SOA

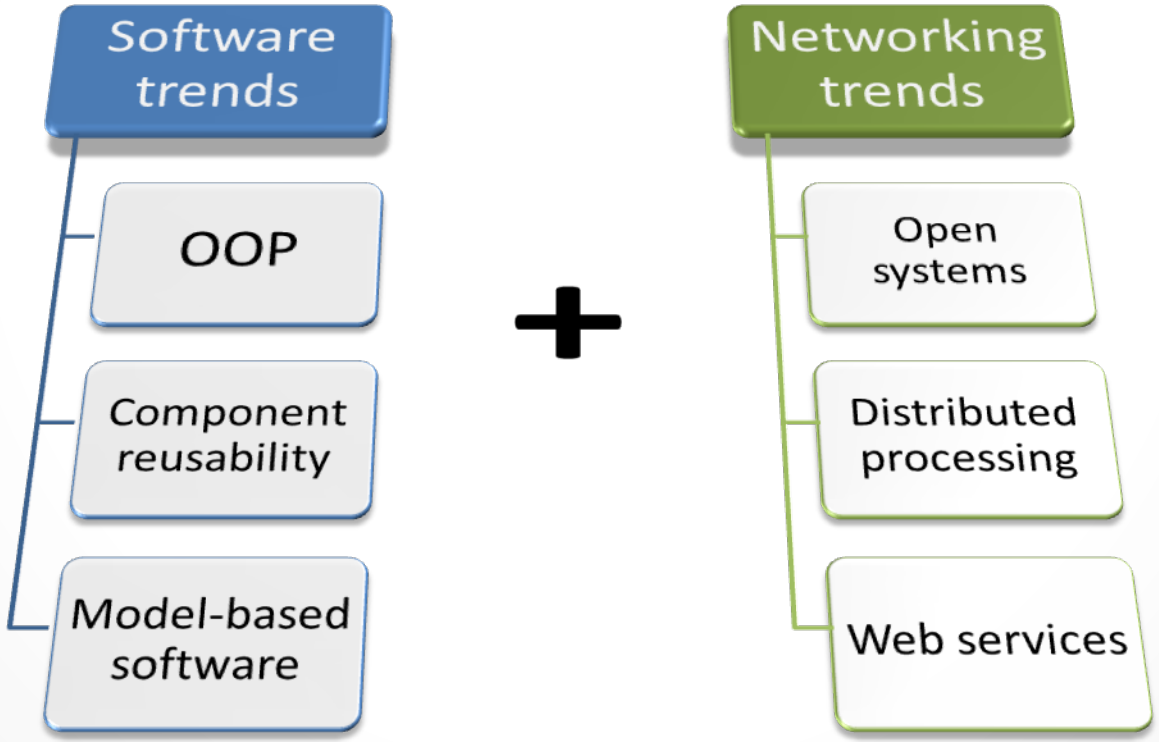
Co je SOA?

*"**S**ervice **O**riented **A**rchitecture"*

"Je kolekce služeb. Tyto služby komunikují mezi sebou. Komunikace využívá předávání dat nebo znamená koordinaci „nějakou" aktivit mezi dvěma více službami. „Nějaká" znamená spojení služeb mezi sebou.

Origin of SOA

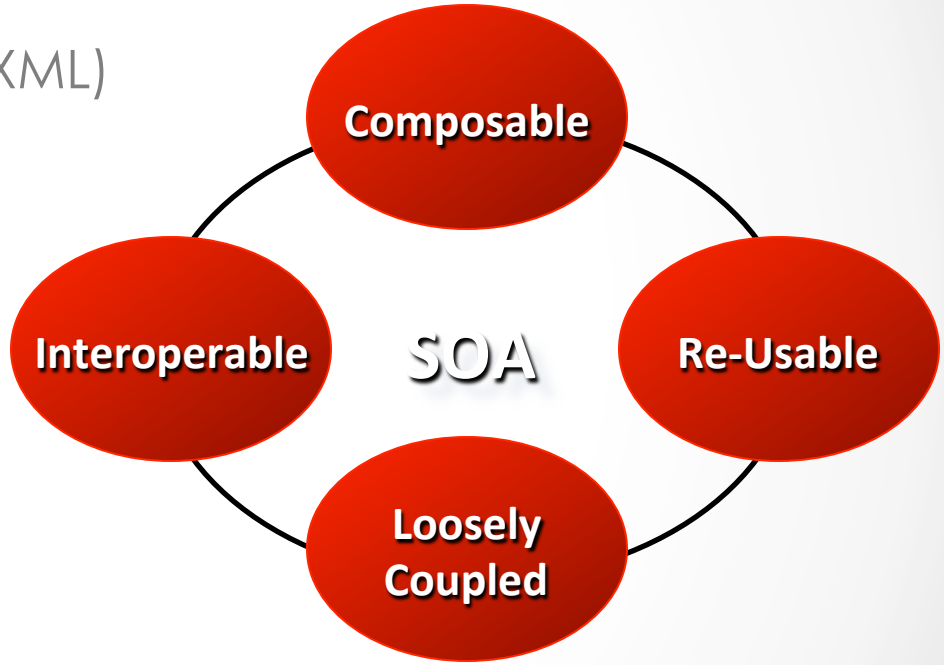
- Výsledek evoluce vývoje software a rozkvětu komunikace (Internet)



SOA

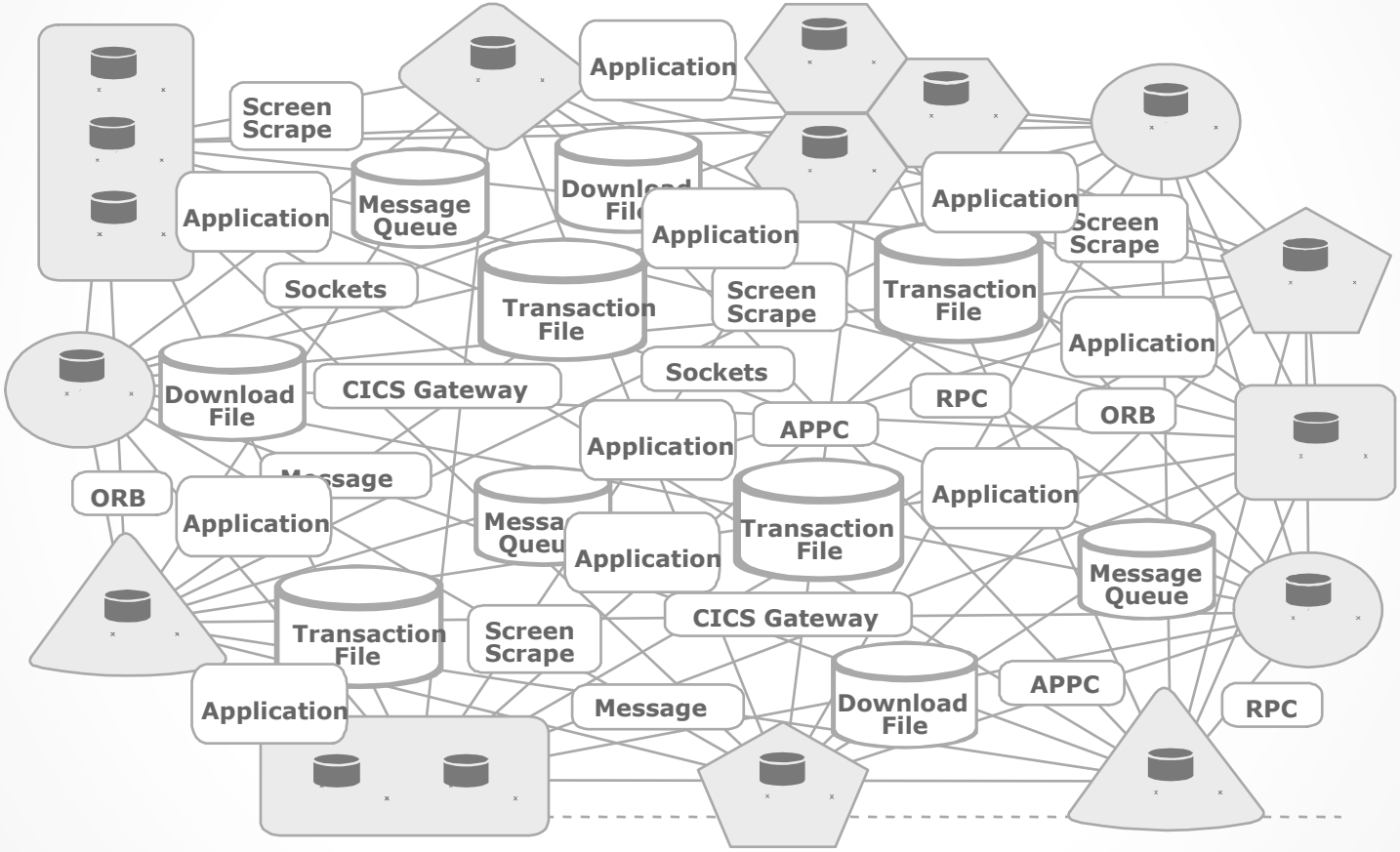
Charakteristika SOA

- Služby mají platformě nezávislé, sama-sebe popisující interfacy (XML)
- Zprávy jsou formálně definovány
- Služby lze nalézt a dohledat
- Služby mají popsané kvality a charakteristiky popsané v pravidlech
- Služby jsou poskytovány platformě nezávisle

