# WebSphere Service Registry & Repository Overview

*Tendances Logicielles - 06 Décembre 2006*

Olivier Delfosse
WebSphere Integration Solution Specialist
delfosse@fr.ibm.com

ON DEMAND BUSINESS™

# 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)?

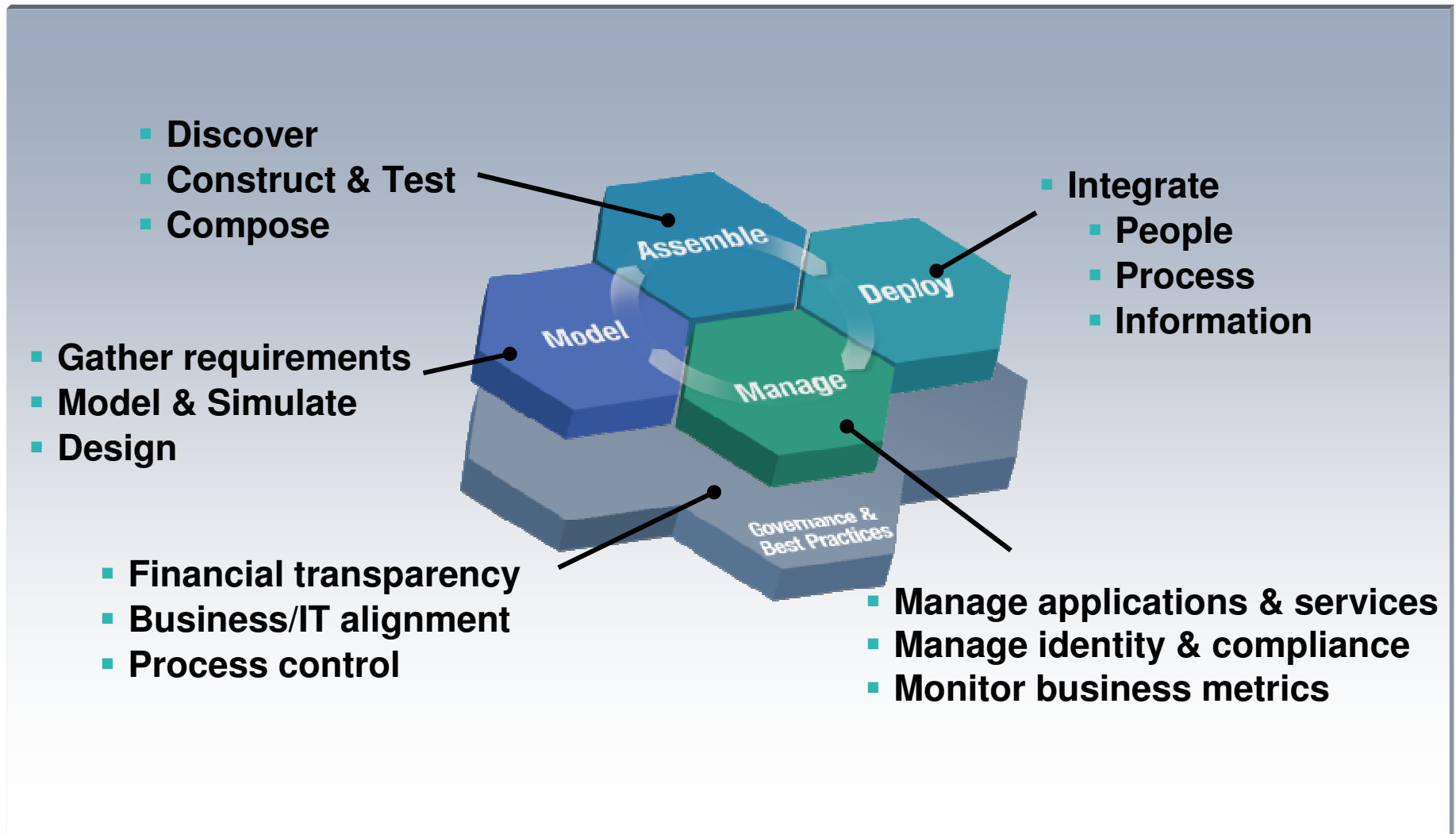An IT **architectural style** that supports service orientation

### … a composite application?

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

# How are customers thinking technically about flexible IT through SOA?
## *The SOA Lifecycle*

- **Discover**
- **Construct & Test**
- **Compose**

**Assemble**

**Deploy**

**Model**

**Manage**

**Governance & Best Practices**

- **Integrate**
  - **People**
  - **Process**
  - **Information**

- **Gather requirements**
- **Model & Simulate**
- **Design**

- **Financial transparency**
- **Business/IT alignment**
- **Process control**

- **Manage applications & services**
- **Manage identity & compliance**
- **Monitor business metrics**

# A Registry Repository answers questions customer have about governing and managing their SOA

How do I eliminate "rogue services" and ensure control of my SOA?

How do I govern services as part of my SOA?

How do I manage the services lifecycle?

How do I increase service reuse?

How do I enable enforcement of policies across all internal and external services?

How can I help my ESB execute in the right context?

How do I help services interact efficiently and dynamically with each other?

How do I optimize service interactions to be better aligned with business process?

# What is a registry … a repository?

**Registry?**
**Contains Service Metadata**
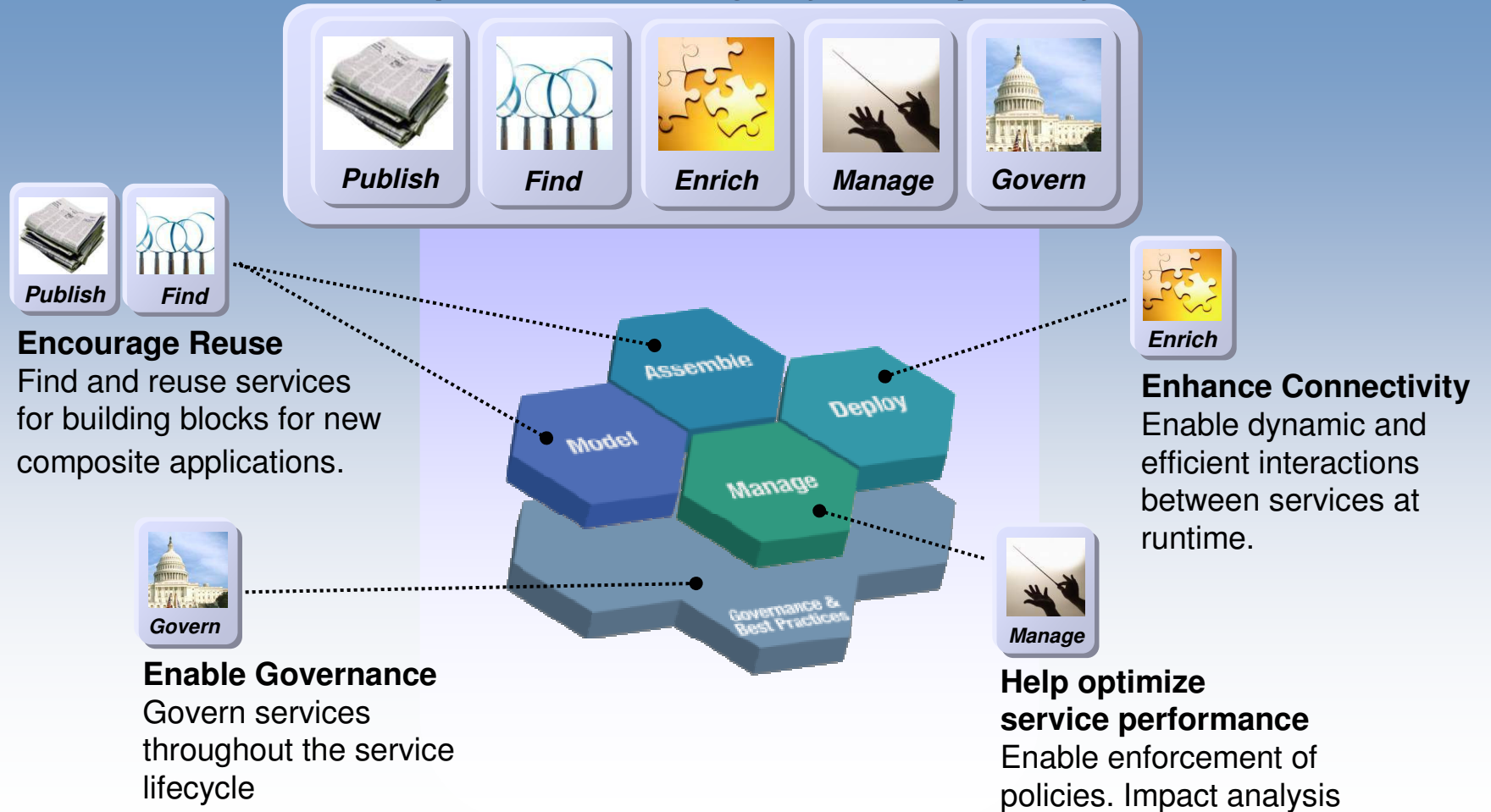
**Repository?**
**Stores Service Artifacts**

