8

Deploying and Testing Composite Applications

Objectives

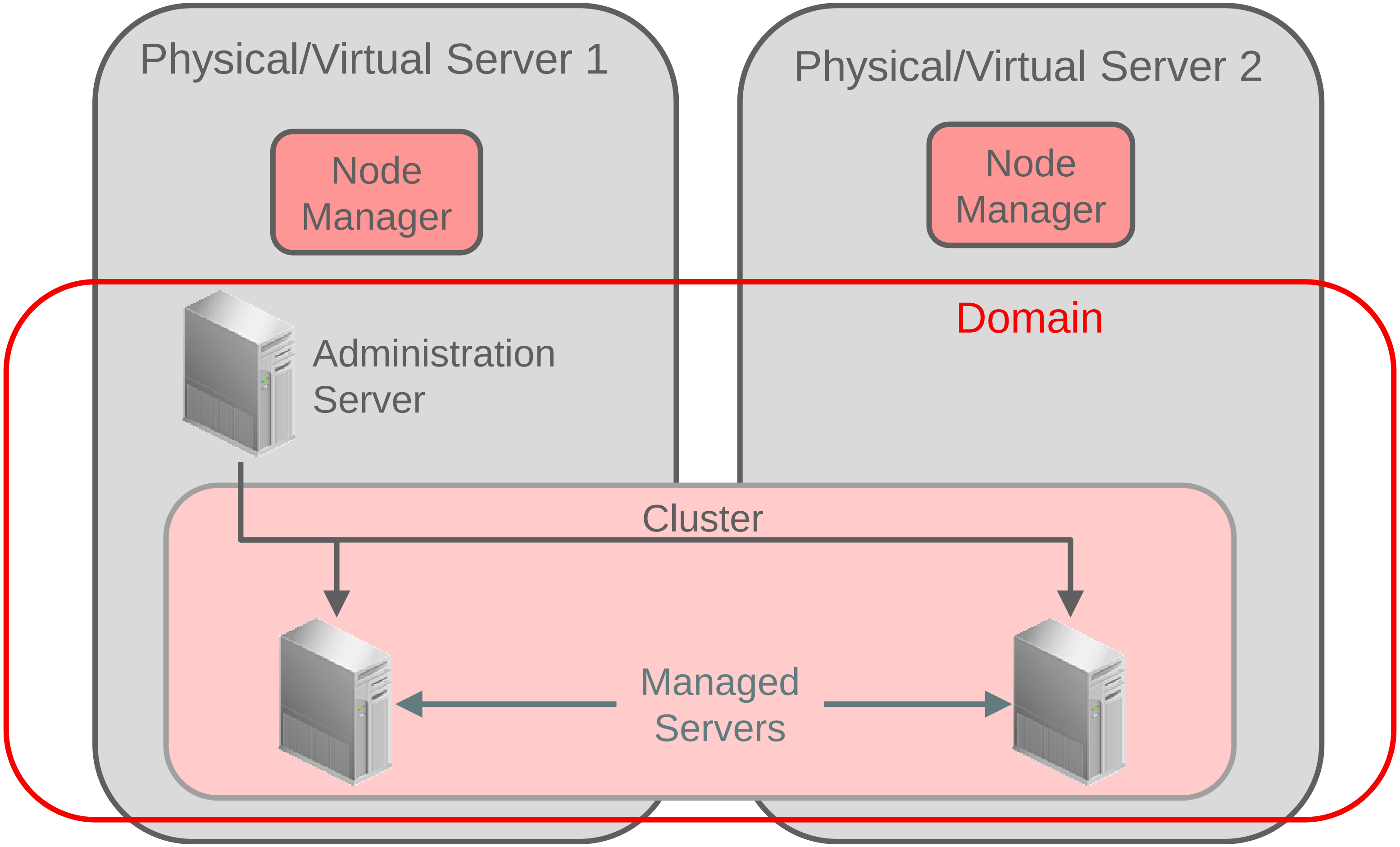
After completing this lesson, you should be able to:

- Deploy a composite application by using Ant
- Test a composite application in Enterprise Manager
- Track message flow through a composite application
- Undeploy a composite application by using Ant
- Describe a composite application package and its dependencies

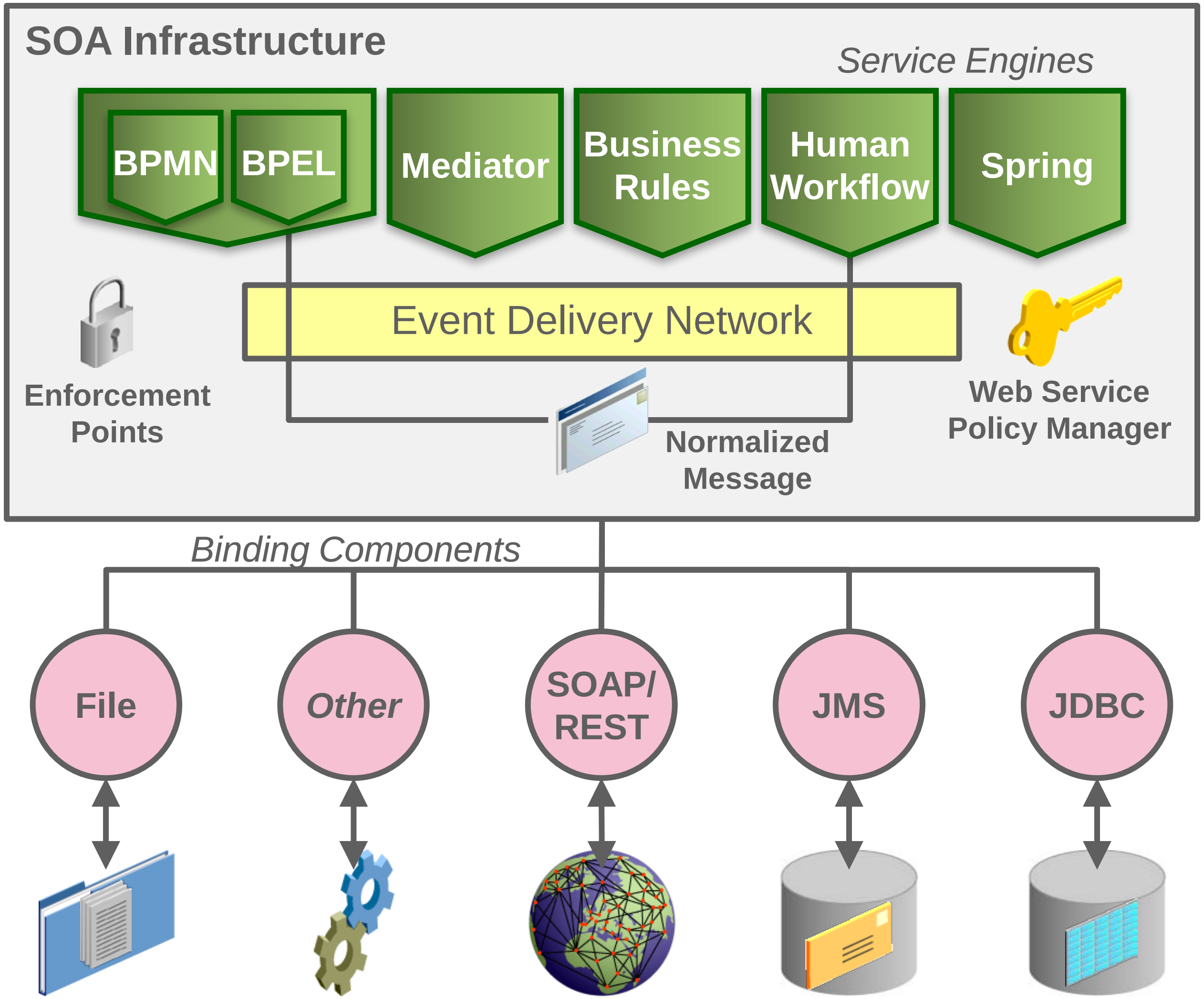
Agenda

- The Runtime Environment
- Packaging and Deploying a Composite Application
- Tools Used by the SOA Administrator

