opentext™

NetIQ Single Sign-on
Administration Guide

August 2024

Legal Notice

Contents

About This Book 7

1 Welcome to Single Sign-on 9

2 Getting Started 11

How to Log in to the Micro Focus SaaS Environment 11
Configuring Authentication for Micro Focus SaaS 11
Requirements for the Service Applications 12
Requirements for External Identity Provider Applications 13
How Do Users Gain Access to Single Sign-on 13

3 Enable and Manage the Application Portal 15

Enable the Application Portal 15
Configure the Application Portal to Create New User Accounts 16
Understanding How Single Sign-on Groups and Displays the SCIM Attributes 16
Understanding How the Administration Console Groups and Displays the SCIM Attributes 17
Understanding How the Application Portal Groups and Displays the SCIM Attributes 18
Manage the Application Portal 19

4 Configuring Account Claiming 21

Enable Account Claiming 21

5 Creating Service Applications 23

6 Creating Appmarks 25

Overview of Appmark Categories 25
Create Appmarks 26

7 Creating an OAuth Application 27

Understanding OAuth 2.0 27
OAuth Components 28
OAuth Authorization Grant 29
Example of Single Sign-on with OAuth 32
Create an OAuth Application 34
Select Chains 35
Configure the OAuth Advanced Settings 36
Configure Grant Types 36
Configure Scopes 38
Configure Claims 39
Obtaining an OAuth Token from an Application 40

8 Creating a SAML Application 41

Understanding SAML . . . 41
SAML Components . . . 42
Example of an IdP-Initiated Authentication with Single Sign-on . . . 44
Example of an SP-Initiated Authentication with Single Sign-on . . . 45
Manage a SAML Application . . . 46
Create a SAML Application . . . 47
Select Chains . . . 48
Configure SAML Advanced Settings . . . 49
Obtain the SAML Metadata . . . 50

9 Creating an External Identity Provider 53

Create an External SAML Identity Provider Application . . . 53
Create an Advanced Authentication Chain . . . 54
Create a Service Application . . . 54

10 Configuring Authorization Policies 55

Overview of Authorization Policies . . . 55
Manage Authorization Policies . . . 55
Create Authorization Policies . . . 56
Edit an Authorization Policy . . . 56
Delete an Authorization Policy . . . 56
Create a Rule-Based Authorization Policy . . . 57
Create a Rule-Based Authorization Policy with User Attributes . . . 57
Understanding the Default Attributes for a Rule-Based Authorization Policy . . . 58
Create a Rule-Based Authorization Policy with Identity Governance Roles . . . 60
How the Authorization Policy Using Identity Governance Roles Matches User Accounts . . . 61
Create an OPA Authorization Policy . . . 62
Example of an OPA Policy Document . . . 64

11 Managing Single Sign-on 67

Manage Authentications . . . 67
Manage Branding . . . 67

12 Manage Applications and Appmarks 69

Create an Application or an Appmark . . . 69
Make an Application a Favorite . . . 70
Filter Applications . . . 70
Edit an Application . . . 70
Manage the Client Secret for an OAuth Application . . . 71
View the Status of an Application . . . 71
Change the Application Tile Size . . . 71
Change the Settings for the Most Recent Applications . . . 72
Troubleshooting Issues . . . 72
Troubleshooting the Salesforce Application . . . 72

A Understanding Secure Communications **73**

Understanding the Public Key Infrastructure and TLS Components to Establish Secure Communication 73
Example of Establishing Secure Communication for a Web Server 75
Example of a Secure Handshake for the Client 77

About This Book

The *Administrator Guide* provides conceptual information about NetIQ Single Sign-on. This guide contains descriptions of common standards used to create a secure, single sign-on experience for your end users, and step-by-step guidance for common tasks.

Intended Audience

This book provides information for individuals responsible for planning and maintaining single sign-on and federation configurations between their organization and external services in a cloud software-as-a-service (SaaS) environment. A working knowledge of network operations, network security, and cloud SaaS technologies is assumed. As well as a good knowledge of federation technologies such as OAuth and SAML.

Additional Documentation

For the most recent version of this guide and other Single Sign-on documentation resources, visit the Single Sign-on Documentation website (https://www.microfocus.com/documentation/single-sign-on/help/).

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the **comment on this topic** link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact Micro Focus Customer Care at https://www.microfocus.com/support-and-services/.

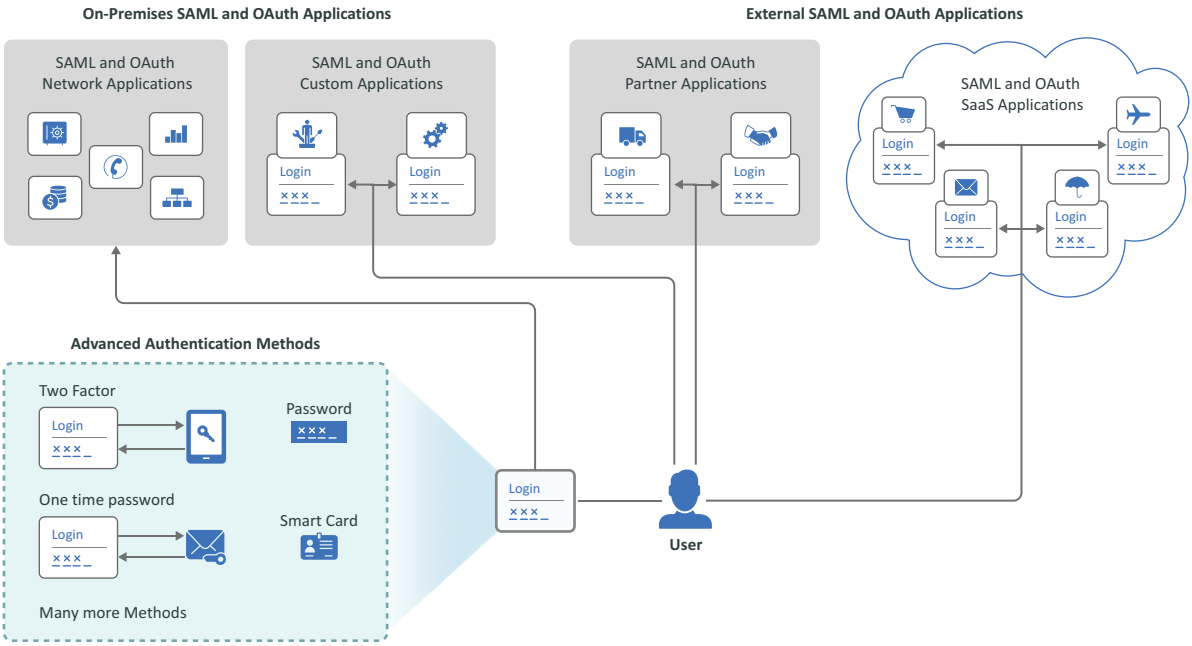
1 Welcome to Single Sign-on

Single Sign-on is a solution to help to reduce the complexity and cost of managing your users' access to services that use industry standards to provide a single sign-on experience for your users. It is a service hosted in the Micro Focus Software as a Service (SaaS) environment and can securely connect to your existing systems.

Single Sign-on provides secure access for the users in your organization to any application, resource, or service that supports single sign-on. Using industry standards such as Security Assertion Markup Language (SAML) and OAuth you create a two-way, trusted connection between Single Sign-on and the external application, resource, or service. The two-way, trusted connection is a **federated connection** that allows the users to have a single sign-on experience with these trusted resources.

As an organization, you spend a lot of time and resources ensuring that your users have access to the applications required to perform their jobs. Without a single sign-on solution, the users face a complex environment to be able to access and use the applications or resources required to complete their jobs. The following figure depicts the log-in experience from a user's point of view.

Figure 1-1 *Log-in Experience for a User without Single Sign-on*



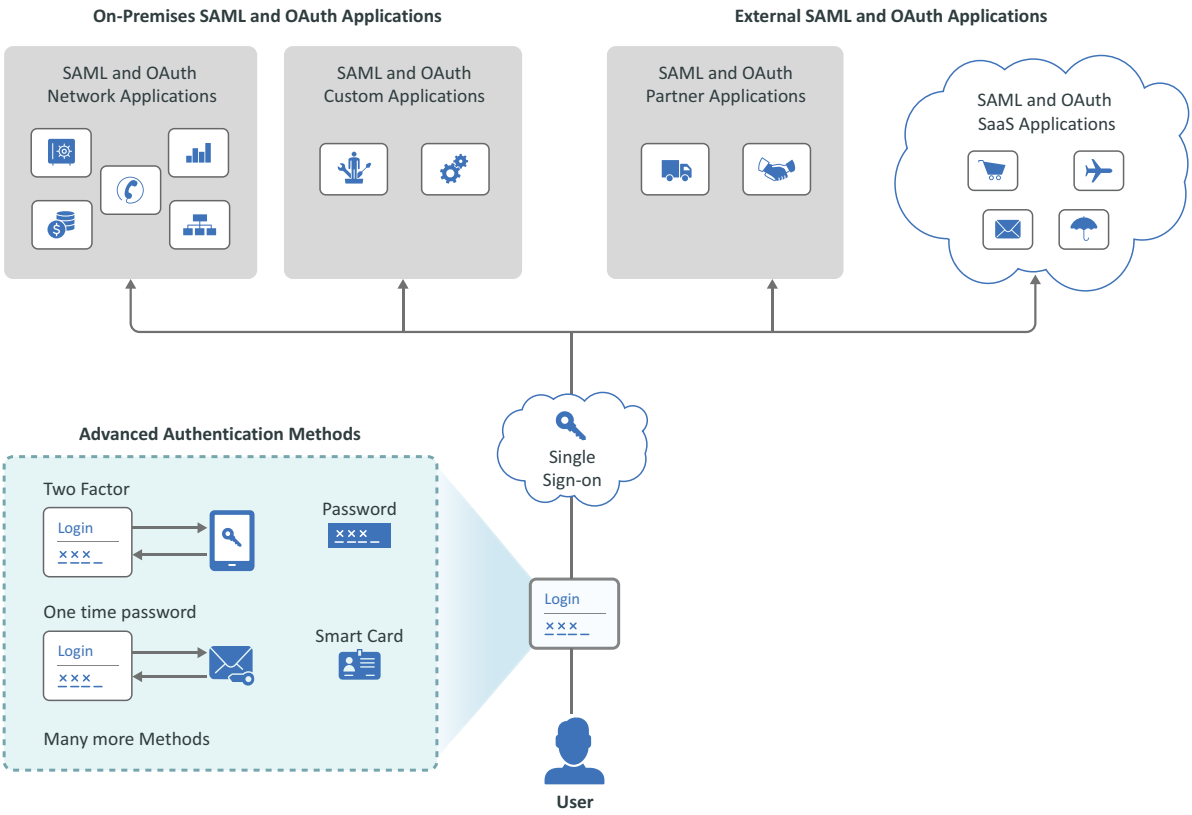
The user must obtain and remember a separate user name and password for each application. This type of experience:

- Causes confusion for the user.
- Generates many support calls because of forgotten passwords and IDs.
- Causes users to write down passwords and IDs to be able to log in or access the applications and resources they need.

- Creates a security issue because it tempts the users to write down their passwords and IDs.
- Requires many hours of managing the user's access and authentication tokens to the different systems.

Single Sign-on provides a solution that allows you to configure the single sign-on experience for your users once per application. It allows to you create a secure, federated connection between your organization's systems and the applications or resource the users need to access. After you have Single Sign-on configured for the different applications or resources, the user's log-in experience is much simpler. The following figure shows this experience from the user's point of view.

Figure 1-2 *Log-in Experience for a User with Single Sign-on*



Single Sign-on simplifies the user's log-in experience and it also reduces the administrative overhead of distributing multiple authentication tokens to the users. Single Sign-on:

- Simplifies the user's log-in experience by providing one password and ID for the different single sign-on enabled applications or resources.
- Reduces the number of support calls because of forgotten passwords and IDs.
- Reduces the temptation of users writing down their passwords and IDs to access the applications and resources that they need.
- Minimizes the security issue of users writing down their passwords and IDs because they need to remember only one password and user ID.
- Reduces the administrative overhead of granting authentication tokens to each user to access each application and resource.

2 Getting Started

Single Sign-on is a service that is hosted in the Micro Focus SaaS environment. You must ensure that the following items have been completed before you can use Single Sign-on.

- "How to Log in to the Micro Focus SaaS Environment" on page 11
- "Configuring Authentication for Micro Focus SaaS" on page 11
- "Requirements for the Service Applications" on page 12
- "Requirements for External Identity Provider Applications" on page 13
- "How Do Users Gain Access to Single Sign-on" on page 13

How to Log in to the Micro Focus SaaS Environment

If your organization has one more services configured in the SaaS environment, contact the administrator of these services for an administrative account to be able to configure Single Sign-on and the URL.

If your company has no services configured, Micro Focus sends an email to the person that placed the purchase order for Single Sign-on. If you are the person that placed the purchase order, look for the email from Micro Focus with your unique URL.

If you are not the person that placed the purchase order but are an administrator, you must ensure that you have created an account for yourself on the Software Licenses and Downloads (https://sld.microfocus.com) portal. After you have an account, you must ask the person that placed the purchase order to delegate access to you for this unique URL and provide the password included in the email for the first time you log in to the Micro Focus SaaS environment. After you configure the authentication service and Single Sign-on, you do not have to use that user account and password.

Configuring Authentication for Micro Focus SaaS

If you have the advanced version of Advanced Authentication, it enables Single Sign-on. You must configure Advanced Authentication for Single Sign-on to function. Configure the minimum following items once in the Advanced Authentication service to be able to configure and use Single Sign-on:

- **Repository:** Configure one or more repositories (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/add_repo.html). The repositories must contain the user account information to provide a single sign-on experience for these users.
- **Methods:** Configure one or more authentication methods (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_methods.html) for the user accounts.

- **Chains:** (Conditional) Advanced Authentication allows the users to have more than one authentication method. Create one or more chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html) if the service you want to provide single sign-on access to requires multiple authentication methods.

NOTE: Step-up authentication occurs when you have multiple chains configured. If a user does not meet the requirements defined in the chain, Single Sign-on prompts the user to provide more information from the additional chains to complete the authentication.

Requirements for the Service Applications

Single Sign-on contains objects named **applications**. The applications can be service applications or identity provider applications. The **service applications** contain all of the required artifacts to create a trusted, two-way connection to external services. A **service** is an application, service, or resource that you want to provide a single sign-on experience for your users. The applications contain certificates, metadata, appmarks, connectors, and any additional information required for the trusted, two-way connection. The trusted, two-way connection is a federated connection. A **federated connection** establishes a trust between Single Sign-on and a service to create a single sign-on experience for the users. You configure both Single Sign-on and the service to create the federated connection.

Each service that you want to provide a single sign-on experience for is different and has different requirements for single sign-on. The standardized authentication protocols like SAML and OAuth give all developers a structure to use but it is not a template. The resulting services have differences.

To create a single sign-on experience you take information from Single Sign-on and add it to the service and you take information from the service and add it to Single Sign-on. After you complete this configuration the two services trust each other to allow the user to have a single sign-on experience. The configuration for the trust is different for each protocol.

The fields can be different for each service with the same protocol because they are developed by different people. For example, when creating an OAuth connection, you must share the client ID and client secret to create a secure connection. The field names for these items in Single Sign-on are **Client ID** and **Client Secret**, however, the field names for these same items in Salesforce are **Consumer Key** and **Consumer Secret**.

You must gather the required items to create an application that contains the required information for a federated connection. You must gather or know the following information:

- **Single Sign-on enabled service:** The service that you want to create a single sign-on experience for your users must be enabled for single sign-on.
- **An administrative account in the service:** You must have an account with administrative privileges in the service to be able to configure a single sign-on connection.
- **An administrative account in Single Sign-on:** You must have an account with administrative privileges in Single Sign-on to be able to configure a single sign-on connection.
- **Protocol of the service:** You must know which protocol (such as SAML or OAuth) the service uses to be able to know how to properly configure the single sign-on trusted connection. Each protocol requires different information to create a trusted connection. The required items for each protocol are documented in the section that describes that protocol.

- **User accounts:** You must have the accounts for the users in the Advanced Authentication repository to provide a single sign-on experience. The external service might require that the user accounts exist in their system as well. Each service has different requirements.

Single Sign-on provides applications for frequently used external services. Each application for the service contains federation instructions that are unique for the service. Single Sign-on also allows you to create a custom application for an external service that we do not provide in the administration console.

Requirements for External Identity Provider Applications

Single Sign-on allows you to create external identity provider applications. An **external identity provider application** represents an external identity provider that Single Sign-on uses to verify the user accounts. By default, Single Sign-on uses the Advanced Authentication repository that you have configured. If you do not want to use the Advanced Authentication repository, you can create an external identity provider application to use to verify the user accounts.

