Was ist FabAccess?

Video-Intro

Ein kurzes Erklärvideo zeigt dir, was FabAccess ist und was es kann:

https://www.youtube.com/embed/CjBEGqvV_ms

Allgemein

FabAccess ist ein leistungsstarkes, zentrales Zugangs- und Ressourcenkontrollsystem für den automatisierten Verleih, Management und Zuweisung von Maschinen und Geräten, Türen, Schließfächern, Meetingräumen und sonstigen Dingen. Diese

Selbstbedienungsfunktion ermöglicht es Benutzern über unsere App (Borepin) den Benutzungs-, Rückgabe- oder 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 ist im GitLab Projekt zu finden.

Ein Zugangssystem 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 ... und alle anderen
- **Es gibt 2 Arten von Werkstattnutzern:** die, die aufräumen ... alle anderen

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

Warum ist das 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 Ressourcen
- Zugänglichkeit
- Zeit & Geld
- verschiedene Werkstätten
- komplexe Aktivierungsschemata

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

Perspektiven von FabAccess

Für die Werkstattleiter sollte das System ...

- einen sicheren Betrieb garantieren
- soziale Verhaltensrichtlinien etablieren bzw. erzwingen

... und ...

- einfach installierbar und konfiguerbar sein
- stabil in laufender Benutzung agieren
- Wartungsaufwand mit wenig bis keinem Aufwand

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

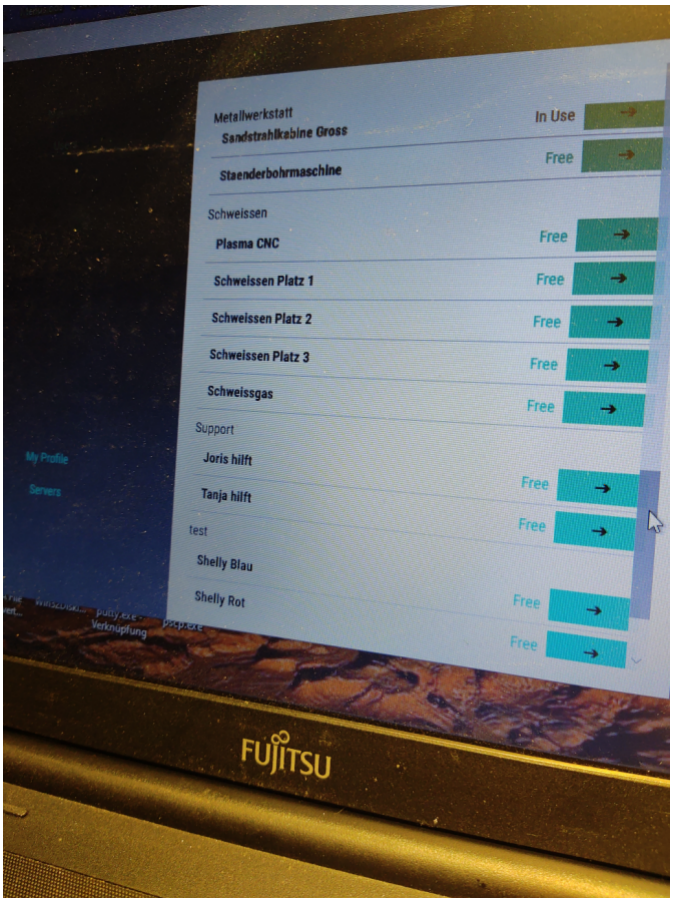
Für die Werkstattbenutzer sollte das System ...

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

Ressourcenverwaltung

Das Prinzip von FabAccess ist es, Maschinen bzw. allgemein Ressourcen über den Stromanschluss ein- und auszuschalten. So sind Maschinen im Normalfall stromlos und damit relativ 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.

Eine Ressource kann aber im generischen Fall auch eine Person sein, die für Support gebucht wird.



Beispiel für Supporter-Ressourcen "Joris hilft" und "Tanja hilft", Makerspace Bocholt

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 umsetzten, 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 Ressourcen 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).

Zweck der API ist es auch eine Anbindung für Leute anzubieten, die keine Core-Coder sind.

Modularität per Plugins und Scripts

Die Modularität von FabAccess ermöglicht eine vollständige Anpassung an die Anforderungen unterschiedlicher Spaces. Die Ansteuerung der Ressourcen erfolgt über Plugins (Aktoren, Initiatoren).

Zudem bietet FabAccess die Möglichkeit, eigene Skripte über die FabAcess-API (angebunden werden kann alles, was Cap'n Proto anspricht - egal ob in Python, Bash, C, Perl, ...) 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. Eine Referenzimplementierung für die FabAccess-API wird in Python bereitgestellt und nennt sich pyfabapi.

Einfachheit in Installation und Konfiguration

- Die Installation von FabAccess wird aktiv dokumentiert
- Die Konfguration 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 Ressourcen ist einfach - z.B. über Python, Bash, etc.
  - eine wachsende Zahl von Beispielen finden sich auf GitLab, Github und anderen Plattformen

On-premise Ansatz

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

Wartung

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

Zugänglichkeit (Client App)

Die mobile App ist in den meisten App Stores zu finden (inkl. F-Droid). Eine APK ist frei zugänglich. Weitere Details siehe https://fab-access.org/downlooad.

Aufwand / Nutzungsbeschleunigung

- Ressourcen können neben der manuellen Auswahl in der App-Übersichtsliste auch per QR-Code oder mit NFC Tags vorausgewählt und schneller aktiviert werden
- Die Benutzerauthentifizierung kann auch mit Mifare DESFire EV2 Karten und FabReadern umgesetzt werden. In diesem Fall ist dann keine Client App für den Benutzer notwendig!

Unterstützte Hardware - out of the box

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

Was ist FabAccess nicht?

Wie kann man verhindern, dass die Leute die Stecker aus der Maschine ziehen, um sie ein- oder auszuschalten? Die kurze Antwort: FabAccess ist kein System zum Lösen sozialer Probleme, z.B. wenn Nutzer in Spaces Maschinen unzweckmäßig verwenden, indem sie beispielsweise in der Werkstatt Maschinen oder Kabel manipulieren, um sie trotzdem einzuschalten! 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!

Über Fragen und Antworten, wie unpflegsames Benutzerverhalten geahndet werden soll, sollte sich jede Werkstatt ein eigenes Konzept überlegen und die entsprechenden Möglichkeiten erörtern - zum Beispiel das Auswerten des Audit Logs.

FabAccess ist zudem kein ERP-System und kein System für die Abrechung von Kosten, der Pflege von Maschinenführerscheinen oder dem Inventar. Für diese Komponenten bedarf es separater Lösungen, die über geeignete Schnitstellen sinnvoll kommunizieren sollten.

Wir haben jedoch eine Reihe von Softwaresammlungen angelegt, die ggf. bei genau diesen Fragestellungen weiterhelfen.

Schreibweise von FabAccess

Die richtige Schreibweise von FabAccess ist "FabAccess"

Falsch sind

- "Fab Access"
- "Fab-Access"

Außerdem gab es in den Anfangsjahren die Schreibweise "Fab:Access", als es aus dem FAB:UNIverse Kontext kam. Dadurch, dass jedoch verschiedene Begriffe wie FabInfra, FabHardware, FabFireCard, FabFire und FabReader dazu kamen, wurde auf den Doppelpunkt verzichtet.

Version #34
Erstellt: 2024-10-11 16:55:35 CEST von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-03-05 22:30:47 CET von Mario Voigt (Stadtfabrikanten e.V.)