

First Steps With WebSphere Message Broker: Application Integration for the Messy

Dave Crighton – WebSphere Message Broker L3 Service Delivery Lead
IBM Hursley – davicrig@uk.ibm.com

Monday 6th August 2012
Session Number 11385

Notes

- Welcome to this Technical Introduction to WebSphere Message Broker.
- All slides in this presentation have at least one corresponding notes slide like this one, which contains further information on the topic being discussed, and/or links to web pages.
- Only this notes slide will be shown during the presentation. To view all other notes slides, please download and view a copy of this presentation.
- The WebSphere Message Broker homepage can be found at <http://www.ibm.com/software/integration/wbimessagebroker/>

Agenda

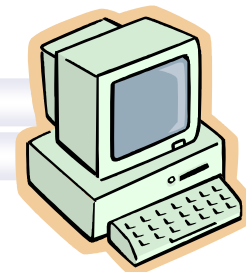
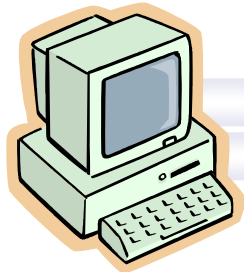
- What is Message Broker?
- Programming Concepts
 - Message Flows
 - Nodes
 - Message Model
- Product Overview
 - Components
 - User Roles and Environments
 - Connectivity Scenarios
- Demonstration

What is Message Broker?

- Message Broker enables “universal connectivity” by integrating protocols, message formats and mediation patterns
 - Emphasis on application re-use
- Fits naturally with WebSphere MQ
 - Robust, scalable architecture
 - Optimized for high throughput
 - Flexible broker topologies
- Three programming constructs are used:
 - Message Flows
 - Nodes
 - Message Models

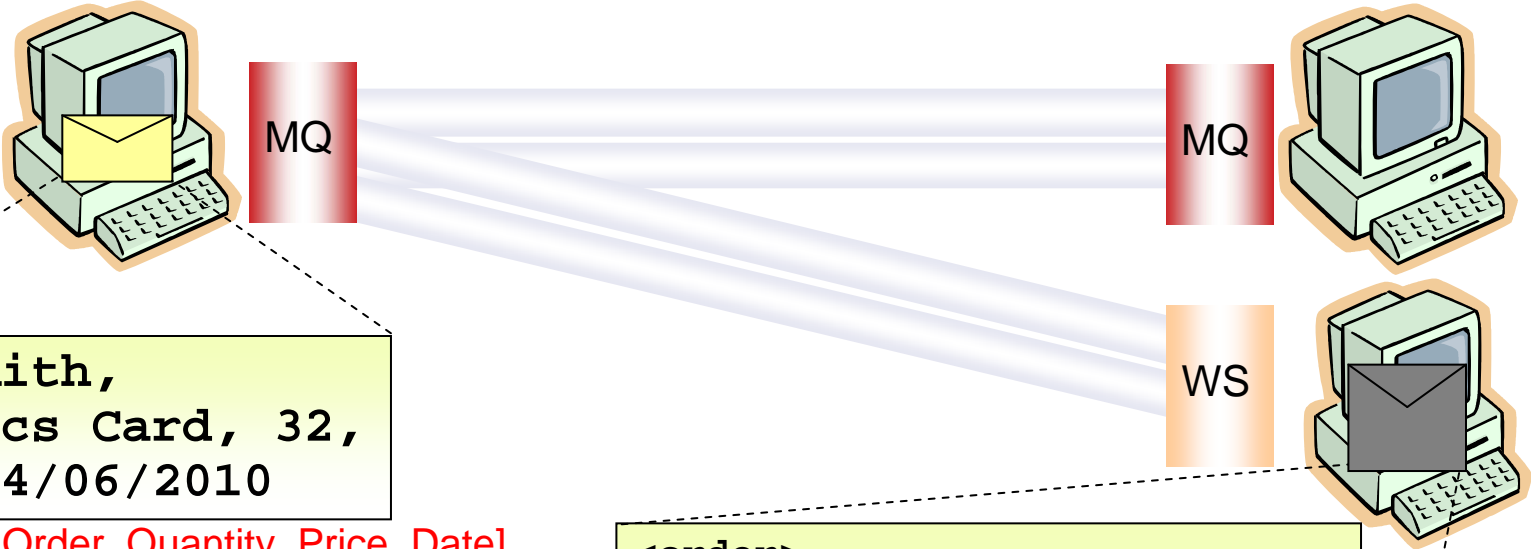


Application Connectivity



- Protocols
 - e.g. MQ, TCP/IP, HTTP, File system, FTP, SMTP, POP3 etc.
- Message Formats
 - e.g. Binary (C/COBOL), XML, Industry (SWIFT, EDI, HL7), User-defined
- Mediation Patterns
 - e.g. Route, Transform, Enrich, Filter, Monitor, Distribute, Decompose, Correlate, Fire and Forget, Request/Reply, Publish/Subscribe, Aggregation, Fan-in, Complex Event Processing

Mediation Patterns – Routing and Transformation



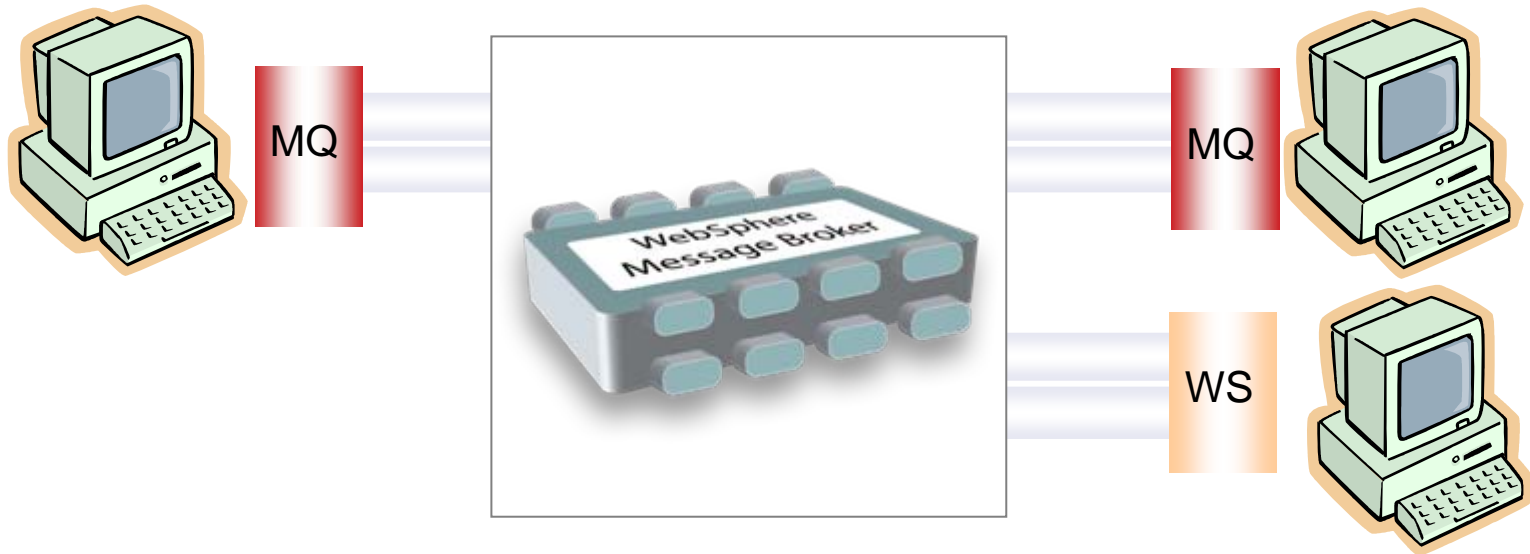
Mr. Smith,
Graphics Card, 32,
100, 24/06/2010

[Customer, Order, Quantity, Price, Date]

```
<order>  
  <name>  
    <first>John</first>  
    <last>Smith</last>  
  </name>  
  <item>Graphics Card</item>  
  <quantity>32</quantity>  
  <price>200</price>  
  <date>06/24/2010</date>  
</order>
```

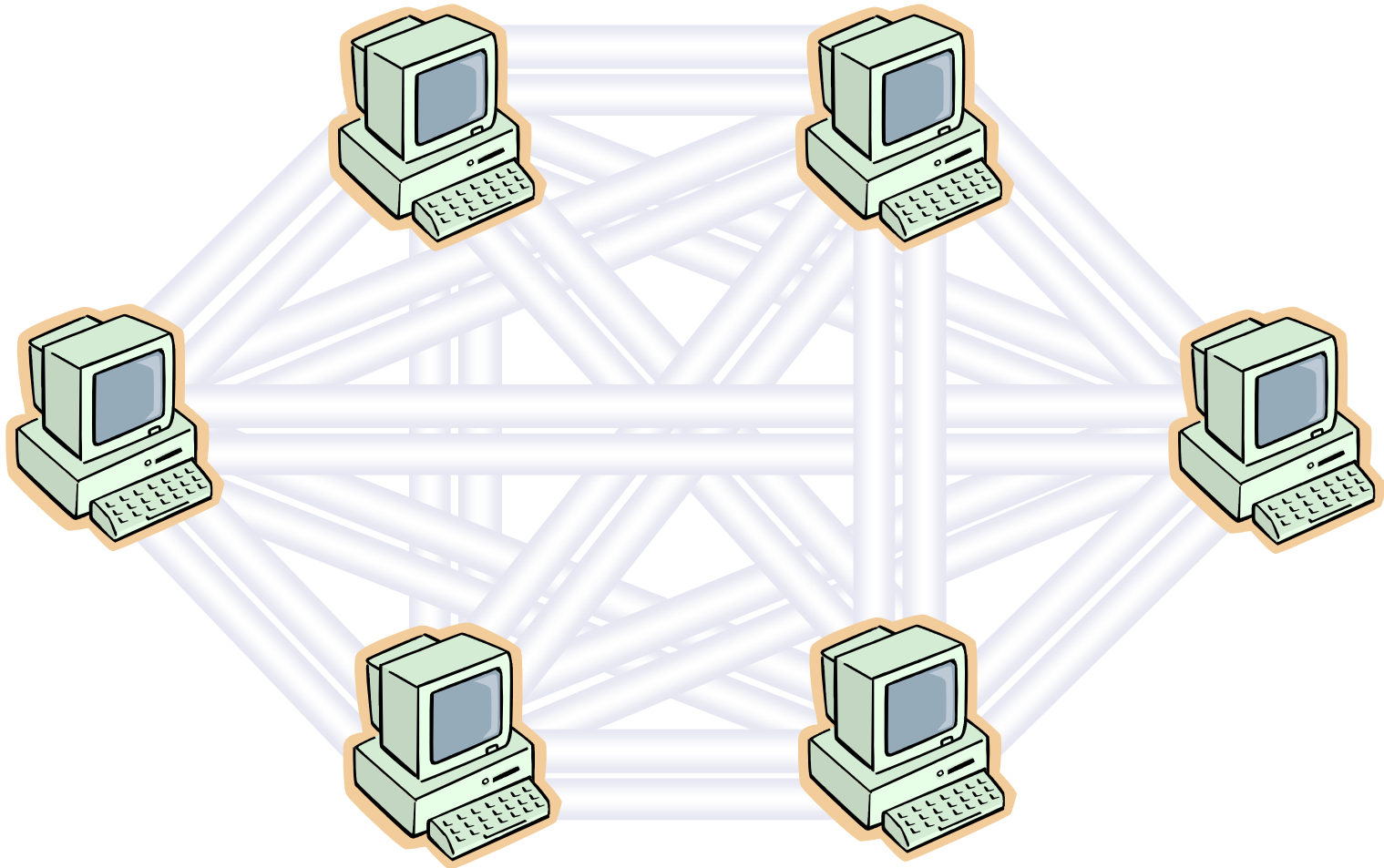
[Customer, Order, Quantity, Price, Date]

