



Java is a trademark of Sun Microsystems, Inc.



Model Driven Solutions
Where Business Meets Technology

JavaOneSM

Enhancing the Role of a Large
US Federal Agency as an
Intermediary in the Federal
Supply Chain via a Service
Registry and a JBI-based ESB

Walt Melo and Wen Zhu

Model Driven Solutions
www.modeldriven.com

Agenda

- > Context
- > Case Studies
 - Definitions
 - The Service Registry Experience
 - The JBI ESB Experience
- > Advanced topics

Environment

- > Work was performed at a large US federal agency that acts as an intermediary in the federal supply chain
- > Model Driven Solutions is a leading provider of professional services and products that leverage *Enterprise Service Oriented Architecture (SOA)*, *Model Driven Architecture (MDA)* and the *Semantic Web* techniques and standards.

Challenges

> Business Challenges

- Improve the way its Suppliers, Vendors, and Consumers publish and discover services
- Enhance Enterprise Architecture Maturity
- Improve SOA Governance to better manage its service portfolio
- Respond better and faster to change
- Improve the enforcement of government policies
- Low TCO and High ROI

> Technical Challenges

- Improve integration of new services and legacy applications
- Improve the visibility of planned and existing services
- Better manage the impact of change
- Improve the life-cycle management of SOA assets, e.g., service descriptions
- Enhance its understanding of all its services
 - Purpose and behavior / Business processes
 - Service level agreements
 - Dependencies

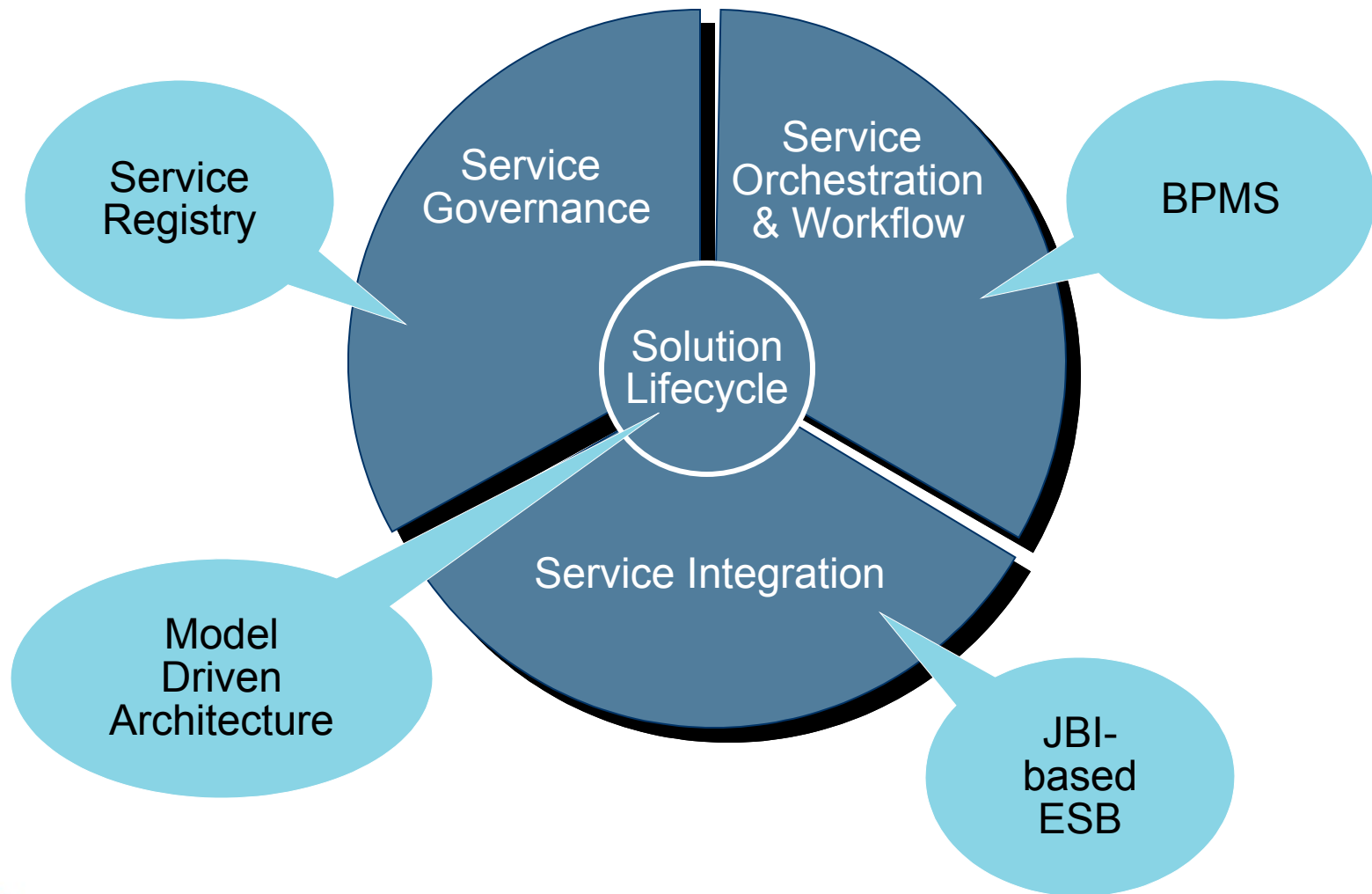
New Administration, Open Government

Government should be
... Transparent
... Participatory
... Collaborative
-- President Obama



SOA is a key enabler

Enterprise SOA as a Solution Approach



Definition: Service Registry

> What is SOA?

- An architectural style for a community of providers & consumers of services to achieve mutual value.
- Business and Technology Levels

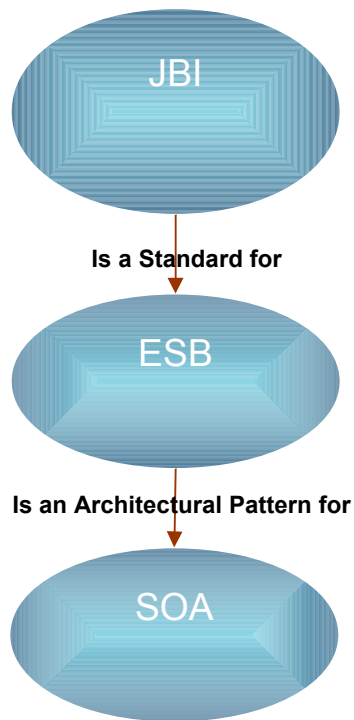
> What is a technology service?

- “a service is a modular piece of software (a service provider) with a well-described interface that can be activated by another modular piece of software (a service consumer).” (Gartner, 2005)

> What is a Service Registry?

- A service registry is a software component which allows an organization to catalog and reference SOA assets required to support the deployment and use of services.

Definition: Java Business Integration (JBI)



> JBI

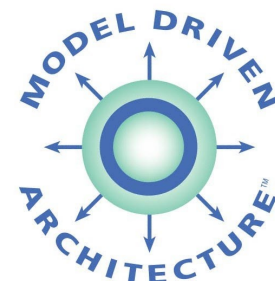
- Defined with in Java Community Process (JCP) as JSR 208
- JBI Specification 1.0 published in 2005
- Standard basis for a Java based ESB

> ESB

- A pattern of middleware that unifies and connects services, applications and resources

More Definitions

- > **Unified Modeling Language (UML)™**
“provides a key foundation for OMG's, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution.”
- > **Model Driven Architecture (MDA)®**
“[t]he three primary goals of MDA are portability, interoperability and reusability through architectural separation of concerns.”
- > **Service Oriented Modeling Language (SoaML)**
“support the activities of service modeling and design and to fit into an overall model-driven development approach.”
- > **Semantic Web**
“provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.”



Agenda

- > Context
- > Case Studies
 - **The Service Registry Experience**
 - The ESB JBI Experience
- > Advanced topics

Service Registry Experience

> Product: Sun Service Registry

- Product selected by OCIO
- Bundle with Sun Java Enterprise System
- Based on freebXML (ebXML Registry)
- ebXML centric
- Partial support for UDDI



Features

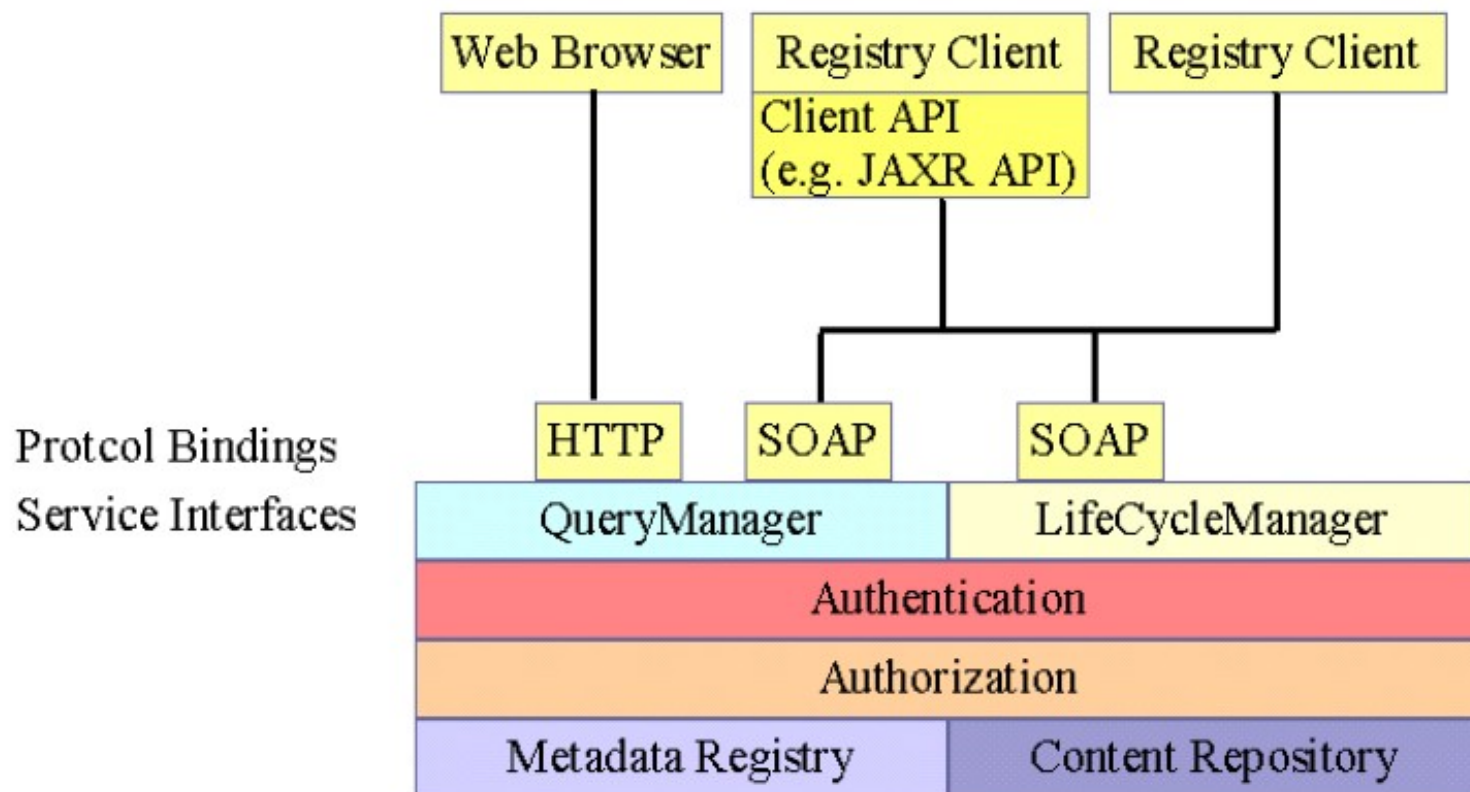
- > Standards
 - Provides standards-based way to manage information assets
- > Classification and affiliation
 - Manages user-defined organization of and relationships among content and metadata
- > Validation and cataloging
 - Enforces conformance of content to user-defined standards
- > Lifecycles
 - Governance capabilities for managing information asset lifecycles
- > Query
 - Provides flexible mechanisms for content discovery
- > Security
 - Manages secure access to information assets
- > Event notification
 - Facilitates event-based delivery of information to appropriate personnel or systems
- > Federation
 - Enables integration of information assets across organizational boundaries

