

# 18

## **Security Concepts and Configuration**

# Objectives

After completing this lesson, you should be able to do the following:

- Use the WLS security architecture
- Configure security realms
- Configure users and groups
- Configure roles
- Configure policies
- Configure protection for:
  - Web application resources
  - EJBs

# Road Map

- Security overview
  - Oracle Platform Security Services
  - Oracle WLS Security
  - Oracle WLS Security Models
  - Introduction to WLS Security components
- Users and groups
- Protecting application resources

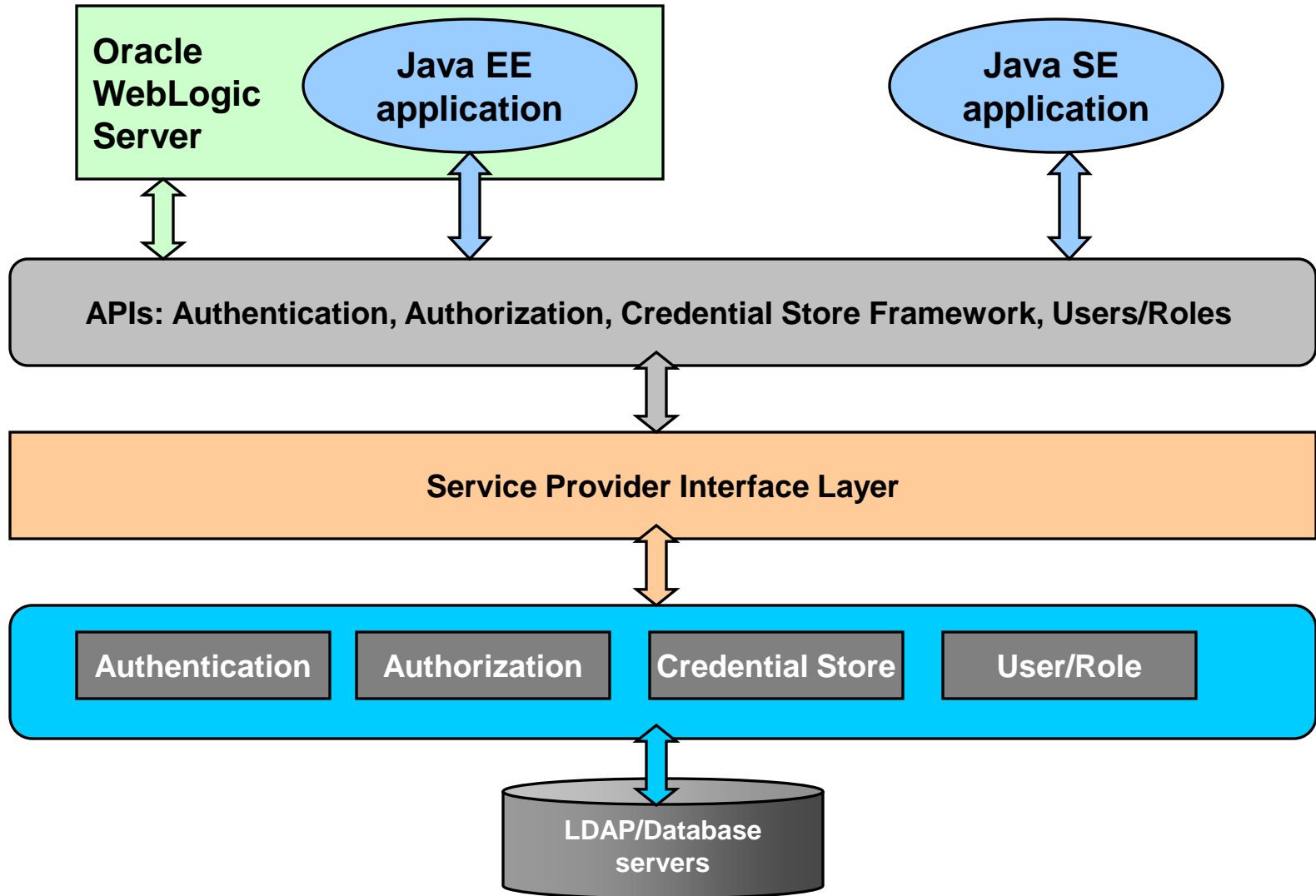


# Introduction to Oracle WebLogic Security Service

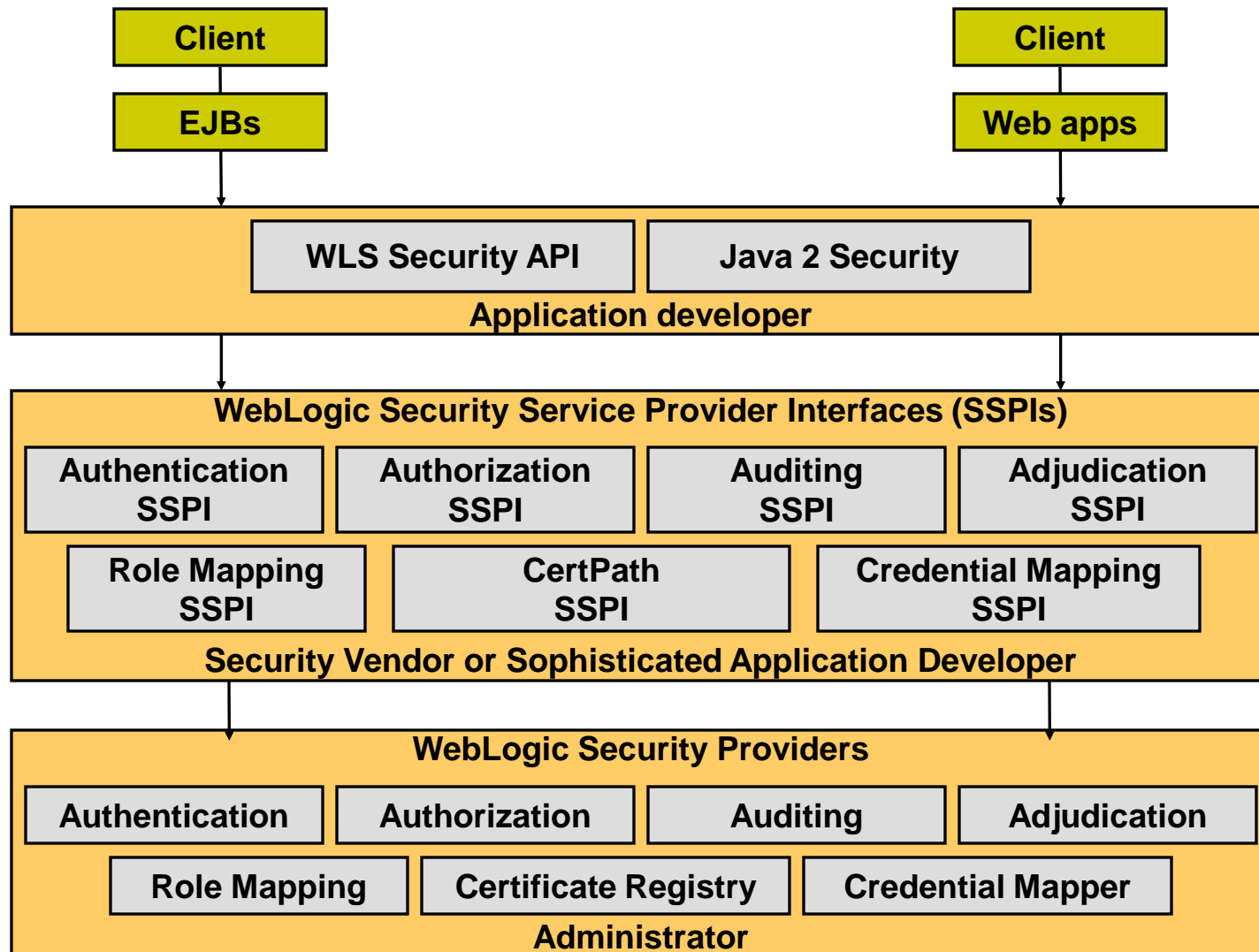
- Security is a challenge in environments with diverse applications and Web-based services.
- This requires established and well-communicated security policies and procedures.
- You can use Oracle WebLogic Server as a comprehensive and flexible security infrastructure to protect applications.



