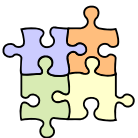
- SOA e obiettivi di business

- Ai fini del successo di un'organizzazione (o di un'azienda), è di fondamentale importanza la costruzione di sistemi informatici *che soddisfano e sostengono gli obiettivi di business, correnti e futuri, dell'organizzazione* – in questo contesto, SOA affronta i seguenti problemi e obiettivi di business
 - allineare business e IT, in modo che possano variare insieme
 - sostenere agilità di business
 - sulla base dello sviluppo agile di nuove applicazioni
 - realizzate come applicazioni composte
 - basate sul riuso di servizi software già esistenti all'interno dell'organizzazione
 - nonché sulla possibilità di fruire anche di servizi software esterni all'organizzazione



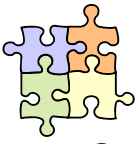
SOA e obiettivi di business

- Detto in altro modo [SEI 2005, SEI 2011], ecco i principali obiettivi di business e driver architetturali, comuni a molte organizzazioni, affrontati dallo stile SOA
 - abilitare un'integrazione semplice e flessibile con i propri sistemi legacy – *interoperabilità*
 - ottimizzare i propri processi di business, per aumentarne efficienza ed efficacia e ridurre i costi operativi – *manutenibilità, modificabilità*
 - agilità per gestire rapidamente il cambiamento dei processi di business (ad es., per offrire servizi innovativi ai clienti e adattarsi a opportunità e minacce competitive) – *estendibilità*
- Questi obiettivi possono essere raggiunti applicando un insieme di principi di progettazione per i sistemi orientati ai servizi – che saranno descritti nel seguito



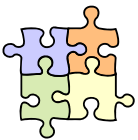
* Dai servizi alle SOA

- Come già detto, SOA è uno stile per la costruzione di sistemi o applicazioni sulla base della composizione di un insieme di servizi
 - **SOA** è un approccio architetturale per la costruzione di sistemi o applicazioni che usano un insieme di servizi
 - e non semplicemente per la costruzione di un sistema come un insieme di servizi
 - un **servizio** è un'implementazione di un ben definito pezzo di funzionalità di business, con un'interfaccia pubblicata e che può essere scoperta e che può essere usata dai consumatori del servizio nel costruire diverse applicazioni e processi di business
 - si noti che, nella definizione di una SOA, non si fa riferimento a nessuna particolare tecnologia per l'implementazione dei servizi
 - inoltre, in una SOA è fondamentale la composizione dei servizi



Dai servizi alle SOA

- Organizzazione di una SOA
 - ogni servizio implementa una funzionalità di business discreta
 - ogni applicazione che ha bisogno di eseguire quella particolare funzionalità, può usare quel servizio condiviso
 - ogni applicazione è creata assemblando e coordinando le attività tra quell'insieme appropriato di servizi che serve a realizzare un processo di business di interesse
 - il sistema orientato ai servizi di un'organizzazione comprende più applicazioni/processi di business
 - ciascun servizio può essere riusato in più applicazioni
 - i servizi sono debolmente accoppiati – tra loro e con le applicazioni/processi
 - anche la granularità dei servizi è importante
 - può essere raccomandato avere servizi a grana grossa, da utilizzare scambiando pochi messaggi a grana grossa anziché tanti messaggi a grana fine



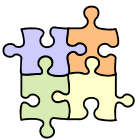
Dai servizi alle SOA

- Dunque, una SOA è organizzata soprattutto attorno a due livelli (strati) fondamentali, i cui componenti rappresentano, rispettivamente
 - servizi di business
 - processi di business
- Inoltre, i processi di business sono definiti come composizione di servizi
 - la composizione è di solito un'attività di assemblaggio – e solo raramente un'attività di sviluppo
 - per questo i processi di business possono essere definiti o modificati “rapidamente”



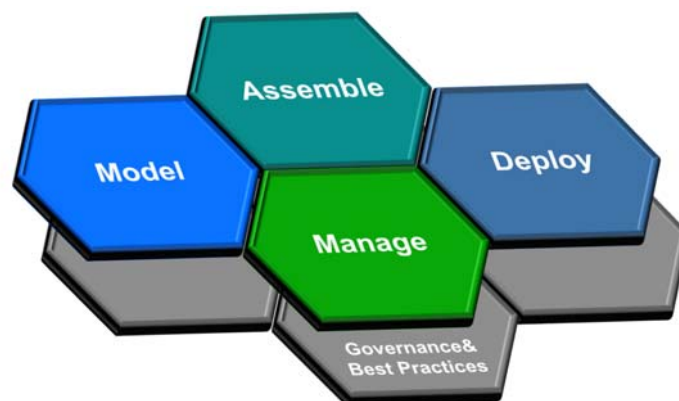
Dai servizi ai processi di business

- Quando si presenta una nuova opportunità di business per l'organizzazione, gli sviluppatori possono implementare rapidamente un nuovo processo di business – assemblando una nuova applicazione a partire dai servizi disponibili, e creando nuovi servizi se necessario
 - l'organizzazione potrebbe aver già definito dei servizi – da riusare nella realizzazione di nuove applicazioni
 - in alcuni casi può essere necessario implementare nuove funzionalità – bisogna valutare la possibilità di implementare queste nuove funzionalità come servizi, per poterli poi usare in altre applicazioni
 - inoltre, può essere anche utile considerare servizi forniti dall'esterno dell'organizzazione



- Ciclo di vita nelle SOA

- In una SOA, lo sviluppo del software non avviene sulla base di un processo tradizionale progetta/compila/ esegui, ma piuttosto sulla base di un processo iterativo modella/ assembla/ rilascia- configura/ monitora-gestisci





Ciclo di vita nelle SOA

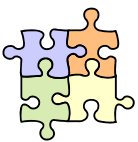
□ Ciclo di vita SOA

- model
 - trova i requisiti (di business) – modella (il business, e non l'applicazione) e simula – progetta (i servizi)
- assemble
 - scopri – costruisci e verifica – componi
 - poiché il green-field development è raro, questa attività può essere di solito svolta piuttosto rapidamente
- deploy
 - essenzialmente integrazione – di persone, processi e informazioni
- manage
 - gestisci (i servizi e i processi di business, e non le applicazioni) – monitora (metriche di business)

