

# /e/OS - das entgooglete Android

## Ein Akt der digitalen Selbstverteidigung

Jan-Christoph Bornschlegel <jcborn@posteo.de>

23. November 2025

# Navigation

1 Einleitende Worte

2 Was ist /e/OS

3 Installation

4 Warum

# Einleitung

Ich möchte versuchen, Antworten auf diese drei Punkte zu geben:

- Was ist /e/OS?
- Wie installiert man das?
- Warum überhaupt?

# Navigation

1 Einleitende Worte

2 Was ist /e/OS

3 Installation

4 Warum

# Android-Derivat

/e/OS ist ein Android-Derivat. Es gibt verschiedene Androiden mit verschiedenen Eigenschaften. Diese Liste ist nur ein kleiner Auszug.

- Custom roms von XDA Developers Forum
- CyanogenMod ab ca. 2009
- LineageOS (Fork von CyanogenMod)
- ...
- Graphene OS
- /e/ OS

## Besonderheiten von /e/

Die zentrale Idee ist, möglichst wenig Abhängigkeiten zu Google, Inc. zu haben.

- Online-Check nicht an “8.8.8.8” (DNS-Server von Google, Inc.)
- Keine Google-Dienste per Default installiert (können aber natürlich installiert und verwendet werden)
- microG Bibliothek installiert
- Eigenes App Store (“App Lounge”) erlaubt anonymen Zugriff zu freien Apps
- FDroid, Aptoide, ... können installiert werden
- Installation aus anderen Quellen möglich (z.B. Threema Shop)
- Cloud-Dienste von Murena verfügbar

## Woher kenne ich /e/

Ich hatte ein Fairphone 2 mit dem Hersteller-OS “Fairphone Open”.

- hatte keine Google Apps vorinstalliert
- man konnte es einfach rooten
- ...
- ... wurde eingestellt.

Dann kaufte ich das Fairphone 4 und brauchte eine Alternative.

# Zeitschriftenartikel

In unserer Firmenküche liegt ein bekanntes Magazin, in dem zufälligerweise Artikel über verschiedene Andriod-Derivate stehen.

GrapheneOS klingt nach dem, was ich will, aber geht nur auf wenigen (Google-)Geräten.

Auswahl zwischen LineageOS und /e/ OS.

Probiert und hängengeblieben



# Navigation

1 Einleitende Worte

2 Was ist /e/OS

3 Installation

4 Warum

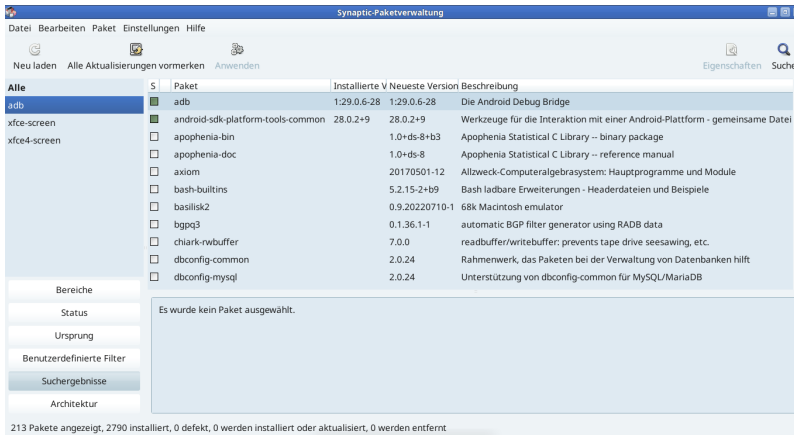
# Voraussetzungen

Die Installation ist denkbar einfach.