*An integrated Registry / Repository Solution*
*is needed govern and manage SOA for maximum value*

*Answers… What?... Where?... Why?... How?... of services in your SOA*

# The WebSphere Service Registry and Repository provides value throughout the SOA lifecycle
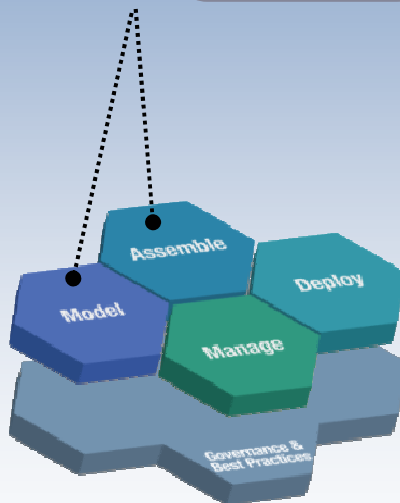
**WebSphere Service Registry and Repository**



| Publish | Find | Enrich | Manage | Govern |

**Encourage Reuse**
Find and reuse services for building blocks for new composite applications.

**Enhance Connectivity**
Enable dynamic and efficient interactions between services at runtime.

**Enable Governance**
Govern services throughout the service lifecycle

**Help optimize service performance**
Enable enforcement of policies. Impact analysis

ON DEMAND BUSINESS™

# IBM WebSphere Service Registry and Repository Capabilities
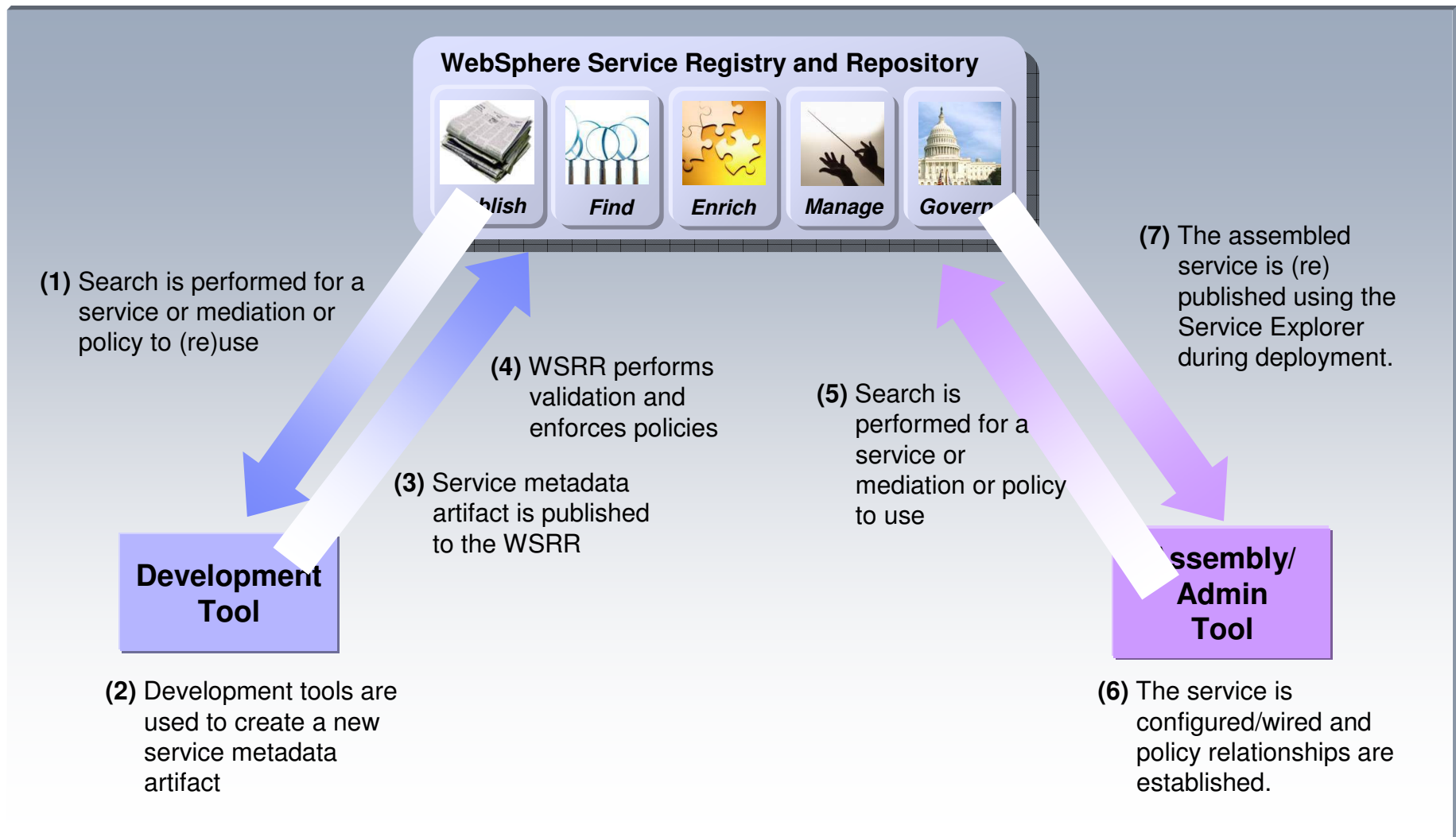
**Publish**

**Find**

## *Encourage Greater Reuse*

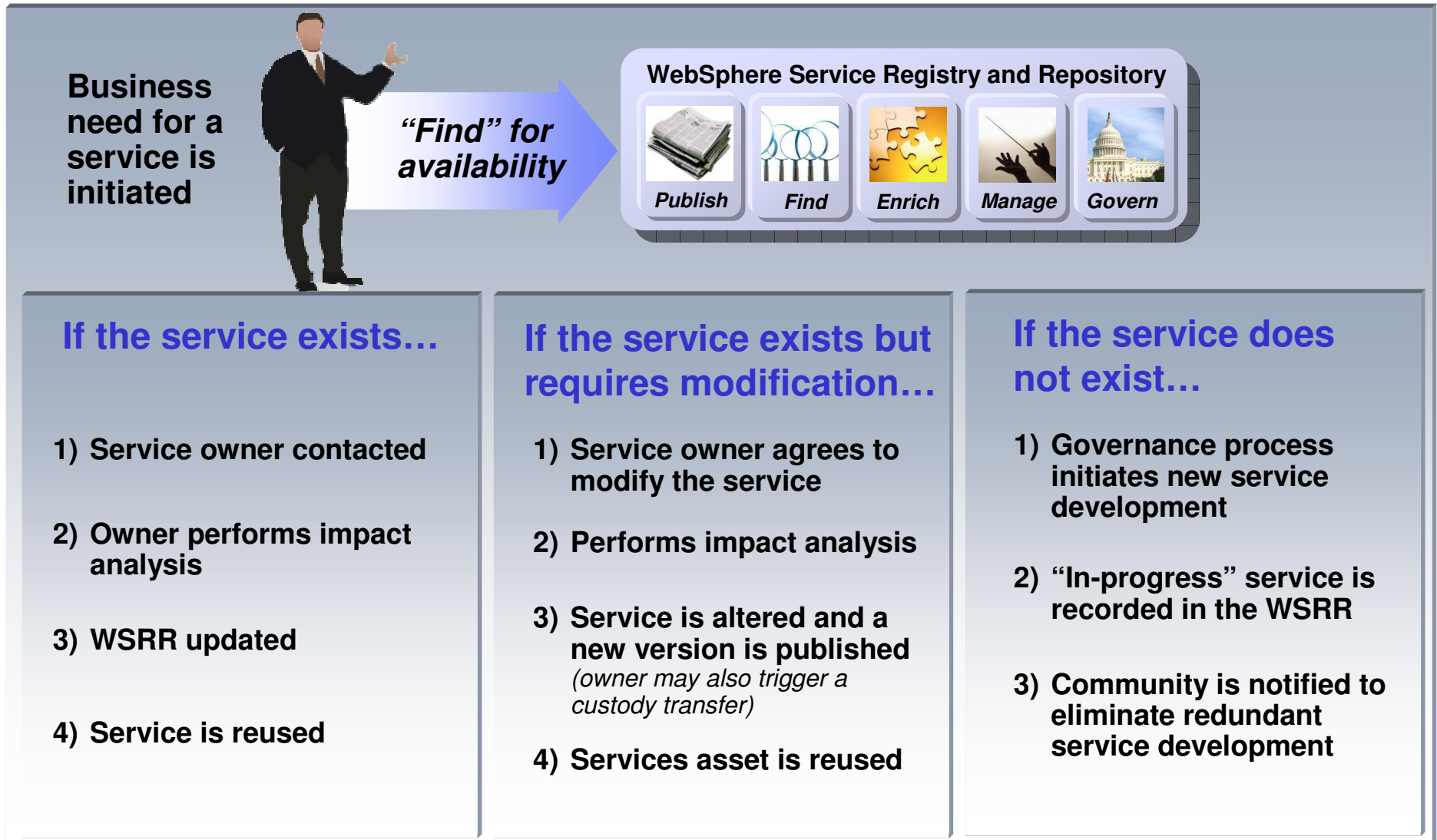Find and reuse services for building blocks for new composite applications.

## *Publish and find…*

- **Services descriptions and capabilities**

- **Service interactions, dependencies and redundancies**

- **Service lifecycle stage**
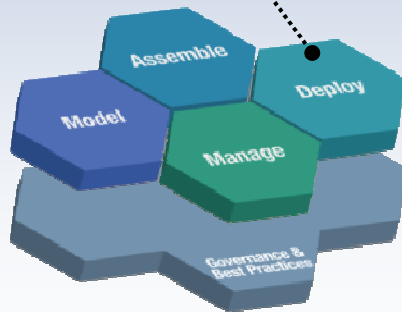
- **Policies for service usage**

Assemble

Deploy

Model

Manage

Governance & Best Practices

ON DEMAND BUSINESS™

# How it works: Publish and Find Interactions

**WebSphere Service Registry and Repository**



| blish | Find | Enrich | Manage | Govern |

