SYNTHESYS+

Synthesis of Systematic Resources a DiSSCo project

D6.2 Piloting Access Through an AAI infrastructure

Sam Leeflang (Naturalis), Nicolas Liampotis (GRNET), Costas Georgilakis (GRNET), Sharif Islam (Naturalis)

Grant Agreement Number | 823827

Acronym | SYNTHESYS PLUS

Call | H2020-INFRAIA-2018-2020

Start date | 01/02/2019

Duration | 48 months

Work Package | 6

Work Package Lead | [W Addink]

Delivery date | [30 Jan 2023]

Contents

Summary

This document describes the background of the authentication and authorization mechanism piloted for the European Loans and Visits System (ELViS). It details the technical setup of the Authentication and Authorization Infrastructure (AAI) that was used during the SYNTHESYS+ Transnational Access (TA) and Virtual Access (VA) calls in 2021 and 2022. This is followed by an outline of the rationale behind the pilot activities, a detailed technical setup, a summary of code changes, an evaluation of the pilot and a list of future activities to use the pilot results further in the future DiSSCo production environment.

1. Introduction

European Loans and Visits System (ELViS) along with other DiSSCo services needs an authentication and authorization mechanism that is based on open standards and complies with current community standards followed by European Research Infrastructures. We also want a system that is user-friendly and easy to maintain. An Authentication and Authorization Infrastructure (AAI) provides such capabilities. Based on these principles, one of the goals of SYNTHESYS+ was to run a pilot where the European Loans and Visits System (ELViS) is accessed through a generic AAI infrastructure. ELViS was developed with its own Identification and Access Management (IAM) tooling. During the pilot, ELViS was disconnected from its current IAM tooling and connected to an AAI infrastructure provided by GRNET. The result of this pilot is described in this document. We also discuss the advantages of connecting to generic AAI services in this document and the findings we made during piloting. Concluding the deliverable is an evaluation of the requirements for the AAI as proposed in SYNTHESYS+ Milestone 48.

2. AAI Setup

In this chapter we describe the different setups between the initial ELViS implementation and the piloted AAI implementation to be used in the future DiSSCo infrastructure. This will help clarify the differences between the two setups.

Initial setup used during the SYNTHESYS+ TA and VA calls

ELViS is hosted on a single server where all necessary applications are deployed as Docker containers. As IAM tooling, it uses Keycloak, which has its own Postgres database container. All traffic between the applications can be kept internal as all the applications run on the same machine. All applications and data are deployed and managed by the development team (see Fig. 1).

