Experiences and status quo of IAM / CAAI in DataPLANT

Authors:
Dirk von Suchodoletz*, Jonathan Bauer[2], and Marcel Tschöpe[3]
*Lead presenter
[1]dsuchod@uni-freiburg.de, Universität Freiburg
[2]jonathan.bauer@rz.uni-freiburg.de, DataPLANT, Universität Freiburg
[3]marcel.tschoepe@rz.uni-freiburg.de, DataPLANT, Universität Freiburg

Abstract:
The various tools and services in DataPLANT need to be authenticated to allow access to certain resources. There are primarily two types of authentication and authorization required: users provide credentials to run tools and interact with services. Tools and services need to authenticate to each other e.g. to exchange information and data. Already in the very beginning of the project we identified key objectives to implement proper identity and access management for our service landscape and later on further scientific communities in both centralized cloud and decentralized on-premises installation:

- Provide community members with access to the various resources within DataPLANT without requiring to create a completely new online identity
- Derive a DataPLANT identity from various AAI, primarily Life Sciences AAI or ORCID iD, and provide an SSO environment
- Be flexible and open to include further identity providers to adapt to ongoing developments and future requirements like the integration with other scientific communities within the NFDI
- Build upon existing standards, use Open Source tools and create as few as possible solutions proprietary to DataPLANT
- Be open and flexible to integrate with third party services like Galaxy or Nextflow provided by others
- Provide an easy to use interface to manage DataPLANT identities both by administrators and DataPLANT users, allow for as much self-service as possible

The DataPLANT user management builds upon existing Community AAIs (CAAI). Well established services like Life Sciences AAI and ORCID can be combined with local authentication within the central DataPLANT authentication service. The infrastructure is based on Keycloak, a widely used AAI proxy developed by RedHat, that supports modern authentication protocols like OpenID-Connect or SAML. It allows the integration of multiple AAIs and identity brokering in an unified environment. Providing an AAI identity management, which can easily be connected with GitLab and other services through either protocols, simplifies the DataPLANT wide user management. The connection of multiple AAIs through KeyCloak enables our community to use their existing accounts, for example from the Life Sciences AAI, their home institution or ORCID. During authorization we can assign different roles depending on the account source or on specific attributes. Permissions can be derived from these roles to differentiate between users. These range from privileged users having full access to the data and the ability to create archives/publications, to underprivileged users that have only a reporting function and/or read-only access to raw data. However, authorisation and accounting is still a major challenge. To avoid lengthy negotiations between organizations, it is necessary to ensure that authorized users can only

use the resources to which they are entitled, otherwise a potentially unfair pricing model emerges on both sides. To a large extent, this can be addressed through a well-defined resource request process that takes into account funding and partnerships - which can be integrated into an automated process through user attributes in the respective community AAIs. This issue needs to be discussed at all levels: NFDI AAIs, other federated AAIs (such as the state project bwIDM2) and local institutions. In the last couple of months we started to interact with the IAM4NFDI team (https://base4nfdi.de/projects/iam4nfdi) to explore further options of cross-NFDI integration. We started to discuss the following issues with the RegApp team at KIT, one of the four IAM4NFDI solution options presented in the workshops.

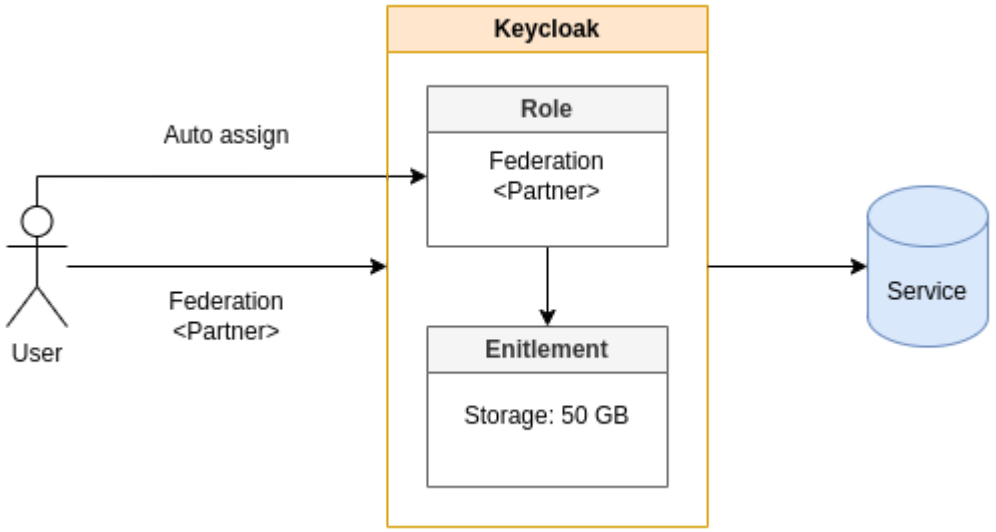

Fig: Assignment of authorisations of a user of a federated cooperation for a service

In this example, a user logs on to the system through a federation partner to access a service. When the user logs in through the federation partner's IDP, the user is automatically added to a pre-defined role that includes all users who log in through the federation partner. All users associated with this role are automatically granted permission to use 50 Gigabytes of storage on the service. To accomplish this, the "Storage: 50 GB" entitlement is added to the aforementioned role. [Fig]

Während Authentication weitflächig gelöst, sind sinnvolle Abbildung von Gruppenzugehörigkeiten oder die Zuweisung von spätere Abrechnung von genutzten Ressourcen eine größere Herausforderung. Das gilt beispielsweise für signifikante Speicher- oder Compute-Bedarfe.
* Wie kann man individuellen Nutzern bestimmte Ressourcen zuweisen, die sie bei der Nutzung des Dienstes anfordern können
* Dabei Mapping von verschiedenen Rollen auf verschiedene Ressourcen
* **Wie stellt man sicher, dass diese Ressourcen nicht überschritten werden (d.h. dass der Anbieter "kein Geld verliert" bzw. der Nutzer bzw. dessen Einrichtung keine überraschend hohen Rechnungen erhält, für die keine Mittel hinterlegt wurden)**
*** Wie bildet man das sinnvoll ab, so dass hier nicht längliche Aushandlungsprozesse zwischen Nutzern, ihrer Einrichtung und den Dienstanbietern notwendig sind**

Fragen/Überlegungen aus bwIDM2 zu Attributen, Ressourcen-Zuweisungen im Rahmen von Authorization und Gruppenverwaltung mit Blick auf NFDI ...

Lösungsmöglichkeiten zur Kopplung mit Regapp / Community? Hinsichtlich NFDI (aber nicht nur) wäre es sicher sinnvoll, wenn man eine Ressourcenverwaltung hätte, die – ähnlich Jards – mit der RegApp gekoppelt wäre.
So ganz grob:
* Jards: Ressourcenanträge
* RegApp: AAI und Community (delegierte PI und Gruppen)
* ResMan: Verwaltung von Ressourcen/Kontingenten/evtl. in Verbindung mit hinterlegten monetären Mitteln/… Das wäre sicherlich für diverse Services eine interessante Ergänzung.