



# Android mit Google Befreiung, PlayStore Apps im eigenen Repo

Ubucon 2015, Berlin

25. Oktober 2015



Alexander Rudolf  
Linux Consultant  
B1 Systems GmbH  
rudolf@b1-systems.de

# Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- über 70 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
  - Beratung & Consulting
  - Support
  - Entwicklung
  - Training
  - Betrieb
  - Lösungen
- dezentrale Strukturen

# Schwerpunkte

- Virtualisierung (XEN, KVM & RHEV)
- Systemmanagement (Spacewalk, Red Hat Satellite, SUSE Manager)
- Konfigurationsmanagement (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (OpenStack & SUSE Cloud & RDO)
- Hochverfügbarkeit (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- Dateiaustausch (ownCloud)
- Paketierung (Open Build Service)
- Administratoren oder Entwickler zur Unterstützung des Teams vor Ort

# Google Android mit Google Konto

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...



# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Ein Smartphone mit Google Android und Google Konto synchronisiert eine Menge von Daten zwischen Gerät und Google Servern:

- Browserverlauf, Favoriten
- Standortverlauf, Suchverlauf in Google Maps
- WLAN Einstellungen inkl. Passwörter
- Nutzungsdaten der Apps, Installationsprotokoll
- Kalender und Kontakteinträge
- Youtube Such- und Wiedergabeverlauf
- Fotos, Videos, Sprach- und Audioaktivitäten
- ...

# Android mit Google Konto

Manche Synchronisierungsoptionen lassen sich abschalten. Teilweise sind diese nach einem Android Update aber wieder aktiviert oder neue kommen hinzu.

Tipp: In Google „Mein Konto“ hat man die Möglichkeit die Daten einzusehen und zu löschen

<https://myaccount.google.com/>

# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- **Apps aus alternativen App Stores beziehen (F-Droid, Amazon)**

# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- **Apps aus alternativen App Stores beziehen (F-Droid, Amazon)**

# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- **Apps aus alternativen App Stores beziehen (F-Droid, Amazon)**



# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- **Apps aus alternativen App Stores beziehen (F-Droid, Amazon)**

# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- Apps aus alternativen App Stores beziehen (F-Droid, Amazon)

# Android mit Google Konto

Was kann man neben Abschalten der Google Synchronisierung noch tun?

- alternative Browser (z.B. Firefox, Dolphin)
- anderes Kartenmaterial und Apps (z.B. OpenStreetMap, OsmAnd, Locus)
- Network Location Provider anderer Anbieter (z.B.  $\mu$ g UnifiedNlp, Mozilla, Apple, Nominatim)
- Kalender, Kontakte auf eigenem Server (z.B. DAVdroid, ownCloud)
- Fotos, Videos mit eigenem Server synchronisieren (z.B. ownCloud, Tine 2.0)
- **Apps aus alternativen App Stores beziehen (F-Droid, Amazon)**

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!



# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

Warnung: Separat heruntergeladene Apps

Diese Apps erhalten keine automatischen Updates!

# Android mit Google Konto

Apps ausschließlich aus alternativen Appstores zu beziehen, bedeutet:

- Verzicht auf viele Apps die nur im Google Playstore zu finden sind.
  - z.B. DB Navigator, Dolphin, Llama, ...
- einige Apps sind weiterhin verfügbar
  - z.B. Firefox, OsmAnd (F-Droid)
- einzelne Apps sind beim Anbieter separat herunterladbar:
  - z.B. Öffi, Threema

**Warnung: Separat heruntergeladene Apps**

Diese Apps erhalten keine automatischen Updates!



# Google Android mit F-Droid

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- **F-Droid Repository**
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- **F-Droid Client** (apk)
- **F-Droid Server**

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- F-Droid Server

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- F-Droid Server

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- F-Droid Server

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- F-Droid Server



# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- F-Droid Server

# Android mit F-Droid

F-Droid bietet eine vollständige Infrastruktur für die Paketverwaltung unter Android.

