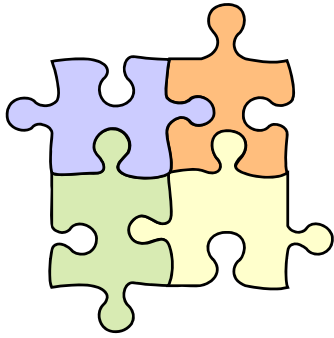


Luca Cabibbo



Architetture Software

Service Oriented Architecture

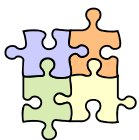
Dispensa PA 5

ottobre 2008

1

Service Oriented Architecture

Luca Cabibbo – SwA



- Fonti

- [Papazoglou] Web Services – Principles and Technology
- [ACKM] Web Services – Concepts, Architectures and Applications
- [Marks&Bell] Service-Oriented Architecture – a planning and implementation guide for business and technology

2

Service Oriented Architecture

Luca Cabibbo – SwA



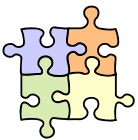
- Obiettivi e argomenti

□ Obiettivi

- conoscere alcuni aspetti fondamentali delle architetture orientate ai servizi

□ Argomenti

- introduzione
- web services e servizi
- service oriented architecture
- enterprise service bus
- discussione



* Introduzione

□ I Web Services rappresentano la tecnologia basata su servizi “dominante”

- una tecnologia per l’integrazione di applicazioni distribuite, volta a risolvere problemi di interoperabilità, basata su standard accettati dalla maggior parte dei produttori di software

□ Le architetture orientate ai servizi forniscono il contesto metodologico (e di business) in cui utilizzare al meglio le tecnologie basate su servizi



Servizi e architetture orientate ai servizi

- Il paradigma orientato ai servizi è basato sui servizi come costruito per sostenere sviluppo e composizione di applicazioni distribuite
 - un **servizio** – può essere implementato come WS
 - ha l'obiettivo di incapsulare una funzionalità di business (logica applicativa) per renderla disponibile ed accessibile come servizio software da parte client software sul web
 - **SOA (Service-Oriented Architecture)**
 - architettura concettuale di business, in cui le funzionalità di business vengono esposte agli utenti come servizi riusabili e condivisi in rete
 - il nuovo approccio “orientato ai servizi” è basato sull'idea di comporre applicazioni sulla base della scoperta e dell'invocazione di servizi disponibili in rete
 - anziché costruire nuove applicazioni o invocare le applicazioni disponibili solo per svolgere compiti specifici



Relazione tra WS e SOA

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



- Motivazioni

- In quale contesto si muovono oggi le grandi organizzazioni?
 - alta competitività ed alta incertezza
 - integrazione globale – di informazioni e processi
 - le organizzazioni devono poter cambiare rapidamente i propri processi di business
 - per offrire i propri servizi in modo più efficiente/efficace e/o per offrire nuovi servizi e/o offrire i propri servizi a nuovi mercati
 - devono cambiare e poter continuare a cambiare in modo flessibile
 - le organizzazioni del futuro richiedono agilità
- Le tecnologie informatiche devono sostenere – e non ostacolare – l'agilità di business delle organizzazioni
 - in che modo?

7

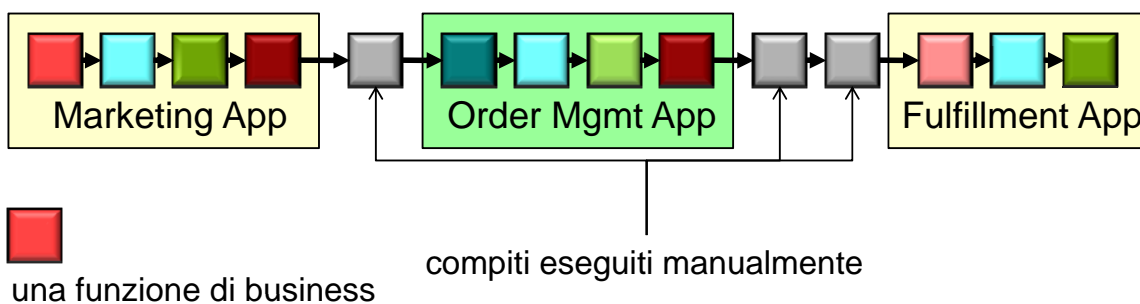
Service Oriented Architecture

Luca Cabibbo – SwA



Esempio

- Un processo di business tradizionale

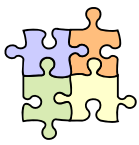


- processo realizzato in più applicazioni separate
- funzioni di business accoppiate alle applicazioni
- alcune funzioni sono replicate in più applicazioni – con interfacce proprietarie, il riuso è difficile
- i passi manuali complicano la situazione
- difficile monitorare il processo di business
- difficile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

8

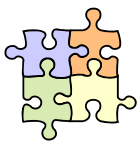
Service Oriented Architecture

Luca Cabibbo – SwA

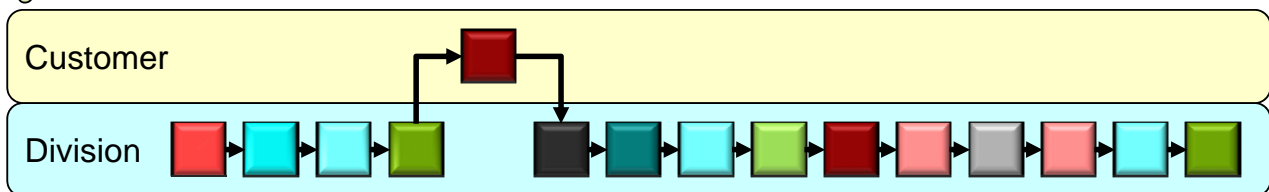


Esempio

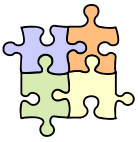
- Quale organizzazione del software per sostenere flessibilità nel cambiamento dei processi di business?
 - fornire la possibilità a più sistemi informatici di lavorare insieme
 - tecnologicamente – mediante un insieme di standard per l'interoperabilità
 - ma una interoperabilità tecnologica è insufficiente
 - è necessario anche allineare tecnologia e business
 - consentendo all'uno di cambiare insieme all'altro
- Nel seguito dell'esempio
 - sono mostrate alcune motivazioni per aumentare il riuso e la flessibilità di business – mediante una SOA



