



Imola, march 2007

# gruppoimola



**i**nterpreta

le TECNOLOGIE sul mercato

**i**nteragisce

con il CLIENTE

**i**nnova

i PROCESSI tecnologici del cliente

**i**struisce

con percorsi di SKILL TRANSFER

**i**nforma

con MOKABYTE.it

**Come prepararsi concretamente alla SOA:  
dallo studio di fattibilita' all'integrazione dei sistemi**

**Ing. Claudio Bergamini**

gruppoimola





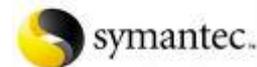
Imola, march 2007



- Piu' di 20 anni di esperienza nella consulenza Enterprise IT a livello di strategia, architettura, implementazione.
- Area finanza, servizi, software.
- Membri o partner tecnologici di:



gruppaimola



Copyright Gruppoimola (2007)





Imola, march 2007

## Chi siamo

- Siamo esperti di Enterprise Architecture con piu' di 5 Roadmap SOA Enterprise in corso da 3 anni fa ad oggi
- Siamo esperti di architettura, progettazione, implementazione di Service Oriented Architecture
- Siamo gli autori di

## **Jbi4cics    Jbi4corba    (jbi4semweb)**

- Sono i componenti standard JBI adottati da
  - **Apache ServiceMix**
- Sono i componenti standard adottati da Sun Microsystem in
  - **NetBeans Enterprise**
  - **OpenEsb**
  - **Libreria OpenJbi Components**

gruppoimola



Imola, march 2007

## Ed inoltre



La prima rivista web italiana dedicata a Java®

MokaByte è la comunità online di architetti e sviluppatori enterprise Java, che fornisce daily news, interviste tecniche con figure chiave dell'industria IT, design patterns, forum di discussione, libri gratuiti, tutorials, ed altro.

MokaByte è nata nel 1996 ed è la più grande comunità indipendente Java in Italia.

sensiblelogic  
SEMANTIC WEB NOW

Sensible Logic S.r.l. è uno spin off accademico nato per applicare, sviluppare e diffondere il tremendo potenziale delle tecnologie Semantic Web. Sensible Logic è composta dal leader mondiali nella ricerca sul Semantic Web e da Imola Informatica.

DBin (OS), la sua creazione 2005, ha ricevuto il 3° premio alla ISWC 2006 negli USA ed è referenziata da TBL come "una delle applicazioni che meglio esprime il futuro del Semantic Web"

gruppoimola



Imola, march 2007

# Agenda

1. **Asset evaluation**
2. **SOA Maturity Model & Roadmap**
3. **Cosa chiedere ai Fornitori, cosa conviene**
4. **Impatti sulle architetture IT da considerare e superare**
5. **Case History**



Imola, march 2007

## Asset evaluation: uno dei fondamenti di SOA

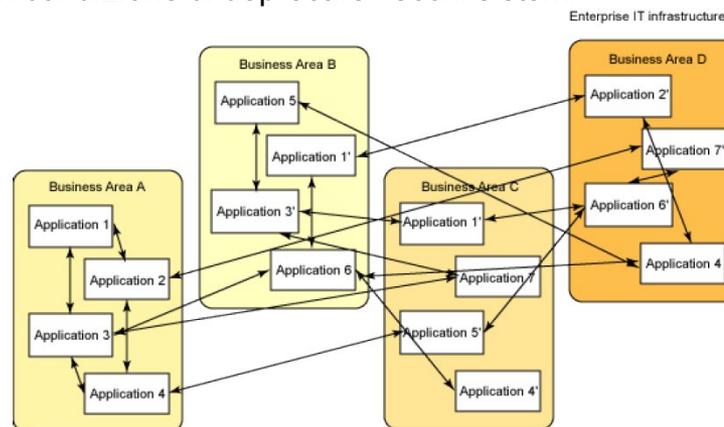
**Nell'IT si sta spesso adottando una filosofia (compra ed integra) contraria a quanto si fa nel resto dell'azienda (compra, consolida, ottimizza e pulisci).**

Ma Service Oriented Architecture e':

- Uso via evoluzione dell'esistente
- Reingegnerizzazione dell'esistente
- Qualificazione dell'esistente

Quindi:

- Occorre fare pulizia quando possibile
- In un certo senso occorre piu' Garbage collection e meno integration
- Occorre mettersi in condizione di deprecare vecchi sistemi





Imola, march 2007

## Cosa e' un Asset

**Una collezione di “artifacts” che fornisce una soluzione per un problema in un contesto dato.**

### **Caratteristiche**

Gli Assets devono includere istruzioni o regole per l'uso.