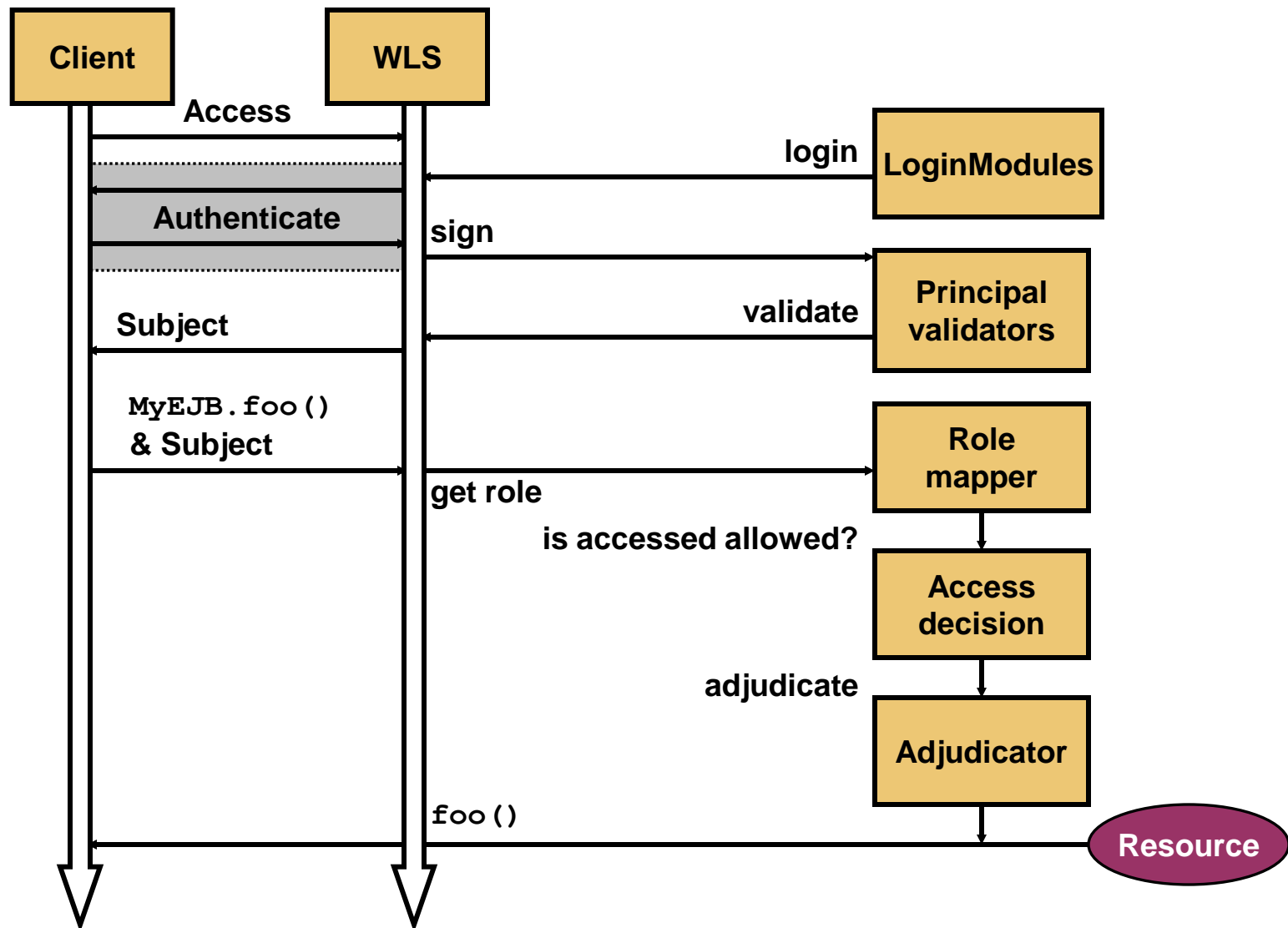
# Oracle Platform Security Services



# Oracle WLS Security Architecture



# Security Services



# Overview of Security Concepts

- Authentication providers handle identity information and make it possible to associate with users, groups, or roles.
- Identity assertion providers map a valid token to an Oracle WebLogic Server user.
- An authorization provider is a process that is used to control the interactions between users and resources based on user identity.
- The adjudication provider weighs the results that multiple access decisions return to determine the final decision.
- The credential mapping process is initiated when application components access the authentication mechanism of a legacy system to obtain a set of credentials.
- Auditing provides a trail of activity. The auditing provider is used to log activity before and after security operations.



# Confidentiality

- Oracle WebLogic Server supports the Secure Sockets Layer (SSL) protocol to secure the communication between the clients and the server.
- The SSL client authentication allows a server to confirm a user's identity by verifying that a client's certificate and public ID are valid and are issued by a Certificate Authority (CA).
- The SSL server authentication allows a user to confirm a server's identity by verifying that the server's certificate and public ID are valid and are issued by a CA.

# Credential Mapping

- The credential mapping process is used when application components access the authentication mechanism of an external system to obtain a set of credentials.
- The requesting application passes the subject as part of the call and information about the type of credentials required.
- Credentials are returned to the security framework, which is then passed to the requesting application component.
- The application component uses the credentials to access the external system.

