

1

Introduction to Service-Oriented Architecture

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Agenda

- **Course Overview**
- Introduction to Service-Oriented Architecture
- Oracle SOA Suite 12c Design Time: Overview
- Oracle SOA Suite 12c Run Time: Overview
- Installing Oracle SOA Suite 12c
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Target Audience

This course places strong emphasis on hands-on practices. It is written for a developer who has technical background, but does not have any experience with Oracle SOA Suite.

The course provides an overview of SOA features and functionality. The goal of the course is to provide you with an understanding of Oracle SOA Suite, and help you to arrive at a level of comfort by using it to create and configure composite applications.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This course is primarily targeted at application architects, business analysts, IT managers, and SOA project managers. In addition, technical personnel who are interested in using the SOA platform, SOA administrators, and developers will find this course useful for explaining basic concepts and introducing the Oracle SOA product suite. However, it does not teach how to develop production-level composite applications or how to administer SOA deployments.

Prerequisites

This course assumes that:

- You have completed the Oracle University course titled *XML Fundamentals* or have equivalent knowledge of XML. This helps in understanding principles related to XML documents, XPath, XML Schema, and XML Namespace.
- Your technical background includes an understanding of web service concepts such as Web Services Description Language (WSDL), SOAP, and Universal Description, Discovery and Integration (UDDI)

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Students who have some knowledge of the concepts listed in the slide will benefit from most of the course's content.

Course Objectives

After completing this course, you should be able to:

- Orchestrate business process flows by using BPEL
- Work with Mediator components with routing rules, filters, and transformations
- Implement a Business Rule component with IF/THEN rules and Decision Tables
- Test, debug, and troubleshoot a SOA composite application
- Initiate SOA composite applications by using the Event Delivery Network
- Apply security policies to service endpoints to secure interactions
- Implement Human Workflow with Human Task components
- Describe SOA concepts and related technology
- Create a SOA composite application by using JDeveloper

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Course Scope

SOA Suite Installation

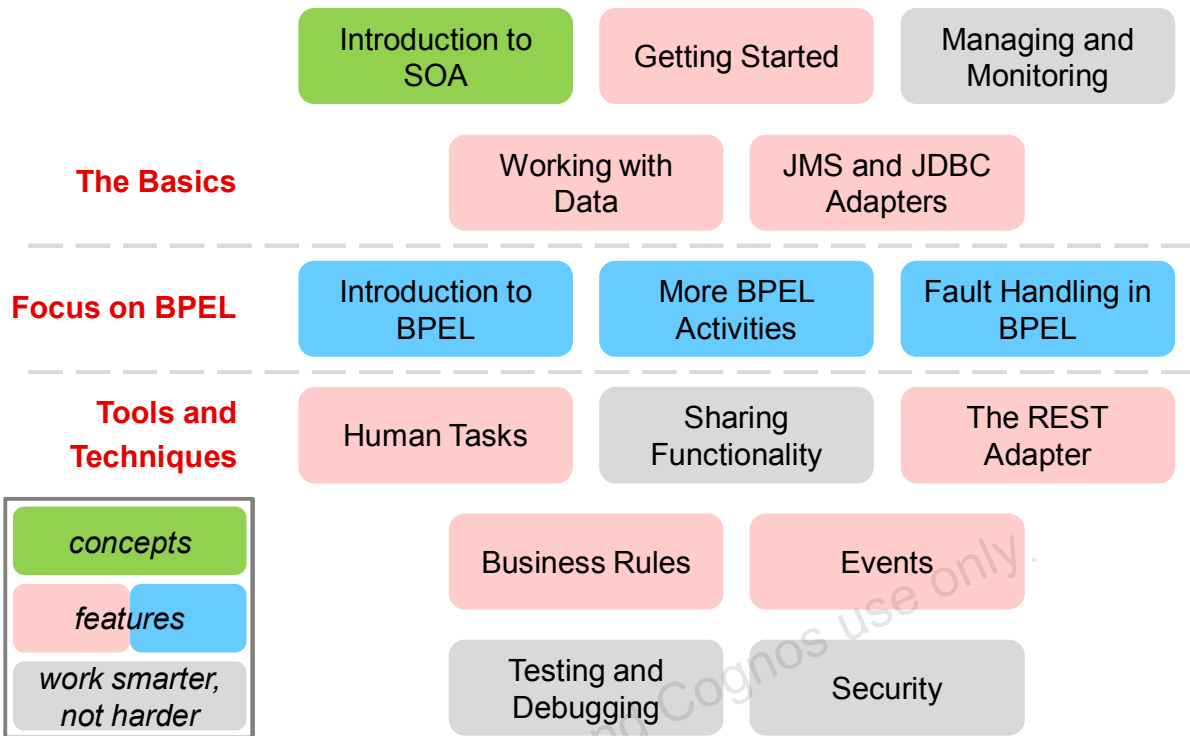
- Oracle JDeveloper
- Oracle Enterprise Manager
- Service Infrastructure
- Oracle Mediator
- Oracle Adapters
- Oracle Business Rules
- Oracle BPEL Process Manager
- Human Workflow
- Business Events and Events Delivery Network
- Oracle Metadata Repository
- Oracle WSM Policy Manager
- Oracle User Messaging Service
- Spring Context
- Oracle Business Activity Monitoring
- Oracle B2B
- Oracle Service Bus
- Oracle Enterprise Scheduler
- Managed File Transfer
- Oracle Event Processing

This course explores these topics.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Course Map



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Where Can I Learn More?

Topic	Website
Education and Training	http://education.oracle.com
Product Documentation	http://docs.oracle.com
Product Downloads	http://www.oracle.com/technology/software
Product Articles	http://www.oracle.com/technetwork/articles
Product Support	http://www.oracle.com/support
Product Forums	http://forums.oracle.com
Product Tutorials	http://www.oracle.com/oll
Sample Code	http://www.samplecode.oracle.com/

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe Service-Oriented Architecture (SOA)
- Explain services and key standards
- Describe Service Component Architecture (SCA)
- Describe the functionality of Oracle SOA Suite 12c service components and adapters
- Describe how WebLogic Application Server is used by Oracle SOA Suite

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

This introductory lesson is intended to present some important vocabulary and concepts related to Service-Oriented Architecture. It is also intended to present a high-level overview of the components and features of Oracle SOA Suite 12c, and to relate those components and features to SOA.

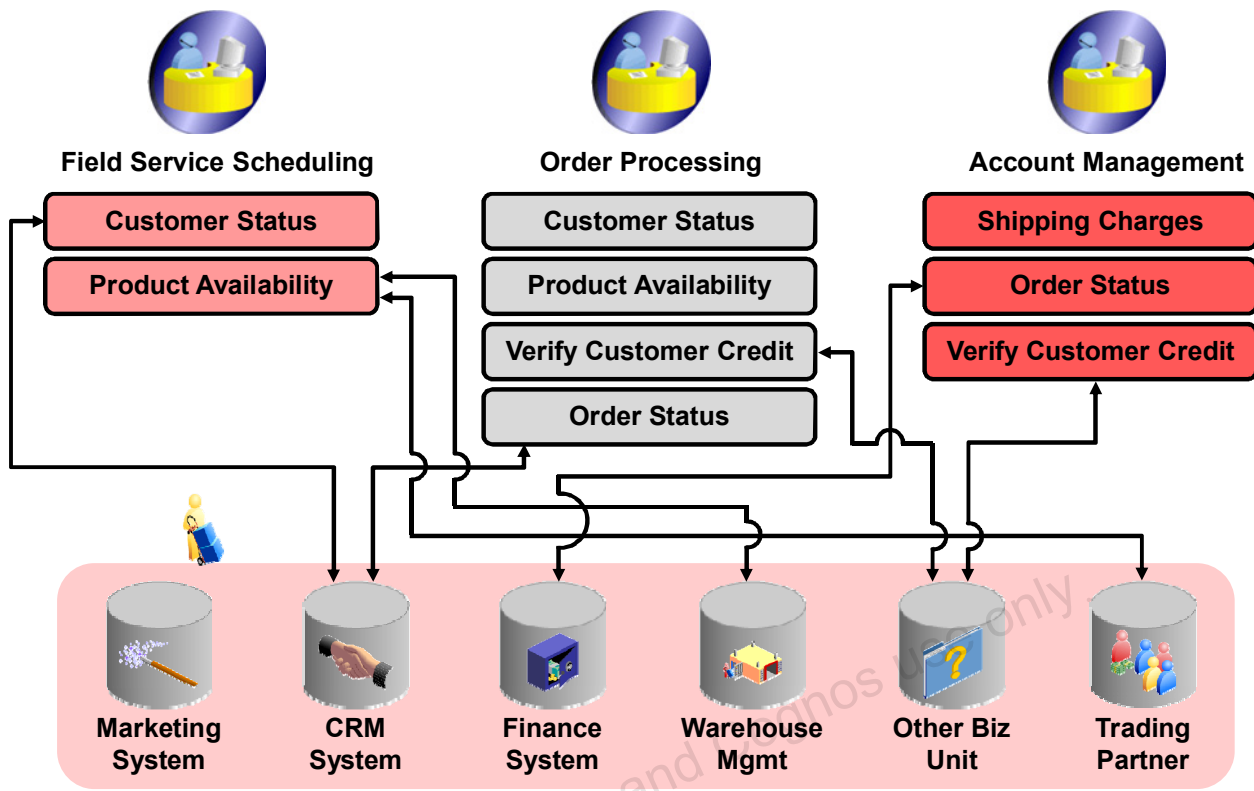
Agenda

- Course Overview
- Introduction to Service-Oriented Architecture
- Oracle SOA Suite 12c Design Time: Overview
- Oracle SOA Suite 12c Run Time: Overview
- Installing Oracle SOA Suite 12c
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Problem Statement



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Integration can be done in many ways, some of which are proprietary approaches that have enabled silo (stand-alone) systems to be integrated, typically in a point-to-point manner. However, the point-to-point approach tends to be brittle, and difficult to maintain over time. Functionality is often duplicated across applications. Changes to a single application interface have the potential to impact large numbers of systems, each of which may need modification—a time-consuming and expensive proposition.

Vendor-specific integration technologies such as Tuxedo have enabled systems to standardize the way integration is done within their own implementation, but because Tuxedo does not provide native support for web services, it keeps the technology confined to the organization that embraced it.

The computing industry worked to create a standardized protocol and software integration method called Common Object Request Broker Architecture (CORBA) that enabled systems to integrate by using a standard (non-XML-based) interface definition language for generating code templates that could be distributed. Although it is possible to apply a SOA approach with CORBA, its lack of widespread adoption and accessibility, in addition to its remote procedure call mechanisms, make it costly to implement and less friendly to use across an intranet or the Internet.

