



Oracle Develop

The Premier Conference for Developers

DEVELOP

OCTOBER 11-13, 2009 / HILTON / SAN FRANCISCO

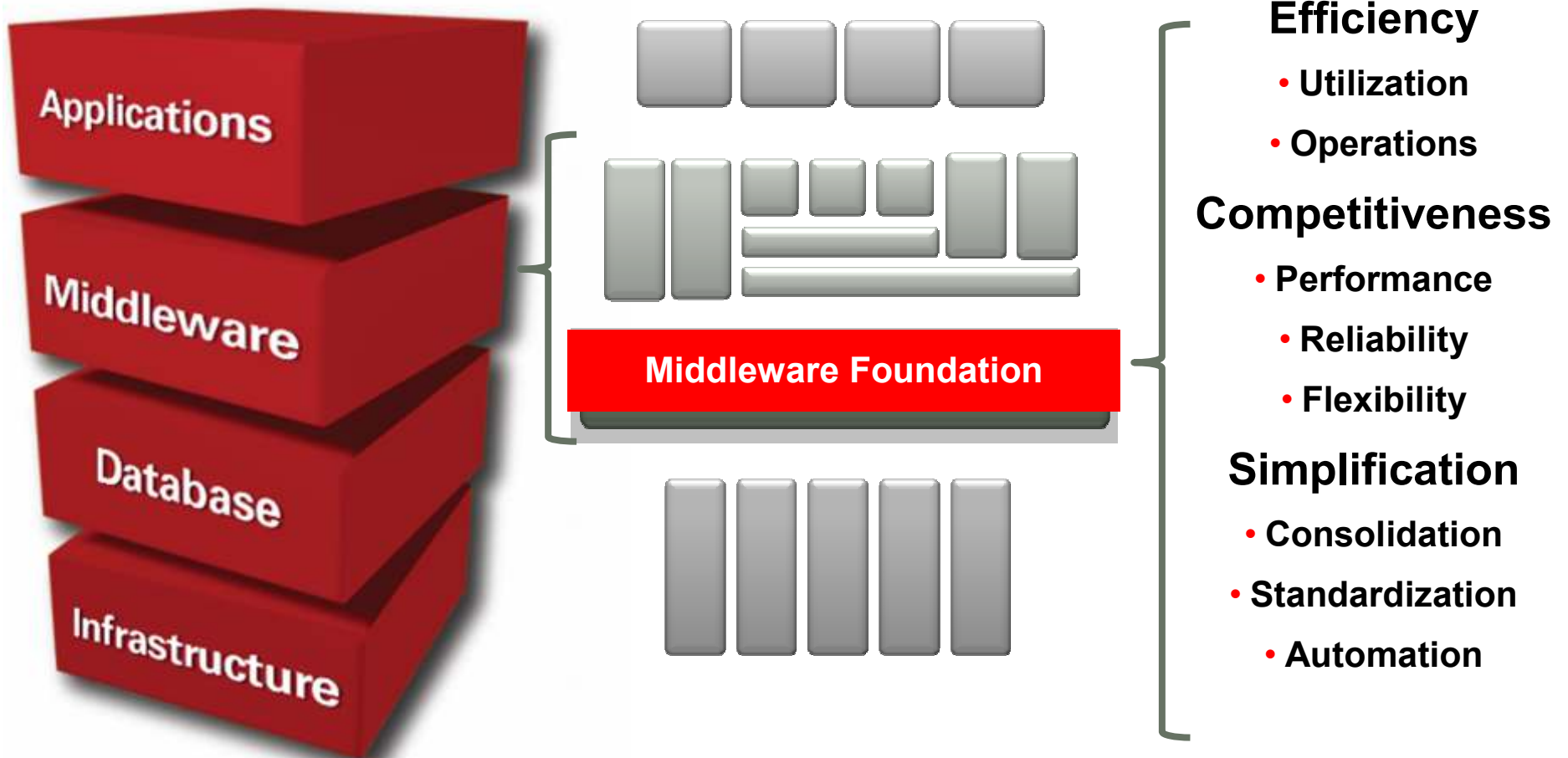
ORACLE®

Closer Look at Enterprise Service Bus

Deb L. Ayers
Sr. Principle Product Manager
Oracle Service Bus
SOA Fusion Middleware Division

The Role of the Foundation

Addressing the Challenges



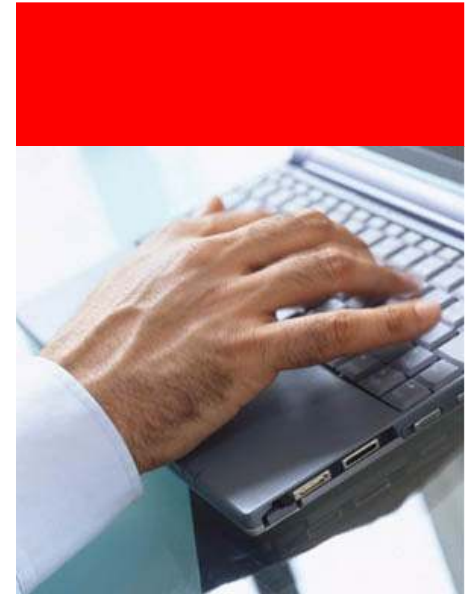
ORACLE®

Oracle Fusion Middleware Technologies



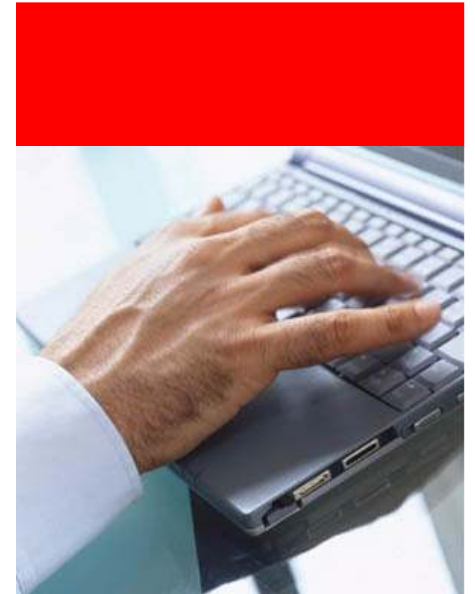
Agenda

- Why Service Bus ?
- Enterprise Patterns
- Closer Look
 - Sync to Async Bridge
 - Service Result Cache
 - EJB Proxy
- Q & A