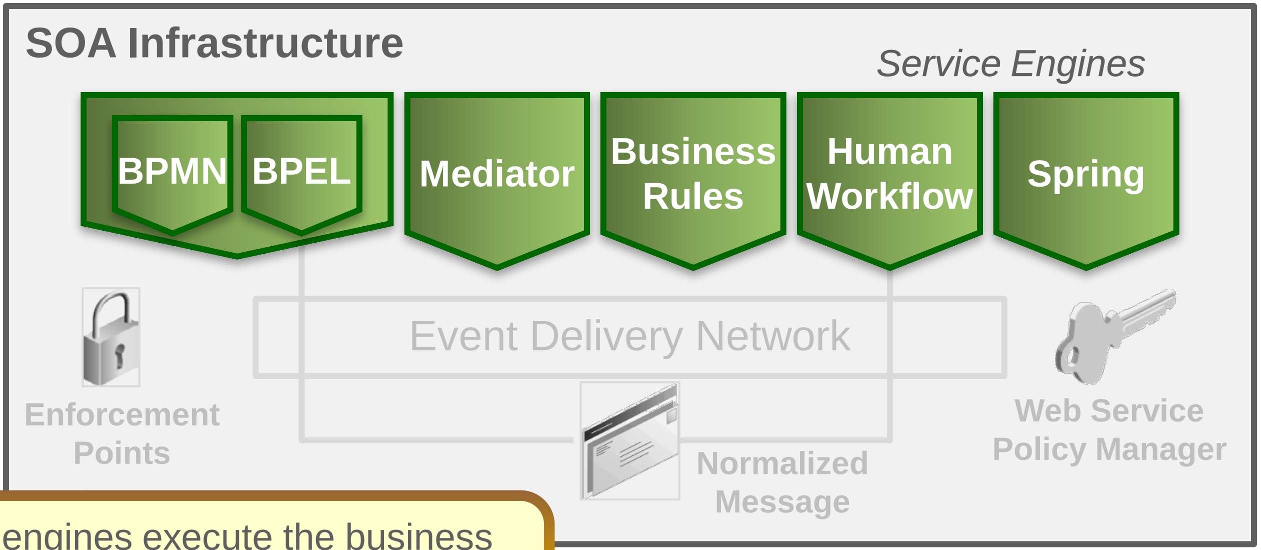
WebLogic Application Server



SOA Infrastructure



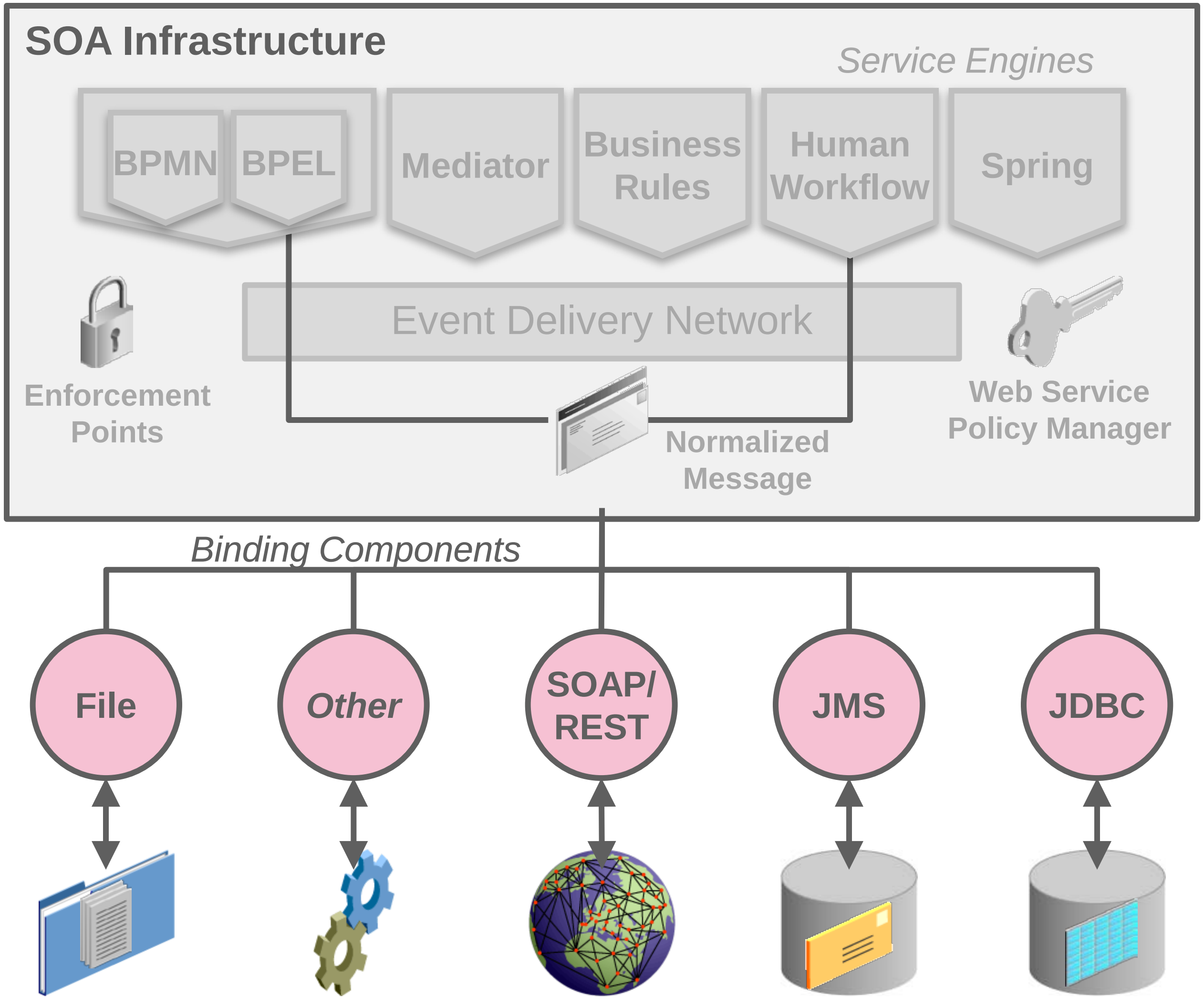
Service Engines



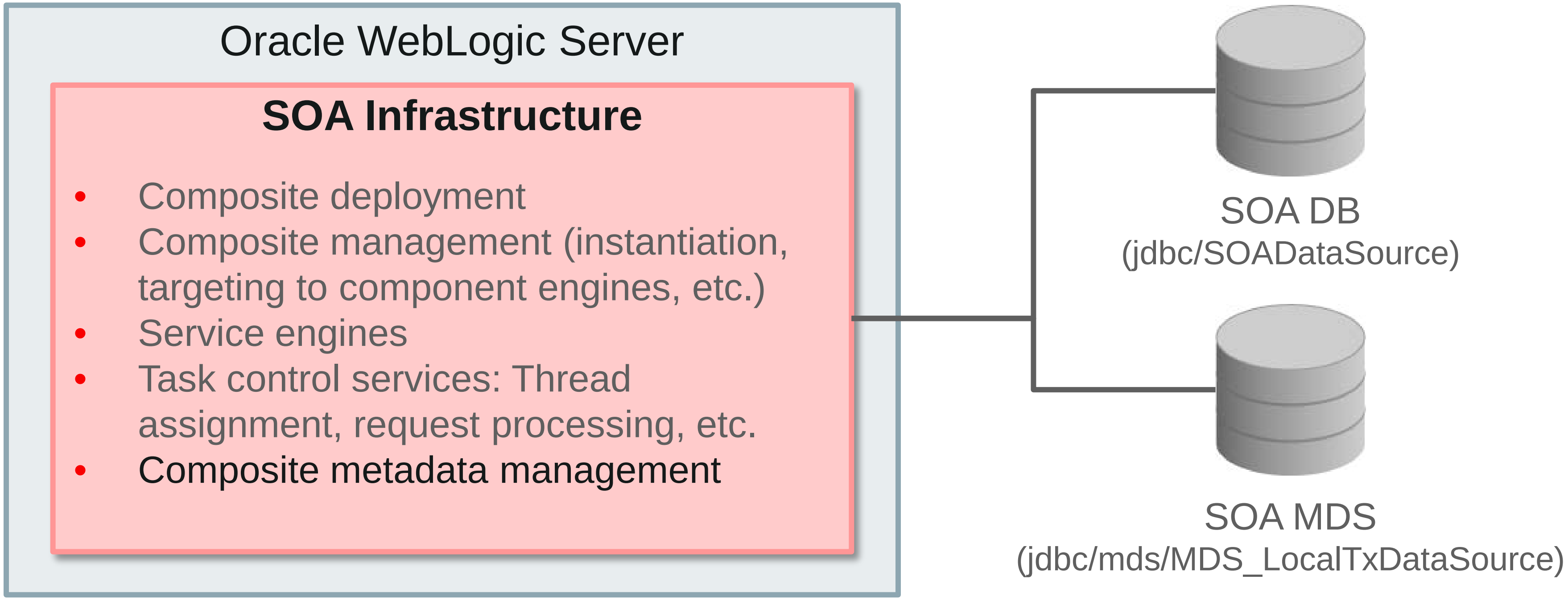
Service engines execute the business logic of their respective components within the SOA composite application.

In Enterprise Manager, you can monitor service components and composites, view message request and thread statistics, and configure properties that are specific to a service engine.

Binding Components



SOA Infrastructure Database Connections



Oracle WebLogic Server

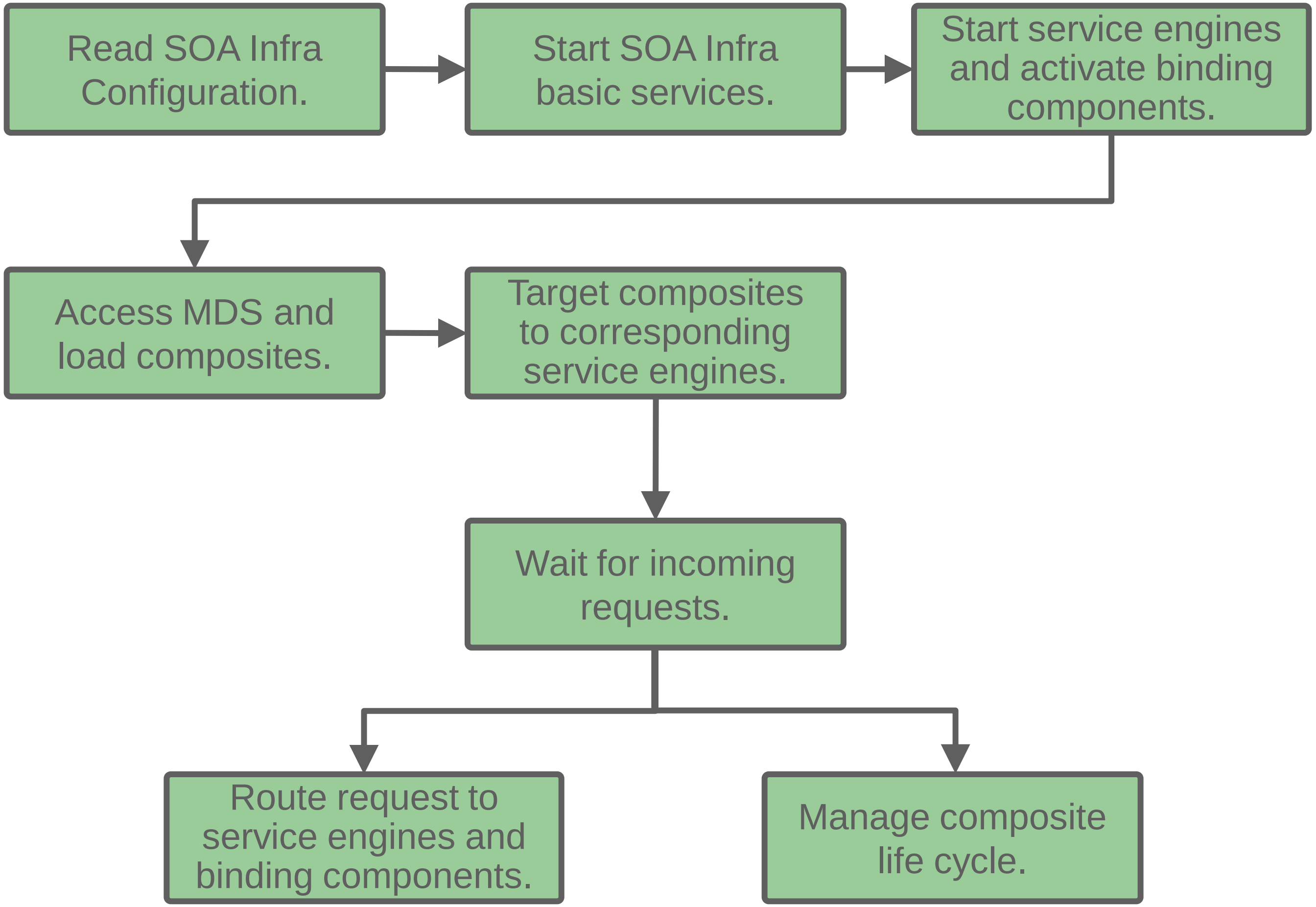
SOA Infrastructure

- Composite deployment
- Composite management (instantiation, targeting to component engines, etc.)
- Service engines
- Task control services: Thread assignment, request processing, etc.
- Composite metadata management

SOA DB
(jdbc/SOADataSource)

SOA MDS
(jdbc/mds/MDS_LocalTxDataSource)

SOA Infrastructure Life Cycle



Read SOA Infra Configuration.

Start SOA Infra basic services.

Start service engines and activate binding components.

Access MDS and load composites.

Target composites to corresponding service engines.

Wait for incoming requests.

Route request to service engines and binding components.

Manage composite life cycle.

Quiz

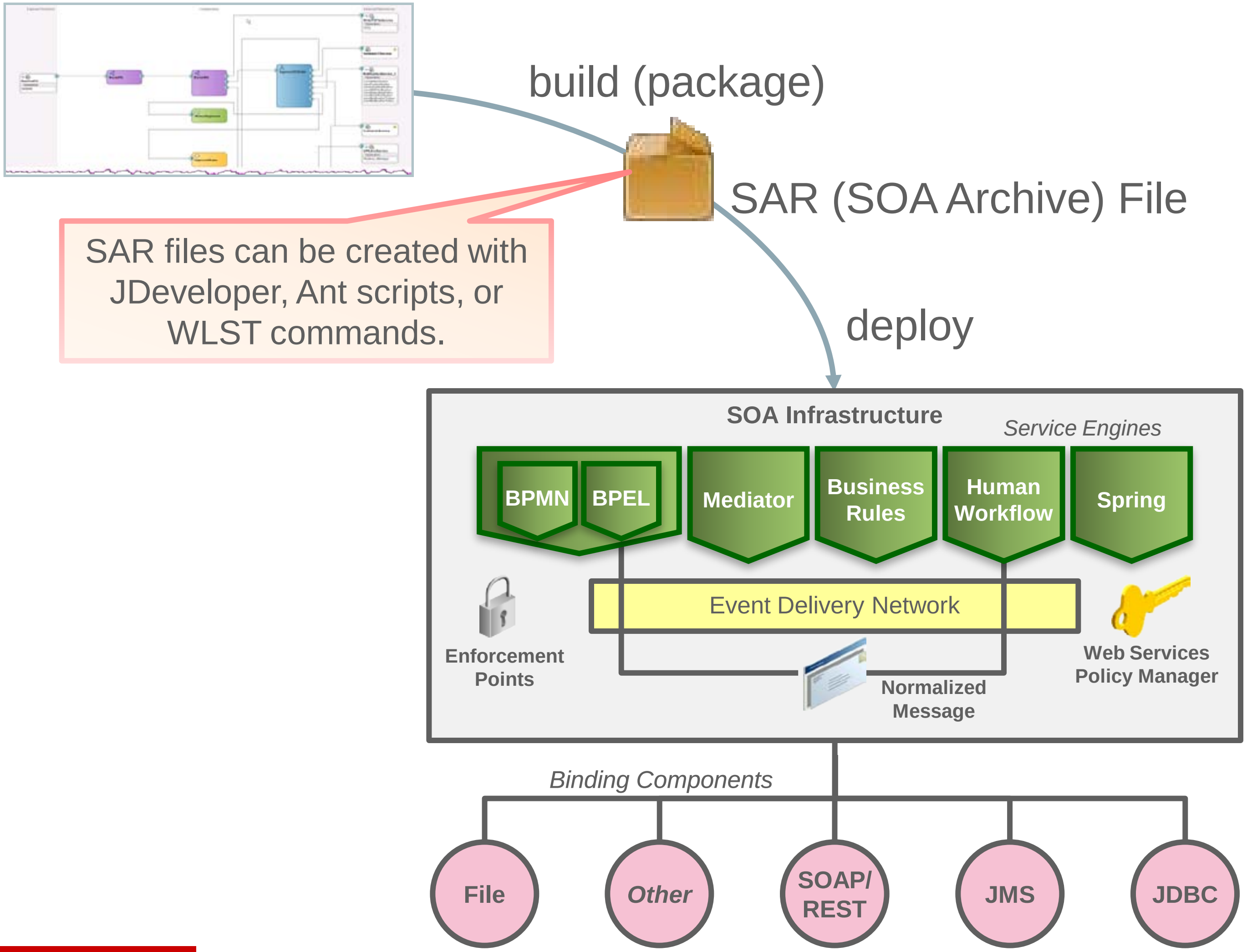
Which of the following services is not provided by SOA Infrastructure?