- F-Droid Repository
  - ausschließlich freie und quelloffene Software (FOSS)
  - Quellcode jeder App bei F-Droid verfügbar
  - ohne „Anti-Features“ (Werbung, User-Tracking, Abhängigkeiten zu nicht freier Software)
  - anonyme Benutzung, da keine Anmeldung erforderlich ist
- F-Droid Client (apk)
- **F-Droid Server**

# Android mit F-Droid

F-Droid

<https://f-droid.org/>

F-Droid Wiki

<https://f-droid.org/wiki/>

F-Droid Server (fdroidserver)

<https://gitlab.com/fdroid/fdroidserver>

# Google Android mit PlayStore Apps

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo bekomme ich die Installationspakete her?
  - Im PlayStore Download ohne Authentifizierung nicht möglich
  - Download-Portale (z.B. [apk-downloader.org](http://apk-downloader.org)) funktionieren nicht perfekt. Unsichere Drittquelle!
  - **Ein ausrangiertes Smartphone** ist der perfekte Ersatz: Apps lassen sich beliebig installieren. Automatische Aktualisierung! Google Konto notwendig!

Empfehlung: Neues Google Konto

Ein neues Konto anlegen und nur für diesen Zweck benutzen!

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo bekomme ich die Installationspakete her?
  - Im PlayStore Download ohne Authentifizierung nicht möglich
  - Download-Portale (z.B. [apk-downloader.org](http://apk-downloader.org)) funktionieren nicht perfekt. Unsichere Drittquelle!
  - **Ein ausrangiertes Smartphone** ist der perfekte Ersatz: Apps lassen sich beliebig installieren. Automatische Aktualisierung! Google Konto notwendig!

Empfehlung: Neues Google Konto

Ein neues Konto anlegen und nur für diesen Zweck benutzen!

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo bekomme ich die Installationspakete her?
  - Im PlayStore Download ohne Authentifizierung nicht möglich
  - Download-Portale (z.B. [apk-downloader.org](http://apk-downloader.org)) funktionieren nicht perfekt. Unsichere Drittquelle!
  - **Ein ausrangiertes Smartphone** ist der perfekte Ersatz: Apps lassen sich beliebig installieren. Automatische Aktualisierung! Google Konto notwendig!

Empfehlung: Neues Google Konto

Ein neues Konto anlegen und nur für diesen Zweck benutzen!

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo bekomme ich die Installationspakete her?
  - Im PlayStore Download ohne Authentifizierung nicht möglich
  - Download-Portale (z.B. [apk-downloader.org](http://apk-downloader.org)) funktionieren nicht perfekt. Unsichere Drittquelle!
  - **Ein ausrangiertes Smartphone** ist der perfekte Ersatz: Apps lassen sich beliebig installieren. Automatische Aktualisierung! Google Konto notwendig!

Empfehlung: Neues Google Konto

Ein neues Konto anlegen und nur für diesen Zweck benutzen!



# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo bekomme ich die Installationspakete her?
  - Im PlayStore Download ohne Authentifizierung nicht möglich
  - Download-Portale (z.B. [apk-downloader.org](http://apk-downloader.org)) funktionieren nicht perfekt. Unsichere Drittquelle!
  - **Ein ausrangiertes Smartphone** ist der perfekte Ersatz: Apps lassen sich beliebig installieren. Automatische Aktualisierung! Google Konto notwendig!

**Empfehlung: Neues Google Konto**

Ein neues Konto anlegen und nur für diesen Zweck benutzen!

