B13

Web Services for Dummies

Hilon Potter

IBM
SYSTEM z9 AND zSERIES EXPO
October 9 - 13, 2006

Orlando, FL

© IBM Corporation 2006

RETURN TO INDEX

Agenda

- So what are we talking about
- Web Services Standards
- The basics of Web Services
- SOA What? ☺

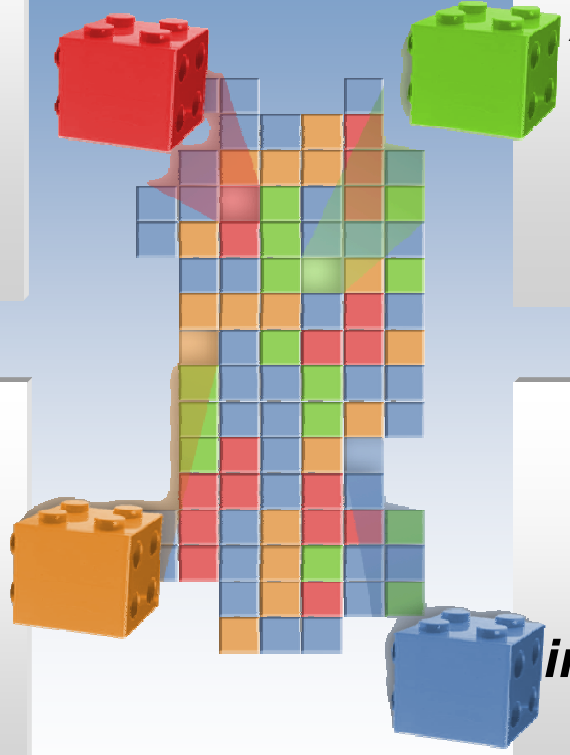
What is …..

… a service?

*A **repeatable business task** – e.g., check customer credit; open new account*

… service orientation?

*A way of integrating your **business as linked services** and the outcomes that they bring*



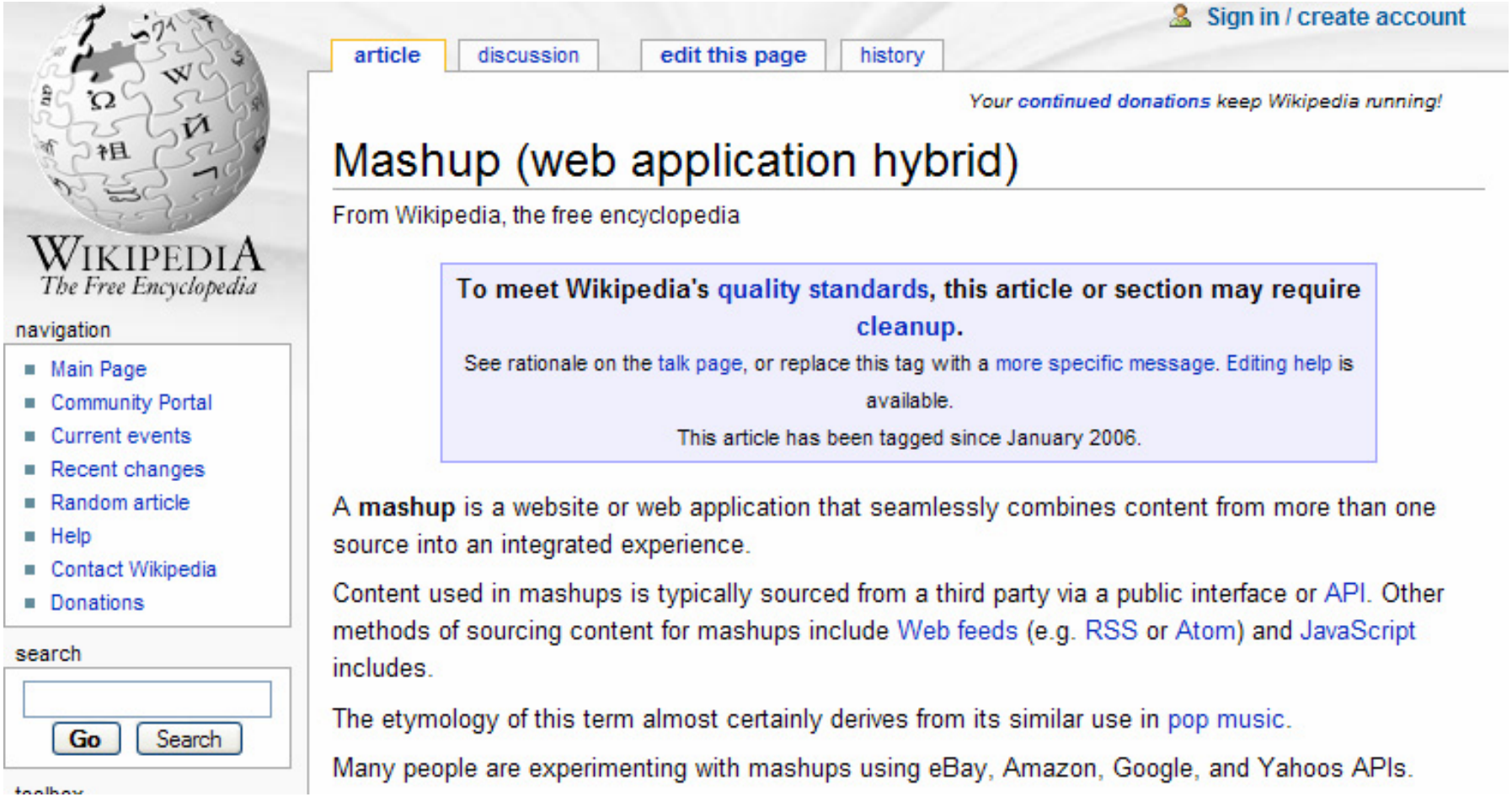
… service oriented architecture (SOA)?