<https://doc.e.foundation/devices> bietet einen Installer. Man benötigt zusätzlich:

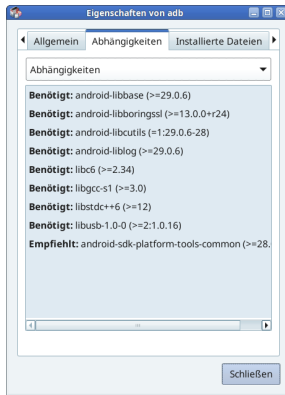
- USB-C-Kabel
- irgendeinen PC (Windows und Linux gehen beide, bei MacOS weiß ich es nicht)
- adb (Android Debug Bridge), gibt es sowohl für Linux als auch Windows
- Da das Gerät neu war, brauchte ich auch kein Backup
- (habe aber sowieso wenig Daten auf dem Gerät, dazu später)

# Verbinden 1



**Abbildung:** Installation “adb” mittels Paketmanager

# Verbinden 2



**Abbildung:** Abhängigkeiten des adb-Paketes

# Verbinden 3

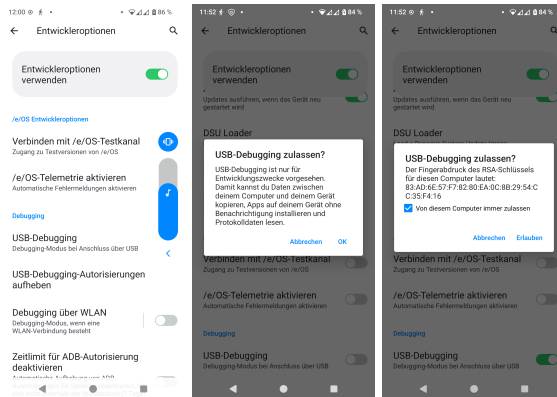
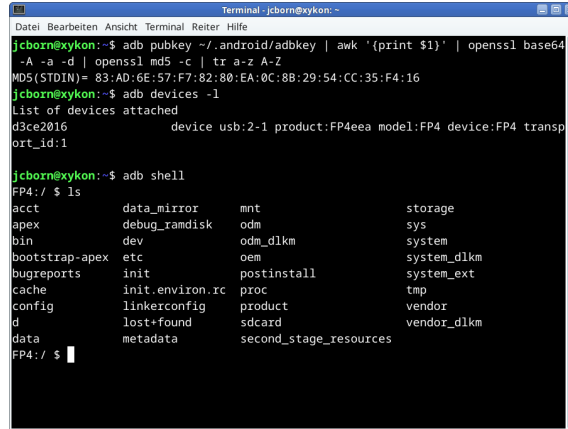


Abbildung: Einschalten und Erlauben von USB-Debugging

# Verbinden 4



```
Terminal - jcborn@xykon: ~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe

jcborn@xykon:~$ adb pubkey ~/.android/adbkey | awk '{print $1}' | openssl base64
-A -a -d | openssl md5 -c | tr a-z A-Z
MD5(STDIN)= 83:AD:6E:57:F7:82:80:EA:0C:8B:29:54:CC:35:F4:16
jcborn@xykon:~$ adb devices -l
List of devices attached
d3ce2016          device usb:2-1 product:FP4eea model:FP4 device:FP4 transp
ort_id:1

jcborn@xykon:~$ adb shell
FP4:/ $ ls
acct          data_mirror   mnt           storage
apex          debug_ramdisk odm            sys
bin           dev           odm_dkkm      system
bootstrap-apex etc           oem           system_dkkm
bugreports    init          postinstall   system_ext
cache         init.environ.rc proc           tmp
config        linkerconfig  product       vendor
d             lost+found    sdcard        vendor_dkkm
data          metadata      second_stage_resources

FP4:/ $
```

Abbildung: Schlüsselprüfung und Verbindung

# Installation 1

- Als ich /e/ installiert hatte, gab es noch ein Image zum Herunterladen und man mußte manuell per adb-Kommando flashen
- Mittlerweile gibt es eine Schritt-für-Schritt-Anleitung auf <https://e.foundation/installer>
- Erklärt leider nicht, wie man den adb-Schlüssel überprüft, aber das macht ja eh niemand außer mir
- Außerdem ist ja nur ein Gerät verbunden, und wenn man einen MITM auf USB-Ebene auf dem Rechner hat, dann hat man Probleme einer anderen Kategorie :)

“MITM”: man in the middle

## Installation 2

- ...irgendwann wird man zum Reboot aufgefordert
- Einrichten wie bei jeder anderen Erstinstallation auch
- Optional: Backup wiederherstellen (ich hatte keins, weil neu)
- WLANs, Bluetooth-Pairings, etc.
- Und dann hat man ein /e/OS-Android
- Updater prüft regelmäßig, bisher gab es keinerlei Probleme
- Leider 4.19er Kernel und Sicherheitspatches nicht mit Graphene vergleichbar



# Navigation

1 Einleitende Worte

2 Was ist /e/OS

3 Installation

4 Warum

# Warum das Ganze?

“**Nimm halt Emacs.**” –talk, ca. 2008

Warum tut man sich das an und nimmt nicht einfach “den Default”?

