TABLE OF CONTENTS

1  INITIAL TENANT SETUP .................................................................................................................. 3
   1.1  Configure Trust .......................................................................................................................... 3
   1.1.1  Technical Background ............................................................................................................ 6
   1.2  Establish Trust ............................................................................................................................. 6
   1.3  Set Up Destinations for SAP Cloud Platform .............................................................................. 12

2  AUTHORIZATION IN SAP IOT APPLICATION ENABLEMENT .................................................... 13
   2.1  Functional Authorization .......................................................................................................... 13
   2.1.1  Example: Configuration User Administrator .............................................................................. 16
   2.1.2  Example: Configuration Thing Engineer ..................................................................................... 16
   2.2  Instance Authorization .............................................................................................................. 17

3  CLOUD FOUNDRY USER ACCOUNT AND AUTHENTICATION SERVER ....................................... 18
   3.1  User Authentication .................................................................................................................... 19
   3.1.1  Application Authentication .................................................................................................... 19

4  SAP CLOUD PLATFORM IDENTITY AUTHENTICATION SERVICE ............................................... 19
   4.1  User ............................................................................................................................................ 19
   4.2  User Groups ............................................................................................................................... 19

5  USER PERSPECTIVES ON AUTHORIZATION ............................................................................. 19
   5.1  Adding a New Object Authorization ............................................................................................ 20
   5.2  Adding a Custom Functional Authorization ................................................................................... 23
   5.3  Result of the Authorization ......................................................................................................... 25

6  AUTHORIZATION ON API CALLS ................................................................................................ 27
   6.1  For Testing Purposes (with User Context) .................................................................................. 27
   6.2  Calling an API from an App Deployed in Cloud Foundry (with User Context) ........................... 27
   6.3  Calling an API from SAP Cloud Platform – Neo-Stack (with User Context) ................................ 27
   6.4  Calling an API from Another Cloud or App (Without User Context) ......................................... 27
This guide explains how to set up authorization and authentication of the SAP® IoT Application Enablement toolkit to meet your business needs with focus on the software-as-a-service (SaaS) use case. The guide assumes you are familiar with the basic concepts of SAP IoT Application Enablement from the solution documentation.

The guide starts with setting up and configuring the toolkit. Section 2 introduces the authorization concept used in SAP IoT Application Enablement. Section 3 discusses setting up authorization in the Cloud Foundry environment of SAP Cloud Platform. Section 4 focuses on the SAP Cloud Platform Identity Provisioning service. Section 5 provides an example, and the last section lists the API calls that can be used in a custom-developed application.

1 Initial Tenant Setup

The initial setup for the tenant is discussed in this section. Before we begin, here are some links that could be very useful to you. Please note that the links contain placeholders that can be customized per individual user or customer.

- SAP Cloud Platform Identity Provisioning service: https://<SCI-Tenant>.accounts400.ondemand.com/admin
- Trust configuration for SAP Cloud Platform: https://account.hana.ondemand.com/cockpit#/acc/<SAP CP ACCOUNT>/trust

1.1 Configure Trust

This section explains how to configure trust between the Neo environment of SAP Cloud Platform and the Cloud Foundry environment of SAP Cloud Platform. The work is performed using the user account and authentication (UAA) configuration tool for SAML2 within SAP IoT Application Enablement.

In the OAuth2 SAML bearer flow, the account in SAP Cloud Platform acts as an identity provider (IDP). The destination service for SAP Cloud Platform issues an SAML assertion for the logged-in user and assigns it as IDP using the private key of the account. For this to succeed, Cloud Foundry must be configured to trust this IDP.

1. To use the default IDP, set the trust configuration as shown below (see Figure 1). Save your entries.

![Figure 1: Configuring Trusted Identity Provider](image)
2. Switch to Custom by selecting the configuration type. Then click Generate Key Pair. The following window appears (see Figure 2).

![Trust Management](image)

**Figure 2: Managing Local Provider Settings**

3. From the SAP Cloud Platform cockpit, export the metadata from the Trust tab of the account on SAP Cloud Platform. To do so:
   - Rename the local provider to a name less than 20 characters long using only letters and numbers.
   - Set the Principal Propagation field to Enabled.
   - Export it.