*An IT **architectural style** that supports service orientation*

… a composite application?

*A set of **related & integrated** services that support a business process built on an SOA*

Have you heard about Mashup's



Sign in / create account

article | discussion | edit this page | history

Your continued donations keep Wikipedia running!

WIKIPEDIA
The Free Encyclopedia

navigation

- Main Page
- Community Portal
- Current events
- Recent changes
- Random article
- Help
- Contact Wikipedia
- Donations

search

Go | Search

Mashup (web application hybrid)

From Wikipedia, the free encyclopedia

To meet Wikipedia's quality standards, this article or section may require cleanup.

See rationale on the talk page, or replace this tag with a more specific message. Editing help is available.

This article has been tagged since January 2006.

A **mashup** is a website or web application that seamlessly combines content from more than one source into an integrated experience.

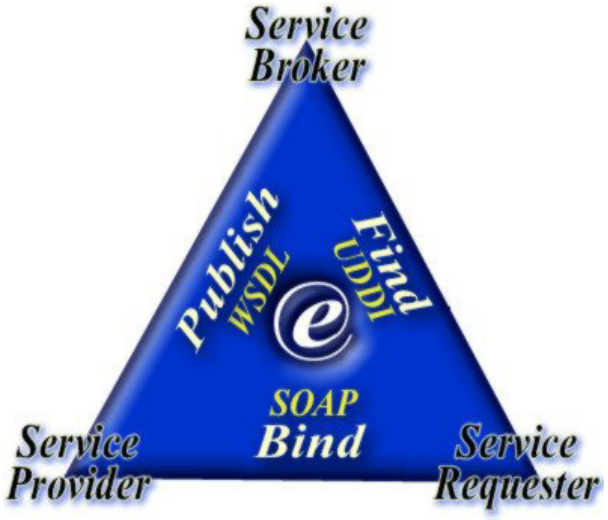
Content used in mashups is typically sourced from a third party via a public interface or API. Other methods of sourcing content for mashups include Web feeds (e.g. RSS or Atom) and JavaScript includes.

The etymology of this term almost certainly derives from its similar use in pop music.

Many people are experimenting with mashups using eBay, Amazon, Google, and Yahoos APIs.

Web Services: Key Functions

- **Service Broker**
- **Owns a searchable repository of service descriptions**
- **Service Providers publish their services to the broker**
- **Service requesters access broker to find services**



Service Provider

- **Owns a group of services**
- **Provides applications as Web Service**
- **Publishes their services**

- **Service Requester**
- **A client that requires a service**
- **Finds a matching service**
- **Invokes the service**

