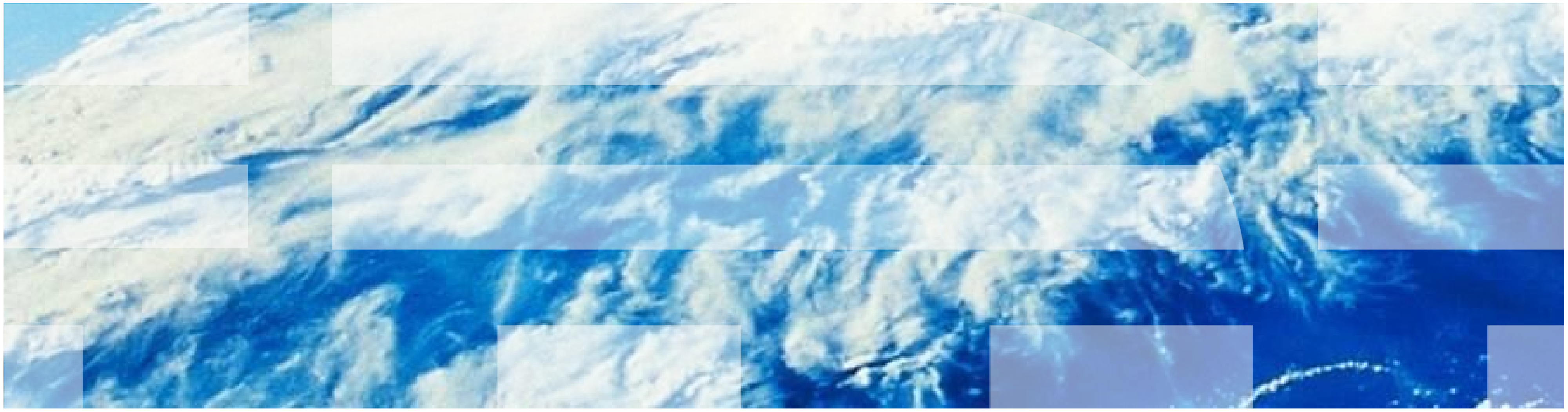
Andy Piper, WebSphere Messaging Community Lead
March 2011

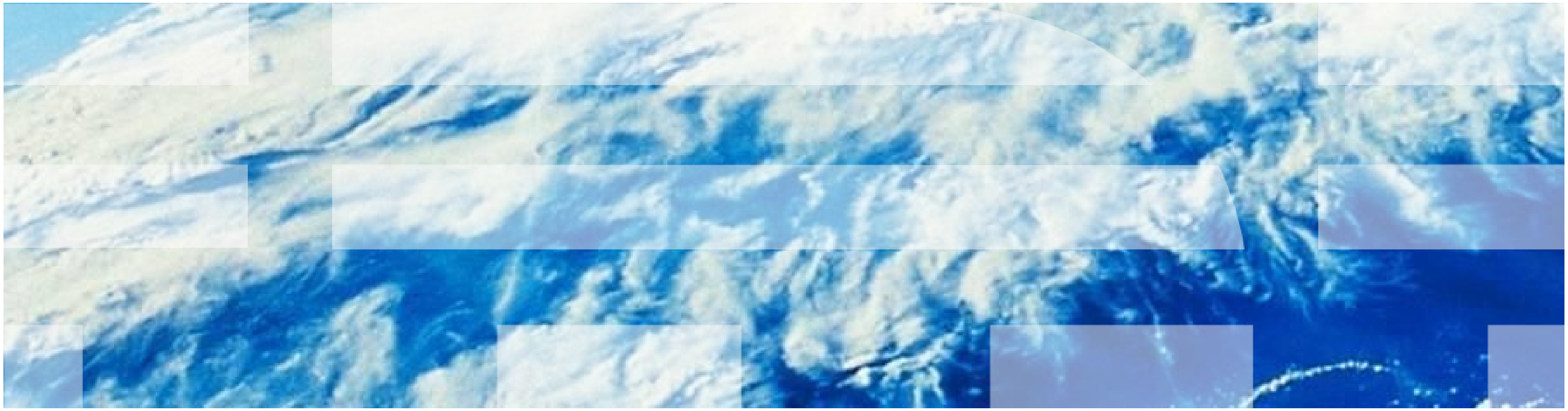
ESB Messaging and Enrichment



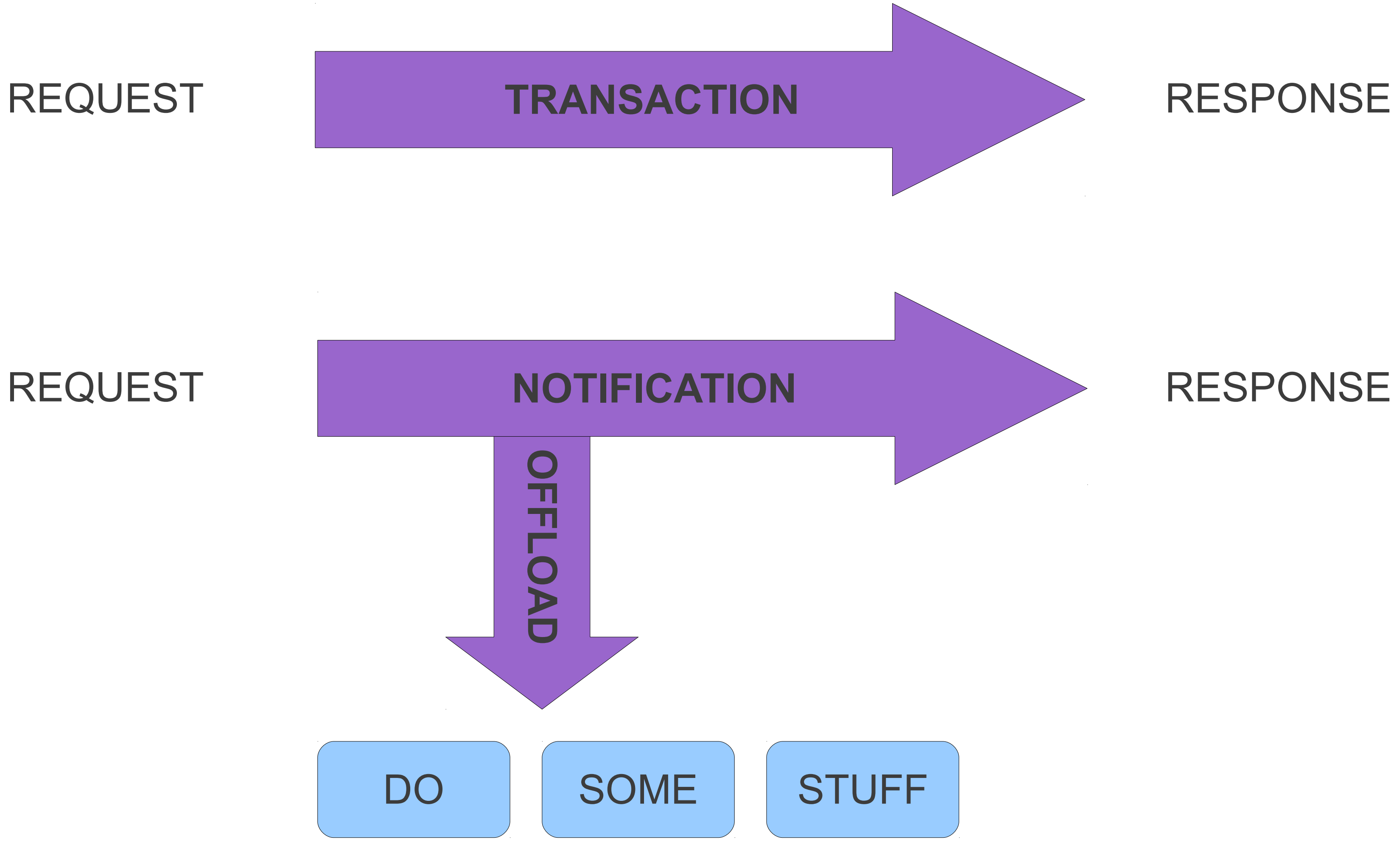
Agenda

- ESB and Connectivity Overview
- Processing Scenarios & Usage Patterns
- Pattern Technology

