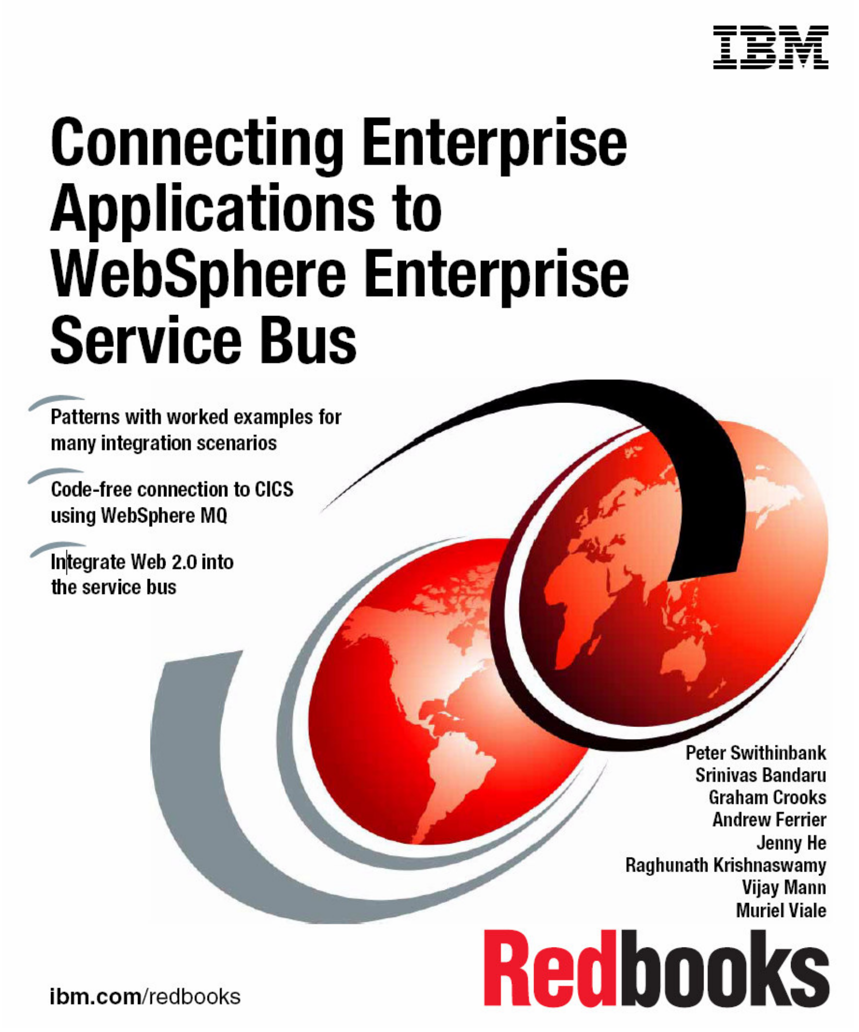
IBM Software Services for WebSphere

Connecting to WebSphere ESB and WebSphere Process Server

Andrew Ferrier, IT Consultant
WebSphere ESB Specialist
andrew.ferrier@uk.ibm.com

History

- **Loosely based on Redbook *Connecting Enterprise Applications to WebSphere Enterprise Service Bus* (and Redbooks workshop of the same name)**
- **Will also mention some new 6.1 items (e.g. HTTP binding, WTX data bindings, data handlers)**



IBM

Connecting Enterprise Applications to WebSphere Enterprise Service Bus

Patterns with worked examples for many integration scenarios

Code-free connection to CICS using WebSphere MQ

Integrate Web 2.0 into the service bus

Peter Swithinbank
Srinivas Bandaru
Graham Crooks
Andrew Ferrier
Jenny He
Raghunath Krishnaswamy
Vijay Mann
Muriel Viale

