

# Client für Android - mit Waydroid auf Linux 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>.

## Basic Setup

Diese Dokumentation beruht auf folgendem Beispielsystem:

- Setup Datum: 04.11.2024
- OS: Kubuntu 24.04 mit aktiviertem Wayland
- Waydroid 1.4.3 mit LineageOS 18.1 (Android 11)

Das grundlegende Setup:

```
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
```

## 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
```

# Journal und Log File:

```
sudo journalctl -f -u waydroid-container.service
sudo less /var/lib/waydroid/waydroid.log
```

#oder kurz:

```
waydroid log
```

## Python pip und venv installieren

```
sudo apt 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

```
sudo apt install wl-clipboard
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. screcklich. 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
```

## 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](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 -r requirements.txt
```

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

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

```
? Select Android version (↑↓ Select Item) ☐
> Android 11
  Android 13
  Exit
```

```
? Select Android version Android 11
? Please select an action ([↑↓]: Select Item) ☐
> Install
  Remove
  Hack
  Get Google Device ID to Get Certified
```

Mit der Leertaste können Pakete ab- oder zugewählt werden:

```
? Select Android version Android 11
? Please select an action Install
? Select apps ([↑↓]: Select Item. [Space]: Toggle Choice), [Enter]: Confirm ☐
  ○ gapps
  ○ microg
  ○ libndk
  ○ magisk
  ○ smartdock
  ○ fdroidpriv
> ● libhoudini
  ○ widevine
```

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

## 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.fabaccess-
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
```

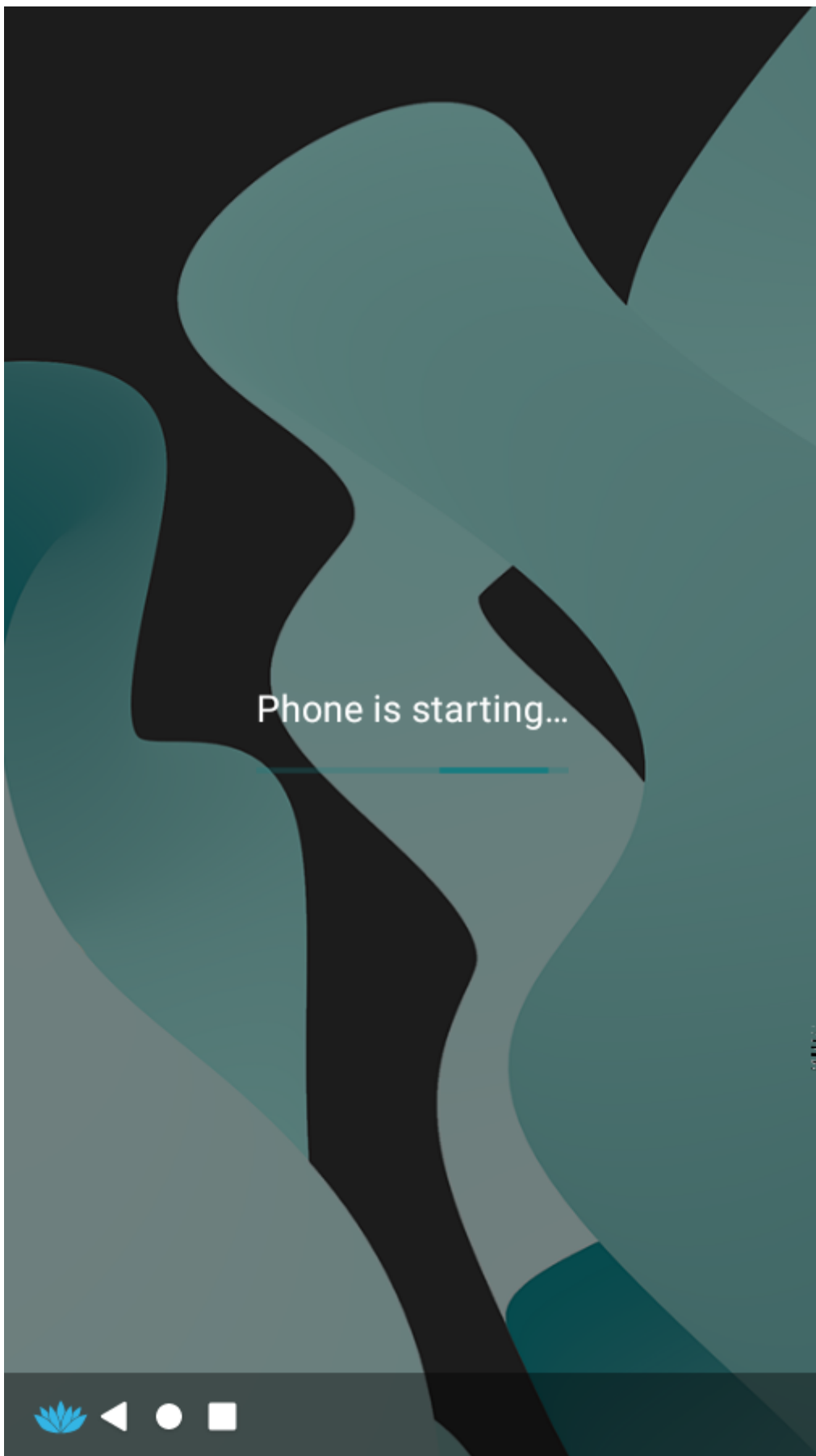
Siehe auch [Downloads / Demo](#)