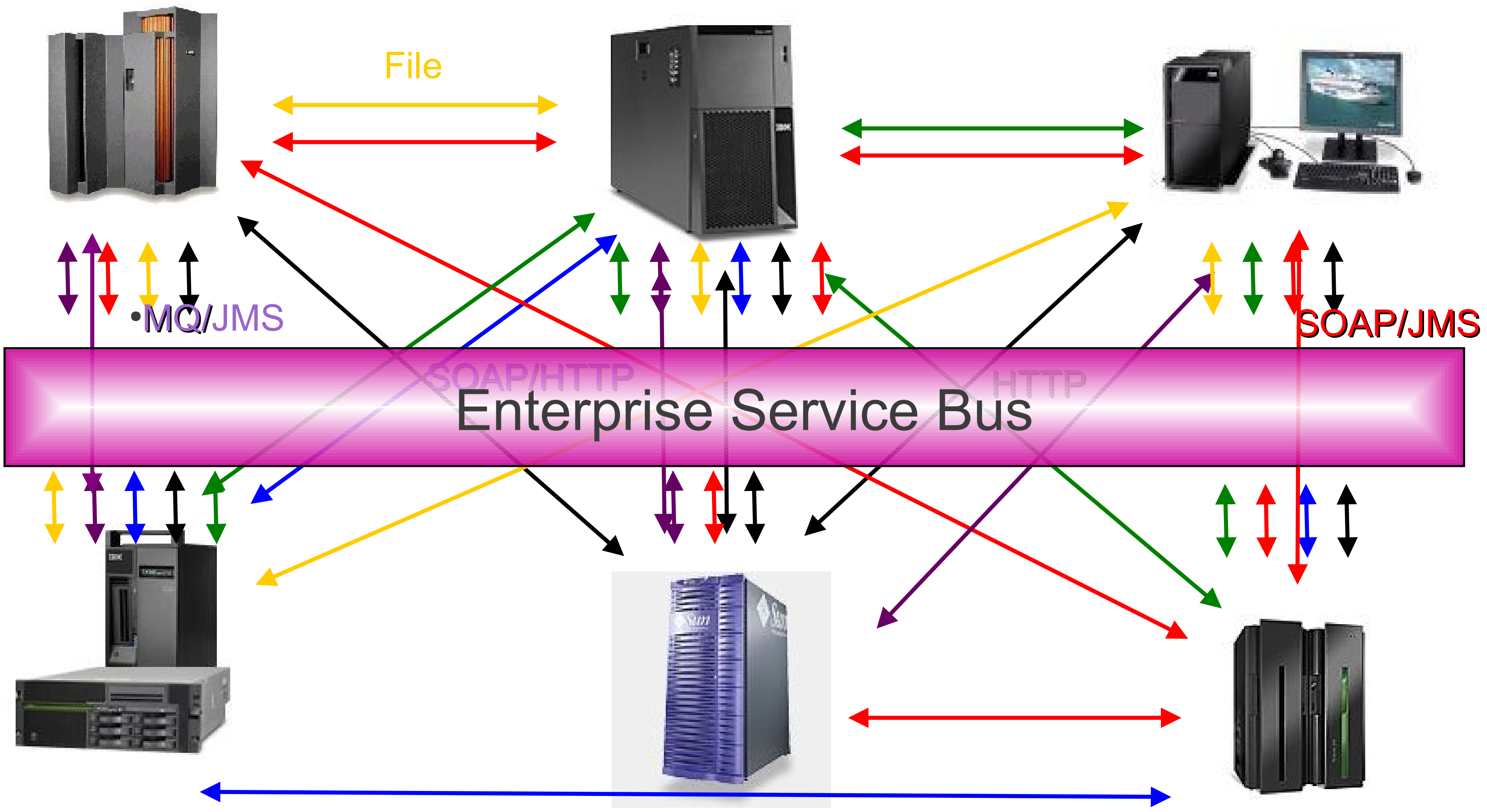
ESB and Connectivity Overview



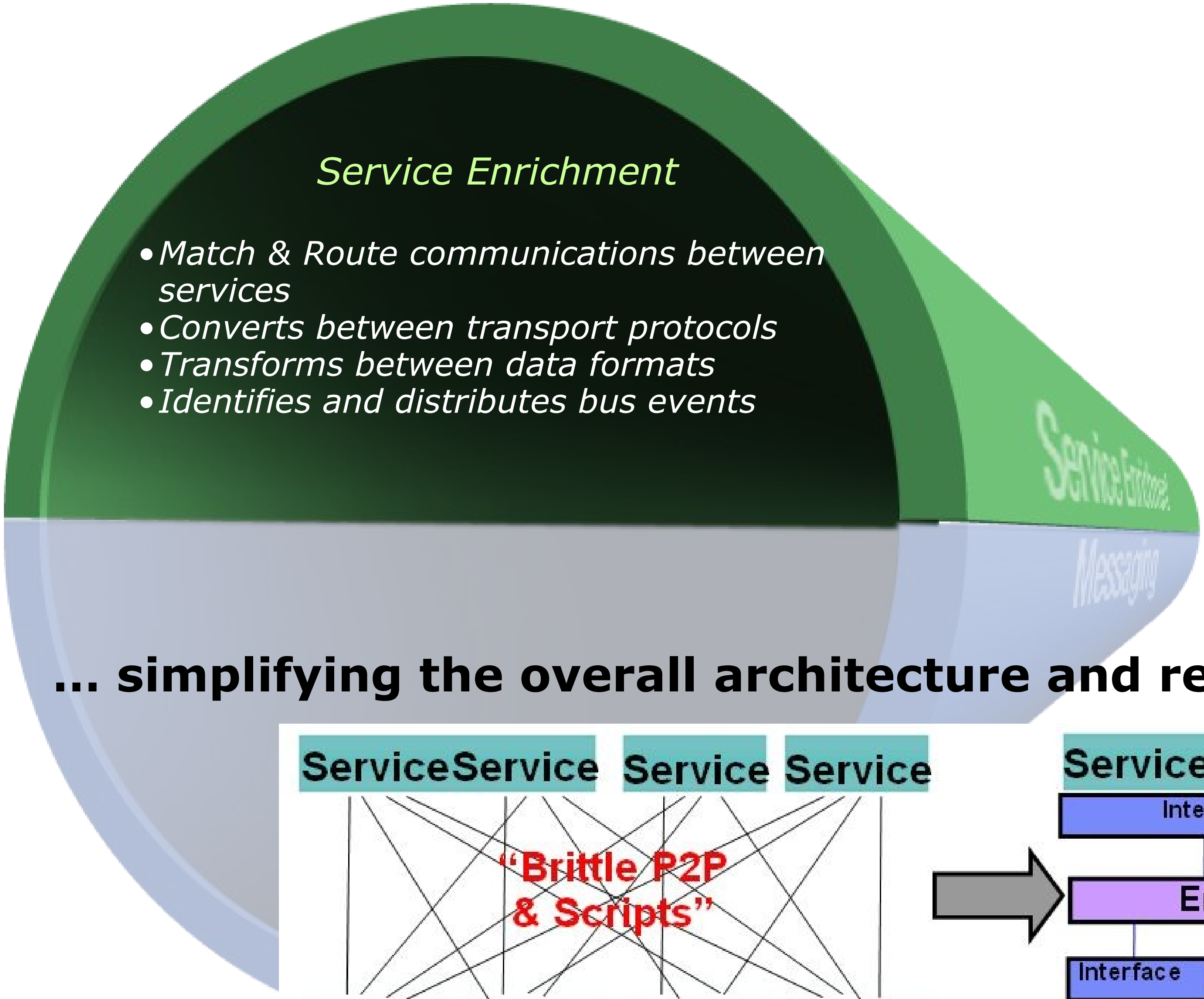
Message-driven architecture basics



ESBs Simplify Connectivity



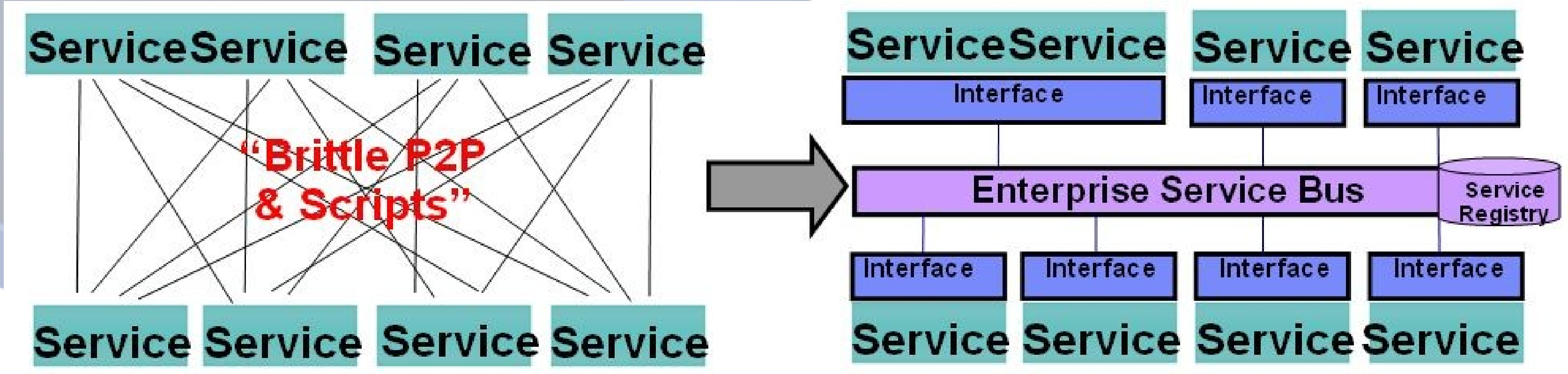
Enriched connectivity for SOA



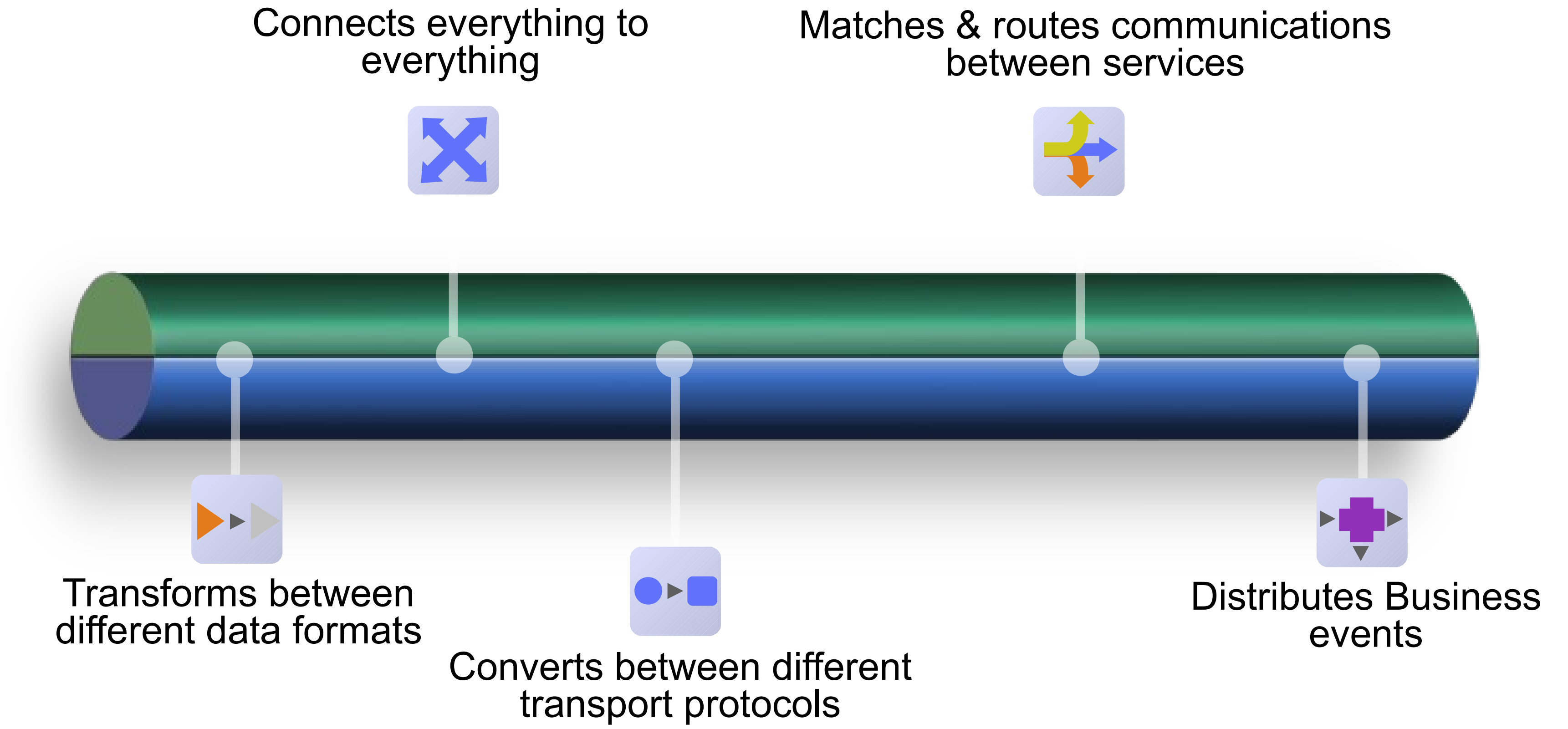
Service Enrichment

- *Match & Route communications between services*
- *Converts between transport protocols*
- *Transforms between data formats*
- *Identifies and distributes bus events*

