

# Frequently Asked Questions (FAQs)

Verschiedene häufig gestellte Fragen und Antworten. Deine Frage ist nicht beantwortet? Dann [schreib uns gern!](#) Viele Fragen beantworten sich auch beim Lesen vom [Getting Started](#).

## Allgemeine Fragen zu FabAccess (Konzept)

### Was ist FabAccess?

FabAccess ist ein leistungsstarkes, zentrales Zugangs- und Ressourcenkontrollsystem für den automatisierten Verleih von Dingen wie Maschinen und Geräten (Management und Zuweisung). 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.

### Was kann mit FabAccess gesteuert werden?

Eigentlich so ziemlich alles - egal ob physische Hardware oder virtuelles Script! Es ist lediglich ein Fragekomplex aus Zeit, Nutzen, Aufwand, Ressourcenbedarf. Vieles ist möglich, jedoch sollten die umzusetzenden Sachen in der Regel leicht reproduzierbar, stabil und preiswert sein.

Denkbare Steuerziele können z.B. sein:

- Ein-/Ausschalten von Steckdosen (Zwischenstecker wie z.B. Nous A1T oder Shelly Plug)
- Ein-/Ausschalten von Deckenlichtern oder Hofbeleuchtung
- Öffnen und Schließen eines Garagen- oder Hallentores
- Öffnen und Schließen einer Eingangstür
- Öffnen und Schließen von Schließfächern
- Öffnen und Schließen von Lüftungsklappen von Absaugungen
- Öffnen und Schließen von Druckluftventilen (z.B. für Arbeitsplätze oder für Zublasung von Geräten)
- automatisierte Audioansagen in der Werkstatt ("wir schließen bald", "Achtung, Achtung!", ...)

## 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, das 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 Ressourcen sekundengenau abrechnen. Oder wenn du ein Grafana Dashboard für einen Infomonitor brauchst, ist das 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 wir für alle unsere [APIs](#) eine Python Bibliothek zur Verfügung.

## Wie lässt sich Social Engineering (Manipulation) verhindern?

Wenn jemand seine FabFireCard oder seinen Login weitergibt, dann wird das zum Problem desjenigen, der die Karte bzw. den Login an jemand anderen weitergegeben hat und dann ggf. etwas schlimmes passiert ist (z.B. Maschinenschaden, Personenschaden). Wer ein System für die Ahndung etablieren will, der kann sich auf dem [Audit Log](#) stützen.

Ansonsten empfehlen wir das Verwenden von sichtbaren Kennzeichnungen wie zum Beispiel [Stickern](#) und [Safeboxes](#) für Kabel, um klar und deutlich für den Benutzer die Aufmerksamkeit auf das System FabAccess zu leiten.

Trotz dieser Maßnahmen kann die Manipulation durch einzelne Nutzer meist nicht komplett verhindert, aber effektiv eingeschränkt werden.

## Wo sind die Unterschiede zu Home Assistant, openHAB und Co.? Hat das Vorteile?

FabAccess ist speziell für die Bedarfe und Abläufe Offener Werkstätten entwickelt und ist weniger ein klassisches Heimautomatisierungssystem. Softwareprodukte (und teilweise auch Hardwarehersteller) wie [Homematic](#), [FRITZ!](#), [Shelly](#), [Develo](#), [ioBroker](#), [openHAB](#), [Home Assistant](#), [Node-RED](#) oder [n8n](#) unterscheiden sich im Aufbau wesentlich von FabAccess. Sie bieten in der Regel Dashboard- und regelgesteuerte Szenen-Lösungen an, um für Räume oder Hausbereiche Monitorings darzustellen und Schnellaktionen bereit zu stellen, wie das Öffnen von Garagentoren oder das Einschalten von Lichtern oder Messen von Temperaturen.

Solche Softwaresysteme können auch die Werkstatt generell bereichern und ggf. auf die gleiche Hardware zugreifen, die auch in FabAccess eingebunden ist, zum Beispiel über Protokolle wie MQTT, Z-Wave, Wifi oder andere.

FabAccess ist jedoch als gesonderter Softwarekomplex zu verstehen. In vielen Fällen gibt es inhaltliche Überschneidungen, sodass im Praxisbetrieb die Frage gestellt werden kann, welche Software in diesem Fall zu nutzen ist - so zum Beispiel auch beim Messen von Stromverbräuchen. Diese Fragestellung kann nicht simpel beantwortet werden und hängt vom Szenario ab.