The requirements for creating an external identity provider are:

- Obtain the metadata configuration information from the service that you want to be the identity provider. For example, you can use Google as your external identity provider instead of Advanced Authentication.
- Ensure that Advanced Authentication is configured and running. Single Sign-on automatically creates events for the external identity provider in the Advanced Authentication, then you must create a chain with the proper event selected.

How Do Users Gain Access to Single Sign-on

There multiple different ways to provide access to Single Sign-on. They are:

Users Claim Existing Accounts

If there are an existing accounts for the users in the identity repository (SCIM database, Active Directory, and so forth), you can send the users a specific link that walks them through claiming their account. You configure the Account Claiming process to have users provide additional information to validate the accounts, select the appropriate log in methods, and any other such tasks required to complete their accounts.

Users Create New Accounts

If there are no existing accounts for the users, Single Sign-on has an option on the Application Portal page for the users to create new accounts.

3 Enable and Manage the Application Portal

Single Sign-on provides an Application Portal where your users log in to access the applications, services, and tasks that they must perform for their jobs. Single Sign-on also allows new users register their accounts and configure how they will log in to the Application Portal.

- "Enable the Application Portal" on page 15
- "Configure the Application Portal to Create New User Accounts" on page 16
- "Understanding How Single Sign-on Groups and Displays the SCIM Attributes" on page 16
- "Manage the Application Portal" on page 19

Enable the Application Portal

Applications

By default, Single Sign-on does not enable the Application Portal. The first time that you log in to the administration console and select **Applications**, there is a **System** application available. The System application helps you configure the OAuth Client that Single Sign-on uses to establish secure communication between the Application Portal and the secure authentication system.

To enable the System application for the Application Portal:

1. On the **Applications** page, select **New Application**.
2. Under system applications, select **Application Portal**.
3. By default, Single Sign-on populates some of the data for the OAuth client for the Application Portal. For more information about the options that Single Sign-on populated, see Create an OAuth Application.
4. Use the following options to complete the configuration for your environment:

   ADDITIONAL CONFIGURATIONS

   Under **ADDITIONAL CONFIGURATION**, select the plus sign to configure the user registration option for the users when they first access the Application Portal. Use the information in Configure the Application Portal to Create New User Accounts to configure this option for your users.

   AUTHENTICATION CHAINS

   Select the appropriate authentication chains for your environment.
5. Select **Save** to save and create the Application Portal OAuth client.

Single Sign-on creates the Application Portal OAuth client, that redirects you to the Application Portal.

Configure the Application Portal to Create New User Accounts

Applications > Application Portal > Additional Configuration

Single Sign-on allows your users to provide their details so that it can create the user accounts automatically. The users can provide their accounts details when they access the Application Portal for the first time. There is an option on the Application Portal log in page of **Create New Account**. You define the attributes that the users must provide for Single Sign-on to automatically create the user accounts.

The Application Portal URL is: `https://unique-URL/portal/login`. When a user accesses the Application Portal, there is a **Sign in** and **New Account** options. The second option requires that there is an existing SCIM repository (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/add_repo.html).

IMPORTANT: If you do not have an SCIM repository configured, this feature does not work.

To configure the create new accounts option:

1. Either when you configure the OAuth client for the Application Portal or edit it, under **ADDITIONAL CONFIGURATIONS** select the plus sign.

   IMPORTANT: Currently, Single Sign-on only supports retrieving attributes from one SCIM repository.

   ---

2. Select **Edit** on the SCIM repository that will contain the user accounts.
3. Select the appropriate SCIM attributes to create the new user accounts.

   NOTE: By default, Single Sign-on requires **User Email Addresses** and **User Name**. You cannot remove these attributes.

   ---

4. Select **Done**.
5. Select **Save** to save the policy, then select **Save** to save the changes to the appmark for the Application Portal.

Understanding How Single Sign-on Groups and Displays the SCIM Attributes

Single Sign-on displays the attributes to use when you want to create new user accounts. The Administration Console and the Application Portal display the attributes differently. Use the following information to understand how Single Sign-on displays the attributes to administrators and to the users.

- “Understanding How the Administration Console Groups and Displays the SCIM Attributes” on page 17
- “Understanding How the Application Portal Groups and Displays the SCIM Attributes” on page 18

Understanding How the Administration Console Groups and Displays the SCIM Attributes

Single Sign-on groups attributes together to help make it easier to create the user registration policy. Single Sign-on always requires that you include the **User Email Addresses** and **User Name**. These options are always located at the top of the list of the options to include the policy that you create.

The **Password** option is not required. If you do not include **Password**, then a password authentication auto-enrollment does not happen when Single Sign-on creates the account. If you do include **Password**, the Single Sign-on automatically enrolls the user in the Advanced Authentication **PASSWORD** chain (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html).

The following table lists the group name with the associated SCIM attributes.

Table 3-1 *Group Names and Corresponding SCIM Attributes*

Group Name	Single Attribute Name	Attribute Name Displays below Group Name
User Name	◆ Formatted Complete User Name* ◆ User Family, Last, or Surname* ◆ User Given of First Name* ◆ User Middle Name* ◆ User Honorific Name Prefix* ◆ User Honorific Name Postfix*	◆ Formatted Complete Name ◆ Family Name ◆ First Name ◆ Middle Name ◆ Honorific Prefix ◆ Honorific Postfix
Physical Mailing Address	◆ Formatted Complete Address* ◆ Street Address* ◆ City or Locality* ◆ State, Province, or Region* ◆ Postal Code*	No attributes
User Email Addresses* (REQUIRED)	◆ User Email Address ◆ User Email Type ◆ User Email Address Descriptive Name	◆ Email ◆ Type ◆ Descriptive Name
User Phone Numbers*	◆ User Phone Number ◆ User Phone Number Type ◆ User Phone Number Descriptive Name	◆ Phone Number ◆ Type ◆ Descriptive Name
User Instant Messing Specifiers*	◆ User Instant Messaging Specifier ◆ User Instant Messaging Specifier Type ◆ User Instant Messaging Specifier ◆ Descriptive Name	◆ IMS ◆ Type ◆ Descriptive Name

Group Name	Single Attribute Name	Attribute Name Displays below Group Name
No Group	◆ Password* ◆ User Name (REQUIRED)* ◆ User Display Name* ◆ User Nickname* ◆ User Title* ◆ User Preferred Language for Localization* ◆ User Location for Localization* User Time Zone*	No Group

NOTE: Items marked with an asterisk (*) appear in the Administration Console.

Understanding How the Application Portal Groups and Displays the SCIM Attributes

When the users select the option to create a new account, the Application Portal groups and displays the attributes differently from what is in the Administration Console. The Application Portal groups similar attributes to appear together in the UI, no matter what the administration selects in the Administration Console. The following describes how the Application Portal displays the groups and attributes to the users.

1. USERNAME is require and the most important part of the account. The Application Portal always lists it first.
2. (Conditional) PASSWORD is next, if you selected to require a password for the user.
3. EMAIL attribute is third in the list. It is a group with Email and Descriptive Name as attributes under the group.
4. The Application Portal lists all of the attributes in the User Name group next. If there is only one attribute, the Application Portal displays it as full display name, without the User Name group header. If there are two or more attributes, then the Application Portal displays the User Name group header with all of the children of the group that you selected.
5. Next, the Application Portal adds the Physical Mailing Address group. If there is just one attribute, then the Application Portal displays it with its full display name without the Physical Mailing Address group header. If there are two or more attributes, then the Application Portal displays the Physical Mailing Address group header with the attributes for this group header with all of the children of the group that you selected.
6. Next, the Application Portal displays the attributes in the User Phone Numbers group. If there is just one attribute, then the Application Portal displays it with the full display name without the User Phone Numbers group header. If there are two or more, then the Application Portal displays the User Phone Numbers group header with the attributes for this group header with all of the children of the group that you selected.

7. Next, the Application Portal adds the attributes in the User Instant Messaging Specifier group. If there is just one, then the Application Portal displays it with the full display name without the User Instant Messaging Specifier group header. If there are two or more, then the Application Portal displays the User Instant Messaging Specifier group header with all of the attributes for this group.
8. Lastly, all of the Application Portal displays the Miscellaneous attributes without a group header.

Manage the Application Portal

Single Sign-on provides an Application Portal to simplify access to the services. When you create an appmark, Single Sign-on automatically adds it to the Application Portal. If you modify any of the existing appmarks on the Applications page, Single Sign-on automatically updates the appmarks on the Application Portal. Users can see and select the appmarks but they are not allowed to edit the appmarks.

You must provide your users with the URL for the Application Portal so that they can access the appmarks. You can email it to your users or you can embed it on your organization's main website for easy access. When the users select the link for the Application Portal, Single Sign-on prompts the users to log in using the methods you defined for your users. After the users log in, when they select the appmarks, they are automatically signed in to these external services because of the federation connections that you have configured for your users.

4 Configuring Account Claiming

Single Sign-on provides a way for users to claiming existing accounts in the identity repository (SCIM database, Active Directory, and so forth). You configure the Account Claiming process to have users provide additional information to validate the accounts, select the appropriate log in methods, and any other such tasks required to complete their accounts.

- "Enable Account Claiming" on page 21

Enable Account Claiming

Applications > New Applications

Single Sign-on provides a **System** application you must configure to enable Account Claiming. The System application for Account Claiming allows you to configure a specific URL that the users access to start the Account Claiming process.

The System application for Account Claiming creates an OAuth client to secure communication. There must be accounts for the users in the identity repository (SCIM database, Active Directory, and so forth) for the Account Claiming process to function.

To enable the System application for Account Claiming:

1. On the **Applications** page, select **New Application**.
2. Under the system applications, select **Account Claiming**.
3. Use the following information to configure the Account Claiming application:

   Name and Description

   Single Sign-on populates the name and description of the application for you. The default values are:

   - **Name:** Account Claiming
   - **Description:** Account Claiming OAuth Client

   Client ID Prefix

   Specify a unique name for the prefix of the URL for account claiming that Single Sign-on generates for you. The Account Claiming URl is:

   `https://dns/portal/ssoclaim?type=client id prefix`

   NOTE: After you save the Account Claiming application, you cannot edit the Client ID Prefix. If you need to change the value, you need to delete the Account Claiming application, then configure a new Account Claiming application.

   Client Secret

   Single Sign-on automatically populates the **Client Secret** for the Account Claiming application.

AUTHENTICATION CHAINS

Select an authentication chain, then select **Select Chains**. The **Authentication Chain** field is empty when you first create the application.

NOTE: To select a different authentication chain or to change your selection, **Select Chains**.

AUTHORIZATION POLICIES

(Conditional) **Select Authorization Policies** to select the appropriate authorization policies for Account Claiming application.

4 Select **Save** to save the Account Claiming application.

5 Creating Service Applications

Applications > New Application

Single Sign-on provides applications for frequently used services that allow federated connections. The applications contain instructions to help you create an application in the Single Sign-on environment. An **application** is an object that contains all the necessary items to create a federated connection to an external service that supports a single sign-on connection. A **federation connection** establishes a trust between Single Sign-on and a service. A **service** is an application, service, or resource that you want to provide a single sign-on experience for your users.

Single Sign-on allows you to create applications that contain all the items required to create the federated connection to different services. Currently, Single Sign-on supports OAuth applications and SAML applications. It also allows you to create appmarks for the services at the same time.

If you have obtained the metadata from the service you can import the metadata or you can provide the details that are contained in the metadata. The options are unique per application.

To create an application:

1. Gather the required information from the service to create the application.
2. On the New Applications page, select the appropriate service you want to connect to with a federation connection.
3. Specify a unique name for the application.
4. Specify a detailed description for the application to describe what the application does.
5. Select **Create Appmark** to create one or more appmarks for the application.
6. Select **Enable** to enable the application.
7. Select **Toggle Instructions** to view the unique instructions for the application.
8. (Conditional) Select **Edit Metadata XML** to import the metadata XML for the application.
9. (Conditional) Use the following information and the instructions to create the metadata for the application.

   Entity ID

   Specify the URL of the application where Single Sign-on obtains the entity ID for the users.

   Login URL

   Specify the login URL of the application.

   Logout URL

   Specify the logout URL of the application.

   NOTE: Not all applications have a logout URL.

   Signing Certificate

   Specify the signing certificate to have secure communications to the application.
10. Select **Edit Chains**, the select the appropriate authentication chain for the application.

11 (Conditional) Select the appropriate authorization policies for the application.

12 (Conditional) Select **Advanced Settings** to make changes to the OAuth or SAML **Advanced Settings**.

13 Select **Save** to save the application.

6 Creating Appmarks

New Application > Appmarks

Appmarks are enhanced bookmarks for resources or services that you have configured to allow a single sign-on experience for your users. The **applications** you create in Single Sign-on contain all of the required information to create a single sign-on experience for your users including the appmarks. You can create multiple appmarks per application. Each appmark contains a URL for the resource and an image that you upload for the appmark.

Single Sign-on displays the appmarks on the Application Portal where the users log in and access the services. After you create an appmark, Single Sign-on automatically adds the appmark to the Application Portal for you.

You can also see the appmarks for each application when you edit an application. You can create one or more appmarks when you or you can create appmarks without having to create an application as well.

- "Overview of Appmark Categories" on page 25
- "Create Appmarks" on page 26

Overview of Appmark Categories

Single Sign-on allows you to categorize appmarks. You select the category type when you create the appmark. The administration console and the Application Portal display the appmarks by category. If you only have one type of category, the administration console and the Application Portal only display the category of appmark that you have.

The application categories are:

Applications

The **Applications** category is for the appmarks you click to access the contected service or bookmark.

System

The **System** category is for administration or management items administrators or uses must perform.

Task

The **Task** category is for one time tasks that administrators or users must perform.

Create Appmarks

Appmarks are enhanced bookmarks for resources or services that you have configured to allow a single sign-on experience for your users. As an administrator, you view the appmarks when you edit an application. You can create one or more appmarks when you create an application or you can create an application as well.

To create an appmark:

1. Select **Change Image**, then browse to an image and select it to use the image to represent the appmark. The users see this image.
2. Specify a name for the appmark.
3. Specify a description for the appmark. Add enough details that other administrators can understand what the appmark does.
4. Select **Enable** to enable the appmark.
5. Select the appropriate **Application Portal Category** for the appmark. The categories are:

   Applications

   Select **Applications** for an appmarks that you create for applications or services for your users.

   Launch Mode

   Select the appropriate mode to launch the appmark. The options are:

   Same Tab

   Single Sign-on launches the appmark URL in your current tab. This option is best for mobile devices or if you do not want a lot of tabs open.

   New Tab

   Single Sign-on launches the appmark URL in a new tab.

   System

   Select **System** for an appmark that you create for administration or management items.

   Tasks

   Select **Tasks** for an appmark that you create that is a task for your users.
6. In **Appmark URL**, specify the URL for the appmark. The URL is what the users access to have a single sign-on experience to the service.
7. (Conditional) Select the appropriate authorization policies that you want to apply to this appmark.
8. Select **Save** to save the appmark.

7 Creating an OAuth Application

The following information helps you understand what OAuth is and how to create an OAuth application in Single Sign-on. In the Single Sign-on environment, an **application** is an object that contains all of the necessary items to create a federated connection to an external service that supports single sign-on connections. A **federation connection** establishes a trust between Single Sign-on and a service. A **service** is an application, service, or resource that you want to provide a single sign-on experience to for your users.

Single Sign-on supports OAuth 2.0.

- "Understanding OAuth 2.0" on page 27
- "Create an OAuth Application" on page 34
- "Select Chains" on page 35
- "Configure the OAuth Advanced Settings" on page 36
- "Obtaining an OAuth Token from an Application" on page 40

Understanding OAuth 2.0

OAuth 2.0 is an authorization framework that enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its behalf. It provides a framework that allows users to grant websites or services access to their information from other websites or services but without giving them the passwords.

Single Sign-on supports OpenID Connect. OpenID Connect is a simple identity layer on top of the OAuth protocol. It enables Clients to verify the identity of the end user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end user in an interoperable and REST-like manner.

The following information is intended for customers who want to create OAuth services for their environments. The information was taken from the OAuth 2.0 RFC. If you only configuring Single Sign-on to function with an existing OAuth service, proceed to Create an OAuth Application.