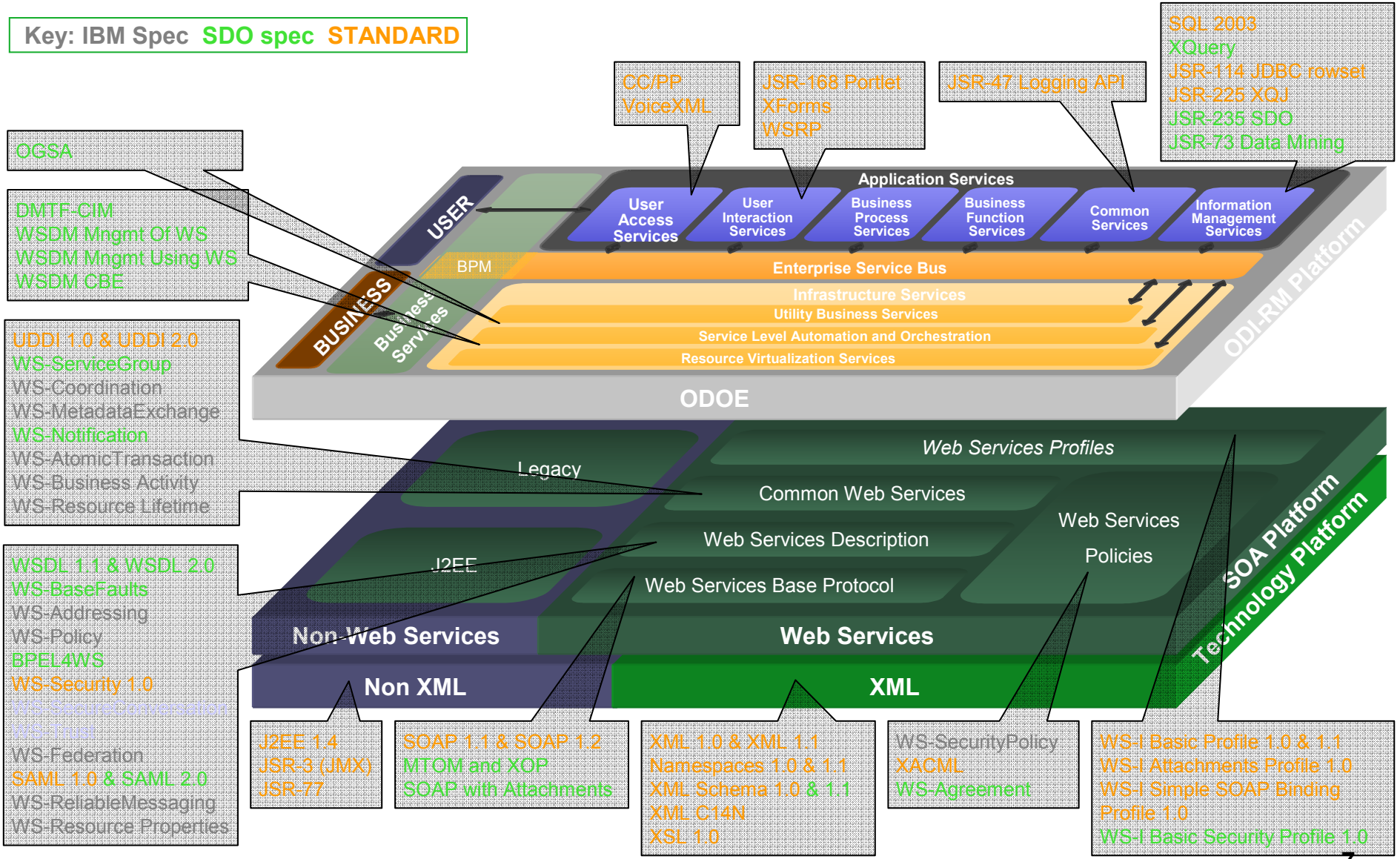
J2EE Standards: JSR 101, JSR 109

Web Services Gateway, UDDI

A quick mention about Standards

- When you read about standards related to Web Services, they usually use acronyms that start with WS, So:
  - WS-Security
  - WS-Policy
  - WS-I Basic Profile
  - WS-TX (WS Transactions)
- Each one will define a specific area of the web services space.
- Most of these relate to the SOAP component.

On Demand Operating Environment Standards



Web Services

- Say you were a store that wanted to sell pencils on the network but you didn't want to setup the payment process. So you need to find someone that will process the payments for you. In this case you'd be a Service Requester. Your online application would have to
  - Lookup the name and location of the service you need to use
- Say your a bank and you want to provide a service to anyone who needs to accept payments, but you don't want to tie your service directly to anyone. In this case you'd be a Service Provider. You would have to
  - Build you service and register it for someone to lookup
- Both of the above need a middle man to handle the directory and support the registration of services and the lookup. This would be the Service Broker. You would have to
  - Make your service available to the providers and requesters

Each of these three services requires a set of standards and definitions that is commonly called Web Services. Some development groups are using parts of the standard, i.e. XML or SOAP over HTTP and they say they are using Web Services

Web Services: What Does It Take?

- A structured way for exchanging information
  - XML - Extensible Markup Language
  - Provides a platform/vendor neutral way to structure data
- A service access protocol
  - SOAP - Simple Object Access Protocol
  - Provides a platform/vendor neutral application communication protocol
- A way to describe services
  - WSDL - Web Services Description Language
  - Uses XML to describe a Web service
- A way to advertise and find out about available services
  - UDDI - Universal Discovery Description and Integration
  - A structured directory or registry used for publishing and finding a Web Service

So what is used for Standards based Web Services

- WSDL -Web Services Description Language
  - Describes the interface to the Service in a XML tagged language
  - Describes the location and port type
  - Usually generated at the end of the service development cycle
    - Can be registered in a UDDI registry for easy access
    - If the Service is registered in a UDDI, the developer can look it up using the development environment and import it.
  - Can be sent to a developer that wants to use the service
    - The developer imports the WSDL into the development tool WSAD (WebSphere Studio Application Developer)
  - Externalizes the service interface and decouples the requester from the service. The requester only needs to have the WSDL. WSDL can be looked up.
  - A lot of Businesses are using WSDL for both web services and for defining interfaces for local calls within a component architecture model, but they're not using UDDI.

Cont….