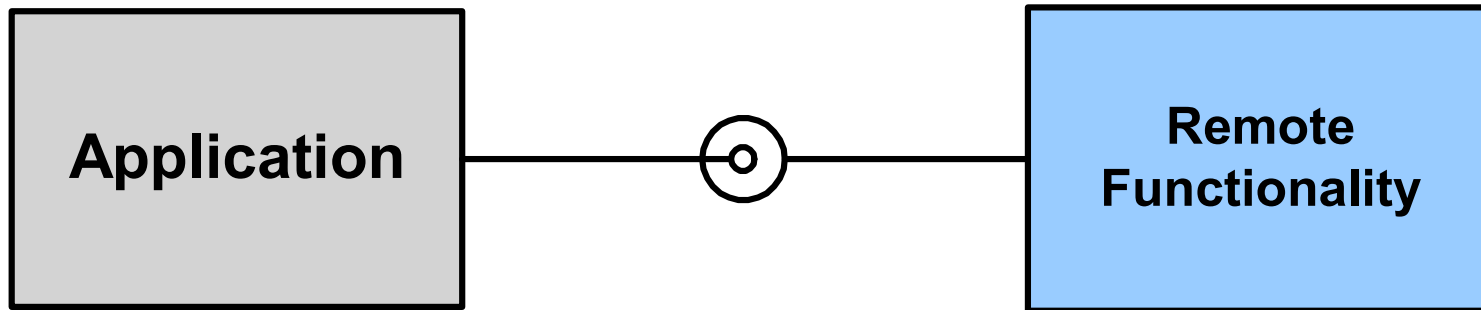


Agenda

- Why Service Bus ?
- Enterprise Patterns
- Closer Look
 - Sync to Async Bridge
 - Service Result Cache
 - EJB Proxy
- Q & A