Definition of SOA

An IT strategy that organizes the **discrete functions** contained in enterprise applications into **interoperable, standards-based services** to be combined and **reused quickly** to meet business requirements

ORACLE

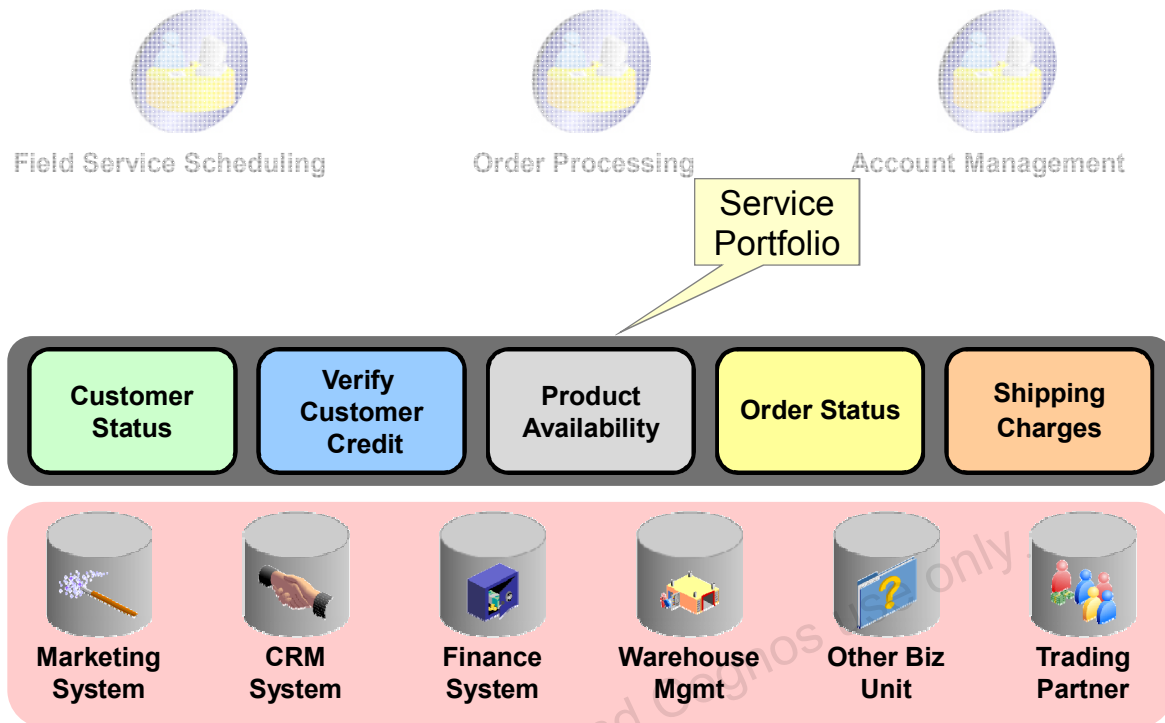
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Service-oriented architecture attempts to overcome many of the shortcomings of earlier integration technologies by embracing the concept of a *service*. A service is a discrete, decoupled unit of business functionality that is exposed and available to (ideally) all applications in the enterprise.

Services are the building blocks for a SOA-enabled application. They have functionality-described standard interface and message structure definitions. Multiple services are assembled and reused to more quickly create applications that can better support changing business processes.

The SOA approach to software systems enables a consumer of a service to be decoupled from the service provider. With SOA, services are accessed by using standard protocols (the glue), which enable interoperability from decoupled functions.

Building a Service Portfolio



ORACLE

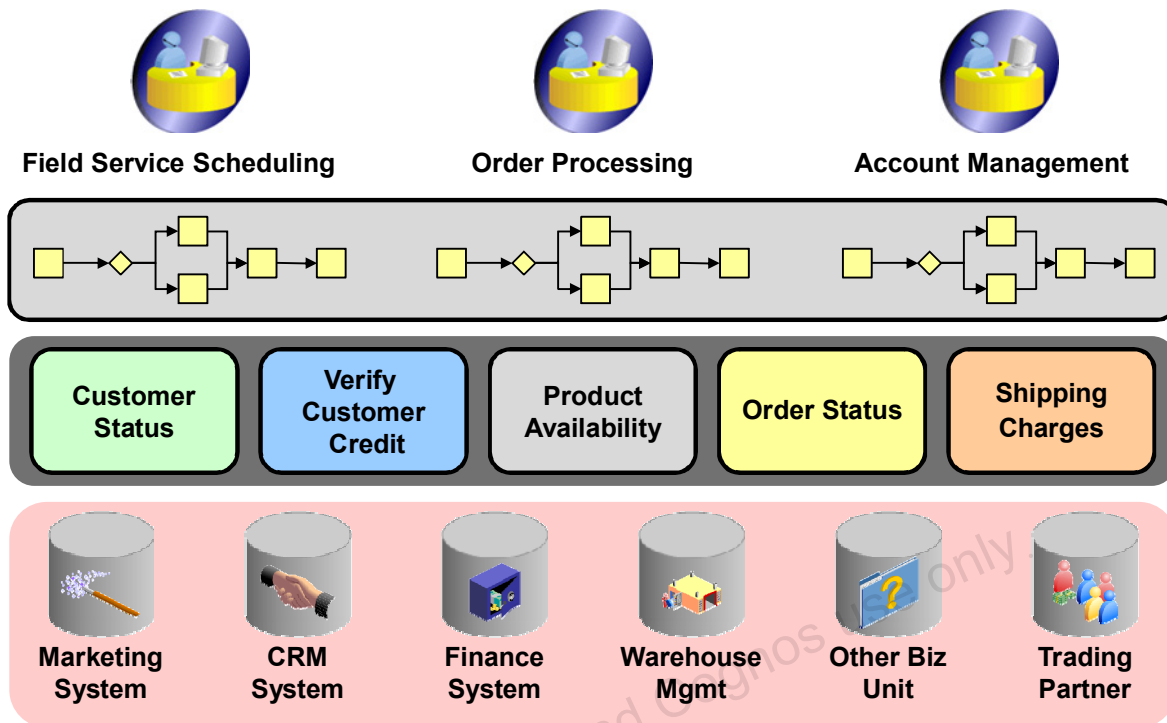
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When embarking on a SOA approach to systems integration, integrators begin by examining business process requirements to identify functionality to be used as a “service” unit to help complete a business process. A *service portfolio* is identified from a set of services that are required for implementing one or more business processes within a business domain.

A service portfolio can be realized by a combination of new and reused functionality. Existing functionality can be wrapped as services by using the adapter technology. New services can be created with SOA-enabled technologies, including web services, SCA composite applications, BPEL, and a service bus.

When a collection of services has been identified to serve a business process, it is recommended to store information about these services, their interfaces, and their message structures in a common location such as an enterprise repository. This facilitates sharing and reusability from design through production. Coupled with a service registry, an enterprise repository can migrate runtime service information into the development, test, and production operational environments. Coupling an enterprise repository and a service registry enables an organization to implement strong SOA governance strategies for services throughout a service’s life cycle, from the design-time environment to the runtime environment.

Assembling Services as Composite Applications



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

When a satisfactory service portfolio is created, the necessary services are assembled to form a *composite application* to address a specific business requirement. Services are reused as appropriate in multiple composite applications. As a result, development time and redundant functionality are dramatically reduced.

Service Component Architecture

Service Component Architecture (SCA) is a set of specifications that describe a model for building applications by using a Service-Oriented Architecture.

- Services are assembled to form a *composite application* that creates a solution that addresses a specific business requirement.
- Composite applications may contain new services (specifically for the application) and business functions from existing systems and applications (*reused* in the composite application).
- SCA provides a model for both the composition of services and the creation of service components, including reuse of existing application functions.

ORACLE

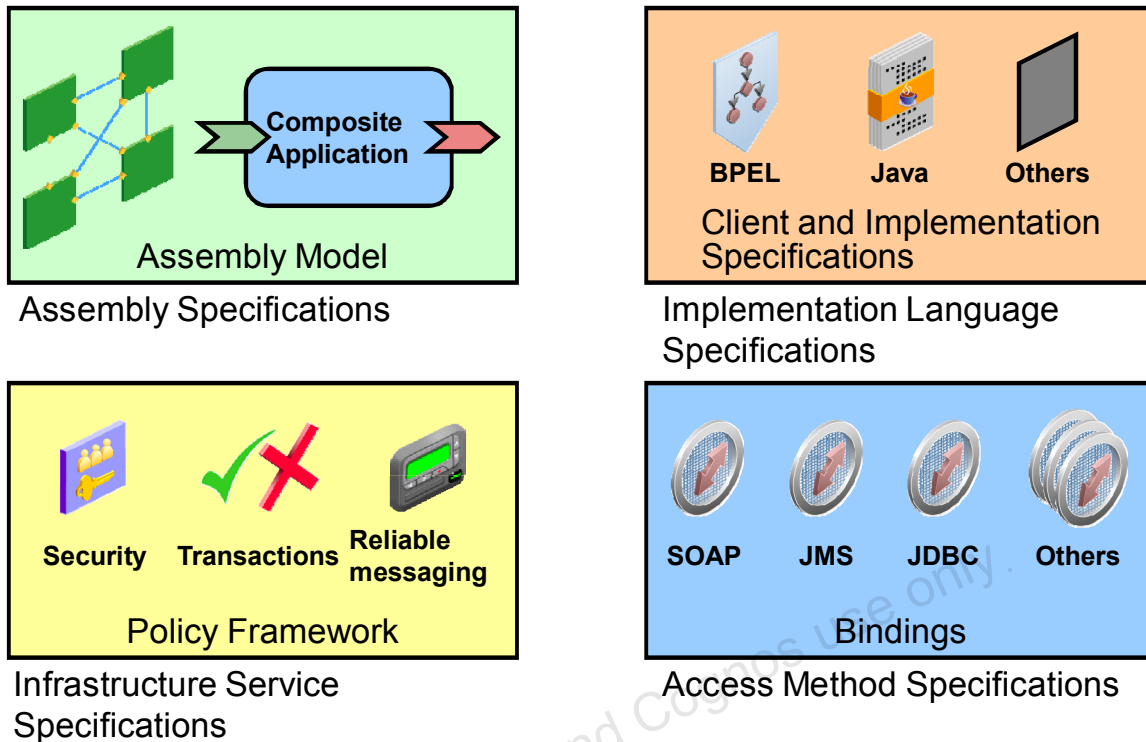
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The Service Component Architecture (SCA) specification provides a programming model for building applications and systems based on a SOA approach. SCA is based on the concept that a business function is provided as a series of services. These composite applications may contain new services (specifically for the application) and business functions from existing systems and applications (reused in the composite application). SCA provides a model for both the composition of services and the creation of service components, including reuse of existing application functions.

When assembled together into a composite application, they are managed, maintained, and deployed together. This streamlines and simplifies management of cooperating service components compared to earlier technologies that managed SOA applications as a set of individual services. Because its functionality can be exposed as a service through service entry points, the SOA composite application is itself a coarse-grained service.

For more information about the SCA initiative, visit: <http://www.oasis-open.org/scs>.

Elements of a Service Component Architecture



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

SCA is a collection of specifications.

Assembly model specifications define a standard set of XML elements that are used to define the structure and configuration of an SCA *composite* application.