a. Composite deployment
b. Notification
c. Composite lifecycle management
d. Communication across service components

Agenda

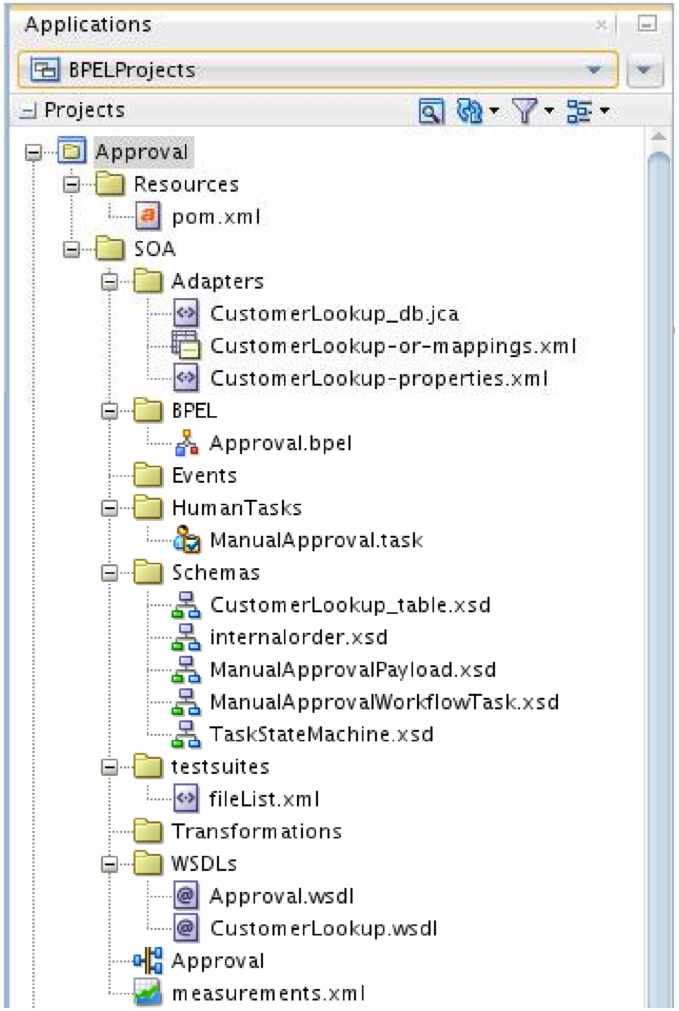
- The Runtime Environment
- Packaging and Deploying a Composite Application
- Tools Used by the SOA Administrator

Packaging and Deploying a Composite Application

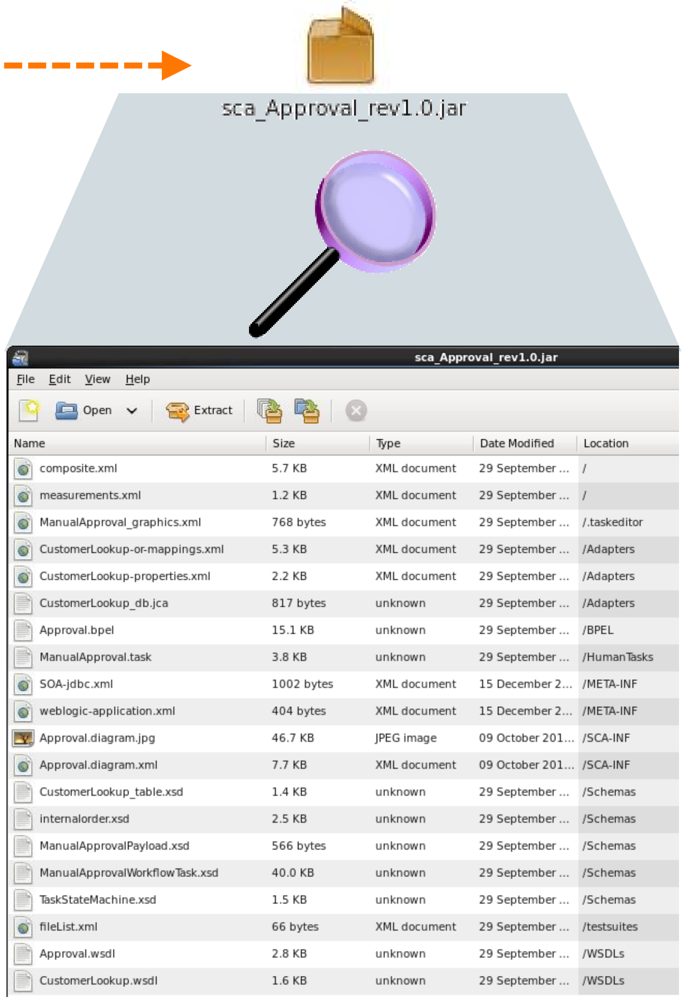


SAR files can be created with JDeveloper, Ant scripts, or WLST commands.

Building a Service Archive



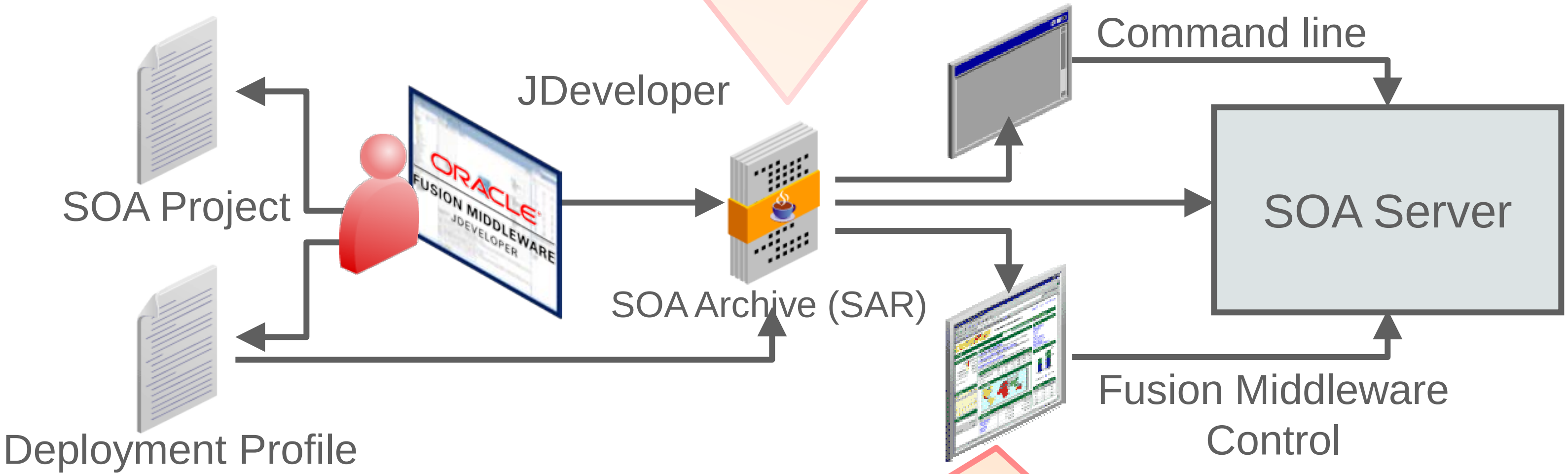
- Approval
  - Resources
    - pom.xml
  - SOA
    - Adapters
      - CustomerLookup_db.jca
      - CustomerLookup-or-mappings.xml
      - CustomerLookup-properties.xml
    - BPEL
      - Approval.bpel
    - Events
    - HumanTasks
      - ManualApproval.task
    - Schemas
      - CustomerLookup_table.xsd
      - internalorder.xsd
      - ManualApprovalPayload.xsd
      - ManualApprovalWorkflowTask.xsd
      - TaskStateMachine.xsd
    - testsuites
      - fileList.xml
    - Transformations
    - WSDLs
      - Approval.wsdl
      - CustomerLookup.wsdl
    - Approval
    - measurements.xml



Name	Size	Type	Date Modified	Location
composite.xml	5.7 KB	XML document	29 September ...	/
measurements.xml	1.2 KB	XML document	29 September ...	/
ManualApproval_graphics.xml	768 bytes	XML document	29 September ...	/.taskeditor
CustomerLookup-or-mappings.xml	5.3 KB	XML document	29 September ...	/Adapters
CustomerLookup-properties.xml	2.2 KB	XML document	29 September ...	/Adapters
CustomerLookup_db.jca	817 bytes	unknown	29 September ...	/Adapters
Approval.bpel	15.1 KB	unknown	29 September ...	/BPEL
ManualApproval.task	3.8 KB	unknown	29 September ...	/HumanTasks
SOA-jdbc.xml	1002 bytes	XML document	15 December 2...	/META-INF
weblogic-application.xml	404 bytes	XML document	15 December 2...	/META-INF
Approval.diagram.jpg	46.7 KB	JPEG image	09 October 201...	/SCA-INF
Approval.diagram.xml	7.7 KB	XML document	09 October 201...	/SCA-INF
CustomerLookup_table.xsd	1.4 KB	unknown	29 September ...	/Schemas
internalorder.xsd	2.5 KB	unknown	29 September ...	/Schemas
ManualApprovalPayload.xsd	566 bytes	unknown	29 September ...	/Schemas
ManualApprovalWorkflowTask.xsd	40.0 KB	unknown	29 September ...	/Schemas
TaskStateMachine.xsd	1.5 KB	unknown	29 September ...	/Schemas
fileList.xml	66 bytes	XML document	29 September ...	/testsuites
Approval.wsdl	2.8 KB	unknown	29 September ...	/WSDLs
CustomerLookup.wsdl	1.6 KB	unknown	29 September ...	/WSDLs