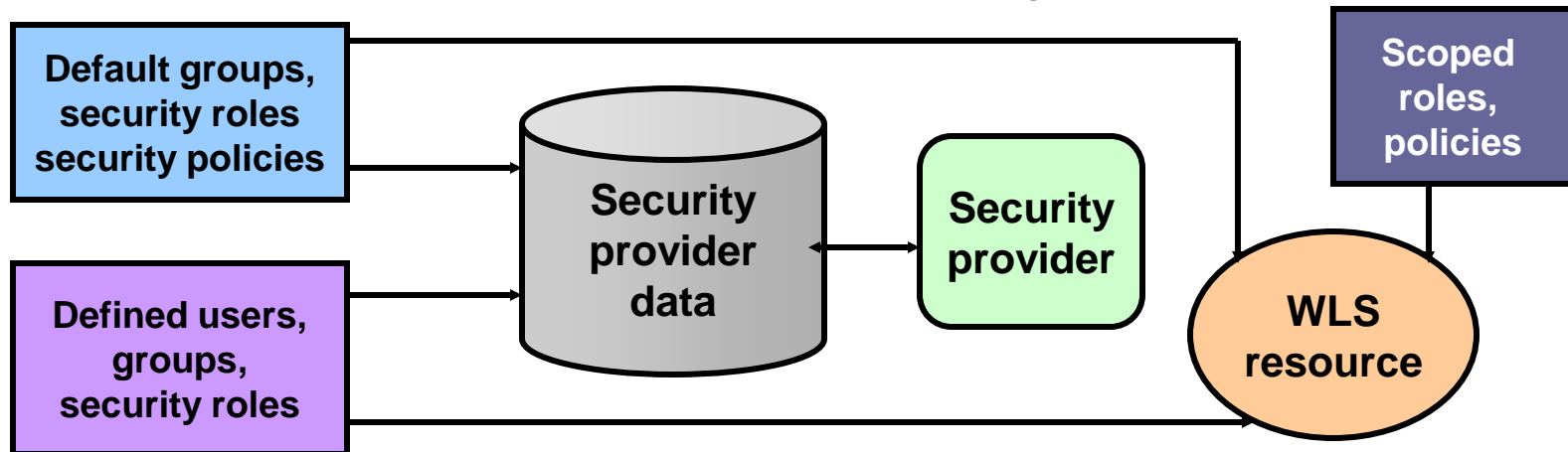
# Road Map

- Security overview
- Users and groups
  - Security realms
  - Embedded LDAP
  - Configuring users, groups, and roles
- Protecting application resources



# Security Realms

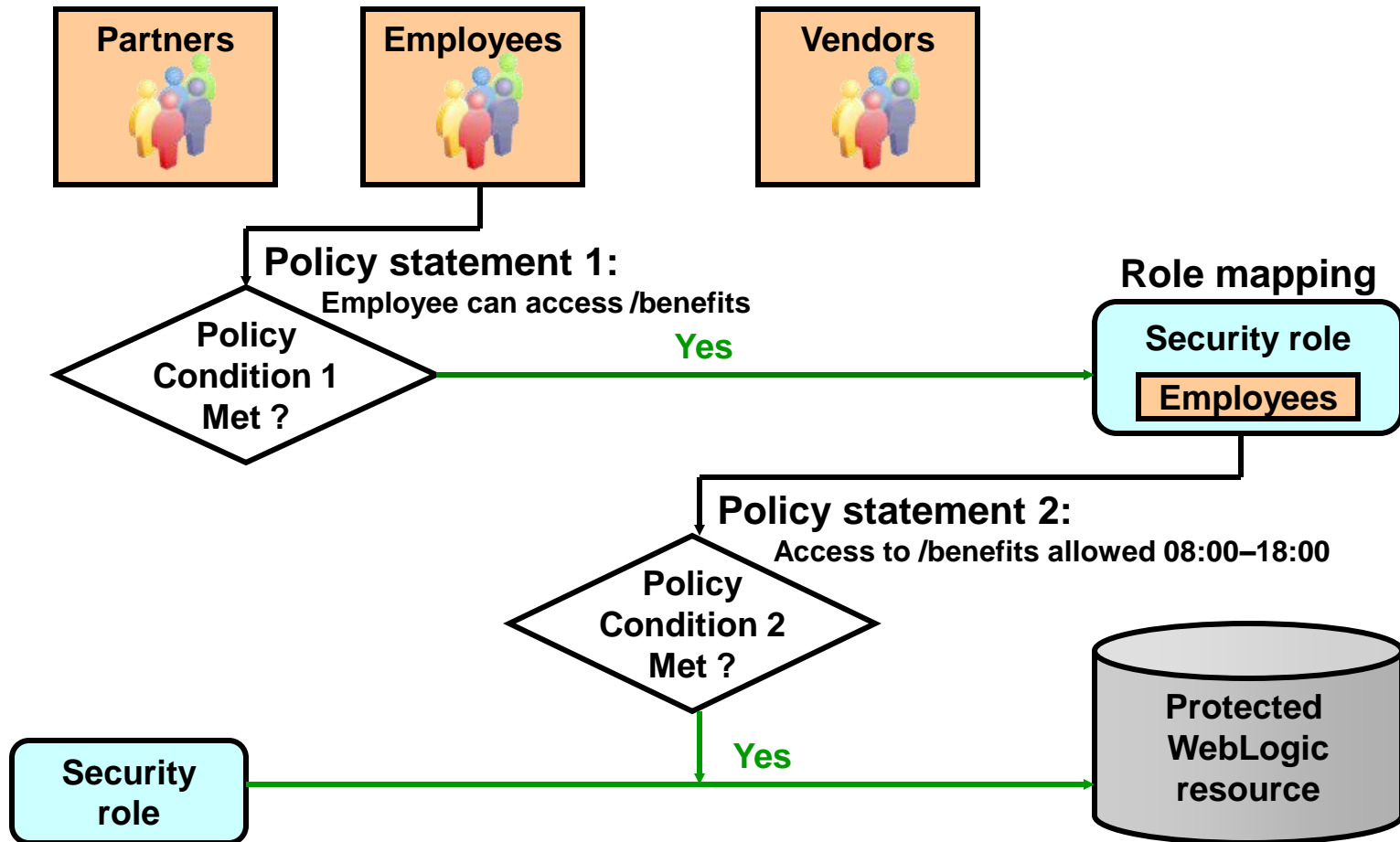
- A security realm is a collection of system resources and security service providers.
- A valid user must be authenticated by the authentication provider in the security realm.
- Only one security realm can be active at a given time.
- A single security policy can be used in any realm.
- Administration tasks include creating security realms.