… simplifying the overall architecture and reducing IT cost



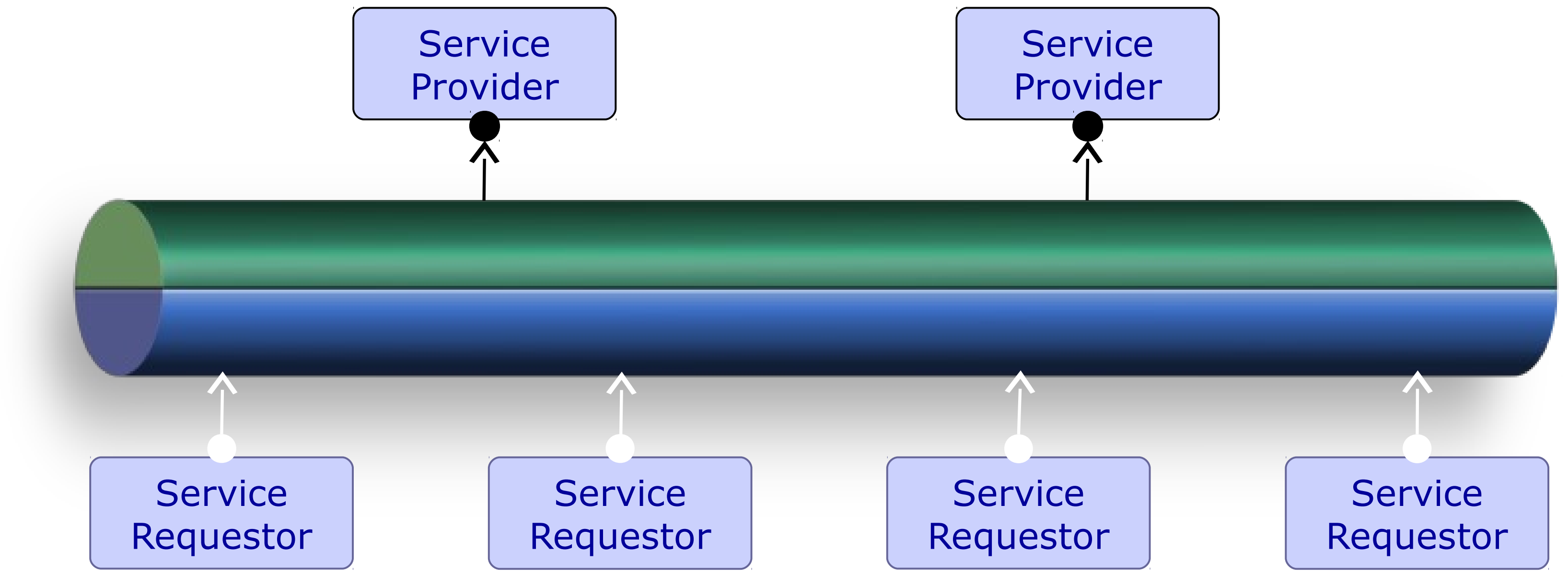
Agile Connectivity


Connects everything to everything
Matches & routes communications between services
Transforms between different data formats
Converts between different transport protocols
Distributes Business events

An ESB enables flexible SOA connectivity for integrating business applications, services and processes

Two core principles enable flexibility

The ESB facilitates the *decoupling of interactions* between requestor(s) and provider(s)



The ESB fulfils *two core principles* in support of *separation of concerns*

Service Virtualization

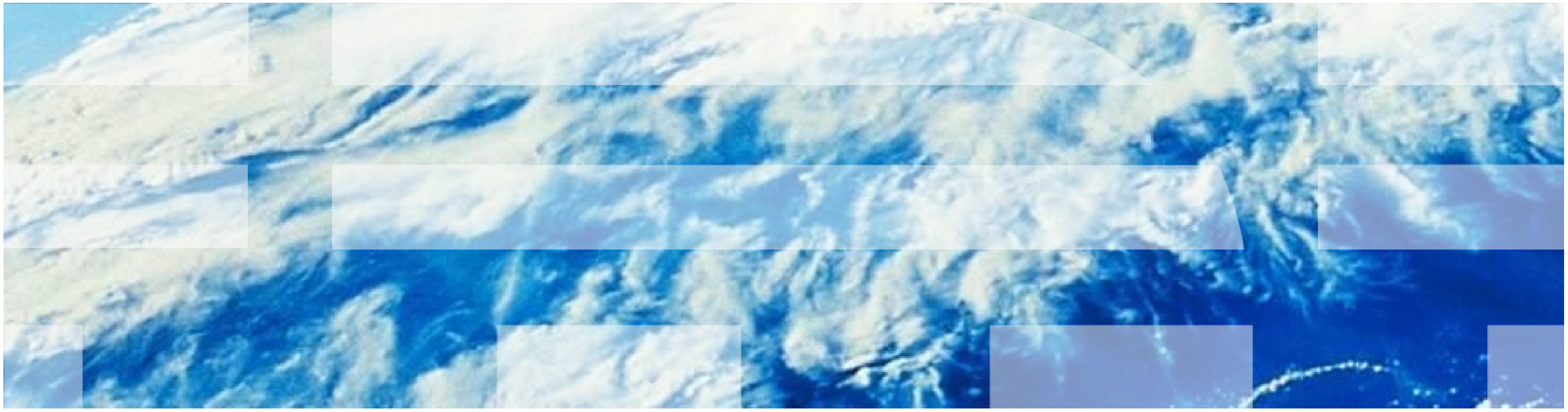
- ★ *Routing*
- ★ *Protocol and transports*
- ★ *Transformation of interfaces*

Aspect Oriented Connectivity

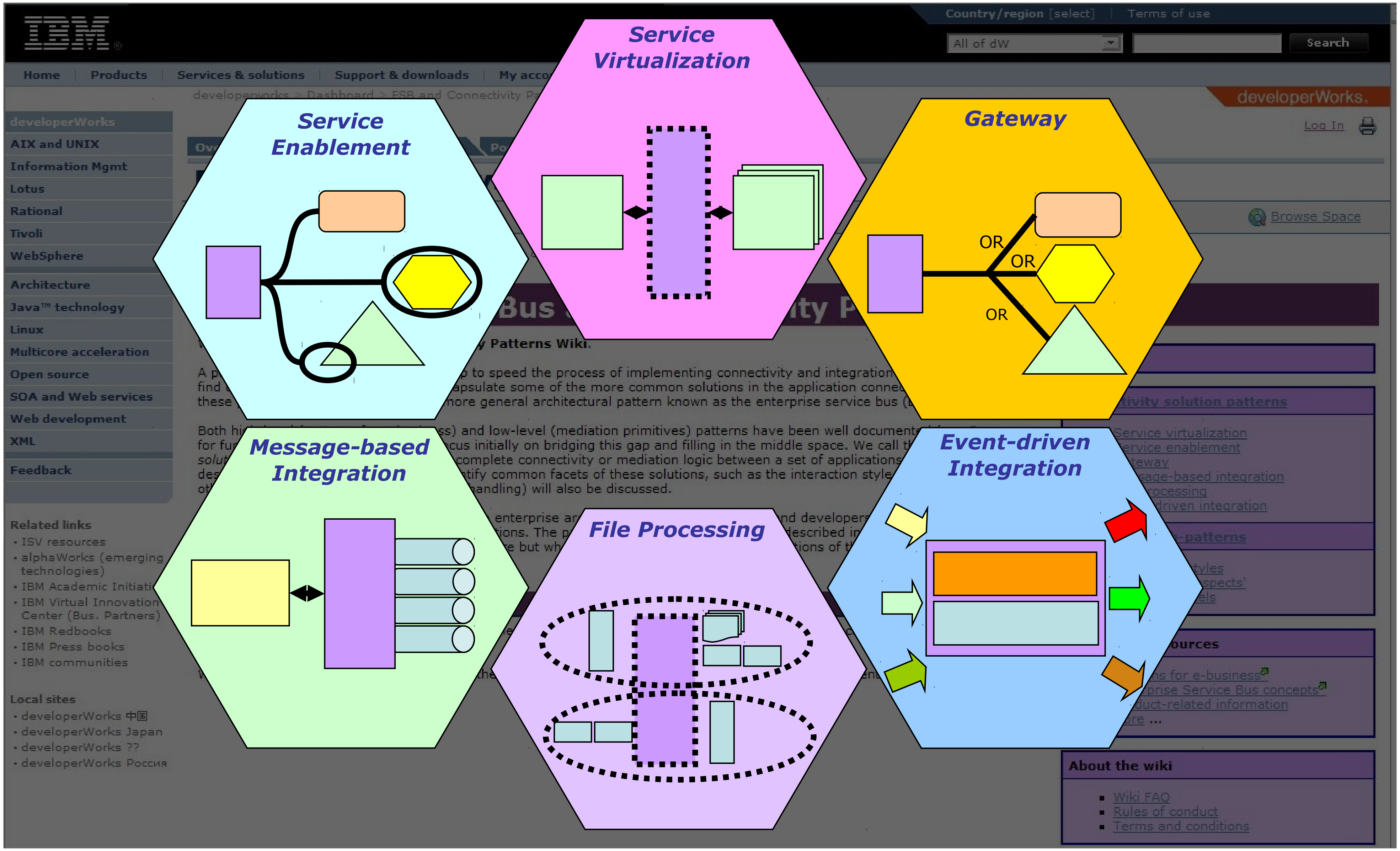
- ★ *Security*
- ★ *Management*
- ★ *Log and Audit*
- ★ *Event tracking*

etc …

Processing Scenarios & Usage Patterns



Many Defined Patterns for ESB-based Solutions



http://www.ibm.com/developerworks/wikis/display/esbpatterns/

Key Scenarios Deliver Business Value

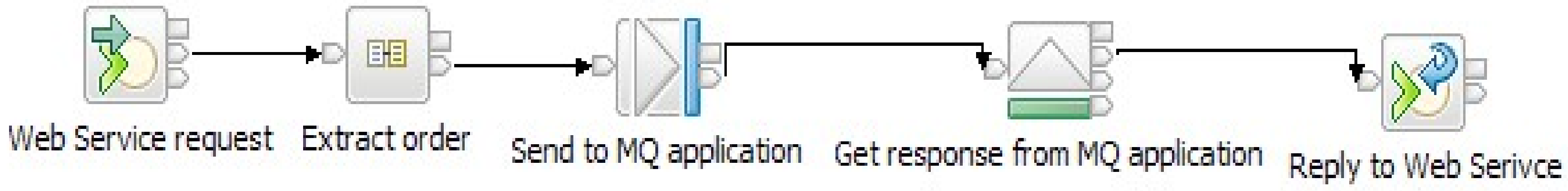
- Extend the Reach of Existing Applications: Multi-channel Processing
- Easily transform batch-oriented file work into online requests
- Get maximum value from Packaged Applications
- Connect Devices to the Enterprise
- Provide a Policy Enforcement Point for secure application connectivity
- Create an Application Inventory and Govern Processing with a Registry
- Apply Business Rules to achieve Smart Connectivity
- Monitor your Business Activity and Act Intelligently
- Initiate and Support Business Processes
- A Flexible Infrastructure to Support Change

Extend the Reach of Existing Applications (1/2)

