AquaLogic Service Bus

Wolfgang Weigend

Principal Systems Engineer

bea
Think liquid.

What to consider when looking at ESB?

- Number of planned business access points
- Reuse across organization
- Reduced cost of ownership via configuration
- IT Productivity
- Flexibility and availability
- Scalability and Interoperability
- Cross Integration based on standards
- Single product suite
- Ease of use and Implementation

AquaLogic Service Bus

- **What it does**
  - Heterogeneous service virtualization
  - Dynamic routing transformation
  - Service registration and discovery
  - Monitoring & management
  - B2B connectivity
- **What's special**
  - Fully configuration based
  - Comprehensive SOA integrity
  - Life-cycle management
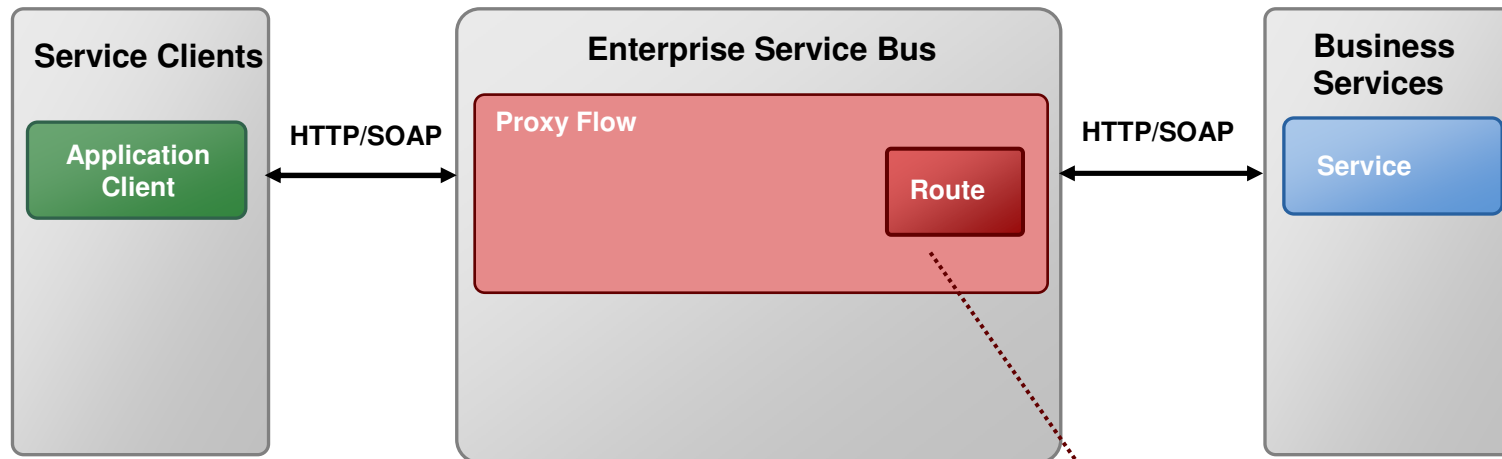 - UDDI registry integration
  - Proven: R-A-S-P
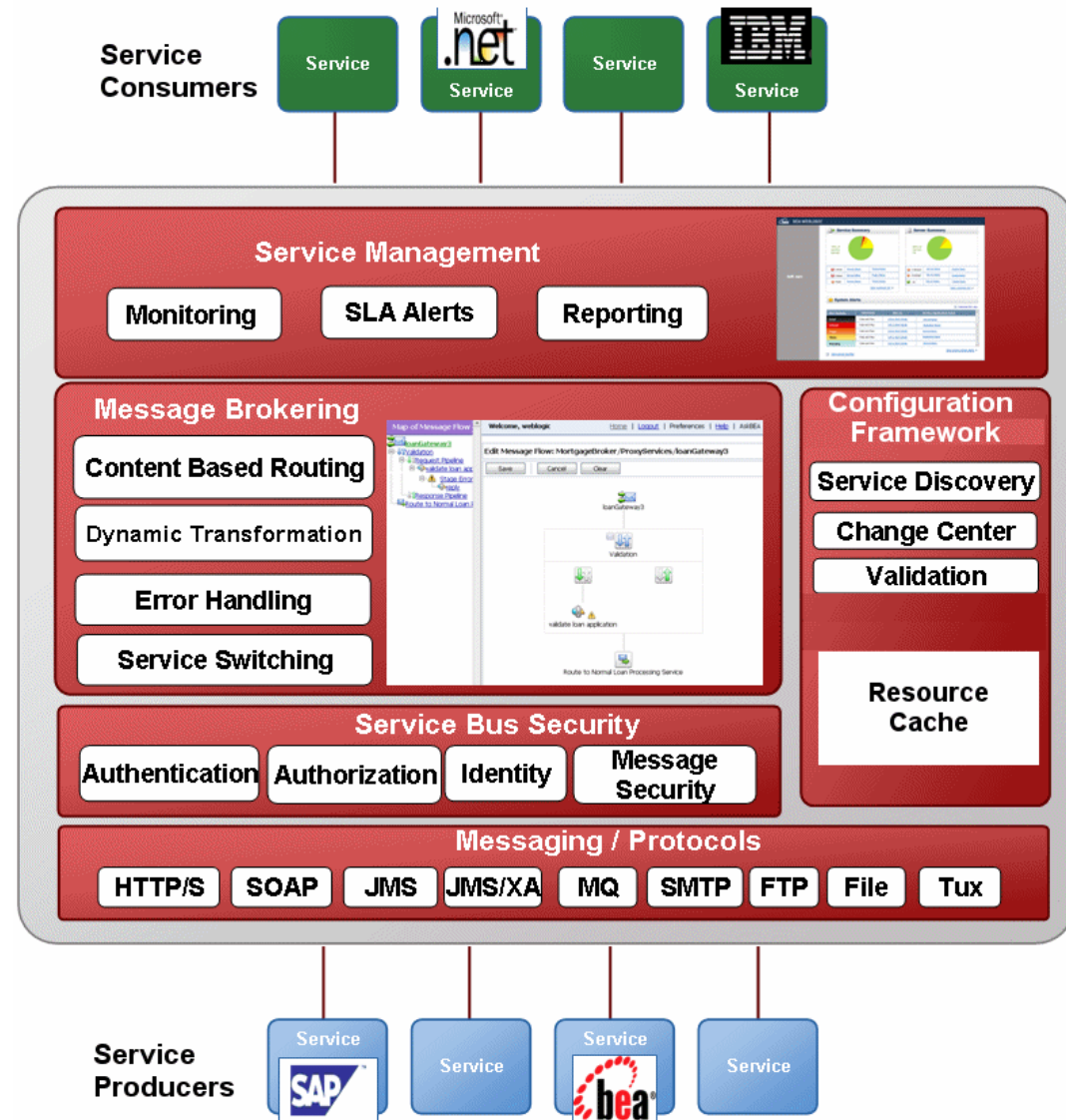
Message Brokering
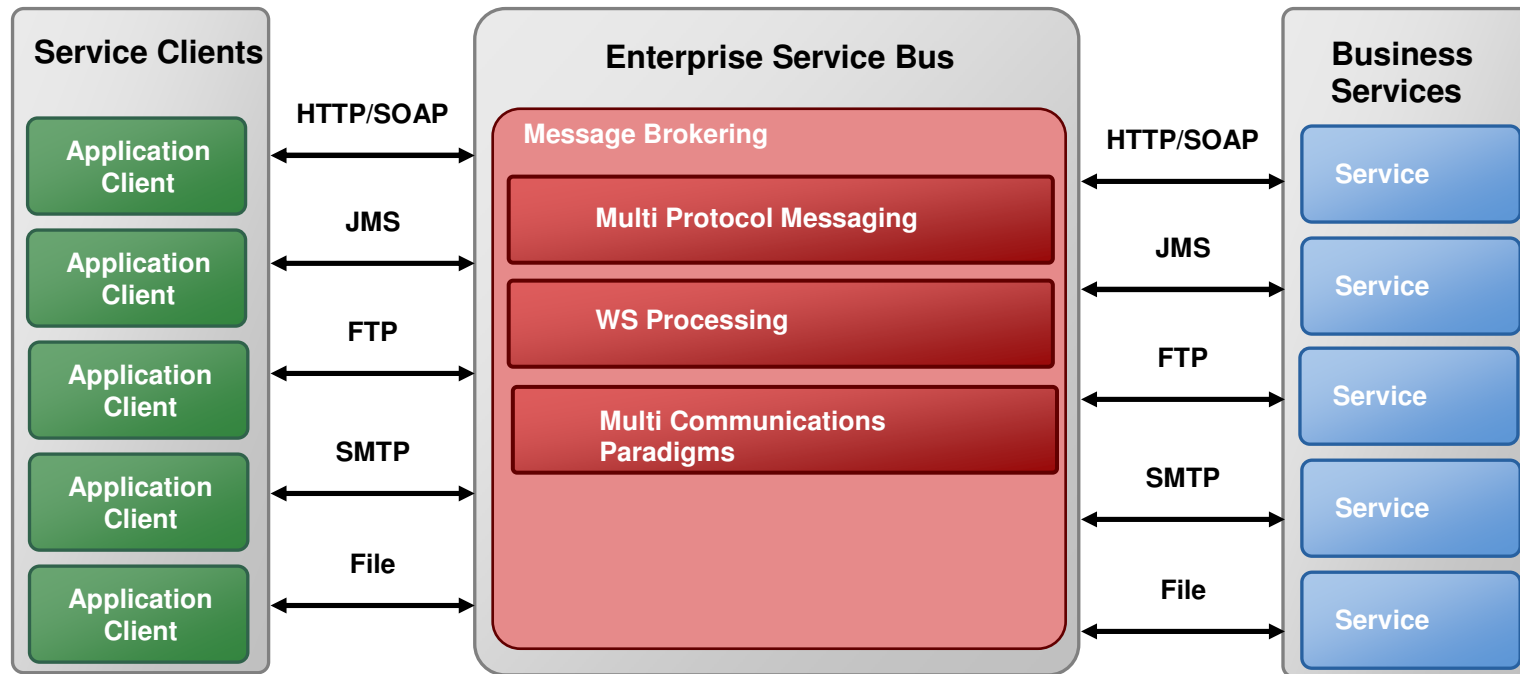
- **Multi-protocol messaging**
- **WS Processing**
  - Full support for "proper" Web Services (WSDL, SOAP enveloping) and non-SOAP-enveloped messages
- **Multiple communications paradigms**
  - Request/response (synchronous and asynchronous), asynchronous messaging, one-to-many publish
 - Mix-and-match (e.g. sync-to-async bridging)

