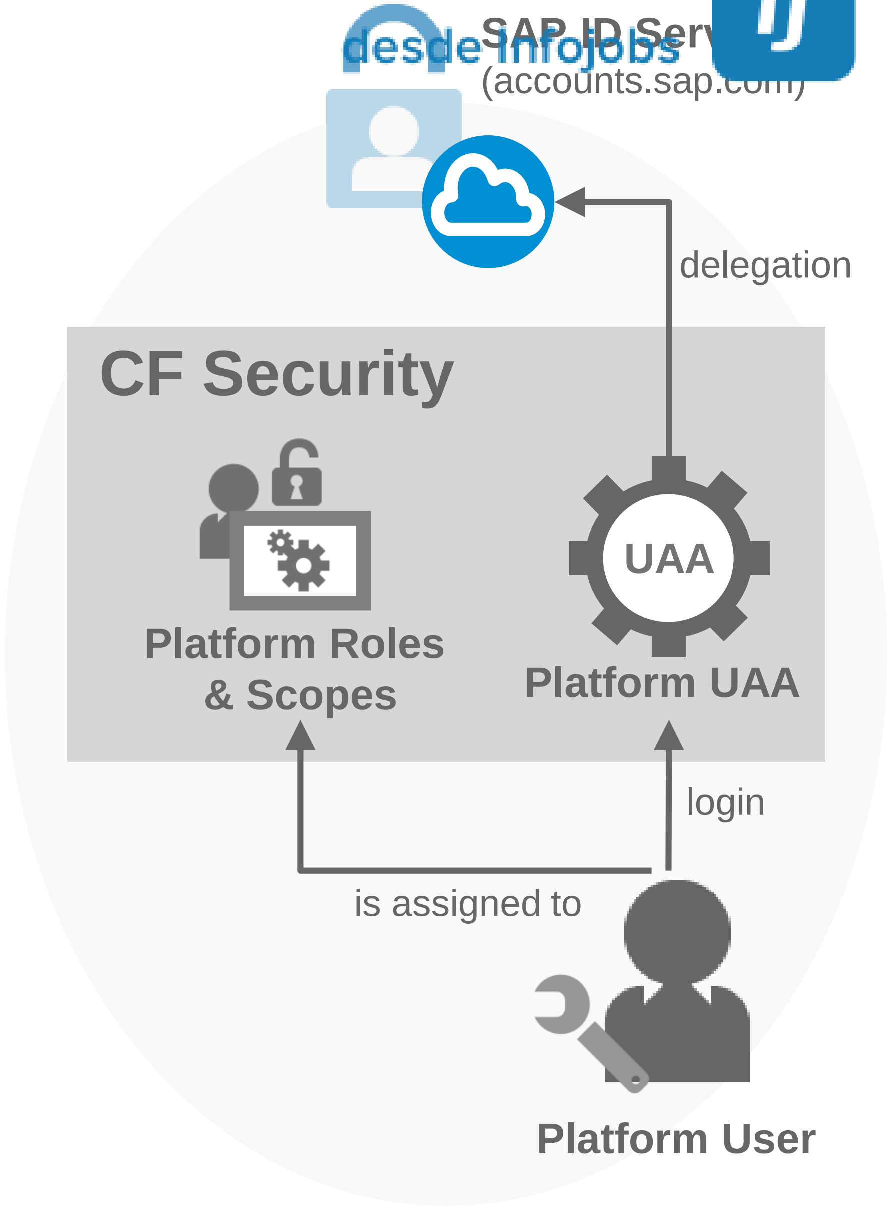
Week 5: Security

Unit 5: Securing Cloud Foundry Applications – Part I

Securing Cloud Foundry Applications – Part I

Cloud Foundry native security model

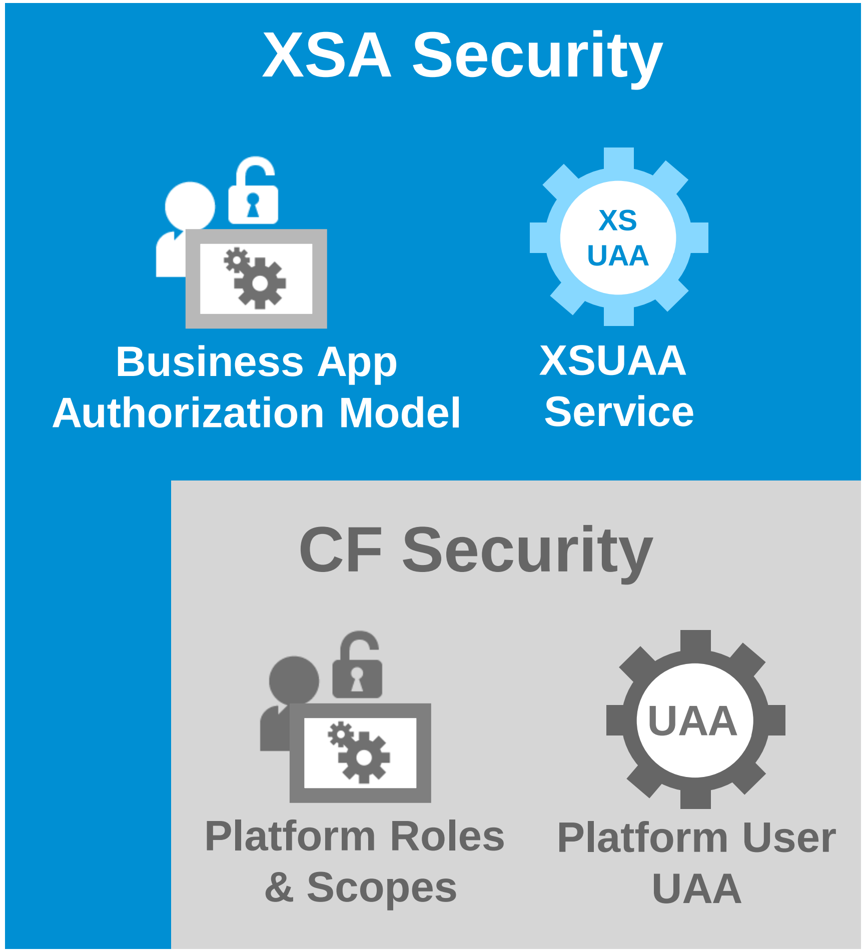
- Cloud Foundry's (CF) native security model provides a framework for authentication and authorization for **platform users** accessing the CF environment
- By default, platform user accounts are persisted in the **User Account and Authentication (UAA)** service of the Cloud Foundry installation
- On SAP Cloud Platform, the **platform (user) UAA** delegates authentication to **SAP ID Service**
- The authorization model of CF has a **predefined set of platform authorizations**
- **Assignments** of platform users to org and space-level platform roles are persisted in the Cloud Controller database



Securing Cloud Foundry Applications – Part I

Cloud Foundry security model enhancements in SAP Cloud Platform

- The SAP Cloud Platform Cloud Foundry environment offers the **SAP HANA extended application services**, **advanced model** (XSA) as a programming model to choose from
- XSA enhances the Cloud Foundry security model by adding security functionality for **web-based business applications**
- XSA defines a flexible authorization model for business applications by introducing design and runtime components
- With XSA, authentication and identity management for business users can be delegated to any SAML 2.0-compliant identity provider via the multitenant XSUAA service
- Business applications integrate the XSUAA service via a service broker



