

# Getting Started with Composite Applications

# Objectives

After completing this lesson, you should be able to:

- Describe how services use WSDL files to communicate
- Use Mediator components and File adapters to perform basic receiving, routing, and writing of messages
- Name and describe the contents of some of the files that are created as part of a composite application

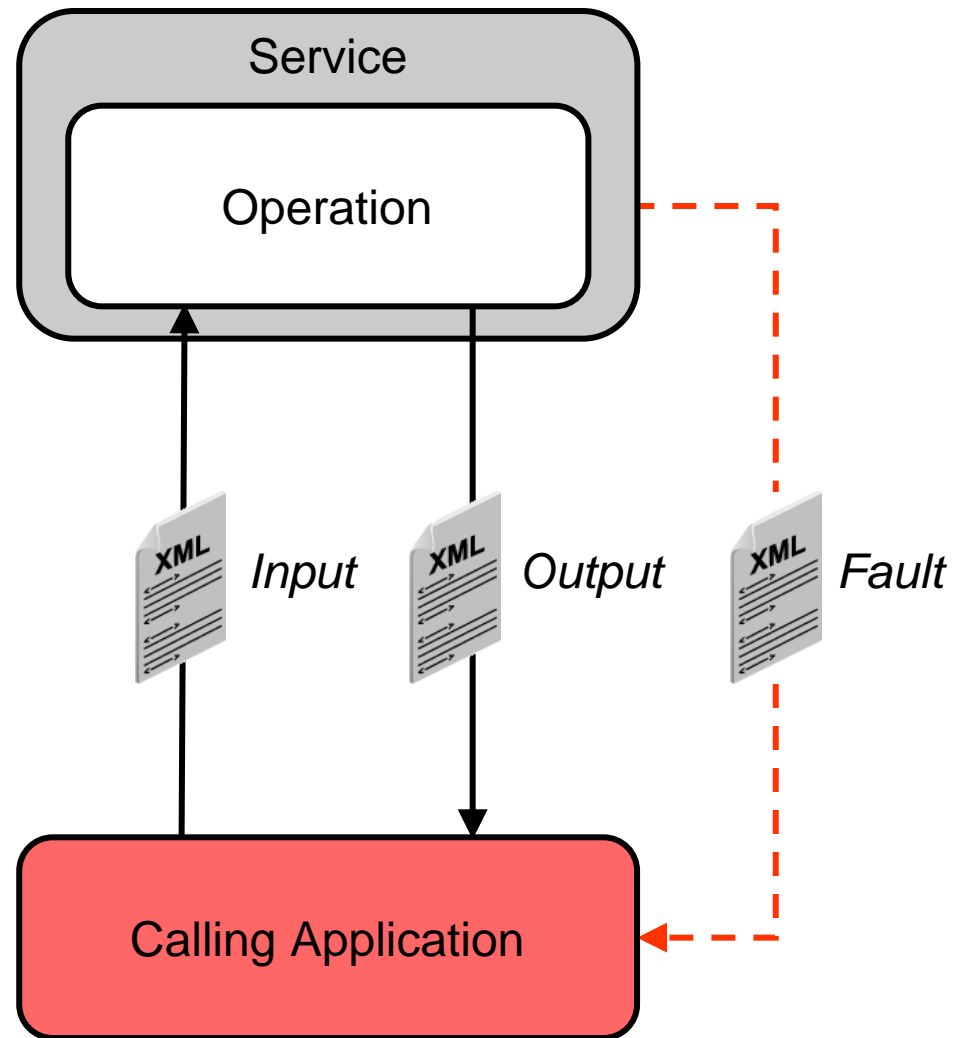


# Agenda

- How Services Use WSDL Files to Communicate
- Mediator Components
- File Adapters
- Composite Application Files



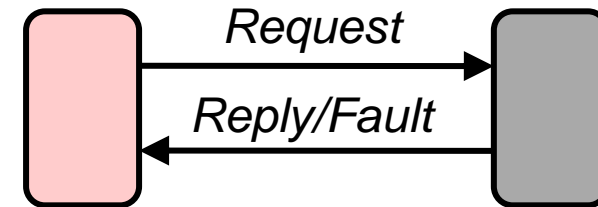
# How Services Communicate



# Synchronous and Asynchronous Interactions

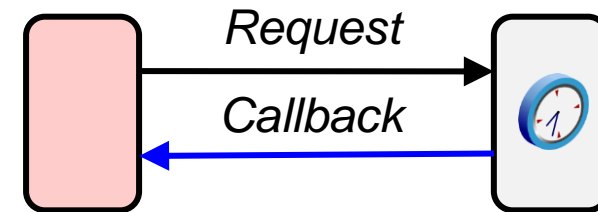
## Synchronous request/response

- Real-time response or error feedback
- Client in waiting mode



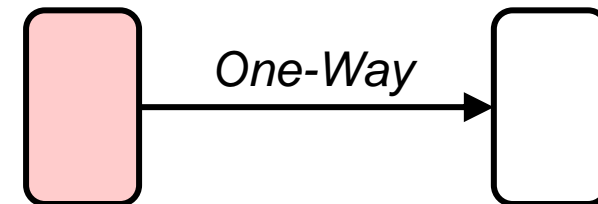
## Asynchronous request/callback

- Client free after request submission
- Separate service invocation for response

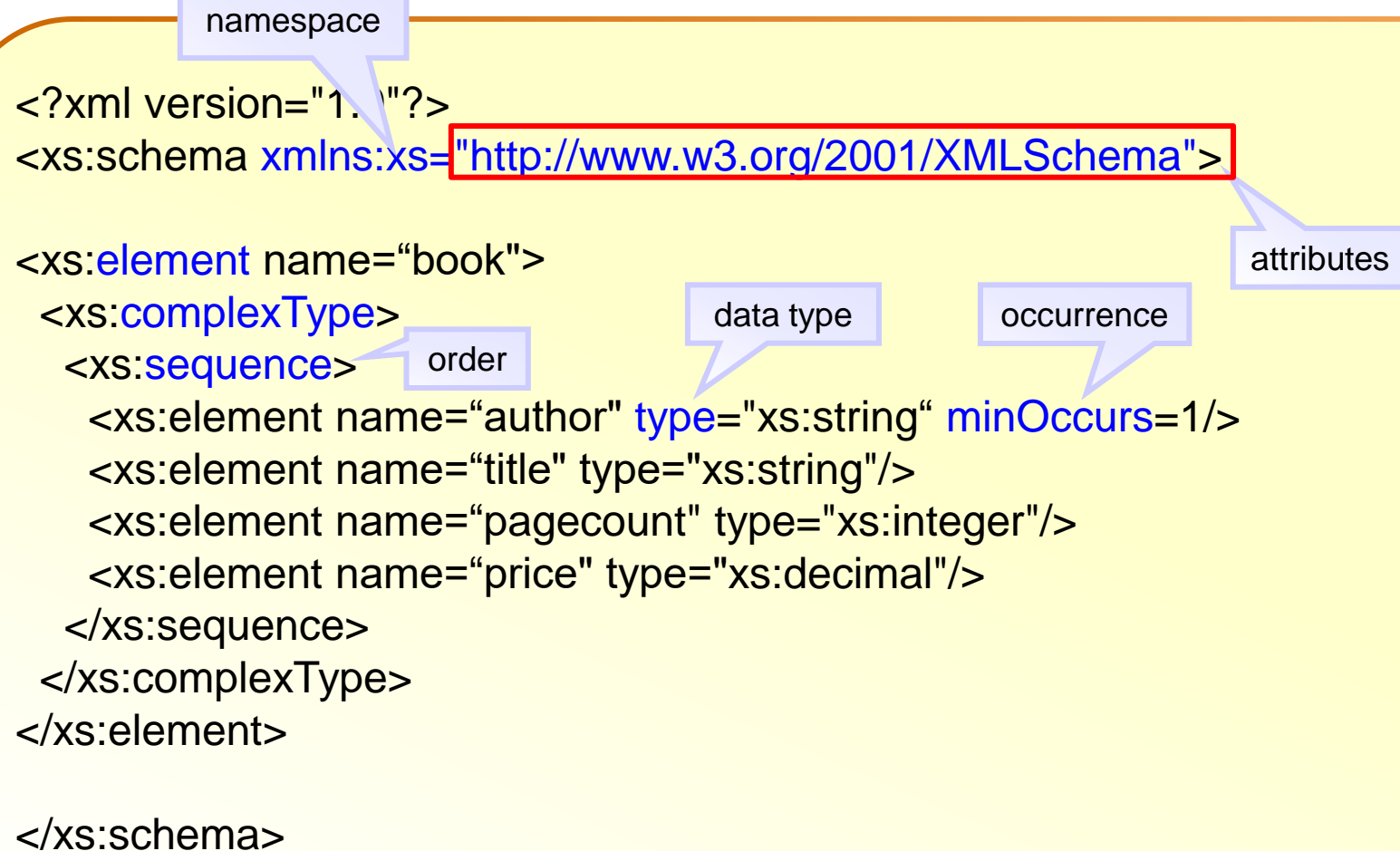


## Asynchronous request only

- Also known as “fire and forget”
- Client free after request submission
- No response message (ACK only)