Servono per minimizzare il tempo che gli sviluppatori usano per

- scoprire,
- analizzare,
- consumare,
- testare l'asset.

Gli Asset descrivono anche il contesto tecnico e business in cui puo' essere usato.

Il tipo di “artifacts” che l'asset contiene dipende dal contesto d'uso.



Imola, march 2007

## Cosa e' un Asset

Per un contesto di sviluppo un Asset puo' contenere **requisiti, modelli, source code, e test.**

Chi costruisce servizi puo' includere questo tipo di artifacts per aiutare gli sviluppatori a usare i servizi in modo efficiente.

Ci possono essere punti di variabilita' per personalizzare l'asset usando parametri.

Un asset e' quindi molto simile a un pattern. Infatti il concetto dietro a entrambi e' di mettere insieme contesto, problema e soluzione.

Un asset e' un concetto piu' generale di un pattern. I punti di variabilita' di una asset sono a livello di artifact, mentre in un pattern ci sono oltre ai parametri i "participants" cioe' dei punti di variabilita' che si applicano all'intero pattern, e non ad un artifact particolare.

gruppoimola



Imola, march 2007

## Reusable Asset Specification

Come organizziamo e strutturiamo un asset?

Che informazioni dobbiamo conoscere sull'asset?

La **Reusable Asset Specification (RAS)** fornisce la struttura.

E' uno standard di Object Management Group (OMG), definito nel 2005.

Ci sono diversi tipi di “software development artifacts”, con varie forme e vari stili.

Specificare come organizzare struttura, descrizione e packaging degli artifacts, **RAS fornisce consistenza e predicibilità attraverso i vari asset**, cosa che riduce in modo significativo i costi di asset management e di utilizzo.

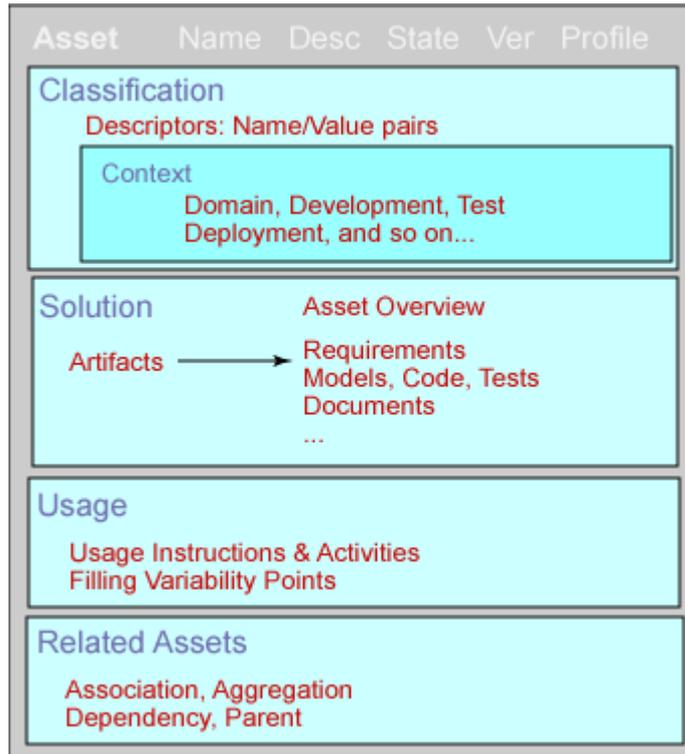
Utilizzando RAS, si puo' confezionare ogni asset con un wrapper di meta-data che include le informazioni di alto livello sull'asset.

gruppoimola



# Reusable Asset Specification

Imola, march 2007



## Reusable Asset Specification

**Classification:** facilita ricerca e browsing

**Solution:** descrive la collezione degli artifacts

**Usage:** guida su come si applica e personalizza. Alcune possono essere dotate di scripts e wizards, memorizzate nella sezione "solution".

**Related:** definisce le relazioni con altri assets e aiuta a creare collezioni e famiglie di assets per avere soluzioni a grana piu' grossa.

RAS ha oggi tre possibili "profile".