Client & implementation (C&I) specifications define an implementation language, such as:

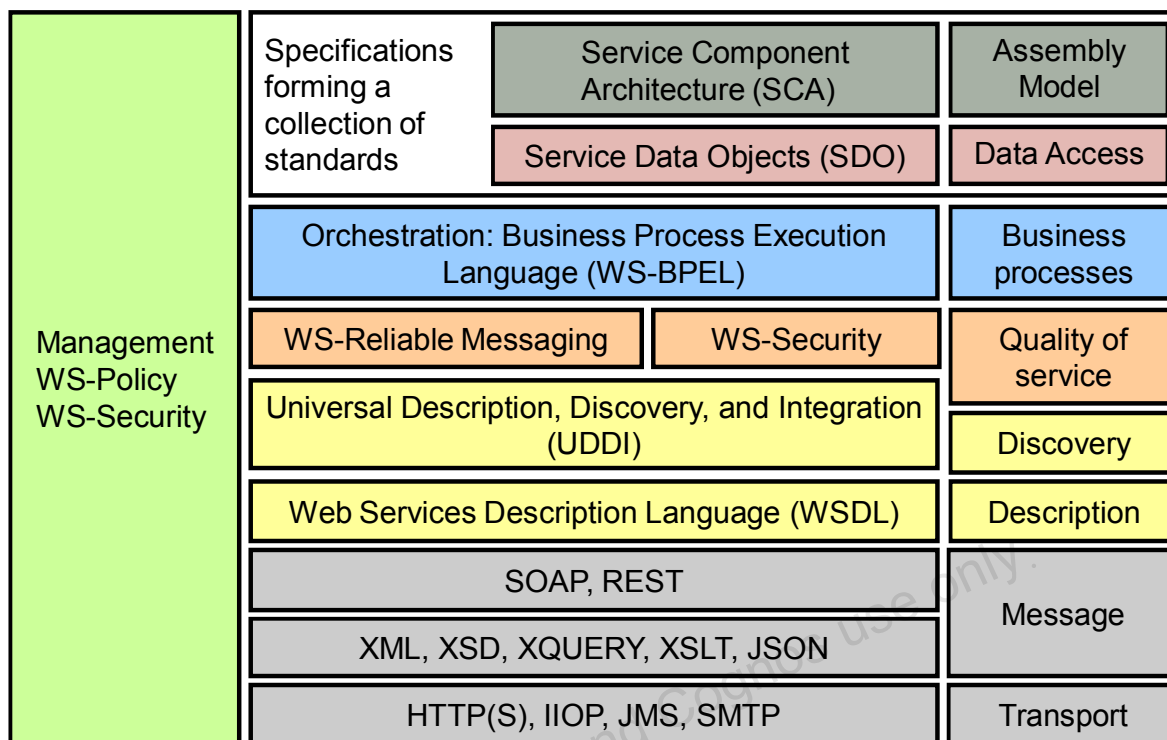
- **BPEL:** The WS-BPEL Client and Implementation (BPEL C&I) model specifies how WS-BPEL processes can be used as SCA components.
- **Java, C++, PHP:** This specifies language-specific APIs and annotations, which enable the creation of service components and service clients that are built in these languages.

Note: Oracle SOA Suite 12c implements BPEL and other components such as Mediator. At this time, it does not implement C++ or PHP. The spring framework can be used to integrate components that use Java interfaces into SOA composite applications.

Policy framework specifications capture and express nonfunctional requirements, such as security of a service definition. The policy framework impacts the life cycle of components and compositions. It supports specification of constraints, capabilities, and quality of service (QoS) from component design through concrete deployment.

Binding specifications define how services and references are accessed by using different transport types. There is a binding specification available for each different access method.

Why Standards Are Important in SOA



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Services are business functions that form the foundation for using a SOA approach to building applications. SOA applications aggregate functionality of related services to automate a business process. In order to reuse services (a fundamental goal of SOA), service functionality must be described by using standard interface and message structures.

SOA-based approaches have, naturally, embraced many of the message standards illustrated in the graphic in the slide, including XML, XSD, and SOAP. These document structures are easily exchanged by using standard Internet protocols such as HTTP. This results in easier interoperability across intranet and Internet networks. Discovery and description standards such as WSDL and UDDI contribute to the reusability of services and help achieve independence of hardware, operating systems, and implementation languages.

As their names suggest, the WS-Security and WS-Reliable Messaging standards describe standards for secure and reliable message delivery among service-based applications.

BPEL provides XML grammar for describing a business process as a series of activities.

The Service Component Architecture (SCA) and Service Data Objects (SDO) specifications are works in progress, guided by the OASIS organization. This work may lead to a new collection of standards with a SOA-enabled approach in mind.

Benefits of SOA

- **Reusability**
 - Existing functionality within an application is reused across business organizations and processes.
- **Interoperability**
 - Communication between services is not dependent on the platform. Services are loosely coupled to the application.
- **Scalability**
 - Applications are flexible to changing business requirements.
- **Cost efficiency**
 - Costs are reduced and delivery of new functionality is accelerated because existing resources are reused and integration of business resources is standards based.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Quiz

Service Component Architecture (SCA) is a competing standard to Service-Oriented Architecture (SOA).

- a. True
- b. False

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Answer: b

False. The Service Component Architecture (SCA) specification is complementary to SOA. It provides a programming model for building applications and systems based on a SOA approach.

Agenda

- Course Overview
- Introduction to Service-Oriented Architecture
- **Oracle SOA Suite 12c Design Time: Overview**
- Oracle SOA Suite 12c Run Time: Overview
- Installing Oracle SOA Suite 12c
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle SOA Suite 12c: Introduction

Oracle SOA Suite 12c provides a comprehensive suite of components for developing, securing, and monitoring SOA, including:

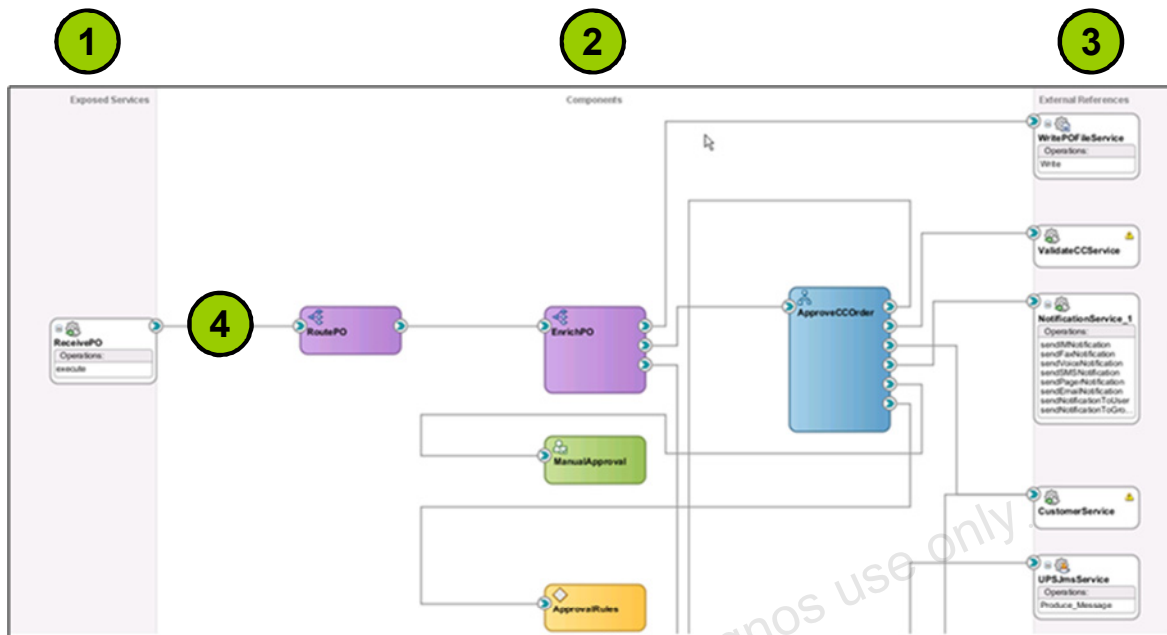
- **Service components**, which are the building blocks that are used to construct a SOA composite application
- **Service engines**, which implement service components at run time
- **Adapters**, which provide connectivity to functionality outside the composite application
- **Service infrastructure**, which provides the internal message transport infrastructure capabilities for connecting service components and enabling data flow

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle SOA Suite is a complete set of service infrastructure components that enable services to be created, managed, and orchestrated into SOA composites by using Business Process Execution Language (BPEL) and SCA. Together, these components provide task services, decision (rule-based) services, and adapter services. The unified architecture provides a comprehensive, standards-based platform for deploying and managing composite business applications with content and human workflow steps.

Elements of a SOA Composite Application



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

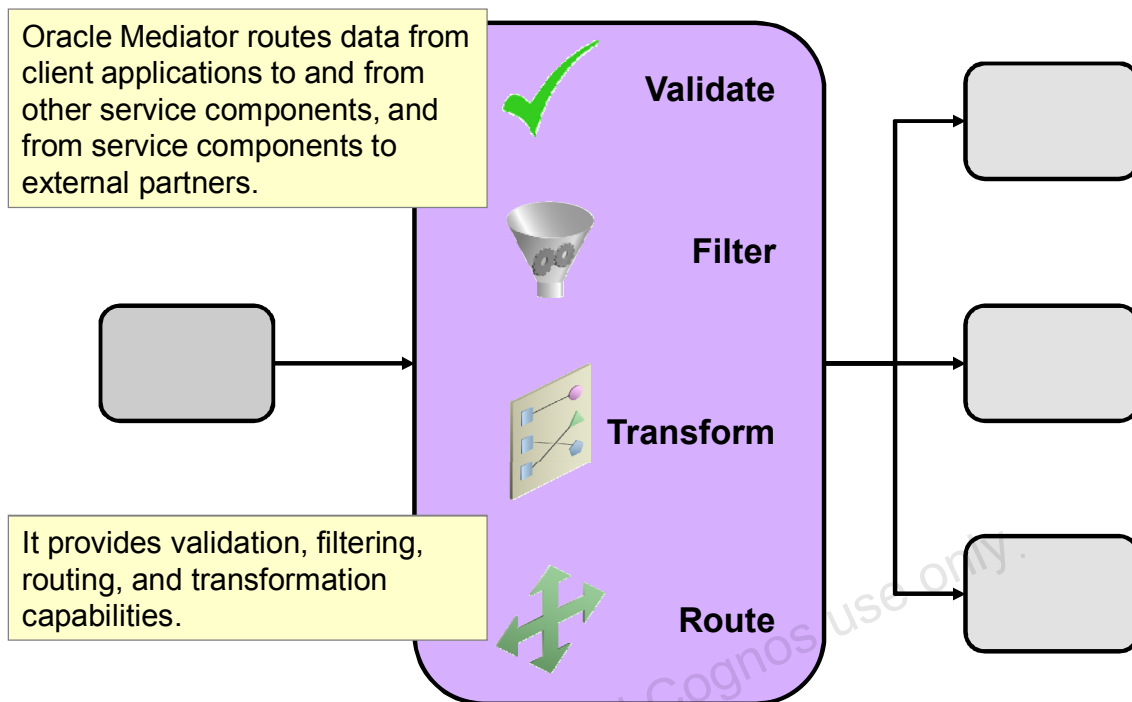
A SOA composite is an assembly of services, service components, and references that are designed and deployed together in a single application. Wiring between the service, service component, and reference enables message communication. The composite processes the information described in the messages. The graphic in the slide provides an example of a composite that includes an (inbound) exposed service, a variety of service components interacting, and several external references.

1. **Services** provide the composite client with an entry point to the SOA composite application. The service advertises its capabilities (also known as *operations*) to external applications with a *Web Services Description Language* (WSDL) file. The *binding* of the service describes the protocols that can communicate with the application. Examples include SOAP/HTTP or a JCA adapter.
2. **Service components** are the building blocks of a SOA composite application. Oracle SOA Suite 12c includes the following components:
 - *The BPEL Process component* enables design and execution of a business process that integrates a series of business activities and services into an end-to-end process flow.
 - *The Business Rules component* provides the means of making business decisions based on defined rules.

