




*ISV and Developer Relations*

## ESB Implementation Options from IBM

## Topics

- Introduce the concepts of a Service Oriented Architecture (SOA)
- Introduce the Enterprise Service Bus architectural pattern
-  ■ Evaluate the appropriate application of several ESB implementations
- Review a methodology for designing an SOA using an ESB
- Introduce the concepts of SOA governance and the role of a service registry and repository

## Topic Agenda

- Product selection criteria
- Product selection
- Selecting multiple products -- Federated ESB scenarios

# Product Selection Criteria

## Features which are Common Across all (or almost all) ESBs are not key decision criteria

- Communication Protocols & Interaction Patterns
  - SOAP, HTTP(S), synchronous, one-way, pub/sub, JMS (for Java ESBs)
- Message Models & Meta-models
  - XML, text, binary, XSD, WSDL
- Mediation Flows and Mediation Patterns
  - XSLT transformation, logging, DB enrichment, protocol transformation
  - Filtering, basic routing, context-based routing, exception handling
- Qualities of Service
  - Basic failover, basic scalability
- Additional Features
  - Graphical tool environment
  - Endpoint lookup in a Registry
  - Ability to be monitored
  - Basic runtime configuration change capability

## Top Questions which Guide ESB Product Selection ..But will vary for each situation

- Does the ESB product support for my messaging backbone (e.g. MQ, non-IBM)?
- How can it connect to and how easy is it to integrate my EIS and legacy systems (e.g. Adapter to SAP, CICS)?
- To what degree does the ESB supports my industry schema (e.g. HL7)?
- Does the ESB support WS \* standards which are needed immediately? How well does the product roadmap for WS\* match feature requirements?
- How many of my mediation requirements are fulfilled by prebuilt mediations?
  - Including custom mediation and extensibility capabilities, can it meet all the functional requirements?
- Can it meet my throughput and latency requirements? In combination with assured delivery?
- Does the ESB meet my transaction and reliability requirements?
- To what degree does it support my security standards and infrastructure?
- Is an SOA Appliance (combined hardware and software) an option?
- What is the software license cost? For development, for initial production, to scale up as demand grows?
- How well does the ESB match the skills of my organization? What is the skill gap?

## Key Criteria for ESB selection

- Communication Protocols & Interaction Patterns
  - e.g., MQ, SOAP/HTTP, pub/sub
  - “Associated” standards (e.g., WS-Security, headers)
  - Adapters
  
- Message Models & Meta-models
  - e.g., XSD, specific XML schema
  - “Associated standards” (e.g., SOAP headers, attachments)
  - “Optimizing capabilities” (e.g., Weak-typing, validation)
  - Adapters
  
- Mediation Flows and Mediation Patterns
  - Metadata driven
    - Determines level of dynamic behavior at run time
    - Registry access

## Key Criteria for ESB selection (con'd)

- Mediation Flows and Mediation Patterns (con'd)
  - Service virtualization
    - Conversion (protocols, interaction patterns)
    - Transformation (use of message meta-models, enrichment, filtering)
    - Routing (simple, customization capabilities)
  - Aspect oriented connectivity
    - Logging, monitoring, auditing integration
      - Database, CEI, file, ...
    - Management integration
    - Security integration
  - Breadth of pre-built mediation primitives
    - Custom mediation capability and Programming model
    - Weak-typing
  - Support for pre-built mediation flows (templates)



## Key Criteria for ESB selection (con'd)

- Qualities of service
  - Heterogeneous transaction coordination
  - Reliable/assured delivery
  - Performance
    - Message size
    - Throughput
  - Scalability
  - Reliability
  - Availability

## Key Criteria for ESB selection (cont.)

- Non-functional
  - Affinity to SOA environment (e.g., WebSphere Process Server)
  - Affinity to IT environment (e.g. JEE application server)
  - Development tooling capabilities and affinity to current tools
  - Configuration and administration tooling capabilities
  - Existing and required skill set (e.g., JEE skills)
  - Product maturity and comfort level with leading edge products
  - Price and total cost of ownership
- Ease of integration with
  - Monitoring and management infrastructure
  - Security infrastructure

# Product Selection

# ESB offerings from IBM WebSphere

*WebSphere delivers the most complete ESB solution*



**WebSphere ESB**  
*Built on WebSphere  
Application Server for an  
integrated SOA platform*



**WebSphere  
Message Broker**  
*Universal connectivity  
and transformation in  
heterogeneous  
IT environments*

**ESB  
Offerings from  
IBM WebSphere**



**WebSphere DataPower  
Integration Appliance**  
*Purpose-built hardware ESB  
for simplified deployment and  
hardened security*