> Encourage reuse

> Enhance Governance

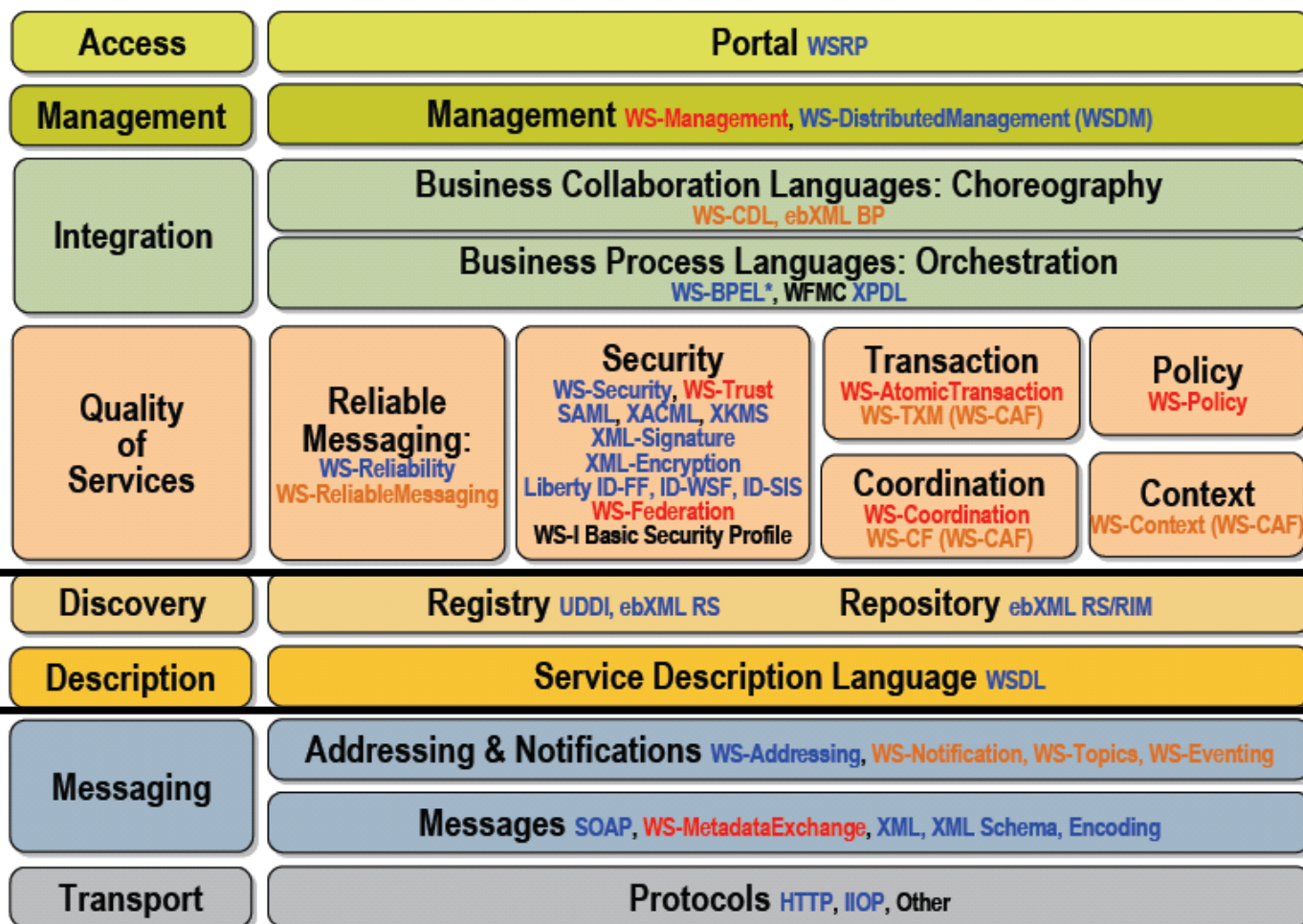
• **Lower TCO**
• **Higher ROI**

Service Registry: High Level Architecture



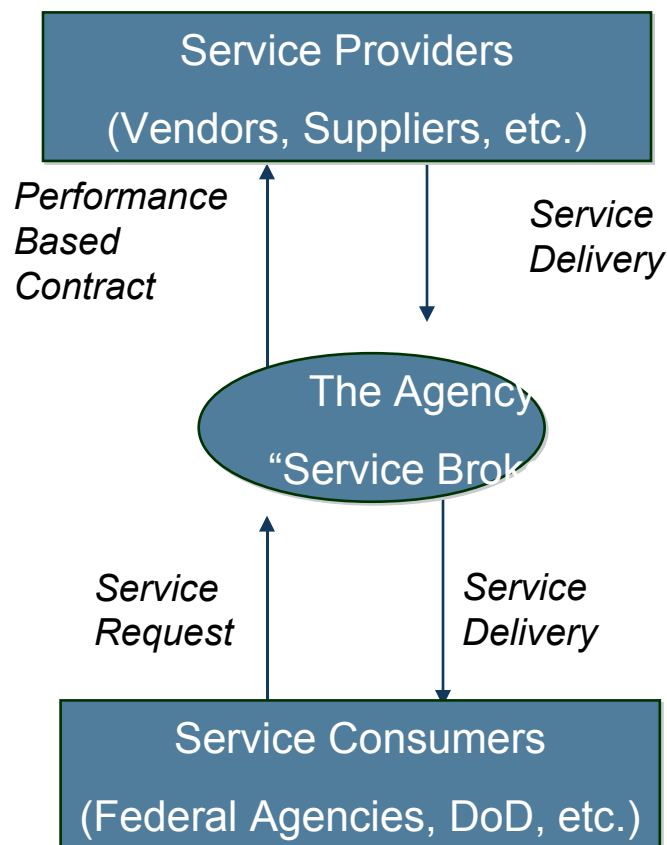
Source: OASIS, ebXML Registry Service, 2007

SOA Standards: Where the Service Registry Fits In

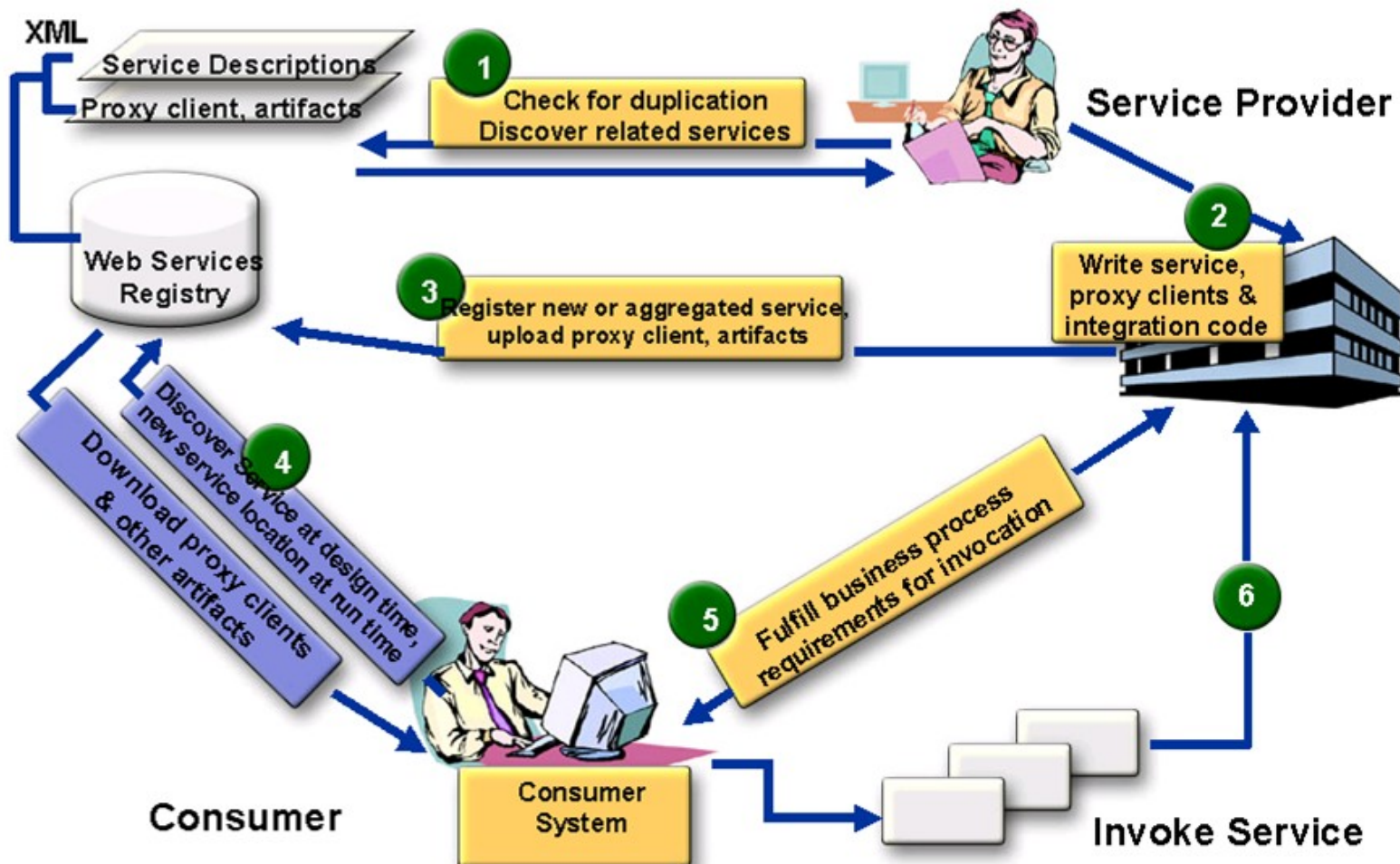


The Agency As An Intermediary in the Federal Supply Chain

- > The agency is both a provider and a consumer of services
- > This agency as a Service Broker
 - Eases Access to Services
 - Accounting, Measurement across Services
 - Unified Face to the Customer
 - Drives Reuse
 - Increases Security
 - Improves Policy Management and Enforcement
 - Enhances SLA Management & Administration
 - Allows Shared Services Financing



