FHNW OpenID Connect pilot

The (SAML) World is not enough …

Agenda

- Business use case
- Technical requirements
- OpenID Connect (OIDC)
- SAML vs. OIDC vs. Oauth
- Implicit Flow
- Architecture
- Implementation
- Demo

Business use case

- Webbased registration for any kind of «events» managed in administration system «Evento», e.g. certificate courses, conferences, information events, sports activities, …
- Required information during registration is very diverse between various event types and target groups (eg. catering options, workshop sign-up, etc.)
- Authentication via AAI (existing FHNW students/staff) and
  Swiss edu-ID must be possible, as well as unauthenticated registration (guest)
- Flexible web application that can be easily integrated in any kind of
  web page, CMS, infrastructure, etc. (e.g. SharePoint-based intranet).

Technical requirements

- Solution based on JavaScript-Webclient and REST-Services
- Strategic technologies AngularJS and ASP.NET (WebApi)
- Lightweight authentication and authorization protocol with good support for untrusted clients and delegated access scenarios
- Generic approach and infrastructure

OpenID Connect (OIDC)

«A Simple Identity layer on top of OAuth 2.0»

From the website (http://openid.net):

- OIDC is an interoperable authentication protocol based on the OAuth 2.0 family of specifications. It uses straightforward REST/JSON message flows with a design goal of «making simple things simple and complicated things possible». It's uniquely easy for developers to integrate, compared to any preceding Identity protocol.
- OIDC allows for clients of all types, including browser-based JavaScript and native mobile apps, to launch sign-in flows and receive verifiable assertions about the identity of signed-in users.

SAML vs OIDC vs OAuth

Simply said:

- SAML: single sign-on for enterprise users
- OAuth: API authorization between applications
- OIDC: single sign-on for consumers + API access

Token format:

- SAML: XML
- OIDC: JWT (JSON web token)

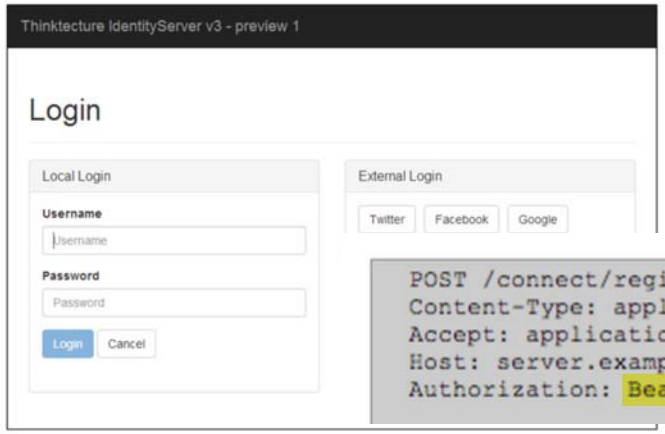
The whole story:

«Unifying Authentication & Delegated API Access for Mobile, Web and the Desktop with OpenID Connect and OAuth2 by Dominick Baier», https://vimeo.com/113604459