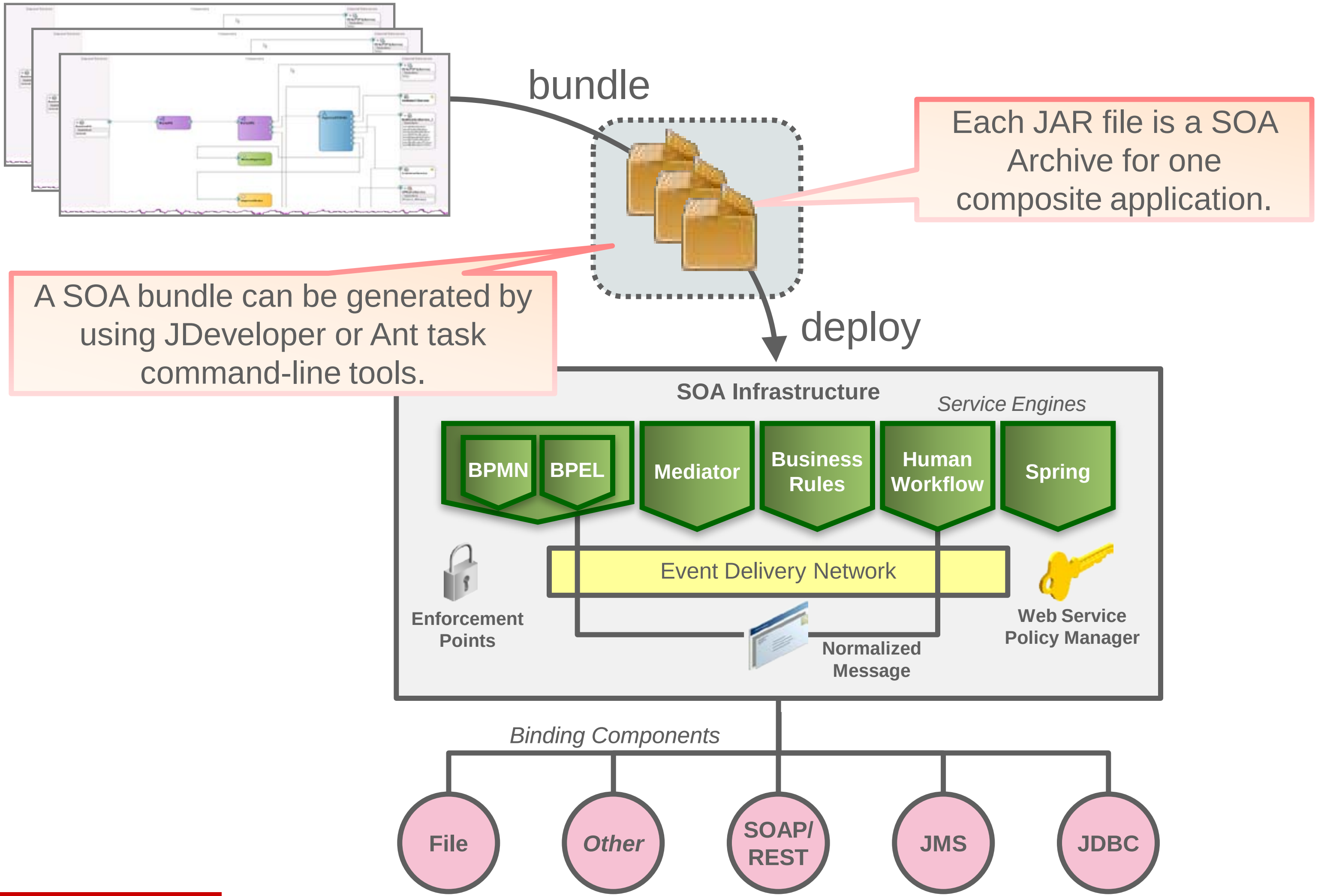
Deploying a Composite Application

The application is deployed as a SOA Archive (SAR file) that includes all service components in a single application package.



The application can be deployed by using JDeveloper, Enterprise Manager, or command-line tools such as Ant and WLST.

Packaging Multiple SOA Projects



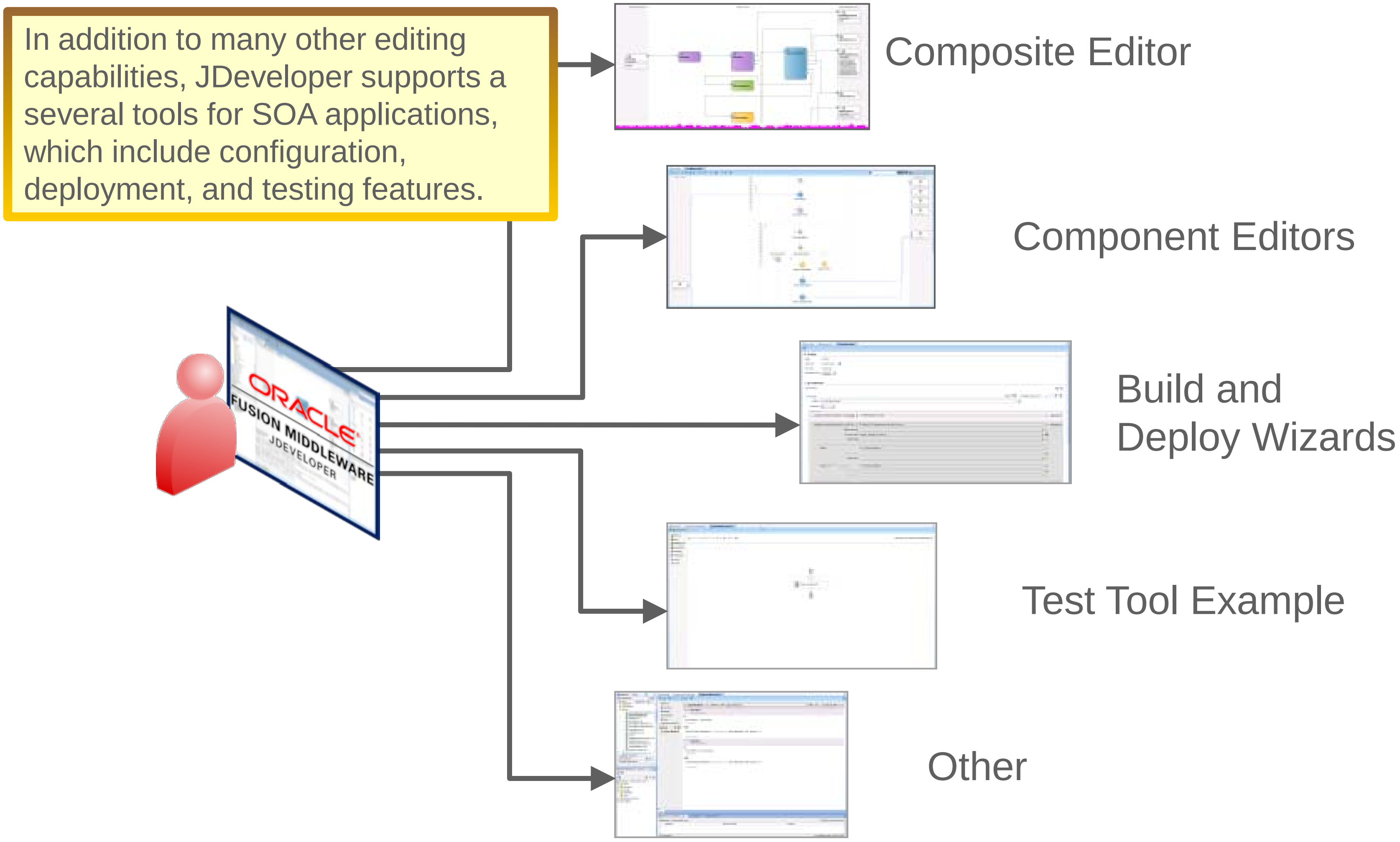
Each JAR file is a SOA Archive for one composite application.

A SOA bundle can be generated by using JDeveloper or Ant task command-line tools.

Agenda

- The Runtime Environment
- Packaging and Deploying a Composite Application
- Tools Used by the SOA Administrator

JDeveloper



In addition to many other editing capabilities, JDeveloper supports a several tools for SOA applications, which include configuration, deployment, and testing features.

Composite Editor

Component Editors

Build and
Deploy Wizards

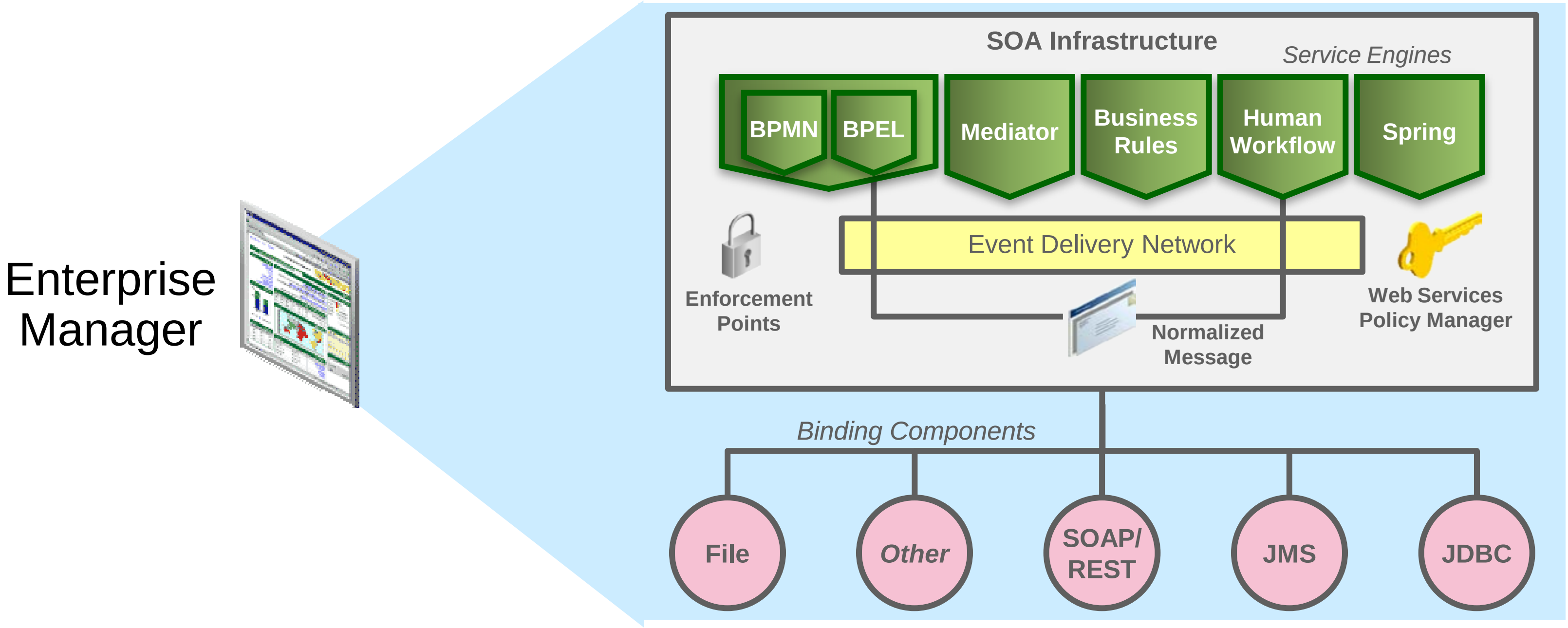
Test Tool Example

Other

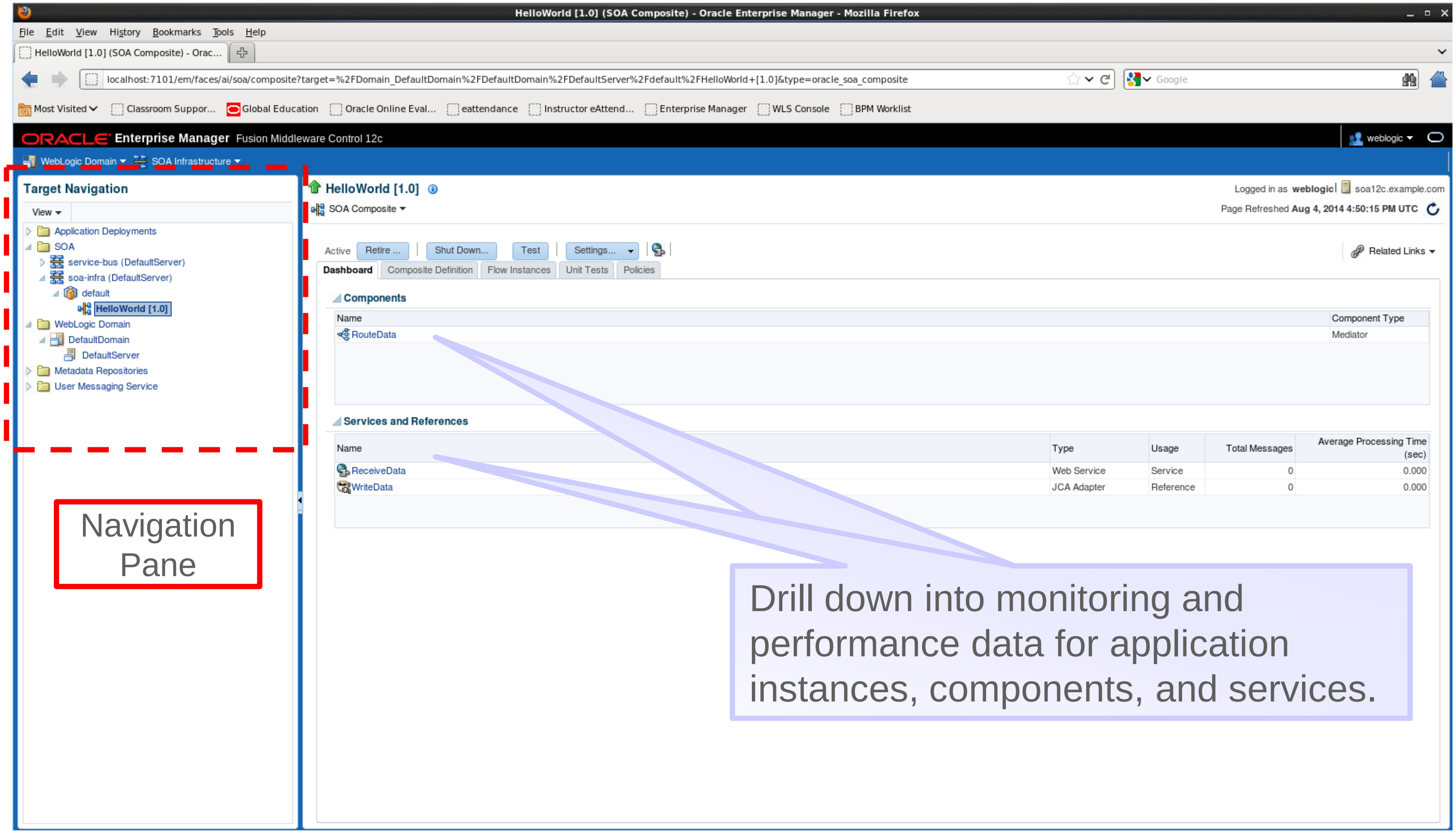
Enterprise Manager

Enterprise Manager provides tools to:

- Deploy and undeploy applications
- Manage the state of applications
- Test applications



Oracle Enterprise Manager: The Application Dashboard



HelloWorld [1.0] (SOA Composite) - Oracle Enterprise Manager - Mozilla Firefox

localhost:7101/em/faces/ai/soa/composite?target=%2FDomain_DefaultDomain%2FDefaultDomain%2FDefaultServer%2Fdefault%2FHelloWorld+[1.0]&type=oracle_soa_composite

Target Navigation

- Application Deployments
- SOA
  - service-bus (DefaultServer)
  - soa-infra (DefaultServer)
    - default
      - HelloWorld [1.0]
- WebLogic Domain
  - DefaultDomain
    - DefaultServer
- Metadata Repositories
- User Messaging Service

HelloWorld [1.0]

SOA Composite

Logged in as weblogic | soa12c.example.com

Page Refreshed Aug 4, 2014 4:50:15 PM UTC

Components

Name	Component Type
RouteData	Mediator

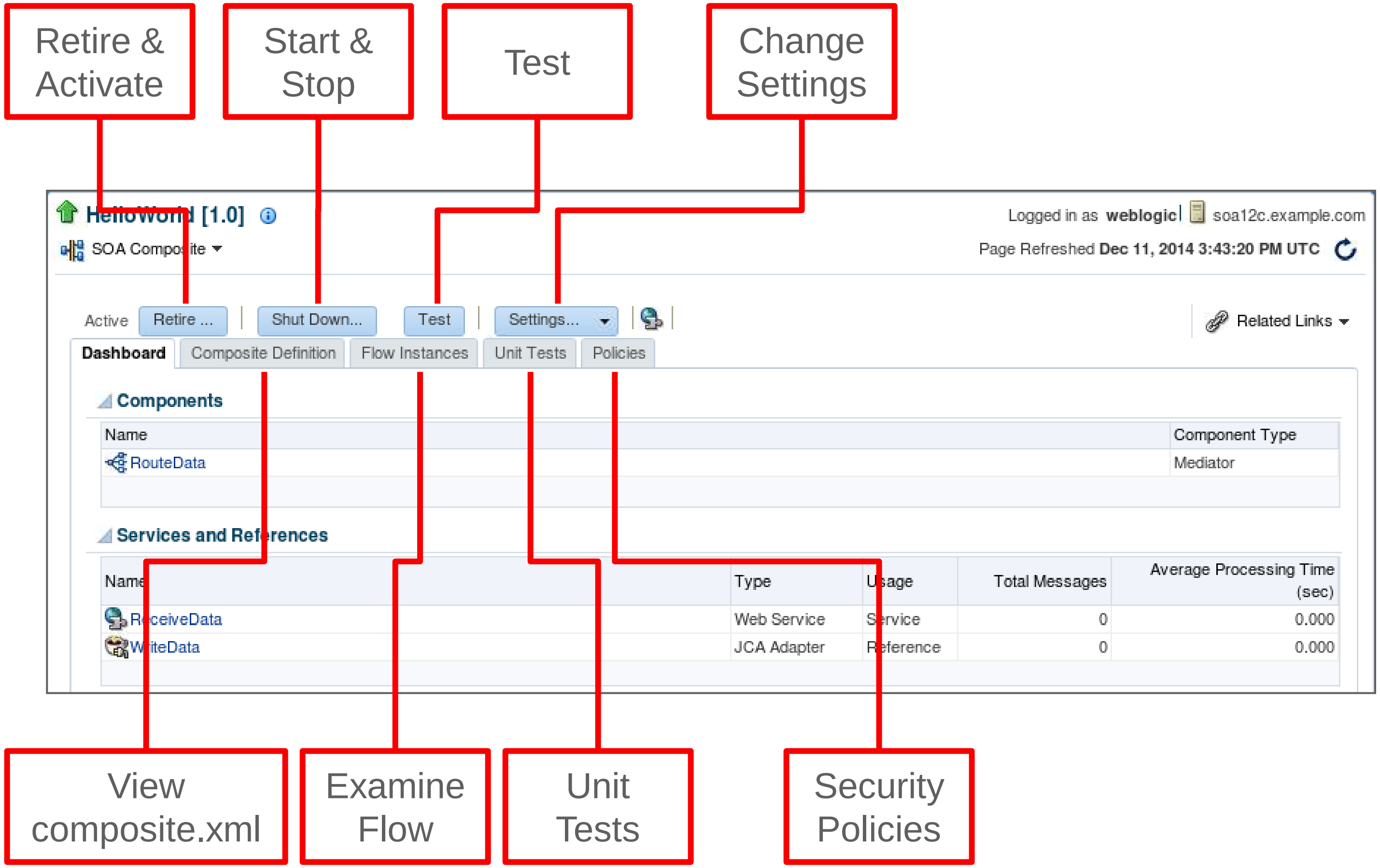
Services and References

Name	Type	Usage	Total Messages	Average Processing Time (sec)
ReceiveData	Web Service	Service	0	0.000
WriteData	JCA Adapter	Reference	0	0.000

Navigation Pane

Drill down into monitoring and performance data for application instances, components, and services.

Composite Application Management Tools



Retire & Activate

Start & Stop

Test

Change Settings

HelloWorld [1.0]

SOA Composite

Logged in as **weblogic** | soa12c.example.com

Page Refreshed **Dec 11, 2014 3:43:20 PM UTC**

Active Retire ... | Shut Down... | Test | Settings... | Related Links

Dashboard | Composite Definition | Flow Instances | Unit Tests | Policies

Components

Name	Component Type
RouteData	Mediator

Services and References

Name	Type	Usage	Total Messages	Average Processing Time (sec)
ReceiveData	Web Service	Service	0	0.000
WriteData	JCA Adapter	Reference	0	0.000

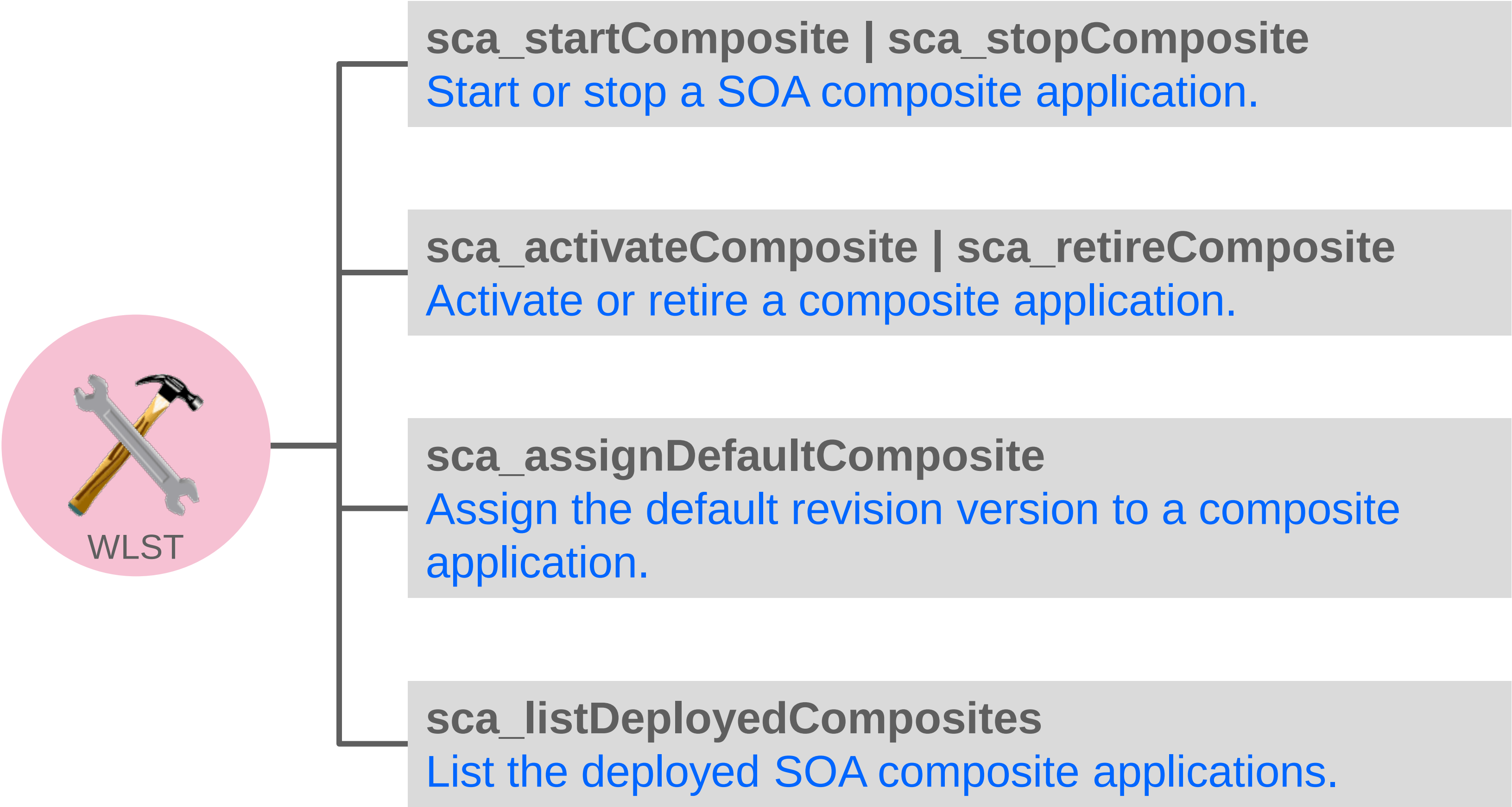
View composite.xml

Examine Flow

Unit Tests

Security Policies

Managing SOA Applications with the WLST Utility



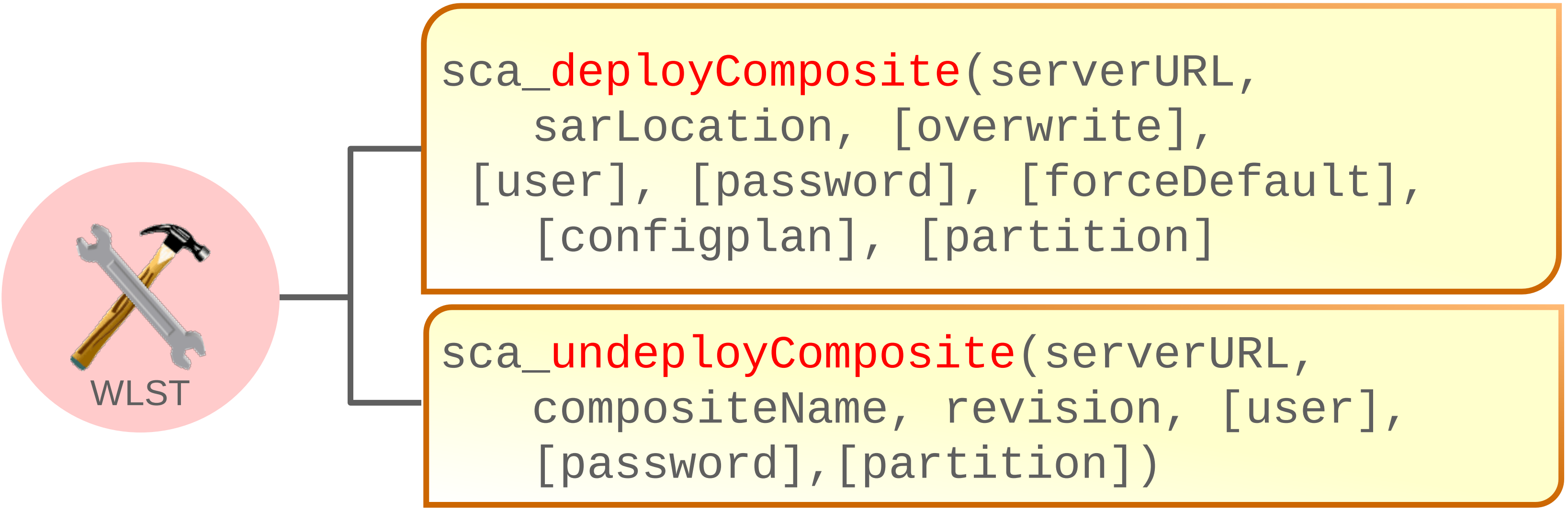
sca_startComposite | sca_stopComposite
Start or stop a SOA composite application.

sca_activateComposite | sca_retireComposite
Activate or retire a composite application.

sca_assignDefaultComposite
Assign the default revision version to a composite application.

sca_listDeployedComposites
List the deployed SOA composite applications.

