



# **Feide Integration Guide**

February 2012

Integrating a service provider with Feide

---

## Document History

Version	Date	Initials	Comments
I.0	Nov 2009	HV	First version of this document
I.1	Dec 2009	HV	Updated URLs and test information
I.2	May 2010	HV	Updated URLs
I.3	Aug 2010	HV	Updated URLs
I.4	Feb 2012	HV	Contact information for Feide updated

UNINETT  
Abels gate 5 – Teknobyen  
NO-7465 Trondheim  
telephone: +47 73 55 79 00  
fax: +47 73 55 79 01  
email: [info@uninett.no](mailto:info@uninett.no)  
web: [www.uninett.no](http://www.uninett.no)

---

## Table of Contents

<b>1</b>	<b>Introduction and purpose</b>	<b>1</b>
<b>2</b>	<b>Feide</b>	<b>3</b>
2.1	A Feide overview	3
2.2	What Feide provides	4
2.3	The user interface	6
2.4	The service provider's responsibilities	8
<b>3</b>	<b>Attributes</b>	<b>9</b>
3.1	Only necessary attributes are released	9
3.2	Attributes have limited availability	9
3.3	How to handle missing attributes	9
<b>4</b>	<b>Technical overview</b>	<b>11</b>
4.1	SAML 2.0	11
4.2	Feide architecture	12
<b>5</b>	<b>Integration</b>	<b>13</b>
5.1	Identify your authentication and attribute requirements	13
5.2	Choose and deploy SAML 2.0 Service Provider software	14
5.3	SAML 2.0 used in Feide	15
5.4	Metadata	16
5.5	Testing	17
<b>6</b>	<b>Production</b>	<b>19</b>
6.1	HTTPS	19
6.2	Logo	19
6.3	Service description	19
6.4	Log out	19
6.5	Contract	20
6.6	Opening of the service	20
6.7	Pricing	20
6.8	Feide and Shibboleth	20
<b>7</b>	<b>Feide contact information</b>	<b>21</b>
<b>8</b>	<b>References</b>	<b>23</b>

## I Introduction and purpose

Identity management is the handling of information about who and what a person is. This is becoming more important because of the increasing use of digital services. With Feide, students and employees in the educational sector get one digital identity that gives them access to web services in the educational field.

Feide is technology and platform independent, and offer all educational establishments common guidelines for identity management. A Feide name is valid throughout the Norwegian educational sector and can be used to login to all Feide services a person has access to.

The purpose of this document is to give service providers a guide on how to integrate with Feide's central login service. The Feide concept is explained. We also give a short introduction to the SAML 2.0 protocol and it's use in Feide. In addition, the document introduces the different choices a service provider integrating with Feide has to make. The goal is to simplify the Feide integration process for service providers.



## 2 Feide

Feide is a centralized identity management solution for the educational sector of Norway, and is short for «common electronic identity» (in Norwegian: «Felles Elektronisk IDEntitet»). The solution is widely used by universities, university colleges, high schools and lower education.

The Norwegian Ministry of Education and Research has chosen Feide as the sector's identity management solution. The final goal is that all students and employees will have a Feide identity. A Feide identity can be used for single sign on (SSO) to an increasing number of services connected to the central login service operated by Feide.

### 2.1 A Feide overview

Illustration 1 shows the different participants that contribute to Feide.



*Illustration 1: Feide is symbolized by the keyhole. The other participants are users, services and home organizations.*

#### 2.1.1 Users

People that use web services in their daily work at schools and universities.

#### 2.1.2 Services

These are web services offered to the educational sector. Examples of such services are learning management systems, digital learning resources, digital tests, registration systems, services selling student licences and so on.

All services connected to Feide are web services.

The services are available for subscription for chosen home organizations. Feide grants access to

each service to specific home organizations. Further authorization (e.g only employees should be given access) must be done by the service itself.

### 2.1.3 Home organizations

These are the educational institutions where students and employees have their daily work and their affiliation. For primary and secondary schools the home organizations are local or county municipalities (in Norwegian: kommuner/fylkeskommuner). For higher education it is the university colleges and universities. Private school owners can also be Feide home organizations.

All users in Feide are affiliated with home organizations. This means that persons can only get a Feide identity by being a student or employee at a home organization.