Implicit Flow

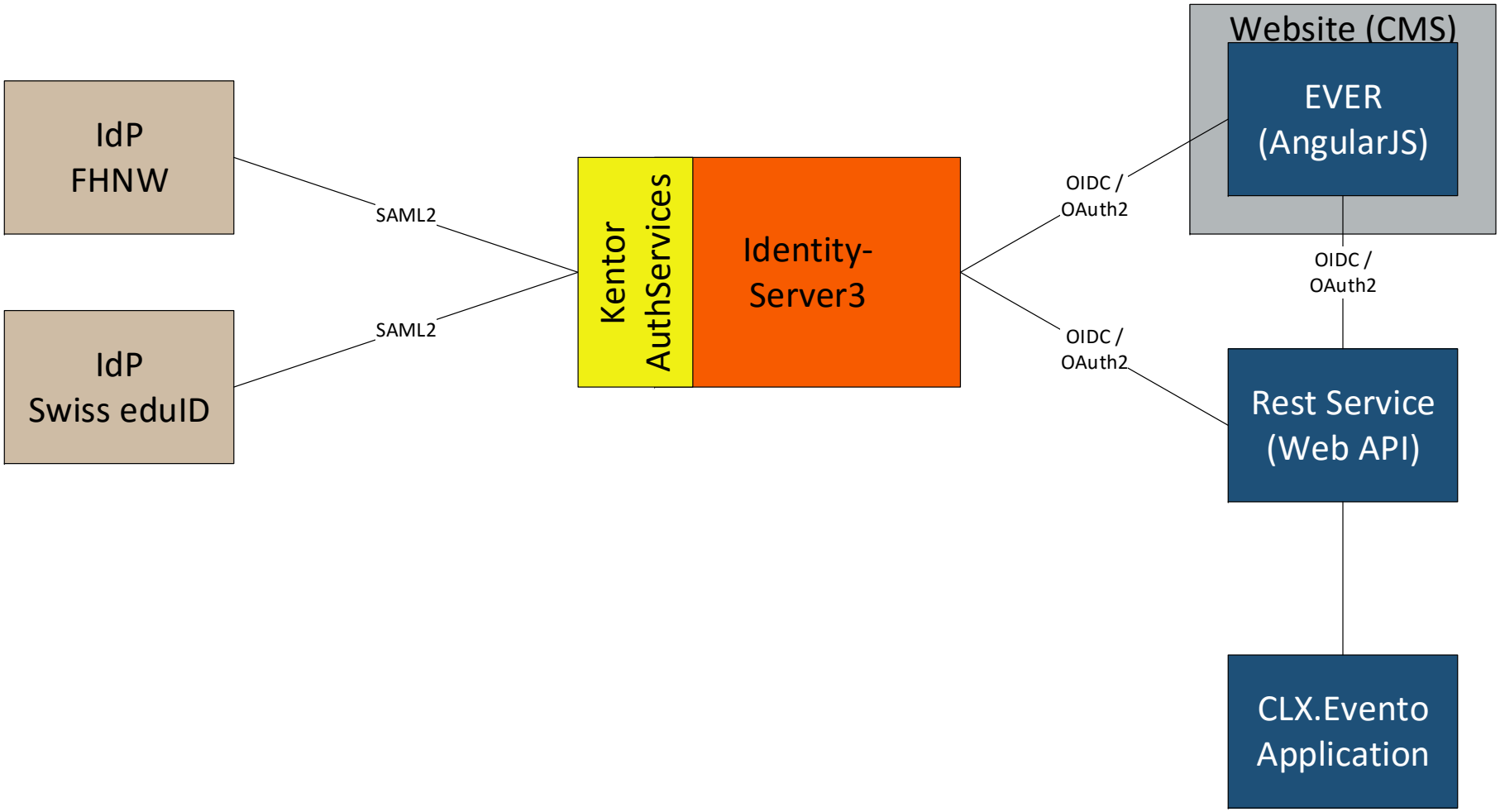
- One of multiple flows (processes) defined in the specification
- Specifically designed for «untrusted clients» such as JavaScript apps
- Key steps:
  - Client sends the request to the Authorization Server.
  - Authorization Server authenticates the End-User, obtains consent, sends him/her back to the Client with an ID Token and, if requested, an Access (Bearer) Token.
  - Client validates the tokens and retrieves the End-User's Subject Identifier.
  - Client uses the Access Token for calls to Backend Web Services

```
GET /authorize
?client_id=app1
&scope=openid profile
&redirect_uri=https://app.com/cb
&response_type=id_token token
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
```




```
POST /connect/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJ ...
```