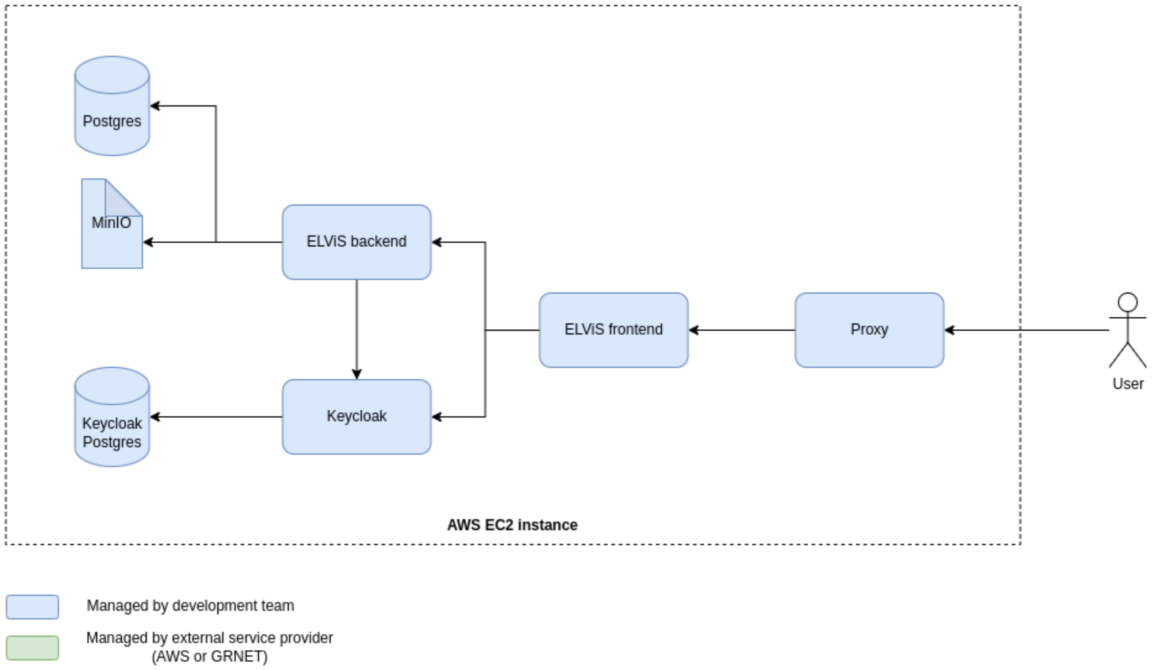


Figure 1. Overview of current ELViS infrastructure, the development team manages all services.

An important distinction with the proposed setup is that user registration and identification runs completely through the Keycloak server. New users register through ELViS, which will create a new entry in the database. User information is stored both in the Keycloak as user attributes and in a user information table in the ELViS database.

Future setup

The proposed setup is that ELViS no longer uses its own internal Keycloak setup but uses a generic solution built and maintained by the National Infrastructures for Research and Technology (GRNET). This means we decouple the AAI component from the ELViS application. ELViS's applications (frontend and backend) will be managed within a Kubernetes cluster on AWS. However, both the data and the AAI infrastructure will be managed externally. The data will be housed within AWS-managed solutions such as an Amazon Relational Database Service (AWS RDS) for Postgres and S3 buckets. The AAI will be managed by GRNET, who also manages the hosting of this infrastructure. This means that the development team can focus on developing DiSSCo services instead of data and user management (see Fig. 2).

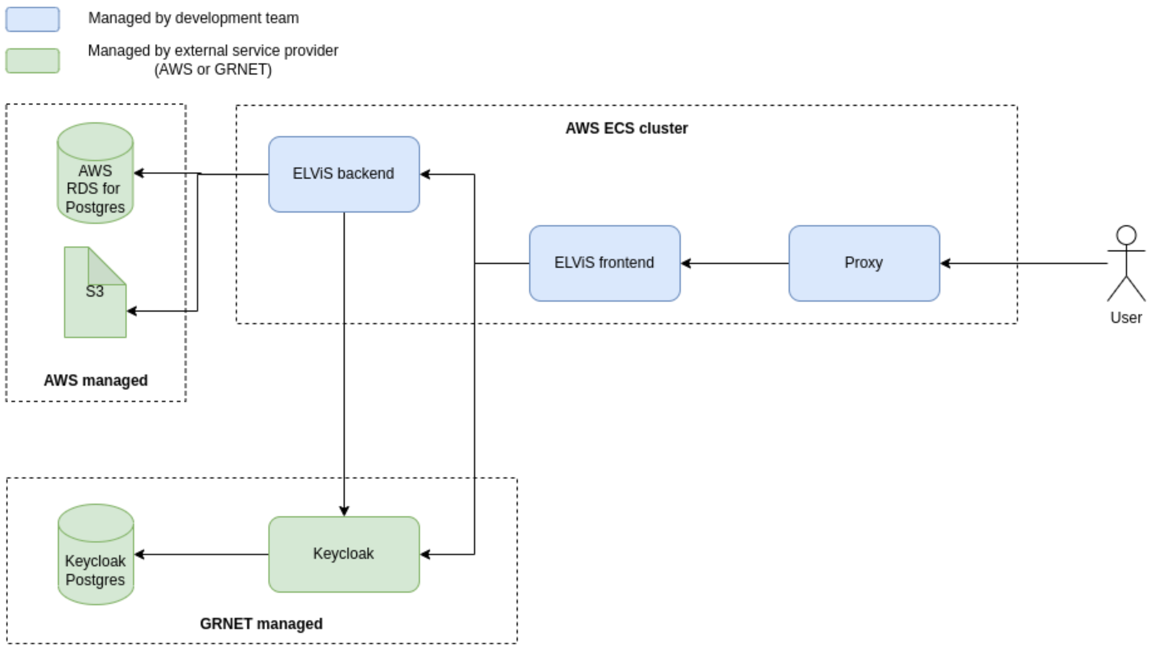


Figure 2. Overview of proposed ELViS infrastructure, some services are managed by service providers such as GRNET and AWS.