# Security Model Options for Applications

Security Model	Location of Users, Roles, and Policies	Security Checks Performed
Deployment Descriptor Only (Java EE standard)	Deployment descriptors: <ul style="list-style-type: none"><li>• <code>web.xml</code> and <code>weblogic.xml</code></li><li>• <code>ejb-jar.xml</code> and <code>weblogic-ejb-jar.xml</code></li></ul>	Only when clients request URLs or EJB methods that are protected by a policy in the deployment descriptor
Custom Roles	Role mappings from a role mapping provider that you configure for the security realm  Policies are defined in the <code>web.xml</code> and <code>ejb-jar.xml</code> deployment descriptors.	Only when clients request URLs or EJB methods that are protected by a policy in the deployment descriptor.
Custom Roles and Policies	Role mappings and authorization from providers that you configure for the security realm	For all URLs and EJB methods in the application
Advanced	This model is fully flexible. You can import security data from deployment descriptors into the security provider databases to provide a baseline.	Configurable

# How WLS Resources Are Protected



# Users and Groups

- Users are entities that use WLS, such as:
  - Application end users
  - Client applications
  - Other Oracle WebLogic Servers
- Groups are:
  - Logical sets of users
  - More efficient for managing a large number of users



# Configuring New Users

The screenshot displays the Oracle Identity Management console interface. On the left, the 'Settings for myrealm' sidebar shows the 'Users and Groups' tab selected, with the 'Users' sub-tab active. A 'New' button is highlighted with a mouse cursor. The main area on the right is the 'Create a New User' dialog box, which contains the following fields and controls:

- Buttons:** 'OK' and 'Cancel' buttons at the top and bottom of the dialog.
- Name:** A text field labeled '\* Name:' containing the value 'John Doe'.
- Description:** A text field labeled 'Description:' containing the value 'User for Benefits application'.
- Provider:** A dropdown menu labeled 'Provider:' with 'DefaultAuthenticator' selected.
- Password:** A text field labeled 'Password:' containing masked characters (dots).
- Confirm Password:** A text field labeled 'Confirm Password:' containing masked characters (dots).

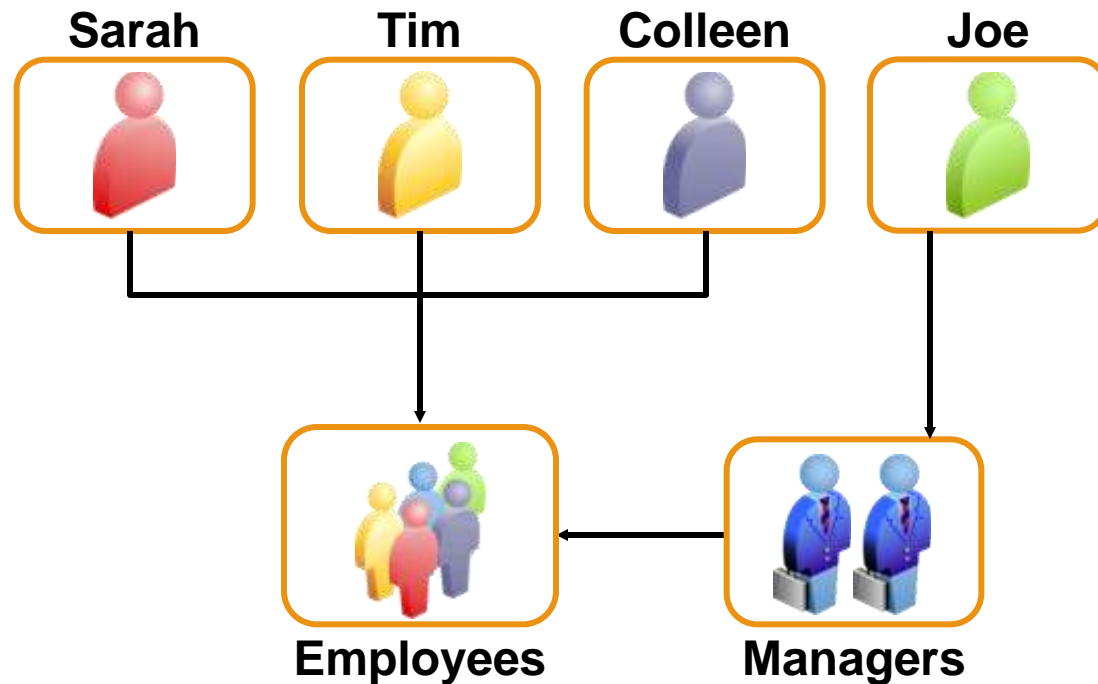
The 'OK' button at the bottom left of the dialog is being clicked by the mouse cursor.