**(1)** Search is performed for a service or mediation or policy to (re)use

**(4)** WSRR performs validation and enforces policies

**(3)** Service metadata artifact is published to the WSRR

**(5)** Search is performed for a service or mediation or policy to use

**(7)** The assembled service is (re) published using the Service Explorer during deployment.

**Development Tool**

**Assembly/ Admin Tool**

**(2)** Development tools are used to create a new service metadata artifact

**(6)** The service is configured/wired and policy relationships are established.

# How it's used: Greater reuse of services

**Business need for a service is initiated**

*"Find" for availability*

**WebSphere Service Registry and Repository**

| | | | | |
|---|---|---|---|---|
| *Publish* | *Find* | *Enrich* | *Manage* | *Govern* |

### If the service exists…

1) Service owner contacted

2) Owner performs impact analysis

3) WSRR updated

4) Service is reused

### If the service exists but requires modification…

1) Service owner agrees to modify the service

2) Performs impact analysis

3) Service is altered and a new version is published
*(owner may also trigger a custody transfer)*

4) Services asset is reused

### If the service does not exist…

1) Governance process initiates new service development

2) "In-progress" service is recorded in the WSRR

3) Community is notified to eliminate redundant service development

# IBM WebSphere Service Registry and Repository Capabilities

**Enrich**

## *Enhance Connectivity*

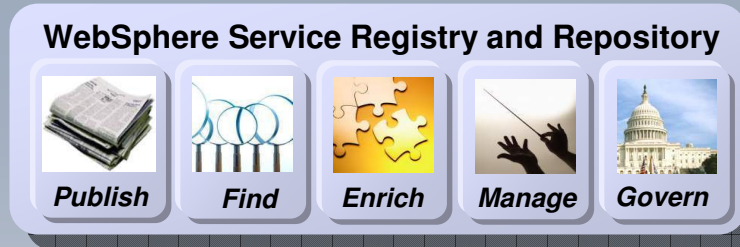Enable dynamic and efficient interactions among services at runtime.