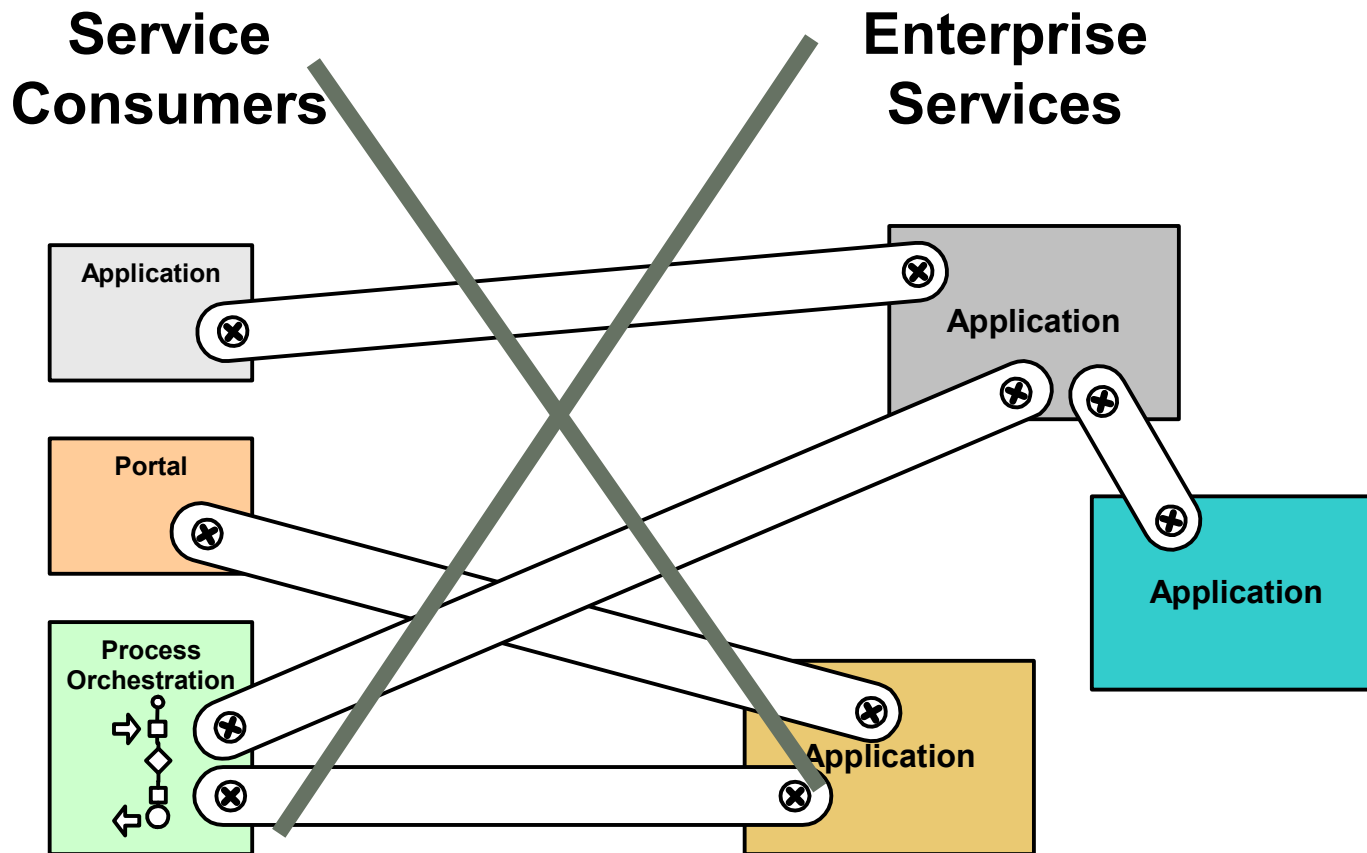


Service Integration: Developer Perspective

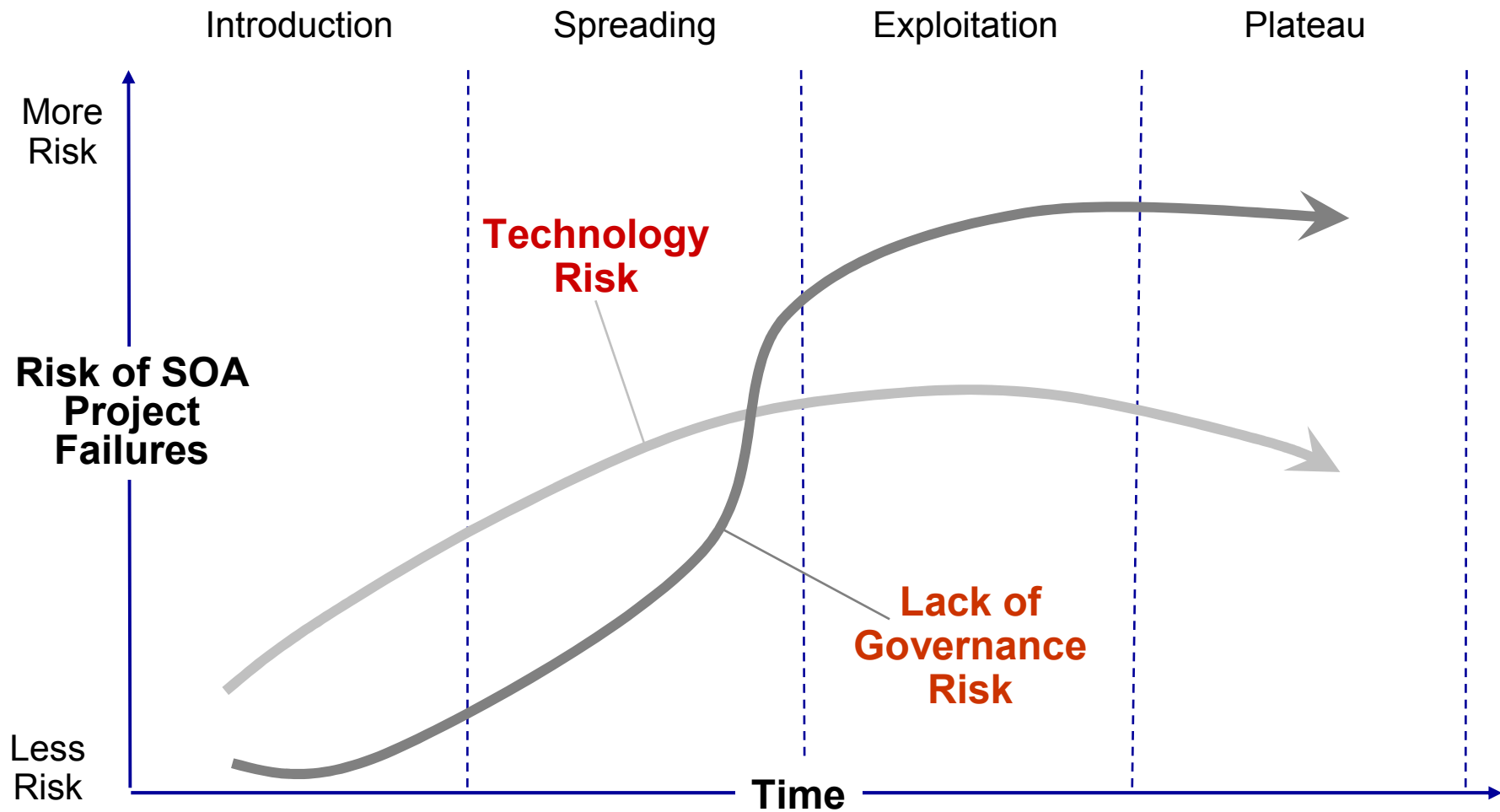


Brittle Over Time

With or Without Web Services

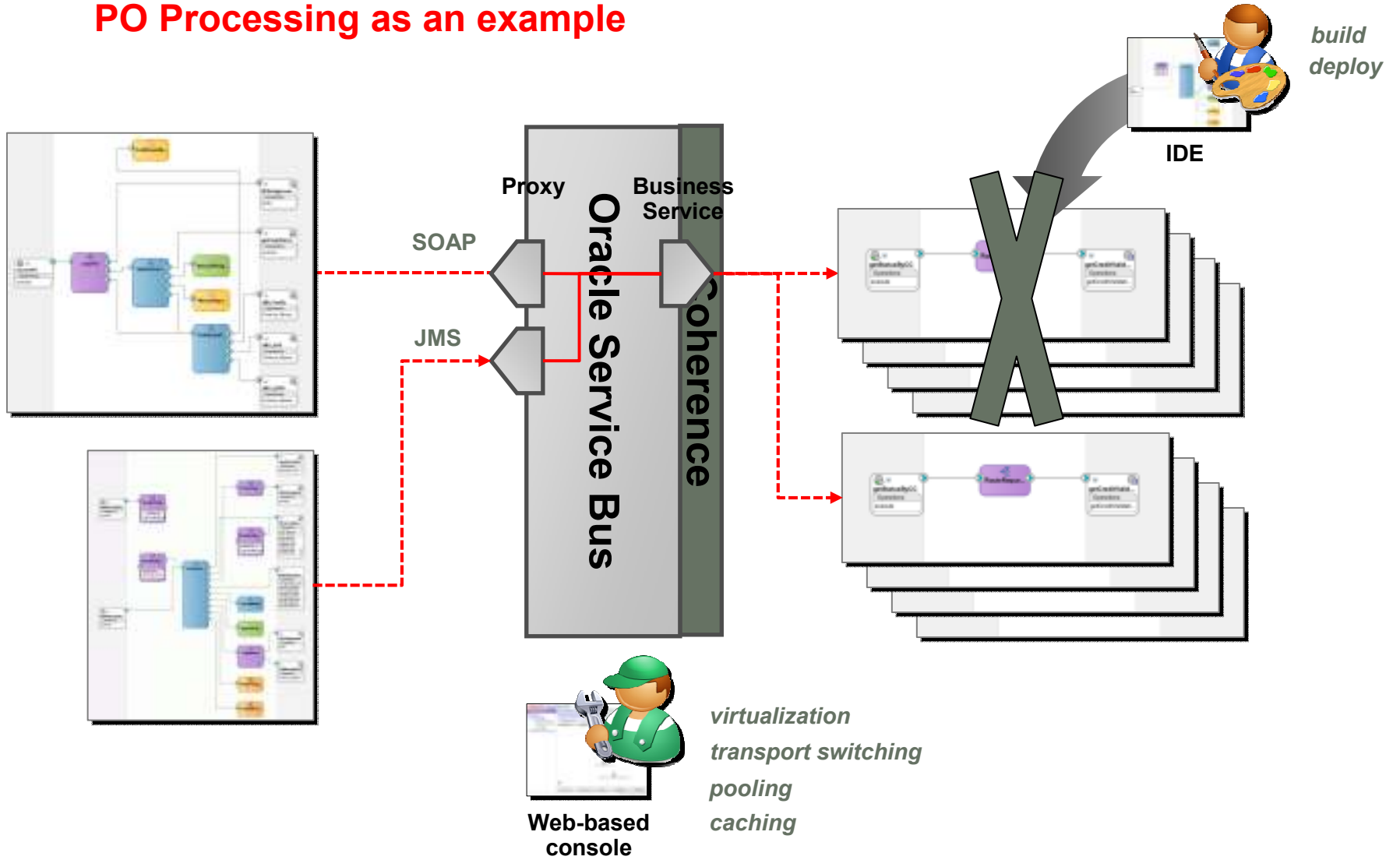


Why SOA Initiatives Fail: Technology or Governance?