- SOAP -Simple Object Access Protocol
  - Not necessarily Simple ☺
  - Think of SOAP as the message or payload and there can be different transport protocols, usually HTTP or JMS (MQ).
    - A purist will say it needs to be HTTP to meet the current standards.
  - Some customers use XML/HTTP instead of SOAP/HTTP and refer to it as Web Services.
  - Some customers use SOAP, some use XML, and some use both

Cont….

- UDDI -Universal Description Discovery and Integration protocol
  - Not just a standard or protocol, but a registry.
  - A place where you can register a service if you are a provider and a place to do a lookup if you are a requester.
  - You can use internal or external UDDI registry's.
  - They can be used only during development for cataloging and retrieving service definitions, or can be used during production runtime for dynamic discovery.
  - If you don't use a UDDI, how do you
    - Control versions of the WSDL floating around?
    - Handle version control?
    - Where do you store all the current WSDL related to available services?
  - I have not really seen a big business use UDDI in production, yet

Sample WSDL for Google

```
- <definitions name="GoogleSearch" targetNamespace="urn:GoogleSearch">
  - <!--
       Types for search - result elements, directory categories
    -->
  - <types>
    - <xsd:schema targetNamespace="urn:GoogleSearch">
      - <xsd:complexType name="GoogleSearchResult">
        - <xsd:all>
            <xsd:element name="documentFiltering" type="xsd:boolean"/>
            <xsd:element name="searchComments" type="xsd:string"/>
            <xsd:element name="estimatedTotalResultsCount" type="xsd:int"/>
            <xsd:element name="estimateIsExact" type="xsd:boolean"/>
            <xsd:element name="resultElements" type="typens:ResultElementArray"/>
            <xsd:element name="searchQuery" type="xsd:string"/>
            <xsd:element name="startIndex" type="xsd:int"/>
            <xsd:element name="endIndex" type="xsd:int"/>
            <xsd:element name="searchTips" type="xsd:string"/>
            <xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray"/>
            <xsd:element name="searchTime" type="xsd:double"/>
          </xsd:all>
        </xsd:complexType>
      - <xsd:complexType name="ResultElement">
        - <xsd:all>
            <xsd:element name="summary" type="xsd:string"/>
```

Sample WSDL for Google

Bindings and transport

Input's and Output's

```
<!--
    Binding for Google Web APIs - RPC, SOAP over HTTP
-->
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="doGetCachedPage">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded" namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>
  <operation name="doSpellingSuggestion">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded" namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>
```

Time out for performance

- Interpretation and parsing
  - Java is an interpreted language.
  - SOAP is a parsed message.
  - XML is a parsed protocol.
  - Using UDDI dynamic lookup is not a cheap processes.
- None of this is free, it costs in cycles to do it all.
  - It may be easier for the developer and improve their development time, but it directly affects the performance characteristics of an application.
  - It's a tradeoff. How much of a performance hit can you afford to improve developer productivity and improve reuse.