### 2.1.4 Feide

Feide maintains a central login service for the educational sector. Feide is also responsible for:

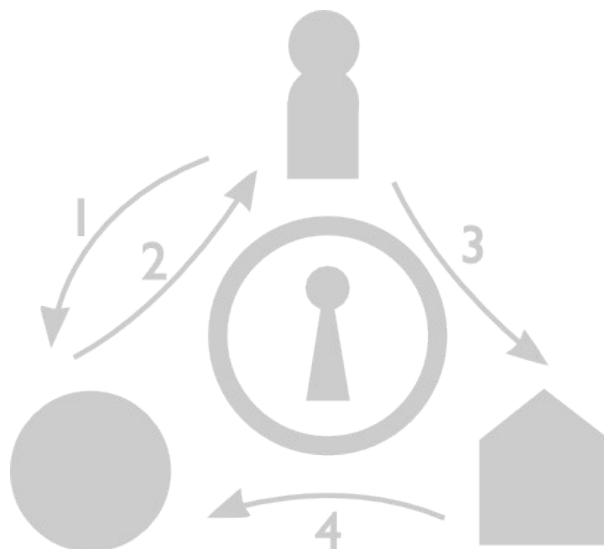
- Entering contracts with both home organizations and service providers.
- Standardizing requirements for the home organizations' identity management.

## 2.2 What Feide provides

As mentioned in the previous section, Feide runs a central login service offering single sign on (SSO), and thus provides the participants privacy protection and security.

### 2.2.1 Central login service

The central login service acts as an intermediary in the authentication process between users, services and home organizations. There is no central user store in Feide. All information about users is stored at and managed by the home organizations.



When a user logs in to a service with his Feide identity, the following steps take place:

1. The user accesses the service login page with a browser.

2. The service redirects the user to Feide's login service.
3. The user enters the Feide username and password, which Feide sends to the user's home organization.
4. Username and password are verified by the home organization, and, if they are verified by the home organization, the user's personal data can be sent to the service via Feide. Each service receives only the personal data that the service and Feide have agreed on in advance.

To read about the actual message flow, see section 5.3.1 and Feide technical guide [15].

One of the major benefits of Feide is that it facilitates single sign on (SSO): A user may authenticate once for an entire work session. After logging in once, the user can access a number of services from different service providers without having to login to each one of them.

Another advantage is that the user never gives his or her username and password to the services. Instead there is one single login page for the user.

### 2.2.2 Privacy protection

As stated earlier – all personal data are stored and managed by home organizations and there is no central user store in Feide. In the process of becoming a Feide home organization, educational institutions have to ensure that their users' personal data is correct, and review their routines and guidelines for managing personal data according to Feide standards [1].

Feide is restrictive about distributing personal data to the services. Only personal data that is necessary for the operation of the service is released to a service. A Feide service is required to make a formal agreement with Feide specifying what personal data the service should receive. When a user logs in, only the personal data agreed on will be sent to the service.

Services integrated with Feide must be restrictive in how they treat personal data. Personal data should not be distributed further. If any personal data is saved locally, measures should be taken to make sure the data is kept correct and up-to-date.

### 2.2.3 Security

One important security aspect of Feide is the distributed nature of the Feide solution. A service only receives information about the person who is logged in, and only the information that the service needs.

To provide a secure service, measures have been taken on many levels. The central Feide system is implemented on a robust platform and is subject to a strict operating and monitoring regime.

Feide's technical components have been developed by UNINETT in cooperation with the University of Oslo and other educational organizations. Feide's central login service is based on open-source code, so that everyone has access to study its functionality and security.

## 2.3 The user interface

The user interface for the Feide login process is thoroughly tested, and designed to simplify the user experience.

### 2.3.1 Login through Feide

When a user logs in to a service through Feide he/she is shown a window similar to this:

Here the user should provide his/her username (Feide ID) and password. More information about the service information shown in the login window is given in section 6.3.

If the user is on a shared computer, or the user is a student or employee of more than one institution, and wants to change affiliation, this is possible by clicking «Change?».

### 2.3.2 Choose affiliation

The first time a user logs in to a service through Feide, or he/she clicked on «Change?» in the login window, a pulldown menu similar to the one in the following window is shown:

The user has to choose which home organization he/she belongs to. The user's choice is stored in a cookie, so the user doesn't have to choose affiliation every time he/she logs in.

The drop down list corresponds to the list of home organizations that should have access to a particular service.

### 2.3.3 Give consent

The first time a user logs in to a particular service, he or she is asked to give consent to the service receiving personal information through Feide. A window similar to this is presented to the user:

English | Bokmål | Nynorsk

Nasjonal Digital Læringsarena requires that the information below is transferred.