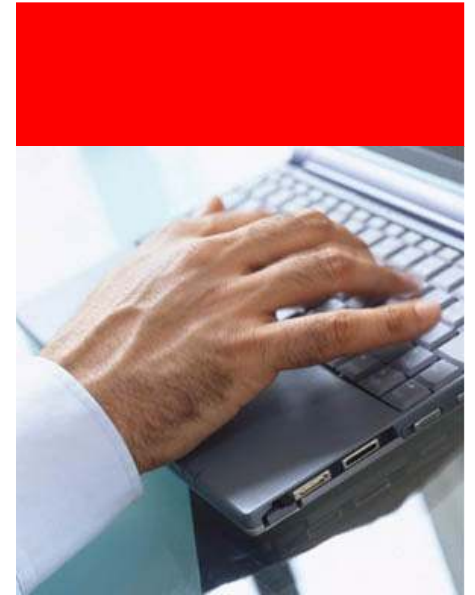
Service Bus

PO Processing as an example

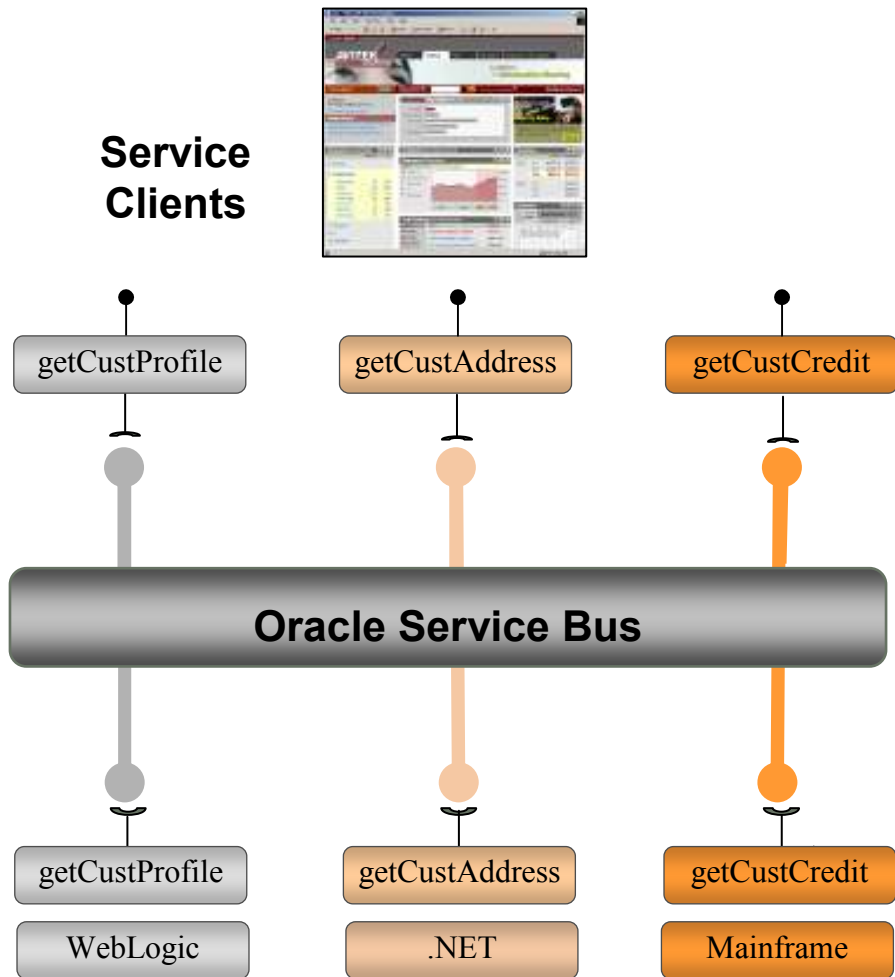


Agenda

- Why Service Bus ?
- Enterprise Patterns
- Closer Look
 - Service Result Cache
 - EJB Proxy
- Q & A



Patterns for Service Bus



Customer Use Cases:

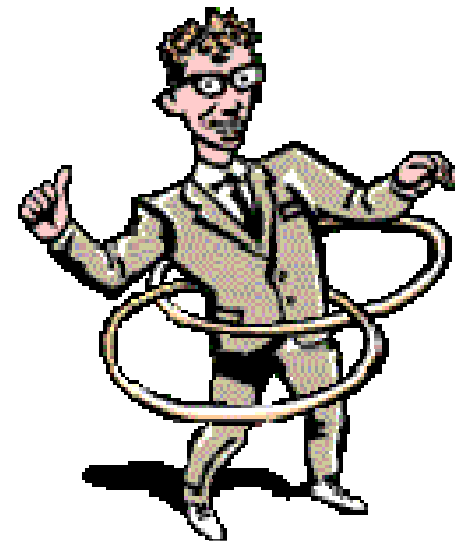
- Backend Application Integration
- Portal application consumes business services from back-end applications.
- Eliminate the brittleness of services using a SOA based architecture.