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo kann ich die Installationspakete (apk) finden?
  - Wenn das Smartphone ge'rooted' ist, lassen sich die apk-Dateien der installierten Pakete im Dateisystem finden.
  - Apps liegen in /data/app
  - System Apps liegen in /system/app

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo kann ich die Installationspakete (apk) finden?
  - Wenn das Smartphone ge'rooted' ist, lassen sich die apk-Dateien der installierten Pakete im Dateisystem finden.
  - Apps liegen in `/data/app`
  - System Apps liegen in `/system/app`

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo kann ich die Installationspakete (apk) finden?
  - Wenn das Smartphone ge'rooted' ist, lassen sich die apk-Dateien der installierten Pakete im Dateisystem finden.
  - Apps liegen in /data/app
  - System Apps liegen in /system/app

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wo kann ich die Installationspakete (apk) finden?
  - Wenn das Smartphone ge'rooted' ist, lassen sich die apk-Dateien der installierten Pakete im Dateisystem finden.
  - Apps liegen in /data/app
  - System Apps liegen in /system/app

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wie kann ich die apk-Dateien kopieren?
  - In Terminal Emulator App wie z.B. ConnectBot kann man die Dateien finden und über die WLAN-Netzwerkverbindung transferieren. Zeitaufwändige, manuelle Arbeit.
  - Eine App, die den cron-Dienst benutzt, ist eine gute Lösung. Beim `cronscheduler` ist in der freien Version nach jedem Reboot manueller Start notwendig. Einige Vorbereitung (SSH Keys, Skript) notwendig.
  - **Android Debug Bridge** (ADB) funktioniert direkt über USB: Dateien kopieren, Befehle abzusetzen, Firmware Update, Reboot, usw.

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wie kann ich die apk-Dateien kopieren?
  - In Terminal Emulator App wie z.B. ConnectBot kann man die Dateien finden und über die WLAN-Netzwerkverbindung transferieren. Zeitaufwändige, manuelle Arbeit.
  - Eine App, die den cron-Dienst benutzt, ist eine gute Lösung. Beim `cronscheduler` ist in der freien Version nach jedem Reboot manueller Start notwendig. Einige Vorbereitung (SSH Keys, Skript) notwendig.
  - **Android Debug Bridge** (ADB) funktioniert direkt über USB: Dateien kopieren, Befehle abzusetzen, Firmware Update, Reboot, usw.

# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wie kann ich die apk-Dateien kopieren?
  - In Terminal Emulator App wie z.B. ConnectBot kann man die Dateien finden und über die WLAN-Netzwerkverbindung transferieren. Zeitaufwändige, manuelle Arbeit.
  - Eine App, die den cron-Dienst benutzt, ist eine gute Lösung. Beim `cronscheduler` ist in der freien Version nach jedem Reboot manueller Start notwendig. Einige Vorbereitung (SSH Keys, Skript) notwendig.
  - **Android Debug Bridge** (ADB) funktioniert direkt über USB: Dateien kopieren, Befehle abzusetzen, Firmware Update, Reboot, usw.



# Android mit PlayStore Apps

Wie bekomme ich Google PlayStore Apps ohne Google Konto installiert?

- Wie kann ich die apk-Dateien kopieren?
  - In Terminal Emulator App wie z.B. ConnectBot kann man die Dateien finden und über die WLAN-Netzwerkverbindung transferieren. Zeitaufwändige, manuelle Arbeit.
  - Eine App, die den cron-Dienst benutzt, ist eine gute Lösung. Beim `cronscheduler` ist in der freien Version nach jedem Reboot manueller Start notwendig. Einige Vorbereitung (SSH Keys, Skript) notwendig.
  - **Android Debug Bridge** (ADB) funktioniert direkt über USB: Dateien kopieren, Befehle abzusetzen, Firmware Update, Reboot, usw.

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereitstellen?

- apk-Dateien auf das aktive Smartphone kopieren: Zeitaufwändige, manuelle Arbeit.
- Die Dateien auf einen Webserver legen und von dort per Browser installieren: Nur bei wenigen Apps übersichtlich, keine Übersicht über Versionen/Aktualisierungen.
- Ein eigenes Repository, das die apk-Dateien vorhält, Versionen verwaltet und Metadaten anbietet, ist die beste Lösung.
  - **F-Droid Server** is your friend!

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereitstellen?