Remember

**Information that will be sent to Nasjonal Digital Læringsarena**

<b>Surname</b>	Vada
<b>Common name</b>	Hildegunn Vada
<b>Given name</b>	Hildegunn
<b>Mail</b>	hildegunn.vada@uninett.no
<b>Person's principal name at home organization</b>	

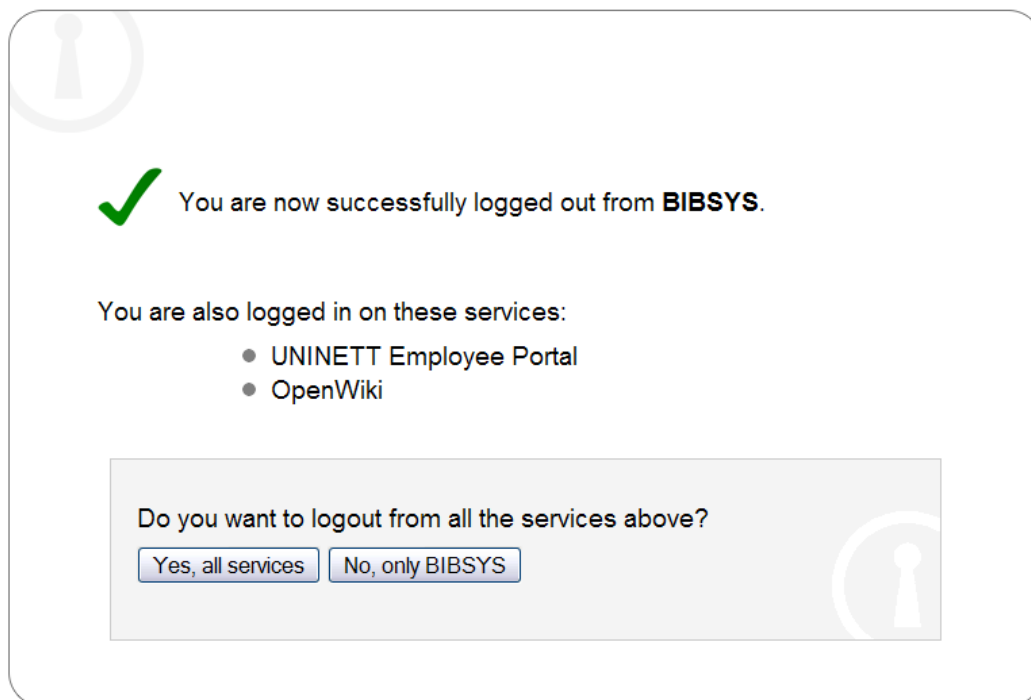
The information shown in the «Consent window» will correspond to the set of attributes that is sent to a particular service.

The user can choose to unhook the «Remember»-field if he/she wants to give consent on every login. By default, consent is stored by the Feide login service. If the user chooses not to give consent, the login process is interrupted and will not be completed.

### 2.3.4 Single logout

Single logout means that when you hit the logout-button, you logout from all Feide services. Sometimes this is not what the user wants. To enhance user experience Feide has implemented a log out solution that is both secure and user friendly.

When a user tries to log out from a particular service, the user gets a list of all other services the user is logged into at this moment, and the question: «Do you want to log out from all the services above?». The user can then choose whether he wants to log out from all services, or just this particular one. This is shown in the following screenshot:



## 2.4 The service provider's responsibilities

As stated earlier, Feide provides authentication and release of some personal data. There are several responsibilities that the service providers must take care of:

- Authorization of users. Based on the authentication response and the personal data provided by Feide, the service provider must decide whether or not the user should get access to the service.
- All aspects concerning operation, management and development of the service.
- If the service requires any payment from, or other formal agreements with, home organizations, the service provider has to communicate directly with home organizations.

### 3 Attributes

During the login process the service can receive personal data about the user and the user's home organization. We refer to the personal data as attributes. Attributes available through Feide are defined in «norEdu\* Object Class Specification» [14].

For technical details on attributes, see [15].

#### 3.1 Only necessary attributes are released

Feide will release to the service only those user attributes for which the service can demonstrate a need. The service provider and Feide must in advance make an agreement on what user attributes the service is going to receive from Feide. The selection is specified in the service description form (see section 6.3 for more information).

A service's needs may be as simple as to restrict access to users who can present a valid user name and password. Other services would like to make user specific adaptation, or require user specific information to determine access rights or authorization.

#### 3.2 Attributes have limited availability

In some cases, the service will not receive all the attributes it has requested. Some attributes are mandatory for home organizations to register, but most are optional. The service should not rely on optional attributes being available for all users.