Application Connectivity with WMB

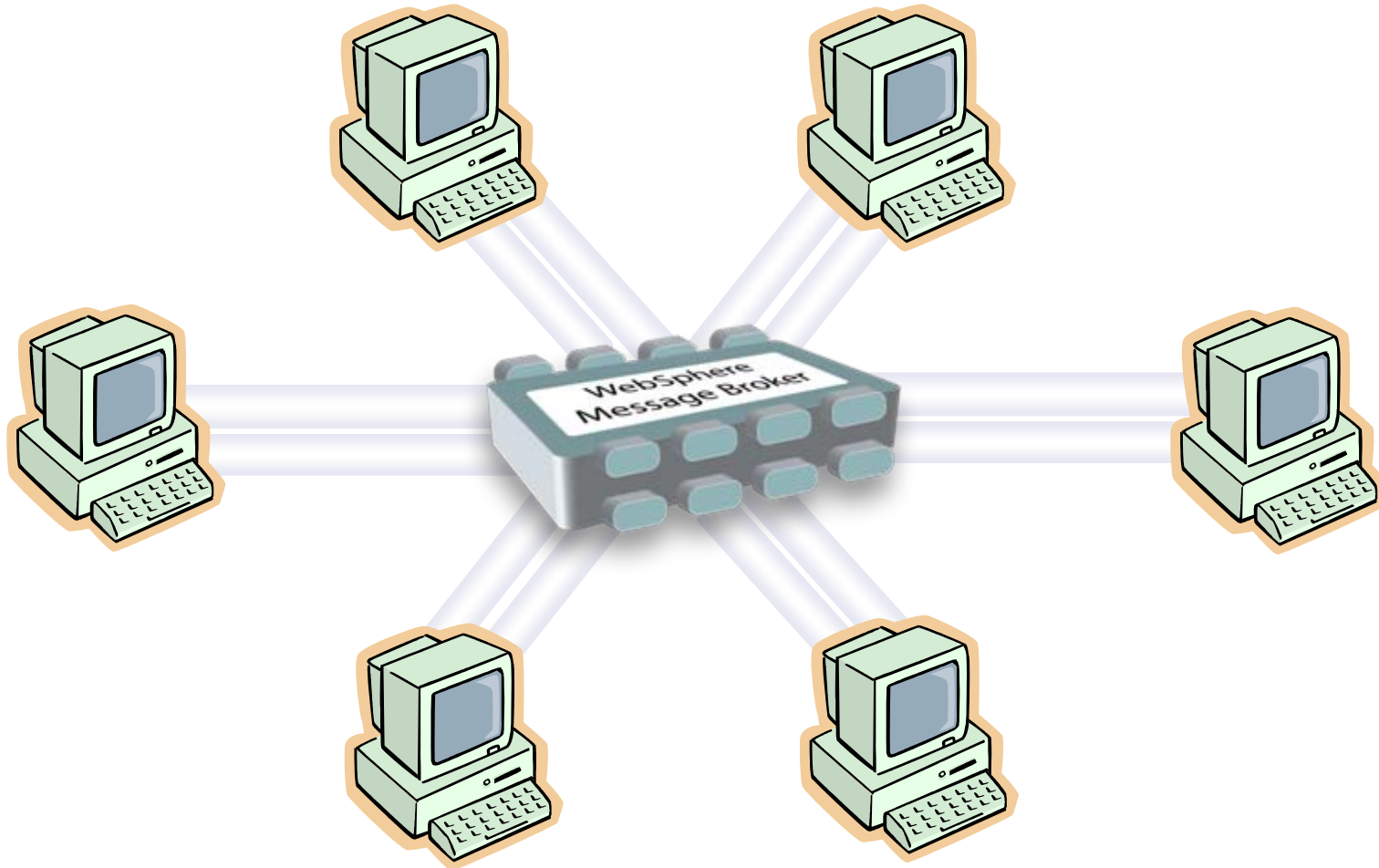


- WMB can act as an intermediary
- Flexible topologies

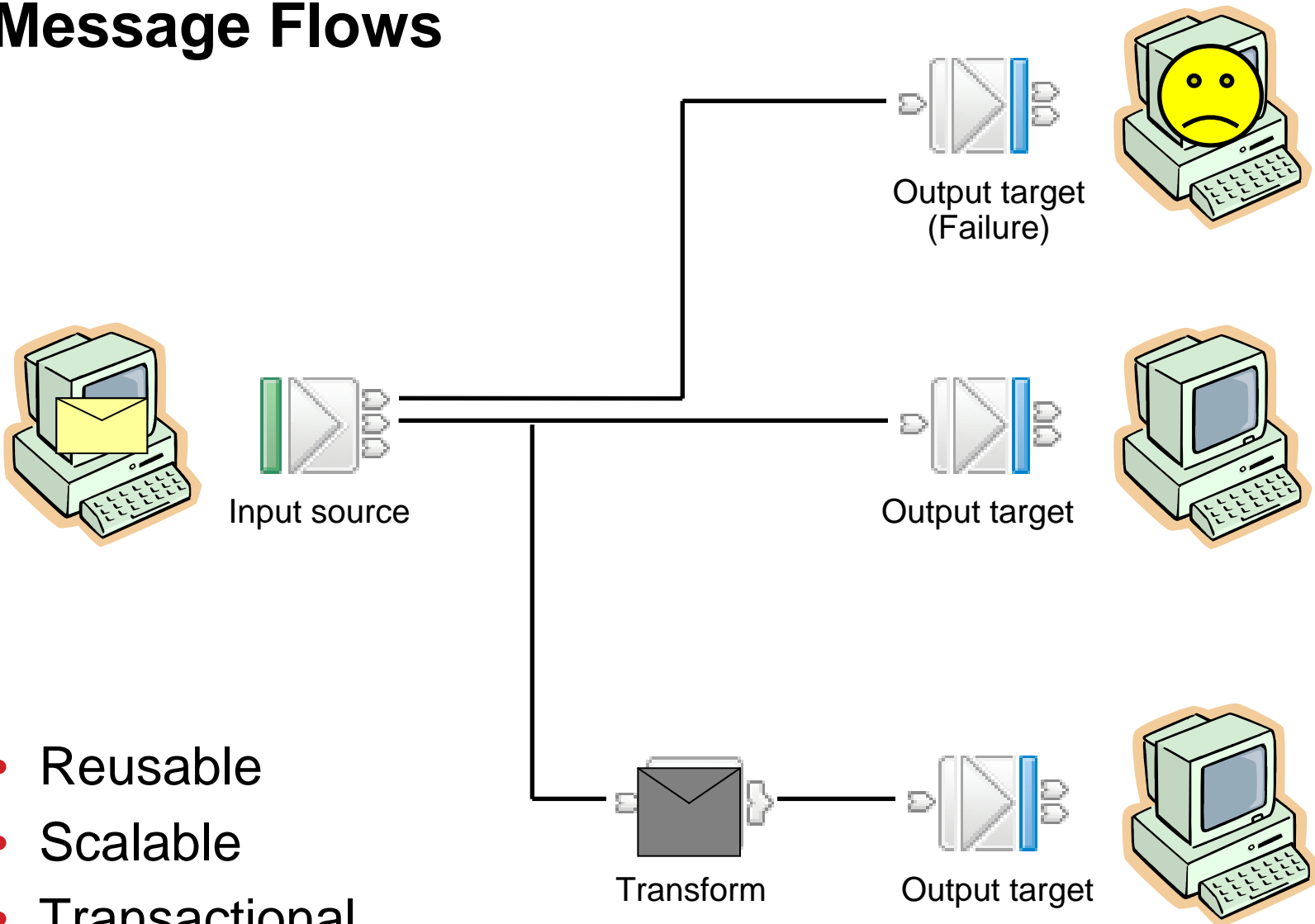
Application Integration for the Messy



Application Integration made less Messy

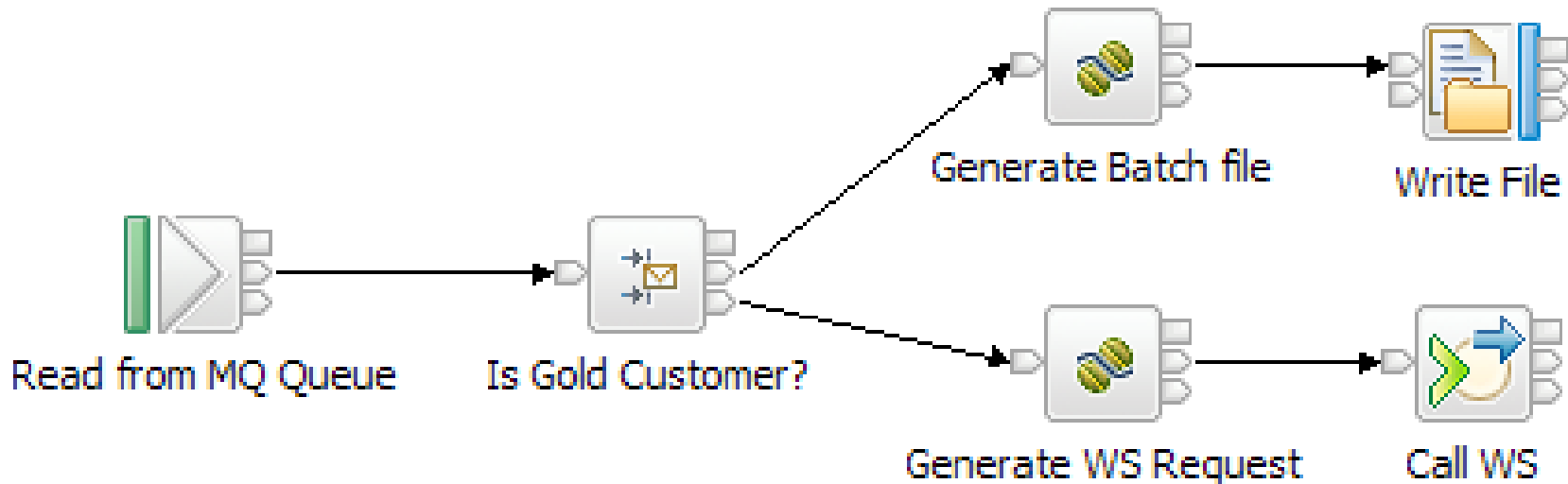


Message Flows



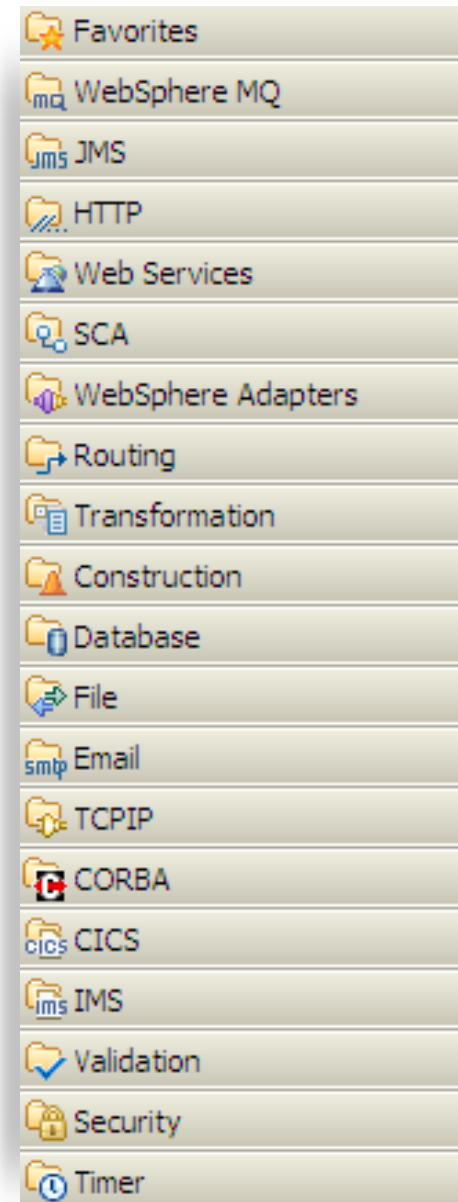
- Reusable
- Scalable
- Transactional

Message Flow Example



Nodes

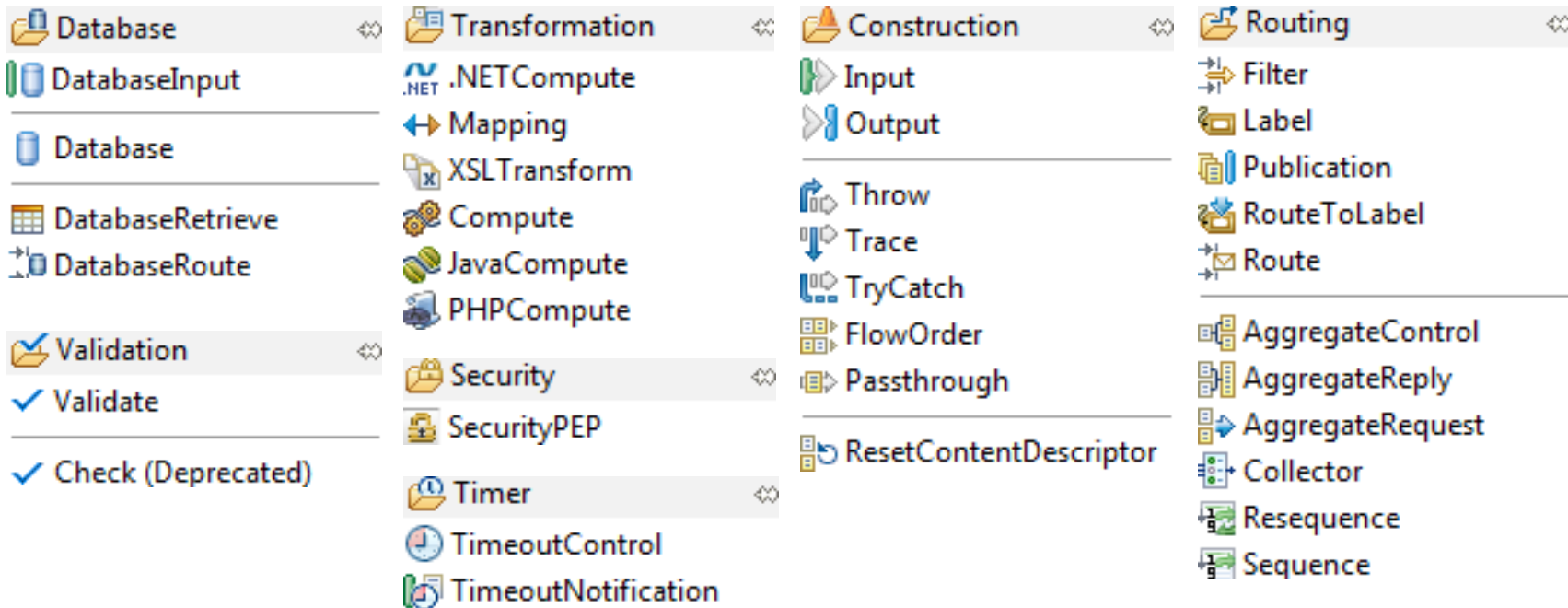
- The building blocks of message flows
- Each node type performs a different (input, output or processing) action
- Many different node types
 - Grouped into logical categories in the message flow editor
 - Nearly 100 nodes available out-of-the-box (as of WMB V8)



Lots of Nodes are Built in [1]

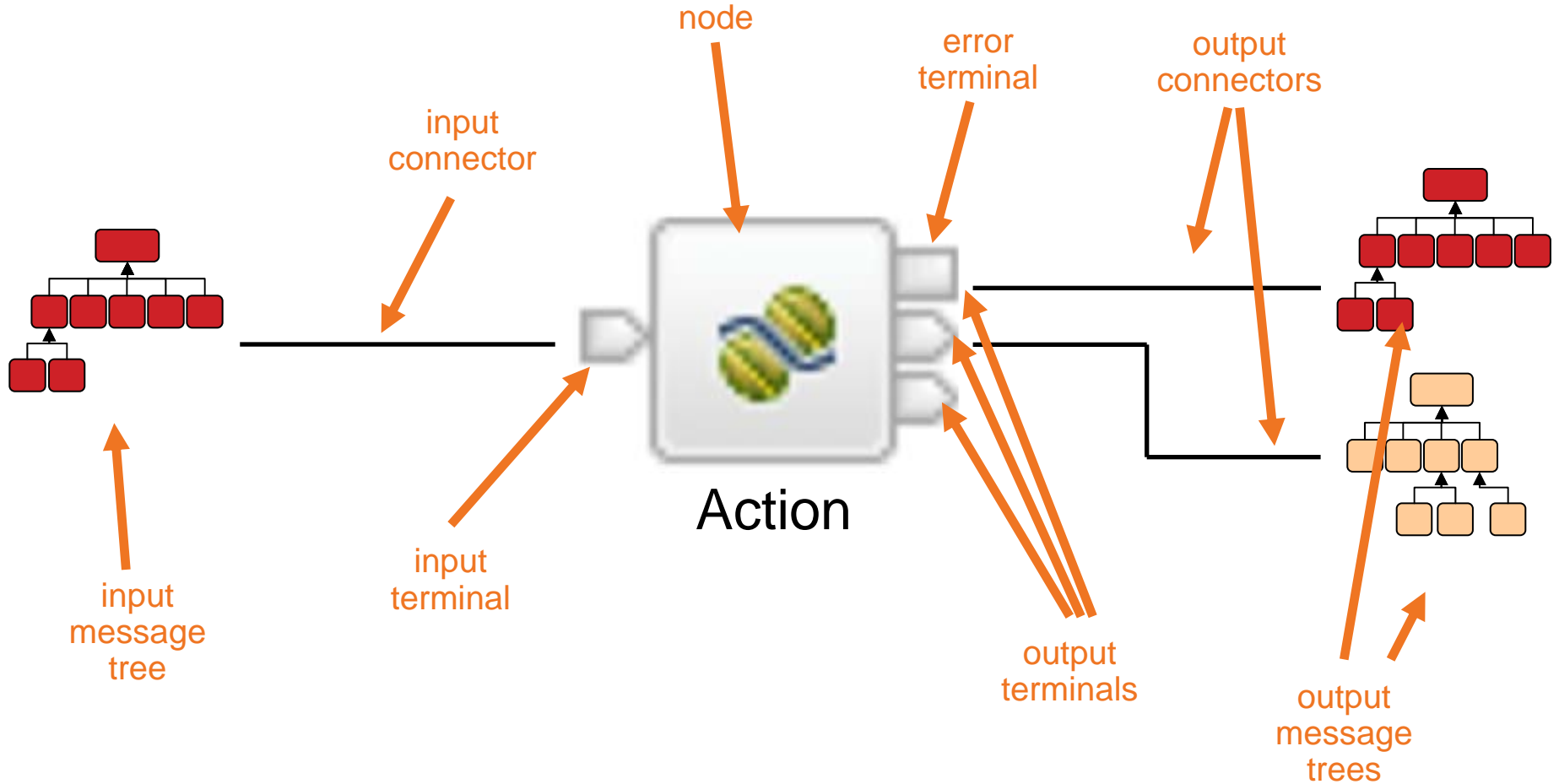
WebSphere MQ	Web Services	WebSphere Adapters	File