- the Default profile,
- the Default Component profile
- the Default Web Service profile



Imola, march 2007

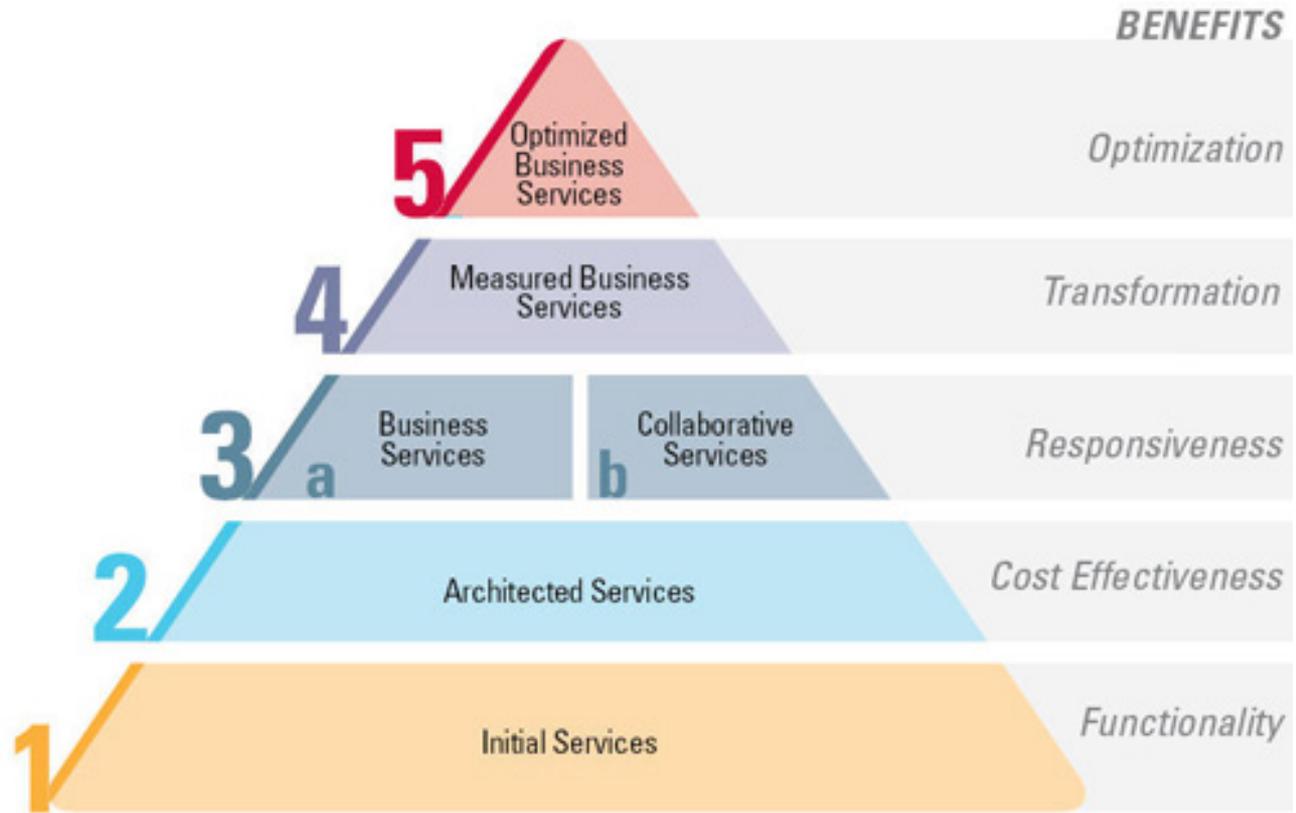
# Agenda

1. Asset evaluation
2. SOA Maturity Model & Roadmap
3. Cosa chiedere ai Fornitori, cosa conviene
4. Impatti sulle architetture IT da considerare e superare
5. Case History



# SOA Maturity Model (SOA-MM)

Imola, march 2007



I livelli del SOA-Maturity Model (Sonic Software, Systinet, AmberPoint e BearingPoint)



Imola, march 2007

## I livelli di SOA-MM

### **Primo livello - Initial Services:**

- Livello più semplice
- Limitato ad esportare come servizi alcune funzionalità IT (progetti pilota, "confinati" per lo più nell'ambito di sperimentazione all'interno dell'IT)
- Si inizia a prendere confidenza con gli standard di prima generazione come XML, XSLT, SOAP e WSDL



Imola, march 2007

## I livelli di SOA-MM

### **Secondo livello - Architected Services**

- Il coinvolgimento ai principi SOA si allarga a tutto lo staff IT e si iniziano a definire gli obiettivi, acquisire competenza ed a strutturare il Team di conseguenza
- Si analizzano gli standard di seconda generazione (UDDI, BPEL4WS, WS-Security, ...) e si individuano le tecnologie e prodotti da utilizzare e le linee guida da seguire
- A tale livello si ha una "sponsorizzazione" da parte del CIO

gruppoimola



Imola, march 2007

## I livelli di SOA-MM

### **Terzo livello - Business Services / Collaborative Services**

- L'implementazione SOA esce dall'ambito IT per coinvolgere anche i responsabili delle linee di business
- E' il primo passo verso l'"allineamento" tra IT e business



