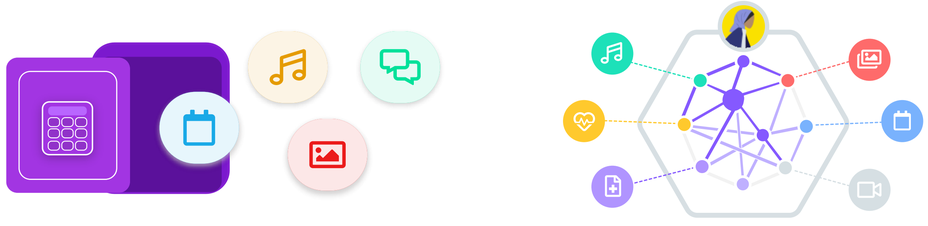
Decentralized collaborative note making application using Solid

Fraunhofer FIT

Pulkit Arora[1], Timothy Clemens Borrell[2], Omar Mahmoud[3], Haihua Yang[4]

Computer Science 5 - Information Systems and Databases | RWTH AACHEN UNIVERSITY

Solid Overview[2,4]



- One of many data exchange models
- Data storage option through decentralized pods
- There are no data type restrictions
- User has power over access rights for other users and is able to change them every minute
- Access rights are controlled through personalized and unique IDs
- Features that allow data sovereignty (Access control lists)

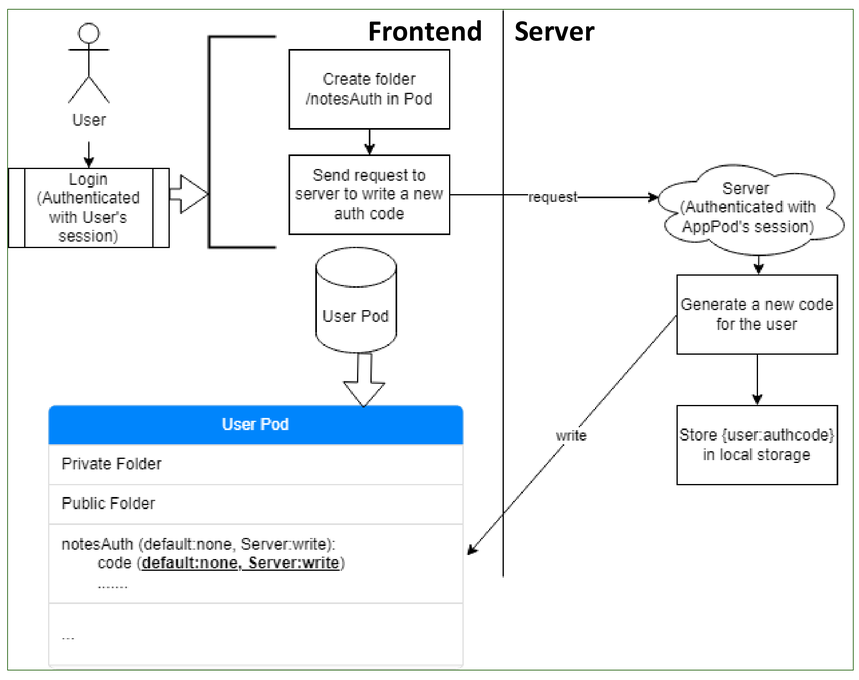
Use cases for Solid:

The Solid data exchange model is being used in many different fields. These range from social applications to games, ratings of products, movies, project management, notes and publishing, finances, geolocations and many others.

This project has focused its usage on note creation and the handling of its access rights.

Architecture Overview[1,2]

Authentication and Session Management



Functionalities of the model

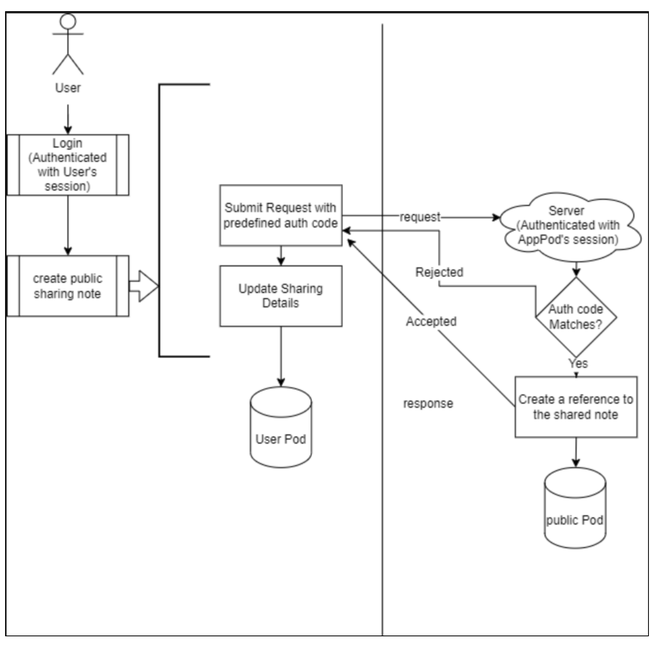
Frontend
- Notes management
- Adding access for public and friends