Dynamic Message Routing

Service Clients	Enterprise Service Bus	Business Services
Application Client	**Proxy Flow** — **Route**	**Service**

HTTP/SOAP ⟷ ⟷ HTTP/SOAP

- Dynamic message routing
  - ➤ Routing Based on SOAP headers, transport headers, JMS user properties, MQ headers, file directory, email subject, and message content (XML and structured non-XML)
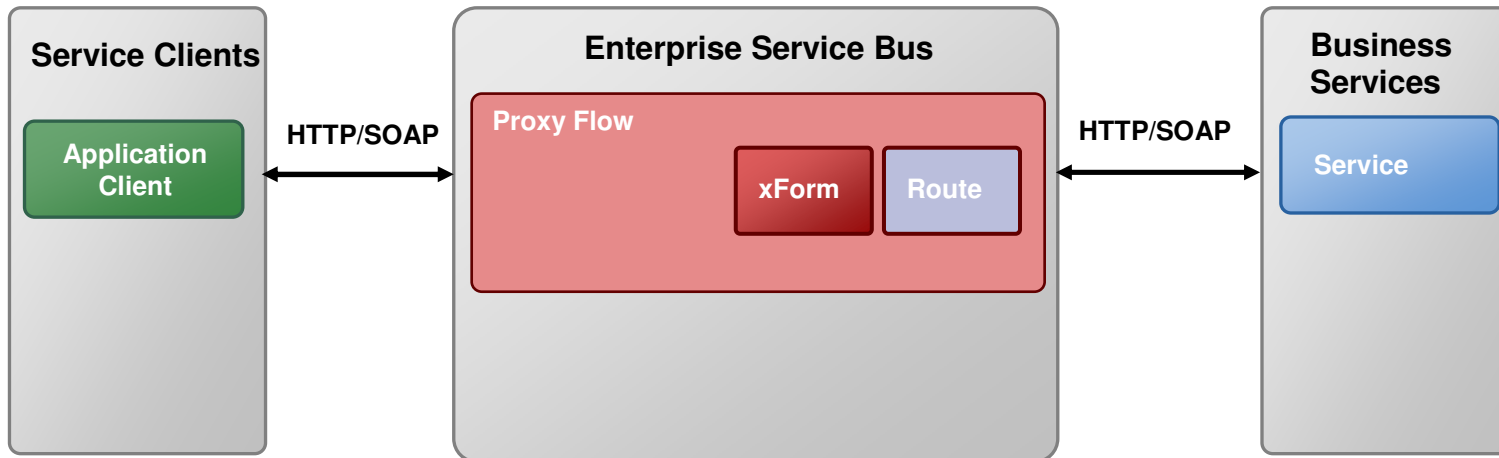
$header