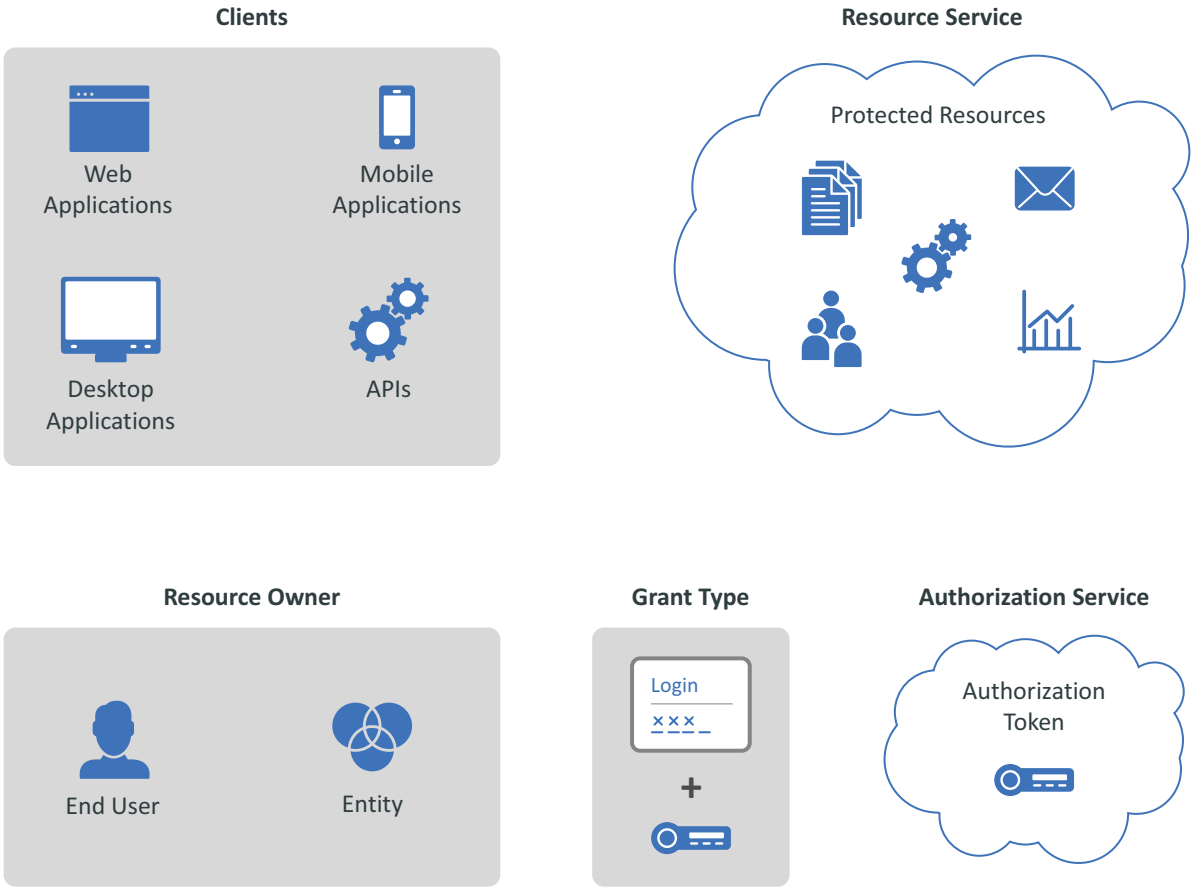
- "OAuth Components" on page 28
- "OAuth Authorization Grant" on page 29
- "Example of Single Sign-on with OAuth" on page 32

OAuth Components

OAuth 2.0 is an open standard for ownership consent and access delegation to create a federation connection with Single Sign-on and a protected resource. It provides a framework that allows users to grant websites or services access to their information from other websites or services but without giving them the passwords.

OAuth consists of different components that participate in the framework to create the federation connection. Figure 7-1 depicts these different components.

Figure 7-1 *OAuth 2.0 Components*



Resource Owner

An entity capable of granting access to a protected resource. When a resource owner is a person, the person is an end-user of the protected resource.

Resource Service

A service hosting the protected resources, capable of accepting and responding to protected resource requests using authorization tokens.

Client

An OAuth client is an application making protected resource requests on behalf of the resource owner and with its authorization. An OAuth client can execute on a server, a desktop, or a mobile.

Authorization Service

An authorization service issues authorization tokens to the client after successfully authenticating the resource owner and obtaining authorization.

Authorization Grant

An OAuth authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the OAuth client to obtain an authorization token.

When you configure an application in Single Sign-on using OAuth, the application is the protected resource and Single Sign-on provides the authorization service for the protected resource.

OAuth Authorization Grant

An OAuth authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the OAuth client to obtain an authorization token. OAuth supports four grant types: authorization code, implicit, resource owner password credentials, and client credentials. OAuth also contains an extensibility mechanism for defining additional types.

- "Authorization Code Grant (Web Server)" on page 29
- "Implicit Grant" on page 30
- "Resource Owner Credential Grant" on page 31
- "Client Credential Grant" on page 32

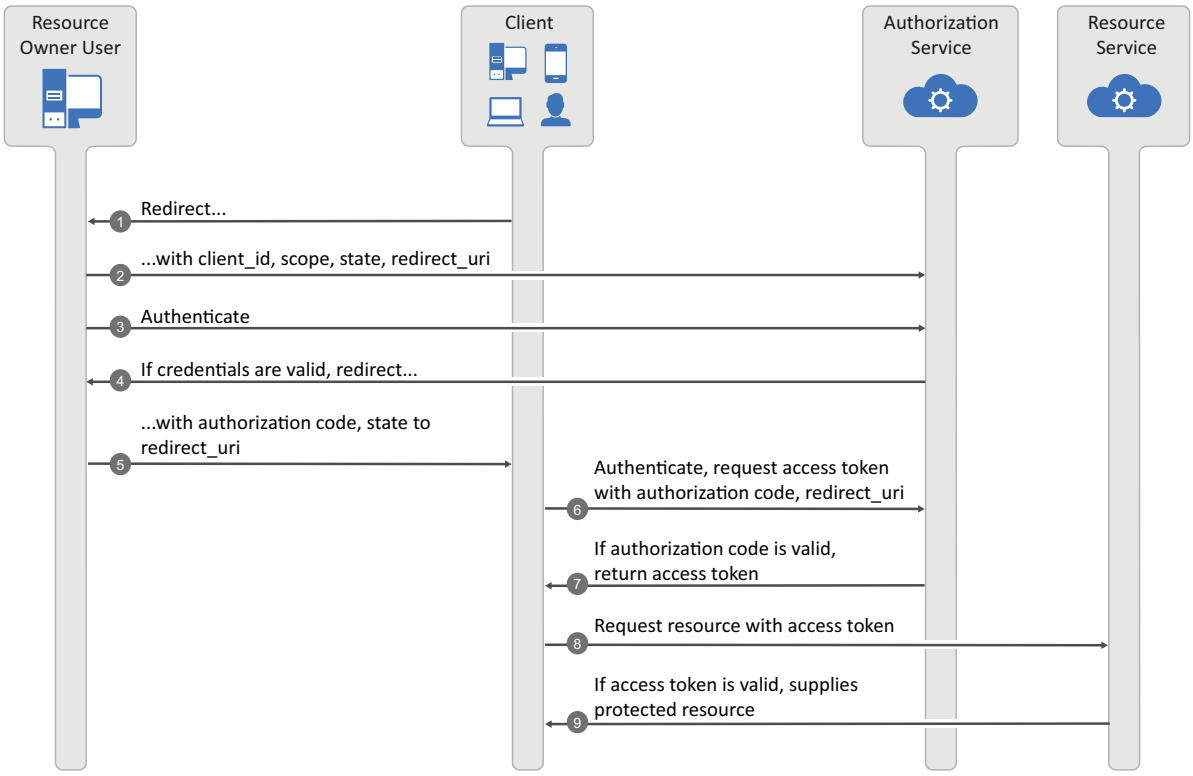
Authorization Code Grant (Web Server)

You use the authorization code grant with client applications hosted on a secure server or by a secure service use the Authorization Code Grant. Client applications use this grant to obtain both Access tokens and Refresh tokens. This grant ensures that both types of tokens remain with the client web application (the web server side) and the authorization service does not send these to the browser. Only the authorization code is visible to the browser.

The client application redirects the resource owner to the authorization service through the web browser. The resource owner authenticates at the authorization service. The authorization service obtains the resource owner's consent and then redirects the web browser with the authorization code to the client application.

This flow is suitable for client applications that can interact with the resource owner's user-agent and can receive incoming requests from the authorization service.

Figure 7-2 *OAuth Authorization Code Grant Flow*



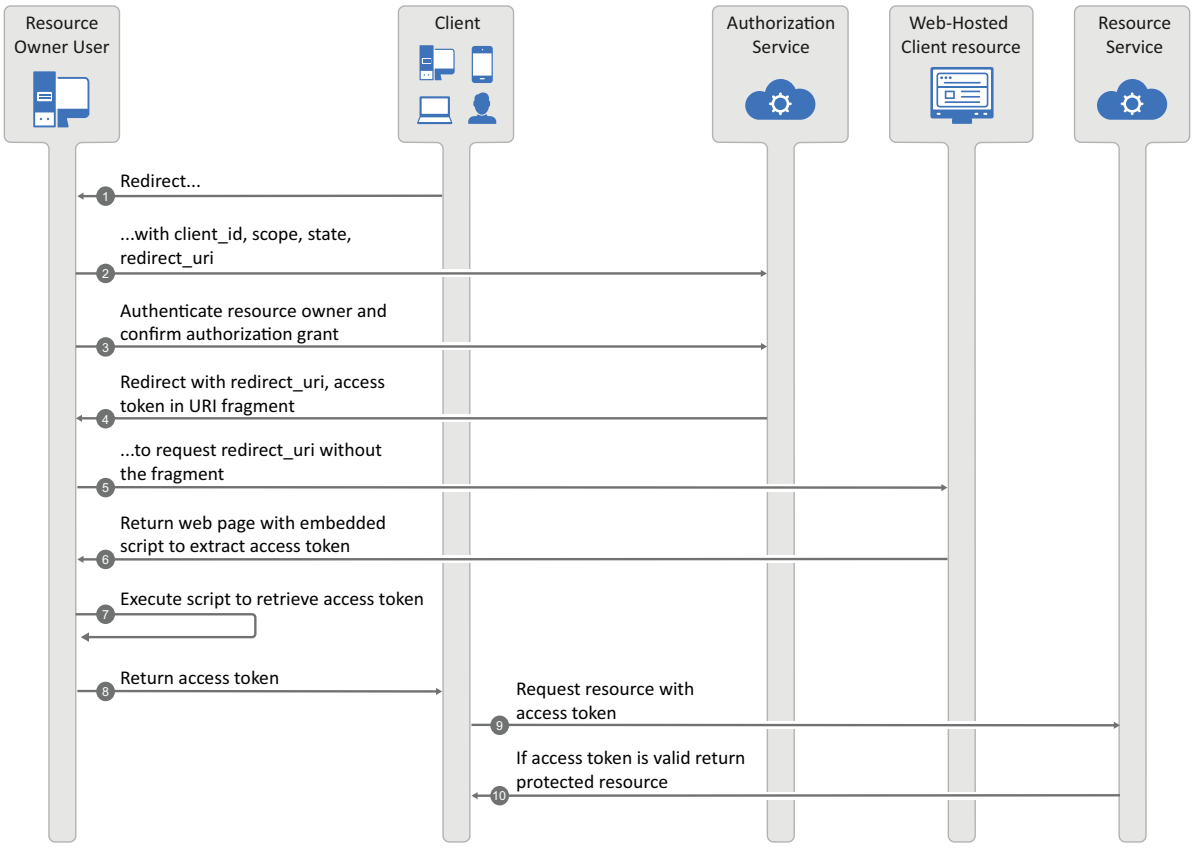
Implicit Grant

You use the implicit flow grant with:

- Single-page applications, where the user navigates between different screens of the website without loading different web pages in the browser, such as Google Mail.
- Applications that run on the user's device, such as mobile apps.
- Web applications that do not require high security. Applications that require high-security use the authorization code flow.

A client application can implement this flow in a browser using a scripting language such as JavaScript or Flash, from a mobile device, or a desktop application. After a user grants the requested authorization, the authorization service returns an access token to the application. An intermediate authorization code is not required. As the authorization service sends the access token to the web browser, this flow offers less security than the authorization code.

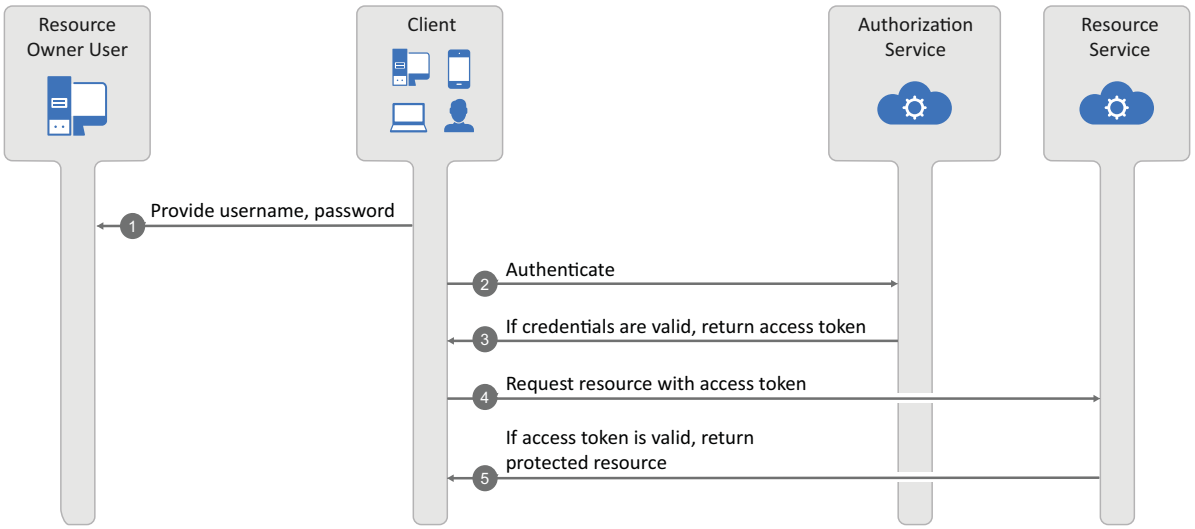
Figure 7-3 *OAuth Resource Own Implicit Grant*



Resource Owner Credential Grant

You use the resource owner credential grant flow for highly-trusted applications, applications owned by the service, and client applications that have a trust relationship with resource owners. In this flow, the client application sends the user's credentials along with its credentials to the authorization service. The authorization service provides an access token and a refresh token to the client application. The user does not need to log in to approve the request.

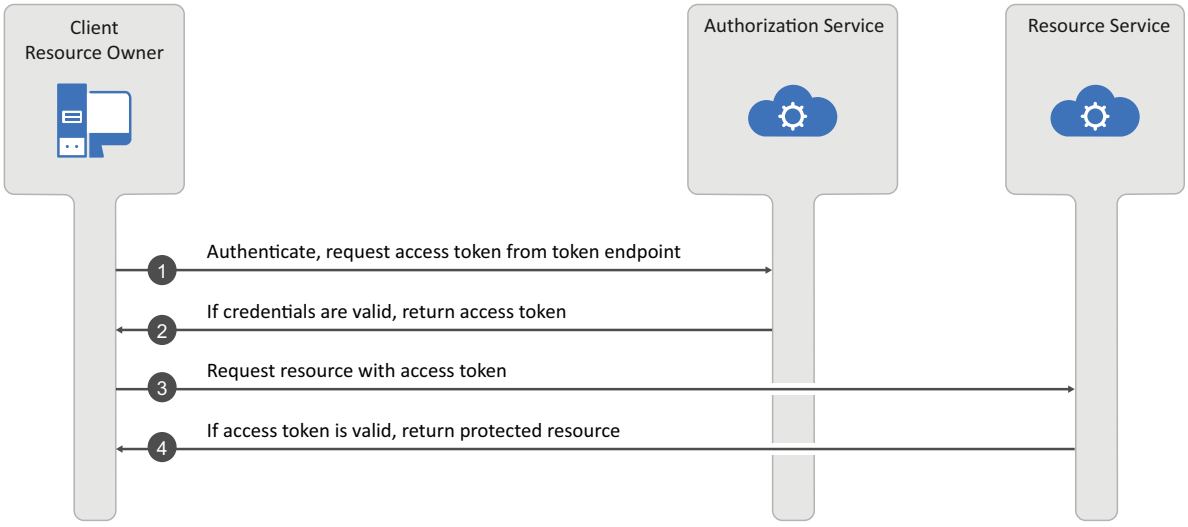
Figure 7-4 *OAuth Resource Owner Credential Grant Flow*



Client Credential Grant

You use the client credential grant is useful for headless clients and batch processing of scripts. The applications access their resources from the resource service. This grant type only requires the client application's credentials. The resource owner's credentials are not required.

