Setting up a SAML Service Provider

From your resource or web app to a federated Service Provider within CLARIN

Sander Maijers

Purpose

- To advise and help
- CLARIN centres’ technical decision makers and implementers
- with setting up a SAML Service Provider (SP)
- within the CLARIN Service Provider Federation (SPF).

My assumptions about you

1. You have at least a basic experience working with command lines on e.g. a Linux operating system.
2. You know the purpose of the SPF.
3. You don’t necessarily know much about SAML (the technical set of standards the SPF is based on).
4. You are interested about setting up an SP for language resources and web apps, similarly to other centres.

Structure

- Background.
  - Your aims.
  - The SPF for your materials.
  - SAML and the other technological context.
- Scenario: a more concrete elaboration of your aims.
- Overview of software, focusing on Shibboleth SP.
  - Alternatives.
  - Properties of Shibboleth SP, interactions with your aims.
- Practical example of a solution for the scenario.

Approach

- Overview and discussion of software packages.
- High-level walk-through of setting up one common, reasonably advanced SP.
- Limited coverage of documented, implementation-dependent technical details.

Instead of explaining all technical considerations, I’ll approach it like this:

- Solving such issues before you would have to,
- in an explicit, standard & open way (open source),
- using prepackaged, preconfigured software distributions almost ready to run.

i NOTE Willem explains Docker based app virtualization in his upcoming presentation.

The SPF: for you?

If your centre wants to …

- offer resources or web applications to the CLARIN community;
- restrict access to these resources by technical means;
- avoid keeping your own administration of identities,

then creating an SP for the SPF would be of interest to you.

Your materials from an IT viewpoint

Your materials can be offered digitally in various ways.

Technical context

1. **Web resources**: user-facing websites.
2. Visited in a **web browser** on a regular computer (i.e., no mobile apps).

Types

1. Language resources: *static content*.
2. Partly server-side *web applications*.

I’ll generalize these types to *website*.

Authentication and authorization

authn: Who are you?

authz: Why do you have access to this?

auth: generalized term I’ll use where appropriate.

The whole IT/organizational infra around this is often called *Authentication and Authorization Infrastructure* (AAI).

SAML

attribute: a typically relevant well-defined property of each person, such as name, encoded in a technically standard way, e.g., your surname with leading capital.

Identity Provider (IdP): a networked registry with identity information about certain people, as expressed with some attributes.

Service Provider: a technical abstraction layer over one or more websites, with the aim of regulating access to (things published via) them based on user identity and/or attributes.

federation: collectives of such providers, administered and monitored by an administrative and technical staff.

AAI standards

SAML: a set of standards, protocols that help technically implement these abstractions.

SAML 2 is the *de facto* standard in national research and education network (NREN) AAI federation.

Academic institutions are very well, yet exclusively, represented as SAML IdPs.

Though SAML is rather complex and dated, we must use it within the SPF for this reason.

Scenario and sketch of requirements

- You are to operate an SP for your centre, meaning to filter academic users of your websites (i.e., users registered in academic institutional IdPs).
- You use this IdP to retrieve basic attributes describing your user, to allow even more fine-grained access control.
- You want offer your materials via multiple auth-requiring websites, with different levels of traffic.
- Your goal is thus to become part of the CLARIN SPF.

Concretely, your centre wishes to offer:

- A: one resource requiring mixed authn and authz;
- B: a web application requiring authn.

Like most other centres, your plan is restrict access to these using Shibbboleth SP.

Software packages for this scenario

There are many proprietary and a few open source alternatives, with quite different properties.

1. OpenAM
2. Shibboleth SP
3. SimpleSAMLphp

i NOTE Exhaustive list.

Shibboleth SP

- Shibboleth SP is most popular in academia general and in the SPF.
- It has particular design issues and limitations.
- But it will probably satisfy most centres' needs.

Shibboleth SP will be the basis of the practical example walked through later on.

Alternatives to Shibboleth SP

Some notes about alternatives to Shibboleth SP …

- OpenAM: has REST and C/Java API, supports more AAI standards (e.g., OpenID Connect).
- SimpleSAMLphp: not well-suited for non-PHP apps, less functionality, easier to maintain.