Mandatory attributes can also have limited availability. The reasons for this are:

- A user may, for privacy reasons, request that for him- or herself, one or more attributes are *not* to be forwarded by Feide (similar to having an «unlisted» telephone number).
- Some attribute values may be absent because no value is applicable, e.g. because the user doesn't have any telephone, email account or web home page.
- Even if an attribute is mandatory in the current specification, the user's home organization may use an old version of the specification where the attribute was optional.

#### 3.3 How to handle missing attributes

A service may handle an essential, but missing parameter according to different strategies, e.g. by:

- Providing restricted functionality, such as allowing library catalog search, but disabling any functions for email notification if no email address attribute is available for the requesting user.
- Allowing the user to interactively enter the missing information, to enable the full functionality. If this alternative is chosen, either the value entered by the user should be deleted as soon as it is no longer needed, or it should be made clear to the user (before the user enters the information) that the information will be stored.
- Terminating after having informed the user the reason why continued operation is not possible.

If at all possible, either of the two first alternatives are preferred instead of the third option.



## 4 Technical overview

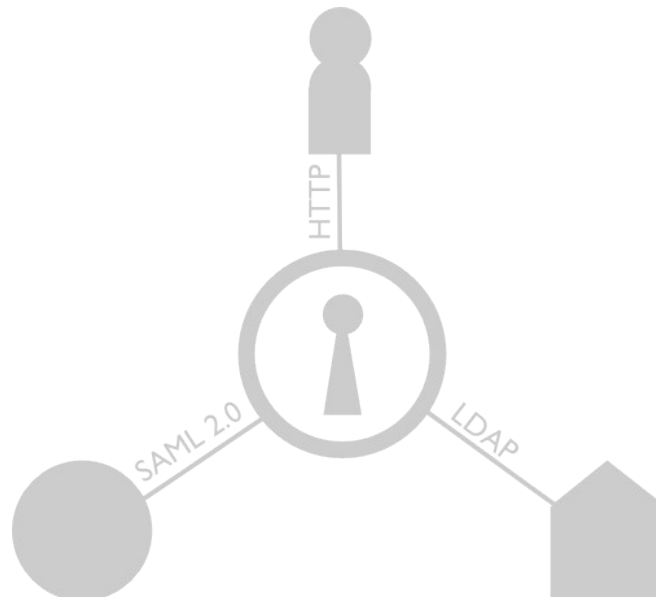


Illustration 2: Protocols used in Feide

Illustration 2 shows the protocols used in Feide. For the service to communicate with Feide, it must use the SAML 2.0 protocol [13]. Between Feide and home organizations LDAP is used. Between users and Feide HTTP is used. But service providers only need to focus on SAML 2.0.

### 4.1 SAML 2.0

The Security Assertion Markup Language (SAML), developed by OASIS, is an XML-based framework for communicating user authentication, entitlement, and attribute information.

In the identity management field, we use the following terms:

- A Service Provider (**SP**) represents the service.
- An Identity Provider (**IdP**) authenticates users. Feide is an identity provider.
- A **federation** is a collection of SPs and IdPs in a trust relationship.

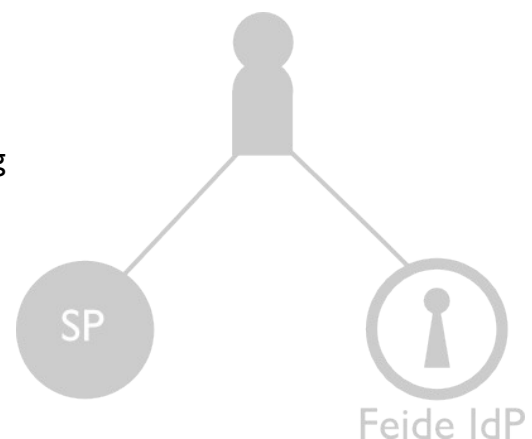


Illustration 3: Basic SAML SSO model

SAML enables web single sign on (SSO) through the communication of an authentication assertion which contains information about the user, as shown in illustration 3. The assertion is sent from the IdP to the SP. The service can verify the origin of the assertion, and choose whether to allow access to the user.

## 4.2 Feide architecture

In the Feide federation there is a single IdP – the central login service. This means that all the home organizations share one common IdP. All SPs only connect to this IdP. This makes it very easy for SPs to reach many users by only connecting with a single IdP. Illustration 4 shows this.



Illustration 4: Feide architecture. The Feide keyhole symbolizes the IdP.

Note that this is from a technical point of view. Service providers decide which organizations should be able to subscribe to the service through Feide. If the service requires payment, formal agreements or contracts, the service provider has to communicate directly with home organizations.

