

Keycloak @ DESY

applications for PUNCH4NFDI

Instance for Mapping between AAI and Resources / Services

Johannes Reppin & Peter van der Reest

Agenda

Talking points

Motivation

- Why Keycloak
 - User Federation
 - OpenID Connect

At DESY

- Implementation
- Integration Helmholtz AAI
 - OIDC Groups
- DESY user management

Lessons Learned

- Setup
- Configuration
- Realm Management

Outlook

- Plans for Keycloak

What is Keycloak?

Overview



- “Identity and Access Management”
 - Authentication layer for Services
 - OpenID Connect, OAuth2 & SAML2.0 Applications
 - Web services / Token based services
- User Federation
 - Integration with DESY User Registry
- Role Base Authorization

Why Keycloak

Overview



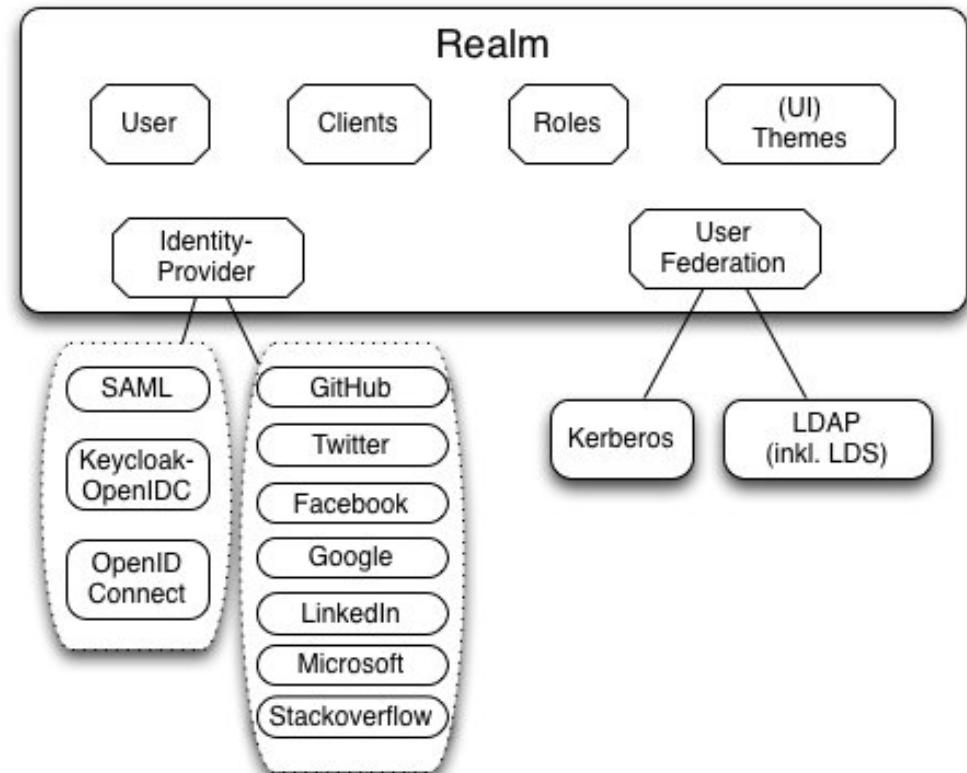
- Getting started with OIDC
 - Standard conform
- Already established in many other Sites
- Open Source (Apache 2.0 licence)
- Many Extensions / Community Modules / Examples available
 - PrivacyIDEA plugin for 2FA
- De-facto standard for many communities (e.g. PaN)
- Java Application built on top of Wildfly Quarkus
 - Cloud Native approach, Kubernetes integration

Keycloak

Concepts



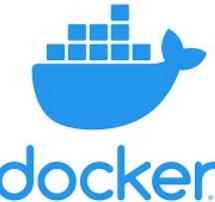
- Identity Brokering:
 - Integration of other IdPs
 - Google, Facebook, *Helmholtz AAI*, ...
- Separated in individual "Realms"
 - Every Realm independent
 - SSO *within one Realm*
 - Applications are "Clients" in Realms
- LDAP login for DESY Users
- Roles "realm-" or "client" based
 - Use for access control or *token attributes*



