



WebSphere Software

IBM WebSphere ESB V6.0.1

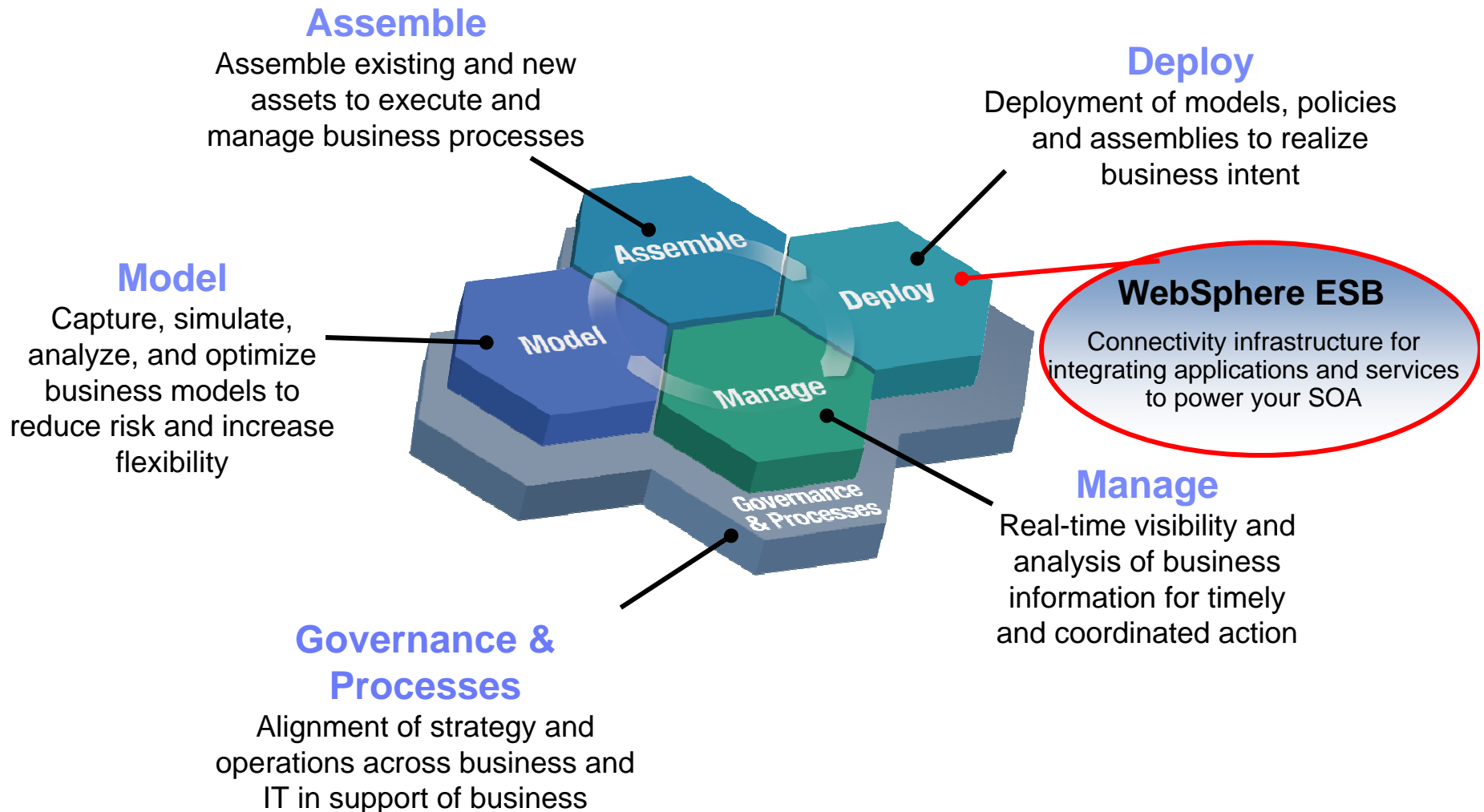
Technical Product Overview

SOA on your terms and our expertise

ON DEMAND BUSINESS™

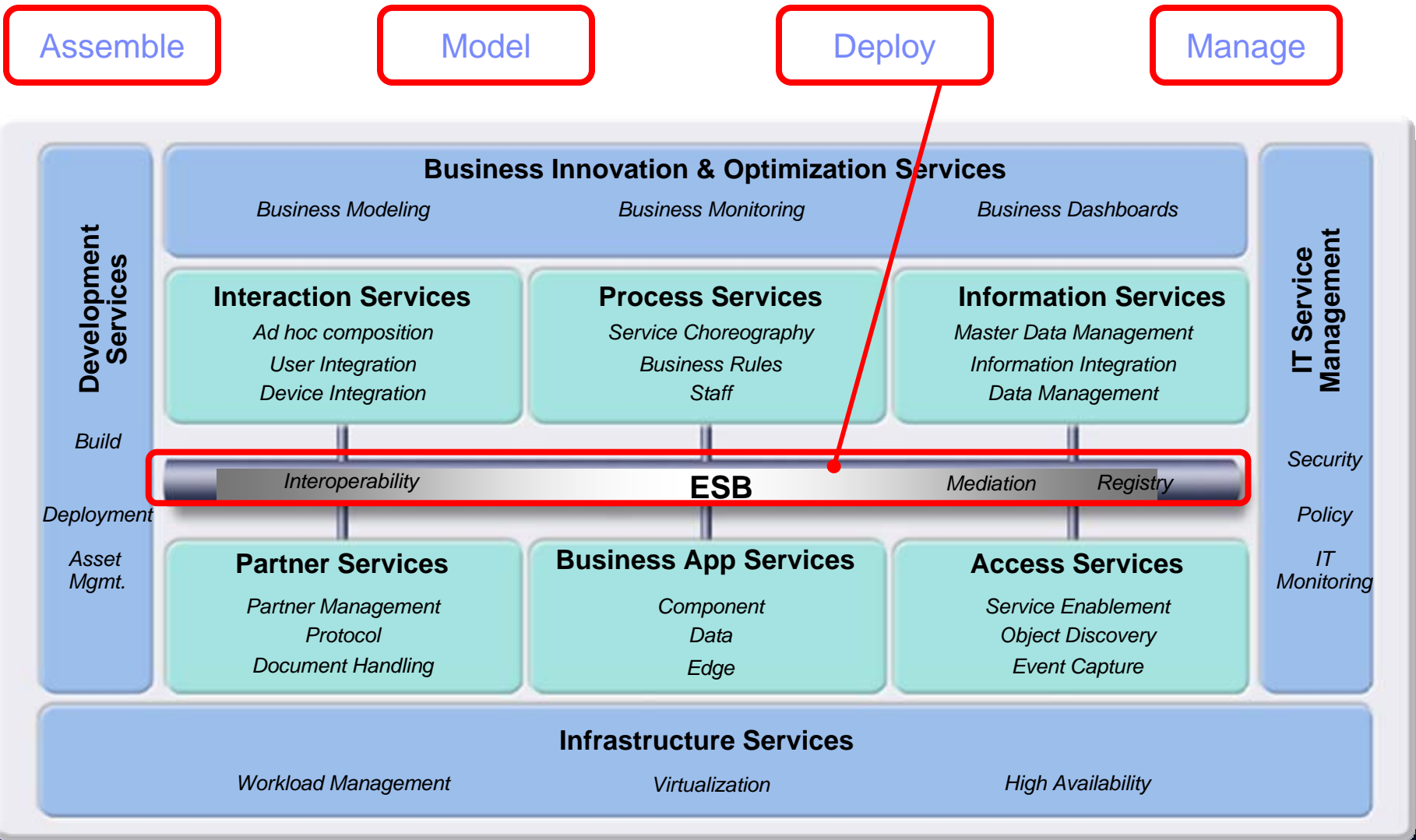
© 2005 IBM Corporation

The SOA Lifecycle .. For Flexible Business & IT



SOA Reference Model

Comprehensive services in support of your SOA



What is an Enterprise Service Bus?

An Enterprise Service Bus (ESB) is a flexible connectivity infrastructure for integrating applications and services.

An ESB powers your SOA by reducing the number, size, and complexity of interfaces.

An ESB performs the following between requestor and service

- **ROUTING** messages between services
- **CONVERTING** transport protocols between requestor and service
- **TRANSFORMING** message formats between requestor and service