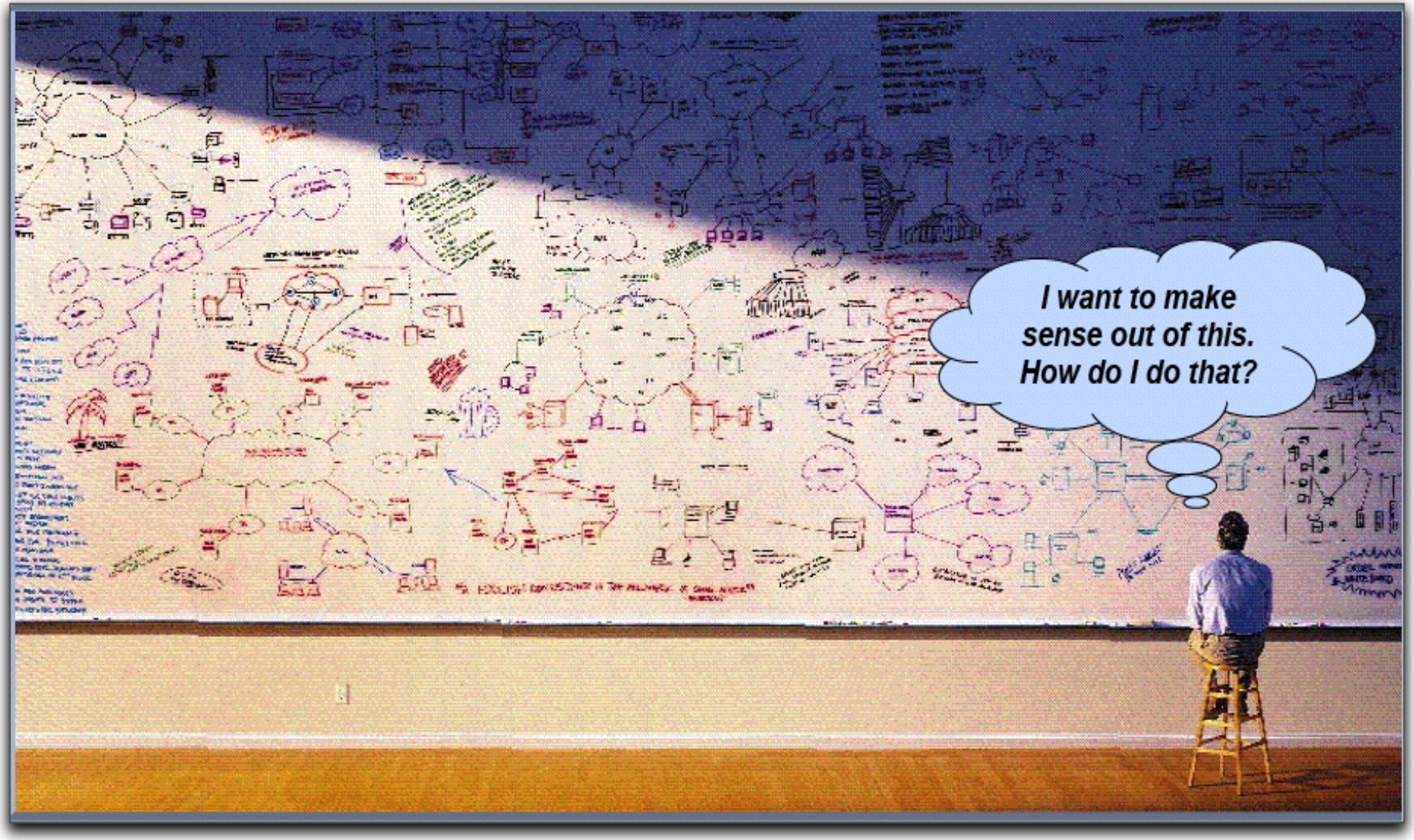


SOA

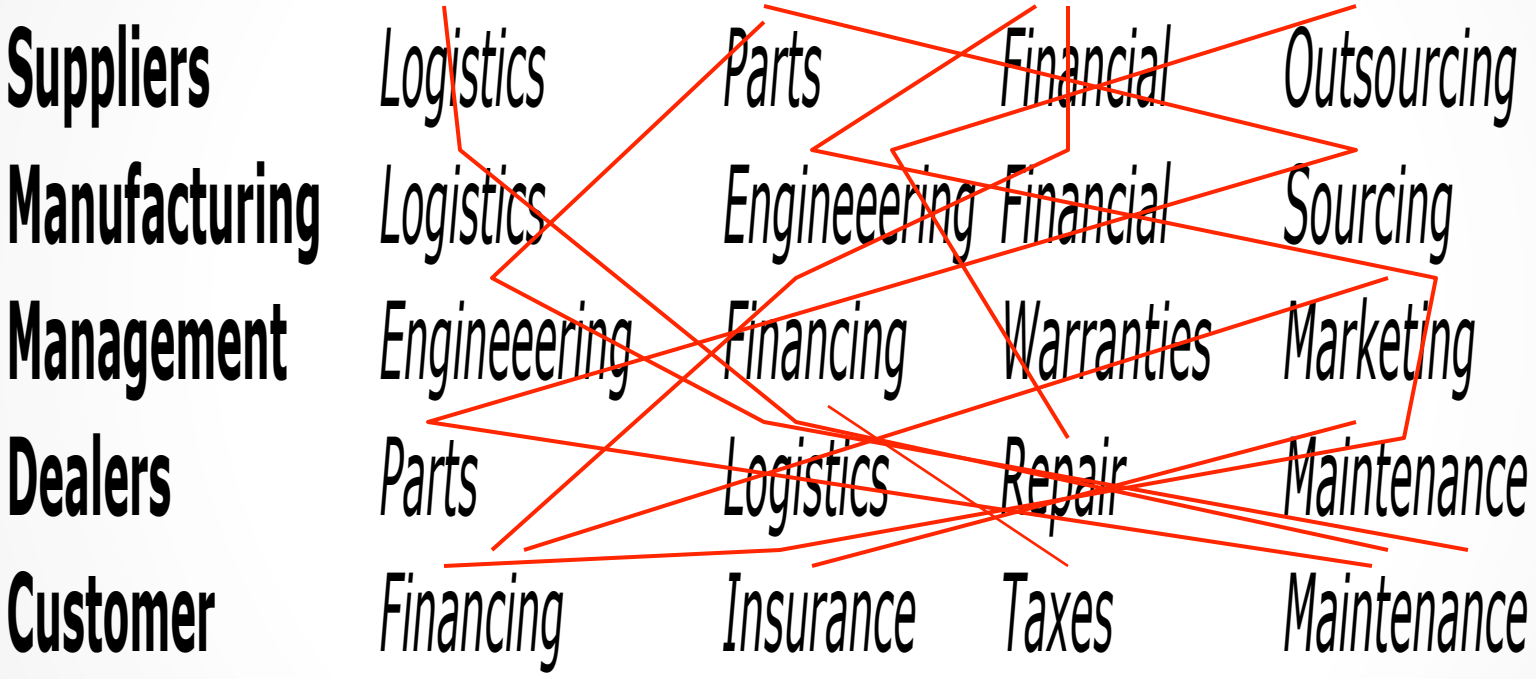
Proč SOA


Application
Screen Scrape
Application
Message Queue
Application
Application
Screen Scrape
Sockets
Transaction File
Screen Scrape
Transaction File
Application
Download File
CICS Gateway
Sockets
RPC
ORB
Application
APPC
Application
ORB
Application
Application
Transaction File
Message Queue
Application
CICS Gateway
Transaction File
Screen Scrape
APPC
Application
Message
Download File
RPC

Problems Addressed by a Service Oriented Architecture

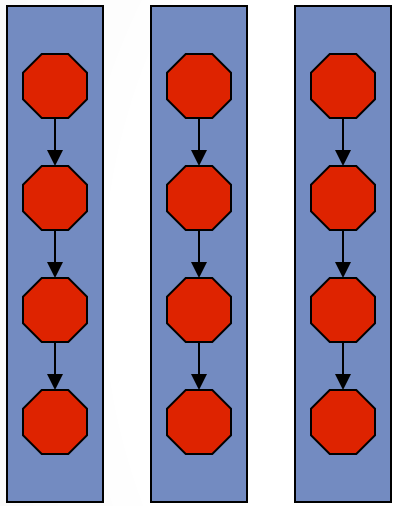


Purpose of Architecture: To Manage Interdependencies



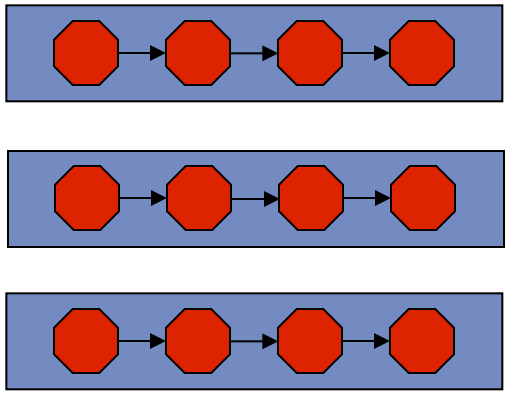
Directions of System Architecture

1960 - 1980



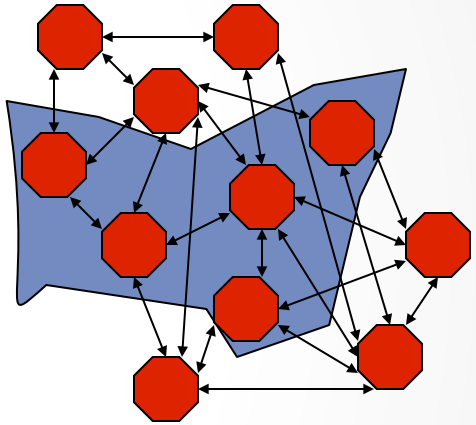
- Organization Focus
- Mainframe Centric
- Internal Use
- Unique Data

1990 - 2000



- Process Focus
- Client Server
- Partial Connectivity
- EDI File Transfer

2010 - 2050



- Distributed Functions
- Data Centric
- Universal Interoperability
- Real-time Connectivity

Co je Service Oriented Architecture (SOA)?

- Metoda návrhu, nasazení a řízení aplikací a softwarové infrastruktury, kde:
  - Veškerý software je organizován do byznys služeb, je přístupný po síti a je vykonatelný.
  - Rozhraní služeb je postaveno na standardech umožňících propojení služeb..