What are some Business's doing, cont…

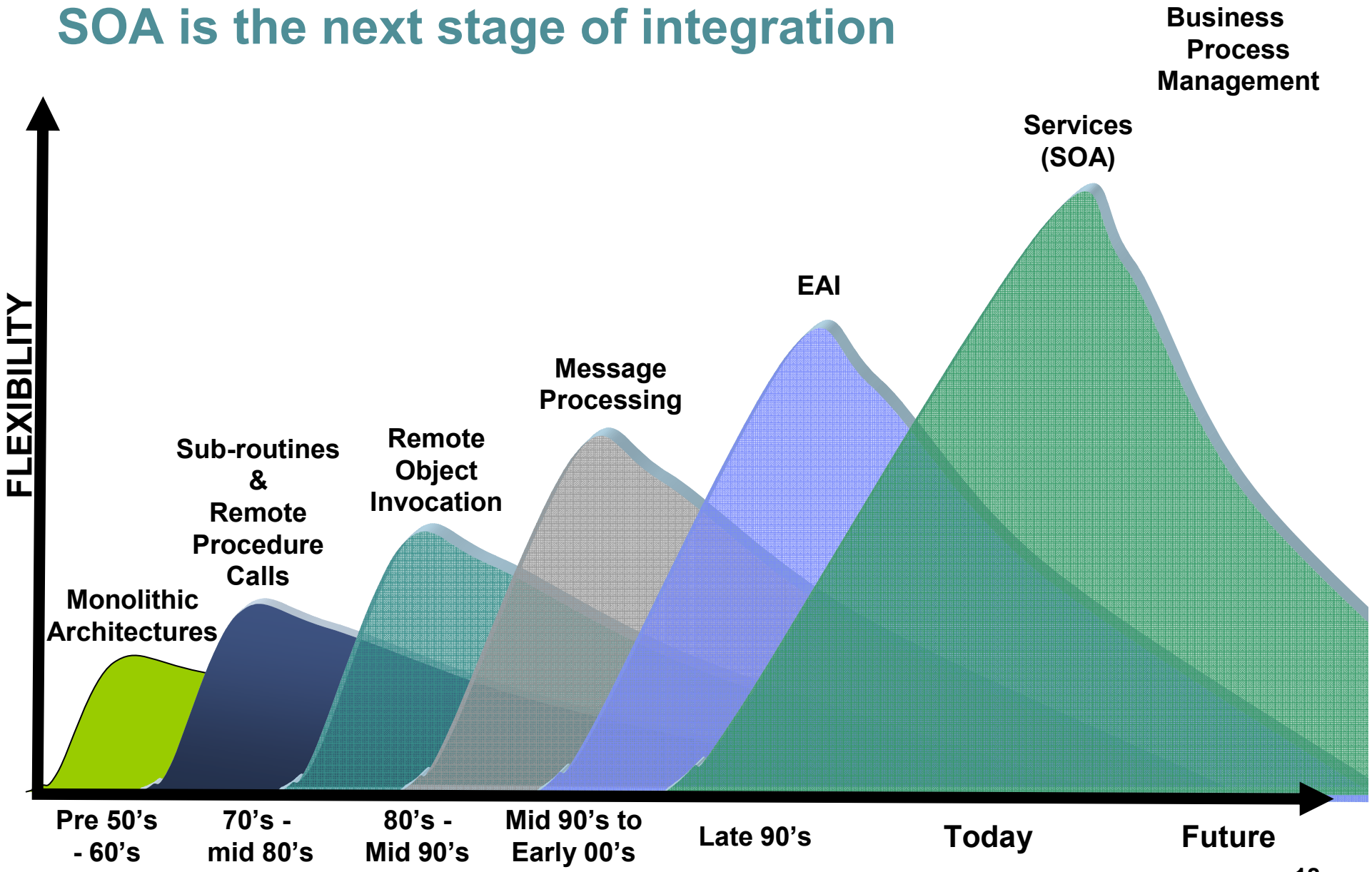
- XML or SOAP over HTTP
  - As an example Banks have rewritten teller based applications to issue service requests from .NET clients to WebSphere front ends to existing IMS or CICS transactions.
  - No UDDI involved
    - WSDL is "sent" to developers
- MQ is still heavily used
  - Normal MQ message format (no XML or SOAP), into CICS
  - MDB's used in WebSphere
  - Used as a migration to Web Services (Phase 1)

So what is an SOA

- **SOA** (Service Oriented Architecture) A system for linking resources on demand. In an SOA, resources are made available to other participants in the network as independent services that are accessed in a standardized way. This provides for more flexible loose coupling of resources than in traditional systems architectures.

- This is not new
  - Business's have been separating components in their application architectures for years.
    - Everything from DB2 stored procedures to load libraries, DLL's etc.
  - The concept of reusable Services maybe new, but it is an "evolution of abstraction", making a transaction, business logic, or function available to multiple clients, and simplifying the use of the service.

SOA is the next stage of integration



What is an Enterprise Service Bus?

An Enterprise Service Bus (ESB) is a flexible connectivity infrastructure for integrating applications and services.

An ESB powers your SOA by reducing the number, size, and complexity of interfaces.

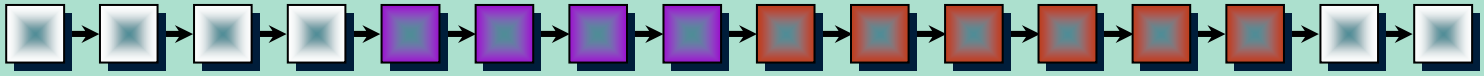
An ESB performs the following between requestor and service

- **ROUTING** messages between services
- **CONVERTING** transport protocols between requestor and service
- **TRANSFORMING** message formats between requestor and service
- **HANDLING** business events from disparate sources

