

Wer ist FabAccess?

- Eine Community aus mehreren Mitgliedern des Verbund Offener Werkstätten e.V. und dem FAB:UNlverse
- Idee entstand zw. 2018 und 2019
- Projekt wurde ursprünglich von Studenten der Beuth Hochschule für Technik (seit 2021 BHT Berliner Hochschule für Technik) betreut

Wer ist FabAccess?

- das Projekt wurde in verschiedenen Abschnitten teilweise gefördert, u.a. Prototype Fund, Fab City / Interfacer, DSEE und anstiftung
- wurde von Anfang an auch in der VOW-Gemeinschaft für notwendig + wünschenswert empfunden
- in Zeit von 2019 - 2024 hauptsächlich durch das studentische Entwicklerteam voller Ehrgeiz betreut (später als RLKM UG geführt)

Wer ist FabAccess?

- RLKM UG löst sich 2024 auf. FabInfra wird nahezu vollständig an die Community überlassen → **Neufindungsphase Q3/2024**
- Projekt geht weiter mit anderen Köpfen, restlicher DSEE-Förderung und Ehrenamt, sowie verschiedenen Werkstätten quer in Deutschland verteilt

FabAccess ?

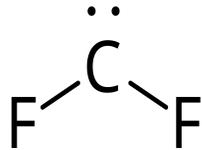
- Was ist es, was ist es nicht?
- Wozu braucht's das?
- Welche Vor- und Nachteile?

FabInfra

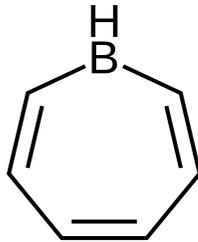
- hinter dem Gebrauchsbegriff **FabAccess** verbirgt sich vielmehr der Infrastrukturbegriff **FabInfra** = Werkstätten + FabAccess + FabHardware + Addons

Naming - die Chemie stimmt

- Server: "Difluoroborane" (kurz BFFH)



- Client: Borepin aka "FabAccess"



Licensing / Use

Ist komplett Free Open Source Software / Hardware

- Difluoroborane - GNU GPL v3
- Borepin - GNU GPL v3
- FabAccess API - MIT License

Kein Fertig-Hosting für Konsumenten

Jeder betreibt seine eigene Instanz self-hosted/on-premise - schon allein aus Datenschutzgründen

Sinn von FabAccess ?

Komplizierte, teure, seltene und/oder gefährliche Maschinen vor dem Benutzer bewahren ... und vice versa. Denn Vorsicht ist besser als Sichtverlust.

Kann Probleme lösen wie zum Beispiel:



www.epiloglaser.com

oriental

ESCH
(500

epilog

Häufige Probleme in Werkstätten

- Energie wird vergeudet
- Verbrauchsmaterial wird zum Inflationbegriff
- Maschinen gehen wie durch Zauberhand kaputt, aber niemand war's
- Die Kümmerner/Besitzer der einzelnen Maschinen (Ressourcen) werden nicht gefragt
- Maschinen, die aktuell defekt sind und gewartet werden müssen: "ach! geht schon noch das eine mal"

Der Satz in Klärung beginnt dann in der Regel mit "ich dachte ..."

Eine Software kann diese sozialen Probleme nicht lösen. Aber sie kann zumindest dabei unterstützen! Und sie kann Abläufe schaffen

Konzept

Wir bauen ein System, was sich dem Nutzer (Werkstattleiter/Administrator) anpasst
und nicht anders herum (Theorie/These)

Wir schalten den Strom der Maschinen per FabAccess.