- apk-Dateien auf das aktive Smartphone kopieren: Zeitaufwändige, manuelle Arbeit.
- Die Dateien auf einen Webserver legen und von dort per Browser installieren: Nur bei wenigen Apps übersichtlich, keine Übersicht über Versionen/Aktualisierungen.
- Ein eigenes Repository, das die apk-Dateien vorhält, Versionen verwaltet und Metadaten anbietet, ist die beste Lösung.
  - **F-Droid Server** is your friend!

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereitstellen?

- apk-Dateien auf das aktive Smartphone kopieren: Zeitaufwändige, manuelle Arbeit.
- Die Dateien auf einen Webserver legen und von dort per Browser installieren: Nur bei wenigen Apps übersichtlich, keine Übersicht über Versionen/Aktualisierungen.
- Ein eigenes Repository, das die apk-Dateien vorhält, Versionen verwaltet und Metadaten anbietet, ist die beste Lösung.
  - **F-Droid Server** is your friend!

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereitstellen?

- apk-Dateien auf das aktive Smartphone kopieren: Zeitaufwändige, manuelle Arbeit.
- Die Dateien auf einen Webserver legen und von dort per Browser installieren: Nur bei wenigen Apps übersichtlich, keine Übersicht über Versionen/Aktualisierungen.
- Ein eigenes Repository, das die apk-Dateien vorhält, Versionen verwaltet und Metadaten anbietet, ist die beste Lösung.
  - **F-Droid Server** is your friend!

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereit stellen?

- Dateien werden auf einen Rechner kopiert, der 24/7 läuft. Z.B. ein RasperryPi im Heimnetz am DSL-Anschluss.
- In regelmäßigen Abständen kopiert ein Job die apk-Dateien per ADB vom Smartphone mit Google Konto (Updates!)
- Ein weiterer Job aktualisiert per `fdroidserver`-Skript die Metadaten und stellt damit automatisch alle Apps und Updates bereit. Benötigt Android SDK (`aapt`).
- Ein Webserver (Apache `httpd`, `nginx`, ...) veröffentlicht das Repository.
- Mit Hilfe von Portforwarding und einen dynamischen DNS-Dienst ist es möglich, das Repository auch via Internet freizugeben.

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereit stellen?

- Dateien werden auf einen Rechner kopiert, der 24/7 läuft. Z.B. ein RasperryPi im Heimnetz am DSL-Anschluss.
- In regelmäßigen Abständen kopiert ein Job die apk-Dateien per ADB vom Smartphone mit Google Konto (Updates!)
- Ein weiterer Job aktualisiert per `fdroidserver`-Skript die Metadaten und stellt damit automatisch alle Apps und Updates bereit. Benötigt Android SDK (`aapt`).
- Ein Webserver (Apache `httpd`, `nginx`, ...) veröffentlicht das Repository.
- Mit Hilfe von Portforwarding und einen dynamischen DNS-Dienst ist es möglich, das Repository auch via Internet freizugeben.

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereit stellen?

- Dateien werden auf einen Rechner kopiert, der 24/7 läuft. Z.B. ein RasperryPi im Heimnetz am DSL-Anschluss.
- In regelmäßigen Abständen kopiert ein Job die apk-Dateien per ADB vom Smartphone mit Google Konto (Updates!)
- Ein weiterer Job aktualisiert per `fdroidserver`-Skript die Metadaten und stellt damit automatisch alle Apps und Updates bereit. Benötigt Android SDK (`aapt`).
- Ein Webserver (Apache `httpd`, `nginx`, ...) veröffentlicht das Repository.
- Mit Hilfe von Portforwarding und einen dynamischen DNS-Dienst ist es möglich, das Repository auch via Internet freizugeben.



# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereit stellen?

