



# Technical Track Session

## Service-Oriented Architecture

Terry Woods





# Agenda

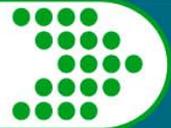
- A little history
- What is Service-Oriented Architecture?
- How do you build a Service-Oriented Architecture Solution?
- What is an Enterprise Service Bus?
- Summary





# A Little History

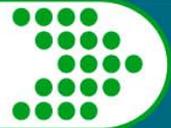




# In the Beginning ...

- Computers were the domain of scientists for scientists
- Business and Government discovered the value of computing
- Business requirements needed to be captured and programmed





# Some Past Attempts ...

- COBOL – It was designed to be business-problem oriented, machine-independent, and capable of continuous change and development.
- SQL – The SQL "language" allows anyone with a computer terminal to access and use relational databases. Originally thought to be an end-user tool.
- SAD – Structured Analysis and Design (Yourdon/DeMarco)
- IE – Information engineering (design by pictures) James Martin
- OOAD – Object-Oriented Analysis and Design





# Revolution ...

- Revolution

- COBOL
- Basic programming language
- C programming language
- SQL
- Structured analyses and design
- Electronic data exchange

- Evolution

- Web Services
- Information Engineering
- CASE
- C++
- Java
- Object-Oriented Analysis and Design
- HTML
- Common Record (XML)

- Ecosystem

- Standards
- SOA





# What is Service-Oriented Architecture?





# What is Service-Oriented Architecture?

- Service-Oriented Architecture (SOA) is an architectural style. Applications built using an SOA style deliver functionality as services that can be used or reused when building applications or integrating within the enterprise or trading partners.

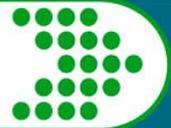




# SOA

- Uses open standards to integrate software assets as services
- Standardizes interactions of services
- Services become building blocks that form business flows
- Services can be reused by other applications





# What is a Service?

- A service is a reusable component that can be used as a building block to form larger, more complex business-application functionality.
- A service may be as simple as “get me some person data,” or as complex as “process a disbursement.”





# What is a Service?

- A service provides a discrete business function that operates on data. Its job is to ensure that the business functionality is applied consistently, returns predictable results, and operates within the quality of service required.





# What is a Service?

- How the service is implemented, and how a user of the service accesses it, are limited only by the SOA infrastructure choices of the enterprise.
- From a theory point of view, it really doesn't matter how a service is implemented.





# Characteristics of a Service

- Supports open standards for integration: Although proprietary integration mechanisms may be offered by the SOA infrastructure, SOA's should be based on open standards. Open standards ensure the broadest integration compatibility opportunities.





# Characteristics of a Service

- Loose coupling: The consumer of the service is required to provide only the stated data on the interface definition, and to expect only the specified results on the interface definition. The service is capable of handling all processing (including exception processing).

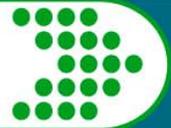




# Characteristics of a Service

- **Stateless:** The service does not maintain state between invocations. It takes the parameters provided, performs the defined function, and returns the expected result. If a transaction is involved, the transaction is committed and the data is saved to the database.





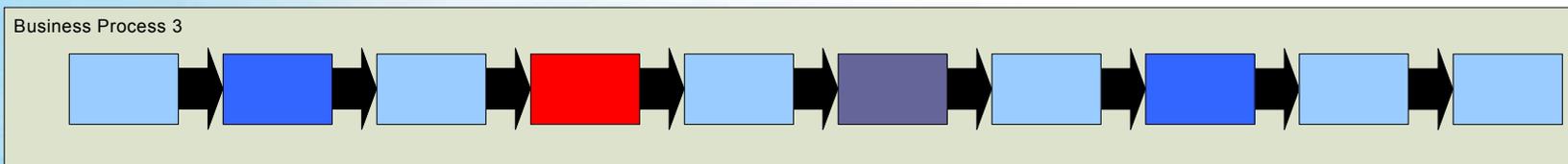
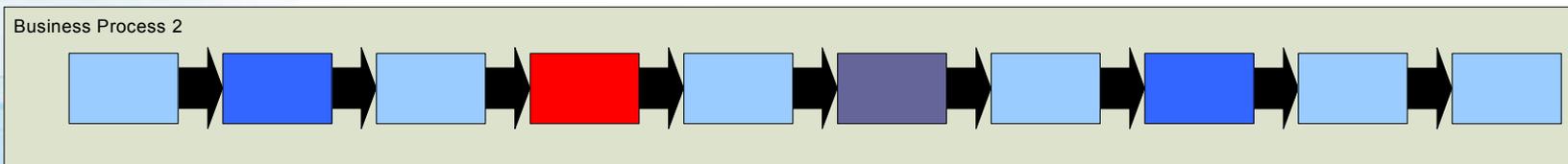
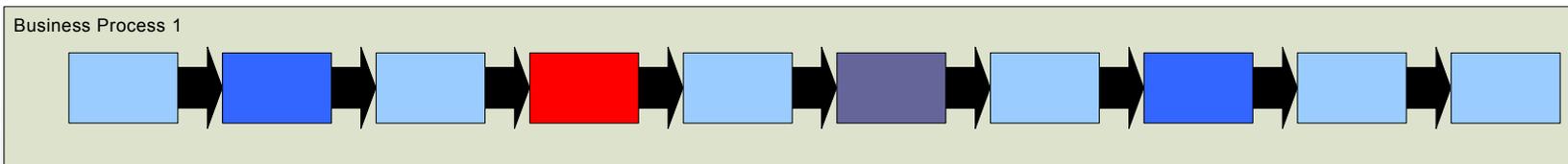
# Characteristics of a Service