Requires ESB Characteristics:

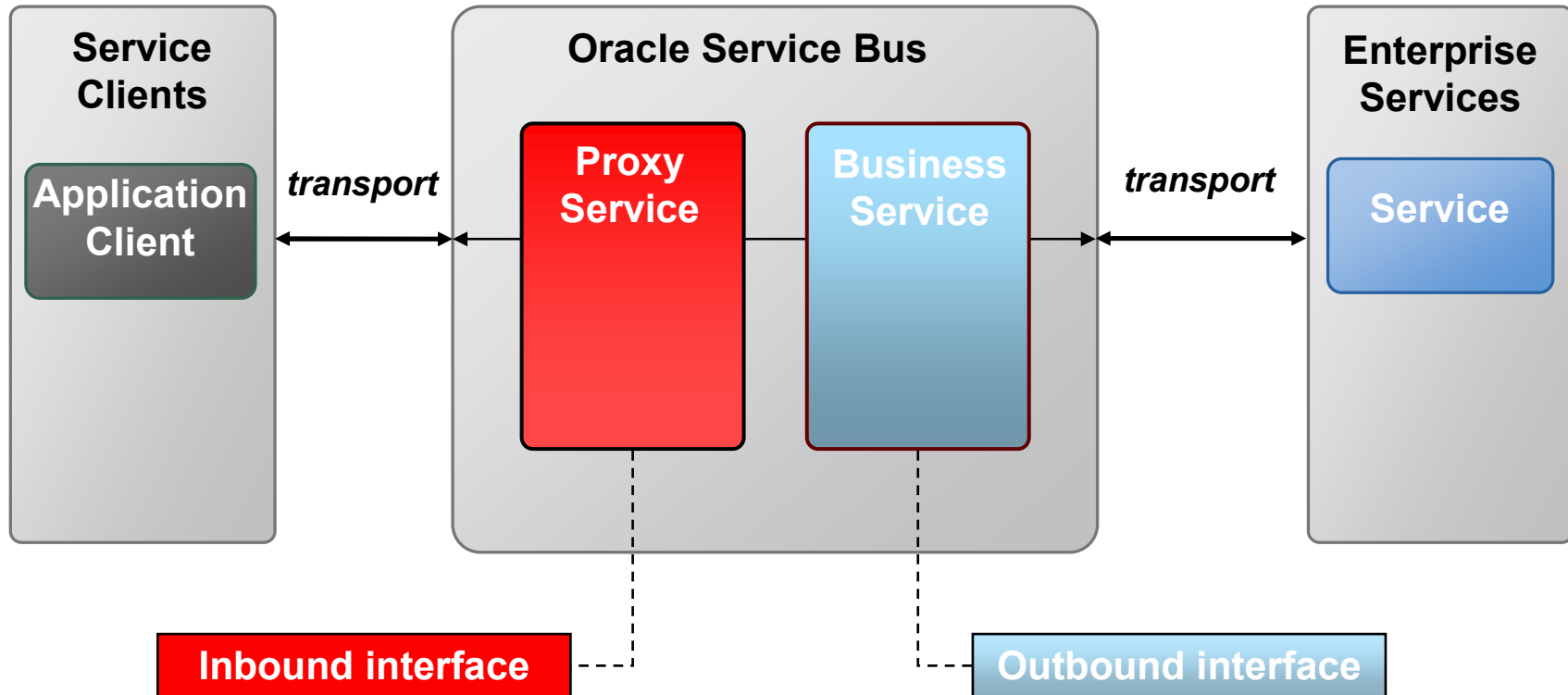
- Heterogeneous messaging backbone
- Content based routing
- Service enrichment
- Monitoring and reporting
- SOA based security
- Service workload and management
- Message Guarantees
- Service orchestration
- Distributed services across the enterprise
- Service discovery

Common Service Types

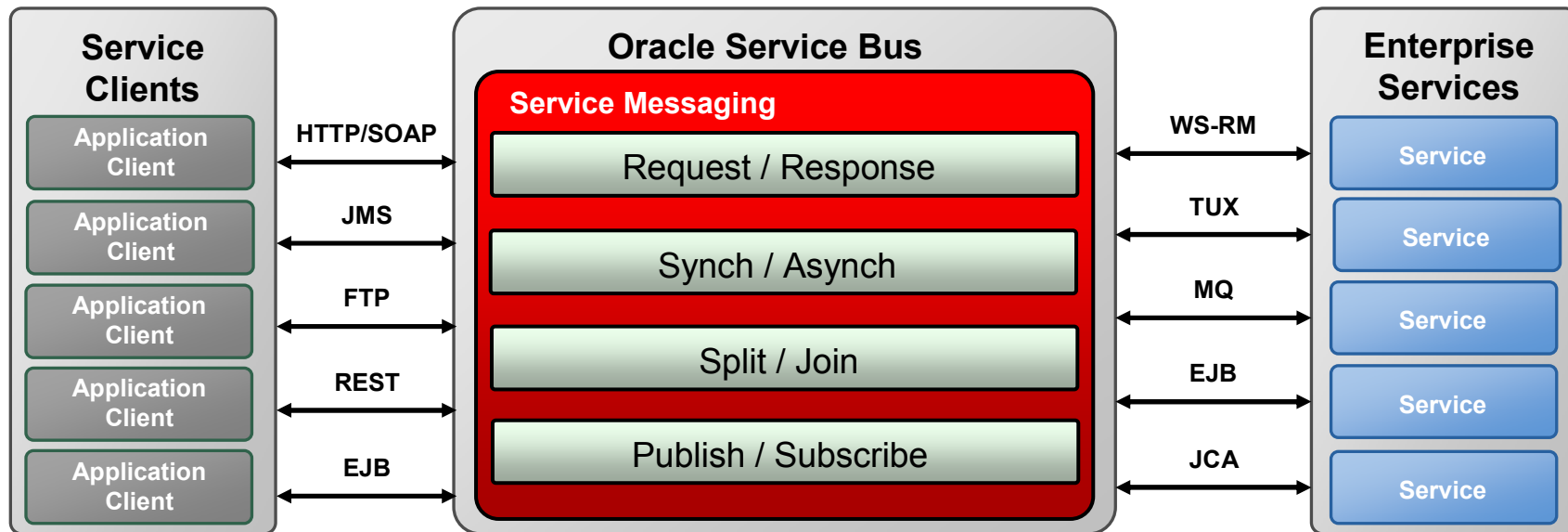
- *Traditional Web Services*
 - Pre-negotiated Interfaces Contract (WSDL)
 - Standards in place, supported by many vendors
 - SOAP over HTTP
- Legacy Services
 - Non-XML (XML) over File, FTP, MQ, JMS
- POX (Plain Old XML)
 - Structure of Payload to determines action
 - XML over HTTP
- REST (Representational State Transfer)
 - Based on Pattern of Service Invocation
 - Nouns vs. Verbs
 - URIs over HTTP



Proxy & Business Services



Adaptive Messaging In a Nutshell...