$body

$attachment
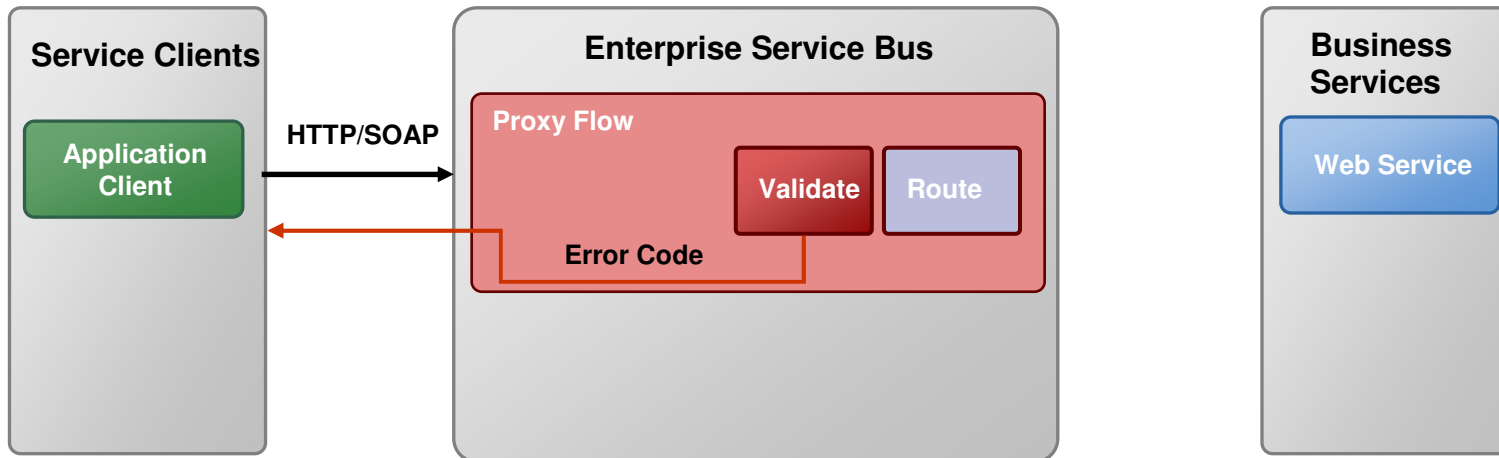
$fault

Service Bus Message Flow
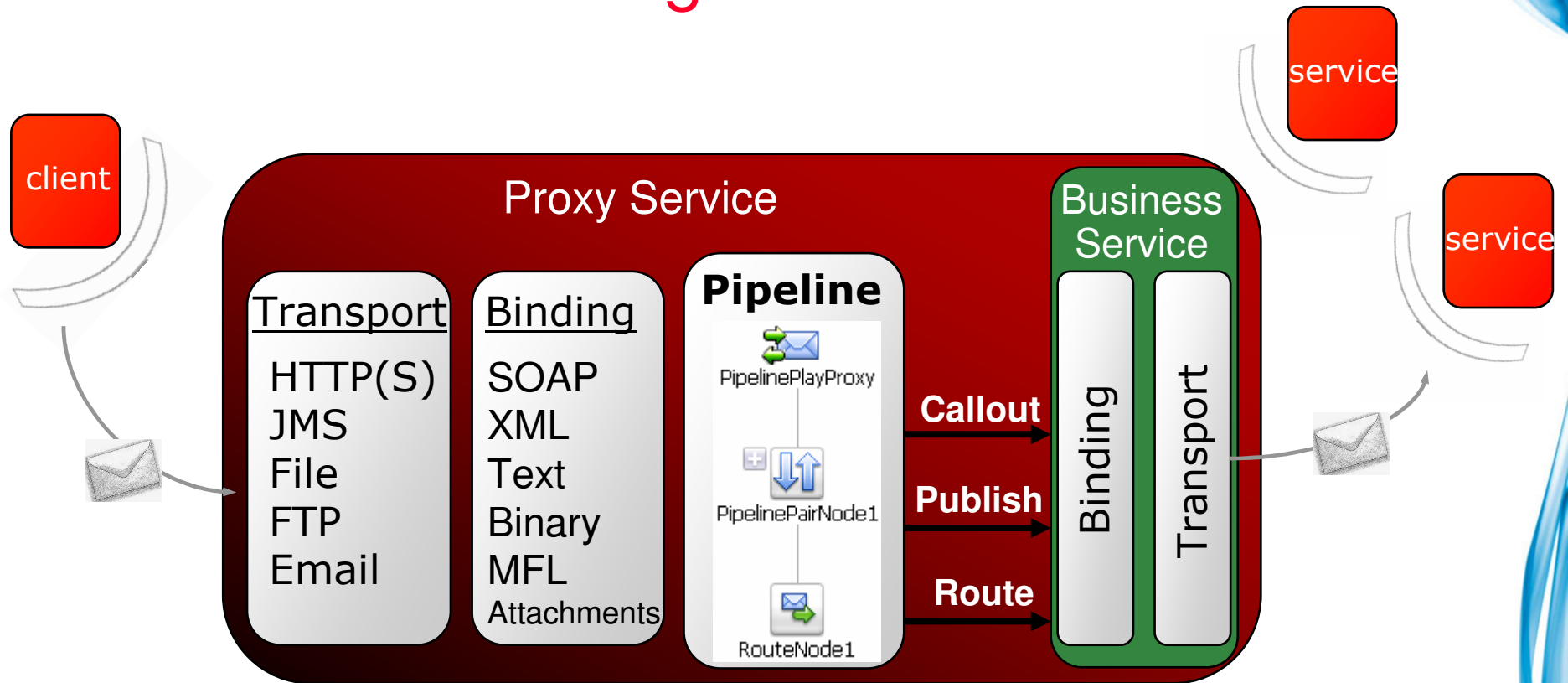
- Pipeline is independent of transport/binding
- Outbound is independent of inbound and vice versa
- Streaming interface for message between transport and pipeline
  - Large messages (e.g. 100 MB) aren't materialized in memory unless needed by pipeline (e.g. XQuery on it)

Integrated Transformation

- **Support for XML and non-XML messages**
  - XML processing optimized with XQuery engine
  - XML transformation uses XQuery; XUpdate like actions used to update header or content
  - Non-XML messages handled via MFL (Message Format Language). Non-XML messages always look like XML to the message flow actions (auto conversions in place)
- **Format handling tools**
  - Schema-driven console tools for navigating message content/headers, creating routing rules and publish filters
- **Format definition and transformation**
  - XQuery maps and MFL definitions are deployed into Service Bus
  - XSLT transformations are supported
- **Transformation call-outs**
  - Transformations (Java) hosted outside the service bus invoked via the call-out action

