Introduction to SOA (II)

Huibert Aalbers
Senior Certified Software IT Architect

IT Insight podcast

- This podcast belongs to the IT Insight series
  You can subscribe to the podcast through iTunes.
- Additional material such as presentations in PDF format or white papers mentioned in the podcast can be downloaded from the IT insight section of my site at http://www.huibert-aalbers.com
- You can send questions or suggestions regarding this podcast to my personal email, huibert_aalbers@mac.com

The origins of SOA

- SOA (Service Oriented Architecture) is a direct answer to the needs of the business users that require more flexible software applications that can be quickly adapted to rapidly changing market conditions
- SOA is based on technologies initially developed for B2B applications but that have proven very useful inside the intranet

Service Oriented Architecture (SOA)

- Instead of creating large complex applications, smaller reusable components are developed which are easier to test and maintain
- Applications are created by graphically designing the process that interacts with the components
- Each new application reuses existing components and only required components that do not exist are developed
- Components are known as services

What is a service?

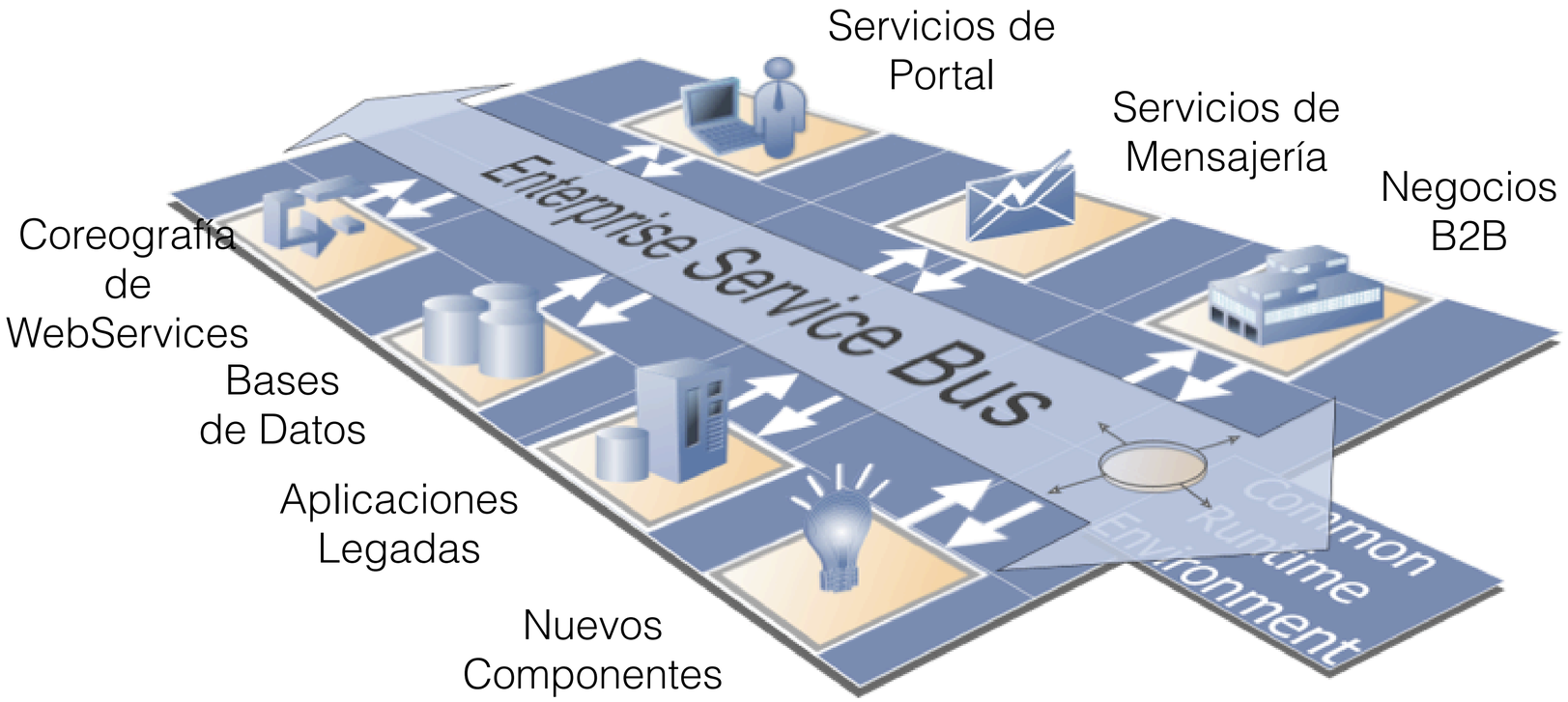
- A service represents a clearly defined business function that can be remotely invoked through standard communication protocols
- Services are defined through explicit interfaces that are completely independent of the service implementation (WSDL)
- Services should be invoked using standards that focus on interoperability and location independence