4. To enable an import to an IDP, you must convert the metadata from SP to IDP metadata. To do this:
   - Replace all "SPSSO" strings with "IDPSSO".
   - Insert an XML declaration at the top "<?xml version=1.0"?>"; the Cloud Foundry import tool requires this.

5. If you were using Default Trust, you need to restore it, but only partially. To do so:
   - Switch Configuration Type to Default.
   - Make sure Principal Propagation is set to Enabled.
   - Make sure Force Authentication is set to Disabled.
   - Click Save.
You still have trust to the SAP ID service, but when you send outbound SAML assertions, the configured local provider will be used to sign them (see Figure 3).

![Figure 3: Configuring the Local Service Provider](image)

6. Go to the UAA configuration tool for SAML2 using:
   Administrative access is required (see Figure 4).

7. Create a new SAML identity provider, paste the metadata, and parse it. Set the state field to Inactive, and save the configuration.

Note: Setting an additional IDP results in a confusing login screen for Cloud Foundry. It is advisable to enable only an additional IDP during testing and to give it a name that indicates it is not the central IDP. (You can specify the name in the UAA configuration tool within SAP IoT Application Enablement.)

![Figure 4: XS Advanced Error Message](image)
1.1.1 Technical Background

To summarize, the configured IDP authenticates user X for the account on SAP Cloud Platform as userX@idpNeoAccount and calls the application running in SAP Cloud Platform.

The app looks up an OAuth2SAMLBearer destination and makes a call through it. The destination assembles an SAML assertion using the local IDP configuration and sends it to the UAA service running in the Cloud Foundry landscape through the OAuth2 SAML bearer assertion flow. The UAA service must have been previously configured to trust the local IDP from the account on SAP Cloud Platform. It then responds with a Java Web toolkit access token that contains the authorized scopes for that user. The destination service calls the configured URL in the destination, providing the access token. This must be the URL to the app router.

Upon receiving the access token, the app router validates it using the internally stored validation key. If the app router accepts it, it then checks to see if the scopes configured for a given route match the scopes from the assertion. Finally, it forwards the call to the back-end app as configured and delegates the access token. When the back-end app receives a call, it uses filters configured in the Spring Security framework from SpringSource Inc., where the token is checked again. If it matches the configuration, it lets the call pass through and calls the app.

1.2 Establish Trust

In this phase of the work, you establish trust between the UAA component of SAP IoT Application Enablement and the SAP Cloud Platform Identity Authentication service. Before you begin, make sure you have the administrator privileges required to perform SAML configuration tasks in the relevant subaccount. They are XS_AUTHORIZATION_ADMIN, full access, no access restrictions.

In the following steps, you assign the identity provider’s metadata file and define attribute mappings. The attributes are included in the SAML 2.0 assertion and are used to assign UAA authorizations automatically based on information maintained in the identity provider.

2. Add a new SAML 2.0 identity provider. To do so, click + Add in the Applications panel of Applications & Resources to create a new application (see Figure 5).

![Figure 5: Adding a New SAML 2.0 Identity Provider](image)

3. Specify a name for the application that identifies it as your new identity provider. Save your changes (see Figure 6).

![Figure 6: Adding an Application](image)
4. Choose SAML 2.0 Configuration and import the relevant metadata file. To do so, use the metadata file of your subaccount. The subdomain name is usually identical to the tenant name (see Figure 7). You can find the metadata file at https://<tenant_name>.authentication.eu10.hana.ondemand.com/saml/metadata.

![Figure 7: Importing the Metadata File](image)
5. Choose *Name ID Attribute* (see Figure 8), select *E-Mail* as the unique attribute, and click *Save* (see Figure 9).

**Figure 8: Choosing the Name ID Attribute**

**Figure 9: Selecting E-Mail As the Unique Attribute**
6. Choose SAML Assertion Attributes (see Figures 10 and 11) and enter Groups (capitalized) in the Assertion Attribute field (see Figure 12). Save your changes.