Error Handling

- **Error handling Features**
  - Capture multiple levels of exceptions
  - Process SOAP Faults
  - Enrich custom error messages to invoking services
  - Search / Report by error conditions
  - Validation
  - Security: authentication, authorization, signing, decryption
  - Can raise error on any condition
- **Error Handlers**
  - Service level
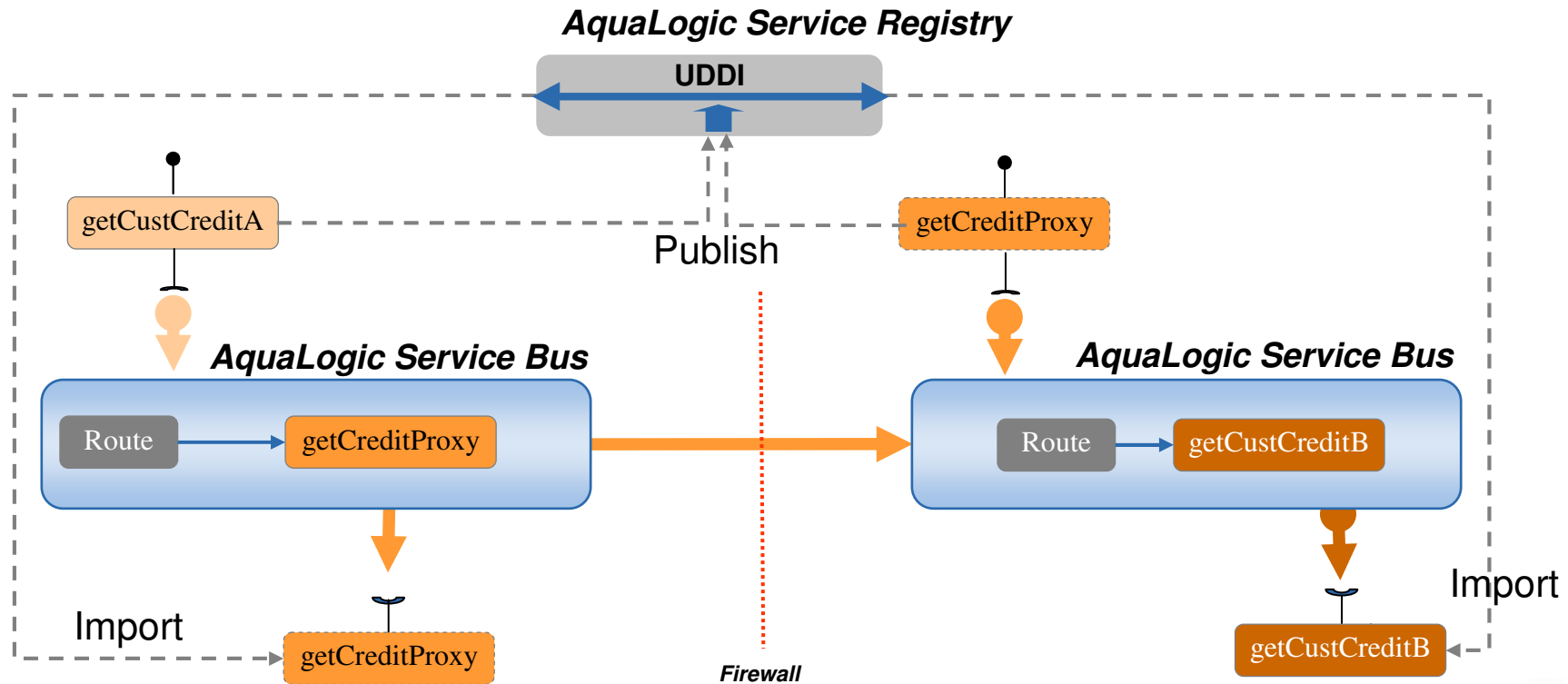 - Pipeline level
  - Stage level