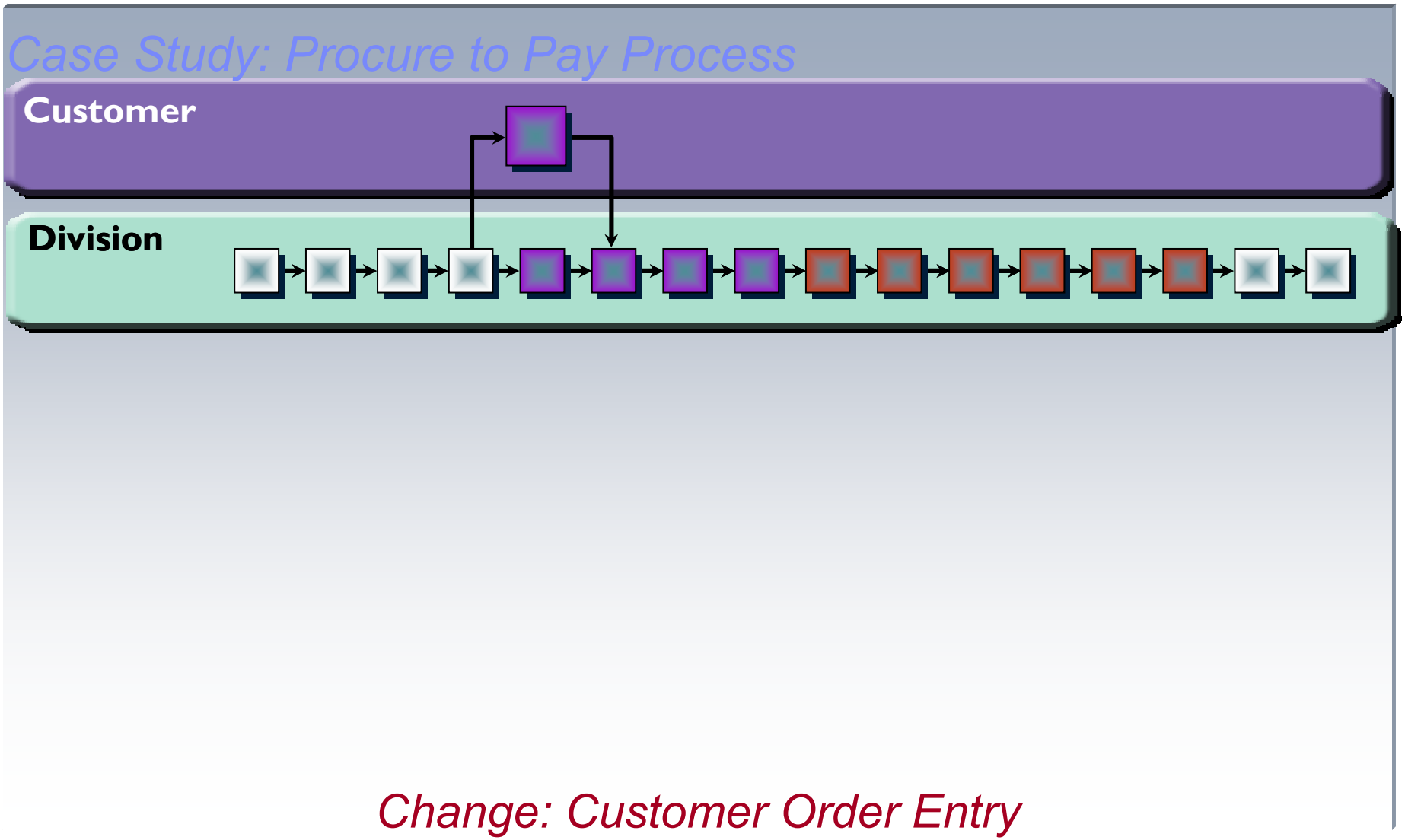
SOA in action

Case Study: Procure to Pay Process

Division

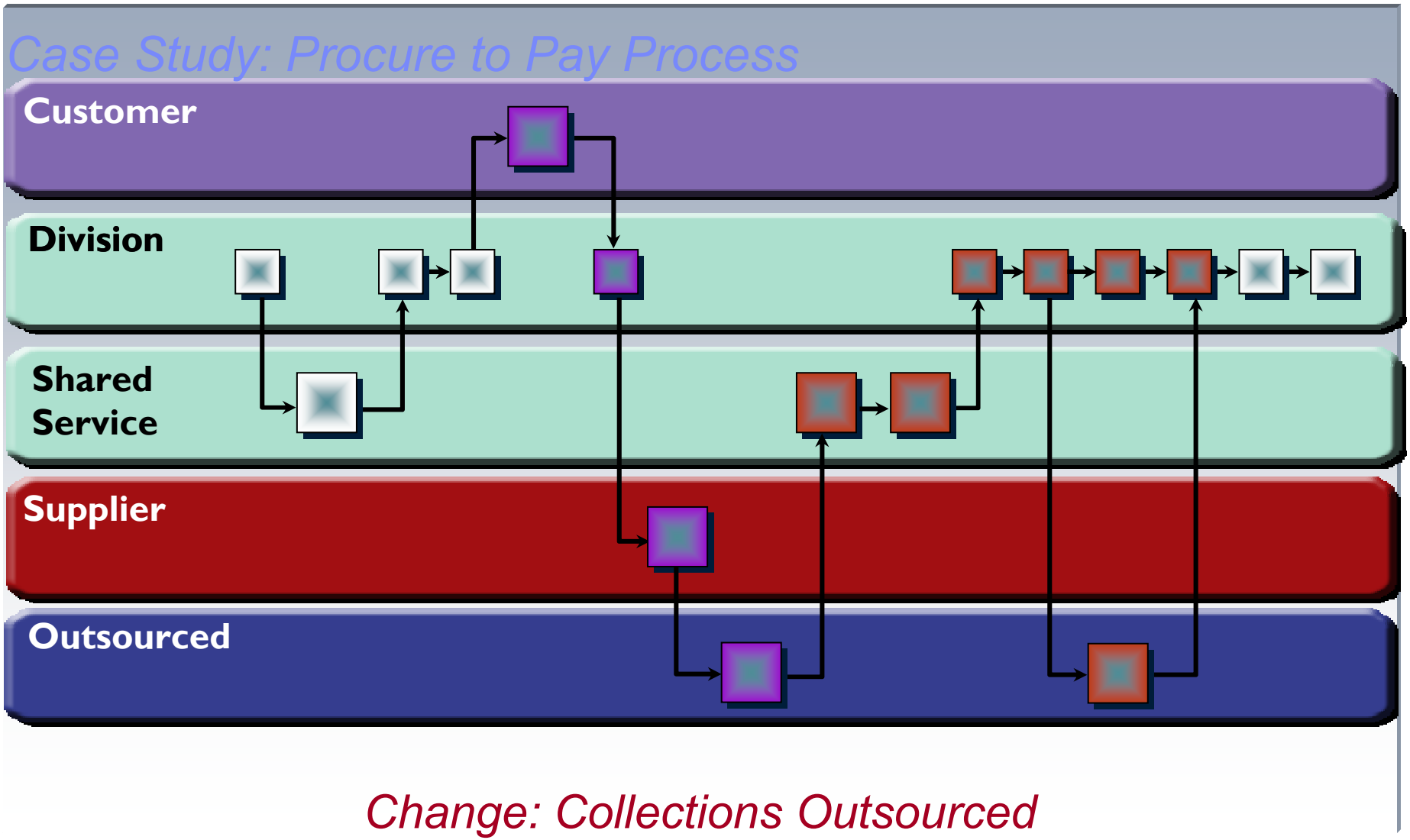


SOA in action



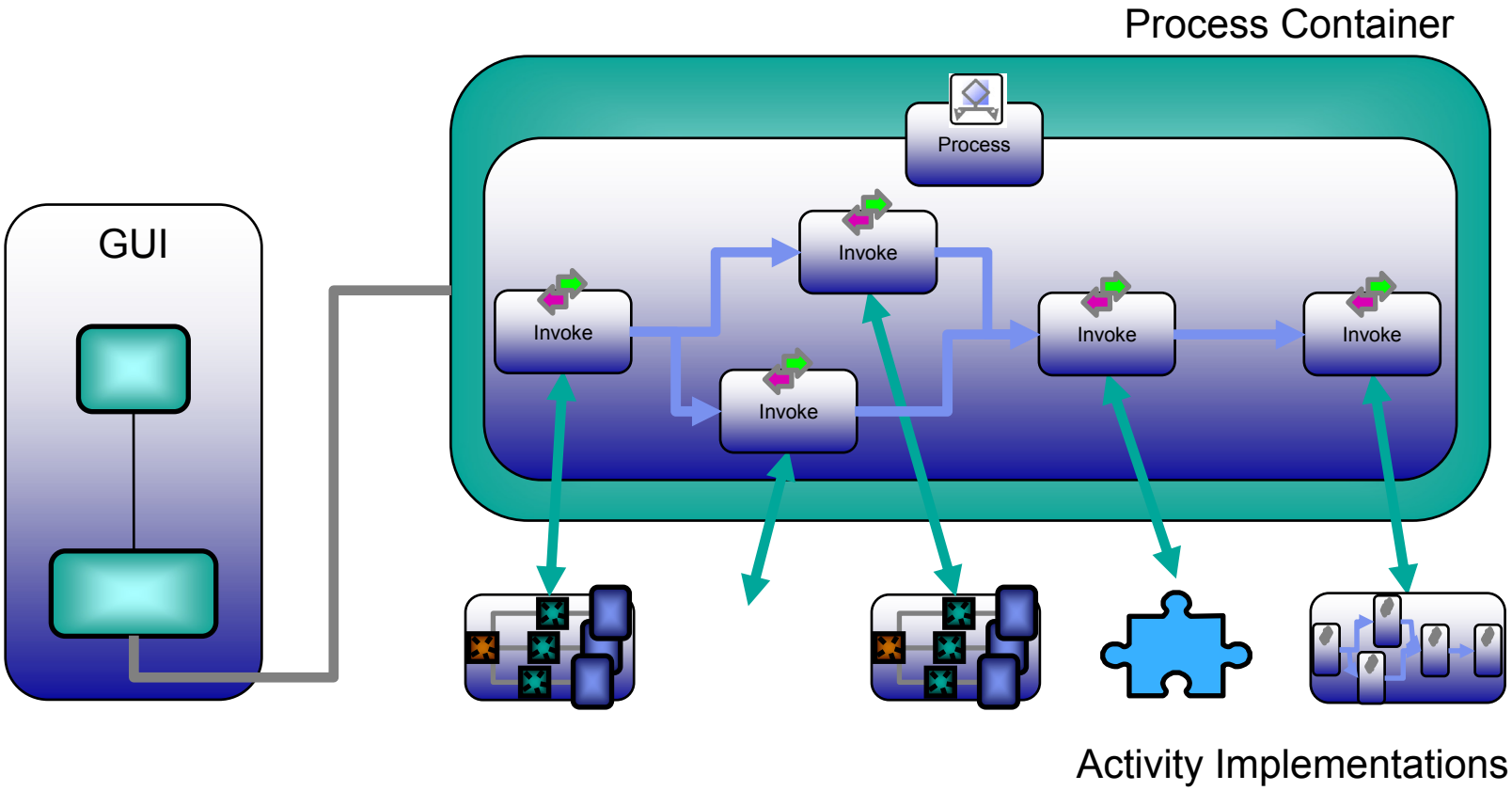
Change: Customer Order Entry

SOA in action



Change: Collections Outsourced

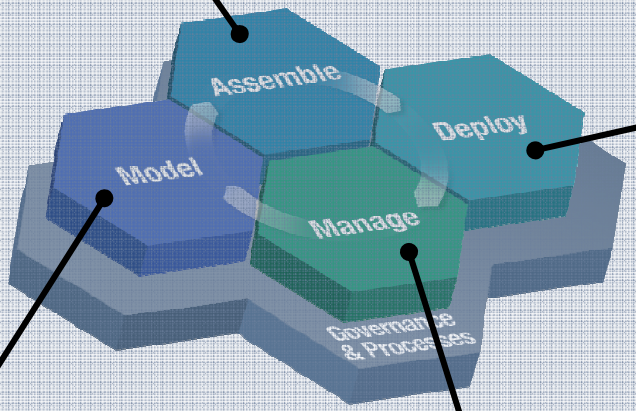
Business Process Choreography



So how does this relate to WebSphere?

- The SOA Foundation builds on and extends the WebSphere Core
- WebSphere is the heart of IBM's SOA Core Foundation

WebSphere Integration Developer
Rational Application Developer
Workplace Designer



Process:
WebSphere Process Server
WebSphere ESB & Message Broker
WebSphere Partner Gateway & Adapters
People:
WebSphere Portal
WebSphere Everyplace Deployment
Workplace Collaboration Services
Information:
WebSphere Information Integrator
Application Infrastructure:
WebSphere Application Server & XD

WebSphere Business Modeler
Rational Software Architect

WebSphere Business Monitor
Tivoli Composite Application Manager
Tivoli Identity & Access Manager

Inhibitors to Web Services and SOA

- People Roles within the corporation
  - From Architects thru support
- Process
  - At times “the business” does not really understand their complete business process?
- Funding
  - If the first set of reusable services cost 1.5 times what it would cost for the same thing, non-service based, what business unit will fund it? because that’s where the application funding is
- Governance
  - Internal standards