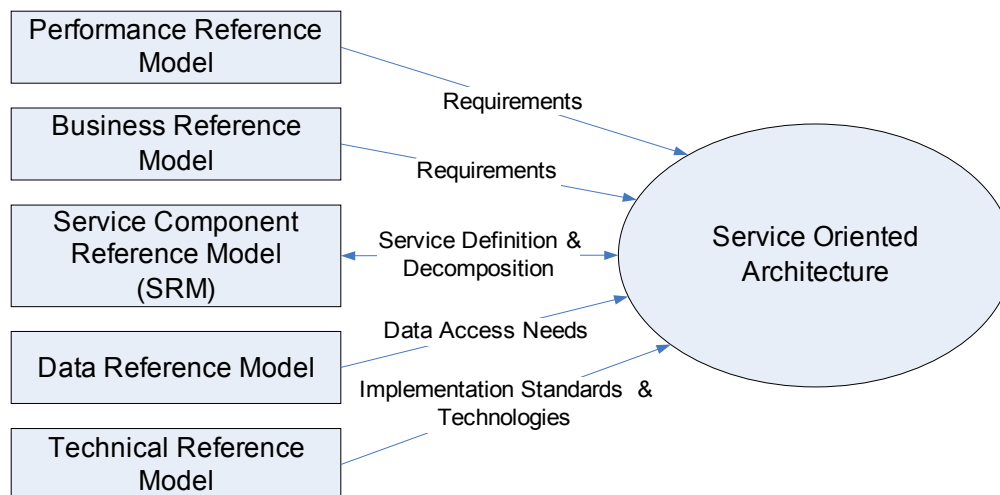
Enterprise Use Scenario



source: Sun

The Role of Federal Enterprise Architecture (FEA)

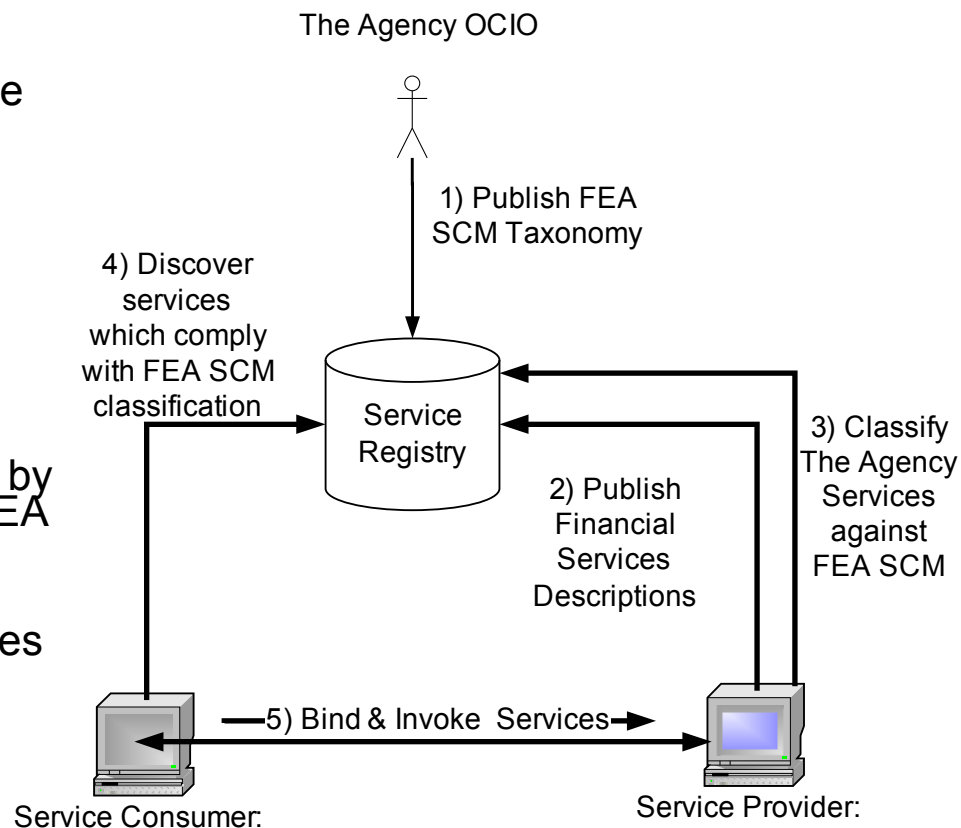
- > FEA aligns Federal business functions and IT via a set of common (reference) models
- > FEA Reference Models provide a foundation for SOA
- > FEA Service Component Reference Model (SRM) provides service definition and decomposition for a SOA



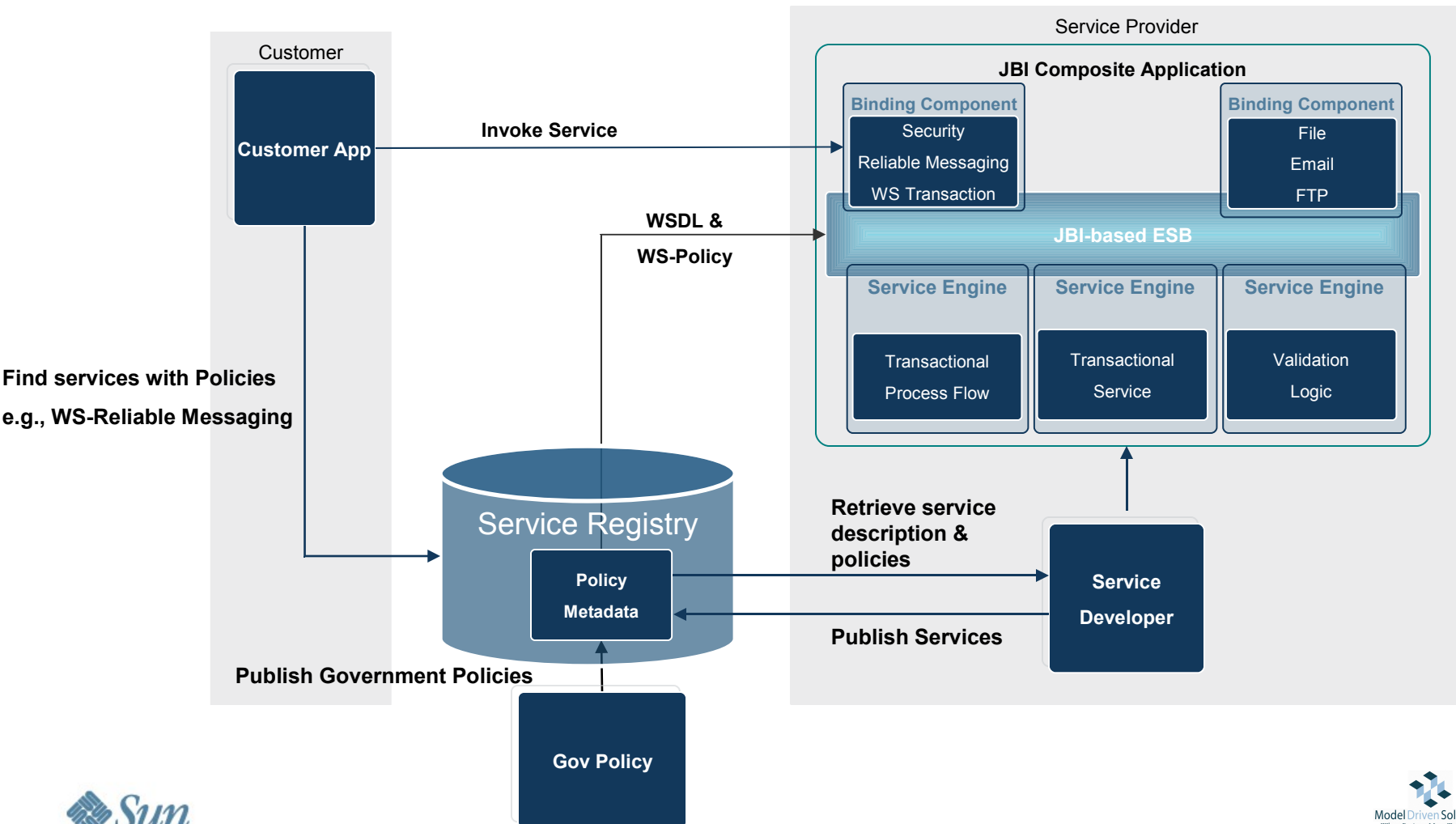
Enterprise Scenario: Using FEA

> FEA SRM Model is used for service classification.