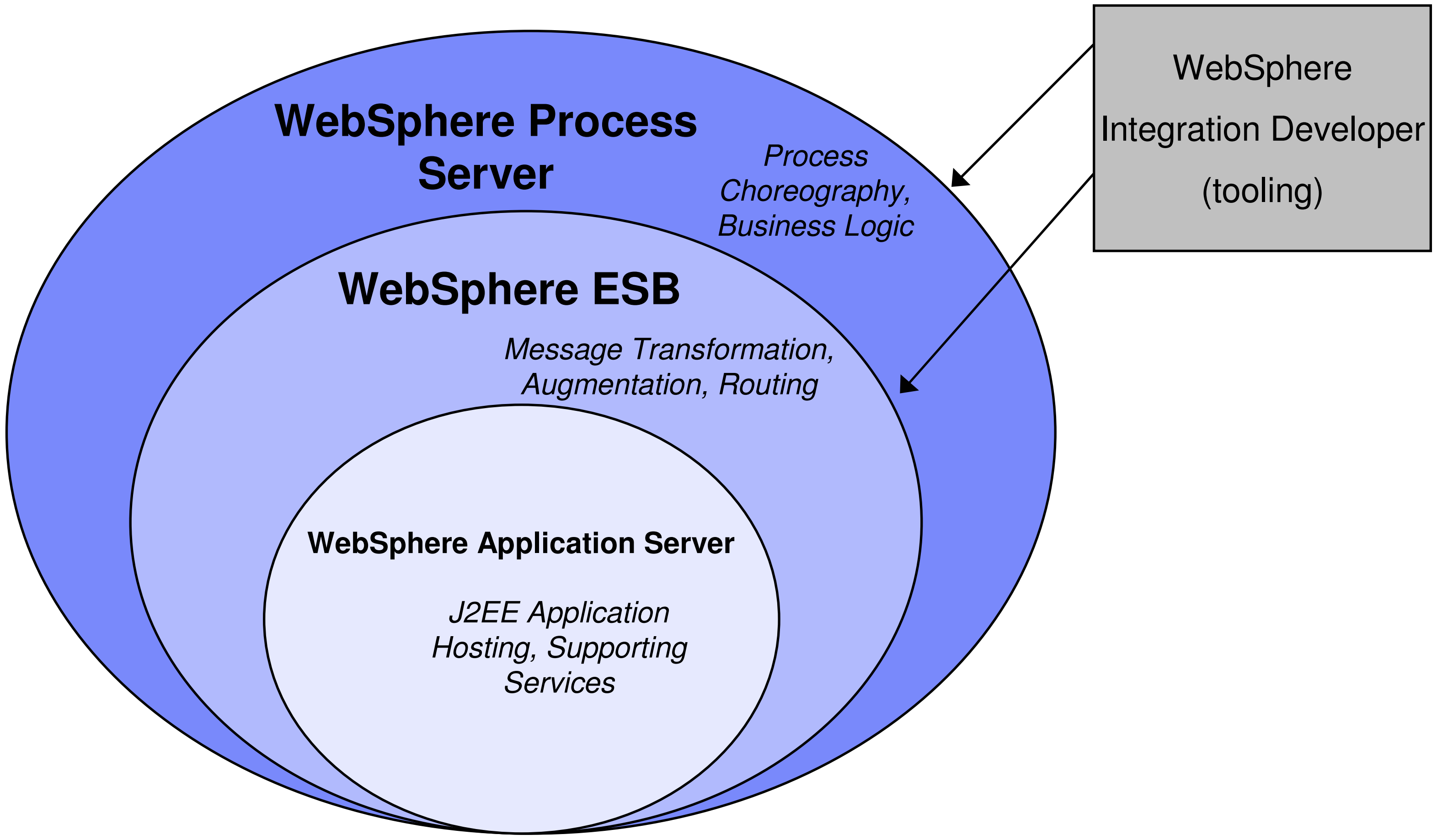
ibm.com/redbooks

Redbooks

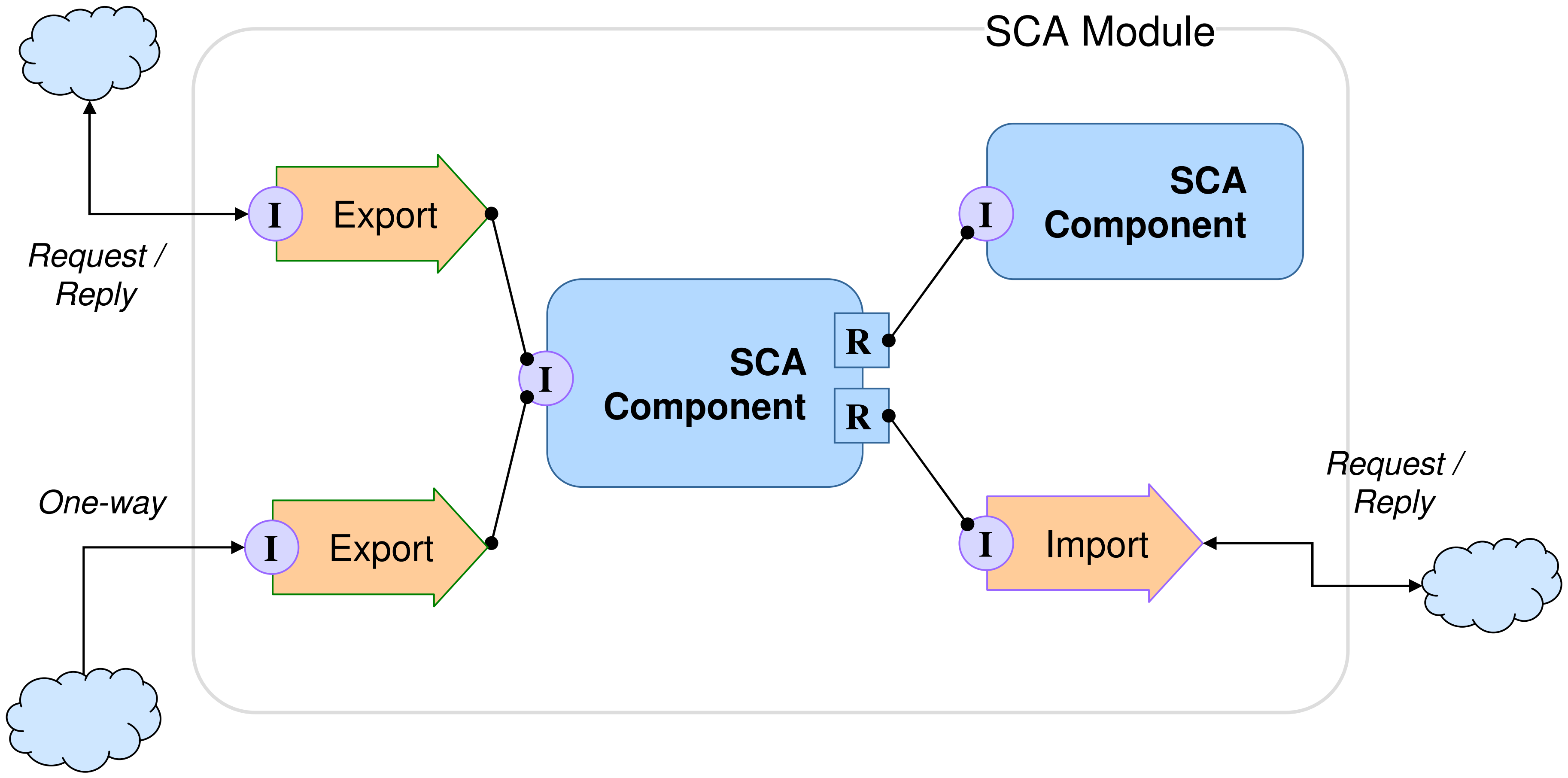
Agenda

- **WebSphere ESB and WebSphere PS – What are they?**
- **Overview of SCA, SDO, and Bindings**
- **Simple Example of WebSphere MQ Data Binding**
- **Overview of Other Bindings**
- **Overview of Data Handlers**

WebSphere SOA Stack

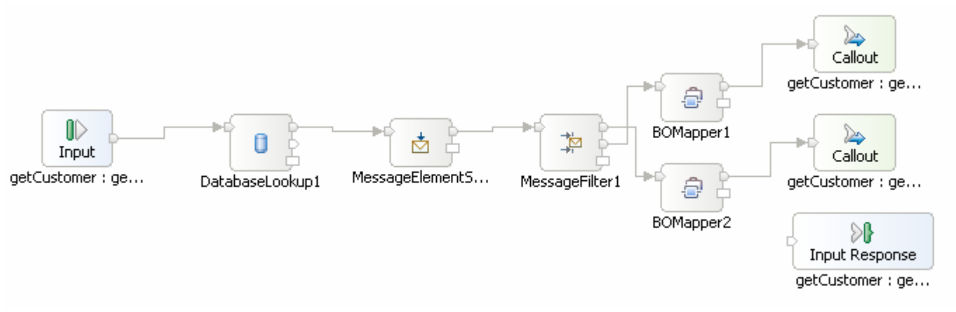

WebSphere Process Server
Process Choreography, Business Logic
WebSphere ESB
Message Transformation, Augmentation, Routing
WebSphere Application Server
J2EE Application Hosting, Supporting Services
WebSphere Integration Developer (tooling)

Service Component Architecture



Types of Module and Component

- **WebSphere ESB supports only Mediation Modules, with:**
  - Mediation Flow Components
  - Java Components
- **WebSphere PS also supports (Integration) Modules, with:**
  - BPEL (Process)
  - …