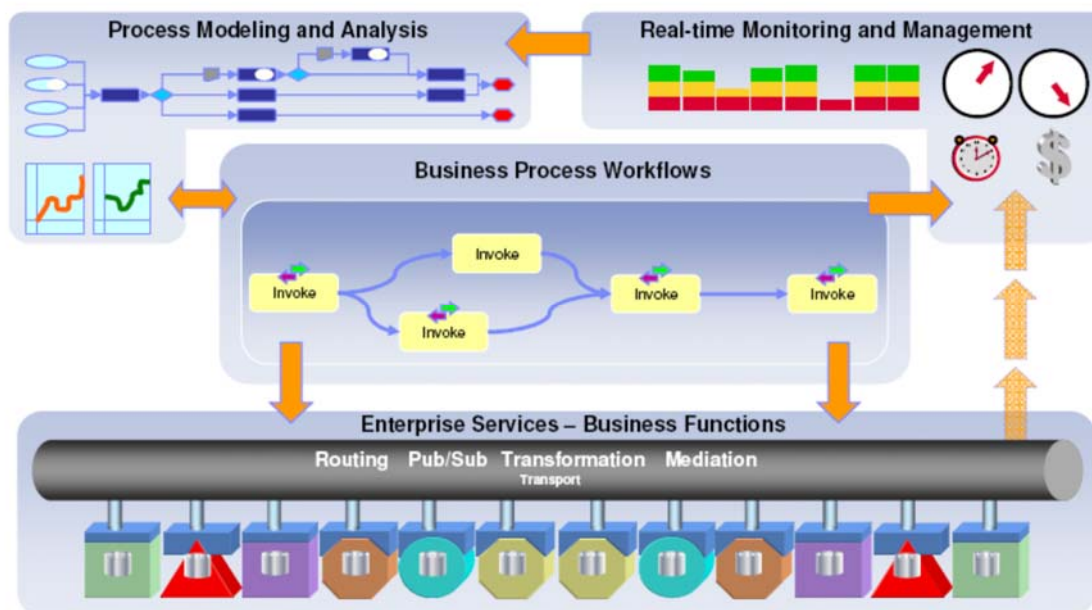
37

Service-Oriented Architecture

Luca Cabibbo – ASw



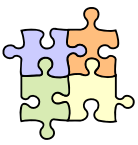
Ciclo di vita nelle SOA



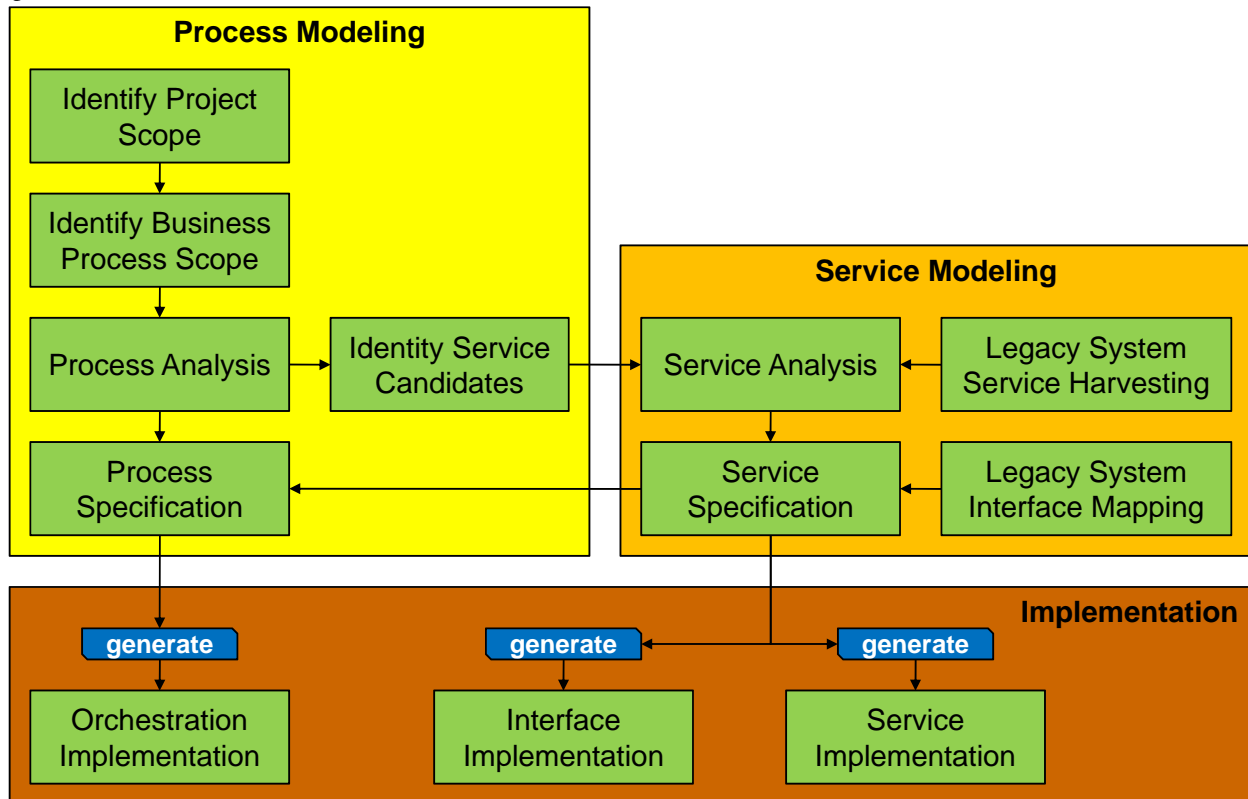
38

Service-Oriented Architecture

Luca Cabibbo – ASw



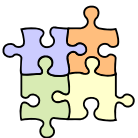
Ciclo di sviluppo nelle SOA



39

Service-Oriented Architecture

Luca Cabibbo - ASw



* Principi per la progettazione dei servizi

- I servizi – che sono il costrutto fondamentale in una SOA – devono essere progettati in modo opportuno, affinché la SOA possa raggiungere i propri obiettivi
 - per questo, la progettazione dei servizi viene di solito guidata dall'applicazione di un insieme di principi di progettazione
 - non esiste un insieme di principi “universalmente accettati” – qui vengono presi in considerazione alcuni principi comuni, descritti da [Erl]
 - alcuni di questi principi sono sostenuti (ma non garantiti!) dalla tecnologia a servizi
 - i principi fondamentali riguardano
 - astrazione dei servizi (incapsulamento)
 - contratto formale (interfaccia)
 - autonomia dei servizi
 - accoppiamento debole, riusabilità, componibilità