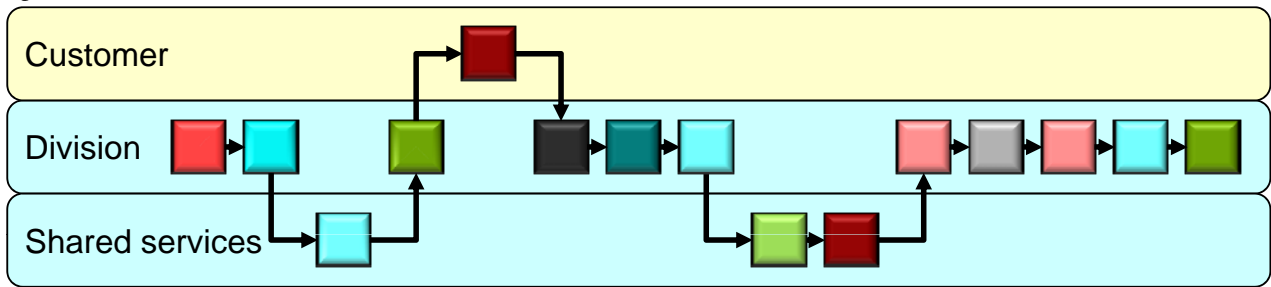
Flessibilità - interazione diretta con il cliente



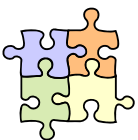
- ora il cliente può acquistare direttamente dal web
- un partner commerciale (cliente) può fare ordini B2B mediante un web service
- i vari clienti sono serviti meglio



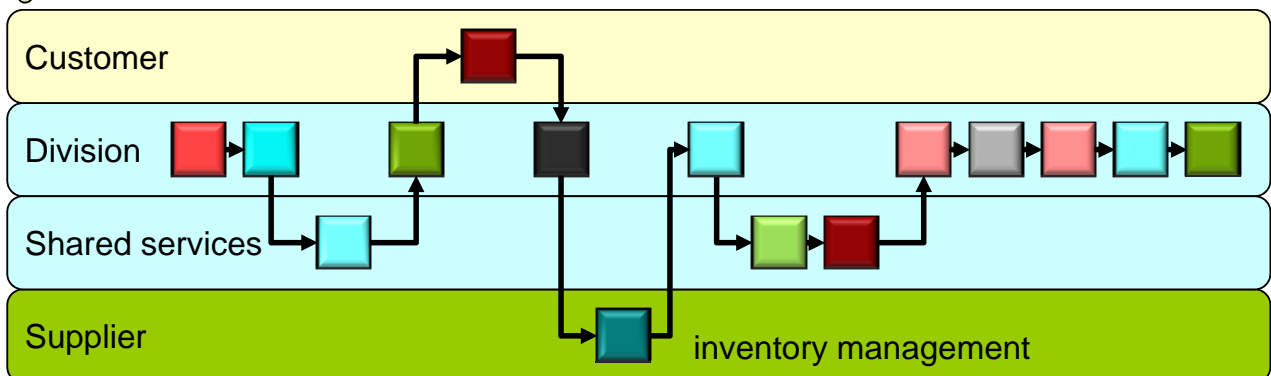
Flessibilità - condivisione di servizi



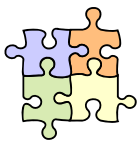
- i servizi di business comuni possono essere condivisi dall'intera organizzazione – e consolidati
- riduzione delle ridondanze – dei relativi costi di sviluppo e di gestione – possibilità di ottenere economie di scala



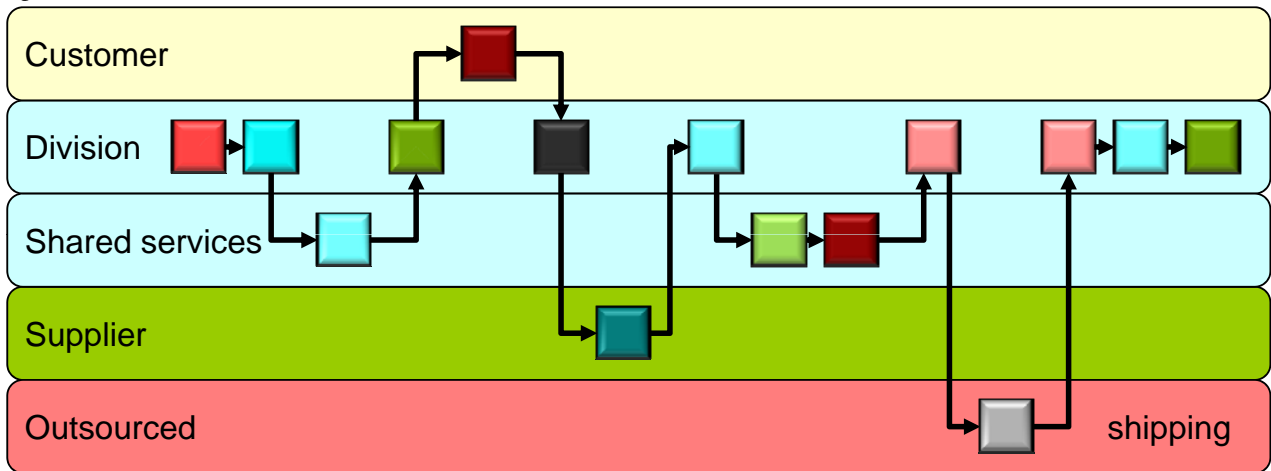
Flessibilità - inventario gestito dal fornitore