Provide and Consume Web Services

- **Web services are now established as an interoperability standard**
  - Vitally important from a business to business connectivity perspective
  - Businesses to consume each others' services using these well defined standards
  - Internal standardization between parts of the same organization via Web Services

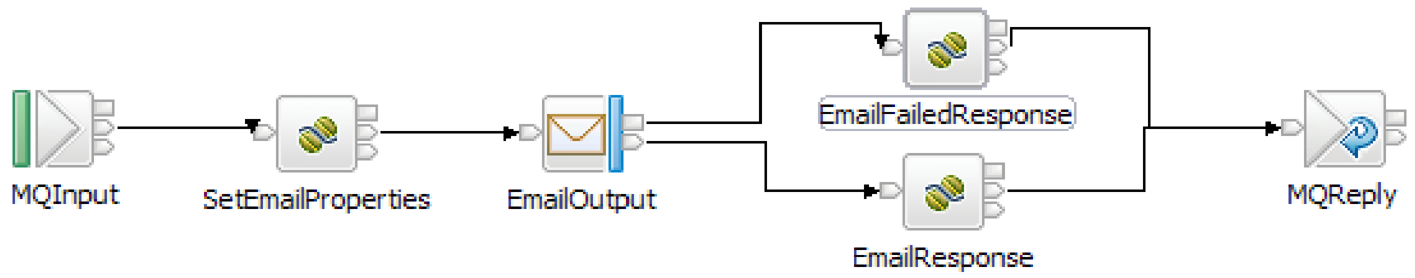
- **Adoption of Web Services by many subsystems is not universal**
  - ESB allows your existing applications to be exposed as web services
  - ESB 'universal translator' converts web service to existing formats and protocols
  - Existing applications can consume web services without change
    - Exploit web services with limited new development skills and platforms



Extend the Reach of Existing Applications (2/2)

MQ enable all your applications

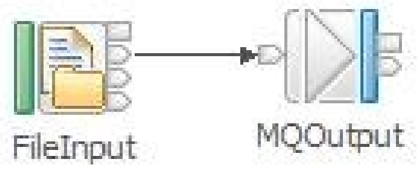
- **ESBs allows you to use MQ technology to the fullest extent**
  - Robust, transactional, reliable, high-performance messaging
  - ESB provides an incredibly broad range of connectivity mechanisms available to MQ
  - Any application can easily connect to the MQ infrastructure inbound or outbound
- **Examples**
  - Transform a TCP/IP based application by allowing it to consume regular MQ messages
  - MQ applications access an external Web Services provided by a Business partner
  - MQ applications access ERP systems such as SAP, SEBL, PeopleSoft…
- **The Goal: Multi-Channel Connectivity**
  - Consuming Services and Applications independent of client implementation
  - Increasingly relevant in world of device proliferation

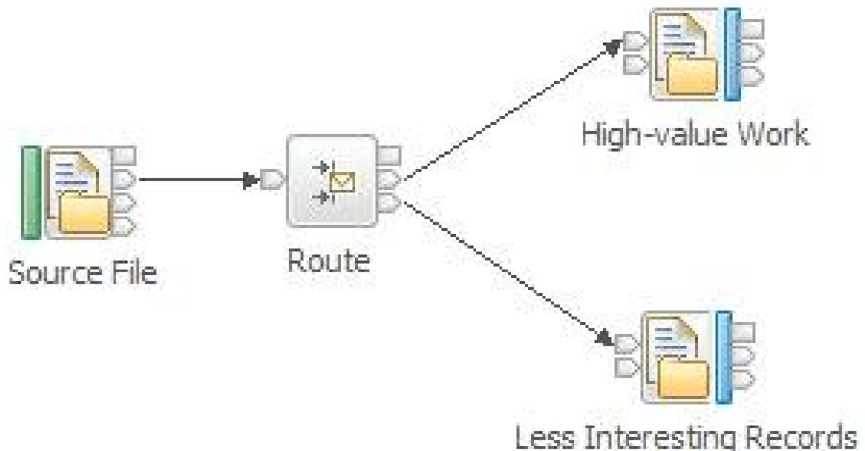


Combine File-based and On-line Processing

Unlock the valuable business data in your files

- **Files exchange between applications still popular and effective**
  - Flexible method of exchange: Neither enterprise has to mandate technology
- **There are legitimate reasons for using files to exchange information**
  - Usually relate to the way businesses run or physical processes occur
- **Examples**
  - A cargo ship has thousands of containers each with hundreds of palettes
  - Reduce unit transaction costs by aggregating numerous clients requests
- **End to End File Movement and File Processing**
  - Reliable and secure delivery File Transfer with **WMQ FTE**
  - File processing allows clients to get file/batch work online, easily





Get maximum value from Packaged Applications

Move information to and from packaged applications

- Packaged applications play a vital role
  - SAP for purchasing, sales, inventory…
  - SEBL for Sales, PeopleSoft for HR
  - Oracle, JDEdwards…
- Interfaces are often non standard: e.g. SAP BAPIs, IDOCs
  - Processing and data are isolated from other applications
  - Result: packaged apps have difficultly using/generating information for other apps
  - Inhibits adoption of a best of breed philosophy
- Support for SAP, SEBL, PeopleSoft, inbound and outbound
  - Adapter components built-in to ESB
  - Drive new work into its packaged application from any other supported source
  - Can send information from packaged application to any other supported target
  - Packaged applications can focus on what they do best **and** be integrated

Connect Devices to the Enterprise

To and from a broad range of devices

Industry Observation
- "How to I get information from everywhere, understand it, and act?"
- Medical, Energy and Utilities, Distribution, Transport, Gaming…
- Issues based e.g. traffic congestion, efficient energy, timely supply…

A Smarter Planet is full of devices
- Data is generated *outside* the enterprise
  - Typically very large numbers of devices
  - Often concentrator technology; differentiate, integrate & forward
- MQTT for standards based device integration
  - Small footprint client, embeddable
  - Lightweight protocol for bandwidth cost (by-the-byte)
  - Fragile network support for hostile environments

Connect Devices, Apply Intelligence
- ESB connects devices to enterprise systems
- Apply intelligence in near real-time
  - Passive and active systems

IBM is working with Brisbane, London, Singapore and Stockholm to deploy smarter traffic systems. Stockholm has seen approximately 20 percent less traffic, a 12 percent drop in emissions and a reported 40,000 additional daily users of public transportation.

Provide a PEP for Secure Application Connectivity

Secure application identity, authentication and authorization

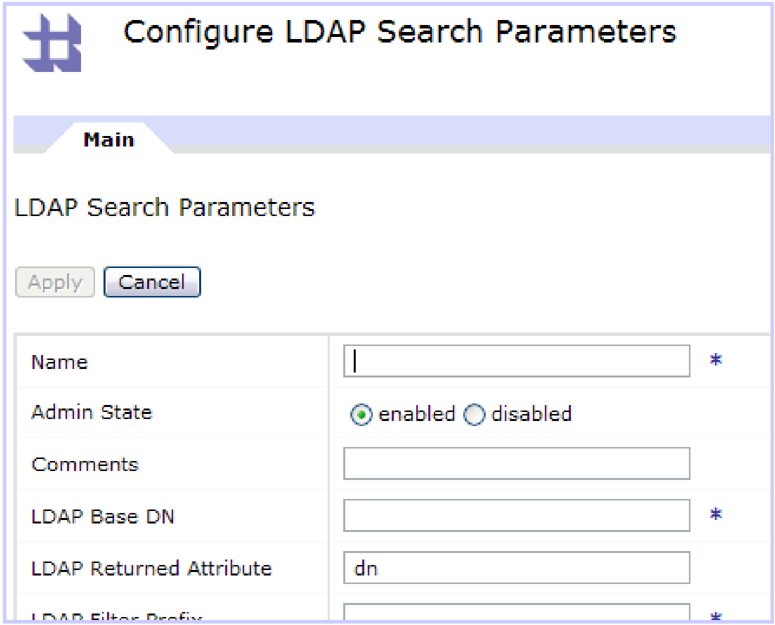
Identity management, access control, authorization, and authentication mechanisms (AAA) are essential

ESB support many protocols and transports
- Web Services, MQ, JMS, HTTP and HTTPS

ESB supports a broad variety of security tokens
- Userid/pw, X509, SAML, Kerberos, LTPA…

ESB performs role of Policy Enforcement Point (PEP)
- Provides a secure infrastructure
- Ensures conformance to centralized security policy

Many different technologies supported
- Lightweight Directory Access Protocol (LDAP)
- Microsoft Active Directory, Open LDAP…
- Tivoli Federated Identity Manager (TFIM)
- zOS SAF including RACF
- Security hardened DMZ device strengths

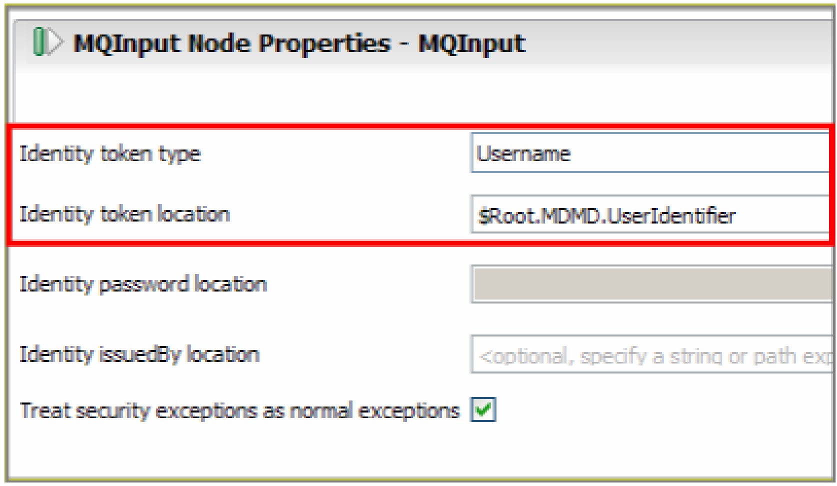

Configure LDAP Search Parameters

Main

LDAP Search Parameters

Apply Cancel

Name		*
Admin State	[✓ enabled] disabled	
Comments		
LDAP Base DN		*
LDAP Returned Attribute	dn	


MQInput Node Properties - MQInput

Identity token type	Username
Identity token location	$Root.MDMD.UserIdentifier
Identity password location	
Identity issuedBy location	<optional, specify a string or path exp
Treat security exceptions as normal exceptions	✓