XSA Security

Business App Authorization Model

XSUAA Service

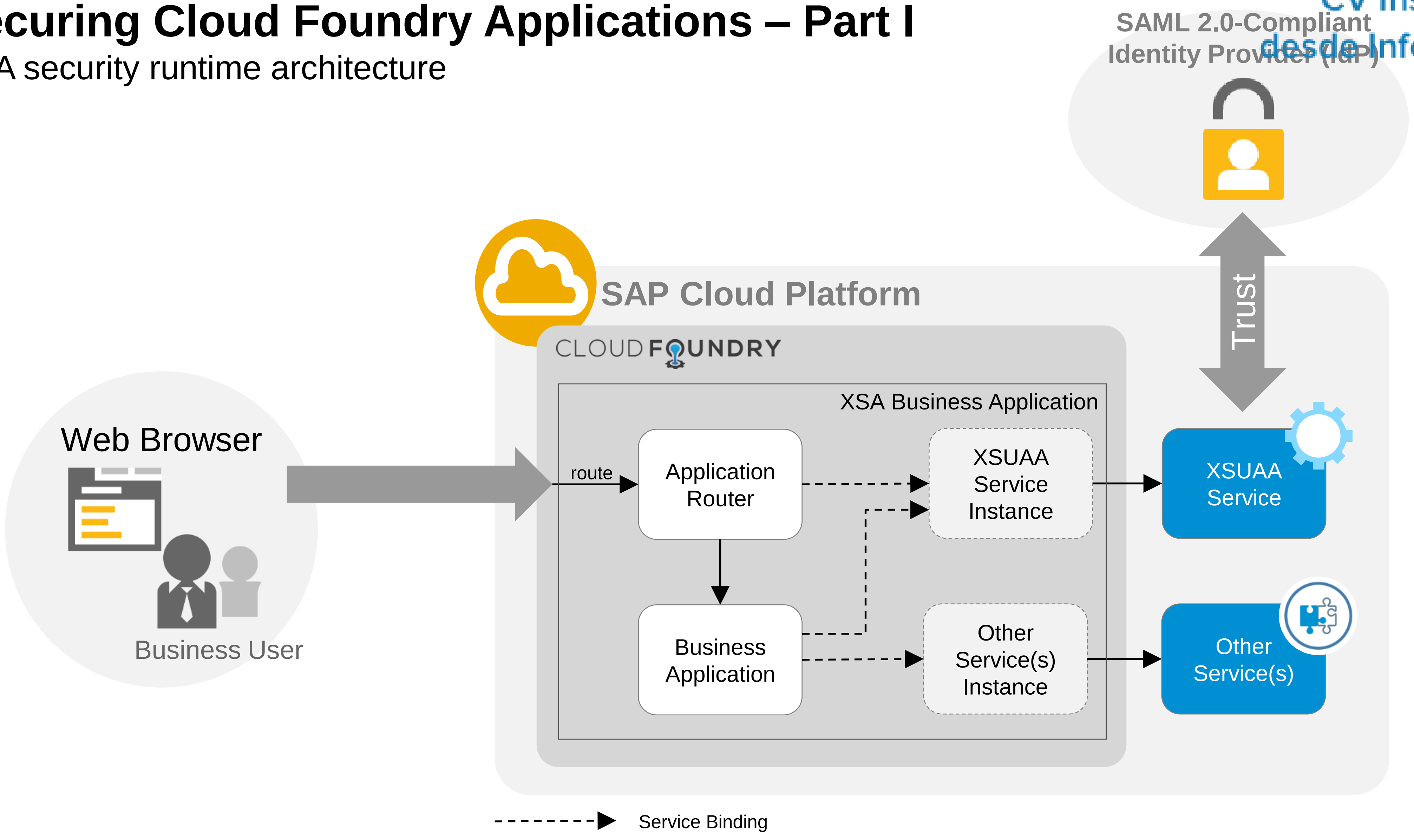
CF Security