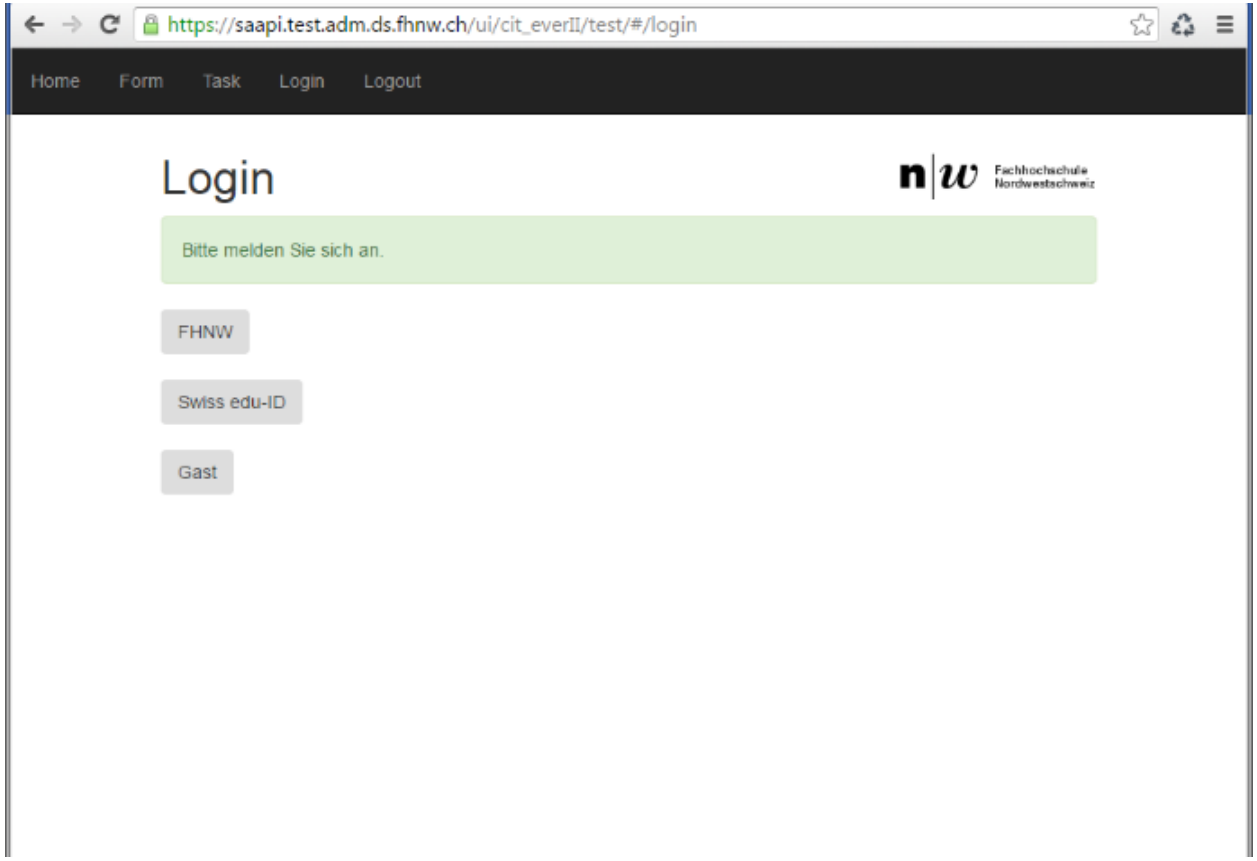
Architecture



Implementation

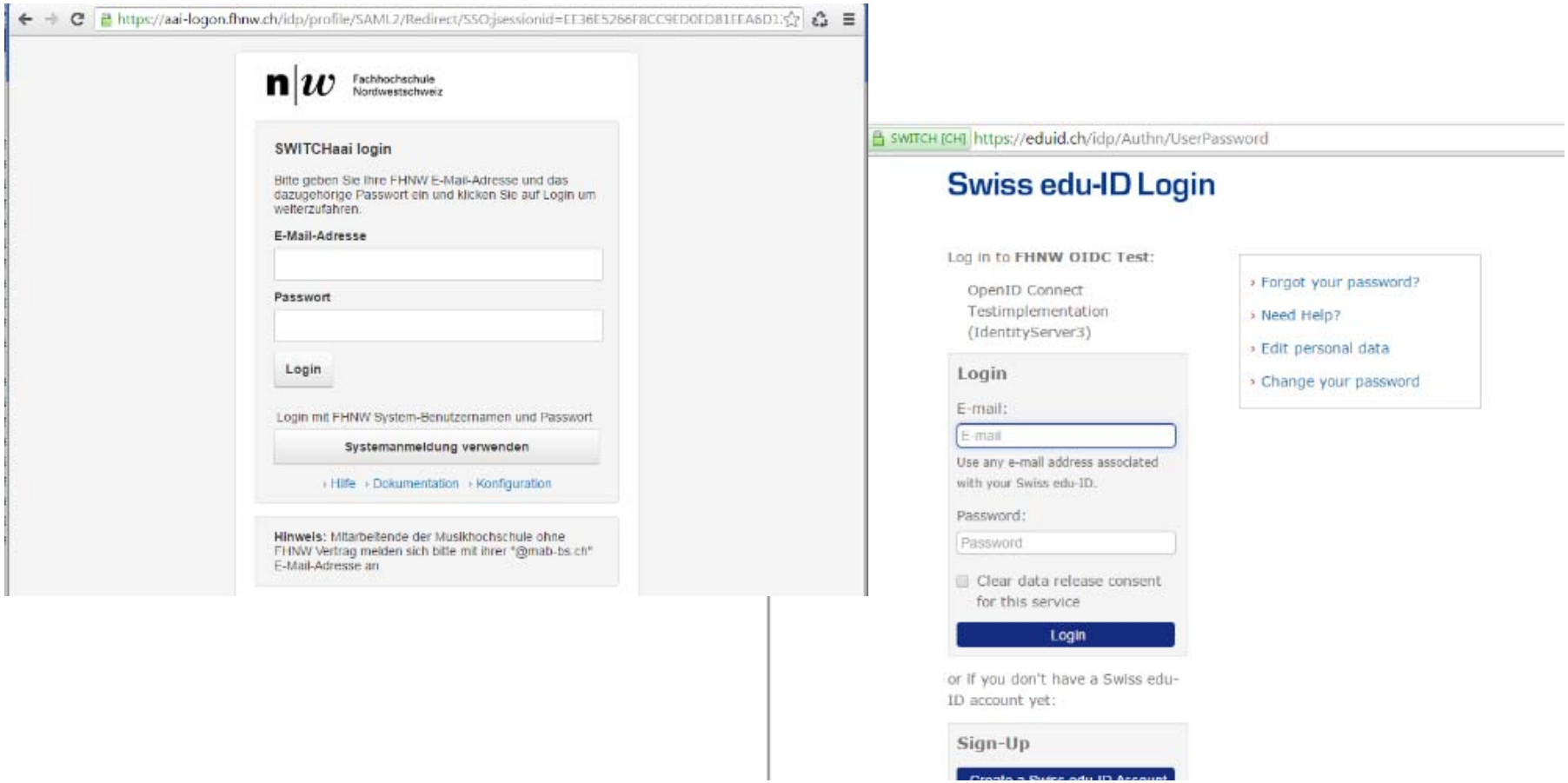
- **IdentityServer3**
  - OpenID Connect Provider and OAuth 2.0 Authorization Server Framework
  - Certified OpenID Connect implementation
  - Open Source
  - Based on ASP.NET web framework
  - https://github.com/IdentityServer/IdentityServer3
- **Kentor Authentication Services**
  - SAML2 Authentication services for ASP.NET
  - Open Source
  - https://github.com/KentorIT/authservices
- some extension code developed in-house