Derive Value from an Application Inventory

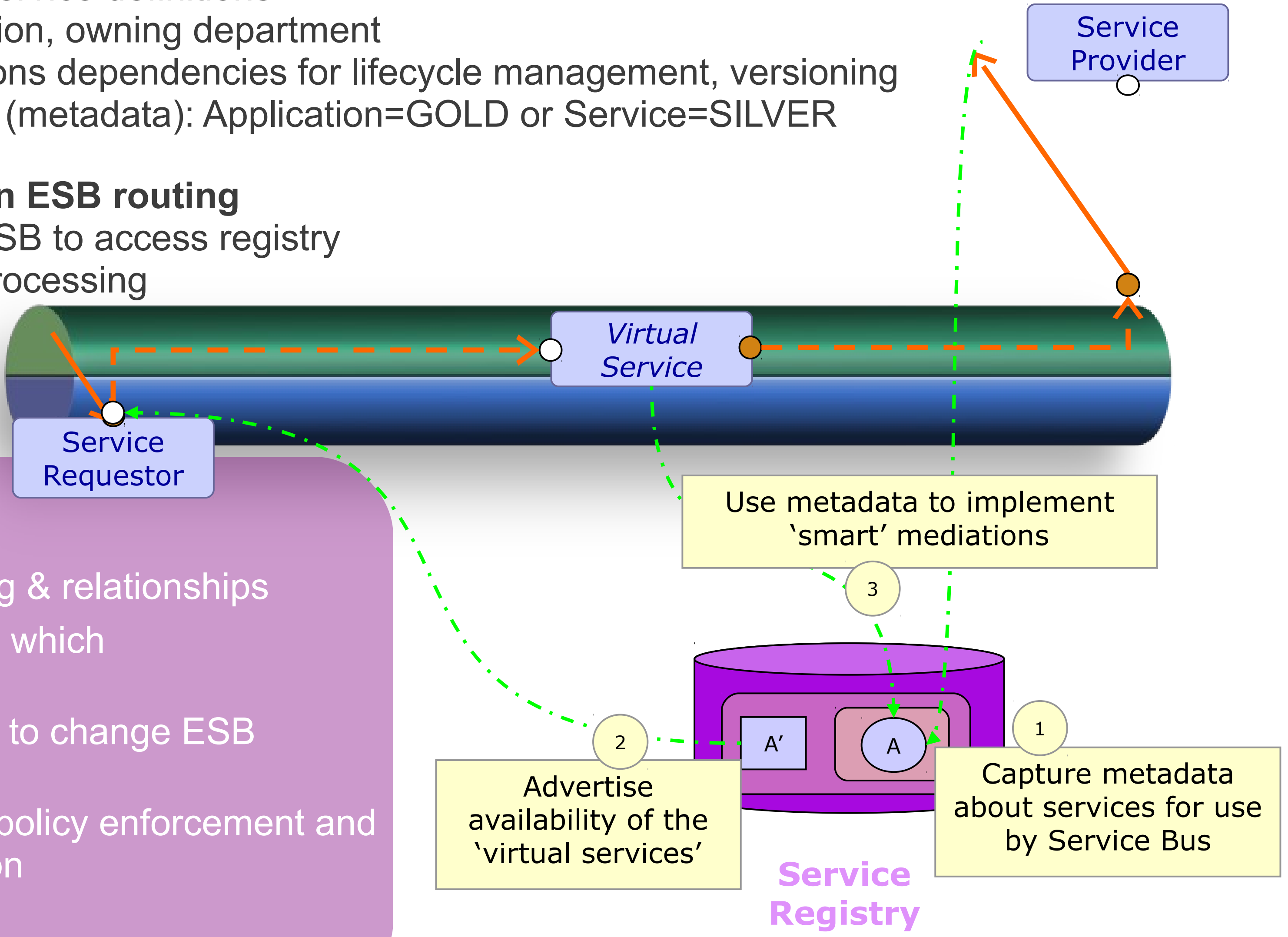
Understand your application assets and control their access dynamically

Catalog application and service assets using a registry, e.g. WSRR

Web Service and MQ Service definitions
Classifications: by function, owning department
Relationships: applications dependencies for lifecycle management, versioning
User defined properties (metadata): Application=GOLD or Service=SILVER

Use registry information in ESB routing

Built-in facilities allow ESB to access registry
Enables policy based processing

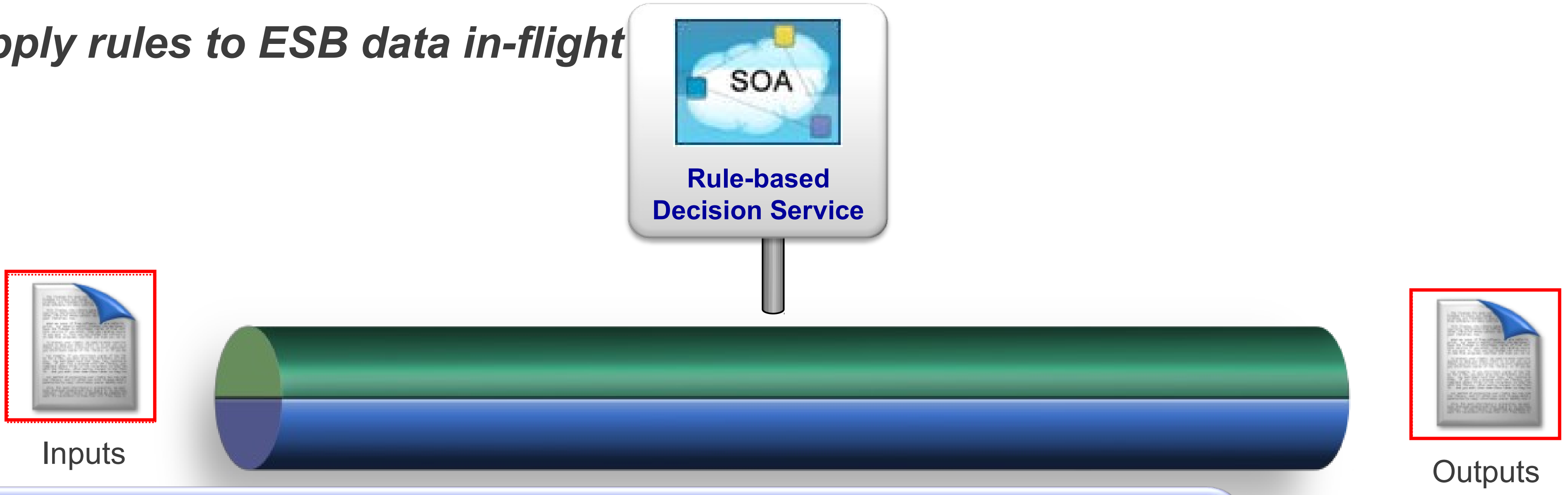

Service Provider
Virtual Service
Service Requestor
Use metadata to implement 'smart' mediations
3
2
A'
A
1
Advertise availability of the 'virtual services'
Capture metadata about services for use by Service Bus
Service Registry

Primary use cases:

- **Visibility:** application catalog & relationships
- **Governance**: who accesses which applications/services
- **Dynamicity**: update registry to change ESB behaviour without redeploy
- **Policy based Processing:** policy enforcement and policy based service selection

Business Rules for Smart Connectivity

Apply rules to ESB data in-flight



Rule-based Decision Services render decisions on input data
Most often this data comes from a variety of data sources
i.e. aggregation, transformation is needed

Rule-based Decision Services send outcome decisions to other systems
Output data needs to be transported and dispatched to one or many systems

Automate decisions
Implement, manage & share decisions services across IT infrastructure
ILOG JRules for Embedded rules and ILOG Rules Server subsystem

Business Activity Monitoring & Event Intelligence

Understand the importance of ESB data and detect business situations

ESB connectivity allows processing of events from many sources, targets

- Capture business relevant information to feed to WebSphere Business Monitor
  - Examples: total dollar trade value per day, total number of orders per hour
- Capture business events for correlation using WebSphere Business Events
  - Look for correlations in data, e.g. fraud, Up-sell and Cross-sell opportunities, CRM
- Audit, Repair and Replay transported events

Generate Business Monitoring Events from existing connectivity

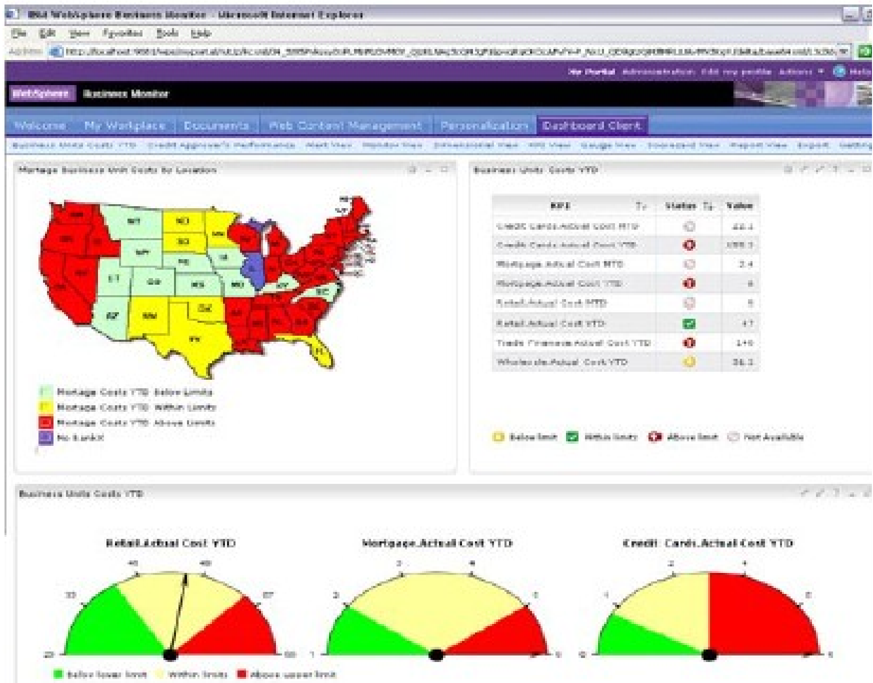
- Enables integration with WebSphere Monitor to display and analyze KPIs
- Design time and operational time event activation
- Notification via CEI & Publish subscribe

WebSphere Business Events

- Capture events from ESB and other sources
- Analyse to generates interesting new event

Capture Events for Audit and Logging

- Verify transport of traffic; dates and payloads
- Replay recorded messages to consumers
  - Includes replay to ESB for reprocessing



Initiate and Support Business Processes

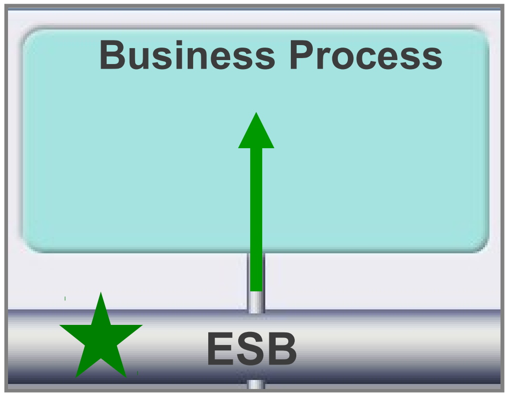
Compose existing applications and services to create new value