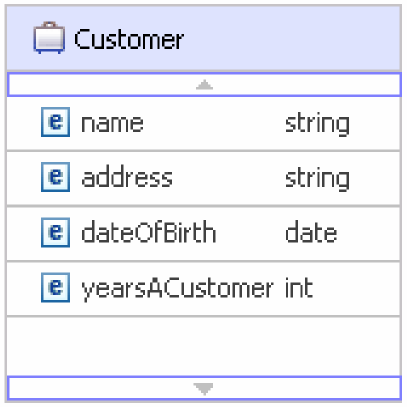
Service Data Object

- **Dynamic Java API used for accessing (mostly) structured data**

```
DataObject customer = createCustomer();
customer.setString(“name”, “Fred”);
customer.setString(“address”, “123 Anytown”);
customer.setDate(“dateOfBirth”, new Date(1975, 2, 1));
customer.setYearsACustomer(0);
```

```
<customer>
  <name>Fred</name>
  <address>123 Anytown</address>
  <date>1975-02-01</date>
  <yearsACustomer>0</yearsACustomer>
</customer>
```

- **Business Object is definition of SDO at design time (underlying representation is XML Schema)**



Customer

name	string
address	string
dateOfBirth	date
yearsACustomer	int

Business Object

- **Definition of a Service Data Object at design time**
- **Underlying definition is an XML Schema**

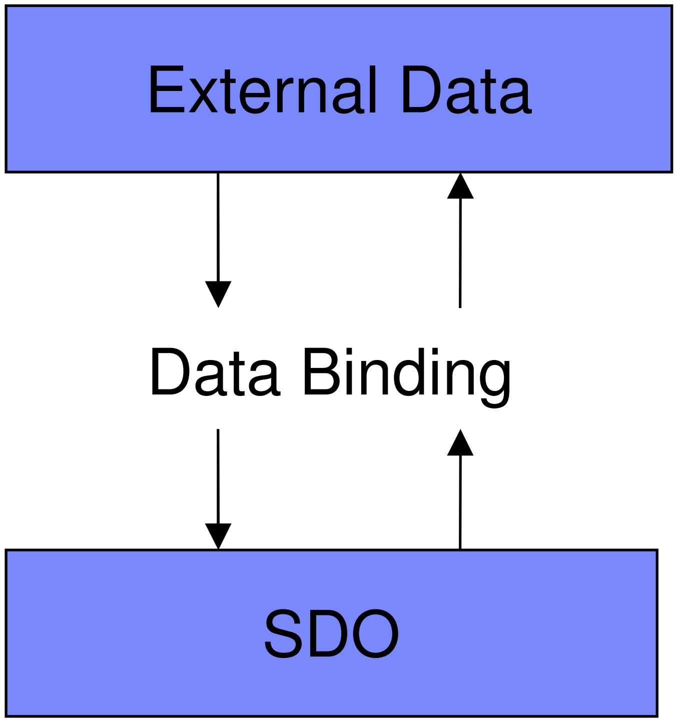
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mm1">
  <xsd:complexType name="Customer">
    <xsd:sequence>
       <xsd:element minOccurs="0" name="name" type="xsd:string"/>
       <xsd:element minOccurs="0" name="address“ type="xsd:string“/>
      <xsd:element minOccurs="0" name="dateOfBirth“ type="xsd:date“/>
      <xsd:element minOccurs="0" name="yearsACustomer“ type="xsd:int“/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

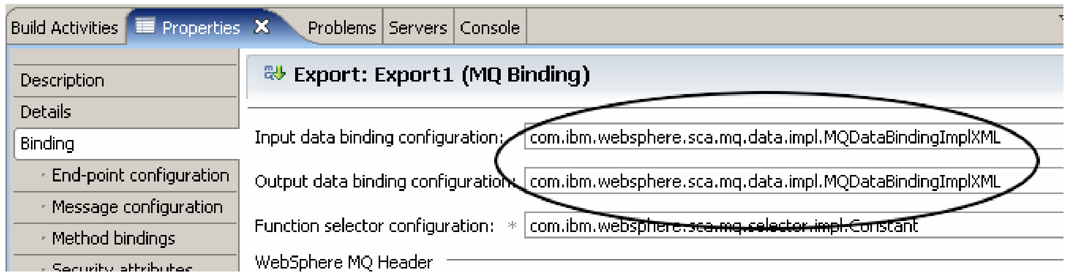
Types of Import and Export (Bindings)

- **Web Services Bindings**
- **Messaging Bindings:**
  - WebSphere MQ and MQ / JMS Bindings
  - JMS Bindings (incl. Generic – new in 6.1)
- **HTTP Binding (new in 6.1)**
- **Stateless Session Bean Binding (import only)**
- **Standalone Reference (export only)**
- **JCA (WebSphere) Adapters**
  - Application
  - Technology
- **WebSphere Business Integration Adapters**

Data Bindings

- **'Bindings' are really transport bindings**
- **Also the concept of data bindings**
  - Some inbuilt, some custom
- **Web Services (transport) binding is the only one without a data binding**





Export: Export1 (MQ Binding)

Input data binding configuration: com.ibm.websphere.sca.mq.data.impl.MQDataBindingImplXML

Output data binding configuration: com.ibm.websphere.sca.mq.data.impl.MQDataBindingImplXML

Function selector configuration: * com.ibm.websphere.sca.mq.selector.impl.Constant

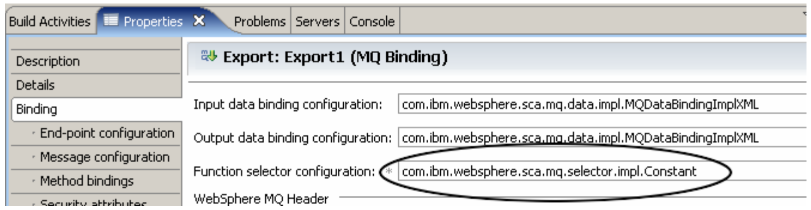
WebSphere MQ Header

Data Binding Common Superinterface

```
import commonj.connector.runtime.DataBinding;

public interface DataBinding extends Serializable {
  public DataObject getDataObject();
  public void setTransportDataObject(DataObject dataObject);
  public void setSpecificData(xyz);
  public XYZ getTransportSpecificData();
}
```