40

Service-Oriented Architecture

Luca Cabibbo - ASw



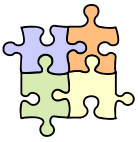
Principi per la progettazione dei servizi (1)

- Alcuni principi che guidano la progettazione dei servizi nelle SOA
 - *i servizi condividono un contratto formale* – per consentire l'interazione, ciascun servizio condivide un contratto formale, che descrive lo scambio di informazioni con il servizio – nonché ogni altra informazione aggiuntiva di interesse
 - *i servizi realizzano un'astrazione della logica sottostante* – la sola parte del servizio che è visibile all'esterno è ciò che è esposto tramite la sua descrizione (il contratto formale del servizio) – tutta la logica sottostante (l'implementazione del servizio) è invisibile e irrilevante per chi vuole richiedere il servizio



Principi per la progettazione dei servizi (2)

- Alcuni principi che guidano la progettazione dei servizi nelle SOA
 - *i servizi hanno un'interfaccia accessibile in rete* – gli utilizzatori di un servizio devono poter invocare il servizio in rete, in modo remoto
 - *i servizi possono essere scoperti* – la descrizione dei servizi deve poter essere scoperta e compresa da chi vuole poter utilizzare la logica rappresentata dai servizi stessi – la scoperta dei servizi può essere facilitata dall'uso di un servizio di directory da parte dei fornitori di servizi
 - *la locazione dei servizi è trasparente* – gli utilizzatori di un servizio non devono essere obbligati ad accedere al servizio mediante un indirizzo di rete assoluto – piuttosto, devono poter scoprire dinamicamente la posizione di un servizio consultando un registry dei servizi – questo consente di cambiare la posizione di un servizio senza avere impatto sul richiedente



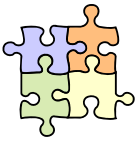
Principi per la progettazione dei servizi (3)

- Alcuni principi che guidano la progettazione dei servizi nelle SOA
 - *i servizi sono componibili* – i servizi possono comporre altri servizi – questo consente di avere la logica applicativa rappresentata a diversi livelli di granularità – sostiene la riusabilità e la creazione di livelli di astrazione – inoltre, la componibilità dei servizi sostiene la possibilità di cambiare rapidamente applicazioni e processi di business
 - *i servizi sono riusabili* – i servizi sono progettati per sostenere un potenziale riuso, indipendentemente dal fatto che esistano o meno delle opportunità immediate di riuso
 - *i servizi sono debolmente accoppiati* – i servizi devono essere progettati per interagire in modo debolmente accoppiato



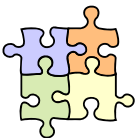
Principi per la progettazione dei servizi (4)

- Alcuni principi che guidano la progettazione dei servizi nelle SOA
 - *i servizi sono stateless* – i servizi non devono gestire informazioni sullo stato delle conversazioni con i loro client (questo limiterebbe la possibilità di un accoppiamento debole) – i servizi devono essere progettati per massimizzare questa assenza di stato – eventualmente delegando altrove la gestione dello stato delle conversazioni
 - *i servizi sono autonomi* – l'autonomia di un servizio riguarda la sua implementazione e il suo ambiente di esecuzione – l'implementazione di ogni servizio deve essere autonoma (nel senso che non ci devono essere vincoli nella sua implementazione) – inoltre, ogni servizio deve avere capacità di auto-governo (un'autonomia di governo completa) entro il suo ambiente di esecuzione



Una caratterizzazione alternativa

- Un memo di Jeff Bezos indirizzato al team di Amazon [circa 2002] fornisce un'altra caratterizzazione di questi principi
 - All teams will henceforth expose their data and functionality through service interfaces.
 - Teams must communicate with each other through these interfaces.
 - There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
 - It doesn't matter what technology they use. HTTP, Corba, Pub/sub, custom protocols – doesn't matter. Bezos doesn't care.
 - All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
 - Anyone who doesn't do this will be fired.
 - Thank you; have a nice day!



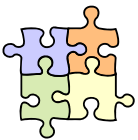
* SOA e obiettivi di business II

- In che modo una SOA sostiene gli obiettivi di business di un'organizzazione?
 - esaminiamo in particolare il modo in cui sono raggiunti i seguenti obiettivi di business
 - agilità – capacità di adattarsi a opportunità e minacce competitive
 - capacità di offrire servizi innovativi
 - ottimizzare i processi
 - integrazione con i sistemi legacy



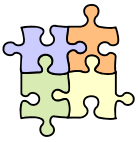
Agilità

- Agilità – essere capaci di adattarsi rapidamente a nuove opportunità e potenziali minacce in un contesto competitivo
 - l'agilità è sostenuta da alcuni principi per la progettazione dei servizi
 - in particolare, riusabilità e componibilità dei servizi – ma anche accoppiamento debole e condivisione della specifica formale
 - inoltre, l'agilità è favorita dall'uso di servizi standardizzati, dalla conformità a standard, e dalla definizione di servizi a grana (abbastanza) grossa
 - ad es., se cambiano le regole di business associate a una specifica funzione, allora è necessario modificare solo il servizio che implementa quella funzione – tutte le applicazioni che usano il servizio vengono automaticamente adattate all'uso delle nuove regole di business



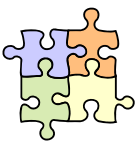
Offrire servizi innovativi

- Essere i primi nel mercato con servizi innovativi per i propri clienti
 - questo obiettivo è sostenuto dalla possibilità di
 - comporre servizi, combinandoli in modi nuovi e diversi
 - usare sia servizi interni che servizi esterni
 - aggiungere nuovi servizi ove necessario
 - 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