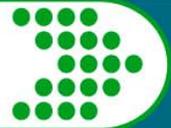
- Location agnostic: Users of the service do not need to worry about the implementation details for accessing the service. The SOA infrastructure will provide standardized access mechanisms with service-level agreements.





# Legacy Business Process





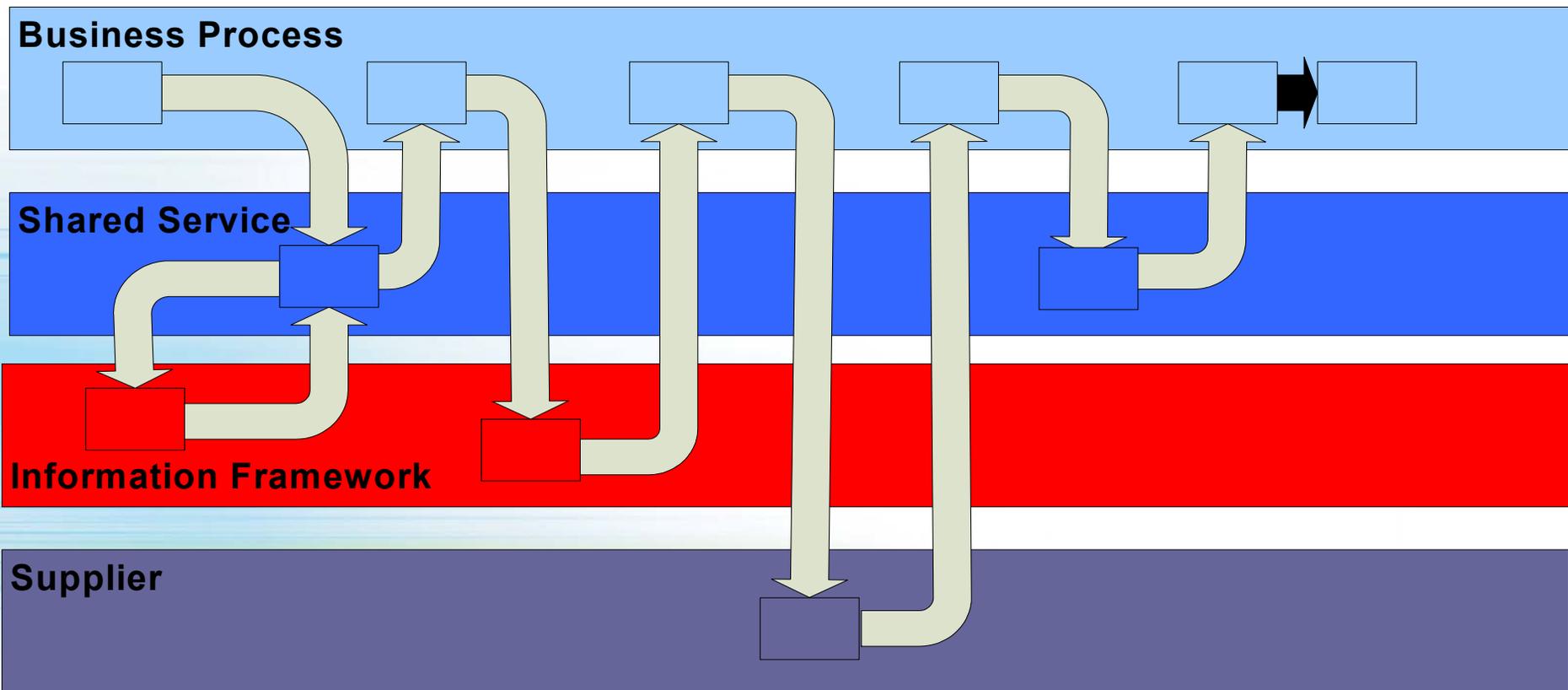
# Architectural Choke Point

- Monolithic
- Extremely complex
- Very tightly coupled
- Difficult to find clean integration points
- Lack of standards makes it difficult to integrate
- Rigid architecture makes even small changes complex and costly