Klare Kernfunktionalitäten von FabAccess sind:

- FabFire-Protokoll: erlaubt die Verwendung von Smartcards (DESFire EV2) an Terminals und erlaubt die Benutzung von Ressourcen ohne Smartphone oder Tablet
- Anbindung SSO/OIDC/LDAP-Benutzerverzeichnis
- Mechanismen für das Setzen von Zuständen wie "In Benutzung", blockiert, deaktiviert, sowie das Übertragen auf andere Benutzer oder Ausleihen
- direktes Scannen von QR-Codes und NFC Tags zum Zugreifen auf Ressourcen
- Konzept der Föderation
- komplexes Berechtigungsmanagement für Admins, Werkstattdirektoren, Bereichsleiter, Mitglieder, Nutzer und Co.
- Flexible Zuweisung von Ressourcen, die bedient werden sollen. Es lassen sich mehrere Ressourcen gleichzeitig über die Aktionsbuttons im Client triggern. Durch geeignete Maßnahmen sind auch Primary-Secondary Schaltungen möglich.

**Wir haben Geräte mit laufender Garantie oder Zertifikaten und möchten diese mit FabAccess benutzen. Müssen wir diese umbauen?**

Der Umbau von Geräten wird nicht empfohlen, da dies ggf. einen Eingriff in die Betriebssicherheit bedeuten kann (Risikofrage). Insbesondere bei neuen Geräten kann das z.B. zum Erlöschen von Garantie-, Gewährleistungs-, Ersatz- oder Versicherungsleistungen führen. Außerdem ist jede modifizierte Ressource potentiell auch eine durch Nutzer möglicherweise angezweifelte Ressource, die von der Norm abweicht. So muss nicht unbedingt ein ESP32 in einen nagelneuen 3D-Drucker oder ein Schütz in eine Drehbank eingebaut werden. Das würde ggf. auch einen späteren Wiederverkauf erschweren.

Wir empfehlen stattdessen die bestehenden Geräte möglichst von außen zu ergänzen, zu sichern und zu steuern. Am besten eignen sich für 230 V Geräte einfache Zwischenstecker von Shelly oder mit Tasmota Firmware. Bei 400 V Drehstromgeräten ist diese Frage etwas komplizierter zu beantworten. Jedoch findet sich in der Regel mit etwas Nachdenken und Kreativität eine gute und sichere Lösung, die auch Manipulation erschwert oder verhindert und zum gewünschten Steuereffekt mit FabAccess führt. Viele Geräte können so ergänzt werden, dass sie nicht dauerhaft modifiziert werden müssen. Hier helfen z.B. 3D-gedruckte Adapter, geformte Bleche, Aufkleber, genormte Abdeckkästen und sonstige Komponenten. Auch bei der Montage gibt es verschiedenste Techniken wie das Anwenden banaler Kabelbinder, Klebepads, Rohrschellen, Magnete und Co.

Es ist teilweise leichter Gerätschaften mit gültiger [CE-Kennzeichnung](#) zu nutzen und ggf. um ebenso CE-zertifizierte Zwischensteckdosen zu ergänzen, als DIY-Lösungen zu implementieren. Jedoch hängt auch die Fragestellung wiederum von der Art und Größe der Werkstatt ab. Jede FabAccess-kompatible Umbauaktion ist letztlich auch eine Frage von Preis und Aufwand/Nutzen.

Wir sammeln als Community verschiedene Lösungen und stellen sie unter anderem im Bereich [FabHardware](#) vor.

Du hast eine Lösung, die noch fehlt? Teile sie uns gerne mit! Du hast die Lösung nicht gefunden? Frag uns gern! Siehe <https://fab-access.org/join>

### **Wie sollten wir unsere Elektrik planen, wenn wir FabAccess nutzen wollen?**

Das kommt drauf an, was es für eine Werkstatt ist, wieviele Nutzer sie hat und welche Ressourcen dort stehen. Viele "normale" Geräte mit 230 V Schuko-Stecker können ganz einfach über Schaltsteckdosen (z.B. Nous A1T mit Tasmota) installiert werden.

Bei 400 V Geräten ist das etwas aufwendiger (z.B. CNC-Fräse, Drehbank, Drechselbank, Plasmaschneider, Schweißgerät, Keramikbrennofen, etc.). Hier ist es häufig angebracht, die Schalt- und Messelektrik direkt in die in der Nähe befindlichen Schaltkästen zu integrieren.