Figure 10: Choosing SAML Assertion Attributes

Figure 11: Choosing Groups As an Assertion Attribute
Figure 12: Entering Groups in the Assertion Attributes Field

7. Test the SAML 2.0 configuration using the following URL:

You should see information output similar to the following:
- **userName** john.doe@acme.com
- **origin** MySAML2IDP
- **zoneId** acme-saas
- **SAML Issuer** xs2security.accounts.ondemand.com
- currently resolved authorities [idps.write, xs_authorization.write, xs_authorization.read, idps.read]
- **SAML groups** [acme-saas-useradmin]
- **email** john.doe@acme.com
- **userId** 4e0efa4e-5e2f-3a1a-b070-69d7be1acdfb
- **externalId** john.doe@acme.com
- **authorities** [idps.write, xs_authorization.write, xs_authorization.read, idps.read]
1.3 Set Up Destinations for SAP Cloud Platform

After the identity tenant is configured, the tenant administrator for the SAP Cloud Platform Identity Authentication service receives an e-mail with login credentials. With the help of these credentials, the administrator must complete the configuration using the SAP Web IDE development environment. To do so:

1. Go to http://help.sap.com. At the Web site, search for SAP IoT Application Enablement and navigate to:
   → SAP IoT Application Enablement Reuse Controls and Templates
      ○ IoT Application Projects in SAP Web IDE
         • Developing IoT Applications Using the Storyboard Perspective
         • Prerequisites for Creating Applications Using IoT Reuse Controls

2. Follow the instructions to configure the following destinations:
   - IOTAS
   - IOTAS-COMPOSITE-EVENTS-ODATA
   - IOTAS-DETAILS-THING-ODATA
   - IOTAS-FILEIMAGE
   - IOTAS-ADVANCEDLIST-THING-ODATA
   - IOTAS_CONTROLS
   - sapui52

You now have a working setup of SAP IoT Application Enablement, the SAP Cloud Platform Internet of Things service, and SAP Web IDE.
2 Authorization in SAP IoT Application Enablement

The authorization concept of the SAP IoT Application Enablement toolkit includes the following components:

- **The SAP Cloud Platform Identity Authentication service** serves as an identity provider for creating users and user groups. For each customer (or tenant) using the platform services, there is a dedicated identity authentication instance. (See Section 4 for details.)

- **The user account and authentication service** from SAP is used as an authorization server. For each tenant, a dedicated UAA instance (or identity zone) is provided. SAP UAA assigns and checks functional authorizations (or UAA scopes). Functional authorizations determine whether a user has permission to start a given service. The authorizations are assigned to roles provided by the UAA service. For easier administration, the roles are bundled into UAA role collections, which in turn are assigned to user groups provided by the identity authentication service. (See Section 3 for details.)

- **IoT business partner services** are onboarding services that let you associate an employee of a platform customer with a user of the authentication service and to assign the employee to a user group of the service.

- **IoT authorization services** let an administrator define access rights for individual object instances. This authorization is used to filter access to single instances of application-specific objects managed in the SAP IoT Application Enablement toolkit. A prerequisite for calling application services is assignment of the functional authorizations to the user groups within the UAA role collections. This concept establishes a two-step process for checking whether a user has sufficient authorizations for a particular service request. It checks to see if the user is authorized for the desired activity (create, read, update, delete). If functional authorization is sufficient, it checks to see if the user is authorized to perform the desired activity for the given object.

2.1 Functional Authorization

Figure 13 shows the dependencies of functional authorization. Functional authorization is handled by the application router of SAP IoT Application Enablement. Each application of the toolkit, for example, the thing modeler or person app, comes with predefined scopes. Scopes cover specific business scenarios and cannot be changed. Each scope is assigned to a role template, which is mapped to a user group within the Cloud Foundry UAA. (See Section 3 for details on UAA.)

![Figure 13: Dependencies of Functional Authorization](image-url)
You can review the existing scopes within the tenant-specific UAA (see Figure 14). Currently, the following scopes are available.

![Figure 14: Reviewing Scopes Within the Tenant-Specific User Account and Authorization](image1)

Each application in the user interface (UI) has its own scope. The next step is the preconfigured role templates (see Figure 15).