## When to use WebSphere ESB?

- You use WebSphere Application Server
  - Your team has skills with WAS Administration and Java coding
- You are now or planning on developing business process using WebSphere Process Server
  - WebSphere ESB and WPS have common tooling, programming model, and runtime
- You are integrating with ISV business applications hosted on WAS or 3<sup>rd</sup> party solutions which extend and support WAS
- All connectivity requirements are standards based interactions using XML, SOAP, and WS\*
- You want to mediate between Web services and existing systems using JMS and WebSphere JCA Adapters
- Reliability and extensive transactional support are key requirements
- You want to co-host WebSphere services and ESB in one application server

## When to use WMB?

- You are currently using WebSphere Message Broker but not as an ESB
  - Migrate to newer version; implement ESB Patterns
  - Leverage existing WMB skills
- You have extensive heterogeneous infrastructures, including both standard and non-standards-based applications, protocols, and data formats
  - You have extensive MQ skills and infrastructure
  - You are using Industry formats such as SWIFT, EDI, HL7
- You are implementing a wide range of messaging and integration patterns
  - Complex event processing, message splitting and aggregation
- You need extensive pre-built mediation support
- You have very complex transformation needs
- Reliability and extensive transactional support are key requirements
- To achieve very high-performance with horizontal and vertical scaling

## When to use DataPower XI50?

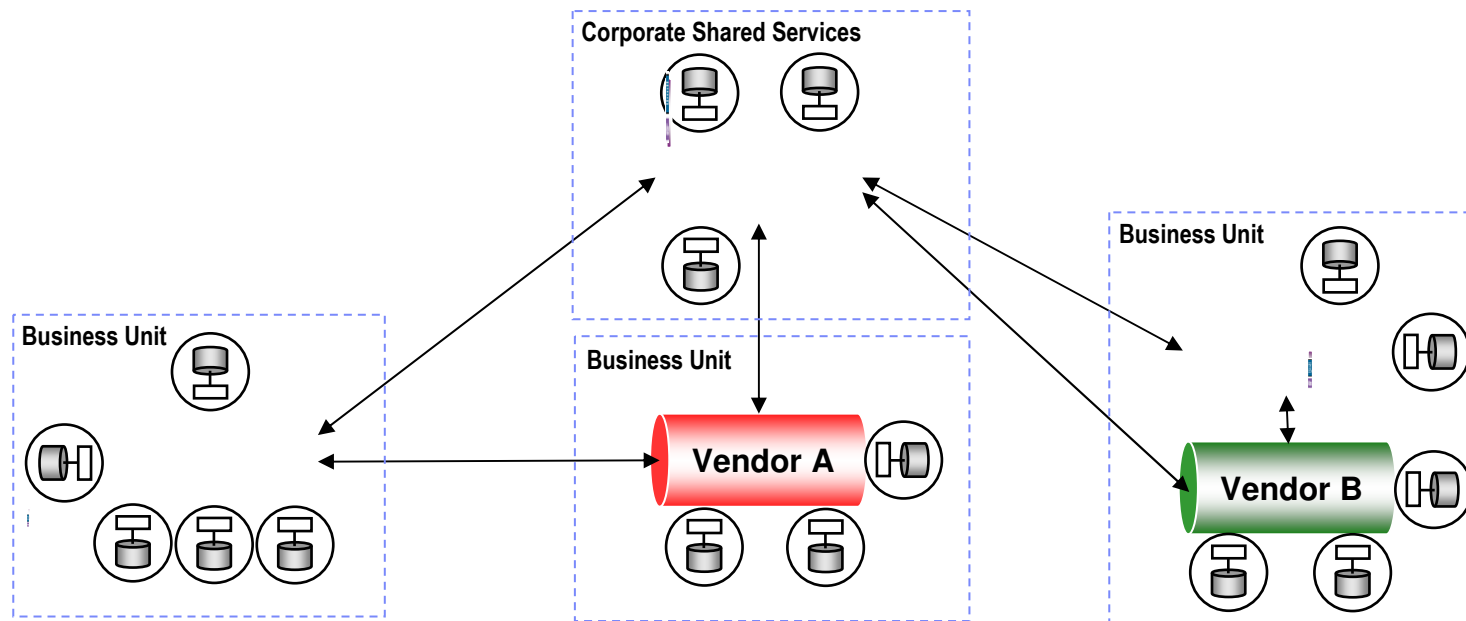
- Ease of use is a pre-dominant consideration
  - Simple experience of drop-in installation and admin-based configuration with no or minimal development required
- You are transforming between XML-and-XML or XML-and-any other format
- Your interaction patterns are relatively simple
- Your mediation requirements are met by the existing DP mediations and minimal extensibility is needed
- You are using XML-based or WS-Security extensively
- You require use of advanced Web services standards
- You need to minimize message latency when adding an ESB layer
- You are doing extensive XML processing combined with high performance reqs
- Your ESB must be in production very quickly

