CUE User Manager

User Guide

1.0.4-3

Table of Contents

1 Introduction

CUE User Manager is CCI / Escenic's new solution for managing user access to CUE. It provides a standard interface to a centralized **Identity and Access Management (IAM)** system that can be shared by all CUE-related back-end systems. Instead of the CUE Content Store having its own integrated user management functionality, it can delegate responsibility for access control to CUE User Manager.

The use of such a centralized IAM system has a number of advantages:

- Single sign-on for users
- Simpler user management: no problems keeping identities consistent between systems, no redundancies or duplication
- Simpler auditing: all user activities are reported by one system in a single, consistent format
- Reduced resource usage
- Deal with upgrades to the IAM system: any changes in the IAM system's API will only affect CUE User Manager rather than requiring changes in all back-end systems.

CUE User Manager needs an IAM system that provides the following user management tasks:

- **Authentication**: is the user who he says he is?
- **Authorization**: what is this user allowed to do?
- **Identity management**: what information should be stored for each user (mail address, profile and so on).

IAM systems can be configured either to act as a complete, standalone IAM system or to co-operate with other authentication systems such as Active Directory, Google and Facebook.

CUE Content Store and CUE Print are already able to make use of a common IAM system (Active Directory) to enable single sign-on for users, so what benefit does CUE User Manager offer? It can best be seen as an insulating layer between the CUE back-end systems and the IAM system, making it easier to deal with upgrades to the IAM system: any changes in the IAM system's API will only affect CUE User Manager rather than requiring changes in all back-end systems. In theory it is possible to use any IAM system that supports:

- OpenID Connect Core 1.0 for authentication.
- OpenID Connect Discovery 1.0 for service discovery.
- System for Cross-domain Identity Management 2 (SCIM 2) for identity management.
- User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization for web resource authorization.

2 Getting Started

This chapter contains step-by-step instructions for installing, configuring and running CUE User Manager.

2.1 Repository Code Names

CCI Europe is in the process of introducing **repository code names** to help with the installation of CUE software. A repository code name is an easy to remember word used to name an APT repository. A repository that has been given a code name contains packages for a set of product versions that are guaranteed to be inter-operable. Instead of having to remember a lot of version numbers and check them for compatibility with one another, you can just use the same code name when installing all the components of your system, and be sure that they will work together.

An alphabetic sequence is followed when choosing code names, so a repository with the code name **`eyre`** can be assumed to contain later versions of products than a repository with the code name **`dedman`**.

If you have been given a code name for installing CUE software, then you should use it as follows when adding CUE repositories to your list of sources:

```
# echo "deb https://user:password@apt.escenic.com code-name main non-free" >> /etc/
apt/sources.list.d/escenic.list
```

If your code name is **`eyre`**, for example, then you should enter:

```
# echo "deb https://user:password@apt.escenic.com eyre main non-free" >> /etc/apt/
sources.list.d/escenic.list
```

This ensures that all products installed using the **`apt`** command will be installed from the **`eyre`** repository.

If you haven't been given a code name, then use the default name **`stable`**. This will add the most recent stable CUE APT repository to your list of sources.

2.2 Installation Procedure

To install the CUE User Manager on an Ubuntu or other Debian-based system, do the following:

1. Log in as **`root`**.

2. If necessary, add the CUE APT repository to your list of sources:

```
# echo "deb https://user:password@apt.escenic.com code-name main non-free" >> /
etc/apt/sources.list.d/escenic.list
```

   where:

   - *user* and *password* are your CUE download credentials (the same ones you use to access the CUE Maven repository). If you do not have any download credentials, please contact CUE support.
   - *code-name* is either a repository code name supplied to you by CCI Europe or, if no code name has been supplied, the default name **`stable`**. For further information, see section 2.1.

3. Enter the following commands:

```
# apt-get update
# apt-get install cue-user-manager
```

On RedHat / CentOS systems, enter the following command as **`root`**:

```
# rpm -Uvh https://user:password:yum.escenic.com/rpm/cue-user-
manager-1.0.4-3.x86_64.rpm
```

You should now have CUE User Manager installed. The next step is to secure it with TLS, see section 7.6 and then to integrate it with your IAM system, see the chapter 7.