Key Characteristics of SOA

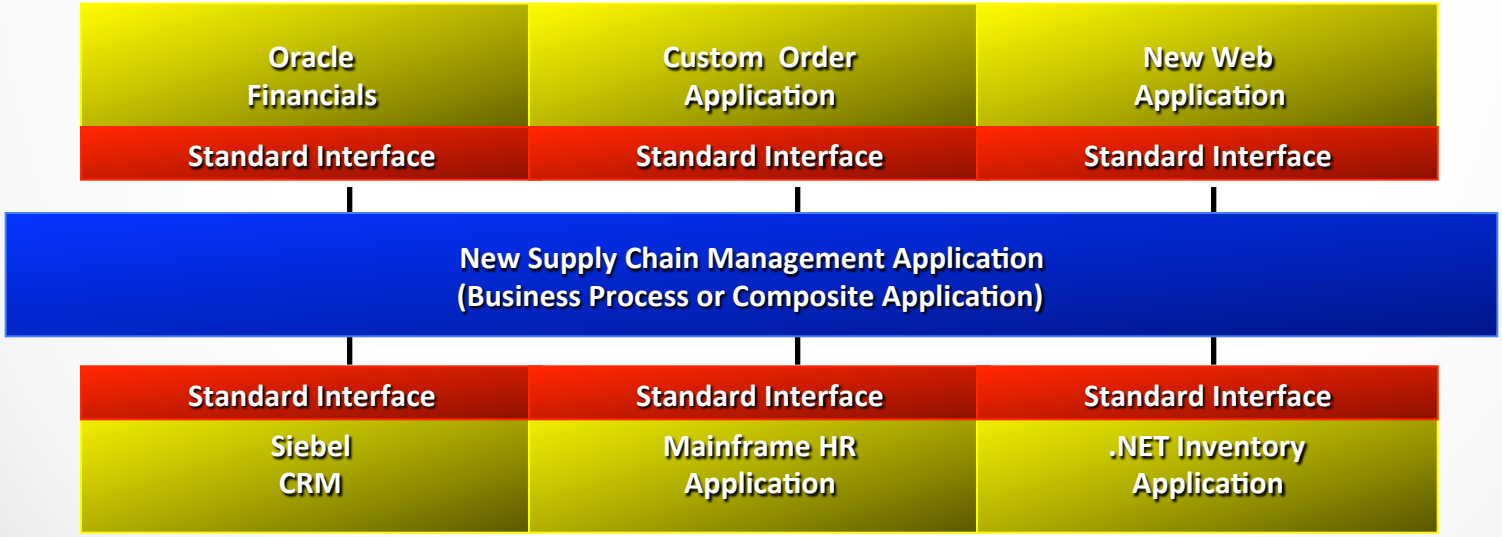
- Dopředu známá specifikace pro
  - Quality of service, security and performance
- Softwarová infrastruktura je zodpovědna za zprávu a řízení.
- Služby jsou dostupné v katalogu a jsou dohledatelné.
- Data jsou dostupné v katalogu a jsou dohledatelné.
- Protokoly jsou postavené na průmyslových standardech.

What is a “Service” ?

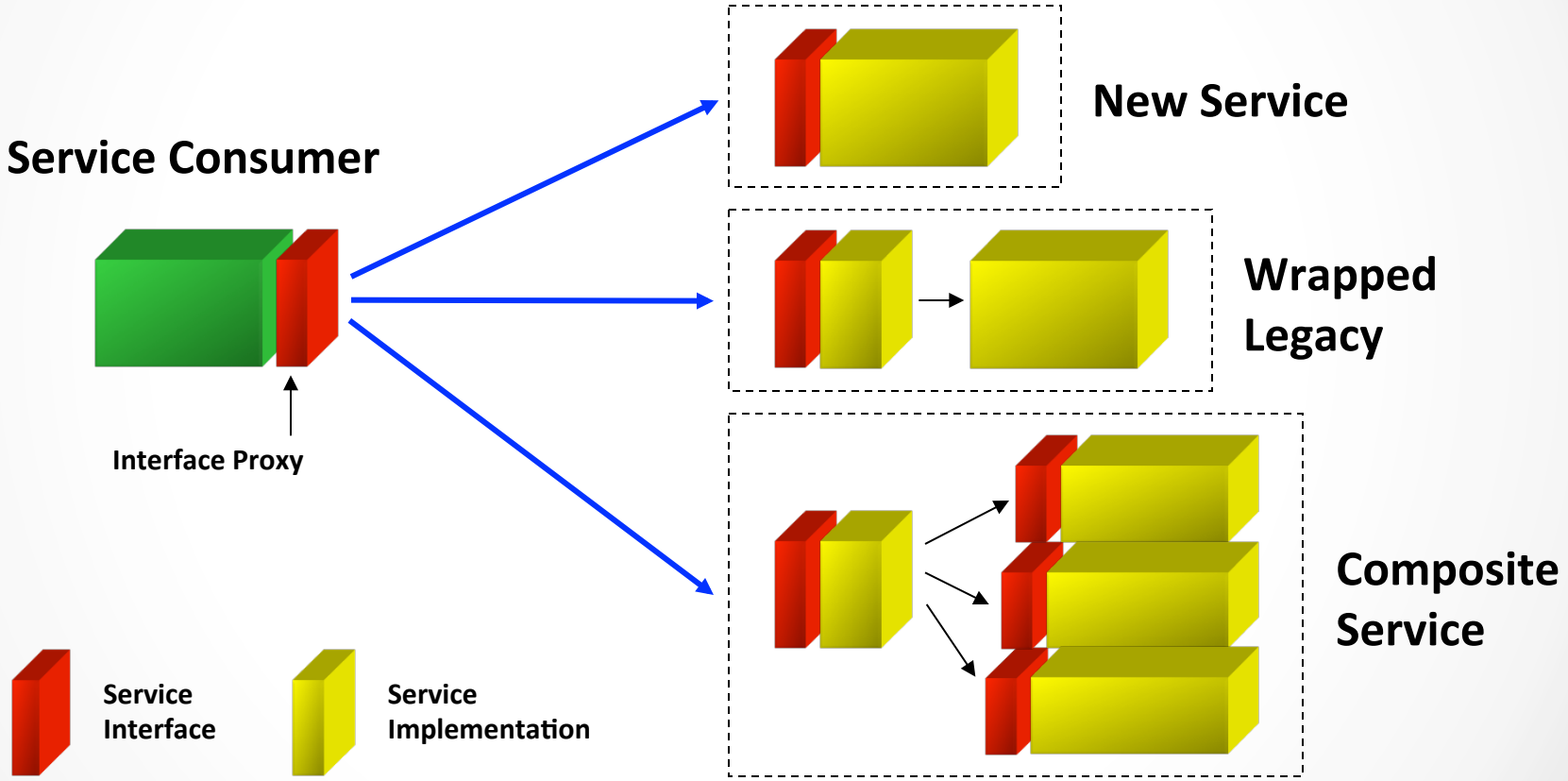
- Služba je recyklovatelná komponenta.
- Služba mění byznys data z jednoho stavu do jiného.
- Jen služba má přístup k datům
  - Není jiné cesty.
- Pokud lze komponentu popsat v WSDL, pak je to Služba.

Why SOA?

- Odpověď na měnínící se byznys pravidla
- Adresuje nové potřeby v existujících aplikacích
- Odemyká existující aplikace rozšíření
  - Expedia API, Paypal, Amazon API, Airfare, Heureka..
- Podpora novým postupům a komplexní interakci
- Podporuje přirozený vývoj