Ottimizzare i processi

- Ottimizzare i propri processi di business – per aumentare efficienza ed efficacia, e ridurre i costi operativi
 - l'integrazione e 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
 - usare sia servizi interni che servizi esterni
 - ad es., è possibile cambiare la definizione di un processo di business in termini di una nuova composizione di servizi
 - 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



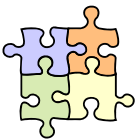
Integrazione con i sistemi legacy

- Abilitare un'integrazione semplice e flessibile con i propri sistemi legacy
 - molti sistemi legacy implementano un insieme ricco di funzionalità – queste funzionalità possono essere esposte come servizi, per essere (ri)usati come elementi di una SOA
 - se questo viene fatto usando tecniche e strumenti non invasivi, allora, allo stesso tempo
 - i sistemi legacy possono rimanere integri, e continuare ad offrire le loro funzionalità
 - queste funzionalità possono essere fruite anche da altri consumatori di servizi
 - poiché il costo di re-implementare un sistema legacy è di solito inaccettabilmente alto, rendere le sue funzionalità accessibili come servizi è un'alternativa economica che viene perseguita da molte organizzazioni



Discussione

- Un aspetto importante nella progettazione dei servizi è identificare quali pezzi di funzionalità diventeranno servizi, e definire con cura le relative interfacce
 - infatti, quando i servizi sono stati implementati, e le loro interfacce pubblicate e usate da varie applicazioni, è in genere difficile modificare la definizione dei servizi, perché questo potrebbe richiedere cambiamenti in tutte le applicazioni che li utilizzano
 - questi cambiamenti vanno evitati
 - quando una regola di business associata a una specifica funzionalità cambia, gli sviluppatori possono modificare (una sola volta) l'implementazione del servizio per quella funzionalità – tutte le applicazioni che usano quel servizio adotteranno automaticamente la nuova regola di business
 - una modificabilità di questo tipo va invece perseguita



- Ulteriori definizioni

- **SOA** – *Service-Oriented Architecture* [SEI]
 - un **servizio** è un'implementazione di un pezzo ben definito di funzionalità di business – con un'interfaccia che è pubblicata e può essere cercata/trovata – che può essere usato da consumatori di servizi nella costruzione di diversi processi di business e applicazioni
 - **SOA** è un approccio architetturale per costruire sistemi e applicazioni che usano un insieme di servizi – e non solo un singolo sistema come un insieme di servizi



Ulteriori definizioni

- **SOA – Service-Oriented Architecture** [Marks&Bell]
 - un **servizio** è una funzionalità di business con un'interfaccia esposta, che può essere invocato dai suoi consumatori mediante messaggi
 - **SOA** è un'architettura concettuale di business in cui le funzionalità di business (logica applicativa) vengono esposte agli utenti SOA come servizi riusabili e condivisi in rete
 - un **servizio** è un'unità, modulare e riusabile, di capacità di business, processo o funzione tecnica, che può essere acceduto/utilizzato in modo ripetuto da una molteplicità di consumatori
 - i servizi sono la risorsa architeturale primaria di una SOA
 - **SOA** è una disciplina critica per far sì che i servizi lavorino insieme per aiutare l'organizzazione a raggiungere i propri obiettivi di business



Ulteriori definizioni

- **SOA – Service-Oriented Architecture** [Papazoglou]
 - lo scopo essenziale di una **SOA** è di abilitare l'interoperabilità tra tecnologie esistenti, nonché l'estendibilità a scopi e architetture futuri
 - **SOA** è uno stile architeturale il cui obiettivo è consentire alle organizzazioni di sviluppare, connettere e mantenere applicazioni e servizi di tipo enterprise in modo efficiente ed economico
 - una **SOA** fornisce un insieme di linee guida, principi e tecniche per cui i beni, le informazioni, e i processi di business di un'organizzazione possono essere ri-organizzati efficacemente per sostenere e abilitare piani strategici e livelli di produttività come richiesto da ambienti di business competitivi



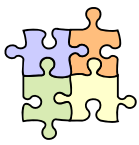
- SOA e qualità

- Riassumendo, SOA è uno stile architetturale che persegue un certo numero di obiettivi di qualità – tra cui
 - **obiettivi di business**
 - agilità di business
 - integrazione dei processi di business all'interno dell'organizzazione
 - integrazione dei processi di business con partner, fornitori e clienti
 - possibilità di monitorare/governare l'efficacia dei miglioramenti nei processi di business
 - **obiettivi tecnici**
 - interoperabilità – indipendenza dalle tecnologie e dalle piattaforme
 - riduzione della complessità
 - modello architetturale uniforme per l'integrazione di applicazioni interne ed esterne



SOA e qualità

- Riassumendo, SOA è uno stile architetturale che persegue un certo numero di obiettivi di qualità – tra cui
 - **obiettivi di business**
 - agilità di business
 - integrazione dei processi di business all'interno dell'organizzazione
 - integrazione dei processi di business con partner, fornitori e clienti
 - possibilità di monitorare/governare l'efficacia dei miglioramenti nei processi di business
 - **obiettivi tecnici**
 - interoperabilità – indipendenza dalle tecnologie e dalle piattaforme
 - riduzione della complessità
 - modello architetturale uniforme per l'integrazione di applicazioni interne ed esterne



SOA e qualità

- SOA è uno stile architetturale che sostiene agilità di business e altri obiettivi di qualità – a tal fine, lo stile SOA è basato su numerosi principi di progettazione
 - uso di una tecnologia a servizi, che favorisce l'interoperabilità
 - Layers come primo criterio di decomposizione – per sostenere modificabilità, con riferimento a diversi livelli di astrazione
 - con due strati di business fondamentali (servizi e processi)
 - accoppiamento debole
 - tra servizi ed altri servizi – tra servizi e processi
 - ottenuto mediante incapsulamento (separazione netta tra interfaccia e implementazione, interazioni basate solo sull'interfaccia), messaging come stile preferito di interazione, servizi stateless, composizione di servizi, ...
 - riuso
 - ...