# Describing a Message with XSD

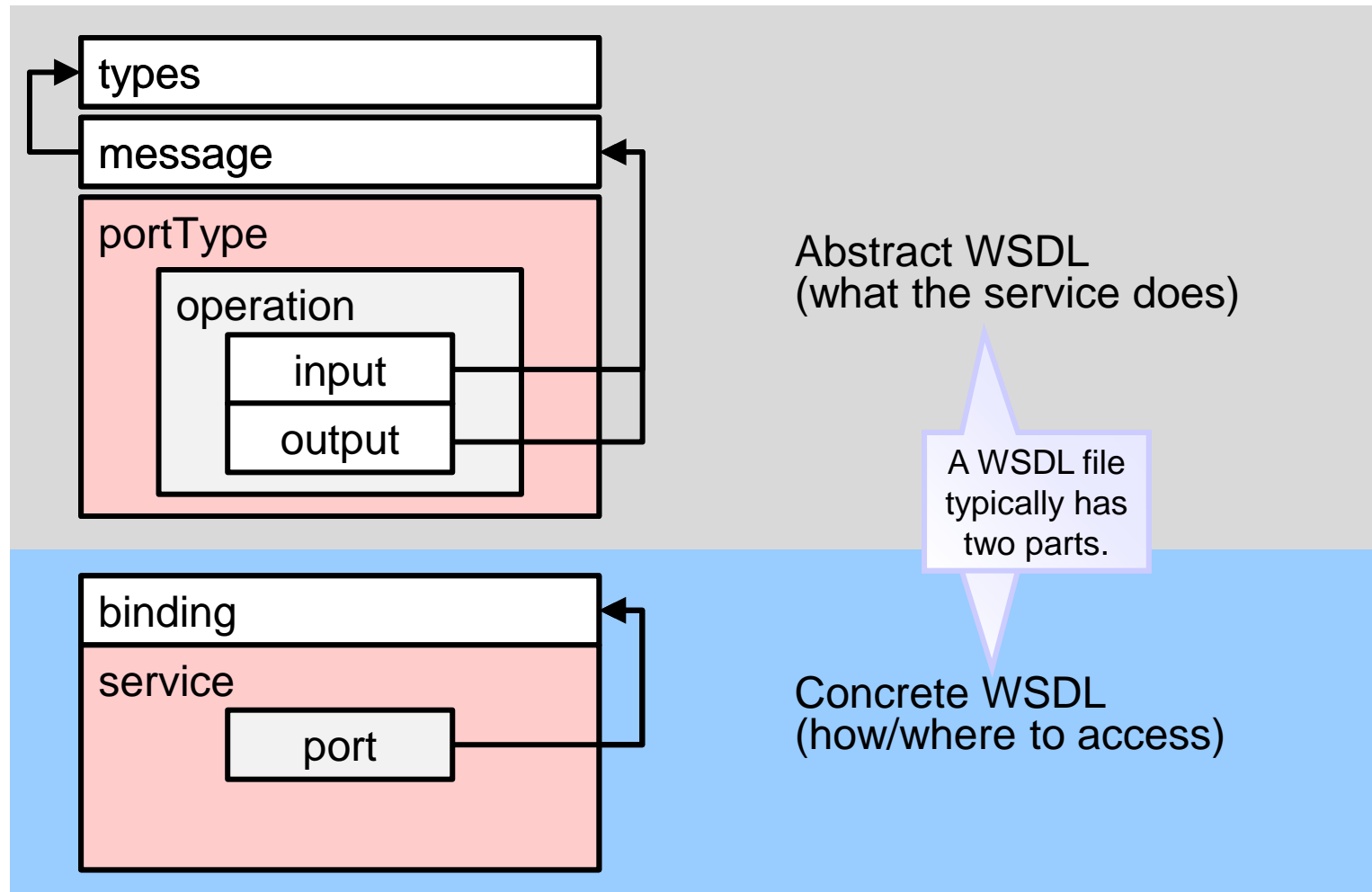


The diagram shows an XSD schema for a 'book' element. Annotations point to various parts of the schema:

- namespace**: Points to the `xmlns:xs="http://www.w3.org/2001/XMLSchema"` attribute.
- attributes**: Points to the `minOccurs=1` attribute on the `author` element.
- data type**: Points to the `type="xs:string"` attribute on the `author` element.
- occurrence**: Points to the `minOccurs=1` attribute on the `author` element.
- order**: Points to the `<xs:sequence>` tag.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="author" type="xs:string" minOccurs=1/>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="pagecount" type="xs:integer"/>
        <xs:element name="price" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# Web Services Description Language