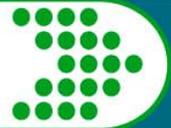
# SOA Business Process





# How Do You Build a Service-Oriented Architecture Solution?





# IT Infrastructure is an Ecosystem

- COBOL
- C++
- Java
- J2EE
- Networks
- TCP/IP
- Web Service
- Mid-Tier Servers
- .NET
- Messaging
- HTML
- Operating Systems
- CICS
- Routers
- Databases
- Mainframes





# Managing the Ecosystem

## Technology Management

- Product duplication
- Standards
- Product/Vendor lock-in
- Support skill sets

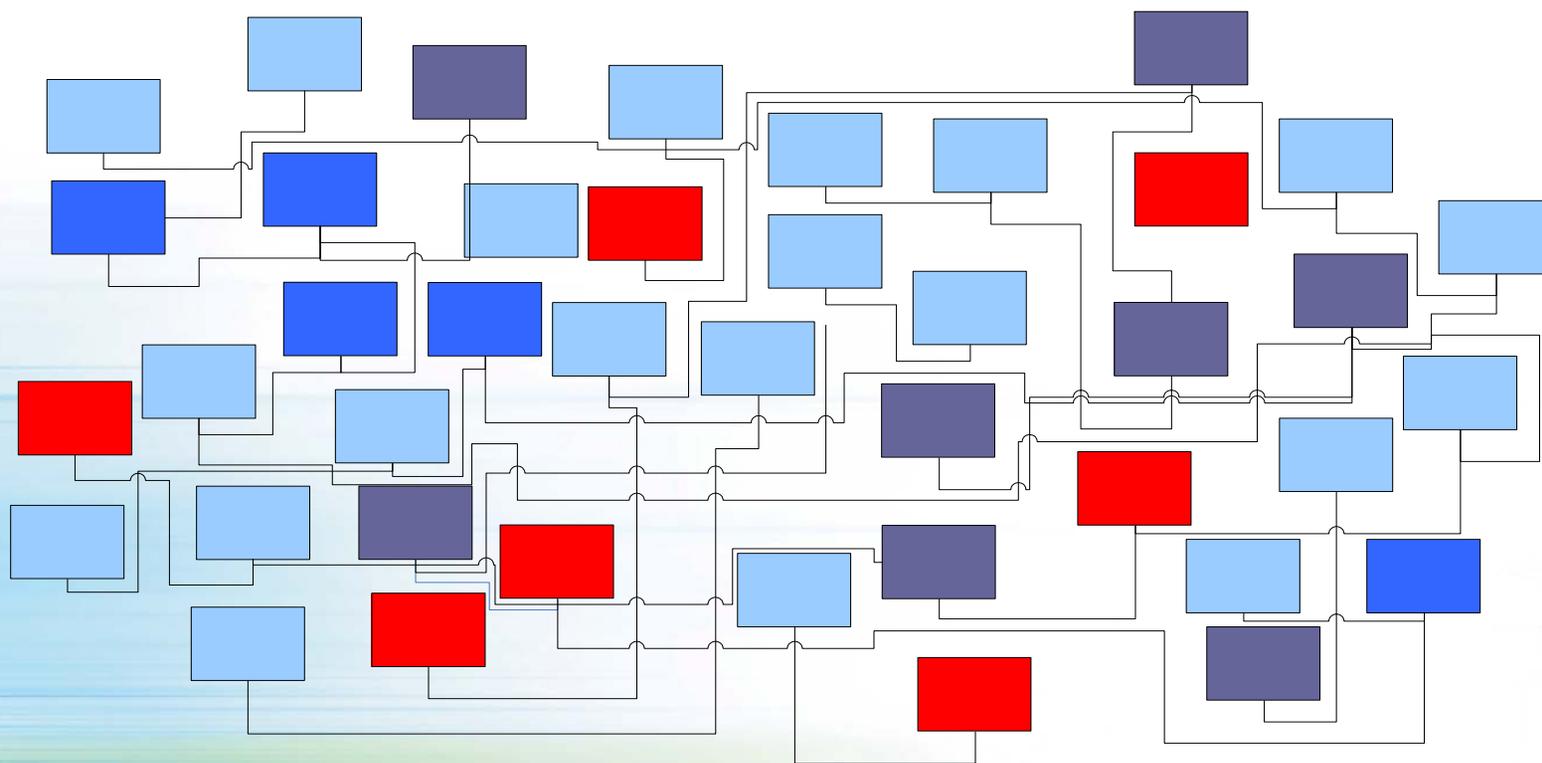
## Version Management

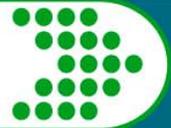
- Managed Evergreening
- Limited versions in production