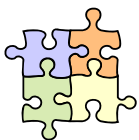
- possibilità di delegare funzionalità a partner commerciali (fornitori) mediante interazioni B2B
- riduzione dei costi e servizio migliore



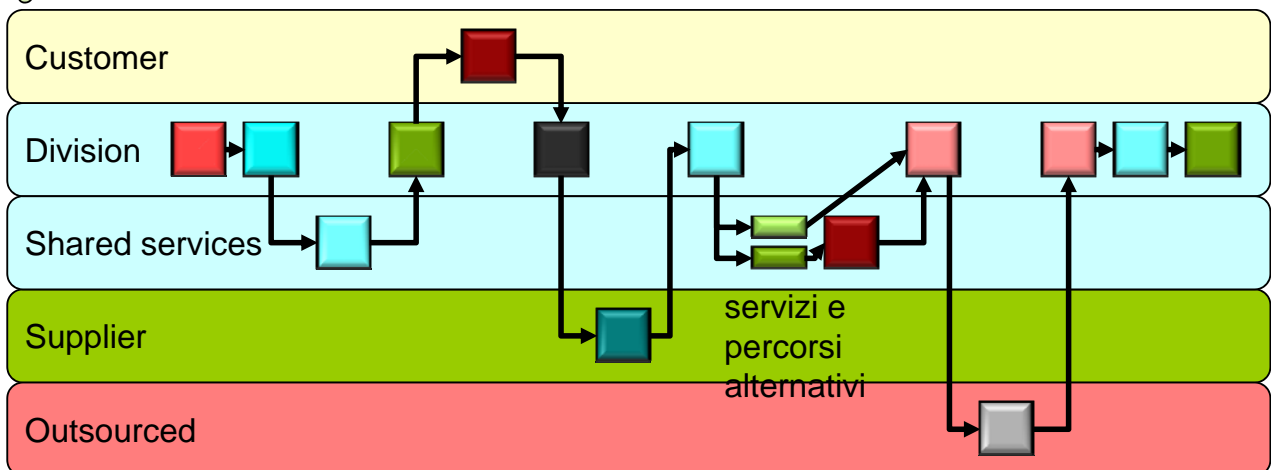
Flessibilità - outsourcing



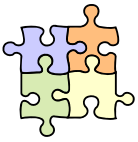
- possibilità di dare in outsourcing funzionalità relative a competenze non fondamentali
- riduzione di costi e delle infrastrutture di spedizione



Flessibilità - miglioramento del processo

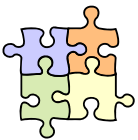


- possibilità di modificare il processo e le regole di gestione dello stesso
- la definizione e la ridefinizione di un processo è vista più come un'attività di *assemblaggio* di servizi/compiti che non come un'attività di *sviluppo*



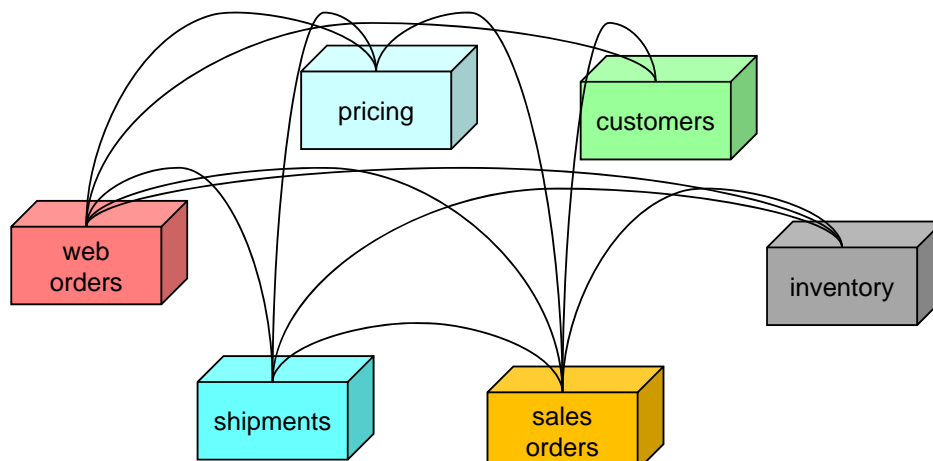
Agilità di business e SOA

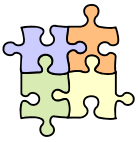
- Il business di un'organizzazione può essere flessibile solo quanto la sua infrastruttura tecnologica (IT)
 - se l'IT non può cambiare, non può cambiare nemmeno il business
- SOA si propone come stile architetturale, con i seguenti obiettivi
 - consentire di allineare business e IT
 - in modo che possano variare insieme
 - sostenere agilità di business
 - consentire lo sviluppo rapido di nuove applicazioni
 - realizzate come applicazioni composte
 - basate sul riuso di elementi software già esistenti all'interno dell'organizzazione
 - nonché sulla possibilità di fruire anche di elementi software esterni all'organizzazione