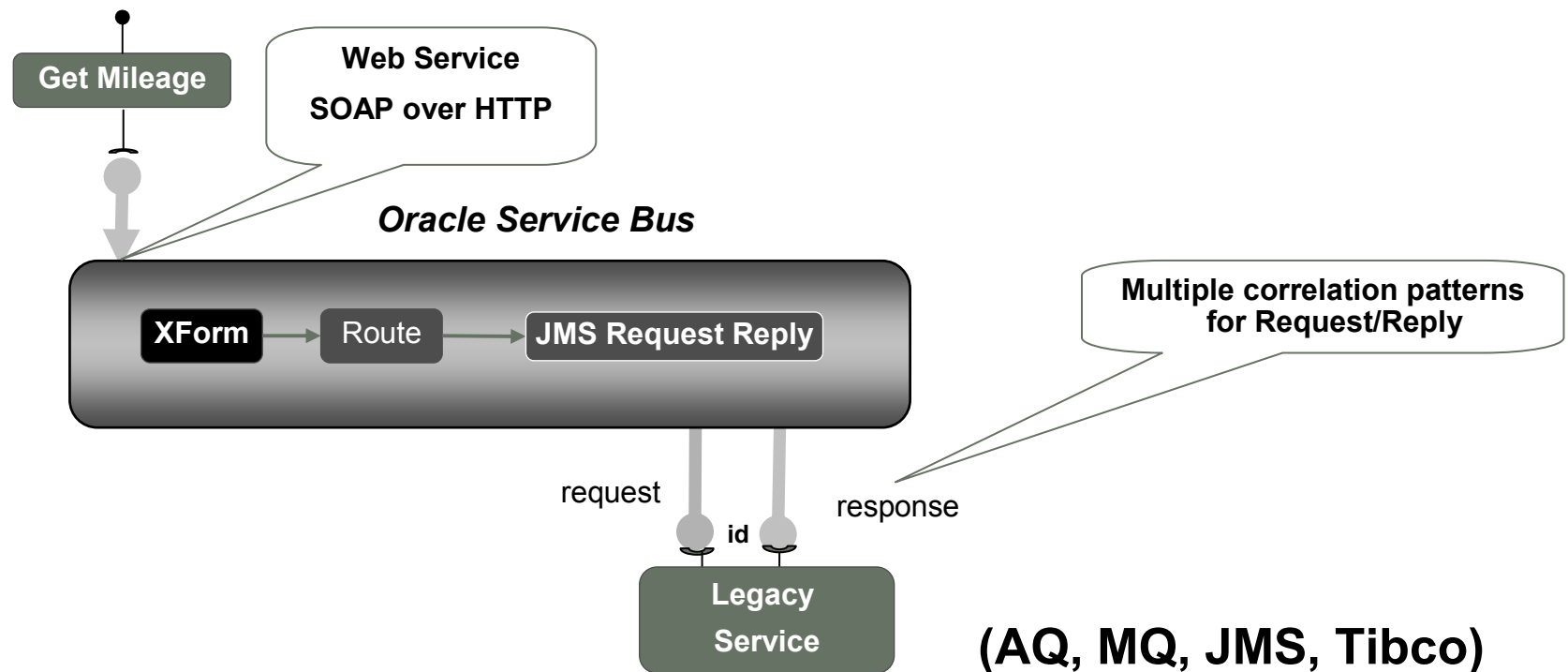


- Any to Any Protocol
- Any to Any Payload
 - XML
 - non-XML
 - Binary
- No WSDL Required
- Multiple communications paradigms
 - Request/response
 - Synchronous and asynchronous
 - One-to-many, many-to-one
 - Pub-sub
 - Mix-and-match (e.g. sync-to-async)

Web Service (SOAP) -> MQSeries or JMS

Let's Build It...

- Create a simple service with a Web services interface that performs data translation and protocol conversion to invoke an existing service through a asynchronous message, wait for a response, and return a synchronous reply to the caller.



(AQ, MQ, JMS, Tibco)

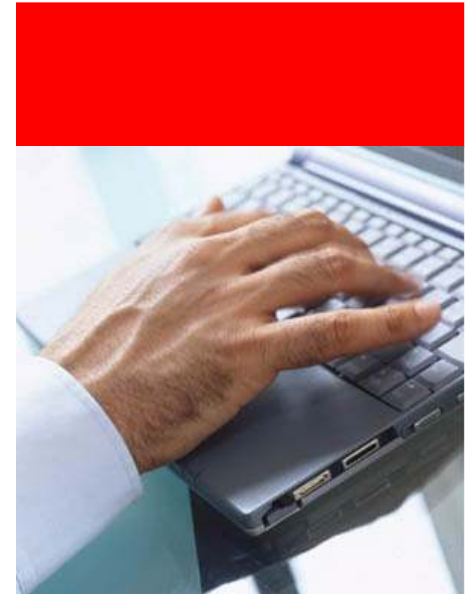
ORACLE

Summary of ESB Usage Patterns

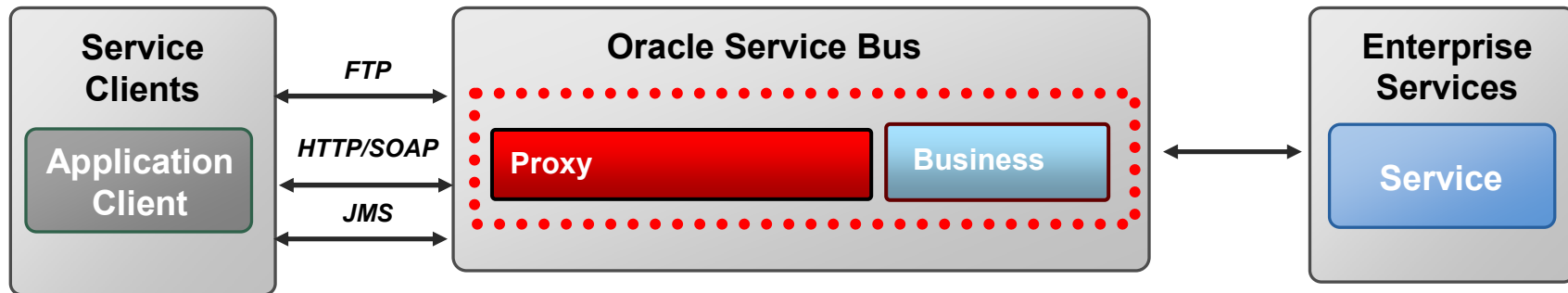
Typical Use Cases	Customer
Content, Metadata, Identity, -Based Routing	✓
Asynchronous Request/Reply	✓
Sync client to Async Business Service	✓
Security and Access Control	✓
Data services integration	✓
Customized Data Reporting	✓
Monitoring / SLA	✓
Heterogeneous Messaging	✓
Error Handling in Pipeline	✓
Business Service Load-balancing & Failover	✓
Transport Header Override	✓
Service Lifecycle Management	✓