- **Manage dynamic and efficient access to services information by runtimes**

  - **Service endpoint selection**

  - **Service availability management**

  - **Policy enforcement**

- **Identify users of metadata**

- **Notify users of changes**

- **Securely transmit service information**

**ON** DEMAND BUSINESS™

# How it works: Runtime selection and invocation interactions

## Dynamic selection

**WebSphere Service Registry and Repository**

| Publish | Find | Enrich | Manage | Govern |

**(3)** Retrieve <u>requestor</u> information

**(4)** Retrieves candidate <u>providers</u> information

Message → ESB → Message → **ESB Mediation** → Message → **Service**

**(1)** A Message is received

**(2)** Invokes a selection mediation.

**(5)** Executes matching algorithm to identify the provider service for requestor service

**(6)** Message is transformed and routed to the selected endpoint.

# How it's used: Enhancing Connectivity

**ESB**

**+**

**WebSphere Service Registry and Repository**

| Publish | Find | Enrich | Manage | Govern |

## Dynamic Endpoint Selection

1) ESB mediation is invoked

2) Mediation queries WSRR for information about the requestor and candidate provider

3) Mediation matches requestor with best candidate provider

4) Message is routed

## Availability Management

1) Selected provider fails to respond due to failure

2) Mediation queries WSRR to find other candidate providers

3) Mediation matches requestor with best candidate provider

4) Message is routed

## Policy Enforcement

1) Mediation queries WSRR for information about the requestor and candidate provider

2) Mediation retrieves policy information from registry

3) Requestor and provider are matched based on these policies

4) Message is routed

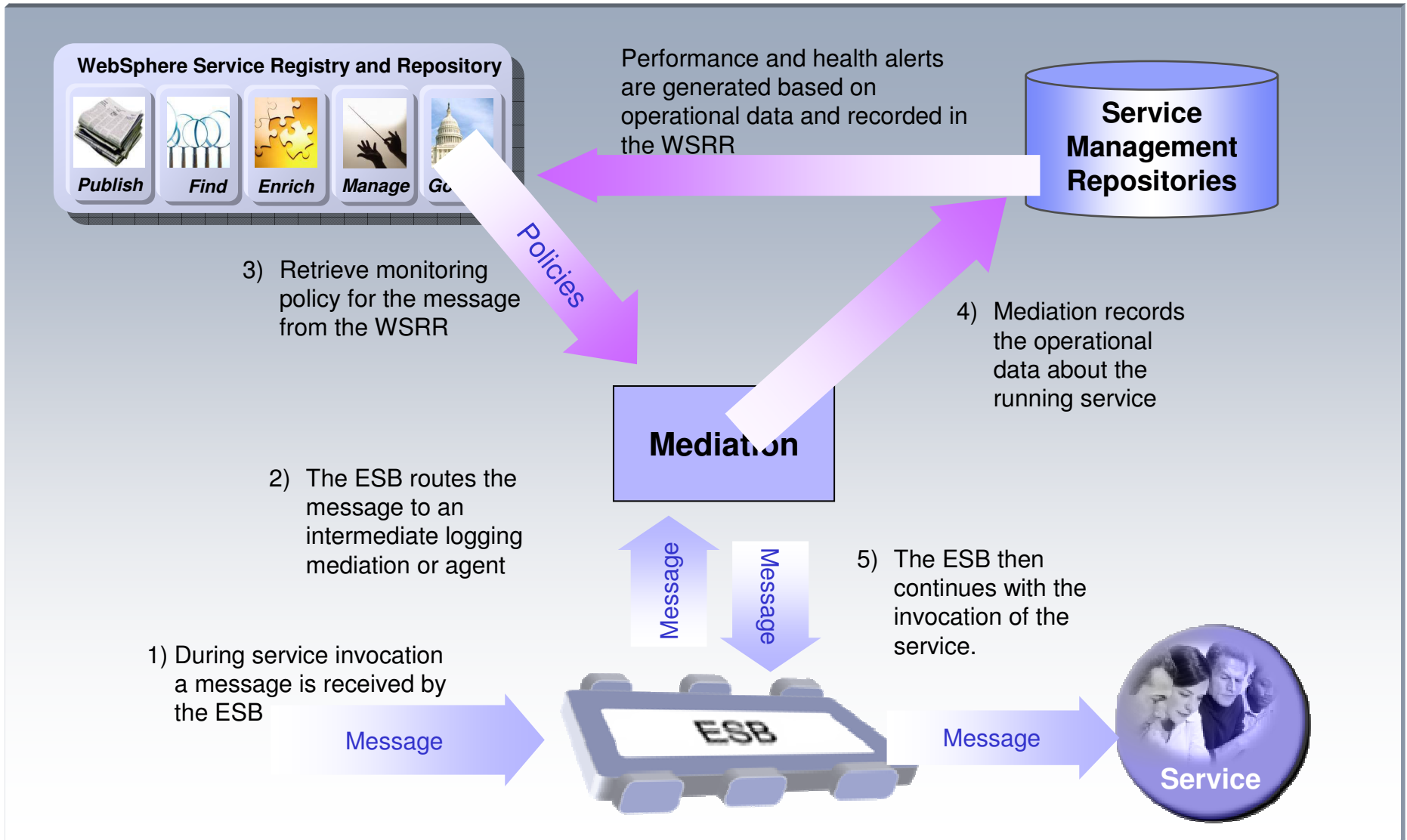# WebSphere Service Registry & Repository Runtime Integration