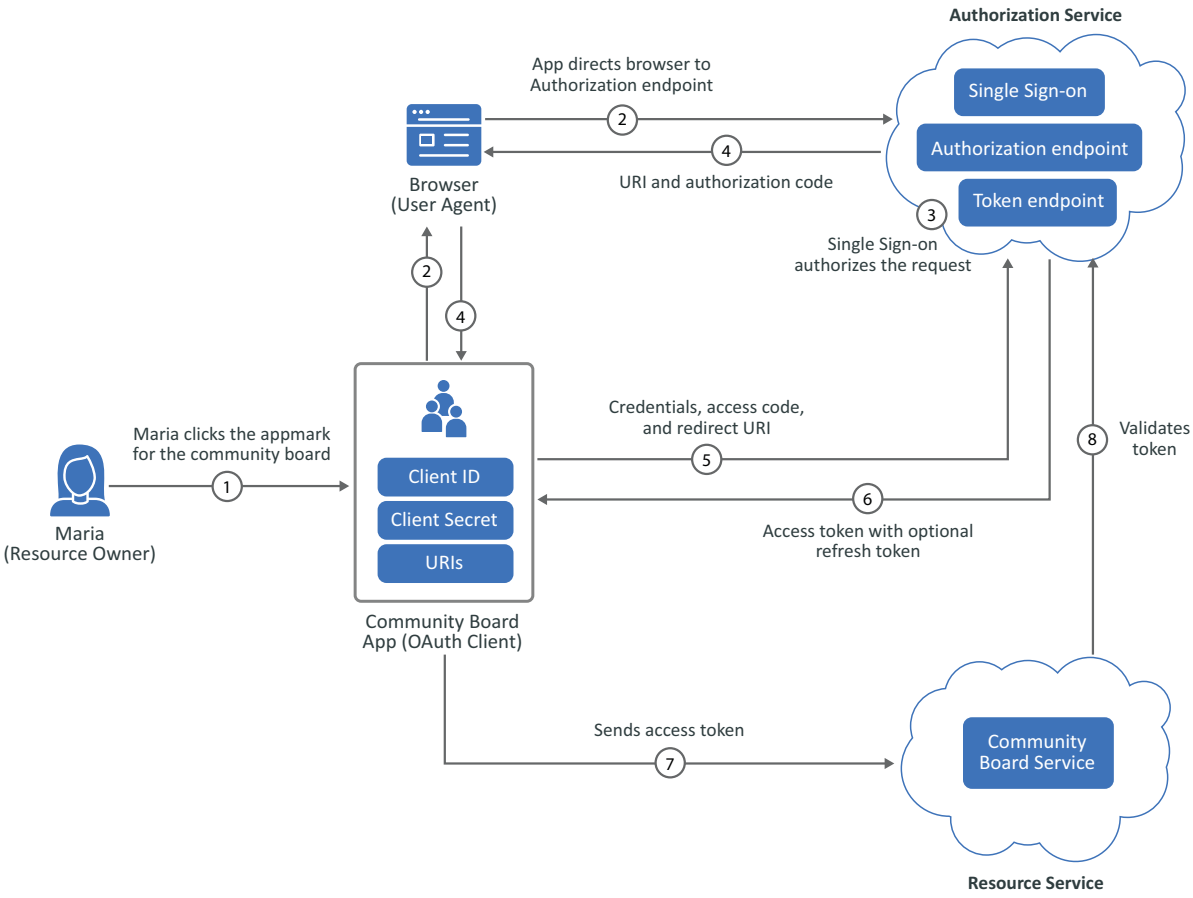
Figure 7-5 *OAuth Client Credential Grant Flow*



Example of Single Sign-on with OAuth

The following example is using the authorization code grant. During the authorization code process, an authorization service acts as an intermediary between the OAuth client and the resource. Instead of requesting authorization directly from the resource owner, the OAuth client directs the resource owner to an authorization service, which in turn directs the resource owner back to the client with the authorization code. Figure 7-6 depicts that Maria Belefonte wants to access and use her organization's online community board.

Figure 7-6 *OAuth Authorizations to a Community Board*



1. Maria Belafonte, the resource owner, selects the appmark for the web app of the community board. The browser acts as Maria's agent during the authorization process.
2. The app initiates the process by directing the request for authorization through the browser to the authorization endpoint. The app is an OAuth client that contains the client ID, client secret, and any URIs added to the configuration. The authorization endpoint is Single Sign-on.
3. Single Sign-on authenticates Maria through the browser, and it either grants or denies Maria's access request.
4. The app uses the redirection URI provided earlier to redirect the browser to the community board app. The redirection URI includes an authorization code and any local state previously provided by the OAuth client.
5. The app requests an access token from Single Sign-on through the token endpoint. The app authenticates with its client credentials and includes the authorization code received in the previous step. The app also includes the redirection URI used to obtain the authorization code for verification.
6. Single Sign-on validates the client credentials and the authorization code. It also ensures that the redirection URI received matches the URI used to redirect the app in Step 4. If valid, Single Sign-on responds with an access token.
7. The app sends the access token to the community board service to get access.
8. The community board service sends the token to Single Sign-on for validation. On successful validation, Maria accesses the community board service.

Create an OAuth Application

Applications > New Application > OAuth Application

An **application** contains all of the required configuration information to allow you to create a federated connection using OAuth from Single Sign-on to other OAuth services. A **federation connection** establishes a trust between Single Sign-on and a service. A **service** is an application, service, or resource for which you want to provide a single sign-on experience for your users.

The application contains the URL of the OAuth service. The application also contains an appmark that provides simple access to the service with a single sign-on experience for your users. After you save the application, Single Sign-on automatically generates a client ID and a client secret for you to use with the OAuth clients.

1. Gather the required information about the service to create the application.
2. (Optional) Select **Change Image**, then browse and select an image to use for this OAuth application.
3. Use the following information to configure the OAuth application:

   Application Name

   Specify a unique name for the OAuth application.

   Application Info

   Specify the details about this application to explain what the application contains.

   Enable

   Select **Enable** to enable this application after you save it.

   Advanced Settings

   Use the information in Configure the OAuth Advanced Settings to define or enable the appropriate options for your environment.

   Redirect URIs

   Specify one or more of the URIs where Single Sign-on redirects the OAuth clients for the OAuth authentication process.

   IMPORTANT: If you do not specify one or more URIs, Single Sign-on does not authenticate the OAuth clients.

   Client ID

   Single Sign-on automatically generates the **Client ID** for the OAuth application. You can copy the **Client ID** as needed.

   Client Secret

   Single Sign-on automatically generates the **Client Secret** for the OAuth application.You can only view the **Client Secret** when you create the application. You must record the **Client Secret** for use later.

   Select **Generate New Secret** if you have forgotten the **Client Secret**.

   AUTHENICATION CHAINS

   Select an authentication chain, then select **Select Chains**. The **Authentication Chain** field is empty when you first create the application.

NOTE: To select a different authentication chain or to change your selection, **Select Chains**.

AUTHORIZATION POLICIES

(Conditional) **Select Authorization Policies** to select the appropriate authorization policies for this application.

4 Create one or more appmarks for the OAuth application.

5 Select **Save** to save the OAuth application.

6 Access the client ID and client secret to configure the OAuth service for a federated connection.

6a On the Applications page, select the menu for this OAuth application.

6b Select **Edit**.

6c Copy the client ID and the client secret to add to APIs or applications that require authorization with this OAuth application. Where you place them depends on the service to which you are connecting.

Select Chains

Single Sign-on uses Advanced Authentication for all authenticate processes. Advanced authentication uses methods and chains to provide different authentication methods to the users. When you create an application, Advanced Authentication automatically creates two default chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html): LDAP Password Only and Password Only.

If you want to use additional chains than the default chains, you need to create new methods (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_methods.html) and chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html). After you create the new chains, they appear when you create or edit a custom OAuth application.

You can select one or more chains for the users to use when authenticating to the OAuth application. The users must be able to successfully complete all of the chains selected to be authenticated to the OAuth application.

Select chains:

1 Select one of the following options for **Allow Users to Select Chains**:

NOTE: There is a corresponding option in the Advanced Authentication chains settings that display the option that you select.

ON

Enables users to select any authentication chain during the authentication process.

OFF

Designates the top selected chain as the primary authentication method during the authentication process.

OPTIONAL

Allows users to customize authentication preferences by selecting the first chain and providing additional options during the authentication process.

2 Select one or more of the default chains listed, then select **Done**.

3 (Conditional) To select different chains.

3a Create the methods (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_methods.html) and new chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html) in Advanced Authentication.

3b Create a new or edit an existing OAuth application.

3c Click **Select Chains**.

3d Select the new chain or chains, then select **Done**.

Configure the OAuth Advanced Settings

Applications > New Application > OAuth Application > Advanced Settings

By default, Single Sign-on enables all of the OAuth grant types. The OAuth **Advanced Settings** allow you to use specific OAuth grant types and define scopes and claims for your OAuth application.

- "Configure Grant Types" on page 36
- "Configure Scopes" on page 38
- "Configure Claims" on page 39

Configure Grant Types

Applications > New Application > OAuth Application > Advanced Settings > Grant Types

By default, Single Sign-on enables all of the OAuth grant types. If you enable any of the advanced settings for grant types disables all grant types and the default settings for those grant types.

To change the default grant types:

1 While creating the OAuth application, select **Advanced Settings**.

2 Select the appropriate Grant Types for your organization:

Grant Types > Support Authorization Code

Enable this option to allow the OAuth application to support the Oauth authorization code grant type. When you enable this option, specify an **Authorization Code Timeout** in seconds.

Grant Types > Support Implicit

Enable this option to allow the OAuth application to support the OAuth authorization implicit grant type.

Grant Types > Support Client Credentials

Enable this option to allow the OAuth application to support OAuth client credentials grant type.

Grant Types > Support Resource Owner Credentials

Enable this option to allow the OAuth application to support OAuth authorization resource owner credentials grant type. This option is disabled by default.

IMPORTANT: When you enable this grant type, Single Sign-on disables **Support Authorization Code** and **Support Implicit** grant types because this grant type only allows non-browser authorizations. For example, this option enables APIs to request authorizations.

3 Use the following information to define the options for the different grant types:

Options > Enable OpenID Connect

Enable this option to allow Single Sign-on to implement OpenID Connect, which is a single sign-on protocol, on top of the OAuth authorization process. It allows client applications to verify the identity of a user based on the authentication performed by Single Sign-on. It also allows client applications to obtain a user's basic profile information.

Options > Enable Public Client

Enable this option to allow Single Sign-on to authorize public OAuth clients without requiring a token. By default, Single Sign-on always enables Proof Key for Code Exchange (PKCE) (https://www.rfc-editor.org/rfc/rfc7636) for all public clients. If you enable to **Require PKCE** for confidential clients, the public and confidential clients use PKCE.

NOTE: If you enable **Public Client**, Single Sign-on removes the client secret. Public clients to not have a client secret.

Options> Require PKCE

By default, Single Sign-on enables Proof Key for Code Exchange (PKCE) (https://www.rfc-editor.org/rfc/rfc7636) for all clients. You can disable PKCE for the confidential clients, but Single Sign-on keeps it enabled for the public clients.

Options > Enable All Claims in Token ID

Enable this option to allow Single Sign-on to accept all OpenID Connect claims (https://openid.net/specs/openid-connect-core-1_0.html#Claims) that have a specific token ID.

Options > Enable Token Revocation

Enable this option to allow clients to notify the authorization server that they no longer need a previously obtained refresh or access token. This option allows the authorization server to clean up security credentials. A revocation request invalidates the actual token and, if applicable, other tokens based on the same authorization grant.

Options > Enable Token Sharing

Enable this option to allow Single Sign-on to support OAuth clients that share a token.

Options > Enable Session Token Revocation

Enable this option to allow Single Sign-on to enable session token revocation. When you enable this option, specify a **Session Token Revocation Timeout** in seconds. The default value is 30 seconds.

Options > Disable RFC 9068 Tokens

Enable this option if you do not want to use JSON Web Tokens (JWT). RFC 9068 (https://www.rfc-editor.org/info/rfc9068) defines the JWT profile for OAuth tokens.

Options > Allow Token Reuse

Enable this option if you want to allow users to apply the one-time password (OTP) multiple times during authentication. This option is applicable for Email OTP, SMS OTP, and Voice OTP methods.

OTP is an authentication method (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/config_methods.html) you configure to use in chains (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/config_chain.html).

Options > Rotate Refresh Tokens

Enable this option to refresh the access token on behalf of the user without requiring interaction from the user.

4 Set the appropriate global timeout values for the grant types you selected:

Timeout > Access Token Timeout

Specify the length of time in seconds after which the access token expires. The access token includes the specific scopes and this option allows you to specify the duration of the granted access. The default value is 30 seconds.

Timeout > Refresh Token Timeout

Specify the length of time in seconds after which the refresh token expires. The default value is 30 seconds.

Timeout > Public Refresh Token Timeout

Specify the length of time in seconds after which the public refresh token expires. The default value is 30 seconds.

5 (Optional) Select **Scopes** or **Claims** to continue configuring the OAuth **Advanced Settings**.

6 (Optional) Select **Done** to leave the **Advanced Settings**, then select **Save** to save the changes.

Configure Scopes

Applications > New Application > OAuth Application > Advanced Settings > Scopes

The OAuth 2.0 RFC (https://www.rfc-editor.org/rfc/rfc6749#section-3.3) defines **scope** as scope request parameters defined in the OAuth client and the authorization service to limit the scope of the access token issued. Scopes allow you to limit the authorizations the OAuth application provides

OpenID Connect requires one scope and provides additional optional scopes that you can use. If you enable **OpenID Connect** as an option for the grant types, Single Sign-on displays all of the OpenID Connect scopes.

NOTE: You cannot edit or delete the OpenID Connect scopes. You can view the claims defined in the scopes.

The OpenID Connect scopes are:

- OpenID - mandatory
- Profile
- Email

- Address
- Phone

When you select an OpenID Connect scope Single Sign-on displays the attributes defined for the scope.

To view scopes or create a custom scope:

1. To view the details of a scope, select the name of the scope.
2. To create a custom scope, select the plus sign.
   - **2a** Use the following information to create the custom scope:

     Name

     Specify a name for the scope. The scope names appear at the top of this page.

     Title

     Specify the title of the screen the users see when presented with messages about the authorization process. For example, if you enabled the Client Credentials grant type, Single Sign-on presents a message asking the user if they want to use their email account for authorization.

     Description

     Specify a detailed description of the scope so that any other administration can understand what it does.

     Require User Permission

     Select this option if you want to present the user with a dialog box that requires them to select Yes or No to proceed with the authorization process.

     Make Scope Mandatory

     Select this option if you

     Attribute Mappings

     Add the attribute mappings to match the attributes in the local identity repository with the client identity repository.

   - **2b** Select **Done**.

   NOTE: Single Sign-on does not display the **Claims** tab when you create a custom scope.

3. (Optional) Select **Done** to leave the **Advanced Settings**, then select **Save** to save the changes.

After you create the scope and make OAuth calls through the OAuth client, Single Sign-on only sends the selected attributes in the OAuth tokens.

Configure Claims

Applications > New Application > OAuth Application > Advanced Settings > Claims

A claim (https://openid.net/specs/openid-connect-core-1_0.html#Claims) is part of the OpenID Connect (OIDC) specification. A **claim** is a piece of information that OIDC asserts about an entity. A claim contains name/value pairs that contain information about entities (users, APIs, and so forth) that are part of the OIDC authentication process. Claims are attributes.

Single Sign-on provides a list of default attribute mappings. If you can delete any of the attributes that you do not need. As soon as you make a change to the attributes, Single Sign-on removes the default list and your changes overwrite the list with your changes.

Make the appropriate changes to the attribute mappings. Single Sign-on shows the attributes from your local identity repository and maps the attributes to the attributes in the OAuth service. You creating the attribute mappings so that the OAuth service and Single Sign-on can communicate. Select a local attribute to see additional attributes that are available to select.

(Optional) Change the current attribute mappings to create a new claim:

1. Select an attribute from your local identity repository to change the current attribute mapping.
2. Specify an attribute name from the OAuth service.
3. Select whether to include the attribute in the OAuth Access Token.
4. Repeat for each required attribute mapping.
5. (Optional) Select **Grant Types** or **Scopes** to review your changes.
6. (Optional) Select **Done** to leave the **Advanced Settings**, then select **Save** to save the changes.

After you modify or create a claim and make OAuth calls through the OAuth client, Single Sign-on only sends the selected attributes in the OAuth tokens.

Obtaining an OAuth Token from an Application

To obtain an OAuth token Single Sign-on provides a well-known URL that you can access. The URL displays the attributes, scopes, claims, and tokens for the grant types. The URL is:

`https://unique-URL/osp/a/organization-name/auth/oauth2/.well-known/openid-configuration`

8 Creating a SAML Application

The following information helps you understand what Security Assertion Markup Language (SAML) is and how to create a SAML application in Single Sign-on. In the Single Sign-on environment, an **application** is an object that contains all of the necessary items to create a federated connection to an external service that supports single sign-on. A **federation connection** establishes a trust between Single Sign-on and a service. A **service** is an application, service, or resource that you want to provide a single sign-on experience to for your users.

