Andreas Åkre Solberg, Wednesday 3 September 2014

UNINETT

Feide Connect

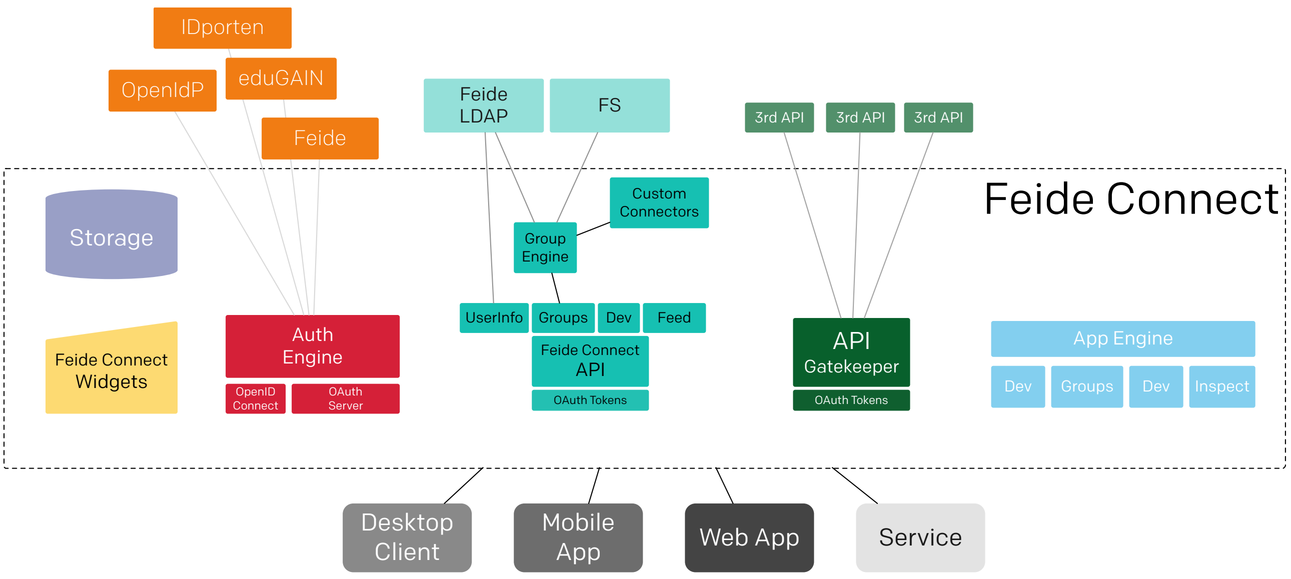
Technical architecture

Introduction

Feide Connect consists of four main components:

- **Auth Engine** authenticates and authorizes a user upon request from a registered client. After users permission, an access token is
- **Core API** is offering a set of APIs, such as userinfo, groups and people search.
- **API Gatekeeper** offers authorization management to external APIs.
- **App Engine** and **Apps** - represents user interfaces and self service.

In addition Feide Connect is offering support for **Federated Widgets**, a concept where a service may offer widgets to be embedded into other web services.



Auth Engine

The Auth Engine compoment implements an OAuth 2.0 Server. It receives an incomming OAuth 2.0 Authorization Request from a pre-registered client. Clients may be web, desktop or mobile applicaitons, extending the range of options from the traditional Feide SAML 2.0 interface.

Next, the auth engine will ensure that the end user is authenticated, acting as a SAML 2.0 Service Provider connected to an Identity Provider, such as Feide. If more than one Identity Provider is enabled, a discovery service, DiscoJuice, is enabled to allow the user to select which provider to use for login.