## WebSphere Service Registry and Repository

| Publish | Find | Enrich | Manage | Govern |

| Enterprise Service Bus | Message Broker | Process Server | CICS | Generic Client (.Net or other) |
| --- | --- | --- | --- | --- |
| Mediations based on WSRR Lookup | MB Routing Nodes based on WSRR lookup | Mediations based on WSRR Lookup | Any CICS Web services provider program publish and read capability | Web services client can publish and search |
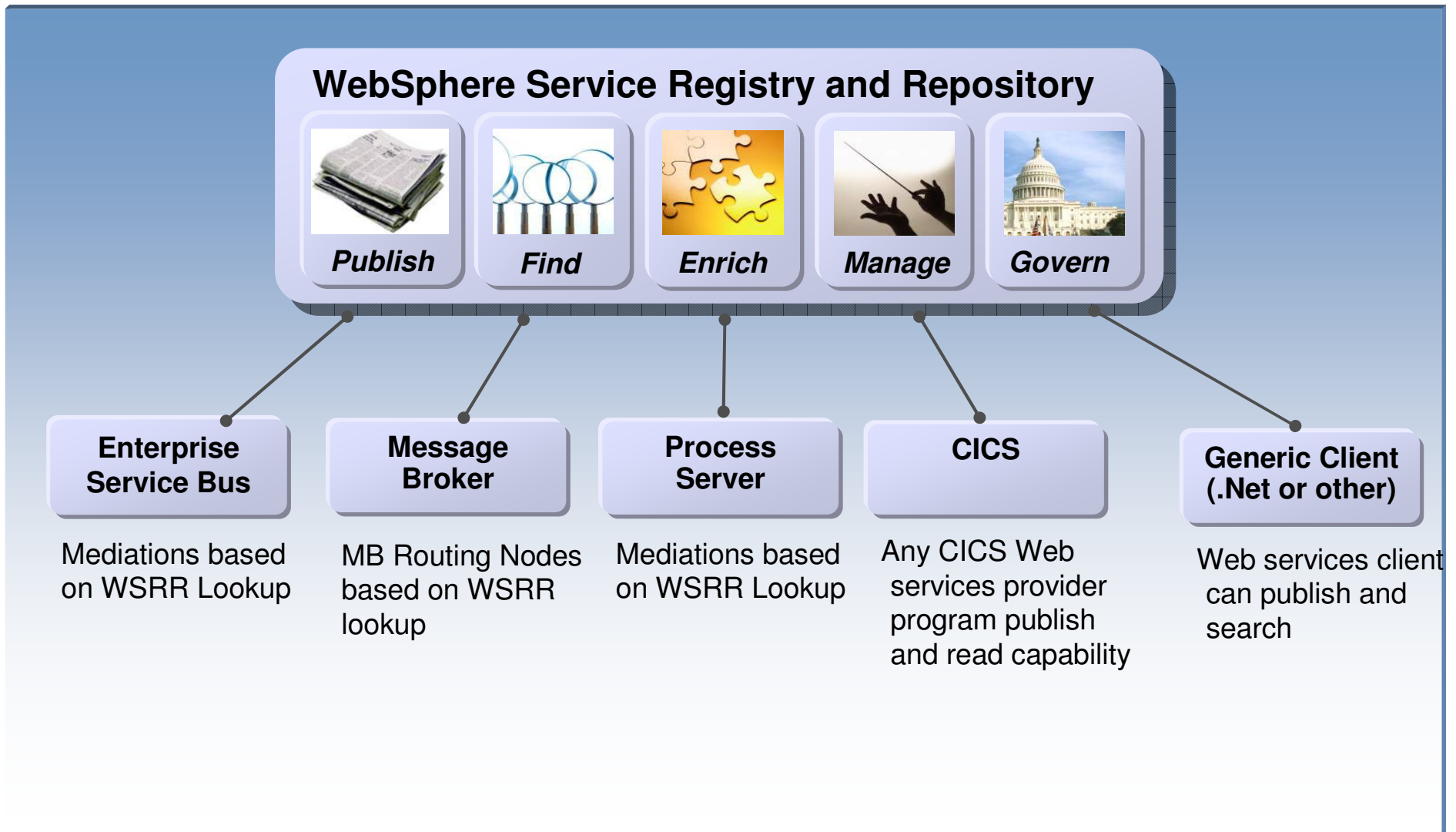
# IBM WebSphere Service Registry and Repository
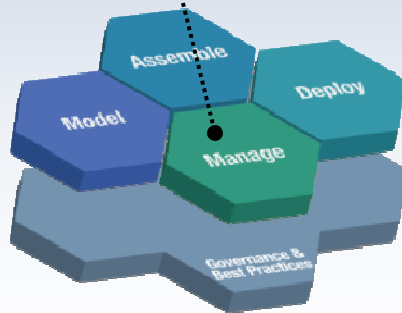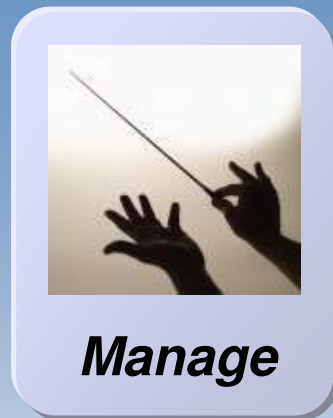
**Manage**

## Help optimize service usage and performance

- **Manage service interactions, dependencies, relationships and redundancies**

- **Classify services into meaningful groupings based on business objectives**

- **Manage policies for service usage and governance**

- **Manage change and versioning of services**

- **Analyze services usage, history and business impact**
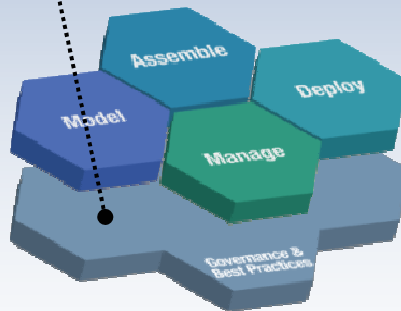
- **Promote and encourage optimal services usage**

Assemble
Model
Deploy
Manage
Governance &
Best Practices

# How it works (How it's used): Operational Monitoring Interactions

**WebSphere Service Registry and Repository**

*Publish*  *Find*  *Enrich*  *Manage*  *Go...*
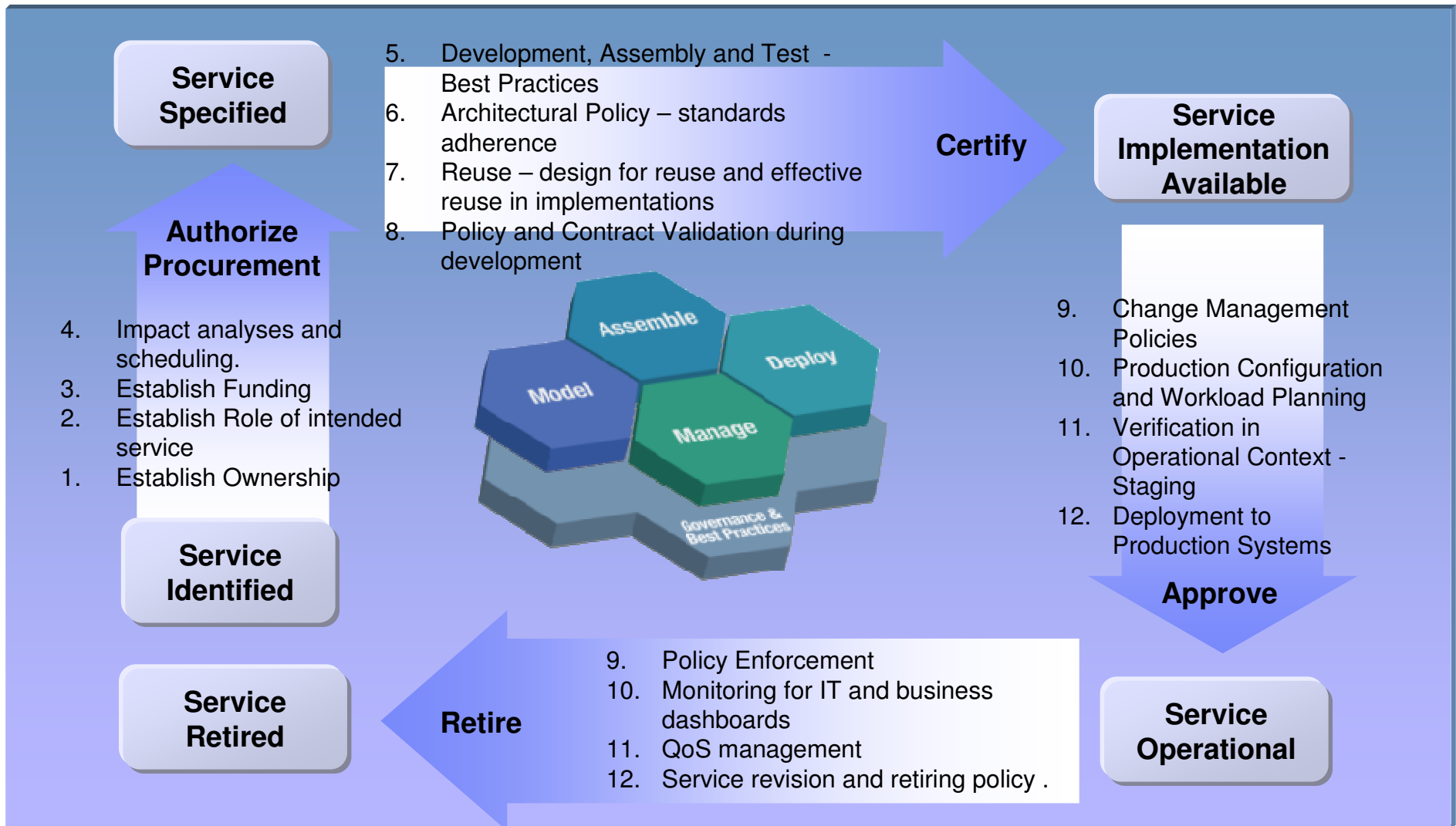
Performance and health alerts are generated based on operational data and recorded in the WSRR

**Service Management Repositories**

*Policies*

3) Retrieve monitoring policy for the message from the WSRR

4) Mediation records the operational data about the running service

**Mediation**

2) The ESB routes the message to an intermediate logging mediation or agent

*Message*  *Message*

5) The ESB then continues with the invocation of the service.