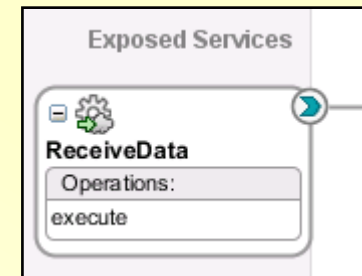
# Abstract WSDL

```
<?xml version= '1.0' encoding= 'UTF-8' ?>
<wsdl:definitions
  name="ReceiveData"
  targetNamespace="http://oracle.com/sca/soapservice/Basics/HelloWorld/ReceiveData"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:inp1="http://www.example.org/ns/porder"
  xmlns:tns="http://oracle.com/sca/soapservice/Basics/HelloWorld/ReceiveData"
>
  <wsdl:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" >
      <import namespace="http://www.example.org/ns/porder" schemaLocation="xsd/po.xsd" />
    </schema>
  </wsdl:types>
  <wsdl:message name="requestMessage">
    <wsdl:part name="part1" element="inp1:PurchaseOrder"/>
  </wsdl:message>
  <wsdl:portType name="execute_ptt">
    <wsdl:operation name="execute">
      <wsdl:input message="tns:requestMessage"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

1

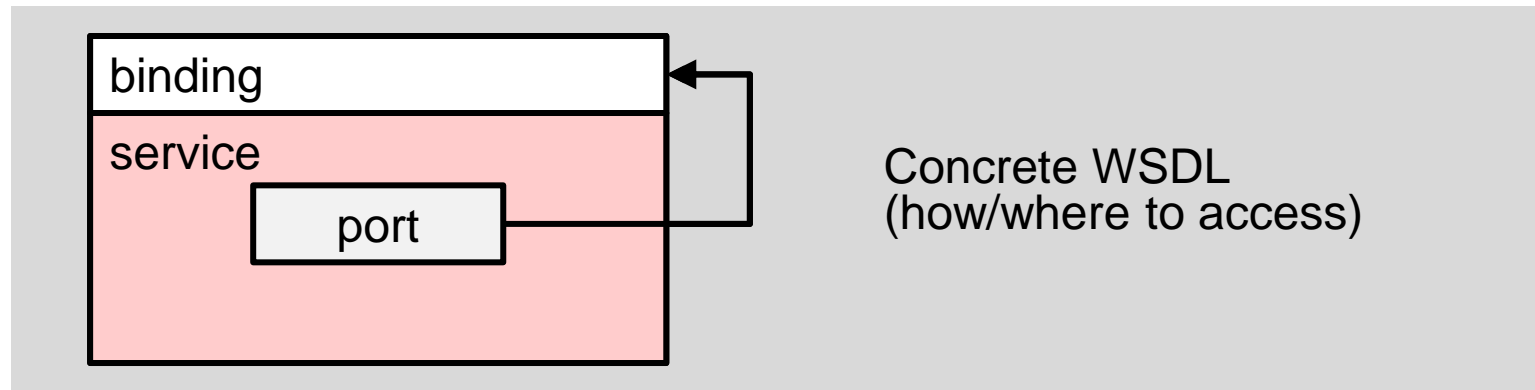
2

3





