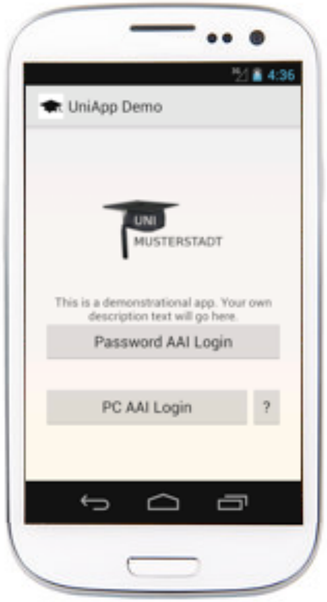
AAI for Mobile Apps

How mobile Apps can use SAML Authentication and Attributes



This is a demonstrational app. Your own description text will go here.

SWITCH

Serving Swiss Universities

Lukas Hämmerle
lukas.haemmerle@switch.ch

Berne, 13. August 2014

Introduction

App by University of St. Gallen

- Universities offer apps, e.g. for e-learning and campus info
- Apps need authentication
- Apps usually are non-browser applications
- Authentication and Authorisation Infrastructure (AAI) based on SAML2 are difficult to use for non-browser applications

Prerequisites for a Solution

- App users from **many AAI organisations**
  - Excludes authentication with LDAP or HTTP Basic Auth

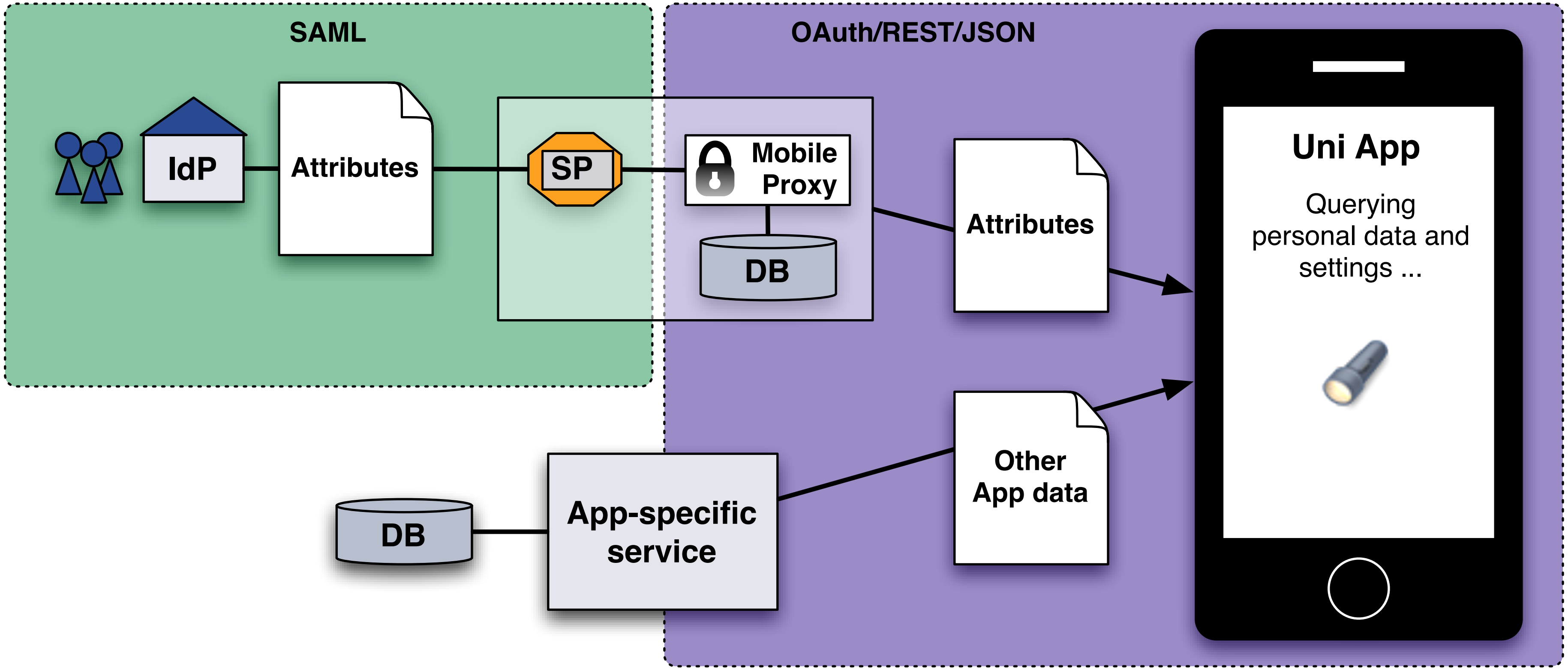
- **No changes**/updates/plugins for Identity Provider needed
  - Excludes SAML Enhanced Client and Proxy (ECP) profile

