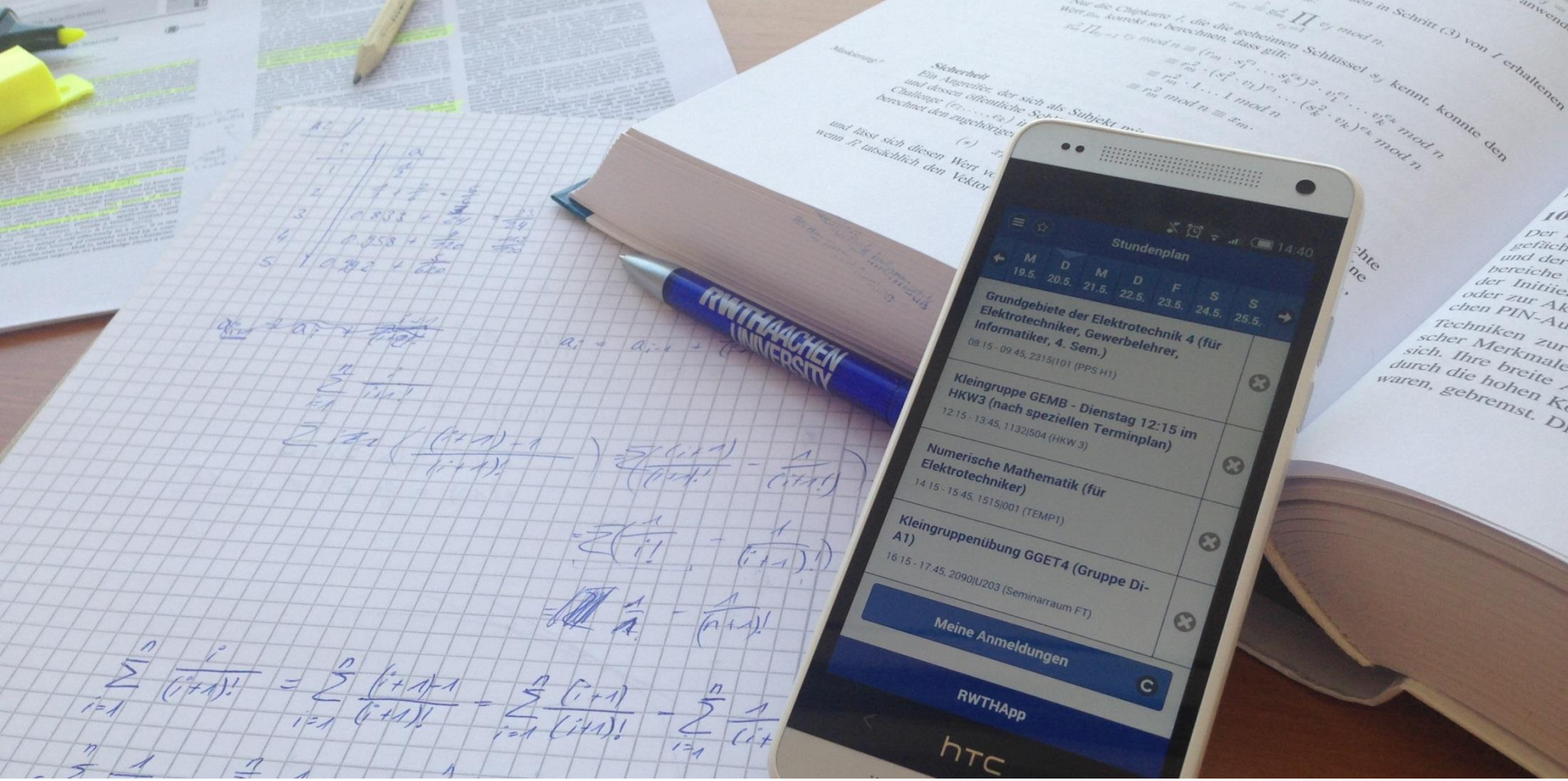


Authentifizierungs- und Autorisierungsschnittstelle für (mobile) Geräte als Komponente des RWTH-IdMs

ZKI AK Verzeichnisdienste, 02.10.2015
B. Decker, M. Politze

IT Center | RWTH AACHEN UNIVERSITY

Agenda

- OAuth2 zur Autorisierung an der RWTH Aachen
  - Einleitung
  - Umsetzung
- Authentifizierung für OAuth2 über Shibboleth
  - Aktueller Stand
  - Datenfluss
- Einsatzszenarien
  - Schnittstellen an der RWTH Aachen
  - Absicherung von Eduroam mit OAuth2
  - OAuth2 im DFN AAI
- Fazit
  - Ausblick

Einleitung

Ausgangslage

- RWTHApp soll entwickelt werden (Schnittstellen zu CAMPUS, SOS/POS, LMS, UB, …)
- Entwicklung von Apps über „Screenscraping“ und mit Weitergabe von RWTH-Credentials (zum Teil über HTTP)
- Entwicklung einer API für das E-Learning Portal L²P, für Studierende, Seminare etc.