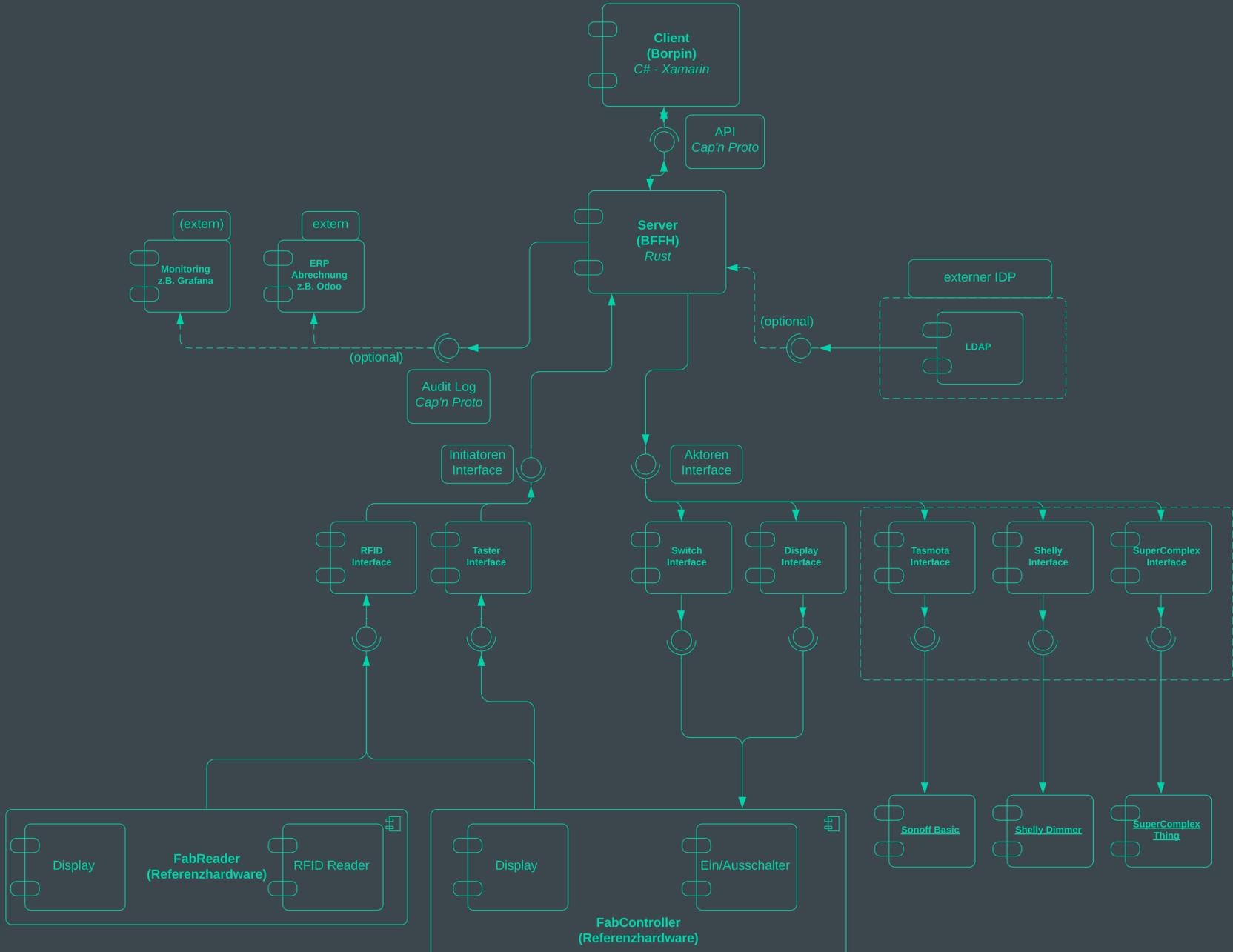
Dafür gibt es Definitionen:

Benutzer + Rollen + Berechtigungen + Ressourcen + Mappings (Aktoren, Initiatoren)

Schema

FabAccess: Struktur

TheJoKILa | May 27, 2021



Notes

FabAccess ist ...