In addition to Feide, these providers are of interest for Feide Connect:

- Guest login, using Feide OpenIdP
- Kalmar Union – cross-federated login in nordic countries
- eduGAIN – cross-federated login in Europe

From an Identity Provider, the auth engine will extract a set of one or more secondary user keys. In example, when login through Feide, I will be represented these two identifiers:

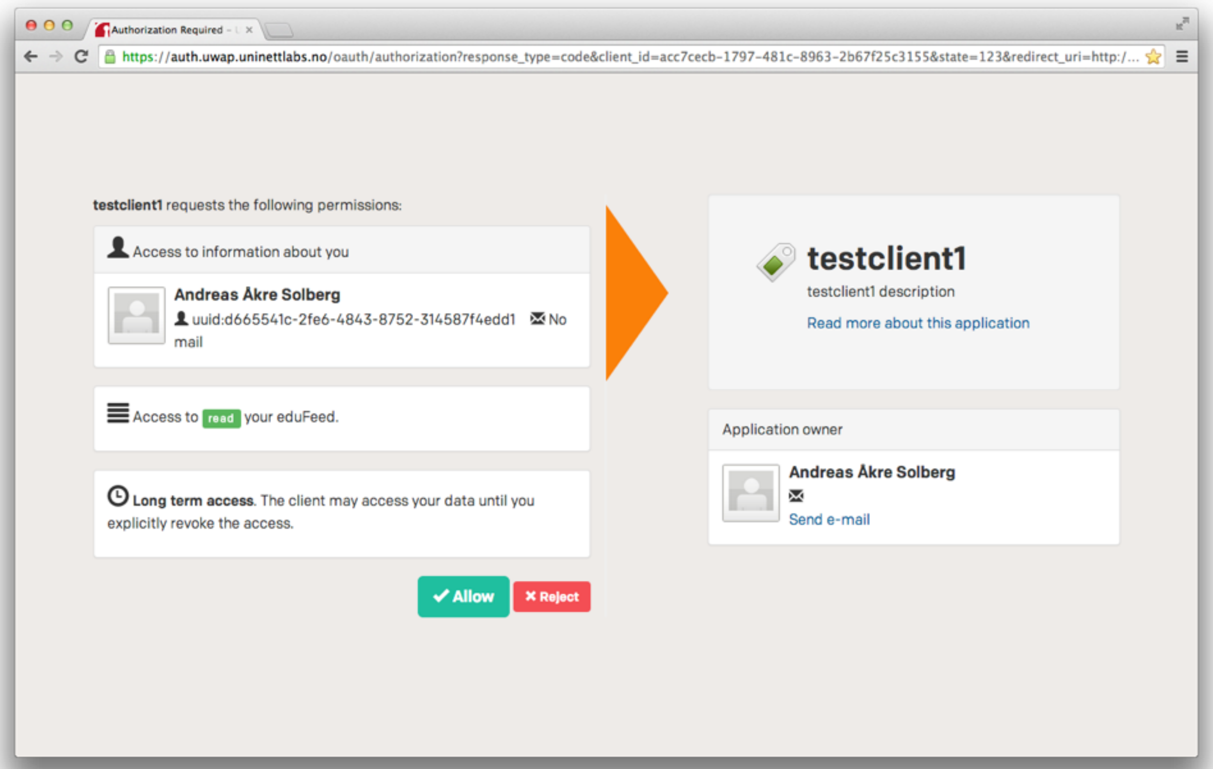
feide:andreas@uninett.no
nin:10108099999

Feide Connect maintains a user store of user identifiers and a very limited set of cached user attributes. Here is an example of a stored user object:

Display Name	Andreas Åkre Solberg
Primary key	uuid:ff0b3590-dcbd-47b7-a454-ac8233d5c984
Secondary keys	feide:andreas@uninett.no feide:andrs@ntnu.no nin:10108099999
Mobile	41107700
Profile photo	

Most of the user information obtained from Feide will not be associoated with the user object, but instead as role membership attributes to the organization(s) that the user is associated with.

Feide Connect operates with user identities that represents persons and not user roles. When the user is successfully authenticated, and the requesting client is verified, the user is presented with a informative authorization grant web page, asking the user to accept specified permissions to be given to the requested client.


Authorization Required

https://auth.uwap.uninettlabs.no/oauth/authorization?response_type=code&client_id=acc7cecb-1797-481c-8963-2b67f25c3155&state=123&redirect_uri=http:/...

testclient1 requests the following permissions:

Access to information about you

Andreas Åkre Solberg
uuid:d665541c-2fe6-4843-8752-314587f4edd1 No mail

Access to read your eduFeed.