2.3 Configuration

To configure CUE User Manager you must go to the IAM system and create an OIDC client. After selecting your secret passphrase, it will give you an OIDC client ID. CUE User Manager will use this client ID and secret for all authentication services against the IAM system. You can now add the client ID and secret, together with the OIDC discovery endpoint and SCIM endpoint to **`/etc/escenic/user-manager/user-manager.yaml`**:

```
provider:
  oidcEndpoint: https://iam.mycompany.com/.well-known/openid-configuration
  clientId: my-um-oidc-id
  clientSecret: foo-bar-baz
  scimEndpoint: https://iam.mycompany.com/scim/v2/
```

The endpoint URIs should be a part of the IAM system documentation.

The configuration file has many more options, but the ones above should be enough to get you started. The configuration file has lots of helpful comments and examples. See the section on user-manager.yaml (section 4.1) for further documentation.

2.4 Starting and Stopping

To start CUE User Manager, enter:

```
# /etc/init.d/user-manager start
```

In addition to **`start`**, the **`init.d`** script has these options for controlling CUE User Manager

stop
To stop CUE User Manager.

restart
To restart CUE User Manager.

status
To list status information about CUE User Manager.

3 Daily Use

3.1 Logging in to CUE

Test & staging environments

When CUE User Manager is available, CUE Editor will add a link to the login screen that says **Log in with User Manager**:

This takes you to the IAM system's login page, where you then enter your login credentials (Active Directory credentials, for example):

After logging in here, you will be returned to CUE.

Production environments

A link on the login screen is fine for test environments or when you want to have the possibility to login using users that only exist in CUE Content Store, however for production environments, we would normally recommend that the user is taken directly to the IAM system's login screen, rather than first having to click on a link in the CUE Editor login screen.

To make CUE Editor go directly to the IAM login screen, set the following in **`/etc/escenic/cue-web/config.yml`**:

```
redirectToLoginPage: true
```

See the CUE User Guide for further information on how you configure CUE Editor.

4 Configuration files

4.1 user-manager.yaml

The User Manager Java application itself is configured with **`/etc/escenic/user-manager/user-manager.yaml`**. This section will describe the most important settings, for a full list of available options, you can read the file and the comments therein.

Configuration items under the **`provider`** YAML block:

`oidcEndpoint`
The OIDC discovery endpoint of your IAM, e.g.: **`https://iam.mycompany.com/.well-known/openid-configuration`**

`clientId`
The id of the OIDC client you've created in the IAM system. CUE User Manager uses this when authenticating with the IAM system.

`clientSecret`
The secret of the OIDC client you've created in the IAM system.

`redirectURI`
Redirection URL after login through OIDC. This needs to be same as you configured in the IAM system when creating the OIDC client.

Example: **`https://iam.mycompany.com/successful-login`**

`scimEndpoint`
The SCIM2 endpoint of your IAM, e.g.: **`https://iam.mycompany.com/scim/v2/`**

`groupSync`
If you've set up your IAM system to sync from AD and it doesn't sync groups too, you can set this to **`true`** to have CUE User Manager create these users in the IAM system for you. The pre-requisite for this to work is that the groups are present in a string field called **`userGroups`** on the SCIM user object.

Newsroom publication mapping

CUE User Manager uses pattern matching to find the name of the newsroom and then utilizes predefined mappings between newsroom and publication to find out which roles to assign to the user in what publication.

Valid identifiers for groups are **`${publication}`**, **`${newsroom}`**, **`${name}`** and **`${ignore}`**. If the group identifier is **`${ignore}`**, the SCIM user objects must have the field **`homePublication`** set.