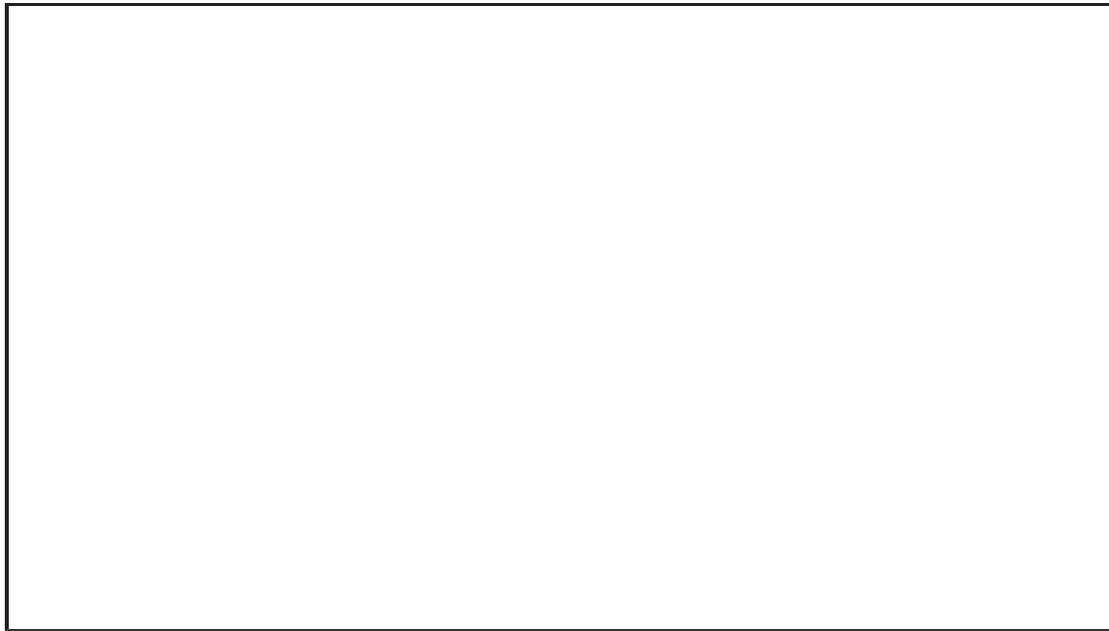
- generisch gebaut. Möglichst universell einsetzbar
- prozesstechnisch an Werkstattabläufe angelehnt (Unterschied zu “klassischen” Heimautomatisierungssystemen wie openHAB oder Home Assistant)
- hat Anbindungen:
 - API (Cap’n Proto)
 - kann z.B. mit Python-Plugins gefüttert werden, die MQTT und Co. auswerten
 - Audit Log (z.B. Abrechnung, DuDu-Schimpf, Selbstzerstörung)
 - Monitoring (Prometheus, Grafana und Co.)

Config

Per DHALL und TOML - maximal generisch und flexibel! Aber auch gewöhnungsbedürftig für den Anfang