Long term access. The client may access your data until you explicitly revoke the access.

✓ Allow ✗ Reject

testclient1
testclient1 description
Read more about this application

Application owner

Andreas Åkre Solberg
Send e-mail

Next, an OAuth 2.0 Access Token, is issued, stored, and sent back to the requesting client. The token is associated with a user, a client and a set of permissions. The token will usually expire after a predefined time, but a client may also request permanent tokens. An end-user may easily review all issued tokens, and may instantly pull-back granted access.

Here is an example of the cached access token in the Feide Connect storage:

```
    {
      "issued": new Date("2014-09-02T02:10:54+0200"),
      "validuntil": new Date("2014-09-02T10:10:54+0200"),
      "client_id": "415a44da-75b5-496d-8e9f-da5182ac6cd0",
      "userid": "uuid:cdea2433-3c7f-47dd-90d2-178b5522bd5d",
      "access_token": "4b1b7c97-d1de-43fb-9eb7-d15845b3ce0f",
      "token_type": "bearer",
      "scope": [
        "appconfig",
        "userinfo"
      ]
    }
```

Feide Connect aims toward supporting OpenID Connect 1.0. OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol. OpenID Connect is a very fresh protocol, but with significant momentum. Feide Connect should support the very basic level of OpenID Connect, including authorization code flow and implicit flow.

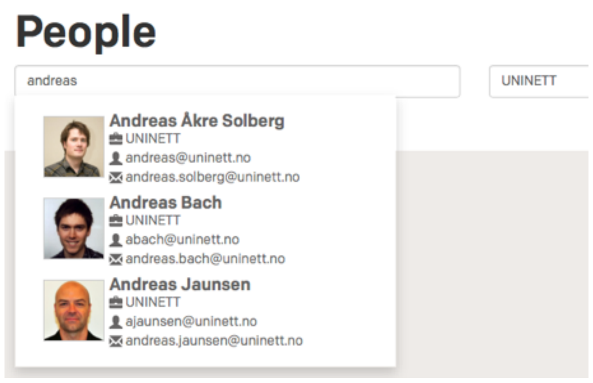
Feide Connect needs to implement a good user experience wrt termination of sessions. There is no support for Single Logout in OAuth 2.0. Feide Connect needs to define a userfriendly approach to handle incomming logout requests from Feide.

Core APIs

Feide Connect offers three public APIs:

- **People search** allows a client to search for people at an educational instituion.
- **Userinfo** gives access to user information about the current user.
- **Groups** exposes information about a users group memberships.

The **people search** API may be used to realize search boxes or dropdowns for end-users to build collaboration groups, or select individuals to share a resource with. Feide Connect uses the Feide-LDAP-directory to search for users. Access control may be configured per institution on who can search for who. In example students may only search for students at the same institution.



People

andreas

UNINETT

Andreas Åkre Solberg
UNINETT
andreas@uninett.no
andreas.solberg@uninett.no

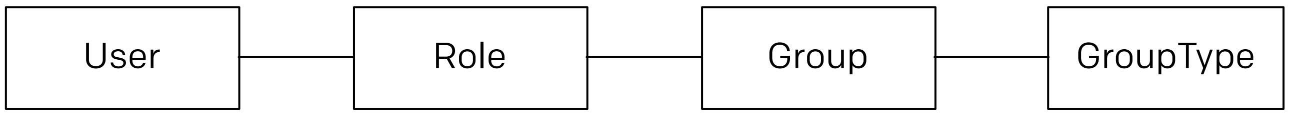
Andreas Bach
UNINETT
abach@uninett.no
andreas.bach@uninett.no