MQInput	SOAPInput	PeopleSoftInput	FileInput
MQOutput	SOAPReply	PeopleSoftRequest	FileOutput
MQReply	SOAPRequest		FileRead
MQGet	SOAPAsyncRequest	SAPInput	FTEInput
MQHeader	SOAPAsyncResponse	SAPRequest	FTEOutput
		SAPReply	
MQOptimizedFlow (De...)	SOAPEnvelope	SiebelInput	CDInput
JMS	SOAPExtract	SiebelRequest	CDOutput
JMSInput	RegistryLookup	JDEEdwardsInput	Email
JMSOutput	EndpointLookup	JDEEdwardsRequest	EmailInput
JMSReply			EmailOutput
JMSReceive	SCA	TwineBallInput	TCPIP
JMSHeader	SCAInput	TwineBallRequest	TCPIPClientInput
JMSMQTransform	SCAReply	CORBA	TCPIPClientOutput
MQJMSTransform	SCARequest	CORBARequest	TCPIPClientReceive
HTTP	SCAAsyncRequest		
HTTPInput	SCAAsyncResponse	IMS	TCPIPServerInput
HTTPReply	CICS	IMSRequest	TCPIPServerOutput
HTTPRequest	CICSRequest		TCPIPServerReceive
HTTPHeader			