Agenda

- Why Service Bus ?
- Enterprise Patterns
- Closer Look
 - Service Result Cache
 - EJB Proxy
- Q & A



Message Flow Transaction

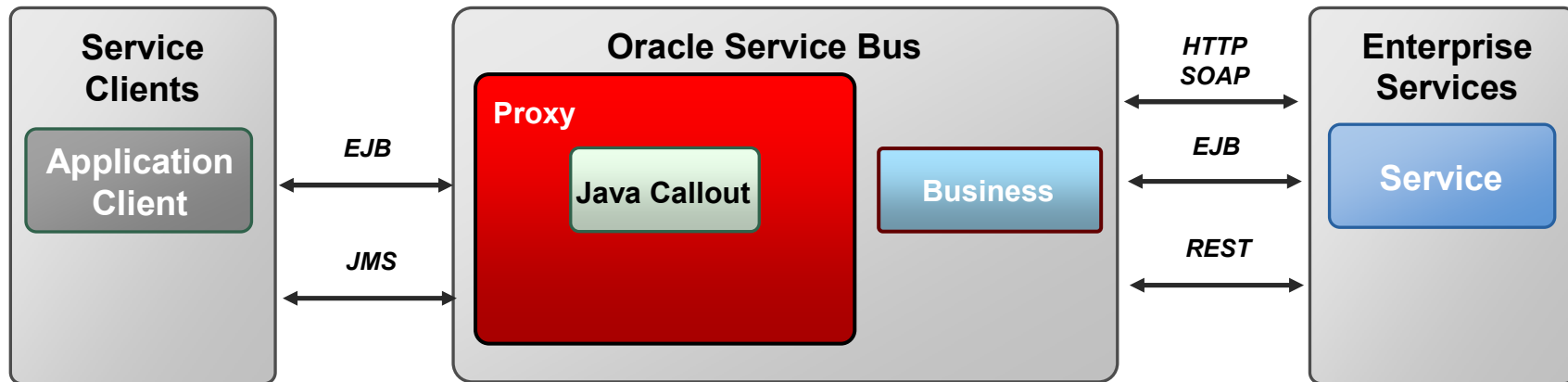


- Atomic Message Flow
 - Simple as checking a box
 - All work committed or aborted
- Two settings for Proxy
 - Transaction Required ?
 - Same Transaction For Response ?

Benefits

Message flow can execute within TX regardless of the Inbound Protocol

Native EJB Inbound and Outbound Transport



- New native EJB Transport
 - Supports 2.1 and 3.0
 - Native Java Types can be passed along to JMS, Java Callout
 - Full Transaction semantics supported