![Figure 15: Viewing Preconfigured Role Templates](image2)
An individual scope is enhanced by attributes, which allow a certain action to be performed within that scope, for example, read, create, update, or delete. These attributes are shown as scope references (see Figure 16).

![Figure 16: Viewing Scope References](image)

Each role is assigned a role template. Roles are cumulative within individual role collections. The collections are accessible through the UAA service. The role collection must be mapped to a user group within the identity provider. This is performed with the SAP Cloud Platform Identity Authentication service using assertions (see Figure 17).

![Figure 17: Viewing Assertion-Based Role Collections](image)
2.1.1 Example: Configuration User Administrator

Figure 18 shows the role collection `useradmin` and its assigned roles `Location_Editor` and `Tenant_Administrator`.

![Role Collection useradmin](image)

To determine the individual scopes provided by the role collection `useradmin`, you must review the assigned role (Tenant_Administrator) and the template used (tenant_administrator), as well as the individual scopes. This reveals that a user assigned to the useradmin role collection is granted the following scopes:

```
auth!t5.c,auth!t5.d,auth!t5.oga.c,auth!t5.oga.d,auth!t5.oga.r,auth!t5.oga.u,auth!t5.r,auth!t5.u,bpl!t5.c,bpl!t5.d,bpl!t5.r,bpl!t5.tenant.r,bpl!t5.tenant.u,bpl!t5.u,bpl!t5.vh.r,iotas!t5.Company.Admin,iotas!t5.ObjectAuthorization.Admin,iotas!t5.Person.Admin,iotas!t5.Usergroup.Admin,tenant!t5.org.c,tenant!t5.org.d,tenant!t5.org.r,tenant!t5.org.u,tenant!t5.pers.c,tenant!t5.pers.d,tenant!t5.pers.r,tenant!t5.pers.u.
```

The user can perform every action within the person, companies, user groups, and object authorization applications.

2.1.2 Example: Configuration Thing Engineer

The preconfigured role collection `thingsuperuser` is assigned these roles:

- Location_Editor
- ReuseUI_Viewer
- Thing_Engineer

By way of an example, a Thing Engineer would be able to perform the following operations because all associated scopes are inherited by the role collection `Thing_Engineer`, which itself contains different roles.

```
loc!t5.c,loc!t5.d,loc!t5.r,loc!t5.u
advis!t5.r,conf!t5.r,cod!t5.r,cod!t5.c,cod!t5.d,cod!t5.r,cod!t5.u,cod!t5.thingconf!t5.r,cod!t5.thingconf!t5.d,cod!t5.thingconf!t5.u,cod!t5.thing!t5.c,cod!t5.thing!t5.d,cod!t5.thing!t5.r,cod!t5.thing!t5.u
```

A user assigned as Thing Engineer would be able to work with the thing modeler application, create locations, add locations to things, and use the Reuse UIs provided through SAP Web IDE.

For more information on functional authorization, go to the solution documentation for the SAP IoT Application Enablement toolkit and select Authorization > Functional Authorization.

2.2 Instance Authorization

The instance authorization functionality is used to assign access rights to specific objects of a certain type that are managed within SAP IoT Application Enablement. Access rights are generally limited to objects within a particular tenant, which ensures that a user can access only the objects that belong to the same tenant to which he or she is assigned.

An administrator uses the IoT authorization functionality to grant access authorizations on an object instance level to user groups within SAP Cloud Platform Identity Authentication. Instance authorizations are based on a hierarchy of authorization groups (see Figure 19). These authorization groups are represented by the ObjectGroup services offered by the platform. The system administrator grants permission to read, write, or delete objects of a given type to a user group. With that, users are able to access object instances of that type.

Figure 19: Object Group Hierarchy

To determine instance authorization, the starting point is Capability Types. These are associated with Applications. Capability type objects define a certain capability that can be granted to a user through specific objects. The relation between a capability type and a capability is similar to that between a class and an object.
The capability types currently delivered are **Thing**, **Organization**, and **Person**. Actions that can be assigned include **Read**, **Write**, and **Delete**. The capability types are instantiated to a capability, which is later associated with a user group. You can use **Capability** objects to specify a set of objects in the database on which users can perform actions defined by the capability type from which the capability has been derived.