Function Selector

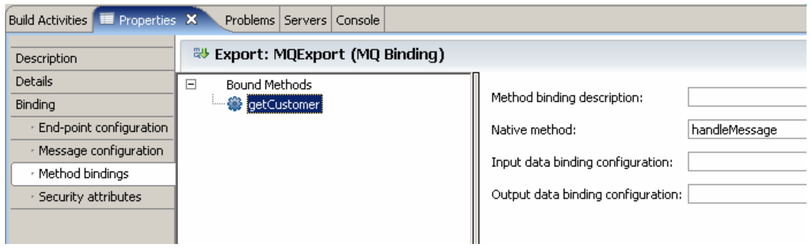


Build Activities | Properties | Problems | Servers | Console

Description
Details
Binding
- End-point configuration
- Message configuration
- Method bindings
- Security attributes

Export: Export1 (MQ Binding)

Input data binding configuration:	com.ibm.websphere.sca.mq.data.impl.MQDataBindingImplXML
Output data binding configuration:	com.ibm.websphere.sca.mq.data.impl.MQDataBindingImplXML
Function selector configuration:	com.ibm.websphere.sca.mq.selector.impl.Constant
WebSphere MQ Header	



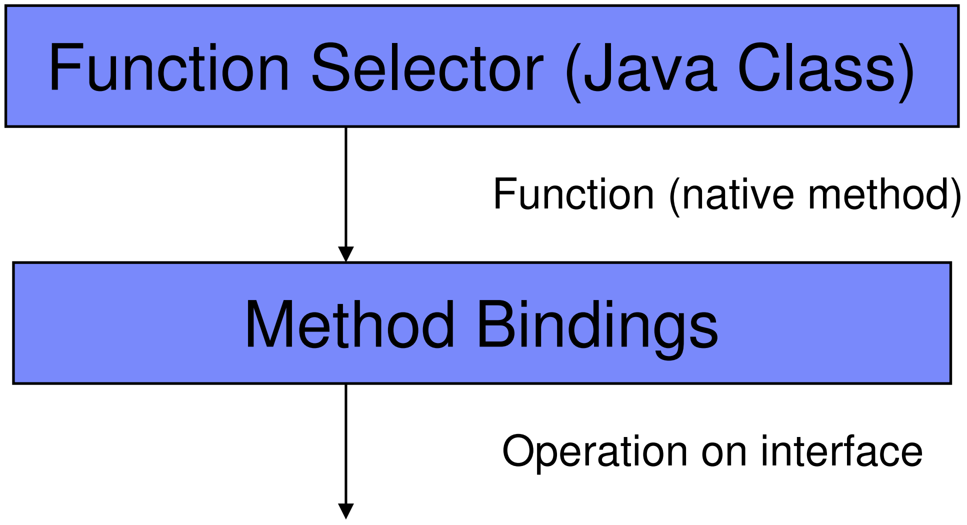
Build Activities | Properties | Problems | Servers | Console

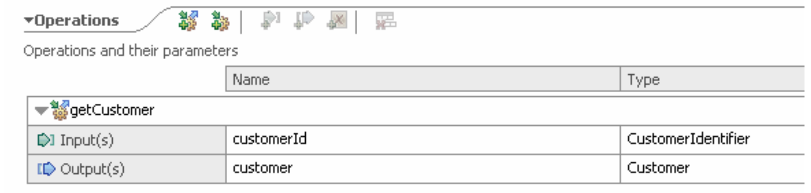
Description
Details
Binding
- End-point configuration
- Message configuration
- Method bindings
- Security attributes

Export: MQExport (MQ Binding)

Bound Methods
- getCustomer

Method binding description:	
Native method:	handleMessage
Input data binding configuration:	
Output data binding configuration:	





Operations

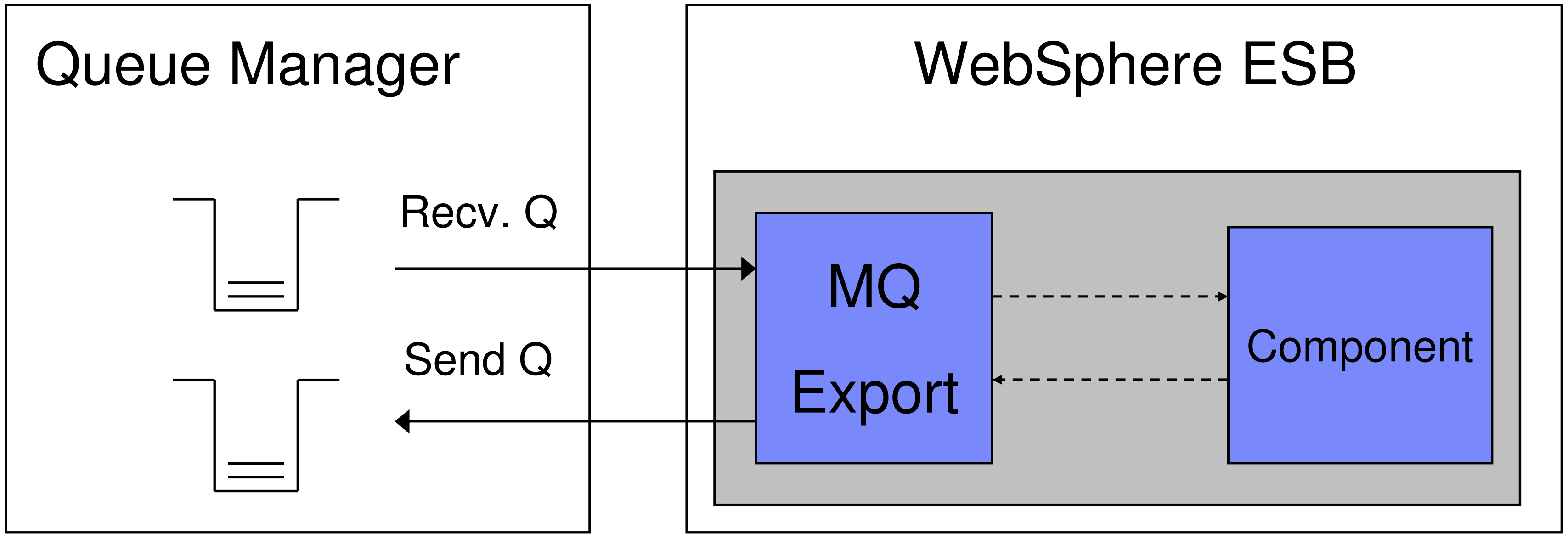
Operations and their parameters

	Name	Type
getCustomer		
Input(s)	customerId	CustomerIdentifier
Output(s)	customer	Customer

WebSphere MQ Bindings

- **Provide a way to read and write messages to WebSphere MQ**
  - Expose services via exports
  - Invoke services via imports
- **MQ data binding needed for body of message**
- **Inbuilt support for MQRFH and MQRFH2 headers**
- **Other headers supported via custom header bindings**

