

Frequently Asked Questions (FAQs)

Ist FabAccess erweiterbar?

Ja, du kannst FabAccess um deine eigenen Bedürfnisse erweitern. Wenn du mehr Steuerung deiner Nutzer benötigst, dann kannst du zum Beispiel über Python ein Skript schreiben, dass unsere Cap'n Proto API von BFFH anspricht und so Gruppen zuteilt. Um die Anbindung von Abrechnungssystemen zu ermöglichen, bietet BFFH ein Audit Log an, das dir in Echtzeit die aktuellen Aktionen in deinem Space ausgibt. So kannst du zum Beispiel Nutzungszeiten für Maschinen sekundengenau abrechnen. Oder wenn du ein Grafana Dashboard für einen Infomonitor brauchst, ist dass auch darüber möglich.

Sollten dir die von uns unterstützten Initiatoren oder Aktoren nicht ausreichen, dann kannst du deine eigenen Module einbinden, dafür musst du nur unsere Initiatoren und Aktoren API ansprechen.

Um den Einstieg in die erweiterbare Konfigurierbarkeit zu ermöglichen, stellen für alle unsere APIs eine Python Bibliothek zur Verfügung.

Wie schaltet man Maschinen?

Für den Anfang trennen wir Maschinen vom Strom, somit können wir kostengünstige Lösungen wie den Shelly Plug oder jedes mit Tasmota flashbares System unterstützen. Die Erweiterung auf andere Möglichkeiten unterstützen wir dabei durch unser modulares System. Da die Kommunikation mit der FabCard etwas Schwieriger ist, stellen wir unseren eigenen FabReader als Open Hardware Projekt zur Verfügung sowie die dazu passenden Software. Unseren FabReader findest du als Projekt bei GitLab.

Wie lässt sich Social Engineering verhindern?

Wenn jemand seine FabCard oder seinen Login weitergibt, dann wird das zum Problem desjenigen, der die Karte bzw. den Login an jemand anderen weitergegeben hat.

Warum gibt es Borepin nicht für Linux? Wird es einen Web-Client geben?

Weil sowohl Xamarin, als auch .NET MAUI beim Kompilieren auf Linux Probleme machen, gibt es keinen fertigen und funktionierenden Linux-Build. Ein Wechsel auf AvaloniaUI war zuletzt angedacht, jedoch wurde dies wegen mangelndem

WebAssembly Support aktuell eher verworfen.

Gedanken der Entwickler in kurz: Es ist nicht nötig einen Web-Client aus C# nach WebAssembly zu bauen. Je nachdem, wie genau der Stack am Ende aussehen soll, ist es möglicherweise einfacher den Web-Client komplett getrennt von Borepin zu entwickeln. Dazu bedarf es einer vervollständigten Cap'n Proto Übersetzung für JavaScript oder das Backend müsste eine andere API anbieten. Alternativ ließe sich auch nur der API-Adapter in WebAssembly entwickeln, während das User Interface über übliche JavaScript Web Frameworks entwickelt wird. Eventuell ist es einfacher, dem Rust Backend eine Web API zu spendieren.

Wo ist der Unterschied zwischen einer Maschine in Benutzung und einem geblockten Schließfach?

Welche Karten sind mit FabAccess kompatibel?

Kurz: Ausssschließlich Mifare **DESFire** von NXP Semiconductors. Karten wie Mifare Classic oder NTAG werden **nicht** unterstützt. DESFire Karten sind bis heute nicht gehackt und gelten als sicher!

Warum verwendet FabAccess MIFARE DESFire Karten?

Zunächst allgemein: DESFire Karten sind kostengünstig, und bisher ungeknackt und deshalb sehr sicher - im Gegensatz zu MIFARE Classic oder MIFARE NTAG. DESFire Karten sind quasi nicht so einfach klonbar, wenn man den Inhalt nicht kennt. So kann sichergestellt werden, dass es von einer Karte nicht mehrere Exemplare auf dem Schwarzmarkt gibt.

Die RFID-Karten sollen die ID des Nutzers speichern, und das verschlüsselt. Denn wer die Karte des Nutzers besitzt, kann damit Schabernack machen.

Im Übrigen empfehlen wir das Bedrucken der Karten mit FabAccess / FabFire Logo!

Warum ist die FabInfra Community auf Telegram und Element gleichzeitig?

Weil FabInfra historisch gewachsen ist und der Verbund Offener Werkstätten dafür eine immer größere Rolle einnahm. Deshalb wurde Der Zulip-Chat 2022 niedergelegt und die Telegram-Gruppe (wo mehr Mitglieder sind; Stand 14.10.2024) wurde zwar nicht von Element abgelöst, aber für den Verbund entsprechend geöffnet. Das macht es allerdings auch nicht einfacher für die Kommunikation.

Wieso nutzt FabAccess keine SQL-Datenbank für die Verwaltung von Zuständen und Benutzern?

Das ist eine gute Frage! Die Antwort fehlt leider noch. Kritische Stimmen wünschen sich eine Integration von SQLite, MariaDB, PostgreSQL oder anderen Good Practices.

Aktuell werden Benutzer per erstmaligem Import einer `users.toml` in die interne Datenbank geladen. Ist einmal ein Admin im System vorhanden, kann dieser Admin die Nutzer dann auch per App anlegen und verändern. Die Benutzerbasis kann im Ganzen auch durch Re-Import einer (angepassten) `users.toml` aktualisiert bzw. wiederhergestellt werden.

Wie ist die Interaktion von User und FabAccess?

Erster Weg: Jeder Nutzer kann sich die App herunterladen, bekommt eigene Zugangsdaten und kann sich selber anmelden, bzw. Maschinen anschalten. Auf den Maschinen kleben ausgedruckte QR-Codes mit spezieller URN, die per Smartphone gescannt werden können.

Zweiter Weg: Die Nutzer*innen bekommen Chip-Karten, die auf einen Kartenleser aufgelegt werden, dann geht die Maschine an. Es gibt auch die Variante, dass die Karte nur kurz drangehalten wird, die Maschine angeht, die Karte nochmal drangehalten wird und die Maschine aus geht.

Erfolgt der Zugriff von der App auf den Server lokal via WLAN oder muss der Server öffentlich erreichbar sein?

Der Server kann, aber muss nicht öffentlich erreichbar sein. Der Benutzer muss nur entsprechenden Netzwerkzugriff auf den Zielserver bekommen. Das ist eine Aufgabe für den lokalen Netzwerkverwalter des jeweiligen Spaces.

Wie geht man mit Mehrfachanmeldungen um, z.B. wenn jemand die Berechtigung für drei Maschinen hat?

Bisher ist es möglich und auch gewünscht, dass mehr als eine Maschine genutzt werden kann. Das bedeutet, dass es durchaus in Ordnung ist, wenn ein Benutzer ggf. sogar von mehreren Geräten aus gleichzeitig eingeloggt ist. Falls ein Benutzer sich einloggt und einen anderen Benutzer an die Maschine lässt und etwas passiert, dann ist das das Problem des Benutzers, der den anderen ran gelassen hat (siehe *Wie lässt sich Social Engineering verhindern?*)

Version #36

Erstellt: 14 Oktober 2024 16:19:01 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 27 Dezember 2024 01:41:48 von Mario Voigt (Stadtfabrikanten e.V.)