Lots of Nodes are Built in [2]

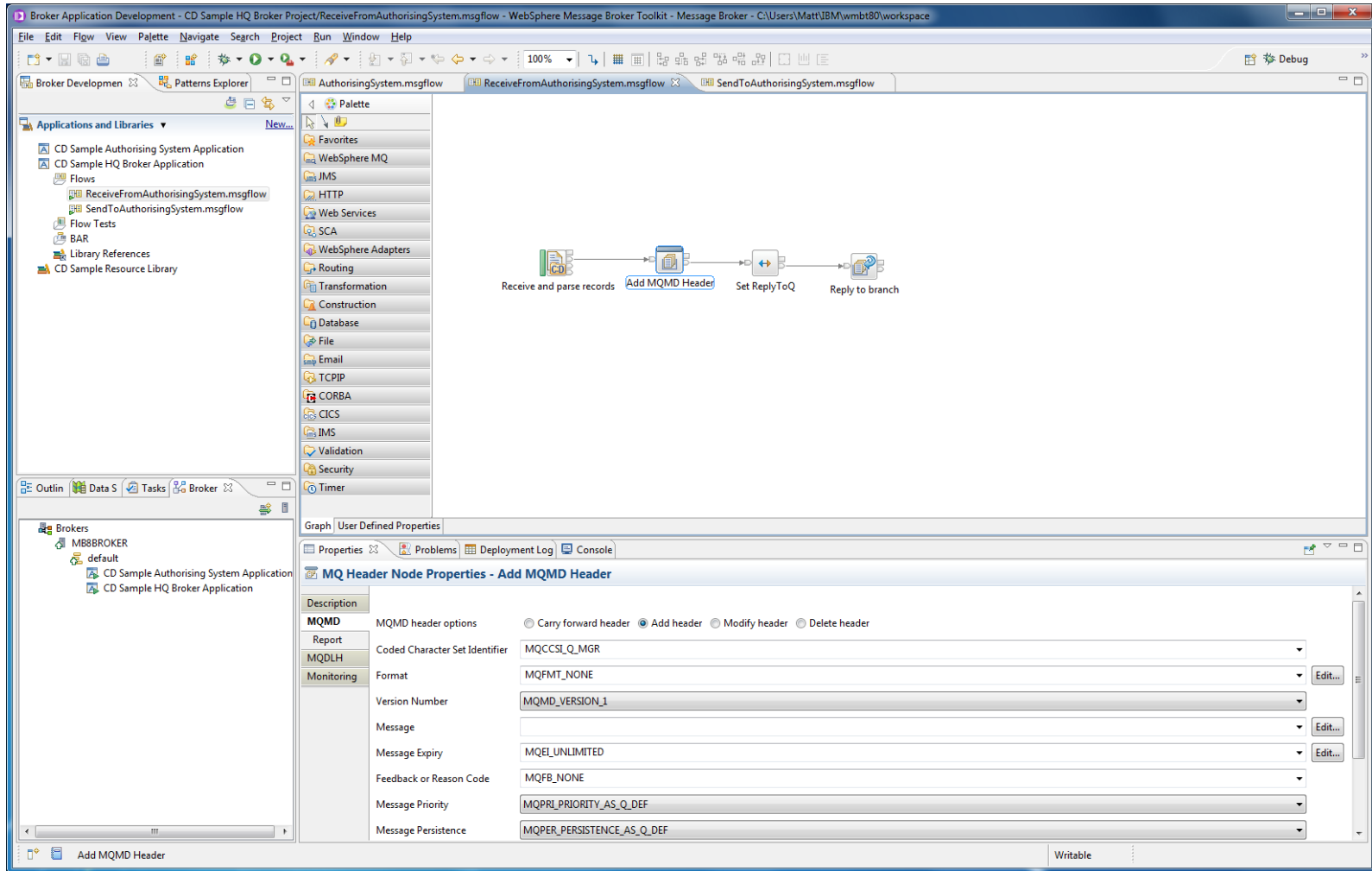


- Many other nodes available through product extensions and supportpacs
 - For example, WebSphere TX, Tibco RV, VSAM, QSAM
- Write your own User-Defined Nodes in C or Java

Node Terminology



Open, Extensible Tooling for Creating Message Flows

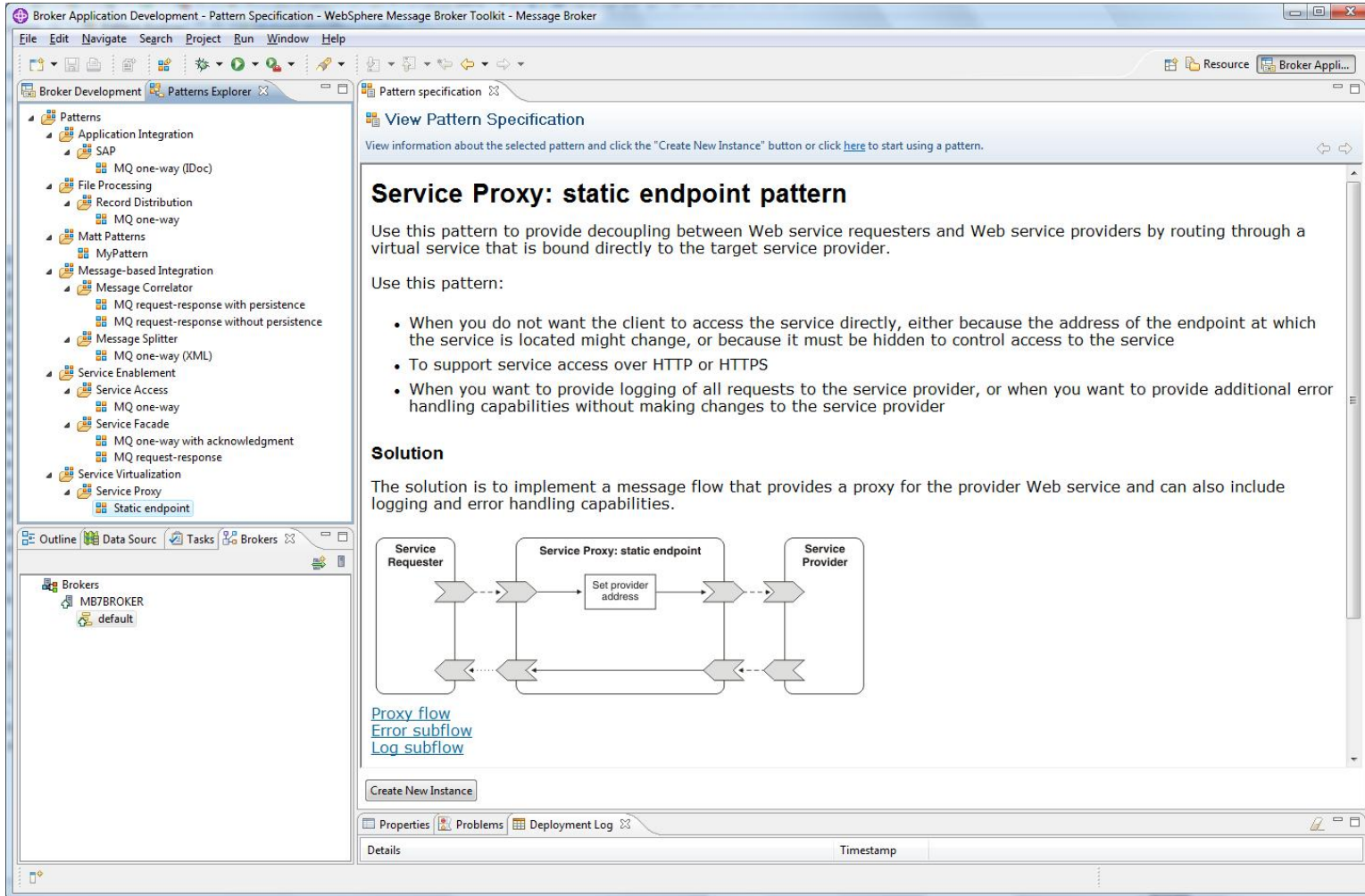


The screenshot displays the IBM WebSphere Message Broker Toolkit interface. The main workspace shows a message flow diagram with four nodes: "Receive and parse records", "Add MQMD Header", "Set ReplyToQ", and "Reply to branch". The "Add MQMD Header" node is selected, and its properties are shown in the bottom panel.