How is a service programmed?

- Services can be developed in almost any programming languages (Java, C/C++, C#, Cobol, RPG, Perl, etc.)
- This is very useful to modernize legacy systems
- However, for new services it is better to choose a single language based on its merits for scalability, portability and high-availability. My personal choice is Java
- Developing web services is the same as developing any kind of application. It is therefore highly recommended to adopt a proven development methodology to get it done right

Service Oriented Architecture

The adoption of a service oriented architecture requires a scalable and secure communications infrastructure to connect components.This is known as an Enterprise Service Bus.



The Enterprise Service Bus (ESB)

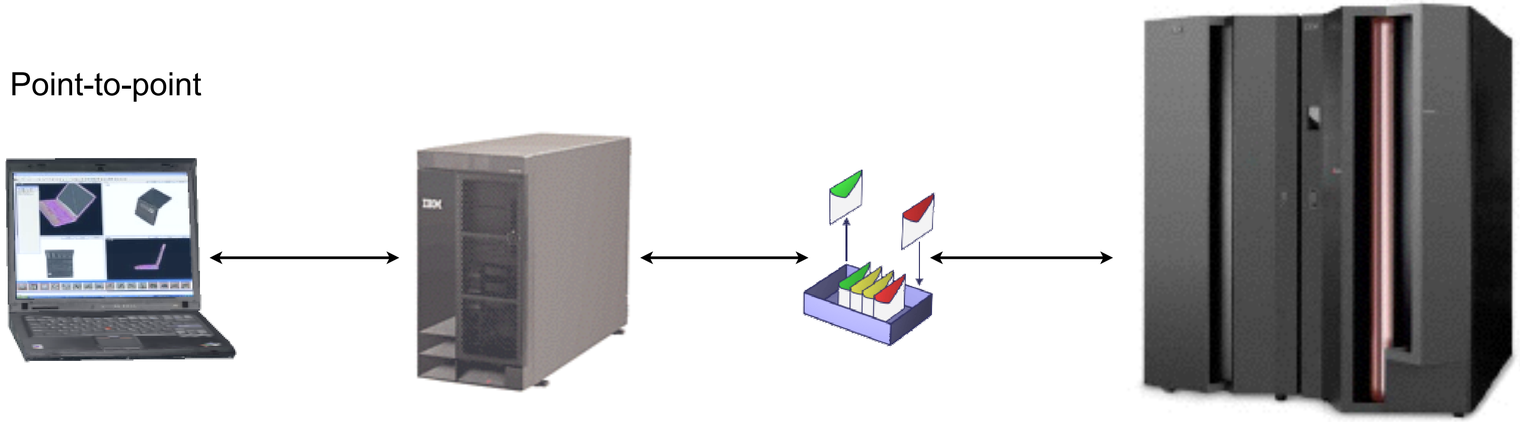
- The Enterprise Service Bus is a concept that refers to the message transport infrastructure software that reliably connects the process engine with the enterprise services
  - The bus has to be built on a rock solid infrastructure that guarantees message delivery, no matter what problems could be affecting either the network or particular services or systems
  - The ESB has to offer complete security as well as connectivity to all the legacy enterprise applications
  - The bus should be easy to monitor since it becomes the spinal cord connecting all the company business systems