Andreas Jaunsen
UNINETT
ajaunsen@uninett.no
andreas.jaunsen@uninett.no

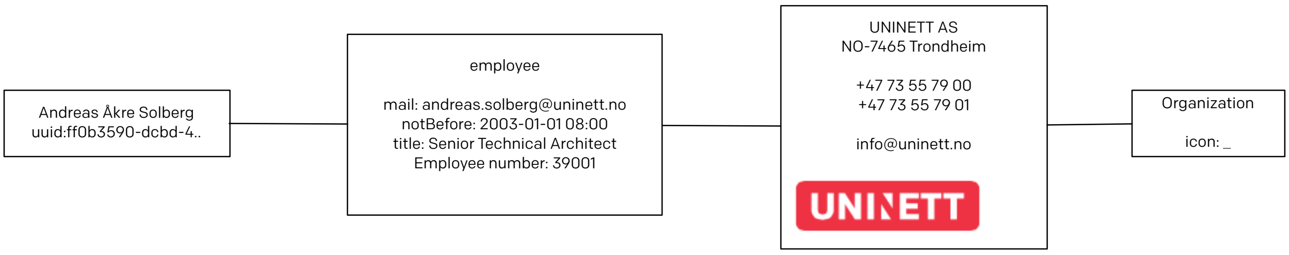
The **groups** API exposes mainly two protocol requests:

- Get a list of all group memberships of the current user.
- Get information about the specific group, and its members.

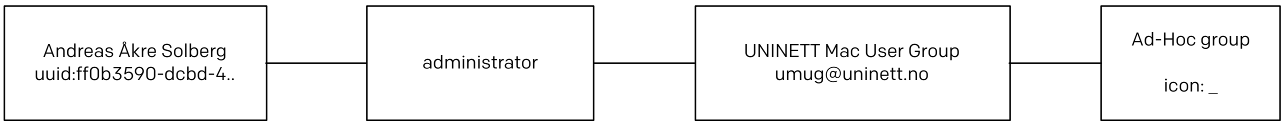
The information model for groups is defining users, roles, groups and group types. All groups is of a specific type. Example of group types are: organization, organization unit, ad-hoc, cohort, study, subjects.

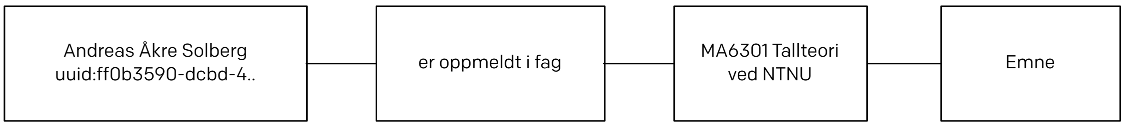


Here is an example of a user's membership to a group of type organization, an employeer at UNINETT. In this example is most of the attributs derived from Feide associated with the role object, rather than the user object.



Here is two other examples, of an ad-hoc group and a subject.





The group API is a harmonized presentation of data from multiple distributed sources, including:

- **Ad-hoc** groups. Any user may create and manage groups with other people in order to established a shared digital collaboration. These ad-hoc groups are stored within Feide Connect, and there is a user interface to manage groups as part of the Feide Connect.
- **Feide directory** is the source of organizations and organizational units.
- Feide Connect has a specific integration with **FS**, for exposing groups for cohorts, studies and subject.
- Institutions may also offer **custom connectors** to provide groups from existing authorative sources.

API Gatekeeper

Feide Connect offers API Gatekeeper functionality to API owners, that would like to facilitate self-registration, authorization workflow, authentication and more from the Feide Connect platform.

An API owner, controlling an API hosted at example.org, may register an API Gatekeeper for this API using the developer dashboard. The API will be publicly exposed through a deidicated subdomain, such as example.uwap.org. All requests to this API is required to contain a valid Feide Connect access token. Client developers, may easily register new clients, and request access to this API. The API owner may if needed grant access to newly registered APIs.

