Verteiltes Ressourcenmanagement mit der SaxID API

FRMS

Daniel Schreiber

TU Chemnitz, Universitätsrechenzentrum

21. März 2017

Projekt SaxID – Projektziele

- Projekt von 7 sächsischen Hochschulen, später 13
- Evaluieren:
  - Vereinheitlichung von Diensten
  - Vereinheitlichung Identitymanagementsysteme
  - einheitlicher einfacher Dienstzugang
  - Dienste sachsenweit anbieten
  - Abstraktionsschicht für Identitymanagement in Sachsen, Metaverzeichnis?
  - Welche Dienste? Dienstetypisierung
- gefördert aus SMWK Initiativbudget

Festlegungen

- Dienste klassifiziert
  - reine Webdienste
  - Webdienste mit weiteren Schnittstellen
  - Windowsdienste
  - Unixdienste
- Metaverzeichnis verworfen
- einheitliches IdM verworfen
- bestehende Infrastruktur nutzen
  - DFN AAI
  - Eduroam

DFN AAI an der TU Chemnitz

- Strategie: alle neuen Dienste mit Shibboleth oder Kerberos Authentifizierung
- aktuell 147 Serviceprovider produktiv, davon 82 im URZ
- IDP: SimpleSAMLphp
- SP: Apache + mod_shib
- Shibboleth SP in Plattformmanagement integriert

Motivation

- Nutzer:
  - „Ich will es bequem und einfach.“
  - „Mir ist egal, wer mir den Dienst erbringt.“
  - „Ich will **ein** Portal, wo ich meine Ressourcen verwalten kann.“
- Betreiber:
  - „Ich will erprobte Technologie“
  - „Ich will nicht jeden Dienst anbieten müssen, der gerade in Mode gekommen ist“
  - „Ich will Ordnung. Keine überflüssigen Accounts/Daten.“
  - „Wann kann man Ressourcen wieder freigeben?“
  - „Wie kann ich fremde Dienste in mein Service-Management-Portal integrieren?“
  - „Wie integriert man Dienste so, dass sich Dienste, die von einer Partnereinrichtung erbracht werden, so anfühlen wie von der Heimathochschule?“

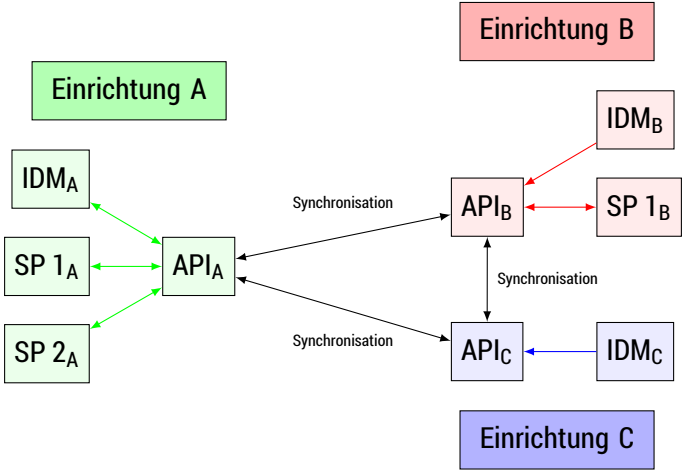
Warum nicht nur DFN AAI nutzen?

- Authentifizierung: DFN AAI ✓
- Fragestellungen bei Deprovisionierung:
  - Ist Account noch gültig? ✓ (z.B. mit Attribute Query)
  - Können Daten des Accounts gelöscht werden? ✗
- Push: Konfiguration des Dienstes im Servicemanagement der Heimathochschule ✗
- Reporting:
  - Wieviele Ressourcen belegen sie dort? ✗
  - Welche Dienste von Fremdeinrichtungen nutzen meine Nutzer?

Lösungsansatz: SaxID API

- verteilte Datenbank
- Synchronisiert Wissen über Ressourcenzustand und Nutzer
- Datensparsamkeit als Designziel → pro Instanz folgende Daten:
  - Daten eigener Nutzer, die eigene und fremde Dienste nutzen
  - Daten fremder Nutzer, die eigene Dienste nutzen
- Schnittstelle zwischen Serviceprovider und Heimateinrichtungen
- keine GUI für Endnutzer