Limiti delle architetture a componenti

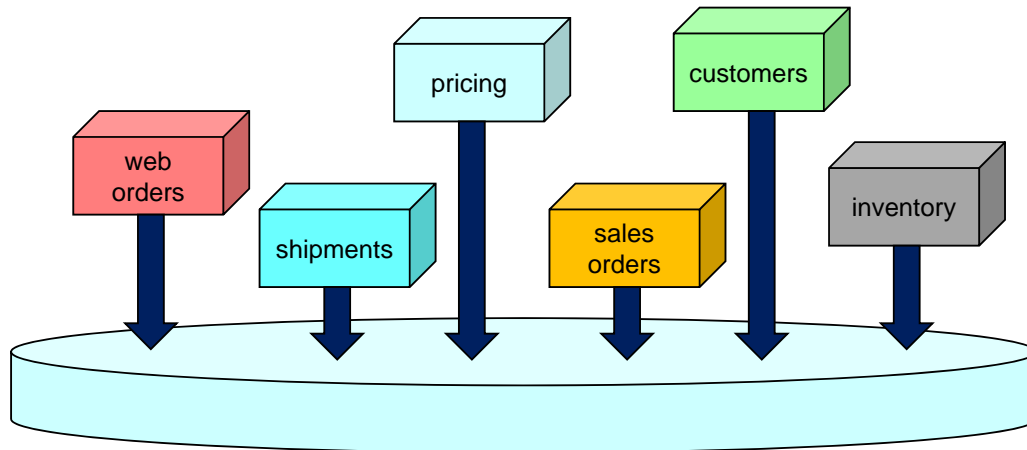
- Le architetture a componenti offrono numerosi vantaggi
 - ad esempio, poter definire componenti che incapsulano funzionalità di business
 - tuttavia, nei contesti di integrazione, ciascun componente deve essere opportunamente collegato ad altri componenti – aumentando la complessità del sistema





Verso le SOA

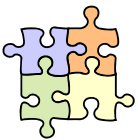
- In un'architettura a servizi
 - ciascun componente (servizio) incapsula una funzionalità di business
 - la logica IT è portata ad di fuori dei servizi
 - i servizi sono debolmente accoppiati – e possono essere integrati e composti più facilmente e più agilmente



17

Service Oriented Architecture

Luca Cabibbo – SwA



* Web Services e servizi

- Un **Web Service**
 - è un modulo o componente software, auto-contenuto ed 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
- Un servizio realizzato come WS può rappresentare
 - un compito di business auto-contenuto – ad es., aprire un conto corrente o effettuare un bonifico bancario
 - un intero processo di business – ad es., l'acquisto automatizzato di forniture d'ufficio
 - un'applicazione – ad es., gestione di assicurazioni sulla vita
 - l'accesso (come servizio) ad una risorsa – ad es., a cartelle mediche

18

Service Oriented Architecture

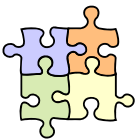
Luca Cabibbo – SwA



Che cosa sono i Web Services

□ I Web Services

- possono essere messi in corrispondenza e composti
 - favoriscono l'integrazione di servizi, per creare processi di business completi, con un costo di sviluppo ridotto
 - sia nell'ambito di una singola organizzazione (EAI) che tra diverse organizzazioni (integrazione di e-business)
- questa possibilità è basata sulle capacità di
 - descrivere servizi – WSDL
 - scoprire servizi – ad es., con UDDI
 - invocare servizi – SOAP
 - comporre più servizi – ad es., con BPEL
- queste funzionalità sono fornite agli sviluppatori sulla base di opportuni (e numerosi) standard, basati su XML



* Service-Oriented Architecture

□ 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



Service-Oriented Architecture

- **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



Service-Oriented Architecture

- **SOA – Service-Oriented Architecture** [Papazoglou]
 - lo scopo essenziale di una **SOA** è di abilitare l'interoperabilità tra tecnologie esistenti e l'estensibilità a scopi ed 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 ed abilitare piani strategici e livelli di produttività come richiesto da ambienti di business competitivi



- SOA e qualità

- SOA è uno stile architetturale che si propone di perseguire 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
 - stesso modello architetturale per l'integrazione di applicazioni interne ed esterne



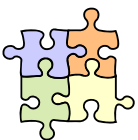
SOA e qualità

- SOA è uno stile architetturale che si propone di perseguire 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
 - stesso modello architetturale per l'integrazione di applicazioni interne ed esterne



SOA e qualità

- SOA è uno stile architetturale che si propone di sostenere agilità di business ed altri obiettivi di qualità – a tal fine, lo stile SOA è basato su numerosi principi di progettazione
 - l'uso di WS come tecnologia 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 – 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
 - ...

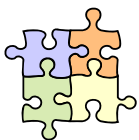
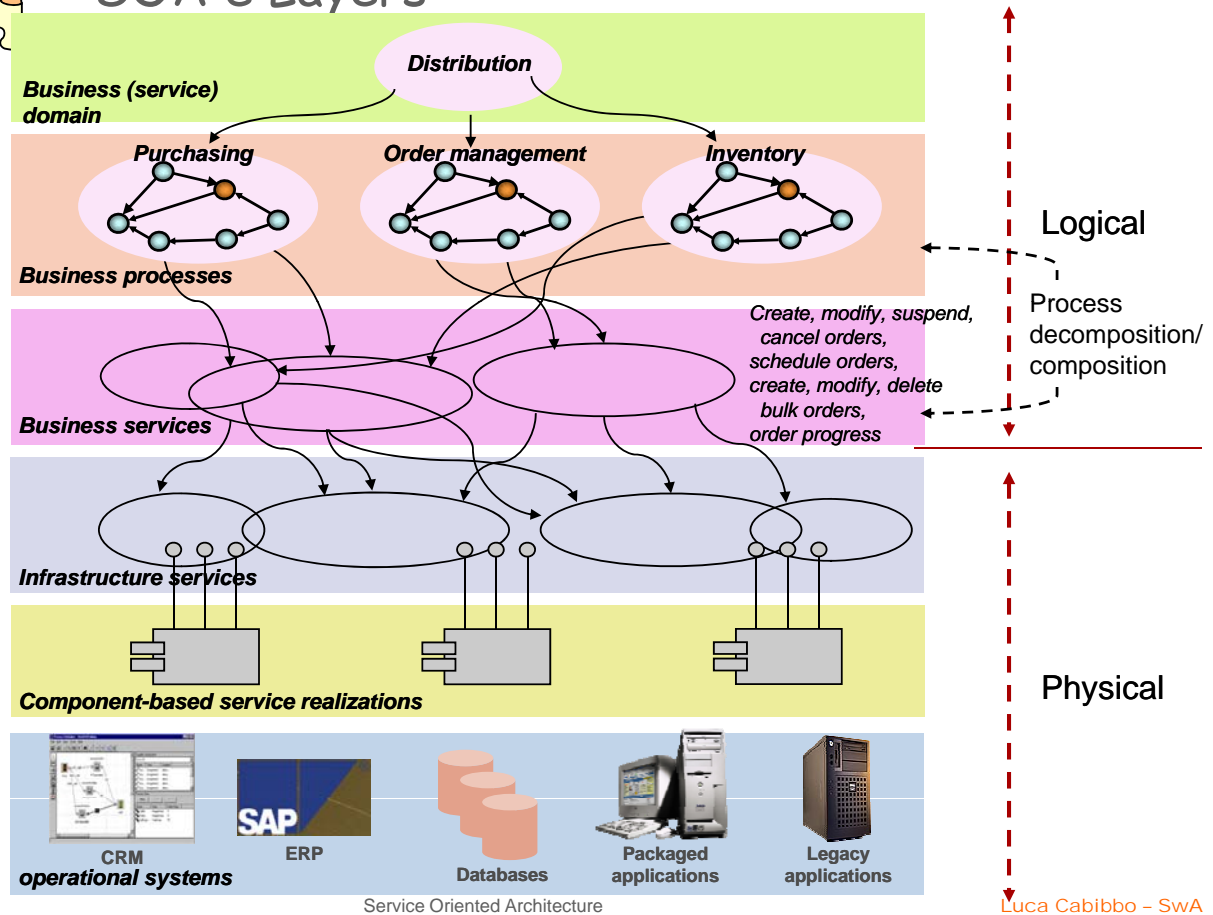