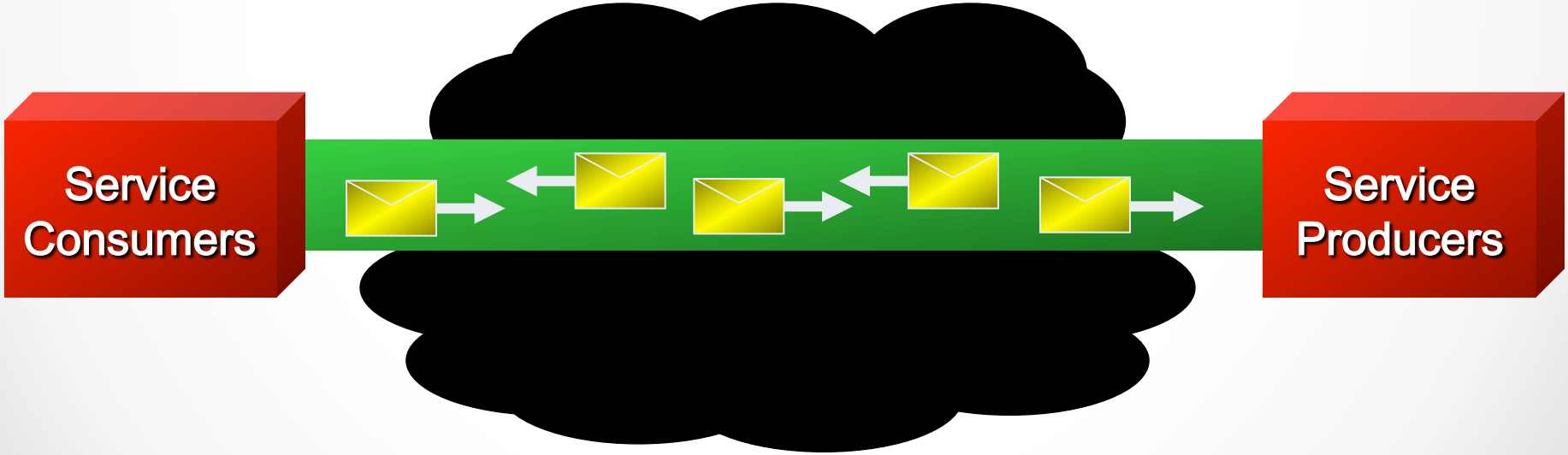


Anatomy of a Service

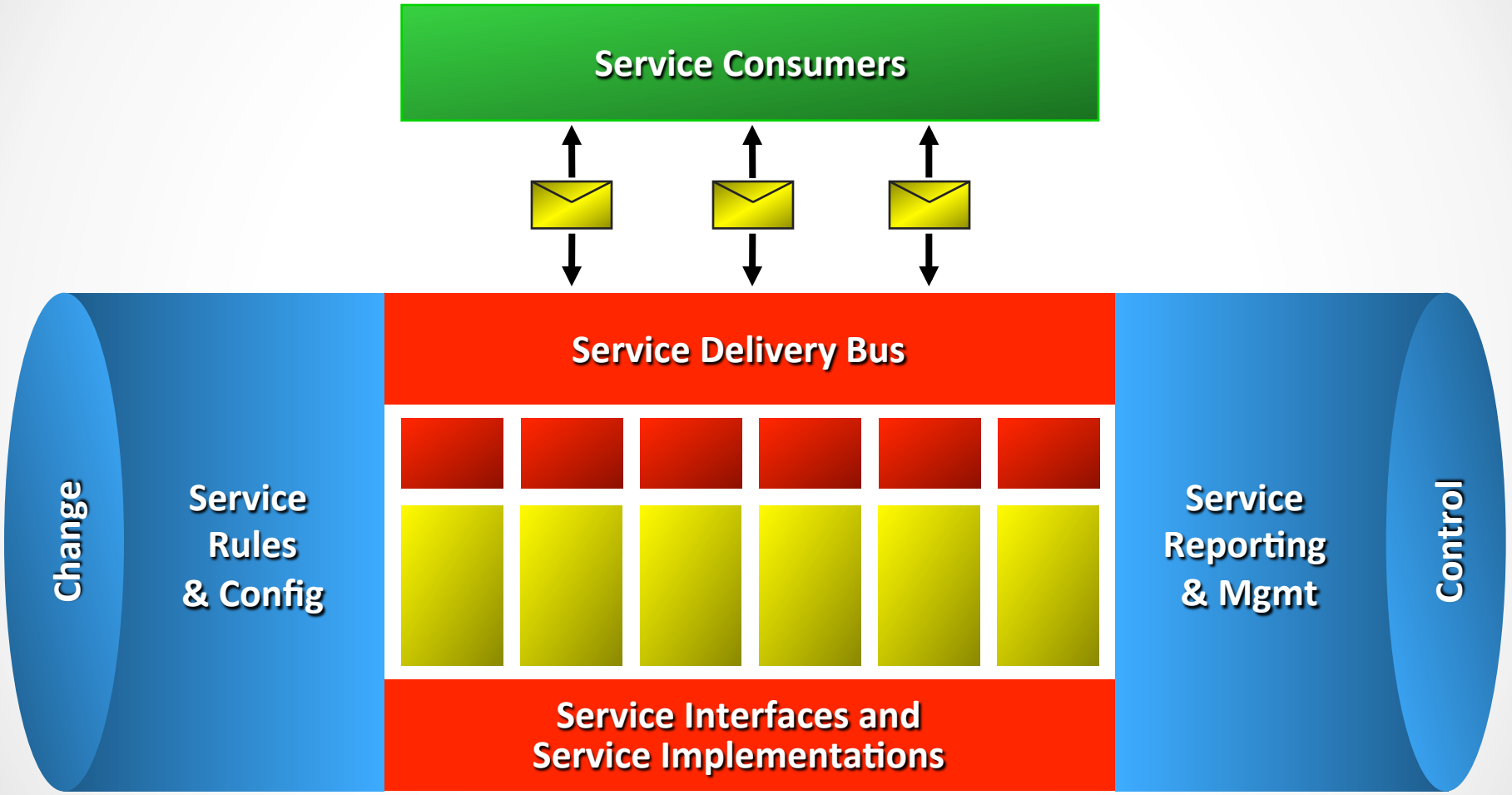


Service Communication

- Communicate with messages
- No knowledge about partner
- Likely heterogeneous



Service Platform

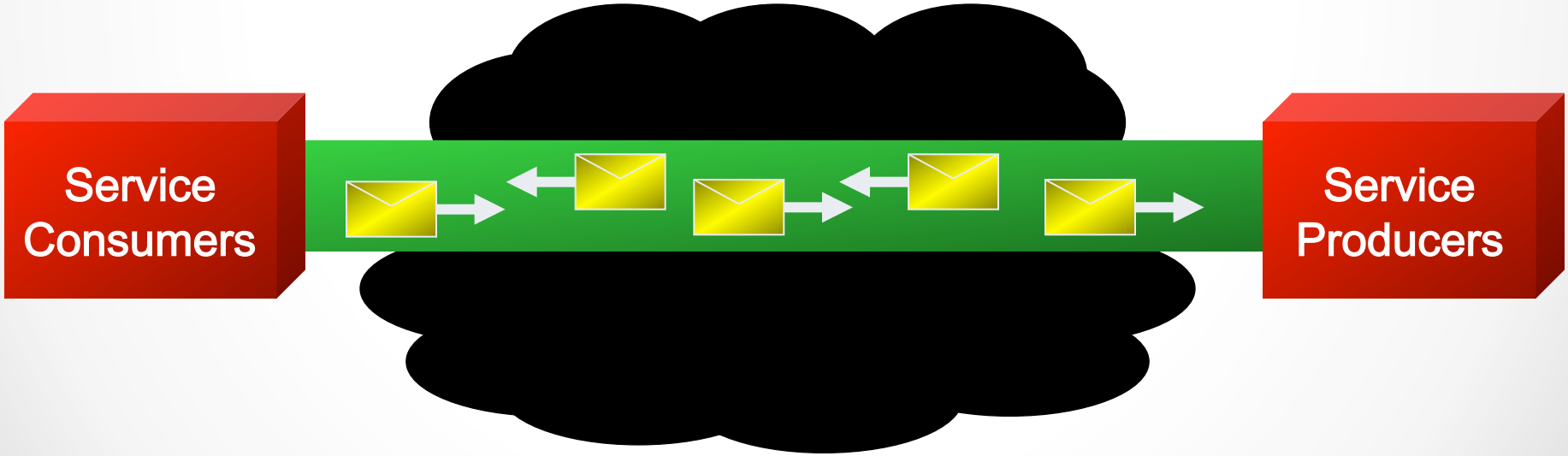


Benefits of SOA

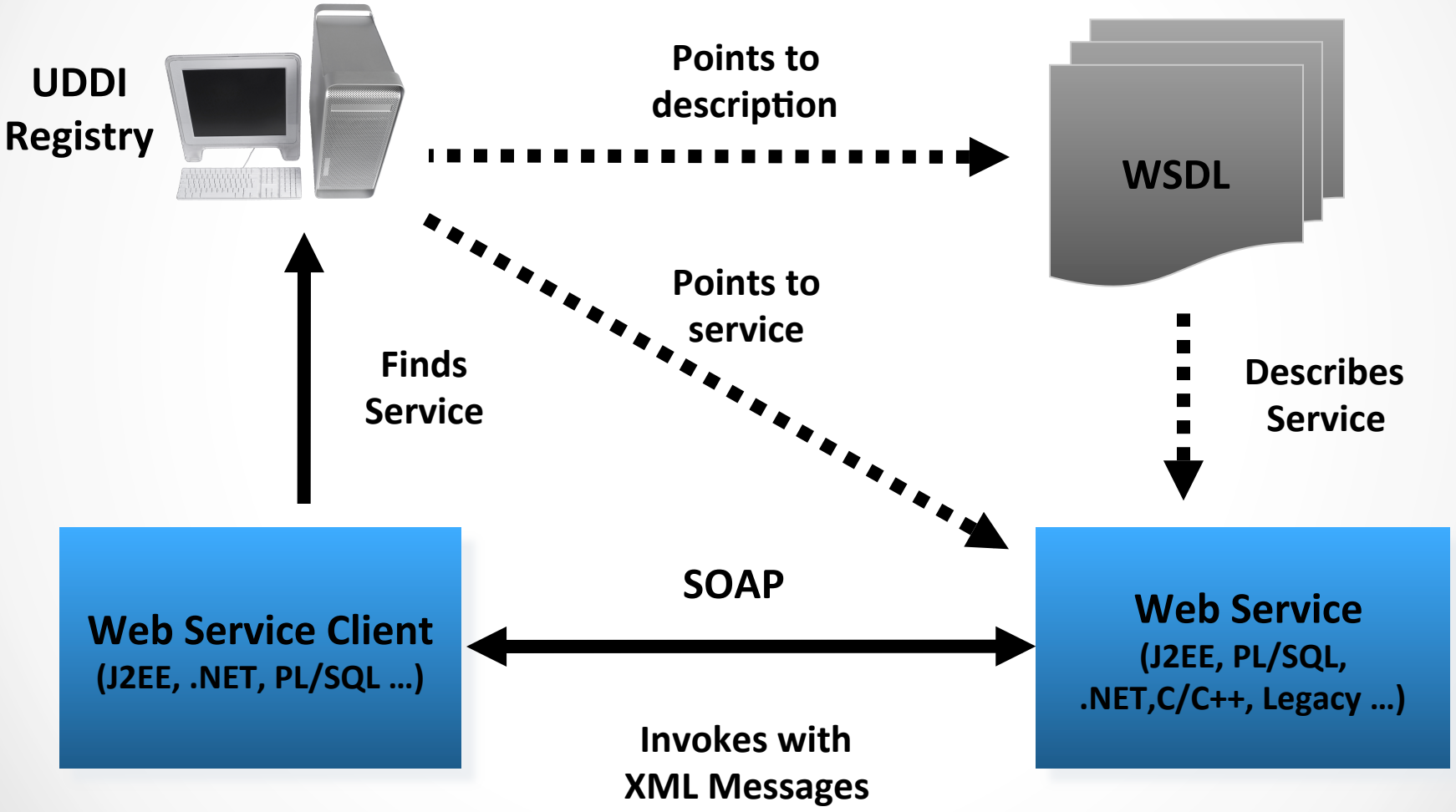
- Better reuse
  - Build new client functionality on top of existing Business Services
- Well defined interfaces
  - Make changes without affecting clients
- Easier to maintain
  - Changes/Versions are not all-or-nothing
- Better flexibility

Services Communicate With Messages

- Providing reliability and security to messages
- Sending messages across consumers and producers
- Service Orchestration



Basic Web Services



Servis vlastnosti – contract

Service description (contract)

- Header
  - Obecné informace a zodpovědnosti
- Functional parameters
  - Funkčnost a volání
- Non-functional parameters
  - Security, QoS
- Typicky WSDL(Web Service Description Language)

