CLARIN Federated Content Search (CLARIN-FCS) - AAI 1.0

Oliver Schonefeld, Leif-Jöran Olsson, Thomas Eckart, André Moreira, Erik Körner

Version 1.0, 2024-12-04

Table of Contents

Chapter 1. Introduction

This specification is an extension specification to the CLARIN-FCS Core 2.0 specification and describes an access mechanism for restricted resources.

The Federated Content Search currently does not support the restriction of access to resources to specific user groups or users. There is no mechanism in the aggregator or other FCS clients that limits access to announced resources and endpoints are typically accessible via the FCS protocol directly (omitting the use of the FCS aggregator). The goal is to allow restricting access to FCS resources to users that are authenticated using the established AAI infrastructure.

This contains the following issues:

1. Shibbolizing the FCS aggregator frontend so that
   - unauthenticated users still get access to all unrestricted endpoints
   - authenticated users get access to all unrestricted endpoints and additionally to the restricted endpoints
2. Specifying (and implementing) changes for FCS endpoints so that:
   - endpoints can announce restricted resources to FCS clients, e.g. the aggregator
   - endpoints can rely on authenticated FCS user requests

1.1. Terminology

The key words `MUST`, `MUST NOT`, `REQUIRED`, `SHALL`, `SHALL NOT`, `SHOULD`, `SHOULD NOT`, `RECOMMENDED`, `MAY`, and `OPTIONAL` in this document are to be interpreted as in RFC2119.

1.2. Glossary

AAI

Authentication & Authorization Infrastructure. A service and a procedure that enables members of different institutions to access protected information that is distributed on different web servers. See Shibboleth.

IdP

Identity Provider, a system entity that issues authentication assertions, see SAML and AAI.

JWK

JSON Web Key, see RFC7517.

JWKS

JSON Web Key Set or *JWK Sets*, see JWK and RFC7517.

JWT

JSON Web Token, see RFC7519.

RSA

Asymmetric public-key cryptography system by *Rivest-Shamir-Adleman* for digital signatures and encryption.

SAML

Security Assertion Markup Language, an open standard for exchanging authentication and authorization data.

Shibboleth

Shibboleth is a single sign-on log-in system for computer networks and the Internet. See also SAML.

SP

Service Provider, a system entity that receives and accepts authentication assertions, see SAML and AAI.

1.3. Normative References

RFC2119

Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997, https://www.ietf.org/rfc/rfc2119.html

RFC7515

JSON Web Signature (JWS), IETF RFC 7515, May 2015, https://www.ietf.org/rfc/rfc7515.html

RFC7517

JSON Web Key (JWK), IETF RFC 7517, May 2015, https://www.ietf.org/rfc/rfc7517.html

RFC7519

JSON Web Token (JWT), IETF RFC 7519, May 2015, https://www.ietf.org/rfc/rfc7519.html

XML-Namespaces

Namespaces in XML 1.0 (Third Edition), W3C, 8 December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

LOC-DIAG

SRU Version 1.2: SRU Diagnostics List, Library of Congress, http://www.loc.gov/standards/sru/diagnostics/diagnosticsList.html

CLARIN-FCS-Core 2.0

CLARIN Federated Content Search (CLARIN-FCS) - Core 2.0, SCCTC FCS Task-Force, May 2017, https://trac.clarin.eu/wiki/FCS/Specification, https://github.com/clarin-eric/fcs-misc/tree/main/fcs-core-2.0, https://office.clarin.eu/v/CE-2017-1046-FCS-Specification-v20230426.pdf

1.4. Non-Normative References

RFC6838

Media Type Specifications and Registration Procedures, IETF RFC 6838, January 2013, https://www.ietf.org/rfc/rfc6838.txt

RFC3023

XML Media Types, IETF RFC 3023, January 2001, https://www.ietf.org/rfc/rfc3023.txt

RFC8017

PKCS #1: RSA Cryptography Specifications Version 2.2, IETF RFC 8017, November 2016, https://www.ietf.org/rfc/rfc8017.txt

1.5. Typographic and XML Namespace conventions

The following typographic conventions for XML fragments will be used throughout this specification:

- `<prefix:Element>`

  An XML element with the Generic Identifier *Element* that is bound to an XML namespace denoted by the prefix *prefix*.