- Dateien werden auf einen Rechner kopiert, der 24/7 läuft. Z.B. ein RasperryPi im Heimnetz am DSL-Anschluss.
- In regelmäßigen Abständen kopiert ein Job die apk-Dateien per ADB vom Smartphone mit Google Konto (Updates!)
- Ein weiterer Job aktualisiert per `fdroidserver`-Skript die Metadaten und stellt damit automatisch alle Apps und Updates bereit. Benötigt Android SDK (`aapt`).
- Ein Webserver (Apache `httpd`, `nginx`, ...) veröffentlicht das Repository.
- Mit Hilfe von Portforwarding und einen dynamischen DNS-Dienst ist es möglich, das Repository auch via Internet freizugeben.

# Android mit PlayStore Apps

Wie kann ich die apk-Dateien für mein aktives Smartphone bereit stellen?

- Dateien werden auf einen Rechner kopiert, der 24/7 läuft. Z.B. ein RasperryPi im Heimnetz am DSL-Anschluss.
- In regelmäßigen Abständen kopiert ein Job die apk-Dateien per ADB vom Smartphone mit Google Konto (Updates!)
- Ein weiterer Job aktualisiert per `fdroidserver`-Skript die Metadaten und stellt damit automatisch alle Apps und Updates bereit. Benötigt Android SDK (`aapt`).
- Ein Webserver (Apache `httpd`, `nginx`, ...) veröffentlicht das Repository.
- Mit Hilfe von Portforwarding und einen dynamischen DNS-Dienst ist es möglich, das Repository auch via Internet freizugeben.

# Android mit PlayStore Apps

Wie bekomme ich die Apps auf mein aktives Smartphone?

- 1 F-Droid Client installieren.
- 2 Die Repository URL als neue Paketquelle konfigurieren.
- 3 Paketquellen aktualisieren.
- 4 Apps installieren.
- 5 Updates erscheinen automatisch und können direkt installiert werden.

# Android mit PlayStore Apps

Wie bekomme ich die Apps auf mein aktives Smartphone?

- 1 F-Droid Client installieren.
- 2 Die Repository URL als neue Paketquelle konfigurieren.
- 3 Paketquellen aktualisieren.
- 4 Apps installieren.
- 5 Updates erscheinen automatisch und können direkt installiert werden.

# Android mit PlayStore Apps

Wie bekomme ich die Apps auf mein aktives Smartphone?

- 1 F-Droid Client installieren.
- 2 Die Repository URL als neue Paketquelle konfigurieren.
- 3 Paketquellen aktualisieren.
- 4 Apps installieren.
- 5 Updates erscheinen automatisch und können direkt installiert werden.

# Android mit PlayStore Apps

Wie bekomme ich die Apps auf mein aktives Smartphone?

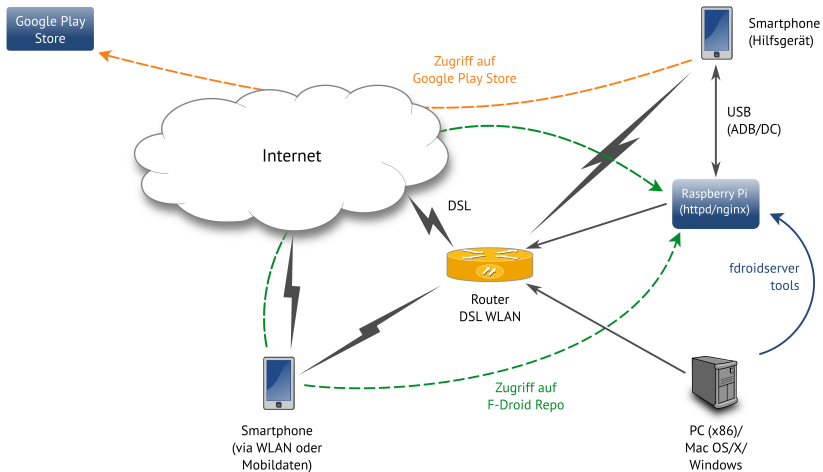
- 1 F-Droid Client installieren.
- 2 Die Repository URL als neue Paketquelle konfigurieren.
- 3 Paketquellen aktualisieren.
- 4 Apps installieren.
- 5 Updates erscheinen automatisch und können direkt installiert werden.