Keycloak

Setup at DESY

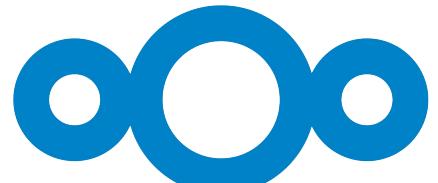
- Rancher Managed Kubernetes Cluster
- Gitlab CI to build custom Keycloak Docker image
 - Integration of extensions
 - Add scripts
 - [WIP] themes
- Use *Helm Chart* to deploy image
- PostgreSQL Database
 - Docker Container is *stateless*



Established Integrations

Overview

- JupyterHub
- Nextcloud
- Hedgedoc
- Gitlab
- ...
- *“If it speaks OAuth2 / OIDC it works with Keycloak.”*
- OIDC examples *usually* with Keycloak
- **BUT: Most Applications only use Authentication layer!**
 - Already provided by Helmholtz AAI



Nextcloud



HedgeDoc

Mapping Resources

“Claims Mappers“

- Helmholtz AAI gives certain Attributes
 - eppn, eduperson_entitlement
 - VOs
 - eduPerson not understood by all Applications
 - URN schema
 - Keycloak gets Attributes and can *transform*
 - Built in *mappers* for standard attributes
 - Add *Javascript Mappers* for easy scripting
 - Used by **Nextcloud** for groups

```
10 var groupClaim = user.getAttribute('groups');
11
12 /**
13  * urn:geant:h-df.de:group:Helmholtz-member#login.helmholtz.de#
14  * urn:geant:h-df.de:group:DESY#login.helmholtz.de
15  * urn:geant:h-df.de:group:HIFIS#login.helmholtz.de
16  * urn:geant:h-df.de:group:HDF#login.helmholtz.de
17 */
18 var ArrayList = Java.type("java.util.ArrayList");
19 var groups = new ArrayList();
20 /* var groupList = groupClaim.split('##'); */
21 var forEach = Array.prototype.forEach;
22 forEach.call(groupClaim, function(group) {
23     groupParts = group.split(':');
24     var gIndex = groupParts.indexOf('group');
25     if (gIndex > -1) {
26         groups.add(groupParts.splice(gIndex + 1).join(':'));
27     }
28 });
29
30 /* token.addClaim('group', newGroups.toString()); */
31 exports = groups;
```

DESY User Storage Backend

“Registry” Integration



- Keycloak works Out of the box for most settings
- Add federated users to DESY “*User Registry*”
 - Software developed at DESY
 - for delegated account, group and resource access management
 - Integration with Keycloak needed
 - Write Java extension
 - Call REST API with User JSON to create user in background
- Open Source ecosystem helpful
 - But: Java skills necessary

Lessons learned

Ongoing learning curve

- Easy Setup: Docker image / Helm Charts
 - **Tip:** Start with Kubernetes deployment!
 - Single node k8s works well (*kind*, *k3s*, ...)
 - *Initial Configuration:* not complicated
 - *Deep dive:* Many options available, can be overwhelming
 - New concepts (authentication flows, scopes, roles)
 - Apps often just use *authentication*
 - *Use Helmholtz AAI directly?*
 - Extra layer of complexity
 - Authorization is *hard*

Outlook

Work in progress

- 2FA / MFA with PrivacyIDEA
 - Works with OTP and webAuthN
- Allow x509 certificate authentication with user certs
- Add Kerberos Authentication
- More authorization concepts
- Multi Realm federation
 - Keycloak can add its own Realms as IdP
 - Allows for better separation and cross-authentication