Imola, march 2007

## I livelli di SOA-MM

### **Quarto livello - Measured Business Services**

- Si ha ormai un'architettura SOA ben delineata con strumenti di Business Activity Monitoring e di Event Stream Processing
- Con gli strumenti a disposizione è possibile collezionare i dati di business, valutare il ROI del progetto ed identificare eventuali punti da migliorare



Imola, march 2007

## I livelli di SOA-MM

### **Quinto livello - Measured Business Services**

- Nel quinto livello l'architettura SOA è ormai consolidata e rodada
- Il CEO e l'IT sono "sintonizzati" e condividono la mission aziendale
- L'obiettivo finale è quindi l'ottimizzazione dei processi e dei servizi di business



Imola, march 2007

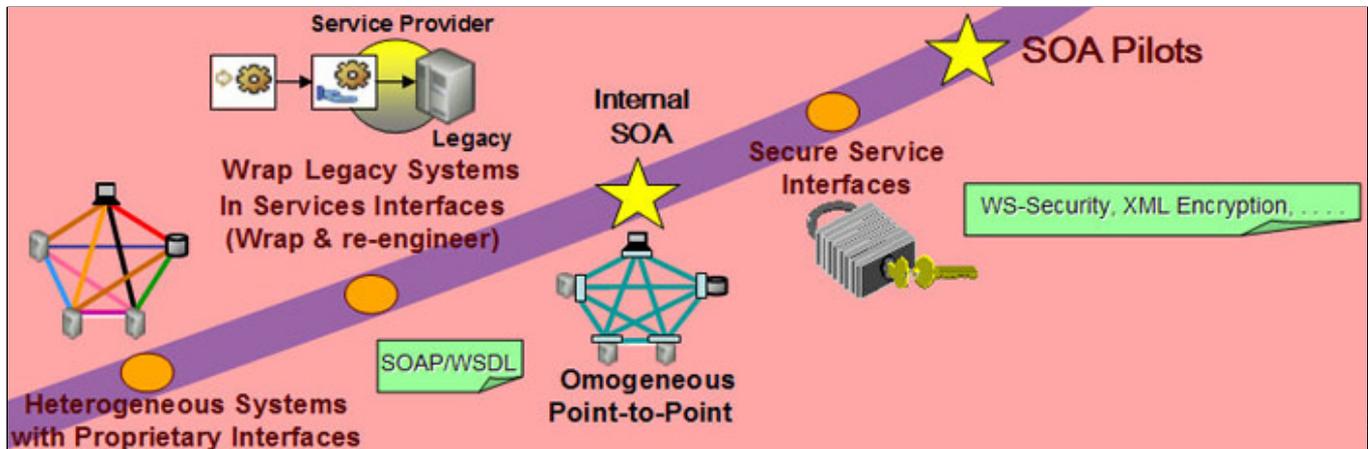
## Initial Services

### P2P integration

- semplificare i problemi tecnologici d'integrazione
- razionalizzare il modus-operandi d'integrazione (Service Wrappers / Service Adapters)
- rimpiazzare le API proprietarie con API standard-based

**A questo punto è possibile introdurre componenti infrastrutturali ed affrontare problematiche tecnologiche (es. introduzione della gestione della sicurezza per l'accesso ai servizi)**

**Prima versione prototipale di SOA (SOA Pilots), utilizzata per verificare l'architettura e dimostrare al management che i principi enunciati siano effettivamente utilizzabili**



gruppoimola



sensiblelogic



Imola, march 2007

## Architected Services

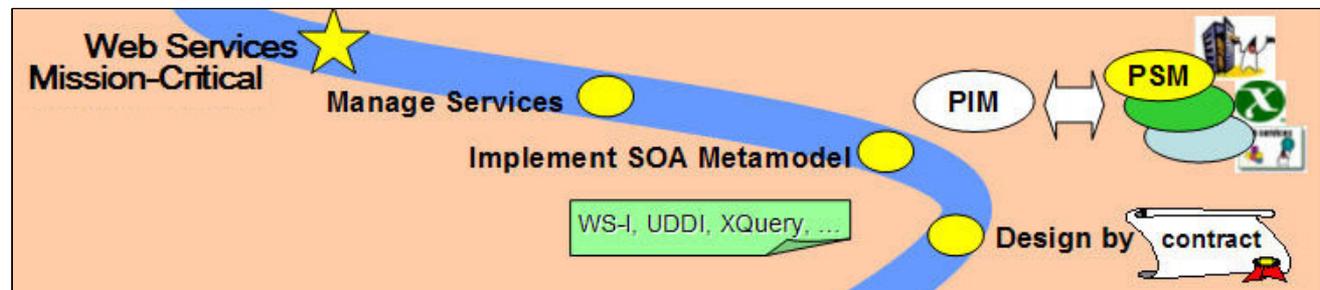
**Si comincia ad introdurre il Service Model, che permette di definire i requisiti e l'interfaccia delle implementazioni tecniche (Service Contract)**

