Flatpak verstehen und paketieren

About me

Christian Loritz
Senior Software Engineer
Open Source und Linux Expert

Kontaktmöglichkeit:

Instagram: @chris.loritz

Übersicht

- Was ist Flatpak
- Für welche Systeme gibt es Flatpak?
- Warum Flatpak?
- Flatpak Basis Konzepte
- Flatpak vs. x
- Flatpak einsetzen
- Flatpak bauen

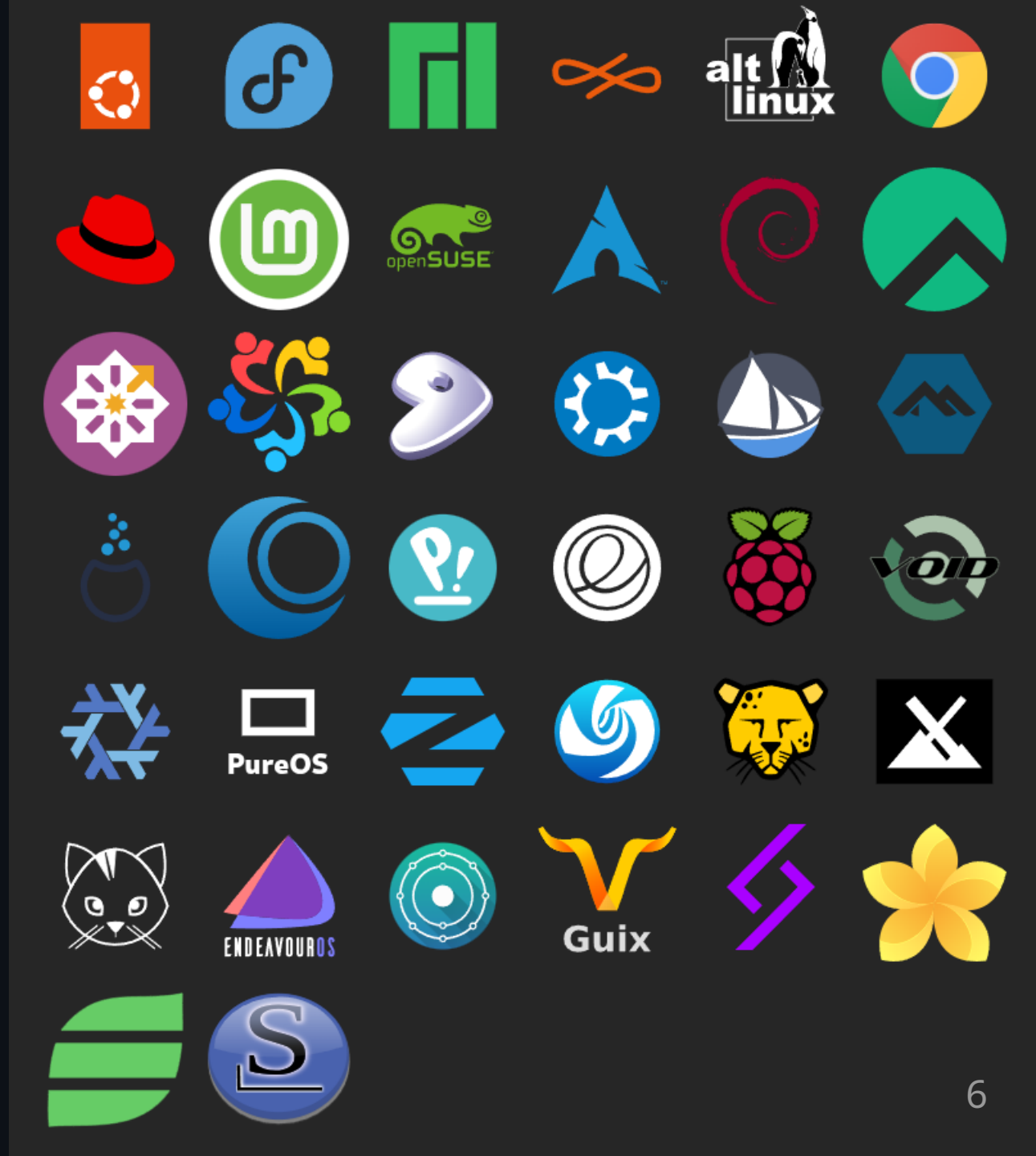
Allgemeines

- Fragen können jederzeit gestellt werden, bevorzugt wird es aber eher am Ende des Vortrags im Q&A Bereich
- Bitte verzeiht Fehler, zeitlich war es sportlich

Was ist Flatpak

- Sandbox für Desktop Anwendungen
- Die Flatpak Anwendung gibt es für das Terminal und als integration in z.B. Gnome Software oder KDE Discover
- Basiert auf OSTree und nennt sich auch "Git for Apps"

Für welche Systeme gibt es Flatpak?



Warum Flatpak?

