

Was ist FabAccess?

FabAccess ist ein leistungsstarkes, zentrales Zugangs- und Maschinenkontrollsystem für den automatisierten Verleih von Maschinen und Geräten (Management und Zuweisung von Ressourcen). Diese Selbstbedienungsfunktion ermöglicht es Benutzern über unsere App (Borepin) den Verleihprozess eigenständig durchzuführen, wodurch Betreuer von offenen Werkstätten ihre Aufmerksamkeit auf andere Tätigkeiten wie Ausbildung und Wissensvermittlung richten können und einen besseren Überblick behalten.

Die Anforderungen für FabAccess wurden mit Mitgliedern des Verbunds Offener Werkstätten e.V. und mit Betreibern von FabLabs an Hochschulen (Fab:UNiverse) entwickelt. Das daraus entstandene Lastenheft findet sich in unserem GitLab Projekt.

Maschinenverwaltung

Das Prinzip von FabAccess ist es, Maschinen über den Stromanschluss ein- und auszuschalten. So sind Maschinen im Normalfall stromlos und damit ungefährlich. Erst, wenn Nutzer an der Maschine sind, die für die sichere Bedienung befähigt und eingewiesen sind, wird die Maschine mit dem Stromnetz verbunden.

Wir wollen dabei die Idee von Neil Gershenfeld, dass Nutzer selbstständig den Umgang mit Maschinen lernen, mit den Anforderungen von offenen Werkstätten kombinieren, sodass Nutzer sich dabei nicht verletzen. Folglich benötigen Nutzer für gefährliche Maschinen idealerweise eine Einweisung - dieses Prinzip wird dann im Berechtigungssystem abgebildet.

Noch nicht unterstützt, aber geplant: Die Abnahme von Maschinen, die eine Reinigung nach Gebrauch benötigen. So wird die Nutzbarkeit von Maschinen für den nächsten Nutzer sichergestellt und der Verschleiß verringert.

Noch nicht unterstützt, aber geplant: das Reservieren von Maschinen, wollen aber keine Zeitplanung für die Nutzer übernehmen. Daher ist der Kompromiss, den wir umsetzen, dass Nutzer Maschine nur beispielsweise eine halbe Stunde reservieren können, um so den Weg zum FabLab sich keine Sorgen darüber machen müssen, dass jemand anderes in der Zeit die Maschine für 5 Stunden verwendet.

Berechtigungsverwaltung

Ein zentraler Aspekt von FabAccess ist also das Management von Berechtigungen, das die Steuerung des Zugriffs auf bestimmte Ressourcen ermöglicht. Diese Berechtigungen bieten eine feingranulare Kontrolle darüber, welcher Benutzer auf welche Ressourcen zugreifen kann, und dienen als wichtiges Instrument für die Sicherheit und Effizienz des Systems. Mit FabAccess soll der Zugriff auf Maschinen in Offenen Werkstätten verwaltet werden, um so Unfälle zu vermeiden. Auch der Zugang zu Räumen und Schränken kann organisiert werden.

Ein Anwendungsbeispiel: Die Kreissäge soll nur von denjenigen Personen angeschaltet werden können, die eine Einführung besucht haben. Nach der Einführung bekommen die Teilnehmer*innen über ihre Chip-Karte die Berechtigung für die Aktivierung der Kreissäge. Nur wer diese Berechtigung hat, kann die Kreissäge mit Hilfe der Chip-Karte einschalten.

API und Audit Log

Mit seiner flexiblen API können Ressourcenbesitz und -zustände effektiv abgebildet werden, wobei jede Ressource einem bestimmten Benutzer zugeordnet werden kann. Darüber hinaus bietet es die Möglichkeit, Statusänderungen von Ressourcen zu verfolgen, was eine präzise Überwachung und Verwaltung ermöglicht (Audit Log).

On-premise Ansatz

FabAccess ist ausschließlich selbst gehostet! Es gibt keinen Dienstleister dafür. Jeder Space soll seine eigene Infrastruktur damit abbilden und verwalten können.

