Client für Linux - mit Waydroid installieren

Waydroid ist ein unter GNU GPL v3 lizensierter und einer von vielen gängigen Emulatoren für Android auf Linux-Systemen. Wir nutzen es, um **FabAccess** für Linux verfügbar zu machen (alternativ zum GTK Mono Client Build). Waydroid verwendet einen LXC-Unterbau (Container, ähnlich Docker), um zu spawnen. Wir berufen uns grundlegend auf die offizielle Dokumentation unter https://docs.waydro.id/usage/install-on-desktops.

Intro

Es gibt zu viele Linux-Distros, um auf jede einzugehen, was die Installation angeht. Auf dieser Seite finden sich Details zum Setup auf Kubuntu (analog: Debian, Ubuntu) und Fedora (analog: CentOS).

Diese Dokumentation beruht auf folgenden Beispielsystemen:

- OS: Kubuntu 24.04 und Fedora 40, jeweils mit aktiviertem Wayland
- Waydroid 1.5.0 mit LineageOS 18.1 (Android 11)

Erfordernisse (Requirements)

Die benötigten Systemeigenschaften sind in https://wiki.archlinux.org/title/Waydroid#Wayland_session_manager beschrieben.

- CPU/Architektur: https://developer.android.com/ndk/guides/abis?hl=de#sa
- GPU: Intel oder AMD; **nicht Nvidia**
- Display Session: Wayland; **nicht X11**
- ca. 1 GB Speicherplatz (für die Images und libhoudini)
- ca. 400-500MB RAM

> Es wurde berichtet, dass Waydroid auf auch Raspberry Pi's laufen kann!

One-Liner: Setup als fertiges Bash Script

Folgende einzelne Zeile installiert das aktuellste FabAccess Borepin Release automatisch und vollständig auf Linux-Systemen:

```
curl -L -s https://y.fab-access.org/waydroid | bash
```

Wer dem Kurzlink https://y.fab-access.org/waydroid nicht traut, der kann folgende voll ausgeschriebene Zeile verwenden:

```
curl -s https://gitlab.com/fabinfra/fabaccess/borepin/-/raw/main/waydroid-installer.sh | bash
```

Das grundlegende Setup

Wer das Setup manuell durchführen möchte, kann auch alle nachfolgenden Schritte verfolgen.

Kubuntu:

```
sudo apt update && sudo apt upgrade
sudo apt install curl ca-certificates -y
curl -s https://repo.waydro.id | sudo bash
sudo apt install waydroid -y

# Die installierte Waydroid Version checken
dpkg -l | grep waydroid

#Init ausführen (Images installieren)
# Folgendes Images werden dabei automatisch von
https://sourceforge.net/projects/waydroid/files/images/system/lineage/waydroid_x86_64/
geladen:
# - LineageOS 18.1 (20241102 VANILLA) - das Image ist ca. 800 MB groß
# - LineageOS 18.1 (20241102 MAINLINE) - das Image ist ca. 180 MB groß
sudo waydroid init
```

Fedora

```
sudo dnf update
curl -s https://repo.waydro.id | sudo bash
sudo dnf install waydroid -y

# Die installierte Waydroid Version checken
dnf list installed | grep waydroid
```

```
#Init ausführen (Images installieren)
# Folgendes Images werden dabei automatisch von
https://sourceforge.net/projects/waydroid/files/images/system/lineage/waydroid_x86_64/
geladen:
# - LineageOS 18.1 (20241102 VANILLA) - das Image ist ca. 800 MB groß
# - LineageOS 18.1 (20241102 MAINLINE) - das Image ist ca. 180 MB groß
sudo waydroid init --system_channel https://ota.waydro.id/system --vendor_channel
https://ota.waydro.id/vendor --rom_type lineage --system_type VANILLA
```

Fehlerbehebung (Fedora)

Falls das bisherige Setup mit Fehlern scheitert, müssen ggf. die SELinux Regeln für `systemd` und `waydroid` angepasst werden:

```
ausearch -c 'systemctl' --raw | audit2allow -M my-systemctl
semodule -X 300 -i my-systemctl.pp

ausearch -c 'waydroid' --raw | audit2allow -M my-waydroid
semodule -X 300 -i my-waydroid.pp
```

Temporär - d.h. nur Testweise - kann auch mit `setentforce 0` gearbeitet werden.