- **HANDLING** business events from disparate sources



WebSphere ESB and WebSphere Message Broker

ESB:
WebSphere ESB

New product

Advanced ESB:
WebSphere Message Broker

New version

Web Services connectivity and data transformation

Universal connectivity and data transformation

HTTP JMS

WebSphere MQ

Web Services XML

WebSphere Adapters

HTTP JMS WebSphere MQ
Web Services XML WebSphere Adapters

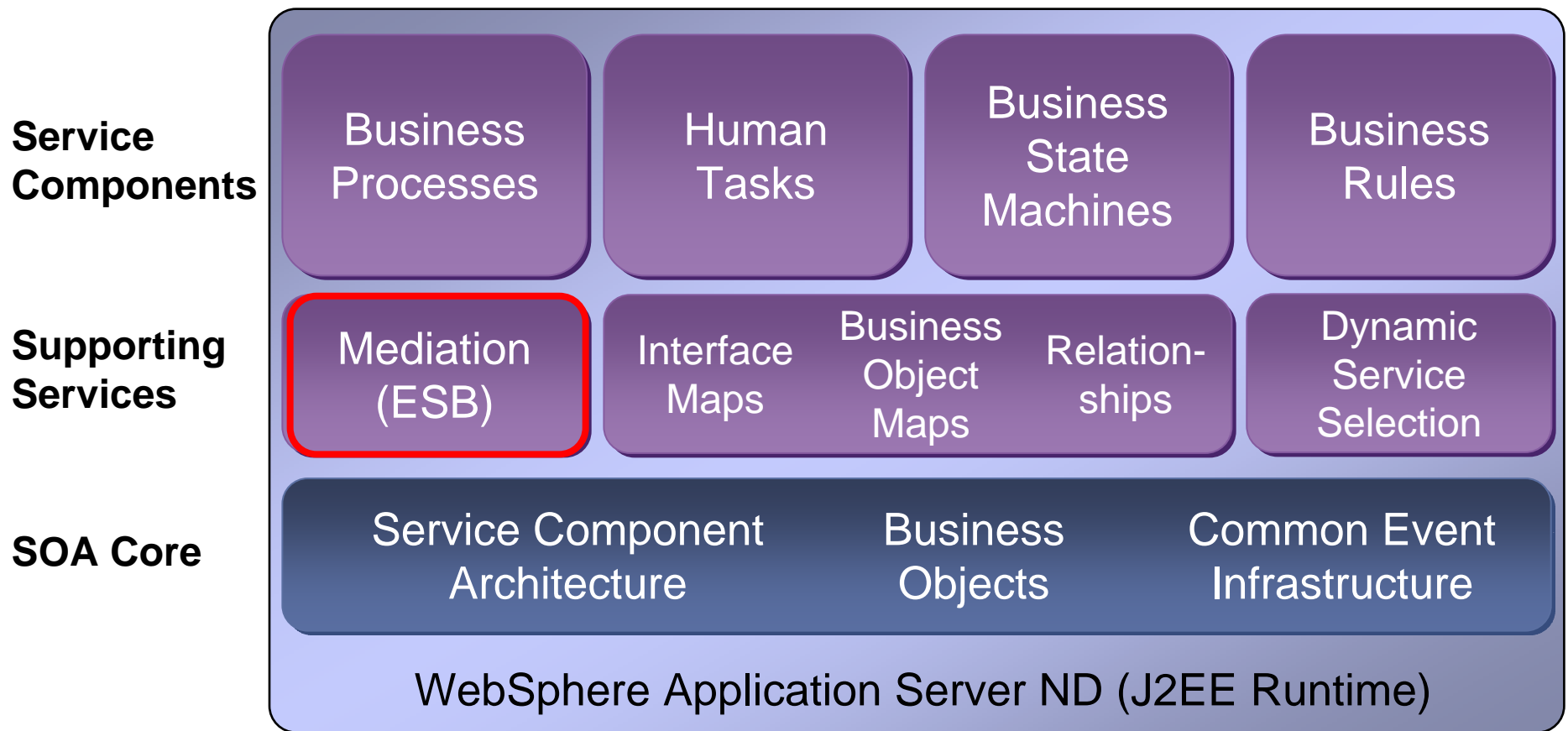
Plus the following:

Weblogic JMS® Biztalk® TIBCO Rendezvous®
MQe Multicast Tuxedo® FTP TIBCO EMS JMS®
COBOL HIPAA EDI-FACT HL7 SonicMQ JMS®
Copybook ACORD Real-time IP AL3 Word/Excel/PDF
SWIFT FIX ebXML EDI-X.12 MQTT Custom Formats

Customers face a range of ESB requirements.

As a result, any given project might require an ESB or an Advanced ESB... OR BOTH.