# Android mit PlayStore Apps

Wie bekomme ich die Apps auf mein aktives Smartphone?

- 1 F-Droid Client installieren.
- 2 Die Repository URL als neue Paketquelle konfigurieren.
- 3 Paketquellen aktualisieren.
- 4 Apps installieren.
- 5 Updates erscheinen automatisch und können direkt installiert werden.

# Android mit PlayStore Apps







# Umsetzung

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und fdroidserver
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und `fdroidserver`
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und `fdroidserver`
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und fdroidserver
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und fdroidserver
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Voraussetzungen für die Umsetzung der Lösung:

- Google Konto zum Herunterladen der Apps
- Smartphone (Hilfsgerät), rooted + USB Kabel
- Raspberry Pi mit einem Webserver
- PC (x86) mit Android SDK und fdroidserver
- Internet Verbindung inkl. WLAN
- Android Smartphone(s)

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen `rsync`)
- 3 Auf dem Raspberry Pi `adb` installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 `rsync` Job anlegen



# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen rsync)
- 3 Auf dem Raspberry Pi adb installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 rsync Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen rsync)
- 3 Auf dem Raspberry Pi adb installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 rsync Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen `rsync`)
- 3 Auf dem Raspberry Pi `adb` installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 `rsync` Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen `rsync`)
- 3 Auf dem Raspberry Pi `adb` installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 `rsync` Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen `rsync`)
- 3 Auf dem Raspberry Pi `adb` installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 `rsync` Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Zuerst das alte Smartphone und den Raspberry Pi einrichten ...

- 1 Google Konto auf Smartphone einrichten (ggf. erste Apps installieren)
- 2 BusyBox installieren (wir brauchen `rsync`)
- 3 Auf dem Raspberry Pi `adb` installieren
- 4 Verbindung über USB Kabel testen
- 5 SSH Key anlegen und den Public Key übertragen
- 6 Verzeichnisse anlegen, Webserver konfigurieren
- 7 `rsync` Job anlegen

# Umsetzung

Alle Schritte im Überblick:

Dann das Android SDK auf den PC und zum Schluß F-Droid auf das Smartphone ...

- 8 Auf dem PC werden Android SDK und `fdroidserver` benötigt
- 9 Initialisierung des F-Droid Repositorys
- 10 Einen Job anlegen, der die `apk` Dateien synchronisiert ... und das F-Droid Repository aktualisiert.
- 11 F-Droid auf dem aktuellen Smartphone installieren
- 12 F-Droid Repository einrichten

# Umsetzung

Alle Schritte im Überblick:

Dann das Android SDK auf den PC und zum Schluß F-Droid auf das Smartphone ...

- 8 Auf dem PC werden Android SDK und `fdroidserver` benötigt
- 9 Initialisierung des F-Droid Repositorys
- 10 Einen Job anlegen, der die `apk` Dateien synchronisiert  
... und das F-Droid Repository aktualisiert.
- 11 F-Droid auf dem aktuellen Smartphone installieren
- 12 F-Droid Repository einrichten



# Umsetzung

Alle Schritte im Überblick:

Dann das Android SDK auf den PC und zum Schluß F-Droid auf das Smartphone ...

- 8 Auf dem PC werden Android SDK und `fdroidserver` benötigt
- 9 Initialisierung des F-Droid Repositorys
- 10 Einen Job anlegen, der die `apk` Dateien synchronisiert ... und das F-Droid Repository aktualisiert.
- 11 F-Droid auf dem aktuellen Smartphone installieren
- 12 F-Droid Repository einrichten

# Umsetzung

Alle Schritte im Überblick:

Dann das Android SDK auf den PC und zum Schluß F-Droid auf das Smartphone ...