Etwaige Anpassungen sind außerdem an der Config von Nöten, falls "Failed to load config for waydroid" auftritt (siehe auch Issue 493 und 652):

```
sudo vim /var/lib/waydroid/lxc/waydroid/config
```

Auskommentieren:

```
#lxc.apparmor.profile = unconfined
```

Service und Logs

Waydroid wird automatisch als Service installiert und auch gestartet. Wir können den Service starten, stoppen und neustarten, und darüberhinaus Logs auslesen:

```
sudo systemctl status waydroid-container.service
sudo systemctl start waydroid-container.service
sudo systemctl stop waydroid-container.service
```

```
sudo systemctl restart waydroid-container.service

# Journal und Log File:
sudo journalctl -f -u waydroid-container.service
sudo less /var/lib/waydroid/waydroid.log

#oder kurz:
waydroid log

#oder über LXc:
sudo lxc-info -P /var/lib/waydroid/lxc -n waydroid -sH
```

Python pip und venv installieren

Kubuntu:

```
sudo apt install python3-pip python3-venv
```

Fedora:

```
sudo dnf install python3-pip python3-venv
```

Zwischenablage (Clipboard) reparieren

Bei der Installation klappt u.U. die geteilte Zwischenablage nicht. In diesem Fall benötigt es die folgenden Pakete.

Kubuntu:

```
sudo dnf install wl-clipboard
```

Fedora:

```
sudo apt install wl-clipboard
```

und dann:

```
sudo pip install pyclip --break-system-packages #unsauberer Trick. Hat jemand eine bessere Lösung?
```

Session starten

```
# als normaler Nutzer
waydroid session start

# oder im Hintergrund:
nohup waydroid session start &
```

Optionen konfigurieren

Wir stellen an ein paar Schrauben!

> Für das Ausführen muss eine Waydroid Session bereits laufen!

Multi-Window Option setzen

Aktiviert/deaktiviert die Fensterintegration mit dem Desktop

```
waydroid prop set persist.waydroid.multi_windows true
```

Auflösung des Fensters anpassen

Standardmäßig wird die Session im Vollbildmodus gestartet. Auf einem großen Monitor ist das ggf. scrècklich. Das lässt sich optional setzen. Siehe

https://github.com/waydroid/waydroid/issues/700

```
# Zum Zurücksetzen der Standardeinstellungen folgendes ausführen:
waydroid prop set persist.waydroid.width ""
waydroid prop set persist.waydroid.height ""

# Höhe aus Breite berechnen per Verhältnis (z.B. 16:9):
#width = 900 / 16 * 9 =~ 506
waydroid prop set persist.waydroid.height 900
waydroid prop set persist.waydroid.width 506
```

Netzwerk-Traffic zwischen Gast und Host erlauben

Falls nach der Installation Anwendungen in Waydroid keine Internetverbindung haben, fehlt möglicherweise eine Konfiguration in der Firewall und/oder in einem Waydroid-Script.