WebSphere Process Server v6 - Components

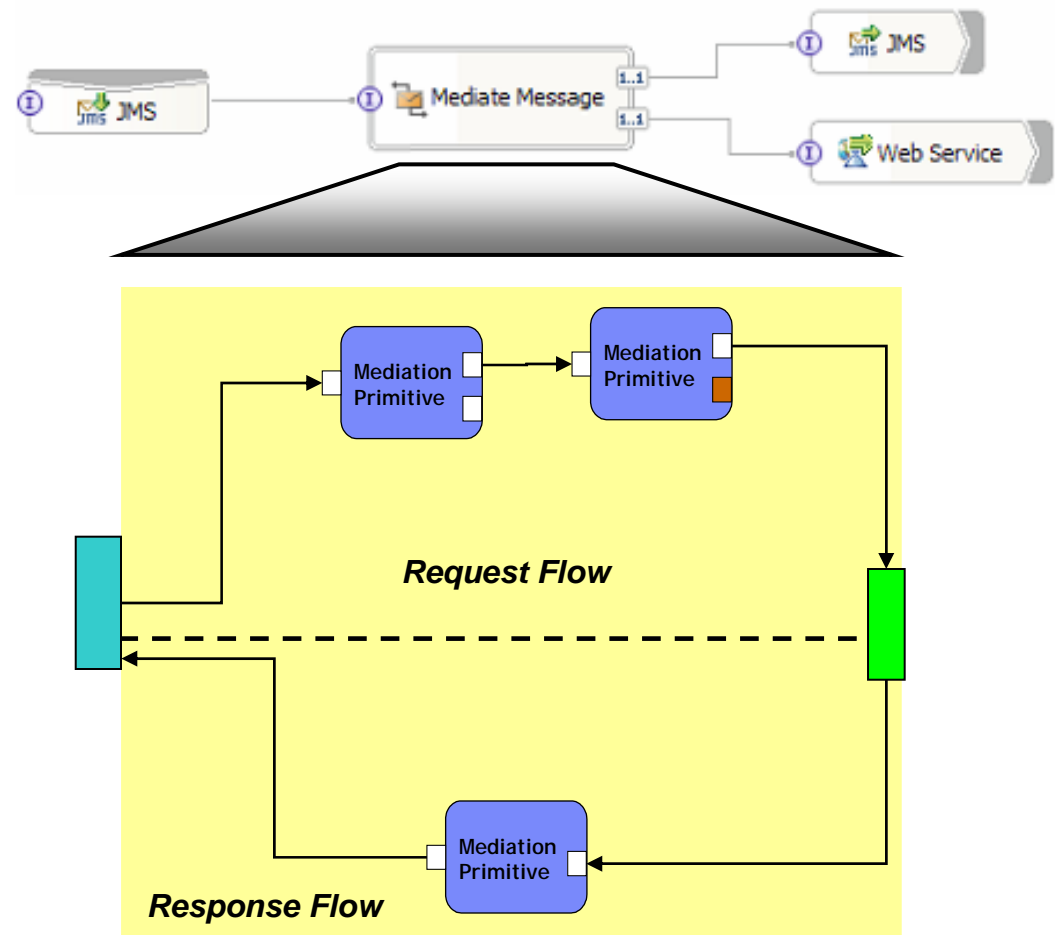


How WebSphere ESB uses SCA

- New component type
 - Mediation flow component
- ESB introduces mediation primitives at the subcomponent level:
 - Notice the difference in programming models between SCA components and mediation primitives
 - Mediation primitives work on Service Message Object (SMO)
- Introduces SCA admin commands related to SCA modules
- User can add new custom mediation primitives implemented as a Java SCA components

ESB Mediation Component

- Provide the Implementation of mediation “logic”
 - “flows” that operate on messages/events as they are processed by the system
 - Operate on both One-Way and Request-Response interactions
- Pre-Supplied primitives allow flows to be visually composed
 - XSLT Transformation
 - Message Logger
 - Message Filter
 - Fail
 - Stop
 - Database Lookup
 - Custom (Java) Component
 - CEI Emitter (Post GA)



The Service Message Object

- Import and export bindings interact with some specific service provider type
- Uses data structure called the Service Message Object
 - Elements
 - Business Object
 - SDO representation of the 'application' data
 - Binding-specific information
 - E.g. SOAP headers, JMS headers
 - Context information
 - Used to pass information from one element of a flow to another

Integration Developer: Typical Task Flow



Integration
Developer

- 1. Identifies the service endpoints that need to be integrated**
 - Service requesters and Service providers
- 2. Asserts the basic connectivity between these endpoints**
 - Which requester operation is linked to which provider operation
- 3. Decides on the mediation function required to allow endpoints to communicate effectively**
 - Selects from supplied function
 - Customizes selected function
 - Optionally: constructs and integrates custom-written function
- 4. Tests and Debugs the mediation composed/customized function**
- 5. Deploys the result to the runtime**