Example Interaction with MQ Export

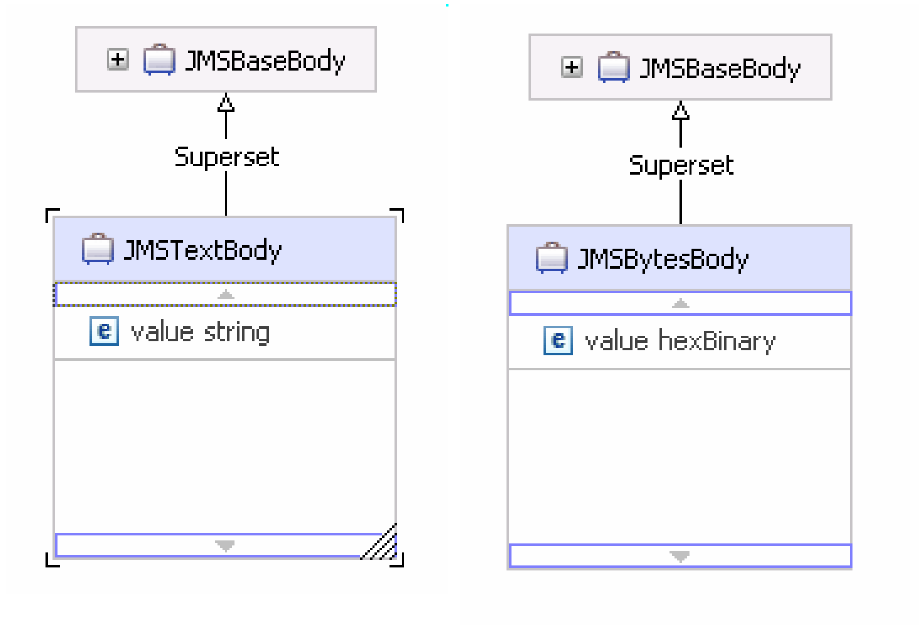


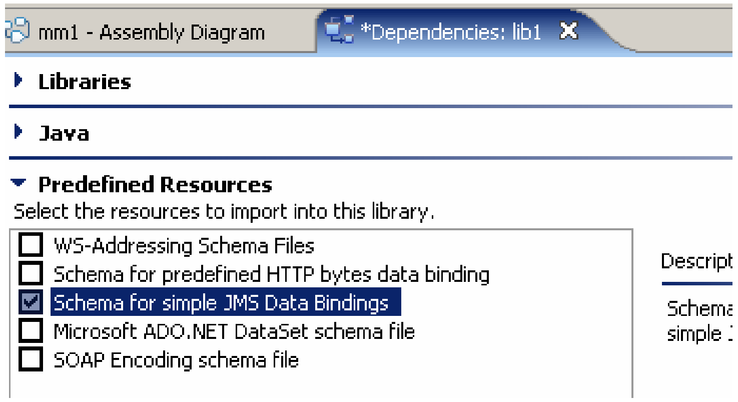
Supplied MQ Bindings

Name	Class `com.ibm.websphere.sca.mq.data.impl.`
Serialized as XML	`MQDataBindingImplXML`
Serialized Java Object	`MQDataBindingImplJava`
Unstructured Text Message	`MQDataBindingImplText` (uses `JMSTextBody` BO)
Unstructured Binary Message	`MQDataBindingImplBinary` (uses `JMSBytesBody` BO)
WTX Data Binding	`com.ibm.wbiserver.databinding.wtx.WTXDataBinding`

Unstructured Data Bindings

- **Use JMSTextBody and JMSBytesBody pre-defined Business Objects**





Select the resources to import into this library.

- [ ] WS-Addressing Schema Files
- [ ] Schema for predefined HTTP bytes data binding
- [x] Schema for simple JMS Data Bindings
- [ ] Microsoft ADO.NET DataSet schema file
- [ ] SOAP Encoding schema file