- *The Human Task component* allows you to model a workflow that describes tasks for users or groups to perform as part of an end-to-end business process flow.
 - *The Mediator component* is used for validation, transformation, and routing of message data between components.
3. *References* enable messages to be sent from the SOA composite application to services that are external to the composite.
 4. *Wires* enable you to graphically define the flow of messages within a single SOA composite application among services, service components, and references.

Note: The wire is not an actual call, but an ability to call. (For example, the routing rules in Mediator or the invoke/receive activities in BPEL are actual calls, but they can only go alongside previously defined wires.)

Oracle Mediator



ORACLE

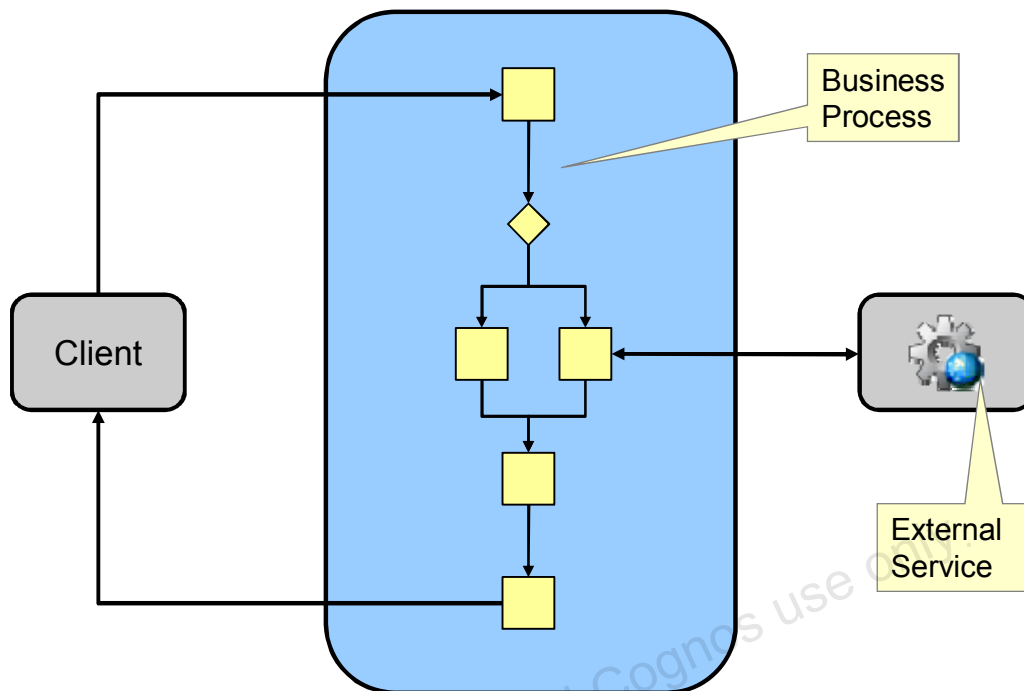
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Mediator routes data from client applications to and from other service components, and from service components to external partners. Using Oracle Mediator, you create routing services and rules for them. A routing service is the key component for moving a message through the composite application—from its entry point to its exit point. The rules determine how a message instance processed by the routing service gets to its next destination. Using the rules, Oracle Mediator can perform the following actions:

- **Route:** Determines the service component (BPEL process, business rule, human task, or Mediator) to which to send the messages
- **Validate:** Provides support for validating the incoming message payload by using a schematron or an XSD file
- **Filter:** If specified in the rules, applies a filter expression that specifies that the contents (payload) of a message be analyzed before any service is invoked
- **Transform:** If specified in the rules, transforms document data from one XML schema to another, thus enabling data interchange among applications by using different schemas

At run time, Oracle Mediator evaluates routing rules, performs transformations, applies optional time delays, and invokes another service. In addition, it can subscribe to and publish business events. A routing service can handle returned responses, callbacks, faults, and time-outs.

Oracle BPEL Process Manager



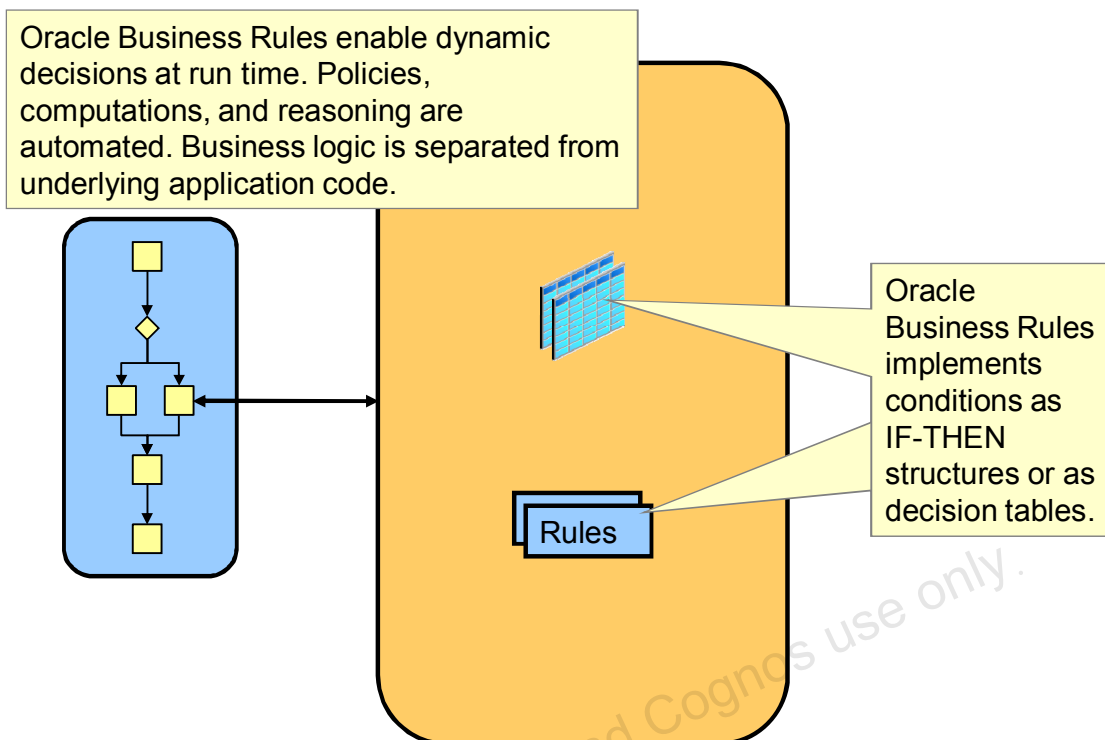
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle BPEL Process Manager provides the standard for assembling a set of discrete services into an end-to-end process flow, thus radically reducing the cost and complexity of process integration initiatives. Oracle BPEL Process Manager enables you to orchestrate synchronous and asynchronous services into end-to-end BPEL process flows. You integrate BPEL processes with external services (known as partner links). You also integrate technology adapters and services, such as human tasks, transformations, notifications, and business rules within the process.

Note: Business processes that are based on Business Process Modeling Notation (BPMN) can be built by using Oracle BPM (OBPM) Suite. The OBPM Suite is layered on top of the SOA Suite, and shares a unified run time.

Oracle Business Rules



ORACLE

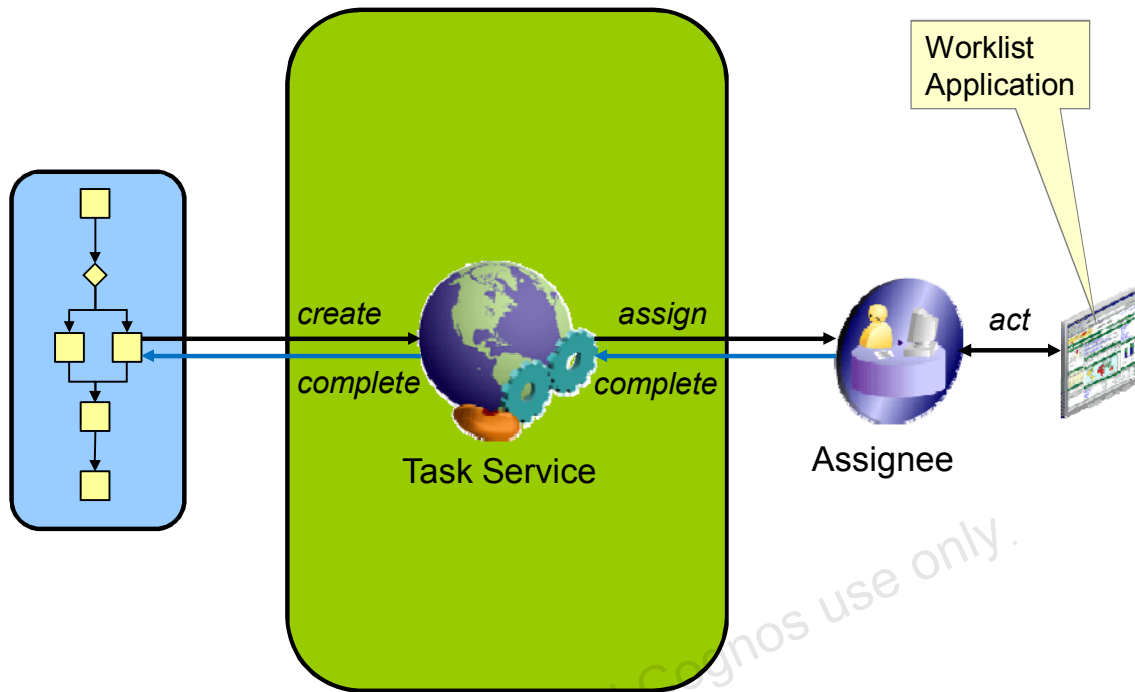
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Business Rules enable dynamic decisions at run time, allowing you to automate policies, constraints, computations, and reasoning while separating rule logic from underlying application code. Oracle Business Rules implements conditions as IF-THEN structures or as decision tables. These conditions can be edited by a business analyst to update policies and computational values with little or no assistance from a programmer.

Examples for using business rules include:

- **Dynamic processing:** Rules can determine intelligent routing paths within a business process based on service-level agreements or other guidelines.
- **Data validation and constraint checks:** Rules can validate input documents or apply additional constraints on requests.
- **Human task routing:** Rules can be used to perform policy-based task assignments to dispatch tasks to specific roles or users, or for balancing tasks among users to control the task assignment load.