Modularität per Plugins und Scripts

Die Modularität von FabAccess ermöglicht eine vollständige Anpassung an die Anforderungen unterschiedlicher Spaces. Der Ausleihprozess ist in unserem Server (BFFH) modelliert, und die Ansteuerung der Maschinen erfolgt über Plugins. Zudem bietet FabAccess die Möglichkeit, eigene Skripte über die API anzubinden, um eine höhere Automatisierung zu erreichen. Diese Vielseitigkeit eröffnet nicht nur die effiziente Nutzung von Maschinen und Geräten, sondern bietet auch Raum für Flexibilität und Anpassungsfähigkeit an die individuellen Bedürfnisse verschiedener Umgebungen

Schreibweise

Die richtige Schreibweise von FabAccess ist "FabAccess", nicht "Fab Access", nicht "Fab-Access" und nicht "Fab:Access"

Ein Maschinenzugangssystem für offene Werkstätten. Das klingt einfach, oder? Hier 1 Bit setzen, Maschine an! 1 Bit zurücksetzen, Maschine aus! Fertig. Leider stellt sich heraus, dass es nicht ganz so einfach ist. Warum brauchen wir das trotzdem?

Es gibt 2 Arten von Maschinen:

- die, die Menschen verletzen
- alle anderen

Es gibt 2 Arten von Werkstattnutzern:

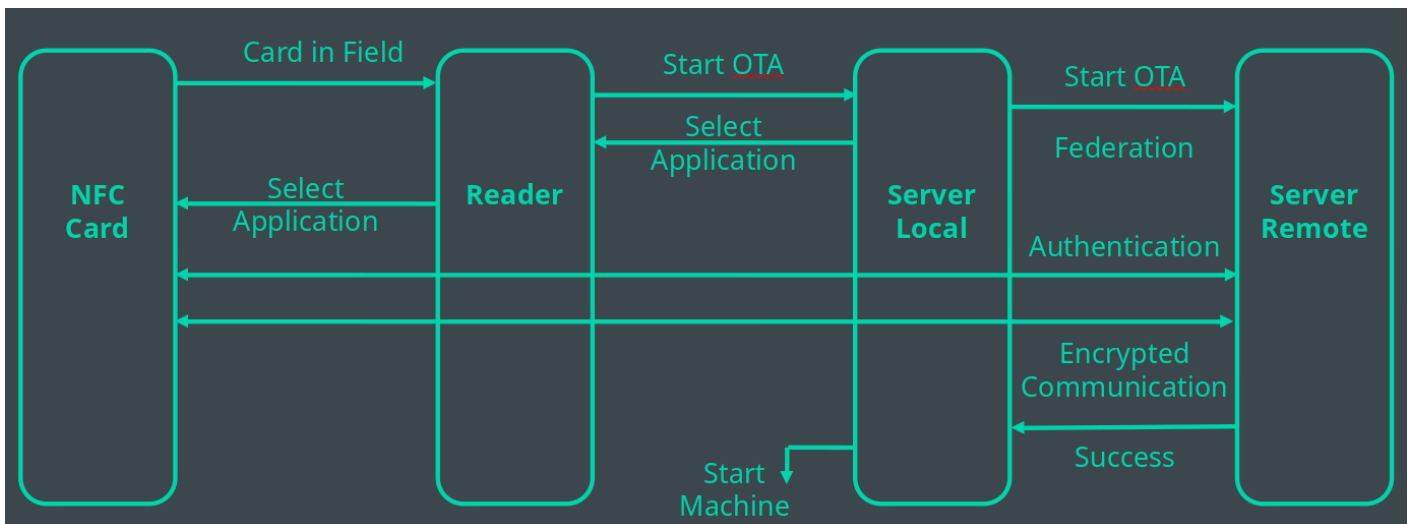
- die, die aufräumen
- alle anderen

Wenn wir mit alledem klarkommen wollen, dann viel Spaß! Oder nutze FabAccess!