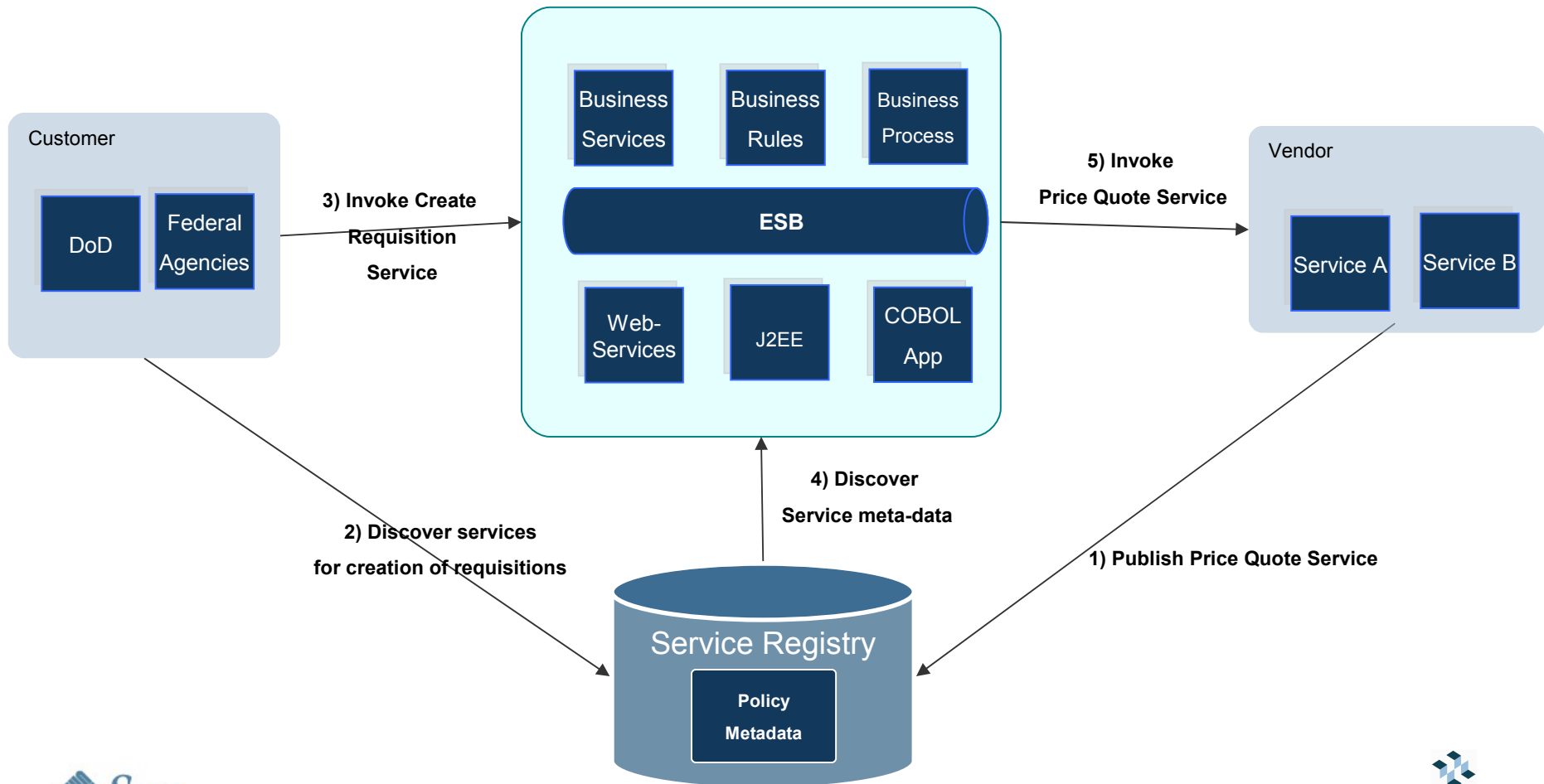
1. FEA SRM taxonomy is published in the Service Registry
1. Services are published in the Service Registry
1. Services are classified against FEA SRM Taxonomy
1. Service Consumers discover services by querying the Service Registry using FEA Taxonomy as search criteria
1. Service Consumers bind to the Services which match the query criteria
1. Service Consumers invoke the appropriate services



Government Policy Publication, Discovery, and Enforcement



Usage Scenario: The Agency as a Service Broker



Status

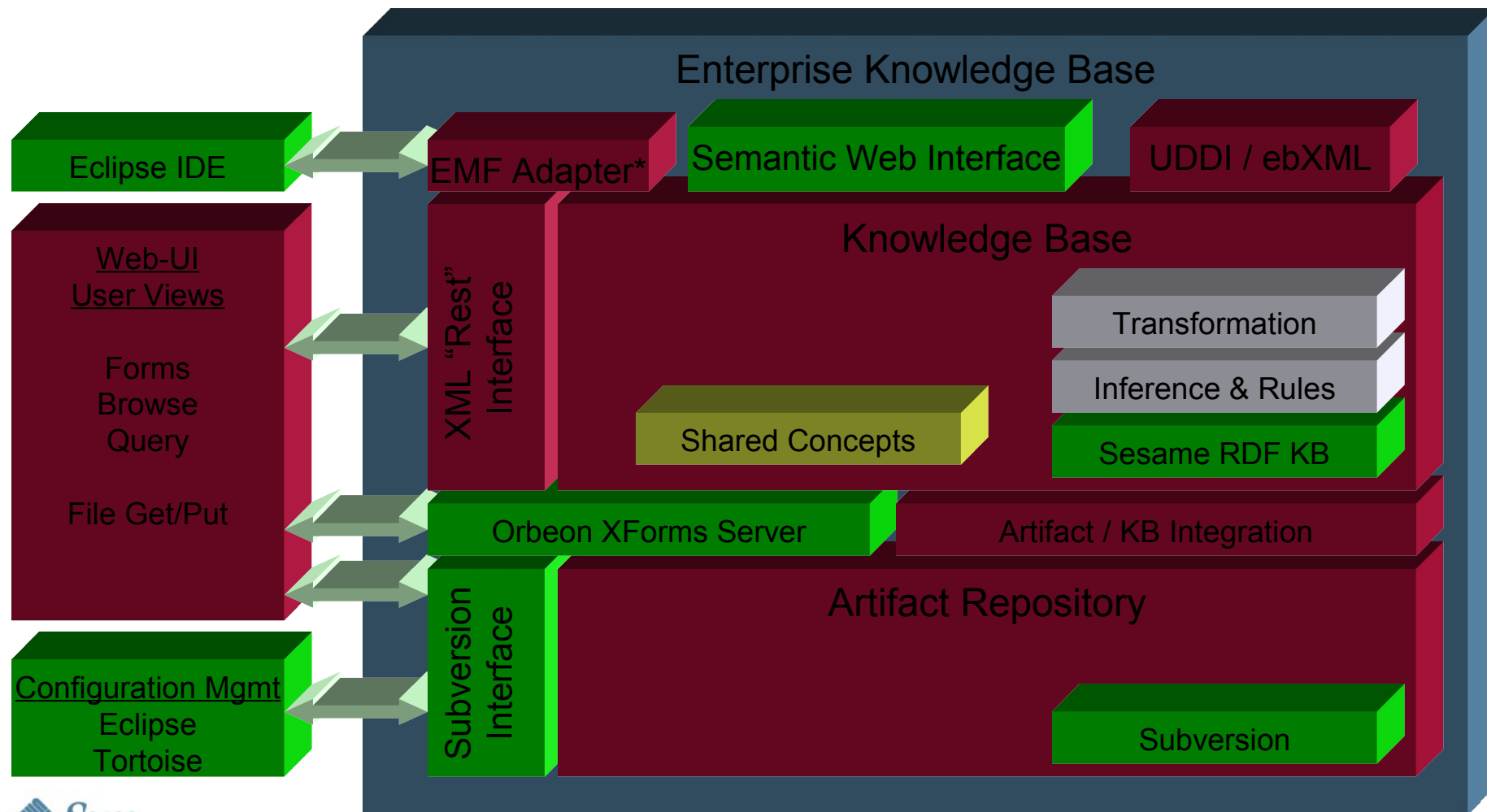
> Current

- Operational prototype deployed
- Registry populated with financial services

> Future

- What would like to do
 - Provide a semantic-based service registry / repository
 - Include adequate semantics to support validation and reasoning
 - Provide capabilities to support service assets categorization, provenance, dependencies

Enterprise Knowledge Base: High Level Architecture



Agenda

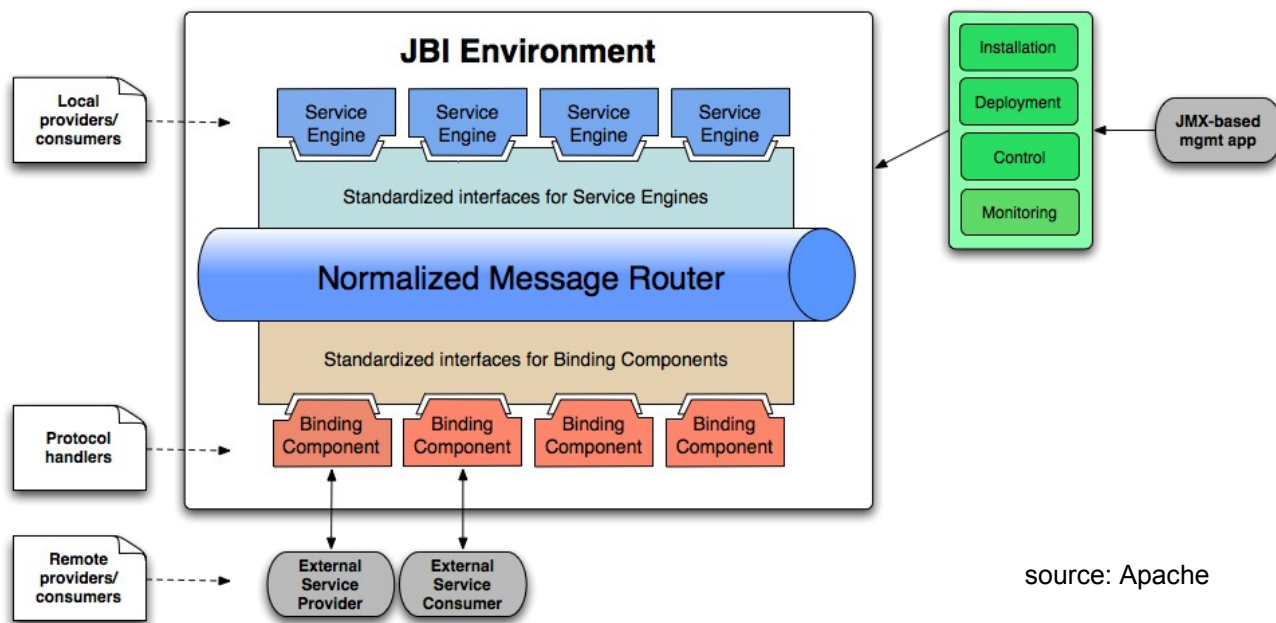
- > Context
- > Case Studies
 - The Service Registry Experience
 - **The JBI ESB Experience**
- > Advanced topics

JB1 ESB Experience

- > Focus on Java EE 5 and JBI infrastructures
 - Open source, open standard based ESB
 - Glassfish ESB
 - Apache ServiceMix (IONA FUSE ESB)
 - Products selected by OCIO
 - Refactor existing application (J2EE/WS) for JEE5/JBI
 - Open Source JBI platform comparison and assessment

- > Outcomes
 - Principles and patterns for reuse, interoperability, and agility
 - Alignment with SOA Reference Architecture
 - Elaborate the relationship of ESB and BPMS