Service Discovery
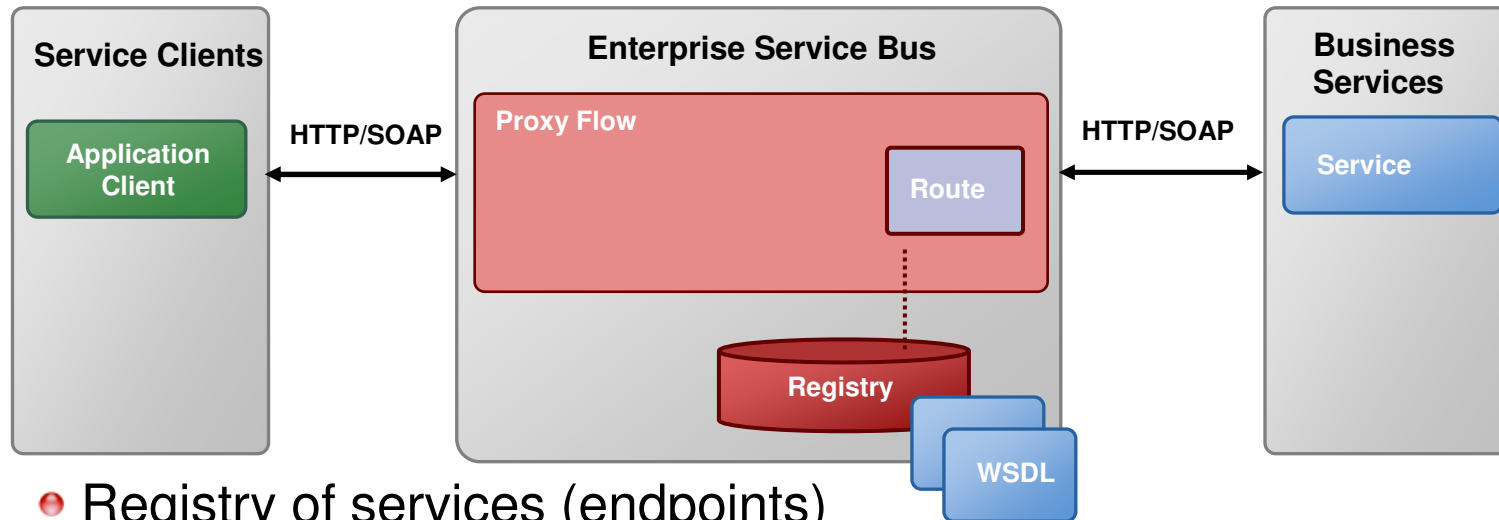
- Registry of services (endpoints)
  - Services/endpoints mediated by Service Bus
  - Service proxies exposed by Service Bus
- Store of service metadata
  - Transport Interface details (URIs/WSDL/schemas/WSPolicies),
  - Transport characteristics (load balancing, retries, failover, timeouts)
  - Associated service providers
  - Transformation maps used by services.
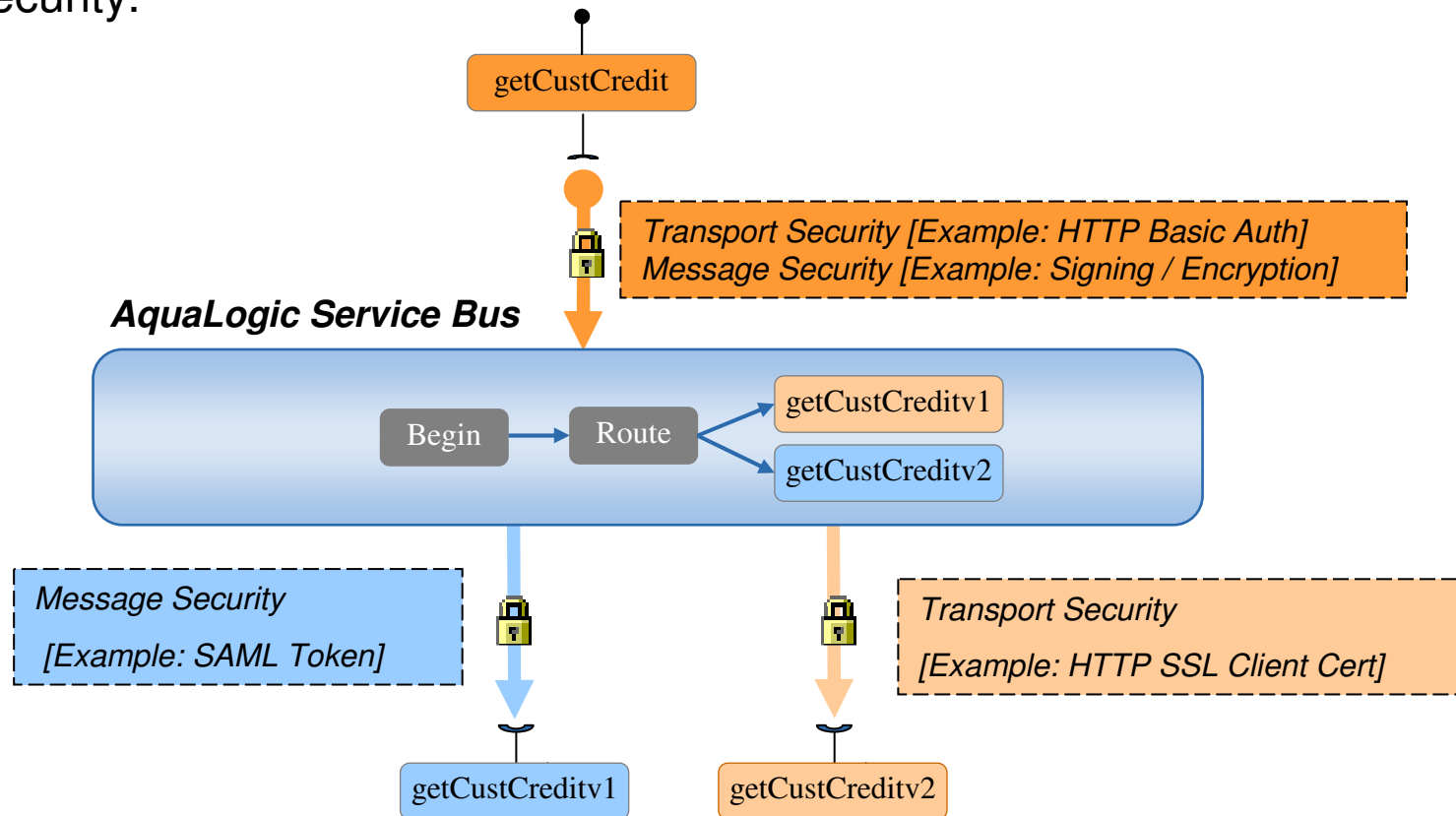- Configuration data migration between environments (i.e. dev to test)

Service Discovery

- ESBs can discover services across the enterprise either locally or in a distributed model leveraging UDDI
- Publish Proxies to UDDI, import business services from UDDI