**Si inizia a comprendere l'importanza di disegnare servizi “coarse-grained” e debolmente accoppiati con il service consumer. L'acquisizione di questi concetti è poi di supporto a tutta l'attività dei livelli successivi della roadmap.**

**Di fatto si inizia ad introdurre il linguaggio comune con cui l'organizzazione può descrivere il suo sistema informativo e ad “istituzionalizzare” i concetti di SOA a livello organizzativo.**

**Si iniziano ad individuare delle categorie di servizi infrastrutturali progettati ed implementati a supporto dei servizi di business (es. la sicurezza/autorizzazione, logging, alerting, ...)**

**Alla fine di questa parte del percorso saranno stati definiti dei servizi mission-critical supportati dall'infrastruttura e gestibili a livello organizzativo**





Imola, march 2007

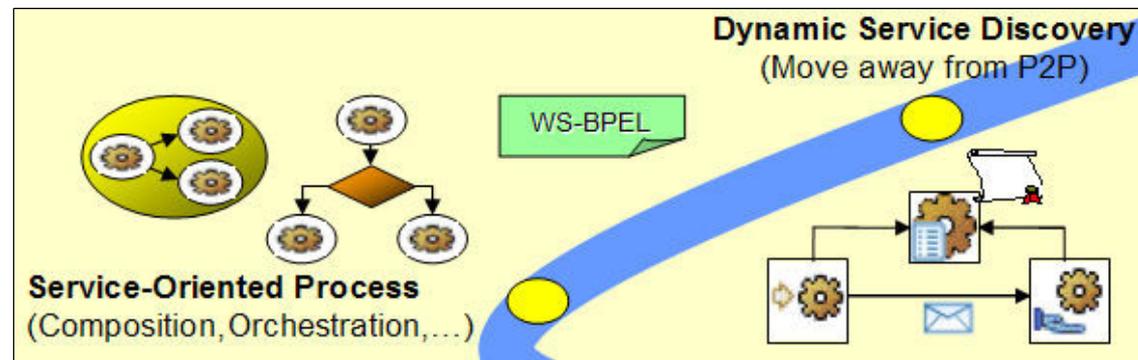
## Business Services

**Il passo successivo è l'introduzione di strumenti per la gestione dei processi. Da questo momento, le applicazioni vengono disegnate (o ristrutturate) come orchestrazione di servizi, a loro volta eventualmente costruiti componendo altri servizi**

**Si comincia ad avere un forte ritorno sull'adozione di SOA, si comincia a praticare il riuso**

**Analisti e Architetti lavorano in sinergia utilizzando il concetto di servizio, i concetti SOA sono diffusi a tutti i livelli IT**

**Si può abbandonare il concetto di servizio “statico” muovendosi da un'integrazione punto a punto a una integrazione dinamica (cioè: ricerca di un service provider da parte di un service consumer). Il consumer “cerca” il servizio pubblicato nel Service Registry per poi utilizzarlo**





Imola, march 2007

## SOA Nirvana

**Si può affrontare il problema dell'integrazione a livello dei dati superando la necessità di conoscere a priori la semantica dell'informazione che il sistema deve elaborare introducendo il concetto di ontologia**

- **Un'ontologia (definita in Web Ontology Language, uno standard W3C) permette di definire un modello con relazioni e regole con cui è possibile "catturare" le relazioni semantiche tra i dati**
- **La descrizione del data model tramite ontologie può permettere l'automatizzazione dell'integrazione tra servizi**

**Un'organizzazione a questo livello è in grado di adattarsi velocemente al cambiamento con costo minimo ed integrare il proprio sistema informativo con altre aziende di livello analogo**

**Il riuso è incoraggiato ed il management è in grado di comprendere l'architettura fino ad un dettaglio che in precedenza non possedeva**



gruppoimola



Imola, march 2007

## Roadmap: i fattori determinanti del successo

- Definizione di un percorso operativo incrementale
  - benefici tangibili al raggiungimento di ogni gradino evolutivo
  - obiettivi rispondenti alle capacità operative delle risorse
- Pianificazione del percorso IT nel medio-lungo periodo
  - probabilità di insuccesso contenute
  - garanzie per l'investimento di ogni singola fase

gruppoimola



Imola, march 2007

## Roadmap: quattro aspetti da curare

- **Progetti e Applicativi**

- Priorità guidate dai progetti
- Valutazione dell'impatto evolutivo

- **Processi organizzativi**

- definizione dei rapporti tra consumatori e fornitori dei servizi
- evoluzione delle politiche di gestione e controllo

