MESH AAI

Technical workshop #2

David Antoš, Milan Daneček

Task 2.1

19. 8. 2020

Introduction

- brief introduction to AAI in the Mesh
  - mainly from user's perspective
- scope/limits of this presentation
  - we explain mainly the distributed approach
  - we discuss personal data protection quite briefly
  - we omit error handling completely (it gets ugly)
- for picky details enjoy our documents
  - available at `wiki.cs3mesh4eosc.eu/wps/2/Task21`

What We Build Upon

- we have OCM protocols/APIs to establish sharing
  - between pairs of sync’n’share system instances
  - needs to be enabled on peer-to-peer basis
  - we kinda-sorta expect users to know
    - which system their partner is using
    - what is their partner’s ID in the remote system
- two levels of unfriendliness
  1. difficult to set up for admins
  2. next-to-impossible to use for average users
     - I don’t know federated IDs of my colleagues myself

cesnet datacare

Goals of the AAI

1. establishing a way to “connect a service to the Mesh” without peer-to-peer agreements
   - we focus just on AAI-related topics here
   - there are many operational aspects as well
2. enabling users to establish sharing with minimal knowledge of their partners

- minimise personal data disclosure
   - need good reasons for personal data transfers

First Ideas

- OK, let’s unify our identity management
  - eduGAIN as a source of identities
  - unified group management based on eduGAIN
    - such as eduTEAMS

First Ideas Crashing I

- there are too many sources of primary identities
  - e.g. Sciebo uses the federation only to create local accounts kept alive for grace periods
- there are even more sources of group membership information
  - “field-of-research”/“it’s-for-our-infrastructure” group mgmt systems are currently the leading edge (ELIXIR/LifeScienceID, EUDAT, ...)
- there are sync’n’share installations that handle group membership internally

First Ideas Crashing II

- EOSC services are expected to have “AARC-compatible” user management
  - the AARC Blueprint Architecture expects IdPs and/or Attribute Authorities as AuthZ sources
    - “group membership” ≈ “AuthZ source”
    - through attributes/entitlements
  - but it doesn’t solve group management
- we do not expect sufficient group management unification under EOSC in 3 years
  - and total unification… never?

Terminology

- resources—files/folders/app access to be shared
- originating system—the sync’n’share system with the resource
- originating user—the user initiating the sharing (aka “Peter”)
- target system—the sync’n’share system to share to
- target user—who shall gain access to the resource (aka “Albert”)
- sharing policy—describes what is allowed
  - typically incoming and outgoing

Design Starting Points

- resource sharing information is kept in the originating system
  - for local as well as remote users
- group membership source is the sync'n'share system
  - we don't care where groups are defined
    - e.g., an external identity mgmt
- centralised or distributed?
  - central broker for share set-up
    - Central Component (CC)
  - vs. distributing the functionality
    - central Configuration Database (CD) and distributed Executive Modules (EMs)

cesnet datacare

User Scenarios

- we discuss data sharing scenarios from users' point of view
- note that data transfer is (nearly) the same
- scenarios
  - 1.a target user's identity is known by the originating user
  - 1.b (user discovery as a central service)
  - 1.c invitations via a separate channel
  - 2 group sharing
  - 3 access from "Mesh-enabled" apps
  - 4 access to remote apps

Target User’s ID Known I

- let’s pretend (for a moment) that the originating user knows target user’s system and user ID there
  - we’ll need it as a building block later
- let’s have a Configuration Database holding Mesh metadata
  - nodes, public keys, services running there, contacts, . . .
  - similar to identity federations
- let’s have Executive Modules sitting in front of the sync’n’share systems
  - config of the Mesh is regularly pushed to the EMs
  - the EM execute policies of the site (outgoing, incoming)

Target User’s ID Known II

- sharing, Peter to Albert:
  1. Peter initiates sharing
  2. EM of the originator verifies outgoing policy and contacts the target system
  3. EM of the target verifies signatures, its incoming policy, target user ID
  4. Albert is offered a new share (to accept/reject)
  5. access tokens are exchanged
- the architecture solves “how to add nodes to the Mesh”—to register their metadata
- this scenario may be useful for targets previously used by Peter
  - personal “target IDs address book” for Peter

User Discovery

- “how Peter learns Albert’s system and ID?”
- an argument for centralised broker: user discovery service
- but:
  - all systems would have to propagate users to the discovery service
  - based on... well... consent?
  - what user data should be shown to make it useful?
    - name and obfuscated email address?
  - next-to-impossible to decentralise
- result: impractical, legally unwise

Invitations by Email I

- or “Invitations via a Separate Communication Channel”
- how not to force Peter to know Albert’s ID or even system?
  - email is the most probable Albert’s contact Peter knows
  - any textual communication can be used as well
- let Peter send an email invitation to share
- and let Albert choose the target system
- $\rightsquigarrow$ some kind of Service Discovery/Where Are You From-like (WAYF) service is needed
  - running at Executive Modules

Invitations by Email II

- the procedure
  - Peter composes an invite in the originating system (using just Albert's mail)
    - containing link to originating system WAYF instance
    - and authorisation code
  - Albert gets a mail with a link to originating system's WAYF
  - Albert opens the link, chooses the target system and logs into it
  - tokens are exchanged to grant access
- optionally, Peter may approve the share in the last phase
  - to verify Albert's usage of the link is legitimate

Group Sharing I

- Peter creates a share and delegates management of Albert's colleagues to Albert
- sharing in the originating system can be set up for local or remote users, local groups, and remote groups
- group enumeration
  - direct: originating system can list all individual users
    - including members of groups defined at targets
  - reverse: target system can list all individual users
    - including members of all groups defined elsewhere

Group Sharing II

- enumeration needs crazy level of complexity
  - avoid enumeration hell completely
  - group management is a tool for Peter to delegate management of groups local to the target system to Albert
    - no enumeration supported
    - Peter trusts Albert (or just doesn’t care ;))
    - otherwise Peter should invite individual users and manage the group purely by himself
- procedure:
  - configuration—similar to “Target user’s ID known”
  - Albert must tell Peter the target system and the group name
  - for advanced users

Access Data from Apps

- an app with “save to/load from the Mesh” capability
- WAYF/Service Discovery interface necessary
  - Mesh metadata must contain list of available apps
  - can run at the application site
- “save to the Mesh” $\rightsquigarrow$ WAYF $\rightsquigarrow$ log in
  - policy of the sites is checked throughout the process
  - tokens are exchanged
- note: this can be achieved by sharing the data to the app site
  - use case nice, but low priority

Access to Remote Apps

- “open this file with a remote app” $\rightsquigarrow$ WAYF with app list $\rightsquigarrow$ selecting the app
  - must handle token exchange for temporary data access
- permission to access an app $\approx$ to access a share
  - e.g. by invitation
  - access granted by app administrator
  - similar to previously discussed methods

Summary

- we have described use cases
  - mostly from user's point of view
- designed to solve
  - usability of data sharing for the users
  - scalability of Mesh administration
- to do
  - "invitation by email" being implemented, other scenarios to follow
  - more detailed specs needed
    - site administration side, error handling, ...
  - define processes to establish the Mesh in greater detail