Platform Roles & Scopes

Platform User UAA

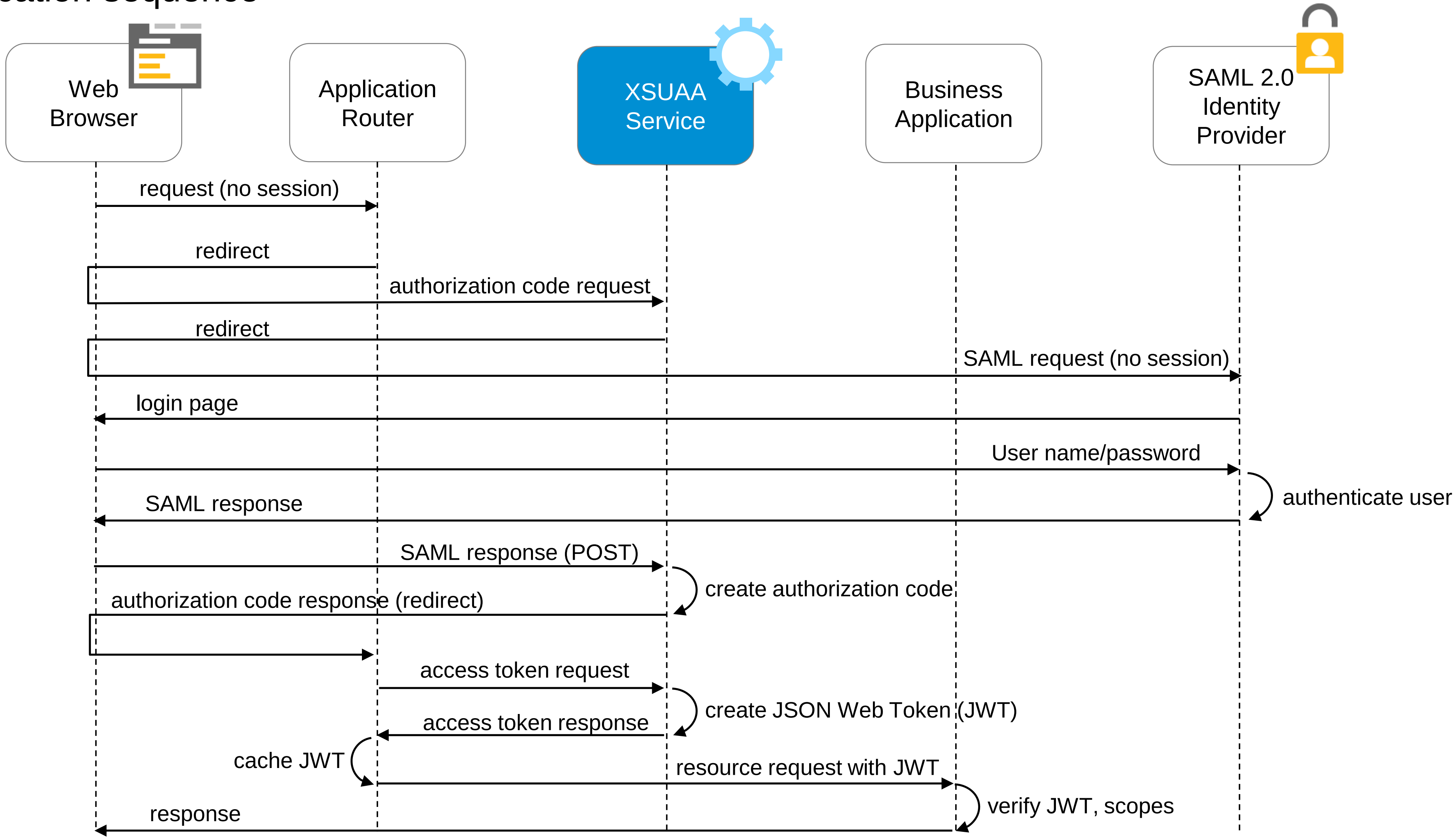
Securing Cloud Foundry Applications – Part I