Single Sign-on supports SAML 2.0.

- "Understanding SAML" on page 41
- "Manage a SAML Application" on page 46
- "Create a SAML Application" on page 47
- "Select Chains" on page 48
- "Configure SAML Advanced Settings" on page 49
- "Obtain the SAML Metadata" on page 50

Understanding SAML

SAML 2.0 is an open-source industry-standard XML-based protocol that allows you to create a federated, secure connection between two providers to enable web-based, single sign-on for your users. Enabling single sign-on for the users of your organization reduces the administrative overhead of distributing multiple authentication tokens to the users.

SAML supports two different flows for authentication: identity provider (IdP)-initiated and service provider (SP)-initiated authentications. Single Sign-on support both authentication flows.

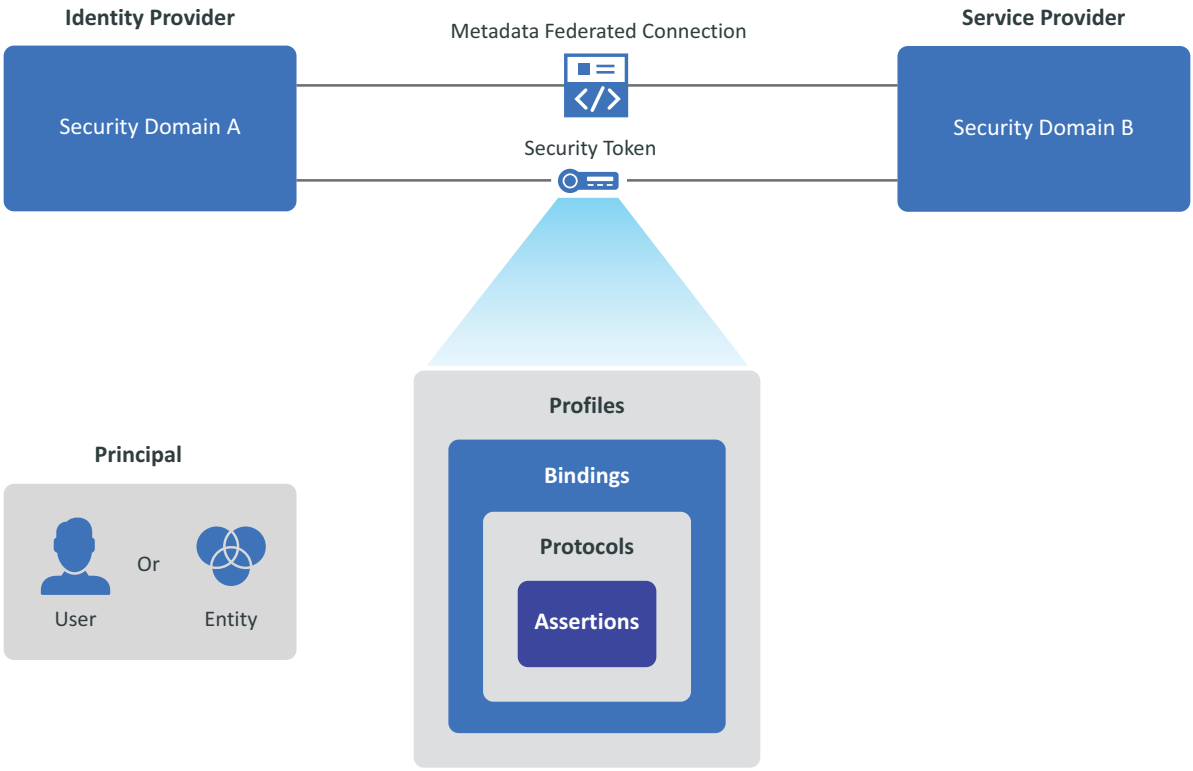
- "SAML Components" on page 42
- "Example of an IdP-Initiated Authentication with Single Sign-on" on page 44
- "Example of an SP-Initiated Authentication with Single Sign-on" on page 45

SAML Components

SAML 2.0 is an open-source industry-standard XML-based protocol that allows you to create a federated, secure connection between two providers to enable web-based, single sign-on for your users. **Federation** is a practice that allows user identities to be stored across discrete services and organizations. SAML allows these federated services and organizations to communicate with and trust one another's users.

SAML uses security tokens containing assertions to pass information about a principal (usually an end user) between an identity provider and a service provider. SAML is a two-way communication channel. You must configure both security domains to trust each other and allow single sign-on to occur. The following graphic and information provide more details about SAML 2.0.

Figure 8-1 *SAML Components*



The graphic lists the different components involved in permitting the transfer of identity, authentication, attribute, and authorization information between autonomous organizations that have an established trust relationship.

Principal

A principal is an entity that a computer system or network can authenticate. A principal is usually a user, but it can also be a group, computers, services, computational entities such as processes and threads, or any group of such things.

Identity Provider

An identity provider (IdP) is a service that creates, manages, and maintains the identity information about the principals and provides authentication services to services within a federation or a distributed network that rely on that identity information.

Service Provider

A service provider (SP) is an organization that provides services to your organization. It can be a separate division in an organization, a software-as-a-service provider, a third-party organization, or any such entity. The service providers either host or provide the applications and services to different organizations.

Security Domain

A security domain is a separate network of computers. Each security domain uses firewalls and other such tools to separate the information and resources on the network separate from other networks for security reasons.

SAML Core Components

The following items are the core components that allow SAML to provide secure, single sign-on between two different security domains.

Metadata

Metadata defines a way to express and share configuration information between two security domains. For example, the metadata could contain the identity provider information, the service provider information, supporting identity attributes, and key information for encryption and signing. You configure the metadata between two security domains to allow a secure, web-based, single sign-on experience for the users. SAML Metadata is defined by its XML schema. You must configure each security domain with the metadata information for each identity provider and service provider.

Profiles

SAML profiles exist to satisfy a particular business use case, such as the Web Browser SSO profile. Profiles typically define constraints on the contents of SAML assertions, protocols, and bindings to solve the business use case in an interoperable fashion. There are also Attribute Profiles, which do not refer to any protocol messages and bindings, that define how to exchange attribute information using assertions in ways that align with several common usage environments such as X.500/LDAP directories or DCE.

Bindings

SAML bindings transport SAML protocol messages between the participants through lower-level communication or messaging protocols (such as HTTP or SOAP).

Protocols

SAML protocol messages make the SAML-defined requests and return appropriate responses. The SAML-defined protocol XML schema defines the structure and contents of these messages.

Assertions

SAML assertions carry statements about a principal that an asserting party claims to be true. The SAML assertion XML schema defines the valid structure and contents of an assertion. An asserting party usually creates assertions based on a request of some sort from a relying party, although, under certain circumstances, the asserting party delivers the assertions in an unsolicited manner to the party that relies on them.

Example of an IdP-Initiated Authentication with Single Sign-on

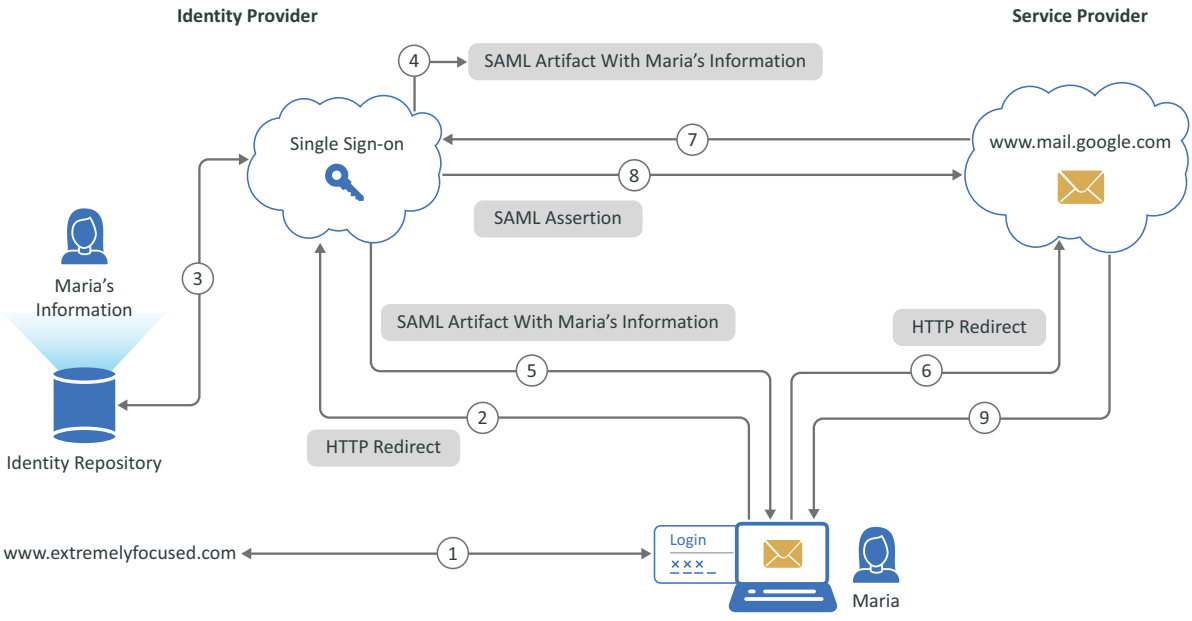
Figure 8-2 is an example of the single sign-on IdP-initiated authentication process for a user named Maria Belafonte to Google Mail. Maria's company is named Extremely Focused. Maria starts the authentication process by selecting an appmark in the Application Portal for Google Mail.

There are configuration steps that you must perform to create a federation between Single Sign-on and Google Mail. **Federation** is a practice that allows user identities to be stored across discrete services and organizations. SAML allows these federated services and organizations to communicate with and trust one another's users. You must create the federation between Single Sign-on and Google Mail before the SAML single sign-on authentication process functions. The configuration steps for the federation are the same for IdP-initiated and SP-initiated authentications. The required configuration steps are:

- **Share metadata:** You must add the metadata for Google Mail to Single Sign-on and you must add the metadata for Single Sign-on to Google Mail. This step allows each domain to know about and trust the other domain to allow authentications. The metadata can include certificates to create a secure trust between the two security domains.
- **Create an account in the identity repository:** Create an account for Maria Belafonte in the identity repository for Single Sign-on. You must add the users to whom you want to provide a single sign-on experience in the identity repository. This step allows Single Sign-on to know about and obtain the required information to create the single sign-on experience.
- **Create an account in Google Mail:** Create an account for Maria Belafonte in Google Mail. Not all security domains that use SAML require that the entity accounts exist in their system but Google Mail does.

Figure 8-2 and the steps explain the IdP-initiated authentication process that occurs after you have configured Single Sign-on and Google Mail to allow single sign-on authentication. Maria's experience as a user is simply to log in to her company's website and select the appmark for Google Mail. Single Sign-on does not expose the steps required to perform the SAML IdP-initiated authentication to Google Mail to any users.

Figure 8-2 *SAML **IdP-Initiated** Authentication with Single Sign-on to Google Mail*



Single Sign-on IdP-initiated authentication to Google Mail using SAML 2.0:

1. Maria Belafonte logs in to her company's website www.extremelyfocused.com.
2. Maria selects the link in a browser for her company's Google Mail and the browser sends the authentication request to Single Sign-on.
3. Single Sign-on performs a lookup of Maria's information for the SAML attributes defined for her account.
4. Single Sign-on creates a SAML artifact that contains Maria's information.
5. Single Sign-on sends the SAML artifact that contains Maria's information and an HTTP Redirect back to the browser.
6. When the HTTP Redirect reaches the browser, it sends the authentication request to Google Mail.
7. Google Mail sends a request to Single Sign-on for the SAML assertion to validate that Maria has the proper permissions to authenticate to Google Mail.
8. Single Sign-on generates a SAML assertion for Maria with the artifact and Maria's information, then Single Sign-on sends the assertion back to Google Mail for Maria.
9. Google Mail sees that Maria's information in the SAML assertion allows her access to Google Mail for her company. Google Mail establishes an authentication session for Maria.

Example of an SP-Initiated Authentication with Single Sign-on

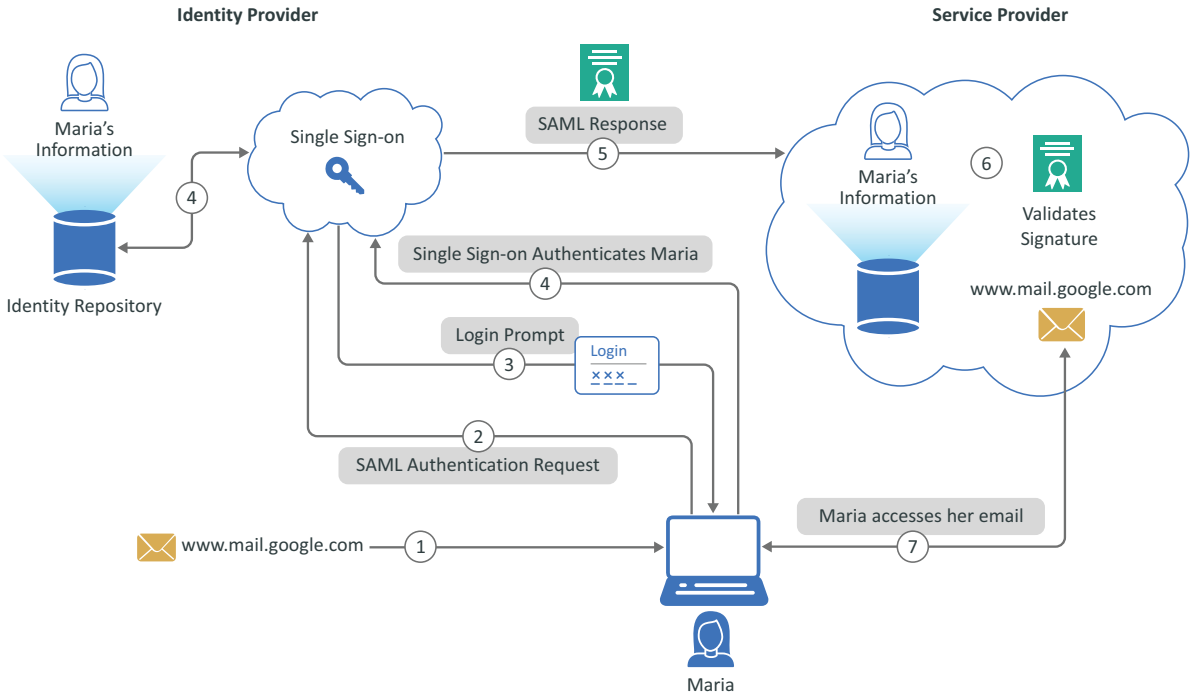
Figure 8-3 is an example of the single sign-on SP-initiated authentication process for a user named Maria Belafonte to Google Mail. Maria's company is named Extremely Focused. Maria starts the SP-initiated authentication process by selecting a link for the URL to Google Mail in her browser.

There are configuration steps that you must perform to create a federation between Single Sign-on and Google Mail. **Federation** is a practice that allows user identities to be stored across discrete services and organizations. SAML allows these federated services and organizations to communicate with and trust one another's users. You must create the federation between Single Sign-on and Google Mail before the SAML single sign-on authentication process functions. The configuration steps for the federation are the same for IdP-initiated and SP-initiated authentications. The required configuration steps are:

- **Share metadata:** You must add the metadata for Google Mail to Single Sign-on and you must add the metadata for Single Sign-on to Google Mail. This step allows each domain to know about and trust the other domain to allow authentications. The metadata can include certificates to create a secure trust between the two security domains.
- **Create an account in the identity repository:** Create an account for Maria Belafonte in the identity repository for Single Sign-on. You must add the users to whom you want to provide a single sign-on experience in the identity repository. This step allows Single Sign-on to know about and obtain the required information to create the single sign-on experience.
- **Create an account in Google Mail:** Create an account for Maria Belafonte in Google Mail. Not all security domains that use SAML require that the entity accounts exist in their system but Google Mail does.

Figure 8-3 and the steps explain the SP-initiated authentication process that occurs after you have configured Single Sign-on and Google Mail to allow single sign-on authentication. Maria Belafonte's experience as a user is simply to select a link or bookmark for Google Mail. Single Sign-on does not expose the steps required to perform the SAML SP-initiated authentication from Google Mail to Single Sign-on to any users.

Figure 8-3 *SAML **SP-Initiated** Authentication to Google Mail through Single Sign-on*



1. Maria Belafonte enters the URL for Google Mail of www.mail.google.com or selects a bookmark for the URL.
2. The select in the browser Google workspace initiates login by sending a SAML authentication request to Single Sign-on.
3. Single Sign-on sends Maria to a login page and she enters her credentials.
4. Single Sign-on authenticates the Maria searching in the identity repository
5. Single Sign-on sends cryptographically signed SAML response to Google workspace. The SAML response contains a SAML assertion that tells the service provider who the user is.
6. The Google workspace validates the signature in the SAML response and identifies the user Maria
7. Google logs Maria into the Google workspace and she can access her Google mail.