Message Queues

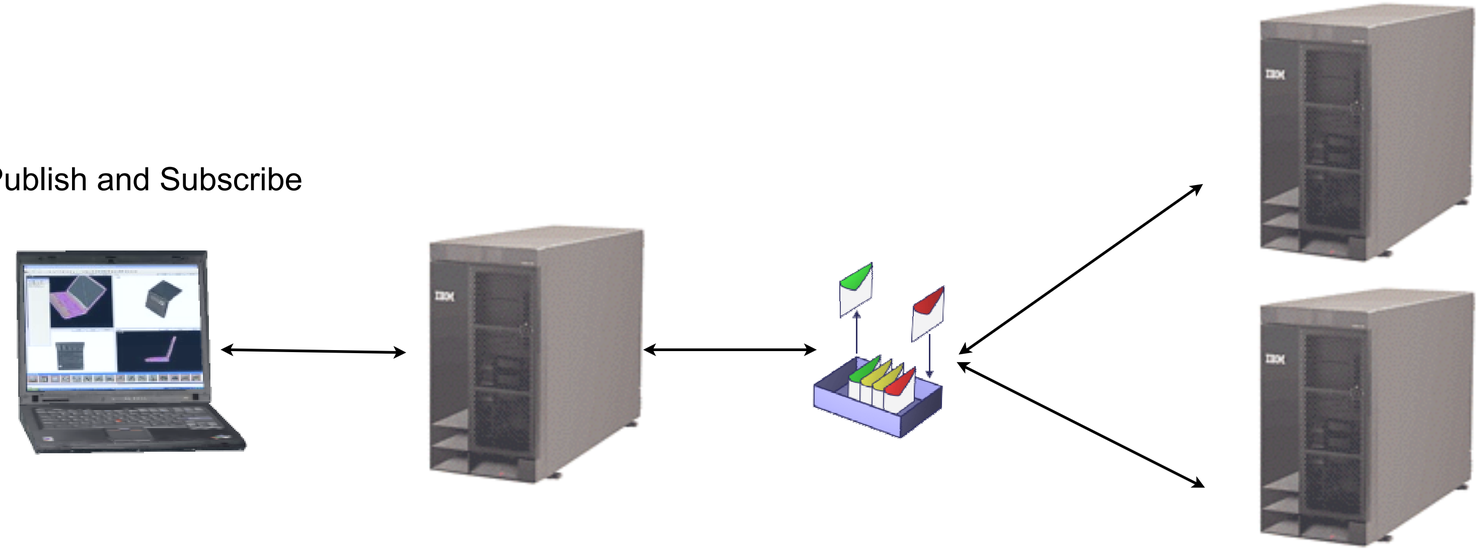
- Message queues are used to communicate different applications, assuring message delivery and reducing application development complexity
- They are usually employed to improve the availability and scalability of critical IT systems
- Queues can be persistent or not and are used in either point-to-point or publish and subscribe models