## 5 Integration

This section gives an overview on how to integrate your service with Feide. There are five steps:

1. Identify your authentication and attribute requirements.
2. Choose and deploy an appropriate SAML software package for your service.
3. Enter the test phase.
4. Prepare your service for production.
5. Sign the contract. Your service is now ready for Feide authentication.

### 5.1 Identify your authentication and attribute requirements

#### 5.1.1 Identify authentication requirements

An important task is to identify your authentication requirements. Is Feide authentication used as base for:

- authorization
- storing user profiles
- customizing content

or maybe all of these together? Other questions of importance are:

- Should users be able to anonymously access your service?
- Do you have to combine Feide authentication with local authentication?
- What identifier do you want to use to map Feide users to local user accounts?

#### 5.1.2 Identify attribute requirements

When the authentication requirements are determined, it's time to identify which user attributes the service needs. See section 3 for more information about attributes. Feide is restrictive about releasing attributes; if you have any questions – please discuss it with the Feide staff.

Some attributes are singlevalued, others are multivalued. Here are some examples on what the service will receive from Feide when a fictional student «Lise Hansen Berg» logs in:

- **norEduPersonNIN** (national identity number): This attribute is singlevalued. The service will receive «03088248201».
- **sn** (surname): This attribute is multivalued. The service will receive the values «Berg» and «Hansen» in no specific order.
- **eduPersonTargetedID** (a persistent privacy-perserving attribute). This is an opaque identifier which doesn't reveal anything about the user. The service will receive a single value on the form «kl83HlsnblqYskgh72Kfqkl».

## 5.2 Choose and deploy SAML 2.0 Service Provider software

We recommend that you use pre-made SAML 2.0 software for Feide integration. When considering different software alternatives, you should consider the following questions:

- What web server is used for the service?
- What OS is used for the service?
- What programming language is the service written in?

When you know your requirements, you should consider the different alternatives of software for Feide integration. We encourage service providers in this phase to inform us on both your requirements and what software you have selected.

In this section we give an overview on use of existing SAML 2.0 SP software, some interfaces between your service and the SAML 2.0 SP software, and some examples on such software packages.

### 5.2.1 Use existing SAML 2.0 SP software

Since the SAML 2.0 protocol is a rather complex protocol, we recommend that you use a pre-made SAML 2.0 service provider (SP) software component instead of implementing it yourself. Such a software component or library will handle all the communication with Feide, and leaves a simple API / interface towards your service. This software is communicating with Feide using SAML 2.0. This is illustrated in Illustration 5.

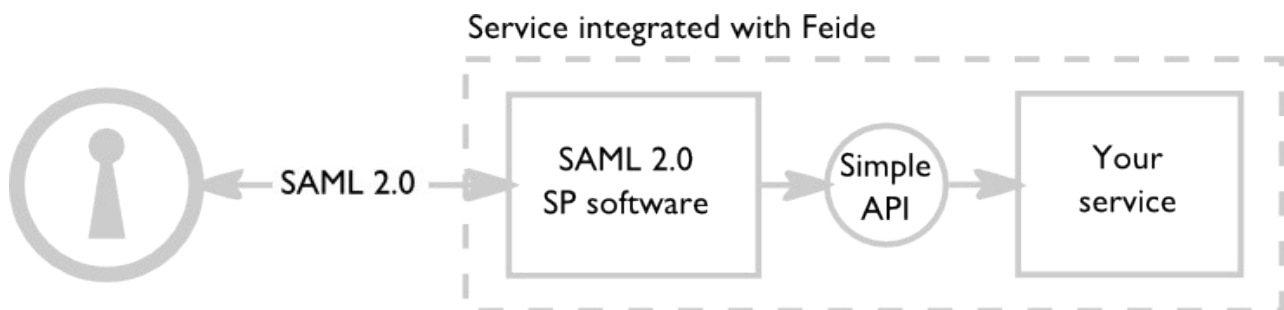


Illustration 5: Your application using a simple API towards a SAML 2.0 SP software component.

### 5.2.2 Interfaces between SAML 2.0 SP software and your application

The interface between the SAML 2.0 SP software and the service will differ from software to software. This section gives some general examples.

One of the typical interfaces to SAML 2.0 SP software is using a native programming interface in the same programming language that your service is written in. For example, if your service is written in Java, you may include an interface in form of a jar-file into your application, and access some classes to ask whether the user is authenticated or not. An example (from a fictional library):

```
if (SAML20spAPI.isUserAuthenticated()) {
    String userid = SAML20spAPI.getAttribute("eduPersonPrincipalName");
} else {
    SAML20spAPI.authenticateUser(); return;
}
```

If the SP software is written in another programming language than your service there are other ways for them to communicate:

- The SAML 2.0 SP software is implemented as a module to your web server. E.g. the SAML 2.0 SP is an Apache module or a Microsoft IIS authentication module. In this scenario, the standard way of providing authentication information is based on the web server. In Apache, this could be the REMOTE\_USER header.
- Information about authentication is often inserted into the HTTP request header, and can be retrieved as any other HTTP header from the user. Examples of this are a java servlet filter or a reverse proxy.