# ESB Federation



## Managing Multiple ESBs – Federated ESB Topologies

➤ ***A single enterprise-wide ESB is rarely attainable – most businesses will have multiple ESBs across business units***



WebSphere Services Registry and Repository

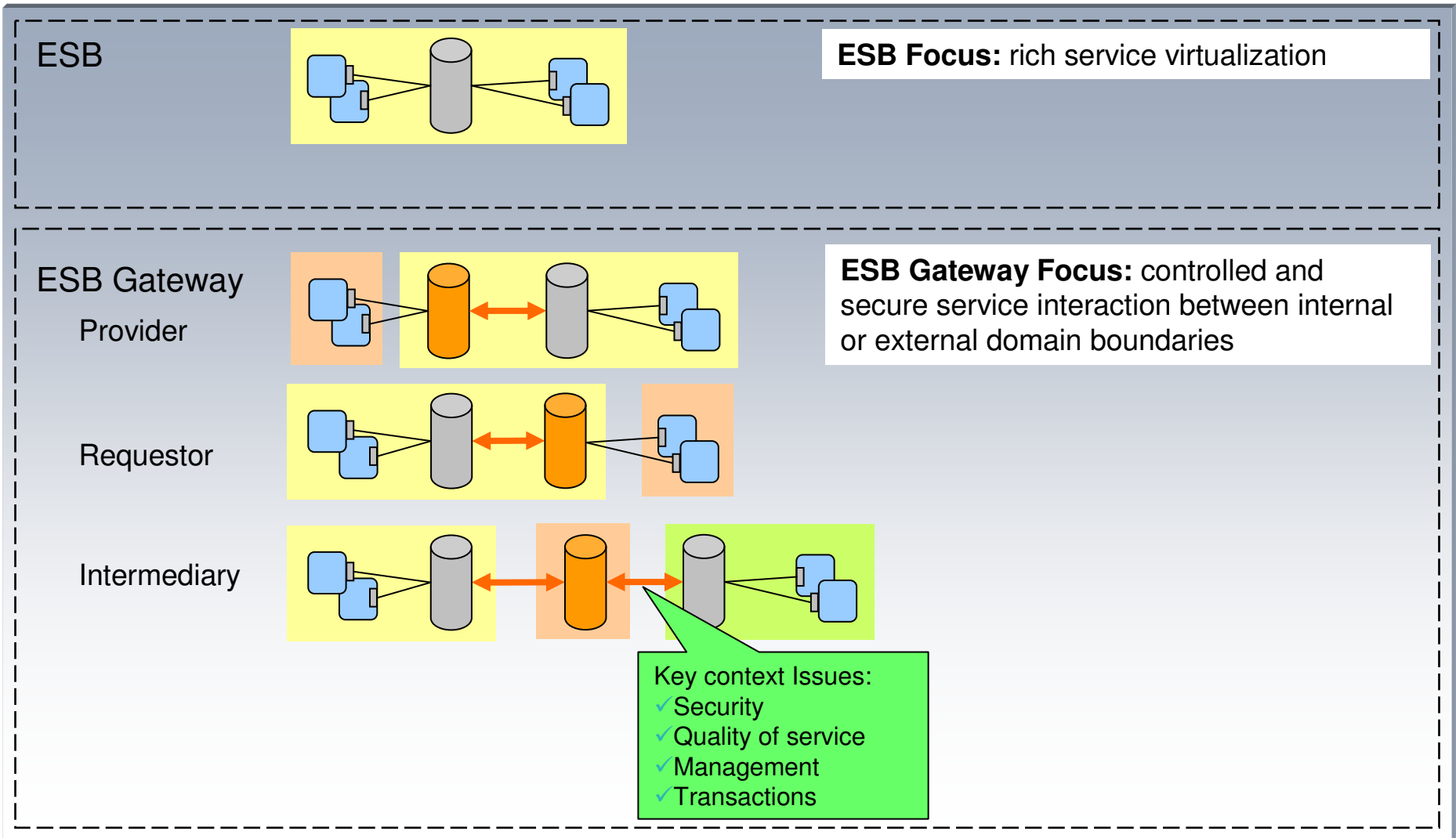
Tivoli Composite Application Management for SOA