# Concrete WSDL



# Quiz



Abstract and concrete WSDLs are two separate but related files that describe the conversation between a service and the calling application.

- a. True
- b. False

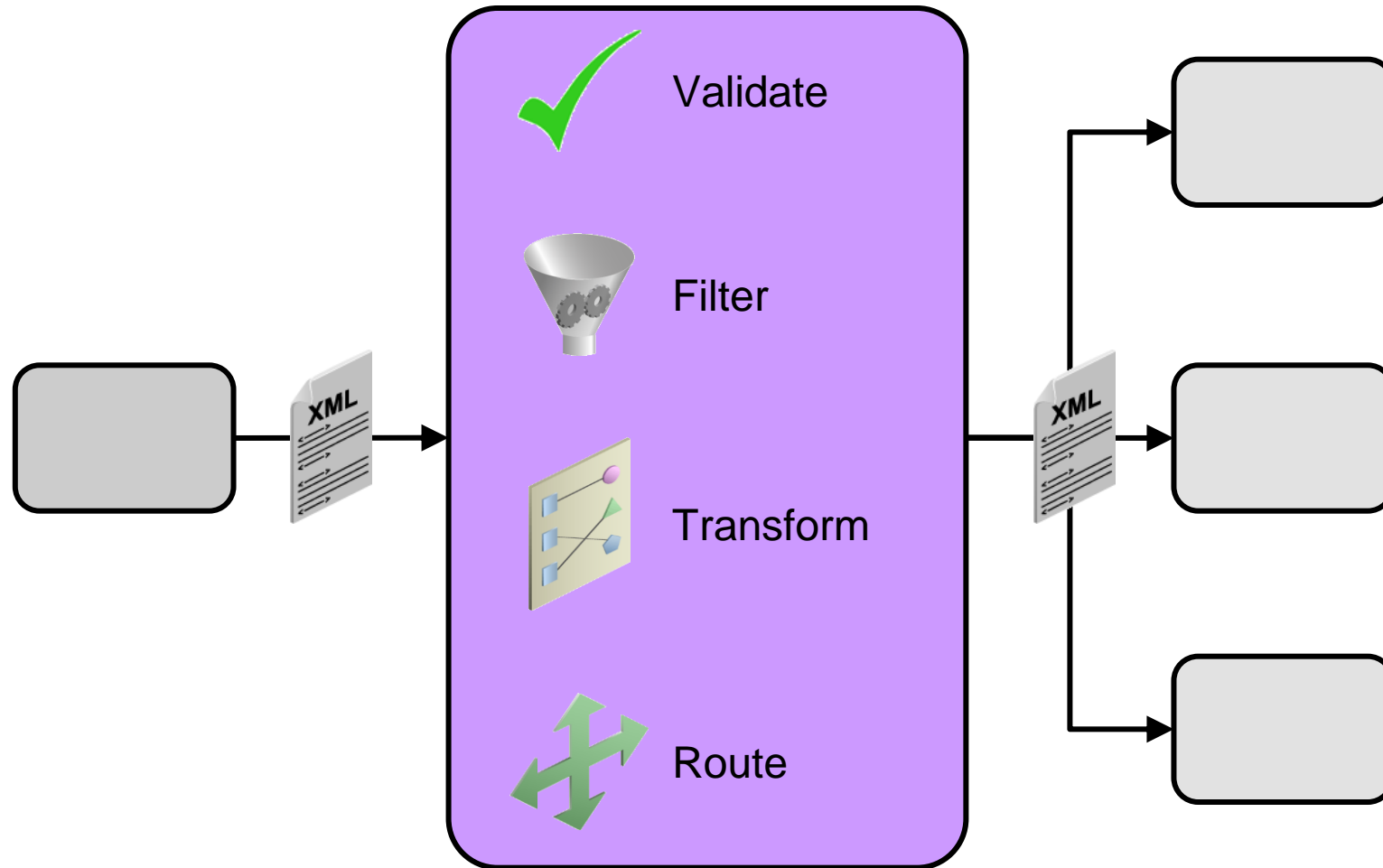


# Agenda

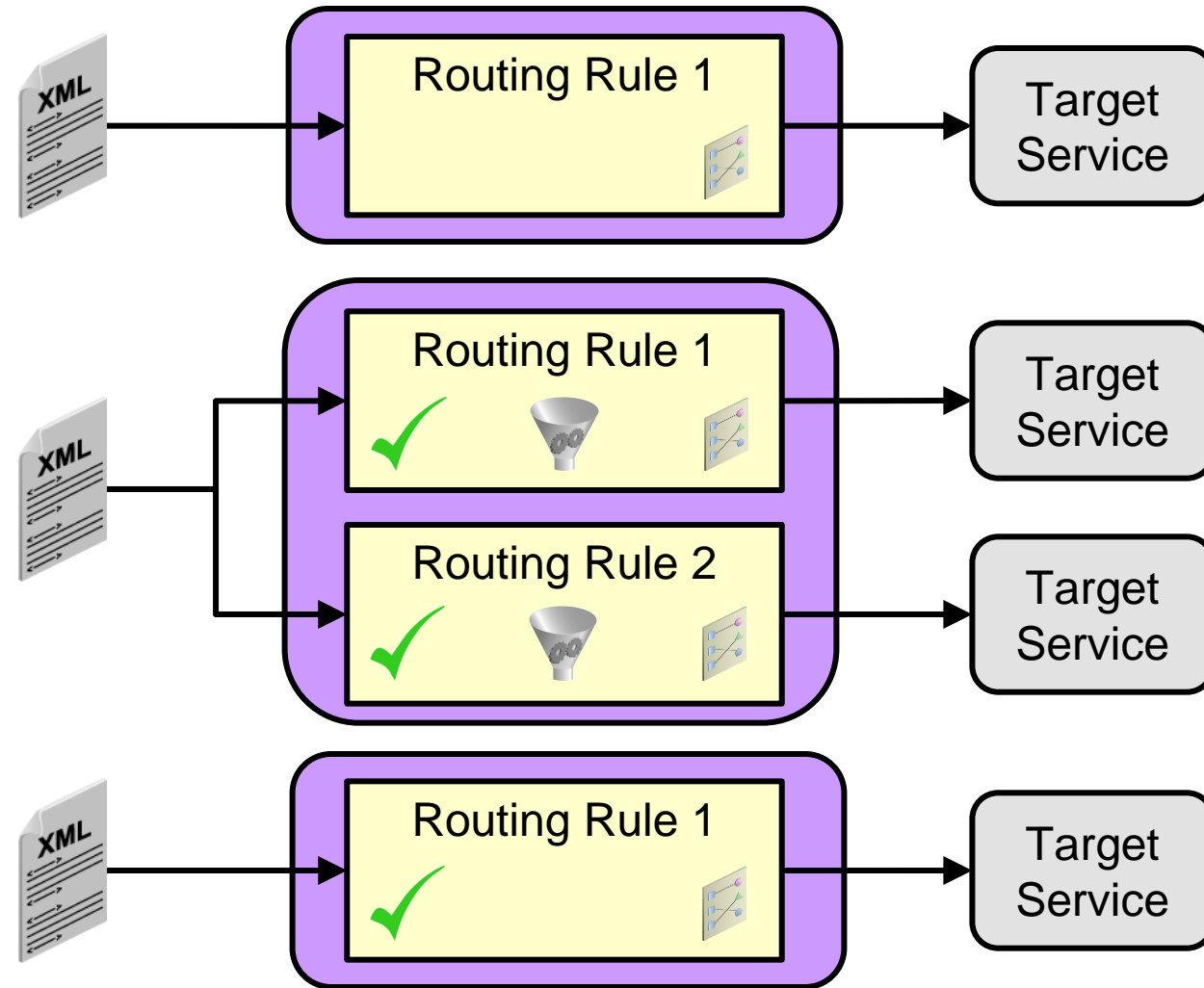
- How Services Use WSDL Files to Communicate
- **Mediator Components**
- File Adapters
- Composite Application Files



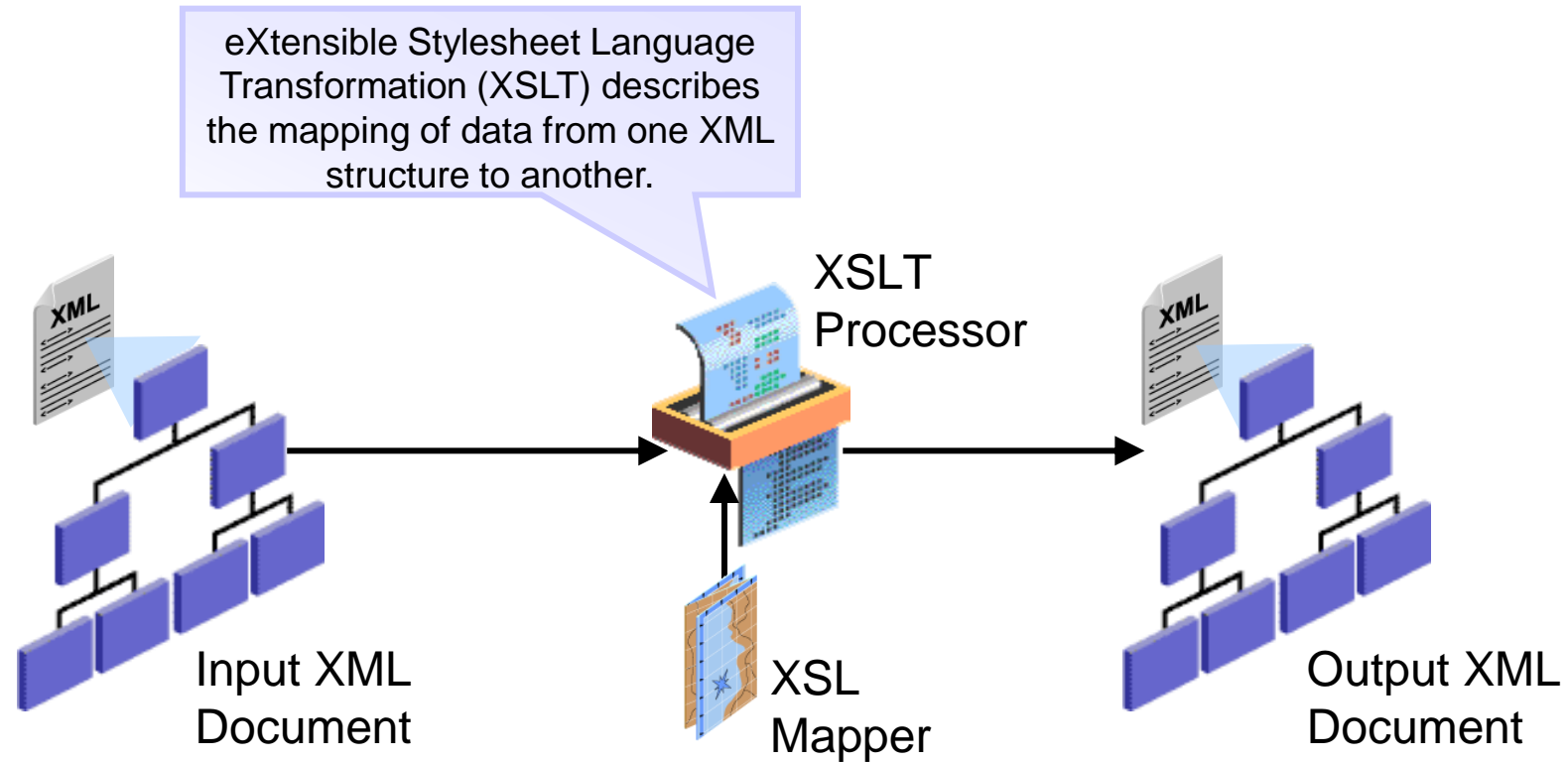
# Mediator Components: Introduction



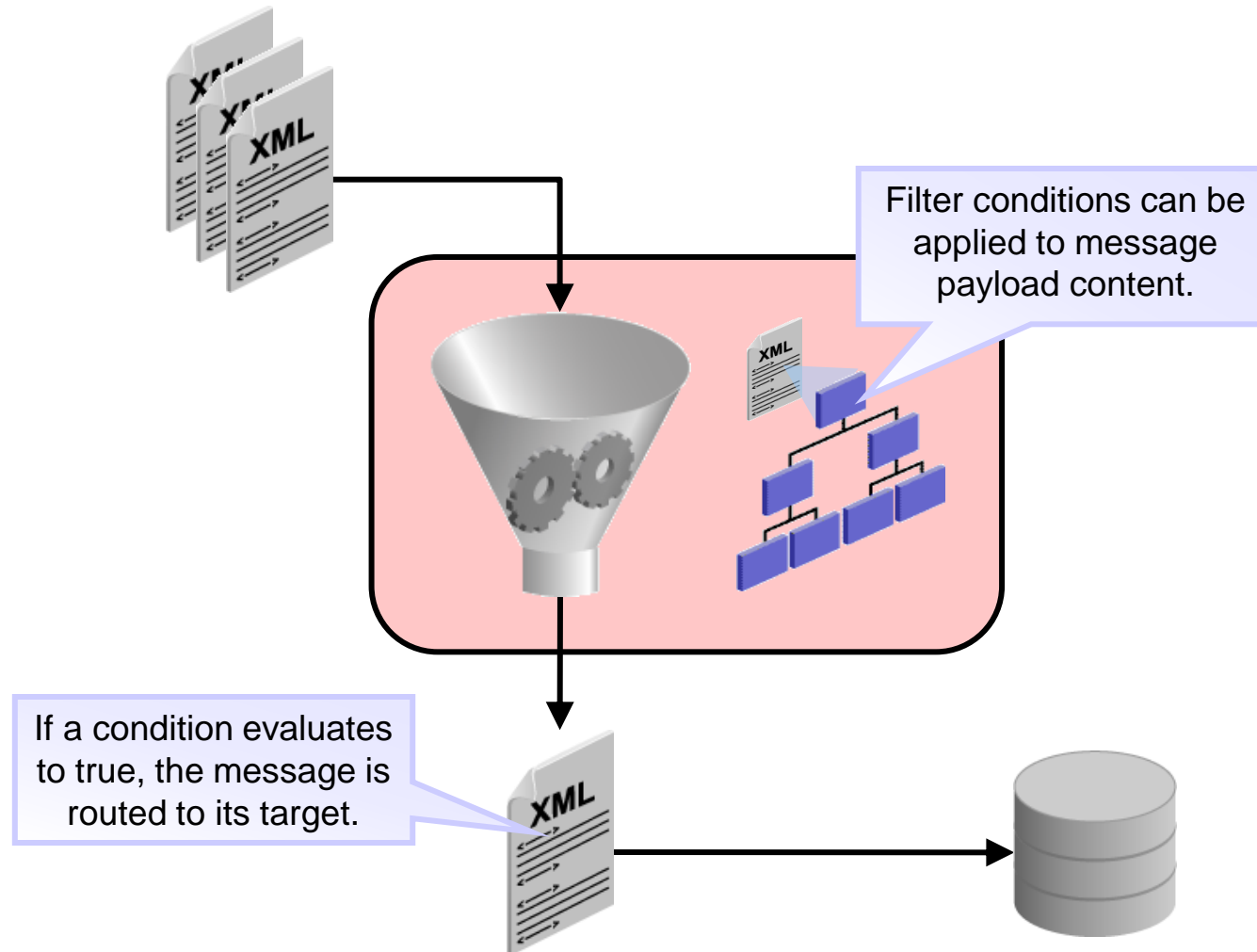
# Routing Data



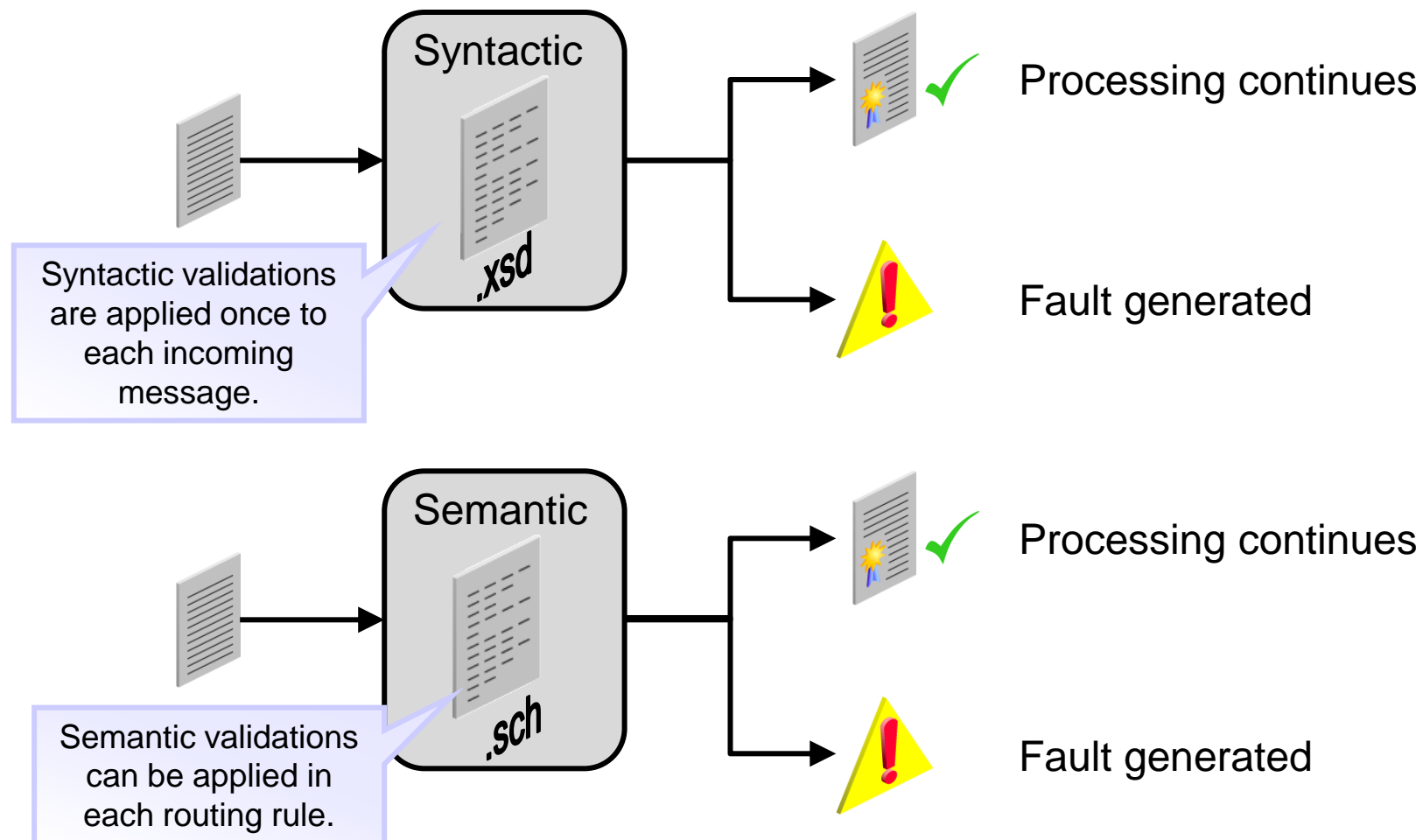
# Transforming Data



# Filtering Data

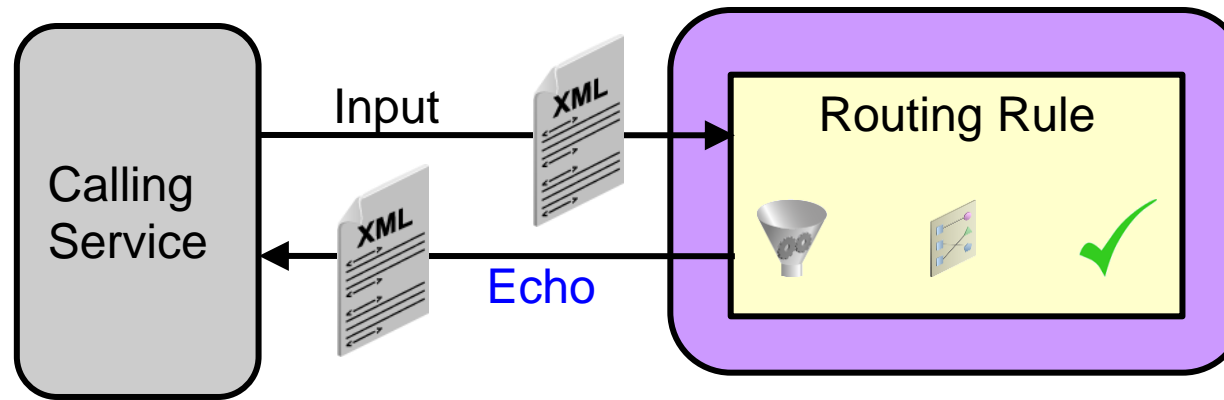


# Validating Data

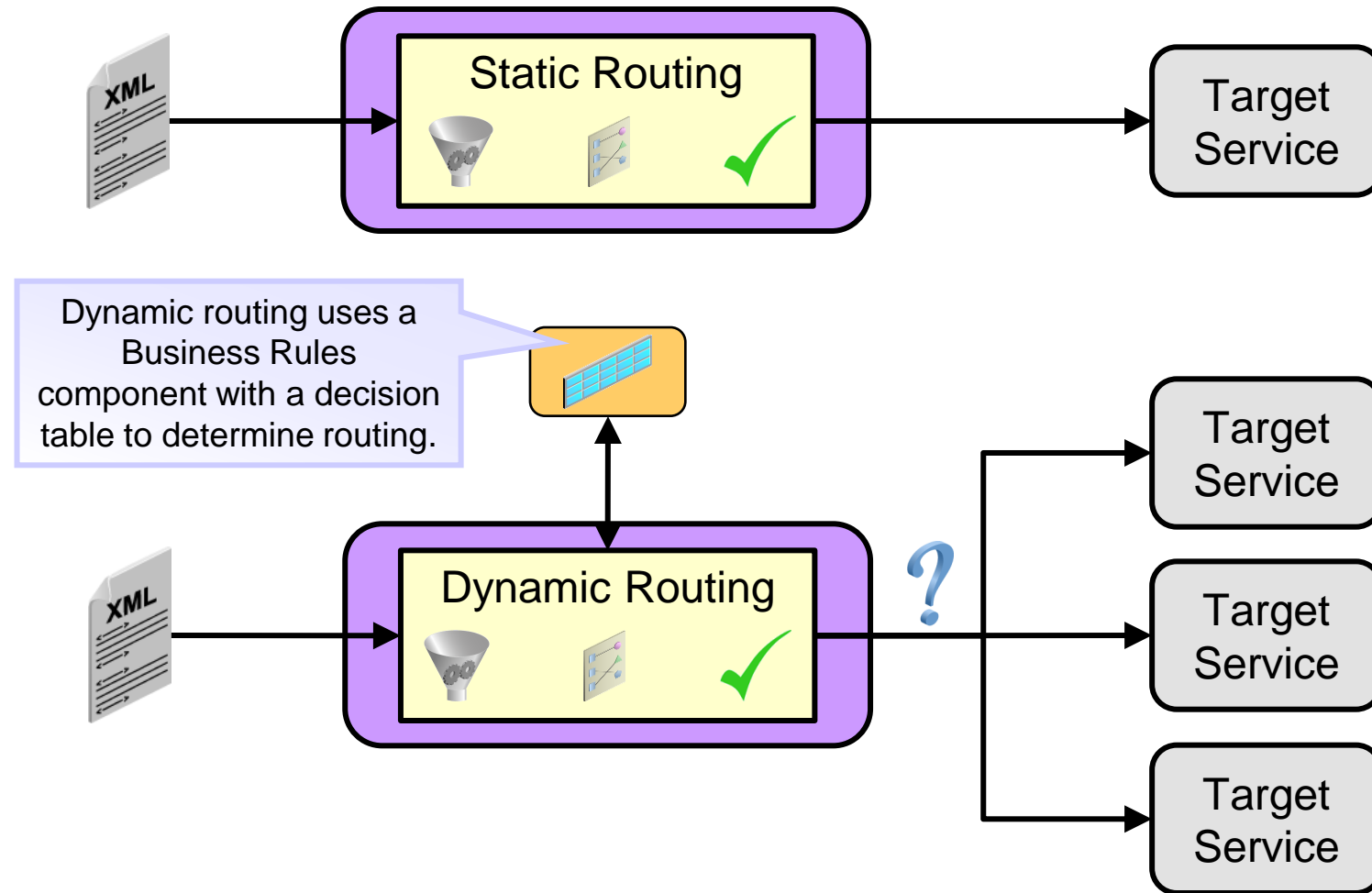




# Mediator as a Callable Service



# Dynamic and Static Routing Rules



# Additional Features of Mediators

Mediators provide the following additional features, which are covered in later lessons:

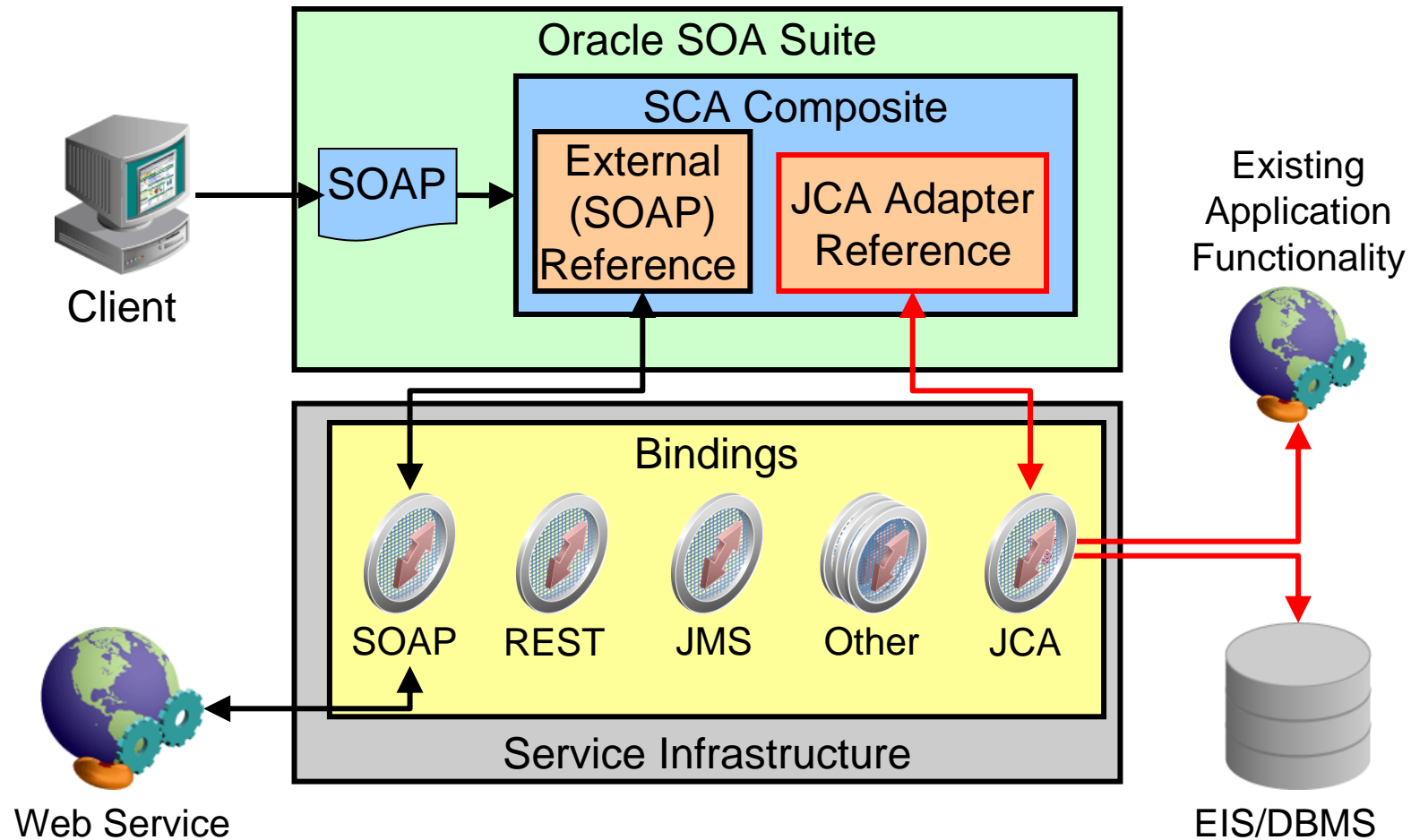
- Error routing and management