What does Shibboleth SP offer?

In sum:

Unlike authentication systems that expose an API, the SP operates inside the web server, isolated from applications, and places authentication and attribute information into the web application environment using environment variables, or using custom HTTP request headers in less functional web servers. (Shibboleth documentation)

Key design choices in Shibboleth SP

Shibboleth SP consists of two parts: `mod_shib`, an Apache httpd module, and `shibd`, the workhorse doing the SAML magic.

1. Merely an add-on to the popular Apache httpd or IIS general-purpose web servers.
2. No remote API (e.g., RESTful).
3. No API at all (i.e., software library that makes SP of app).

Web apps and Shibboleth SP

So, somewhat simplified, your app must:

→ either rely on HTTP header message passing (advised against in Shibboleth docs).

→ or use classical *Common Gateway Interface* (CGI)-like non-networked communication.

Not very well-suited for:

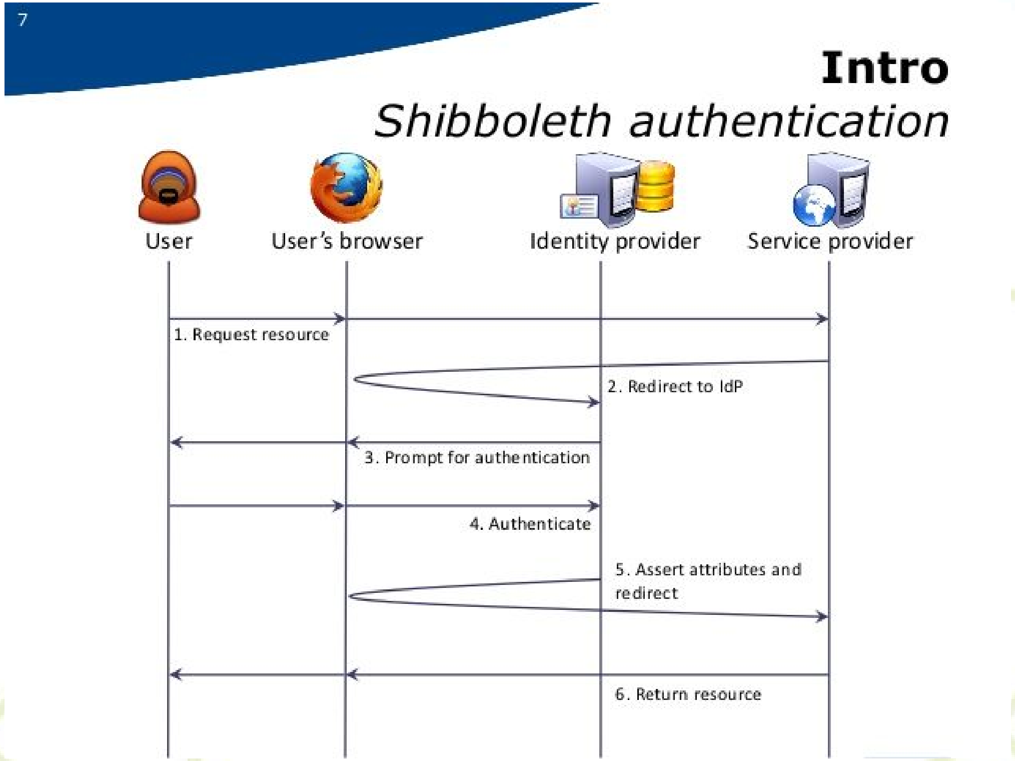
- single-page web apps;
- native (mobile, computer) networked apps;
- *application servers* e.g., Node.js.

Strategic issue

The current scenario does not cover these previous types of apps.

Does your centre possibly want to develop/use any such apps and involve SAML-based auth?

The SAML Web Browser profile



SAML IdPs and SPs need to know each other

Practically every NREN runs an *identity federation*, often covering most higher-education institutions in a country.

Identity federations exchange so-called *SAML metadata* between each other.

SAML metadata describes entities (SPs, IdPs).

Identity federations assign **trust** to entities. If they fancy ...

