A specification for hinting an IdP which discovery service to use

Publication Date	2021-10-11
Authors:	Marcus Hardt (ed), Jens Jensen, Ivan Kanakarakis, Christos Kanellopoulos, Nicolas Liampotis, Mikael Linden, Mischa Sallé
Document Code:	AARC-G062
DOI:	10.5281/zenodo.5763858
Reference as:	https://aarc-community.org/guidelines/aarc-g062
Community:	Architecture Area

Abstract

This document defines a generic browser-based protocol for conveying - to services - hints about the discovery service that should be used for letting the user choose an IdP.

Status of this document

This specification defines how services can suggest specific Discovery Services to be used for letting the user choose an identity provider.
A list of all AARC community guidelines and the latest revision of each guideline can be found at https://aarc-community.org/guidelines.

Introduction

Authentication at a service in an environment with multiple Identity Providers (IdPs) requires the service to send users to a proper IdP. The selection of the IdP is often assisted by IdP discovery services, where users can find and select their IdP.

In the AARC Blueprint Architecture (BPA) [AARC-G045], the service that the user tries to access is usually not directly connected to the authenticating IdP (i.e. the IdP identifying the user as in [SAML2Core Sec. 3.4.1.5]), but the connection is through one or more upstream SP-IdP-proxies. The service and each of these SP-IdP-proxies need to be able to route the user authentication request to the IdP interface of the next upstream SP-IdP-Proxy until the authentication request reaches the authenticating IdP. Each time an SP-IdP-Proxy is connected to more than one upstream SP-IdP-proxies or IdPs, the SP-IdP-Proxy will have to rely on an IdP discovery mechanism where the user will need to provide input about the preferred upstream path.

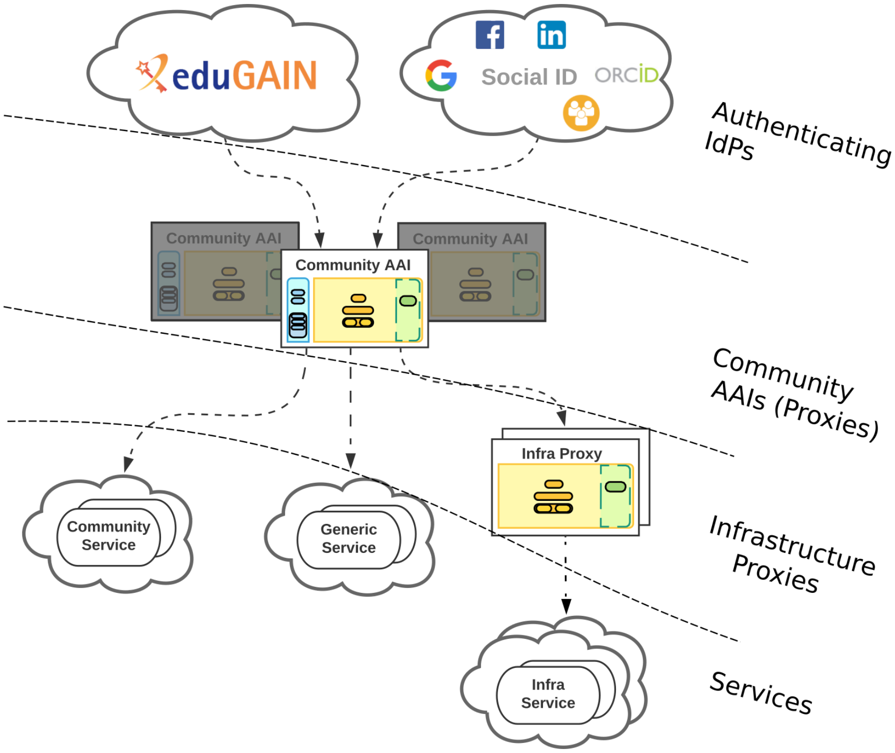

Figure 1: Different entity types shown in the Blueprint Architectures 2019 diagram [AARC-G045].