MQ Header Node Properties - Add MQMD Header

Property	Value
MQMD	<input type="radio"/> Carry forward header <input checked="" type="radio"/> Add header <input type="radio"/> Modify header <input type="radio"/> Delete header
Report	
MQDLH	Coded Character Set Identifier: MQCCSLQ_MGR
Monitoring	Format: MQFMT_NONE
	Version Number: MQMD_VERSION_1
	Message: [Empty]
	Message Expiry: MQEL_UNLIMITED
	Feedback or Reason Code: MQFB_NONE
	Message Priority: MQPRI_PRIORITY_AS_Q_DEF
	Message Persistence: MQPER_PERSISTENCE_AS_Q_DEF

Develop best practices quickly using patterns



The screenshot shows the IBM WebSphere Message Broker Toolkit interface. The main window displays the 'View Pattern Specification' for the 'Service Proxy: static endpoint pattern'. The left sidebar shows a tree view of patterns, with 'Service Proxy' selected. The main content area includes a description of the pattern, a list of use cases, and a solution section. A diagram illustrates the message flow between a Service Requester, a Service Proxy (containing a 'Set provider address' step), and a Service Provider. Below the diagram are links for 'Proxy flow', 'Error subflow', and 'Log subflow'. At the bottom, there is a 'Create New Instance' button and a 'Details' section with a 'Timestamp' field.

Service Proxy: static endpoint pattern

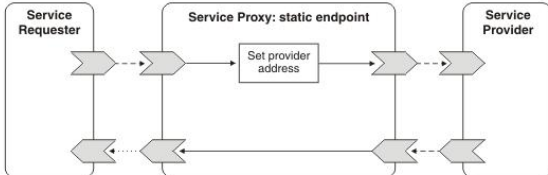
Use this pattern to provide decoupling between Web service requesters and Web service providers by routing through a virtual service that is bound directly to the target service provider.

Use this pattern:

- When you do not want the client to access the service directly, either because the address of the endpoint at which the service is located might change, or because it must be hidden to control access to the service
- To support service access over HTTP or HTTPS
- When you want to provide logging of all requests to the service provider, or when you want to provide additional error handling capabilities without making changes to the service provider

Solution

The solution is to implement a message flow that provides a proxy for the provider Web service and can also include logging and error handling capabilities.



```
graph LR
    SR[Service Requester] --> SP[Service Proxy: static endpoint]
    subgraph SP_Box [Service Proxy: static endpoint]
        SP_Set[Set provider address]
    end
    SP_Box --> SP_Provider[Service Provider]
    SP_Provider --> SP_Box
    SP_Provider --> SR
```

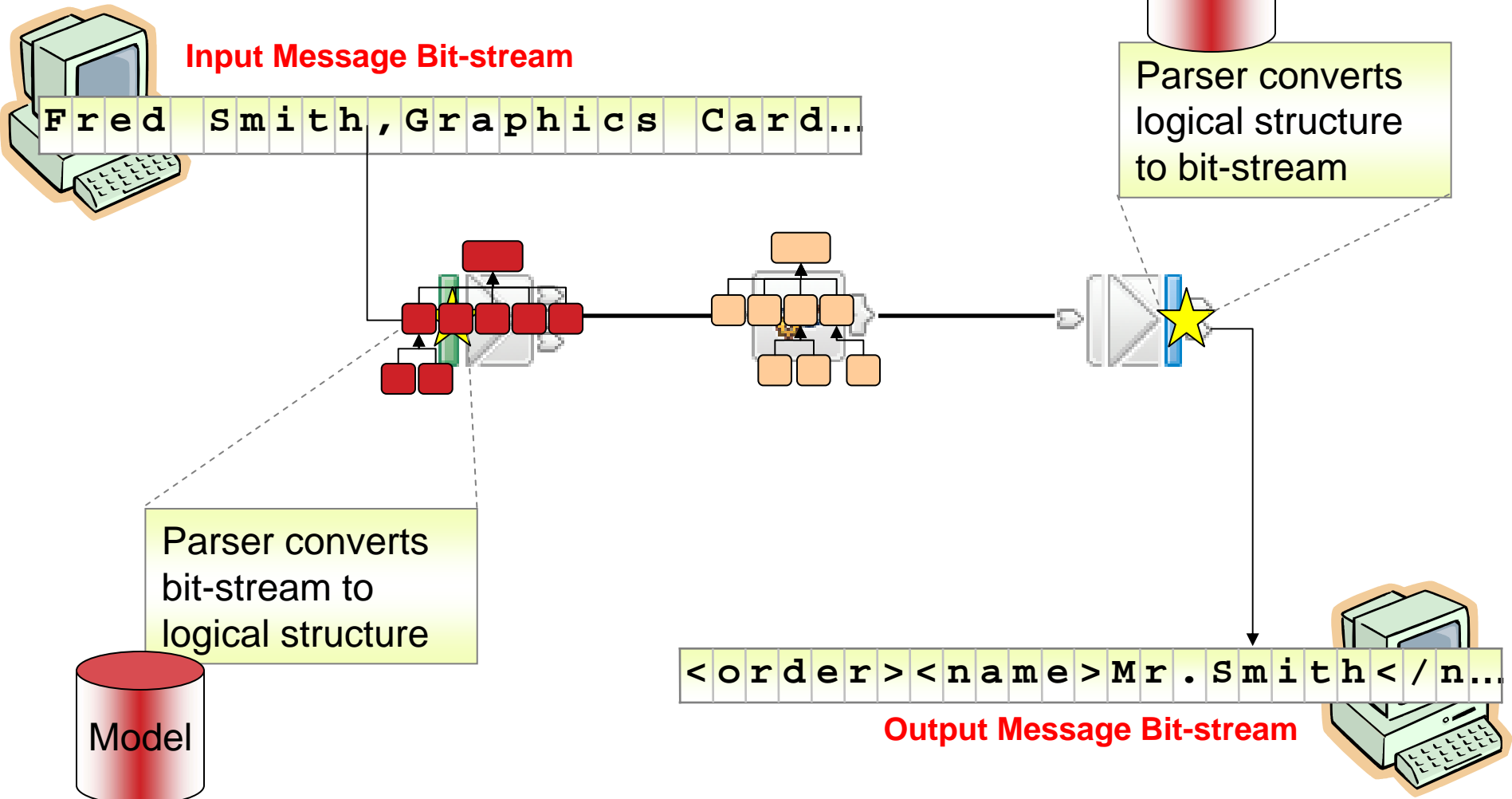
[Proxy flow](#)
[Error subflow](#)
[Log subflow](#)