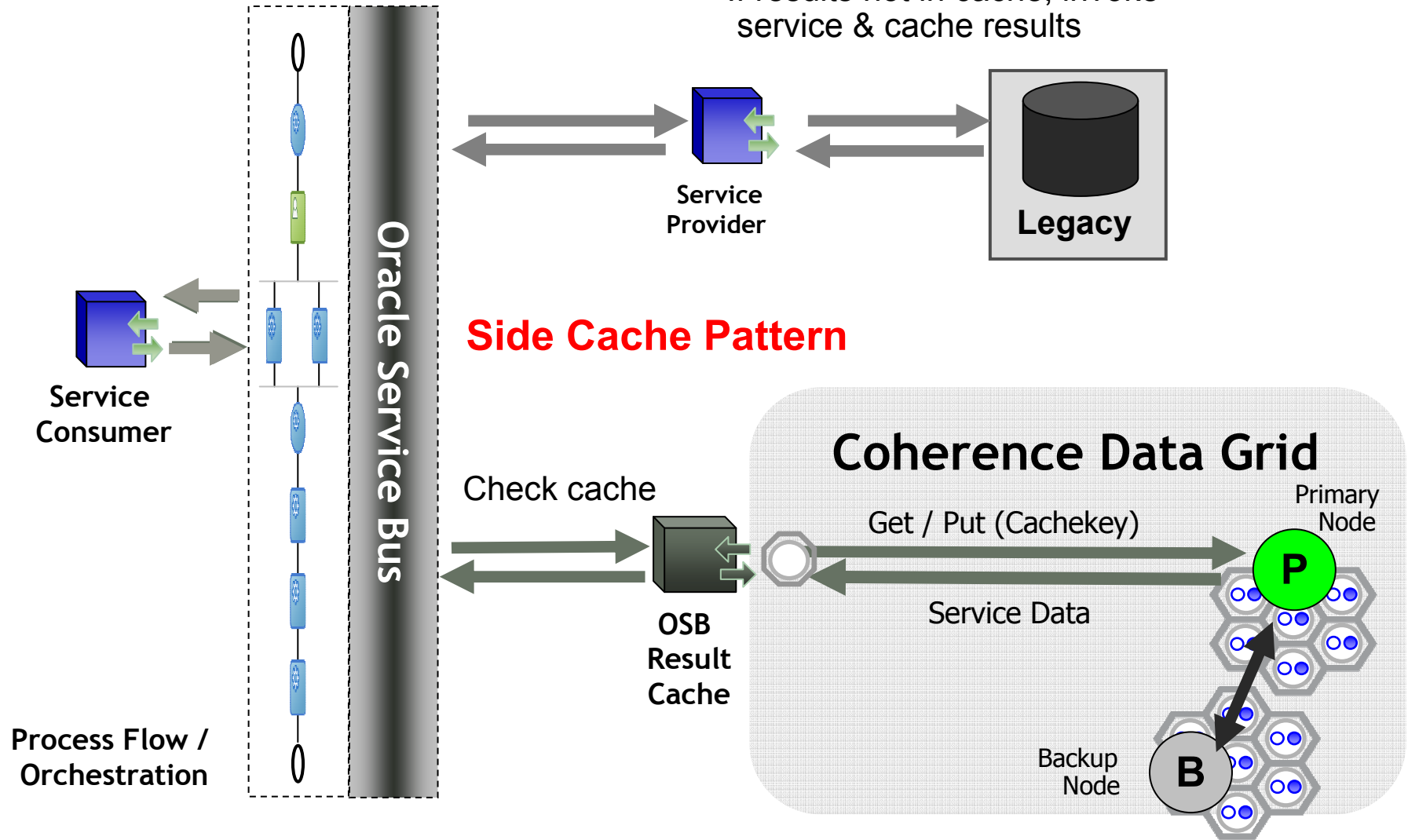
Benefits

- Mediate existing legacy EJB infrastructure

Service Result Cache

Scaling Application Infrastructure

If results not in cache, invoke service & cache results



Service Result Cache with Coherence

Checkbox Caching

Business Service

- ✓ Cache Results?
- ✓ Cache Token
- ✓ Time To Live

The screenshot shows the 'Edit a Business Service' configuration page for 'demo/stock/StockQuote'. The 'Cache token expression' field contains the value 'jcoo:stock:getQuote?token=/test/1'. The 'Expiration Time' section is highlighted with a red circle and contains the following options:

- Use Default
- Duration: 0 days 02 hours 00 min : 00 sec
- XQuery Expression: Request

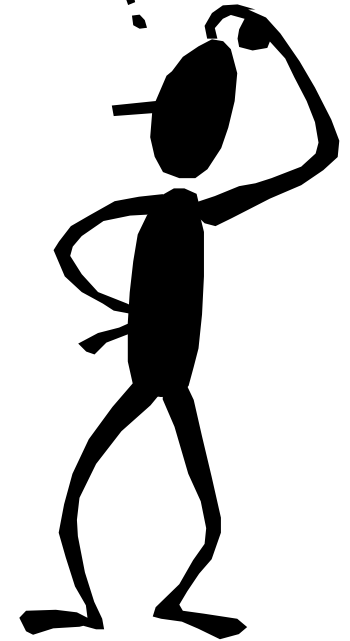
A tooltip for the XQuery Expression option reads: 'Enter an XQuery expression. It is used to calculate the Time To Live(TTL) in mill seconds for cached entries'. The 'DEMO' watermark is overlaid on the center of the screenshot.

Result Cache is an implementation of Side Cache pattern

Service Bus vs. Process Orchestration ?

Process Orchestration

- Service contains business logic
- Service requires complex transaction management
 - Requires multiple transactions
 - Compensation logic required on rollback
 - Short or long-lived process
- Exception handling requires Human workflow
- Service needs to handle asynchronous callbacks reliably



Service Bus

- Service contains protocol, routing and transformation logic
- Service has short-lived, single transaction semantics

Final Thoughts...

Trends Emerging



- **Federated ESB**
 - Not enough to simply distribute across buses.
 - Management is KEY.
 - Homogenous until vendors decide on management standards to allow uniform service provisioning & routing updates.
- **Event-driven SOA**
 - Dynamic, unpredictable business events being correlated real-time feeding into SOA infrastructure
- **Web 2.0 meets SOA & Traditional IT infrastructure**
 - Watch out - IT Culture class emerging!
 - Service Bus will help adapt Traditional SOA to more ad-hoc services like REST, POX with security.

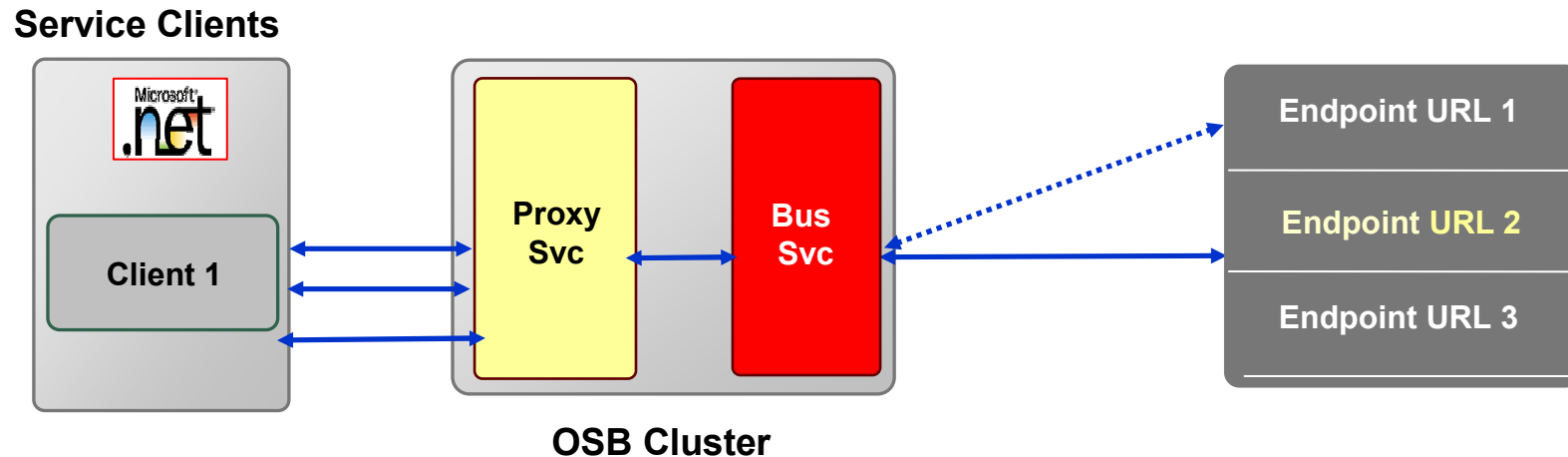


Discussion



Advanced Service Pooling

Routing to active endpoints, service load balancing



**If a URI is non-responsive, take the URI out of the pool
Bring the URI back in the pool when it is back-up**

- Option for the system to automatically take non-responsive URIs out of the pool, and put them back in as they become responsive
- Alerts will be generated when the status of the endpoint changes from Up → Down and vice versa