# Groups

WLS provides the flexibility to organize groups in various ways:

- Groups can contain users.
- Groups can contain other groups.



# Configuring New Groups

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users **Groups**

[Customize this table](#)

**Groups**

Create a New Group

What would you like to name your new Group?

\* **Name:**

How would you like to describe the new Group?

**Description:**

Please choose a provider for the group.

**Provider:**

# Configuring Group Memberships

Settings for jdoe

General Passwords **Groups**

Save

**Parent Groups:**

Available	Chosen
AdminChannelUsers Administrators <b>AppTesters</b> CrossDomainConnectors Deployers Monitors	

Annotations: A green circle with the letter 'A' is positioned above the 'Groups' tab. A mouse cursor is pointing at the right arrow button between the 'Available' and 'Chosen' columns.

Settings for Deployers

General **Membership**

**Parent Groups:**

Available	Chosen
AdminChannelUsers Administrators CrossDomainConnectors Monitors Operators OrderSystemGroup	AppTesters

Annotations: A green circle with the letter 'B' is positioned above the 'Membership' tab. A mouse cursor is pointing at the right arrow button between the 'Available' and 'Chosen' columns. A tooltip box contains the text: "Move selected items from Available column to Chosen column".

# Road Map

- Security overview
- Users and groups
- Roles and policies
  - Security roles
  - Security policies
  - Defining policies and roles
  - Protecting Web resources
  - Protecting other resources



# Security Roles

- A role refers to a set of permissions granted to a user or group.
- A role differs from a group; a group has static membership, whereas a role is conditional.
- A user and group can be granted multiple roles.
- The two types of roles are global-scoped roles and resource-scoped roles.
- The global roles that are available by default are Admin, Operator, Deployer, Monitor, AppTester, and Anonymous.
- Roles defined in deployment descriptors can be inherited.
- You can manage role definitions and assignments without editing deployment descriptors or redeploying the application.

# Configuring the Global Security Role

The screenshot displays the Oracle Identity Manager console interface. On the left, the 'Domain Structure' tree shows 'MedRecDomain' expanded, with 'Security Realms' highlighted. A red box around 'Security Realms' has an arrow pointing to the 'Roles and Policies' tab. The 'Roles and Policies' tab is also highlighted with a red box. Below it, the 'Realm Roles' sub-tab is highlighted. A red box around the 'Roles' link at the bottom of the console has an arrow pointing to the 'Create a New Role for this Realm' dialog box. The dialog box is open, showing the 'Role Properties' section. The 'Name' field is filled with 'MyTest' and the 'Provider Name' dropdown is set to 'XACMLRoleMapper'. The 'OK' button is highlighted with a red box. The main console area shows a table of roles with columns 'Name' and 'Type'. The 'Global Roles' section is expanded, showing a list of roles. The 'Roles' link at the bottom of the console is highlighted with a red box.

Lock & Edit  
Release Configuration

Configuration Users and Groups **Roles and Policies** Credential Mappings Providers

**Realm Roles** Realm Policies

Domain Structure

- MedRecDomain
  - Environment
  - Deployments
  - Services
  - Security Realms**
  - Interoperability
  - Diagnostics

How do I...

- Manage security roles
- Create scoped roles for JNDI resources
- Create scoped roles for WorkContext resources

System Status

Use this table to view, add, modify or remove global or scoped security roles for this security realm. The Name column under the Global Roles node. Scoped roles are listed in the Name column under the scoped roles node.

Notes:

- This table does not allow you to view the Security Roles page.
- If you imported a role from a Security Roles page, the role is listed in the Name column under the scoped roles node.

**Roles**

Edit Role

Name
Deployments
Domain
Global Roles

**Create a New Role for this Realm**

OK Cancel

**Role Properties**

The following properties will be used to identify your new role.

\* Indicates required fields

What would you like to name your new role?

\* Name: MyTest

Which role mapper would you like to use with this role?

Provider Name: XACMLRoleMapper

OK Cancel

# Security Policies

- Security policies implement parameterized authorization.
- Security policies comprise rules and conditions.
- Users and groups that adhere to the security policy are granted access to resources protected by the policy.
- Security policies follow a hierarchy. The policy of a narrower scope overrides that of a broader scope.
- When you install Oracle WebLogic Server, some default root-level policies are provided.