Create New Instance

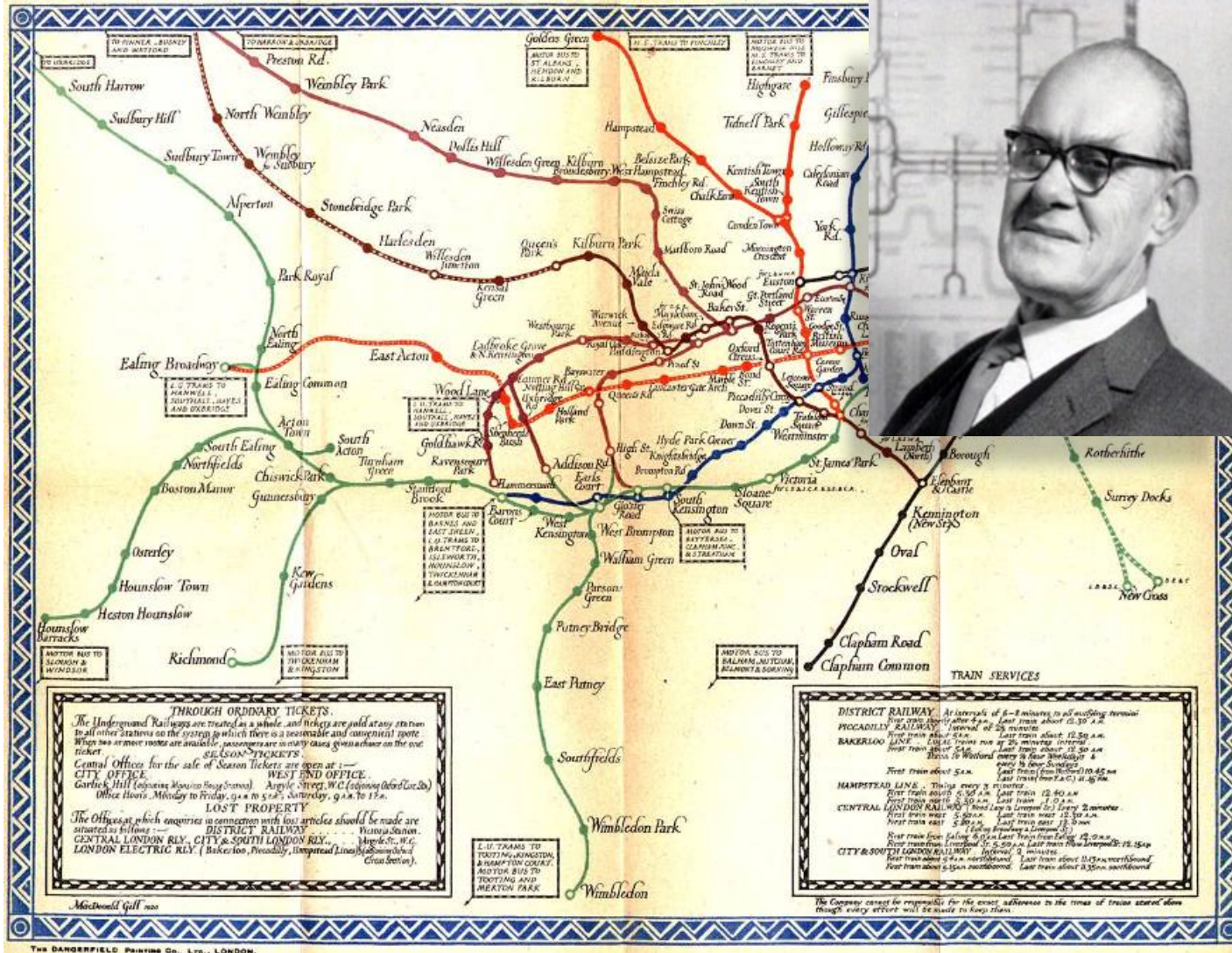
Properties Problems Deployment Log

Details Timestamp

Parsers

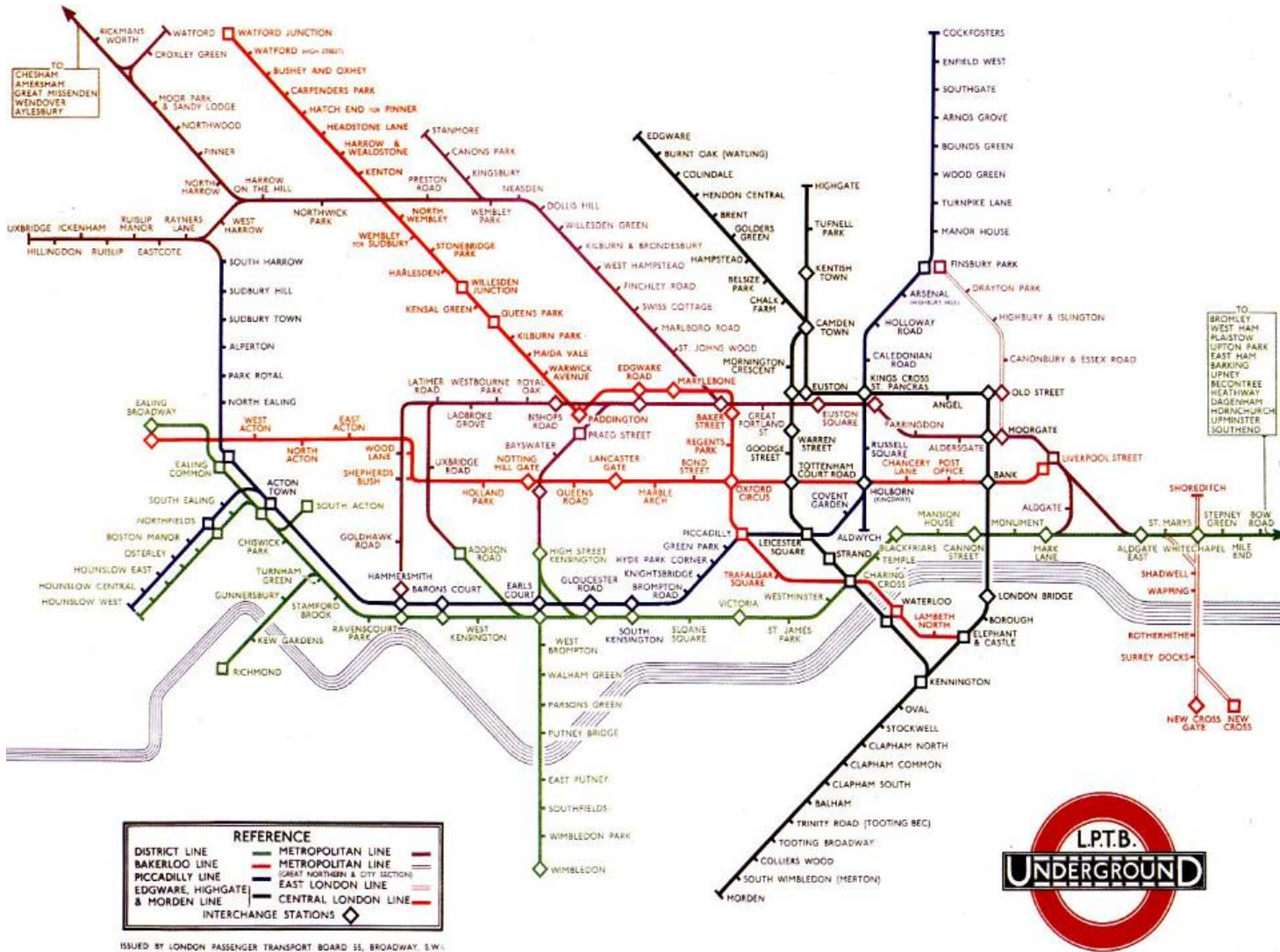


London Underground – Physical Model



© Transport for London

London Underground – Logical Model



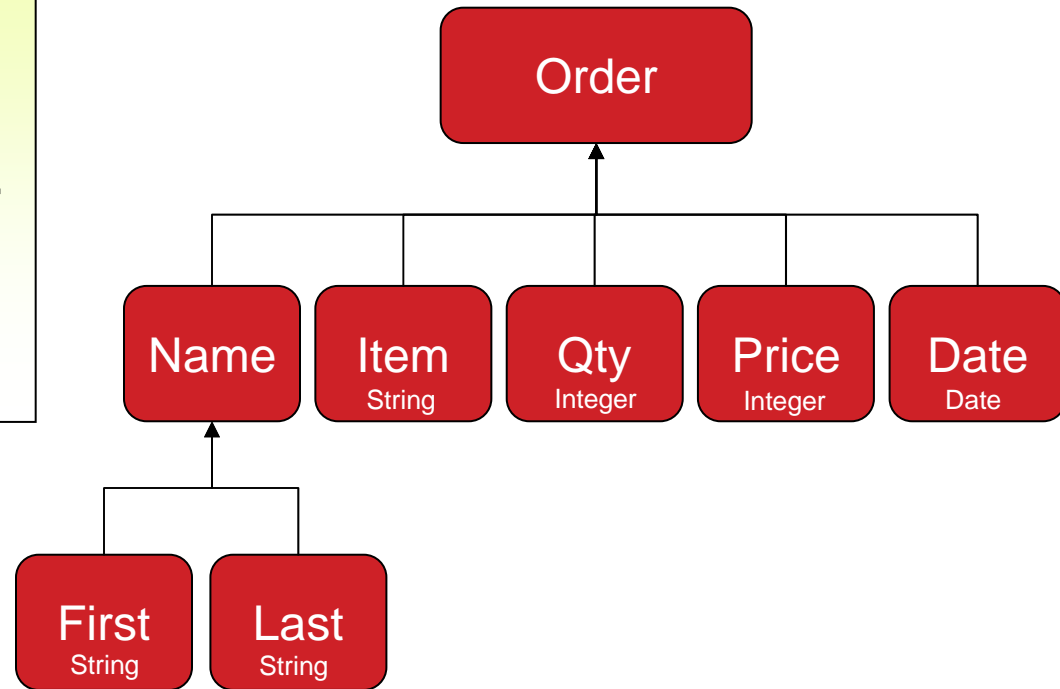
© Transport for London

Message Modelling

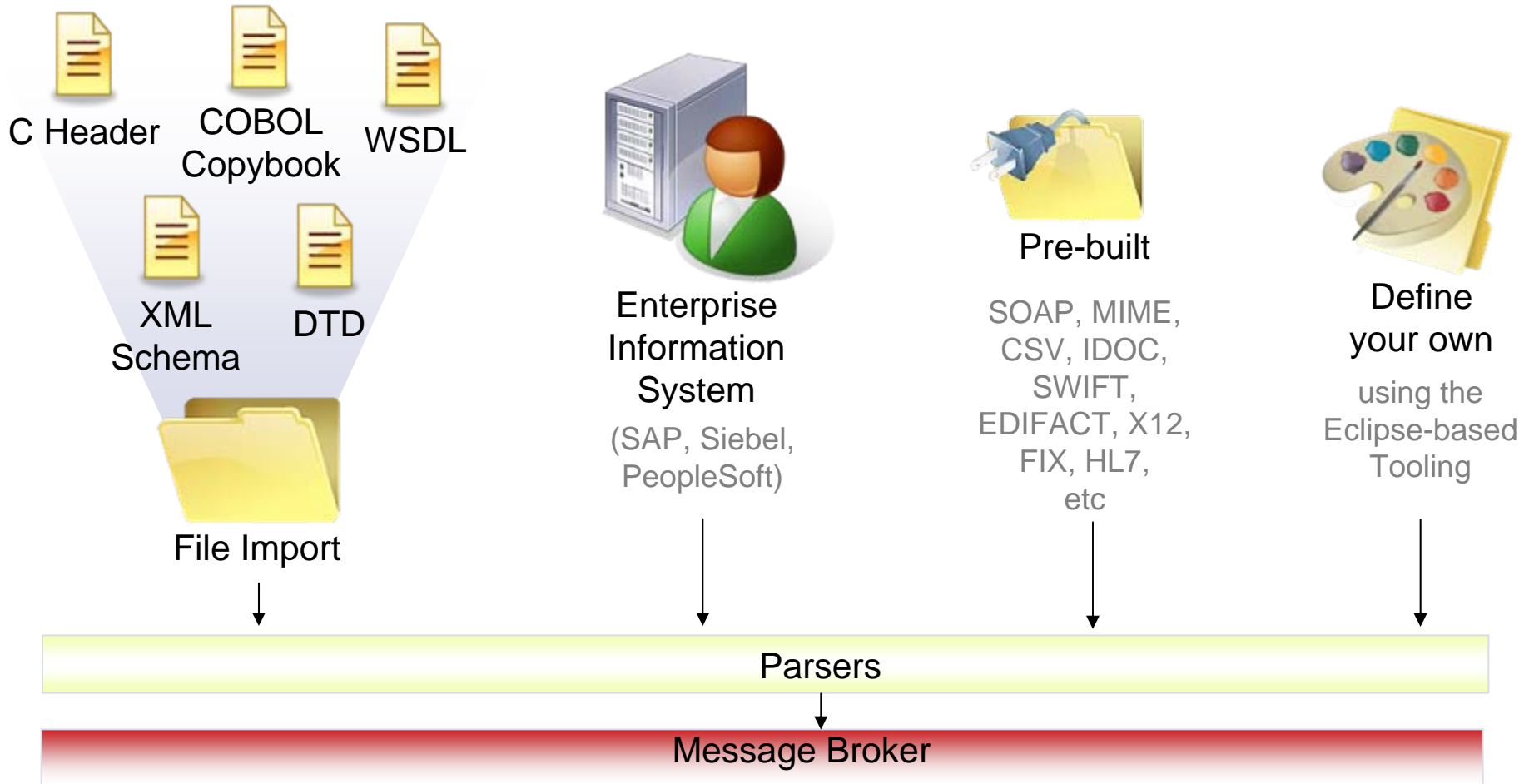


```
<order>
  <name>
    <first>John</first>
    <last>Smith</last>
  </name>
  <item>Graphics Card</item>
  <quantity>32</quantity>
  <price>200</price>
  <date>07/11/09</date>
</order>
```