Java Business Integration

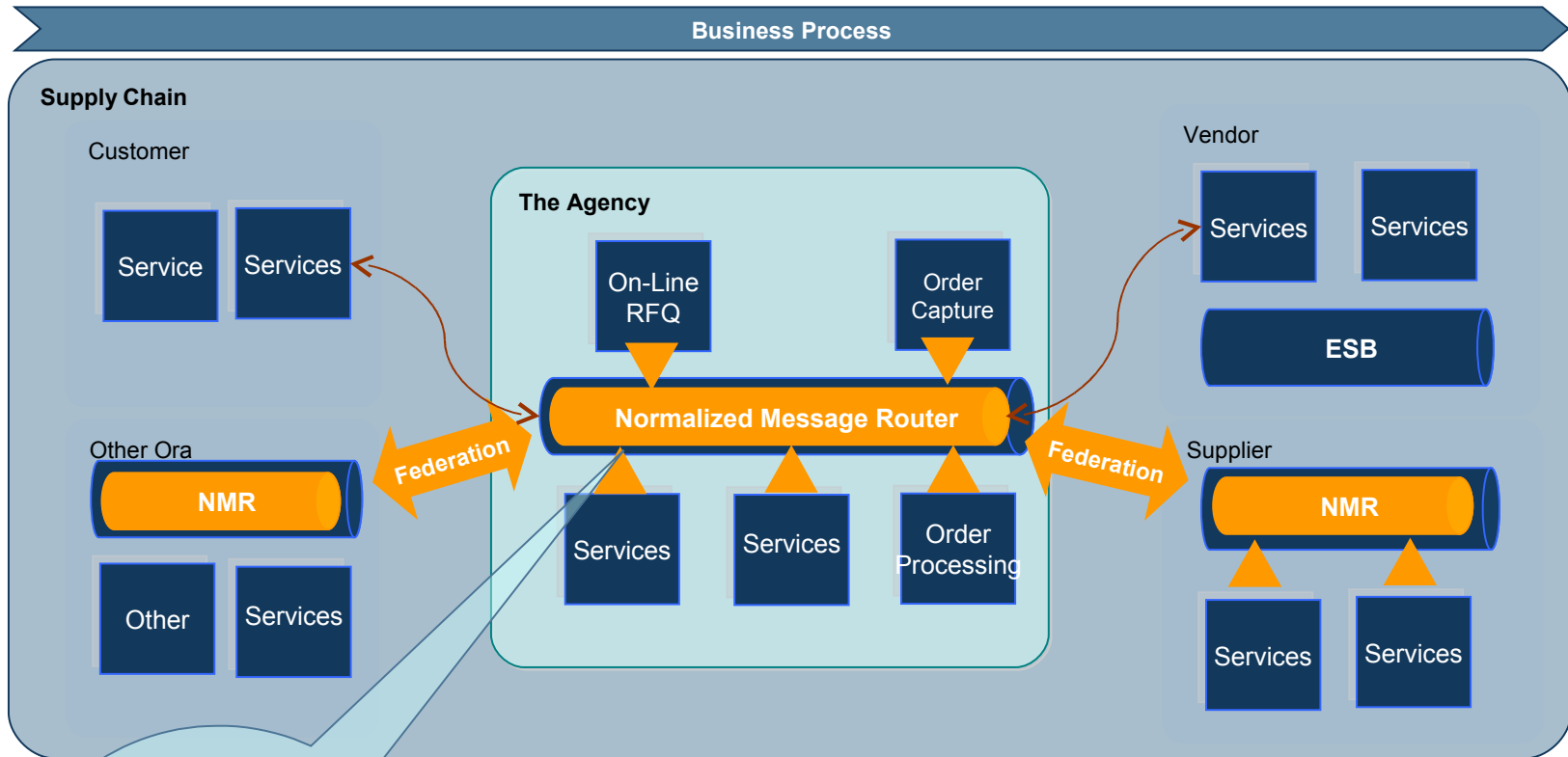


source: Apache

- > Apache ServiceMix
 - Deployed: IONA Fuse ESB Version 3.3.1.1
- > OpenESB
 - Deployed: Sun Java Application Server 9.1 Update 2



Why JBI?



"Traditional" ESB Challenges:

- JMS dependency?
- Span?
- Organizational buy in?
- Scaling?
- Reuse?



Reusability



Interoperability / Portability

Why JBI?

- > Reusability
 - Reuse existing business service implementation:
 - Service engines support standard programming models: EJB, JAX-WS, XSLT, BPEL, etc.
 - Reuse ESB component:
 - Standardized component interface: Based on WSDL 2.0
 - No vendor lock-in. No open source community lock-in either
- > Loosely Coupled Services
 - Message oriented: Normalized Message defined by JBI
 - Contract driven: Interfaces described using WSDL
- > Interoperability
 - ESB Federation via standard protocol such as HTTP
- > Agility
 - Enable continuous business process improvement
- > Maintainability
 - Metadata driven

JBI Exclusive!

JBI Exclusive!

JBI Exclusive!

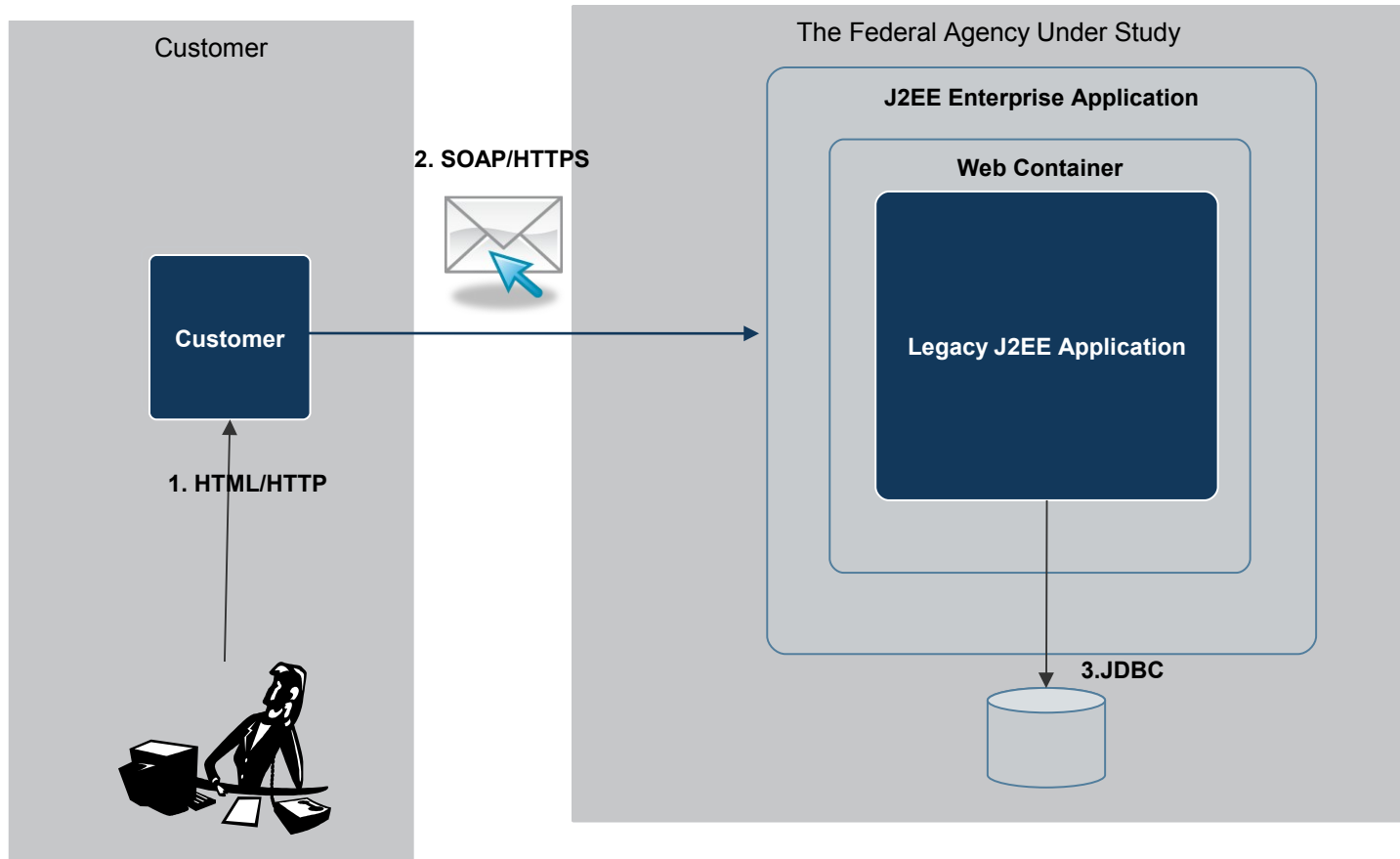
• Standard-based approach

• Commercially supported open source

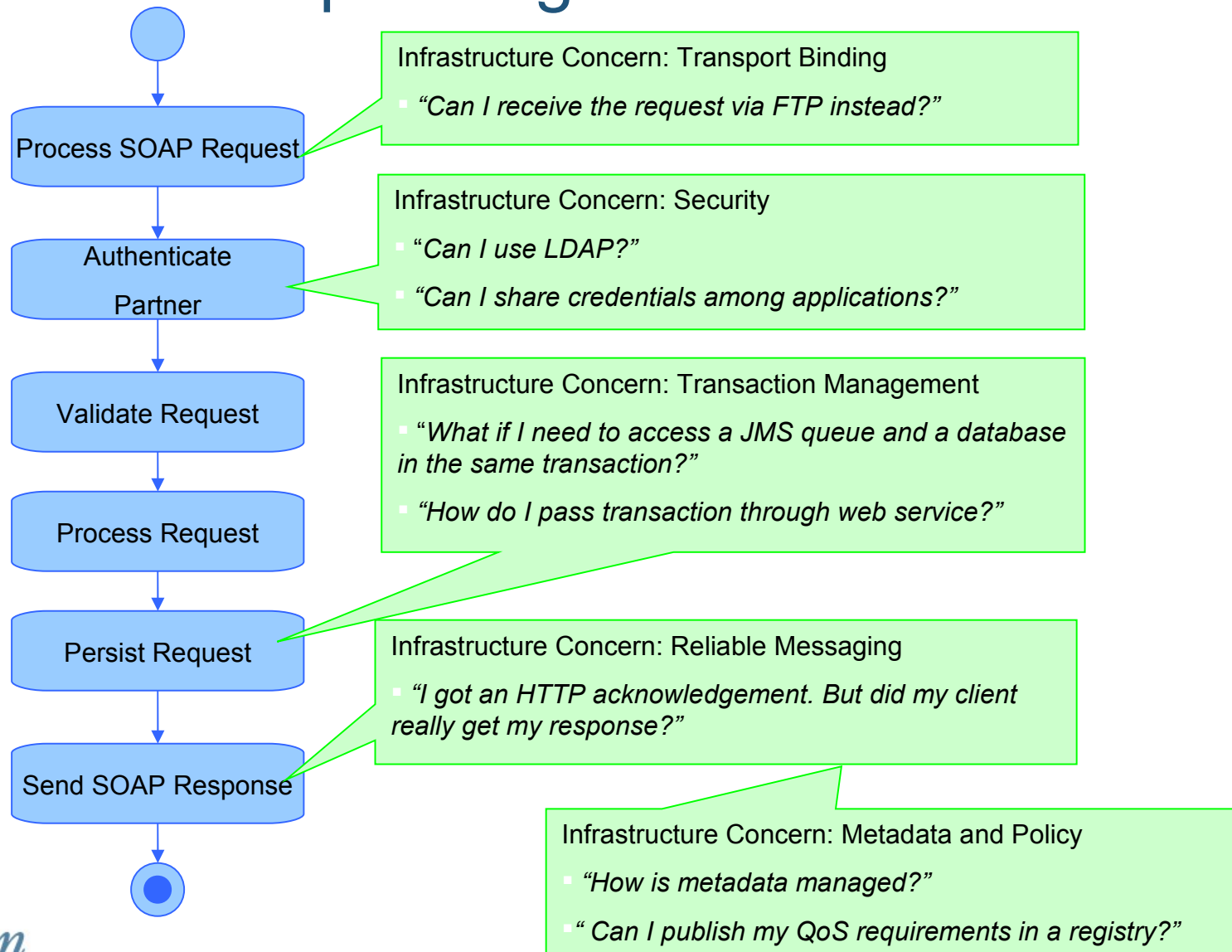
• Foster reuse of solutions and components