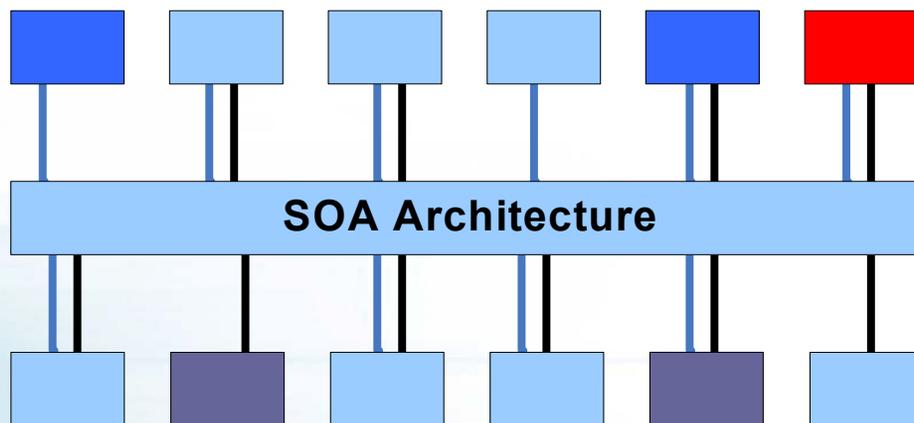


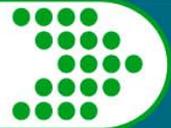
# Legacy Integration





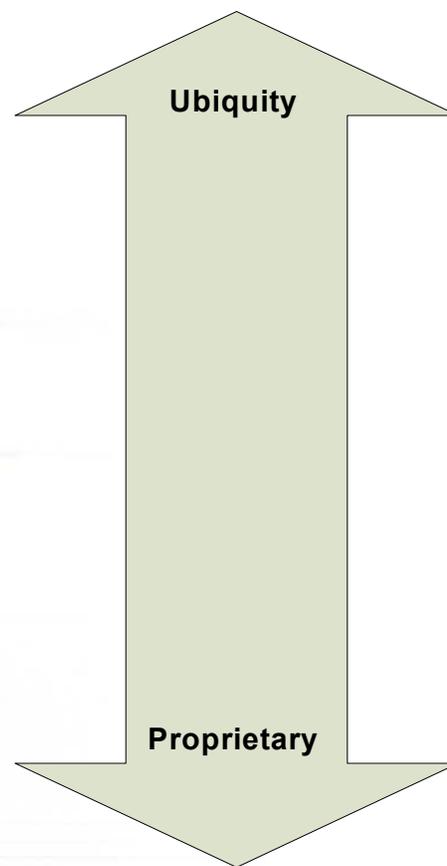
# SOA Integration





# Integration Stack

- Industry Standards
- Language Standards
- Custom API's
- Native Interfaces





# Enterprise Service Bus





# What is an Enterprise Service Bus (ESB)?

- An enterprise service bus is an infrastructure used for building compound applications
- The enterprise service bus is the glue that holds the compound application together
- The enterprise service bus is an emerging style for integrating enterprise applications in an implementation-independent fashion
- An enterprise service bus can be thought of as an abstraction layer on top of an Enterprise Messaging System





# Key Characteristics of an ESB

- Streamlines development
- Supports multiple binding strategies
- Performs data transformation
- Intelligent routing
- Real time monitoring
- Exception handling
- Service security





# What is an ESB?



In its simplest form:

An ESB delivers a message from one point to another.

Don't EAI's already do this?



# EAI Limitations

- An EAI can be a tightly coupled solution
- An EAI can be location-specific
- An EAI can be based on proprietary transport technology



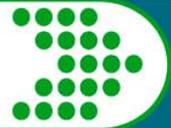


# Let's Walk Through the ESB

- Get Person Data
- Use Standard Student Identification Method (SSIM) identifiers to find the person
- Return the Person Data

This is a fictional-use case, and is only intended to illustrate how an ESB would provide a service.