XSA security runtime architecture



Securing Cloud Foundry Applications – Part I

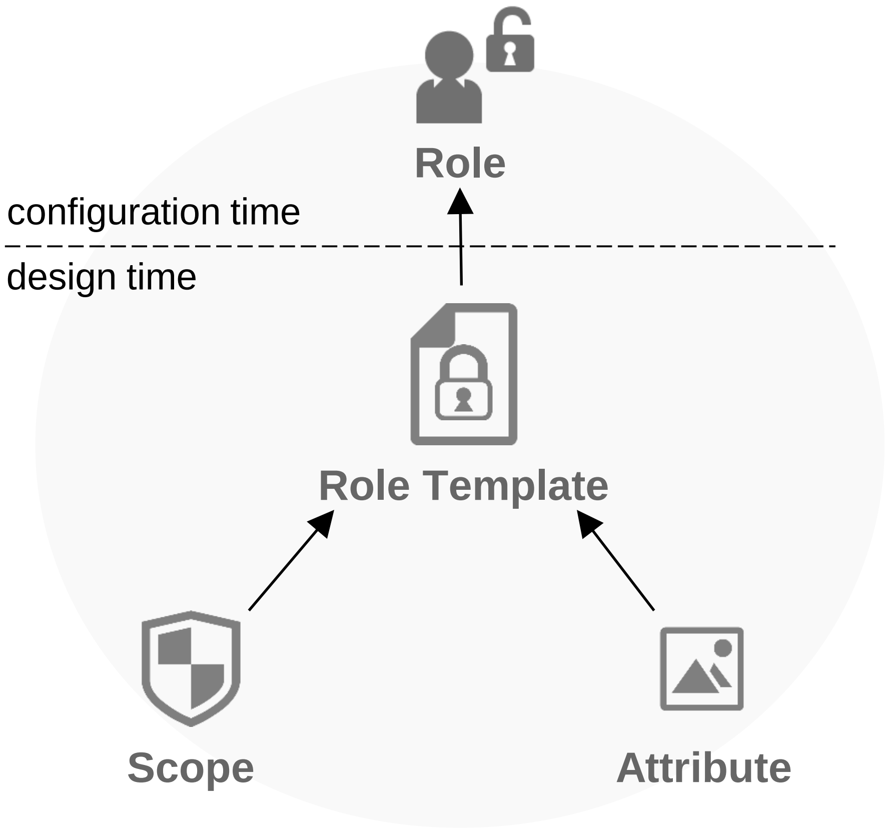
XSA authentication sequence



Securing Cloud Foundry Applications – Part I

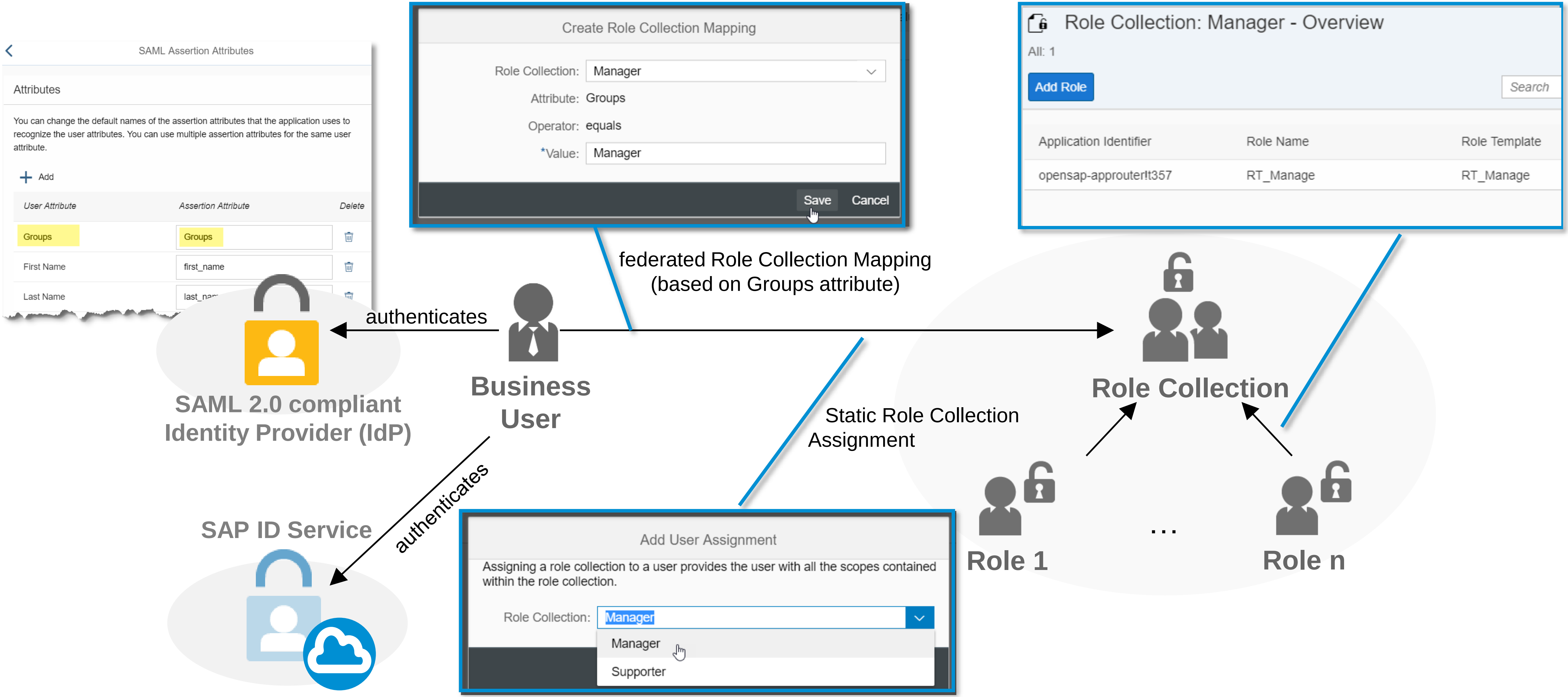
XSA authorization model for business applications – Design and configuration-time artifacts

- **Scope**: For functional authorization checks
- **Attribute**: For instance-based (data) authorizations (e.g. the name of a cost center)
- **Role template**: Description of roles (for example, “employee” or “manager”) to apply to a user and any attributes that apply to the roles
- **Roles**: Are created based on role templates at configuration time in the SAP Cloud Platform cockpit



Securing Cloud Foundry Applications – Part I

XSA authorization model for business applications – Federated role assignment at runtime via role collection



SAML Assertion Attributes

Attributes

You can change the default names of the assertion attributes that the application uses to recognize the user attributes. You can use multiple assertion attributes for the same user attribute.

+ Add

User Attribute	Assertion Attribute	Delete
Groups	Groups	
First Name	first_name	
Last Name	last_nam	

Create Role Collection Mapping

Role Collection: Manager

Attribute: Groups

Operator: equals

*Value: Manager

Save Cancel

Role Collection: Manager - Overview

All: 1

Add Role

Search

Application Identifier	Role Name	Role Template
opensap-approuter!t357	RT_Manage	RT_Manage

Add User Assignment

Assigning a role collection to a user provides the user with all the scopes contained within the role collection.