Solution wanted that works <u>today</u> in AAI!

App Requirements

- App should **not “emulate” a web-browser** for authentication
  - Excludes already known approaches
- App should **not save user’s university password**
  - Would cause problems (app data stolen by other app, commercial company offering app, password change)
- App should **not ask user to authenticate too often**
  - Apps should be easy to use and behave like other apps
- App should **always get up-to-date user attributes** on start
  - Excludes approaches based on caching user attributes

Solution



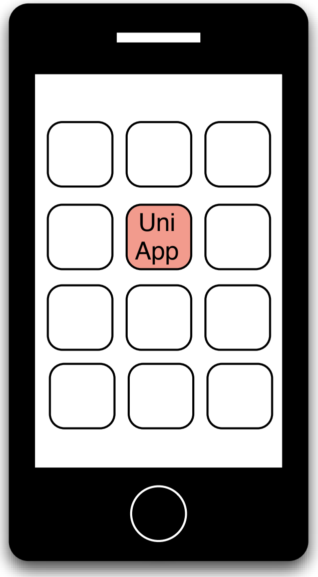
- A **(Mobile) Proxy** translates authentication/attribute information from SAML2 to OAuth/REST/JSON
- Mobile Proxy includes an OAuth2 Server that grants access tokens, which are mapped to a SAML2 persistent ID

Concept of Mobile Proxy

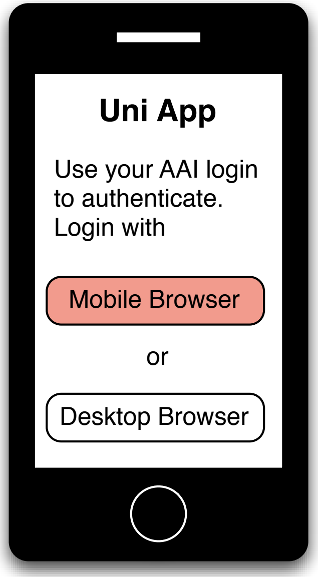
1. User authenticates once at Mobile proxy via web browser
2. Mobile Proxy gets persistent ID of user
3. Proxy stores persistent ID and binds it to an OAuth2 access token, which is stored in the App
4. App queries Mobile proxy for AAI attributes with token
5. Mobile Proxy uses persistentId to query user’s AAI attributes via a SAML Attribute Query