• **Lower TCO**
• **Higher ROI**

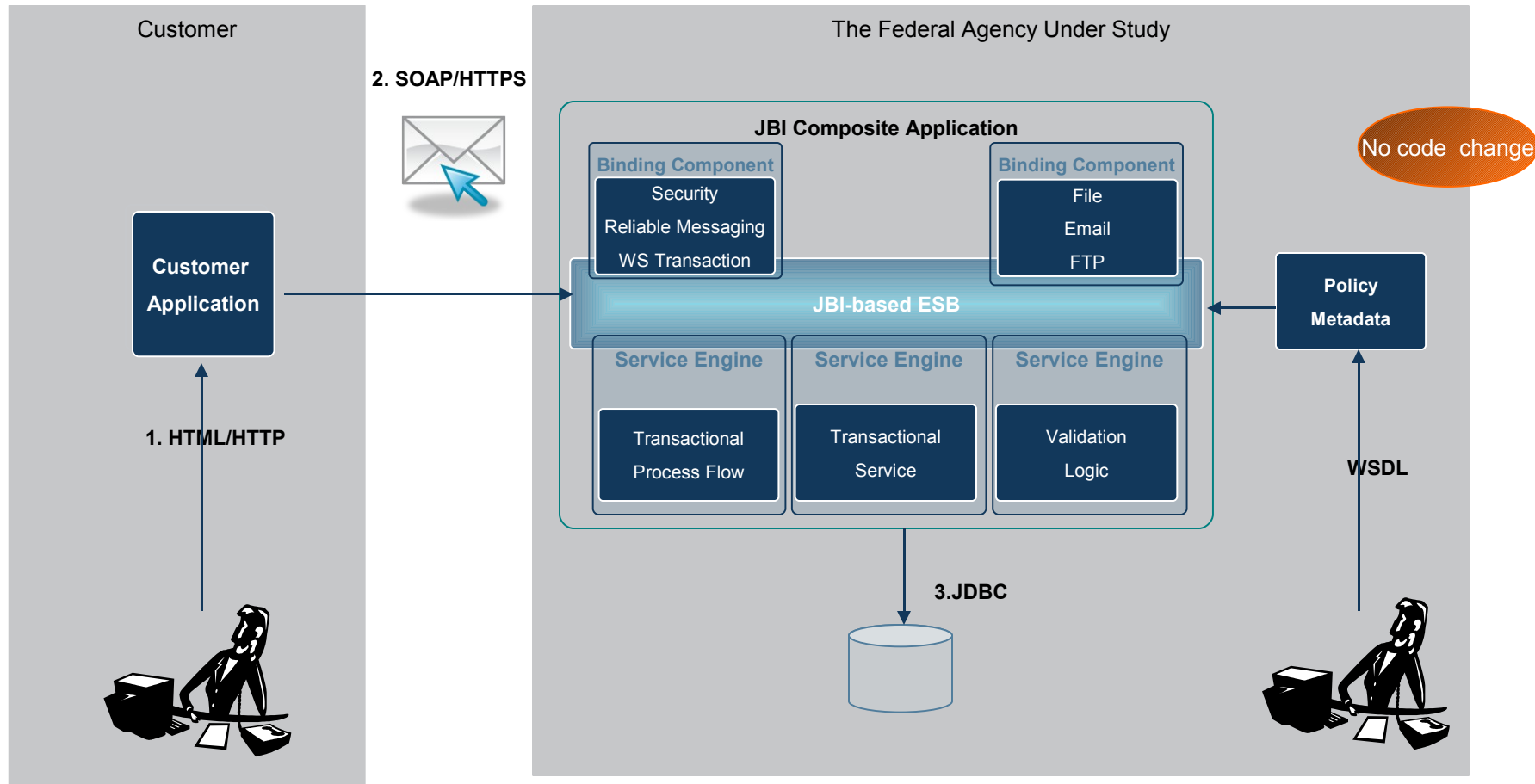
As-is J2EE Application



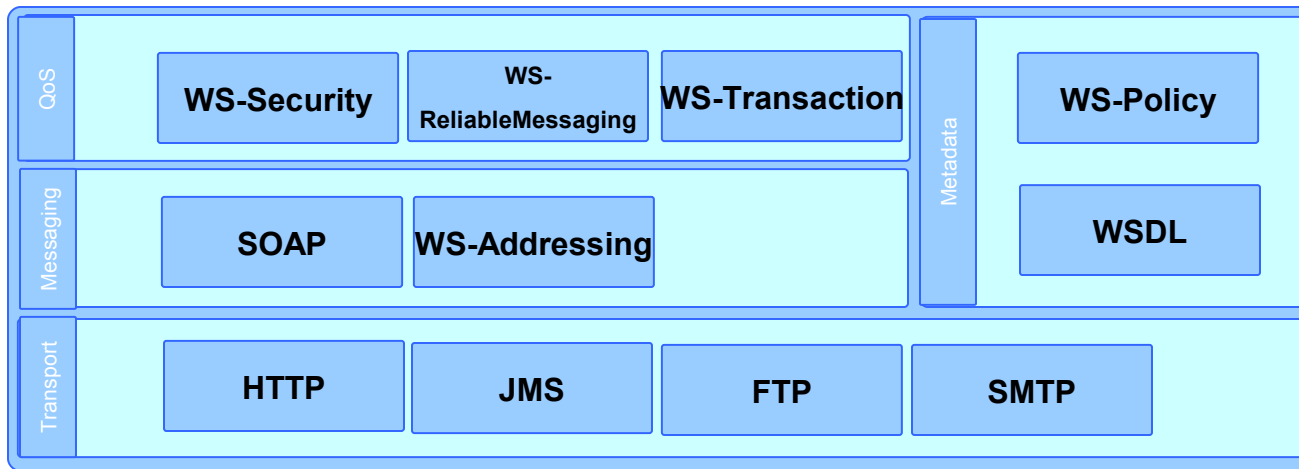
Approach: Separating Infrastructure Concerns



Refactored JBI Application



Standards Addressing Infrastructure Concerns



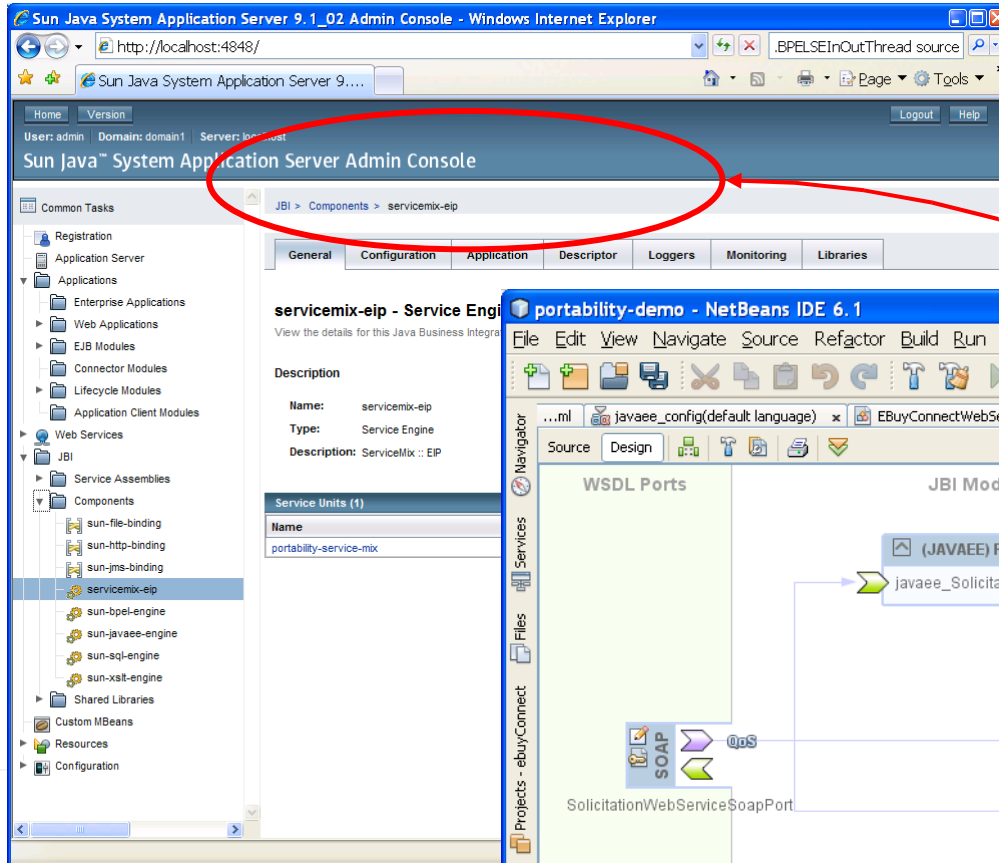
- > JBI messaging model is based on abstract WSDL
 - Transport binding is specified the WSDL
- > Web Service Specifications (WS-*) address many QoS concerns
 - WS-* support can be specified in the WSDL
 - WS-* are leveraged in the refactored technical architecture