An API request from a third party client, with an issued access token, may perform a request like this:

```
GET /my-profile HTTP/1.0
Host: example.uwap.org
Authorization: Bearer abcd-1234-abcd-6789
```

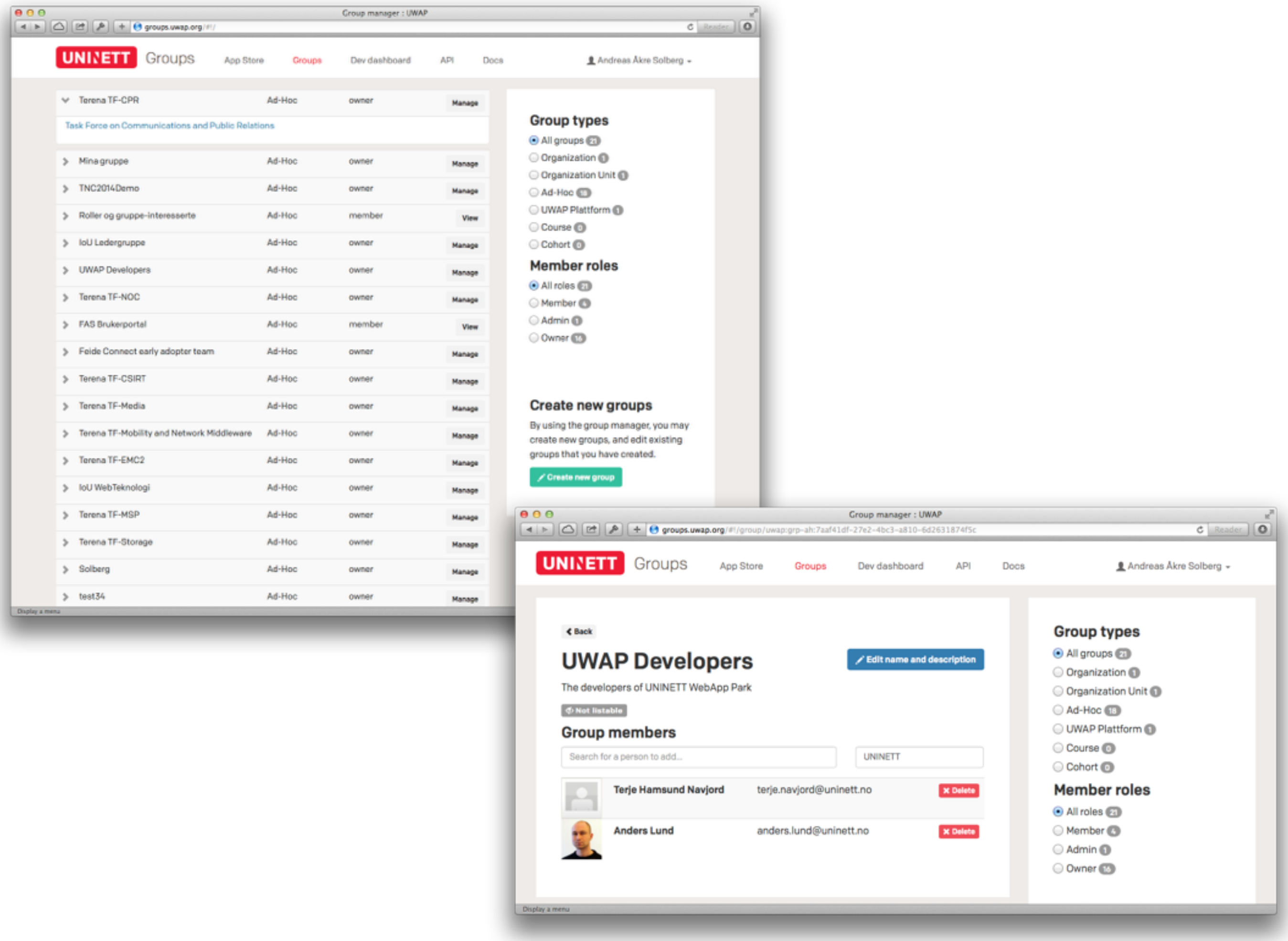
This request will then be passed on to example.org, with additional information about the user and client:

```
GET /my-profile HTTP/1.0
Host: example.uwap.org
X-Connect-UserID: uuid:ff0b3590-dcbd-47b7-a454-ac8233d5c984
X-Connect-DisplayName: Andreas Åkre Solberg
X-Connect-Groups: org:uninett,adhoc:12341234
X-Connect-ClientID: testclient1
```

App Engine

The App Engine is a simple app environment for hosting a set of built-in web apps. These web-apps together implements self-service of all parts of the platform.