Oracle Human Workflow



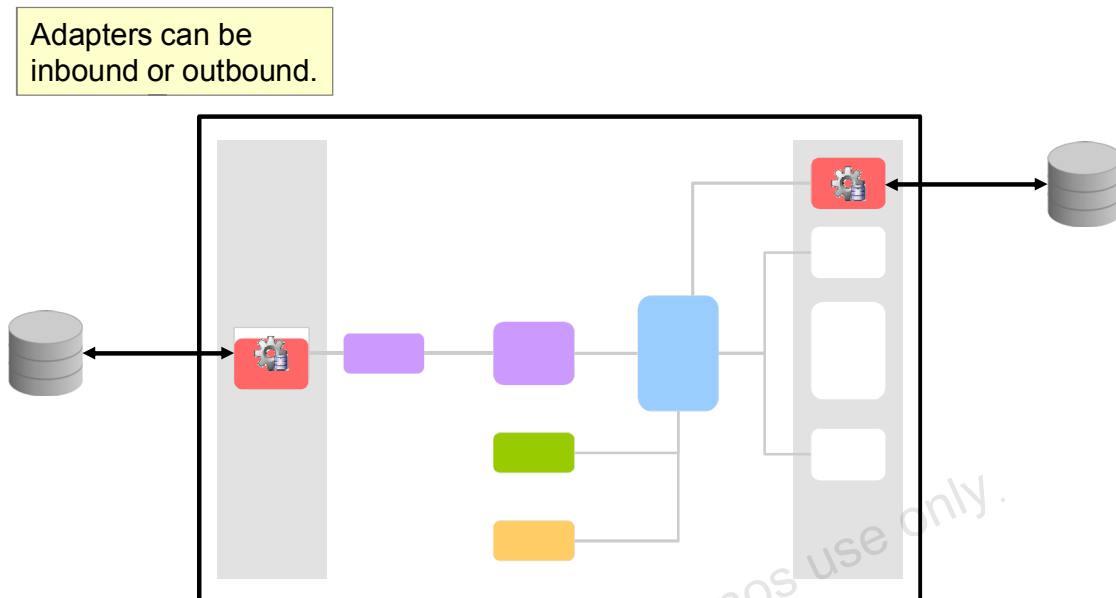
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Many end-to-end business processes require human interactions for approvals, exception management, or other activities that are required to advance the business process. The human workflow component provides the following features:

- Human interactions with processes, including assignment and routing of tasks to the correct users or groups
- Deadlines, escalations, notifications, and other features that are required for ensuring timely performance of a task (human task activity)
- Presentation of tasks to end users through a variety of mechanisms, including a worklist application
- Organization, filtering, prioritization, and other features that are required for end users to productively perform their tasks
- Reports, reassignments, load balancing, and other features that are required by supervisors and business owners to manage the performance of tasks

Oracle Adapters



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

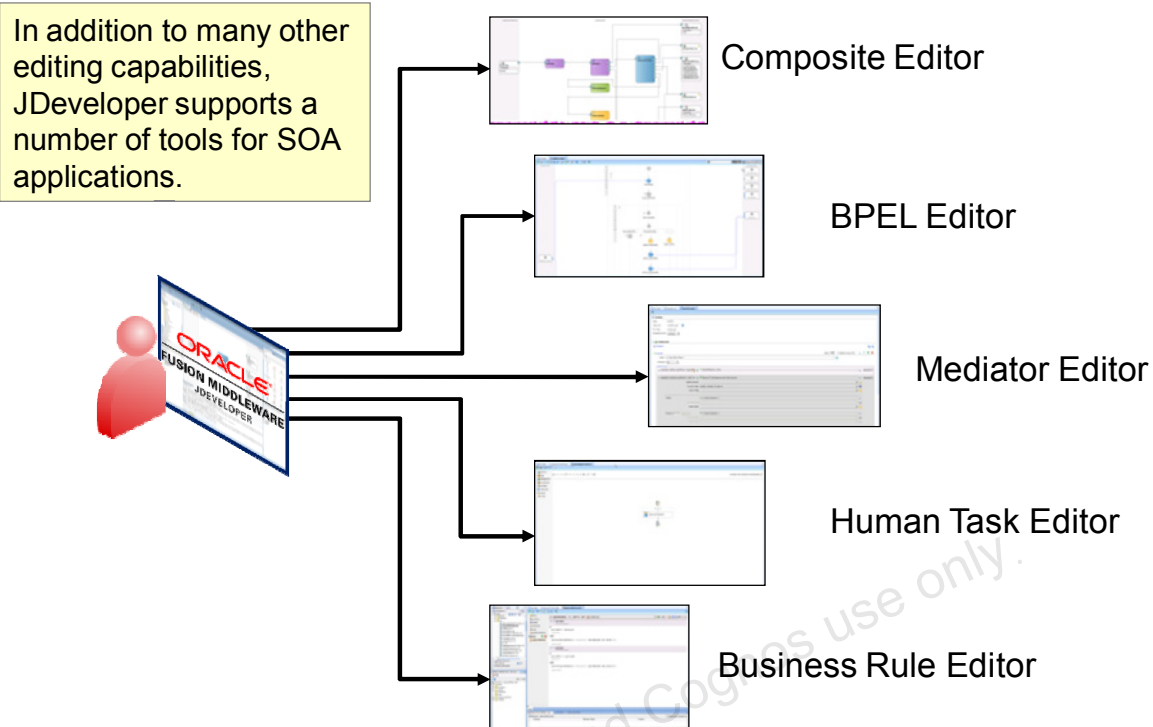
Adapters provide a service interface that exposes external application functionality in a form that can be used by the SOA composite application components. An adapter converts requests and responses into a form that is suitable for other (external) systems.

Oracle SOA Suite 12c is shipped with a variety of adapters that are commonly used to access functionality that is not normally available in a service-oriented context, such as:

- File and FTP adapters for reading, writing, and transferring files
- Database Adapter for interacting with relational database data
- JMS Adapter for communicating asynchronously with other applications through message-oriented middleware (MOM) services, including Oracle Advanced Queuing and IBM MQ

Because JCA adapters implement interfaces by using the Java Connector Architecture (JCA) API standards, custom adapters can be developed to expose existing functionality that is not usually available in a SOA application context. Adapters are an integration technology that extends the reach of the SOA composite application.

JDeveloper



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle JDeveloper is the development component of Oracle SOA Suite. It forms a comprehensive integrated service environment (ISE) for creating and deploying composite applications and managing the composite. Oracle JDeveloper enables developers to model, create, discover, assemble, orchestrate, test, deploy, and maintain composite applications based on services.

The SOA Composite Editor enables you to create, edit, and deploy services, and also to assemble them in a composite application, all from a single location. These components are integrated into one application, and they communicate with the outside world through binding components such as web services and JCA adapters.

The SOA Composite Editor enables you to use either of two approaches for designing SOA composite applications:

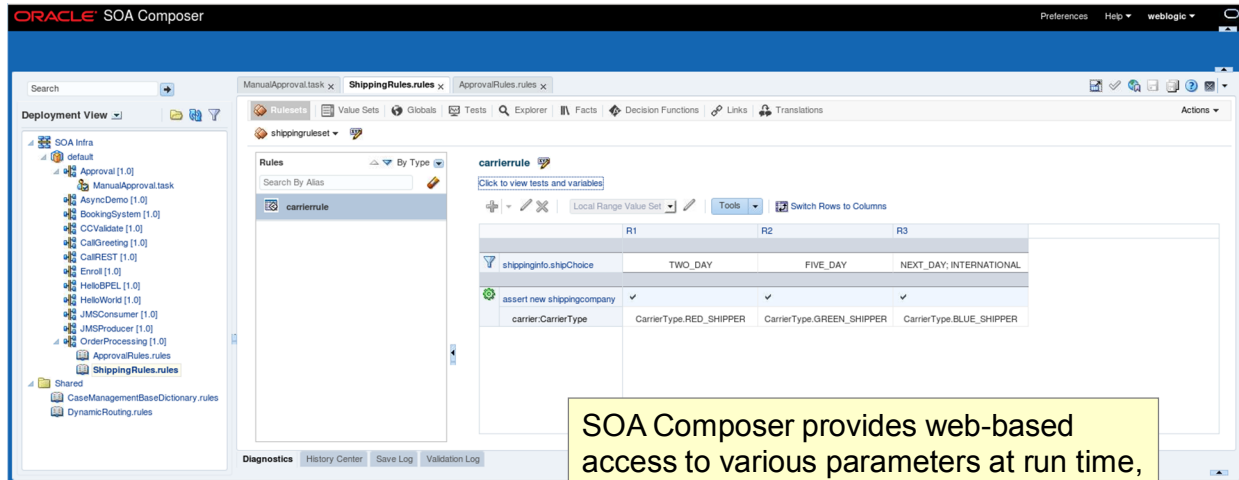
- The top-down approach of building a composite application puts interfaces first and implementation next. For example, you first add BPEL processes, human tasks, business rules, and the Oracle Mediator routing services components to an application, and later define the specific content of these service components.

- The bottom-up approach takes existing implementations of service components and wraps them with web service interfaces for assembly into a composite application. For example, you first create and define the specific content of BPEL processes, human tasks, business rules, and the Oracle Mediator routing services components, and later create a SOA composite application to which you add these service components.

Oracle JDeveloper provides the following additional editors to design service components:

- **Oracle BPEL Designer:** You use Oracle JDeveloper to create a BPEL process service component, and then you use BPEL Designer to design the component. BPEL Designer is displayed when you double-click a BPEL process in the SOA Composite Editor.
- **Oracle Mediator Editor:** You use Oracle JDeveloper to create a Mediator service component, and then you use Mediator Editor to design the component. Mediator Editor is displayed when you double-click a Mediator component in the SOA Composite Editor.
- **Human Task Editor:** You use Oracle JDeveloper to create a human task service component, and then you use Human Task Editor to design the component. Human Task Editor is displayed when you double-click a human task in the SOA Composite Editor.
- **Business Rules Designer:** You use Oracle JDeveloper to create a business rules service component, and then you use Business Rules Designer to design the component. Business Rules Designer is displayed when you double-click a business rule in the SOA Composite Editor.

SOA Composer



SOA Composer provides web-based access to various parameters at run time, allowing them to be edited without redeploying the project that uses them.

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle SOA Composer is an EAR file that is installed as part of the Oracle SOA Suite installation. It provides access to various configuration options at run time, allowing those options to be updated without needing to redeploy the project that uses them. Options typically accessed include domain value maps and business rules.

Quiz

One capability of Oracle Mediator is to route data within a composite application.

- a. True
- b. False

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Answer: a

True. Oracle Mediator routes data from client applications to and from other service components, and from service components to external partners.