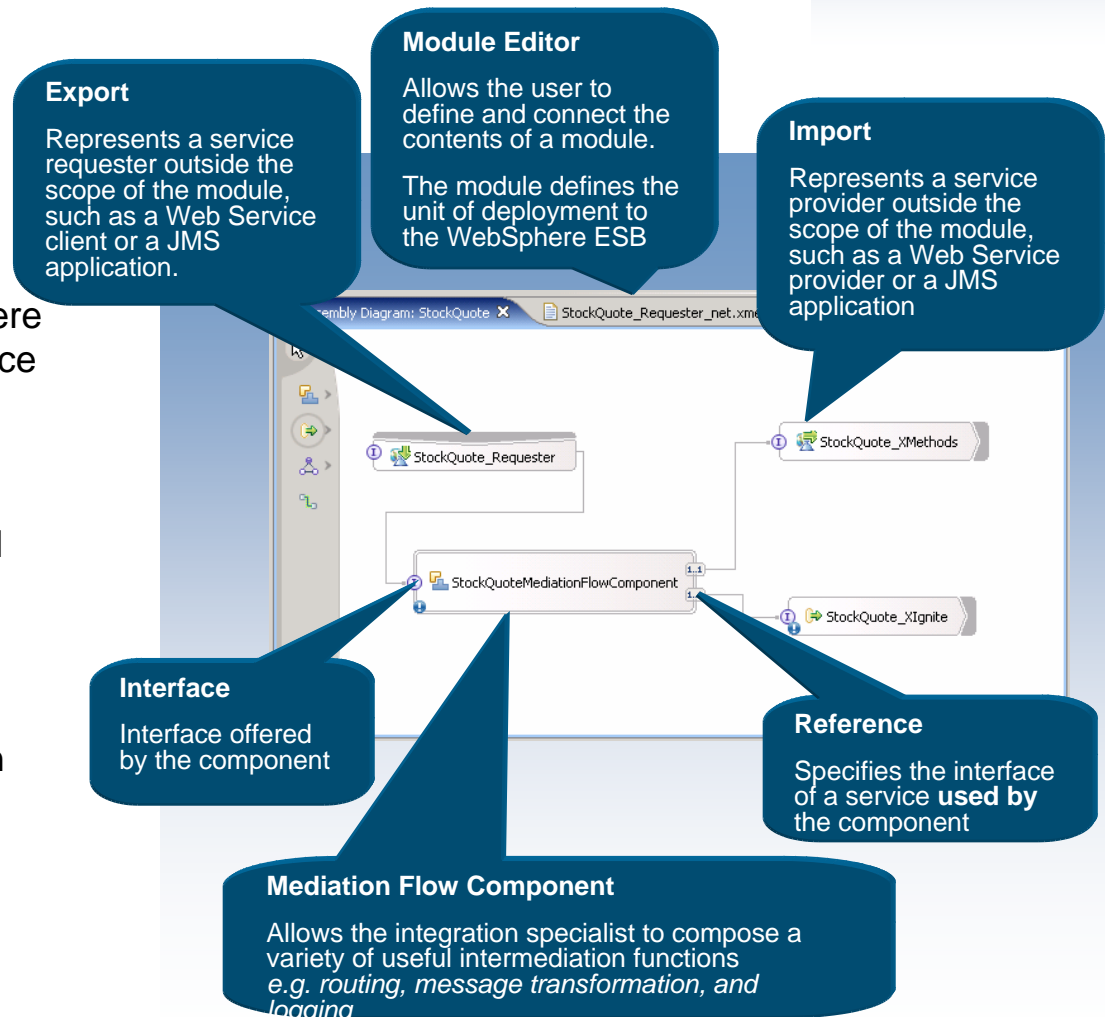
1. Integration developer specifies the service endpoints that need to be integrated

- **Uses the 'module editor' to construct a mediation module**

- specifies how a subset of WebSphere ESB's service requesters and service providers interact

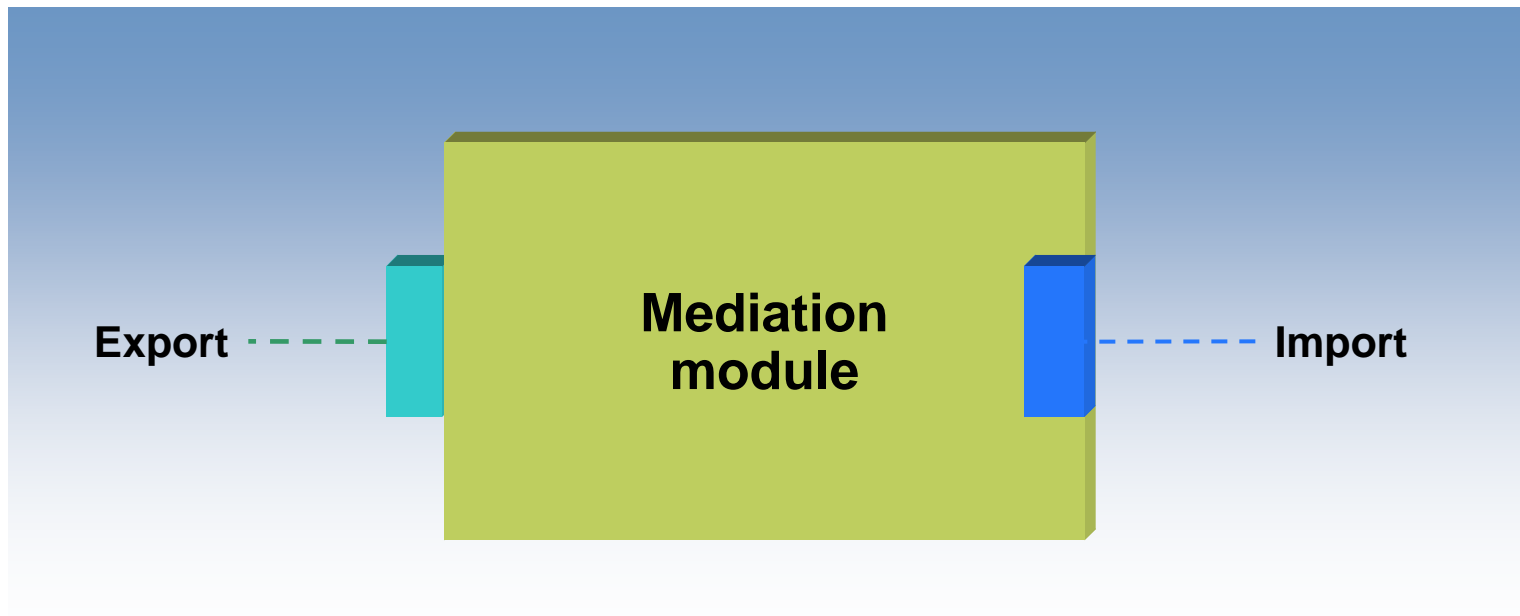
- **Within the module**

- Service requesters are represented as 'exports'
- Service providers are represented as 'imports'
- The integration (mediation) function is represented as a 'mediation flow component'
- Imports and exports are connected to the mediation flow component



Concepts: Mediation Module





- **Interactions with external service requesters and providers defined by imports and exports**
 - Import/export interfaces are defined using the Web Services Description Language (WSDL)
 - Which may contain several *service operations*
 - Different kinds of requester and provider are made available via different *bindings* for the imports and exports



Concepts: Import and Export bindings

- Each **binding** enables an interaction with a particular kind of service requester or service provider
- **WebSphere ESB provides support for**
 - JMS Binding
 - JMS 1.1 provided by WebSphere Platform Messaging
 - can exploit a variety of transports
 - > TCP/IP, SSL, HTTP(S)
 - Allows interoperation with the WebSphere family
 - > WAS, WebSphere MQ, WebSphere Message Broker
 - Web Services
 - SOAP/HTTP, SOAP/JMS, WSDL 1.1
 - Service Registry – UDDI 3.0
 - WS-Security, WS-Atomic Transactions
 - WebSphere Adapters
 - Support for a wide range of application environments
 - Ariba, PeopleSoft, SAP, Siebel, etc.
- **WebSphere ESB also provides a built-in ‘default binding’**
 - Used for module to module communication
 - Supports both synchronous and asynchronous communication
 - WebSphere ESB supports update this binding via the admin console allowing module to module connectivity to be changed