User’s Perspective: First App Start

- User starts app for the first time

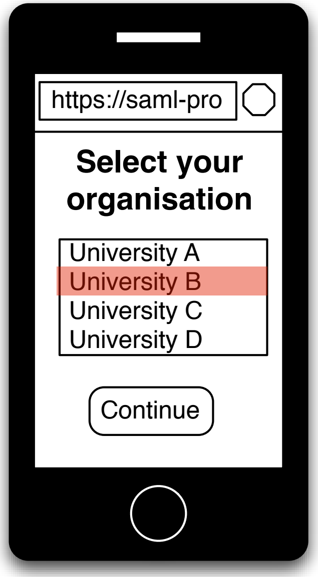


- App asks user to authenticate with AAI on device or desktop PC



Use your AAI login to authenticate. Login with

- Mobile browser opens and user selects his organisation

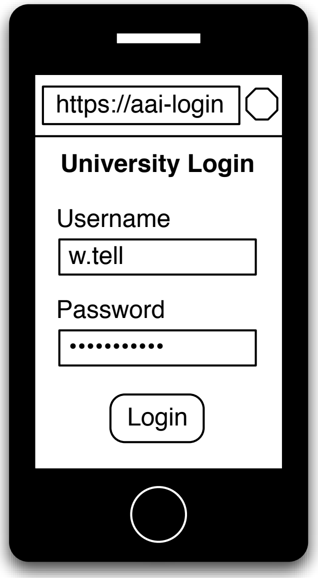


User’s Perspective: First App Start Continued

- Authentication with AAI at home organisation in web browser

- Mobile Proxy SP gets user’s attributes including persistentId and issues OAuth token

- Uni App uses token to get user attributes from Mobile Proxy



https://aai-login

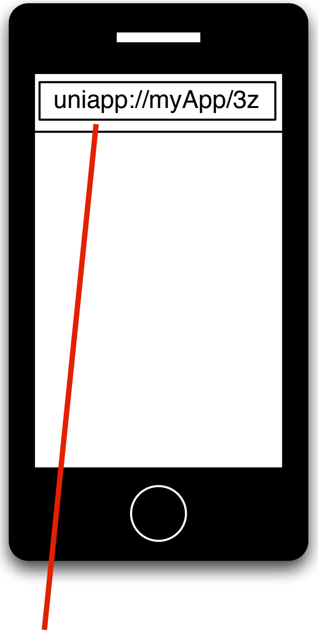
University Login

Username

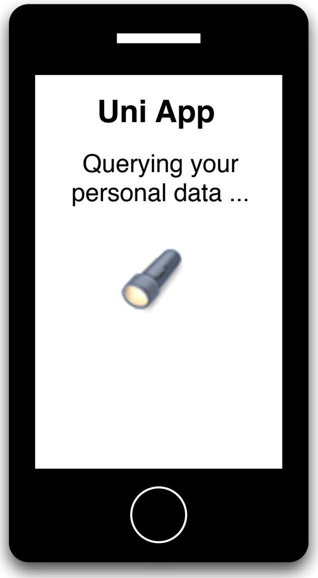
w.tell

Password

•••••••••••



uniapp://myApp/3z



Uni App

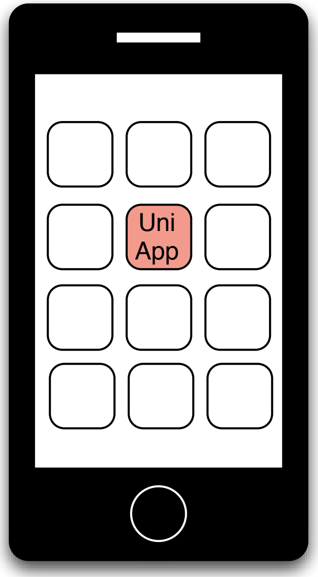
Querying your personal data ...

Link with custom URL scheme is opened automatically

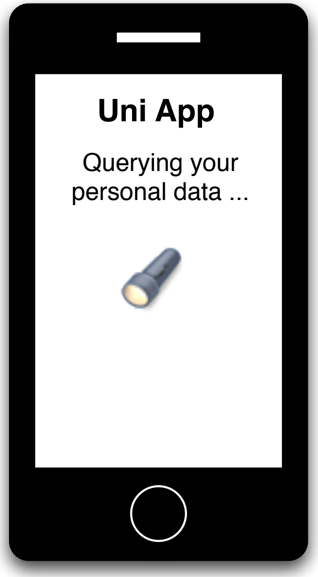
E.g. uniapp://{App-Identifier}/{40-Byte-Access Token}

User's Perspective: Further App Starts

- User starts app



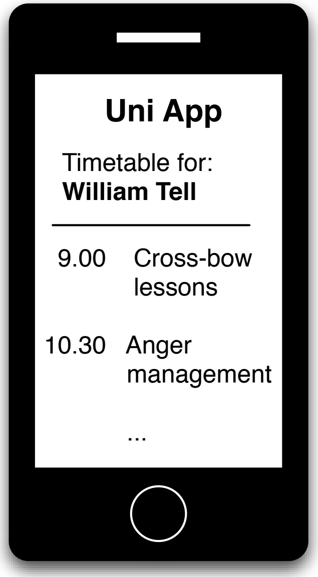
- App fetches user attributes with OAuth access token from proxy



Uni App

Querying your personal data ...