Point-to-Point & Publish and Subscribe models

Point-to-point

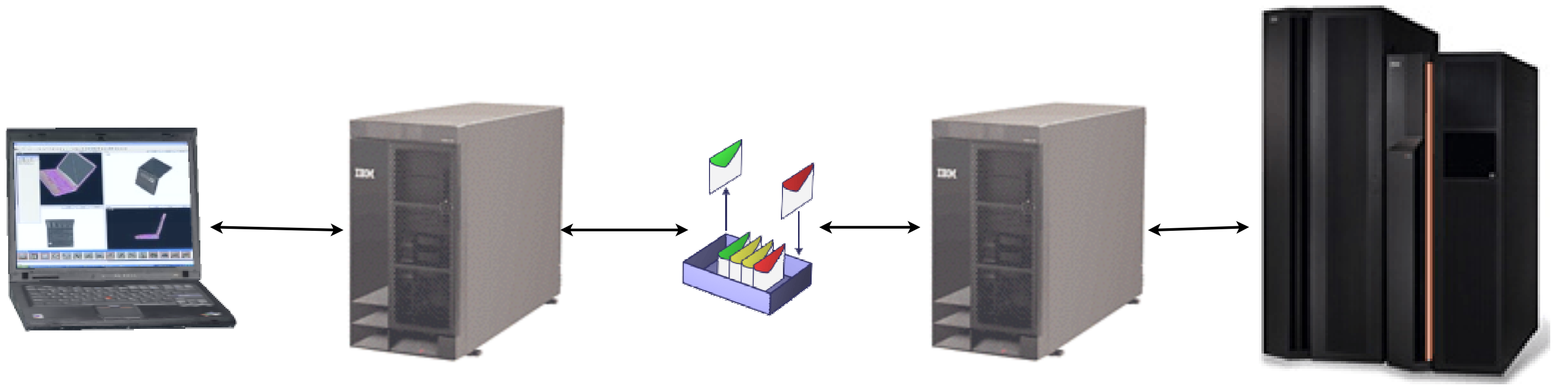


Publish and Subscribe



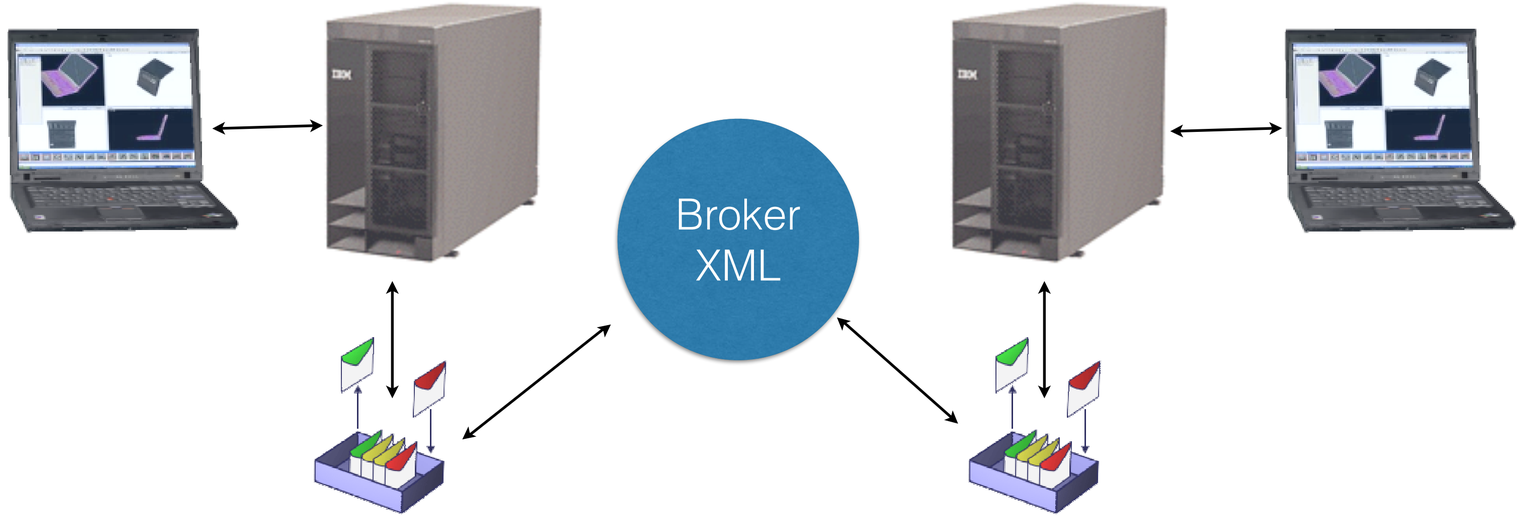
JMS

- JMS (Java Message Service) is the Java API that allows to access message queues in a standard way, independently of your software provider (IBM,Tibco, Progress or others).
- In order to simplify the use of message queues, J2EE 1.3 introduced a new type of EJB, called Message Driven Bean



