Maschienenzugangs system (und Türen und....)

Rechte zu Zugängen und Maschienen

Siehe auch Siehe auch 22 Automatisierte Maschinenberechtigungsprüfung - dem Post-Corona-Projekt

- Systeme
- Überblick
- FabAccess Notizen

Systeme

Fabaccess

Siehe auch 22 Automatisierte Maschinenberechtigungsprüfung - dem Post-Corona-

Links

- Stand des aktuellen ZAM-Fabaccess-Experiements
- Konzept
- Ein *wenig* Doku
- Gitlab Repo
- Hardware für Fabacces
- NFC
  - Fabreader-Repo
  - Doku
- Artikel
  - Heise
- Einführungsvieos
  - 1. Einführung; 2. Einführung (Gleicher Inhalt, aber andere Leute. Die zweite ist länger und hat mehr Inhalt)
  - Vorstelleung

Funktionsumfang

TODO

- Freigabe von Maschienen
- 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

Maschienenverwaltungsystem mit Hardware

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.

Überblick

Zielsetzung

Ziel ist es Zugang zu Maschinen, und Räumen zu verwalten.

Wichtige Links

- Übersicht über die verwendeten Systeme in anderen Makerspaces der DWGD
- PDF über die vorhandenen Zugrangssysteme der Offenen Werkstätten mit Bewertung
- Sammlung der Zugangssysteme der Offenen Werkstätten (mäßig Hilfreich)
-

Anforderungen

- Original Pad
- Übersicht über die untersuchten Systeme

(H) High Prio
(M) Medium Prio
(L) Low Prio
(R) Rejected

 Wird nicht unterstützt
 Wird unterstützt
 Wird vielleicht unterstützt
 Nicht zutreffend -> Muss z.B. in Hardware implementiert werden

Prio	Anforderung	BFFH	Fabmanager
	Berechtigung		

H	Anbindung SSO / LDAP	 (LDAP für Passwort angedacht)	
H	*Einfaches einpflegen von Berechtigungen/Einweisungen*, durch Einweisende		
M	Zeitabhängige Berechtigungen z.B. Presslufthammer nur zwischen 6 und 18 Uhr,		
H	Auslaufende Berechtigungen z.B. Zugang zum Aquarium vom 1.1.23 bis 3.1.23.	In planung	
H	Gruppenbasierte Berechtigungen		Kann über Trainings abgebildet werden
	Maschienen		
H	aktivieren/freischalten z.B. durch * Strom ein/aus * Interlock trennen/verbinden		- Aber API/Pluginsystem
M	Zustand überwachen (z.B. gerade in Nutzung) Material verbrauch, z.B. bei Drucker In Betrieb Betriebsstundenzähler	Alle aktivitäten werden in eine Logdatei geschrieben, die man auswerten kann	- Müsste man erweitern
M	Präsenzsprüfung ("Totmannschalter")		
H	Maschinenspezifische An-/Abmeld/Betriebs Verhalten		
H	Maschinenabhängigkeiten (z.B. Lüfung und Klappen korrekt geschaltet)		
L	Admin Tokens/Zugänge sollten offline noch funktionieren		
M	Internet-Unabhängiger Funktionserhalt		

H	Türen öffnen		Rechteabfrage über Plugin
L	Spinde/Fächer für Privatzeugs		
M	Spinde/Fächer für Geräte		
M	Reservierungen von Maschinen (und Räumen)		
H	Protokollierung von Nutzung	(Logdatei)	
H	Anbindung an Abrechnungssystem		
H	Letzte x Nutzer:innen	Nur der letzte / Sonst logdatei	
TODO	Materialabrechnung		
R	Warteschlange: -> Wird mit Papier/Whiteboard gmeacht		
H	RFID Nutzerschnittstelle		- Aber Plugin-system
M	Display für Feedback		
H	Sicheres RFID-Protokoll (Für Türen) - Siehe auch extra Paragraph		
H	Basisfunktionalität sollte komplette per RFID gehen Maschinen Räume		
	App / Webapp zur Bedienung	Nur Textdateien	

RFID

Zum Öffnen der Türen muss es sicher sein. D.h. es sollte nicht möglich sein durch Demontage des Lesers an den Schlüssel zu kommen. Im Inneren ist es ok, wenn man an den Schlüssel kommt - der muss dann halt ggf ausgetauscht werden. -> Verwendung von zwei unterschiedlichen Schlüsseln.

FabAccess Notizen

Community Call 04.03.2023

- SSO/LDAP wurde angefangen in die API einzubauen, klingt aber noch nach work in progress
- Claims auf Ressource können künftig überschrieben werden durch andere Nutzer, wenn Ressource in einem bestimmten Zustand (z.B. fertigen 3D-Drucker übernehmen, der nicht freigegeben wurde)
- "Ablaufzeit" über "Script dranhängen an API"
- Verleihzustand vs. Gerätezustand soll dargestellt werden
- Gerätezustand über Initiator z.B. von MQTT in FabAccess übertragen; z.B. Stromverbauch für x min < y, dann freigeben
- Komplexe Abhängigkeiten z.B. in Homeassistant abbilden und Ergebnis-Signale an FabAccess übertragen
- Ablaufende Berechtigung über API möglich, Implementierung in Kern WIP
- OAuth soll implementiert werden, wird aber nicht alle Mechaniken erlauben
- Login mit gescanntem QR Code als Alternative zu Benutzername/Passwort
- Hackwerk Aalen baut eigenen FabReader (Gehäuse), Zusammenbau der existierenden Designs hat nicht geklappt
    - https://gitlab.com/sfz.aalen/projekte/fabaccess-case