- `@attr`

  An XML attribute with the name *attr*.

- `string`

  The literal *string* must be used either as element content or attribute value.

Endpoints and Clients `MUST` adhere to the XML-Namespaces specification. The CLARIN-FCS interface specification generally does not dictate whether XML elements should be serialized in their prefixed or non-prefixed syntax, but Endpoints `MUST` ensure that the correct XML namespace is used for elements and that XML namespaces are declared correctly. Clients `MUST` be agnostic regarding syntax for serializing the XML elements, i.e. if the prefixed or un-prefixed variant was used, and `SHOULD` operate solely on expanded names, i.e. pairs of namespace name and local name.

Chapter 2. Restricted Access to Resources

The FCS supports restriction of access to resources to specific user groups or users that are authenticated using the established AAI infrastructure while still allowing unauthenticated access to all unrestricted resources. This mechanism limits access to announced resources that are typically accessible via the FCS protocol directly (omitting the use of the FCS aggregator).

2.1. Technical Description

The aggregator allows an optional login via Shibboleth. Authentication of search queries which are sent to an FCS endpoint is implemented using authentication headers with JSON Web Token (JWT). User information (e.g. mail address or similar) is encoded in the token as claims.

JWTs include an RSA signed token which `SHOULD` be checked by the respective endpoint. In case of missing or insufficient authorization when accessing restricted resources, the endpoint `MUST` rise an error using SRU diagnostic "Authentication error" (`info:srw/diagnostic/1/3`). The diagnostic "Not authorised to send record" (`info:srw/diagnostic/1/68`) `MAY` be sent as an error when an authenticated request was made but additional authorization is still required, e.g. if a resource might only allow certain users access.

The aggregator owns a single common RSA key for all endpoints. For all endpoints with restricted resources its public key has to be manually transferred and included in their configuration. Communication between aggregator and endpoints is only allowed via SSL. Claims *iss*, *sub* and *aud* `MUST` be encoded in the JWT. The endpoint `MUST` check *aud* to see if it is the correct recipient and `MAY` check the fields *iss* and *exp*.

The availability and nature of a personal identifier attribute in the aggregator itself, is not guaranteed for a successful login. In SAML this is limited by the configuration of the external IdP selected by the user. When available, one of three SAML attributes is used as user personal identifier: *eduPersonPrincipalName*, *eduPersonTargetedID* or *mail*. These attributes are mapped by the aggregator to *userID* in the same order of preference. *userID* is then passed to the endpoint. The aggregator issues this information to JWTs and sends it to the appropriate endpoints. The endpoints alone announces which authentication information is required (see next section) and decides at runtime whether the information provided is sufficient for access. While the aggregator front end can inform and guide the user in advance, in case authentication information is still missing.

2.1.1. JWT Signature Algorithm

The JWT will be signed with the "RS256" algorithm (see JSON Web Token (JWT), Section 8 "Implementation Requirements"). The digital signature is created using RSASSA-PKCS1-v1_5 with the SHA-256 hash function. For this a RSA key size of 2048 bits or larger `MUST` be used. Endpoints and clients `MUST` support 2048 bit RSA keys but `SHOULD` be able to handle larger key sizes.

Signing of the JWT will require the private-public RSA key pair in the FCS client. FCS endpoint will only require the public RSA key for validating the JWT signature. The JWT will not be encrypted.

An example of using the RS256 algorithm can be found in the Appendix A.2 of JSON Web Signature (JWS).