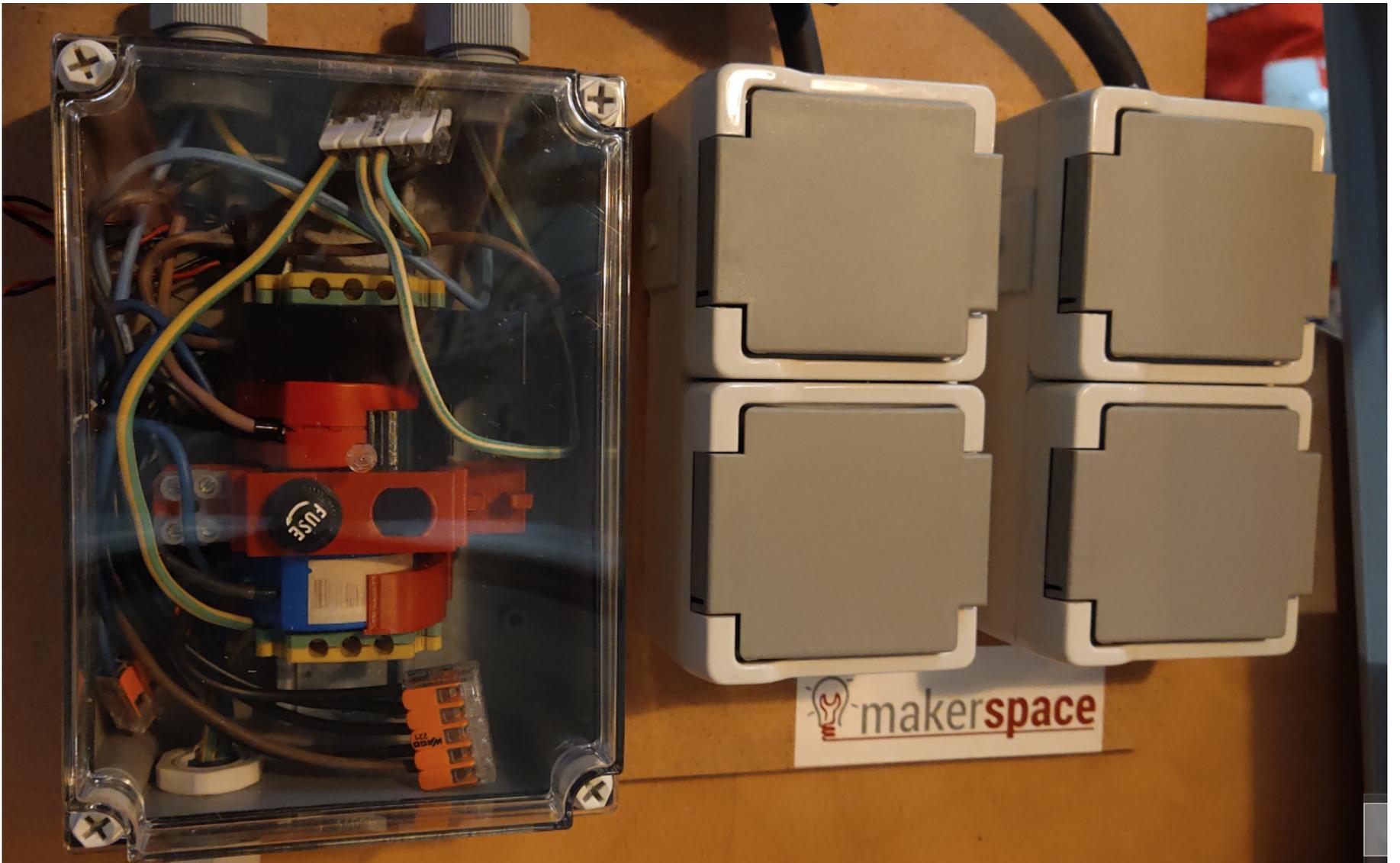
```
1 {
2  spacename = "fabaccess.sample.space",
3  instanceurl = "https://fabaccess.sample.space",
4  listens = [{address = "127.0.0.1"}],
5  certfile = "/etc/bffh/bffh-selfsigned.cert",
6  keyfile = "/etc/bffh/bffh-selfsigned.key",
7  mqtt_url = "mqtt://127.0.0.1:1883",
8  db_path = "/var/lib/bffh/bffh.db",
9  auditlog_path = "/var/log/bffh-audit.json",
10 roles = {=},
11 machines = {=},
12 actors = {=},
13 actor_connections = [] : List { machine : Text, initiator : Text },
14 initiators = {=},
15 init_connections = [] : List { machine : Text, initiator : Text }
16 }
```

Hardware - Beispiel Türschloss



Hardware - Primary-Secondary von Joris Bijkerk / Makerspace Bocholt

realisiert mit Shelly Modulen



Hardware - Sandstrahlkabine mit "FabCounter"

auch von Joris Bijkerk / Makerspace Bocholt



000

HBM
Machines

Maschine
Sandstrahler (groß)



15cl / 1 Minuten

Erweisung: eingeklebt

Betrieb:

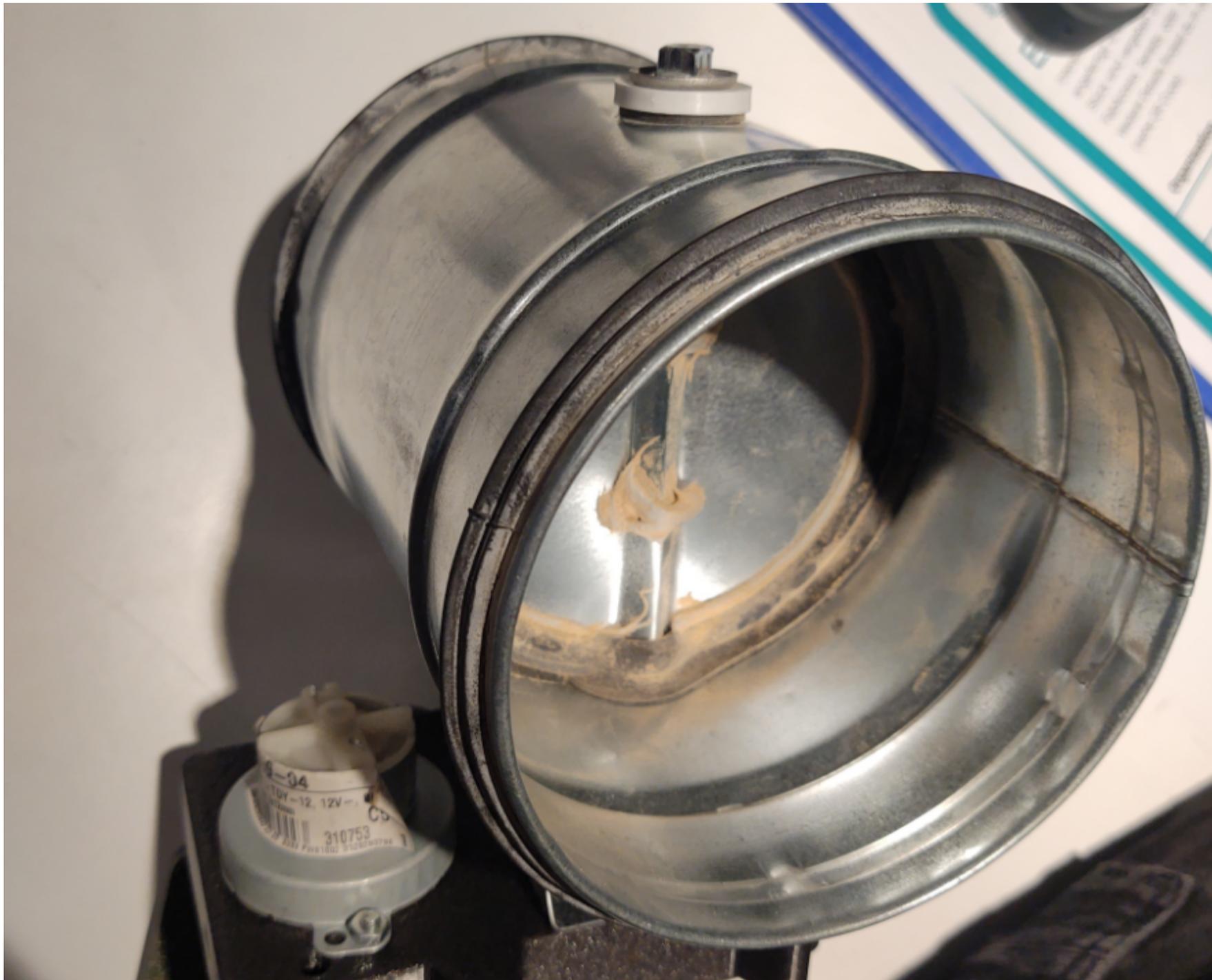
WICHTIG - WICHTIG

Bitte lesen Sie die Bedienungsanleitung sorgfältig durch, bevor Sie die Maschine in Betrieb nehmen. Die Maschine ist für den Einsatz in geschlossenen Räumen geeignet. Die Maschine ist für den Einsatz in geschlossenen Räumen geeignet. Die Maschine ist für den Einsatz in geschlossenen Räumen geeignet.