- **Linee guida architettura applicativa**

- requisiti, standard, architettura di riferimento, modello infrastruttura

- **Costi e benefici**

- definizione di metriche oggettive per valutare il ritorno degli investimenti effettuati

gruppoimola



Imola, march 2007

# Agenda

1. Asset evaluation
2. SOA Maturity Model & Roadmap
3. Cosa chiedere ai Fornitori, cosa conviene
4. Impatti sulle architetture IT da considerare e superare
5. Case History



Imola, march 2007

## Cosa chiedere a un fornitore

### **Di Consulenza**

Trovare il valore Business per poter gestire una Roadmap

### **Di Architettura**

Buona metodologia, modelli, esperienza, Best Practice, Linee Guida

Approcci per evitare i lock-in

Per favore, non presentare alcuna “killer solution”

### **Di Tecnologia**

Prodotti che aiutino a fare le cose oggi piu' facilmente e velocemente

Prodotti che aderiscano agli standard

Prodotti che non vadano oltre le specifiche

Supporto all'utilizzo dei prodotti

### **Di Sviluppo**

Accettare requisiti di architettura

Accettare di vedere oltre la singola applicazione e il singolo progetto

Imparare a usare le linee guida altrui

gruppoimola

**Accettare con sofferenza lo “1 stop shop” (a causa dei rischi)**



Imola, march 2007

## Cosa conviene

Capire, supportati dai consulenti:

- Quali attività' conserveranno valore anche a medio-lungo termine
  - Ad es.: valutazione degli asset
  - estrazione dei servizi
  - creazione dei servizi
  - gestione dei servizi
- Valutazione dei rischi
- Quali attività' presentano rischi a medio-lungo termine
- Quanto investire a medio-lungo termine in infrastruttura orizzontale
- In ogni scelta va considerato se ci sono:
  - Standard o no
  - Specifiche complete o no
  - Vincoli forti dovuti all'esistente o no

gruppoimola



Imola, march 2007

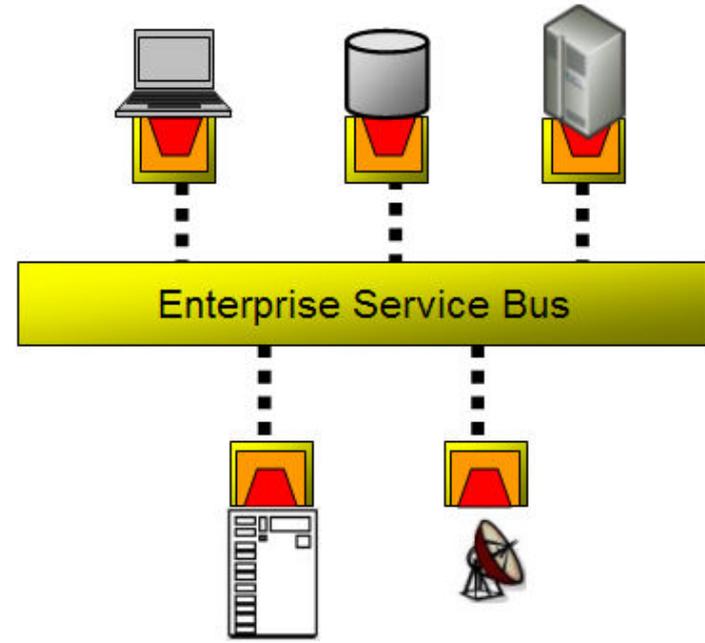
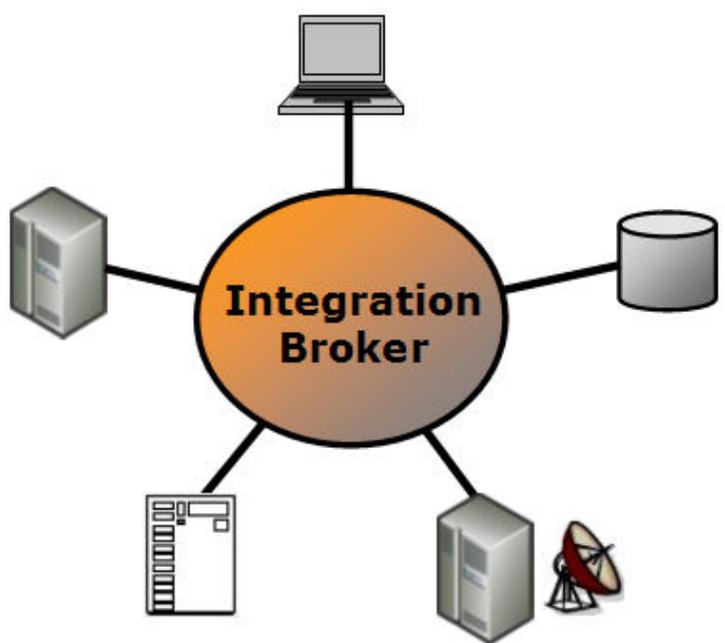
# Agenda