Problem

- Wildwuchs, ohne Regelungen bzgl. Sicherheit, Datenschutz
- Verbieten quasi unmöglich

Maßnahmen

- Zentrales Angebot für Entwickler schaffen
- Komfortabel zu benutzen (einfach, stabil)

Anforderungen

- Keine Weitergabe von Benutzernamen und Passwort an die App
  - Credentials werden bei Verlust des Geräts nicht kompromittiert
- Apps explizit für bestimmte Anwendungen autorisieren
- (De-)Autorisierung einer App ohne Auswirkungen auf andere Apps
- Nur bekannte Apps erhalten Zugriff auf Quellsysteme
- Datenintegrität sicherstellen

Lösung: OAuth2

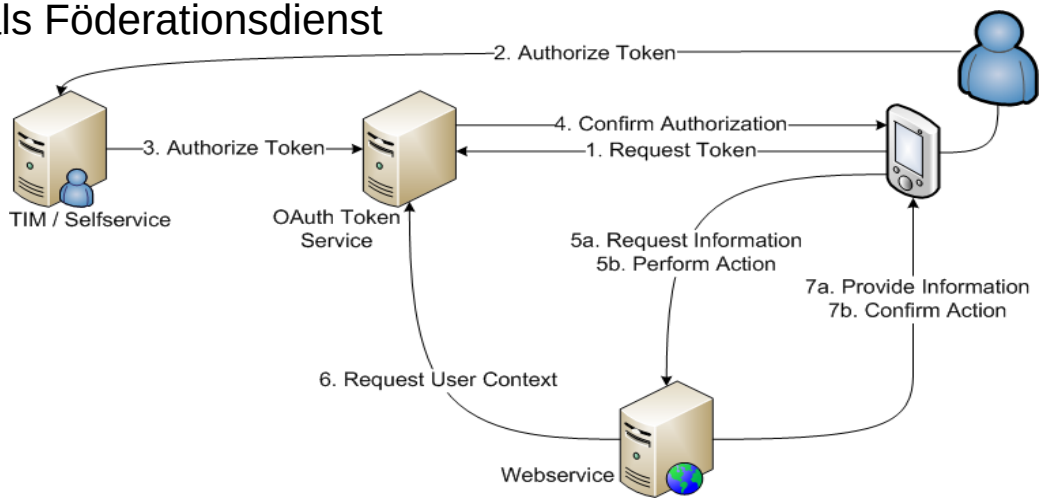
Umsetzung

Sicherheit und Datenschutz als Designgrundlage

- Datensparsamkeit
  - Übertragung und Speicherung von benötigten Daten
- Nutzung von ausreichend langen Tokens
  - Verminderung von Auswirkungen durch Brute Force Angriffe
  - Weitere Schutzmechanismen (z.B. Flood Prevention) nötig
- Nutzung guter Zufallsgeneratoren
- Gesicherte Kommunikation
  - Webservices ausschließlich per HTTPS
  - Signatur der Kommunikation zwischen den Servern per JWT
- Verschlüsselte Speicherung der Autorisierungen
  - Lesen von Autorisierungen nur mit dem Token