If all the vendors, delivered all the technology today, to support what SOA promises, would you be ready?

Questions?

Quick Glossary

- **HTTP** (HyperText Transfer Protocol) The Web's communication standard. Stabilized at HTTP/1.1, it defines the universal mechanism for exchanging application-level messages between Web devices. All web services run over HTTP.
- **URI** (Uniform Resource Identifier) The address of an Internet resource. A URI is the unique name used to access the resource. It is not necessarily a specific file location (it may be a call to an application or a database, for example), which is why it is preferred over the similar acronym URL (Uniform Resource Locator).
- **XML** (eXtensible Markup Language) The data tagging language of web services. XML is not so much a language as a standardized set of rules for adding structure to any form of data using a system of markup tags. Anyone can create their own markup vocabulary (called an XML Schema), and XML ensures that the structure will be intelligible to anyone else who consults the XML Schema document. More importantly, referring to an XML Schema enables XML-aware software to automatically manipulate the data without needing advance knowledge of the structure.
- **loose coupling** The friction-free linking enabled by web services (or any SOA). Loosely coupled services, even if they use incompatible system technologies, can be joined together on demand to create composite services, or disassembled just as easily into their functional components. Participants must establish a shared semantic framework to ensure messages retain a consistent meaning across participating services.

Copied from http://looselycoupled.com/glossary/

Quick Glossary (Cont.

- **SOAP** (Simple Object Access Protocol) The standard for web services messages. Based on XML, SOAP defines an envelope format and various rules for describing its contents. Seen (with WSDL and UDDI) as one of the three foundation standards of web services, it is the preferred protocol for exchanging web services, but by no means the only one; proponents of REST say that it adds unnecessary complexity.

- **UDDI** (Universal Description, Discovery and Integration protocol) A directory model for web services. UDDI is a specification for maintaining standardized directories of information about web services, recording their capabilities, location and requirements in a universally recognized format. Seen (with SOAP and WSDL) as one of the three foundation standards of web services, UDDI is currently the least used of the three.

- **WSDL** (Web Services Description Language) The standard format for describing a web service. Expressed in XML, a WSDL definition describes how to access a web service and what operations it will perform. Usually pronounced "whizz-dul" (to rhyme with 'whistle'), WSDL is seen (with SOAP and UDDI) as one of the three foundation standards of web services.

Copied from http://looselycoupled.com/glossary/

RETURN TO INDEX