The CLARIN SPF collects our CLARIN entities and makes sure that they are trusted by identity federations across Europe.

Whether identity federations will trust your SP, depends on **what attributes you request** and the **quality of the SAML metadata** about your SP provided to them.

How to let Shibboleth SP treat your websites all differently

You may need to fine-tune your configuration for each website, e.g. extract different attributes for app 1 vs. app 2.

Shibboleth SP offers the *application* abstraction. An application in this context is a category of websites. Another term used to describe this category is a *logical SP*.

One of the most common things you'll do when creating an override is to assign it a special `entityID`, making it a distinct logical SP living inside the same physical installation. This is done by adding an `entityID` property to the `<ApplicationOverride>` element.

(Shibboleth documentation)

Use a single SP entity for all your websites

- Your centre offers multiple websites;
- Yet a proliferation of SP entities carries *a lot* of administrative and technical overhead for everyone involved.

So for our scenario, we may want logical SPs, but we do not want a proliferation of entities.

Architectures for our scenario

→ In sum: multiple websites served by httpd handled by a single Shibboleth SP.

→ Somewhat challenging, more so if you desire maximum flexibility, isolation and separation of concerns.

I'll focus on two possible architectures:

1. reverse proxy frontend with Shibboleth SP plus website backends (I)
2. website frontends plus part of Shibboleth SP (`shibd`) in a single backend (II)

Properties of both architectures

1. You need to safeguard integrity and confidentiality of front-back traffic.
2. Some *Single Point of Failure* (SPoF) exists, as long as the Shibboleth SP isn't fully clustered (not covered here).
3. Use a low-latency network link between websites and Shibboleth SP `shibd` backend.
   → Important for responsivity when browsing on the restricted parts of your website.

Websites behind reverse proxy frontend (I)

1. Needs a low-latency network link between proxy and websites.
2. `httpd` OR `shibd` OR `mod_shib` → SPoF.

i NOTE If the proxy is down the websites are unreachable, even their unrestricted parts.

Properties (I)

1. Proxy → bottleneck. All HTTP traffic to all websites passes.
2. Attribute passing: problematic.
   - If attributes are passed as environment variables, then all websites must be hosted on the proxy host itself.
   - If not, then attributes must be passed as HTTP header fields, which is against best practice (inefficient, some security risks).

`shibd` backend, websites are each frontends (II)

1. Needs a low-latency network link between `shibd` and websites.
2. `shibd` → SPoF.

Properties (II)

1. Allows user traffic to flow to different network hosts.
2. Requires separate endpoints in SAML metadata. Automatic SAML metadata generation cannot be used fully automatically anymore.
3. Attributes can always be passed as environment variables, since `mod_shib` runs separately on every website.

Walkthrough of a solution for the scenario

Two websites in Docker containers, backed by a `shibd` backend container.

Varying authn and authz configuration per ‘subresource’.

https://resource_a.clarin.eu

A CGI web app.

https://web_app_b.clarin.eu

Registering your new SP

1. Make sure your centre is in the Centre Registry. Contact us via this page.
2. Make sure you have a developer account.
3. Read the documentaton on https://www.clarin.eu/spf, including the guidelines for SAML metadata.
4. Add the final SAML metadata about your SP to the preproduction SAML metadata batch in the CLARIN SVN.

5. Check whether there are any issues left to resolve with your SAML metadata. If so, commit your improved SAML metadata back to the CLARIN SVN.
6. Create a Trac ticket for the ‘AAI’ component, requesting that your SP be registered with the identity federations.
7. Monitor the progress of the registration via https://centres.clarin.eu/spf.

Support

In case you need help with an issue that can’t be answered/solved straight away.

→ Create a Trac ticket for the ‘AAI’ component.

Advice

1. Take and adapt the SAML metadata of other centre’s SPs!
2. Use a preconfigured Shibboleth SP container image provided by CLARIN ERIC and adapt it.
3. Carefully adhere to the SPF SAML metadata guidelines.
4. Use the right tool for the job. If you are open to other, better software than Shibboleth SP, use it.

Questions, discussion

Thanks for attending!