Manage a SAML Application

Single Sign-on presents the same UI when you create a SAML application or edit it. When you create a SAML application, Single Sign-on provides two different methods:

Template

Single Sign-on provides multiple templates for common SAML applications. It also provides a default template to use when creating a SAML application. The template simplifies creating the SAML application.

Manually Create the Metadata

Single Sign-on allows you to manually create a SAML application by creating the required metadata. You still must configure the federation connection on the connected services.

Create a SAML Application

Applications > New Application > SAML Application

An **application** contains all of the required configuration information to allow you to create a SAML connection from Single Sign-on to other SAML services. Single Sign-on is the identity provider to any SAML service providers that support single sign-on or federation connections. A **federation connection** establishes a trust between Single Sign-on and a service. A **service** is an application, service, or resource that you want to provide a single sign-on experience to for your users.

NOTE: Currently, Single Sign-on only supports SP-initiated authentication applications.

The application contains the metadata for the service provider to allow single sign-on to occur for your users. The application also contains an appmark that provides simple access to the service with a single sign-on experience for your users.

To create an application for a SAML service provider:

1. Gather the required information about the service to create the application.
2. (Optional) Select **Change Image**, then browse and select an image to use for this SAML application.
3. Use the following information to configure the SAML service provider:

   Application Name

   Specify a unique name for the application that contains the SAML service provider metadata.

   Applicaiton Info

   Specify the details about this application to help others understand what the application contains.

   Enable

   Select **Enable** to enable the SAML connection between Single Sign-on and the SAML service provider.

   Advanced Settings

   Use the information in Configure SAML Advanced Settings to define or enable the appropriate options for your environment.

   METADATA

   Use one of the following options to populate the metadata for the SAML service provide application:

Manually create the metadata

Select **Edit Metadata XML**, then specify the metadata in properly formated XML, or select **Use Metadata File**, then browse to and select the metadata file you want to use.

Use the default SAML application template

Populate the following fields to use the default SAML application template.

Entity ID

Specify the Entity ID to use in the SAML authentication.

Attribute Consume Service Endpoint

Specify the attribute for the consume service endpoint.

Single Logout Service Endpoint

Specify the single logout endpoint for the connected service.

Signing Certificate

Specify the signing certificate Single Sign-on uses to encrypt the authentication process.

Federation Instructions

Follow the federation instructions to configure the federation connection to the connected service.

AUTHENTICATION CHAINS

Select an authentication chain, select **Select Chains**. The **Authentication Chain** field is empty when you first create the application.

NOTE: To select a different authentication chain or to chain your selection, click **Select Chains**.

AUTHORIZATION POLICIES

(Conditional) **Select Authorization Policies** to select the appropriate authorization policies for this application.

4 Select **New Appmark** to create one or more appmarks for the SAML application.

5 Select **Save** to save the SAML service provider application.

Select Chains

Single Sign-on uses Advanced Authentication for all authenticate processes. Advanced authentication uses methods and chains to provide different authentication methods to the users. When you create an application, Advanced Authentication automatically creates two default chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html): LDAP Password Only and Password Only.

If you want to use additional chains than the default chains, you need to create new methods (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_methods.html) and chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html). After you create the new chains, they appear when you create or edit a custom SAML application.

You can select one or more chains for the users to use when authenticating to the SAML application. The users must be able to successfully complete all of the chains selected to be authenticated to the SAML application.

To select chains:

1 Select one of the following options for **Allow Users to Select Chains**:

NOTE: There is a corresponding option in the Advanced Authentication chains settings that display the option that you select.

ON

Enables users to select any authentication chain during the authentication process.

OFF

Designates the top selected chain as the primary authentication method during the authentication process.

OPTIONAL

Allows users to customize authentication preferences by selecting the first chain and providing additional options during the authentication process.

2 Select one or more of the default chains listed, then select **Done**.

3 (Conditional) To select different chains.

3a Create the methods (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_methods.html) and new chains (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html) in Advanced Authentication.

3b Create a new or edit an existing SAML application.

3c **Select Chains**.

3d Select the new chain or chains, then select **Done**.

Configure SAML Advanced Settings

Applications > New Application > SAML Application > Advanced Settings

Single Sign-on provides advanced settings for the SAML applications if you need them. You define the advanced settings when you create the SAML application.

1 While creating an application, select **Advanced Settings**.

2 Use the following information to enable the advanced settings when creating a SAML application:

NameID Format

Select the appropriate attribute in the local Advanced Authentication repository to send as the NameID attribute to the connected application. Plus, you define the format of the NameID attribute Single Sign-on sends to the connected application:

Transient (Default)

Enable this option so that Single Sign-on generates a new value for the NameID attribute for each authentication. Single Sign-on generates a new value for each authentication so there is no attribute for you to select.

Send E-Mail as NameID

Enable this option to use the attribute `mail` from the Advanced Authentication repository to the connected application.

NOTE: You cannot select any other attribute if you are using the user's email as the NameID attribute.

Persistent

Enable this option to always use the value from the attribute that you select in the Advanced Authentication repository as the NameID attribute Single Sign-on sends to the connected application.

Unspecified

Enable this option to use a custom attribute as the NameID attribute. In **NameID Attribute**, select the appropriate attribute in the Advanced Authentication repository to send to the connected application to use as the NameID attribute.

Allow Token Reuse

Enable this option if you want to allow users to apply the one-time password (OTP) multiple times during authentication. This option is applicable for Email OTP, SMS OTP, and Voice OTP methods.

OTP is an authentication method (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/config_methods.html) you configure to use in chains (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/config_chain.html).

Attribute Mapping

Map the appropriate attributes in your local identity repository to the attributes in the SAML services. You map the attributes between your identity repository and the SAML service so that the two services can communicate. Select an attribute to see additional attributes that are available to select. Single Sign-on provides a list of default attributes.

3 Select **Done** to save these options and close **Advanced Settings**.

Obtain the SAML Metadata

To complete the SAML configuration you must obtain the SAML metadata from Single Sign-on to add to the service where you are creating a federated connection.

To obtain the Single Sign-on metadata:

1 Log in to your Advanced Authentication service.

2 Select **Policies > Web Authentication**.

3 Select **Download IdP SAML 2.0 Metadata**. Advanced Authentication displays the metadata in a new browser tab.

4 Copy the SAML metadata and save it in a `.xml` file.

5 Import the SAML metadata file into the application or service to create the federation connection between it and Single Sign-on.

9 Creating an External Identity Provider

Single Sign-on uses the Advanced Authentication repository as the identity provider for the OAuth and SAML protocols. You can configure external identity provider applications to provide the user verification instead of Advanced Authentication. Currently, Single Sign-on only supports one identity provider application at a time.

Creating an external identity provider requires different processes.

- "Create an External SAML Identity Provider Application" on page 53
- "Create an Advanced Authentication Chain" on page 54
- "Create a Service Application" on page 54

Create an External SAML Identity Provider Application

Applications > New Application > SAML Application Identity Provider

You must create an application that represents the external identity provider. This application contains the metadata configuration information for the external identity provider. You obtain the metadata information from the documentation for the external identity provider. You also need to know what attribute you want to Single Sign-on to use to validate the user accounts when then authenticate. Currently, Single Sign-on only supports one identity provider application at a time.

To create an external SAML identity provider application:

1. Gather the SAML metadata for the external identity provider from the documentation for the external identity provider.
2. Determine which attribute you want use to Single Sign-on to use to validate the user accounts in the external identity provider.
3. (Optional) Select **Change Image**, then browse and select an image to use for this SAML external identity provider.
4. In **Application Name**, specify a unique name for the external SAML identity provider application.
5. In **Application Info**, specify a detailed description of the SAML identity provider application so that other administrations can know its purpose.
6. Select **Enable** to enable the SAML connection between Single Sign-on and the SAML identity provider.
7. Select **Advanced Settings** to specify the attribute to validate the user accounts.
   - **7a** In **Assertion Attribute**, specify the name of the attribute.
   - **7b** Select **Done**, to save the attribute and close the side window.

8 (Conditional) To manually create the metadata in XML, select **Edit Metadata XML** to manually created the metadata in XML

8a (Conditional) Select **Edit Metadata**, then specify the metadata in properly formated XML.

8b (Codnitional) Select **Use Metadata File**, then browse to and select the metadata file you want to use.

8c Select **Done**.

9 (Conditional) Populate the following fields to use the default SAML application template.

Entity ID

Specify the Entity ID to use in the SAML authentication.

Login URL

Specify the URL the external identity provider uses to initiate the login event.

Signing Certificate

Specify the signing certificate the external identity provider uses to allow secure authentications for the users.

10 Select **Save**, to save the SAML identity provider.

Single Sign-on automatically creates an entry for this SAML identity provider in the Access Manager **Methods** under **SAML Service Provider**. You use this entry to create an authentication chain to use in a service application to use this SAML identity provider.

Create an Advanced Authentication Chain

Advanced Authentication Dashboard > Chains

You must create or modify an Advanced Authentication chain to include the new SAML identity provider as part of the authentication process. Single Sign-on automatically creates an entry for this SAML identity provider in the Access Manager **Methods** under **SAML Service Provider**.

When you create or modify a chain (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/config_chain.html), ensure that you add the **Method** of **SAML Service Provider** in the chain.

The last step is to create a service application that uses this chain to require authentication to the SAML identity provider to access the application. Currently, Single Sign-on only supports one identity provider application at a time.

Create a Service Application

After you have created the SAML identity provider application and the Advanced Authentication chain, you must create an application for a service that requires authentication to the SAML identity provider.

Single Sign-on supports OAuth and SAML applications authenticating through an external SAML identity provider. While you are creating the application, select the Advanced Authentication chain that contains the SAML Service Provider method.

10 Configuring Authorization Policies

Single Sign-on uses the open standard Open Policy Agent (OPA) (https://www.openpolicyagent.org/) as the authorization policy engine to evaluate authorization policies for access to the applications. You can create rule-based or OPA Rego (https://www.openpolicyagent.org/docs/latest/policy-language/) authorization policies using attributes defined on the users in the Advanced Authentication repository. The authorization policies allow you to limit access to applications or appmarks depending on the LDAP attributes and values that Single Sign-on reads when a user accesses an application.

Single Sign-on contains an authorization service that manages the rule-based authorization policies. You create the authorization policies when you create the applications or edit existing applications. You can also create authorization policies on stand-alone appmarks.

- "Overview of Authorization Policies" on page 55
- "Manage Authorization Policies" on page 55
- "Create a Rule-Based Authorization Policy" on page 57
- "Create an OPA Authorization Policy" on page 62
- "Example of an OPA Policy Document" on page 64

Overview of Authorization Policies

It is important to understand the distinction between **authentications** and **authorizations**. Applications use **authentications** to verify that users who are logging into the applications have valid credentials. Applications use **authorizations** based on authorization policies to determine if users have been assigned the correct permissions to access the application or specific components of an application.

The **authorization policies** consist of rule sets and rules that define attributes and values from the Advanced Authentication repository. The authorization service uses any attributes that are available when a user authenticates. An authorization policy is a shared resource that you can apply to appmarks and applications. By defining the attributes and values, you can create policies that limit access to specific components in the service. For example, Salesforce contains multiple products. The authorization policies allow you to limit access to the Sales product in Salesforce for only the sales people in your organization.

Manage Authorization Policies

You can create, edit, and delete the authorization policies on an applications or appmark.

- "Create Authorization Policies" on page 56
- "Edit an Authorization Policy" on page 56
- "Delete an Authorization Policy" on page 56

Create Authorization Policies

Single Sign-on uses the open standard Open Policy Agent (OPA) (https://www.openpolicyagent.org/) as the authorization policy engine to create, delete, and apply the authorization policies. Single Sign-on allows you to define attributes and values for users to provide authorizations to applications and appmarks. OPA allows you to do much more than our current use case. However, Single Sign-on does allow you to create OPA policies using Rego (https://www.openpolicyagent.org/docs/latest/policy-language/) through code editors. Rego is the policy authoring language that OPA developed to create policies.

Single Sign-on allows you to create authorization policies using two different methods. The two different methods are:

- Rule-based authorization policies
- OPA authorization polices

Edit an Authorization Policy

Applications > *Appmark* or ***Application*** **> Actions > Edit**

Single Sign-on allows you to edit the authorizations policies that you have created. You edit the authorization policies when you edit an application or appmark.

To edit an authorization policy:

1. On the Applications page, select the appropriate application or appmark that contains the authorization policies that you want to edit.
2. Select **Actions > Edit**.
3. Select **Authorization Policies**, then select **Edit**.
4. Make the appropriate changes to the authorization policies, rule sets, and rules.
5. After editing the rule sets, select **Done**.
6. On the Authorization Policies panel, select **Done** to save your changes.

Delete an Authorization Policy

Applications > *Appmark* or ***Application*** **> Actions > *Authorization Policies* > Delete**

Single Sign-on allows you to delete authorization policies. However, an authorization policy must be disabled before you can delete it.

To delete an authorization policy:

1. On the Applications page, select the application or appmark that contains the authorization policy that you want to delete.
2. Select **Actions > Edit**.
3. Select **Authorization Policies**, then select **Edit**.
4. On the appropriate authorization policy, select the check to disable the authorization policy.
5. Select **Delete**, then select **Done** to save the changes.

Create a Rule-Based Authorization Policy

Applications > New Application > Appmark or ***Application*** **> Authorization Policies > Rule-based**

Single Sign-on provide different types of rule-based authorization policies. A rule-based authorization policy consists of multiple rules and rule sets. A **rule** contains a name, description, and one or more rule sets. A **rule set** is where you select the type of rule-based authorization policy you want to use. It also combines multiple rules together using AND OR qualifiers. You can also combine multiple rule sets together with the AND and OR qualifiers.

- "Create a Rule-Based Authorization Policy with User Attributes" on page 57
- "Understanding the Default Attributes for a Rule-Based Authorization Policy" on page 58
- "Create a Rule-Based Authorization Policy with Identity Governance Roles" on page 60
- "How the Authorization Policy Using Identity Governance Roles Matches User Accounts" on page 61

Create a Rule-Based Authorization Policy with User Attributes

Applications > New Application > Appmark or ***Application*** **> Authorization Policies > Rule-based > Rule Sets + > User Attributes**

Single Sign-on allows you to create an authorization policy with user attributes to limit access to applications. You do this by defining the attributes in the Advanced Authentication repository that the authorization service checks when a user accesses an application or appmark. You can create authorization policies when you create applications or on stand-alone appmarks.

To create an authorization policy with user attributes:

1. On the application or appmark that you are creating, select **Authorization Policies**.
2. Select the plus sign (**+**) to add an authorization policy, then select **Rule-based**.
3. Use the following information to create an authorization policy:

   Name

   Specify a name for the rule-based authorization policy.

   Enabled

   Select **Enable** to enable the rule-based authorization policy after you save it.

   Description

   Provide a detailed description of what the rule-based authorization policy does. The description helps people know what the authorization policy does without having to open the authorization policy.
4. At the end of **Rule Sets**, select the appropriate qualifier (**AND**, **OR**) for the rule-based policy.
5. Next to **Rule Sets**, select the plus sign (**+**) to create a rule set.
6. Select **User Attribute** to create a user attribute authorization policy.
7. Select **New User Attribute Set** to expand the options.
8. Use the following information to define the rule sets for the authorization policy using the user attributes:

Set Name

Specify a name for the rule set.

Description

Specify a detailed description for the rule set so that other administrators will understand its purpose.

Rules

Select the plus sign + next to **Rules** to create a rule.

Qualifier

Select **AND** or **OR** for the proper qualifier for this rule.

Attribute

Select the appropriate attribute from the list of attributes the repository that Single Sign-on evaluates when a user accesses the application or appmark.

Equals and Not Equal

Select the equal (=) or not equal (≠) to change it, defining how Single Sign-on evaluates the value of the attribute.

Value

Specify the value of the attribute that Single Sign-on evaluates when a user accesses the application or appmark.

9 (Optional) Add additional attributes that you want included in this rule set.

10 (Optional) Repeat Step 5 and Step 8 for each additional rule set that you want to create.

11 Select **Done** to create the authorization policy.

12 (Optional) Repeat Step 3 through Step 11 to create additional policies.

13 Select **Done** on the Authorization Policies panel to save the authorization policies that you have created.

The next time a user accesses an application or appmark, Single Sign-on applies the user attribute authorization policies. The authorization service either grants the user access to the application or appmark or displays a message stating the user is not authorized to access the application or appmark.

Understanding the Default Attributes for a Rule-Based Authorization Policy

The authorization service provides a list of default attributes for you to use in the rule-based authorization policies. The attributes the authorization service provides comes from the Advanced Authentication identity repositories. You can customize the list of attributes by changing the configuration of your repositories (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/add_repo.html).