The most powerful feature defines a set of objects for which access is granted. A complex filter expression can be defined, which restricts access to objects of a particular type as well as to objects of a particular type with a particular value in one of its fields (or with a set of values in a number of its fields). This lets you define access rights on a highly detailed level. It in turn serves as a prerequisite for a clear segregation of duties, which is required to fulfill the data protection requirements for a multitenancy platform.

Each individual object is assigned to an object group associated with a capability.

### 3 Cloud Foundry User Account and Authentication Server

The primary role of the UAA is as an OAuth2 provider, issuing tokens for client apps to use when they act on behalf of Cloud Foundry users. In collaboration with the login server, the UAA can authenticate users with their Cloud Foundry credentials and can act as a single-sign-on (SSO) service using those, or other, credentials. The UAA has end points for managing user accounts and for registering OAuth2 clients, as well as various other management functions.

The Cloud Foundry UAA service holds all application-related security entities and relations to user information. It stores the association between identity user group and role collection as well as the association between role collection and role (see Figure 20). These associations are separated by an identity zone, which allows individual per-tenant assignments within SAP IoT Application Enablement.

![Figure 20: The Cloud Foundry UAA Server](image-url)
3.1 User Authentication
When a user calls an API of SAP IoT Application Enablement, for example, to create a package within the package app, each HTTP request is sent to an application router. The application router:

- Checks the functional authorization of the user by identifying the user at the identity provider
- Checks the assigned user groups, role collections, and role with the Cloud Foundry UAA server to make sure they correspond to the application the user is calling
- Forwards the request to the application itself

3.1.1 Application Authentication
For developing your application, refer to Section 6, which covers authentication for API calls.

4 SAP Cloud Platform Identity Authentication Service
The SAP Cloud Platform Identity Authentication service is a cloud service that provides services for authentication, single sign-on, and user management in on-premise and cloud solutions from SAP. The service includes on-premise integration as well as user self-services, such as registration and password reset for employees, customer partners, and consumers. The identity authentication service provides security features for protecting access to applications. It covers the definition of risk-based authentication rules and two-factor authentication. It can delegate authentication to on-premise user stores and other identity providers.

4.1 User
When SAP IoT Application Enablement manages users who interact with the IoT platform, all user-related actions – including creating, importing, and updating users – should be performed with the toolkit as well. The toolkit invokes APIs within the SAP Cloud Platform Identity Authentication service. If you create users within SAP Cloud Platform Identity Authentication, they can interact with the toolkit but are not visible within the user management applications the toolkit offers.

4.2 User Groups
A user group is a collection of users who have something in common, for example, who work in the same department or have similar tasks in a company. When working with SAP IoT Application Enablement, you should work with user groups within the toolkit’s authorization functionality. The roles you create there are recognized by the SAP Cloud Platform Identity Authentication service. Each user group can be assigned to Capabilities (the authorization instance) and to Role Collection (functional authorization) within the toolkit (see Section 2).

5 User Perspectives on Authorization
This example shows how the authorization concept within SAP IoT Application Enablement works. The company Fresh Air Company wants to restrict access to the Things object by location and business user role.
5.1 Adding a New Object Authorization

1. Create an authorization group within the Object Authorizations functionality (see Figure 21).

![Figure 21: Creating an Authorization Group](image-url)
2. Create a collection (see Figure 22) and assign authorization to the collection (see Figure 23).

Figure 22: Creating a Collection

Figure 23: Assigning Authorization to a Collection
3. Create a capability that restricts access to the *Things* object that belongs to the authorization group created earlier. Restrict user access to read only (see Figure 24). The object authorization is now set up.

![Figure 24: Creating a Capability and Restricting User Access](image)

4. To assign the capability to a user group (see Figure 25) and open the user group app, create a user group and assign the capability to the user group.

![Figure 25: Assigning a Capability to a User Group](image)

Note: If a user belongs to only one user group, you must assign a functional authorization as well.
5.2 Adding a Custom Functional Authorization

1. Navigate to the application role builder functionality within the UAA service of Cloud Foundry (see Figure 26). Create a role collection.