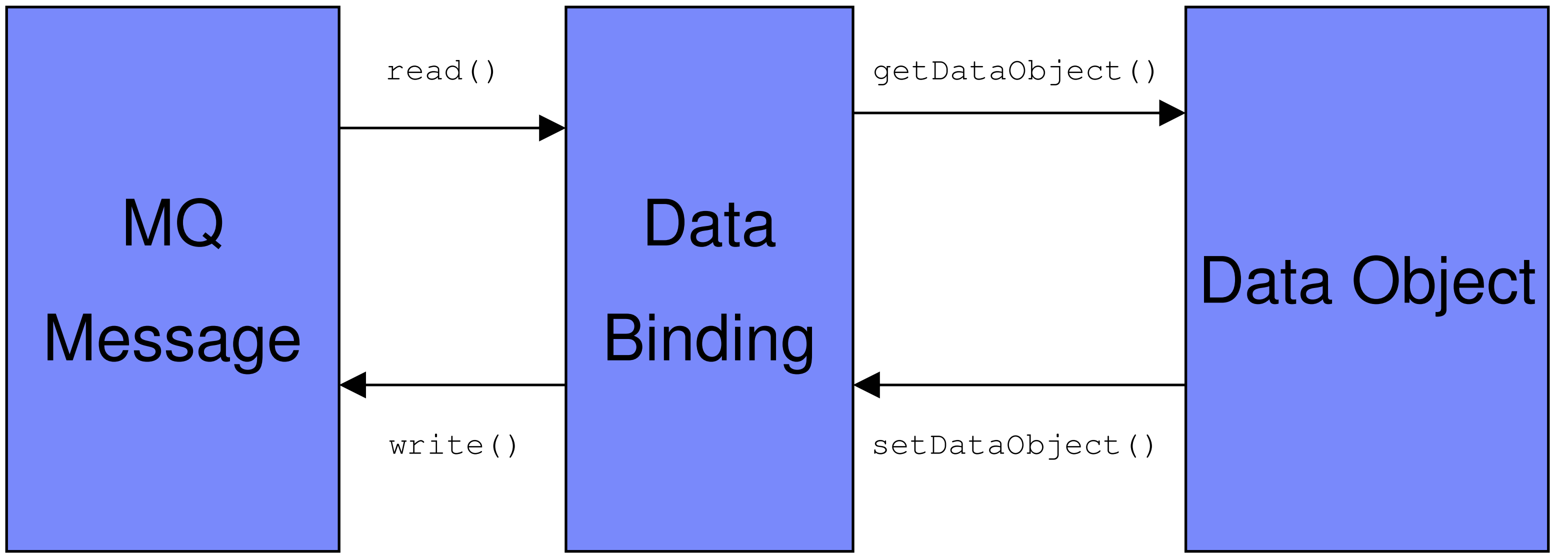
Descript

Schema
simple

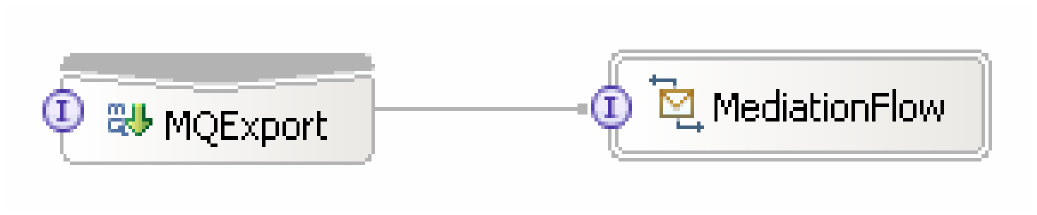
Implementing a Custom MQ Data Binding

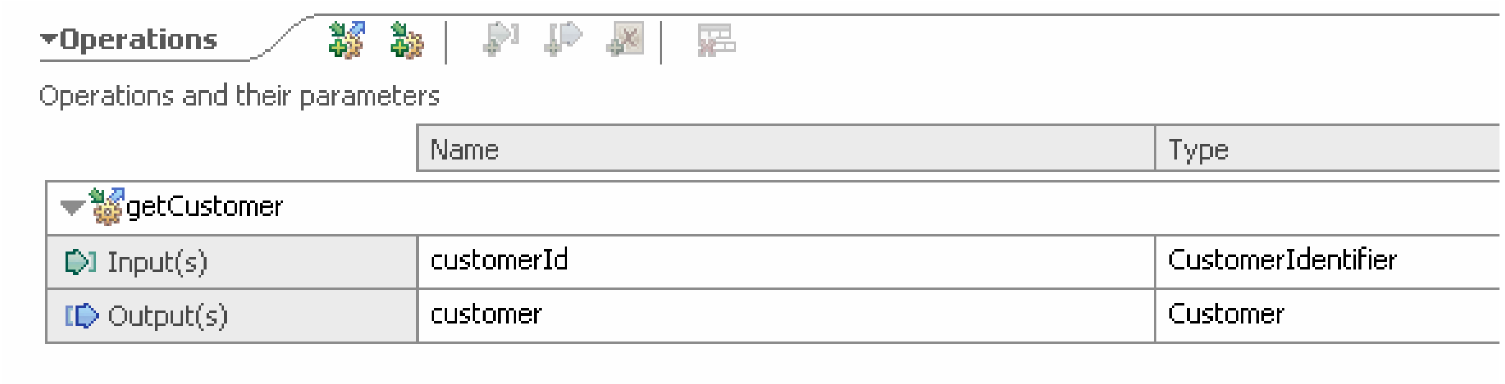
- **Interface is**
  - `com.ibm.websphere.sca.mq.data.MQBodyDataBinding`
    - `(implements commonj.connector.runtime.DataBinding)`
- **Most important additional methods:**
  - `public void read(MQMD md, List headers, MQDataInputStream input);`
  - `public void write(MQMD md, List headers, MQDataOutputStream output);`
- **Other methods:**
  - `public boolean isBusinessException();`
  - `public void setBusinessException(boolean isBusinessException);`
  - `public void setFormat(String format);`
  - `public String getFormat();`

Implementing a Custom MQ Data Binding



Simple Example – Getting Customer Info





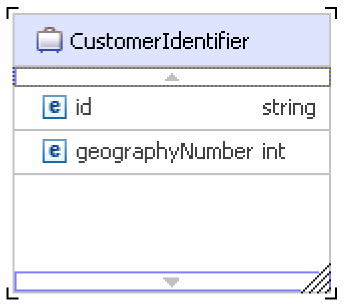
Operations and their parameters

	Name	Type
getCustomer		
Input(s)	customerId	CustomerIdentifier
Output(s)	customer	Customer

Customer and CustomerInfo

- **CustomerInfo:**

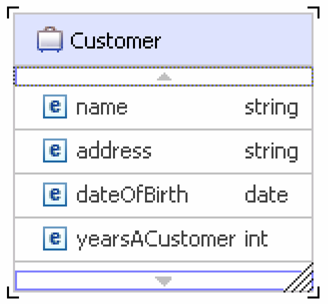
```
struct CustomerIdentifier {
  MQCHAR[8] id;
  MQINT[32] geographyNumber;
}
```


CustomerIdentifier

id	string
geographyNumber	int

- **Customer:**

```
struct Customer {
  MQCHAR[32] name;
  MQCHAR[256] address;
   MQCHAR[8] dateOfBirth;   // YYYYMMDD
  MQINT[32] yearsACustomer;
}
```


Customer

name	string
address	string
dateOfBirth	date
yearsACustomer	int

read() method

```
public class CustomerMQBinding implements MQBodyDataBinding {
  private int geographyNumber;
  private String id;

  public void read(MQMD mqmd, List headers, MQDataInputStream
  inputStream)
  {
    id = inputStream.readMQCHAR(8);
    geographyNumber = inputStream.readMQINT32();
  }
```

getDataObject() method

```
public DataObject getDataObject() throws DataBindingException {
  DataObject object = constructBusinessObject("http://mm1",
          "CustomerIdentifier");
  object.setString("id", id);
  object.setInt("geographyNumber", geographyNumber);
  return object;
}

private DataObject constructBusinessObject(String namespace, String name) {
   BOFactory bofactory = (BOFactory) ServiceManager.INSTANCE
                   .locateService("com/ibm/websphere/bo/BOFactory");
   DataObject object = bofactory.create(namespace, name);
   return object;
}
```

setDataObject() method

```
private String name;
private String address;
private int totalSpend;
private Date dateOfBirth;

public void setDataObject(DataObject arg0)
{
  name = arg0.getString("name");
  address = arg0.getString("address");
  dateOfBirth = arg0.getDate("dateOfBirth");
  totalSpend = arg0.getInt("totalSpend");
}
```

write() method

```
public void write(MQMD mqmd, List headers, MQDataOutputStream
  outputStream)
{
  outputStream.writeMQCHAR(32, name);
  outputStream.writeMQCHAR(256, address);
  outputStream.writeMQCHAR8(new SimpleDateFormat("yyyyMMdd")
        .format(dateOfBirth));
  outputStream.writeMQINT32(totalSpend);
}
```