### 5.2.3 SAML 2.0 SP software packages

There are several vendors of SAML 2.0 software packages, both open source and commercial. For commercial products, see Liberty Alliance [2].

Here is a list of some open source implementations available:

- **simpleSAMLphp** [3]: A PHP implementation of SAML 2.0 SP. Further development is managed by Feide personell, with contributions from a handful of other institutions.
- **Sun OpenSSO** [4]: A project from Sun written in Java.
- **mod\_mellon** [5]: An Apache module that makes use for the Lasso library to provide authentication for web sites that uses the Apache web server.
- **Shibboleth 2.X** [6]: An alternative developed under the Internet2 umbrella. Available as an Apache module and a Java application. Note that Shibboleth version 1.3 uses an older protocol that is not compatible with Feide.
- **SAML2 API** [7]: A java-library for SAML 2 authenticating.
- **OIOSAML.NET** [8]: A .Net based SAML 2.0 implementation for Microsoft Internet Information Server. Is used for federation in the Danish public sector.

## 5.3 SAML 2.0 used in Feide

To enhance interoperability, Feide uses two deployment profiles of the SAML 2.0 protocol:

- For the authentication process: «Interoperable SAML 2.0 Web Browser SSO Deployment Profile» [9].
- For the logout process: «Front-channel SAML 2.0 Single Logout Deployment Profile» [10].

This section gives you a short summary of the Feide SAML 2.0 profiles. Read more about this in

«Feide technical guide» [15].

### 5.3.1 SAML messages and bindings

For the login process the Service Provider (SP) must handle the following messages:

- **Authentication request:** The authentication request issued by the SP must be sent to the Feide Identity Provider (IdP) using the `HTTP-Redirect` binding.
- **Authentication response:** Authentication response messages are sent to the SP from the IdP using the `HTTP-POST` binding.

For the logout process the SP must handle the following messages:

- **Logout request:** The logout request must be sent to the IdP using the `HTTP-Redirect` binding. The SP must also be able to handle incoming logout requests from the IdP.
- **Logout response.** The logout response must be sent to the IdP using the `HTTP-Redirect` binding. The SP must also be able to handle incoming logout responses from the IdP.

Read more about the bindings in «SAML 2.0 Bindings» [11].

In addition the service must be able to receive attributes that comes with the authentication response message:

- No extra messages are exchanged for attribute release; the service does not explicitly ask for attributes during the login process.
- The service only sends an authentication request, and both the authentication assertion and agreed-upon attributes are sent to the service in the authentication response.

### 5.3.2 Signing of SAML messages

Feide assumes no signing of SAML messages by SPs. Since Feide's SAML 2.0 deployment profiles only allows the use of the front-channel `HTTP-Redirect` binding for the authentication request, logout request and logout response, signing of these messages to ensure integrity is not considered necessary.

If the SP is signing the authentication request, it cannot assume that Feide is validating the signature unless an explicit agreement is made about doing so.

Feide requires that the service uses HTTPS, and will only send authentication responses to HTTPS-enabled endpoints.

The IdP will sign the authentication response. SPs must check the signature of incoming authentication responses to ensure that it is sent from Feide.

## 5.4 Metadata

Sending messages between Feide and your service is done by HTTP calls addressed to specified URLs; these URLs are called endpoints. The metadata for your service specifies these endpoints. Feide needs your metadata to send messages to your service. For your service to send messages to the Feide IdP, you need Feide's metadata.

### 5.4.1 Metadata contents

Metadata is specified as an XML document according to a schema which is part of the SAML 2.0 standard. For more details, see the SAML 2.0 Metadata specification [12].

Feide requires the following metadata entries to be set:

- **entityID.** This is a unique ID for your service. It should either be a URN or a URL. A URN entityID should be on the form "urn:mace:feide.no:services:no.example.service". If the entityID is a URL it should be a URL pointing to the service's metadata.
- **AssertionConsumerService.** This is the URL where authentication responses sent from the IdP will be posted.
- **SingleLogoutService.** This is the URL where logout requests and responses are processed.

Read more about the metadata in «Feide technical guide» [15].