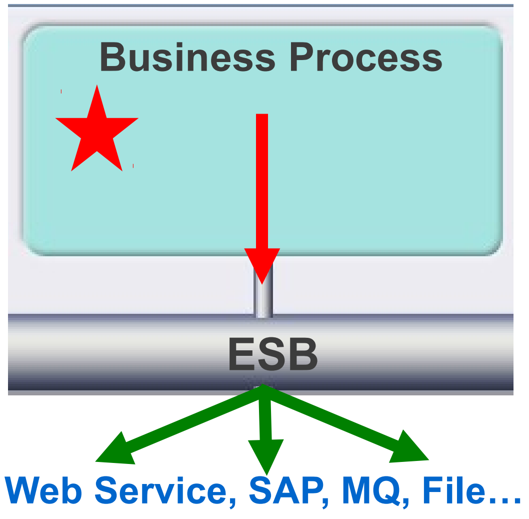
ESB Event Capture and Process Initiation

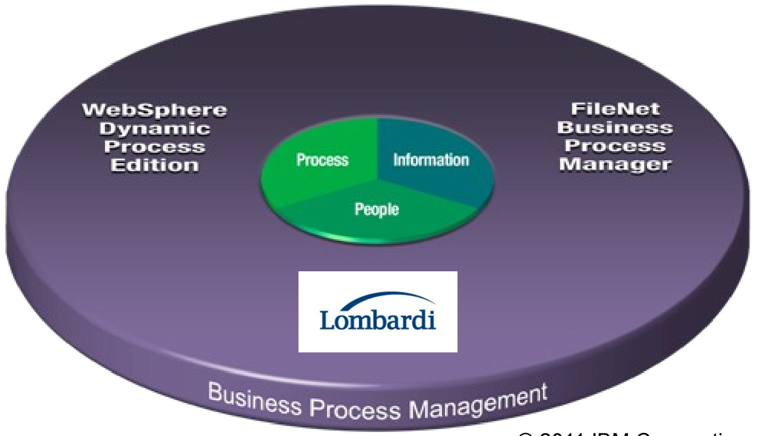
- Breadth of ESB connectivity enables multiple business process starting points
  - Identify event and initiate business process
  - e.g. message, file, web service, device endpoints can start business process
- Synchronous and asynchronous invocation for short & long running transactions
  - Multiple options with Process Server, Lombardi, FileNet…

Business Process Connectivity

- Exploit range of ESB connectivity to abstract and simplify BPM
- Process focus on WHAT rather than ESB focus on WHERE, HOW concerns
- ESB receives service request and routes, re-formats, interacts with provider





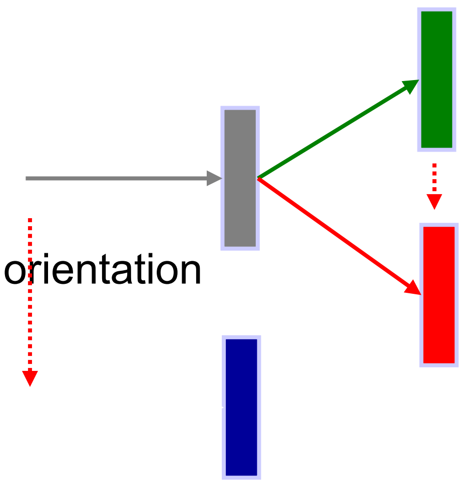

WebSphere Dynamic Process Edition
FileNet Business Process Manager
Process
Information
People
Lombardi
Business Process Management

A Flexible Infrastructure to Support Change

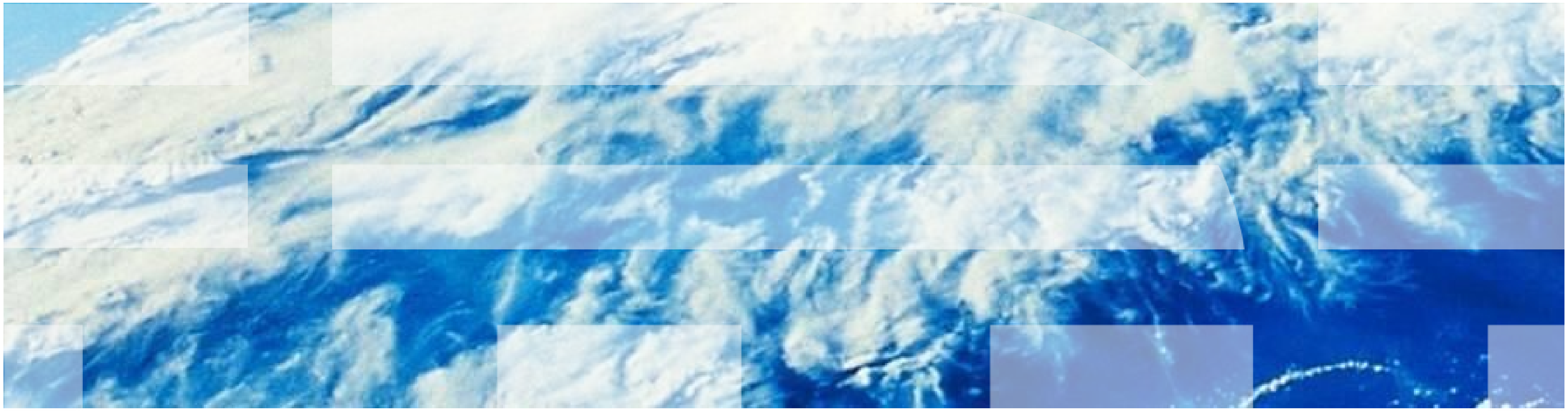
Enable Application and Service Replacement with minimum risk

- ESB creates a Virtual Service
  - Implementation details of a service to be hidden
  - Flexibility in implementation; change implementations without affecting consumers
  - Introduce new interfaces to existing service in parallel with new interfaces

- Examples include M&A, Decommissioning & External partner communication
  - Connect newly acquired systems, particularly relevant in M&A
    - Formats and Protocols of acquired technology differ from current systems
    - ESB provides managed interface to acquired systems for in-house systems
      - Provides new interface for acquired systems
  - Staged decommission of legacy implementations
    - Maintain existing interface to new implementation
      - Allows Managed risk of client migration
      - Often combined with new interface definition, often to enable service orientation
  - External partner communication
    - ESB provides interface to external systems
    - Allows partners to be swapped in and out without affecting consumers



Pattern Technology



Message Broker 7 Overview

- **Simplicity and Productivity**
  - Radically streamlined product prerequisites and components
  - Simplified connectivity solution development using IBM pre-supplied patterns
  - Impact Analysis to manage development artefact changes including ESQL, Maps and Message sets
  - MB Explorer for dedicated administration tooling
  - SCA nodes for WPS Interoperability

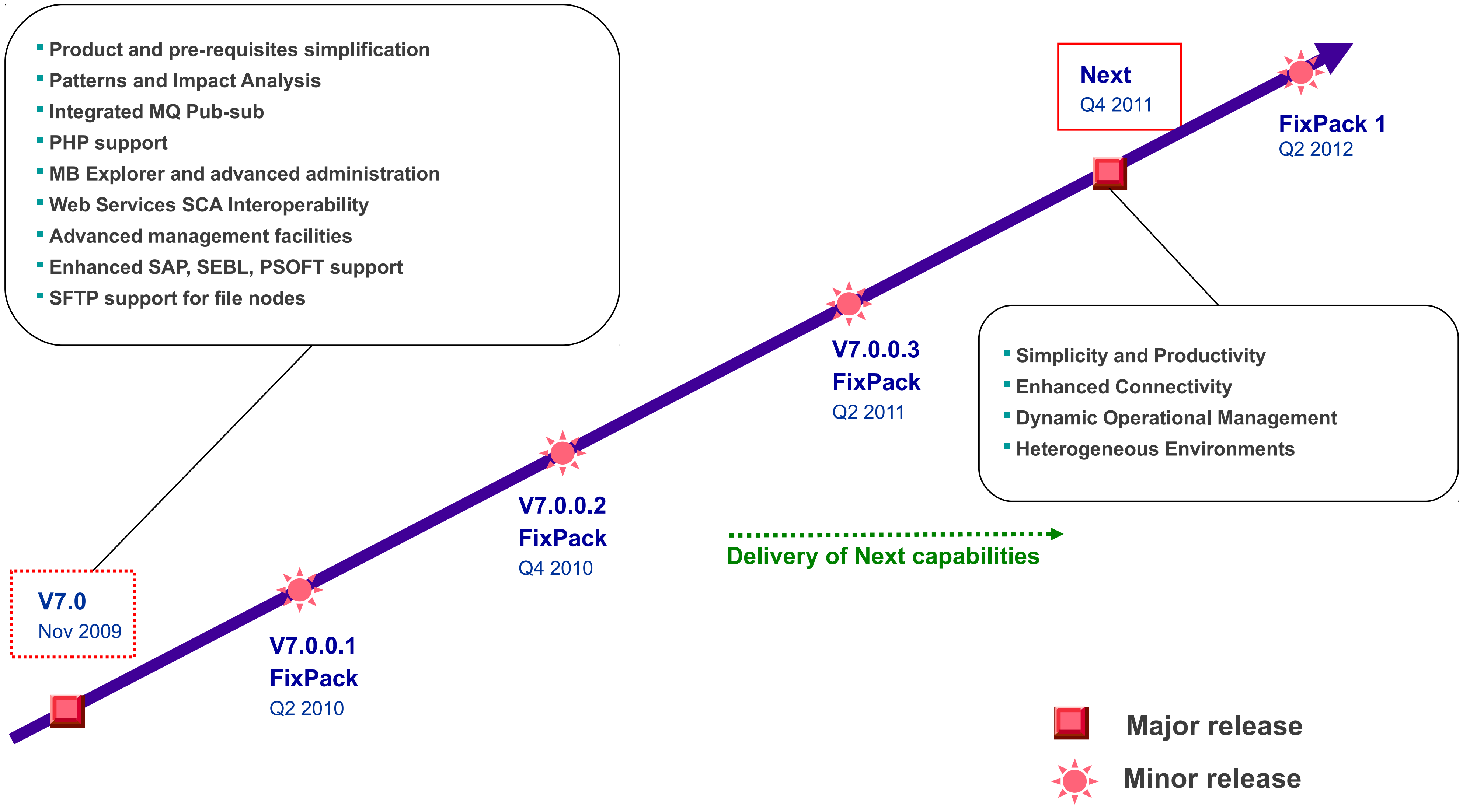
- **Universal Connectivity for SOA**
  - Extended & integrated publish subscribe: common management & security with new MQ capabilities
  - PHP nodes for Web 2.0 support
  - Enhanced SAP, Siebel, PeopleSoft packaged application support
  - New Sequence and Resequence nodes

- **Dynamic Operational Management**
  - New operational facilities for audit and monitoring, including WBM
  - Enhanced statistics to understand broker performance, including memory usage
  - Improved user trace to easily understand message flow behaviour
  - Enhancements for WSRR processing including support for FSM protocol
  - Support and Exploit MQ Multi-instance Queue Managers for High Availability

- **Platforms, Environments and Performance**
  - Exclusively 64bit Broker support
  - Performance monitoring tools and very reduced memory footprint

Message Broker Product Roadmap

IBM's plans, directions, and intent are subject to change or withdrawal



- Product and pre-requisites simplification
- Patterns and Impact Analysis
- Integrated MQ Pub-sub
- PHP support
- MB Explorer and advanced administration
- Web Services SCA Interoperability
- Advanced management facilities
- Enhanced SAP, SEBL, PSOFT support
- SFTP support for file nodes