  - Mediator components support fault policy–based error handling. A fault policy consists of conditions and actions. Conditions specify the action to be carried out for a particular error condition.
- Event Handling
  - Mediator components provide support for subscribing to or raising business events that are delivered through the Oracle SOA Suite 12c Event Delivery Network (EDN).

# Agenda

- How Services Use WSDL Files to Communicate
- Mediator Components
- **File Adapters**
- Composite Application Files

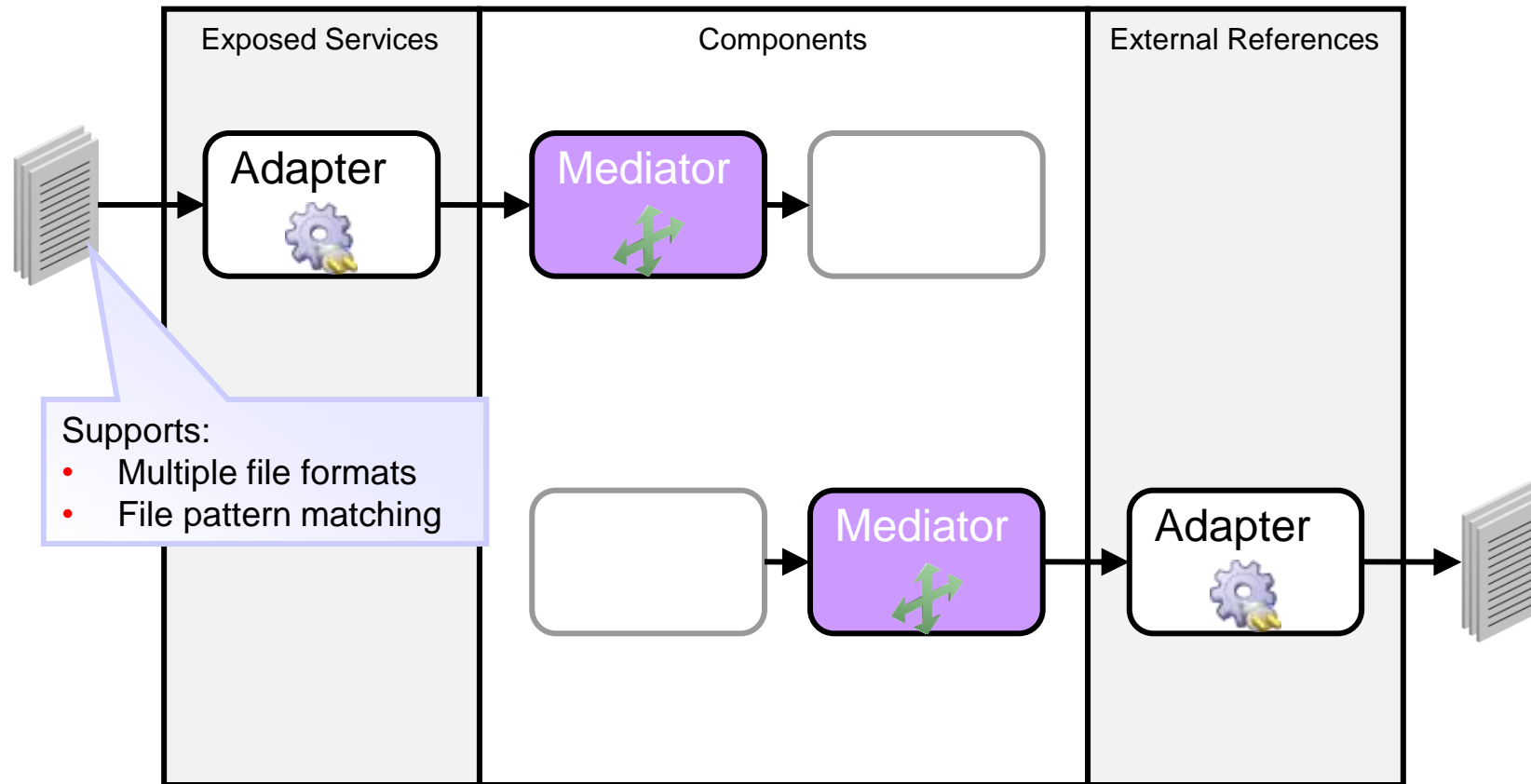


# Adapters: Introduction





# File Adapter



# Quiz



Which Mediator component feature enables target services to receive data in their desired format?

- a. Filter expression
- b. XSL transformation
- c. Validation
- d. Routing rule





# Quiz



The File adapter is capable of representing many file formats as XML data.

- a. True
- b. False



# Agenda

- How Services Use WSDL Files to Communicate
- Mediator Components
- File Adapters
- **Composite Application Files**



# Examining the composite.xml Source