Demo (1/5)



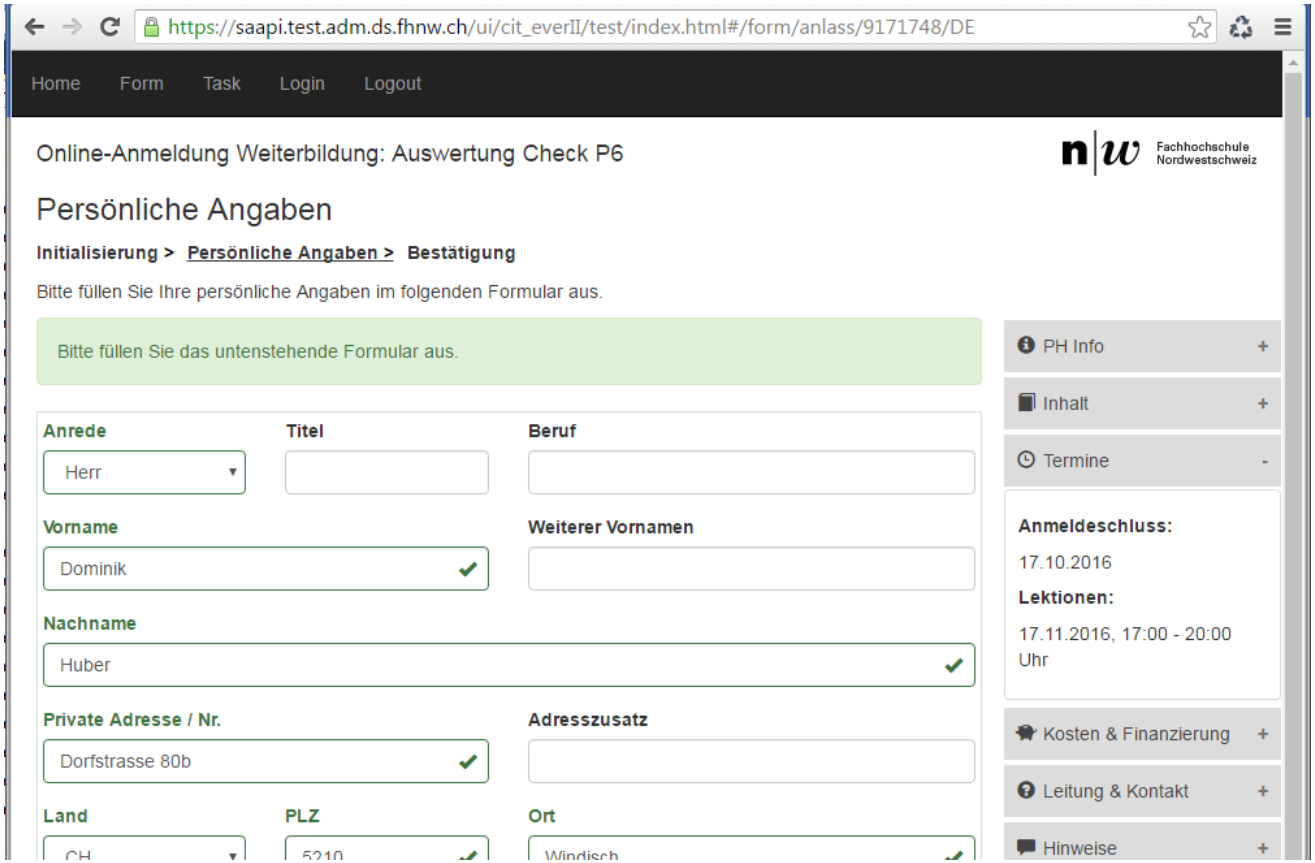
Discovery (to be integrated in application logic later)

Demo (2/5)