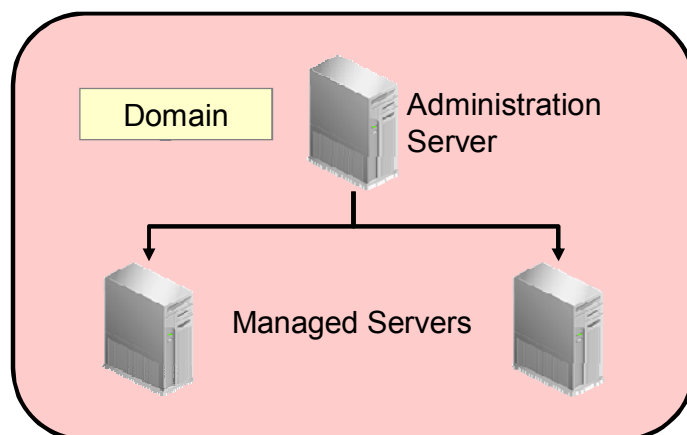
Agenda

- Course Overview
- Introduction to Service-Oriented Architecture
- Oracle SOA Suite 12c Design Time: Overview
- **Oracle SOA Suite 12c Run Time: Overview**
- Installing Oracle SOA Suite 12c
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

WebLogic Application Server



ORACLE

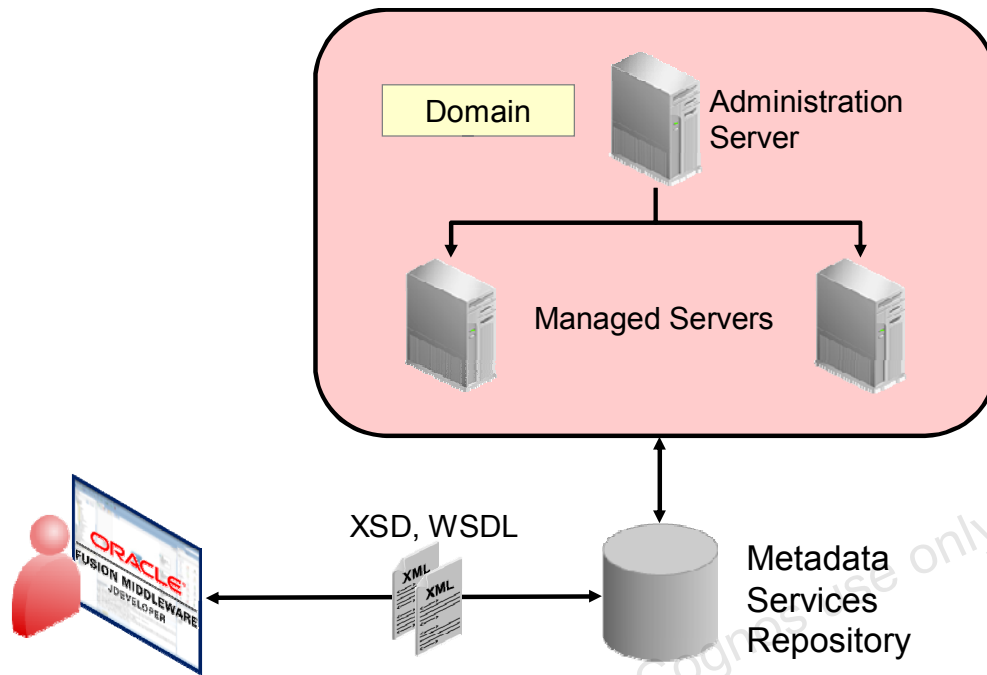
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Because SOA composite applications are Java Enterprise applications, they must run in a Java container. WebLogic Server (WLS) provides the Java EE environment for these components to run. WLS is written in Java and executes on the standard JVM.

The Oracle SOA Suite 12c platform comprises one Oracle WebLogic *Administration server* and one or more Oracle WebLogic *Managed servers*. Managed servers host application components and resources. The Administration server provides a central point for managing logical groups of resources that are organized into *domains*. Domains consist of one or more Managed server instances that are managed through a single Administration server.

Note: There are no WLS programming interfaces that refer to domains. All domain-related information is stored in configuration files. Only an administrator is aware of domains. WLS domains can separate development, test, and production applications; administration and operational responsibilities; or organizational and business divisions.

Oracle Metadata Repository



ORACLE

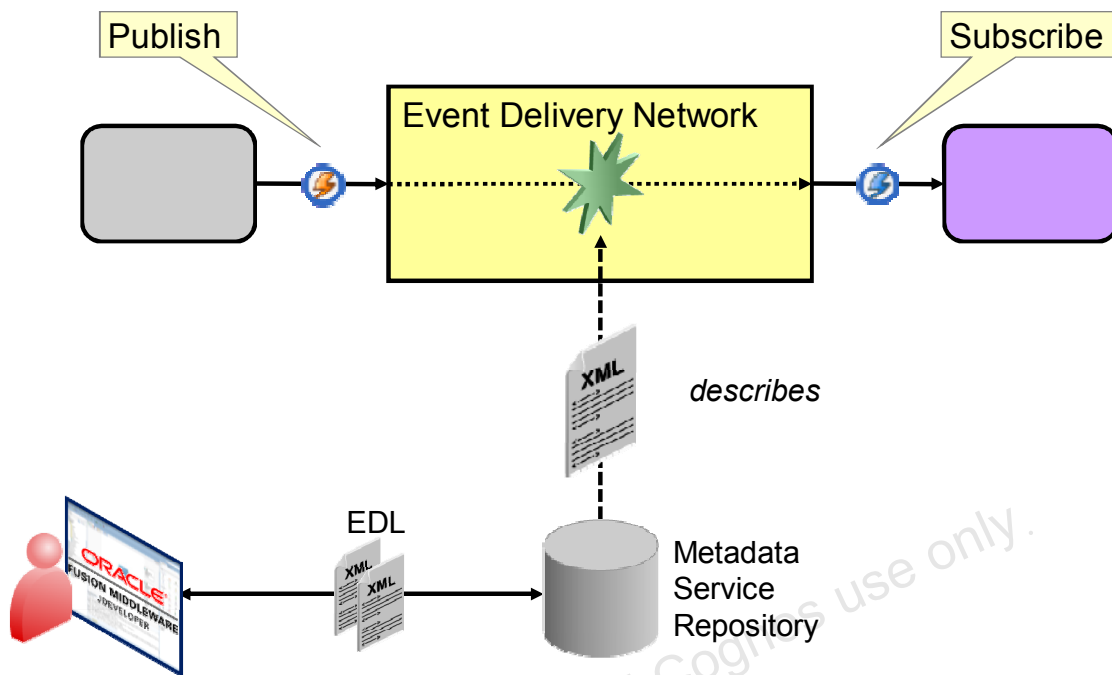
Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The typical Oracle SOA Suite 12c platform comprises one Oracle WebLogic Administration server and one or more Oracle WebLogic Managed servers. The example in the slide shows a typical installation. It includes a domain named `soa_domain`, which includes an Administration server and two Managed servers.

The Oracle SOA Suite 12c runtime environment requires an Oracle database to maintain the SOA application configuration and runtime information. This database is known as the Oracle Metadata Services Repository, or MDS (for MetaData Services). MDS is used to manage deployed services and composite applications and this management happens transparently for the application developers and Oracle SOA Suite 12c administrators.

MDS can also be used as a central location for storing and referencing shared service artifacts, such as business events, rule sets for Oracle Business Rules, XSLT files for Oracle Mediator, XSD and WSDL documents for Oracle BPEL Process Manager, and other service documents, which can be deployed in a sharable archive format known as the Metadata archive (`.mar` files).

Business Events and the Event Delivery Network



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A business *event* is a way for one application to notify another application of a significant occurrence to the business. When a business event is published, another application (or service component) can subscribe to it and initiate whatever processing is implied by that event. For example, when product stock levels are updated in an inventory database, an event can serve as a trigger or signal for another process to fulfill orders that have been on hold until products become available.

A business event definition includes a unique name and a structure to contain information about the event. Developers declaratively define business events and specify conditions that dictate when the event is raised. As data is changed, these conditions are evaluated and all events whose raise conditions are met are fired. The Mediator component can publish or subscribe to events.

Business events are typically asynchronous fire-and-forget (one-way) notifications of a business occurrence. The publisher does not rely on any service component receiving the business event to complete. It does not need to know where subscribers are and what they do with the data.

Note: If the author of the event depends on the receiver of the event, messaging typically must be accomplished through service invocation rather than through a business event. Unlike direct service invocation, the business event separates the client from the server.

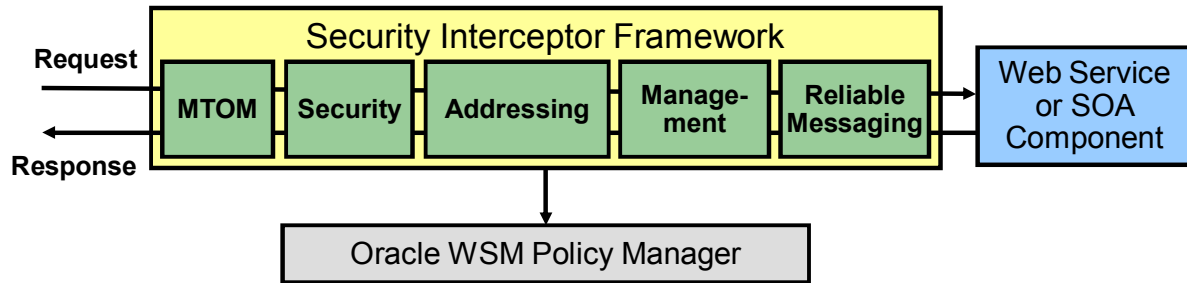
Events are defined by using the Event Definition Language (EDL) to specify the name and structure of an event. Definitions for business events are stored in the Oracle Metadata Services Repository (MDS), and published in the *Event Delivery Network* (EDN).

The Event Delivery Network is designed to handle asynchronous messaging arising from a business or system event. EDN is not messaging infrastructure. It provides declarative publish-subscribe implementations. Subscribers can listen for events by XML namespace, by event name, or by applying content-based XPath filters.

The Event Delivery Network (EDN) provides an application with a way to publish events so that a composite application with a Mediator component can subscribe to events that trigger execution of the composite application. The enforcement points are implemented in such a way that all request and response messages flow through a security pipeline (or filters) that can apply security policies to internal and external message exchanges.

Note: Events will be covered in more detail in the lesson titled “Designing and Managing Business Events.”

Oracle Web Service Manager Policy Manager

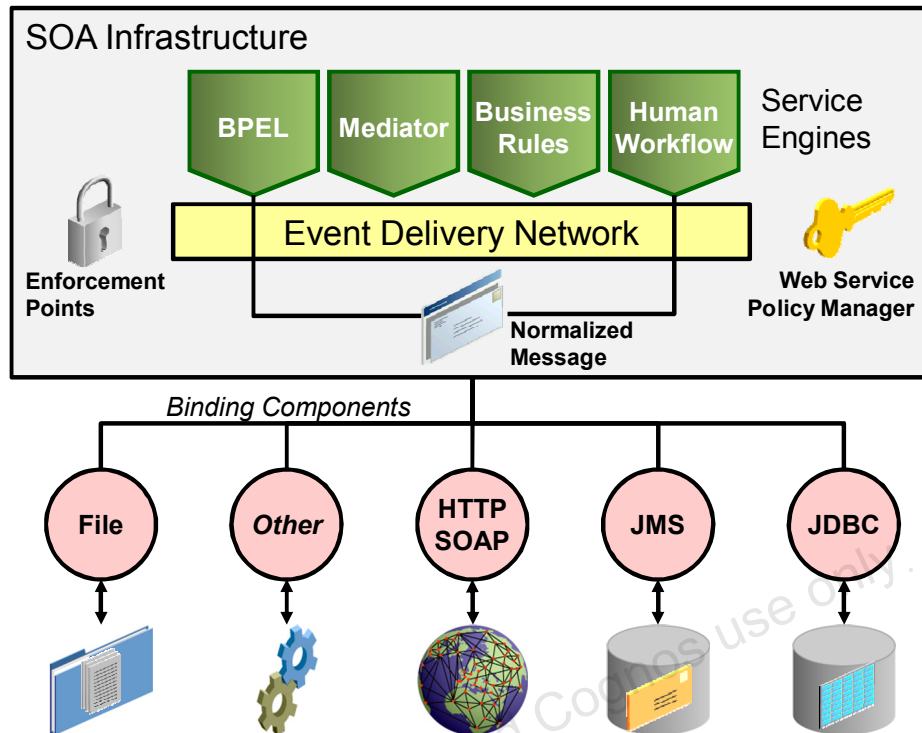


ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Oracle Web Service Manager (WSM) Policy Manager provides the infrastructure for enforcing global security and auditing policies in the service infrastructure. By securing various endpoints and setting and propagating identity, it secures applications. Oracle WSM Policy Manager provides a standard mechanism for signing messages, performing encryption, performing authentication, and providing role-based access control. You can also change a policy without having to change the endpoints or clients for these endpoints, thus providing greater flexibility and security monitoring for your enterprise. In addition, Oracle WSM Policy Manager collects monitoring statistics with information about the quality, uptime, and security threats and displays them in a web dashboard. As a result, Oracle WSM Policy Manager provides better control and visibility over web services.