- SOA e Layers

- 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]



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

□ Processi di business

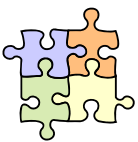
- ciascun dominio di business è composto da un certo numero di **processi di business** fondamentali
 - ad es., erogazione di un mutuo per l'acquisto di una casa
- ciascun processo di business sarà ottenuto come (ri)composizione di un certo numero di servizi di business



SOA e Layers

□ Servizi di business

- un **servizio di business** rappresenta un compito di business automatizzato ed elementare (non ulteriormente decomponibile) – 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

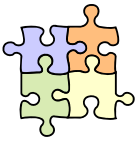
□ Servizi infrastrutturali

- servizi tecnici – abilitano sviluppo e gestione di servizi e processi di business
- servizi d'accesso – per trasformare dati e integrare funzioni o applicazioni pre-esistenti nell'ambiente SOA
- servizi di gestione e monitoraggio
- servizi di interazione – supportano l'interazione tra applicazioni e utenti finali

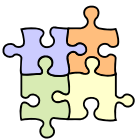
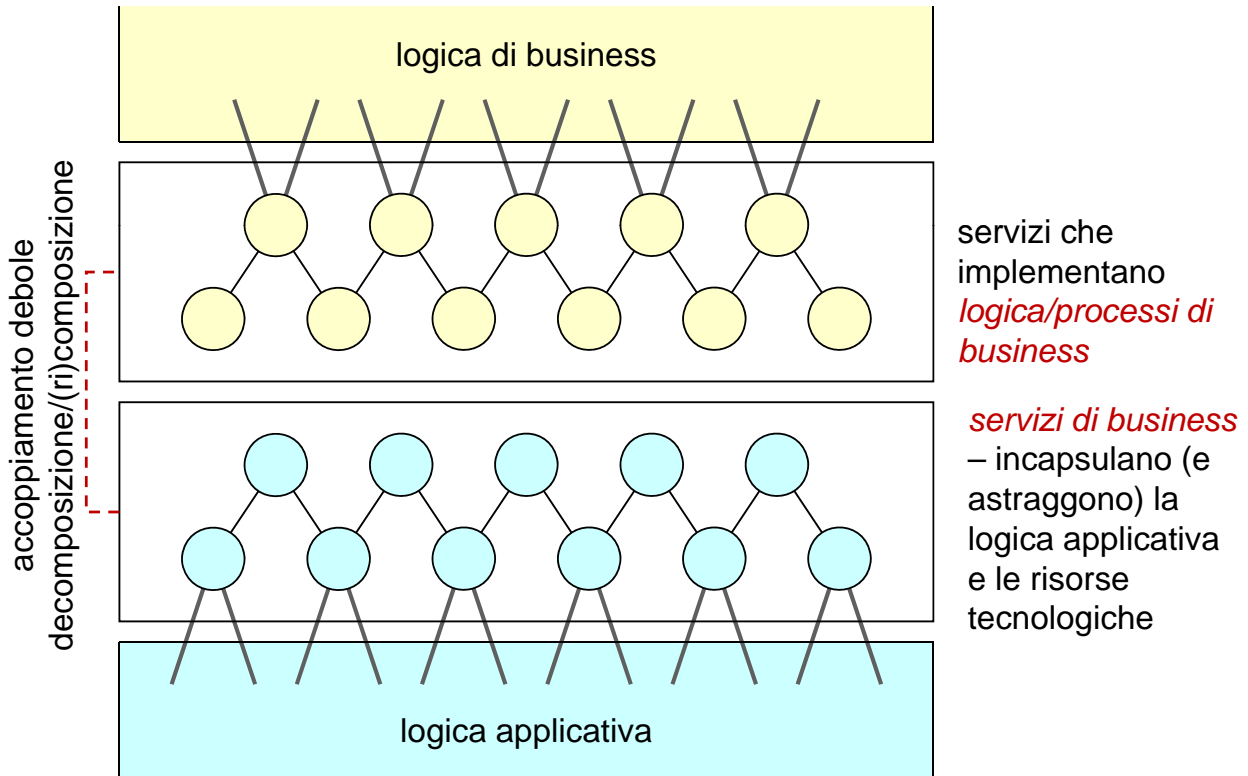
□ Realizzazione dei servizi basata su componenti