If the AD group name is **`newroomx_journalist`** where the name of the newsroom is **`newsroomx`** and the role the user should have in this newsroom is journalist, then the template should be **`${newsroom}_${name}`**.

```
newsroom:
  # publicationMapping:
    # sportsdesk:
        # Corresponding publication names
```

```
            # - football.com
            # - cricket.com
            # - golf.com
        # tabloid:
            # - beats.com
```

If using **publicationMapping**, set the template to:

```
${newsroom}_${name}
```

Default template is:

```
${ignore}_${name}
```

The mapping between the **name** fragment of the group and the roles this group should get in CUE Content Store is defined in the **roleMapping** block:

```
roleMapping:
  reader:
    - reader
    - articleWithContentTypeReader
  journalist:
    - journalist
    - reader
    - articleWithContentTypeWriter
    - articleWithContentTypeReader
  editor:
    - editor
    - journalist
    - reader
    - articleWithContentTypeWriter
    - articleWithContentTypeReader
  admin:
    - publicationadmin
    - useradmin
    - administrator
    - editor
    - journalist
    - reader
    - articleWithContentTypeWriter
    - articleWithContentTypeReader
```

When AD user group naming is consistent and corresponding CUE User Manager mapping is correct, users will get correct permissions in CUE Content Store automatically, making AD not only the master source of users but also of authorization/access control to the different publications inside CUE Content Store.

4.2 user-manager.conf

This file configures the **/usr/bin/user-manager** command. You will probably never need to edit this file, but if in case you want to, you edit this in the same way as user-manager.yaml (section 4.1).

5 Architecture

The blue dotted line defines what's the full CUE User Manager solution, whereas the green dotted line defines what's a part of the IAM backend which UM uses.

The green boxes are the CUE User Manager components.

CUE
The CUE editor running in a web browser.

NG
CCI Newsgate

NGINX
NGINX listening on port **80** and **443** (**https**).

UM
CUE User Manager. A Java micro service. Provides a REST interface which CUE, Content Store and NG uses to authenticate users and get user information.

OIDC
Server having endpoints for OpenID Connect Core 1.0 and OpenID Connect Discovery 1.0

SCIM
Server having endpoints for SCIM 2

LDAP
Internal user/group storage for the IAM system. Typically syncs (pulls) from AD, but can also be used standalone.

6 Diagnostics

CUE User Manager provides diagnostic information about itself that you can use to monitor the application's status. The diagnostic information is available in the form of JSON data that can be retrieved from two web application endpoints.

6.1 Health Check

CUE User Manager provides a "health check" at the following URL:

```
https://umhost:8681/healthcheck
```

where *umhost* is the host name or IP address of the machine on which you installed CUE User Manager. If you are logged in on that machine, then you can get nicely formatted output on the command line as follows:

```
$ curl https://localhost:8681/healthcheck | jq .
```

The health check provides information about the current state of both the CUE User Manager itself and the associated IAM manager. For example:

```
{
 "deadlocks": {
   "healthy": true
 },
 "provider-1-gluu": {
   "healthy": true,
   "message": "Connected"
 },
 "provider-1-stateoriginmappercache": {
   "healthy": true,
   "message": "CacheState(size=1)"
 }
}
```

6.2 Metrics

CUE User Manager provides a range of application metrics at the following URL:

```
https://localhost:8681/metrics
```

where *umhost* is the host name or IP address of the machine on which you installed CUE User Manager. If you are logged in on that machine, then you can get nicely formatted output on the command line as follows:

```
$ curl https://localhost:8681/metrics | jq .
```

The information provided at the **`metrics`** endpoint includes:

- System memory: total and free system memory, in KB

- System uptime: both server uptime and application context uptime
- JVM memory usage: initial, used and maximum available heap and non-heap sizes
- Thread statistics: total thread count, number of threads started and number of daemon threads
- Classloader: total classes loaded, number currently loaded and number unloaded
- Garbage collector: garbage collection algorithm, number of times run and time taken on last garbage collection run
- HTTP connections: total number of connections and total active connections
- Data source
- Timers: total, maximum and minimum duration of HTTP requests (all types)

Most of the information in the response is fairly easy to understand and provides a detailed view of the CUE User Manager's current performance.

7 IAM Integration Guide

This chapter will outline the different components of an IAM system that CUE User Manager needs to function properly: OpenID Connect Core 1.0 (authentication), OpenID Connect Discovery 1.0 (service discovery), System for Cross-domain Identity Management 2 (SCIM 2) (identity management) and User-Managed Access 2.0 (UMA) for web resource authorization.