1) During service invocation a message is received by the ESB

Message

ESB

Message

**Service**

# IBM WebSphere Service Registry and Repository

**Govern**

## *Enable Governance*

Govern services throughout the service lifecycle

- **Infrastructure to help organize and discover services assets, govern access and monitor service vitality**

- **Classification of services by lifecycle phase**

- **Policies for publishing, using and retiring services**

- **Roles based access**

# How it works: Enabling governance and management of the services lifecycle

**Service Specified**

5. Development, Assembly and Test - Best Practices
6. Architectural Policy – standards adherence
7. Reuse – design for reuse and effective reuse in implementations
8. Policy and Contract Validation during development

**Certify**

**Service Implementation Available**

**Authorize Procurement**

4. Impact analyses and scheduling.
3. Establish Funding
2. Establish Role of intended service
1. Establish Ownership

9. Change Management Policies
10. Production Configuration and Workload Planning
11. Verification in Operational Context - Staging
12. Deployment to Production Systems

**Service Identified**

**Approve**

*Assemble*

*Model*

*Deploy*

*Manage*

*Governance & Best Practices*

**Service Retired**

**Retire**

9. Policy Enforcement
10. Monitoring for IT and business dashboards
11. QoS management
12. Service revision and retiring policy .

**Service Operational**

# WebSphere Service Registry & Repository Architecture
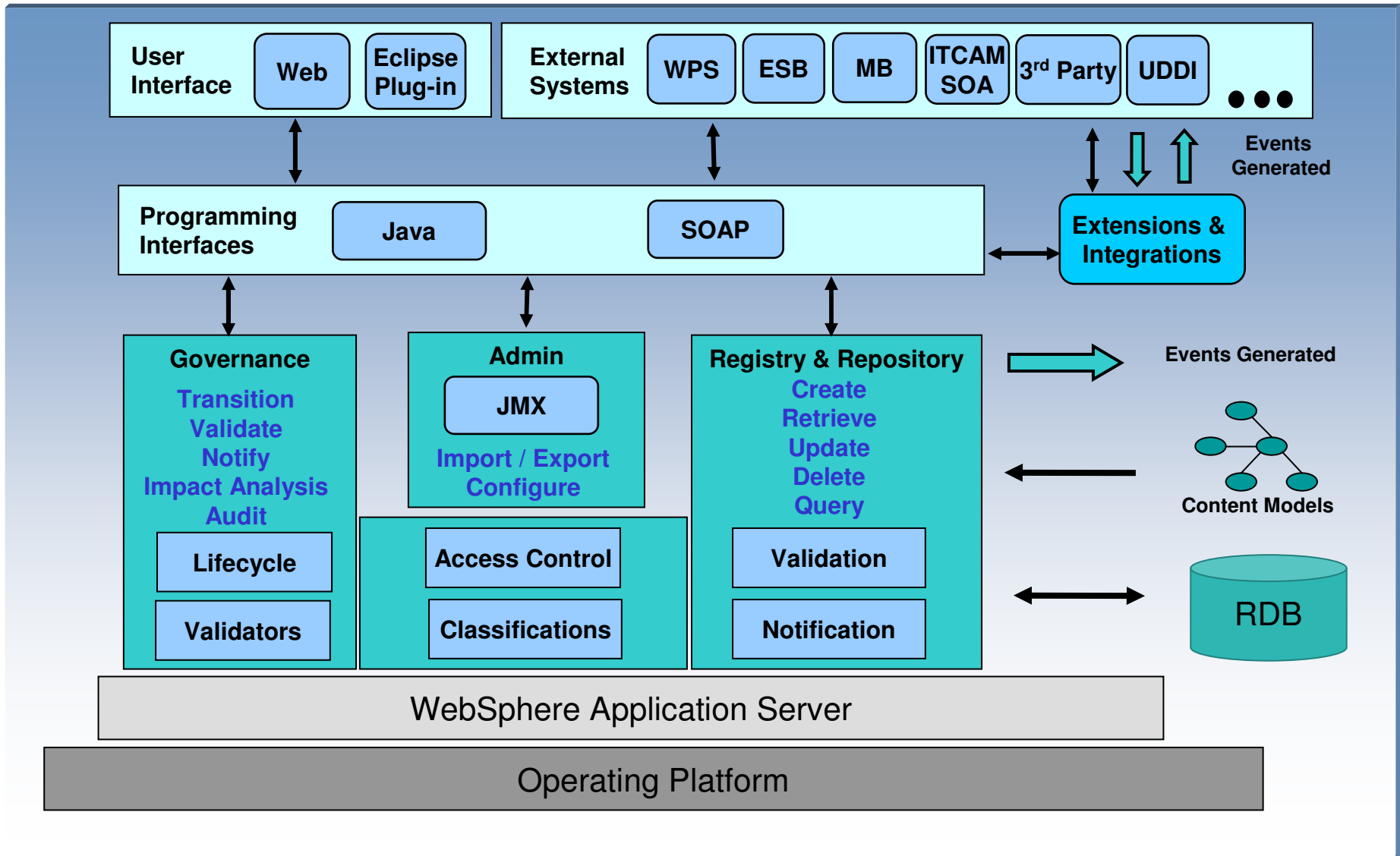
# Browser based console

- User role based browser perspectives

- Standards based service metadata documents support

- "Shredding" documents into meaningful and optimized organization

- Query
  - Canned and user wizard based search

- Classification
  - Helps Manage service metadata (more on this later under the Manage capability)
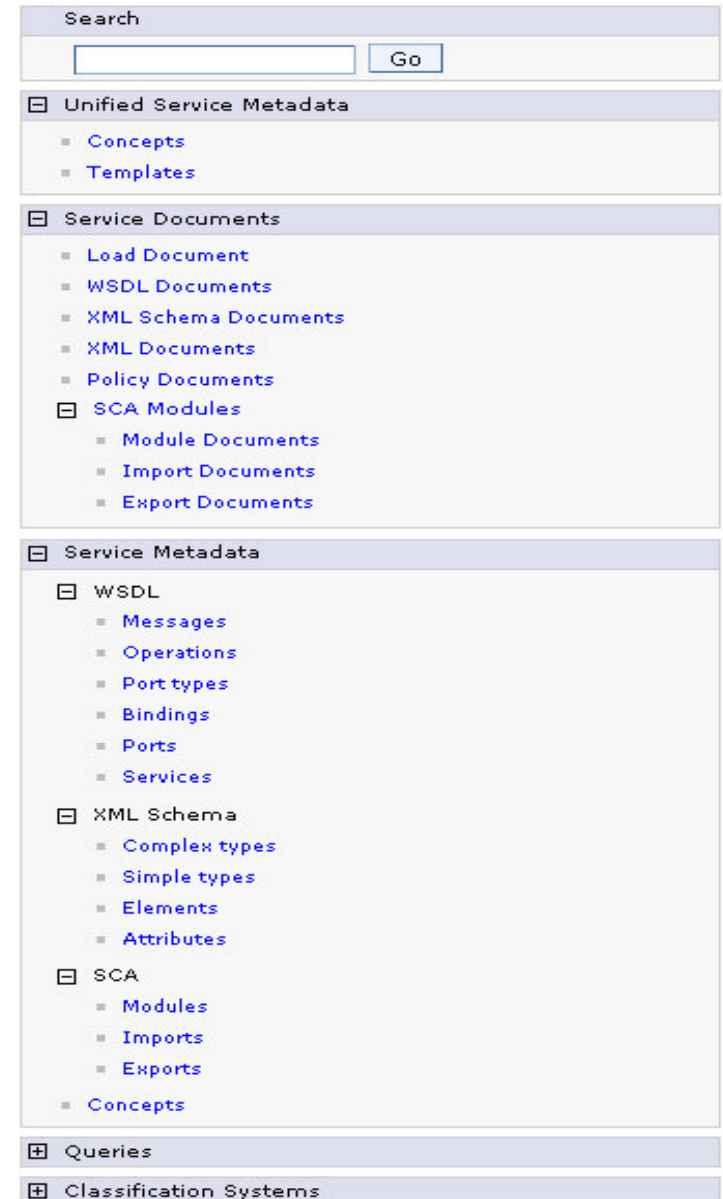
# WSRR Eclipse Plug-in

- **Generalized Eclipse plug-in**
  - Read and/or write access to WSRR
  - Selectively browse for WSRR meta-data
  - Import meta-data into a selected project
    - "Deep" import available to include nested resources
  - Publish meta-data to WSRR
  - Uses published Java interface

- **Available for any Eclipse based component**
  - Rational Application Developer
  - WebSphere Integration Developer
  - Portlet development environment
  - Other eclipse environment
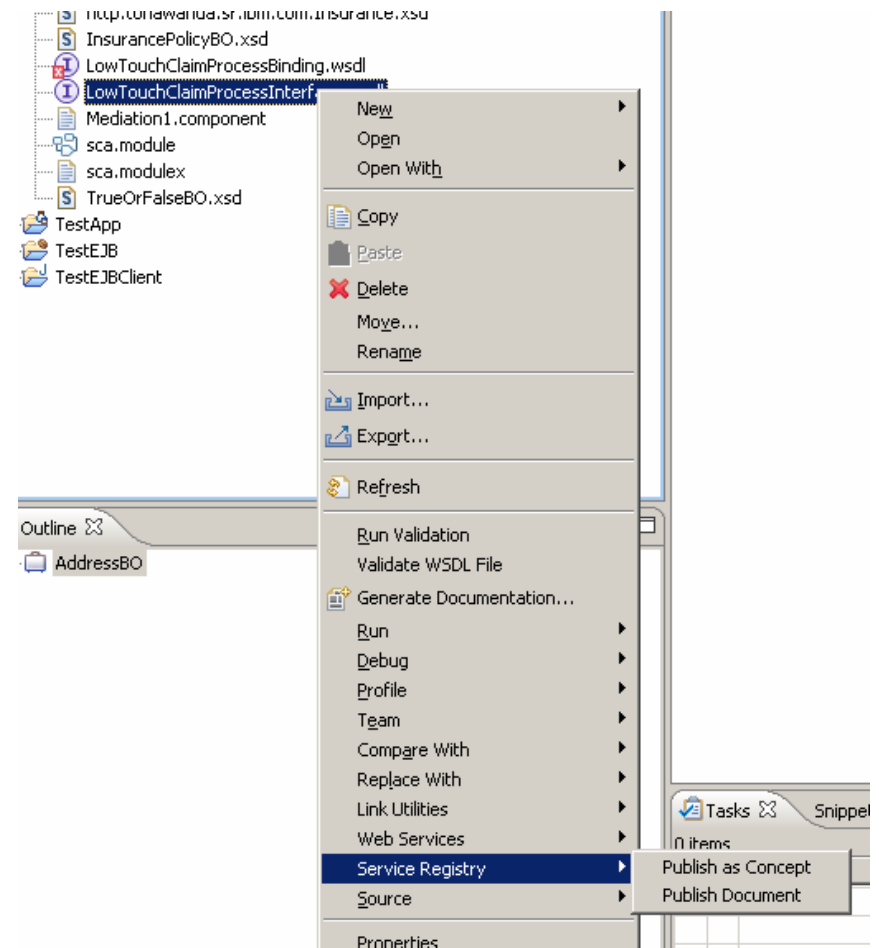
From the *Resource* Perspective