Warum ist es nicht so einfach? Weil es verschiedene Hürden bzw. Aspekte gibt, die dafür Berücksichtigung finden sollten:

- Rollenbasierter Zugriff
- Föderation
- Arbeitssicherheit
- IT-Sicherheit
- attributgesteuerter Zugriff
- Stabilität
- Einfachheit der Installation
- Modularität
- verschiedene Maschinen
- Zugänglichkeit
- Zeit & Geld
- verschiedene Werkstätten
- komplexe Aktivierungsschemata

Wir wollen (und brauchen) viel mehr als ein "funktioniert für mich" Tool.



Perspektiven

Für die Werkstattleiter sollte das System ...

- einen sicheren Betrieb garantieren
- soziale Verhaltensrichtlinien etablieren bzw. erzwingen
- ... und ...
- einfach installierbar und konfigurierbar sein
- stabil in laufender Benutzung agieren
- Wartungsaufwand mit wenig bis keinem Aufwand

.... es sollte sich der Werkstatt anpassen, nicht die Werkstatt an FabAccess

Für die Werkstattbenutzer sollte das System ...

- einfach zugänglich sein
- schnell und einfach zu nutzen sein
- den Aufwand reduzieren

Einfachheit in Installation und Konfiguration

- Die Installation von FabAccess wird aktiv dokumentiert
- Die Konfiguration erfolgt über dhall / toml Dateien
 - eine GUI für das Erstellen von Konfigurationen ist wünschenswert
 - die Dateien sollten gut dokumentiert sein - und du fasst sie nur einmal an
- Das Anbinden neuer Maschinen ist einfach über Python, Bash, etc.
 - eine wachsende Zahl von Beispielen finden sich auf GitLab, Github und anderen Plattformen

Wartung

- abwärtskompatible Konfiguration und API ab Version 1.0
- strukturierte Log-Dateien von Ereignissen

Zugänglichkeit

- Die mobile App ist in den meisten App Stores zu finden (inkl. F-Droid)
- eine APK ist frei zugänglich

Benutzung

- Sensible Benachrichtigungen per OLED / e-Ink Display
- Föderation

Aufwand / Nutzungsbeschleunigung

- Maschinen können mit NFC Karten aktiviert werden

Unterstützte Hardware

Out of the box

- DESfire EV2
 - FabReader
- Shelly Plugs (v1 und v2)
- Raspberry Pi / LXC Container / Proxmox

Einfach erweiterbar

- Python, Bash, alles, was Cap'n Proto anspricht

Probleme:

- jede Werkstatt, jede Maschine ist anders. Deshalb gibt es so viele Varianten, dass es nur einen generischen Ansatz geben kann, wenn es ein Tool für alle oder viele von uns werden soll

Zweck der API

- Anbindung für Leute, die keine Core-Coder sind
- nicht jeder hat ein smartphone und in nicht jedem fall macht es sinn!

Fragen aus den Reihen:

- did you guys think about interoperation already existing systems like 3id?
- how do you stop the people from unplugging plugs from the machine to switch on/off -> hier ist es eine Frage zwischen technischen UND sozialen Fragestellungen! wie man bestrafen kann: der letzte gemeldete Nutzer im System muss sich rechtfertigen, was passiert ist (Log File) -> log parsing ist nicht mehr im Scope von FabAccess
- do you have the possibility to get the signal from the machine to check if its working? -> it depends! the core of FabAccess is switching on/off. we do not need MQTT data, but we CAN use it to make even more with it
- fabaccess soll kein ERP System sein und es dient auch nicht primär zur Abrechnung von Maschinen

Was ist es nicht?

FabAccess ist kein System zum Lösen sozialer Probleme, z.B. wenn Nutzer in Spaces Maschinen unzweckmäßig verwenden! FabAccess dient lediglich dazu, Maschinen zu schalten und bei der Auswertung und Organisation zu helfen. Vertrauen und Umgang muss trotzdem von Mensch zu Mensch geschaffen werden!

Version #19

Erstellt: 11 Oktober 2024 16:55:35 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 14 Dezember 2024 12:02:03 von Mario Voigt (Stadtfabrikanten e.V.)