SSO Infrastructure Technical Profile

Version history

Version	Date	Notes
0.1	2025-Q1	First pilot draft
0.2	2026-02-05	**Explicit target handling added** Introduced `target_value` to explicitly define the value sent via `target_parameter`. If either field is present, both are now required. **Support for relative target paths** `target_value` now supports absolute URIs as well as absolute-path references starting with `/`, optionally including query and fragment. **Improved validation of link map login URLs** In `link_mapping`, login URLs are now strictly required to be HTTPS, while IdP entityID keys may be any non-empty string (e.g. HTTPS URIs or URNs), due to legacy SAML entityID requirements.
0.3	2026-03-01	Introduced IdP Discovery Metadata feed**.** Refactored document structure for consistency and clarity.
0.3.1	2026-03-09	Updating JSON schemas to version `2020-12` and examples

- Version history
- Summary
- SP SSO Links Metadata
  - SP SSO Links Data Model
    - Organization Information (organization_info)
    - List of SSO Links (sso_links)
  - Example metadata
  - JSON schema
- IdP Discovery Metadata
  - IdP Discovery Metadata Data Model
  - Example metadata
  - JSON schema
- Validation and Encoding Requirements

Summary

This technical profile defines two independent JSON-based metadata feeds used within the SSO infrastructure:

1. *SP SSO Links Metadata Feed* – Describes Service Providers (SPs) and how SSO can be initiated.
2. *IdP Discovery Metadata Feed* – Provides a simplified Identity Provider (IdP) catalog in JSON format for UI presentation, discovery services, and portal implementations.

The feeds are independent and may be implemented separately.

Both feeds:

- MUST use `application/json`
- MUST use UTF-8 encoding
- MUST validate against their respective JSON Schemas defined in this document

SP SSO Links Metadata

The SP SSO Links Metadata feed provides information about how SSO can be initiated for Service Providers within the federation.

It supports two SSO types:

- *EntityID-based login*: Relies on the identity provider's `entityID` and parameters such as `idp_parameter`, `target_parameter`, and an SP-initiated login URL.
- *Link map-based login*: Uses explicit mappings between known IdP entityIDs and proprietary login URLs.

Each metadata entry includes descriptive and branding information about the organization and its services, including display names, logos, and landing pages, facilitating consistent presentation in discovery services and login portals.

SP SSO Links Data Model

The SP SSO Links Metadata feed MUST be a JSON array.

Each array element MUST be a JSON object containing `organization_info` and `sso_links`.

Organization Information (`organization_info`)

This section contains metadata about the organization offering the SSO services.

The `organization_info` object MUST contain the following properties:

Field	Required	Type	Description
`description`	YES	string	A short description of the organization and its services
`display_name`	YES	string	A user-friendly name of the organization
`logotype`	YES	string (URI)	The URL of the organization's logo
`name`	YES	string	The official legal name of the organization
`url`	YES	string (URI)	The official website of the organization

All properties are REQUIRED.
Additional properties MUST NOT be included.

Example

```
{
        "description": "An example description of the service provider organization",
        "display_name": "Example Service Provider",
        "logotype": "https://example.com/logo.svg",
        "name": "Example Service Provider AB",
        "url": "https://example.com"
}
```

List of SSO Links (`sso_links`)

This section contains a list of services that support SSO authentication.

`sso_links` MUST be an array containing one object per service.

Elements in each SSO link

Each SSO link object MUST contain the following properties unless otherwise specified:

Field	Required	Type	Description
`description`	YES	string	A short description of the service
`display_name`	YES	string	The name of the service displayed to users
`entity_id`	YES	string (URI)	The SAML entityID of the service
`logo`	YES	string (URI)	The URL of the service's logo
`sso_type`	YES	string	The type of SSO configuration
`disabled`	NO	boolean	Indicates whether the SSO link is disabled
`target_parameter`	NO	string	The parameter used to pass a target address
`target_value`	NO	string	The value used to pass a target address

Field Details

description (REQUIRED, string)

A short description of the service.

Example:
"Example Digital Math Service is the number one digital math companion in Schengen"

---

display_name (REQUIRED, string)

The name of the service displayed to users.

Example:
"Math Rocket"

---

entity_id (REQUIRED, string, URI)

The entity ID of the service in the SAML federation.

Example:
"https://example.com/math"

---

logo (REQUIRED, string, URI)

The URL of the service's logo.

Example:
"https://example.com/math/logo.svg"

---

sso_type (REQUIRED, string)

Defines how SSO is initiated for the service.

Allowed values:

- `"entity_id"` – Standard SAML SSO using an entity ID.
- `"link_map"` – Direct mapping between Identity Providers (IdPs) and specific login URLs.

Example:

"sso_type": "entity_id"

---

disabled (OPTIONAL, boolean)

Indicates whether the SSO link is disabled.

If omitted, the value MUST be interpreted as `false`.

Allowed values:

- `true` – The service is disabled and MUST NOT be presented to users.
- `false` – The service is active.