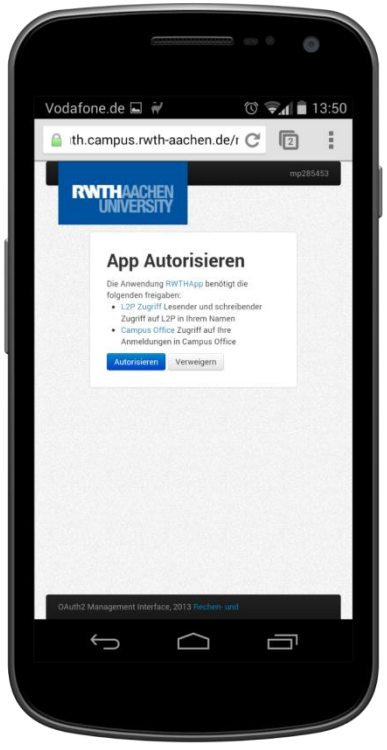
OAuth an der RWTH Aachen

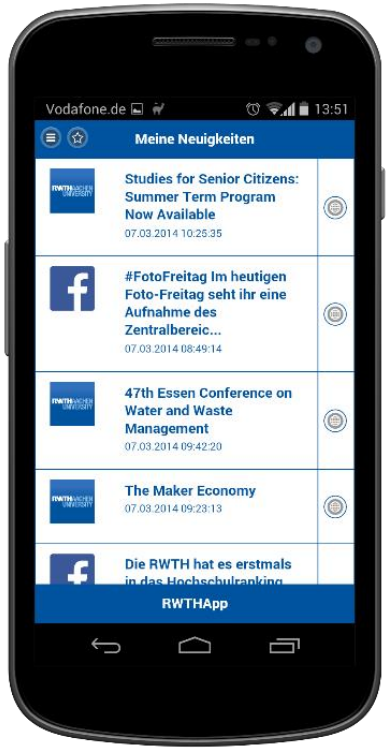
- Sichere, gerätebasierte Autorisierung
  - (De)Autorisierung über Webinterface
  - Keine Weitergabe von Credentials
- OAuth2 als Dienst
  - Integriert mit Shibboleth zur Authentifizierung
  - Möglich auch als Föderationsdienst
- An der RWTH etabliert
  - z.B. die RWTHApp mit ~20.000 Nutzern
  - Verfahren skaliert auf unterschiedliche Anwendungen



Funktionsweise aus Sicht des Nutzers

Prozess starten (RWTHApp) → Autentifizierung (Shibboleth) → App Autorisieren (OAuth) → Personalisierte Informationen Anzeigen (RWTHApp)

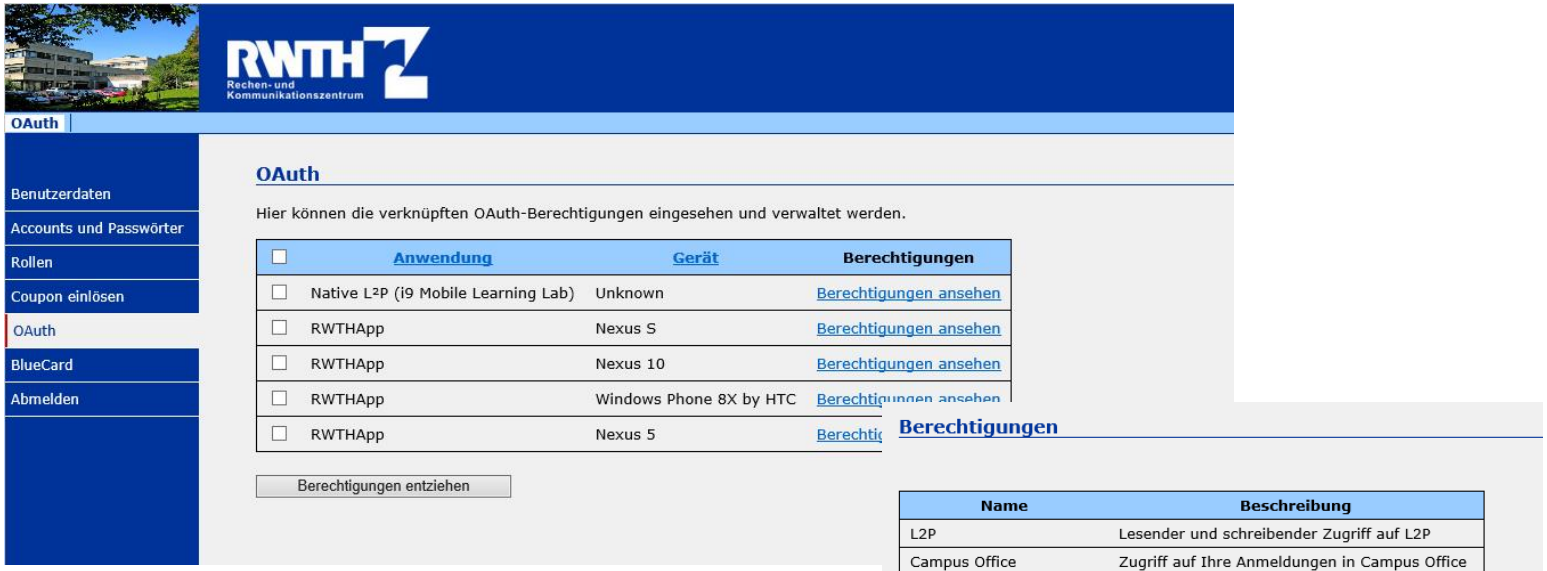




Umsetzung

Management von Autorisierungen

- Integriert in den SelfService des Identity-Managements
  - Übersicht der autorisierten Geräte und Berechtigungen
  - Deautorisierung über Browser