### **Generelle Tipps:**

- Netzwerk ausbauen. In der Nähe der Geräte sollte sich mindestens ein gutes Wifi-Netz befinden. Hierzu müssen meistens Access Points installiert werden. Entsprechende Ethernet-/Glasfaserkabel für das Netzwerk und etwaige Switches und Racks sind dann in der Aufplanung nutzbringend, um eine gute Basisinfrastruktur zu garantieren.
- es lohnt sich eine Übersicht zu erstellen, die auflistet, welche Ressourcen wie z.B. Maschinen, Handwerkzeuge und Schließfächer mit FabAccess genutzt werden sollten oder könnten. Hiervon kann relativ gut abgeleitet werden, welche Infrastruktur im Gesamten benötigt wird (z.B. Sicherungsautomaten, Netzwerkports, Wifi Spots) und mit welchen Aktoren Strom geschaltet werden könnten
- nichts ist so beständig wie der Wandel - das betrifft auch die Werkstattausrüstung: Wenn immer möglich sollten die Aktuatoren an den Ressourcen verbaut werden (auch die mit Drehstrom
- mit FabAccess lässt sich auch das Licht der Hausinstallation schalten, wenn es entsprechend elektrisch so konnektiert ist!
- Weiterhin lassen sich auch super Hauptschalter für Werkstätten mit FabAccess realisieren. Wird im jeweiligen Untersicherungskasten der Werkstatt ein Hauptschalter verbaut, warum dann nicht auch diesen mit FabAccess steuern?
- manche Geräte lassen sich auch direkt über [PoE](#) bestromen!
- jedes Gerät mehr, was an einer Steckdose sitzt, ist ein potentieller Kandidat für eine DGUV-3 Prüfung

### **Tipps für Drehstrommaschinen (400 V):**

- ganz pauschal: ausreichend Platz für weitere Hutschienenmodule in den Sicherungsschränken, z.B. für Shellys zum Schalten und Strommessen lassen
- Bei 400 V Maschinen ist die Strommessung generell eine Herausforderung. Wer z.B. über Shelly und Schütz arbeitet, braucht ein zusätzliches Gerät für die Strommessung. Das sorgt möglicherweise für sehr viele Geräte im WLAN.
- Steckdosen einzeln zum Sicherungskasten verlegen. So kann z.B. über einen [12-fach Schaltaktor mit Strommessung von KNX](#) mit einem einzigen LAN Anschluss ein großer Pulk Maschinen abgedeckt werden

## Welche Nachteile hat die Nutzung von FabAccess?

Nachteile sind zum einen der Aufwand der Installation und Konfiguration der Software (Serverdienst), das Ausrollen beim jeweiligen Nutzer (Clients auf verschiedensten Geräten) und das Installieren von schaltbarer Hardware an den zu steuernden Geräten (Tasmota, Shelly, FabReader, etc.).

Ist der FabAccess Server einmal in Verwendung und Geräte fest damit verbunden, dann ist der Server für den Werkstattbetrieb eine essenzielle Komponente, die nicht mehr wegzudenken ist. Entsprechend stabil und sicher (Backups!) sollte die Infrastruktur dann ausgelegt werden, damit es eine möglichst gute Benutzererfahrung gibt. Hier bedarf es etwas Konzept. Wir empfehlen deshalb unter anderem den [Getting Started](#) Guide.

## Könnt ihr FabAccess für mich hosten?

Die kurze Antwort ist **nein**. Es ist zwar technisch möglich für verschiedene Spaces jeweils eine FabAccess Instanz zu hosten. Allerdings gibt es hier verschiedene Fallstricke, die das unpraktikabel machen:

- jeder Space wäre vom Hoster (z.B. uns) abhängig: gibt es hier Probleme, gibt es wahrscheinlich bei allen Spaces gleichzeitig Probleme. Wenn FabAccess ersteinmal in der Werkstatt läuft, dann sind die Benutzer darauf angewiesen.
- externes Hosting bedeutet nicht immer, aber oft: Der Nutzer ist davon abhängig, wann der Hoster entscheidet, dass es Zeit für eine Downtime oder ein Update ist
- Datenschutz! Was wollen wir mit euren Daten? Was wollen andere mit euren Daten?
- Es bedürfte einem SSH/(S)FTP Zugang oder Web Interface für jeden Nutzer, um die BFFH Config Files hochzuladen. Spätestens mit eingebundenen Aktoren für die Hardware wäre das ein schwierigeres Unterfangen
- es wäre vermutlich unbezahlbarer, unterbezahlter oder unbezahlter Aufwand
- externes Hosting über weite Strecken kann anfällig sein für Latenzen und Unterbrechungen.