1. Asset evaluation
2. SOA Maturity Model & Roadmap
3. Cosa chiedere ai Fornitori, cosa conviene
4. Impatti sulle architetture IT da considerare e superare
5. Case History



# Architettura It: dagli Integration Broker agli ESB

Imola, march 2007





Imola, march 2007

## “Strategie di wrapping” Vs “Rip and Replace”

In ottica SOA, la strategia di evoluzione suggerita è quella di **isolare i punti di integrazione** delle applicazioni, **disegnare i servizi** di integrazione coerenti con le regole architetturali e **pubblicarli** (es: su un Service Bus)

Il risultato è un servizio:

- estratto dalla logica che si vuole integrare, disegnato per il riuso (e non per un'integrazione ad-hoc)
- disponibile a tutte le applicazioni in quanto “pubblicato” su un middleware di integrazione.

Incapsulando la logica di business esistente in opportuni servizi e condividendoli mediante un Service Bus è possibile riusare in un nuovo layer le funzionalità già sviluppate: questo permette di salvaguardare il progresso (strategia di “wrapping”)



Imola, march 2007

## “Rip and Replace” Vs “Strategie di wrapping”

In antitesi con la strategia di “wrapping”, esiste l’approccio di **radicale sostituzione** (“Rip-and-replace”). Questo approccio è stata la promessa dei sistemi ERP della seconda metà degli anni 90, quando si è tentato di avere un'unica applicazione che coprisse tutti i requisiti del sistema informativo (in linea teorica con un unico sistema si evita la Spaghetti Integration e si consolida l’infrastruttura IT)

Al crescere della complessità dei sistemi informativi il “Rip and Replace” è parzialmente fallito perché è risultato evidente che nei sistemi Enterprise una applicazione difficilmente riesce a coprire più del 40% delle necessità funzionali e implica forti trade-off come la riscrittura massiva di codice nonché nuove problematiche tecniche, sistemistiche e organizzative (budget)



Imola, march 2007

## Possibili strategie di wrapping

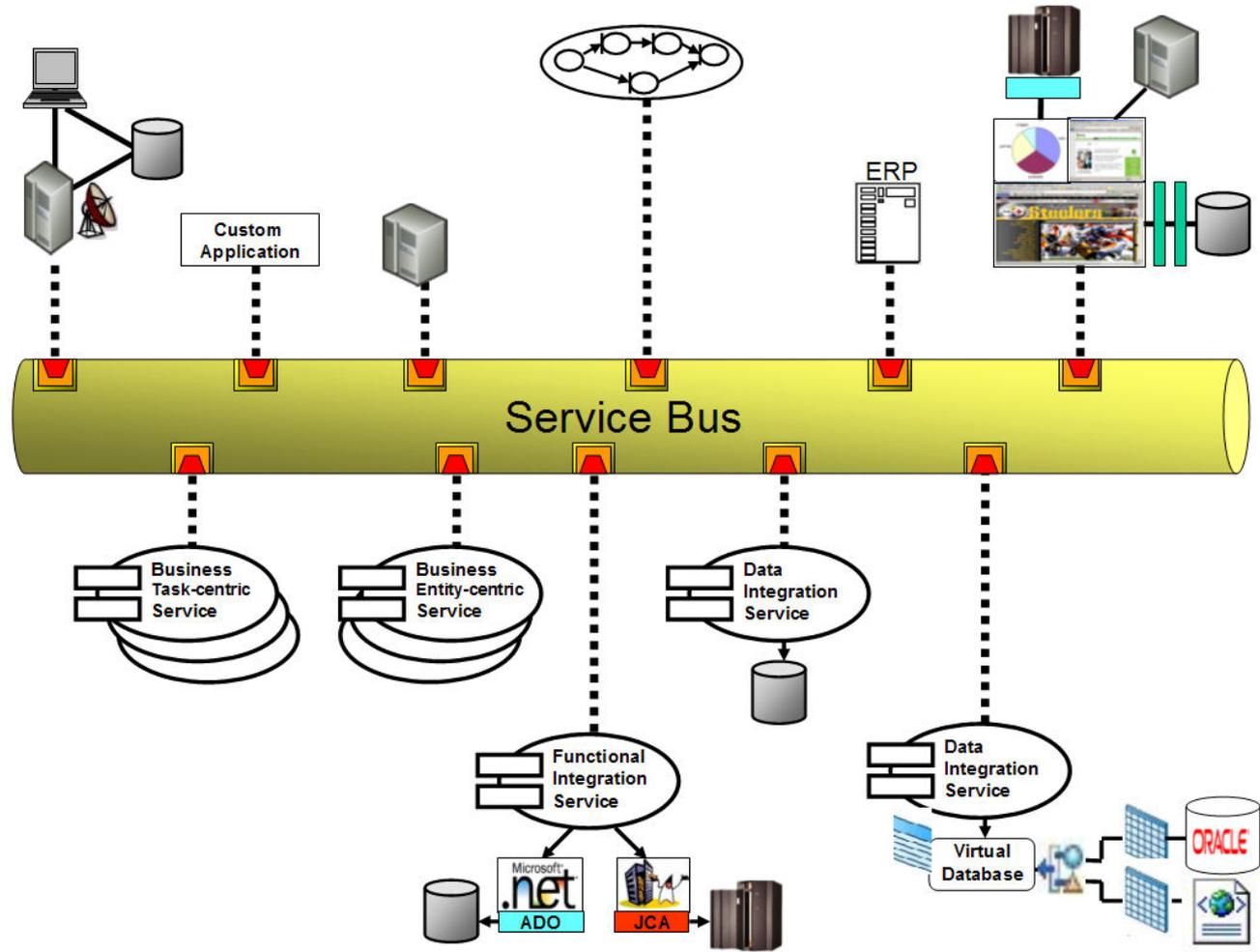
Piuttosto che operare con una strategia di sostituzione “Rip and Replace”, è preferibile attuare strategie di wrapping “Wrap and re-engineer” o “Leave and layer”, che siano più conservative:

- **“Wrap and re-engineer”**: si incapsulano applicazioni legacy con nuove interfacce tecnologiche più flessibili ed interoperabili. Il wrapping di sistemi legacy è ad oggi molto usato: lo si può trovare consigliato come modalità architetturale d’integrazione, oppure presentato come pattern e/o come linea guida “SOA-enabling”
- **“Leave and layer”**: consiste nel conservare l’esistente portfolio dei servizi Legacy e di applicazioni esistenti, riconciliando le loro differenze (logica e dati) in un nuovo strato architetturale. Tale strato permette di riusare l’esistente tramite operazioni di razionalizzazione e consolidamento, mascherando logiche e dati all’esterno dei punti da integrare



# Il disegno complessivo

Imola, march 2007



gruppoimola





Imola, march 2007

## Impatti su Architettura IT: Organizzazione

**Da fare in piu' (non dovrebbe esserlo ma spesso lo e')**

- Inventario dei possibili servizi
- Documentazione dell'esistente
- Definire servizi, interfacce, asset
- Adottare un modello che supera i silos

**Ma tenendo presente**

- Necessita' transitoria di piu' risorse
- Organizzazione che deve evolversi da BU a BU + Service Provider Centralizzato
- Meno sviluppare , piu' progettare

gruppoimola



Imola, march 2007

## Impatti su Architettura IT: Organizzazione

### **Architettura centralizzata**

- Asset Management
- Supporto alla progettazione e evoluzione degli applicativi
- Linee guida sull'utilizzo dei servizi centralizzati da parte degli applicativi

### **Infrastruttura ed esercizio**

- Scalabilità
- Monitoraggio
- Business Activity Monitoring
- Predittivo su Capacity e Performance Management

gruppoimola



Imola, march 2007

## Impatti su Architettura IT: Ruoli

Evolve e si biforca una figura professionale e diventano due

Architettura CASA

Architettura PIANO REGOLATORE (con servizi centralizzati)

Quindi:

Architetto Applicativo

Architetto Enterprise



Imola, march 2007

# Agenda

1. Asset evaluation
2. SOA Maturity Model & Roadmap
3. Cosa chiedere ai Fornitori, cosa conviene
4. Impatti sulle architetture IT da considerare e superare
5. Case History



Imola, march 2007

# Grazie

## **Imola Informatica**

<http://www.imolinfo.it/> (also english version)

<http://gruppoimola.wikidot.com> Wiki

<http://gruppoimola.wordpress.com> Blog

## **MokaByte**

[www.MokaByte.it](http://www.MokaByte.it)

[www.MokaByte-swp.com](http://www.MokaByte-swp.com)

## **SensibleLogic**

[www.SensibleLogic.com](http://www.SensibleLogic.com)

gruppoimola