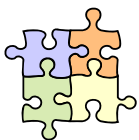
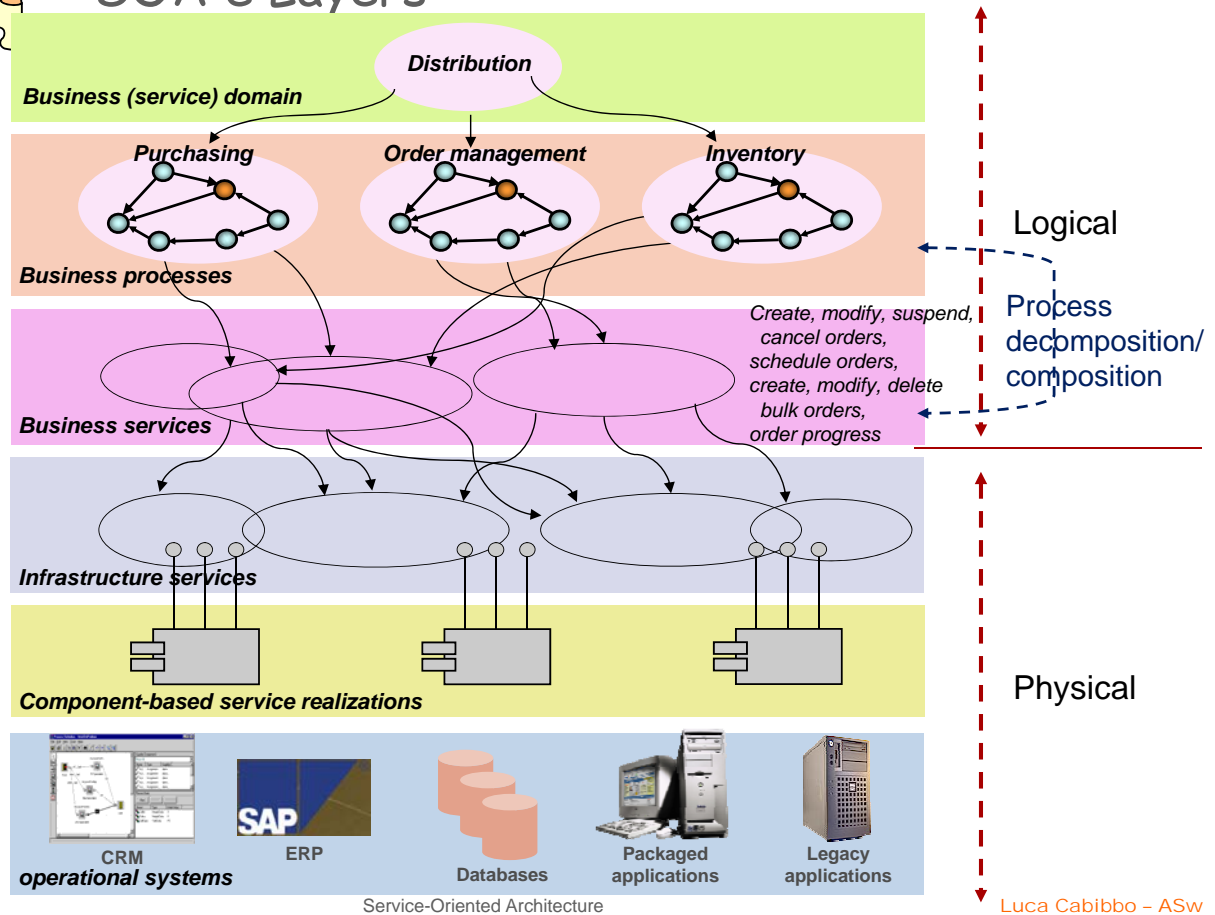


* SOA e Layers

- Lo stile SOA adotta solitamente Layers come primo criterio di decomposizione
 - Layers sostiene la modificabilità in sistemi complessi che devono occuparsi della gestione di diversi aspetti, a differenti livelli di astrazione
 - nota
 - è possibile trovare in letteratura molti modelli per SOA, più o meno complessi, più o meno simili
 - qui viene considerato il modello proposto da [Papazoglou]
 - i due strati che costituiscono il focus di interesse principale per le metodologie di progettazione orientate ai servizi sono quello dei processi di business e quello dei servizi di business



SOA e Layers



SOA e Layers

□ Dominio di business

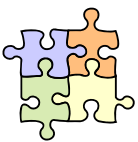
- un'organizzazione può essere partizionata in un insieme di **domini funzionali di business** disgiunti – ciascuno di questi domini comprende diversi processi di business, che condividono capacità e funzionalità
 - ad es., servizio bancario per privati, servizio bancario per aziende
- il SEI chiama questo strato "Presentazione"
 - lo scopo è disaccoppiare l'implementazione della presentazione lato client dall'implementazione dei servizi e dei processi



SOA e Layers

□ Processi di business

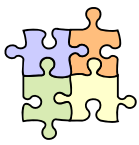
- l'organizzazione eroga un certo numero di *processi di business* fondamentali – ciascuno nell'ambito di un dominio di business
 - ad es., erogazione di un mutuo per l'acquisto di una casa
- ciascun processo di business viene definito come composizione di un certo numero di servizi di business



SOA e Layers

□ Servizi di business

- un *servizio di business* rappresenta un compito di business elementare (non ulteriormente decomponibile) e automatizzato – che può fornire valore all'organizzazione e che può essere usato in uno o più processi di business
- due tipi
 - funzionalità di business
 - ad es., apertura di un conto corrente
 - servizi di utilità – riusabili da molteplici servizi di business
 - ad es., un servizio di directory



SOA e Layers

- Servizi infrastrutturali
 - servizi tecnici, d'accesso, di gestione e monitoraggio, di interazione – riusabili in tutti i servizi e processi di business
- Realizzazione dei servizi basata su componenti
 - i **componenti** sono unità autonome di funzionalità – rappresentano una demarcazione naturale del lavoro di implementazione
 - talvolta adattatori verso sistemi legacy (preesistenti)
- Sistemi operazionali
 - questo strato comprende le applicazioni e i sistemi informatici esistenti
 - uno degli scopi di una SOA è proprio quello di continuare a sfruttare questi sistemi, nonché di consentire l'interoperabilità e l'integrazione con altri sistemi