SOA Run Time



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The SOA infrastructure is a Java EE–compliant application running in Oracle WebLogic Server. The application manages composites and their life cycles, *service engines*, and *binding components*. Service engines execute the business logic for service components. Binding components execute the logic for adapters.

The SOA infrastructure uses a *normalized* message structure while it provides the internal message routing infrastructure capabilities for connecting components and enabling data flow.

Note: A normalized message is a Java Map object where the key(s) is the part name of the message and the value is a Document object representing the XML content.

The SOA infrastructure:

- Receives messages from the service providers or external partners through SOAP binding components, adapters, or the delivery API in the form of XML
- Routes messages based on composite definition to the appropriate service engine

If response messages are returned from the service engine, they are sent to their targets.

Quiz

Business events are a means for service components to exchange data synchronously.

- a. True
- b. False

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Answer: b

False. Business events are typically asynchronous fire-and-forget (one-way) notifications of a business occurrence. The publisher does not rely on any service component receiving the business event to complete. It does not need to know where subscribers are and what they do with the data.

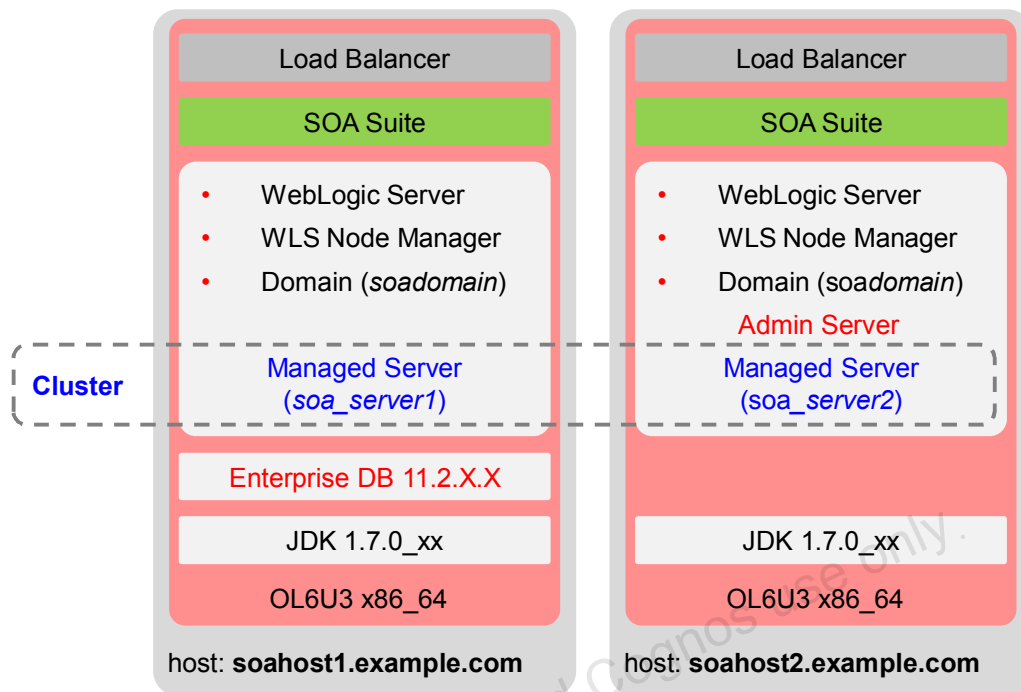
Agenda

- Course Overview
- Introduction to Service-Oriented Architecture
- Oracle SOA Suite 12c Design Time: Overview
- Oracle SOA Suite 12c Run Time: Overview
- **Installing Oracle SOA Suite 12c**
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Production Environment: Example



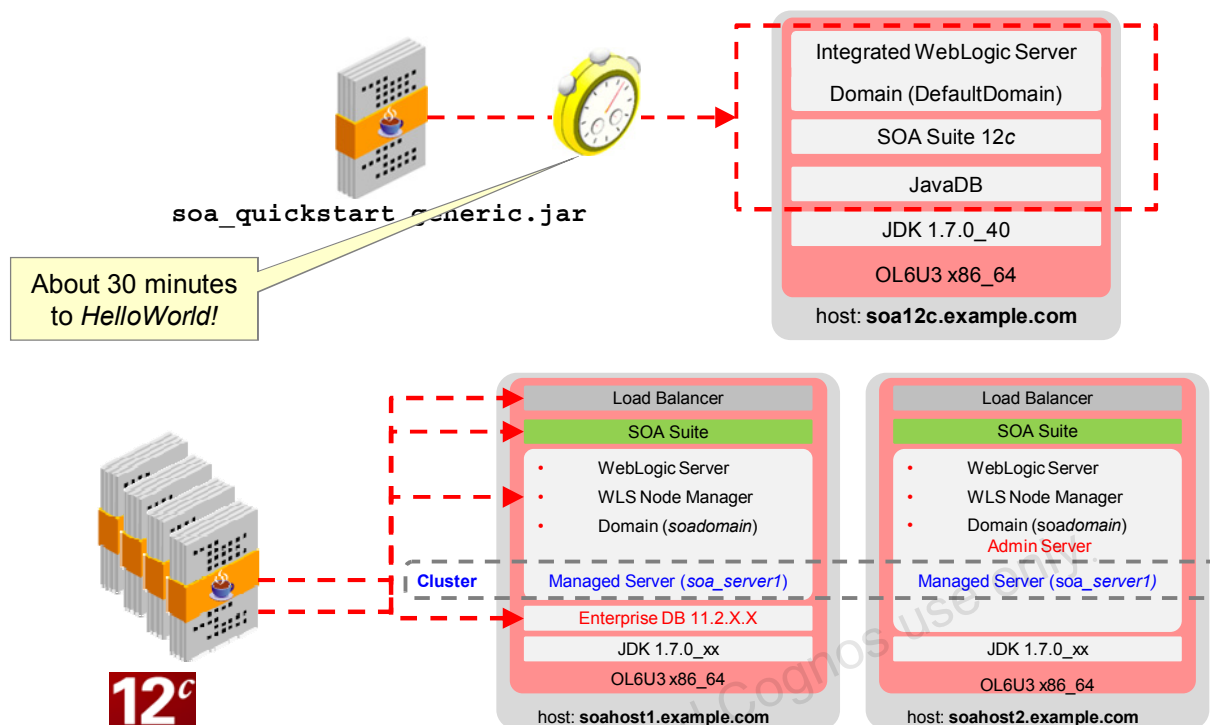
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of a multi-host production topology. It includes two hosts.

- Each is running a version of Linux that is certified to work with 12c.
- Each has a JDK that is certified to work with 12c.
- Each has a WebLogic Server, SOA Suite, WebLogic Server Node managers, and a domain named *soadomain*.
- One host has an Oracle database whose version is certified to work with 12c.
- The other host has the Admin Server running in its domain.
- Both have Managed servers, and those Managed servers are clustered.
- Each has a stand-alone instance of Oracle HTTP Server as a load balancer.

SOA Quick Start Installation



Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A Quick Start installation differs from a full installation in setup speed and limited uses.

The Quick Start installer creates a new Oracle home that contains all the software required for a development or an evaluation environment on a single host.

After spending about 30 minutes running the installation wizard and completing related tasks, you can start developing and testing your SOA applications by running Oracle JDeveloper from the Oracle home.

However, Quick Start installations are limited to one Administration server with no option to add Managed servers.

You also cannot upgrade your Quick Start distribution to a production environment.

Note: If you know that you want a production environment with additional Managed servers or clusters, you should not use Quick Start. Instead, see "Planning the Oracle SOA Suite and Business Process Management Installation" in *Installing and Configuring Oracle SOA Suite and Business Process Management* to install a full-scale development environment that can be scaled to a production environment.

By default, the Oracle SOA Suite Quick Start installation contains Oracle JDeveloper and an Integrated WebLogic Server.

JDeveloper is pre-configured with some JDeveloper IDE extensions so that you can create various kinds of applications immediately. These pre-installed IDE extensions include the following:

- Oracle BPEL Process Manager
- Oracle Human Workflow
- Oracle Business Rules
- Oracle Mediator
- Oracle Service Bus
- Oracle Enterprise Scheduler
- SOA Spring Component Design Time
- Oracle Event Processing Design Time

The Integrated WebLogic Server is pre-configured with Java DB and the runtime software for all the components listed in the preceding paragraph except Oracle Enterprise Scheduler and Oracle Event Processing. This means that you can run any JDeveloper application that does not use Oracle Enterprise Scheduler or Oracle Event Processing on the Integrated WebLogic Server without any additional configuration.

Creating a Domain

Default	Stand-Alone	Compact
<ul style="list-style-type: none">Includes SOA Suite runtime componentsWorks with a pre-configured Java DBLaunches Integrated WLS from JDeveloper. Integrated WLS is shut down every time you end a JDeveloper session.	<ul style="list-style-type: none">Includes SOA Suite runtime componentsWorks with a pre-configured Java DB	<ul style="list-style-type: none">Is configured by the userWorks with an Oracle database that you must install, and then configure with RCU
<ul style="list-style-type: none">Is best for demos or evaluation	<ul style="list-style-type: none">Is best for developers, especially those who want to use the OSB Design Time Console instead of JDeveloper	<ul style="list-style-type: none">Can be used if you want to add SOA components that are incompatible with JavaDB, such as Enterprise Scheduler, MFT, B2B, Healthcare, or BAM

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

After you have installed a Quick Start distribution, you need to configure a domain to test applications during development. JDeveloper is bundled with an integrated application server called the Integrated WebLogic Server. The integrated application server is a Java EE runtime for services that are optimized for the iterative code development cycle.

Launching the Integrated WebLogic Server from JDeveloper generates a domain, which is designated as the default domain. However, you also have the option of launching a JDeveloper-independent version of the WebLogic server to configure a stand-alone domain. The three possible domain types are described and compared in the table in the slide.