AquaLogic Service Registry

UDDI

getCustCreditA

getCreditProxy

Publish

AquaLogic Service Bus

Route → getCreditProxy

AquaLogic Service Bus

Route → getCustCreditB

Import

getCreditProxy
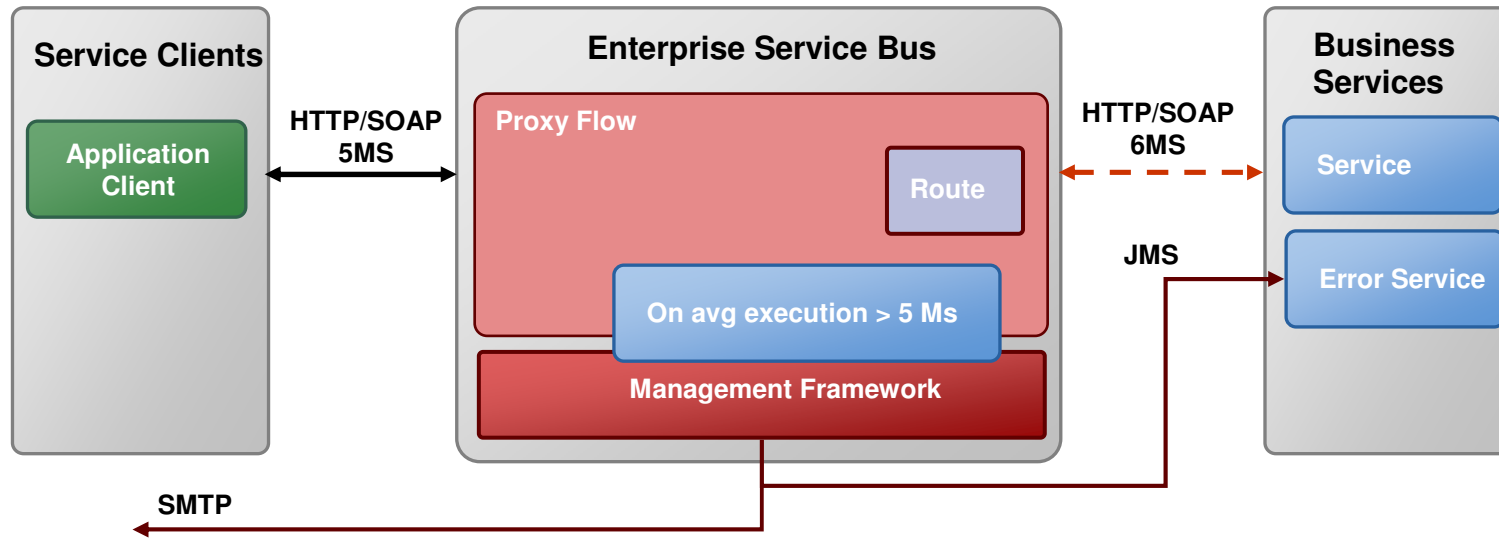
Firewall

Import

getCustCreditB

SOA Security Models

- ESB requires Loose coupling of security models, Identity propagation, and WS-Security (Message Security) or Transport Security.

getCustCredit

Transport Security [Example: HTTP Basic Auth]
Message Security [Example: Signing / Encryption]

AquaLogic Service Bus

Begin → Route → getCustCreditv1
               → getCustCreditv2

Message Security
[Example: SAML Token]

Transport Security
[Example: HTTP SSL Client Cert]

getCustCreditv1

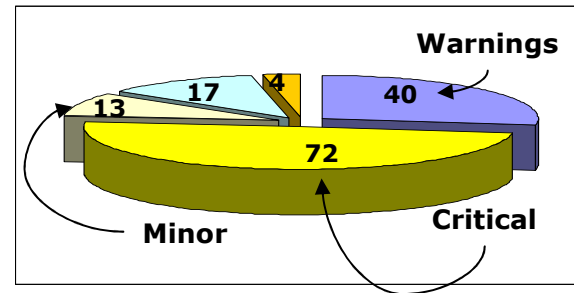getCustCreditv2

bea Think liquid.

SLA Capabilities

- **Establish thresholds based on performance and error events**
  - elapsed times
  - average execution times
  - # of errors
  - success ratio
  - Alerts by severity
- **Flexible notification**
  - Email, JMS

ESB Service Monitoring

- Monitor System Operations
  - Alerting and reporting key monitoring points
  - Gauge system health, slowdown notification
  - Monitoring is optional per service
- Custom Alerts
  - Alert framework allows defining the conditions that need attention
- Dashboard
  - Show fault and performance metrics aggregated cluster wide or per server
- Service health
  - # of Alerts by Severity
  - Configurable Aggregation Intervals

Warnings 40
17
4
13
72
Minor
Critical

Error Responses
- # of Generated Errors
  - By Service

- Service metrics
  - Response times
  - Message or error counts
- For pipelines or routing nodes
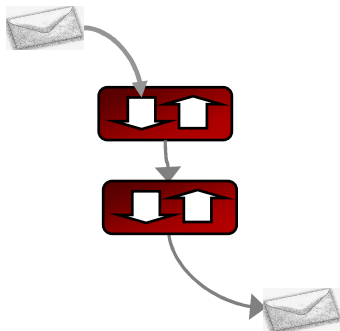 - Transit times
  - Message or error counts