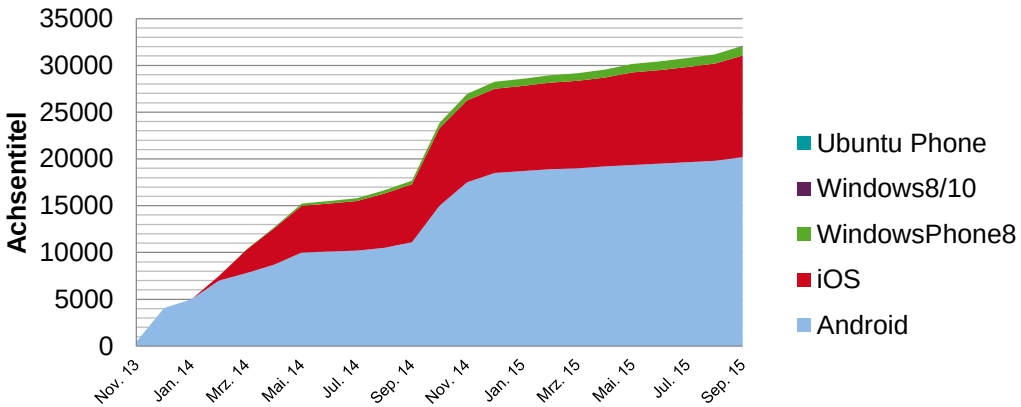
Aktueller Stand

Fakten

OAuth Schnittstelle
- > 50.000 (aktive) Autorisierungen (personalisiert)
- ~150.000 Requests pro Tag, davon ca. 15.000 anonym

Angeschlossene Anwendungen
- RWTHApp
- SyncMyL²P
- Moodle
- Vispa
- StudyCrowd
- Eduroam
- InfoDisplays
- Studierendenapps (>25)
- SharePoint
- …

Probleme
- Viele zufriedene Nutzer ;-)



Oauth Autorisierungen im Kontext der RWTHApp

Funktionsweise aus Sicht der App (Autorisierung)

```
POST /oauth2waitress/oauth2.svc/code HTTP/1.1
Host: oauth.campus.rwth-aachen.de
Content-Type: application/x-www-form-urlencoded

client_id=QhV1IXlzticl9JCKgH01bhOMlu.app.rwth-aachen.de&
scope=l2p.rwth campus.rwth
```

```
{
    "device_code" : "BaUAJHPFYFi6wKU0WY5xLC",
    "user_code" : "SFW7WZXK7G",
    "verification_url" : "https://oauth.campus.rwth-aachen.de/manage",
    "expires_in" : 1800,
    "interval" : 5
}
```

```
https://oauth.campus.rwth-
aachen.de/manage?authorize=SFW7WZXK7G
```

```
GET manage?authorize=SFW7WZXK7G
Host: oauth.campus.rwth-aachen.de
```

```
POST /oauth2waitress/oauth2.svc/token HTTP/1.1
Host: oauth.campus.rwth-aachen.de
Content-Type: application/x-www-form-urlencoded

client_id=QhV1IXlzticl9JCKgH01bhOMlu.app.rwth-aachen.de&
code=BaUAJHPFYFi6wKU0WY5xLC&
grant_type=device
```

```
{
    "access_token" : "1bAiOVYtFmxSaOsSlwTh9o0ZUFK4AWS2FWQgmVhw3t1Y",
    "token_type" : "Bearer",
    "expires_in" : 3600,
    "refresh_token" : "s4KJhlt9ON8jPJVgz3npdShhjDq5Ucu3coIZv5nkQajFc"
}
```

Funktionsweise aus Sicht der App (Datenabfrage)

```
GET /proxy/api.svc/GetNewsFeed?
accessToken=1bAiOVYtFmxSaOsSlwTh9o0ZUFK4AWS2FWQgmVhw3t1Y HTTP/1.1
Host: moped.ecampus.rwth-aachen.de
```

```
{
  NewsFeed: [
    {Title: "Studies for Senior ...", Date: "2014-03-07T15:35Z"},
    {Title: "#FotoFreitag Im heu...", Date: "2014-03-07T17:31Z"},
    ...
  ]
}
```

Funktionsweise aus Sicht des Informationsdienstes

```
GET /proxy/api.svc/GetNewsFeed?
accessToken=1bAiOVYtFmxSaOsSlwTh9o0ZUFK4AWS2FWQgmVhw3t1Y HTTP/1.1
Host: moped.ecampus.rwth-aachen.de
```

```
GET /oauth2waitress/oauth2.svc/token?
accessToken=1bAiOVYtFmxSaOsSlwTh9o0ZUFK4AWS2FWQgmVhw3t1Y&
serviceId=asder34daf3hbdh34jsk5l.svc.rwth-aachen.de HTTP/1.1
Host: oauth.campus.rwth-aachen.de
```