- i **componenti** sono unità autonome di funzionalità – rappresentano una demarcazione naturale del lavoro di implementazione
- spesso adattatori verso sistemi pre-esistenti

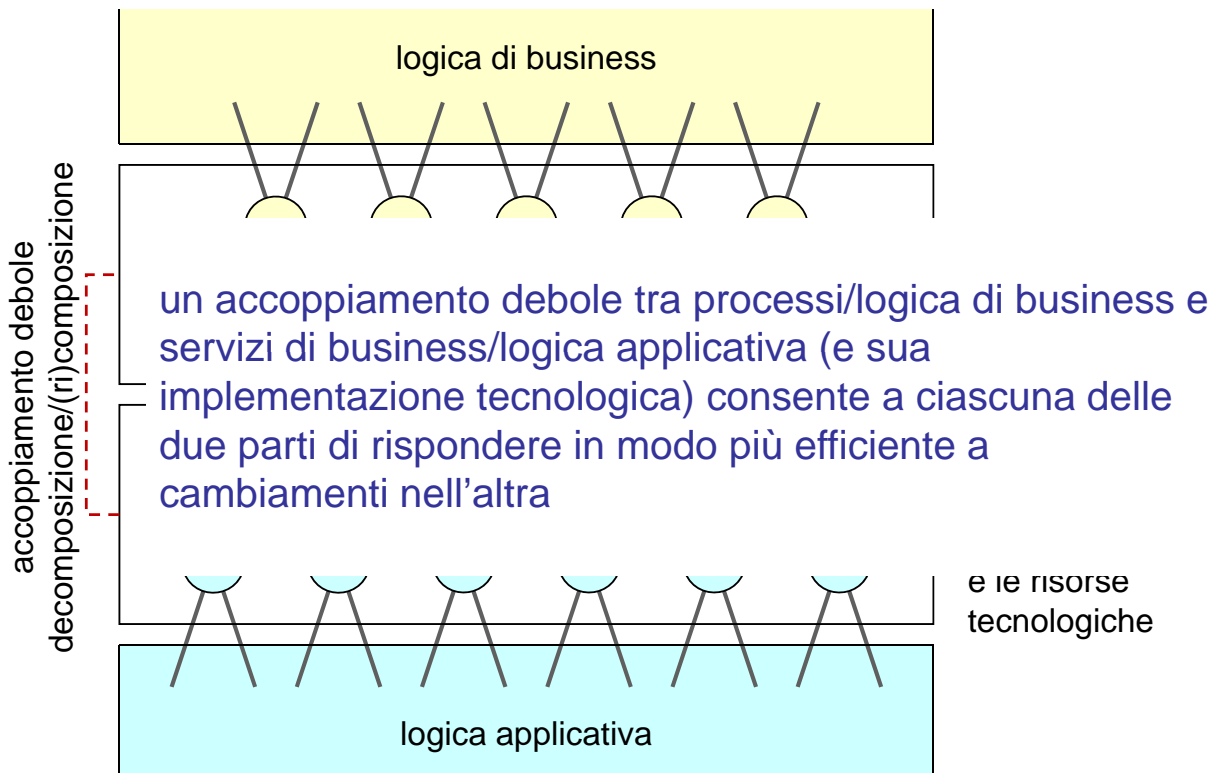
□ Sistemi operazionali



Organizzazione a strati e agilità di business



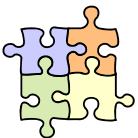
Organizzazione a strati e agilità di business





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
 - questo ha impatto sulle proprietà richieste per i servizi che contribuiscono alla definizione di una SOA



- Proprietà dei servizi in una SOA

- I servizi, realizzati come Web Services, sono gli elementi costitutivi per la creazione di applicazioni distribuite nel paradigma orientato ai servizi
 - sono basati su standard aperti
 - possono essere composti, per realizzare processi di business completi
- Ma quali proprietà caratterizzano i servizi in una SOA? E caratteristiche devono avere i servizi di business per poter essere composti in modo efficace?



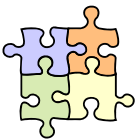
Proprietà dei servizi in una SOA

- Alcuni principi di progettazione fondamentali in una SOA
 - accoppiamento debole – minimizzare le dipendenze tra servizi
 - contratti espliciti (buona definizione) – i servizi aderiscono a vincoli imposti dalle descrizioni (interfacce) dei servizi
 - astrazione – il contratto è esposto – la logica interna è nascosta
 - autonomia – i servizi hanno il controllo della logica che incapsulano – e sono indipendenti dagli altri servizi
 - riusabilità – i servizi sono progettati per massimizzare il riuso
 - componibilità – collezioni di servizi possono essere coordinati e assemblati per formare servizi composti
 - scopribilità – i servizi sono progettati per essere “trovati” mediante degli opportuni meccanismi di ricerca
 - assenza di stato (statelessness) – minimizzare la memorizzazione dello stato conversazionale con i propri clienti
 - granularità – dei servizi e dei parametri