Comparing JBI Containers

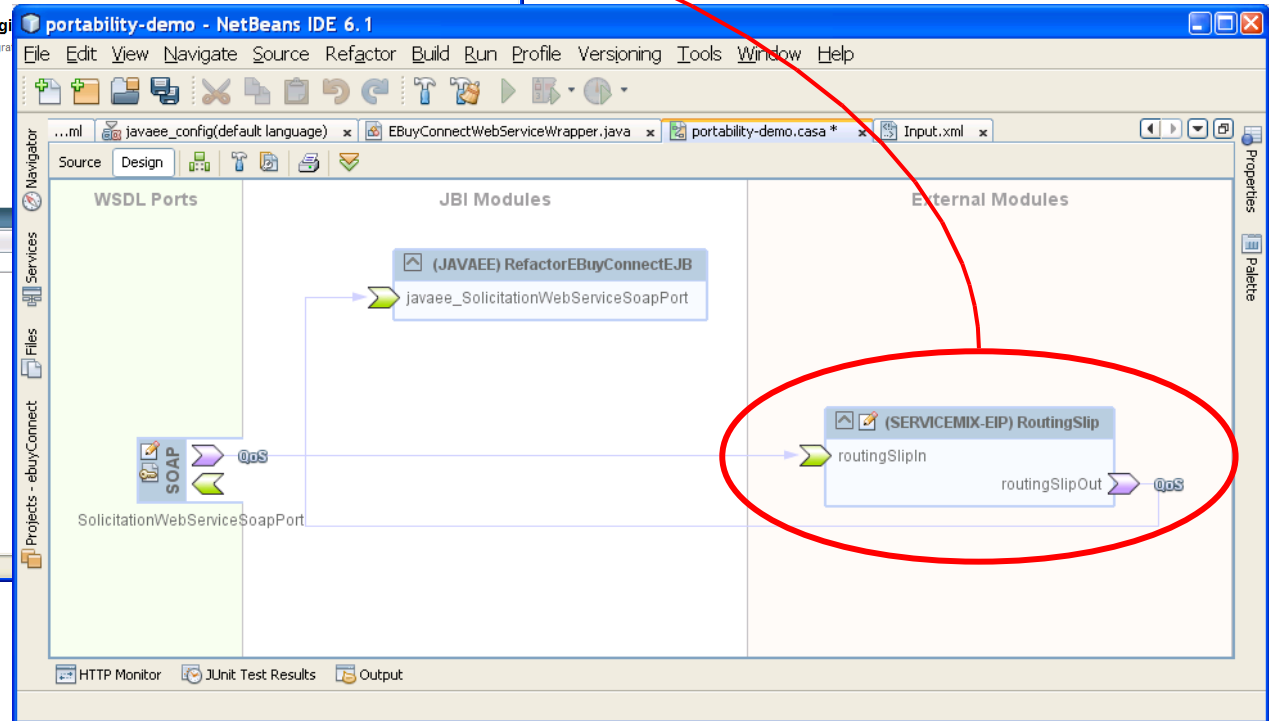
	OpenESB 2.0	ServiceMix 3.3
<i>Web Service</i>	Through Metro	Through Apache CXF
<i>JMS</i>	Sun Java MQ (Default)	Apache ActiveMQ
<i>Transaction Support</i>	Through Glassfish AS	
<i>Tooling (Easy of use)</i>	Integrated with NetBeans IDE	Fully support Maven and Spring Framework, and interceptors
<i>Policy Support</i>	Support WS-Policy	WS-Policy support is weak
<i>Enterprise Integration</i>	Enterprise Integration Patterns through Camel SE	Enterprise Integration Patterns through EIP Engine, Camel
<i>Registry</i>		
<i>Business Process</i>	BPEL SE	Apache ODE
<i>Business Rules</i>		DROOLS
<i>Clustering</i>	Through AS and Protocol Clustering	Through ActiveMQ Clustering
<i>WS-* Support</i>	WS-Security, WS-Reliable Messaging, WS-Coordination, WS-AtomicTransaction, WS-	WS-Security, WS-Notification

Component Ruse
Across ESB

Porting JBI Components



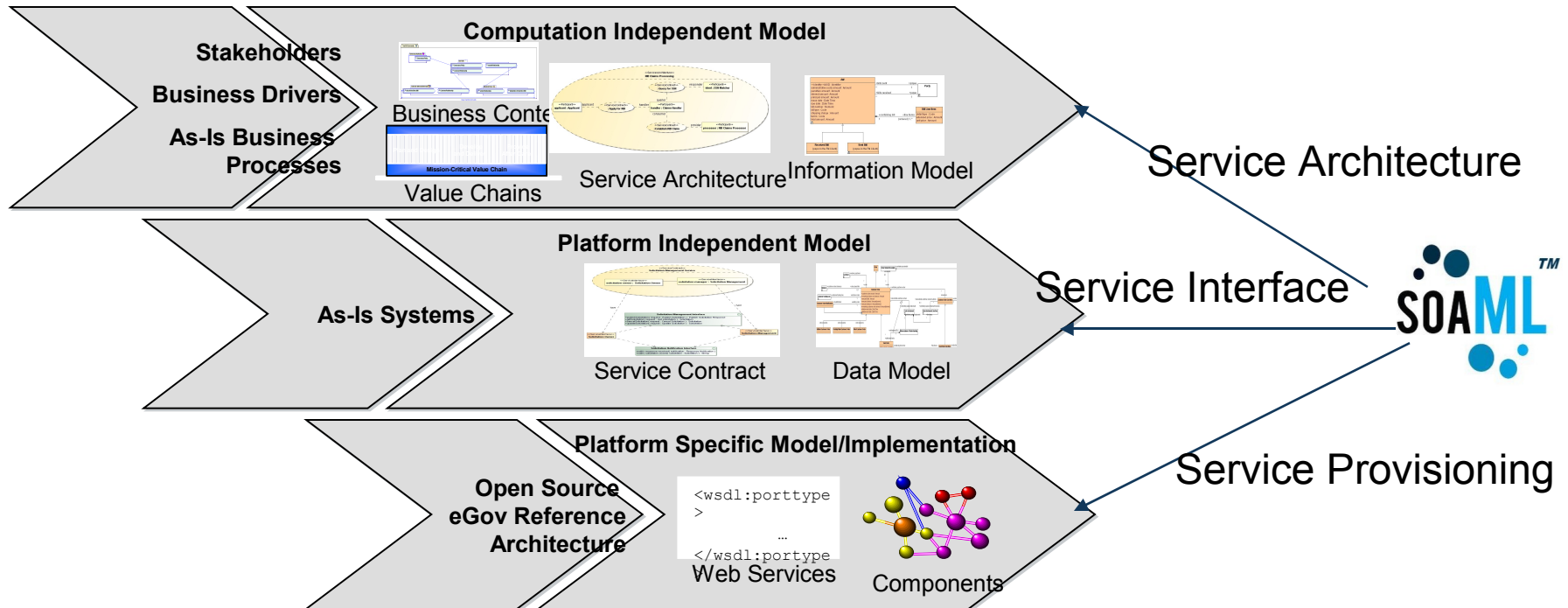
•Standardized
component interface:
Based on WSDL 2.0



Agenda

- > Context
- > Case Studies
 - Definitions
 - The Service Registry Experience
 - The JBI ESB Experience
- > **Advanced topics**

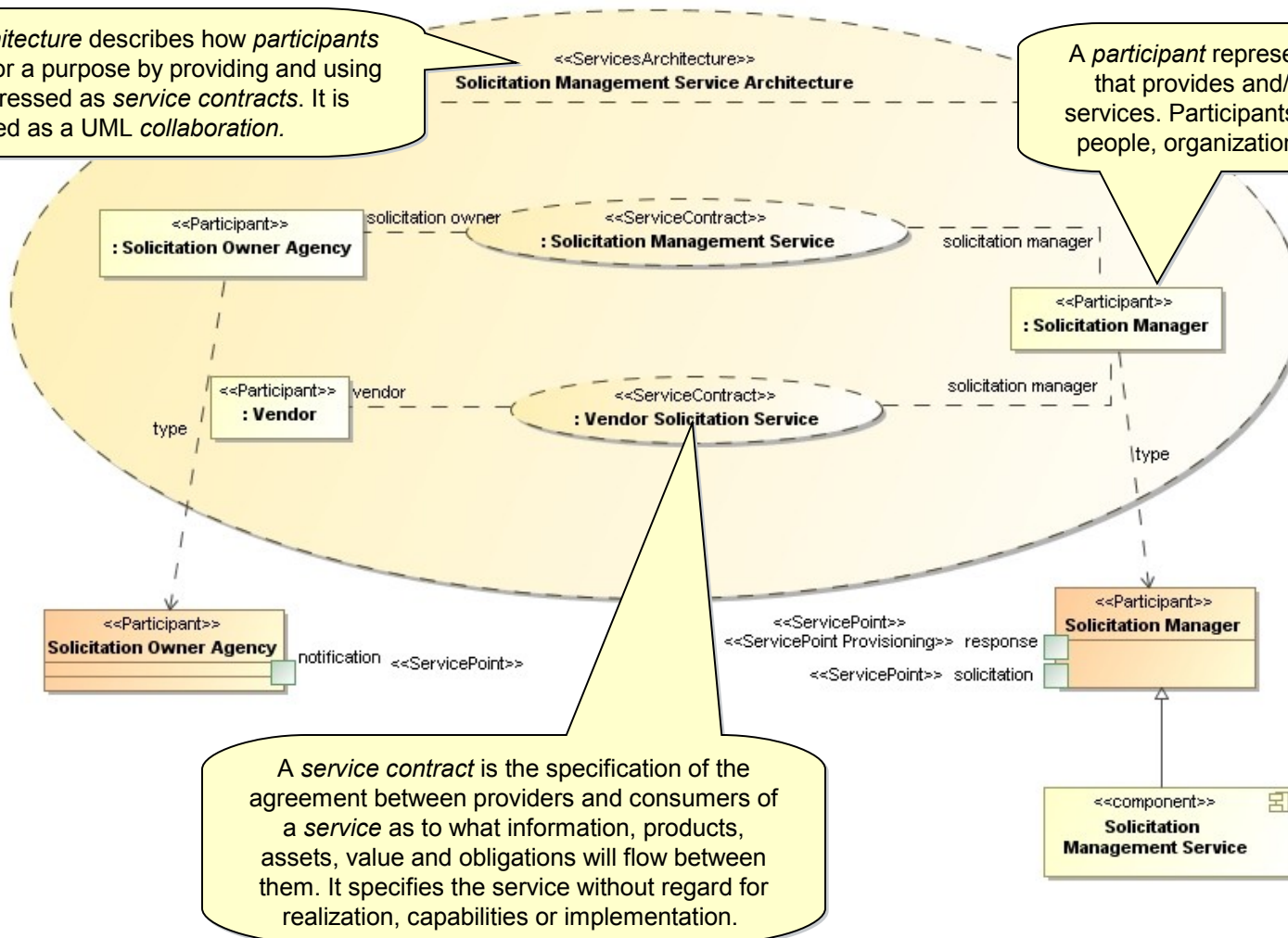
SoaML and Model Driven Architecture (MDA)



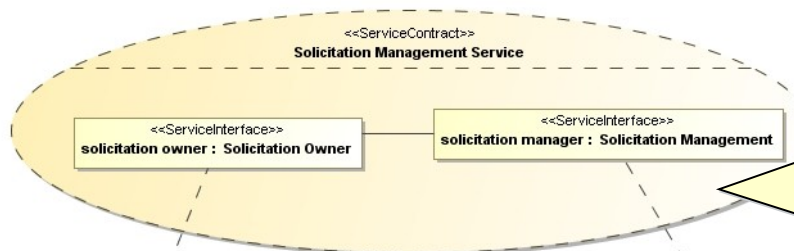
SoaML: Modeling Business Services

A *services architecture* describes how *participants* work together for a purpose by providing and using services expressed as *service contracts*. It is modeled as a UML *collaboration*.