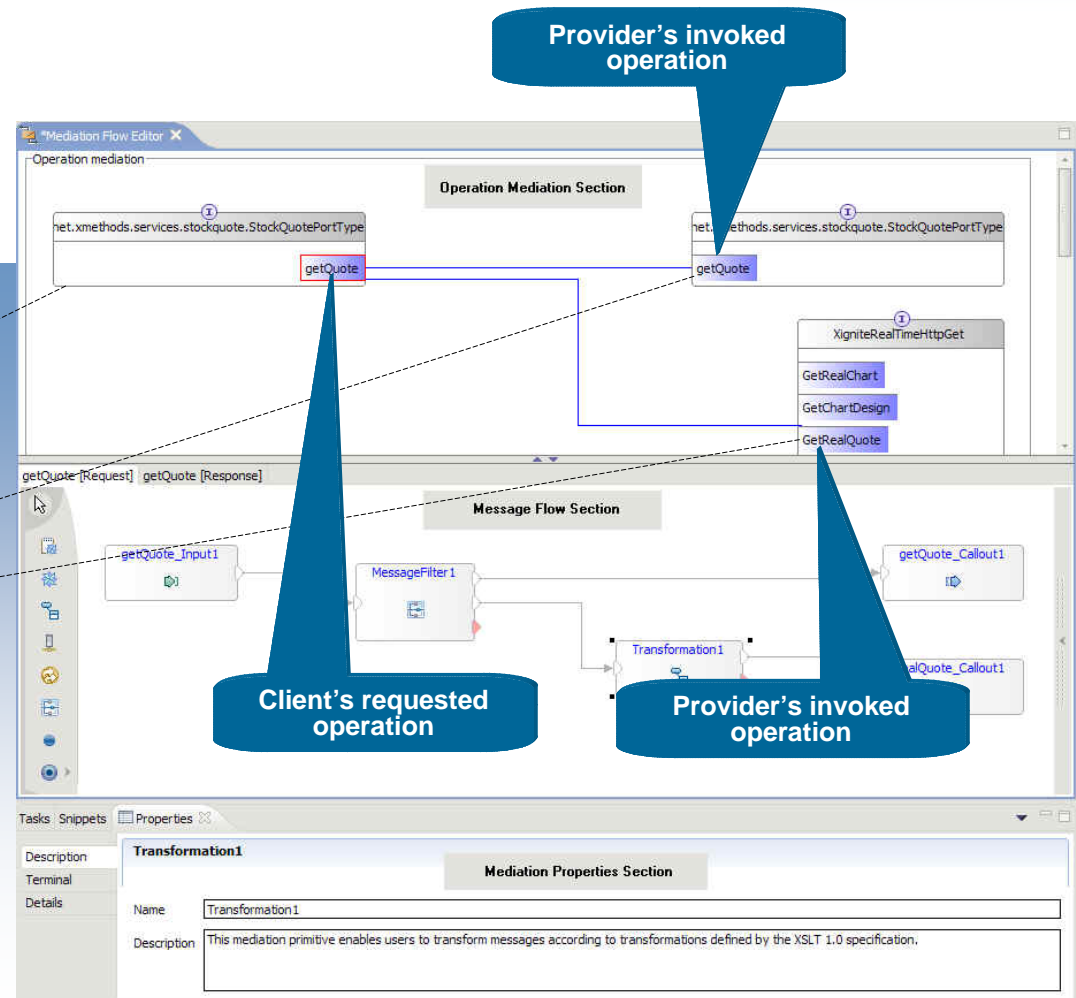
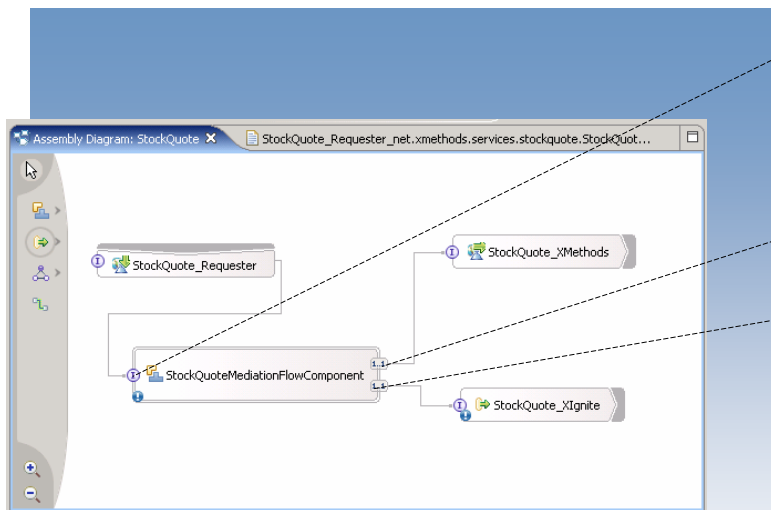
Supported Bindings