Transformation and message routing

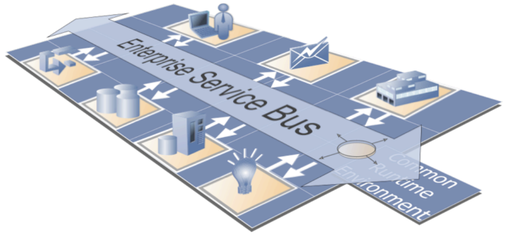
- For improved flexibility, it is better that instead of sharing a single queue, each system be assigned its own queue which exchanges messages with other queues through a message broker
- The mission of a broker is to take messages from a queue, transform them and finally route them to another system, all within a single business transaction



Enterprise Application Integration

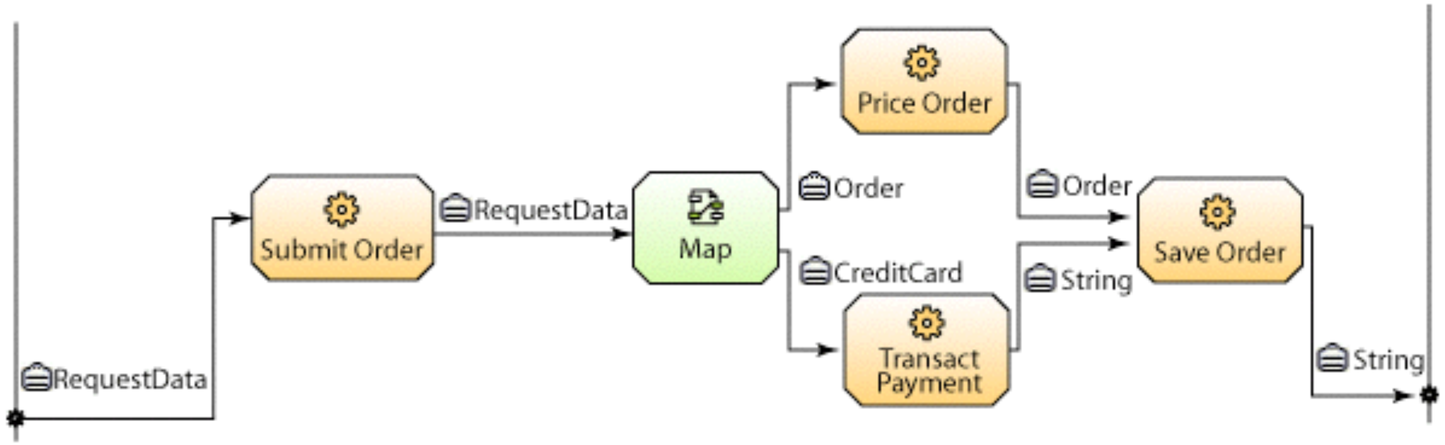
The standard for connecting J2EE applications to legacy applications is known as Java Connector Architecture (J2C o JCA)

- J2C connectors are the easiest way to link J2EE applications to legacy systems and exposing their functionality as services
- This makes it possible for legacy applications to actively participate in new processes
- While J2EE application server providers are not required to also sell J2C connectors, some do and, there are companies, specializing in providing connectors



Service Orchestration

- By using the ESB infrastructure, it is possible to combine services to create a new, more complex service
- This is known as web service orchestration
- There is a standard called BPEL (Business Process Execution Language) to define automated processes



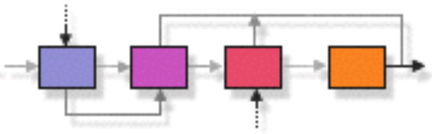
BPEL limitations

- BPEL is very useful to orchestrate web services but it is clearly insufficient to define complex business processes
- No explicit support for human intervention, automatic distribution of tasks between persons who share a role or escalation
- Does not include mechanisms to include in the processes components that are not services, such as
  - Rules engine
  - State machines
  - Etc.