## Waydroid Client anzeigen und benutzen

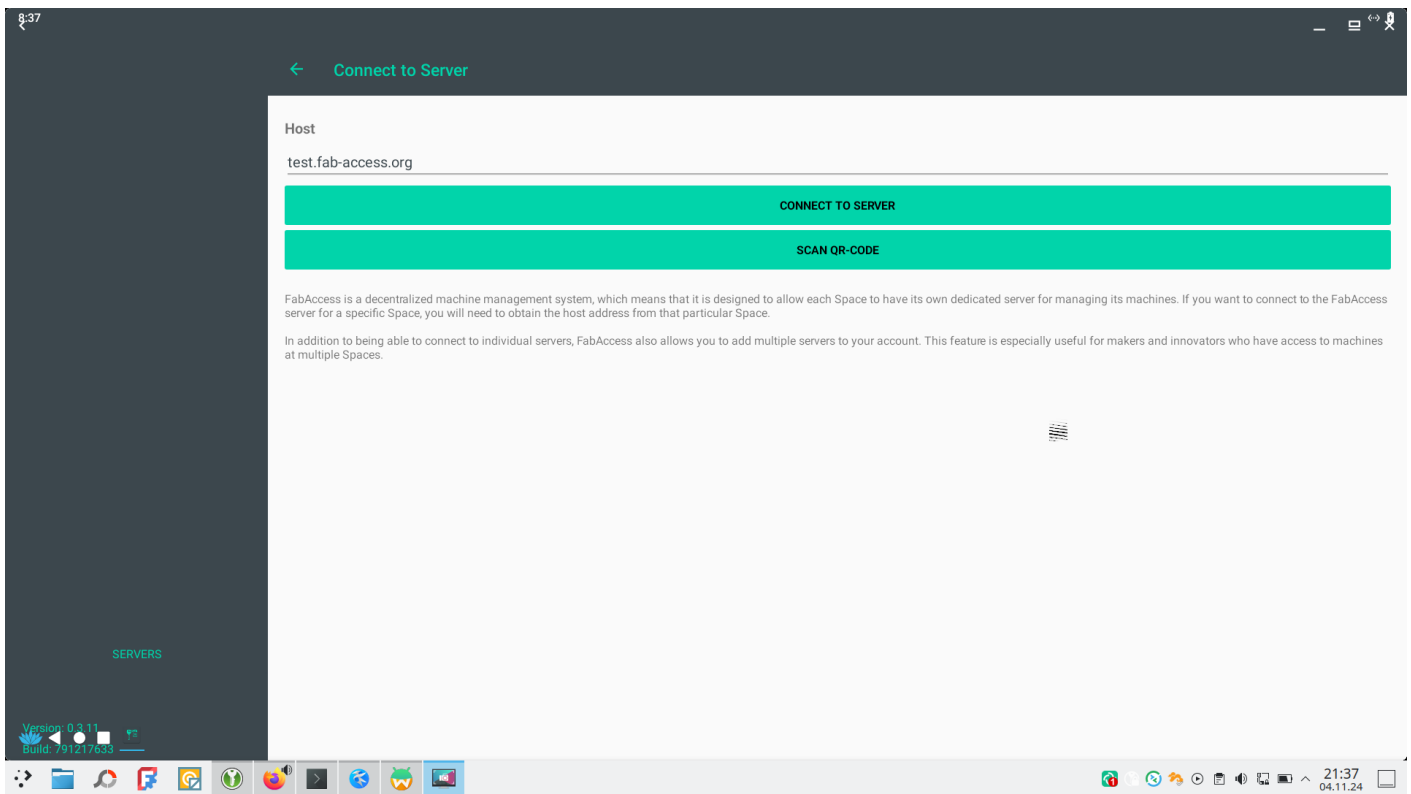
Nachdem der Container gestartet ist und die Session läuft, können wir das User Interface anzeigen:

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

Die Standardoberfläche bei/nach dem Start:



Und letztlich Borepin in Waydroid:



# Waydroid in einem Fenster anzeigen

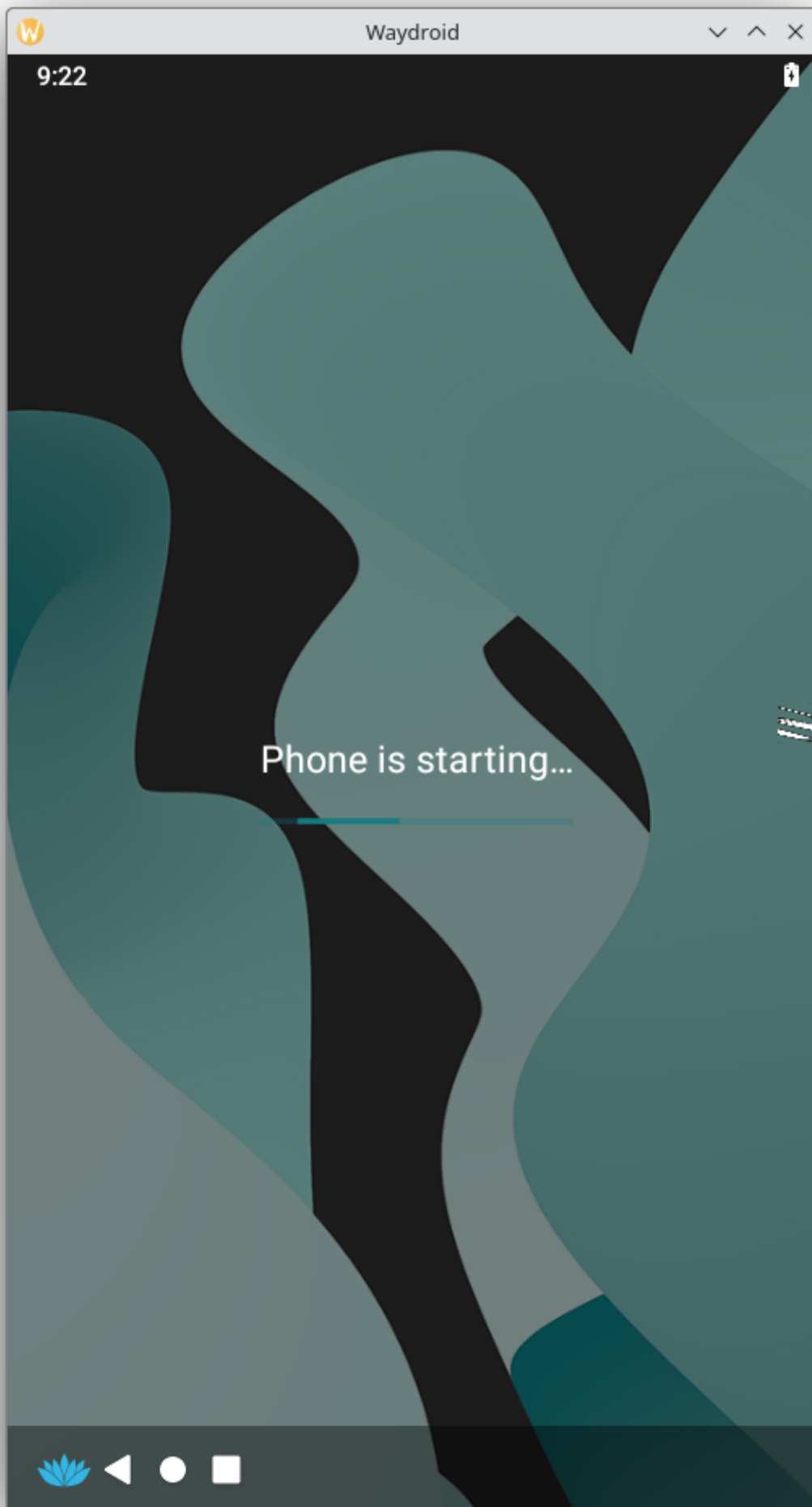
Unter Wayland hat das Client-Fenster leider keine Drag&Drop-Funktion. Wir können es in einen Rahmen einsperren, in dem wir das Tool "cage" verwenden

```
sudo apt install cage
```