## Fragen zum Server

## Kann man mehrere BFFH Instanzen auf einem Server laufen lassen?

Ja das ist problemlos möglich! Es gibt eine Binary, die mehrfach ausgeführt werden kann. Es ist hier nur wichtig, dass dann jeweils individuelle Einstellungen getroffen werden. Das betrifft die Speicherorte für die Benutzerdatenbank ([users.toml](#)), die [Hauptkonfiguration bffh.dhall](#) (und etwaige Tochterdateien), sowie den [Audit Log](#). Das lässt sich jedoch relativ simpel mit systemd Service Files und entsprechenden Dateinamen lösen.

Eine Alternative für Multi-Instance Hosting bietet auch die Verwendung von Containerdiensten, wie z.B. Docker oder Podman.

## 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.

## Fragen bei der täglichen Nutzung (Client)

### 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 Zielservice bekommen. Das ist eine Aufgabe für den lokalen Netzwerkverwalter des jeweiligen Spaces.

### 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.

### **Wie geht man mit Mehrfachanmeldungen um, z.B. wenn jemand die Berechtigung für drei Ressourcen 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 (z.B. von verschiedenen Geräten aus, z.B. Smartphones oder [FabReadern](#)). 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?*)

### **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.

## 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 [FabFireCard](#) 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.

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

Eine Maschine zu benutzen heißt in der Regel ihr Strom zuzuführen, also sie einzuschalten. Ein Schließfach hingegen schaltet man nicht ein. Es wird geöffnet und wieder geschlossen. Eine Maschine kann also aktiv genutzt oder geblockt werden. Ein Schließfach hingegen kann nur geblockt werden werden.

# Fragen zu NFC / Smartcards (FabFireCards) / FabReader

## Sind bereits existente Studentenausweise für FabAccess erweiterbar?

Das kommt auf die Beschaffenheit der Karte an. Generell benötigt wird Mifare DESFire EV2. Falls der Ausweis diesen Standard unterstützt, das herzlichen Glückwunsch. Nun ist die Frage, ob noch freie Slots auf der Karte bespeicherbar sind und ob die Hochschule außerdem unsere Anwendung zulässt.

Im Übrigen: Eine Smartcard kann auch mehrere NFC Tags parallel enthalten. Diese bezeichnen sich z.B. als Dual Frequency Smartcards oder Hybridkarten und sehen intern z.B. so aus:



### Warum verwendet FabAccess MIFARE DESFire Karten für die Benutzerauthentifizierung?

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!

### Welche Karten sind mit FabAccess kompatibel?

Für das Einloggen von Benutzern am System: Ausschließlich 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! Weitere Details siehe [Funktionsprinzip / Grundlagen](#) und [FabFire Tools](#).

Für das Scannen von Ressourcen (z.B. Maschinen) per NFC: zum Beispiel Mifare NTags. Das Schema ist in [Ressourcen per NTag scannen](#) beschrieben.

# Fragen zur API

## Warum verwenden wir Cap'n Proto und keine REST Schnittstelle?

Der einfachste Punkt, weswegen REST eher ungeeignet ist, ist der, dass unsere Verbindung bidirektional ist, also beide Seiten zu jedem Zeitpunkt Daten an die jeweils andere schreiben können. Das würde WebRTC prinzipiell problemlos ermöglichen (Datenaustausch z.B. per JSON), aber eine einfache TCP- oder QUIC-Verbindung ebenso und mit wesentlich weniger Setupaufwand. Sobald es um WebRTC geht, sind in einem webbasierten Client große Mengen angepasster JavaScript-Code nötig. Ist das der Fall, könnten diese TCP-/QUIC-Verbindungen über Websockets tunneln und die bestehende Cap'n Proto API ansprechen.

# Fragen zur Community und Partnern

## 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.

## Was muss ich beachten, wenn wir mit Werkstätten föderieren wollen?

Das allgemeine Konzept der [Föderation](#) ist hier erklärt.

## Wo finde ich Föderationspartner? Wo können wir uns eintragen?

Föderationspartner für FabAccess findest du unter anderem auf der Seite des Verbunds Offener Werkstätten oder auch hier auf unserer Webseite. Siehe [hier](#).