Siehe auch https://github.com/waydroid/waydroid/issues/143

Fix in Waydroid Script

```
sudo sed -i -E 's/=.\$\(command -v (nft|ip6?tables-legacy).*/=/g'
/usr/lib/waydroid/data/scripts/waydroid-net.sh
sudo systemctl restart waydroid-container.service
```

ARM-Builds auf x86-Systemen erlauben (libhoudini) und mehr

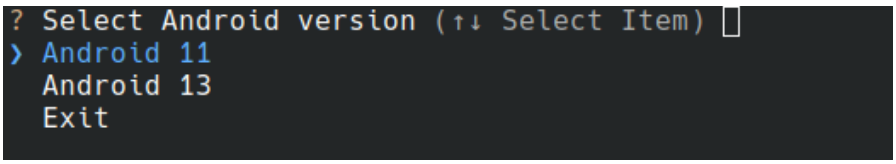
Normalerweise wird Waydroid vermutlich eher auf einem x86-System installiert werden. Android Applications (APKs) sind aber häufig nur für arm7/arm8 kompiliert. Dafür gibt es einen Workaround namens **libhoudini.** Dafür nutzen wir folgendes Script aus dem Projekt https://github.com/casualsnek/waydroid_script. Wir legen uns eine virtuelle Umgebung (venv) für das Projekt an und installieren die Requirements.

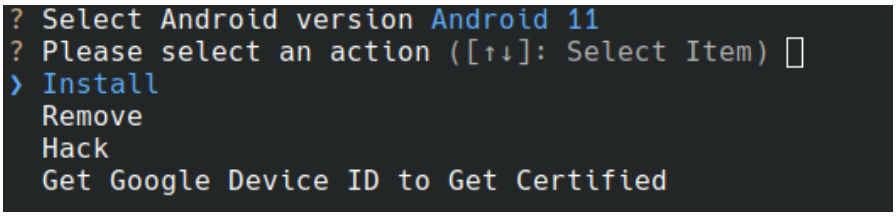
```
# als normaler Nutzer
cd ~
git clone https://github.com/casualsnek/waydroid_script
cd waydroid_script/

python3 -m venv venv
venv/bin/pip install --upgrade pip
venv/bin/pip install -r requirements.txt
```

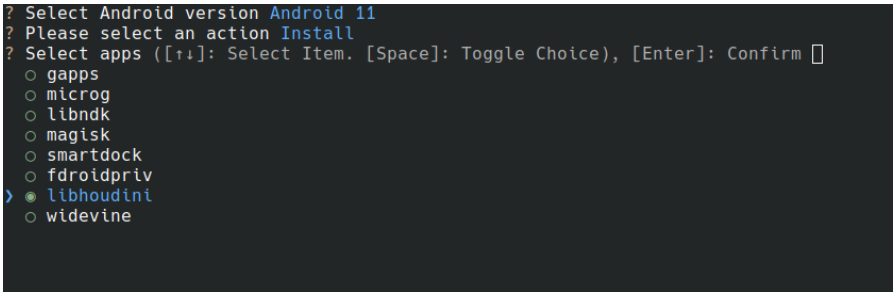
Danach führen wir aus und wählen **libhoudini** aus:

```
sudo venv/bin/python3 main.py
```





Mit der Leertaste können Pakete ab- oder zugewählt werden:



Danach startet die Installation. libhoudini ist ca. 100 MB groß.

Wir können das Menü auch überspringen und einen vollständigen CLI-Befehl eingeben:

```
sudo venv/bin/python3 main.py --android-version 11 install libhoudini
```

FabAccess Client (Borepin) installieren

Wir laden den Client als APK-Datei direkt herunter:

```
# als normaler Nutzer
cd ~
wget
https://gitlab.com/api/v4/projects/20862588/packages/generic/borepin/v0.3.11/org.fab_infra.fab
access-Signed.apk
waydroid app install org.fab_infra.fabaccess-Signed.apk

# wir prüfen, ob das geklappt hat:
waydroid app list | grep fabaccess

# der Output sollte sein:
packageName: org.fab_infra.fabaccess
```