Hardware - Lüfterklappensteuerung für Bandschleifer

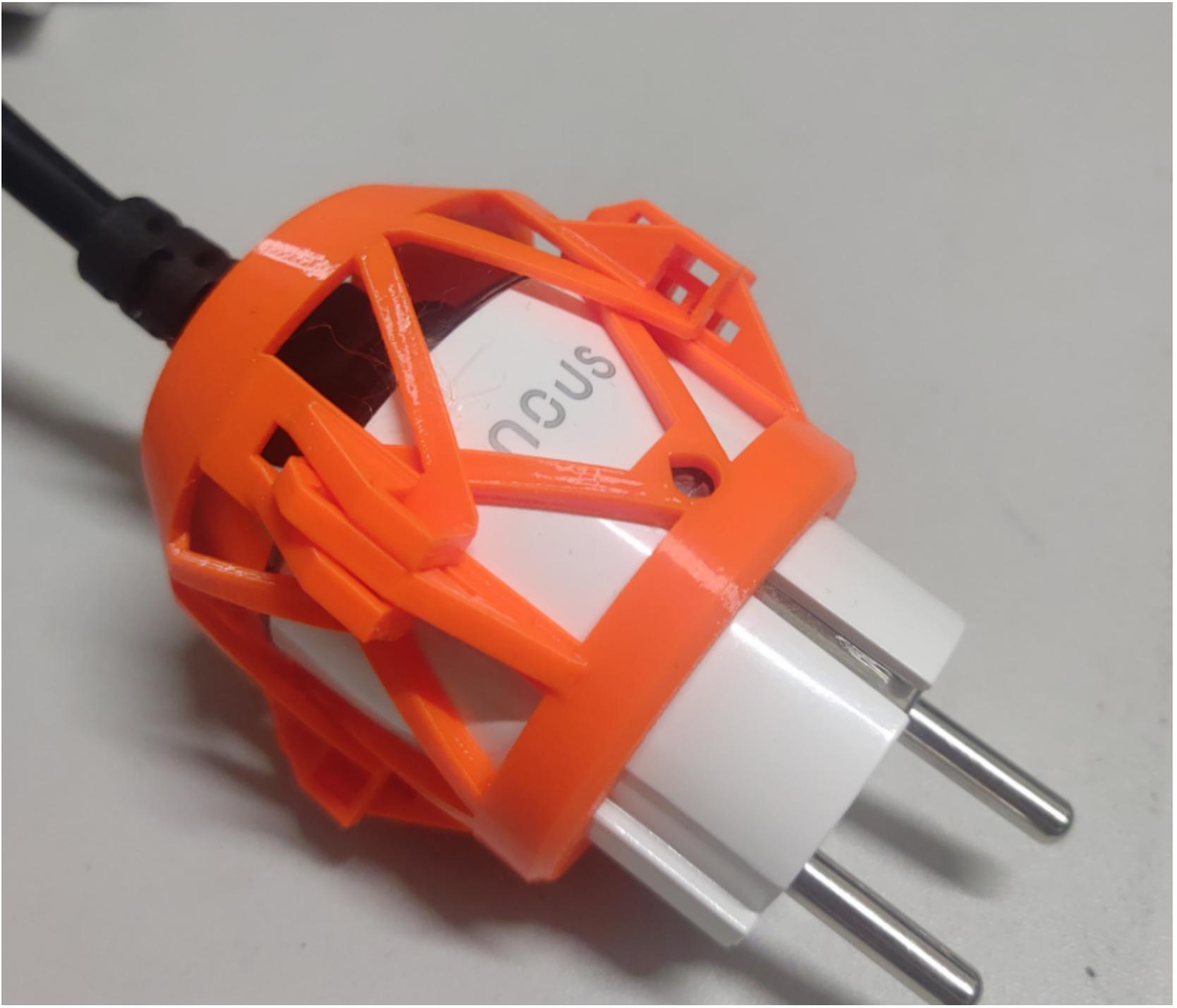
... und das auch von Joris Bijkerk / Makerspace Bocholt ^



Hardware - “FabAccess” Kabel

Manipulationsschutz / Aufmerksamkeitsgenerator

Nous A1T Safebox von Michael Prange / Makerspace Gütersloh



Client - Beispielansicht GUI

Scan QR-Code		
Schlüssel		
Schließfach Schlüssel	In Use	→
Tools		
Handbohrmaschine	In Use	→
Poliermaschine	Free	→
Tauchsäge	Free	→
Winkelschleifer	Free	→
Werkstatt		
Bohrmaschine B15	Free	→
Bohrmaschine Unimax 1	In Use	→
Drehbank MLZ48	Free	→
Metall Kappsäge	Free	→
Schleifbock SL 175	Free	→

Client - Features

- multiple Verbindungen zu div. Instanzen
- kann Barcodes scannen
- integrierte Nutzerverwaltung
- Aktuell Funktionen integriert für Benutzen (USE), Freigeben (GIVEBACK), Blockieren (BLOCK) und Deaktivieren (DISABLE)