The chapter will also describe the typical scenario of having users in an LDAP server, typically AD and using these users in the IAM, CUE User Manager and ultimately CUE Content Store, Newsgate and CUE Editor.

7.1 OpenID Connect

OpenID Connect (OIDC) is ubiquitous these days and must be supported in the IAM system you want to use. Note that OpenID Connect is not the same as OpenID 1.0 or OpenID 2.0. The IAM system must also support OpenID Connect Discovery 1.0 which gives you a URL which lets a client discover all the OIDC related services, including where to request an access token and where to refresh it.

Granted that the IAM system supports these two standards, CUE User Manager should be able to use it by entering the service discovery URI in **`/etc/escenic/user-manager/user-manager.yaml`**:

```
provider:
  oidcEndpoint: https://iam.mycompany.com/oidc-discovery
```

7.2 System for Cross-domain Identity Management 2 (SCIM)

System for Cross-domain Identity Management 2 (SCIM) is a standard for identify management. It describes the various resources (user, group++) and a REST interface to on these. CUE User Manager requires the IAM system to support version 2 of SCIM.

CUE User Manager needs the following extensions must be present in the SCIM user schema:

- userGroups (multi value **`string`**)
- homePublication (**`string`**)

Granted that the IAM system supports SCIM and that a valid OIDC access token passed in the request grants access to the REST API, CUE User Manager should be able to use it by entering the SCIM2 REST endpoint URI in **`/etc/escenic/user-manager/user-manager.yaml`**:

```
provider:
  scimEndpoint: https://iam.mycompany.com/scim/v2/
```

7.3 User-Managed Access 2.0 (UMA)

By default, CUE User Manager assumes the the SCIM web resources are UMA protected:

```
provider:
  umaProtectedSCIM: yes
```

When enabled, CUE User Manager will perform the UMA authorization flow before making the actual SCIM HTTP request.

It's up to the IAM provider to validate that the access token is valid.

If the SCIM endpoint is **not** UMA protected, you must set this to **`false`**. The SCIM endpoint will then receive requests from with CUE User Manager with **`Authorization`** headers like:

```
Authorization: Bearer <access-token>
```

Again, it's up to the IAM system to validate that the access token is valid before allowing the client to perform the requested SCIM operation.

7.4 AD synchronisation

A nice feature of CUE User Manager is that you can instantly login into CUE Editor using the users you have in AD. For this to work, the users and their groups must be synced from AD to the IAM system. CUE User Manager will then in turn on demand, create these users and groups inside CUE Content Store if they don't already exist.

How the IAM systems copies users and groups from AD is system dependent and you need to look into the appropriate documentation on this. Typically, you want the IAM system to pull the data from AD, leaving AD oblivious to the existence of the IAM system.

If the IAM doesn't sync groups

Some IAMs, like Gluu, will not sync the groups from AD, in which case you must ensure the group IDs are present as a list of strings in the SCIM user object's **`userGroups`** field. This field is a custom extension that you must add to your SCIM user model, see section 7.2.

You must be sure to set up the AD synchronisation to map from AD's memberOf attribute to the IAM LDAP user's **`userGroups`** field and that this field is of a multi value string type.

CUE User Manager can now create native IAM user groups using this **`userGroups`**field, see the groupSync configuration in user-manager.yaml (section 4.1).

7.5 AD Authorization

Authorization through AD, i.e. that the user can use his AD password to login to CUE Editor, is in the domain of the IAM system and its configuration. All such systems will all have a way of forwarding the authentication challenges to AD. But again, this configuration is system specific and you must read the IAM vendor's documentation on this topic.

7.6 Secure all endpoints

All LDAP and HTTP endpoints that make up the CUE User Manager & IAM system should be protected with TLS. It's worth creating a good strategy for how to create and maintain such certificates for all environments, including development, test and staging. This way you can ensure that all environments are as production-like as possible. Turning off CUE User Manager in non-production systems is not advisable as problems with login, single sign on and authorization will then first be discovered in production.