SCA (Service Component Architecture)

- SCA is a new standard created by BEA and IBM to overcome the limitations of BPEL
- With SCA it becomes possible to model complex processes that include diverse kinds of components
  - BPEL processes
  - Human interactions
  - Rules engines, etc.
- SCA uses SDO (Service Data Objects JSR 235) to represent XML data flowing through the process

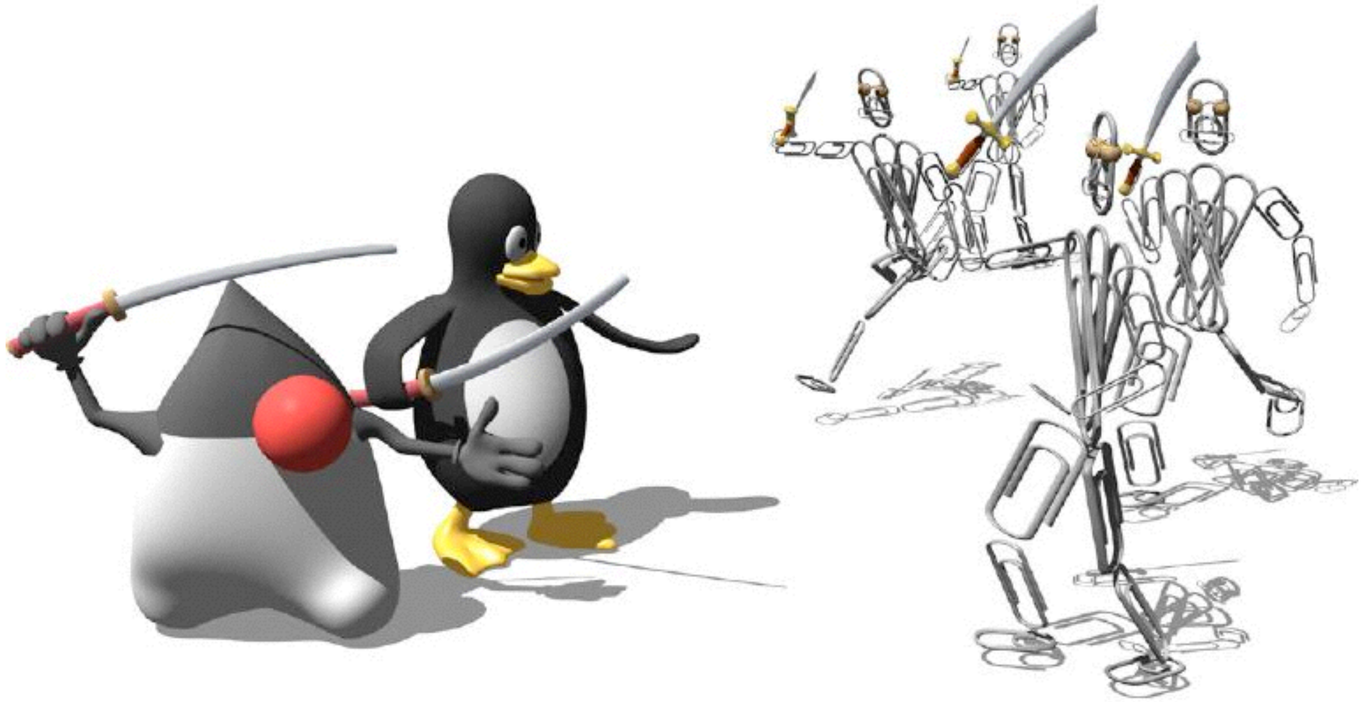
Java Community Process



Community Development of Java Technology Specifications

Business Process Management

- One of the main benefits of adopting SCA and BPEL is that the business processes are totally separated from the components that implement business functions
- This will progressively allow business users to model the business processes
- New tools allow the monitoring of business processes created using BPEL and SCA by measuring key performance indicators during process execution
- This information allows to improve the processes, closing the BPM virtuous circle



For more information, please contact me at
huibert_aalbers@mac.com