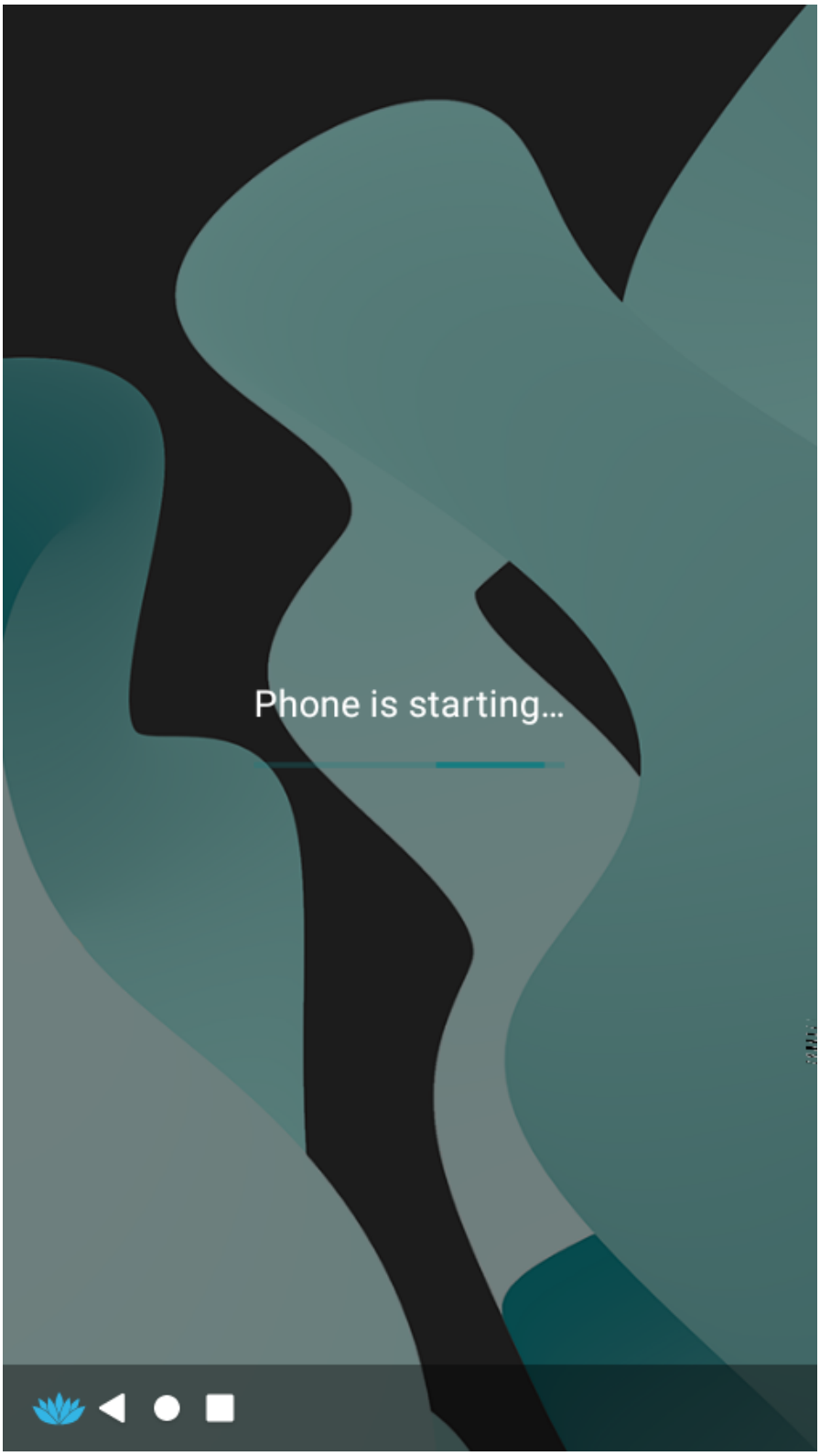
Siehe auch Downloads / Demo

Waydroid Client anzeigen und benutzen

Nachdem der Container gestartet ist und die Session läuft, können wir das User Interface anzeigen:

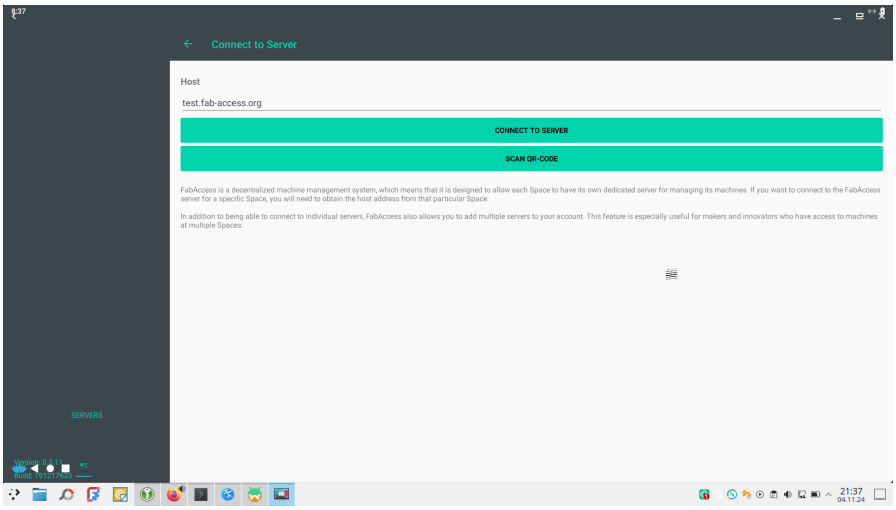
```
# als normaler Nutzer
waydroid show-full-ui
```

Die Standardoberfläche bei/nach dem Start:



Und letztlich Borepin in Waydroid starten:

> Der Kaltstart von Borepin bzw. Waydroid kann ca. 30-60 Sekunden dauern.



Waydroid in einem Fenster anzeigen

Unter Wayland hat das Client-Fenster leider keine Drag&Drop-Funktion. Wir können es in einen Rahmen einsperren, in dem wir das Tool "cage" verwenden