- Simplicity and Productivity
- Enhanced Connectivity
- Dynamic Operational Management
- Heterogeneous Environments

Patterns for Simplified Development

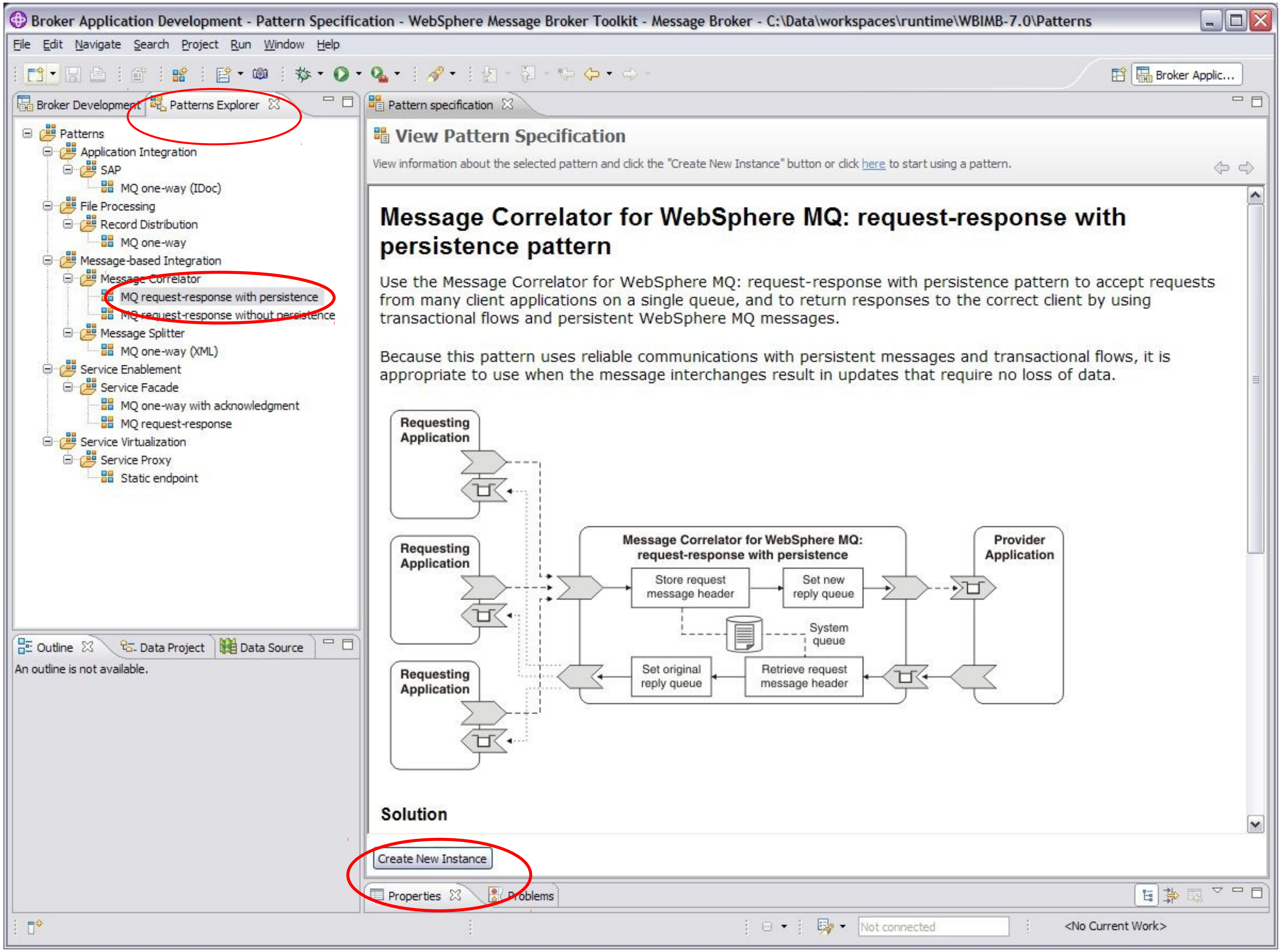
- **Patterns Based Development**
  - Create top-down, parameterized connectivity solutions
    - e.g. Web Service façades, Message oriented processing, Queue to File
  - IBM pre-supplied patterns
    - Simplifies creation of most common scenarios according to best practices
  - Complements existing bottom-up constructional approach for bespoke connectivity

- **Patterns Explorer**
  - Inventory of key patterns available for solution generation
  - Each pattern contains clear help to explain context and applicability

- **Pattern Generation**
  - Enables simple creation of solution artefacts from pre-supplied pattern
  - Pattern Properties allow configuration of behaviour
  - Solutions can be modified and/or regenerated

- **Evolution**
  - Pattern Capture creates user patterns from solution artefacts
  - Pattern Management: provides post deployment customization and operation of solutions

Pattern Technology Demo (1/4)



Broker Application Development - Pattern Specification - WebSphere Message Broker Toolkit - Message Broker - C:\Data\workspaces\runtime\WBIMB-7.0\Patterns

File Edit Navigate Search Project Run Window Help

Broker Development | Patterns Explorer

- Patterns
  - Application Integration
    - SAP
      - MQ one-way (IDoc)
  - File Processing
    - Record Distribution
      - MQ one-way
  - Message-based Integration
    - Message Correlator
      - MQ request-response with persistence
      - MQ request-response without persistence
    - Message Splitter
      - MQ one-way (XML)
  - Service Enablement
    - Service Facade
      - MQ one-way with acknowledgment
      - MQ request-response
  - Service Virtualization
    - Service Proxy
      - Static endpoint

Outline | Data Project | Data Source

An outline is not available.

Pattern specification

View Pattern Specification

View information about the selected pattern and click the "Create New Instance" button or click here to start using a pattern.

Message Correlator for WebSphere MQ: request-response with persistence pattern

Use the Message Correlator for WebSphere MQ: request-response with persistence pattern to accept requests from many client applications on a single queue, and to return responses to the correct client by using transactional flows and persistent WebSphere MQ messages.

Because this pattern uses reliable communications with persistent messages and transactional flows, it is appropriate to use when the message interchanges result in updates that require no loss of data.

Solution

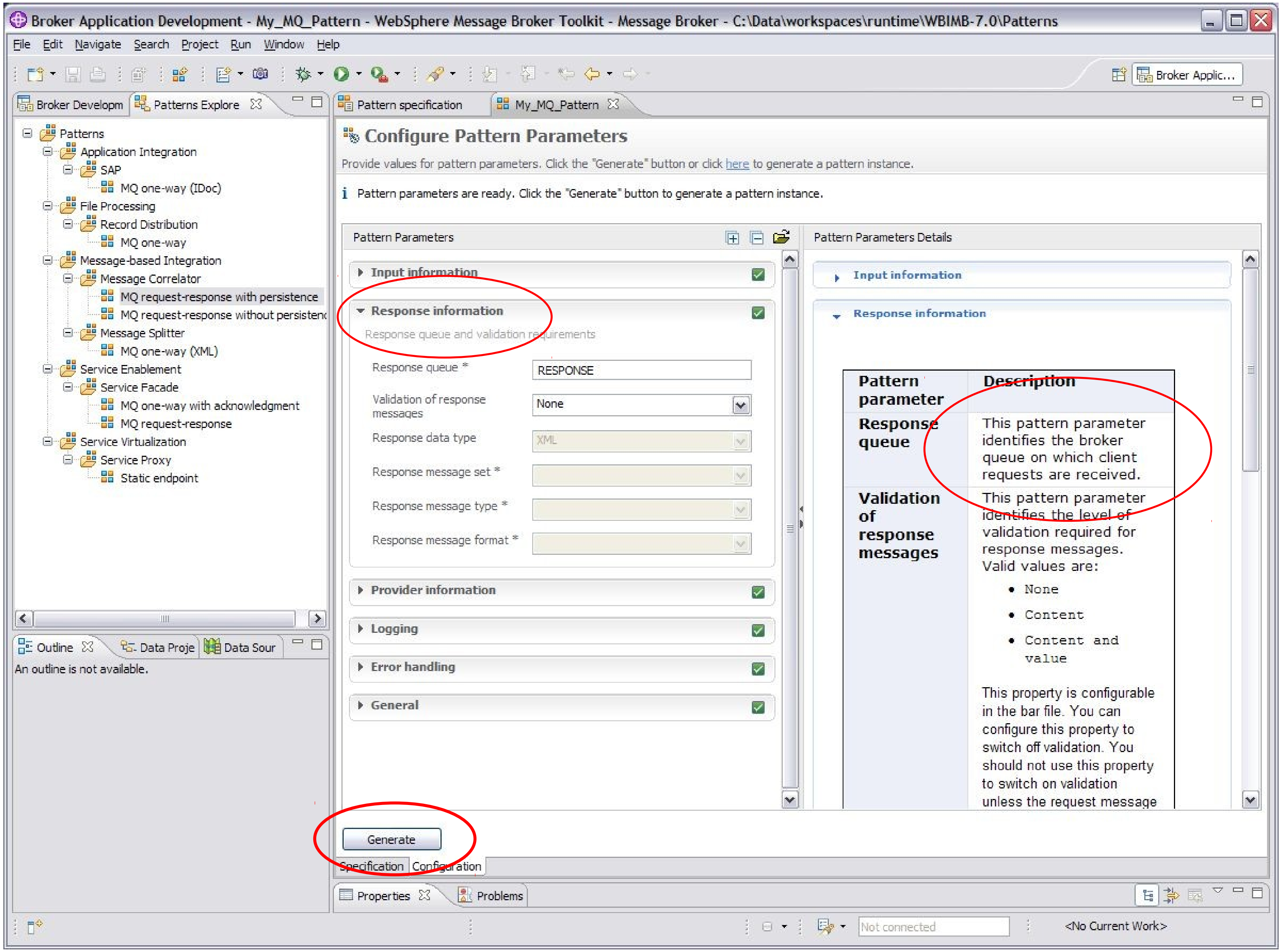
Create New Instance

Properties | Problems

Not connected

<No Current Work>

Pattern Technology Demo (2/4)



Broker Application Development - My_MQ_Pattern - WebSphere Message Broker Toolkit - Message Broker - C:\Data\workspaces\runtime\WBIMB-7.0\Patterns

File Edit Navigate Search Project Run Window Help

Broker Develop | Patterns Explore | Pattern specification | My_MQ_Pattern

- Patterns
  - Application Integration
    - SAP
      - MQ one-way (IDoc)
  - File Processing
    - Record Distribution
      - MQ one-way
  - Message-based Integration
    - Message Correlator
      - MQ request-response with persistence
      - MQ request-response without persistenc
    - Message Splitter
      - MQ one-way (XML)
  - Service Enablement
    - Service Facade
      - MQ one-way with acknowledgment
      - MQ request-response
  - Service Virtualization
    - Service Proxy
      - Static endpoint

Outline | Data Proje | Data Sour

An outline is not available.