- 8 Auf dem PC werden Android SDK und `fdroidserver` benötigt
- 9 Initialisierung des F-Droid Repositorys
- 10 Einen Job anlegen, der die `apk` Dateien synchronisiert ... und das F-Droid Repository aktualisiert.
- 11 F-Droid auf dem aktuellen Smartphone installieren
- 12 F-Droid Repository einrichten

# Umsetzung

Alle Schritte im Überblick:

Dann das Android SDK auf den PC und zum Schluß F-Droid auf das Smartphone ...

- 8 Auf dem PC werden Android SDK und `fdroidserver` benötigt
- 9 Initialisierung des F-Droid Repositorys
- 10 Einen Job anlegen, der die `apk` Dateien synchronisiert ... und das F-Droid Repository aktualisiert.
- 11 F-Droid auf dem aktuellen Smartphone installieren
- 12 F-Droid Repository einrichten

# Umsetzung

Installation adb und Verbindung testen:

```
pi@raspberrypi ~ $ sudo apt install android-tools-adb

pi@raspberrypi ~ $ adb devices -l
List of devices attached
0123456789ABCDEF          device usb:1-1.2 product: \
                           cm_ancora model: GT_I8150 device:ancora

pi@raspberrypi ~ $ sudo adb shell
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
root@android:/ #
```

# Umsetzung

## SSH Key anlegen und den Public Key übertragen

```
root@android:/ # ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/data/.ssh/id_rsa): \
        Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /data/.ssh/id_rsa.
Your public key has been saved in /data/.ssh/id_rsa.pub.
root@android:/ # cat /data/.ssh/id_rsa.pub # kopieren
root@android:/ # ^D
pi@raspberrypi ~ $ mkdir .ssh
pi@raspberrypi ~ $ chmod 700 .ssh
pi@raspberrypi ~ $ cd .ssh
pi@raspberrypi ~ $ vi .ssh/authorized_keys # einfügen
```

# Umsetzung

Verzeichnisse anlegen, Webserver konfigurieren

```
pi@raspberrypi ~ $ sudo mkdir -p /storage/android/data/app
pi@raspberrypi ~ $ sudo chown pi.pi -Rv /storage/android/
pi@raspberrypi ~ $ sudo mkdir -p /var/www/html/fdroid/
pi@raspberrypi ~ $ sudo chown pi.pi -Rv /var/www/html/fdroid/
pi@raspberrypi ~ $ sudo apt-get install nginx
pi@raspberrypi ~ $ sudo systemctl start nginx.service
```

# Umsetzung

## rsync Job anlegen

```
pi@raspberrypi ~ $ sudo crontab -e
0 * * * * /usr/bin/adb shell "rsync -av          \
  -e 'ssh -i /data/.ssh/id_rsa' /data/app/      \
  pi@raspberrypi:/storage/android/data/app/"    \
  2>&1 | /usr/bin/logger -t apksync
pi@raspberrypi ~ $ sudo journalctl -f | grep apksync
apksync[6813]: sending incremental file list
apksync[6813]:
apksync[6813]: sent 406 bytes received 12 bytes 33 bytes/sec
apksync[6813]: total size is 156784064 speedup is 375081.49
```

## Umsetzung

Auf dem PC werden Android SDK und fdroidserver benötigt

[https://f-droid.org/wiki/page/Installing\\_the\\_Server\\_and\\_Repo\\_Tools](https://f-droid.org/wiki/page/Installing_the_Server_and_Repo_Tools)

<https://developer.android.com/sdk/index.html#Other>

```
$ tar -xzvf android-sdk_r24.3.4-linux.tgz
$ export ANDROID_HOME=~/.android-sdk-linux/
$ export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/ \
    platform-tools
$ vi ~/.bashrc      # die zwei Zeilen übertragen
$ android update sdk --no-ui --filter build-tools-23.0.1
$ ll android-sdk-linux/build-tools/19.1.0/aapt
-rwxrwxr-x. 1 android android 1109734 23. Okt 11:40 \
    android-sdk-linux/build-tools/19.1.0/aapt
$ sudo apt install fdroidserver
```