The following lists contains the default attributes for the different identity repository types.

Common Attributes

The following attributes are the same in the SCIM and LDAP repositories.

user_name

Contains the user name.

repo_id

Contains the identifier of the server where the repository that contains the users accounts reside.

user_mobile_phone

Contains the user's mobile phone number if present.

user_email

Contains the user's email address if present.

SCIM Attributes

You can view these attributes when you edit the SCIM repository.

user_name_netbios

Contains the user's NETBIOS name, if the user came from Active Directory.

tenant_name

Contains your tenant name. You only have access to your own tenant information.

user_repository_alias

Contains the user's preferred name.

LDAP Attributes

You can view and edit these attributes through your LDAP repositories administration tools.

user_cn

Contains the user's canonical domain name.

user_sid

Contains the user's SID user_name. For example, the user name COMPANY\JSmith. This attribute is not in eDirectory by default.

user_sid_hex

Contains the user's SID as a hex-string. This attribute is not in eDirectory by default.

user_upn

Contains the user's principal name, if a user came from Active Directory.

user_dn

Contains the user's fully qualified domain name (FQDN).

user_framedIpAddress

Contains a RADIUS attribute for the user that provides network access with a user's IP address before user authentication.

user_first_name

Contains the user's first name if present.

user_last_name

Contains the user's last name if present.

Create a Rule-Based Authorization Policy with Identity Governance Roles

Applications > New Application > Appmark or ***Application*** **> Authorization Policies > Rule-based > Rule Sets + > Identity Governance role**

Single Sign-on allows you to create rule-base authorization policies using Identity Governance authorization (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/b16d32bh.html), business (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/business-roles.html), and technical (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/technical-roles.html) roles. You can limit which users have access to which applications using the Identity Governance authentication, business, and technical roles.

You must create the authorization, business, and technical roles created in Identity Governance as a Service for authorization service to display the roles in the rule-based authorization policies.

To create an authorization policy with Identity Governance roles:

1. Ensure that the user accounts in Advanced Authentication and Identity Governance have an attribute that you match.
2. On the application or appmark that you are creating, select **Authorization Policies**.
3. Select the plus sign (**+**) to add an authorization policy, then select **Rule-based**.
4. Use the following information to create an authorization policy using Identity Governance roles:

   Name

   Specify a name for the Identity Governance authorization policy.

   Enabled

   Select **Enable** to enable the Identity Governance authorization policy after you save it.

   Description

   Provide a detailed description of what the Identity Governance authorization policy does. The description helps people know what the authorization policy does without having to open the authorization policy.
5. At the end of **Rule Sets**, select the appropriate qualifier (**AND**, **OR**) for the authorization policy using Identity Governance roles.
6. Next to **Rule Sets**, select the plus sign (**+**) to create a rule set.
7. Select **Identity Governance role** to create an authorization policy using Identity Governance roles.
8. At the end of **Rules**, select the appropriate qualifier (**AND**, **OR**) for the authorization policy using Identity Governance roles.
9. Next to **User Role**, select **Edit** to expand the available Identity Governance roles to select.
10. Select one of the following roles to include in the authorization policy:
    - **Authorization**
    - **Business**
    - **Technical**
11. Select **Done** to create the authorization policy.

12 (Optional) Repeat Step 8 through Step 11 for each additional Identity Governance role you want to add to this rule.

13 (Optional) Repeat Step 4 through Step 12 to create additional policies.

14 Select **Done** on the Authorization Policies panel to save the authorization policies that you have created.

The next time a user accesses an application or appmark, the authorization service applies the Identity Governance role authorization policies. The authorization service either grants the user access to the application or appmark or displays a message stating the user is not authorized to access the application or appmark.

How the Authorization Policy Using Identity Governance Roles Matches User Accounts

The authorization policies that use Identity Governance roles require that use account attributes have common values in both Advanced Authentication and Identity Governance. You must manually create this mapping through the standard UI to create and manage user attributes.

For the authorization policies using Identity Governance roles to work, it must be able to identify an Identity Governance user. The authorization service does this using the Advanced Authentication attribute values using the following mappings:

Table 10-1 *Attribute Mappings between Advanced Authentication and Identity Governance*

AA Attributes	IG Attributes
email (user_email)	emails
dn (user_dn)	dn
◆ user_first_name ◆ given_name	firstName
◆ user_last_name ◆ family_name	lastName
user_mobile_phone	otherPhone

The above table goes from highest priority on the top to lowest priority on the bottom -- meaning the code will start to collect attributes starting with "emails" working downward until an Advanced Authentication attribute with a value is found.

The authorization service sends a single attribute value to Identity Governance for all the instances above except for "first/last" name. In that case, there must be both a first and last name value in Advanced Authentication and the authorization service sends them together to Identity Governance.

If the authorization service cannot uniquely identify an Identity Governance user, the question, "Does this user have these roles?" cannot be asked and the evaluation result of the policy will be FALSE.

Create an OPA Authorization Policy

Applications > New Application > Appmark or ***Application*** **> Authorization Policies > OPA Policy**

Single Sign-on uses the Open Policy Agent (OPA) (https://www.openpolicyagent.org/) as the authorization policy engine. You can create an OPA policy in Single Sign-on using the OPA policy language . The Single Sign-on allows you to create a Policy Document (https://www.openpolicyagent.org/docs/latest/philosophy/#the-opa-document-model) using Rego (https://www.openpolicyagent.org/docs/latest/policy-language/) and a standard OPA Data Document. The Data Document editor validates JSON, helping ensure that you provide valid JSON for the Data Document.

OPA provides The Rego Playground (https://play.openpolicyagent.org/) that contains simple examples of OPA policies created using the Rego language.

To create an OPA policy:

1. Select the plus sign **+** to add an authorization policy, then select **OPA Policy**.
2. Use the following information to create an OPA authorization policy:

   Name

   Specify a name for the OPA authorization policy.

   Description

   Specify a detailed description so that other administrators can understand what this OPA authorization policy does.

   Enabled

   Select **Enable** to enable the OPA authorization policy after you save it. Single Sign-on ignores any disabled policy when it evaluates the policies.

   Policy Package

   Specify the unique path for your policy package. Single Sign-on appends the path to the end of the Authorization Service's unique policy package path when it sends the Policy Document to the Rego policy engine. You designate the unique path for the policy package by adding the following command to the start of the Policy Document:

   ```
   package{{PACKAGE}}
   ```

   The Authorization Service replaces the `{{PACKAGE}}` tag with the Authorization Service's unique policy package path and the Policy Package you specify here. For example, if you specified the value of the unique path as `your.unique.path`, the result of the final Policy Document would be:

   ```
   authz.svc.unique.path.your.unique.path
   ```

   Every Policy Document must be unique by defining a namespace for the Policy Document using the Rego keyword of `package` at the start of every Policy document. The Authorization Service also must add a unique path to the `package` designation.

   To allow both the Authorization Service and the Policy Document creator to add a namespace to the Policy Document, Single Sign-on uses the following mechanism:

Evaluation Path

Specify the **Evaluation Path** that the Authorization Service uses to evaluate the Policy. The Authorization Service appends this path to the end of the URL used to send the policy evaluation request to the OPA Policy Engine.

You can create a Rego Policy Document with multiple evaluation paths. An evaluation path is defined as a dot-separated path into the Rego Policy Document that, if specified for evaluation, returns only the evaluation for that path. If a Rego Policy Document has many evaluation paths, evaluating the Rego Policy document without specifying an evaluation path executes and returns all the evaluation path results.

For example, in the following Rego Policy Document, the possible evaluation paths are `hello` and `allow`.

```
import future.keywords.if
default hello := false
default allow := false

hello if input.message == "world"
allow if input.message == "universe"
```

If you want to execute the `allow` evaluation point, then in **Evaluation Path** specify the path `"allow"`. Otherwise, in this example Single Sign-on executes the Rego at the `"root"` evaluation path which returns both `"hello"` and `"allow"` as the results.

Policy Document

Create the OPA policy document (https://www.openpolicyagent.org/docs/latest/philosophy/#the-opa-document-model) that contains the OPA Rego policy definition. Standard Rego is required with the following exceptions:

- You must designate the location of the Rego Policy Document policy package by placing the following command at the start of the Policy Document:

  `package{{PACKAGE}}`

- You must designate the location where the OPA Data document namespace must be inserted into the Rego Policy document by placing the following tag anywhere a Data Document path is used. The following example illustrates the tag and its use:

  ```
  allow if
  data.{{ENGINENAMESPACE}}.user_attr[input.user].title ==
  "alloweduser"
  ```

  When the Authorization Service sends the OPA Data Document to the OPA engine, it will namespace the data document using a unique identifier. You must include the auto-generated unique identifier in any Rego Policy Document path referencing the Data Document.

  In the example above, the Authorization Service replaces the tag `{{ENGINENAMESPACE}}` with the auto-generated namespace for the data document. For example:

  ```
  allow if data.authz.svc.path1.user_attr[input.user].title ==
  "alloweduser"
  ```

Data Document

Create the OPA data document (https://www.openpolicyagent.org/docs/latest/philosophy/#how-does-opa-work) that contains the hierarchical structured data in JSON format. This structured data represents static data that the Rego Policy Document uses to evaluate the policy.

The OPA Policy engine uses the Policy Document as the executable script; the Data Document as the static hierarchical structured data; and the Input Document, that is auto generated at run-time, as the dynamic hierarchical structured data, to evaluate the policy.

3 Select **Save** to save the OPA authorization policy.

4 (Conditional) Repeat Step 1 through Step 3 to create additional OPA authorization policies.

Example of an OPA Policy Document

The following is an example of an OPA Policy Document written in Rego.

```
package {{PACKAGE}}

import future.keywords.if

default allow := false

allow if user_is_owner

allow if {
    user_is_employee
    action_is_read
}

allow if {
    user_is_employee
    user_is_senior
    action_is_update
}

allow if {
    user_is_customer
    action_is_read
    not pet_is_adopted
}

user_is_owner if
data.{{ENGINENAMESPACE}}.user_attributes[input.user].title == "owner"

user_is_employee if
```

```
data.{{ENGINENAMESPACE}}.user_attributes[input.user].title == "employee"

user_is_customer if
data.{{ENGINENAMESPACE}}.user_attributes[input.user].title == "customer"

user_is_senior if
data.{{ENGINENAMESPACE}}.user_attributes[input.user].tenure > 8

action_is_read if input.action == "read"

action_is_update if input.action == "update"

pet_is_adopted if
data.{{ENGINENAMESPACE}}.pet_attributes[input.resource].adopted == true
```

The following is the associated Data Document:

```
{
    "user_attributes": {
        "alice": {
            "tenure": 20,
            "title": "owner"
        },
        "bob": {
            "tenure": 15,
            "title": "employee"
        },
        "eve": {
            "tenure": 5,
            "title": "employee"
        },
        "dave": {
            "tenure": 5,
            "title": "customer"
        }
    },
    "pet_attributes": {
        "dog123": {
            "adopted": true,
            "age": 2,
            "breed": "terrier",
            "name": "toto"
        },
        "dog456": {
            "adopted": false,
            "age": 3,
            "breed": "german-shepherd",
            "name": "rintintin"
        },
```

```
            "dog789": {
                "adopted": false,
                "age": 2,
                "breed": "collie",
                "name": "lassie"
            },
            "cat123": {
                "adopted": false,
                "age": 1,
                "breed": "fictitious",
                "name": "cheshire"
            }
        }
    }
```

11 Managing Single Sign-on

Single Sign-on allows you to easily manage the different components. You can update the configuration of the authentications, you can manage the applications, appmarks, and access to the Application Portal.

- "Manage Authentications" on page 67
- "Manage Branding" on page 67

Manage Authentications

Advanced Authentication

You can add or remove Advanced Authentication chains, methods, or repository any time that you need. These are the items that you configure to enable authentications in the Micro Focus SaaS environment. If you have a new user repository in your environment, you would create a new repository for it so that Single Sign-on support these new users as well.

Manage Branding

Advanced Authentication > Policies > Custom Branding

We allow you to change the default branding to use custom branding for the Applications administration pages and the Applications portal. You configure custom branding (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/custombranding.html) in Advanced Authentication, and Single Sign-on inherits the custom branding. You do not have to configure the custom branding multiple times.

Single Sign-on does not support the branding option of **Use Custom Branding File for Web Authentication** and the suboption **Web Authentication Branding File**

If you enable **Use Custom Branding File for Web Authentication**, Single Sign-on does not apply the custom branding file to any web authentication (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/web_auth.html) pages Single Sign-on displays to the users.

12 Manage Applications and Appmarks

Applications

You manage the applications and the appmarks on the Applications page in the Micro Focus SaaS environment. You can create, delete, edit, sort, and manage the status of the applications on the Applications page. The Applications page also allows you to create and manage appmarks independently from an application.

- "Create an Application or an Appmark" on page 69
- "Make an Application a Favorite" on page 70
- "Filter Applications" on page 70
- "Edit an Application" on page 70
- "Manage the Client Secret for an OAuth Application" on page 71
- "View the Status of an Application" on page 71
- "Change the Application Tile Size" on page 71
- "Change the Settings for the Most Recent Applications" on page 72
- "Troubleshooting Issues" on page 72

Create an Application or an Appmark

Applications > New Application

Single Sign-on allows you to create **applications** that provides single sign-on and secure access to web-based applications, external services (software as a service), and federated business-to-business interactions. It allows you create applications in different ways:

- **Appmark:** Single Sign-on provides the ability to create appmarks independently from an application. This ability allows you to create additional appmarks to separate endpoints for the external services.
- **Application:** Single Sign-on provides applications for frequently used services that allow single sign-on. The applications simplify the configuration of creating a secure, federated connection to external service.
- **OAuth:** Single Sign-on provides the ability to create applications for any external OAuth service.
- **SAML:** Single Sign-on provides the ability to create applications for any external SAML service.
- **SAML Identity Provider:** Single Sign-on provides the ability to create applications for any external SAML identity provider.

Make an Application a Favorite

Applications > *A specific application*

Single Sign-on allows you to make an application a favorite so that you can easily sort on the applications you access the most.

To make an application a favorite:

1. On the application, select **Actions** in the upper right corner.
2. Select **Make Favorite**.

Filter Applications

Applications > *Filter Icons*

By default, the applications are sorted alphabetically. You can select the Filter icon and select to only display OAuth applications, SAML applications, SAML identity providers, or appmarks. You can change the sort order by selecting the different sort options in the upper right corner of the Applications page. You can sort by applications that need attention, favorite applications, and applications you accessed recently.

Edit an Application

Applications > *A specific application* > Actions > Edit

When you create an application, Single Sign-on automatically creates an event for the application in Advanced Authentication. If you manually edit the event for the application, Single Sign-on does display those changes. Next, if you edit the application and save the changes in the Single Sign-on UI, Single Sign-on overwrites the changes you made in Advanced Authentication with these new changes.

It is important to edit the applications in Single Sign-on. Single Sign-on automatically updates the event for you. Single Sign-on allows you to edit and change any settings on an application at any time.

To edit an application:

1. On the application that you want to edit, in the upper right corner, select **Actions**.
2. Select **Edit**.
3. Make the appropriate changes to the application.
4. (Conditional) If you used the instructions in the applications, you can edit the metadata XML file.
   - **4a** Select **Edit Metadata XML** to edit the metadata in XML format.
   - **4b** In the new window, select **Edit Metadata** to edit the metadata.

4c (Conditional) If you do not want to use the edits you made, select **Use Metadata File** to revert back to the initial creation inputs. The manual edits are not saved.

4d Select **Done** to save the changes.

5 Select **Save** to save your changes.

Manage the Client Secret for an OAuth Application

Applications > *An OAuth Application* > Client Secret > Reset

Single Sign-on allows you to reset the **Client Secret** for the OAuth applications. Public clients do not use client secrets. In **OAuth Advanced Settings**, if you enable **Public Client**, Single Sign-on automatically removes the client secret from the OAuth application. Single Sign-on generates a new client secret whenever you save the OAuth application if there is no client secret. The only exception is when you enable public clients.

To reset an OAuth client secret:

1 On the Applications page, select the appropriate OAuth application to edit.

2 At the end of **Client Secret**, select **Reset**.

3 Select **Save** and Single Sign-on generates a new client secret for the OAuth application.

View the Status of an Application

Applications > *A specific application* > Actions > Application Summary

Single Sign-on provides an Application Summary page for an application. The Application Summary page displays the status of the application that allows you to validate the configuration of an appmark and to complete the federation connection to the external service. The Application Summary page displays all of this information in one location for you to easily access the information.

To view the status of an application:

1 On the appropriate application, select **Actions > Application Summary**.

2 View the status of the application.

Change the Application Tile Size

Applications > More View Options > *Select tile size*

Single Sign-on allows you to change the tile size of the applications to what functions best for you.