- App gets other app-specific data with access token



Uni App

Timetable for:
William Tell

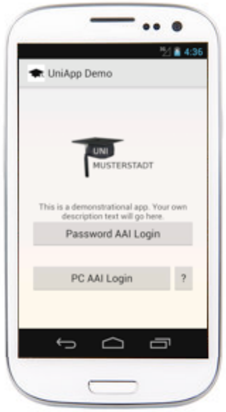
9.00 Cross-bow lessons

10.30 Anger management

...

Demo of Sample Uni App

- A quick demo is available on the AAI for Apps web page: https://www.switch.ch/aai/support/tools/aai-for-apps.html



This is a demonstrational app. Your own description text will go here.

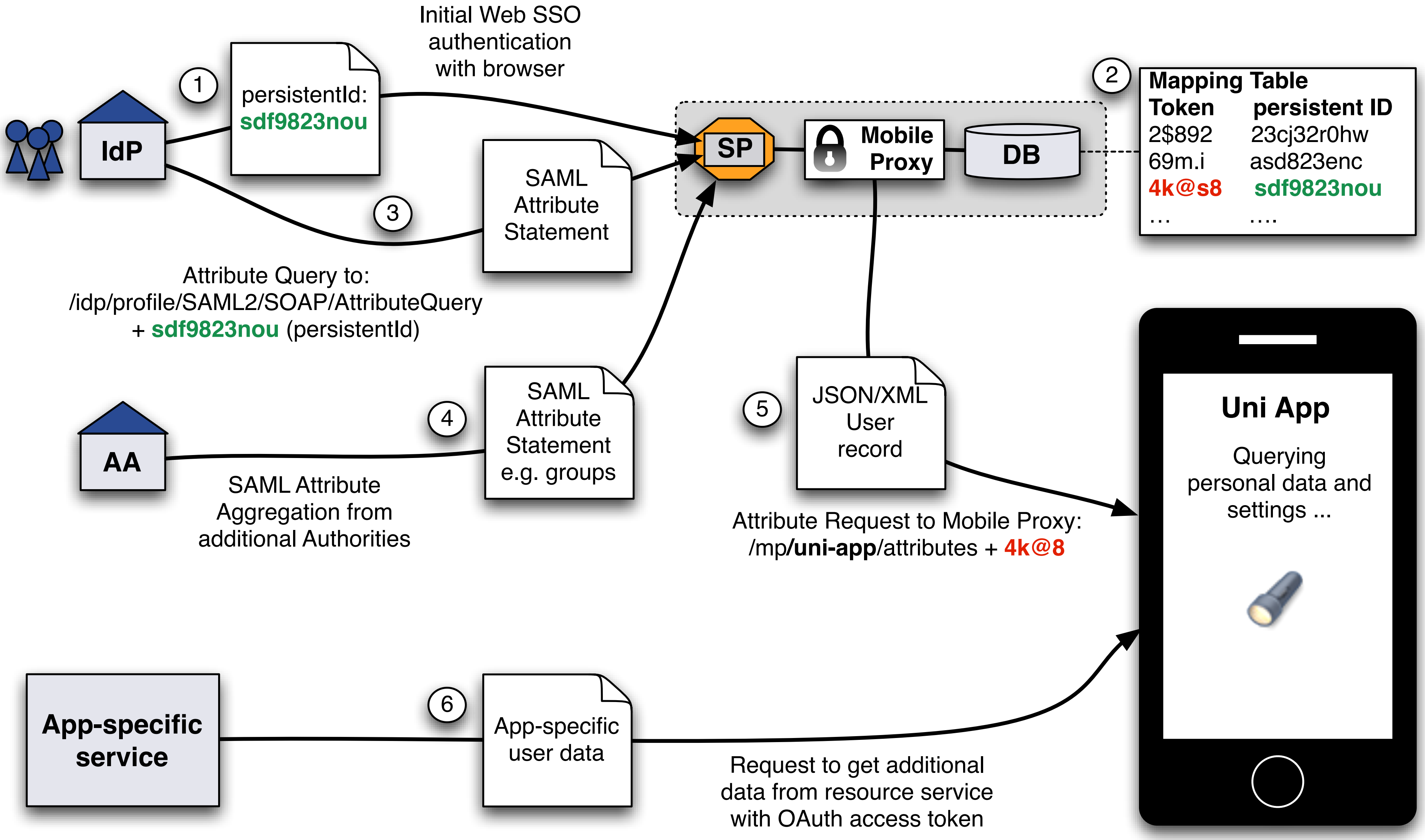
- Two options for initial AAI login:
  - Browser on mobile device
  - Browser on another computer (requires typing or scanning QR code)