Supplied MQ Function Selectors

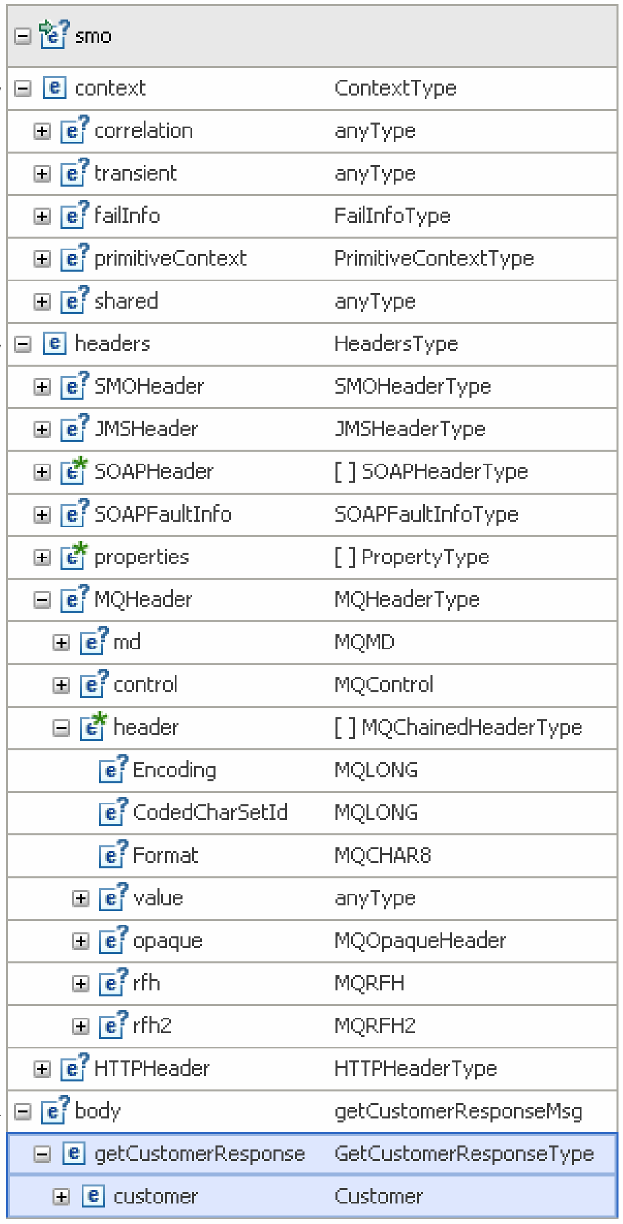
Name	**Class** **`com.ibm.websphere.sca.mq.selector.impl.`**
Use *handleMessage* as the native function	**`Constant`**
Use message body's format as the native function	**`Format`**
Use type information as the native function	**`Type`** (URL containing Msd, Set, Type, and Format properties from MQRFH2)
Use JMS default function selector	**`TargetFunctionNameProperty`**

- **Or for a custom selector: implement interface** `com.ibm.websphere.sca.mq.selector.MQFunctionSelector`
- **One method:**

```
public abstract String generateEISFunctionName(MQMD md, String bodyFormat, List
headers, MQDataInputStream input)
```

Service Message Object

- Only used inside mediation flows
- Contains context (scratchpads) for various mediation functions
- Gives access to headers inaccessible in other SCA components
- As well as message body content



smo	
context	ContextType
correlation	anyType
transient	anyType
failInfo	FailInfoType
primitiveContext	PrimitiveContextType
shared	anyType
headers	HeadersType
SMOHeader	SMOHeaderType
JMSHeader	JMSHeaderType
SOAPHeader	[ ] SOAPHeaderType
SOAPFaultInfo	SOAPFaultInfoType
properties	[ ] PropertyType
MQHeader	MQHeaderType
md	MQMD
control	MQControl
header	[ ] MQChainedHeaderType
Encoding	MQLONG
CodedCharSetId	MQLONG
Format	MQCHAR8
value	anyType
opaque	MQOpaqueHeader
rfh	MQRFH
rfh2	MQRFH2
HTTPHeader	HTTPHeaderType
body	getCustomerResponseMsg
getCustomerResponse	GetCustomerResponseType
customer	Customer

MQ Header Binding Interface – Most important methods

```
public interface MQHeaderDataBinding extends DataBinding
{
  public boolean isSupportedFormat(String format);

  public DataObject getDataObject();
  public void setDataObject(DataObject dObj);

  public void read(String format, MQDataInputStream input);
  public void write(String format, MQDataOutputStream output);
}
```

JMS, MQ/JMS and Generic JMS Bindings

- **JMS messaging systems**
  - JMS binding via SIB (a.k.a. WPM)
  - MQ/JMS binding via MQ
  - Generic JMS via any other JMS messaging provider, using Generic JMS API
- **All data bindings inherited from**
  `com.ibm.websphere.sca.jms.data.JMSDataBinding`
- **All function selectors based on**
  `com.ibm.websphere.jms.data.bindings.JMSFunctionSelector`

JMS Supplied Data Bindings

Name	Class `com.ibm.websphere.sca.jms.data.impl.`
Business Object XML using JMSTextMessage	`JMSDataBindingImplXML`
Serialized Business Object using JMSObjectMessage	`JMSDataBindingImplJava`
Simple JMS BytesMessage Data Binding	`JMSBytesDataBinding`
Simple JMS MapMessage DataBinding	`JMSMapDataBinding`
Simple JMS Message Data Binding	`JMSBaseDataBinding`
Simple JMS ObjectMessage Data Binding	`JMSObjectDataBinding`
Simple JMS StreamMessage DataBinding	`JMSStreamDataBinding`
Simple JMS TextMessage DataBinding	`JMSTextDataBinding`
WTX Data Binding	`com.ibm.wbiserver.databinding.wtx.WTXDataBinding`

JMS Supplied Function Selectors

Name	**Class** `com.ibm.websphere.sca.jms.selector.impl.`
Default JMS Function Selector (uses *TargetFunctionName* property)	**`JMSFunctionSelectorImpl`**
(Hardcoded) JMS Function Selector (uses *JMSType* property)	**`JMSFunctionSelector`**

HTTP Binding

- **Allows exposure of, or invocation of, services with arbitrary HTTP content**
- **Access to HTTP headers**
- **All data bindings inherited from**
  `com.ibm.websphere.http.data.bindings.HTTPStreamDataBinding`
- **All function selectors inherited from**
  `com.ibm.websphere.http.selectors.HTTPFunctionSelector`