### 5.4.2 Submitting metadata to Feide

Your metadata must be delivered to Feide by email to [support@feide.no](mailto:support@feide.no) and loaded into the Feide login server, before we can accept authentication requests from your service. It is recommended that you zip your metadata or put it as an attachment to the email.

For security reasons, and to check that metadata are reasonable and correct, loading metadata into the login server is a manual operation. So metadata should be prepared and delivered to Feide well before you plan to start testing (normally the next business day after metadata was submitted).

### 5.4.3 Feide login service (IdP) metadata

Your service needs information about how to contact the Feide login service. The format of this information is similar to the metadata for your service. Metadata for the Feide login service is common for all service providers. A copy can be obtained here:

- <https://idp.feide.no/simplesaml/saml2/idp/metadata.php?output=xhtml>

Please note that Feide's metadata can be changed, and in that case you must be able to quickly update these on your side. You will find more information about Feide's requirements regarding contact information in section 6.3.

For testing purposes, you should use the Feide test environment. Metadata can be obtained here:

- <https://idp-test.feide.no/simplesaml/saml2/idp/metadata.php?output=xhtml>

## 5.5 Testing

After installing and integrating your SAML 2.0 software, you are ready for testing. Follow these steps to test your service:

- I. Send the following to [support@feide.no](mailto:support@feide.no):
  - Your (test) metadata. Please be sure to specify that this is metadata for testing, since it has to be manually loaded into our test server.

- The set of attributes that your service should receive. As described in section 3 and 5.1.2, you need to justify this selection.
2. Obtain metadata for Feide test login service here:  
<https://idp-test.feide.no/simplesaml/saml2/idp/metadata.php?output=xhtml>
  3. Wait for confirmation from the Feide staff that your metadata has been loaded into the test login server.
  4. You can now test logging on to your test service with a valid Feide ID. If you don't have a Feide ID, contact the Feide staff to be assigned a temporary Feide ID and password for test purposes. For more details about the test process, see Feide Technical Guide [15].

Note that we recommend you start the contract signing process at the same time as you start testing. See section 6.5 for more information.

## 6 Production

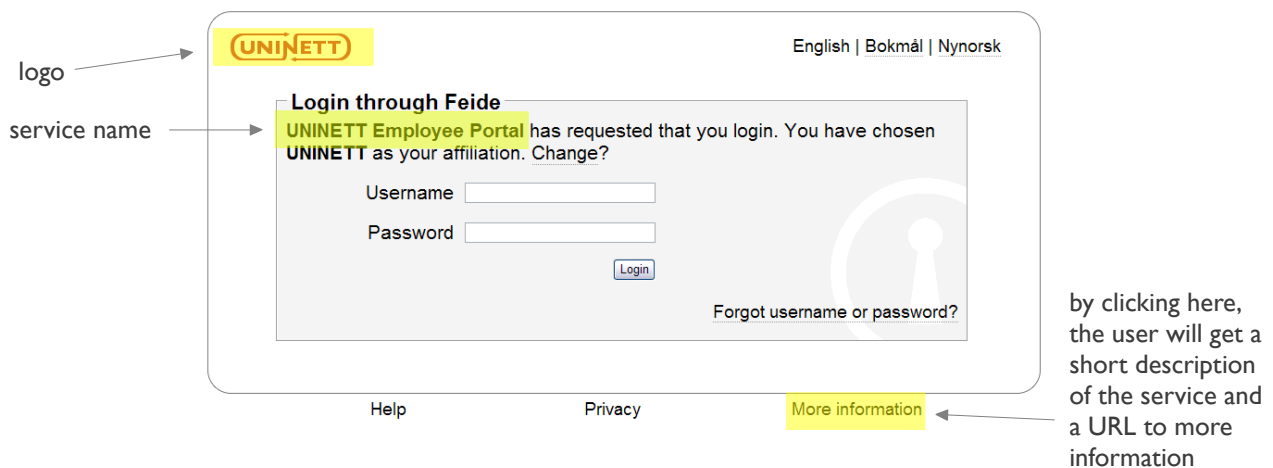
Before the service is ready to move from test to production environment, Feide has some requirements.

### 6.1 HTTPS

Because of Feide's use of SAML bindings, we require that the service uses HTTPS when in production.

### 6.2 Logo

The picture below shows the Feide login user interface. To customize this to your service we need your service logo. The logo should preferably have white/transparent background and a height of 50-60 pixels. The logo size should not exceed 400 (width) x 100 (height) pixels.