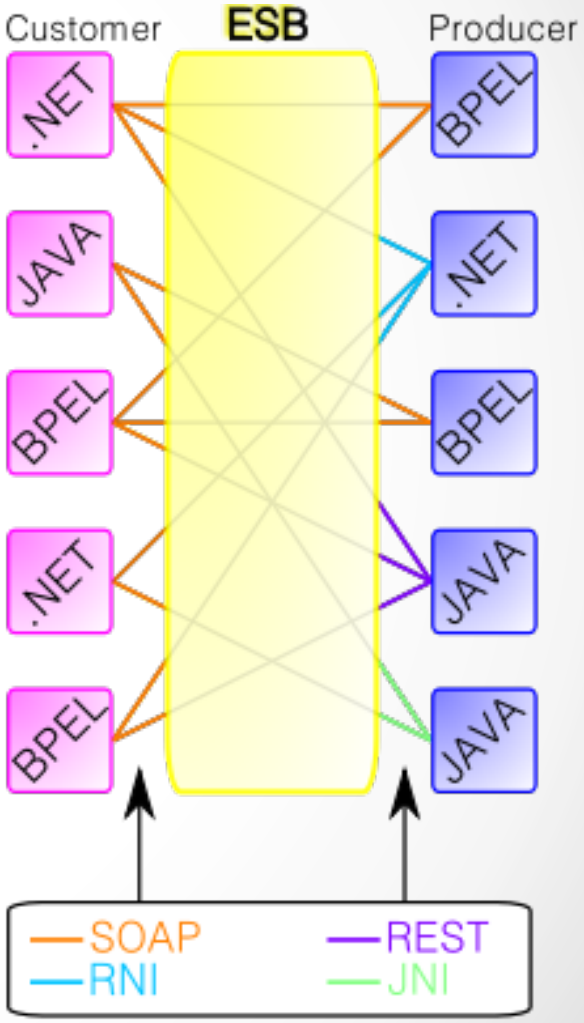
SOA

SOA implementace

- Web services příklady
  - SOAP
  - REST
  - RPC
  - JSON
  - other
- SOA Frameworks
  - (Oracle SOA Suite, Sun Java CAPS…)

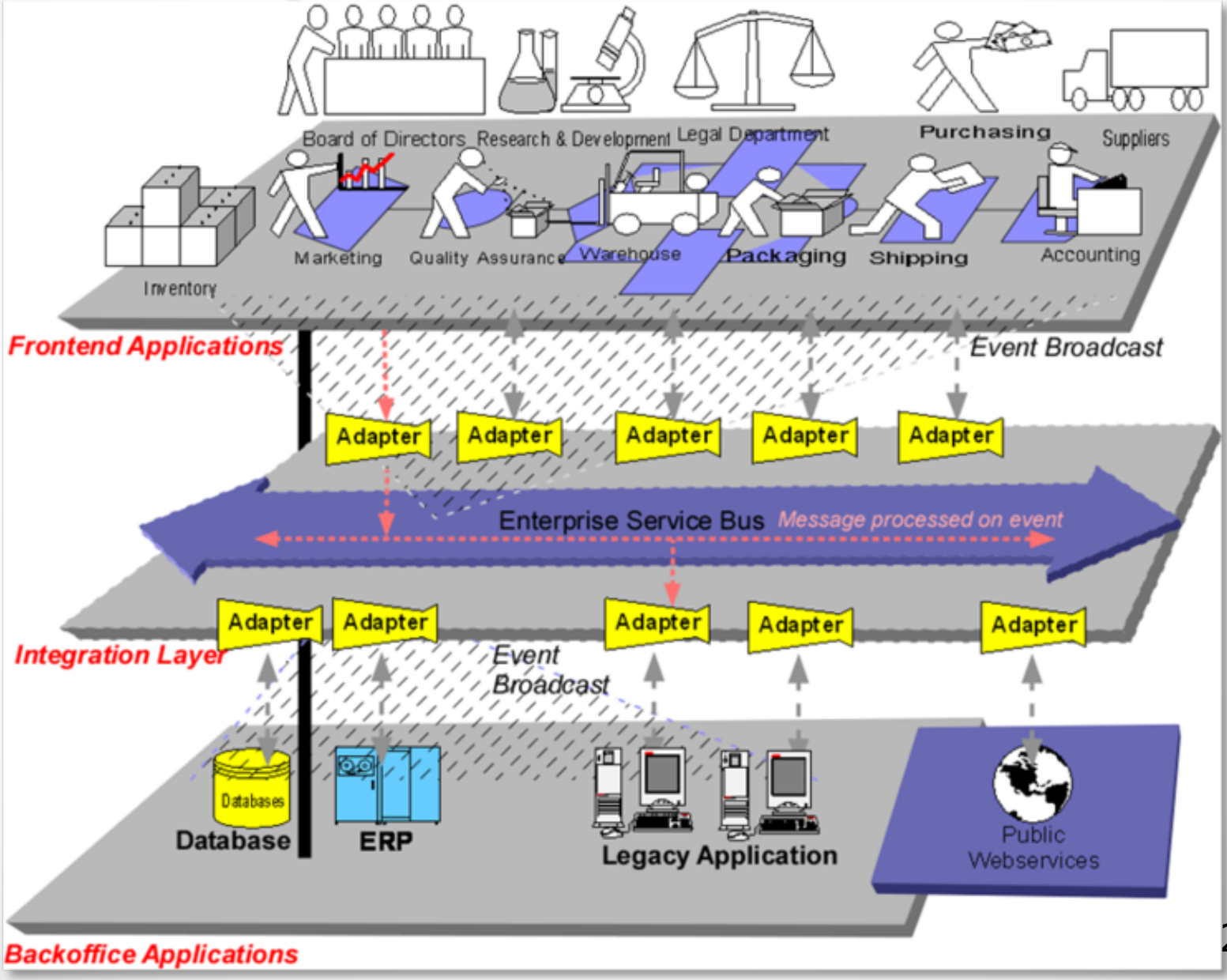
Enterprise service bus ESB

It is **a software architecture model** used for designing and implementing the interaction and **communication** between mutually interacting **software applications in** service-oriented architecture (**SOA**).

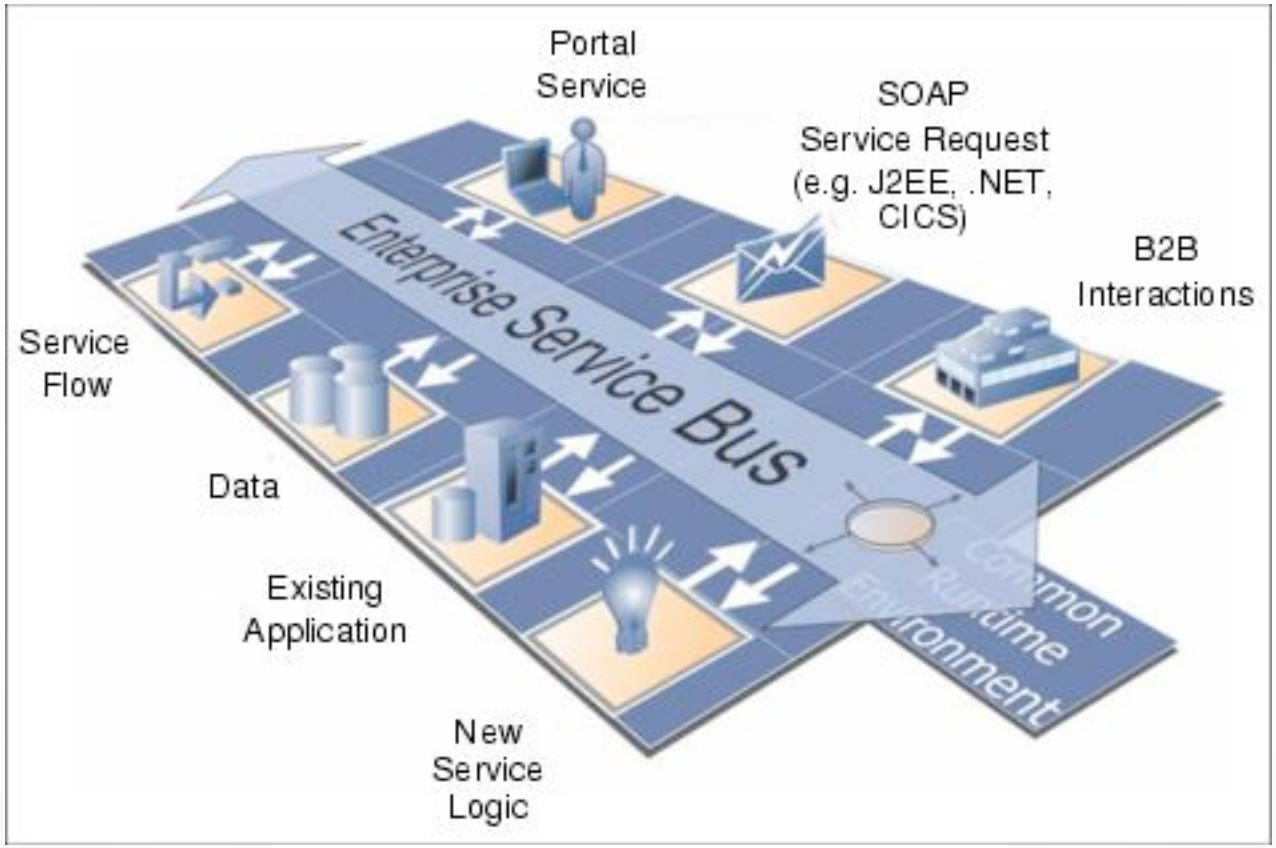


- Model for distributed computing
- Variant of client server software architecture model
- Promotes flexibility with regards to communication & interaction between applications.
- Primary use in enterprise application integration (EAI) of heterogeneous and complex landscapes.