Mobile Browser vs Desktop Browser

To get persistent ID, User must login with a web browser at least once with AAI. But with which browser?

- **In-App browser:**
  - In app browser might not have access to browser saved passwords user has to type in again username password at IdP
- **Browser on mobile device:**
  - Benefit from SSO session that user might have already
  - Default browser on device is used
- **Browser on Desktop:**
  - Most flexible browser that might support authentication methods other than username/password. E.g. X.509
  - Requires user to type URL/token or scan a QR code

Data Flow



Mapping Table

Token	persistent ID
2$892	23cj32r0hw
69m.i	asd823enc
4k@s8	sdf9823nou
…	….

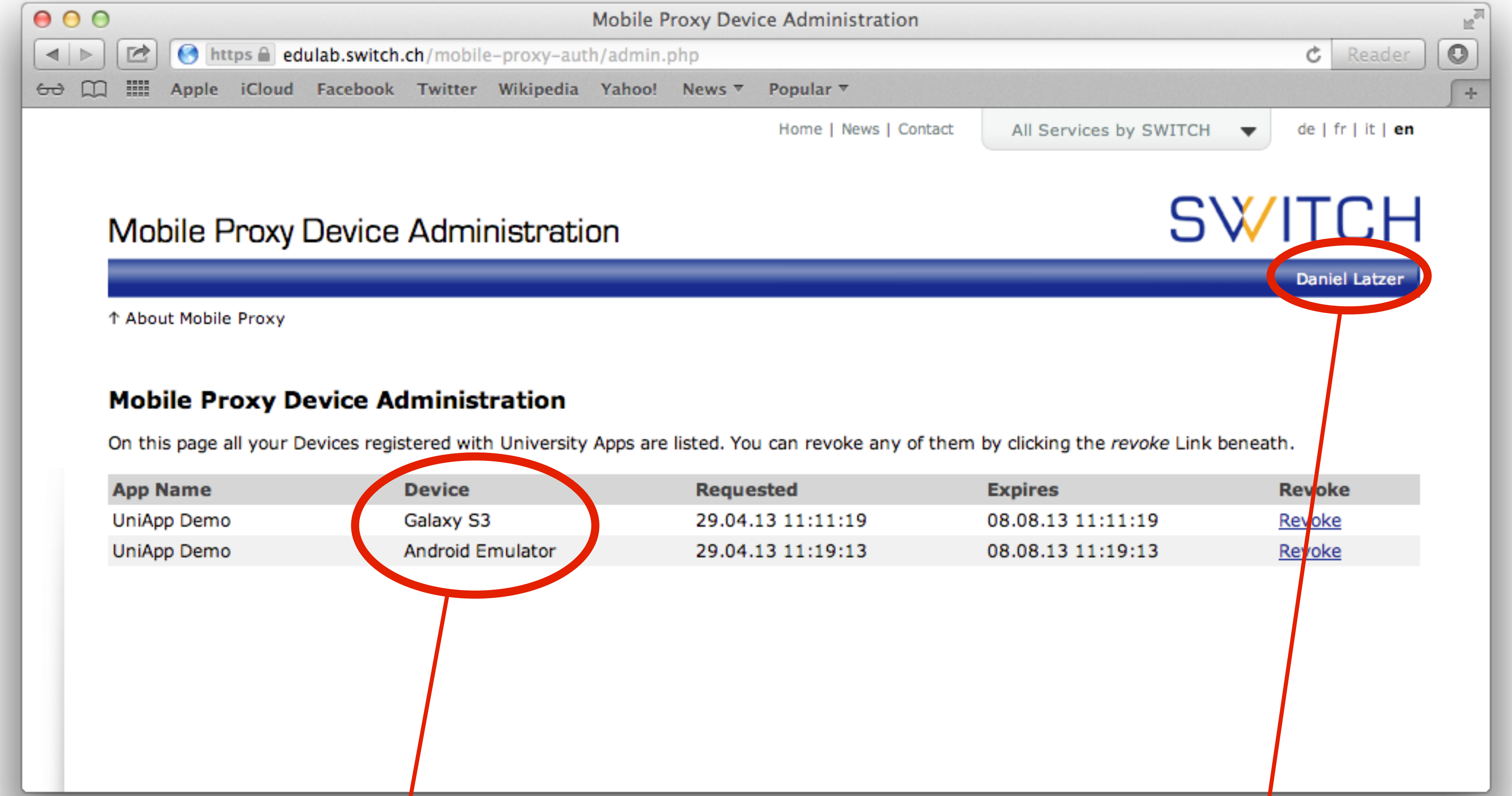
App Logout / Access Token Revocation

How about revocation of OAuth access token?
For example in case the device is sold or lost.

- OAuth Access token is used to:
  - Authenticate with Mobile Proxy
  - Retrieve up-to-date AAI attributes from Mobile Proxy
  - Retrieve arbitrary protected resources from third party resource server

- Token can be revoked by:
  - Expiration because validity is configurable
  - User within App by clicking on “Logout”
  - User via administration interface with web browser

Logout/Token Revocation via Web Interface



Mobile Proxy Device Administration
https edulab.switch.ch/mobile-proxy-auth/admin.php
Reader
Apple iCloud Facebook Twitter Wikipedia Yahoo! News Popular
Home | News | Contact
All Services by SWITCH
de | fr | it | en
SWITCH
Daniel Latzer
↑ About Mobile Proxy