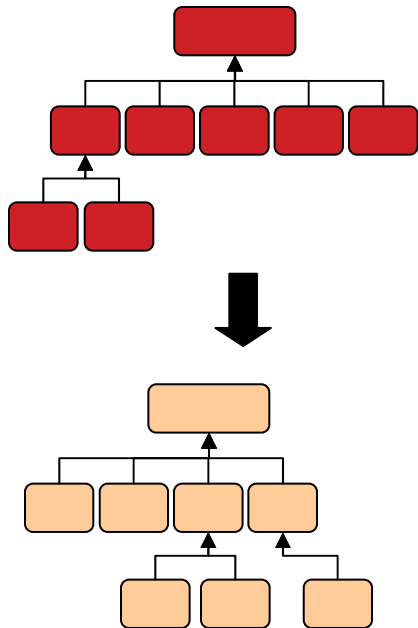
```
John, Smith, Graphics Card,
32, 200, 07/11/09
```



Creating Message Models



Powerful Message Transformation Options



Mapping

- Graphical, easy to use
- Drag and Drop fields, apply functions



Java Compute

- Embed Java programs
- Ability to use XPath for tree access



Compute

- Describe powerful transformations quickly
- Uses SQL-based language (ESQL)



XSL Transform

- Convert XML to anything
- Uses standard XSL Style sheets



PHP Compute

- Transform using PHP scripts
- PHP 5.2 compliant



.NET Compute

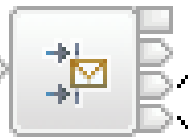
- Use any of the 40+ .NET languages (e.g. C#, VB.NET)
- Access COM objects

Easily Address Message Elements



JavaCompute

```
public class jcn extends MbJavaComputeNode {
    public void evaluate(MbMessageAssembly assembly) throws MbException {
        ...
        String lastName =
            (String)assembly.getMessage().evaluateXPath("/Body/Order/Name/Last");
        ...
    }
}
```



Route

Route Node Properties - Route

Filter table*

Filter pattern	Routing output terminal
\$Body/Order/Price > 1000	BigSpenders
\$Body/Order/Price < 200	Cheapskates

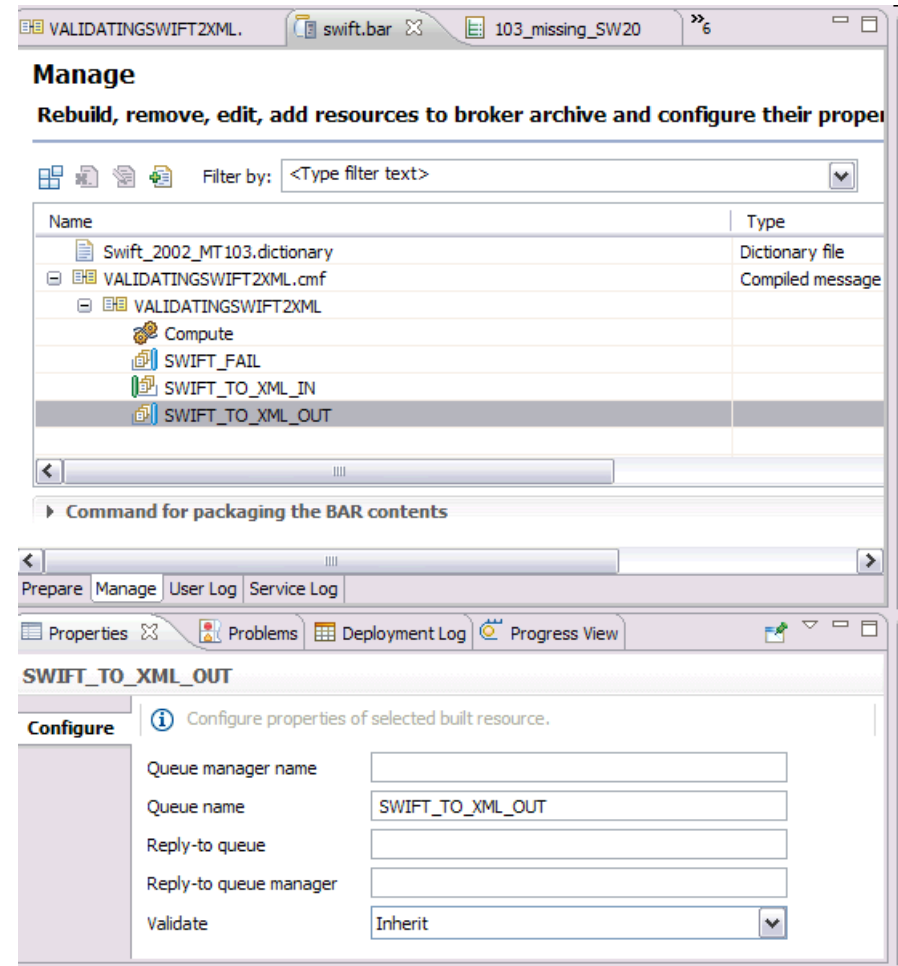


DataInsert

```
IF Body.Order.Date < '2008/01/01' THEN
    INSERT INTO Database.OldOrders (LastName,Item,Quantity)
    VALUES (Body.Order.Name.Last,
            Body.Order.Item,
            Body.Order.Quantity);
ENDIF;
```


Broker Archives (Bar files)

- Deployment container for Broker resources
- Allows overrides to be applied to properties at deploy time
 - Move between environments without rebuilding
- Is just a zip file containing the Broker artifacts
 - Similar to a .war or .ear file for app server



The screenshot shows the SHARE IDE interface for managing a broker archive. The top pane shows a tree view of resources under the 'Manage' view. The selected resource is 'SWIFT_TO_XML_OUT'. The bottom pane shows the configuration properties for this resource.

Name	Type
Swift_2002_MT103.dictionary	Dictionary file
VALIDATINGSWIFT2XML.cmf	Compiled message
VALIDATINGSWIFT2XML	
Compute	
SWIFT_FAIL	
SWIFT_TO_XML_IN	
SWIFT_TO_XML_OUT	

Command for packaging the BAR contents

SWIFT_TO_XML_OUT

Configure

Queue manager name:

Queue name:

Reply-to queue:

Reply-to queue manager:

Validate:

Application and Libraries



Application

- means of encapsulating resources to solve a specific connectivity problem
- application can reference one or more libraries



Library

- a logical grouping of related routines and/or data
- libraries help with reuse and ease of resource management
- library can reference one or more libraries

Feature of Application and Libraries

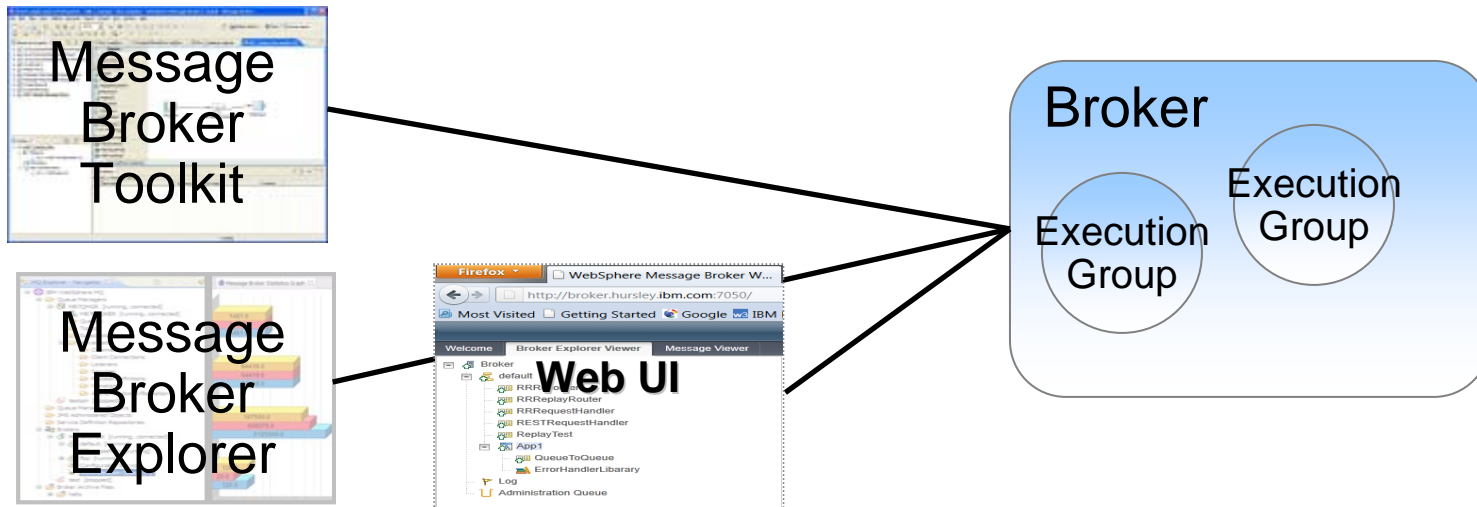


- Applications promote encapsulation and isolation
 - Typically contain “main” message flows and dependent resources
 - ESQL, Java, Maps, Message models, subflows, Adapter files, etc.
 - dependent resources could live in referenced libraries
 - Multiple applications can be packaged into a single BAR
 - Multiple applications can be deployed to an execution group
 - Visibility of resource restricted to containing application
 - Referenced libraries are deployed inside application container (by copy)



- Libraries facilitate re-use and simplify resource management
 - Typically contain reusable helper routines and resources
 - Subflows, ESQL, Java, Maps, Message models, Adapter files, etc.
 - Use multiple libraries to group related resources (e.g. by type or function)
 - Multiple applications can reference the same library
 - Each application gets its own copy of the library during package/deploy
 - Libraries are packaged as part of referencing application in the BAR
 - Library can reference other Libraries