# und dann Waydroid ins Fenster zwingen:

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

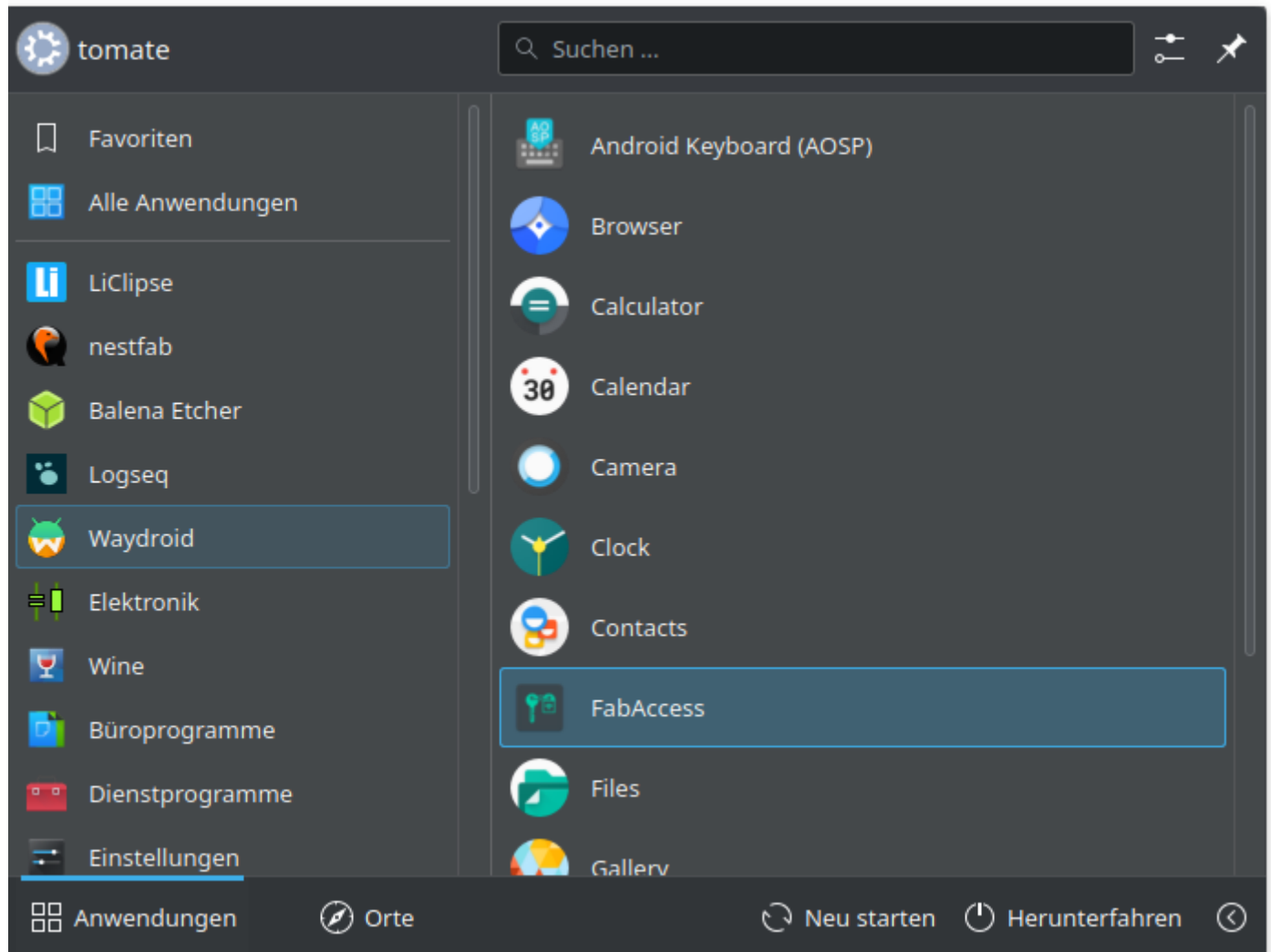
Dadurch erhalten wir folgende Ansicht:



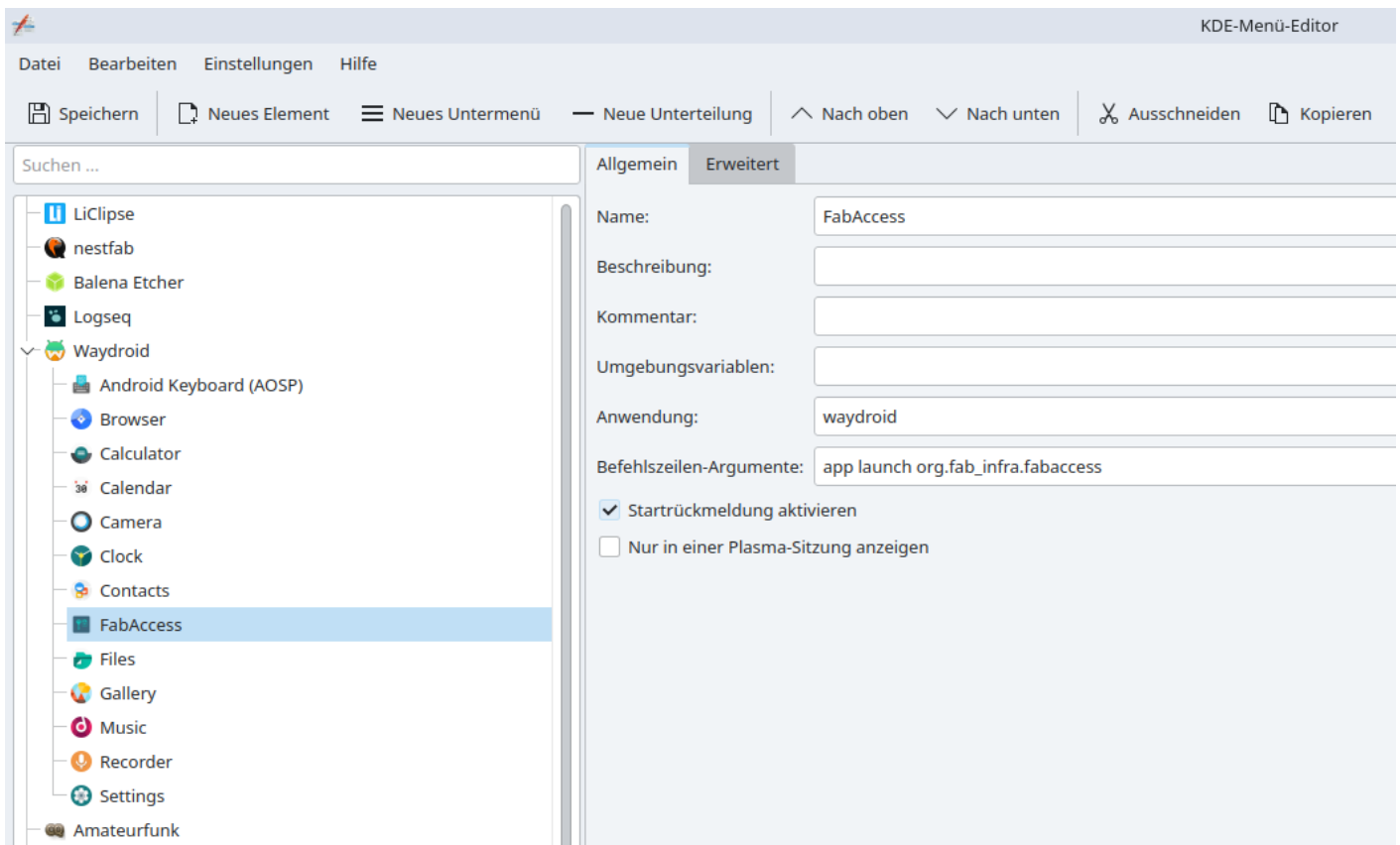


# FabAccess Borepin aus dem Startmenü heraus starten

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
```

## Waydroid Upgrade

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

```
sudo waydroid upgrade
```

Version #16

Erstellt: 4 November 2024 21:40:27 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 27 November 2024 10:17:24 von Mario Voigt (Stadtfabrikanten e.V.)