Example:

"disabled": false

---

target_parameter (OPTIONAL, string)

The parameter used to pass a target address after authentication.

Example:

"target_parameter": "Target"

---

target_value (OPTIONAL, string)

The value used to pass a target address after authentication.

Example:

"target_value": "https://example.com/math/resource"

If either `target_parameter` or `target_value` is present, the other MUST also be present.

`target_value` MAY be:

- An absolute URI, or
- An absolute-path reference beginning with `/`, optionally including query and fragment components, interpreted relative to the service's own origin.

If `sso_type` is `"entity_id"`

These fields are required for a standard SAML SSO setup using the entity ID as parameter for identifying the IdP.

- **`idp_parameter`** REQUIRED *(str)*
  - The parameter used to pass the IdP's entity ID in the SSO request.
  - Example: `"entityID"`.
- **`sp_init`** REQUIRED *(str, URL)*
  - The URL for SP-initiated authentication.
  - Example: `"https://example.com/math/Shibboleth.sso/Login?"`.

If `sso_type` is `"link_map"`

Fields used when the service uses direct mapping between IdPs and specific login URLs

- **`link_mapping`** REQUIRED *(object)*
  - A mapping where each IdP's entity ID is used as a key, and the value is the specific login URL for that IdP.
  - Example:

```
"link_mapping": {
        "https://idp.example1.com": "https://example.com/bjj/login?idp=example1-idp",
        "https://idp.example2.com": "https://example.com/bjj/login?idp=example2-idp",
        "https://idp.example3.com": "https://example.com/bjj/login?idp=example3-idp"
},
```

Example metadata

```
[
  {
    "organization_info": {
      "description": "An example description of the service provider organization",
      "display_name": "Example Service Provider",
      "logotype": "https://example.com/logo.svg",
      "name": "Example Service Provider AB",
      "url": "https://example.com"
    },
    "sso_links": [
      {
        "description": "Example Digital Math Service is the number one digital math companion in Schengen",
        "disabled": false,
        "display_name": "Math Rocket",
        "entity_id": "https://example.com/math",
        "idp_parameter": "entityID",
        "logo": "https://example.com/math/logo.svg",
        "sp_init": "https://example.com/math/Shibboleth.sso/Login?",
        "sso_type": "entity_id",
        "target_parameter": "Target",
        "target_value": "https://example.com/math/resource"
      },
      {
        "description": "Example Digital Brazilian Jiu-Jitsu Service is the number one digital BJJ teacher in
Schengen",
        "disabled": false,
        "display_name": "Example Digital Brazilian Jiu-Jitsu Service",
        "entity_id": "https://example.com/bjj",
        "link_mapping": {
          "https://idp.example1.com": "https://example.com/bjj/login?idp=example1-idp",
          "https://idp.example2.com": "https://example.com/bjj/login?idp=example2-idp",
          "https://idp.example3.com": "https://example.com/bjj/login?idp=example3-idp"
        },
        "logo": "https://example.com/bjj/logo.png",
        "sso_type": "link_map"
      }
    ]
  }
]
```

JSON schema

The SSO links metadata MUST validate against the JSON schema found at https://fed.skolfederation.se/pilot/schemas/sso-links-metadata-schema-0_1.json.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://fed.skolfederation.se/pilot/schemas/sso-links-metadata-schema-0_1.json",
  "title": "Skolfederation SSO Links Metadata",
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "properties": {
      "organization_info": {
        "type": "object",
        "properties": {
          "description": {
            "type": "string",
            "minLength": 1
          },
          "display_name": {
            "type": "string",
            "minLength": 1
          },
          "logotype": {
            "type": "string",
```

```
        "format": "uri",
        "pattern": "^https://"
      },
      "name": {
        "type": "string",
        "minLength": 1
      },
      "url": {
        "type": "string",
        "format": "uri",
        "pattern": "^https://"
      }
    },
    "required": [
      "description",
      "display_name",
      "logotype",
      "name",
      "url"
    ],
    "additionalProperties": false
  },
  "sso_links": {
    "type": "array",
    "minItems": 1,
    "items": {
      "type": "object",
      "properties": {
        "description": {
          "type": "string",
          "minLength": 1
        },
        "disabled": {
          "type": "boolean",
          "default": false
        },
        "display_name": {
          "type": "string",
          "minLength": 1
        },
        "entity_id": {
          "type": "string",
          "format": "uri"
        },
        "idp_parameter": {
          "type": "string",
          "minLength": 1
        },
        "logo": {
          "type": "string",
          "format": "uri",
          "pattern": "^https://"
        },
        "sp_init": {
          "type": "string",
          "format": "uri",
          "pattern": "^https://"
        },
        "sso_type": {
          "type": "string",
          "enum": [
            "entity_id",
            "link_map"
          ]
        },
        "target_parameter": {
          "type": "string",
          "minLength": 1
        },
        "target_value": {
          "$ref": "#/$defs/uriOrRelativePath"
```