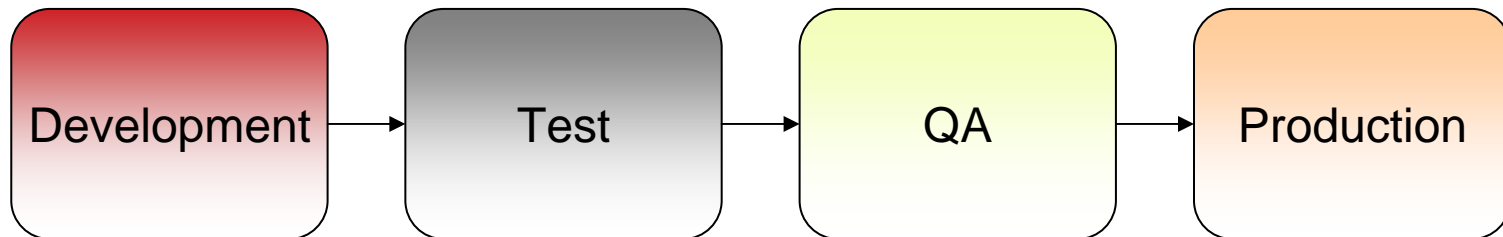
Architected for High Performance and Scalability



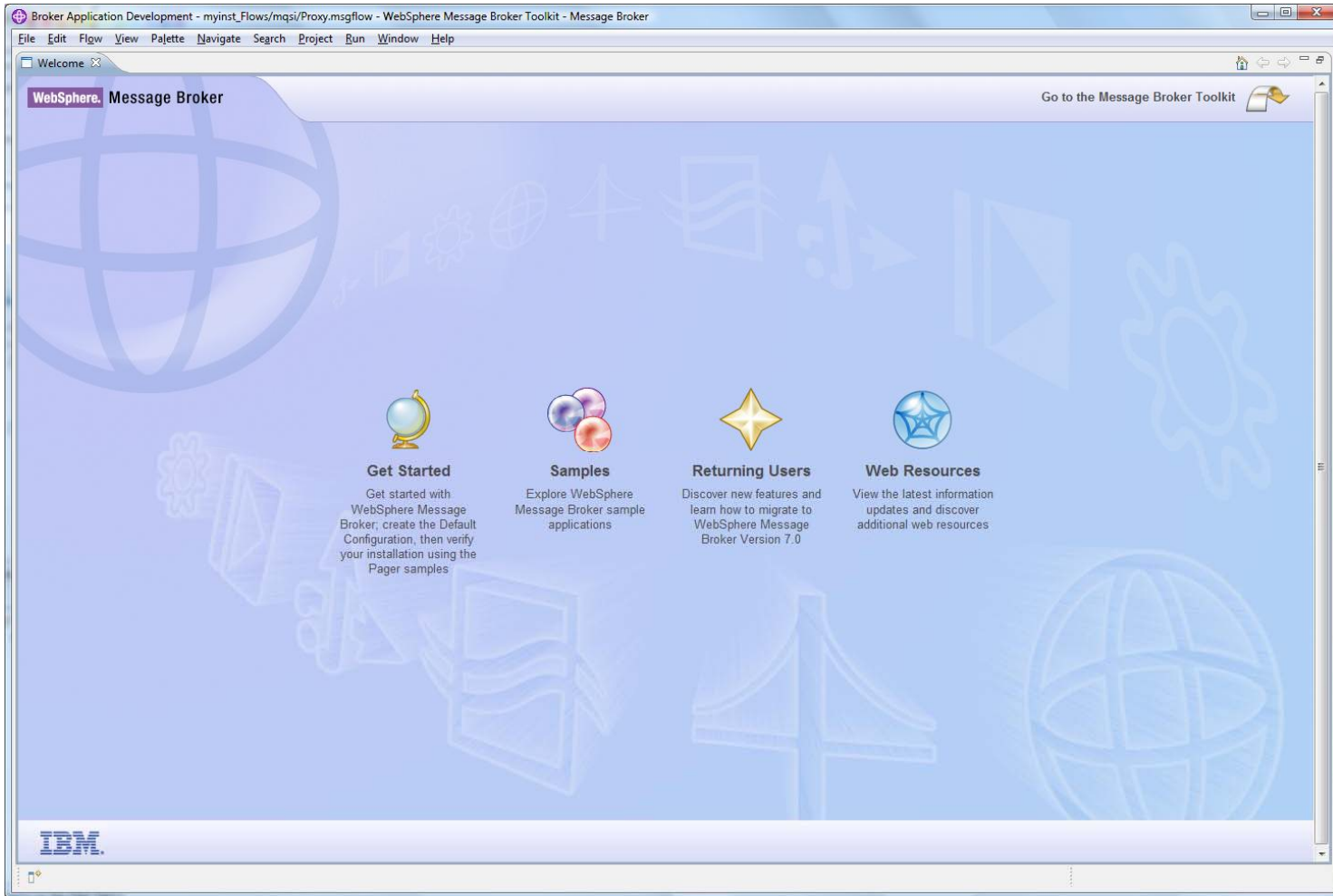
- Message Broker Toolkit
 - Development and Test Environment
 - Built on Rational Application Developer
- Message Broker Explorer
 - Advanced Administration Tool
 - Built on MQ Explorer
- Web UI
 - No install required
 - Built on Broker REST API
- Broker
 - Standalone runtime environment that runs message flows
 - Execution groups for isolation and scalability
 - Many different platforms
 - Builds on an MQ queue manager

Support for All User Roles and Environments

- Application Developer
 - Develops message flows, message models etc.
 - Unit Tests on local machine
 - Creates Broker Archive (BAR) files containing required artefacts
- Administrator
 - Customizes BAR for target environment (message flow properties including queues, database names etc.)
 - Deploys BAR to target broker
 - Broker management and operational control
 - Monitoring



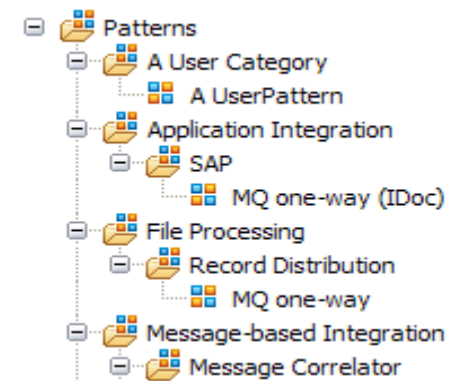
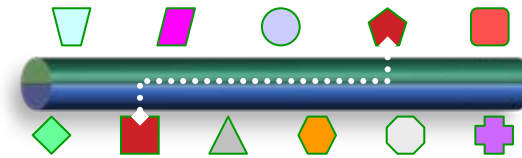
Demo



WebSphere Message Broker



- Universal Connectivity FROM anywhere, TO anywhere
 - Simplify application connectivity for a flexible & dynamic infrastructure
- Protocols, Transports, Data Formats & Processing
 - Supports a wide range of built-in transports, protocols & systems
 - MQ, JMS 1.1, HTTP(S), SOAP, REST, File (incl. FTP & FTE), Database, TCP/IP, MQTT...
 - CICS, IMS, SAP, SEBL, PeopleSoft, JDEdwards, SCA, CORBA, email...
 - Supports a broad range of data formats
 - Binary (C/COBOL), XML, CSV, JSON, Industry (SWIFT, EDI, HL7...), IDOCs, User Defined
 - Message Processors
 - Route, Filter, Transform, Enrich, Monitor, Publish, Decompose, Sequence, Correlate, Detect...
- Simple Programming with Patterns & Graphical Data Flows
 - Patterns for top-down, parameterized connectivity of common use cases
 - e.g. Service façades, Message processing, Queue2File...
 - IBM & User defined patterns for development reuse & governance
 - Graphical data flows represent application & service connectivity
 - Custom logic via graphical mapping, PHP, Java, ESQL, XSL & WTX
- Extensive Management, Performance & Scalability
 - Extensive Administration & Systems Management facilities for developed solutions
 - Wide range of operating system & hardware platforms supported, including virtual & WCA Hypervisor
 - High performance transactional processing, additional vertical & horizontal scalability
 - Deployment options include Trial, Remote Deployment, Starter, Entry, Enterprise
- Connectivity Packs for Industry Specific Content
 - Connectivity Pack for Healthcare includes HL7 Connectors, Patterns & Tooling



This was session 11385 - The rest of the week



	Monday	Tuesday	Wednesday	Thursday	Friday
08:00					Free MQ! - MQ Clients and what you can do with them
09:30	Clustering – the easier way to connect your Queue Managers	MQ on z/OS – vivisection	The Dark Side of Monitoring MQ - SMF 115 and 116 record reading and interpretation		
11:00		Diagnosing problems for Message Broker	Lock it down - WebSphere MQ Security	Using IBM WebSphere Application Server and IBM WebSphere MQ Together	Spreading the message – MQ pubsub
12:15	Highly Available Messaging - Rock solid MQ	Putting the web into WebSphere MQ: A look at Web 2.0 technologies	The Doctor is In and Lots of Help with the MQ family - Hands-on Lab		
01:30	WebSphere MQ 101: Introduction to the world's leading messaging provider	What's new in the WebSphere MQ Product Family	Extending IBM WebSphere MQ and WebSphere Message Broker to the Cloud	MQ Performance and Tuning on distributed including internals	
03:00	First steps with WebSphere Message Broker: Application integration for the messy	What's new in Message Broker V8.0	Under the hood of Message Broker on z/OS - WLM, SMF and more	The Do's and Don'ts of z/OS Queue Manager Performance	
04:30	The MQ API for Dummies - the Basics	What the **** is going on in my Queue Manager!?	Diagnosing problems for MQ	Shared Q using Shared Message Data Sets	
06:00			For your eyes only - WebSphere MQ Advanced Message Security	MQ Q-Box - Open Microphone to ask the experts questions	

Complete your sessions evaluation online at SHARE.org/AnaheimEval



Copyright and Trademarks

© IBM Corporation 2012. All rights reserved. IBM, the IBM logo, ibm.com and the globe design are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml. Other company, product, or service names may be trademarks or service marks of others.

Session 11385: QR