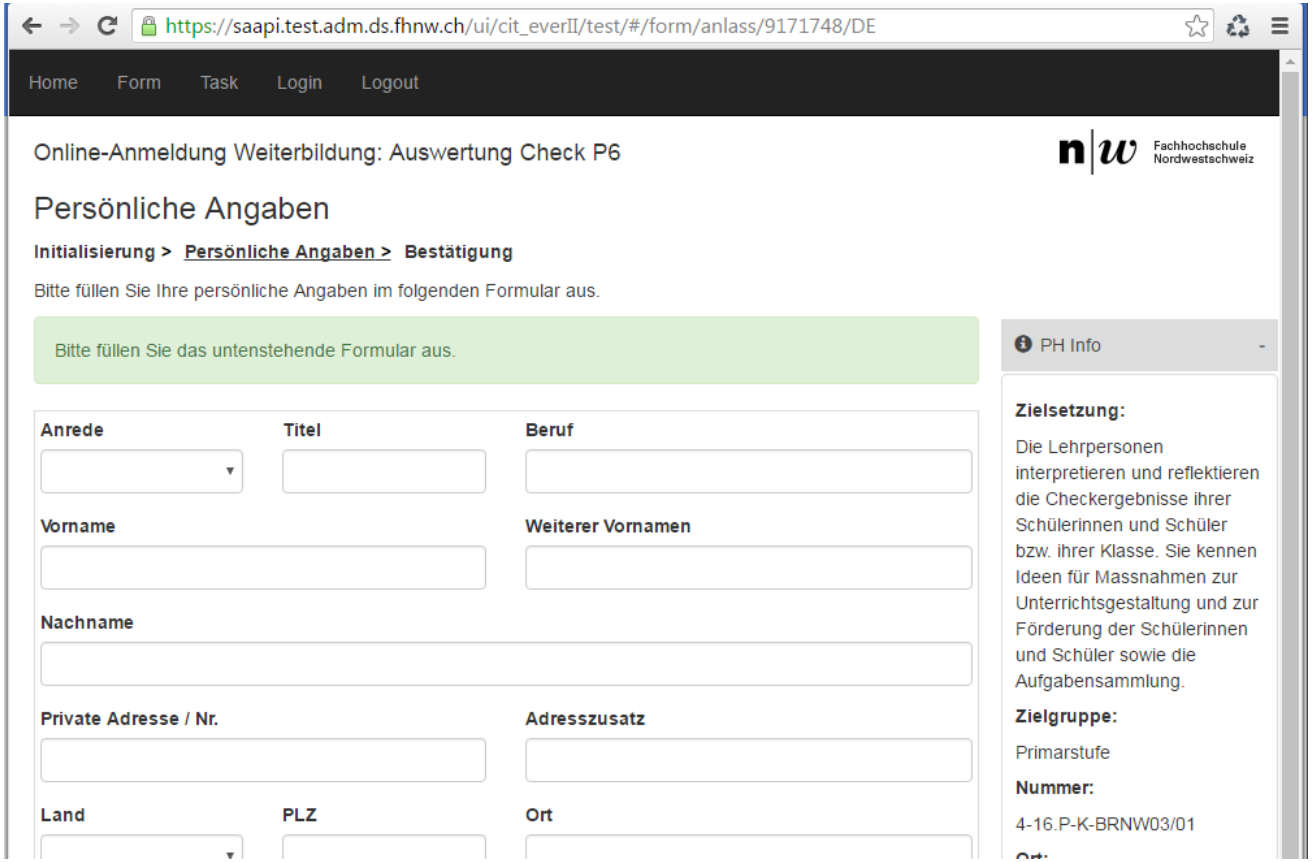
Login (AAI FHNW or Swiss edu-ID)

Demo (3/5)