Server
- Writes note references to the respective folders, public or with friends

Access Options[1,3,4]

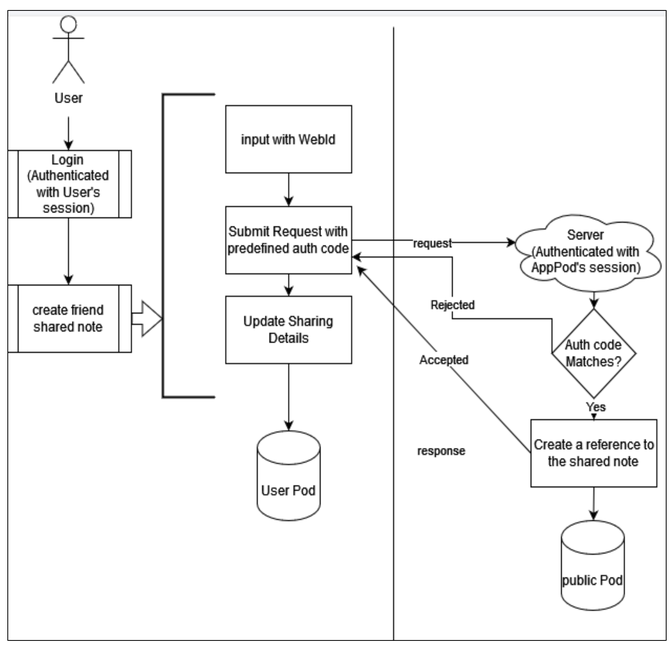
Overview of the system architecture for granting access

Public sharing



User saves the note to their own pod
Reference to the public pod will be created while saving the public note
Then it is visible to all users

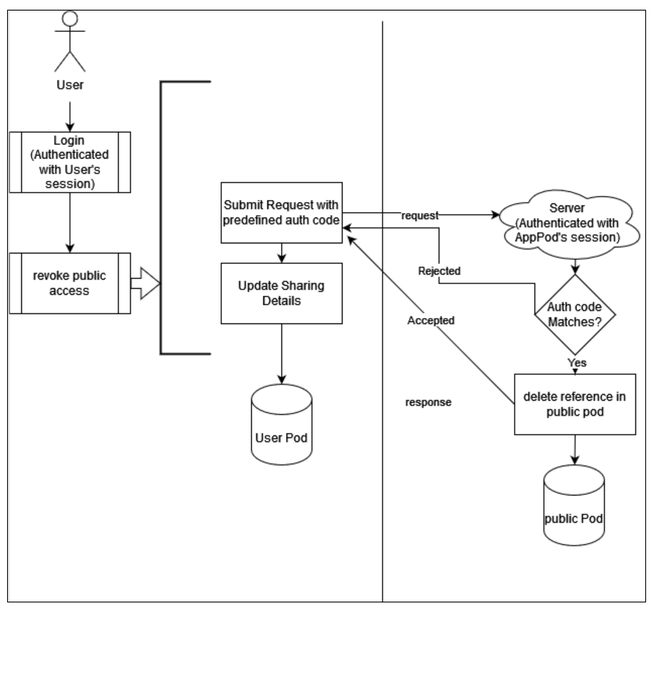
User sharing



- User enters the WebID for another particular user
- User chooses the writing access or reading access sharing
- Create and save note in user pod, while saving the reference to another user
- Create reference to this note in public pod