The AAI service managed by GRNET is a dedicated instance of RCIAM, which has been named "DiSSCo Login". RCIAM uses the eosc-kc fork of Keycloak that adds support for importing SAML-based Identity Providers participating in identity federations. Support for SAML Identity Provider federations is necessary to connect DiSSCo Login to eduGAIN, the international interfederation of research and education identity federations. The demo instance of the DiSSCo Login service (https://login-demo.dissco.eu/account) has already been registered as a SAML Service Provider in eduGAIN. Through eduGAIN, services connected to DiSSCo Login can become available - with little or no administrative involvement - to researchers using their existing accounts from more than 5,200 Universities and Institutes from 80 National Identity Federations. This enables the use of trusted Identity Providers (IdPs) to authenticate users. We need the IdPs to provide digitally signed (often partially encrypted) statements about an entity's digital identity. An IdP can be the user's own organization, but it can also be a platform such as ORCID providing the credentials. The users don't need to create a new account with credentials in ELViS, but can use their existing account. In other words, the user information will be provided to ELViS through the IdPs.

All critical service components of the DiSSCo Login service are operated in High Availability mode. The deployment architecture can scale horizontally by provisioning more nodes, if required to increase service capacity. The backend database store (based on Postgres) has been configured to operate in clustered mode, supporting streaming replication and Point-in-Time Recovery for a period of six months. The role of the demo instance of DiSSCo Login is twofold: (1) testing new features without affecting the production environment and (2) testing the integration with new services, i.e. OpenID Connect relying on parties or SAML Services Providers.

3. Piloting activities

During the pilot, we undertook several steps towards the envisioned setup. The software and the data needed to be migrated, and changes in the application code were necessary.

Software migration

To enable the proposed setup, we needed to migrate the software from the single server setup to a Kubernetes cluster. This meant that we needed to create an external container image registry to which the ELViS images were pushed, after which we could migrate the docker-compose file to a Kubernetes deployment file.

The next step was to migrate the application. We initially migrated ELViS as-is to minimize the impact the AAI migration would have. Only after we successfully deployed and tested the new setup did we change the Keycloak server to the AAI of GRNET.

Data Migration

Besides the software migration we also ran a data migration. Part of the user's information is located in the ELViS database. We migrated this data from the internal ELViS server to an AWS-managed Postgres server. This provides several benefits, such as automated backups, automated scalability and higher uptime. Besides the data coming from the database, we also migrated the Keycloak setup. ELViS expects certain clients, roles, and groups to be in the Keycloak instance. We made a backup of the ELViS Keycloak setup and provided this to GRNET. GRNET imported this setup and added this to their Keycloak.

Code Changes

Besides infrastructural changes, we also needed to make a few code changes. As mentioned, ELViS will no longer support the registration of users. Users will have to come in through a trusted identity provider. This identity provider will provide user attributes to the application. However, during the pilot, we noticed that not all expected attributes are provided by all identity providers. This means we have to request these attributes from the user after they have registered through an identity provider and store these ourselves. This is similar to the setup which we are using for our pilot with the Unified Curation and Annotation System (UCAS) for DiSSCo Prepare.

As part of the information expected by ELViS was no longer present in the token, we had to rewrite part of the authentication mechanism. We tried to keep the changes to a minimum to only impact the specific part which provided the issue. In the future, it would be good to rewrite a large part of this functionality.