This document defines a portable and technology-agnostic specification that enables entities to produce and send *hints* that can be consumed by SP-IdP-proxies to route the user to the correct discovery service, thereby simplifying the discovery process for the end-user.

This mechanism provides a way of customising a single service for different communities by directing the user to community-specific discovery services.

Other, similar AARC hinting specifications include [AARC-G061].

1.1. Terms and Definitions

The term "Identity Provider" (or "IdP") may refer to SAML IdPs or to OpenID Connect (OIDC) Providers (OPs). Similarly, the term "Service Provider" (or "SP"), may refer to SAML SPs or to OpenID Connect Relying Parties (RPs).

With a Discovery Service (DS) we mean the logical element providing the user with a list of IdPs to choose from. It can be integrated into other services or IdP-SP-proxies.

This document also defines the following terms (see also Fig.1 above):

- An "entity identifier" identifies an entity, which could for example be an SP, an authenticating IdP, an SP-IdP-Proxy or a Discovery Service. For the scope of this document, the entity identifier refers to a Discovery Service.
- A "hint consumer" receives and consumes hints. This may be an SP or an SP-IdP-Proxy.
- A "hint producer" produces and sends hints. This may be an SP, or an SP-IdP-Proxy.
  Note that valid values of hints - i.e. the entity identifiers - are defined by the hint consumer and must be communicated out of band to the hint producers.
- The “authenticating IdP” is the IdP at the end of the authentication chain at which the user ultimately identifies. In the AARC BPA [AARC-G045], the authentication chain typically contains one or more intermediate SP-IdP-proxies ([SAML2Core] sec. 3.4.1.5).
- “Community AAIs” [AARC-G045] serve as IdPs or SP-IdP-Proxies for specific communities.
- “Multi-tenant Community AAIs” [AARC-G045] serve as IdPs or SP-IdP-Proxies for multiple communities.
- “Discovery Services” (DS) [IDPDSP] allow users to choose an IdP for authentication.
- “Infrastructure Proxies” [AARC-G045] are used to connect all services of a specific infrastructure. Infrastructure Proxies may be connected to multiple Community AAIs.
- “Community services” [AARC-G045] are connected to a single Community AAI.
- “Infrastructure services” [AARC-G045] are connected to a single Infrastructure Proxy.
- “Generic services” [AARC-G045] may be connected to multiple Infrastructure- and/or Community AAIs.

1.2. Use Case

The main use case of the hint defined in this document is to support “Context-Specific Discovery”. For example, this allows services that are offered to multiple communities to route users to a community-specific Discovery Service. This is intended to improve the user experience for a given community or organisation.

1.3. Conventions

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 described in RFC 2119 [RFC2119].

2. Context

This specification focuses on web-SSO-based flows (both SAML and OIDC), but it does not exclude non-web-browser based flows, nor does it exclude other protocols.

3. Specification

The hint parameters specified in this document is:
`aarc_ds_hint`: a hint about which DS to use, consumed by an SP or SP-IdP-Proxy.

1. The hint consumer MUST be capable of processing each of the hint parameters in GET requests.
2. The hint consumer SHOULD be capable of processing each of the hint parameters in POST requests.
3. A hint consumer MAY ignore all or part of the value of the received hint parameter.
4. The value of a hint MUST be a single URL-encoded entity identifier with the following additional rules:
   a. Forward slashes ('/') MUST be percent-encoded ([RFC3986] section-2.1]).
   b. Case sensitivity of the encoded value MUST follow the underlying specification of the original unencoded value.
5. The entity identifier[1] in an `aarc_ds_hint` MUST identify a single discovery service.
6. The `aarc_idp_hint` and `idphint` hints [AARC-G061] have precedence over the `aarc_ds_hint` if present in the same request.