-  JMS Binding
-  Web Services binding
-  Default binding
-  WBI adapter binding



2. Integration Developer asserts the basic connectivity between these endpoints

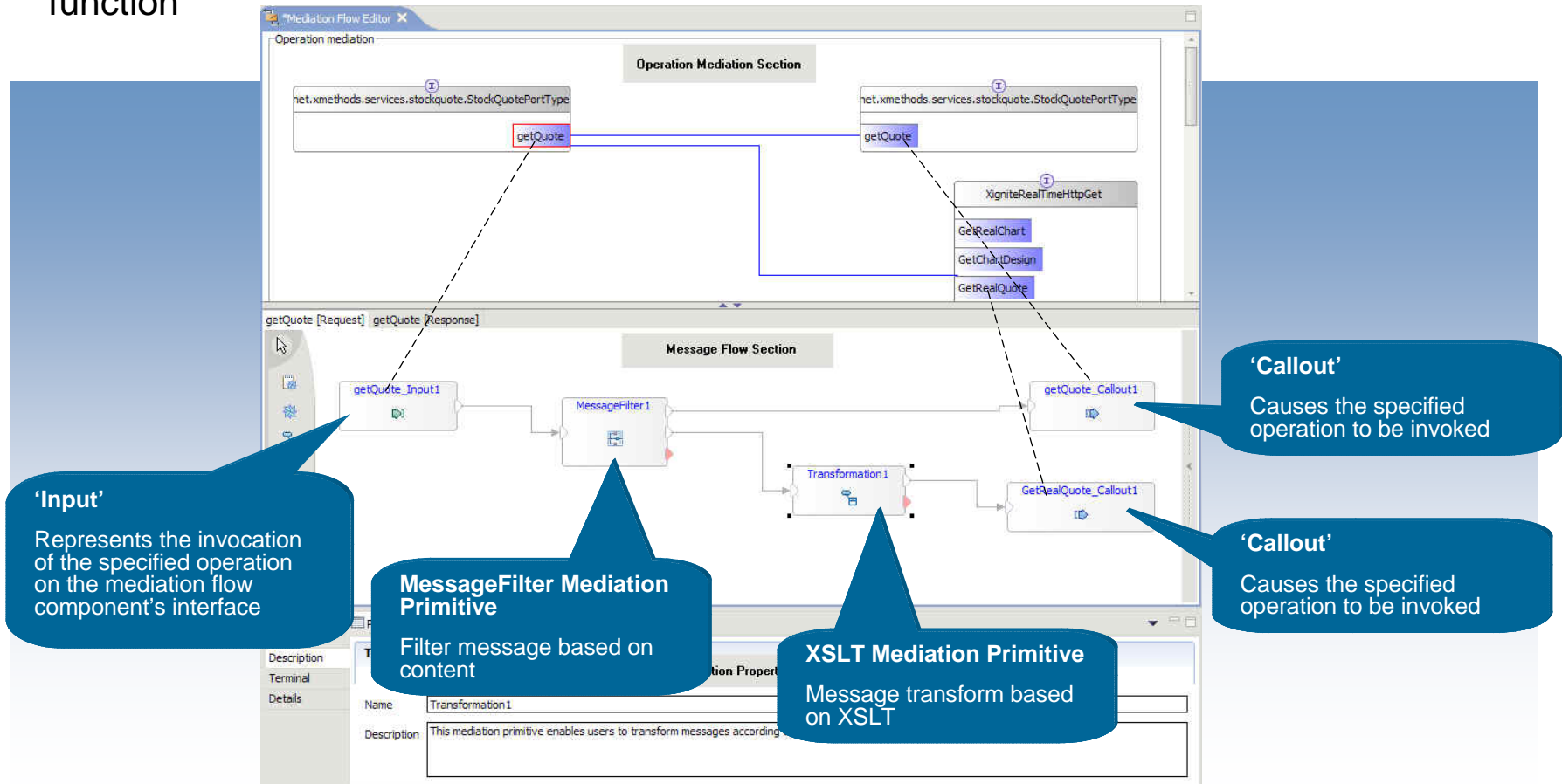
- The integration developer uses the *mediation tooling* to specify the essential connectivity between a requester and one or more service providers





3. Decides on the mediation function required to allow endpoints to communicate effectively

- The integration developer constructs a *mediation flow* for the service request by selecting and connecting *mediation primitives* from supplied function





4. Integration developer customizes the elements of the mediation flow

- e.g. Customizes the XSLT transform mediation primitive by using mapping tool to construct an XSLT transform

The structure of the message is represented graphically

A properties view is provided where the details of the mapping can be specified

Target	Source	Applied Function/Grouping
StockQuoteXIgnite_GetRealQuote...	StockQuoteXMethod_getQuoteRequest1	
tns:GetRealQuote		
Exchange		string
Symbol [0..1]	symbol	
IncludeBidAsk		boolean

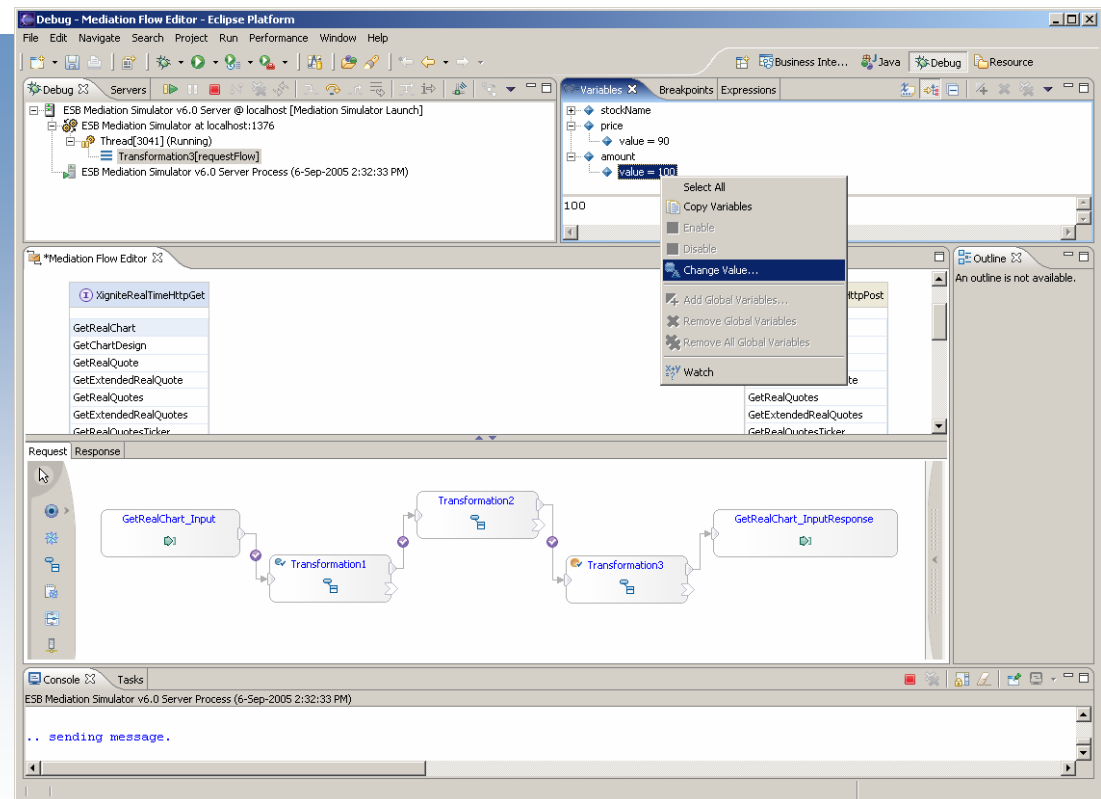
Define functions that apply to the mapping