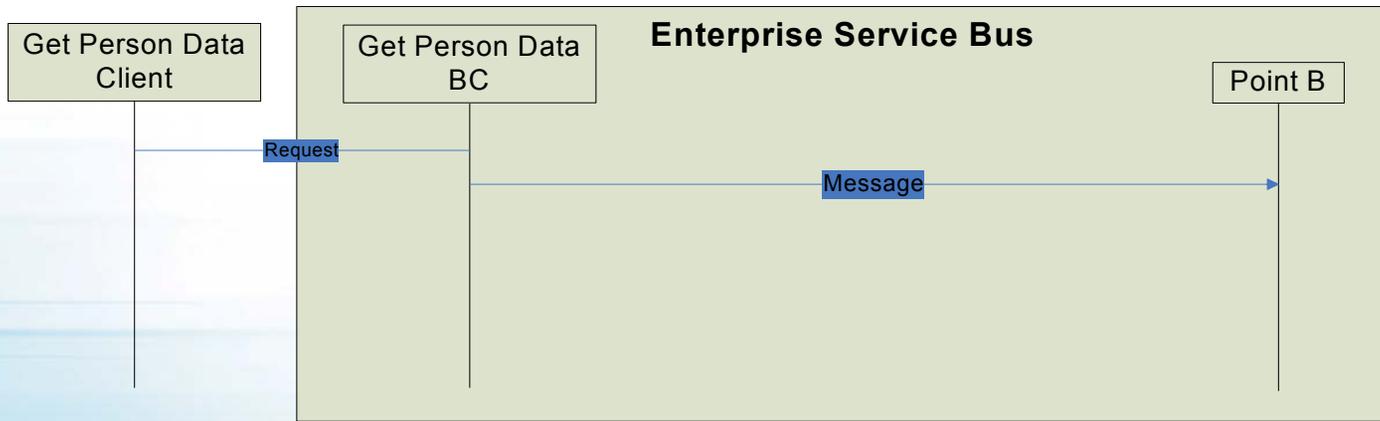
# The ESB Boundaries



The ESB (in its simplest form) is responsible for getting a message from point A to point B.



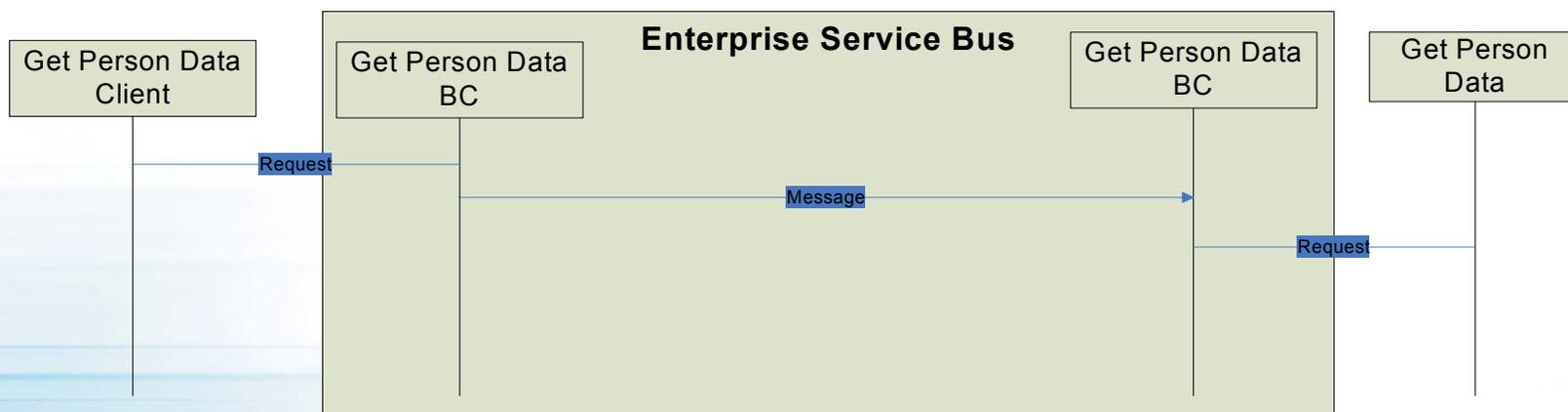
# Get the Message on the Bus



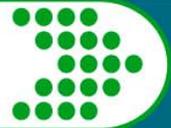
A binding component “speaks” the service’s protocol, which happens to be SOAP over JMS.