References

[AARC-G045]	AARC Blueprint Architecture-2019 https://aarc-community.org/guidelines/aarc-g045/
[AARC-G061]	A specification for IdP hinting; https://aarc-community.org/guidelines/aarc-g061/
[IDPDSP]	SAML Identity Provider Discovery Service Protocol and Profile http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf
[RFC2119]	Key words for use in RFCs to indicate Requirement levels https://tools.ietf.org/html/rfc2119
[RFC3986]	Uniform Resource Identifier (URI): Generic Syntax https://tools.ietf.org/html/rfc3986
[SAML2Core]	SAML2-Core-OS §3.4.1.2 and §3.4.1.3 http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[1] Valid values of hints - i.e. the entity identifiers - are defined by the hint consumer and are communicated out of band to the hint producers.

Appendix A: Examples

The examples shown contain the actual links to be used for hinting. For clarity, we use the pseudocode that generates the URI. Backslash-escaped newlines are inserted for readability purposes only.

Examples for entity identifiers

The values of the `aarc_ds_hint` are entity identifiers. These are opaque strings which are defined by the hint consumer for the hint producer to use.

- `urn:idp:discovery`
- `https://discovery.org/idp/shibboleth`
- `my_first_discovery_service`
- `community_a_discovery`

Entities Overview

In our examples we will deal with services in different contexts. Those contexts are defined in the AARC BPA [AARC-G045] and repeated here for reference:

- Authenticating IdPs are the IdPs at the end of the authentication chain at which the user ultimately identifies.
- Community AAIs serve as IdPs or SP-IdP-Proxies for specific communities.
- Multi-tenant Community AAIs serve as SP-IdP-Proxies for multiple communities.
- Discovery Services allow users to choose an IdP for authentication.
- Infrastructure Proxies are used to connect all services of a specific infrastructure. Infrastructure Proxies may be connected to multiple Community AAIs.
- Community services are connected to a single Community AAI.
- Infrastructure service are connected to a single Infrastructure Proxy.
- Generic services may be connected to multiple Infrastructure- and/or Community AAIs.

Producing an aarc_ds_hint

Example 1. A community service behind a multi-tenant Community AAI produces a hint to inform this multi-tenant community AAI to send the user to a community specific discovery service [2]. The URL to which the service redirects the user's browser is then:

```
https://multi-tenant-aai.org/

      ?aarc_ds_hint=$(urlencode(community_a_discovery))
```

Producing a combination of an aarc_idp_hint and an aarc_ds_hint

Example 2. A community service behind a multi-tenant community AAI produces a hint to inform this multi-tenant community AAI to preferably send the user to a specific home IdP for authentication. In case this cannot be fulfilled, the user is sent to a community specific discovery service. The URL to which the service redirects the user's browser is then:

[2] Note this was initially called `disco_hint` for the fun (🕺) of it, but was deemed inconsistent naming.

```
https://multi-tenant-aai.org/

      ?aarc_idp_hint=$(urlencode(https://multi-tenant-aai.org))

      &aarc_ds_hint=$(urlencode(https://discovery.org/idp/shibboleth))
```

Note that the order of processing is not determined by the order of the parameters, but by the fact that the `aarc_idp_hint` has precedence over the `aarc_ds_hint` (see also Section 3): If the proxy uses the `aarc_idp_hint` to send the user to the hinted home IdP, the `aarc_ds_hint` is not used. If the `aarc_idp_hint` cannot be fulfilled, the `aarc_ds_hint` is used.

Producing an aarc_idp_hint with a nested aarc_ds_hint

Example 3. An infrastructure service produces a hint towards an infrastructure proxy to request sending the user to a specific multi-tenant community AAI, asking that the community AAI uses a specific discovery service. The URL to which the service redirects the users browser is then:

```
https://infra-proxy.org/

      ?aarc_idp_hint=$(urlencode(https://multi-tenant-aai.org

            ?aarc_ds_hint=$(urlencode(urn:idp:discovery))

      ))
```