Oracle Enterprise Service Bus:
Technical Architecture and Product Update

Dave Berry: Sr. Manager, Oracle Integration
Demed L'Her: Sr. Principal Product Manager, Oracle Integration
Program

- Oracle Enterprise Service Bus Overview
  - Foundation for Oracle SOA Suite
  - Features: Virtualize, Transform, Route

- Architecture
  - Clusters, Life Cycle Management
  - Transactions, Exception Handling

- Conclusion
  - Summary
Oracle Enterprise Service Bus: The Foundation for SOA

ESB is a multi-protocol fabric to separate integration concerns from applications and business logic

How is this achieved?

1. **Virtualize Endpoints**: From resources to services
2. **Transform**: Convert data to target formats
3. **Route**: Reliably transport and route data over a variety of protocols

Expose everything as web services

1. Standards-based: XPath, XSLT, SOAP, JMS, JCA, …
2. Hot-pluggable: J2EE, JMS, Database, …
Patterns

**Virtualized Service**
- 2 way sync
- Content based responder
- Static inbound WSDL
- Operational flexibility

**Fan In/Out**
- 1 way store and fwd
- Sync or Async
- Multiple transactions
- End to end

**Response Forward**
- 1 way inbound
- 2 way outbound
- Route based on callout response
- Forward original document
Architecture

- **Metadata Server – Integration Server**
  - Central DT interface: JDeveloper, ESB Control, Import/Export
  - Changes pushed to RT server as transactions through JMS
  - ESB Control centralized management/monitoring
  - Configurable back end repository: database, webdav

- **Runtime Server**
  - Fast In Memory Service and Artifact Cache
  - Flexible topology, clustering, external load balancing
  - ESB System is underlying unit of scalability
  - Async Routing Rules enable distributed Bus topology

- **Leverages underlying J2EE architecture**
  - Runs as J2EE application
  - Utilizes JMS provider
  - Transactions, scalability, high availability
3-Tiered Architecture

UI Tier
- JDev (design)
- ESB Control (monitoring)
- register
- sync
- update
- monitor

Middle Tier
- Metadata Server
- Runtime Servers
- JMS
- JDBC

Data Tier
- Artifacts: XSD, XSLT, WSDL, Maps
- Relational: Service MD, Routing Rules, Instances, Errors
- JDBC
Oracle ESB Queuing Architecture

MDS
- Control Topic (Content Msgs)
- Monitor Topic (Tracking)
- Error Topic (Error)

Runtime Servers
- Control
- Async Topic
- Resubmission Topic
- Monitor Topic (Tracking)
- Error Topic (Error)

Data Tier
- Oraesb Schema
  - Service data
  - Instance data
- Internal Topics
  - Control Topic
  - Async Topic
  - Resub Topic
  - Monitor Topic
  - Error Topic
- Error Hospital
Shared Meta Data Cluster

- **Flexible Cluster Topology**
  - No Single Point of Failure - Guaranteed Delivery
  - ESB System is lowest level of granularity
  - ESB System configured to ESB cluster: many to 1
  - ESB RT server configured to 1 ESB cluster (many systems)

- **Symmetric for High Availability**
  - All systems configured to same cluster
  - All RT servers load the same cluster

- **Asymmetric for Scalability**
  - Many clusters with multiple systems
  - RT servers load different clusters
  - Optimized service memory cache

- **Best Practice: Hybrid Cluster Topology**
  - Combination of Symmetric and Asymmetric services clusters
  - External Load Balancer: ESB system has external “virtual” host/port
Flexible Deployment Strategies

Symmetric

Services in Metadata Repository

Update Customer
New Employee
Create Invoice

Asymmetric

Siebel
PSFT

Update Customer
New Employee
Create Invoice

ESB default cluster

Cluster = Siebel

Cluster = PSFT

Load Balancer

/Siebel
/PeopleSoft

/ORACLE/
ESB System/Cluster Configuration

- ESB_Parameter table “DT_OC4J_HOST”
- File “esb_config.ini” cluster_name=ESB
Life Cycle Management

- **JDeveloper Primary Developer Tool**
  - Model services, adapters, transformations, routing rules
  - Register service metadata to MDS/Integration server
  - Sync with MDS for any ESB Control MD changes

- **ESB Control for Some Service Reconfiguration**
  - Configure ESB system: cluster and load balancer information
  - Configure endpoint properties: service location, retry, file, directory...
  - Define Trackable Fields and Domain Value Maps
  - Edit Routing Rules, filter expressions

- **Export/Import**
  - Ant based scripts with System or Service level granularity
  - Dev to Test to Production: JDev $\rightarrow$ MDS $\rightarrow$ Export $\rightarrow$ zip $\rightarrow$ Import $\ldots$
  - Manage endpoints with JNDI, endpoint properties, custom ant tasks
Life Cycle Management

DEVELOPMENT
- JDeveloper + RCS

DEPLOYMENT
- From JDev
- Via ANT-scriptable import/export scripts

MONITORING
- ESB Console
- AS Control
Transactions

• Global End-to-End JTA/XA Transactions
  • BPEL ↔ ESB ↔ BPEL
  • JCA ↔ ESB ↔ WSIF

• ESB Inherits Inbound Global Transactions
  • “Async” Routing Rules ends scope of current transaction
  • New ESB initiated transactions grouped by ESB System

• Transaction Exception Handling and Rollback
  • Errors on existing inbound transactions rolled back to initiator
  • Errors on ESB initiated transactions can be resubmitted
  • End-to-end message flow terminates on first failed service regardless of transaction state or owner
Transaction Overlay

Async Rule is New Transaction

1. Only Sync Rules
2. Async in Same ESB System Rule
3. Async in Different ESB System Rule
4. Async Rule is New Transaction
Exception Handling

• **ESB Control Error Hospital**
  - View all failed flows including original document
  - Resubmit/edit “retryable” errors on ESB initiated transactions
  - Roadmap: Batch resubmit, auto retry

• **ESB Control Instance Tracking**
  - View all failed flows including original document
  - Resubmit/edit “retryable” errors on ESB initiated transactions
  - Roadmap: Management API

• **Non Transactional Endpoints**
  - File, FTP, SOAP ...
  - Default and custom exception handlers: File:// JMS:// BPEL://
Exception Handling in Instance Data

- Transaction Impact
- Error Message, Trace, Payload
- Resubmitable Errors
Conclusion

• The Right Architecture for SOA Infrastructure
  • Deployed to any J2EE Application Server

• Flexible Topologies
  • Highly Available and Scalable
  • External Load Balancing

• Reliable Architecture
  • Full Transaction Support
  • Configurable Exception Handling

• Configurable Life Cycle Management
  • Metadata server

For more info: visit http://otn.oracle.com/soa