4. Future development

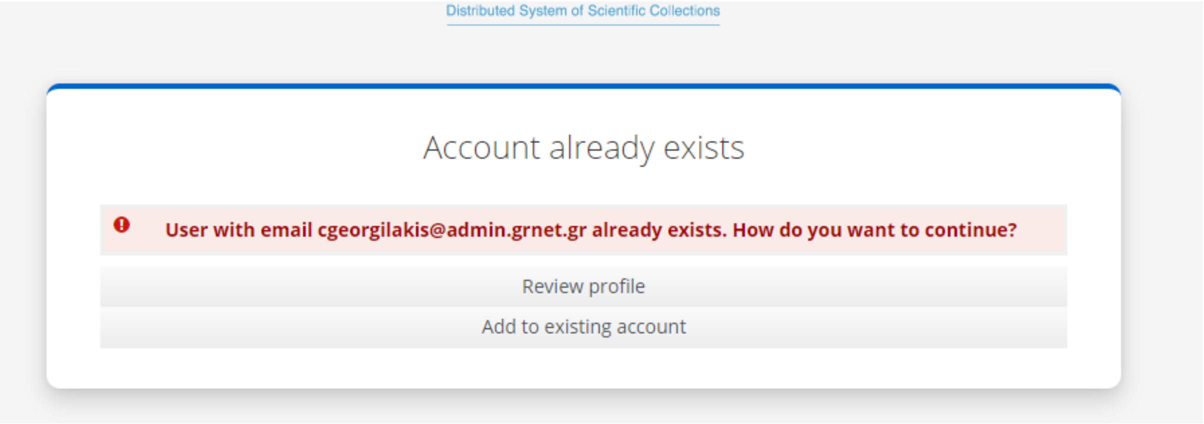
Moving forward from a pilot to an actual production environment, there are a couple of changes that need to be made. These changes will be described in this chapter.

User migration

During the pilot phase, we created a test environment in which we implemented the envisioned architecture. All changes were implemented within this test environment. However, the test environment has only a limited number of test users. With these users, we tested the generic AAI implementation and the SSO with other DiSSCo applications.

When we want to promote this setup to a production environment and keep all the current user and user data intact, we would need to migrate all users from the old Keycloak instance to the GRNET-managed AAI. This user migration is possible, and a migration plan has been formulated together with GRNET and tested in the demo instance of DiSSCo Login.

After migration, a user will get a new internal identifier. Within the ELViS database, we need to change the old user identifier to the new identifier and ensure all information pointing to the old identifier will now point to the new user identifier. The user itself will only be able to keep their information if the email address used to register in ELViS is the same as the one they will be provided through the identity provider. The user will then get the following screen.


Distributed System of Scientific Collections

Account already exists

User with email cgeorgilakis@admin.grnet.gr already exists. How do you want to continue?

If the user adds the identity provider to an existing account, then all the ELViS authorization and data will become available to the user. Before we migrate the user data, we will inform all users about the migration and ask for their permission.

Additional code changes

As the registration process will no longer flow through an ELViS registration, we must remove this functionality from the application. The user will only be able to authenticate through one of the IdPs. This change impacts both the frontend and backend of the application. We would also need to rewrite part of the documentation to describe how users can register if they don't have an account at an IdP.

As the IdP provides only a limited set of information about the user, we would need to request the user to provide ELViS with the remaining information. This will be done after a user has been authenticated. The additional information will then be stored only within the database instead of the Keycloak server. Ideally, we would like to use the same database as where we store the user information for DiSSCo. This would ensure that always the same information about the user is used. It would also help the user as they wouldn't have to submit personal information in the separate DiSSCo applications.

5. Basic Requirements

In Milestone document 48 of SYNTHESYS+, a list of basic requirements was described. After the pilot has been completed, we can evaluate if we can achieve these requirements with the proposed setup.

Requirement:

Federated identity: A user-friendly infrastructure that does not require registration of a new username and password for each individual service. Users should be able to authenticate with their existing institutional credentials.

Conclusion:

By using a single AAI service for the DiSSCo infrastructure, we can guarantee Single Sign-On access to all DiSSCo services. Once a user has registered for a single DiSSCo service through an IdP, it will be known to all other DiSSCo services. This ensures that a user doesn’t need to re-register or login when using another DiSSCo service.

Requirement:

Trust: User information from the institutions (“Identity Providers”) are connected to various DiSSCo and non-DiSSCo applications (“Service Providers”). Some of these services will be open, some will be password protected. Mutual trust needs to be established between various Identity and Service Providers.

Conclusion:

We support federated identity. Users can log in through a wide range of identity providers. This can be through an organization account or through a generic service such as ORCID.

Requirement:

Policy and agreements need to be in place to release various user attributes, such as name and email addresses, in accordance with institutional and GDPR policies.