Role Collection: Manager

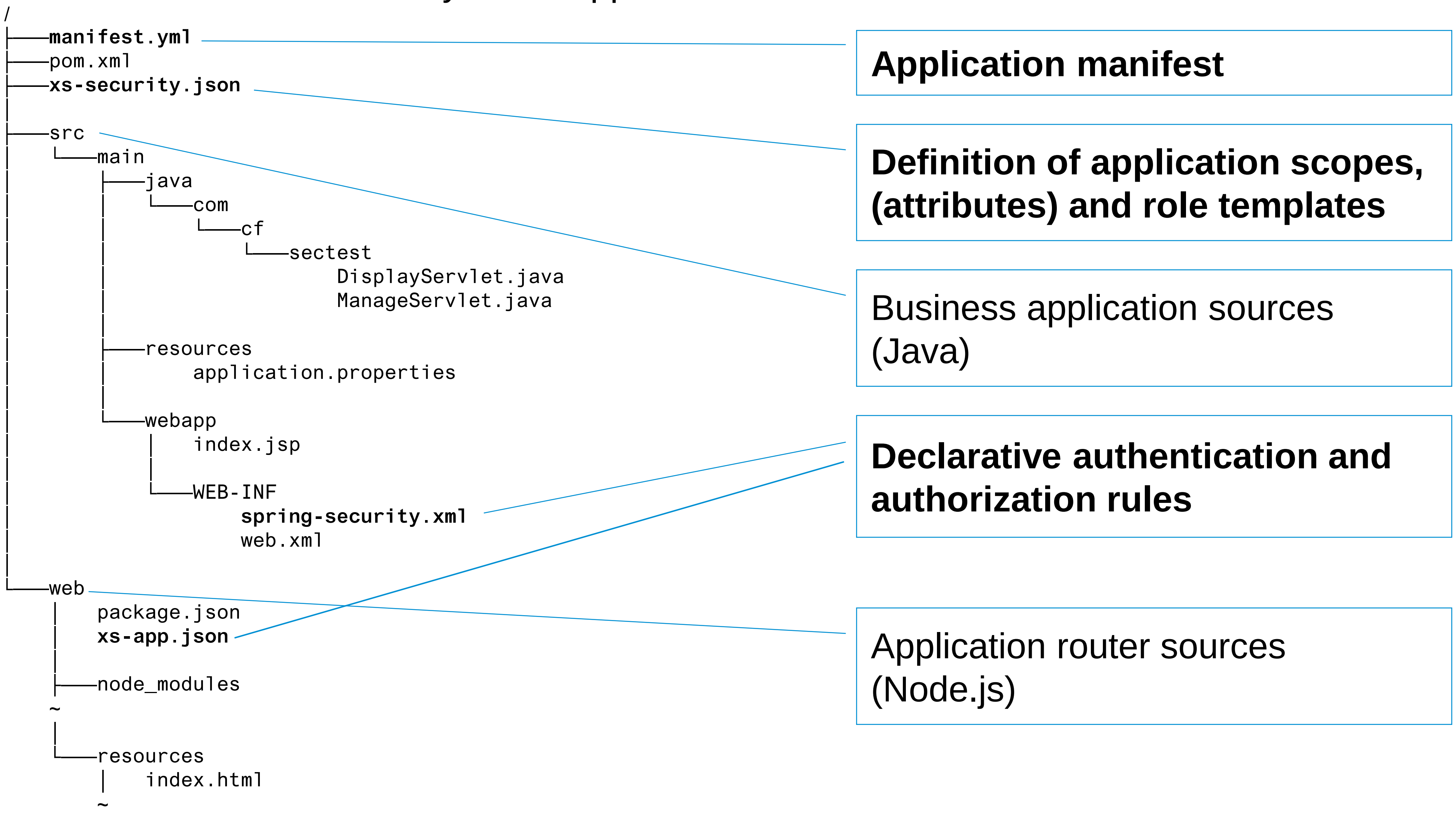
Manager

Supporter

authenticates
Business User
SAML 2.0 compliant Identity Provider (IdP)
federated Role Collection Mapping (based on Groups attribute)
Role Collection
Static Role Collection Assignment
authenticates
SAP ID Service
Role 1
…
Role n

Securing Cloud Foundry Applications – Part I

Structure of the XSA security demo application



```
/
├───manifest.yml
├───pom.xml
├───xs-security.json
│
├───src
│   └───main
│       ├───java
│       │   └───com
│       │       └───cf
│       │           └───sectest
│       │                   DisplayServlet.java
│       │                   ManageServlet.java
│       │
│       ├───resources
│       │       application.properties
│       │
│       └───webapp
│           │   index.jsp
│           │
│           └───WEB-INF
│                   spring-security.xml
│                   web.xml
│
└───web
    │   package.json
    │   xs-app.json
    │
    ├───node_modules
    ~
    │
    └───resources
        │   index.html
        ~
```

Application manifest

Definition of application scopes, (attributes) and role templates

Business application sources (Java)

Declarative authentication and authorization rules

Application router sources (Node.js)