Service Monitoring Summary

Search Name: Path:
- Has Alerts
- Has Errors
- Invoked by proxy: Name:

Search   View All

Name	Path	Avg. Execution Time (in msecs)	# of Messages	# of Errors	# of Alerts
creditRatingService	Folder:Mortgage Broker:Business Services	0	0	0	0
loanGateway1	Folder:Mortgage Broker:Proxy Service	0	0	0	0
loanGateway2	Folder:Mortgage Broker:Proxy Service	0	0	0	0
loanGateway3	Folder:Mortgage Broker:Proxy Service	0	0	0	0
loanSaleProcessor	Folder:Mortgage Broker:Business Services	0	0	0	0
managerLoanReviewService	Folder:Mortgage Broker:Business Services	0	0	0	0
normalLoanProcessor	Folder:Mortgage Broker:Business Services	0	0	0	0

bea
Think liquid.™

Distributed deployments
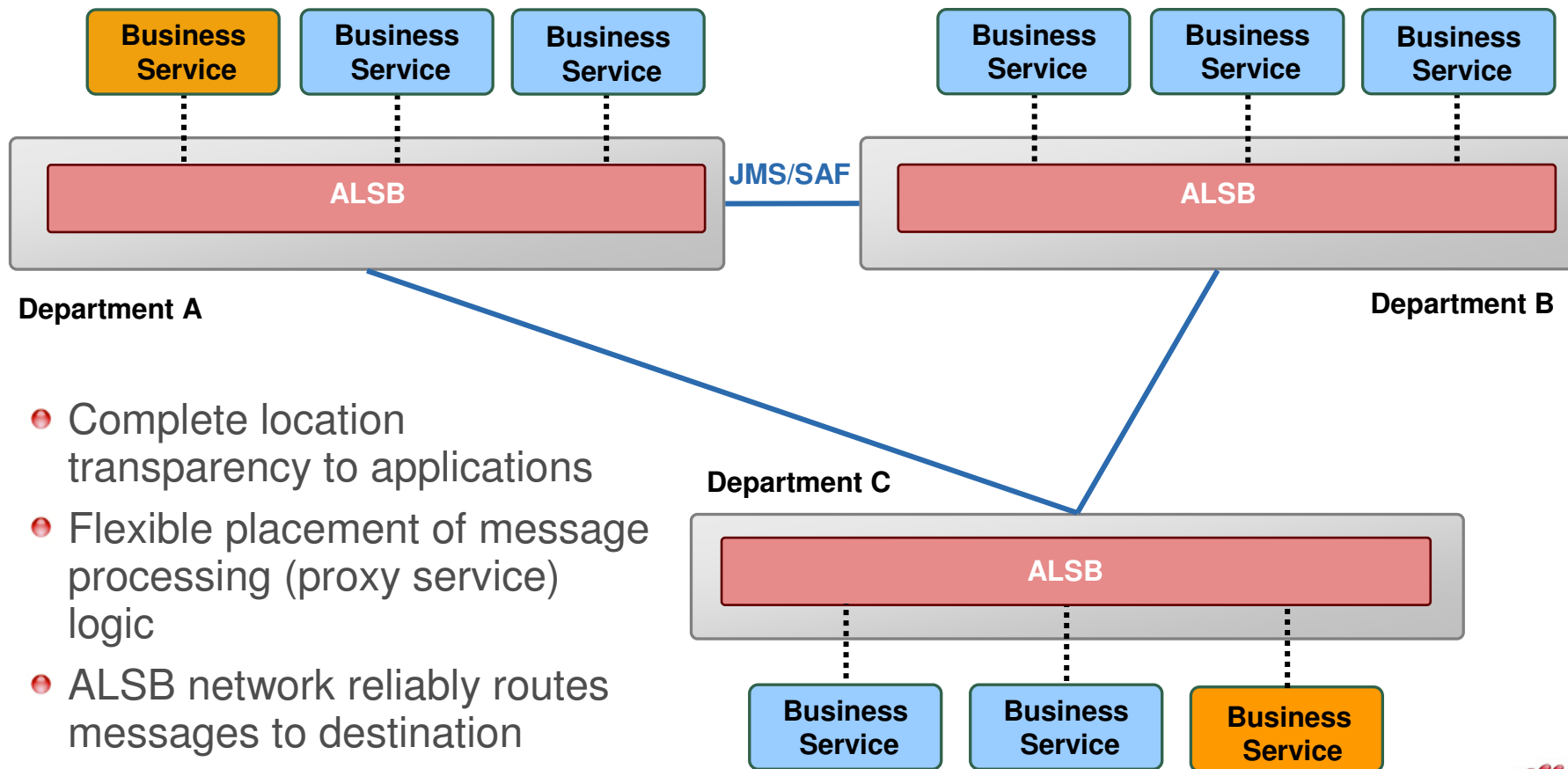
- Complete location transparency to applications
- Flexible placement of message processing (proxy service) logic
- ALSB network reliably routes messages to destination

Reference Architecture