```
[header]
<composite [attributes]>
  <import namespace=[namespace] location="ReceiveData.wsdl" importType="wsdl"/>
  <import namespace=[namespace] location="WriteData.wsdl" importType="wsdl"/>

  <service name="ReceiveData" ui:wsdlLocation="ReceiveData.wsdl">
    <interface.wsdl interface=[path]ReceiveData#wsdl.interface(execute_ptt)"/>
    <binding.ws port=[path]ReceiveData#wsdl.endpoint(ReceiveData/execute_pt) ">
      <property [config properties]>
    </binding.ws>
  </service>

  <component name="RouteData">
    <implementation.mediator src="RouteData.mplan"/>
  </component>

  <reference name="WriteData" ui:wsdlLocation="WriteData.wsdl">
    <interface.wsdl interface=[path]WriteData#wsdl.interface(Write_ptt)"/>
    <binding.jca config="WriteData_file.jca"/>
  </reference>

  <wire>
    <source.uri>ReceiveData</source.uri>
    <target.uri>RouteData/RouteData</target.uri>
  </wire>

  ...
</composite>
```

1

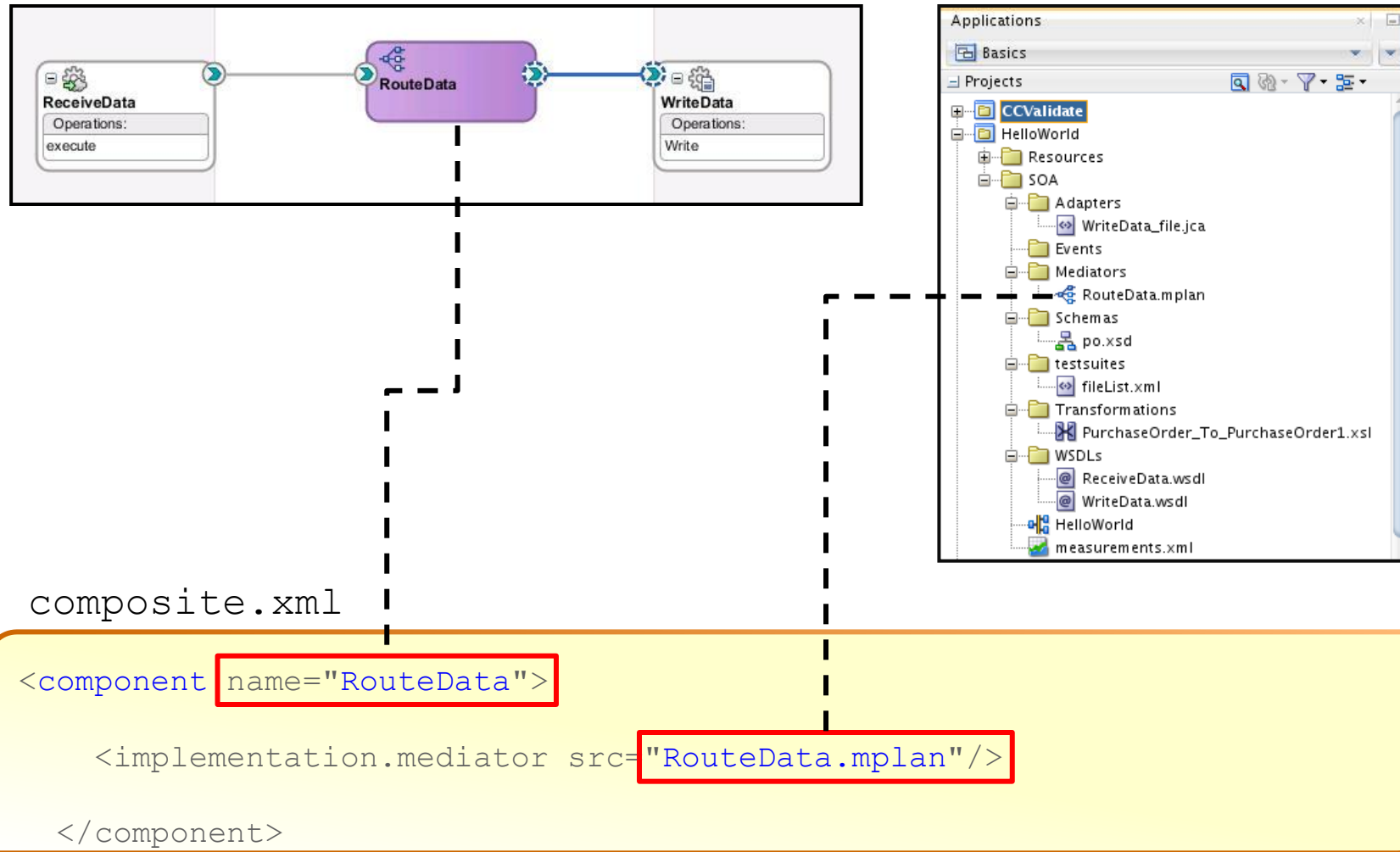
2

3

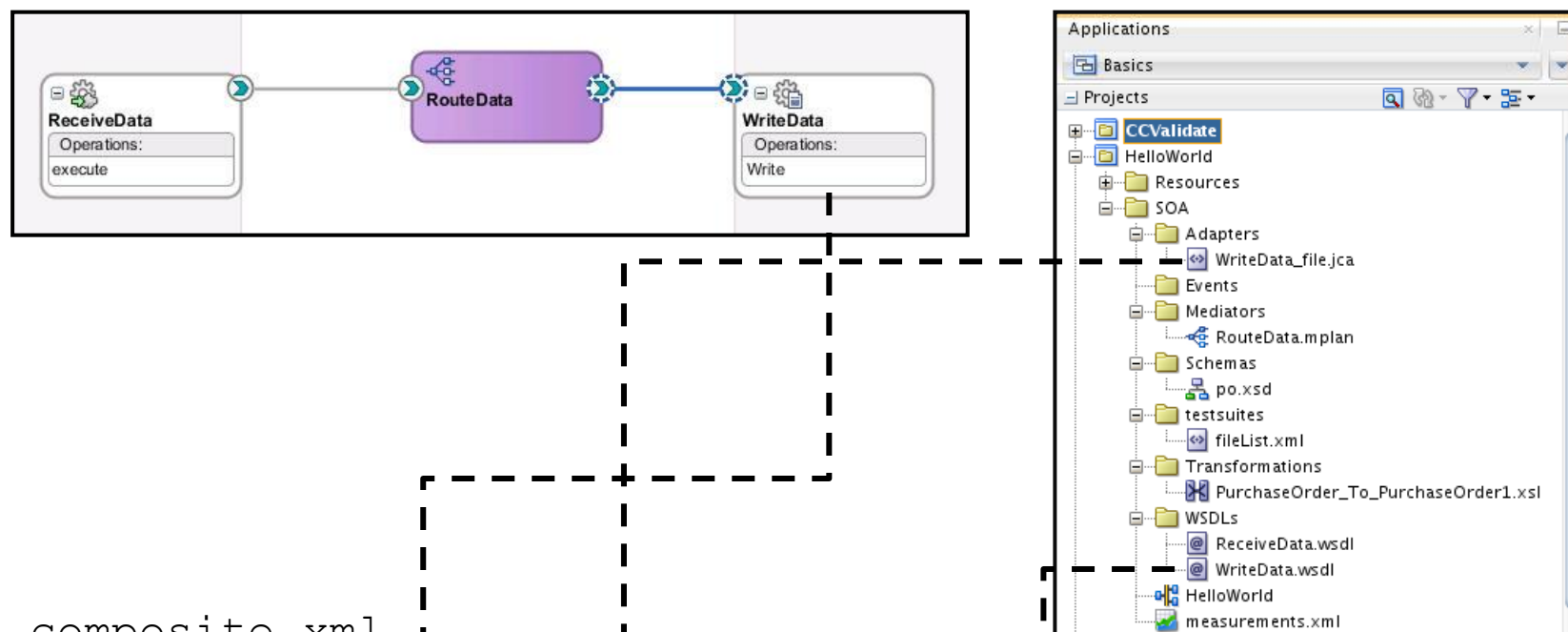
4

5

# Creating a Mediator



# Creating an Adapter



composite.xml