5. Debugs the composed/configured mediation function: WID Mediation Visual Debug

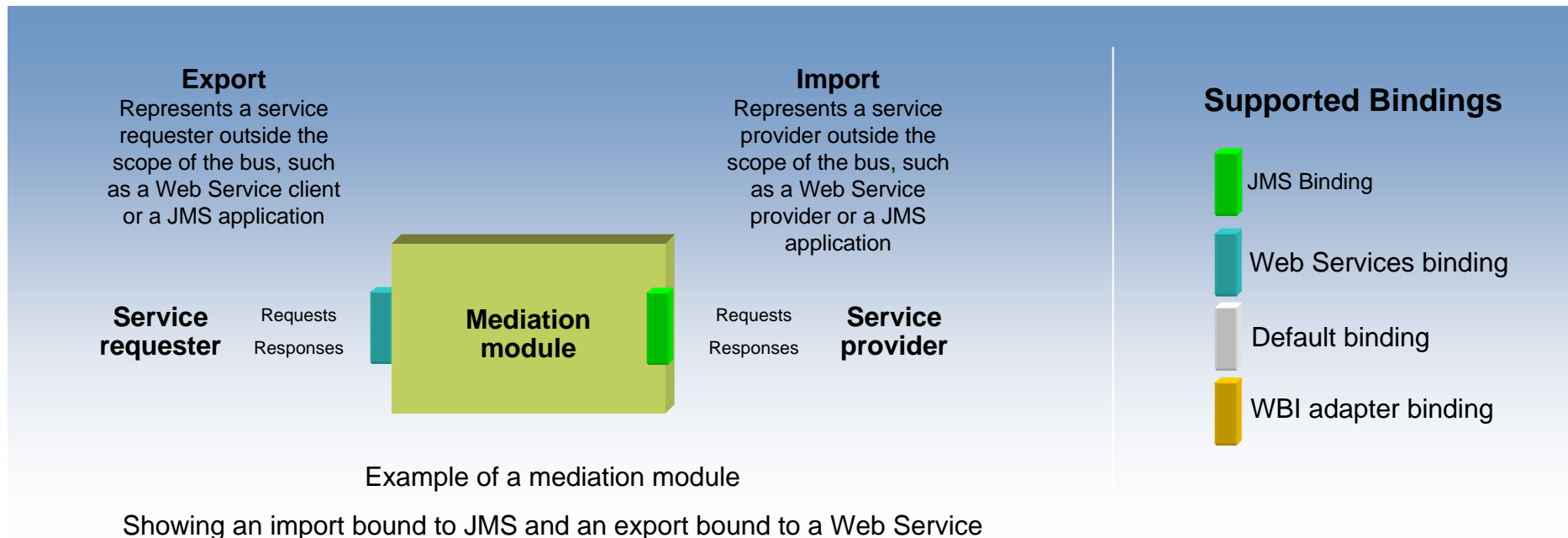
Use the visual debugging tools to debug a solution

- Debug mediation flows using an in-place visual debugger
- Breakpoints can be added, step into, through, or over areas of interest while inspecting the values of the messages



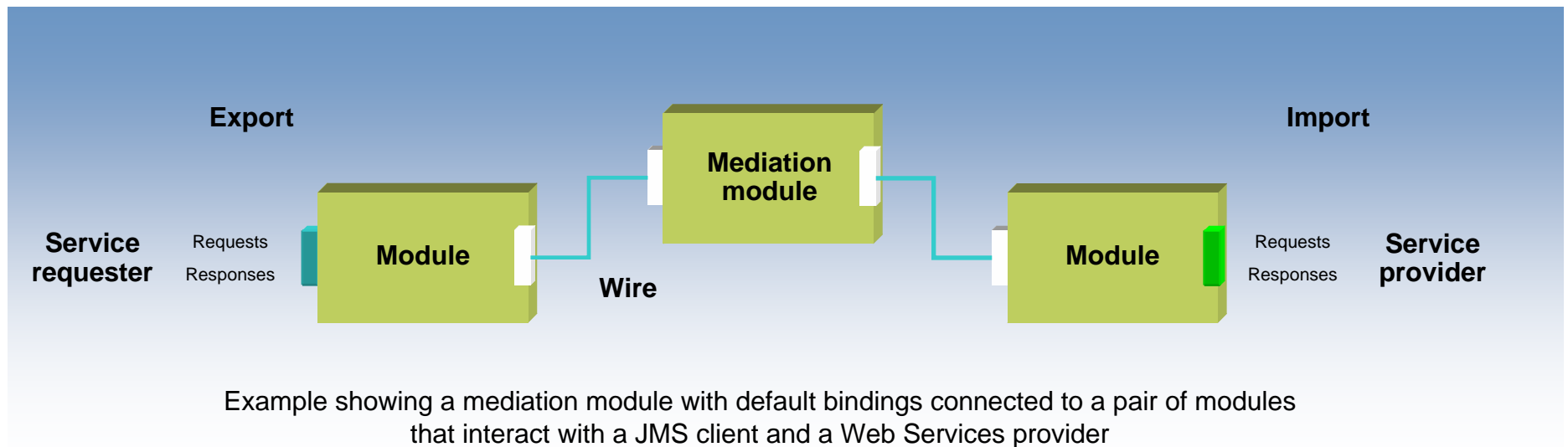
Integration Model: Approach 1

- **Mediation module is interposed between a service requester and the ultimate service provider**
 - It operates on all service interactions (both requests and responses)
 - Service interfaces are defined using the Web Services Description Language (WSDL)
- **Mediation module also encompasses interactions with external service endpoints**
 - Used when the Integration Specialist needs fine control over external interactions



Integration Model: Approach 2

- In this approach, the Mediation module uses only default bindings
- Communication with service requesters and providers is moved to separate modules
- This enables the solution administrator to 'rewire' to different requesters/provider
- This approach is used when the solution administrator needs more control over which service providers and mediation modules are used



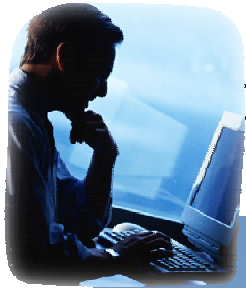
StockQuote Scenario

A financial services company is planning a phased rollout of an interactive web-based stock market service to its customers using Servlet technology. This service will include a simple stock quote service built using an Enterprise Service Bus (ESB) infrastructure. The company wants to differentiate itself from its competition and has chosen to do this by offering tiered levels of service and by proactively offering real-time stocks to their “gold” customers (customers who pay a subscription).

An ESB is used as the basis for deployment of these capabilities so as to allow the company to respond in a flexible and non-disruptive way to the changing requirements which are expected to arise.