35

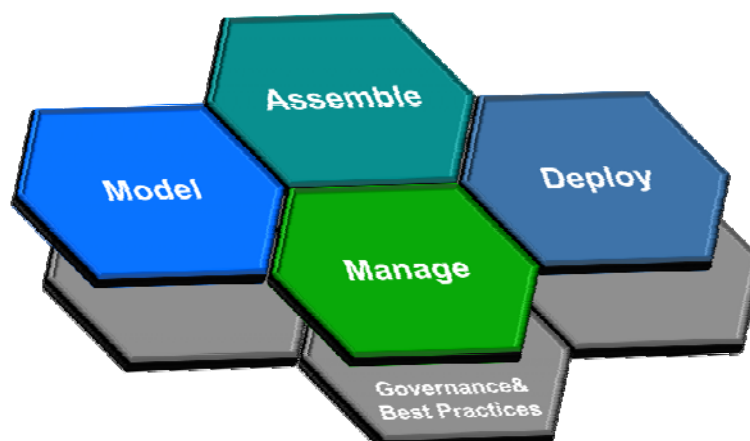
Service Oriented Architecture

Luca Cabibbo – SwA



- Ciclo di vita in una SOA

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



36

Service Oriented Architecture

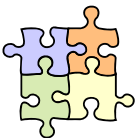
Luca Cabibbo – SwA



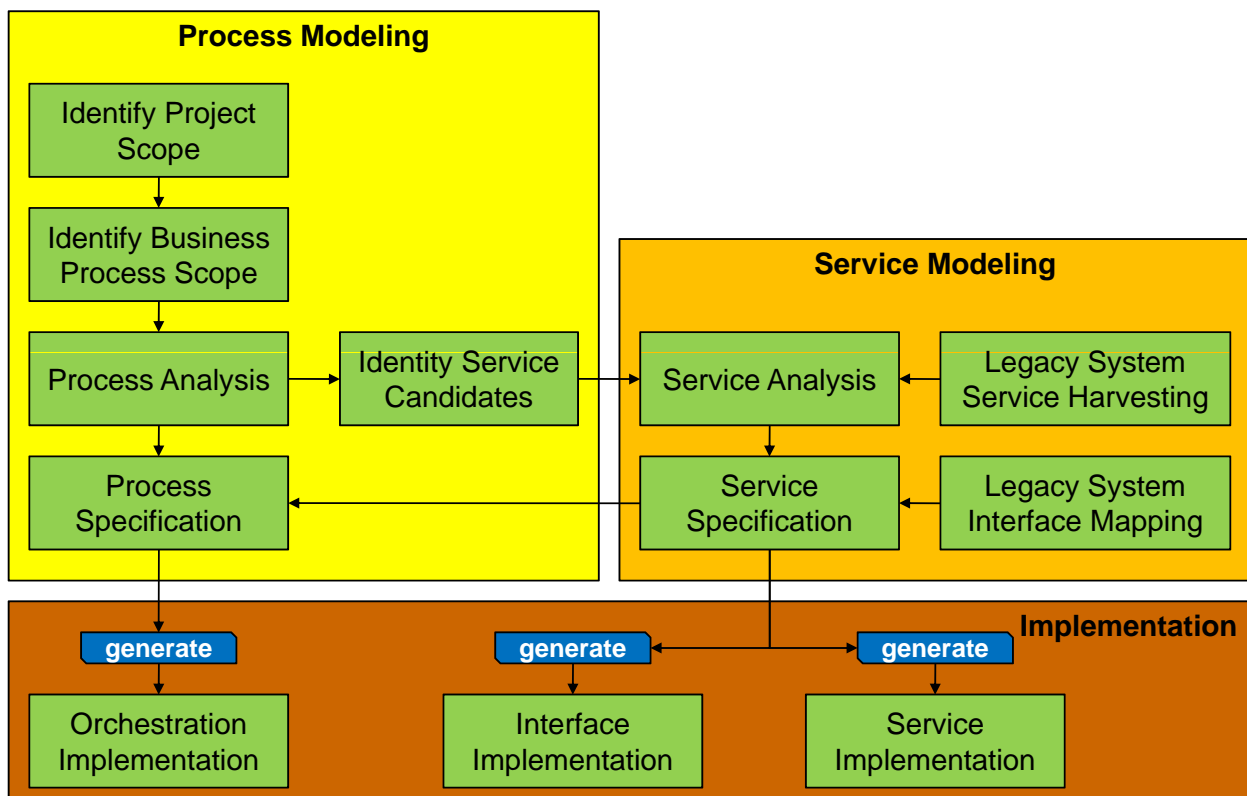
Ciclo di vita in una SOA

□ Ciclo di vita SOA

- model
 - trova i requisiti (di business) – modella (il business – non l'applicazione) e simula – progetta (i servizi)
- assemble
 - scopri – costruisci e testa – componi
 - (il green-field development è raro)
- deploy
 - essenzialmente integrazione – di persone, processi e informazioni
- manage
 - gestisci (i servizi e i processi di business – non le applicazioni) – monitora (metriche di business)



Ciclo di sviluppo - semplificato





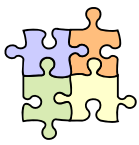
* Enterprise Service Bus

- Uno degli obiettivi delle architetture orientate ai servizi è favorire l'interoperabilità e l'integrazione di applicazioni e componenti sviluppati autonomamente
 - è pertanto necessaria un'infrastruttura di comunicazione distribuita che sostenga l'interoperabilità tra le tecnologie – in modo flessibile e scalabile
- L'*Enterprise Service Bus* è una possibile infrastruttura che sostiene la realizzazione e il deployment di SOA
 - ESB affronta le problematiche di deployment delle SOA – sicuramente significative in presenza di servizi realizzati in ambienti distribuiti/eterogenei
 - consente così di affrontare separatamente gli aspetti “funzionali” da quelli di deployment