The groups app will allow end-users to create and mange ad-hoc groups:



Terena TF-CPR	Ad-Hoc	owner	Manage
Task Force on Communications and Public Relations			
Mina gruppe	Ad-Hoc	owner	Manage
TNC2014Demo	Ad-Hoc	owner	Manage
Roller og gruppe-interesserte	Ad-Hoc	member	View
IoU Ledergruppe	Ad-Hoc	owner	Manage
UWAP Developers	Ad-Hoc	owner	Manage
Terena TF-NOC	Ad-Hoc	owner	Manage
FAS Brukerportal	Ad-Hoc	member	View
Feide Connect early adopter team	Ad-Hoc	owner	Manage
Terena TF-CSIRT	Ad-Hoc	owner	Manage
Terena TF-Media	Ad-Hoc	owner	Manage
Terena TF-Mobility and Network Middleware	Ad-Hoc	owner	Manage
Terena TF-EMC2	Ad-Hoc	owner	Manage
IoU WebTeknologi	Ad-Hoc	owner	Manage
Terena TF-MSP	Ad-Hoc	owner	Manage
Terena TF-Storage	Ad-Hoc	owner	Manage
Solberg	Ad-Hoc	owner	Manage
test34	Ad-Hoc	owner	Manage

Group types

- ● All groups 21
- ○ Organization 1
- ○ Organization Unit 1
- ○ Ad-Hoc 18
- ○ UWAP Plattform 1
- ○ Course 0
- ○ Cohort 0

Member roles

- ● All roles 21
- ○ Member 4
- ○ Admin 1
- ○ Owner 16

Create new groups

By using the group manager, you may create new groups, and edit existing groups that you have created.

UWAP Developers

The developers of UNINETT WebApp Park

Group members

Terje Hamsund Navjord	terje.navjord@uninett.no	Delete
Anders Lund	anders.lund@uninett.no	Delete