Deploying and Undeploying Applications with WLST



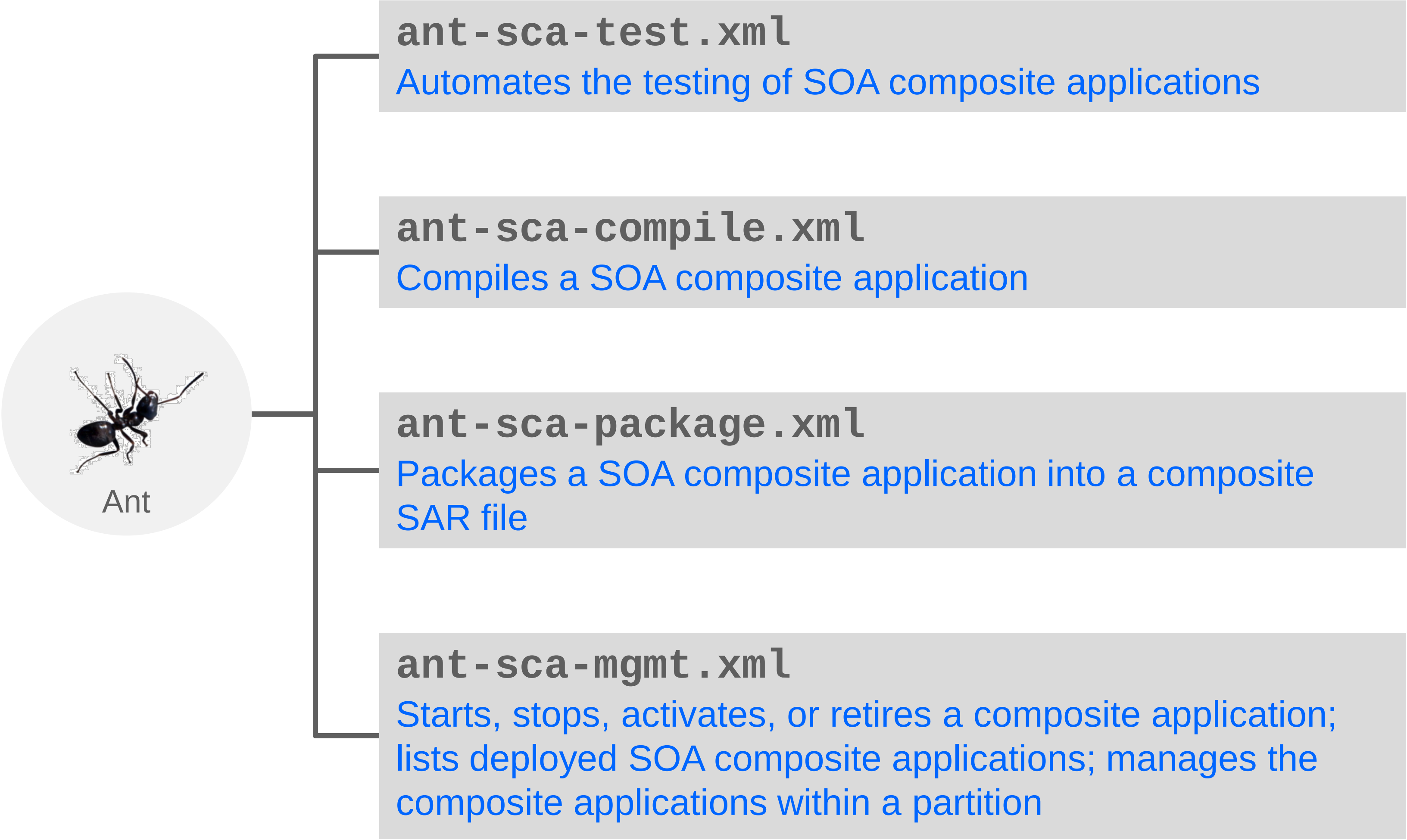
```
sca_deployComposite(serverURL,
   sarLocation, [overwrite],
 [user], [password], [forceDefault],
   [configplan], [partition]
```

```
sca_undeployComposite(serverURL,
   compositeName, revision, [user],
   [password],[partition])
```

deployment example

```
sca_deployComposite("http://myhost:7001",
   "/tmp/sca_HelloWorld_rev1.0.jar",
   partition="myPartition")
```

Managing SOA Applications with Ant Scripts



Ant

`ant-sca-test.xml`

Automates the testing of SOA composite applications

`ant-sca-compile.xml`

Compiles a SOA composite application

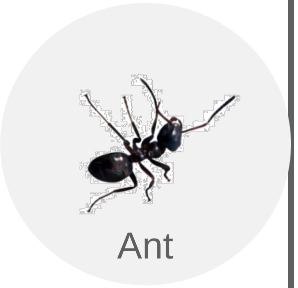
`ant-sca-package.xml`

Packages a SOA composite application into a composite SAR file

`ant-sca-mgmt.xml`

Starts, stops, activates, or retires a composite application; lists deployed SOA composite applications; manages the composite applications within a partition

Deploying and Undeploying Applications with Ant



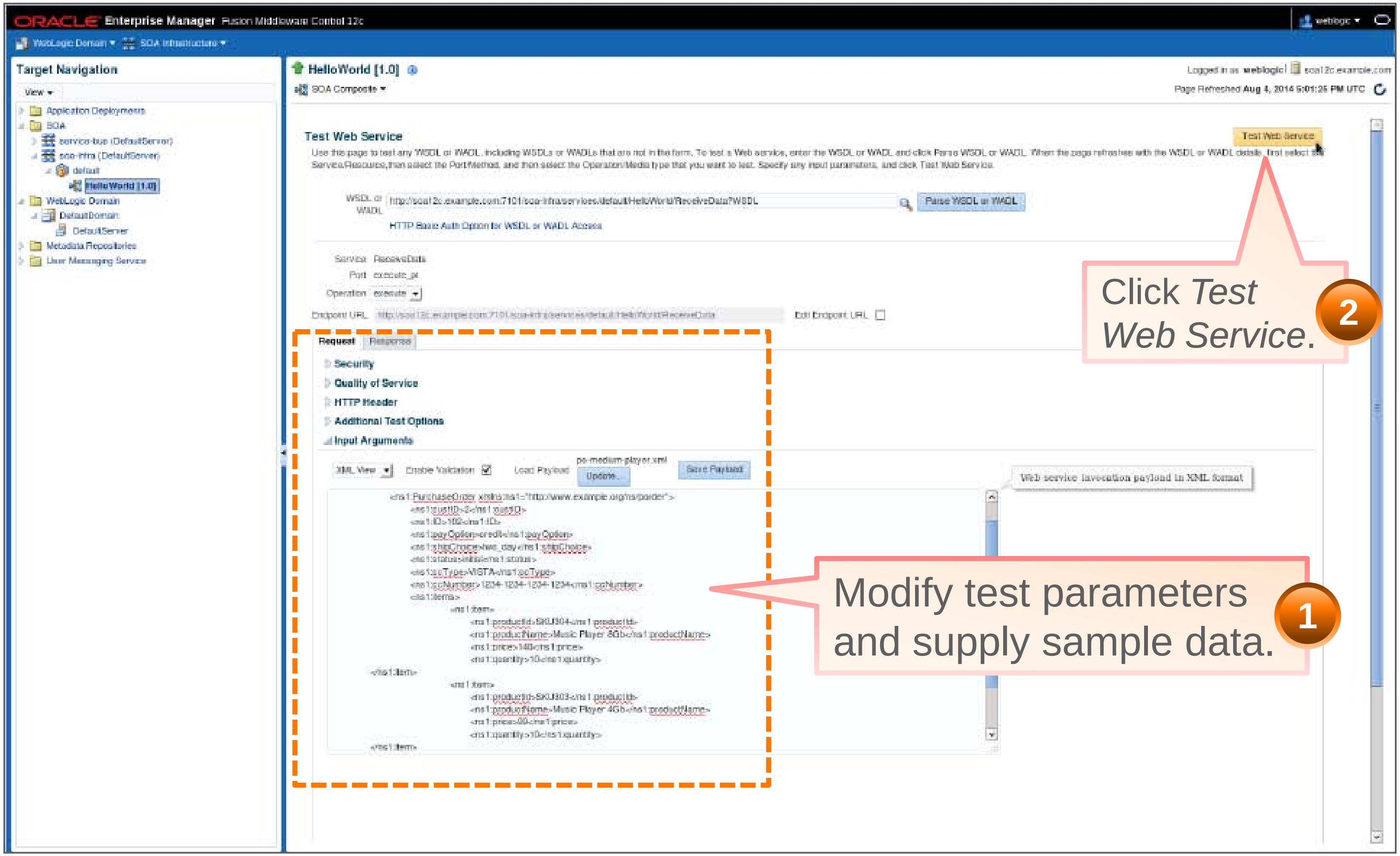
```
ant –f ant-sca-deploy.xml
-DserverURL=server.url
-DsarLocation=sar.location
-Doverwrite=true/false
-Duser=user -Dpassword=password
```

```
ant –f ant-sca-deploy.xml undeploy
-DserverURL=server.url
-DcompositeName=composite.name
-Drevision=revision
-Dpassword=password
```

Understanding Composite Application Instances

- A new composite instance is created when a SOA composite application is initialized.
- Each instance is identified through its unique instance ID.
- The state of composite application instances can be monitored in Oracle Enterprise Manager Fusion Middleware Control.

Testing a Composite Application



Test Web Service

Use this page to test any WSDL or WADL, including WSDLs or WADLs that are not in the farm. To test a Web service, enter the WSDL or WADL and click Parse WSDL or WADL. When the page refreshes with the WSDL or WADL details, first select the Service/Resource, then select the Port/Method, and then select the Operation/Media type that you want to test. Specify any input parameters, and click Test Web Service.