SECTIGO



GÉANT
Networks • Services • People

Integration User Registry 2

Konzept



Neue User in der Registry bekannt machen

Erweiterung: Keycloak - Registry Interface

Personen-Provisionierung in IAM

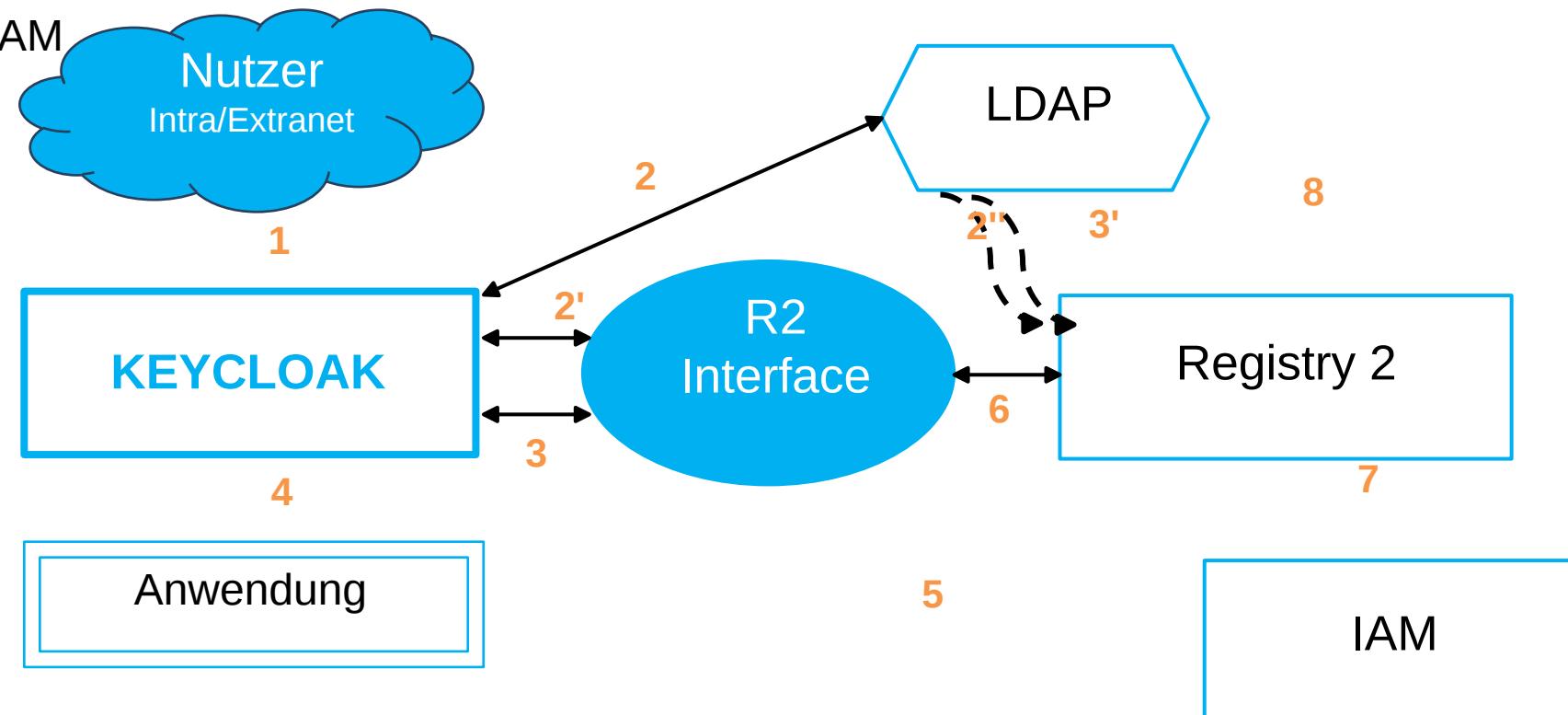
Problem

Artefakte-Lifecycle Mgmt

(Dateien, Beiträge in Wiki's)

Nachweisbarkeit (wer war's),

Zuordnung (wem gehört's)



Keycloak Entwicklung:
viele Beispiele

[https://github.com/
thomasdarimont/keycloak-extension-playground](https://github.com/thomasdarimont/keycloak-extension-playground)

Status Implementierung

•Custom OIDC Claims



OIDC Token von Keycloak

Token kann beliebig verändert werden.