Securing Cloud Foundry Applications – Part I

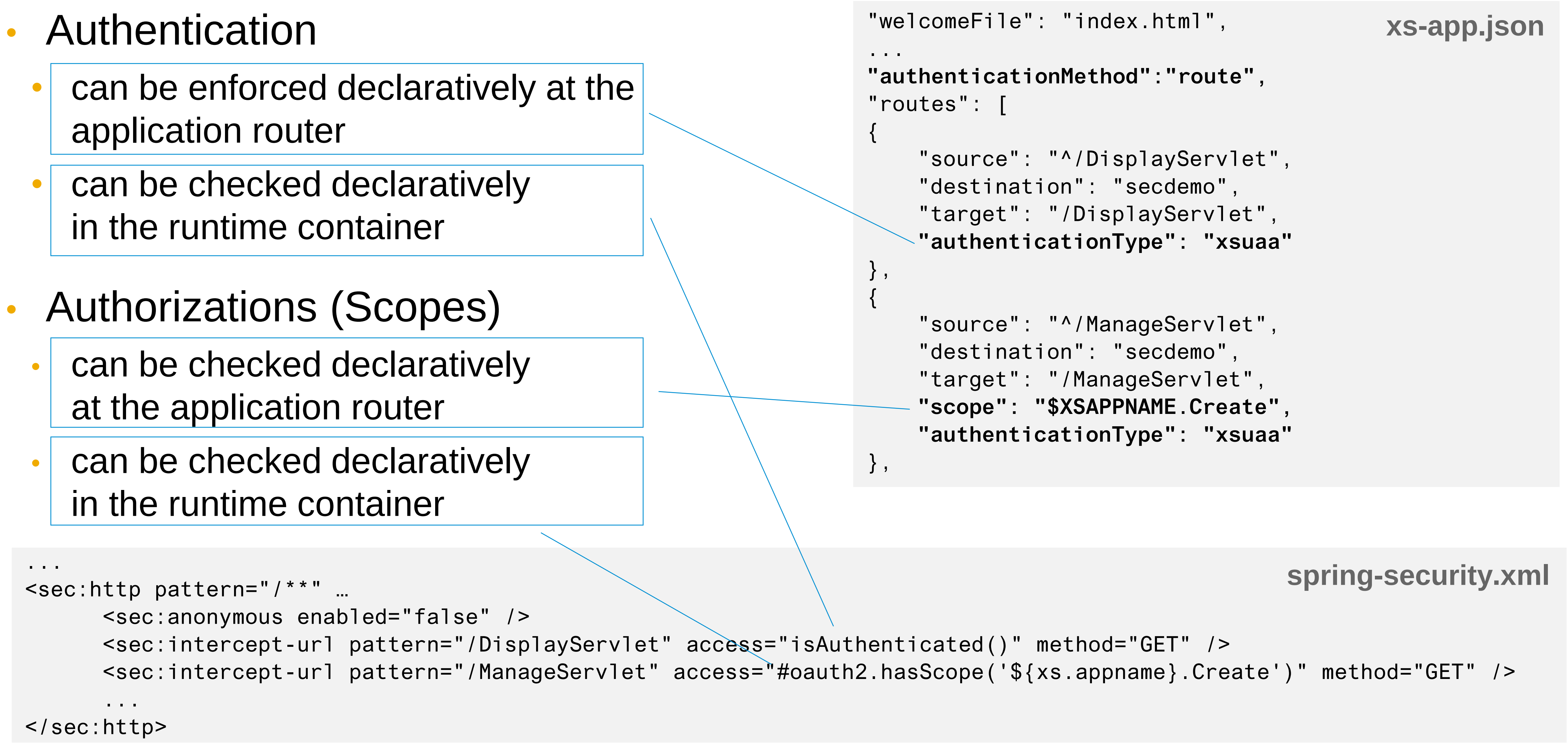
Definition of application scopes, (attributes) and role templates

xs-security.json

```
"xsappname"     : "opensap-approuter",
...
"scopes"        : [ {
                      "name"            : "$XSAPPNAME.Display",
                      "description"     : "display"
                    },
                    {
                      "name"            : "$XSAPPNAME.Create",
                      "description"     : "create"
                    },
                    {
                      "name"            : "$XSAPPNAME.Delete",
                      "description"     : "delete"
                    } ],
"role-templates": [ {
                      "name"            : "RT_Manage",
                      "description"     : “Manage things",
                      "scope-references": [ "$XSAPPNAME.Create", "$XSAPPNAME.Delete" ]
                    },
                    {
                      "name"            : "RT_Display",
                      "description"     : "View things",
                      "scope-references": [ "$XSAPPNAME.Display“ ]
                    } ]
```

Securing Cloud Foundry Applications – Part I

Declarative authentication and authorization



- Authentication
  - can be enforced declaratively at the application router
  - can be checked declaratively in the runtime container
- Authorizations (Scopes)
  - can be checked declaratively at the application router
  - can be checked declaratively in the runtime container

xs-app.json

```
"welcomeFile": "index.html",
...
"authenticationMethod":"route",
"routes": [
{
    "source": "^/DisplayServlet",
    "destination": "secdemo",
    "target": "/DisplayServlet",
    "authenticationType": "xsuaa"
},
{
    "source": "^/ManageServlet",
    "destination": "secdemo",
    "target": "/ManageServlet",
    "scope": "$XSAPPNAME.Create",
    "authenticationType": "xsuaa"
},
```

spring-security.xml

```
...
<sec:http pattern="/**" …
      <sec:anonymous enabled="false" />
      <sec:intercept-url pattern="/DisplayServlet" access="isAuthenticated()" method="GET" />
      <sec:intercept-url pattern="/ManageServlet" access="#oauth2.hasScope('${xs.appname}.Create')" method="GET" />
      ...
</sec:http>
```

Securing Cloud Foundry Applications – Part I

Programmatic authorization with the XSA Security API (Java, Node.js)

- Business applications receive an HTTP header *Authorization: Bearer <JWT token>* from the application router
- The JSON Web Token (JWT) issued by the XSUAA contains the business user and scope information
- It MUST be validated using the **XSA Security API**. Applications can use the API to check if scope values have been assigned to the user/application
- The API must be downloaded from Service Marketplace* (XS_JAVA_4-70001362.ZIP**) and installed in your local Maven repository with `mvn clean install`

```
import
  com.sap.xs2.security.container.SecurityContext;
import
  com.sap.xs2.security.container.UserInfo;
...
UserInfo userInfo =
  SecurityContext.getUserInfo();
String name = userInfo.getLogonName();
String email = userInfo.getEmail();
String[] attribute =
  userInfo.getAttribute(“costcenter");
boolean hasDeleteScope =
  userInfo.hasLocalScope(“Create");
...
```