63

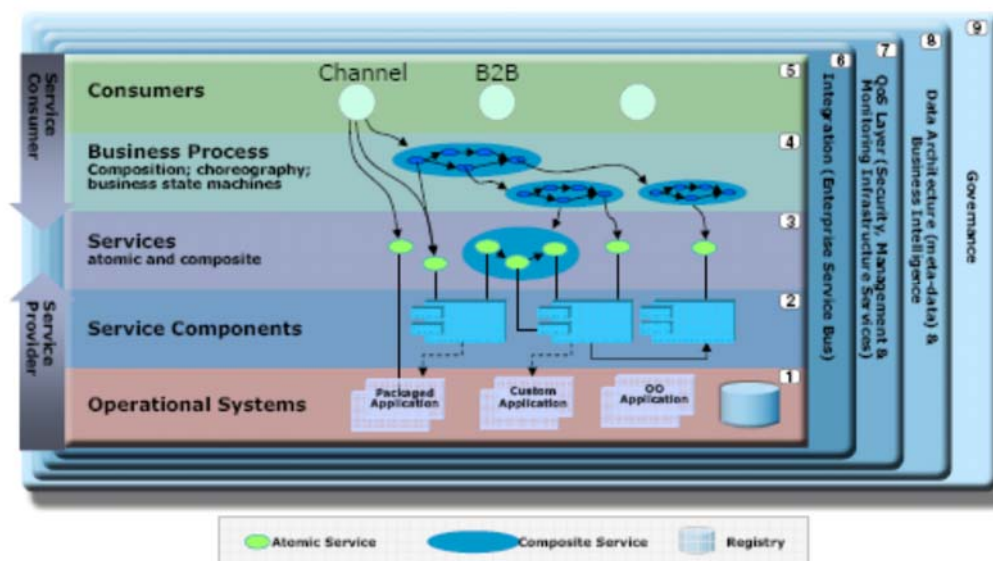
Service-Oriented Architecture

Luca Cabibbo – ASw



SOA e Layers

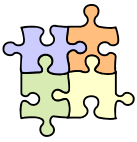
- Il modello SOA dell'IBM



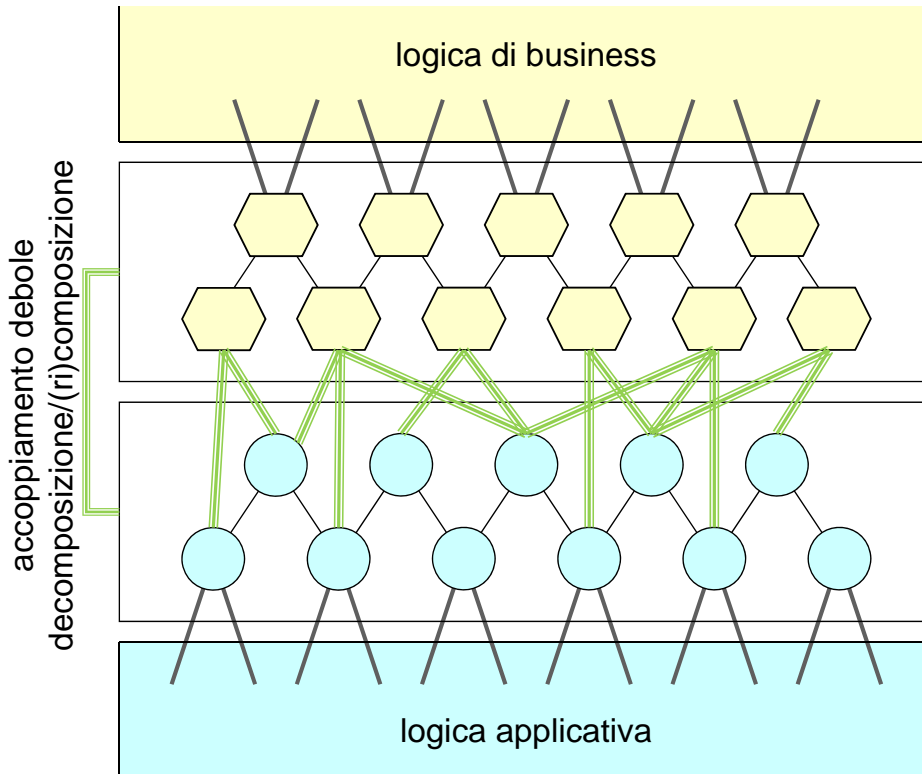
64

Service-Oriented Architecture

Luca Cabibbo – ASw

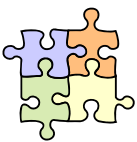


Strati e agilità di business

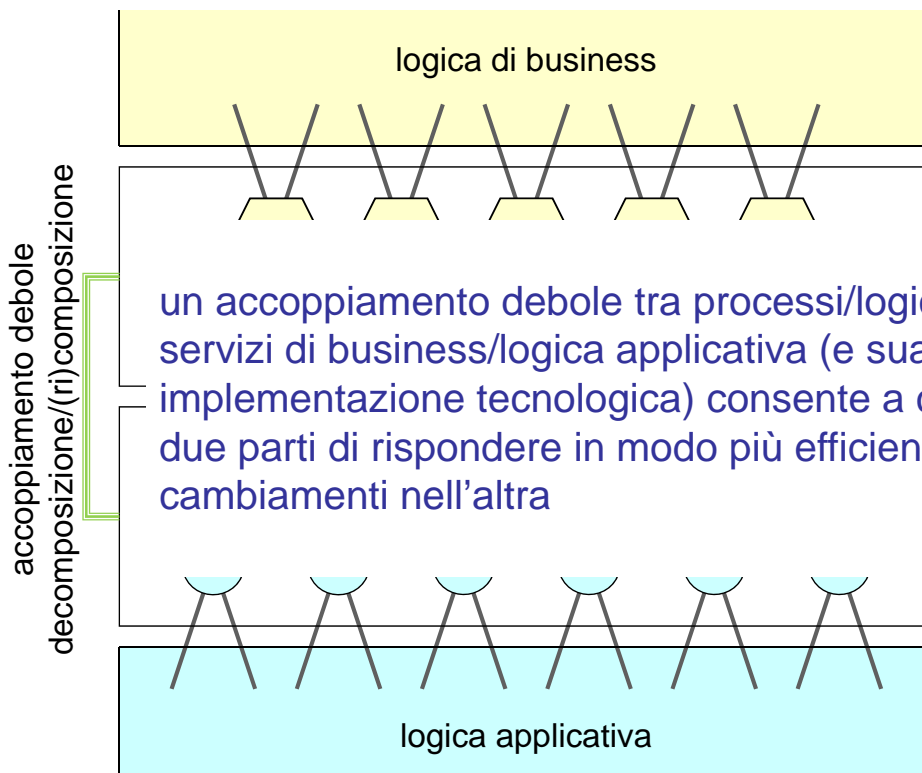


servizi che implementano *logica/processi di business*

servizi di business – incapsulano (e astraggono) la logica applicativa e le risorse tecnologiche

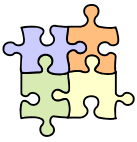


Strati e agilità di business



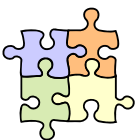
un accoppiamento debole tra processi/logica di business e servizi di business/logica applicativa (e sua implementazione tecnologica) consente a ciascuna delle due parti di rispondere in modo più efficiente a cambiamenti nell'altra

e le risorse tecnologiche



Agilità di business, processi e servizi

- Per consentire un'agilità di business
 - i servizi di business devono fornire unità coese di funzionalità di business – debolmente accoppiate tra loro e con i processi – che possono essere utilizzate in più processi di business
 - i processi di business devono essere definiti come composizione di servizi di business – in una forma semplice da modificare
- Questa è una delle idee fondamentali alla base delle architetture a servizi
 - questi requisiti hanno un impatto importante sulle proprietà che devono possedere i servizi che contribuiscono alla definizione di una SOA
 - e motivano i principi per la progettazione dei servizi che sono stati illustrati in precedenza



* Enterprise Service Bus

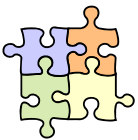
- Nelle SOA è necessaria un'infrastruttura di comunicazione distribuita per sostenere concretamente l'interoperabilità tra diverse tecnologie, in modo flessibile e scalabile