OAuth Token Service

```
{
  uid: "abc123456"
}
```

```
{
  NewsFeed: [
    {Title: "Studies for Senior ...", Date: "2014-03-07T15:35Z"},
    {Title: "Studies for Senior ...", Date: "2014-03-07T15:35Z"},
    ...
  ]
}
```

On Top of OAuth2

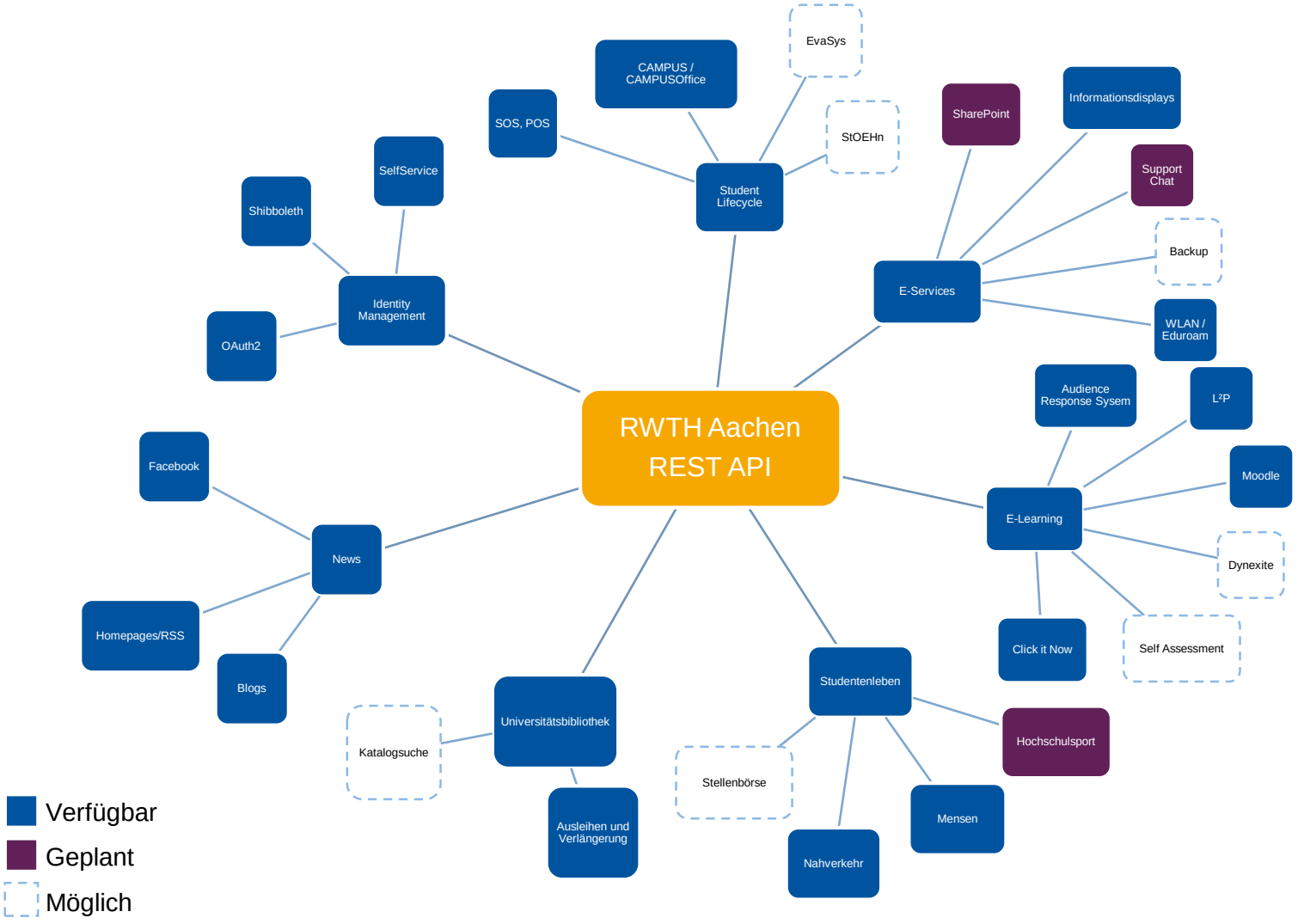
Erweiterung der Einsatzszenarien durch…

- Anonymen Zugriff
  - Identifikation der Anwendung aber nicht des Nutzers

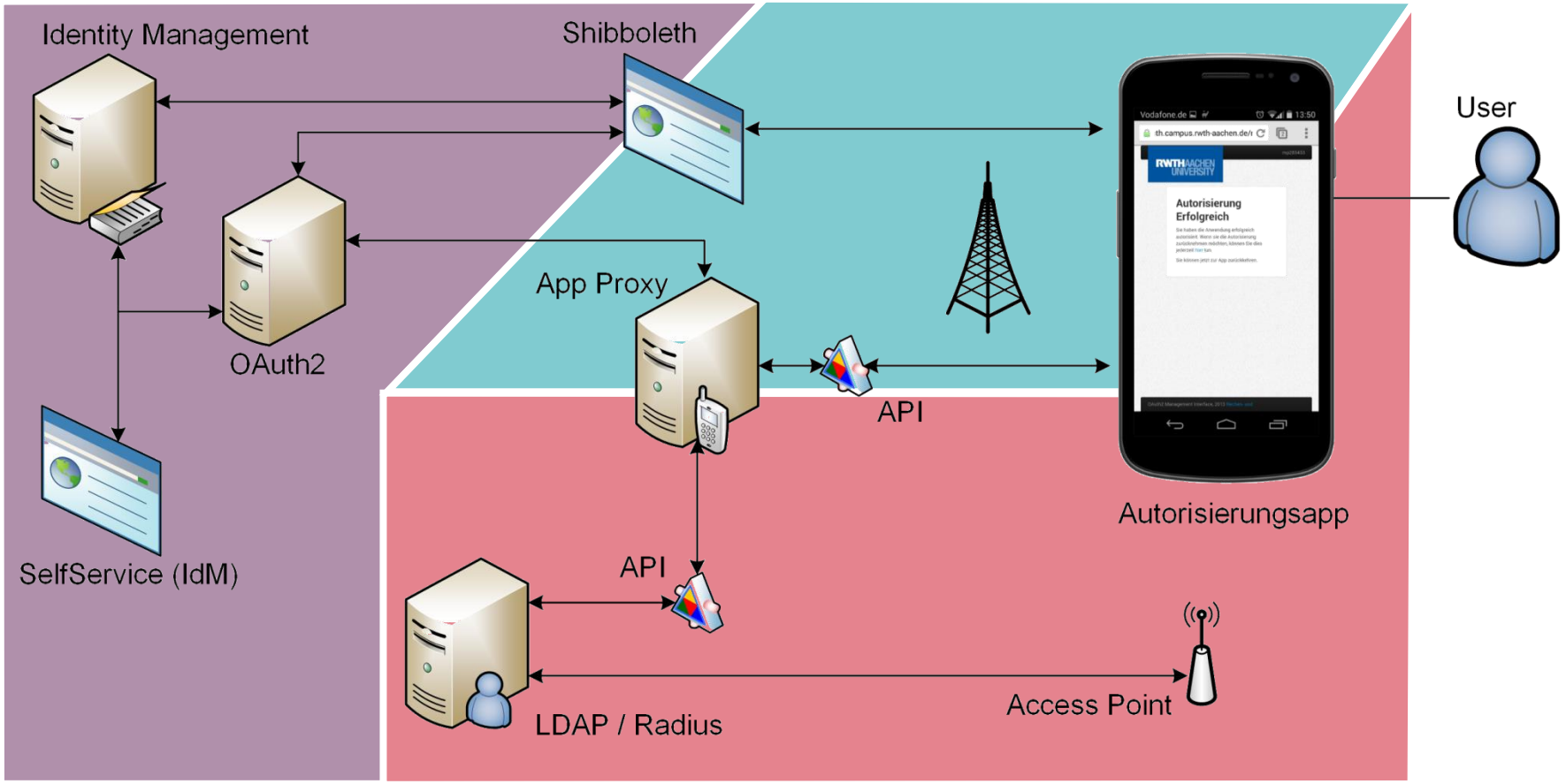
- Autorisierung von Apps und Webanwendungen
  - Vertauenslevel für unterschiedliche Anwendungen und Anwendungstypen
  - Transparenz sowohl für den Nutzer als auch für den Service

- Claim-Basierte Autorisierung
  - Für „Full Trust“ B2B Anwendungen
  - Selbstautorisierung für eigene Webservices
  - Mehrere Authentifizierungsmechanismen

OAuth2 gesicherte Services an der RWTH Aachen



Gerätebasierte Autorisierung für Eduroam



Ziele Gerätebasierte Autorisierung für Eduroam

Reduzieren der Auswirkungen von Evil Twin Attacken [1]