Configure Pattern Parameters

Provide values for pattern parameters. Click the "Generate" button or click here to generate a pattern instance.

Pattern parameters are ready. Click the "Generate" button to generate a pattern instance.

Pattern Parameters

- Input information ✓
- Response information ✓

Response queue and validation requirements

Response queue *	RESPONSE
Validation of response messages	None
Response data type	XML
Response message set *	
Response message type *	
Response message format *	

- Provider information ✓
- Logging ✓
- Error handling ✓
- General ✓

Generate

Specification | Configuration

Pattern Parameters Details

- Input information
- Response information

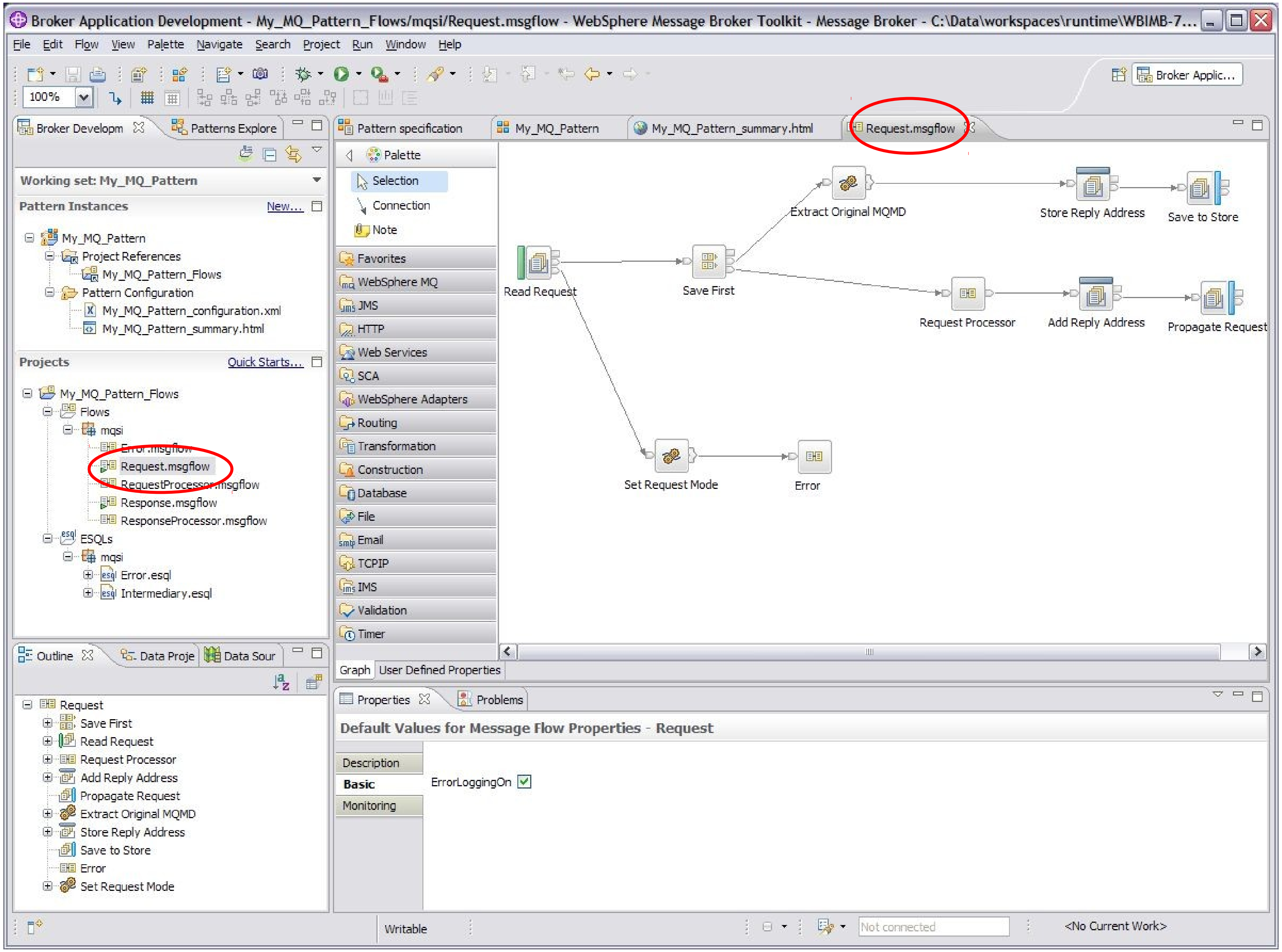
Pattern parameter	Description
Response queue	This pattern parameter identifies the broker queue on which client requests are received.
Validation of response messages	This pattern parameter identifies the level of validation required for response messages. Valid values are: • None • Content • Content and value This property is configurable in the bar file. You can configure this property to switch off validation. You should not use this property to switch on validation unless the request message

Properties | Problems

Not connected

<No Current Work>

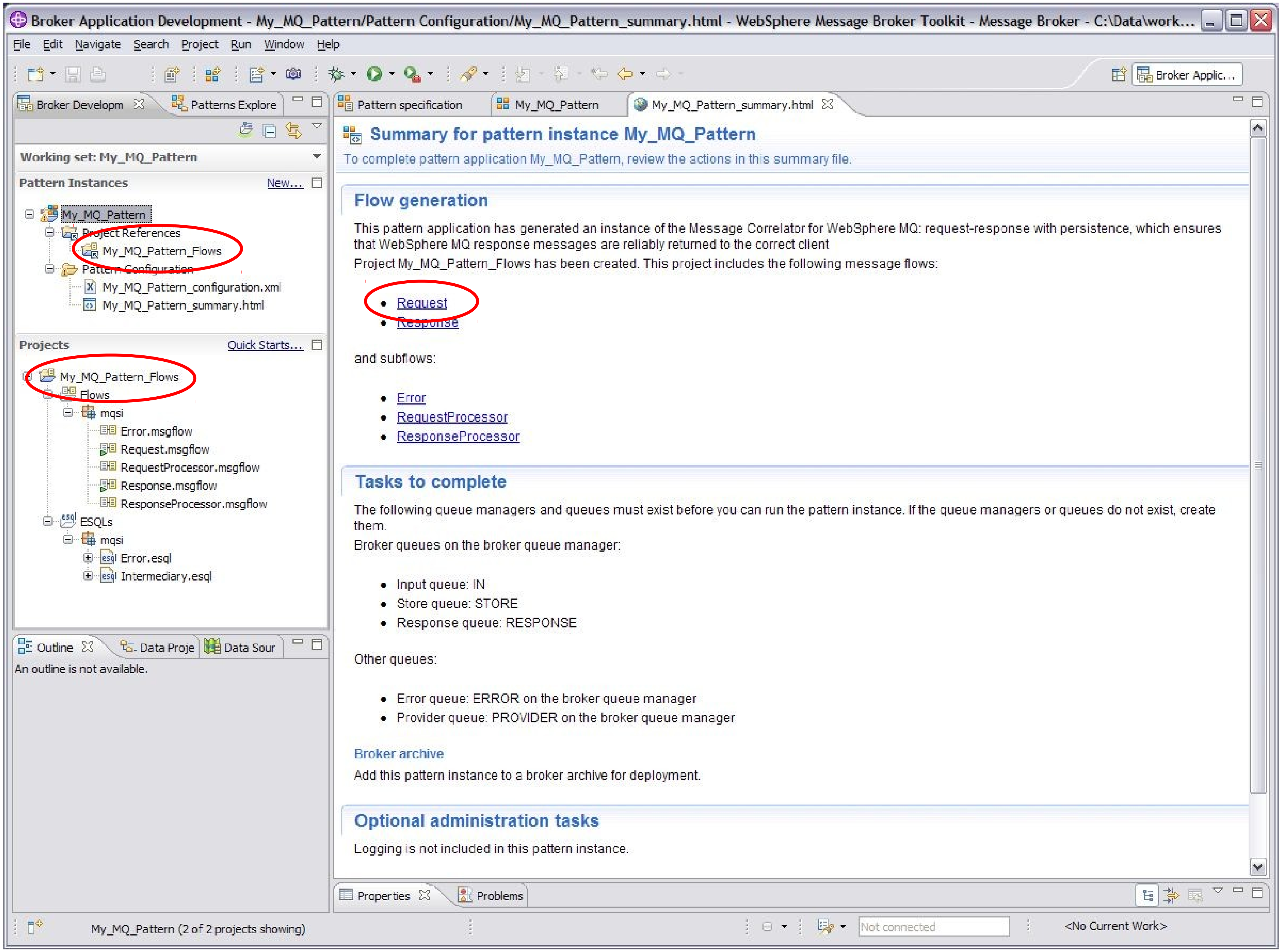
Pattern Technology Demo (3/4)



Default Values for Message Flow Properties - Request

ErrorLoggingOn ✓

Pattern Technology Demo (4/4)



Broker Application Development - My_MQ_Pattern/Pattern Configuration/My_MQ_Pattern_summary.html - WebSphere Message Broker Toolkit - Message Broker - C:\Data\work...
File Edit Navigate Search Project Run Window Help
Broker Develop
Patterns Explore
Working set: My_MQ_Pattern
Pattern Instances
New...
My_MQ_Pattern
Project References
My_MQ_Pattern_Flows
Pattern Configuration
My_MQ_Pattern_configuration.xml
My_MQ_Pattern_summary.html
Projects
Quick Starts...
My_MQ_Pattern_Flows
Flows
mqsi
Error.msgflow
Request.msgflow
RequestProcessor.msgflow
Response.msgflow
ResponseProcessor.msgflow
ESQLs
mqsi
Error.esql
Intermediary.esql
Outline
Data Proje
Data Sour
An outline is not available.
Pattern specification
My_MQ_Pattern
My_MQ_Pattern_summary.html
Properties
Problems
My_MQ_Pattern (2 of 2 projects showing)
Not connected
<No Current Work>
Broker Applic...

Summary for pattern instance My_MQ_Pattern

To complete pattern application My_MQ_Pattern, review the actions in this summary file.

Flow generation

This pattern application has generated an instance of the Message Correlator for WebSphere MQ: request-response with persistence, which ensures that WebSphere MQ response messages are reliably returned to the correct client

Project My_MQ_Pattern_Flows has been created. This project includes the following message flows:

- Request
- Response

and subflows:

- Error
- RequestProcessor
- ResponseProcessor

Tasks to complete

The following queue managers and queues must exist before you can run the pattern instance. If the queue managers or queues do not exist, create them.

Broker queues on the broker queue manager:

- Input queue: IN
- Store queue: STORE
- Response queue: RESPONSE

Other queues:

- Error queue: ERROR on the broker queue manager
- Provider queue: PROVIDER on the broker queue manager

Broker archive

Add this pattern instance to a broker archive for deployment.

Optional administration tasks

Logging is not included in this pattern instance.

Thank you!

Contact: Andy Piper andy.piper@uk.ibm.com
http://twitter.com/andypiper | http://andypiper.co.uk