# Umsetzung

## Initialisierung des F-Droid Repositorys

```
$ mkdir -p fdroid/bin/ fdroid/staging/  
$ cd fdroid/bin/  
$ touch refreshfdroid blacklist getoeffiaosp.sh  
$ chmod +x refreshfdroid getoeffiaosp.sh  
$ cd fdroid/staging/  
$ fdroid init
```

# Umsetzung

fdroid/staging/config.py anpassen!

```
repo_url = "http://ubucon-fdroid.selfhost.eu/fdroid/repo"
repo_name = "Ubucon F-Droid Repo"
repo_description = """
rudolf@b1-systems.de
"""
archive_older = 3
archive_url = "http://ubucon-fdroid.selfhost.eu/fdroid \
              /archive"
archive_name = "Ubucon F-Droid Repo Archive"
archive_description = """
The repository of older versions of applications
"""
serverwebroot = 'pi@raspberrypi:/var/www/html/fdroid'
```

# Umsetzung

fdroid/bin/refreshfdroid

```
#!/bin/bash

export ANDROID_HOME=~/.android-sdk-linux/
export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/ \
    platform-tools

cd ~/fdroid
[ -d "staging/repo/" ] || mkdir -p "staging/repo/"
cd "staging/repo/"
rsync -avz --exclude-from="../../bin/blacklist" \
    pi@raspberrypi:/storage/android/data/app/ ./
~/fdroid/bin/getoeffiaosp.sh
cd ~/fdroid/staging
~/fdroidserver/fdroid update --create-metadata
~/fdroidserver/fdroid server update -v
```

# Umsetzung

fdroid/bin/blacklist

```
com.adobe.flashplayer*.apk  
com.google.android*.apk  
com.limbenjamin.cronscheduler*.apk  
com.android.vending*.apk  
de.komoot.android-1.apk  
de.komoot.android-2.apk
```

# Umsetzung

```
fdroid/bin/getoeffiaosp.sh
```

```
#!/bin/bash
```

```
OEFFIDL="http://oeffi.schildbach.de/"
```

```
APKFILE=$(curl -s ${OEFFIDL}/download.html | \
    egrep "oeffi-[0-9]+.[0-9]+.*-aosp.apk" | \
    sed -e 's/.*a href=\"oeffi-/oeffi-/' \
    -e 's/.apk\".*/.apk/')
```

```
echo curl -s -o "$APKFILE" -C - "${OEFFIDL}/${APKFILE}"
curl -s -o "$APKFILE" -C - "${OEFFIDL}/${APKFILE}"
```

## Umsetzung

Einen Job anlegen, der die apk Dateien synchronisiert  
... und das F-Droid Repository aktualisiert.

```
$ crontab -e
10 * * * * /home/android/fdroid/bin/refreshfdroid 2>&1 | \
            /usr/bin/logger -t fdroidupdate
$ journalctl -f | grep fdroidupdate
fdroidupdate[751]: sent 209 bytes received 128 bytes 67...
fdroidupdate[751]: total size is 5,980,951 speedup is 17.
fdroidupdate[751]: curl -s -o oeffi-8.96.2-aosp.apk -C - \
            http://oeffi.schildbach.de//oeffi-8.96.2-aosp.apk
fdroidupdate[751]: DEBUG: Reading config.py
fdroidupdate[751]: INFO: rsyncing repo to \
            pi@raspberrypi:/var/www/html/fdroid/
fdroidupdate[751]: building file list ... done
fdroidupdate[751]: repo/oeffi-8.96.2-aosp.apk
```

# Umsetzung

F-Droid auf dem aktuellen Smartphone installieren

`https://f-droid.org/FDroid.apk`

F-Droid Repository einrichten

`http://ubucon-fdroid.selfhost.eu/fdroid/repo`

`http://\[UbuconIPimWLAN\]/fdroid/repo`

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an [info@b1-systems.de](mailto:info@b1-systems.de)  
oder +49 (0)8457 - 931096