Kubuntu:

```
sudo apt install cage
```

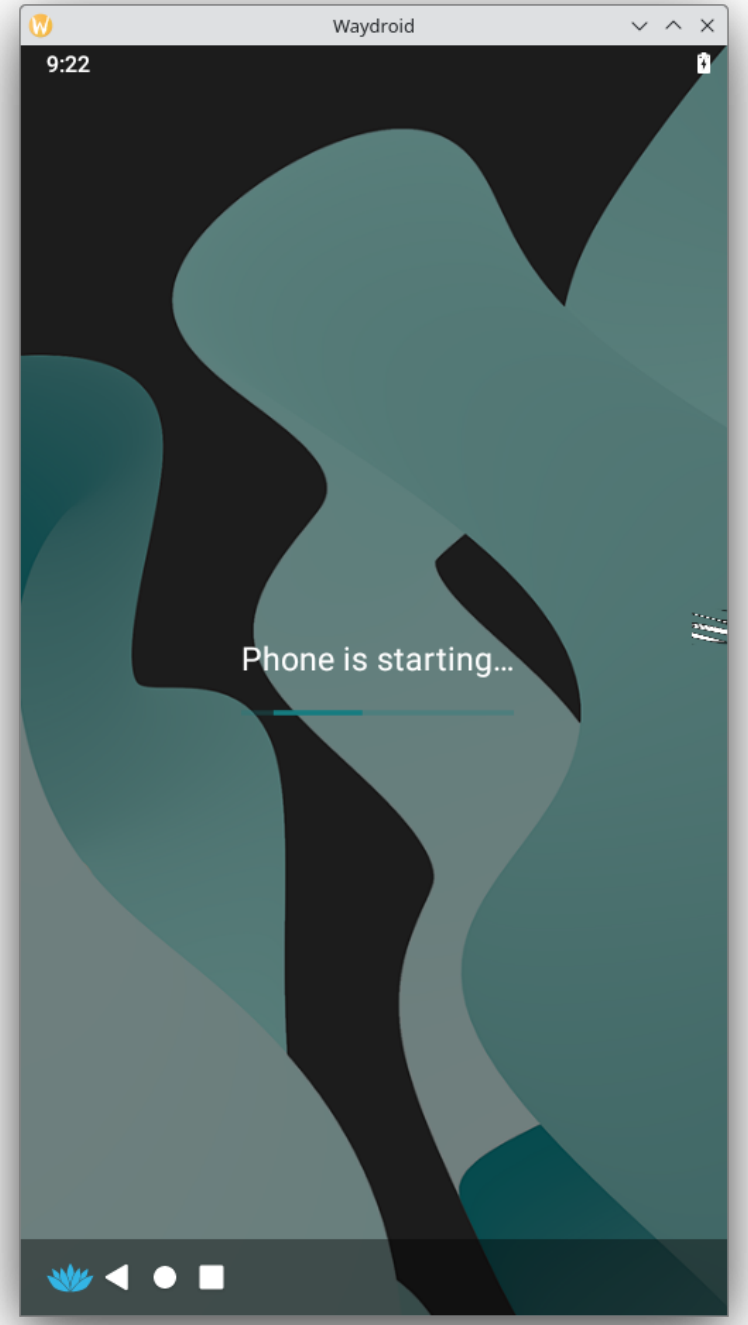
Fedora:

```
sudo dnf install cage
```

und dann Waydroid ins Fenster zwingen:

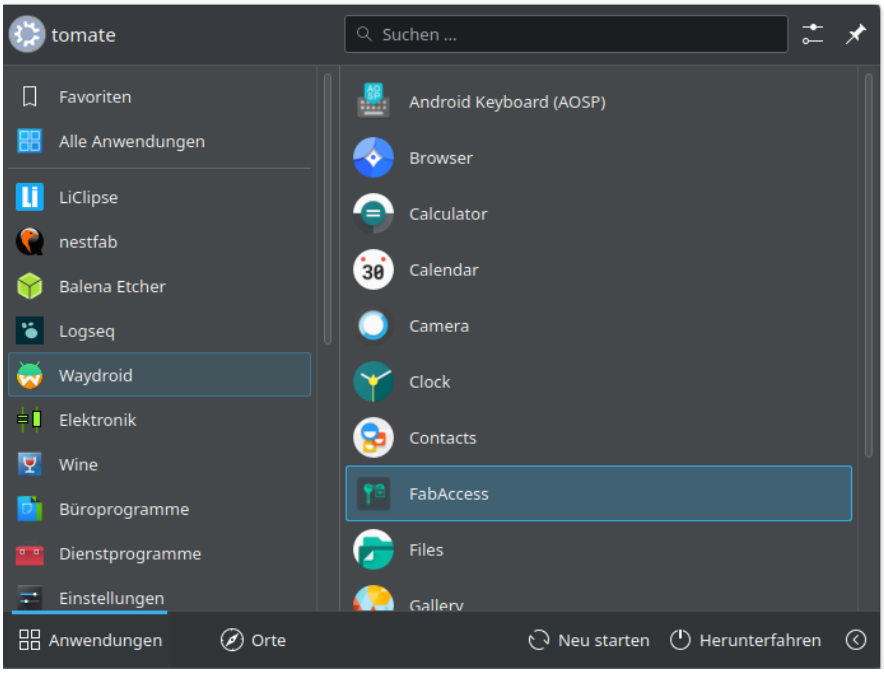
```
cage waydroid show-full-ui
```

Dadurch erhalten wir folgende Ansicht:

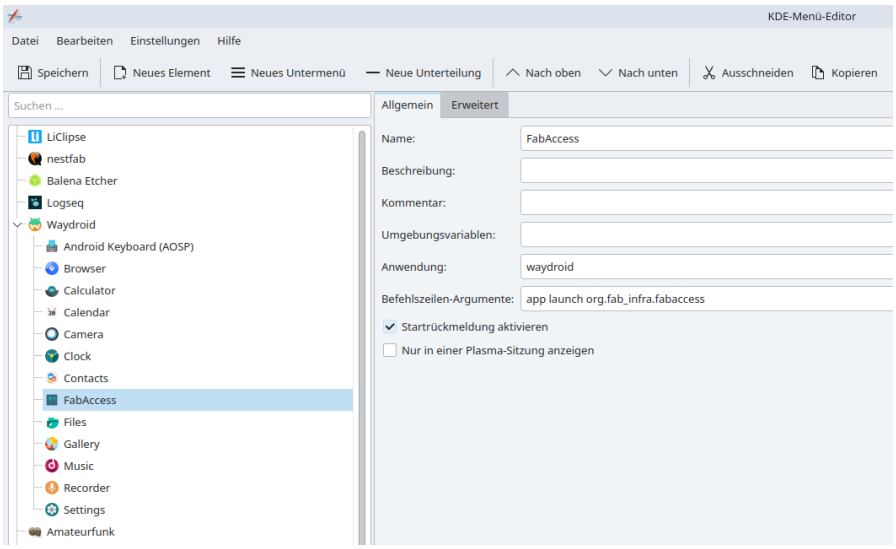


FabAccess Borepin aus dem Startmenü heraus starten

Waydroid wird standardmäßig in das Menü eingebunden:



Im Menü-Editor kann dieser Eintrag auch angesehen werden:



FabAccess Borepin per Kommandozeile starten

Das geht auch:

```
waydroid app launch org.fab_infra.fabaccess
```

oder mit Cage:

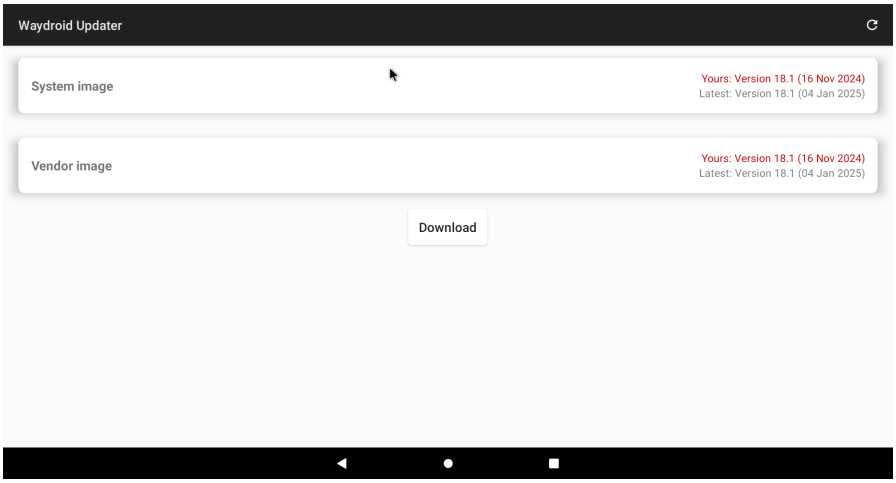
```
cage waydroid app launch org.fab_infra.fabaccess
```

Waydroid Upgrade

Waydroid bekommt regelmäßig Updates. Wir können mit folgendem Befehl aktualisieren:

```
sudo waydroid upgrade
```

Das im Hintergrund befindliche Base-Image für das Betriebssystem macht sich auch visuell bemerkbar und kann direkt per `Download` aktualisiert werden:



Drag & Drop Fenster

Durch Gedrückthalten der `Super`-Taste (in der Regel die Taste zwischen linkem `Ctrl` und `Alt`) und Anklicken einer beliebigigen Stelle des Wayland-Fensters kann dieses an die gewünschte Stelle verschoben werden.

Version #33
Erstellt: 2024-11-04 21:40:27 CET von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-03-22 22:16:45 CET von Mario Voigt (Stadtfabrikanten e.V.)