Summary

In this lesson, you should have learned how to:

- Describe Service-Oriented Architecture (SOA)
- Explain services and key standards
- Describe Service Component Architecture (SCA)
- Describe the functionality of Oracle SOA Suite 12c service components and adapters
- Describe how WebLogic Application Server is used by Oracle SOA Suite

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Agenda

- Course Overview
- Introduction to Service-Oriented Architecture
- Oracle SOA Suite 12c Design Time: Overview
- Oracle SOA Suite 12c Run Time: Overview
- Installing Oracle SOA Suite 12c
- Practice Overview

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Practice 1 Overview

This practice covers the following topics:

- Performing the SOA Quick Start installation
- Configuring memory settings for improved performance
- Starting the integrated server and configuring a domain

ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

In this practice, you are guided through the Developer “Quick Start” installation process and you should be able to use Oracle SOA Suite in about 30 minutes. The installer does the following:

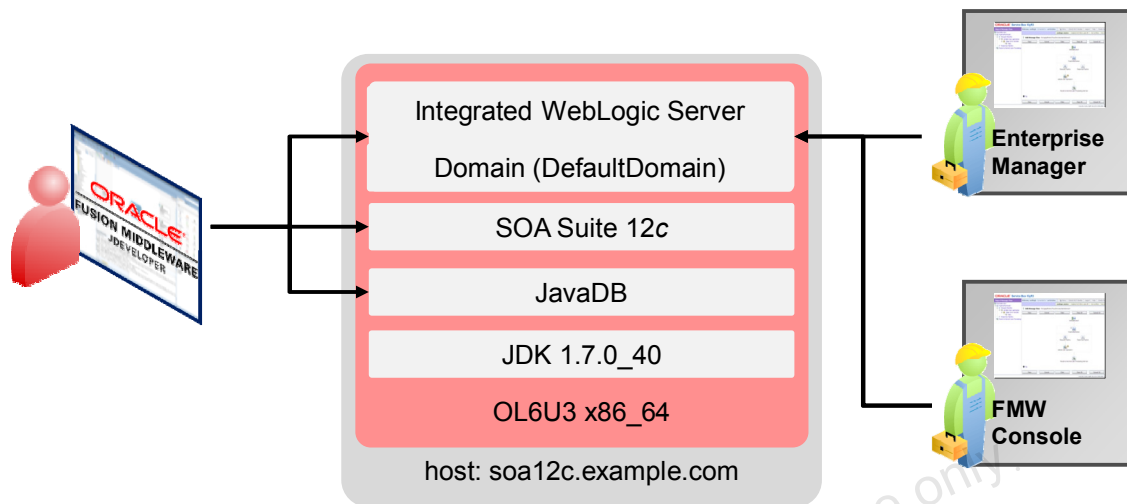
- Installs all the components that are necessary for development with the core of Oracle SOA Suite (BPEL, Business Rules, Mediator, and Human Work Flow)
- Launches JDeveloper after the installation, automatically registering the IDE plug-ins for Oracle SOA Suite

The SOA Suite 12c Developer installation contains:

- JDeveloper
- WebLogic Server
- SOA Suite runtime components
- BPEL, Human Workflow, Rules, Mediator
- Service Bus (not used in this course)
- JDeveloper IDE plug-ins for the preceding components

All software is installed in a single middleware home directory that you specify.

Course Environment



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Your environment consists of a server configured with Linux and Java already installed. In this and subsequent practices, you first install Oracle SOA Suite. You then use JDeveloper, Oracle Enterprise Manager, and Fusion Middleware Console to develop, deploy, monitor, and manage a number of example applications.

Launching the Quick Start Installer

```
# startInstall.sh
```

```
export JAVA_HOME=/usr/bin  
export ORACLE_HOME=/u01/app/fmw12c/Oracle_Home  
cd /home/oracle/labs/software  
$JAVA_HOME/java -jar soa_quickstart_generic.jar
```

Set the environment variable
JAVA_HOME to point to the
location of your JDK.

Optionally, set
ORACLE_HOME.

Run the appropriate .jar file.

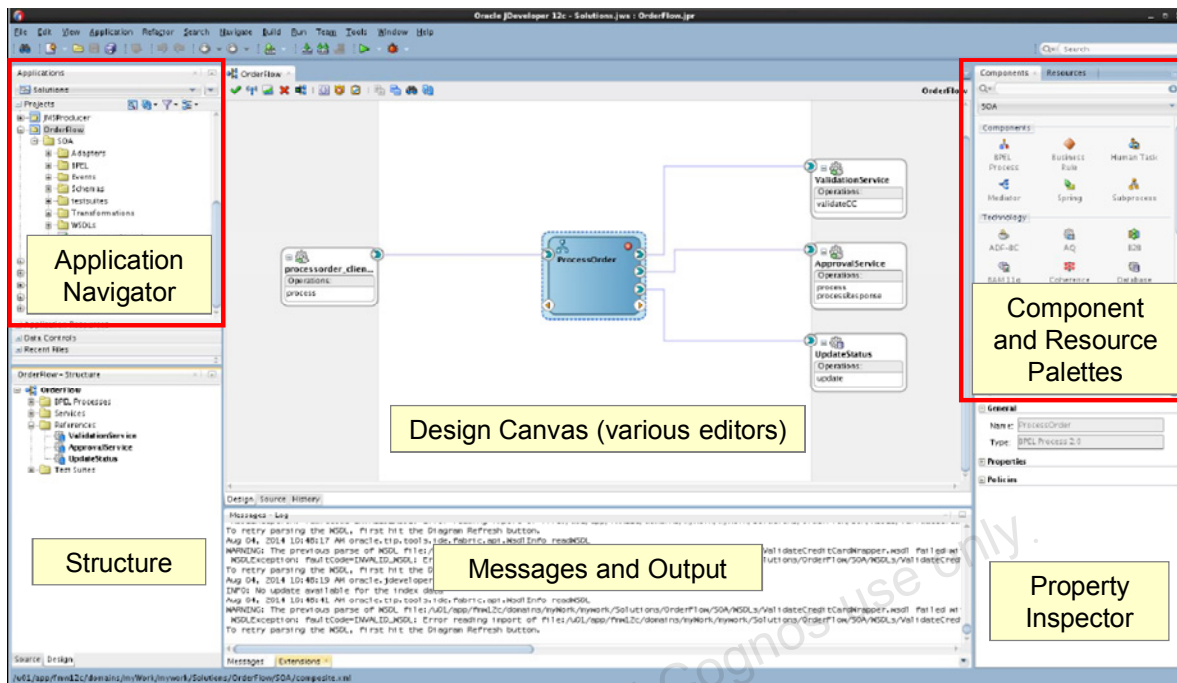
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Before launching the Quick Start installer, set the environment variable `JAVA_HOME` to point to the location of your JDK. You can also set `ORACLE_HOME`. Although this step is not necessary, it will simplify the process of navigating to the various directories when you are running the installation and configuration commands.

Navigate to the directory that includes the installation files. These files may include several .jar files, but you will only need to run the one named `soa_quickstart_generic.jar`.

First Look: JDeveloper



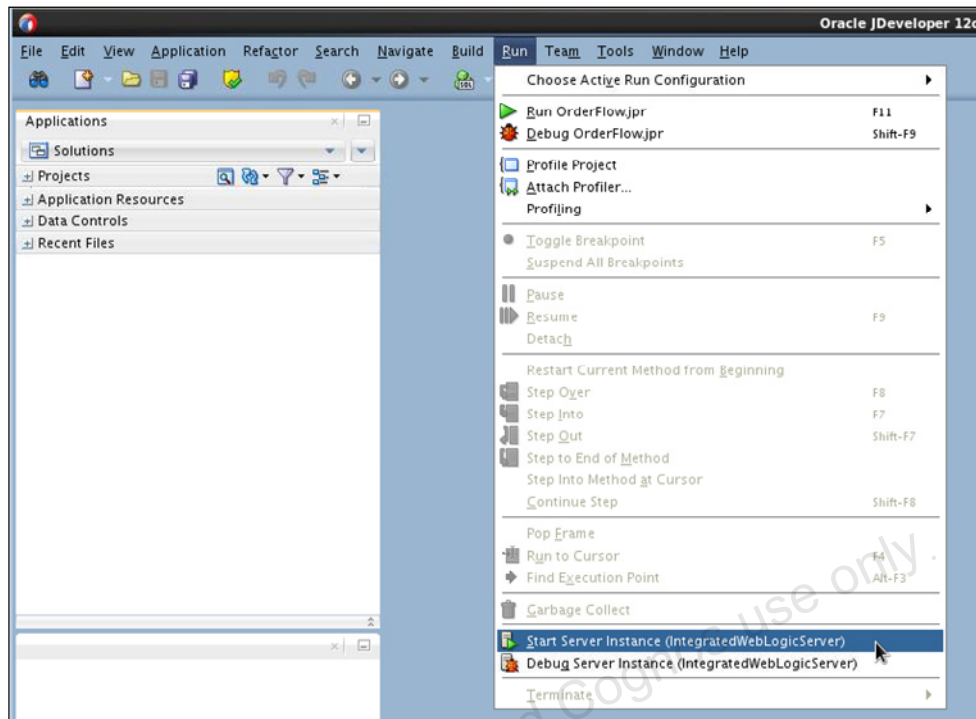
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

A significant amount of work in this course is completed by using JDeveloper. Some important elements of the interface are highlighted in the slide.

- In the upper-left region of the editor is the Application Navigator, which provides access to your applications and projects, as well as the individual files that comprise those projects.
- The large pane in the center is the Design Canvas. It displays different editors, depending on the kind of files that are currently open for editing. Tabs across the top provide access to the individual files that are currently open for editing.
- To the right of the Design Canvas is the Component palette, which provides collections of icons to support the editing functions of whatever editor is currently active in the Design Canvas.
- The Structure pane can be found in the lower-left region of the interface. It provides a hierarchical view of the component or file that is currently being edited in the Design Canvas.
- Properties for the actively edited file can be viewed and edited in the Property Inspector pane, which is found in the lower-right region of the window.
- The lower-center section of the window is reserved for messages and output resulting from various actions, such as deploying an application. You use each of these panes in this course.

Starting the Integrated Server



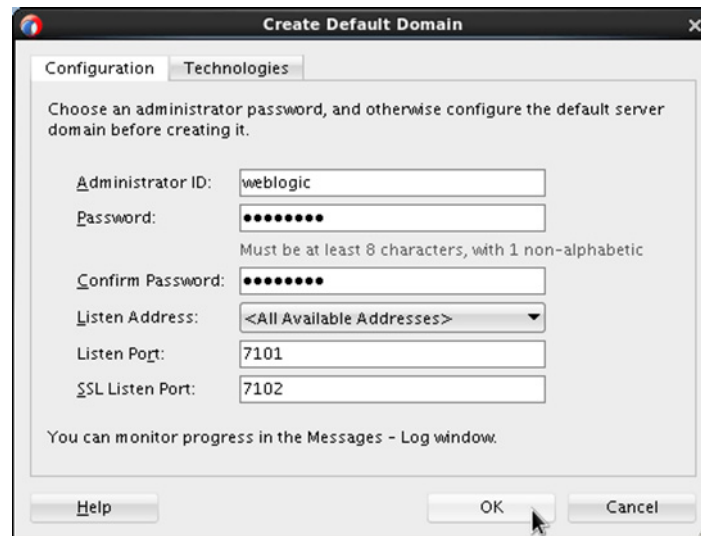
ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

JDeveloper is bundled with an integrated application server called the Integrated WebLogic Server. The integrated application server is a Java EE runtime for services that are optimized for the iterative code development cycle.

Launch the Integrated WebLogic Server by selecting Run from top menu bar. Select Start Server Instance from the drop-down menu, as shown in the slide.

Configuring a Domain



ORACLE

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

The first time that you launch the server instance, you are prompted to enter a password for your default domain, as seen in the screenshot in the slide. The Administrator ID, Listen Address, Listen Port, and SSL Listen Port fields should already have values. Review them and make any appropriate changes.

Launching the integrated server will take several minutes. You can track the server's launch in the Messages window pane at the bottom of the JDeveloper screen. When the following messages appear in the log, the Integrated WebLogic Server has launched successfully.

- `<Notice> <WebLogicServer> <BEA-000360> <The server started in RUNNING mode.>`
- `<Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING.>`
- SOA Platform is running and accepting requests

Oracle University and Cognos use only.