Eigene `Mapper`

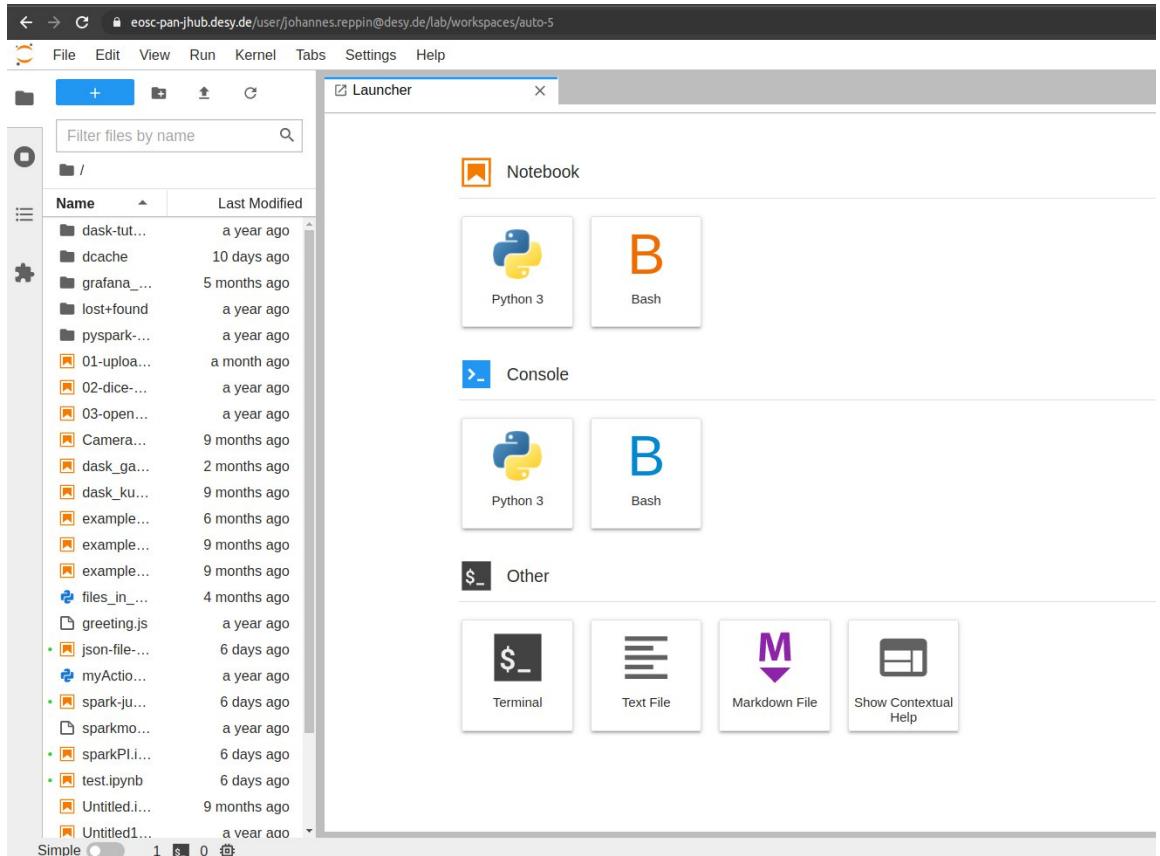
Attribute verändern und hinzufügen

UID aus LDAP

LDAP Gruppen

Rollen

...



Cloud JupyterHub mit dCache Mount und Unix UID