```
    },
    "link_mapping": {
      "type": "object",
      "minProperties": 1,
      "patternProperties": {
        "^.+$": {
          "type": "string",
          "format": "uri",
          "pattern": "^https://"
        }
      },
      "propertyNames": {
        "type": "string",
        "minLength": 1
      },
      "additionalProperties": false
    }
  },
  "required": [
    "description",
    "display_name",
    "entity_id",
    "logo",
    "sso_type"
  ],
  "dependentRequired": {
    "target_parameter": [
      "target_value"
    ],
    "target_value": [
      "target_parameter"
    ]
  },
  "oneOf": [
    {
      "properties": {
        "sso_type": {
          "const": "entity_id"
        }
      },
      "required": [
        "idp_parameter",
        "sp_init"
      ],
      "not": {
        "required": [
          "link_mapping"
        ]
      }
    },
    {
      "properties": {
        "sso_type": {
          "const": "link_map"
        }
      },
      "required": [
        "link_mapping"
      ],
      "not": {
        "anyOf": [
          {
            "required": [
              "idp_parameter"
            ]
          },
          {
            "required": [
              "sp_init"
            ]
          }
```

```
                ]
              }
            }
          ],
          "additionalProperties": false
        }
      }
    },
    "required": [
      "organization_info",
      "sso_links"
    ],
    "additionalProperties": false
  },
  "$defs": {
    "uriOrRelativePath": {
      "anyOf": [
        {
          "type": "string",
          "format": "uri"
        },
        {
          "type": "string",
          "pattern": "^/(?!/)[^\\s]*$"
        }
      ]
    }
  }
}
```

IdP Discovery Metadata

The IdP Discovery Metadata feed provides a structured JSON representation of Identity Providers intended for consumption by:

- Discovery services (WAYF / inverse discovery)
- Login portals
- Federation UI implementations
- IdP selection interfaces

The purpose of this feed is to enable consistent and predictable presentation of Identity Providers across user-facing services.

This feed does **not** replace SAML metadata.
SAML metadata remains the authoritative source for federation trust and protocol configuration.

The IdP Discovery Metadata feed may be generated from one or more authoritative sources, including but not limited to federation SAML metadata.

IdP Discovery Metadata Data Model

The IdP Discovery Metadata feed MUST be a JSON array containing one object per Identity Provider.

Each IdP object MUST contain the following fields:

Field	Required	Type	Description
`entity_id`	YES	string (URI)	The SAML entityID of the IdP in the federation
`description`	YES	string	A short description of the organization and its services
`display_name`	YES	string	A user-friendly name for display purposes
`logotype`	YES	string (URI)	URL to the organization's logo
`name`	YES	string	The official legal name of the organization
`url`	YES	string (URI)	The official website of the organization

All properties are REQUIRED.

Additional properties MUST NOT be included.

Example metadata

```
[
  {
    "entity_id": "https://idp1.example.com/idp",
    "description": "Example Organization One Identity Provider for staff and student login.",
    "display_name": "Example Organization One",
    "logotype": "https://idp1.example.com/static/logo.svg",
    "name": "Example Organization One AB",
    "url": "https://www.example.com"
  },
  {
    "entity_id": "https://idp2.example.com/idp",
    "description": "Example Organization Two Identity Provider used for federated authentication.",
    "display_name": "Example Organization Two",
    "logotype": "https://idp2.example.com/static/logo.svg",
    "name": "Example Organization Two AB",
    "url": "https://org2.example.com"
  },
  {
    "entity_id": "https://idp3.example.com/idp",
    "description": "Example Organization Three Identity Provider supporting staff and student access.",
    "display_name": "Example Organization Three",
    "logotype": "https://idp3.example.com/static/logo.svg",
    "name": "Example Organization Three AB",
    "url": "https://org3.example.com"
  }
]
```

JSON schema

The IdP Discovery feed MUST validate against the JSON schema found at https://fed.skolfederation.se/pilot/schemas/idp-discovery-feed-schema-0_1.json

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://fed.skolfederation.se/pilot/schemas/idp-discovery-feed-schema-0_1.json",
  "title": "Skolfederation IdP Discovery metadata feed",
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "properties": {
      "entity_id": {
        "type": "string",
        "format": "uri"
      },
      "description": {
        "type": "string",
        "minLength": 1
      },
      "display_name": {
        "type": "string",
        "minLength": 1
      },
      "logotype": {
        "type": "string",
        "format": "uri",
        "pattern": "^https://"
      },
      "name": {
        "type": "string",
        "minLength": 1
      },
      "url": {
        "type": "string",
        "format": "uri",
        "pattern": "^https://"
      }
    },
    "required": [
      "entity_id",
      "description",
      "display_name",
      "logotype",
      "name",
      "url"
    ],
    "additionalProperties": false
  }
}
```

Validation and Encoding Requirements

For both metadata feeds:

- Content-Type MUST be `application/json`
- Encoding MUST be UTF-8
- Feeds MUST validate against their respective JSON Schemas