### 6.3 Service description

When your service is ready for the production environment, we will send you a service description form. The following items must be entered in the form:

- Service name both in Norwegian and English (if different).
- Short description of the service both in Norwegian and English.
- A URL to more information about the service.
- Technical contact. Note that we do not want personal email addresses.
- If available, a direct URL to login to the service.
- A final list of attributes that the service will receive when a user logs on.

### 6.4 Log out

Since Feide requires single logout, a visible logout link must be available to users of the service.

## 6.5 Contract

We recommend that you start the application process when your service is ready for testing. You will find the Feide application in [17]. After processing your application, we will send you two copies of the contract.

Two signed copies of the contract must be returned to Feide. Feide will then process the contract, and one copy is returned to you.

## 6.6 Opening of the service

The service provider has to inform Feide about which home organizations should be able to subscribe to your service. We will then make it possible for the home organizations to subscribe to your service, and we will inform the home organizations about your service.

The home organizations can then choose to subscribe to your service. As mentioned earlier, the service provider has to communicate directly with the home organizations about any payment, agreements outside of Feide, etc.

Based on Feide authentication and possible authorization in your service, users will then be able to use your service.

## 6.7 Pricing

For the Feide price list, see [16].

## 6.8 Feide and Shibboleth

Feide and Shibboleth federations are based on the same concepts, but Shibboleth federations are somewhat different from Feide. In Shibboleth, it is common to have a mesh of IdPs and SPs, where each service that wants to integrate with a new institution must talk to the different organizations that operate IdPs.

Feide operates one central IdP. A service integrates once with this central IdP, and Feide adjusts access according to the service provider's and home organization's requests.

If your service already is integrated with a Shibboleth federation, you can easily integrate with Feide (or the other way around) if the following is in place:

- Feide uses the SAML 2.0 protocol. You need to ensure that your service supports SAML 2.0, older versions are not compatible with Feide.
- Feide requires the use of HTTPS.
- Feide requires support for single logout.

## 7 Feide contact information

- Questions regarding attributes, SAML 2.0 and metadata exchange: [support@feide.no](mailto:support@feide.no)
- Feide's home page: <http://www.feide.no>



## 8 References

- [1] «Feide system architecture» - [http://www.feide.no/sites/feide.no/files/documents/Feide\\_systemarkitektur.pdf](http://www.feide.no/sites/feide.no/files/documents/Feide_systemarkitektur.pdf)
- [2] Liberty interoperable implementations table - [http://www.projectliberty.org/liberty/liberty\\_interoperable/implementations/saml\\_2\\_0\\_test\\_procedure\\_v1\\_0\\_interoperable\\_implementation\\_table](http://www.projectliberty.org/liberty/liberty_interoperable/implementations/saml_2_0_test_procedure_v1_0_interoperable_implementation_table)
- [3] The simpleSAMLphp homepage - <http://simplesamlphp.org/>
- [4] The OpenSSO homepage - <https://opensso.dev.java.net/>
- [5] The modmellon homepage - <http://code.google.com/p/modmellon/>
- [6] The Shibboleth homepage - <http://shibboleth.internet2.edu/>
- [7] The SAML2 API homepage - <http://sourceforge.net/projects/saml2api/>
- [8] The OIOSAML.NET homepage - <http://www.softwareborsen.dk/projekter/softwarecenter/brugerstyring/oiosaml.net/>
- [9] «Interoperable SAML 2.0 Web Browser SSO Deployment Profile» - <http://saml2int.org/profile/current>
- [10] «Front-Channel Single Logout Deployment Profile» - <http://rnd.feide.no/content/front-channel-single-logout-deployment-profile>
- [11] «Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0» - <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>
- [12] «Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0» - <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>
- [13] «Security Asserption Markup Language (SAML) V2.0 Technical Overview» - <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>
- [14] «norEdu\* Object Class Specification» - [http://www.feide.no/sites/feide.no/files/documents/norEdu\\_spec.pdf](http://www.feide.no/sites/feide.no/files/documents/norEdu_spec.pdf)
- [15] «Feide technical guide» - [http://www.feide.no/sites/feide.no/files/documents/Feide\\_technical\\_guide.pdf](http://www.feide.no/sites/feide.no/files/documents/Feide_technical_guide.pdf)
- [16] In English: «Feide Service Provider Pricing» - <http://www.feide.no/service-provider-prices>  
In Norwegian: «Priser for Feide tjenesteleverandører» - <http://www.feide.no/priser-tjenesteleverandorer>
- [17] In English: «Application for association with Feide as a Service Provider» - <http://www.feide.no/sites/feide.no/files/documents/Application%20form.pdf>  
In Norwegian: «Søknad om tilknytning som tjenesteeier i Feide» - <http://www.feide.no/soknad-tjenesteleverandor>