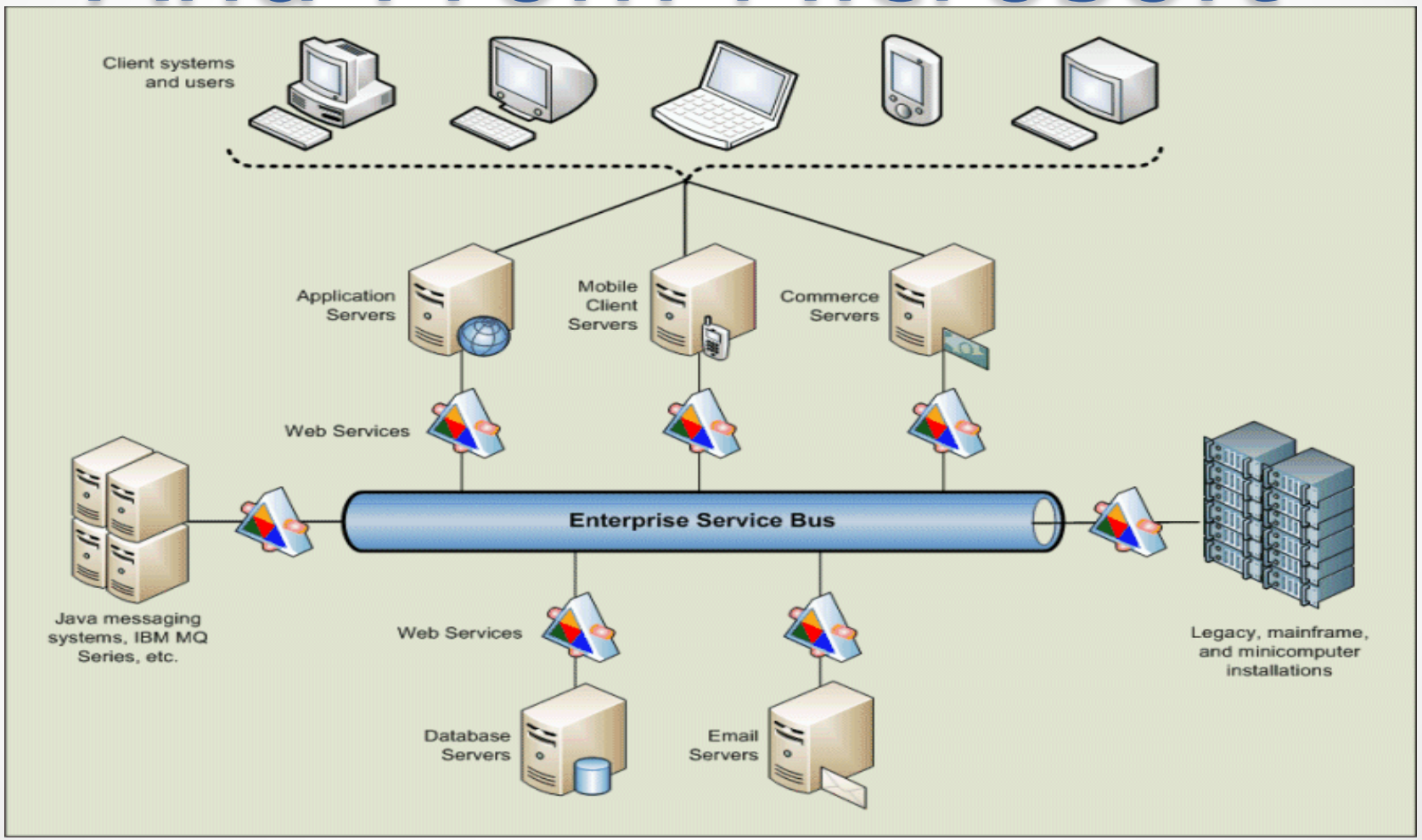
Enterprise service bus

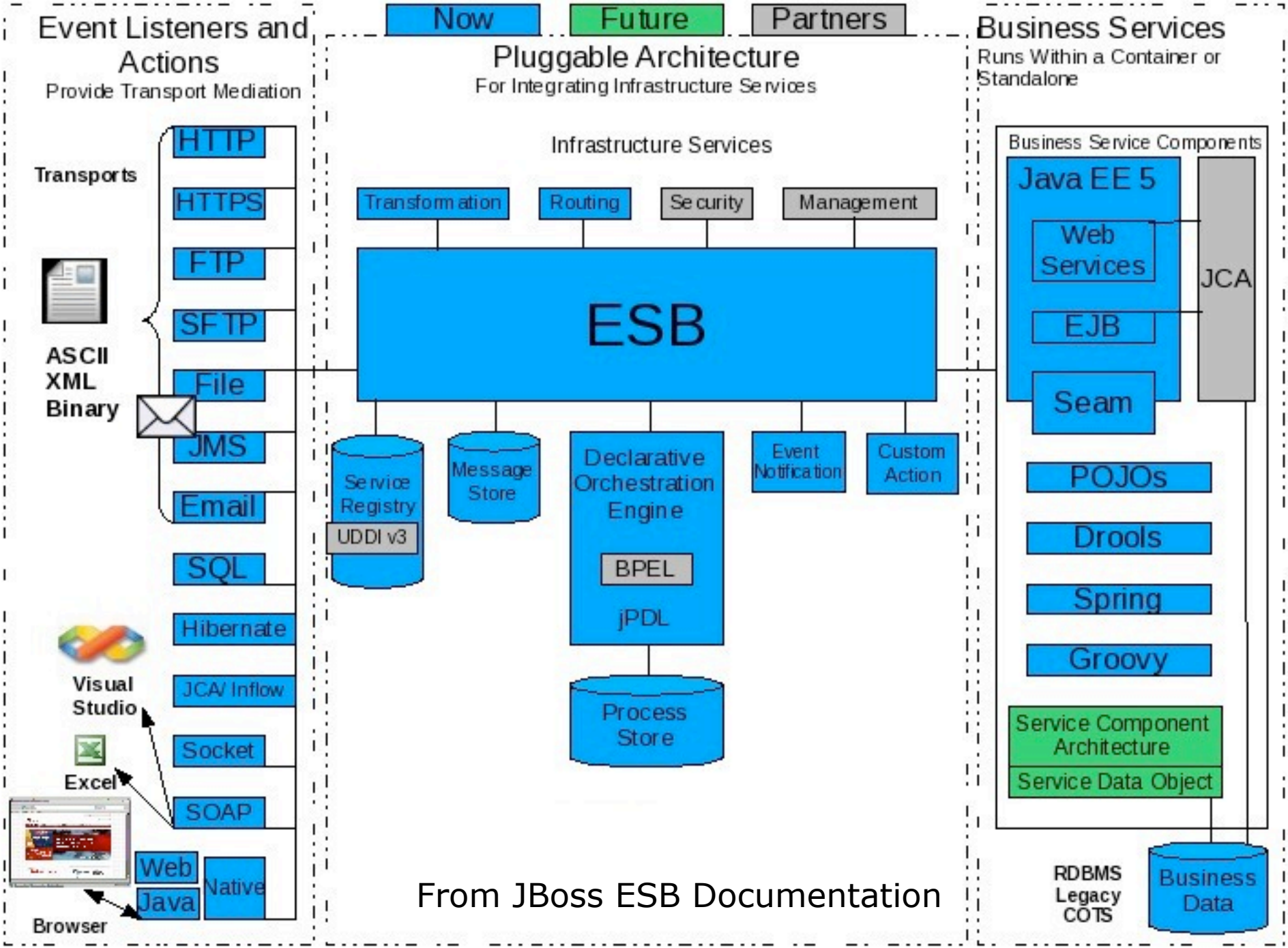


ESB's From IBM (1)