Enterprise Service Bus

- *Enterprise Service Bus* 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

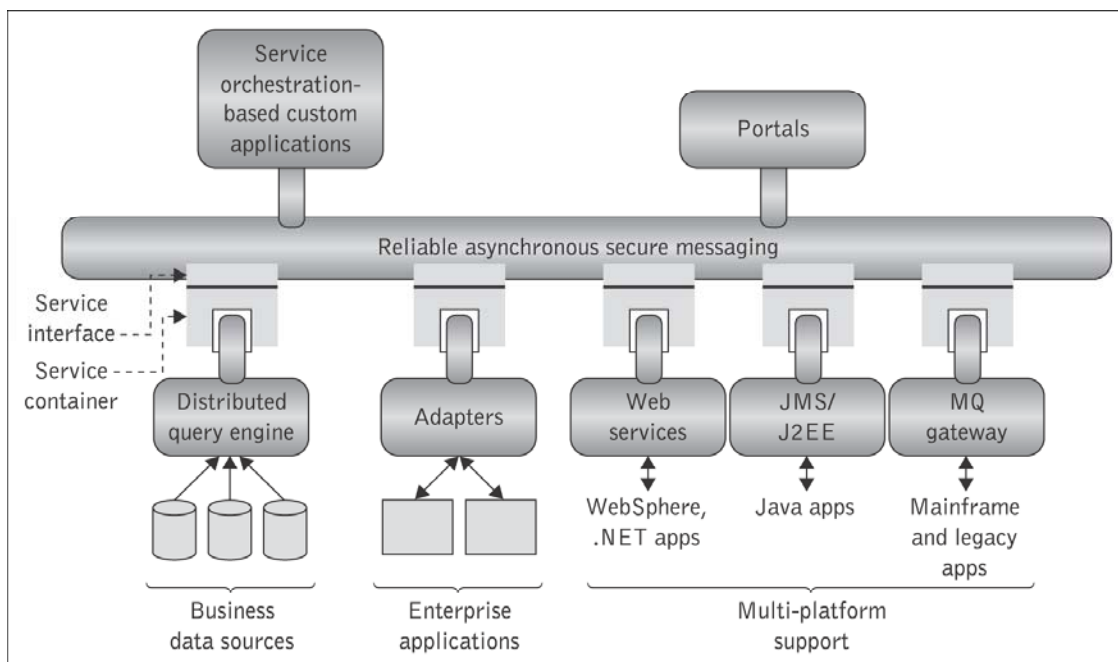


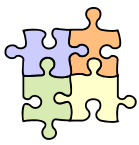
Funzionalità di un ESB

- Funzionalità essenziali di un ESB
 - routing – di messaggi tra consumatori di servizi e servizi
 - conversione – di protocolli di trasporto tra consumatori di servizi e servizi
 - trasformazione – del formato dei messaggi tra consumatori di servizi e servizi
 - distribuzione – di eventi di business
- In pratica, un ESB fornisce un insieme di capacità infrastrutturali
 - consente molteplici modalità di interazione – stile RPC, ORB, MOM, WS, ...
 - introduce facilitazioni a valore aggiunto – affidabilità, sicurezza, validazione e trasformazione di messaggi, routing basato sul contenuto, composizione, ...
 - supporta architetture altamente distribuite ed eterogenee



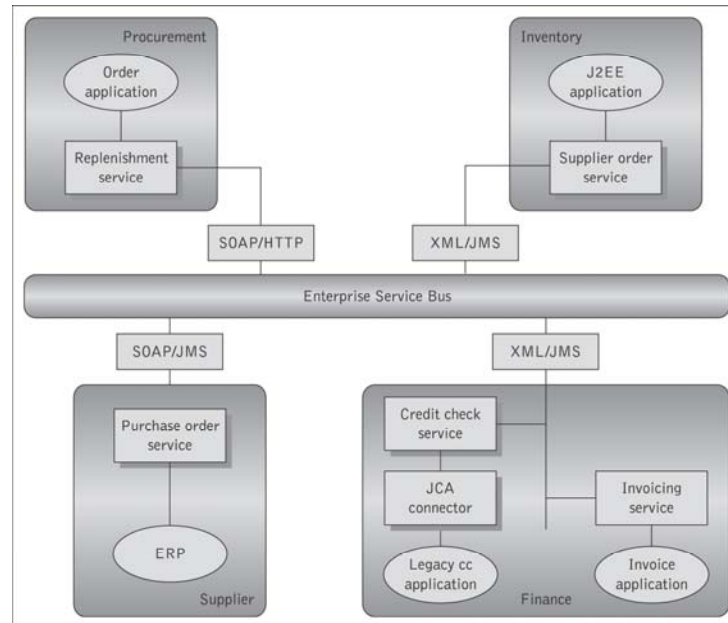
ESB - connessione di diverse tecnologie





ESB - connessione di servizi remoti

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



43

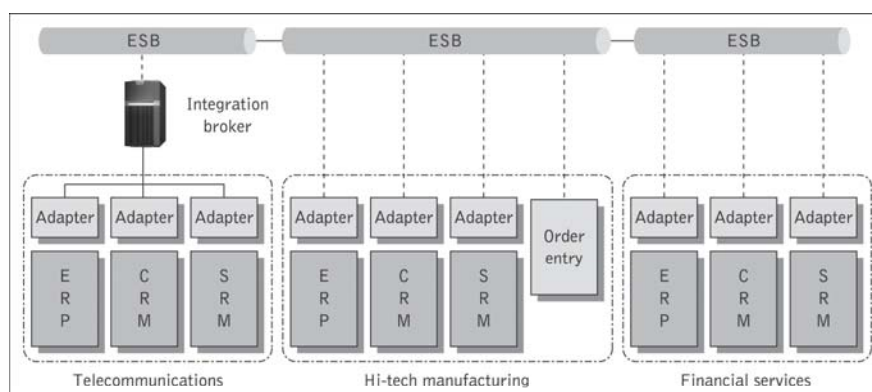
Service Oriented Architecture

Luca Cabibbo - SwA



Possibili implementazioni di un ESB

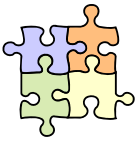
- Sono possibili diverse implementazione alternative per un ESB
 - ESB unico centralizzato
 - ESB come federazione di ESB
- Esempio
 - cooperazione basata su una federazione di ESB distribuiti



44

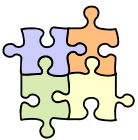
Service Oriented Architecture

Luca Cabibbo - SwA



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 non basate su 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

- 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 progettazione (specifica) dei servizi
 - che peraltro non può essere basata solo su metodologie consolidate come OOAD, CBD e BPM
- Per ulteriori aspetti metodologici relativi alle SOA, si veda il tutorial su
 - Web services lifecycle management