 - infatti, in una SOA devono convivere, interoperando, numerose applicazioni e componenti sviluppati autonomamente
- Una possibile infrastruttura per la realizzazione e il deployment di una SOA è fornita dal pattern architetturale *Enterprise Service Bus*
 - un ESB affronta le problematiche di deployment della SOA – sicuramente significative in presenza di servizi realizzati in ambienti distribuiti/eterogenei
 - consente così di affrontare separatamente gli aspetti funzionali da quelli di deployment
 - ESB è un pattern architetturale per il brokering tra servizi – in generale, dunque, il termine ESB non indica un prodotto



Enterprise Service Bus

□ *Enterprise Service Bus (ESB)*

- è un'infrastruttura di connettività flessibile
- per integrare applicazioni, sistemi eterogenei e servizi
- sulla base di standard e funzionalità MOM
- riducendo numero, dimensione e complessità delle interfacce
- al fine di abilitare implementazione, deployment e gestione di soluzioni basate su SOA



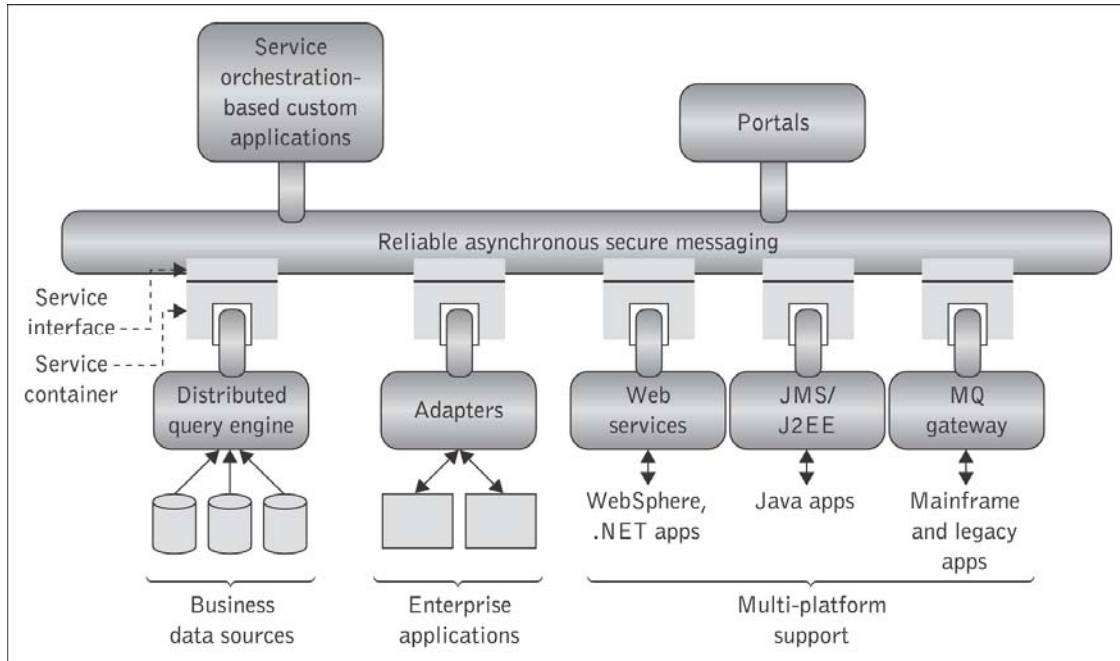
Funzionalità di un ESB

□ Alcune funzionalità e caratteristiche essenziali di un ESB

- supporto fondamentale per i web services, per la loro invocazione e composizione
- routing di messaggi tra applicazioni e servizi – anche basato sul contenuto dei messaggi
- conversione di protocolli di trasporto
- validazione e trasformazione di dati e messaggi
- distribuzione di eventi di business
- possibilità di distribuire (e coordinare) le funzionalità su più server
- supporto per la connessione a sistemi legacy
- supporto per sicurezza, transazioni, affidabilità, ...
- strumenti per l'amministrazione integrata, la gestione della sicurezza, per il monitoraggio runtime dei servizi e dei processi