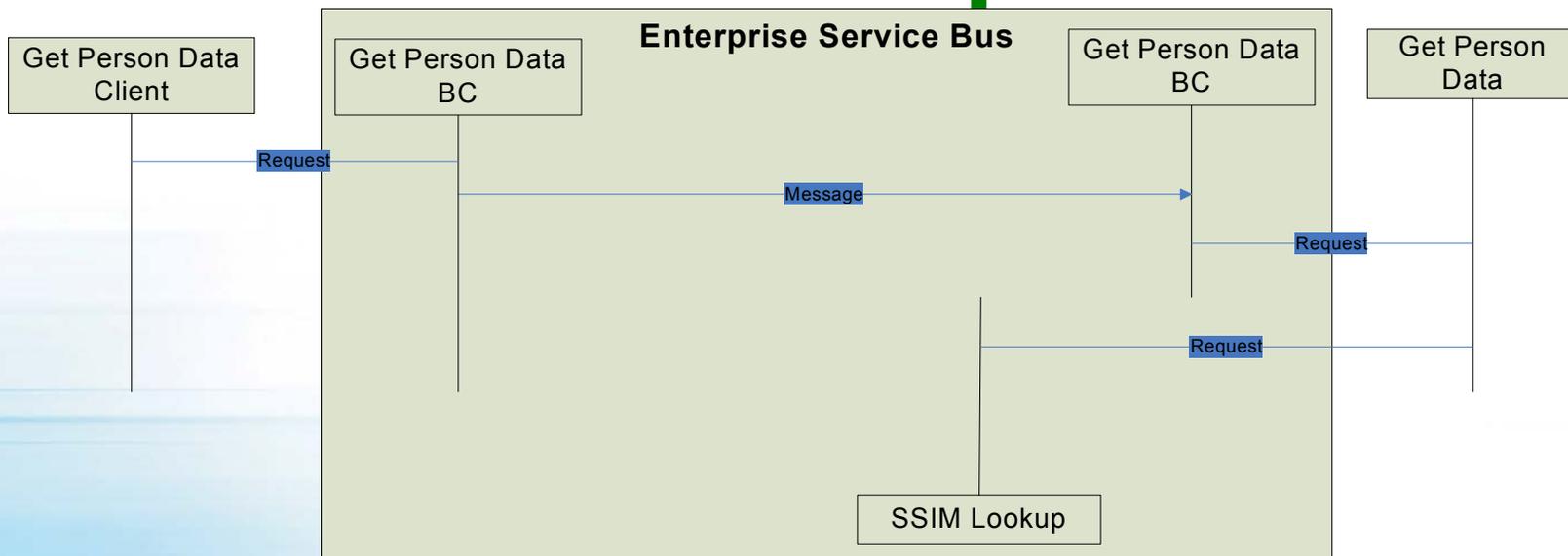
# Perform the Person Read



The request is now routed to the Get Person Data Service, which will perform the business logic.



# Do the SSIM Lookup



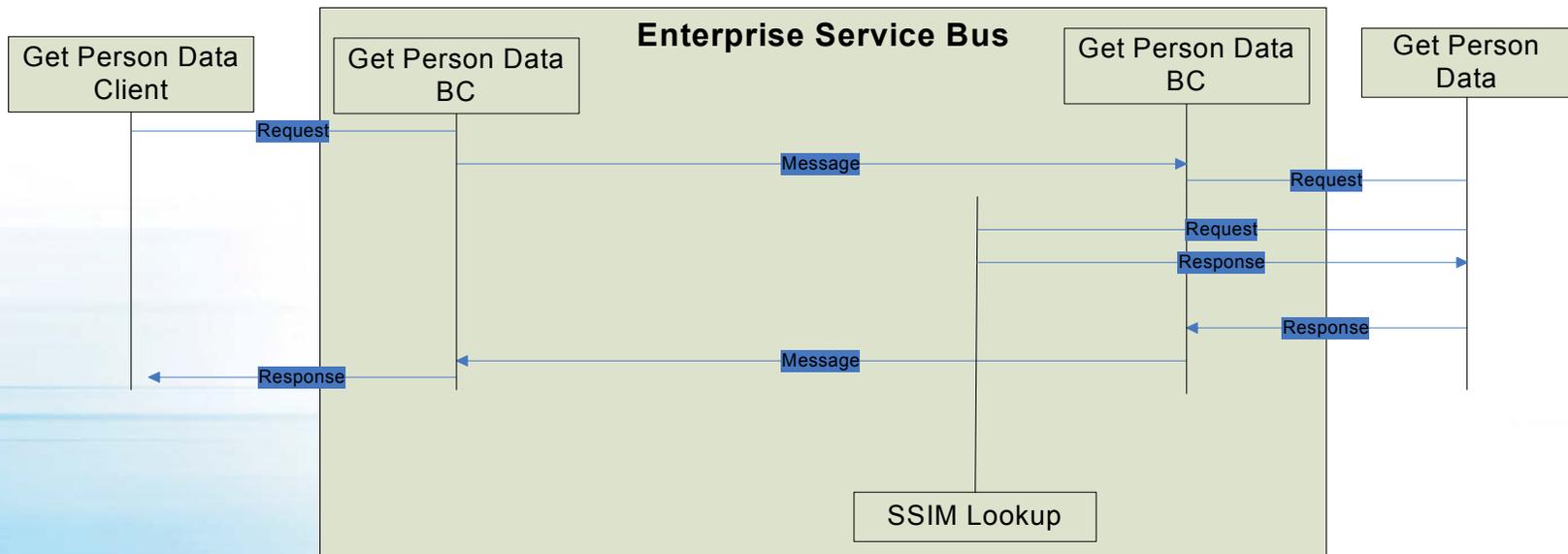
A call is made to the SSIM service to perform a lookup of the Student Identifier (SID). The SSIM service lives inside the bus.

Note: The SSIM binding components are not shown so the diagram can remain simple.





# Return the Person Data



The process is reversed, returning the response to the requester.