WSDL or WADL: http://soa12c.example.com:7101/soa-infra/services/default/HelloWorld/ReceiveData?WSDL

HTTP Basic Auth Option for WSDL or WADL Access

Service: ReceiveData

Port: execute_pt

Operation: execute

Endpoint URL: http://soa12c.example.com:7101/soa-infra/services/default/HelloWorld/ReceiveData

Edit Endpoint URL ☐

Enable Validation ☑

Load Payload: po-medium-player.xml

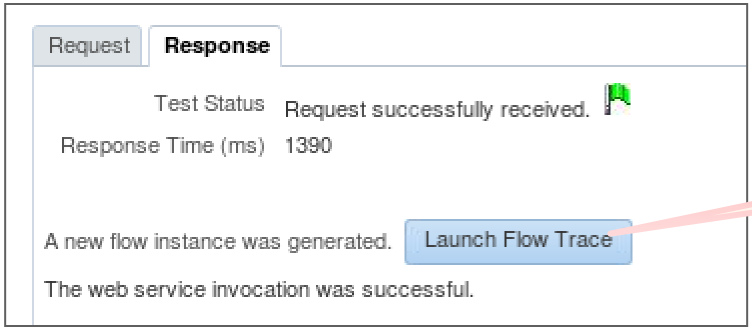
```
        <ns1:PurchaseOrder xmlns:ns1="http://www.example.org/ns/porder">
            <ns1:custID>2</ns1:custID>
            <ns1:ID>102</ns1:ID>
            <ns1:payOption>credit</ns1:payOption>
            <ns1:shipChoice>two_day</ns1:shipChoice>
            <ns1:status>initial</ns1:status>
            <ns1:ccType>VISTA</ns1:ccType>
            <ns1:ccNumber>1234-1234-1234-1234</ns1:ccNumber>
            <ns1:items>
                   <ns1:item>
                       <ns1:productId>SKU304</ns1:productId>
                       <ns1:productName>Music Player 8Gb</ns1:productName>
                       <ns1:price>140</ns1:price>
                       <ns1:quantity>10</ns1:quantity>
    </ns1:item>
                   <ns1:item>
                       <ns1:productId>SKU303</ns1:productId>
                       <ns1:productName>Music Player 4Gb</ns1:productName>
                       <ns1:price>90</ns1:price>
                       <ns1:quantity>10</ns1:quantity>
    </ns1:item>
```

Click *Test Web Service*. 2

Modify test parameters and supply sample data. 1

The Response Tab

- The web page is refreshed with the Response tab displaying either of the following:
  - The synchronous response data or fault message
  - The fact that no response was possible because it was an asynchronous request (regardless of whether it was a one-way or two-way operation), because the web client cannot receive a response as an asynchronous callback message



Test Status Request successfully received.

Response Time (ms) 1390

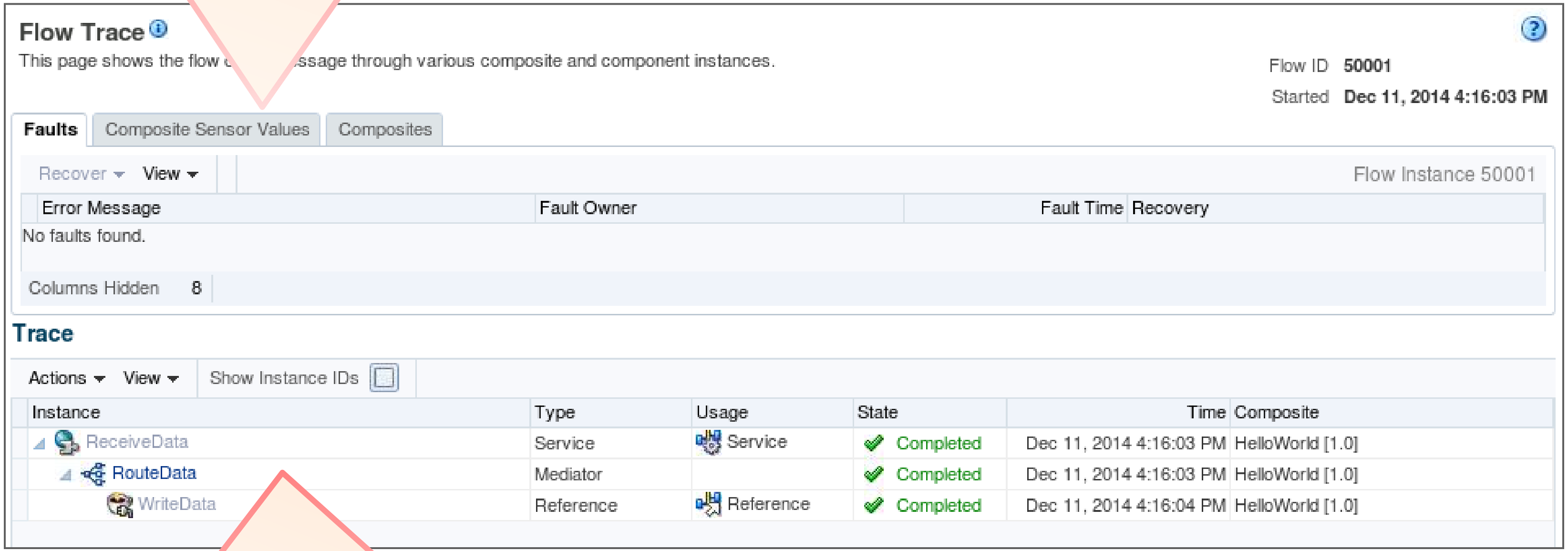
A new flow instance was generated.

The web service invocation was successful.

Use the Flow Trace feature to examine the execution of the test in more detail.

Flow Trace

Fault and sensor information is available from the Flow Trace pane.



Flow Trace

This page shows the flow c ssage through various composite and component instances.

Flow ID **50001**
Started **Dec 11, 2014 4:16:03 PM**

Flow Instance 50001

Error Message	Fault Owner	Fault Time	Recovery
No faults found.			

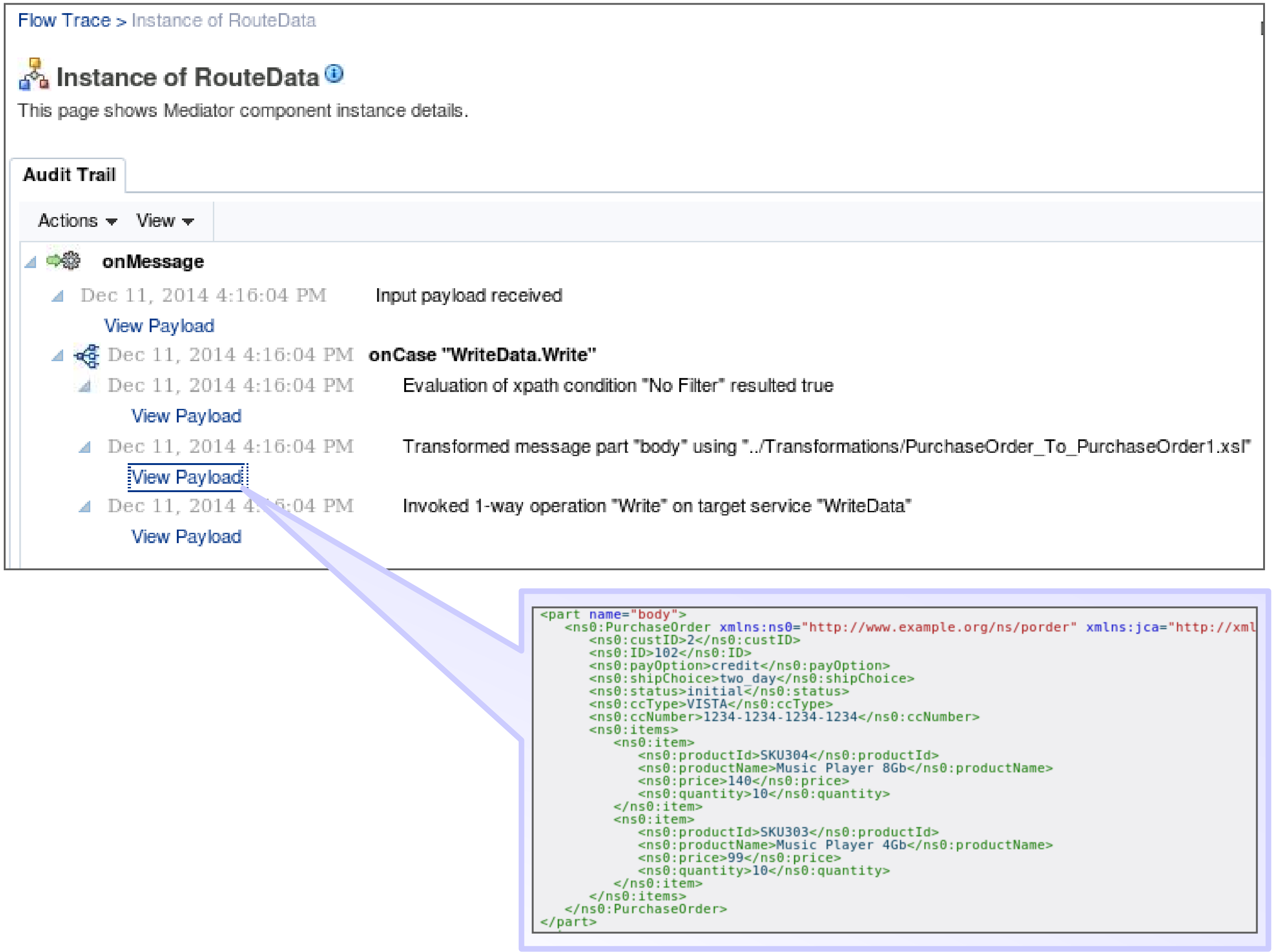
Columns Hidden 8

Trace

Instance	Type	Usage	State	Time	Composite
ReceiveData	Service	Service	✓ Completed	Dec 11, 2014 4:16:03 PM	HelloWorld [1.0]
RouteData	Mediator		✓ Completed	Dec 11, 2014 4:16:03 PM	HelloWorld [1.0]
WriteData	Reference	Reference	✓ Completed	Dec 11, 2014 4:16:04 PM	HelloWorld [1.0]

Execution of the application instance is logged on a per-component basis. Detailed audit information (*component audit trail*) is available by clicking a component name.

The Component Audit Trail



Flow Trace > Instance of RouteData

Instance of RouteData

This page shows Mediator component instance details.

onMessage

Dec 11, 2014 4:16:04 PM Input payload received

View Payload

Dec 11, 2014 4:16:04 PM **onCase "WriteData.Write"**

Dec 11, 2014 4:16:04 PM Evaluation of xpath condition "No Filter" resulted true

View Payload

Dec 11, 2014 4:16:04 PM Transformed message part "body" using "../Transformations/PurchaseOrder_To_PurchaseOrder1.xsl"

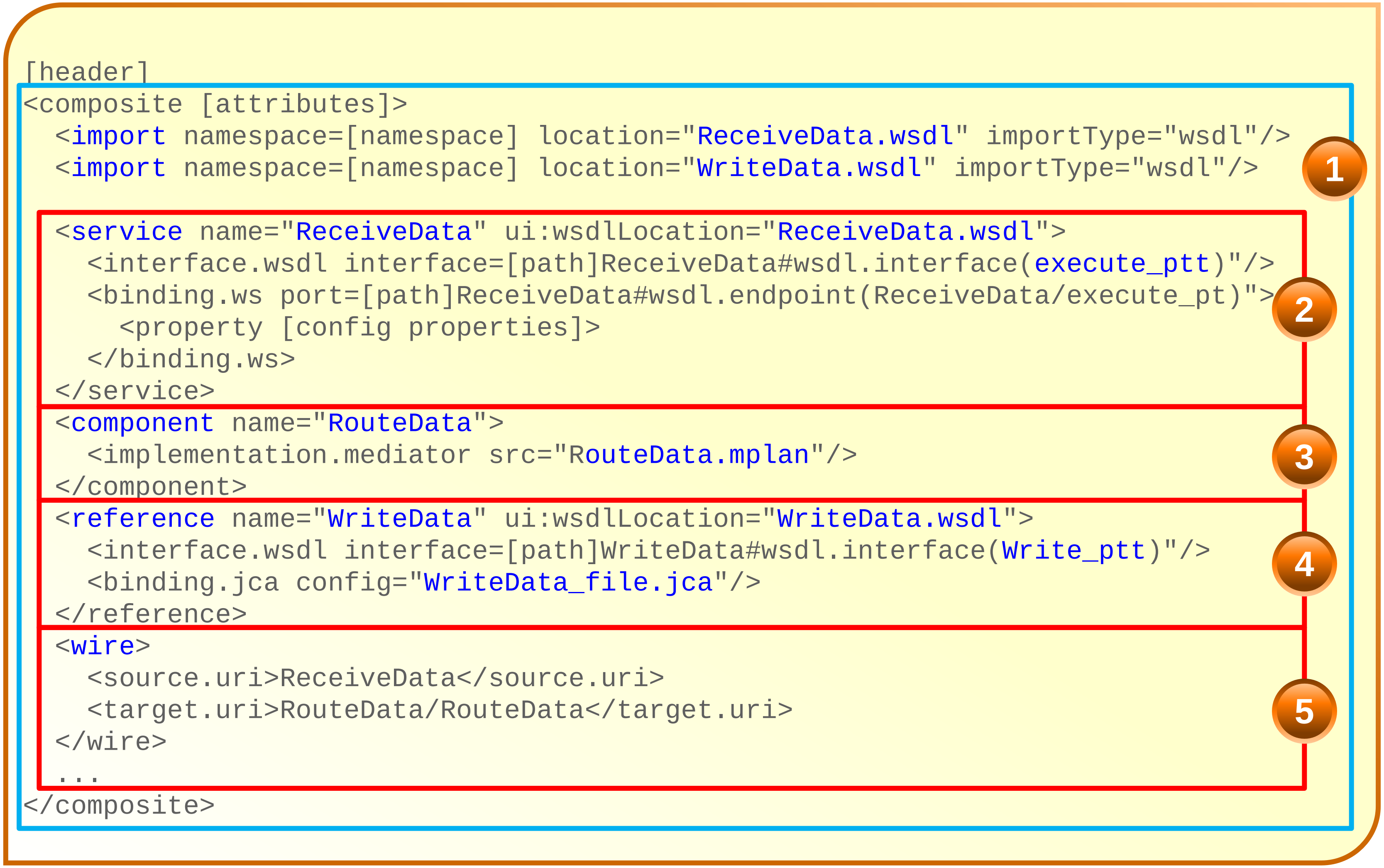
View Payload

Dec 11, 2014 4:16:04 PM Invoked 1-way operation "Write" on target service "WriteData"