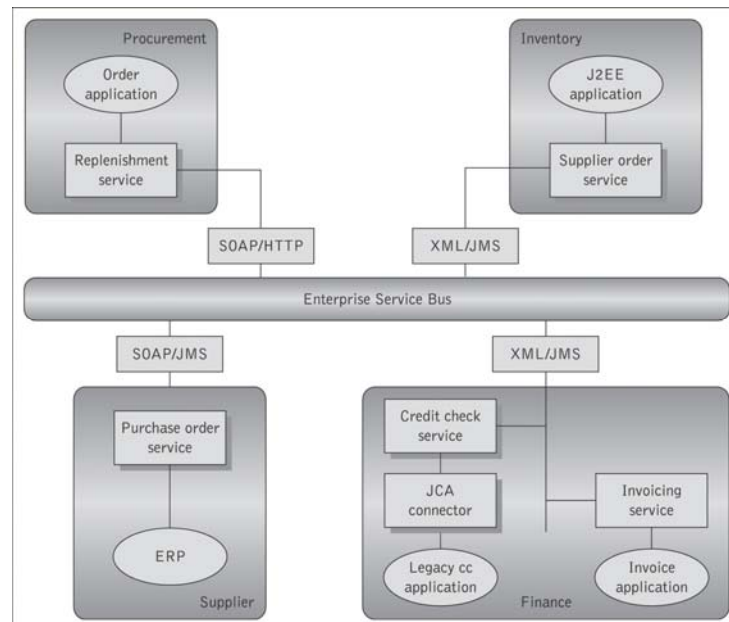


ESB - connessione di diverse tecnologie



ESB - connessione di servizi remoti

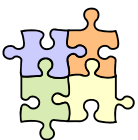
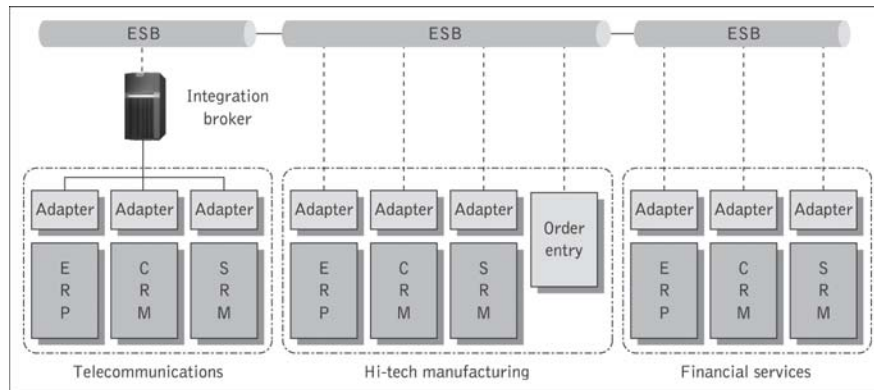
- L'infrastruttura ESB comprende un framework per l'elaborazione distribuita e il supporto per l'orchestrazione del comportamento di servizi in un processo distribuito





Possibili implementazioni di un ESB

- Sono possibili diverse implementazione alternative per un ESB
 - ESB unico centralizzato
 - ESB come federazione di ESB
- Esempio
 - cooperazione basata su una federazione di ESB distribuiti



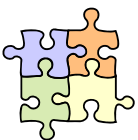
Elementi di un ESB

- Elementi di un ESB – per fornire le facilitazioni di supporto ai diversi stili di interazione
 - integration broker
 - facilita il movimento (e la trasformazione) di informazioni tra più partecipanti
 - application server
 - offrono un ambiente integrato di sviluppo ed esecuzione, per il deployment di applicazioni basate (o meno) sul web
 - business process management
 - supporta l'esecuzione e il coordinamento di processi di business di lunga durata, che coinvolgono più applicazioni e più utenti
 - supportano il monitoraggio della qualità dei servizi erogati



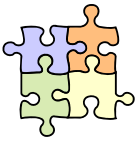
* Discussione

- In questa dispensa sono stati presentati alcuni aspetti fondamentali delle architetture orientate ai servizi
 - SOA è uno stile per l'organizzazione e la fruizione di un insieme distribuito di capacità funzionali – che sono controllate da una o più organizzazioni
 - una SOA fornisce un modo uniforme per offrire, scoprire, interagire con e usare le capacità di un insieme di servizi software debolmente accoppiati ed interoperabili
 - una SOA sostiene alcuni importanti obiettivi di business, comuni a molte organizzazioni
 - l'infrastruttura tecnologica per una SOA può essere realizzata mediante un ESB



Discussione

- Una *metodologia* è di importanza critica per poter specificare, costruire, raffinare, personalizzare e far evolvere nel tempo processi di business altamente volatili a partire da web services disponibili internamente e/o esternamente
 - in particolare, di solito non è opportuno/sufficiente introdurre un “sottile strato di web services” sopra i componenti e le applicazioni esistenti
 - è critica l'analisi e la progettazione (specifica) dei servizi
- Per aspetti metodologici relativi alle SOA, si veda
 - Papazoglou, *Web Services – Principles and Technology*, 2008, Chapter 15: Web services lifecycle management
 - Bell, *Service-Oriented Modeling – Service Analysis, Design, and Architecture*, 2008



Discussione

- La **governance** di una SOA si riferisce all'organizzazione, ai processi, alle politiche e alle metriche richieste per gestire una SOA con successo – ovvero, per far in modo che l'IT soddisfi gli obiettivi di business nel tempo

- Per aspetti relativi alla governance di una SOA, si veda
 - Marks&Bell, *Service-Oriented Architecture – A Planning and Implementation Guide for Business and Technology*, 2006