To change the tile size:

1 In the upper right corner of the Applications page, select **More View Options**.

2 Select the appropriate tile size of **Large Tile**, **Medium Tile**, or **Small Tile**.

Change the Settings for the Most Recent Applications

Applications > More View Options > Settings

Single Sign-on shows five of the most recently viewed applications by default. You can change how many of the most recently viewed applications you want to appear.

To change the number of most recently viewed applications:

1. In the upper right corner of the Applications page, select **More View Options > Settings**.
2. Change the number of recently accessed applications to display.
3. Select **Save**.

Troubleshooting Issues

Use the following information to help trouble issues with the Single Sign-on applications.

- "Troubleshooting the Salesforce Application" on page 72

Troubleshooting the Salesforce Application

Issue: If you have Single Logout (SLO) enabled in Salesforce, but you did not provide the logout URL in the Salesforce application configuration, Single Sign-on does not terminate the session to Salesforce when a user logs out of the Application portal.

Solution: If you enable Single Logout (SLO) in Salesforce, provide the logout URL. If you do not have SLO enabled in Salesforce, do not provide a logout URL during the configuration of the Salesforce application.

A Understanding Secure Communications

Single Sign-on uses common industry standards to secure communication such as X.509 certificate, public key infrastructure (PKI), and transport layer security (TLS). You must decide which security mode that you will use during the deployment of the Single Sign-on products and the infrastructure products.

- "Understanding the Public Key Infrastructure and TLS Components to Establish Secure Communication" on page 73
- "Example of Establishing Secure Communication for a Web Server" on page 75
- "Example of a Secure Handshake for the Client" on page 77

Understanding the Public Key Infrastructure and TLS Components to Establish Secure Communication

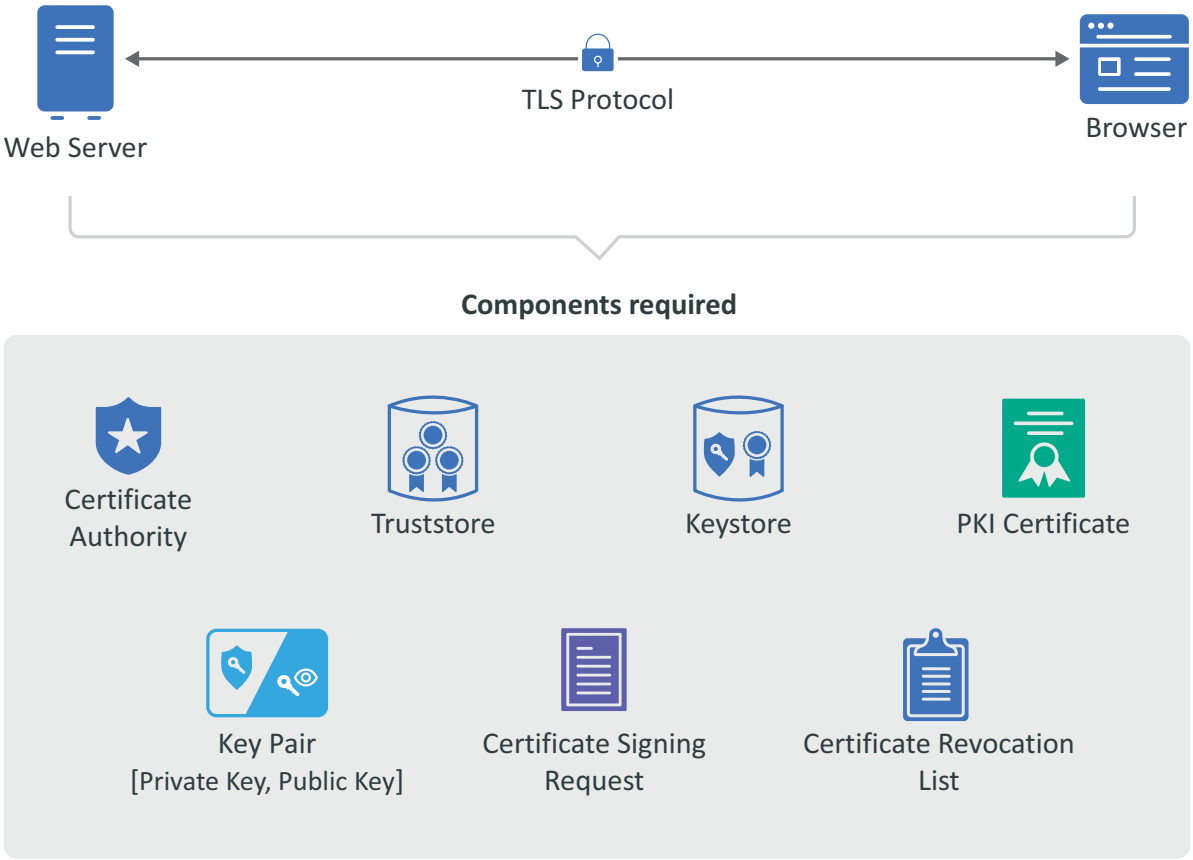
To be able to configure secure communication between Single Sign-on and applications that require secure communication, requires that you have a good understanding of the components that allow the secure communication to occur. Single Sign-on uses industry standards of X.509 certificates, public key infrastructure (PKI), and transport layer security (TLS).

This section provides a basic introductions to these components. For more detailed information, see:

- Internet X.509 Public Key Infrastructure Certificates and Certificate Revocation List (CRL) Profile (https://datatracker.ietf.org/doc/html/rfc5280)
- The Transport Layer Security (TLS) Protocol Version 1.2 (https://datatracker.ietf.org/doc/html/rfc5246)
- The Transport Layer Security (TLS) Protocol Version 1.3 (https://datatracker.ietf.org/doc/html/rfc8446)

You want to secure the communication channels between servers and clients to protect your data and stop security breaches from happening in your environment. The following graphic depicts the different components required for secure communication using certificates, PKI, TLS, and tools to manage the keys.

Figure A-1 *Components of Secure Communications*



The secure communication occurs between a server and a client. In the graphic, that is a web server and the browser is the client. The following items are the terms that you need to understand to create a secure connection between a server and a client.

- **Certificate Authority:** It is an entity that issues digital certificates. A certificate authority (CA) acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate. There are two different types of CAs.
  - **Well-known:** It is certificate authority that provide server certificates signed by well-known CAs such as IdenTrust or DigiCert.
  - **Self-signed:** It is a certificate authority that other products such as openSSL, eDirectory, and Active Directory that contain a certificate authority. You can create self-signed certificates through the certificate authorities in these other products to use in test environments.

  A security recommendation is to use a well-known CA to issue certificates in productions environments.
- **Public Key Infrastructure (PKI) Certificates:** Are are digital certificates that the CA issues that prove ownership of the certificate. The CA can issue certificates for users, applications, or devices. The PKI certificates contain the following information:
  - Version number
  - Unique serial number
  - CA digital signature and algorithm used

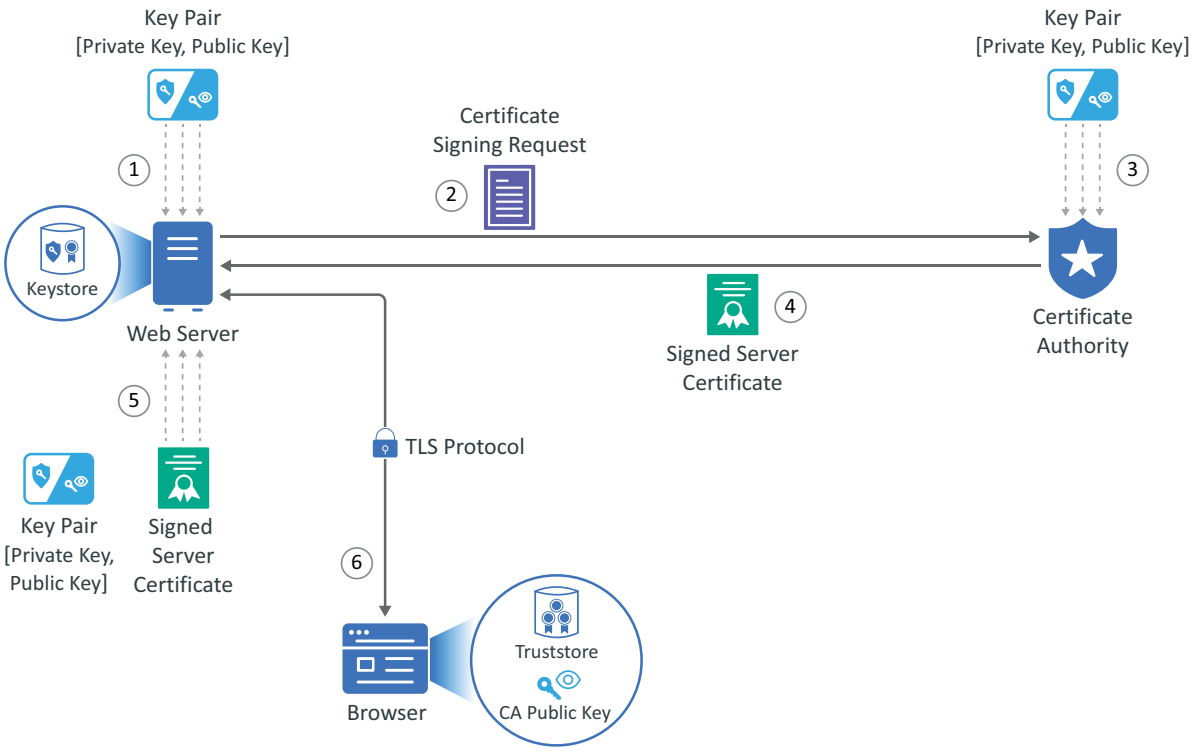
- Validity period
- Certificate Usage
- Subject name, URL, email address
- Public and private keys (sometimes it is only the public key)

- **Key Pair:** Consists of a private key and public key that work together to encrypt and decrypt messages. PKI is based on the fact that everyone will trust any communication encrypted with a public key or trust any certificate signed by a private key.
  - **Private Key:** It is an cryptographic key that you use it to decrypt any communication encrypted by the public key. Only the private key of the key pair can decrypt the communication encrypted with the corresponding public key. You keep the private key private and do not share it.
  - **Public Key:** It is a cryptographic key that you use to encrypt communications to keep the communication secure. Only someone with the private key can decrypt the communications. You share the public key so that anyone with access to the public key can verify that any communication signed with this public key is really from the sending source.
- **Certificate Revocation List:** It is a list that the CA creates and manage that contains a list of unique serial numbers that it has revoked. The CA uses the certificate revocation list (CLR) to denied requests from any user, application, or device that contain a serial number on the CLR.
- **Certificate Signing Request:** It is a message sent from an applicant to the CA to apply for a PKI certificate. Usually the certificate signing request (CSR) contains a copy of the public key of the applicant making the request, identifying information such as a domain name, and a digital signature.
- **KeyStore:** It is a secure Java repository that stores the private key and identity certificate for the server in the trust relationship. The information is stored encrypted on the server with a KeyStore password that you set and manage. Use either the keytool or keytoolgui tools to set and manage the KeyStore passwords.
- **TrustStore:** It is a secure Java repository that stores the certificates signed by a CA in a secure repository on the client. The information is stored and encrypted on the client with a TrustStore password that you set and manage. Use either the keytool or keytoolgui tools to set and manage the TrustStore passwords.
- **Transport Layer Security Protocol:** It is the secure protocol created by all of the components defined in this section. It allows the server and client to communicate securely by using certificates and key pairs to prove identity on the server and client.

Example of Establishing Secure Communication for a Web Server

When you install a web server, the communication is not secure by default. Or if the communication is secure, it is usually using a self-signed certificate. The following example shows how the web server obtains a server certificate signed by a well-known certificate authority (CA) to use in establishing secure communications with any client.

Figure A-2 *Obtaining a Signed Server Certificate from a Well-known Certificate Authority*



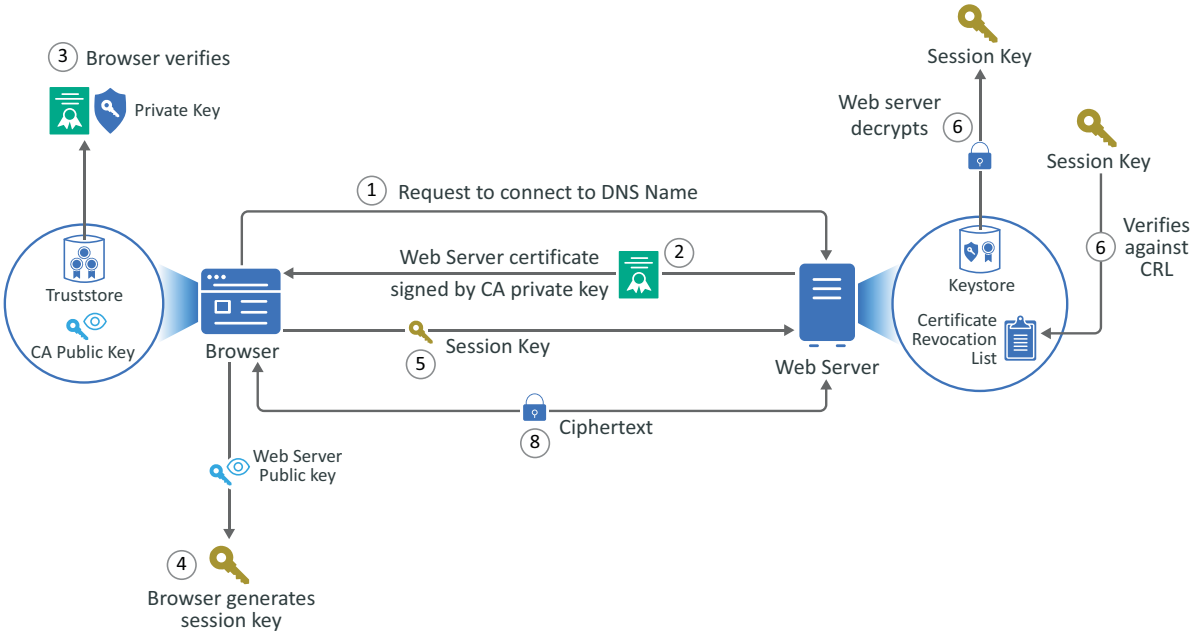
The example is of Adam the administrator requesting a signed server certificate from the well-known CA and using the certificate to establish secure communications with a client that is a web application.

1. Adam generates a key pair on the web server using keytool. Adam use the key pair to create a certificate signing request (CSR) using keytool. The CSR contains the fully qualified DNS name of the server, the key pair, and other such information to help identity the web server.
2. Adam sends the CSR that contains the web server's information to a well-known CA such as DigCert.
3. The CA uses the CSR to generate a server certificate for the web server. The CA uses it private key to sign the certificate. The server certificate contains the key pair and the web server's information included in the CSR. The CA signs your certificate with its private key.
4. The CA sends the signed web server certificate back to Adam.
5. Adam imports the signed web server certificate into the web server and the web server's certificate and private key are stored in the KeyStore on the web server.
6. When a browser access the web server, the web server sends a certificate signed by the private key of the CA to the browser. The browser has a copy of the CA's public key in its TrustStore and uses the public key to decrypt the signature of the CA. Now, the browser knows to trust any communication coming from this web server. For more information, see Example of a Secure Handshake for the Client.

Example of a Secure Handshake for the Client

The web server example shows how a sever receives a certificate from a well-known certificate authority (CA) to be able to communicate securely with any client. This examples show how the secure handshake occurs between a client and a server so that they can create their own secure communication channel that no other entities can uses or access.

Figure A-3 *A Browser Establishes a Secure Communication Channel to a Web Server*



The example is of Adam the administrator logging into the administration console that is a web application. Every action except for Adam entering the URL of the web browser happens automatically between the browser and the web server. No user interaction is required.

1. Adam adds a the URL into the browser. The browser sends a request to connect to the fully qualified DNS names of the web server.
2. The web servers sends a copy of its server certificate that has been signed by the private key of a well-known CA.
3. The browser access the public key of the well-known CA that is stored in the browser's TrustStore. The browser uses the public key of the well-known CA to decrypt the signature on the web server's certificate to verify that the certificate is valid.
4. The browser generates a session key using the public key in the web server's certificate.
5. The browser sends the newly generated session key back to the web server.
6. The web server uses its private key stored in the KeyStore to decrypt the session key.
7. The web server verifies that the session key is not on the certificate revocation list (CLR). At this point the secure handshake between the browser and web server is established.
8. The web server encrypts the data using the session key and sends the data back in ciphertext to the browser. The browser uses the session key to decrypt the data and then uses the session key to encrypt data and then it sends the data back in ciphertext. This secure communication continues until the session ends.