Wir müssen über Vertrauen reden.

# Trust is hard

Etwas Hintergrund zu mir

- Ich bin mißtrauisch gegenüber jeder Art von Autorität
- Es ist nie die Frage, **ob** Macht mißbraucht wird, sondern nur **wann und von wem zuerst**
- Zentralisierung ist nicht zu unserem Vorteil
- Marktlogiken erzwingen Zentralisierung und Skalierung
- Freiheit erfordert genau das Gegenteil
- Defaults sind böse (Dateiendungen unter Windows per Default aus usw.)

## Rückblick 1

- Mobiltelefone müssen stets wissen, wo sie sind, bzw. das Netz muß es wissen
- ich wollte nie eines, weil ich nicht permanent erreichbar sein wollte
- Dann passierte ein Unfall, und ich war froh, daß jemand eines dabei hatte
- erstes Mobiltelefon “Alcatel One Touch Easy” ca. 2000
- Gmail-Konto, als man noch Invites brauchte und Google noch “don’t be evil” als Motto auf der Seite hatte
- Twitter-Account, als das noch per SMS ging
- Erstes “Smartphone” Nokia E61, als Firmenhandy, 2010
- Erstes eigenes “Smartphone” HTC Desire HD, ca. 2011

## Rückblick 2

Besagtes HTC Desire war zum Kaufzeitpunkt schon veraltet.  
Es war Android 2.3.5 drauf, als der “heiße Scheiß” gerade 4.0 war.  
Das war frustrierend, weil gefühlt die Hälfte nicht funktionierte, was alle Anderen hatten.  
Abhilfe:

- in xda-developers nach einem neuerem Custom ROM gesucht
- Registrieren bei HTC als developer
- Bootloader unlocked, das ging damals nur von HTC's Gnaden für registrierte Entwickler (wie “Hans Dampf <blafasel@wegwerfemail.net>”)
- Image eingespielt, neueres Android.
- Natürlich keine Updates, schon gar nicht automatisch

## Rückblick 3

Das nächste Gerät war dann das Fairphone 2. Ich lege Wert auf folgende Eigenschaften:

- Gerät kann geöffnet und vollständig entstromt werden
- Gerät hat ordentlichen GPS-Empfänger und genug Speicher für Offline-Karten
- es gibt Ersatzteile zu kaufen, insbesondere Display und Akku

# Meine Mobiltelefonnutzung

- Navigation mit OSMAnd und Offlinekarten
- Datentarif für E-Mail und Messenger, E-Mail aber nur unverschlüsselt
- Musikstream auf Bluetooth-Kopfhörer
- als Kamera, wenn ich die Kamera nicht mit habe
- Manchmal als WLAN-Hotspot für andere Geräte
- Zweithandy ohne SIM-Karte als Fernbedienung für eine GoPro und eine Drohne

# Vertrauen

“Ja, aber welchen Geräten vertraust Du denn dann?”

**Keinem.**

Es gibt hier leider keine schöne Antwort. Es gibt nur verschiedene Levels an Mißtrauen, und Abwägungen, um im Alltag klarzukommen.

Schönes Paper: “Reflections on trusting Trust” von Ken Thompson, 1984.

<https://dl.acm.org/doi/10.1145/358198.358210>



# Klarkommen

“Und wie *kommst* Du dann klar?”

- Coreboot, Qubes, Festplattenverschlüsselung, GnuPG, Signal, Threema
- Geräte bei Grenzübertritten ausmachen
- Datensicherungen auf Festplatten, die ich selbst unter Kontrolle habe
- Kabelnetz
- keine “smarten” Geräte wie Amazon Echo
- Fernseher hinter Proxy und PiHole

## Closing

Ich bitte die eventuell entstandene schlechte Laune zu entschuldigen.

# Vielen Dank für die Aufmerksamkeit.