- Geräten kann einzeln der Zugriff auf Eduroam entzogen werden
  - z.B. nach Verkauf oder Verlust des Geräts
  - Regelmäßig nach Inaktivität oder in festen Intervallen
- Kennungen werden automatisiert angelegt
  - Für das Anlegen der Kennungen ist initial eine Internetverbindung nötig
  - Die App übernimmt die Konfiguration der WLAN-Verbindung
- Das Password wird zufällig generiert
  - Kein Auslesen von Passworten für andere Services
  - Passworte können über die App automatisch neu generiert werden

[1] S.Brenza et.al. (2015): A Practical Investigation of Identity Theft Vulnerabilities in Eduroam
http://syssec.rub.de/media/infsec/veroeffentlichungen/2015/05/07/eduroam_WiSec2015.pdf

OAuth2 Lokales Szenario

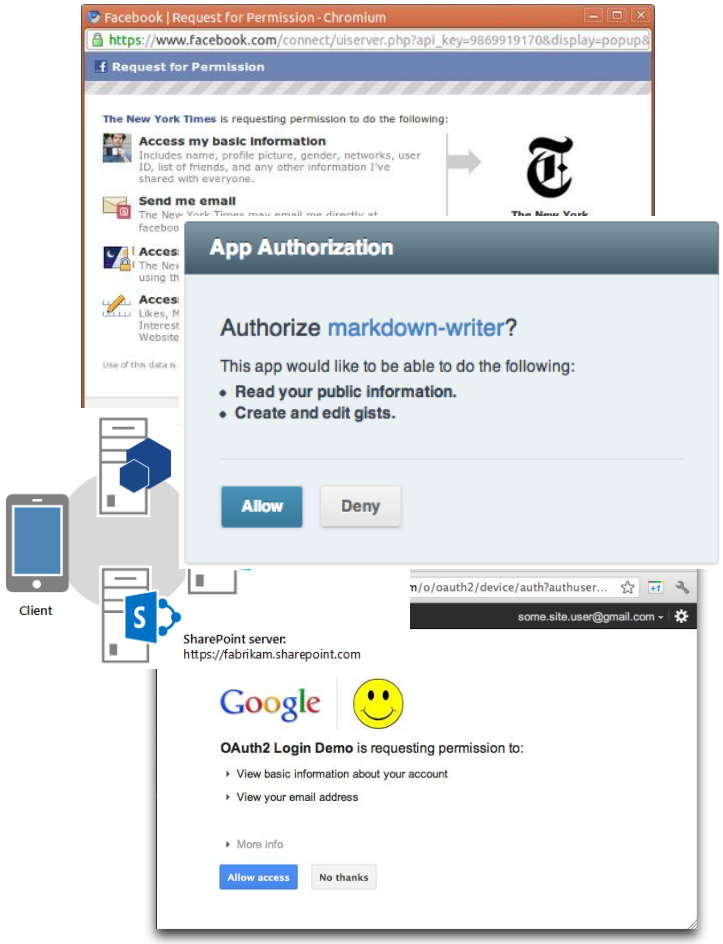
OAuth2 Instanzen verwalten eigene Autorisierungen

- Zugriff auf einzelnen Service
- Ein Schlüssel zur Identifikation des Nutzers
- Alle Nutzer sind bekannt

- So zu finden bei
  - Facebook, Google, GitHub, SharePoint,…

- Interaktion einer Anwendung mit dem Service im Namen des Benutzers

- Randbedingungen:
  - Autorisierung für jedes Einzelsystem
  - System „spricht“ OAuth oder nicht