- Ein "Package Manager" bzw. ein Format für alle Linux Distributionen
- Probleme mit unterschiedlichen Systemen und Package Manager beheben
- Breitere Zugänglichkeit
- Für Entwickler ein Tool bereitstellen, dass der Verbreitungsprozess beschleunigt wird
- Alte oder Outdated Packages

Flatpak Basis Konzepte

Runtimes

- Besitzen die Grund Software Dependencies für die Anwendungen
- Alle Anwendungen müssen mit einer Runtime erstellt werden
- Flatpak installiert automatisch die benötigte Runtime
- Verschiedene Runtimes mit verschiedenen Versionen können gleichzeitig installiert werden
- SDKs sind Entwicklungs-Tools für Bibliotheken und bieten keine Laufzeitumgebung

Flatpak Basis Konzepte

Bundled Libraries

- Wenn Dependencies benötigt werden, aber nicht in der Runtime sind, können sie mit der Anwendung gebüdelt werden
- Verschiedene Dependencies mit verschiedenen Versionen können gleichzeitig installiert werden

Flatpak Basis Konzepte

Sandboxes

- Flatpaks werden in isolierten Umgebungen gebaut und ausgeführt
- Sandboxen enthalten die Anwendung und die Runtime
- Zugriff auf Benutzerdaten, Netzwerk, Grafik Sockets, Bus-System und auf Geräte muss expliziert gewährt werden.
- Notwendige Dateien wie Desktop Files oder Icons werden "exportiert" (Werden auch "exports" genannt)

Flatpak Basis Konzepte

Portals

- Portals sind Möglichkeiten mit dem Host System zu interagieren
- Sie ermöglichen den Zugang zu Dateien und Services ohne zustätzliche Sandbox Berechtigungen
- Beispiele: Dateien öffenen mit dem nativen Datei-Dialog, Öffnen von URIs, Drucken, Benachrichtigung senden, Screenshots erstellen, den Netzwerkstatus abfragen

Flatpak Basis Konzepte

Repositories

- Anwendungen und Runtimes werden in Repositories gehostet
- Sind ähnlich wie Git Repositories aufgebaut
- Jedes Objekt im Repository ist versioniert, sodass Upgrades und Downgrades möglich sind
- Flatpak Systeme können für mehrere Repositories eingerichtet werden
- Standardmäßig wird mindestens das Flathub-Repository verwendet

Flatpak vs. Snap

- Snap ist nicht vollständig Open Source
- Getrennte Anwendung vs. gebündelte Anwendung
- Dezentrale vs. zentralisiertes Repository
- Flatpak startet bei Desktop-Anwendungen im Allgemeinen schneller (20–40%)
- Snap ist bei Ubuntu und Flatpak bei den anderen Distrubutionen

Flatpak vs. AppImage

- AppImage eher portable, self-contained executables, die keine Installation brauchen
- AppImage ist nicht sandboxed und es gibt keine Isolation von Systemressourcen
- AppImages enthalten alle Dependencies
- Flatpaks sind auch für Updates desingt

Flatpak vs. konventionelle Package Manager

- Integration ins System
- Sandbox vs. Non Sandbox
- Oftmals sind Flatpak und das klassisches Package System vorhanden

Flatpak einsetzen

Allgemeine Commands

```
$ flatpak install flathub org.gimp.GIMP
$ flatpak uninstall org.gimp.GIMP
$ flatpak update
$ flatpak list
$ flatpak search gimp
```

Flatpak einsetzen

Commands für Remotes

```
$ flatpak remotes
$ flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
$ flatpak remote-delete flathub
```

Flatpak einsetzen

Ausführen

```
$ flatpak run org.gimp.GIMP
```

Kleiner Tipp, alias setzen oder eine Anwendung dafür verwenden

Flatpak einsetzen

Nützliche Commands

```
$ flatpak history
$ flatpak repair
$ flatpak uninstall --unused
```

Flatpak bauen

Building Introduction / Was braucht man / Infos

- flatpak-builder
- Runtime und SDKs (Auch eigene sind möglich)
- Requirements & Conventions
- Manifest
- Jeder Commit muss signiert werden
- Unterstützt mehrere Build Systeme

Flatpak bauen

flatpak-builder Vorgang

Folgendes passiert, wenn der flatpak-builder einen Build Vorgang ausführt

- Build Verzeichnis erstellen, wenn nicht vorhanden
- Source Code für alle Module herunterladen und verfizieren
- Source Code kompilieren und installieren
- Finished mit Sandbox Permissons
- Das Build Ergebnis, wird in das Repository hochgeladen

Flatpak bauen

Manifest

- Beinhaltet die Informationen zur Anwendung und die Build Instruktionen
- Ist im YAML Format

Flatpak bauen

Manifest

```yaml
id: org.flatpak.Hello
runtime: org.freedesktop.Platform
runtime-version: '25.08'
sdk: org.freedesktop.Sdk
command: hello
modules:
  - name: hello
    buildsystem: simple
    build-commands:
      - install -Dm755 hello.sh /app/bin/hello
    sources:
      - type: script
        dest-filename: hello.sh
        commands:
          - echo "Hello world, from a sandbox"
```

Flatpak bauen

Manifest

Mit finish-args kann man zustätzliche Berechtigungen anfodern

```
finish-args:
  # X11 + XShm access
  - --share=ipc
  - --socket=fallback-x11
  # Wayland access
  - --socket=wayland
  # GPU acceleration if needed
  - --device=dri
  # Needs to talk to the network:
  - --share=network
  # Needs to save files locally
  - --filesystem=xdg-documents
```

Q&A Bereich

Danke für die Aufmerksamkeit! <3