# WSRR Eclipse Plug-in: Retrieve/Import
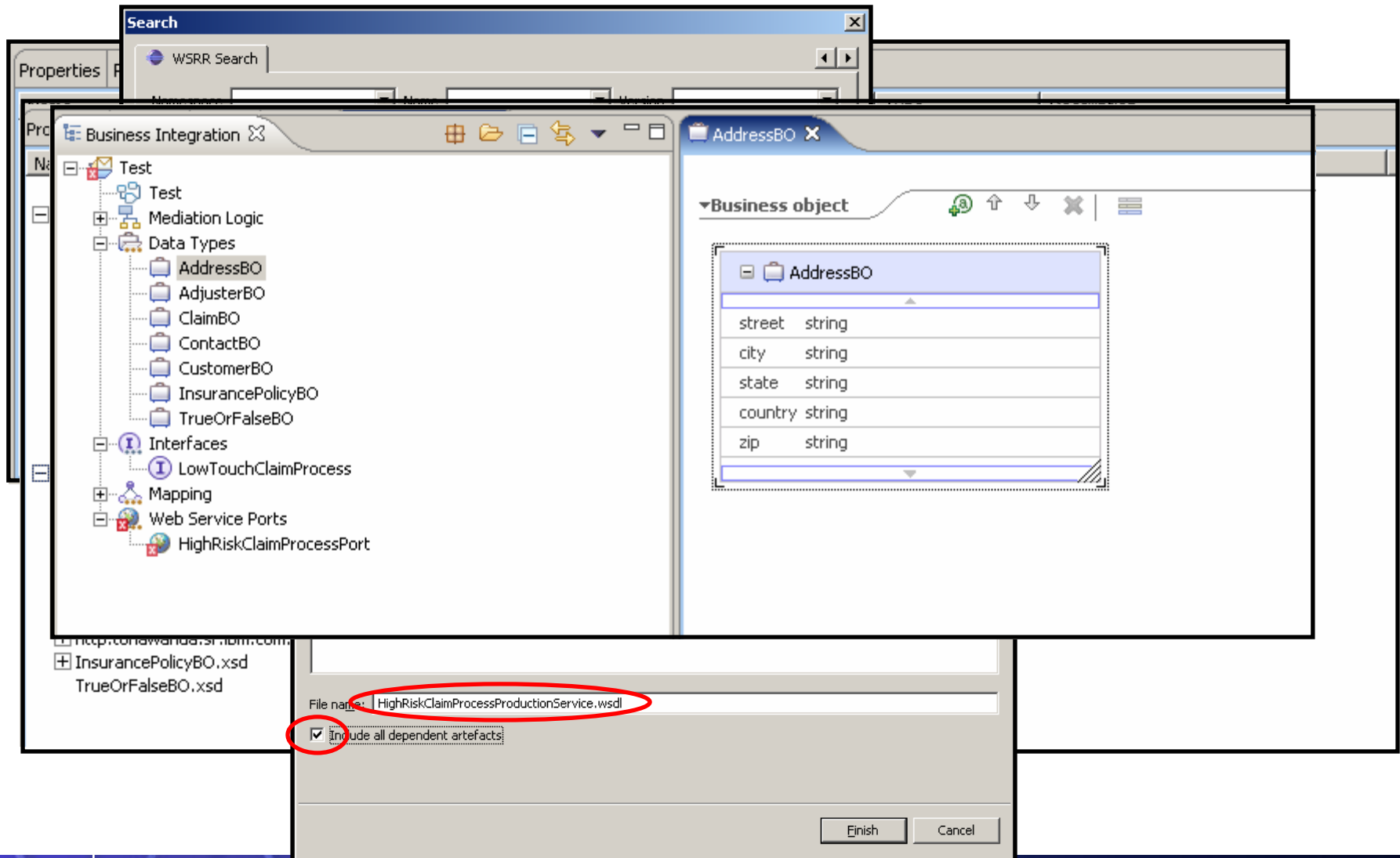
# Discussion / Q&A…

# UDDI alone is not sufficient to handle the demands of SOA

## UDDI

*(Universal Description, Discovery and Integration)*

Designed as "phonebook" for external WEB services

Only allows publish and find of WEB services

Lacks metadata repository to help manage and govern service interactions

Inflexible data model

Highly technical and not readily useable by end users

Not widely adopted

## SOA needs:

- **Service metadata repository**
- **Optimized service registry**
- **Based on latest web services standards**

**Based on emerging web services management standards optimized for SOA**

- *WS – Resource Transfer*
- *WS – Event Notification*
- *WS – Metadata Exchange*

**Driven by industry leaders including**

**Integrated with UDDI**

# WebSphere Service Registry & Repository UDDI Integration

**Integration based on Copy and Synchronize**
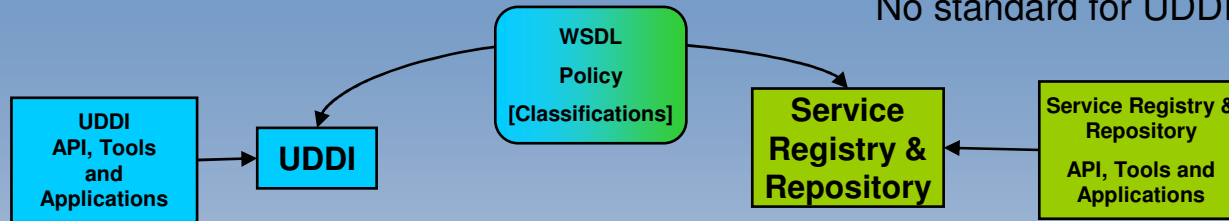
UDDI to WSRR mapping needed to capture categories provided on the UDDI entries

No standard for UDDI taxonomies



**Initial Population of WSRR from UDDI**
**Utility to extract and copy service metadata**
  •**WSDL, as defined by Best Practice/Technical Notes**
  •**Policy, as defined by WS-PolicyAttachment**
**Because UDDI entities are not time-stamped, this is run only once and other techniques are used to keep UDDI and the WSRR synchronized**
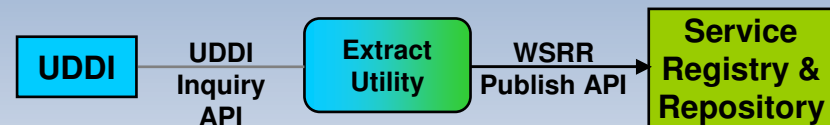
**UDDI to WSRR Synchronization.**
**WSRR listens for additions/change made in UDDI using the UDDI Subscription API and makes the appropriate changes in the WSRR using the standard WSRR API.**
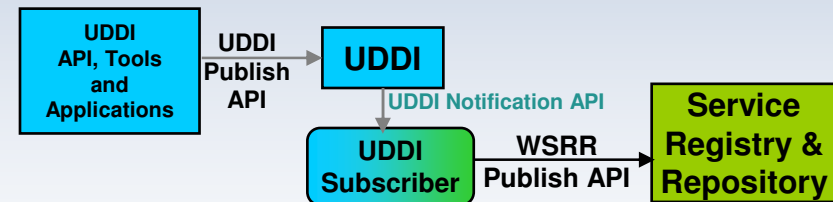
**WSRR to UDDI Synchronization**
**Additions/changes to WSDL and Policy made to the WSRR using the WSRR API, tools and applications are reflected in UDDI using the standard UDDI Publish API**

*All plans and proposed features subject to change*

# WSRR R1 Features

**Core Function**

| |
|---|
| C2 Access Control |
| C4 Admin |
| C5 Web Service API |
| C7 SDO Support - Change Tracking & Tuscany |
| C9 Query Language including XPATH |
| C10 Classification Query extensions / Minerva |
| C11 Event Processing framework |
| C12 Caching |
| C15 Maintain Support for Current MDS version |
| C16 Install |
| C18 SCA Support |
| C19 Core XSD based Entity Typing and Validators |
| C20 Expose predecessor |
| W4.2. Eclipse plug-ins for Discovery and Publish to integrate WSRR with tooling. |
| W4.4. Import/Export facilities to provide simple support for staged environments. |

**Governance**

| |
|---|
| G1 Governance Architecture |
| G2 State Machines |
| G3 Governed Entity Typing |
| G6 Transition Extensions |
| G7 State Based controls |
| G9 Governance Socialization - email and subscription |
| G11 Impact Analysis Queries |

*Integration (delivery via EAP at eGA)*

| |
|---|
| W4.3. UDDI integration to support co-existence for customers already committed to UDDI. |
| W4.6. Integration with CAM4SOA to provide essential QoS metrics to the BSR |
| W5.1. ESB Integration – Mediation & Tooling |
| W5.2. Message Broker Integration – Node & Tooling |
| W5.5. Support for Tivoli security integration with WSRR (T2) |
| W5.6. Support for Rational RAM integration with WSRR (R2) |

# Candidate items for WSRR R2

- **Enhance Standardisation Support**

  – Drive and implement early standard for registries / service meta-data management

  – Extend support for existing standards

- **Leverage Integration with IBM and 3rd parties:**

  – Policy Enforcement

  – Service Monitoring & Management

  – Development & Asset Management

  – Deployment and Change management

  – SOA Foundation run-times

- **Improve Governance** capabilities

  – Service life-cycle management

  – Policy Management

  – Service Version support

  – Extend Validation and Notification plug-ins based on additional customer requirements

- **Improve Usability**

  – Administration Facilities for:

    • Classification systems

    • Access Control - XACML

    • Perspective editor / wizard

    • Life-cycle governance

    • Entity templates

- **Exploit Service Semantics**

  – Exploit and extend WSRR meta-data semantics to describe SOA services to meet customer requirements

- **Extending Deployment environments**

  – Support WAS Clustering

  – Support additional OS & DBs

  – Federation between registries

  – Synchronisation with UDDI registries