# ESB Summary

- Loose Coupling
- Location Transparency
- Transport Neutral





# Defining the Message

- Web Services Description Language
- Open Standard for describing Interfaces to Services (<http://www.w3.org/TR/wsdl>)
- Characteristics
  - Describes data expected to be sent and received
  - Describes what the service can do
  - Describes how to reach the service
- WSDL description is an XML document

Services

Bindings

Port Types

Operations

Messages





# Message-Exchange Patterns

- **One-way.** The endpoint receives a message.
- **Request-response.** The endpoint receives a message, and sends a correlated message.
- **Solicit-response.** The endpoint sends a message, and receives a correlated message.
- **Notification.** The endpoint sends a message.



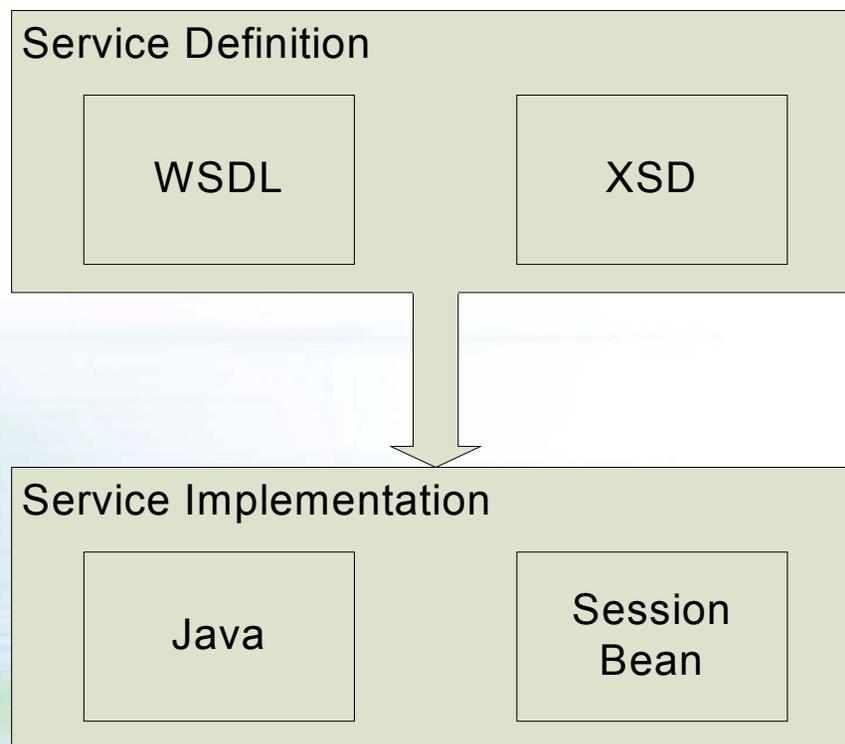


# Building a Compound Application





# The Ingredients





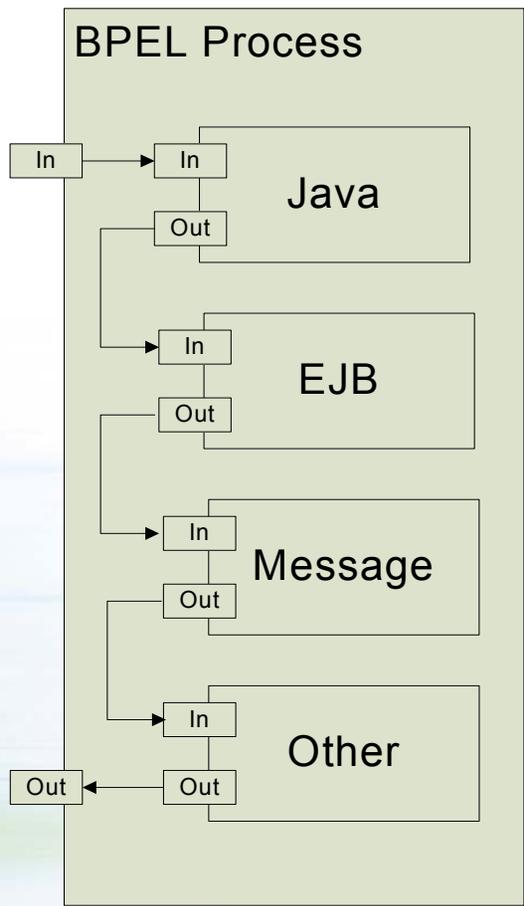
# Business Processes Flow

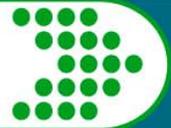
- Business processes are a set of activities, supported by services, that support a particular business activity.
- Business processes are business services built using other business services.
- Business Process Execution Language (BPEL) is a specification for describing business processes in a portable XML format. BPEL is widely supported in both commercial and open source products.
- BPEL defines how services interact to form complex business process. It provides a unit of work context, fault handling, and compensation (transaction rollback).