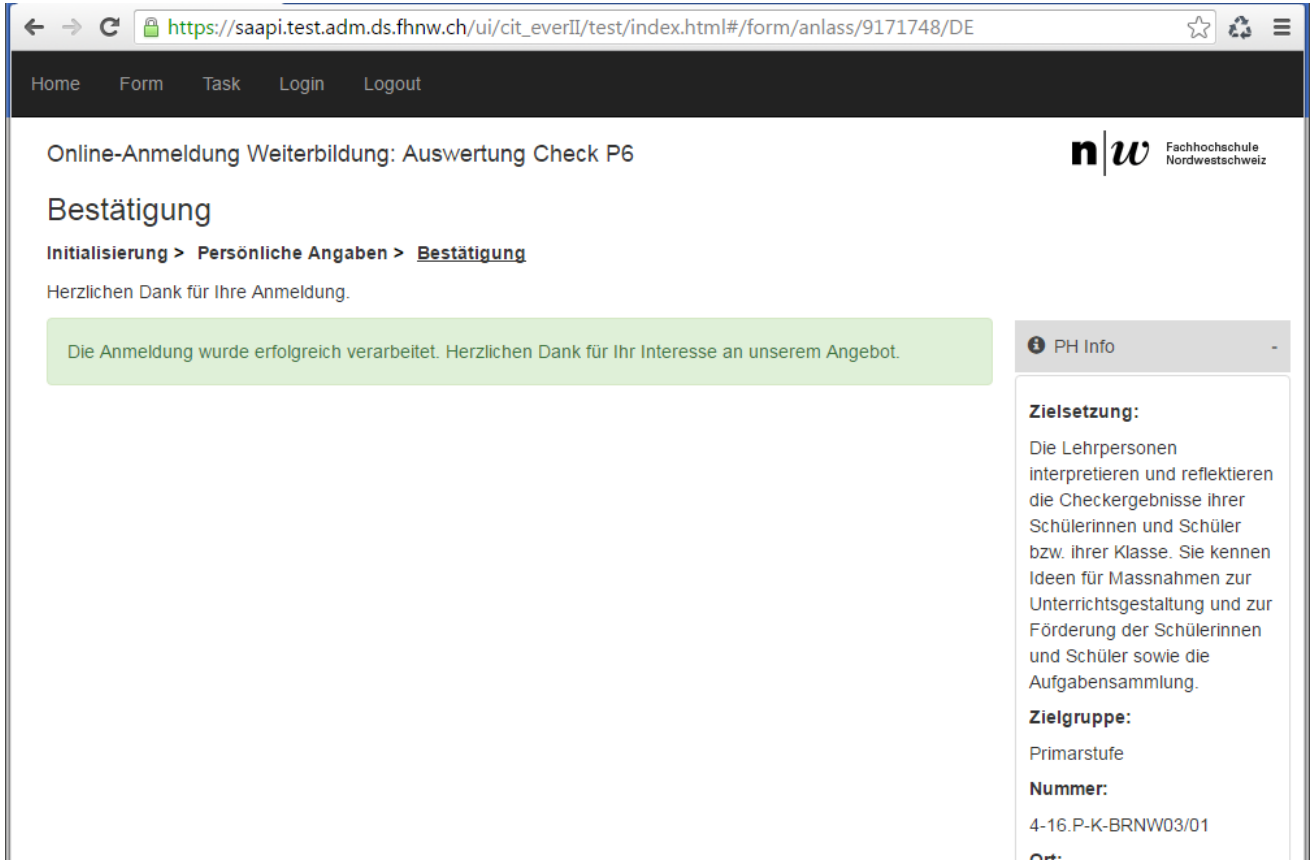
Personal data (authenticated)

Demo (4/5)



Personal data (unauthenticated)

Demo (5/5)



Confirmation

Questions?

Contact:

– Michael Hausherr, Enterprise Architect
T: +41 56 202 71 56, E: michael.hausherr@fhnw.ch
– Dominik Huber, Application Developer
T: +41 56 202 70 27, E: dominik.huber@fhnw.ch