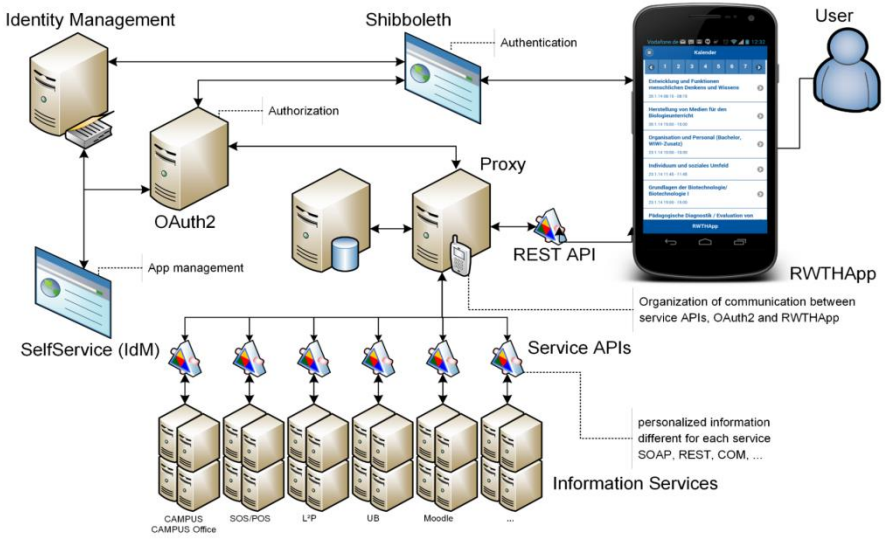
OAuth2 Kooperatives Szenario

Einzelne OAuth2 Instanz verwaltet alle Autorisierungen

- Zugriff auf mehrere Services
- Gemeinsame Schlüssel zur Identifikation des Nutzers
- Alle Nutzer sind bekannt

- Nur eine Autorisierung für alle Services
- Interaktion mit allen Systemen im Namen des Nutzers
- OAuth kann „vorgeschaltet“ werden

- Randbedingungen:
  - Vertrauen zwischen OAuth und Services
  - Zentralisiert nicht verteilt



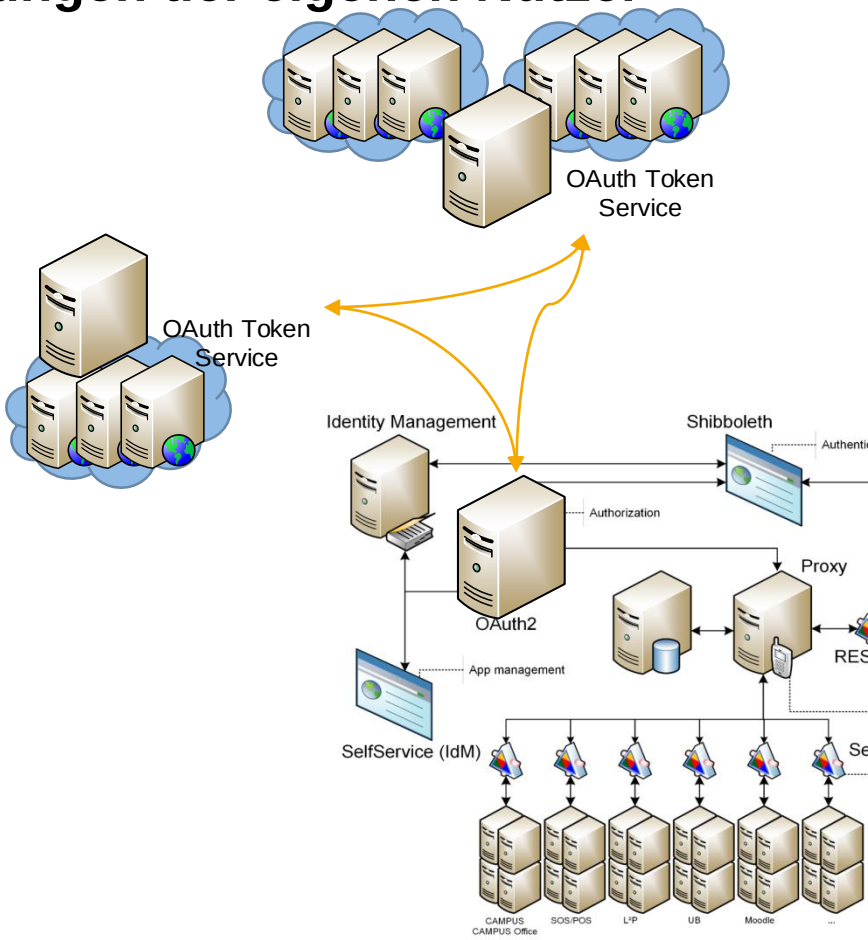
OAuth2 Föderatives Szenario

OAuth2 Instanzen verwalten Autorisierungen der eigenen Nutzer

- Zugriff auf mehrere Services
- Keine gemeinsamen Schlüssel zur Identifikation des Nutzers
- Nutzer sind nicht in allen Services bekannt

- Lokal wie das Kooperative Szenario
- Interaktion von verteilten Systemen
- Autorisierungen bleiben in der Heimatorganisation

- Randbedingungen:
  - Services müssen mit unbekannten Nutzern umgehen
  - Vertrauen zwischen OAuth Inszanzen



Ausblick

- Erweitertes Reporting für
  - App Entwickler
  - Service Betreibern
  - Nutzer
- Implementierung des Föderativen Szenarios
- Anbindung weiterer Services an der RWTH Aachen