Federated Security

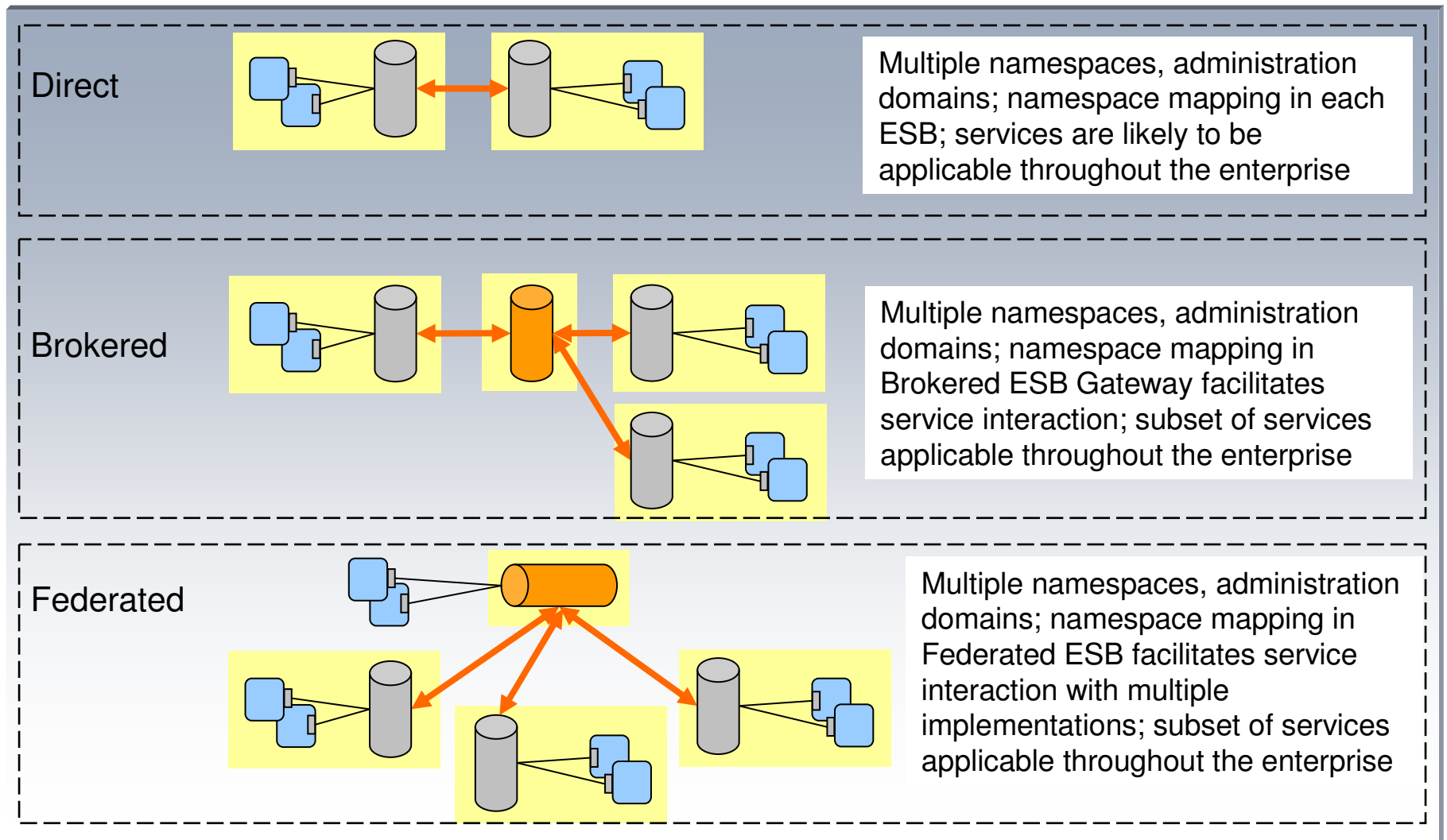
## Why Federated?

- Business and Funding Models are Distributed or Federated
  - E.g., by business unit
  - Best Practice – Architecture aligned to business model
- Distributed geographical location
- Distributed Governance
- 'Domains' have differing ESB requirements which are best met by different products
- Domain service ESBs need to be decoupled to allow asynchronous development and deployment
  - Best Practice – Isolate critical environments
- Backbone ESB is needed for reuse of services and to have Enterprise level services across domains

# ESB Runtime Patterns



# ESB Integration Topology Patterns



## (Federated) ESB

- The Enterprise Service Bus consists of all the domain service buses, and the backbone bus if needed
- Service messages flow across more than one bus
- Majority of service requests to providers are within a domain and are mediated by a single domain service bus
- Some shared services may be **provided** by a domain for use by other domains
- Some consumers in one domain may **request** services provided by another domain
- All service requests between domains usually routed by a 'backbone bus'
- Enterprise services can be available directly on the backbone bus
- Also federated
  - Service registry
  - Security
  - management

# Review

- Product selection criteria
- Product selection
- Selecting multiple products -- Federated ESB scenarios