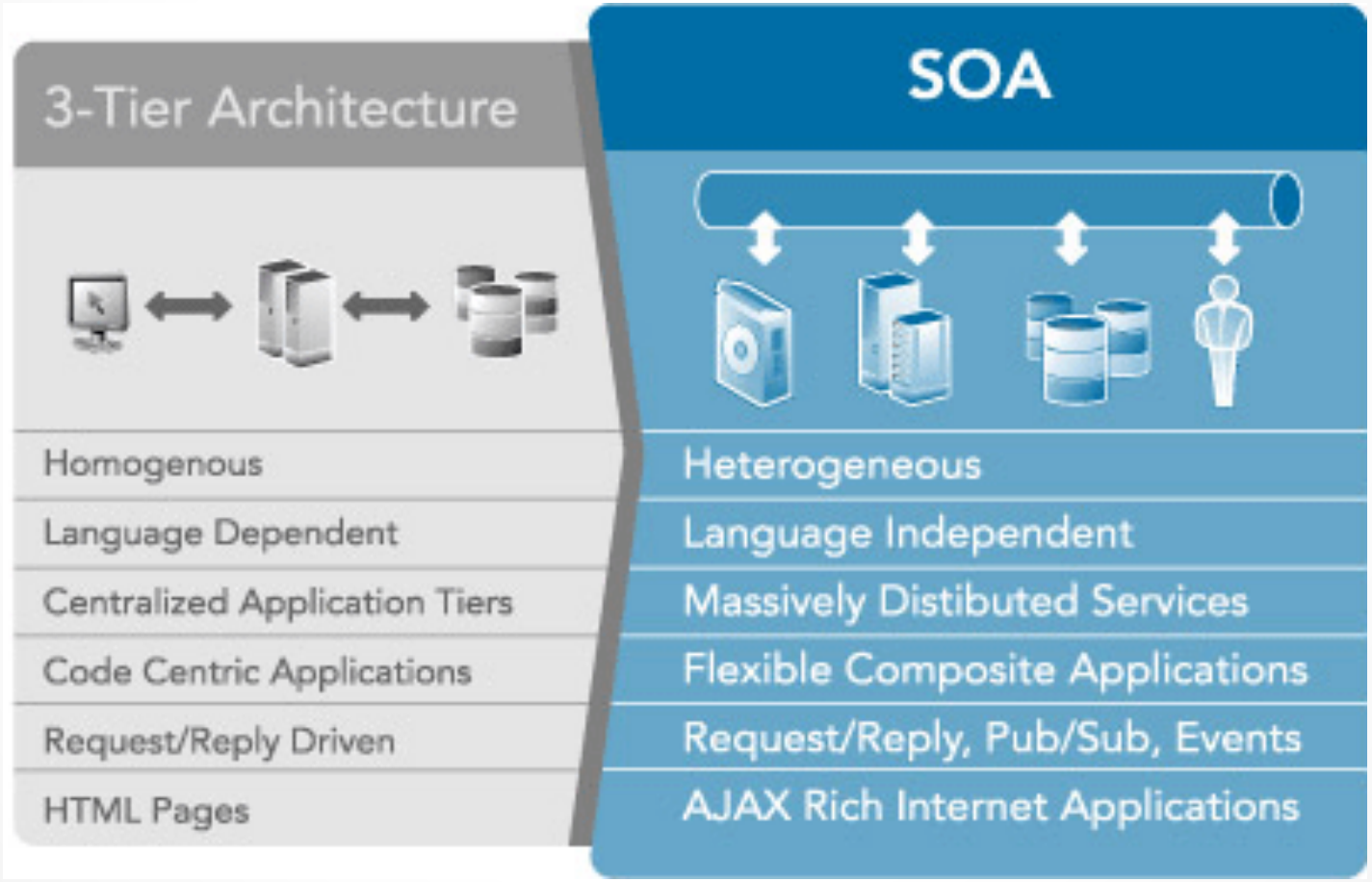
And From Microsoft





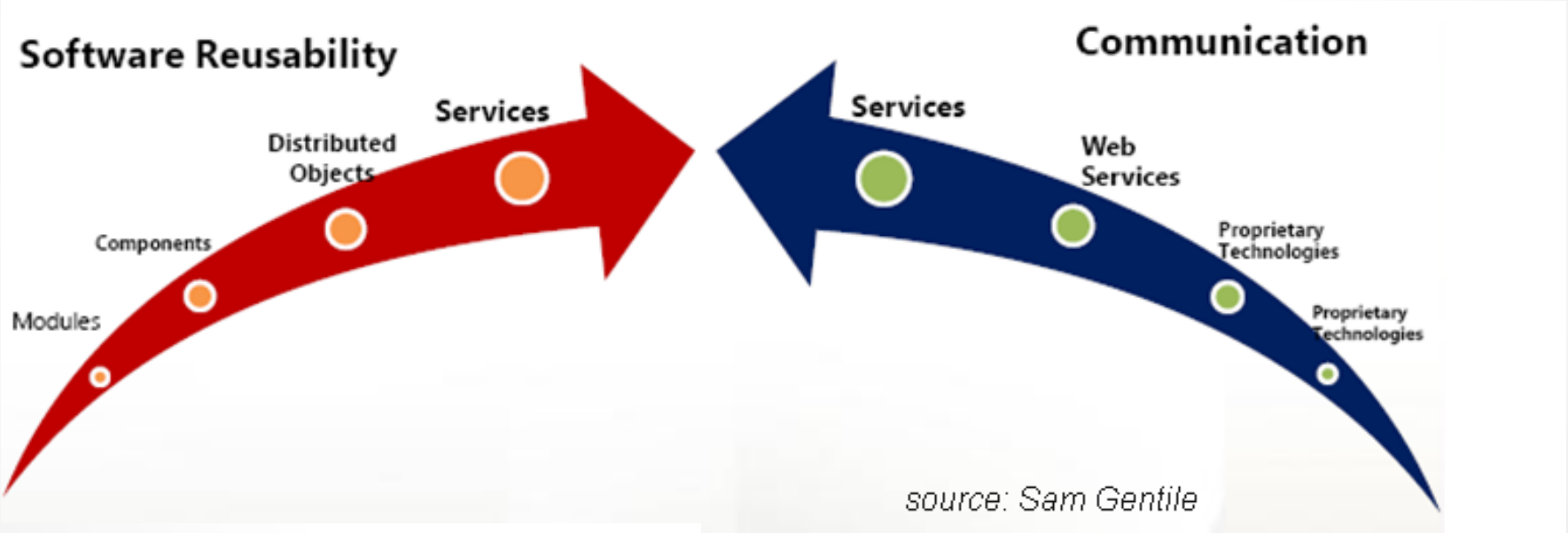
From JBoss ESB Documentation

SOA is an evolutionary step



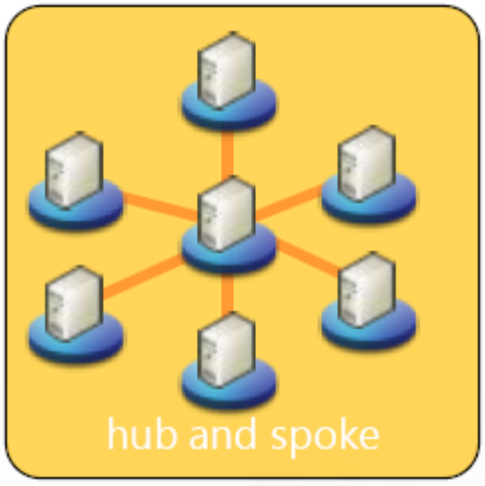
SOA is an evolutionary step

- in reusability and communication



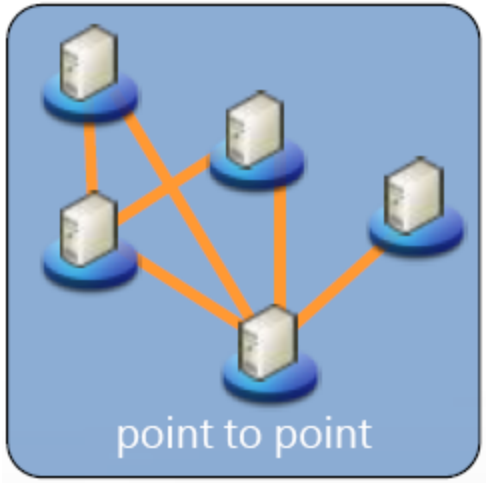
SOA is an evolutionary step

- in distributed communications



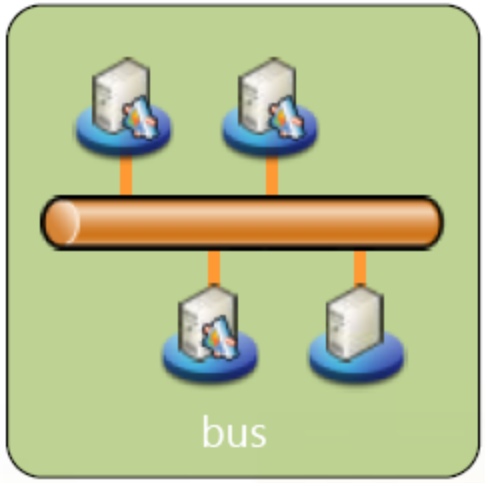
"too centralized

EAI



"too decentralized"

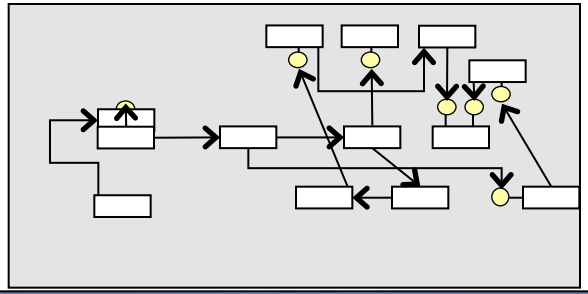
Project-ware



"just right"

SOA

To enable Business Process Optimization and the Real Time Enterprise (RTE)



BPM Expressed in terms of Services Provided/Consumed

Seamless End to End Process

Service to Customers

Service from Multiple Suppliers

Enterprise

Smart Clients

Stores POS

Mobile

3rd Party Agents

Portal

Internal Systems

SELF-SERVE CHANNELS

ATM/Kiosk

IVR/Speech

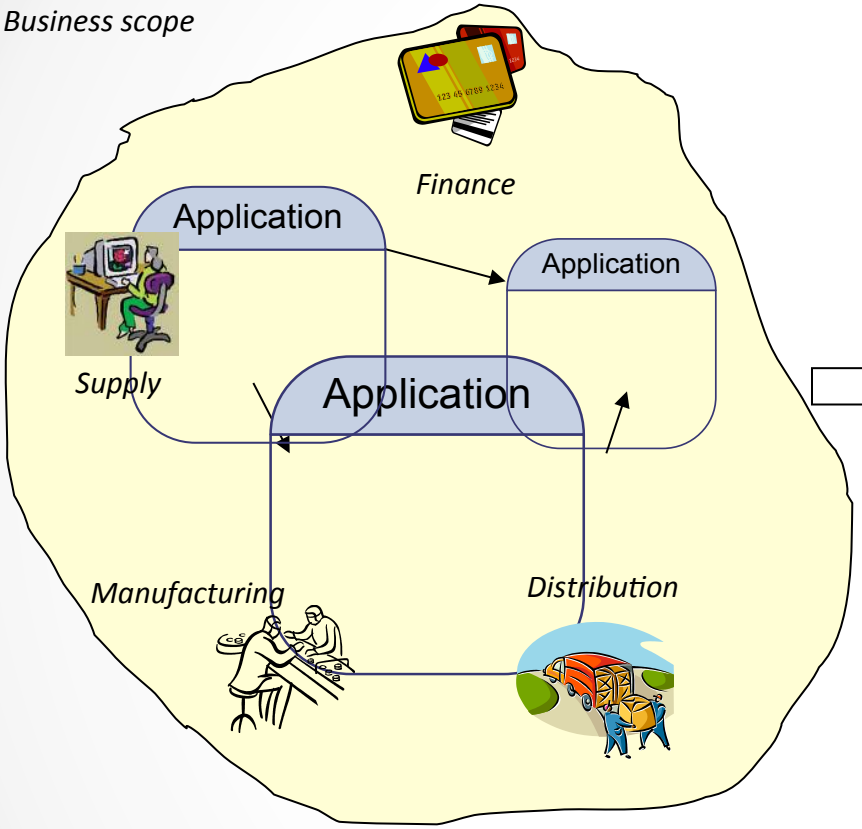
Smart Device

Web Portal

SOA Patterns: Single, Multi-Channel Service for consistency

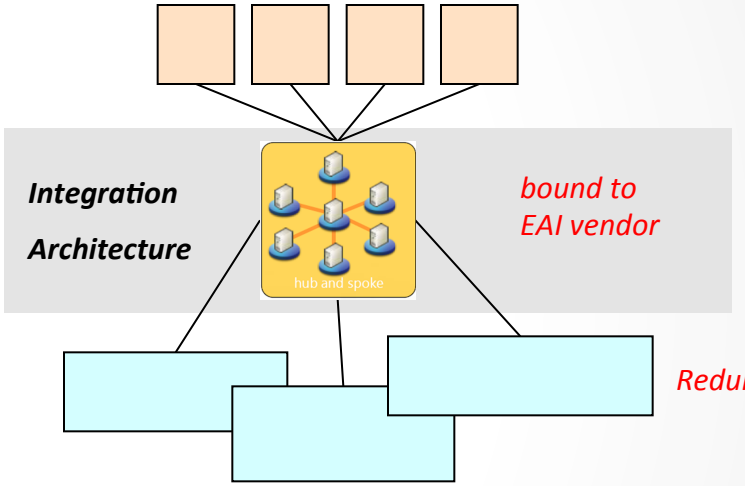
SOA Pattern: Standardized Service provided by multiple suppliers

Application Centric



Business functionality is duplicated in each application that requires it.