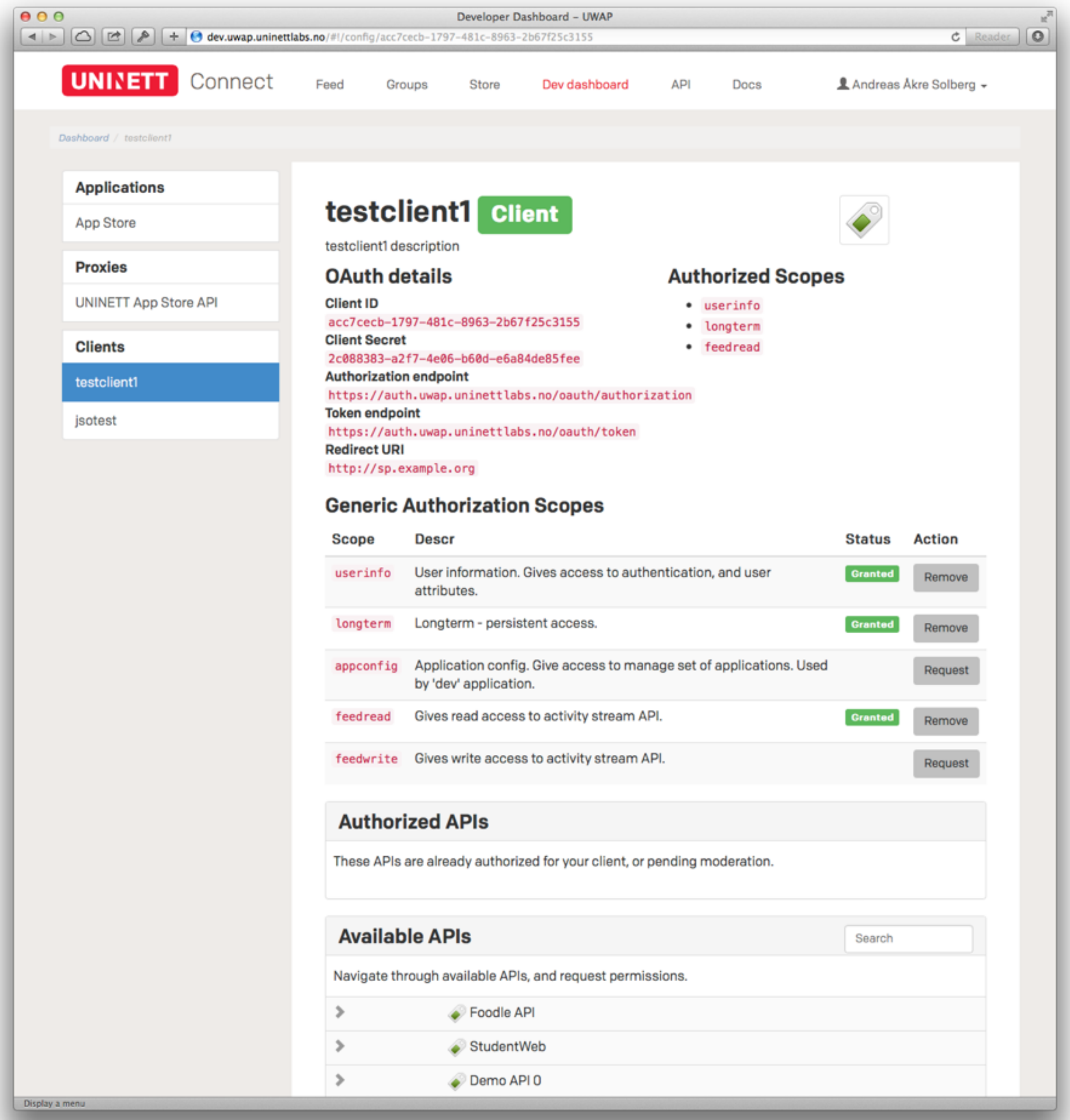
Group types

- ● All groups 21
- ○ Organization 1
- ○ Organization Unit 1
- ○ Ad-Hoc 18
- ○ UWAP Plattform 1
- ○ Course 0
- ○ Cohort 0

Member roles

- ● All roles 21
- ○ Member 4
- ○ Admin 1
- ○ Owner 16

The developer dashboard allows registration of new clients and API Gatekeepers, and for administration of these.



testclient1 Client

testclient1 description

OAuth details

Client ID
`acc7cecb-1797-481c-8963-2b67f25c3155`
Client Secret
`2c088383-a2f7-4e06-b60d-e6a84de85fee`
Authorization endpoint
`https://auth.uwap.uninettlabs.no/oauth/authorization`
Token endpoint
`https://auth.uwap.uninettlabs.no/oauth/token`
Redirect URI
`http://sp.example.org`

Authorized Scopes

- `userinfo`
- `longterm`
- `feedread`

Generic Authorization Scopes

Scope	Descr	Status	Action
`userinfo`	User information. Gives access to authentication, and user attributes.	Granted	Remove
`longterm`	Longterm - persistent access.	Granted	Remove
`appconfig`	Application config. Give access to manage set of applications. Used by 'dev' application.		Request
`feedread`	Gives read access to activity stream API.	Granted	Remove
`feedwrite`	Gives write access to activity stream API.		Request

Authorized APIs

These APIs are already authorized for your client, or pending moderation.

Available APIs Search

Navigate through available APIs, and request permissions.

- Foodle API
- StudentWeb
- Demo API 0

Feide Connect will also contain a web app for inspection of authorized applications, with the option to terminate access.

In addition to this; Feide Connect offers a set of developer tools, and documentation.

Federated Widgets

Federated widgets are web content from one service intended to be re-used as a widget in other services. The concept makes use of HTML iFrames to embed frames, together with a shim javascript to handle communication. The widget it self would need to handle user authentication in a very specific way, to avoid issues with third party cookies, and to avoid user interaction within the iframe. Some of the components that Feide Connect needs to properly support this functionality is passive OAuth and SAML requests.

Supporting infrastructure

Testing
Monitoring
Statistics
Automated deployment