Kurze Demo



Sticker Generator vom Sternenlabor

<https://sticker.fab-access.org>


STERNENLABOR
Makerspace für Plauen



RD6012 DC POWER SUPPLY



Config Generator vom MakerSpace Leipzig

<https://github.com/elem74/fabaccess-config-generator>

Grundlagen



Startseite

Einführung

[Grundlagen](#)

Arbeiten mit der CSV

Skript-Nutzung

Referenz

Bereiche & Rollen

CSV-Datei

actors.ini

settings.ini

Inhaltsverzeichnis

[Ordnungskonzept](#)

[Rollenkonzept](#)

[Administrator](#)

[Benutzer](#)

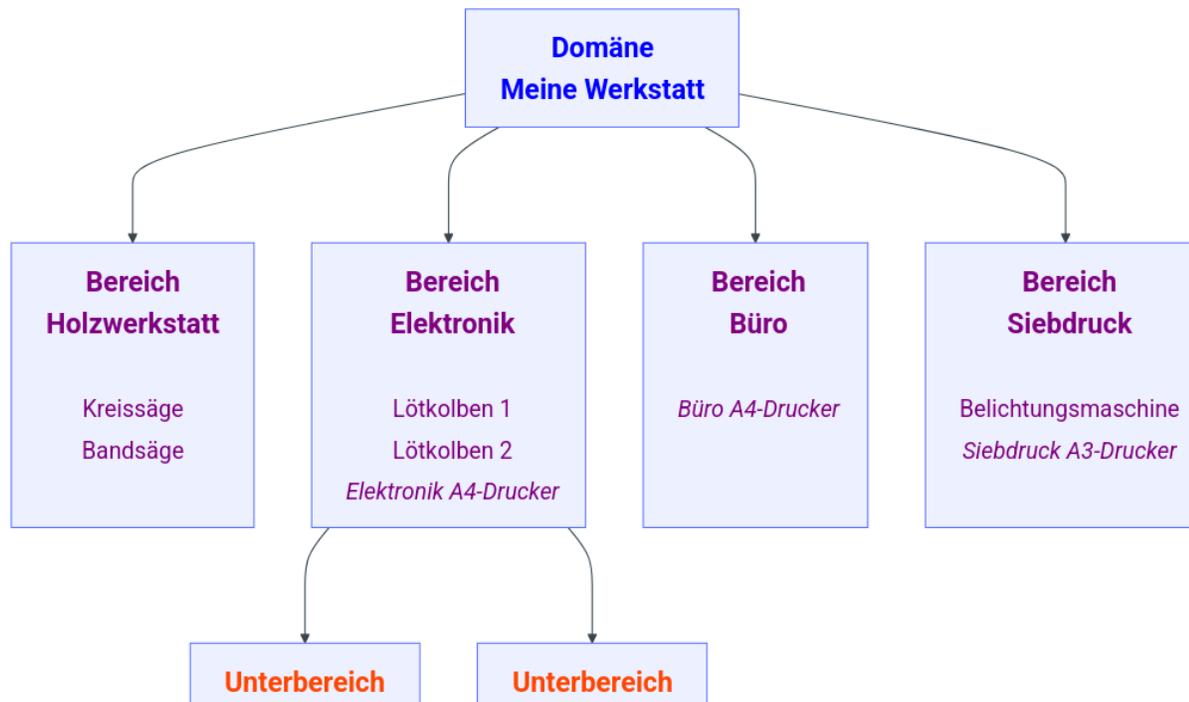
[Manager](#)

[Alternativrolle](#)

[Erweiterte Beispielwerkstatt](#)

Erweiterte Beispielwerkstatt

Nachdem wir die verschiedenen Rollen kennengelernt haben, erweitern wir unsere Beispielwerkstatt. Unsere Beispielwerkstatt sieht nun so aus:



Wo geht die Reise hin?

... das ist alles Community-gesteuert!

Aktuelle und alte ToDo-Themen sind u.a.:

- neues Server-Release mit Bugfixes und aktueller Rust-Version
- neuer Client-Reupload im PlayStore + gefixte Versionen in MS Store, iOS Store
- Web Client + REST API
- Client-Features wie Ausleihen (LEND) und Übertragen (TRANSFER)
- LDAP / OIDC
- Etablierung in weiteren Werkstätten
- Debian-Paket
- Doku aufräumen, erweitern und leere Phrasen herausoperieren
- Beispiele für div. Setups (Software, Hardware, Netzwerk)
- Vernetzung und kontinuierliche Vernetzung der User Group(s)
- echte Federation
- FabReader
- GitLab aufräumen
- ...