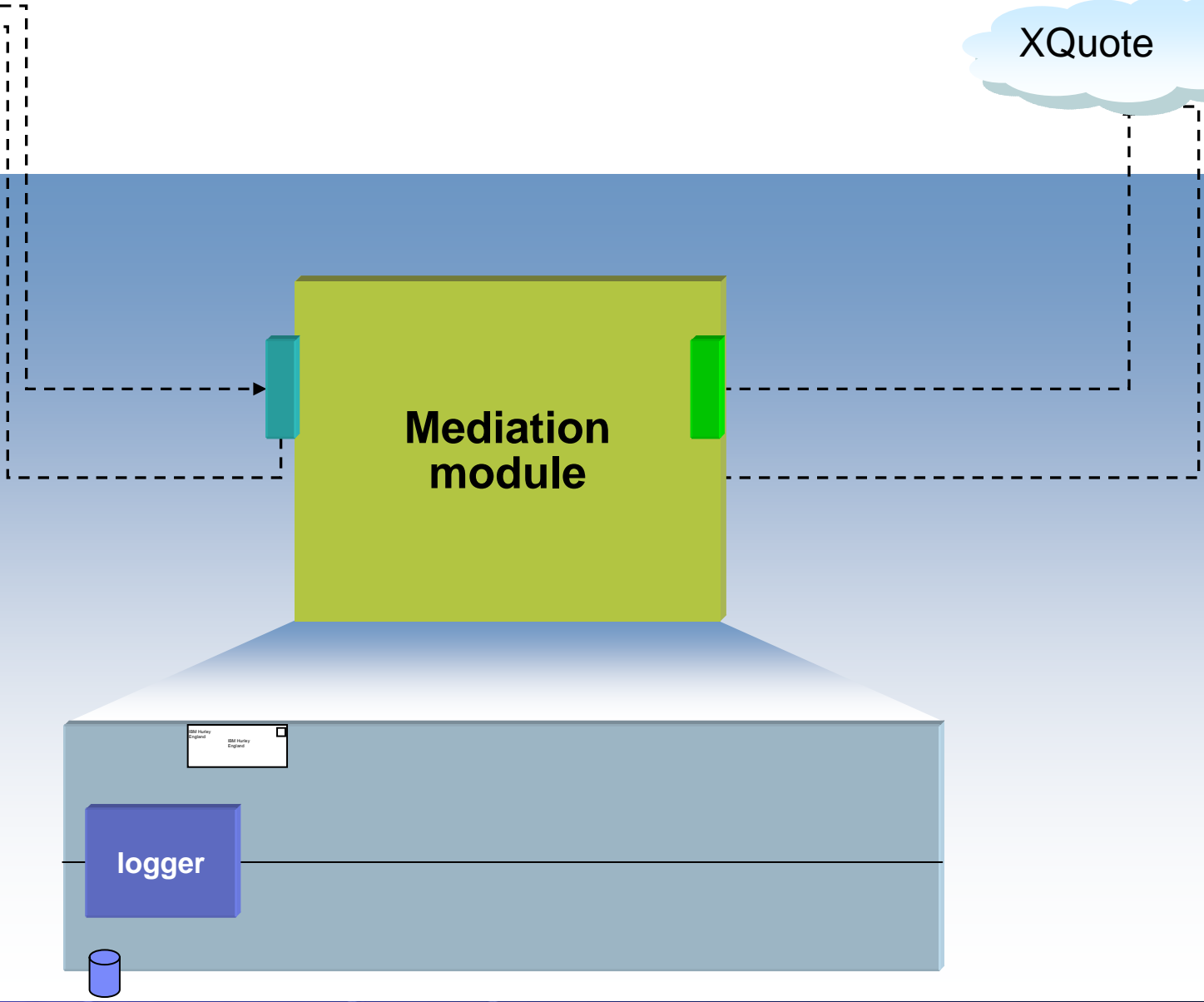
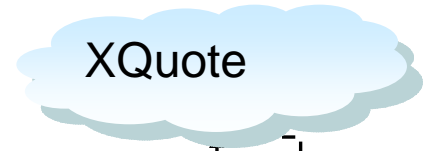
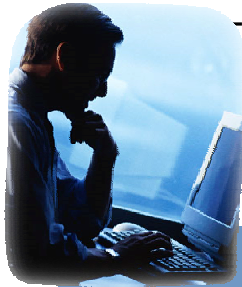
This scenario is implemented in five phases:

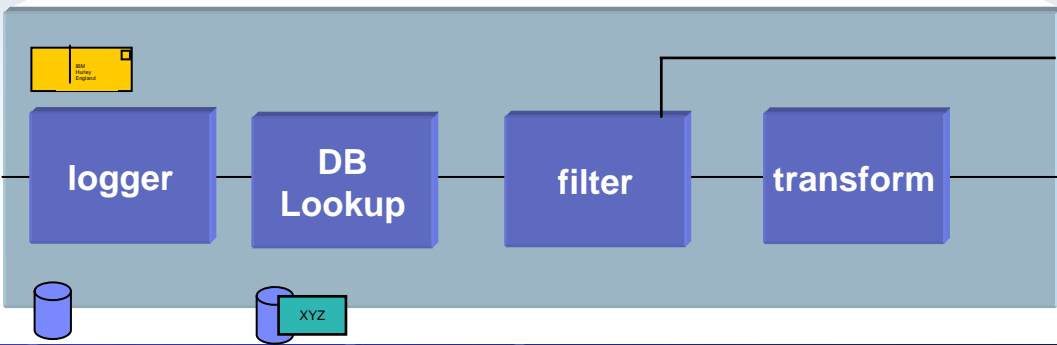
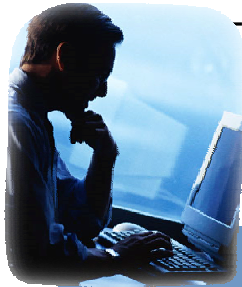
1. The basic quote service is made available to customers by invoking an XQuote Web Service to retrieve 20 minute delayed quotes.
2. Once the service is up and running the company will start logging all requests to the service to satisfy audit requirements.
3. A new subscription service is added for gold customers. This service allows users who pay a subscription fee to receive real time stock quotes rather than delayed pricing for standard customers. The Xignite stock quote service is used to retrieve real-time quotes for these gold customers.
4. Behavior of the XQuote Service identifies a bug in the Servlet. The problem is fixed by updating the Mediation to avoid the problem.
5. More error handling is added to the solution so that failures in the flow are reported to a business dashboard via a CEI event (the dashboard application itself is out of scope of this scenario).



IBM Italy
England
IBM Italy
England







Thanks!