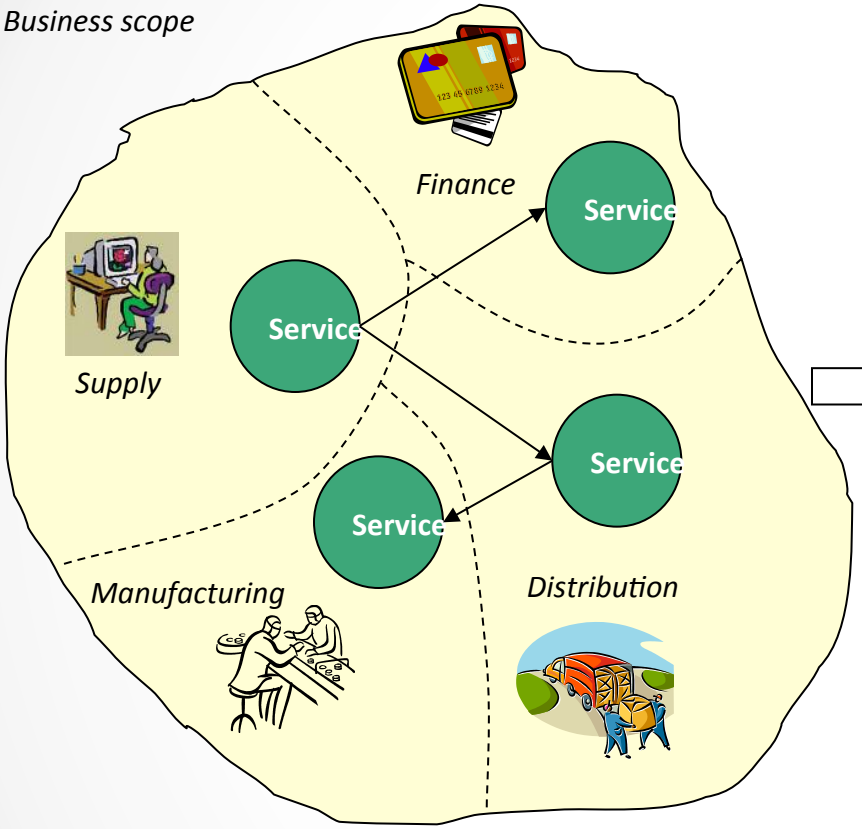
Narrow Consumers
Limited Business Processes



Overlapped resources
Overlapped providers

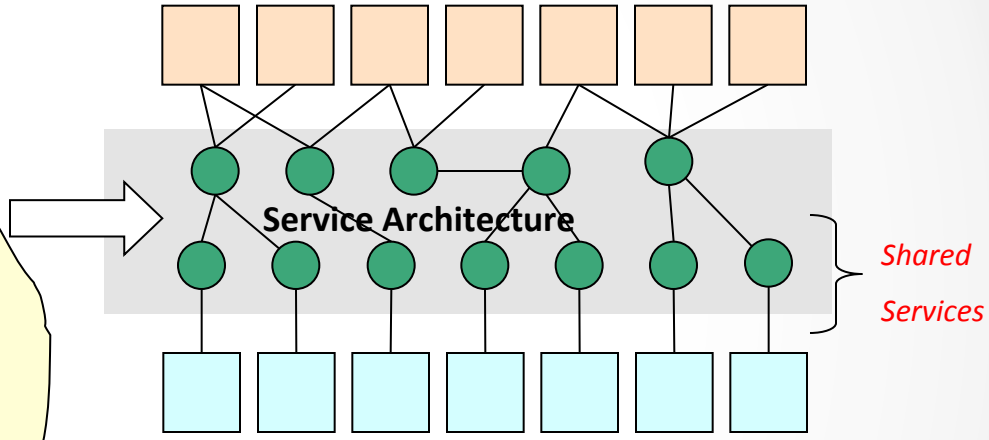
EAI ‘leverage’ application silos with the drawback of data and function redundancy.

Service Centric



SOA structures the business and its systems as a set of capabilities that are offered as Services, organized into a Service Architecture

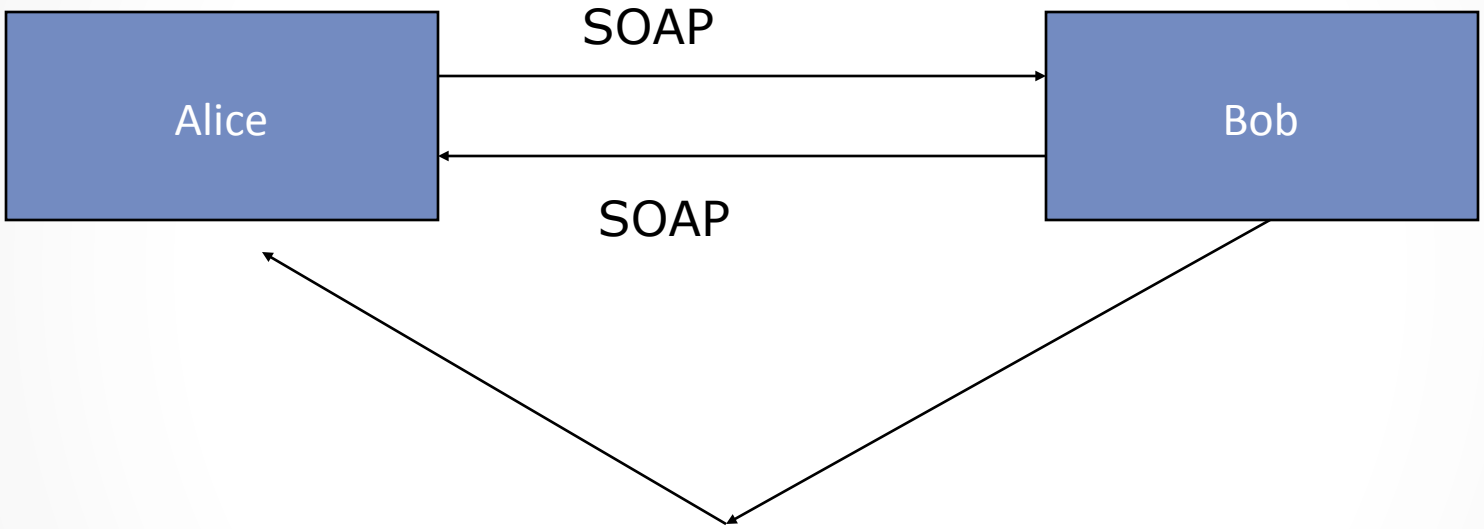
Multiple Service Consumers
Multiple Business Processes



Multiple Discrete Resources
Multiple Service Providers

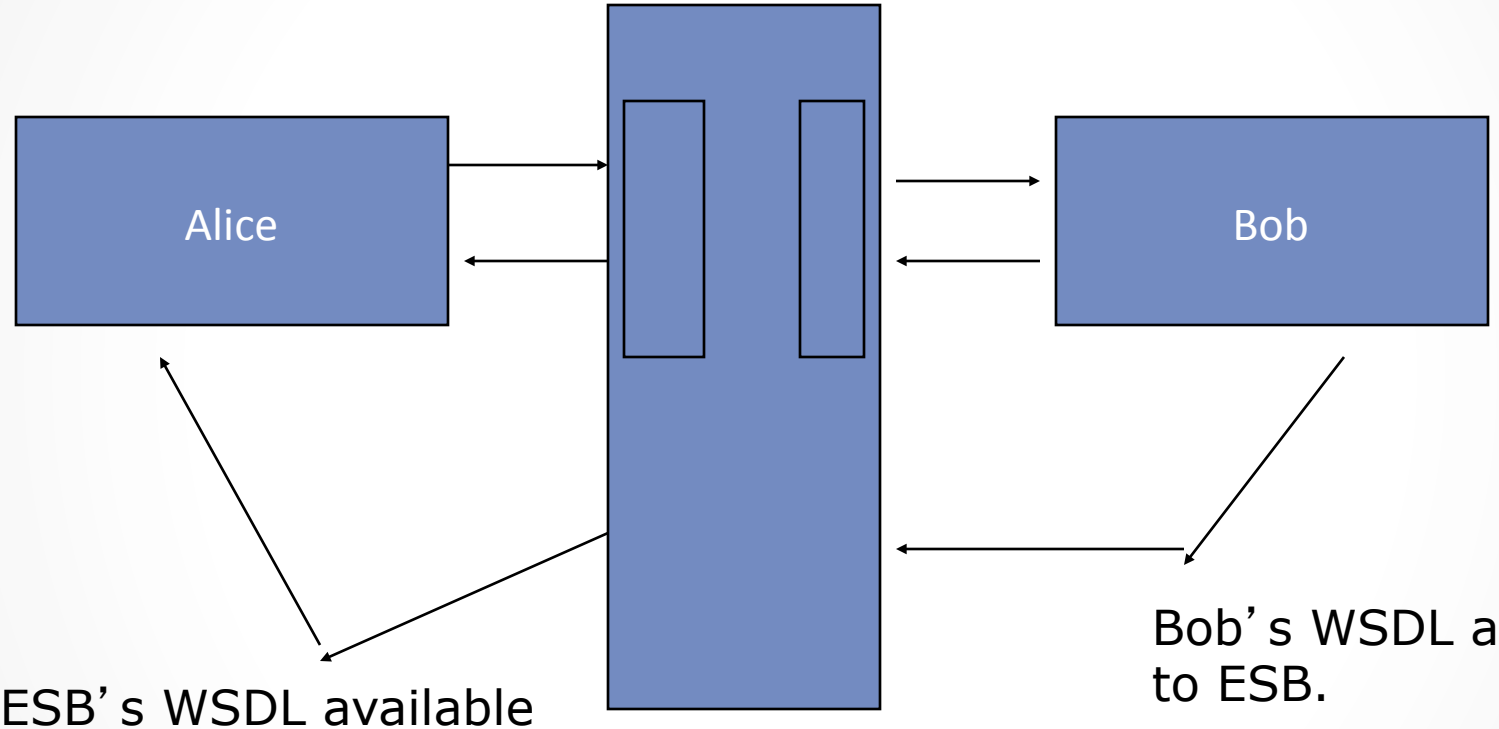
Service virtualizes how that capability is performed, and where and by whom the resources are provided, enabling multiple providers and consumers to participate together in shared business activities.

Simple WS Without an ESB



Bob's WSDL is available to Alice.

Simple WS With an ESB



ESB's WSDL available to Alice.

Bob's WSDL available to ESB.

The backend protocol may be different from the front end.

Bob may be a legacy application with a web service front end.

REST examples

CMS XWiki

Web page (HTML):
http://icpc.baylor.edu/xwiki/wiki/public/view/**worldfinals/worldfinals**

REST (XML)
http://icpc.baylor.edu/xwiki/rest/wikis/virtpublic/spaces/**worldfinals**/pages/**worldfinals**

^ service

Client

http://icpc.baylor.edu/worldfinals/worldfinals

* Parse XML || of lucky just a lib to use