Kompatibilität



Mitmachen

Code:

<https://gitlab.com/fabinfra>

Dokumentation:

<https://docs.fab-access.org>

Downloads:

<https://fab-access.org/download>

Community (Matrix, Telegram & Co.):

<https://fab-access.org/join>



**35 SERVICES E. V. -
OFFENE WERKSTATT**

Lehrter Str. 30, 10557 Berlin
<https://35services.de>



**FABLAB CHEMNITZ (PROJEKT
DES STADTFABRIKANTEN E.V.)**

Schüffnerstraße 1, 09130 Chemnitz
<https://stadtfabrikanten.org>



MAKERSPACE BOCHOLT

Westend 31, 46399 Bocholt
<http://www.makerspace-bocholt.de>



MAKERSPACE GÜTERSLOH

Bogenstraße 1-8, 33330 Gütersloh
<https://makerspace-gt.de>



Verstärkung gesucht

Das Projekt lebt von Werkstätten, die FabAccess nutzbringend finden und dafür beisteuern, z.B.: Fehlermeldungen, Code-Verbesserungen, Feature-Wünsche, Plugins, Hardware-Projekte und mehr.

Server:

- Cap'n Proto
- Rust
- Python

Client:

- Xamarin
- C#
- Mono
- AvaloniaUI

Aktuell ist das Projekt nicht finanziert und lebt von Freiwilligen

Ansprechpartner

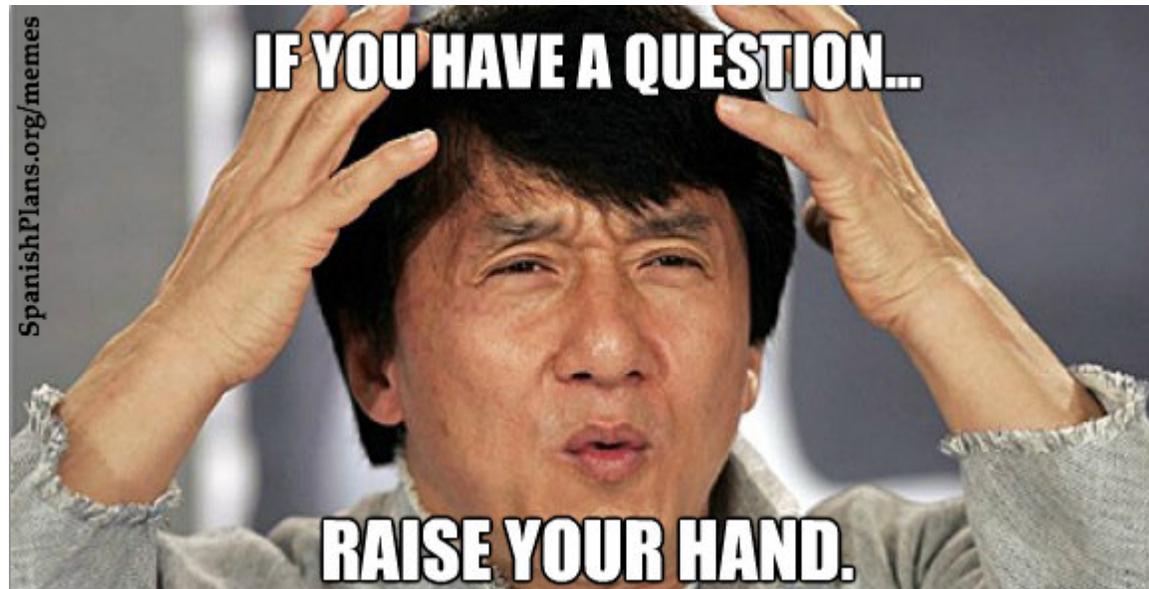
... gibt es ein paar verschiedene. Siehe Community!

(in Chemnitz)

Mario / Stadtfabrikanten e.V.
mario.voigt@stadtfabrikanten.org



AbgeFAQt?



Bildquelle: <https://spanishplans.org/memes>