![Figure 26: Navigating to Application Role Building Functionality](image)

2. Assign the preconfigured `ReuseUI_Viewer` role to the role collection (see Figure 27). This ensures that the user works only with an IoT application that was created with an IoT template from SAP Web IDE. This completes the setup of the user account and authentication.

![Figure 27: Assigning a Role to a Role Collection](image)
3. Assign the functional authorization to a user group in the user group app (see Figure 28).

![Figure 28: Assigning Functional Authorization to a User Group](image)

4. Add the user group to a user within the Persons app of SAP IoT Application Enablement (see Figure 29).

![Figure 29: Adding the User Group to a User](image)
5.3 Result of the Authorization

When the fresh_air_company authorization group is assigned to a Things object, only users in the user group created above (see “Adding a New Object Authorization”) are able to access the Things object. See Figure 30.

Figure 30: A Finished Authorization

All users assigned to a parent authorization group also have read access to the Things object. So a user administrator who has functional authorization for the Reuse Viewer role and is assigned to the parent authorization group will be able to see all Things objects in the IoT application within a given package. See Figure 31.

Figure 31: Users Assigned to a Parent Authorization Group
The operator user will see only the *Things* objects in cases where the authorization group was changed to *Fresh Air Company*. See Figure 32.

![Figure 32: Effects for the Operator of an Authorization Group](image-url)
6 Authorization on API Calls

This section provides four scenarios for accessing the APIs of the SAP IoT Application Enablement toolkit.

6.1 For Testing Purposes (with User Context)

The easiest way to access APIs of SAP IoT Application Enablement is to use Postman with the interceptor plugin for the Google Chrome browser. To do so, simply log in to the launch page of SAP IoT Application Enablement using Google Chrome with the activated interceptor. Postman uses the established session to authenticate your requests. Requests that come from a browser always go through the app router (https://<TENANT>.iot-sap.cfapps.eu10.hana.ondemand.com) with a specific route (for example, /appiot-mds or /business-partner) to the microservice required.

6.2 Calling an API from an App Deployed in Cloud Foundry (with User Context)

It is assumed you already have a UI application running on SAP Cloud Platform in the Cloud Foundry environment and want to access the API of SAP IoT Application Enablement. To do so:

- Define the role templates in your xs-security.json for your XSUAA service instance with the scopes from SAP IoT Application Enablement that your application or requests require.
- To ensure SAP IoT Application Enablement trusts your application, specify which scopes are granted to your application.
- If the trust is set up properly and the scopes are defined in your role templates and assigned to the users, you can pass the Java Web token (JWT) created by the OAuth client of your app router.
- For UI consumption, you can add the routes to your app router to the microservices (back-end apps) of SAP IoT Application Enablement (forwarding JWT = true). Or you can send requests from your back-end application to the microservices of SAP IoT Application Enablement. In neither case does the request go over the mentioned URL. You must call the back-end app URL directly, for example, https://appiot-mds.cfapps.eu10.hana.ondemand.com, for Thing services.

6.3 Calling an API from SAP Cloud Platform – Neo-Stack (with User Context)

If you have a Neo application and want to call an API of SAP IoT Application Enablement, you must create trust between your account on SAP Cloud Platform, your identity zone in XSUAA on Cloud Foundry, and your tenant in the SAP Cloud Platform Identity Authentication service. Once the trust is set up, you can create destinations to each microservice in the toolkit you want to use with the authentication type “OAuth2SAMLBearerAssertion.” All credentials and information needed are provided during the onboarding process of your tenant in the toolkit through e-mail. (This includes documentation for the destination setup.)

6.4 Calling an API from Another Cloud or App (Without User Context)

If you have an application running in the cloud and want to push or pull data from the toolkit through a job, there is an OAuth client (iotas_consumer) that supports the OAuth client credential flow. The credentials are sent during the onboarding process of your tenant by e-mail. (See Section 6.3.) As the client credential flow has only scopes (functional authorizations) and no user context, the instance-based authorization will have no effect on requests. The requesting application must determine which users are authorized to see specific instances and which are not.

---

27