*https://launchpad.support.sap.com/#/softwarecenter/template/products/%20_APP=00200682500000001943&_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&NE=NAVIGATE&ENR=73555000100200004333&V=MAINT&TA=ACTUAL&PAGE=SEARCH/XS%20JAVA%201

** version/filename may change

Securing Cloud Foundry Applications – Part I

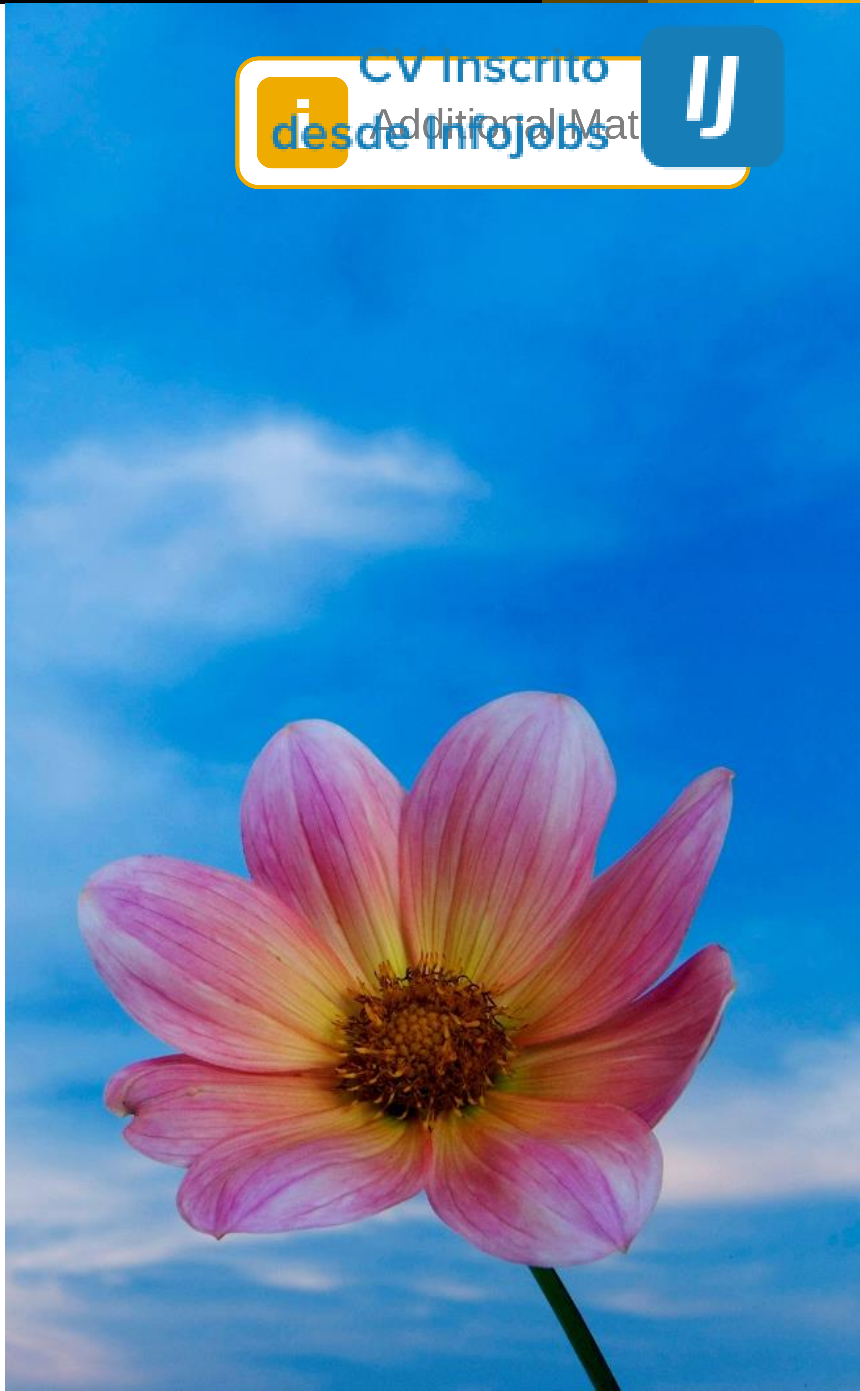
What you've learned in this unit

- The scope of Cloud Foundry's native security model
- The motivation and scope for the XSA security model
- The key components and basic structure of XSA applications
- The integration of these components when the user authenticates
- The key elements of the XSA authorization model and how they are defined by the application developer
- How to check a user's authorizations declaratively in the app router and in the business application
- How to use the XSA Security API in your business application

Securing Cloud Foundry Applications – Part I

Further reading

- Configure Authentication and Authorization: https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/53671c1034d44c83b90b104904d9fb07.html



Thank you.

Contact information:

open@sap.com

© 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE’s or its affiliated companies’ strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
See http://global.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.