API Betriebsmodell



Einsatzszenario

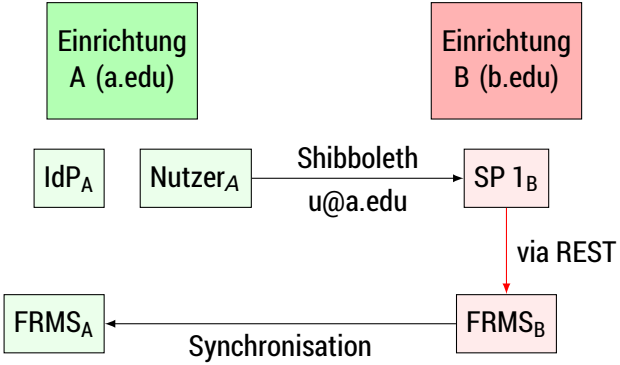


Abbildung: Initialer Ablauf schematisch

Einsatzszenario

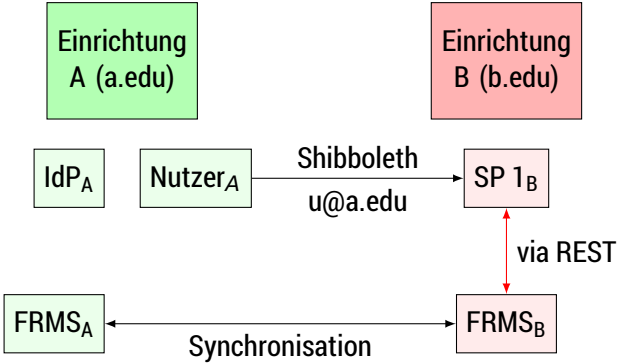
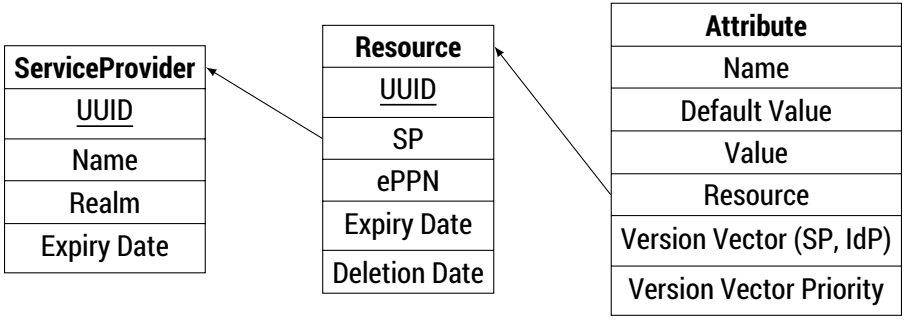


Abbildung: Ablauf im Betrieb

Datenmodell



Synchronisationsprotokoll

- Datenaustausch per REST/JSON
- Ressourcen/SPs per UUID identifiziert
- Attributaktualisierung auf beiden Seiten möglich
- Konflikterkennung über Versionsvektor
- pro Attribut Vorrangregel für Konfliktlösung
- wechselseitige Authentifizierung per Shared Token
- URI Schema
  - `/res` POST Ressource registrieren
  - `/res/<uuid>` GET/PUT/DELETE Ressourcendetails
  - `/services/<uuid>` GET Serviceprovider
  - `/health` GET Monitoring

Dedeployment

- Steuerung komplett über Heimateinrichtung
  - IdM setzt Ablauf- und Sperrdatum
  - SP muß nicht zwischen Nutzerstatus abgelaufen/gesperrt/. . . unterscheiden
- SP vergleicht Expiry Date ⟶ Ressource blockieren
- SP vergleicht Deletion Date ⟶ Ressource löschen

Anforderungen, Laufzeitumgebung

- Python 2.7 oder >= 3.4
- MySQL oder PostgreSQL
- WSGI kompatibler Webserver z.B. Apache
- Linux, FreeBSD

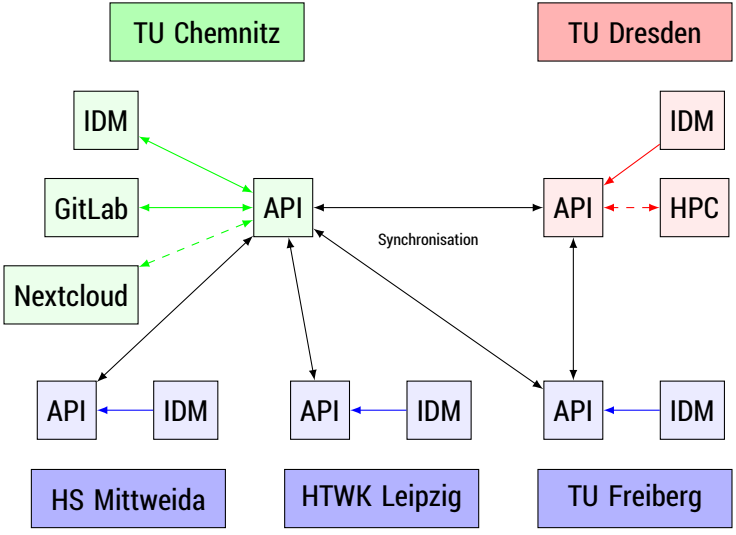
Beispiel SP: GitLab

- Event: „Neuer Nutzer“
  - nutzt Gitlab System Hooks
  - neuer Nutzer: Gitlab macht POST-Request an definierten URI
  - eigenes Djangoprojekt übersetzt in SaxID API
- Regulärer Betrieb
  - Synchronisation zwischen GitLab und SaxID API per Cronjob
  - nutzt GitLab REST API

Beispiel SP: Nextcloud

- Shibboleth Integration über eigene Nextcloud App
- Sync Client über Gerätepasswörter oder eduroam Authentifizierung
- SaxID-Integration über eigene Nextcloud App
- Event: „Neuer Nutzer“
  - Listener auf Nextcloud Hook
  - registriert Nutzer in SaxID API
- Regulärer Betrieb
  - Synchronisation zwischen Nextcloud und SaxID API per Cronjob
  - Nutzung der Nextcloud Provisioning API
- Konfiguration der Quota über IdM/Service Management der Heimateinrichtung

Aktueller Stand Umsetzung



Projektergebnisse

- SaxID API an mehreren Hochschulen produktiv
- GitLab
  - GitLab-Dienst wird gut angenommen, über 500 Nutzer, davon mehr als 100 anderer Einrichtungen
  - durchweg positives Feedback
  - Datenschutz ist bei Nutzern relevantes Thema
  - Public Cloud Dienste werden genutzt, wenn es keine Alternativen gibt
- nächste Schritte
  - weitere Dienste sollten integriert werden:
    - Produktivbetrieb Nextcloud (TUC)
    - Integration HPC (TU Dresden)
    - Integration Sharepoint (HTWK Leipzig)
  - Integration weiterer Hochschulen
  - Verbesserung der Dokumentation
  - Initialisieren von Partnerbeziehungen sollte einfacher werden