A *participant* represents some party that provides and/or consumes services. Participants may represent people, organizations or systems.



SoaML: Platform Independent Models



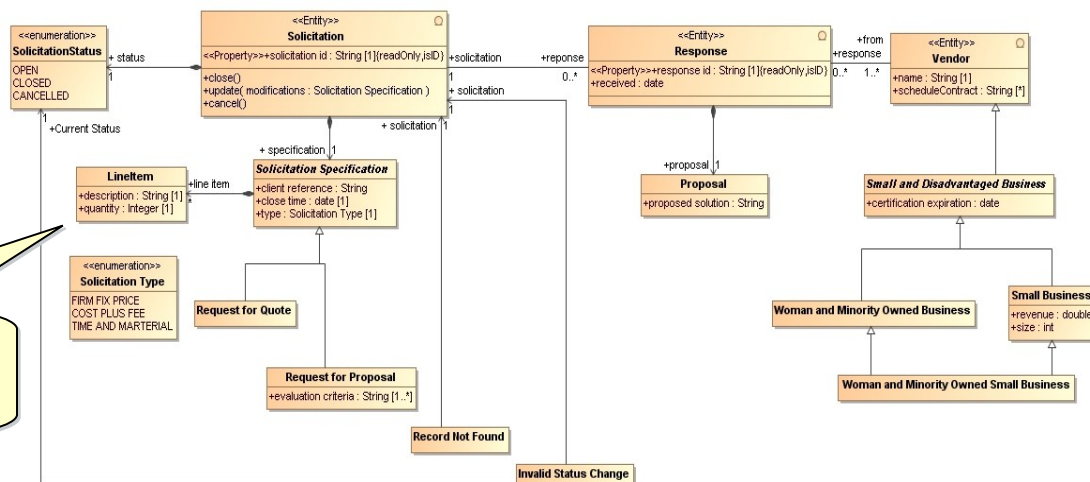
A *service contract* is the specification of the agreement between providers and consumers of a *service* as to what information, products, assets, value and obligations will flow between them. It specifies the service without regard for realization, capabilities or implementation. It is modeled as a UML *collaboration*.

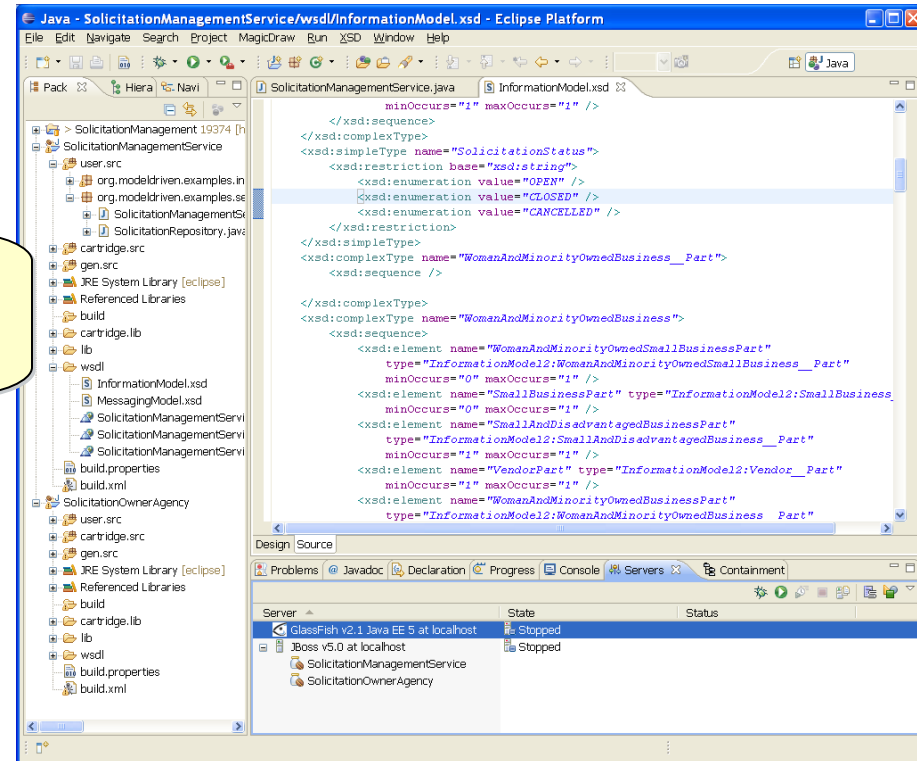


A *service interface* defines the interface and responsibilities required for a participant to play a role in a service contract. It is the means for specifying how a participant is to interact to provide or consume a service according to the contract. It is modeled as a UML *class*.

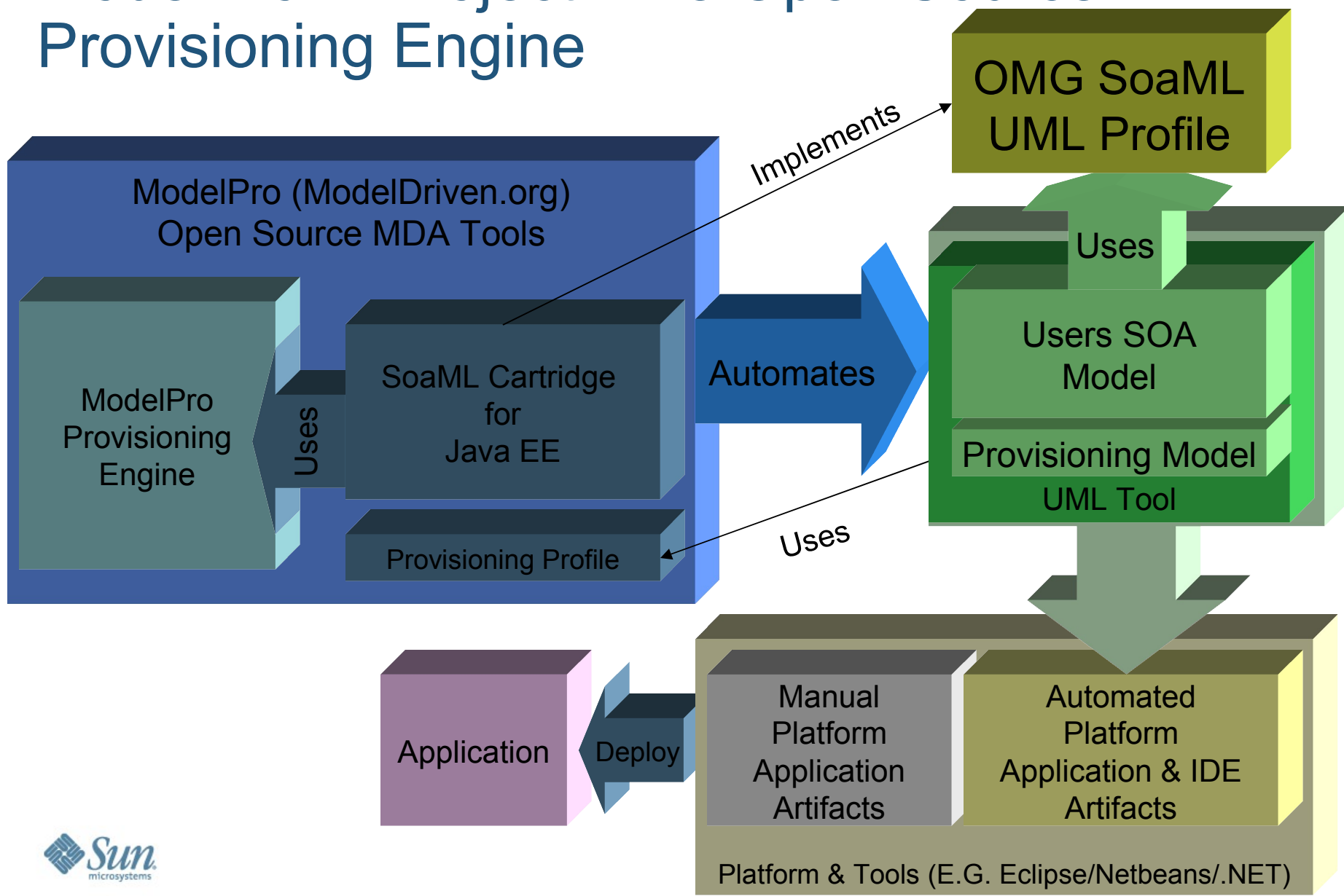


The Information model (not part of of SoaML) specifies the data classes used by the services





ModelPro™ Project: The Open Source Provisioning Engine



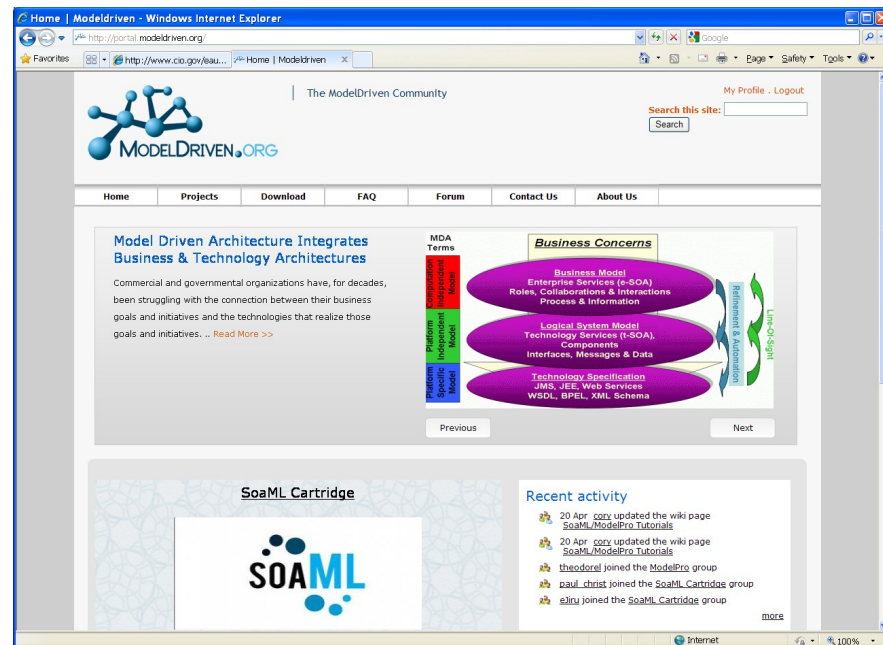
ModelDriven.org

The *Open Source* Model Driven Community

> Current Projects

- The ModelPro™ Project
- ModelPro™ Cartridge Projects
- Foundational UML Reference Implementation

> Get Involved!





JavaOneSM

Thank You

Walt Melo

Wen Zhu

{walt-m, wen-z}@modeldriven.com



Model Driven Solutions
Where Business Meets Technology



Enterprise Knowledge Base: The Future is Here