```
<composite [attributes]>
  . . .
  <reference name="WriteData" uri:wsdlLocation="WriteData.wsdl">
    <interface.wsdl interface=[path]WriteData#wsdl.interface(Write_ptt)"/>
    <binding.jca config="WriteData_file.jca"/>
  </reference>
  . . .
</composite>
```

# Summary

In this lesson, you should have learned how to:

- Describe how services use WSDL files to communicate
- Use Mediator components and File adapters to perform basic receiving, routing, and writing of messages
- Name and describe the contents of some of the files that are created as part of a composite application

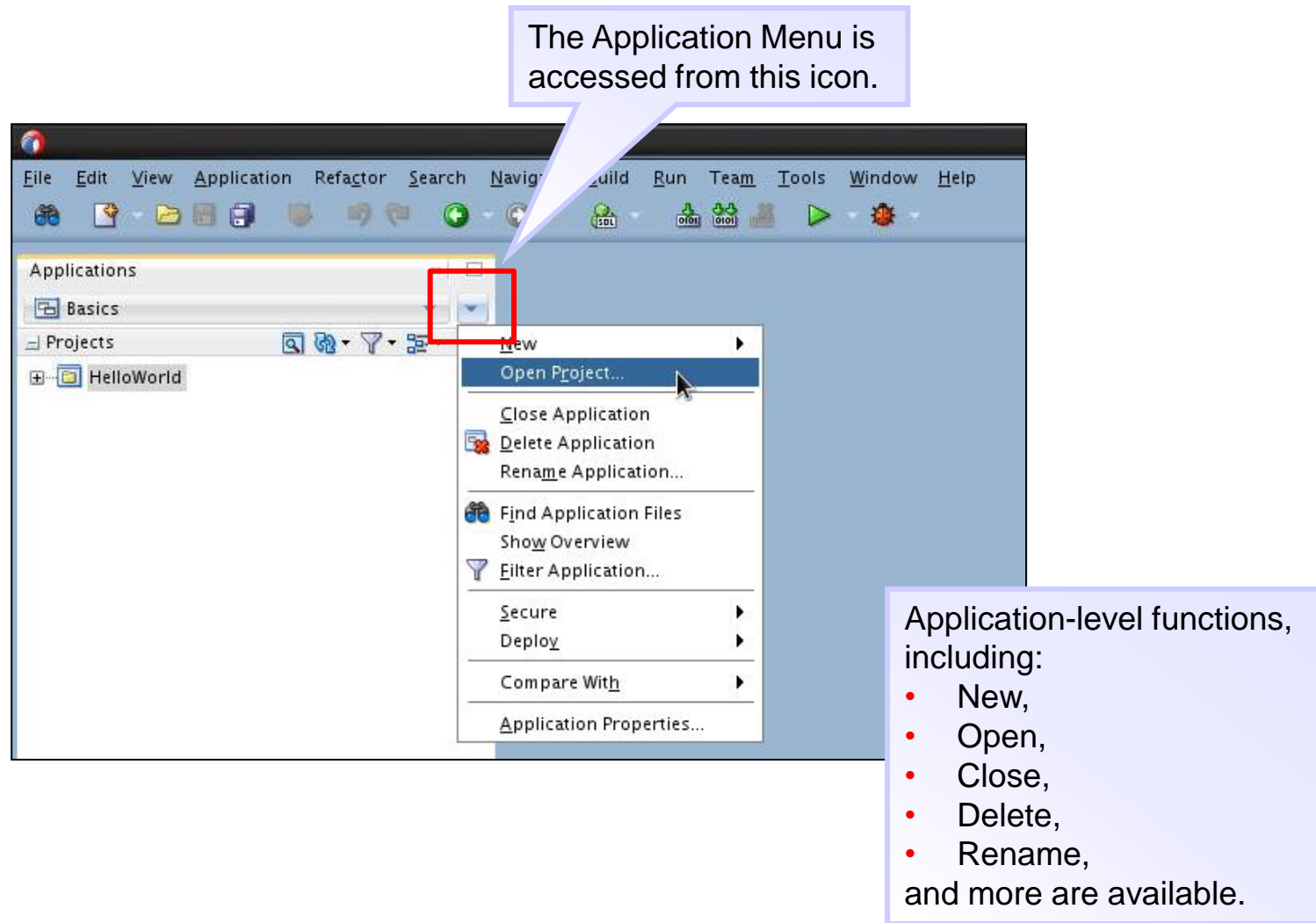


# Practice 2 Overview

This practice covers the following topics:

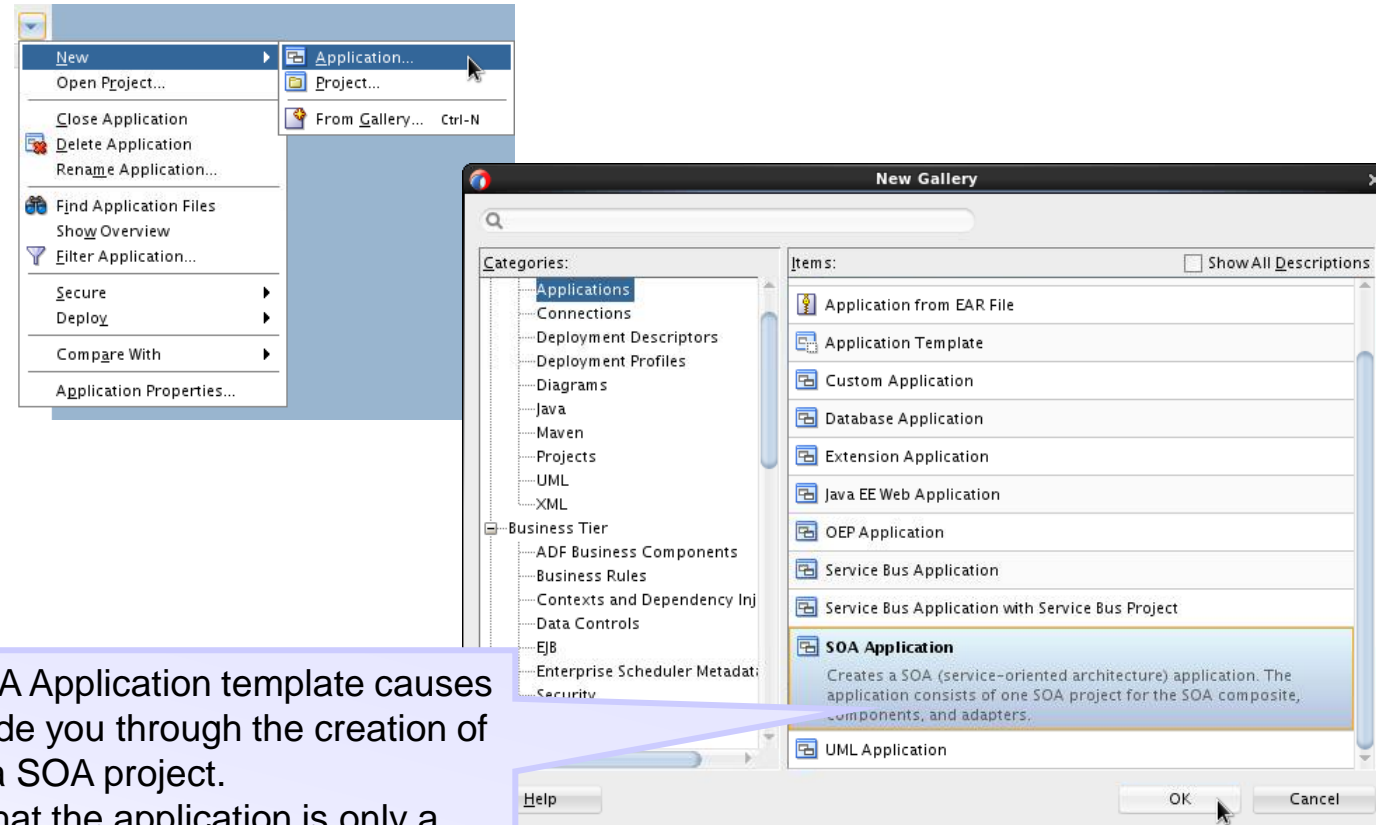
- Creating a Composite Application
- Adding a Mediator Component to a Composite Application
- Adding a File Adapter to a Composite Application
- Examining Composite Application Files

# Application Navigator





# Creating a SOA Application



Selecting the SOA Application template causes the wizard to guide you through the creation of a SOA project.

(Remember that the application is only a container. The “real work” is defined in the projects that you add to that container.)

# Creating a SOA Application

The image displays three sequential screenshots of the 'Create SOA Application' wizard, numbered 1, 2, and 3 in orange circles.

**Step 1 of 3: Name your application**

Application Name: Basics

Directory: /u01/app/fmw12c/domains/myWork/mywork/Basics

Application Package Prefix:

**Step 2 of 3: Name your project**

Project Name: HelloWorld

Directory: /fmw12c/domains/myWork/mywork/Basics/HelloWorld

Project Features:

- SOA Suite
- SOA Suite is a suite of tools to model SOA(Service Oriented Architecture) applications.

**Step 3 of 3: Configure SOA settings**

Composite Name: HelloWorld

Start from: ☒ Standard Composite ☐ SOA Template

☒ Empty Composite

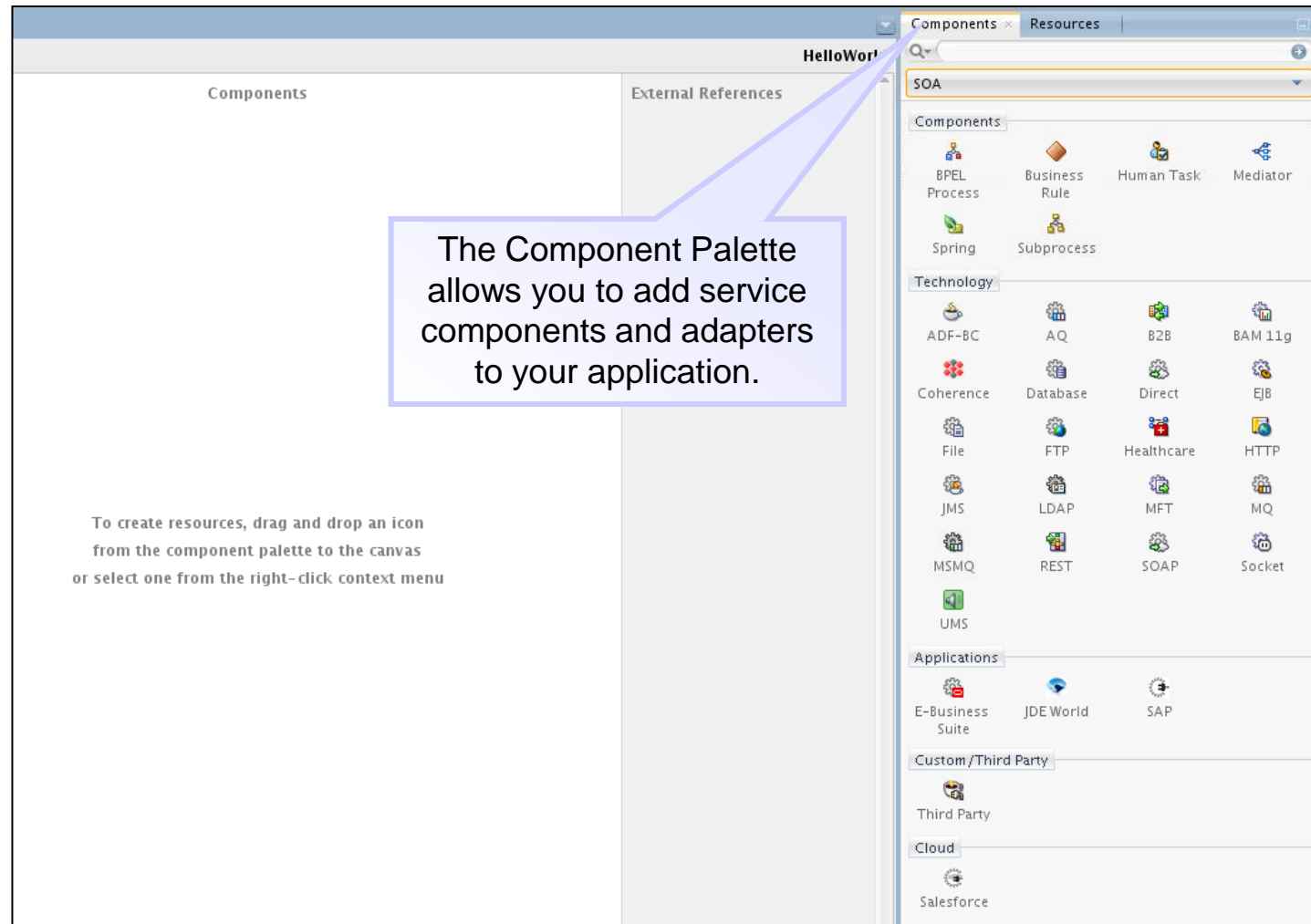
- Composite With Spring
- Composite With Human Task
- Composite With BPEL Process
- Composite With Subprocess
- Composite With Business Rule
- Composite With Mediator

The first steps in building a new application are to assign it a name and to specify the directory in which to save the source files.

Add a project to the application, assigning it a name and directory as well.

Specify the starting point for the project.

# Component Palette



# Creating a WSDL

**Create WSDL**

File Name:

Directory:

Namespace:

Port Type:

Operation:

Interface Type:

Input:

Message Part Name	Element or Type	Schema URL
request	PurchaseOrder	Sc

☐ Generate partnerlinkType extension

The Create WSDL wizard builds the necessary WSDL files automatically. You specify XSD files to describe the messages and modify any other values that you desire.

# Creating a File Adapter

FILE Adapter Configuration Wizard - Step 5 of 7

### File Configuration

Specify the parameters for the Write File operation.

Directory specified as ☒ Physical Path ☐ Logical Name

Directory for Outgoing Files (physical path):

File Naming Convention (po\_%SEQ%.txt):

☐ Append to existing file

Write to output file when any of these conditions are met:

<input checked="" type="checkbox"/> Number of Messages Equals:	<input type="text" value="1"/>	
<input type="checkbox"/> Elapsed Time Exceeds:	<input type="text" value="1"/>	<input type="text" value="minutes"/>
<input type="checkbox"/> File Size Exceeds:	<input type="text" value="1000"/>	<input type="text" value="kilobytes"/>

The File Adapter wizard guides you through the configuration of an adapter.

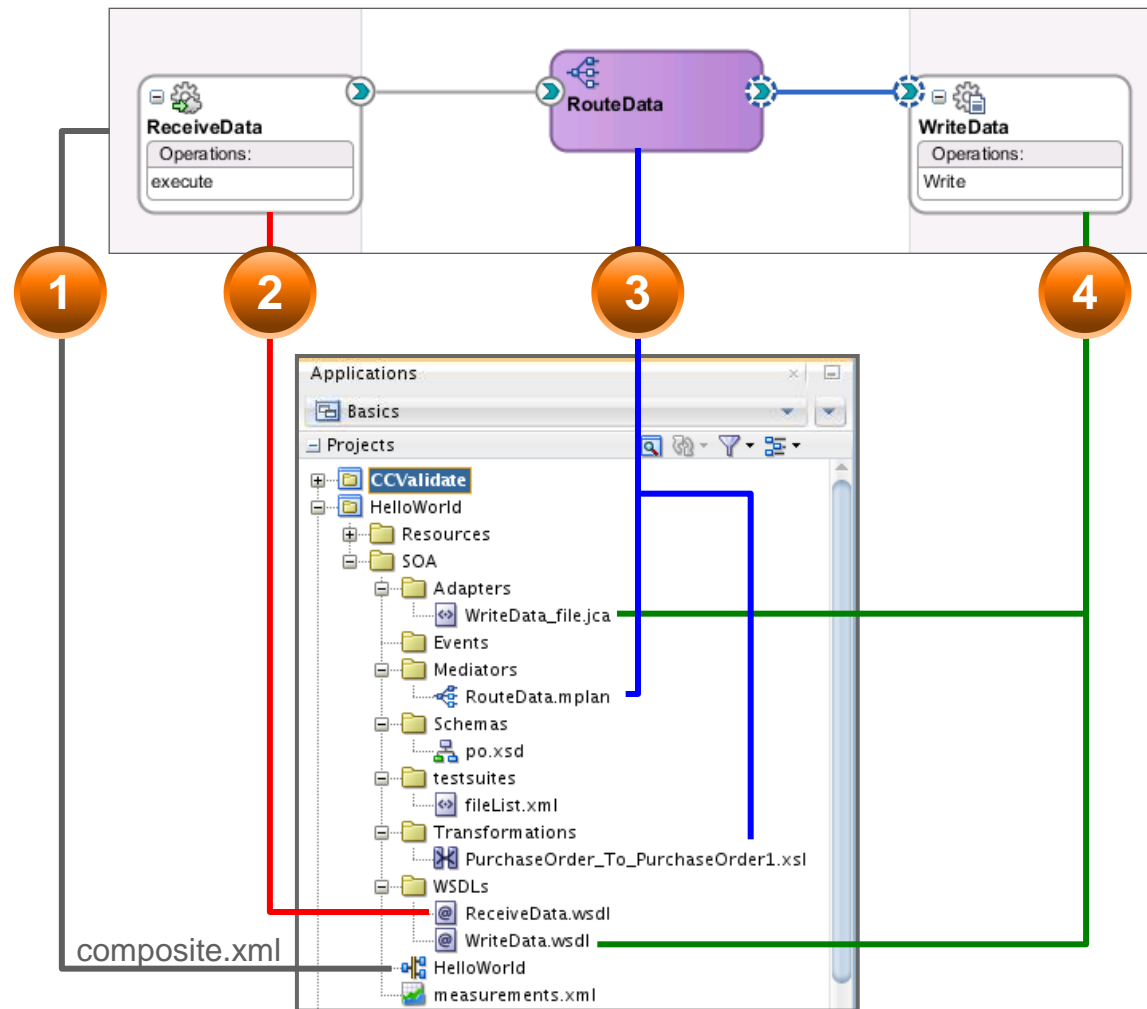
# Configuring Mediator Routing Rules

Add or delete routing rules as needed.

The screenshot shows the 'Routing Rules' configuration window in Oracle Mediator. The window has a title bar with a gear icon and the text 'Routing Rules'. Below the title bar is a tab labeled 'Operations'. The main area is divided into two sections. The top section is for the 'execute' operation, with a 'Priority' of 4 and a 'Validate Syntax (XSD)' checkbox. It contains fields for 'Translate From Native' (set to '<<No Translation Needed>>'), 'Callout To' (set to '<<Java Callout Class>>'), and 'Resequence' (set to 'Off'). The bottom section is for a specific routing rule, with a 'Filter Expression' field set to '<<Filter Expression>>'. To the right of the filter expression is a 'WriteFile::Write' button. Below this are several configuration options: 'Validate Semantic' (a dropdown menu), 'Translate To Native' (set to '<<No Translation Needed>>'), 'Transform Using' (set to '<<Transformation Map>> --> body'), 'Assign Values' (a dropdown menu), and 'Override Using' (a dropdown menu). On the right side of the bottom section, there are several icons: a gear, a person, a document, a folder, and a green arrow. A 'Sequential' dropdown menu is also present on the right.

Specify the validation, filtering, and transformation details for each Mediator routing rule.

# Project Files



# Project Files