As a starting point the following endpoints should be secured with TLS:

- The ODIC discovery document and all URIs it lists: **`https://iam.mycompany.com/.well-known/openid-configuration`**
- The SCIM endpoint and all its URIs: **`https://iam.mycompany.com/scim/v2/`**
- CUE User Manager itself: **`https://um.mycompany.com`**
- The LDAP server holding the master source of users and groups. In many cases, this will mean Active Directory.
- The IAM system will have its own storage of users, and the communication between the OIDC and SCIM services and that storage should also go over TLS.

  For instance, Gluu's internal storage for users and application configuration is an LDAP server. This server has its own TLS certificate to ensure that communication between the SCIM component (called **`oxTrust`** or **`identity`**) and the OIDC component (called **`oxAuth`**) and this storage backend can be performed securely.

7.7 Two factor authentication (2FA)

Two factor and multi factor authentication are both in the sole domain of the IAM system and has nothing to do with the CUE User Manager integration. If the IAM system is set up to use 2FA, it will do so regardless of what CUE User Manager tells it.

7.8 Google SSO

Single sign on using the user's Google identification is in the sole domain of the IAM system and has nothing to do with the CUE User Manager integration. If the IAM system is set up to use Google OIDC login , it will do so regardless of what CUE User Manager tells it.

That said, many IAMs, like Gluu, will support users logging in using **either** an AD user **or** a Google user. This means it's possible to separate users, for instance, having in-house editorial users in Active Directory, while freelancers use their Google login.

When the user logs in through Google SSO, a (shadow) user is created inside the IAM system with a link to the Google identity, e.g. in Gluu, the LDAP entry for the user logged in over Google OIDC contains an attribute like **`oxExternalId=gplus:4112343241234`**. Before this user can do anything in CUE User Manager it must be assigned groups matching the newsroom publication mapping configuration that you've configured in **`/etc/escenic/user-manager/user-manager.yaml`**. CUE User Manager uses this to create the appropriate user in CUE Content Store with roles granting it permissions to **do** something. How these groups are added is in the realm of the IAM system.

Preferably, the IAM system allows the integrator to hook onto the IAM system's user creation process and add the necessary groups immediately after the user logs in through Google, ensuring a smooth single sign on experience.

7.9 Migrating existing Escenic users

In cases where you have an existing user base in the CUE Content Store database and these don't exist anywhere else (i.e. the CUE Content Store database is the master source for these users), you can migrate these to the IAM system using CUE User Manager's REST interface. This will typically be a one time job to bootstrap the system.

The CUE User Manager is documented "live" using the OpenAPI Specification and is accessible on your CUE User Manager system under the URI **`https://um.mycompany.com/openapi.json`**. There's also an interactive web interface through which you can experiment with the API under **`https://um.mycompany.com/swagger-ui/`**

Here's an example of creating a user using CUE User Manager's REST interface:

```
$ curl \
  --request POST \
  --header 'Content-type: application/json' \
  --header 'Authorization: Bearer <access-token>' \
  --data '{
     "userName": "lisa",
     "givenName": "Lisa",
     "familyName": "Doe",
     "displayName": "Lisa Doe",
     "email":"lisa@example.com"
    }' \
  https://um.mycompany.com/user
```

The user data can be exported from the CUE Content Store in several ways. Check out the documentation for the version you have installed at: cuedocs.escenic.com. Once you have them on a machine readable format, you can write a script to create these users in the IAM system by using the mentioned CUE User Manager REST interface. These users will be created in the IAM system through the SCIM2 interface. If these users then are to be further exported to Active Directory, the appropriate LDAP synchronisation must be set up. Again, this is in the realm of the IAM system.

The code you write to migrate the users will need a long lived access token from the IAM system (since the code is not a person that can authorise requests in a web browser). You thus need to configure the IAM system to issue access tokens with a long TTL, e.g. 30 minutes and then obtain one on the behalf of your integration script by pointing your web browser at: **`https://um.mycompany/auth`** . At the end of the OIDC authentication exchange, you will see the access token in the URI in your browser. This token can then be used in requests to the **`/user`** endpoint of CUE User Manager to create the users and groups you need.