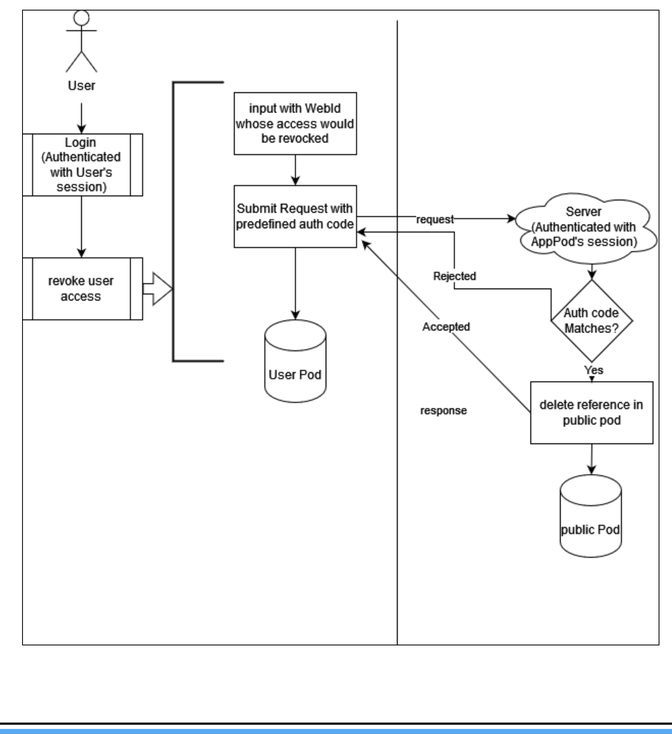
Overview of the system architecture for revoking access

Public revoke access



- User chooses to revoke public access
- Post request to public pod
- Delete reference in public pod

User revoke access



- User inputs with the WebID
- User chooses to revoke access for this WebID
- Post request to public pod to delete reference

AppPod storage overview / Note Structure[1,2]

Note vocabulary
- Every personal Pod must contain the folder *"Notesdump"*, in which all notes are being stored
- Note vocabulary is uniquely identified through RDF (Resource Description Framework)
- The schema that has been used in this project was: *<http://schema.org/...>* (E.g. note descriptions were stored and then accessed via *<http://schema.org/description>*)
- Information stored in notes:
  - Name of the note
  - Description
  - Note Creation date
- A "sharedNoteRef" is a reference to the shared note with a user.
- It contains the URL to the actual note which is stored in the pod of the note Author

AppPod Directory Structure

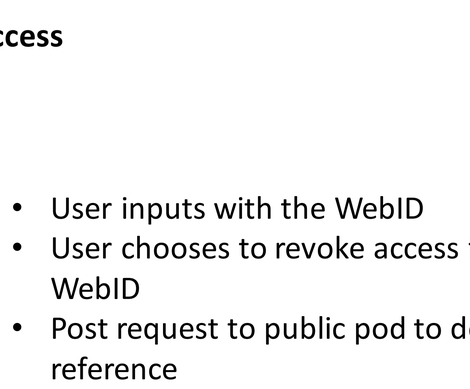
```
.
├── <AppPod>
|   ├── profile
|   ├── private
|   .
|   .
|   ├── publicSolidPodFile (default: append)
|   |   ├── #sharednote1ref (default: read)
|   |   ├── #sharednote2ref
|   └── ├── #sharednote3ref
|   ├── Users
|   |   ├── User1
|   |   |   ├── #sharedNote1ref (default: none, user1: read)
|   |   |   ├── #sharedNote2ref
|   |   ├── User2
|   └── ├── User3
└── ├── public
```

Functionalities for notes in the Frontend[1,2,3,4]

Implemented Functionalities
- Personal data management
  - Add note
  - Edit note
  - Delete note
- Sharing
  - Create a public note
  - Share a note with a different user (friend)
- Control
  - Revoke public access to a note
  - Revoke a friend's access to a note
  - Give reading or also writing access
- Collaboration
  - Edit a shared writable note

Note frontend overview

Adding Note Template



Displaying Existing Note

Team Information and roles

1. **Pulkit Arora:** Frontend-server interaction, Pod login, Session managment, Authentication, Save Note, User/Public sharing, Revoke Access
2. **Timothy Clemens Borrell:** Frontend visuals, Show shared notes, Save note, Edit note, Read note, Creating Note structure, Bug fixes, README.md
3. **Omar Mahmoud:** Frontend routing, Code restructure, Delete note, Revoke access, Designing features, Creating illustrative flowcharts
4. **Haihua Yang:** Frontend and server setup, Public sharing, Revoke Access, Load public Note, Show public note, Bug fixes, Read note, Flowcharts