

Systeme

Fabaccess

Siehe auch [22 Automatisierte Maschinenberechtigungsprüfung](#) - dem Post-Corona-

Links

- [Stand des aktuellen ZAM-Fabaccess-Experiments](#)
- [Konzept](#)
- Ein *wenig* [Doku](#)
- [Gitlab Repo](#)
- [Hardware für Fabaccess](#)
- NFC
 - [Fabreader-Repo](#)
 - [Doku](#)
- Artikel
 - [Heise](#)
- Einführungsvideos
 - [1. Einführung](#); [2. Einführung](#) (Gleicher Inhalt, aber andere Leute. Die zweite ist länger und hat mehr Inhalt)
 - [Vorstellungung](#)

Funktionsumfang

TODO

- Freigabe von Maschinen
- Ausführen beliebiger Skripte
- Abrechnung von Material
- Passende Hardware
- Kommunikation über MQTT
- Hat eine App (was kann die?)
- Abrechnung zwischen den Fablabs (?)
-

Einschätzung

Konzepte

Viele der Konzepte finde ich (morty) überaus Problematisch. Sehr viel "not invented here" für Probleme, die bereits von anderer Software gelöst sind.

- Benutzer und Maschinenverwaltung in Plain-Text-Dateien. (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> ab 13:40)
- Sie sind der Meinung, dass ihre Dateibasierte eigene Datenbank besser ist als eine fertige DB zu verwenden (Morty: Ich Zweifel ernsthaft an deren Kompetenz). (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> ab 15:00)
- Auditlog in einer Datei und nicht der DB (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> 23:20)
- Rechteverwaltung nur hierarchisch mit Wildcard am Ende, wobei read/write, etc teil der Hierarchie ist. Immerhin gibt es Rollen (kommen danach) (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> ab 29:50)
- Sie haben ein sehr interessantes Bild von Rechtehierarchien "Parents" sind eigentlich Children - Evtl denken sie auch einfach Objekthierarchien.... *schrug* (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> 23:20)
- LDAP als externe Benutzerverwaltung geplant - aber keine Rechte/Gruppen (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> (104:00)
- Zertifikate werden vom Client aktuell nicht geprüft (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> 107:00)
- Man kann entweder Rollen vergeben oder auch nicht - Rollen nur für bestimmtes zu vergeben geht nicht und ist auch nicht geplant (
<https://recordings.prototypefund.de/presentation/2392dda5fdeb3d4ae538b4738c98e3aeb557847b-1702480827153/meeting.mp4> 113:00)

Diflouroborane / bffh

Windfisch bestätigt den Eindruck, dass sie sehr viel neu erfunden haben und sehr viel Zeit für Nebenschauplätze drauf geht. Grundsätzlich hat er aber den Eindruck, dass der Code OK ist.

Fabreader-Code

Der Code hat weder Design noch Architektur. Zum Glück ist er nicht sehr umfangreich, so dass er mit überschaubarem Aufwand umgebaut werden kann. Einige der Architekturentscheidungen sollte man noch mal diskutieren - hält sich aber alles in Grenzen.

Fabmanger

Kommerzielle Lösung, die aber unter AGPL zur Verfügung steht. Eher auf betreute bzw kommerzielle Spaces ausgelegt. Open Source, in Ruby geschrieben.

- [Webseite](#)
- [Source auf Github](#)
- [Funktionsumfang](#)
- [Online Demo](#)

Funktionsumfang

Siehe auch [Funktionsumfang](#)

- Kalender
- Räume
- Maschinen
- Reservierungen
- SSO
- Mitgliederverwaltung (viel mehr als wir wohl brauchen)
- Abrechnung
- Materielverwaltung
- [API-Schnittstelle](#) Leider sind die meisten Zugriffe nur lesend, was für viele unsere Anforderungen nicht ausreichend ist.
- ...

Einschätzung

Auf Kommerzielle Projekte ausgelegt. Dafür wirkt der Code aufgeräumt und dokumentiert. Es gibt eine Architekturdokumentation.

Rosenguarden

Links

- [Konzept](#)
- [Github](#)

Funktionsumfang

*

Einschätzung

Das Konzept ist konträr zu Fabaccess: Möglichst wenig selber machen, auf fertiges zurückgreifen, Ordentliche Schnittstellen um leicht erweitern zu können.

Das Backend ist in Python und das Frontend in Vue geschrieben.

Fabman

[Fabman](#) kann wahrscheinlich alles was wir wollen, ist aber einfach viel zu teuer.

Revision #8

Created 2024-01-08 16:25:35 UTC by Moritz S

Updated 2025-06-28 10:44:30 UTC by Moritz S