2.1.2. Key Exchange

Verification of JWTs at an endpoint requires it to know the public RSA key used to generate and sign the JWT by the requesting client. Endpoints with restricted resources need to be configured to include the public key. This can be done in two ways, by manually transferring and including the public key in the configuration, or by using a public JSON Web Key Set endpoint (`https://<fcs-client.url>/.well-known/jwks.json`) to allow for automatic retrieval of the public key.

2.1.3. JWT Claims and Verification

The claims *iss* (issuer), *sub* (subject) and *aud* (audience) `MUST` be encoded in the JWT by the client.

- *iss* `MUST` contain a unique identifier for the client, e.g. the canoncial public URL for web applications like the FCS aggregator.
- *aud* `MUST` contain the URL of the endpoint which is used to register the endpoint at the client and is also being used to for SRU/FCS requests.
- *sub* `MUST` contain the `userID` if a resource has a requirement of `personalIdentifier`. If `authOnly` then the *sub* claim `SHOULD` be empty and should be ignored.

The claims *exp* (expiration time), *nbf* (not before) and *iat* (issued at) are `OPTIONAL` but `SHOULD` be used to limit token lifetime. Values `SHOULD` therefore be low, e.g., 15 seconds. Clocks of endpoint and client need to be synchronized. Setting the *jti* (JWT ID) will also further secure communication by helping to prevent replay attacks.

The endpoint `MUST` check *aud* to see if it is the correct recipient and `MAY` check the fields *iss* and *exp*, *nbf*, *iat*, *jti*.

The *sub* (subject) field `SHOULD` be used retrieve the `userID` for resources with `personalIdentifier` requirement and to perform advanced authorization.

2.2. Announcing Restricted Resources by the Endpoint

In order to announce restricted resources the endpoint needs to explain in the `<Resources>` section of the `<EndpointDescription>` that it does support the aforementioned procedure for a resource. This is done via a dedicated element `<AvailabilityRestriction>`:

Example of AvailabilityRestriction in Endpoint Description for a Resource

```xml
<Resources>
  <Resource>
    <!-- ... -->
    <Languages>
      <Language>swe</Language>
      <Language>deu</Language>
    </Languages>
    <AvailabilityRestriction>requirementName</AvailabilityRestriction>
    <!-- ... -->
  </Resource>
```

```
</Resources>
```

The `<AvailabilityRestriction>` element can be defined for each `<Resource>` element. In case of sub-resources, each resource `MUST` declare restrictions independently. Restrictions in sub-resource `MAY` differ from their parent resource, i.e. there is no implicit inheritance semantic.

From the backwards compatibility perspective this means that if you do not define the `<AvailabilityRestriction>` element all resources will be available for searching through the endpoint. That will also apply to sub-resources without `<AvailabilityRestriction>` element while their parent `<Resource>` has a restriction defined.

The text node `requirementName` in `<RestrictionRequirement>` is either of `authOnly` or `personalIdentifier`.

- `authOnly`: the resource does not require any attributes except for the authentication via home institution.
- `personalIdentifier`: extends `authOnly` by passing a `userID` using the procedure described above to the endpoint. The `userID` will be one of *eduPersonPrincipalName* or *eduPersonTargetedID* or *email*.

Changelog

2024-12-04 — Update specification based on current prototype implementation

- Add details about key/signature algorithms, RS256
- Add key exchange details, using JWKS
- Add details about JWT claims (usage, requirments)
- Add SRU 68 dignostic
- Change `<AvailabilityRestriction>` element in `<EndpointDescription>`
- Change order of SAML attributes for `userID` computation
- Change formulation to generalize for any FCS client, not only AAI for FCS Aggregator

2023-06-12 — Conversion to AsciiDoc and Migration of specification documents to Github

- Convert specification documents for FCS Core 2.0, Core 1.0, DataView and AAI to AsciiDoc
- Migrate from CLARIN Trac to CLARIN Github
- Add Github Actions workflow to automate build process

2021-01-27 — FCS Update Proposal for AAI

2020-02-19 — Meeting about Shibbolizing FCS