Mobile Proxy Device Administration

Mobile Proxy Device Administration

On this page all your Devices registered with University Apps are listed. You can revoke any of them by clicking the *revoke* Link beneath.

App Name	Device	Requested	Expires	Revoke
UniApp Demo	Galaxy S3	29.04.13 11:11:19	08.08.13 11:11:19	Revoke
UniApp Demo	Android Emulator	29.04.13 11:19:13	08.08.13 11:19:13	Revoke

Multiple devices for same user and same app

Authenticated user

Advantages of this Approach

- App never gets user’s AAI credentials
  - Any type of authentication can be used
- Can be deployed immediately without changes to federation
  - Requires that IdPs support persistentId (with storedId) and attribute queries. This is the case for all SWITCHaai IdPs.
  - Approach also works when SP aggregates attributes from additional attribute authorities (Virtual Organization/Group attribute providers)
- One instance of Mobile Proxy can serve multiple apps
  - Apps can have different attribute requirements
  - Individual <EntityDescriptors> for each app possible

Availability and Future Plans

- Software available as Open Source software (BSD license)
  - **Sample Uni App**: Java, Android App ready for customization
  - **Mobile Proxy**: PHP, Includes OAuth server and simple web interface
  - **Resource Server**: PHP, Returns back a default time table
- Developed as Prototype. No production quality yet.
- More information and link to SVN repository: http://swit.ch/aai-for-apps
- SWITCH is considering to turn Mobile Proxy into a service if community is interested and contacts us!