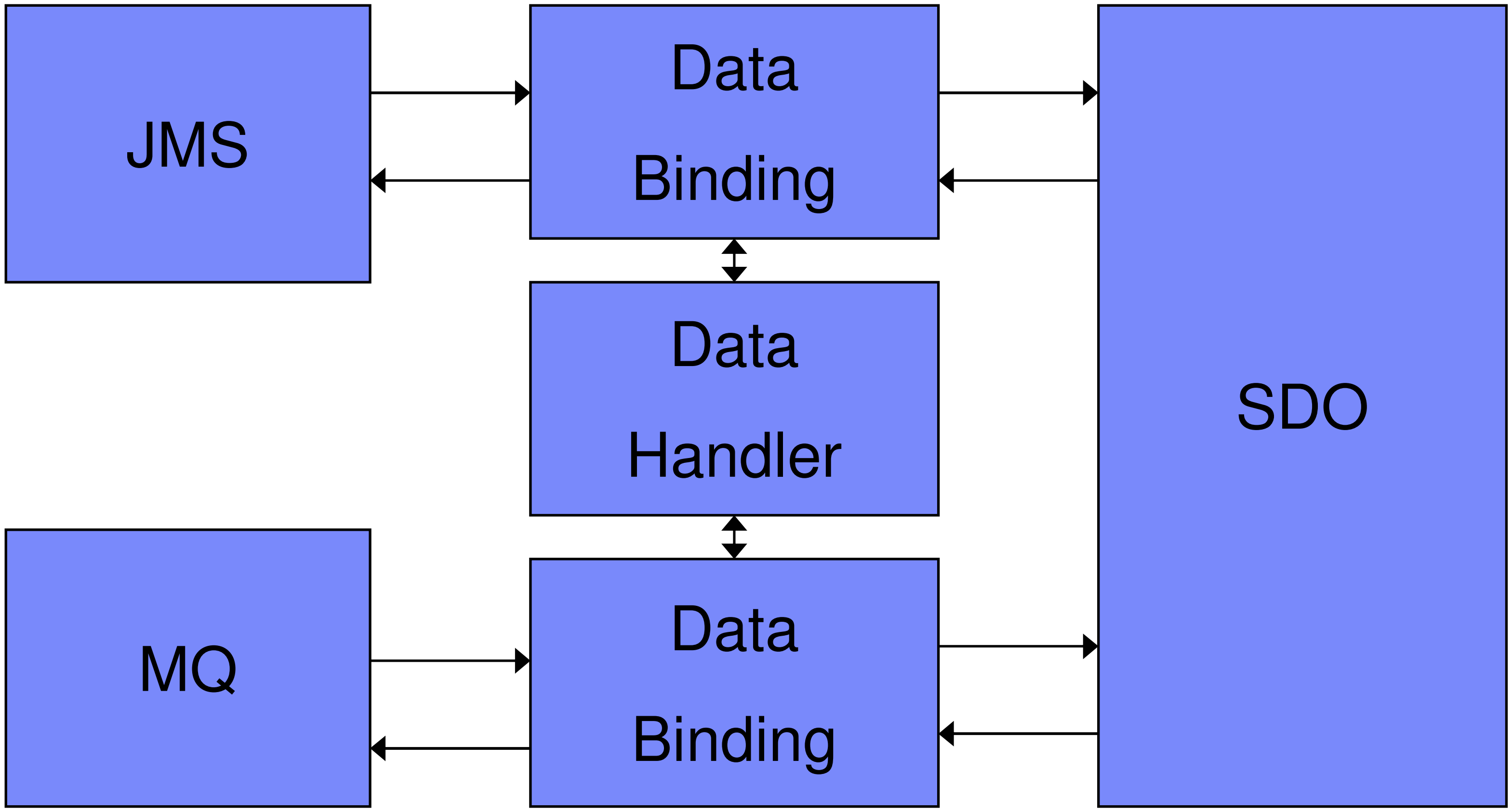
HTTP Supplied Data Bindings

Name	Class **`com.ibm.websphere.http.data.bindings.`**
HTTP Bytes Message Data Binding	`HTTPStreamDataBindingBytes`
HTTP SOAP Message Data Binding	`HTTPStreamDataBindingSOAP`
HTTP XML Message Data Binding	`HTTPStreamDataBindingXML`
WTX Data Binding	`com.ibm.wbiserver.databinding.wtx.WTXDataBinding`

HTTP Supplied Function Selectors

Name	Class `com.ibm.websphere.http.selectors.`	Notes
TargetFunctionName Header	`HeaderHTTPFunctionSelector`	
URL and HTTP Method	`URLMethodFunctionSelector`	e.g. `/ExportName/methodName@GET`

Data Handlers – new in 6.1



Data Handlers

- **Interface is `commonj.connector.runtime.DataHandler`**
- **Need to implement:**
  - `public Object transform(Object source, Class target, Object options)`
  - `public void transformInto(Object source, Object target, Object options)`
  - `public void setBindingContext(Map context)`
- **Objects typically InputStream, Reader, OutputStream, Writer, DataObject**
- **e.g.:**

```
reader = new StringReader(inputStream.readMQCHAR(inputStream.available()));
dataObject = dataHandler.transform(reader, DataObject.class, null);
```

Summary

- **SCA**
- **SDO and Business Objects**
- **(Transport) Bindings**
- **Data Bindings**
- **Function Selector and Method Bindings**
- **MQ Binding**
  - Supplied Data Bindings
  - Custom Data Binding
  - Function Selector(s)
- **JMS, MQ/JMS and Generic JMS Bindings**
- **HTTP Binding**
- **Data Handlers**

More Information

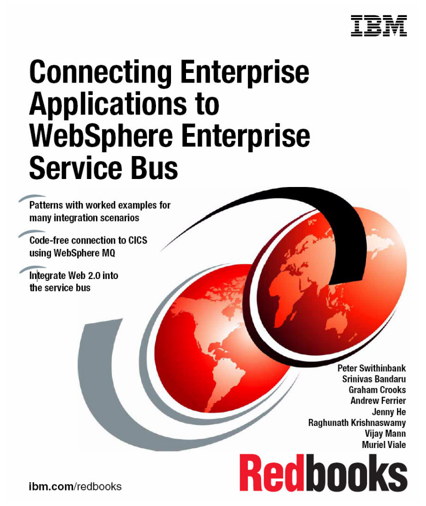
- **Redbook:**

  http://www.redbooks.ibm.com/abstracts/sg247406.html
- **Workshop (1st April 2008):**

  http://www.redbooks.ibm.com/workshops/GR8222?Open
- **WebSphere SOA Products InfoCenter:**

  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp
- **SOA Tips 'n' Tricks:**

  http://soatipsntricks.wordpress.com/



IBM

Connecting Enterprise Applications to WebSphere Enterprise Service Bus

Patterns with worked examples for many integration scenarios

Code-free connection to CICS using WebSphere MQ

Integrate Web 2.0 into the service bus

Peter Swithinbank
Srinivas Bandaru
Graham Crooks
Andrew Ferrier
Jenny He
Raghunath Krishnaswamy
Vijay Mann
Muriel Viale

Redbooks

ibm.com/redbooks

andrew.ferrier@uk.ibm.com