# BPEL

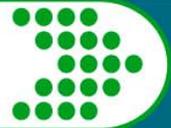




# What BPEL does ...

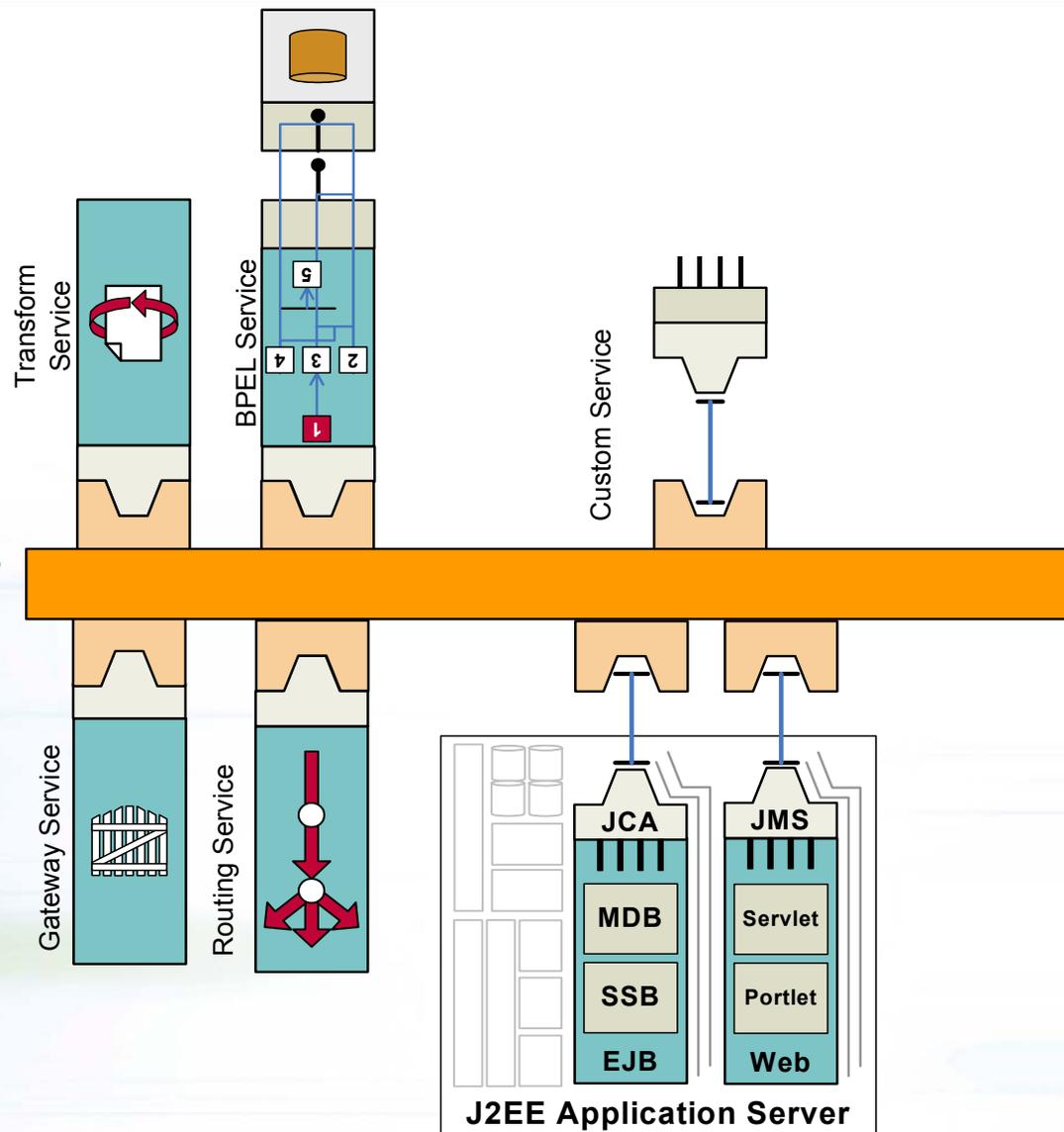
- BPEL binds services together to form larger complex business services
- Control Flow (branch, loop, parallel)
- Asynchronous correlation
- Transaction support, Units of Work
- Compensation

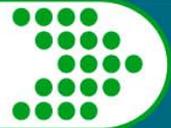




# Sample ESB

- Custom Services
- Transformation Services
- Orchestration
- Routing
- Application Server





# Summary

- What is Service-Oriented Architecture?
- How do you build a Service-Oriented Architecture solution?
- What is an Enterprise Service Bus?





# Contact Information

We appreciate your feedback and comments. We can be reached at:

Phone: 202 377 3023

Email: [Terry.Woods@ed.gov](mailto:Terry.Woods@ed.gov)