Conclusion:

ELViS will provide a privacy statement stating how user information will be stored and used. Users will always be able to remove their user information. Removing user information might degrade the user experience.

Requirement:

Use of open standards such as OIDC.

Conclusion:

ELViS will make use of the OAuth 2.0 protocol for authorization. On top of OAuth 2.0 it will use OpenID Connect (OIDC) as an identity layer to verify the user's identity and provide user information. We will use OIDC in combination with JSON Web Tokens (JWT) as a means of user information exchange.

Requirement:

Establish a unique DiSSCo identifier for all users across all DiSSCo Services. This is in addition to the federated identifier, the AAI system will create a permanent, opaque, and non-reassignable identifier

Conclusion:

The AAI infrastructure, with as an IAM tooling Keycloak, will generate UUID's for user identifiers. These user identifiers will be used within the application to refer to a particular user. These identifiers are permanent, within a Keycloak instance, opaque and non-reassignable.

Requirement:

Ability to use one or more associated authentication providers.

Conclusion:

Through the long and extensible range of identity providers a user can log in through their own preferred provider. Identification is based on email address. As long as the user uses the same email address, it can associate multiple IdPs to a single user account.

Requirement:

Integration with Identity Providers such as ORCID and other social media services.

Conclusion:

ORCID and several other social media services are among the IdPs available for the users to log in.

Requirement:

Compliance with eduGAIN, ELIXIR and other RI services. This is to ensure that ELViS and other DiSSCo services can interact with other RI services and also with the European Open Science Cloud (EOSC) ecosystem.

Conclusion:

The AAI setup of GRNET complies with eduGAIN, ELIXIR and other RI services. GRNET is part of ELIXIR-GR and also involved in the EOSC Futures project.

Requirement:

A system to manage user, group membership, and roles. Each user can belong to one or several groups. A group member can have different roles in the group (such as administrator, manager, and editor).

Conclusion:

Keycloak as an IAM tool provides a way to manage users, clients, groups, and roles. A user can be part of various groups, and groups can also be part of another group. Roles can be defined on the level of an individual user as well as on a group, which is the preferred way. This provides a fine-grained way to control access over different users and clients.

Requirement:

Integration with FAIR Digital Object repository. Based on DiSSCo design principles, an underlying Digital Object Architecture will be an integral part of the DiSSCo ecosystem.

Conclusion:

As we use a single AAI within DiSSCo there is an automatic integration with the DiSSCo FAIR Digital Object repository.

Requirement:

Access Protocol Translation: define an administrative, policy and the technical boundary between the internal/external services and resources.

Conclusion:

By extracting the AAI from the rest of the application, we create a technical boundary between the DiSSCo services and the identity services managed by GRNET. DiSSCo as a service provider, will implement policies and procedures. Some of these have been discussed during the ELViS development work, for example, the privacy statement.

Requirement:

DiSSCo reserves the right to revoke access in cases of user misbehavior or if it is discovered that the user has provided false information. There should be a mechanism to revoke access and update the record to prevent further access.

Conclusion:

Through the admin portal of Keycloak an administrator can immediately revoke any token of the user and remove the user account. It is also possible to invalidate any current session that the user has, forcing the user to log out of the system. Within the DiSSCo services, we will create provenance so that any misbehavior by a user can be reverted.

6. Conclusion

After evaluating the basic requirements, we can conclude that the pilot has shown that all requirements can be met with the proposed approach. By using the existing AAI provided by GRNET we can create a single point of user authentication and authorization for the complete DiSSCo infrastructure. During the pilot, we had to make several changes to both the architecture and the code of ELViS. However, due to the use of the same IAM tooling, Keycloak, in both the ELViS setup and the AAI the migration was relatively easy.

We will need to make some changes before we can start using the pilot in production. The most importantly will be the user migration. By migrating the users, we can ensure that all available permissions and data will stay connected to the user and no information is lost. Additionally, we will need to make a generic user information service within DiSSCo to store all information unavailable through the AAI.

References

Wouter A., Islam S., Koumantaros K. , Laskaris N. 2020. MS48 AAI INFRASTRUCTURE DESIGN. SYNTHESYS PLUS https://osf.io/sc7vw