# Policy Conditions

- Policy conditions are the essential components of a policy.
- The WebLogic Server authorization provides three kinds of built-in policy conditions in the Administration Console:
  - Basic policy conditions
  - Date and Time policy conditions
  - Context Element policy conditions



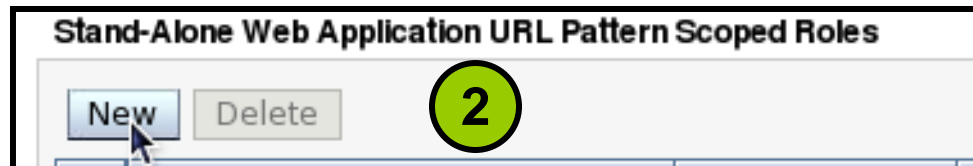
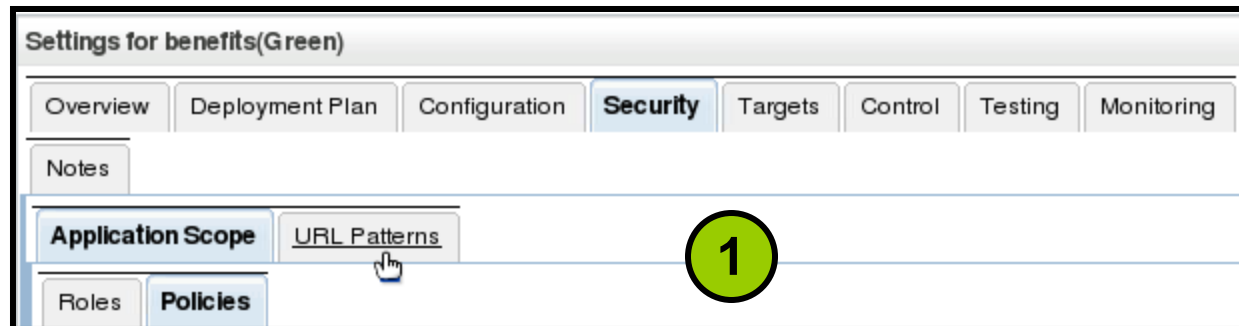
# Protecting Web Applications

To protect a Web application with declarative security, perform the following steps:

1. Define the roles that should access the protected resources.
2. Determine the Web application resources that must be protected.
3. Map the protected resources to roles that should access them.
4. Map roles to users or groups in the WLS security realm.
5. Set up an authentication mechanism.

# Specifying Protected Web Resources

Protection for Web resources are defined based on URL patterns.



Example URL patterns:

URL Pattern	Role Name
/*	Some role name (for example, director)
/*.jsp	employee
/EastCoast/*	east-coaster

# Defining Policies and Roles for Other Resources

You can define roles and policies for other resources, such as JDBC and JMS.

**Summary of JDBC Data Sources**

A JDBC data source is an object bound to the JNDI tree that provides a way for applications to look up a data source on the JNDI tree and then use it.

This page summarizes the JDBC data sources that are available to your application.

[Customize this table](#)

**Data Sources(Filtered - More Controls)**

Click the **Lock & Edit** button in the toolbar to edit the data source.

<input type="checkbox"/>	Name
<input type="checkbox"/>	MedRecGlobalSource
<input type="checkbox"/>	testSample

**Settings for testSample**

Configuration Targets Monitoring Control **Security** Notes

**Roles** Policies Credential Mappings

**Policy Conditions**

These conditions determine the access control to your JDBC data source.

Add Conditions Combine Uncombine Move Up Move Down Remove Negate

No Policy Specified

**Choose a Predicate**

Choose the predicate you wish to use as your new condition.

The predicate list is a list of available predicates which can be used to define the access control to your JDBC data source.

**Predicate List:** Access occurs between specified

Back Next Finish Cancel

**Edit Arguments**

On this page you will fill in the arguments that pertain to the predicate you have chosen.

The earliest permissible time in the format of "h:mm AM/PM". E.g.: "12:45:35 PM".

Starting time: 08:00 AM

The latest permissible time in the format of "h:mm AM/PM". E.g.: "12:45:35 PM".

Ending time: 07:00 PM

The time ahead of GMT (enter as "GMT+h:mm") or behind GMT (enter as "GMT-h:mm"). E.g.: "GMT+5:00". Time in the USA would be "GMT-5:00".

GMT offset: GMT-8:00

Back Next Finish Cancel

# Embedded LDAP Server


- In WLS, users, groups, and authorization information are stored in an embedded LDAP server.
- Several properties can be set to manage the LDAP server, including:
  - Credentials
  - Backup settings
  - Cache settings
  - Replication settings


# Configuring an Embedded LDAP


Settings for MedRecDomain


Configuration Monitoring Control **Security** Web Service Security Notes


General Filter Unlock User **Embedded LDAP** Roles Policies


 **Credential:**


 **Confirm Credential:**


 **Backup Hour:**


 **Backup Minute:**


 **Backup Copies:**


☒  **Cache Enabled**


 **Cache Size:**

 **Cache TTL:**

☐  **Refresh Replica At Startup**

☐  **Master First**

 **Timeout:**

☐  **Anonymous Bind Allowed**

# Configuring Authentication

Configure how a Web application determines the security credentials of users:

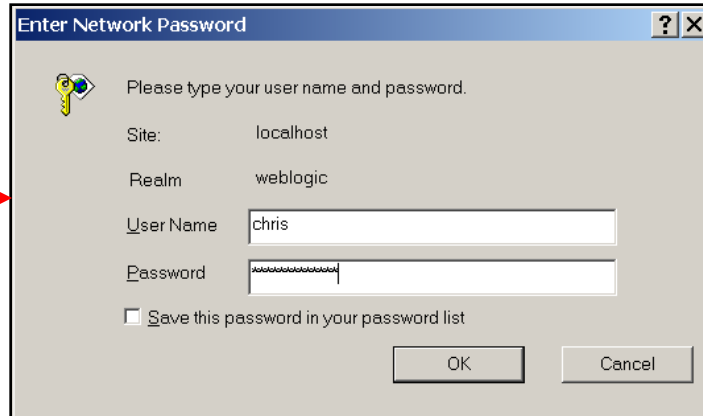
- BASIC: The Web browser displays a dialog box.
- FORM: Use a custom HTML form.
- CLIENT-CERT: Request a client certificate.

Configure the authentication using the `<login-config>` element:

```
<login-config>
  <auth-method>BASIC, FORM, or CLIENT-CERT</auth-method>
  <form-login-config>
    <form-login-page>login.jsp</form-login-page>
    <form-error-page>badLogin.jsp</form-error-page>
  </form-login-config>
</login-config>
```

# Authentication Examples

**BASIC  
authentication**



A Windows-style dialog box titled "Enter Network Password". It contains a key icon and the text "Please type your user name and password." Below this, it shows "Site: localhost" and "Realm: weblogic". There are input fields for "User Name" (containing "chris") and "Password" (masked with dots). A checkbox labeled "Save this password in your password list" is unchecked. At the bottom are "OK" and "Cancel" buttons.

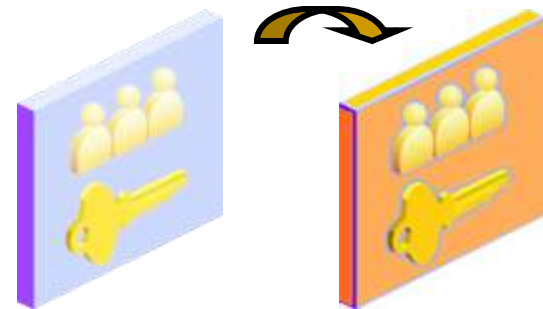
**FORM-based  
authentication**



The Oracle WebLogic Server 11g Administration Console login screen. It features the Oracle logo and "WebLogic Server® 11g Administration Console" text. A large "11g" watermark is in the background. On the right, a "Welcome" box says "Log in to work with the WebLogic Server domain" and contains "Username:" and "Password:" input fields, followed by a "Log In" button.

# Migrating Security Data

- You can export users and groups, security policies, security roles, or credential maps between security realms or domains.
- It is useful, for example, in transitioning from development to QA to production.
- You can use migration constraints (key/value pairs) to specify the export/import options.
- Currently, the system supports migrating only security data between the WLS security providers.





# Exporting the WLS Default Authenticator Provider

The screenshot shows the 'Settings for DefaultAuthenticator' dialog box with the 'Migration' tab selected. The 'Export' button is highlighted. The 'Export Format' is set to 'DefaultAtn'. The 'Export File on Server' field contains the path '/u01/app/oracle/user\_projects/domains/MedRecDomain/I'. The 'Overwrite' checkbox is unchecked. The 'Supported Export Constraints' section lists 'users, groups, passwords'. The 'Export Constraints (key=value):' field is empty.

Settings for DefaultAuthenticator

Configuration Performance **Migration**

Import **Export**

Save

Export Format: DefaultAtn

\* Export File on Server: /u01/app/oracle/user\_projects/domains/MedRecDomain/I

☐ Overwrite

Supported Export Constraints: users, groups, passwords

Export Constraints (key=value):

# Importing into a Different Domain

Settings for DefaultAuthenticator

Configuration Performance **Migration**

**Import** Export

Save

Import Format: DefaultAtn ▼

\* Import File on Server: /u01/app/oracle/user\_projects/domains/MedRecDomain/I

Supported Import Constraints: None

Import Constraints (key=value):

# Summary

In this lesson, you should have learned how to:

- Use the WLS security architecture
- Configure users, groups, and roles
- Configure roles
- Configure policies
- Configure protection for:
  - Web application resources
  - EJBs
- Configure security realms

# **Practice 18: Overview**

## **Configuring Security for WLS Resources**

This practice covers the following topics:

- Creating new users using the Administration Console
- Creating groups of employees and managers
- Assigning groups to users
- Configuring groups-to-role mapping
- Defining resources that are protected by the security you have configured
- Verifying that the security protection that you enabled is working