View Payload

```
<part name="body">
   <ns0:PurchaseOrder xmlns:ns0="http://www.example.org/ns/porder" xmlns:jca="http://xml
      <ns0:custID>2</ns0:custID>
      <ns0:ID>102</ns0:ID>
      <ns0:payOption>credit</ns0:payOption>
      <ns0:shipChoice>two_day</ns0:shipChoice>
      <ns0:status>initial</ns0:status>
      <ns0:ccType>VISTA</ns0:ccType>
      <ns0:ccNumber>1234-1234-1234-1234</ns0:ccNumber>
      <ns0:items>
         <ns0:item>
            <ns0:productId>SKU304</ns0:productId>
            <ns0:productName>Music Player 8Gb</ns0:productName>
            <ns0:price>140</ns0:price>
            <ns0:quantity>10</ns0:quantity>
         </ns0:item>
         <ns0:item>
            <ns0:productId>SKU303</ns0:productId>
            <ns0:productName>Music Player 4Gb</ns0:productName>
            <ns0:price>99</ns0:price>
            <ns0:quantity>10</ns0:quantity>
         </ns0:item>
      </ns0:items>
   </ns0:PurchaseOrder>
</part>
```

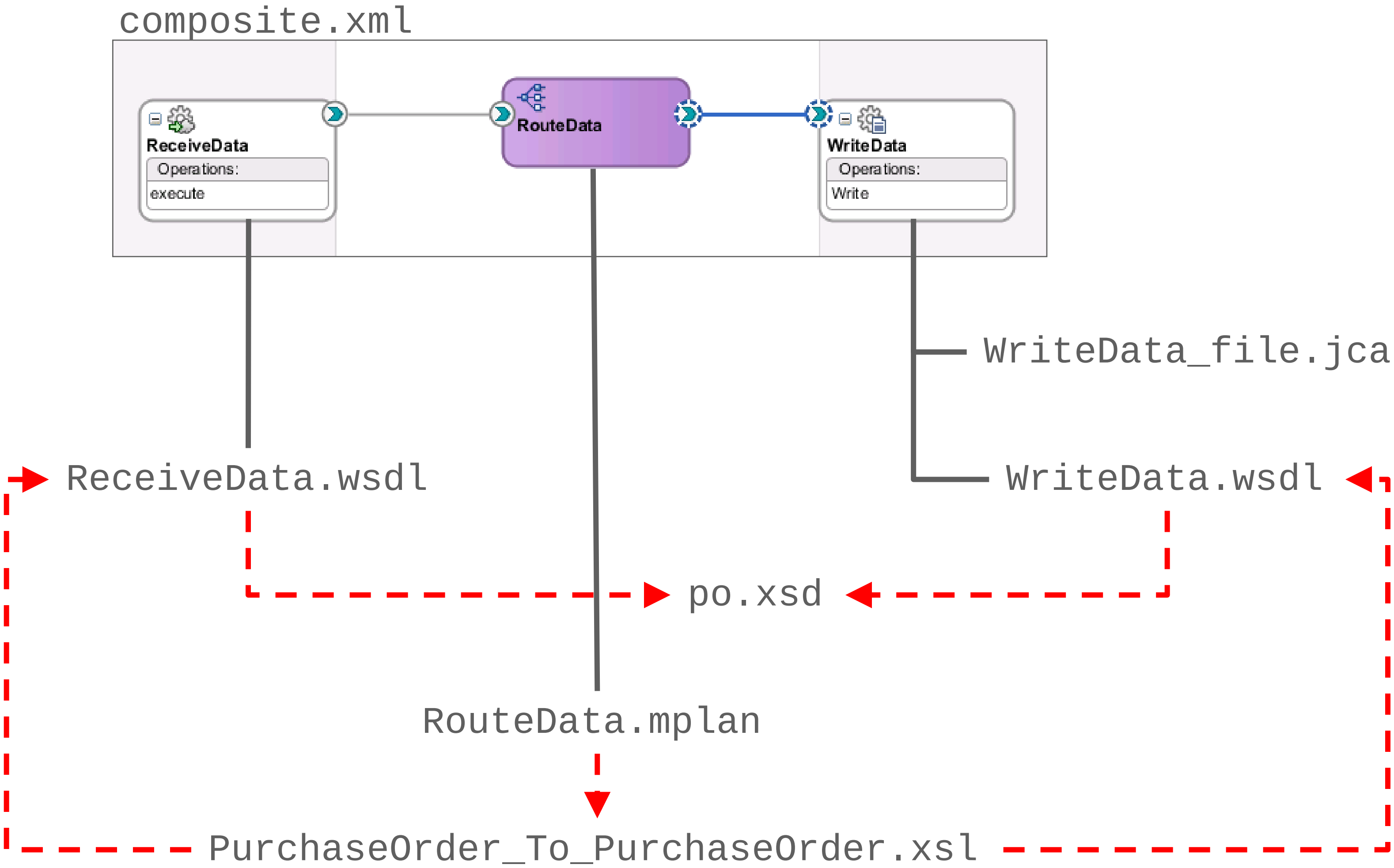
Examining the `composite.xml` Source



```
[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>
```

Project Files



composite.xml

WriteData_file.jca

ReceiveData.wsdl

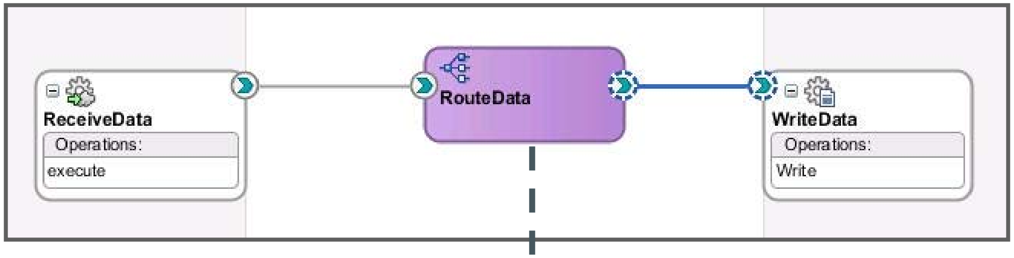
WriteData.wsdl

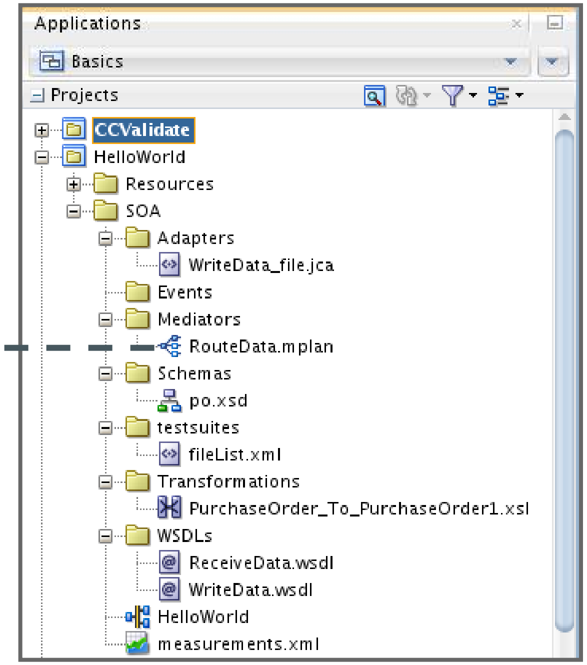
po.xsd

RouteData.mplan

PurchaseOrder_To_PurchaseOrder.xsl

Creating a Mediator




Applications

Basics

Projects

- CCValidate
- HelloWorld
  - Resources
  - SOA
    - Adapters
      - WriteData_file.jca
    - Events
    - Mediators
      - RouteData.mplan
    - Schemas
      - po.xsd
    - testsuites
      - fileList.xml
    - Transformations
      - PurchaseOrder_To_PurchaseOrder1.xsl
    - WSDLs
      - ReceiveData.wsdl
      - WriteData.wsdl
    - HelloWorld
    - measurements.xml

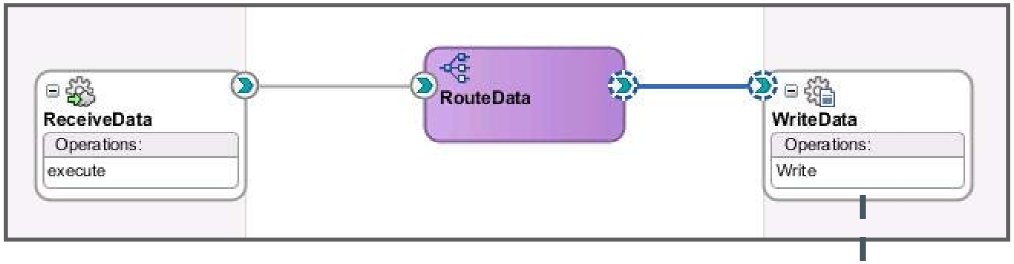
composite.xml

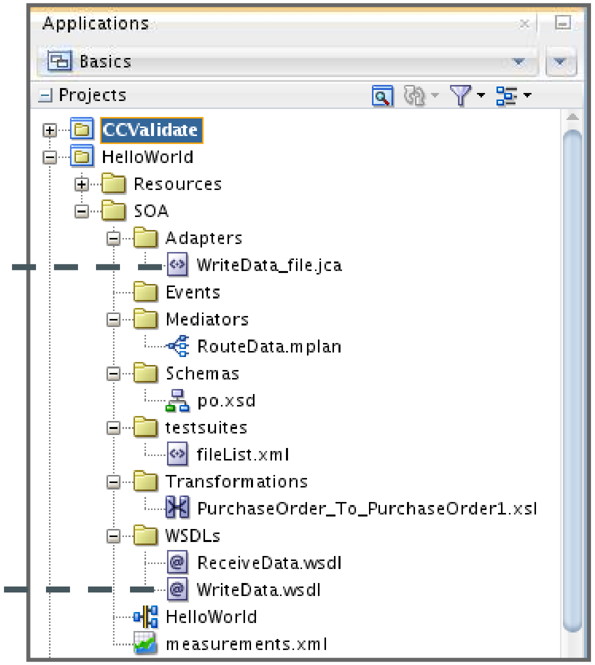
```
<component name="RouteData">

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

  </component>
```

Creating an Adapter





- CCValidate
- HelloWorld
  - Resources
  - SOA
    - Adapters
      - WriteData_file.jca
    - Events
    - Mediators
      - RouteData.mplan
    - Schemas
      - po.xsd
    - testsuites
      - fileList.xml
    - Transformations
      - PurchaseOrder_To_PurchaseOrder1.xsl
    - WSDLs
      - ReceiveData.wsdl
      - WriteData.wsdl
    - HelloWorld
    - measurements.xml

composite.xml

```
<composite [attributes]>
    . . .
  <reference name="WriteData" ui: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:

- Deploy a composite application by using Ant
- Test a composite application in Enterprise Manager
- Track message flow through a composite application
- Undeploy a composite application by using Ant
- Describe a composite application package and its dependencies

Practice 8: Overview

This practice covers the following topics:

- Practice 8-1: Deploying and Testing the HelloWorld Composite Application
- Practice 8-2: Testing the Application
- Practice 8-3: Undeploying an Application with an Ant Script
- Practice 8-4: Exploring the Composite Application Files